



Kompendium wiedzy dla administratorów



Microsoft®

SQL Server® 2014

Podręcznik administratora

Adam Jorgensen, Bradley Ball, Steven Wort,
Ross LoForte, Brian Knight



Tytuł oryginału: Professional SQL Server® 2014 Administration

Tłumaczenie: Tomasz Walczak

ISBN: 978-83-283-0676-9

Copyright © 2014 by John Wiley & Sons, Inc., Indianapolis, Indiana

All Rights Reserved.

This translation published under license with the original publisher John Wiley & Sons, Inc.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise without either the prior written permission of the Publisher.

The Wrox Brand trade dress is a trademark of John Wiley & Sons, Inc. in the United States and/or other countries. Used by permission.

Wiley, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. SQL Server is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book

Translation copyright © 2015 by Helion S.A.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem: <ftp://ftp.helion.pl/przyklady/sq14pa.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

http://helion.pl/user/opinie/sq14pa_ebook

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

- [Poleć książkę na Facebook.com](#)
- [Kup w wersji papierowej](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

*Książkę dedykuję mojej żonie
(pierwszy raz mogę tak ją nazwać).
Kocham cię, Słonko!
— Adam Jorgensen*

*Dla Coltona — mam nadzieję,
że odkryjesz swojego wewnętrznego geeka.
— Brian Knight*

*Chcę zadedykować tę książkę mojemu tacie, Robertowi
Ballowi. Zawsze mnie inspirowałeś. Twoje rękodzieła,
oddanie rodzinie i integralność zdumiewają mnie.
Im jestem starszy, tym bardziej cię podziwiam.
Chwile z tobą zawsze należały do najcenniejszych
w moim życiu: pozwalałeś mi włączyć syrenę w twoim
służbowym samochodzie, kiedy byłem dzieckiem,
wspinaliśmy się na wzgórze na wybrzeżu Pacyfiku,
aby zobaczyć latarnię morską, mogłem wejść na zaplecze
twojej restauracji, jedliśmy płatki śniadaniowe na plaży,
a kiedyś pozowałeś do zdjęcia przed Białym Domem,
zanim poszedłeś do pracy. Każde moje zwycięstwo jest
także twoim. Ta książka jest jednym z twoich osiągnięć.
Kocham cię, tato.
— Bradley Ball*

Spis treści

O autorach	21
O współpracownikach	23
O redaktorach technicznych	25
Podziękowania	27
Wprowadzenie	29
Rozdział 1. Architektura systemu SQL Server 2014	35
Środowisko systemu SQL Server 2014	35
Nowe ważne funkcje wersji 2014	36
Administratorzy produkcyjnych baz danych	36
Administrator rozwoju baz danych	37
Administratorzy i programiści odpowiedzialni za analizy biznesowe	38
Architektura systemu SQL Server	39
Pliki bazy danych i dziennik transakcji	39
SQL Server Native Client	40
Standardowe systemowe bazy danych	41
Schematy	43
Synonimy	44
Obiekty DMO	44
Typy danych w systemie SQL Server 2014	45
Edycje systemu SQL Server	52
Przegląd edycji	52
Licencje	55
Podsumowanie	56
Rozdział 2. Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014	57
Planowanie systemu	58
Opcje sprzętowe	58
Możliwości w obszarze oprogramowania i instalacji	63

Instalowanie systemu SQL Server	66
Nowe instalacje	66
Instalacje równoległe	66
Aktualizowanie	67
Instalacje nadzorowane	67
Instalacja nienadzorowana	70
Instalowanie usług Analysis Services	76
Tryb UDM	78
Tryb tabelaryczny	78
Instalowanie funkcji PowerPivot for SharePoint	80
Sprawdzanie systemu	81
Konfiguracja po instalacji	81
Ustawianie opcji systemu SQL Server pod kątem wydajności	81
Baza tempdb	83
Konfigurowanie ustawień systemu SQL Server	
w obszarze bezpieczeństwa	85
Narzędzie SQL Server Configuration Manager	86
Tworzenie kopii zapasowej	86
Usuwanie instalacji systemu SQL Server	87
Usuwanie instalacji usług Reporting Services	87
Usuwanie instalacji usług Analysis Services	87
Usuwanie instalacji silnika bazodanowego systemu SQL Server	88
Rozwiązywanie problemów z nieudaną instalacją	88
Podsumowanie	88

Rozdział 3. Najlepsze praktyki aktualizowania systemu

do wersji SQL Server 2014	91
Po co aktualizować system do wersji SQL Server 2014?	92
Ograniczanie ryzyka — wkład Microsoftu	93
Niezależni producenci oprogramowania	
i wkład społeczności skupionej wokół systemu SQL Server	93
Aktualizowanie systemu do wersji SQL Server 2014	94
Aktualizacja w miejscu	94
Aktualizowanie przy równoległe zainstalowanych wersjach	96
Wybór między aktualizowaniem w miejscu	
a podejściem z równoległymi egzemplarzami	97
Kroki i narzędzia potrzebne przed instalacją	97
Zadania wykonywane przed aktualizacją	97
Narzędzia przydatne przed aktualizacją	98
Zgodność wstecz	106
Funkcje nieobsługiwane lub wycofane w wersji SQL Server 2014	106
Funkcje baz danych uznane za przestarzałe w wersji SQL Server 2014	108
Inne zmiany w działaniu funkcji w wersji SQL Server 2014	108

Uwagi dotyczące komponentów systemu SQL Server	109
Aktualizowanie katalogu pełnotekstowego	109
Aktualizowanie usług Reporting Services	109
Aktualizowanie do wersji 64-bitowej	110
Testy po aktualizacji	110
Podsumowanie	111

Rozdział 4. Zarządzanie silnikiem bazodanowym i rozwiązywanie związanych z nim problemów 113

Narzędzia konfiguracyjne i administracyjne	114
SQL Server Configuration Manager	114
Parametry uruchomieniowe	116
Uruchomieniowe procedury składowane	118
Częściowo niezależne bazy danych	120
Narzędzia do rozwiązywania problemów	121
Połączenie DAC	121
Odtwarzanie systemowych baz danych	122
SQL Server Management Studio	123
Raporty	124
Konfigurowanie systemu SQL Server	
w narzędziu SQL Server Management Studio	126
Filtrowanie obiektów	131
Dzienniki błędów	131
Narzędzie Activity Monitor	132
Monitorowanie procesów w języku T-SQL	136
Procedury sp_who i sp_who2	137
Widok sys.dm_exec_connections	137
Funkcja sys.dm_exec_sql_text	138
Zarządzanie wieloma serwerami	138
Centralne serwery zarządzania i grupy serwerów	139
Opcje śledzenia	140
Udzielanie pomocy przez obsługę techniczną	141
Narzędzie SQLDumper.exe	141
Narzędzie SQLDiag.exe	142
Podsumowanie	144

Rozdział 5. Automatyzowanie pracy systemu SQL Server 147

Plany konserwacji	148
Kreator planów konserwacji	148
Projektant planów konserwacji	153
Automatyzowanie systemu SQL Server	
za pomocą narzędzia SQL Server Agent	156
Zadania	156
Harmonogramy	162
Operatorzy	163
Alerty	166

Zabezpieczenia w narzędziu SQL Server Agent	171
Konto usługowe	171
Dostęp do narzędzia SQL Server Agent	172
Jednostki pośredniczące narzędzia SQL Server Agent	173
Konfigurowanie narzędzia SQL Server Agent	176
Właściwości z kategorii General	176
Właściwości z kategorii Advanced	177
Właściwości z kategorii Alert System	178
Właściwości z kategorii Job System	179
Właściwości z kategorii Connection	180
Właściwości z kategorii History	180
Zarządzanie wieloma serwerami	180
Podstawianie tokenów	181
Przekazywanie zdarzeń	183
Używanie usług WMI	184
Zarządzanie wieloma serwerami	
— używanie serwerów nadrzędnych i docelowych	185
Podsumowanie	187
Rozdział 6. Service Broker w systemie SQL Server 2014	189
Asynchroniczne przesyłanie komunikatów	189
Ogólne omówienie Service Brokera	191
Service Broker a inne kolejki komunikatów	192
Konfigurowanie Service Brokera	193
Ustawianie stanu Service Brokera	193
Typy komunikatów	195
Kontrakty	196
Kolejki	197
Usługi	199
Trasy	199
Priorytety	201
Grupy konwersacji	202
Używanie Service Brokera	203
Wysyłanie komunikatów	203
Odbieranie komunikatów	206
Przesyłanie komunikatów między egzemplarzami	207
Zewnętrzna aktywacja	209
Przykład ilustrujący zapisywanie danych o użytkownikach w dzienniku	210
Podsumowanie	217
Rozdział 7. Integrowanie systemu SQL Server ze środowiskiem CLR	219
Wprowadzenie do środowiska CLR	220
System SQL Server jako host środowiska	
uruchomieniowego platformy .NET	221
Domeny aplikacji	221
Język T-SQL a środowisko CLR	222
Włączanie integracji ze środowiskiem CLR	222

Tworzenie komponentów CLR	223
Metoda bez używania środowiska Visual Studio	224
Używanie narzędzi Microsoft SQL Server Data Tools	226
Zabezpieczenia przy integracji ze środowiskiem CLR	227
Monitorowanie wydajności	228
Windows System Monitor	228
Rozszerzone zdarzenia	230
Widoki DMV	231
Cele projektowe dotyczące integracji ze środowiskiem CLR	231
Podsumowanie	232
Rozdział 8. Zabezpieczanie egzemplarzy baz danych	233
Rodzaje uwierzytelniania	234
Uwierzytelnianie z systemu SQL Server	234
Uwierzytelnianie z systemu Windows	235
Porównanie obu metod uwierzytelniania	236
Loginy i użytkownicy	236
Autoryzacja dostępu	237
Uprawnienia na serwerze	237
Zabezpieczane obiekty baz danych	243
Łańcuchy uprawnień	244
Łańcuchy uprawnień obowiązujące między bazami	245
Zabezpieczenia na poziomie wierszy	247
Podsumowanie	248
Rozdział 9. Technologia In-Memory OLTP	249
Stosowanie i implementowanie technologii In-Memory OLTP	249
Włączanie technologii In-Memory OLTP	250
Struktury tabel używanych w technologii In-Memory OLTP	252
Rekordy	252
Indeksy	253
Kwestie związane z procesorem	258
Zagadnienia związane z wirtualizacją	258
Kwestie związane z pamięcią	258
Zarządzanie pamięcią za pomocą funkcji Resource Governor	261
Tworzenie natywnie kompilowanych procedur składowanych	263
Omówienie narzędzia ARM	266
Podsumowanie	278
Rozdział 10. Konfigurowanie serwera pod kątem optymalnej wydajności	279
Co każdy administrator baz danych powinien	
wiedzieć na temat wydajności?	280
Cykl dostrajania wydajności	280
Konfiguracja	282
Plan zasilania	282
Natychniastowe inicjowanie plików bazy danych	284

Opcje śledzenia	287
Definicja dobrej wydajności	287
Koncentracja na tym, co najważniejsze	288
Co administrator programista powinien wiedzieć na temat wydajności?	289
Użytkownicy	289
Instrukcje w języku SQL	290
Wzorce używania danych	290
Schemat bazy danych	290
Co o wydajności powinien wiedzieć administrator	
produkcyjnej bazy danych?	291
Optymalizowanie sprzętu serwera	292
Zarządzanie sprzętem	293
Procesor	294
Architektura x64	294
Pamięć podręczna	295
Technologia Hyper-Threading	295
Pojęcia związane z systemami wielordzeniowymi	296
Pamięć	297
Pamięć fizyczna	297
Fizyczna przestrzeń adresowa	298
Menedżer pamięci wirtualnej	298
Plik stronicowania	299
Błędy strony	300
Operacje wejścia-wyjścia	300
Sieć	301
Dyski magnetyczne	302
Dyski SSD (dyski z pamięcią flash)	305
Rozważania związane z przechowywaniem danych	305
Podsumowanie	309

Rozdział 11. Optymalizowanie systemu SQL Server 2014 311

Optymalizowanie aplikacji	311
Definiowanie obciążenia	312
Cichy zabójca — problemy z operacjami wejścia-wyjścia	312
Model obsługi wejścia-wyjścia w systemie SQL Server	312
Rozmieszczanie plików bazy danych	313
Kwestie związane z bazą tempdb	314
Wewnętrzne mechanizmy systemu SQL Server i alokacja plików	318
Podział tabel i indeksów na partycje	320
Po co stosować partycje?	320
Tworzenie funkcji partycjonującej	322
Tworzenie grup plików	324
Tworzenie schematu partycjonowania	324
Kompresja danych	325
Kompresja wierszy	326
Kompresja stron	328

Szacowanie oszczędności miejsca	329
Monitorowanie kompresji danych	331
Uwagi dotyczące kompresji danych	332
System SQL Server i procesory	333
Architektura NUMA i dynamicznie dodawane procesory	334
Spójność pamięci podręcznej	334
Maska koligacji	336
Ustawienie Max Degree of Parallelism (MAXDOP)	336
Ustawienie Cost Threshold for Parallelism	337
Uwagi na temat pamięci i związanych z nią usprawnień	339
Rozszerzenia puli buforów	339
Dostrajanie pamięci systemu SQL Server	341
Lokalność danych	343
Maksymalna ilość pamięci dla serwera	343
Resource Governor	344
Podstawowe elementy technologii Resource Governor	345
Używanie technologii Resource Governor w programie SQL Server 2014 Management Studio	349
Monitorowanie pracy mechanizmu Resource Governor	350
Podsumowanie	351

Rozdział 12. Monitorowanie systemu SQL Server 353

Cele monitorowania	354
Określanie celów monitorowania	354
Określanie punktu odniesienia	355
Porównywanie aktualnych wartości z punktem odniesienia	355
Wybór odpowiednich narzędzi monitorujących	356
Monitor wydajności	358
Liczniki dotyczące zasobów procesora	359
Aktywność dysków	360
Użytkowanie pamięci	366
Narzędzia do monitorowania wydajności	369
Monitorowanie zdarzeń	371
Domyślny ślad	373
Sesja system_health	374
SQL Trace	374
Powiadomienia o zdarzeniach	378
Zdarzenia rozszerzone systemu SQL Server	381
Monitorowanie systemu za pomocą widoków DMV i funkcji DMF	400
Co dzieje się w systemie SQL Server?	401
Wyświetlanie informacji o blokadach z bazy danych	404
Wyświetlanie informacji o blokadach z serwera	405
Używanie indeksu w bazie danych	405
Indeksy nieużywane w bazie danych	406
Wyświetlanie kwerend oczekujących na pamięć	407

Informacje o podłączonych użytkownikach	408
Wolna przestrzeń w grupach plików	408
Plan wykonywania i kod aktualnie działających kwerend	409
Zajmowana pamięć	409
Zajęta pamięć w puli buforów	409
Monitorowanie dzienników	410
Monitorowanie dziennika błędów systemu SQL Server	410
Monitorowanie dzienników zdarzeń systemu Windows	411
Standardowe raporty systemu SQL Server	411
Narzędzie System Center Advisor	411
Podsumowanie	414

Rozdział 13. Dostrajanie wydajności kodu w języku T-SQL 415

Omówienie procesu przetwarzania kwerend	416
Generowanie planów wykonywania	418
Statystyki	418
Usprawnienia optymalizatora kwerend w systemie SQL Server 2014	
— nowy mechanizm szacowania kardynalności	419
Używanie nowego mechanizmu szacowania kardynalności	419
Identyfikowanie problemów przy dostrajaniu	
wydajności kwerend w języku SQL	420
Monitorowanie wydajności kwerend	420
Co robić po wykryciu kwerendy o niskiej wydajności?	421
Generowanie planów wykonywania kwerend	423
Czytanie planów wykonywania kwerend	424
Operatory dostępu do danych w planach wykonywania kwerend	427
Skanowanie tabeli	428
Skanowanie indeksów klastrowanych	429
Skanowanie indeksów nieklastrowanych	432
Łączenie operatorów dostępu	434
Operatory złączeń	435
Pętla zagnieżdżona (złączanie w pętli)	435
Złączenia z haszowaniem	437
Złączanie przez scalanie	438
Plany wykonywania kwerend modyfikujących dane	440
Przetwarzanie kwerend działających na tabelach i indeksach	
podzielonych na partycje	442
Operacje, na które wpływa podział na partycje	442
Strategie współbieżnego wykonywania kwerend	
dla obiektów podzielonych na partycje	444
Analizowanie wydajności kwerend w środowisku produkcyjnym	444
Systemowe widoki DMV	445
Łączenie wszystkich elementów	446
Podsumowanie	447

Rozdział 14. Indeksowanie baz danych	449
Nowe mechanizmy w indeksach w systemie SQL Server 2014	450
Indeksy i tabele podzielone na partycje	454
Omówienie indeksów	454
Tworzenie indeksów	459
Używanie tabel i indeksów podzielonych na partycje	461
Konserwacja indeksów	461
Monitorowanie poziomu fragmentacji indeksu	462
Porządkowanie indeksów	463
Zwiększanie wydajności kwerend za pomocą indeksów	464
Narzędzie DTA	469
Podsumowanie	472
Rozdział 15. Replikacja	473
Wprowadzenie do procesu replikacji	474
Elementy w procesie replikacji	474
Typy replikacji	476
Modele replikacji	478
Jeden wydawca i dowolna liczba subskrybentów	478
Wielu wydawców i jeden subskrybent	480
Wielu wydawców będących jednocześnie subskrybentami	480
Subskrybent aktualizujący dane	481
Model P2P	481
Implementowanie replikacji	483
Konfigurowanie replikacji migawkowej	483
Konfigurowanie dystrybucji	483
Implementowanie replikacji migawkowej	487
Implementowanie replikacji transakcyjnej i ze skalaniem	500
Replikacja w trybie P2P	501
Konfigurowanie replikacji w trybie P2P	501
Konfigurowanie architektury dla replikacji P2P	502
Replikacja obsługiwana za pomocą skryptów	505
Monitorowanie replikacji	506
Monitor replikacji	506
Monitor wydajności	508
Widoki DMV związane z replikacją	508
Procedura sp_replcounters	509
Podsumowanie	509
Rozdział 16. Klastrowanie w systemie SQL Server 2014	511
Klastrowanie a sytuacja firmy	512
Co można uzyskać przy użyciu klastrow?	512
Czego klastrowanie nie umożliwia?	513
Właściwe powody stosowania klastrowania	
dla systemu SQL Server 2014	514
Co zastosować zamiast klastrowania?	514

Klastrowanie — ogólne omówienie	517
Jak działa klastrowanie?	517
Rodzaje klastrów	522
Aktualizowanie klastrów z systemem SQL Server	524
Rezygnacja z aktualizacji	524
Aktualizowanie klastra do wersji SQL Server 2014 w miejscu	524
Tworzenie klastra od nowa	525
Plan wycofywania zmian	527
Która metoda aktualizacji jest najlepsza?	527
Przygotowanie do klastrowania	527
Przygotowywanie infrastruktury	528
Przygotowywanie sprzętu	528
Tworzenie klastra w systemie operacyjnym Windows Server 2012 R2	531
Co trzeba zrobić przed instalacją klastra w systemie Windows 2012 R2?	531
Instalowanie klastra WSFC z systemu Windows Server 2012 R2	532
Przygotowywanie systemu Windows Server 2012 R2 do pracy w klastrze	535
Używanie koordynatora MSDTC w klastrze	536
Dodawanie systemu SQL Server 2014 do klastra	537
Instrukcje instalowania systemu SQL Server w klastrze	538
Instalowanie pakietów poprawek SP	
i aktualizacji typu Cumulative Update	544
Testuj, testuj i jeszcze raz testuj	544
Zarządzanie klastrem i monitorowanie go	546
Rozwiązywanie problemów z klastrem	547
Jak podejść do rozwiązywania problemów z klastrami WSFC?	547
Właściwe rozwiązanie już za pierwszym razem	548
Zbieranie informacji	548
Rozwiązywanie problemów	549
Współpraca z Microsoftem	549
Podsumowanie	549

Rozdział 17. Kopie zapasowe i przywracanie stanu551

Usprawnienia związane z tworzeniem kopii zapasowej	
i przywracaniem stanu	552
Tworzenie kopii zapasowych z zapisem na adres URL	552
Zarządzane tworzenie kopii zapasowych	
z zapisem w platformie Windows Azure	552
Szyfrowanie kopii zapasowych	553
Przegląd procesu tworzenia kopii zapasowej i przywracania stanu	553
Jak przebiega tworzenie kopii zapasowej?	553
Kopiowanie baz danych	556
Kompresja kopii zapasowej	568
Modele odzyskiwania	569
Wybór modelu odzyskiwania	571
Przełączanie modeli odzyskiwania	572

Tworzenie tabel z historią kopii zapasowych	573
Uprawnienia potrzebne do tworzenia kopii zapasowych i przywracania stanu	574
Tworzenie kopii zapasowych systemowych baz danych	575
Kopie zapasowe indeksów pełnotekstowych	576
Sprawdzanie poprawności obrazów z kopią zapasową	576
Przebieg przywracania	577
Przygotowania do odzyskiwania	579
Wymagania związane z odzyskiwaniem	579
Wzorce użytkowania danych	580
Czas na konserwację	581
Inne rozwiązania zapewniające wysoką dostępność	582
Opracowywanie i wykonywanie planu tworzenia kopii zapasowych	583
Używanie programu SQL Server Management Studio	583
Plany konserwacji bazy danych	587
Używanie instrukcji tworzenia kopii zapasowych z języka T-SQL	591
Zarządzanie kopiami zapasowymi	593
Wydajność tworzenia i przywracania kopii zapasowych	593
Odzyskiwanie baz danych	594
Proces przywracania	594
Przywracanie baz za pomocą programu SQL Server Management Studio	598
Polecenie RESTORE z języka T-SQL	602
Przywracanie systemowych baz danych	603
Archiwizowanie danych	604
Podział tabel w systemie SQL Server na partycje	605
Widoki podzielone na partycje	606
Podsumowanie	606
Rozdział 18. Przesyłanie dzienników w systemie SQL Server 2014	607
Scenariusze stosowania przesyłania dzienników	608
Przesyłanie dzienników w celu utworzenia aktywnego serwera rezerwowego	608
Przesyłanie dzienników jako mechanizm odzyskiwania systemu po katastrofie	609
Przesyłanie dzienników jako mechanizm tworzenia baz potrzebnych do generowania raportów	610
Architektura mechanizmu przesyłania dzienników	611
Serwer główny	612
Serwer pomocniczy	612
Serwer monitorujący	612
Proces przesyłania dzienników	613
Wymagania systemowe	613
Sieć	613
Serwery o identycznych możliwościach	614
Przechowywanie danych	614
Oprogramowanie	614

Wdrażanie przesyłania dzienników	615
Wstępne konfigurowanie	615
Wdrażanie przesyłania dzienników za pomocą programu	
SQL Server Management Studio	616
Wdrażanie przesyłania dzienników za pomocą instrukcji języka T-SQL	624
Monitorowanie systemu i rozwiązywanie problemów	625
Monitorowanie z wykorzystaniem programu	
SQL Server Management Studio	626
Monitorowanie z wykorzystaniem procedur składowanych	626
Rozwiązywanie problemów	627
Zarządzanie zmianą ról	627
Synchronizowanie zależności	627
Przełączanie ról między serwerami głównym i pomocniczym	630
Przełączanie między serwerami głównym i pomocniczym	632
Przekierowywanie klientów na serwer pomocniczy	633
Plan tworzenia kopii zapasowych baz danych	634
Integrowanie przesyłania dzienników z innymi rozwiązaniami	
zapewniającymi wysoką dostępność	635
Kopie lustrzane danych w systemie SQL Server 2014	635
Klastry WSFC	635
Replikacja w systemie SQL Server 2014	636
Kończenie przesyłania dzienników	637
Kończenie przesyłania dzienników w programie SQL Server	
Management Studio	637
Kończenie przesyłania dzienników za pomocą języka T-SQL	637
Wydajność przesyłania dzienników	638
Aktualizacja przesyłania dzienników do wersji z systemu SQL Server 2014	639
Podejście z minimalnym przestojem	639
Podejście ze standardowym przestojem	639
Podejście z instalowaniem przesyłania dzienników	640
Podsumowanie	640

Rozdział 19. Kopie lustrzane baz danych 643

Wprowadzenie do kopii lustrzanych baz danych	644
Tryby działania kopii lustrzanych	645
Przykłady tworzenia kopii lustrzanych baz danych	648
Przygotowywanie punktów końcowych	648
Przygotowywanie bazy danych do tworzenia kopii lustrzanych	653
Początkowa synchronizacja serwerów głównego i lustrzanego	654
Uruchamianie sesji tworzenia kopii lustrzanych	655
Tryb wysokiego bezpieczeństwa bez automatycznego	
przełączania awaryjnego	656
Tryb wysokiego bezpieczeństwa z automatycznym	
przełączaniem awaryjnym	657
Tryb wysokiej wydajności	658

Kopie lustrzane i różne edycje systemu SQL Server 2014	659
Zmiana ról w mechanizmie tworzenia kopii lustrzanych	659
Automatyczne przełączanie awaryjne	660
Ręczne przełączanie awaryjne	662
Wymuszone przełączenie awaryjne	664
Używanie monitora tworzenia kopii lustrzanych	665
Ustawianie wartości progowych liczników i wysyłanie alertów	668
Przygotowywanie serwera lustrzanego do przełączenia awaryjnego	670
Sprzęt, oprogramowanie i konfiguracja serwera	670
Dostępność bazy danych w trakcie planowanego przestoju	672
Konfigurowanie zadań narzędzia SQL Server Agent	
na serwerze lustrzanym	673
Przekierowywanie klientów na serwer lustrzany	674
Tworzenie kopii lustrzanych baz i inne metody	
zapewniania wysokiej dostępności	675
Kopie lustrzane baz danych a klastry	675
Kopie lustrzane baz danych a replikacja transakcyjna	676
Kopie lustrzane baz danych a przesyłanie dzienników	676
Kopie lustrzane baz danych a grupy dostępności	676
Snapshoty baz danych	677
Podsumowanie	678

Rozdział 20. Zarządzanie usługami Integration Services

i dostrajanie ich wydajności	679
Przegląd usług SSIS	680
Zastosowania usług SSIS	680
Główne elementy usług SSIS	681
Zarządzanie projektem i kontrola zmian	683
Zarządzanie usługą SSIS	683
Wprowadzenie do usługi SSIS	683
Konfiguracja	684
Dzienniki zdarzeń	687
Monitorowanie aktywności	688
Zarządzanie pakietami SSIS w modelu z instalowaniem pakietów	688
Zarządzanie pakietami za pomocą programu	
SQL Server Management Studio	689
Instalowanie	691
Zarządzanie pakietami SSIS w modelu z instalacją projektów	694
Konfigurowanie katalogu SSIS	695
Instalowanie pakietów	696
Konfigurowanie pakietów	697
Wykonywanie i planowanie uruchamiania pakietów	701
Uruchamianie pakietów w narzędziu SSDT	701
Uruchamianie pakietów za pomocą kreatora Import and Export Wizard	701
Uruchamianie pakietów za pomocą narzędzia DTEXEC	702

Uruchamianie pakietów za pomocą narzędzia DTExecUI (w modelu z instalowaniem pakietów)	702
Uruchamianie pakietów za pomocą narzędzia Execute Package (w modelu z instalowaniem projektów)	703
Planowanie wykonywania za pomocą narzędzia SQL Server Agent	704
Uruchamianie pakietów za pomocą języka T-SQL	706
Zabezpieczanie usług SSIS	706
Przegląd zabezpieczeń usług SSIS	706
Zabezpieczenia w modelu z instalowaniem pakietów	707
Podsumowanie	709

Rozdział 21. Zarządzanie usługami Analysis Services i dostrajanie ich wydajności 711

Przegląd usług SSAS	712
Elementy modelu MOLAP	713
Elementy modelu tabelowego	713
Elementy architektury serwera SSAS	714
Zarządzanie serwerem SSAS	715
Właściwości serwera	715
Wymagane usługi	717
Język ASSL	717
Zarządzanie bazami z serwera SSAS	719
Instalowanie baz na serwerach SSAS	719
Przetwarzanie obiektów SSAS	721
Tworzenie i przywracanie kopii zapasowych baz SSAS	727
Synchronizowanie baz SSAS	729
Monitorowanie i dostrajanie wydajności usług SSAS	730
Monitorowanie zdarzeń SSAS	731
Używanie narzędzia Flight Recorder do analiz następnych	731
Podsumowanie	732

Rozdział 22. Zarządzanie usługami Reporting Services w systemie SQL Server 733

Narzędzie Configuration Manager usług SSRS	734
Konto usługowe	736
Adres URL usługi sieciowej	737
Bazy SSRS	739
Adres URL narzędzia Report Manager	741
Ustawienia poczty elektronicznej	742
Konto wykonawcze	743
Klucze szyfrowania	744
Skalowanie systemu	745
Dziennik wykonywania raportów	746
Report Manager	747
Używanie narzędzia Report Manager	747
Zarządzanie raportami	754
Podsumowanie	767

Rozdział 23. Integracja systemu SQL Server 2014 z platformą SharePoint 2013	769
Komponenty uczestniczące w integracji	770
PowerPivot	770
Usługi Reporting Services	772
Power View	775
Odświeżanie danych	776
Używanie połączeń z danymi w Excelu	777
Połączenia danych z narzędzia PerformancePoint	782
Odświeżanie danych w usługach Visio	783
Odświeżanie danych w dodatku PowerPivot	785
Podsumowanie	794
Rozdział 24. Administrowanie bazami SQL Database i ich konfigurowanie	795
Wprowadzenie do technologii Windows Azure SQL Database	795
Architektura bazy SQL Database	796
Warstwa klientów	797
Warstwa usług	797
Warstwa platformy	797
Warstwa infrastruktury	797
Różnice między środowiskami	797
Konfigurowanie baz SQL Database	798
Udostępnianie serwera i bazy danych	798
Ograniczanie dostępu do zasobów i równoważenie obciążenia	805
Konfigurowanie zapory dla baz SQL Database	806
Nawiązywanie połączenia z bazami SQL Database	807
Administrowanie bazami SQL Database	809
Tworzenie loginów i kont użytkowników	809
Przypisywanie uprawnień dostępu	811
Praca z bazami SQL Database	812
Tworzenie kopii zapasowych baz SQL Database	813
Okno Object Explorer dla baz SQL Database	813
Czego brakuje w bazach SQL Database?	815
Podsumowanie	816
Rozdział 25. Grupy dostępności AlwaysOn	817
Architektura	818
Repliki i role w grupach dostępności	819
Tryby dostępności	819
Obsługiwane typy przełączania awaryjnego	820
Umożliwianie dostępu do replik pomocniczych w trybie tylko do odczytu	821
Przykład ilustrujący grupę dostępności	822
Konfigurowanie nowej grupy dostępności	823
Konfigurowanie istniejących grup dostępności	830
Przełączanie awaryjne w grupach dostępności	834
Zawieszanie pracy baz wysokiej dostępności	836

Wznawianie pracy baz wysokiej dostępności837

Połączenia aplikacji klienckich837

Aktywna replika pomocnicza udostępniana w trybie tylko do odczytu839

Dostęp w trybie tylko do odczytu839

Łączenie się klientów z repliką pomocniczą840

Wydajność841

Tworzenie kopii zapasowych w replice pomocniczej842

Analizowanie metadanych z replik,
w których wykonywane są kopie zapasowe843

Panel kontrolny grup dostępności AlwaysOn844

Monitorowanie i rozwiązywanie problemów846

Podsumowanie847

Skorowidz.....849

O autorach



Adam Jorgensen to nagradzany i znany na całym świecie lider odnoszący sukcesy w dużych firmach oraz organizacjach nastawionych na rozwój. Obecnie koncentruje się na pomaganiu klientom w budowaniu opartej na danych kultury, w której wykorzystuje się odpowiednie rozwiązania analityczne do identyfikowania, analizowania i wykorzystywania okazji biznesowych. Jorgensen kieruje strategicznymi i operacyjnymi aspektami działania firmy Pragmatic Works oraz służy społeczności jako wiceprezes ds. wykonawczych i dyrektor stowarzyszenia PASS (jest to największa na świecie organizacja zrzeszająca profesjonalistów zajmujących się danymi). Obie te jednostki pomagają profesjonalistom poznawać nowe i przydatne metody pracy z danymi.



Bradley Ball posiada uzyskane w latach 2005 i 2008 certyfikaty MCITP (ang. *Microsoft Certified IT Professional*) dla administratorów baz danych, ma też ponad piętnaście lat doświadczenia w branży. Przez osiem lat pracował jako kontraktor Departamentu Obrony Stanów Zjednoczonych. Jego klientami były takie organizacje jak Armia Stanów Zjednoczonych, Urząd Wykonawczy Prezydenta Stanów Zjednoczonych i supermarkety Publix. Obecnie Bradley jest kierownikiem ds. zarządzania platformą danych w firmie Pragmatic Works. Ponadto posiada tytuł VTSP (ang. *Virtual Technology Solution Professional*) Microsoftu i współpracuje z tą firmą w zakresie tworzenia rozwiązań technicznych, rozwiązywania problemów i promowania systemu SQL Server. Wygłaszał wiele odczytów z cyklu SQL Saturday, a także prowadził w całych Stanach Zjednoczonych (od Kalifornii po Florydę i New Hampshire) wykłady dla grup SSUG (ang. *SQL Server User Group*) organizacji PASS. Był też wykładowcą na konferencjach SQL Rally, DevConnections, SQL Live 360 i PASS Summit 2011, 2012 i 2013. Obecnie mieszka w środkowej części stanu Floryda. Jego blog na temat systemu SQL Server i innych interesujących Bradleya zagadnień znajdziesz pod adresem <http://www.sqlballs.com>.



Steven Wort pracował z systemem SQL Server od 1993 roku. Zaczynał od wersji 4.2 działającej w systemie operacyjnym OS2. Legitymuje się ponad trzydziestoletnim doświadczeniem w tworzeniu aplikacji w branży IT. Został zatrudniony w Microsoftzie w 2000 roku jako inżynier odpowiedzialny za wsparcie techniczne w zespole SIE (ang. *Systems Integration Engineering*), gdzie był współautorem wielu warsztatów poświęconych debugowaniu systemu Windows i platformy .NET. W roku 2004 trafił do zespołu zajmującego się systemem SQL Server, gdzie pracował nad skalowalnością wersji SQL Server 2005. Po krótkim epizodzie w grupie odpowiedzialnej za system Windows (zajmował się tam skalowaniem dużych systemów bazodanowych) wrócił do zespołu pracującego nad systemem SQL Server, gdzie rozwijał narzędzia dla tego systemu. Obecnie jest architektem w zespole opracowującym usługi CRM i koncentruje się na wydajności oraz skalowalności systemu SQL Server. Wart jest też współautorem kilku książek poświęconych zarządzaniu systemem SQL Server, rozwiązywaniu związanych z nim problemów i poprawie wydajności tego systemu.



Ross LoForte jest architektem technologii w centrum technologicznym Microsoftu w Chicago i zajmuje się głównie rozwiązaniami związanymi z systemem SQL Server. LoForte ma ponad dwadzieścia lat doświadczenia w tworzeniu oprogramowania dla biznesu, zarządzania projektami i budowaniu rozwiązań opartych na języku SQL. Od trzynastu lat pracuje w centrach technologicznych Microsoftu, gdzie kierował projektowaniem architektury i weryfikowaniem koncepcji rozwiązań dla największych oraz najważniejszych klientów Microsoftu. W ramach tej pracy projektował duże i krytyczne rozwiązania oparte na systemie SQL Server. LaForte prowadzi też wykłady dotyczące systemu SQL Server na Uniwersytecie DePaula w Chicago oraz regularnie występuje na spotkaniach SQL Saturday, TechEd, SQL PASS, Gartner, TDWI i wewnętrznych konferencjach Microsoftu. Ponadto publikuje książki i od wielu lat jest aktywnym członkiem grup Professional Association for SQL Server i Chicago SQL Server Users Group oraz społeczności związanej z systemem SQL Server.



Brian Knight jest współwłaścicielem i założycielem organizacji Pragmatic Works, a także „seryjnym” przedsiębiorcą tworzącym inne firmy. Knight prowadził kolumny w kilku magazynach technicznych i regularnie udostępnia webcasty. Jest autorem kilkunastu książek technicznych. Występował na dziesiątkach konferencji, takich jak PASS, SQL Connection, TechEd i Code Camp. Otrzymał liczne nagrody stanowe, prasowe (Top CEO *Jacksonville Magazine*) i gubernatora (Business Ambassador Award). Prowadzi blog na stronie <http://www.bidn.com>.

O współpracownikach

Chad Churchwell to doświadczony profesjonalista z branży baz danych. Ma ponad dziesięć lat doświadczenia w pracy z systemem SQL Server. Obecnie zajmuje stanowisko inżyniera PFE w Microsoftzie. Pracował też jako konsultant ds. systemu SQL Server i zarządzał zespołem administratorów baz danych w dziale operacyjnym. Jest bardzo zaangażowany w życie społeczności skupionej wokół systemu SQL Server (poświęca swój czas m.in. na prowadzenie wykładów na spotkaniach SQL Saturday na południowym wschodzie i na konferencjach SQL Live 360). W wolnym czasie można go znaleźć w okolicach Jacksonville Beach, gdzie rozkoszuje się przyjemnym florydzkim klimatem.

Dan Clark jest starszym konsultantem ds. analiz biznesowych i programistą specjalizującym się w technologiach Microsoftu. Uwielbia poznawać nowe technologie i uczyć innych, jak optymalnie je stosować. Clark napisał kilka książek poświęconym platformie .NET i programowaniu baz danych. Ponadto regularnie prowadzi prelekcje na konferencjach dla programistów i osób pracujących z bazami danych, a także na spotkaniach grup użytkowników.

Kim Hathaway jest starszą konsultantką z zakresu baz danych (dwadzieścia siedem lat doświadczenia w pracy z różnymi systemami zarządzania bazami danych). Specjalizuje się w rozwiązywaniu problemów z błędami i wydajnością, zarządzaniu wieloma klientami za pomocą usług bazodanowych, ocenie stanu baz SQL-owych, tworzeniu projektów, aktualizowaniu platform SQL-owych, rekomendowaniu i wdrażaniu rozwiązań zapewniających wysoką dostępność systemu, a także rozwijaniu rozwiązań SQL-owych dostosowanych do potrzeb firm. Ponadto posiada praktyczne doświadczenie w pracy z wieloma narzędziami Microsoftu przeznaczonymi do analiz biznesowych. Jej misją jest zapewnianie wysokiej jakości kompletnych rozwiązań wszelkich problemów biznesowych. Nieustannie rozwija swoje umiejętności, poznając nowe technologie, a także pracuje dla społeczności i prowadzi wykłady na różnych imprezach.

Bradley Schacht to starszy konsultant i szkoleniowiec w firmie Pragmatic Works w Jacksonville na Florydzie. Jest współautorem trzech książek na temat SharePointa i systemu SQL Server. Ma doświadczenie w zakresie SharePointa 2010, SharePointa 2013, hurtowni danych i platformy Microsoftu do analiz biznesowych. Pomagał wielu firmom w udanym tworzeniu i wdrażaniu nowych rozwiązań związanych z analizami biznesowymi. Schacht często prowadzi wykłady na imprezach w całym kraju. Publikuje teksty w witrynie <http://SQLServerCentral.com> i innych, a także jest aktywnym członkiem grupy JSSUG (ang. *Jacksonville SQL Server User Group*).

Jorge Segarra (znany też jako SQLChicken) posiada tytuł SQL Server MVP (ang. *Most Valuable Professional*) i jest autorem, blogerem oraz wykładowcą. Jorge mieszka pod Tampą na Florydzie i obecnie jest starszym konsultantem ds. platformy danych w firmie Pragmatic Works. Specjalizuje się w wirtualizacji, technologii PDW (ang. *Parallel Data Warehouse*), dostrajaniu wydajności i pisanu na swój temat w trzeciej osobie.

Kathy Vick jest starszą konsultantką ds. analizy biznesowej w firmie Pragmatic Works. Przez dwanaście lat była konsultantką w Microsoftzie, a przez dwa lata pracowała dla serwisu Expedia.com. Używa systemu SQL Server od 1994 roku i w trakcie swojej kariery koncentrowała się na danych. Podoba się jej życie konsultantki i współpraca z wieloma dużymi firmami z listy Fortune 100 możliwa dzięki zatrudnieniu w Microsoftzie i Pragmatic Works. Vick prowadzi blog na stronie <http://mskathy.com> i ma konto @MsKathyV na Twitterze. Jej pasją jest fotografia, a ponadto uwielbia podróżować. Mieszka pod Seattle z mężem, dziećmi i wnukami.

Roger Wolter od ponad dwudziestu pięciu lat zajmuje się bazami danych i systemami rozproszonymi. Posiada duże doświadczenie w tworzeniu rozbudowanych, skomplikowanych aplikacji dla systemu Windows, w których ważne są: bazy danych, przetwarzanie transakcji, architektura klient-serwer, niezawodne przekazywanie komunikatów, interfejsy komunikacyjne i technologia MDM (ang. *Master Data Management*). W trakcie dwunastu lat pracy w Microsoftzie był menedżerem programu w ramach prac nad systemem SQL Server, a także architektem w zespołach zajmujących się różnymi produktami. Uczestniczył w pracach nad związanymi z systemem SQL Server narzędziami Service Broker, SQLXML i SQL Express. Po odejściu w 2010 roku z Microsoftu Wolter jest architektem baz danych specjalizującym się w wydajności, konsolidowaniu danych oraz technologiach PDW (ang. *Parallel Data Warehouse*) i MDS (ang. *Master Data Services*).

O redaktorach technicznych

Kathi Kellenberger jest konsultantką ds. analiz biznesowych i była administratorką baz danych. Ponadto posiada tytuł SQL Server MVP. Kellenberger lubi uczyć innych, prowadzić wykłady i pisać na temat systemu SQL Server. Jest autorką lub współautorką kilku książek na temat systemu SQL Server. Gdy nie pracuje, chętnie spędza czas z rodziną i przyjaciółmi. Jeśli słyszałeś kiedyś o kanale *#sqlkaraoke*, wiedz, że powstał on właśnie dzięki Kellenberger. To ona na konferencji PASS Summit w 2006 razem z trójką przyjaciół zorganizowała pierwszą imprezę karaoke.

Jason Strate jest architektem rozwiązań w firmie Pragmatic Works. Pracuje z systemem SQL Server od ponad piętnastu lat. Ma doświadczenie m.in. w tworzeniu architektury oraz wdrażaniu rozwiązań transakcyjnych i hurtowni danych opartych na technologiach platformy danych Microsoftu (takich jak system SQL Server i Analytics Platform System). Strate od 2009 roku jest posiadaczem tytułu Microsoft MVP (ang. *Most Valuable Professional*) z zakresu systemu SQL Server. Ponadto posiada certyfikat Microsoft Certified Master, także dotyczący tego systemu. Prowadzi blog na stronie <http://www.jasonstrate.com> i kanał @StrateSQL na Twitterze.

Daniel Taylor rozpoczął karierę techniczną w dziale kontroli jakości w firmie Nielsen Media Research Company. Potem zajął się programowaniem i pracował z systemami Sybase ASE (wersje od 11 do 12.5) i Sybase IQ, a ostatecznie wybrał system SQL Server. Taylor używał wersji systemu SQL Server od 7.0 do 2014. Pracował też dla firmy Publix jako starszy administrator baz danych i specjalista ds. technicznych. Był też liderem zespołu odpowiedzialnego za zarządzanie ponad 2000 egzemplarzy systemu SQL Server. Obecnie jest starszym specjalistą ds. SQL-owych baz danych w firmie New York Life w oddziale Tampa, gdzie należy do zespołu zarządzającego środowiskiem do analiz biznesowych i działającymi na zapleczu bazami SQL Server. Daniela znajdziesz na Twitterze (kanał @DBABullDog), na blogu <http://dbabulldog.wordpress.com> i w portalu LinkedIn (strona <http://www.linkedin.com/pub/daniel-taylor/5/B1/A96>).

W pracach nad książką udział wzięli:

REDAKTOR NACZELNY
Robert Elliott

REDAKTOR PROWADZĄCY
Kevin Shafer

REDAKTORZY TECHNICZNI
Kathi Kellenberger
Jason Strate
Daniel Taylor

KIEROWNIK PRODUKCJI
Kathleen Wisor

ADIUSTACJA
Kim Cofer

KIEROWNIK DZIAŁU ROZWOJU TREŚCI
I SKŁADU
Mary Beth Wakefield

DYREKTOR DS. MARKETINGU
W MEDIACH SPOŁECZNOŚCIOWYCH
David Mayhew

MENEDŻER DS. MARKETINGU
Carrie Sherrill

DYREKTOR HANDLOWY
Amy Knies

WICEPREZES I KIEROWNIK DZIAŁU
WYDAWNICZEGO
Richard Swadley

ASYSTENT WYDAWCY
Jim Minatel

KOORDYNATOR PROJEKTU (OKŁADKA)
Todd Klemme

KOREKTA
Sarah Kaikini, WordOne New York

INDEKSACJA
Robert Swanson

PROJEKT OKŁADKI
Wiley

Podziękowania

Dziękuję członkom wspaniałemu zespołowi z wydawnictwa Wiley, którzy pomogli nam w pracach nad tą książką, a także Bradleyowi Ballowi za jego czas i zaangażowanie w powstanie tej pozycji oraz w życie społeczności. Specjalne podziękowania należą się społeczności skupionej wokół systemu SQL Server i osobom, dzięki którym jest ona tak wspaniała — jest to książka dla was. Na specjalne wyrazy wdzięczności zasłużył też Dan Taylor; to jemu zawdzięczamy to, że udało się ukończyć tę książkę i uczciwie pracować!

— Adam Jorgensen

Chcę wyrazić wdzięczność wielu osobom. Przede wszystkim dziękuję osobom z wydawnictwa Wiley — Kevinowi Shaferowi, Mary Beth Wakefield i Robertowi Elliottowi za to, że pomagali nam skupić się na zadaniu i zgodnie współpracować. Adamowi Jorgensenowi jestem wdzięczny za to, że pozwolił mi spróbować swych sił jako autorowi. Dziękuję mojemu bratu z bractwa studenckiego Sigma Nu Timowi Picklesowi za to, że przekonał mnie do zmiany kierunku ze sztuki na informatykę. To najlepsza decyzja, jaką kiedykolwiek podjąłem na trzeźwo na uczelni! Dziękuję mojemu kumpłowi Danielowi Taylerowi, który przez prawie pięć lat udzielał mi wskazówek w kwestiach technicznych. Jestem wdzięczny współautorom tej książki. Tworzą wspaniały, utalentowany zespół i jestem dumny ze współpracy z nimi. Najważniejsze podziękowania należą się jednak moim dzieciom: Chesneyowi, Zacharemu, Williamowi i Serenity, za to, że przetrwały dni, gdy nie mogłem się z nimi bawić. Najwspanialszą rzeczą, jaką kiedykolwiek dane mi było robić, jest bycie waszym ojcem. Kocham was niezależnie od okoliczności!

— Bradley Ball

Dziękuję mojej wspaniałej żonie Tracy oraz moim córkom Eleanor i Caitlin za to, że musiały znosić mnie w czasie, gdy pisałem tę książkę. Jestem wdzięczny też wszystkim członkom zespołu SQL Program Group z Microsoftu za udzielanie odpowiedzi na moje pytania.

— Steven Wort

Chcę podziękować mojej żonie Annie i mojej córce Jennifer za wsparcie i poświęcenie w czasie, gdy pisałem tę książkę. Dziękuję osobom z centrum technologicznego Microsoftu za pomoc. To dzięki nim centrum w Chicago jest wspaniałym miejscem do uczenia się i pracy. Przede wszystkim dziękuję zespołowi pracującemu nad systemem SQL Server za udostępnienie kolejnej świetnej i pełnej nowych możliwości wersji SQL Server 2014. Na zakończenie nie mniej gorące

podziękowania składam redaktorom technicznym, redaktorowi prowadzącemu Kevinowi Shaferowi i innym osobom z wydawnictwa Wiley (w tym Robertowi Elliottowi), dzięki którym te słowa ukazały się w druku, co pozwoliło podzielić się z czytelnikami moją pasją, czyli systemem SQL Server.

— Ross LoForte

Dziękuję wszystkim, którzy przyczynili się do powstania tej książki. Jak zawsze, mam wielki dług u mojej żony Jenn za znoszenie tego, że pracowałem do późna, a także u dzieci: Coltona, Liama, Camille i Johna za cierpliwość dla zmęczonego taty, który ciągle miał za dużo do zrobienia. Dziękuję producentom Guinnessa za zapewnienie mi specjalnego napoju, który dodawał sił w trakcie pracy nad tą książką. Jestem wdzięczny także wszystkim liderom grup użytkowników, którzy ciężko pracują, aby pomóc innym zyskać biegłość w posługiwaniu się technologią. Zyskaliście mój szacunek!

— Brian Knight

Wprowadzenie

SQL Server 2014 zapewnia największy spośród wszystkich wersji tego narzędzia przeskok w skalowalności, wydajności i użyteczności dla administratorów baz danych, programistów oraz analityków biznesowych. System SQL Server 2014 pobił już dwa rekordy świata w testach TCP-H dla konfiguracji 3TB i 10TB, należące wcześniej do Oracle/SPARC. Funkcje i wydajność systemu SQL Server 2014 oferują możliwości dostępne wcześniej tylko w narzędziach PDW (ang. *Parallel Data Warehouse*) i APS (ang. *Analytic Platform System*). Teraz te możliwości są dostępne dla zwykłych użytkowników systemu SQL Server oraz dużych i małych firm. Zrozumienie poszczególnych funkcji i wiedza o tym, kiedy należy je stosować, to tylko niewielka część Twojej pracy. Po zbudowaniu systemu trzeba zadbać o jego konserwację.

Obecnie już nikogo nie dziwi używanie w systemie SQL Server baz o pojemności 40 terabajtów. Zarządzanie tym systemem było w przeszłości zadaniem dla administratorów baz danych. Jednak system SQL Server rozpowszechnił się także w mniejszych firmach, zatem wielu programistów zaczęło pełnić funkcje administratorów i analityków biznesowych. Źle skonfigurowane narzędzia mogą prowadzić do niskiej wydajności. Obecnie SQL Server zapewnia wszystkim rolom znacznie ulepszone narzędzia, lepszą integrację zabezpieczeń i bardzo duże usprawnienia w zakresie integrowania danych, zarządzania nimi, korzystania z nich i dostępności do nich.

Microsoft SQL Server 2014. Podręcznik administratora to kompletny podręcznik, za pomocą którego szybko nauczysz się, jak skonfigurować system SQL Server 2014 i jak nim zarządzać.

Dla kogo przeznaczona jest ta książka?

Niezależnie od tego, czy jesteś administratorem, czy programistą używającym systemu SQL Server, nie możesz uniknąć wykonywania od czasu do czasu zadań administracyjnych. Programiści często mają zainstalowany system SQL Server na własnym komputerze i muszą przekazywać administratorom wskazówki na temat pożądanej konfiguracji produkcyjnej wersji tego systemu. Programiści nieraz odpowiadają także za tworzenie tabel i indeksów bazy danych. Administratorzy zajmują się serwerami produkcyjnymi i często przejmują bazę danych od programisty.

Książka ta jest przeznaczona dla programistów, administratorów i zwykłych użytkowników, którzy chcą zarządzać systemem SQL Server 2014 (lub już to robią) oraz jego mechanizmami z zakresu analiz biznesowych (np. usługami SSIS — ang. *SQL Server Integration Services*). Jest to książka dla *profesjonalistów*, dlatego zakładamy, że znasz już podstawy tworzenia kwerend

w systemie SQL Server i posiadasz ogólną wiedzę na temat tego narzędzia. Nie omawiamy tu np. tego, jak tworzyć bazy danych lub jak za pomocą kreatora zainstalować system SQL Server. Zamiast tego autor rozdziału o instalacji wyjaśnia, jak wykorzystać zaawansowane ustawienia dostępne w tym procesie. Choć w tej książce nie omawiamy tworzenia kwerend do baz z systemu SQL Server, dowiesz się, jak dopracować gotowe kwerendy.

Struktura książki

W tym wydaniu książki znajdziesz większość cennej wiedzy zawartej w poprzednim jej wydaniu (*Professional SQL Server 2012 Administration*). Ponadto znajduje się tu rozdział poświęcony nowej funkcji systemu SQL Server 2014 — przetwarzaniu transakcji OLTP (ang. *Online Transaction Processing*) w pamięci. Autorzy (posiadacze tytułu SQL Server MVP, obecni i byli pracownicy Microsoftu oraz czołowi prelegenci na międzynarodowych konferencjach) wykorzystali swoją wiedzę, aby przygotować wyczerpującą aktualizację z wieloma nowymi informacjami na temat systemu SQL Server 2014.

Wymagało to omówienia licznych nowych funkcji poprawiających wydajność i skalowalność systemu SQL Server, a sprawdzone techniki dostrajania wydajności zostały uzupełnione nowymi rozwiązaniami i konfiguracjami zapewniającymi optymalną wydajność systemu SQL Server 2014. Nowa wersja tego narzędzia ma przede wszystkim przyspieszać pracę oraz oferować większą skalowalność serwera i wydajność środowiska. Dzięki temu można wykonać więcej zadań w krótszym czasie oraz z wykorzystaniem mniejszej ilości zasobów i mniejszej liczby ludzi.

Poniżej znajdziesz krótkie omówienie każdego rozdziału.

- Rozdział 1. „Architektura systemu SQL Server 2014”. Książka rozpoczyna się od przeglądu zmian w architekturze i ogólnego omówienia komponentów systemu SQL Server 2014.
- Rozdział 2. „Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014”. W tym rozdziale poznasz różne sposoby instalowania systemu SQL Server 2014 i najlepsze praktyki z tym związane.
- Rozdział 3. „Najlepsze praktyki aktualizowania systemu do wersji SQL Server 2014”. Rozdział dotyczy aktualizowania systemu do wersji SQL Server 2014, a także najlepszych praktyk, o których warto pamiętać w czasie wykonywania tej operacji. Znajdziesz tu też omówienie wymagań, korzyści płynących z aktualizacji i wyboru najlepszych metod aktualizowania systemu.
- Rozdział 4. „Zarządzanie silnikiem bazodanowym i rozwiązywanie związanych z nim problemów”. W tym rozdziale omówiono przygotowywanie i konfigurowanie silnika bazodanowego oraz radzenie sobie z występującymi problemami. Opisano tu również zarządzanie tym silnikiem i potrzebne do tego narzędzia.
- Rozdział 5. „Automatyzowanie pracy systemu SQL Server”. W tym rozdziale omówiono automatyzowanie systemu SQL Server 2014, w tym zadania (ang. *job*), powłokę PowerShell i inne metody automatyzacji.
- Rozdział 6. „Service Broker w systemie SQL Server 2014”. Service Broker to doskonałe narzędzie do obsługi komunikatów w bazie danych. W tym rozdziale znajdziesz omówienie konfigurowania tego narzędzia oraz opis stosowania go do asynchronicznego przetwarzania danych, wysyłania komunikatów i odbierania komunikatów.
- Rozdział 7. „Integrowanie systemu SQL Server ze środowiskiem CLR”. SQL Server i platforma .NET współdziałają w środowisku CLR (ang. *Common Language Runtime*).

Rozdział dotyczy integrowania platformy .NET i środowiska CLR z systemem SQL Server. Omówiono tu podzespoły i inne mechanizmy.

- Rozdział 8. „Zabezpieczanie egzemplarzy baz danych”. Bezpieczeństwo to bardzo ważny aspekt silnika bazodanowego. Ten rozdział pomoże Ci opracować i wdrożyć plan zabezpieczeń, a także zrozumieć nowe ustawienia systemu SQL Server 2014 związane z bezpieczeństwem.
- Rozdział 9. „Technologia In-Memory OLTP”. Możliwość przetwarzania transakcji OLTP w pamięci to jedna z najważniejszych zmian w architekturze systemu SQL Server od ponad dziesięciu lat. W tym rozdziale opisano nową technologię, która to umożliwia, a także dostęp do aktualnych tabel, dostosowywanie sprzętu i implementowanie nowych struktur działających w pamięci.
- Rozdział 10. „Konfigurowanie serwera pod kątem optymalnej wydajności”. Właściwe skonfigurowanie serwera jest ważne, jeśli chcesz zmaksymalizować wydajność aplikacji i bazy danych. W tym rozdziale opisano opcje przechowywania i serwera, a także inne ustawienia wpływające na wydajność systemu.
- Rozdział 11. „Optymalizowanie systemu SQL Server 2014”. Po przeczytaniu tego rozdziału będziesz umiał ocenić i przeanalizować wydajność. Omówiono tu także ustawienia i opcje konfiguracyjne związane z wydajnością systemu SQL Server.
- Rozdział 12. „Monitorowanie systemu SQL Server”. Monitorowanie pracy systemu SQL Server jest bardzo ważne, jeśli chcesz utrzymać wysoką wydajność. W tym rozdziale opisano ważne aspekty i narzędzia związane z monitorowaniem systemu SQL Server 2014.
- Rozdział 13. „Dostrajanie wydajności kodu w języku T-SQL”. Pisanie wydajnego i skutecznego kodu w języku T-SQL jest ważne, jeśli chcesz uzyskać wysoką wydajność i skalowalność aplikacji. W tym rozdziale wyjaśniono, jak zoptymalizować kod w tym języku, aby poprawić wydajność systemu. Głównym tematem jest tu wczytywanie i wykonywanie kwerend przez silnik systemu SQL Server i jego wewnętrzne mechanizmy. Dowiesz się też, jak można zoptymalizować te procesy i jak stosować najlepsze praktyki.
- Rozdział 14. „Indeksowanie baz danych”. Indeksowanie jest bardzo ważne, jeśli baza danych ma mieć wysoką wydajność. W tym rozdziale znajdziesz uwagi i strategie, które pomogą skutecznie indeksować bazy danych. Opisano tu też nowe mechanizmy indeksowania wprowadzone w wersji SQL Server 2014.
- Rozdział 15. „Replikacja”. Replikacja to ważna funkcja systemu SQL Server, która pomaga synchronizować tabele i bazy danych, a także powiązane z nimi aplikacje. W tym rozdziale opisano różne typy replikacji, ich konfigurowanie, wady i zalety.
- Rozdział 16. „Klastrowanie w systemie SQL Server 2014”. Z tego rozdziału dowiesz się, jak skonfigurować i przetestować system SQL Server 2014 z klastrami.
- Rozdział 17. „Kopie zapasowe i przywracanie stanu”. Kopie zapasowe i proces przywracania stanu są niezbędne w planie zachowania ciągłości pracy i zapewnienia zdolności operacyjnej systemu. W tym rozdziale opisano, jakie możliwości oferuje system SQL Server 2014 w tym zakresie. Znajdziesz tu też rekomendacje, które pomogą optymalnie wykorzystać oferowane możliwości, oraz poznasz nowe opcje tworzenia kopii zapasowych dostępne w systemie SQL Server 2014.
- Rozdział 18. „Przesyłanie dzienników w systemie SQL Server 2014”. Z tego rozdziału dowiesz się, jak włączyć i skonfigurować proces przesyłania dzienników oraz jak nim zarządzać.
- Rozdział 19. „Kopie lustrzane baz danych”. Proces tworzenia kopii lustrzanych został uznany w systemie SQL Server 2012 za przestarzały mechanizm, jednak w wersji SQL

Server 2014 nadal jest dostępny. W edycji Standard jest to jedyna naprawdę niezawodna funkcja, która chroni przed utratą danych w wyniku katastrofy i jednocześnie pozwala zachować wysoką dostępność danych. W tym rozdziale dokładnie opisano tworzenie kopii lustrzanych.

- Rozdział 20. „Zarządzanie usługami Integration Services i dostrajanie ich wydajności”. Integrowanie to najważniejszy proces zapewniający synchronizację systemów. W tym rozdziale opisano zarządzanie tą wartościową funkcją systemu SQL Server i jej dostrajanie.
- Rozdział 21. „Zarządzanie usługami Analysis Services i dostrajanie ich wydajności”. Usługi Analysis Services to najczęściej wybierany produkt OLAP (ang. *Online Analytical Processing*), którego administratorzy danych nie mogą nie znać. Ten rozdział pomoże przygotować się do stosowania usług.
- Rozdział 22. „Zarządzanie usługami Reporting Services w systemie SQL Server”. Za zarządzanie usługami Reporting Services często odpowiada administrator bazy danych. Ten rozdział przygotowuje do poradenia sobie ze związanymi z tymi usługami wyzwaniem, niezależnie od funkcji, jaką pełnisz.
- Rozdział 23. „Integracja systemu SQL Server 2014 z platformą SharePoint 2013”. SharePoint jest obecnie ważniejszy w systemie SQL Server niż kiedykolwiek wcześniej. Czytając ten rozdział, dowiesz się, jak zintegrować program SharePoint 2013 z systemem SQL Server. Dzięki temu będziesz mógł lepiej porozumieć się z zespołem używającym programu SharePoint 2013 lub sam zająć się administrowaniem bazami za pomocą tego narzędzia.
- Rozdział 24. „Administrowanie bazami SQL Database i ich konfigurowanie”. Rozdział zawiera wprowadzenie do platformy SQL Database PaaS (ang. *Platform as a Service*) w Azure i pomoże szybko zacząć pracę w chmurze.
- Rozdział 25. „Grupy dostępności AlwaysOn”. W tym rozdziale opisano mechanizm grup dostępności w technologii AlwaysOn. Grupy te umożliwiają kontrolowanie zbiorów egzemplarzy i serwerów, a także ustawianie priorytetów oraz uzyskiwanie wielu możliwości w zakresie obsługi awarii i zapewniania wysokiej dostępności.

Co będzie potrzebne w trakcie czytania tej książki?

Aby wykonać przykłady z tej książki, potrzebujesz zainstalowanego systemu SQL Server 2014. Jeśli chcesz nauczyć się zarządzania mechanizmami do analizy biznesowej, musisz zainstalować komponenty Analysis Services i SQL Server Integration Services (SSIS). Potrzebujesz komputera spełniającego minimalne wymagania sprzętowe systemu SQL Server 2014. Ponadto powinienś zainstalować bazy AdventureWorks2012 i AdventureWorksDW2012. Gdy w kodzie pojawia się nazwa AdventureWorks, oznacza to wersję 2012 tej bazy. Instrukcje opisujące pobieranie wspomnianych baz zawiera plik *Readme* z poświęconej tej książce strony witryn <http://www.wrox.com>.

Niektóre mechanizmy opisywane w tej książce (zwłaszcza w miejscach dotyczących wysokiej dostępności) wymagają edycji Enterprise lub Developer systemu SQL Server. Jeśli nie masz żadnej z nich, możesz wykonać niektóre przykłady za pomocą edycji Standard.

Konwencje

Aby pomóc w jak najlepszym wykorzystaniu tekstu tej książki i zrozumieniu omawianego materiału, zastosowano szereg konwencji.

OSTRZEŻENIE Ramki tego typu zawierają ważne i bezpośrednio powiązane z otaczającym je tekstem informacje, które należy zapamiętać.

UWAGA Uwagi, wskazówki, porady, sztuczki i dygresje związane z tekstem są wyróżnione w taki sposób.

W tekście stosowane są następujące style.

- Nowe pojęcia i ważne słowa są *wyróżnione* w miejscu, w którym są wprowadzane.
- Skróty klawiaturowe są wyróżnione w następujący sposób: *Ctrl+A*.
- Nazwy plików i adresy URL są wyróżnione kursywą: *persistence.properties*, a kod w tekście wygląda tak: `persistence.properties`.
- Kod jest prezentowany w dwa różne sposoby:
 - Większość przykładowego kodu jest zapisana czcionką o stałej szerokości bez wyróżnienia.
 - **Pogrubienie służy do wyróżniania fragmentów wyjątkowo ważnych w kontekście danego rozdziału.**

Errata

Dokładamy wszelkich starań, aby zapewnić, że tekst i kod są wolne od błędów. Jednak nikt nie jest doskonały, dlatego zdarzają się pomyłki. Jeśli w jednej z naszych książek znajdziesz błąd (np. literówkę w tekście lub usterkę w kodzie), będziemy wdzięczni za informacje. Jeżeli prześlesz erratę, możesz zaoszczędzić innym czytelnikom wielu godzin frustracji i jednocześnie pomóc nam w udostępnianiu informacji w jeszcze wyższej jakości.

Aby znaleźć stronę z erratą, otwórz witrynę <http://helion.pl> i znajdź stronę książki za pomocą pola wyszukiwania lub listy tytułów. Następnie otwórz zakładkę *Szczegóły* i kliknij przycisk *Zgłoś erratę*. W formularzu, który się pojawi, możesz przesłać erratę.

Architektura systemu SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Omówienie ważnych nowych funkcji systemu SQL Server 2014.
- Wyjaśnienie powiązań nowych funkcji z różnymi rolami pełnionymi przez profesjonalistów zajmujących się danymi.
- Wprowadzenie do architektury systemu SQL Server.
- Omówienie edycji systemu SQL Server i ich wpływu na pracę profesjonalistów zajmujących się danymi.

W systemie SQL Server 2012 przyjęto nowe podejście do tego, jak firmy i zatrudnieni w nich programiści, pracownicy odpowiedzialni za informacje oraz menedżerowie używają danych i je integrują. SQL Server 2014 to kontynuacja rozwoju tego systemu. Jego autorzy starali się myśleć o przyszłości i zintegrowali z systemem technologie pracy w chmurze i w pamięci. Ekscytujące nowe funkcje i usprawnienia zapewniają przełomowy wzrost wydajności, dostępności i możliwości obsługi krytycznego obciążenia lokalnie lub w chmurze. W rozdziale tym zamieszczone zostały informacje wystarczające do zrozumienia działania systemu SQL Server.

Środowisko systemu SQL Server 2014

Platforma danych systemu SQL Server w kilku ostatnich wersjach stała się zdecydowanie rozbudowana. W pierwszym podrozdziale znajduje się przegląd środowiska systemu SQL Server. Obecnie jest on rzadziej nazywany produktem, a częściej platformą danych. Liczne interakcje tego systemu z innymi produktami i funkcjami prowadzą do wyższej wydajności, skalowalności i użyteczności. Twórcy systemu SQL Server 2014 skoncentrowali się na opisanych niżej obszarach.

- **Wydajność zadań o znaczeniu strategicznym.** Nowe funkcje przetwarzania transakcji OLTP (ang. *Online Transaction Processing*) w pamięci umożliwiają uzyskanie wyższej wydajności bez modyfikowania aplikacji, tworzenie modyfikowalnych indeksów kolumnowych (ang. *columnstore*), rozszerzanie puli buforów za pomocą dysków SSD i rozszerzenia związane z technologią AlwaysOn (m.in. obsługę do ośmiu replik bazy). To wszystko sprawia, że nowa wersja systemu SQL Server oferuje większe możliwości od wszystkich jej wcześniejszych wydań.
- **Szybsze analizowanie danych.** Przy użyciu nowych, opartych na pakiecie Office narzędzi do analiz biznesowych (ang. *Business Intelligence* — BI), takich jak Power Query i Power Map, a także usprawnień w mechanizmach Power View i PowerPivot analizowanie danych nigdy nie było łatwiejsze. Ponadto rozwiązania dla dużych firm, np. tworzenie baz PDW za pomocą technologii PolyBase, umożliwiają organizacjom łatwe eksplorowanie danych typu Big Data i dochodzenie do niedostępnych wcześniej wniosków przy wykorzystaniu wszystkich możliwości zestawu narzędzi Microsoftu przeznaczonych do analiz biznesowych.
- **Platforma do pracy częściowo w chmurze.** Niezależnie od tego, czy środowisko działa w pełni lokalnie, wirtualnie, czy w całości w chmurze, system SQL Server 2014 udostępnia potrzebne narzędzia. Nowe funkcje, np. Microsoft SQL Server Backup do Windows Azure Tool, umożliwiają tworzenie kopii zapasowych w magazynie Windows Azure Blob Storage oraz szyfrowanie i kompresowanie kopii zapasowych systemu SQL Server przechowywanych lokalnie lub w chmurze. Obecnie możesz też wybrać w konfiguracji narzędzia AlwaysOn Availability Group maszynę wirtualną technologii Windows Azure jako źródło repliki.

Nowe ważne funkcje wersji 2014

Różne nowe funkcje z pewnością wzbudzą Twoją ekscytację — zależy to od tego, jaką rolę pełnisz i jak używasz systemu SQL Server. W tym podrozdziale omawiamy mechanizmy, które powinieneś wypróbować. Liczne z nich można w krótkim czasie przygotować i uruchomić, co jest atrakcyjne dla osób, które chcą jak najszybciej zacząć wpływać na pracę firmy.

Administratorzy produkcyjnych baz danych

Administratorzy produkcyjnych baz danych są dla firmy polisą ubezpieczeniową chroniącą przed zaprzestaniem pracy przez produkcyjną bazę danych. Jeśli baza danych zawiedzie, firma realizuje polisę i otrzymuje odtworzoną bazę danych. Administratorzy produkcyjnych baz danych zapewniają też optymalne działanie serwera oraz wprowadzają w bazie produkcyjnej zmiany sugerowane przez programistów i dział kontroli jakości. Oto nowe funkcje dla administratorów.

- **Transakcje OLTP przetwarzane w pamięci.** W tej wersji systemu SQL Server znajdziesz nowy, zoptymalizowany pod kątem pamięci silnik bazodanowy transakcji OLTP. Zapewnia on znaczną poprawę wydajności i skalowalności aplikacji, przy czym konieczne są tylko niewielkie zmiany w kodzie (a czasem w ogóle nie trzeba ich wprowadzać).
- **Funkcja AlwaysOn Availability Groups.** Mechanizmy z zakresu dostępności obejmują technologię Availability Groups i możliwość wyłączania bazy danych w grupach funkcji przypominających aplikacje. Choć nie jest to zupełna nowość wprowadzona w wersji 2014, dodane usprawnienia zapewniają obsługę do ośmiu baz pomocniczych (w porównaniu z czterema w wersji 2012), a także poprawę wydajności i ułatwienia w zarządzaniu.

- **Narzędzie SQL Server Backup to Windows Azure.** To bezpłatne narzędzie umożliwia tworzenie kopii zapasowych w usłudze Windows Azure Blob Storage, a także szyfruje i kompresuje kopie zapasowe przechowywane lokalnie lub w chmurze. Współdziała ono z systemem SQL Server 2005 i nowszymi wersjami.
- **Narzędzie SQL Server Backup to URL.** Początkowo pojawiło się ono w wersji SQL Server 2012 SP1 CU2, a obecnie jest w pełni zintegrowane z interfejsem narzędzia Management Studio, co pozwala tworzyć kopie zapasowe w usłudze Windows Azure Blob Storage oraz przywracać je za pomocą tej usługi.
- **SQL Server Managed Backup to Windows Azure.** Ta nowa funkcja pozwala w pełni zautomatyzować tworzenie kopii zapasowych systemu SQL Server (kompletnych i z dziennikiem transakcji) w usłudze Windows Azure Blob Storage na podstawie wieku danych i obciążenia bazy transakcjami.
- **Indeksy kolumnowe.** W wersji 2014 wprowadzono modyfikowalne dzielone na klastry indeksy kolumnowe służące do optymalizowania dużych zbiorów danych.
- **Szyfrowane kopie zapasowe.** Obecnie obsługiwane są szyfrowane kopie zapasowe. Dostępne są następujące algorytmy AES (ang. *Advanced Encryption Standard*) 128, AES 192, AES 256 i 3DES (ang. *Triple Data Encryption Standard*). Aby zaszyfrować dane w trakcie tworzenia kopii zapasowej, musisz zastosować certyfikat lub klucz asymetryczny.
- **Opóźnione utrwalanie.** Funkcja umożliwia zmniejszenie latencji za pomocą skonfigurowania wybranych lub wszystkich transakcji jako utrwalanych z opóźnieniem. W takim asynchronicznym procesie instrukcja COMMIT jest zgłaszana jako udana przed zapisaniem rekordów dziennika na dysku. Transakcje z opóźnionym utrwalaniem zostają utrwalone, gdy wpisy z dziennika transakcji są w porcjach zapisywane na dysku. Funkcja jest dostępna tylko dla transakcji OLTP przetwarzanych w pamięci.
- **Kompresja i partycje.** Usprawnione mechanizmy kompresji i korzystania z partycji obejmują możliwość odtwarzania pojedynczych partycji.
- **Resource Governor.** Funkcja ta obecnie umożliwia określenie limitów fizycznych operacji wejścia-wyjścia w ramach puli zasobów.

Administrator rozwojowych baz danych

Od czasu pojawienia się systemu SQL Server 2000 pojawił się trend odchodzenia od zatrudniania pełnoetatowych administratorów produkcyjnych baz danych. Ich zadania zostały połączone z rolą *administratorów rozwojowych baz danych*. Jednak ten trend może się zmienić z powodu regulacji, takich jak ustawa Sarbanesa-Oxleya, która wymaga opracowywania zmian i ich wprowadzania przez odrębne osoby. W dużych organizacjach administrator produkcyjnych baz danych może pracować w dziale operacyjnym, który obejmuje wielu administratorów i pracowników pomocy technicznej dla systemu Windows. W innych firmach administrator produkcyjnych baz danych może pracować w dziale programistycznym. W takiej sytuacji nie następuje podział obowiązków, który czasem jest niezbędny ze względów prawnych.

Administratorzy rozwojowych baz danych odgrywają w firmie tradycyjną rolę. Są bardziej programistami oraz pełnią funkcję ekspertów baz danych w gronie programistów. Te osoby dbają, aby wszystkie procedury składowane były optymalnie napisane oraz by model bazy danych był poprawny fizycznie i logicznie. Administratorzy rozwojowych baz danych mogą też pisać oprogramowanie do migracji, które aktualizuje bazę danych ze starszej wersji do nowej. Osoby pełniące tę rolę zwykle nie są budzone telefonami o 2.00 w nocy, co zdarza się

administratorom produkcyjnych baz danych, gdy nie powiedzie się tworzenie kopii zapasowej lub wystąpią podobne problemy. Poniżej opisano wybrane narzędzia z nowej wersji systemu SQL Server, które mogą zainteresować administratorów rozwojowych baz danych.

- Usprawnienia w języku T-SQL (ang. *Transact SQL*) — przykładowo wewnętrzne specyfikacje indeksów dla tabel przechowywanych na dysku. Obecnie instrukcja `SELECT...INTO` może działać równolegle w bazach danych o trybie zgodności (ustawienie `COMPATIBILITY_LEVEL`) ustawionym na 110 lub wyższym.
- Aktualizacja narzędzia SSDT-BI (ang. *SQL Server Data Tools for Business Intelligence*), w tym dodanie obsługi tworzenia raportów typu Power View z wykorzystaniem modeli wielowymiarowych. Inne usprawnienia tego narzędzia obejmują możliwość tworzenia projektów dla starszych wersji (od 2005) usług Analysis Services i Reporting Services. Zgodność wstecz dla usług Integration Services nie została jeszcze zapewniona.

Administratorzy rozwojowych baz danych zwykle odpowiadają przed zespołem programistycznym i otrzymują zapytania od analityków biznesowych lub programistów. Przy tradycyjnym podziale ról administratorzy rozwojowych baz danych nigdy nie powinni mieć możliwości modyfikowania baz produkcyjnych. Jednak muszą mieć dostęp do nich w trybie tylko do odczytu, co umożliwia debugowanie bazy po wystąpieniu problemów.

Administratorzy i programiści odpowiedzialni za analizy biznesowe

Administrator BI to rola, która powstała w wyniku zwiększenia możliwości systemu SQL Server. W wersji SQL Server 2012 funkcje BI zostały rozbudowane i stały się niezwykle ważną grupą narzędzi, bez których wiele firm nie wyobraża sobie pracy. Administrator (lub programista) BI to ekspert w zakresie tych narzędzi. W wersji SQL Server 2014 zwiększono komfort pracy użytkownika końcowego BI (proces ten zapoczątkowano w wersji SQL Server 2012) dzięki nowym połączeniom rozwiązania Power BI z pakietem Office365, a także gotowym funkcjom BI.

Administrator — programista BI specjalizuje się w najlepszych praktykach związanych z BI oraz optymalizowaniu i stosowaniu narzędzi BI. W małych firmach taka osoba może tworzyć pakiety oparte na usługach SSIS (ang. *SQL Server Integration Services*), wykonujące procesy ETL (ang. *extract, transform, load*, czyli pobieranie, przekształcanie i wczytywanie) lub generujące raporty dla użytkowników. W dużych organizacjach programiści tworzą pakiety oparte na usługach SSIS i raporty oparte na usługach SSRS (ang. *SQL Server Reporting Services*). Administratorzy — programiści BI są często pytani o opinię w kwestii fizycznej implementacji pakietów SSIS, kostek SSAS (ang. *SQL Server Analysis Services*) lub modeli tabelarycznych SSAS.

Administratorzy — programiści BI mogą odpowiadać za takie zadania jak:

- budowanie kostek, modeli tabel i rozwiązań opartych na usługach Analysis Services oraz udzielanie konsultacji z tego zakresu,
- tworzenie raportów za pomocą usług Reporting Services,
- tworzenie procesów ETL za pomocą usług SSIS i udzielanie konsultacji z tego zakresu,
- tworzenie pakietów instalacyjnych przekazywanych administratorom produkcyjnych baz danych.

Wymienione poniżej nowe funkcje sprawiają, że nadeszły ciekawe czasy dla osób zajmujących się BI. Oto one.

- Szybkie odkrywanie danych za pomocą narzędzi Power View i PowerPivot.
- Błyskawiczne generowanie wyników kwerend przy użyciu indeksów kolumnowych.
- Możliwość samodzielnego, łatwego odkrywania, używania i łączenia różnych źródeł danych przez użytkowników z wykorzystaniem narzędzia Power Query.
- Możliwość tworzenia rozbudowanych wizualizacji przestrzennych z zastosowaniem narzędzia Power Map.
- Zarządzana samodzielna BI za pomocą SharePointa i modelu BISM (ang. *BI Semantic Model*).
- Wiarygodne i spójne dane otrzymywane z wykorzystaniem usług Data Quality Services i technologii Master Data Management.
- Niezawodne rozwiązania oparte na hurtowniach danych tworzone po zastosowaniu technologii Parallel Data Warehouse i Reference Architecture.

Architektura systemu SQL Server

W systemie SQL Server 2014 wprowadzono wiele usprawnień platformy danych. Nie tylko zapewniają one przełomową poprawę wydajności i skalowalności, ale też umożliwiają samodzielną analizę biznesową (BI), co pozwala użytkownikom końcowym eksplorować i opracowywać dane szybciej i łatwiej niż kiedykolwiek wcześniej. Nowe mechanizmy, np. przetwarzanie transakcji OLTP w pamięci, prowadzą do znacznej poprawy wydajności aplikacji przy niewielkich (lub żadnych) zmianach w schemacie. Inne funkcje, przykładowo AlwaysOn Availability Groups, umożliwiają szybkie i łatwe skalowanie aplikacji bazodanowych.

Aby wiedzieć, jak wykorzystać te ekscytujące funkcje i mechanizmy, trzeba zrozumieć architekturę systemu SQL Server. W tym podrozdziale opisano podstawowe typy plików systemu SQL Server 2014, zarządzanie tymi plikami, narzędzie SQL Client i bazy danych. Znajdziesz tu także omówienie schematów, synonimów i obiektów DMO (ang. *Dynamic Management Objects*). Ponadto poznasz dostępne typy danych, edycje i licencje.

Pliki bazy danych i dziennik transakcji

Architektura bazy danych i dziennika transakcji nie zmieniła się znacznie w porównaniu z poprzednimi wersjami systemu SQL Server. Pliki bazy danych — w zależności od ich typu — pełnią dwa podstawowe zadania. *Pliki danych* przechowują dane, indeksy i inne struktury pomocnicze. *Pliki dziennika* zawierają dane zatwierdzonych transakcji i umożliwiają zapewnienie spójności bazy.

Pliki bazy danych

Baza danych może zawierać wiele *grup plików*. Każda taka grupa musi obejmować przynajmniej jeden fizyczny plik danych. Grupy plików ułatwiają administrowanie zbiorami plików. Pliki danych są podzielone na strony o wielkości 8 kilobajtów tworzące ekstenty o wielkości 64 kilobajtów. Za pomocą opcji w instrukcji `create/alter index` języka T-SQL można określić, w jaki stopniu wypełnione mają być strony danych. W systemie SQL Server 2014 nadal można częściowo udostępniać bazę danych, jeśli uszkodzony jest jeden plik. W takiej sytuacji administrator bazy danych może udostępnić pozostałe pliki do odczytu i zapisu, a użytkownicy zobaczą błąd, gdy spróbują skorzystać z niedostępnych części bazy.

W wersji SQL Server 2000 i starszych wydaniach wiersze mogły mieć do 8060 bajtów. Wyjątkami od tej reguły były kolumny typów `text`, `ntext`, `image`, `varchar(max)`, `varbinary(max)` i `nvarchar(max)`, które mogły zawierać do 2 gigabajtów danych i były zarządzane niezależnie. Od wersji SQL Server 2005 limit 8 kilobajtów dotyczy tylko kolumn o stałej długości. Suma wielkości takich kolumn i wskaźników do kolumn innych typów nie może przekraczać 8060 bajtów na wiersz. Jednak każda kolumna o zmiennej długości może obejmować do 8 kilobajtów, dlatego łączna wielkość wiersza może znacznie przekraczać 8060 bajtów. Gdy tak się stanie, może nastąpić spadek wydajności, ponieważ logiczny wiersz trzeba dzielić na kilka fizycznych wierszy o długości 8060 bajtów każdy.

Dziennik transakcji

Dziennik transakcji ma gwarantować, że wszystkie zatwierdzone transakcje zostaną utrwalone w bazie danych i będzie można je odtworzyć (w ramach przywracania lub odtwarzania punktu w czasie). Dziennik transakcji jest *zapisywany z góry*. Gdy użytkownik wprowadza zmiany w bazie z systemu SQL Server, dane są zapisywane w dzienniku, a dopiero potem strony, które trzeba zmodyfikować, zostają wczytane do pamięci. Strony te są następnie traktowane jak brudnopis i zapisywane są w nich zmiany. Po dojściu do punktu kontrolnego strony z brudnopisu zostają zapisane na dysku. Od tego momentu dostępne są trwałe strony, których nie trzeba już uwzględniać w buforze zapisu. To dlatego dziennik transakcji czasem znacznie rozrasta się w trakcie długich transakcji, nawet przy ustawionym prostym modelu przywracania stanu. Więcej informacji na ten temat podano w rozdziale 17., „Kopie zapasowe i przywracanie stanu”.

SQL Server Native Client

SQL Server Native Client (SNAC) to technika dostępu do danych, która pojawiła się w systemie SQL Server 2005 i została usprawniona w wersji 2012. Jest ona używana zarówno w bazach OLE (ang. *Object Linking and Embedding*), jak i w technologii ODBC (ang. *Open Database Connectivity*) przy dostępie do systemu SQL Server. SNAC upraszcza dostęp do systemu SQL Server, ponieważ łączy biblioteki technologii OLE i ODBC w jednej technice. Ten sposób dostępu pozwala korzystać z funkcji systemu SQL Server, takich jak:

- tworzenie kopii lustrzanych bazy,
- routing bazy pomocniczej do odczytu w technologii AlwaysOn,
- wyniki MARS (ang. *Multiple Active Result Sets*),
- obsługa natywnego klienta baz LocalDB,
- obsługa strumieni FILESTREAM,
- izolowanie snapshotów,
- powiadamianie o kwerendach,
- obsługa XML-owych typów danych,
- typy UDT (ang. *User-Defined Data Types*),
- szyfrowanie,
- wykonywanie operacji asynchronicznych,
- używanie dużych typów danych,
- kopiowanie masowe,
- parametry w postaci wartości z tabeli,

- zdefiniowane przez użytkownika duże typy środowiska CLR (ang. *Common Language Runtime*),
- wygasanie haseł,
- obsługa nazw SPN (ang. *Service Principal Name*) w połączeniach klienckich.

Za pomocą tych funkcji możesz korzystać z mechanizmów innych warstw danych (np. komponentów MDAC — ang. *Microsoft Data Access Components*), choć wymaga to więcej pracy. Komponenty MDAC nadal są dostępne i możesz ich używać, jeśli nie potrzebujesz nowych funkcji z systemu SQL Server 2008 i 2012. Jeżeli tworzysz aplikacje oparte na modelu COM, powinieneś używać klienta SNAC. Ponadto jeśli tworzysz aplikacje z kodem zarządzanym (np. w języku C#), pomyśl o zastosowaniu dostawcy .NET Framework Data Provider for SQL Server. Jest to solidne rozwiązanie z obsługą funkcji systemów SQL Server 2008 i 2012.

System SQL Server 2014 instaluje klienta SQL Server 2012 Native Client, ponieważ nie istnieje wersja SQL Server 2014 Native Client. Inne ważne zmiany to rezygnacja ze sterownika ODBC w kliencie SQL Server Native Client. Z tego powodu w tym kliencie sterownik ten nie będzie już aktualizowany. Następcą sterownika ODBC w kliencie SQL Server Native Client jest sterownik Microsoft ODBC Driver 11 for SQL Server on Windows. System SQL Server 2014 domyślnie instaluje właśnie ten sterownik.

Standardowe systemowe bazy danych

Systemowe bazy danych w SQL Server są niezbędne do działania tego systemu i zwykle nie należy ich modyfikować. Jedyny wyjątek można zrobić w przypadku baz model (baza ta umożliwia wprowadzanie zmian, np. dodawanie procedur składowanych do nowo utworzonej bazy) i tempdb (prawdopodobnie będziesz musiał ją zmodyfikować, aby zapewnić lepsze rozkładanie obciążenia). W kolejnych podpunktach znajdziesz szczegółowy opis standardowych systemowych baz danych.

OSTRZEŻENIE Zmodyfikowanie lub uszkodzenie niektórych systemowych baz danych czasem uniemożliwia uruchomienie systemu SQL Server. Baza master zawiera wszystkie procedury składowane i tabele potrzebne do działania systemu SQL Server.

Baza danych resource

W systemie SQL Server 2005 wprowadzono bazę resource. Zawiera ona tabele, metadane i procedury składowane krytyczne dla działania systemu SQL Server. Są one dostępne w trybie tylko do odczytu. Baza nie zawiera żadnych informacji na temat danego egzemplarza systemu ani przechowywanych w nim baz danych, ponieważ jest zapisywana tylko w trakcie instalowania nowego pakietu poprawek. Baza resource obejmuje wszystkie fizyczne tabele i procedury składowane, które występują jako jednostki logiczne w pozostałych bazach. W domyślnej instalacji baza ta znajduje się w katalogu `C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Binn`. Jeden egzemplarz systemu zawiera tylko jedną bazę resource.

UWAGA Dysk C: w ścieżce jest używany w standardowej konfiguracji. Jeśli Twój komputer jest skonfigurowany w inny sposób, dostosuj ścieżkę do użytych ustawień. Ponadto `.MSSQLSERVER` to nazwa egzemplarza systemu. Jeśli podałeś inną nazwę, wpisz ją w ścieżce.

W wersji SQL Server 2000 przy instalowaniu nowego pakietu poprawek trzeba było uruchamiać wiele długich skryptów, aby usuwały i ponownie tworzyły obiekty systemowe. Proces zajmował wiele czasu i sprawiał, że po zainstalowaniu pakietu poprawek nie można było przywrócić wcześniejszej wersji systemu. Od wersji SQL Server 2012 zastosowanie nowego pakietu poprawek lub aktualizacji prowadzi do zastąpienia dawnej bazy resource nową. Dzięki temu można zarówno szybko zaktualizować katalog systemu SQL Server, jak i przywrócić starszą wersję.

Baza danych resource nie jest widoczna w Management Studio i nigdy nie należy jej modyfikować, chyba że zaleci to obsługa techniczna Microsoftu. Możesz połączyć się z tą bazą w pewnych warunkach w trybie jednodostępnym za pomocą polecenia **USE MSSQLSystemResource**. Zwykle administrator bazy danych uruchamia proste kwerendy na omawianej bazie, gdy jest połączony z dowolną inną bazą, zamiast łączyć się bezpośrednio z bazą resource. Microsoft udostępnia kilka funkcji, które to umożliwiają. Jeśli przykładowo uruchomisz poniższą kwerendę, gdy jesteś połączony z dowolną bazą, poznasz wersję bazy resource i czas jej ostatniej aktualizacji:

```
SELECT serverproperty('resourceversion') ResourceDBVersion,  
serverproperty('resourcelastupdatedatetime') LastUpdateDate;
```

UWAGA Nie umieszczaj bazy resource na zaszyfrowanym lub skompresowanym dysku, ponieważ grozi to problemami z aktualizacjami lub wydajnością.

Baza danych master

Baza master zawiera metadane dotyczące używanych baz danych (konfiguracje baz i lokalizacje ich plików), loginy i dane o konfiguracji danego egzemplarza systemu. Poniższa kwerenda pozwala wyświetlić niektóre metadane przechowywane w bazie master. Kwerenda zwraca informacje na temat baz dostępnych na serwerze:

```
SELECT * FROM sys.databases;
```

Główna różnica między bazami resource i master polega na tym, że baza master zawiera dane specyficzne dla danego egzemplarza systemu, a baza resource obejmuje tylko schemat i procedury składowane potrzebne do uruchomienia tego egzemplarza (nie ma w niej żadnych danych specyficznych dla danego egzemplarza).

OSTRZEŻENIE Rzadko trzeba tworzyć obiekty w bazie master. Jeśli opracowujesz w niej obiekty, powinieneś często tworzyć jej kopie zapasowe.

Baza danych tempdb

Baza tempdb działa podobnie jak plik stronicowania w systemie operacyjnym. Jest używana do przechowywania tymczasowych obiektów utworzonych przez użytkowników, tymczasowych obiektów potrzebnych silnikowi bazodanowemu, a także nieprzetworzonych informacji. Baza tempdb powstaje za każdym razem, gdy uruchomisz system SQL Server. Jest wtedy odtwarzana w swym pierwotnym rozmiarze. Ponieważ jest generowana przy każdym uruchomieniu systemu, nie można utworzyć jej kopii zapasowej. Korzyścią modyfikowania danych w bazie tempdb jest ograniczenie zapisu zmian w dzienniku. Bazie tempdb trzeba przydzielić odpowiednią ilość pamięci, ponieważ wiele operacji wykonywanych przez aplikacje bazodanowe korzysta z tej bazy. Zwykle należy ustawić tę bazę tak, aby automatycznie zwiększała wielkość, gdy potrzebna będzie dodatkowa pamięć. Przeważnie rozmiar bazy tempdb jest zmienny, jednak trzeba określić

szczytowe wymagania pamięciowe aplikacji i zapewnić odpowiednią ilość pamięci z nadmiarowymi 15 – 20% na ewentualne rozrastanie się bazy. Jeśli zabraknie pamięci, mogą wystąpić błędy. Oto one.

- **1101** lub **1105** — sesja łącząca się z systemem SQL Server musi przydzielić pamięć w bazie tempdb.
- **3959** — pamięć wersji jest pełna.
- **3967** — trzeba zmniejszyć pamięć wersji, ponieważ baza tempdb jest pełna.

Baza danych model

Ta systemowa baza danych jest szablonem używanym przez system SQL Server przy tworzeniu nowych baz. Gdy budowana jest nowa baza, system SQL Server kopiuje bazę model. Jedyny wyjątek to sytuacja, w której użytkownik odtwarza lub dołącza bazę z innego serwera.

Jeśli pewna tabela, procedura składowana lub opcja bazy powinny znajdować się w każdej nowej bazie tworzonej na serwerze, możesz uprościć sobie pracę i opracować potrzebny obiekt w bazie model. W momencie tworzenia nowej bazy zostanie wtedy skopiowana baza model z dodanymi do niej specjalnymi obiektami lub ustawieniami. Jeżeli do bazy model dodałeś własne obiekty, powinieneś uwzględnić je w kopiach zapasowych lub przygotować skrypt, który wprowadza potrzebne zmiany.

Baza danych msdb

Ta baza zawiera informacje używane przez narzędzia SQL Server Agent i SSIS, a także przy przesyłaniu dzienników oraz w systemie tworzenia i przywracania kopii zapasowych silnika relacyjnej bazy danych. Baza msdb zawiera wszystkie informacje na temat zadań, operatorów, alertów, polityk i historii wykonywania zadań. Ponieważ obejmuje ona ważne dane systemowe, należy regularnie tworzyć jej kopię zapasową.

Schematy

Schematy umożliwiają grupowanie obiektów bazy danych. Ułatwia to administrowanie bazą, ponieważ można zastosować zabezpieczenia do wszystkich obiektów ze schematu. Schematy służą też do porządkowania obiektów, dzięki czemu użytkownicy mogą łatwo znaleźć potrzebne jednostki. Możesz przykładowo utworzyć schemat HumanResources i umieścić w nim wszystkie tabele z danymi pracowników oraz procedury składowane, a następnie zastosować do schematu polityki bezpieczeństwa w celu określenia dostępu do zawartych w nim obiektów.

Gdy wskazujesz obiekt, zawsze powinieneś używać dwuczłonowej nazwy. Domyślnym schematem bazy danych jest dbo. Aby wskazać tabelę Employee z tego schematu, należy podać nazwę dbo.Employee. Nazwy tabel muszą być unikatowe w ramach schematu. W schemacie HumanResources możesz utworzyć inną tabelę o nazwie Employee. Wskazać ją można za pomocą nazwy HumanResources.Employee. Taka tabela znajduje się w przykładowej bazie AdventureWorks z systemu SQL Server 2014. Wszystkie omawiane przykładowe bazy systemu SQL Server 2014 trzeba pobrać pod adresem <ftp://ftp.helion.pl/przyklady/sq14pa.zip> i zainstalować. Oto przykładowa kwerenda z dwuczłonową nazwą:

```
USE AdventureWorks2014
GO
SELECT BusinessEntityID, JobTitle
FROM HumanResources.Employee;
```

W wersjach starszych niż SQL Server 2005 pierwszą częścią dwuczłonowej nazwy była nazwa właściciela obiektu. Jednak to rozwiązanie prowadziło do problemów z konserwacją. Gdy właściciel obiektu odchodził z firmy, przed usunięciem jego loginu z systemu SQL Server trzeba było się upewnić, że wszystkie należące do tej osoby obiekty przypisano do innych właścicieli. Cały kod korzystający z tych obiektów trzeba było zmodyfikować, tak aby występowali w nim nowi właściciele. Dzięki oddzieleniu własności obiektów od nazwy schematu w wersji SQL Server 2005 i nowszych edycjach problem ten został rozwiązany.

Synonimy

Synonym to alias (nazwa zastępcza) obiektu. Zapewnia warstwę abstrakcyjną między obiektem bazy danych a użytkownikami. Warstwa ta pozwala zmienić wybrane aspekty fizycznej implementacji i odizolować zmiany od użytkownika. Poniższy przykład związany jest ze stosowaniem serwerów połączonych (ang. *linked server*). Na innym serwerze mogą znajdować się tabele, które trzeba złączyć z tabelami na serwerze lokalnym. Aby wskazać obiekty z tego innego serwera, należy zastosować czteroczłonową nazwę:

```
SELECT Column1,Column2
FROM LinkedServerName.DatabaseName.SchemaName.TableName;
```

Możesz utworzyć dla nazwy `LinkedServerName.DatabaseName.SchemaName.TableName` synonim `SchemaName.SynonymName`. Użytkownicy danych powinni wtedy wskazywać obiekt w następujący sposób:

```
SELECT Column1,Column2
FROM SchemaName.SynonymName;
```

Taka warstwa abstrakcyjna umożliwia przeniesienie tabeli na inny serwer, użycie nazwy innego serwera połączonego, a nawet zreplicowanie danych i umieszczenie ich na serwerze lokalnym w celu poprawy wydajności, przy czym nie wymaga to żadnych zmian w kodzie korzystającym z danej tabeli.

UWAGA W synonimie nie można używać innego synonimu. Funkcja `object_id` zwraca id synonimu, a nie powiązanego obiektu podstawowego. Jeśli potrzebujesz warstwy abstrakcyjnej na poziomie kolumn, zastosuj widoki.

Obiekty DMO

Obiekty DMO (ang. *Dynamic Management Object*) i funkcje zwracają informacje na temat danego egzemplarza systemu SQL Server i systemu operacyjnego. Są dwa rodzaje obiektów DMO: DMV (ang. *Dynamic Management View*) i DMF (ang. *Dynamic Management Function*). Obiekty DMV i DMF upraszczają wyświetlanie danych oraz nowych informacji niedostępnych w wersjach systemu SQL Server starszych niż 2005. Obiekty DMO udostępniają dane różnego rodzaju (m.in. o podsystemie wejścia-wyjścia, pamięci RAM lub narzędziu Service Broker).

Gdy uruchamiasz egzemplarz systemu SQL Server, zapisuje on w pamięci stan serwera i informacje diagnostyczne, dostępne za pomocą obiektów DMV i DMF. Po zatrzymaniu i ponownym uruchomieniu egzemplarza systemu informacje są usuwane z widoków i rozpoczyna się zapisywanie nowych danych. Dla tych widoków można tworzyć kwerendy w taki sam sposób jak dla innych tabel systemu SQL Server, czyli przy użyciu dwuczłonowych nazw. Przykładowo poniższa kwerenda używa obiektu DMV `sys.dm_exec_sessions` do pobrania liczby sesji podłączonych do danego egzemplarza i pogrupowania ich według loginów:

```
SELECT login_name, COUNT(session_id) as NumberSessions
FROM sys.dm_exec_sessions GROUP BY login_name;
```

Niektóre obiekty DMF to funkcje przyjmujące parametry. I tak w poniższym kodzie użyto obiektu DMF `sys.dm_io_virtual_file_stats` do pobrania statystyk wejścia-wyjścia dotyczących pliku danych AdventureWorks:

```
USE AdventureWorks
GO
SELECT * FROM
sys.dm_io_virtual_file_stats(DB_ID('AdventureWorks2014'),
FILE_ID('AdventureWorks_Data'));
```

W systemie SQL Server 2012 pojawiło się wiele nowych obiektów DMV i DMF. Zapewniają one lepszy wgląd w nowe i istniejące już wcześniej mechanizmy, takie jak:

- AlwaysOn Availability Groups,
- związane z rejestrowaniem zmian w danych,
- związane ze śledzeniem zmian,
- związane z środowiskiem CLR,
- związane z tworzeniem kopii lustrzanych baz danych,
- związane z bazami danych,
- związane z wykonywaniem operacji,
- rozszerzone zdarzenia systemu SQL Server,
- FileStream i FileTable,
- wyszukiwanie pełnotekstowe i semantyczne,
- związane z indeksami,
- związane z operacjami wejścia-wyjścia,
- związane z obiektami,
- związane z powiadomieniami o kwerendach,
- związane z replikacją,
- związane z funkcją Resource Governor,
- związane z funkcją Service Broker,
- związane z danymi przestrzennymi,
- związane z systemem SQLSOS,
- związane z transakcjami.

W systemie SQL Server 2014 wprowadzono nowe obiekty DMV i DMF związane z:

- technologią CSV (ang. *Cluster Shared Volume*),
- rozszerzaniem pul buforów,
- transakcjami OLTP przetwarzanymi w pamięci.

Typy danych w systemie SQL Server 2014

Typy danych są podstawowym elementem w trakcie tworzenia tabel w systemie SQL Server. W czasie budowania tabeli musisz określić typ każdej kolumny. W tym punkcie opisano popularne typy danych z systemu SQL Server. Jeśli nawet tworzysz niestandardowy typ danych,

musi on być oparty na standardowym typie z tego systemu. Możesz np. dodać niestandardowy typ danych Address za pomocą pokazanej poniżej składni, zauważ jednak, że jest on oparty na standardowym typie danych varchar systemu SQL Server:

```
CREATE TYPE Address
FROM varchar(35)NOT NULL;
```

Jeśli zmieniasz typ danych kolumny z dużej tabeli w interfejsie do projektowania tabel w narzędziu SQL Server Management Studio, operacja ta może zająć dużo czasu. Aby poznać przyczyny takiej sytuacji, należy podejrzewać skrypt wprowadzający zmiany w tym interfejsie. Narzędzie Management Studio tworzy tymczasową tabelę pomocniczą o nazwie w formacie tmpTableName, a następnie kopiuje dane do tej tabeli. Następnie usuwa pierwotną tabelę i zmienia nazwę nowej tabeli po zastosowaniu w niej nowego typu danych. Inne operacje odpowiadają za obsługę indeksów i relacji tabeli.

W tabeli zawierającej milion rekordów proces może zająć ponad 10 minut, a czasem nawet ponad godzinę. Aby tego uniknąć, możesz zmienić typ kolumny za pomocą prostej, jednowierszowej instrukcji w języku T-SQL wpisanej w oknie kwerend. Aby przykładowo zmienić typ danych kolumny JobTitle z tabeli Employees na varchar(70), zastosuj następującą składnię:

```
ALTER TABLE HumanResources.Employee ALTER COLUMN JobTitle Varchar(70);
```

OSTRZEŻENIE W trakcie konwersji danych na niezgodny typ może nastąpić utrata ważnych informacji. Przykładowo konwersja z typu numerycznego z wartością 15,415 na liczbę całkowitą spowoduje zaokrąglenie tej wartości do liczby całkowitej.

Być może będziesz chciał napisać raport wyświetlający typy danych każdej kolumny wybranej tabeli z systemu SQL Server. To zadanie ma wiele rozwiązań. W poniższym przykładzie zastosowano popularną metodę, która łączy tabelę sys.objects z tabelą sys.columns. Możliwe, że nie znasz dwóch funkcji zastosowanych w przykładowym kodzie. Funkcja TYPE_NAME() przekształca identyfikator typu danych na nazwę tego typu (aby wykonać odwrotną operację, zastosuj funkcję TYPE_ID()). Druga warta uwagi funkcja to SCHEMA_ID(). Zwraca ona identyfikator schematu, przydatny głównie przy pisaniu raportów wykorzystujących metadane systemu SQL Server.

```
USE AdventureWorks2014;
GO
SELECT o.name AS ObjectName,
       c.name AS ColumnName,
       TYPE_NAME(c.user_type_id) as DataType
FROM sys.objects o
JOIN sys.columns c
ON o.object_id = c.object_id
WHERE o.name = 'Department'
and o.Schema_ID = SCHEMA_ID('HumanResources');
```

Kod ten zwraca następujące wyniki (*Name* reprezentuje typ danych zdefiniowany przez użytkownika):

ObjectName	ColumnName	DataType
Department	DepartmentID	smallint
Department	Name	Name
Department	GroupName	Name
Department	ModifiedDate	datetime

Znakowe typy danych

Znakowe typy danych to varchar, char i text. Typy danych z tej grupy przechowują dane znakowe. Główna różnica między typami varchar i char polega na dopełnianiu danych. Jeśli w kolumnie o nazwie FirstName typu varchar(20) zapiszesz wartość „Jacek”, fizycznie dane zajmą tylko 5 bajtów (plus miejsce na dane pomocnicze). Gdy zapiszesz tę samą wartość w kolumnie typu char(20), wykorzystanych zostanie 20 bajtów. SQL Server wstawi wtedy na końcu spacje, aby uzyskać 20 znaków.

UWAGA Może się zastanawiasz, po co stosować typ danych char, skoro zajmuje więcej pamięci? Z używaniem typu varchar związane są pewne koszty. Jeśli na przykład przechowujesz dwuliterowe skróty nazw stanów, lepiej zastosować kolumnę typu char(2). Choć niektórzy administratorzy baz danych mają bardzo sztywne przekonania na ten temat, ogólnie uważa się, że warto ustalić w firmie próg wielkości danych, poniżej którego używany jest typ char zamiast varchar. Zwykle dla kolumn o długości mniejszej lub równej 5 bajtów powinno się stosować typ char zamiast varchar, choć zależy to od wymagań związanych z daną aplikacją. Powyżej tego poziomu zalety stosowania typu varchar zaczynają przeważać nad kosztami.

Typy nvarchar i nchar działają tak samo jak ich odpowiedniki varchar i char, ale obsługują międzynarodowe znaki Unicode. Związane jest to jednak z kosztami. Każdy znak Unicode zajmuje 2 bajty. Jeśli chcesz zapisać wartość „Jacek” w kolumnie typu nvarchar, zajmie ona 10 bajtów, a użycie typu nchar(20) będzie oznaczać zajęcie 40 bajtów. Z powodu tych kosztów nie należy stosować kolumn ze znakami Unicode, chyba że wynika to z potrzeb biznesowych lub używanego języka. Zastanów się nad przyszłością i pomyśl o sytuacjach, w których możesz potrzebować takich typów danych. Jeśli nie będą potrzebne, unikaj ich.

W tabeli 1.1 przedstawiono znakowe typy danych razem z krótkimi opisami i podaną potrzebną ilością pamięci.

Tabela 1.1. Typy danych z systemu SQL Server

Typ danych	Opis	Potrzebna pamięć
Char(n)	n oznacza liczbę znaków i przyjmuje wartości od 1 do 8000.	n bajtów
Nchar(n)	n oznacza liczbę znaków Unicode i przyjmuje wartości od 1 do 8000.	$2 \times n$ bajtów
Nvarchar(max)	Do $2^{30} - 1$ (1 073 741 823) znaków Unicode.	$2 \times$ zapisane znaki + 2 bajty kosztów
Text	Do $2^{31} - 1$ (2 147 483 647) znaków.	1 bajt na zapisany znak + 2 bajty kosztów
Varchar(n)	n oznacza liczbę znaków i przyjmuje wartości od 1 do 8000.	1 bajt na zapisany znak + 2 bajty kosztów
Varchar(max)	Do $2^{31} - 1$ (2 147 483 647) znaków.	1 bajt na zapisany znak + 2 bajty kosztów

Dokładne liczbowe typy danych

Liczbowe typy danych to: `bit`, `tinyint`, `smallint`, `int`, `bigint`, `numeric`, `decimal`, `smallmoney` i `money`. Każdy z tych typów danych przechowuje wartości liczbowe innego rodzaju. Pierwszy typ danych, `bit`, przechowuje tylko wartości `null`, 0 i 1, które w większości aplikacji oznaczają fałsz i prawdę. Typ danych `bit` doskonale nadaje się dla opcji włączona lub wyłączona i zajmuje tylko 1 bajt pamięci. W tabeli 1.2 opisano ten i inne popularne liczbowe typy danych.

Tabela 1.2. Dokładne liczbowe typy danych

Typ danych	Opis	Potrzebna pamięć
Bit	0, 1 lub null	1 bajt na każde 8 kolumn tego typu danych
tinyint	Liczby całkowite od 0 do 255	1 bajt
smallint	Liczby całkowite od -32 768 do 32 767	2 bajty
Int	Liczby całkowite od -2 147 483 648 do 2 147 483 647	4 bajty
bigint	Liczby całkowite od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807	8 bajtów
numeric(p, s) lub decimal(p, s)	Liczby od $-10^{38} + 1$ do $10^{38} - 1$	do 17 bajtów
money	Od -922 337 203 685 477,5808 do 922 337 203 685 477,5807	8 bajtów
smallmoney	Od -214 748,3648 do 214 748,3647	4 bajty

Liczbowe typy danych, takie jak `decimal` i `numeric`, przechowują zmienną liczbę cyfr na lewo i na prawo od przecinka. *Skala* (parametr `s`) określa liczbę cyfr po przecinku, a precyzja (parametr `p`) wyznacza łączną liczbę cyfr (razem z cyframi znajdującymi się po przecinku). Przykładowo wartość 14,88531 może mieć typ `numeric(7,5)` lub `decimal(7,5)`. Jeśli wstawisz wartość 14,25 do kolumny typu `numeric(5,1)`, wartość ta zostanie zaokrąglona do liczby 14,3.

Przybliżone liczbowe typy danych

Grupa obejmuje typy `float` i `real`. Należy je stosować do reprezentowania liczb zmiennoprzecinkowych. Ponieważ jednak te typy przechowują wartości przybliżone, nie pozwalają na precyzyjne reprezentowanie wszystkich liczb.

Parametr `n` w typie `float(n)` to liczba bitów używanych do przechowywania mantysy liczby. SQL Server stosuje tylko dwie wartości tego parametru. Jeśli podasz wartość z przedziału od 1 do 24, SQL Server użyje liczby 24. Jeżeli ustawisz wartość z przedziału od 25 do 53, SQL Server zastosuje liczbę 53. Gdy nie podasz w nawiasie żadnej liczby (`float()`), domyślnie użyta zostanie wartość 53.

W tabeli 1.3 przedstawiono przybliżone liczbowe typy danych z krótkim opisem i podaną potrzebną ilością pamięci.

Tabela 1.3. Przybliżone liczbowe typy danych

Typ danych	Opis	Potrzebna pamięć
float[(n)]	Od $-1,79\text{E} + 308$ do $-2,23\text{E} - 308, 0$ i od $2,23\text{E} - 308$ do $1,79\text{E} + 308$	4 bajty dla precyzji do 24 włącznie i 8 bajtów dla precyzji od 25 do 52
real()	Od $-3,40\text{E} + 38$ do $-1,18\text{E} - 38, 0$ i od $1,18\text{E} - 38$ do $3,40\text{E} + 38$	4 bajty

UWAGA Synonimem typu real jest float (24).

Binarne typy danych

Binarne typy danych (np. varbinary, binary i varbinary(max)) przechowują dane binarne, takie jak pliki graficzne, dokumenty Worda i pliki MP3. Używane są w nich wartości szesnastkowe od 0x0 do 0xf. Typ image przechowuje do 2 gigabajtów danych poza stroną danych. Zalecaną alternatywą dla typu image jest typ varbinary(max), który może przechowywać znacznie ponad 8 kilobajtów danych binarnych i zwykle zapewnia wyższą wydajność niż typ image. W systemie SQL Server 2012 wprowadzono możliwość przechowywania obiektów typu varbinary(max) w plikach systemu operacyjnego za pomocą mechanizmu FileStream. W tym podejściu dane są zapisywane jako pliki i nie podlegają ograniczeniu 2 gigabajtów, które standardowo obowiązuje dla typu varbinary(max).

W tabeli 1.4 wymieniono binarne typy danych z krótkimi opisami i podaną ilością potrzebnej pamięci.

Tabela 1.4. Binarne typy danych

Typ danych	Opis	Potrzebna pamięć
Binary(n)	n określa liczbę cyfr szesnastkowych i przyjmuje wartość od 1 do 8000.	n bajtów
Varbinary(n)	n określa liczbę cyfr szesnastkowych i przyjmuje wartość od 1 do 8000.	1 bajt na zapisany znak + 2 bajty kosztów
Varbinary(max)	Do $2^{31} - 1$ (2 147 483 647) znaków.	1 bajt na zapisany znak + 2 bajty kosztów

Typy danych z datą i czasem

Typy datetime i smalldatetime przechowują datę i czas. Typ smalldatetime zajmuje 4 bajty i przechowuje datę oraz czas od 1 stycznia 1900 roku do 6 czerwca 2079 roku z dokładnością do jednej minuty.

Typ datetime zajmuje 8 bajtów i przechowuje datę i czas od 1 stycznia 1753 roku do 31 grudnia 9999 roku z dokładnością do 3,33 milisekundy.

W wersji SQL Server 2012 wprowadzono 4 nowe typy danych związane z datami: `datetime2`, `datetimeoffset`, `date` i `time`. Przykłady stosowania tych typów znajdziesz w dokumentacji SQL Server Books Online.

Typ danych `datetime2` to rozwinięcie typu `datetime`, obejmujące większy zakres dat. Czas zawsze jest zapisywany za pomocą godzin, minut i sekund. W typie danych `datetime2` można ustawić parametr, np. `datetime2(3)`. Liczba 3 w tym wyrażeniu oznacza, że części ułamkowe sekund należy zapisywać z dokładnością do trzech miejsc po przecinku (0,999). Parametr ten przyjmuje wartości od 0 do 7, a jego wartość domyślna to 3.

Typ danych `datetimeoffset` działa podobnie jak typ `datetime2`, przy czym obsługuje przesunięcie czasu. Może ono wynosić do 14 godzin w obie strony i uwzględnia czas UTC, co pozwala porównywać czas rejestrowany w różnych strefach czasowych.

Typ danych `date` przechowuje samą datę (jest to od dawna oczekiwana możliwość). Z kolei typ danych `time` przechowuje tylko czas. Typ `time` można też zadeklarować jako `time(n)`, aby kontrolować precyzję części ułamkowej sekund. Podobnie jak w typach `datetime2` i `datetimeoffset`, tak i tu n przyjmuje wartości z przedziału od 0 do 7.

W tabeli 1.5 przedstawione zostały typy danych z datą i czasem wraz z krótkimi opisami i podaną ilością potrzebnej pamięci.

Tabela 1.5. Typy danych z datą i czasem

Typ danych	Opis	Potrzebna pamięć
Date	Od 1 stycznia 1 roku do 31 grudnia 9999 roku.	3 bajty
Datetime	Od 1 stycznia 1753 roku do 31 grudnia 9999 roku; dokładność do 3,33 milisekundy.	8 bajtów
Datetime2(n)	Od 1 stycznia 1 roku do 31 grudnia 9999 roku; n określa precyzję części ułamkowej sekund i przyjmuje wartości od 0 do 7.	od 6 do 8 bajtów
Datetimeoffset(n)	Od 1 stycznia 1 roku do 31 grudnia 9999 roku; n określa precyzję części ułamkowej sekund i przyjmuje wartości od 0 do 7; obsługuje też przesunięcie na + lub na –.	od 8 do 10 bajtów
SmallDateTime	Od 1 stycznia 1900 roku do 6 czerwca 2079 roku; precyzja do 1 minuty.	4 bajty
Time(n)	Godziny:minuty:sekundy.9999999; n określa precyzję części ułamkowej sekund i przyjmuje wartości od 0 do 7.	od 3 do 5 bajtów

Inne systemowe typy danych

Tabela 1.6 zawiera kilka nieopisanych wcześniej typów danych.

Tabela 1.6. Inne systemowe typy danych

Typ danych	Opis	Potrzebna pamięć
Cursor	Zawiera referencję do kursora i można go używać tylko jako zmiennej lub parametru procedury składowanej.	nie dotyczy
HierarchyId	Zawiera referencję do miejsca w hierarchii.	od 1 do 892 bajtów + 2 bajty kosztów
SQL_Variant	Może zawierać wartość dowolnego systemowego typu danych z wyjątkiem typów: text, ntext, image, timestamp, xml, varchar(max), nvarchar(max), varbinary(max) oraz typów zdefiniowanych przez użytkownika. Maksymalna dozwolona wielkość to 8000 bajtów danych + 16 bajtów metadanych.	8016 bajtów
Table	Służy do przechowywania zbioru danych na potrzeby dalszego przetwarzania. Definicja przypomina instrukcję Create Table. Jest używany głównie do zwracania zbiorów wyników z funkcji tabelowych. Wartości tego typu można też stosować w procedurach składowanych i w operacjach wsadowych.	zależnie od definicji tabeli i liczby przechowywanych wierszy
Timestamp lub Rowversion	Unikatowa dla tabeli i automatycznie zapisywana wartość. Zwykle służy do znakowania wersji. Wartość ta jest automatycznie zmieniana przy wstawianiu i aktualizowaniu danych.	8 bajtów
Uniqueidentifier	Może obejmować identyfikator GUID (ang. <i>Globally Unique Identifier</i>). Identyfikatory można pobrać za pomocą funkcji Newsequentialid(). Zwraca ona wartości unikatowe dla grupy komputerów. Choć wartość ta jest zapisywana za pomocą typu binary(16), jest wyświetlana jako wartość typu char(36).	16 bajtów
XML	Z definicji zawiera znaki Unicode.	do 2 gigabajtów

UWAGA Typu danych cursor nie można stosować w instrukcji Create Table.

Typ danych XML przechowuje dokument w formacie XML lub jego fragment. Dane są zapisywane podobnie jak wartości typu nvarchar(max), a ich wielkość zależy od tego, czy w dokumencie użyto kodowania UTF-16, czy UTF-8. Typ danych XML pozwala korzystać ze specjalnych technik wyszukiwania i indeksowania. Więcej informacji na ten temat zamieszczono w rozdziale 15., „Replikacja”.

Integracja z środowiskiem CLR

W systemie SQL Server można też tworzyć własne typy danych, funkcje i procedury składowane za pomocą części środowiska CLR nazywanej SQLCLR. Pozwala to pisać bardziej skomplikowane typy danych dostosowane do potrzeb biznesowych i używane np. w językach Visual Basic i C#. Typy te są definiowane jako klasy w podstawowym języku środowiska CLR.

Edycje systemu SQL Server

Istnieje kilka edycji systemu SQL Server 2014. Funkcje dostępne w każdej z nich znacznie się różnią. Edycje, jakie można zainstalować na danej stacji roboczej lub na serwerze, zależą od używanego systemu operacyjnego. Najprostszą wersją systemu SQL Server jest SQL Express, a najbardziej rozbudowaną — Enterprise Edition.

Przegląd edycji

Oferowane są trzy podstawowe edycje systemu SQL Server 2014. Ponadto istnieją ich podwersje (nieopisane w tym miejscu) przeznaczone dla konkretnych sytuacji. Więcej informacji znajdziesz na stronie <http://www.microsoft.com/sqlserver>. W tabeli 1.7 pokazano, że system SQL Server 2014 jest dostępny w postaci trzech podstawowych produktów.

- **Enterprise Edition.** Ta edycja zapewnia wszystkie nowe możliwości systemu SQL Server, w tym wysoką dostępność, wyższą wydajność i funkcje sprawiające, że SQL Server można stosować jako system o znaczeniu strategicznym. Edycja obejmuje też wszystkie rozwiązania z zakresu BI.

Tabela 1.7. Edycje systemu SQL Server 2014

Enterprise Edition (obejmuje możliwości wersji Business Intelligence Edition)	Business Intelligence Edition (obejmuje możliwości wersji Standard Edition)	Standard Edition
Aplikacje o znaczeniu strategicznym i z 1. warstwy	Korporacyjne i skalowalne raporty analityczne	Bazy danych działów firm
Hurtownie danych	Narzędzia Power View i PowerPivot umożliwiające samodzielne przeprowadzanie analiz	Proste projekty z wykorzystaniem analiz biznesowych
Prywatna chmura i wysoce zwirtualizowane środowiska		
Rozbudowane, scentralizowane i wykorzystujące dane zewnętrzne analizy biznesowe		

Uwaga: wersja Web Edition będzie dostępna tylko na licencji SPLA (ang. *Service Provider License Agreement*).

- **Business Intelligence Edition.** Wersja BI Server Edition udostępnia pełen zestaw rozbudowanych możliwości systemu SQL Server 2014 z zakresu BI, w tym narzędzia PowerPivot, Power View, Data Quality Services i Master Data Services. Edycja ma przede wszystkim zapewnić użytkownikom końcowym funkcje związane z BI. Wersja ta doskonale sprawdzi się w projektach, w których potrzebne są zaawansowane funkcje związane z BI, ale zbędna jest zapewniana przez transakcje OLTP wydajność i skalowalność z edycji Enterprise Edition. Nowa wersja BI Edition obejmuje możliwości edycji Standard Edition, a także podstawową obsługę transakcji OLTP.
- **Standard Edition.** Ta edycja działa podobnie jak w przeszłości. Jest zaprojektowana do użytku w średniej wielkości działach firm. Posiada podstawowe mechanizmy bazodanowe i z zakresu BI. Obejmuje też pewne nowe funkcje, np. obsługę kompresji.

W tabeli 1.8 zamieszczony został przegląd wybranych, najważniejszych możliwości systemu SQL Server 2014 w poszczególnych edycjach.

Tabela 1.8. Możliwości systemu SQL Server w poszczególnych edycjach

Możliwości systemu SQL Server		Edycje systemu SQL Server	
	Standard	Business Intelligence	Enterprise
Maksymalna liczba rdzeni	16 rdzeni	16 rdzeni (dla bazy danych; maksimum systemu operacyjnego dla analiz biznesowych)	Maksimum obsługiwane przez system operacyjny
Obsługiwana pamięć	128 gigabajtów	128 gigabajtów	Maksimum obsługiwane przez system operacyjny
Podstawowa obsługa transakcji OLTP	X	X	X
Obsługa transakcji OLTP w pamięci			X
Podstawowe raporty i analizy	X	X	X
Możliwość programowania i narzędzia dla programistów	X	X	X
Zarządzanie (SSMS, zarządzanie oparte na politykach)	X	X	X
Zarządzanie danymi w korporacjach (jakość danych, Master Data Services)		X	X

Tabela 1.8. Możliwości systemu SQL Server w poszczególnych edycjach — ciąg dalszy

Możliwości systemu SQL Server		Edycje systemu SQL Server	
	Standard	Business Intelligence	Enterprise
Samoobsługowe funkcje BI (Power View, PowerPivot dla SPS)		X	X
Korporacyjne funkcje BI (model semantyczny, zaawansowane analizy)		X	X
Zaawansowane zabezpieczenia (zaawansowane inspekcje, przezroczyste szyfrowanie danych)			X
Hurtownie danych (indeksy kolumnowe, kompresja, podział na partycje)			X
Maksymalna skalowalność i wydajność			X
Wysoka dostępność	W ograniczonym stopniu	Podstawowy poziom	X
Indeksy kolumnowe			X
Funkcja AlwaysOn			X
Rozszerzenia puli buforów	X	X	X
Licencja na egzemplarze wirtualne	1 maszyna wirtualna	1 maszyna wirtualna	Bez ograniczeń

Podstawowe funkcje zapewnia wersja Standard Edition. Najważniejsze korporacyjne funkcje BI zostały umieszczone w edycji Business Intelligence. Jeśli potrzebujesz zaawansowanych hurtowni danych i wysokiej dostępności, odpowiednią wersją będzie Enterprise Edition.

W tabeli 1.8 pokazano, że silnik bazodanowy w wersji Standard Edition wykorzystuje maksymalnie 16 rdzeni. W wersji Business Intelligence Edition obowiązuje ten sam limit przy korzystaniu z bazy danych, natomiast przy stosowaniu funkcji BI (np. usług Analysis Services i Reporting Services) można użyć tylu rdzeni, ile potrafi obsłużyć system operacyjny. W wersji Enterprise Edition można zastosować wszystkie rdzenie obsługiwane przez system operacyjny.

Wersje Business Intelligence Edition i Enterprise Edition udostępniają kompletne funkcje BI systemu SQL Server 2014, w tym zarządzanie danymi korporacyjnymi, samoobsługowe funkcje BI i korporacyjne funkcje BI. W wersji Enterprise Edition dodano mechanizmy umożliwiające wykonywanie zadań o znaczeniu strategicznym oraz funkcje warstwy 1. o maksymalnej skalowalności, wydajności i dostępności. W wersji Enterprise Edition w programie Software Assurance użytkownicy otrzymują nieograniczone możliwości wirtualizacji i licencje na dowolną liczbę maszyn wirtualnych z możliwością ich przenoszenia.

Licencje

Od wersji SQL Server 2012 znacznie zmodyfikowano licencje. Jeśli klient nie uczestniczy w programie Software Assurance, nowy schemat licencji może wpływać na środowisko pracy. W tym punkcie znajdziesz tylko przegląd zmian; nie jest to wyczerpujące omówienie dostępnych licencji. Szczegółowe informacje na temat zmian uzyskasz od opiekuna klienta z Microsoftu.

Obecnie ceny i licencje systemów SQL Server są lepiej dostosowane do sposobów, w jakie klienci kupują produkty bazodanowe i do BI. Nowe podejście zapewnia klientom szereg korzyści. Oto niektóre z nich.

- **SQL Server zapewnia najlepszy na rynku poziom całkowitego kosztu posiadania.**
 - Jeśli chodzi o koszty, SQL Server nadal jest zdecydowanym liderem wśród głównych produktów bazodanowych. Wynika to z siatki cen, funkcji dostępnych w wersjach innych niż Enterprise i pomocy technicznej na najwyższym poziomie (niezależnie od edycji).
 - Klienci uczestniczący w programie Software Assurance otrzymują istotne korzyści i pomoc przy przechodzeniu na nowy model licencjonowania. Niektóre z tych korzyści to dostęp do aktualizowania i rozszerzona pomoc techniczna w czasie obowiązywania licencji.
 - Klienci, którzy podpisali umowę Enterprise Agreement, mają zapewnione największe ułatwienia przy przechodzeniu do najnowszych wersji systemu SQL Server i oszczędzają najwięcej.
- **SQL Server 2014 jest zoptymalizowany do pracy w chmurze.**
 - SQL Server 2014 to najlepiej dostosowany do wirtualizacji system bazodanowy. Dostępne są rozszerzone licencje związane z wirtualizacją, możliwość wykupienia licencji na maszyny wirtualne i doskonała obsługa technologii Hyper-V.
 - W pełni obsługiwane są hybrydowe rozwiązania łączące pracę lokalnie i w chmurze. Odpowiadają za to takie funkcje jak: obsługa plików danych systemu SQL Server na platformie Windows Azure, funkcja SQL Server Managed Backup to Windows Azure, możliwość umieszczania bazy z lokalnego egzemplarza bezpośrednio w maszynie wirtualnej platformy Windows Azure, obsługa tworzenia kopii zapasowych baz w pamięci Windows Azure BLOB Storage, a także możliwość dodawania replik bazy do grup AlwaysOn Availability Group w maszynie wirtualnej platformy Windows Azure.
- **Ceny i licencje systemu SQL Server mają umożliwiać klientom dokupowanie funkcji wraz z rozwojem firmy.**
 - Nowe usprawnione edycje są dostosowane do potrzeb związanych z bazami danych i z BI.
 - Dla firm tworzących centra danych licencje są lepiej dopasowane do możliwości sprzętu.
 - W zakresie BI licencje są dostosowane do zapewniania dostępu poszczególnym użytkownikom (większość klientów kupuje narzędzia BI właśnie w ten sposób).

Licencje na rdzeń procesora (a nie na cały procesor)

Wraz z pojawieniem się wersji SQL Server 2012 Microsoft wprowadził model licencji na rdzeń procesora. Licencje (dostępne tylko dla edycji Standard i Enterprise) są sprzedawane w dwurdzeniowych „paczkach”, dlatego dla czterordzeniowego procesora potrzebne są

po 2 paczki na gniazdo. Każda paczka kosztuje połowę tego, co licencja na procesor na wersję SQL Server 2008 R2. Haczyk polega na tym, że musisz wykupić przynajmniej 4 rdzenie na procesor.

Oto kilka przykładów:

- 2 gniazda procesora po 2 rdzenie każde wymagają 4 paczek (8 licencji na rdzeń),
- 2 gniazda procesora po 4 rdzenie każde wymagają 4 paczek (8 licencji na rdzeń),
- 2 gniazda procesora po 6 rdzeni wymagają 6 paczek (12 licencji na rdzeń),
- 2 gniazda procesora po 8 rdzeni wymagają 8 paczek (16 licencji na rdzeń).

System SQL Server w środowisku wirtualnym i licencje na serwer

Gdy uruchamiasz system SQL Server w środowisku wirtualnym, potrzebujesz licencji na przynajmniej 4 rdzenie na każdą maszynę wirtualną. Gdy maszyna wirtualna ma ponad 4 wirtualne procesory, potrzebna jest licencja na rdzenie dla każdego wirtualnego procesora przypisanego takiej maszynie.

W wersji SQL Server 2014 nadal obowiązują licencje na serwer dla klientów uczestniczących w programach Software Assurance i Enterprise Agreement. Licencje na serwer działają tak jak wcześniej: wystarczy wykupić wystarczającą liczbę licencji na rdzenie dla wersji Enterprise Edition dla danego serwera i można uruchamiać dowolną liczbę maszyn wirtualnych z systemem SQL Server. Dla wielu użytkowników jest to korzystniejsze rozwiązanie. Klienci, którzy nie uczestniczą w programach Software Assurance lub Enterprise Agreement, powinni skontaktować się z przedstawicielem handlowym Microsoftu lub pośrednikiem, ponieważ nie mogą korzystać z licencji na serwer.

Jak widać, pojawiło się wiele zmian. Zmieniły się także ceny, ale nie omawiamy ich w tym miejscu z powodu dużego zróżnicowania, zależnego od indywidualnych umów z Microsoftem. Microsoft bardzo starał się uniknąć znacznego wzrostu kosztów, dlatego nie musisz się obawiać; warto rozważyć dostępne możliwości i skontaktować się z opiekunem klienta.

Podsumowanie

W architekturze systemu SQL Server 2014 wprowadzono usprawnienia, które poprawiają wydajność, ułatwiają pracę programistom, zwiększają dostępność systemu, a także zmniejszają ogólne koszty działania. System SQL Server 2014 zapewnia też ekscytujące nowe funkcje, które umożliwiają skalowanie i zapewniają wyższą wydajność niż w którejkolwiek z wcześniejszych wersji. Warto uwzględnić pełnione obecnie i potencjalne role, a także przeanalizować, które funkcje i edycje będą odpowiednie w konkretnej sytuacji dla danej firmy.

W rozdziale 2. poznasz najlepsze praktyki związane z aktualizowaniem wcześniejszych wydań do wersji SQL Server 2014.

Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Planowanie i przeprowadzanie udanej instalacji systemu SQL Server 2014.
- Omówienie niezbędnej konfiguracji ustawianej po instalacji.
- Rozwiązywanie standardowych problemów z instalacją.

Proces instalowania systemu SQL Server 2014 jest bardzo prosty, gdy uruchomisz kreator instalacji z nośnika z systemem i zastosujesz się do instrukcji w poszczególnych ekranach kreatora, który podejmuje za użytkownika kilka ważnych decyzji. W tym rozdziale poznasz te decyzje, a także odpowiednią konfigurację pozwalającą na uzyskanie bezpiecznej i skalowalnej instalacji systemu.

Podstawowy proces instalacji obejmuje następujące kroki.

1. Planowanie systemu.
2. Przygotowywanie sprzętu i oprogramowania.
3. Instalowanie systemu operacyjnego i pakietów poprawek.
4. Konfigurowanie podsystemu wejścia-wyjścia.
5. Instalowanie systemu SQL Server i pakietów poprawek.
6. Sprawdzanie systemu.
7. Ustawienie konfiguracji po instalacji (jeśli trzeba).
8. Uporządkowanie środowiska i rozpoczęcie pracy!

W tym rozdziale koncentrujemy się na planowaniu i przeprowadzaniu instalacji.

Planowanie systemu

Pierwszym krokiem przed rozpoczęciem instalowania systemu SQL Server jest przygotowanie właściwego planu. Udana instalacja systemu SQL Server rozpoczyna się od dobrego planu. Porażka w przygotowaniu planu oznacza planowanie porażki.

Oto wybrane, niezbędne zadania i aspekty, które trzeba uwzględnić w trakcie planowania.

- Ustalenie obecnego obciążenia systemu.
- Oszacowanie wzrostu obciążenia.
- Ustalenie minimalnych wymagań sprzętowych i programowych.
- Właściwe ustawienie pamięci dla systemu i ustalenie wymagań dla operacji wejścia-wyjścia.
- Ustalenie odpowiedniej edycji systemu SQL Server.
- Określenie kolacji (ang. *collation*), lokalizacji plików i wielkości bazy tempdb dla systemu SQL Server.
- Wybranie kont powiązanych z usługami.
- Opracowanie planu konserwacji i tworzenia kopii zapasowych bazy danych.
- Ustalenie minimalnych poziomów czasu nieprzerwanej pracy i czasu reakcji.
- Przygotowanie strategii przywracania stanu po awarii.

Lista zawiera tylko kilka kwestii, o których warto pamiętać w trakcie instalowania, aktualizowania lub przenoszenia egzemplarza systemu SQL Server 2014. W następnych punktach szczegółowo opisano niektóre z tych zagadnień i wybrane najlepsze praktyki.

Opcje sprzętowe

Wybór odpowiedniej konfiguracji sprzętowej nie zawsze jest łatwy. Microsoft podaje minimalne wymagania sprzętowe dla systemu SQL Server 2014, jednak — jak sama nazwa wskazuje — są to wymagania *minimalne*, które niekoniecznie będą optymalne. Najlepszym podejściem jest zapewnienie sprzętu, który przekracza zalecane minimum i pozwoli obsłużyć obecne oraz przyszłe wymagania dotyczące zasobów.

To dlatego trzeba ustalić punkt odniesienia w postaci obecnych wymagań, a także oszacować przyszłe potrzeby. Posiadanie odpowiedniego sprzętu, który spełni wymagania w przyszłości, pozwala nie tylko zaoszczędzić pieniądze, ale też uniknąć przestojów nieuniknionych przy wymianie urządzeń.

Aby zagwarantować płynną instalację i przewidywalny poziom wydajności, zapoznaj się z minimalnymi wymaganiami sprzętowymi zalecanymi przez Microsoft. Znajdziesz je w tabeli 2.1.

Procesory

Egzemplarze systemu SQL Server 2014 obsługujące wiele transakcji i jednoczesnych połączeń powinny mieć jak największą moc obliczeniową. Można ją zapewnić za pomocą dużej liczby szybkich procesorów. Ponadto kilka wolniejszych procesorów działa szybciej niż jeden szybki procesor (np. dwa procesory z taktowaniem 1,6 GHz są szybsze niż jeden procesor z taktowaniem 3,2 GHz).

Tabela 2.1. Minimalne wymagania sprzętowe systemu SQL Server 2014

Komponent	Wymagania
Procesor	<p>Dla instalacji 64-bitowych zalecane są procesory o szybkości 1,4 GHz lub wyższej: AMD Opteron, Athlon 64, Intel Pentium IV z obsługą architektury Intel EM64T lub Xeon z obsługą architektury Intel EM64T.</p> <p>Uwaga</p> <p>Choć w dokumentacji Books Online opisane są wymagania dla komputerów 32-bitowych, proces instalacji na takim sprzęcie kończy się błędem i wyświetleniem informacji o tym, że maszyny 32-bitowe nie są obsługiwane.</p>
Pamięć operacyjna	1 gigabajt (512 megabajtów dla Express Edition); zalecane są 4 gigabajty.
Pamięć dyskowa	<p>Silnik bazy danych, pliki z danymi, replikacja, wyszukiwanie pełnotekstowe i usługi Data Quality Services: 811 megabajtów</p> <p>Usługi Analysis Services i pliki z danymi: 345 megabajtów</p> <p>Usługi Reporting Services i narzędzie Report Manager: 304 megabajty</p> <p>Usługi Integration Services: 591 megabajtów</p> <p>Usługi Master Data Services: 243 megabajty</p> <p>Komponenty klienckie (z wyłączeniem komponentów dokumentacji SQL Server Books Online i narzędzi usług Integration Services): 1823 megabajty</p> <p>Komponenty dokumentacji SQL Server Books Online pozwalające wyświetlać pomoc i zarządzać nią: 375 kilobajtów</p>

Nowsze procesory mają po kilka rdzeni w jednym fizycznym gnieździe. Procesory wielordzeniowe oferują wiele zalet, w tym oszczędność miejsca i zużycia energii. Procesory wielordzeniowe pozwalają uruchomić więcej niż jeden egzemplarz systemu SQL Server 2014 na tym samym fizycznym serwerze. Aby uzyskać ten efekt, zastosuj egzemplarze nazwane lub maszyny wirtualne. Na jednym fizycznym serwerze można wtedy uruchomić tyle egzemplarzy systemu SQL Server 2014, na ile pozwalają sprzęt i licencje. Pozwala to znacznie zmniejszyć ilość miejsca potrzebnego na centrum danych, ponieważ wiele serwerów fizycznych można zastąpić jednym. Taka konsolidacja pomaga ograniczyć koszty zużycia energii, ponieważ do sieci podłączonych jest mniej fizycznych serwerów.

W wersji SQL Server 2012 wprowadzono licencje oparte na rdzeniach. Są one dostępne także w wersji SQL Server 2014. W używanym dla tego systemu modelu licencji opartych na rdzeniach wymagane jest uzyskanie licencji na każdy rdzeń wielordzeniowego procesora. Ten model dotyczy zarówno serwerów fizycznych, jak i maszyn wirtualnych. Więcej informacji o licencjach systemu SQL Server 2014 podano w rozdziale 1.

Pamięć

Pamięć to ważny zasób dla optymalnego działania systemu SQL Server 2014. Dobrze zaprojektowane systemy bazodanowe wykorzystują dostępną pamięć i wczytują jak najwięcej danych ze stron pamięci podręcznej z buforów.

Pamięć trzeba udostępnić zarówno dla egzemplarza systemu SQL Server, jak i dla systemu operacyjnego. Na serwerze z systemem Windows, na którym działa egzemplarz systemu SQL Server, w miarę możliwości należy unikać instalowania innych aplikacji o dużych wymaganiach pamięciowych.

Dobrym punktem wyjścia przy ustalaniu potrzebnej ilości pamięci jest określenie liczby stron danych na każdą bazę przechowywaną w danym egzemplarzu systemu SQL Server. Należy też uwzględnić statystyki dotyczące wykonywania kwerend, np. minimalne, maksymalne i średnie wykorzystanie pamięci przy typowym obciążeniu. Celem jest umożliwienie systemowi SQL Server zapisania w pamięci podręcznej jak największej liczby stron danych i planów wykonania, aby uniknąć kosztownych odczytów stron danych z dysku i generowania planów wykonania.

Pamiętaj też o ograniczeniach pamięciowych obowiązujących dla poszczególnych edycji systemu SQL Server. Przykładowo edycja SQL Server 2014 Enterprise Edition obsługuje do 2 terabajtów pamięci RAM, edycje Standard Edition i Business Intelligence Edition obsługują do 128 gigabajtów pamięci RAM, a edycja Express Edition — 1 gigabajt takiej pamięci.

Nową funkcją wprowadzoną w systemie SQL Server 2014 jest wbudowana obsługa transakcji OLTP w pamięci. Ten mechanizm (nazwany *Hekaton*) jest dostępny tylko w wersji Enterprise Edition. Obecnie dane mogą w całości znajdować się w pamięci, co zmniejsza koszty wykonywania operacji wejścia-wyjścia związane z dostępem do dysku. Szczegółowe omówienie mechanizmu Hekaton zamieszczono w rozdziale 9.

Pamięć dyskowa

System pamięci w SQL Server 2014 wymaga specjalnego omówienia, ponieważ może w znacznym stopniu obniżyć wydajność bazy, jeśli działa powoli. W trakcie planowania pamięci dla baz systemu SQL Server 2014 trzeba uwzględnić kwestie dostępności, niezawodności, przepustowości i skalowalności.

Aby przetestować i sprawdzić wydajność systemu pamięci, musisz zebrać ważne wskaźniki, np. maksymalną liczbę żądań wejścia-wyjścia na sekundę, przepustowość (w megabajtach na sekundę) i opóźnienie operacji wejścia-wyjścia. W tabeli 2.2 znajduje się krótki opis tych trzech najważniejszych wskaźników.

Tabela 2.2. Podstawowe wskaźniki związane z pamięcią

Wskaźnik	Opis
Liczba żądań wejścia-wyjścia na sekundę	Liczba równoległych żądań, które system pamięci potrafi obsłużyć w sekundę. Wartość powinna być wysoka, zwykle od 150 do 250 na jeden dysk SAS o szybkości 15 000 RPM oraz od 1000 do 1 000 000 dla korporacyjnych dysków SSD i sieci SAN (w zależności od konfiguracji i producenta).
Przepustowość (w megabajtach na sekundę)	Ilość danych, jaką system pamięci potrafi wczytać lub zapisać w ciągu sekundy. Wartość ta powinna być wysoka.
Opóźnienie operacji wejścia-wyjścia (w milisekundach)	Opóźnienie między operacjami wejścia-wyjścia. Wartość ta powinna być zerowa lub bliska zeru.

Omówione wskaźniki możesz zebrać za pomocą bezpłatnych narzędzi (np. SQLIO, SQLIOSim, IOMeter i CrystalDiskMark). Opisywanie korzystania z tego typu narzędzi wykracza poza zakres rozdziału. Na stronie <http://msdn.microsoft.com/en-us/library/cc966412.aspx> znajdziesz dobrą dokumentację tego zagadnienia.

Dwa podstawowe rodzaje pamięci używane w instalacjach systemu SQL Server to DAS i SAN. Poniższe punkty zawierają ich omówienie.

Pamięć DAS

DAS (ang. *Direct-Attached Storage*) to najprostsze do zrozumienia podejście. Dyski są zamknięte w obudowie serwera i podłączone bezpośrednio do kontrolera. Dyski można też umieścić w zewnętrznej obudowie bezpośrednio podłączonej kablem do adaptera HBA (ang. *Host Bus Adapter*). Nie wymaga to dodatkowego sprzętu, np. przełączników.

Głównymi zaletami pamięci DAS jest łatwość stosowania i konserwacji oraz niskie koszty. Główna wada to ograniczona skalowalność. Choć w ostatnich latach w systemach pamięci DAS zaczęły pojawiać się funkcje dostępne wcześniej tylko w zaawansowanych jednostkach SAN, nadal występują ograniczenia, takie jak liczba dysków i ilość danych, jakie można obsłużyć, liczba serwerów, do których można podłączyć pamięć DAS, oraz odległość między jednostkami pamięci a serwerem.

Najważniejszymi cechami różnicującymi oba podejścia są połączenia z serwerem i odległość od niego, ponieważ pamięć DAS wymaga fizycznego łącza między jednostkami pamięci a serwerami, co ogranicza liczbę serwerów, które można jednocześnie podłączyć do pamięci. Ograniczona jest też odległość między jednostkami pamięci a serwerem. Zwykle nie przekracza ona kilku metrów.

Sieci SAN

SAN (ang. *Storage Area Network*) to wyspecjalizowane sieci łączące urządzenia pamięciowe udostępniane serwerom jako pamięci DAS. Sieć urządzeń pamięciowych jest oparta na bardzo szybkich urządzeniach FC (ang. *Fibre Channel*) nazywanych przełącznikami fabric lub za pomocą protokołu iSCSI przy użyciu zwykłych przełączników ethernetowych.

Jedną z wielkich zalet sieci SAN jest możliwość rozciągnięcia ich na dużych obszarach, zwykle z wykorzystaniem trasowania TCP/IP i dedykowanych połączeń sieci WAN (ang. *Wide Area Network*). To rozwiązanie pozwala firmom stosować mechanizmy, takie jak replikacja pamięci, w odległych od siebie centrach danych, co chroni przed skutkami poważnych awarii.

Ponadto sieci SAN zapewniają najwyższą niezawodność i skalowalność dla systemów bazodanowych o znaczeniu strategicznym. Dobrze zaprojektowana sieć SAN oferuje znacznie wyższą przepustowość i niższe opóźnienia operacji wejścia-wyjścia niż pamięć DAS. Sieci SAN mogą też obsługiwać znacznie więcej macierzy dyskowych niż pamięć DAS.

Głównymi wadami sieci SAN są wyższe koszty oraz skomplikowany proces instalowania i konserwacji.

Wybór pamięci odpowiedniego rodzaju

Rodzaj pamięci stosowany w instalacji systemu SQL Server zależy od potrzeb użytkownika. Z wcześniejszego porównania dowiedziałeś się, że pamięć DAS jest znacznie tańsza i łatwiejsza w konfigurowaniu oraz konserwowaniu niż sieć SAN. Jednak sieci SAN zapewniają wiele korzyści związanych z wydajnością, dostępnością i skalowalnością.

Ważną kwestią, którą należy uwzględnić przy wybieraniu systemu pamięci, są używane dyski i sposób ich połączenia. W podejściach DAS i SAN używane są macierze dyskowe, zwykle konfigurowane w taki sposób, aby uzyskać pulę pamięci udostępnianą serwerowi jako pojedyncza jednostka.

W następnym podpunkcie opisano różne rodzaje dysków i sposoby łączenia ich za pomocą technologii RAID.

Dyski

Wcześniej wspomnieliśmy, że należy rozważyć przepustowość potrzebną do spełnienia wymagań związanych z operacjami wejścia-wyjścia. Aby poradzić sobie z wysokimi wymaganiami w tym obszarze, często trzeba podzielić operacje zapisu i odczytu na dużą liczbę szybkich dysków.

Rozdzielanie operacji wejścia-wyjścia oznacza, że na każdym dysku zapisywane są małe porcje danych, które później są scalane. W tego typu pamięci rozproszonej żaden dysk nie zawiera kompletnych danych. Dlatego awaria jednego dysku może prowadzić do utraty wszystkich informacji. Przy wyborze systemów pamięci trzeba więc zawsze pamiętać o niezawodności. Aby uniknąć utraty danych w wyniku awarii dysku, dyski są w specjalny sposób łączone w *macierze dyskowe* typu RAID. Można je tak skonfigurować, aby zapewnić zarówno wysoką przepustowość, jak i niezawodność. Wybór odpowiedniego poziomu macierzy RAID to bardzo istotna decyzja, która wpływa na ogólną wydajność serwera. W tabeli 2.3 opisano poziomy macierzy RAID najczęściej stosowane w środowisku systemu SQL Server.

Tabela 2.3. Często stosowane poziomy macierzy RAID

Poziom RAID	Opis
RAID 0	Jest to tzw. <i>striping</i> . Polega na połączeniu dwóch lub więcej dysków w celu uzyskania większej pojemności. Nie zapewnia odporności na awarie. Pozwala na szybki odczyt i zapis.
RAID 1	Są to tzw. <i>dyski lustrzane</i> . Na obu dyskach są zapisywane identyczne dane. Dzięki temu awaria jednego dysku nie prowadzi do utraty danych. Spowalnia zapis, a dostępna jest tylko połowa całej pamięci.
RAID 1+0	Nazywany też RAID 10. Dyski lustrzane dla macierzy ze <i>stripingiem</i> . Poprawia wydajność zapisu i zapewnia odporność na awarie. Dostępna jest tylko połowa całej pamięci.
RAID 0+1	<i>Striping</i> dysków lustrzanych. Daje mniejszą odporność niż poziom RAID 1+0. Zapewnia wysoką wydajność zapisu.
RAID 5	Zapewnia odporność na awarię jednego dysku. Zapis jest rozproszony między dyskami. Daje szybki odczyt przy wolnym zapisie. Łączna pojemność jest nieco mniejsza niż suma pojemności dysków.
RAID 6	Zapewnia odporność na awarię dwóch dysków. Odczyt jest szybszy w porównaniu z macierzami RAID 5, ale zapis jest wolniejszy z powodu obliczeń parzystości. Utrata pojemności jest podobna jak w macierzach RAID 5.

Technologia SSD (ang. *Solid State Drive*) staje się coraz popularniejsza dzięki spadkowi kosztów i wzrostowi niezawodności. Dyski SSD zapewniają nawet 100 razy szybszy odczyt i zapis niż tradycyjne dyski obrotowe. W systemie SQL Server może to być przydatne — zwłaszcza dla baz danych o wysokich wymaganiach dotyczących operacji wejścia-wyjścia.

Dyski SSD są używane coraz częściej. Po części jest to spowodowane wyższą niezawodnością i spadkiem ich cen. Niektórzy producenci systemów SAN udostępniają też macierze oparte na dyskach SSD.

UWAGA Nawet przy wyższej niezawodności, jaką zapewniają dyski SSD dzięki wyeliminowaniu obracających się talerzy, należy chronić dane za pomocą macierzy RAID, ponieważ dyski SSD są narażone na awarie i uszkodzenia komponentów elektronicznych.

Innym ważnym elementem, który trzeba uwzględnić przy wyborze systemu pamięci (zwłaszcza wtedy, kiedy myślisz o zastosowaniu pamięci DAS), jest kontroler dysków. W następnym podpunkcie zapoznasz się z cechami kontrolerów dysków, od których zależy zwiększenie wydajności i niezawodności pamięci DAS.

Kontrolery dysków

Kontrolery dysków to ważne urządzenia, które przy używaniu bezpośrednio podłączonych dysków trzeba wybierać ze specjalną starannością. Kontrolery mogą być głównym źródłem wąskiego gardła w operacjach wejścia-wyjścia, ponieważ mają limit przepustowości.

Szybkie dyski nie wystarczą, aby zbudować szybki system pamięci. Niewłaściwie skonfigurowane kontrolery dysków mogą być dodatkowym powodem opóźnień. Większość kontrolerów dysków ma ustawienia, które można dostosować do obciążenia. Przykładowo w systemach z dużą liczbą transakcji możesz tak skonfigurować kontroler dysków, aby działał optymalnie dla dużego odsetka operacji zapisu. Z kolei w systemach bazodanowych używanych do generowania raportów (np. w hurtowniach danych lub magazynach danych operacyjnych) możesz ustawić kontrolery pod kątem dużego odsetka operacji odczytu.

Następnym ważnym zagadnieniem związanym z kontrolerami dysków jest pamięć podręczna dla operacji zapisu. Choć ten mechanizm pozwala przyspieszyć zapis, może prowadzić do nieoczekiwanych skutków, np. utraty danych lub uszkodzenia bazy.

Kontrolery dysków przyspieszają zapis dzięki tymczasowemu przechowywaniu danych w pamięci podręcznej i przesyłaniu ich w porcjach na dysk. Gdy dane są zapisywane w pamięci podręcznej kontrolera, system SQL Server uznaje, że transakcja jest zatwierdzona. W rzeczywistości dane nie są wtedy zatwierdzone na dysku i istnieją tylko w pamięci. Jeśli serwer niespodziewanie zakończy pracę bez oczekiwania na zapisanie danych z pamięci podręcznej na dysku, dane nie trafią do dziennika transakcji bazy i zostaną utracone.

W środowiskach z bazami o znaczeniu strategicznym należy rozważyć zastosowanie kontrolerów korporacyjnych z dodatkowymi źródłami zasilania (np. zasilaczem UPS lub wewnętrznymi bateriami), aby uniknąć możliwej utraty danych.

Możliwości w obszarze oprogramowania i instalacji

Następny krok wymaga upewnienia się, że wybrane ważne opcje konfiguracyjne są poprawnie ustawione. Musisz sprawdzić, czy ustawione są odpowiednie konta usług i właściwa lokalizacja dla plików bazy danych.

Kolacja

Kolacja (ang. *collation*) systemu SQL Server określa zestaw reguł przechowywania, sortowania i porównywania znaków. Kolacja jest ważna, ponieważ określa używane strony kodowe. Poszczególne strony kodowe obsługują różne znaki i działają w inny sposób przy sortowaniu i porównywaniu łańcuchów znaków. Ustawienie niewłaściwej kolacji może wymagać ponownej instalacji systemu SQL Server, ponieważ zmienianie jej w działającym egzemplarzu sprawia problemy.

Aby ustalić odpowiednią kolację, musisz poznać wymagania firmy i klientów dotyczące ustawień regionalnych, sortowania, traktowania wielkich i małych liter, a także akcentów.

Stronę kodową używaną na serwerze z systemem Windows znajdziesz w oknie *Panel sterowania/Opcje regionalne i językowe*. Strona kodowa ustawiona dla serwera nie zawsze jest odpowiednia dla egzemplarza systemu SQL Server.

Poniżej opisano dwa rodzaje kolacji: systemu SQL Server i systemu Windows.

Kolacja systemu SQL Server

Kolacje systemu SQL Server wpływają na stronę kodową używaną do zapisywania danych w kolumnach typów char, nchar, nvarchar, varchar i text. Od strony kodowej zależy przebieg porównywania i sortowania wartości tych typów. Jeśli np. zastosujesz pierwszą z poniższych instrukcji SELECT dla bazy danych z kolacją, w której wielkość znaków ma znaczenie, nie zwróci ona pracownika o imieniu Marcin zapisanym w standardowy sposób:

```
SELECT FROM Employees
WHERE EmployeeFirstName='MARCIN'
```

Wyniki: <none>

```
SELECT FROM Employees
WHERE EmployeeFirstName='Marcin'
```

Wyniki: Marcin

Kolacja systemu Windows

Kolacje systemu Windows wykorzystują reguły oparte na ustawieniach regionalnych systemu operacyjnego. Domyślnie porównania i sortowanie są zgodne z regułami słownikowymi wybranego języka. Możesz określić sposób traktowania danych binarnych, wielkości znaków, akcentu, znaków kana i znaków o różnej długości. Ważne jest to, że kolacje systemu Windows gwarantują, iż jedno- i dwubajtowe zbiory znaków działają tak samo przy sortowaniu i porównywaniu.

Uwzględnianie wielkości znaków

Kolacja może uwzględniać lub ignorować wielkość znaków. Przy *uwzględnianiu wielkości znaków* U jest różne od u. Dotyczy to wszystkich elementów, dla których obowiązuje dana kolacja (baz master, model, resource, tempdb i msdb), oraz wszystkich danych w wymienionych bazach. Oto ważna uwaga — pomyśl, jakie dane są przechowywane w tych bazach. Otóż obejmują one dane ze wszystkich tabel systemowych, co oznacza, że wielkość znaków jest uwzględniana także w nazwach obiektów.

Kolejność sortowania

Wybrana kolacja wpływa też na sortowanie. Przy kolejności opartej na wartościach binarnych (np. Latin1_General_BIN) ważna jest wartość bitów znaków. W tym podejściu wielkość znaków ma znaczenie. Przyjrzyj się instrukcji SELECT z rysunku 2.1. Załóżmy, że uruchomiono ją dla tabeli z następującymi imionami pracowników: Marta, Tomek, marta i tomek. Jeśli użyjesz sortowania słownikowego (np. Latin1_General_CS_AI), otrzymasz zbiór wyników widoczny na rysunku 2.2.

The screenshot shows a SQL query window with the following text:

```
SELECT name FROM names
ORDER BY name
COLLATE Latin1_General_BIN ASC;
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	name
1	Marta
2	Tomek
3	marta
4	tomek

Rysunek 2.1. Zapytanie z sortowaniem binarnym

The screenshot shows a SQL query window with the following text:

```
SELECT name FROM names
ORDER BY name
COLLATE Latin1_General_CS_AI ASC;
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	name
1	marta
2	Marta
3	tomek
4	Tomek

Rysunek 2.2. Zapytanie z sortowaniem słownikowym

Konta usługowe

Konta usługowe są ważnym elementem modelu zabezpieczeń. Gdy wybierasz konta usługowe, stosuj zasadę *najmniejszych uprawnień*. Konta usługowe powinny mieć minimalne uprawnienia potrzebne do działania. Dla każdej usługi należy ustawić inne konto, aby można było śledzić pracę poszczególnych usług. Dla kont usługowych zawsze trzeba stosować silne hasła.

Przy tworzeniu kont usługowych masz do wyboru kilka możliwości.

- **Konto systemu Windows lub domeny.** Tego typu konto usług Active Directory lub systemu Windows jest preferowane dla usług systemu SQL Server wymagających dostępu do sieci.
- **Lokalne konto systemowe.** Tego konta o wysokich uprawnieniach nie należy stosować dla usług, ponieważ działa jak komputer w sieci i nie ma hasła. Skuteczny atak na proces, w którym używa się lokalnego konta systemowego, może prowadzić do przejęcia systemu bazodanowego.
- **Lokalne konto usługowe.** To specjalne, wbudowane konto ma takie same uprawnienia jak członkowie grupy Users. Dostęp do sieci odbywa się w postaci anonimowej sesji puste. To konto nie jest obsługiwane.

- **Konto usług sieciowych.** Działa podobnie jak lokalne konto usługowe, przy czym zezwala na dostęp do sieci z danymi uwierzytelniającymi, takimi jak dla konta komputera. Nie należy stosować go dla kont systemu SQL Server ani usług SQL Agent Service.
- **Lokalne konto serwera.** Tego rodzaju lokalne konto systemu Windows to najbezpieczniejsza metoda używania usług, które nie wymagają dostępu do sieci.

Dla systemów produkcyjnych powinieneś zastosować dedykowane konta systemu Windows lub domeny. Niektóre firmy używają jednego konta domeny dla wszystkich egzemplarzy systemu SQL Server, natomiast inne dla każdej usługi tworzą odrębne konta domeny.

Korzystanie z jednej usługi dla wszystkich egzemplarzy systemu SQL Server to prosta droga do katastrofy. Jeśli jedno konto zostanie przejęte lub zablokowane, wszystkie egzemplarze przestaną funkcjonować poprawnie. Jeżeli konto usługowe systemu SQL Server zostanie zablokowane, egzemplarza nie będzie można uruchomić po ponownym włączeniu usługi. W środowisku z klastrami może to prowadzić do nieudanego testu stanu i wyłączenia danego klastra. Najlepiej tworzyć odrębne konta usługowe dla każdego egzemplarza systemu SQL Server.

Instalowanie systemu SQL Server

W tym podrozdziale poznasz różne typy instalacji: nowe instalacje, instalacje równoległe i aktualizacje. Dowiesz się też, jak przeprowadzać instalacje nadzorowane i nienadzorowane za pomocą graficznego interfejsu użytkownika, wiersza poleceń, plików konfiguracyjnych i skryptów powłoki PowerShell systemu Windows. Więcej informacji na temat aktualizowania podajemy w rozdziale 3., „Najlepsze praktyki aktualizowania systemu do wersji SQL Server 2014”.

Nowe instalacje

Nowa instalacja ma miejsce, gdy przeprowadzasz ją na serwerze, na którym nie ma żadnych komponentów systemu SQL Server. Sprawdź katalogi i rejestr, aby się przekonać, że w systemie operacyjnym nie ma pozostałości po wcześniejszych instalacjach systemu SQL Server.

Instalacje równoległe

SQL Server obsługuje *instalacje równoległe*. Polegają one na tym, że na jednym serwerze działa wiele egzemplarzy systemu SQL Server. Wersja SQL Server 2014 pozwala używać na tym samym komputerze wielu egzemplarzy silnika bazy danych i usług Reporting Services oraz Analysis Services. Ponadto może pracować równoległe ze starszymi wersjami tego systemu. Na jednym serwerze może działać tylko jeden egzemplarz domyślny, jeśli zatem domyślny jest istniejący egzemplarz, nowy trzeba zainstalować jako nazwany.

Największym problemem w instalacjach równoległych jest współzawodnictwo o zasoby. Trzeba skonfigurować pamięć w taki sposób, aby żaden egzemplarz nie próbował zająć całej fizycznej pamięci. Problemem może być także współzawodnictwo o zasoby wejścia-wyjścia, jeśli dla plików baz danych z różnych egzemplarzy używane są te same zasoby.

Aktualizowanie

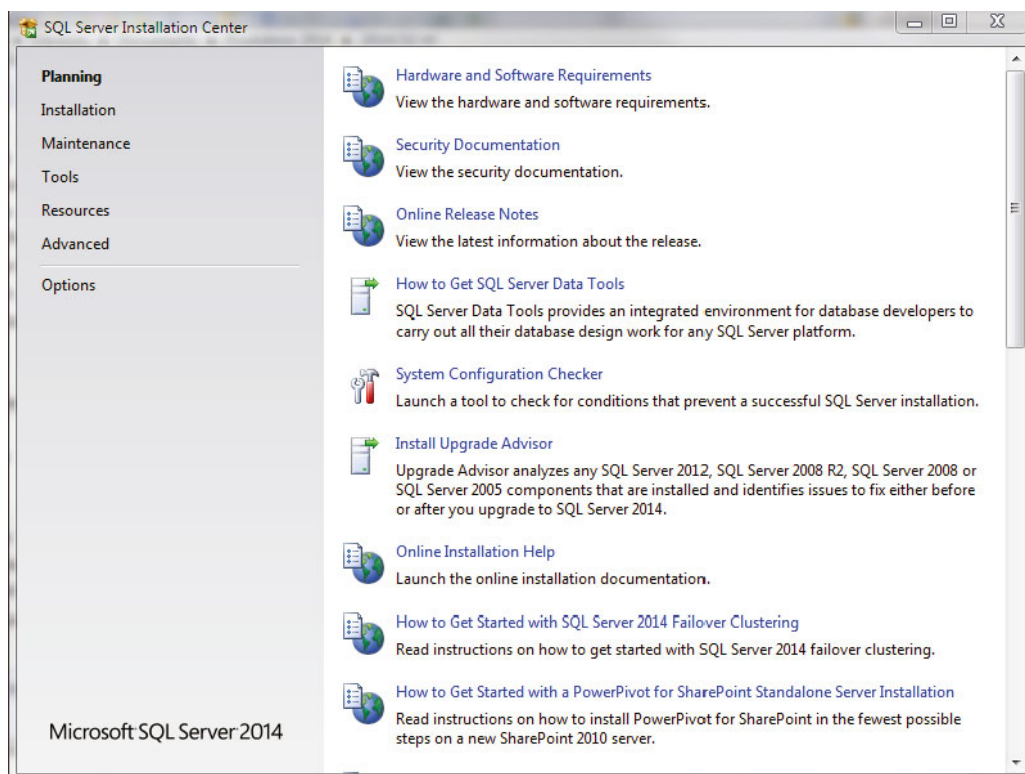
Jeśli w komputerze znajdują się już komponenty systemu SQL Server, można zaktualizować istniejący egzemplarz. Wtedy system SQL Server jest instalowany na istniejącym egzemplarzu (jest to tzw. *aktualizacja w miejscu*). Aby zaktualizować system SQL Server 2014 ze starszej wersji, uruchom kreator aktualizacji w narzędziu *SQL Server Installation Center* (wybierz opcję *Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012* w zakładce *Installation*).

Instalacje nadzorowane

Najprostszy i najczęściej stosowany sposób instalowania systemu SQL Server polega na instalacji nadzorowanej przy użyciu graficznego interfejsu użytkownika kreatora instalacji. Instalacja nadzorowana wymaga częstych interwencji użytkownika, który podaje informacje i parametry potrzebne do zainstalowania systemu SQL Server 2014.

Aby rozpocząć nadzorowaną instalację systemu SQL Server 2014, wykonaj następujące kroki.

1. Uruchom plik *Setup.exe* z nośnika instalacyjnego z systemem SQL Server 2014. Pojawi się okno *Installation Center* przedstawione na rysunku 2.3.



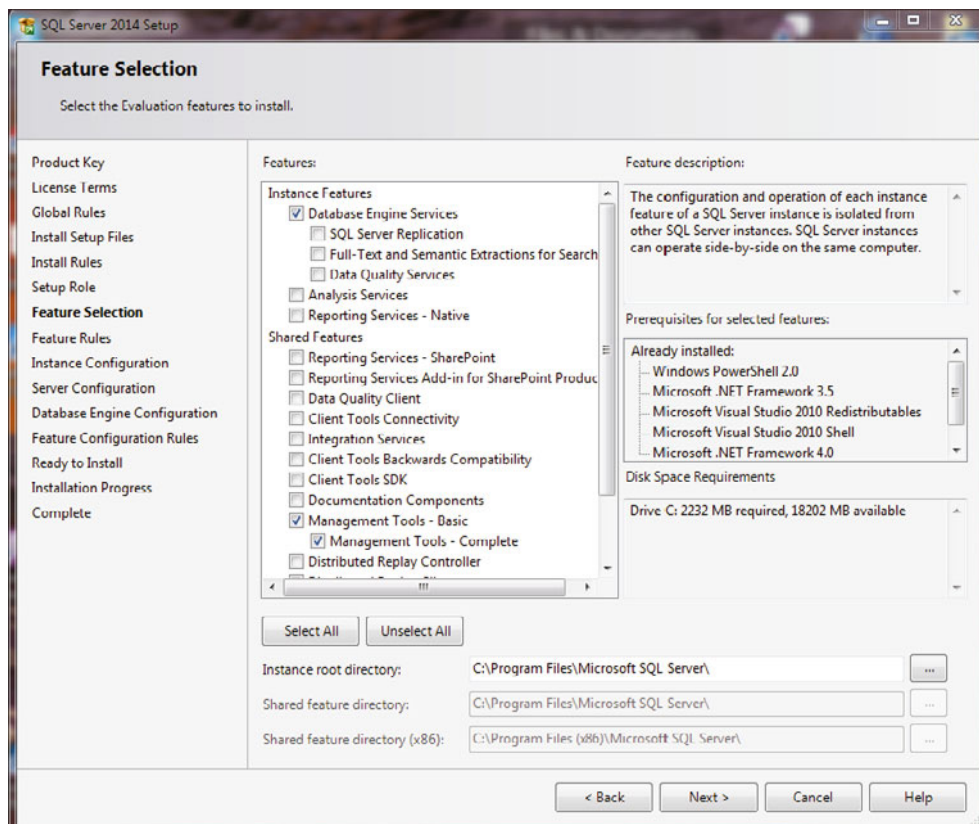
Rysunek 2.3. Okno Installation Center

2. Otwórz zakładkę *Installation* po lewej stronie, a następnie kliknij pierwszą opcję po prawej (*New SQL Server Stand-alone Installation* lub *Add Feature to an Existing Installation*). Uruchomiony zostanie kreator instalacji systemu SQL Server 2014.

UWAGA Pojawi się okno wyskakujące z informacją, że instalacja jest w toku, i z prośbą o poczekanie.

3. Zainicjowanie instalacji może chwilę potrwać.
4. W zależności od używanego nośnika i umowy licencyjnej na następnym ekranie może pojawić się prośba o wybranie edycji systemu SQL Server i wprowadzenie klucza produktu. Kliknij przycisk *Next*, aby kontynuować.
5. Pojawi się ekran *License Terms*. Zaakceptuj warunki licencji i kliknij przycisk *Next*.
6. Pojawi się ekran *Global Roles*, po czym kreator przejdzie do następnego ekranu.
7. Otworzy się ekran *Microsoft Update*. To firma powinna określić minimalny numer wersji. Nie zaznaczaj widocznego tu pola. Kliknij przycisk *Next*, a w przyszłości aktualizuj system SQL Server zgodnie z wytycznymi firmy.
8. Pojawi się ekran *Product Updates*. Kliknij przycisk *Next*, aby kontynuować.
9. Otworzy się ekran *Install Setup Files*, po czym kreator przejdzie do następnego ekranu.
10. Na ekranie *Install Rules* widoczna jest lista problemów, które mogą wystąpić w trakcie ładowania plików wspomagających proces instalacji. Po zakończeniu tego kroku kliknij przycisk *Next*, jeśli wszystkie warunki zostały spełnione.
11. Pojawi się ekran *Setup Role*. Wybierz opcję *SQL Server Feature Installation* i kliknij przycisk *Next*.
12. Otworzy się ekran *Feature Selection*. Wybierz opcje *Database Engine Services* i *Management Tools — Basic*, po czym kliknij przycisk *Next*. Na rysunku 2.4 przedstawiono listę funkcji dostępnych do zainstalowania w systemie SQL Server 2014.
13. Jeśli jakiś element może zablokować proces instalacji, pojawi się ekran *Feature Rules*. Kliknij przycisk *Next*.
14. Otworzy się ekran *Instance Configuration*. Możesz na nim określić, czy egzemplarz ma zostać zainstalowany jako domyślny, czy jako nazwany. Możesz tu także podać identyfikator egzemplarza i zmienić domyślny katalog główny. Następnie kliknij przycisk *Next*.
15. Pojawi się ekran *Server Configuration* przedstawiony na rysunku 2.5. Podaj konta usługowe dla silnika bazy danych systemu SQL Server oraz narzędzi SQL Server Agent i SQL Server Browser. Określ też kolację systemu SQL Server 2014. Następnie kliknij przycisk *Next*.

UWAGA Działając według najlepszych praktyk z obszaru bezpieczeństwa, zawsze wybieraj konta usługowe zgodnie z zasadą minimalnych uprawnień. Jeśli to możliwe, unikaj ustawiania kont usługowych mających podniesione uprawnienia do domeny lub do serwera.



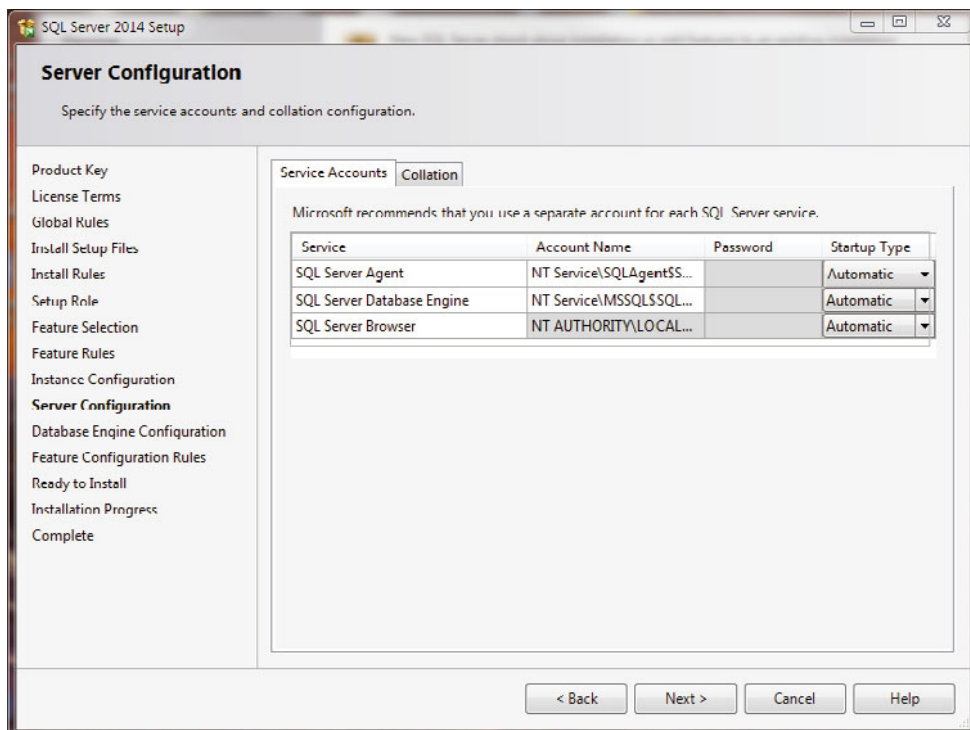
Rysunek 2.4. Funkcje, jakie można zainstalować w systemie SQL Server 2014

16. Pojawi się ekran *Database Engine Configuration*. Ustaw opcje *Authentication Mode*, *SQL Server Administrators* i domyślne katalogi danych oraz włącz mechanizm *FILESTREAM*. Ważne jest, aby na tym ekranie ustawić przynajmniej jedno konto z uprawnieniami administratora systemu SQL Server. Ponadto trzeba ustawić tryb uwierzytelniania (pod nagłówkiem *Authentication Mode*) na wartość *Windows authentication mode* lub *Mixed Mode*. Gdy wybrana jest opcja *Windows authentication mode*, logować się mogą tylko uwierzytelnieni użytkownicy systemu Windows. W trybie *Mixed Mode* logować się mogą zarówno użytkownicy systemu Windows, jak i użytkownicy systemu SQL Server. Kliknij przycisk *Next*.

UWAGA Nie pozostawiaj pustego hasła dla konta SysAdmin. Zawsze przypisuj do niego mocne hasło, a po zakończeniu instalacji wyłączaj to konto, aby uniknąć prób włamania.

17. Otworzy się okno *Ready to Install*. Na tym etapie kreator instalacji zebrał już wszystkie potrzebne informacje i wyświetla je w celu przejrzania przed rozpoczęciem procesu instalacji. Kliknij przycisk *Install*, aby rozpocząć ten proces.

To kończy proces nadzorowanej instalacji systemu SQL Server 2014.



Rysunek 2.5. Konfigurowanie systemu SQL Server 2014

UWAGA Microsoft udostępnia przykładowe bazy danych dla systemu SQL Server 2014. Możesz je bezpłatnie pobrać (razem z plikami przykładowych projektów) z witryny Codeplex.com: <http://msftdbprodsamples.codeplex.com/>.

Instalacja nienadzorowana

System SQL Server umożliwia przeprowadzenie instalacji nienadzorowanej z parametrami podanymi w wierszu poleceń lub w pliku konfiguracyjnym. Instalacja nienadzorowana pozwala zainstalować ten system z dokładnie taką samą konfiguracją na wielu serwerach, przy czym interwencje użytkownika w proces instalacji są potrzebne w bardzo niewielkim stopniu (lub w ogóle nie są konieczne). Wszystkie dane na ekranach i odpowiedzi w oknach dialogowych są automatycznie uzupełniane na podstawie informacji z pliku konfiguracyjnego lub parametrów z wiersza poleceń.

Instalacja nienadzorowana uruchamiana z wiersza poleceń

Aby przeprowadzić instalację systemu SQL Server 2014 z wiersza poleceń, wykonaj następujące kroki.

1. Otwórz okno wiersza poleceń z podniesionymi uprawnieniami administratora. W tym celu kliknij plik wiersza poleceń prawym przyciskiem myszy i wybierz opcję *Uruchom jako administrator*. Pojawi się okna wiersza poleceń.

2. W wierszu poleceń wpisz poniższą instrukcję i wciśnij klawisz *Enter*:

```
D:\setup.exe /ACTION=install /QS /INSTANCENAME="MSSQLSERVER"  
/IACCEPTSQLSERVERLICENSETERMS=1  
/FEATURES=SQLENGINE,SSMS  
/AGTSVCACCOUNT="TwojaDomena\Administrators"  
/AGTSPASSWORD="TwojeHaslo"  
/SQLSYSADMINACCOUNTS="TwojaDomena\Administrators"
```

UWAGA Ścieżka do pliku instalacyjnego może się różnić w zależności od używanego nośnika. Parametry są zależne od instalowanych funkcji. Ponadto należy zmienić wartość wpisu /SQLSYSADMINACCOUNTS na poprawną nazwę domeny i konta użytkownika.

Ta instrukcja wiersza poleceń wywołuje nienadzorowaną instalację silnika bazodanowego systemu SQL Server i narzędzi z grupy SQL Server Management Tools — Basic. W tabeli 2.4 opisane są parametry wiersza poleceń użyte w tej instrukcji.

Tabela 2.4. Użyte parametry wiersza poleceń

Parametr	Opis
/ACTION	Określa wykonywaną operację. Tu jest to instalacja.
/QS	Powoduje uruchomienie instalacji i wyświetlenie informacji o postępie, przy czym dane wejściowe nie są pobierane i nie pojawiają się komunikaty o błędach.
/INSTANCENAME	Określa nazwę egzemplarza.
/IACCEPTSQLSERVERLICENSETERMS	Potrzebne do zaakceptowania warunków licencji, gdy używane są parametry /Q lub /QS.
/FEATURES	Wymagany parametr określający instalowane funkcje.
/SQLSYSADMINACCOUNTS	Wymagany parametr określający konta o roli administratora systemu.
/AGTSVACCOUNT	Wymagany parametr określający konto narzędzia SQL Server Agent.
/AGTSVCPASSWORD	Wymagany parametr określający konto usługi SQL Server Agent Service.

UWAGA Kompletną listę parametrów wiersza poleceń znajdziesz na stronie [http://msdn.microsoft.com/en-us/library/ms144259\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms144259(v=sql.120).aspx).

Nienadzorowana instalacja za pomocą pliku konfiguracyjnego

Proces instalacji systemu SQL Server 2014 domyślnie tworzy plik konfiguracyjny, który rejestruje wartości opcji i parametrów podanych w trakcie instalacji. Plik ten jest przydatny przy kontrolach poprawności i inspekcjach, a przede wszystkim podczas instalowania dodatkowych egzemplarzy systemu SQL Server za pomocą tych samych ustawień.

Aby utworzyć plik konfiguracyjny, wykonaj następujące kroki.

1. Uruchom plik *Setup.exe* z nośnika instalacyjnego systemu SQL Server 2014. Uruchomiona zostanie instalacja systemu.
2. Określ opcje i parametry instalacji systemu SQL Server 2014. W trakcie używania kreatora instalacji wszystkie podane opcje i wartości zostaną zapisane w pliku konfiguracyjnym.
3. Przejdź w kreatorze instalacji do ekranu *Ready to Install*.

UWAGA Na tym etapie plik konfiguracyjny jest gotowy i wszystkie wartości opcji oraz parametrów podane na wcześniejszych ekranach są zapisane w pliku *ConfigurationFile.ini*.

4. Otwórz eksplorator plików i przejdź do katalogu, w którym utworzony został plik konfiguracyjny. Zwykle jest to podkatalog w katalogu *C:\Program Files\Microsoft SQL Server\120\Setup Bootstrap\Log*. Nazwa tego podkatalogu to data i czas utworzenia pliku konfiguracyjnego *ConfigurationFile.ini*. Kliknij przycisk *Cancel*, aby zakończyć działanie kreatora instalacji systemu SQL Server 2014.
5. Znajdź plik *ConfigurationFile.ini* i skopiuj go do katalogu, którego będziesz używał przy instalacji nienadzorowanej. Możesz np. wykorzystać katalog współużytkowany *\\fileserver\myshare*.
6. Otwórz plik *ConfigurationFile.ini*, aby go zmodyfikować. Wprowadź następujące zmiany przygotowujące plik do instalacji nienadzorowanej:
 - ustaw `QUIET = "True"`,
 - ustaw `SQLSYSADMINACCOUNTS = "YourDomain\Administrators"`,
 - ustaw `IACCEPTSQLSERVERLICENSETERMS= "True"`,
 - usuń parametr `ADDCURRENTUSERASSQLADMIN`,
 - usuń parametr `UIMODE`.

Po dostosowaniu pliku konfiguracyjnego do instalacji nienadzorowanej wywołaj w wierszu poleceń plik *Setup.exe* i podaj ścieżkę do omawianego pliku konfiguracyjnego. Składnia instrukcji powinna wyglądać tak:

```
D:\Setup.exe /ConfigurationFile=\\fileserver\myshare\ConfigurationFile.ini
```

Instalacja za pomocą skryptów powłoki PowerShell systemu Windows

Nienadzorowaną instalację możesz też przeprowadzić za pomocą powłoki PowerShell systemu Windows. Można napisać proste skrypty tej powłoki, które przeprowadzą instalację systemu SQL Server 2014 za pomocą jej wiersza poleceń. Przykładowo skrypt wiersza poleceń opisany we wcześniejszym podpunkcie można uruchomić w następujący sposób:

```
$cmd = "d:\setup.exe /ACTION=install /Q /INSTANCENAME="MSSQLSERVER"  
/IACCEPTSQLSERVERLICENSETERMS=1  
/FEATURES=SQLENGINE,SSMS  
/SQLSYSADMINACCOUNTS="TwojaDomena\Administrators"  
/AGTSVCACCOUNT="TwojaDomena\Administrators"  
/AGTSVPASSWORD="TwojeHasło";  
Invoke-Expression -command $cmd |out-null;
```


Dla rozbudowanych instalacji systemu SQL Server 2014 można napisać bardziej skomplikowane skrypty powłoki PowerShell. Często stosuje się funkcje tej powłoki przyjmujące parametry niezbędne do przeprowadzenia instalacji nienadzorowanej. Funkcje te są wykonywane wsadowo lub w procesie, który w pętli przechodzi po liście nazw serwerów powiązanych z odpowiednimi parametrami.

Przykładowo funkcję powłoki PowerShell można zapisać w pliku skryptu i wywołać z parametrami instalacji, aby przeprowadzić rozbudowaną nienadzorowaną instalację systemu SQL Server 2014. W listingu 2.1 (plik *Instalacja-Sql2014.ps1*) przedstawiono przykładową funkcję powłoki PowerShell, którą można wykorzystać do nienadzorowanej instalacji systemu SQL Server 2014.

Listing 2.1. Plik Instalacja-Sql2014.ps1

```
Function Install-Sql2014
{
    param
    (
        [Parameter(Position=0,Mandatory=$false)][string] $Path,
        [Parameter(Position=1,Mandatory=$false)][string] $InstanceName =
            "MSSQLSERVER",
        [Parameter(Position=2,Mandatory=$false)][string] $ServiceAccount,
        [Parameter(Position=3,Mandatory=$false)][string] $ServicePassword,
        [Parameter(Position=4,Mandatory=$false)][string] $SaPassword,
        [Parameter(Position=5,Mandatory=$false)][string] $LicenseKey,
        [Parameter(Position=6,Mandatory=$false)][string] $SqlCollation =
            "SQL_Latin1_General_CP1_CI_AS",
        [Parameter(Position=7,Mandatory=$false)][switch] $NoTcp,
        [Parameter(Position=8,Mandatory=$false)][switch] $NoNamedPipes
    )
    #Tworzenie instrukcji instalacji z trybem instalacji
    if ($Path -eq$null -or $Path -eq"")
    {
        #Brak ścieżki oznacza, że plik instalacyjny znajduje się w tym samym katalogu
        $command = 'setup.exe /Action="Install"'
    }
    else
    {
        #Upewnianie się, że ścieżka kończy się lewym ukośnikiem
        if (!$Path.EndsWith("\"))
        {
            $Path += "\"
        }
        $command = $path + 'setup.exe /Action="Install"'
    }
    #Akceptacja umowy licencyjnej (niezbędna przy instalacji z poziomu wiersza poleceń)
    $command += ' /IACCEPTSSQLSERVERLICENSETERMS'
    #Zastosowanie trybu QuietSimple (z paskiem postępu, ale bez interakcji)
    $command += ' /QS'
    #Ustawienie instalowanych funkcji
    $command += ' /FEATURES=SQLENGINE,CONN,BC,SSMS,ADV_SSMS'
    #Ustawienie nazwy egzemplarza
    $command += (' /INSTANCENAME="{0}"' -f $InstanceName)
    #Ustawianie klucza licencyjnego tylko wtedy, gdy go podano
    #W przeciwnym razie instalowana jest wersja próbna
    if ($LicenseKey -ne $null -and $LicenseKey -ne "")
    {

```

```

$command += ( ' /PID="{0}"' -f $LicenseKey)
}
#Sprawdzanie, czy podano konto usługowe
if ($ServiceAccount -ne $null -and $ServiceAccount -ne "")
{
#Ustawianie konta usługowego dla silnika bazodanowego
$command += ( ' /SQLSVCACCOUNT="{0}" /SQLSVCPASSWORD="{1}"'
/SQLSVCSTARTUPTYPE="Automatic" ' -f
$ServiceAccount, $ServicePassword)
#Ustawianie konta usługowego dla usługi SQL Agent Service
$command += ( ' /AGTSVCACCOUNT="{0}" /AGTSVCPASSWORD="{1}"'
/AGTSVCSTARTUPTYPE="Automatic" ' -f
$ServiceAccount, $ServicePassword)
}
else
{
#Ustawianie konta usługowego dla silnika bazodanowego na lokalne konto systemowe
$command += ' /SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"
/SQLSVCSTARTUPTYPE="Automatic" '
# Ustawianie konta usługowego dla usługi SQL Agent Service na lokalne konto systemowe
$command += ' /AGTSVCACCOUNT="NT AUTHORITY\SYSTEM"
/AGTSVCSTARTUPTYPE="Automatic" '
}
#Ustawianie serwera na tryb uwierzytelniania SQL Server, jeśli podano hasło konta SA
if ($SaPassword -ne $null -and $SaPassword -ne "")
{
$command += ( ' /SECURITYMODE="SQL" /SAPWD="{0}"' -f $SaPassword)
}
#Ustawianie bieżącego użytkownika jako administratora systemu
$command += ( ' /SQLSYSADMINACCOUNTS="{0}"' -f
[Security.Principal.WindowsIdentity]::GetCurrent().Name)
#Ustawianie kolacji bazy danych
$command += ( ' /SQLCOLLATION="{0}"' -f $SqlCollation)
#Włączanie lub wyłączanie protokołu TCP
if ($NoTcp)
{
$command += ' /TCPENABLED="0"'
}
else
{
$command += ' /TCPENABLED="1"'
}
#Włączanie lub wyłączanie obsługi potoków nazwanych
if ($NoNamedPipes)
{
$command += ' /NPENABLED="0"'
}
else
{
$command += ' /NPENABLED="1"'
}
if ($PSBoundParameters['Debug'])
{
Write-Output $command
}
else

```

```
{
Invoke-Expression $command
}
}
```

Po pobraniu kodu z listingu 2.1 z witryny wydawnictwa Helion zapisz plik w katalogu (np. *c:\skrypty*). Ponieważ jest to plik pobrany z internetu, możliwe, że będziesz musiał kliknąć go prawym przyciskiem myszy i odblokować. Następnie wykonaj poniższe kroki, aby wywołać funkcję.

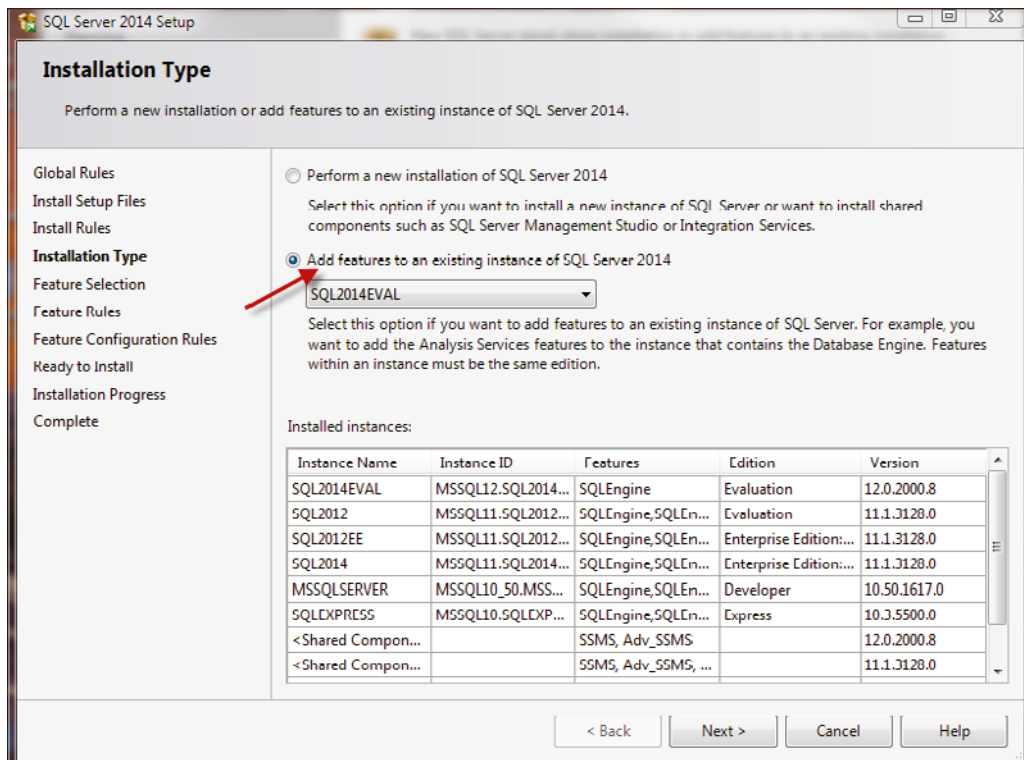
1. Uruchom wiersz poleceń powłoki PowerShell z podniesionymi uprawnieniami administratora. W tym celu kliknij plik wykonywalny powłoki prawym przyciskiem myszy i wybierz opcję *Uruchom jako administrator*. Otworzy się wiersz poleceń powłoki PowerShell.
2. Sprawdź, czy możesz uruchamiać i wczytywać niepodpisane skrypty i pliki powłoki PowerShell. W wierszu poleceń powłoki wpisz instrukcję **get-executionpolicy**, aby sprawdzić, jaka polityka wykonywania kodu jest ustawiona. Jeśli jest to polityka inna niż *RemoteSigned*, musisz zmienić ją na tę wartość za pomocą następującej instrukcji:
Set-ExecutionPolicy RemoteSigned
3. Następnie wczytaj funkcję powłoki PowerShell z pliku skryptu. W tym celu wywołaj poniższe polecenie:
. c:\scripts\Instalacja-Sql2014.ps1
Zwróć uwagę na kropkę i spację przed ścieżką do pliku skryptu. Ta kombinacja znaków jest niezbędna przy wczytywaniu pliku skryptu metodą „kropka — źródło”.
4. Upewnij się, że funkcja została wczytana. W tym celu uruchom następującą instrukcję:
get-command Install-Sql2014
Powłoka zwróci jeden wiersz z informacjami *CommandType* *Function* i *Name* *Install-Sql2014*.
5. Teraz możesz już wywołać wczytaną funkcję powłoki PowerShell. Uruchom funkcję *Install-Sql2014* w następujący sposób:
Install-Sql2014 -Param1 Param1Value -Param2 Param2Value ..
Przykładowo poniższe polecenie wywołuje funkcję *Install-Sql2014* i ustawia konto usługowe systemu SQL Server, hasło i nazwę egzemplarza oraz inicjuje instalację systemu SQL Server 2014:
Install-Sql2014 -Path d:\-ServiceAccount "winserver\Administrator"
-ServicePassword "P@ssword"
-SaPassword "P@ssword"
-InstanceName "NazwaEgzemplarza"

UWAGA Ścieżka do pliku instalacyjnego systemu SQL Server 2014 zależy od używanego nośnika.

UWAGA W witrynie codeplex.com dostępny jest rozwijany przez społeczność projekt SPADE, który automatyzuje instalowanie systemu SQL Server za pomocą powłoki PowerShell. Więcej informacji o tym projekcie znajdziesz na stronie <http://sqlspade.codeplex.com/>.

Instalowanie usług Analysis Services

Instalowanie usług Analysis Services jest proste. Możesz dodać je razem z innymi funkcjami i usługami systemu SQL Server 2014 (patrz rysunek 2.6) albo zainstalować odrębnie (patrz rysunek 2.7).



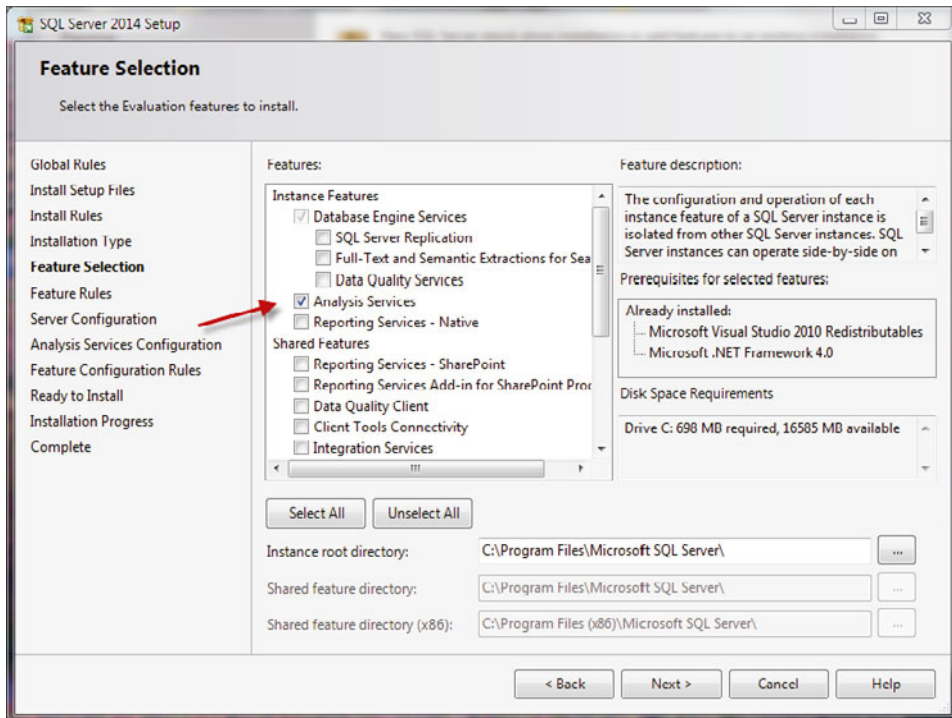
Rysunek 2.6. Dodawanie zestawu funkcji

Od wersji SQL Server 2012 usługi Analysis Services można instalować w dwóch trybach:

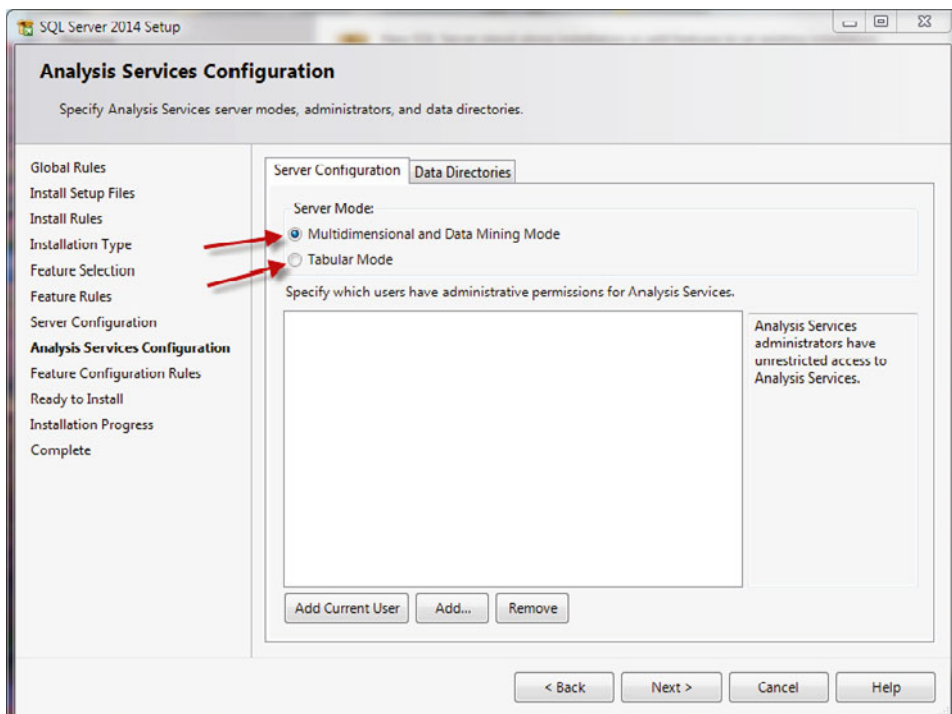
- w trybie wielowymiarowym i drążenia danych (tryb UDM),
- w trybie tabelarycznym.

Wybór trybu odbywa się na przedstawionym na rysunku 2.8 ekranie *Analysis Services Configuration* w kreatorze SQL Server 2014 Setup.

W następnych punktach pokrótce opisujemy oba tryby usług Analysis Services.



Rysunek 2.7. Instalowanie samych usług Analysis Services



Rysunek 2.8. Wybieranie trybu usług Analysis Services

Tryb UDM

Tryb UDM powoduje zainstalowanie tradycyjnego silnika usług Analysis Services dostępnego od wersji SQL Server 2005. Silnik jest oparty na modelu UDM (ang. *Unified Dimensional Model*).

W trybie UDM usługi są warstwą pośrednią między źródłami danych i obejmują wszystkie reguły biznesowe. Usługi pełnią funkcję huba w modelu wielowymiarowym, co pozwala użytkownikom błyskawicznie tworzyć kwerendy, agregować dane, drażnić je, a także szczegółowo analizować duże bazy za pomocą usług Analysis Services.

Tryb UDM usług Analysis Services obsługuje składowanie i przetwarzanie danych za pomocą technik wielowymiarowego przetwarzania analitycznego OLAP (ang. *multidimensional online analytical processing* — MOLAP), relacyjnego przetwarzania analitycznego OLAP (ang. *relational online analytical processing* — ROLAP) i hybrydowego przetwarzania analitycznego OLAP (ang. *hybrid online analytical processing* — HOLAP). Opis tych podejść znajdziesz w następnych podpunktach.

Tryb MOLAP

W trybie MOLAP dane są przechowywane i agregowane w jednej partycji lub kilku partycjach wielowymiarowej bazy danych. Ten tryb składowania danych ma maksymalizować wydajność przetwarzania kwerend. Dane w trybie MOLAP nie zawsze są aktualne.

Tryb ROLAP

W trybie ROLAP dane nie są przechowywane w bazie usług Analysis Services. Jeśli kwerendy nie można obsłużyć na podstawie danych z pamięci podręcznej, używane są wewnętrzne indeksowane widoki do pobrania informacji z relacyjnego źródła danych. Czas przetwarzania kwerend bywa znacznie dłuższy niż w modelu MOLAP. Dane są za to bardziej aktualne, ponieważ pobiera się je z systemu transakcyjnego.

Tryb HOLAP

W trybie HOLAP niektóre fragmenty wielowymiarowej bazy są przechowywane w trybie MOLAP, natomiast inne jej części są pobierane bezpośrednio z relacyjnego źródła danych. Kwerendy, w których używa się zagregowanych danych, są przetwarzane za pomocą wielowymiarowego magazynu danych, tak jak w trybie MOLAP. Kwerendy związane z drażeniem danych są obsługiwane bezpośrednio za pomocą relacyjnych źródeł danych, tak jak w trybie ROLAP.

Tryb tabelaryczny

Tabelaryczny tryb usług Analysis Services wykorzystuje nowy silnik bazodanowy Vertipaq. *Silnik Vertipaq* to oparta na kolumnach baza, która umożliwia znaczną kompresję danych, ponieważ zaawansowane algorytmy kompresji sprawiają, że nie trzeba przechowywać wszystkich odrębnych wartości.

Możliwość kompresji dużych ilości danych sprawia, że silnik Vertipaq potrafi przechowywać i pobierać dane z pamięci RAM oraz manipulować nimi znacznie szybciej niż tradycyjny silnik usług Analysis Services, oparty na danych z dysku.

Tryb tabelaryczny jest zgodny z nowym modelem semantycznym BISM (ang. *Business Intelligence Semantic Model*). Środowisko rozwojowe jest tu podobne do środowiska w dodatku PowerPivot dla Excela. Dodatek ten korzysta z tego samego silnika Vertipaq, co tryb tabelaryczny usług Analysis Services, choć działa na mniejszą skalę.

Tryb tabelaryczny, podobnie jak tradycyjny silnik UDM usług Analysis Services, obsługuje przetwarzanie kwerend za pomocą danych zapisanych w pamięci, informacji pobieranych bezpośrednio z relacyjnego źródła danych lub podejścia mieszanego. Opcje trybów przetwarzania kwerend są dostępne we właściwościach modelu BISM. Oto cztery dostępne opcje:

- *In-Memory*,
- *DirectQuery*,
- *In-Memory with DirectQuery*,
- *DirectQuery with In-Memory*.

W następnych podpunktach opisujemy te cztery tryby.

Tryb In-Memory

W tym trybie dane są gromadzone i pobierane ze zbiorów danych przechowywanych w pamięci. Opóźnienie jest tu bardzo małe lub zerowe, ponieważ koszty dyskowych operacji wejścia-wyjścia są zminimalizowane lub całkowicie wyeliminowane. Dzięki dostępowi do zbiorów danych z pamięci złożone obliczenia i operacje sortowania są wykonywane niemal natychmiast.

Tryb ten przypomina podejście MOLAP w tradycyjnym silniku usług Analysis Services, ponieważ w obu tych modelach przechowywane są dane z pewnego punktu w czasie. Jeśli dane w relacyjnym źródle się zmieniają, przechowywane w pamięci informacje trzeba będzie odświeżyć za pomocą nowych zbiorów danych.

Tryb DirectQuery

W tym trybie (można go nazwać *trybem przekazywania*) kwerendy są przetwarzane z wykorzystaniem źródła danych, którym zwykle jest relacyjna baza. Przewaga tego trybu nad trybem In-Memory silnika Vertipaq polega na możliwości udostępniania w czasie rzeczywistym dużych zbiorów danych, których nie można pomieścić w pamięci.

Tryb DirectQuery, podobnie jak podejście ROLAP w tradycyjnym silniku usług Analysis Services, jest zaprojektowany do spełniania wymagań dostępu do źródła danych w czasie rzeczywistym.

Tryb In-Memory with DirectQuery

W tym trybie — jeśli nie określono inaczej w łańcuchu znaków połączenia — kwerendy domyślnie używają zbioru danych przechowywanego w pamięci podręcznej. Ten tryb umożliwia klientowi przełączenie się na dane pobierane w czasie rzeczywistym.

Tryb DirectQuery with In-Memory

Tryb ten, w przeciwieństwie do trybu *In-Memory with DirectQuery*, domyślnie korzysta z relacyjnego źródła danych, chyba że określono inaczej w łańcuchu znaków połączenia. Ten tryb umożliwia klientowi przełączenie się na dane z pamięci podręcznej.

Instalowanie funkcji PowerPivot for SharePoint

Opcja *PowerPivot for SharePoint* na ekranie z opcjami instalacji funkcji jest dostępna od wersji SQL Server 2008 R2. Powoduje ona zainstalowanie wersji nowego silnika Vertipaq usług Analysis Services na potrzeby obsługi zadań po stronie serwera i zarządzania skoroszytami dodatku PowerPivot publikowanymi w programie SharePoint 2010 lub 2013.

Funkcję PowerPivot for SharePoint trzeba zainstalować na serwerze dołączonym do domeny, na serwerze SharePoint Enterprise SP1 lub na serwerze SharePoint 2013 RTM, a egzemplarz o nazwie PowerPivot musi być dostępny na docelowym serwerze.

Aby zainstalować funkcję PowerPivot for SharePoint, wykonaj następujące kroki.

1. Uruchom plik *Setup.exe* z nośnika instalacyjnego systemu SQL Server 2014. Otworzy się okno *Installation Center*.
2. Otwórz zakładkę *Installation* po lewej stronie, a następnie kliknij pierwszą opcję po prawej (*New SQL Server Standard Installation* lub *Add Feature to an Existing Installation*). Uruchomiony zostanie kreator *SQL Server 2014 Setup*.
3. Na ekranie *Setup Support Rules* pojawi się lista problemów, które mogą wystąpić w trakcie instalowania plików wspomagających proces instalacji. Po zakończeniu tego kroku kliknij przycisk *OK*. Rozpocznie się proces instalowania plików pomocniczych.
4. Po zakończeniu instalowania plików pomocniczych trzeba sprawdzić drugi zestaw reguł instalacji. Kliknij przycisk *OK*, aby kontynuować.
5. Na następnym ekranie, w zależności od nośnika z plikami instalacyjnymi i umowy licencyjnej, może pojawić się prośba o wybranie edycji systemu SQL Server i wprowadzenie klucza produktu. Kliknij przycisk *OK*, aby kontynuować.
6. Otworzy się ekran z umową licencyjną. Zaakceptuj jej warunki i kliknij przycisk *Next*.
7. Otworzy się ekran *Setup Role*. Wybierz opcję *PowerPivot for SharePoint*. Możesz też zainstalować usługi bazodanowe systemu SQL Server, jeśli zaznaczysz związane z nimi pole wyboru. Kliknij przycisk *Next*.
8. Pojawi się ekran *Feature Selection*. Opcje na nim są już wstępnie wybrane i wyświetlane w celach informacyjnych. Kliknij przycisk *Next*.
9. Otworzy się ekran *Installation Rules*, który określa, czy jakieś aspekty mogą spowodować problemy z instalacją. Kliknij przycisk *Next*.
10. Zobaczysz ekran *Instance Configuration*. Nie można zmienić nazwy egzemplarza, ponieważ musi to być *PowerPivot*. Możesz ustawić jedynie identyfikator egzemplarza. Kliknij przycisk *Next*.
11. Pojawi się ekran *Disk Space Requirements* z podsumowaniem ilości pamięci potrzebnej dla wybranych funkcji. Kliknij przycisk *Next*.
12. Otworzy się ekran *Server Configuration*. Należy na nim podać konto usługowe, w którym ma działać silnik usług Analysis Services. Powinno to być konto domeny. Kliknij przycisk *Next*.
13. Zobaczysz ekran *Analysis Services Configuration*. Dodaj konta domeny wymagające uprawnień administracyjnych do usług *Analysis Services*.
14. Klikaj przycisk *Next* do momentu dojścia do ekranu *Ready to Install*. Przejrzyj stronę z podsumowaniem instalacji i kliknij przycisk *Install*.

To kończy proces instalowania funkcji PowerPivot for SharePoint.

Sprawdzanie systemu

Zanim wprowadzisz system do użytku, powinieneś go sprawdzić pod obciążeniem. Zdarza się, że przez wiele miesięcy lub lat serwer działa w środowisku produkcyjnym mimo problemu sprzętowego, który występował od samego początku. Usterki mogą nie być widoczne, gdy obciążenie serwera jest niewielkie, jednak stają się oczywiste po zwiększeniu wymagań.

Istnieją bezpłatne narzędzia do przeprowadzania testów obciążeniowych serwerów bazodanowych w celu upewnienia się, że system pamięci potrafi obsłużyć oczekiwane obciążenie operacjami wejścia-wyjścia, wymagania pamięciowe i wymagania dotyczące mocy obliczeniowej procesora. Oto niektóre z tych narzędzi.

- **SQLIOSim.** Jest to bezpłatne narzędzie Microsoftu zaprojektowane w celu generowania podobnych wzorców odczytu i zapisu danych na serwerze SQL. Program doskonale nadaje się do testowania obsługi procesów z wieloma operacjami wejścia-wyjścia, takich jak DBCC CHECKDB oraz masowe wstawianie, usuwanie i aktualizowanie danych. To narzędzie możesz pobrać ze strony <http://support.microsoft.com/kb/231619>.
- **IOMeter.** To następne, doskonałe, bezpłatne narzędzie do przeprowadzania testów obciążeniowych. Może symulować obciążenie generowane przez aplikacje współbieżne.
- **Prime95.** To bezpłatne narzędzie zaprojektowano w celu wyznaczania liczb pierwszych Mersenne'a. Mocno obciąża pamięć RAM. Można je dostosować, aby zapewnić wysokie obciążenie procesora i pamięci przez określony czas.

W internecie znajdziesz też inne, bezpłatne i komercyjne aplikacje do przeprowadzania wstępnych kontroli i testów obciążeniowych. Niektórzy producenci serwerów i ich komponentów także udostępniają programy do pomiaru wydajności i przeprowadzania testów obciążeniowych oferowanych urządzeń.

Konfiguracja po instalacji

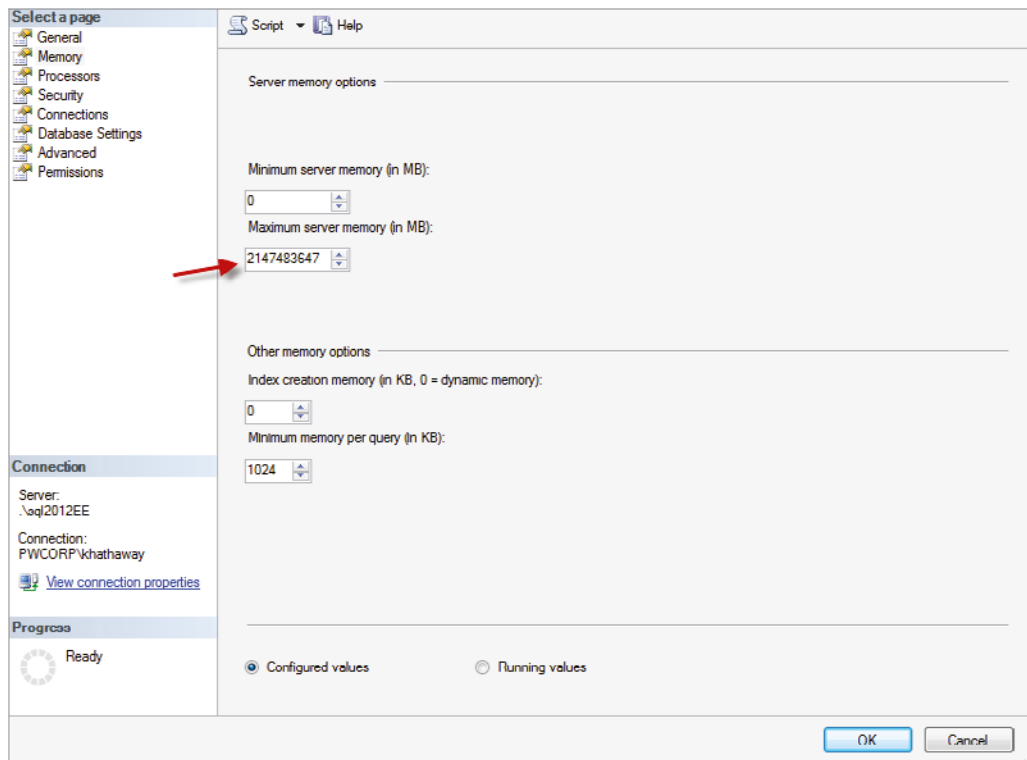
Po zainstalowaniu systemu SQL Server 2014 trzeba ustawić dodatkowe opcje i wykonać zadania potrzebne do przygotowania serwera do pracy w środowisku produkcyjnym. Niektóre z tych opcji (w tym maksymalna pamięć serwera, próg równoległych operacji i wielkość pakietów sieciowych) umożliwiają zapewnienie optymalnej wydajności danego egzemplarza systemu SQL Server. Inne ustawienia i zadania (np. zmienianie domyślnego portu, inspekcja loginów i wyłączenie konta SA) są związane z zabezpieczaniem, inspekcją i monitorowaniem egzemplarza serwera SQL Server.

Ustawianie opcji systemu SQL Server pod kątem wydajności

System SQL Server 2014 udostępnia ustawienia, które można zoptymalizować dla danego środowiska i wzorców obciążenia. Wybrane najważniejsze opcje związane z ustawieniem opisano w poniższych podpunktach.

Pamięć

Dwa ważne ustawienia serwera dotyczą maksymalnej i minimalnej ilości przydzielanej mu pamięci. Domyślnie minimalna ilość pamięci to 0 megabajtów, a maksymalna — 2 147 483 647 megabajtów (2 terabajty), tak jak na rysunku 2.9.



Rysunek 2.9. Ustawienia pamięci

Administratorzy często nie rozumieją i nie uwzględniają skutków pozostawienia wartości domyślnych dla tych dwóch ustawień. Minimalna ilość pamięci dla serwera określa ilość pamięci, która nie jest zwracana do systemu operacyjnego przez system SQL Server po przydzieleniu. Oznacza to, że system SQL Server zajmuje tę pamięć nawet wtedy, gdy nie jest potrzebna.

UWAGA Panuje powszechne, ale błędne przekonanie, że system SQL Server natychmiast po uruchomieniu zajmuje ustawioną minimalną ilość pamięci. W rzeczywistości pamięć jest alokowana tylko wtedy, gdy jest potrzebna, a jej poziom może (choć nie musi) dojść do ustawionej wartości minimalnej.

Nie trzeba zmieniać ustawienia minimalnej ilości pamięci, chyba że system operacyjny nieustannie żąda zasobów dla innych aplikacji współużytkujących tę samą pulę pamięci. Warto unikać zwracania zbyt dużej ilości pamięci do systemu operacyjnego, ponieważ może to doprowadzić do braku pamięci dla danego egzemplarza systemu SQL Server.

Maksymalna ilość pamięci określa, jak dużo pamięci może zaalokować dany egzemplarz systemu SQL Server. Zbyt wysoka wartość może spowodować, że zabraknie pamięci dla systemu operacyjnego. To ustawienie nie powinno przekraczać łącznej ilości pamięci dostępnej na serwerze. Dobrze, gdy podana wartość jest przynajmniej o 4 gigabajty mniejsza niż łączna ilość pamięci serwera.

Wielkość pakietów sieciowych

Domyślna wielkość pakietów sieciowych w systemie SQL Server to 4096 bajtów. Ustawienie wyższej wartości może poprawić wydajność egzemplarzy tego systemu, które wykonują wiele operacji masowych lub przesyłają duże ilości danych.

Jeśli rozszerzenie Jumbo Frames jest obsługiwane i włączone na serwerze oraz w infrastrukturze sieci, możesz zwiększyć wielkość pakietów sieciowych do 8192 bajtów.

Natychmiastowe inicjowanie plików

Za każdym razem, gdy plik bazy danych jest tworzony lub wymaga powiększenia, najpierw zostaje zapełniony przez system operacyjny zerami, a dopiero potem możliwy jest zapis danych. Operacja ta bywa kosztowna, ponieważ zapis jest blokowany do momentu zakończenia dodawania zer. Aby uniknąć oczekiwania na wykonanie tej operacji, można włączyć natychmiastowe inicjowanie plików. W tym celu należy dodać konto usługowe systemu SQL Server do listy użytkowników w zasadzie *Wykonuj zadania konserwacji woluminów z grupy Przypisywane praw użytkownika* w zasadach lokalnych w ustawieniach zabezpieczeń serwera.

Baza tempdb

Jedną z najważniejszych systemowych baz danych, która wymaga specjalnej uwagi i planowania, jest tempdb. W ostatnich latach do bazy tej przypisano dodatkowe zadania. W przeszłości baza tempdb była używana w procesach wewnętrznych, np. przy budowaniu indeksów i jako pamięć na zmienne tabeli, a także jako tymczasowa pamięć dla programistów. Oto lista wybranych obecnych zastosowań omawianej bazy:

- operacje wczytywania masowego uruchamiane za pomocą wyzwalaczy,
- wyrażenia CTE (ang. *Common Table Expression*),
- operacje DBCC (ang. *Database Console Command*),
- powiadomienia o zdarzeniach,
- reindeksacja, w tym operacje SORT_IN_TEMPDB, sortowanie indeksów dla partycji i operacje na indeksie w trybie online,
- zmienne i parametry typów LOB,
- operacje na wielu aktywnych zbiorach wyników,
- powiadamianie dla kwerend,
- kontrola wersji wierszy,
- zmienne tabel,
- operatory sortowania,
- operacje, które nie mieszczą się w pamięci.

Utworzenie dodatkowych plików bazy tempdb pozwala znacznie zwiększyć wydajność w środowiskach, w których baza ta jest intensywnie używana. Rozważ, w zależności od obciążenia, dodanie zestawu takich plików proporcjonalnie do liczby procesorów logicznych, aby wątki robocze systemu SQL Server mogły wykorzystywać wybrane pliki. Zwykle przyjmuje się, że stosunek liczby plików bazy tempdb do liczby procesorów logicznych powinien wynosić między 1:2 a 1:4. W skrajnych przypadkach możesz nawet utworzyć jeden taki plik na każdy procesor logiczny (stosunek 1:1). Jedyny sposób na ustalenie optymalnej liczby plików bazy tempdb polega na przeprowadzeniu testów.

Ważną kwestią jest też lokalizacja pliku bazy tempdb. Pliki należy odizolować od plików bazy danych i dziennika, aby uniknąć współzawodnictwa o operacje wejścia-wyjścia. Jeśli korzystasz z zestawu plików tempdb, pomyśl o odizolowaniu każdego z nich w odrębnej jednostce LUN (ang. *Logical Unit Number*) i na innych dyskach fizycznych.

Właściwy dobór wielkości bazy tempdb ma duży wpływ na optymalizację wydajności systemu. Pomyśl o ustawieniu początkowej wielkości na wartość inną niż domyślna, aby uniknąć kosztownego zwiększania wielkości plików. Ilość wstępnie przydzielanej pamięci zależy od oczekiwanego obciążenia i funkcji włączonych w danym egzemplarzu systemu SQL Server.

Dobłą techniką szacowania początkowej wielkości bazy tempdb jest przeanalizowanie planu wykonania kwerend uruchamianych w typowych sytuacjach. Operatory z planu wykonania kwerend (np. SORT, HASH MATCH i SP00L) zapewniają ważne informacje pomocne przy obliczaniu wymaganej ilości pamięci. Aby oszacować pamięć potrzebną dla każdego operatora, zwróć uwagę na liczbę i wielkość wierszy, które są podawane przez poszczególne operatory. W celu obliczenia ilość wymaganej pamięci pomnóż rzeczywistą (lub szacowaną) liczbę wierszy przez ich szacowaną wielkość. Choć technika nie jest precyzyjna, zapewnia dobry punkt wyjścia. Tylko doświadczenie i testy pozwolą Ci uzyskać dokładniejsze wartości.

Bazy model i bazy użytkowników

Baza model to najczęściej niedoceniana z systemowych baz danych. Stanowi szablon dla wszystkich baz użytkownika. Oznacza to, że wszystkie ustawienia bazy model są dziedziczone przez każdą nową bazę tworzoną w danym egzemplarzu systemu SQL Server.

Ustawienie początkowej wielkości, automatycznego rozrastania się i sposobu przywracania pozwala zapewnić, że wszystkie tworzone bazy użytkownika będą odpowiednio skonfigurowane pod kątem optymalnej wydajności.

Początkowa wielkość bazy powinna być na tyle duża, aby mogła obsłużyć oczekiwaną liczbę transakcji przez odpowiednio długi czas. Ważne jest, aby uniknąć nieustannego powiększania wielkości bazy. Gdy baza wymaga więcej pamięci niż jej początkowo przydzielono, można pozwolić na automatyczne zwiększanie rozmiaru, włączając ustawienie *Autogrowth*. Choć powinieneś włączyć automatyczne powiększanie bazy, pamiętaj, że ta operacja jest kosztowna i czasochłonna. Traktuj ją jak wyjście awaryjne. Gdy ta opcja jest włączona, wybierz wystarczająco duży przyrost wielkości, aby nie trzeba było często powtarzać operacji powiększania bazy.

OSTRZEŻENIE Nigdy nie włączaj opcji automatycznego pomniejszania baz ani nie planuj wykonywania takich operacji w harmonogramie. Pomniejszanie bazy może prowadzić do długich opóźnień i blokad, zajmować dużo zasobów procesora, pamięci i zasobów wejścia-wyjścia, a także zwiększać poziom fragmentacji.

Konfigurowanie ustawień systemu SQL Server w obszarze bezpieczeństwa

System SQL Server 2014 zawiera ustawienia, które umożliwiają zbudowanie lepiej kontrolowanego i bezpieczniejszego środowiska. W następnych podpunktach opisano najważniejsze ustawienia z obszaru bezpieczeństwa.

Konto SA

Konto SysAdmin (SA) w systemie SQL Server to domyślne konto systemowe z wysokimi uprawnieniami. Ponieważ to znane konto, jest obiektem wielu ataków. Aby wyeliminować zagrożenie tymi atakami, zawsze ustawiaj dla tego konta mocne hasło, które znasz tylko Ty, i nigdy nie używaj tego konta. Umieść hasło w bezpiecznym miejscu i wyłącz konto SA.

Porty TCP/IP

Do komunikowania się z klientami system SQL Server korzysta domyślnie z portu TCP/IP 1433. Natomiast do nazwanych egzemplarzy porty TCP/IP są przypisywane dynamicznie po uruchomieniu usługi. Aby zapobiec włamaniom i podczas konfigurowania zapory czasem trzeba zmienić porty domyślne i kontrolować numery portów używane do komunikowania się przez nazwane egzemplarze systemu SQL Server.

System SQL Server 2014 zawiera narzędzie SQL Server Configuration Manager (jego dokładniejsze omówienie znajdziesz dalej w tym podrozdziale), za pomocą którego można zarządzać usługami systemu SQL Server i konfiguracją sieci. Narzędzie znajduje się w katalogu *Microsoft SQL Server 2014/Configuration Tools* w menu *Start*. Na rysunku 2.10 przedstawiono okno dialogowe *TCP/IP Properties* tego narzędzia. W tym oknie możesz zmienić domyślny port 1433 na inny.

Pakiety poprawek i aktualizacje

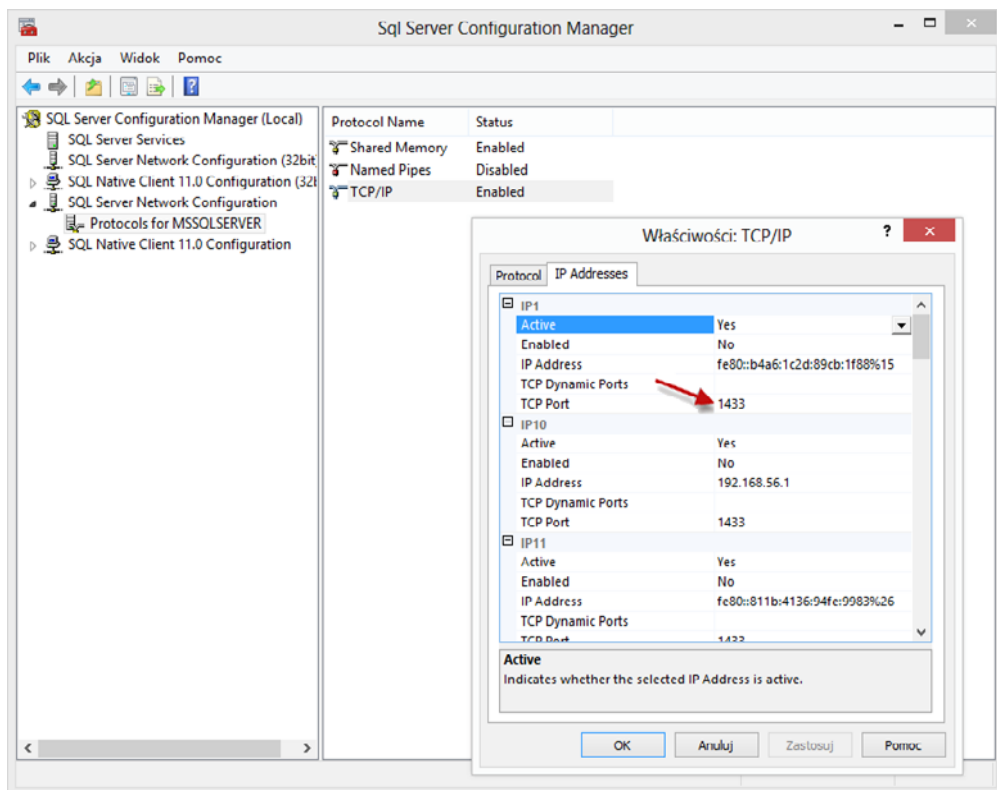
Po zainstalowaniu nowego egzemplarza systemu SQL Server należy przejrzeć dostępne aktualizacje. Są one dostępne w postaci łatek, aktualizacji zbiorczych i pakietów poprawek. Starannie przejrzyj wszystkie aktualizacje przed zastosowaniem, aby uniknąć ich negatywnego wpływu na pracę aplikacji. Koniecznie zainstaluj poprawki bezpieczeństwa oznaczone jako krytyczne, aby chronić system przed znanymi zagrożeniami, robakami i lukami. W produkcyjnych egzemplarzach systemu SQL Server nie włączaj automatycznych aktualizacji. Testuj wszystkie zmiany w kontrolowanym środowisku testowym, a dopiero potem wprowadzaj je w środowisku produkcyjnym.

Dodatkowe ustawienia systemu SQL Server

Dodatkowe ustawienia i właściwości systemu SQL Server znajdziesz w narzędziu SQL Server Management Studio oraz za pomocą systemowej procedury składowanej `sp_configure`.

UWAGA Kompletną listę (razem z opisami) wszystkich opcji konfiguracyjnych systemu SQL Server dostępnych za pomocą procedury `sp_configure` znajdziesz na stronie [http://msdn.microsoft.com/en-us/library/ms188787\(v=sql.110\).aspx](http://msdn.microsoft.com/en-us/library/ms188787(v=sql.110).aspx).

W rozdziale 4., „Zarządzanie silnikiem bazodanowym i rozwiązywanie związanych z nim problemów”, znajduje się dokładniejsze omówienie konfigurowania systemu SQL Server.



Rysunek 2.10. Ustawianie portu dla protokołu TCP/IP

Narzędzie SQL Server Configuration Manager

Narzędzie SQL Server Configuration Manager umożliwia ustawienie opcji usług systemu SQL Server oraz określenie, czy mają być uruchamiane automatycznie, czy ręcznie po rozpoczęciu pracy przez system Windows. Za pomocą tego narzędzia można konfigurować takie ustawienia usług jak konta usługowe, protokoły sieciowe i porty odbiorcze systemu SQL Server.

Aby otworzyć omawiane narzędzie, wybierz opcję *SQL Server Configuration Manager* z menu *Configuration Tools* programu Microsoft SQL Server 2014. Jest ono dostępne także w grupie *Usługi i aplikacje* w konsoli *Zarządzanie komputerem*.

Tworzenie kopii zapasowej

Zaplanowanie tworzenia kopii zapasowej powinno być jednym z zadań wykonywanych po zakończeniu instalowania systemu SQL Server. Trzeba zdefiniować harmonogram tworzenia kopii zapasowych baz systemowych oraz użytkowników i ich lokalizację. Jeśli ponadto stosujesz szyfrowanie, utwórz też kopię zapasową klucza szyfrującego.

Pliki z kopią zapasową zawsze umieszczaj na współużytkowanym napędzie sieciowym lub na maszynie archiwizacyjnej — nigdy nie zapisuj tych plików na serwerze, którego kopię zapasową tworzysz. Pomyśl o utworzeniu kilku kopii zapasowych, aby zapewnić ich bezpieczeństwo i natychmiastową dostępność w sytuacji, gdy wystąpi awaria.

Kopie zapasowe baz danych tworzy się w całości lub przyrostowo. Czasem w harmonogramie tworzenia kopii zapasowej należy też uwzględnić dziennik przywracania, aby można było odtworzyć bazę na podstawie tego dziennika. Więcej informacji na ten temat podano w rozdziale 17., „Kopie zapasowe i przywracanie stanu”.

Ustal politykę czasu przechowywania kopii zapasowych, aby uniknąć zachowywania bardzo dawnych kopii. Regularnie odtwarzaj kopie zapasowe, żeby mieć pewność, że proces kończy się powodzeniem. Dzięki temu unikniesz niespodzianek, gdy wystąpią problemy. Pamiętaj, że jakość kopii zapasowej jest zależna od jakości danych, jakie można przy jej użyciu przywrócić.

Usuwanie instalacji systemu SQL Server

Czasem trzeba całkowicie usunąć egzemplarz systemu SQL Server, np. z powodu niezgodności z nowszą wersją oprogramowania lub w celu ujednolicenia licencji. Można to zrobić za pomocą opcji *Programy i funkcje* w panelu sterowania. W trakcie usuwania instalacji można usunąć wszystkie lub wybrane funkcje danego egzemplarza systemu. Jeśli zainstalowanych jest kilka egzemplarzy, w procesie usuwania instalacji pojawi się prośba o wybranie usuwanego egzemplarza.

Dodatkowe komponenty i wymagane oprogramowanie mogą nie zostać odinstalowane. Trzeba to zrobić osobno.

Usuwanie instalacji usług Reporting Services

Gdy usuwasz instalację usługi Reporting Services, musisz ręcznie wykasować niektóre elementy. Jednak przed dezinstalacją musisz zebrać pewne informacje. Upewnij się, że wiesz, które bazy są używane przez dany egzemplarz usług Reporting Services. Informacje uzyskasz za pomocą narzędzia Reporting Services Configuration Manager. Określ, w którym katalogu zainstalowany jest dany egzemplarz usług Reporting Services. Dowiesz się tego przy użyciu narzędzia SQL Server Configuration Manager. Trzeba też ustalić, w którym katalogu znajdują się pliki z danymi o użytkowaniu i dziennikami omawianych usług.

Usunięcie usług Reporting Services nie powoduje skasowania bazy ReportServer. Trzeba to zrobić ręcznie. Jeśli tego nie zrobisz, będzie z niej korzystał nowy egzemplarz usług Reporting Services.

Usuwanie instalacji usług Analysis Services

Także usunięcie usług Analysis Services wymaga ręcznego skasowania pewnych elementów. Dlatego przed dezinstalacją należy ustalić potrzebne informacje. Określ, w którym katalogu zainstalowany jest dany egzemplarz usług Analysis Services (umożliwia to narzędzie SQL Server Configuration Manager).

Choć standardowy proces usuwa wszystkie bazy danych, nie kasuje plików dziennika omawianych usług. Domyślnie pliki te znajdują się w katalogu instalacyjnym usług lub w innej, wcześniej wybranej lokalizacji. Aby usunąć te pliki, wystarczy skasować odpowiednie katalogi.

Usuwanie instalacji silnika bazodanowego systemu SQL Server

Usunięcie instalacji silnika bazodanowego systemu SQL Server (podobnie jak innych usług) nie prowadzi do skasowania plików dziennika. Aby to zrobić, wystarczy usunąć odpowiednie katalogi. Czasem trzeba osobno usunąć narzędzie SQL Server Native Client i kilka dodatkowych katalogów.

Jeśli na komputerze zainstalowany jest tylko jeden egzemplarz systemu SQL Server, to zamiast usuwać sam katalog 120, możesz w folderze *Program Files* skasować cały katalog systemu SQL Server. Na komputerze pozostanie także platforma .NET. Jeśli chcesz ją usunąć, zrób to za pomocą opcji *Programy i funkcje* w panelu sterowania, ale najpierw się upewnij, że żadne inne aplikacje nie korzystają z tej platformy.

Rozwiązywanie problemów z nieudaną instalacją

Nieudana instalacja najczęściej ma miejsce wtedy, gdy złamane zostaną reguły związane z plikami pomocniczymi i samą instalacją. W trakcie instalacji sprawdzanych jest szereg reguł, co pozwala wykryć problemy, które mogą uniemożliwić udane zainstalowanie systemu SQL Server.

Po zidentyfikowaniu niespełnionej reguły należy rozwiązać problem, a potem przejść dalej. W trakcie instalacji nadzorowanej zawsze udostępniany jest odnośnik do raportu o niespełnionej regule i opis problemu. Ponadto generowane są pliki dziennika błędów, które można później przejrzeć.

Po wystąpieniu błędu zawsze dostępny jest szczegółowy raport. Raporty zapewniają cenne informacje, które pomagają wykryć główne przyczyny problemów. W wielu sytuacjach możesz rozwiązać usterki po zainstalowaniu brakujących funkcji lub aplikacji.

Raporty o błędach znajdziesz w katalogu `%Program Files%\Microsoft SQL Server\120\Setup Bootstrap\Log`. Każda próba instalacji powoduje wygenerowanie opatrzonego znacznikiem czasu katalogu, który zawiera szczegółowe informacje w pliku dziennika. Ten plik może pomóc w rozwiązaniu problemów.

UWAGA Kompletną listę (wraz z opisami) plików dziennika generowanych w trakcie instalacji można znaleźć na stronie [http://msdn.microsoft.com/en-us/library/ms143702\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms143702(v=sql.120).aspx).

Podsumowanie

Instalowanie systemu SQL Server 2014 jest zwykle stosunkowo łatwe i można je przeprowadzić przy niewielkich interwencjach użytkownika lub nawet bez interakcji. Opracowanie planu przed rozpoczęciem instalowania systemu jest bardzo ważne, jeśli chcesz, aby cały proces zakończył się powodzeniem. Udana instalacja systemu SQL Server 2014 rozpoczyna się od solidnego planu i dobrej definicji wymagań. W wymaganiach trzeba uwzględnić niezbędne oprogramowanie i sprzęt, prerekwizyty, uwierzytelnianie, kolację, konta usługowe, lokalizację plików itd.

Instalowanie systemu SQL Server 2014 nie kończy się z chwilą zamknięcia kreatora instalacji. Trzeba jeszcze zmodyfikować szereg domyślnych ustawień, takich jak maksymalna ilość pamięci, próg dla równoległych operacji, porty TCP/IP, obsługa poprawek itd. Do zmiany domyślnych opcji konfiguracyjnych możesz wykorzystać narzędzia SQL Server Management Studio i SQL Server Configuration Manager. Serwery bazodanowe trzeba sprawdzić i przeprowadzić testy obciążeniowe, aby uniknąć nieoczekiwanych problemów przy wysokim obciążeniu.

Jeśli planujesz aktualizację systemu, w rozdziale 3. znajdziesz dobre wskazówki na ten temat.

Najlepsze praktyki aktualizowania systemu do wersji SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Planowanie aktualizacji.
- Poznanie funkcji uznanych za przestarzałe i wycofanych.
- Wybór metody aktualizacji.
- Na co należy zwrócić uwagę po aktualizacji?

W rozdziale 2., „Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014”, opisano instalowanie SQL Server 2014 w czystym systemie. Ten rozdział dotyczy aktualizowania starszej wersji systemu SQL Server. Najlepszym sposobem na udaną aktualizację jest opracowanie planu i odpowiednie przygotowanie. Najpierw ustal powody aktualizowania systemu do wersji SQL Server 2014. Następnie określ wady i zalety różnych strategii aktualizacji oraz zapoznaj się z narzędziami, które pomagają ograniczyć zagrożenia procesu aktualizacji. Potem ustal zmiany w działaniu systemu SQL Server 2014 w porównaniu ze starszymi wersjami i wycofywane funkcje. Pod koniec rozdziału omówione zostały nieoczekiwane problemy, które mogą wystąpić po aktualizacji. Gdy skończysz czytać ten rozdział, będziesz wiedział wszystko, co jest potrzebne do udanego zaktualizowania systemu do wersji SQL Server 2014.

Po co aktualizować system do wersji SQL Server 2014?

W książce opisano znaczące usprawnienia w produkcie SQL Server. W trakcie rozwijania tego systemu skoncentrowano się na trzech głównych aspektach: bezpieczeństwie wykonywania zadań o znaczeniu strategicznym, wyciąganiu ważnych wniosków i wygodnej pracy w chmurze. Wraz z pojawieniem się wersji SQL Server 2014 Microsoft usprawnił liczne funkcje w obszarach skalowalności, niezawodności, dostępności i bezpieczeństwa. Poniżej wymieniono liczne korzyści zapewniane przez nowe funkcje i możliwości. Oto one.

- Usprawnienia w wykonywaniu zadań o znaczeniu strategicznym:
 - Funkcje In-Memory OLTP wbudowane w bazy danych przydatne przy przetwarzaniu transakcji OLTP i w hurtowniach danych,
 - Modyfikowalne działające w pamięci indeksy kolumnowe z kompresją danych archiwalnych,
 - Możliwość tworzenia puli buforów systemu SQL Server na dyskach SSD za pomocą rozszerzenia Buffer Pool Extension,
 - Opóźnione utrwalanie, co pozwala odroczyć zapis dzienników niezależnie od używanego modelu przywracania stanu,
 - Usprawnienia optymalizatora kwerend (w tym zmiany w mechanizmie szacowania liczby wierszy),
 - Szyfrowanie kopii zapasowych bez konieczności stosowania mechanizmu TDE,
 - Nowe role związane z bezpieczeństwem, zapewniające lepszy podział obowiązków,
 - Zautomatyzowane tworzenie kopii zapasowych i przesyłanie takich kopii do systemu Windows Azure,
 - Wyższa dostępność dzięki rozszerzeniu funkcji AlwaysOn na osiem baz zapasowych i udostępnieniu kreatora replikacji.
- Szybsze wyciąganie wniosków na podstawie danych:
 - Łatwy dostęp do dużych i małych zbiorów danych za pomocą funkcji Power Query, Windows Azure Marketplace i Parallel Data Warehouse (PDW) oraz technologii Polybase,
 - Szybsze i bardziej złożone wyciąganie wniosków przy użyciu Excela, dodatków PowerPivot i Power View, dodatków do drążenia danych i mobilnych aplikacji BI,
 - Kompletne rozwiązania BI obejmujące strumieniowe przesyłanie danych z Accessa w czasie zbliżonym do rzeczywistego, usługi Data Quality Service, Master Data Services i Analysis Services oraz model semantyczny BI.
- Platforma do hybrydowej pracy w chmurze:
 - Zmniejszenie kosztów CAPEX i OPEX potrzebnych do zapewnienia ciągłości pracy firmy,
 - Pełne współdziałanie systemu SQL Server z systemem Windows Azure,
 - Szybkie tworzenie działających w chmurze aplikacji o zmieniającym się obciążeniu.

Ograniczanie ryzyka — wkład Microsoftu

Podobnie jak we wszystkich wcześniejszych wersjach systemu SQL Server, zespół odpowiedzialny za ten produkt dołożył wszelkich starań, aby zapewnić jak najwyższą jakość wersji SQL Server 2014. Omawianie cyklu pracy w obszarze inżynierii oprogramowania wykracza poza zakres tej książki, jednak w tym miejscu opisujemy wybrane najważniejsze zagadnienia (zakładamy, że wiesz, czym jest proces codziennego budowania wersji).

Obecnie codziennie tworzone są *wersje* (ang. *build*) kodu systemu SQL Server 2014 dla architektur x86 i x64. Wersje te przechodzą całą baterię testów. Proces jest stosowany zarówno przy rozwijaniu nowych wydań, jak i tworzeniu pakietów poprawek dla systemu SQL Server 2014. Używane są przy tym samodzielnie opracowane testy wersji, zarejestrowane u klientów procesy pracy, a także procesy z programu Trustworthy Computing. Dział badawczy Microsoftu pracuje nad wprowadzaniem innowacji do produktów tej firmy. W obszarze rozwijania oprogramowania ten dział wnosi ważny wkład w procesy inżynierii i testów oprogramowania. Ulepsza środowisko testowe dzięki różnorodnym usprawnieniom w dziedzinach modelowania zagrożeń, wydajności testów i analizy podatności na ataki.

Ponadto w środowisku testowym wykorzystywanych jest wiele procesów pracy zarejestrowanych u klientów. Procesy te są rejestrowane w ramach programów, takich jak Customer Playback, oraz w trakcie testów laboratoryjnych (np. w ramach badań zgodności wersji SQL Server 2014 ze starszymi rozwiązaniami).

Codziennie budowane wersje są testowane na podstawie zarejestrowanych informacji tego rodzaju. Pozwala to uzyskać różne wskaźniki wydajności i bezpieczeństwa oraz wykryć usterki. Błędy są następnie rejestrowane i przypisywane do pracowników, po czym określone są priorytety usterek. Potem błędy są monitorowane do momentu ich naprawienia. Po rozwiązaniu problemów kod w ramach procesu inżynierii oprogramowania przechodzi przez testy bezpieczeństwa. Ma to miejsce przed ponownym dodaniem kodu do drzewa w celu wykonania następnego cyklu testów.

Ten rygorystyczny proces tworzenia kodu i kontroli jego jakości pomaga zagwarantować, że udostępniony produkt jest niezawodny i gotowy do pracy w środowisku produkcyjnym. Tak więc powiedzenie: „Poczekaj z aktualizacją do pojawienia się pierwszego pakietu poprawek” w przypadku systemu SQL Server 2014 nie jest prawdziwe.

Niezależni producenci oprogramowania i wkład społeczności skupionej wokół systemu SQL Server

Od wersji SQL Server 2005 aż do edycji SQL Server 2014 udostępniane są wersje CTP (ang. *Community Technology Preview*). Kilka takich pierwszych wersji to CTP1 z czerwca 2013 roku i CTP2 z września 2013 roku. Oprócz nich pojawiają się wersje RC (ang. *Release Candidate*). Decyzja o udostępnieniu wersji kodu z danego momentu doprowadziła do pobrania setek tysięcy egzemplarzy edycji CTP i RC, co zapewniło niespotykany wcześniej dostęp do zaktualizowanego kodu zarówno niezależnym producentom oprogramowania, jak i społeczności testującej system SQL Server. Dostęp do wersji beta kodu ułatwiono po to, aby można było wykryć dodatkowe błędy, przeprowadzić dodatkowe testy poprawek oprogramowania i wprowadzić nowe usprawnienia na podstawie informacji zwrotnych od społeczności.

Aktualizowanie systemu do wersji SQL Server 2014

W rozdziale 2. przedstawiono wskazówki dotyczące procesu instalacji. W tym miejscu koncentrujemy się na strategiach i uwagach związanych z aktualizowaniem systemu SQL Server 2014. Płynna aktualizacja wymaga dobrego planu. Gdy przygotowujesz plan, musisz podzielić proces aktualizacji na odrębne zadania. W planie należy wyodrębnić zestawy zadań wykonywanych przed aktualizacją, w jej trakcie i po niej.

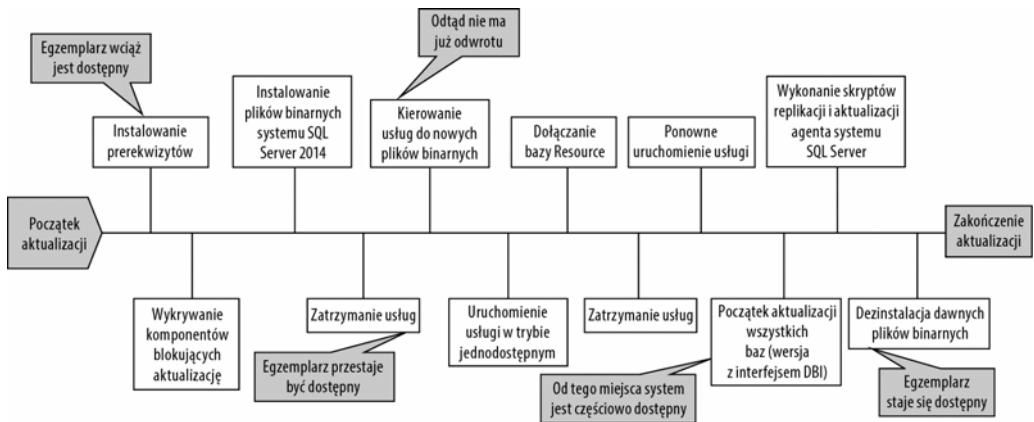
- *Zadania wykonywane przed aktualizacją* to określenie minimalnych wymagań sprzętowych i programowych dla systemu SQL Server, a także sprawdzenie, czy nowa wersja systemu jest zgodna z innymi narzędziami. Przygotuj listę aplikacji, które korzystają z serwera, wymagań dotyczących kolacji, zależności serwera (w tym łańcuchów znaków połączeń) i wymagań związanych z używanymi systemami (np. metodami dostępu do danych). Lista kontrolna powinna też obejmować sprawdzanie spójności baz danych i tworzenie kopii zapasowych wszystkich baz. Ponadto należy przygotować plan testów działania zaktualizowanej wersji i aplikacji. Ustal wszystkie problemy ze zgodnością oraz udokumentuj ich rozwiązania i poprawki. Wykorzystaj narzędzie SQL Server 2014 Upgrade Advisor (opisane dalej w tym rozdziale) do zidentyfikowania i rozwiązania takich trudności. W sieci MSDN (ang. *Microsoft Developers Network*) sprawdź funkcje, z których zrezygnowano w systemie SQL Server 2014.
- *Proces przeprowadzania aktualizacji* związany jest z płynnym przejściem przez dobrze udokumentowany i znany plan. Jeden krok jest bardzo ważny, dlatego warto go przypomnieć — przed przejściem do aktualizowania konieczne utwórz kopię zapasową wszystkich baz danych. Im większej liczby środowisk dotyczy aktualizacja, tym bardziej szczegółowy powinien być jej plan.
- *Zadania wykonywane po aktualizacji* obejmują przegląd procesu aktualizacji, ponowne udostępnienie systemów, monitorowanie ich i testowanie. Przed udostępnieniem systemu użytkownikom należy wykonać prace konserwacyjne na bazach danych. Te i inne zalecane kroki opisujemy dalej w tym rozdziale. Po aktualizacji uruchom bazę w trybie zgodności wstecz, aby zminimalizować liczbę zmian potrzebnych w środowisku. Zadbaj o zgodność baz z innymi narzędziami, jeśli używane funkcje z wcześniejszych wydań zostały uznane za przestarzałe w systemie SQL Server 2014.

W ramach planowania strategii aktualizacji masz do wyboru aktualizację w miejscu i migrację do równoległe zainstalowanej wersji.

Aktualizacja w miejscu

Aktualizacja systemu w miejscu to łatwiejsze, ale bardziej ryzykowne rozwiązanie. Jest to podejście typu wszystko albo nic, co oznacza, że po zainicjowaniu aktualizacji nie można po prostu cofnąć tego procesu. W tej technice przed rozpoczęciem prac niezbędne są solidne testy, które pozwalają uniknąć późniejszego skomplikowanego wycofywania zmian. Zaletą tej metody jest to, że nie trzeba martwić się synchronizowaniem użytkowników i loginów ani wprowadzać zmian w połączeniach aplikacji z bazami danych. Ponadto w procesie aktualizacji przenoszone są zadania narzędzia SQL Server Agent.

Na rysunku 3.1 przedstawiony jest krok po kroku ogólny scenariusz przeprowadzania aktualizacji w miejscu.



Rysunek 3.1. Aktualizacja w miejscu

1. Zainstaluj w systemie potrzebne pliki. Przed zaktualizowaniem systemu do wersji SQL Server 2014 potrzebujesz przynajmniej następujących elementów:
 - platformy .NET 3.5 z pakietem SP1,
 - platformy .NET 4.0 (dodawanej w ramach instalacji systemu SQL Server 2014),
 - powłoki Windows PowerShell 2.0,
 - egzemplarza systemu SQL Server 2005 SP4, SQL Server 2008 SP3, SQL Server 2008 R2 SP2 lub SQL Server 2012 SP1.
2. Pobierz i uruchom narzędzie SQL Server 2014 Upgrade Advisor. Analizuje ono zainstalowane komponenty starszych wersji systemu SQL Server, a następnie generuje raport identyfikujący problemy, które trzeba rozwiązać przed rozpoczęciem aktualizacji lub po jej zakończeniu.
3. Uruchom narzędzie SCC (ang. *System Configuration Checker*). Sprawdza ono docelowy komputer pod kątem elementów, które mogą uniemożliwić aktualizację (może się zdarzyć np., że nie są spełnione minimalne wymagania sprzętowe lub programowe). Jeśli narzędzie wykryje takie problemy, proces instalacji zostanie zakończony i instalator usunie komponenty systemu SQL Server 2014.
4. Po zakończeniu weryfikacji program instalacyjny systemu SQL Server może zacząć umieszczać na dysku elementy wersji 2014 i pliki pomocnicze zapewniające zgodność wstecz. Użytkownicy wciąż mogą korzystać z wersji SQL Server 2012, 2008 R2, 2008 lub 2005 SP4. Nie planuj jednak aktualizowania serwera w czasie, gdy użytkownicy mają do niego dostęp. Program instalacyjny wyłącza serwer, ponieważ zatrzymuje istniejące usługi systemu SQL Server. Następnie usługi z wersji 2014 przejmują kontrolę nad bazą master i serwerem. Na tym etapie usługi systemu SQL Server przejmują bazy danych i zaczynają je aktualizować, a użytkownicy nie mają dostępu do środowiska.
5. Na zakończenie trzeba usunąć instalację starszych plików binarnych. Ten krok należy wykonać tylko wtedy, gdy na serwerze nie ma już dawnych egzemplarzy systemu SQL Server. Na tym etapie zadania narzędzia SQL Server Agent zostały przeniesione do nowej wersji.

Oto zalety aktualizacji w miejscu.

- Jest szybka, łatwa i zautomatyzowana (najlepiej sprawdza się przy małych systemach).
- Nie jest potrzebny dodatkowy sprzęt.

- Aplikacje mogą korzystać z tej samej nazwy egzemplarza.
- Automatycznie zachowywane są funkcje wersji SQL Server 2012, 2008 R2, 2008 i 2005.

Poniżej wymienione są wady aktualizacji w miejscu.

- Następuje przestój, ponieważ w trakcie aktualizacji egzemplarz systemu SQL Server jest niedostępny.
- Nie są obsługiwane aktualizacje na poziomie komponentów.
- Niezbędna jest skomplikowana i ręczna strategia wycofywania zmian.
- Trzeba rozwiązać problemy ze zgodnością wstecz danego egzemplarza systemu SQL Server.
- Dla niektórych komponentów systemu SQL Server aktualizacja w miejscu nie jest dostępna.
- Wycofywanie zmian dla dużych baz danych zajmuje sporo czasu.

Jeśli ponadto w trakcie aktualizowania systemu chcesz zmienić jego edycję, natrafisz na pewne ograniczenia. Istnieją specjalne sposoby aktualizacji systemu SQL Server 2014 dla poszczególnych wersji i edycji. Jeśli jesteś zainteresowany tym tematem, zapoznaj się z dokumentacją SQL Server 2014 Books Online (artykuł *Supported Version and Edition Upgrades* na stronie [http://msdn.microsoft.com/en-us/library/ms143393\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms143393(v=sql.120).aspx), punkt *Upgrades from Earlier Versions to SQL Server 2014*).

Aktualizowanie przy równolegle zainstalowanych wersjach

Przy aktualizowaniu równoległym system SQL Server 2014 jest instalowany równolegle do wydania SQL Server 2012, 2008 R2, 2008 lub 2005 SP4 jako odrębny egzemplarz lub na innym serwerze. W tym podejściu system instalowany jest od nowa, po czym następuje migracja bazy danych. Z powodu czasu potrzebnego do utworzenia kopii zapasowej i jej przywracania przy wycofywaniu zmian jest to dobre rozwiązanie, gdy używane bazy danych są duże.

Przy stosowaniu tej techniki można utworzyć kopie zapasowe baz z pierwotnego serwera, a następnie odtworzyć je w egzemplarzu systemu SQL Server 2014. Inna możliwość to ręczne odłączenie baz z dawnego egzemplarza i dołączenie ich do nowego egzemplarza (z wykorzystaniem przenoszenia dzienników lub tworzenia kopii lustrzanych baz). Jeszcze inne rozwiązanie to użycie kreatora Copy Database i migracja baz danych na nowy serwer. Choć to podejście ułatwia przywracanie stanu, związane są z nim dodatkowe wymagania. Trzeba np. zmienić dawną nazwę serwera na nową, zadbać o połączenia w aplikacjach, a także zsynchronizować użytkowników i loginy.

Poniżej wymieniono argumenty na rzecz aktualizowania równoległego.

- Możliwa jest większa kontrola nad procesem aktualizowania komponentów (baz danych, usług Analysis Services itd.).
- Można uruchomić egzemplarze systemu SQL Server równolegle na potrzeby testów i sprawdzania poprawności.
- Można łatwo wycofać zmiany, ponieważ pierwotny serwer wciąż działa.
- Technika jest najlepsza dla dużych baz danych, ponieważ czas wycofywania zmian w ich przypadku może być bardzo długi.

A oto wady aktualizacji równoległej.

- Trzeba zmienić nazwy egzemplarza w aplikacjach łączących się z bazami.
- Trzeba wykupić dwie licencje na system SQL Server na jeden komputer. Obecnie konieczna jest licencja na przynajmniej jeden egzemplarz każdej używanej wersji.

Wybór między aktualizowaniem w miejscu a podejściem z równoległymi egzemplarzami

Przed wybraniem strategii należy przemyśleć szereg czynników. W strategii trzeba uwzględnić konieczność aktualizacji poszczególnych komponentów, możliwość wycofania zmian po niepowodzeniu, wielkość baz danych i potrzebę częściowej aktualizacji. To, które czynniki okażą się najważniejsze, zależy od możliwości zainwestowania w nowy sprzęt, zmodyfikowania strategii pracy środowiska (np. przeprowadzenia konsolidacji serwerów) i zaplanowania krótkiego przestoju na czas aktualizacji. Podsumowanie obu sposobów aktualizacji przedstawiono w tabeli 3.1.

Tabela 3.1. Porównanie aktualizowania w miejscu i podejścia z równoległymi egzemplarzami

Proces	Aktualizacja w miejscu	Podejście równoległe
Liczba uzyskanych egzemplarzy	Jeden	Dwa
Transfer plików danych	Automatyczny	Ręczny
Konfigurowanie egzemplarza systemu SQL Server	Automatyczne	Ręczne
Narzędzie wspomagające aktualizację	Instalator systemu SQL Server	Różne techniki migracji i transferu danych

Kroki i narzędzia potrzebne przed instalacją

Teraz, gdy już rozumiesz powody i opcje związane z aktualizowaniem systemu do wersji SQL Server 2014, możesz przejść do wyboru narzędzi, które pomogą w procesie aktualizacji i wykonaniu zadań niezbędnych przed jej rozpoczęciem. Najpierw należy podjąć kroki zapobiegawcze, aby uniknąć często występujących problemów (takich jak brak miejsca na dysku lub uruchomienie rozruchowych procedur składowanych w trakcie aktualizacji). Istnieje wiele narzędzi, które pomagają wykryć potencjalne problemy w środowisku. Dwa najczęściej stosowane narzędzia wspomagające proces aktualizacji to SQL Server Upgrade Advisor i SQL Server Upgrade Assistant. Oba narzędzia przeprowadzają analizę sytuacji przed aktualizacją i pozwalają się upewnić, że proces zakończy się powodzeniem. Upgrade Assistant wykonuje testy obciążeniowe, aby sprawdzić działanie aplikacji po aktualizacji. Upgrade Advisor przeprowadza analizę bazy danych pod kątem problemów ze zgodnością.

Zadania wykonywane przed aktualizacją

Przed przejściem do procesu aktualizacji należy wykonać szereg zadań. Te zapobiegawcze operacje pomagają wyeliminować niespodzianki w trakcie aktualizacji.

- Ustaw pliki danych i dzienników tak, aby mogły się automatycznie rozrastać w trakcie aktualizacji.
- Wyłącz wszystkie rozruchowe procedury składowane, ponieważ proces aktualizacji zatrzymuje i uruchamia usługi aktualizowanego egzemplarza systemu SQL Server.
- Zapewnij wystarczającą ilość pamięci bazie tempdb, aby mogła się rozrastać w trakcie aktualizacji. Jest to główna baza danych używana do zarządzania obiektami tymczasowymi, kontrolowania wersji wierszy i odbudowywania indeksów.

- Przed aktualizacją do wersji SQL Server 2014 wyłącz wszystkie opcje śledzenia. Możliwe, że w tej wersji dane opcje działają inaczej niż w starszej lub w ogóle nie są dostępne. Po zakończeniu aktualizacji sprawdź, które z opcji (jeśli w ogóle) nadal są potrzebne.
- Przeprowadź migrację do narzędzia Database Mail. Z programu SQL Mail zrezygnowano w wersji SQL Server 2012.

Narzędzia przydatne przed aktualizacją

Aktualizowanie systemu bywa trudne i stresujące. Aby ograniczyć ryzyko nieudanej aktualizacji lub nieoczekiwanych sytuacji po jej zakończeniu, sprawdź używane egzemplarze. Dwa narzędzia przydatne w przygotowaniach do aktualizacji to SQL Server Upgrade Advisor (SSUA) i Upgrade Assistant for SQL Server 2014 (UAFS).

SQL Server Upgrade Advisor

Reguły sprawdzane przez to narzędzie dotyczą warunków, sytuacji i znanych błędów, które mogą wpływać na aktualizację systemu do wersji SQL Server 2014. Jeśli chcesz wykorzystać doświadczenia innych użytkowników tego systemu, wybierz SQL Server 2014 Upgrade Advisor.

Narzędzie jest oparte na informacjach zwrotnych od pierwszych użytkowników i danych z wewnętrznych testów laboratoryjnych. Jest ono bezpłatne i należy do pakietu Microsoft SQL Server 2014 Feature Pack oraz znajduje się na nośnikach instalacyjnych wszystkich edycji systemu SQL Server 2014.

Identyfikuje znane problemy związane z aktualizacją oraz sugeruje rozwiązania wykrytych trudności z poszczególnymi komponentami serwera. Microsoft włożył dużo pracy w przygotowanie tego narzędzia, aby ograniczyć zagrożenia i ułatwić użytkownikom systemów SQL Server 2005, SQL Server 2008 i SQL Server 2012 aktualizację do wersji SQL Server 2014. Dlatego niezależnie od tego, czy używasz komponentów Analysis Services, Integration Services, Reporting Services lub dowolnej ich kombinacji, Upgrade Advisor się przyda.

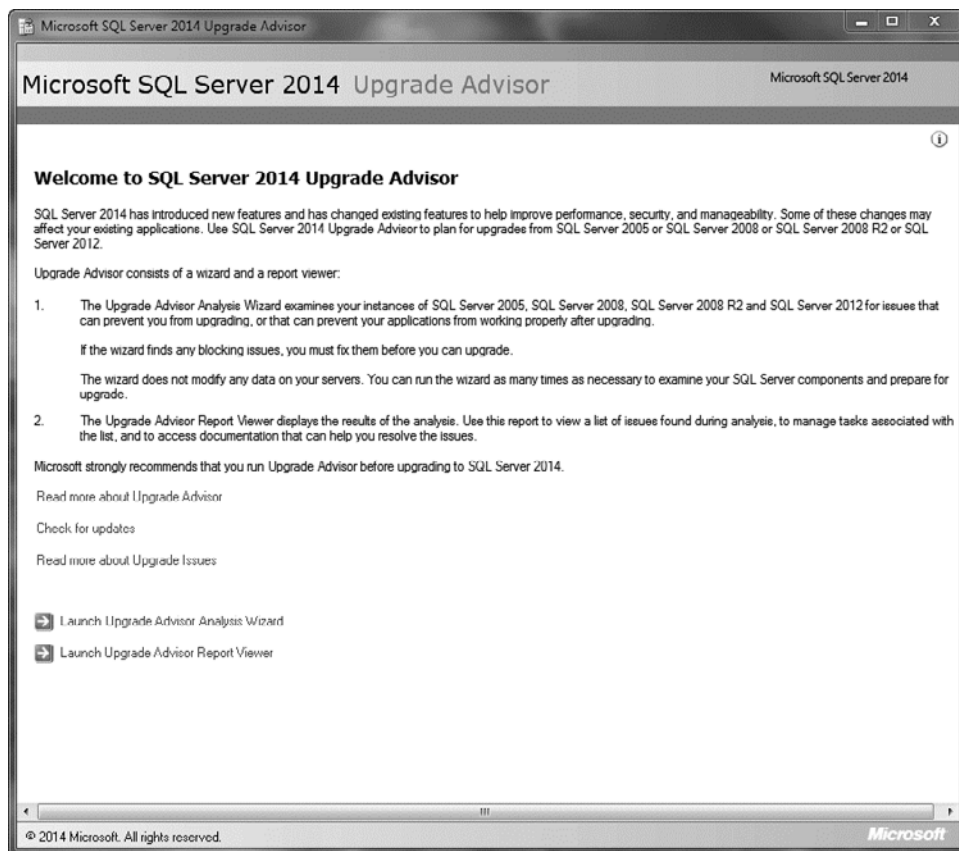
Instalowanie narzędzia SQL Server 2014 Upgrade Advisor

Instalowanie i używanie narzędzia Upgrade Advisor jest stosunkowo łatwe. Opcję służącą do zainstalowania go znajdziesz w sekcji *Planning* na domyślnym ekranie instalatora. Narzędzie to można ponadto pobrać z witryny Microsoftu.

Po zainstalowaniu narzędzia przejdź do niego w menu *Programy*. Po uruchomieniu programu pojawi się ekran powitalny przedstawiony na rysunku 3.2. Wybierz opcję *Check for Updates*, ponieważ w internecie dostępne mogą być zaktualizowane wersje narzędzia.

Omawiane narzędzie jest nieustannie ulepszane na podstawie doświadczeń administratorów baz danych, którzy wcześniej zdecydowali się na aktualizację. Narzędzie wymaga do działania platformy .NET 4.0, którą można pobrać za pomocą usługi Windows Update lub z sieci MSDN. Możesz też zainstalować jeden egzemplarz określonej wersji omawianego narzędzia, aby testować różne serwery w firmie. To podejście pozwala badać serwery w trybie tylko do odczytu bez negatywnego wpływu na ich pracę.

UWAGA Opisywane narzędzie wykonuje dużo operacji odczytu, dlatego należy je sprawdzić na serwerze testowym, aby ocenić potencjalny wpływ na używane systemy.



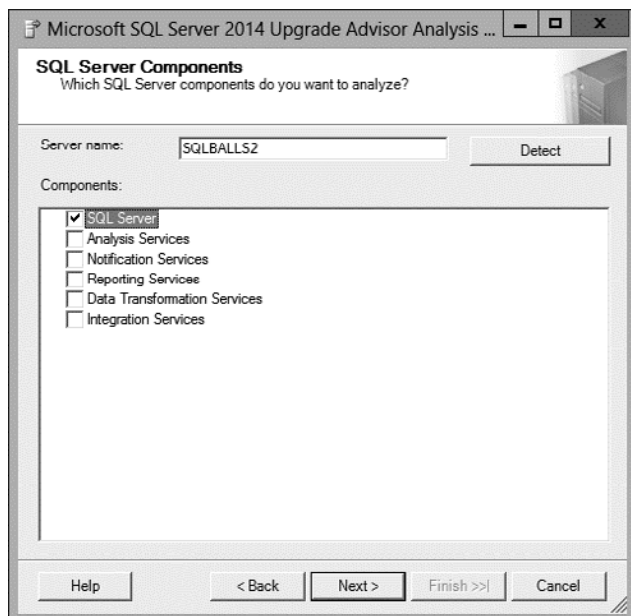
Rysunek 3.2. Ekran powitalny narzędzia Upgrade Advisor

Sam proces instalacji jest prosty. Możesz jedynie wskazać katalog, w którym chcesz zainstalować narzędzie. Domyślnie jest ono umieszczane w katalogu `C:\Program Files (x86)\Microsoft SQL Server Upgrade Advisor`.

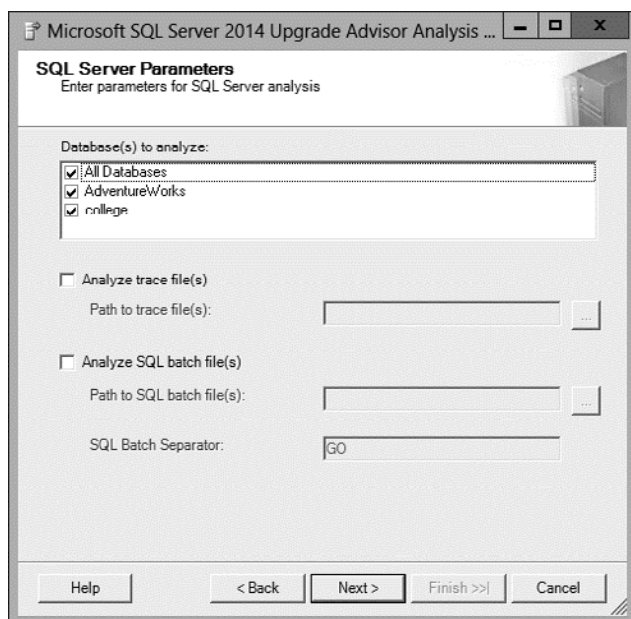
Używanie narzędzia Upgrade Advisor

Po zainstalowaniu narzędzie Upgrade Advisor wyświetla dwie opcje: *Upgrade Advisor Analysis Wizard* i *Upgrade Advisor Report Viewer*. Wybierz opcję *Upgrade Advisor Analysis Wizard*, aby uruchomić narzędzie. Na rysunku 3.3 widać, że należy wskazać serwer i komponenty, które program ma przeanalizować pod kątem aktualizacji. Możesz też kliknąć przycisk *Detect*, który uruchamia proces inspekcji prowadzący do wybrania komponentów zainstalowanych w systemie.

Po wybraniu testowanych komponentów należy określić sprawdzane bazy danych, co pokazano na rysunku 3.4. Najlepsze jest to, że można przeanalizować plik śladu wygenerowany przez narzędzie SQL Profiler i pliki wsadowe systemu w SQL-u, co pomaga przeprowadzić wyczerpujące badania. Jeśli dodasz takie pliki do procesu analizy, Upgrade Advisor nie tylko zbada bazę danych, ale też ślad i skrypty w SQL-u. Przeprowadzając analizę takich dodatkowych informacji, Upgrade Advisor sprawdzi zarówno bazę danych w jej obecnej postaci, jak i dane na temat jej użytkowania oraz działania dostępne w plikach śladu i plikach wsadowych. Użytkownik musi tylko wskazać ścieżkę do katalogu z plikami śladu lub plikami wsadowymi.



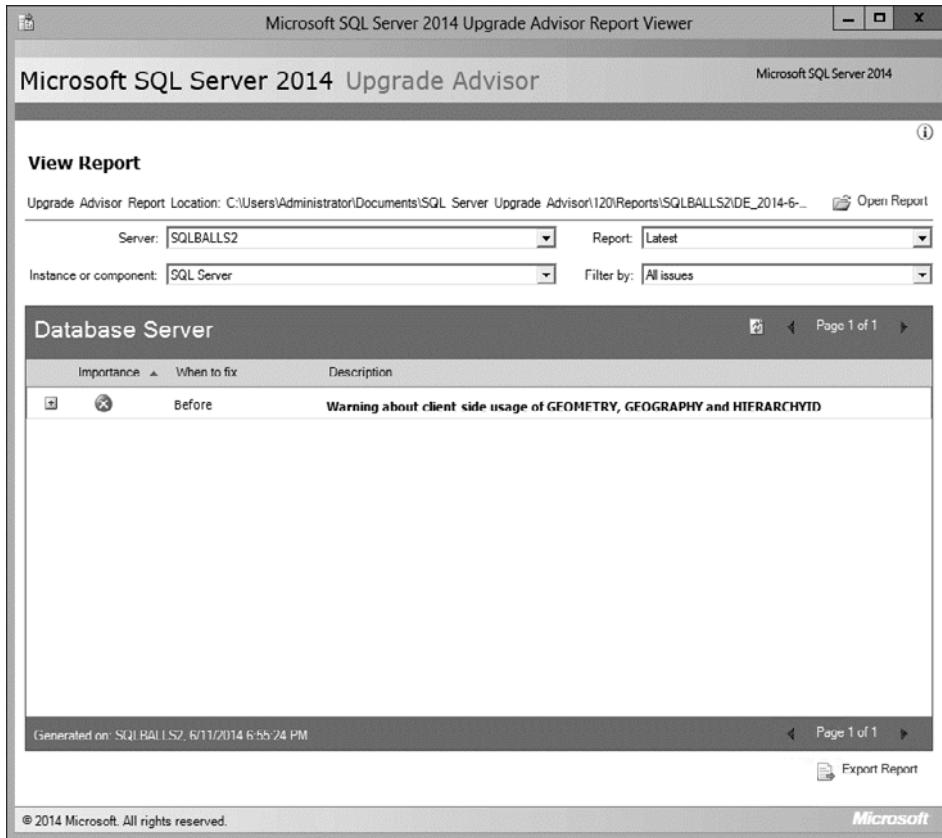
Rysunek 3.3. Ustawianie analizowanych komponentów



Rysunek 3.4. Wybieranie analizowanych baz danych

Po zakończeniu ustawiania sprawdzanych komponentów pojawia się pytanie o to, czy uruchomić proces analiz. Jeśli w trakcie konfiguracji będziesz miał jakieś pytania, użyj przycisku *Help*, który wyświetla dokumentację Book Online na temat narzędzia Upgrade Advisor. Zawiera ona bogate informacje i pomaga zrozumieć dostępne opcje. Po zakończeniu analiz poszczególnych komponentów zobaczysz zielone, żółte lub czerwone okna dialogowe z informacjami o wynikach testów.

Po zakończeniu testów możesz wyświetlić wykryte problemy za pomocą narzędzia Upgrade Advisor Report Viewer. Na rysunku 3.5 pokazano, że raporty mają postać podobną do stron w przeglądarce internetowej. Aby przeanalizować informacje, przefiltruj raport na podstawie serwera, egzemplarza, komponentu lub typu problemu. Dalej w tym rozdziale dowiesz się, jak interpretować uzyskane wyniki.



Rysunek 3.5. Raport na temat wykrytych problemów

Uruchamianie narzędzia Upgrade Advisor za pomocą skryptów

Jeśli używasz farmy serwerów lub wolisz stosować skrypty, możesz wykorzystać narzędzie uruchamiane z poziomu wiersza poleceń. Za pomocą programu *UpgradeAdvisorWizardCmd* możesz skonfigurować narzędzie przy użyciu pliku konfiguracyjnego w formacie XML i otrzymać wyniki w plikach w tym samym formacie. Program *UpgradeAdvisorWizardCmd* przyjmuje następujące parametry:

- parametr wyświetlający pomoc wiersza poleceń,
- ścieżkę i nazwę pliku z konfiguracją,
- login i hasło do systemu SQL Server (jeśli używane jest uwierzytelnianie SQL Server zamiast uwierzytelniania systemu Windows),
- opcjonalną opcję określającą, czy raport ma mieć format CSV (ang. *Comma-Separated Values*).

W pliku konfiguracyjnym można ustawić wszystkie właściwości i parametry opisane w podpunkcie poświęconym kreatorowi. Wyniki otrzymane za pomocą narzędzia uruchomionego w wierszu poleceń można wyświetlić w oknie *Report Viewer* (gdy generowane są dokumenty w formacie XML) lub w Excelu (podczas używania plików CSV). Poniższy przykładowy dokument w formacie XML wygenerowany przez narzędzie Upgrade Advisor odzwierciedla wyniki analizy wszystkich baz danych, usług Analysis Services i pakietów SSIS z egzemplarza SQL2014 z serwera SQL14Demo:

```
<Configuration>
  <Server>SQL14Demo</Server>
  <Instance>SQL2014</Instance>
  <Components>
    <SQLServer>
      <Databases>
        <Database>*</Database>
      </Databases>
    </SQLServer>
  </Components>
</Configuration>
```

Plik ten możesz zmodyfikować w edytorze XML-a (np. w Visual Studio) i zapisać z nową nazwą. Następnie nowy plik może posłużyć za dane wyjściowe dla narzędzia Upgrade Advisor wywoływanego z wiersza poleceń. Przykładowo poniższy fragment kodu to instrukcja potrzebna do uruchomienia wiersza poleceń narzędzia Upgrade Advisor z uwierzytelnianiem systemu Windows. Użyty tu plik konfiguracyjny zawiera nazwy zdalnego serwera SQL2014 i egzemplarza o nazwie SQL2014. Ścieżka do narzędzia Upgrade Advisor musi być zapisana w zmiennej środowiskowej PATH.

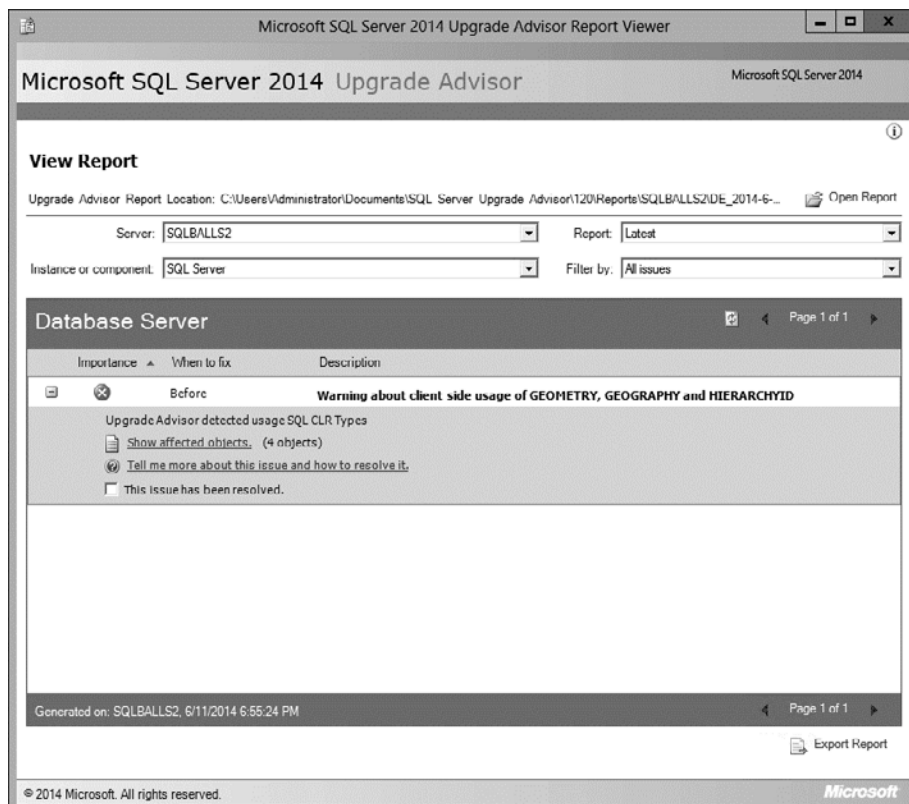
```
C:\>UpgradeAdvisorWizardCmd -ConfigFile "SQL2014Config.xml"
```

W wierszu poleceń możesz też zainstalować lub usunąć aplikację Upgrade Advisor. Proces instalacji można przeprowadzić za pomocą interfejsu użytkownika lub bez niego. Ponadto można skonfigurować ścieżkę instalacji i opcje rejestrowania zdarzeń w dzienniku.

UWAGA Więcej informacji na temat plików konfiguracyjnych narzędzia Upgrade Advisor znajdziesz w punkcie „UpgradeAdvisorWizardCmd Utility” systemu pomocy dla tej aplikacji.

Rozwiązywanie problemów z aktualizacją

Raport narzędzia Upgrade Advisor zawiera bogate informacje. Ważne jest, aby wiedzieć, jaką mają postać, czym trzeba się zająć i w jaki sposób to zrobić. Na rysunku 3.6 pokazano, że pierwsza kolumna (*Importance*) określa wagę odkrycia lub rekomendacji, druga (*When to Fix*) informuje, kiedy należy zająć się daną kwestią, a kolumna *Description* obejmuje opis problemu. Zaczynaj analizy od podzielenia informacji na kategorie według kolumn *Importance* (czyli znaczenie) i *When to Fix* (czyli kiedy naprawić). Od kombinacji wartości z tych kolumn zależy, czy problem należy rozwiązać przed aktualizacją czy po jej zakończeniu. W tabeli 3.2 zawarto rekomendacje z tego obszaru.



Rysunek 3.6. Dodatkowe informacje na temat problemów z aktualizacją

Tabela 3.2. Kiedy należy rozwiązać problemy związane z aktualizacją?

Znaczenie	Kiedy naprawić?	Rekomendacja
Czerwony	Przed	Rozwiąż przed aktualizacją
Czerwony	W dowolnym czasie	Rozwiąż przed aktualizacją
Czerwony	Po	Rozwiąż po aktualizacji
Żółty	W dowolnym czasie	Rozwiąż po aktualizacji
Żółty	Po	Rozwiąż po aktualizacji
Żółty	Wskazówka	Rozwiąż po aktualizacji

Problemy oznaczone na czerwono w kolumnie *Importance* i o wartościach *Before* (przed) lub *Any Time* (w dowolnym czasie) w kolumnie *When to Fix* należy rozwiązać przed procesem aktualizacji. Zwykle są one związane ze zmianami działania funkcji w systemie SQL Server 2014 (np. z wycofaniem pewnych mechanizmów). Pozostałe problemy przeważnie można rozwiązać po zakończeniu aktualizacji, ponieważ albo naprawiają powstałe w jej trakcie usterki, albo w ogóle jej nie dotyczą. Jeśli rozwiniesz wpis dotyczący danego błędu, zobaczysz dodatkowe informacje, co pokazano na rysunku 3.6.

Odnosnik *Show affected objects* wyświetla obiekty oznaczone przez narzędzie Upgrade Advisor jako powiązane z danym problemem, a za pomocą odnośnika *Tell me more about this issue and how to resolve it* możesz przejść do odpowiedniego fragmentu dokumentacji UABOL (ang. *Upgrade Advisor Books Online*). Dokumentacja UABOL opisuje warunki wystąpienia błędów i zawiera wskazówki z opisem rozwiązań problemów. W tej dokumentacji znajdziesz także porady wykraczające poza zakres działania omawianego narzędzia (np. związane z replikacją, usługą SQL Server Agent lub wyszukiwaniem pełnotekstowym).

Znaczek w polu *This issue has been resolved* pozwala zobaczyć, które trudności zostały już rozwiązane. Znaczek ułatwia też proces naprawiania problemów, ponieważ umożliwia przefiltrowanie raportu i wyświetlenie np. tylko wykonanych zadań lub nierozwiązanych trudności.

Jeśli wolisz stosować skrypty w wierszu poleceń, pamiętaj, że opisana tu przeglądarka tylko wyświetla za pomocą transformacji XSLT plik z wynikami w formacie XML zapisany w katalogu *My Documents\SQL Server 2014 Upgrade Advisor Reports*. Znajdziesz tam wyniki dla poszczególnych komponentów, a w katalogach o nazwach odpowiadających nazwom serwera umieszczone są pliki konfiguracyjne. Raporty z przeglądarki możesz też wyeksportować do innych formatów (np. jako pliki CSV lub pliki tekstowe).

Zapisywanie środowiska

Powinieneś przygotować punkt początkowy przez utworzenie kopii zapasowych wszystkich systemów SQL Server 2008 (lub 2005) i baz danych użytkowników z używanego serwera. Następnie należy rozpocząć rejestrowanie pliku śladu, aby uniknąć luk w danych. Plik śladu musi dobrze odzwierciedlać procesy pracy charakterystyczne dla środowiska. Dlatego czasem trzeba wygenerować sztuczne procesy pracy, które lepiej reprezentują typowe działanie aplikacji. W trakcie rejestrowania pliku śladu warto unikać operacji z udziałem wielu serwerów, przykładowo wywołań do połączonych serwerów lub kopiowania masowego z wykorzystaniem jednostek zależnych. Pamiętaj, że ze śledzeniem związane są pewne koszty. Pliki śladu w połączeniu z kopiami zapasowymi baz tworzą powtarzalny proces pracy wielokrotnego użytku (ang. *playback*).

Przy migracji danych z systemu SQL Server 2012 do wersji SQL Server 2014 można zastosować narzędzie DRU (ang. *Distributed Replay Utility*) do oceny skutków aktualizacji systemu. Narzędzie to wprowadzono w wersji SQL Server 2012, aby zastąpić uznaną za przestarzałą funkcję odtwarzania śladu z komponentu SQL Profiler. Narzędzie DRU wzbogaca możliwości programu profilującego i może odtwarzać procesy pracy z wielu komputerów, co lepiej odzwierciedla wykonywanie zadań o znaczeniu strategicznym. Możesz zastosować funkcję zapisywania rozszerzonych informacji o zdarzeniach, aby rejestrować także zdarzenia `sqlserver.sql_batch_completed` i `sqlserver.sql_statement_completed` oraz akcję `sqlserver.sql_text`, a potem wykorzystać je w DRU.

Narzędzie DRU działa w dwóch trybach.

- *Tryb synchronizacji* powoduje, że procesy pracy zarejestrowane w śladzie są odtwarzane w pierwotnej postaci — z przerwami między poleceniami itd.
- *Tryb obciążeniowy* tylko odtwarza procesy pracy i generuje obciążenie docelowego serwera.

Oto kilka ważnych zagadnień związanych z narzędziem DRU.

- **Narzędzie administracyjne dla DRU.** Do komunikowania się z kontrolerem rozproszonego odtwarzania służy aplikacja konsolowa *DReplay.exe*. Za pomocą tego narzędzia administracyjnego możesz sterować rozproszonym odtwarzaniem.

- **Kontroler narzędzia DRU.** Jest to komputer, na którym działa usługa systemu Windows o nazwie *SQL Server Distributed Replay controller*. Kontroler narzędzia DRU koordynuje działanie klientów biorących udział w rozproszonym odtwarzaniu. W każdym środowisku, w którym działa DRU, może funkcjonować tylko jeden kontroler.
- **Klienci narzędzia DRU.** Jest to jeden lub kilka komputerów (fizycznych albo wirtualnych), na których działa usługa systemu Windows o nazwie *SQL Server Distributed Replay client*. Te klienci współdzielają ze sobą i symulują procesy pracy na danym egzemplarzu systemu SQL Server. W każdym środowisku, w którym działa DRU, może pracować do 16 klientów po 255 wątków każdy.
- **Docelowy serwer.** Jest to egzemplarz systemu SQL Server, którego klienci narzędzia DRU używają przy odtwarzaniu danych ze śladu. Zaleca się, aby docelowy serwer pracował w środowisku testowym.

Przygotowywanie serwera używanego jako punkt odniesienia

Po zarejestrowaniu odtwarzanych operacji za pomocą narzędzia SQL Profiler lub mechanizmu rozszerzonych zdarzeń możesz przygotować system używany jako punkt odniesienia do dalszych testów. Serwer ten powinien mieć zainstalowany system SQL Server 2008 R2 SP1, 2008 SP2 lub 2005 SP2 i spełniać minimalne wymagania potrzebne do zaktualizowania systemu do wersji 2014. W systemach źródłowym i docelowym należy użyć tej samej kolacji i tych samych poprawek.

Następnie narzędzie sprawdzi, czy używane serwery do siebie pasują. Jeśli będzie trzeba, pojawi się prośba o zainstalowanie poprawek lub odtworzenie bazy master. Następnie narzędzie odtwarza bazy danych w odpowiedniej kolejności, tak aby ich identyfikatory pasowały do identyfikatorów z systemu produkcyjnego. Czasem wymaga to także utworzenia dodatkowych pustych baz. Potem narzędzie SSUA odtwarza loginy i sprawia, że wygenerowane identyfikatory pasują do systemu produkcyjnego (jest to niezbędne do uruchomienia pliku śladu). Następny krok polega na uruchomieniu narzędzia Upgrade Advisor, co opisano wcześniej. Po przygotowaniu środowiska można przejść do następnych zadań, aby odtworzyć ślad.

Uruchamianie narzędzia DRU

Przy uruchamianiu śladu najpierw dla wszystkich baz danych aktualizowane są statystyki. Następnie narzędzie odtwarzające korzysta z interfejsu API i uruchamia po kolei wszystkie kwerendy z pliku śladu. Narzędzie działa w trybie jednowątkowym i może nastąpić blokowanie operacji. Jeśli ślad jest odtwarzany wolno lub się zatrzymuje, warto sprawdzić, czy nie nastąpiła blokada systemu SQL Server. Jeśli sytuacja sama się nie rozwiąże, konieczne może okazać się zamknięcie blokujących procesów. Dane wyjściowe z tego kroku to plik z danymi wygenerowanymi na podstawie śladu, używany do porównań w końcowych analizach.

Aktualizowanie systemu do wersji SQL Server 2014

Teraz możesz przejść do aktualizowania systemu do wersji SQL Server 2014. Dostępne są dwa podejścia. Możesz użyć narzędzia SSUA i odtworzyć stan systemu SQL Server 2005, 2008, 2008 R2 lub 2012 do stanu będącego punktem odniesienia, a następnie zaktualizować system w miejscu do wersji SQL Server 2014, lub przenieść dane do istniejącego egzemplarza systemu SQL Server 2014.

Wcześniej w rozdziale wyjaśniono, że wybór między aktualizacją w miejscu a migracją do równoległego systemu zależy od wielu czynników specyficznych dla używanego środowiska. Na tym etapie nie mierzysz wydajności, dlatego używane serwery nie muszą być identyczne. Mierzone jest jednak działanie dwóch wersji systemu SQL Server dla tych samych procesów pracy.

Po odtworzeniu punktu odniesienia w systemie SQL Server 2014 ponownie uruchom narzędzie DRU, ale tym razem dla wersji 2014. Dane wyjściowe z tego kroku to plik z danymi wygenerowanymi na podstawie śladu, używany do porównań w końcowych analizach.

Końcowe analizy

Po wykonaniu wszystkich opisanych zadań dochodzisz do końcowego etapu. Polega on na porównaniu plików wyjściowych wygenerowanych na podstawie śladu i znalezieniu w nich rozbieżności. Narzędzie Report Viewer wyświetla po jednym warunku błędu i pokazuje ostatni poprawny krok, krok prowadzący do problemów i następną poprawną sekwencję z plików wsadowych. Po wykryciu warunku błędu można go odfiltrować z procesu sprawdzania błędów, aby umożliwić administratorowi bazy danych zidentyfikowanie nowych warunków błędu. Po zakończeniu aktualizacji do systemu SQL Server 2014 zmień tryb zgodności baz danych na 120 i uruchom aplikację, której chcesz używać, aby sprawdzić, czy działa w trybie zgodności w tym systemie. To pozwala się upewnić, że aplikacja dla baz danych z ustawionym poziomem zgodności 120 będzie działała tak samo jak wcześniej.

Zgodność wstecz

W tym podrozdziale opisujemy najważniejsze zmiany w systemie SQL Server 2014 podzielone na trzy kategorie: nieobsługiwane funkcje, wycofane funkcje i rozwiązania wpływające na działanie starszych wersji systemu SQL Server. Choć narzędzie Upgrade Advisor wyświetla wykryte niezgodności, jeśli występują w sprawdzanym środowisku, warto zapoznać się z tym podrozdziałem i z wprowadzonymi zmianami.

Funkcje nieobsługiwane lub wycofane w wersji SQL Server 2014

Aby technologia mogła się rozwijać, od czasu do czasu trzeba z czegoś zrezygnować. W tabeli 3.3 pokazano wybrane funkcje systemu SQL Server 2008 wycofane w wersji SQL Server 2014.

Tabela 3.3. Wycofane funkcje

Kategoria	Wycofana funkcja	Zastępnik	Nazwa funkcji
Tworzenie i przywracanie kopii zapasowych	RESTORE { DATABASE LOG } WITH [MEDIA]PASSWORD jest uważana za przestarzałą, a BACKUP { DATABASE LOG } WITH PASSWORD i BACKUP { DATABASE LOG } WITH MEDIAPASSWORD zostały wycofane	Brak	BACKUP DATABASE lub LOG WITH PASSWORD BACKUP DATABASE lub LOG WITH MEDIAPASSWORD

Tabela 3.3. Wycofane funkcje — ciąg dalszy

Kategoria	Wycofana funkcja	Zastępnik	Nazwa funkcji
Poziomy zgodności	Aktualizacja z poziomu 90 (reprezentującego wersję SQL Server 2005 Database Engine)	Poziomy zgodności są dostępne tylko dla dwóch ostatnich wersji. W SQL Server 2014 można zaktualizować bazę z wersji SQL Server 2005, ale w trakcie aktualizacji poziom zgodności zostanie zmieniony z 90 na 100	Poziom zgodności baz danych równy 90
Obiekty bazy danych	Możliwość zwracania zbiorów wyników z wyzwalaczy	Brak	Zwracanie wyników z wyzwalaczy
Szyfrowanie	Szyfrowanie za pomocą algorytmów RC4 i RC4_128 zostało uznane za przestarzałe; te opcje mają zostać usunięte w następnej wersji. Nie dotyczy to deszyfrowania za pomocą tych algorytmów	Należy zastosować inny algorytm szyfrowania (np. AES)	Przestarzały algorytm szyfrowania
Zdalne serwery	sp_addremotelogin	Należy zastąpić zdalne serwery serwerami połączonymi; funkcję sp_addserver można stosować tylko z opcją local	sp_addremotelogin
	sp_addserver		sp_addserver
	sp_dropremotelogin		sp_dropremotelogin
	sp_helpremotelogin		sp_helpremotelogin
	sp_remotoption		sp_remotoption
Zdalne serwery	@@remserver	Należy zastąpić zdalne serwery serwerami połączonymi	Brak

Tabela 3.3. Wycofane funkcje — ciąg dalszy

Kategoria	Wycofana funkcja	Zastępnik	Nazwa funkcji
Zdalne serwery	SET REMOTE_PROC_TRANSACTIONS	Należy zastąpić zdalne serwery serwerami połączonymi	SET REMOTE_PROC_TRANSACTIONS
Opcje instrukcji SET	SET ROWCOUNT w instrukcjach INSERT, UPDATE i DELETE	Słowo kluczowe TOP	SET ROWCOUNT
Wskazówka dotycząca tabel	Wskazówka HOLDLOCK bez nawiasu	Należy używać wskazówki HOLDLOCK w nawiasie	Wskazówka HOLDLOCK bez nawiasu
Narzędzia	Narzędzie sqlmaint	Należy zastosować plan konserwacji systemu SQL Server	Brak

UWAGA To tylko niektóre wycofane funkcje. Kompletną listę funkcji uznanych za przestarzałe i wycofanych znajdziesz na stronie <http://msdn.microsoft.com/en-us/library/ms144262.aspx>.

Funkcje baz danych uznane za przestarzałe w wersji SQL Server 2014

Funkcje z tej grupy nie są dostępne w wersji SQL Server 2014 i nie pojawią się w przyszłych wydaniach tego produktu. Poniżej wymieniono niektóre z tych mechanizmów. Staraj się stopniowo zastępować je zalecanymi odpowiednikami.

- Punkty końcowe protokołów SOAP/HTTP utworzone za pomocą instrukcji CREATE ENDPOINT lub ALTER ENDPOINT. Zostały one zastąpione platformą WCF (ang. *Windows Communication Framework*) i technologią ASP.NET.
- Poziom zgodności 90 nie będzie dostępny w wersjach nowszych niż SQL Server 2012.
- Możliwość szyfrowania za pomocą algorytmów RC4 i RC4_128 ma zostać usunięta w następnej wersji. Pomyśl o zastosowaniu innego algorytmu szyfrowania, np. AES.
- W następnych wersjach systemu SQL Server nie będzie można pomijać średnika na końcu instrukcji języka T-SQL.

Inne zmiany w działaniu funkcji w wersji SQL Server 2014

Zmiany w działaniu wymienionych poniżej funkcji mogą utrudnić migrację do wersji SQL Server 2014.

- Gdy stworzysz nowe zadanie za pomocą kopiowania skryptu istniejącego zadania, nowe zadanie może przypadkowo wpłynąć na istniejące. Dzieje się tak, ponieważ nie należy kopiować parametru @schedule_uid. Usuń ten parametr ręcznie w skryptach nowego zadania.

- Deterministyczne funkcje zdefiniowane przez użytkownika zwracające wartości skalarne typów środowiska CLR i deterministyczne metody ze zdefiniowanych przez użytkownika typów CLR są wstępnie przetwarzane (ang. *folding*). Jeśli jednak niedeterministyczna funkcja lub metoda zostanie błędnie oznaczona jako deterministyczna, może zwracać nieoczekiwane wyniki.
- Aktualizacja bazy danych z indeksem podzielonym na partycje może prowadzić do uzyskania innego histogramu dla indeksu. Dzieje się tak, ponieważ wersja SQL Server 2014 generuje statystyki za pomocą algorytmu próbkującego, a nie metodą pełnego skanowania.
- Program *sqlcmd.exe* w trybie XML działa w systemie SQL Server 2014 inaczej niż we wcześniejszych wersjach.

UWAGA Omówienie innych zmian w działaniu funkcji znajdziesz w dokumentacji SQL Server 2014 Books Online i na stronie <http://msdn.microsoft.com/en-us/library/ms143359%28v=sql.120%29>.

Zmiany w działaniu niektórych funkcji mogą utrudnić migrację do wersji SQL Server 2014. Przykładowo we wcześniejszych wersjach systemu SQL Server kwerendy dotyczące dokumentów w formacie XML zawierających długie łańcuchy znaków (ponad 4020 znaków) mogły zwracać błędne wyniki. W wersji SQL Server 2014 takie kwerendy działają prawidłowo.

Uwagi dotyczące komponentów systemu SQL Server

W tym podrozdziale omawiamy poszczególne komponenty i przedstawiamy uwagi, które należy uwzględnić w trakcie aktualizowania systemu. Komponenty pominięte w tym miejscu są opisane dalej, w poświęconych im rozdziałach.

Aktualizowanie katalogu pełnotekstowego

W trakcie aktualizowania systemu dla wszystkich baz danych z katalogiem pełnotekstowym ten katalog jest wyłączany. Dzieje się tak z powodu czasu, jaki jest potrzebny na odtworzenie takich katalogów. Przed zaktualizowaniem środowiska obsługującego wyszukiwanie pełnotekstowe powinieneś zapoznać się z niektórymi usprawnieniami tej funkcji. Proces podłączania i odłączania baz danych także powoduje wyłączenie katalogu pełnotekstowego. Więcej informacji na ten temat zawiera dokumentacja SQL Server 2014 Books Online.

Aktualizowanie usług Reporting Services

Usługi Reporting Services 2005 i 2008 umożliwiają aktualizację do wersji 2014. Przed aktualizacją uruchom narzędzie SQL Server 2014 Upgrade Advisor i zastosuj się do przedstawionych przez nie rekomendacji oraz wskazówek pozwalających rozwiązać problemy i wykonaj zalecane kroki. Następnie, przed aktualizacją, utwórz kopię zapasową bazy danych, aplikacji, plików konfiguracyjnych i klucza szyfrowania.

Poniżej opisano dwa sposoby aktualizowania usług Reporting Services.

- **Aktualizacja w miejscu.** Aby ją przeprowadzić, uruchom plik *setup.exe* systemu SQL Server 2014, a następnie wybierz starszą wersję usług, by zaktualizować ją do wersji Reporting Services 2014. To podejście ma te same wady i zalety, jakie opisano wcześniej w kontekście aktualizacji w miejscu całego systemu. Jest to metoda typu „wszystko albo nic” — trudno wycofać zmiany bez ponownego zainstalowania dawnej wersji.
- **Aktualizacja przy równoległe działających wersjach.** W tym podejściu egzemplarz usług Reporting Services 2014 jest instalowany razem z inną wersją tych usług na tym samym lub odrębnym serwerze fizycznym. Po zainstalowaniu usług Reporting Services 2014 zawartość raportów można przenieść pojedynczo lub masowo za pomocą jednej z technik, takich jak:
 - ponowne dodanie raportów za pomocą narzędzia SSDT (ang. *SQL Server Data Tools*),
 - użycie pliku *rs.exe* do pobrania i dodania raportów,
 - użycie menedżera raportów.

Opublikowane raporty i raporty dotyczące snapshotów są aktualizowane. Po zaktualizowaniu usług do wersji Reporting Services 2014 ponownie dodaj wszystkie niestandardowe rozszerzenia i komponenty, przetestuj aplikacje w działającej wersji tych usług, a następnie usuń nieużywane aplikacje i narzędzia z wcześniejszej wersji.

UWAGA W trakcie aktualizowania systemu do wersji SQL Server 2014 domyślny poziom zgodności bazy zostaje zachowany. Musisz ręcznie zmienić go z 90 (wersja 2005), 100 (wersja 2008 R2) lub 110 (wersja 2012) na 120 (wersja 2014), aby wykorzystać nowy optymalizator kwerend i dostępne w nim usprawnienia.

Aktualizowanie do wersji 64-bitowej

Aktualizowanie systemu z 32-bitowych wersji SQL Server 2005 lub 2008 do 64-bitowej wersji SQL Server 2008 nie jest możliwe. Choć można uruchomić 32-bitowe wersje SQL Server 2005 SP2 lub SQL Server 2008 w 64-bitowym systemie Windows, aktualizowanie ich do 64-bitowego systemu SQL Server 2014 nie jest obsługiwane. Jedyny sposób na przeniesienie baz danych z wersji 32-bitowych do systemu 64-bitowego to migracja przy równoległe zainstalowanych wersjach.

Testy po aktualizacji

W tym podrozdziale opisano zachowania systemu zaskakujące wiele osób, które przeprowadziły aktualizację. Nie ma nic gorszego niż na pozór udana aktualizacja systemu, po której użytkownicy zaczynają narzekać na nową wersję z powodu wolnego przetwarzania kwerend. Aktywne uwzględnienie potencjalnych trudności zmniejsza ryzyko tego, że staranne plany i ciężka praca zostaną zniweczone problemami występującymi po aktualizacji.

Powodem niskiej wydajności kwerend po aktualizacji systemu do wersji SQL Server 2014 jest to, że dawne statystyki są uznawane za przestarzałe i nie są używane przez optymalizator kwerend. W większości sytuacji nie stanowi to problemu, pod warunkiem jednak, że administrator włączył automatyczne aktualizowanie i generowanie statystyk. Dzięki temu statystyki są domyślnie automatycznie aktualizowane, gdy są potrzebne przy kompilowaniu kwerend.

Automatycznie generowane statystyki są oparte na próbkach danych. Dlatego mogą być mniej precyzyjne od statystyk utworzonych na podstawie pełnego zbioru danych. W bazach danych z dużymi tabelami lub w tabelach, dla których wcześniejsze statystyki wygenerowano w trybie pełnego skanowania, różnica w precyzji może spowodować, że optymalizator kwerend systemu SQL Server 2014 przygotowuje nieoptymalny plan wykonania kwerendy.

UWAGA W systemie SQL Server 2014 przy tworzeniu indeksu do generowania statystyk używany jest domyślny algorytm próbujący optymalizatora kwerend.

Aby rozwiązać ten problem, po zaktualizowaniu systemu do wersji SQL Server 2014 należy natychmiast zaktualizować także statystyki. Funkcja `sp_updatestats` z parametrem `resample` odtwarza statystyki na podstawie wielkości próbek użytych do otrzymania poszczególnych istniejących statystyk. Zwykle dla statystyk dotyczących indeksów używane są pełne dane, a dla pozostałych kolumn — próbki. Dodatkowa korzyść z tego procesu jest taka, że jeśli dane zajmują mniej niż 8 megabajtów (jest to minimalna wielkość próbki), statystyki są budowane na podstawie pełnego skanowania.

Podsumowanie

Jest wiele dobrych powodów, by zaktualizować system do wersji SQL Server 2014. Do przeprowadzenia aktualizacji można zastosować narzędzia Upgrade Assistant wersji SQL Server 2014 lub SQL Server 2014 Upgrade Advisor. Proces aktualizacji obejmuje operacje wstępne, samą aktualizację i zadania wykonywane po jej przeprowadzeniu.

Powinieneś zwrócić uwagę na różne wycofane funkcje (w tym poziom zgodności bazy 90 i narzędzie SQL Mail), a także mechanizmy, których działanie się zmieniło, co może wpływać na pracę systemu po aktualizacji (dotyczy to np. histogramów indeksów podzielonych na partycje i programu `sqlcmd.exe` w trybie XML-a). Po wykonaniu podstawowych zadań potrzebnych do udanej aktualizacji pora przejść do omówienia systemu SQL Server 2014.

Z rozdziału 4. dowiesz się, jak zarządzać silnikiem bazodanowym systemu SQL Server i jak rozwiązywać związane z nim problemy.

Zarządzanie silnikiem bazodanowym i rozwiązywanie związanych z nim problemów

ZAWARTOŚĆ ROZDZIAŁU:

- Konfigurowanie egzemplarza systemu za pomocą narzędzia SQL Server Management Studio.
- Używanie dedykowanego połączenia administracyjnego do uzyskania dostępu do niereagującego serwera.
- Śledzenie procesów za pomocą obiektów DMO (ang. Dynamic Management Objects).

Czytając ten rozdział, dowiesz się, jak konfigurować i monitorować egzemplarze systemu SQL Server. Właściwa konfiguracja i aktywne śledzenie pracy to podstawa, jeśli chcesz uzyskać działające bez przeszkód bazy danych. Ponadto poznasz techniki rozwiązywania problemów, które pomogą ustalić, co się dzieje, gdy dany egzemplarz nie działa zgodnie z oczekiwaniami. W tym rozdziale zakładamy, że znasz podstawy używania narzędzia SQL Server Management Studio, i koncentrujemy się na tym, co jest potrzebne administratorom baz danych. W wielu innych rozdziałach tej książki omówione są różne aspekty korzystania z tego narzędzia, dlatego nie opisujemy ich w tym miejscu (np. tworzenie kopii zapasowych bazy danych przedstawiamy w rozdziale 17., „Kopie zapasowe i przywracanie stanu”).

Narzędzia konfiguracyjne i administracyjne

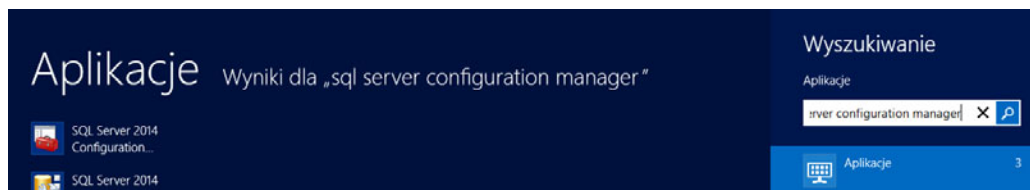
Po zainstalowaniu systemu SQL Server 2014 lub zaktualizowaniu oprogramowania do tej wersji prawdopodobnie będziesz musiał skonfigurować go do własnych potrzeb. W wersji SQL Server 2014 Microsoft nadal stosuje zasadę zapewniania bezpieczeństwa od początku (po raz pierwszy wprowadzono ją w systemie SQL Server 2008). Polega ona na wyłączeniu różnych funkcji po instalacji, co ogranicza podatność na ataki. To, które funkcje są wyłączane, zależy od edycji systemu SQL Server. Przykładowo obsługa protokołu TCP/IP jest domyślnie wyłączona w edycji Developer Edition, a w każdym wydaniu wyłączona jest integracja z środowiskiem CLR. Dzięki temu środowisko jest łatwiejsze w użyciu dla administratorów, ponieważ niepotrzebne im funkcje nie zajmują miejsca na panelu administracyjnym. Ponadto zmniejsza to ryzyko, że haker lub — co bardziej prawdopodobne — złośliwy użytkownik włamie się do systemu. Takie zabezpieczenia jeszcze bardziej uzasadniają poświęcenie czasu na skonfigurowanie systemu SQL Server pod kątem własnych wymagań. Z tego podrozdziału dowiesz się, jak skonfigurować system SQL Server na potrzeby własnego środowiska i pod kątem wymagań z zakresu bezpieczeństwa. Poznasz tu kilka technik, takich jak używanie narzędzia SQL Server Configuration Manager, korzystanie z parametrów uruchomieniowych, zastosowanie uruchomieniowych procedur składowanych i częściowo niezależnych baz danych. Najlepiej zacząć od narzędzia SQL Server Configuration Manager.

SQL Server Configuration Manager

Narzędzie SQL Server Configuration Manager służy do konfigurowania usług systemu SQL Server. Działa więc podobnie jak aplet Usługi z Panelu sterowania, jednak udostępnia znacznie więcej funkcji. Może np. zmienić porty używane przez system SQL Server i protokoły wykorzystywane w poszczególnych egzemplarzach.

Aby rozpocząć konfigurowanie systemu SQL Server na potrzeby własnego środowiska, wykonaj następujące kroki.

1. Przesuń kursor w prawy górny róg (w systemach Windows 8, Windows 2012 i nowszych), aby pojawiła się ikona lupy. Kliknij ją, a następnie wpisz **SQL Server Configuration Manager** i kliknij ikonę omawianego narzędzia. Przedstawiono to na rysunku 4.1.



Rysunek 4.1. Uruchamianie programu SQL Server Configuration Manager

2. Aby skonfigurować pojedynczą usługę, np. SQL Server, kliknij ją dwukrotnie; otworzy się strona *Właściwości*. W zakładce *Log On* możesz skonfigurować, które konto ma uruchamiać tę usługę. Usługę SQL Server należy uruchamiać ze zwykłego konta użytkownika domeny z minimalnymi uprawnieniami. To konto nie powinno np. mieć uprawnień do lokalnego logowania się. Nie ma powodu, dla którego dla tej usługi w systemie SQL Server 2014 warto byłoby stosować konto administratora lokalnego lub domeny.

- Następnie utwórz niewygasające hasło, tak aby system SQL Server nie przestał się uruchamiać po ewentualnym wygaśnięciu hasła. Jeśli usługi systemu SQL Server nie muszą komunikować się z jednostkami spoza lokalnego komputera, powinieneś uruchamiać usługę za pomocą lokalnego konta systemowego, jednak możliwe, że ma ono wysokie uprawnienia lokalne, co jest niepożądane. Dokładne omówienie tego zagadnienia podajemy w rozdziale 8., „Zabezpieczanie egzemplarzy baz danych”.

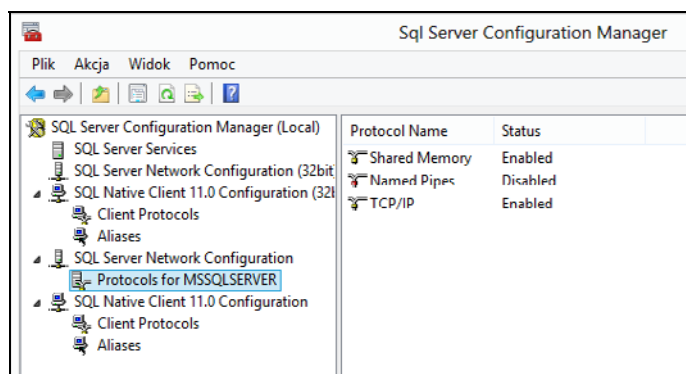
UWAGA Do zmiany właściwości usługi należy używać narzędzia SQL Server Configuration Manager zamiast okien dialogowych usług systemu Windows z narzędzi administracyjnych. W wersji SQL Server 2005 zmieniono sposób szyfrowania haseł dla kont usługowych. Jeśli zmienisz hasło w ustawieniach usług systemu Windows, a nie w programie SQL Server Configuration Manager, usługa nie uruchomi się po ponownym rozruchu systemu SQL Server.

- W zakładce *Service* możesz określić, czy dana usługa ma być uruchamiana automatycznie, ręcznie, czy ma pozostać wyłączona.

Pozostałe ustawienia są opcjonalne, ale gorąco zachęcamy do ich stosowania. W zakładce *Advanced* każdej usługi możesz ustawić ciekawe opcje. Możesz np. wyłączyć przysyłanie raportów z informacjami zwrotnymi od użytkownika (opcja *Customer Feedback Reporting*). Funkcja umożliwia Microsoftowi otrzymywanie raportów na temat użytkowania danego egzemplarza systemu SQL Server. Nawet gdybyś chciał udostępniać takie raporty, w większości środowisk produkcyjnych system SQL Server nie może ich wysyłać z powodu braku dostępu serwerów produkcyjnych do internetu.

Możesz też włączyć opcję *Error Reporting* z zakładki *Advanced*, aby system wysyłał do Microsoftu e-mail po wystąpieniu każdego błędu krytycznego. Tego rodzaju ograniczone i anonimowe informacje są przysyłane bezpiecznym protokołem HTTPS.

Jeśli ponadto w narzędziu SQL Server Configuration Manager wybierzesz opcję *SQL Server 2014 Network Configuration* (patrz rysunek 4.2), zobaczysz listę protokołów sieciowych używanych przez poszczególne egzemplarze systemu SQL Server. Jeśli zechcesz włączyć lub wyłączyć konkretny protokół, kliknij go prawym przyciskiem myszy i wybierz opcję *Enable* lub *Disable*. Gdy włączysz tylko protokół *Shared Memory*, jedynie klienci działające na tym samym komputerze będą mogły łączyć się z danym egzemplarzem systemu SQL Server.



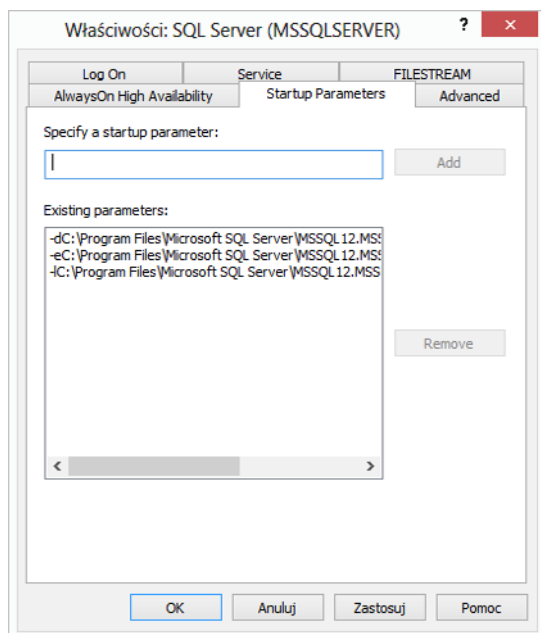
Rysunek 4.2. Włączanie i wyłączanie protokołów

UWAGA W wersji SQL Server 2012 wycofano protokół VIA. Nie jest dostępny także w wersji SQL Server 2014.

Parametry uruchomieniowe

System SQL Server zawiera szereg opcji, które pozwalają włączyć zaawansowane ustawienia silnika bazodanowego lub są pomocne przy rozwiązywaniu problemów. Opcje te można ustawiać na dwa sposoby: w narzędziu SQL Server Configuration Manager i za pomocą wiersza poleceń. Aby włączyć opcje przy użyciu narzędzia SQL Server Configuration Manager, należy zmodyfikować parametry uruchomieniowe usługi SQL Server. W tym podejściu system SQL Server będzie używał ustawionych parametrów przy każdym uruchomieniu. Możesz też tymczasowo ustawić opcje, gdy uruchamiasz usługę SQL Server z poziomu wiersza poleceń. Aby włączyć wybraną opcję, wykonaj następujące kroki.

1. Na stronie usług systemu SQL Server 2014 kliknij dwukrotnie usługę SQL Server (domyślnie jej nazwa to *MSSQLServer*, jednak może być inna, co zależy od nazwy egzemplarza).
2. Otwórz zakładkę *Startup Parameters*. Dodaj parametry uruchomieniowe (*opcje śledzenia*), które mają być włączone przy uruchamianiu egzemplarza systemu SQL Server. Zilustrowano to na rysunku 4.3.



Rysunek 4.3. Włączanie parametrów uruchomieniowych

Jeśli chcesz włączyć daną opcję przy uruchamianiu systemu z poziomu wiersza poleceń, wykonaj następujące operacje.

1. Uruchom program *sqlservr.exe* w wierszu poleceń. Program znajduje się w katalogu *C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Binn.*

2. Włącz wszystkie potrzebne parametry, które nie są domyślne. Podaj je po nazwie *sqlservr.exe*. Jest to zalecany sposób jednorazowego uruchamiania systemu SQL Server na potrzeby debugowania, ponieważ nie grozi pozostawieniem niepożądanych zmian w ustawieniach. Aby zatrzymać system SQL Server, użyj kombinacji *Ctrl+C* lub zamknij okno wiersza poleceń.

OSTRZEŻENIE Nigdy nie uruchamiaj systemu SQL Server w wierszu poleceń, jeśli chcesz go używać w zwykły sposób, ponieważ po wylogowaniu się z komputera wiersz poleceń zostanie zamknięty, co spowoduje zatrzymanie systemu.

Oto składnia służąca do uruchamiania programu *SQLServr.exe*:

```
SQLServr.exe [-dścieżka_do_głównego_pliku] [-lścieżka_do_głównego_dziennika]
              -eścieżka_do_dziennika_błędów] [-snazwa_egzemplarza][-c] [-f]
              [-grezerwowana_pamięć] [-h] [-kszybkość_w_megabajtach_na_sekundę]
              [-m] [-n] [-tnumer_śladu] [-tnumer_śladu] [-x]
              [-yzzzut_stosu_dla_danego_błędu] [-B] [-K]
```

Dostępne opcje uruchomieniowe są przydatne przy rozwiązywaniu problemów lub drobnych, jednorazowych trudności. Zamiast opisywać wszystkie opcje, poniżej przedstawiamy najczęściej używane.

- **Opcje -d i -l.** Za pomocą opcji -d i -l można zmienić bazę master używaną przez system SQL Server:
`SQLServr.exe -d C:\temp\TempMasterDB.mdf -l C:\temp\TempMasterLog.ldf`
 Opcja -d określa plik bazy danych, a opcja -l wskazuje plik dziennika. Są one przydatne, gdy chcesz zastosować tymczasową, nieuszkodzoną bazę master.
- **Opcja -T.** Jest to inna przydatna opcja. Umożliwia ona użycie danych opcji śledzenia dla wszystkich połączeń z konkretnym egzemplarzem systemu SQL Server. W ten sposób można np. włączyć opcję śledzenia w celu wykrywania zakleszczeń w danym egzemplarzu (zwróć uwagę, że używana jest wielka litera T):
`SQLServr.exe -T1204`

UWAGA System SQL Server udostępnia też opcję śledzenia włączaną za pomocą małej litery t: `SQLServr.exe -t1204`. Należy jednak stosować ją z rozważą, ponieważ ustawia ona inne wewnętrzne opcje śledzenia.

OSTRZEŻENIE Jeśli spróbujesz uruchomić egzemplarz systemu SQL Server w czasie, gdy już działa, pojawiają się błędy. SQL Server otwiera plik master w trybie na wyłączność, aby zapobiec zapisowi danych w tym samym pliku przez inny egzemplarz. Jeden z błędów informuje, że plik jest używany w trybie wyłączności i jest niedostępny. Taka sytuacja nie powoduje jednak żadnych szkód.

- **Opcja -f.** Uruchamia ona system SQL Server w trybie minimalnym z włączonym tylko jednym połączeniem. W tym trybie system SQL Server rozpoczyna pracę z minimalną konfiguracją z zatrzymanymi: procesem CHECKPOINT, uruchomieniowymi procedurami składowanymi i połączeniami zdalnymi.

UWAGA Gdy używasz narzędzia SQL Server Configuration Manager, zadбай o to, aby zatrzymać agenta SQL Server przed przełączeniem systemu SQL Server w tryb jednego użytkownika. W przeciwnym razie ten agent zajmie jedyne dostępne połączenie.

- **Opcja g.** Pozwala zarezerwować dodatkową pamięć poza główną pulą pamięci systemu SQL Server. Pamięć jest używana przez rozszerzone procedury składowane.
- **Opcja -m.** Uruchamia system SQL Server w trybie jednego użytkownika (można go nazwać *trybem przywracania bazy master*) i zawiesza proces CHECKPOINT, który zapisuje dane z dysku na urządzenie z bazą danych. Jest to przydatne, gdy administrator chce przywrócić bazę master z kopii zapasowej lub wykonać inne awaryjne zadania konserwacyjne.
- **Opcja -k.** Wpływa na częstotliwość wykonywania procesu CHECKPOINT. Wymusza odtworzenie nadrzędnego klucza systemu, jeśli jest używany. Stosuj tę opcję z należytą ostrożnością i tylko wtedy, gdy zaleci to pracownik pomocy technicznej Microsoftu.
- **Opcja -s.** Uruchamia egzemplarz systemu SQL Server o określonej nazwie. Gdy uruchamiasz system SQL Server z wiersza poleceń, używany jest egzemplarz domyślny, chyba że wskażesz odpowiedni katalog BINN danego egzemplarza i zastosujesz opcję -s. Jeśli np. potrzebny egzemplarz to SQL2014, powinien się on znajdować w katalogu `C:\Program Files\Microsoft SQL Server\MSSQL12.SQL2014\MSSQL\Binn`. Aby uruchomić ten egzemplarz, wywołaj następujące polecenie:
`sqlservr.exe -sSQL2014`
- **Opcja -c.** Umożliwia skrócenie czasu rozruchu przy włączaniu systemu SQL Server z wiersza poleceń. Jest to możliwe, ponieważ w takiej sytuacji silnik bazodanowy nie jest uruchamiany jako usługa. Opcja pozwala pominąć narzędzie Service Control Manager, które w tym podejściu jest zbędne.

Kompletną listę opcji możesz wyświetlić za pomocą przełącznika -?, co pokazano poniżej (kompletne informacje znajdziesz w dokumentacji systemu SQL Server):

```
sqlservr.exe -?
```

Uruchomieniowe procedury składowane

Uruchomieniowe procedury składowane działają podobnie jak zwykłe procedury składowane, są jednak wykonywane przy każdym uruchomieniu egzemplarza systemu SQL Server. Możesz np. utworzyć procedurę wysyłającą do Ciebie e-maila przy uruchomieniu danego egzemplarza. Ponadto uruchomieniowe procedury składowane można wykorzystać do tworzenia obiektów w bazie tempdb i wczytywania ich razem z danymi przy włączaniu systemu SQL Server. Takie procedury działają z rolą administratora systemu i tylko administrator może je tworzyć. Błędy zgłoszone w takich procedurach są zapisywane w dzienniku błędów systemu SQL Server.

W opisanych poniżej krokach pokazano, jak włączyć uruchomieniowe procedury składowane dla całego systemu i jak utworzyć przykładową procedurę tego rodzaju.

OSTRZEŻENIE Przykłady z tego punktu wykonuj tylko na serwerze rozwojowym, chyba że jesteś pewien, iż chcesz wprowadzić zmiany w środowisku produkcyjnym.

1. Procedura składowana `sp_configure` włącza uruchomieniowe procedury składowane, ale aby ją skonfigurować, musisz włączyć ustawienie `show advanced options`, co ilustruje poniższy kod:

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
```

2. System SQL Server domyślnie nie sprawdza dostępności uruchomieniowych procedur składowanych. Aby umożliwić wykrywanie takich procedur, musisz zastosować procedurę `sp_configure`:

```
sp_configure 'scan for startup procs', 1;
GO
RECONFIGURE;
GO
```

3. Po wywołaniu tych instrukcji uruchom ponownie egzemplarz systemu SQL Server, aby zatwierdzić zmiany. Spróbuj wykonać następujący prosty przykład. Utwórz w bazie master tabelę o nazwie `SQLStartupLog`, w której zapisywany będzie czas uruchomienia egzemplarza systemu SQL Server:

```
CREATE TABLE master.dbo.SQLStartupLog
(StartTime datetime);
GO
```

4. Teraz utwórz procedurę składowaną zapisującą dane w nowej tabeli. Procedurę koniecznie umieść w bazie master. Przedstawiona poniżej procedura składowana zapisuje bieżącą datę we wspomnianej wcześniej tabeli:

```
USE master
GO
CREATE PROC dbo.InsertSQLStartupLog
as
INSERT INTO master.dbo.SQLStartupLog
SELECT GETDATE();
GO
```

5. Za pomocą procedury składowanej `sp_procoption` przekształć zwykłą procedurę składowaną na uruchomieniową. Procedura `sp_procoption` ustawia tylko jeden parametr; jest to parametr `startup` o wartościach 1 (włączony) i 0 (wyłączony). Przed tym parametrem należy podać skonfigurowaną procedurę składowaną. Zanim uruchomisz procedurę `sp_procoption`, upewnij się, że system SQL Server sprawdza dostępność uruchomieniowych procedur składowanych.

```
sp_procoption @ProcName = 'master.dbo.InsertSQLStartupLog',
@OptionName= 'startup',
@OptionValue = 1;
```

6. Zatrzymaj i uruchom egzemplarz systemu SQL Server, a następnie sprawdź w tabeli `master.dbo.SQLStartupLog`, czy rekord został zapisany. Zanim przejdiesz do następnego punktu, wyłącz włączone wcześniej ustawienie. Użyj do tego następującej kwerendy:

```
sp_procoption @ProcName = 'master.dbo.InsertSQLStartupLog',
@OptionName= 'startup',
@OptionValue = 0;
```

```
USE MASTER
GO
DROP TABLE master.dbo.SQLStartupLog;
DROP PROC dbo.InsertSQLStartupLog;
```

Częściowo niezależne bazy danych

Częściowo niezależne bazy danych (ang. *partially contained database*) pojawiły się jako nowość w wersji SQL Server 2012. Są one doskonałym rozwiązaniem w określonych scenariuszach wymagających bezpieczeństwa. Pełne omówienie korzyści w zakresie bezpieczeństwa, jakie daje ta technika, znajdziesz w rozdziale 8., „Zabezpieczanie egzemplarzy baz danych”. *Niezależna baza danych* charakteryzuje się tym, że wszystkie jej ustawienia i metadane nie są zależne od egzemplarza systemu SQL Server, w którym ta baza działa. Użytkownicy mogą łączyć się z bazą bez uwierzytelniania na poziomie egzemplarza systemu. Taki poziom izolacji sprawia, że tak skonfigurowane bazy można łatwo przenosić między egzemplarzami, co jest wygodne, jeśli baza ma działać w wielu egzemplarzach (np. w środowisku rozwojowym).

W systemie SQL Server 2014 w pełni niezależne bazy danych nie są dostępne. Można używać tylko częściowo niezależnych baz danych, co oznacza, że dozwolone jest korzystanie z obiektów lub funkcji przekraczających granice aplikacji.

UWAGA *Granica aplikacji* to granica między modelem aplikacji (bazą danych) a egzemplarzem systemu. Przykładowo tabela systemowa `sys.endpoints` działa poza granicami aplikacji, ponieważ używa obiektów z poziomu egzemplarza. Z kolei tabela systemowa `sys.indexes` działa w granicach aplikacji.

Domyślnie w egzemplarzu systemu SQL Server 2014 niezależne bazy danych nie są dostępne. Aby umożliwić przeniesienie istniejącej bazy do tego systemu, zastosuj procedurę `sp_configure` (przedstawioną w poniższym fragmencie kodu):

```
sp_configure 'contained database authentication', 1;
```

```
GO
RECONFIGURE WITH OVERRIDE
GO
```

Przed przeniesieniem bazy danych do modelu niezależnego użyj nowego obiektu DMO `sys.dm_db_uncontained_entities`, aby określić poziom zamknięcia danej bazy. Dane wyjściowe z poniższej kwerendy obejmują obiekty, które mogą przekraczać granice aplikacji:

```
SELECT so.name, ue.*
FROM sys.dm_db_uncontained_entities ue
LEFT JOIN sys.objects so
ON ue.major_id = so.object_id;
```

UWAGA Dodatkowy sposób na identyfikowanie zależnych zdarzeń w aplikacji to użycie rozszerzonego zdarzenia `database_uncontained_usage_event`. Zdarzenie to jest uruchamiane za każdym razem, gdy w aplikacji wystąpi zależne zdarzenie. Więcej informacji na temat rozszerzonych zdarzeń zamieszczono w rozdziale 12., „Monitorowanie systemu SQL Server”.

Jeśli dana baza dobrze nadaje się do przekształcenia w bazę niezależną i mechanizm obsługi takich baz jest włączony, można zastosować opcję `CONTAINMENT`, aby przekształcić bazę w częściowo niezależną. Bazy w tej postaci nie powinny używać funkcji z poziomu egzemplarza, takich jak usługa Service Broker. Do modyfikowania bazy służy instrukcja `ALTER DATABASE`, co ilustruje poniższy fragment kodu:


```
USE master
GO
ALTER DATABASE AdventureWorks SET CONTAINMENT = PARTIAL;
```

UWAGA Więcej informacji o przechodzeniu na częściowo niezależne bazy oraz o zagrożeniach i ograniczeniach związanych z tym mechanizmem znajdziesz pod adresem [http://msdn.microsoft.com/en-us/library/ff929139\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ff929139(v=sql.120).aspx).

Narzędzia do rozwiązywania problemów

Wyobraź sobie, że otrzymujesz telefon z informacją, iż serwer nie odpowiada. Włączasz narzędzie SQL Server Management Studio, aby połączyć się z serwerem i sprawdzić, co się dzieje. Jednak żądanie połączenia się po długim oczekiwaniu przekracza limit czasu. Nie możesz się połączyć, nie możesz zdiagnozować problemu i nie możesz zobaczyć, co się stało. W takiej sytuacji potrzebujesz narzędzi, które pomogą wykryć i rozwiązać problemy z danym egzemplarzem systemu. DAC (ang. *Dedicated Administrator Connection*) to niezawodne narzędzie używane w takich sytuacjach. Inna metoda rozwiązywania problemów, także opisana w tym podrozdziale, polega na odtworzeniu systemowych baz danych, gdy któraś z nich zostanie uszkodzona lub utracona.

Połączenie DAC

DAC to specjalne połączenie diagnostyczne, które można wykorzystać, gdy nie da się nawiązać połączenia z serwerem w standardowy sposób. Jeśli musisz połączyć się z serwerem w celu zdiagnozowania i rozwiązania problemów, DAC jest nieocenionym narzędziem administracyjnym.

OSTRZEŻENIE System SQL Server próbuje nawiązać połączenie DAC niezależnie od sytuacji i w większości przypadków to się udaje. Gdy jednak problemy są bardzo poważne, czasem nie da się połączyć z nieodpowiadającym na żądania serwerem.

System SQL Server oczekuje na połączenie DAC w dynamicznie przypisywanym porcie. Połączenie z tym portem mogą nawiązać tylko jednostki pełniące rolę administratora systemu i używające narzędzi SSMS lub sqlcmd. Aby użyć połączenia DAC za pomocą narzędzia SQL Server Management Studio, dodaj odpowiedni przedrostek do nazwy serwera. Jeśli np. nazwa serwera to SQL2014, połącz się z serwerem admin:SQL2014. Wystarczy więc dodać przedrostek admin: do nazwy serwera.

Jeśli chcesz nawiązać połączenie za pomocą narzędzia sqlcmd, użyj opcji -A:

```
sqlcmd -sSQL2014 -E -A -d master
```

Domyślnie połączenia DAC są dostępne tylko dla klientów działających na lokalnym serwerze. Jednak za pomocą procedury sp_configure możesz włączyć zdalne połączenia administracyjne:

```
sp_configure 'remote admin connection', 1;
GO
RECONFIGURE WITH OVERRIDE
GO
```

Powinieneś włączyć zdalne połączenia administracyjne na egzemplarzach systemu, na których procedury bezpieczeństwa i odpowiedzialny za nie zespół na to pozwalają. W niektórych sytuacjach trzeba połączyć się za pomocą adresu TCP i numeru portu połączenia DAC znalezionych w dzienniku błędów (dotyczy to zwłaszcza egzemplarzy działających w klastrach), co pokazano poniżej:

```
Sqlcmd -S<AdresSerwera>,<PortDac>
```

Jeśli łączysz się lokalnie przy użyciu adresu IP, skorzystaj z adresu karty źródłowej (ang. *loopback adapter*), tak jak w poniższej instrukcji:

```
Sqlcmd -S127.0.0.1,1434
```

UWAGA Adres karty źródłowej to adres komputera lokalnego. W większości systemów jest to adres IPv4 127.0.0.1. Połączenie się z tym adresem daje ten sam efekt, co nawiązanie połączenia z komputerem lokalnym.

Przy łączeniu się zdalnym trzeba znać numer portu przypisanego połączeniu DAC. Domyślnie używany jest port 1434, jednak możliwe, że do egzemplarza przypisano inny numer, jeśli połączenie z domyślnym portem nie powiodło się w czasie rozruchu systemu. Jeśli ponadto zdalne połączenia administracyjne są włączone, przy nawiązywaniu połączenia DAC trzeba podać numer portu. Dobrym narzędziem do znalezienia tego numeru jest powłoka PowerShell systemu Windows, ponieważ pozwala ona zobaczyć numery portów DAC przypisane do różnych egzemplarzy. Poniższa instrukcja powłoki PowerShell umożliwia zobaczenie, jakie porty są ustawione dla połączeń DAC w każdym egzemplarzu zapisanym w zmiennej \$instances:

```
$instances = "PRODUCTION", "PRODUCTION\R2", "PRODUCTION\SQL2012"
foreach($instance in $instances
{
  get-SQLErrorlog -SQLServer $instance |
  where {($_.Text -match "Dedicated admin connection") } |
  format-table $DisplayResults -AutoSize
}
```

Odtwarzanie systemowych baz danych

Jeśli jedna z systemowych baz danych zostanie uszkodzona i nie można znaleźć jej kopii zapasowej, ostatnią deską ratunku jest odtworzenie systemowych baz danych. Technika polega na ponownym zainstalowaniu systemowych baz danych i usunięciu z systemu wszystkich komponentów, które mogą powodować jego nieprzewidywalne działanie. Skutkiem ubocznym jest konieczność ponownego zainstalowania wszystkich pakietów poprawek i zniknięcie wszystkich baz zdefiniowanych przez użytkownika (w tym baz używanych przez usługi Reporting Services). Ponadto ponownie trzeba ustawić loginy i skonfigurować serwer.

OSTRZEŻENIE Nie należy pochopnie odtwarzać systemowych baz danych. To rozwiązanie ma poważne skutki techniczne i należy je stosować wtedy, gdy nie istnieją żadne inne dobre możliwości.

Aby odtworzyć systemowe bazy danych, wykonaj następujące kroki.

1. Otwórz wiersz poleceń.
2. W wierszu poleceń uruchom program *setup.exe*, tak jak przy instalowaniu systemu SQL Server, ale zastosuj kilka opcji:

```
setup.exe /QUIET /ACTION=REBUILDDATABASE /INSTANCENAME=instance_name
        /SQLSYSADMINACCOUNTS=accounts /SAPWD=sa password
```

Oto znaczenie tych opcji.

- Opcja /QUIET blokuje błędy i ostrzeżenia w trakcie odtwarzania baz. Podczas odtwarzania widzisz wtedy pusty ekran. Błędy zostaną jednak zapisane w dzienniku błędów.
- Opcja /ACTION oznacza tu, że wykonywana akcja to odtwarzanie bazy danych. Określa to parametr REBUILDDATABASE.
- Opcja /INSTANCENAME określa nazwę egzemplarza, w którym należy odtworzyć systemowe bazy danych. MSSQLSERVER oznacza domyślny egzemplarz.
- Opcja /SQLSYSADMINACCOUNTS określa grupy lub konta systemu Windows, którym należy przyznać uprawnienia administratora systemu. Zastosuj tę opcję, jeśli w systemie SQL Server użyto uwierzytelniania systemu Windows.
- Opcja /SAPWD określa hasło administratora systemu. Użyj tej opcji, jeśli w systemie SQL Server zastosowano mieszany tryb uwierzytelniania.

UWAGA W wersjach starszych niż SQL Server 2008 przy odtwarzaniu systemowych baz danych niezbędny był oryginalny nośnik z programem instalacyjnym. Obecnie systemowe bazy danych i pliki dziennika są w ramach instalacji kopiowane do katalogu *C:\Program Files\Microsoft SQL Server\MSSQL 12.MSSQLSERVER\MSSQL\Binn\Templates*. W trakcie odtwarzania baz program *setup.exe* używa plików zapisanych w tym katalogu.

3. Po odtworzeniu baz danych przywróć domyślną konfigurację i dodaj potrzebne bazy. Musisz przywrócić bazę master (więcej na ten temat dowiesz się z rozdziału 17.) lub ponownie dodać każdą bazę zdefiniowaną przez użytkownika i loginy. Zalecany rozwiązaniem jest oczywiście przywrócenie bazy master z kopii zapasowej zamiast odtwarzania baz. Wtedy loginy i bazy danych użytkowników będą dostępne automatycznie.

UWAGA Po odtworzeniu systemowych baz danych może się wydawać, że bazy użytkowników znikły, jednak ich pliki nadal znajdują się w systemie operacyjnym i można je ponownie podłączyć lub przywrócić. Ponowne podłączenie baz danych to zwykle najprostsze podejście.

SQL Server Management Studio

Administratorzy baz danych spędzają dużo czasu, używając tego narzędzia. Pozwala ono wykonywać większość zadań administracyjnych i uruchamiać kwerendy. Powłoką dla systemu SQL Server 2014 jest środowisko Visual Studio 2010. Ponieważ jest to książka dla profesjonalistów,

nie omawiamy tu wszystkich aspektów narzędzia SQL Server Management Studio. Zamiast tego przedstawiamy najczęściej używane i zaawansowane funkcje, które mogą przydać się podczas administrowania systemem.

Raporty

Jedną z najbardziej imponujących funkcji środowiska administracyjnego systemu SQL Server są zintegrowane raporty związane z każdym obszarem administrowania bazą. Takie standardowe raporty są dostępne dla egzemplarzy serwera, baz danych, loginów i elementów z węzła *Management*. Każdy taki raport jest tworzony za pomocą usług Reporting Services w narzędziu SQL Server Management Studio. Raporty z poziomu serwera obejmują informacje na temat egzemplarza systemu SQL Server i systemu operacyjnego. Raporty z poziomu bazy danych zawierają informacje z każdej bazy. Musisz mieć dostęp do bazy, na temat której chcesz uzyskać raport, lub musisz posługiwać się loginem zapewniającym uprawnienia do generowania raportów z poziomu serwera. Można też tworzyć niestandardowe raporty i łączyć je z wieloma innymi węzłami w oknie *Object Explorer*.

Raporty z poziomu serwera

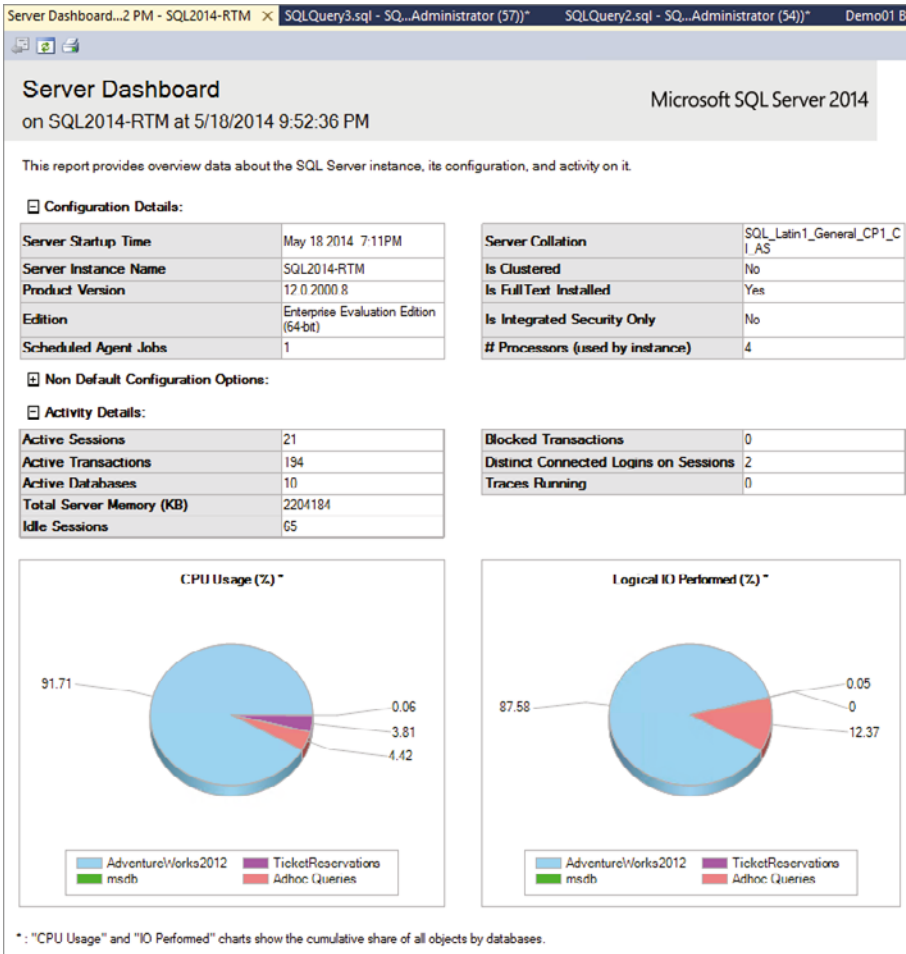
Dostęp do raportów z poziomu serwera można uzyskać w oknie *Object Explorer* narzędzia SQL Server Management Studio. W tym celu należy kliknąć prawym przyciskiem myszy wybrany egzemplarz systemu SQL Server i wybrać z menu opcję *Reports*. Jednym z najczęściej używanych raportów z poziomu serwera jest Server Dashboard widoczny na rysunku 4.4. Raport udostępnia bogate informacje na temat danego egzemplarza systemu SQL Server 2014. Oto niektóre z nich.

- Używana edycja i wersja systemu SQL Server.
- Wszystkie ustawienia egzemplarza różne od ustawień domyślnych systemu SQL Server.
- Statystyki operacji wejścia-wyjścia i wykorzystania procesora dla różnych typów zadań (np. dla kwerend ad hoc, usługi Reporting Services itd.).
- Wysokopoziomowe informacje o konfiguracji, np. o tym, czy egzemplarz działa w klastrze.

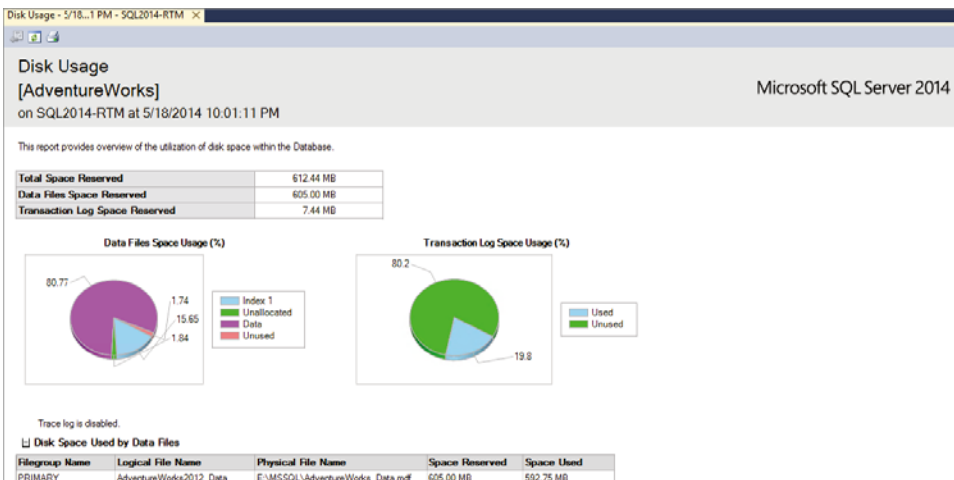
Większość informacji statystycznych obejmuje wyłącznie dane zebrane od czasu ostatniego uruchomienia systemu SQL Server. Przykładowo raport Server Dashboard zawiera kilka wykresów pokazujących wykorzystanie procesora dla różnych typów kwerend. Wykresy nie obejmują danych historycznych, a tylko informacje z czasu aktywności systemu SQL Server. Zachowaj ostrożność przy wyciąganiu wniosków na podstawie tych danych z raportu Server Dashboard. Zawsze pamiętaj, że widoczne informacje dotyczą tylko danego wycinka czasu pracy serwera, a nie całej historii.

Raporty o bazach danych

Raporty o bazach danych działają podobnie jak raporty z poziomu serwera. Aby otrzymać raport o bazie, kliknij jej nazwę prawym przyciskiem myszy w oknie *Object Explorer* narzędzia SQL Server Management Studio. W tego typu raportach widoczne są informacje na temat wybranej bazy. Możesz np. wyświetlić wszystkie transakcje przetwarzane obecnie w bazie, zablokowanych użytkowników lub zajmowaną przestrzeń na dysku. Przedstawiono to na rysunku 4.5.



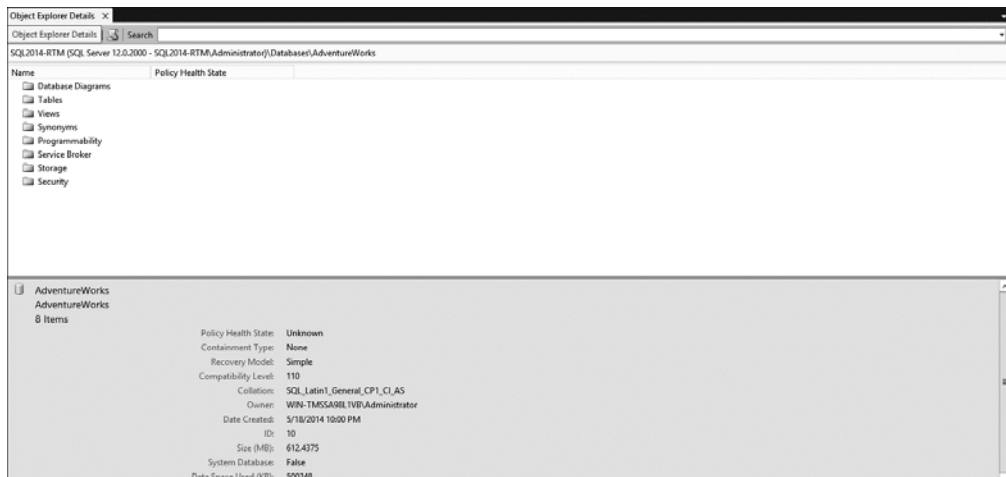
Rysunek 4.4. Raport Server Dashboard



Rysunek 4.5. Raport o bazie danych

Okno Object Explorer Details

Okno *Object Explorer Details* zawiera wiele informacji w zwięzłym interfejsie użytkownika. Dostęp do tego okna można uzyskać na dwa sposoby: możesz wybrać opcję *Object Explorer Details* z menu *View* lub wcisnąć klawisz *F7*. Jeśli zaznaczyłeś węzeł w oknie *Object Explorer*, wcisnięcie klawisza *F7* otworzy szczegółowe informacje na temat wybranego obiektu, co pokazano na rysunku 4.6. Przycisk *Synchronize* z okna *Object Explorer* pozwala zsynchronizować je z panelem *Object Explorer Details*.



Rysunek 4.6. Informacje z okna Object Explorer Details

Możesz przefiltrować zawartość tego okna i dostosować je do własnych potrzeb. Kliknięcie nagłówka kolumny powoduje posortowanie danych według niej, a kliknięcie nagłówka kolumny prawym przyciskiem myszy pozwala przefiltrować informacje.

Konfigurowanie systemu SQL Server w narzędziu SQL Server Management Studio

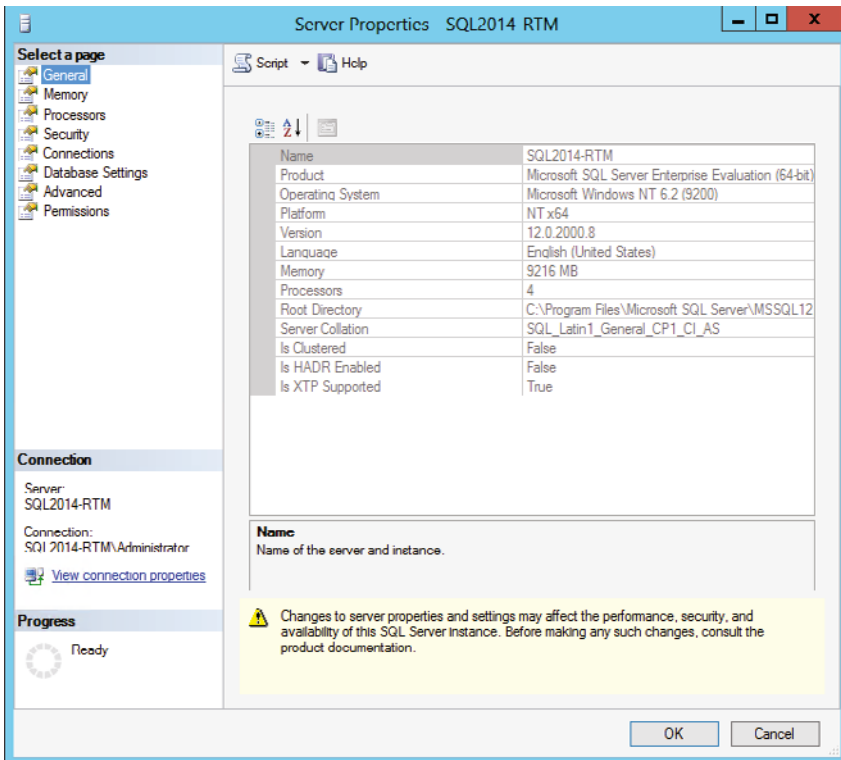
Istnieje kilka metod konfigurowania systemu SQL Server. Wcześniej używano do tego narzędzia SQL Server Configuration Manager. Pomagało ono włączyć różne funkcje i usługi. Tu opisujemy inne sposoby konfigurowania systemu i szczegółowo omawiamy procedurę `sp_configure`. Egzemplarz silnika bazodanowego można skonfigurować na dwa sposoby, przy użyciu procedury składowanej `sp_configure` i na ekranie *Server Properties*. Aby otworzyć ten ekran, w narzędziu SQL Server Management Studio kliknij prawym przyciskiem myszy silnik bazodanowy, który chcesz skonfigurować, a następnie wybierz opcję *Properties*. Zachowaj ostrożność przy modyfikowaniu konfiguracji używanego egzemplarza. Zmiana niektórych ustawień może wpływać na jego wydajność lub bezpieczeństwo. W tym punkcie omawiamy wybrane ważne ustawienia. Opis innych opcji znajdziesz w innych miejscach książki.

Stosowanie ekranu Server Properties

Korzystanie z ekranu *Server Properties* jest znacznie wygodniejsze niż używanie procedury `sp_configure`, jednak oferuje mniej możliwości. W następnych punktach opisane zostały wszystkie zakładki tego ekranu.

Zakładka General

Zakładka *General* ekranu *Server Properties* wyświetla informacje na temat egzemplarza systemu SQL Server, których nie można zmienić. Są to np. zainstalowana wersja systemu SQL Server i informacja o tym, czy dany egzemplarz działa w klastrze. Dostępne są też informacje o serwerze, takie jak liczba procesorów i ilość pamięci dostępnej w komputerze (patrz rysunek 4.7).



Rysunek 4.7. Zakładka General

OSTRZEŻENIE Gdy serwer ma np. 128 gigabajtów pamięci RAM, nie oznacza to, że cała ta ilość jest dostępna dla systemu SQL Server. System operacyjny i inne procesy działające na serwerze także używają dostępnej pamięci.

Zakładka Memory

W zakładce *Memory* na ekranie *Server Properties* możesz sprawdzić, ile pamięci według konfiguracji ma wykorzystywać system SQL Server. Domyślnie system używa tyle pamięci, ile umożliwiają mu system operacyjny i dana edycja systemu. Zwykle warto ustawić ilości pamięci minimalną i maksymalną, używane przez egzemplarze w środowisku.

Zakładka Processors

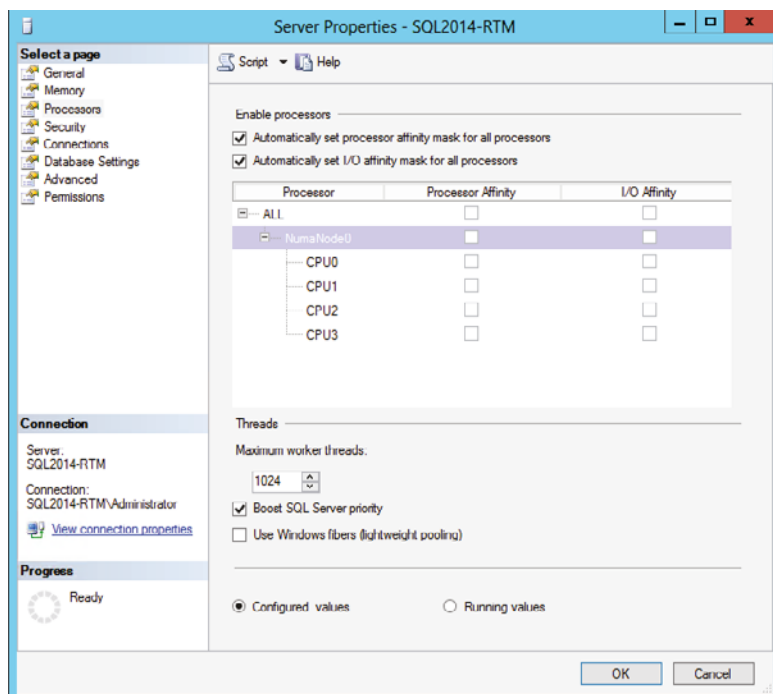
W zakładce *Processors* możesz tak ustawić silnik systemu SQL Server, aby korzystał z procesorów o określonych nazwach. Możesz też przypisać wybrane lub wszystkie procesory do operacji wejścia-wyjścia lub wątków. Jest to przydatne, jeśli używanych jest wiele procesorów i więcej

niż jeden egzemplarz systemu SQL Server. Jeśli skonfigurujesz koligację procesorów dla egzemplarzy, jeden egzemplarz może korzystać z czterech określonych procesorów, a drugi — z czterech innych procesorów. W pewnych sytuacjach, gdy obsługiwana jest duża liczba równoległych połączeń z systemem SQL Server, przydatne jest ustawienie maksymalnej liczby wątków roboczych (opcja *Maximum Worker Threads*). Ustaw ją na 0 (jest to wartość domyślna), aby system SQL Server automatycznie i dynamicznie określał optymalną liczbę wątków uruchamianych na jednym procesorze. Wątki mogą obsługiwać połączenia i inne funkcje systemowe, np. wykonywać operację CHECKPOINT. Zwykle pozostawienie tego ustawienia bez zmian pozwala na uzyskanie optymalnej wydajności.

Możesz zaznaczyć opcję *SQL Server Priority*, aby wymusić na systemie Windows przypisanie wyższego priorytetu procesowi systemu SQL Server. Myśl o zmianie tego ustawienia jest zapewne kusząca, ale należy to robić tylko po dokładnych testach, ponieważ efektem może być załadowanie innych wątków systemowych. Zwykle nie zaleca się stosowania tej opcji, chyba że bezpośrednio proponuje to pracownik pomocy technicznej.

Opcję *Lightweight Pooling* (czyli pulę lekkich wątków) stosuj tylko w tych rzadkich sytuacjach, gdy procesor jest mocno obciążony i następuje bardzo częste przełączanie kontekstu. W rozdziale 12., „Monitorowanie systemu SQL Server”, znajdziesz więcej informacji na temat puli lekkich wątków i działania tego mechanizmu.

Zwróć uwagę na widoczne na rysunku 4.8 przyciski opcji *Configured values* i *Running values* w dolnej części ekranu *Properties*. Gdy wybierzesz opcję *Running values*, zobaczysz wartości używane obecnie przez system SQL Server. Opcja *Configured values* spowoduje wyświetlenie wartości, które system zastosuje po ponownym uruchomieniu. Dzieje się tak, ponieważ niektóre wartości nie są wprowadzane do momentu ponownego rozruchu systemu.



Rysunek 4.8. Ustawienia związane z procesorami

Zakładka Security

W zakładce *Security* można określić, czy system SQL Server ma akceptować połączenia z uwierzytelnianiem systemów SQL Server i Windows, czy tylko z uwierzytelnianiem systemu Windows. Ustawienie zwykle konfiguruje się w trakcie instalowania egzemplarza, a na tym ekranie można je zmienić. W sekcji *Login Auditing* zawsze powinna być zaznaczona przynajmniej opcja *Failed Logins Only*. Wtedy system SQL Server może wykrywać, że ktoś błędnie wpisał hasło lub próbuje włamać się do danego egzemplarza. Więcej informacji o innych ustawieniach zabezpieczeń z tej zakładki zamieszczono w rozdziale 8.

Zakładka Connections

W zakładce *Connections* możesz zmodyfikować domyślne ustawienia połączeń. Jedną z opcji, którą możesz zechcieć tu ustawić, jest *NO COUNT*. Zapobiega ona wysyłaniu do klientów komunikatów w postaci (*8 Rows Affected*), jeśli dany klient ich nie zażądał. Często można wywołać polecenie *SET NOCOUNT* języka T-SQL w procedurach lub kwerendach, aby uzyskać ten sam efekt. Pozwala to uzyskać drobną poprawę wydajności, ponieważ wspomniane komunikaty są dodatkowym zbiorem rekordów wysyłanym przez system SQL Server i mogą powodować niepotrzebny transfer danych.

Zakładka Database Setting

W zakładce *Database Settings* prawdopodobnie nie będziesz wprowadzał wielu zmian. Zalecane jest pozostawienie domyślnego współczynnika wypełnienia dla indeksów równego 0. Możesz zmienić współczynniki wypełnienia dla wybranych indeksów, jednak zwykle nie należy modyfikować wartości domyślnej tego współczynnika. Omawiana zakładka pozwala też ustawić czas oczekiwania na taśmę przy tworzeniu kopii zapasowej (ile czasu system SQL Server ma oczekiwać na taśmę) i czas przechowywania kopii zapasowych przed ich wygaśnięciem (liczony w dniach domyślny czas przechowywania nośnika z kopią zapasową). Możesz dostosować te ustawienia do swoich planów. Ustal, czy chcesz kompresować kopie zapasowe baz, a następnie odpowiednio ustaw opcję *Compress Backup* (jest to zalecane rozwiązanie). Szczegóły dotyczące ustawień z tej zakładki omawiamy w rozdziale 17., „Kopie zapasowe i przywracanie stanu”.

Ostatnie ustawienie na tej stronie pozwala wskazać domyślne lokalizacje plików bazy danych i dziennika. Ustawienie domyślne używane w trakcie instalacji powoduje umieszczenie plików danych i plików dziennika na tym samym dysku w katalogach *%SystemDrive%\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\Data* i *Log*. Jeśli nie instalujesz systemu SQL Server tylko do nauki, nie umieszczaj plików danych i plików dziennika na dysku C:\.

Zakładka Advanced

W zakładce *Advanced* można włączyć dwie funkcje — niezależne bazy danych i *FILESTREAM*. Ponadto znajdziesz tu wiele różnych ustawień. W większości sytuacji należy pozostawić ich wartości domyślne. W opcjach sieciowych możesz ustawić limit czasu dla zdalnego logowania (w sekundach), który umożliwia kontrolowanie, przez jaki czas zdalni użytkownicy mogą próbować nawiązywania połączeń. Ponadto można skonfigurować ustawienia związane z współbieżnością. Maksymalny poziom współbieżności (opcja *Max Degree of Parallelism* — *MAXDOP*) równy 0 oznacza, że wszystkie procesory będą obsługiwać równoległe kwerendy.

Wartość ustawienia *Cost Threshold for Parallelism* można podnieść do 20. Jest ona używana przy określaniu, czy optymalizator kwerend powinien zastosować równoległy plan wykonania kwerendy. Pierwotna wartość tego ustawienia to 5. Jest ona stosowana od wersji SQL Server 7.0. Była to wartość kosztu kwerendy uzyskana przez członka zespołu pracującego nad optymalizatorem

kwerend. Gdy na komputerze tej osoby uzyskano wynik 5, plan należało wykonać równolegle. Optymalizator zastosuje plan równoległy, gdy wartość MAXDOP jest różna od 1, koszt planu równoległego plus koszt przekazywania iteratorów jest niższy niż koszt planu sekwencyjnego oraz koszt planu przekracza wartość ustawienia *Cost Threshold for Parallelism*.

Zakładka Permissions

Zakładka *Permissions* wyświetla wszystkie loginy i role dostępne w danym egzemplarzu. W tym miejscu, w zakładce *Explicit* możesz bezpośrednio przyznawać i wycofywać bardzo szczegółowe uprawnienia dla poszczególnych loginów i ról. Ponadto zakładka *Effective* pozwala zobaczyć, które uprawnienia są obecnie przyznane dla loginów i ról.

Używanie procedury sp_configure

Procedura składowna `sp_configure` umożliwia zmianę wielu opcji konfiguracyjnych systemu SQL Server. Poniżej wymieniono wybrane, często modyfikowane ustawienia.

- **Max degree of parallelism.** W środowiskach OLTP zwykle ustawia się tę opcję na liczbę dostępnych gniazd.
- **CLR enabled.** Umożliwia wykonywanie procedur CLR.
- **Blocked processes threshold.** Pozwala ustawić limit czasu, po przekroczeniu którego generowane są raporty na temat zablokowanych procesów.

Gdy uruchomisz procedurę `sp_configure` bez parametrów, zobaczysz dostępne opcje i ich obecne wartości. Domyślnie zwracane są tylko podstawowe i często używane ustawienia (jest ich 15). Aby wyświetlić dodatkowe opcje, musisz odpowiednio skonfigurować egzemplarz. W tym celu wywołaj procedurę `sp_configure` w następujący sposób:

```
sp_configure 'show advanced options', 1;
RECONFIGURE;
```

Zmiana nie zostanie wprowadzona do momentu wywołania polecenia `RECONFIGURE`. System SQL Server przy wykonywaniu procedury `sp_configure` wykrywa nieprawidłowe i niezalecane ustawienia. Jeśli podałeś wartość, która nie przechodzi jednego z testów, system SQL Server wyświetla komunikat ostrzegawczy w następującej postaci:

```
Msg 5807, Level 16, State 1, Line 1
Recovery intervals above 60 minutes not recommended. Use the RECONFIGURE WITH
OVERRIDE statement to force this configuration.
```

OSTRZEŻENIE Wywołanie polecenia `RECONFIGURE` prowadzi do kompletnego opróżnienia pamięci podręcznej z planem wykonania. Oznacza to, że w pamięci podręcznej nie pozostają żadne dane. Dlatego wszystkie operacje są ponownie kompilowane przy pierwszym ich późniejszym wywołaniu. Zachowaj ostrożność podczas korzystania z procedury `sp_configure` w środowisku produkcyjnym i upewnij się, że rozumiesz skutki opróżnienia pamięci podręcznej w używanej bazie danych.

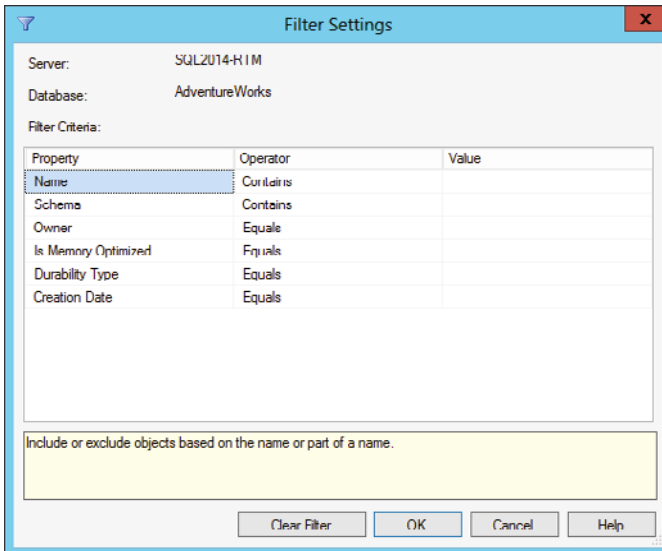
To ostrzeżenie daje możliwość ponownego przemyślenia potencjalnie błędnej decyzji — np. ustawienia większej ilości pamięci niż ilość dostępna w systemie. W poniższym kodzie pokazano, jak wymusić wprowadzenie nowych ustawień:

```
EXEC sp_configure 'recovery interval', 90;
RECONFIGURE WITH OVERRIDE;
GO
```

Filtrowanie obiektów

W narzędziu SQL Server Management Studio można filtrować obiekty w kilku prostych krokach. Jest to przydatne, gdy w systemie pojawiają się dziesiątki obiektów.

1. Zaznacz węzeł, który chcesz przefiltrować, i wybierz ikonę *Filter* w oknie *Object Explorer*.
2. Otworzy się okno dialogowe *Object Explorer Filter Settings* widoczne na rysunku 4.9. W tym miejscu możesz przefiltrować obiekty na podstawie nazw, schematów lub czasu utworzenia.



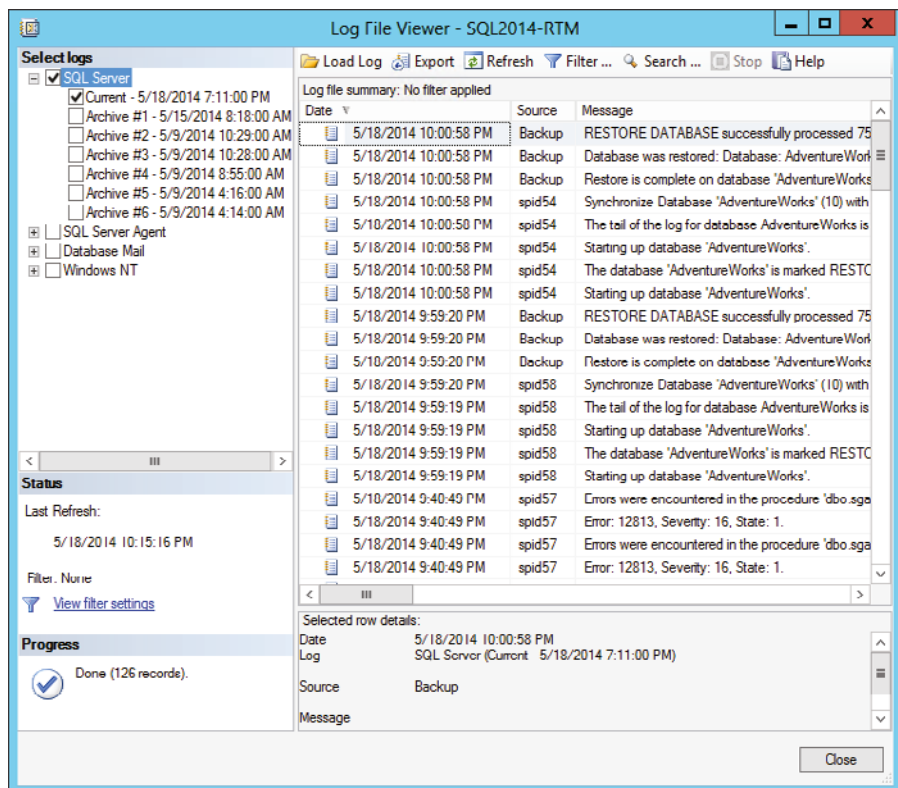
Rysunek 4.9. Okno dialogowe Object Explorer Filter Settings

3. Użyj listy rozwijanej *Operator*, aby wybrać, jak chcesz filtrować dane. Następnie wpisz odpowiednią wartość w kolumnie *Value*.

Dzienniki błędów

Prawdopodobnie wiesz już, że gdy coś nie działa w aplikacji, jednym z czynników, które trzeba sprawdzić, jest baza danych. Administrator bazy danych powinien wspomagać rozwiązywanie problemu i potwierdzić, że baza nie jest źródłem kłopotów. Pierwszą rzeczą, jaką administrator zwykle robi, jest połączenie się z serwerem i sprawdzenie dziennika błędów danego egzemplarza systemu SQL Server oraz dziennika zdarzeń systemu Windows.

W systemie SQL Server 2014 za pomocą narzędzia SQL Server Management Studio można szybko przejrzeć dzienniki w jednym miejscu. Aby wyświetlić dzienniki, kliknij węzeł *SQL Server Logs* w węźle *Management*, a następnie wybierz opcję *View/SQL Server and Windows Log*. Spowoduje to otwarcie ekranu *Log File Viewer*, na którym można zaznaczyć pliki dziennika, aby je wyświetlić. Można połączyć dane z dzienników systemu SQL Server, narzędzia Agent i dzienników zdarzeń systemu Windows, co pokazano na rysunku 4.10.



Rysunek 4.10. Wyświetlanie zawartości dzienników

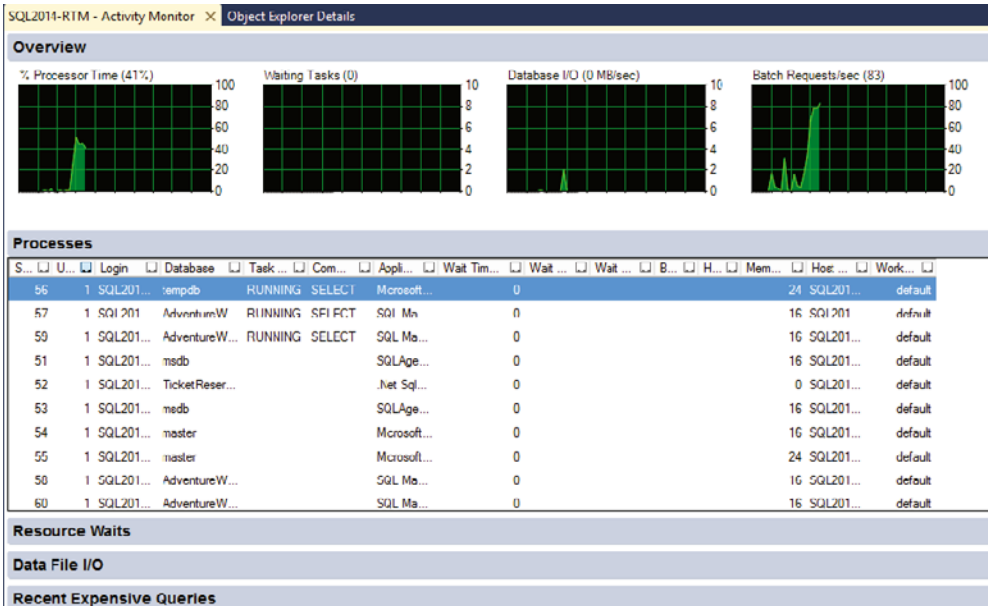
W niektórych sytuacjach warto scalić dzienniki z kilku komputerów, aby ustalić, co powoduje problemy w aplikacji. W tym celu kliknij przycisk *Load Log* w górnej części ekranu i przejdź do potrzebnego pliku z rozszerzeniem *.log*. Może to być plik z dziennikiem błędów systemu Windows zapisany w formacie *.log* lub dziennik z innego egzemplarza systemu SQL Server. Za pomocą tej metody możesz np. połączyć dzienniki z wszystkich egzemplarzy z jednego serwera, aby uzyskać ogólny przegląd problemów z danej maszyny.

Narzędzie Activity Monitor

To narzędzie wyświetla aktualne połączenia danego egzemplarza, co pozwala ustalić, czy pewne procesy nie blokują innych procesów. Aby otworzyć narzędzie Activity Monitor w programie SQL Server Management Studio, kliknij prawym przyciskiem myszy serwer w oknie *Object Explorer*, a następnie wybierz opcję *Activity Monitor*.

Activity Monitor to kompletne narzędzie; za jego pomocą można zobaczyć, kto jest połączony z komputerem i co robi. W górnej części okna widoczne są cztery wykresy (*Show Processor Time*, *Waiting Tasks*, *Database I/O* i *Batch Requests/Sec*) pełniące funkcję wskaźników wydajności serwera. Pod wykresami znajdują się cztery listy: *Processes*, *Resource Waits*, *Data File I/O* i *Recent Expensive Queries*. Dla każdej z tych list można zastosować filtry, aby uwzględnić tylko wybrane komputery, loginy lub połączenia o zużyciu zasobów wyższym niż ustawiona wartość progowa. Kliknięcie nagłówka kolumny umożliwia sortowanie danych.

W zakładce z informacjami o procesach (patrz rysunek 4.11) widoczne są wszystkie loginy połączone z danym komputerem. Loginy są identyfikowane za pomocą identyfikatorów SPID (ang. *Server Process ID*). Łatwo nie docenić bogactwa informacji dostępnych na tej stronie. Przewiń ją w prawo, aby zobaczyć wiele ważnych danych na temat każdego połączenia.



Rysunek 4.11. Informacje o procesach

W trakcie debugowania przydatne są zwykle następujące kolumny.

- **Session ID.** Jest to unikatowy numer przypisany do procesu połączonego z systemem SQL Server, nazywany także identyfikatorem SPID. Ikona obok numeru informuje, co dzieje się z połączeniem. Jeśli widoczna jest ikona klepsydry, wiadomo, że proces oczekuje na wykonanie zadania lub jest zablokowany przez inne połączenie.
- **User Process Flag.** Pozwala zidentyfikować połączone wewnętrzne procesy systemu SQL Server. Domyślnie procesy te są odfiltrowane. Aby zmienić ustawienia i wyświetlić wewnętrzne procesy systemu SQL Server, kliknij listę rozwijaną i wybierz odpowiednią wartość.
- **Login.** To login, z którym powiązany jest dany proces.
- **Database.** Jest to baza danych używana obecnie dla danego połączenia.
- **Task State.** Określa, czy użytkownik jest aktywny, czy nie.
 - **Done.** Zadanie zostało ukończone.
 - **Pending.** Proces oczekuje na wątek roboczy.
 - **Runnable.** Proces był aktywny, ma dostępne połączenie, ale nie ma żadnych zadań do wykonania.
 - **Running.** Proces obecnie wykonuje zadanie.
 - **Suspended.** Proces ma zadanie do wykonania, ale został zatrzymany. Dodatkowe informacje na temat przyczyn wstrzymania pracy znajdziesz w kolumnie *Wait Type*.

- **Command.** Wyświetla typ wykonywanego polecenia (np. SELECT, CBCC, INSERT lub AWAITING COMMAND). Ta kolumna nie pokazuje kwerendy wykonywanej przez użytkownika, ale określa, jakiego rodzaju operacje są wykonywane na serwerze. Aby zobaczyć pierwotne polecenie, zaznacz wiersz omawianej tabeli, kliknij go prawym przyciskiem myszy i wybierz opcję *Details*.
- **Application.** Wyświetla aplikację połączoną z egzemplarzem systemu. Może być ona określona przez programistę w łańcuchu znaków połączenia.
- **Wait Time (ms).** Jeśli proces jest zablokowany lub oczekuje na zakończenie innego procesu, ta kolumna określa, jak długo trwa oczekiwanie (w milisekundach). Jeżeli proces nie oczekuje, wyświetlana wartość to 0.
- **Wait Type.** Określa rodzaj zdarzenia, na jakie proces oczekuje.
- **Wait Resource.** Przedstawia tekstową reprezentację zasobu, na który proces oczekuje.
- **Blocked By.** Określa identyfikator SPID blokujący dane połączenie.
- **Head Blocker.** Wartość 1 oznacza, że identyfikator z kolumny *Blocked By* należy do procesu z początku łańcucha blokujących procesów. Dla pozostałych procesów wyświetlana jest wartość 0.
- **Memory Use (KB).** Ilość pamięci wykorzystywana w danym momencie przez połączenie. Jest ona reprezentowana przez liczbę kilobajtów w pamięci podręcznej powiązanych z danym połączeniem. W wersjach starszych niż SQL Server 2008 wartość ta była podawana w stronach.
- **Host.** Określa, na jakiej stacji roboczej lub na jakim serwerze pracuje jednostka o danym loginie. Jest to przydatna informacja, gdy jednak z systemem SQL Server łączy się serwer WWW, dane te są mniej istotne.
- **Workload Group.** Określa nazwę powiązanej z kwerendą grupy procesów pracy funkcji Resource Governor.

UWAGA Jeśli klikniesz proces prawym przyciskiem myszy, będziesz mógł wyświetlić ostatnią kwerendę uruchomioną za pomocą danego połączenia, obserwować proces przy użyciu profilera lub zamknąć połączenie. Możesz też kliknąć prawym przyciskiem myszy wykres i wybrać opcję *Refresh*, aby ręcznie odświeżyć dane, lub opcję *Refresh Interval*, by zobaczyć, jak często narzędzie Activity Monitor odświeża informacje. Domyślnie są one odświeżane co 10 sekund. Unikaj zbyt częstego odświeżania, ponieważ powoduje to nieustanne zgłaszanie kwerend do serwera, co może obniżyć wydajność systemu.

Inne ważne zadanie, które można wykonać za pomocą narzędzia Activity Monitor, to wykrywanie warunków powodujących blokadę. W poniższych punktach pokazano, jak wywołać blokadę transakcji i jak rozwiązać problem z wykorzystaniem opisywanego narzędzia.

1. Uruchom poniższą kwerendę w oknie kwerendy przy aktywnym połączeniu z bazą AdventureWorks (przed wykonaniem opisanych kroków utwórz kopię zapasową tej bazy):

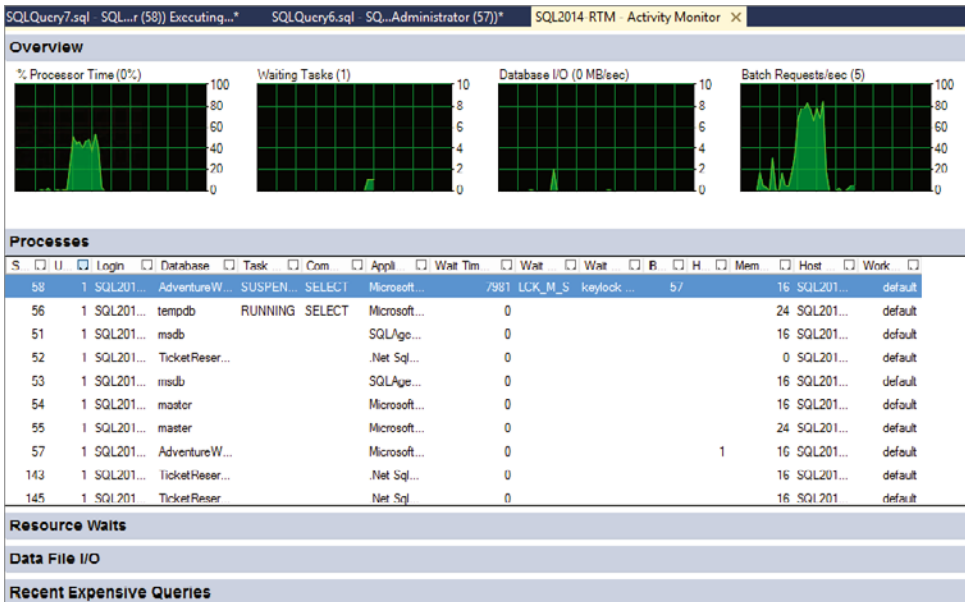
```
BEGIN TRAN
DELETE FROM Production.ProductCostHistory
WHERE ProductID = 707;
```

Kwerenda ta celowo nie jest zatwierdzana. Występuje tu polecenie BEGIN TRAN, ale bez odpowiadających mu instrukcji ROLLBACK lub COMMIT. To oznacza, że wiersze usuwane z tabeli Production.ProductCostHistory są zablokowane na wyłączność.

- Następnie bez zamykania pierwszego okna otwórz nowe okno kwerendy i uruchom następujące polecenie:

```
SELECT * FROM Production.ProductCostHistory;
```

Ta kwerenda zostanie wstrzymana w oczekiwaniu na wykonanie instrukcji DELETE na tabeli `Production.ProductCostHistory`, ponieważ pierwsza transakcja blokuje pierwszy wiersz. Nie zamykaj żadnego z okien. W górnej części każdego okna kwerendy widoczny jest identyfikator sesji, a na dole okna znajdziesz używany login. Jeśli nie widzisz identyfikatora SPID na górnym pasku okna, umieść nad paskiem kursor myszy, a pojawi się okienko z całym nagłówkiem obejmującym potrzebny identyfikator. Przy otwartych oknach kwerend przejdź do narzędzia Activity Monitor, aby sprawdzić, jak wyglądają w nim używane połączenia (patrz rysunek 4.12).



Rysunek 4.12. Sprawdzanie połączeń w narzędziu Activity Monitor

- Otwórz narzędzie Activity Monitor. Zauważ, że stan zadania dla jednego z połączeń to *Suspended*. Dotyczy on kwerendy, która próbuje wykonać instrukcję SELECT. Aby się o tym upewnić, porównaj identyfikator zawieszonoj sesji z identyfikatorem sesji wywołującej tę instrukcję. Powód oczekiwania to *LCK_M_S*, co oznacza, że sesja oczekuje na współużytkowaną blokadę w trybie do odczytu. Gdy umieścisz kursor nad wartością w kolumnie *Wait Resource*, zobaczysz szczegółowe informacje na temat blokady, w tym identyfikatory *object_ID* zasobów. W kolumnie *Blocked By* widoczny jest identyfikator sesji blokującego procesu. Powinien on pasować do identyfikatora SPID pierwszej kwerendy. Masz możliwość zamknięcia blokującego procesu. W tym celu zaznacz wiersz blokującego procesu, kliknij go prawym przyciskiem myszy i wybierz opcję *Kill Process*.
- Kiedy jeszcze blokada istnieje, przyjrzyj się standardowemu raportowi o niej dostępnemu w systemie SQL Server 2014. W oknie *Object Explorer* kliknij nazwę serwera prawym przyciskiem myszy i wybierz opcję *Reports/Standard Reports/Activity — All Blocking Transactions*. Pojawi się raport widoczny na rysunku 4.13.

Activity - All Blo...SQL2014-RTM X SQL Query (sql - SQL... (58)) Executing... SQLQuery6.sql - SQL...Administrator (D/J)* SQL2014-RTM - Activity Monitor

Activity - All Blocking Transactions

Microsoft SQL Server 2014

on SQL2014-RTM at 5/18/2014 10:23:37 PM

This report provides information about each transaction on the instance which is blocking one or more other transactions.

All Blocking Transactions

The description of transactions which are blocking other transactions.

Transaction ID	# Directly Blocked Transactions	# Indirectly Blocked Transactions	Transaction Name	State	Transaction Type	Start Time	Resource Type	Session ID	Blocking SQL Statement
2200041	1	0	user transaction	Active	Full Transaction	5/18/2014 10:23:11 PM	KEY	57	RL ..
Direct/Indirect	Blocked Transaction ID	Blocked Transaction Name	State	Transaction Type	Start Time	Resource Type	Session ID	Blocked SQL Statement	
Direct	2200041	SELECT	Active	Read Only	5/18/2014 10:23:12 PM	KEY	50	RL select * from production.ProductCostHistory	

* The transactions span multiple databases on the same instance.

Rysunek 4.13. Raport o blokadach

5. Następnie w narzędziu Activity Monitor przenieś kursor myszy nad kolumnę *Wait Resource*. Zobaczysz informację o trybie — *mode=X*. Tryb ten oznacza, że zasób jest zablokowany na wyłączność. *Blokada na wyłączność* powoduje, że żadna inna jednostka nie może odczytać danych. Tryb *S* oznacza, że system SQL Server przyznał blokadę współużytkowaną. W większości sytuacji jest ona nieszkodliwa i umożliwia innym jednostkom dostęp do tych samych danych. Aby zażądać odczytu niezatwierdzonych danych, wywołaj poniższą instrukcję, która zmienia poziom izolacji transakcji i umożliwia odczyt takich danych. *Odczyt niezatwierdzonych danych* (ang. *dirty read*) umożliwia wczytanie danych w niespójnym stanie, ponieważ transakcja nie jest ani zatwierdzona, ani wycofana. Oto potrzebne polecenie:

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
```

```
SELECT *
FROM Production.ProductCostHistory;
```

OSTRZEŻENIE W środowisku produkcyjnym dotycząca kwerend wskazówka (NOLOCK) powinna być stosowana rzadko (lub wcale). Obecnie aby można było odczytać niezatwierdzone dane, należy zmienić poziom izolacji transakcji. Od biznesowych zastosowań danych zależy, czy odczyt takich danych jest dopuszczalny. Gdy odczyt niezatwierdzonych danych jest niezbędny, zamiast stosować wskazówkę (NOLOCK), można ustawić poziom izolacji transakcji na SNAPSHOT. Pozwala to na kontrolę wersji rekordów.

6. Przedstawiona kwerenda zwróci dane. Zanim przejdziesz do dalszych punktów, wywołaj poniższą instrukcję SQL-a w oknie kwerendy z instrukcją DELETE:

```
ROLLBACK TRAN;
```

Monitorowanie procesów w języku T-SQL

Aktywność serwera można też śledzić za pomocą języka T-SQL. Administratorzy baz danych wolą stosować tę metodę do rozwiązywania problemów z powolnie działającymi kwerendami lub reagowania na skargi użytkowników dotyczące niskiej wydajności aplikacji. Używanie języka T-SQL jest preferowane, ponieważ daje on większą swobodę w zdobywaniu informacji niż narzędzie Activity Monitor.

Procedury `sp_who` i `sp_who2`

Procedura `sp_who` zwraca informacje o tym, kto jest połączony z danym egzemplarzem systemu (podobne dane udostępnia Activity Monitor). Prawdopodobnie będziesz jednak wolał używać nieudokumentowanej procedury składowanej `sp_who2`, która zwraca bogatsze informacje na temat każdego procesu. Obie te procedury przyjmują te same parametry. Tu szczegółowo omawiamy procedurę `sp_who2`. Pamiętaj, że procedura `sp_who` wyświetla mniej informacji.

UWAGA Doświadczeni administratorzy zamiast procedury `sp_who2` stosują narzędzie o nazwie `sp_whoisactive`. Ta popularna procedura została napisana przez Adama Machanica, posiadacza tytułu SQL Server MVP i niezależnego konsultanta z okolic Bostonu. Jeśli nie znasz tego narzędzia, pobierz je i zapoznaj się z wpisami z bloga Adama, aby się dowiedzieć, w jaki sposób możesz znacznie usprawnić swoją pracę w zakresie monitorowania systemu i rozwiązywania problemów. Plik do pobrania i instrukcje dotyczące instalacji znajdują się na stronie <http://tinyurl.com/WholsActive>. Warto też zapoznać się z dostępnymi na blogu Adama artykułami z serii „A Month of Activity Monitoring” (<http://tinyurl.com/WholsActiveDocs>). Dowiesz się z nich, dlaczego powinieneś jak najszybciej zacząć używać procedury `sp_whoisactive`. Narzędzie to jest bezpłatnie dostępne dla społeczności użytkowników (choć obowiązują przy tym pewne wyjątki) i można je pobrać ze strony <http://bit.ly/WholsActiveLicensing>.

Aby wyświetlić wszystkie połączenia z serwerem, uruchom procedurę `sp_who2` bez żadnych parametrów. Zobaczysz wtedy informacje tego samego rodzaju, co w narzędziu Activity Monitor. Możesz też przekazać parametr `'active'`, aby wyświetlić tylko aktywne połączenia z serwerem:

```
sp_who2 'active';
```

Ponadto można przekazać identyfikator SPID, co pokazano poniżej, aby zobaczyć szczegółowe informacje na temat poszczególnych procesów:

```
sp_who2 55;
```

UWAGA Choć w tym przykładzie używany jest identyfikator SPID 55, identyfikator procesu na Twoim komputerze może być inny. Znajdziesz go za pomocą opisanego wcześniej narzędzia Activity Monitor.

Widok `sys.dm_exec_connections`

Widok DMV (ang. *Dynamic Management View*) `sys.dm_exec_connections` udostępnia jeszcze więcej informacji, które pomagają rozwiązywać problemy z silnikiem bazodanowym systemu SQL Server. Widok ten zwraca wiersz dla każdej sesji z systemu SQL Server. Ponieważ jest to widok DMV, wyświetlany jest jako tabela i umożliwia pisanie złożonych kwerend odfiltrowujących niepotrzebne informacje. Tak działa poniższa kwerenda wyświetlająca wyłącznie te połączenia, które wykonały operację zapisu:

```
SELECT * FROM
    sys.dm_exec_sessions
WHERE is_user_process = 1
AND writes > 0;
```

Oprócz informacji wyświetlanych przez opisane wcześniej techniki, ten widok DMV określa, ile wierszy użytkownik pobrał od czasu otwarcia połączenia, a także ile wykonał operacji odczytu, zapisu i odczytu logicznego. Ten widok możesz wyświetlić w ustawieniach każdego połączenia; możesz też sprawdzić czas ostatniego wystąpienia błędu.

Funkcja sys.dm_exec_sql_text

Za pomocą funkcji DMF (ang. *Dynamic Management Function*) `sys.dm_exec_sql_text` możesz pobrać tekst wybranej kwerendy. Tę funkcję można wykorzystać razem z widokiem DMV `sys.dm_exec_query_stats`, aby pobrać np. 10 najmniej wydajnych kwerend z wszystkich baz danych. Kod z listingu 4.1 (plik *10NajwolniejszychKwerend.sql*) pobiera liczbę wykonań kwerendy, średni czas jej przetwarzania (czas pracy procesora i łączny) i tekst kwerendy.

Listing 4.1. Plik 10NajwolniejszychKwerend.sql

```
SELECT TOP 10 execution_count as [Liczba wykonań],
    total_worker_time/execution_count as [Średni czas pracy procesora],
    Total_Elapsed_Time/execution_count as [Średni czas łącznie],
    (
        SELECT SUBSTRING(text,statement_start_offset/2,
            (CASE WHEN statement_end_offset = -1
                THEN LEN(CONVERT(nvarchar(max), [text]))* 2
            ELSE statement_end_offset END - statement_start_offset) /2)
        FROM sys.dm_exec_sql_text(sql_handle)
    )as query_text
FROM sys.dm_exec_query_stats
ORDER BY [Average CPU Time] DESC;
```

Widok DMV `sys.dm_exec_query_stats` wyświetla też wiele innych przydatnych informacji. Pokazuje np. wiersz dla każdego uruchomionego planu wykonania kwerendy. Możesz pobrać kolumnę `sql_handle` z tego widoku i zastosować ją w funkcji `sys.dm_exec_sql_text`. Ponieważ ten widok działa na poziomie planów wykonania, po zmianie tekstu kwerendy nowa kwerenda będzie widoczna jako nowy wiersz.

To tylko kilka przykładów monitorowania procesów za pomocą języka T-SQL. Po wprowadzeniu i rozpowszechnieniu obiektów DMO masz teraz więcej możliwości do wyboru, gdy chcesz śledzić problemy z bazą za pomocą języka T-SQL.

Zarządzanie wieloma serwerami

Żyjemy w epoce, w której ilość danych nieustannie rośnie. Wynika z tego wzrost liczby baz danych, którymi trzeba zarządzać. Administratorzy baz z systemu SQL Server często opiekują się licznymi egzemplarzami tego systemu i bazami. Operacja tak prosta jak dodanie nowego loginu lub ustawienie opcji serwera może stać się prawdziwym wyzwaniem, jeśli trzeba ją przeprowadzić na dziesiątkach serwerów. Na szczęście Microsoft udostępnił zestaw nowych narzędzi ułatwiających pracę administratorom baz danych.

UWAGA Jeśli jesteś jednym z coraz większej liczby administratorów baz danych, którzy zarządzają wieloma serwerami, poświęć czas na zapoznanie się z powłoką PowerShell systemu Windows. Język używany w tej powłoce to rozbudowane narzędzie pomocne przy tworzeniu skryptów wielokrotnego użytku przeznaczonych do spójnego wykonywania tych samych zadań w różnych serwerach, egzemplarzach lub bazach danych.

Centralne serwery zarządzania i grupy serwerów

W systemie SQL Server 2008 wprowadzono funkcje ułatwiające pracę administratorom, takie jak centralne serwery zarządzania i grupy serwerów. Mechanizmy te umożliwiają uruchamianie skryptów w języku T-SQL dla grup serwerów i stosowanie dla takich grup zasad PBM (ang. *Policy-Based Management*). Możesz wykonać kod w języku T-SQL na grupie serwerów i połączyć wyniki w jeden zbiór lub zachować odrębne zbiory wyników. Przy łączeniu zbiorów wyników można dodać kolumnę z informacją, z których serwerów pochodzą poszczególne wiersze. Za pomocą tego narzędzia można administrować wieloma serwerami i wykonywać wiele innych zadań. Jeśli każdy serwer udostępnia tabelę błędów, możesz w jednej instrukcji pobrać dane z wszystkich tych tabel.

Omawiane funkcje są dostępne w narzędziu SQL Server Management Studio, jednak zanim zaczniesz z nich korzystać, musisz najpierw zarejestrować centralny serwer zarządzania. Kliknij prawym przyciskiem myszy węzeł *Central Management Servers* w oknie dialogowym *Registered Servers* narzędzia SQL Server Management Studio i wybierz opcję *Register Central Management Server*. W oknie dialogowym, które się pojawi, możesz wybrać serwer z konfiguracją. Serwer będzie przechowywał metadane i wykonywał niektóre zadania pomocnicze.

Po ustawieniu centralnego serwera zarządzania możesz tworzyć grupy serwerów i rejestrować serwery w grupach. Po skonfigurowaniu serwerów kliknij prawym przyciskiem myszy dowolny obiekt (serwer rejestracji, grupy serwerów lub poszczególne serwery w grupie) i wybierz opcję *New Query*, *Object Explorer* lub *Run Policy*. Następnie wybierz opcję *New Query*, dodaj kod w języku T-SQL i uruchom kwerendę dla wszystkich serwerów z grupy.

UWAGA W grupach umieszczaj serwery, którymi chcesz jednocześnie administrować. Kilka możliwości to: pogrupowanie ich według środowiska (np. programistyczne, kontrola jakości i produkcyjne) lub według zagadnień (np. dział kadr, rachunkowość i dział operacyjny). Zastanów się, jakie zadania administracyjne zwykle wykonujesz jednocześnie, i wykorzystaj te informacje przy tworzeniu grup.

Oto kilka informacji dotyczących serwera rejestracji.

- Taki serwer nie może należeć do grupy, dla której jest już używany jako serwer konfiguracji.
- Wszystkie kwerendy są wykonywane za pomocą zaufanych połączeń.
- Jeśli serwery działają w różnych domenach, trzeba utworzyć dla nich zaufane relacje.
- Możesz skonfigurować wiele centralnych serwerów zarządzania.

UWAGA Aby ustawić opcje dotyczące zbiorów wyników z wielu serwerów, wybierz opcję *Tools/Options/Query Results/SQL Server/Multi-server results*.

Opcje śledzenia

Opcje śledzenia to zaawansowany mechanizm pozwalający na uzyskanie dostępu do ukrytych funkcji systemu SQL Server i technik rozwiązywania problemów. Czasem umożliwiają one zmianę zalecanego działania systemu SQL Server i włączenie takich funkcji jak obsługa dysków sieciowych dla plików baz danych. W innych sytuacjach za pomocą opcji śledzenia można włączyć dodatkowe monitorowanie. Istnieje zestaw opcji (w tym opcja 1204), które pomagają diagnozować zakleszczenie. Aby włączyć opcję śledzenia, zastosuj polecenie `DBCC TRACEON` i podaj po nim potrzebną opcję:

`DBCC TRACEON (1204)`

W celu wyłączenia opcji wywołaj polecenie `DBCC TRACEOFF`. Należy po nim podać wyłączane opcje (jeśli jest ich kilka, trzeba je rozdzielić przecinkami):

`DBCC TRACEOFF (1204, 3625)`

UWAGA Opcja 3625 użyta w przykładowym kodzie ogranicza zakres informacji zwracanych użytkownikom, którzy nie mają przypisanej roli administratora systemu. Efekt ten uzyskuje się dzięki ukryciu parametrów w niektórych komunikatach o błędach. Opcję tę można zastosować w celu zwiększenia bezpieczeństwa.

Gdy włączasz opcję śledzenia, domyślnie jest ona aktywowana dla jednego połączenia. Przykładowo po włączeniu opcji 1224 (wyłącza ona eskalowanie blokad w zależności od ich liczby) eskalowanie zostanie wyłączone tylko dla połączenia, w którym wywołano polecenie `DBCC TRACEON`. Można też włączyć opcję na poziomie serwera. W tym celu należy wywołać instrukcję z ustawieniem -1:

`DBCC TRACEON (1224, -1)`

Po włączeniu opcji śledzenia prawdopodobnie zechcesz sprawdzić, czy śledzenie jest aktywne. Służy do tego polecenie `DBCC TRACESTATUS`. Za pomocą tego polecenia można np. sprawdzić, czy działa konkretny mechanizm śledzenia:

`DBCC TRACESTATUS (3635)`

Jeśli dany mechanizm śledzenia nie jest włączony, instrukcja ta zwróci następujące dane:

TraceFlag	Status	Global	Session

3625	0	0	0

(1 row(s)affected)

Jeżeli chcesz wyświetlić wszystkie ślady powiązane z połączeniem, uruchom omawiane polecenie z parametrem -1:

`DBCC TRACESTATUS (-1)`

Poniższe wyniki zwrócone przez tę kwerendę pokazują, że włączone są dwie opcje śledzenia. Opcja 1224 jest aktywna globalnie dla każdego połączenia z danego systemu SQL Server, a opcja 3625 jest włączona dla danej sesji:

```
TraceFlag Status Global Session
-----
1224      1      1      0
3625      1      0      1
```

(2 row(s) affected)

Jeśli żadne opcje śledzenia nie są włączone, pojawi się następujący komunikat:

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

W egzemplarzach systemu SQL Server nie należy w nieskończoność pozostawiać włączonych opcji śledzenia, chyba że zasugerował to pracownik pomocy technicznej Microsoftu. Gdy takie opcje są włączone przez cały czas, mogą spowodować nieoczekiwane działanie egzemplarza systemu. Ponadto obecnie używane opcje mogą nie być dostępne w przyszłych wersjach systemu SQL Server lub pakietach poprawek. Jeśli pracujesz w trybie diagnostycznym, możesz włączyć opcję śledzenia w wierszu poleceń przy uruchamianiu systemu SQL Server. Wcześniej w tym rozdziale wspomniano także, że można uruchomić śledzenie przy rozruchu systemu SQL Server za pomocą programu `sqlservr.exe` z opcją `-T`. Więcej informacji na temat opcji śledzenia podano w rozdziale 11., „Optymalizowanie systemu SQL Server 2014”.

Opcje śledzenia to obszerny temat. Choć w tym miejscu opisano tylko kilka z nich, dalej w tej książce znajdziesz liczne przykłady stosowania takich opcji w praktyce.

UWAGA Niektóre mechanizmy dostępne za pomocą opcji śledzenia (np. monitorowanie zakleszczenia) można zastosować w bardziej wydajny sposób przy użyciu zdarzeń rozszerzonych. Kompletne omówienie implementowania zdarzeń rozszerzonych w środowisku znajdziesz w rozdziale 12., „Monitorowanie systemu SQL Server”.

Udzielanie pomocy przez obsługę techniczną

Gdy nie potrafisz rozwiązać problemu z systemem SQL Server, zwykle możesz zadzwonić do pracownika pomocy technicznej z określonego poziomu. Niezależnie od tego, czy jest to poziom Microsoftu, czy sprzedawcy systemu, dostępnych jest wiele narzędzi do komunikowania się z ich przedstawicielami. Aby ułatwić sobie komunikację z pracownikami pomocy technicznej i precyzyjnie przedstawić środowisko oraz trudności, zastosuj przy odtwarzaniu błędu programy `SQLDumper.exe` i `SQLDiag.exe`.

Narzędzie SQLDumper.exe

Narzędzie `SQLDumper.exe` jest dostępne od wersji SQL Server 2000 SP3 i umożliwia systemowi SQL Server wykonanie zrzutu środowiska po wystąpieniu wyjątku. Firma zajmująca się pomocą techniczną (np. Microsoft) może zażądać użycia tego programu, gdy natrafisz na problem (kiedy przykładowo serwer się zawiesi).

Jeśli chcesz na żądanie utworzyć plik ze zrzutem środowiska, musisz ustalić w systemie Windows identyfikator procesu egzemplarza systemu SQL Server. Możesz w tym celu uruchomić menedżer zadań i sprawdzić wpis dotyczący serwera SQL Server lub otworzyć program SQL Server Configuration Manager opisany wcześniej w tym rozdziale. Na stronie z usługami systemu SQL Server 2014 w programie Configuration Manager znajdziesz wszystkie usługi tego systemu i identyfikatory procesów.

Program *SQLDumper.exe* domyślnie znajduje się w katalogu *C:\Program Files\Microsoft SQL Server\120\Shared*, ponieważ jest współużytkowany przez wszystkie egzemplarze systemu SQL Server zainstalowane na serwerze.

Jednak może to być także inny katalog. Zależy to od miejsca, w którym zainstalowane są narzędzia systemu SQL Server. Aby utworzyć plik zrzutu dla pomocy technicznej, przejdź w wierszu poleceń do katalogu *C:\Program Files\Microsoft SQL Server\120\Shared*. Podobnie jak w wielu innych narzędziach uruchamianych w wierszu poleceń, dostępne opcje można wyświetlić za pomocą pokazanej poniżej instrukcji. Dodatkowe informacje znajdziesz w dokumentacji systemu SQL Server.

```
SQLdumper.exe -?
```

W wierszu poleceń można utworzyć pełny zrzut lub mały zrzut. Mały zrzut zajmuje mniej niż megabajt, natomiast pełny może zajmować nawet 110 megabajtów. Aby utworzyć pełny zrzut, wywołaj następujące polecenie:

```
SqlDumper.exe <IDProcesu> 0 0x1100
```

Człon *<IDProcesu>* to identyfikator procesu danego egzemplarza systemu SQL Server. Instrukcja zapisuje pełny zrzut w aktywnym katalogu. Nazwa pliku ze zrzutem dla pierwszego uruchomienia programu *SQLDumper.exe* to *SQLDmpr0001.mdmp*. Plików ze zrzutami nie można otwierać w edytorach tekstu (np. w Notatniku). Potrzebne są zaawansowane narzędzia do rozwiązywania problemów, takie jak środowisko Visual Studio lub programy związane z pomocą techniczną.

Bardziej praktyczne są małe zrzuty. Zawierają one najważniejsze informacje potrzebne pracownikom pomocy technicznej. Aby utworzyć mały zrzut, zastosuj następujące polecenie:

```
SqlDumper.exe <IDProcesu> 0 0x0120
```

Następnie zajrzyj do pliku *SQLDUMPER_ERRORLOG.log*, aby ustalić, czy w trakcie tworzenia pliku zrzutu zarejestrowano błędy i czy zrzut jest dostępny. Aby uruchomić program *SQLDumper.exe*, musisz być lokalnym administratorem systemu Windows lub pracować z konta użytkownika do uruchomienia usługi SQL Server.

Narzędzie SQLDiag.exe

Program *SQLDumper.exe* działa jak czarna skrzynka. Narzędzie *SQLDiag.exe* jest pod tym względem inne. Łączy ono informacje o systemie pobierane z kilku źródeł, takich jak:

- monitor systemu Windows (proces sysmon),
- dzienniki zdarzeń systemu Windows,
- ślady profilera systemu SQL Server,
- dzienniki błędów systemu SQL Server,
- dane o blokadach w systemie SQL Server,
- konfiguracja systemu SQL Server.

Ponieważ program *SQLDiag.exe* zbiera tak wiele informacji diagnostycznych, należy go używać tylko na żądanie lub w ramach przygotowań do rozmowy z pracownikiem pomocy technicznej. Pliki śladów profilera systemu SQL Server szybko się rozrastają, dlatego trzeba je zapisywać na dyskach o dużej ilości wolnej pamięci. Ponadto proces zbierania informacji wymaga dużej mocy obliczeniowej. Omawiane narzędzie można uruchomić w wierszu poleceń lub jako usługę. Dostępne opcje można wyświetlić za pomocą parametru - ?.

UWAGA Narzędzie SQL Nexus (dostępne w serwisie Codeplex na stronie <http://sqlnexus.codeplex.com>) ułatwia wydajne wczytywanie i analizowanie danych wyjściowych z programu *SQLDiag.exe*.

Program *SQLDiag.exe* może przyjmować plik konfiguracyjny jako dane wejściowe. Domyślnie jest to plik *SQLDiag.xml*, jednak można zastosować także inny. Jeśli plik konfiguracyjny w formacie XML nie istnieje, zostanie utworzony plik *##SQLDiag.xml*. Możesz go zmodyfikować w dowolny sposób, a następnie zmienić jego nazwę na *SQLDiag.xml*.

Skoro znasz już możliwości programu *SQLDiag.exe*, zastosuj go do lokalnego serwera rozwojowego. Jeśli nie masz bezpośredniego dostępu do serwera, musisz wykorzystać do zdalnej komunikacji z serwerem narzędzie pomocnicze (np. usługi Terminal Services). Programu *SQLDiag.exe* nie da się uruchomić dla zdalnego egzemplarza systemu.

1. Aby uruchomić program, otwórz wiersz poleceń.
2. Ponieważ w ramach instalacji systemu SQL Server do zmiennej środowiskowej PATH dodawany jest odpowiedni katalog, nie musisz przechodzić do folderu, w którym znajduje się plik omawianego programu. Zamiast tego przejdź do katalogu *C:\Temp* lub do podobnego folderu na dysku, który ma przynajmniej 100 megabajtów wolnej pamięci.
3. Domyślna lokalizacja pliku opisywanego programu to *C:\Program Files\Microsoft SQL Server\120\Tools\Binn\SQLDIAG.EXE*. Za pomocą opcji /O możesz zmienić tę lokalizację.
4. Wpisz poniższe polecenie (zwróć uwagę na brak spacji po znaku +):

```
sqldiag /B +00:03:00 /E +00:02:00 /OC:\temp /C1
```

Instrukcja ta nakazuje programowi *SQLDiag.exe* rozpoczęcie rejestrowania śladu po 3 minutach od wciśnięcia klawisza *Enter* i kontynuowanie tej operacji przez 2 minuty. Do określania tych wartości służą opcje /B i /E. Przy ich użyciu możesz też uruchomić i zatrzymać diagnozowanie w określonym czasie (podawanym według zegara 24-godzinnego). To polecenie nakazuje też programowi zapisanie wyników śledzenia i dzienników w katalogu *C:\Temp*, a opcja /C powoduje kompresję plików za pomocą algorytmów z systemu Windows. Jeśli wywołasz to polecenie w swoim środowisku, powinieneś poczekać z próbą odtworzenia problemu do czasu pojawienia się właściwej instrukcji programu *SQLDiag.exe* (wyświetlanej zieloną czcionką). W sytuacji przedstawionej na rysunku 4.14 program *SQLDiag.exe* rejestruje dane w katalogu domyślnym. Wciśnij kombinację *Ctrl+C*, jeśli chcesz szybciej zakończyć zbieranie danych.

5. Po zakończeniu pracy przez program *SQLDiag.exe* przejdź do katalogu *C:\Temp*, spakuj dane i prześlij je do Microsoftu. W tym katalogu znajdziesz wiele pomocnych informacji. Oto niektóre z nich.
 - *##files.txt*. Zawiera listę plików z katalogu *C:\Program Files\Microsoft SQL Server\120\Tools\Binn* wraz z datami ich utworzenia. Na tej podstawie można ustalić, czy używana jest poprawka, której instalację zalecił pracownik pomocy technicznej.

```

Administrator: Command Prompt
02/21/2014 06:20 AM 286,912 xmlrw.dll
13 File(s) 3,279,673 bytes
4 Dir(s) 14,904,651,776 bytes free

C:\Program Files\Microsoft SQL Server\120\Tools\Binn>sqldiag /B +00:03:00 /E +00:00 /OC:\temp\Cq
2014/05/18 22:34:28.28 SQLDIAG Begin time +00:03:00 specified. Waiting
2014/05/18 22:35:26.76 SQLDIAG Ctrl+C pressed. Shutting down the collector
2014/05/18 22:35:26.76 SQLDIAG Ctrl+C pressed. Shutting down subprocesses
2014/05/18 22:35:26.76 SQLDIAG WARNING: End time passed before begin time reached or wait canceled by user
2014/05/18 22:35:26.76 SQLDIAG Begin time +00:03:00 specified. Waiting
2014/05/18 22:35:26.76 SQLDIAG Collector version
2014/05/18 22:35:26.76 SQLDIAG

IMPORTANT: Please wait until you see "Collection started" before attempting to reproduce your issue

2014/05/18 22:35:26.76 SQLDIAG Output path: C:\temp\
2014/05/18 22:35:26.82 SQLDIAG Collecting from 1 logical machine(s)
2014/05/18 22:35:26.83 SQL2014-RIM* SQL Server version: 12
2014/05/18 22:35:26.83 SQL2014-RIM* Machine name: SQL2014-RIM (this machine)
2014/05/18 22:35:26.83 SQL2014-RIM* Target machine is not a cluster
2014/05/18 22:35:26.83 SQL2014-RIM* Instance: (Default) (64-bit)

C:\Program Files\Microsoft SQL Server\120\Tools\Binn>

```

Rysunek 4.14. Używanie programu SQLDiag.exe

- *##envvars.txt*. Zawiera listę wszystkich zmiennych środowiskowych serwera.
- *SERVERNAME__sp_sqldiag_Shutdown.OUT*. Zawiera dzienniki danego egzemplarza systemu SQL Server i wyniki kwerend.
- *log_XX.trc*. Jest to seria plików ze śladem z profilera opisujących szczegółowo operacje wykonywane w systemie SQL Server.
- *SERVERNAME_MSINFO32.TXT*. Zawiera liczne szczegółowe informacje na temat systemu i sprzętu serwera.

Pliki te przydadzą się nie tylko pracownikom pomocy technicznej. Warto rozważyć regularne zbieranie takich informacji, aby otrzymać punkt odniesienia dla pracy serwera w ważnych momentach (np. przed instalacją poprawek, raz na miesiąc itd.). W takiej sytuacji zadбай o to, aby profiler programu *SQLDiag.exe* nie działał dłużej niż kilka sekund.

Okresowe uruchamianie tego narzędzia w trybie snapshot pozwala zebrać przydatne informacje używane jako punkt odniesienia. Ten tryb działa zgodnie z wcześniejszym opisem, ale kończy pracę natychmiast po uzyskaniu niezbędnych danych. W poniższej instrukcji użyto opcji */X*, aby uruchomić program *SQLDiag.exe* w trybie snapshot, oraz opcji */N* z wartością 2 w celu tworzenia nowego katalogu przy każdym wywołaniu tego programu:

```
sqldiag /OC:\temp\baseline /X /N 2
```

Pierwszy tworzony katalog to *baseline_0000*, a każdy następny ma numer o jeden większy. Wiele firm uruchamia opisany program za pomocą narzędzia SQL Server Agent lub menedżera zadań pierwszego dnia miesiąca lub przed ważnymi zmianami. Pozwala to automatycznie uzyskać punkt odniesienia opisujący pracę serwera i egzemplarza systemu.

Podsumowanie

Jedną z najważniejszych kwestii przy korzystaniu z systemu SQL Server jest umiejętność zarządzania nim i rozwiązywania związanych z tym narzędziem problemów. W tym celu trzeba przygotować odpowiednią konfigurację, stale śledzić pracę systemu i wydajnie rozwiązywać problemy. Przy konfigurowaniu należy zastosować program SQL Server Configuration Manager, parametry uruchomieniowe i uruchomieniowe procedury składowane. Do aktywnego monitorowania baz danych służą narzędzie SQL Server Management Studio i język T-SQL.

Połączenia DAC to doskonały sposób na połączenie się z egzemplarzem w celu rozwiązania powstałych problemów. Gdy nie zdołasz sam poradzić sobie z trudnościami, za pomocą programów *SQLDumper.exe* i *SQLDiag.exe* możesz zebrać informacje udostępniane potem pracownikom pomocy technicznej.

Po przeczytaniu rozdziału 5. będziesz wiedział, jak zautomatyzować wykonywanie zadań w systemie SQL Server.

Automatyzowanie pracy systemu SQL Server

ZAWARTOŚĆ ROZDZIAŁU:

- Automatyzowanie standardowych zadań administracyjnych za pomocą zadań konserwacji.
- Planowanie wykonywania konserwacji z wykorzystaniem narzędzia SQL Server Agent.
- Chronienie zadań przy użyciu zabezpieczeń narzędzia SQL Server Agent.
- Konfigurowanie narzędzia SQL Server Agent pod kątem własnych potrzeb.
- Planowanie wykonywania zadań na wielu serwerach z zastosowaniem funkcji zarządzania wieloma serwerami.

Liczne zadania wykonywane przez administratorów baz danych są powtarzalne. Dotyczy to np. tworzenia kopii zapasowych baz danych, odtwarzania indeksów lub sprawdzania wielkości plików i dostępności miejsca na dysku. Do codziennych zadań administratorów baz danych należy też reagowanie na różne zdarzenia, takie jak brak miejsca na dysku na wpisy z dziennika transakcji. Problemy rosną, gdy administrator musi opiekować się dużą liczbą serwerów. Zautomatyzowanie zadań to coś więcej niż ułatwienie sobie pracy — w systemach korporacyjnych jest to konieczność.

Pomocą dla administratorów baz danych są dwie funkcje systemu SQL Server 2014 — plany konserwacji (ang. *maintenance plans*) i SQL Server Agent. Plany konserwacji umożliwiają zautomatyzowanie standardowych zadań konserwacji dotyczących baz danych. W ten sposób można zautomatyzować m.in. tworzenie kopii zapasowych, sprawdzanie integralności baz danych lub konserwację indeksów. Opracowanie planów konserwacji ułatwia specjalny kreator. Narzędzie SQL Server Agent pozwala ręcznie utworzyć harmonogram zadań wykonywanych w systemie SQL Server. Dodatkowo zwiększa to możliwości administratora w zakresie automatyzowania powtarzalnych operacji.

Plany konserwacji

Plany konserwacji pozwalają szybko i łatwo zautomatyzować rutynowe zadania administracyjne w systemie SQL Server. Ten mechanizm jest tylko interfejsem użytkownika, który umożliwia wykonywanie zwykłych zadań narzędzia SQL Server Agent. Jednak zadania w planie nie są odpowiednikiem kroków zadań agenta, ponieważ plany konserwacji są oparte na usługach SSIS (ang. *SQL Server Integration Services*) i działają jako jeden krok w zadaniu SSIS powiązany z nazwą danego planu. W obszarze rutynowych zadań konserwacji takie plany w wielu egzemplarzach systemu SQL Server mogą okazać się wystarczającym mechanizmem automatyzowania operacji.

Plany konserwacji można tworzyć i modyfikować na dwa sposoby. Szybkie i łatwe podejście polega na wykorzystaniu kreatora. Ręczny sposób wymaga użycia projektanta planów konserwacji.

OSTRZEŻENIE Jeśli używasz nowej instalacji systemu SQL Server 2014, opcja Agent XPs jest domyślnie wyłączona. Uniemożliwia to uruchomienie narzędzia SQL Server Agent. Jeśli SQL Server Agent nie jest włączony, próba uruchomienia kreatora planów konserwacji zakończy się niepowodzeniem. Aby zapobiec takiemu błędowi, włącz opcję Agent XPs za pomocą poniższego skryptu, a następnie uruchom usługę SQL Server Agent (w programie SQL Server Management Studio lub SQL Server Configuration Manager):

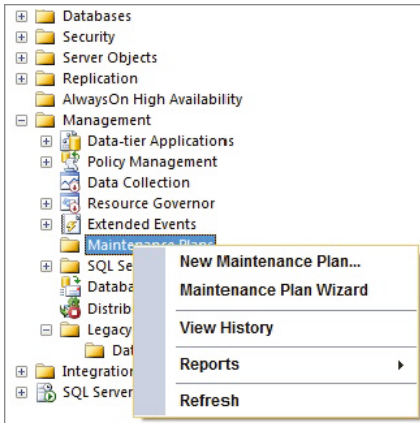
```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Agent XPs', 1;
GO
RECONFIGURE
GO
```

Po wykonaniu tego skryptu przy węźle *SQL Server Agent* w programie SQL Server Management Studio nie będzie już znajdowała się uwaga *Agent XPs disabled*.

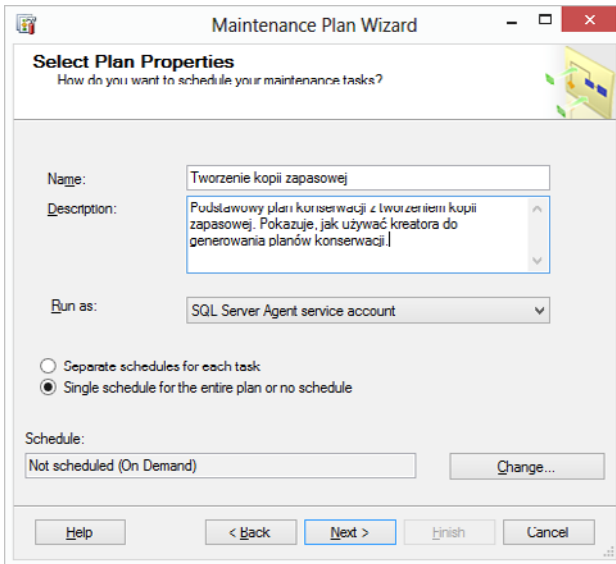
Kreator planów konserwacji

W tym punkcie omawiamy kroki potrzebne do utworzenia kopii zapasowej za pomocą kreatora planów konserwacji.

1. Najpierw uruchom kreator. Jest on dostępny w menu kontekstowym węzła *Maintenance Plans* w oknie *Object Explorer* programu SQL Server Management Studio. Wybierz z menu kontekstowego opcję *Maintenance Plan Wizard*, aby otworzyć pierwszą stronę kreatora. Przedstawiono ją na rysunku 5.1.
2. Możesz włączyć opcję blokującą ponowne wyświetlanie tego okna. Następnie kliknij przycisk *Next*. Pojawi się strona *Select Plan Properties*. Służy ona do ustawiania wybranych opcji planu. Na rysunku 5.2 widać, że na tej stronie można podać nazwę i opis planu, a także wybrać opcje harmonogramów zadań.

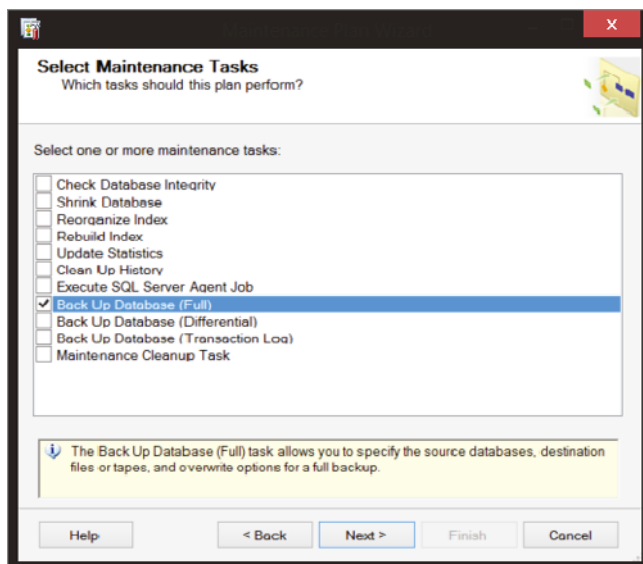


Rysunek 5.1. Pierwsza strona kreatora Maintenance Plan Wizard

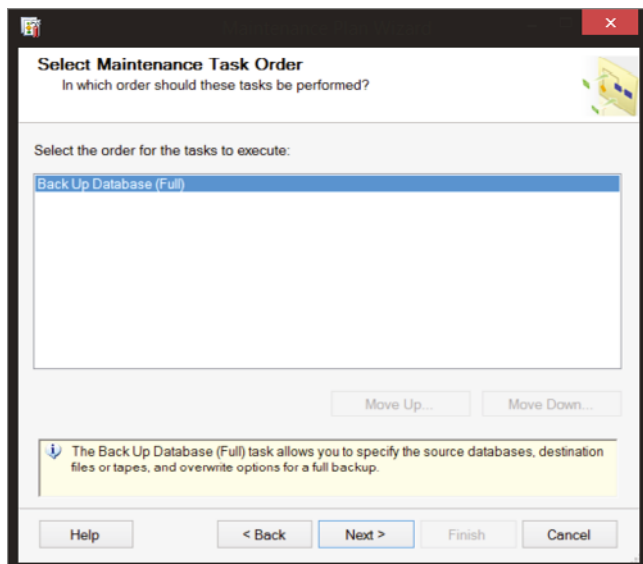


Rysunek 5.2. Konfigurowanie właściwości planu konserwacji

3. Kliknij przycisk *Next*, aby przejść do ekranu *Select Maintenance Tasks*. Możesz na nim wybrać zadania, które mają być wykonywane w ramach planu. W tym przykładzie wybierz opcję *Back Up Database (Full)*, co przedstawiono na rysunku 5.3.
4. Kliknij przycisk *Next*, by przejść do ekranu *Select Maintenance Task Order* (patrz rysunek 5.4). Jeśli na poprzedniej stronie wybrałeś kilka zadań, tu możesz zmienić ich kolejność, aby były wykonywane w odpowiadającym Ci porządku. Tu używane jest tylko jedno zadanie, dlatego kliknij przycisk *Next*.

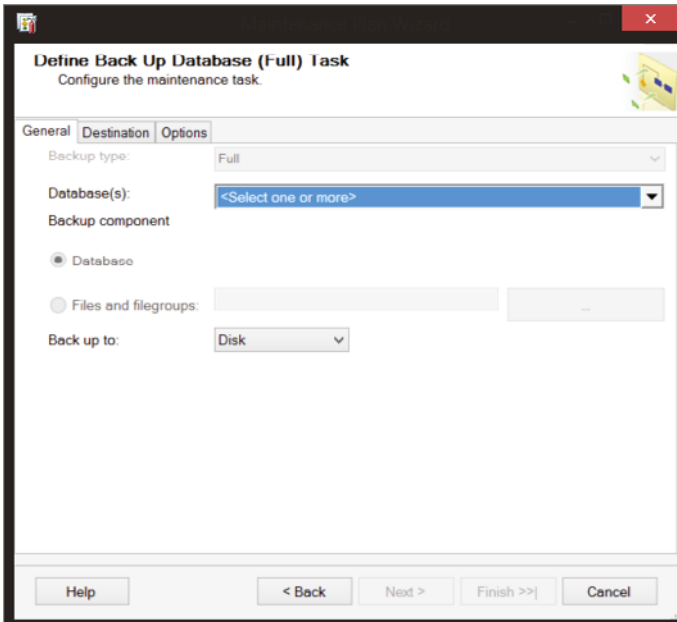


Rysunek 5.3. Wybieranie wykonywanego zadania



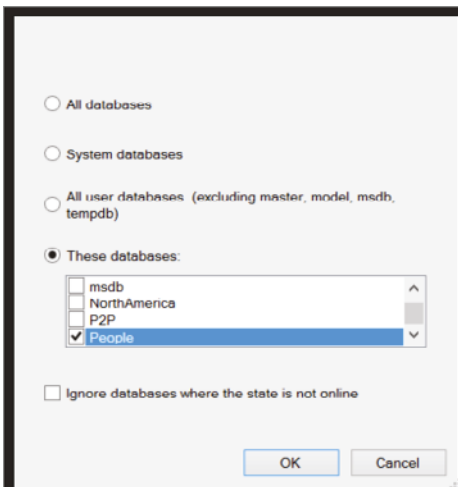
Rysunek 5.4. Tu możesz ustawić kolejność wykonywania zadań

5. Następną stroną jest *Define Back Up Database (Full) Task*, widoczna na rysunku 5.5. Na tej stronie ustaw szczegóły zadania tworzenia kopii zapasowej. Jeśli na ekranie *Select Maintenance Tasks* wybrałeś inne zadanie, tu ustaw dotyczące go szczegóły. Jeżeli planujesz kilka zadań, w tym kroku zobaczysz odrębną stronę dla każdego uwzględnionego w planie zadania.



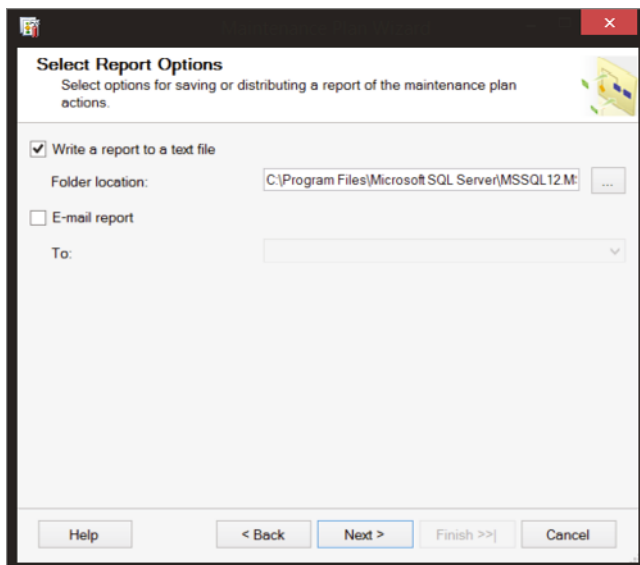
Rysunek 5.5. Konfigurowanie zadania

6. Na rysunku 5.6 przedstawiono okno dialogowe przeznaczone do ustawienia baz danych, których kopie zapasowe mają być utworzone. To okno wygląda inaczej niż w wersji SQL Server 2012, gdzie obejmowało tylko jedną stronę. Obecnie, jak widać na rysunku 5.5, dostępne są trzy zakładki: *General*, *Destination* i *Options*.



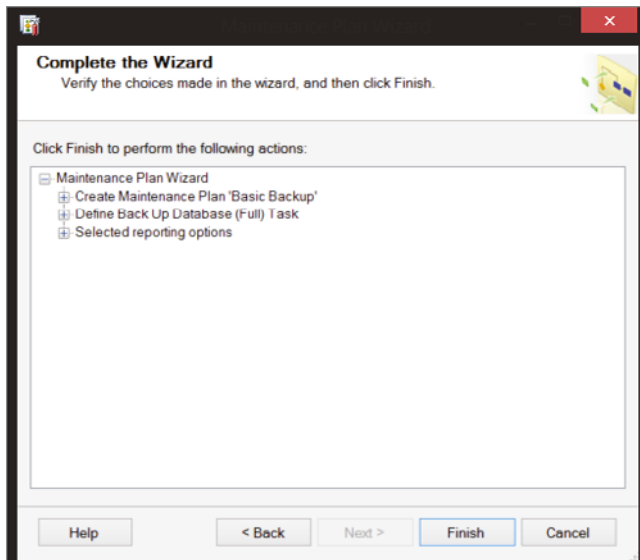
Rysunek 5.6. Wybieranie baz danych, których kopie zapasowe mają być utworzone

7. Na następnej stronie (pokazanej na rysunku 5.7) ustaw opcje raportów: *Write report to a text file*, *E-mail report* lub obie z tych opcji.



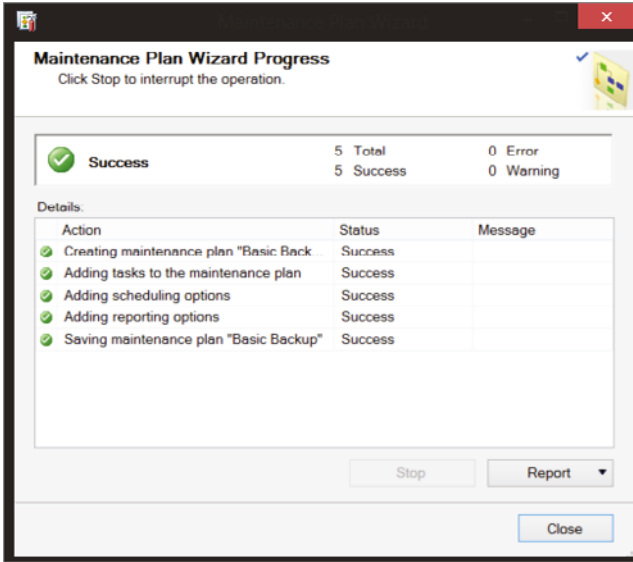
Rysunek 5.7. Ustawianie opcji generowania raportów

8. Kliknij przycisk *Next*, aby przejść do ostatniej strony kreatora, gdzie możesz zatwierdzić ustawione opcje (patrz rysunek 5.8).



Rysunek 5.8. Zatwierdź ustawienia planu

9. Kliknij przycisk *Finish*, aby zakończyć opracowywanie planu. W trakcie tworzenia planu na stronie stanu wyświetlane są informacje o postępie wykonywania każdego etapu tego procesu (patrz rysunek 5.9).



Rysunek 5.9. Informacje o przebiegu tworzenia planu

Nowy plan pojawi się w węźle *Maintenance Plans* w oknie *Object Explorer*. Możesz uruchomić ten plan ręcznie za pomocą menu podręcznego węzła.

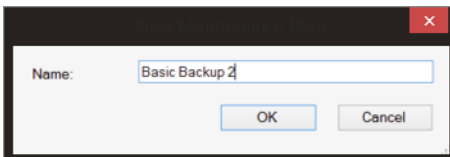
Może zauważyłeś, że kreator planów konserwacji obsługuje tylko wybrane operacje. Są to jednak najważniejsze zadania z zakresu konserwacji serwera. Zastosowanie kreatora automatyzuje liczne podstawowe zadania związane z systemem SQL Server.

Aby zapoznać się ze szczegółami utworzonego planu, przyjrzyj się zadaniom, jakie dany plan obejmuje. Są one dostępne w węźle *SQL Server Agent/Jobs*. Zadania mają nazwy w formacie `<nazwa planu>.subplan_1`, tak więc tu będzie to zadanie `MaintenancePlan.Subplan_1`.

Projektant planów konserwacji

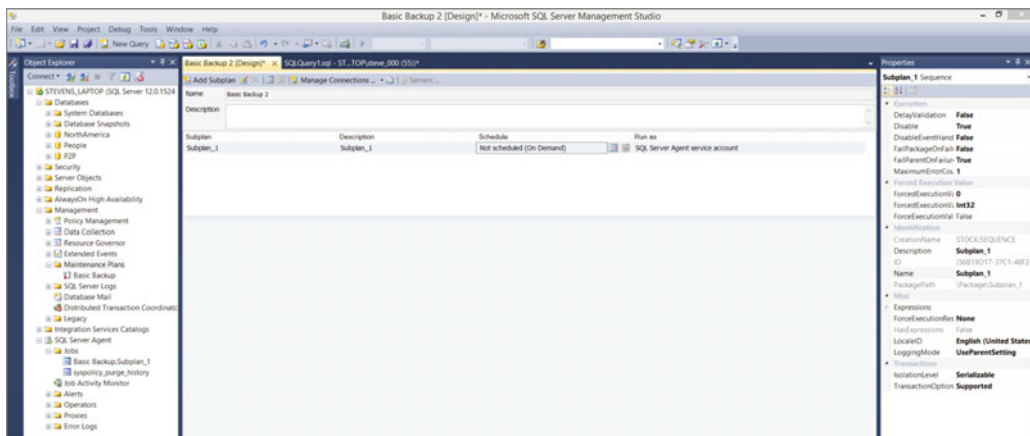
Po zastosowaniu kreatora planów konserwacji do utworzenia prostego zadania tworzenia kopii zapasowej pora zobaczyć, jak uzyskać ten sam efekt za pomocą projektanta. Wykonaj następujące kroki.

1. Rozwiń węzeł *Management* w oknie *Object Explorer*, a następnie kliknij prawym przyciskiem myszy węzeł *Maintenance Plans*. Teraz wybierz opcję *New Maintenance Plan*. Pojawi się okno dialogowe *New Maintenance Plan* widoczne na rysunku 5.10. Wprowadź nową nazwę planu, *Basic Backup 2*. Zadbaj o to, aby różniła się ona od nazwy planu utworzonego wcześniej przy użyciu kreatora planów konserwacji. Kliknij przycisk *OK*.



Rysunek 5.10. Podawanie nazwy nowego planu konserwacji

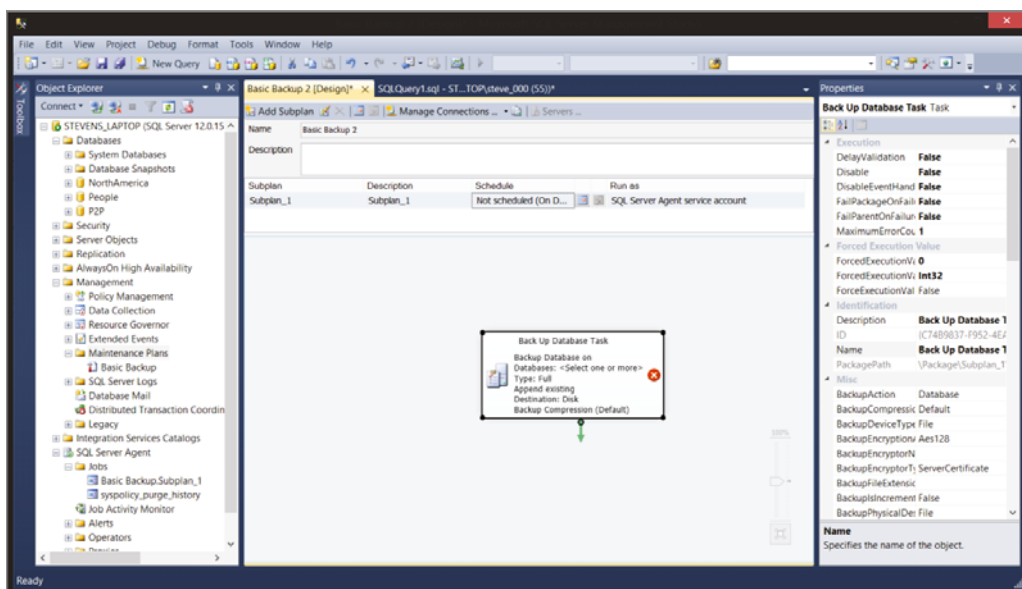
Na rysunku 5.11 przedstawiono okno dialogowe *Plan Designer*, które się pojawi. W narzędziu Management Studio zobaczysz dwa nowe okna. Okno *Plan Designer* pojawia się na środku ekranu, a po prawej stronie widoczne jest okno *Properties*. Jeśli okno *Properties* nie pojawia się automatycznie, to aby je wyświetlić, kliknij okno *Plan Designer* prawym przyciskiem myszy i wybierz z menu opcję *Properties*.



Rysunek 5.11. Główne okno projektanta planów konserwacji

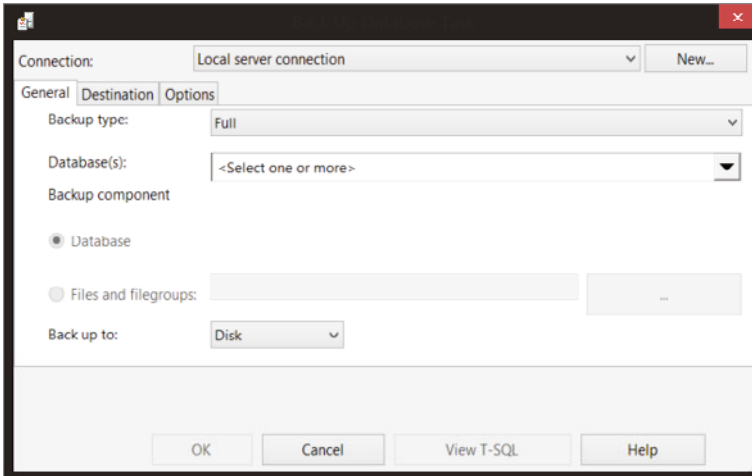
Po lewej stronie ekranu dodawana jest zakładka *Toolbox* projektanta planów konserwacji. Aby ją rozwinąć, kliknij zakładkę o tej nazwie widoczną przy lewej krawędzi ekranu.

2. Aby utworzyć podstawowe zadanie tworzenia kopii zapasowej, kliknij zadanie *Back Up Database Task* na panelu narzędzi i przeciągnij je do okna projektowego. Okno projektowe powinno wyglądać teraz tak, jak na rysunku 5.12.



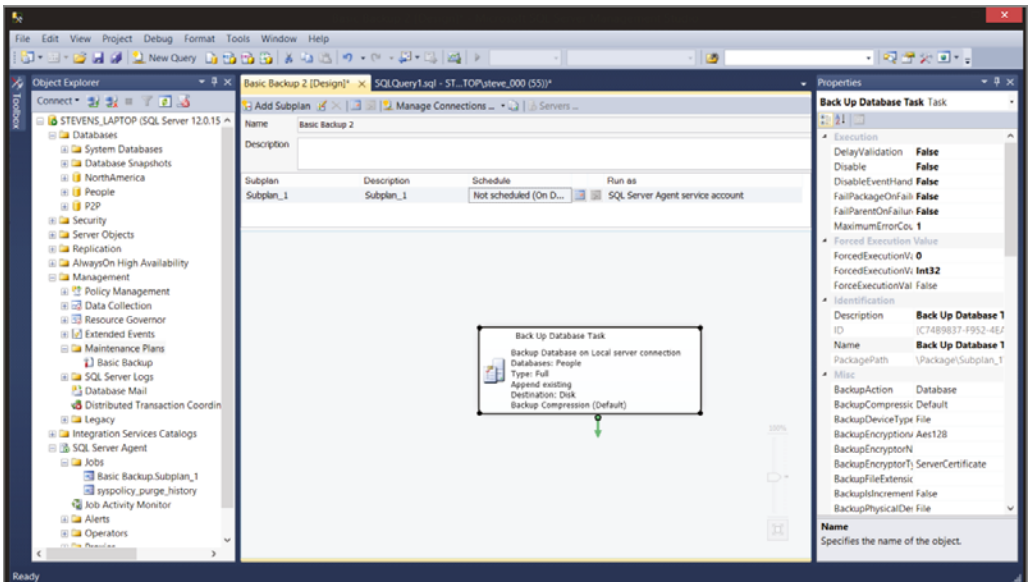
Rysunek 5.12. Okno projektowe z zadaniem tworzenia kopii zapasowej

3. Na tym etapie opracowałeś już podstawowe zadanie tworzenia kopii zapasowej, ale nie zdefiniowałeś, jak powinno działać. Aby ustawić te same parametry, co przy korzystaniu z kreatora, zmodyfikuj właściwości zadania. W tym celu kliknij dwukrotnie zadanie w oknie projektowym. Pojawi się okno właściwości zadania widoczne na rysunku 5.13.



Rysunek 5.13. Okno z właściwościami zadania

4. To jest to samo okno dialogowe, które uzupełniłeś za pomocą kreatora. Wybierz tę samą bazę danych i te same opcje, które zastosowałeś przy korzystaniu z kreatora planów konserwacji. Po wprowadzeniu zmian kliknij przycisk OK, aby wrócić od okna projektowego. Tym razem przy zadaniu *Back Up Database Task* nie powinna być widoczna czerwona ikona ostrzegawcza. Na tym etapie okno powinno zawierać informację o wybranej bazie i wyglądać tak, jak na rysunku 5.14.



Rysunek 5.14. Okno po skonfigurowaniu zadania

5. Aby utworzyć zaprojektowany właśnie plan, wystarczy go zapisać. Plan zostanie wygenerowany.

W dowolnym momencie możesz użyć projektanta planów konserwacji do zmodyfikowania właściwości danego planu.

Automatyzowanie systemu SQL Server za pomocą narzędzia SQL Server Agent

Gdy plan konserwacji nie zapewnia potrzebnego poziomu automatyzacji systemu SQL Server lub gdy potrzebujesz większych możliwości niż oferowane przez plany konserwacji, użyj bezpośrednio narzędzia SQL Server Agent.

Narzędzie SQL Server Agent obejmuje cztery podstawowe komponenty opisane w dalszych punktach. Oto one.

- **Zadania (ang. *jobs*)** — definiują wykonywane operacje.
- **Harmonogramy (ang. *schedules*)** — określają czas wykonywania zadań.
- **Operatorzy (ang. *operators*)** — określają osoby, które należy powiadamiać o stanie zadań i alertach.
- **Alerty (ang. *alerts*)** — umożliwiają skonfigurowanie automatycznych odpowiedzi lub powiadomień przesyłanych w reakcji na wystąpienie zdarzeń.

Zadania

Ważnym powodem stosowania narzędzia SQL Server Agent jest możliwość kreowania zadań (np. tworzenia kopii zapasowej bazy) i planowania ich automatycznego wykonywania. *Zadanie* w narzędziu SQL Server Agent określa, jakie operacje mają zostać wykonane. Samo zadanie nie wykonuje operacji, a jest kontenerem na wykonywane kroki. Zadanie ma nazwę, opis, właściciela i kategorię oraz może być aktywne lub nieaktywne. Zadania są uruchamiane na kilka sposobów:

- przez przypisanie zadania do harmonogramów,
- w odpowiedzi na wystąpienie alertów,
- za pomocą instrukcji `sp_start_job`,
- ręcznie przy użyciu narzędzia SQL Server Management Studio.

Kroki zadań

Zadanie składa się z *kroków*. Kroki zadania obejmują wykonywane operacje. Każdy krok ma nazwę i typ. Koniecznie nadawaj zadaniom i krokom dobre, opisowe nazwy, które będą przydatne w komunikatach o błędach i dziennikach. Dostępnych jest wiele typów kroków. Oto wybrane z nich.

- **Skrypty Active X (*ActiveX Script*)**. Umożliwiają wykonywanie kodu w językach VBScript, Jscript i innych instalowanych językach skryptowych.
- **Polecenia systemu operacyjnego (*Operating System (CmdExec)*)**. Umożliwiają wykonywanie instrukcji wiersza poleceń. W ten sposób można uruchamiać pliki `.bat` oraz polecenia dozwolone w plikach `.bat` i `.cmd`.

- **Zadania powłoki PowerShell (*PowerShell*).** Umożliwiają wykonywanie w zadaniach skryptów powłoki PowerShell systemu Windows.
- **Polecenia usług SSAS (*SQL Server Analysis Services Command*).** Umożliwiają wykonywanie poleceń w formacie XMLA (ang. *XML for Analysis*). Wymagają stosowania metody *Execute*, która pozwala pobierać dane, zarządzać obiektami usług Analysis Services i przetwarzać je.
- **Kwerendy usług SSAS (*SQL Server Analysis Services Query*).** Umożliwiają wykonywanie wyrażeń MDX (ang. *Multidimensional Expression*) dla kostek danych. Wyrażenia MDX pozwalają na pobieranie danych z kostek.
- **Wykonywanie pakietów usług SSIS (*SQL Server Integration Services Package*).** Krok ten umożliwia wykonywanie pakietów usług SSIS. Można przy tym ustawiać wartości zmiennych, konfigurację i inne elementy potrzebne do uruchomienia pakietu. Jeśli masz gotowe skomplikowane pakiety usług SSIS i chcesz je wykonywać w ramach kroku narzędzia SQL Server Agent, ten typ kroków pozwala zaoszczędzić dużo czasu.
- **Skrypty w języku T-SQL (*Transact-SQL Script (T-SQL)*).** Umożliwiają wykonywanie skryptów w języku T-SQL. Takie skrypty nie korzystają z opisanych dalej w tym rozdziale kont pośredniczących narzędzia SQL Server Agent. Jeśli użytkownik nie jest przypisany do roli sysadmin na serwerze, krok ze skryptem w języku T-SQL może zostać wykonany za pomocą danych uwierzytelniających użytkownika danej bazy. Gdy kroki z takimi skryptami są tworzone przez osoby z rolą sysadmin, można określić, że dany krok ma działać w kontekście bezpieczeństwa konkretnego użytkownika bazy. Jeśli podany jest użytkownik, krok jest wykonywany z konta tego użytkownika. W przeciwnym razie krok działa w kontekście bezpieczeństwa konta usług SQL Server Agent.

UWAGA Graficzny interfejs użytkownika służący do ustawiania kontekstu bezpieczeństwa dla skryptów języka T-SQL jest nieintuicyjny. Choć na pierwszej stronie okna dialogowego *Job Step Properties*, w którym konfigurowane są kroki zadań, znajduje się lista rozwijana *Run As*, to nie tu ustawiany jest kontekst bezpieczeństwa dla skryptów języka T-SQL. Lista rozwijana *Run As* służy do określania kontekstu bezpieczeństwa dla kroków innych typów. Aby ustawić kontekst bezpieczeństwa dla kroków ze skryptami języka T-SQL, otwórz zakładkę *Advanced*. W dolnej części otwartego okna znajduje się lista rozwijana *Run As User*. To tu określony jest kontekst bezpieczeństwa dla skryptów języka T-SQL.

Istnieją też inne typy kroków zadań, jednak zwykle nie tworzy się ich samodzielnie. Te inne kroki i zadania przeważnie opracowywane są w ramach konfigurowania replikacji. Każdy krok zadania działa w *kontekście bezpieczeństwa*. Konteksty bezpieczeństwa dla pozostałych typów kroków opisano dalej w tym rozdziale. W ramach konfigurowania replikacji należy zdefiniować zadania obejmujące kroki następujących typów:

- *Replication Distributor*,
- *Replication Merge*,
- *Replication Queue Reader*,
- *Replication Snapshot*,
- *Replication Transaction-Log Reader*.

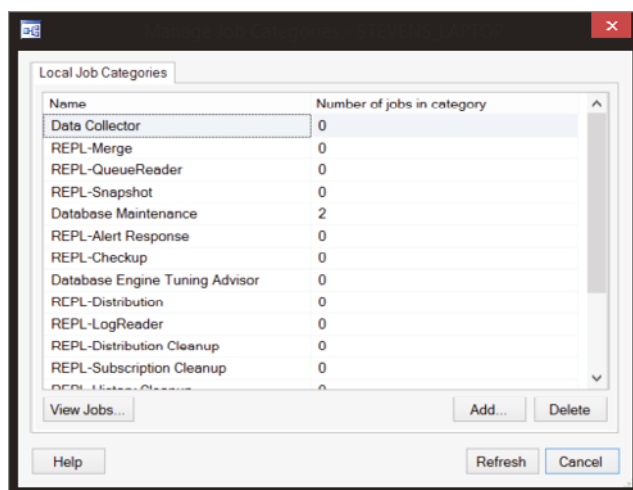
Z krokami zadań związany jest też przepływ sterowania. Możesz określić operacje wykonywane, gdy dany krok zakończy się powodzeniem lub porażką. Oto dostępne operacje.

- Zakończenie zadania po powodzeniu.
- Zakończenie zadania po porażce.
- Przejście do innego kroku zadania.

Możesz też dodać wymóg, aby przed zgłoszeniem niepowodzenia następowała próba ponownego wykonania kroku zadania. Można określić liczbę ponownych prób i odstępy między nimi (w minutach). Po ustawieniu tych wartości krok będzie ponawiany tyle razy, ile podano w polu *Retry Attempts*, a dopiero potem w przepływie sterowania wybrana zostanie ścieżka dla niepowodzenia. Jeśli wartość w polu *Retry Interval in Minutes* jest ustawiona, przed ponowieniem próby wykonania kroku następuje przerwa. Jest to przydatne, gdy zadania są zależne od siebie. Przykładowo krok zadania może wykonywać operację masowego wstawiania danych z pliku tekstowego, który jest umieszczany w odpowiednim katalogu przez inny, później uruchamiany proces.

W trakcie tworzenia zadania określana jest jego kategoria. Każde zadanie może być przypisane do tylko jednej kategorii. Dostępnych jest kilka wbudowanych kategorii, w tym *[Uncategorized (Local)]* i *Database Engine Tuning Advisor*. Można też tworzyć własne kategorie zadań. W tym celu wykonaj poniższe kroki.

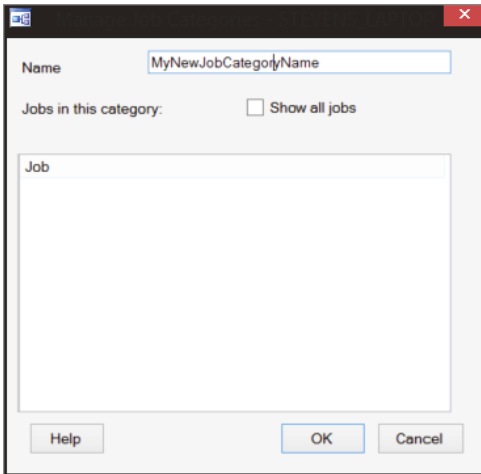
1. W oknie *Object Explorer* narzędzia *SQL Server Management Studio* otwórz węzeł *SQL Server Agent* i kliknij prawym przyciskiem myszy węzeł *Jobs*.
2. Wybierz z menu opcję *Manage Job Categories*. Pojawi się okno dialogowe widoczne na rysunku 5.15.



Rysunek 5.15. Okno z kategoriami zadań

3. Kliknij przycisk *Add*. Zobaczysz okno dialogowe przedstawione na rysunku 5.16.
4. Wprowadź nazwę nowej kategorii zadań w polu *Name* i kliknij przycisk *OK*.

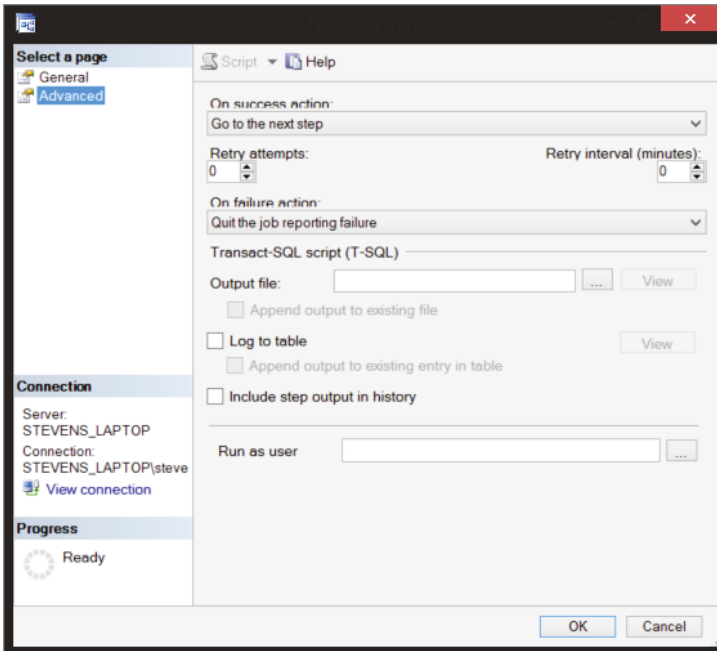
Sam proces tworzenia kategorii jest bardzo prosty, jednak najpierw należy zastanowić się nad tym, jak uporządkować zadania. Prawdopodobnie zaskoczy Cię to, jak szybko rośnie liczba zadań na serwerze, co utrudnia wyszukiwanie właściwych.



Rysunek 5.16. Okno dialogowe do dodawania kategorii zadań

Rejestrowanie danych o krokach zadań

Przy każdym uruchomieniu zadania tworzona jest *historia zadania*. Określa ona czas rozpoczęcia wykonywania zadania, czas jego zakończenia i to, czy proces zakończył się sukcesem. Także dla każdego kroku można włączyć rejestrowanie danych i tworzenie historii. Konfigurowanie tych aspektów odbywa się w zakładce *Advanced* ekranu właściwości kroku zadania, co przedstawiono na rysunku 5.17.



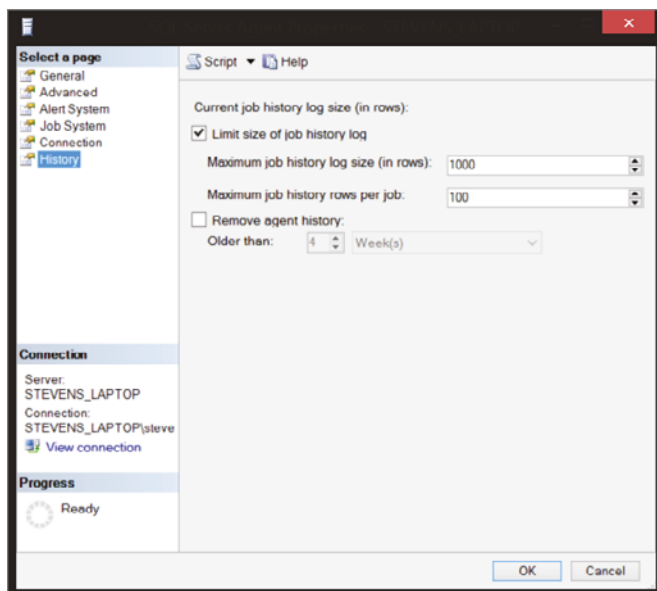
Rysunek 5.17. Konfigurowanie rejestrowania danych o krokach zadań

Oto najważniejsze opcje wpływające na rejestrowanie danych.

- **Output file.** Historia dotycząca kroków zadań wykonywanych przez osoby o roli sysadmin może być zapisywana w pliku. Aby uzyskać ten efekt, wpisz nazwę pliku w polu *Output file*. Jeśli nie chcesz zastępować istniejącego pliku, zaznacz opcję *Append output to existing file*. Kroki zadań wykonywane przez pozostałych użytkowników mogą być rejestrowane tylko w tabeli `dbo.sysjobstepslogs` w bazie `msdb`.
- **Log to table.** Możesz też ustawić rejestrowanie informacji w tabeli `dbo.sysjobstepslogs` w bazie `msdb`. W tym celu zaznacz opcję *Log to table*. Aby uwzględnić historię kroku z wielu uruchomień zadania, zaznacz także opcję *Append output to existing entry in table*. W przeciwnym razie dostępne będą informacje tylko na temat ostatniego wykonania zadania.
- **Include step output in history.** Aby dodać historię kroku do historii zadania, zaznacz opcję *Include the step output in history*.

UWAGA Gdy używasz zasobów sieciowych, np. plików systemu operacyjnego, upewnij się, że odpowiednie konto pośredniczące ma wystarczające uprawnienia. Ponadto zawsze stosuj dla plików nazwy UNC, tak aby zadania i kroki nie były zależne od odwzorowań katalogów. Jeśli nie zachowasz ostrożności, narazisz się na problemy przy przenoszeniu rozwiązań ze środowiska testowego do produkcyjnego.

System SQL Server domyślnie przechowuje tylko 1000 wierszy w dzienniku historii zadań i maksymalnie 100 wierszy na jedno zadanie. Historia zadań jest przechowywana w nadpisywanym dzienniku, w którym najstarsze rekordy są usuwane, aby zwolnić miejsce na nowszą historię. Jeśli zadań jest dużo lub są często uruchamiane, dziennik historii zadań szybko staje się pełny, co prowadzi do kasowania starszych rekordów. Jeśli musisz zmienić wielkość dziennika, użyj do tego właściwości w narzędziu SQL Server Agent (patrz rysunek 5.18).

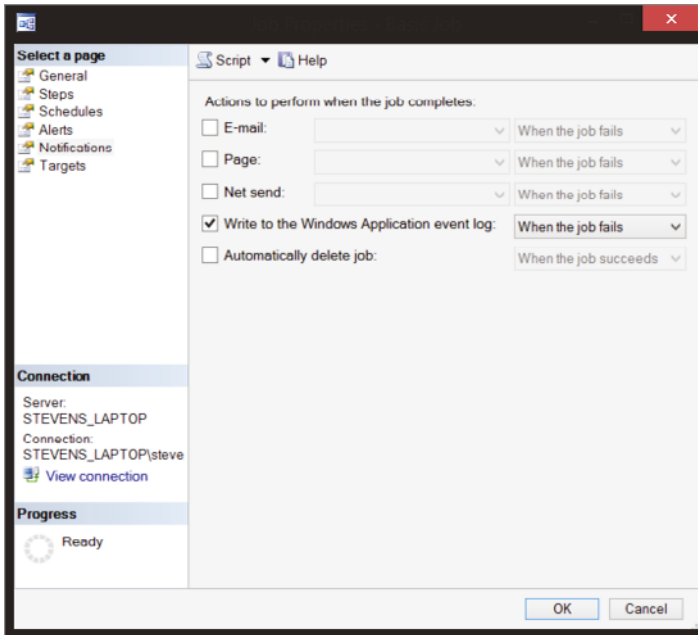


Rysunek 5.18. Na tej stronie można zmienić wielkość dziennika historii zadań

Powiadomienia o zadaniach

Możesz skonfigurować narzędzie SQL Server Agent, aby powiadamiało o zakończeniu, sukcesie lub niepowodzeniu zadania. W tym celu wykonaj następujące kroki.

1. W oknie dialogowym *Job Properties* wybierz opcję *Notifications*, aby wyświetlić okno dialogowe widoczne na rysunku 5.19.



Rysunek 5.19. Ustawianie powiadomień o zadaniach

2. Zadanie może przysyłać powiadomienie e-mailem, na pager lub za pomocą usługi Net Send. Ponadto zadania mogą zapisywać dane w dzienniku zdarzeń aplikacji systemu Windows. Na rysunku 5.19 widać, że w oknie dialogowym znajdują się wiersze powiązane z każdą z wymienionych metod. Zaznacz opcję przy technice, którą chcesz zastosować. Możesz wybrać kilka metod.
3. Rozwiń menu przy każdej z opcji i wybierz operatora, który ma otrzymywać powiadomienia. Operator umożliwia zdefiniowanie docelowego adresu e-mail. Konfigurowanie operatorów omawiamy dalej w tym rozdziale.
4. Wybierz zdarzenie, które ma prowadzić do wysłania powiadomienia. Takim zdarzeniem może być zakończenie, niepowodzenie lub sukces zadania. Możliwe, że nie interesują Cię powiadomienia o niektórych zadaniach (np. o rutynowych operacjach, takich jak konserwacja indeksu). Natomiast przy operacjach o znaczeniu strategicznym zawsze możesz zażyczyć sobie otrzymywanie e-maili o zakończeniu zadania i powiadomień za pomocą usługi Net Send, gdy zadanie zakończy się niepowodzeniem, dzięki czemu natychmiast się o tym dowiesz. Przykładowe ważne zadania, dla których przydatne mogą być powiadomienia, to tworzenie kopii zapasowych i wykonywanie polecenia DBCC CHECKDB.

UWAGA Aby możliwe było wysyłanie powiadomień za pomocą usługi Net Send, na serwerze z narzędziem SQL Server Agent musi działać usługa Windows Messenger Service. Komunikaty można przysyłać do dowolnych stacji roboczych i użytkowników dostępnych z poziomu serwera z narzędziem SQL Server Agent. By odbierać powiadomienia, na docelowej stacji roboczej także trzeba uruchomić usługę Windows Messenger Service.

Harmonogramy

Jedną z zalet narzędzia SQL Server Agent jest możliwość tworzenia harmonogramów wykonywania zadań. Możesz zaplanować uruchamianie zadań w dowolnym z wymienionych momentów:

- przy uruchamianiu narzędzia SQL Server Agent,
- raz określonego dnia o podanej godzinie,
- regularnie,
- gdy procesor serwera jest beczynny.

Aby utworzyć harmonogram w narzędziu Management Studio, wybierz węzeł *SQL Server Agent*, kliknij prawym przyciskiem myszy węzeł *Jobs* i wybierz opcję *Manage Schedules*. Dostępny program do tworzenia harmonogramów jest bardzo łatwy w obsłudze. Możesz np. utworzyć harmonogram z zadaniem uruchamianym w ostatni dzień roboczy każdego miesiąca. Wygodne jest to, że nie musisz za każdym razem sprawdzać, który dokładnie jest to dzień. Harmonogram możesz dodać w trakcie tworzenia zadania lub przygotować go niezależnie, a następnie powiązać z zadaniami.

Po opracowaniu harmonogramu możesz połączyć go z zadaniami. Jedno zadanie może znajdować się w wielu harmonogramach. Załóżmy, że istnieje harmonogram z wykonywanymi każdej nocy operacjami wsadowymi i harmonogram z operacjami przetwarzanymi na koniec miesiąca. Zadanie można powiązać z oboma takimi harmonogramami. Jeśli zadanie z harmonogramu zostanie uruchomione w czasie, gdy jest już wykonywane, zostanie pominięte.

Należy zachować staranność przy tworzeniu nazw harmonogramów, muszą być zrozumiałe. Najczęściej trudno wybrać, czy nazwa harmonogramu ma informować o *czasie* wykonywania zadań, czy o ich *rodzaju*. Oczywiście w jednej nazwie można uwzględnić informacje obu tych rodzajów. Oto przykładowe nazwy: *Harmonogram codziennego tworzenia kopii zapasowej* lub *Harmonogram codziennej konserwacji indeksu*. Dla harmonogramów z zadaniami biznesowymi można tworzyć nazwy, takie jak *Zobowiązania na koniec miesiąca* lub *Listy płac w cyklu dwutygodniowym*. Zastosowanie nazw biznesowych jest wygodne, ponieważ pozwala szybko znaleźć harmonogram powiązany z określonymi operacjami lub procesami. Dodanie czasu wykonywania operacji jest pomocne, jeśli chcesz zmienić częstotliwość wykonywania zadań z harmonogramu.

Czasem procesor serwera jest beczynny; warto wtedy wykonać jakieś zadania. Aby określić, w jakiej sytuacji procesor jest uznawany za beczynny, ustaw warunek *Idle CPU Condition* we właściwościach narzędzia SQL Server Agent w zakładce *Advanced*. Możesz zdefiniować wymagany minimalny poziom obciążenia procesora i czas, przez jaki takie obciążenie nie może zostać przekroczone. Gdy obciążenie procesora spadnie poniżej progowej wartości i utrzyma się przez zdefiniowany czas, uruchomione zostaną harmonogramy dla beczynnego procesora. Jeśli procesor nie jest zajęty, można wykonać operacje wsadowe. Zachowaj jednak ostrożność. Jeśli w harmonogramie dla beczynnego procesora umieścisz dużą liczbę zadań, mogą one

szybko zacząć pracę i doprowadzić do przeciążenia systemu. Dlatego ogranicz liczbę zadań dodawanych do takich harmonogramów.

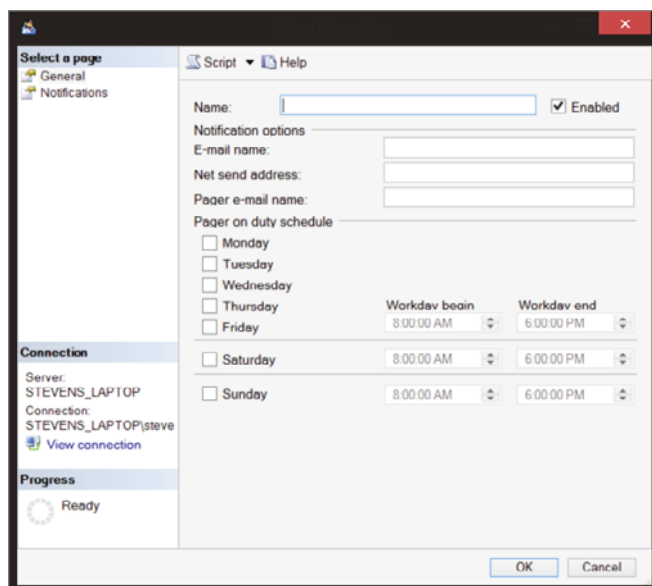
Funkcją, której niestety brakuje w narzędziu SQL Server Agent, jest możliwość łączenia zadań w łańcuch, tak aby jedno było rozpoczynane po zakończeniu innego. Taki efekt można jednak uzyskać po dodaniu w pierwszym zadaniu końcowego kroku, który uruchamia następne zadanie. Służy do tego instrukcja `sp_start_job`. Jednak w tym podejściu mechanizm poruszania się między zadaniami znajduje się w krokach. Przechodzenie między zadaniami nie powinno odbywać się na poziomie kroków — lepiej obsługiwać je na poziomie zadań. Istnieją dobre niezależne narzędzia, które to umożliwiają. Samodzielne uzyskanie potrzebnego efektu nie jest łatwe.

Operatorzy

Operator to obiekt narzędzia SQL Server Agent o przyjaznej nazwie i danych kontaktowych. Operatorzy mogą być powiadamiani o zakończeniu zadań narzędzia SQL Server Agent i o występowaniu alertów (omówienie alertów zawiera następny punkt). Możliwe, że chcesz, aby operator, który potrafi naprawić problemy związane z zadaniami i alertami, otrzymywał powiadomienia i mógł w efekcie zabrać się do pracy. Możliwe też, że chcesz automatycznie powiadamiać menedżerów o zdarzeniach o znaczeniu strategicznym, np. o niepowodzeniu przy generowaniu listy płac.

Operatorów należy zdefiniować przed utworzeniem alertów. Dzięki temu w trakcie definiowania alertów będziesz mógł wybrać powiadamianych operatorów, co przyspieszy pracę. Aby utworzyć nowego operatora, wykonaj następujące kroki.

1. Rozwiń węzeł *SQL Server Agent* w oknie *Object Explorer* narzędzia SQL Server Management Studio.
2. Kliknij węzeł *Operators* prawym przyciskiem myszy i wybierz opcję *New Operator*. Pojawi się okno dialogowe *New Operator* widoczne na rysunku 5.20. W tym oknie możesz dodać nowego operatora. Nazwa operatora musi być unikatowa i nie może zawierać więcej niż 128 znaków.



Rysunek 5.20. Dodawanie operatora

Powiadomienia operatorów

Zadania umożliwiają powiadamianie operatorów za pomocą trzech metod.

- **E-mail.** Aby zastosować powiadamianie za pomocą e-maili lub na pager, trzeba skonfigurować i włączyć funkcję Database Mail, a także skonfigurować narzędzie SQL Server Agent. Przy powiadomieniach na e-mail należy podać adres. Można podać kilka adresów rozdzielonych średnikami lub wybrać grupę zdefiniowaną w systemie poczty elektronicznej. Jeśli chcesz wysyłać powiadomienia do wielu osób, wygodniej zdefiniować taką grupę. Dzięki temu możesz zmienić listę powiadamianych osób bez konieczności modyfikowania każdego zadania.
- **Pager.** Przy powiadomieniach na pager także należy podać adres e-mail. Narzędzie SQL Server Agent samo w sobie nie obsługuje pagerów. Musisz wykupić od niezależnego dostawcy narzędzie do wysyłania powiadomień na pager na podstawie e-maili. SQL Server Agent jedynie wysyła e-mail na adres pagera, a oprogramowanie do obsługi pagerów odpowiada za pozostałe operacje. Niektóre systemy obsługi pagerów wymagają dodatkowych znaków konfiguracyjnych w polach z tematem, odbiorcą oraz odbiorcami kopii. Odpowiednie ustawienia można skonfigurować w narzędziu SQL Server Agent (opisano to w końcowej części rozdziału).

Zwróć uwagę na opcję *Pager on duty schedule* powiązaną z ustawieniem *Pager e-mail name*. Ta opcja dotyczy tylko pagerów. Możesz ustawić harmonogram z godzinami pracy danego operatora, aby wysyłać mu na pager powiadomienia o alertach lub zakończeniu zadań tylko wtedy, gdy jest dostępny.

- **Net Send.** Do powiadamiania operatorów można też wykorzystać usługę Net Send. Aby ją zastosować, na serwerze z narzędziem SQL Server Agent trzeba uruchomić usługę Windows Messaging Service. Ponadto należy podać nazwę stacji roboczej operatora. Po przesłaniu powiadomienia na komputerze operatora pojawi się wyskakujące okno dialogowe z komunikatem. Spośród trzech metod przesyłania komunikatów usługa Net Send jest najmniej pewna, ponieważ komunikat jest widoczny tylko przez krótki czas. Jeśli w momencie nadejścia powiadomienia operator nie znajduje się przy biurku lub z jakichkolwiek powodów docelowy serwer jest niedostępny, komunikat nie trafi do odbiorcy.

Powiadomienia o alertach mogą dotrzeć do wielu operatorów. Daje to ciekawe możliwości. Możesz np. określić operatorów dla każdej zmiany (operatorzy dla pierwszej zmiany, dla drugiej zmiany i dla trzeciej zmiany), skonfigurować grupę adresów e-mail i grupę adresów pagerów dla każdej zmiany, dopasować harmonogram godzin pracy do poszczególnych zmian i powiązać wszystkich operatorów z każdym alertem. Jeśli alert zostanie zgłoszony o 2:00 w nocy, powiadomienia zostaną wysłane tylko na pagery operatorów z trzeciej zmiany. Jeżeli alerty wystąpią o godzinie 10:00 rano, komunikaty otrzymają tylko operatorzy z pierwszej zmiany.

Zauważ, że harmonogram dla wszystkich dni roboczy musi być taki sam, choć można ustawić różne harmonogramy dla soboty i niedzieli. Ponadto nie istnieją narzędzia do ustawiania firmowych świąt i wakacji. Możesz wyłączyć operatorów (np. z powodu ich urlopu), ale nie możesz z góry uwzględnić tego w harmonogramie.

Operator awaryjny

Co się stanie, gdy wystąpi alert, a według harmonogramu żaden z operatorów w danym momencie nie pracuje? Jeśli nie ustawiłeś operatora awaryjnego, nikt nie otrzyma powiadomienia. *Operator awaryjny* jest traktowany jako zabezpieczenie umożliwiające dostarczenie na pager

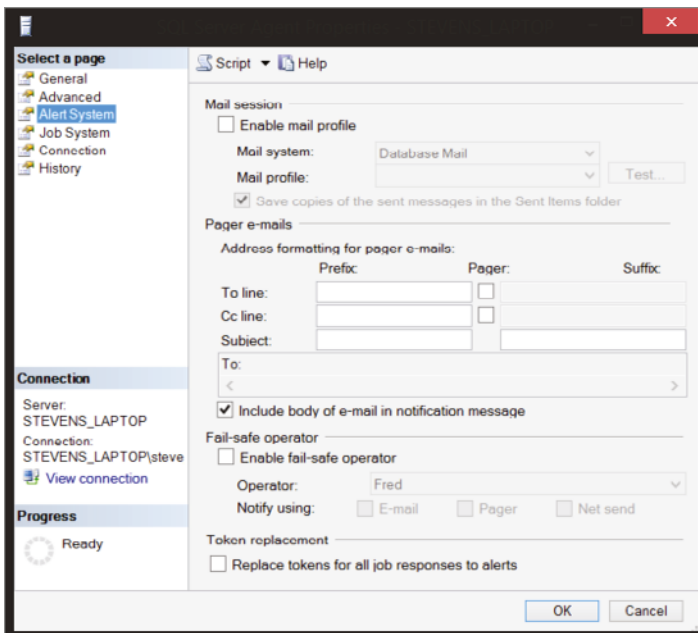
(nie na e-mail lub w usłudze Not Send) powiadomienia o alercie (powiadomienia o zadaniach nie są uwzględniane), którego nie można przesłać w standardowy sposób. Oto warunki, gdy system uznaje, że powiadomienia nie można przesłać na pager.

- Żaden z uwzględnianych operatorów nie jest w pracy.
- SQL Server Agent nie ma dostępu do odpowiednich tabel z bazy msdb.

UWAGA Jeśli nie zdefiniowano żadnych operatorów, opcja ustawiania operatora awaryjnego jest nieaktywna.

Aby ustawić operatora awaryjnego, wykonaj następujące kroki.

1. Otwórz właściwości narzędzia SQL Server Agent.
2. Otwórz zakładkę *Alert System*, co pokazano na rysunku 5.21.



Rysunek 5.21. Ustawianie operatora awaryjnego

3. W sekcji *Fail-safe operator* zaznacz opcję *Enable fail-safe operator*.

Operator awaryjny jest powiadamiany tylko wtedy, gdy nie można przesłać *żadnego* z ustawionych powiadomień na pager lub gdy baza msdb jest niedostępna. Jeśli z danym alertem powiązanych jest trzech operatorów i jeden z nich otrzyma powiadomienie, a w przypadku dwóch się to nie uda, operator awaryjny *nie zostanie powiadomiony*.

Możesz określić, jakimi z opisanych wcześniej metod ma być powiadamiany operator awaryjny. Operator ten otrzymuje informacje tylko wtedy, gdy nie można dostarczyć powiadomień na pager. Wtedy można go powiadomić za pomocą e-maila, pagera, usługi Net Send lub dowolnej kombinacji tych metod.

Ponieważ operator awaryjny jest traktowany jak zabezpieczenie, nie można usunąć operatora przypisanego do tej roli. Gdy chcesz zrezygnować z danego operatora awaryjnego, musisz najpierw wyłączyć tę funkcję w narzędziu SQL Server Agent lub wybrać innego operatora tego rodzaju. Następnie możesz usunąć wybranego operatora. Wyłączenie operatora awaryjnego powoduje, że przestanie on otrzymywać zwykłe alerty i powiadomienia o zadaniach, jednak nadal będzie odbierał powiadomienia awaryjne.

Alerty

Alert to zautomatyzowana odpowiedź na zdarzenie. Zdarzeniem może być dowolna z następujących sytuacji:

- zdarzenie systemu SQL Server,
- warunek związany z wydajnością systemu SQL Server,
- zdarzenie usługi WMI (ang. *Windows Management Instrumentation*).

Możesz utworzyć alert reagujący na dowolne z takich zdarzeń. W odpowiedzi na alert można wywołać takie reakcje jak:

- uruchomienie zadania narzędzia SQL Server Agent,
- powiadomienie jednego lub kilku operatorów.

UWAGA Choć po zakończeniu zadania każdym z obsługiwanych kanałów można powiadomić tylko jednego operatora, po wystąpieniu alertu można przesłać powiadomienia do wielu operatorów.

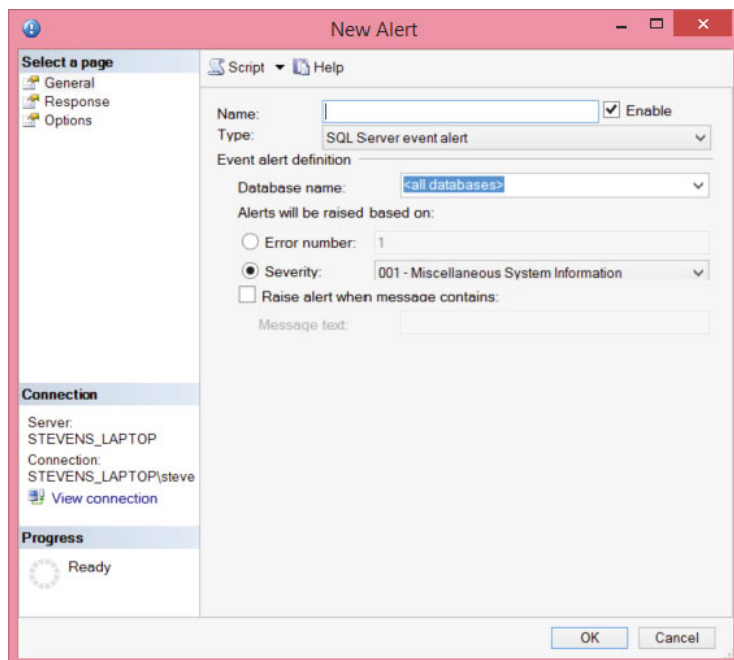
Aby utworzyć alert, wykonaj następujące kroki.

1. Otwórz okno dialogowe *New Alert*, przedstawione na rysunku 5.22. W tym celu wybierz opcję *New Alert* z menu kontekstowego węzła *SQL Server Agent/Alerts* w programie SQL Server Management Studio.
2. W trakcie tworzenia alertu ustaw jego nazwę. Zadbaj o to, aby informowała o przyczynie alertu. Nazwa pojawia się we wszystkich przesyłanych komunikatach. Przydatne są nazwy, takie jak *Alert — pełny dziennik* lub *Alert poziomu 18 w systemie produkcyjnym*.
3. Wybierz typ zdarzenia, z którym powiązany jest alert (patrz rysunek 5.22). Omówienie zdarzeń systemu SQL Server i warunków związanych z wydajnością systemu SQL Server znajdziesz w następnych podpunktach. Kompletne omawianie zdarzeń usług WMI wykracza poza zakres tej książki, dlatego to zagadnienie opisano tylko pokrótce.

Alerty związane ze zdarzeniami systemu SQL Server

Alerty dotyczące zdarzeń systemu SQL Server są najczęściej oparte na komunikatach o błędach. Możesz utworzyć alert na podstawie jednej z dwóch rzeczy — numeru błędu lub poziomu błędu.

Alerty oparte na numerze błędu mogą dotyczyć np. błędu numer 9002 (pełny dziennik) lub 1105 (brak miejsca na dysku). Alert można zgłaszać dla wybranej bazy lub dla wszystkich baz. Możliwe, że dany alert jest istotny tylko wtedy, gdy wypełnił się dziennik transakcji produkcyjnej bazy danych *Production*, natomiast nie ma znaczenia, jeśli to zdarzenie dotyczy innych testowych



Rysunek 5.22. Okno dialogowe New Alert

lub rozwojowych baz danych. W takiej sytuacji wybierz bazę *Production* z listy rozwijanej *Database name*. Jeśli chcesz otrzymywać alert dotyczący dwóch wybranych baz, natomiast nie chcesz zgłaszać go dla pozostałych baz, musisz utworzyć dwa odrębne alerty. System SQL Server obecnie nie obsługuje alertów dla kilku wybranych baz.

Każdy błąd ma też poziom krytyczności i można utworzyć alert oparty na tym poziomie. I tak błędy z poziomu 19. i wyższych to krytyczne błędy serwera. Prawdopodobnie będziesz chciał otrzymywać alerty o krytycznych błędach systemu SQL Server. W tym celu utwórz alerty dla każdego poziomu krytyczności od 19. do 25.

Przy stosowaniu różnych kombinacji alertów dla numerów błędów i poziomów krytyczności musisz pamiętać, że alerty dla numerów błędów są traktowane priorytetowo. Oznacza to, że jeśli utworzysz jeden alert dla numeru błędu z 16. poziomu krytyczności i drugi dla wszystkich błędów z tego poziomu, zgłaszany będzie tylko alert dla numeru błędu. Alerty dla poziomów krytyczności są więc alertami rezerwowymi. Po wystąpieniu błędu zgłaszany jest alert powiązany z numerem tego błędu. Dla innych błędów z tego poziomu krytyczności, jeśli trzeba, zgłaszany jest alert ustawiony dla danego poziomu.

Jeśli utworzysz dla jednej bazy dwa alerty dotyczące tego samego numeru błędu lub poziomu krytyczności, zgłaszany będzie tylko jeden z dostępnych alertów. Załóżmy, że utworzyłeś alert dla numeru błędu 50001 dla bazy *Production* i alert dla tego samego numeru błędu dla wszystkich baz (opcja *<all databases>*). Wtedy przy wystąpieniu tego błędu w bazie *Production* zgłoszony zostanie tylko alert ustawiony dla tej bazy; alert dla wszystkich baz zostanie zignorowany. Alert dla wszystkich baz jest przesyłany po wystąpieniu błędu numer 50001 w dowolnej bazie różnej od *Production*. Wynika z tego, że lokalna obsługa zdarzeń jest używana zamiast bardziej ogólnych alertów.

Możesz też opracować alert z dodatkowymi warunkami dotyczącymi tekstu komunikatu. W tym celu utwórz alert w sposób opisany na początku punktu „Alerty”, ale zaznacz pole *Raise alert when message contains* i wpisz w polu tekstowym łańcuch znaków. Alert będzie wtedy zgłaszany dla komunikatów obejmujących wpisany tekst. Możesz np. utworzyć zdarzenie zgłaszane, gdy w komunikacie o błędzie znajduje się tekst *Powiadom Adama na pager*. Aplikacja może wtedy zgłaszać błędy użytkownika powodujące przesłanie tego alertu i powiadomienie Adama na pager. Obowiązują tu opisane wcześniej zasady — w reakcji na przesłanie komunikatu o podanym tekście zgłoszony zostanie powiązany alert. Ogólny alert zostanie wywołany tylko dla komunikatów bez pasującego tekstu.

UWAGA Alerty systemu SQL Server działają na podstawie dziennika zdarzeń aplikacji systemu operacyjnego. Jeśli dane zdarzenie nie zostanie zapisane w dzienniku, alert nie zadziała. Możesz zastosować systemową procedurę składowaną `sp_altermessage` i ustawić parametr `@parameter = 'write_to_log'`, aby rozpocząć rejestrowanie zdarzeń.

Komunikaty o błędach można tworzyć za pomocą procedury składowanej `sp_addmessage`. Ponadto możesz określić, czy komunikat ma zostać zapisany w dzienniku. Poniższy kod w języku SQL tworzy prosty komunikat:

```
sp_addmessage 50001,16 , 'MESSAGE', @with_log = 'TRUE'
```

Numer błędu w tym komunikacie to 50001, a poziom krytyczności jest równy 16. Następnie możesz utworzyć alerty sprawdzające system. Ustaw je tak, aby wysyłały e-maile. W celu przetestowania alertu użyj następującego kodu:

```
Raiserror(50001,16,1)with log
Select * from msdb.dbo.sysmail_allitems
```

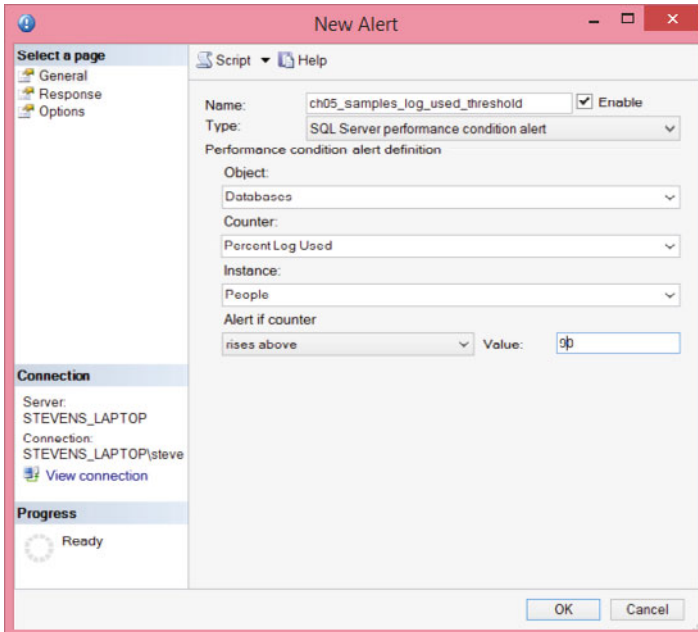
Instrukcja `Raiserror` wysyła komunikat o błędzie. Jeśli masz odpowiednie uprawnienia, możesz przy użyciu tej samej instrukcji zapisać komunikat w dzienniku.

Polecenie `Select` wyświetla wszystkie e-maile. Przewiń ich listę do końca, aby sprawdzić, czy pojawiło się na niej powiadomienie wygenerowane w odpowiedzi na alert.

Alerty dotyczące warunków związanych z wydajnością w systemie SQL Server

Gdy instalujesz system SQL Server, dodawane są też liczniki narzędzia Windows Performance Monitor. To narzędzie umożliwia pracownikom działu operacyjnego śledzenie wydajności serwerów, w tym obciążenia procesora, zużycia pamięci itd. W trakcie instalowania systemu SQL Server dodawana jest kolekcja liczników, za pomocą których administratorzy baz danych mogą śledzić wydajność i stan egzemplarzy tego systemu. Możesz przygotować alert dotyczący warunku związanego z dowolnym licznikiem dla systemu SQL Server. Więcej informacji o monitorowaniu systemu SQL Server przy użyciu liczników wydajności znajdziesz w rozdziale 12., „Monitorowanie systemu SQL Server”. Alert dla warunku związanego z wydajnością systemu SQL Server przedstawiono na rysunku 5.23.

UWAGA Nie można tworzyć alertów dotyczących wielu liczników. Nie możesz np. dodać alertu zgłaszanego, gdy dziennik jest zajęty w ponad 80% (licznik *Percent Log Used*), a liczba transakcji na sekundę przekracza 100 (licznik *Transactions/sec*). Alerty muszą dotyczyć pojedynczych liczników.



Rysunek 5.23. Alert dotyczący wydajności

Liczniki wydajności są grupowane według obiektów. Przykładowo obiekt *Databases* obejmuje liczniki dla konkretnej bazy danych (m.in. *Percent Log Used* i *Transactions/sec*). Obiekt *Buffer Manager* zawiera liczniki dotyczące zarządzania buforem.

Aby rozpocząć tworzenie alertu, wykonaj następujące kroki.

1. Wybierz obiekt, a następnie licznik, dla którego chcesz opracować alert.

UWAGA Nie można tworzyć alertów systemu SQL Server dla liczników, które nie są powiązane z tym systemem (np. dla liczników dotyczących obciążenia procesora). Do budowania tego rodzaju alertów służy narzędzie Performance Monitor.

2. Następnym wyborem czeka na Ciebie w polu tekstowym *Instance*. Jeśli wybrałeś obiekt *Databases*, w polu tekstowym *Instance* zobaczysz listę baz danych. Wybierz bazę, dla której chcesz utworzyć alert.
3. Następnie przyjrzyj się polu *Alert if Counter*. System może wysyłać alerty, gdy licznik spadnie poniżej podanej wartości, stanie się jej równy lub wzrośnie powyżej określonej wartości. Potrzebną wartość ustaw w polu tekstowym *Value*.

Choć możesz utworzyć alert informujący o zapełnieniu dziennika transakcji, nie jest to idealne rozwiązanie, ponieważ w tym podejściu wiadomość dociera do operatora zbyt późno. Lepiej byłoby otrzymać informację w sytuacji, gdy istnieje zagrożenie zapełnienia dziennika. W tym celu można utworzyć alert dla warunku związanego z wydajnością — zastosować licznik *Percent Log Used* dla obiektu *Databases* odpowiadającego odpowiedniej bazie.

1. Wybierz poziom przekroczenia przez wspomniany licznik bezpiecznej wartości (np. między 80 a 95%). Dzięki temu otrzymasz powiadomienie, zanim dziennik się zapełni.

2. Dostosuj wartość w taki sposób, aby nie otrzymywać powiadomień zbyt wcześnie. Jeśli uważasz, że ustawiłeś odpowiednią wielkość dziennika, możliwe, że zechcesz otrzymywać powiadomienia o automatycznym powiększaniu go.

Alerty dotyczące zdarzeń narzędzia WMI

WMI (ang. *Windows Management Instrumentation*) to mechanizm o bardzo dużych możliwościach, a jednocześnie najmniej znana technologia generowania alertów.

W systemie SQL Server 2005 wprowadzono dostawcę WMI Provider for Server Events, który przekształca kwerendy z języka WQL (ang. *WMI Query Language*) na powiadomienia o zdarzeniach z konkretnej bazy danych. Więcej informacji o stosowaniu powiadomień o zdarzeniach zamieszczono w rozdziale 12., „Monitorowanie systemu SQL Server”.

Aby utworzyć alert o zdarzeniach WMI, wybierz opcję *WMI event alert* jako typ alertu, sprawdź poprawność przestrzeni nazw, a następnie wpisz kwerendę w języku WQL.

Reagowanie na alerty

Wcześniej napisaliśmy, że w reakcji na alert można uruchomić zadanie narzędzia SQL Server Agent lub powiadomić operatorów. Podejmowane działania można skonfigurować w zakładce *Response* w oknie dialogowym *Create Alert*. Aby uruchamiać zadanie, zaznacz odpowiednią opcję i wybierz istniejące zadanie lub utwórz nowe. Jeśli chcesz powiadamiać operatora, zaznacz właściwą opcję i wybierz operatorów, do których chcesz wysyłać informacje za pomocą poszczególnych metod. W przypadku alertów warto ustawić operatora dla każdej zmiany. Wymaga to skonfigurowania powiadomień na pager w godzinach pracy odpowiednich osób, co zostało opisane w punkcie „Operatorzy”, wcześniej w tym rozdziale.

Jak wykorzystać te możliwości we własnej firmie? Załóżmy, że dziennik transakcji się zapełnia. Możesz ustawić alert wydajności, aby powiadamiać operatorów o tej sytuacji, a także uruchamiać zadanie, które powoduje powiększenie dziennika. Ponadto można przygotować alert, który tworzy kopię zapasową dziennika, gdy ten wypełni się w 80%.

Oto możliwa sytuacja. Jesz właśnie obiad, gdy otrzymujesz na pager powiadomienie o tym, że dziennik jest w 70% zapełniony. Automatycznie uruchamiane jest wtedy zadanie, które próbuje utworzyć kopię zapasową dziennika w celu zwolnienia pamięci. Za kilka minut otrzymujesz na pager informację o udanym wykonaniu tego zadania. Po kilku frytkach pager znów się odzywa — okazuje się, że dziennik jest zapełniony już w 80%. Wcześniejsze tworzenie kopii zapasowej nie spowodowało zwolnienia pamięci. Możliwe, że powodem jest długa transakcja. Ponownie uruchomione zostanie zadanie tworzące kopię zapasową, a po chwili otrzymujesz powiadomienie o jego zakończeniu. Do zakończenia obiadu nie dostajesz żadnych powiadomień. Oznacza to, że ostatnie zadanie zwolniło pamięć i system działa poprawnie.

Po pewnym czasie możesz otrzymać na pager informację, że dziennik ponownie jest prawie pełny i został powiększony automatycznie lub został rozszerzony na dysku awaryjnym. Prawdopodobnie powinieneś wrócić do pracy, jednak automatyczne narzędzia skonfigurowane w systemie zajmowały się już problemem, gdy jadłeś obiad, i powiadamiały o każdym kroku. Dobre przemyślenie takich scenariuszy pozwala uwzględnić wiele planowanych reakcji, co upraszcza pracę i umożliwia szybsze wykonywanie potrzebnych operacji.

Na stronie *Alert Options* w oknie dialogowym *Create Alert* możesz wykonać kilka czynności.

- **Określenie, w jakiej sytuacji należy umieścić w powiadomieniu szczegółowe informacje.** Czasem tekst komunikatu o błędzie bywa długi. Ponadto może obowiązywać limit liczby

znaków wyświetlanych na danym urządzeniu. W niektórych pagerach ten limit to tylko 32 znaki. Nie należy więc przysyłać długich komunikatów tekstowych na urządzenia (chodzi tu głównie o pagery), które nie mogą wyświetlić wszystkich informacji.

- **Dodawanie informacji do powiadomień.** W tym oknie dialogowym znajduje się duże pole tekstowe o nazwie *Additional Notification Message to Send*. Możesz w nim wpisać dowolny tekst, który chcesz dodać do powiadomienia. Dobrym przykładem jest „Rusz się, przyjeźdź i natychmiast napraw ten problem”.
- **Opóźnienie między odpowiedziami.** W dolnej części okna dialogowego możesz ustawić długość opóźnienia między odpowiedziami. Domyślnie opóźnienie jest równe 0. Wyobraź sobie sytuację, gdy alert jest zgłaszany wielokrotnie w krótkim czasie. Możliwe, że program raz za razem wywołuje instrukcję `Raiserror` lub zgłaszany jest alert dla warunku związanego z wydajnością. Problem ten wyjątkowo często występuje przy alertach dotyczących ograniczonej ilości zasobów. W systemie zaczyna brakować pamięci, co powoduje zgłoszenie alertu lub uruchomienie zadania, co dodatkowo zajmuje pamięć. Może to prowadzić do ponownego zgłoszenia alertu i zużycia jeszcze większej ilości pamięci, co skutkuje wielokrotnym wysyłaniem powiadomień na pager.

Możesz kliknąć prawym przyciskiem myszy dowolny obiekt powiązany z narzędziem SQL Server Agent i utworzyć skrypt, który usunie lub utworzy taki obiekt. Jeśli chcesz, aby ten sam obiekt znajdował się na wielu serwerach, możesz zapisać operację tworzenia obiektu w skrypcie, zmienić w nim nazwę serwera i przenieść skrypt na ten serwer. To jednak oznacza, że trzeba zadbać o synchronizację operatorów, zadań, alertów i jednostek pośredniczących na wielu serwerach, co jest bardzo kłopotliwe i grozi popełnieniem błędów. Gdy zarządzasz wieloma serwerami, pracę może ułatwić mechanizm przekazywania zdarzeń. Zadania wykonywane na wielu serwerach i przekazywanie zdarzeń omawiamy dalej w tym rozdziale, w podrozdziale „Zarządzanie wieloma serwerami”.

Zabezpieczenia w narzędziu SQL Server Agent

Zabezpieczenia w narzędziu SQL Server Agent można ustawiać bardzo precyzyjnie. W tym miejscu opisujemy nie tylko konta usługowe, ale też zabezpieczenia związane z tym, kto może tworzyć, przeglądać i uruchamiać zadania tego narzędzia. W systemie SQL Server 2014 obsługiwanych jest wiele odrębnych kont pośrednich, które można powiązać z różnymi krokami zadań. Te konta pośrednie mają ustawione loginy systemu SQL Server, co pozwala dokładnie kontrolować kroki różnego typu.

Konto usługowe

Jeśli zamierzasz korzystać z funkcji Database Mail lub potrzebujesz połączenia z siecią, kontem usługowym narzędzia SQL Server Agent powinno być konto domeny. Konto powinno być powiązane z loginem, do którego na serwerze przypisana jest rola sysadmin.

Dostęp do narzędzia SQL Server Agent

Po zainstalowaniu systemu tylko osoby, do których przypisana jest rola sysadmin, mają dostęp do obiektów narzędzia SQL Server Agent. Inne osoby nie widzą nawet tych obiektów w oknie

Object Explorer programu Management Studio. Aby zapewnić innym dostęp do narzędzia SQL Server Agent, trzeba przypisać im jedną z trzech ról w bazie msdb:

- SQLAgentUserRole,
- SQLAgentReaderRole,
- SQLAgentOperatorRole.

Pierwsza z tych ról daje najmniejsze możliwości, a rola SQLAgentOperatorRole zapewnia największe uprawnienia. Każda z wyższych ról obejmuje uprawnienia powiązane z niższymi rolami, dlatego użytkownikowi wystarczy przypisać jedną z ról.

UWAGA Osoby z przypisaną rolą sysadmin serwera mają dostęp do wszystkich możliwości narzędzia SQL Server Agent, dlatego nie trzeba im przypisywać żadnej z wymienionych ról.

Rola SQLAgentUserRole

Osoby przypisane do tej roli mają najbardziej ograniczony dostęp do narzędzia SQL Server Agent. Mogą jedynie zobaczyć zawartość węzła *SQL Server Agent/Jobs* i uzyskać dostęp do lokalnych zadań oraz harmonogramów, których są właścicielami. Nie mogą stosować opisanych dalej w tym rozdziale zadań dla wielu serwerów. Mogą tylko tworzyć, modyfikować, usuwać, wykonywać, uruchamiać i zatrzymywać własne zadania oraz harmonogramy. Mogą też wyświetlić historię własnych zadań, choć nie mogą jej usunąć. Ponadto mają uprawnienia do wyświetlania i wybierania operatorów powiadamianych o zakończeniu wykonywania ich zadań, a także do ustawiania jednostek pośredniczących potrzebnych w krokach zadań.

Rola SQLAgentReaderRole

Rola ta obejmuje wszystkie uprawnienia poprzedniej roli. Osoby przypisane do tej roli mogą tworzyć i uruchamiać obiekty opisane w poprzednim podpunkcie, a ponadto widzą listę zadań dla wielu serwerów (w tym ich właściwości i historię). Mają też dostęp do wszystkich zadań i harmonogramów z lokalnego serwera (poprzednia rola pozwala zobaczyć tylko własne obiekty). Mogą jednak wyświetlić tylko węzły z kategorii *SQL Server Agent/Jobs*.

SQLAgentOperatorRole

Ta rola wprowadza najmniejsze ograniczenia i obejmuje wszystkie uprawnienia wcześniej opisanych ról. Zapewnia dodatkowe możliwości w zakresie wyświetlania i wykonywania obiektów. Osoby przypisane do tej roli mogą wyświetlać właściwości jednostek pośredniczących i operatorów. Mają dostęp do listy wszystkich dostępnych jednostek pośredniczących i alertów z serwera. Osoby te mogą też wykonywać, uruchamiać i zatrzymywać lokalne zadania, a także włączać i wyłączać zadania oraz operatorów, choć muszą do tego używać procedur `sp_update_job` i `sp_update_schedule`. Mogą również usunąć historię dowolnego zadania. Osoby, którym przypisano tę rolę, widzą węzły *Jobs*, *Alerts*, *Operators* i *Proxies* z węzła *SQL Server Agent*. Ukryty jest tylko węzeł *Error Log*.

Jednostki pośredniczące narzędzia SQL Server Agent

Jednostki pośredniczące (ang. *proxy*) narzędzia SQL Server Agent definiują kontekst bezpieczeństwa, w którym uruchamiane są różne kroki zadań. Gdy użytkownik, który tworzy zadanie narzędzia SQL Server Agent, nie ma uprawnień do zasobów potrzebnych dla zadania, może ustawić jednostkę pośredniczącą. Zawiera ona dane uwierzytelniające konta systemu Windows mającego uprawnienia do niezbędnych zasobów. Dla kroków zadań z ustawioną jednostką pośredniczącą SQL Server Agent używa konta tej jednostki pośredniczącej i uruchamia te kroki tak, jakby były wykonywane z poziomu określonego konta.

Podsystemy narzędzia SQL Server Agent

Podsystemy narzędzia SQL Server Agent to obiekty, które łączą zbiory podobnych funkcji udostępnianych jednostkom pośredniczącym. Podsystemy te zapewniają granice bezpieczeństwa, co umożliwia zastosowanie bardziej złożonego modelu zabezpieczeń jednostek pośredniczących.

Narzędzie SQL Server ma 12 podsystemów, dla których można ustawić zabezpieczenia. Gdy dodajesz krok zadania, dostępne podsystemy pojawiają się jako typ nowego kroku. Oto wszystkie podsystemy:

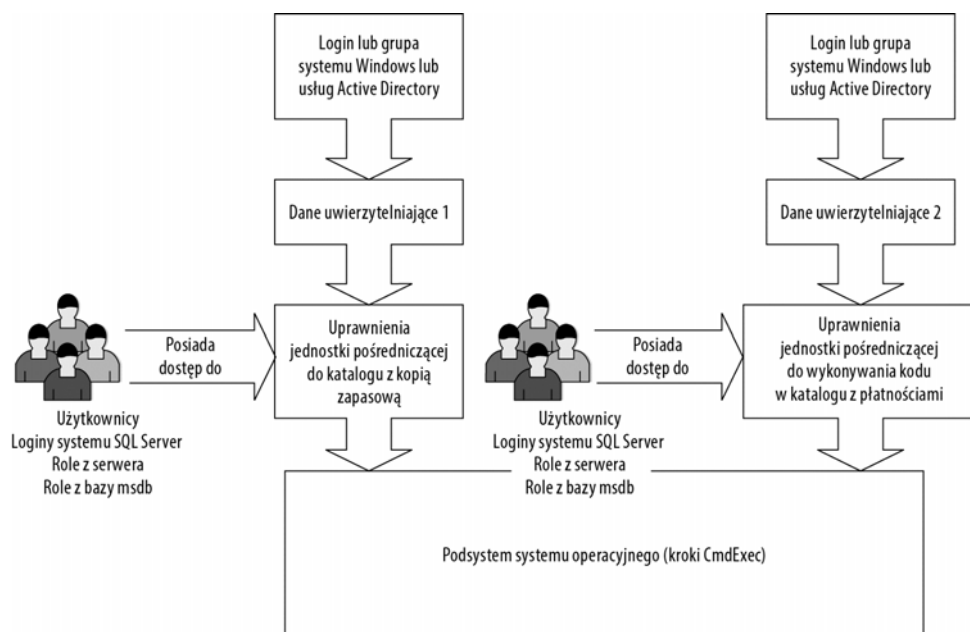
- *ActiveX Script,*
- *Operating System (CmdExec),*
- *PowerShell,*
- *Replication Distributor,*
- *Replication Merge,*
- *Replication Queue Reader,*
- *Replication Snapshot,*
- *Replication Transaction Log Reader,*
- *SQL Server Analysis Services Command,*
- *SQL Server Analysis Services Query,*
- *SQL Server Integration Services Package Execution,*
- *Transact-SQL script.*

Za uprawnienia związane z językiem T-SQL nie odpowiadają jednostki pośredniczące. Wszyscy użytkownicy wykonują kod w tym języku z poziomu własnego konta. Jeśli należysz do grupy sysadmin, możesz jako używane konto wybrać dowolny login z systemu SQL Server. Dla podsystemów innych niż podsystem języka T-SQL uprawnienia są określane na podstawie jednostek pośredniczących.

Uprawnienia podsystemów

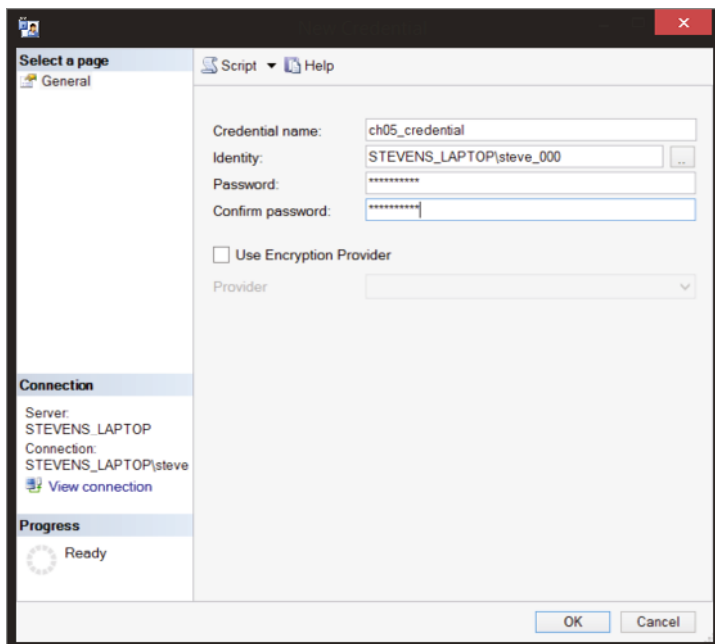
Każdy podsystem ma własne uprawnienia, przy czym jednostka pośrednicząca łączy uprawnienia dla kroku CmdExec i uprawnienia kont powiązanych z tą jednostką. Na rysunku 5.24 pokazano zależności między poszczególnymi elementami.

Ponieważ jednostka pośrednicząca łączy uprawnienia, trudno określić, które uprawnienia systemu operacyjnego są używane, gdy wykonywany jest krok CmdExec zadania. Zmianianie uprawnień jednostek pośredniczących może prowadzić do problemów, dlatego ważne jest, aby od początku odpowiednio skonfigurować uprawnienia. Poniżej opisujemy, jak ustawić uprawnienia dla podsystemu systemu operacyjnego (krok CmdExec).



Rysunek 5.24. Uprawnienia podsystemów

1. Najpierw musisz przygotować dane uwierzytelniające. Najłatwiej zrobić to w programie Management Studio. Rozwiń węzeł *Security*, kliknij węzeł *Credentials* prawym przyciskiem myszy i wybierz opcję *New Credential*. Pojawi się okno dialogowe przedstawione na rysunku 5.25.

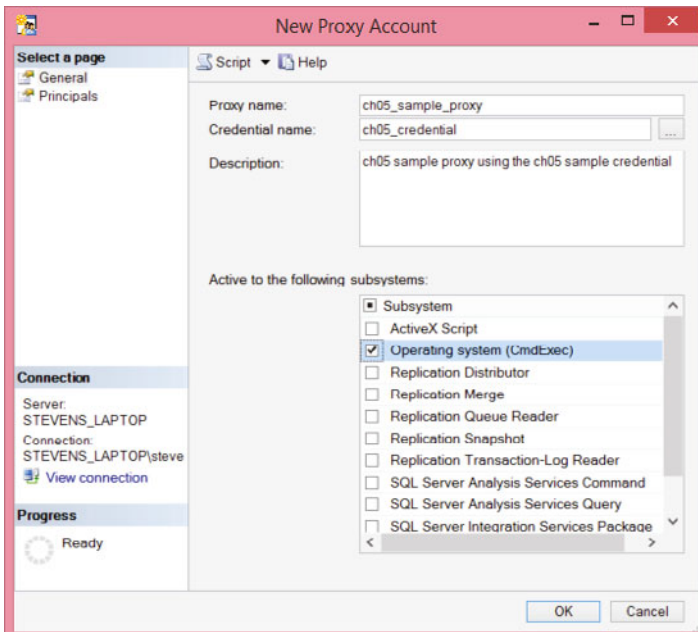


Rysunek 5.25. Ustawianie danych uwierzytelniających

- Przypisz danym uwierzytelniającym przyjazną nazwę i powiąż je z loginem lub grupą systemu Windows. Musisz także podać hasło. Uprawnienia powiązane z loginem lub grupą będą używane przy wykonywaniu kroku CmdExec zadania.

UWAGA Jeśli serwer SMTP wymaga loginu, możesz utworzyć określone konto lokalne z minimalnymi uprawnieniami specjalnie do wysyłania poczty z tego serwera. Takie konto powinno mieć jak najmniejsze uprawnienia i nie powinno być używane do wykonywania jakichkolwiek innych operacji.

- Teraz możesz utworzyć jednostkę pośredniczącą. W programie Management Studio rozwiń węzeł *SQL Server Agent*, kliknij węzeł *Proxies* prawym przyciskiem myszy, a następnie wybierz opcję *New Proxy*. Pojawi się okno dialogowe *New Proxy Account* widoczne na rysunku 5.26.



Rysunek 5.26. Tworzenie nowej jednostki pośredniczącej

- Ustaw dla jednostki pośredniczącej nazwę, która określa jej poziom bezpieczeństwa i przeznaczenie. Następnie powiąż tę jednostkę z danymi uwierzytelniającymi. Jednostka pośrednicząca zapewnia uprawnienia powiązane z używanymi danymi uwierzytelniającymi. Podaj szczegółowy opis funkcji danej jednostki pośredniczącej i określ, jak i kiedy należy z niej korzystać.
- Następnie wybierz podsystemy, które mają stosować daną jednostkę. Jednostkę pośredniczącą można powiązać z wieloma podsystemami.
- Utwórz listę użytkowników, którzy mogą korzystać z danej jednostki pośredniczącej. Służy do tego strona *Principals*. Użytkownikiem może tu być rola z serwera, login systemu SQL Server lub rola z bazy msdb.

7. Teraz założmy, że utworzyłeś dwie jednostki pośredniczące dla podsystemu wykonującego kroki CmdExec (patrz rysunek 5.24). Z oboma powiązany jest login systemu SQL Server. Przyjmijmy, że chcesz utworzyć zadanie z krokiem CmdExec. W trakcie dodawania tego kroku otwórz listę rozwijaną *Run As*. Zawiera ona wszystkie jednostki pośredniczące, które można wykorzystać dla danego kroku. Każda z tych jednostek ma własne uprawnienia. Wybierz jednostkę potrzebną dla danego kroku.

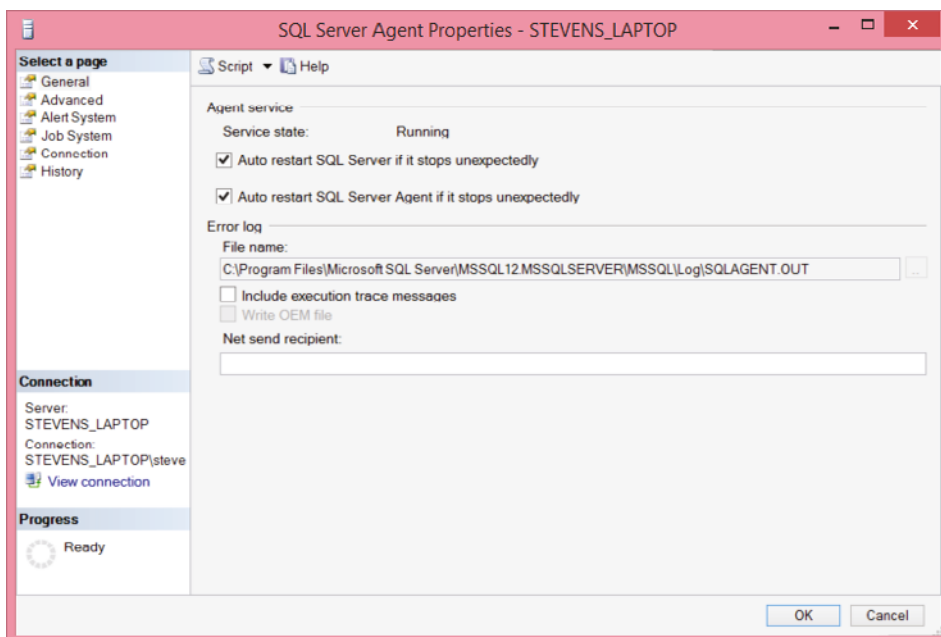
Konfigurowanie narzędzia SQL Server Agent

Skoro wiesz już, jak działa narzędzie SQL Server Agent, możesz przejść do konfigurowania go. Znasz już niektóre opcje, a dalej zapoznasz się z innymi ekranami służącymi do ustawiania właściwości tego narzędzia.

Aby rozpocząć proces konfigurowania, kliknij prawym przyciskiem myszy węzeł *SQL Server Agent* w programie Management Studio i wybierz opcję *Properties*.

Właściwości z kategorii General

Pojawi się strona *General* widoczna na rysunku 5.27.



Rysunek 5.27. Właściwości ze strony General

Przyjrzyj się każdej sekcji z tej strony i zastanów się nad wykonaniem następujących operacji.

- Zaznacz dwa górne pola opcji: *Auto restart SQL Server if it stops unexpectedly* i *Auto restart SQL Server Agent if it stops unexpectedly*. Proces Service Control Manager śledzi usługi SQL Server oraz SQL Server Agent i automatycznie restartuje je, gdy przestaną działać.

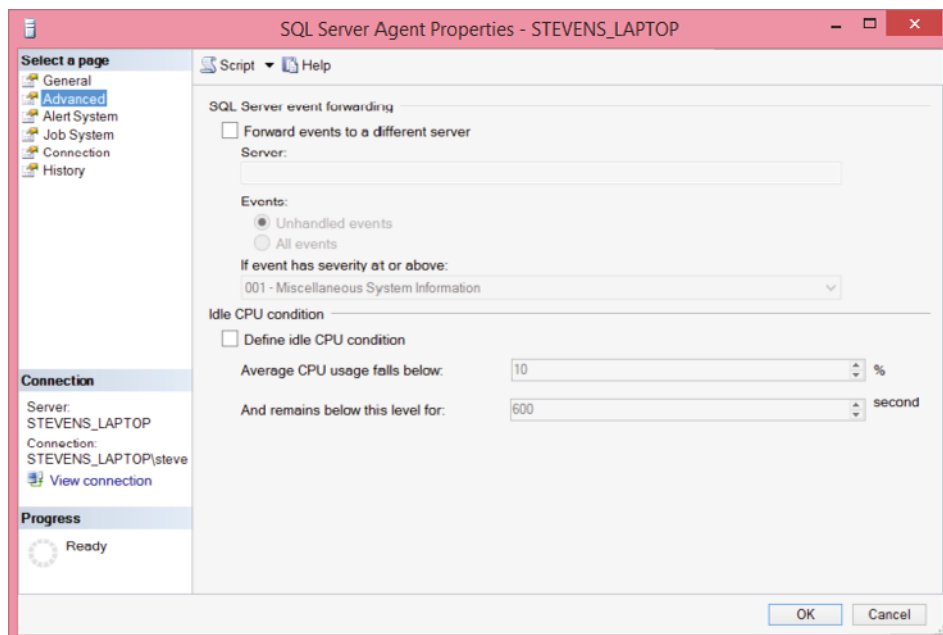
- Zwykle nie trzeba zmieniać domyślnej lokalizacji dziennika błędów, jednak możesz ustawić ją w dowolny sposób. Jeśli potrzebujesz rejestrować dodatkowe dane, zaznacz opcję *Include execution trace messages*.

Komunikaty o śladzie wykonania zapewniają szczegółowe informacje na temat pracy narzędzia SQL Server Agent, zapisywane w dzienniku błędów tego narzędzia. Włączenie wspomnianej opcji powoduje zwiększenie ilości zajętego miejsca w dzienniku narzędzia SQL Server Agent, dlatego trzeba uwzględnić to, czy ten dziennik jest wystarczająco pojemny.

- Aby po zarejestrowaniu błędów otrzymywać powiadomienia za pomocą usługi Net Send, wprowadź nazwę stacji roboczej w polu tekstowym *Net Send recipient*. Na serwerze musi pracować usługa Windows Messaging Service, by usługa Net Send mogła działać.

Właściwości z kategorii Advanced

Otwórz zakładkę *Advanced* w lewym górnym rogu, aby wyświetlić okno dialogowe widoczne na rysunku 5.28.



Rysunek 5.28. Właściwości z kategorii Advanced

Na tym ekranie masz dostęp do kilku opcji.

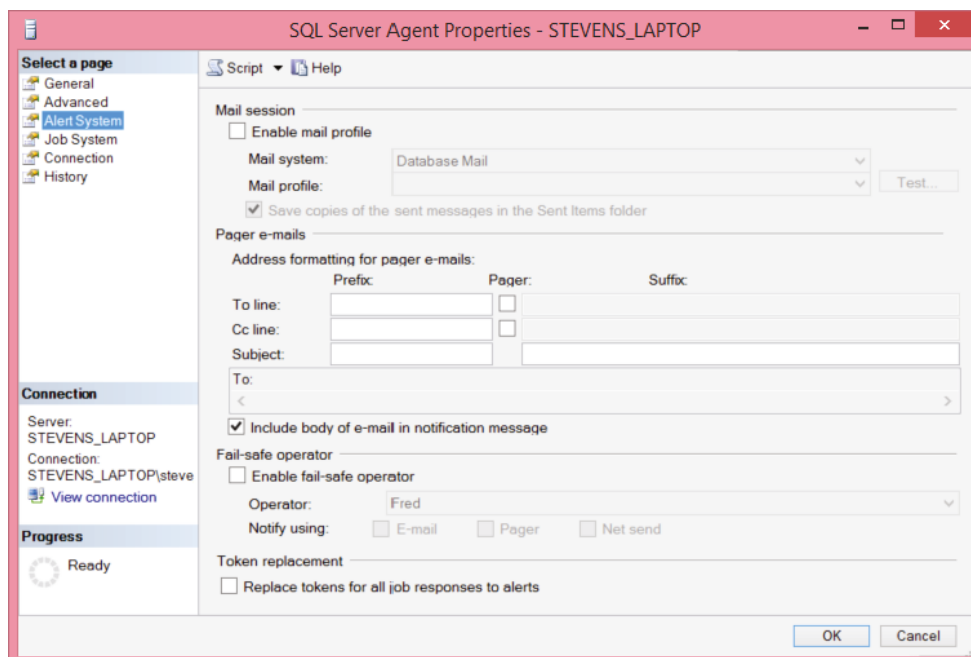
- Górna sekcja, *SQL Server event forwarding*, umożliwia włączenie przekazywania zdarzeń między serwerami. Można ustawić operatorów i alerty na jednym serwerze, a następnie przekazywać do niego zdarzenia z innych serwerów. Aby zastosować ten mechanizm, trzeba znać zasady używania tokenów narzędzia SQL Server Agent. To zagadnienie omawiamy w punkcie „Podstawianie tokenów”, dalej w tym rozdziale. Jeśli chcesz zastosować opisaną funkcję, zaznacz opcję *Forward events to a different server*. Następnie podaj nazwę serwera. Możesz przekazywać wszystkie zdarzenia lub tylko zdarzenia nieobsłużone. *Zdarzenie nieobsłużone* to takie, dla którego nie zdefiniowano alertu.

Możesz też określić, jak krytyczne muszą być przekazywane błędy. Możliwe, że nie zechcesz przekazywać błędów o poziomie krytyczności niższym niż 16. (czyli różnych błędów użytkownika). To, czy błędy o poziomie 16. będą przekazywane, zależy od tego, czy w aplikacji zdefiniowane są błędy ze skonfigurowanymi powiadomieniami.

- Druga sekcja to *Idle CPU condition*. Pamiętaj, że możesz utworzyć harmonogram uruchamiany wtedy, gdy procesor staje się bezczynny. Tu można zdefiniować, czym jest *bezczynność*. Domyślnie uznaje się, że procesor jest bezczynny, gdy przez 10 minut pracuje na poziomie poniżej 10% swoich możliwości.

Właściwości z kategorii Alert System

Następny ekran to *Alert System*. Przedstawiono go na rysunku 5.29.



Rysunek 5.29. Właściwości z kategorii Alert System

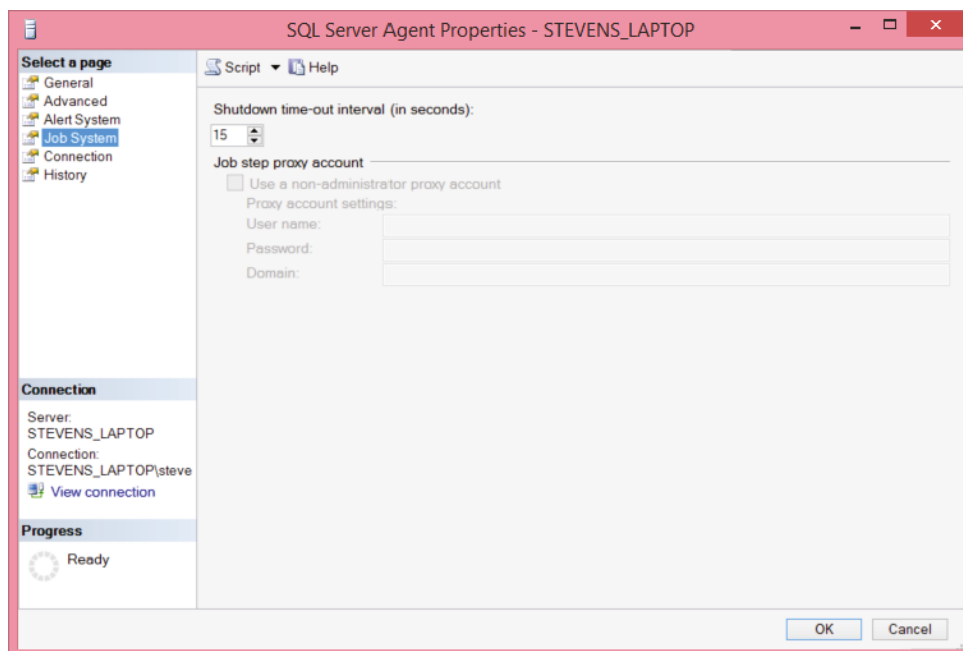
- Jeśli planujesz używać systemu Database Mail, ustaw opcje na tym ekranie. Choć w systemie Database Mail dostępnych może być wiele profili, narzędzie SQL Server Agent używa tylko jednego z nich. W tym miejscu określ używany system pocztowy i profil.
- Druga sekcja dotyczy e-maili wysyłanych na pager. Jeśli używany system pagera wymaga specjalnych znaków kontrolnych w wierszach odbiorcy, odbiorcy kopii i tematu, możesz podać je w tym miejscu przed danym elementem (przedrostki w kolumnie *Prefix*) lub po nim (przyrostki w kolumnie *Suffix*). W trakcie wprowadzania zmian zobaczysz ich działanie w małym polu pod sekcją przeznaczoną na informacje. Możesz też dodać lub wykluczyć treść e-maila z wiadomości przesyłanej na pager. W tym celu zaznacz odpowiednią opcję.
- Trzecia sekcja zawiera ustawienia związane z operatorami awaryjnymi. Jeśli wysyłasz jakiegokolwiek powiadomienia, skonfiguruj tę sekcję. Łatwo przez przypadek zmienić

harmonogram w taki sposób, że nikt nie otrzyma powiadomienia, a ustawienie operatora awaryjnego pozwala się przed tym zabezpieczyć. Włącz główną opcję, wybierz operatora i określ, jakim kanałem mają być przesyłane do niego informacje (za pomocą e-maili, na pager, usługą Net Send lub przy użyciu kombinacji tych kanałów).

- Ostatnia opcja pozwala określić, czy tokeny mają być podmieniane w zadaniach uruchamianych w reakcji na zgłoszenie alertu. Szczegółowe omówienie podstawiania tokenów zamieszczono w podrozdziale „Zarządzanie wieloma serwerami”, dalej w tym rozdziale.

Właściwości z kategorii Job System

Następny ekran to *Job System* (patrz rysunek 5.30).



Rysunek 5.30. Właściwości z kategorii Job System

- W pierwszej sekcji w polu *Shutdown time-out interval* można określić limit czasu oczekiwania na zamknięcie (w sekundach). Załóżmy, że próbujesz zamknąć narzędzie SQL Server Agent w momencie, gdy wciąż działają zadania. W tym miejscu możesz określić, ile czasu narzędzie powinno czekać na wykonanie zadań przed ich zamknięciem i zakończeniem pracy.
- Druga sekcja jest dostępna tylko wtedy, gdy zarządzasz narzędziem SQL Server 2000 Agent. W ten sposób można ustawić zgodną wstecz nieadministracyjną jednostkę pośredniczącą. W wersji SQL Server 2000 obsługiwana była tylko jedna taka jednostka. Wersje SQL Server 2005, 2008, 2012 i 2014 zawierają wiele jednostek pośredniczących, dlatego ta sekcja nie jest potrzebna przy zarządzaniu narzędziem SQL Server Agent dla tych wersji.

Właściwości z kategorii Connection

Większość użytkowników nie potrzebuje strony *Connection*. Narzędzie SQL Server Agent domyślnie nawiązuje połączenie z systemem SQL Server i używa do tego nazwy serwera, domyślnego portu, swojego konta usługowego oraz najwyższego obsługiwanego protokołu zarówno występującego w konfiguracji klienta, jak i włączonego w systemie SQL Server. Poniżej podane są warunki, w których warto zmienić ustawienia domyślne.

- W serwerze działa kilka kart sieciowych, a administrator chce ustawić konkretny adres IP lub port.
- Administrator chce używać w połączeniach konkretnego protokołu (np. IP).
- Administrator chce, aby narzędzie SQL Server Agent łączyło się z serwerem za pomocą loginu innego niż login konta usługowego.

Aby utworzyć alias dla systemu SQL Server, wykonaj następujące kroki.

1. Otwórz narzędzie Connection Manager.
2. Rozwiń węzeł *Sql Native Client Configuration*, kliknij prawym przyciskiem myszy węzeł *Aliases* i wybierz opcję *New Alias*.
3. Wpisz wybrany przez siebie alias.
4. Na stronie *SQL Server Agent Connection* wprowadź alias i informacje o połączeniu dla narzędzia SQL Server Agent. Choć dozwolone jest stosowanie uwierzytelniania systemu SQL Server, nie jest to zalecane podejście.

Właściwości z kategorii History

Ostatni ekran to *History*, przedstawiony wcześniej na rysunku 5.18. W tym miejscu możesz ograniczyć wielkość dziennika historii zadań do określonej liczby wierszy. Bardzo dużym udogodnieniem jest opcja *Maximum job history rows per job* (czyli maksymalna liczba wierszy historii na jedno zadanie). Wyobraź sobie, że zadanie jest uruchamiane wielokrotnie. Możliwe, że użytkownik zaplanował wykonywanie go co sekundę lub jest ono uruchamiane w reakcji na wielokrotnie zgłaszany alert. W takiej sytuacji wpisy dotyczące tego zadania mogą zapełnić całą historię, dlatego nie znajdziesz w niej informacji o innych zadaniach. Trudno będzie wtedy diagnozować inne zadania. Właśnie takim sytuacjom ma zapobiegać opcja *Maximum job history rows per job*. Jej domyślna wartość to 100 wierszy, jednak można ją zmienić w zależności od potrzeb.

Zarządzanie wieloma serwerami

System SQL Server 2014 udostępnia kilka technik, które pozwalają łatwo zarządzać paroma serwerami. Techniki te polegają głównie na centralizowaniu zarządzania. Aby uzyskać pożądany efekt, należy przekazywać zdarzenia do centralnego serwera zarządzania zdarzeniami, który umożliwia scentralizowanie obsługi alertów. Inne podejście to wykorzystanie serwera nadrzędnego i serwerów docelowych. W tym modelu jeden serwer nadrzędny tworzy zadania i uruchamia je na wielu serwerach docelowych.

Podstawianie tokenów

System SQL Server 2014 udostępnia wygodne funkcje związane z tokenami zadań narzędzia SQL Server Agent. *Token* to literał używany w krokach zadaniach (skryptach w języku T-SQL, krokach CmdExec lub skryptach Active Script). Przed wykonaniem zadania narzędzie SQL Server Agent podstawia za token wartość. Tokeny są obsługiwane tylko w zadaniach narzędzia SQL Server Agent.

Jednym z dostępnych tokenów jest (STRDT). Możesz np. dodać poniższy kod w kroku zadania w języku T-SQL:

```
PRINT 'Data uruchomienia zadania(RRRRMMDD): $ESCAPE_NONE(STRDT))';
```

Jeśli zapiszesz dane wyjściowe, zobaczysz następujące informacje:

```
Data uruchomienia zadania(RRRRMMDD):20141223
```

UWAGA W tokenach ma znaczenie wielkość znaków.

Poniższa lista zawiera wybrane tokeny, które możesz zastosować w dowolnym zadaniu.

- (DATE) — bieżąca data w formacie RRRRMMDD.
- (INST) — nazwa egzemplarza systemu SQL Server. Dla egzemplarza domyślnego ten token zwraca pusty łańcuch znaków.
- (JOBID) — identyfikator zadania narzędzia SQL Server Agent.
- (MACH) — nazwa komputera, na którym uruchomiono dane zadanie.
- (MSSA) — nazwa nadrzędnej usługi SQLServerAgent.
- (OSCMD) — przedrostek określający program używany do uruchamiania kroków CmdExec zadań.
- (SQLDIR) — katalog, w którym zainstalowany jest system SQL Server. Katalog domyślny to *C:\Program Files\Microsoft SQL Server\MSSQL*.
- (STEPCT) — liczba wykonań danego kroku. Wartość tę można wykorzystać w pętli, aby zakończyć wykonywanie kroku po określonej liczbie powtórzeń. Ten licznik nie uwzględnia ponawianych prób po nieudanych uruchomieniach kroku. Aktualizacja odbywa się po każdym uruchomieniu kroku zadania (licznik działa w czasie rzeczywistym).
- (STEPID) — identyfikator kroku zadania.
- (SVR) — nazwa serwera na komputerze z systemem SQL Server (w tym nazwa egzemplarza).
- (TIME) — aktualny czas w formacie GGMMSS.
- (STRTTM) — czas rozpoczęcia wykonywania zadania GGMMSS.
- (STRDT) — data rozpoczęcia wykonywania zadania RRRRMMDD.

Poniżej znajduje się lista tokenów, które są dostępne tylko dla zadań uruchomionych w reakcji na alert. Jeśli te tokeny pojawiają się w zadaniu uruchomionym w inny sposób, zadanie zgłosi błąd.

- (A-DBN) — nazwa bazy danych, w której zgłoszony został alert.
- (A-SVR) — nazwa serwera, na którym zgłoszony został alert.
- (A-ERR) — numer błędu powiązany z alertem.
- (A-SEV) — poziom krytyczności błędu powiązanego z alertem.
- (A-MSG) — tekst komunikatu powiązany z alertem.

Poniższy token jest dostępny tylko w zadaniach uruchomionych w odpowiedzi na alert usługi WMI (patrz punkt „Używanie usług WMI”, dalej w tym rozdziale).

- `(WMI(właściwość))` — zwraca wartość dla właściwości WMI o podanej nazwie. Przykładowo `$(WMI(DatabaseName))` zwraca wartość właściwości `DatabaseName` z alertu usług WMI, który spowodował uruchomienie zadania.

Wszystkich wymienionych tokenów trzeba używać w makrach kontrolujących znaki specjalne. Ma to na celu zwiększenie bezpieczeństwa przy stosowaniu tokenów z nieznanymi źródłami. Przyjrzyj się poniższemu tokenowi, który można zastosować w zadaniu w kroku z kodem w języku T-SQL:

```
Print 'Komunikat o błędzie: $(A-MSG)'
```

Ten krok jest uruchamiany w wyniku wystąpienia błędu użytkownika (`Raiserror`). Złośliwy użytkownik może zgłosić błąd w następującej postaci:

```
Raiserror('';Delete from dbo.Employee',16,1)
```

Zwrócony błąd będzie wyglądał tak:

```
';Delete from dbo.Employee
```

Wyświetlany komunikat przyjmie następującą postać:

```
Print 'komunikat o błędzie: '';Delete from dbo.Employee
```

Taka sytuacja oznacza, że padłeś ofiarą ataku przez wstrzyknięcie kodu w języku SQL. Jeśli krok z kodem w języku T-SQL w zadaniu ma odpowiednie uprawnienia, wykonana zostanie instrukcja `Delete`.

Aby zapobiec takim atakom, należy zastosować makro kontrolujące znaki specjalne. Ponieważ w instrukcji `print` używane są apostrofy, do ataku przez wstrzyknięcie kodu w SQL-u wystarczy dodać apostrof zamykający, a następnie wstawić własny kod w SQL-u. Jeśli chcesz zapobiec takim atakom, możesz dodać nowy apostrof dla każdego apostrofu występującego w tokenie. Tak właśnie działa makro `ESCAPE_SQUOTE`. Stosuje się je w następujący sposób:

```
Print 'Komunikat o błędzie: $(ESCAPE_SQUOTE(A-MSG))'
```

W omawianym przykładzie spowoduje to utworzenie poniższej instrukcji:

```
Print 'Komunikat o błędzie: '';Delete from dbo.Employee
```

Niedopasowane apostrofy spowodują błąd, a wykonywanie kroku zakończy się niepowodzeniem, co zapewnia bezpieczeństwo.

Poniżej znajduje się lista makr kontrolujących znaki specjalne.

- `$(ESCAPE_SQUOTE(token))` — zastępuje apostrof (') dwoma jego wystąpieniami w podstawianym łańcuchu znaków.
- `$(ESCAPE_DQUOTE(token))` — zastępuje cudzysłów (") dwoma jego wystąpieniami w podstawianym łańcuchu znaków.
- `$(ESCAPE_RBRACKET(token))` — zastępuje prawy nawias kwadratowy (]) dwoma jego wystąpieniami w podstawianym łańcuchu znaków.
- `$(ESCAPE_NONE(token))` — podstawia dane za token bez wprowadzania żadnych zmian. Ta opcja jest potrzebna tylko w celu zachowania zgodności wstecz.

Jeśli wykorzystasz właściwe typy danych, możesz też bezpośrednio stosować potrzebne wartości. Zadanie obejmujące wykonywany w pętli skrypt w SQL-u z tokenami znajdziesz w poniższym kodzie, który kończy krok zadania po jego pięciokrotnym wykonaniu. Pierwszy wiersz przekształca wartość z tokenu `STEPCT` na liczbę całkowitą, zatem można ją wykorzystać w porównaniu.

Następnie token `JOBID` zadania jest przekształcany na typ `binary(16)` i przekazywany do procedury składowanej `sp_stop_job`, która przyjmuje identyfikator zadania przeznaczonego do zamknięcia.

```
IF Convert(int,(ESCAPE_NONE(STEPCT))) >5
BEGIN
    DECLARE @jobid binary(16)
    SELECT @jobid =Convert(Uniqueidentifier,(ESCAPE_NONE(JOBID)))
    EXEC msdb.dbo.sp_stop_job @job_id = @jobid
END
```

Pomyśl, jak wykorzystać tokeny oparte na alertach. Możesz utworzyć alert związany z wydajnością systemu SQL Server zgłaszany, gdy dziennik transakcji dowolnej bazy danych (ustawienie *<any database>*) zostanie zapełniony w ponad 80%. Utwórz zadanie z następującym krokiem z kodem w języku T-SQL:

```
DECLARE @a varchar(100)
SELECT @a = 'BACKUP LOG $(ESCAPE_QUOTE(A-DBN))
    TO DISK = "\\UNCName\Share\$(ESCAPE_QUOTE(A-DBN))\log.bak"'
SELECT @a
BACKUP LOG $(ESCAPE_QUOTE(A-DBN))
    TO DISK = '\\UNCName\Share\\$(ESCAPE_QUOTE(A-DBN))\log.bak'
```

W tym fragmencie `UNCName` to nazwa serwera, na którym ma znaleźć się kopia zapasowa, a `Share` to nazwa współużytkowanego zasobu na serwerze. Upewnij się, że alert prowadzi do uruchomienia tego zadania. Gdy alert zostanie zgłoszony dla bazy `NorthAmerica`, polecenie tworzenia kopii zapasowej będzie wyglądało tak:

```
BACKUP LOG NorthAmerica TO DISK = \\UNCName\Share\\NorthAmerica\log.bak
```

Najpierw musisz utworzyć potrzebny katalog i przyznać uprawnienia do niego używanej jednostce pośredniczącej. Możesz przygotować krok `CmdExec`, który na bieżąco będzie dodawał odpowiednie katalogi. Dzięki temu jedno zadanie będzie mogło utworzyć kopię zapasową dowolnego dziennika transakcji. Aby poprawić czytelność nazwy katalogu utworzonego w kroku `CmdExec`, można dodać do niej datę i czas.

Przekazywanie zdarzeń

W kontekście używania zdarzeń i alertów można utworzyć operatorów i alerty w jednym systemie, a następnie przekazywać zdarzenia z innych systemów do jednego, centralnego systemu SQL Server obsługującego alerty i reagującego na nie, jeśli będzie trzeba.

Docelowy serwer dla zdarzeń można ustawić na ekranie *Advanced* okna dialogowego z właściwościami narzędzia SQL Server Agent (patrz rysunek 5.28). Zaznacz opcję *Forward events to a different server*, a następnie ustaw serwer, do którego mają trafiać zdarzenia.

Za pomocą opcji z sekcji *Events* możesz ustawić, które zdarzenia mają być przekazywane. Dostępne opcje to *Unhandled events* (czyli nieobsłużone zdarzenia) i *All events* (czyli wszystkie zdarzenia). Jeśli wybierzesz opcję *All events*, możesz dodać filtr oparty na poziomie krytyczności *If event has severity at or above*. Ponadto można ustawić operatorów z głównego systemu zarządzania zdarzeniami. Utwórz zadania, które mają reagować na alerty. Następnie w nadrzędnym systemie zarządzania zdarzeniami dodaj alerty, które mają być zgłaszane po wystąpieniu określonych zdarzeń. W utworzonych zadaniach możesz wykorzystać tokeny narzędzia SQL Server Agent, aby ustalić, na którym serwerze i w której bazie danych wystąpiło pierwotne zdarzenie.

Używanie usług WMI

Usługi WMI to zbiór funkcji z jądra systemów operacyjnych i serwerów firmy Microsoft (w tym systemu SQL Server). Umożliwiają one lokalne i zdalne monitorowanie serwerów oraz zarządzanie nimi. Usługi WMI są oparte na standardach i zgodne ze specyfikacjami WBEM (ang. *Web-Based Enterprise Management*) i CIM (ang. *Common Information Model*) organizacji DMTF (ang. *Distributed Management Task Force*).

WMI to rozbudowane usługi, którym warto poświęcić całą książkę. Ważne jest to, że usługi WMI udostępniają wiele zdarzeń związanych z systemem SQL Server. Otwórz dokumentację Books Online i poszukaj informacji o WMI. Znajdziesz w ten sposób wiele zdarzeń. Możesz tworzyć związane z nimi alerty. Dostępne są np. zdarzenia języka DDL (ang. *Data Definition Language*), zgłaszane w momencie tworzenia lub usuwania baz danych i tabel.

Dla usług WMI powstał specjalny język WQL (ang. *WMI Query Language*) przeznaczony do sprawdzania obsługiwanych zdarzeń. Przypomina on język T-SQL i jest tak prosty, że szybko nauczysz się nim posługiwać.

Poszukaj w dokumentacji Books Online artykułu *WMI Provider for Server Events Classes and Properties*. Ten materiał pomoże Ci zapoznać się z licznymi dostępnymi zdarzeniami i wybrać konkretne, które będziesz obserwował. Dla każdego zdarzenia istnieje lista atrybutów (podobnie w tabelach znajdują się listy kolumn). Gdy używasz usług WMI, możesz posługiwać się atrybutami zdarzeń w alertach.

Aby utworzyć alert, użyj programu SQL Server Management Studio.

1. W oknie *Object Explorer* otwórz węzeł *SQL Server Agent*, kliknij prawym przyciskiem myszy węzeł *Alerts* i wybierz opcję *New Alert*. Na liście rozwijanej *Alert Type* wybierz opcję *WMI Event Alert*. Przestrzeń nazw wypełni się na podstawie danych z podłączonego serwera i powinna wyglądać w następujący sposób:

```
\\.\root\Microsoft\SqlServer\ServerEvents\SQL2014
```

Kropka (.) reprezentuje nazwę serwera, którą możesz zmienić. Może to być np. nazwa `\\MYSQLSERVER\`. Ostatnim członem nazwy powinno być `MSSQLSERVER` dla egzemplarzy domyślnych lub `<nazwa egzemplarza>` dla egzemplarzy o określonej nazwie. W tym przykładzie nazwa egzemplarza to `SQL2014`.

2. W dostępnym polu tekstowym możesz wpisać kwerendę w języku WQL:

```
SELECT * FROM DDL_DATABASE_LEVEL_EVENTS
```

Aby pobrać tylko atrybut `TSQLCommand`, zastosuj następującą kwerendę:

```
Select TSQLCommand from DDL_DATABASE_LEVEL_EVENTS
```

3. Kliknij przycisk *OK*. Nastąpi chwila przerwy. Jeśli użyta została błędna przestrzeń nazw, nieprawidłowa składnia lub niewłaściwa nazwa atrybutu albo zdarzenia, natychmiast zobaczysz informujący o tym komunikat.
4. Następnie w zadaniu można zastosować token `WMI` (atrybut). W tym przykładzie wygląda to tak:

```
Print '$(ESCAPE_SQUOTE(WMI(TSQLCommand)))'
```

5. Aby uzyskać informacje o zdarzeniach z bazy danych, w bazie włączone muszą być powiadomienia Service Broker. Żeby je włączyć dla bazy `NorthAmerica`, zastosuj następujące polecenie:

```
ALTER DATABASE NorthAmerica SET ENABLE_BROKER;
```


Jeśli alerty zostały zgłoszone, ale nie nastąpiło podstawienie tekstu za token usług WMI, prawdopodobnie musisz włączyć funkcję Service Broker dla danej bazy.

UWAGA Konto usługowe narzędzia SQL Server Agent musi mieć uprawnienia do używanej przestrzeni nazw i uprawnienia ALTER ANY EVENT NOTIFICATION. Jeśli skonfigurowałeś konta za pomocą programu SQL Server Configuration Manager, potrzebne uprawnienia są zapewniane automatycznie. Aby dostosować je ręcznie, w oknie *Uruchamianie* wpisz **wmimgmt.msc**. Pojawi się wtedy administracyjne okno dialogowe, w którym można ustawić uprawnienia.

Jeśli chcesz wypróbować usługi WMI, na serwerze znajdziesz specjalny program testowy. Aby uruchomić go w wierszu poleceń, wpisz **WBEMTest**. Program jest zainstalowany w katalogu *WBEM*, w katalogu systemu Windows. Microsoft w swojej witrynie udostępnia całą sekcję poświęconą usługom WMI, tam możesz dowiedzieć się więcej na ich temat. Wystarczy poszukać informacji o WMI w witrynie <http://www.microsoft.com>.

Zarządzanie wieloma serwerami — używanie serwerów nadrzędnych i docelowych

System SQL Server umożliwia skonfigurowanie serwera nadrzędnego (ang. *master server* — MSX). Serwer nadrzędny może przysyłać zadania do wykonania do serwerów docelowych (ang. *target server* — TSX), ale sam nie może wtedy pełnić funkcji serwera docelowego otrzymującego zadania z innego serwera nadrzędnego. Serwery docelowe odbierają i uruchamiają zadania z jednego serwera nadrzędnego (oprócz tego wykonują własne zadania lokalne). W środowisku może działać wiele serwerów nadrzędnych, ale serwer docelowy może być powiązany z tylko jednym serwerem nadrzędnym. Jest to prosta dwupoziomowa hierarchia: serwer może być nadrzędny, docelowy lub nie pełnić żadnej z tych funkcji. Serwery docelowe są *dodawane* (ang. *enlist*) lub *zwalniane* (ang. *defect*).

Konfigurowanie serwerów jest łatwe. Wystarczy wykonać następujące kroki.

1. W programie SQL Server Management Studio kliknij węzeł *SQL Server Agent* prawym przyciskiem myszy, wybierz opcję *Multi-Server Administration*, a następnie *Make This a Master*.
2. Po pierwszym oknie dialogowym zobaczysz ekran, w którym możesz podać adres e-mail, adres pagera i lokalizację odbiorcy usługi Net Send, aby skonfigurować *operatora serwera nadrzędnego*. Odpowiednio uzupełnij te pola. Wybrany operator zostanie ustawiony dla serwera nadrzędnego i wszystkich serwerów docelowych. Jest to *jedyny* operator, którego można powiadamiać o zadaniach przesyłanych do wielu serwerów.
3. W następnym oknie dialogowym można wybrać wszystkie serwery docelowe. Na liście znajdują się serwery zarejestrowane w programie SQL Server Management Studio. Jeśli nie zarejestrowałeś żadnych serwerów, najpierw otwórz widok serwerów zarejestrowanych (opcja *View/Registered Servers*). Następnie kliknij prawym przyciskiem myszy węzeł *Local Servers Group* i wybierz opcję *New Server Registration*. Pojawi się okno dialogowe *New Server Registration*, w którym możesz ustawić nowy serwer.

Gdy zarejestrowane serwery będą już dostępne, możesz wybrać, które z nich mają być serwerami docelowymi. Następnie kliknij przycisk *Next*. Aby dodać kolejne serwery, kliknij przycisk *Add Connection*.

4. Zamknij okno dialogowe. System SQL Server sprawdzi, czy wersje serwerów nadrzędnego i docelowych są zgodne ze sobą. Jeśli są niezgodne, usuń problematyczny serwer docelowy z listy, a następnie kontynuuj pracę. W przyszłości możesz zaktualizować serwery docelowe i nadrzędny, tak aby używane na nich wersje były kompatybilne.
5. Przejdź do następnego okna dialogowego i użyj kreatora, aby (jeśli to konieczne) utworzyć login na serwerze docelowym i przyznać mu uprawnienia do serwera nadrzędnego. Serwery docelowe muszą łączyć się z serwerem nadrzędnym, aby przekazywać informacje o stanie zadań. Po zakończeniu konfiguracji odśwież węzły narzędzia SQL Server Agent i przejrzyj zmiany. Pojawi się informacja o serwerze nadrzędnym i serwerze docelowym.

Teraz możesz utworzyć zadania przeznaczone do wykonywania na wielu serwerach docelowych. Zauważ, że na serwerze nadrzędnym w węźle *Jobs* występują dwie sekcje — z zadaniami lokalnymi i zadaniami dla wielu serwerów. Aby utworzyć zadanie, wykonaj następujące kroki.

1. Kliknij prawym przyciskiem myszy węzeł *Multi-Server Jobs* i wybierz opcję *New Job*, aby utworzyć proste zadanie.
2. Utwórz proste zadanie na serwerze nadrzędnym i ustaw je tak, by było uruchamiane na serwerach docelowych (jednym lub kilku). Koniecznie zajrzyj przy tym na stronę z powiadomieniami. Jedyny operator, który może otrzymywać powiadomienia o omawianych zadaniach, to *MSXOperator*.

Tworzenie zadań wykonywanych na wielu serwerach to dobry sposób na zarządzanie dużymi systemami bez konieczności zakupu dodatkowych rozwiązań niezależnych producentów. Nikt na docelowym serwerze nie może modyfikować przesyłanych zadań. Za pomocą programu SQL Server Management Studio możesz teraz połączyć się z serwerem docelowym jako administrator i przejrzeć właściwości utworzonego wcześniej zadania, które pobrano z serwera nadrzędnego. Możesz wyświetlić takie zadanie, sprawdzić jego historię, a nawet je uruchomić. Nie możesz jednak usunąć zadania, zmienić harmonogramu jego wykonywania, zmodyfikować kroków itd. To zadanie nie należy do Ciebie — należy ono do serwera nadrzędnego.

Gdy zaczniesz zastanawiać się nad tym, jak zastosować ten model, koniecznie uwzględnij skutki uruchamiania jednego zadania na wielu serwerach. Wszelkie referencje do katalogów, baz danych i innych elementów muszą być poprawne na wszystkich serwerach docelowych. Możesz np. utworzyć jeden współużytkowany zasób, na którym zapisywane będą wszystkie kopie zapasowe.

Ponieważ zadania mogą uruchamiać inne zadania, można opracować zadanie nadrzędne z krokiem wywołującym inne zadanie. To inne zadanie jest tworzone na każdym serwerze docelowym i jest specyficzne dla każdego z tych serwerów. Dzięki temu w razie potrzeby można dostosować zadania do serwerów. Aby utworzyć zadanie nadrzędne, wykonaj następujące kroki.

1. W programie SQL Server Management Studio kliknij prawym przyciskiem myszy węzeł *SQL Server Agent* z serwera nadrzędnego.
2. Wybierz opcję *Multi-Server Administration*. W oknie dialogowym, które się pojawi, możesz dodawać serwery docelowe i zarządzać nimi.

3. Kliknij *Manage Target Servers*. W wyświetlonym oknie dialogowym możesz śledzić stan różnych obiektów. Gdy stworzysz zadanie dla serwera docelowego, jest ono automatycznie pobierane na ten serwer. Jeśli wartość licznika nieodczytanych instrukcji nie spadnie do 0, wyślij zapytanie do docelowego serwera. To wzbudzi serwer i sprawi, że pobierze instrukcje.
4. Kliknij odpowiednią zakładkę, aby wyświetlić informacje o pobranych instrukcjach. W ten sposób zobaczysz szczegółowe dane na temat czasu pobrania i aktualizacji zadań.
5. Za pomocą przycisku *Post Instructions* z okna dialogowego *Target Server Status* możesz zsynchronizować zegary między serwerami, zwolnić docelowe serwery, ustawić interwały odpytywania i uruchamiać zadania. Zadanie możesz też wywołać bezpośrednio za pomocą węzła *Jobs* z serwerów nadrzędnego lub docelowego.
6. Możesz wyświetlić historię zadania na serwerze nadrzędnym (podobnie jak w przypadku innych zadań), jednak nie zobaczysz szczegółowych informacji o krokach tego zadania. W celu zapoznania się z takimi informacjami wyświetl historię zadania na serwerze docelowym.
7. Serwer docelowy możesz zwolnić z poziomu węzła *SQL Server Agent* z serwera docelowego lub w oknie dialogowym *Manage Target Servers* z serwera nadrzędnego. Po zwolnieniu wszystkich serwerów docelowych serwer nadrzędny przestaje pełnić tę funkcję.

Podsumowanie

Automatyzowanie pracy systemu SQL Server to jedna z najważniejszych rzeczy, jakich możesz się nauczyć, aby uprościć swoje obowiązki i zapewnić płynne działanie firmy. Plany konserwacji pomagają łatwo automatyzować rutynowe operacje i są doskonałym sposobem na rozpoczęcie automatyzowania standardowych zadań z obszaru konserwacji. Ponadto narzędzie SQL Server Agent udostępnia liczne funkcje i usługi upraszczające pracę. Już samo opracowanie kilku zadań tworzących kopie zapasowe i wysyłających powiadomienia do operatorów pozwala zautomatyzować wiele codziennych czynności. Jeśli chcesz stosować bardziej wymyślne rozwiązanie, nic nie stoi na przeszkodzie, jednak najpierw opracuj odpowiedni plan (zrób to zwłaszcza wtedy, kiedy myślisz o używaniu zadań dla wielu serwerów).

Stosowanie alertów to doskonały sposób na automatyzowanie powiadomień o ważnych wydarzeniach w systemie bazodanowym. Możesz wysyłać powiadomienia na pager oraz wykorzystać harmonogramy z godzinami pracy do wysyłania zwykłych e-maili i informacji na pager. Jest to dobry mechanizm zapewniający, że powiadomienia trafią do właściwych osób. Jeśli powiadomienia o alertach ma otrzymywać wielu operatorów, pomyśl o utworzeniu grup odbiorców e-maili i przerzuceniu niektórych operacji związanych z powiadamianiem na serwer poczty elektronicznej. Zacznij od prostych rozwiązań i nie spiesz się. Gdy lepiej poznasz plany konserwacji i narzędzie SQL Server Agent, będziesz mógł przejść do bardziej wyrafinowanych pomysłów.

W rozdziale 6., „Service Broker w systemie SQL Server 2014”, przeczytasz o usłudze Service Broker w systemie SQL Server 2014.

Service Broker w systemie SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Asynchroniczne przetwarzanie danych.
- Konfigurowanie usługi Service Broker za pomocą języka T-SQL.
- Wysyłanie i odbieranie komunikatów przy użyciu usługi Service Broker.

W tym rozdziale opisujemy obiekty usługi Service Broker systemu SQL Server. W trakcie lektury zapoznasz się z różnymi typami obiektów i sposobami ich tworzenia. W końcowej części rozdziału znajdziesz kod w języku T-SQL pozwalający korzystać z takich obiektów przy wysyłaniu i odbieraniu komunikatów przekazywanych w ramach bazy systemu SQL Server, między bazami z tego samego serwera i między bazami z różnych serwerów.

Asynchroniczne przesyłanie komunikatów

Sercem większości systemów przetwarzających duże ilości danych jest *kolejka*, w której żądania są przechowywane do czasu zwolnienia się zasobów potrzebnych do ich obsłużenia. Przy użyciu kolejek ograniczona liczba wątków roboczych może obsługiwać żądania tysięcy klientów. Wątek roboczy pobiera żądanie z kolejki, przetwarza je, a następnie zwraca wynik. Potem przechodzi do obsługi następnego żądania z kolejki.

Jest to dużo wydajniejsze niż uruchamianie nowego wątku w celu przetwarzania każdego żądania, ponieważ nie trzeba ponosić kosztów tworzenia i zamykania wątku. Ponadto kolejki pozwalają na wydajniejsze wykorzystanie zasobów, bo gdy liczba nadsyłanych żądań przekracza możliwości systemu, żądania są przechowywane w kolejce do momentu zwolnienia niezbędnych zasobów. Kolejki wykorzystuje się do zarządzania danymi wejściowymi np. w serwerach WWW

i serwerach baz danych. Żądania kierowane przez klienty do serwera SQL Server nie są zapisywane w kolejkach, jednak przy przetwarzaniu żądań za pomocą wątków roboczych używane są asynchroniczne kolejki, co zwiększa skalowalność systemu.

Na serwerach WWW i w systemie SQL Server używane są *kolejki działające w pamięci*. Jeśli nastąpi przerwa w zasilaniu lub zdarzy się awaria systemu, dane z kolejki zostaną utracone. Użytkownik zobaczy wtedy komunikat o przekroczeniu limitu czasu oczekiwania lub o błędzie i będzie mógł ponowić żądanie.

W wielu aplikacjach można poprawić wydajność po zastosowaniu *trwałych kolejek*, co umożliwia oddzielenie od siebie komponentów aplikacji. Przykładowo aplikacja służąca do handlu akcjami może interaktywnie przyjmować i składać zlecenia, a następnie umieszczać je w kolejce, aby w tle można było wykonać potrzebne rozliczenia. W tej sytuacji transakcja jest zawierana przed rozpoczęciem przetwarzania w tle, więc kolejka przechowywana w pamięci się nie sprawdzi, ponieważ spowoduje utratę danych. Idealnym rozwiązaniem jest przechowywanie kolejki w bazie danych, ponieważ potrzebne informacje można wtedy umieścić w ramach przetwarzania transakcji i usuwać podczas przetwarzania danych o niej w tle. Model ACID (ang. *Atomicity, Consistency, Isolation, Durability*, czyli atomowość, spójność, izolacja i trwałość) gwarantuje, że dane nie zostaną utracone przed pełnym zakończeniem ich przetwarzania.

Choć zalety przechowywania kolejki w tabelach bazy danych są oczywiste, technika ta ma także wady. W wysoce równoległych systemach działać mogą setki wątków, które dodają elementy do kolejki i pobierają je z niej. Często prowadzi to do blokad i zakleszczenia. Jednym z głównych zadań usługi Service Broker jest obsługa kolejek będących pełnoprawnymi obiektami systemu SQL Server. Ponieważ kolejki usługi Service Broker są częścią silnika bazy danych, usługa ta ma bezpośredni dostęp do blokad, zamków i pamięci podręcznej. Dzięki temu kolejki usługi Service Broker są dużo wydajniejsze od zwykłych kolejek przechowywanych w tabelach.

Service Broker to coś więcej niż kolejki zintegrowane z bazą danych. Usługa ta zapewnia infrastrukturę potrzebną do tworzenia niezawodnych, asynchronicznych usług bazodanowych, które można wykorzystać podczas budowania aplikacji trudnych do uzyskania bez takich usług. Od tego wzięła się nazwa „Service Broker” (czyli broker usług) — jest to broker niezawodnych, asynchronicznych usług bazodanowych.

Aplikacje oparte na Service Brokerze, podobnie jak inne aplikacje usługowe, muszą działać w trybie rozproszonym. Aby uzyskać ten efekt bez rezygnacji z niezawodności kolejek Service Brokera, dane trzeba przekazywać między kolejkami za pomocą solidnego mechanizmu przesyłania komunikatów. Service Broker korzysta z opracowanego przez Microsoft protokołu przekazywania komunikatów, który za pomocą kolejek Service Brokera gwarantuje niezawodne przesyłanie danych z zachowaniem integralności transakcji.

Zintegrowanie niezawodnego protokołu przekazywania komunikatów z silnikiem systemu SQL Server zapewnia niespotykaną do tej pory wydajność przesyłania komunikatów dotyczących transakcji. Wiele osób uważa, że Service Broker to solidny mechanizm przesyłania informacji. Choć Service Broker przekazuje komunikaty, to ponadto zapewnia kompletne, oparte na usługach środowisko do budowania architektury dla baz danych. Pamiętaj o tym w trakcie poznawania funkcji tego narzędzia. Niektóre części interfejsu Service Brokera mogą wydawać się dziwne programistom, którzy chcą tylko przesyłać komunikaty między serwerami. Jednak te same elementy stają się zrozumiałe, gdy potraktujemy usługę Service Broker jak środowisko do budowania niezawodnych, asynchronicznych usług bazodanowych.

Service Broker z systemu SQL Server przydaje się w różnych scenariuszach. Był stosowany z powodzeniem w aplikacjach, które wykonują w czasie rzeczywistym operacje ETL (ang. *extract, transform, load*) na bazach OLTP i hurtowniach danych, w aplikacjach bankowych, gdzie obsługiwał conocne przetwarzanie wsadowe danych o transakcjach, oraz w mediach społecznościowych, gdzie obsługiwał żądania dodania do listy znajomych w serwisie MySpace. Usługi te są używane także w wielu innych aplikacjach, w których trzeba w niezawodny sposób wykonywać polecenia, ale niekoniecznie wtedy, gdy zostały zgłoszone.

Ogólne omówienie Service Brokera

Jak opisano to wcześniej, Service Broker umożliwia tworzenie niezawodnych asynchronicznych usług w bazach systemu SQL Server. Usługi Service Brokera przekazują komunikaty w niezawodny i transakcyjny sposób. Aby zapewnić maksymalną elastyczność, Service Broker nie wymusza struktury przekazywanych komunikatów. Obejmują one do 2 gigabajtów danych binarnych. Możesz powiązać z komunikatem XML-owy schemat, a Service Broker sprawdzi na jego podstawie poprawność danych. Nie jest to jednak konieczne. Komunikat może zawierać dowolne dane — od życiorysu po plik JPG.

Komunikaty przesyłane do usług Service Brokera są umieszczane w kolejce powiązanej z daną usługą. Logika przetwarzania znajduje się w procedurze składowanej lub w zewnętrznej aplikacji, która usuwa komunikaty z kolejki, przetwarza je, a następnie zwraca odpowiedź do Service Brokera. Podobnie działają usługi sieciowe w technologii WCF (ang. *Windows Communication Foundation*) z platformy .NET. Usługa sieciowa otrzymuje komunikat w formacie SOAP (ang. *Simple Object Access Protocol*), przetwarza go, a następnie zwraca inny komunikat w tym formacie.

Jednym z problemów dotyczących każdej usługi jest to, że kod trzeba uruchamiać w celu przetwarzania komunikatów wtedy, gdy nadchodzą. Zwykle w tym celu tworzy się jeden lub kilka procesów tła, które działają nieustannie i na bieżąco przetwarzają przychodzące komunikaty. W wielu sytuacjach ten model działa dobrze, jeśli jednak częstotliwość nadchodzenia komunikatów znacznie się zmienia (np. następuje duży wzrost obciążenia, gdy użytkownicy rozpoczynają rano pracę i gdy wracają po przerwie obiadowej do biura), czasem brakuje zasobów na obsłużenie wszystkich komunikatów.

Service Broker rozwiązuje problem braku zasobów za pomocą mechanizmu *aktywacji*. W tle działa proces *monitora aktywacji*, który oczekuje na pojawienie się komunikatów w kolejce. Gdy komunikat nadejdzie, a nie ma dostępnego procesu, który mógłby go przetworzyć, monitor aktywuje odpowiedni proces. Jeśli komunikaty przychodzą szybciej, niż proces odbiorczy potrafi je przetwarzać, monitor uruchamia kolejne procesy do momentu, aż ich liczba pozwala na obsłużenie kolejki. Procesy odbiorcze działają w taki sposób, że gdy wykryją pustą kolejkę, kończą pracę. Dzięki temu liczba procesów odbiorczych rośnie wraz ze wzrostem obciążenia i spada, gdy obciążenie staje się mniejsze. To gwarantuje, że dostępna jest wystarczająca ilość zasobów, aby przetworzyć nadchodzące komunikaty, a jednocześnie pozwala uniknąć marnowania zasobów przy niższym obciążeniu.

Usługi wymagają dwukierunkowej wymiany komunikatów. Żądania są kierowane do usługi, a usługa zwraca odpowiedzi. Service Broker używa trwałych obiektów, tzw. *konwersacji*, aby śledzić wymiany komunikatów. Konwersacja Service Brokera jest tworzona po każdej stronie biorącej udział w wymianie danych i śledzi, jakie komunikaty zostały wysłane, jakie zostały odebrane, jakie zostały potwierdzone, a także jakie potwierdzenia otrzymano. *Punkty końcowe* konwersacji obejmują też adres i informacje związane z zabezpieczeniami, co pozwala niezawodnie i bezpiecznie przekazywać komunikaty między usługami.

Konwersacje są przechowywane jako trwałe obiekty bazy danych i zachowywane przy ponownym uruchamianiu bazy. Konwersacje mogą przetrwać wiele miesięcy, jeśli interakcja między usługami jest tak długa. Przykładowo przetwarzanie zamówienia czegoś tak skomplikowanego jak samolot może potrwać wiele miesięcy i obejmować wymianę tysięcy komunikatów. Tak złożoną interakcję można obsługiwać za pomocą jednej konwersacji Service Brokera. Taka konwersacja gwarantuje, że wszystkie komunikaty są dostarczane w niezawodny i bezpieczny sposób, nawet gdy jeden z punktów końcowych na pewien czas przestanie być aktywny.

Service Broker udostępnia trzy polecenia używane do wysyłania i odbierania komunikatów. Omówienie wszystkich tych instrukcji znajdziesz dalej w tym rozdziale. Pierwsza z nich to `CREATE CONVERSATION`. Tworzy ona konwersację, w ramach której potem przesyłane są komunikaty. Druga to `SEND`. Instrukcja ta służy do wysyłania komunikatów w ramach wcześniej przygotowanej konwersacji. Trzecie polecenie to `RECEIVE`. Jest ono przeznaczone do pobierania komunikatów z kolejki w celu ich przetworzenia.

Dobre w Service Brokerze jest to, że komunikaty są przetwarzane raz (i tylko raz) w kolejności ich przesłania — pod warunkiem, że zostały przekazane w ramach tej samej konwersacji. W większości innych protokołów niezawodnego przekazywania komunikatów limity czasu oczekiwania i ponawiane próby mogą zmienić kolejność dostarczania informacji, a ponadto nie ma gwarancji, że dana wiadomość nie zostanie dostarczona kilkakrotnie. Przesyłanie komunikatów dokładnie raz i zgodnie z pierwotną kolejnością znacznie upraszcza pisanie usług opartych na Service Brokerze. Protokoły używane w Service Brokerze nie zachowują natomiast kolejności komunikatów przesyłanych w różnych konwersacjach.

Service Broker a inne kolejki komunikatów

Dostępne są różne technologie obsługi kolejek komunikatów. Microsoft udostępnia dwa takie narzędzia — Service Broker i MSMQ (ang. *Microsoft Message Queue*). Możesz też zastosować inne technologie niezależnych producentów działające podobnie jak MSMQ.

Ważną różnicą między Service Brokerem a innymi technologiami obsługi kolejek jest to, że Service Broker jest wbudowany w bazę danych, natomiast inne technologie działają jako procesy zewnętrzne. Dzięki temu komunikaty transakcyjne (gdy operacje pobierające i wysyłające komunikaty działają w tej samej transakcji, w której przeprowadzana jest aktualizacja bazy danych spowodowana przetwarzaniem kolejki) są w Service Brokerze obsługiwane dużo wydajniej i bardziej niezawodnie. Service Broker przechowuje komunikaty w bazie danych. Zatem kolejki są transakcyjnie spójne z danymi, które są zapisywane w kopii zapasowej i przywracane. Zaletą tego podejścia jest to, że po przywróceniu bazy danych dostępne będą te same komunikaty, co w momencie tworzenia kopii zapasowej. Gdy używasz kolejek komunikatów, takich jak MSMQ, to po przywróceniu starszej wersji bazy danych komunikaty przetworzone od czasu utworzenia kopii zapasowej zostają utracone.

Nie oznacza to, że Service Broker zawsze jest lepszy od innych technologii obsługi kolejek komunikatów. W niektórych sytuacjach lepiej zastosować usługę Service Broker, a w innych — kolejki komunikatów działające poza bazą danych.

Gdy np. używasz dwóch usług (lub aplikacji) systemu Windows, które muszą przekazywać między sobą komunikaty, a nie potrzebujesz bazy danych, używanie Service Brokera nie jest najlepszym pomysłem. To podejście spowodowałoby zależność od systemu SQL Server, ponieważ baza danych musiałaby być aktywna i dostępna, aby aplikacje mogły przekazywać między sobą informacje za pomocą kolejki Service Brokera. Nie można przysyłać komunikatów

do Service Brokera bez bezpośredniego zalogowania się do bazy danych. W takim scenariuszu kolejka komunikatów, taka jak MSMQ, jest lepszym rozwiązaniem, ponieważ umożliwia przekazywanie informacji bez logowania się do bazy danych.

Większość systemów przekazywania komunikatów, np. MSMQ i MQ Series, może obsługiwać transakcje, jednak transakcje obejmują wtedy dwa różne magazyny danych — miejsce na komunikaty i bazę. Dlatego niezbędna jest transakcja rozproszona z bardziej skomplikowanym dwuetapowym zatwierdzaniem. Systemy nietransakcyjne i systemy bez zapewniania niezawodności są zwykle szybsze niż Service Broker, ponieważ nie wymagają zatwierdzania transakcji. Jeśli jednak potrzebny jest transakcyjny system przekazywania komunikatów, Service Broker jest znacznie szybszy od większości innych rozwiązań tej klasy.

W innej sytuacji, gdy usługa (lub aplikacja) systemu Windows musi przysyłać komunikaty do bazy, a baza przetwarza je bezpośrednio za pomocą języka T-SQL (lub nawet przy użyciu technologii SQLCLR), Service Broker może okazać się dobrym wyborem. Jest tak, ponieważ aplikacja musi zalogować się do bazy danych, aby przekazać do niej komunikat, a dostęp do Service Brokera jest dostępny tylko za pomocą języka T-SQL. Dlatego sensowne jest przechowywanie komunikatów w kolejce, do której można uzyskać bezpośredni dostęp za pomocą języka T-SQL, bez niezależnych, rozszerzonych procedur składowanych.

Konfigurowanie Service Brokera

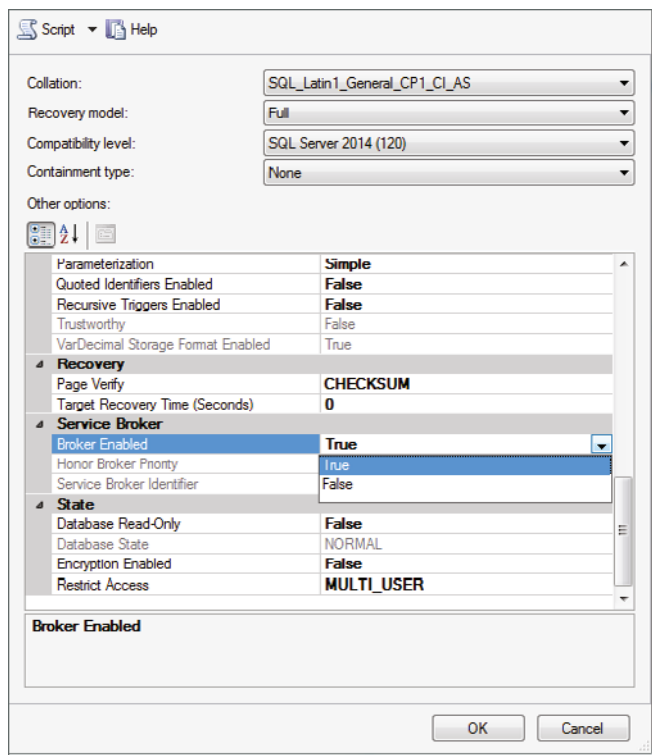
Service Broker to zaawansowana funkcja systemu SQL Server, dlatego w programie SQL Server Management Studio dostępne są tylko podstawowe mechanizmy do konfigurowania tej funkcji. Stan Service Brokera to właściwość bazy danych, którą można ustawić za pomocą okna dialogowego z właściwościami bazy w programie SQL Server Management Studio lub przy użyciu polecenia `ALTER DATABASE` w języku T-SQL. Obiekty Service Brokera możesz tworzyć za pomocą szablonów z sekcji *Service Broker* każdej bazy danych lub przy użyciu poleceń języka T-SQL. Tu koncentrujemy się na instrukcjach języka T-SQL, ponieważ ich najczęściej używa się przy tworzeniu aplikacji opartych na Service Brokerze.

UWAGA Konfigurowanie Service Brokera bywa trudne, ponieważ trzeba poznać wiele nowych pojęć, które komplikują cały proces. Jeśli początkowo nie jesteś pewien, co oznacza dane pojęcie, czytaj dalej — prawdopodobnie wyjaśnienie znajdziesz dalej w tym rozdziale.

Ustawianie stanu Service Brokera

Aby włączyć lub wyłączyć Service Broker za pomocą programu SQL Server Management Studio, najpierw połącz się z odpowiednią bazą danych w oknie *Object Explorer*. Kliknij nazwę bazy prawym przyciskiem myszy, wybierz opcję *Properties*, a następnie otwórz stronę *Options*. Przewiń panel *Other Options* do sekcji *Service Broker*, co przedstawiono na rysunku 6.1. Jedyne dostępne ustawienie to *Broker Enabled*. Przyjmuje ono wartości *True* i *False*.

Service Broker można też włączyć za pomocą języka T-SQL, używając polecenia `ALTER DATABASE`. Oprócz włączenia lub wyłączenia Service Brokera można wtedy dodatkowo ustawić jego stan. Stan służy do zarządzania kopiami bazy danych zawierającymi kolejki Service Brokera.



Rysunek 6.1. Konfigurowanie stanu Service Brokera

Wcześniej wspomniano, że jedną z zalet wbudowania Service Brokera w bazę danych jest to, że można utworzyć i przywrócić kopię zapasową bazy (lub odłączyć i podłączyć bazę) bez utraty komunikatów. Mechanizm ten zapewnia dużą swobodę przy przywracaniu stanu po awarii i przy równoważeniu obciążenia, gdy bazę trzeba przenieść na inny serwer bez utraty niedostarczonych lub nieprzetworzonych komunikatów. Wadą tego modelu jest to, że jeśli zastosujesz proces tworzenia i przywracania (lub podłączania) kopii zapasowej bazy w celu zbudowania jej nowego egzemplarza, otrzymasz dwie bazy systemu SQL Server, które będą próbowały wysyłać i przetwarzać te same komunikaty.

Aby zapobiec próbie wysyłania tych samych komunikatów przez różne bazy danych, przy przywracaniu lub podłączaniu bazy usługa Service Broker domyślnie jest wyłączona. Jeśli baza została przywrócona lub podłączona np. po przerwie spowodowanej brakiem zasilania, w instrukcji RESTORE można zastosować opcję `ENABLE_BROKER`, co spowoduje włączenie Service Brokera po zakończeniu przywracania bazy. Jeżeli baza zostanie przywrócona bez tej opcji, można później skorzystać z polecenia `ALTER DATABASE` i włączyć Service Brokera. Oto przykład:

```
ALTER DATABASE sample_database
SET ENABLE_BROKER
```

Mamy dostęp do kilku opcji służących do zarządzania komunikatami i konwersacjami.

Pierwsza z nich to opcja `NEW_BROKER`, która usuwa wszystkie istniejące konwersacje i komunikaty, a następnie tworzy nowy identyfikator `SERVICE_BROKER_GUID`. Baza danych wygląda wtedy jak nowa baza z Service Brokerem, skonfigurowana dokładnie tak samo jak pierwotna baza, ale z nowym identyfikatorem. Technika przydaje się do skalowania usług Service Brokera, ponieważ

pozwala utworzyć wiele kopii danej bazy. Jest użyteczna także przy tworzeniu kopii zapasowej bazy na potrzeby wykonywania testów lub rozwijania nowych rozwiązań.

Druga opcja to `ERROR_BROKER_CONVERSATIONS`. Nie powoduje ona zmiany identyfikatora Service Brokera, ale prowadzi do ustawienia stanu błędu we wszystkich istniejących konwersacjach. Pozwala to w uporządkowany sposób usunąć konwersacje po obu ich stronach. Technikę tę stosuje się najczęściej po testach, aby mieć pewność, że środowisko z Service Brokerem zostało przywrócone do pierwotnego stanu.

Aby ustalić, czy Service Broker jest włączony, sprawdź wartość kolumny `is_broker_enabled` widoku `sys.databases` z katalogu. Ta kolumna to pole bitowe o wartości 0 lub 1. Wartość 1 oznacza, że zadania Service Brokera są włączone dla danej bazy.

Po włączeniu Service Brokera dla określonej bazy tworzone są potrzebne obiekty i można zacząć wysyłanie oraz odbieranie komunikatów. Gdy Service Broker jest włączony dla danej bazy, uruchamianych jest kilka zadań systemowych zarządzających obiektami Service Brokera w bazie. Jeśli baza nie obejmuje żadnych obiektów Service Brokera, należy wyłączyć tę funkcję, aby wyeliminować niepotrzebne koszty. Możesz też wyłączyć Service Brokera w trakcie konserwacji bazy, aby zapobiec przetwarzaniu komunikatów.

Typy komunikatów

Pierwszym typem obiektu, który trzeba utworzyć w trakcie konfigurowania Service Brokera, jest *typ komunikatu*. Typ komunikatu może być używany w usłudze do określania, jakiego rodzaju dane zawiera komunikat, co pozwala ustalić sposób przetwarzania tych danych. Typy komunikatów opcjonalnie umożliwiają sprawdzanie, czy dane są w odpowiednim formacie. Dostępne są cztery wymienione poniżej ustawienia sprawdzania poprawności.

- **NONE.** Oznacza, że w ciele przesyłanego komunikatu mogą znajdować się dowolne dane. Gdy dla opcji `VALIDATION` nie jest ustawiona żadna wartość, domyślnie używana jest właśnie `NONE`.
- **EMPTY.** Oznacza, że przysyłać można tylko puste komunikaty. Niektóre typy komunikatów o stanie nie muszą zawierać żadnych danych, dlatego ten typ komunikatu chroni kod wywołujący usługę przed wysłaniem danych, które i tak nie będą przetwarzane.
- **WELL_FORMED_XML.** Oznacza, że przysyłać można tylko dobrze uformowane komunikaty w formacie XML. Komunikat jest wtedy przetwarzany przez parser XML-a, a jeśli ten parser wykryje błędy, usługa odrzuci komunikat.
- **VALID_XML WITH SCHEMA COLLECTION.** Oznacza, że używane mogą być tylko dokumenty XML-a pasujące do określonego schematu. Gdy stosujesz opcję `VALID_XML WITH SCHEMA COLLECTION`, potrzebny XML-owy schemat musi znajdować się w bazie danych (do tworzenia schematów służy polecenie `CREATE XML SCHEMA COLLECTION`).

Dwie ostatnie z tych opcji są używane głównie wtedy, gdy usługę może wywołać dowolny klient, który jej potrzebuje. W wielu sytuacjach usługa jest wywoływana tylko przez znany zestaw klientów, dlatego wiadomo, że przychodzące dane są prawidłowe. Wtedy można poprawić wydajność dzięki rezygnacji ze sprawdzania poprawności komunikatów. Sprawdzanie poprawności jest kosztowne, ponieważ komunikaty trzeba wczytać do parsera XML-a.

Polecenie `CREATE MESSAGE TYPE` oprócz ustawień sprawdzania poprawności przyjmuje tylko dwie inne opcje.

- Nazwę typu komunikatu (musi być ona zgodna ze standardowymi regułami tworzenia nazw obiektów obowiązującymi w systemie SQL Server).
- Opcję `AUTHORIZATION`, która określa właściciela tworzonego typu komunikatu.

Gdy osoba tworząca typ komunikatu ma przypisaną rolę `sysadmin` serwera lub rolę `db_owner` bazy danych, wartością opcji `AUTHORIZATION` może być dowolna rola lub dowolny użytkownik bazy danych. Jeśli twórca typu komunikatu nie ma ustawionej żadnej z tych ról, wartością opcji `AUTHORIZATION` musi być dany użytkownik lub inny użytkownik, za którego bieżący może się podawać. Brak wartości parametru `AUTHORIZATION` powoduje, że typ komunikatu należy do bieżącego użytkownika.

W poniższym fragmencie kodu pokazujemy, jak utworzyć typ komunikatu:

```
CREATE MESSAGE TYPE TypKomunikatu
AUTHORIZATION dbo
VALIDATION = WELL_FORMED_XML
```

Kontrakty

Drugi typ tworzonych obiektów to *kontrakt*. Kontrakty definiują typy komunikatów, które można stosować w konwersacjach. Kontrakt gwarantuje, że kod przetwarzający konwersację nie otrzyma komunikatu, którego nie będzie mógł obsłużyć. Kontrakty, podobnie jak typy komunikatów, mają kilka parametrów, które należy ustawić w instrukcji `CREATE CONTRACT`. Oto one.

- Nazwa kontraktu (musi być zgodna z regułami tworzenia nazw obiektów obowiązującymi w systemie SQL Server).
- Wartość opcji `AUTHORIZATION` (musi być to użytkownik lub rola z danej bazy).

Instrukcja `CREATE CONTRACT` wymaga podania listy typów komunikatów powiązanych z kontraktem. Każdy typ komunikatu może być używany tylko przez określonych uczestników konwersacji. Poniżej podane są trzy opcje określające, które obiekty mogą używać poszczególnych typów komunikatów.

- **INITIATOR.** Obiekt `SERVICE` Service Brokera, który zainicjował konwersację za pomocą polecenia `BEGIN DIALOG CONVERSATION`. Szczegółowe omówienie obiektu `SERVICE` znajdziesz dalej w tym rozdziale.
- **TARGET.** Obiekt `SERVICE` Service Brokera ustawiony jako docelowy przez obiekt `INITIATOR`.
- **ANY.** Zarówno obiekt `TARGET`, jak i `INITIATOR` mogą używać danego typu komunikatów. Można podać listę z kilkoma typami komunikatów rozdzielonymi przecinkami, tak jak w poniższym przykładzie:

```
CREATE CONTRACT MyContract
AUTHORIZATION dbo
(YourMessageType SENT BY ANY)
CREATE CONTRACT MyContract
AUTHORIZATION dbo
(YourMessageType SENT BY INITIATOR,
AnotherMessageType SENT BY TARGET)
```

Kolejki

Trzecim typem tworzonych obiektów są *kolejki*. Wcześniej w rozdziale napisano, że w kolejkach Service Brokera przechowywane są komunikaty już odebrane, ale jeszcze nieprzetworzone. Choć pozostałe tworzone obiekty są obiektami logicznymi, którym odpowiadają tylko rekordy w tabelach systemowych, kolejki są obiektami fizycznymi — tabelami, w których przechowywane są komunikaty.

Ponieważ kolejki to tabele fizyczne, jedną z wielu opcji dostępnych osobom tworzącym kolejki jest dodanie grupy plików przechowujących daną kolejkę. Zauważ, że kolejki to tabele ukryte, do których nie ma bezpośredniego dostępu. Możesz wysyłać komunikaty do kolejek (instrukcja SEND) i pobierać komunikaty z kolejek (instrukcja RECEIVE), natomiast nie można stosować do kolejek poleceń INSERT, UPDATE i DELETE.

Przy tworzeniu kolejki można też ustawić kilka innych parametrów. Wszystkie one są opcjonalne — wymagana jest tylko nazwa kolejki.

Pierwsza opcja to stan kolejki (STATUS). Przyjmuje ona wartości ON i OFF. Gdy kolejka jest włączona (ON), może przyjmować komunikaty i je udostępniać. Jeśli kolejka jest wyłączona (OFF), to gdy procedura składowana lub kod w języku T-SQL spróbują pobrać komunikaty z kolejki za pomocą instrukcji RECEIVE, zwrócony zostanie komunikat o błędzie. Wartość domyślna tej opcji to ON.

Druga opcja to RETENTION. Także ona przyjmuje wartości ON i OFF. Przy ustawieniu ON pobrany komunikat nie jest usuwany z kolejki; zostaje w niej zachowany na potrzeby inspekcji. Choć komunikatu nie można pobrać po raz wtóry, pozostaje on zapisany na dysku i można go później wyświetlić za pomocą instrukcji SELECT z nazwą kolejki w klauzuli FROM. Przy ustawieniu OFF komunikaty są usuwane z kolejki bezpośrednio po ich pobraniu.

UWAGA Choć opcja zachowywania komunikatów w kolejkach jest bardzo przydatna w kontekście inspekcji, ma pewną wadę: nie można wtedy łatwo usunąć tylko wybranych danych z kolejki. Możesz wprowadzić za pomocą instrukcji ALTER QUEUE zmienić wartość opcji RETENTION z ON na OFF, ale spowoduje to skasowanie wszystkich informacji z kolejki. Jeśli chcesz pozbyć się tylko wybranych danych — np. wszystkich informacji starszych niż 90 dni, nie istnieje wbudowana technika, która to umożliwi. Zamiast tego trzeba najpierw wyeksportować dane z ostatnich 90 dni do tabeli, a następnie skasować wszystkie informacje z kolejki. Opisywana opcja jest najbardziej przydatna w czasie pisania aplikacji. Zachowane dane można wtedy wykorzystać przy diagnozowaniu problemów, ponieważ dokładnie zapisane są komunikaty otrzymane przez aplikację. W bazach produkcyjnych technika ta nie jest zalecana.

Trzecia opcja to POISON_MESSAGE_HANDLING (czyli obsługa „zatrutych” komunikatów). Jej wartości to ON i OFF. Gdy ta opcja jest włączona (jest to ustawienie domyślne), kolejka zostaje automatycznie wyłączona po pięciu kolejnych nieudanych próbach wycofania transakcji. Jeśli ta opcja jest wyłączona, za obsługę komunikatów odpowiada sama aplikacja.

„Zatrute” komunikaty to częsty problem przy wymianie wiadomości w transakcjach. W trakcie przetwarzania komunikatu błąd może spowodować wycofanie transakcji, co prowadzi do umieszczenia komunikatu z powrotem w kolejce. Jest to korzystne, jeśli przyczyna błędu jest tymczasowa (np. nastąpiło zakleszczenie), ponieważ przy następnym wczytaniu komunikat zostanie z powodzeniem przetworzony. Jeżeli jednak błąd jest trwały (np. aplikacja próbuje

wstawić do tabeli wiersz z powtarzającym się kluczem głównym), komunikat będzie przetwarzany w nieskończoność i uniemożliwi obsługę innych wiadomości. Jest wtedy „zatrutym” komunikatem.

Obsługę „zatrutych” komunikatów należy wbudować w usługę. Przykładowo błąd, którego nie można naprawić za pomocą ponownej próby przetworzenia komunikatu, należy zarejestrować w dzienniku, po czym można zatwierdzić transakcję, aby usunąć „zatrute” komunikat z kolejki. Jeśli „zatrute” komunikat nie zostanie wykryty przez kod usługi, Service Broker domyślnie wyłączy kolejkę po pięciu kolejnych wycofanych transakcjach. Jest to skrajne rozwiązanie, ponieważ cała kolejka przestaje działać. Jeśli jednak usługa jest poprawnie napisana, taka sytuacja nigdy nie powinna mieć miejsca. Jeżeli masz pewność, że aplikacja prawidłowo obsługuje „zatrute” komunikaty, i chcesz uniknąć odbierania w środku nocy telefonów z informacją, że kolejka została zablokowana, możesz wyłączyć automatyczną obsługę „zatrutych” komunikatów.

Czwarta opcja, ACTIVATION, służy do konfigurowania procedury składowanej. Przyjmuje ona cztery ustawienia.

- STATUS. Ustawienie STATUS w opcji ACTIVATION służy do włączania lub wyłączania danej procedury. Gdy opcja ACTIVATION jest wyłączona, nie można tworzyć nowych wątków dla aktywowanej procedury składowanej. Wątki, które już działają, mogą jednak kontynuować pracę.
- PROCEDURE_NAME. Wartością tego parametru jest nazwa aktywowanej procedury składowanej.
- MAX_QUEUE_READERS. Określa maksymalną liczbę egzemplarzy procedury składowanej, jaką można aktywować. Dodatkowe egzemplarze są tworzone, jeśli komunikaty nadchodzą szybciej, niż procedura może je przetworzyć. Aby uniknąć zajęcia zbyt dużej ilości zasobów przez jedną kolejkę, ustaw maksymalną liczbę aktywowanych procedur. Jeśli dla kolejki uruchomionych jest zbyt wiele procedur, liczbę tę można dynamicznie zmodyfikować w trakcie działania Service Brokera.
- EXECUTE AS. Ten parametr określa nazwę użytkownika, którego konta ma używać dana procedura. Dostępne wartości to SELF, OWNER i dowolny użytkownik określonej bazy danych. Wartość SELF oznacza, że stosowane jest konto użytkownika wywołującego daną usługę. Wartość OWNER powoduje wykorzystanie konta użytkownika, który utworzył kolejkę.

W poniższym kodzie zastosowano różne dostępne opcje:

```
CREATE QUEUE YourQueue_Source
WITH STATUS=ON,
    RETENTION=OFF,
    ACTIVATION
        (STATUS=ON,
         PROCEDURE_NAME=dbo.MySourceActivationProcedure,
         MAX_QUEUE_READERS=10,
         EXECUTE AS OWNER),
    POISON_MESSAGE_HANDLING (STATUS = ON);
```

Przy tworzeniu kolejki z aktywowaną procedurą składowaną można skonfigurować tę procedurę w momencie dodawania kolejki (tak jak w pokazanym fragmencie kodu). Jednak procedura musi wtedy istnieć przed utworzeniem kolejki. Dlatego kolejki często tworzy się bez konfigurowania ustawienia ACTIVATION. Następnie można opracować procedurę składowaną i zmodyfikować kolejkę za pomocą polecenia ALTER QUEUE, aby zmienić ustawienia aktywacji. Efekt końcowy w obu podejściach jest dokładnie taki sam — kolejka wywołuje procedurę składowaną po otrzymaniu komunikatu.

Usługi

Obiekty usług Service Brokera definiują interfejs udostępniany przez usługę. Usługi Service Broker to obiekty konfigurowane za pomocą instrukcji `CREATE SERVICE` w języku T-SQL. Usługi określają, jakie kontrakty (a tym samym i typy komunikatów) można stosować przy przysyłaniu wiadomości do określonej kolejki. Komunikat jest przesyłany do konkretnej usługi, która następnie dostarcza dane do odpowiedniej kolejki. Aby przesłać komunikat do usługi Service Brokera, należy podać jako docelową lokalizację nazwę tej usługi. Nazwa kolejki odbierającej komunikaty i trasa przekazywania wiadomości do tej kolejki są ukryte za nazwą usługi. Można zatem zmienić fizyczną implementację usługi Service Brokera bez modyfikowania wywołującej ją aplikacji.

Instrukcja `CREATE SERVICE` udostępnia tylko kilka parametrów. Oto one.

- **Nazwa obiektu.** To ustawienie, podobnie jak w innych obiektach, wymaga zastosowania się do standardów tworzenia nazw. Koniecznie w sensowny sposób stosuj wielkie i małe litery, ponieważ w niektórych sytuacjach wielkość znaków w nazwie usługi ma znaczenie.
- **AUTHORIZATION.** Parametr działa tak samo jak przy tworzeniu typu komunikatu lub kontraktu.
- **Nazwa kolejki.** Nazwa kolejki, do której dostarczane będą komunikaty przesyłane do usługi.
- **Rozdzielona przecinkami lista kontraktów.** Kontrakty należy uwzględnić przy przysyłaniu konwersacji do danej usługi.

Oto przykład:

```
CREATE SERVICE YourService_Source
AUTHORIZATION dbo
ON QUEUE dbo.YourQueue_Source
(MyContract)
GO
```

Gdy stworzysz kolejki dla każdej strony konwersacji, musisz też opracować usługi dla obu stron. Komunikaty są przysyłane od usługi do usługi. Nazwy usług należy podać w instrukcji `BEGIN DIALOG CONVERSATION`, omówionej szczegółowo dalej w tym rozdziale.

UWAGA Gdy tworzysz nazwy usług, nie stosuj słowa `ANY`. W Service Brokerze `ANY` to słowo zarezerwowane, które powoduje zastosowanie priorytetów Service Brokera (opisanych dalej w tym rozdziale) dla wszystkich usług zamiast dla konkretnej usługi o nazwie `ANY`.

Trasy

Obiekty tras Service Brokera służą do definiowania ścieżek w sieci, którymi komunikaty są przysyłane do usług. Trasa obejmuje nazwę usługi, do której kierowany jest komunikat, a także adres sieciowy następnego egzemplarza systemu SQL Server na trasie. Trasa jest kontynuowana do momentu znalezienia serwera, dla którego adres to `LOCAL`. To oznacza, że jest to serwer z szukaną usługą.

Tworzona w każdej bazie trasa `AutoCreatedLocal` jest używana domyślnie dla wszystkich usług, dla których nie ustawiono innej trasy. Dla tej trasy nie ma ustawionej nazwy usługi, dlatego można ją stosować do wszystkich usług. Adres w tej trasie to `LOCAL`, co oznacza, że komunikaty są kierowane do lokalnego serwera bazodanowego.

Trasy można tworzyć w bazie, w której działa Service Broker, lub w bazie MSDB. Trasy z bazy MSDB służą do przekierowywania komunikatów. Przekierowywane komunikaty trafiają do danego egzemplarza systemu SQL Server i wychodzą z niego bez zapisywania ich w żadnej bazie.

Trasy mają różne parametry, które można skonfigurować za pomocą instrukcji `CREATE ROUTE`. Oto te parametry.

- **Nazwa trasy.** Jest to nazwa trasy zgodna ze standardowymi regułami nazywania obiektów.
- **Nazwa usługi.** Jest to nazwa usługi, dla której stosowana jest dana trasa. W instrukcji `CREATE ROUTE` można albo podać nazwę konkretnej usługi, albo ją pominąć. W tym drugim przypadku trasa obowiązuje dla wszystkich usług. Przy podawaniu nazwy usługi w instrukcji `CREATE ROUTE` wielkość znaków ma znaczenie i ignorowana jest kolacja bazy danych. Dzieje się tak, ponieważ system SQL Server przeprowadza porównanie binarne nazwy usługi z trasy z nazwą usługi zapisaną w bazie danych. Wielkie i małe litery mają różne wartości binarne, jeśli zatem któryś ze znaków nie będzie pasował, trasa nie zostanie zastosowana.
- **BROKER_INSTANCE.** Jest to opcjonalny parametr określający trasę, którą baza danych z serwera ma przysyłać komunikaty. Wartość `BROKER_INSTANCE` można sprawdzić w kolumnie `service_broker_guid` widoku `sys.database` w egzemplarzu systemu zawierającym bazę, do której kieruje trasa. Jeśli parametr `BROKER_INSTANCE` nie jest ustawiony, Service Broker próbuje zidentyfikować docelową bazę w egzemplarzu przez dopasowanie nazwy docelowej usługi do nazwy usługi w bazie danych ze zdalnego egzemplarza. Jeśli określone są przynajmniej dwie trasy z tą samą nazwą usługi, ale bez baz danych, Service Broker zakłada, że usługi te są używane w ramach skalowania, dlatego losowo przekazuje komunikaty do dostępnych tras.
- **LIFETIME.** Ten opcjonalny parametr informuje Service Broker o tym, ile sekund trasa ma być aktywna. Gdy czas życia trasy wygaśnie, będzie ona ignorowana. Jeżeli parametr `LIFETIME` jest pominięty lub gdy ma wartość `NULL`, trasa nigdy nie wygasa. Istnieje niewiele zastosowań dla tras tymczasowych, dlatego parametr `LIFETIME` jest rzadko stosowany.
- **ADDRESS.** Ten wymagany parametr informuje Service Broker o tym, jak należy kontaktować się ze zdalną bazą danych. W tym parametrze można podać adres IP, adres sieciowy lub kompletną nazwę domeny połączoną z numerem portu TCP w punkcie końcowym Service Brokera (ten port trzeba utworzyć na zdalnym egzemplarzu). Ta ostatnia nazwa ma format `TCP://NazwaSerwera:NumerPortu`.

Jeśli docelowa baza danych znajduje się w tym samym egzemplarzu systemu, co baza źródłowa, do parametru `ADDRESS` można przypisać wartość `LOCAL`. Gdy wartość tego parametru to `TRANSPORT`, Service Broker próbuje ustalić, z którym zdalnym egzemplarzem ma się połączyć. Robi to na podstawie nazwy usługi. Wtedy nazwa usługi pełni funkcję adresu URL. Trasy z ustawieniem `TRANSPORT` zwykle służą do zwracania adresów, gdy setki klientów mogą przysyłać komunikaty do tej samej usługi. W docelowej usłudze nie ma wtedy sensu przechowywanie tras do setek nadawców komunikatów.

- **MIRROR_ADDRESS.** Ten opcjonalny parametr określa trasę obsługującą kopie lustrzane bazy danych (jeśli dla docelowej bazy jest skonfigurowany ten mechanizm). Jeżeli docelowa baza danych ma włączoną obsługę kopii lustrzanych, a parametr `MIRROR_ADDRESS` nie jest ustawiony, to gdy w bazie wystąpią problemy i potrzebne będzie użycie kopii lustrzanej, komunikaty nie zostaną dostarczone do momentu przywrócenia pracy egzemplarza podanego w parametrze `ADDRESS`. Wartość parametru `MIRROR_ADDRESS`

powinna mieć ten sam format, co wartość parametru ADDRESS. Gdy parametr MIRROR_ADDRESS jest ustawiony, Service Broker automatycznie wysyła komunikaty do głównego w danym momencie systemu SQL Server z lustrzanej pary.

W poniższym fragmencie kodu pokazujemy, jak zastosować różne parametry instrukcji CREATE ROUTE:

```
CREATE ROUTE ExpenseRoute
WITH SERVICE_NAME = 'MyService',
BROKER_INSTANCE = '53FA2363-BF93-4EB6-A32D-F672339E08ED',
ADDRESS = 'TCP://sql12:1234',
MIRROR_ADDRESS = 'TCP://sql14:4567' ;
```

UWAGA Gdy w Service Brokerze używane są nazwy zdalnych usług, zawsze mają one postać łańcuchów znaków, w których wielkość liter ma znaczenie. Jest tak, ponieważ zgodnie z architekturą Service Brokera zdalne usługi nie muszą działać w systemie Windows ani SQL Server oraz mogą być niezgodne z regułami nazewnictwa obowiązującymi w systemie SQL Server. Choć obecnie wszystkie usługi Service Brokera działają w systemie Windows i SQL Server, struktura dostępnych poleceń sprawia, że w przyszłości obsługiwane mogą być także usługi innych rodzajów.

Priorytety

W wielu aplikacjach do przesyłania komunikatów niektóre komunikaty są ważniejsze od pozostałych. Priorytety pozwalają zagwarantować, że istotne komunikaty będą przetwarzane przed mniej ważnymi. Komunikaty w konwersacjach Service Brokera zawsze są przetwarzane w kolejności ich dodania. Dlatego priorytety komunikatów w konwersacjach nie mają znaczenia. Priorytety w Service Brokerze są przypisywane do samych konwersacji.

W Service Brokerze konwersacje o wyższym priorytecie są przetwarzane przed konwersacjami o niższym priorytecie. Może to mieć znaczenie w środowiskach z wysokim obciążeniem, w których niektóre komunikaty należy przetwarzać przed innymi. Aby ustawić priorytet konwersacji, trzeba wybrać nazwę kontraktu, nazwę usługi źródłowej i nazwę usługi docelowej powiązane z odpowiednim priorytetem Service Brokera.

Ponieważ użytkownik nie ustawia priorytetu konwersacji w momencie jej tworzenia, warto przygotować kilka kontraktów z tymi samymi typami komunikatów i skonfigurować je do użytku w określonych usługach. Aby utworzyć konwersacje o wysokim i niskim priorytecie, możesz zbudować kontrakty o nazwach ContractLow i ContractHigh. Następnie należy utworzyć wysoki priorytet powiązany z kontraktem ContractHigh.

Priorytety Service Brokera tworzy się za pomocą instrukcji CREATE BROKER PRIORITY języka T-SQL. Ta instrukcja przyjmuje pięć różnych wartości.

- Pierwszą wartością jest nazwa priorytetu.
- Trzy następne wartości umożliwiają określenie, dla których konwersacji ustawiany jest dany priorytet. Aby wskazać konwersację, możesz podać jej nazwę, nazwę lokalnej usługi i nazwę zdalnej usługi. Każdy z tych trzech parametrów przyjmuje również wartość ANY, oznaczającą, że dany parametr nie jest uwzględniany przy przypisywaniu priorytetu do konwersacji.
- Ostatni parametr to poziom priorytetu używany dla docelowych konwersacji. Dopuszczalne są wartości od 1 do 10, gdzie 10 oznacza najwyższy priorytet. Dla konwersacji bez ustawionego priorytetu używana jest wartość 5.

W poniższym fragmencie kodu pokazano, jak stosować instrukcję `CREATE BROKER PRIORITY`. Tu dla komunikatu przesyłanego do dowolnej usługi z kontraktem `MyHighPriority` używany jest priorytet 8 (zamiast domyślnego priorytetu 5):

```
CREATE BROKER PRIORITY HighPriority
FOR CONVERSATION
SET ( CONTRACT_NAME = MyHighPriority ,
      LOCAL_SERVICE_NAME = ANY ,
      REMOTE_SERVICE_NAME = N'ANY' ,
      PRIORITY_LEVEL = 8
)
```

Komunikaty zawsze są odbierane w kolejności zgodnej z priorytetami. Przy wysyłaniu priorytetu są uwzględniane wtedy, gdy opcja `Honor_Broker_Priority` ma w danej bazie wartość `TRUE`.

Grupy konwersacji

Jednym z problemów w trakcie tworzenia skalowalnych aplikacji wykorzystujących kolejki jest zarządzanie wieloma wątkami, które równolegle odczytują dane z kolejek. Jeśli komunikaty nie są powiązane, rozwiązanie działa prawidłowo. Jeżeli jednak komunikaty dotyczą tej samej operacji, może się zdarzyć, że komunikaty przetwarzane w różnych wątkach spowodują konflikty.

Załóżmy, że usługa przetwarzająca zamówienie wysyła komunikaty do usługi obsługującej stan magazynu i usługi odpowiadającej za dostawy. Po zakończeniu pracy usługi zwracają wyniki. Gdy usługa przetwarzająca zamówienia otrzyma wyniki, zaktualizuje zamówienie. Jeśli dwa wyniki zostaną zwrócone w tym samym czasie i przetworzone w dwóch odrębnych wątkach, w trakcie aktualizacji mogą wystąpić konflikty prowadzące do błędów w zamówieniu.

Komunikaty w ramach konwersacji nie stanowią problemu. W konwersacjach nie można pobrać nowego komunikatu przed przetworzeniem poprzedniego. Nie dotyczy to jednak różnych konwersacji powiązanych z jedną operacją. Aby rozwiązać problem, można połączyć powiązane konwersacje w *grupę konwersacji*. Gdy komunikat w grupie konwersacji jest przetwarzany, tworzona jest blokada, która uniemożliwia innym wątkom przetwarzanie kolejnych komunikatów z dowolnej konwersacji z grupy. To oznacza, że choć przetwarzanie kolejek może odbywać się równolegle, przetwarzanie komunikatów w grupie konwersacji przebiega jednowątkowo. Pozwala to znacznie uprościć kod potrzebny do przetwarzania komunikatów.

Domyślnie każda konwersacja jest umieszczana we własnej grupie, chyba że w momencie tworzenia konwersacji podana została inna grupa. Grupę konwersacji, w której ma znaleźć się nowa konwersacja, można podać na dwa sposoby:

- przez określenie docelowej grupy konwersacji,
- przez podanie uchwytu konwersacji, z którą należy zgrupować nową konwersację.

Nie istnieje odrębne polecenie tworzące nową grupę konwersacji. Gdy powstaje nowa konwersacja, automatycznie dodawana jest nowa grupa, dla której generowany jest nowy identyfikator GUID. Aby przypisać nową konwersację do grupy obejmującej konkretną konwersację, wystarczy podać w parametrze `RELATED_CONVERSATION_GROUP` instrukcji `BEGIN DIALOG CONVERSATION` (jej opis znajdziesz dalej w tym rozdziale) wartość identyfikatora GUID.

Aby sprawdzić grupę konwersacji, do której należy następny przetwarzany komunikat, trzeba wywołać instrukcję `GET CONVERSATION GROUP`. W tej instrukcji podajemy zmienną, w której zapisana zostanie następna używana grupa konwersacji, a także nazwę kolejki stosowanej w tej grupie. Ilustruje to poniższy kod:

```
DECLARE @conversation_group_id AS UNIQUEIDENTIFIER;
GET CONVERSATION GROUP @conversation_group_id
FROM YourQueue;
```

Używanie Service Brokera

Wysyłanie i pobieranie komunikatów za pomocą Service Brokera to proste zadanie. Zamiast wstawiać komunikaty do kolejki za pomocą instrukcji `INSERT` (jak robi się to w tabelach) i pobierać je przy użyciu instrukcji `SELECT`, należy stosować polecenia `SEND` (do wysyłania wiadomości) i `RECEIVE` (do pobierania danych z kolejki).

Wysyłanie komunikatów

Do wysyłania komunikatów służy polecenie `SEND`. Przyjmuje ono tylko dwa parametry; są to identyfikator konwersacji i treść komunikatu, co przedstawiono w poniższym fragmencie kodu. Identyfikator konwersacji można pobrać za pomocą instrukcji `BEGIN DIALOG CONVERSATION`, co pokazano w przykładowym kodzie ilustrującym przypisywanie priorytetów.

```
DECLARE @message_body AS XML, @dialog_handle as UNIQUEIDENTIFIER
SET @message_body = (SELECT *
    FROM sys.all_objects as object
    FOR XML AUTO, root('root'))
BEGIN DIALOG CONVERSATION @dialog_handle
    FROM SERVICE [YourSourceService]
    TO SERVICE 'YourDestinationService'
    ON CONTRACT [YourContract];
SEND ON CONVERSATION @dialog_handle
MESSAGE TYPE YourMessageType
(@message_body)
GO
```

W wersji SQL Server 2012 polecenie `SEND` przyjmowało rozdzieloną przecinkami listę identyfikatorów konwersacji. Gdy podany jest więcej niż jeden identyfikator, kopia komunikatu jest przesyłana do każdej zdalnej usługi określonej na liście konwersacji. Dla takiego rozwiązania łatwiej napisać kod niż dla wielu instrukcji `SEND` dotyczących pojedynczych konwersacji. Ponadto Service Broker zarządza wtedy konwersacjami, dlatego po stronie nadawczej przechowywana jest tylko jedna kopia komunikatu. To bardzo wydajne rozwiązanie. Ten mechanizm nosi nazwę *rozsyłania* (ang. *multicasting*).

Polecenie `BEGIN DIALOG CONVERSATION` przyjmuje kilka parametrów. Niektóre z nich są wymagane, a inne — opcjonalne.

Pierwsze trzy parametry są wymagane.

- Dwa parametry określają usługi źródłowe i docelowe używane przy wysyłaniu komunikatu. Opcjonalnie po nazwie usługi docelowej można podać identyfikator docelowej bazy danych lub wartość `CURRENT DATABASE`. To ustawienie jest niezbędne, jeśli istnieje więcej niż jedna usługa o danej nazwie.
- Trzeci parametr to kontrakt definiujący typy komunikatów używane przy wysyłaniu danych.

Pozostałe parametry są opcjonalne.

- **RELATED_CONVERSATION.** Parametr określa grupę konwersacji, co pozwala powiązać nową konwersację z inną. Można tu podać identyfikator innej, istniejącej już konwersacji.
- **RELATED_CONVERSATION_GROUP.** Także ten parametr określa grupę konwersacji i wiąże nową konwersację z inną. Można tu podać identyfikator określonej grupy konwersacji, do której nowa konwersacja ma należeć.
- **LIFETIME.** Określa ilość czasu, przez jaki konwersacja ma pozostać otwarta. Należy podać liczbę sekund do momentu automatycznego zamknięcia konwersacji. Wartość tę należy podać jako liczbę całkowitą. Domyślnie używana jest maksymalna wartość typu danych INT, czyli $2^{31}-1$ (2 147 483 647), co odpowiada 68 latom.
- **ENCRYPTION.** Określa, czy komunikaty w konwersacji mają być szyfrowane, gdy są przesyłane do innego egzemplarza systemu SQL Server. Parametr przyjmuje tylko wartości ON (jest to wartość domyślna) i OFF. Przy przesyłaniu komunikatów między egzemplarzami systemu zalecaną wartością jest ON. Gdy parametr ENCRYPTION ma wartość ON i komunikaty są przesyłane w ramach jednego egzemplarza, dane nie są szyfrowane, ale klucz główny bazy danych i certyfikaty używane przy szyfrowaniu są niezbędne do rozpoczęcia konwersacji i przesyłania wiadomości.

Przedstawiony wcześniej kod to dobry punkt wyjścia, jednak nie jest on odpowiedni przy wysokim obciążeniu Service Brokera. Powodem jest to, że tworzenie odrębnej konwersacji dla każdego komunikatu jest kosztowne. W systemach pracujących pod wysokim obciążeniem, gdy przesyłane są setki tysięcy lub miliony komunikatów dziennie, warto powtórnie wykorzystywać konwersacje i wysłać po kilka komunikatów na konwersację, aby ograniczyć koszty. Efekt ten można łatwo uzyskać dzięki zapisaniu uchwytu konwersacji (jest to wartość `@dialog_handle` ustawiana w instrukcji `BEGIN DIALOG CONVERSATION`) w tabeli. Uchwyt ten można potem pobrać na potrzeby przyszłych sesji. Potrzebną tabelę przedstawiono w poniższym fragmencie kodu:

```
CREATE TABLE dbo.SSB_Settings
([Source] sysname NOT NULL,
[Destination] sysname NOT NULL,
[Contract] sysname NOT NULL,
[dialog_handle] uniqueidentifier
CONSTRAINT PK_SSB_Setting PRIMARY KEY ([Source], [Destination], [Contract]))
```

Przy przesyłaniu wielu komunikatów w jednej konwersacji nadawca potrzebuje sposobu na poinformowanie odbiorcy o tym, że nie będzie już przekazywał danych w tej konwersacji. Łatwym rozwiązaniem jest użycie w bazie dodatkowego typu komunikatów, informującego odbiorcę o zakończeniu konwersacji.

W środowiskach z wysokim obciążeniem można zastosować procedurę składowaną, która określi, czy należy utworzyć nową konwersację, i — jeśli będzie trzeba — zapisze wartość tej konwersacji. Na listingu 6.1 (plik *KodR06.sql*) wartość `send_sequence` z widoku `DMV sys.conversation_endpoints` informuje, czy należy zakończyć konwersację. Sygnałem dla odbiorcy określającym, że należy zamknąć konwersację, jest komunikat typu `EndOfConversation`.

Listing 6.1. Tworzenie konwersacji wielokrotnego użytku

```
CREATE PROCEDURE dbo.CreateConversation
    @Destination sysname,
    @Source sysname,
    @Contract sysname,
    @MessageType sysname,
```

```

        @MessageBody XML,
        @dialog_handle uniqueidentifier
AS
/* Pobieranie identyfikatora konwersacji. */
SELECT @dialog_handle = dialog_handle
FROM dbo.SSB_Settings
WHERE [Source]= @Source
      AND [Destination]= @Destination
      AND [Contract]= @Contract;
/* Jeśli uchwyt nie istnieje, należy utworzyć nową konwersację. */
IF @dialog_handle IS NULL
BEGIN
    BEGIN TRANSACTION
    /* Jeśli uchwyt do konwersacji istnieje, należy przesłać do
       odbiorcy sygnał zakończenia poprzedniej konwersacji. */
    IF @dialog_handle IS NOT NULL
    BEGIN
        UPDATE dbo.SSB_Settings
        SET dialog_handle = NULL
        WHERE [Source]= @Source
              AND [Destination] = @Destination
              AND [Contract]= @Contract;
        SEND ON CONVERSATION @dialog_handle
        MESSAGE TYPE EndOfConversation;
    END
    /* Konfigurowanie nowej konwersacji. */
    BEGIN DIALOG CONVERSATION @dialog_handle
    FROM SERVICE @Source
    TO SERVICE @Destination
    ON CONTRACT @Contract;
    /* Zapisywanie identyfikatora nowej konwersacji. */
    UPDATE dbo.SSB_Settings
    SET dialog_handle = @dialog_handle
    WHERE [Source]= @Source
          AND [Destination] = @Destination
          AND [Contract]= @Contract;
    IF @@ROWCOUNT = 0
        INSERT INTO dbo.SSB_Settings
        ([Source], [Destination], [Contract], [dialog_handle])
        VALUES
        (@Source, @Destination, @Contract, @dialog_handle);
END;
/* Wysyłanie komunikatu. */
SEND ON CONVERSATION @dialog_handle
MESSAGE TYPE @MessageType
(@XML);
/* Sprawdzanie, czy uchwyt konwersacji wciąż jest taki sam jak uchwyt zapisany w tabeli.
   Jeśli uchwyty są różne, należy oznaczyć daną konwersację jako zakończoną. */
IF (SELECT dialog_handle
    FROM dbo.SSB_Settings
    WHERE [Source]= @Source
          AND [Destination] = @Destination
          AND [Contract] = @Contract) <> @dialog_handle
    SEND ON CONVERSATION @dialog_handle
    MESSAGE TYPE EndOfConversation;
GO

```

Odbieranie komunikatów

Do odbierania komunikatów służy polecenie RECEIVE. Działa ona bardzo podobnie jak polecenie SELECT. Możesz określić zwracane kolumny, a kolejka jest podawana w klauzuli FROM, co pokazano w poniższym fragmencie kodu. Po zapisaniu danych w zmiennej można wykonać na nich dowolne operacje.

```
DECLARE @dialog_handle UNIQUEIDENTIFIER, @messagetype nvarchar(128),
        @message_body XML;

BEGIN TRANSACTION;
RECEIVE TOP (1) @dialog_handle = conversation_handle,
        @messagetype = [message_type_name],
        @message_body = CAST(message_body as XML)
FROM MyDestinationQueue
IF @@ROWCOUNT = 0
    BEGIN
        COMMIT TRANSACTION;
    END
ELSE IF @messagetype = N'http://schemas.microsoft.com/SQL/ServiceBroker/Error'
    BEGIN
        -- Rejestrowanie błędu w dzienniku aplikacji.
        END CONVERSATION @dialog_handle;
        COMMIT TRANSACTION;
    END
ELSE IF @messagetype = N'http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog'
    BEGIN
        END CONVERSATION @dialog_handle;
        COMMIT TRANSACTION;
    END
ELSE
    BEGIN
        /* Wykonanie potrzebnych operacji na dokumencie XML. */
        COMMIT TRANSACTION;
    END
END
```

Przedstawiony w tym przykładzie podstawowy kod nie jest najwydajniejszym sposobem pobierania danych. Lepiej odbierać wiele komunikatów jednocześnie i wczytywać treść komunikatu z kolejki jako dane binarne, a dopiero po usunięciu ich z kolejki przekształcać je na format XML (lub na inny typ, w jakim zostały przesłane).

W poniższym kodzie pokazano, jak za pomocą jednej instrukcji pobrać wiele komunikatów:

```
DECLARE @dialog_handle UNIQUEIDENTIFIER, @message_body XML,
        @message_type nvarchar(128)
DECLARE @Messages TABLE
(conversation_handle uniqueidentifier,
message_type sysname,
message_body VARBINARY(MAX))
BEGIN TRANSACTION
WAITFOR (
    RECEIVE TOP (1000) conversation_handle, message_type_name, message_body
    FROM MyDestinationQueue
    INTO @Messages), TIMEOUT 5000;
DECLARE cur CURSOR FOR select conversation_handle, message_type,
        CAST(message_body AS XML)
    FROM @Messages
    WHERE message_body IS NOT NULL
```

```

OPEN cur
FETCH NEXT FROM cur INTO @dialog_handle, @message_type, @message_body
WHILE @@FETCH_STATUS= 0
BEGIN
IF @message_type = N'http://schemas.microsoft.com/SQL/ServiceBroker/Error'
BEGIN
-- Rejestrowanie błędu w dzienniku aplikacji.
END CONVERSATION @dialog_handle;
END
ELSE IF @message_type = N'http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog'
BEGIN
END CONVERSATION @dialog_handle;
END
ELSE
BEGIN
/* Wykonywanie potrzebnych operacji na dokumencie XML. */
SELECT @message_body
END
FETCH NEXT FROM cur INTO @dialog_handle, @message_body
END
CLOSE cur
DEALLOCATE cur;
IF EXISTS (SELECT * FROM @Messages WHERE message_type = 'EndOfConversation')
END CONVERSATION @dialog_handle
COMMIT TRANSACTION
GO

```

Przesyłanie komunikatów między egzemplarzami

Jedną z najbardziej wartościowych funkcji Service Brokera jest możliwość przesyłania komunikatów między bazami danych z różnych egzemplarzy systemu SQL Server, działających na innych serwerach fizycznych (lub wirtualnych). Konfigurowanie Service Brokera na potrzeby przekazywania komunikatów między egzemplarzami przebiega bardzo podobnie jak ustawianie tego narzędzia w celu wymiany komunikatów między bazami z tego samego egzemplarza systemu SQL Server. Główna różnica związana jest z krokami, jakie trzeba wykonać, aby skonfigurować uwierzytelnianie komunikacji między egzemplarzami.

Kroki te są opisane poniżej. Powinieneś wykonać je na obu egzemplarzach wymieniających komunikaty przy użyciu Service Brokera.

1. Skonfiguruj klucz główny bazy danych w nadrzędnej bazie. Klucz posłuży do szyfrowania kluczy głównych.
2. Skonfiguruj klucz główny bazy danych w bazie aplikacji. Posłuży on do zabezpieczania certyfikatów przechowywanych w tej bazie.
3. Utwórz certyfikat w każdej bazie.
4. Prześlij certyfikaty między bazami.
5. Utwórz punkty końcowe Service Brokera w każdym egzemplarzu systemu SQL Server.
6. Skonfiguruj trasy łączące punkty końcowe Service Brokera ze zdalnych egzemplarzy systemu SQL Server.

Po wykonaniu tych kroków można przekazywać komunikaty między dwoma skonfigurowanymi bazami danych.

Klucz główny bazy danych

Wcześniej napisano, że przed rozpoczęciem używania Service Brokera dla dwóch egzemplarzy systemu trzeba przygotować klucz główny dla obu baz danych. W tym celu należy wywołać polecenie `CREATE MASTER KEY` w obu tych bazach. Jeśli ta instrukcja została już wykonana wcześniej, nie trzeba jej ponawiać. Tworzenie klucza głównego jest łatwe, co pokazano w poniższym fragmencie kodu. Potrzebna instrukcja przyjmuje tylko jeden parametr. Jest nim hasło używane do zabezpieczania klucza głównego bazy danych.

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'YourSecurePassword1!'
```

Klucz główny jest symetryczny i chroni wszystkie pozostałe klucze w bazie danych, w tym inne klucze symetryczne, certyfikaty i klucze asymetryczne.

Po wygenerowaniu klucza głównego bazy danych utwórz jego kopię zapasową za pomocą polecenia `BACKUP MASTER KEY`. Pozwoli to odzyskać klucz główny po awarii bazy danych. Zapisz kopię zapasową klucza głównego w bezpiecznej lokalizacji poza serwerem.

Tworzenie certyfikatów

Gdy dla punktów końcowych stosujesz uwierzytelnianie oparte na certyfikatach, musisz utworzyć certyfikaty w nadrzędnych bazach danych egzemplarzy systemu, między którymi wymieniane są dane. W celu przygotowania certyfikatu wywołaj instrukcję `CREATE CERTIFICATE` w nadrzędnej bazie danych, co ilustruje poniższy fragment kodu. Przy tworzeniu certyfikatów w poszczególnych egzemplarzach systemu przypisz każdemu z nich unikatową nazwę. Najłatwiej w tym celu dodać nazwę serwera i nazwę egzemplarza do nazwy certyfikatu.

```
CREATE CERTIFICATE MyServiceBrokerCertificate
WITH SUBJECT = 'Service Broker Certificate',
     START_DATE = '1/1/2013',
     EXPIRY_DATE = '12/31/2099'
```

Certyfikaty tworzone w ten sposób to *certyfikaty z podpisem własnym*, ponieważ są podpisane za pomocą wewnętrznych kluczy. Certyfikaty podpisane przez jednostkę certyfikacyjną są uznawane za bezpieczniejsze, jednak certyfikaty z podpisem własnym są zwykle wystarczające, aby zapewnić bezpieczeństwo punktów końcowych Service Brokera.

Wymiana certyfikatów

Certyfikaty po utworzeniu trzeba wymienić między bazami. W tym celu należy przygotować kopie zapasowe certyfikatów. Służy do tego przedstawiona w poniższym fragmencie kodu instrukcja `BACKUP CERTIFICATE`. Należy ją wywołać na komputerze, na którym znajduje się dany certyfikat. Następnie trzeba przywrócić certyfikat na zdalnym egzemplarzu, używając polecenia `CREATE CERTIFICATE` pokazanego w drugim fragmencie kodu.

```
BACKUP CERTIFICATE MyServiceBrokerCertificate
TO FILE='C:\MyServiceBrokerCertificate.cer'
```

```
CREATE CERTIFICATE MyServiceBrokerCertificate
FROM FILE='c:\MyServiceBrokerCertificate.cer'
```

Punkty końcowe Service Brokera

Punkty końcowe umożliwiają użytkownikom i innym egzemplarzom systemu SQL Server łączenie się z egzemplarzem, na którym utworzono dany punkt. Aby egzemplarze systemu SQL

Server mogły przysyłać komunikaty do innego egzemplarza, po obu stronach trzeba utworzyć punkty końcowe.

Do dodawania punktów końcowych służy instrukcja `CREATE ENDPOINT` przedstawiona w poniższym kodzie. Na każdym egzemplarzu systemu SQL Server może znajdować się tylko jeden punkt końcowy Service Brokera. Jeśli nawet wiele egzemplarzy przesyła komunikaty na jeden serwer, cała komunikacja na tym serwerze odbywa się za pomocą jednego punktu końcowego. Punkty końcowe Service Brokera udostępniają różne techniki uwierzytelniania, w tym uwierzytelnianie w trybie NTLM, KERBEROS i CERTIFICATE, a także różne kombinacje tych trzech technik. Przy uwierzytelnianiu różnych egzemplarzy na potrzeby Service Brokera zaleca się stosowanie trybu CERTIFICATE, ponieważ pozwala to uniknąć zależności od katalogu Active Directory.

```
USE master
GO
CREATE ENDPOINT ServiceBrokerEndpoint
STATE = STARTED
AS TCP (LISTENER_PORT = 1234, LISTENER_IP=ALL)
FOR SERVICE_BROKER
(AUTHENTICATION = CERTIFICATE MyServiceBrokerCertificate,
 ENCRYPTION = REQUIRED ALGORITHM AES);
GO
```

W punktach końcowych można też skonfigurować szyfrowanie. Dostępne są opcje `DISABLED`, `SUPPORTED` i `REQUIRED`. Obsługiwane jest szyfrowanie za pomocą algorytmów RC4 i AES (ang. *Advanced Encryption Standard*), a także kombinacje obu algorytmów (podawane jako AES RC4 i RC4 AES). Algorytm RC4 nie jest już uznawany za bezpieczny, dlatego zawsze należy stosować algorytm AES.

Przy konfigurowaniu punktu końcowego Service Brokera trzeba ustawić określony port TCP (oprócz domyślnego portu TCP systemu SQL Server, w którym dany egzemplarz oczekuje na dane). Należy też podać adres IP, pod którym punkt końcowy ma oczekiwać na dane. W przedstawionym kodzie do oczekiwania na dane używany jest port TCP 1234, a punkt końcowy może odbierać dane kierowane pod wszystkie adresy IP skonfigurowane na serwerze, na którym działa dany egzemplarz. Jeśli punkt końcowy ma odbierać dane tylko z określonego adresu IP, ustaw adres IPv4 lub IPv6 z ustawieniem `LISTENER_IP`.

Zewnętrzna aktywacja

Zewnętrzna aktywacja różni się od normalnej aktywacji procedur składowanych, związanej z instrukcjami `CREATE QUEUE` i `ALTER QUEUE`. Przy zewnętrznej aktywacji uruchamiana jest odrębna usługa systemu Windows (*ssbea.exe*), która monitoruje kolejkę systemu SQL Server w oczekiwaniu na pojawienie się nowych komunikatów.

Gdy do kolejki trafia nowy komunikat, usługa odpowiedzialna za zewnętrzną aktywację uruchamia aplikację systemu Windows określoną w konfiguracji tej usługi. Usługa ta monitoruje kolejkę Service Brokera. Użytkownik musi skonfigurować dla tej kolejki mechanizm powiadamiania o zdarzeniach, który przesyła komunikat do drugiej kolejki (kolejki powiadomień). Gdy komunikat trafia do kolejki aplikacji, mechanizm powiadamiania o zdarzeniach przesyła komunikat do kolejki powiadomień, co powoduje, że usługa odpowiedzialna za zewnętrzną aktywację uruchamia odpowiednią aplikację.

PO CO STOSOWAĆ ZEWNĘTRZNĄ AKTYWACJĘ?

Istnieje kilka sytuacji, w których użytkownicy korzystają z zewnętrznej aktywacji. Najczęściej stosuje się ją wtedy, gdy coś dzieje się poza systemem SQL Server w reakcji na wystąpienie określonego zdarzenia w tym systemie.

Wyobraź sobie, że na serwerze plików znajdują się pliki, a każdy z nich odpowiada wierszowi z tabeli z systemu SQL Server. Należy zadbać o to, aby po usunięciu każdego wiersza kasowany był też powiązany z nim plik. Proces zewnętrznej aktywacji to dobry sposób na uzyskanie takiego efektu. Wystarczy utworzyć dla operacji usuwania danych z tabeli wyzwalacz, który będzie wysyłał do kolejki komunikat z informacją o przeznaczonym do usunięcia pliku. Następnie zewnętrzny aktywator wykryje nadejście komunikatu i uruchomi aplikację usuwającą pliki z serwera plików. Potem aplikacja wczyta dane z kolejki, do której wyzwalacz przesłał komunikat, i na podstawie tych danych usunie odpowiedni plik.

Można zastosować inną technikę, aby uzyskać ten sam efekt bez zewnętrznej aktywacji. W tym celu należy uruchamiać usuwającą pliki aplikację jako usługę, która np. co minutę sprawdza, czy w kolejce są komunikaty do przetworzenia. Jest to jednak mniej wydajne niż zastosowanie zewnętrznej aktywacji.

Możesz wykorzystać poniższy przykładowy fragment kodu do utworzenia powiadomień o zdarzeniach. Usługa zewnętrznej aktywacji jest dostępna do pobrania na stronie <http://filedir.com/windows/servers/microsoft-sqlserver-2012-service-broker-external-activator-2586093.html>.

```
CREATE QUEUE dbo.MyDestinationQueueEA
GO
CREATE SERVICE MyDestinationServiceEA
ON QUEUE dbo.MyDestinationQueueEA
(
    [http://schemas.microsoft.com/SQL/Notifications/PostEventNotification]
)
GO
CREATE EVENT NOTIFICATION MyDestinationNotificationEA
ON QUEUE MyDestinationQueue
FOR QUEUE_ACTIVATION
TO SERVICE 'MyDestinationServiceEA', 'current database'
GO
```

W kodzie z tego fragmentu przyjęto, że powiadomienia o zdarzeniach i wszystkie obiekty Service Brokera są tworzone w tej samej bazie. Dlatego zastosowano ustawienie 'current database'.

Przykład ilustrujący zapisywanie danych o użytkownikach w dzienniku

Aby połączyć wszystkie zaprezentowane tu informacje, w końcowej części rozdziału przedstawiamy kompletną przykładową aplikację opartą na Service Brokerze. W kodzie wykorzystano jedną z pierwszych aplikacji produkcyjnych napisanych z zastosowaniem tej technologii.

Duża firma programistyczna miała wiele zdalnych serwerów, których pracownicy używali do łączenia się z siecią spoza domeny. Zgodnie z regułami bezpieczeństwa przed przyznaniem dostępu do sieci każdy login musiał być zapisany w centralnej bazie danych. To oznaczało, że nie można było uzyskać zdalnego dostępu do sieci, gdy serwer bazodanowy nie działał. W firmie nie używano konwersacji Service Brokera do przesyłania rekordów z loginami

na serwer bazodanowy. Jednak gdy serwer rejestrujący dane był niedostępny, komunikaty można było zapisywać w kolejce w celu ich późniejszego przetworzenia, dzięki czemu użytkownicy mogli się logować do sieci.

Poniższy skrypt tworzy obiekty Service Brokera i procedurę składowaną potrzebną do odbierania komunikatów oraz wstawiania ich do tabeli z rejestrowanymi danymi (ten kod znajdziesz w pliku *RejestrUzytKonfigPobierania.sql*).

```
CREATE DATABASE LogReceiveExample
GO
USE LogReceiveExample
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'YourSecurePassword!'
GO
ALTER DATABASE LogReceiveExample SET TRUSTWORTHY ON
GO
-- Możliwe, że nie będziesz tego potrzebował. Ten fragment zastosowano, aby można było
-- uruchomić przykład na laptopie niepodłączonym do domeny. Inna możliwość to utworzenie
-- baz danych za pomocą uwierzytelnionego konta systemu SQL Server, a nie
-- przy użyciu uwierzytelnionego konta systemu Windows.
-- Jeśli uruchamiasz kod na serwerze podłączonym do domeny, ta instrukcja
-- nie jest potrzebna.
ALTER AUTHORIZATION ON DATABASE::LogReceiveExample TO SA
GO

-- Tworzenie obiektów Service Brokera definiujących konwersację.
CREATE MESSAGE TYPE LoginMessage
AUTHORIZATION dbo
VALIDATION = NONE
GO

CREATE CONTRACT LoginContract
AUTHORIZATION dbo
(LoginMessage SENT BY ANY)
GO

-- To tabela, która przechowuje rekordy wysłane przez klienta.
CREATE TABLE dbo.LoginRecords
(LoginEntry varchar(MAX))
GO

CREATE PROCEDURE dbo.ReceiveQueueHandler
AS
RETURN;
GO

ALTER PROCEDURE dbo.ReceiveQueueHandler
AS
BEGIN -- ReceiveQueueHandler
    DECLARE @dialog_handle UNIQUEIDENTIFIER, @messagetype nvarchar(128);
    -- Tworzenie zmiennej tabelowej przeznaczonej na otrzymane komunikaty.
    DECLARE @LoginMessage TABLE (
        [conversation_handle] uniqueidentifier,
        message_type_name nvarchar(128),
        [message_body] varchar(MAX)
    )
    )
```

```

-- Główna pętla do pobierania danych.
WHILE (1 = 1)
    BEGIN
        BEGIN TRY
            BEGIN TRANSACTION;

            WAITFOR (
                RECEIVE TOP (1000) -- Pobieranie wielu komunikatów jednocześnie jest
                    -- wydajniejsze niż wczytywanie ich pojedynczo.
                    [conversation_handle],
                    [message_type_name],
                    CAST(message_body AS varchar(MAX)) AS LoginEntry
                FROM LoginReceiveQueue
                INTO @LoginMessage
            ), TIMEOUT 100;
            -- Jeśli nie otrzymano żadnych komunikatów, można zakończyć pracę.
            IF @@ROWCOUNT = 0
                BEGIN
                    COMMIT TRANSACTION;
                    BREAK;
                END
            ELSE
                BEGIN
                    -- Sprawdzanie, czy otrzymano komunikaty systemowe.
                    SELECT TOP (1) @dialog_handle = [conversation_handle],
                        @messagetype = [message_type_name]
                    FROM @LoginMessage
                    WHERE [message_type_name] IN
(N'http://schemas.microsoft.com/SQL/ServiceBroker/Error',
 N'http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog')
                    -- Obsługa komunikatów systemowych.
                    IF @@ROWCOUNT > 0
                        BEGIN
                            IF @messagetype = N'http://schemas.microsoft.com/SQL/ServiceBroker/Error'
                                BEGIN
                                    -- Rejestrowanie błędów w dzienniku aplikacji.
                                    END CONVERSATION @dialog_handle;
                                END
                            ELSE IF @messagetype = N'http://schemas.microsoft.com/SQL/ServiceBroker/EndDialog'
                                BEGIN
                                    END CONVERSATION @dialog_handle;
                                END
                            END
                        END
                    -- Wstawianie otrzymanych rekordów do tabeli z rejestrowanymi danymi.
                    INSERT INTO dbo.LoginRecords
                    SELECT [message_body] AS LoginEntry FROM @LoginMessage

                    COMMIT TRANSACTION
                END TRY
            Begin Catch
                Rollback Transaction;

                DECLARE @ErrorMessage NVARCHAR(4000);
                DECLARE @ErrorSeverity INT;
                DECLARE @ErrorState INT;

```

```

        DECLARE @ErrorNumber INT;
        DECLARE @ErrorLine INT;
        DECLARE @ErrorProcedure NVARCHAR(128);
        SET @ErrorLine = ERROR_LINE();
        SET @ErrorSeverity = ERROR_SEVERITY();
        SET @ErrorState = ERROR_STATE();
        SET @ErrorNumber = ERROR_NUMBER();
        SET @ErrorMessage = ERROR_MESSAGE();
        SET @ErrorProcedure = ISNULL(ERROR_PROCEDURE(), 'None');
        RAISERROR (99124, @ErrorSeverity, 1, @ErrorNumber,
                  @ErrorSeverity, @ErrorState, @ErrorProcedure,
                  @ErrorLine, @ErrorMessage);

    EndCatch;
END -- WHILE
END -- ReceiveQueueHandler
GO

```

```

CREATE QUEUE LoginReceiveQueue
WITH STATUS=ON,
    RETENTION=OFF,
    ACTIVATION
        (STATUS=ON,
         PROCEDURE_NAME=dbo.ReceiveQueueHandler,
         MAX_QUEUE_READERS=10,
         EXECUTE AS OWNER),
    POISON_MESSAGE_HANDLING (STATUS = ON);
GO

```

```

CREATE SERVICE LoginReceiveSvc
AUTHORIZATION dbo
ON QUEUE dbo.LoginReceiveQueue
(LoginContract)
GO

```

```
-- SELECT * FROM LoginReceiveQueue
```

W tym przykładzie dla uproszczenia dwie bazy są tworzone w tym samym egzemplarzu systemu. Umieszczenie baz na różnych serwerach wymaga zabezpieczenia punktów końcowych i obsługi tras. Poniższy fragment kodu konfiguruje serwer, na którym rejestrowane są dane (ten kod znajdziesz w pliku *RejestrUzytKonfigPrzykladu.sql*):

```
-- Ten skrypt konfiguruje bazę wysyłającą dane w przykładzie rejestrowania danych o logowaniu.
-- Używana jest tu logika wysyłania danych opisana w tym rozdziale.
```

```

CREATE DATABASE LogSendExample
GO
USE LogSendExample
GO
-- Dla baz Service Brokera zawsze trzeba utworzyć klucz główny.
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'YourSecurePassword1!'
GO
-- Dane są przekazywane między bazami, a dla uproszczenia nie konfigurujemy
-- zabezpieczeń Service Brokera za pomocą certyfikatów, dlatego używane
-- muszą być zaufane bazy danych.
ALTER DATABASE LogSendExample SET TRUSTWORTHY ON
GO
-- Ta instrukcja może być zbędna, jest jednak potrzebna, jeśli nawiązujesz połączenie spoza domeny.

```

```
-- Użyliśmy konta SA, jednak można tu zastosować dowolne
-- uwierzytelnione konto systemu SQL Server.
ALTER AUTHORIZATION ON DATABASE::LogSendExample TO SA
GO
```

-- Tworzenie obiektów potrzebnych do nawiązania konwersacji z serwerem.

```
CREATE MESSAGE TYPE LoginMessage
AUTHORIZATION dbo
VALIDATION = NONE
GO
```

```
CREATE CONTRACT LoginContract
AUTHORIZATION dbo
(LoginMessage SENT BY ANY)
GO
```

```
-- Ta procedura obsługuje kolejkę inicjującą. Tu serwer nie wysyła odpowiedzi,
-- dlatego jedyne typy komunikatów, które trzeba obsługiwać, to komunikaty
-- o błędach i sygnały zakończenia konwersacji.
```

```
CREATE PROCEDURE dbo.SendQueueHandler
AS
BEGIN -- SendQueueHandler
    DECLARE @dialog_handle UNIQUEIDENTIFIER, @messagetype nvarchar(128);
    WHILE (1 = 1)
        BEGIN
            BEGIN TRY
                BEGIN TRANSACTION;

                RECEIVE TOP (1) @dialog_handle = [conversation_handle],
                    @messagetype = [message_type_name]
                FROM LoginSendQueue
                IF @@ROWCOUNT = 0
                    BEGIN
                        COMMIT TRANSACTION;
                        BREAK;
                    END
                ELSE IF @messagetype = N'http://schemas.microsoft.com/
                    SQL/ServiceBroker/Error'
                    BEGIN
                        -- Rejestrowanie błędu w dzienniku aplikacji.
                        END CONVERSATION @dialog_handle;
                        COMMIT TRANSACTION;
                        BREAK;
                    END
                ELSE IF @messagetype = N'http://schemas.microsoft.com/
                    SQL/ServiceBroker/EndDialog'
                    BEGIN
                        END CONVERSATION @dialog_handle;
                        COMMIT TRANSACTION;
                        BREAK;
                    END
            END TRY
            BEGIN Catch
                Rollback Transaction;
                DECLARE @ErrorMessage NVARCHAR(4000);
```

```

        DECLARE @ErrorSeverity INT;
        DECLARE @ErrorState INT;
        DECLARE @ErrorNumber INT;
        DECLARE @ErrorLine INT;
        DECLARE @ErrorProcedure NVARCHAR(128);
        SET @ErrorLine = ERROR_LINE();
        SET @ErrorSeverity = ERROR_SEVERITY();
        SET @ErrorState = ERROR_STATE();
        SET @ErrorNumber = ERROR_NUMBER();
        SET @ErrorMessage = ERROR_MESSAGE();
        SET @ErrorProcedure = ISNULL(ERROR_PROCEDURE(), 'None');
        RAISERROR (99123, @ErrorSeverity, 1, @ErrorNumber,
                  @ErrorSeverity, @ErrorState, @ErrorProcedure,
                  @ErrorLine, @ErrorMessage);

    EndCatch;
END -- WHILE
END -- SendQueueHandler
GO

-- Do kolejki inicjującej trafiają tylko komunikaty systemowe.
CREATE QUEUE LoginSendQueue
WITH STATUS=ON,
    RETENTION=OFF,
    ACTIVATION
        (STATUS=ON,
         PROCEDURE_NAME=dbo.SendQueueHandler,
         MAX_QUEUE_READERS=2,
         EXECUTE AS OWNER),
    POISON_MESSAGE_HANDLING (STATUS = ON);
GO

CREATE SERVICE LoginSendSvc
AUTHORIZATION dbo
ON QUEUE dbo.LoginSendQueue
(LoginContract)
GO

-- W tej tabeli przechowywany jest uchwyt do konwersacji z serwerem.
-- Zapisanie uchwytu w tym miejscu pozwala ponownie go wykorzystać.
CREATE TABLE dbo.SSB_Settings
([Source] sysname NOT NULL,
[Destination] sysname NOT NULL,
[Contract]sysname NOT NULL,
[dialog_handle] uniqueidentifier
CONSTRAINT PK_SSB_Setting PRIMARY KEY ([Source], [Destination], [Contract]))
GO

-- Ta procedura wysyła rekordy z danymi o loginach na serwer.
CREATE PROCEDURE dbo.SendUserLog
    @Destination sysname,
    @Source sysname,
    @Contract sysname,
    @MessageType sysname,
    @MessageBodyvarchar(MAX)
AS
DECLARE @dialog_handle uniqueidentifier

```

```
BEGIN TRANSACTION
```

```

/* Pobieranie identyfikatora konwersacji. */
SELECT @dialog_handle = dialog_handle
FROM dbo.SSB_Settings
WHERE [Source]= @Source
      AND [Destination] = @Destination
      AND [Contract]= @Contract;
/* Jeśli uchwyt nie jest dostępny, tworzona jest nowa konwersacja. */
IF @dialog_handle IS NULL
BEGIN
    /* Jeśli istnieje uchwyt konwersacji, należy przesłać do docelowego
       kodu sygnał o zakończeniu dawnej konwersacji. */
    IF @dialog_handle IS NOT NULL
    BEGIN
        UPDATE dbo.SSB_Settings
        SET dialog_handle = NULL
        WHERE [Source]= @Source
              AND [Destination] = @Destination
              AND [Contract]= @Contract;
        SEND ON CONVERSATION @dialog_handle
        MESSAGE TYPE EndOfConversation;
    END
    /* Konfigurowanie nowej konwersacji. */
    BEGIN DIALOG CONVERSATION @dialog_handle
    FROM SERVICE @Source
    TO SERVICE @Destination
    ON CONTRACT @Contract;
    /* Zapisanie identyfikatora nowej konwersacji. */
    UPDATE dbo.SSB_Settings
        SET dialog_handle = @dialog_handle
    WHERE [Source]= @Source
          AND [Destination] = @Destination
          AND [Contract] = @Contract;
    -- Jeśli instrukcja Update nie znalazła potrzebnego rekordu, należy go wstawić.
    IF @@ROWCOUNT = 0
        INSERT INTO dbo.SSB_Settings
            ([Source], [Destination], [Contract], [dialog_handle])
            VALUES
                (@Source, @Destination, @Contract, @dialog_handle);
END;
/* Wysyłanie komunikatu. */
SEND ON CONVERSATION @dialog_handle
MESSAGE TYPE @MessageType
(@MessageBody);

COMMIT TRANSACTION
GO

```

Ten fragment kodu tworzy aplikację kliencką wysyłającą komunikaty na serwer.

Następny fragment to kod testowy pokazujący, jak aplikacja kliencka wywołuje główną aplikację (kod znajdziesz w pliku *RejestrUzytTest.sql*). W prawdziwej aplikacji kod ten jest zaimplementowany jako usługa systemu Windows obsługująca zdalne loginy.

```

USE LogSendExample
GO

```


- Wywołaj ten skrypt, aby sprawdzić proces rejestrowania loginów. W rzeczywistym
- rozwiązaniu tę procedurę powinien wywoływać serwer bezpieczeństwa.

```
DECLARE @msg varchar(MAX)
SET @msg = 'Sam Smith - ' + CONVERT (varchar(100), GetDate(),109)
```

```
EXEC dbo.SendUserLog
    @Destination = 'LoginReceiveSvc',
    @Source = 'LoginSendSvc',
    @Contract = 'LoginContract',
    @MessageType = 'LoginMessage',
    @MessageBody = @msg
```

- Sprawdzanie, czy kod zadziałał.
- *SELECT * FROM LogReceiveExample.dbo.LoginRecords*

Podsumowanie

Choć Service Broker jest stosunkowo skomplikowany do skonfigurowania, daje bardzo duże możliwości, gdy potrzebne jest asynchroniczne przesyłanie komunikatów. Service Broker to elastyczne rozwiązanie, umożliwiające przekazywanie komunikatów w ramach bazy danych, między bazami z tego samego egzemplarza systemu SQL Server, a także między bazami z różnych serwerów (niezależnie od tego, czy znajdują się one obok siebie, czy na drugim krańcu świata).

W tym rozdziale zobaczyłeś, że trzeba skonfigurować wiele różnych obiektów. Choć Service Broker początkowo może wydawać się skomplikowany, po nabraniu doświadczenia w korzystaniu z niego okazuje się, że jest dość prosty w użyciu, a wszystkie jego elementy stają się zrozumiałe.

Integrowanie systemu SQL Server ze środowiskiem CLR

ZAWARTOŚĆ ROZDZIAŁU:

- Wykorzystanie środowiska CLR w systemie SQL Server.
- Tworzenie i instalowanie komponentów SQLCLR.
- Zabezpieczanie komponentów SQLCLR.
- Monitorowanie wydajności.

Programiści używający systemu SQL Server po raz pierwszy mogli zintegrować go ze środowiskiem .NET Common Language Runtime (CLR) w wersji SQL Server 2005. Wraz z ewoluowaniem wersji platformy .NET kolejne wydania systemu SQL Server umożliwiały integrację rozwiązań ze środowiskiem CLR. To środowisko daje wiele korzyści programistom i administratorom baz danych, dlatego jest cennym narzędziem, które warto znać.

Choć szczegółowe omawianie pisania kodu zarządzanego za pomocą języków platformy .NET (takich jak C#) wykracza poza zakres tego rozdziału, dowiesz się tu, jak utworzyć i zainstalować prosty komponent platformy .NET i umożliwić jego integrację z systemem SQL Server. Następnie dowiesz się, kiedy technologia SQLCLR sprawdza się lepiej od tradycyjnych rozwiązań opartych na języku T-SQL. W tym rozdziale omawiamy też wydajność i bezpieczeństwo technologii SQLCLR. Są to bardzo ważne kwestie, gdy programista decyduje się zastosować tę technologię zamiast tradycyjnego języka T-SQL.

Wprowadzenie do środowiska CLR

Dzięki integracji komponentów środowiska CLR platformy .NET z systemem SQL Server programiści mogą pisać procedury składowane, wyzwalacze, typy i funkcje zdefiniowane przez użytkownika oraz funkcje tabelowe za pomocą dowolnych języków obsługiwanych w tym środowisku (najpopularniejsze z nich to C# i Visual Basic .NET). Środowisko CLR umożliwia pisanie kodu zarządzanego, w którym można wykorzystać wiele usług nieudostępnianych przez silnik bazodanowy. Te usługi to np.: zarządzanie czasem życia obiektów, zabezpieczanie dostępu do kodu i integracja kodu w różnych językach. A to tylko wierzchołek góry lodowej. Jeśli dobrze przyjrzyć się możliwościom, jakie środowisko CLR zapewnia w silniku bazodanowym, odkryjesz wiele ważnych korzyści. Oto niektóre z nich.

- **Bogaty model programowania.** T-SQL to język dostępu do danych i zarządzania nimi. Zarządzany kod ze środowiska CLR lepiej nadaje się do tworzenia skomplikowanej logiki aplikacji. Ponadto platforma .NET udostępnia wiele wbudowanych klas. Istnieją np. klasy do zaawansowanego manipulowania łańcuchami znaków, wykonywania skomplikowanych operacji matematycznych i obsługi szyfrowania. Ponadto zarządzany kod ze środowiska CLR zapewnia solidną obsługę błędów i zarządzanie pamięcią.
- **Wyższa wydajność i skalowalność.** Ponieważ kod zarządzany ze środowiska CLR jest kompilowany, zapewnia wyższą wydajność w porównaniu z wykonywaniem skomplikowanej agregacji lub podobnych zadań za pomocą kursorów w języku T-SQL. Ponadto dzięki technologii SQLCLR interfejs API systemu SQL Server może kontrolować zarządzanie pamięcią w środowisku CLR, co pozwala skutecznie zsynchronizować pracę wątków środowiska CLR i systemu SQL Server.
- **Większe bezpieczeństwo.** Przy instalowaniu komponentów można przyznać im różne zbiory uprawnień. Najbezpieczniejszy jest zbiór SAFE. Pozwala on komponentowi na dostęp tylko do lokalnej bazy danych i uniemożliwia wywoływanie kodu niezarządzanego. Zbiór UNSAFE należy stosować dla zaufanego kodu, ponieważ pozwala na wywoływanie kodu niezarządzanego i daje dostęp do zewnętrznych zasobów. W ramach integracji ze środowiskiem CLR używane są też bardzo pewne i niezawodne zabezpieczenia dostępu do kodu. Pozwala to kontrolować uprawnienia kodu zarządzanego.
- **Wydajne programowanie za pomocą standardowego środowiska.** W ostatnich latach środowiska programistyczne platformy .NET i baz danych zostały lepiej zintegrowane w ramach narzędzia Visual Studio IDE (ang. *Integrated Development Environment*). Programiści mogą korzystać z tych samych narzędzi i z tego samego środowiska do pisania obiektów baz danych i skryptów, a także komponentów klienckich i warstwy pośredniej. Znacznie ułatwia to debugowanie kodu i przeprowadzanie zautomatyzowanych testów. Możesz też zintegrować rozwiązanie z systemem Team Foundation Server, aby zarządzać cyklem prac nad oprogramowaniem.

Jak widać, w niektórych sytuacjach wykorzystanie środowiska CLR to najbardziej niezawodne, skalowalne i bezpieczne rozwiązanie. Pamiętaj jednak, aby dostosować narzędzia do zadania. Nigdy nie traktuj środowiska CLR jako zastępnika języka T-SQL (wyjątkiem mogą być rozszerzone procedury składowane). Środowisko CLR należy stosować do szybkiego i wydajnego rozwiązywania problemów programistycznych, z którymi trudno sobie poradzić za pomocą języka T-SQL. Dotyczy to obszarów, takich jak parsowanie łańcuchów znaków, skomplikowane obliczenia matematyczne, operacje obciążające procesor i dostęp do zewnętrznych zasobów.

Wybór narzędzia utrudnia to, że język T-SQL cały czas jest rozwijany. Wiele zadań, z którymi w przeszłości trudno było sobie poradzić za pomocą tego języka, w jego nowszych wersjach można wykonać łatwo i wydajnie. Dlatego odpowiedź na to, której technologii należy używać,

brzmi: „To zależy”. Przeprowadź testy. Napisz rozwiązania przy użyciu języka T-SQL i platformy .NET, aby sprawdzić, które z nich działa lepiej. Im więcej będziesz korzystał z obu tych technologii, tym łatwiej będzie Ci stwierdzić, którą z nich powinieneś wykorzystać w danej sytuacji.

Zobaczmy teraz, w jaki sposób środowisko CLR jest zintegrowane z silnikiem bazodanowym.

System SQL Server jako host środowiska uruchomieniowego platformy .NET

Host środowiska uruchomieniowego to proces, który wczytuje środowisko uruchomieniowe platformy .NET i uruchamia kod w środowisku zarządzanym. Dzięki integracji ze środowiskiem CLR (technologia SQLCLR) system SQL Server umożliwia programistom używającym platformy .NET pisanie procedur składowanych, funkcji zdefiniowanych przez użytkownika, wyzwalaczy, typów zdefiniowanych przez użytkownika i typów złożonych w dowolnym języku zgodnym z platformą .NET (zwykle są to języki C# i VB .NET).

Jedną z wielkich zalet technologii SQLCLR jest to, że kod platformy .NET uruchamiany w systemie SQL Server jest zupełnie odizolowany od samego systemu SQL Server. Kod platformy .NET działa w przestrzeni procesów systemu SQL Server, jednak w systemie SQL Server stosowana jest wtedy *domena aplikacji* (AppDomain) z platformy .NET, która pozwala całkowicie odizolować i oddzielić wszystkie zasoby używane przez kod platformy .NET od zasobów wykorzystywanych przez system SQL Server. Obiekt AppDomain (opisany dalej) chroni system SQL Server przed szkodliwymi operacjami na zasobach systemowych. Warto zauważyć, że system SQL Server zarządza własnymi mechanizmami szeregowania wątków, synchronizacją, blokadami i pamięcią. Dodatkowo poprawia to bezpieczeństwo i wydajność technologii SQLCLR.

Domeny aplikacji

Głównym celem projektowym związanym z umieszczaniem komponentów w domenach aplikacji jest zapewnienie skalowalności, bezpieczeństwa i izolacji. Domeny aplikacji są dostępne od dość dawna i służą do izolowania aplikacji. Izolacja pozwala zagwarantować, że kod jednej aplikacji nie będzie wpływał na inne, niepowiązane aplikacje. Tego rodzaju izolacja występuje np. między systemem operacyjnym a środowiskami uruchomieniowymi.

Granice izolacji wyznaczone przez domeny aplikacji pomagają też zapewnić bezpieczeństwo i niezawodność niezbędne przy tworzeniu aplikacji. Ważne jest zwłaszcza izolowanie aplikacji działających na tym samym komputerze. Gdy na jednej maszynie pracuje wiele aplikacji, każda z nich jest wczytywana do odrębnych procesów. Pozwala to uzyskać potrzebny poziom izolacji, ponieważ adresy pamięci są powiązane z procesami.

Podobnie system SQL Server izoluje kod z różnych baz danych za pomocą domen aplikacji. Dla każdej bazy danych istnieje domena aplikacji, która umożliwia utworzenie, wczytanie i zarejestrowanie komponentu, a także wywoływanie metod i funkcji w bazie niezależnie od komponentów zarejestrowanych w innych bazach.

W jednej bazie może działać wiele komponentów. Ponadto komponent w trakcie pracy może wykrywać inne komponenty za pomocą interfejsów API mechanizmu refleksji z platformy .NET.

Język T-SQL a środowisko CLR

Po zintegrowaniu środowiska CLR z systemem SQL Server granica między warstwą logiki biznesowej a warstwą danych zaczęła się zacierać. Nie ma w tym nic złego — oznacza to tylko tyle, że wybór miejsca na kod warstwy pośredniej i warstwy dostępu do danych nie jest tak oczywisty jak kiedyś.

Zgodnie z najlepszymi praktykami do pobierania danych należy stosować język T-SQL. Natomiast manipulowanie danymi oraz funkcje i procedury mocno obciążające procesor można umieścić w kodzie zarządzanym, zwłaszcza gdy na zwróconych danych wykonywane mają być skomplikowane operacje (takie jak złożone obliczenia matematyczne lub parsowanie łańcuchów znaków).

Trzeba też uwzględnić miejsce wykonywania kodu. Czy aplikacja kliencka to najlepsze miejsce na umieszczenie danego kodu? A może lepiej będzie uruchamiać go po stronie serwera? W aplikacjach wielowarstwowych zwykle występuje warstwa danych i duża część kodu związanego z danymi jest uruchamiana na odrębnym serwerze lub na klienckiej stacji roboczej. Jednak w technologii SQLCLR zarówno kod w języku T-SQL, jak i kod zarządzany mogą działać na serwerze bazodanowym. Pozwala to wykorzystać duże możliwości obliczeniowe serwera i skrócić odległość między danymi a kodem. Czasem jednak lepiej umieścić kod po stronie klienta, ponieważ stacje robocze oferują coraz większe możliwości i mogą wykonywać dużą część zadań aplikacji bez wyraźnego spadku wydajności. W praktyce moc obliczeniowa wielu komputerów klienckich jest często w dużym stopniu niewykorzystana. To oznacza, że wiele zadań aplikacji można wykonywać po stronie klienta, co zwalnia zasoby serwera i pozwala wykorzystać je w innych celach.

Jeśli chodzi o wydajność, środowisko CLR umożliwia znacznie szybsze zwracanie informacji z plików systemu operacyjnego niż podejście oparte na języku T-SQL. Jest tak, ponieważ automatyzacja oparta na technologii OLE powoduje większe koszty niż różne metody używane w środowisku CLR. Jednak sama szybkość to nie wszystko. Należy też uwzględnić łatwość pisania i konserwowania kodu. W niektórych sytuacjach czynniki te są ważniejsze niż szybkość.

Jak widać, oba podejścia mają wiele wad i zalet. Pamiętaj, że kod zarządzany może działać zarówno po stronie klienta, jak i na serwerze. Kod w języku T-SQL działa tylko na serwerze.

Włączanie integracji ze środowiskiem CLR

Środowisko CLR domyślnie jest wyłączone dla użytkowników (jest natomiast zawsze dostępne dla systemu do użytku wewnętrznego). To oznacza, że nie można wykonywać kodu w języku platformy .NET, jeśli nie zostanie włączona obsługa środowiska CLR. Nie każdy może ją włączyć. Należy użyć do tego konta o roli sysadmin lub serveradmin z serwera lub dowolnego konta z przyznanym na poziomie serwera uprawnieniem do wykonywania instrukcji ALTER SETTINGS.

Aby włączyć środowisko CLR, wystarczy wywołać kwerendę. Kilkadziesiąt zaawansowanych ustawień systemu SQL Server można zmienić tylko za pomocą języka T-SQL. Nie znajdziesz ich w oknie dialogowym z właściwościami — są dostępne tylko przy użyciu opcji procedury `sp_configure`. Oto składnia służąca do włączania obsługi środowiska CLR:

```
EXEC sp_configure 'clr enabled', 1
GO
RECONFIGURE
GO
```

Jednak na razie nie włączaj środowiska CLR. Najpierw musisz zastanowić się nad kilkoma kwestiami.

Dostępne jest zaawansowane ustawienie `lightweight pooling`, którego włączenie blokuje wykonanie kodu środowiska CLR. Według dokumentacji MSDN przy włączonej puli lekkich wątków wykonywanie kodu środowiska CLR nie jest obsługiwane. Opcja `lightweight pooling` umożliwia zmniejszenie kosztów systemowych związanych ze zbyt częstym przełączaniem kontekstu. Oto opis z dokumentacji Books Online (<http://technet.microsoft.com/en-us/library/ms178074.aspx>):

„Gdy przełączanie kontekstu jest zbyt częste, pula lekkich wątków może zapewnić wyższą wydajność dzięki przełączaniu kontekstu w miejscu, co pomaga ograniczyć przejścia między trybem użytkownika a trybem jądra”.

Zgodnie z dokumentacją Books Online należy wyłączyć jedną lub drugą opcję (ustawić jedną z wartości na 0). Nie można włączyć ich obu (nie można ustawić wartości 1 dla obu wspomnianych opcji). Mechanizmy, które wymagają środowiska CLR, ale nie działają prawidłowo przy włączonej opcji `lightweight pooling`, to m.in. typ danych `Hierarchy`, replikacja i zarządzanie oparte na zasadach.

Opcja aktywująca obsługę środowiska CLR nie działa na zasadzie „włącz i zapomnij”. Gdy ta opcja jest włączona, należy bacznie śledzić dziennik błędów systemu SQL Server pod kątem kilku komunikatów o błędach. Oto dwa takie komunikaty.

- Failed Virtual Allocate Bytes: FAIL_VIRTUAL_RESERVE <size>
- Failed Virtual Allocate Bytes: FAIL_VIRTUAL_COMMIT <size>

Każdy z tych błędów może oznaczać, że system SQL Server próbuje zwolnić fragmenty puli pamięci systemu, aby uzyskać miejsce na elementy, takie jak rozszerzone procedury składowane, pliki `.dll` (pliki z kodem środowiska CLR) lub obiekty związane z automatyzacją.

Jeśli regularnie znajdujesz takie komunikaty w dzienniku błędów, pomoc może jedna z opcji uruchomieniowych systemu SQL Server. W programie SQL Server Configuration Manager użyj opcji `-g`, która informuje system SQL Server, aby pozostawił wolną pamięć możliwą do przydzielenia w procesie systemu SQL Server, ale poza pulą pamięci.

Jeśli jednak zastosujesz opcję `-g`, powinieneś obserwować zużycie pamięci. Opcja to dobry sposób na usprawnienie przydziału pamięci, jednak tylko wtedy, gdy dostępna pamięć fizyczna jest większa niż limit ustawiony przez system operacyjny. Niewłaściwe korzystanie z tej opcji może doprowadzić do wystąpienia błędów w działaniu systemu SQL Server lub uniemożliwić jego uruchomienie. Jeśli komponenty CLR wymagają dużej ilości pamięci, powinieneś przyrzeć się temu, co robią, a następnie wprowadzić niezbędne poprawki (np. podzielić komponent na mniejsze części).

Tworzenie komponentów CLR

W tym podrozdziale znajdziesz kilka przykładów ilustrujących tworzenie, instalowanie i rejestrowanie komponentów SQLCLR. W przykładach pokazano, jak utworzyć i zainstalować komponent CLR bez używania środowiska Visual Studio. Z dalszych przykładów dowiesz się, jak tworzyć i instalować takie komponenty za pomocą narzędzi SQL Server Data Tools ze środowiska Visual Studio.

Metoda bez używania środowiska Visual Studio

Tworzenie komponentów instalowanych w systemie SQL Server 2014 przypomina ich przygotowywanie we wcześniejszych wersjach. W ramach przykładu wykonaj opisane poniżej kroki.

1. Najpierw otwórz wybrany edytor tekstu i wprowadź skrypt z listingu 7.1 (jest on dostępny w pliku *PierwszyKompSqlClr.cs*).

Listing 7.1. Plik PierwszyKompSqlClr.cs

```
using System;
using System.Data;
using Microsoft.SqlServer.Server;
using System.Data.SqlTypes;
public class FirstSQLCLRProc
{
    [Microsoft.SqlServer.Server.SqlProcedure]
    public static void FirstSQLCLR(out string text)
    {
        SqlContext.Pipe.Send("Witaj, świecie!" + Environment.NewLine);
        text = "Mój pierwszy komponent SQLCLR!";
    }
}
```

Na listingu 7.1 zdefiniowana jest publiczna klasa zawierająca jedną metodę statyczną. Metoda ta korzysta z klas `SqlContext` i `SqlPipe`, które służą do tworzenia zarządzanych obiektów bazy danych umożliwiających wyświetlenie prostego komunikatu tekstowego. Obiekt `SqlContext` zapewnia środowisko, w którym kod komponentu jest aktywowany i uruchamiany. Kod zarządzany jest uruchamiany z poziomu serwera i działa dla połączenia z danym użytkownikiem (czyli w *kontekście* konta tego użytkownika). Na tym etapie obiekt `SqlPipe` jest już dostępny. `SqlPipe` to komponent systemu SQL Server, który umożliwia zarządzanym procedurom składowanym zwracanie wyników do jednostki wywołującej. Wyniki z kwerendy są odsyłane do klienta przy użyciu *potoku* jednostki wywołującej. Dla obiektów bazodanowych w środowisku CLR wygląda to tak samo — wyniki są odsyłane do klienta za pomocą metod obiektu `SqlPipe` (`Send` i `ExecuteAndSend`).

2. Po dodaniu tego kodu zapisz plik *PierwszyKompSqlClr.cs* w wybranym katalogu. Zapamiętaj tę lokalizację. W tym przykładzie używany jest język C#.
3. Następnie skompiluj komponent. System SQL Server domyślnie instaluje pliki platformy .NET, w tym pliki *csc.exe* i *vbc.exe*, czyli uruchamiane z wiersza polecen kompilatory języków C# i VB .NET. W platformie .NET 4.0 te pliki znajdują się w katalogu:

C:\Windows\Microsoft.NET\Framework\v4.0.30319

Otwórz wiersz poleceń i przejdź do katalogu, w którym wcześniej zapisałeś plik .cs.

4. Za pomocą poniższej składni skompiluj kod, aby uzyskać komponent:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc /
    target:library pierwszykompsqlclr.cs
```

Opcja `/target` to parametr, który informuje, że należy skompilować kod do postaci biblioteki.

5. Po skompilowaniu programu do postaci biblioteki w katalogu powinien pojawić się nowy plik o takiej samej nazwie jak plik źródłowy, ale z rozszerzeniem *.dll*. To ten plik zostanie wczytany do systemu SQL Server.

- Uruchom program SQL Server Management Studio i otwórz nowe okno kwerendy. Wybierz bazę, do której chcesz wczytać komponent, a następnie wpisz poniższy kod w języku T-SQL. W tym przykładzie zakładamy, że na dysku głównym napędu C znajduje się katalog *temp* (C:\temp). W przeciwnym razie musisz podać w klauzuli FROM lokalizację komponentu.

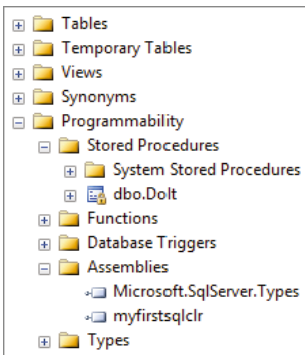
```
CREATE ASSEMBLY myfirstsqlclr
FROM 'c:\temp\pierwszykompsqlclr.dll'
WITH PERMISSION_SET = SAFE
```

Ten kod wczytuje komponent do systemu SQL Server i tworzy referencję do niego na podstawie lokalizacji skompilowanego komponentu przygotowanego we wcześniejszym kroku. Zwróć uwagę na opcję PERMISSION_SET. Opcja ta określa poziom bezpieczeństwa komponentu. Więcej na temat dostępnych opcji dowiesz się dalej.

- Następnie utwórz w języku T-SQL procedurę składowaną o nazwie DoIt. Zastosuj w tej procedurze referencję do komponentu przygotowaną w kroku 6. Wprowadź poniższy kod w języku T-SQL w oknie kwerend i wykonaj go:

```
CREATE PROCEDURE DoIt
@i nchar(50) OUTPUT
AS
EXTERNAL NAME myfirstsqlclr.FirstSQLCLRProc.FirstSQLCLR
```

Po utworzeniu tej procedury możesz zobaczyć nowy komponent i nową procedurę składowaną w oknie *Object Explorer*, co ilustruje rysunek 7.1.



Rysunek 7.1. Nowe elementy w oknie Object Explorer

- Gdy procedura składowana jest już gotowa, możesz ją uruchamiać, tak jak dowolne inne procedury, za pomocą standardowego kodu w języku T-SQL. W oknie kwerend wpisz poniższy kod w tym języku i uruchom go. Jeśli jeszcze nie włączyłeś integracji ze środowiskiem CLR, koniecznie zrób to przed wykonaniem tego kodu.

```
DECLARE @var nchar(50)
EXEC DoIt @var out
PRINT @var
```

W oknie *Results* powinieneś zobaczyć następujące wyniki:

```
Witaj, świecie!
Mój pierwszy komponent SQLCLR!
```

Nie jest to jedyny sposób dodawania komponentów. W następnym ćwiczeniu do tworzenia i instalowania komponentów CLR wykorzystasz środowisko Visual Studio i narzędzia SQL Server Data Tools.

Używanie narzędzi Microsoft SQL Server Data Tools

Tworzenie i instalowanie obiektów SQLCLR za pomocą narzędzi SSDT (ang. *SQL Server Data Tools*) to łatwy sposób przygotowywania i edytowania obiektów bazodanowych. W tym punkcie zamieszczono krótki przykład pokazujący, jak używać narzędzi SSDT do tworzenia i instalowania procedur składowanych w technologii SQLCLR. Jeśli jeszcze nie włączyłeś integracji ze środowiskiem CLR, koniecznie zrób to przed wykonaniem ćwiczenia.

1. Zainstaluj narzędzia SSDT. Możesz je pobrać z adresu URL <http://msdn.microsoft.com/en-us/data/tools.aspx>. Wybierz wersję, która obsługuje system SQL Server 2014.
2. Po zainstalowaniu narzędzi otwórz środowisko Visual Studio i wybierz opcję *View/SQL Server Object Explorer (SSOE)*. Po lewej stronie środowiska IDE Visual Studio pojawi się okno *SQL Server Object Explorer*.
3. W oknie SSOE kliknij przycisk *Add SQL Server*, aby zarejestrować nowy egzemplarz systemu SQL Server. W oknie dialogowym *Connect to Server* wprowadź nazwę serwera i dane uwierzytelniające, a następnie kliknij przycisk *Connect*. To spowoduje zarejestrowanie nowego systemu SQL Server w oknie SSOE.
4. Po zarejestrowaniu nowego serwera rozwiń jego węzeł. Zobaczysz węzły baz danych. Wybierz bazę, w której chcesz zainstalować komponent SQLCLR, a następnie kliknij tę bazę prawym przyciskiem myszy. W menu kontekstowym wybierz opcję *Create New Project*. Pojawi się okno dialogowe o tej samej nazwie. W tym oknie wprowadź nazwę i lokalizację nowego projektu. Następnie zaznacz pola *Create New Solution* i *Create Directory for Solution*. Kliknij przycisk *Start*, aby utworzyć nowy projekt bazodanowy powiązany z bazą wybraną w oknie *SQL Server Object Explorer*.
5. Po utworzeniu nowego projektu kliknij jego węzeł w oknie *Solution Explorer*. W menu kontekstowym wybierz opcję *Properties*, aby otworzyć ekran właściwości projektu.
6. W zakładce *Project Settings* sprawdź, czy docelową platformą (*Target Platform*) jest SQL Server 2014. Jako platformę można też ustawić kilka innych wersji systemu SQL Server. W zakładce *SQLCLR* możesz określić poziom uprawnień, a także docelową wersję platformy .NET. Zamknij ekran właściwości.
7. Ponownie kliknij projekt prawym przyciskiem myszy w oknie *Solution Explorer* i w menu kontekstowym wybierz opcję *Add/New Item*. W oknie *Add New Item* wybierz *SQL CLR C#* z listy zainstalowanych szablonów. Jeśli nie widzisz tej opcji, musisz zainstalować potrzebny szablon lub zastosować domyślny szablon dla języka VB. Następnie wybierz opcję *SQL CLR C# Stored Procedure*. Zachowaj domyślną nazwę i kliknij przycisk *OK*. Do projektu dodany zostanie nowy plik klasy C# SQLCLR. Plik ten jest automatycznie otwierany i możesz zacząć wpisywać w nim kod.

W klasie zmodyfikuj metodę, tak aby zwracała łańcuch znaków. Dodaj też kod w języku C# zastosowany w przykładzie, w którym nie używano środowiska Visual Studio.

```
public static void SqlStoredProcure1(out string text)
{
    SqlContext.Pipe.Send("Witaj, świecie!" + Environment.NewLine);
    text = "Mój drugi komponent SQLCLR!";
}
```

8. Kliknij prawym przyciskiem myszy projekt w oknie *Solution Explorer* i wybierz opcję *Build* z menu kontekstowego. Po skompilowaniu projektu następnym krokiem jest zainstalowanie nowej procedury składowanej CLR. Kliknij projekt prawym przyciskiem

myszy w oknie *Solution Explorer* i wybierz opcję *Publish* z menu kontekstowego, aby otworzyć okno dialogowe *Publish Database*.

9. W oknie dialogowym *Publish Database* kliknij przycisk *Edit*, aby ustawić docelowy serwer i bazę danych. Następnie w tym oknie dialogowym kliknij przycisk *Publish*. To spowoduje spakowanie zawartości projektu bazodanowego i zainstalowanie go w wybranej bazie. W tej sytuacji projekt obejmuje wyłącznie procedurę składowaną SQLCLR, dlatego tylko ona zostanie zainstalowana w bazie.

Informacje o procesie publikacji projektu zobaczysz w oknie *Data Tools Operations*. Okno to wyświetla informacje o postępie publikacji, komunikaty i błędy. Publikowanie komponentu CLR zajmie około minuty, a po zakończeniu tego procesu powinieneś zobaczyć listę komponentów w oknie *SQL Server Object Explorer* w środowisku Visual Studio.

W oknie *SQL Server Object Explorer* można uruchamiać procedury składowane, klikając je prawym przyciskiem myszy. Wystarczy w menu kontekstowym wybrać opcję *Execute Procedure*. Powinieneś zobaczyć komunikat zwrócony przez daną procedurę.

Jak widać, tworzenie i publikowanie komponentów SQLCLR za pomocą narzędzi SSDT jest łatwiejsze, ponieważ ta metoda zapewnia prostszy i wydajniejszy sposób pracy z obiektami bazodanowymi (w tym z obiektami CLR).

Zabezpieczenia przy integracji ze środowiskiem CLR

Podobnie jak przy wykonywaniu dowolnego kodu, także i teraz trzeba zadbać o to, by kod SQLCLR był bezpieczny i odizolowany. Jeśli chcesz zapewnić bezpieczne wykonywanie kodu środowiska CLR, powinieneś uwzględnić następujące kwestie.

- Uruchamianie zarządzanego kodu w systemie SQL Server nie powinno naruszać integralności, stabilności i niezawodności tego systemu.
- Zarządzany kod nie powinien umożliwiać niewierzytelnionego dostępu do danych użytkownika i innego kodu użytkownika.
- Należy dodać mechanizmy, które będą ograniczać kodowi użytkownika możliwość dostępu do zasobów spoza systemu SQL Server.
- To, że zarządzany kod działa w procesie systemu SQL Server, nie oznacza, że powinien mieć dostęp do zasobów systemowych.

Aby ułatwić zastosowanie się do tych wskazówek, w środowisku CLR wprowadzono dla zarządzanego kodu model bezpieczeństwa CAS (ang. *Control Access Security*). W tym modelu uprawnienia są przyznawane komponentom na podstawie tożsamości kodu. Zestaw uprawnień, które można przyznać komponentom w hoście systemu SQL Server, jest zależny od uprawnień określonych w trakcie tworzenia tego komponentu w tym systemie. W SQLCLR dostępne są trzy zestawy uprawnień.

- **SAFE.** Przy tym poziomie dozwolone są tylko dostęp do danych lokalnych i obliczenia wewnętrzne. Jeśli w trakcie tworzenia komponentu nie ustawiono uprawnień, domyślnie używany jest zestaw **SAFE**. Nie daje on dostępu do zewnętrznych zasobów systemowych (np. plików lub rejestru).

- **EXTERNAL_ACCESS.** Zapewnia te same uprawnienia, co zestaw SAFE, a dodatkowo umożliwia dostęp do zasobów zewnętrznych, takich jak system plików, rejestr, sieci i zmienne środowiskowe.
- **UNSAFE.** Oferuje nieograniczony dostęp do wszystkich zasobów z systemu SQL Server i spoza niego. Jest to najmniej bezpieczne ustawienie i nie należy go używać zbyt często.

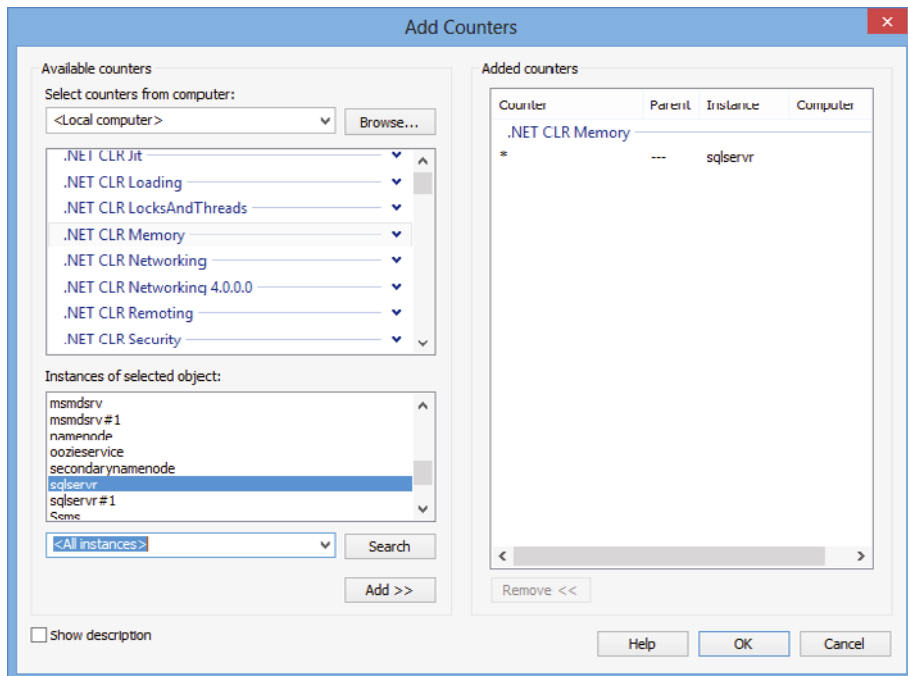
Ważne jest, aby ustawić dla komponentów właściwy poziom bezpieczeństwa. Większość z nich powinna działać z zestawem SAFE. Technologia SQLCLR niesłusznie uważana jest za niebezpieczną. W rzeczywistości jest to bezpieczne rozwiązanie, dla którego obowiązują liczne ograniczenia.

Monitorowanie wydajności

Wcześniej wspomniano już, że integracji ze środowiskiem CLR nie należy włączać na zasadzie „ustaw i zapomnij”. Gdy obsługa tego środowiska jest włączona, trzeba sprawdzać, czy kod działa zgodnie z oczekiwaniami i robi to, co powinien. To monitorowanie ułatwiają różne narzędzia i opcje, w tym Windows System Monitor, zdarzenia rozszerzone i widoki DMV (ang. *Dynamic Management Views*).

Windows System Monitor

Windows System Monitor (program *PerfMon.exe*) pozwala monitorować działanie kodu CLR w systemie SQL Server. Aby śledzić liczniki CLR na potrzeby systemu SQL Server, należy wybrać licznik z grupy .NET CLR oraz ustawić egzemplarz *sqlservr*, co przedstawiono na rysunku 7.2.

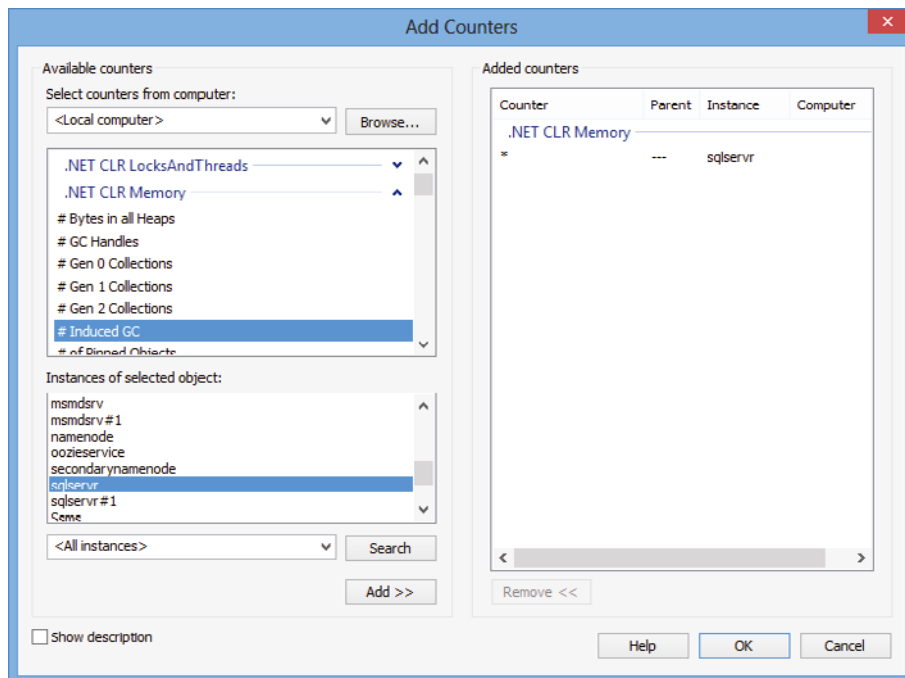


Rysunek 7.2. Monitorowanie liczników CLR

Wymienione poniżej liczniki są bardzo przydatne, jeśli chcesz zrozumieć stan i aktywność programów działających w środowisku systemu SQL Server.

- **.NET CLR Memory.** Zapewnia szczegółowe informacje na temat pamięci na sterce CLR i przywracania pamięci. Licznik ten można wykorzystać do monitorowania wykorzystania pamięci przez CLR i zgłaszania alertów, gdy ilość zajętej pamięci jest za duża. Jeśli kod kopiuje do pamięci duże ilości danych, czasem trzeba przyjrzeć się jego działaniu i zastosować inne rozwiązanie, aby zmniejszyć wykorzystanie pamięci (inna możliwość to udostępnienie dodatkowej pamięci).
- **.NET CLR Loading.** System SQL Server izoluje kod z różnych baz danych za pomocą domen aplikacji (obiekty AppDomain). Licznik ten umożliwia śledzenie liczby domen aplikacji i liczby komponentów wczytanych w systemie. Za pomocą tego licznika można określić wczytane komponenty CLR.
- **.NET CLR Exceptions.** Licznik *Exceptions/Sec* pozwala określić, jak dużo wyjątków generuje dany kod. Rejestrowane wartości są różne w zależności od aplikacji, ponieważ programiści czasem używają wyjątków do testowania funkcji programów. Dlatego powinienś monitorować wyniki przez pewien czas, aby ustalić punkt odniesienia. Wzrost tej wartości związany jest ze spadkiem wydajności.

Na rysunku 7.3 przedstawiono licznik *.NET CLR Memory: # Induced GC*, który wyświetla maksymalną liczbę operacji przywracania pamięci w wyniku bezpośredniego wywołania polecenia `GC.Collect`. Warto pozwolić mechanizmowi przywracania pamięci na sterowanie częstotliwością wykonywania tej operacji. Jeśli w tym liczniku zauważysz wartość większą od 0, powinienś sprawdzić, czy obiekty CLR nie wywołują bezpośrednio mechanizmu przywracania pamięci. Jeśli to robią, upewnij się, że jest to uzasadnione.

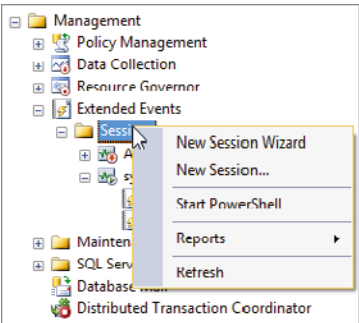


Rysunek 7.3. Licznik *.NET CLR Memory: # Induced GC*

UWAGA Więcej informacji na temat różnych liczników platformy .NET znajdziesz na stronie <http://msdn.microsoft.com/en-us/library/w8f5kw2e.aspx>.

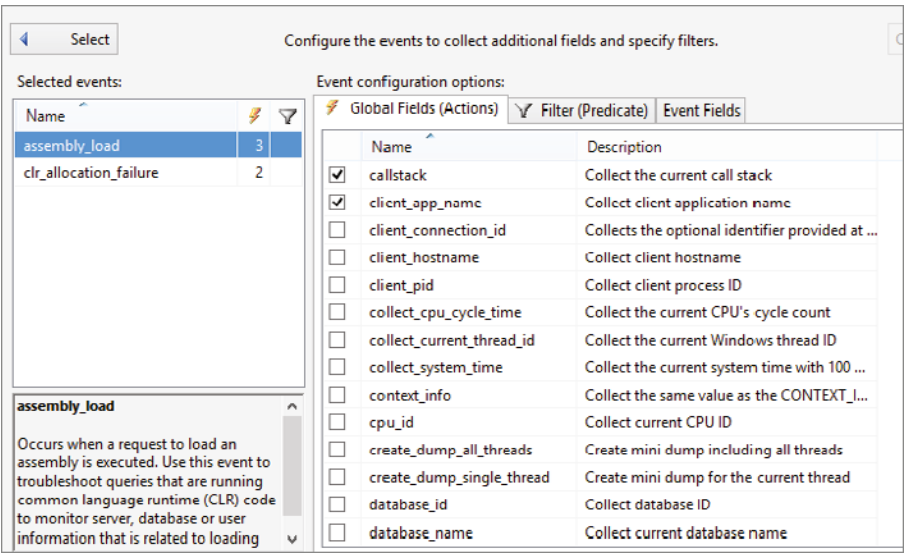
Rozszerzone zdarzenia

Używanie rozszerzonych zdarzeń to doskonały sposób na monitorowanie stanu systemu SQL Server. Na rysunku 7.4 przedstawiono węzeł *Extended Events* w węźle *Management* w programie SQL Server Management Studio. Po kliknięciu katalogu *Sessions* prawym przyciskiem myszy możesz utworzyć nową sesję.



Rysunek 7.4. Węzeł *Extended Events* w programie SQL Server Management Studio

Dwa najważniejsze rozszerzone zdarzenia związane z monitorowaniem obiektów CLR to *assembly_load* i *clr_allocation_failure*. Zdarzenie *assembly_load* jest związane z żądaniem wczytania komponentu i służy do monitorowania kwerend uruchamiających kod CLR. Zdarzenie *clr_allocation_failure* dotyczy niepowodzenia przy przydziale pamięci w kodzie zarządzanym. Na rysunku 7.5 przedstawiono opcje konfiguracyjne zdarzenia *assembly_load*.



Rysunek 7.5. Konfigurowanie zdarzenia *assembly_load*

Widoki DMV

Widoki DMV zwracają informacje o stanie serwera, które można wykorzystać do monitorowania stanu egzemplarza systemu, diagnozowania problemów i optymalizowania wydajności. Oto cztery widoki DMV związane z technologią SQLCLR.

- `sys.dm_clr_appdomains`. Widok zwraca wiersze z danymi każdej domeny aplikacji z serwera.
- `sys.dm_clr_loaded_assemblies`. Widok zwraca wiersze dla każdego zarządzanego komponentu użytkownika wczytanego do przestrzeni adresowej serwera.
- `sys.dm_clr_properties`. Widok zwraca wiersz dla każdej właściwości związanej z mechanizmami integracji systemu SQL Server ze środowiskiem CLR (w tym wersję i stan używanego środowiska CLR).
- `sys.dm_clr_tasks`. Widok zwraca wiersz dla każdego wykonywanego obecnie zadania CLR.

Choć te cztery widoki DMV udostępniają bardzo wartościowe informacje, nie zapewniają danych potrzebnych do optymalizowania wydajności. Aby lepiej zrozumieć działanie komponentów CLR, można wykorzystać poniższe widoki DMV. Udostępniają one dane na temat opisanych niżej komponentów.

- `sys.dm_exec_cached_plans`. Za pomocą tego widoku można wyświetlić zapisany w pamięci podręcznej plan wykonania kwerendy CLR.
- `sys.dm_exec_query_stats`. Widok zawiera wiersz dla każdej instrukcji z zapisanego w pamięci podręcznej planu wykonania kwerendy.

Przy użyciu tych widoków DMV możesz uzyskać zagregowane dane na temat wydajności zapisanych w pamięci podręcznej planów wykonania kwerend. Informacje te pomogą w ustaleniu wydajności kwerend z komponentów.

Cele projektowe dotyczące integracji ze środowiskiem CLR

Przy integrowaniu technologii SQLCLR z rozwijaną architekturą należy uwzględnić następujące cele projektowe.

- **Wydajność.** Zgodnie z najlepszymi praktykami komponent CLR nie powinien być używany do pobierania danych. Do takich zadań służy kod w języku T-SQL. Nie należy marnować czasu komponentów CLR na dostęp do danych. Prześlij potrzebne dane do komponentu, zamiast wymuszać na nim pobieranie ich z systemu SQL Server.
- **Skalowalność.** System SQL Server i technologia SQLCLR zarządzają pamięcią w odmienny sposób. W systemie SQL Server ważniejsze jest współdziałanie i obsługiwany jest model wątków bez wyłuszczenia. W środowisku CLR używany jest model wątków z wyłuszczeniem i nie obowiązuje rozróżnienie na pamięci fizyczną i wirtualną. Te różnice są ciekawym wyzwaniem dla programistów budujących systemy, które mają być skalowalne. Dlatego w trakcie projektowania architektury bazy danych zadaj o to, aby nie naruszyć skalowalności i integralności systemu przez kod użytkownika niepotrzebnie bezpośrednio wywołujący interfejsy API do obsługi wątków i pamięci.
- **Bezpieczeństwo.** Opracuj plan zabezpieczeń i zestaw zasad na potrzeby korzystania z technologii SQLCLR. Stosuj się do ustalonych reguł. Upewnij się, że zarządzany kod działa zgodnie z regułami uwierzytelniania i autoryzacji obowiązującymi w systemie SQL Server. Ponadto nie każdy fragment zarządzanego kodu musi mieć dostęp do zewnętrznych zasobów.

- **Niezawodność.** Najważniejsze w tym aspekcie jest to, że kod użytkownika nie może mieć uprawnień do wykonywania operacji, które mogą naruszyć integralność bazy danych i silnika bazodanowego (w tym nie może zastępować danych w wewnętrznych strukturach danych i buforach pamięci).

Podsumowanie

Integracja z technologią SQLCLR otwiera przed programistami liczne możliwości w zakresie pobierania i przetwarzania danych. Nie jest to żadna nowość, ponieważ technologia jest dostępna już od ponad 6 lat. W tym rozdziale chcieliśmy jedynie pokazać, dlaczego warto rozważyć zastosowanie technologii SQLCLR i czego można od niej oczekiwać. Znalazły się tu także porady dotyczące korzystania z niej.

Doskonałą pomocą przy stosowaniu technologii SQLCLR są narzędzia SQL Server Data Tools. Umożliwiają one łatwą pracę z obiektami baz danych. Można używać różnych rodzajów obiektów SQLCLR, w tym procedur składowanych i funkcji zdefiniowanych przez użytkownika. Niezależnie od tworzonego typu obiektu CLR ważne jest, aby zrozumieć, dlaczego używana jest technologia SQLCLR zamiast języka T-SQL. Choć SQLCLR nie jest zastępnikiem języka T-SQL, stanowi doskonałe narzędzie, umożliwiające wykonywanie zadań, których realizacja za pomocą samego języka T-SQL jest trudna lub niemożliwa.

Kod zarządzany ma swoje miejsce w systemie SQL Server. Nawet w silniku systemu SQL Server technologia CLR jest używana w niektórych typach danych (np. w typie danych xml i w typach związanych z geografią). Używanie technologii SQLCLR ma zalety i im więcej będziesz się nią posługiwał, tym łatwiej będzie Ci znaleźć miejsca, w których pozwala usprawnić aplikację.

W tym rozdziale opisano pewne zagadnienia z zakresu bezpieczeństwa przy integracji ze środowiskiem CLR. W rozdziale 8. omawiamy ważny temat zabezpieczania całych baz danych. Ochrona baz danych jest oparta na uwierzytelnianiu i autoryzacji. Uwierzytelnianie polega na potwierdzeniu tożsamości użytkownika, a autoryzacja związana jest z przyznawaniem lub odmową dostępu do obiektów bazy danych. Po zakończeniu lektury tego rozdziału będziesz dobrze rozumiał oba te procesy.

Zabezpieczanie egzemplarzy baz danych

ZAWARTOŚĆ ROZDZIAŁU:

- Metody uwierzytelniania w systemie SQL Server.
- Metody uwierzytelniania w systemie Windows.
- Zabezpieczenia oparte na autoryzacji na poziomie obiektów.
- Zapewnianie bezpieczeństwa na poziomie wierszy.

Zabezpieczanie egzemplarzy systemu Microsoft SQL Server to jeden z najmniej popularnych tematów związanych z tym narzędziem. Niestety, sytuacja ta w najbliższych latach (a może i nigdy) prawdopodobnie się nie zmieni. Jednak właściwe zabezpieczenie egzemplarza bazy danych jest niezwykle istotne, ponieważ bez tego nie da się zagwarantować, że dane przechowywane w egzemplarzu systemu SQL Server są zgodne z oczekiwanymi.

Napastnik może wprowadzić w danych z egzemplarza systemu SQL Server drobne zmiany, np. zmodyfikować nazwy lub ceny produktów, ale może też wstrzyknąć kod w językach JavaScript lub HTML, który zostanie przesłany do przeglądarek klientów i pracowników, a następnie spowoduje wykonanie niepożądanego kodu na komputerach tych osób. Zmiany wprowadzane przez taki kod mogą być niewielkie, jednak częściej prowadzi on do zainstalowania na komputerach użytkowników niebezpiecznych aplikacji, np. trojanów lub programów rejestrujących wciśnięte klawisze.

Dlatego najlepiej przygotować się na wszelkie możliwe scenariusze. Korporacyjne bazy danych to strategiczny zasób, dlatego trzeba chronić przechowywane w nich dane przed kradzieżą i manipulacją. Nie chcesz chyba, żeby Twoja firma trafiła na okładki gazet z powodu przejęcia danych klientów przez złodziei tożsamości.

Przy zabezpieczaniu oprogramowania podstawowymi kwestiami są uwierzytelnianie i autoryzacja. Uwierzytelnianie to sposób identyfikowania przez oprogramowanie osoby, która próbuje nawiązać połączenie. Autoryzacja polega na określaniu, jakie operacje użytkownik może wykonywać po nawiązaniu połączenia. Uwierzytelnianie blokuje dostęp niepożądanym osobom, a autoryzacja uniemożliwia uwierzytelnionym użytkownikom wykonywanie niedozwolonych operacji. System SQL Server 2014 udostępnia bardzo zaawansowane funkcje uwierzytelniania i autoryzacji, które dają wiele możliwości w zakresie zabezpieczania danych. W tym rozdziale poznasz dostępne opcje i dowiesz się, jak je skonfigurować.

Rodzaje uwierzytelniania

Zabezpieczanie danych obejmuje dwa podstawowe etapy — *uwierzytelnianie* (polega na ustaleniu, czy użytkownik jest tym, za kogo się podaje) i *autoryzację* (określa dane, do których dana osoba ma dostęp, i operacje, jakie można na nich wykonywać). W systemie SQL Server uwierzytelnianie wymaga podania czegoś, do czego masz dostęp tylko Ty sam — odcisków palców, karty elektronicznej, hasła itd.

W egzemplarzu systemu Microsoft SQL Server obsługiwane są dwie metody uwierzytelniania, uwierzytelnianie z systemu SQL Server i uwierzytelnianie z systemu Windows. Gdy instalujesz egzemplarz systemu SQL Server, możesz określić, czy ma on obsługiwać tylko uwierzytelnianie z systemu Windows, czy obie dostępne metody.

Uwierzytelnianie z systemu SQL Server

Jest to pierwotna metoda uwierzytelniania obsługiwana w systemie SQL Server jeszcze wtedy, gdy był oparty na kodzie firmy Sybase. W uwierzytelnianiu z systemu SQL Server aplikacja lub użytkownik muszą podać nazwę i hasło, aby uwierzytelnić się w egzemplarzu tego systemu. Nazwa użytkownika i hasło są zapisywane w łańcuchu znaków połączenia, którego aplikacja używa przy nawiązywaniu połączenia z egzemplarzem systemu SQL Server. W tym modelu nazwa użytkownika i hasło są przechowywane w bazie master w danym egzemplarzu systemu.

Przy korzystaniu z uwierzytelniania z systemu SQL Server konto i hasło są przekazywane do egzemplarza systemu, który tworzy skrót hasła i porównuje nazwę użytkownika oraz skrót hasła z listą kont systemu SQL Server zapisanych w bazie master. Jeśli przekazane nazwa użytkownik i skrót hasła pasują do konta zapisanego w tej bazie, uwierzytelnianie kończy się powodzeniem. Można nawiązać połączenie i użytkownikowi przyznawane są uprawnienia powiązane z jego kontem w systemie. Jeżeli podane informacje nie pasują do danych z bazy, zwracana jest informacja o niepowodzeniu uwierzytelniania.

Możesz tak skonfigurować uwierzytelnianie w systemie SQL Server, aby było zgodne z zasadami bezpieczeństwa hasel z systemu Windows. W ten sposób można wymusić siłę hasła i określić czas jego wygasania. To oznacza, że nawet jeśli w danym egzemplarzu używane jest uwierzytelnianie z systemu SQL Server, hasła można chronić za pomocą zasad bezpieczeństwa zdefiniowanych w firmie dla hasel z systemu Windows.

UWAGA Konfigurowanie zasad dotyczących hasel jest bardzo skomplikowane. Omawianie tego tematu wykracza poza zakres książki.

W domenie można włączyć dwie zasady z tego obszaru: pierwsza związana jest z przestrzeganiem zasad domeny kontrolujących złożoność haseł, a druga dotyczy wygasania haseł.

Zasady dotyczące haseł można ustawić w programie SQL Server Management Studio. W tym celu należy rozpocząć edycję loginu, a następnie włączyć lub wyłączyć wybrane zasady. Do włączania i wyłączania ustawień służy instrukcja `ALTER LOGIN`, co ilustruje poniższy fragment kodu:

```
ALTER LOGIN chain_test
WITH CHECK_POLICY = ON, CHECK_EXPIRATION=ON
```

Zasady dotyczące haseł są sprawdzane tylko w momencie tworzenia konta lub zmiany hasła. Można więc utworzyć login systemu SQL Server z hasłem niezgodnym z zasadami z domeny, a dopiero potem włączyć sprawdzanie tych zasad. Wtedy system przyjmie, że konto jest zgodne z zasadami, choć będzie to niezgodne z prawdą.

Uwierzytelnianie z systemu Windows

Uwierzytelnianie z systemu Windows wprowadzono w systemie SQL Server w wersji 6.0. W tym modelu system SQL Server przekazuje zadanie uwierzytelniania użytkowników systemowi Windows. Windows po nawiązaniu połączenia przez użytkownika przekazuje jego tożsamość do systemu SQL Server. Gdy stworzysz login w tym podejściu, wiążesz tożsamość użytkownika z systemu Windows z loginem z systemu SQL Server. Oprócz stosowanej metody uwierzytelniania loginy w obu tych systemach niczym się nie różnią. Uprawnienia do łączenia się z egzemplarzem systemu SQL Server i bazami danych z tego egzemplarza można przyznawać poszczególnym kontom systemu Windows, a także grupom tworzonym i zarządzanym w usługach Active Directory systemu Windows.

W trakcie łączenia się z bazą SQL Server za pomocą uwierzytelniania z systemu Windows to nie system SQL Server odpowiada za uwierzytelnianie. W momencie logowania się użytkownika do systemu operacyjnego Windows kontroler domeny generuje token uwierzytelniający, który jest następnie przekazywany do komputera klienckiego i zapisywany w jego pamięci. Przy próbie nawiązania połączenia z systemem SQL Server za pomocą uwierzytelniania z systemu Windows token ten jest przekazywany z systemu operacyjnego komputera klienckiego do systemu SQL Server. System SQL Server kontaktuje się wtedy z kontrolerami domeny dla swojego systemu operacyjnego, aby ustalić, czy dany token jest prawidłowy. System sprawdza, czy określony identyfikator SID (ang. *Security Identifier*) może uzyskać dostęp do danego egzemplarza systemu SQL Server i jakie uprawnienia powinien otrzymać.

Używana przy uwierzytelnianiu tożsamość z systemu Windows może odpowiadać pojedynczemu użytkownikowi lub grupie. Na potrzeby uwierzytelniania w systemie SQL Server zaleca się wykorzystywanie grup systemu Windows. Dzięki temu administrator systemu SQL Server musi tylko skonfigurować niewielką liczbę grup, a następnie administratorzy zabezpieczeń systemu Windows mogą kontrolować dostęp do baz dzięki przypisywaniu użytkowników do grup systemu Windows powiązanych z loginami do systemu SQL Server.

UWAGA Kompletny proces uwierzytelniania z systemu Windows jest skomplikowany. Jego dokumentację znajdziesz w kilku miejscach. Gdy poszukasz informacji w witrynie Microsoft TechNet, znajdziesz różne artykuły. Ponadto rozdział 3. książki *Securing SQL Server: Protecting Your Database from Attackers* (Danny Cherry, Amsterdam, Holandia, Syngress, 2010) zawiera szczegółowy opis procesu uwierzytelniania z systemu Windows.

Porównanie obu metod uwierzytelniania

Z powodu różnic w uwierzytelnianiu z systemów SQL Server i Windows to pierwsze jest uznawane za dużo mniej bezpieczne. Jeśli to możliwe, nie należy korzystać z uwierzytelniania z systemu SQL Server. Gdy używana jest ta technika, napastnicy mogą wykorzystać atak siłowy, aby spróbować włamać się do systemu SQL Server. Dzieje się tak, ponieważ nazwa użytkownika i hasło są bezpośrednio przesyłane do systemu SQL Server. W procesie uwierzytelniania z systemu Windows przekazywane są tokeny. W tym modelu użytkownik łączy się z bazą z poziomu systemu Windows i nie musi ponownie podawać hasła, co zapewnia większy komfort pracy.

Loginy i użytkownicy

Login w systemie SQL Server określa tożsamość skonfigurowaną w danym egzemplarzu tego systemu. Login służy do uwierzytelniania klienta w systemie SQL Server. Gdy administrator tworzy login, łączy go z danymi uwierzytelniającymi — hasłem systemu SQL Server lub tożsamością z systemu Windows. Login można powiązać z uprawnieniami z poziomu egzemplarza systemu SQL Server, jednak większość uprawnień dostępu do danych jest łączona z użytkownikami baz z tego systemu. Dane użytkowników są zapisywane w każdej bazie egzemplarza, a ich tożsamość jest określana na podstawie powiązanego loginu. Każdy login można powiązać z jednym lub wieloma użytkownikami bazy danych.

Ponieważ użytkownik bazy danych jest powiązany z loginem przechowywanym w bazie master egzemplarza, po przeniesieniu bazy do innego egzemplarza systemu SQL Server trzeba utworzyć w nim odpowiednie loginy, aby umożliwić użytkownikom dostęp do bazy. Prowadziło to do problemów przy tworzeniu kopii lustrzanych i przenoszeniu dzienników. Gdy twórcy systemu SQL Server pracowali nad grupami dostępności AlwaysOn (ang. *AlwaysOn Availability Groups*) w wersji SQL Server 2012, zauważyli, że synchronizowanie loginów w wielu pomocniczych bazach danych może sprawiać spore trudności.

Dlatego w systemie SQL Server 2012 wprowadzono *niezależnych użytkowników* (ang. *contained user*) obsługiwanych w niezależnych bazach danych. Niezależni użytkownicy są przypisani do jednej bazy i nie są powiązani z loginem z poziomu egzemplarza. To oznacza, że do niezależnego użytkownika nie można przypisywać uprawnień z poziomu egzemplarza (np. możliwości łączenia się z różnymi bazami danych). Aby utworzyć niezależnego użytkownika, należy wywołać instrukcję `CREATE USER` i podać parametr `PASSWORD` zamiast parametru `FROM LOGIN`. Ilustruje to poniższy fragment kodu:

```
CREATE USER MyContainedUser WITH PASSWORD='MySecurePassword'  
GO
```

W bazie danych, w której utworzono niezależnego użytkownika, uprawnienia do obiektów i łańcuchy uprawnień działają dokładnie tak samo jak dla standardowych użytkowników.

Niezależnych użytkowników można tworzyć na podstawie kont lokalnych i kont domeny systemu Windows, a także jako użytkowników systemu SQL Server. *Niezależny użytkownik systemu Windows* to konto systemu Windows, z którym na poziomie serwera nie jest powiązany login. Dla niezależnych użytkowników systemu SQL Server (inaczej niż dla tradycyjnych loginów z tego systemu) nie można ustawić zgodności z zasadami domeny.

Autoryzacja dostępu

Odpowiednie zabezpieczenia na poziomie obiektów w bazach danych to podstawa ochrony danych przed napastnikami w egzemplarzach systemu SQL Server. Zabezpieczenia na poziomie obiektów dotyczą m.in. obiektów egzemplarza (np. grup dostępności), możliwości wyświetlania obiektów stanu serwera i zabezpieczania konkretnych obiektów w bazach danych.

Uprawnienia można przyznawać na poziomie serwera, na poziomie bazy danych i do konkretnych obiektów. Ponadto uprawnienia można łączyć ze sobą w łańcuch, co upraszcza ich stosowanie zarówno w konkretnych bazach, jak i w ich grupach.

Do modyfikowania uprawnień w systemie SQL Server służą trzy poniższe instrukcje.

- GRANT — pozwala przyznawać uprawnienia.
- DENY — służy do blokowania dostępu.
- REVOKE — usuwa uprawnienia ustawione za pomocą instrukcji GRANT i DENY.

Gdy przyznajesz uprawnienia w systemie SQL Server, pamiętaj, że instrukcja DENY zawsze zastępuje polecenie GRANT. Jeśli użytkownik jest przypisany do trzech ról i dwie z nich mają uprawnienia do wykonywania kwerend dla danej tabeli, natomiast w trzeciej ta operacja jest zabroniona, użytkownik nie będzie mógł uruchamiać takich kwerend.

UWAGA Zwizualizowanie różnych opisanych w tym rozdziale uprawnień do obiektów może być trudne na podstawie samej lektury opisów. Microsoft udostępnia graficzny diagram, który ułatwia to zadanie. Diagram znajdziesz na stronie http://social.technet.microsoft.com/wiki/cfs-filesystemfile.ashx/___key/communityserver-wikis-components-files/00-00-00-00-05/2727.Permissions_5F00_Poster_5F00_2014.bmp.

To samo dotyczy przyznanych zbiorów uprawnień do bardziej ogólnych obiektów. Jeśli np. użytkownik otrzymał uprawnienia do wykonywania instrukcji SELECT dla danego schematu, ale odmówiono mu uprawnień do wykonywania kwerend dla konkretnej tabeli, nie będzie mógł wykonywać takich instrukcji dla tej tabeli. Jeżeli ponadto użytkownik otrzymał uprawnienia do wykonywania kwerend dla danej tabeli, ale odmówiono mu uprawnień do przeprowadzania tej operacji dla schematu, nie będzie mógł jej wykonywać dla żadnej tabeli z tego schematu (niezależnie od uprawnień przyznanych dla poszczególnych tabel).

Przy przyznawaniu uprawnień użytkownikom można też wykorzystać składnię z członem WITH GRANT. Dodanie członu WITH GRANT na końcu instrukcji GRANT (tak jak w poniższym fragmencie kodu) powoduje, że użytkownik, który otrzymał dane uprawnienia, może je także przyznawać innym użytkownikom określonej bazy:

```
GRANT SELECT, INSERT ON dbo.Users TO MyUser WITH GRANT
```

Uprawnienia na serwerze

Na poziomie egzemplarza można przyznać dziesiątki uprawnień. Dotyczą one łączenia się z różnymi punktami końcowymi egzemplarza systemu SQL Server, zarządzania takimi punktami, zarządzania loginami egzemplarza. Dotyczą ustawień z poziomu egzemplarza, grup dostępności AlwaysOn, a także definiowanych przez użytkownika ról serwera wprowadzonych w wersji SQL Server 2012.

Najważniejsza różnica między uprawnieniami z poziomu egzemplarza i uprawnieniami z poziomu bazy danych polega na tym, że uprawnienia z poziomu egzemplarza są przyznawane bezpośrednio dla loginu, natomiast uprawnienia z poziomu baz danych przyznaje się użytkownikom związanym z loginami.

UWAGA Pojęcia „login” i „użytkownik” często stosuje się zamiennie. Jednak w kontekście systemu Microsoft SQL Server reprezentują one dwie zupełnie inne rzeczy. Loginy służą do logowania się do egzemplarza bazy danych, natomiast w ramach baz systemu SQL Server użytkownicy są wiązani z loginami.

Teraz przyjrzyjmy się różnym uprawnieniom do punktów końcowych i innych obiektów egzemplarza systemu SQL Server.

Punkty końcowe

Dostępnych jest pięć wymienionych poniżej uprawnień dla punktów końcowych z egzemplarza systemu SQL Server.

- **Alter.** Uprawnienie umożliwia loginom wprowadzanie zmian w konfiguracji punktu końcowego.
- **Connect.** Uprawnienie pozwala użytkownikowi nawiązywanie połączenia z danym punktem końcowym. Domyślnie wszystkie loginy mogą łączyć się z domyślnymi punktami końcowymi.
- **Control.** Uprawnienie daje wszystkie cztery pozostałe uprawnienia.
- **Take Ownership.** Uprawnienie pozwala, by login stał się właścicielem danego punktu końcowego.
- **View Definition.** Uprawnienie umożliwia loginowi wyświetlanie konfiguracji danego punktu końcowego (ale bez możliwości modyfikowania tego punktu).

Loginy

Cztery wymienione poniżej uprawnienia służą do zarządzania loginami w egzemplarzach systemu SQL Server.

- **Alter.** Uprawnienie umożliwia posiadającemu je loginowi wprowadzanie zmian w innym loginie, którego uprawnienie dotyczy. Załóżmy, że istnieją dwa loginy — login1 i login2. Za pomocą tego uprawnienia login1 może modyfikować login2. W ramach modyfikacji loginu można zmienić: hasło, domyślną bazę danych, domyślny język i inne ustawienia docelowego loginu.
- **Control.** Uprawnienie daje posiadającemu je użytkownikowi wszystkie trzy pozostałe uprawnienia do docelowego loginu. Załóżmy, że istnieją dwa loginy — login1 i login2. Przy użyciu tego uprawnienia login1 może kontrolować login2.
- **Impersonate.** Uprawnienie pozwala posiadającemu je użytkownikowi na używanie składni EXECUTE AS z podanym docelowym loginem. Umożliwia to wykonywanie kodu jako docelowy login.
- **View Definition.** Uprawnienie umożliwia posiadającemu je użytkownikowi wyświetlanie konfiguracji docelowego loginu.

Ustawienia z poziomu egzemplarza

W tabeli 8.1 przedstawiono uprawnienia (z opisami), które można przyznać wybranemu loginowi.

Tabela 8.1. Uprawnienia z poziomu egzemplarza i ich znaczenie

Nazwa uprawnienia	Definicja uprawnienia
Administrator Bulk Options	Umożliwia użytkownikowi masowe wstawianie danych do egzemplarza systemu SQL Server za pomocą instrukcji BULK INSERT, uruchamianej z poziomu wiersza poleceń aplikacji bcp i operacji OPENROWSET (BULK).
Alter Any Availability Group	Umożliwia użytkownikowi zmianę lub przełączenie się do dowolnej grupy dostępności AlwaysOn, a także zapewnia uprawnienie Create Availability Group.
Alter Any Connection	Przyznaje użytkownikowi prawo do zamknięcia połączenia dowolnej osoby.
Alter Any Credential	Przyznaje użytkownikowi prawo zmiany dowolnych danych uwierzytelniających w egzemplarzu bazy danych.
Alter Any Database	Daje użytkownikowi możliwość zmiany opcji dowolnej bazy danych z egzemplarza systemu, a ponadto zapewnia uprawnienie Create Any Database.
Alter Any Endpoint	Pozwala użytkownikowi zmieniać dowolne punkty końcowe utworzone w danym egzemplarzu systemu SQL Server, a ponadto zapewnia uprawnienie Create Any Endpoint.
Alter Any Event Notification	Pozwala użytkownikowi zmieniać dowolne powiadomienia o zdarzeniach utworzone w danym egzemplarzu systemu SQL Server, a ponadto zapewnia uprawnienie Create Trace Event Notification.
Alter Any Linked Server	Umożliwia użytkownikowi zmianę dowolnego serwera połączonego w danym egzemplarzu systemu SQL Server.
Alter Any Login	Daje użytkownikowi uprawnienia do zmiany dowolnego loginu w egzemplarzu systemu.
Alter Any Server Audit	Daje użytkownikowi uprawnienia do zmiany dowolnej specyfikacji inspekcji serwera.
Alter Any Server Role	Umożliwia zmianę zdefiniowanych przez użytkownika ról serwera w danym egzemplarzu systemu SQL Server.
Alter Resources	Umożliwia użytkownikowi modyfikowanie zasobów systemowych.
Alter Server State	Daje użytkownikowi prawo do zmiany stanu serwera, a ponadto zapewnia uprawnienie View Server State.
Alter Settings	Umożliwia użytkownikowi zmianę ustawień z poziomu egzemplarza.
Alter Trace	Umożliwia zmianę śladów z profilera i generowanych po stronie serwera śladów innych użytkowników.
Authenticate Server	Daje użytkownikowi uprawnienia do uwierzytelnienia się w danym egzemplarzu systemu SQL Server.
Connect SQL	Daje użytkownikowi uprawnienia do łączenia się z danym egzemplarzem systemu SQL Server.

Tabela 8.1. Uprawnienia z poziomu egzemplarza i ich znaczenie — ciąg dalszy

Nazwa uprawnienia	Definicja uprawnienia
Control Server	Daje zestaw uprawnień z poziomu egzemplarza: Administrator bulk options, Alter Any Availability Group, Alter Any Connection, Alter Any Credential, Alter Any Database, Alter Any Endpoint, Alter Any Event Notification, Alter Any Linked Server, Alter Any Login, Alter Any Server Audit, Alter Any Server Role, Alter Resources, Alter Server State, Alter Settings, Alter Trace, Authenticate Server, Connect SQL, External Access Assembly, Shutdown, Unsafe Assembly i View Any Definition.
Create Any Database	Umożliwia użytkownikowi tworzenie nowych baz danych i przywracanie ich na podstawie kopii zapasowej.
Create Availability Group	Umożliwia użytkownikowi tworzenie nowych grup dostępności AlwaysOn.
Create DDL Event Notification	Umożliwia użytkownikowi tworzenie wyzwalaczy DDL.
Create Endpoint	Umożliwia użytkownikowi tworzenie punktów końcowych systemu SQL Server.
Create Server Role	Umożliwia tworzenie zdefiniowanych przez użytkownika ról serwera.
Create Trace Event Notification	Umożliwia użytkownikowi tworzenie powiadomień o zdarzeniach śledzenia.
External Access Assembly	Umożliwia użytkownikowi tworzenie komponentów z ustawieniem EXTERNAL ACCESS.
Shutdown	Umożliwia użytkownikowi zamykanie egzemplarza systemu SQL Server za pomocą instrukcji SHUTDOWN języka T-SQL.
Unsafe Assembly	Umożliwia użytkownikowi tworzenie komponentów z ustawieniem UNSAFE.
View Any Database	Umożliwia użytkownikowi wyświetlenie definicji dowolnej bazy z danego egzemplarza systemu SQL Server.
View Any Definition	Umożliwia użytkownikowi wyświetlenie definicji dowolnego obiektu z danego egzemplarza systemu SQL Server, a ponadto daje uprawnienie View Any Database.
View Server State	Umożliwia użytkownikowi wyświetlenie obiektów z informacjami o stanie serwera. Obiekty te to widoki DMV (ang. <i>Dynamic Management View</i>) i funkcje systemu SQL Server.

OSTRZEŻENIE Uprawnienie Control Server można przyznać użytkownikom, którzy potrzebują dużych możliwości, a jednocześnie nie wymagają kompletnych uprawnień administratora związanych z przynależnością do roli sysadmin serwera. Pamiętaj, że nie należy nadużywać tego uprawnienia, ponieważ daje ono użytkownikowi duże prawa w danym egzemplarzu systemu SQL Server.

Nowe ustawienia w systemie SQL Server 2014

W systemie SQL Server 2014 wprowadzono trzy nowe uprawnienia z poziomu egzemplarza i jedno nowe uprawnienie z poziomu bazy danych. Są one przedstawione w tabeli 8.2.

Tabela 8.2. Nowe uprawnienia w systemie SQL Server 2014

Nazwa uprawnienia	Definicja uprawnienia
Connect Any Database	Uprawnienie umożliwia loginowi łączenie się z wszystkimi bazami z danego egzemplarza. Odmowa tego uprawnienia sprawia, że użytkownik nie może połączyć się z żadną bazą.
Impersonate Any Login	Uprawnienie umożliwia loginowi podawanie się za dowolny inny login zdefiniowany na danym serwerze. Jest to niebezpieczne uprawnienie. Należy je stosować ostrożnie i to głównie dla testów oraz podczas debugowania. Odmowa tego uprawnienia uniemożliwia użytkownikowi podawanie się za inne osoby. Bywa to przydatne, jednak należy to sprawdzić — czasem podawanie się za innych użytkowników jest niezbędne przy wykonywaniu dozwolonych operacji.
Select All User Securables	Uprawnienie umożliwia loginowi odczyt dowolnych danych z całego egzemplarza. Jest to przydatne dla audytorów, którzy potrzebują dostępu do wszystkich danych, ale nie powinni ich modyfikować. To uprawnienie jest ciekawsze w kontekście jego odmowy (instrukcja DENY). Jeśli dla loginu zastosowano polecenie DENY Select All User Securables, login nie będzie mógł odczytać żadnych danych z egzemplarza. To oznacza, że można utworzyć konto administratora z uprawnieniami do tworzenia kopii zapasowych baz danych, odtwarzania indeksów itd., ale bez prawa do odczytu danych. Audytorzy zabezpieczeń często domagali się stosowania takiego podejścia, a do tej pory bardzo trudno było uzyskać ten efekt.
Alter Any Database Event Session	Uprawnienie umożliwia zarządzanie sesjami zdarzeń rozszerzonych.

Grupy dostępności

Dla grup dostępności istnieją cztery poniższe uprawnienia. Można je przyznać zdefiniowanym przez użytkownika rolem serwera.

- **Alter.** Umożliwia użytkownikowi posiadającemu to uprawnienie modyfikowanie grup dostępności AlwaysOn.
- **Control.** Zapewnia użytkownikowi wszystkie trzy pozostałe uprawnienia.
- **Take Ownership.** Uprawnienie pozwala posiadającemu je użytkownikowi stać się właścicielem danej grupy dostępności.
- **View Definition.** Uprawnienie umożliwia posiadającemu je użytkownikowi wyświetlanie definicji danej grupy dostępności.

Role serwera zdefiniowane przez użytkownika

W systemie SQL Server 2012 wprowadzono role serwera zdefiniowane przez użytkownika. Przypominają one wbudowane (czyli stałe) role serwera, ale są tworzone przez administratora systemu SQL Server, a nie przez Microsoft. Zdefiniowane przez użytkownika role serwera mogą

być przypisane do dowolnej innej roli (wbudowanej lub zdefiniowanej przez użytkownika). Dowolne uprawnienie z poziomu serwera (opisano je wcześniej w tym rozdziale), które można przyznać loginowi, można też przypisać do zdefiniowanej przez użytkownika roli serwera. Takie role pozwalają zdefiniować rolę administratora z tylko tymi uprawnieniami, których dana osoba potrzebuje do wykonywania swojej pracy. Można zatem łatwiej zastosować zasadę najniższych niezbędnych uprawnień.

Do loginu można przypisać cztery poniższe uprawnienia dotyczące roli serwera zdefiniowanej przez użytkownika.

- **Alter.** Umożliwia użytkownikowi posiadającemu to uprawnienie modyfikowanie roli serwera zdefiniowanej przez użytkownika. Możliwe jest m.in. przypisywanie innych loginów do wbudowanych ról serwera.
- **Control.** Zapewnia użytkownikowi wszystkie trzy pozostałe uprawnienia.
- **Take Ownership.** Uprawnienie pozwala posiadającemu je loginowi stać się właścicielem danej roli serwera zdefiniowanej przez użytkownika.
- **View Definition.** Uprawnienie umożliwia posiadającemu je loginowi wyświetlanie danej roli serwera zdefiniowanej przez użytkownika (nie pozwala przy tym na modyfikowanie roli).

Wbudowane role serwera

W systemie SQL Server dostępnych jest dziewięć ról serwera wbudowanych przez Microsoft. Nie można ich modyfikować. Osiem tych ról istnieje od wersji SQL Server 7 lub wcześniejszych edycji tego systemu. Dziewiątą rolę wprowadzono w wersji SQL Server 2005. Ta nowa rola to *bulkadmin*. Daje osobom przypisanym do niej prawo do masowego wstawiania danych do bazy. We wcześniejszych wersjach systemu SQL Server masowe wczytywanie danych do bazy wymagało przynależności do najbardziej rozbudowanej z wbudowanych ról serwera, *sysadmin*, która umożliwia wykonywanie dowolnych operacji na bazach danych. Pozostałe wbudowane role zapewniają przypisanym użytkownikom różne uprawnienia. Omówienie tych ról znajdziesz poniżej.

- Wbudowana rola serwera *dbcreator* umożliwia użytkownikowi tworzenie baz danych.
- Wbudowana rola serwera *diskadmin* umożliwia użytkownikowi zarządzanie fizycznymi plikami bazy danych.
- Wbudowana rola serwera *setupadmin* umożliwia użytkownikowi dodawanie i usuwanie serwerów połączonych.
- Wbudowana rola serwera *processadmin* umożliwia zamykanie procesów innych użytkowników w ramach egzemplarza systemu SQL Server.
- Wbudowana rola serwera *securityadmin* umożliwia wywoływanie instrukcji GRANT, DENY i REVOKE dotyczących wszystkich uprawnień z poziomu serwera, a także uprawnień z poziomu baz danych, do których dana osoba ma potrzebne prawa.
- Wbudowana rola serwera *serveradmin* umożliwia zmianę dowolnych opcji konfiguracyjnych z poziomu serwera, a także wywoływanie polecenia SHUTDOWN w celu zamknięcia danego egzemplarza systemu SQL Server.
- Wbudowana rola serwera *public* nie daje żadnych uprawnień. Wszystkie loginy w egzemplarzu systemu mają przypisaną tę rolę.
- Rola *sysadmin* (nie zawsze jest ona uznawana za wbudowaną rolę serwera) zapewnia wszystkie uprawnienia do danego egzemplarza. System nie sprawdza uprawnień użytkowników przypisanych do tej roli.

Zabezpieczane obiekty baz danych

W każdej bazie systemu SQL Server znajdują się różne obiekty. Każdy z nich jest powiązany z określonymi uprawnieniami. Uprawnienia zapewniają użytkownikom prawo do korzystania z poszczególnych obiektów. Dzięki temu użytkownicy mogą wykonywać potrzebne operacje. Najlepiej przyznawać minimalne uprawnienia niezbędne do ukończenia zadań. Użytkownicy nie powinni mieć uprawnień do obiektów i danych z bazy, jeśli ich nie potrzebują. Korzyścią ze stosowania tego podejścia jest to, że napastnik, który włamie się do bazy, nie uzyska dostępu do bardziej chronionych danych.

Uprawnienia do bazy danych

Uprawnienia można przyznać do samej bazy danych. Niektóre z nich dotyczą wyłącznie baz, natomiast inne obowiązują także dla obiektów z bazy (np. tabel, widoków i procedur składowanych).

Uprawnienia do tabel i widoków

Istnieje dziesięć uprawnień do tabel i widoków, które można przyznać konkretnym użytkownikom lub rolom zdefiniowanym przez użytkownika. Listę tych uprawnień zamieszczono w tabeli 8.3.

Tabela 8.3. Uprawnienia do tabel i widoków

Uprawnienie	Definicja
Alter	Umożliwia użytkownikowi zmianę schematu obiektu.
Control	Przyznaje użytkownikowi wszystkie pozostałe uprawnienia do obiektu.
Delete	Umożliwia użytkownikowi usunięcie danych z obiektu.
Insert	Umożliwia użytkownikowi wstawianie danych do tabeli.
References	Umożliwia użytkownikowi tworzenie kluczy obcych w tabeli. To uprawnienie nie dotyczy widoków.
Select	Umożliwia użytkownikowi pobieranie danych z obiektu.
Take Ownership	Umożliwia użytkownikowi zmianę właściciela obiektu.
Update	Umożliwia użytkownikowi modyfikowanie danych w tabeli.
View Change Tracking	Umożliwia użytkownikowi wyświetlanie informacji z mechanizmu śledzenia zmian dla danego obiektu.
View Definition	Umożliwia użytkownikowi wyświetlenie definicji obiektu.

Procedury składowane i funkcje

Dla procedur składowanych, funkcji i większości innych obiektów z systemu SQL Server istnieje tylko pięć uprawnień, które można przypisywać użytkownikom i rolom. Uprawnienia zostały wymienione w tabeli 8.4.

Tabela 8.4. Uprawnienia dotyczące procedur składowanych, funkcji i większości innych obiektów z baz danych

Upewnienie	Definicja
Alter	Umożliwia użytkownikowi zmianę schematu obiektu bazy danych, do którego przyznano to upewnienie.
Control	Przyznaje użytkownikowi wszystkie pozostałe upewnienia do obiektu.
Execute	Umożliwia użytkownikowi wykonywanie obiektu.
Take Ownership	Umożliwia użytkownikowi zmianę właściciela obiektu.
View Definition	Umożliwia użytkownikowi wyświetlenie schematu obiektu (nie daje możliwości modyfikowania danego obiektu).

Łańcuchy upewnien

Upewnienia w bazach danych są łączone w łańcuch, gdy użytkownik ma upewnienia do korzystania z obiektu, który wywołuje obiekt niższego poziomu. W poniższym przykładowym fragmencie kodu pokazano, że gdy dostępne są tabela i procedura składowana, a użytkownik otrzymał upewnienia do wykonywania procedury składowanej, łańcuch upewnien pozwala użytkownikowi pobierać dane z tej tabeli, ale tylko w kontekście procedury składowanej. Kwerendy uruchamiane spoza procedury składowanej nie zadziałają, chyba że użytkownik bezpośrednio otrzyma upewnienia do tabeli.

```
CREATE USER SampleUser WITHOUT LOGIN
GO
CREATE TABLE dbo.Users
(UserId INT IDENTITY (1,1) PRIMARY KEY,
UserName varchar(100),
Password varchar(100))
go
CREATE PROCEDURE dbo.SignIn
    @UserName varchar(100),
    @Password varchar(100)
AS
SELECT UserId
FROM Users
WHERE UserName = @UserName
and Password = @Password
GO
GRANT EXEC ON dbo.SignIn to SampleUser
GO
```

Łańcuch upewnien działa tylko w kontekście wykonywania obiektu nadrzędnego. Jest to wymyślny sposób na napisanie, że łańcuch upewnien nie działa w dynamicznym SQL-u. Gdy używasz dynamicznego SQL-a, użytkownik musi posiadać upewnienia do konkretnych obiektów wywoływanych w kodzie.

Łańcuch upewnien jest stosowany we wszystkich natywnych obiektach systemu SQL Server, w tym w procedurach składowanych, funkcjach skalarnych, funkcjach tabelowych i widokach. Pozwala to korzystać z innych procedur składowanych, funkcji skalarnych, funkcji tabelowych, widoków, tabel i obiektów Service Brokera. Obsługa łańcucha upewnien jest domyślnie włączona i nie można jej wyłączyć. Łańcuch upewnien umożliwia stosowanie się do zalecanej praktyki przyznawania minimalnych potrzebnych upewnien, ponieważ sprawia, że wystarczy

zapewnić użytkownikowi uprawnienia do wykonywania procedur składowanych. Uprawnienia do podstawowych obiektów nie są wtedy wymagane.

Łańcuchy uprawnień obowiązujące między bazami

Łańcuchy uprawnień obowiązujące między bazami to funkcja systemu SQL Server wprowadzona w wersji SQL Server 2000 SP3a. Technika ta jest rozwinięciem tradycyjnych łańcuchów uprawnień, umożliwiającym obejmowanie takimi łańcuchami różnych baz danych. Bez tej metody dostęp do obiektów z innej bazy przy użyciu trzyczłonowej nazwy obiektu wymaga, aby użytkownik miał uprawnienia do danego obiektu. Przyjrzyj się skryptowi z poniższego fragmentu kodu. Aby procedura składowana w tej postaci zadziałała bez omawianej techniki, użytkownik musi mieć uprawnienia do wykonywania instrukcji SELECT.

```
USE master
GO
CREATE DATABASE Sample1
GO
CREATE DATABASE Sample2
GO
USE Sample1
GO
CREATE TABLE dbo.Users
(UserId INT IDENTITY (1,1) PRIMARY KEY,
UserName varchar(100),
Password varchar(100))
go
USE Sample2
GO
CREATE TABLE dbo.Users
(UserId INT IDENTITY (1,1) PRIMARY KEY,
UserName varchar(100),
Password varchar(100))
go
CREATE PROCEDURE dbo.VerifyUsers
AS
SELECT *
FROM dbo.Users b
WHERE NOT EXISTS (SELECT * FROM Sample1.dbo.Users a WHERE a.UserId = b.UserId)
GO
```

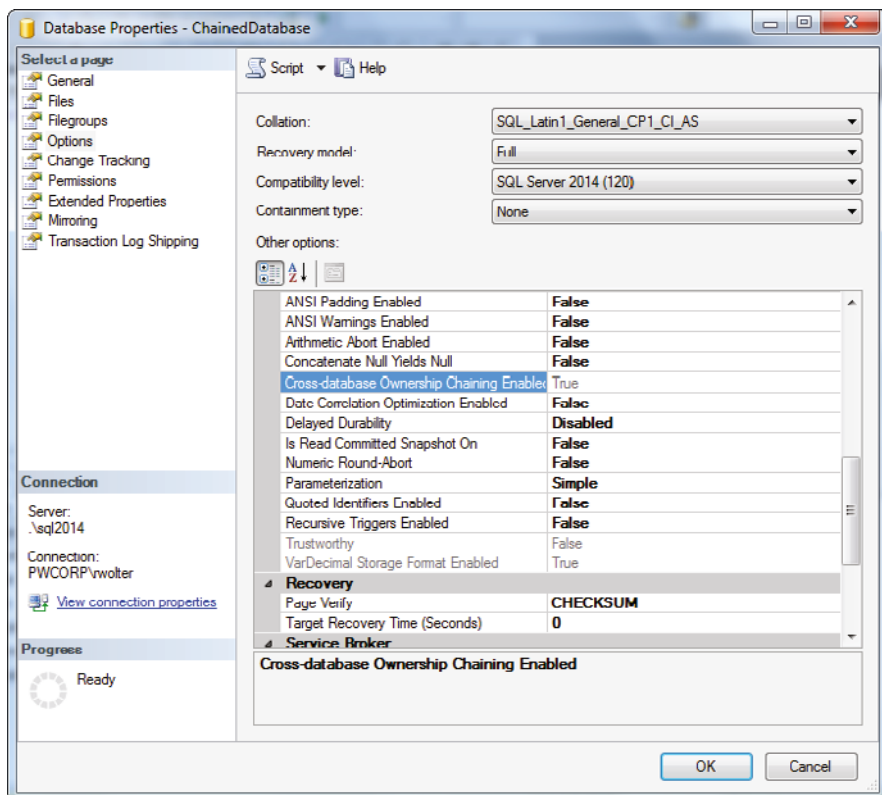
Łańcuchy uprawnień obowiązujące między bazami to ustawienie z poziomu bazy danych. Domyślnie we wszystkich bazach jest ono wyłączone. Istnieje kilka sposobów na sprawdzenie, czy w danej bazie ta technika jest włączona. Najłatwiej pobrać dane z widoku katalogu sys.database. Kolumna is_db_chaining ma wartość 0, gdy łańcuchy uprawnień obowiązujące między bazami nie są obsługiwane, a 1, jeśli ta metoda jest włączona (patrz rysunek 8.1).

	name	is_db_chaining_on
1	master	1
2	tempdb	1
3	model	0
4	msdb	1
5	SampleDatabase	0
6	ChainedDatabase	1
7	QueueReceiver	0

Rysunek 8.1. Sprawdzanie, czy w poszczególnych bazach włączone są łańcuchy uprawnień obowiązujące między bazami

Obsługę łańcuchów uprawnień obowiązujących między bazami można sprawdzić także w oknie *Properties* bazy danych. W tym celu należy połączyć się z daną bazą w oknie *Object Explorer*. Wykonaj zatem następujące kroki.

1. Przejdź do węzła *Databases* i zaznacz odpowiednią bazę.
2. Kliknij bazę prawym przyciskiem myszy i wybierz z menu kontekstowego opcję *Properties*.
3. Otwórz stronę *Options* i przewiń ją do przedstawionej na rysunku 8.2 sekcji *Other Options*.



Rysunek 8.2. Okno z właściwościami bazy danych

4. W sekcji *Miscellaneous* widoczne (choć niedostępne) jest pole *Cross-database Ownership Chaining Enabled*. Możesz sprawdzić jego wartość na ekranie przedstawionym na rysunku 8.2. Jednak aby zmienić to ustawienie, musisz użyć instrukcji *ALTER DATABASE* języka T-SQL.

Wcześniej wspomniano, że do włączania łańcuchów uprawnień obowiązujących między bazami służy instrukcja *ALTER DATABASE* języka T-SQL. Należy przy jej użyciu włączyć opcję *DB_CHAINING*, co pokazano w poniższym fragmencie kodu. Aby przedstawiona wcześniej procedura składowana zadziałała, instrukcję *ALTER DATABASE* trzeba wywołać dla obu używanych baz danych.

```
ALTER DATABASE Sample1 SET DB_CHAINING ON
GO
ALTER DATABASE Sample2 SET DB_CHAINING ON
GO
```

Trzeba przyznać jeszcze jedno uprawnienie, aby łańcuchy uprawnień działały poprawnie między bazami. Dana osoba musi zostać dodana także jako użytkownik drugiej bazy. Nie musi mieć do niej żadnych dodatkowych uprawnień, natomiast trzeba ją ustawić jako użytkownika drugiej bazy. Można w tym celu albo utworzyć użytkownika powiązanego z tym samym loginem, z którym powiązany jest użytkownik z pierwszej bazy, albo włączyć konto guest w drugiej bazie. Nie zaleca się włączania konta guest, ponieważ powoduje to przyznanie użytkownikowi niepotrzebnych mu uprawnień do danej bazy.

Choć łańcuchy uprawnień obowiązujące między bazami to wygodna technika pozwalająca wykonywać kwerendy między powiązаныmi bazami, należy ją stosować ostrożnie, ponieważ może prowadzić do nieoczekiwanych konsekwencji. Załóżmy, że złośliwy właściciel niewielkiej bazy danych korzysta z tego samego egzemplarza systemu, w którym działa baza z danymi kart kredytowych. Jako właściciel bazy danych konkretna osoba może tworzyć użytkowników powiązanych z loginami używanymi przez użytkowników bazy kart kredytowych. Gdy łańcuchy uprawnień obowiązujące między bazami danych są włączone, złośliwy właściciel innej bazy może uzyskać dostęp do obiektów, które nie powinny być dla niego widoczne.

Zabezpieczenia na poziomie wierszy

Użytkownicy od dawna domagali się wprowadzenia zabezpieczeń na poziomie wierszy, ponieważ w systemie Oracle działają wirtualne prywatne bazy danych, umożliwiające administratorom określenie, do których wierszy użytkownik powinien mieć dostęp.

Aby uzyskać pożądany efekt, najłatwiej utworzyć widok dla każdego użytkownika, który potrzebuje dostępu do danych, lub przygotować jeden widok dla każdej grupy wymagającej dostępu do określonych wierszy z tabeli bazy danych. Użytkownikowi należy następnie przyznać uprawnienia tylko do widoku, ale już nie do tabeli. Użytkownik może wtedy wyświetlać tylko te wiersze, które spełniają warunek z klauzuli WHERE widoku.

Inna technika polega na zaprojektowaniu zabezpieczeń na poziomie wierszy w schemacie bazy danych. Jeśli np. istnieje tabela działu firmy i wszyscy menedżerowie powinni mieć możliwość pobierania z niej danych, ale tylko na temat swoich bezpośrednich podwładnych, można zbudować w systemie jeden widok używany przez wszystkich menedżerów. W tym celu należy zastosować funkcje systemowe, takie jak `suser_sname()` lub `current_user()`. Na listingu 8.1. przedstawiono, jak wykorzystać funkcję `suser_sname()` (ten kod znajdziesz w pliku *Funkcja_suser_sname.sql*).

Listing 8.1. Używanie funkcji `suser_sname()` (plik *Funkcja_suser_sname.sql*)

```
CREATE TABLE dbo.Employee
(EmployeeId INT IDENTITY(1,1) PRIMARY KEY,
LastName varchar(100),
FirstName varchar(100),
EmailAddress varchar(255),
ManagerId INT,
Username varchar(100))
GO
INSERT INTO dbo.Employee
(LastName, FirstName, EmailAddress, ManagerId, UserName)
VALUES
('Smith', 'John', 'jsmith@contonso.com', 0, 'CONTOSO\jsmith'),
('Gates', 'Fred', 'fgates@contonso.com', 1, 'CONTOSO\fgates'),
```

```

('Jones', 'Bob', 'bjones@contonso.com', 1, 'CONTOSO\bjones'),
('Erickson', 'Paula', 'perickson@contonso.com', 1, 'CONTOSO\perickson')
GO
CREATE VIEW dbo.EmployeeView
AS
SELECT *
FROM dbo.Employee
WHERE ManagerId = (SELECT EmployeeId FROM Employee WHERE UserName = suser_name())
GO

```

UWAGA Microsoft udostępnił doskonały dokument opisujący szczegółowo zabezpieczenia na poziomie wierszy. Więcej informacji na temat tego dokumentu znajdziesz na blogu Public Sector w witrynie MSDN (<http://blogs.msdn.com/b/publicsector/archive/2011/08/23/row-level-security-for-sql-server-2008.aspx>). Microsoft opracował też mechanizmy zabezpieczeń z poziomu wierszy i komórek, udostępnione na serwisie Codeplex (<http://sqlserver1st.codeplex.com/>). Narzędzia zostały utworzone przez zespół inżynierów odpowiedzialny za system SQL Server, dlatego kod jest dobrze konserwowany i niezawodny.

Podsumowanie

Trzeba właściwie zaimplementować zabezpieczenia bazy danych, ponieważ w przeciwnym razie wszystkie informacje będą dostępne dla niepożądanych osób. To sprawi, że baza będzie podatna na wewnętrzne i zewnętrzne zagrożenia. Bezpieczeństwo należy zapewnić nie tylko bazie danych i egzemplarzowi systemu.

Musisz wprowadzić różne poziomy zabezpieczeń w bazie i poza nią. Powinieneś odpowiednio zabezpieczyć sieć, system operacyjny Windows, w którym działają bazy, egzemplarz systemu SQL Server, uprawnienia do baz danych, a także aplikacje (jest to niezbędne, aby mieć pewność, że ataki polegające na wstrzyknięciu kodu w SQL-u nie zakończą się powodzeniem).

W każdej wersji systemu Microsoft SQL Server dostępne zabezpieczenia się zmieniają, dlatego administratorzy muszą się do tego dostosować. Zdefiniowane przez użytkownika role serwera to model zabezpieczeń dający dużą swobodę.

W rozdziale 9. zamieszczamy omówienie technologii In-Memory OLTP. Jest to nowość wprowadzona w wersji SQL Server 2014. Dowiesz się, jak używać tabel przechowywanych w pamięci i natywnie przechowywanych procedur składowanych, aby zoptymalizować najważniejsze tabele OLTP i znacznie zwiększyć wydajność aplikacji. W rozdziale tym opisujemy, jak ustalić, dla których tabel warto zastosować technologię In-Memory OLTP, a także jak konfigurować wybrane tabele i jak nimi zarządzać.

Technologia In-Memory OLTP

ZAWARTOŚĆ ROZDZIAŁU:

- Stosowanie i implementowanie rozwiązań w technologii In-Memory OLTP.
- Tworzenie natywnie kompilowanych procedur składowanych.
- Omówienie narzędzia ARM.

Dostępny do pobrania kod dla tego rozdziału znajdziesz w zakładce *Szczegóły* na poświęconej tej książce stronie w witrynie wydawnictwa Helion — <http://helion.pl>. Po rozpakowaniu pobranego pliku kod dla rozdziału 9. znajdziesz w katalogu o odpowiedniej nazwie.

Przetwarzanie transakcji OLTP (ang. *Online Transaction Processing*) w pamięci (czyli technologia In-Memory OLTP) to jedna z najważniejszych zmian wprowadzonych w wersji SQL Server 2014. Aby w pełni docenić tę technologię, musisz zrozumieć, czym jest, do czego ma służyć i jak wypada w porównaniu z innymi mechanizmami przetwarzania danych w pamięci, oferowanymi przez konkurentów Microsoftu. Całościowy obraz tej technologii pomoże ustalić, czy i kiedy powinieneś z niej korzystać.

Z tego rozdziału dowiesz się, jak tworzyć tabele dla technologii In-Memory OLTP, jak opracowywać natywnie kompilowane procedury składowane, jak technologia In-Memory OLTP jest zintegrowana z tradycyjnym stosem mechanizmów systemu SQL Server, a także jak korzystać z narzędzia ARM (ang. *Analyze, Report, Migrate*).

Stosowanie i implementowanie technologii In-Memory OLTP

In-Memory OLTP to funkcja z poziomu systemów korporacyjnych zapewniająca i firmom różnej wielkości wydajność dostępną wcześniej tylko dla organizacji, które mogły wydać setki tysięcy dolarów na sprzęt. Technologia działa tylko w edycjach SQL Server 2014 Developer,

Evaluation i Enterprise. Aby wykonać opisane dalej ćwiczenia, musisz korzystać z jednej z tych edycji. Technologię In-Memory OLTP można też nazwać *zoptymalizowanymi tabelami działającymi w pamięci*.

Zoptymalizowane tabele działające w pamięci są przechowywane w zupełnie innym formacie niż zwykłe tabele. Wymaga to dodania do istniejącej bazy grupy plików FILESTREAM lub wskazania takiej grupy plików w trakcie tworzenia bazy. Dane w tym modelu nie są przechowywane na stronach — zapisuje się je w pamięci w zoptymalizowanym formacie. Na dysku znajduje się tylko plik z punktami kontrolnymi.

UWAGA Omówienie rekordów danych (skompresowanych i nieskompresowanych), stron, ekstentów, bitmap alokacji, łańcuchów map IAM (ang. *Index Allocation Map*) i jednostek alokacji znajdziesz w rozdziale 11., „Optymalizowanie systemu SQL Server 2014”.

Po utworzeniu grupy plików możesz zdecydować, czy tabela ma być trwała, czy nietrwała. *Dane trwałe* są zachowywane na dysku. Takie dane są dostępne także po ponownym uruchomieniu systemu SQL Server. *Dane nietrwałe* (*dane schematu*) obejmują tylko strukturę tabeli, ale po restarcie systemu nie zachowują zapisanych w niej informacji. Dane nietrwałe znajdują się w pamięci, a ponowne uruchomienie systemu powoduje jej opróżnienie i wyzerowanie. Intuicyjnie może się wydawać, że wszystkie dane powinny być trwałe. Jednak w niektórych scenariuszach najlepszą wydajność zapewniają dane nietrwałe. Trzeba uwzględnić sposób używania danych przez aplikację.

Technologia In-Memory OLTP nie rozwiązuje wszystkich problemów z wydajnością systemu SQL Server, jednak pozwala uzyskać bardzo dużą poprawę wydajności — nawet od 9 do 150 razy. Omówienie wielu przykładowych przypadków znajdziesz w witrynie Microsoft Case Studies (<http://www.microsoft.com/casestudies/>). Wzrost wydajności o 150 razy jest opisany w studium przypadku dotyczącym firmy Baltika Breweries (<http://www.microsoft.com/casestudies/Microsoft-SQL-Server-2014/Baltika-Breweries/Brewer-Increases-CRM-System-Capacity-by-54-Percent-with-DataSolution/710000004225>).

Jednak technologia In-Memory OLTP sprawdza się tylko dla określonych procesów pracy. Pamiętaj, aby odpowiednio dostosować narzędzie do zadania. Dalej w tym rozdziale poznasz narzędzie ARM i dowiesz się, jak przy jego użyciu znaleźć tabele dobrze dopasowane do omawianej technologii.

Włączanie technologii In-Memory OLTP

Pierwszą rzeczą, jaką trzeba zrobić, aby rozpocząć korzystanie z technologii In-Memory OLTP, jest utworzenie odpowiedniej bazy danych. Należy w trakcie tworzenia bazy danych ustawić dla niej grupę plików FILESTREAM lub dodać taką grupę do istniejącej bazy. Możesz wykorzystać poniższy kod (patrz plik *KodR09.sql*) do utworzenia nowej bazy danych lub dodać grupę plików FILESTREAM do używanej bazy AdventureWorks. Zwróć uwagę na słowa kluczowe CONTAINS MEMORY_OPTIMIZED_DATA. Zauważ też, że nie trzeba włączać obsługi mechanizmu FILESTREAM w danym egzemplarzu systemu SQL Server, aby używać technologii In-Memory OLTP.

/ Tworzenie przykładowej bazy danych na potrzeby technologii In-Memory OLTP.*

**/*

USE [master]

GO

```

CREATE DATABASE [example_InMemOLTP]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'example_InMemOLTP', FILENAME = N'E:\MSSQL\example_InMemOLTP_xtp.mdf' ,
  SIZE = 3264KB ,MAXSIZE = UNLIMITED,FILEGROWTH = 1024KB ),
FILEGROUP [college_xtp_mod] CONTAINS MEMORY_OPTIMIZED_DATA DEFAULT
( NAME = N'example_InMemOLTP_mod1', FILENAME = N'F:\MSSQL\example_InMemOLTP_mod1'
  , MAXSIZE = UNLIMITED)
LOG ON
( NAME = N'example_InMemOLTP_log',
  FILENAME = N'E:\MSSQL\example_InMemOLTP_log.ldf' ,
  SIZE = 270016KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO

```

/*

Modyfikowanie bazy AdventureWorks w celu dodania grupy plików ze zoptymalizowanymi danymi przechowywanymi w pamięci.

*/

```

IF NOT EXISTS(SELECT * FROM AdventureWorks.sys.data_spaces WHERE type='FX')
ALTER DATABASE AdventureWorks
ADD FILEGROUP [AdventureWorks_Mod]CONTAINS MEMORY_OPTIMIZED_DATA
GO
IF NOT EXISTS(SELECT * FROM AdventureWorks.sys.data_spaces ds join
  AdventureWorks.sys.database_files df on ds.data_space_id=df.data_space_id
  where ds.type='FX')
ALTER DATABASE AdventureWorks
ADD FILE (name='AdventureWorks_Mod', filename='F:\MSSQL\Adventure_Works_Mod')
TO FILEGROUP AdventureWorks_Mod

```

Po dodaniu bazy danych (lub zmodyfikowaniu istniejącej) można przejść do tworzenia pierwszej tabeli. Jednak zanim zaczniesz, zapoznaj się z niektórymi ograniczeniami zoptymalizowanych tabel działających w pamięci.

- Po utworzeniu zoptymalizowanej tabeli działającej w pamięci nie można wprowadzać zmian w schemacie. Niedozwolone jest wywoływanie instrukcji ALTER TABLE, a wszystkie indeksy trzeba tworzyć wewnątrzwierszowo. Ponadto nie są obsługiwane instrukcje CREATE, DROP i ALTER INDEX.
- Każda tabela musi mieć przynajmniej jeden indeks.
- Dla zoptymalizowanych tabel można tworzyć tylko indeksy z haszowaniem i indeksy oparte na przedziałach.
- Dla typów danych z jednostkami alokacji innymi niż `in_row_data` nie można używać technologii In-Memory OLTP.
- Dla jednej zoptymalizowanej tabeli działającej w pamięci nie można utworzyć więcej niż 8 indeksów. Oprócz indeksu klucza głównego nie są stosowane indeksy unikatowe.
- Właściwość `IDENTITY` jest obsługiwana, ale musi zaczynać się od 1 i mieć krok równy 1.
- Niedozwolone są wyzwalacze DML i DDL, klucze obce i ograniczenia CHECK.
- Typy danych `LOB` i `ROW_OVER_FLOW` nie są obsługiwane. Nie można zdefiniować w tabeli wiersza o długości przekraczającej 8060 bajtów.
- Zoptymalizowanych tabel działających w pamięci nie można dzielić na partycje ani kompresować.
- Nie można określić kolejności sortowania dla indeksów z haszowaniem.

- Instrukcje SAVEPOINT w jawnych transakcjach, transakcjach powiązanych i w usłudze DTC (ang. *Distributed Transaction Coordinator*) nie mogą używać zoptymalizowanych tabel działających w pamięci.

UWAGA Pełną listę nieobsługiwanych instrukcji i funkcji w tej wersji technologii In-Memory OLTP znajdziesz w artykule *Transact-SQL Constructs Not Supported by In-memory OLTP* w witrynie MSDN: <http://msdn.microsoft.com/en-us/library/dn246937.aspx>.

Struktury tabel używanych w technologii In-Memory OLTP

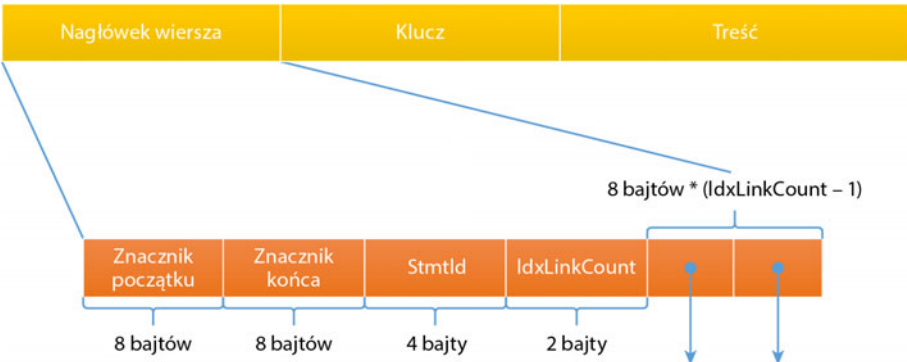
W trakcie poznawania In-Memory OLTP ważne jest, aby dobrze zrozumieć struktury danych używane w tabelach w tej technologii. W tym celu należy poznać hierarchię danych i zobaczyć, w jaki sposób są one przechowywane od najniższego do najwyższego poziomu. Na rysunku 9.1 przedstawiono listę struktur używanych na różnych poziomach (od górnego do dolnego).



Rysunek 9.1. Struktury danych

Rekordy

Zacznijemy od dołu i będziemy przechodzić w górę przez poszczególne poziomy danych w technologii In-Memory OLTP. Najniższym poziomem przechowywania danych są rekordy odpowiadające wierszom tabeli. Wiersze w technologii In-Memory OLTP składają się z nagłówka, klucza i treści. Na rysunku 9.2 pokazano szczegółową reprezentację graficzną tej struktury.



Rysunek 9.2. Struktura wiersza

Występują tu dwa pola po 8 bajtów. Przechowują one znaczniki opisane jako *Znacznik pocz.* i *Znacznik końc.* Są one wypełniane na podstawie dwóch wewnętrznych liczników (Transaction-ID i Global Transaction), które generują znaczniki dla baz danych obsługujących technologię In-Memory OLTP.

Licznik Transaction-ID generuje unikatowe wartości i jest zerowany w momencie ponownego uruchomienia systemu SQL Server. Licznik Global Transaction jest globalny i też generuje unikatowe wartości. Nie jest zerowany w momencie restartu systemu SQL Server. Przy przywracaniu stanu jest inicjowany najwyższym znacznikiem transakcji znalezionym w przywracanych rekordach. Wartość ta jest zwiększana po zakończeniu każdej transakcji. Nowa wartość staje się aktualnym znacznikiem dla bieżącej transakcji (dla nowego wiersza).

Zajmujący 4 bajty identyfikator instrukcji to następna wartość w nagłówku. Identyfikator ten, StmtId, jest unikatowy dla transakcji. Dzięki temu transakcja może dotrzeć do utworzonych przez siebie wierszy, jeśli potrzebny jest ponowny dostęp do nich.

Nagłówek obejmuje też zajmującą 2 bajty wartość IdxLinkCount. Jest to licznik referencji określający, w jak wielu indeksach występuje dany wiersz. Dalej znajdują się wskaźniki do wszystkich indeksów tabeli.

Pozostałe kolumny klucza i danych w wierszu to jego treść. Treść jest zależna od definicji tabeli. Wszystkie struktury przechowywane w pamięci to kod skompilowany, dlatego struktury obiektów po ich utworzeniu nie można zmienić. Aby dodać, usunąć lub zmodyfikować kolumnę, trzeba usunąć i ponownie utworzyć tabelę. To oznacza też, że wszystkie indeksy trzeba zdefiniować w trakcie tworzenia tabeli.

Indeksy

Podczas definiowania tabeli w technologii In-Memory OLTP trzeba określić przynajmniej jeden indeks. W tej technologii nie są używane strony IAM. Wskaźniki indeksów to jedyna metoda połączenia wierszy w tabelę. To dlatego każda tabela musi mieć przynajmniej jeden indeks.

Technologia In-Memory OLTP umożliwia stosowanie dwóch typów indeksów — indeksów z haszowaniem i indeksów opartych na przedziałach.

Indeksy z haszowaniem

W *indeksach z haszowaniem* używana jest funkcja haszująca, która przyjmuje wartości o dowolnym stałym rozmiarze (przy czym różnice w danych wejściowych są niewielkie) i zwraca wartości znacznie różniące się od siebie. Każda wartość po haszowaniu trafia do *kubelka*. Kubelki są tablicami wskaźników. Wskaźniki te tworzą indeks z haszowaniem. Na rysunku 9.3 przedstawiono jeden wiersz indeksu z haszowaniem dla kolumny Student.

W tym przykładzie w tablicy haszowania znajduje się osiem kubelków. Liczba kubelków zawsze jest zaokrąglana do najbliższej potęgi liczby 2. Więcej na ten temat dowiesz się dalej w tym rozdziale. Rekord z tabeli Student dotyczący Roberta Barana jest zapisany w kubelku 4. Wartość null wskaźnika indeksu oznacza, że jest to pierwszy rekord w indeksie. Na rysunku 9.4 pokazano, jak dodać do tabeli nowy wiersz z danymi Roberta Nowaka. Tym razem istnieje już początek łańcucha indeksu. Wiersz z danymi Roberta Nowaka trafia do tego samego kubelka, w którym znajduje się wiersz z danymi Roberta Barana. Wskaźnik indeksu w wierszu Roberta Nowaka prowadzi do poprzedniego wiersza.

Indeks z haszowaniem
dla kolumny Student

0
1
2
3
4
5
6
7

Znaczniki	Wskaźnik indeksu	Student	Kurs

Rysunek 9.3. Indeks z haszowaniemIndeks z haszowaniem
dla kolumny Student

0
1
2
3
4
5
6
7

Znaczniki	Wskaźnik indeksu	Student	Kurs

Rysunek 9.4. Dodawanie nowej wiersza do tabeli

Zwykle wartości po haszowaniu powinny trafić do odrębnych kubełków. Zbyt duża liczba wartości w jednym kubełku może obniżyć wydajność przetwarzania kwerend. W następnym ćwiczeniu wykorzystasz poniższy skrypt do utworzenia prostej tabeli i indeksu z haszowaniem.

```
use AdventureWorks
go
if exists(select name from sys.tables where name='mytable1')
begin
    drop table mytable1
end
go
use AdventureWorks
go
create table mytable1(
    myid int identity(1,1)
    ,mychar1 char(500) COLLATE Latin1_General_100_bin2 default
        'a' not null
    ,mychar2 char(3000) default 'b' not null
    constraint pk_mytbl_1 primary key
    nonclustered hash (myid,mychar1) with (bucket_count = 10)) with
    (memory_optimized = on)
declare @i int
set @i=0
while (@i<16)
begin
    insert into mytable1
    default values
    set @i=@i+1
end
```

Za pomocą widoku DMV `sys.dm_db_xtp_hash_index_stats` można sprawdzić liczbę utworzonych kubełków i poziom ich zapelnienia. Na rysunku 9.5 przedstawiono wyniki pobrane z tego widoku DMV.

```
SELECT OBJECT_NAME(S.object_id) as TableName
, I.name AS IndexName
, S.total_bucket_count
, S.empty_bucket_count
, S.avg_chain_length
, S.max_chain_length
FROM sys.dm_db_xtp_hash_index_stats S
JOIN sys.indexes I
on S.index_id = I.index_id
AND
S.object_id = I.object_id;
```

1	TableName	IndexName	total_bucket_count	empty_bucket_count	avg_chain_length	max_chain_length
1	mytable1	pk_mytbl_1	16	4	1	2

Rysunek 9.5. Informacje na temat kubełków

Jak widać, choć zdefiniowano tylko 10 kubełków, utworzonych zostało ich 16. Dzieje się tak, ponieważ liczba kubełków tworzonych przez system SQL Server musi być potęgą liczby 2. Liczba 8 jest potęgą liczby 2, 10 jest większe niż 8, dlatego SQL Server musi użyć następnej potęgi liczby 2, czyli wartości 16.

Kubełków jest 16, a 4 z nich są puste. Średnia długość łańcucha w kubełku równa się 1. To dobrze. Haszowane wartości powinny być w jak największym stopniu unikatowe, aby przy wyszukiwaniu można było szybko znajdować unikatowe wiersze na podstawie kubełka i przechowywanej w nim tablicy wskaźników.

Maksymalna długość łańcucha to 2, co oznacza, że w niektórych kubełkach znajduje się więcej niż 1 wiersz. Jednak 2 to akceptowalna wartość. Należy przy tym pamiętać, że w trakcie definiowania tabeli trzeba określić liczbę kubełków. Nie można później wprowadzać zmian za pomocą instrukcji DDL. Nie można zastosować polecenia `ALTER TABLE` lub `ALTER INDEX` i zmienić definicję indeksu. Jeśli do omawianej tabeli wstawisz tysiące wierszy, w kubełkach znajdzie się zbyt wiele elementów.

Jeśli operacja wyszukiwania sprawdza kubełek i próbuje znaleźć w nim wiersz, będzie musiała sprawdzić wiele wartości, co spowolni proces pobierania danych. Dlatego należy bardzo precyzyjnie zdefiniować indeks.

Pamiętaj, że indeks z haszowaniem też jest indeksem i służy do pobierania danych z tabeli. Dlatego indeks z haszowaniem utwórz dla kolumn, które zamierzasz stosować jako predykaty w kwerendach. Kwerenda używająca kolumny `mychar2` wymagałaby pełnego skanowania tabeli.

Pewnie zastanawiasz się, kiedy należy stosować indeks z haszowaniem? To doskonałe pytanie. Tego rodzaju indeks warto stosować, gdy wiadomo, że wykonywanych będzie wiele operacji wyszukiwania, a programista chce tworzyć precyzyjne kwerendy zwracające tylko jeden wiersz lub zbiór określonych wierszy. Przyjmijmy, że napisał następującą kwerendę:

```
select * from dbo.mytable1 where myid=1
```

Szukany jest konkretny wiersz. Jeśli kwerenda dotyczy tabeli zamówień, w ten sposób można pobierać po jednym zamówieniu. Jeżeli jednak programistę interesuje przedział wartości, np. zamówienia złożone po określonej dacie, wtedy indeks z haszowaniem nie jest odpowiedni.

Gdy kwerendy dotyczą przedziałów, należy zastosować indeks oparty na przedziałach.

Indeksy oparte na przedziałach

Indeks oparty na przedziałach działa bardzo podobnie do indeksów, do których użytkownicy systemu SQL Server powinni być przyzwyczajeni. Taki indeks przechowuje i sortuje dane w przedziałach oraz ułatwia korzystanie z nich. Przykładowo A – Z to przedział liter, a 1 – 100 to przedział liczb. Indeksy oparte na przedziałach służą do porządkowania wartości w określony sposób.

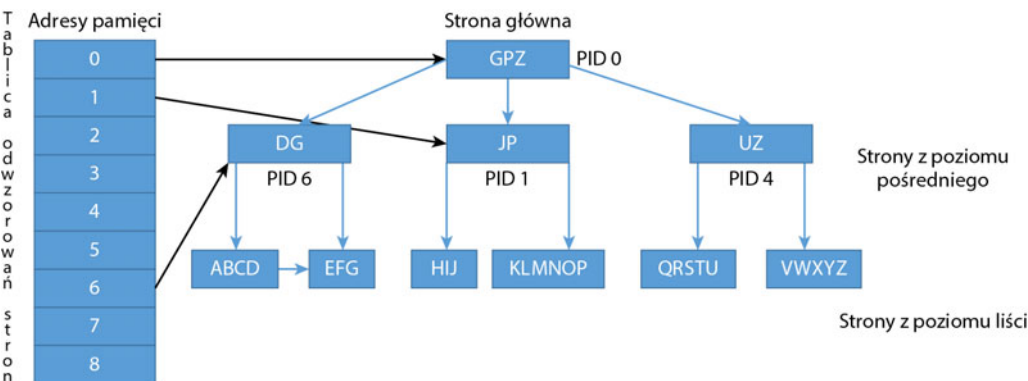
Wewnętrznie indeksy oparte na przedziałach wykorzystują Bw-drzewo (ang. *Bw-tree*) opracowane przez dział badawczy Microsoftu w 2011 roku. Ogólna struktura Bw-drzewa jest bardzo podobna do struktury B-drzewa używanego w indeksach klastrowanych i nieklastrowanych. Jednak Bw-drzewo może działać współbieżnie, co pozwala stosować struktury bez zamków i blokad. Zapewnia to wzrost wydajności w porównaniu z tabelami przechowywanymi na dysku.

UWAGA Aby dowiedzieć się więcej na temat Bw-drzew, zapoznaj się z pracą *The Bw-Tree: A B-tree for New Hardware Platforms* (<http://research.microsoft.com/pubs/178758/bw-tree-icde2013-final.pdf>). Opisano w niej strukturę bez zamków i blokad, którą wykorzystano do budowania opartych na przedziałach indeksów stosowanych w technologii In-Memory OLTP.

Strony indeksów opartych na przedziałach mogą mieć różną wielkość, jednak po utworzeniu strony rozmiaru nie można zmienić. Domyślnie wielkość strony jest równa 8 kilobajtom, jednak w zależności od danych zapisanych w indeksie wartość ta może się zmieniać. Każda strona indeksu zawiera zestaw uporządkowanych wartości kluczy.

Istnieją strony główne, pośrednie i liście. Stron nie można aktualizować, zatem w momencie wstawiania lub usuwania rekordów tworzone są nowe wersje stron.

Ponieważ strony indeksów są tak często modyfikowane, prawdopodobnie zastanawiasz się, w jaki sposób kwerendy poruszają się po łańcuchach indeksów przy wyszukiwaniu danych. Używana jest wtedy *tablica odwzorowań stron* (ang. *page mapping table*). W tej tablicy zapisane są identyfikatory PID, czyli logiczne identyfikatory stron, a nie fizyczne numery stron. Gdy w wyniku aktualizacji tworzona jest nowa strona indeksu, następuje też aktualizacja tablicy odwzorowań stron, a gdy kwerendy nie używają już danej strony indeksu, może ona zostać usunięta przez mechanizm przywracania pamięci. Strukturę indeksów opartych na przedziałach przedstawiono na rysunku 9.6.



Rysunek 9.6. Struktura indeksów opartych na przedziałach

Zauważ, że najwyższa wartość z każdej strony z poziomu liści jest przenoszona do stron z wyższego poziomu. Na tym polega różnica między Bw-drzewami a B-drzewami. Strony z poziomu liści obejmują wartości kluczy, ale zamiast identyfikatorów PID przechowywanych na stronach z wyższego poziomu zawierają adres pamięci, gdzie zapisane są dane.

Zmiany na stronach z poziomu liści są śledzone za pomocą *tablicy różnic*. W tej tablicy zapisywany jest jeden wiersz dla każdej operacji wstawienia lub usunięcia danych. Dla operacji aktualizacji wstawiany jest wiersz z poprzednią i zaktualizowaną wartością.

Na każdej stronie indeksu znajduje się nagłówek obejmujący następujące wartości.

- *Identyfikator PID*. Jest to wskaźnik do tablicy odwzorowań.
- *Typ strony*. Określa typ strony w indeksie.
- *Identyfikator PID prawej strony*. Jest to identyfikator PID strony zapisanej na prawo od bieżącej.
- *Wysokość*. Określa liczbę poziomów od bieżącej strony do liścia.
- *Statystyki strony*. Jest to liczba rekordów na stronie plus liczba rekordów różnic.
- *Maksymalny klucz*. Jest to górny limit wartości przechowywanych na stronie.

Zakładamy, że już utworzyłeś nową bazę lub zmodyfikowałeś istniejącą. Poznałeś też wewnętrzne struktury omawianej technologii. Pora przejść do tworzenia pierwszej tabeli, czego ilustracją jest poniższy kod (patrz plik *KodR09.sql*).

```
/*
Pierwsza zoptymalizowana tabela działająca w pamięci.
*/
use example_InMemOLTP
go
CREATE TABLE first_InMemOLTP(
    c1 int identity(1,1) NOT NULL PRIMARY KEY NONCLUSTERED
    ,c2 int NOT NULL INDEX first_xtp_nclx_index
        hash with(bucket_count=10000000)
    ,c3 int NOT NULL INDEX first_xtp_nclx_index2
        hash with(bucket_count=10000000)
    ,c4 int NOT NULL INDEX first_xtp_nclx_index3
        hash with(bucket_count=10000000)
    ,c5 int
    ,c6 char(100) COLLATE Latin1_General_100_bin2 NOT NULL INDEX
        first_xtp_nclx_char
    ,c7 char(80)
)WITH(memory_optimized=on, DURABILITY=SCHEMA_AND_DATA)
```

Zwróć uwagę na różnice w porównaniu ze standardową definicją tabeli. Wszystkie indeksy są tu zdefiniowane wewnątrzwerszowo. Ponadto nie są używane indeksy klastrowane. Aby zdefiniować kolumnę typu CHAR, VARCHAR lub typu ze znakami Unicode, trzeba wewnątrzwerszowo ustawić kolację z rodziny *_BIN2. Aby ustalić wszystkie takie kolacje, zastosuj następującą kwerendę:

```
select * from sys.fn_helpcollations() where name like '%BIN2'
```

Wiesz już, że możesz określić, czy tabele w technologii In-Memory OLTP mają być trwałe. Tu parametr DURABILITY ma wartość SCHEMA_AND_DATA. To sprawia, że dane są utrwalane w dzienniku za pomocą mechanizmu zapisu z wyprzedzeniem (ang. *write-ahead logging*) stosowanego w systemie SQL Server. Dla danych poza pamięcią tworzony jest punkt kontrolny, a ponadto dane są zapisywane w zoptymalizowanym pliku działającym w pamięci.

Dla parametru `DURABILITY` można też ustawić wartość `SCHEMA_ONLY`. To powoduje wyłączenie zapisu z wyprzedzeniem dla danej tabeli. Dla danych poza pamięcią nie są wtedy tworzone punkty kontrolne, a przy ponownym uruchamianiu systemu SQL Server wszystkie dane zostają utracone. Gdy tabela jest zapisywana z powrotem w pamięci w trakcie restartu systemu, odtwarzany jest wyłącznie jej schemat.

Kwestie związane z procesorem

Aby możliwe było stosowanie technologii In-Memory OLTP, sprzęt, na którym działa system SQL Server 2014, musi obsługiwać instrukcję `cmpxchg16b` (umożliwia to wykonywanie atomowych operacji na słowach w systemie ósemkowym). Jest ona używana w algorytmach współbieżnych, które porównują i przedstawiają dane większe niż wielkość wskaźnika. Instrukcja ta jest często wykorzystywana w algorytmach działających bez blokad i konieczności oczekiwania. Instrukcję `cmpxchg16b` obsługują wszystkie współczesne procesory oparte na architekturze x64.

Obecna wersja technologii In-Memory OLTP ma działać optymalnie na serwerach mających od 2 do 4 gniazd i mniej niż 60 rdzeni.

Zagadnienia związane z wirtualizacją

Jeśli stosujesz wirtualizację, host musi obsługiwać instrukcję `cmpxchg`. Gdy korzystasz z oprogramowania Hyper-V, instrukcja ta jest obsługiwana niezależnie od używanego sprzętu.

W typowych środowiskach wirtualnych należy zapewnić wirtualnemu systemowi operacyjnemu wystarczającą ilość zasobów hosta. Zaleca się ustawienie minimalnej ilości pamięci serwera (nazwa tej opcji zależy od producenta oprogramowania do obsługi wirtualizacji). Inną dobrą praktyką jest unikanie ustawiania zbyt dużej ilości początkowej pamięci. Pozwala to uniknąć zagłodzenia innych procesów z powodu braku potrzebnej im pamięci.

W systemach wirtualnych, w których używa się technologii In-Memory OLTP, sytuacja wygląda *inaczej*. Gdy baza danych wykorzystująca tę technologię staje się aktywna, próbuje przywrócić dane z pliku z punktem kontrolnym. Technologia In-Memory OLTP przenosi dane do pamięci w bardziej agresywny sposób niż mechanizm dynamicznego przydziału pamięci udostępnia ją bazie danych. Może to prowadzić do tego, że baza pozostanie w trybie „przywracanie stanu w toku” nawet wtedy, kiedy dostępna jest wystarczająca ilość pamięci na zapisanie w niej danych.

Dlatego zalecaną praktyką jest wstępne przydzielenie wystarczającej ilości pamięci, aby pomieściła wszystkie dane przechowywane w technologii In-Memory OLTP. Jeśli system hosta dynamicznie przydziela pamięć na żądanie (w zależności od ustawienia minimalnej ilości pamięci serwera), mogą wystąpić problemy z danymi używanymi w omawianej technologii.

Kwestie związane z pamięcią

Przed zastosowaniem technologii In-Memory OLTP trzeba zrozumieć wymagania dotyczące pamięci. W tym celu należy oszacować ilość potrzebnej pamięci. Ten proces przebiega tak samo niezależnie od tego, czy przenosisz do tej technologii tabelę z dysku, czy stworzysz nową zoptymalizowaną tabelę działającą w pamięci.

Do liczby bajtów przechowywanych w wierszu należy dodać po 24 bajty na nagłówek wiersza i znacznik. Trzeba też uwzględnić wskaźniki indeksów. Każdy indeks w wierszu zajmuje 8-bajtowy wskaźnik do adresu następnego wiersza w danym indeksie.

Przeanalizuj poniższy przykładowy kod (patrz plik *KodR09.sql*).

```
CREATE TABLE xtp_Estimate(
    c1 int identity(1,1) NOT NULL PRIMARY KEY NONCLUSTERED
    ,c2 int NOT NULL INDEX xtp_nclx_index
        hash with(bucket_count=10000000)
    ,c3 int NOT NULL INDEX xtp_nclx_index2
        hash with(bucket_count=10000000)
    ,c4 int NOT NULL INDEX xtp_nclx_index3
        hash with(bucket_count=10000000)
    ,c5 int NOT NULL INDEX xtp_nclx_index4
    ,c6 char(100)
    ,c7 char(80)
)WITH(memory_optimized=on)
```

W tym kodzie używane są cztery indeksy. Gdy zaczniesz dodawać bajty wiersza, możesz zacząć od metadanych. Nagłówek zajmuje 24 bajty, a indeksy — 32 bajty, co razem daje 56 bajtów.

Tabela obejmuje pięć kolumn liczb całkowitych po 4 bajty każda, co daje 20 bajtów. W sumie mamy więc 76 bajtów. Ponadto używane są dwie kolumny danych znakowych. Jednak zajmuje 100, a druga — 80 bajtów. Łącznie potrzeba więc 256 bajtów na wiersz.

W tabeli 9.1 przedstawiono kompletną analizę wierszy.

Tabela 9.1. Łączna liczba bajtów na wiersz

Element wiersza	Wielkość w bajtach
Wskaźniki indeksów	8
Liczba indeksów	4
Łączna wielkość indeksów	32
Znacznik i nagłówek	24
Dane	200
Łączna liczba bajtów na wiersz	256

Ta wartość jest bardzo ważna. Po jej obliczeniu możesz oszacować, ile pamięci będzie potrzebne. W tym celu określ liczbę wierszy danych w tabeli. Ponieważ w technologii In-Memory OLTP używany jest współbieżny magazyn wersji, oszacowaną wartość należy pomnożyć przez 4, aby mieć pewność, że ilość pamięci będzie wystarczająca.

Jeśli tabela ma zawierać 10 milionów wierszy, potrzebnych będzie 9,54 gigabajta pamięci, 50 milionów wierszy będzie wymagać 47,68 gigabajta pamięci, dla 100 milionów wierszy należy zapewnić 95,37 gigabajta pamięci itd.

W tabeli 9.2 przedstawiono kompletną analizę wymagań pamięciowych.

Wielkość tablicy haszowania można ustawić za pomocą parametru `bucket_count=<wartość>` w definicji tabeli. Wartość parametru `bucket_count` musi być potęgą liczby 2. Jeśli podasz inną wartość, zostanie ona zaokrąglona w górę do najbliższej potęgi tej liczby.

W omawianym przykładzie liczba wierszy to 10 000 000. Najbliższą potęgą liczby 2 jest 2^{24} , czyli 16 777 216. Każdy kubełek zajmuje 8 bajtów. Należy więc pomnożyć $16\,777\,216 \cdot 8$. Po podzieleniu uzyskanej wartości przez 1024 (w celu otrzymania liczby kilobajtów) i ponownie przez 1024 (w celu wyznaczenia liczby megabajtów) stwierdzisz, że na indeks z haszowaniem potrzeba 128 megabajtów. W przedstawianym przykładzie używane są 3 takie indeksy, co oznacza, że łącznie wymagają one 384 megabajtów pamięci.

Tabela 9.2. Kompletne szacunki rozmiaru danych

Wielkość wiersza (w bajtach)	Liczba wierszy	Wielkość w kilobajtach	Wielkość w megabajtach	Wielkość w gigabajtach	Wielkość w gigabajtach (pamięć wersji × 4)
256	10 000 000	2 500 000	2 441,41	2,38	9,54
256	50 000 000	12 500 000	12 207,03	11,92	47,68
256	100 000 000	25 000 000	24 414,06	23,84	95,37
256	500 000 000	125 000 000	122 070,31	119,21	476,84
256	1 000 000 000	250 000 000	244 140,63	238,42	953,67

W tabeli 9.3 przedstawiono przegląd różnych potęg liczby 2 i wymaganej ilości pamięci.

Tabela 9.3. Analiza pojemności pamięci potrzebnej na indeks z haszowaniem dla różnych potęg liczby 2

Potęga liczby 2	Wartość	Pojemność (w bajtach)	Pojemność (w kilobajtach)	Pojemność (w megabajtach)
2 ¹⁵	32 768	262 144	256	0,25
2 ¹⁶	65 536	524 288	512	0,5
2 ¹⁷	131 072	1 048 576	1024	1
2 ¹⁸	262 144	2 097 152	2048	2
2 ¹⁹	524 288	4 194 304	4096	4
2 ²⁰	1 048 576	8 388 608	8192	8
2 ²¹	2 097 152	16 777 216	16 384	16
2 ²²	4 194 304	33 554 432	32 768	32
2 ²³	8 388 608	67 108 864	65 536	64
2 ²⁴	16 777 216	134 217 728	131 072	128

W zoptymalizowanych tabelach działających w pamięci można stosować *indeksy oparte na przedziałach* i *indeksy z haszowaniem*. W omawianym przykładzie w definicji tabeli tworzone są 4 indeksy: 3 indeksy z haszowaniem i indeks oparty na przedziałach. Określenie „indeks oparty na przedziałach” może wydawać się nowe i wymyślne, jest to jednak indeks tego samego rodzaju, jaki standardowo stosuje się w systemie SQL Server. Indeksy oparte na przedziałach wykorzystują przedziały wartości (od 1 do 1 000 000, od A do Z lub dowolną inną listę, którą można uporządkować).

Obliczanie ilości pamięci potrzebnej na indeks oparty na przedziałach nie jest tak proste jak pokazano wcześniej. Takie indeksy wykorzystują B+-drzewo, które działa podobnie (choć nie identycznie) jak B-drzewa używane na dysku. Węzły, które nie są liśćmi, zajmują procentowo bardzo niewiele pamięci, dlatego można je pominąć.

Aby obliczyć pamięć dla liści, trzeba uwzględnić unikatowy klucz zdefiniowany dla tabeli. Tu jest to kolumna c1 zawierająca 4-bajtowy identyfikator. Tę wielkość należy dodać do wielkości kolumny z indeksem opartym na przedziałach (tu jest to kolumna c5) i wielkości wskaźnika

wiersza. Jeśli istnieje grupa wierszy o tym samym kluczu (jest to możliwe np. w indeksach nieklastrowanych i nieunikatowych), wiersze te są łączone. W omawianym przykładzie obliczenia powinny wyglądać tak:

Wielkość indeksu opartego na przedziałach = (Wskaźnik (8 bajtów) + suma (Kolumna klucza (4 bajty) + Kolumna indeksu (4 bajty)) * Unikatowe wiersze)

Wielkość indeksu opartego na przedziałach = $(8 + 8) * 100\,000\,000$ bajtów = 1,49 gigabajta

Aby ustalić łączną ilość pamięci potrzebną do zapisania 100 000 000 wierszy z przykładowej tabeli, należy przeprowadzić następujące obliczenia: 95,37 gigabajta + 1,49 gigabajta + 0,384 gigabajta. To daje w sumie 97,24 gigabajta.

Jeśli używasz w kwerendach zmiennych reprezentujących tabele przechowywane w pamięci, pamięć na te zmienne jest zajmowana niezależnie od pamięci zdefiniowanej wcześniej dla przykładowej tabeli. Ponadto należy uwzględnić pamięć potrzebną na powiększanie rozmiaru tabeli.

Dlatego ważne jest, aby z góry ustalić potrzebną ilość pamięci. Pozwoli to uwzględnić otrzymaną wartość przy zakupie sprzętu i ustawianiu opcji MAX MEMORY w systemie SQL Server. Domyślnie technologia In-Memory OLTP korzysta z puli buforów i używa pamięci standardowo przydzielanej innym procesom systemu SQL Server.

Zarządzanie pamięcią za pomocą funkcji Resource Governor

Wiesz już, jak oszacować ilość pamięci potrzebną technologii In-Memory OLTP. Bazy danych korzystające z tej technologii powiązane są z domyślną pulą obejmującą wszystkie zasoby użytkownika dostępne dla systemu SQL Server.

Bazę danych można powiązać z pulą zasobów. Dla zwykłych baz danych to ustawienie jest ignorowane, chyba że użytkownik jest powiązany z daną pulą. Kompletne omówienie funkcji Resource Governor zamieściliśmy w rozdziale 11., „Optymalizowanie systemu SQL Server 2014”.

Po powiązaniu bazy danych z pulą można kontrolować pamięć przydzielaną tej bazie. Aby powiązać bazę z pulą, należy najpierw ustalić wartości MIN_MEMORY_PERCENT i MAX_MEMORY_PERCENT. Wartości te powinny odpowiadać górnej granicy szacowanej ilości pamięci potrzebnej dla planowanych zoptymalizowanych tabel działających w pamięci i ich indeksów.

Przy określaniu minimalnego i maksymalnego procentu zajmowanej pamięci należy pamiętać, że technologia In-Memory OLTP nie ma dostępu do 100% ustawionej wartości. W tabeli 9.4 pokazano, w jaki sposób ilość fizycznej pamięci wpływa na jej procent dostępny dla zoptymalizowanych tabel działających w pamięci.

Tabela 9.4. Procent pamięci dostępnej dla technologii In-Memory OLTP

Docelowa ilość przydzielanej pamięci	Procent dostępny dla tabel przechowywanych w pamięci
<= 8 gigabajtów	70%
<= 16 gigabajtów	75%
<= 32 gigabajty	80%
<= 96 gigabajtów	85%
> 96 gigabajtów	90%

Jeśli tabela zajmuje 60 gigabajtów, a dostępne jest tylko 100 gigabajtów, trzeba ustalić maksymalny procent przydzielanej pamięci. W tym celu należy podzielić 60 gigabajtów przez 90% dostępnej docelowej pamięci. To daje $(60 / 0,90) = 66,7$ (po zaokrągleniu). Tak więc opcję `MAX_MEMORY_PERCENT` trzeba ustawić na 67%.

Wartość opcji `MAX_MEMORY_PERCENT` można zmieniać na bieżąco, co oznacza, że nie trzeba w tym celu ponownie uruchamiać systemu. Jednak zastosowanie zarządzanej przez funkcję Resource Governor bazy danych w technologii In-Memory OLTP wymaga wyłączenia tej bazy. Należy wykonać ten krok przed udostępnieniem bazy w środowisku produkcyjnym lub — jeśli baza już została udostępniona — w trakcie konserwacji systemu.

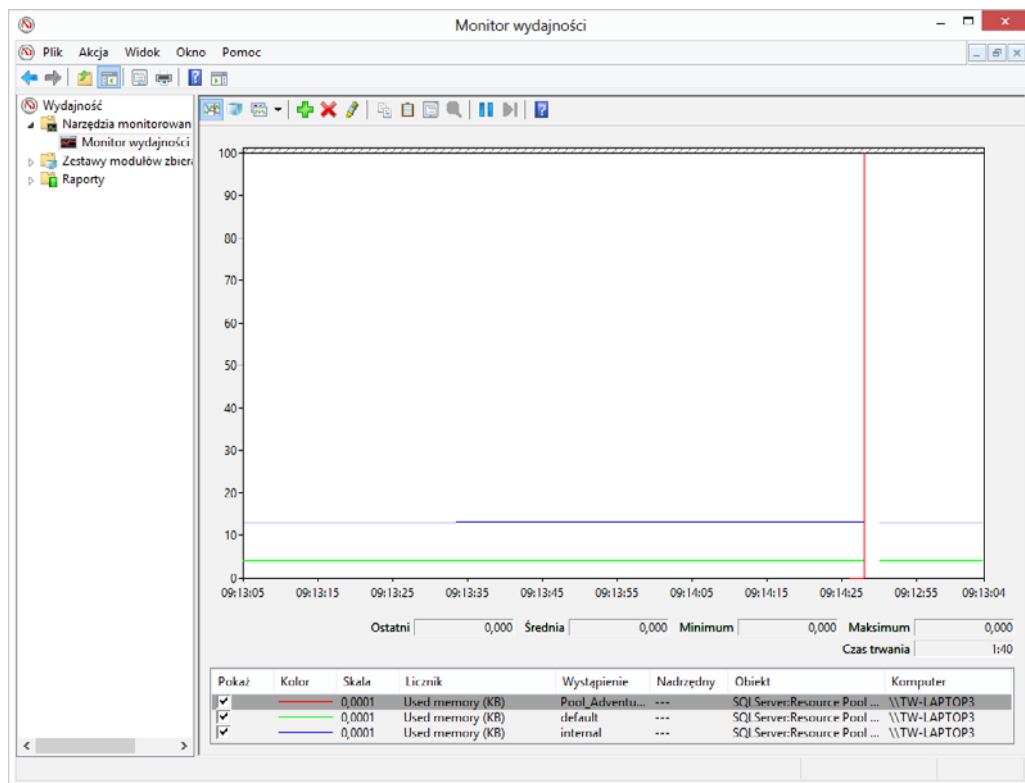
Najpierw utwórz pulę zasobów (potrzebny kod zawiera plik *KodR09.sql*). Następnie wywołaj systemową procedurę składowaną `sp_xtp_bind_db_resource_pool`, aby powiązać bazę danych z pulą zasobów. W ostatnim kroku użyj poniższego skryptu, aby wyłączyć bazę danych i włączyć ją ponownie:

```
/*
Tworzenie puli zasobów na potrzeby przechowywanej w pamięci bazy AdventureWorks.
*/
USE master
GO
IF NOT EXISTS (SELECT * FROM sys.resource_governor_resource_pools
               WHERE name=N'Pool_AdventureWorks')
BEGIN
    CREATE RESOURCE POOL Pool_AdventureWorks
        WITH ( MAX_MEMORY_PERCENT = 80 );
    ALTER RESOURCE GOVERNOR RECONFIGURE;
END
GO
/*
Wiązanie bazy AdventureWorks z pulą zasobów.
*/
EXEC sp_xtp_bind_db_resource_pool 'AdventureWorks', 'Pool_AdventureWorks'
GO
/*
Wyłączanie bazy AdventureWorks i ponowne jej włączanie.
*/
ALTER DATABASE AdventureWorks SET OFFLINE WITH ROLLBACK IMMEDIATE
GO
ALTER DATABASE AdventureWorks SET ONLINE
GO
```

Do śledzenia poziomu wykorzystania pamięci możesz wykorzystać monitor wydajności i liczniki z grupy *SQLServer:Resource Pool Stats*. Na rysunku 9.7 przedstawiono poziom wykorzystania pamięci dla grup zasobów.

Na rysunku 9.8 można zobaczyć dane o wykorzystaniu pamięci uzyskane za pomocą widoku DMV i poniższego skryptu (patrz plik *KodR09.sql*):

```
SELECT pool_id
       , Name
       , min_memory_percent
       , max_memory_percent
       , max_memory_kb/1024 AS max_memory_mb
       , used_memory_kb/1024 AS used_memory_mb
       , target_memory_kb/1024 AS target_memory_mb
FROM sys.dm_resource_governor_resource_pools
```



Rysunek 9.7. Poziom wykorzystania pamięci w monitorze wydajności

QLQuery13.sql - S...Administrator (70) BradleyBall_Chapter...Administrator (67) In-Memory OLTP Sa...administrator (65) find hash bucket co...Administrator (69) SQLQuery9.sql - SQ...Ad							
Editor Results Messages							
	pool_id	Name	min_memory_percent	max_memory_percent	max_memory_mb	used_memory_mb	target_memory_mb
1	1	internal	0	100	6901	333	6901
2	2	default	0	100	6901	2548	6901
3	256	Pool_AdventureWorks	0	80	5521	855	5521

Rysunek 9.8. Poziom wykorzystania pamięci uzyskany za pomocą widoku DMV

Tworzenie natywnie kompilowanych procedur składowanych

Za pomocą języka T-SQL można uzyskać dostęp i do zoptymalizowanych tabel działających w pamięci, i do zwykłych tabel przechowywanych na dysku. System SQL Server umożliwia ponadto pisanie natywnie kompilowanych procedur składowanych. Można w nich używać języka T-SQL, jednak obowiązują przy tym pewne ograniczenia i dostępnych jest kilka nowych słów kluczowych.

Natywnie kompilowane procedury składowane są kompilowane w momencie ich tworzenia lub w trakcie ładowania systemu SQL Server. Skompilowany kod zawiera znacznie mniej instrukcji przeznaczonych do wykonania przez procesor. To pozwala na szybsze wykonywanie procedur składowanych i wydajniejsze wykorzystanie mocy procesora.

W natywnie kompilowanych procedurach składowanych trzeba deklarować transakcje atomowe (słowo kluczowe `ATOMIC`). Inaczej niż w zwykłym kodzie w języku T-SQL należy tu jawnie deklarować transakcje i nie można stosować transakcji zagnieżdżonych. Jeśli opracujesz transakcję zagnieżdżoną, system SQL Server automatycznie utworzy punkt zapisu przed przejściem do następnej transakcji atomowej.

Natywnie kompilowaną procedurę składowaną i używaną przez nią tabelę trzeba powiązać ze schematem, z którego korzystają te elementy. To sprawia, że tabeli nie można usunąć bez wcześniejszego usunięcia danej procedury.

Natywnie kompilowane procedury składowane nie obsługują klauzuli `EXECUTE AS A CALLER`. Takie procedury działają tylko w trybie `EXECUTE AS OWNER`. Wykorzystaj poniższy kod (patrz plik *KodR09.sql*), aby utworzyć natywnie kompilowaną procedurę składowaną wstawiającą rekordy do przygotowanej wcześniej tabeli `xtp_Estimate`:

```
if exists(select name from sys.procedures where name='p_ins_new_record')
begin
    DROP PROCEDURE dbo.p_ins_new_record
end
go
CREATE PROCEDURE dbo.p_ins_new_record (@rowcount int)
WITH NATIVE_COMPILATION
,SCHEMABINDING
,EXECUTE AS OWNER
AS
BEGIN ATOMIC WITH
(TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE='us_english')
declare @i int=0
while @i<@rowcount
    begin
        set @i=@i+1
        INSERT INTO dbo.xtp_Estimate(c2, c3, c4, c5, c6, c7)
        VALUES(@i, @i, @i, @i, 'a', 'b')
    end
END
```

Aby zobaczyć, jakie możliwości i wydajność oferują natywnie kompilowane procedury składowane, utwórz na dysku tabelę o strukturze podobnej do używanej wcześniej tabeli `xtp_Estimate` (patrz plik *KodR09.sql*). Następnie napisz dwie dodatkowe procedury składowane. Pierwsza powinna wstawiać nowy wiersz danych do tabeli `disk_Estimate`, a druga — wiersz do tabeli `xtp_Estimate` za pomocą standardowej procedury składowanej w języku T-SQL. Następnie wstaw do każdej tabeli po 10 000 wierszy i sprawdź, która procedura jest najszybsza.

```
/*
Tworzenie struktur tabeli zapisanej na dysku.
*/
use example_InMemOLTP
go
if exists(select name from sys.objects where name='disk_Estimate')
begin
    drop table disk_Estimate
end
go
CREATE TABLE disk_Estimate(
    c1 int identity(1,1) NOT NULL PRIMARY KEY NONCLUSTERED
    ,c2 int NOT NULL INDEX disk_nclx_index
    ,c3 int NOT NULL INDEX disk_nclx_index2
```



```

,c4 int NOT NULL INDEX disk_nclx_index3
,c5 int NOT NULL INDEX disk_nclx_index4
,c6 char(100)
,c7 char(80)
)

if exists(select name from sys.procedures where name='p_ins_new_disk_record')
begin
    DROP PROCEDURE dbo.p_ins_new_disk_record
end
go
CREATE PROCEDURE dbo.p_ins_new_disk_record (@c2 int, @c3 int, @c4 int, @c5 int,
@c6 char(100),@c7 char(80))
AS
BEGIN
INSERT INTO dbo.disk_Estimate(c2, c3, c4, c5, c6, c7)
VALUES(@c2, @c3, @c4, @c5, @c6, @c7)
END
GO
if exists(select name from sys.procedures where name='p_ins_tsq1_xtp_record')
begin
    DROP PROCEDURE dbo.p_ins_tsq1_xtp_record
end
go
CREATE PROCEDURE dbo.p_ins_tsq1_xtp_record (@c2 int, @c3 int, @c4 int, @c5 int,
@c6 char(100), @c7 char(80))
AS
BEGIN
INSERT INTO dbo.xtp_Estimate(c2, c3, c4, c5, c6, c7)
VALUES(@c2, @c3, @c4, @c5, @c6, @c7)
END
GO
/*
Wykonywanie procedury składowanej.
*/
declare @starttime datetime2=sysdatetime()
declare @timems int
declare @i int
-- Tabela zapisana na dysku i procedura składowana w zwykłym języku T-SQL.
set @starttime=sysdatetime()
set @i=0
while @i<10000
begin
    set @i=@i+1
    exec dbo.p_ins_new_disk_record @i, @i, @i, @i, 'a', 'b'
end

set @timems = datediff(ms, @starttime, sysdatetime())
select 'Tabela z dysku i procedura w języku T-SQL: ' + cast(@timems as varchar(10))
+ 'ms'
-- Zoptymalizowana tabela działająca w pamięci i procedura składowana w zwykłym języku T-SQL.
set @starttime=sysdatetime()
set @i=0
while @i<10000
begin
    set @i=@i+1
    exec dbo.p_ins_tsq1_xtp_record @i, @i, @i, @i, 'a', 'b'
end

```

```

set @timems = datediff(ms, @starttime, sysdatetime())
select 'Tabela w pamięci i procedura w języku T-SQL: '
      + cast(@timems as varchar(10)) + ' ms'
-- Natywnie kompilowana procedura.
set @starttime=sysdatetime()
exec dbo.p_ins_new_record 10000

```

```

set @timems = datediff(ms, @starttime, sysdatetime())
select 'Natywnie kompilowana procedura: ' + cast(@timems as varchar(10)) + ' ms'

```

Na rysunku 9.9 przedstawiono uzyskane wyniki. Procedura składowana w zwykłym języku T-SQL działająca na tabeli zapisanej na dysku wykonała zadanie w 3 628 milisekund. Podobna procedura używająca zoptymalizowanej tabeli działającej w pamięci zakończyła pracę w 3 103 milisekundy. Natywnie kompilowana procedura składowana wykonała zadanie w 79 milisekund.

Results		Messages	
(No column name)			
1	Tabela z dysku i procedura w języku T-SQL: 2760ms		
(No column name)			
1	Tabela w pamięci i procedura w języku T-SQL: 3233 ms		
(No column name)			
1	Natywnie kompilowana procedura: 70 ms		

Rysunek 9.9. Czas wykonania zadania przez różne rodzaje procedur

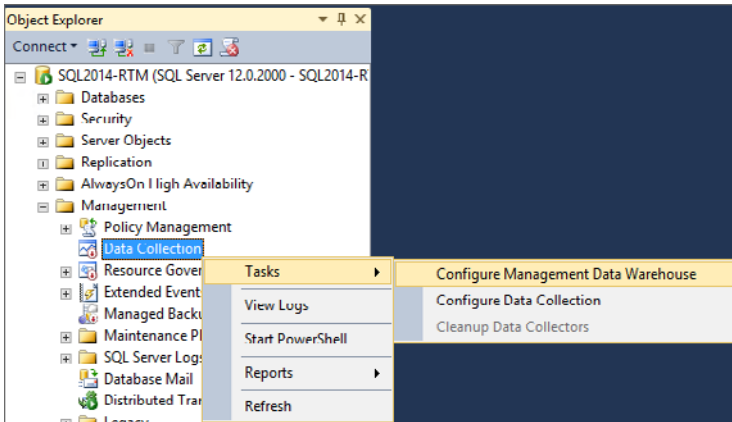
Omówienie narzędzia ARM

W systemie SQL Server 2014 znacznie zmodyfikowano bazy MDW (ang. *Management Data Warehouse*). Dodano do nich m.in. nowe funkcje oparte na kolektorach danych Transaction Performance. Kolektory te badają fizyczną strukturę i sposób użytkowania tabel za pomocą kolektora danych Table Usage Analysis. Bazy MDW wskazują następnie tabele i procedury, które warto zastosować w technologii In-Memory OLTP. Procedury składowane są analizowane pod kątem liczby wykonań i wykorzystania procesora przez kolektor danych Stored Procedure Usage Analysis.

UWAGA Więcej informacji o kolektorach danych zamieszczono w rozdziale 12., „Monitorowanie systemu SQL Server”.

Dalej pokazujemy, jak zastosować wymienione kolektory danych i zbadać zwrócone przez nie wyniki.

Nawiąż za pomocą programu SQL Server Management Studio połączenie z egzemplarzem systemu SQL Server 2014. Rozwiń węzeł *Management* i kliknij prawym przyciskiem myszy węzeł *Data Collection*. Wybierz opcję *Tasks/Configure Management Data Warehouse*, co przedstawiono na rysunku 9.10.



Rysunek 9.10. Uruchamianie opcji Tasks/Configure Management Data Warehouse

Na ekranie powitalnym kreatora konfiguracji bazy MDW kliknij przycisk *Next* (patrz rysunek 9.11).



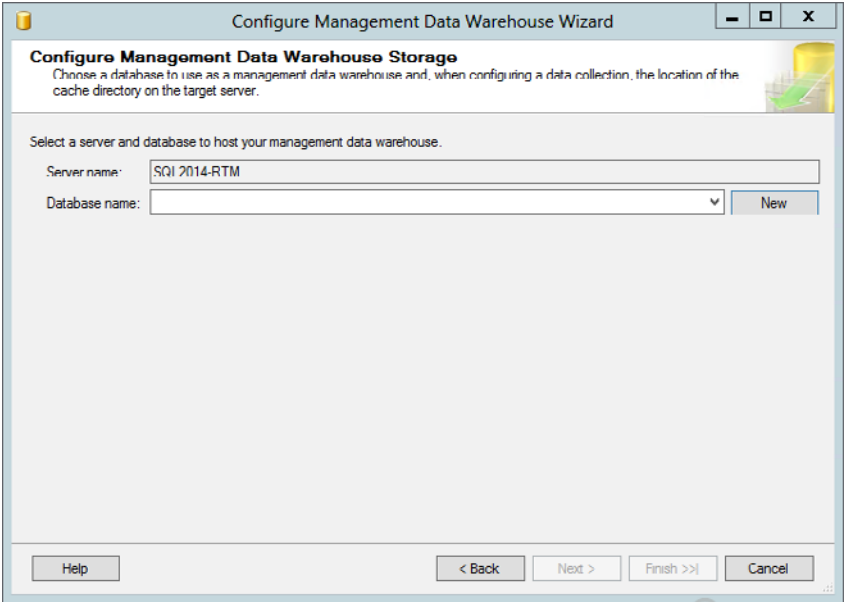
Rysunek 9.11. Ekran powitalny kreatora konfiguracji bazy MDW

Wybierz bazę danych używaną jako baza MDW. Jeśli chcesz utworzyć bazę MDW, kliknij przycisk *New* widoczny na rysunku 9.12.

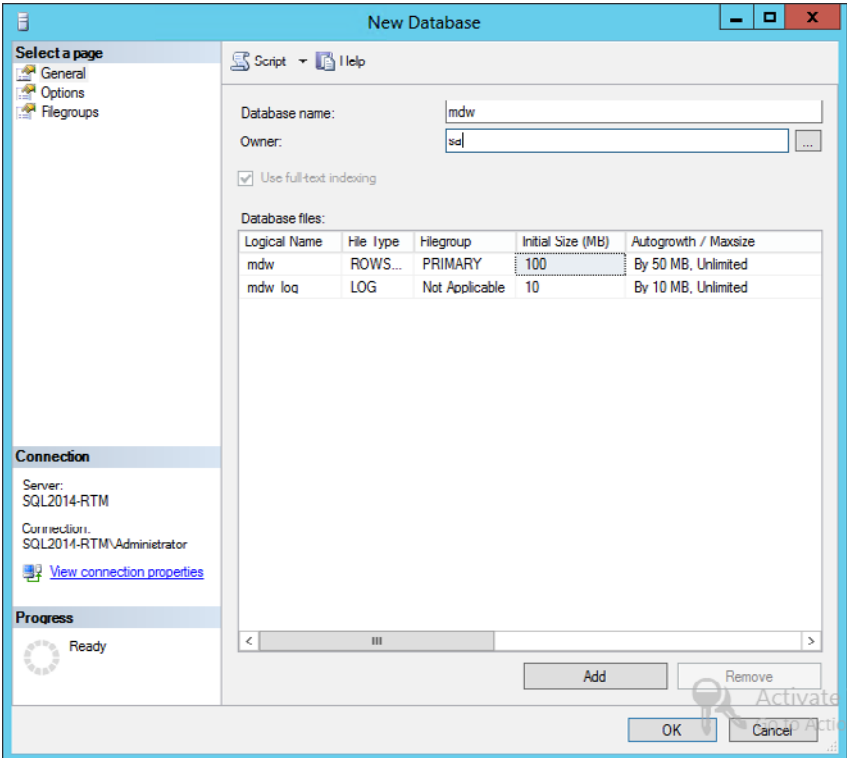
Na rysunku 9.13 pokazano ustawianie nazwy bazy MDW. W tym przykładzie nadaj bazie nazwę *mdw*.

Po dodaniu nazwy kliknij przycisk *Next* widoczny na rysunku 9.14.

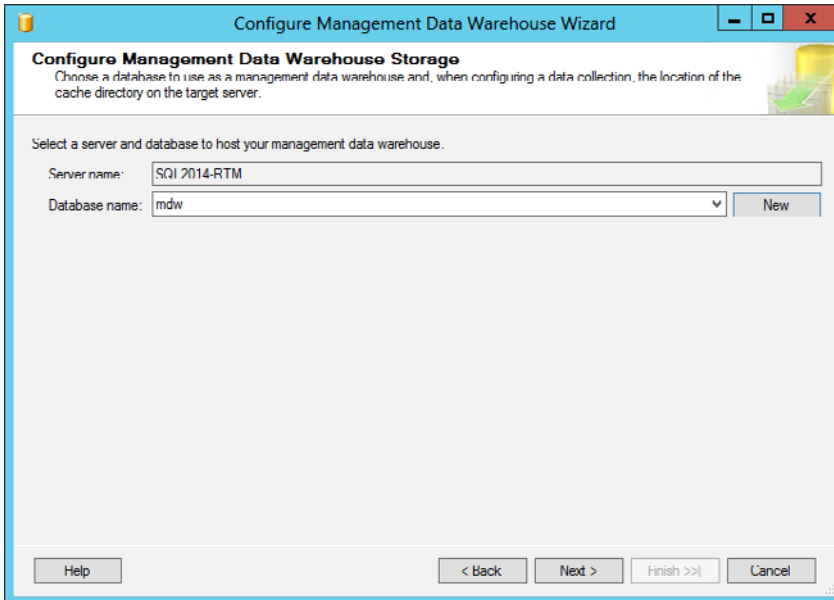
Na ekranie *Map Logins and Users* (patrz rysunek 9.15) wybierz potrzebne uprawnienia. W tym przykładzie zaznacz rolę *mdw_admin*. Następnie kliknij przycisk *Next*.



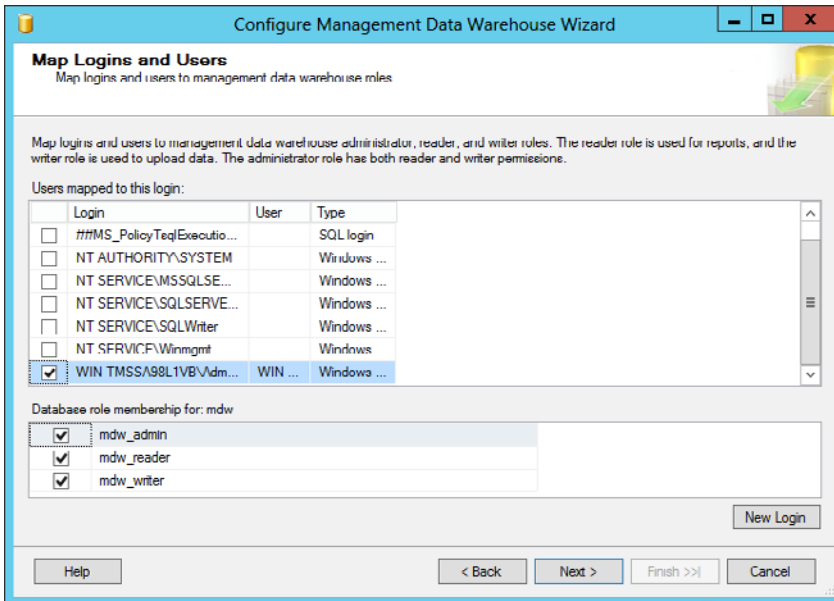
Rysunek 9.12. Wybieranie bazy MDW



Rysunek 9.13. Ustawianie nazwy bazy MDW

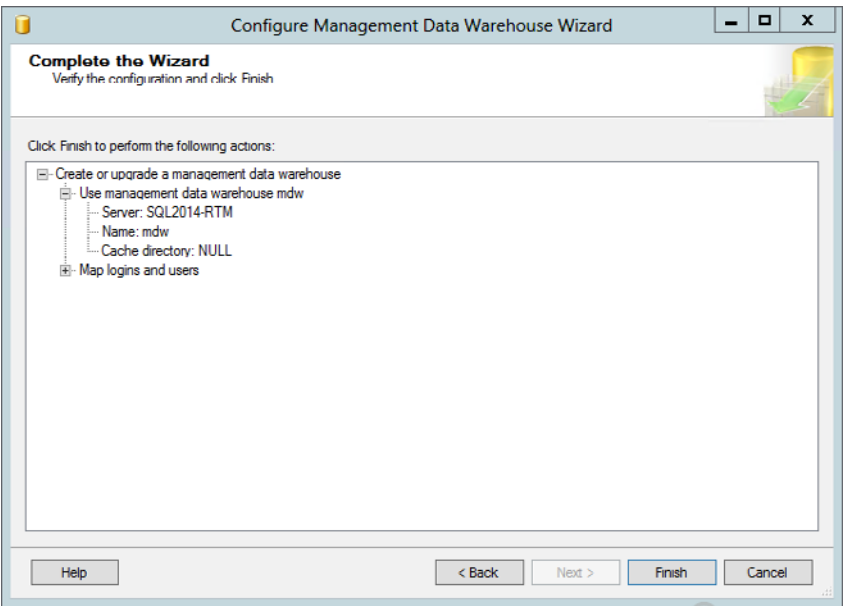


Rysunek 9.14. Nazwa bazy MDW została już ustawiona



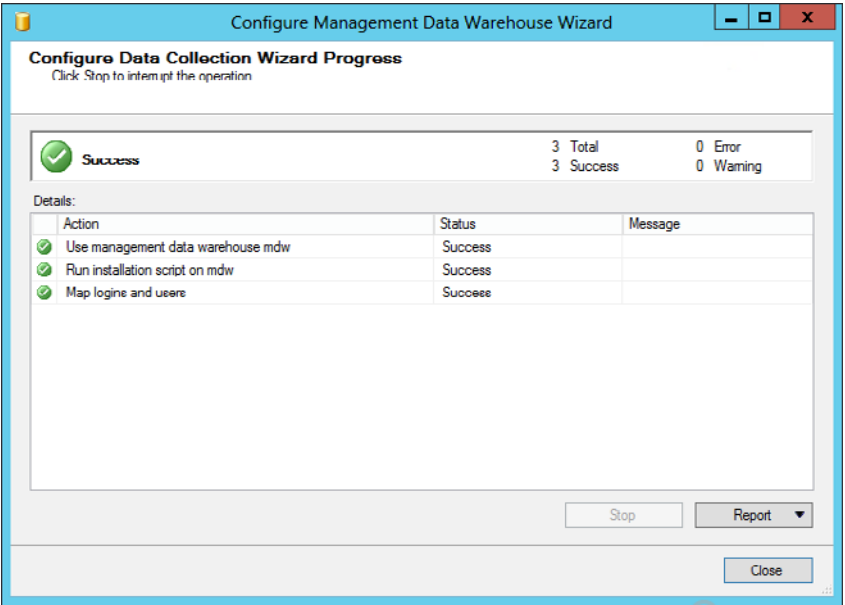
Rysunek 9.15. Ustawianie uprawnień

Powinna pojawić się strona *Complete the Wizard* widoczna na rysunku 9.16. Kliknij przycisk *Finish*.



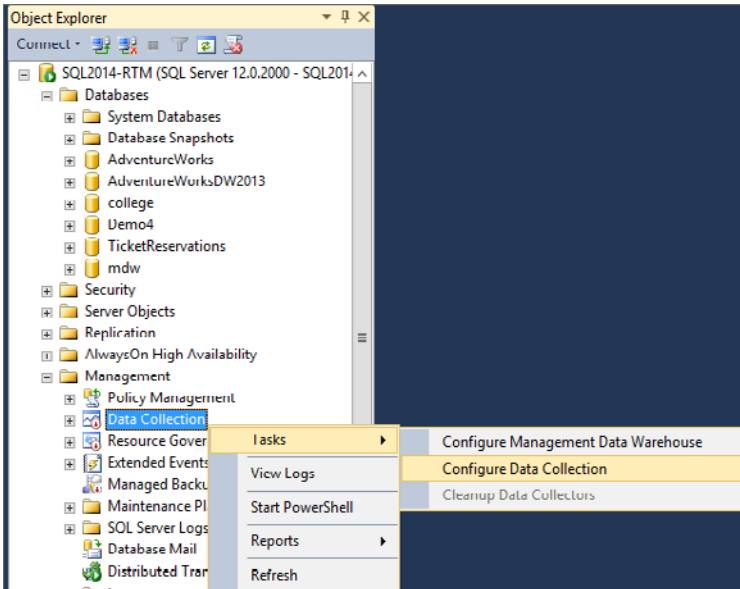
Rysunek 9.16. Końcowa strona kreatora

W czasie oczekiwania na zainstalowanie kolektorów danych możesz obserwować informacje o postępie tego procesu (patrz rysunek 9.17).



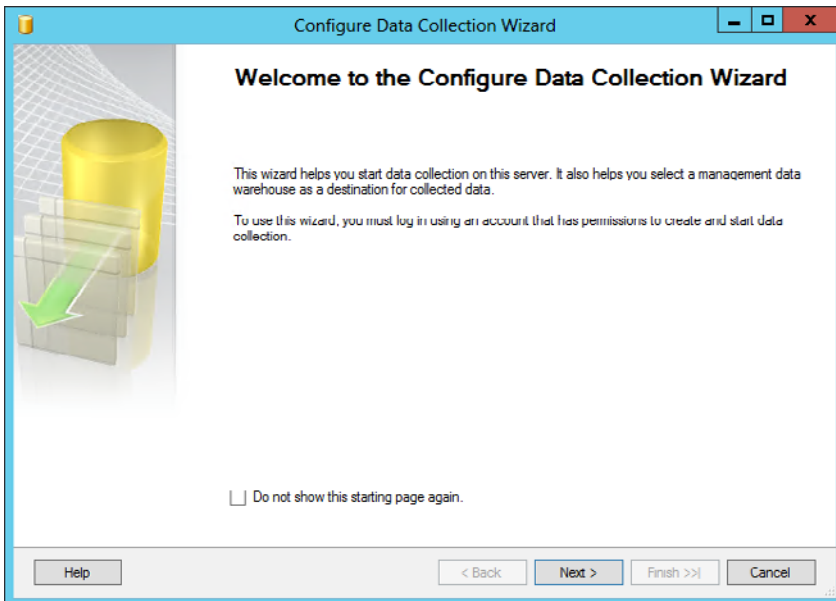
Rysunek 9.17. Informacje o postępie wykonywanych operacji

Po dodaniu bazy o odpowiednim schemacie należy skonfigurować proces zbierania danych. Ponownie rozwiń węzeł *Management* i kliknij węzeł *Data Collection* prawym przyciskiem myszy. Wybierz opcję *Tasks/Configure Data Collection*, co przedstawiono na rysunku 9.18.



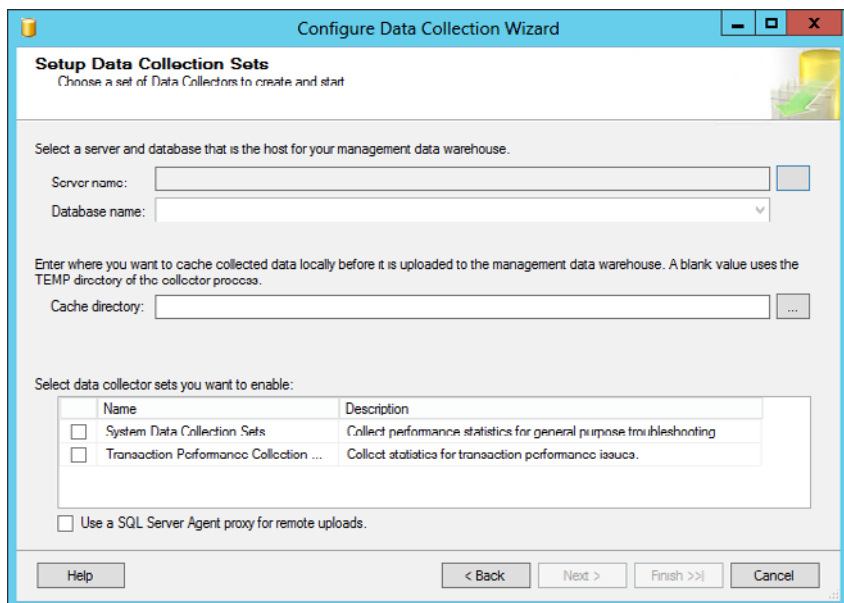
Rysunek 9.18. Uruchamianie opcji Tasks/Configure Data Collection

Na ekranie powitalnym widocznym na rysunku 9.19 kliknij przycisk *Next*.



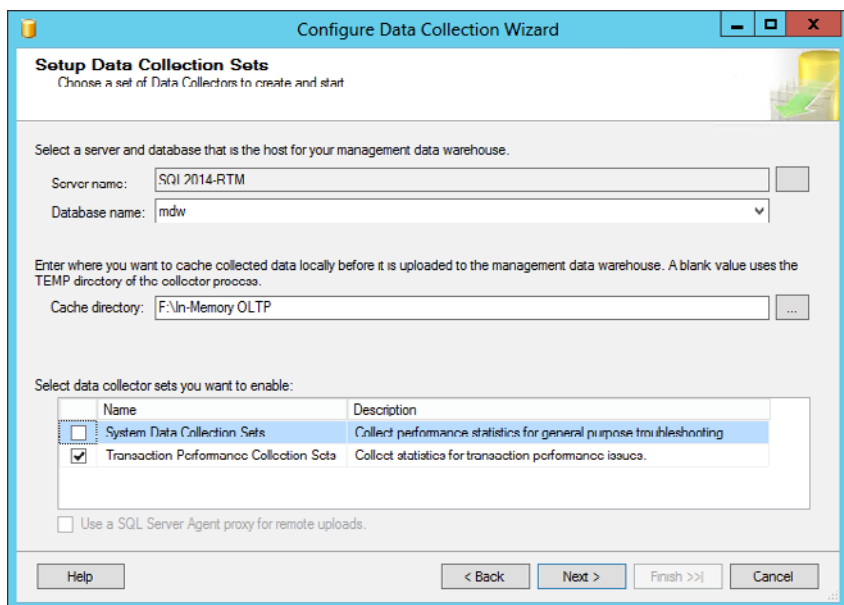
Rysunek 9.19. Ekran powitalny kreatora konfiguracji procesu zbierania danych

Należy wskazać egzemplarz systemu SQL Server, w którym znajdzie się baza MDW z danymi zebranymi przez kolektory. Na rysunku 9.20 przedstawiono ekran, na którym można ustawić ten egzemplarz.



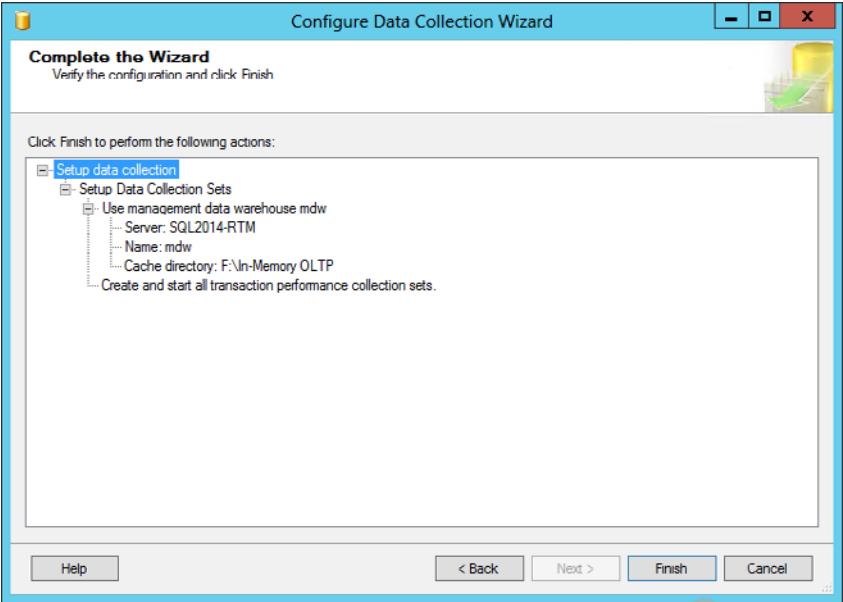
Rysunek 9.20. Wybieranie egzemplarza systemu, w którym ma działać baza MDW

Kliknij przycisk z wielokropkiem, aby nawiązać połączenie z egzemplarzem systemu SQL Server i wybrać utworzoną wcześniej bazę MDW. Ustaw katalog na pamięć podręczną używaną przez kwerendy monitorujące korzystanie z tabeli. Zaznacz też kolektor danych Transaction Performance Collection Sets (patrz rysunek 9.21). Następnie kliknij przycisk *Next*.



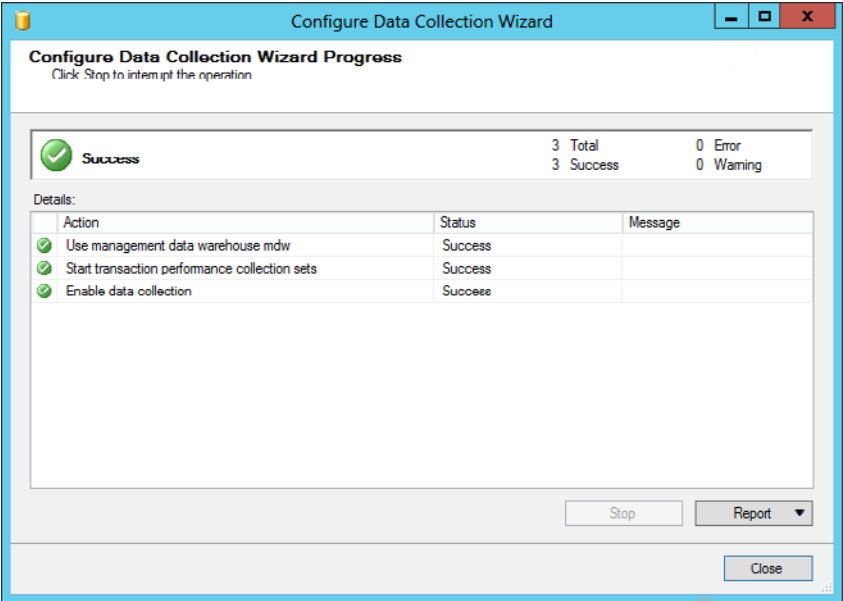
Rysunek 9.21. Konfigurowanie procesu zbierania danych

Powinna pojawić się strona *Complete the Wizard* widoczna na rysunku 9.22. Kliknij przycisk *Finish*.



Rysunek 9.22. Końcowa strona kreatora

W trakcie oczekiwania na zainstalowanie kolektorów danych możesz obserwować informacje o postępie tej operacji (patrz rysunek 9.23). Po zakończeniu instalacji kolektorów kliknij przycisk *Close*.

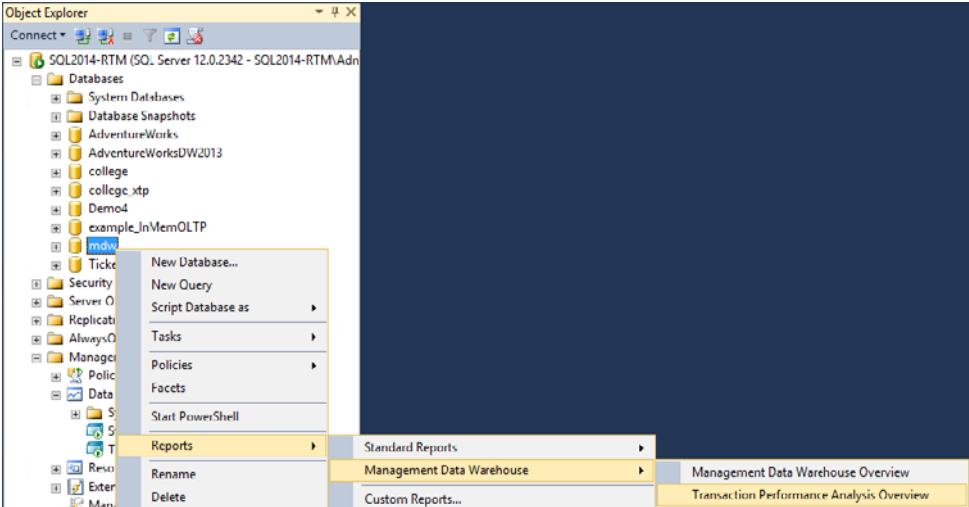


Rysunek 9.23. Informacje o zakończeniu instalacji kolektorów danych

Standardowo dane są zbierane godzinę przed przesłaniem plików z pamięci podręcznej do analizy. Po zebraniu danych powinieneś rozpocząć analizę raportów przekazanych przez kolektory.

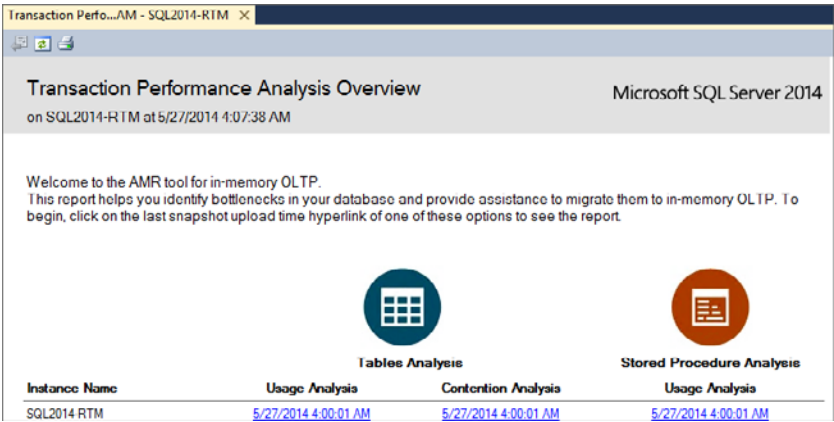
UWAGA Kolektory danych można zapisać w formie skryptów i zainstalować w wersjach SQL Server 2008 i nowszych, aby znaleźć bazy, które warto zastosować w technologii In-Memory OLTP. Wymaga to jednak zastosowania języka i funkcji specyficznych dla wersji SQL Server 2014. To spowoduje błąd przy instalacji skryptów w wersjach systemu SQL Server starszych niż 2014. Dlatego skrypty przed zainstalowaniem kolektorów trzeba zmodyfikować.

Rozwiń węzeł baz danych i kliknij prawym przyciskiem myszy utworzoną wcześniej bazę mdw. Wybierz opcję *Reports/Management Data Warehouse/Transaction Performance Analysis Overview*, co przedstawiono na rysunku 9.24.



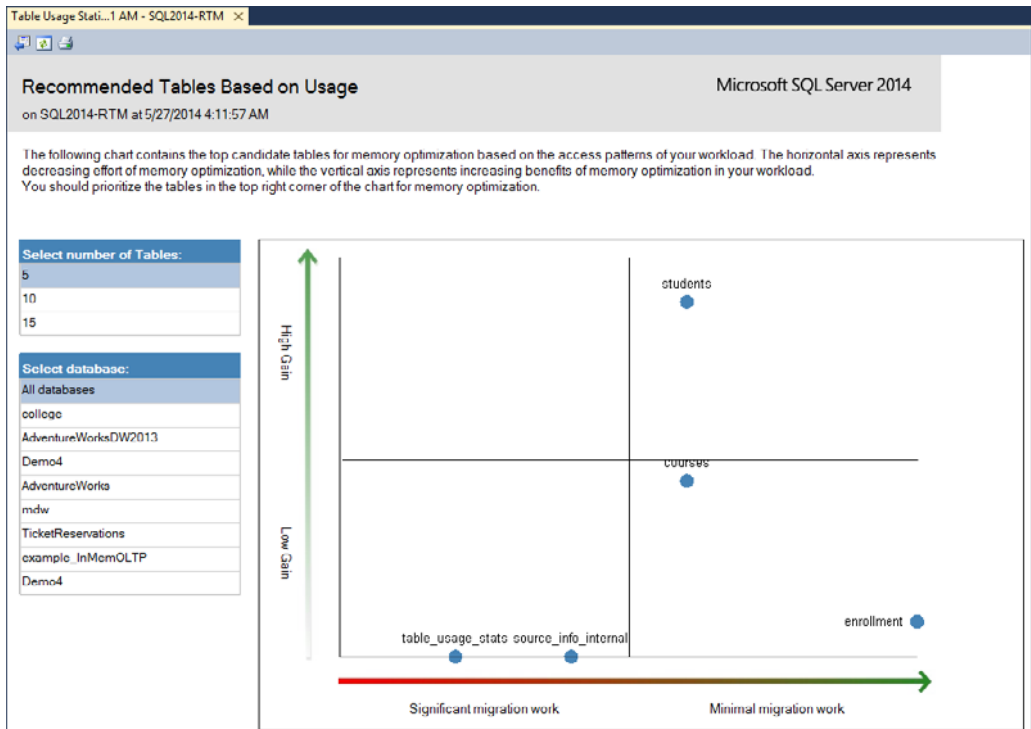
Rysunek 9.24. Otwieranie ekranu Transaction Performance Analysis Overview

To spowoduje otwarcie ekranu *Transaction Performance Analysis Overview*. W tym miejscu można wybrać wyświetlane raporty dotyczące używanego systemu. W tym ćwiczeniu zacznij od raportu *Usage Analysis* (ta opcja jest widoczna na rysunku 9.25).



Rysunek 9.25. Wybieranie wyświetlanych raportów

Jak widać na rysunku 9.26, raport *Usage Analysis* rekomenduje tabele do użycia w technologii In-Memory OLTP na podstawie ich użytkowania. Narzędzie zbiera dane dla poszczególnych baz. Domyślnie początkowo wyświetlanych jest tylko pięć tabel, jednak można zwiększyć tę liczbę za pomocą opcji widocznych na lewo od wykresu. Domyślnie uwzględniane są wszystkie bazy (ustawienie *All databases*). Aby zobaczyć ocenę tabel z wybranej bazy systemu SQL Server 2014, kliknij tę bazę.

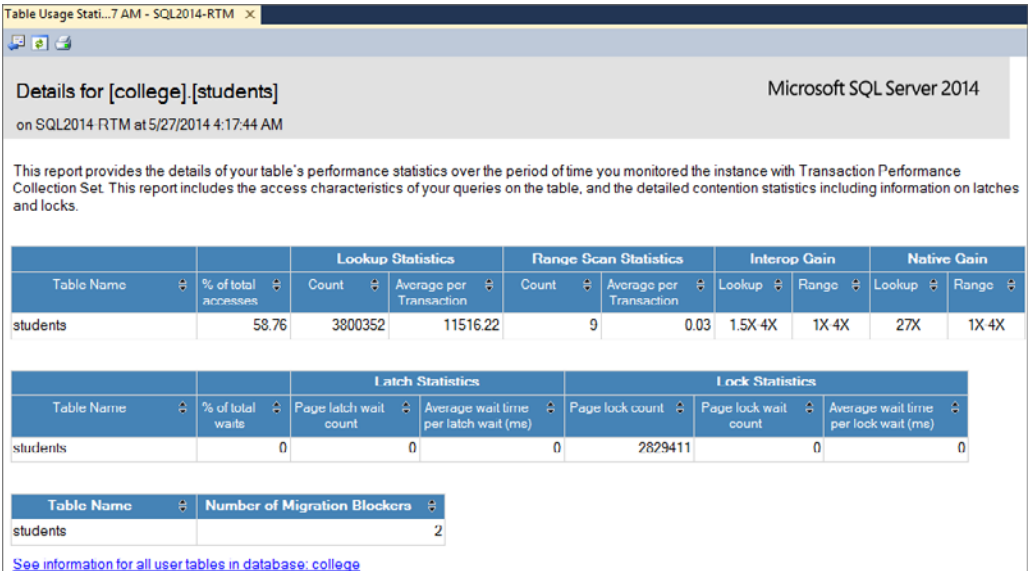


Rysunek 9.26. Raport Usage Analysis

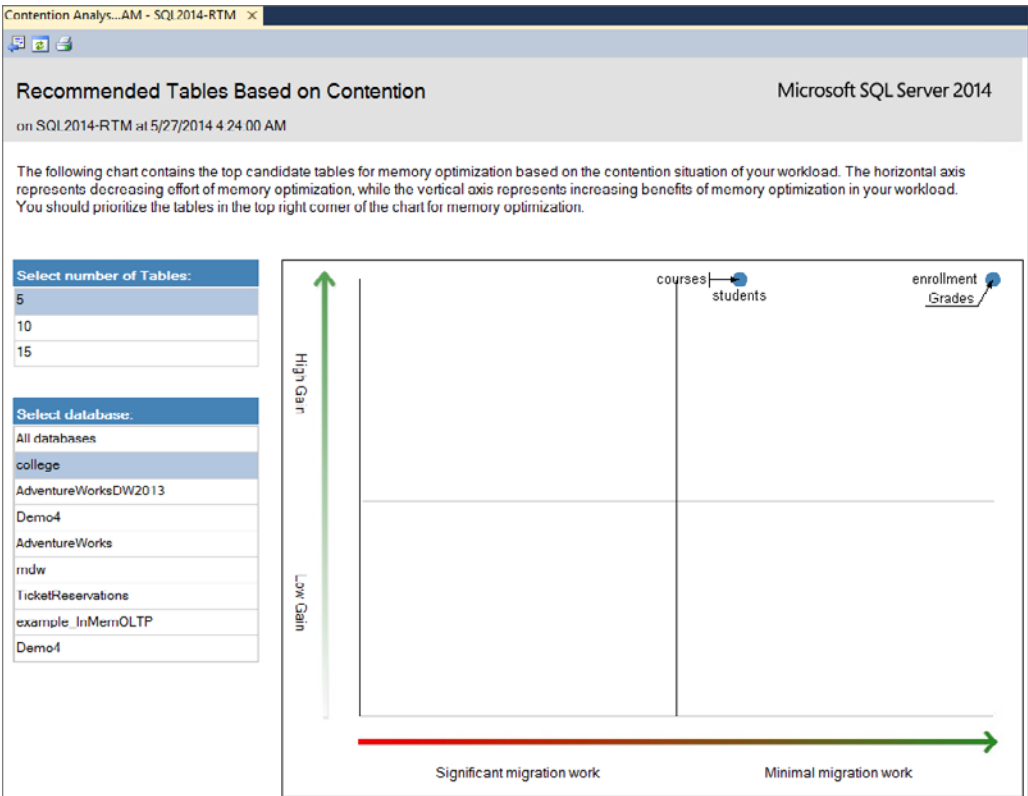
Jak widać, nie każda tabela nadaje się do stosowania w technologii In-Memory OLTP. Niektóre są używane w taki sposób, że zyski w wydajności byłyby niewielkie. Część tabel bardzo trudno stosować w omawianej technologii z powodu dużej liczby kluczy zewnętrznych, niekompatybilnych typów danych lub korzystania z określonych funkcji. Jednak inne tabele wymagają tylko niewielkiego nakładu pracy przy przenoszeniu do In-Memory OLTP, a na podstawie wzorców ich użytkowania można spodziewać się znacznego wzrostu wydajności.

Jeśli klikniesz tabelę *students*, zobaczysz szczegółowe informacje na jej temat, przedstawione na rysunku 9.27. Widoczne są tu statystyki na temat wyszukiwania, skanowania przedziałów, zamków i blokad. Dostępne są też informacje na temat zysków w wydajności poszczególnych operacji.

Po powrocie do ekranu *Transaction Performance Analysis Overview* powinieneś zapoznać się z raportem *Contention Analysis*, przedstawionym na rysunku 9.28. Raport przedstawia tabele, które najlepiej nadają się do przechowywania w pamięci. Lista ta jest tworzona na podstawie bieżącego obciążenia.



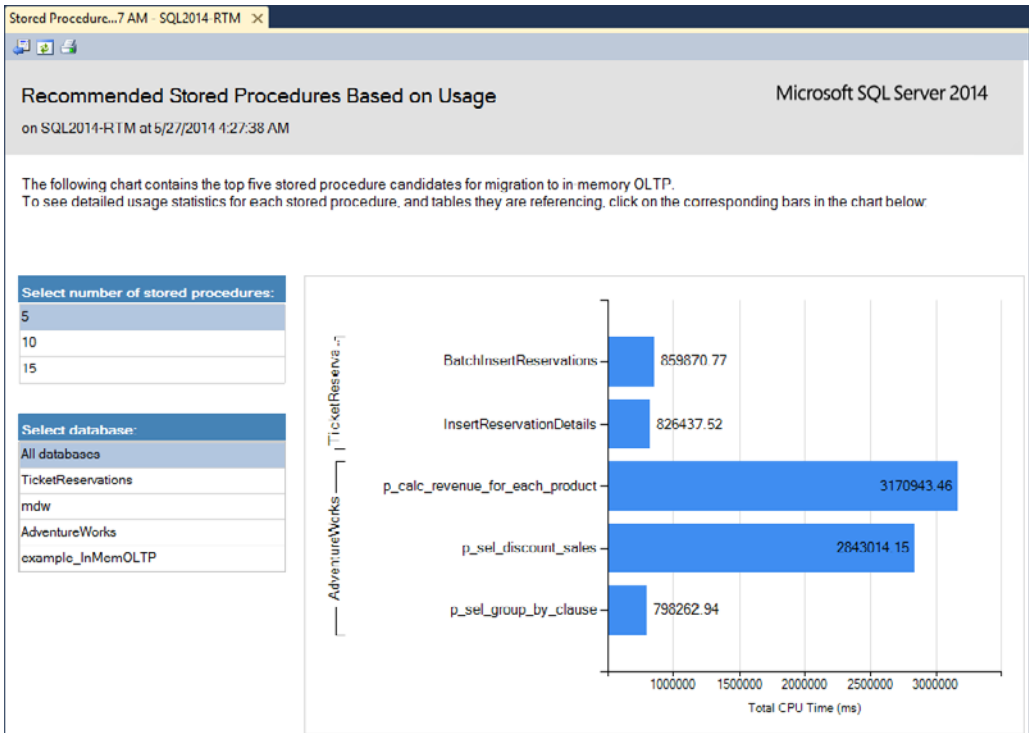
Rysunek 9.27. Szczegółowe informacje na temat tabeli



Rysunek 9.28. Report Contention Analysis

Gdy klikniesz jedną z tabel, zobaczysz raport pokazany już na rysunku 9.27, zawierający szczegółowe informacje na temat danej tabeli, a także szacowane zyski z przeniesienia jej do technologii In-Memory OLTP.

Teraz wróć do ekranu *Transaction Performance Analysis Overview* i przyjrzyj się raportowi *Stored Procedure Based on Usage Analysis*, przedstawionemu na rysunku 9.29. Raport domyślnie wyświetla pięć pierwszych procedur składowanych, które nadają się do natywnej kompilacji. Możesz w dowolnym momencie rozszerzyć tę listę do piętnastu procedur. Ponadto możesz wskazać określoną bazę danych, aby zobaczyć listę procedur składowanych tylko z tej bazy.






Rysunek 9.29. Raport *Stored Procedure Based on Usage Analysis*

W bazie AdventureWorks znajduje się procedura `p_calc_revenue_for_each_product`. Procedura ta według raportu dobrze nadaje się do kompilacji. Jeśli klikniesz daną procedurę składowaną, zobaczysz szczegółowy raport przedstawiony na rysunku 9.30.

Widac tu czas ostatniego zapisania procedury w pamięci podręcznej, łączny czas używania procesora (w milisekundach), łączny czas działania (też w milisekundach), liczbę przypadków nieznaalezienia danych w pamięci podręcznej i liczbę uruchomień. Widoczne są także schematy tabel używanych przez daną procedurę.

Stored Procedure...2 AM - SQL2014-RTM X

Details for [AdventureWorks].[p_calc_revenue_for_each_product]

Microsoft SQL Server 2014

on SQL2014-RTM at 5/27/2014 6:42:40 AM

In-memory OLTP improves CPU utilization of queries by reducing the number of instructions used for query operations, and by eliminating most I/O waits and lock/latch contention.

To see information about tables being referenced from this stored procedure, click on their names in the table below.

Stored Procedure's Execution Statistics

Cached Time	Total CPU Time (ms)	Total Execution Time (ms)	Total Cache Missed	Execution Count
5/18/2014 10:18:50 PM	3170943.46	3401906.37	0	4319

Stored Procedure's Table References:

Referenced Server	Referenced Database	Referenced Schema	Referenced Table
SQL 2014-RTM			
	AdventureWorks		
		Production	Product
		Sales	SalesOrderDetail

Rysunek 9.30. Szczegółowe informacje na temat procedury składowanej

Podsumowanie

W systemie SQL Server 2014 wprowadzono jedną z największych zmian, będącą usprawnieniem w zakresie wydajności, w historii tego systemu. W odróżnieniu od większości konkurencyjnych produktów (takich jak SAP Hannah lub Oracle Times 10) nie trzeba kupować drogiego sprzętu ani przenosić kodu i bazy danych na nową platformę. Ponadto można nadal korzystać z zaawansowanych funkcji dostępnych w edycji SQL Server Enterprise Edition.

Wydajność technologii In-Memory OLTP zależy od sytuacji, a to rozwiązanie nadaje się tylko dla wybranych fragmentów kodu i obiektów bazy danych. Szukając informacji o tej technologii w witrynie Microsoft Case Studies (<http://www.microsoft.com/casestudies/>), można znaleźć studia przypadków dotyczące firm, w których dzięki In-Memory OLTP wydajność systemu wzrosła od 9 do nawet 150 razy. Wiele z opisanych implementacji przeprowadzono bez konieczności wymiany sprzętu.

Pamiętaj, by ocenić sposób korzystania z tabel, oszacować ilość pamięci potrzebnej na tabele i indeksy, a także użyć narzędzia ARM do pomocy przy zbieraniu i analizowaniu danych. Zawsze sprawdzaj uzyskane wyniki w porównywalnym środowisku rozwojowym (lub używanym do kontroli jakości) przed wprowadzeniem zmian w środowisku produkcyjnym.

Z rozdziału 10. dowiesz się, jak skonfigurować system SQL Server pod kątem optymalnej wydajności. Opisano tam także sprzęt i jego wpływ na pracę systemu SQL Server.

Konfigurowanie serwera pod kątem optymalnej wydajności

ZAWARTOŚĆ ROZDZIAŁU:

- Definiowanie wysokiej wydajności.
- Co każdy administrator baz danych powinien wiedzieć o wydajności?
- Konfigurowanie sprzętu serwera.
- Szczegółowe omówienie konfiguracji procesora.
- Omówienie konfiguracji pamięci i opcji z tego obszaru.
- Projekt i opcje dotyczące operacji wejścia-wyjścia.

W branży informatycznej za bazy danych i systemy, w których te bazy się znajdują, odpowiadają osoby pełniące wiele różnych ról. *Administrator programista* odpowiada głównie za projektowanie bazy, a także za pisanie kodu (kwerend, procedur składowanych itd.). *Administrator produkcyjnej bazy danych* zajmuje się przede wszystkim konfiguracją, konserwacją i zapewnieniem dostępności bazy oraz systemu. *Administrator procesów BI* odpowiada głównie za procesy BI oparte na systemie SQL Server i powiązanych narzędziach. Jedna osoba może nawet odpowiadać za kilka tych zadań. Wtedy można ją nazwać *hybrydowym administratorem baz danych*. Ponadto niektórzy administratorzy baz danych nie posiadają formalnego wykształcenia z tego zakresu lub zajmują się serwerami z konieczności, ponieważ brakuje odpowiednich pracowników. Ich można określić mianem *okazyjnych administratorów baz danych*. Osoby te zwykle pełnią wiele obowiązków i z powodu braku zasobów muszą szybko wdrożyć się do pracy. Szczegółowe omówienie różnych profesjonalistów zajmujących się bazami danych znajdziesz w rozdziale 1., „Architektura systemu SQL Server 2014”.

Administrator programista musi wiedzieć, jak optymalizować wydajność systemu, aby projektowane rozwiązania wykorzystywały potencjał sprzętu. Osoby takie muszą zadbać o wydajną pracę systemów także po wprowadzeniu w nich zmian (które są nieuniknione). Proces rozrastania się systemów (ze względu na ilość danych, liczbę użytkowników i nowe funkcje) powinien przebiegać w taki sposób, by cały czas działały one optymalnie.

Także administrator produkcyjnych baz danych musi rozumieć kwestie związane z wydajnością, aby zarządzany przez niego system działał poprawnie zarówno na początku, jak i w całym cyklu życia. Należy przy tym uwzględnić kilka czynników, takich jak poprawna konfiguracja serwera, śledzenie pracy systemu w momencie uruchomienia, wdrożenie kompletnej strategii monitorowania pozwalającej zapewnić optymalne działanie systemu itd.

Poniżej wymieniono najważniejsze aspekty związane z zapewnianiem wysokiej wydajności w zakresie skalowalności, czasu reakcji, niezawodności i używalności. Oto one.

- Zmiana konfiguracji zasilania ze zrównoważonej na wysoką wydajność.
- Włączenie funkcji Instant File Initialization w celu zminimalizowania liczby operacji wejścia-wyjścia na sekundę w trakcie powiększania plików.
- Zapoznanie się z opcjami śledzenia udostępnianymi przez Microsoft (opcje umożliwiają stosowanie domyślnie wyłączonych funkcji systemu SQL Server).
- Ustalenie możliwości systemu w zakresie szybkości procesora, pamięci i operacji wejścia-wyjścia.
- Wykrycie wąskich gardeł i ustalenie, jak rozwiązać te problemy.

W tym rozdziale omawiamy wszystkie wymienione kwestie oraz odpowiadamy na najważniejsze pytania dotyczące wydajności i konfigurowania serwera.

Co każdy administrator baz danych powinien wiedzieć na temat wydajności?

W tym rozdziale znajdziesz wiele rekomendacji, które umożliwią poprawienie wydajności systemu. Jednak przyspieszenie transakcji o kilka milisekund nie zawsze jest warte Twojego czasu i pieniędzy wydawanych na sprzęt. Często dobry początkowy plan jest wart więcej niż późniejsze pomysłowe optymalizacje. W kontekście wydajności zawsze pamiętaj o trzech rzeczach:

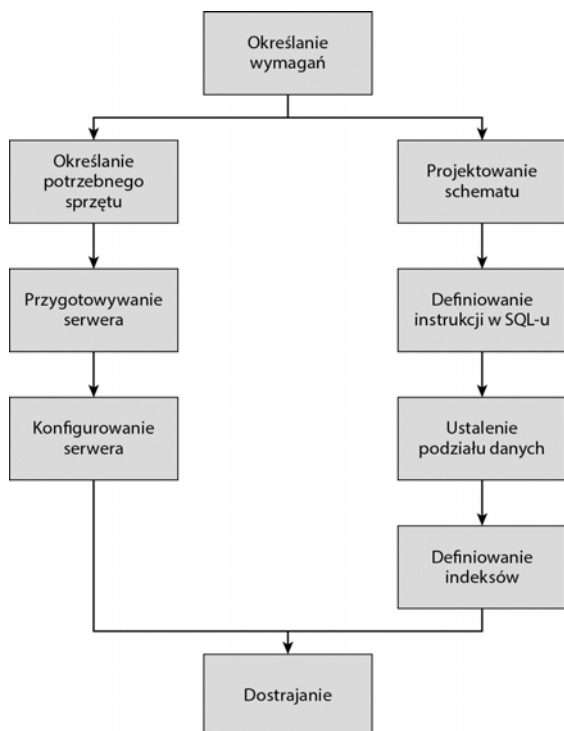
- cyklu dostrajania wydajności,
- definiowaniu dobrej wydajności,
- koncentrowaniu się na tym, co najważniejsze.

Cykl dostrajania wydajności

Zbyt często prace nad wydajnością i optymalizacją zostawia się na sam koniec. *Dostrajanie wydajności* to proces iteracyjny i najlepiej go zacząć już na początku projektowania systemu. Zapewnianie wysokiej wydajności rozpoczyna się od właściwego skonfigurowania serwera, po czym należy zaprojektować wydajny schemat, opracować szybkie instrukcje w SQL-u i odpowiednio dobrać indeksy. Monitorowanie i analiza wydajności zapewniają informacje, dzięki którym można wprowadzić zmiany w konfiguracji serwera, projekcie schematu i innych elementach.

Istnieje wiele narzędzi niezależnych producentów, które wspomagają monitorowanie egzemplarzy systemu SQL Server i środowiska. W samym systemie Microsoft SQL Server dostępne są bazy MDW (więcej informacji na ich temat podajemy w rozdziale 12., „Monitorowanie systemu SQL Server”). Korzystanie z narzędzi do monitorowania skryptów to ważne elementy procesu dostrajania wydajności.

Na rysunku 10.1 przedstawiono plan prac projektowych.



Rysunek 10.1. Prace projektowe

Przy tworzeniu nowych aplikacji nie istnieje system, w którym można przeprowadzić pomiary. W najlepszej sytuacji dostępne są wskaźniki oparte na obecnej grupie użytkowników lub prognozy zarządu określające, kim będą użytkownicy, co będą robili i jak wpłynie to na rozwijaną aplikację.

Jeśli system już istnieje i chcesz przenieść go na nowy serwer lub rozbudować, możesz wykonać pomiary wykorzystania zasobów systemowych i użyć wyników jako punktu wyjścia. Następnie możesz uwzględnić informacje na temat nowych funkcji. Odpowiedz przy tym na następujące pytania.

- Czy nastąpi wzrost liczby użytkowników?
- Czy zwiększy się obciążenie serwera?
- Czy zmieni się ilość danych?

Informacje te pomogą oszacować wpływ wprowadzenia nowego systemu na zasoby. Przed zaimplementowaniem nowego systemu, w trakcie testów, możesz zacząć porównywać szacunki z rzeczywistym zapotrzebowaniem na zasoby i wydajnością serwerów testowych.

Konfiguracja

Domyślna konfiguracja systemu SQL Server nie jest optymalna. Przed zainstalowaniem systemu i po zakończeniu tego procesu powinieneś wprowadzić zmiany w ustawieniach zasilania i zasadach grup, skonfigurować bazę tempdb, a także włączyć potrzebne opcje śledzenia. Kroki te wymagają zastosowania zaawansowanych opcji konfiguracyjnych wykraczających poza zakres rozdziału 2., „Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014”.

Plan zasilania

Od wersji Windows 2003 w systemie Windows dostępne są plany zasilania. *Plan zasilania* określa, jak system operacyjny ma korzystać z energii. Każdy system operacyjny Windows ma trzy domyślne plany zasilania. Ponadto można tworzyć i instalować niestandardowe plany zasilania. Oto trzy domyślne plany.

Zrównoważony. W tym planie zasilanie jest dostosowane do potrzeb. Moc obliczeniowa procesora jest wykorzystywana w stopniu zależnym od ogólnego obciążenia powodowanego przez aplikację. Plan ten ma zapewniać wydajne zużycie pamięci przy minimalnym wpływie na wydajność.

Wysoka wydajność. To ustawienie zwiększa wydajność procesora i innego sprzętu serwera kosztem wyższego zużycia energii.

Oszczędzanie energii. Ten plan powoduje ograniczenie wydajności w celu zmniejszenia zużycia energii i kosztów pracy serwera. W odróżnieniu od poprzednich plan ten ogranicza szybkość procesora do określonego procentu jego możliwości.

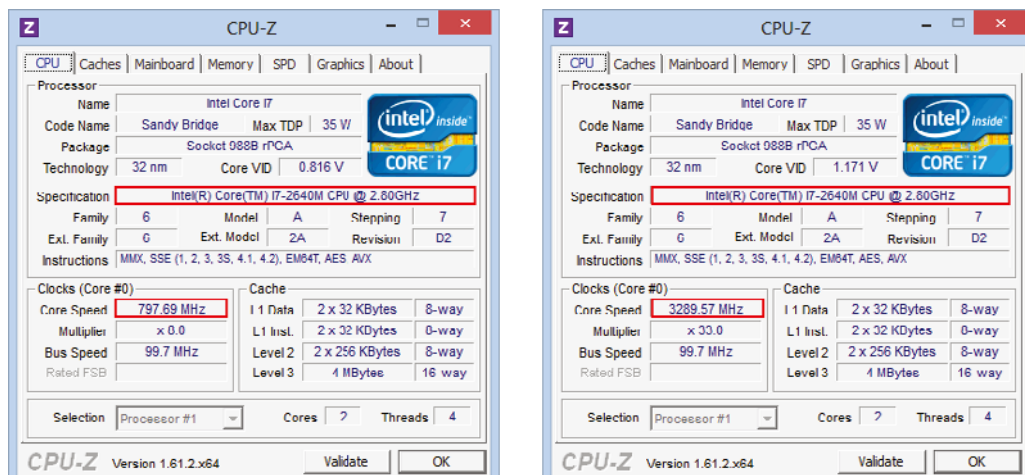
Od systemu Windows 2008 (i Windows Server 2012 R2) domyślnym planem zasilania jest Zrównoważony. Nie jest to korzystne dla systemu SQL Server. W opisie tego ustawienia znalazła się wzmianka o *minimalnym wpływie na wydajność*. Wszystko, co wpływa na wydajność systemu SQL Server, jest niekorzystne, dlatego powinieneś włączyć plan Wysoka wydajność.

Pozostawienie planu Zrównoważony może prowadzić do problemu *parkowania rdzeni* (ang. *core parking*). Polega to na tym, że niektóre rdzenie procesora są wyłączane. Problem może też wystąpić niezależnie od niskiej wydajności związanej z planem zasilania ustawionym na Zrównoważony.

Powinieneś zmierzyć wykorzystanie poszczególnych procesorów. Aby zbadać efekt niewłaściwie dobranego planu zasilania, wykorzystaj bezpłatne narzędzie CPU-Z. Możesz je pobrać ze strony <http://www.cpuid.com/softwares/cpu-z.html>.

Okno widoczne po lewej stronie rysunku 10.2 pochodzi z serwera z włączonym planem Zrównoważony. Jak widać, mamy tu ograniczenie szybkości procesora o wydajności 2,8 GHz do poziomu 0,798 GHz, co oznacza spadek o prawie 70%. Okno widoczne po prawej stronie dotyczy tego samego serwera po włączeniu planu Wysoka wydajność. Widać, że rdzenie komputera działają teraz z maksymalną wydajnością.

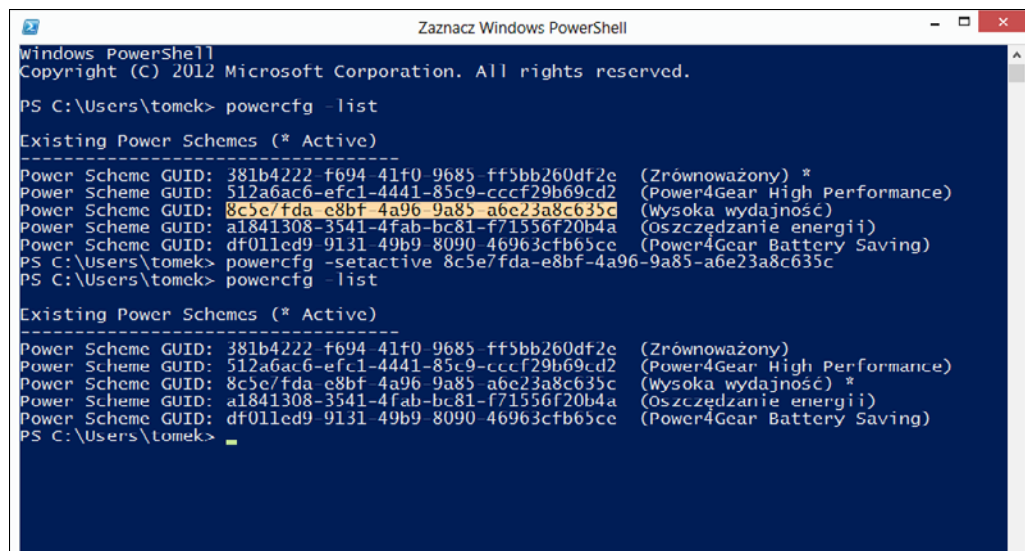
UWAGA Od systemu Windows Server 2012 parkowanie rdzeni jest domyślnie wyłączone, choć nadal trzeba zmienić ustawienia zasilania, aby w pełni wykorzystać możliwości procesora. Jednak od tej wersji nie trzeba się już przejmować parkowaniem rdzeni. Szczegółowe omówienie parkowania rdzeni i wykrywania tego zjawiska za pomocą monitora zasobów znajdziesz w artykule 281479 z bazy Microsoft Knowledge Base — <http://support.microsoft.com/kb/2814791>.



Rysunek 10.2. Wydajność procesora przy włączonych planach zasilania Zrównoważony (po lewej) i Wysoka wydajność (po prawej)

Aby włączyć odpowiednie ustawienia, najpierw należy wejść do BIOS-u serwera i umożliwić w nim zarządzanie zasilaniem przez system operacyjny. Jeśli nie zmienisz ustawień w BIOS-ie, system operacyjny nawet po wybraniu planu Wysoka wydajność nadal będzie działał zgodnie z planem Zrównoważony. Aby wprowadzić zmiany w BIOS-ie, należy ponownie uruchomić serwer i wejść do menu BIOS-a. Związane jest to z przestojem w pracy serwera.

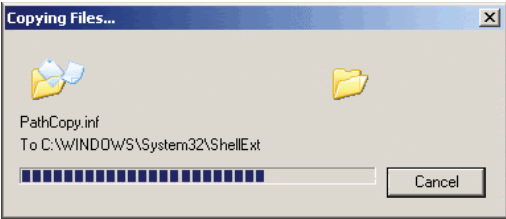
Po zmianie ustawień w BIOS-ie trzeba skonfigurować system operacyjny. W tym celu otwórz powłokę PowerShell na serwerze. Na rysunku 10.3 przedstawiono polecenia, które należy wpisać. Aby sprawdzić, który plan zasilania jest używany, wpisz instrukcję **powercfg -list**. Przy aktywnym planie widoczna jest gwiazdka (*). Aby zmienić plan, wprowadź polecenie **powercfg -setactive** i podaj identyfikator GUID planu zasilania, który chcesz zastosować.



Rysunek 10.3. Ustawianie planu zasilania w oknie powłoki PowerShell

Natychmiastowe inicjowanie plików bazy danych

Gdy kopiujesz duży plik na serwer plików lub między komputerami, czasem pojawia się okno, w którym widoczne są dokumenty przepływające między teczkami. W zależności od wielkości pliku ten proces może trwać przez pewien czas, a dopiero potem pojawiają się informacje o tym, jaka część pliku została już przesłana i kiedy proces kopiowania zostanie zakończony. Na rysunku 10.4 przedstawiono przykładową operację tego rodzaju.



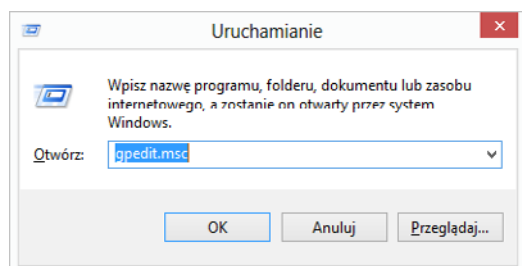
Rysunek 10.4. Kopiowanie dużego pliku

W czasie przed wyświetleniem informacji na temat kopiowanego pliku system operacyjny Windows zeruje docelowy plik. Proces polega na zainicjowaniu zerem każdego bitu danych w docelowym pliku. Dopiero potem rozpoczyna się kopiowanie. W systemie SQL Server można pominąć etap inicjowania zerami plików danych oraz kopii zapasowych bazy danych i dzienników. W tabeli 10.1 pokazano, że w ten sposób można poprawić wydajność systemu.

Tabela 10.1. Przykładowe wyniki operacji z natychmiastowym inicjowaniem plików bazy danych

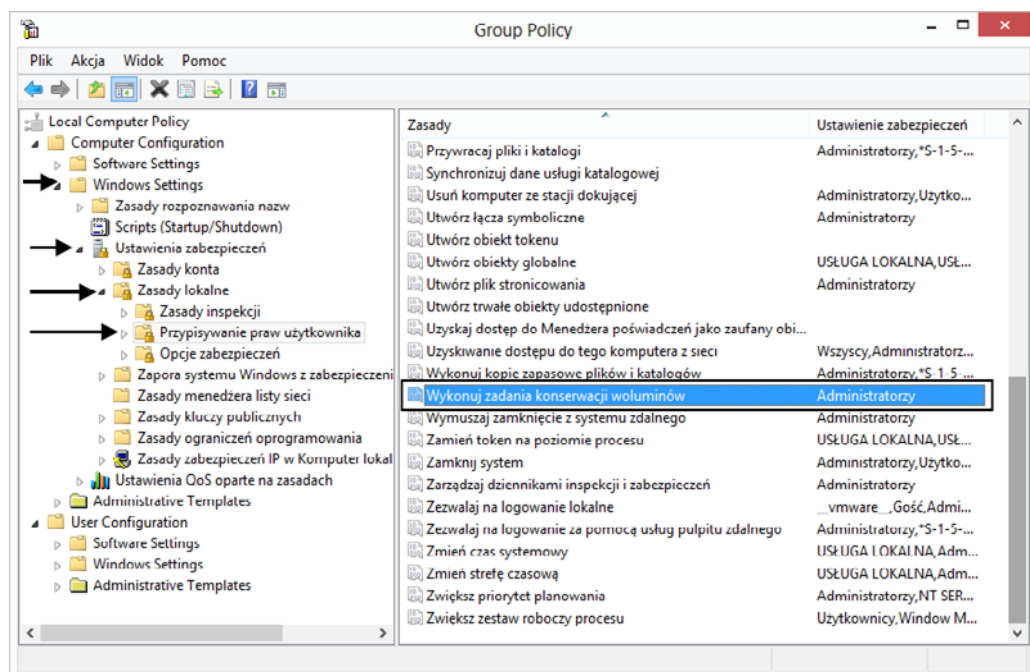
Testy wydajności bez natychmiastowego inicjowania plików bazy danych	Wyniki (minuty:sekundy)
Tworzenie bazy danych (plik danych równy 20 gigabajtom)	11:57
Zmiana wielkości bazy o 10 gigabajtów	5:59
Tworzenie kopii zapasowej pustej bazy o pojemności 30 gigabajtów	00:17
Przywracanie pustej bazy o pojemności 30 gigabajtów	17:46
Przywracanie bazy o pojemności 30 gigabajtów z 12 gigabajtami danych	25:07
Tworzenie kopii zapasowej bazy o pojemności 30 gigabajtów z 12 gigabajtami danych	17:21
Testy wydajności z natychmiastowym inicjowaniem plików bazy danych	Wyniki (minuty:sekundy)
Tworzenie bazy danych (plik danych równy 20 gigabajtom)	00:12
Zmiana wielkości bazy o 10 gigabajtów	00:00
Tworzenie kopii zapasowej pustej bazy o pojemności 30 gigabajtów	00:13
Przywracanie pustej bazy o pojemności 30 gigabajtów	00:07
Przywracanie bazy o pojemności 30 gigabajtów z 12 gigabajtami danych	6:50
Tworzenie kopii zapasowej bazy o pojemności 30 gigabajtów z 12 gigabajtami danych	13:43

Aby włączyć natychmiastowe inicjowanie plików, musisz ustawić konto usługowe (domeny lub lokalne) dla zasady *Wykonuj zadania konserwacji woluminów*. W tym celu otwórz program *gpedit.msc* w konsoli MMC. Najłatwiejszy sposób na wykonanie tej operacji to otwarcie okna *Uruchamianie* i wpisanie instrukcji *gpedit.msc*, co przedstawiono na rysunku 10.5.



Rysunek 10.5. Uruchamianie programu *gpedit.msc*

Rozwiń węzeł *Windows Settings/Ustawienia zabezpieczeń/Zasady lokalne/Przypisywanie praw użytkownika* przedstawiony na rysunku 10.6. Kliknij dwukrotnie *Wykonuj zadania konserwacji woluminów* i dodaj konto usługowe systemu SQL Server.



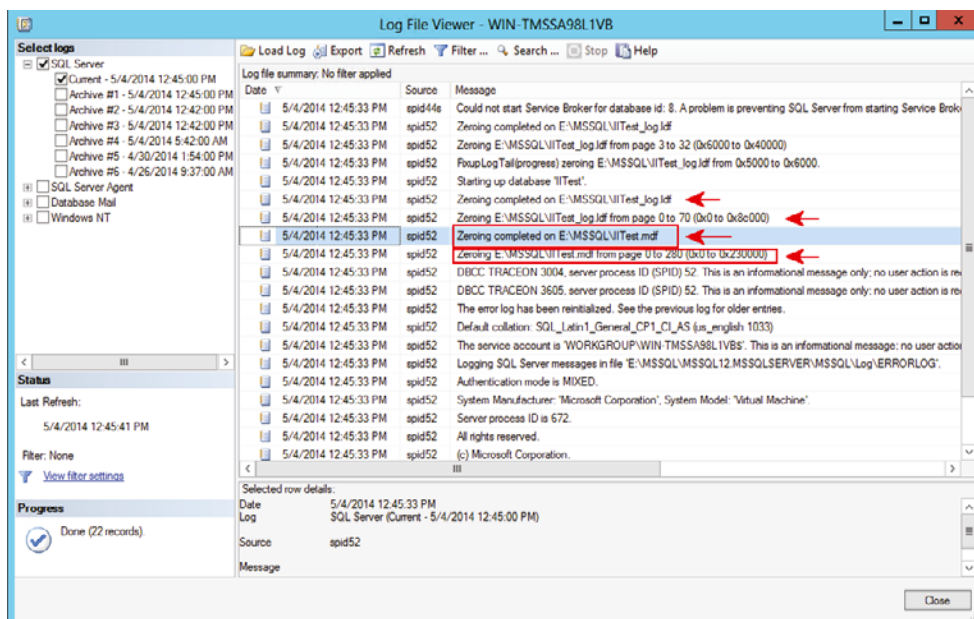
Rysunek 10.6. Dodaj odpowiednie konto do zasady *Wykonuj zadania konserwacji woluminów*

Aby sprawdzić, czy dla danej bazy włączone jest natychmiastowe inicjowanie plików, uruchom poniższą kwerendę i sprawdź zawartość dziennika błędów systemu SQL Server:

```
exec sp_cycle_errorlog
go
DBCC TRACEON(3605)
DBCC TRACEON(3004)
```

```
go
USE master
go
CREATE DATABASE IITest
DROP DATABASE IITest
go
```

Na rysunku 10.7 przedstawiono wyniki dla systemu, w którym natychmiastowe inicjowanie plików *nie jest włączone*. Opcja śledzenia 3605 włącza rejestrowanie poleceń DBCC w dzienniku błędów systemu SQL Server. Opcja śledzenia 3004 włącza rejestrowanie informacji i zmian w fizycznych plikach danych. Na rysunku 10.7 widać, że pliki *.mdf* i *.ldf* zostały wyzerowane w momencie tworzenia bazy danych. Gdyby natychmiastowe inicjowanie plików było włączone, wyzerowany zostałby tylko plik *.ldf*. Dla pliku *.mdf* ta operacja zostałaby pominięta.



Rysunek 10.7. Zawartość dziennika z informacjami o zerowaniu plików

Możesz uruchomić przedstawiony kod dla istniejącej bazy danych, aby sprawdzić, czy natychmiastowe inicjowanie plików jest w niej włączone. Jeśli ten mechanizm nie jest włączony, musisz najpierw dodać konto do wspomnianej wcześniej zasady, a następnie ponownie uruchomić usługę systemu SQL Server.

UWAGA Jeśli zasady są przekazywane na lokalne serwery z nadrzędnego serwera domeny Active Directory, zastępują ustawienia lokalne. To oznacza, że jeśli chcesz wykorzystać zasady związane z usługą Active Directory, powinieneś porozmawiać z administratorem sieci o dodaniu odpowiedniej zasady do zasad tej usługi. Ponadto natychmiastowe inicjowanie plików nie współdzieli z technologią TDE (ang. *Transparent Data Encryption*) i dziennikami transakcji. Może się zdarzyć, że bity z wcześniej usuniętych plików znajdą się w zapisanych w kopii zapasowej i przywróconych plikach *.mdf* i *.ndf*. Za pomocą narzędzi z obszaru informatyki śledczej można pobrać i odzyskać takie dane. Dlatego zawsze sprawdzaj, komu dajesz dostęp do plików danych.

Opcje śledzenia

W użytku zawsze jest wiele wersji systemu Microsoft SQL Server. Jego autorzy wciąż znajdują i naprawiają błędy oraz dodają nowe funkcje. Opcje śledzenia można dodać do systemu SQL Server za pomocą parametrów uruchomieniowych, np. `-T<numer opcji śledzenia>`, i rozdzielić przecinkami. Aby się dowiedzieć, jak ustawiać parametry uruchomieniowe, zajrzyj do rozdziału 2. Za pomocą opcji śledzenia można włączyć określone funkcje, które domyślnie są nieaktywne. Niektóre opcje śledzenia znacznie poprawiają wydajność. Ich opis zamieszczono w tabeli 10.2.

Tabela 10.2. Opcje śledzenia poprawiające wydajność

Opcja śledzenia	Definicja
-T1118	Wyłącza alokację pojedynczych stron we wszystkich bazach z systemu SQL Server. Zwykle służy do zapobiegania współzawodnictwu o strony SGAM w bazie tempdb.
-T2371	Umożliwia częstsze aktualizowanie statystyk w bazie danych. Domyślnie statystyki są aktualizowane po zmianie 20% zawartości bazy plus 500 wierszy. W małych tabelach to ustawienie sprawdza się dobrze. Jednak w dużych bazach zawierających miliony lub miliardy wierszy może spowodować, że statystyki szybko staną się nieaktualne. Ta opcja śledzenia powoduje obniżenie procentu wierszy, które muszą się zmienić, aby nastąpiła aktualizacja statystyk.
-T4199	Dodaje wszystkie poprawki wprowadzone w procesorze kwerend w pakietach Service Pack i aktualizacjach wcześniejszych wersji systemu SQL Server.
-T3226	Wyłącza nieustanne zapisywanie informacji o kopiach zapasowych w dzienniku błędów systemu SQL Server. Jeśli w danym egzemplarzu systemu działa od kilkudziesięciu do kilkuset baz danych, zobaczysz komunikat na temat każdej z nich. Jeżeli bazy pracują w trybie pełnego przywracania i zapisywane są kopie zapasowe dzienników transakcji, dziennik błędów stanie się nieczytelny.

Istnieją też inne opcje śledzenia, które można włączyć. Większość z nich ma ściśle określone zastosowania. Opcje T1222 i T1204 są używane do śledzenia i wykrywania zakleszczeń. Opcja T610 pozwala usprawnić masowe wczytywanie indeksów klastrowanych i nieklastrowanych. Jednak opcja ta może negatywnie wpływać na pracę systemu OLTP z pełnym rejestrowaniem operacji i zwykle jest stosowana w hurtowniach danych.

Każdą opcję śledzenia należy przetestować przed zastosowaniem w systemie produkcyjnym. Pozwala to sprawdzić ogólny wpływ danej opcji na system.

Definicja dobrej wydajności

Podstawowe pytanie, które każdy administrator baz danych musi sobie zadać przed przystąpieniem do usprawniania systemu, brzmi: „Czy dany system pracuje obecnie z dobrą wydajnością?”. Bez określonego poziomu docelowego lub punktu odniesienia nie da się tego stwierdzić. Aby zebrać informacje potrzebne do udzielenia odpowiedzi na przedstawione pytanie, należy przygotować plan działań, oszacować ilość potrzebnych zasobów, przeprowadzić testy i monitorować pracę systemu. Cały ten proces można podzielić na trzy etapy.

1. Zacznij od określenia krytycznego docelowego poziomu obciążenia procesora, pamięci i procesów wejścia-wyjścia.

2. Następnie ustal punkt odniesienia.
3. W ostatnim kroku, przed wdrożeniem systemu, obserwuj najważniejsze wskaźniki.

Wyobraź sobie wymagania dotyczące wydajności sklepu internetowego. Czas reakcji na działania użytkowników jest bardzo ważny, aby zachęcić ich do kontynuowania zakupów. Dlatego przy tworzeniu bazy takiego sklepu zwykle warto precyzyjnie zdefiniować czas reakcji na najważniejsze kwerendy. Menedżerowie mogą też zażądać, aby przetwarzanie dowolnej kwerendy zajmowało nie więcej niż 2 – 3 sekundy. Na innym serwerze bazodanowym, który generuje raporty z liczbą sztuk produktów w magazynie, można oczekiwać, że kwerendy będą potrzebowały czasu na pobranie potrzebnych informacji. W tej sytuacji akceptowalny jest czas reakcji na poziomie kilku minut. Nadal jednak może się okazać, że niektóre kwerendy powinny działać znacznie szybciej. W jeszcze innej bazie najważniejszym kryterium wydajności może być czas tworzenia kopii zapasowej bazy danych lub czas wczytywania albo przekazywania danych.

Po ustaleniu najważniejszych poziomów docelowych trzeba zmierzyć używany system, aby uzyskać punkt odniesienia. Techniki monitorowania pracy systemu SQL Server opisujemy szczegółowo w rozdziale 12. tej książki.

Koncentracja na tym, co najważniejsze

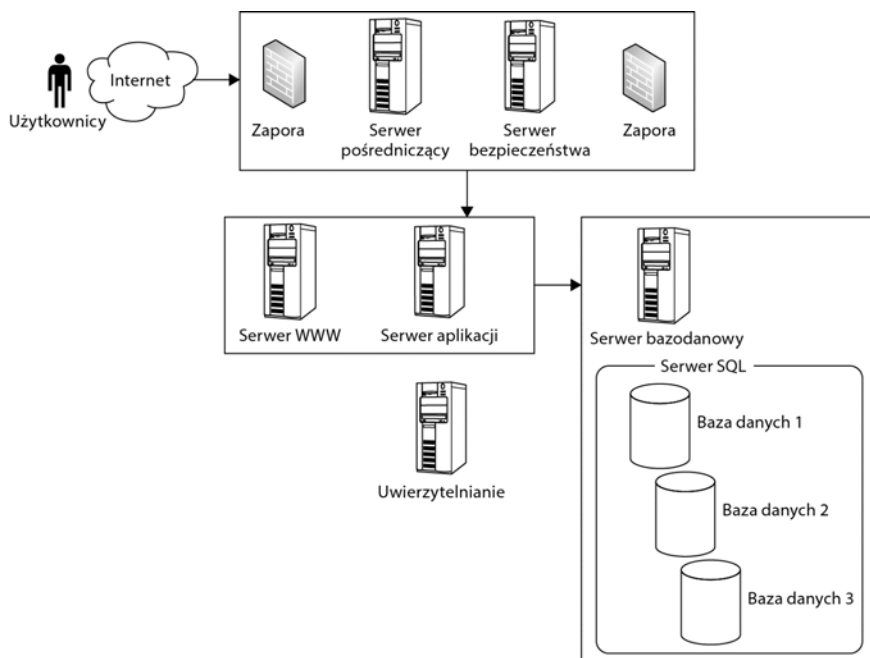
Ostatni ważny aspekt w obszarze wydajności to koncentracja na tym, co istotne — na uzyskaniu wydajności oczekiwanej przez użytkowników. Musisz ustalić, co masz zmierzyć, jak to zrobić, a także jakie błędy pomiaru mogą występować.

Pomyśl o typowym systemie. Odczucia użytkownika końcowego zależą od wydajności wielu elementów: od komputera danej osoby, przez wiele warstw pośrednich, po serwer bazodanowy (i z powrotem). Ponieważ ta książka dotyczy administratorów baz danych i systemu SQL Server 2014, możesz skoncentrować się na pomiarze wydajności tego systemu. Warto jednak zrozumieć ogólny kontekst, w jakim pracuje administrator baz danych, a także poznać narzędzia i miary, które mogą być przydatne w używanym systemie.

Na rysunku 10.8 przedstawiono schematyczny diagram typowej architektury opartej na sieci. Omówione tu rozwiązanie jest typowe dla klientów korporacyjnych używających architektury WSSRA (ang. *Windows Server System Reference Architecture*) do wdrażania rozbudowanych rozwiązań. Możliwe, że wielu czytających to administratorów baz danych po raz pierwszy widzi taki schemat i ma okazję zobaczyć, jakie miejsce w ogólnej architekturze zajmuje interesujący ich fragment (czyli baza danych).

W trakcie analizowania diagramu infrastruktury określ, na czym *powinna* polegać rola administratora bazy danych, a także na czym *polega* rola dobrego członka zespołu. W trakcie rozwiązywania problemów nie należy przyjmować podejścia „to nie moja sprawa”. Aby zapewnić właściwą pracę systemu SQL Server, trzeba zrozumieć, w jaki sposób aplikacje użytkowników, konfiguracja sieci i zabezpieczenia wpływają na ten system lub korzystają z niego.

Gdy użytkownik dzwoni do działu pomocy technicznej i narzeka na niską wydajność, liczba potencjalnych winowajców jest duża. Dlatego ustalenie, który fragment skomplikowanej architektury systemu jest przyczyną problemu, może zająć sporo czasu. Niestety, duże, złożone systemy wymagają wielu pracowników pomocy technicznej koncentrujących się na konkretnych częściach architektury. Przykładowo za zapory i serwer bezpieczeństwa odpowiada zespół ds. sieci, serwerem WWW i serwerem aplikacji zajmuje się zespół ds. aplikacji, o uwierzytelnianie



Rysunek 10.8. Diagram architektury WSSRA

dba zespół ds. systemu Windows, a systemem SQL Server opiekuje się zespół ds. administrowania bazami danych. Proces określania, kto w dużej lub małej firmie odpowiada za poszczególne zadania, polega na rozmowie z wszystkimi zespołami, co pomaga w całościowym rozwiązywaniu problemów.

Co administrator programista powinien wiedzieć na temat wydajności?

Dobra wydajność wynika z solidnych fundamentów, na których można zbudować resztę aplikacji. W bazach z systemu SQL Server tą podstawą jest dobrze zaprojektowany schemat bazy danych. Opisane dalej zasady dostrajania wydajności są bardziej skomplikowane niż tradycyjne wskazówki w rodzaju „znormalizuj bazę do n-tej postaci normalnej”. Przedstawione tu porady wymagają dobrego zrozumienia sposobu korzystania z systemu, w tym wzorców użytkowania, instrukcji w SQL-u i danych. Optymalny schemat dla systemu OLTP może gorzej sprawdzać się w systemach DSS (ang. *Decision Support System*) lub DW (ang. *Data Warehousing*).

Użytkownicy

Najpierw musisz ustalić, kto będzie korzystał z systemu. Określ liczbę użytkowników i jak często jednocześnie korzystają oni z systemu. Ponadto ustal szczytowy poziom obciążenia i operacje wykonywane przez użytkowników. Użytkowników zwykle można przypisać do różnych grup na podstawie stanowisk lub stosowanych funkcji. W systemach obsługujących sklep internetowy grupami użytkowników mogą być: przeglądający produkty, kupujący, sprawdzający stan zamówienia, potrzebujący pomocy i inni. W systemach do analizy sprzedaży główną grupą

użytkowników są analitycy wczytujący dane za pomocą narzędzi do generowania raportów (np. programów PerformancePoint Server, Power View lub Excel) albo tworzący raporty dla zespołów sprzedawców. W systemie obsługującym sklep internetowy można utworzyć procesy pracy w bazie OLTP zoptymalizowane pod kątem mniejszej liczby szybszych operacji odczytu, aktualizacji i zapisu, natomiast w systemie analitycznym przydatne będą procesy pracy w bazie DSS zoptymalizowane pod kątem rozbudowanych kwereń potrzebnych przy generowaniu raportów.

Instrukcje w języku SQL

Po ustaleniu różnych grup użytkowników należy określić wykonywane przez nie operacje, wywoływane instrukcje w języku SQL i częstotliwość ich wykonywania w ramach działań użytkowników. W sklepie internetowym przeglądarka może kierować żądanie do witryny, która zwraca stronę główną. Może to wymagać wykonania 20 – 30 różnych procedur składowanych lub instrukcji w języku SQL. Gdy użytkownik kliknie element na stronie głównej, będzie to wymagać wykonania innego zestawu procedur składowanych i zwrócenia danych z następnej strony. Na razie wszystkie opisywane operacje dotyczą tylko odczytu, jednak na stronach ASP.NET należy też uwzględnić stan sesji, który może być przechowywany w bazie z systemu SQL Server. W takiej sytuacji opisane wcześniej czynności wymagają wielu operacji zapisu.

Wzorce używania danych

Ostatnim omawianym aspektem są dane w bazie. Musisz określić łączną ilość danych w każdej tabeli. Ustal, jak dane trafiają do tabel i jak się zmieniają. W systemie sklepu internetowego główne dane z witryny znajdują się w katalogu ze sprzedawanymi produktami. Produkty z katalogu mogą być pobierane bezpośrednio z witryn dostawców przez portal internetowy. Po początkowym wczytaniu dane można odświeżać, gdy dostawcy zmieniają oferowane produkty lub ich ceny. Ogólna ilość danych zmienia się w bardzo niewielkim zakresie, chyba że dodasz lub usuniesz produkty wybranych dostawców.

Zmieniają się natomiast (miejmy nadzieję, że szybko): liczba zarejestrowanych użytkowników, dane z systemu śledzenia zachowań użytkowników, liczba złożonych zamówień, liczba sprzedanych produktów i liczba dostarczonych zamówień. Oczywiście liczysz na to, że będziesz sprzedawał dużo produktów, co doprowadzi do codziennego wzrostu ilości rejestrowanych danych.

Dobra znajomość danych, ich rozmieszczenia i zmian pomoże wykryć kluczowe miejsca. Mogą nimi być np. punkty, w których dane są często pobierane, wstawiane lub aktualizowane. We wszystkich tych miejscach mogą wystąpić zatory obniżające wydajność systemu.

Schemat bazy danych

Wiedzę na temat wszystkich opisanych wcześniej elementów (użytkowników, instrukcji w języku SQL i danych) trzeba połączyć w całość, aby przygotować dobrze zaprojektowaną i wydajną aplikację. Jeśli podstawa, czyli schemat bazy danych, nie będzie solidna, wszystko, co zostanie na niej zbudowane, będzie niestabilne. Choć można wtedy uzyskać akceptowalne rozwiązanie, prawdopodobnie nie będzie ono optymalne.

W jaki sposób wszystkie wymienione informacje pomagają w dostrajaniu serwera? Dzięki znajomości używanych typów danych łatwiej będzie Ci zrozumieć, które zaawansowane funkcje można wykorzystać. Przykładowo kompresować można tylko dane przechowywane w wierszach.

Jeśli zatem korzystasz z typów LOB, np. `varbinary(max)`, typów tekstowych lub graficznych, nie będziesz mógł zastosować kompresji stron i wierszy. Podobnie nie można używać indeksów kolumnowych i technologii In-Memory OLTP, gdy niezbędne są ograniczenia.

Kolumny z ograniczeniami w postaci klucza zewnętrznego dobrze nadają się do tworzenia indeksów nieklastrowanych. Klucze używane do podziału na partycje zwykle powinny być statyczne, aby zapobiec przenoszeniu danych między partycjami. To proste przykłady, jednak pokazują, że znajomość danych przechowywanych w bazie pomaga zrozumieć, jak optymalnie skonfigurować system SQL Server.

Aby utworzyć maksymalnie wydajny projekt fizyczny, musisz ustalić kluczowe miejsca w danych. Jeśli projektujesz logiczny model danych, zwykle nie tworzysz go z myślą o optymalnej wydajności. Jednak na etapie projektowania modelu fizycznego trzeba uwzględnić zebrane informacje i dostosować projekt do wzorców dostępu do danych.

Co o wydajności powinien wiedzieć administrator produkcyjnej bazy danych?

Praca administratora produkcyjnej bazy danych znacznie różni się od zadań administratora programisty. Administrator produkcyjnych baz danych zarządza przejętym systemem, zaprojektowanym i zbudowanym przez kogoś innego (przy czym system może być nowy lub działać od jakiegoś czasu). Administrator może zmagać się z niską wydajnością starszych systemów, w których dawne aplikacje działają na przestarzałym sprzęcie. W takiej sytuacji zadanie nie polega na projektowaniu wydajnego systemu, ale na sprawieniu, aby przejęty system działał jak najlepiej przy obecnych ograniczeniach sprzętowych.

W tym scenariuszu najpierw trzeba ustalić możliwości sprzętu, zasoby sprzętowe potrzebne systemowi, a także oczekiwania użytkowników w zakresie czasu reakcji. Najważniejsze aspekty sprzętu to szybkość i typ procesora oraz wielkość jego pamięci podręcznej. Ponadto trzeba określić ilość dostępnej pamięci i szybkość magistrali. Należy też ustalić liczbę dysków używanych w operacjach wejścia-wyjścia, konfigurację tych dysków i liczbę kart sieciowych.

Następny krok wymaga określenia, jak wydajny ma być każdy komponent systemu. Czy między firmą a zespołem administratorów obowiązują umowy o gwarantowanym poziomie świadczenia usług dotyczące wydajności? Jeśli ustalone są wytyczne związane z wydajnością, to czy system spełnia je, przekracza, a może działa zbyt wolno? Ponadto zawsze powinienś znać obecny trend. Czy wydajność pozostaje bez zmian, poprawia się, a może — jak zdarza się najczęściej — powoli spada? Administratorzy produkcyjnych baz danych muszą zrozumieć wszystkie wymienione obszary, a także wiedzieć, jak wykrywać zatory i radzić sobie z nimi, aby system działał na oczekiwanym poziomie.

Poniżej znajdziesz listę narzędzi, z których administratorzy produkcyjnych baz danych korzystają przy wykonywaniu swoich zadań.

- **Menedżer zadań.** Narzędzie pozwala szybko zapoznać się z ogólnym obrazem wydajności serwera i wykorzystania zasobów.
- **Monitor wydajności systemu (w systemach Windows 2012 i Windows 8 używany jest Monitor wydajności; inna nazwa to Perfmon).** Zapewnia szczegółowe informacje o wydajności serwera z systemem Windows i liczniki specyficzne dla poszczególnych egzemplarzy systemu SQL Server.

- **Baza MDW systemu SQL Server.** Baza MDW to relacyjna baza danych rejestrująca i przechowująca informacje z monitora wydajności i kolektorów danych. Administrator może pobrać te dane, gdy rozwiązuje problemy systemowe. W systemie SQL Server 2014 można też profilować dane pod kątem ich wykorzystania w technologii In-Memory OLTP.
- **SQL Server Management Studio (SSMS).** Umożliwia analizowanie długich transakcji oraz wykrywanie i naprawianie zatorów. SSMS umożliwia administratorom wywoływanie kwerend kierowanych do widoków DMV i stosowanie zdarzeń rozszerzonych przy zbieraniu danych.
- **Widoki DMV.** Są to obiekty systemowe zawierające informacje o stanie serwera, pomocne przy diagnozowaniu problemów i monitorowaniu stanu systemu SQL Server.
- **Zdarzenia rozszerzone (ang. *extended events*).** Jest to prosty system monitorowania rejestrujący dane na temat wydajności systemu SQL Server. Dane z tego systemu można wyświetlać za pomocą interfejsu użytkownika sesji wprowadzonego w wersji SQL Server 2012. W wersji SQL Server 2014 dodano nowe zdarzenia rozszerzone pomocne przy monitorowaniu systemu i rozwiązywaniu problemów.

Szczegółowe omówienie tych narzędzi zamieszczono w rozdziale 13., „Dostrajanie wydajności kodu w języku T-SQL”.

UWAGA Po wprowadzeniu w wersji SQL Server 2014 narzędzia Transaction Performance Analysis Overview korzystanie z baz MDW zostało znacznie usprawnione. Teraz można zainstalować kolektory w systemach SQL Server 2008 i nowszych, a następnie zbierać dane w egzemplarzu systemu SQL Server 2014, aby ustalić tabele i procedury, które można wykorzystać w technologii In-Memory OLTP. Więcej informacji na ten temat znajdziesz w rozdziale 9., „Technologia In-Memory OLTP”.

Optymalizowanie sprzętu serwera

Pozostała część tego rozdziału dotyczy optymalizowania serwera. Dowiesz się, jak dobrać sprzęt i skonfigurować system operacyjny, aby zapewnić systemowi SQL Server najlepsze środowisko pracy. Przy omawianiu optymalizacji i wydajności zawsze powinieneś pamiętać o trzech podstawowych zasobach:

- procesorze,
- pamięci,
- mechanizmach wejścia-wyjścia.

Zacznijmy od procesora. Możliwości w tym zakresie są dość ograniczone — trzeba określić tylko liczbę i rodzaj procesorów. W tym fragmencie rozdziału koncentrujemy się na różnych cechach procesorów, co pomoże w zakupie odpowiedniego sprzętu.

W zakresie pamięci dostępnych jest znacznie więcej możliwości. Ponadto dużo łatwiej dodać pamięć RAM do systemu lub ją usunąć, niż zmienić liczbę i typ procesorów serwera. Gdy konfigurujesz serwer, powinieneś oszacować, ile pamięci możesz potrzebować. Określ też dostępne możliwości. Omów je z wybranym sprzedawcą. Ponieważ system SQL Server 2014 jest (i pozostanie) dostępny wyłącznie jako serwer 64-bitowy, powinieneś zainstalować go na komputerze 64-bitowym z systemem operacyjnym Windows Server 2012 R2 (też 64-bitowym), co opisujemy dalej w tym rozdziale.

Wydajność operacji wejścia-wyjścia jest prawdopodobnie najważniejszym aspektem konfiguracji serwera, ponieważ wszystkie operacje są wykonywane na dyskach. Cały kod uruchamiany w systemie operacyjnym, systemie SQL Server i innych aplikacjach początkowo znajduje się w plikach na dysku. Wszystkie dane używane w systemie SQL Server także są przechowywane na dysku — są na nim zapisane, potem są wczytywane do pamięci, a następnie ponownie zapisywane na dysku w celu utrwalenia zmian. Każda modyfikacja w bazie z systemu SQL Server jest zapisywana w pliku dziennika transakcji bazy danych, który także znajduje się na dysku. Wszystko to sprawia, że poprawne skonfigurowanie operacji wejścia-wyjścia to bardzo ważny aspekt każdego egzemplarza systemu SQL Server.

W kolejnych punktach szczegółowo omawiamy podstawowe zasoby. Jednak najpierw warto zastanowić się nad tym, jak te trzy zasoby zależą od siebie w kontekście wydajności. Możesz teraz zapisać w książce początkową wydajność procesora, pamięci i operacji wejścia-wyjścia, aby w przyszłości zobaczyć, jak zmieniła się wydajność każdego z tych elementów.

W systemie SQL Server do wersji 2014 nie można było w dużym stopniu zmienić ilości danych przetwarzanych w każdym cyklu procesora. Jednak obecnie natywnie kompilowane procedury składowane (opisane w rozdziale 9., „Technologia In-Memory OLTP”) pozwalają skompilować procedury składowane w języku T-SQL, aby zmniejszyć zbiór instrukcji potrzebnych w trakcie przetwarzania kodu.

Nie każdy kod w języku T-SQL nadaje się do wykorzystania w natywnie kompilowanych procedurach składowanych. Przykładowo skomplikowane kwerendy, ograniczenia i funkcje nie sprawdzają się dobrze w takich procedurach. Ponadto w omawianych procedurach można używać tylko typów danych technologii In-Memory OLTP. Jednak niezależnie od stosowania takich procedur warto pomyśleć o zakupie procesorów z dużą pamięcią podręczną i szybkością. Warto dodać więcej pamięci i zaprojektować podsystem przechowywania danych w taki sposób, aby zapewnić jego maksymalną wydajność w ramach ograniczeń dotyczących szybkości, pojemności i kosztów.

Wniosek z tego fragmentu jest taki, że najwolniejszy element sprzętu ogranicza pozostałe komponenty. Magnetyczne dyski fizyczne nigdy nie będą tak szybkie jak pamięć RAM. System SQL Server (niezależnie od tego, czy używana jest technologia In-Memory OLTP, czy tradycyjny język T-SQL) to aplikacja działająca w pamięci, co oznacza, że wszystkie zadania są wykonywane na danych wczytanych najpierw do pamięci. Gdy określasz wymagania sprzętowe dla systemu SQL Server, upewnij się, że zapewnisz mu wystarczającą ilość pamięci. Raczej nie zdarza się, aby administrator bazy danych narzekał na za dużą ilość pamięci. Z drugiej strony zbyt mała jej ilość często prowadzi do problemów z wydajnością.

Zarządzanie sprzętem

W większości małych i średnich serwerów bazodanowych często włącza się w BIOS-ie technologię Hyper-Threading (szczegółowe informacje na ten temat znajdziesz w dokumentacji serwera). W punkcie „Technologia Hyper-Threading”, dalej w tym rozdziale opisano, jak ustalić, czy ta technologia umożliwi poprawienie wydajności systemu. Po podjęciu decyzji dotyczącej ustawień technologii Hyper-Threading należy fizycznie zainstalować pamięć RAM, urządzenia wejścia-wyjścia (np. karty sieciowe) i adaptery dysków SCSI (ang. *Small Computer Systems Interface*) lub SATA (ang. *Serial Advanced Technology Attachment*). Dodatkowe informacje o zalecanych ustawieniach znajdziesz w dokumentacji udostępnionej przez producentów.

W prawie wszystkich systemach dostępne jest oprogramowanie pomocne przy konfigurowaniu i obsłudze sprzętu oraz zarządzaniu nim. Większość producentów sprzętu udostępnia własne oprogramowanie tego rodzaju, oferujące wiele możliwości i opcji. Są to np. narzędzia iLO firmy Hewlett Packard, RSA firmy IBM i DRAC firmy Dell.

W dużych systemach korporacyjnych, takich jak HP Superdome 2, NEC Express5800 lub SGI Altix UV, konfigurowanie sprzętu serwera odbywa się na zupełnie innym poziomie. W takich systemach używany jest *procesor serwisowy* (ang. *management processor*). Procesor serwisowy i powiązane z nim oprogramowanie pozwalają kontrolować sprzęt: od ładowania partycji, przez konfigurowanie różnych partycji i zmianę układu pamięci, po zarządzanie zasilaniem różnych komponentów. Za wykonywanie wszystkich tych zadań odpowiada właśnie procesor serwisowy.

Zadania wykonywane przy zarządzaniu różnymi dużymi systemami są podobne, jednak każdy producent stosuje inny rodzaj interfejsu: od witryn opartych na Javie w systemach SGI po telnetowy interfejs w wierszu poleceń w rozwiązaniach firm HP i NEC.

UWAGA Gdy rozważasz zakup nowego sprzętu, powinieneś zacząć od przejrzenia katalogu Windows Server Catalog (<http://www.windowsservercatalog.com>). Jeśli dany sprzęt nie jest wymieniony w tym katalogu, nie współdziała z systemem Windows Server 2012 i nie będzie obsługiwał także systemu SQL Server.

Procesor

System SQL Server 2014 działa w innym środowisku niż wcześniejsze jego wersje. Gdy wprowadzono na rynek system SQL Server 2000, duże serwery używane do uruchamiania takich systemów miały od 4 do 8 procesorów. Obecnie system SQL Server 2014 może działać na największych serwerach mających do 64 procesorów i do 320 rdzeni. Ponadto system SQL Server 2014 współdziała z maszynami mającymi do 4 terabajtów pamięci RAM pracującymi pod kontrolą systemu Windows Server 2012 R2 Standard Edition. Jedyny powód, dla którego warto wybrać system operacyjny Windows Server 2012 R2 Data Center, to możliwość obsługi więcej niż dwóch gniazd na procesor. System SQL Server 2014 współdziała z tylko jedną architekturą procesorów — x64 (czyli z architekturą 64-bitową).

Architektura x64

Architektura x64 została wprowadzona przez firmę AMD. Intel zaimplementował tę architekturę pod nazwą EM64T. Jest ona zgodna z kodem maszynowym dla procesorów x86 i obsługuje mikrokod 64-bitowy. Platforma z architekturą x64 obsługuje system SQL Server 2014 i system operacyjny Windows Server 2012 R2 oraz pamięć RAM przekraczającą 4 gigabajty (aż do 4 terabajtów natywnie adresowalnej pamięci) i do 64 fizycznych procesorów z 320 logicznymi rdzeniami.

UWAGA Platforma z architekturą x64 jako jedyna obsługuje system SQL Server 2014. Procesory z rodzin Itanium 64 (IA64) i x32 (x86-32) nie współdziałają z tym systemem. Dostępne są jednak różne edycje i wersje systemów SQL Server 2008 i 2008 R2 obsługiwane przez procesory z wymienionych rodzin.

Pamięć podręczna

W nowoczesnych procesorach potrzebna jest pamięć podręczna, ponieważ działają one z szybkością 2 – 3 GHz. Choć szybkość głównej pamięci RAM jest coraz większa, wciąż jest za niska w porównaniu z prędkością procesora. Aby ograniczyć skutki tego problemu, projektanci procesorów dodali kilka poziomów pamięci flash, co pozwala na przechowywanie używanych ostatnio danych w małych, szybkich blokach pamięci podręcznej. Gdy zatem dane te są ponownie potrzebne, można uzyskać do nich szybki dostęp.

Pamięć podręczna w nowoczesnych procesorach ma zwykle kilka warstw: L1, L2 i L3. Każda kolejna warstwa jest bardziej oddalona od rdzenia procesora i większa (a zarazem wolniejsza). Ostatnią warstwą jest główna pamięć RAM. Niektóre warstwy pamięci podręcznej są przeznaczone do użytku ogólnego i przechowują kopie danych z dowolnej pamięci (np. z L2 lub L3).

Warto przyjrzeć się wydajności pamięci podręcznej w kontekście innych rodzajów pamięci. Pamięć systemowa działa średnio z opóźnieniem 50 – 100 nanosekund i ma pojemność od 16 do 1024 gigabajtów. Dyski SSD cechują się opóźnieniem 30 – 100 mikrosekund i pojemnością 50 – 1024 gigabajtów. Profesjonalne dyski twarde mają opóźnienie na poziomie 2 – 50 milisekund i pojemność od 80 do 2048 gigabajtów.

Wydajność systemu SQL Server w dużym stopniu zależy od pojemności pamięci podręcznej. Producenci procesorów udostępniają wiele modeli z pamięciami L2 i L3 o różnej pojemności. Uzyskanie wysokiej wydajności oferowanej przez pamięć podręczną jest kosztowne. Dlatego procesory o dużej ilości pamięci podręcznej są drogie. Do serwera powinienś zakupić możliwie najszybszy procesor o jak największej pojemności pamięci podręcznej. Jeśli musisz wybrać kompromisowe rozwiązanie, pamiętaj, że zawsze łatwiej i taniej dokupić pamięć RAM, niż zaktualizować procesor.

UWAGA W 2013 roku Intel udostępnił pierwszy chipset z rozbudowaną pamięcią podręczną L4 o pojemności 128 megabajtów. Ten chipset jest obecnie dostępny tylko w procesorach na urządzenia przenośne. Pojawiły się jednak plotki, że procesor Broadwell (następca procesora Haswell) też będzie miał pamięć podręczną L4. W czasie, gdy powstawała ta książka, procesor Broadwell nie był jeszcze dostępny. Zgodnie z doniesieniami pamięć podręczna L4 działa z opóźnieniem 50 – 60 nanosekund.

Technologia Hyper-Threading

Hyper-Threading to zastrzeżona technologia firmy Intel, która logicznie duplikuje wybrane fragmenty fizycznego rdzenia procesora, aby zwiększyć współbieżność obliczeń. W efekcie dla każdego rdzenia fizycznego w systemie operacyjnym pojawiają się dwa rdzenie logiczne. Choć system szereguje do wykonania w procesorze wiele wątków, współużytkowanie zasobów może powodować, że niektóre wątki muszą oczekiwać na zakończenie pracy przez inne.

W kontekście technologii Hyper-Threading trzeba zastanowić się nad tylko jedną kwestią: czy ma być włączona, czy nie. Technologię Hyper-Threading należy domyślnie włączać. Wylączyć ją trzeba tylko wtedy, gdy powoduje spadek wydajności.

Ważnym czynnikiem związanym z technologią Hyper-Threading jest ustalenie maksymalnego teoretycznego wzrostu wydajności, jaki umożliwia. Według dokumentacji Intela maksymalna teoretyczna poprawa wydajności wynosi 30%. Zauważ więc, że ta technologia może w najlepszym razie zwiększyć maksymalną wydajność tylko 1,3 razy w porównaniu z sytuacją, gdy nie jest stosowana. W praktyce ten wzrost może być bliższy 1,1 – 1,15 razy.

W niektórych sytuacjach technologia Hyper-Threading (przynajmniej teoretycznie) nie przynosi żadnych korzyści. Przykładowo w procesach pracy, w których kod działa w krótkiej pętli i wszystkie dane są przechowywane w pamięci podręcznej, omawiana technologia nie jest pomocna, ponieważ używany jest tylko jeden silnik wykonania. Ten scenariusz może prowadzić do spadku wydajności, ponieważ system operacyjny próbuje wtedy przydzielić zadania procesorowi, który fizycznie nie istnieje.

Aby zmierzyć przyrost wydajności, powinieneś przeprowadzić testy porównawcze. Wykonaj następujące kroki.

1. Najpierw sprawdź w dokumentacji systemu, jak wyłączyć technologię Hyper-Threading i zrób to.
2. Potem przeprowadź kilkakrotnie test, aby otrzymać średni czas wykonania.
3. Następnie ponownie włącz technologię Hyper-Threading.
4. Przeprowadź test tyle samo razy, co wcześniej, i porównaj wyniki. Celem jest ustalenie, o ile krótszy jest średni czas wykonania testu z włączoną technologią Hyper-Threading. Oblicz różnicę między czasem z obu testów i na podstawie uzyskanego wyniku ustal, czy warto stosować omawianą technologię.

Pojęcia związane z systemami wielordzeniowymi

W tym miejscu warto zapoznać się z definicjami wybranych pojęć, aby uniknąć wątpliwości podczas poznawania systemów wielordzeniowych.

- *Gniazdo* to fizyczne gniazdo, w które wkładany jest procesor. Przed pojawieniem się systemów wielordzeniowych jednemu gniazdu odpowiadała jedna jednostka wykonawcza.
- *Rdzeń to jednostka wykonawcza* (wcześniej za taką jednostkę uznawano procesor). W procesorach wielordzeniowych na gniazdo przypadają dwa rdzenie lub większa ich liczba.
- *Wątek* w tym kontekście nie jest tym samym, co wątek tworzony w programie lub wątek systemu operacyjnego. Tu wątek ma znaczenie tylko w kontekście technologii Hyper-Threading. Taki wątek nie jest nową jednostką wykonania; bardziej przypomina nowy potok istniejącej jednostki wykonania. Więcej informacji na temat tego rozwiązania znajdziesz w poprzednim punkcie, „Technologia Hyper-Threading”.

Dla każdego egzemplarza systemu SQL Server 2014 trzeba wykupić licencję przynajmniej na 4 rdzenie. Niezależnie od tego, czy będą one fizyczne, czy wirtualne, każdy egzemplarz systemu SQL Server 2014 powinien mieć dostęp do 4 rdzeni. W kontekście technologii Hyper-Threading istotne jest to, że użytkownik płaci tylko za fizyczne rdzenie używane przez fizyczny egzemplarz systemu SQL Server. Nie trzeba ponosić opłat za dodatkowe rdzenie logiczne udostępniane za pomocą tej technologii.

Architektura NUMA

Nazwa NUMA to akronim *non-uniform memory access* (czyli niejednorodny dostęp do pamięci). Architektura ta często jest nazywana ccNUMA (ang. *cache-coherent NUMA*), czyli NUMA spójna z pamięcią podręczną. Główne różnice między starszym systemem SMP a architekturą NUMA dotyczą miejsca podłączenia pamięci i uporządkowania procesorów na magistrali systemowej.

W systemie SMP pamięć jest podłączona do wszystkich procesorów symetrycznie za pomocą współużytkowanej magistrali. W architekturze NUMA każda grupa procesorów ma własną pulę lokalnej pamięci. Zaletą tego modelu jest to, że procesory przy dostępie do pamięci nie muszą ponosić kosztów wychodzenia poza pamięć lokalną (pod warunkiem, że potrzebne dane się w niej znajdują). Gdy szukane dane są zapisane w puli pamięci innego węzła architektury NUMA, koszt dostępu do nich jest nieco wyższy niż w systemie SMP. Dlatego jednym z celów przy stosowaniu architektury NUMA jest zmaksymalizowanie ilości danych pobieranych z pamięci lokalnej i zminimalizowanie dostępu do danych z innych węzłów.

W architekturze NUMA zwykle stosuje się dwa gniazda na węzeł i tworzy się tyle węzłów, ile pozwala na to system.

Mniejsze systemy z węzłami, w których liczba gniazd jest wielokrotnością dwójki, można zwykle modyfikować. Dzięki temu serwery mają duże możliwości w zakresie konfigurowania i skalowania. Przykładowo firma może zacząć od jednego węzła z dwoma gniazdami, a następnie rozbudować system do szesnastu węzłów po cztery gniazda każdy (w sumie daje to sześćdziesiąt cztery gniazda).

Procesory Intel Xeon i AMD Opteron wykorzystują odmienną architekturę dostępu do pamięci. W procesorach Intela używa się magistrali FSB (ang. *Front Side Bus*), w której gniazda są za pomocą magistrali podłączone przez zewnętrzny kontroler do pamięci. W efekcie wszystkie gniazda znajdują się w tej samej odległości od pamięci. W procesorach AMD w każdym gnieździe używany jest zintegrowany kontroler pamięci zapewniający dostęp do lokalnej pamięci i do pamięci innych gniazd. Do transmisji danych stosowana jest tu technologia HyperTransport. Taki niejednorodny dostęp do pamięci nazywany jest architekturą NUMA. Opóźnienie w dostępie do danych zależy tu od tego, w którym miejscu pamięci znajdują się dane zażądane przez określony rdzeń procesora.

Jeśli np. dane znajdują się w bezpośrednio podłączonym bloku pamięci, czas dostępu jest krótki. Jeżeli jednak dane znajdują się w zdalnym bloku pamięci w innym gnieździe, opóźnienie jest większe. Choć w architekturze Intela magistrala FSB zapewnia każdemu rdzeniowi procesora dostęp do pamięci oddalonej zawsze o identyczną odległość, współzawodnictwo o samą magistralę może prowadzić do spadku wydajności. Intel ogranicza ten problem za pomocą pamięci podręcznej o większej pojemności.

Pamięć

Innym podsystemem sprzętowym, który trzeba uwzględnić, jest pamięć. Ważna jest głównie pamięć serwera, w tym związane z nią problemy, dostępne opcje i jej wpływ na wydajność serwera. W następnym punkcie znajdziesz podstawowe wprowadzenie do pamięci systemu operacyjnego. Dalej szczegółowo omawiamy konfigurowanie serwera w obszarze pamięci.

Pamięć fizyczna

Pamięć fizyczna to pamięć RAM instalowana na serwerze. Prawdopodobnie znasz już pamięć DIMM (ang. *Dynamic Inline Memory Modules*) używaną w komputerach PC i serwerach. Jest to jedna z odmian pamięci fizycznej (pamięci RAM). Pojemność takiej pamięci mierzy się w megabajtach, gigabajtach lub — jeśli masz szczęście — w terabajtach, ponieważ najnowsze systemy operacyjne Windows Server 2012 R2 Datacenter i Standard obsługują do 4 terabajtów

pamięci RAM. W nowszych wersjach tego systemu operacyjnego ilość obsługiwanej pamięci będzie jeszcze większa, ponieważ klienci oczekują jeszcze wydajniejszych narzędzi do rozwiązywania coraz bardziej skomplikowanych problemów biznesowych.

Fizyczna przestrzeń adresowa

Fizyczna przestrzeń adresowa to zbiór adresów używany przez procesor do dostępu do elementów z magistrali. Dużą część tej przestrzeni zajmuje pamięć, natomiast inne fragmenty są zarezerwowane np. na odwzorowania buforów sprzętowych i specyficzną pamięć (taką jak np. pamięć RAM dla przetwarzania grafiki). W procesorach 32-bitowych wielkość przestrzeni adresowej była ograniczona do 4 gigabajtów. W przeznaczonych dla serwerów 32-bitowych procesorach Intel'a z technologią PAE (ang. *Physical Address Extension*) używana była 36-bitowa magistrala adresowa, co pozwalało na obsługę 64 gigabajtów adresów pamięci. Mogłoby się wydawać, że w procesorach 64-bitowych stosowana jest 64-bitowa magistrala adresowa, jednak ponieważ na razie nie są potrzebne systemy adresujące 18 eksabajtów pamięci (nie jest też możliwe zbudowanie tak dużego systemu), producenci ograniczyli magistralę adresową do 48 bitów, co wystarcza na adresy dla 256 terabajtów pamięci. Używana architektura umożliwia rozszerzenie magistrali adresowej do 52 bitów, co pozwoli na tworzenie systemów mających do 4 petabajtów pamięci.

Menedżer pamięci wirtualnej

Menedżer pamięci wirtualnej (ang. *Virtual Memory Manager* — VMM) to część systemu operacyjnego zarządzająca całą pamięcią fizyczną i udostępniającą ją wszystkim procesom, które jej potrzebują. Zadaniem menedżera VMM jest udostępnianie każdemu procesowi pamięci, gdy jej zażąda, przy czym fizyczna pamięć jest współużytkowana przez wszystkie procesy działające w danym momencie w systemie.

Menedżer VMM zarządza pamięcią wirtualną każdego procesu i — jeśli trzeba — przywraca powiązaną z nią pamięć fizyczną oraz umieszcza zapisane w niej dane w pliku stronicowania, aby nie zostały utracone. Gdy proces ponownie będzie potrzebował tej pamięci, menedżer VMM przywróci dane z pliku stronicowania, znajdzie wolną stronę pamięci (albo na liście wolnych stron, albo w innym procesie), umieści na niej dane z pliku stronicowania i powiąże nową stronę z wirtualną przestrzenią adresową procesu. Związane z tym opóźnienie lub przerwanie to *błąd strony* (ang. *page fault*). Aby ustalić, czy system SQL Server lub inny proces nie powodują nadmiernego stronicowania, sprawdź wartość licznika *Proces: Błędy stron/s* dla procesu systemu SQL Server. Szczegółowe informacje na ten temat zamieszczono w punkcie „Błędy strony”, dalej w tym rozdziale.

W systemach z wystarczającą ilością pamięci RAM, w których każdemu procesowi można przydzielić całą potrzebną pamięć, menedżer VMM nie ma wiele pracy. Musi tylko udostępniać pamięć i wykonywać operacje porządkujące, gdy proces już jej nie potrzebuje. W systemach, w których brakuje pamięci RAM, sytuacja jest bardziej skomplikowana. Menedżer VMM musi wykonać określone zadania, aby w odpowiednim czasie udostępnić każdemu procesowi potrzebną pamięć. Używany jest do tego plik stronicowania. Zawiera on dane ze stron, których proces nie używa lub które według menedżera VMM można usunąć z procesu.

Plik stronicowania

Plik stronicowania (ang. *page file*) to przechowywany na dysku plik, używany przez komputer do zwiększenia ilości fizycznej pamięci wykorzystywanej w pamięci wirtualnej. Gdy pamięć używana przez wszystkie istniejące procesy przekracza ilość dostępnej pamięci RAM, system operacyjny Windows przenosi strony z wirtualnych przestrzeni adresowych do pliku stronicowania przechowywanego na fizycznym dysku. To pozwala zwolnić pamięć RAM dla innych zadań. Przeniesione strony są przechowywane w plikach stronicowania w katalogu głównym partycji. W każdej partycji może znajdować się tylko jeden taki plik.

Na serwerze z systemem SQL Server należy postarać się, aby ten system używał tylko dostępnej pamięci fizycznej. Sam system SQL Server w dużym stopniu dba o to, by nie zajmować zbyt dużej ilości pamięci, i próbuje nie przekraczać limitu dostępnej pamięci fizycznej.

Dzięki temu w większości sytuacji plik stronicowania jest potrzebny tylko w małym stopniu. Jednak wielu użytkowników zadaje pytanie: „Jaka jest zalecana wielkość pliku stronicowania?”. Oczywiście odpowiedź na to pytanie brzmi: „To zależy”. Wpływa na to ilość zainstalowanej pamięci RAM i zapotrzebowanie na pamięć wirtualną ze strony procesów innych niż procesy systemu SQL Server. Zwykle plik stronicowania powinien mieć pojemność 1,5 – 2 razy większą niż ilość pamięci RAM serwera.

W rozbudowanych systemach o dużej ilości pamięci RAM (powyżej 128 gigabajtów) czasem nie da się skorzystać z tej porady z powodu braku miejsca na dysku. Oto wskazówki, które mogą okazać się przydatne w takich sytuacjach.

- Utwórz na dysku systemowym plik stronicowania o pojemności 8 gigabajtów.
- Upewnij się, że parametry uruchomieniowe systemu operacyjnego Windows powodują zapis rzutu jądra przy wystąpieniu awarii.

UWAGA Aby się dowiedzieć, jak skonfigurować te parametry, zapoznaj się z artykułem <http://support.microsoft.com/kb/307973> z witryny z pomocą techniczną Microsoftu.

- Opcjonalnie możesz skonfigurować kilka plików stronicowania (na woluminach innych niż wolumin systemowy), udostępnianych systemowi operacyjnemu wtedy, gdy potrzebna jest większa pojemność tego pliku.

W niektórych sytuacjach system SQL Server i system operacyjny nie współdziałają dobrze w zakresie współużytkowania dostępnej pamięci. Pojawiają się wtedy ostrzeżenia systemowe dotyczące małej ilości pamięci wirtualnej. Wtedy najlepiej dodać do serwera więcej pamięci RAM, zmienić konfigurację systemu SQL Server, aby zużywał mniej pamięci, lub zwiększyć rozmiar pliku stronicowania. Czasem lepiej tak ustawić system SQL Server, aby nie wykroczył poza dostępną pamięć fizyczną, niż powiększać plik stronicowania. Ograniczenie stronicowania zawsze prowadzi do wyższej wydajności. Jeśli stronicowanie jest nieuniknione, to aby uzyskać optymalną wydajność, należy umieścić plik stronicowania na szybkim i rzadko używanym dysku. Dysk ten należy okresowo defragmentować, aby plik stronicowania zajmował ciągły obszar pamięci. Zmniejsza to konieczność przesuwania głowicy dysku i zwiększa wydajność. Do pomiaru wykorzystania pliku stronicowania służy wskaźnik *Plik stronicowania: Użycie%* w monitorze wydajności systemu Windows. Jego wartość nie powinna przekraczać 70%.

Błędy strony

Błędy strony zwykle sprawiają problemy w systemie SQL Server, jednak nie wszystkie błędy tego rodzaju są takie same. Niektórych nie da się uniknąć, inne mają niewielki wpływ na wydajność, a jeszcze inne mogą prowadzić do poważnego spadku wydajności — tych należy się wystrzegać.

System SQL Server jest zaprojektowany tak, aby korzystać z takiej ilości pamięci, jaka fizycznie jest dostępna. Ma to chronić przed szkodliwymi błędami strony. Niestety, licznik błędów strony w monitorze wydajności nie informuje, czy skutki danego błędu są niewielkie, czy poważne. Dlatego nie da się określić wpływu danego błędu na wydajność.

Miękkie błędy strony

Najczęściej występującym typem omawianego problemu są *miękkie błędy strony*. Występują one wtedy, gdy potrzebna jest nowa strona pamięci. Za każdym razem, gdy system SQL Server chce zająć dodatkową pamięć, żąda od menedżera VMM nowej strony pamięci. Menedżer zgłasza wtedy miękki błąd strony, aby przenieść daną pamięć do wirtualnej przestrzeni adresowej systemu SQL Server. Dzieje się to, gdy SQL Server po raz pierwszy chce użyć danej strony, a nie wtedy, gdy po raz pierwszy jej zażąda. System SQL Server musi wywołać instrukcję `VirtualAlloc`, aby otrzymać stronę pamięci. Tego rodzaju błąd strony występuje tylko wtedy, gdy system SQL Server po raz pierwszy próbuje zapisać dane na określonej stronie.

Twarde błędy strony

Twarde błędy strony to te, których powinienś unikać. Błędy te występują wtedy, gdy system SQL Server próbuje uzyskać dostęp do strony pamięci przeniesionej do pliku stronicowania. W takiej sytuacji menedżer VMM musi wykonać odpowiednie operacje, aby pobrać potrzebną stronę z pliku stronicowania z dysku, znaleźć pustą stronę pamięci, wczytać stronę z dysku, zapisać ją na nowej pustej stronie, a następnie powiązać nową stronę z przestrzenią adresową systemu SQL Server. Przez cały ten czas wątek systemu SQL Server musi czekać. Dopiero po zastąpieniu brakującej strony pamięci przez menedżer VMM system SQL Server może kontynuować wykonywane zadania.

Operacje wejścia-wyjścia

Konfigurowanie operacji wejścia-wyjścia to zbyt obszerny temat, aby można go było opisać w jednym rozdziale. To zagadnienie zasługuje na odrębną książkę. W tym podrozdziale omawiamy wybrane z dostępnych opcji wejścia-wyjścia, a także przedstawiamy kilka scenariuszy. Pomoże Ci to zrozumieć, jak podejmować właściwe decyzje w trakcie konfigurowania systemu pamięci.

Operacje wejścia-wyjścia dotyczą zarówno sieci, jak i dysków twardych. W systemie SQL Server w większości sytuacji istotne są przede wszystkich operacje wejścia-wyjścia na dyskach twardych, ponieważ to tam przechowywane są dane. Powinieneś jednak zrozumieć także skutki wolnego działania operacji wejścia-wyjścia w sieci, ponieważ mogą one powodować zatory i obniżyć wydajność systemu.

Konfigurowanie operacji wejścia-wyjścia dla systemu pamięci serwera to obszar, w którym masz prawdopodobnie najwięcej możliwości, a także możesz w największym stopniu wpłynąć na wydajność systemu SQL Server. Gdy wyłączysz komputer, jedyną istniejącą rzeczą są dane zapisane

na dysku twardym. Gdy włączysz zasilanie, procesor rozpocznie pracę, wczytany zostanie system operacyjny i uruchomi się system SQL Server. Wszystko to stanie się w wyniku wczytania danych i kodu z dysku.

Ten krótki opis dotyczy tego, co dzieje się na komputerze. Wszystko rozpoczyna się na dysku i musi zostać wczytane z niego do pamięci, a stamtąd do różnych buforów pamięci podręcznej procesora, skąd dane trafiają do procesora i mogą zostać użyte jako kod lub dane. Następnie wyniki uzyskane przez procesor muszą zostać ponownie zapisane na dysku, aby mogły przetrwać zdarzenia systemowe (np. zamknięcie, awarię lub konserwację systemu itd.).

System SQL Server bardziej niż wiele innych aplikacji jest wrażliwy na wydajność pracy dysku. Dzieje się tak, ponieważ zarządza dużymi ilościami danych przechowywanych w bazach użytkowników. Większość aplikacji może wczytać wszystkie dane z dysku w pamięci, a następnie działać przez długi czas bez konieczności ponownego dostępu do dysku. Twórcy systemu SQL Server też starają się umożliwić taki model działania, ponieważ jest to zdecydowanie najszybsze rozwiązanie. Niestety, gdy ilość danych potrzebna w określonej operacji przekracza pojemność pamięci, system SQL Server musi zacząć przenosić dane, aby uzyskać jak najwyższą wydajność. Rozpoczyna wtedy opróżnianie bufora zapisu i musi zapisywać dane z powrotem na dysk, aby zwolnić pamięć na generowane nowe wyniki.

W cyklu życia danych systemu SQL Server każda porcja danych musi wcześniej lub później zostać wczytana z dysku, a zmodyfikowane informacje trzeba z powrotem na nim zapisać.

Sieć

Wróć teraz do rysunku 10.8. Widać na nim, że sieć jest ważnym elementem systemu SQL Server. Sieć zapewnia połączenie, przez które system SQL Server odbiera wszystkie żądania i odsyła wyniki z powrotem do klientów. Współczesne szybkie sieci mają zwykle wystarczającą przepustowość, aby nie powodować zatorów i umożliwić systemowi SQL Server maksymalne wykorzystanie wszystkich pozostałych zasobów (procesora, pamięci i dysku).

Jednak w niektórych scenariuszach system SQL Server wykonuje stosunkowo niewiele pracy w porównaniu z liczbą żądań przesyłanych do serwera lub ilością danych zwracanych do klientów. W takich sytuacjach sieć może być wąskim gardłem. Zatory mogą powstawać w dowolnym miejscu sieci, np. w karcie sieciowej klienta, gdy klientem jest serwer aplikacji przekazujący do serwera bazodanowego setki tysięcy żądań na sekundę, lub w infrastrukturze sieci łączącej serwer bazodanowy z klientem (serwerem aplikacji, serwerem WWW lub stacją roboczą użytkownika). Infrastruktura sieci obejmuje liczne komponenty i może tworzyć proste systemy, w których dwa komputery są połączone w sieci LAN, a także bardzo skomplikowane połączone siecią systemy pracujące w internecie lub globalnych korporacyjnych sieciach WAN.

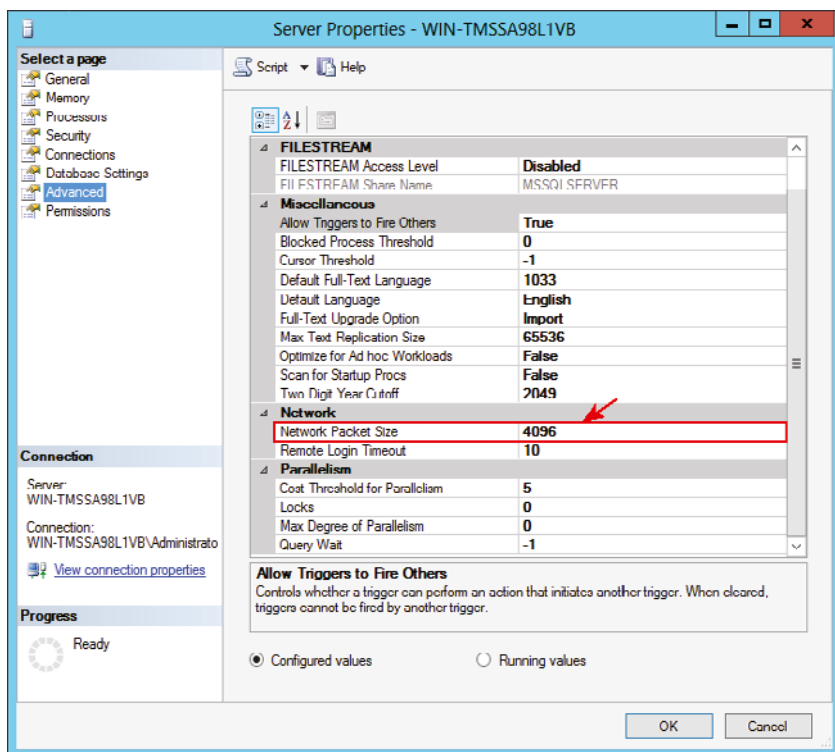
W większych, skomplikowanych systemach duża część sieci może pozostawać poza Twoją kontrolą i powodować nieakceptowalne problemy z przepustowością lub opóźnieniami. W takiej sytuacji możesz tylko zbadać sytuację, udokumentować ją i zgłosić.

W tym miejscu omawiamy elementy sieci, które możesz bezpośrednio kontrolować. Wszystkie one działają w ramach systemu SQL Server. Można założyć, że pozostałe części infrastruktury sieci potrafią obsłużyć zgłoszone żądania i odpowiednio szybko zwrócić wyniki do klienta.

Jedną z rzeczy, które należy sprawdzić, są ustawienia związane z szybkością i transmisją dwukierunkową. Niestety, łatwo niewłaściwie ustawić transmisję dwukierunkową i spowolnić pracę sieci. Standardowo sieć działa w trybie w pełni dwukierunkowym. Oznacza to, że dane są jednocześnie przesyłane w obu kierunkach. Wymaga to zastosowania odpowiednich kabli. Obecnie sieci działają z szybkością 10 GB/s.

W trakcie analizowania statystyk oczekiwania sieci często można natrafić na stan `ASYNC_NETWORK_IO`. Ten typ oczekiwania występuje, gdy system SQL Server zakończy przetwarzanie kwerendy nadesłanej przez aplikację użytkownika. Aby nie przeciążyć aplikacji klienta, system SQL Server wysyła początkowy pakiet i oczekuje, że aplikacja odbierze ten pakiet i odeśle potwierdzenie. W tym czasie stan oczekiwania to właśnie `ASYNC_NETWORK_IO`.

Jedną z możliwości zapobiegania temu stanowi jest upewnienie się, że pakiet o danej wielkości można przesłać za pomocą używanej infrastruktury sieci, i zmiana wielkości pakietu sieciowego na 16383 (jest to bardzo często używany rozmiar pakietów). Ilustruje to rysunek 10.9.



Rysunek 10.9. Zmienianie wielkości pakietu sieciowego

Dyski magnetyczne

W serwerach nadal najczęściej używane są dyski magnetyczne, dlatego warto je omówić. Są one jednak mniej wydajne od dysków SSD i Flash oraz macierzy takich dysków oferowanych przez Violin Memory i innych producentów. Dyski SSD, gdy są poprawnie skonfigurowane, działają szybciej niż magnetyczne dyski twarde. Odporność na błędy w stosowanych obecnie architekturach dysków SSD jest coraz lepsza, jeśli jednak przyjrzymy się cenom, dyski magnetyczne wciąż są bardziej ekonomicznym rozwiązaniem.

Innym ważnym obszarem wejścia-wyjścia są dyskowe operacje tego rodzaju. We wcześniejszych wersjach systemu SQL Server obsługa dysku była stosunkowo prosta, dlatego także liczba dostępnych opcji była niewielka. W większości sytuacji trzeba było zarządzać tylko kilkoma dyskami. Obecnie w systemach korporacyjnych można stosować technologie SAN (ang. *Storage Area Network*) i NAS (ang. *Network Attached Storage*), a także zewnętrzne podsystemy dyskowe

wykorzystujące macierze RAID (ang. *Redundant Array of Independent Disks*). Często używany jest też interfejs SCSI umożliwiający budowanie podsystemów obejmujących setki, a nawet tysiące dysków.

W systemach dyskowych używane są różne interfejsy.

- **ATA (ang. *Advanced Technology Attachment*)**. Inna nazwa to IDE (ang. *Integrated Drive Electronics*), która określa nie tylko działanie złącza i interfejsu, ale też to, że kontroler jest zintegrowany z napędem. Interfejsy PATA (ang. *Parallel ATA*) umożliwiają transfer danych równoległymi strumieniami między płytą główną a dyskiem z szybkością do 133 MB/s. Interfejs SATA (ang. *Serial ATA*) został opracowany w celu zlikwidowania architektonicznych ograniczeń interfejsu równoległego. SATA umożliwia transfer danych z szybkością do 3 GB/s.
- **SCSI (ang. *Small Computer Systems Interface*)**. Jest to zbiór standardów opracowanych dla połączeń i transferu danych między komputerami oraz licznymi urządzeniami peryferyjnymi, w tym dyskami. Interfejs wykorzystuje równoległe przesyłanie strumieni danych z szybkością do 640 MB/s. Rozwinięciem tego standardu jest SAS (ang. *Serial Attached SCSI*), w którym strumienie danych są przesyłane szeregowo z szybkością do 4800 MB/s.

Rozważmy teraz podstawowe czynniki fizyczne wpływające na wydajność dysków. Musisz poznać główne różnice między różnymi rodzajami dysków, ponieważ pozwala to zrozumieć różnice w ich wydajności. To z kolei pomoże wybrać odpowiedni typ dysków. W tabeli 10.3 przedstawione są przykładowe wartości (rejestrowane w idealnych warunków) typowych rodzajów opóźnienia dysków.

Tabela 10.3. Przykładowe opóźnienia dysków twardych

Szybkość obrotowa dysku	Opóźnienie obrotowe	Czas przejścia do następnej ścieżki	Czas wyszukiwania	Szybkość przesyłu danych	Czas przesyłu 8 kilobajtów	Łączne opóźnienie
5400 RPM	5,5 ms	6,5 ms	12 ms	90 MB/s	88 μs	12,1 ms
7200 RPM	4,1 ms	6,5 ms	10,7 ms	120 MB/s	66 μs	10,8 ms
10 000 RPM	3 ms	1,5 ms	4,5 ms	166 MB/s	48 μs	4,6 ms
15 000 RPM	2 ms	1,5 ms	3,5 ms	250 MB/s	32 μs	3,5 ms

Większa liczba dysków

Większa liczba dysków pozwala uzyskać wyższą szybkość. Jeśli np. potrzebujesz 4 terabajtów pojemności, uzyskasz wyższą wydajność, gdy zastosujesz wiele małych dysków (10 dysków po 400 gigabajtów), niż gdy wykorzystasz dwa większe dyski (2 dyski po 2 terabajty). Wynika to z wielu przyczyn. Po pierwsze, mniejsze dyski są zwykle szybsze od większych. Po drugie, szybkość wzrasta, gdy do odczytu i zapisu danych używanych jest więcej wrzecion z wielu dysków pracujących w macierzy. Czasem pozwala to uzyskać przepustowość równą sumie przepustowości poszczególnych dysków. Jeśli wspomniane dyski o pojemności 400 gigabajtów i 2 gigabajtów działają z tą samą szybkością (np. 20 MB/s), dla 2 dysków suma wyniesie tylko 40 MB/s. Jednak dla 10 mniejszych dysków suma jest równa 200 MB/s, czyli pięciokrotnie więcej.

Szybsze dyski

Nie jest zaskoczeniem, że szybsze dyski zapewniają wyższą wydajność niż wolniejsze. Jednak nie chodzi tu wyłącznie o szybkość obrotową, która jest tylko jednym z czynników wpływających na ogólną prędkość dysku. Istotne jest to, czy dysk poradzi sobie z obsługą operacji wejścia-wyjścia w procesach pracy, które są najważniejsze dla użytkowników.

Niestety, producenci dysków rzadko (jeśli w ogóle) podają dane inne niż prędkość obrotowa i teoretyczna szybkość magistrali dysku. Przykładowo często można przeczytać, że dysk SCSI o szybkości 10 000 lub 15 000 RPM ma maksymalną przepustowość 300 MB/s, ponieważ używa magistrali SCSI 320. Jeśli jednak podłączysz taki dysk i przeprowadzisz testy przy użyciu programu SQLIO lub podobnego narzędzia, bardzo możliwe jest, że dla odczytu losowych niewielkich bloków nieprzechowywanych w kolejce uzyskasz przepustowość na poziomie 2 – 4 MB/s. Nawet dla najszybszych operacji wejścia-wyjścia, czyli sekwencyjnych operacji na dużych blokach, przepustowość rzadko będzie przekraczać 60 – 70 MB/s. To dużo mniej niż 300 MB/s.

UWAGA Narzędzie SQLIO Disk Subsystem Benchmark możesz pobrać ze strony www.microsoft.com/download/en/details.aspx?id=20163. Pod adresem http://sqlserverpedia.com/wiki/SAN_Performance_Tuning_with_SQLIO znajdziesz bardzo dobry samouczek, za pomocą którego zapoznasz się z tym programem.

Odczyt i zapis w pamięci podręcznej

W każdym dysku twardym znajduje się obszar pamięci nazywany *buforem dysku* lub *pamięcią podręczną dysku*. Pamięć podręczna pełni funkcję bufora między dyskiem a podłączonym systemem i służy do przechowywania danych wczytywanych z dysku lub zapisywanych na nim. Pojemność tego typu pamięci podręcznej wynosi od 8 do 64 megabajtów. Także kontrolery dysków (niezależnie od tego, czy działają wewnętrznie na serwerze, czy zewnętrznie w ramach architektury SAN) mają pamięć podręczną używaną do przechowywania wczytywanych i zapisywanych danych. Pojemność pamięci podręcznej kontrolera wynosi zwykle od 512 megabajtów do 512 gigabajtów.

Oto wybrane z wielu zastosowań pamięci podręcznej dysku.

- **Odczyt z wyprzedzeniem i odczyt z opóźnieniem.** Gdy dysk ma wczytać dane, może pobrać także informacje, których nie zażądano, jeśli uzna, że mogą być później przydatne systemowi SQL Server.
- **Przyspieszanie zapisu.** Kontroler dysku może natychmiast po otrzymaniu danych poinformować system SQL Server, że operacja zapisu się powiodła, choć dane nie są jeszcze zapisane na dysku. Dane trafiają wtedy do pamięci podręcznej zapisu i są zapisywane później. Grozi to utratą danych, jeśli nastąpi przerwa w zasilaniu przed fizycznym zapisaniem danych. Aby ograniczyć to zagrożenie, większość kontrolerów macierzy dyskowych ma bateryjne zasilanie awaryjne, co gwarantuje zakończenie wszystkich zaległych operacji zapisu nawet wtedy, gdy główne zasilanie zostanie wyłączone.
- **Równoważenie operacji wejścia i wyjścia.** Częstotliwość wykonywania operacji odczytu i zapisu waha się w trakcie pracy systemu. Aby mieć pewność, że żądania zostaną odpowiednio szybko obsłużone, używa się pamięci podręcznej do przechowywania danych oczekujących na przesłanie w obu kierunkach.

W większości konfiguracji pamięć podręczna odczytu i zapisu korzysta z tego samego obszaru pamięci, a procesor dysku (lub macierzy) kontroluje szczegóły jej działania. W niektórych konfiguracjach baz OLTP pamięć podręczna odczytu jest wyłączona, co pozwala zwiększyć ilość pamięci podręcznej zapisu.

Dyski SSD (dyski z pamięcią flash)

Trudno porównywać dyski SSD z dyskami magnetycznymi. Tradycyjne pomiary dysków twardych dotyczą głównie cech, w których takie dyski nie osiągają dobrych wyników (np. opóźnienia obrotowego lub czasu wyszukiwania). Dyski SSD nie muszą się obracać, dlatego nie mają opóźnienia obrotowego. W takich dyskach mogą jednak występować problemy z naprzemiennymi operacjami odczytu i zapisu. Ponadto z czasem wydajność takich dysków spada.

Istnieją dwa rodzaje pamięci flash — NAND i NOR. Pamięć NOR działa podobnie jak pamięć RAM i wymaga zasilania, aby przechowywać dane. Pamięć NAND jest trwała i nie wymaga elektryczności do przechowywania danych. Dlatego skoncentrujemy się tu na pamięci NAND.

Dyski SSD typu NAND obejmują chipy z pamięcią flash dotyczącą komórek SLC (ang. *Single Layer Cells*) lub MLC (ang. *Multi Layer Cells*). Komórki SLC przechowują tylko jeden bit danych (0 lub 1). Komórki MLC mogą przechowywać więcej informacji. Obecnie zwykle zawierają po 2 bity. Gęstość danych w komórkach MLC jest wyższa, dlatego dyski z takimi komórkami są bardziej pojemne od dysków z komórkami SLC.

Odczyt danych z dysków z komórkami MLC może być wolniejszy z powodu korekcji błędów niezbędnej w celu naprawiania pomyłek. Takie dyski wymagają zapisu całych stron, natomiast w pamięciach SLC możliwy jest wydajniejszy zapis części stron. Jednak większość firm odchodzi od dysków SLC SSD w kierunku dysków MLC SSD. Strona na dyskach SSD ma zwykle 4 kilobajty. Strony są zapisywane w blokach mających przeważnie od 128 do 512 kilobajtów.

Początkowy zapis na dysku SSD jest mniej kosztowny, ponieważ zapisywana jest tylko jedna strona. Jednak późniejsze operacje zapisu, np. przy usuwaniu i ponownym dodawaniu danych, wymagają opróżnienia strony i ponownego zapisania bloku. Jeśli chodzi o wydajność, odczyt zajmuje zwykle 25 mikrosekund, zapis pierwszej strony o wielkości 4 kilobajtów trwa 250 mikrosekund, a usunięcie i ponowne zapisanie bloku o pojemności 256 kilobajtów wymaga 2 milisekund.

Dyski SSD po pewnym czasie się zużywają. Można na nich wykonać tylko określoną liczbę operacji. Dyski SLC SSD zwykle zaczynają się psuć po 100 000 cykli kasowania, a dyski MLC SSD zaczynają sprawiać problemy po 5000 – 10 000 takich cykli. Jeśli system co sekundę aktualizuje i kasuje jeden blok, wykonanie 10 000 cykli zajmie ponad 5 lat. Jeśli jednak te operacje dotyczą zawsze tego samego bloku, zużycie nastąpi już po 3 godzinach.

Zaletą dysków SSD jest to, że gdy zapis staje się niemożliwy, nadal można odczytać dane.

Rozważania związane z przechowywaniem danych

Po omówieniu różnych komponentów systemu przechowywania danych pora zastanowić się, jak najlepiej go skonfigurować. Jest to trudne zadanie, ponieważ nie istnieje jeden prosty sposób na skonfigurowanie tego systemu, sprawdzający się w każdej sytuacji. System SQL Server jest używany do wykonywania bardzo zróżnicowanych zadań, dlatego każdy jego egzemplarz może wymagać zupełnie innej konfiguracji pamięci, dostosowanej do konkretnych wymagań z zakresu

wejścia-wyjścia. W kolejnych podpunktach przedstawiamy zbiór wskazówek, a następnie szczegółowo opisujemy, jak wybrać najlepsze rozwiązanie. Wskazówki możesz wykorzystać w trakcie dyskusji z zespołem odpowiedzialnym za przechowywanie danych.

Wykorzystaj wiedzę producentów

Osobami, do których powinieneś się zwrócić o poradę w obszarze konfigurowania różnych komponentów, są ich producenci. Możliwe jednak, że oni nie będą wiedzieć, jak skonfigurować sprzęt pod kątem systemu SQL Server. Dlatego musisz przekazać im wymagania związane z serwerem SQL Server w zrozumiały sposób. Najlepiej podać w tym celu stosunek liczby operacji zapisu do odczytu i operacji wejścia-wyjścia z dostępem bezpośrednim do operacji sekwencyjnych, wielkość bloków, liczbę operacji wejścia-wyjścia na sekundę, oczekiwaną przepustowość w megabajtach na sekundę, a także minimalne i maksymalne opóźnienie. Te informacje pomogą dostawcy w określeniu optymalnych ustawień dla oferowanych komponentów: dysków, kontrolerów macierzy, kabli, sprzętu sieciowego i innych elementów związanych z przechowywaniem danych.

Każdy system jest inny

Każdy egzemplarz systemu SQL Server ma inne wymagania dotyczące operacji wejścia-wyjścia. Pamiętaj o tym i nie próbuj stosować gotowych schematów w konfiguracji systemu wejścia-wyjścia (chyba że przeprowadziłeś wcześniej analizy i stwierdziłeś, iż Twój egzemplarz i egzemplarz wzorcowy mają dokładnie takie same wymagania).

Im prościej, tym lepiej

W inżynierii od dawna uważa się, że proste rozwiązania są łatwiejsze do zaprojektowania, zbudowania, zrozumienia i konserwacji. W większości przypadków dotyczy to także projektów systemu wejścia-wyjścia. Prostsze rozwiązania działają szybciej, są bardziej niezawodne i wymagają mniej konserwacji niż bardziej skomplikowane projekty. Jeśli nie masz konkretnych powodów do tworzenia skomplikowanego projektu systemu przechowywania danych, zachowaj prostotę.

Testy

Testy są nieodzownym elementem konfigurowania, optymalizowania i dostrajania wydajności. Zbyt często można zetknąć się z klientem, który wyłącznie na podstawie własnych przeczuć lub zasłyszanych gdzieś półprawd jest przekonany, że białe jest czarne.

Jeśli nie masz wyników testów, naprawdę nie wiesz, jak działa podsystem wejścia-wyjścia. Zapomnij o spekulacjach i zgadywaniu. Zaczynj testować system wejścia-wyjścia, aby ustalić, co się w nim dzieje. Często używanym narzędziem do pomiarów podsystemu wejścia-wyjścia i badania charakterystyki jest IOMeter. Generuje on obciążenie symulujące operacje wejścia-wyjścia na dysku lub w sieci i wykonuje pomiary. Można je pobrać ze strony <http://www.iometer.org>.

UWAGA Dobry samouczek pomocny przy poznawaniu narzędzia IOMeter znajdziesz na stronie <http://www.techrepublic.com/article/test-storage-system-performance-with-iometer/5735721>.

Często stosowaną miarą wydajności dysku jest opóźnienie. Można je zmierzyć w monitorze wydajności systemu Windows za pomocą liczników *Średni czas dysku w s/Odczyt*, *Średni czas dysku w s/Zapis* i *Średni czas dysku w s/Transfer*. Oto docelowe wartości opóźnienia dysku.

- **Dziennik transakcji bazy danych.** Mniej niż 5 milisekund (najlepiej 0 milisekund).
- **Dane OLTP.** Mniej niż 10 milisekund.
- **Dane w systemach wspomagających podejmowanie decyzji (OLAP i generowanie raportów).** Mniej niż 25 milisekund.

Po przygotowaniu i przetestowaniu systemu trzeba go monitorować, aby wykryć zmiany w działaniu i wydajności podsystemu wejścia-wyjścia bezpośrednio po ich wystąpieniu. Jeśli masz politykę monitorowania, możesz utworzyć historię wydajności systemu. Będzie ona nieoceniona przy późniejszych analizach wydajności. Bez solidnej historii uzyskanej za pomocą monitorowania danych trudno prześledzić początki wolno zmieniającego się trendu. Szczegółowe informacje o tym, co i jak warto monitorować, znajdziesz w rozdziale 13., „Dostrajanie wydajności kodu w języku T-SQL”.

Macierze RAID

W trakcie zastanawiania się nad tym, ile dysków potrzebujesz, pomyśl o poziomie RAID, który chcesz zastosować. Może on wpłynąć na liczbę dysków niezbędnych do zbudowania systemu przechowywania danych o określonej pojemności i cechach z zakresu operacji wejścia-wyjścia. Rozważ następujące kwestie.

- **Dostępność.** Pierwszym czynnikiem analizowanym w kontekście macierzy RAID jest poziom dostępności danych.
- **Cena.** Ważną kwestią przy tworzeniu każdego systemu jest zmieszczenie się w budżecie. Nie ma sensu tworzyć specyfikacji obejmującej najnowszy, najbardziej wydajny system, jeśli jego cena przekracza budżet 10, 100, a nawet 1000 razy.
- **Przestrzeń fizyczna.** Innym ważnym czynnikiem połączonym z kosztami jest to, ile przestrzeni fizycznej będziesz potrzebował.
- **Wydajność.** Wydajność systemu przechowywania danych to następny istotny aspekt, który pomaga ustalić docelowy poziom RAID.

RAID 0 — przeplatanie bez bloków parzystości i kopii lustrzanych

Macierz RAID 0 obejmuje przynajmniej dwa dyski. W tym podejściu dane są przeplatane między wszystkimi dyskami. Ten poziom RAID nie zapewnia nadmiarowości ani odporności na błędy, ponieważ awaria jednego dysku uszkadza macierz. W trakcie zapisu dane są dzielone na bloki, które są równolegle umieszczane na dyskach.

Pozwala to zwiększyć szybkość odczytu, ponieważ można równolegle wczytywać wiele fragmentów określonych danych. Jednak RAID 0 nie obejmuje kontroli błędów. Ten poziom nie jest zalecany dla systemów SQL Server.

RAID 1 — kopie lustrzane z przeplotem lub blokami parzystości (dwa dyski)

W macierzach RAID 1 jeden dysk jest kopiowany na drugi. To oznacza, że takie macierze muszą obejmować dwa dyski. Ten model zapewnia szybkie działanie, ponieważ dane można wczytywać (choć nie zawsze) z obu dysków, a spadek wydajności przy zapisie jest minimalny. RAID 1 zapewnia nadmiarowość i odporność na błędy, jednak zwiększa koszty przechowywania danych, ponieważ można wykorzystać tylko 50% łącznej pojemności dysków. Z powodu kosztów można stwierdzić, że operacje tworzenia kopii zapasowych i wczytywania danych nie wymagają ochrony na tym poziomie.

RAID 10 — przeplot z kopiami lustrzanymi (przynajmniej cztery dyski)

RAID 10 (inna nazwa to RAID 1+0) polega na zastosowaniu przeplotu dla kopii lustrzanych dysków, co wymaga przynajmniej czterech dysków. Macierz tego typu zawsze składa się z parzystej liczby dysków.

Ogólnie jest to najszybsze rozwiązanie. Macierze RAID 5 (opisane poniżej) zapewniają jednak szybszy odczyt, jeśli obejmują tyle samo dysków. W RAID 10 dane są wczytywane z kilku dysków, a zapis prowadzi do tylko minimalnego spadku wydajności. Ten poziom zapewnia też nadmiarowość — macierz może przetrwać awarię więcej niż jednego dysku, przy czym uszkodzone nie mogą być dyski zawierające te same dane. Wadą jest wyższy koszt, ponieważ można wykorzystać tylko 50% całej dostępnej pamięci.

Systemy bazodanowe wymagające wysokiej wydajności odczytu i zapisu oraz nadmiarowości należy przechowywać w macierzach RAID 10. Z powodu kosztów operacje tworzenia kopii zapasowych, wczytywania danych i inne operacje tylko do odczytu mogą nie wymagać ochrony na tym poziomie. Alternatywą dla macierzy RAID 1+0 jest macierz RAID 0+1, w której tworzone są kopie lustrzane dla dysków z przeplotem (w odróżnieniu od poziomu RAID 1+0, gdzie tworzony jest przeplot dla kopii lustrzanych).

RAID 5 — przeplot z blokiem parzystości (minimum trzy dyski)

W macierzach RAID 5 stosowany jest przeplot z blokiem parzystości i niezbędne są przynajmniej trzy dyski. W trakcie zapisu obliczana jest parzystość danych. Przykładowo dla każdej operacji zapisu w trzydyskowej macierzy dane są umieszczane na dwóch dyskach, a na trzecim dysku zapisywana jest parzystość. Oprogramowanie macierzy RAID rozdziela bloki parzystości między wszystkie dyski, aby uniknąć przeciążenia operacjami zapisu.

Obliczanie parzystości prowadzi do spadku wydajności, dlatego macierze RAID 5 nie są dobrym rozwiązaniem do przechowywania baz danych, które muszą obsługiwać wiele operacji zapisu. Inną wadą macierzy tego typu jest to, że awaria dysku może prowadzić do znacznego spadku wydajności w czasie odtwarzania danych na nowym dysku. W trakcie pracy macierzy z uszkodzonym dyskiem wydajność jest niższa także z powodu konieczności obliczania parzystości przy każdym odczycie danych.

Przy odczycie macierze RAID 5 mogą działać szybciej niż inne macierze RAID, ponieważ grupa dysków może równolegle udostępniać dane. Dlatego RAID 5 to dobry wybór dla baz danych używanych głównie do odczytu (np. w systemach wspomagania podejmowania decyzji). Ponadto macierze RAID 5 są bardziej ekonomiczne od macierzy RAID 1 i RAID 10, ponieważ na każdą macierz wymagają tylko jednego dysku do przechowywania bloków parzystości. W macierzach RAID 1 i RAID 10 nadmiarowe dane zajmują aż 50% przestrzeni.

RAID 6 — przeplot z podwójną parzystością (minimum cztery dyski)

Macierz RAID 6 obejmuje przynajmniej cztery dyski, na których przechowywane są dwie kopie bloków parzystości. Zapewnia to wyższą odporność na błędy, ponieważ awaria nawet dwóch dysków nie musi prowadzić do uszkodzenia danych w macierzy. W tym modelu praktyczne jest tworzenie macierzy obejmujących wiele dysków, ponieważ dyski o dużej pojemności wydłużają czas przywracania systemu po awarii.

Jednak nie należy korzystać z macierzy RAID 6. Wykorzystywana jest w nich macierz z podwójną parzystością. W macierzy RAID 5 używany jest pojedynczy blok parzystości, natomiast tu używane są dwa takie bloki. Choć pozwala to przywrócić dane po awarii dwóch dysków, powoduje też obniżenie wydajności.

Rekomendowane poziomy RAID

Dla plików danych i plików dzienników systemu SQL Server powinieneś stosować szybki i niezawodny system przechowywania. W większości egzemplarzy systemu SQL Server dla takich plików zaleca się stosowanie przeplotu z kopiami lustrzanymi (RAID 10).

Gdy nie można zastosować macierzy RAID 10, należy zbudować macierz RAID 5. Jeśli dana baza jest używana w aplikacji, która wykonuje dużo operacji odczytu w porównaniu z liczbą operacji zapisu, lub gdy baza działa w trybie tylko do odczytu, macierz RAID 5 z dużą liczbą dysków to akceptowalne rozwiązanie.

Ponadto jeśli wiesz, że aplikacja będzie często korzystać z bazy tempdb, zastosuj możliwie szybką i niezawodną pamięć. Może się to wydawać dziwne, ponieważ dane w bazie tempdb są tymczasowe, jednak wymóg niezawodności związany jest z zapewnianiem ciągłości pracy systemu, a nie z obawą o utratę danych. Jeśli dla reszty systemu używana jest niezawodna pamięć, a baza tempdb zapisana jest w innym miejscu, awaria jednego dysku może uniemożliwić pracę systemu SQL Server. Wymóg wysokiej szybkości wynika z tego, że na bazie tempdb wykonywanych jest wiele operacji wejścia-wyjścia z dostępem bezpośrednim.

Pliki binarne systemu operacyjnego i systemu SQL Server mogą być zapisane w zwykłej kopii lustrzanej, choć w wielu sytuacjach czas przestoju, gdy konieczne jest przywrócenie tych systemów, jest akceptowalny. Wtedy wystarczy wykorzystać jeden dysk.

OSTRZEŻENIE Pamiętaj: jeśli nawet czas potrzebny na przywrócenie i zainstalowanie poprawek systemu operacyjnego, zainstalowanie systemu SQL Server i poprawek do niego oraz podłączenie baz danych jest krótszy niż limit czasu przestoju, koszt zatrudnienia pracowników niezbędnych do wykonania wszystkich tych zadań jest znacznie wyższy od kosztu drugiego dysku w macierzy RAID 1.

W systemach o znaczeniu strategicznym pliki binarne systemu operacyjnego i systemu SQL Server powinny znajdować się w kopii lustrzanej w macierzy, przy czym wystarczy jedna taka kopia. Dla plików tego rodzaju wymogi związane z operacjami wejścia-wyjścia nie są wysokie. Pliki te są zwykle wczytywane raz w momencie uruchamiania aplikacji. Potem nie trzeba ich modyfikować do momentu, gdy potrzebne będą nowe ścieżki kodu. Wtedy następuje odczyt bezpośredni kilku kolejnych porcji danych po 4 kilobajty, dlatego stosowanie szybkich dysków nie jest tu konieczne.

Podsumowanie

Dostrajanie wydajności bywa skomplikowane, gdy system przez dłuższy czas działa poniżej możliwości. Problemy z wydajnością narastają, jeśli system od początku nie został zaprojektowany w optymalny sposób.

Zanim zaczniesz modyfikować sprzęt, powinieneś zastanowić się nad wieloma obszarami, takimi jak interakcja użytkowników z systemem, wzorce użytkowania danych, liczba i rodzaj instrukcji SQL-a uruchamianych w systemie, schemat itd.

W trakcie konfigurowania serwera pod kątem optymalnej wydajności trzeba podjąć szereg decyzji dotyczących sprzętu. Procesor serwera ma duży wpływ na to, czy system będzie akceptowalnie działał przy zakładanym obciążeniu. Należy zastanowić się nad komponentami

procesora (pamięcią podręczną, technologią Hyper-Threading, wielordzeniowością i architekturą), a następnie rozważyć dostępne opcje. Pomyśl też o pamięci i różnych związanych z nią mechanizmach. Te mechanizmy to m.in. fizyczna i wirtualna przestrzeń adresowa, menedżer pamięci wirtualnej i plik stronicowania.

Najwolniejszą częścią systemu jest podsystem wejścia-wyjścia. Należy starannie wybrać jedną z wielu możliwości dostępnych przy projektowaniu podsystemów sieci i przechowywania danych. Oto niektóre z pytań, na które trzeba udzielić odpowiedzi. Jak szybka powinna być sieć? Ile przestrzeni dyskowej jest potrzebne? Ile dysków i jakiego rodzaju należy użyć? Jak szybkie mają być te dyski? Czy należy zastosować architekturę SAN, czy NAS? Jakiego typu macierz RAID wykorzystać? Jakich adapterów pamięci użyć? Jaka powinna być wielkość jednostek alokacji pamięci?

Znasz już wiele narzędzi, które możesz wykorzystać w trakcie konfigurowania systemu SQL Server w celu uzyskania optymalnej wydajności.

Z rozdziału 11. dowiesz się, jak optymalizować system SQL Server.

Optymalizowanie systemu SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Korzyści płynące z optymalizowania wydajności aplikacji.
- Wykorzystanie partycji i kompresji do poprawy wydajności.
- Dostrajanie operacji wejścia-wyjścia, procesora i pamięci w celu przyspieszenia przetwarzania kwerend.

Konfigurowanie systemu SQL Server to tylko jeden z aspektów jego konserwowania. W każdej wersji tego systemu pojawiają się nowe funkcje (głównie w edycji Enterprise). System SQL Server wyposażony został w dodatkowe funkcje, które pozwalają na uzyskanie wyższej skalowalności i optymalizację pracy w określonych obszarach. Podział na partycje, kompresja, rozszerzenia puli buforów, właściwa konfiguracja plików danych i plików dziennika, a nawet parametry kwerend mogą mieć istotny wpływ na poprawne i wydajne działanie systemu SQL Server.

Optymalizowanie aplikacji

Wydajność systemu SQL Server można podnieść na wiele sposobów. Jednym z nich jest upewnienie się, że aplikacja działa optymalnie. Dlatego pierwszym zadaniem w ramach skalowania systemu SQL Server 2014 w systemie Windows Server jest optymalizacja aplikacji. Jeśli aplikacja jest źle napisana, wymiana sprzętu na lepszy tylko odroczy wystąpienie problemów ze skalowalnością, zamiast rozwiązać sytuację. Tu zapoznasz się z dostrajaniem aplikacji w obszarze obciążenia procesora i współbieżności, natomiast omawianie dostrajania konkretnych aplikacji wykracza poza zakres tego rozdziału.

Celem dostrajania wydajności systemu SQL Server 2014 jest zminimalizowanie czasu przetwarzania instrukcji SQL-a i zwiększenie przepustowości systemu. Pomaga to zmaksymalizować skalowalność całego serwera bazodanowego dzięki skróceniu czasu przekazywania danych w sieci, a także zoptymalizowaniu szybkości operacji wejścia-wyjścia i czasu wykonywania zadań przez procesor.

Definiowanie obciążenia

Warunkiem wstępnym przy dostrajaniu środowiska bazodanowego jest dokładne zrozumienie podstawowych aspektów dotyczących bazy danych. Dwa najważniejsze aspekty to logiczna i fizyczna struktura danych oraz naturalne różnice związane z zastosowaniami określonej bazy. Przykładowo różne są wymagania stawiane środowisku OLTP i środowisku DSS. W środowisku DSS często potrzebny jest wysoce zoptymalizowany podsystem wejścia-wyjścia umożliwiający odczyt bardzo dużych ilości danych. W transakcyjnym środowisku OLTP podsystem wejścia-wyjścia powinien być zoptymalizowany zarówno pod kątem odczytu, jak i zapisu informacji.

W większości sytuacji nie da się przetestować systemu SQL Server w celu zbadania wymagań aplikacji w środowisku produkcyjnym. Zalecanym rozwiązaniem jest przygotowanie środowiska testowego możliwie zbliżonego do systemu produkcyjnego i wykorzystanie generatora obciążenia, np. programów Quest Benchmark Factory, Idera SQLscaler lub wbudowanego narzędzia Distributed Replay Adapter (jest ono dostępne tylko w edycjach Developer i Enterprise). Za pomocą takich narzędzi można zasymulować obciążenie bazy w warstwie danych docelowego środowiska produkcyjnego. Technika ta umożliwia wykonanie pomiarów poza środowiskiem produkcyjnym i dostrojenie systemu bazodanowego przed przeniesieniem go do tego środowiska.

Ponadto zastosowanie opisanej techniki w środowisku testowym umożliwia podział całego rozwiązania na elementy i niezależne ich sprawdzenie. I tak przy użyciu generatora obciążenia można ograniczyć nieznaną i zmienne czynniki wpływające na wydajność oraz przetestować osobno każdy komponent (sprzęt, bazę danych i aplikację).

Cichy zabójca — problemy z operacjami wejścia-wyjścia

Niewłaściwa konfiguracja plików danych i plików dziennika może być równie kosztowna jak używanie fizycznych dysków o niskiej liczbie operacji wejścia-wyjścia na sekundę. W przeszłości za przyczynę problemów z wydajnością zawsze uznawano operacje wejścia-wyjścia. W ostatnich pięciu latach coraz częściej za powód niskiej wydajności uznaje się brak pamięci lub wolny procesor. Jednak operacje wejścia-wyjścia nadal mogą być źródłem problemów, jeśli konfiguracja jest niewłaściwa lub występują kłopoty w podsystemie dyskowym.

Model obsługi wejścia-wyjścia w systemie SQL Server

System SQL Server 2014 współdziała z systemem operacyjnym Windows Server 2012 przy obsłudze żądań wejścia-wyjścia kierowanych do dysku. Menedżer wejścia-wyjścia w systemie Windows Server obsługuje wszystkie operacje wejścia-wyjścia i przetwarza żądania odczytu lub zapisu za pomocą metody rozdzielał-skal (ang. *scatter-gather*) lub asynchronicznie. Metoda rozdzielał-skal obejmuje proces zbierania i rozdzielania danych na dysku lub w buforze.

Silnik przechowywania danych systemu SQL Server zarządza czasem wykonywania operacji wejścia-wyjścia, sposobem ich przeprowadzania, a także ich liczbą. Jednak to system operacyjny

Windows (a dokładnie podsystem menedżera wejścia-wyjścia) przeprowadza same operacje wejścia-wyjścia i zapewnia interfejs do komunikacji z fizycznymi nośnikami.

Zadaniem silnika przechowywania jest zarządzanie operacjami wejścia-wyjścia i maksymalne ograniczanie ich kosztów. Silnik przechowywania np. przeznaczają dużą część pamięci wirtualnej na pamięć podręczną danych. Zarządzanie pamięcią podręczną odbywa się na podstawie analizy kosztów, a jej celem jest zapewnienie optymalnego wykorzystania dostępnej pamięci na dane. W pamięci podręcznej mają znajdować się dane, które są często aktualizowane lub żądane. Przyspiesza to obsługę żądań użytkowników, ponieważ wystarczy wykonać logiczne operacje wejścia-wyjścia i można uniknąć kosztownych operacji fizycznych.

Technologia In-Memory OLTP to ważne usprawnienie systemu SQL Server, które umożliwia optymalizację pracy serwerów i działającego w nich kodu, a także zmniejszenie współzawodnictwa o zasoby wejścia-wyjścia.

Pamiętaj jednak, że gdy korzystasz z technologii In-Memory OLTP, musisz poza pulą buforów zarezerwować pamięć na struktury przechowywane w pamięci. Ilość tej pamięci powinna być mniej więcej cztery razy większa od wielkości tych struktur (jeśli nie są trwałe).

UWAGA Więcej informacji na temat technologii In-Memory OLTP podano w rozdziale 9, „Technologia In-Memory OLTP”.

Rozmieszczanie plików bazy danych

System SQL Server przechowuje bazy danych jako pliki w systemie operacyjnych. Używane są przy tym fizyczne dyski lub jednostki LUN (ang. *Logical Unit Number*) wydzielone z macierzy dyskowej. Bazy danych obejmują trzy rodzaje plików: główny plik danych (.mdf), przynajmniej jeden pomocniczy plik danych (.ndf) i pliki dziennika transakcji (.ldf).

UWAGA W systemie SQL Server 2014, podobnie jak w niektórych wcześniejszych wersjach, rozszerzenia .mdf, .ndf i .ldf są opcjonalne.

Lokalizacja plików bazy danych ma bardzo duży wpływ na wydajność operacji wejścia-wyjścia w systemie DBMS. Używanie szybkiego dedykowanego podsystemu wejścia-wyjścia na pliki bazy danych pozwala uzyskać maksymalną wydajność. W rozdziale 10., „Konfigurowanie serwera pod kątem optymalnej wydajności”, napisaliśmy, że duża pojemność dysków nie zapewnia wyższej wydajności. Natomiast im szybsze są fizyczne dyski (np. dyski SSD), tym wydajniej może działać podsystem wejścia-wyjścia bazy danych. Dane w zależności od sposobu ich użytkowania można przechowywać w plikach danych lub grupach plików zapisanych na wielu fizycznych dyskach. *Grupa plików* to zbiór plików danych używanych do zarządzania rozmieszczeniem takich plików.

UWAGA Aby zmaksymalizować wzrost wydajności, umieść poszczególne pliki danych i pliki dzienników na odrębnych fizycznych jednostkach LUN. Dane archiwalne lub rzadko aktualizowane informacje możesz umieścić w grupie plików przeznaczonych tylko do odczytu. Takie grupy mogą znajdować się na wolniejszych jednostkach LUN, ponieważ są rzadko używane. Pozwala to zwolnić miejsce na dysku i zasoby, dzięki czemu reszta bazy danych może działać szybciej.

Kwestie związane z bazą tempdb

Ponieważ lokalizacja plików bazy danych ma tak duży wpływ na wydajność operacji wejścia-wyjścia, warto rozważyć zmiany w bazie tempdb w trakcie tworzenia strategii rozmieszczania plików danych. Wydajność bazy tempdb ma stosunkowo duży wpływ na wydajność systemu, ponieważ jest to najczęściej modyfikowana baza i dlatego powinna działać jak najszybciej.

Baza tempdb, podobnie jak wszystkie inne bazy, zwykle składa się z pliku danych i pliku dziennika. W tej bazie przechowywane są obiekty użytkowników i obiekty wewnętrzne. W bazie tempdb używa się dwóch magazynów wersji. *Magazyn wersji* (ang. *version store*) to zbiór stron danych przechowujących wiersze potrzebne do obsługi określonych mechanizmów kontroli wersji wierszy. Dostępne są dwa magazyny wersji zawierające:

- wersje wierszy w bazie tempdb wygenerowane przez transakcje modyfikujące dane z poziomami izolacji READ_COMMITTED_SNAPSHOT lub ALLOW_SNAPSHOT_ISOLATION;
- nieprzetworzone wiersze w bazie tempdb wygenerowane przez transakcje modyfikujące dane dla takich mechanizmów jak operacje na indeksach w trybie ONLINE, zbiory wyników MARS (ang. *Multiple Active Result Sets*) i wyzwalacze AFTER.

Od wersji SQL Server 2005 aż po wersję SQL Server 2014 baza tempdb obsługuje liczne mechanizmy tworzące obiekty użytkowników i obiekty wewnętrzne lub magazyny wersji, takie jak:

- zmienne tabelowe, tabele tymczasowe i globalne tabele tymczasowe,
- kwerendy,
- wyzwalacze,
- poziomy izolacji READ_COMMITTED_SNAPSHOT i ALLOW_SNAPSHOT_ISOLATION,
- zbiory wyników MARS,
- tworzenie indeksów w trybie ONLINE,
- tabele tymczasowe, zmienne tabelowe i funkcje tabelowe,
- testy w postaci instrukcji DBCC (ang. *Database Console Command*),
- parametry obiektów LOB,
- kursory,
- Service Broker i powiadomienia o zdarzeniach,
- zmienne XML-owe i typów LOB,
- powiadomienia kwerend,
- system Database Mail,
- tworzenie indeksów,
- funkcje zdefiniowane przez użytkownika.

Dlatego umieszczenie bazy tempdb w dedykowanym i bardzo szybkim podsystemie wejścia-wyjścia pozwala zapewnić wysoką wydajność. Producenci włożyli dużo pracy w wewnętrzne mechanizmy działania tej bazy, aby poprawić jej skalowalność.

UWAGA Warto zapoznać się z artykułem „Capacity Planning for tempdb” z dokumentacji Books Online. Znajdziesz tam dodatkowe informacje i szczegółowe omówienie funkcji związanych z użytkowaniem bazy tempdb. Artykuł jest dostępny na stronie <http://msdn.microsoft.com/en-us/library/ms345368.aspx>.

Przy ponownym uruchamianiu systemu SQL Server tempdb jest jedyną bazą, która przyjmuje pierwotny domyślny rozmiar 8 megabajtów (lub inną wielkość zdefiniowaną przez administratora). Następnie może się rozrastać w zależności od potrzeb. W trakcie automatycznego powiększania bazy wątki mogą zablokować jej zasoby, co obniża współbieżność. Aby uniknąć przestojów, należy ustawić tempo automatycznego rozrastania się bazy odpowiednio do środowiska. Zwykle to tempo powinno być na tyle duże, aby kwerendy mogły działać bez dodatkowego powiększania bazy. Wielkość, o jaką powiększana jest baza, należy ustawić przynajmniej na 1 gigabajt. Jeśli włączone jest natychmiastowe inicjowanie plików, przy powiększaniu plików danych nie będą one blokowane.

Jeśli zauważysz, że pliki bazy tempdb rosną między ponownymi uruchomieniami systemu, ustaw domyślną wielkość plików na maksymalny rozmiar, jaki osiągają. Dla każdego pliku ustaw tę samą wielkość.

Powinieneś oszacować potrzebną pojemność bazy tempdb, aby właściwie ustalić jej rozmiar i umożliwić obsługę wymagań korporacyjnego systemu, którym się zajmujesz. Wykonaj przynajmniej następujące czynności.

1. Uwzględnij wielkość istniejącej bazy tempdb.
2. Obserwuj bazę tempdb w czasie działania procesów, które w największym stopniu wpływają na tę bazę. Poniższa kwerenda wyświetla pięć wykonywanych zadań, które zajmują najwięcej miejsca w tej bazie:


```
SELECT top 5 * FROM sys.dm_db_session_space_usage
ORDER BY (user_objects_alloc_page_count + internal_objects_alloc_page_count) DESC
```
3. Odtwórz indeks największej aktywnej tabeli i obserwuj przy tym bazę tempdb. Nie zdziw się, jeśli zajmowana pamięć okaże się dwa razy większa od pojemności tej tabeli. Jest tak, ponieważ odtwarzanie indeksu odbywa się w bazie tempdb.

Poniżej przedstawiona jest zalecana kwerenda, którą należy regularnie uruchamiać, aby monitorować wielkość bazy tempdb. Radzimy, aby wywoływać tę kwerendę przynajmniej co tydzień. Kwerenda sprawdza i zwraca przestrzeń (w kilobajtach) zajmowaną w bazie tempdb przez wewnętrzne obiekty, puste miejsca, magazyn wersji i obiekty użytkowników.

```
select sum(user_object_reserved_page_count)*8 as user_objects_kb,
       sum(internal_object_reserved_page_count)*8 as internal_objects_kb,
       sum(version_store_reserved_page_count)*8 as version_store_kb,
       sum(unallocated_extent_page_count)*8 as freespace_kb
from sys.dm_db_file_space_usage
where database_id = 2
```

Dane wyjściowe, jakie uzyskasz w swoim systemie, zależą od konfiguracji i użytkowania bazy danych. Oto przykładowe dane wyjściowe z tej kwerendy:

user_objects_kb	internal_objects_kb	version_store_kb	freespace_kb
256	640	0	6208

UWAGA Jeśli dla któregoś z tych wewnętrznych obiektów systemu SQL Server lub magazynów danych zabraknie pamięci, zabraknie jej też dla bazy tempdb i system SQL Server przestanie działać. Więcej informacji na ten temat znajdziesz w dokumentacji Books Online w artykule poświęconym zajmowaniu przestrzeni dyskowej przez omawianą bazę (<http://msdn.microsoft.com/en-us/library/ms176029.aspx>).

Korzystając z uzyskanych wyników, należy skonfigurować bazę tempdb w następujący sposób.

- Na podstawie wyników testów przydziel wstępnie pamięć na pliki bazy tempdb, ale pozostaw włączoną opcję automatycznego powiększania bazy, by uniknąć zatrzymania systemu SQL Server. Przyda się ona, gdy bazie tempdb zabraknie miejsca.
- Zgodnie z ogólną regułą w każdym egzemplarzu systemu SQL Server utwórz jeden plik danych bazy tempdb na procesor lub rdzeń procesora. Pliki te powinny mieć identyczną wielkość i nie powinno ich być więcej niż osiem. Choć utworzenie większej liczby plików może czasem być korzystne, nie należy tego robić, chyba że występuje niepożądane współzawodnictwo o zasoby bazy tempdb.
- Upewnij się, że baza tempdb działa w trybie prostego przywracania, co umożliwia odzyskanie pamięci.
- Umieść bazę tempdb na szybkim i dedykowanym podsystemie wejścia-wyjścia.
- Używaj natychmiastowego inicjowania plików. Więcej informacji na temat konfigurowania tego mechanizmu zamieściliśmy w rozdziale 10.

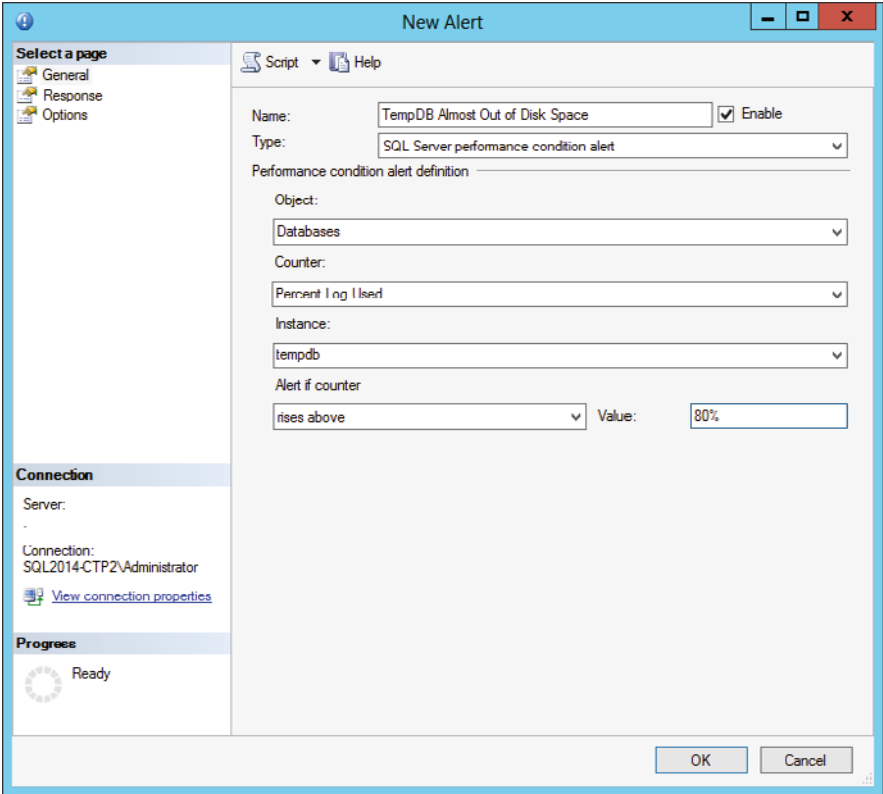
Zwykle należy utworzyć alerty powiązane nie tylko z plikami danych i dziennikami bazy tempdb, ale też z wszystkimi plikami każdej bazy danych z systemu. Można wykorzystać do tego narzędzie SQL Server Agent lub Microsoft System Center Operations Manager z pakietem dla systemu SQL Server i wykrywać błędy 1101 lub 1105 (informujące o zapelnieniu bazy tempdb). Jest to bardzo ważne, ponieważ w reakcji na zgłoszenie tych błędów serwer kończy pracę. Kliknij prawym przyciskiem myszy węzeł *SQL Server Agent* w programie SQL Server Management Studio i uzupełnij okno dialogowe widoczne na rysunku 11.1. Ponadto możesz w monitorze wydajności śledzić wartości następujących liczników.

- **SQLServer:Databases — Log File(s) Size(KB)**. Zwraca łączną wielkość wszystkich plików dziennika z bazy danych.
- **SQLServer:Databases — Data File(s) Size(KB)**. Zwraca łączną wielkość wszystkich plików danych z bazy.
- **SQLServer:Databases — Log File(s) Used(KB)**. Zwraca łączną zajęta przestrzeń we wszystkich plikach dziennika z bazy danych. Jeśli zajęta jest duża część dziennika bazy danych, może to oznaczać, że długa transakcja uniemożliwia uporządkowanie pliku dziennika.

Innym doskonałym narzędziem jest widok DMV `sys.dm_db_task_space_usage`. Pozwala on zbadać wykorzystanie przestrzeni w bazie tempdb dla poszczególnych zadań. Pamiętaj, że po wykonaniu zadania powiązane z nim liczniki są zerowane. Ponadto dla poszczególnych dysków można śledzić średni czas odczytu i zapisu. Oto interpretacja wyników.

- Mniej niż 10 milisekund — bardzo dobrze.
- Od 10 do 20 milisekund — wartości graniczne.
- Od 20 do 50 milisekund — powolne operacje; należy to zbadać.
- Powyżej 50 milisekund — poważny zator w operacjach wejścia-wyjścia.

UWAGA Jeśli system ma duże wymagania związane z użytkowaniem bazy tempdb, zapoznaj się z artykułem Q917047 („Microsoft SQL Server I/O subsystem requirements for tempdb database”, <http://support.microsoft.com/kb/917047>) lub poszukaj artykułu „Optimizing tempdb Performance” w dokumentacji Books Online systemu SQL Server 2012.



Rysunek 11.1. Konfigurowanie nowego alertu

Oto jedno z najczęściej zadawanych pytań dotyczących bazy tempdb: „Ile plików powinienem zastosować?”. Podobnie jak w przypadku większości pytań w świecie systemu SQL Server odpowiedź brzmi: „To zależy”.

Utwórz jeden plik na rdzeń procesora, ale nie więcej niż osiem plików. Microsoft przeprowadził rozbudowane testy, w których wykazano, że używanie powyżej ośmiu plików na grupę daje coraz mniejsze korzyści (choć istnieją wyjątki). W tabeli 11.1 znajdziesz wskazówki dotyczące zalecanej liczby plików.

Tabela 11.1. Określanie liczby plików danych

Liczba rdzeni procesora	Liczba plików danych bazy tempdb
1	1
2	2
4	4
8	8
16	8
32	8

Wewnętrzne mechanizmy systemu SQL Server i alokacja plików

Przyczyną tworzenia kilku plików tempdb dla baz użytkownika w systemie SQL Server jest sposób alokacji danych. Na bardzo ogólnym poziomie używane są tabele, wiersze i kolumny. Jednak wewnątrz systemu SQL Server występują następujące jednostki.

- **Strony.** Dane w systemie SQL Server są przechowywane w stronach mających po 8 kilobajtów. W tym systemie istnieje wiele rodzajów stron: z danymi, z indeksami, z drzewem tekstowym i różne typy stron z wewnętrznymi metadanymi. Każda strona zawiera 96-bajtowy nagłówek, po którym zapisane są dane.
- **Ekstenty.** W systemie SQL Server ekstent to grupa ośmiu przyległych stron. Istnieją dwa rodzaje ekstentów.
 - **Ekstenty niejednorodne.** Jest to pierwszych osiem stron alokowanych w momencie tworzenia tabeli lub indeksu w dowolnym miejscu w grupie plików. To oznacza, że gdy stworzysz dużą tabelę temp, zwykle zgłaszanych jest przynajmniej osiem wywołań do strony SGAM (ang. *Shared Global Allocation Map*) w celu alokacji ekstentów niejednorodnych. Ekstenty niejednorodne mogą też należeć do różnych obiektów.
 - **Ekstenty dedykowane (jednorodne).** Te ekstenty są tworzone po zaalokowaniu pierwszych ośmiu stron. Wszystkie kolejne ekstenty dotyczą odrębnych obiektów. Ekstent reprezentuje osiem stron zaalokowanych dla obiektu, nawet jeśli w danym momencie nie wszystkie te strony są potrzebne. Te ekstenty są alokowane za pomocą stron GAM (ang. *Global Allocation Map*).

Mapy alokacji zawierają informacje o alokacji pamięci i danych na stronach, w tabelach i w obiektach z metadanymi używających ekstentów. Oto trzy najważniejsze mapy alokacji związane z bazami danych (w tym z bazą tempdb).

- **Page Free Space (PFS).** Rejestruje zaalokowane strony i stan stron. Na stronach PFS każdy bajt zawiera informacje o jednej stronie. Strona PFS zawsze jest pierwszą stroną i występuje co 8088 stron (mniej więcej co 64 megabajty).
- **Global Allocation Map (GAM).** Rejestruje dedykowane ekstenty. Każdy ekstent jest reprezentowany przez 1 bit. Jest drugą stroną w bazie i występuje co 64 000 ekstentów (czyli mniej więcej co 4 gigabajty). Strona GAM jest alokowana do 511 230 stron.
- **Shared Global Allocation Map (SGAM).** Rejestruje ekstenty niejednorodne. Każdy ekstent jest reprezentowany przez 1 bit. Jest trzecią stroną w bazie i występuje co 64 000 ekstentów (czyli mniej więcej co 4 gigabajty). Strona SGAM jest alokowana do 511 230 stron.
- **Index Allocation Map (IAM).** Rejestruje wszystkie inne strony zaalokowane dla konkretnego obiektu. Strony IAM są tworzone dla danych używających jednostek alokacji IN_ROW_DATA, ROW_OVERFLOW_DATA i LOB_DATA. Z wszystkimi tabelami, stertami lub indeksami powiązanych może być wiele stron IAM. Dla każdej partycji obiektu tworzone są nie więcej niż trzy strony IAM. Strony IAM nie są współużytkowane między obiektami.

Pojęcie *współzawodnictwo o strony GAM* w kontekście bazy tempdb dotyczy stron tej właśnie mapy alokacji. Tworzenie i usuwanie wielu niepowtarzalnych obiektów tymczasowych może wymagać wielu plików. Większa liczba plików umożliwia rozłożenie obciążenia mapy alokacji między takie mapy z poszczególnych plików.

Dla baz tempdb (inaczej niż dla innych baz) można utworzyć tylko jedną grupę plików. Dla baz użytkowników można tworzyć grupy plików zdefiniowane przez użytkownika i określać lokalizację opracowywanych obiektów. To pozwala ograniczyć współzawodnictwo o zasoby w bazach użytkowników. Bazy tempdb nie obsługują tej techniki.

Poniżej opisane są dwa algorytmy używane do alokowania pamięci we wszystkich plikach danych systemu SQL Server.

- Algorytm *zapełniania proporcjonalnego* (ang. *proportional fill*) określa, ile danych jest zapisanych w każdym pliku grupy. Robi to na podstawie stosunku pustej przestrzeni do zapełnionej. Dzięki temu wszystkie pliki są zapełniane w mniej więcej tym samym czasie (jeśli mają podobną wielkość).
- W algorytmie *cyklicznym* (ang. *round robin*) po kolei wybierane są nowe pliki, których poziom zapełnienia nie przekracza określonego limitu. Gdy nie ma już takich plików, jeden z nich trzeba powiększyć.

Dlaczego te algorytmy mają znaczenie? Gdy tworzona jest nowa tabela, wykonywane są następujące operacje.

1. Wywołanie instrukcji `create table`.
2. Dodanie wpisu do strony SGAM, co prowadzi do zarezerwowania miejsca na stronie IAM przeznaczoną dla danego obiektu (tu jest nim tabela).
3. Dla tej strony IAM dodawany jest wpis na stronie PFS, co pozwala śledzić zapełnienie danej strony IAM.
4. Wywoływana jest instrukcja `insert` w celu utworzenia nowej strony danych.
5. W mapie SGAM dodawany jest wpis w celu zaalokowania jednej strony.
6. Na stronie IAM tworzony jest wpis umożliwiający śledzenie nowej strony.
7. Na stronie PFS tworzony jest wpis umożliwiający śledzenie nowej strony.
8. Nowa strona danych zostaje utworzona.

Kroki od 4. do 8. są wykonywane wielokrotnie do momentu utworzenia pierwszych ośmiu stron. Następnie zamiast strony SGAM używana jest strona GAM i za każdym razem rezerwowanych jest osiem stron dedykowanych ekstentów.

W bazie tempdb (gdzie tabele są nieustannie tworzone i usuwane) opisany proces może prowadzić do zatoru z powodu współzawodnictwa o strony PFS, SGAM i GAM. Każdy plik danych dodawany do bazy zawiera własne strony tego rodzaju. Ponieważ dane są alokowane równomiernie w wielu plikach, pozwala to ograniczyć współzawodnictwo o mapy alokacji.

Należy pamiętać, że system stara się utrzymać podobny rozmiar plików, aby dane były równomiernie rozmieszczone w plikach z grupy. Jeśli jeden plik będzie większy od pozostałych, zacznie się nieproporcjonalnie rozrastać. W bazie tempdb (która alokuje strony za każdym razem, gdy tworzony jest obiekt tymczasowy, i przeprowadza dezałokację w momencie usuwania obiektu) jest to szczególnie kosztowne i grozi nieproporcjonalną alokacją danych w jednym pliku.

Podział tabel i indeksów na partycje

Podział na partycje polega na rozbijaniu dużych obiektów (np. tabel) na mniejsze, łatwiejsze w zarządzaniu fragmenty. Podział na partycje oparty jest na *wierszach*.

Partycje są używane już od pewnego czasu. Technologię wprowadzono w formie widoków DPV (ang. *Distributed Portioned View*) w czasie wprowadzania systemu SQL Server 7.0. Nowe mechanizmy umożliwiały optymalizatorowi wyeliminowanie partycji (lub tabel) złączonych za pomocą instrukcji `UNION ALL`. Takie partycje można też było rozdzielać między połączone serwery. Podobnie działa mechanizm Sharding, który wywołuje kwerendy z wykorzystaniem widoku DPV z kilku połączonych serwerów. Z powodu kosztów pracy w sieci technika ta może okazać się wolniejsza niż wywoływanie takich samych kwerend dla tabel z lokalnego serwera. Ponieważ systemy stają się coraz szybsze i wydajniejsze, obecnie preferowanym rozwiązaniem jest używanie mechanizmów systemu SQL Server służących do podziału tabel bazy danych i ich indeksów na partycje przechowywane w grupie plików w ramach jednej bazy. Ważne jest to, że taki podział na partycje wymaga edycji Enterprise systemu SQL Server, natomiast widoki DPV są obsługiwane także w innych wersjach tego systemu.

Jednym z najważniejszych aspektów pozwalających na zwiększanie wydajności systemu za pomocą partycji jest możliwość wykonywania instrukcji `alter table... switch`.

Za pomocą instrukcji `switch` języka T-SQL można przypisać dane z jednej tabeli do innej na poziomie mapy alokacji. Zamiast wykonywać operację wstawiania w jednej tabeli i usuwania w innej, zajmować liczne blokady i zamki, tworzyć w dzienniku transakcji rekord dla każdej instrukcji wstawiania, aktualizowania lub usuwania użytkowników i metadanych, wystarczy zmodyfikować zajmujące 8 kilobajtów mapy alokacji określające lokalizację danych.

Pamiętaj, że dane można przypisywać w ten sposób tylko do pustych partycji. Ponadto docelowa tabela musi mieć dokładnie ten sam schemat, co tabela źródłowa.

Po co stosować partycje?

Duże tabele tworzy się z różnych powodów. Gdy tabela (lub baza danych) osiąga określony rozmiar, zarządzanie nią zaczyna sprawiać problemy, ponieważ konserwacja bazy, tworzenie kopii zapasowych i przywracanie stanu zajmuje dużo czasu. Ponadto występują problemy w środowisku, np. niski poziom współbieżności spowodowany dużą liczbą użytkowników rozbudowanej tabeli i licznymi blokadami, co prowadzi do dalszych kłopotów.

Przy stosowaniu partycji wielkość danych ma znaczenie. Nie należy dzielić na partycje tabel zajmujących mniej niż 25 gigabajtów. Jeśli tabela jest mała, przyrost wydajności będzie zbyt niski, a generowane będą dodatkowe obciążenia. Powinieneś opracować plan podziału na partycje dla danych, do których chcesz mieć szybki dostęp. Przygotuj też strategię archiwizowania takich danych.

Jeśli nadal nie masz pewności, czy chcesz zastosować partycje, sprawdź wykonywane zadania w narzędziu DTA (ang. *Database Tuning Advisor*). Narzędzie wyświetla zalecenia dotyczące podziału na partycje i generuje potrzebny kod. Omówienie programu DTA znajdziesz w rozdziale 15., „Replikacja”, a sam program jest dostępny w sekcji *Performance Tools* menu systemu Microsoft SQL Server 2014.

UWAGA Partycje są opisane w różnych rozdziałach tej książki, zatem możesz zapoznać się ze szczegółami tej operacji w odpowiednim kontekście.

Poniżej znajdziesz ogólny opis procesu podziału na partycje.

1. Utwórz funkcję partycjonującą i zdefiniuj w niej strategię rozmieszczania danych.
2. Przygotuj grupy plików dostosowane do funkcji partycjonującej.
3. Utwórz schemat partycji, aby zdefiniować strategię fizycznego podziału danych (powiąż dane z funkcji z grupami plików).
4. Utwórz tabelę lub indeks w funkcji partycjonującej.
5. Przeanalizuj plany wykonywania kwerend, aby sprawdzić, które partycje są używane. Ustal, czy zwracany jest tylko podzbiór danych.

Partycje mogą pozytywnie wpłynąć na pracę środowiska i przebieg większości procesów. Koniecznie dobrze zapoznaj się z tym mechanizmem, aby mieć pewność, że będzie przydatny we wszystkich procesach. W poniższej liście znajdziesz wybrane procesy, na które może wpłynąć zastosowanie partycji.

- Strategia tworzenia kopii zapasowej i przywracania bazy danych (możliwość obsługi częściowej dostępności bazy danych).
- Strategia konserwacji indeksów (ich odtwarzania), w tym widoków indeksowanych.
- Strategia zarządzania danymi (duże operacje wstawiania lub usuwania danych w tabelach).
- Obciążenie bazy użytkownika końcowego.
- Współbieżność:
 - **Równoległe przetwarzanie kwerend kierowanych do partycji.** Gdy w systemie SQL Server 2005 kwerenda dotyczy tylko jednej partycji tabeli, wszystkie dostępne wątki procesora mają dostęp do tej partycji i mogą pracować równolegle. Jeśli kwerenda dotyczy więcej niż jednej partycji, dla każdej z nich używany jest tylko jeden wątek procesora, nawet gdy dostępnych jest więcej procesorów. Załóżmy, że w serwerze działa osiem procesorów, a kwerenda używa dwóch partycji tabeli. W takiej sytuacji sześć procesorów nie jest używanych przez kwerendę. W wersji SQL Server 2012 dodano równoległe przetwarzanie partycji, zatem wszystkie dostępne procesory są stosowane równolegle do obsługi kwerendy dotyczącej tabeli podzielonej na partycje. W zależności od wymagań mechanizm równoległego przetwarzania można włączyć lub wyłączyć.
 - **Blokada partycji tabeli.** Możliwe jest tworzenie blokad na poziomie partycji przed przejściem na poziom tabeli. Domyślnie ten mechanizm nie jest włączony, jednak w każdej tabeli podzielonej na partycje należy go aktywować. Aby włączyć ten mechanizm, wywołaj następującą instrukcję:

```
ALTER TABLE TableName SET (LOCK_ESCALATION=AUTO)
```
 - **System SQL Server 2014.** W tej wersji dodano możliwość przyrostowego generowania i odtwarzania statystyk dla pojedynczych partycji.
- Większe możliwości tworzenia rozproszonych lub izolowanych operacji z wykorzystaniem grup plików.

Tworzenie funkcji partycjonującej

Funkcja partycjonująca określa podstawową strategię podziału danych na partycje. W trakcie tworzenia takiej funkcji należy skoncentrować się na ustaleniu tej strategii. Najlepszym sposobem pracy nad strategią jest zidentyfikowanie zadań i określenie ich priorytetów. Niezależnie od tego, czy chcesz przenieść stare dane z tabeli do wolniejszego i taniego podsystemu wejścia-wyjścia, zwiększyć współbieżność operacji w bazie danych lub przeprowadzić konserwację dużej tabeli, musisz zidentyfikować zadania i ustalić ich priorytety. Po opracowaniu strategii utwórz dopasowaną do niej funkcję partycjonującą.

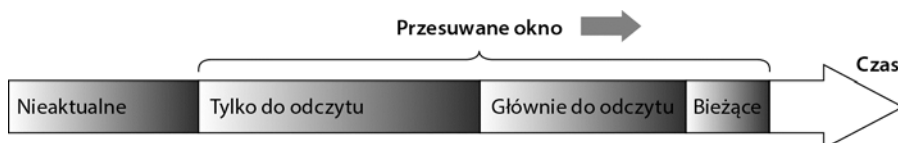
Pamiętaj, aby przeanalizować tabelę pod kątem podziału na partycje, ponieważ funkcja partycjonująca pracuje zależnie od podziału danych (istotne są selektywność kolumny i przedział przechowywanych w niej wartości). Od przedziału wartości zależy, na ile partycji można podzielić daną tabelę. Maksymalna liczba partycji tabeli to 15 000.

Każdy dodatkowy plik bazy danych wydłuża czas jej uruchamiania. Gdy system SQL Server rozpoczyna pracę, musi otworzyć nagłówek każdego pliku i wczytać go do pamięci. Przykładowo 1000 plików bazy danych może wydłużyć czas przywracania stanu nawet o 5 minut. Dlatego zachowaj rozsądek przy dodawaniu plików. Jeśli jest ich mniej niż 100, nie powinny wpływać na czas ponownego uruchamiania systemu. Pamiętaj jednak, aby to przetestować przed zastosowaniem strategii partycjonowania w środowisku produkcyjnym.

Przedziały danych używane do partycjonowania powinny być też dostosowane do strategii generowania plików. Można np. podzielić dużą tabelę z danymi klientów według dat złożenia zamówienia. Bardzo rzadko stosuje się strategię podziału na partycje bez uwzględniania dat. Gdy wybierasz przedział danych dla partycji, upewnij się, że używasz ważnej wartości zawsze wykorzystywanej w klauzuli `where` kwerend. Jeśli dana wartość nie jest używana w tej klauzuli, wystąpią problemy z wydajnością, ponieważ system nie będzie mógł pominąć części partycji i będzie musiał sprawdzać dane w całej tabeli.

UWAGA Do podziału danych na partycje nie można korzystać z następujących kolumn: o typach danych zdefiniowanych przez użytkownika, o typach `xml`, `varchar(max)`, `nvarchar(max)`, `varbinary(max)`, używających aliasów innych typów danych, a także zawierających znaczniki czasu i obrazy.

Wyobraź sobie system zgłaszania problemów w firmie telekomunikacyjnej. Gdy z powodu braku dostępu do usług generowane jest zgłoszenie, trafia ono do bazy danych. W tym momencie inicjowanych jest wiele działań. Pracownicy obsługi technicznej są przydzielani do problemu, następuje wymiana lub rekonfiguracja komponentów, a następnie usługę można ponownie udostępnić. Należy też sprawdzić umowę o poziomie świadczonych usług i — jeśli trzeba — wyescalować problem. Wszystkie te działania są spowodowane zgłoszeniem problemu. W tym systemie w tabeli aktywności i tabeli zgłoszeń znajdują się przeciążone punkty, co pokazano na rysunku 11.2.



Rysunek 11.2. System zgłaszania problemów

Na rysunku 11.2 informacje z obszaru *Bieżące* to nowe dane, które są istotne tylko w trakcie przestoju. Informacje z obszarów *Tylko do odczytu* i *Głównie do odczytu* są zwykle używane do prostych analiz po rozwiązaniu problemu, a później do generowania raportów. Ostatecznie dane przestają być aktualne i należy je przenieść do hurtowni danych. Niestety, z powodu wewnętrznych wymagań ta baza musi pozostać dostępna przez siedem lat.

Podział na partycje w opisywanym środowisku może przynieść istotne korzyści. W scenariuszu z *przesuwającym oknem* (patrz wyjaśnienie w punkcie „Tworzenie schematu partycjonowania”, dalej w tym rozdziale) każdego miesiąca (lub co kwartał) do środowiska dodawana jest nowa partycja przeznaczona na bieżące dane (z obszaru *Bieżące*) z tabel ze zgłoszeniami i działaniami. W ramach tego procesu należy usunąć partycję z danymi starszymi niż siedem lat.

Zgodnie z wcześniejszym opisem między zgłoszeniami a działaniami istnieje relacja jeden do wielu. Choć obie tabele będą miały zupełnie inną wielkość, dla obu trzeba zastosować identyczne procesy. Pozwoli to wykorzystać procesy oparte na zasobach współużytkowanych tylko przez te tabele. Aby ograniczyć koszty codziennych operacji, takich jak tworzenie kopii zapasowych i konserwacja indeksów dla wszystkich danych, tabele należy podzielić na podstawie dat. W listingu 11.1 (plik *KodR11.sql*) przedstawiono odpowiednią funkcję partycjonującą, która wykorzystuje kolumnę *ticketdate*. Jeśli chcesz wywołać tę instrukcję, usuń zbędne znaki podziału wiersza.

Listing 11.1. Funkcja partycjonująca

```
CREATE PARTITION FUNCTION
PFL_Years (datetime)
AS RANGE RIGHT
FOR VALUES (
'20050101 00:00:00.000', '20070101 00:00:00.000',
'20090101 00:00:00.000', '20110101 00:00:00.000',
'20120101 00:00:00.000')
```

UWAGA System SQL Server zaokrągla czas do 0,003 sekundy, dlatego wartość 0,997 jest zaokrąglana do 1 sekundy.

- Pierwsza partycja zawiera wszystkie wartości z datami wcześniejszymi niż '20050101 00:00:00.000'.
- Wartość graniczna '20050101 00:00:00.000' wyznacza początek drugiej partycji, obejmującej wszystkie wartości z datami większymi lub równymi '20050101 00:00:00.000' i mniejszymi niż '20070101 00:00:00.000'.
- Wartość graniczna '20070101 00:00:00.000' wyznacza początek trzeciej partycji, obejmującej wszystkie wartości z datami większymi lub równymi '20070101 00:00:00.000' i mniejszymi niż '20090101 00:00:00.000'.
- Wartość graniczna '20090101 00:00:00.000' wyznacza początek czwartej partycji, obejmującej wszystkie wartości z datami większymi lub równymi '20090101 00:00:00.000' i mniejszymi niż '20110101 00:00:00.000'.
- Wartość graniczna '20110101 00:00:00.000' wyznacza początek piątej partycji, obejmującej wszystkie wartości z datami większymi lub równymi '20110101 00:00:00.000' i mniejszymi niż '20120101 00:00:00.000'.
- Wartość graniczna '20120101 00:00:00.000' wyznacza początek szóstej partycji, obejmującej wszystkie wartości z datami większymi od '20120101 00:00:00.000'.

Funkcje partycjonujące tego rodzaju wyznaczają granice przedziałów. Słowa kluczowe `left` i `right` określają, do którego przedziału (lewego lub prawego) należą poszczególne granice przedziałów. Jeśli nie podasz żadnego z tych słów, domyślnie użyte zostanie słowo kluczowe `left`. W partycjach nie mogą występować luki. Musi istnieć możliwość przypisania wszystkich wartości. W przedstawionym kodzie wszystkie transakcje muszą należeć do jednego z przedziałów wyznaczanych przez przykładowe wartości.

Tworzenie grup plików

Na potrzeby strategii zdefiniowanej w funkcji partycjonującej należy utworzyć grupę plików. Zaleca się, aby obiekty użytkownika tworzyć i wiązać z grupami plików spoza głównej grupy plików. W głównej grupie plików należy przechowywać tylko obiekty systemowe. Można wtedy zapewnić dostęp do bazy danych, gdy wystąpią problemy wpływające na dostępność grup plików spoza głównej grupy.

UWAGA Jeśli chcesz kontynuować wykonywanie omawianego ćwiczenia, przed zbudowaniem schematu partycjonowania utwórz w bazie grupy plików CY04, CY06, CY08, CY10, CY11 i CY12.

Tworzenie schematu partycjonowania

Schemat partycjonowania wiąże obiekty bazy danych (np. tabele) z jednostką fizyczną (np. grupą plików) i docelowo z plikiem. Przy budowaniu takiego schematu należy zastanowić się nad tworzeniem kopii zapasowej, przywracaniem danych i ich archiwizowaniem. Omówienie tych kwestii zamieściliśmy w rozdziale 17., „Kopie zapasowe i przywracanie stanu”. W listingu 11.2 (plik *KodR11.sql*) pokazano, jak powiązać funkcje partycjonującą lub daty z poszczególnymi grupami plików. Ten schemat wymaga, aby dostępna była przedstawiona wcześniej funkcja partycjonująca `PFL_Years`.

Listing 11.2. Tworzenie schematu partycjonowania — sposób pierwszy

```
CREATE PARTITION SCHEME CYScheme
AS
PARTITION PFL_Years
TO ([CY04],[CY06],[CY08],[CY10],[CY11],[CY12])
```

To pozwala розміścić grupy plików w poszczególnych podsystemach z fizycznymi dyskami. W podobny sposób można też przenieść dawne dane do starszego, taniego podsystemu wejścia-wyjścia i przeznaczyć nowszy, szybszy podsystem na „gorące” dane (CY12). Po przeniesieniu dawnych danych do niedrogiego systemu wejścia-wyjścia grupy plików można oznaczyć jako przeznaczone tylko do odczytu. Po utworzeniu kopii zapasowej dawnych danych nie trzeba więcej powtarzać tego procesu. Ponadto system SQL Server automatycznie ignoruje grupy plików z dawnymi danymi w trakcie konserwacji indeksów.

W ramach tworzenia schematu partycjonowania z wykorzystaniem funkcji partycjonującej można też powiązać tę funkcję z jedną grupą plików. W listingu 11.3 (plik *KodR11.sql*) powiązano funkcję partycjonującą z domyślną grupą plików.

Listing 11.3. Tworzenie schematu partycjonowania — sposób drugi

```
CREATE PARTITION SCHEME CYScheme2
AS
PARTITION PFL_Years
TO ([Default],[Default],[Default],[Default],[Default],[Default])
```

OSTRZEŻENIE Jeśli nie utworzyłeś wcześniej funkcji partycjonującej PFL_Years, ten kod nie zadziała.

Partycjonowanie umożliwia łatwe usuwanie i wstawianie gigabajtów danych w podzielonych na partycje tabelach zajmujących ponad 500 gigabajtów. Wystarczy ustawić prostą opcję metadanych, a operacja będzie dotyczyć tylko około 125 map alokacji, co ograniczy zasięg transakcji z 500 gigabajtów do mniej niż 1 megabajta (pod warunkiem, że usuwanie lub wstawianie odbywa się na podstawie kolumny używanej do tworzenia partycji). Proces umożliwiający uzyskanie tego efektu to *przesuwane okno*. W tabeli niepodzielonej na partycje operacja zajęłaby wiele godzin z powodu eskalacji blokad i ponownej synchronizacji indeksów.

Aby do podzielonej na partycje źródłowej tabeli wstawić nową partycję z danymi będącą repozytorium na gorące dane, cały opisany proces można wykonać w odwrotną stronę. Jedyny wpływ takiej operacji na tabelę to krótka przerwa, gdy w trakcie przestawiania partycji tworzona jest blokada z poziomu schematu (może to potrwać kilka sekund). Takie przestawianie można stosować tylko do partycji przechowywanych w tej samej grupie plików. Szczegóły implementacji i najlepsze praktyki z tego obszaru omawiany dalej w tym rozdziale oraz w następnych rozdziałach.

Na współbieżność tabel podzielonych na partycje wpływa też ustawienie *Max Degree of Parallelism* (MAXDOP). Dla każdej partycji można zastosować odrębny wątek. Jeśli kwerenda dotyczy tylko jednej partycji, tworzonych jest tyle wątków, na ile pozwala ustawienie MAXDOP. Więcej o tym ustawieniu dowiesz się dalej w tym rozdziale.

Kompresja danych

Kompresja danych w wersji SQL Server 2014 pozostała w dużym stopniu bez zmian. Technologie wprowadzone we wcześniejszych wersjach nadal są niezawodne i przydatne. Gdy kompresja jest włączona, prowadzi do oszczędności miejsca nie tylko w systemie produkcyjnym, ale też w środowiskach: rozwojowym, testowym, kontroli jakości, zapewniania ciągłości działań i przywracania systemu po awarii.

W środowisku produkcyjnym kompresja pozwala zwiększyć wydajność podsystemu wejścia-wyjścia, ponieważ potrzebnych jest mniej odczytów (zwykłych i z wyprzedzeniem). Dane są kompresowane na dysku i w *pamięci*, dlatego oszczędność dotyczy nie tylko dysku. Pozwala to na wydajniejsze wykorzystanie zasobów puli buforów.

OSTRZEŻENIE Kompresja danych jest dostępna tylko w edycjach SQL Server 2014 Enterprise i Developer. Aby sprawdzić funkcje włączone w bazie danych, wykorzystaj widok `sys.dm_db_persisted_sku_features`.

Kompresję danych można stosować do następujących obiektów baz danych:

- tabel (z wyłączeniem tabel systemowych),
- indeksów klastrowanych,
- indeksów nieklastrowanych,
- widoków indeksowanych,
- podzielonych na partycje tabel i indeksów, przy czym każda partycja może mieć inne ustawienia kompresji.

Kompresja wierszy

Kompresja wierszy to podstawowy poziom kompresji stosowany w systemie SQL Server. Wpływa ona na dane na poziomie wierszy i całkowicie zmienia ich wewnętrzną strukturę. Aby w pełni zrozumieć ten mechanizm, przyjrzyj się rysunkowi 11.3. Widać na nim porównanie struktury danych rekordów nieskompresowanego i skompresowanego.

Zwykły rekord z danymi

Bajty metadanych	Przesunięcie mapy wartości NULL	Dane o stałej długości	Mapa wartości NULL	Tablica przesunięć danych o zmiennej długości	Dane o zmiennej długości
------------------	---------------------------------	------------------------	--------------------	---	--------------------------

Rekord ze skompresowanymi danymi

Nagłówek	Obszar deskryptorów kolumn	Obszar krótkich danych	Obszar długich danych
----------	----------------------------	------------------------	-----------------------

Rysunek 11.3. Struktura danych skompresowanych i nieskompresowanych

W nieskompresowanym rekordzie danych znajdują się *bajty metadanych* określające typ informacji i stan danych (rekordy mogą być np. rekordami typu *ghost* lub służyć do kontroli wersji, zawierać dane o zmiennej długości lub ich nie zawierać). Po bajtach metadanych znajduje się *przesunięcie mapy wartości NULL*. Ten blok został wprowadzony w wersji SQL Server 2005 i wskazuje mapę wartości NULL, co jest optymalizacją pozwalającą pominąć pola o tej wartości. Dalej znajdują się *dane o stałej długości*. Są to wszystkie dane, dla których zarezerwowana jest stała ilość pamięci. Jeśli np. używasz pola typu CHAR(50) i zapiszesz w nim słowo 'Test', zajmie ono 4 bajty, a 46 bajtów pozostanie puste. Jednak niezależnie od tego system SQL Server zajmuje wszystkie 50 bajtów.

Po obszarze danych o stałej długości znajduje się *mapa wartości NULL*, która obejmuje po 2 bajty na wiersz. Reprezentuje ona długość każdego wiersza różnego od NULL i określa, które wiersze zawierają wartości NULL. Dalej znajduje się *tablica przesunięć pól o zmiennej długości*. Zawiera ona wskaźniki do wszystkich *pól z danymi o zmiennej długości*. Tablica przesunięć i pola z danymi o zmiennej długości są wczytywane w pętli — na podstawie przesunięcia pobierane są dane, po czym pętla wczytuje następne przesunięcia.

Różnica między danymi o stałej i zmiennej długości jest prosta. Gdy utworzysz pole typu VARCHAR(50) i wstawisz w nim słowo 'Test', zajęte zostaną tylko 4 bajty. System SQL Server nie zapisuje wtedy dodatkowych 46 pustych znaków.

Na rysunku 11.3 pokazany jest też *skompresowany rekord z danymi*. Od razu widać, że ma on inną strukturę fizyczną. Dzieje się tak, ponieważ kompresja powoduje zwolnienie wolnej przestrzeni w danych o stałej długości.

Na początku znajduje się *nagłówek* skompresowanego rekordu. Nagłówek zawiera metadane opisujące skompresowany rekord (przypominają one bajty metadanych). Jednak w skompresowanym rekordzie nie występują dane o zmiennej długości, dlatego nie są potrzebne ich identyfikatory.

Dalej znajduje się obszar *deskryptorów kolumn*. Obszar ten zawiera jeden bit na wiersz (zamiast bajtów) określający, czy rekord jest zapisany w *obszarze krótkich danych*, w *obszarze długich danych*, czy jest przechowywany poza wierszem w nieskompresowanej postaci. Jedyne strony, na których można przechowywać nieskompresowane dane poza wierszem, to strony LOB_DATA i ROW_OVERFLOW_DATA.

Obszar krótkich danych zawiera wiersze mające nie więcej niż 8 bajtów. Obszar długich danych obejmuje wiersze o długości przekraczającej 8 bajtów, ale na tyle krótkich, aby zmieścić się na zwykłej stronie IN_ROW_DATA.

Kompresję wierszy da się bardzo łatwo włączyć lub wyłączyć. Można zastosować kompresję przy początkowym tworzeniu sterty lub indeksu, a także przy odtwarzaniu tych struktur. Uruchom poniższy kod, aby zobaczyć wyniki.

```
/*
Tworzenie przykładowej bazy danych i tabeli.
*/
USE MASTER
GO
IF EXISTS(SELECT NAME FROM SYS.DATABASES WHERE NAME='COMPRDEMOS')
BEGIN
    DROP DATABASE COMPRDEMOS
END
GO
CREATE DATABASE COMPRDEMOS
GO
USE COMPRDEMOS
GO
CREATE TABLE UNCOMPRESSED
(
    MYID INT IDENTITY(1,1)
    ,MYCHAR CHAR(500) DEFAULT 'A'
)
GO
/*
Wstawianie 5000 nieskompresowanych wierszy do tabeli.
*/
INSERT INTO UNCOMPRESSED
DEFAULT VALUES
GO 5000
/*
Sprawdzanie, ile miejsca zajmuje tabela
przed zastosowaniem kompresji i po skompresowaniu danych.
*/
SP_SPACEUSED 'UNCOMPRESSED'
GO
ALTER TABLE UNCOMPRESSED
REBUILD WITH(DATA_COMPRESSION=ROW)
GO
SP_SPACEUSED 'UNCOMPRESSED'
```

Wyniki pokazują, że kod tworzy tabelę o 5000 wierszy zajmującą 3144 kilobajty danych. Po zastosowaniu kompresji tabela zajmuje tylko 72 kilobajty.

Kompresja stron

W ramach *kompresji stron* przeprowadzana jest kompresja wierszy, po czym wykonywane są dwie inne operacje kompresji.

- **Kompresja przedrostkowa.** Dla każdej strony i kolumny identyfikowany jest przedrostek, który pozwala ograniczyć ilość zajmowanego miejsca. Wartość ta jest zapisywana na każdej stronie w strukturze z informacjami o kompresji. Następnie powtarzający się przedrostek jest zastępowany referencją do przedrostka z tej struktury.
- **Kompresja słownikowa.** Polega na wyszukiwaniu w dowolnym miejscu strony powtarzających się wartości, które są następnie zastępowane referencją do struktury z informacjami o kompresji.

Gdy kompresja stron jest stosowana dla nowej tabeli, dla wstawianych wierszy używana jest kompresja wierszy. Dzieje się tak do momentu zapelnienia strony, kiedy to stosowana jest kompresja strony. Jeśli potem na stronie znajduje się miejsce na nowe wiersze, są one wstawiane i następuje kompresja tej strony. Gdy na stronie zabraknie miejsca, nowe wiersze są wstawiane na następnej stronie. Przy stosowaniu kompresji stron dla indeksów na stronach, które nie są liśćmi, używana jest tylko kompresja wierszy. Aby utworzyć nową skompresowaną tabelę z kompresją stron, wywołaj następujące instrukcje:

```
USE AdventureWorks
GO
CREATE TABLE [Person].[AddressType_Compressed_Page](
    [AddressTypeID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [dbo].[Name] NOT NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
)
WITH (DATA_COMPRESSION=PAGE)
GO
```

Aby zmienić ustawienia kompresji dla tabeli, użyj polecenia ALTER TABLE:

```
USE AdventureWorks
GO
ALTER TABLE Person.AddressType_Compressed_Page REBUILD
WITH (DATA_COMPRESSION=PAGE)
GO
```

W tabelach lub indeksach podzielonych na partycje kompresję można stosować i modyfikować dla poszczególnych partycji. W poniższym kodzie pokazano, jak włączyć kompresję dla tabeli i indeksu z partycjami:

```
USE AdventureWorks
GO

CREATE PARTITION FUNCTION [TableCompression](Int)
AS RANGE RIGHT
FOR VALUES (1,10001,12001,16001);
GO

CREATE PARTITION SCHEME KeyRangePS
AS
```



```
PARTITION [TableCompression]
TO ([Default], [Default], [Default], [Default], [Default])
GO
```

```
CREATE TABLE PartitionTable
(KeyID int,
Description varchar(30))
ON KeyRangePS (KeyID)
WITH
(
DATA_COMPRESSION = ROW ON PARTITIONS (1),
DATA_COMPRESSION = PAGE ON PARTITIONS (2 TO 4)
)
GO
```

```
CREATE INDEX IX_PartTabKeyID
ON PartitionTable (KeyID)
WITH (DATA_COMPRESSION = ROW ON PARTITIONS(1),
DATA_COMPRESSION = PAGE ON PARTITIONS (2 TO 4 ) )
GO
```

Oto cechy operacji na skompresowanych partycjach tabel.

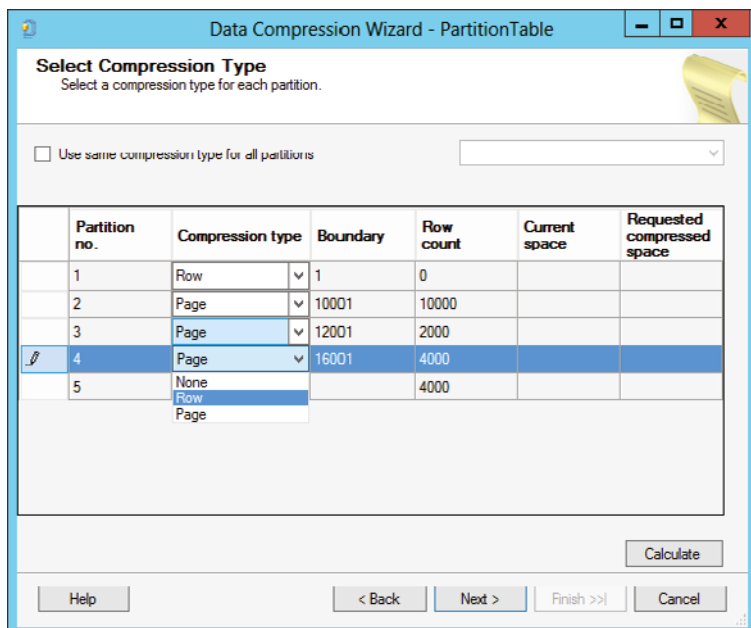
- **Dzielenie partycji.** Obie partycje przyjmują ustawienia pierwotnej partycji.
- **Scalanie partycji.** Wynikowa partycja przyjmuje ustawienia kompresji docelowej partycji.
- **Przestawianie partycji.** Ustawienia kompresji partycji i docelowej tabeli muszą do siebie pasować.
- **Usuwanie indeksu klastrowanego podzielonego na partycje.** Tabela zachowuje ustawienia kompresji.

Ponadto można zarządzać kompresją danych w oknie *Object Explorer* programu SQL Server Management Studio. Wymaga to wybrania tabeli lub indeksu do skompresowania. Aby np. skompresować tabelę, wykonaj następujące czynności.

1. Wybierz tabelę i kliknij ją prawym przyciskiem myszy. W menu podręcznym wybierz opcję *Storage/Manage Compression*.
2. W kreatorze kompresji danych, który się otworzy, kliknij przycisk *Next*. W oknie dialogowym *Select Compression Type* kreatora wybierz typ kompresji (*None*, *Row* lub *Page*) z listy rozwijanej *Compression Type*. To okno dialogowe przedstawiono na rysunku 11.4.
3. Po zmianie typu kompresji kliknij przycisk *Calculate*, aby zobaczyć oszacowaną oszczędność miejsca. Następnie kliknij przycisk *Next* i zakończ pracę kreatora.

Szacowanie oszczędności miejsca

Przed włączeniem kompresji danych możesz oszacować oszczędność miejsca, jaką da ta operacja. Jeśli np. wiersz zajmuje więcej niż 4 kilobajty, a dla danego typu danych precyzja zawsze odpowiada liczbom całkowitym, zyski będą niewielkie. Procedura składowana `sp_estimate_data_compression_savings` tworzy przykładowy zbiór danych w bazie tempdb i oblicza oszczędność miejsca, a następnie zwraca szacunkowe wartości dla tabeli z tymi danymi. Za pomocą tej procedury można wykonywać obliczenia dla tabel, indeksów klastrowanych, indeksów nieklastrowanych, widoków indeksowanych, partycji tabel i indeksów z uwzględnieniem



Rysunek 11.4. Ustawianie typu kompresji w oknie dialogowym Select Compression Type

kompresji stron lub wierszy. Za pomocą tej procedury można też oszacować wielkość, jaką tabela, indeks lub wiersz po kompresji będą zajmować w nieskompresowanym stanie. Omawiana procedura wykonuje te same obliczenia, które są przeprowadzane po kliknięciu przycisku *Calculate* w kreatorze kompresji danych przedstawionym na rysunku 11.4.

Poniżej podajemy składnię procedury składowanej `sp_estimate_data_compression_savings`:

```
sp_estimate_data_compression_savings
[ @schema_name = ] 'schema_name'
, [ @object_name = ] 'object_name'
, [ @index_id = ] index_id
, [ @partition_number = ] partition_number
, [ @data_compression = ] 'data_compression'
[;]
```

Oto opis elementów tego kodu.

- `@schema_name` to nazwa schematu zawierającego dany obiekt.
- `@object_name` to nazwa tabeli lub widoku indeksowanego, dla którego używany jest dany indeks.
- `@index_id` to identyfikator indeksu. Aby wybrać wszystkie indeksy tabeli lub widoku, podaj NULL.
- `@partition_number` to numer partycji z danym obiektem. Dla obiektów bez partycji podaj NULL lub 1.
- `@data_compression` to typ analizowanej kompresji. Dostępne wartości to NONE, ROW i PAGE.

Na rysunku 11.5 przedstawiono szacunkową oszczędność miejsca, jaką można uzyskać po kompresji tabeli `PartitionTable` z przykładowej bazy danych z pliku *KodR11.sql* dostępnego na stronie poświęconej tej książce.

	object_name	schema	index	partition	size_with_current_compression_setting(KB)	size_with_requested_compression_setting(KB)	sample_size_with_current_compression_setting(KB)	sample_size_with_requested_compression_setting(KB)
1	PartitionTable	dbo	0	1	0	0	0	0
2	PartitionTable	dbo	0	2	448	120	448	120
3	PartitionTable	dbo	0	2	152	128	152	128
4	PartitionTable	dbo	0	3	96	32	96	32
5	PartitionTable	dbo	2	3	48	40	48	40
6	PartitionTable	dbo	0	4	184	56	184	56
7	PartitionTable	dbo	2	4	72	64	72	64
8	PartitionTable	dbo	0	5	208	56	200	56
9	PartitionTable	dbo	2	5	88	64	88	64

Rysunek 11.5. Szacowana oszczędność miejsca, jaką można uzyskać przy użyciu kompresji

Oto informacje na temat wyników z rysunku 11.5.

- `index_id` to identyfikator obiektu. Tu 1 to indeks klastrowany (obejmujący tabelę), a 2 i 3 to indeksy nieklastrowane.
- `size_with_current_compression_setting` określa obecną wielkość obiektu.
- `size_with_requested_compression_setting` to wielkość obiektu po kompresji.

W tym przykładzie indeks klastrowany razem z tabelą zajmuje 9952 kilobajty, a po kompresji stron ta wielkość zmniejsza się do 4784 kilobajtów, co oznacza oszczędność 5168 kilobajtów.

OSTRZEŻENIE Zachowaj ostrożność przy korzystaniu z procedury `sp_estimate_data_compression_savings`. Pobiera ona 5% danych z analizowanej tabeli, przenosi je do bazy tempdb, kompresuje je, a następnie ekstrapoluje wynik na całą tabelę. To rozwiązanie sprawdza się dla tabel mających kilka megabajtów wielkości. Nie należy jednak stosować go dla tabel o wielkości terabajta zawierających 2 miliardy wierszy, chyba że wcześniej wykonałeś podobną operację w środowisku rozwojowym.

Monitorowanie kompresji danych

Do monitorowania kompresji danych na poziomie egzemplarza systemu SQL Server służą dwa liczniki dostępne w grupie `SQLServer:Access Methods` w monitorze wydajności systemu Windows.

- Licznik *Page compression attempts/sec* zwraca liczbę prób kompresji stron na sekundę.
- Licznik *Pages compress/sec* informuje o liczbie stron kompresowanych na sekundę.

Jednak te liczniki oferują bardzo ograniczone informacje. Wynika to z natury pracy monitora wydajności. Funkcja DMF `sys.dm_db_index_operational_stats` zapewnia dostęp do kolumn `page_compression_attempt_count` i `page_compression_success_count`, które pozwalają uzyskać statystyki dotyczące kompresji stron w poszczególnych partycjach. Warto zwrócić uwagę na uzyskane wartości, ponieważ nieudane próby kompresji oznaczają marnowanie zasobów systemowych. Jeśli stosunek prób do udanych operacji jest zbyt wysoki, negatywnie wpływa to na wydajność systemu, czego można uniknąć po wyłączeniu kompresji. Ponadto funkcja DNF `sys.dm_db_index_physical_stats` udostępnia kolumnę `compressed_page_count`, która wyświetla liczbę skompresowanych stron dla obiektu i dla partycji.

Aby zidentyfikować skompresowane obiekty w bazie, możesz wyświetlić zawartość kolumny `data_compression` (0 oznacza brak kompresji, 1 to kompresja wierszy, a 2 to kompresja stron) z widoku katalogu `sys.partitions`. W oknie *Object Explorer* programu SQL Server 2014 Management Studio wybierz tabelę i kliknij prawym przyciskiem myszy. Następnie w menu podręcznym wybierz opcję *Storage/Manage Compression*. Pojawi się kreator kompresji danych.

UWAGA Szczegółowe informacje o kompresji danych znajdziesz w artykule „Data Compression” w dokumentacji Books Online systemu SQL Server — [http://msdn.microsoft.com/en-us/library/cc280449\(v=SQL.120\).aspx](http://msdn.microsoft.com/en-us/library/cc280449(v=SQL.120).aspx).

Najlepszym sposobem monitorowania kompresji jest zastosowanie rozszerzonych zdarzeń w celu śledzenia, których stron dotyczyły nieudane próby kompresji i jakie były powody niepowodzeń. W tym celu wykorzystaj pakiet SQLSERVER oraz zdarzenia Page_Compression_Attempt_Failed i Page_Compression_Tracing.

Uwagi dotyczące kompresji danych

Gdy decydujesz, czy warto zastosować kompresję danych, pamiętaj o następujących kwestiach.

- Kompresja danych jest dostępna tylko w edycjach SQL Server 2014 Enterprise i Developer.
- Włączenie i wyłączenie kompresji tabeli lub indeksu klastrowanego może prowadzić do odtworzenia wszystkich indeksów nieklastrowanych.
- Kompresji danych nie można stosować do kolumn z atrybutem SPARSE.
- Dane typów LOB przechowywane poza wierszami nie są kompresowane.
- Dla stron indeksów niebędących liściami stosowana jest tylko kompresja wierszy.
- Indeksy nieklastrowane nie przejmują ustawień kompresji tabeli.
- Przy usuwaniu indeksu klastrowanego tabela zachowuje ustawienia kompresji.
- Jeśli nie jest określone inaczej, tworzony indeks klastrowany przejmuje ustawienia kompresji tabeli.

Gdy chcesz ustalić, które obiekty skompresować, wykonaj następujące kroki.

- Poszukaj największych tabel w bazie danych.
- Sprawdź jednostki alokacji używane w tych tabelach. Pamiętaj, że strony IN_ROW_DATA umożliwiają kompresję, natomiast dla stron ROW_OVERFLOW_DATA i LOB_DATA nie jest ona obsługiwana.
- Zwróć uwagę na wzorce użytkowania tabel. Użyj do tego funkcji `sys.dm_db_index_operational_stats`. Sprawdź liczbę operacji SEEK i SCAN. Duża liczba aktualizacji oznacza, że warto rozważyć rezygnację z kompresji danego obiektu zwłaszcza wtedy, gdy moc obliczeniowa procesorów jest ograniczona.
- Testuj i monitoruj plany wykonywania kwerend oraz ich wydajność przed wprowadzeniem kompresji i po jej zastosowaniu. W ten sposób sprawdzisz, czy kompresja pomaga zwiększyć wydajność systemu.

OSTRZEŻENIE Gdy stosujesz kompresję, wielkość rekordów dziennika transakcji może się zmniejszyć (zależy to od poziomu kompresji danych). Niezależnie od ustawień kompresji każda zarejestrowana operacja wstawiania, aktualizowania lub usuwania danych powoduje zapis poprzedniej i nowej kopii danych. Dane skompresowane mogą zajmować mniej miejsca niż w postaci nieskompresowanej. Próby kompresji stron są rejestrowane poza procesami użytkownika, dlatego nie zwiększają kosztów wykonywania transakcji użytkownika.

System SQL Server i procesory

System SQL Server 2014 (i jego starsze wersje od SQL Server 2005) używa specjalnego algorytmu do szeregowania procesów użytkowników za pomocą systemu SQLOS. SQLOS wykorzystuje dla procesów użytkownika jeden program szeregujący na jeden rdzeń logiczny. W systemie SQL Server stosowane są dwa rodzaje programów szeregujących.

- *Ukryte programy szeregujące*, używane przez wewnętrzne procesy systemu SQL Server. Obecnie dla każdego egzemplarza używany jest jeden taki program, a dodatkowe cztery programy szeregujące służą do monitorowania grup dostępności AlwaysOn, połączeń DAC i technologii In-Memory OLTP.
- *Widoczne programy szeregujące*, dostępne dla kwerend i procesów użytkowników. SQLOS to wewnętrzny system operacyjny wbudowany w system SQL Server. Odpowiada za przydzielanie czasu procesora, szeregowanie wątków, wykonywanie zadań i rozdzielanie pamięci.

Może wydawać się dziwne, że system SQL Server ma wbudowany własny system operacyjny. SQLOS zbudowano dlatego, że Windows nie jest dobrym systemem operacyjnym dla baz danych. Windows stosuje *szeregowanie z wykluczaniem*, natomiast SQLOS *szeregowanie bez wykluczania*. Aby zrozumieć te dwa modele, warto przyrzeć się krótkiemu przykładowi ilustrującemu występujące między nimi różnice.

Załóżmy, że Ty wraz z innym dorosłym znajdujecie się w pomieszczeniu z 15 dziećmi. Drugi dorosły daje dzieciom ciastka, cukierki, oranżadę i cukier. Następnie szepcze coś do dzieci, a te stają się bardzo podekscytowane. Po chwili drugi dorosły wskazuje na Ciebie i mówi do dzieci: „Idźcie, przekazcie mu to, co usłyszeliście”. Początkowo jesteś przytłoczony! Wokół Ciebie podskakuje 15 maluchów, które krzyczą jedno przez drugie, a Twoim zadaniem jest je wysłuchać. Możesz chwycić jedno i powiedzieć: „Mogę słuchać Cię przez sekundę”, następnie złapać inne i stwierdzić to samo, a następnie powtórzyć to z kolejnymi dziećmi. W takim scenariuszu możesz starać się usłyszeć jak najwięcej przez krótki czas, ale nie pozwalasz żadnemu dziecku skończyć, ponieważ gorączkowo przechodzisz od jednego malucha do drugiego.

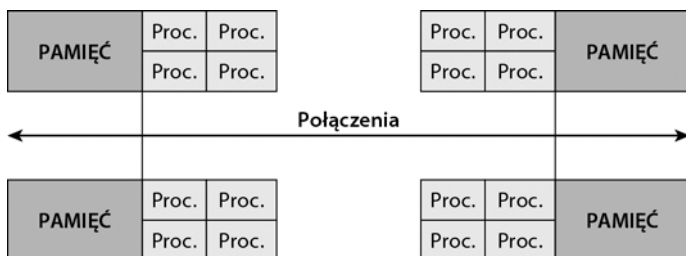
Tak wygląda szeregowanie z wykluczaniem. W systemie operacyjnym wygląda to tak: założmy, że na komputerze działają Internet Explorer z dziesięcioma otwartymi zakładkami, egzemplarz systemu SQL Server 2014, maszyna wirtualna z systemem SQL Server 2014, Excel, Word, Notatnik, Outlook, Lync, zdalny pulpit, Skype i PowerPoint. Procesor przeskakuje od jednej aplikacji do drugiej i komunikuje: „Ciebie mogę słuchać przez sekundę, teraz ty, a teraz ty...”. W przypadku kwerend ten model się nie sprawdza. System operacyjny nie powinien odbierać im zasobów, jeśli nie skończyły pracy.

Teraz założmy, że cała 15 dzieci jest grzeczna, posłuszna i stoi spokojnie w kolejce. Każde z nich czeka, aby po kolei z Tobą porozmawiać. Jeśli pierwsze dziecko musi się zastanowić (pobrać jakieś dane ze swojego małego dysku twardego), wychodzi z kolejki i przepuszcza następną osobę. Dzieci są tu jak wątki robocze. Gdy któryś z wątków czeka, w kolejce oczekiwania generowane są statystyki. Stan takiego wątku to `SUSPENDED`. Gdy dziecko przypomni sobie potrzebne informacje, może wrócić do kolejki. W świecie komputerów jest to kolejka wątków gotowych do uruchomienia, a stan takiego wątku zmienia się w `RUNNABLE`. Gdy wszystkie dzieci już z Tobą porozmawiają, pierwsze dziecko też może to zrobić. Stan analogicznego wątku zmienia się wtedy na `RUNNING` i zadanie zostaje wykonane.

W tym przykładzie pełnisz funkcję programu szeregującego. Słuchasz historyjek, czego odpowiednikiem jest wykonywanie zadań przez kwerendę. Pamiętaj, że na jeden rdzeń logiczny serwera przypada jeden program szeregujący, co pozwala wykonywać jednocześnie wiele procesów.

Architektura NUMA i dynamicznie dodawane procesory

Od lat standardową architekturą sprzętu jest architektura NUMA (ang. *Non-Uniform Memory Access*). Na szczęście system SQL Server obsługuje tę architekturę już od wersji SQL Server 2005. W ramach przypomnienia, jak działają procesory i pamięć w architekturze NUMA, przyjrzyj się rysunkowi 11.6.



Rysunek 11.6. Procesory i pamięć w architekturze NUMA

W systemach Windows 2012 i SQL Server 2014 możliwe jest *dynamiczne dodawanie procesorów*. Polega to na dynamicznym dodawaniu procesorów w czasie działania systemu SQL Server. Oto wymagania, które system musi spełniać, aby możliwe było zastosowanie tej technologii.

- Sprzętowa obsługa dynamicznego dodawania procesorów.
- Zainstalowany 64-bitowy system operacyjny Windows Server 2012.
- Zainstalowany system SQL Server 2014 Enterprise.
- W systemie SQL Server nie może być skonfigurowana programowa architektura NUMA (ang. *soft NUMA*).

UWAGA Aby znaleźć więcej informacji o programowej architekturze NUMA, wpisz w wyszukiwarce wyrażenia *Understanding Non-Uniform Memory Access* i *How to Configure SQL Server to Use Soft-NUMA*.

Jeśli te wymagania są spełnione, możesz wywołać polecenie RECONFIGURE, żeby system SQL Server 2014 wykrył dynamicznie dodany procesor.

Spójność pamięci podręcznej

Aby zachowana została integralność danych, określone informacje w danym momencie może aktualizować tylko jeden procesor. Kopie tych danych przechowywane w pamięci podręcznej innych procesorów zostają unieważnione, dlatego należy ponownie wczytać te informacje. Odpowiada za to mechanizm zapewniania *spójności danych*. Dbą on o to, aby wszystkie pamięci podręczne miały te same informacje na temat lokalizacji wszystkich kopii danych i by określone było, który procesor ma obecnie prawo do aktualizowania tych danych.

Szeregowanie zadań w systemie SQL Server za pomocą maski koligacji

Przy użyciu opcji maski koligacji można przypisać podzbiór dostępnych procesorów do procesu systemu SQL Server. Za szeregowanie wątków roboczych systemu SQL Server odpowiada program szeregujący.

Gdy czas przydzielony przez program szeregujący upłynie, wątek dobrowolnie kończy pracę. Wtedy program szeregujący pozwala na rozpoczęcie pracy innemu wątkowi roboczemu. Jeśli dany wątek nie może kontynuować działania z powodu braku dostępu do zasobów (np. dyskowych operacji wejścia-wyjścia), zostaje uśpiony do momentu, gdy potrzebne zasoby będą dostępne. Kiedy jądro systemu Windows przekazuje kontrolę nad procesorem z wykonywanego procesu do innego, który jest gotowy do uruchomienia, następuje *przełączenie kontekstu*.

Przełączanie kontekstu

Przełączanie kontekstu to kosztowna operacja, ponieważ zmiana jednego działającego wątku na inny wymaga wykonania różnych operacji porządkujących. *Operacje porządkujące* związane są z zarządzaniem kontekstem, czyli wartościami rejestrów procesora i innymi danymi określającymi stan procesu. Jądro systemu Windows wczytuje kontekst nowego procesu, po czym ten proces zaczyna pracę. Przy następnym uruchomieniu procesu wznawia on pracę od miejsca, w którym odebrano mu procesor. Jest to możliwe, ponieważ zapisywany kontekst obejmuje wskaźnik instrukcji. Przełączanie kontekstu w systemie operacyjnym Windows może występować zarówno w *trybie użytkownika*, jak i w *trybie uprzywilejowanym*.

Łączny czas pracy procesora to suma czasu pracy w trybie uprzywilejowanym i w trybie użytkownika.

Tryb uprzywilejowany

Tryb uprzywilejowany to tryb przetwarzania zaprojektowany na potrzeby komponentów systemu operacyjnego i sterowników sprzętu. Umożliwia on dostęp do sprzętu i całej pamięci. Czas pracy w trybie uprzywilejowanym obejmuje czas obsługi przerw i opóźnionych wywołań procedur (ang. *Deferred Procedure Call* — DPC).

Pula lekkich wątków w systemie SQL Server

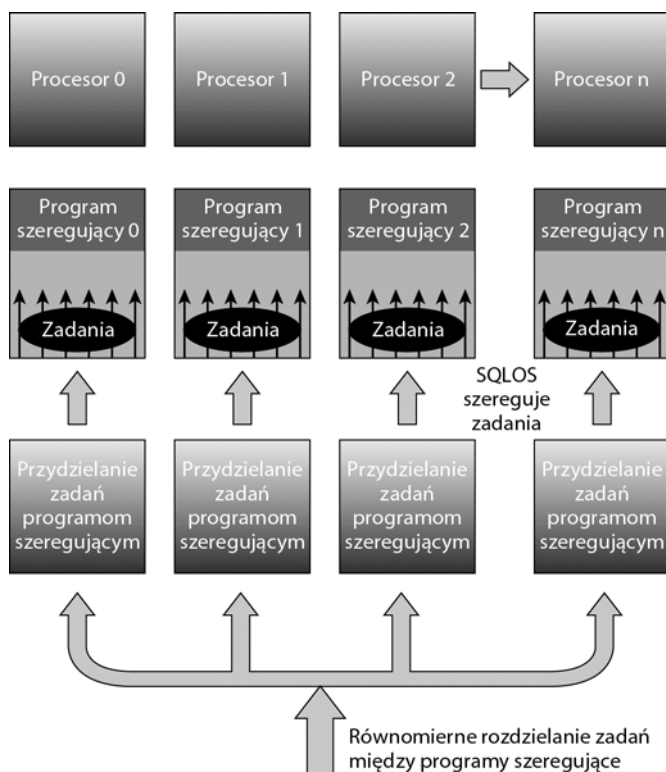
Przełączanie kontekstu często prowadzi do problemów. W większości środowisk przełączanie kontekstu zachodzi rzadziej niż 1000 razy na sekundę na procesor. Pula lekkich wątków systemu SQL Server pozwala ograniczyć związane z tym koszty, ponieważ umożliwia wykorzystanie w zadaniach jednostek roboczych w postaci włókien z systemów NT zamiast wątków.

Włókno to jednostka wykonawcza prostsza od wątku. Działa ona w trybie użytkownika. Gdy włączona jest pula lekkich wątków, każdy program szeregujący używa jednego wątku do kontrolowania szeregowania operacji żądanych przez wiele włókien. Włókno można traktować jak lekki wątek, który w określonych warunkach powoduje mniejsze koszty niż standardowe wątki robocze wymagające przełączania kontekstu. Liczbę włókien można ustawić za pomocą opcji konfiguracyjnej *Max Worker Threads*.

OSTRZEŻENIE W 99,99% sytuacji nie powinieneś zmieniać ustawienia *Max Worker Threads*. Zmianę tej opcji warto rozważyć tylko wtedy, gdy statystyki oczekiwania informują o oczekiwaniu na procesor i nie występują inne przyczyny oczekiwania, a liczniki monitora wydajności to potwierdzają.

Maska koligacji

Opcje maski koligacji pozwalają ustawić dla egzemplarza systemu SQL Server podzbiór procesorów. Gdy system SQL Server 2014 działa na dedykowanym serwerze, udostępnienie systemowi wszystkich procesorów pozwala uzyskać najwyższą wydajność. Aby zapewnić przewidywalną wydajność przy konsolidacji serwerów lub w środowisku z wieloma egzemplarzami, możesz powiązać procesory z konkretnymi egzemplarzami systemu SQL Server, co zmniejsza ryzyko współzawodnicstwa egzemplarzy o zasoby (patrz rysunek 11.7).



Rysunek 11.7. Przebieg szeregowania

Przychodzące połączenia są przypisywane do węzła procesora. System SQL Server przypisuje żądania wsadowe do zadań, którymi zarządzają programy szeregujące. W danym momencie program szeregujący może przeznaczyć do uruchomienia w procesorze tylko jedno zadanie. *Zadanie* to jednostka pracy szeregowana do wykonania w systemie SQL Server. Taka architektura gwarantuje równoważenie obsługi setek (a często nawet tysięcy) połączeń, które mogą występować w dużych systemach.

Ustawienie Max Degree of Parallelism (MAXDOP)

Ustawienie MAXDOP domyślnie ma wartość 0. Umożliwia to systemowi SQL Server uwzględnianie wszystkich procesorów przy tworzeniu planów wykonywania operacji. W większości systemów zaleca się ustawienie tej opcji na liczbę rdzeni dostępnych w jednym węźle architektury NUMA. Pozwala to ograniczyć koszty związane z równoległością.

Zalecenie ustawiania opcji MAXDOP na 1 zwykle ma służyć niwelowaniu problemów z niskiej jakości kodem, nieobsługującym przetwarzania równoległego. Na forach poświęconych systemowi SQL Server zetknęliśmy się z taką wskazówką w kontekście narzekania użytkowników na statystyki oczekiwania CXPACKET. Niektórzy radzili właśnie ustawienie opcji MAXDOP na 1. Jest to zupełnie bezsensowny pomysł. Oczekiwanie CXPACKET związane jest z wymianą pakietów (informacji) między procesorami przy równoległym wykonywaniu operacji. Jeśli system SQL Server jest właściwie dostrojony, oczekiwanie CXPACKET jest jedynie informacją o przebiegu operacji.

Istnieją jednak systemy o tak dużej liczbie transakcji na sekundę, że równoległe przetwarzanie kwerend negatywnie wpływa na pracę aplikacji (można to ustalić na podstawie profilu przebiegu operacji). W takiej sytuacji rzeczywiście warto ustawić opcję MAXDOP na 1. Optymalizator kwerend nie będzie wtedy wybierał planów równoległego wykonywania kwerend. W środowisku OLTP używanie wielu procesorów do wykonywania jednej kwerendy nie zawsze jest właściwym podejściem.

UWAGA Wszelkie zmiany ustawienia MAXDOP na wartość inną niż 0 należy przetestować przed wprowadzeniem ich w środowisku produkcyjnym. Sprawdź modyfikacje w środowisku rozwojowym lub używanym do kontroli jakości, uruchom na nim operacje zarejestrowane na serwerze produkcyjnym i upewnij się, że zmiany nie prowadzą do spadku wydajności przetwarzania kwerend.

Inne ustawienie, które warto przeanalizować, to *Cost Threshold for Parallelism*.

Ustawienie Cost Threshold for Parallelism

Ustawienie *Cost Threshold for Parallelism* (czyli koszt progowy dla współbieżności) wprowadzono w wersji SQL Server 7.0. Wartość ta wynosiła 5 i nie zmieniała się aż do wersji SQL Server 2014. Wyznacza ona koszt progowy dla współbieżności.

Gdy system SQL Server sprawdza ogólny koszt sekwencyjnego planu wykonywania kwerendy, optymalizator może też wygenerować plan współbieżny, aby porównać, który z planów jest mniej kosztowny, a tym samym łatwiejszy do wykonania. Jeśli koszt planu sekwencyjnego jest wyższy niż wartość ustawienia *Cost Threshold for Parallelism*, należy wygenerować plan współbieżny i porównać koszty obu planów.

W czasie, gdy pojawił się system SQL Server 7.0, wartość 5 uzyskano na komputerze jednego z członków zespołu odpowiedzialnego za optymalizator z tego systemu. W ciągu następnych siedemnastu lat komputery stały się znacznie wydajniejsze. Plan wykonywania kwerend, który w 1997 roku warto było stosować w wersji współbieżnej, w sprzeczności z roku 2014 może działać wydajniej w wersji sekwencyjnej. Pamiętaj jednak, że przed wprowadzeniem zmian w udostępnianym systemie produkcyjnym zawsze należy przeprowadzić dokładne testy. Nigdy nie modyfikuj ustawień z poziomu serwera w aktywnym systemie, jeśli nie masz pewności, że nie wpłynie to negatywnie na wydajność.

W ramach testów omawianego ustawienia w aktywnym systemie najpierw należy określić koszt planów współbieżnych. Za pomocą poniższej kwerendy znajdziesz na serwerze plany współbieżne i poznasz ich koszt. Kwerendę należy uruchamiać w czasie poświęconym na konserwację systemu. Skanowanie pamięci podręcznej z planami i pobieranie danych w formacie XML, w momencie gdy system pracuje pod dużym obciążeniem, może pogorszyć jego wydajność.

```

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

WITH XMLNAMESPACES
    (DEFAULT 'http://schemas.microsoft.com/sqlserver/2004/07/showplan')
SELECT
    query_plan AS CompleteQueryPlan
    ,i.value('@StatementText')[1] AS StatementText
    ,i.value('@StatementOptmLevel')[1] AS OptimizationLevel
    ,i.value('@StatementSubTreeCost')[1] AS QueryCost
    ,i.query('.') AS ExecutionPlan
    ,dec.p.usecounts AS TimesExecuted
FROM sys.dm_exec_cached_plans AS decp
CROSS APPLY sys.dm_exec_query_plan(plan_handle) AS deqp
CROSS APPLY query_plan.nodes('/ShowPlanXML/BatchSequence/Batch/Statements/
    StmtSimple') AS qpn(i)
WHERE i.query('.').exist('//*[Re10p[@PhysicalOp="Parallelism"]']') = 1

```

Na rysunku 11.8 przedstawiono wyniki tej kwerendy. Widać tu, że dwie znalezione procedury składowane mają koszt powyżej 5, ale poniżej 15 punktów.

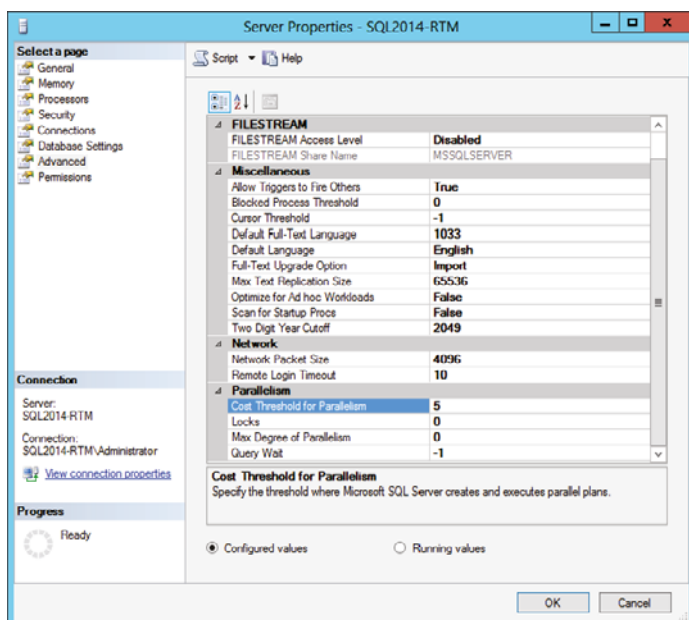


CompleteQueryPlan	StatementText	OptimizationLevel	QueryCost	ExecutionPlan	TimesExecuted
1	CREATE PROCEDURE p_calc_revenue_for_each_produ...	FULL	6.32736	q1 StmtSimple xmlns:p1="http://schemas.microso...	342
2	CREATE PROCEDURE p_set_discount_sales AS BEGIN ...	FULL	6.33778	q1 StmtSimple xmlns:p1="http://schemas.microso...	342

Rysunek 11.8. Koszt wykonania dla planów współbieżnych

Następny krok polega na przetestowaniu tych procedur składowanych ze wskazówką powodującą ustawienie opcji MAXDOP na 1, co wymusi zastosowanie sekwencyjnego planu wykonania. Porównaj czasy wykonania dla obu podejść, aby stwierdzić, która wersja kwerendy jest wydajniejsza (patrz rysunek 11.8).

W nowych systemach przeważnie można bezpiecznie zmienić omawiane ustawienie na wartość z przedziału 15 – 20. Ekran z ustawieniem *Cost Threshold for Parallelism* jest widoczny na rysunku 11.9.



Rysunek 11.9. Zmianie ustawienia Cost Threshold for Parallelism

Uwagi na temat pamięci i związanych z nią usprawnień

Jest kilka kwestii i usprawnień związanych z pamięcią w systemie SQL Server 2014, o których warto pamiętać.

Rozszerzenia puli buforów

System SQL Server 2014 umożliwia skalowanie pamięci w sposób, który wcześniej dotyczył tylko procesorów. Procesory udostępniają pamięć podręczną na trzech poziomach (L1, L2 i L3). Pamięć z poziomu L1 jest najszybsza, a pamięć z każdego następnego poziomu jest o kilka nanosekund wolniejsza. Jednak korzystanie z każdego z tych poziomów jest szybsze niż używanie zwykłej pamięci, która wymaga dodatkowych kilkuset milisekund.

W wersjach starszych niż SQL Server 2014 po wyczerpaniu się pamięci systemowej następowało stronicowanie i przenoszenie danych na dysk fizyczny, co prowadziło do spadku wydajności o rzędy wielkości (z nanosekund do milisekund). Jednak obecnie dzięki rozszerzeniom puli buforów (ang. *Buffer Pool Extension* — BPE) można rozszerzyć taką pulę. Główna pamięć systemowa pełni wtedy funkcję puli buforów poziomu pierwszego (L1), natomiast BPE to pula buforów poziomu drugiego (L2).

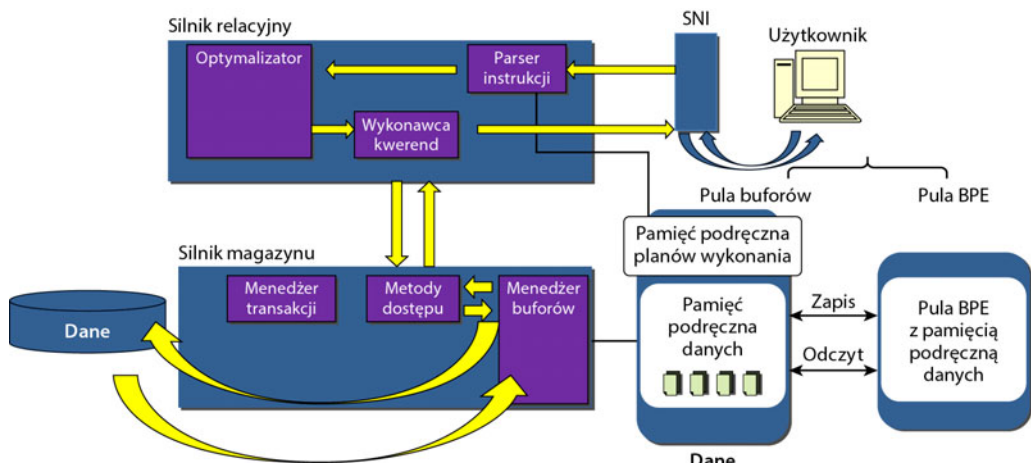
Za pomocą poniższego skryptu możesz utworzyć pulę BPE i umieścić ją na fizycznym dysku. Pokazana poniżej kwerenda pozwala pobrać dane zapisane przez bazę w puli buforów i sprawdzić, czy określone dane znajdują się w zwykłej puli buforów, czy w puli BPE.

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'F:\Example.BPE', SIZE = 20 GB);

SELECT
    DB_NAME(database_id) AS [Database Name]
    , case is_in_bpool_extension
        when 1 then 'BPool Extension'
        when 0 then 'Reg Buffer Pool'
        end as is_in_bpool_extension
    , COUNT(*) * 8/1024.0 AS [Cached Size (MB)]
FROM
    sys.dm_os_buffer_descriptors
WHERE
    database_id > 4 -- Systemowe bazy danych
    AND database_id <> 32767 -- Baza Resource
GROUP BY DB_NAME(database_id), is_in_bpool_extension
ORDER BY [Cached Size (MB)] DESC OPTION (RECOMPILE);
```

Za pomocą puli BPE powstaje wielopoziomowa hierarchia pamięci podręcznej, w której zatwierdzone strony są zapisywane w puli buforów L2 (czyli w puli BPE). Strony niezatwierdzone (czyli strony danych zmodyfikowane przez operacje wstawiania, aktualizowania lub usuwania) pozostają zapisywane w puli buforów L1. Na rysunku 11.10 przedstawiono przykładowy przepływ danych w systemie SQL Server z pulą BPE.

W systemie SQL Server 2014 wprowadzono dodatkowe widoki DMV, kolumny i zdarzenia rozszerzone, aby ułatwić zrozumienie działania pul BPE i tego, jak system z nich korzysta. W tabeli 11.2 przedstawiamy listę widoków DMV i zdarzeń rozszerzonych.



Rysunek 11.10. Przepływ danych w systemie z pulą BPE

Tabela 11.2. Widoki DMV i zdarzenia rozszerzone związane ze śledzeniem i monitorowaniem pul BPE

Nazwa obiektu	Typ obiektu	Opis
sys.dm_os_buffer_pool_extension_configuration	Widok DMV	Jest to kwerenda zwracająca informacje o konfiguracji pul BPE w systemie SQL Server. Kwerenda zwraca ścieżkę do fizycznego pliku, identyfikator pliku, stan, opis stanu i aktualną wielkość puli w kilobajtach.
sys.dm_os_buffer_descriptors	Widok DMV	Ten widok zmodyfikowano w wersji SQL Server 2014; dodano do niego kolumnę <code>is_in_pbool_extension</code> . Za pomocą tej kolumny można ustalić, czy bufor pamięci podręcznej danych znajduje się w pamięci fizycznej, czy w puli BPE.
sqlserver.buffer_pool_extension_pages_written	Zdarzenie rozszerzone	Zdarzenie jest zgłaszane, gdy strona lub grupa przyległych stron zostają przeniesione do puli BPE.
sqlserver.buffer_pool_extension_pages_read	Zdarzenie rozszerzone	Zdarzenie jest zgłaszane w momencie wczytania strony z puli BPE.
sqlserver.buffer_pool_extension_pages_evicted	Zdarzenie rozszerzone	Zdarzenie jest zgłaszane w momencie usunięcia strony z puli BPE.
sqlserver.buffer_pool_eviction_thresholds_recalculate	Zdarzenie rozszerzone	Zdarzenie jest zgłaszane w momencie ponownego obliczenia progu usuwania stron.

OSTRZEŻENIE System SQL Server nie wykrywa różnic między dyskami SSD a zwykłymi magnetycznymi dyskami twardymi. Pule BPE należy umieszczać na dyskach SSD. Jeśli utworzysz taką pulę na magnetycznym dysku twardym, nie uzyskasz wzrostu wydajności. Może nawet nastąpić jej spadek na poziomie podobnym, a nawet większym jak przy stronicowaniu i przenoszeniu danych na fizyczny dysk twardy.

Dostrajanie pamięci systemu SQL Server

Systemowy monitor wydajności udostępnia wiele liczników.

Licznik *SQLServer Cache Hit Ratio* informuje o tym, w jakim stopniu żądania użytkownika są obsługiwane za pomocą danych z pamięci podręcznej, a w jakim stopniu konieczne jest żądanie danych od podsystemu wejścia-wyjścia. Dostęp do danych z pamięci RAM (lub z pamięci podręcznej danych) jest znacznie szybszy niż pobieranie tych samych informacji z podsystemu wejścia-wyjścia. Dlatego korzystne jest wczytywanie wszystkich aktywnych informacji z pamięci RAM. Niestety, pamięć RAM ma ograniczoną pojemność. Dlatego zadowalającą wartością wspomnianego licznika jest wartość wyraźnie przekraczająca 90%. Nie oznacza to, że środowisko danego egzemplarza systemu SQL Server nie wymaga dodatkowej pamięci. Jednak niższa wartość informuje, że ilość pamięci systemowej lub danych zapisywanych w pamięci podręcznej jest poniżej pożądanego poziomu.

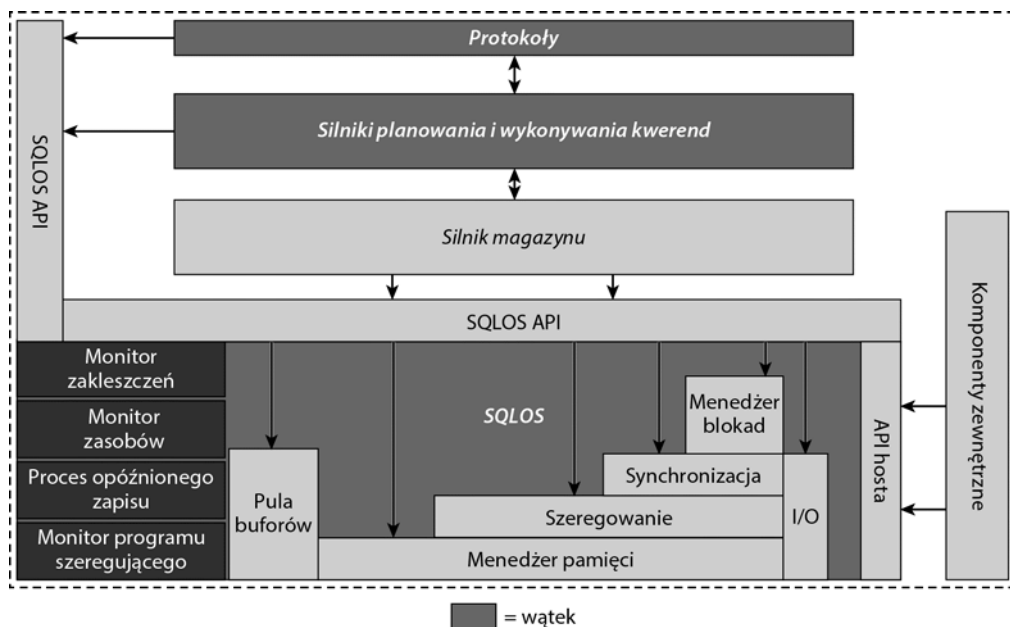
Innym dobrym urządzeniem do pomiaru dużego obciążenia pamięci jest licznik *SQLServer:Buffer manager: Page-life-expectancy* (PLE). Licznik informuje, przez jaki czas strona bufora pozostaje w pamięci (w sekundach). Pożądana wartość tego licznika zależy od ilości pamięci RAM zainstalowanej na danym serwerze, od ilości pamięci przeznaczonej na pamięć podręczną planów wykonania, od używanego systemu operacyjnego itd. Obecnie przyjmuje się, że w celu obliczenia idealnej wartości PLE należy zastosować następujący wzór: *Maximum server memory* (w gigabajtach) \times 75. Tak więc w systemie ze 128 gigabajtami pamięci RAM i ustawieniem *Maximum server memory* o wartości 120 gigabajtów minimalna wartość PLE, której przekroczenie może oznaczać problemy, to 9000. Gdy chcesz monitorować licznik PLE, jest to bardziej realistyczna wartość niż zalecany w przeszłości poziom 300.

Uważaj, aby przydzielić wystarczającą ilość pamięci systemowej; jeśli tego nie zrobisz, system operacyjny będzie musiał zacząć przenosić dane na dysk fizyczny, co spowoduje błędy strony. *Błąd strony* występuje, gdy system operacyjny musi szukać danych nie w pamięci, ale na dysku fizycznym. W systemie operacyjnym może występować stronicowanie, gdy jednak jest ono potrzebne zbyt często, wymaga wielu dyskowych operacji wejścia-wyjścia i dużo czasu procesora, co wprowadza opóźnienia w pracy całego serwera i powoduje spadek wydajności bazy danych. Aby wykryć brak wystarczającej ilości pamięci systemowej, sprawdź wartość licznika *Pamięć: Strony/s*. Jego wartość powinna być jak najbliższa 0, ponieważ większe liczby wskazują na intensywne stronicowanie (może to mieć miejsce np. w trakcie tworzenia kopii zapasowych).

W systemie 2014 dostępnych jest kilka mechanizmów pomocnych przy rozwiązywaniu problemów z pamięcią. W systemie operacyjnym Windows Server 2012 system SQL Server 2014 może wykorzystać dynamicznie dodawaną pamięć, mechanizmy zarządzania serwerem i inne usprawnienia.

Wcześniej w tym rozdziale wspomniano, że SQLOS to prosta warstwa między systemem SQL Server a systemem Windows zarządzająca interakcją między tymi środowiskami. Za jej pomocą system SQL Server może dostosować się do dowolnego sprzętu. Na rysunku 11.11 przedstawiono komponenty systemu SQLOS, które odpowiadają za szeregowanie i synchronizowanie wątków, zarządzają pamięcią systemu SQL Server, zapewniają obsługę wyjątków i obsługują komponenty CLR.

Celem tego systemu jest umożliwienie systemowi SQL Server wykorzystania wszystkich innowacji sprzętowych dostępnych w architekturze x64. System SQLOS zbudowano z myślą o obsłudze danych lokalnych i dynamicznego modyfikowania konfiguracji oraz wykorzystaniu możliwości sprzętu. SQLOS pozwala też na lepszą obsługę architektury CC-NUMA (ang. *Cache Coherent Non-Uniform Memory Access*) w systemie SQL Server 2014.



Rysunek 11.11. Elementy systemu SQLOS

W architekturze systemu SQLOS używane są *węzły pamięci*. Tworzą one poziom między pamięcią a procesorem. Dla każdej grupy procesorów istnieje węzeł pamięci.

Węzeł procesorów to także struktura w hierarchii tego systemu. Zaprojektowano ją na potrzeby tworzenia logicznych grup procesorów. Pozwala to na kierowanie operacji do węzła procesorów i równoważenie obciążenia w jego ramach.

Miedzy węzłami procesorów a węzłem pamięci istnieje prosta relacja — z węzłem pamięci można powiązać wiele węzłów procesorów, ale dla danego węzła procesorów można używać tylko jednego węzła pamięci. Każdy poziom w omawianej hierarchii udostępnia usługi komponentom, którymi zarządza, co pozwala na przetwarzanie operacji i kierowanie nimi w taki sposób, aby wykorzystać możliwości architektury sprzętowej używanej przez system SQL Server. SQLOS obsługuje też dynamiczne zmiany koligacji, równoważenie obciążenia, dynamiczne zarządzanie pamięcią, połączenia DAC i zarządzanie zasobami podzielonymi na partycje.

W systemie SQL Server 2014 używana jest ogólna platforma zarządzania pamięcią podręczną (to także komponent systemu SQLOS), co pozwala precyzyjnie sterować wciąż rosnącą grupą mechanizmów obsługi pamięci podręcznej (magazynami pamięci podręcznej, użytkownika i obiektów). Wspomniana platforma usprawnia pracę tych mechanizmów, ponieważ zapewnia wspólne zasady, które można stosować do wewnętrznych mechanizmów pamięci podręcznej w celu zarządzania nimi w różnych warunkach.

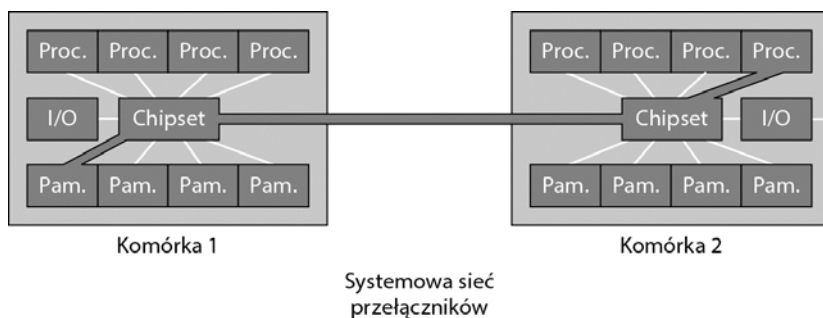
We wcześniejszych wersjach systemu SQL Server wprowadzono usprawnienie śledzenia pamięci — mechanizm *Memory Broker*. Umożliwia on śledzenie zdarzeń związanych z pamięcią w całym systemie operacyjnym. Memory Broker zarządza dynamicznym wykorzystaniem wewnętrznej pamięci systemu SQL Server i śledzi ją. Na podstawie wewnętrznego zużycia pamięci i zapotrzebowania na nią Memory Broker automatycznie wyznacza optymalną konfigurację pamięci dla komponentów, takich jak pula buforów, optymalizator, mechanizm wykonywania kwerend i pamięć podręczna. Informacje o konfiguracji pamięci są przekazywane do poszczególnych

komponentów w celu wprowadzenia zmian. System SQL Server obsługuje też dynamiczne zarządzanie standardową pamięcią z dużymi stronami i blokadami, a także wspomnianą wcześniej funkcję dynamicznego dodawania pamięci.

Dynamiczne dodawanie pamięci pozwala zastosować na serwerze dodatkową pamięć bez wyłączania go. Aby można było użyć tej funkcji, musi ją obsługiwać używany sprzęt, a także trzeba zastosować system Windows Server 2012 i edycję SQL Server 2014 Enterprise.

Lokalność danych

Lokalność danych polega na udostępnianiu w lokalnym węźle architektury NUMA wszystkich danych potrzebnych procesorowi w trakcie przetwarzania żądania. Cała pamięć z systemu jest dostępna dla dowolnego procesora w każdym węźle tej architektury. Warto wprowadzić tu rozróżnienie na *pamięć bliską* i *pamięć daleką*. Lepiej korzystać z pamięci bliskiej, ponieważ jest używana przez procesor z tego samego węzła architektury NUMA. Na rysunku 11.12 pokazano, że dostęp do pamięci dalekiej jest kosztowny, ponieważ żądanie trzeba przesłać z danego węzła NUMA przez sieć przełączników do innego węzła NUMA, w którego pamięci znajdują się potrzebne informacje.



Rysunek 11.12. Dostęp do pamięci bliskiej i dalekiej

Koszt dostępu z pamięci dalekiej jest często trzy, a nawet więcej razy wyższy niż w przypadku pamięci bliskiej. Za lokalność danych odpowiada sam system. Jednym ze sposobów na ograniczenie problemów z pamięcią jest zainstalowanie dodatkowej pamięci w każdym węźle architektury NUMA.

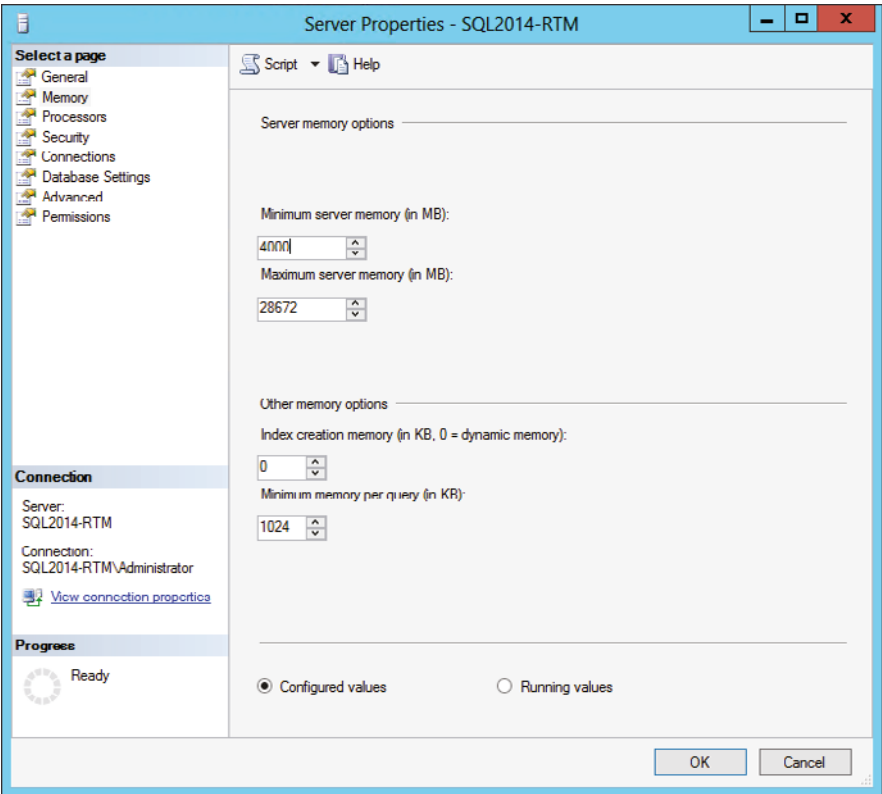
Maksymalna ilość pamięci dla serwera

Gdy maksymalna ilość pamięci dla serwera (opcja *Maximum server memory*) jest określana dynamicznie (jest to ustawienie domyślne), system SQL Server zajmuje i zwalnia pamięć w reakcji na wewnętrzne i zewnętrzne zapotrzebowanie na nią. Jeśli nie będziesz tego kontrolował, system SQL Server zajmie całą dostępną pamięć, dlatego zaleca się ustawienie opcji *Maximum server memory*. System operacyjny Windows potrzebuje pamięci, aby działać, dlatego systemowi SQL Server należy przyznać całą pamięć pomniejszoną o 8 – 16 gigabajtów. W tabeli 11.3 przedstawione są wskazówki dotyczące konfiguracji wspomnianej opcji.

Na rysunku 11.13 pokazane jest przykładowe ustawienie opcji *Maximum server memory* dla systemu SQL Server 2014.

Tabela 11.3. Wskazówki dotyczące opcji Maximum server memory

Pamięć w systemie łącznie (w gigabajtach)	Pamięć zarezerwowana dla systemu operacyjnego (w gigabajtach)	Maksymalna ilość pamięć dla systemu SQL Server (w gigabajtach)
16	4	12
32	4	28
64	4	60
128	8	120
256	8	248



Rysunek 11.13. Ustawianie opcji Maximum server memory

Resource Governor

Resource Governor to technologia z systemu SQL Server ograniczająca ilość zasobów przydzielanych każdej operacji bazy danych z puli wszystkich zasobów dostępnych systemowi SQL Server 2014. Gdy Resource Governor jest włączona, określa kategorię każdej sesji i ustala, do której grupy operacyjnej dana sesja należy. Każda grupa operacyjna jest wiązana z przeznaczoną dla niej pulą zasobów. Ponadto Resource Governor chroni system SQL Server przed niekończącymi

się kwerendami i przed nieprzewidywalnym wykonywaniem operacji, a także określa priorytety operacji. Aby Resource Governor mogła ograniczać dostęp do zasobów, musi rozróżniać operacje w następujący sposób. Powinna:

- klasyfikować przychodzące połączenia i kierować je do określonych grup operacyjnych,
- monitorować wykorzystanie zasobów w każdej grupie operacyjnej,
- tworzyć pule zasobów z ograniczoną ilością czasu procesora, pamięci i zasobów wejścia-wyjścia,
- identyfikować operacje i grupować je, a następnie przypisywać grupy do określonych pul zasobów,
- określać priorytety operacji w ich grupach.

Oto ograniczenia dotyczące technologii Resource Governor.

- Działa ona tylko dla relacyjnego silnika systemu SQL Server. Nie obsługuje usług Analysis Services, Report Services i Integration Services. Ponadto Resource Governor wymaga licencji na edycję Enterprise.
- Resource Governor nie może monitorować zasobów dla kilku egzemplarzy systemu SQL Server. Można jednak wykorzystać menedżera zasobów systemu Windows do monitorowania zasobów różnych procesów, w tym procesów systemu SQL Server.
- Resource Governor może ograniczyć dostęp do czasu procesora, pamięci i zasobów wejścia-wyjścia.
- Typowe operacje OLTP obejmują niewielkie i szybkie operacje bazy danych, z których każda wymaga niewiele czasu procesora i może być za mała, aby Resource Governor miała możliwość nałożenia ograniczeń.

Podstawowe elementy technologii Resource Governor

Główne elementy technologii Resource Governor to pule zasobów, grupy operacyjne i mechanizmy klasyfikacyjne. Ich omówienie podajemy w poniższych podpunktach.

Pule zasobów

Pule zasobów to fizyczne zasoby systemu SQL Server. W trakcie instalowania tego systemu tworzone są dwie pule — *wewnętrzna* i *domyślna*. Zasoby, którymi można zarządzać, to minimalny i maksymalny dostęp do procesora oraz minimalna i maksymalna ilość pamięci. Domyślnie Resource Governor jest wyłączona. Aby ją włączyć, zastosuj polecenie `ALTER RESOURCE GOVERNOR RECONFIGURE`. Jeśli zechcesz ponownie ją wyłączyć, wywołaj instrukcję `ALTER RESOURCE GOVERNOR DISABLE`.

Pula wewnętrzna jest używana dla wewnętrznych funkcji systemu SQL Server. Obejmuje ona tylko wewnętrzną grupę operacji. Tej puli nie można zmodyfikować ani ograniczyć.

Pula domyślna to wstępnie zdefiniowana pula zasobów użytkownika. Obejmuje ona domyślną grupę operacyjną i może zawierać grupy operacyjne zdefiniowane przez użytkownika. Tę pulę można zmodyfikować, ale nie można jej utworzyć ani usunąć.

Do tworzenia pul definiowanych przez użytkownika służą instrukcja DDL `CREATE RESOURCE POOL` i program SQL Server Management Studio. Ponadto można zmodyfikować pulę za pomocą polecenia `ALTER RESOURCE POOL` i usunąć ją przy użyciu instrukcji `DROP RESOURCE POOL`. Można zdefiniować do dwudziestu pul zasobów (ta liczba obejmuje pule wewnętrzne i domyślne).

Poniższy kod tworzy pulę zasobów, którą zarządza Resource Governor:

```
CREATE RESOURCE POOL pool_name
[ WITH
    ( [ MIN_CPU_PERCENT = value ]
      [ [ , ] MAX_CPU_PERCENT = value ]
      [ [ , ] CAP_CPU_PERCENT = value ]
      [ [ , ] AFFINITY { SCHEDULER = AUTO | (Scheduler_range_spec) | NUMANODE =
        (NUMA_node_range_spec) } ] [ [ , ] MIN_MEMORY_PERCENT = value ]
      [ [ , ] MAX_MEMORY_PERCENT = value ]
      [ [ , ] MIN_IOPS_PER_VOLUME = value ]
      [ [ , ] MAX_IOPS_PER_VOLUME = value ] )
] [ ; ]
```

UWAGA Opcje `MIN_IOPS_PER_VOLUME` i `MAX_IOPS_PER_VOLUME` zostały wprowadzone w wersji SQL Server 2014. Więcej informacji na ich temat znajdziesz w dokumentacji Books Online: [http://msdn.microsoft.com/en-us/library/bb934024\(v=SQL.120\).aspx](http://msdn.microsoft.com/en-us/library/bb934024(v=SQL.120).aspx).

Grupy operacyjne

Grupa operacyjna (ang. *workload group*) to kontener obejmujący podobne sesje (według zdefiniowanych reguł klasyfikacyjnych) i stosujący do każdej sesji z grupy określone zasady. Istnieją dwie predefiniowane grupy operacyjne — *wewnętrzna* i *domyślna*. Grupa wewnętrzna jest powiązana z wewnętrzną pulą zasobów i nie można jej modyfikować. Domyślna grupa operacyjna jest połączona z domyślną pulą zasobów. Tę grupę stosuje się, gdy nie istnieje zdefiniowana przez użytkownika funkcja klasyfikacyjna, gdy wystąpi błąd klasyfikacji lub gdy zdefiniowana przez użytkownika funkcja klasyfikacyjna zwraca wartość NULL.

Do tworzenia zdefiniowanych przez użytkownika grup operacyjnych służy polecenie `CREATE WORKLOAD GROUP`, do modyfikowania takich grup używana jest instrukcja `ALTER WORKLOAD GROUP`, a usuwać grupy można przy użyciu polecenia `DROP WORKLOAD GROUP`.

W poniższym kodzie pokazano, jak tworzyć grupy operacyjne, którymi zarządza Resource Governor:

```
CREATE WORKLOAD GROUP group_name
[ WITH
    ( [ IMPORTANCE = { LOW | MEDIUM | HIGH } ]
      [ [ , ] REQUEST_MAX_MEMORY_GRANT_PERCENT = value ]
      [ [ , ] REQUEST_MAX_CPU_TIME_SEC = value ]
      [ [ , ] REQUEST_MEMORY_GRANT_TIMEOUT_SEC = value ]
      [ [ , ] MAX_DOP = value ]
      [ [ , ] GROUP_MAX_REQUESTS = value ] )
]
[ USING { pool_name | "default" } ] [ ; ]
```

Dla grupy operacyjnej można ustawić kilka opcji:

- maksymalną ilość alokowanej pamięci dla żądania,
- maksymalny czas procesora dla żądania,
- maksymalną liczbę operacji wejścia-wyjścia na sekundę dla żądania,
- minimalną liczbę operacji wejścia-wyjścia na sekundę dla żądania,
- limit czasu oczekiwania na zasoby dla żądania,
- względny priorytet żądania,

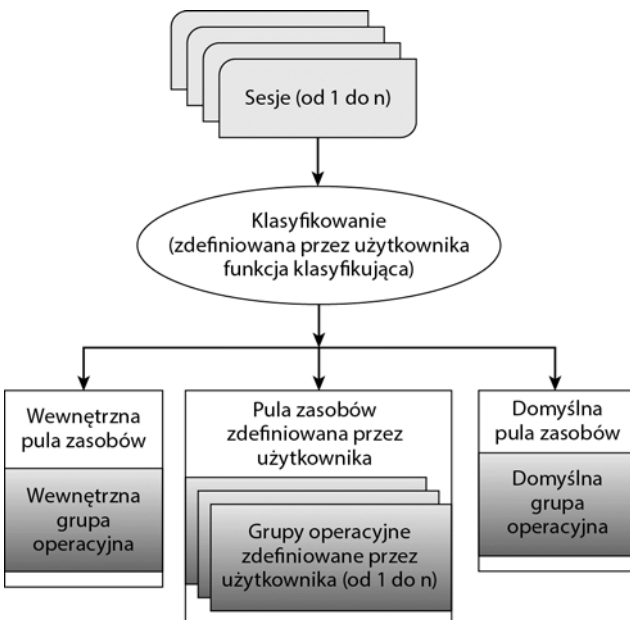
- maksymalną liczbę żądań na grupę,
- maksymalny poziom współbieżności,
- określoną pulę zasobów.

Klasyfikowanie

Resource Governor może klasyfikować przychodzące połączenia i przydzielać je na tej podstawie do istniejących grup operacyjnych. Obsługuje reguły systemowe i zdefiniowane przez użytkownika reguły oparte na określonych atrybutach połączeń. Gdy użytkownik nie zdefiniował funkcji klasyfikacyjnej, wybierana jest domyślna grupa operacyjna. Jeśli w mechanizmie Resource Governor zarejestrowano funkcję klasyfikacyjną, jest ona uruchamiana dla każdego nowego połączenia, po czym połączenie jest przypisywane do jednej z istniejących grup operacyjnych, gdzie ma dostęp do ograniczonej puli zasobów. Można utworzyć tylko jedną funkcję klasyfikacyjną. Aby zaczęła działać, należy ją zarejestrować i wywołać polecenie `ALTER RESOURCE GOVERNOR RECONFIGURE`.

Resource Governor przy identyfikowaniu połączeń korzysta ze specyficznych dla nich funkcji: `HOST_NAME()`, `APP_NAME()`, `SUSER_NAME()`, `SUSER_SNAME()`, `IS_SRVROLEMEMBER()` i `IS_MEMBER()`.

Na rysunku 11.14 przedstawiono komponenty mechanizmu Resource Governor i zależności między nimi.



Rysunek 11.14. Komponenty mechanizmu Resource Governor

Na listingu 11.4 (patrz plik *KodR11.sql*) przedstawiony jest kod używający mechanizmu Resource Governor. Kod tworzy trzy pule zasobów odpowiadające wymaganiom trzech różnych grup operacyjnych.

Listing 11.4. Pule zasobów

```

USE Master;
BEGIN TRAN;
CREATE RESOURCE POOL poolAdhoc with
( MIN_CPU_PERCENT = 10, MAX_CPU_PERCENT = 30,
  MIN_MEMORY_PERCENT= 15, MAX_MEMORY_PERCENT= 25,
  MIN_IOPS_PER_VOLUME=0, MAX_IOPS_PER_VOLUME=2147483647);

CREATE RESOURCE POOL poolReports with
( MIN_CPU_PERCENT = 20, MAX_CPU_PERCENT = 35,
  MIN_MEMORY_PERCENT= 15, MAX_MEMORY_PERCENT= 45,
  MIN_IOPS_PER_VOLUME=0, MAX_IOPS_PER_VOLUME=2147483647);

(MIN_IOPS_PER_VOLUME=0, MAX_IOPS_PER_VOLUME=2147483647);
ALTER RESOURCE POOL Customer2Pool WITH (MIN_IOPS_PER_VOLUME=0,
  MAX_IOPS_PER_VOLUME=2147483647);

CREATE RESOURCE POOL poolAdmin with
( MIN_CPU_PERCENT = 15, MAX_CPU_PERCENT = 25,
  MIN_MEMORY_PERCENT= 15, MAX_MEMORY_PERCENT= 30);

```

1. Najpierw utwórz trzy pule zasobów: dla kwerend jednorazowych, raportów i żądań od administratorów.
2. Następnie dodaj trzy grupy operacyjne i powiąż je z trzeba utworzonymi wcześniej pulami zasobów:

```

CREATE WORKLOAD GROUP groupAdhoc using poolAdhoc;
CREATE WORKLOAD GROUP groupReports with (MAX_DOP = 8) using poolReports;
CREATE WORKLOAD GROUP groupAdmin using poolAdmin;
GO

```

3. Teraz utwórz zdefiniowaną przez użytkownika funkcję klasyfikacyjną, która identyfikuje wszystkie nowe sesje i przypisuje je do jednej z trzech grup operacyjnych. Żądania, których ta funkcja nie potrafi sklasyfikować, są wiązane z domyślną grupą operacyjną:

```

CREATE FUNCTION rgclassifier_v1() RETURNS SYSNAME
WITH SCHEMABINDING
AS
BEGIN
    DECLARE @grp_name AS SYSNAME
    IF (SUSER_NAME() = 'sa')
        SET @grp_name = 'groupAdmin'
    IF (APP_NAME() LIKE '%MANAGEMENT STUDIO%')
        OR (APP_NAME() LIKE '%QUERY ANALYZER%')
        SET @grp_name = 'groupAdhoc'
    IF (APP_NAME() LIKE '%REPORT SERVER%')
        SET @grp_name = 'groupReports'
    RETURN @grp_name
END;
GO

```

4. Następnie zdefiniowaną przez użytkownika funkcję klasyfikacyjną zarejestruj w mechanizmie Resource Governor. Wszystkie wykonywane operacje trzeba umieścić w jawnie zadeklarowanej transakcji. Dzięki temu gdy wystąpi błąd użytkownika, operacje będzie można wycofać. W danym momencie w mechanizmie Resource Governor zarejestrowana może być tylko jedna funkcja klasyfikacyjna.

```

ALTER RESOURCE GOVERNOR WITH (CLASSIFIER_FUNCTION= dbo.rgclassifier_v1);
COMMIT TRAN;

```

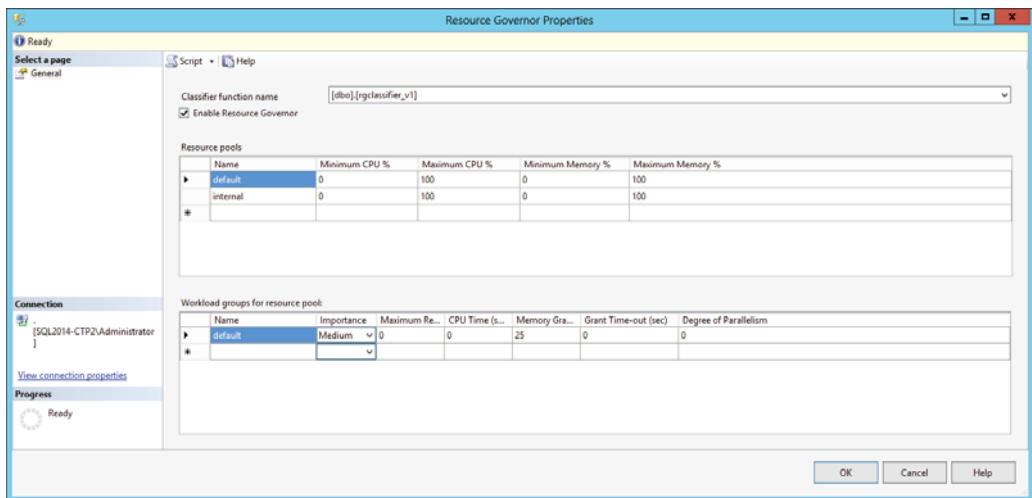
5. W ostatnim kroku w celu wprowadzenia nowych zmian lub włączenia mechanizmu Resource Governor wywołaj następujące polecenie:

```
ALTER RESOURCE GOVERNOR RECONFIGURE;
```

Używanie technologii Resource Governor w programie SQL Server 2014 Management Studio

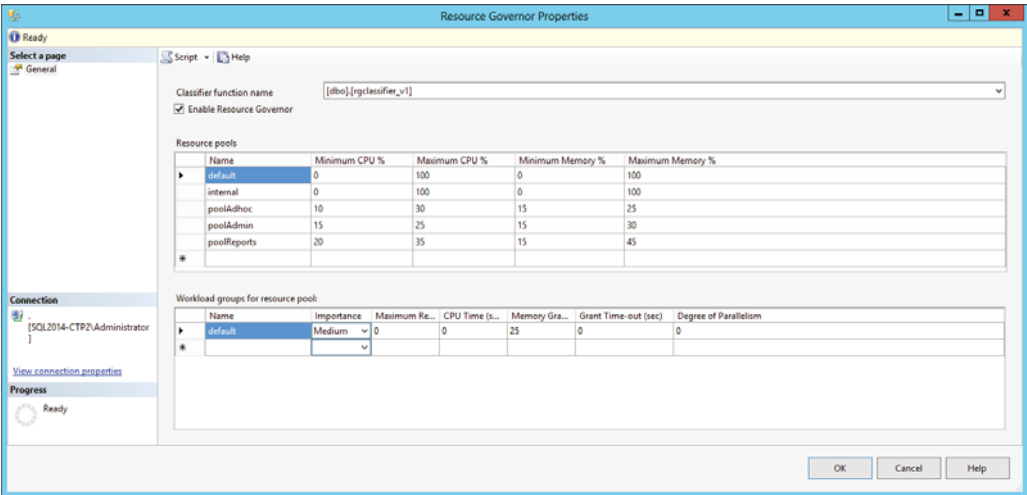
W oknie *Object Explorer* w programie SQL Server 2014 Management Studio węzeł *Resource Governor* jest dostępny w węźle *Management*. Domyślnie Resource Governor jest wyłączony. Aby go włączyć, kliknij prawym przyciskiem myszy węzeł *Resource Governor*, a następnie wybierz opcję *Enable*. Ponadto możesz w oknie *Object Explorer* kliknąć prawym przyciskiem myszy ten sam węzeł, a następnie wybrać opcję *Properties*, aby dodać, usunąć lub zmodyfikować właściwości pul zasobów oraz grup operacyjnych, by dodać lub usunąć zdefiniowaną przez użytkownika funkcję klasyfikacyjną albo włączyć mechanizm Resource Governor.

W oknie dialogowym *Resource Governor Properties* widoczne są dwie pule zasobów (*internal* i *default*) oraz dwie odpowiadające im grupy operacyjne. Są one tworzone przez system SQL Server 2014 w trakcie jego instalowania. Do momentu utworzenia funkcji klasyfikacyjnej nie jest ona dostępna. Jeśli Resource Governor jest włączony, używa domyślnych reguł wbudowanych i domyślnej puli (patrz rysunek 11.15).



Rysunek 11.15. Domyślna konfiguracja mechanizmu Resource Governor

Po uruchomieniu skryptu z listingu 11.6, co pozwala utworzyć trzy zdefiniowane przez użytkownika pule zasobów i grupy operacyjne oraz zdefiniowaną przez użytkownika funkcję klasyfikacyjną, i po włączeniu mechanizmu Resource Governor właściwości powinny wyglądać tak, jak na rysunku 11.16.



Rysunek 11.16. Właściwości mechanizmu Resource Governor po wprowadzeniu opisanych zmian

Monitorowanie pracy mechanizmu Resource Governor

System SQL Server 2014 obejmuje dwa obiekty rejestrujące statystyki dotyczące grup operacyjnych i pul w każdym egzemplarzu tego systemu.

- Obiekt *SQLServer: Resource Pool Stats* rejestruje statystyki dotyczące zasobów. Więcej informacji na ten temat zawiera artykuł na stronie [http://msdn.microsoft.com/en-us/library/cc645958\(v=SQL.120\).aspx](http://msdn.microsoft.com/en-us/library/cc645958(v=SQL.120).aspx).
- Obiekt *SQLServer: Workload Group Stats* rejestruje statystyki dotyczące grup operacyjnych. Więcej informacji na ten temat zawiera artykuł na stronie [http://msdn.microsoft.com/en-us/library/cc627354\(v=SQL.120\).aspx](http://msdn.microsoft.com/en-us/library/cc627354(v=SQL.120).aspx).

Ponadto widoki DMV powiązane z mechanizmem Resource Governor zwracają informacje na temat zasobów opisane w tabeli 11.4.

Tabela 11.4. Widoki DMV mechanizmu Resource Governor

Widok DMV mechanizmu Resource Governor	Opis
sys.dm_resource_governor_configuration	Bieżący stan usługi działającej w pamięci.
sys.dm_resource_governor_resource_pools	Stan, konfiguracja i statystyki dotyczące puli zasobów.
sys.dm_resource_governor_workload_groups	Statystyki i konfiguracja grupy operacyjnej.

UWAGA Szczegółowe informacje na temat technologii Resource Governor znajdziesz w dokumentacji Books Online systemu SQL Server 2014 pod hasłem „Resource Governor” ([http://msdn.microsoft.com/en-us/library/bb933866\(v=SQL.120\).aspx](http://msdn.microsoft.com/en-us/library/bb933866(v=SQL.120).aspx)).

Podsumowanie

W trakcie przygotowywania i konfigurowania systemu SQL Server 2014 trzeba uwzględnić wiele ważnych kwestii. Należy umieścić pliki bazy danych (z danymi i dzienniki) na odrębnych jednostkach LUN opartych na szybkich dyskach. Baza tempdb powinna znajdować się na osobnych, jak najszybszych dyskach. Możesz zastosować partycje, aby zwiększyć wydajność bazy i ułatwić jej konserwację. Za pomocą kompresji można zwiększyć wydajność operacji wejścia-wyjścia, trzeba jednak uważać, aby nie przeciążyć procesora. Procesor i pamięć konfiguruje się z poziomu systemu SQL Server, przy czym trzeba unikać konfiguracji prowadzącej do spadku wydajności. Pamiętaj, aby zawsze przeprowadzać gruntowne testy!

Jeśli poprawnie zastosujesz informacje z tego rozdziału, system SQL Server 2014 będzie automatycznie się dostrajał, aby zapewnić przewidywalną dostępność i wydajność na wymaganym przez firmę poziomie.

Po przeczytaniu rozdziału 12. będziesz wiedział, jak monitorować system SQL Server.

Monitorowanie systemu SQL Server

ZAWARTOŚĆ ROZDZIAŁU:

- Monitorowanie pracy systemu SQL Server za pomocą obiektów DM.
- Monitorowanie dziennika błędów systemu SQL Server i dzienników zdarzeń systemu Windows.
- Monitorowanie systemu SQL Server przy użyciu narzędzia System Center Advisor.

Dobry mechanizm monitorowania pozwala odejść od reaktywnego radzenia sobie z różnymi zdarzeniami w kierunku proaktywnego diagnozowania problemów i naprawiania, zanim jeszcze użytkownicy zdążą je dostrzec. Czytając ten rozdział, dowiesz się, jak proaktywnie monitorować system SQL Server, tak by zapobiegać zdarzeniom lub reagować na nie, zanim użytkownicy zaczną dzwonić z pretensjami.

Oto krótki przykład. Załóżmy, że niedawno przejąłeś istniejący system po administratorze przeniesionym do innego zespołu. Aplikacje w systemie działają poprawnie, jednak codziennie musisz coś poprawiać: zapełnią się dzienniki transakcji, zabranie pamięci dla bazy tempdb, utworzonych zostanie zbyt wiele blokad lub zapełnią się grupy plików. Nie są to żadne poważne problemy, a jedynie powoli i stale wykrywane sprawy, którymi trzeba się zająć. Jednak setki takich drobiazgów mogą być prawdziwym koszmarem administratora. Niezbędne zadania z zakresu konserwacji wymagają coraz więcej uwagi, aż w pewnym momencie okazuje się, że administrator nie ma czasu na nic innego.

Po kilku tygodniach udaje Ci się opracować nowe mechanizmy monitorowania i wprowadzić szereg proaktywnych zmian, które pozwalają rozwiązywać problemy, zanim te zaczną być odczuwalne. W tych zmianach nie ma niczego niezwykłego. Są to proste rozwiązania, np. przeniesienie danych do nowej grupy plików, włączenie automatycznego powiększania dla kilku plików i ustawienie poziomu, do którego mogą się rozrastać, a także odtworzenie złe

pofragmentowanych indeksów zajmujących za dużo miejsca. Wszystko to zajmuje znacznie mniej czasu niż radzenie sobie z ciągłymi usterkami. Ponadto komfort pracy użytkowników jest w nowym modelu znacznie wyższy.

Co jest potrzebne do rozpoczęcia monitorowania systemu? Nie potrzeba dużych nakładów pracy. Wystarczy kilka prostych kroków, czyli monitorowanie za pomocą języka T-SQL wolnego miejsca w tabelach, bazach danych i grupach plików, użytkowania indeksów i fragmentacji. Trzeba też zrobić kilka rzeczy, aby monitorować wykorzystanie zasobów i znaleźć istotne słabe punkty systemu. Po wykryciu słabych punktów można przejść do ich naprawiania.

Skoro już znasz wartość monitorowania systemu SQL Server, pora zobaczyć, jak to robić.

Cele monitorowania

Celem monitorowania baz danych jest ustalenie, co dzieje się w systemie SQL Server. Należy zbadać, jak skutecznie system SQL Server wykorzystuje zasoby serwera (czas procesora, pamięć i podsystem wejścia-wyjścia). Informacje są potrzebne, aby można było zobaczyć, jak dobrze system funkcjonuje. Dane należy zbierać przez pewien czas, aby zbudować profil normalnej pracy systemu. Ile zasobów różnego rodzaju używanych jest w każdym elemencie cyklu pracy systemu? Na podstawie danych zebranych w dłuższym okresie możesz zacząć tworzyć punkt odniesienia opisujący normalne działanie systemu. Ten punkt odniesienia pozwala wykryć nietypowe zachowanie środowiska, które — jeśli nie zostanie zbadane — może prowadzić do problemów.

Nietypowe zachowanie może polegać na wyższym tempie rozrastania się określonej tabeli, zmianie w szybkości replikacji, dłuższym czasie przetwarzania kwerendy lub zadania albo wykorzystaniu nieoczekiwanie dużej ilości ograniczonych zasobów serwera. Możliwość identyfikowania tych anomalii, zanim doprowadzą do problemów, które staną się przyczyną telefonów i skarg od użytkowników, znacznie ułatwia życie. Na podstawie zebranych danych ustalisz, gdzie mogą wystąpić kłopoty, a także jakie zmiany trzeba wprowadzić, aby rozwiązać przyczyny problemu przed jego nasileniem.

Czasem monitorowanie jest związane z problemami z wydajnością, np. z wolno działającymi kwerendami lub zakleszczeniami, jednak w wielu sytuacjach dane są informacją, co można zmienić, aby uniknąć przyszłych kłopotów.

Jest to zgodne z filozofią „lepiej zapobiegać niż leczyć” — opisane podejście to swoista medycyna zapobiegawcza dla systemu SQL Server.

Określanie celów monitorowania

Zanim zaczniesz monitorowanie, najpierw musisz jednoznacznie zidentyfikować powody, dla których się tego podejmujesz. Oto wybrane z nich:

- ustalenie punktu odniesienia,
- zidentyfikowanie trendów zanim wystąpią problemy,
- monitorowanie rozrastania się bazy danych,
- identyfikowanie codziennych, cotygodniowych i comiesięcznych zadań z zakresu konserwacji,

- identyfikowanie zmian w wydajności,
- kontrolowanie działań użytkowników,
- diagnozowanie konkretnych problemów z wydajnością.

Określanie punktu odniesienia

Monitorowanie jest niezwykle ważne, jeśli chcesz zapewnić płynną pracę systemu SQL Server. Jednak samo monitorowanie (i określanie wartości najważniejszych wskaźników w danym momencie) nie przynosi dużych korzyści, jeśli nie masz punktu odniesienia, z którym możesz porównać uzyskane wyniki. Czy poziom 50 transakcji na sekundę to dobry, średni, czy zły wynik? Czy normalne jest, że serwer obciąża procesor na poziomie 75%? Czy ta wartość jest typowa dla danej pory dnia, dnia tygodnia lub miesiąca? Mając punkt odniesienia opisujący wydajność systemu, będziesz miał z czym porównać uzyskane wyniki.

Jeśli punktem odniesienia jest 30 transakcji na sekundę, 50 transakcji to wysoka wartość, oznaczająca, że system może przetworzyć więcej transakcji niż zwykle. Jednak ten wynik może też informować, że w systemie dzieje się coś, co prowadzi do wzrostu liczby transakcji. Format punktu odniesienia zależy — oczywiście — od używanego systemu. W niektórych sytuacjach można zapisać zbiór dzienników monitora wydajności z informacjami o najważniejszych zasobach serwera i wartości liczników systemu SQL Server zarejestrowane w kilku okresach wysokiej aktywności systemu lub w trakcie testów obciążeniowych. W innych scenariuszach punkt odniesienia może być wynikiem analiz śladu z narzędzia SQL Server Profiler lub dziennika zdarzeń rozszerzonych zarejestrowanego przy wysokiej aktywności systemu. Analizy mogą być bardzo proste i polegać na utworzeniu listy wywołań procedur składowanych przez daną aplikację wraz z częstotliwością uruchamiania tych procedur.

Aby ustalić, czy system SQL Server działa optymalnie, regularnie wykonuj pomiary (nawet wtedy, gdy nie występują problemy), aby ustalić wyjściową wydajność serwera. To, ile i jak dużych próbek potrzebujesz, zależy od charakteru operacji wykonywanych na serwerze. Jeśli serwer cyklicznie wykonuje określone operacje, próbki powinny objąć różne punkty kilku cykli, co pozwoli dobrze oszacować minimalną, maksymalną i średnią wydajność. Jeżeli serwer wykonuje jednorodne zadania, do oszacowania minimalnej, maksymalnej i średniej wydajności wystarczy mniejsza liczba próbek zebranych w krótszym czasie. Na podstawie wyjściowej wydajności ustal przynajmniej następujące wskaźniki:

- godziny maksymalnej aktywności systemu,
- czas odpowiedzi przy przetwarzaniu kwerend i operacji wsadowych.

Ważne jest to, jak często należy powtarzać tworzenie punktu odniesienia. Jeśli system szybko się rozrasta, możliwe, że będziesz musiał robić to stosunkowo często. Gdy obecna wydajność zmieniła się o 15 – 25% w stosunku do dawnego punktu odniesienia, warto rozważyć ponowne jego wyznaczenie.

Porównywanie aktualnych wartości z punktem odniesienia

Ważnym aspektem porównywania obecnych wyników do punktu odniesienia jest ustalenie limitu rozbieżności, którego przekroczenie jest nieakceptowalne i sygnalizuje wymagające zbadania problemy.

Akceptowalny poziom rozbieżności zależy od aplikacji i wskaźnika. Przykładowo dla wskaźnika mierzącego wolne miejsce w grupie plików w systemie, w którym ilość danych szybko rośnie i stosowana jest agresywna strategia archiwizacji, limit może wynosić 20%. Gdy ilość wolnego miejsca spadnie poniżej tej wartości, należy zgłosić alert. W innym systemie, w którym ilość danych rośnie wolniej, dla tego samego wskaźnika można ustawić limit równy 5%.

Musisz sam ocenić (na podstawie wiedzy o szybkości rozrastania się systemu i występujących w nim zmian), jakie odchylenie od punktu odniesienia jest akceptowalne.

Wybór odpowiednich narzędzi monitorujących

Po zdefiniowaniu celów monitorowania należy wybrać właściwe narzędzia. Oto lista podstawowych narzędzi z tej grupy.

- **Monitor wydajności.** Jest to przydatne narzędzie śledzące zasoby w systemach operacyjnych Microsoftu. Pozwala monitorować wykorzystanie zasobów serwera i udostępnia informacje specyficzne dla systemu SQL Server (lokalnie oraz dla serwerów zdalnych). Za pomocą monitora wydajności możesz zarejestrować wyjściowy poziom wykorzystania zasobów serwera i monitorować dane przez dłuższy czas, aby wykryć trendy. Narzędzie przydaje się też do doraźnego monitorowania systemu w celu wykrycia zatorów powodujących problemy z wydajnością. Monitor wydajności można skonfigurować tak, aby generował alerty po przekroczeniu zdefiniowanych wartości progowych.
- **Zdarzenia rozszerzone.** Ten mechanizm zapewnia wysoce skalowalną i konfigurowalną architekturę, która umożliwia rejestrowanie informacji potrzebnych do rozwiązywania problemów z systemem SQL Server. Jest to lekki system z graficznym interfejsem użytkownika umożliwiającym łatwe tworzenie nowych sesji.

Zdarzenia rozszerzone udostępniają sesję `system_health`. Jest to domyślna sesja do analizy stanu. Generuje ona bardzo niewielkie koszty i nieustannie rejestruje informacje o systemie pomocne przy rozwiązywaniu problemów z wydajnością. Dzięki temu nie trzeba tworzyć niestandardowych sesji zdarzeń rozszerzonych.

Gdy otworzysz węzeł *Extended Events* w programie SQL Server Management Studio, zobaczysz dodatkową sesję domyślną `AlwaysOn_health`. Jest to nieudokumentowana sesja służąca do monitorowania stanu grup dostępności.

- **SQL Server Profiler.** Jest to aplikacja z graficznym interfejsem użytkownika umożliwiająca rejestrowanie śladu zdarzeń występujących w systemie SQL Server. Za pomocą tego narzędzia można uwzględnić w śladzie wszystkie zdarzenia systemu SQL Server. Ślad można zapisać w pliku lub w tabeli systemu SQL Server.

SQL Server Profiler umożliwia też ponowne uruchamianie zarejestrowanych zdarzeń. Jest to przydatne przy analizie obciążenia, testach i dostrajaniu wydajności. SQL Server Profiler może monitorować lokalne i zdalne egzemplarze systemu SQL Server. Ponadto mechanizmy tego narzędzia można wykorzystać w niestandardowej aplikacji przy użyciu systemowych procedur składowanych z tego programu.

SQL Server Profiler został uznany za przestarzały w wersji SQL Server 2012, dlatego powinienś zaplanować rezygnację z tego narzędzia na rzecz zdarzeń rozszerzonych (jeżeli chcesz rejestrować ślad) i opisanego dalej w tym rozdziale mechanizmu Distributed Replay (jeżeli chcesz ponownie uruchamiać zdarzenia).

- **SQL Trace.** SQL Trace umożliwia tworzenie śladu w systemie SQL Server za pomocą procedur składanych w języku T-SQL, bez konieczności uruchamiania narzędzia SQL Server Profiler. Konfigurowanie tej aplikacji jest bardziej pracochłonne niż używanie narzędzia SQL Server Profiler, ale SQL Trace wymaga niewiele zasobów, a ponadto współdziała ze skryptami, dlatego umożliwia zautomatyzowanie rejestrowania śladu. Można zatem wielokrotnie rejestrować te same zdarzenia.

Ponieważ SQL Server Profiler został uznany za przestarzały, do monitorowania systemu z wykorzystaniem śladów powinienś używać zdarzeń rozszerzonych.

- **Domyślny ślad.** Domyślny ślad został wprowadzony w systemie SQL Server 2005. Jest to prosty mechanizm, który działa w pętli nieskończonej i rejestruje najważniejsze zdarzenia dotyczące bazy danych i serwera. To narzędzie jest przydatne przy diagnozowaniu zdarzeń, które wystąpiły, gdy nie działały inne mechanizmy monitorujące.
- **Narzędzie Activity Monitor w SQL Server Management Studio.** To narzędzie w formie graficznej wyświetla informacje, takie jak:
 - procesy działające w danym egzemplarzu systemu SQL Server,
 - procesy oczekujące na zasoby,
 - operacje wejścia-wyjścia na plikach danych,
 - niedawne kosztowne kwerendy.
- **Widoki DMV i funkcje DMF.** Widoki DMV (ang. *Dynamic Management View*) i funkcje DMF (ang. *Dynamic Management Function*) zwracają informacje o stanie serwera, które można wykorzystać do monitorowania stanu egzemplarza serwera, diagnozowania problemów i dostrajania wydajności. Są to jedne z najlepszych narzędzi do doraźnego monitorowania dostępne w systemie SQL Server. Widoki DMV udostępniają dokładny stan systemu SQL Server z momentu wywołania kwerendy. Jest to niezwykle przydatne, jednak czasem zinterpretowanie znaczenia zwróconych danych wymaga dużo pracy, ponieważ udostępniana jest tylko aktualna wartość wewnętrznych liczników. Aby uzyskać przydatne informacje o trendzie, musisz zastosować dodatkowy kod. W licznych przykładach z podrozdziału „Monitorowanie systemu za pomocą widoków DMV i funkcji DMF”, dalej w tym rozdziale, pokazujemy, jak to zrobić.
- **Systemowe procedury składowane.** Niektóre systemowe procedury składowane (np. `sp_who`, `sp_who2` i `sp_lock`) udostępniają informacje przydatne przy monitorowaniu systemu SQL Server. Te procedury składowane najlepiej sprawdzają się przy doraźnym monitorowaniu systemu, a nie w analizie trendów.
- **Standardowe raporty.** Standardowe raporty dostępne w systemie SQL Server to doskonały sposób na sprawdzenie, co dzieje się w tym systemie, bez konieczności używania widoków DMV, zdarzeń rozszerzonych i domyślnego stanu.
- **System Center Advisor.** To rozszerzenie narzędzia SQL Server Best Practice Analyzer działa w chmurze i analizuje system SQL Server oraz udostępnia informacje zwrotne na temat jego konfiguracji i działania. Wykorzystuje do tego zbiór przyjętych najlepszych praktyk z zakresu konfigurowania i funkcjonowania tego systemu.

Teraz, gdy masz już ogólną wiedzę na temat możliwości tych narzędzi, pora przyjrzeć im się szczegółowo.

Monitor wydajności

Monitor wydajności (inna nazwa to perfmon) to interfejs użytkownika, z którym większość czytelników z pewnością zapozna się w trakcie monitorowania wydajności systemu. Monitor wydajności to narzędzie systemu operacyjnego Windows dostępne w katalogu *Narzędzia administracyjne* każdego komputera lub serwera z tym systemem. Omawiane narzędzie wyświetla w formie graficznej dane z liczników wydajności. Można używać przy tym wykresów (jest to ustawienie domyślne), histogramów, a także raportów tekstowych.

Monitor wydajności to ważne narzędzie, które nie tylko informuje o pracy systemu SQL Server, ale też o wydajności systemu Windows. Monitor wydajności udostępnia bardzo bogaty zbiór liczników, jednak niech Cię to nie przytłacza. W tym podrozdziale omawiamy wybrane z nich — prawdopodobnie nikt nie zna wszystkich.

Podrozdział ten nie jest wprowadzeniem do stosowania monitora wydajności (choć dalej przedstawiamy dwa wartościowe narzędzia, logman i relog, które znacznie ułatwiają posługiwanie się tym monitorem w środowisku produkcyjnym). Zamiast tego koncentrujemy się tu na wykorzystaniu możliwości monitora wydajności do diagnozowania problemów z wydajnością. Ogólne informacje na temat monitora wydajności znajdziesz w dokumentacji systemów Windows 8 i Windows Server 2012.

Jak wcześniej wspomniano, należy monitorować trzy rodzaje zasobów serwera:

- procesor,
- pamięć,
- zasoby wejścia-wyjścia (głównie dyskowe).

OKREŚLANIE CZĘSTOTLIWOŚCI ZBIERANIA PRÓBEK

Często pojawia się pytanie o częstotliwość zbierania próbek. Ta częstotliwość jest wyświetlana w polu *Próbkuj co* na zakładce *Ogólne* strony właściwości wykresu w monitorze wydajności.

Zgodnie z ogólną regułą im krótszy jest czas zbierania próbek, tym większa powinna być ich częstotliwość. Jeśli rejestrujesz dane przez 5 – 10 minut, zbieraj próbki co sekundę. Jeżeli rejestrowanie ma trwać przez kilka dni, lepiej zbierać próbki co 15 lub 30 sekund, a nawet co minutę lub co 5 minut.

Najważniejsze jest, aby plik z zarejestrowanymi danymi nie był zbyt duży, a jednocześnie zawierał wystarczająco precyzyjne informacje, aby można było przeanalizować ciekawe zdarzenia. Jeśli interesują Cię zdarzenia zachodzące w krótkich okresach czasu, powinieneś zwiększyć częstotliwość zbierania próbek. Jeżeli zdarzenie trwa 10 – 15 sekund, powinieneś starać się uzyskać w tym czasie od 3 do 5 próbek. Tak więc dla 15-sekundowego zdarzenia możesz rejestrować próbki co 3 sekundy, otrzymasz zatem 5 próbek.

Ostatnią kwestią wartą uwagi jest to, że jeśli interesuje Cię wydajność zadań z zakresu konserwacji (np. tworzenia kopii zapasowych, konserwacji indeksów, archiwizowania danych itd.), można monitorować ją w nocy i w weekendy lub w innych godzinach, gdy firma nie pracuje. To właśnie w takich okresach standardowo przeprowadza się konserwację.

Aby uzyskać punkt odniesienia opisujący standardowe wartości, należy określić wartości najważniejszych liczników dla typowego okresu użytkowania systemu przez firmę. W zależności od cykli użytkowania systemu może to być dzień lub kilka dni, w których system jest obciążony w maksymalnym stopniu. Nie powinieneś rejestrować danych w weekend lub w okresie świątecznym. Potrzebujesz wiarygodnego obrazu tego, co się dzieje w trakcie typowego użytkowania systemu przez firmę, a nie z okresu, gdy system jest bezczynny. Powinieneś wykorzystać swoją wiedzę na temat firmy i monitorować szczytową aktywność systemu (np. pod koniec tygodnia, pod koniec miesiąca lub w innych wyjątkowych dniach). Po uzyskaniu dobrego punktu odniesienia stale monitoruj wydajność i zapisuj wyniki w repozytorium. Przydadzą się w czasie analizy przyczyn nieoczekiwanych problemów z wydajnością. Zebrane dane są przydatne także przy planowaniu ilości dostępnych zasobów.

Liczniki dotyczące zasobów procesora

Kilka liczników wyświetla stan dostępnych zasobów procesora. Zatory spowodowane przez brak zasobów procesora często są wynikiem nieoczekiwanej dużej liczby użytkowników, wykonywaniem kosztownych kwerend przez użytkowników lub standardowymi operacjami, takimi jak odtwarzanie indeksu.

Pierwszy krok przy ustalaniu przyczyn zatorów to określenie, czy problem wynika z braku zasobów procesora. Pomocne są przy tym następujące liczniki.

- **Obiekt: Procesor, licznik: Czas procesora (%)**. Licznik określa procent czasu, przez jaki poszczególne procesory są zajęte. W systemach wieloprocessorowych występuje też wersja sumaryczna tego licznika, mierząca całkowite wykorzystanie wszystkich procesorów. W komputerach z wieloma procesorami ta wersja licznika może nie wskazywać na zator, nawet jeśli występuje. Dzieje się tak, gdy kwerendy są wykonywane za pomocą jednego wątku lub liczby wątków mniejszej niż liczba procesorów. Taka sytuacja często występuje w systemach OLTP lub gdy opcja MAXDOP jest ustawiona na wartość mniejszą niż liczba dostępnych procesorów.

W takiej sytuacji przy wykonywaniu kwerendy może nastąpić zator na poziomie procesora, ponieważ kwerenda zajmuje 100% czasu procesora, do którego jest przydzielona, lub (w kwerendach równoległych) zużywa 100% czasu kilku procesorów. Jednak w obu przypadkach dostępne są inne, wolne procesory, z których dana kwerenda nie korzysta.

Jeśli wersja sumaryczna tego licznika regularnie wskazuje wartość powyżej 80%, jest to dobra wskazówka, że dany serwer działa na granicy swoich możliwości sprzętowych. Możesz wtedy kupić dodatkowe lub szybsze procesory albo zoptymalizować kwerendy, aby mniej obciążały procesor. Szczegółowe omówienie sprzętu zawiera rozdział 11., „Optymalizowanie systemu SQL Server 2014”.

- **Obiekt: System, licznik: Długość kolejki procesora**. Długość kolejki procesora określa, ile wątków oczekuje w stanie gotowości na to, aby procesor stał się dostępny i mógł je uruchomić. Interpretowanie i stosowanie tego licznika to zaawansowana metoda dostrajania wydajności systemu, potrzebna tylko w trakcie analizowania problemów ze skomplikowanym kodem wielowątkowym. W systemie SQL Server poziom wykorzystania procesora pozwala znacznie łatwiej zidentyfikować zatory w procesorze niż próby interpretowania tego licznika.
- **Obiekt: Procesor, licznik: Czas uprzywilejowany (%)**. Licznik określa, przez jaki procent próbkowanego czasu procesor działał w trybie jądra. W systemie SQL Server czas w trybie jądra to czas działania usług systemowych (np. menedżera pamięci lub, co bardziej prawdopodobne, menedżera wejścia-wyjścia). W większości sytuacji czas uprzywilejowany to czas odczytu i zapisu danych na dysku lub w sieci.

Warto obserwować wartość tego licznika, jeśli inne liczniki wskazują na wysokie obciążenie procesora. Jeśli ten licznik informuje, że 15 – 20% czasu zajmuje wykonywanie uprzywilejowanego kodu, może to świadczyć o problemie, np. ze sterownikami wejścia-wyjścia lub z zainstalowanymi przez oprogramowanie antywirusowe filtrami skanującymi dane albo pliki dziennika systemu SQL Server.

- **Obiekt: Proces, licznik: Czas procesora (%), egzemplarz: sqlservr.** Licznik mierzy procent próbki, przez który proces systemu SQL Server używał dostępnych procesorów. Gdy wartość w liczniku *Procesor: Czas procesora (%)* jest wysoka lub gdy podejrzewasz zator na poziomie procesora, sprawdź ten licznik, aby się upewnić, że procesor jest używany właśnie przez system SQL Server (a nie przez inny proces).
- **Obiekt: Proces, licznik: Czas uprzywilejowany (%), egzemplarz: sqlservr.** Licznik mierzy, przez jaki procent czasu próbki proces systemu SQL Server działał w trybie jądra. Jest to spędzona w trybie jądra część czasu procesora wyświetlanego przez poprzedni licznik. Ten licznik, podobnie jak opisany wcześniej, jest przydatny przy badaniu wysokiego obciążenia procesora na serwerze, ponieważ pozwala stwierdzić, że to system SQL Server zajmuje zasoby procesora.
- **Obiekt: Proces, licznik: Czas użytkownika (%), egzemplarz: sqlservr.** Licznik mierzy, przez jaki procent czasu próbki proces SQL Server działał w trybie użytkownika. Jest to spędzona w trybie użytkownika część czasu procesora wyświetlanego przez licznik *Czas procesora (%)*. Wartości liczników *Czas użytkownika (%)* i *Czas uprzywilejowany (%)* w sumie powinny równać się wartości licznika *Czas procesora (%)*.

UWAGA Po wykryciu zatoru na poziomie procesora należy ustalić główną przyczynę tego problemu. Możesz stwierdzić, że powodem jest jedna kwerenda, kilka kwerend, grupa użytkowników, aplikacja lub określone zadanie. Aby precyzyjnie zidentyfikować źródło problemu, musisz dokładnie zbadać, jakie elementy działają w systemie SQL Server. Po przeczytaniu punktu „Narzędzia do monitorowania wydajności”, dalej w tym rozdziale, będziesz wiedział, jak to zrobić. Po wykryciu zatoru na poziomie procesora zapoznaj się z odpowiednim rozdziałem, w którym znajdziesz szczegółowe wyjaśnienie tego, jak rozwiązać problem. Zwróć uwagę na rozdziały 10., „Konfigurowanie serwera pod kątem optymalnej wydajności”, 11., „Optymalizowanie systemu SQL Server 2014”, 13., „Dostrajanie wydajności kodu w języku T-SQL” i 14., „Indeksowanie baz danych”.

Aktywność dysków

System SQL Server wykonuje operacje wejścia-wyjścia za pośrednictwem systemu operacyjnego Windows. System dyskowy obsługuje zapisywanie i przenoszenie danych oraz ma istotny wpływ na ogólny czas reakcji systemu. Dyskowe operacje wejścia-wyjścia często są źródłem zatorów w systemie. Należy obserwować wiele czynników, aby ustalić wydajność systemu dyskowego. Niektóre z nich to poziom obciążenia, przepustowość, ilość dostępnego miejsca lub tworzenie się kolejki w systemie dyskowym.

Jeśli baza danych nie mieści się w fizycznej pamięci, system SQL Server nieustannie przenosi strony bazy danych do puli buforów i pobiera je z niej. Wymaga to wielu operacji wejścia-wyjścia. Podobnie rekordy dziennika trzeba przenieść na dysk przed uznaniem transakcji za zatwierdzoną. Od systemu SQL Server 2005 do wykonywania coraz większej liczby zadań używana jest baza tempdb. W systemie SQL Server 2014 to się nie zmieniło. Dlatego od wersji SQL Server 2005 także operacje wejścia-wyjścia na bazie tempdb mogą powodować zatory.

Liczne czynniki związane z dyskowymi operacjami wejścia-wyjścia są wzajemnie zależne od siebie. Jeśli np. obciążenie dysku jest wysokie, może zostać osiągnięta granica jego przepustowości, co prowadzi do wydłużania się opóźnień operacji wejścia-wyjścia i powstawania kolejki. Te warunki mogą skutkować dłuższym czasem reakcji i spadkiem wydajności.

Na wydajność operacji wejścia-wyjścia mają wpływ także inne czynniki, takie jak fragmentacja i mała ilość wolnego miejsca. Monitoruj ilość wolnej przestrzeni i podejmuj odpowiednie działania, gdy spadnie ona poniżej określonego progu. Zwykle należy zacząć reagować, gdy ilość wolnego miejsca spada poniżej 15 – 20%. Wtedy szybkość pracy wielu systemów dyskowych spada, ponieważ coraz więcej czasu znajduje wyszukiwanie pofragmentowanej wolnej przestrzeni.

W ramach monitorowania wydajności operacji wejścia-wyjścia zwróć uwagę na następujące podstawowe wskaźniki.

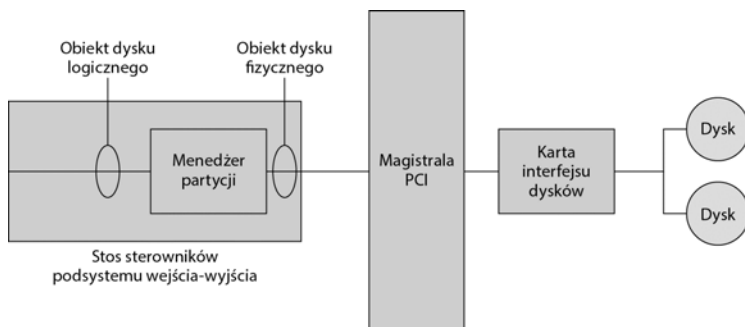
- **Przepustowość w liczbie operacji wejścia-wyjścia na sekundę.** Ile operacji wejścia-wyjścia na sekundę może obsłużyć dany podsystem dyskowy?
- **Przepustowość w megabajtach na sekundę.** Ile megabajtów na sekundę może przesłać podsystem wejścia-wyjścia?
- **Opóźnienie.** Jak długo trwa obsługa każdego żądania wejścia-wyjścia?
- **Długość kolejki.** Ile żądań wejścia-wyjścia oczekuje w kolejce?

Dla każdego z tych wskaźników należy też osobno uwzględnić odczyt i zapis.

Liczniki dysków fizycznych i logicznych

Użytkownicy często nie rozumieją różnic między licznikami dysków fizycznych i logicznych. W tym podpunkcie omawiamy podobieństwa i różnice między tymi licznikami, przedstawiamy przykładowe konfiguracje dysków, a także wyjaśniamy wyniki uzyskane dla poszczególnych konfiguracji.

Oto jeden ze sposobów myślenia o różnicach między licznikami dysków logicznych i fizycznych: *liczniki dysków logicznych* monitorują operacje wejścia-wyjścia dla żądań wychodzących z warstwy aplikacji (lub wchodzących do warstwy jądra), natomiast *liczniki dysków fizycznych* śledzą operacje wejścia-wyjścia wychodzące z dolnego poziomu stosu sterowników jądra. Na rysunku 12.1 przedstawiono stos oprogramowania dla operacji wejścia-wyjścia i pokazano, w których miejscach liczniki dysków logicznych i fizycznych monitorują takie operacje.



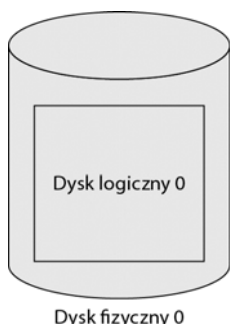
Rysunek 12.1. Dyski logiczne i fizyczne

W niektórych scenariuszach liczniki dysków logicznych i fizycznych zwracają te same wyniki. W innych sytuacjach otrzymywane są różne wyniki.

W następnych podpunktach omawiamy różne konfiguracje podsystemu wejścia-wyjścia wpływające na wartości zwracane przez liczniki dysków logicznych i fizycznych.

Jeden dysk, jedna partycja

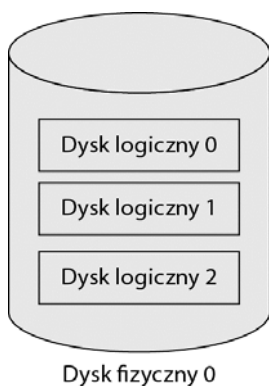
Na rysunku 12.2 przedstawiono jeden dysk z jedną partycją. W tym modelu używany jest jeden zestaw liczników dysku logicznego i jeden zestaw liczników dysku fizycznego. Taka konfiguracja dobrze sprawdza się w małych systemach SQL Server z kilkoma dyskami, gdy pliki danych i pliki dziennika są przechowywane na różnych dyskach o jednej partycji.



Rysunek 12.2. Jeden dysk, jedna partycja

Jeden dysk, wiele partycji

Na rysunku 12.3 pokazujemy jeden dysk podzielony na kilka partycji. Tu używanych jest kilka liczników dysków logicznych (po jednym na partycję) i jeden zestaw liczników dysku fizycznego. Taka konfiguracja nie zapewnia wzrostu wydajności, jednak umożliwia precyzyjne monitorowanie operacji wejścia-wyjścia w różnych partycjach. Jeśli na poszczególnych partycjach umieścisz różne zbiory danych, to za pomocą monitorowania liczników dysków logicznych będziesz mógł zobaczyć, ile operacji wejścia-wyjścia generują poszczególne zbiory.



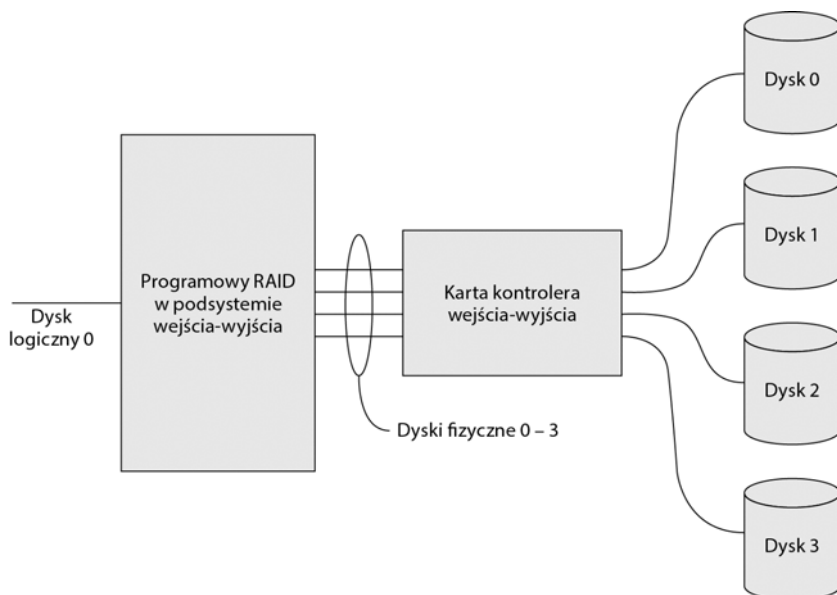
Rysunek 12.3. Jeden dysk, wiele partycji

Wadą tej konfiguracji jest to, że można błędnie przyjąć, iż używane są różne dyski fizyczne, i uznać, że operacje wejścia-wyjścia dla danych, dzienników i bazy tempdb są odizolowane. W rzeczywistości wszystkie te operacje są wykonywane na tym samym dysku fizycznym.

Oto przykładowa konfiguracja tego typu: pliki dziennika systemu SQL Server są przechowywane na jednej partycji, pliki danych bazy tempdb na drugiej, grupa plików dla danych na trzeciej, grupa plików dla indeksów na czwartej, a kopie zapasowe na piątej.

Wiele dysków, jeden wolumin z wykorzystaniem programowej macierzy RAID

Na rysunku 12.4 przedstawiono wiele dysków skonfigurowanych jako programowa macierz RAID i zamontowanych jako jeden wolumin. W tej konfiguracji używany jest jeden zestaw liczników dysków logicznych i wiele zestawów liczników dysków fizycznych.



Rysunek 12.4. Wiele dysków połączonych w jeden wolumin

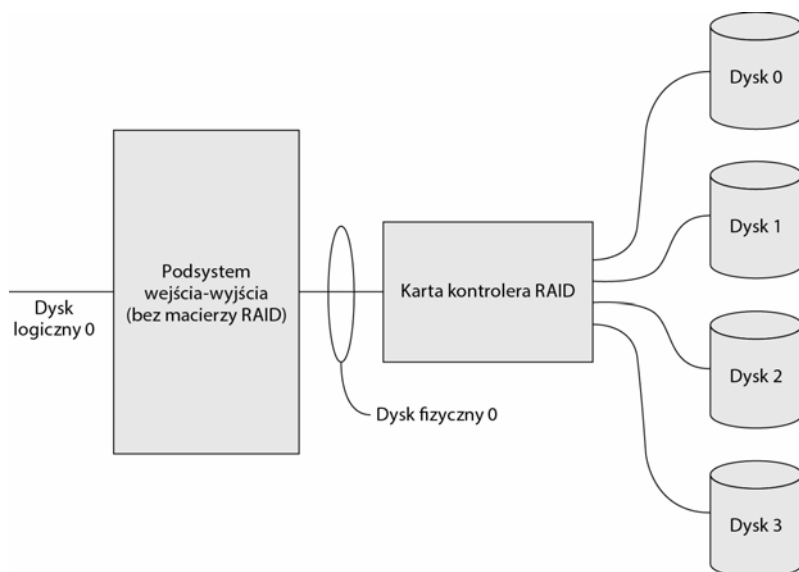
Taka konfiguracja działa dobrze w małych systemach SQL Server, gdy budżet na sprzęt nie pozwala na zakup sprzętowego kontrolera macierzy RAID, ale dostępnych jest kilka dysków i firma chce zbudować obejmujący je wolumin RAID. Więcej informacji o macierzach RAID podajemy w rozdziale 10., „Konfigurowanie serwera pod kątem optymalnej wydajności”.

Wiele dysków, jeden wolumin z wykorzystaniem sprzętowej macierzy RAID

Na rysunku 12.5 można zobaczyć sprzętową macierz RAID. W tej konfiguracji kilkoma dyskami zarządza sprzętowy kontroler macierzy RAID. System operacyjny widzi tylko jeden dysk fizyczny udostępniany przez kartę kontrolera macierzy. Używane są tu te same liczniki, co dla konfiguracji z jednym dyskiem i jedną partycją (czyli jeden zestaw liczników dysku fizycznego i jeden zestaw liczników dysku logicznego).

Monitorowanie przepustowości w operacjach wejścia-wyjścia na sekundę

Liczniki *Dysk fizyczny: Zapisy dysku/s* i *Dysk fizyczny: Odczyty dysku/s* udostępniają informacje o liczbie operacji wejścia-wyjścia na sekundę w badanym przedziale czasu. Te dane pozwalają ustalić, czy podsystem wejścia-wyjścia zbliża się do granic swoich możliwości. Informacje te same



Rysunek 12.5. Sprzętowa macierz RAID

w sobie można porównać z teoretyczną wydajnością podsystemu wejścia-wyjścia oszacowaną na liczbie i rodzajach używanych dysków. Ponadto uzyskane dane można porównać z punktem odniesienia opisującym podsystem wejścia-wyjścia i określić, jak blisko maksymalnej wydajności jest system.

Monitorowanie przepustowości w megabajtach na sekundę

Liczniki *Dysk fizyczny: Bajty zapisu dysku/s* i *Dysk fizyczny: Bajty odczytu dysku/s* informują, ile megabajtów na sekundę jest wczytywanych z dysku i zapisywanych na nim w badanym przedziale czasu. Wyniki są uśredniane dla tego przedziału; jeśli jest długi, bardzo wysokie i bardzo niskie wartości zostaną ukryte. Gdy przedział jest krótki, wyniki mogą się w znacznym stopniu zmieniać, jeśli np. rejestrowane są skutki wykonywania jednej lub dwóch dużych operacji wejścia-wyjścia w podsystemie. Informacje z tych liczników są przydatne przy ustalaniu, czy podsystem wejścia-wyjścia zbliża się do granic swoich możliwości.

Liczniki *Bajty zapisu dysku/s* i *Bajty odczytu dysku/s* (podobnie jak inne liczniki dyskowych operacji wejścia-wyjścia) można stosować w izolacji do porównań z teoretyczną przepustowością oszacowaną na podstawie liczby i rodzajów dysków w podsystemie wejścia-wyjścia. Jednak bardziej przydatne są, gdy porównuje się je z punktem odniesienia określającym maksymalną przepustowość tego podsystemu.

Monitorowanie opóźnień operacji wejścia-wyjścia

Liczniki *Dysk fizyczny: Średni czas dysku w s/Zapis* i *Dysk fizyczny: Średni czas dysku w s/Odczyt* zwracają informacje o długości operacji zapisu i odczytu. Te dwa liczniki wyświetlają średnie opóźnienie. Jest ono obliczane dla wszystkich operacji wejścia-wyjścia z analizowanego okresu.

Informacje są bardzo przydatne i można je stosować niezależnie od innych danych, aby ustalić, jak dobrze podsystem wejścia-wyjścia radzi sobie z bieżącym obciążeniem. W idealnych warunkach wartości tych liczników powinny wynosić poniżej 5 – 10 milisekund.

W dużych magazynach danych i systemach wspomagania podejmowania decyzji akceptowalne są wartości z przedziału 10 – 20 milisekund. Wartości utrzymujące się na poziomie powyżej 50 milisekund są oznaką wysokiego obciążenia podsystemu wejścia-wyjścia. Warto wtedy przeprowadzić szczegółowe analizy.

Liczniki te informują o spadku wydajności, zanim jeszcze powstaną kolejki. Należy je stosować razem z licznikami długości kolejki przy diagnozowaniu zatorów w podsystemie wejścia-wyjścia.

Monitorowanie długości kolejek wejścia-wyjścia

Liczniki *Dysk fizyczny: Średnia długość kolejki zapisu dysku* i *Dysk fizyczny: Średnia długość kolejki odczytu dysku* informują o długości kolejek zapisu i odczytu. Te dwa liczniki wyświetlają średnią długość kolejki z badanego okresu. Jeśli ta wartość dla jednego dysku fizycznego przekracza 2, wskazuje na zator w podsystemie wejścia-wyjścia.

Poprawne interpretowanie wartości tych liczników staje się trudniejsze, gdy w podsystemie wejścia-wyjścia używana jest macierz RAID lub gdy kontroler dysku ma wbudowaną pamięć podręczną i mechanizmy decyzyjne. W takich scenariuszach kontroler używa własnej kolejki, która stanowi bufor dla kolejek z poziomu dysku i ukrywa je przed licznikiem. Dlatego monitorowanie tych liczników jest mniej przydatne niż monitorowanie liczników opóźnienia. Jeśli jednak liczniki informują o tym, że długość kolejek stale przekracza 2, wskazuje to na potencjalny zator w podsystemie wejścia-wyjścia.

Monitorowanie poszczególnych dysków i łącznych wartości

W systemach z wieloma dyskami monitorowanie wszystkich wymienionych liczników dla każdego dysku jest trudne. Dlatego w niektórych sytuacjach przydatnym sposobem wykrywania problemów w podsystemie wejścia-wyjścia jest monitorowanie liczników sumarycznych (łączą one wartości dla wszystkich dysków). Technika ta nie sprawdza się jednak wtedy, gdy operacje wejścia-wyjścia na poszczególnych dyskach mają inne cechy. Wtedy liczniki sumaryczne mogą zwracać akceptowalną uśrednioną wartość, choć niektóre dyski są bezczynne, a inne nie potrafią obsłużyć wszystkich żądań wejścia-wyjścia.

Monitorowanie transferu oraz odczytu i zapisu

Zauważyłeś, że w omówieniu liczników zabrakło liczników dotyczących transferu? Powodem jest to, że liczniki transferu uśredniają wartości dla odczytu i zapisu. W systemach, w których wykonywanych jest wiele odczytów i niewiele zapisów (lub na odwrót), liczniki transferu nie wyświetlają precyzyjnych informacji na temat obciążenia dysku.

Ponadto operacje odczytu i zapisu mają zwykle inne cechy oraz inną wydajność. Dlatego średnia z dwóch potencjalnie różnych wartości nie jest przydatnym wskaźnikiem.

Monitorowanie liczników z grupy ... dysku (%)

Inną pominiętą grupą liczników są liczniki ... *dysku (%)*. Choć czasem udostępniają one przydatne informacje, nieraz są problematyczne (np. ich suma może przekraczać 100%) i nie stanowią precyzyjnych wskaźników.

Jeśli możesz monitorować wszystkie wskaźniki opisane we wcześniejszych podpunktach, uzyskasz kompletny obraz tego, co dzieje się w systemie.

Jeżeli jednak szukasz kilku prostych wskaźników, które pozwalają oszacować ogólną aktywność podsystemu wejścia-wyjścia, możesz zastosować liczniki *Czas dysku (%)*, *Czas odczytu dysku (%)* i *Czas zapisu dysku (%)*.

Wyodrębnienie aktywności dysku wygenerowanej przez system SQL Server

Wcześniej omówione zostały wszystkie liczniki, które należy monitorować w celu wyszukiwania zatorów na dysku. Możliwe jednak, że na serwerze działa wiele aplikacji, a jedna z nich generuje dużo dyskowych operacji wejścia-wyjścia. Aby mieć pewność, że to system SQL Server jest źródłem zatoru, należy wyodrębnić aktywność dysku związaną z tym systemem. Umożliwiają to następujące liczniki:

- *SQLServer: Buffer Manager: Page reads/sec*,
- *SQLServer: Buffer Manager: Page writes/sec*.

Czasem aplikacja ma za duże wymagania sprzętowe i problem, który na pozór związany jest z dyskowymi operacjami wejścia-wyjścia, można rozwiązać, dodając pamięć RAM. Przed podjęciem jakichkolwiek decyzji przeprowadź niezbędne analizy. Przydatne są tu analizy trendów, ponieważ pozwalają zobaczyć zmiany w natężeniu problemu.

Czy to niska wydajność dysku jest źródłem zatoru?

Za pomocą opisanych liczników można ustalić, czy w systemie występują zatory związane z dyskiem. Aby stwierdzić wystąpienie takich zatorów, należy wykryć kilka czynników: trwałą aktywność dysku przekraczającą punkt odniesienia, długość kolejki stałe przekraczającą dwa elementy na dysk i brak nasilonego stronicowania. Jeśli taka kombinacja czynników nie występuje, w systemie prawdopodobnie nie ma zatorów związanych z dyskiem.

Czasem sam dysk jest uszkodzony i generuje dużo przerw procesora. Możliwe też, że źródłem zatoru w procesorze jest podsystem dyskowy (może mieć on wpływ na wydajność całego systemu). Koniecznie uwzględnij takie możliwości w trakcie analizowania danych o wydajności.

UWAGA Jeśli w wyniku monitorowania systemu dojdiesz do wniosku, że na poziomie dysku występują zatory, powinieneś rozwiązać ten problem. Rozdział 10. zawiera szczegółowe informacje o konfigurowaniu systemu SQL Server pod kątem optymalnej wydajności, rozdział 11. jest poświęcony optymalizowaniu systemu SQL Server, a rozdział 13., „Dostrajanie wydajności kodu w języku T-SQL”, dotyczy dopracowywania kwerend w języku SQL.

Użytkowanie pamięci

Pamięć to prawdopodobnie najważniejszy zasób wpływający na wydajność systemu SQL Server. Jeśli pamięci jest za mało, system SQL Server przy wykonywaniu kwerend musi wczytywać i zapisywać dane na dysku. Dostęp do dysku jest od 1000 do 100 000 razy wolniejszy niż używanie pamięci (zależy to od szybkości tych komponentów).

Zapewnienie systemowi SQL Server wystarczającej ilości pamięci to jeden z najważniejszych kroków pozwalających na zapewnienie jego szybkiego działania. Niezwykle ważne jest monitorowanie użytkowania pamięci, jej ilości i tego, czy system SQL Server właściwie z niej korzysta.

W idealnym środowisku system SQL Server działa na dedykowanym komputerze i współużytkuje pamięć tylko z systemem operacyjnym oraz innymi niezbędnymi aplikacjami. Jednak w wielu środowiskach ograniczenia (budżetowe i nie tylko) sprawiają, że system SQL Server musi działać na serwerze razem z innymi programami. Wtedy trzeba monitorować, ile pamięci wykorzystuje każda aplikacja, i sprawdzać, czy wszystkie programy poprawnie współdzielają.

Mała ilość pamięci może spowolnić pracę aplikacji i usług. Należy okresowo monitorować system SQL Server i sprawdzać, czy wartości liczników dotyczących pamięci są typowe. Gdy serwerowi brakuje pamięci, może nastąpić nasilone *stronicowanie* (czyli proces przenoszenia pamięci wirtualnej tam i z powrotem między pamięcią fizyczną a dyskiem), co dodatkowo obciąża dysk. Stronicowanie współzawodniczy wtedy o zasoby z wykonywanymi transakcjami, co nasila zatory na dysku.

SQL Server to jedna z „najgrzeczniejszych” aplikacji serwerowych. Po zgłoszeniu małej ilości pamięci przez system operacyjny SQL Server zwalnia pamięć, aby udostępnić ją innym aplikacjom. Jeśli na komputerze działają aplikacje o dużym zapotrzebowaniu na pamięć, może to doprowadzić do braku pamięci dla systemu SQL Server.

Dobra wiadomość jest taka, że SQL Server zwalnia za każdym razem tylko niewielką ilość pamięci. Dlatego zanim w systemie SQL Server pojawią się problemy, może minąć kilka godzin, a nawet dni. Niestety, jeśli inny program nagle potrzebuje dużej ilości pamięci, SQL Server zwolni ją po paru godzinach. Dopiero po tym czasie inny program będzie mógł zacząć działać bez nasilonego stronicowania. Ponieważ takie sytuacje mogą prowadzić do poważnych problemów, należy monitorować liczniki opisane w następnych podpunktach, aby identyfikować zatory w pamięci.

Monitorowanie ilości dostępnej pamięci

Licznik *Pamięć: Dostępna pamięć (Mb)* informuje, ile megabajtów pamięci jest obecnie dostępnych dla programów. Jest to najlepszy wskaźnik zatorów w pamięci serwera.

Odpowiednia wartość tego licznika zależy od wielkości monitorowanego systemu. Jeśli jednak ten licznik regularnie pokazuje wartość niższą niż 128 megabajtów, oznacza to poważny brak pamięci.

Na serwerze z 4 gigabajtami (lub więcej) fizycznej pamięci RAM system operacyjny wysyła powiadomienie o niskiej ilości pamięci, gdy ilość dostępnej pamięci spada poniżej 128 megabajtów. Przy tym poziomie system SQL Server zwalnia część zajętej pamięci, aby udostępnić ją innym procesom.

Staraj się, aby w systemie było przynajmniej 256 – 200 megabajtów dostępnej pamięci. W większych systemach, mających ponad 16 gigabajtów pamięci RAM, ilość wolnej pamięci należy podnieść do 500 megabajtów – 1 gigabajta. Na serwerach o ponad 64 gigabajtach pamięci RAM ilość dostępnej pamięci powinna wynosić 1 – 2 gigabajty.

Monitorowanie użytkowania pamięci przez proces systemu SQL Server

Gdy już za pomocą licznika *Pamięć: Dostępna pamięć (Mb)* ustaliłeś, że w systemie brakuje pamięci, następnym krokiem jest określenie, które procesy zajmują dostępną pamięć. Ponieważ dla Ciebie ważny jest system SQL Server, można mieć nadzieję, że to on korzysta z pamięci. Zawsze jednak należy się upewnić.

Do sprawdzania wykorzystania pamięci standardowo używany jest obiekt *Proces* dla odpowiedniego procesu. Liczniki istotne dla systemu SQL Server zostały opisane poniżej.

- **Obiekt: Proces, egzemplarz: sqlservr, licznik: Bajty wirtualne.** Licznik określa wielkość wirtualnej przestrzeni adresowej zaalokowanej dla danego procesu. Wirtualna przestrzeń adresowa jest wykorzystywana przez liczne procesy, które nie wpływają na wydajność pamięci. Ten licznik jest przydatny tylko przy szukaniu źródeł błędów braku pamięci w systemie SQL Server.
- **Obiekt: Proces, egzemplarz: sqlservr, licznik: Zestaw roboczy.** Licznik określa wielkość zestawu roboczego dla procesu systemu SQL Server. Zestaw roboczy obejmuje cały zbiór stron przechowywany obecnie w pamięci (nieprzeniesiony na dysk w wyniku stronicowania). Jeśli ta wartość jest wyraźnie niższa niż wartość licznika *Bajty prywatne*, może to wskazywać na duże obciążenie pamięci.
- **Obiekt: Proces, egzemplarz: sqlservr, licznik: Bajty prywatne.** Licznik informuje, ile bajtów pamięci zajętych przez dany proces nie można współużytkować z innymi procesami (czyli ile bajtów jest prywatnych dla procesu). Aby zrozumieć różnicę między tym licznikiem a licznikiem *Bajty wirtualne*, zauważ, że niektóre pliki wczytywane do obszaru pamięci procesu (pliki *.exe*, *.dll* i pliki odwzorowane w pamięci) są automatycznie współużytkowane w systemie operacyjnym. Dlatego licznik *Bajty prywatne* określa, ile pamięci proces potrzebuje na stosy, sterty i inne struktury zaalokowane w pamięci wirtualnej. Należy porównać tę wartość z łączną ilością zajmowaną przez system. Gdy wartość ta stanowi dużą część całej pamięci systemowej, oznacza to, że system SQL Server jest przyczyną braku pamięci na serwerze.

Inne liczniki pamięci dotyczące systemu SQL Server

Poniżej znajduje się lista dodatkowych liczników pamięci związanych z systemem SQL Server. Gdy szukasz źródeł problemów z pamięcią, te liczniki zapewnią bardziej szczegółowe informacje niż liczniki przedstawione wcześniej.

- **SQLServer: Buffer Manager: Buffer Cache Hit Ratio.** Licznik informuje, ile żądanych stron znaleziono w puli buforów. Wartości od 98% w górę to dobry wynik, jednak już 97,9% może oznaczać problemy z pamięcią.
- **SQLServer: Buffer Manager: Free Pages.** Licznik określa, ile wolnych stron ma system SQL Server na potrzeby obsługi żądań nowej strony. To, jakie wartości tego licznika są akceptowalne, zależy od ilości dostępnej pamięci i profilu użytkowania pamięci przez aplikacje. Przydatny jest dobry punkt odniesienia, ponieważ porównanie z nim wartości licznika pozwala ustalić, czy dany poziom oznacza problemy z pamięcią, czy jest typowy dla systemu.

Ten licznik należy interpretować razem z licznikiem *SQLServer: Buffer Manager: Page Life Expectancy*.

- **SQLServer: Buffer Manager: Page Life Expectancy.** Licznik informuje o liczbie sekund, przez jakie szacunkowo strony pozostają w puli buforów przed ich przeniesieniem na dysk. Zgodnie z opracowanymi przez Microsoft najlepszymi praktykami akceptowalne są wartości powyżej 300. Jeśli wartość tylko w niewielkim stopniu przekracza ten poziom, stanowi to powód do obaw. Spadek wartości poniżej 300 oznacza brak pamięci. Jednak obecnie wspomniane najlepsze praktyki mogą okazać się nieaktualne. Pisano je wtedy, gdy w dużych systemach używano 4 procesorów 2- lub 4-rdzeniowych i 16 gigabajtów pamięci RAM. Obecnie w sprzedaży dostępne są systemy dwuprocesorowe z 10, 12, 16 i więcej rdzeniami oraz możliwością zainstalowania 256 gigabajtów pamięci (lub więcej). Dlatego wartości proponowane we wspomnianych najlepszych praktykach są tylko wskazówką. Powinienes więc interpretować wartość tego licznika w kontekście innych liczników. Pomoże to zrozumieć obciążenie pamięci.

Ten licznik można interpretować razem z licznikiem *Free Pages*. Gdy wartość licznika *Page Life Expectancy* spadnie poniżej 300, powinieneś zobaczyć znaczny spadek wartości licznika *Free Pages*. Te dwa liczniki pozwalają stwierdzić, czy obciążenie pamięci jest na tyle duże, że może spowodować zator.

Wykorzystanie liczników pamięci w skryptach narzędzia Logman

W listingu 12.1 przedstawiono skrypt narzędzia Logman (plik *logman_create_memory.cmd*), który tworzy dziennik z danymi z opisanych wcześniej liczników pamięci. Więcej informacji o tym narzędziu znajdziesz w podpunkcie „Logman”, dalej w tym rozdziale.

Listing 12.1. Plik *logman_create_memory.cmd*

```
Logman create counter "Memory Counters" -si05 -v nnnnnn -o
"c:\perflogs\Memory Counters" -c "\Memory\Available MBytes"
"\Process(sqlservr)\Virtual Bytes" "\Process(sqlservr)\Working Set"
"\Process(sqlservr)\Private Bytes" "\SQLServer:Buffer Manager\Database
pages" "\SQLServer:Buffer Manager\Target pages" "\SQLServer:Buffer
Manager\Total pages" "\SQLServer:Memory Manager\Target Server Memory (KB)"
"\SQLServer:Memory Manager\Total Server Memory (KB)"
```

Radzenie sobie z zatorami w pamięci

Łatwym sposobem na rozwiązywanie zatorów w pamięci jest zwiększenie jej ilości. Jednak, jak wcześniej wspomniano, zawsze należy zacząć od dopracowania aplikacji. Postaraj się znaleźć kwerendy zajmujące dużo pamięci (mogą to być kwerendy z dużymi obszarami roboczymi, np. wykonujące złączenia lub sortowanie z haszowaniem) i sprawdź, czy można je usprawnić. Więcej informacji na temat dostrajania kwerend w języku T-SQL podajemy w rozdziale 13., „Dostrajanie wydajności kodu w języku T-SQL”.

Zapoznaj się też z rozdziałem 10., jeśli jeszcze tego nie zrobiłeś, aby się upewnić, że prawidłowo skonfigurowałeś serwer.

Narzędzia do monitorowania wydajności

W systemie operacyjnym Windows od pewnego czasu ukrytych jest kilka narzędzi uruchamianych z poziomu wiersza poleceń. Logman i Relog to dwa takie narzędzia, które są niezwykle przydatne podczas korzystania z monitora wydajności.

Logman

Logman umożliwia tworzenie skryptów wiersza poleceń sterujących dziennikami liczników monitorowania wydajności. Przy użyciu tego narzędzia można tworzyć, modyfikować, uruchamiać i zatrzymywać takie dzienniki.

Wcześniej w tym rozdziale zapoznałeś się już z przykładem pokazującym, jak za pomocą narzędzia Logman utworzyć dziennik liczników. W listingu 12.2 przedstawiamy krótki plik skryptu wiersza poleceń (plik *logman_start_memory.cmd*) uruchamiający i zatrzymujący grupę liczników.

Listing 12.2. Plik logman_start_memory.cmd

```

REM Uruchamianie grupy liczników.
logman start "Memory Counters"
timeout /t 5
REM Ustawienie krótkiej przerwy,
REM aby grupa liczników mogła zacząć
REM rejestrować przydatne informacje.
REM Następnie grupa liczników jest zatrzymywana.
logman stop "Memory Counters"
timeout /t 5
REM Odczekanie 5 sekund, aby mieć pewność, że grupa została zatrzymana.

```

W przeszłości w pomocy systemu Windows znajdowała się pełna dokumentacja narzędzia Logman. W systemie Windows 8 jest jednak niedostępna. Teraz z dokumentacją możesz się zapoznać w internecie. W tym celu wywołaj polecenie `logman /?`, wpisz wyrażenie „Logman” w wyszukiwarce Bing lub otwórz stronę <http://technet.microsoft.com/en-us/library/cc753820.aspx>.

Skrypt narzędzia Logman korzystający z liczników wejścia-wyjścia

W listingu 12.3 przedstawiono skrypt z pliku `logman_create_io.cmd`. Tworzy on nowy dziennik liczników, `IO Counters`, rejestruje dane z wszystkich opisanych wcześniej liczników (dla każdego egzemplarza systemu) w 5-sekundowych przedziałach czasu i zapisuje zebrane informacje w plikach `c:\perflogs\IO Counters...` (na końcu dodawana jest 6-cyfrowa liczba porządkowa).

Listing 12.3. Plik logman_create_io.cmd

```

Logman create counter "IO Counters" -si 05 -v nnnnnn -o "c:\perflogs\IO
Counters" -c "\PhysicalDisk(*)\Avg. Disk Bytes/Read" " \PhysicalDisk(*)\Avg.
Disk Bytes/Write" "\PhysicalDisk(*)\Avg. Disk Read Queue Length"
"\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write"
"\PhysicalDisk(*)\Avg. Disk Write Queue Length" "\PhysicalDisk(*)\Disk Read
Bytes/sec" "\PhysicalDisk(*)\Disk Reads/sec" "\PhysicalDisk(*)\Disk Write
Bytes/sec" "\PhysicalDisk(*)\Disk Writes/sec"

```

Po uruchomieniu tego skryptu wywołaj poniższe polecenie, aby się upewnić, że ustawienia dają oczekiwany efekt:

```
logman query "IO Counters"
```

Relog

Relog to uruchamiane w wierszu poleceń narzędzie umożliwiające odczyt pliku dziennika i zapis jego wybranych fragmentów do nowego pliku dziennika.

Za pomocą tego narzędzia można zmienić format pliku z `blg` na `csv`, zmodyfikować próbkowanie danych i przekształcić duży plik z próbkami dla krótkich okresów na mniejszy plik z próbkami dla dłuższych okresów, a także pobrać z dużego pliku dane z krótkiego okresu dla wybranych liczników.

W przeszłości w pomocy systemu Windows znajdowała się pełna dokumentacja narzędzia Relog. W systemie Windows 8 jest jednak niedostępna. Teraz z dokumentacją możesz się zapoznać w internecie. W tym celu wywołaj polecenie `relog /?`, wpisz wyrażenie „Relog” w wyszukiwarce Bing lub otwórz stronę <http://technet.microsoft.com/en-us/library/cc771669.aspx>.

Monitorowanie zdarzeń

Zdarzenia są zgłaszane wtedy, gdy w systemie SQL Server stanie się coś ważnego. Dzięki zdarzeniom można reagować na sytuację od razu — nie trzeba czekać. System SQL Server generuje wiele różnych zdarzeń i udostępnia szereg narzędzi do monitorowania wybranych z nich.

Na poniższej liście opisujemy różne mechanizmy, które można wykorzystać do monitorowania zdarzeń zachodzących w silniku bazodanowym.

- **Zdarzenia rozszerzone.** Zostały wprowadzone w systemie SQL Server 2008. Stanowią rozszerzenie mechanizmu powiadamiania o zdarzeniach i są wbudowane w platformę ETW (ang. *Event Tracing for Windows*). Zdarzenia rozszerzone to odrębna grupa, niepowiązana ze zdarzeniami używanymi w powiadomieniach. Można je wykorzystać do wykrywania małej ilości pamięci, wysokiego obciążenia procesora i zakleszczeń. Dzienniki tworzone na podstawie zdarzeń rozszerzonych systemu SQL Server można łączyć z innymi dziennikami platformy ETW przy użyciu narzędzia *tracertpt.exe*.

UWAGA Więcej odwołań do źródeł informacji o używaniu platformy ETW i narzędzia *tracertpt.exe* znajdziesz w artykule „Extended Events” w dokumentacji Books Online systemu SQL Server. Szczegółowe omówienie zdarzeń rozszerzonych zamieszczono w punkcie „Zdarzenia rozszerzone systemu SQL Server”, dalej w tym rozdziale.

- **Sesja system_health.** Sesja system_health jest domyślnie uruchamiana dla systemu SQL Server. Rozpoczyna pracę w momencie rozruchu systemu SQL Server i działa bez odczuwalnego wpływu na wydajność. Ta wersja rejestruje niewielki zbiór informacji systemowych pomocny przy rozwiązywaniu problemów z wydajnością.
- **Domyślny ślad.** Mechanizm został wprowadzony w wersji SQL Server 2005 i jest jednym z najlepiej strzeżonych sekretów dotyczących systemu SQL Server. Znalezienie dokumentacji tego mechanizmu jest niemożliwe. Domyślny ślad pełni funkcję „rejestratora parametrów lotu” dla systemu SQL Server. Rejestruje ostatnich 5 megabajtów danych o najważniejszych zdarzeniach. Wybierane są zdarzenia, które zajmują mało miejsca, a przy tym są przydatne przy rozwiązywaniu krytycznych problemów z systemem SQL Server.
- **SQL Trace.** Mechanizm rejestruje określone zdarzenia i zapisuje je w plikach, które można później wykorzystać do analizy danych. Wymaga określenia, które zdarzenia silnika bazodanowego mają być śledzone. Dostęp do danych ze śladu można uzyskać dwoma sposobami:
 - za pomocą narzędzia SQL Server Profiler (jest to aplikacja z interfejsem graficznym),
 - przy użyciu systemowych procedur składowanych w języku T-SQL.
- **SQL Server Profiler.** Narzędzie wykorzystuje wszystkie możliwości rejestrowania zdarzeń dostępne w mechanizmie SQL Trace, a ponadto umożliwia zapisywanie (w postaci odrębnych plików XML) śladu w tabeli, pobieranie śladu z tabeli, zapisywanie szablonów śledzenia, zapisywanie zdarzeń związanych z planami wykonania kwerend i zakleszczeniami, a także odtwarzanie wyników śledzenia na potrzeby diagnozowania i optymalizowania systemu. Inna (mniej znana) możliwość to zapisywanie śladu w tabeli bazy danych. Zapisanie pliku śladu w tabeli bazy danych pozwala wykorzystać kwerendy w języku T-SQL do przeprowadzania złożonych analiz dotyczących zdarzeń ze śladu.

- **Powiadomienia o zdarzeniach.** Mechanizm przesyła do usługi Service Broker informacje o różnych zdarzeniach generowanych przez system SQL Server. Powiadomienia o zdarzeniach (w odróżnieniu od śladu) można wykorzystać do wykonywania w systemie SQL Server operacji w reakcji na zdarzenia. Ponieważ powiadomienia o zdarzeniach działają asynchronicznie, wykonywane operacje nie zajmują zasobów bieżącej transakcji. Jeśli np. chcesz otrzymywać powiadomienia o modyfikacji tabeli w bazie danych, instrukcja `ALTER TABLE` nie będzie zużywać więcej zasobów lub działać wolniej z powodu zdefiniowania powiadomienia o tym zdarzeniu.

Oto kilka powodów, dla których należy monitorować zdarzenia zachodzące w systemie SQL Server.

- **Wyszukiwanie najmniej wydajnych kwerend i procedur składowanych.** Można do tego wykorzystać zdarzenia rozszerzone albo narzędzie SQL Server Profiler lub SQL Trace. Po wykryciu kwerend o najniższej wydajności trzeba ustalić, jaki plan ich wykonania został wygenerowany. Samo sprawdzenie czasu wykonywania operacji wsadowej lub procedury składowanej w języku T-SQL nic nie da. Możesz przefiltrować dane ze śladu. W tym celu ustaw odpowiednią wartość w kolumnie *Duration*, aby pobrać tylko zdarzenia trwające dłużej od podanego czasu. Pozwala to ograniczyć zbiór analizowanych danych.
- **Inspekcja działań użytkowników.** Możesz przeprowadzić inspekcję za pomocą funkcji SQL Audit i zdarzeń rozszerzonych lub wygenerować ślad za pomocą zdarzeń z klasy Audit Login. Jeśli używasz drugiej możliwości, wybierz kolumny EventClass (domyślna), EventSubClass, LoginSID i LoginName. Dzięki temu będziesz mógł badać działania użytkowników w systemie SQL Server. Jeśli trzeba, możesz też dodać więcej zdarzeń z kategorii Security Audit lub kolumn z danymi. Możliwe, że któregoś dnia zarejestrowane informacje przydadzą się nie tylko do celów technicznych, ale też w kwestiach prawnych.
- **Identyfikowanie przyczyn zakleszczenia.** Do tego można wykorzystać zdarzenia rozszerzone. Dużo potrzebnych informacji jest dostępnych w sesji system_health, która jest domyślnie uruchamiana dla każdego egzemplarza systemu SQL Server. Więcej informacji dotyczących tej sesji znajdziesz dalej w tym rozdziale.
- **Zbieranie reprezentatywnego zbioru zdarzeń dla testów obciążeniowych.** Przy przeprowadzaniu testów porównawczych warto odtwarzać wygenerowany ślad. System SQL Server udostępnia standardowy szablon TSQL_Replay służący do rejestrowania śladu, który można później odtworzyć. Jeśli chcesz wykorzystać ślad do odtwarzania, upewnij się, że korzystasz z tego standardowego szablonu, ponieważ system SQL Server wymaga zarejestrowania konkretnych zdarzeń, co dany szablon zapewnia. Dalej w tym rozdziale zobaczysz, jak odtworzyć ślad.
- **Tworzenie obciążenia do użytku w narzędziu Database Engine Tuning Advisor.** SQL Server Profiler udostępnia wbudowany szablon Tuning, który rejestruje odpowiednie zdarzenia języka T-SQL w pliku śladu. Później można je wykorzystać jako obciążenie w narzędziu Database Engine Tuning Advisor.
- **Tworzenie punktu odniesienia opisującego wydajność.** Wcześniej dowiedziałeś się, że należy określić punkt odniesienia, a następnie regularnie go aktualizować, aby porównać go z wcześniejszymi punktami odniesienia i ustalić, jak dobrze działa aplikacja. Załóżmy, że proces wsadowy raz dziennie wczytuje dane, sprawdza ich poprawność, przekształca itd., a następnie umieszcza je w hurtowni po usunięciu poprzedniego zbioru danych. Po pewnym czasie ilość danych wzrasta, a proces zaczyna działać coraz wolniej. Domyślasz się, że powodem jest wzrost ilości danych, czy jednak jest to jedyna przyczyna? Możliwe, że nie. Wygenerowany plan wykonania kwerendy może być inny — np. z powodu błędnych

statystyk lub wzrostu ilości danych. Jeśli masz profil statystyk dla planu wykonania kwerendy wygenerowany przy standardowym obciążeniu, za pomocą innych danych (np. dzienników wydajności) szybko zidentyfikujesz źródło problemu.

W następnych punktach szczegółowo omawiamy poszczególne narzędzia do monitorowania zdarzeń.

Domyślny ślad

Domyślny ślad wprowadzono w systemie SQL Server 2005. Ten ślad jest rejestrowany zawsze i obejmuje minimalny zestaw zdarzeń zajmujących niewiele miejsca. Jeśli po zapoznaniu się z działaniem tego mechanizmu stwierdzisz, że nie chcesz go używać, możesz go wyłączyć za pomocą przedstawionego na listingu 12.4 kodu w języku T-SQL (plik *wylaczSladDomysl.sql*).

Listing 12.4. Plik wylaczSladDomysl.sql

```
-- Włączenie zaawansowanych opcji.
exec sp_configure 'show advanced options', '1'
reconfigure with override
go
-- Wyłączenie domyślnego śladu.
exec sp_configure 'default trace enabled', '0'
reconfigure with override
go
-- Wyłączenie zaawansowanych opcji.
exec sp_configure 'show advanced options', '0'
reconfigure with override
go
```

UWAGA Jeśli wyłączysz ślad domyślny, a następnie stwierdzisz, że jednak będzie przydatny i zechcesz ponownie go włączyć, możesz wykorzystać ten sam kod. Tym razem ustaw w procedurze `sp_configure` wartość parametru `'default trace enabled'` na 1, a nie na 0.

Domyślny ślad rejestruje trzydzieści zdarzeń w pięciu plikach śladu działających jak bufor FIFO. Najstarszy plik jest usuwany, aby zwolnić miejsce na nowe zdarzenia w następnym pliku *trc*.

Pliki z domyślnym śladem są przechowywane w katalogu *Logs* systemu SQL Server. W tym katalogu między plikami dzienników błędów znajdziesz pięć plików śladu. Są to standardowe pliki śladu systemu SQL Server, dlatego możesz otworzyć je w narzędziu SQL Server Profiler.

Ważne jest, aby wiedzieć, jakie zdarzenia są rejestrowane w domyślnym śladzie, i pamiętać, by zajrzeć do niego, gdy w systemie SQL Server pojawiają się problemy. Zdarzenia rejestrowane w domyślnym śladzie są podzielone na sześć kategorii. Oto one.

- **Database (czyli baza danych).** Zdarzenia dotyczą powiększania plików danych i dzienników, a także zmian stanu kopii lustrzanej.
- **Errors and Warnings (czyli błędy i ostrzeżenia).** Zdarzenia rejestrują informacje z dziennika błędów i związane z wykonywaniem kwerend ostrzeżenia dotyczące brakujących kolumn, predykatów złączeń, sortowania i haszowania.

- **Full text (czyli wyszukiwanie pełnotekstowe).** Zdarzenia wyświetlają informacje o wyszukiwaniu pełnotekstowym (o jego rozpoczęciu, zakończeniu lub anulowaniu).
- **Objects (czyli obiekty).** Zdarzenia rejestrują informacje o operacjach Create, Delete i Alter dotyczących obiektów User. Jeśli chcesz wiedzieć, kiedy dany obiekt został utworzony, zmodyfikowany lub usunięty, zajrzyj do informacji z tej kategorii.
- **Security Audit (czyli inspekcja zabezpieczeń).** Kategoria dotyczy głównych zdarzeń z zakresu bezpieczeństwa zgłaszanych w systemie SQL Server. Istnieje dość długa lista zdarzeń podrzędnych, których w tym miejscu nie przedstawiamy. Jeśli szukasz informacji związanych z zabezpieczeniami, powinieneś zacząć od tej kategorii.
- **Server (czyli serwer).** Kategoria obejmuje tylko jedno zdarzenie — Server Memory Change. To zdarzenie informuje o zwiększeniu lub zmniejszeniu ilości pamięci zajmowanej przez system SQL Server o 1 megabajt lub o 5% maksymalnej pamięci serwera (w zależności od tego, która z tych wartości jest większa).

Aby wyświetlić te kategorie, otwórz jeden z plików domyślnego śladu w narzędziu SQL Server Profiler i sprawdź właściwości. Domyślnie użytkownicy nie mają uprawnień do otwierania plików śladu przechowywanych w katalogu *Logs*, dlatego skopiuj dany plik w inne miejsce lub zmień uprawnienia do pliku, który chcesz otworzyć, w narzędziu SQL Server Profiler.

Gdy otworzysz właściwości pliku śladu, zobaczysz, że w domyślnym śladzie w każdej kategorii zaznaczone są kolumny odpowiadające wszystkim zdarzeniom.

Sesja system_health

Sesja `system_health` jest domyślną sesją zdarzeń rozszerzonych tworzoną przez system SQL Server. Zużywa ona niewiele zasobów i ma tylko minimalny wpływ na wydajność systemu. Przy stosowaniu narzędzi SQL Server Profiler i domyślnego śladu użytkownicy czasem obawiają się o ich wpływ na wydajność. Stosując zdarzenia rozszerzone i sesję `system_health`, możesz się o to nie martwić.

Sesja `system_health` obejmuje bogate informacje, które mogą pomóc przy diagnozowaniu problemów z systemem SQL Server. Oto lista wybranych informacji rejestrowanych przez tę sesję.

- Tekst instrukcji SQL-a i identyfikator dla sesji, które:
 - mają poziom krytyczności większy lub równy 20,
 - natrafiły na problem związany z pamięcią,
 - oczekiwały na zamki przez przynajmniej 15 sekund,
 - oczekiwały na blokady przez przynajmniej 30 sekund.
- Zakleszczenia.
- Problemy z nieodpowiadającymi wątkami w programie szeregującym.

SQL Trace

Jak wspomniano wcześniej, istnieją dwa sposoby definiowania pracy mechanizmu SQL Trace: za pomocą procedur składowanych w języku T-SQL i przy użyciu narzędzia SQL Server Profiler. W tym punkcie najpierw omawiamy architekturę śladu. Dalej przedstawiamy przykład ilustrujący tworzenie śladu po stronie serwera za pomocą systemowej procedury składowanej w języku T-SQL.

Najpierw jednak należy zapoznać się z podstawowymi pojęciami związanymi ze śladem.

- **Zdarzenie.** Jest to operacja, która została wykonana w egzemplarzu silnika bazodanowego systemu SQL Server. Może to być np. zdarzenie `Audit:Logout`, występujące, gdy użytkownik wylogowuje się z systemu SQL Server.
- **Kolumna danych.** Jest to atrybut zdarzenia. W zdarzeniu `Audit:Logout` jest to np. kolumna `SPID`, określająca identyfikator `SPID` użytkownika, który się wylogował. Inna kolumna, `ApplicationName`, zawiera nazwę aplikacji powiązanej z danym zdarzeniem.

UWAGA Jeśli w systemie SQL Server kolumna w śladzie ma wartość zajmującą więcej niż 1 gigabajt, zgłaszany jest błąd, a nadmiarowe dane są obcinane w danych wyjściowych ze śladem.

- **Filtr.** Określa kryteria ograniczające liczbę zdarzeń rejestrowanych w śladzie. Jeśli np. interesują Cię tylko zdarzenia wygenerowane przez aplikację SQL Server Management Studio — Query, ustaw filtr dla kolumny `ApplicationName` na nazwę tej aplikacji. Wtedy w śladzie znajdą się tylko zdarzenia wygenerowane przez ten program.
- **Szablon.** W narzędziu SQL Server Profiler szablon to plik definiujący klasy zdarzeń i kolumny danych rejestrowane w śladzie. W systemie SQL Server dostępnych jest wiele szablonów domyślnych. Znajdują się one w katalogu `\Program Files\Microsoft SQL Server\110\Tools\Profiler\Templates\Microsoft SQL Server\110`.

UWAGA Więcej pojęć związanych ze śladami znajdziesz w artykule „SQL Trace Terminology” w dokumentacji Books Online.

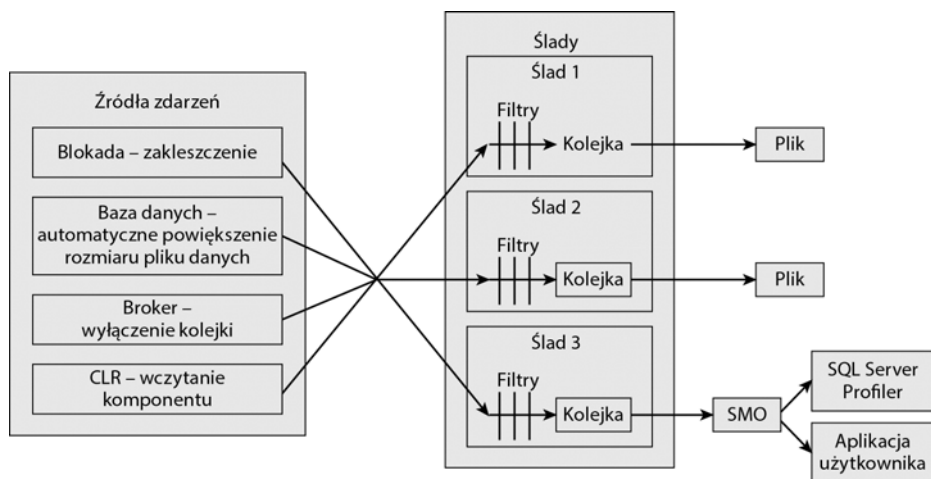
Architektura mechanizmu SQL Trace

Przed przejściem do przykładów powinieneś zapoznać się z działaniem mechanizmu SQL Trace. Na rysunku 12.6 przedstawiamy jego podstawową architekturę. Główną jednostką śledzonych aktywności są zdarzenia. Gdy definiujesz ślad, określasz, które zdarzenia chcesz rejestrować. Jeśli np. interesuje Cię zdarzenie `SP:Starting`, system SQL Server będzie śledził tylko je (wraz z zestawem domyślnych zdarzeń, które zawsze są rejestrowane). Źródłem zdarzenia może być dowolna jednostka: instrukcja w języku T-SQL, zakleszczenie, inne zdarzenia itd.

Jeśli zdarzenie jest wymienione w definicji śladu, wystąpienie tego zdarzenia prowadzi do zarejestrowania informacji na jego temat. Jeśli dla danej klasy zdarzeń w definicji śladu ustawione są filtry (np. użytkownika interesują tylko zdarzenia spełniające warunek `LoginName='foo'`), zostają one uwzględnione, po czym informacje o zdarzeniach trafiają do kolejki. Dane z kolejki są zapisywane w pliku lub mogą zostać wykorzystane przez obiekty SMO (ang. *Server Management Object*) w aplikacjach (przykładowo w narzędziu SQL Server Profiler).

SQL Server Profiler

SQL Server Profiler to rozbudowany interfejs służący do tworzenia śladów, zarządzania nimi, analizowania ich i odtwarzania. SQL Server Profiler pokazuje, jak system SQL Server wewnętrznie przetwarza kwerendy. Pozwala to dokładnie określić, jakie instrukcje języka T-SQL i wielowymiarowe wyrażenia są przekazywane na serwer, a także jak serwer uzyskuje dostęp do bazy lub kostki w celu zwrócenia zbiorów wyników.



Rysunek 12.6. Architektura mechanizmu SQL Trace

UWAGA W systemie SQL Server 2012 narzędzie SQL Server Profiler uznano za przestarzałe. Dlatego powinieneś zastąpić mechanizmy monitorowania wykorzystujące to narzędzie nowszymi rozwiązaniami opartymi na zdarzeniach rozszerzonych.

Za pomocą narzędzia SQL Server Profiler można odczytać plik śladu utworzony przez procedurę składowaną w języku T-SQL. Aby wczytać plik śladu w tym narzędziu, użyj menu *File* i otwórz odpowiedni plik.

UWAGA W systemie SQL Server 2008 serwer informował zarówno o czasie trwania zdarzenia, jak i o czasie używania procesora w milisekundach. W wersji SQL Server 2005 serwer podawał czas trwania zdarzenia w mikrosekundach (to jedna milionowa część sekundy), natomiast czas używania procesora określany był w milisekundach (to jedna tysięczna część sekundy). W wersji SQL Server 2000 obie te wartości były podawane w milisekundach. W systemie SQL Server 2005 interfejs graficzny narzędzia SQL Server Profiler domyślnie wyświetlał wartość kolumny *Duration* w milisekundach, gdy jednak ślad był zapisywany w pliku lub tabeli bazy danych, wartość tej kolumny była podawana w mikrosekundach. Jeśli chcesz wyświetlać w narzędziu SQL Server Profiler wartość tej kolumny w mikrosekundach, wybierz opcję *Tools/Options*, a następnie zaznacz opcję *Show Values in Duration Column in Microseconds* (dotyczy to tylko wersji SQL Server 2005).

W trakcie rozwiązywania problemów bardzo cenna jest możliwość zarejestrowania i zbadania planu wykonania konkretnej kwerendy zapisanego w formacie XML. W narzędziu SQL Server Profiler taki plan można zapisać za pomocą opcji *XML Showplan*. Ponadto powiązanie śladu z tego narzędzia z wykresem z monitora wydajności pomaga diagnozować problemy z wydajnością, ponieważ można połączyć zapisane w śladzie kwerendy z wartościami liczników wydajności zarejestrowanymi w monitorze wydajności. Dzięki temu można dokładnie ustalić, jaki kod był wykonywany, gdy na wykresie w monitorze wydajności zaobserwowano określone wyniki.

UWAGA Gdy przy generowaniu śladu uwzględniana jest klasa zdarzeń Showplan XML, związane z nią koszty negatywnie wpływają na wydajność systemu. Showplan XML powoduje zapisywanie planu wykonania kwerendy generowanego w trakcie jej optymalizowania. Aby zminimalizować koszty, stosuj tę klasę zdarzeń w śladach tylko przy krótkotrwałym monitorowaniu systemu pod kątem konkretnych problemów. Ponadto dodaj dla kolumn danych filtry dostosowane do rejestrowanych danych.

Odtwarzanie śladu

Odtwarzanie polega na ponownym wywoływaniu zdarzeń zapisanych w śladzie. Ten mechanizm pozwala odtworzyć zarejestrowane w śladzie operacje. Gdy stworzysz lub modyfikujesz ślad, możesz go zachować i później odtworzyć. Kiedy za pomocą narzędzia SQL Server Profiler tworzysz ślady, które chcesz potem odtwarzać, zastosuj wbudowany szablon TSQL_Replay. Aby możliwe było odtworzenie śladu w systemie SQL Server, trzeba zarejestrować określone zdarzenia i kolumny danych. Jeśli je pominiesz, system nie odtworzy śladu.

Odtwarzanie śladu jest pomocne przy debugowaniu. Można wtedy zastosować opcje *Toggle Breakpoint* i *Run to Cursor* z menu *Replay* narzędzia SQL Server Profiler. Opcje ułatwiają przede wszystkim analizowanie długich skryptów, ponieważ pozwalają podzielić odtwarzany ślad na krótsze segmenty, które można po kolei sprawdzać.

W trakcie odtwarzania śladu ignorowane są następujące rodzaje zdarzeń.

- **Ślady obejmujące replikację transakcyjną i inne operacje z dziennika transakcji.** Tego rodzaju zdarzenia są pomijane. Inne typy replikacji nie dodają wpisów w dzienniku transakcji, dlatego nie są ignorowane.
- **Ślady obejmujące operacje z identyfikatorami GUID.** Te zdarzenia są pomijane.
- **Ślady z operacjami na kolumnach typów text, ntext i image wykonywanymi za pomocą narzędzia bcp, instrukcje BULK INSERT, READTEXT, WRITETEXT i UPDATETEXT, a także operacje pełnotekstowe.** Te zdarzenia są pomijane.
- **Ślady obejmujące wiązanie sesji (czyli wywołania systemowych procedur składowanych sp_getbindtoken i sp_bindsession).** Te zdarzenia są pomijane.
- **Ślady zarejestrowane przez system SQL Server 7.0 i jego starsze wersje.** SQL Server Profiler nie obsługuje odtwarzania takich śladów.

Ponadto aby odtworzyć ślad na docelowym serwerze, trzeba spełnić określone wymagania.

- Wszystkie loginy (i wszyscy użytkownicy) ze śladu muszą być utworzone na docelowym serwerze i znajdować się w tych samych bazach danych, co na serwerze źródłowym.
- Wszystkie loginy (i wszyscy użytkownicy) na serwerze docelowym muszą mieć te same uprawnienia, co na serwerze źródłowym.
- Wszystkie hasła loginów muszą być takie same jak hasło użytkownika uruchamiającego odtwarzanie. Możesz zastosować zadanie Transfer Login z usługi SSIS, aby przenieść loginy na docelowy serwer, w którym chcesz odtwarzać ślad.
- W idealnych warunkach identyfikatory baz danych na docelowym serwerze powinny być takie same jak na serwerze źródłowym. Jeśli używane są odmienne identyfikatory, można dopasować bazy z obu serwerów na podstawie ich nazw, jeżeli są podane w śladzie. Dlatego zadbaj o to, aby w śladzie używana była kolumna danych DatabaseName.
- Na docelowym serwerze domyślna baza danych dla loginu powinna być taka sama jak na serwerze źródłowym.

- Odtwarzanie zdarzeń powiązanych z nieistniejącymi lub błędnymi loginami prowadzi do błędów, ale cały proces jest wtedy kontynuowany.

Narzędzie Distributed Replay

W systemie SQL Server 2012 wprowadzono narzędzie Distributed Replay, przeznaczone do odtwarzania śladów z różnych komputerów. Choć także SQL Server Profiler potrafi odtwarzać ślad, może to robić tylko dla jednego komputera. Distributed Replay może odtwarzać ślady dla grup komputerów. Dlatego jest to bardziej skalowalne rozwiązanie niż SQL Server Profiler i lepiej nadaje się do symulowania operacji o znaczeniu strategicznym.

Różnice między narzędziami SQL Server Profiler i Distributed Replay

Dostępne są dwa narzędzia do odtwarzania śladów. Którego z nich należy używać? Zwykle powinienś stosować narzędzie SQL Server Profiler i odtwarzać ślady z wykorzystaniem usług Analysis Services. Narzędzie Distributed Replay stosuj tylko wtedy, kiedy współbieżność w zarejestrowanym śladzie jest tak wysoka, że jeden serwer nie zdoła realistycznie zasymulować obciążenia, jakiemu ma zostać poddany serwer docelowy.

System Distributed Replay (nazywany też *Distributed Replay Utility*) obejmuje grupę różnych serwerów, narzędzie administracyjne, kontroler, grupę klientów i docelowy system SQL Server.

Ponieważ Distributed Replay może odtwarzać ślad z różnych serwerów, musisz odpowiednio przygotować plik śladu, zanim będzie można go wykorzystać w tym narzędziu. Przede wszystkim należy wstępnie przetworzyć plik śladu i podzielić go na zestaw strumieni instrukcji odtwarzanych na różnych serwerach klienckich w systemie Distributed Replay.

Wpływ rejestrowania śladu na wydajność

Śledzenie w systemie SQL Server nie powoduje kosztów do momentu, gdy rejestrowane są zdarzenia. Jednak nawet koszty rejestrowania dla większości zdarzeń zwykle są niewielkie. Działanie narzędzia SQL Server Profiler może stać się kosztowne po dodaniu wielu zdarzeń i dużej ilości danych dla każdego z nich. Zwykle maksymalne dodatkowe obciążenie wynosi od 10 do 20%. Jeśli zauważysz przekroczenie tego poziomu lub dojście do niego w systemie produkcyjnym, możesz zmniejszyć liczbę zdarzeń, ograniczyć zakres danych lub zastosować inny mechanizm. Duża część kosztów związana jest z dłuższymi ścieżkami wykonywania kodu — rejestrowanie danych o zdarzeniach nie powoduje dużego obciążenia procesora. Aby zminimalizować spadek wydajności, powinienś używać tylko śladów rejestrowanych po stronie serwera. Pozwala to uniknąć kosztów generowania zbiorów wierszy wysyłanych do klienta narzędzia SQL Server Profiler.

Powiadomienia o zdarzeniach

Powiadomienia o zdarzeniach to specjalne obiekty bazy danych, które przesyłają do Service Brokera (szczegółowy opis tego narzędzia podajemy w rozdziale 6., „Service Broker w systemie SQL Server 2014”) informacje dotyczące zdarzeń serwera lub bazy danych. Powiadomienia można ustawić dla niektórych zdarzeń rejestrowanych przez mechanizm SQL Trace, ale nie dla wszystkich. Możliwe jest też tworzenie powiadomień dla wielu zdarzeń z bibliotek DDL. Powiadomienia o zdarzeniach (w odróżnieniu od śladów) można też wykorzystać do uruchamiania operacji w danym egzemplarzu systemu SQL Server w reakcji na konkretne zdarzenie. Dalej w tym

rozdziale znajdziesz przykład ilustrujący tworzenie powiadomień dla wybranych zdarzeń i wykonywanie określonych działań, jeśli będzie trzeba.

Aby zasubskrybować dane zdarzenie, trzeba utworzyć kolejkę Service Brokera, w której znajdują się szczegółowe informacje o tym zdarzeniu. Ponadto kolejka wymaga usługi Service Brokera, aby mogła odbierać komunikaty. Następnie należy utworzyć powiadomienie o zdarzeniu. Wymaga to przygotowania procedury składowanej wykonującej odpowiednie działania i wywoływania jej, gdy komunikat o zdarzeniu trafi do kolejki. W omówieniu przykładu zakładamy, że wiesz, jak działa Service Broker. Jeśli jeszcze nie znasz tego narzędzia, koniecznie przeczytaj rozdział 6, „Service Broker w systemie SQL Server 2014”.

Można też otrzymywać powiadomienia o grupach zdarzeń. Jeśli np. chcesz dostawać komunikaty w momencie utworzenia, zmodyfikowania lub usunięcia tabeli, nie musisz opracowywać trzech odrębnych powiadomień. Możesz użyć grupy zdarzeń DDL_TABLE_EVENTS i utworzyć jedno powiadomienie. Można też zastosować grupę zdarzeń TRC_LOCKS, aby monitorować wszystkie zdarzenia związane z blokadami. Gdy utworzysz powiadomienie dotyczące tej grupy, otrzymasz komunikaty o następujących zdarzeniach: LOCK_DEADLOCK, LOCK_DEADLOCK_CHAIN, LOCK_ESCALATION i DEADLOCK_GRAPH.

UWAGA Wszystkie grupy zdarzeń są opisane w dokumentacji Books Online w artykule „DDL Event Groups for Use with Event Notifications”.

Powiadomienia o zdarzeniach można stosować do:

- rejestrowania i przeglądania zmian lub operacji z bazy danych albo serwera,
- asynchronicznego wykonywania operacji w reakcji na wystąpienie zdarzenia.

Powiadomienia o zdarzeniach to programistyczna alternatywa dla wyzwalaczy DDL i mechanizmu SQL Trace.

UWAGA Powiadomienia o zdarzeniach można tworzyć na poziomie serwera lub bazy danych.

Możesz utworzyć powiadomienie o zdarzeniach w bazie danych, aby otrzymywać informacje o powstaniu nowej tabeli. W tym celu wykonaj następujące kroki.

1. Otwórz skrypt *tworzBazy.sql* przedstawiony na listingu 12.5 (plik *tworzBazy.sql*). Skrypt tworzy dla przykładu bazę danych StoreEvent.

Listing 12.5. Plik *tworzBazy.sql*

```
IF NOT EXISTS(SELECT 1 FROM sys.databases WHERE name = 'StoreEvent')
CREATE DATABASE StoreEvent
GO

USE StoreEvent
GO

IF OBJECT_ID('EventTable') IS NULL
CREATE TABLE EventTable
(
```

```
RID int IDENTITY (1,1) NOT NULL
,EventDetails xml NULL
,EventDateTime datetime NULL
)
```

2. Następnie otwórz skrypt *tworzKolejki.sql* przedstawiony na listingu 12.6 (plik *tworzKolejki.sql*).

Listing 12.6. Plik *tworzKolejki.sql*

```
USE StoreEvent
GO

-- Tworzenie kolejki w celu pobierania szczegółowych informacji o zdarzeniu.
IF OBJECT_ID('dbo.NotifyQueue') IS NULL
CREATE QUEUE dbo.NotifyQueue
WITH STATUS = ON
    ,RETENTION = OFF
GO

-- Tworzenie usługi. Gdy zdarzenie zostanie zgłoszone, serwer
-- wysła do tej usługi komunikat.
-- Używany jest tu predefiniowany kontrakt.
IF NOT EXISTS(SELECT * FROM sys.services WHERE name =
'EventNotificationService')
CREATE SERVICE EventNotificationService
ON QUEUE NotifyQueue
([http://schemas.microsoft.com/SQL/Notifications/PostEventNotification])

IF NOT EXISTS(SELECT * FROM sys.routes WHERE name = 'NotifyRoute')

CREATE ROUTE NotifyRoute
WITH SERVICE_NAME = 'EventNotificationService',
ADDRESS = 'LOCAL';
GO
```

3. Ten skrypt dodaje do bazy StoreEvent kolejkę, w której zapisywane są dane o zdarzeniu utworzenia w bazie StoreEvent tabeli. Dodawana jest też usługa EventNotificationService Service Brokera, co umożliwia systemowi SQL Server wysyłanie komunikatów po wystąpieniu danego zdarzenia. Trasa NotifyRoute pomaga skierować komunikat do lokalnego egzemplarza systemu SQL Server. Uruchom ten skrypt.
4. Teraz należy utworzyć powiadomienie o zdarzeniach. Otwórz skrypt *tworzPowiadomOZdarz.sql* przedstawiony na listingu 12.7 (plik *tworzPowiadomOZdarz.sql*).

Listing 12.7. Plik *tworzPowiadomOZdarz.sql*

```
USE StoreEvent
GO
CREATE EVENT NOTIFICATION CreateTableNotification
ON DATABASE
FOR CREATE_TABLE
TO SERVICE 'EventNotificationService', 'current database' ;
```

Ten skrypt tworzy powiadomienie o zdarzeniach `CreateTableNotification`. Powiadamia ono o utworzeniu tabeli w bazie `StoreEvent`.

Komunikaty są przysyłane między usługami w sposób opisany w rozdziale 6. Tu utworzono docelowy koniec usługi — `EventNotificationService`. Źródłem jest sam system `SQL Server`.

5. Gdy w bazie `StoreEvent` tworzona jest tabela, w kolejce `NotifyQueue` pojawia się komunikat. Utwórz więc tabelę i uruchom poniższy skrypt, aby zobaczyć, co znajdzie się w kolejce:

```
SELECT CAST(message_body AS xml)
FROM NotifyQueue
```

6. Poniżej pokazujemy, jak wygląda zapisywany w kolejce gotowy komunikat w formacie XML:

```
<EVENT_INSTANCE>
  <EventType>CREATE_TABLE</EventType>
  <PostTime>2014-01-18T21:53:14.463</PostTime>
  <SPID>55</SPID>
  <ServerName>STEVENS_LAPTOP</ServerName>
  <LoginName>MicrosoftAccount\stevan_wort@hotmail.com</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>StoreEvent</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>TestTable1</ObjectName>
  <ObjectType>TABLE</ObjectType>
  <SqlCommand>
    <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON" ANSI_PADDING="ON"
      QUOTED_IDENTIFIER="ON" ENCRYPTED="FALSE" />
    <CommandText>create table test (
      a int null,
      b varchar(5) null,
      c datetime null
    )</CommandText>
  </SqlCommand>
</EVENT_INSTANCE>
```

W reakcji na to zdarzenie możesz podejmować określone działanie. W tym celu należy opracować procedurę składowaną i wywoływać ją w momencie pojawienia się komunikatu w kolejce. Zdarzenia z poziomu serwera są tworzone w ten sam sposób. Aby zapoznać się z pełną listą zdarzeń, dla których można tworzyć powiadomienia, pobierz dane z widoku `sys.event_notification_event_types`. W skrypcie *metadanePowiadomOZdarz.sql* pokazano, jak pobrać z katalogu listę zawierającą metadane o powiadomieniach o zdarzeniach.

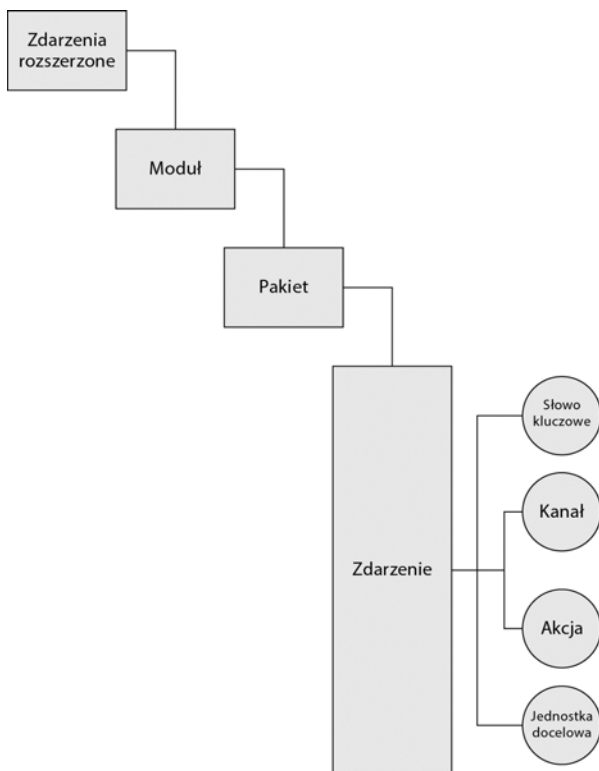
Zdarzenia rozszerzone systemu SQL Server

Zdarzenia rozszerzone systemu `SQL Server` wprowadzono w wersji `SQL Server 2008`. W systemie `SQL Server 2014` zdarzenia rozszerzone zostały rozbudowane o obsługę dodatkowych zdarzeń i nowy interfejs graficzny w programie `SQL Server Management Studio`. Zdarzenia rozszerzone umożliwiają przyjrzenie się wewnętrznym mechanizmom systemu `SQL Server` i są zaprojektowane po to, aby ułatwić szybsze diagnozowanie problemów z tym systemem. Zdarzenia rozszerzone mogą działać synchronicznie lub asynchronicznie względem zdarzeń systemu `SQL Server`. Ponadto są wysoce skalowalne i zajmują bardzo niewiele zasobów. Sesja `system_health` wymaga mniej zasobów niż ślad domyślny, a jednocześnie daje więcej możliwości. Zdarzenia rozszerzone wymagają mniej zasobów i zapewniają większą swobodę niż narzędzia `SQL Trace` i `SQL Server Profiler`.

Zdarzenia rozszerzone mają zastąpić niektóre starsze technologie monitorowania (np. narzędzie SQL Server Profiler). Wszystkie zdarzenia i kolumny dostępne w narzędziu SQL Server Profiler można też wykorzystać do zdarzeń rozszerzonych.

Obiekty zdarzeń rozszerzonych

W tym podpunkcie zawarliśmy wprowadzenie do obiektów zdarzeń rozszerzonych. Na rysunku 12.7 przedstawiamy ich hierarchię.



Rysunek 12.7. Obiekty zdarzeń rozszerzonych

Moduł

Moduł reprezentuje plik binarny zawierający zdarzenia. Byłby to odpowiednik pliku *SQLServr.exe* lub *MyDLL.dll*, gdybyś samodzielnie pisał moduły i wczytywał je do systemu SQL Server. Jedyne miejsce, w którym możesz zetknąć się z modulem, to atrybut pakietu w widoku DMV `sys.dm_xe_packages`.

Pakiet

Pakiet to obiekt kontenerowy w module. Pakiety zdarzeń rozszerzonych dostępne w systemie SQL Server możesz podejrzeć za pomocą widoku DMV `sys.dm_xe_packages`. Poniższy kod wyświetla zawartość tego widoku, a (przetłumaczone) wyniki są pokazane w tabeli 12.1.

```
select name, description
from sys.dm_xe_packages
```

Tabela 12.1. Informacje z widoku sys.dm_xe_packages

Nazwa	Opis
package0	Pakiet domyślny obejmujący wszystkie standardowe typy, odwzorowania, operatory porównywania, operacje i jednostki docelowe
sqlos	Zdarzenia rozszerzone dla systemu operacyjnego SQLOS
sqlserver	Zdarzenia rozszerzone dla systemu SQL Server
SecAudit	Zdarzenia związane z inspekcją zabezpieczeń
Ucs	Zdarzenia rozszerzone dotyczące technologii UCS (ang. <i>Unified Communications Stack</i>)
Sqlclr	Zdarzenia rozszerzone dotyczące środowiska SQL CLR
Filestream	Zdarzenia rozszerzone dotyczące mechanizmów FILESTREAM i FILETABLE
XtpRuntime	Zdarzenia rozszerzone dotyczące środowiska uruchomieniowego technologii XTP
XtpCompile	Zdarzenia rozszerzone dotyczące kompilacji w technologii XTP
XtpEngine	Zdarzenia rozszerzone dotyczące silnika technologii XTP
qds	Zdarzenia rozszerzone dotyczące mechanizmu Query Store

Podobnie jak modułów, nie będziesz sam tworzył pakietów, chyba że zechcesz napisać własny kod i dodać własne zdarzenia.

Zdarzenia

Zdarzenia to pierwsze „rzeczywiste” obiekty w hierarchii. *Zdarzenie* reprezentuje wystąpienie istotnych aktywności w systemie SQL Server. Aby lepiej zrozumieć zdarzenia, przyjrzyj się kilku z nich. Znajdziesz je w widoku DMV sys.dm_xe_objects (są to obiekty typu 'event'). Poniższy kod wyświetli listę typów zdarzeń:

```
select name
from sys.dm_xe_objects
where object_type = 'event'
order by name
```

Instrukcja ta zwraca 777 różnych zdarzeń. Jest to duży wzrost w porównaniu z 254 typami zdarzeń z systemu SQL Server 2008 i niewielka różnica w stosunku do 618 takich typów z wersji SQL Server 2012. Oto wybrane zdarzenia:

```
checkpoint_begin
checkpoint_end
lock_acquired
lock_deadlock
lock_released
locks_lock_waits
sp_statement_completed
sp_statement_starting
sql_statement_completed
sql_statement_starting
wait_info
wait_info_external
```

Wszystkie zdarzenia mają dwa dodatkowe atrybuty, Keyword i Channel. Atrybut Keyword pozwala grupować zdarzenia na podstawie jednostek, które je zgłaszają. Wartości tego atrybutu

to np. memory, broker, server itd. Atrybut Channel określa jednostki zainteresowane informacjami o zdarzeniu. W systemie SQL Server 2014 ten atrybut przyjmuje wartości:

- debug,
- analytical,
- operational,
- administration.

Aby zobaczyć wartości atrybutów Channel i Keyword zdarzeń, trzeba złączyć kilka widoków DMV z informacjami o zdarzeniach rozszerzonych. Ilustruje to poniższy kod:

```
select p.name as package_name
, k.event
, k.keyword
, c.channel
, k.description
from (
select c.object_package_guid as event_package
, c.object_name as event
, v.map_value as keyword
, o.description
from sys.dm_xe_object_columns as c inner join sys.dm_xe_map_values as v
on c.type_name = v.name
and c.column_value = v.map_key
and c.type_package_guid = v.object_package_guid
inner join sys.dm_xe_objects as o
on o.name = c.object_name
and o.package_guid = c.object_package_guid
where c.name = 'keyword'
) as k inner join (
select c.object_package_guid as event_package
, c.object_name as event
, v.map_value as channel
, o.description
from sys.dm_xe_object_columns as c inner join sys.dm_xe_map_values as v
on c.type_name = v.name
and c.column_value = v.map_key
and c.type_package_guid = v.object_package_guid
inner join sys.dm_xe_objects as o
on o.name = c.object_name
and o.package_guid = c.object_package_guid
where c.name = 'channel'
) as c
on
k.event_package = c.event_package and k.event = c.event
inner join sys.dm_xe_packages as p on p.guid = k.event_package
order by keyword
, channel
, event
```

W tabeli 12.2 przedstawiono wybrane zdarzenia razem z wartościami atrybutów Keyword i Channel.

Tabela 12.2. Wybrane zdarzenia rozszerzone

Nazwa pakietu	Zdarzenie	Atrybut Keyword	Atrybut Channel	Opis
sqlserver	broker_activation_task_aborted	broker	Admin	Anulowano zadanie aktywacji brokera
sqlserver	broker_activation_task_started	broker	Analytic	Uruchomiono zadanie aktywacji brokera
sqlserver	change_tracking_cleanup	change_tracking	Debug	Usuwanie informacji o zmianach ze śladu
sqlserver	app_domain_ring_buffer_recorded	clr	Debug	Zarejestrowano bufor cykliczny w domenie aplikacji
sqlserver	cursor_manager_cursor_end	cursor	Analytic	Wykryto koniec kursora
sqlserver	checkpoint_begin	database	Analytic	Początek tworzenia punktu kontrolnego
sqlserver	database_started	database	Operational	Uruchomiono bazę
sqlserver	deadlock_monitor_state_transition	deadlock_monitor	Debug	Zmiana stanu monitora zakleszczeń
sqlserver	error_reported	errors	Admin	Zgłoszono błąd
sqlserver	trace_print	errors	Debug	Wyświetlanie śladu
sqlserver	assert_fired	exception	Debug	Uruchomiono asercję
sqlos	dump_exception_routine_executed	exception	Debug	Wywołano procedurę generowania zrzutu
sqlserver	sql_statement_starting	execution	Analytic	Rozpoczęto wykonywanie instrukcji SQL-a
sqlserver	databases_log_file_size_changed	io	Analytic	Zmiana wielkości pliku dziennika bazy
sqlserver	file_read	io	Analytic	Wczytano plik
sqlserver	flush_file_buffers	io	Debug	Wywołano funkcję FlushFileBuffers

Akcje

Akcje to działania podejmowane w reakcji na zgłoszenie zdarzenia. Są one uruchamiane synchronicznie w wątku, który zgłosił zdarzenie. Dostępne akcje są zapisane w widoku DMV `sys.dm_xe_objects` (są to obiekty typu 'action'). Akcja umożliwia np. powiązanie uchwytu `plan_handle` i stosu instrukcji w języku T-SQL z konkretnym zdarzeniem. Powstaje wtedy niezwykle rozbudowana platforma o możliwościach znacznie przekraczających funkcje narzędzi SQL Trace i SQL Server Profiler.

Poniższa kwerenda zwraca wszystkie dostępne akcje:

```
select name
from sys.dm_xe_objects
where object_type = 'action'
order by name
```

Ta kwerenda zwraca 50 akcji. Niektóre z nich są wymienione poniżej:

```
attach_activity_id
attach_activity_id_xfer
callstack
collect_cpu_cycle_time
collect_system_time
create_dump_all_thread
create_dump_single_thread
database_context
database_id
debug_break
plan_handle
session_id
sos_context
sql_text
transaction_id
tsql_stack
```

Predykat

Predykat to filtr stosowany do zdarzenia bezpośrednio przed jego opublikowaniem. Predykat jest wyrażeniem logicznym, może być lokalny lub globalny i może przechowywać stan.

Predykaty są zapisane w widoku DMV `sys.dm_xe_objects`. Aby je wyświetlić, zastosuj poniższy kod w języku T-SQL:

```
select name, description
from sys.dm_xe_objects
where object_type = 'pred_compare'
order by name
-- 77 wierszy
```

```
select name, description
from sys.dm_xe_objects
where object_type = 'pred_source'
order by name
-- 44 wiersze
```

W tabeli 12.3 przedstawiamy kilka obiektów typu `pred_compare`.

Tabela 12.3. Wybrane obiekty typu `pred_compare`

Nazwa	Opis
<code>divides_by_uint64</code>	Określa, czy dana liczba typu <code>uint64</code> dzieli inną bez reszty
<code>equal_ansi_string</code>	Operator równości dla dwóch łańcuchów znaków ANSI
<code>greater_than_equal_float64</code>	Operator większy lub równy dla dwóch 64-bitowych liczb o podwójnej precyzji

Tabela 12.3. Wybrane obiekty typu pred_compare — ciąg dalszy

Nazwa	Opis
greater_than_i_sql_ansi_string	Operator większości dla dwóch łańcuchów znaków SQL ANSI
less_than_ansi_string	Operator mniejszości dla dwóch łańcuchów znaków ANSI
less_than_equal_i_unicode_string_ptr	Operator mniejszy lub równy dla dwóch wskaźników do łańcuchów znaków Unicode
less_than_int64	Operator mniejszości dla dwóch 64-bitowych liczb typu int ze znakiem
not_equal_ptr	Operator nierówności dla dwóch ogólnych wskaźników

W tabeli 12.4 zamieściliśmy wybrane obiekty typu pred_source.

Tabela 12.4. Wybrane obiekty typu pred_source

Nazwa	Opis
Counter	Określa liczbę wystąpień zdarzenia
cpu_id	Zwraca identyfikator bieżącego procesora
current_thread_id	Zwraca identyfikator bieżącego wątku systemu Windows
database_id	Zwraca identyfikator bieżącej bazy danych
node_affinity	Zwraca koligację bieżącego węzła NUMA
partitioned_counter	Licznik dla poszczególnych procesorów (wartość jest agregowana i podawana w przybliżeniu)
scheduler_address	Zwraca adres bieżącego programu szeregującego
scheduler_id	Zwraca identyfikator bieżącego programu szeregującego
session_id	Zwraca identyfikator bieżącej sesji
system_thread_id	Zwraca identyfikator bieżącego wątku systemowego
task_address	Zwraca adres bieżącego zadania
task_elapsed_quantum	Zwraca czas, jaki upłynął od rozpoczęcia danej porcji czasu
task_execution_time	Zwraca czas wykonywania bieżącego zadania
transaction_id	Zwraca identyfikator bieżącej transakcji
worker_address	Zwraca adres bieżącego wątku roboczego

Jednostka docelowa

Za pomocą *jednostki docelowej* można zdefiniować, co ma się stać z monitorowanymi zdarzeniami. W tabeli 12.5 podano siedemnaście jednostek docelowych zdefiniowanych w systemie SQL Server 2014. Jednostki, podobnie jak inne obiekty zdarzeń rozszerzonych, znajdziesz w widoku DMV sys.dm_xe_objects. Typ tych obiektów to 'target'. Dwie nowe jednostki docelowe w systemie SQL Server 2014 to asynchronous_security_audit_mds_log_target i compressed_history.

Tabela 12.5. Jednostki docelowe

Nazwa	Opis
asynchronous_router	Przekierowuje zdarzenia do odbiorników asynchronicznych
asynchronous_security_audit_event_log_target	Dziennik zdarzeń inspekcji zabezpieczeń systemu Windows NT, zapis asynchroniczny
asynchronous_security_audit_file_target	Plik inspekcji zabezpieczeń, zapis asynchroniczny
asynchronous_security_audit_mds_log_target	Dziennik inspekcji zabezpieczeń usług MDS, zapis asynchroniczny
asynchronous_security_audit_security_log_target	Dziennik zabezpieczeń systemu Windows NT, zapis asynchroniczny
compressed_history	Wykorzystuje historię do zachowania strumienia zdarzeń w wysoce skompresowanej postaci
etw_classic_sync_target	Mechanizm ETW (ang. <i>Event Tracing for Windows</i>), zapis synchroniczny
Event_counter	Zlicza wystąpienia każdego zdarzenia w sesji
Event_file	Zapisuje dane o zdarzeniach do pliku XEL (ang. <i>Expression Encoder Log</i>), który można zarchiwizować i później wykorzystać do analiz lub przeglądu; można scalać pliki XEL w celu wyświetlenia danych z różnych sesji zdarzeń
Event_stream	Strumień rejestrowanych na żywo zdarzeń, zapis asynchroniczny
histogram	Agreguje dane o zdarzeniach na podstawie konkretnej kolumny lub akcji; histogram umożliwia analizę rozkładu danych o zdarzeniach w czasie trwania sesji zdarzeń
pair_matching	Dopasowywanie zdarzeń w pary
ring_buffer	Bufor cykliczny, zapis asynchroniczny
router	Przekierowuje zdarzenia do odbiorników
synchronous_security_audit_event_log_target	Dziennik zdarzeń systemu Windows NT, zapis synchroniczny
synchronous_security_audit_file_target	Plik inspekcji zabezpieczeń, zapis synchroniczny
synchronous_security_audit_security_log_target	Dziennik zabezpieczeń systemu Windows NT, zapis synchroniczny

Poniższy kod w języku T-SQL zwraca listę jednostek docelowych z widoku `sys.dm_xe_objects`:

```
select name, description
from sys.dm_xe_objects
where object_type = 'target'
order by name
```

Sesja zdarzeń

W sesji zdarzeń wszystkie opisane wcześniej obiekty są łączone w celu wykonania określonych operacji. Przy użyciu sesji zdarzeń możesz zdefiniować, które obiekty chcesz uwzględnić przy rejestrowaniu zdarzenia.

Do tworzenia sesji służy instrukcja DDL CREATE EVENT SESSION. Umożliwia ona podanie wszystkich obiektów potrzebnych do utworzenia nowej sesji zdarzeń. Jedyna rzecz, której nie można zrobić w tej instrukcji, to uruchomienie sesji. Aby zachować spójność, najpierw trzeba utworzyć sesję, a dopiero potem można ją uruchomić za pomocą polecenia ALTER EVENT SESSION:

```
ALTER EVENT SESSION <nazwa sesji> STATE = START
```

W listingu 12.8 (plik *XEdlugieKwerendy.sql*) pokazujemy, jak utworzyć nową sesję zdarzeń pobierającą za pomocą obiektów sql_text i tsql_stack dane na temat instrukcji SQL-a przetwarzanych przez ponad 30 milisekund. Dane wyjściowe są zapisywane w pliku w formacie XML podanym w specyfikacji jednostki docelowej. Wyniki są przesyłane z pamięci do tego pliku co sekundę.

Listing 12.8. Plik XEdlugieKwerendy.sql

```
-- Tworzenie nowej sesji zdarzeń.
CREATE EVENT SESSION [long_running_queries] ON SERVER
ADD EVENT sqlserver.sql_statement_completed(
    SET collect_statement=(1)
    ACTION(sqlserver.sql_text
        ,sqlserver.tsql_stack)
    WHERE ([duration]>(30)))
ADD TARGET package0.event_file(SET
    filename=N'C:\ch12_samples\XEvents\long_running_queries.xel')
WITH (MAX_MEMORY=4096 KB
    ,EVENT_RETENTION_MODE=ALLOW_SINGLE_EVENT_LOSS
    ,MAX_DISPATCH_LATENCY=5 SECONDS
    ,MAX_EVENT_SIZE=0 KB
    ,MEMORY_PARTITION_MODE=NONE
    ,TRACK_CAUSALITY=OFF
    ,STARTUP_STATE=OFF)
GO
with (max_dispatch_latency = 1 seconds)
```

Polecenie ALTER SESSION najczęściej jest używane do uruchamiania i zatrzymywania sesji, jednak można je wykorzystać także do dodawania i usuwania zdarzeń z istniejących sesji. W poniższym fragmencie kodu pokazano, jak uruchomić sesję utworzoną na listingu 12.8.

```
-- Określanie, która sesja ma zostać zmodyfikowana.
alter event session long_running_queries on server
-- Teraz można wprowadzić zmiany.
-- Zmiana stanu (atrybut state) na start.
state = start
```

W kodzie z listingu 12.9 (plik *XEnodyfSesji.sql*) pokazujemy, jak za pomocą polecenia ALTER SESSION dodać zdarzenie do istniejącej sesji.

Listing 12.9. Plik XEnodyfSesji.sql

```
-- Określanie, która sesja zdarzeń ma zostać zmodyfikowana.
alter event session long_running_queries on server
-- Wprowadzanie zmian.
-- Dodawanie nowego zdarzenia — long_io_detected.
add event sqlserver.long_io_detected
(
    -- Dla tego zdarzenia używane są akcje sql_text i tsql_stack.
```

```
action(sqlserver.sql_text, sqlserver.tsq_stack)
```

```
-- Predykat nie jest używany; rejestrowane są wszystkie zdarzenia w celu wykrycia korelacji.
```

```
)
```

Aby sprawdzić, które sesje są aktywne, wywołaj następujące kwerendy:

```
select * from sys.dm_xe_sessions
select * from sys.dm_xe_session_events
```

Widoki z katalogu

Oto kilka widoków z katalogu udostępniających informacje na temat zdarzeń rozszerzonych:

- server_event_sessions,
- server_event_session_targets,
- server_event_session_fields,
- server_event_session_events,
- server_event_session_actions.

Widoki DMV

Zobaczyłeś już, jak używać niektórych widoków DMV z danymi o zdarzeniach rozszerzonych. Oto pełna lista tych widoków.

- Widok sys.dm_xe_map_values zwraca odwzorowanie z wewnętrznych kluczy liczbowych na zrozumiały tekst.
- Widok sys.dm_xe_object_columns zwraca informacje o schemacie wszystkich obiektów.
- Widok sys.dm_xe_objects zwraca po jednym wierszu dla każdego obiektu udostępnianego przez pakiet zdarzeń. Oto dostępne obiekty.
 - **Zdarzenia.** Określają istotne punkty w ścieżce wykonania. Wszystkie zdarzenia obejmują informacje o istotnych punktach.
 - **Akcje.** Są uruchamiane asynchronicznie po wystąpieniu zdarzenia. Akcja może dodać do zdarzenia datę jego zgłoszenia.
 - **Jednostki docelowe.** Odbierają zdarzenia albo synchronicznie (w wątku, który zgłosił zdarzenie), albo asynchronicznie (w wątku udostępnionym przez system).
 - **Źródła predykatów.** Pobierają wartości ze źródła zdarzenia i wykorzystują je w porównaniach. Predykaty porównują dane określonego typu i zwracają wartość logiczną.
 - **Typy.** Obejmują długość zbioru bajtów i jego cechy. Jest to niezbędne do zinterpretowania danych.
- Widok sys.dm_xe_packages wyświetla wszystkie pakiety zarejestrowane w silniku zdarzeń rozszerzonych.
- Widok sys.dm_xe_session_event_actions zwraca informacje na temat akcji sesji zdarzeń. Akcje są wykonywane w reakcji na zgłoszenie zdarzenia. Ten widok DMV udostępnia statystyki o liczbie uruchomień akcji i łącznym czasie działania.
- Widok sys.dm_xe_session_events zwraca informacje na temat zdarzeń z sesji. Zdarzeniom odpowiadają konkretne punkty wykonania kodu. Do zdarzeń można zastosować predykaty, co pozwala pominąć zdarzenie, jeśli nie zawiera ono odpowiednich informacji.

- Widok `sys.dm_xe_session_object_columns` wyświetla ustawienia obiektów powiązanych z daną sesją.
- Widok `sys.dm_xe_session_targets` zwraca informacje o jednostkach docelowych sesji.
- Widok `sys.dm_xe_sessions` zwraca informacje o aktywnych sesjach zdarzeń rozproszonych. Sesja w tym kontekście to zbiór zdarzeń, akcji i jednostek docelowych.

Używanie sesji zdarzeń rozszerzonych

Sesje i dostępne w nich dane można tworzyć, modyfikować, wyświetlać oraz analizować na różne sposoby.

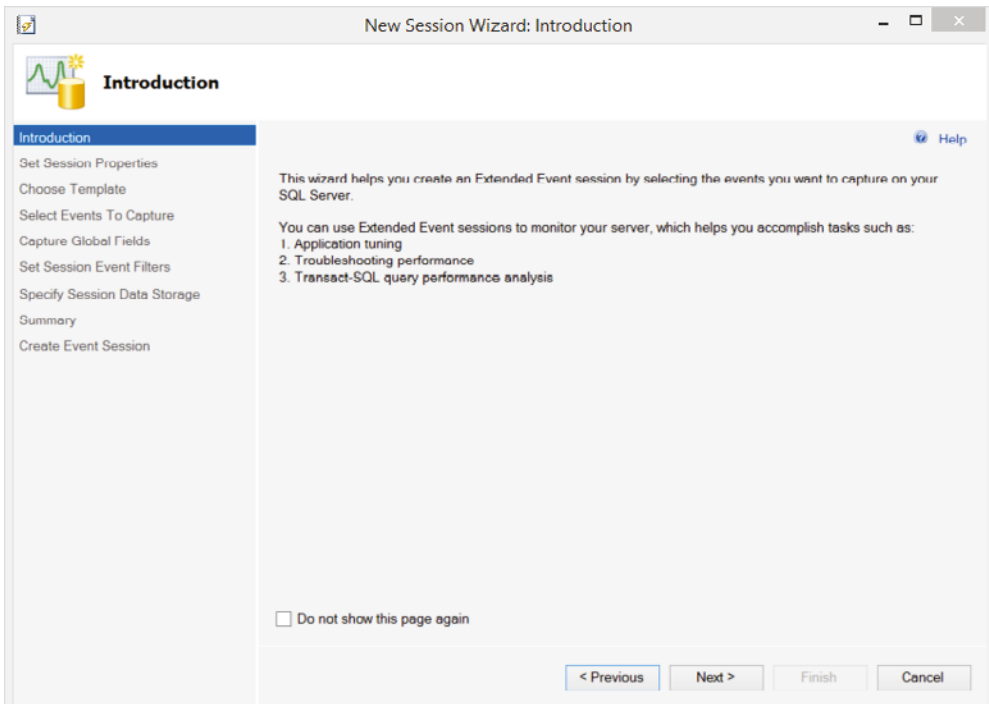
Możesz manipulować zdarzeniami rozszerzonymi za pomocą instrukcji DDL w języku T-SQL, tak jak w kilku przedstawionych wcześniej przykładach. Więcej przykładów ilustrujących tworzenie sesji zdarzeń rozszerzonych za pomocą języka T-SQL znajdziesz dalej w tym rozdziale.

Istnieją dwa interfejsy graficzne, które można stosować do zdarzeń rozszerzonych; są to New Session Wizard i ekran New Session.

Kreator New Session Wizard

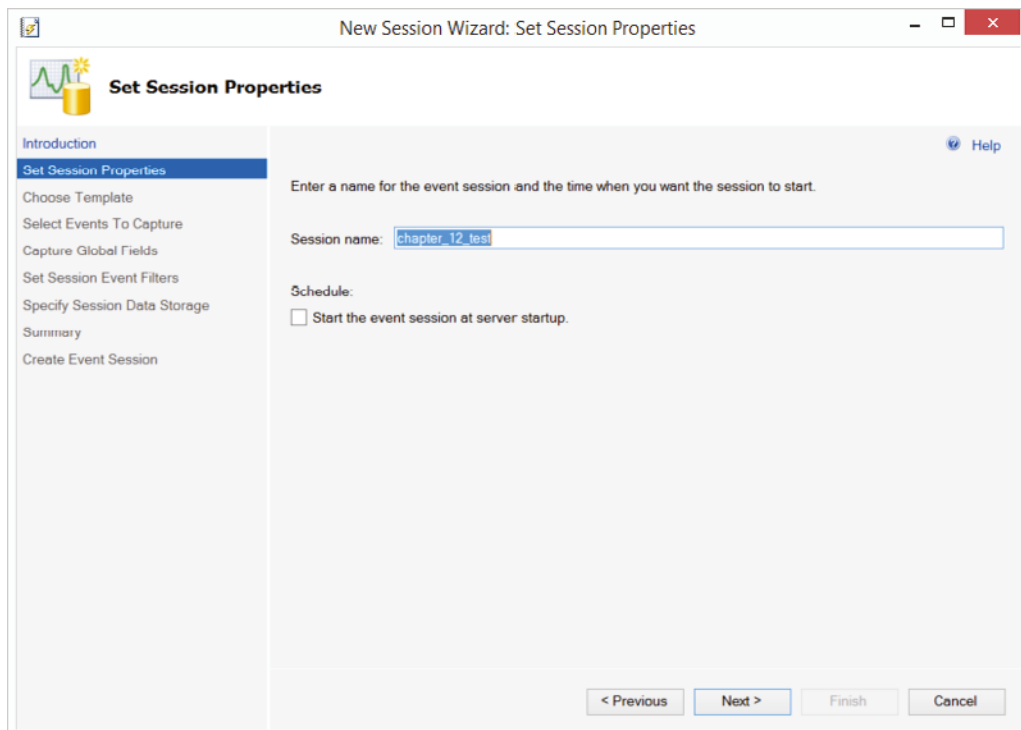
Kreator New Session Wizard prowadzi użytkownika przez proces tworzenia nowej sesji.

1. W programie SQL Server Management Studio rozwiń węzeł *Management/Extended Events/Sessions*. Kliknij węzeł *Sessions* prawym przyciskiem myszy i wybierz opcję *New Session Wizard*. Pojawi się strona *Introduction* kreatora New Session Wizard, widoczna na rysunku 12.8.



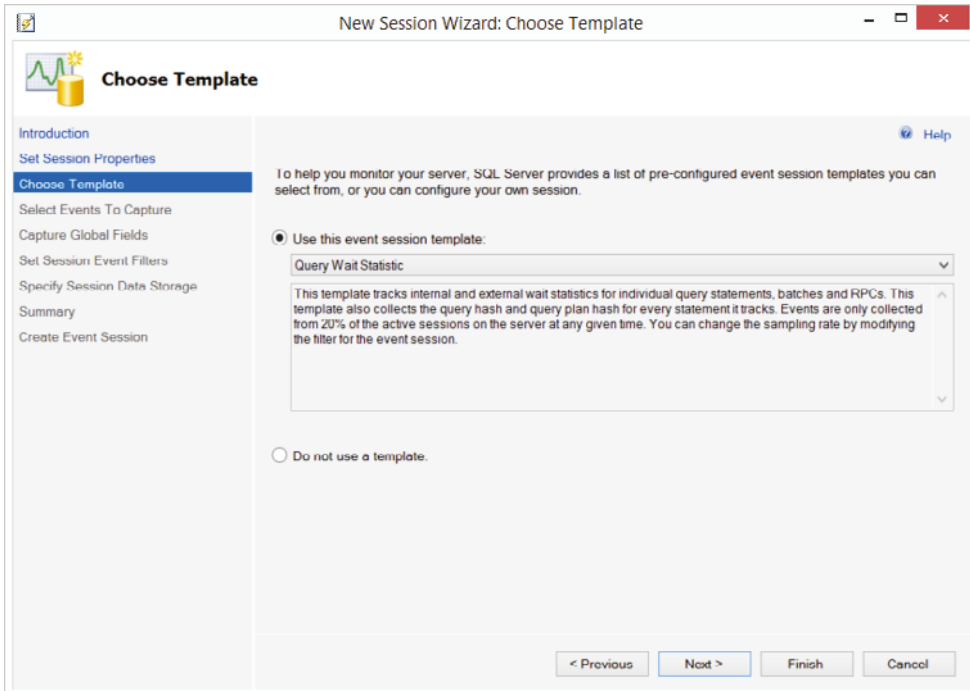
Rysunek 12.8. Pierwsza strona kreatora New Session Wizard

2. Kliknij przycisk *Next*, aby przejść do przedstawionej na rysunku 12.9 strony *Set Session Properties*. Tu można podać nazwę sesji i określić, czy sesja ma się rozpoczynać przy każdym uruchomieniu serwera. W tym przykładzie nadaj sesji nazwę **chapter_12_test**.

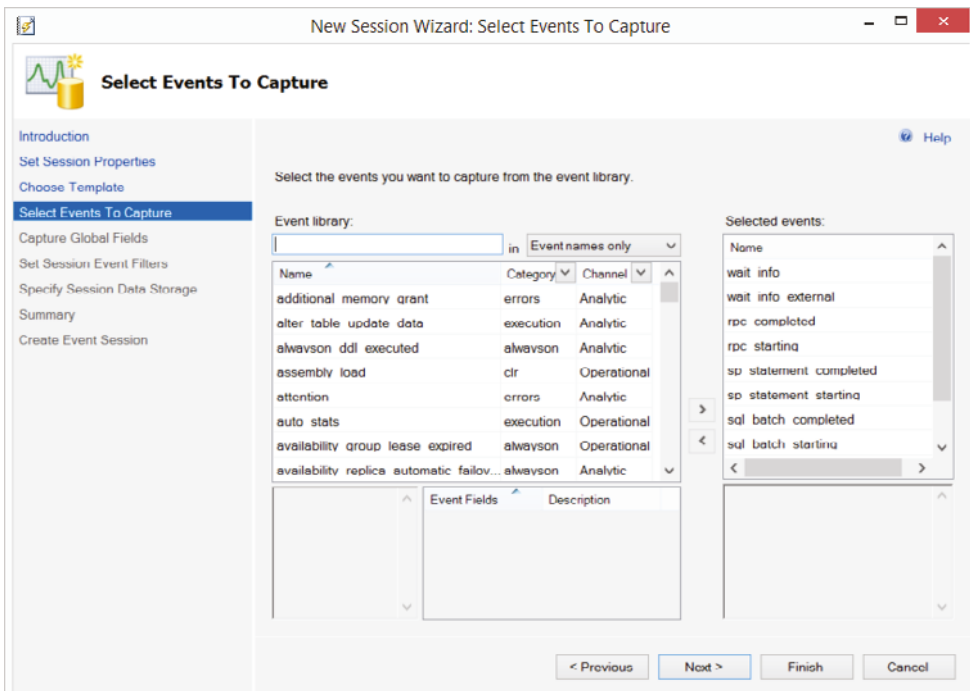


Rysunek 12.9. Ustawianie nazwy sesji

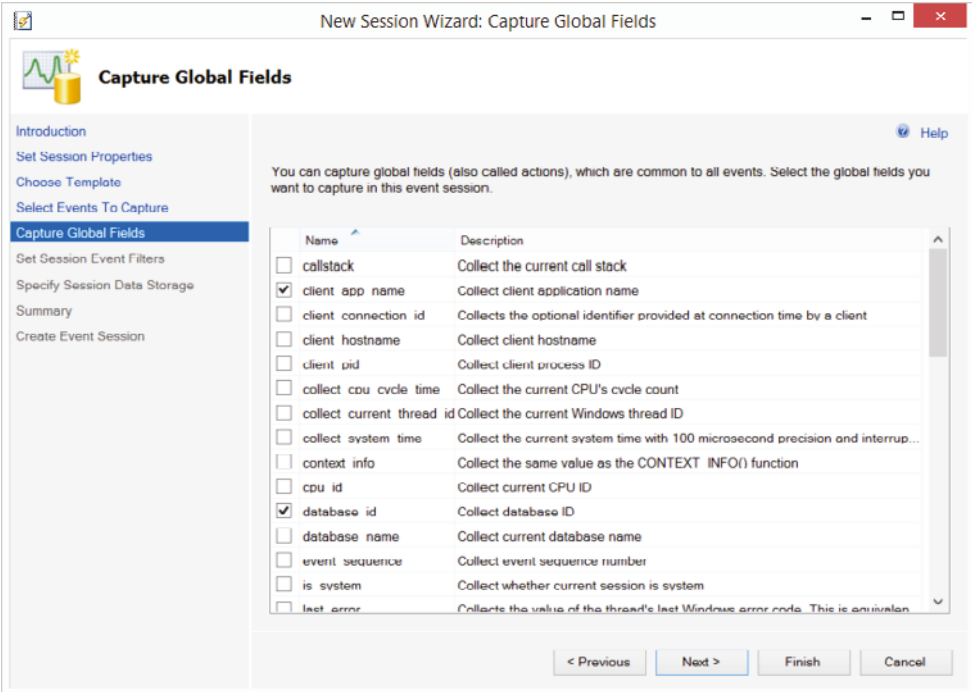
3. Kliknij przycisk *Next*, aby przejść do strony *Choose Template* widocznej na rysunku 12.10. Tu można wybrać predefiniowany szablon sesji zdarzeń lub zrezygnować z niego i dodać zdarzenia ręcznie. W tym przykładzie wybierz opcję *Use this event session template*, co spowoduje wypełnienie listy szablonów sesji zdarzeń. Wybierz szablon *Query Wait Statistics* i kliknij przycisk *Next*.
4. Na rysunku 12.11 przedstawiono następną stronę kreatora — *Select Events To Capture*. Ponieważ używany jest szablon, strona jest już wypełniona występującymi w nim zdarzeniami. Wybrane zdarzenia są widoczne w polu *Selected events*. Gdy wybrany jest szablon, na tej stronie nie trzeba nic robić.
5. Kliknij przycisk *Next*, aby przejść do strony *Capture Global Fields* widocznej na rysunku 12.12. Ponieważ używany jest szablon, niektóre globalne pola są już wybrane. Na rysunku 12.12 widać, że zaznaczone są pola `client_app_name` i `database_id`. Gdy przewiniesz ekran w dół, zobaczysz inne pola wybrane w używanym szablonie.
6. Kliknij przycisk *Next*, aby przejść do strony *Set Session Event Filters* widocznej na rysunku 12.13. Tu można wybrać filtry (*predykaty*), które pozwalają ograniczyć ilość rejestrowanych danych. W tym przykładzie nie dodawaj żadnych filtrów.



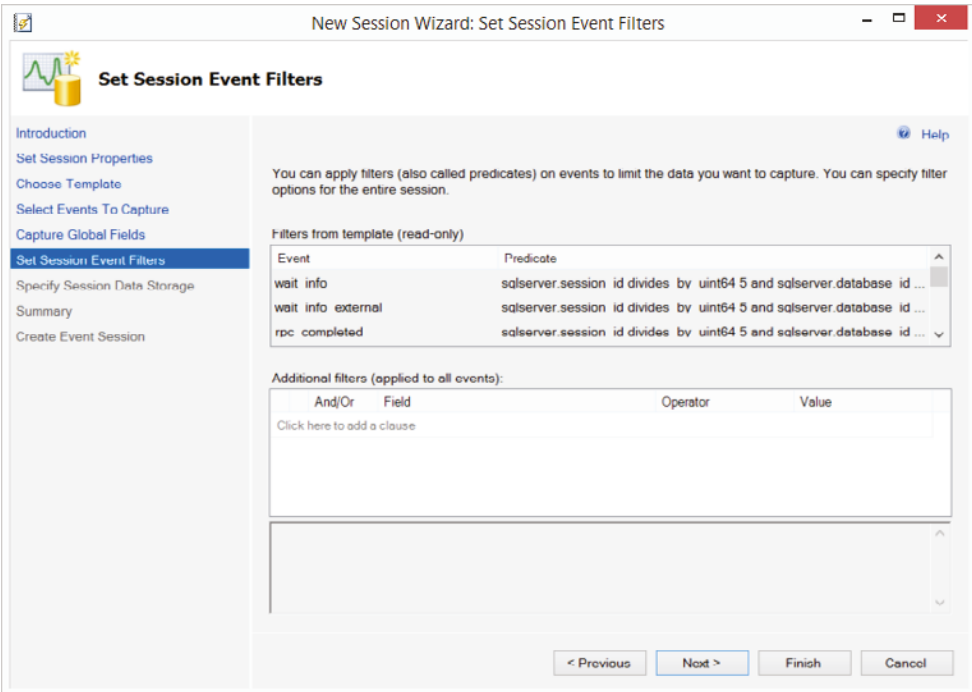
Rysunek 12.10. Wybieranie szablonu sesji zdarzeń



Rysunek 12.11. Na tej stronie widoczna jest lista zdarzeń z sesji

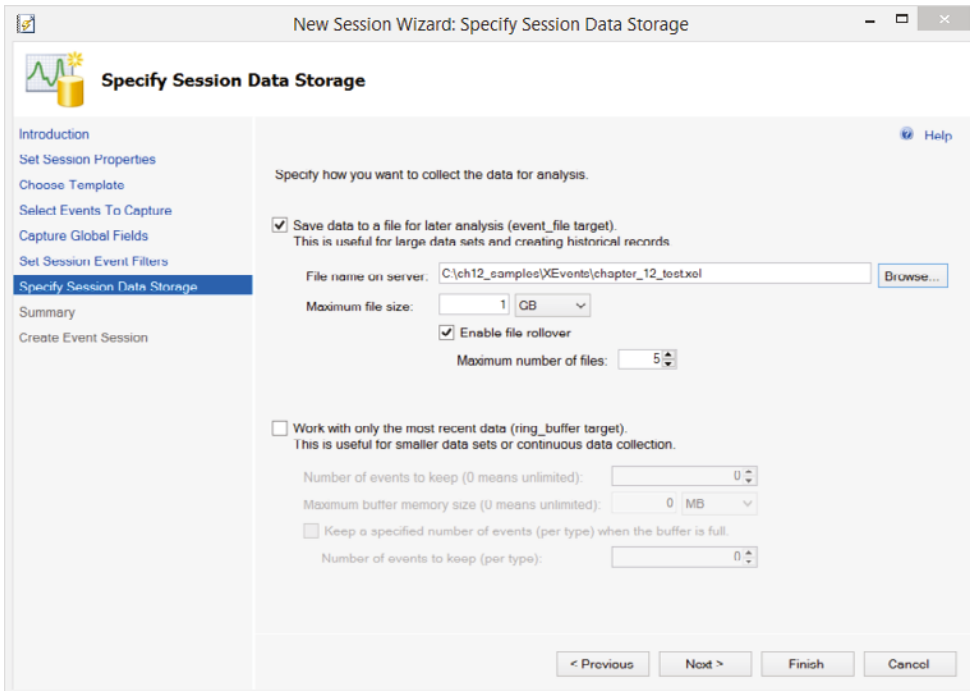


Rysunek 12.12. Niektóre pola są już wstępnie wybrane



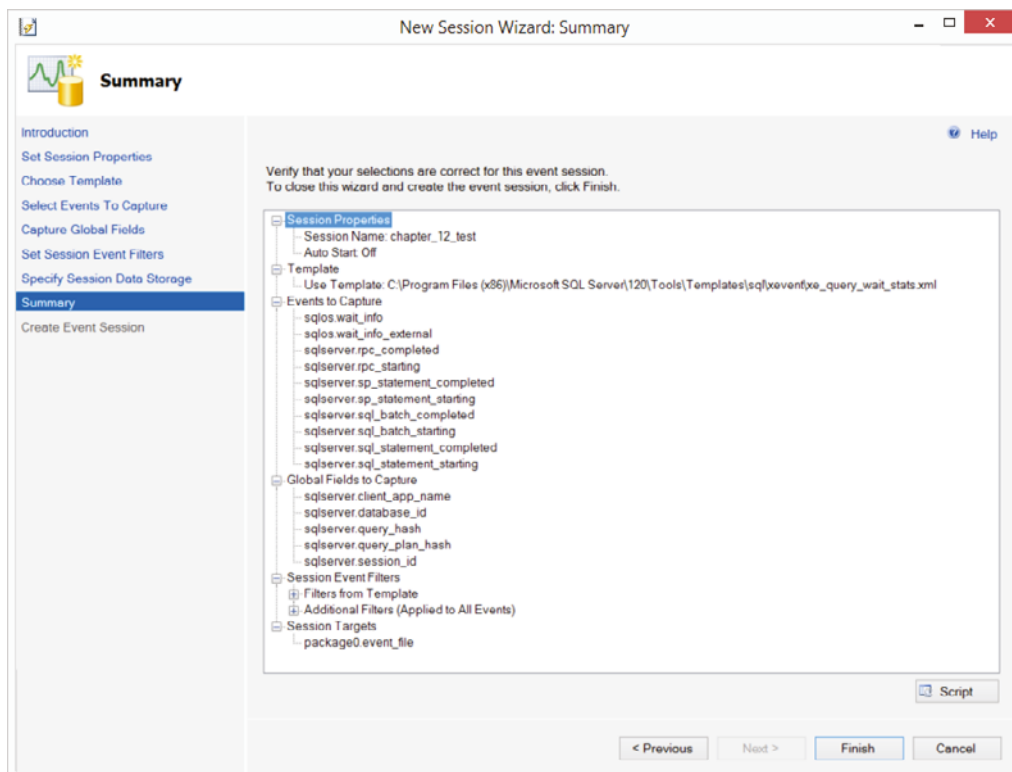
Rysunek 12.13. Na tej stronie można ustawić filtry sesji zdarzeń

7. Kliknij przycisk *Next*, aby wyświetlić stronę *Specify Session Data Storage* przedstawioną na rysunku 12.14. Tu można określić, gdzie dane mają być zapisywane. Dostępne są dwie opcje, *Save data to a file for later analysis* (czyli „zapisz dane do pliku na potrzeby późniejszej analizy”) i *Work with only the most recent data* (powodująca umieszczanie danych w buforze cyklicznym). W tym przykładzie wybierz opcję *Save data to a file for later analysis (event_file target)* i pozostaw wartości domyślne nazwy pliku, maksymalnej wielkości pliku, zastępowania plików i maksymalnej liczby plików.

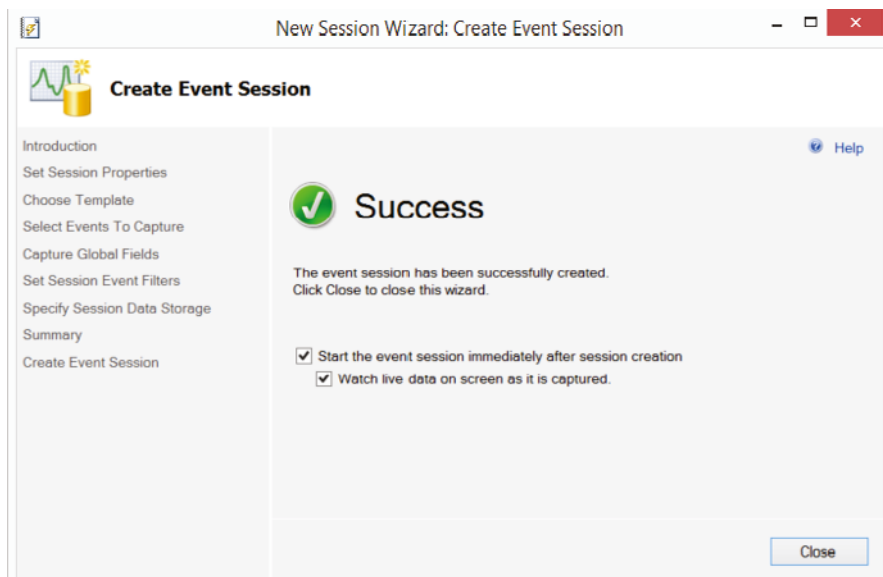


Rysunek 12.14. Ustawianie miejsca zapisu danych

8. Kliknij przycisk *Next*, aby przejść do strony *Summary* pokazanej na rysunku 12.15. Strona zawiera podsumowanie opcji wybranych w kreatorze. Zauważ, że na rysunku 12.15 wszystkie węzły są rozwinięte, dzięki czemu możesz zobaczyć wszystkie ustawione opcje. Jest to ostatnie miejsce, gdzie można zatwierdzić wybrane wartości. Następnie kreator na podstawie tych ustawień tworzy nową sesję zdarzeń.
9. Kliknij przycisk *Finish*, a kreator utworzy nową sesję zdarzeń. Jeśli proces zakończy się powodzeniem, zobaczysz stronę *Create Event Session Success* widoczną na rysunku 12.16. Następnie możesz natychmiast uruchomić sesję zdarzeń i obserwować na żywo, jak dane są rejestrowane. Zaznacz obie dostępne opcje (*Start the event session immediately after session creation* i *Watch live data on screen as it is captured*).
10. Kliknij przycisk *Close*, aby zamknąć kreator, uruchomić sesję zdarzeń i rozpocząć obserwowanie na żywo rejestrowania danych. Program SQL Server Management Studio zamknie kreator i wyświetli nową zakładkę ze zbieranymi na żywo danymi nowej sesji.



Rysunek 12.15. Strona z podsumowaniem ustawionych opcji

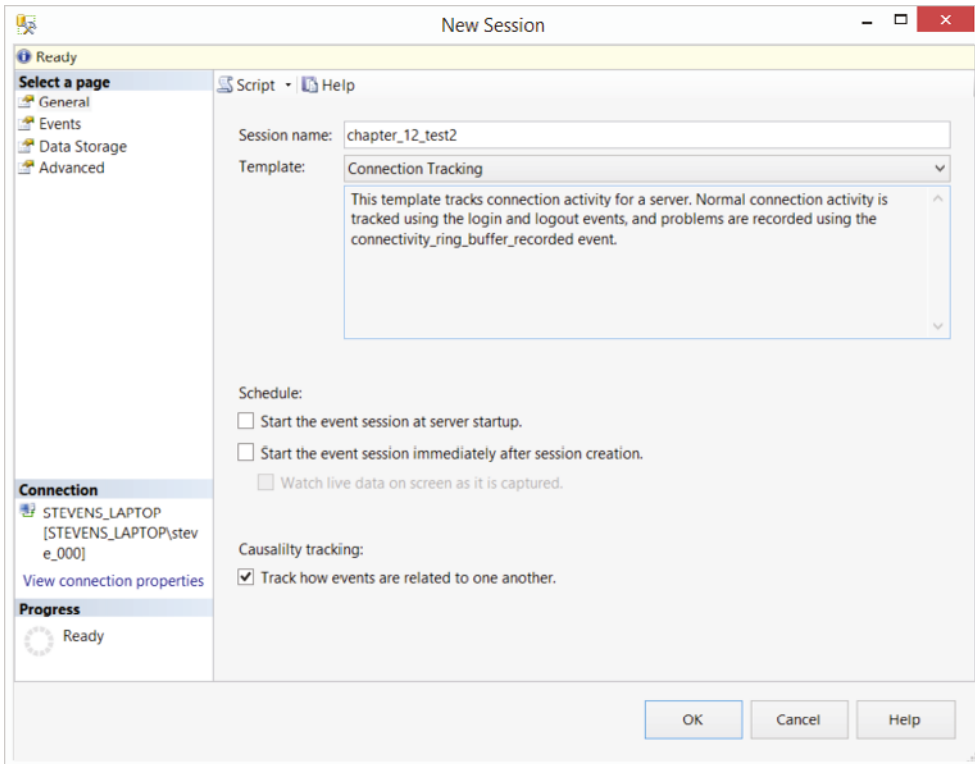


Rysunek 12.16. Teraz możesz od razu uruchomić sesję

Ekran New Session

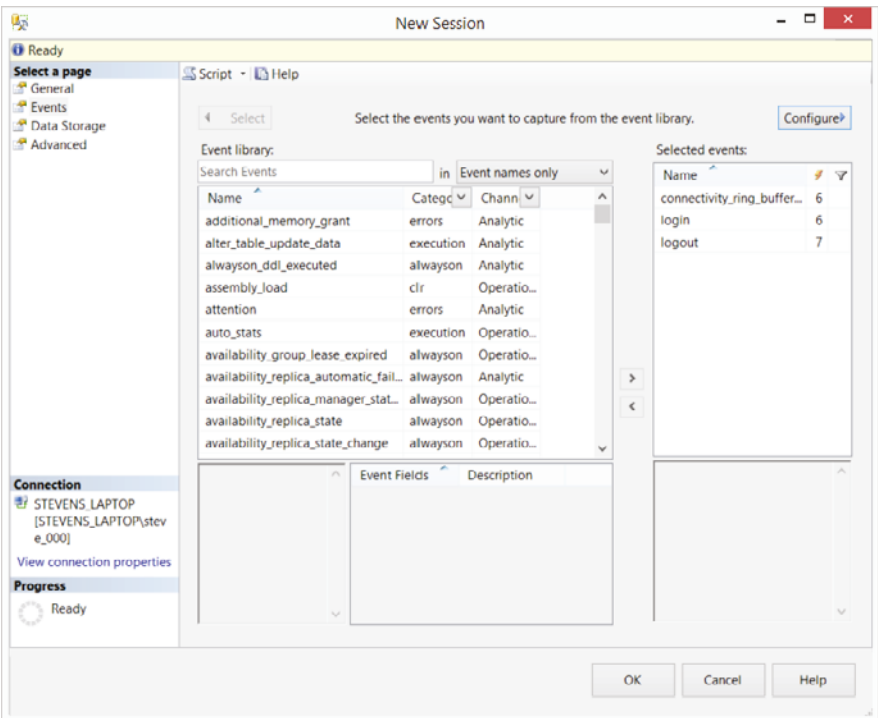
Ekran New Session można otworzyć w programie SQL Server Management Studio. W tym celu wykonaj następujące kroki.

1. Otwórz węzeł *Management/Extended Events/Sessions*. Kliknij węzeł *Sessions* prawym przyciskiem myszy i wybierz opcję *New Session*. Pojawi się strona *General* narzędzia New Session UI. Wprowadź nazwę sesji, **chapter_12_test2**, i wybierz szablon *Connection Tracking* (patrz rysunek 12.17). Zaznacz też opcję *Start the event session at server startup*.

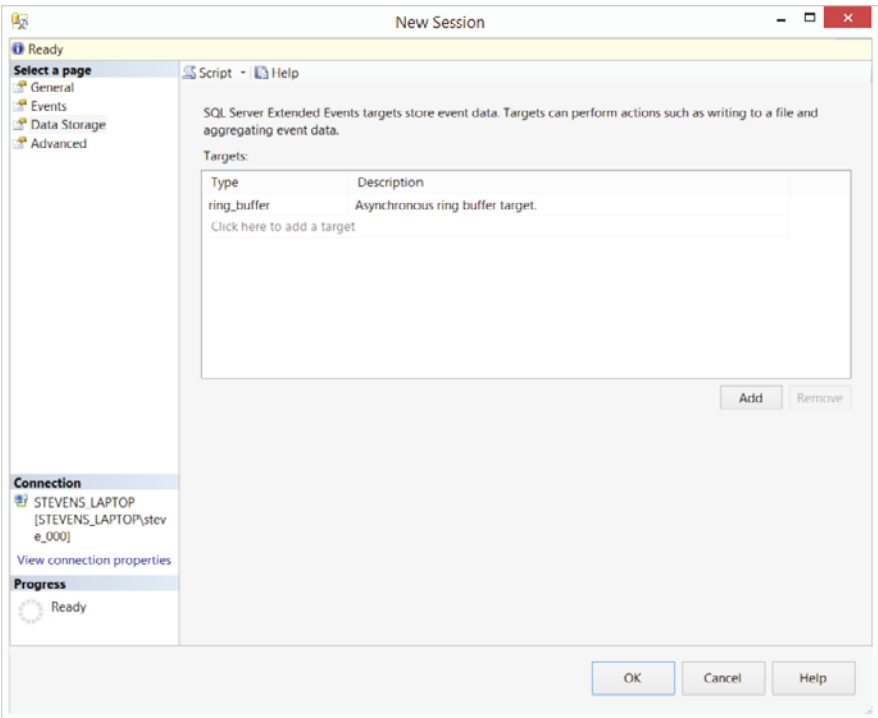


Rysunek 12.17. Strona General na ekranie New Session

2. Po lewej stronie otwórz stronę *Events*, aby zobaczyć, które zdarzenia zostały wybrane dla użytego szablonu (patrz rysunek 12.18). Ponieważ zastosowano szablon, na tej stronie nie trzeba nic zmieniać.
3. Teraz po lewej stronie wybierz stronę *Data Storage*, aby zdefiniować sposób przechowywania danych. Strona jest pokazana na rysunku 12.19. Domyślnie dane są zapisywane w buforze cyklicznym (opcja *ring_buffer*). Nie zmieniaj tego ustawienia. Jeśli zechcesz zmienić daną lokalizację docelową, zaznacz opcję *ring_buffer* i kliknij przycisk *Remove*. Aby dodać lokalizację, kliknij przycisk *Add*. Spowoduje to wstawienie nowego wiersza do tabeli *Targets*, gdzie można wybrać odpowiednie opcje, by skonfigurować nową lokalizację.

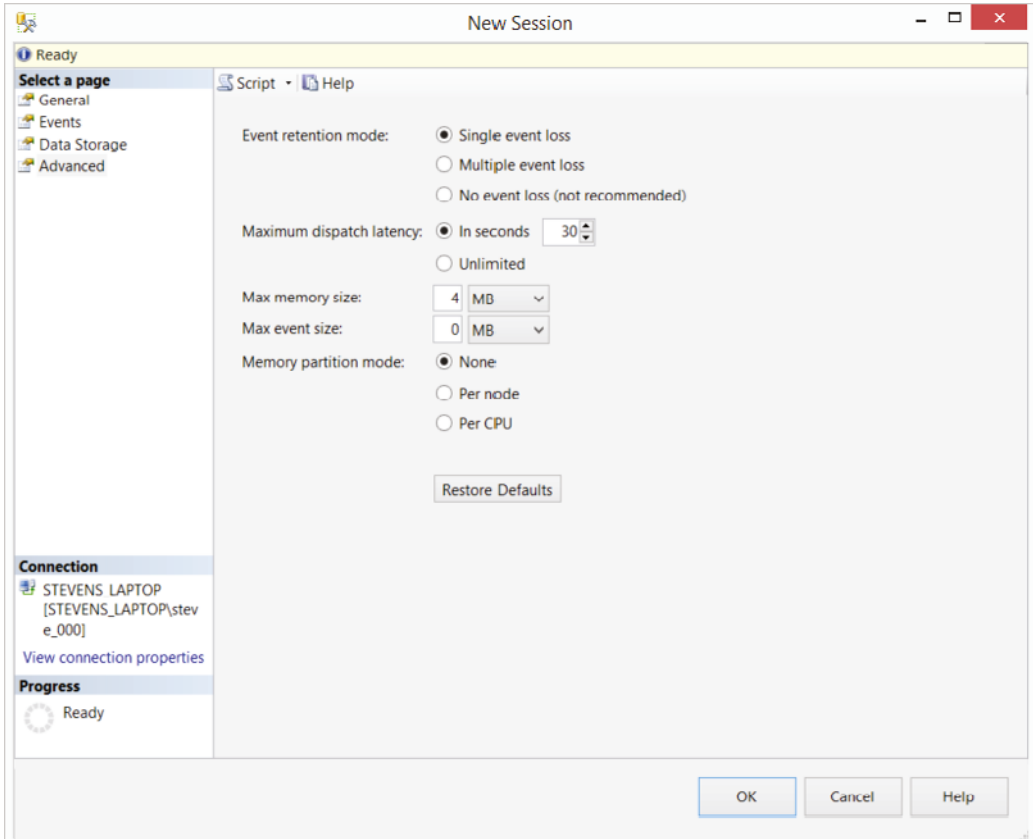


Rysunek 12.18. Lista zdarzeń z wybranego szablonu



Rysunek 12.19. Konfigurowanie miejsca zapisu danych

- Po lewej stronie otwórz stronę *Advanced*, aby ustawić zaawansowane opcje ekranu New Session. Strona jest przedstawiona na rysunku 12.20. W tym przykładzie możesz pozostawić widoczne na rysunku ustawienia.

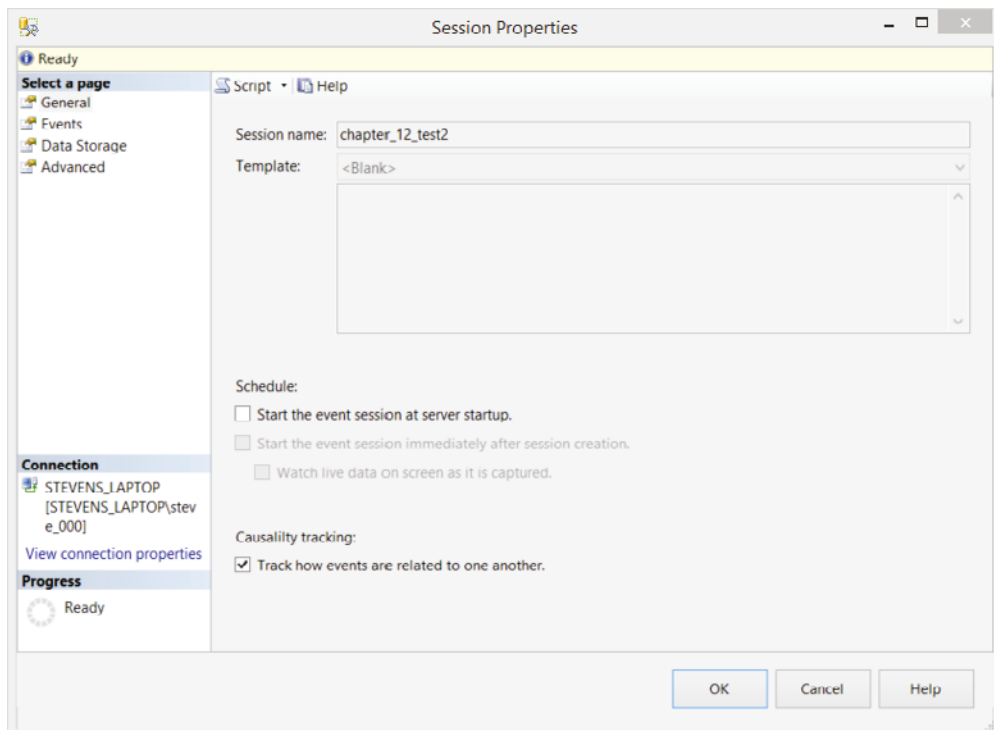


Rysunek 12.20. Zaawansowane ustawienia nowej sesji

- Na tym etapie sprawdziłeś już wszystkie opcje związane z tworzeniem nowej sesji. Aby ją utworzyć, kliknij przycisk OK. Interfejs użytkownika zniknie. Gdy otworzysz węzeł *Management/Extended Events/Sessions* w programie SQL Server Management Studio, zobaczysz, że dodana została nowa sesja o nazwie `chapter_12_test2`. Z powodu ustawionych opcji sesja ta na razie nie działa.
- Aby uruchomić nową sesję, kliknij ją prawym przyciskiem myszy i wybierz opcję *Start Session*. Jeśli chcesz zobaczyć zbierane na żywo dane, kliknij sesję prawym przyciskiem myszy i wybierz opcję *Watch Live Data*.

Modyfikowanie sesji

Aby zmodyfikować sesję, wybierz ją w programie SQL Server Management Studio. Kliknij węzeł sesji prawym przyciskiem myszy i wybierz opcję *Properties*. Zobaczysz ten sam zbiór stron, co na ekranie New Session, jednak tym razem na stronach wczytywane są ustawienia sesji, a niektóre opcje są niedostępne. Wybierz opcję *Properties* sesji `chapter_12_test2`, aby otworzyć okno dialogowe widoczne na rysunku 12.21.



Rysunek 12.21. W tym oknie możesz zmodyfikować ustawienia sesji

To okno dialogowe służy do wprowadzania zmian w licznych właściwościach sesji (niektóre dodatkowe opcje z tego okna zostały tu pominięte). Większość dostępnych ustawień jest zrozumiała, warto jednak sprawdzić je w dokumentacji Books Online.

Monitorowanie systemu za pomocą widoków DMV i funkcji DMF

Widoki DMV i funkcje DMF są ogromnym ułatwieniem dla administratorów baz danych. Narzędzia te zapewniają bogate informacje na temat stanu serwera i bazy danych. Widoki DMV i funkcje DMF są jak okno dające wgląd w to, co dzieje się w systemie SQL Server. Zwracają informacje o stanie serwera, które można wykorzystać do monitorowania sprawności danego egzemplarza systemu, diagnozowania problemów i dostrajania wydajności. Istnieją dwa rodzaje widoków DMV i funkcji DMF:

- z poziomu serwera,
- z poziomu bazy danych.

Wszystkie widoki DMV i funkcje DMF znajdują się w schemacie sys i mają nazwy w formacie dm_*. Aby wyświetlić informacje z widoków DMV z poziomu serwera, trzeba mieć uprawnienie VIEW SERVER STATE. Używanie widoków DMV i funkcji DMF z poziomu bazy danych wymaga uprawnienia VIEW DATABASE STATE. Po otrzymaniu uprawnienia VIEW STATE użytkownik może wyświetlać wszystkie widoki. Aby ograniczyć możliwości użytkownika, należy odmówić mu uprawnienia SELECT do widoków DMV i funkcji DMF, do których dana osoba nie powinna

mieć dostępu. Za pomocą poniższego kodu przyznajemy uprawnienie VIEW SERVER STATE użytkownikowi J_Nowak:

```
GRANT VIEW SERVER STATE TO [Europa\J_Nowak]
```

Jeśli chcesz uniemożliwić użytkownikowi [Europa\J_Nowak] wyświetlanie informacji z widoku sys.dm_os_wait_stats, musisz wywołać polecenie DENY SELECT:

```
DENY SELECT ON sys.dm_os_wait_stats TO [Europa\J_Nowak]
```

Widoki DMV i funkcje DMF są podzielone na następujące kategorie:

- związane z grupami dostępności AlwaysOn,
- związane z rejestrowaniem zmian w danych,
- związane ze śledzeniem zmian,
- związane ze środowiskiem CLR,
- związane z kopiami lustrzanymi baz danych,
- związane z bazami danych,
- związane z wykonywaniem operacji,
- związane z mechanizmami FILESTREAM i FILETABLE,
- związane z wyszukiwaniem pełnotekstowym i semantycznym,
- związane z indeksami,
- związane z operacjami wejścia-wyjścia,
- związane z obiektami,
- związane z powiadomieniami o kwerendach,
- związane z replikacją,
- związane z narzędziem Resource Governor,
- związane z zabezpieczeniami,
- związane z narzędziem Service Broker,
- związane z systemem SQLOS,
- związane z transakcjami.

W tym punkcie nie przedstawiamy wszystkich widoków, a w zamian omawiamy związane z nimi standardowe zadania wykonywane przez administratorów baz danych w ramach monitorowania systemu SQL Server. Szczegółowe informacje o wszystkich widokach DMV i funkcjach DMF znajdziesz w dokumentacji Books Online w dziale „Dynamic Management Views and Functions”.

Poniżej opisujemy przykładowe scenariusze, w których można zastosować widoki DMV i funkcje DMF. Tam znajdują się tylko wybrane przykłady. W przykładowych plikach dotyczących widoków DMV znajdziesz więcej kodu ilustrującego monitorowanie systemu SQL Server.

Co dzieje się w systemie SQL Server?

W następnych podpunktach pokazujemy, jak pobierać dane z widoków DMV, aby ustalić, co dzieje się w systemie SQL Server.

Obecnie wykonywane kwerendy

W listingu 12.10 (plik *wykonywaneKwerendy.sql*) pokazujemy kod w SQL-u aktualnie wykonywanych kwerend. Pomaga on ustalić, które kwerendy w danym momencie działają, i wyświetla ich kod.

Jest to przydatne, gdy użytkownik chce określić nie tylko identyfikatory SPID i `session_id`, ale też wykonywany kod w języku T-SQL.

Listing 12.10. Plik wykonywaneKwerendy.sql

```
select r.session_id
      ,r.status
      ,substring(qt.text,r.statement_start_offset/2,
        (case when r.statement_end_offset = -1
          then len(convert(nvarchar(max), qt.text)) * 2
        else r.statement_end_offset end - r.statement_start_offset)/2)
      as query_text
      ,qt.dbid
      ,qt.objectid
      ,r.cpu_time
      ,r.total_elapsed_time
      ,r.reads
      ,r.writes
      ,r.logical_reads
      ,r.scheduler_id
from sys.dm_exec_requests as r
cross apply sys.dm_exec_sql_text(sql_handle) as qt
inner join sys.dm_exec_sessions as es on r.session_id = es.session_id
where es.is_user_process = 1
order by r.cpu_time desc
```

Które jednostki używają poszczególnych zasobów?

W listingu 12.11 (plik *wykorzystZasobow.sql*) sprawdzamy tabele systemowe, obecnie uznane za przestarzałe. Ten kod próbuje tabelę `sysprocesses` przez 10 sekund i informuje o zmianach między pierwszą a drugą próbką. W ten sposób można sprawdzić obciążenie procesora, fizycznych operacji wejścia-wyjścia i pamięci. Choć tabela `sysprocesses` jest uznawana za przestarzałą, zawiera wszystkie potrzebne informacje w jednym miejscu i dogodnym formacie. Odpowiadające jej widoki DMV wymagają złączania. Ponieważ tabela `sysprocesses` jest tak łatwa w użyciu, zalecamy korzystanie z niej, dopóki będzie dostępna.

Listing 12.11. Plik wykorzystZasobow.sql

```
-- Kto zajmuje poszczególne zasoby?
select spid, kpid, cpu, physical_io, memusage, sql_handle, 1 as sample,
getdate() as sampleTime, hostname, program_name, nt_username
into #Resources
from master..sysprocesses

waitfor delay '00:00:10'

Insert #Resources
select spid, kpid, cpu, physical_io, memusage, sql_handle, 2 as sample,
getdate() as sampleTime, hostname, program_name, nt_username
from master..sysprocesses

-- Określanie różnic
select r1.spid
      , r1.kpid
      , r2.cpu - r1.cpu as d_cpu_total
```

```

, r2.physical_io - r1.physical_io as d_physical_io_total
, r2.memusage - r1.memusage as d_memusage_total
, r1.hostname, r1.program_name, r1.nt_username
, r1.sql_handle
, r2.sql_handle
from #resources as r1 inner join #resources as r2 on r1.spid = r2.spid
    and r1.kpid = r2.kpid
where r1.sample = 1
and r2.sample = 2
and (r2.cpu - r1.cpu) > 0
order by (r2.cpu - r1.cpu) desc

select r1.spid
, r1.kpid
, r2.cpu - r1.cpu as d_cpu_total
, r2.physical_io - r1.physical_io as d_physical_io_total
, r2.memusage - r1.memusage as d_memusage_total
, r1.hostname, r1.program_name, r1.nt_username
into #Usage
from #resources as r1 inner join #resources as r2 on r1.spid = r2.spid
    and r1.kpid = r2.kpid
where r1.sample = 1
and r2.sample = 2
and (r2.cpu - r1.cpu) > 0
order by (r2.cpu - r1.cpu) desc

select spid, hostname, program_name, nt_username
, sum(d_cpu_total) as sum_cpu
, sum(d_physical_io_total) as sum_io
from #Usage
group by spid, hostname, program_name, nt_username
order by 6 desc

```

```

drop table #resources
drop table #Usage

```

Które jednostki oczekują na zasoby?

W listingu 12.12 (plik *ktoCzeka.sql*) wyświetlamy zadania oczekujące obecnie na zasoby. Wykorzystywana jest tu ta sama technika próbkowania, co w poprzedniej kwerendzie.

Listing 12.12. Plik *ktoCzeka.sql*

```

select
* , 1 as sample
, getdate() as sample_time
into #waiting_tasks
from sys.dm_os_waiting_tasks

waitfor delay '00:00:10'

insert #waiting_tasks
select
* , 2
, getdate()
from sys.dm_os_waiting_tasks

```

```
-- Obliczanie różnic
select w1.session_id
, w1.exec_context_id
, w2.wait_duration_ms - w1.wait_duration_ms as d_wait_duration
, w1.wait_type
, w2.wait_type
, datediff(ms, w1.sample_time, w2.sample_time) as interval_ms
from #waiting_tasks as w1 inner join #waiting_tasks as w2 on w1.session_id =
w2.session_id
and w1.exec_context_id = w2.exec_context_id
where w1.sample = 1
and w2.sample = 2
order by 3 desc

-- select * from #waiting_tasks

drop table #waiting_tasks
```

Statystyki oczekiwania

W listingu 12.13 (plik *statOczekiwania.sql*) próbkujemy statystyki oczekiwania, aby sprawdzić, co się zmieniło w badanym okresie.

Listing 12.13. Plik statOczekiwania.sql

```
select *, 1 as sample, getdate() as sample_time
into #wait_stats
from sys.dm_os_wait_stats
waitfor delay '00:00:30'
insert #wait_stats
select *, 2, getdate()
from sys.dm_os_wait_stats

-- Obliczanie różnic

select w2.wait_type
, w2.waiting_tasks_count - w1.waiting_tasks_count as d_wtc
, w2.wait_time_ms - w1.wait_time_ms as d_wtm
, cast((w2.wait_time_ms - w1.wait_time_ms) as float) /
cast((w2.waiting_tasks_count - w1.waiting_tasks_count) as float) as avg_wtm
, datediff(ms, w1.sample_time, w2.sample_time) as interval
from #wait_stats as w1 inner join #wait_stats as w2 on w1.wait_type =
w2.wait_type
where w1.sample = 1
and w2.sample = 2
and w2.wait_time_ms - w1.wait_time_ms > 0
and w2.waiting_tasks_count - w1.waiting_tasks_count > 0
order by 3 desc

drop table #wait_stats
```

Wyświetlanie informacji o blokadach z bazy danych

W listingu 12.14 (plik *blokowanie.sql*) pokazujemy, jak pobrać informacje o blokadach z wybranej bazy danych.

Listing 12.14. Plik blokowanie.sql

```

SELECT l.resource_type, l.resource_associated_entity_id
,OBJECT_NAME(sp.OBJECT_ID) AS ObjectName
,l.request_status, l.request_mode, request_session_id
,l.resource_description
FROM sys.dm_tran_locks l
LEFT JOIN sys.partitions sp
  ON sp.hobt_id = l.resource_associated_entity_id
WHERE l.resource_database_id = DB_ID()

```

Wyświetlanie informacji o blokadach z serwera

Kod z listingu 12.15 (plik *blokady.sql*) zwraca informacje o blokadach z serwera.

Listing 12.15. Plik blokady.sql

```

SELECT
  t1.resource_type
,t1.resource_database_id
,t1.resource_associated_entity_id
,OBJECT_NAME(sp.OBJECT_ID) AS ObjectName
,t1.request_mode
,t1.request_session_id
,t2.blocking_session_id
FROM sys.dm_tran_locks as t1
JOIN sys.dm_os_waiting_tasks as t2
  ON t1.lock_owner_address = t2.resource_address
LEFT JOIN sys.partitions sp
  ON sp.hobt_id = t1.resource_associated_entity_id

```

Używanie indeksu w bazie danych

Za pomocą listingu 12.16 (plik *uzywaneIndeksy.sql*) uzyskamy informacje o używaniu indeksu w bazie danych, w której uruchamiana jest dana kwerenda. Kod z tego listingu tworzy tabelę i zapisuje w niej wyniki, dzięki czemu można je później przeanalizować. Przedstawione kwerendy pomagają ustalić, które indeksy są naprawdę przydatne w aplikacji. Warto wywoływać te kwerendy przez okres kilku dni (zamiast sprawdzać dane z tylko jednego dnia), ponieważ pomaga to lepiej zrozumieć ogólny obraz systemu. Pamiętaj, że widoki DMV się zmieniają — są ponownie inicjowane przy każdym uruchomieniu systemu SQL Server.

Listing 12.16. Plik uzywaneIndeksy.sql

```

IF OBJECT_ID('dbo.IndexUsageStats') IS NULL
CREATE TABLE dbo.IndexUsageStats
(
  IndexName sysname NULL
,ObjectName sysname NOT NULL
,user_seeks bigint NOT NULL
,user_scans bigint NOT NULL
,user_lookups bigint NOT NULL
,user_updates bigint NOT NULL
,last_user_seek datetime NULL

```

```

,last_user_scan datetime NULL
,last_user_lookup datetime NULL
,last_user_update datetime NULL
,StatusDate datetime NOT NULL
,DatabaseName sysname NOT NULL
)

GO
-- Ta kwerenda informuje o indeksach UŻYWANYCH dla tabel bazy danych.
INSERT INTO dbo.IndexUsageStats
(
    IndexName
,ObjectName
,user_seeks
,user_scans
,user_lookups
,user_updates
,last_user_seek
,last_user_scan
,last_user_lookup
,last_user_update
,StatusDate
,DatabaseName
)
SELECT
    si.name AS IndexName
,so.name AS ObjectName
,diu.user_seeks
,diu.user_scans
,diu.user_lookups
,diu.user_updates
,diu.last_user_seek
,diu.last_user_scan
,diu.last_user_lookup
,diu.last_user_update
,GETDATE() AS StatusDate
,sd.name AS DatabaseName
FROM sys.dm_db_index_usage_stats diu
JOIN sys.indexes si
    ON diu.object_id = si.object_id
    AND diu.index_id = si.index_id
JOIN sys.all_objects so
    ON so.object_id = si.object_id
JOIN sys.databases sd
    ON sd.database_id = diu.database_id
WHERE is_ms_shipped <> 1
    AND diu.database_id = DB_ID()

```

Indeksy nieużywane w bazie danych

Kod z listingu 12.17 (plik *nieuzywaneIndeksy.sql*) zwraca informacje o tym, które indeksy nie są używane. Jeśli dany indeks nie jest używany, należy pomyśleć o jego usunięciu, ponieważ jego tworzenie i konserwacja niepotrzebnie zajmują czas. Wyniki są zapisywane w tabeli `NotUsedIndexes`. Warto wywoływać te kwerendy przez okres kilku dni (zamiast sprawdzać dane z tylko jednego dnia), ponieważ pomaga to lepiej zrozumieć ogólny obraz systemu. Pamiętaj, że widoki DMV się zmieniają — są ponownie inicjowane przy każdym uruchomieniu systemu SQL Server.

Listing 12.17. Plik nieuzywaneIndeksy.sql

```

-----
-- Ten kod zapisuje indeksy, które nie są używane.
IF OBJECT_ID('dbo.NotUsedIndexes') IS NULL
CREATE TABLE dbo.NotUsedIndexes
(
    IndexName sysname NULL
,ObjectName sysname NOT NULL
,StatusDate datetime NOT NULL
,DatabaseName sysname NOT NULL
)

-- Ta kwerenda zwraca istniejące w poszczególnych tabelach bazy indeksy, które NIE SĄ używane.
INSERT dbo.NotUsedIndexes
(
    IndexName
,ObjectName
,StatusDate
,DatabaseName
)
SELECT
    si.name AS IndexName
,so.name AS ObjectName
,GETDATE() AS StatusDate
,DB_NAME()
FROM sys.indexes si
JOIN sys.all_objects so
    ON so.object_id = si.object_id
WHERE si.index_id NOT IN (SELECT index_id
                        FROM sys.dm_db_index_usage_stats diu
                        WHERE si.object_id = diu.object_id
                          AND si.index_id = diu.index_id
                        )
    AND so.is_ms_shipped <> 1

```

Wyświetlanie kwerend oczekujących na pamięć

Kod z listingu 12.18 (plik *oczekiwanieNaPamiec.sql*) wyświetla kwerendy oczekujące na przyznanie pamięci. System SQL Server analizuje kwerendę i na podstawie planu wykonania ustala, ile pamięci będzie ona potrzebować. Jeśli w danym momencie pamięć jest niedostępna, kwerenda zostaje wstrzymana. Gdy kwerenda oczekuje na pamięć, w widoku DMV `sys.dm_exec_query_memory_grants` znajduje się wpis jej dotyczący.

Listing 12.18. Plik oczekiwanieNaPamiec.sql

```

SELECT
    es.session_id AS SPID
,es.login_name
,es.host_name
,es.program_name, es.status AS Session_Status
,mg.requested_memory_kb
,DATEDIFF(mi, mg.request_time
, GETDATE()) AS [WaitingSince-InMins]
FROM sys.dm_exec_query_memory_grants mg

```

```
JOIN sys.dm_exec_sessions es
  ON es.session_id = mg.session_id
WHERE mg.grant_time IS NULL
ORDER BY mg.request_time
```

Informacje o podłączonych użytkownikach

W listingu 12.19 (plik *podlaczeniUzyt.sql*) znajdziemy informację, którzy użytkownicy są podłączeni do systemu i ile sesji jest otwartych dla każdego z nich.

Listing 12.19. Plik podlaczeniUzyt.sql

```
SELECT login_name
, count(session_id) as session_count
FROM sys.dm_exec_sessions
GROUP BY login_name
```

Wolna przestrzeń w grupach plików

Kod z listingu 12.20 (plik *wolneMiejsceDlaGrupyPlikow.sql*) określa, ile wolnego miejsca pozostaje w każdej grupie plików. Jest to przydatne, gdy baza wykorzystuje wiele grup plików. Zauważ, że w tej kwerendzie zastosowano widoki z katalogu, a nie widoki DMV.

Listing 12.20. Plik wolneMiejsceDlaGrupyPlikow.sql

```
-- Określanie łącznej wielkości każdej grupy plików.
select data_space_id, (sum(size)*8)/1000 as total_size_MB
into #filegroups
from sys.database_files
group by data_space_id
order by data_space_id

-- Ustalanie, ile miejsca zajęto w każdej grupie plików.
select ds.name, au.data_space_id
, (sum(au.total_pages) * 8)/1000 as Allocated_MB
, (sum(au.used_pages) * 8)/1000 as used_MB
, (sum(au.data_pages) * 8)/1000 as Data_MB
, ((sum(au.total_pages) - sum(au.used_pages) ) * 8 )/1000 as Free_MB
into #Allocations
from sys.allocation_units as au inner join sys.data_spaces as ds
  on au.data_space_id = ds.data_space_id
group by ds.name, au.data_space_id
order by au.data_space_id
-- Łączenie wszystkich informacji.
select f.data_space_id
, a.name
, f.total_size_MB
, a.allocated_MB
, f.total_size_MB - a.allocated_MB as free_in_fg_MB
, a.used_MB
, a.data_MB
, a.Free_MB
from #filegroups as f inner join #allocations as a
```



```
on f.data_space_id = a.data_space_id
order by f.data_space_id
```

```
drop table #allocations
```

```
drop table #filegroups
```

Plan wykonywania i kod aktualnie działających kwerend

Za pomocą kwerendy z listingu 12.21 (plik *planyWykonywanychKwerend.sql*) można znaleźć plan wykonania (w formacie XML) i kod kwerend aktualnie wykonywanych zadań wsadowych z danej sesji. Do wyświetlenia wyników użyj tabeli w programie SQL Server Management Studio. Gdy uzyskasz wyniki, możesz kliknąć odnośnik do planu w formacie XML. Wtedy w programie SQL Server Management Studio otwarty zostanie edytor dokumentów w tym formacie. Jeśli chcesz zobaczyć plan wykonywania kwerendy w postaci graficznej, kliknij odpowiedni odnośnik, a dany plan pojawi się w nowym oknie programu SQL Server Management Studio.

Listing 12.21. Plik *planyWykonywanychKwerend.sql*

```
SELECT
    er.session_id
,es.login_name
,er.request_id
,er.start_time
,QueryPlan_XML = (SELECT query_plan FROM
sys.dm_exec_query_plan(er.plan_handle))
,SQLText = (SELECT Text FROM sys.dm_exec_sql_text(er.sql_handle))
FROM sys.dm_exec_requests er
JOIN sys.dm_exec_sessions es
    ON er.session_id = es.session_id
WHERE es.is_user_process = 1
ORDER BY er.start_time ASC
```

Zajmowana pamięć

Kod z listingu 12.22 (plik *wykorzystPamieci.sql*) informuje, ile pamięci (w kilobajtach) zajmuje każdy wewnętrzny komponent systemu SQL Server.

Listing 12.22. Plik *wykorzystPamieci.sql*

```
SELECT
    name
, type
, pages_kb AS MemoryUsedInKB
FROM sys.dm_os_memory_clerks
```

Zajęta pamięć w puli buforów

Kod z listingu 12.23 (plik *wykorzystPuliBuforow.sql*) wyświetla wszystkie obiekty z puli buforów i ilość pamięci zajmowanej przez każdy z nich. Jest to doskonały sposób na sprawdzenie, które obiekty używają puli buforów.

Listing 12.23. Plik wykorzystujący buforowanie.sql

```

SELECT count(*) AS cached_pages_count
       , name, index_id
FROM sys.dm_os_buffer_descriptors AS bd
     INNER JOIN
     (
       SELECT object_name(object_id) AS name
            , index_id , allocation_unit_id
       FROM sys.allocation_units AS au
            INNER JOIN sys.partitions AS p
              ON au.container_id = p.hobt_id
              AND (au.type = 1 OR au.type = 3)
       UNION ALL
       SELECT object_name(object_id) AS name
            , index_id , allocation_unit_id
       FROM sys.allocation_units AS au
            INNER JOIN sys.partitions AS p
              ON au.container_id = p.partition_id
              AND au.type = 2
     ) AS obj
     ON bd.allocation_unit_id = obj.allocation_unit_id
WHERE database_id = db_id()
GROUP BY name, index_id
ORDER BY cached_pages_count DESC;

```

Monitorowanie dzienników

Innym, często pomijanym aspektem jest monitorowanie różnych plików dzienników. System SQL Server generuje własny dziennik błędów. Ponadto dostępne są też dzienniki zdarzeń systemu Windows: aplikacji, zabezpieczeń i systemu.

W przeszłości do przeglądania dzienników systemów SQL Server i Windows służyły odrębne aplikacje. Dla dzienników systemu Windows używane było narzędzie Podgląd zdarzeń, a dzienniki systemu SQL Server wyświetlane były w edytorze tekstu. Obecnie za pomocą przeglądarki dzienników z programu SQL Server Management Studio można połączyć obie grupy dzienników w jednym widoku. Można to zrobić z poziomu węzłów najwyższego poziomu reprezentujących narzędzia: SQL Server, SQL Server Agent, Database Mail i Windows NT.

Monitorowanie dziennika błędów systemu SQL Server

Dziennik błędów systemu SQL Server to miejsce, w którym system zapisuje wszystkie informacje o błędach, a także wiele dodatkowych danych o wykonywanych zadaniach.

Dziennik błędów to plik tekstowy zapisywany w katalogu *C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Log*. Przy każdym uruchomieniu procesu systemu SQL Server tworzony jest nowy plik dziennika. System SQL Server przechowuje siedem takich plików. Najnowszy nosi nazwę *errorlog*, a najstarszy — *errorlog.6*.

Dziennik zawiera wiele przydatnych informacji. Za każdym razem, gdy wystąpi poważny problem, szukanie dodatkowych informacji powinno rozpocząć właśnie od dziennika błędów systemu SQL Server. Ponadto jeśli potrzebujesz więcej danych, aby rozwiązać określony problem, możesz wykorzystać powiadomienia o zdarzeniach i zdarzenia rozszerzone.

Monitorowanie dzienników zdarzeń systemu Windows

Informacje związane ze zdarzeniami systemu SQL Server mogą zawierać trzy dzienniki zdarzeń systemu Windows:

- dziennik aplikacji,
- dziennik zabezpieczeń,
- dziennik systemu.

W tych dziennikach zdarzeń można szukać dodatkowych informacji o środowisku serwera oraz działających na nim innych procesach i aplikacjach. Ponadto mogą obejmować informacje o procesie systemu SQL Server nierejestrowane w dzienniku błędów tego systemu. Także w wymienionych tu dziennikach warto szukać dodatkowych informacji o problemach z systemem SQL Server.

Standardowe raporty systemu SQL Server

Jednym z najbardziej strzeżonych sekretów dotyczących systemu SQL Server są standardowe raporty, które zaczęto wprowadzać w wersji SQL Server 2005. Od tego czasu w każdej nowej wersji tego systemu rozbudowywano standardowe raporty. Obecnie dostępny jest bogaty zestaw raportów, które zapewniają wiele szczegółowych informacji na temat pracy systemu SQL Server.

Standardowe raporty są dostępne w programie SQL Server Management Studio. Kliknij węzeł serwera prawym przyciskiem myszy, a następnie wybierz opcję *Reports/Standard Reports*, aby zobaczyć listę raportów na temat danego egzemplarza systemu SQL Server. Na rysunku 12.22 przedstawiono listę standardowych raportów dla danego serwera.

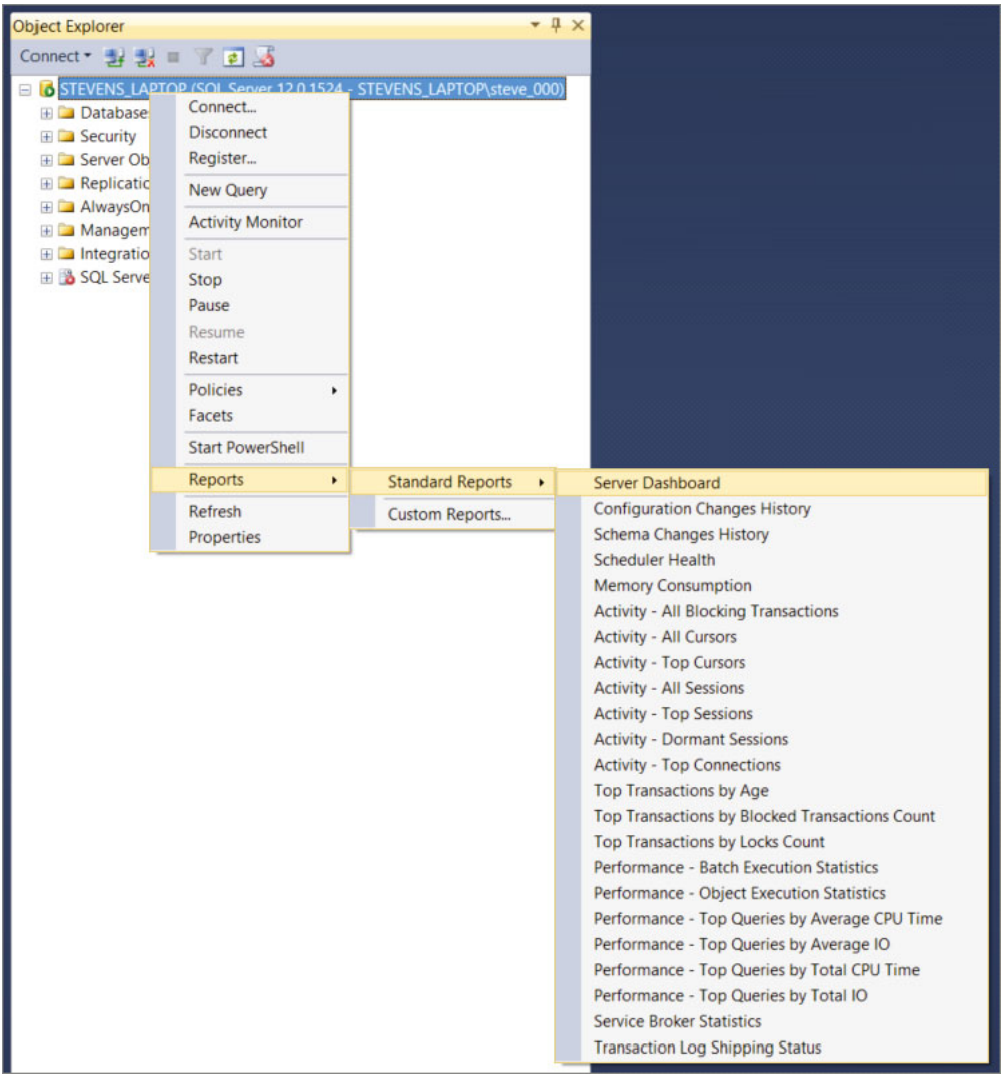
Dla poszczególnych węzłów w oknie *Object Explorer* dostępne są różne raporty. Dla niektórych węzłów nie istnieją raporty standardowe; wtedy widoczna jest tylko opcja *Custom Reports*. W innych węzłach, np. dla poszczególnych baz danych, dostępna jest długa lista standardowych raportów. Na rysunku 12.23 przedstawiamy standardowe raporty dla bazy danych.

Niestety, listy dostępnych raportów nie są nigdzie udokumentowane. Dlatego będziesz musiał sprawdzić różne węzły w oknie *Object Explorer*. W tym celu kliknij wybrany węzeł i wybierz opcję *Reports*, aby zobaczyć dostępne standardowe raporty. Na rysunku 12.24 można zobaczyć jeszcze jeden węzeł, *Security/Logins*, dla którego dostępne są standardowe raporty.

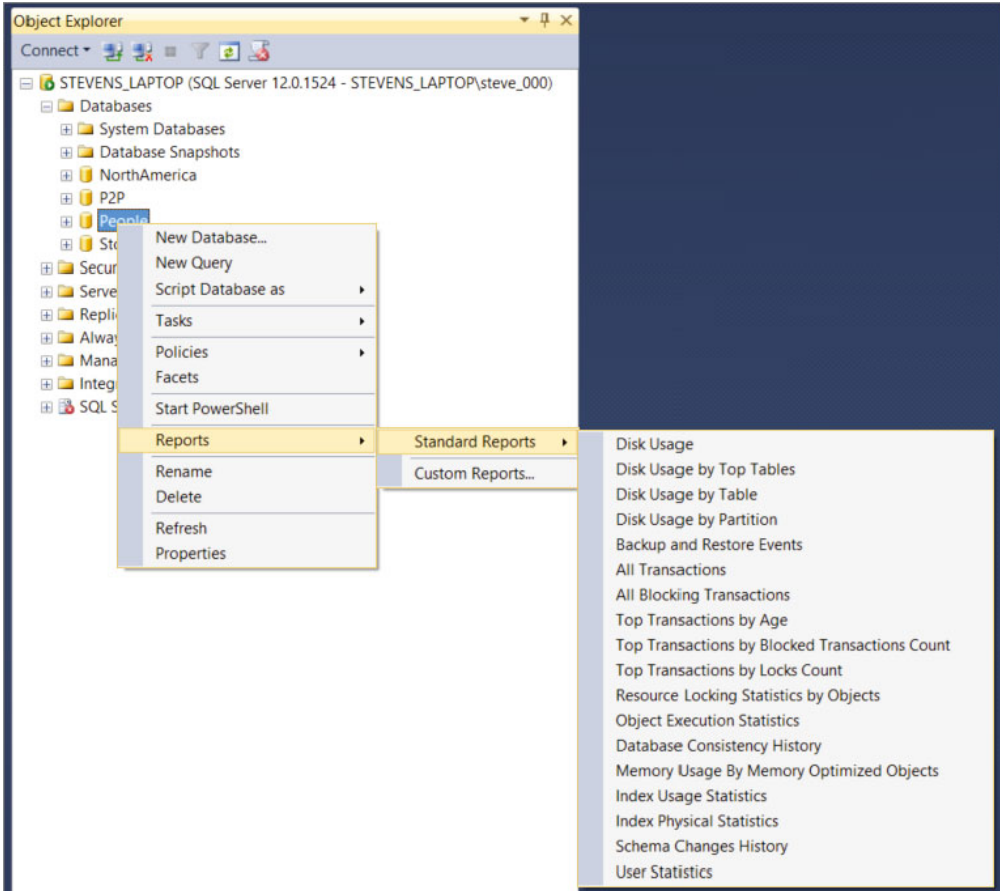
Narzędzie System Center Advisor

System Center Advisor (SCA) to naturalne rozwinięcie narzędzia SQL Server Best Practice Analyzer (BPA). Jednym z problemów z narzędziem BPA jest to, że w każdą jego wersję wbudowany jest zestaw reguł, a wiedza na temat najlepszych praktyk zmienia się szybciej, niż wprowadzane są nowe wydania tego narzędzia. Ponadto BPA trzeba ręcznie uruchamiać na każdym serwerze.

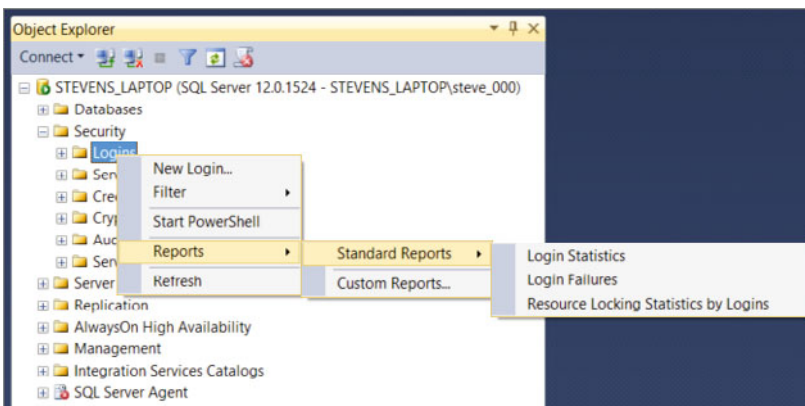
SCA to efekt ciężkiej pracy użytkowników systemu SQL Server i zespołów odpowiedzialnych za ten system w programie PSS (ang. *Product Support Services*). SCA ma być skutecznym narzędziem do oceny konfiguracji systemu SQL Server.



Rysunek 12.22. Lista dostępnych standardowych raportów



Rysunek 12.23. Standardowe raporty dla bazy danych



Rysunek 12.24. Standardowe raporty na temat loginów

SCA to oparte na chmurze narzędzie do monitorowania konfiguracji, które może obserwować dużą liczbę serwerów i na bieżąco informować o prowadzonych analizach. Jedną z zalet wykorzystania chmury jest to, że nowe najlepsze praktyki można szybko wykorzystać w testach poprawności konfiguracji, a użytkownik nie musi w tym celu wykonywać żadnych operacji. Zespół rozwijający narzędzie SCA może łatwo dodać nową regułę, po czym natychmiast będzie ona sprawdzana na wszystkich serwerach monitorowanych przy użyciu tego narzędzia.

Podsumowanie

Regularne monitorowanie pracy systemu SQL Server i zbieranie danych na temat wydajności jest bardzo ważne przy identyfikowaniu problemów z wydajnością. Administratorzy baz danych coraz częściej muszą uwzględniać kilka systemów i zbierać dane z różnych źródeł. Narzędzia i techniki przedstawione w tym rozdziale pomogą administratorom odejść od ręcznych metod do podejścia opartego na zdarzeniach.

Monitor wydajności umożliwia administratorom śledzenie wykorzystania zasobów na serwerze i pomaga rozwiązywać problemy z wydajnością na podstawie danych o używaniu zasobów serwera i systemu SQL Server.

Architektura śledzenia, obejmująca narzędzia SQL Server Profiler i SQL Trace oraz rozproszone odtwarzanie śladu, zapewnia mechanizmy do rejestrowania, analizowania i odtwarzania zdarzeń z systemu SQL Server.

Powiadomienia o zdarzeniach zapewniają mechanizm asynchronicznego wykonywania operacji poza systemem SQL Server w reakcji na zdarzenia zachodzące na serwerze.

Zdarzenia rozszerzone to skalowalny mechanizm do rejestrowania danych w reakcji na wystąpienie określonych zdarzeń w systemie SQL Server. Jest to rozbudowany mechanizm umożliwiający zbieranie złożonych danych pomocnych przy rozwiązywaniu problemów z tym systemem.

Widoki DMV i funkcje DMF systemu SQL Server pozwalają szczegółowo przyjrzeć się pracy tego systemu. W przedstawionych tu przykładach pokazano, jak korzystać z tych komponentów do rozwiązywania konkretnych problemów.

System Center Advisor to oparta na chmurze usługa, która sprawdza zgodność konfiguracji z najlepszymi praktykami i bada, czy zainstalowane są najnowsze poprawki i aktualizacje. To narzędzie pomaga zaktualizować system SQL Server na podstawie najnowszych, najlepszych praktyk.

Po przeczytaniu rozdziału 13. będziesz wiedział, jak dostrajać kod w języku T-SQL.

Dostrajanie wydajności kodu w języku T-SQL

ZAWARTOŚĆ ROZDZIAŁU:

- Omówienie przetwarzania kwerend.
- Czytanie planów wykonywania kwerend.
- Używanie operatorów kwerend.
- Dostrajanie wydajności w środowisku produkcyjnym.

Dostrajanie wydajności w systemie SQL Server to proces analizowania instrukcji w języku T-SQL i identyfikowania kwerend, które nie zwracają wyników w oczekiwanym czasie. Dostrajanie wydajności kodu w języku T-SQL wymaga wiedzy na temat wewnętrznych mechanizmów systemu SQL Server, a także znajomości składni konkretnej kwerendy. Jeśli chcesz pomagać użytkownikom pisać wydajne i szybko przetwarzane instrukcje w języku T-SQL, musisz wiedzieć, jak analizowane są poszczególne kwerendy, a także jak silnik kompilacji systemu SQL Server używa różnych zasobów (np. indeksów, statystyk, planów wykonywania kwerend, tabel danych i kroków kompilacji). Proces dopracowywania takich instrukcji obejmuje pomiar ich wydajności, przegląd planów wykonywania oraz modyfikowanie architektury i samych instrukcji w języku T-SQL w celu zoptymalizowania szybkości działania kwerend. Czasem bardzo prosta zmiana w planie wykonywania kwerendy może prowadzić do znacznego wzrostu wydajności. Jeśli jednak nie wiesz, jak silnik kompilacji z systemu SQL Server przetwarza poszczególne elementy kwerend, proces identyfikowania potrzebnych zmian jest żmudny i frustrujący.

Aby zoptymalizować kwerendę, musisz zrozumieć wpływ każdej instrukcji w języku T-SQL na silnik optymalizacji. Ponadto musisz wiedzieć, jak używać udostępnianych przez system SQL Server narzędzi monitorujących do pomiaru wydajności kwerend. Trzeba też poznać dostępne możliwości z zakresu usprawniania wolnych kwerend.

W tym miejscu dowiesz się, jak kwerendy są przetwarzane, jak zidentyfikować problemy z wydajnością za pomocą narzędzi z systemu SQL Server, a także jak czytać plany wykonywania kwerend, aby wykryć wolny kod.

UWAGA Jednym z głównych usprawnień w wydajności wprowadzonych w systemie SQL Server 2014 jest możliwość przechowywania tabel OLTP w pamięci i używania natywnie kompilowanych procedur składowanych. W tym rozdziale nie koncentrujemy się na tych mechanizmach, ponieważ ich omówienie znajduje się w innym miejscu książki. Techniki dostrajania wydajności przedstawione w tym rozdziale dotyczą zwykłych tabel przechowywanych na dysku, indeksów i procedur składowanych.

Omówienie procesu przetwarzania kwerend

Ten rozdział nie ma być kompletnym omówieniem wewnętrznych mechanizmów silnika kompilacji z systemu SQL Server. Zamiast tego znajdziesz tu przegląd ważnych etapów przetwarzania kwerend. Zrozumienie tych kroków przyda się w trakcie dostrajania wydajności kwerend.

System SQL Server w trakcie przetwarzania kwerendy wykonuje dwa główne kroki, aby wygenerować odpowiedni wynik. Jak można się domyślić, pierwszy krok to kompilacja kwerendy, co prowadzi do wygenerowania planu wykonywania, czyli fizycznych kroków potrzebnych do zwrócenia żądanych informacji. Drugi krok to wykonanie tego planu, co powoduje zwrócenie wyników do programu wywołującego. W tym rozdziale koncentrujemy się na pierwszym kroku, który powoduje wygenerowanie planu wykonywania kwerendy. Dostrajanie wydajności samego wykonywania kwerendy wymaga odpowiedniego ustawienia dysków i systemu przechowywania danych, aby uzyskać maksymalną szybkość operacji wejścia-wyjścia. Tu nie omawiamy tych zadań, ponieważ w każdym systemie wyglądają inaczej.

Etap kompilacji w systemie SQL Server obejmuje następujące kroki: parsowanie, algebraizację i optymalizację. Zespół rozwijający system SQL Server w trakcie prac nad wersją 2012 poświęcił dużo czasu na zmianę architektury i ponowne napisanie kilku elementów procesu kompilacji. Wysiłki te umożliwiły wprowadzenie w wersji SQL Server 2014 dodatkowych usprawnień — lepszego mechanizmu szacowania kardynalności (czyli liczby wierszy) używanego przy optymalizacji opartej na kosztach (opisanego w tym rozdziale) i optymalizacji natywnie kompilowanych procedur składowanych dla tabel przechowywanych w pamięci.

Przyjrzymy się teraz podstawom kompilowania i rekompilowania kwerend. Zaczniemy od omówienia wpływu silnika kompilacji na wydajności i opisanie usprawnień wprowadzonych w systemie SQL Server 2014.

Najpierw przyjrzyj się podstawowym krokom przetwarzania kwerend. Na rysunku 13.1 przedstawiono na ogólnym poziomie, jak instrukcje w języku SQL przechodzą przez silnik kompilacji, są wykonywane, a następnie zwracają wyniki.

Gdy kwerenda w języku T-SQL (może to być instrukcja zwykła lub wsadowa, procedura składowana, wyzwalacz lub inny rodzaj polecenia w tym języku) zostanie wywołana, silnik kompilacji próbuje znaleźć w pamięci podręcznej plan wykonywania tej kwerendy zgodny z wywołanymi instrukcjami. Mechanizm dopasowywania planów z silnika określa, czy istnieje już plan wykonywania dla konkretnej instrukcji zwykłej lub wsadowej. Jeśli potrzebny plan istnieje, jest używany do wykonania danej kwerendy. Pamiętaj, że nie wszystkie tworzone plany



Rysunek 13.1. Proces przetwarzania kwerend

wykonywania kwerend są zapisywane w pamięci podręcznej. Jeśli np. procedura składowana została utworzona z opcją `WITH RECOMPILE`, plan nie trafia do pamięci podręcznej. Dla takiej procedury przy każdym wywołaniu tworzony jest nowy plan. Ponadto nie są zachowywane szacunkowe plany wygenerowane przez narzędzia programistyczne. Gdy pasujący plan wykonywania kwerendy zostanie znaleziony, silnik wykonawczy użyje go przy przetwarzaniu tej kwerendy.

Wiele czynników wpływa na to, czy pasujący plan wykonywania kwerendy zostanie znaleziony i użyty. Przykładowo zmodyfikowane statystyki, dodatkowe indeksy lub opcje `SET` powodują czasem oznaczenie, że plan jest nieprawidłowy, co wymusza jego ponowną kompilację. Zbyt częste lub niepotrzebne rekompilacje mogą wpływać na wydajność całego systemu.

Jeśli plan nie został znaleziony (lub odszukany plan został uznany za niepoprawny), kwerendę trzeba skompilować lub zrekompilować. Kwerenda trafia wtedy do silnika parsującego, który sprawdza poprawność jej składni i tworzy *drzewo parsowania*. Drzewo (nazywane też *drzewem logicznym*) reprezentuje wysokopoziomowe kroki logiczne potrzebne do wykonania całej instrukcji (zwykłej lub wsadowej) w języku T-SQL.

W procesie wiązania drzewo parsowania jest wykorzystywane do sprawdzenia, czy wszystkie obiekty istnieją. Następnie używane nazwy są dopasowywane do identyfikatorów `ObjectID` z katalogu systemu. Potem *drzewo wyrażeń algebraicznych* jest wysyłane do optymalizatora kwerend.

Optymalizator kwerend określa kryteria wydajności wykonywanych instrukcji w języku T-SQL. Ten optymalizator tworzy plany wykonywania kwerend, które określają, jak instrukcje będą wykonywane i w jakiej kolejności. Proces optymalizacji kwerendy obejmuje odwzorowanie zdefiniowanych w procesie wiązania logicznych operatorów na fizyczne operacje przeprowadzane przez silnik wykonawczy. Na ogólnym poziomie silnik optymalizacyjny ustala fizyczne kroki prowadzące do zwrócenia zbioru wyników i kolejność ich wykonywania.

Optymalizator kwerend tworzy kandydujące plany wykonywania, które pozwalają uzyskać oczekiwany efekt końcowy, na podstawie dostępnych informacji ustala koszt każdego planu i wybiera najbardziej ekonomiczny. Optymalizator próbuje ustalić plan o najniższych kosztach, jednak w zależności od liczby możliwości czasem wybiera wystarczająco dobre rozwiązanie. Przyjrzyjmy się teraz bliżej procesowi generowania planów i wybierania jednego z nich.

Generowanie planów wykonywania

Silnik optymalizatora kwerend generuje serię planów wykonywania instrukcji (zwykłych lub wsadowych) w języku T-SQL. Podstawowym zadaniem tego silnika jest przeanalizowanie wszystkich sposobów wykonania kwerendy i ustalenie, z którym z nich związane są najniższe koszty. Nawet proste kwerendy można wykonywać na wiele sposobów, dlatego ważny jest wybór wersji o najniższym koszcie. Pamiętaj, że koszt mierzy się w czasie potrzebnym do zwrócenia wyników użytkownikowi, a także ilości niezbędnych zasobów systemowych. Optymalizator oparty na kosztach teoretycznie powinien generować i analizować wszystkie możliwe plany wykonywania kwerendy, jednak w przypadków skomplikowanych kwerend takich planów są miliony. Dlatego optymalizator poświęca tylko „rozsądną” ilość czasu na ustalenie „rozsądnego” planu; niekoniecznie musi to być najlepszy plan. Jednym ze sposobów dostrajania wydajności systemu jest tworzenie kwerend, które prowadzą do generowania prostych planów wykonywania. Pozwala to ustalić najlepszy plan w „rozsądnym” czasie.

Wygenerowany zoptymalizowany plan wykonywania kwerendy trafia do silnika wykonawczego, gdzie pobierane są dane, po czym plan zostaje przekazany do programu wywołującego. Plan jest zapisywany w pamięci podręcznej, zatem można go później ponownie użyć.

Istnieje kilka sposobów na generowanie planów wykonywania kwerend w celu ich analizowania pod kątem poprawy wydajności. Plany w formie graficznej lub tekstowej można przeglądać w programie SQL Server Management Studio i rejestrować za pomocą zdarzeń rozszerzonych. Ponadto niektóre wskaźniki wydajności kwerend można śledzić przy użyciu widoków DMV i funkcji DMF. Więcej o przeglądaniu i czytaniu planów wykonywania kwerend dowiesz się dalej w tym rozdziale.

Statystyki

Zanim przyjrysz się konkretnym etapom optymalizacji, powinieneś zrozumieć statystyki systemu SQL Server i ich związek z wydajnością kwerend. Na podstawie tych statystyk i zawartości tabel silnik optymalizatora wybiera różne złączenia. Gdy chcesz uzyskać prawidłowe plany wykonywania kwerend, należy zadbać o to, żeby w systemie znajdowały się aktualne statystyki.

System SQL Server zbiera informacje statystyczne na temat rozkładu wartości w kolumnach tabel i widoków indeksowanych. Poziom unikatowości danych w kolumnie określa się za pomocą *kardynalności*. *Wysoką kardynalność* mają kolumny z unikatowymi wartościami, a *niska kardynalność* oznacza, że w kolumnie występuje wiele powtarzających się wartości. Statystyki zbiera się dla pojedynczych kolumn i dla grup kolumn. Każdy obiekt ze statystykami zawiera histogram, za pomocą którego można wyświetlić rozkład wartości w pierwszej kolumnie z listy uwzględnianych w danym obiekcie kolumn.

Optymalizator kwerend na podstawie statystyk określa kardynalność, a tym samym także selektywność wyrażeń. Po wyznaczeniu tych wartości szacowane są rozmiary pośrednich i ostatecznych wyników kwerendy. Dobre statystyki umożliwiają optymalizatorowi precyzyjną ocenę kosztów różnych planów wykonywania kwerend i wybór planu wysokiej jakości.

Wszystkie informacje na temat jednego obiektu ze statystykami są zapisywane w kilku kolumnach jednego wiersza tabeli `sys.indexes`, a także w dużych obiektach binarnych ze statystykami (`statblob`) przechowywanych w *wewnętrznej* tabeli.

Jeśli szacunkowa liczba wierszy z planu wykonywania kwerendy i rzeczywista liczba wierszy znacznie różni się między sobą, należy zacząć od sprawdzenia statystyk dotyczących kolumn używanych w złączeniach i kolumn z klauzuli `WHERE`. Zwróć szczególną uwagę na wewnętrzne złączenia w pętlach. Liczba wierszy powinna być równa szacunkowej liczbie wierszy pomnożonej przez szacunkową liczbę uruchomień pętli.

Jak wcześniej wspomniano, przy dostrajaniu wydajności kwerend ważne jest, aby się upewnić, że statystyki są aktualne. W tym celu można sprawdzić zawartość kolumn `UpdateDate`, `Rows` i `Rows Sampled` z widoku DMV `sys.dm_db_stats_properties`. Statystyki muszą być aktualne, aby uzyskać wysokiej jakości plany wykonywania kwerend zapewniające optymalną wydajność systemu. Aktualne statystyki nie muszą być nowe — ważne jest, aby wiernie odzwierciedlały dane. Jeśli dane się nie zmieniają, statystyki są poprawne w nieskończoność.

Do uzyskania szczegółowych informacji o statystykach można wykorzystać następujące widoki.

- Aby zobaczyć, ile statystyk jest zapisanych w tabeli, sprawdź widok `sys.stats`.
- Aby ustalić, które kolumny są uwzględniane w statystykach, sprawdź widok `sys.stats_columns`.
- Aby zobaczyć histogram i informacje o gęstości danych, zajrzyj do widoku `sys.dm_db_stats_properties`.

Usprawnienia optymalizatora kwerend w systemie SQL Server 2014 — nowy mechanizm szacowania kardynalności

W systemie SQL Server 2014 wprowadzono nowy mechanizm szacowania kardynalności. Ma on bezpośredni wpływ na wydajność kwerend i tworzenie planów ich wykonywania. Szacowana kardynalność to potencjalna liczba wierszy zwracanych przez daną kwerendę. Te szacunki mają bezpośredni wpływ na sposób złączania tabel i typy złączeń używane przy fizycznym przetwarzaniu kwerend.

UWAGA Nowy mechanizm szacowania kardynalności w większości sytuacji nie ma negatywnego wpływu na bieżące kwerendy, jednak Microsoft udostępnił dokument z opisem głównych zmian w tym mechanizmie i ich wpływie na kwerendy. Więcej informacji znajdziesz w dokumencie z dolnej części strony <http://msdn.microsoft.com/en-us/library/dn673537.aspx>.

W tym rozdziale koncentrujemy się na sytuacjach, w których omawiany mechanizm ma wpływ na wydajność kwerendy. Opisujemy też, jak włączać i wyłączać ten mechanizm, aby zachować spójność ze starszymi planami wykonywania kwerend.

Używanie nowego mechanizmu szacowania kardynalności

Nowy mechanizm szacowania kardynalności został napisany pod kątem optymalnego współdziałania z hurtowniami danych i przetwarzaniem OLTP. We wszystkich nowych bazach danych tworzonych w systemie SQL Server 2014 ten mechanizm jest domyślnie włączony (przy założeniu, że dla bazy `model` ustawiony jest poziom zgodności 120).

Jeśli zaktualizowałeś system do wersji SQL Server 2014 i poziom zgodności jest ustawiony na 110, używany będzie mechanizm szacowania kardynalności z systemu SQL Server 2012.

Osoby rozwijające system SQL Server napisały kilka artykułów z wyjaśnieniem, że ich zdaniem nowy mechanizm szacowania kardynalności powinien umożliwić poprawienie wydajności większości kwerend. Jednak zawsze warto sprawdzić działanie kwerend w zmodyfikowanym silniku i zbadać ich wydajność.

Warto uruchomić kwerendy z poziomem zgodności bazy danych ustawionym dla poprzedniej wersji systemu (używanej przed aktualizacją lub przed zainstalowaniem wersji SQL Server 2014). W tym celu wywołaj poniższe polecenie:

```
ALTER DATABASE database_name
SET COMPATIBILITY_LEVEL = 110 | 100)
```

Dla systemu SQL Server 2014 używany jest poziom zgodności 120. Dla wersji SQL Server 2012 jest to poziom 110, a dla systemów SQL Server 2008 i 2008 R2 — poziom 100. System SQL Server 2014 nie obsługuje niższych poziomów zgodności. Jeśli wcześniej używałeś wersji starszej niż SQL Server 2008, powinieneś zaktualizować system do wersji 2008, a następnie przetestować go z poziomem zgodności równym 100.

Zmierz wydajność kwerend dla tego poziomu, zmień poziom zgodności na 120 i ponownie przetestuj kwerendy. Ustal kwerendy, dla których wydajność spadła, a następnie użyj opcji śledzenia 2312, aby dla tych kwerend użyć starszej wersji mechanizmu szacowania kardynalności.

Opcja śladu 2312 umożliwia wykonanie kwerendy z wcześniejszą wersją mechanizmu szacowania kardynalności. Wskazówka QUERYTRACEON pozwala zastąpić nowy mechanizm i wykorzystać jego wersję z systemu SQL Server 2012 przy przetwarzaniu konkretnej kwerendy. Przykładowo instrukcja OPTION (QUERYTRACEON 2312) w końcowej części kwerendy spowoduje, że przy optymalizacji użyty zostanie mechanizm szacowania kardynalności z systemu SQL Server 2012.

Oto wybrane obszary, w których nowy mechanizm szacowania kardynalności działa inaczej niż wcześniejszy:

- niedawno dodane klucze w porządku rosnącym,
- założenie, że filtrowane predykaty dla tej samej tabeli są skorelowane,
- założenie, że filtrowane predykaty dla różnych tabel są niezależne.

Identyfikowanie problemów przy dostrajaniu wydajności kwerend w języku SQL

W trakcie dostrajania wydajności w systemie SQL Server warto wiedzieć, że szybkość wykonywania kwerendy zależy od czasu procesora potrzebnego do jej przetworzenia oraz czasu koniecznego do wczytania danych z dysków i zwrócenia właściwego zbioru wierszy i kolumn. Na serwerze mogą też występować inne problemy z wydajnością (wolny procesor, wolne dyski), jednak w tym rozdziale koncentrujemy się na identyfikowaniu i rozwiązywaniu problemów z wydajnością kwerend w języku SQL-u.

Monitorowanie wydajności kwerend

Istnieje kilka metod identyfikowania i monitorowania problemów z wydajnością procedur składowanych i instrukcji w języku T-SQL w systemie SQL Server. W rozdziale 12., „Monitorowanie systemu SQL Server”, opisano wiele technik z tego obszaru. Często stosowaną metodą jest

uruchomienie jednego z wbudowanych raportów służących do monitorowania kwerend. Na rysunku 13.2 przedstawiono standardowe raporty dostępne dla każdej bazy danych.



Rysunek 13.2. Lista raportów dostępnych dla każdej bazy danych

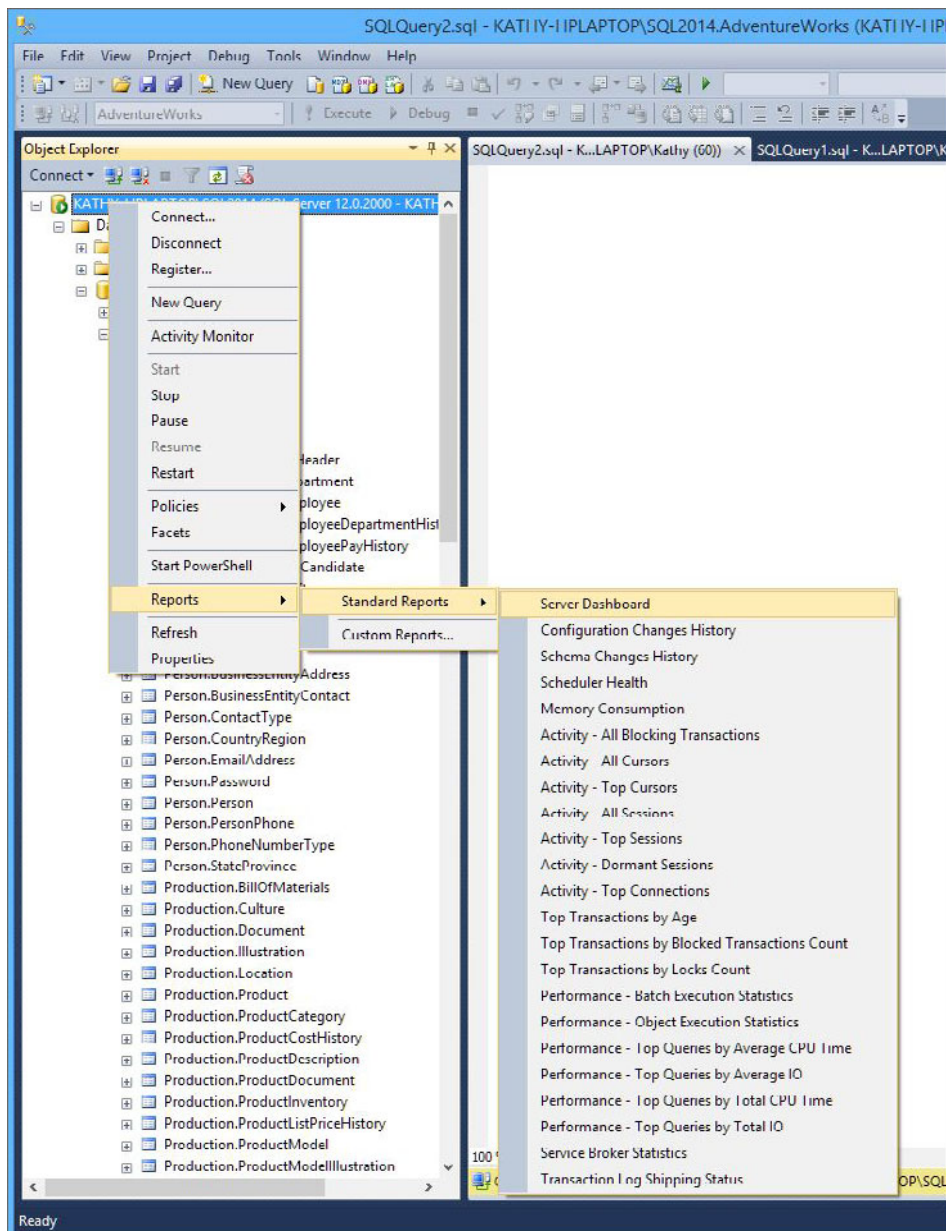
Raporty pomagają administratorom identyfikować długie lub wymagające wielu zasobów kwerendy i procedury składowane. Jest to dobry punkt wyjścia do szukania źródeł problemów z wydajnością systemu. Inny przydatny zbiór raportów to standardowe raporty o wydajności dostępne z poziomu egzemplarza systemu w oknie *Object Explorer*. Tę listę przedstawiono na rysunku 13.3.

Administrator powinien szukać kwerend, które zużywają dużą ilość zasobów systemowych (pamięci, procesora i zasobów wejścia-wyjścia). Jeśli nawet kwerenda działała dobrze na etapie testów, może zostać zrekompilewana bez ostrzeżenia i otrzymać w efekcie nieoptymalny plan wykonywania. Ponadto kwerendy, które używają mało danych, też mogą zostać zrekompilewane i zacząć działać nieoptymalnie. Monitorowanie w czasie rzeczywistym kwerend w środowisku produkcyjnym to najlepszy sposób na zidentyfikowanie problemów, zanim staną się poważne.

Warto pamiętać, że stosunkowo szybkie kwerendy, które są wykonywane wiele razy na sekundę lub minutę, mogą prowadzić do niskiej wydajności, nawet jeśli jedno ich wykonanie zajmuje bardzo mało czasu (mniej niż kilkaset milisekund). Gdy szukasz źródeł niskiej wydajności, sprawdzaj też łączną ilość zasobów systemowych i czasu dla każdej kwerendy.

Co robić po wykryciu kwerendy o niskiej wydajności?

„Kwerenda o niskiej wydajności” to niejednoznaczne określenie. Kwerenda traktowana jako wolna w jednym systemie w innym może zostać uznana za szybką. Czas wykonywania poszczególnych kwerend nie zawsze jest dobrą miarą ich wydajności. Przykładowo kwerenda kończąca pracę po 300 milisekundach wydaje się szybka, jeśli jednak jest wykonywana kilkaset razy na sekundę, może znacznie obciążać procesor. Dlatego pierwszą rzeczą, jaką należy zrobić przy analizowaniu wydajności kwerend, jest sprawdzenie dotyczących ich wskaźników dla jednego wykonania i po zagregowaniu.



Rysunek 13.3. Raporty z poziomu egzemplarza systemu SQL Server

Gdy znajdziesz procedurę składowaną lub instrukcję w języku T-SQL, która zużywa nieoczekiwanie dużo zasobów, w następnym kroku powinieneś dokładniej przyrzeć się wydajności danej kwerendy i przeanalizować używany dla niej plan wykonywania.

Istnieją całe książki poświęcone czytaniu i analizowaniu planów wykonywania kwerend oraz wewnętrznym mechanizmom kompilatora języka SQL. Tu koncentrujemy się na wskazówkach dotyczących rozwiązywania problemów i na podstawowych krokach z zakresu dostrajania wydajności.

Generowanie planów wykonywania kwerend

Pierwszym krokiem przy rozwiązywaniu problemów z kwerendą w języku T-SQL jest wygenerowanie planu wykonywania kwerendy. W trakcie przetwarzania kwerendy logiczne kroki są przekształcane na serię kroków fizycznych prowadzących do pożądaných wyników. Wygenerowanych może zostać wiele planów wykonywania kwerendy, a silnik SQL-a spróbuje wybrać i zastosować najbardziej ekonomiczny z nich. Jednak możliwość precyzyjnego oszacowania kosztów zależy od kilku czynników: dostępności odpowiednich indeksów, zaktualizowanych i prawidłowych statystyk, a także poziomu złożoności kwerendy (wpływa to na to, czy silnik zdoła przeanalizować wystarczająco dużą liczbę planów).

Plan wykonywania kwerend w systemie SQL Server ma różne postacie: tekstową, graficzną i XML-ową. Zapoznanie się z formą graficzną planu pozwala zwykle szybko zidentyfikować problemy, ponieważ w tej postaci widocznych jest wiele wskazówek. Do wyświetlania różnych wersji planów służą zdarzenia z rodziny Showplan silnika przetwarzania kwerend. Poszczególne rodzaje tych zdarzeń zwracają inne informacje. System SQL Server może wyświetlić plan graficzny z wizualnymi wskazówkami dotyczącymi problemów z wydajnością, a także plan tekstowy (jako zwykły tekst lub w formacie XML) zawierający więcej szczegółowych informacji na temat wykonywania danej kwerendy.

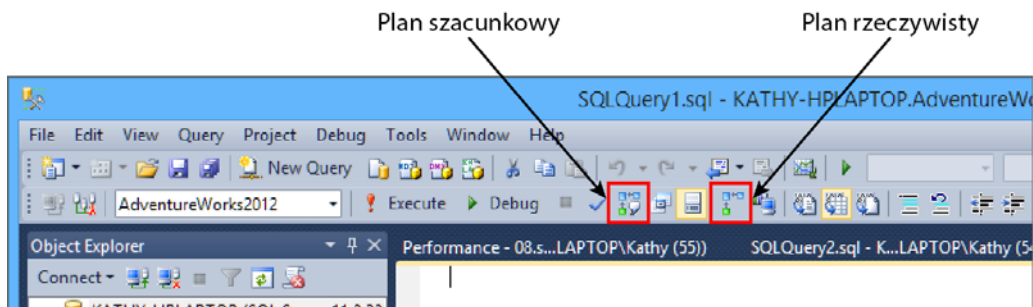
W tabeli 13.1 znajduje się opis różnych formatów wyświetlanych planów. W tym rozdziale używane będą plany graficzne. Gdy opanujesz już czytanie planów wykonywania kwerend, możesz wybrać format tekstowy lub XML-owy.

Tabela 13.1. Formaty wyświetlanych planów

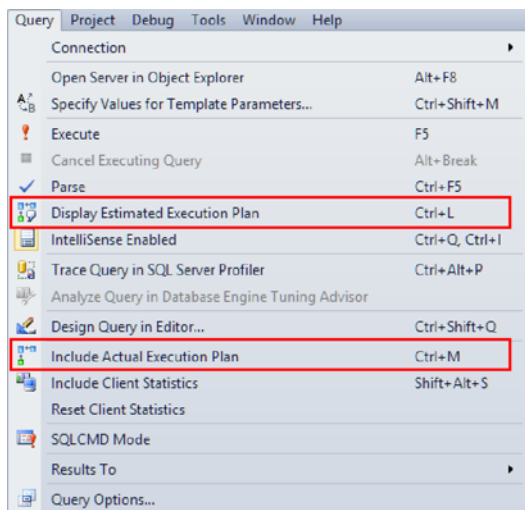
Zawartość planu	Format tekstowy	Format graficzny	Format XML
Operatory	SET SHOWPLAN_TEXT ON	Niedostępny	Niedostępny
Operatory i oszacowany koszt	SET SHOWPLAN_ALL ON	Wyświetla szacunkowy plan wykonywania w programie SQL Server Management Studio	SET SHOWPLAN_XML ON
Operatory, oszacowane kardynalność i koszt, a także informacje o wykonaniu	SET STATISTICS PROFILE ON	Wyświetla rzeczywisty plan wykonywania w programie SQL Server Management Studio	SET STATISTICS XML ON

Są dwa rodzaje planów wykonywania kwerend — *szacunkowe* i *rzeczywiste*. *Szacunkowy plan wykonywania kwerendy* możesz wygenerować przed jej uruchomieniem. Dzięki temu poznasz kroki, które silnik przetwarzający kwerendę prawdopodobnie wykona. Nie zawsze jest to ostateczny plan, jednak zwykle pozwala dobrze oszacować sposób wykonywania danej kwerendy. Aby wyświetlić taki plan, zaznacz odpowiednią instrukcję w programie SQL Server Management Studio, a następnie kliknij przycisk *Estimated Query Plan* na pasku narzędzi (patrz rysunek 13.4).

Szacunkowy plan wykonywania kwerendy pozwala dobrze oszacować działanie kodu, jednak nie jest to prawdziwy plan wykonywania. Na ostateczny plan wpływa wiele czynników, w tym parametry, współbieżność lub nieaktualne statystyki. *Rzeczywisty plan wykonywania kwerendy* zawiera kroki wykonane w trakcie jej przetwarzania. Na rysunku 13.4 zaznaczony jest też przycisk *Actual Query Plan*, służący do wyświetlania takich planów. Oba rodzaje planów można też wyświetlić za pomocą menu *Query* widocznego na rysunku 13.5.



Rysunek 13.4. Generowanie planów wykonywania kwerendy



Rysunek 13.5. Opcje służące do wyświetlania planów wykonywania kwerendy

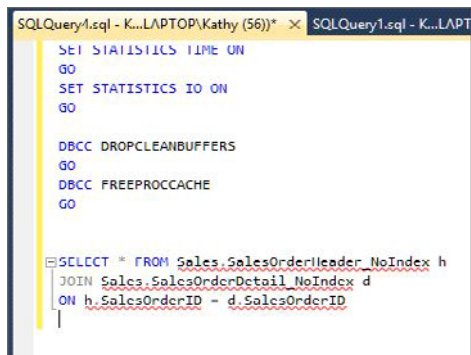
Czytanie planów wykonywania kwerend

Na planie graficznym operacje logiczne i fizyczne są przedstawiane za pomocą ikon, z których każda określa rodzaj wykonywanej operacji.

UWAGA Aby dowiedzieć się czegoś więcej o fizycznych operacjach przedstawionych na graficznym planie wykonywania kwerend, zapoznaj się ze stroną [http://msdn.microsoft.com/en-us/library/ms191158\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms191158(v=sql.120).aspx) z dokumentacji Books Online.

Przy dostrajaniu wydajności należy poszukać konkretnych operacji z nieoczekiwanie dużą liczbą odczytów lub zapisów zajmujących zasoby systemowe. W wielu rodzajach kwerend duża liczba odczytów jest czymś normalnym. Jednak niepotrzebne odczyty mogą spowolnić wykonywanie kwerend. Jeśli np. tabelę przed odczytem trzeba posortować, może to prowadzić do dużej liczby odczytów. W takiej sytuacji dodanie indeksów pozwala wyeliminować sortowanie i znacznie ograniczyć liczbę odczytów przy przetwarzaniu kwerendy.

Załóżmy, że w systemie procesor pracuje pod dużym obciążeniem, a administrator bazy danych zidentyfikował kwerendę, która zajmuje dużą liczbę cykli procesora. W przykładowej kwerendzie łączane są dwie tabele, które jednak nie obejmują indeksów, co pozwoliłoby na wydajne wykonanie zadania. Aby precyzyjnie zmierzyć wydajność, należy przed wykonaniem kwerendy opróżnić pamięć podręczną. Wymusza to zarówno rekompilację procedury składowanej, jak i opróżnienie pamięci podręcznej z danymi. W przykładzie włączone jest zbieranie statystyk związanych z czasem i operacjami wejścia-wyjścia, co pomaga zidentyfikować problemy. Na rysunku 13.6 przedstawiono instrukcje używane do testowania kwerendy na tabelach bez indeksów.



```

SET STATISTICS TIME ON
GO
SET STATISTICS IO ON
GO

DBCC DROPCLEANBUFFERS
GO
DBCC FREEPROCCACHE
GO

SELECT * FROM Sales.SalesOrderHeader NoIndex h
JOIN Sales.SalesOrderDetail NoIndex d
ON h.SalesOrderID = d.SalesOrderID

```

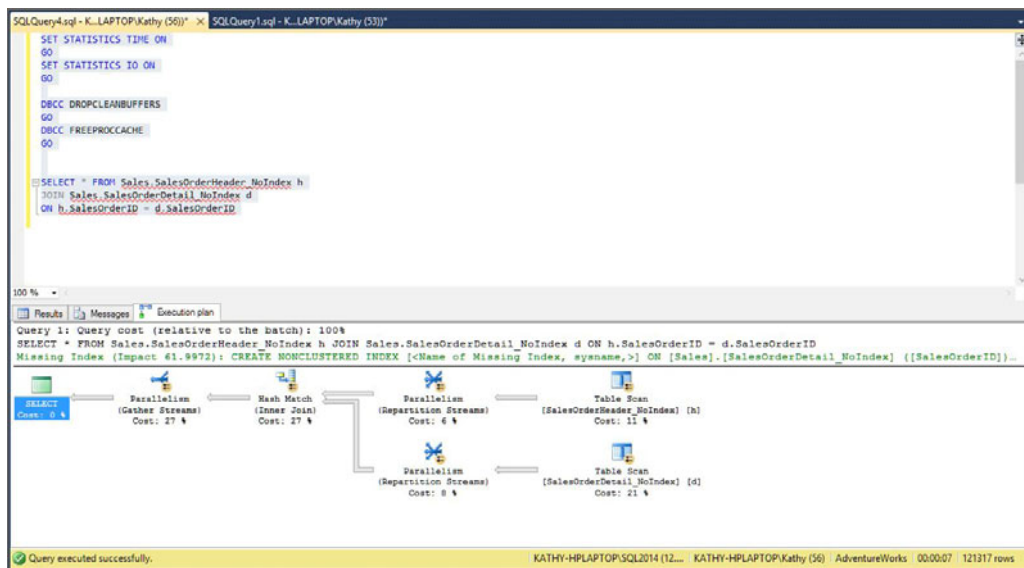
Rysunek 13.6. Instrukcje do testowania wydajności pierwotnej wersji kwerendy

Przykładowa kwerenda działa na kopiach dwóch tabel z bazy AdventureWorks. Pierwotna wersja używa dwóch tabel bez indeksów i kluczy głównych. Przy wykonywaniu tego prostego złączenia generowany jest plan wykonywania kwerendy prowadzący do uzyskania wyników złączenia.

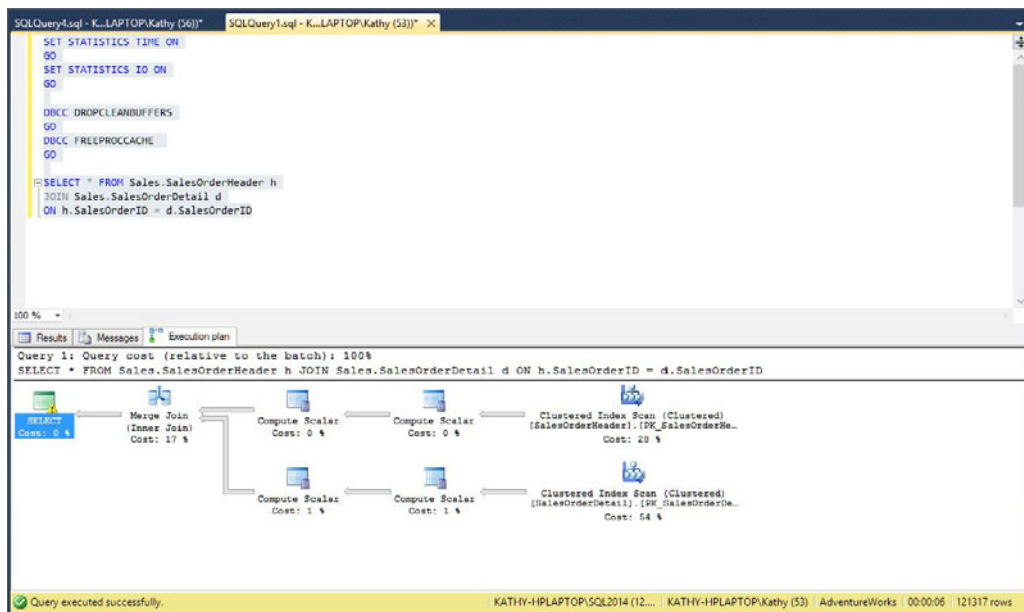
Na rysunku 13.7 przedstawiono plan wykonywania pokazanej wcześniej przykładowej kwerendy. Złączenie z haszowaniem (opisane dalej w tym rozdziale) jest używane do dopasowania wierszy z jednej tabeli do wierszy z drugiej tabeli. Graficzny plan wykonywania kwerendy pokazuje, że większość kosztów związana jest z krokami przetwarzania, a nie z krokami odczytu tabel. Przyjrzyj się planowi z rysunku 13.7 — dwie operacje skanowania tabeli odpowiadają za 32% kosztów, natomiast wszystkie pozostałe kroki przetwarzania (złączenie z haszowaniem i operacje współbieżne) generują pozostałe 68% kosztów. Jeśli źródłem zatorów w systemie jest procesor, a mamy na celu ograniczenie jego obciążenia, zastosowanie wydajniejszej kwerendy z niższymi kosztami kroków przetwarzania pozwoli zwiększyć wydajność.

Generator planów wykonywania kwerend wykrywa, że indeks w przynajmniej jednej z używanych tabel pozwoli przyspieszyć przetwarzanie. Dlatego pojawia się taka rekomendacja. Po dodaniu właściwych indeksów (w tym zalecanego indeksu) ta sama kwerenda powinna w mniejszym stopniu obciążać procesor, co widać na rysunku 13.8. Generator wykrywa też potencjalne problemy z niejawną konwersją danych w końcowej instrukcji SELECT, jednak tą kwestią można się zająć po dodaniu indeksów.

Teraz tę samą kwerendę można wywołać dla tabel z poprawnymi indeksami. Nowy plan wykonywania kwerendy pokazuje, że 82% kosztów generują operacje odczytu. Kroki przetwarzania są teraz mniej kosztowne — zostały zredukowane do 18%. Wczytana została ta sama liczba wierszy (w wyniku skanowania tabel), jednak czas przetwarzania znacznie się skrócił i spadło obciążenie procesora. Ponieważ na początku założyliśmy, że w przykładowym systemie procesor jest źródłem zatorów, opisane usprawnienie pozwala zwiększyć wydajność całego systemu, choć łączny czas wykonywania kwerendy prawie się nie zmienił.



Rysunek 13.7. Graficzny plan wykonywania kwerendy



Rysunek 13.8. Plan wykonywania kwerendy dla tabel po dodaniu indeksów

Statystyki dotyczące operacji wejścia-wyjścia i czasu pracy dla każdej z tych kwerend także pokazują poprawę wydajności pracy procesora. Oto wyniki dla pierwszej kwerendy, działającej dla tabel bez indeksów:

(121317 row(s) affected)

Table 'SalesOrderHeader_NoIndex'. Scan count 9, logical reads 780, physical reads 0, read-ahead reads 780, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

Table 'SalesOrderDetail_NoIndex'. Scan count 9, logical reads 1494, physical reads 0, read-ahead reads 1494, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

(1 row(s) affected)

SQL Server Execution Times:

CPU time = 2375 ms, elapsed time = 6762 ms.

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 0 ms.

A oto wyniki dla drugiej kwerendy, działającej dla tabel z poprawnymi indeksami:

(121317 row(s)affected)

Table 'SalesOrderDetail'. Scan count 1, logical reads 1246, physical reads 3, readahead reads 1277, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0. Table 'SalesOrderHeader'. Scan count 1, logical reads 689, physical reads 2, read-ahead reads 685, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

(1 row(s)affected)

SQL Server Execution Times:

CPU time = 1469 ms, elapsed time = 6638 ms.

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 0 ms.

Choć zmiana w łącznym czasie wykonywania nie jest duża (6638 milisekund w porównaniu z 6762 milisekundami), czas używania procesora znacznie się skrócił (1469 milisekund w porównaniu z 2375 milisekundami), co pozwala ograniczyć zator w procesorze.

Przy dostrajaniu wydajności wolno działającej kwerendy plan jej wykonywania oraz raporty Statistics IO i Statistics Time pomagają określić, gdzie można uzyskać poprawę wydajności. Pozwalają też sprawdzić i zmierzyć usprawnienia.

Operatory dostępu do danych w planach wykonywania kwerend

Graficzny plan wykonywania kwerendy wyświetla różne rodzaje operatorów dostępu używanych przez system SQL Server przy odczycie danych. W tym podrozdziale opisujemy różnice między poszczególnymi operatorami i wpływ metod dostępu na wydajność kwerendy.

Podrozdział nie jest kompletnym omówieniem wszystkich elementów z planów wykonywania kwerend. Opisujemy tu najczęściej używane operatory oraz pokazujemy, jak wykorzystać je w ramach dostrajania wydajności kwerend.

Aby wykonać przykłady z następnych punktów, odtwórz kopię bazy AdventureWorks pod nazwą AW_2, co pozwoli usunąć i ponownie dodać indeksy oraz porównać wyniki dla tabel bez indeksów i z nimi.

Skanowanie tabeli

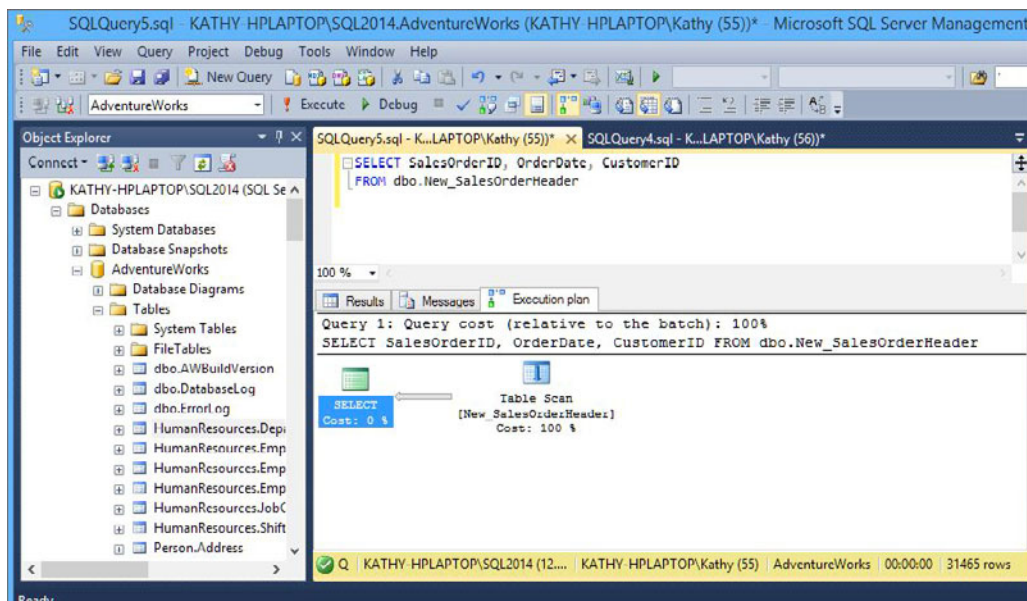
Skanowanie tabeli obejmuje sekwencyjne skanowanie wszystkich stron danych z określonej tabeli. Wywołaj w bazie AW_2 poniższy skrypt:

```
SELECT * INTO dbo.New_SalesOrderHeader
FROM Sales.SalesOrderHeader
```

Po wykonaniu tego skryptu (tworzy on kopię tabeli SalesOrderHeader) wywołaj poniższy skrypt. Pamiętaj, aby kliknąć przycisk *Graphical Query Plan* w górnej części okna z kwerendą, aby wyświetlić plan jej wykonywania. Jeśli wolisz, możesz też za pomocą instrukcji SET STATISTICS PROFILE ON wyświetlić rzeczywisty plan w formie tekstowej.

```
SELECT SalesOrderID, OrderDate, CustomerID
FROM dbo.New_SalesOrderHeader
```

Ponieważ używana jest tabela w formie sterty, instrukcja powoduje skanowanie tabeli. Na rysunku 13.9 przedstawiono graficzny plan wykonywania kwerendy.



Rysunek 13.9. Graficzny plan wykonywania kwerendy

Teraz przyjrzyj się danym wyjściowym dla przykładowej kwerendy przedstawionej na listingu 13.1 (plik *dostępDoIndeksu1.sql*) przy ustawieniu STATISTICS ON. Jest to ustawienie z poziomu sesji generujące statystyki operacji wejścia-wyjścia dla kwerendy, która spowodowała skanowanie tabeli.

Listing 13.1. Plik *dostępDoIndeksu1.sql*

```
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO
SET STATISTICS PROFILE ON
GO
```

```

SELECT SalesOrderID, OrderDate, CustomerID
FROM dbo.New_SalesOrderHeader
GO
SET STATISTICS IO OFF
GO
SET STATISTICS PROFILE OFF
GO

```

Oto uzyskane statystyki operacji wejścia-wyjścia:

```

(31465 row(s) affected)
Table 'New_SalesOrderHeader'. Scan count 1, logical reads 780, physical reads 0,
read-ahead reads 780, lob logical reads 0, lob physical reads 0, lob read-ahead
reads 0.

```

Wartość `Scan count` informuje, ile razy w kwerendzie uzyskano dostęp do tabeli. Jeśli kwerenda dotyczy wielu tabel, pojawią się statystyki operacji wejścia-wyjścia dla każdej z tych tabel. Tu kwerenda raz uzyskała dostęp do tabeli `New_SalesOrderHeader`.

Licznik `logical reads` informuje, ile stron wczytano z pamięci podręcznej danych. Tu operację wykonano 780 razy. Na Twoim komputerze ta wartość może być nieco inna. Ponieważ w tym przykładzie skanowana jest cała tabela, wartość licznika `logical reads` jest równa liczbie stron zaalokowanych dla tej tabeli.

Aby sprawdzić liczbę stron zaalokowanych dla tabeli, można też uruchomić poniższą kwerendę:

```

select in_row_reserved_page_count
from sys.dm_db_partition_stats
WHERE OBJECT_ID = OBJECT_ID('New_SalesOrderHeader')

```

Licznik `physical reads` zwraca liczbę stron wczytanych z dysku. Dla przykładowego kodu ta wartość to 0. Nie oznacza to jednak, że nie nastąpił fizyczny odczyt danych z dysku.

Licznik `read-ahead reads` określa liczbę stron z fizycznego dysku umieszczonych w wewnętrznej pamięci podręcznej danych. Te strony są przenoszone do pamięci podręcznej, gdy system SQL Server uzna, że będzie ich później potrzebował przy przetwarzaniu kwerendy. Tu wartość tego licznika to 780. Jest to łączna liczba fizycznych odczytów. Liczniki `physical reads` i `read-ahead reads` określają poziom aktywności dysku fizycznego. Także te wartości na Twoim komputerze mogą być inne.

Liczniki `lob logical reads`, `lob physical reads` i `lob read-ahead reads` działają podobnie jak wcześniej opisane liczniki, ale dotyczą dużych obiektów (np. kolumn o typach `varchar(max)`, `nvarchar(max)`, `xml` lub `varbinary(max)`). Gdy instrukcje w języku T-SQL pobierają kolumny typów `lob`, niektóre operacje wymagają wielokrotnego przechodzenia po drzewie danych takich typów. Dlatego ustawienie `SET STATISTICS IO` może zwracać wyższą od oczekiwanej liczbę odczytów logicznych.

Skanowanie indeksów klastrowanych

Na potrzeby następnego ćwiczenia dodaj do tabeli indeks klastrowany, aby zobaczyć jego wpływ na kwerendę. Indeks klastrowany powoduje przechowywanie wszystkich danych z tabeli na poziomie liści. Dane są wtedy posortowane według *klucza indeksu klastrowanego*.

UWAGA Indeks klastrowany nie jest kopią danych tabeli — można powiedzieć, że taki indeks jest danymi.

Teraz uruchom skrypt przedstawiony na listingu 13.2 (plik *dostepDoIndeksu2.sql*), aby zobaczyć efekty dodania indeksu klastrowanego.

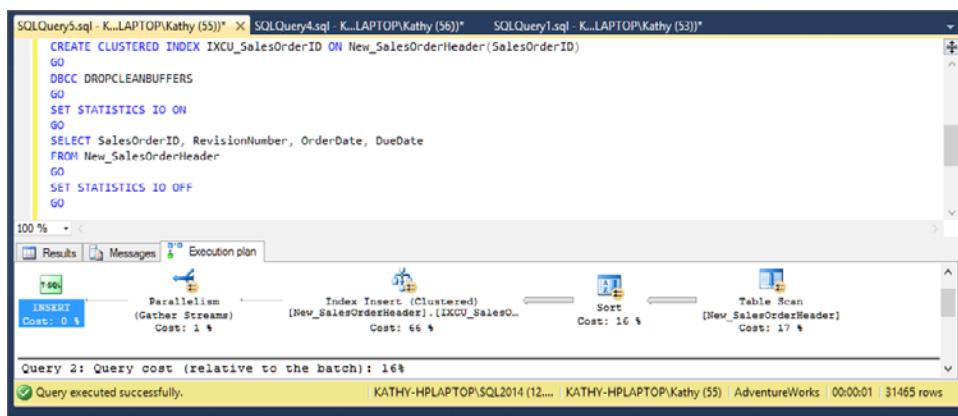
Listing 13.2. Plik *dostepDoIndeksu2.sql*

```
CREATE CLUSTERED INDEX IXCU_SalesOrderID ON New_SalesOrderHeader(SalesOrderID)
GO
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO
SELECT SalesOrderID, RevisionNumber, OrderDate, DueDate
FROM New_SalesOrderHeader
GO
SET STATISTICS IO OFF
GO
```

Poniżej pokazane są wyniki zwrócone dzięki ustawieniu `STATISTICS IO`. Plan wykonywania kwerendy przedstawiono na rysunku 13.10.

(31465 row(s) affected)

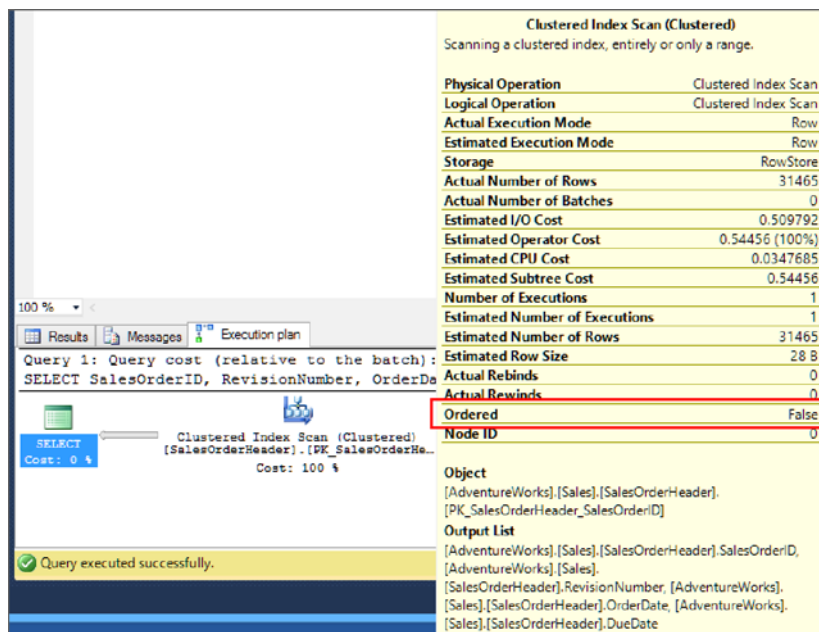
Table 'New_SalesOrderHeader'. Scan count 1, logical reads 799, physical reads 0, read-ahead reads 788, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.



Rysunek 13.10. Plan wykonywania kwerendy dla tabeli z indeksem klastrowanym

Dane wyjściowe uzyskane za pomocą ustawienia `STATISTICS IO` pokazują, że liczba odczytów logicznych (`logical reads`) to 799. Jest ona nieco wyższa niż przy skanowaniu tabeli zapisanej w postaci sterty, ponieważ wielkość wierszy w indeksie klastrowanym jest nieco większa. Wiersze indeksu klastrowanego obejmują unikatowe wartości kluczy oraz wskaźniki do poprzedniego i następnego wiersza. Tu w procesie skanowania nie były używane wskaźniki z wierszy — operacja wczytywała wszystkie strony bez sprawdzania ich kolejności. Możesz się o tym przekonać za pomocą interfejsu graficznego. Aby wyświetlić szczegółowe informacje o skanowaniu tabeli, umieść kursor myszy nad ikoną *Clustered Index Scan*, co ilustruje rysunek 13.11.

Ze szczegółowych informacji o skanowaniu wynika, że indeks nie był uporządkowany (`ORDERED = FALSE`). Oznacza to, że do wczytywania stron nie była używana lista powiązana, a strony z poziomu liści zachowały logiczną kolejność danych. Dlatego przeprowadzona operacja była podobna do zwykłego skanowania tabeli.



Rysunek 13.11. Szczegółowe informacje o skanowaniu tabeli z indeksem klastrowanym

Skanowanie uporządkowanego indeksu klastrowanego także jest pełne, jednak dane są zwracane w kolejności wyznaczanej przez klucz klastrowania. Strony są wtedy wczytywane na podstawie listy powiązanej, która prowadzi od jednego wiersza do następnego zgodnie z kolejnością sortowania.

Tym razem uruchom kwerendę z listingu 13.3 (plik *dostepDoIndeksu3.sql*). W tej kwerendzie używana jest ta sama tabela, co wcześniej, przy czym tabela jest uporządkowana według kolumny SalesOrderID (jest to kolumna klucza indeksu klastrowanego).

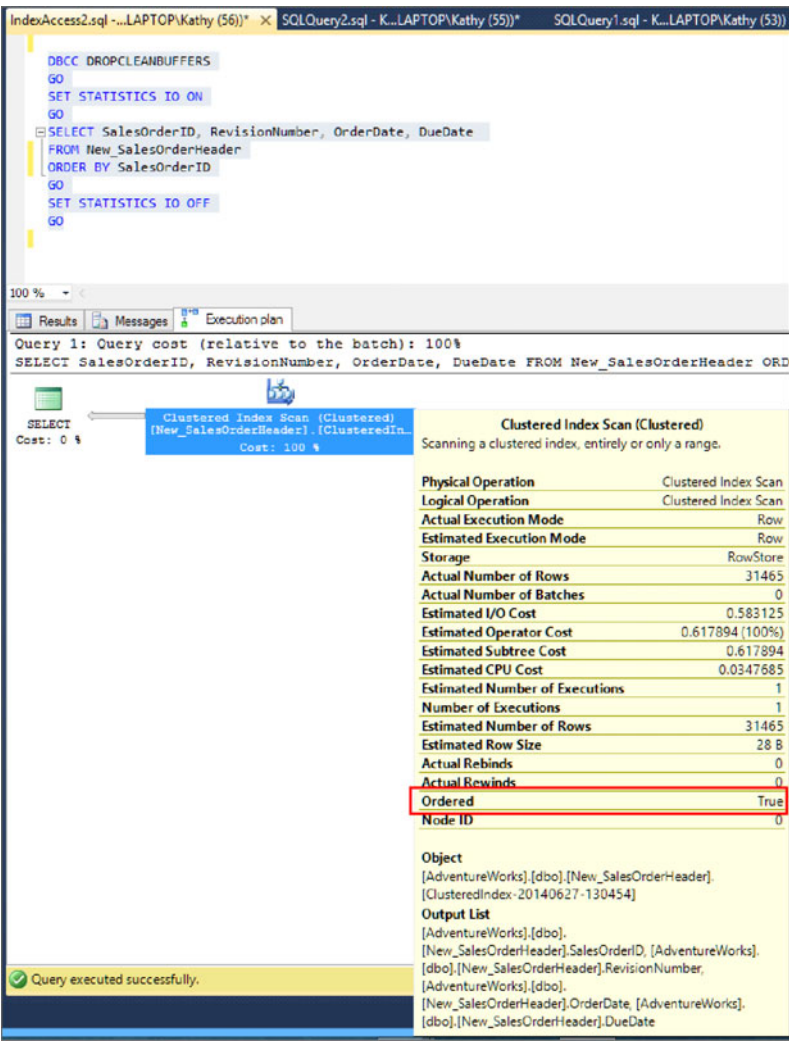
Listing 13.3. Plik *dostepDoIndeksu3.sql*

```
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO
SELECT SalesOrderID, RevisionNumber, OrderDate, DueDate
FROM New_SalesOrderHeader
ORDER BY SalesOrderID
GO
SET STATISTICS IO OFF
GO
```

Oto informacje uzyskane dzięki ustawieniu STATISTICS IO:

```
(31465 row(s) affected)
Table 'New_SalesOrderHeader'. Scan count 1, logical reads 799, physical reads 0,
read-ahead reads 788, lob logical reads 0, lob physical reads 0, lob read-ahead
reads 0.
```

Plan wykonywania kwerendy przedstawiony jest na rysunku 13.12.



Rysunek 13.12. Plan wykonywania kwerendy używającej tabeli z uporządkowanym indeksem

Plan wykonywania kwerendy na rysunku 13.12 wygląda tak samo jak na rysunku 13.11, jednak tu właściwość `Ordered` ma wartość `True`. Także informacje uzyskane dzięki ustawieniu `STATISTICS IO` są takie same jak przy skanowaniu nieuporządkowanego indeksu klastrowanego. Wydajność skanowania uporządkowanego indeksu klastrowanego (inaczej niż w przypadku indeksu nieuporządkowanego) zależy od poziomu fragmentacji pliku. W systemach z dyskami obrotowymi strony, które nie są fizycznie zapisane obok siebie, spowalniają proces odczytu sekwencyjnego.

Skanowanie indeksów nieklastrowanych

Skanowanie indeksu nieklastrowanego ma miejsce, gdy indeks nieklastrowany może zwrócić wszystkie potrzebne pola bez odczytu stron z danymi. Każdy indeks nieklastrowany obejmuje pola klucza klastrowania (lub identyfikator wierszy dla tabel w postaci sterty), pola indeksu

nieklastrowanego (na potrzeby sortowania) i pola dodane w klauzuli `INCLUDE`, które nie wpływają na sortowanie. Jeśli kwerendę można wykonać za pomocą indeksu nieklastrowanego, dany indeks jest *indeksem pokrywającym*. Indeks pokrywający zapewnia zwykle wyższą wydajność niż indeks klastrowany, ponieważ jego wiersze są mniejsze (bo nie zawierają wszystkich pól tabeli). Indeks może być pokrywający nawet wtedy, gdy nie zawiera dodatkowych pól, jednak musi wtedy umożliwiać zwrócenie pól potrzebnych w kwerendzie.

Dobrym przykładem są kwerendy `COUNT(*)`. Dla takich kwerend optymalizator prawie zawsze skanuje indeks zamiast tabeli z danymi, dzięki czemu ilość wczytywanych danych (czyli liczba stron) jest mniejsza. Kwerenda zwraca wtedy łączną liczbę wierszy na podstawie zliczania wierszy z indeksu zamiast wierszy z tabeli.

Poniższa kwerenda tworzy dla tabeli `New_SalesOrderHeader` indeks nieklastrowany, który można wykorzystać w kwerendach. Uruchom kod z listingu 13.4 (plik *dostepDoIndeksu4.sql*) w bazie danych, a następnie przejrzyj dane wyjściowe uzyskane za pomocą ustawienia `STATISTICS IO` i plan wykonywania kwerendy.

Listing 13.4. Plik `dostepDoIndeksu4.sql`

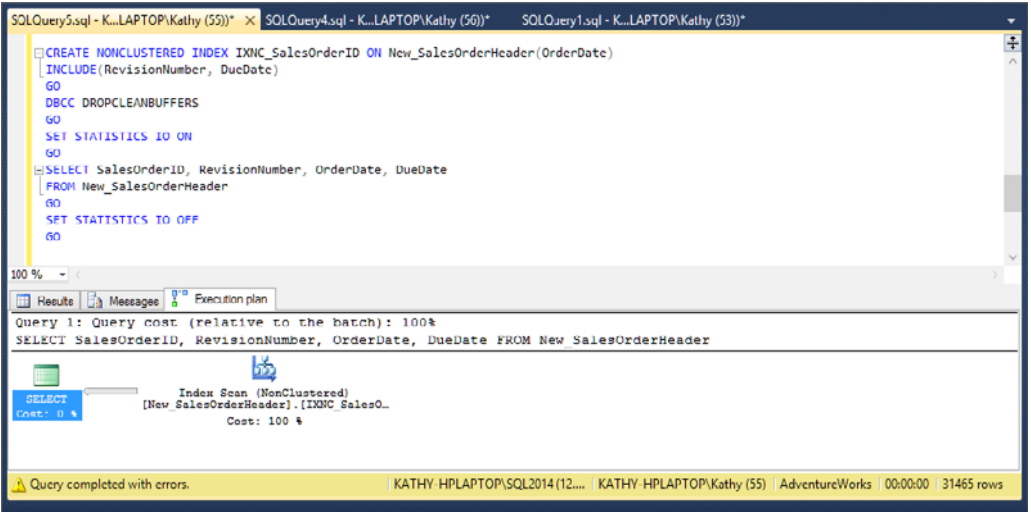
```
CREATE NONCLUSTERED INDEX IXNC_SalesOrderID ON New_SalesOrderHeader(OrderDate)
INCLUDE(RevisionNumber, DueDate)
GO
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO
SELECT SalesOrderID, RevisionNumber, OrderDate, DueDate
FROM New_SalesOrderHeader
GO
SET STATISTICS IO OFF
GO
```

Skrypt tworzy nieklastrowany indeks oparty na kolumnie `OrderDate`. Klucz indeksu klastrowanego znajduje się w każdym wierszu indeksu, a klauzula `INCLUDE` powoduje zapisanie w indeksie także kolumn `RevisionNumber` i `DueDate`. Te kolumny są dołączane, ponieważ kwerenda ich wymaga. Optymalizator korzysta z tego indeksu przy przetwarzaniu kwerendy, ponieważ wszystkie potrzebne pola są zapisane w tym indeksie. Oto informacje uzyskane przy użyciu ustawienia `STATISTICS IO`:

```
(31465 row(s) affected)
Table 'New_SalesOrderHeader'. Scan count 1, logical reads 108, physical reads 1,
read-ahead reads 106, lob logical reads 0, lob physical reads 0, lob read-ahead
reads 0.
```

Plan wykonywania kwerendy przedstawiony jest na rysunku 13.13.

Wyniki uzyskane dzięki ustawieniu `STATISTICS ON` pokazują, że przy wykonywaniu kwerendy potrzebnych było tylko 108 operacji `logical reads`. Wyniki zwrócone przez kwerendy wykorzystujące indeks klastrowany i pokrywający indeks nieklastrowany są identyczne (obejmują tyle samo kolumn i wierszy), jednak indeks klastrowany wymagał 799 operacji `logical reads`, a przy skanowaniu indeksu nieklastrowanego tych operacji było tylko 108. Wynika to z tego, że używany indeks nieklastrowany pokrywa kwerendę i udostępnia dane zapisane na poziomie liści.

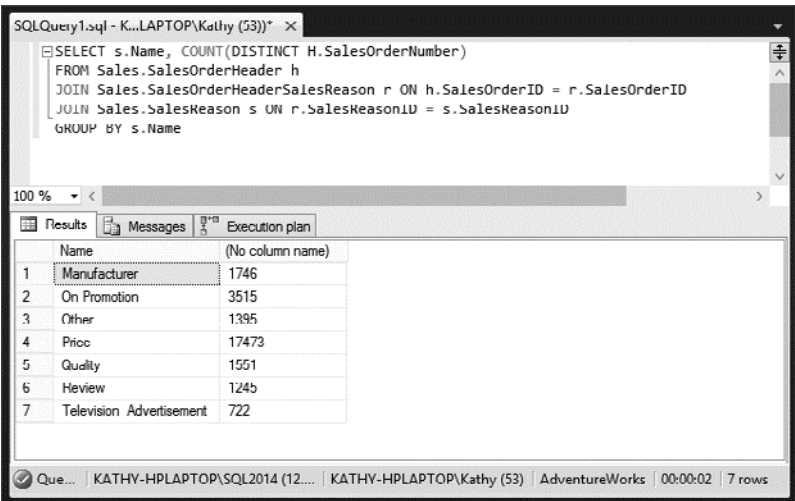


Rysunek 13.13. Plan wykonywania kwerendy

Skanowanie indeksu nieklastrowanego (podobnie jak indeksu klastrowanego) może przebiegać w uporządkowany lub nieuporządkowany sposób. Aby ustalić, czy kwerenda jest uporządkowana, umieść kursor myszy nad ikoną *NonClustered Index Scan* w graficznym planie i sprawdź wartość opcji *ORDERED*.

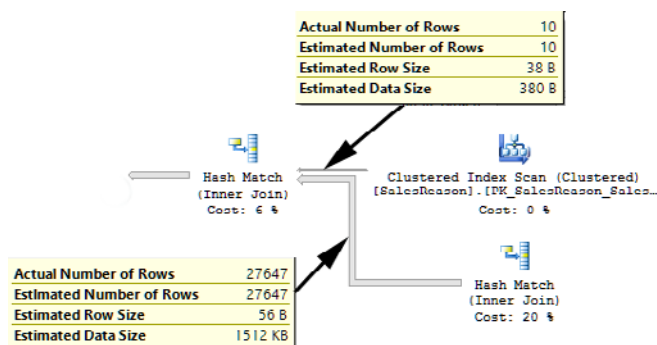
Łączenie operatorów dostępu

W trakcie czytania planów wykonywania kwerend jedną z wizualnych wskazówek udostępnianych w celu ułatwienia rozwiązywania problemów z wydajnością są strzałki łączące operatory dostępu do danych. Wielkość strzałki określa liczbę wierszy przetwarzanych między poszczególnymi krokami. Na rysunku 13.14 przedstawiamy kwerendę agregującą działającą na tabelach z bazy AdventureWorks. Kwerenda zwraca siedem wierszy danych.



Rysunek 13.14. Przykładowa kwerenda

Plan wykonywania tej instrukcji pokazuje, że w trakcie przetwarzania kwerendy trzeba wczytać dużo danych, aby uzyskać siedem końcowych wierszy. Ten fragment kwerendy ilustruje rysunek 13.15.



Rysunek 13.15. W trakcie wykonywania kwerendy wczytywanych jest dużo danych

Ta część planu wykonywania kwerendy pozwala łatwo zauważyć miejsca wskazujące na przetwarzanie dużej liczby wierszy, podczas gdy w rzeczywistości ta liczba jest mniejsza. Gdy umieścisz kursor nad strzałką łączącą operatory, zobaczysz szacunkową i rzeczywistą liczbę wierszy przesyłanych wzdłuż danej strzałki. W trakcie rozwiązywania problemów warto przyjrzeć się różnicy między rzeczywistą liczbą przetwarzanych wierszy a ich szacunkową liczbą. Jeśli ta różnica jest duża, oznacza to, że używane statystyki są nieaktualne.

Operatory złączeń

Do wersji SQL Server 7.0 używany był tylko jeden algorytm złączania — pętla zagnieżdżona. Od wersji 7.0 system SQL Server obsługuje też *złączanie z haszowaniem* i *złączanie przez scalanie*. W tym podrozdziale opisujemy wszystkie te podejścia i wyjaśniamy, w jakich warunkach poszczególne metody są najbardziej wydajne.

Pętla zagnieżdżona (złączanie w pętli)

Przy złączaniu w pętli zagnieżdżonej (inaczej za pomocą *iteracji zagnieżdżonych* lub *złączania w pętli*) jedna strona złączenia jest używana jako zewnętrzna tabela wejściowa (widoczna jako górne dane wyjściowe na graficznym planie wykonywania na rysunku 13.16), natomiast druga strona służy jako wewnętrzna (dolna) tabela wejściowa. Pętla zewnętrzna używa zewnętrznej tabeli wejściowej wiersz po wierszu. Pętla wewnętrzna, wykonywana raz dla każdego wiersza zewnętrznego, wyszukuje pasujące wiersze w wewnętrznej tabeli wejściowej. Listing 13.5 (plik *zlaczenie.sql*) to przykład złączania w pętli zagnieżdżonej.

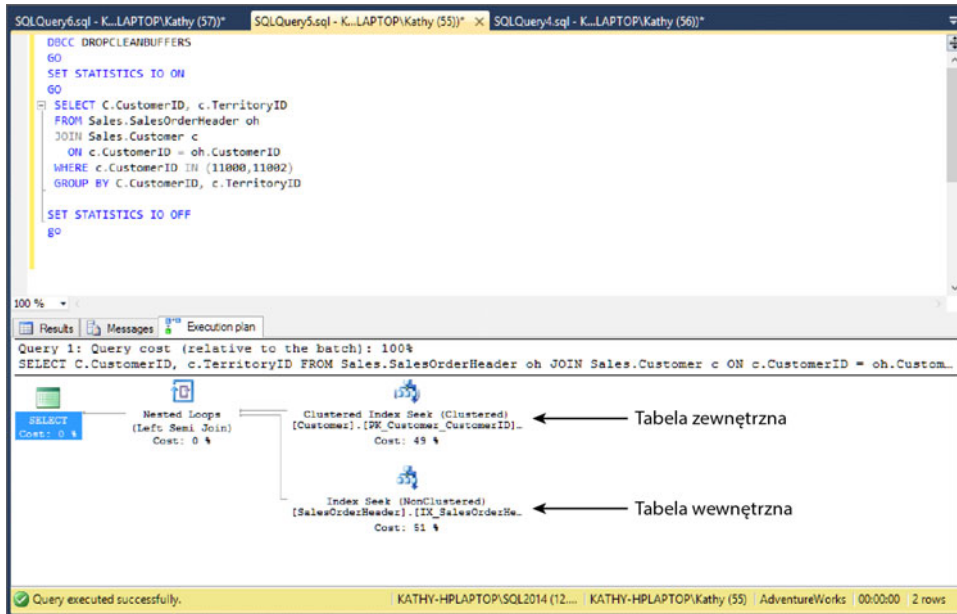
Listing 13.5. Plik *zlaczenie.sql*

```
-- Złączanie w pętli zagnieżdżonej.
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO
SELECT C.CustomerID, c.TerritoryID
FROM Sales.SalesOrderHeader oh
```

```

JOIN Sales.Customer c
  ON c.CustomerID = oh.CustomerID
WHERE c.CustomerID IN (11000,11002)
GROUP BY C.CustomerID, c.TerritoryID
GO
SET STATISTICS IO OFF
GO

```



Rysunek 13.16. Plan wykonywania kwerendy ze złączeniem w pętli

W tym przykładzie z tabeli Customer wczytywany jest wiersz, po czym znajdowane są dla niego wszystkie pasujące wiersze z tabeli SalesOrderHeader. Zgodnie z klauzulą WHERE pobierane są tylko dwa rekordy z tabeli Customer, dlatego tabela SalesOrderHeader jest wczytywana tylko dwukrotnie. Ponieważ dla tabeli SalesOrderHeader używany jest indeks nieklastrowany obejmujący potrzebne pola, w operacji SEEK można wykorzystać właśnie go (należy wczytać drzewo indeksu, aby znaleźć konkretny wiersz). Ta kwerenda jest bardzo wydajna, ponieważ z zewnętrznej tabeli pobieranych jest niewiele wierszy, a w wewnętrznej tabeli używany jest odpowiedni indeks przyspieszający znajdowanie pasujących wierszy.

Złączanie w pętli zagnieżdżonej jest wydajne zwłaszcza wtedy, gdy zewnętrzne dane wejściowe są niewielkie, a wewnętrzne dane wejściowe są duże i posortowane. W wielu małych transakcjach (dotyczących niewielkich zbiorów wierszy) złączenia w pętli zagnieżdżonej z wykorzystaniem indeksów są szybsze niż złączenia przez scalanie i złączenia z haszowaniem (szczegółowo opisane dalej). Jednak w dużych kwerendach złączenia w pętlach zagnieżdżonych często nie są optymalnym rozwiązaniem. Obecność operatora złączenia w pętli zagnieżdżonej w planie wykonania nie oznacza, że dany plan jest wydajny; w trakcie dostrajania wydajności trzeba sprawdzić, czy zastosowany sposób złączania jest najbardziej wydajną metodą zwracania wierszy w danym kontekście. Powinno się też ustalić, czy dla złączenia w pętli zagnieżdżonej dostępny jest odpowiedni indeks, który pozwala wykonać przeszukiwanie w wewnętrznej kwerendzie bez skanowania tabeli lub indeksu.

Złączenia z haszowaniem

Złączenie z haszowaniem, tak jak każde inne, przyjmuje dwa zbiory danych wejściowych — *zbiór budowania* (tabela zewnętrzna) i *zbiór próbkowania* (tabela wewnętrzna). Optymalizator kwerend tak rozdziela te role, aby mniejszy z wejściowych zbiorów danych był używany jako zbiór budowania. Jedną z odmian złączania z haszowaniem (z *operatorem fizycznej agregacji*) pozwala usuwać powtórzenia i grupować dane; służą do tego np. instrukcje SUM (OrderQty) lub GROUP BY TerritoryID. W tych metodach jeden zbiór danych wejściowych pełni funkcje zbiorów budowania i próbkowania.

W listingu 13.6 (plik *zlaczanie2.sql*) przedstawiono przykład złączania z haszowaniem. Graficzny plan wykonywania kwerendy jest przedstawiony na rysunku 13.17.

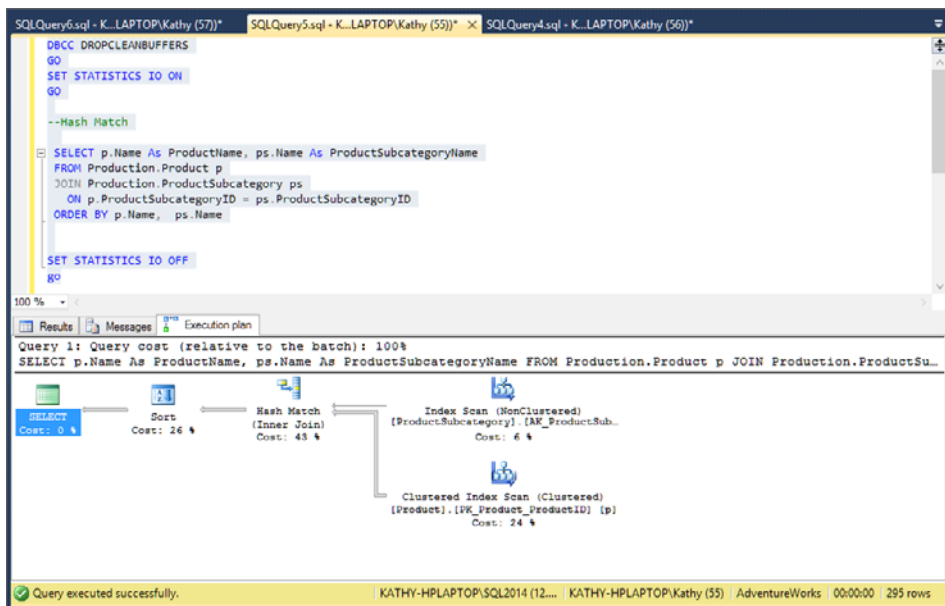
Listing 13.6. Plik *zlaczanie2.sql*

-- Złączanie z haszowaniem.

```
DBCC DROPLEANBUFFERS
GO
SET STATISTICS IO ON
GO
```

```
SELECT p.Name As ProductName, ps.Name As ProductSubcategoryName
FROM Production.Product p
JOIN Production.ProductSubcategory ps
ON p.ProductSubcategoryID = ps.ProductSubcategoryID
ORDER BY p.Name, ps.Name
```

```
SET STATISTICS IO OFF
GO
```



Rysunek 13.17. Plan wykonywania kwerendy ze złączaniem z haszowaniem

Przy złączaniu z haszowaniem najpierw następuje skanowanie lub obliczanie wartości skrótów dla całego zbioru budowania, po czym budowana jest tablica z haszowaniem. Na rysunku 13.17 zbiorem budowania jest tabela `Production.ProductSubcategory`.

Po etapie budowania następuje etap próbkowania. Cały zbiór próbkowania (tabela `Production.Product` na rysunku 13.17) jest skanowany lub obliczany wiersz po wierszu. Dla każdego sprawdzanego wiersza z tabeli `Production.Product` należy obliczyć wartość klucza po haszowaniu, przejrzeć odpowiedni kubełek (utworzony na podstawie tabeli `Production.ProductSubcategory`) i dopasować wiersze do siebie. Ta strategia to *złączanie z haszowaniem w pamięci*.

Tablica z haszowaniem standardowo jest generowana w pamięci. Gdy tabele są bardzo duże, system może nie mieć wystarczająco dużo pamięci, aby pomieścić kompletną tablicę z haszowaniem. Jeśli zbiór budowania nie mieści się w pamięci, w trakcie złączania potrzebne są dodatkowe kroki związane z zapisywaniem tablicy z haszowaniem na dysku. Może to spowolnić przebieg złączania z haszowaniem dla dużych tabel, ponieważ z powodu braku pamięci ich przetwarzanie wymaga kilku przebiegów. W takiej sytuacji tablica z haszowaniem jest dzielona na mniejsze, mieszczące się w pamięci grupy, które są przetwarzane w odrębnych krokach. Funkcja odpowiadająca za partycjonowanie na podstawie haszowania gwarantuje, że wiersze, które należy złączyć, znajdują się w tych samych partycjach. *Rekurencyjne złączanie z haszowaniem* jest konieczne, gdy tablice z haszowaniem są tak duże, że trzeba je dzielić na partycje w wielu krokach i na wielu poziomach w każdym kroku.

UWAGA System SQL Server zawsze zaczyna od złączania z haszowaniem w pamięci i zmienia strategię tylko wtedy, gdy jest to konieczne.

Rekurencyjne złączanie z haszowaniem (ang. *hash bailout*) może prowadzić do spadku wydajności. Jeśli w pliku śladu zdarzeń rozszerzonych znajdziesz wiele zdarzeń *Hash Warning* (należą one do kategorii *Errors and Warnings*), zaktualizuj statystyki złączanych kolumn. Jeśli w kwerendzie występuje wiele złączeń z haszowaniem, powinieneś rejestrować wspomniane zdarzenie. Dzięki temu upewnisz się, że rekurencyjne złączanie z haszowaniem nie prowadzi do problemów z wydajnością. Gdy dla złączanych kolumn nie istnieją odpowiednie indeksy, optymalizator standardowo wybiera złączanie z haszowaniem. Jest to dobra wskazówka oznaczająca, że dodanie do tabeli indeksu ułatwiającego złączanie może spowodować poprawę wydajności.

Złączanie przez scalanie

Złączanie ze scalaniem wymaga posortowanych danych wejściowych przy dopasowywaniu rekordów z dwóch tabel. Jest to wydajny algorytm, jeśli oba zbiory wejściowe są posortowane w ten sam sposób.

W kwerendzie z listingu 13.7 (plik *zlaczanie3.sql*) obie tabele mają indeks klastrowany oparty na kolumnie `SalesOrderID`. Dlatego optymalizator wybiera złączanie przez scalanie, co ilustruje rysunek 13.18. Czasem optymalizator wybiera złączanie przez scalanie nawet wtedy, gdy jeden ze zbiorów wejściowych nie jest wstępnie posortowany według indeksu. Wtedy do planu wykonywania kwerendy dodawane jest sortowanie. Optymalizator stosuje tę technikę, jeśli zbiór wejściowy jest na tyle mały, że koszty sortowania są akceptowalne w porównaniu z zastosowaniem złączania z haszowaniem lub złączania w pętli zagnieżdżonej. Jeśli optymalizator przed scalaniem sortuje dane, warto sprawdzić, czy dodanie do tabeli indeksu pokrywającego umożliwi przyspieszenie przetwarzania kwerendy i wyeliminuje konieczność sortowania danych.

Indeks pokrywający pozwala wcześniej posortować potrzebne dane i zachować wyniki. Dzięki temu w momencie wykonywania kwerendy potrzebne dane są dostępne.

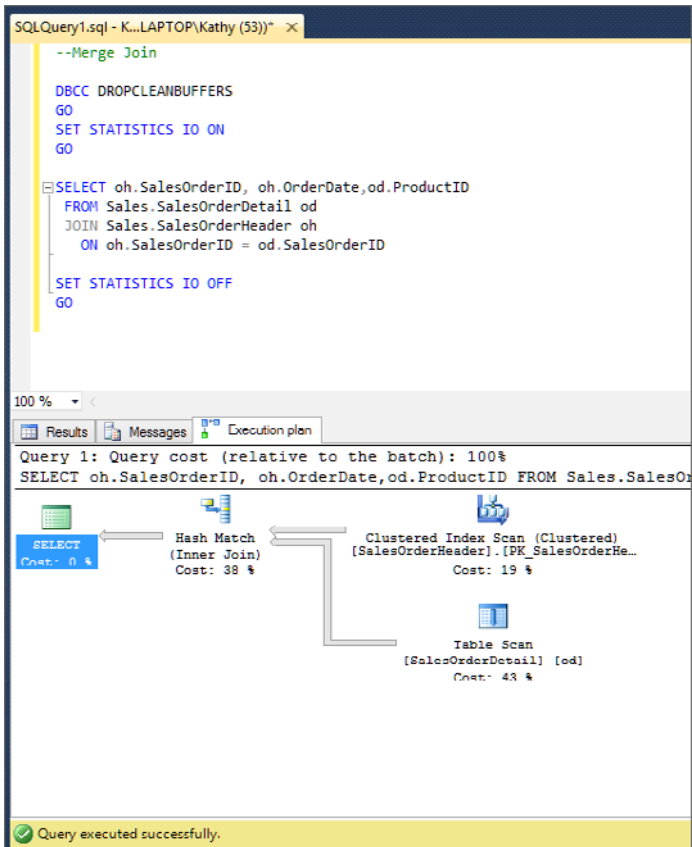
Listing 13.7. Plik zlaczenie3.sql

-- Złączenie ze scalaniem.

```
DBCC DROPCLEANBUFFERS
GO
SET STATISTICS IO ON
GO

SELECT oh.SalesOrderID, oh.OrderDate, od.ProductID
FROM Sales.SalesOrderDetail od
JOIN Sales.SalesOrderHeader oh
ON oh.SalesOrderID = od.SalesOrderID

SET STATISTICS IO OFF
GO
```



Rysunek 13.18. Plan wykonywania kwerendy ze złączaniem przez scalanie

W tej kwerendzie posortowane wiersze z każdej tabeli są skanowane jeden po drugim. Po wykryciu pasujących wierszy zwracany jest złączony wiersz. Istnieją dwa rodzaje złączania przez scalanie.

Złączanie przez scalanie jest bardzo wydajne, gdy kwerenda dopasowuje dwie duże tabele o tych samych początkowych wartościach w kluczu indeksu klastrowanego lub tabele mające tak samo posortowane indeksy pokrywające. Gdy klucze są unikatowe, złączanie przez scalanie działa szybciej (następuje wtedy scalanie jeden do wielu). Jeśli klucze indeksu nie są unikatowe, tworzona jest tabela tymczasowa, używana do dopasowywania rekordów w relacji wiele do wielu (są one następnie zwracane). W obu sytuacjach ważne jest, aby przy dostrajaniu wydajności sprawdzić, czy dodanie indeksu pozwoli wyeliminować kosztowne operacje i przyspieszyć uzyskanie wyników.

Plany wykonywania kwerend modyfikujących dane

Przy modyfikowaniu danych generowany plan ma dwa etapy. Pierwszy dotyczy tylko odczytu danych i określa, które wiersze zostaną wstawione, zaktualizowane lub usunięte.

Na pierwszym etapie plan wykonywania kwerendy generuje strumień danych opisujących zmiany. W instrukcjach INSERT potrzebne są wartości kolumn, dlatego to one znajdują się w strumieniu danych. W instrukcjach DELETE ważne są kolumny klucza, a w instrukcjach UPDATE — strumień danych, wartości modyfikowanych kolumn i klucz. Jeśli używane są klucze zewnętrzne, plan obejmuje sprawdzanie ograniczeń. Ponadto należy zaktualizować indeks, a jeżeli z tabelą powiązane są wyzwalacze, trzeba je wywołać.

Istnieją dwie strategie używane w instrukcjach INSERT, UPDATE i DELETE; są oparte na wierszach i oparte na indeksach. Przyjrzyj się poniższej kwerendzie DELETE, dla której generowany jest plan wykonywania oparty na wierszach:

```
DELETE FROM New_SalesOrderHeader
WHERE OrderDate = '2007-07-01 00:00:00.000'
```

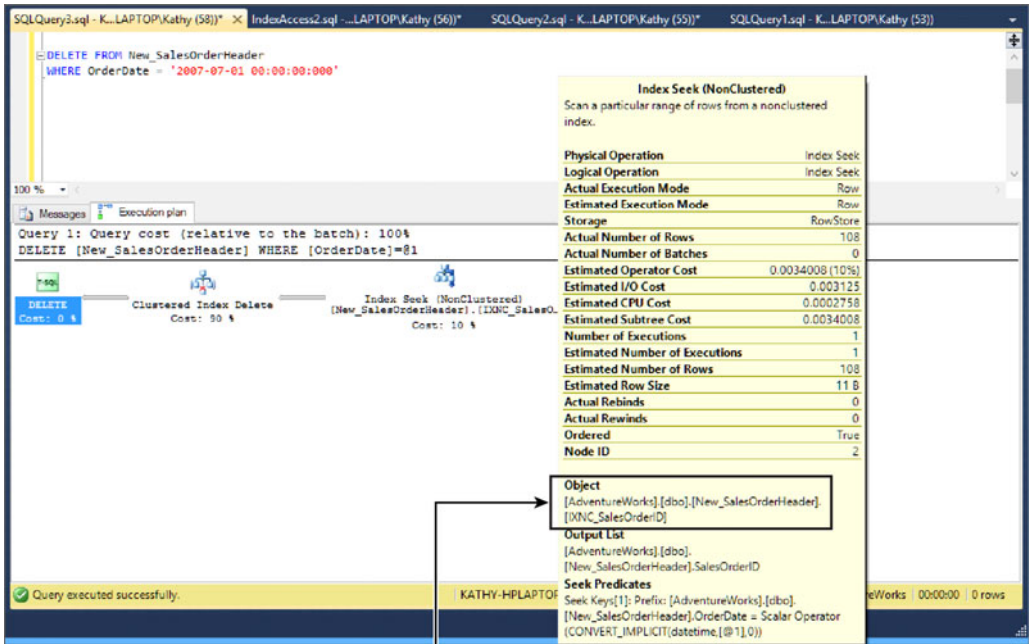
Ten plan jest przedstawiony na rysunku 13.19.

W planie opartym na wierszach system SQL Server 2014 zajmuje się konserwacją indeksu i tabeli bazowej dla każdego wiersza modyfikowanego przez kwerendę. Aktualizacja wszystkich indeksów nieklastrowanych odbywa się wtedy w momencie aktualizowania każdego wiersza tabeli bazowej. Tabela bazowa może mieć postać sterty lub indeksu klastrowanego. Przyjrzyj się informacjom o usuwaniu indeksu klastrowanego w polu na rysunku 13.19. Zauważ, że w wyróżnionej sekcji *Object* podane są oba indeksy (klastrowany i nieklastrowany), co oznacza, że indeksy są obsługiwane w ramach operacji dotyczących poszczególnych wierszy.

UWAGA Dzięki krótkiej ścieżce kodu oraz aktualizowaniu wszystkich indeksów i tabel wspólnie strategia aktualizacji oparta na wierszach jest wydajniejsza ze względu na cykl procesora.

Teraz przyjrzyj się innemu planowi wykonywania kwerendy wygenerowanemu dla poniższego kodu:

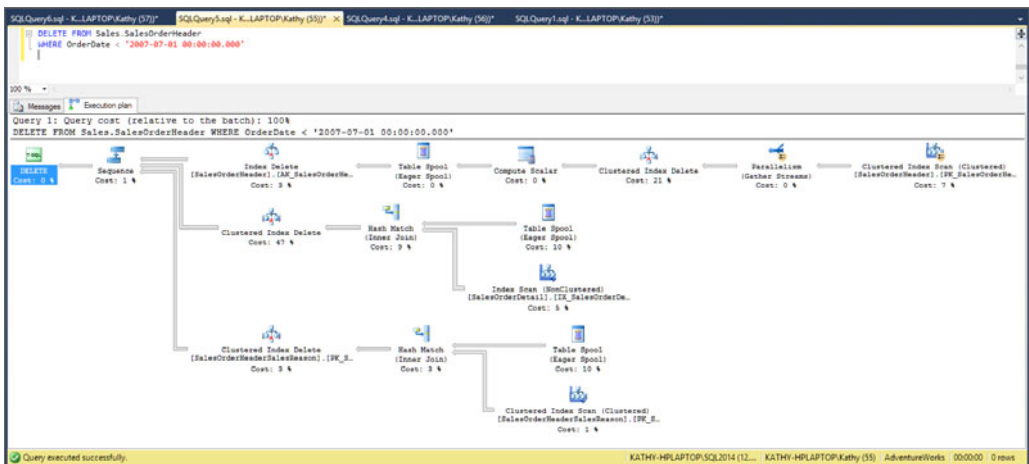
```
DELETE FROM Sales.SalesOrderHeader
WHERE OrderDate < '2006-07-01 00:00:00.000'
```

Wymienione są indeksy
klastrowany i nieklastrowany

Rysunek 13.19. Plan wykonywania kwerendy DELETE

W tej kwerendzie zmodyfikowano klauzulę WHERE (teraz używany jest operator <), co powoduje usunięcie większej grupy wierszy. Wykorzystano tu tabelę Sales.SalesOrderHeader. Ta tabela ma sześć indeksów. Plan wykonywania nowej kwerendy przedstawiono na rysunku 13.20.



Rysunek 13.20. Plan wykonywania kwerendy usuwającej większą liczbę wierszy

Ten plan wykonywania kwerendy przeprowadza konserwację opartą na indeksach. Najpierw usuwane są odpowiednie wiersze z indeksu klastrowanego, a jednocześnie tworzona jest tymczasowa tabela zawierająca wartości klucza klastrowania na potrzeby indeksów nieklastrowanych, które wymagają zmodyfikowania. System SQL Server 2014 wczytuje odpowiednie wartości

kluczy tyle razy, ile jest nieklastrowanych indeksów w tabeli. Operator `sort` między operatorami `delete` i `spool` oznacza, że system SQL Server 2014 sortuje dane według kolumny klucza indeksu, który ma usunąć. Pozwala to na optymalny dostęp do stron indeksu. Operator `sequence` wyznacza kolejność przetwarzania poszczególnych gałęzi. System SQL Server aktualizuje indeksy jeden po drugim (od górnej części planu do dolnej).

Ten plan wykonywania kwerendy pokazuje, że konserwacja oparta na indeksach jest bardziej skomplikowana. Jednak ponieważ poszczególne indeksy są przetwarzane po ich posortowaniu według klucza (zwróć uwagę na operator `Sort`), nigdy nie trzeba wracać do poszczególnych stron, co pozwala zmniejszyć liczbę operacji wejścia-wyjścia. Dlatego przy aktualizowaniu wielu wierszy optymalizator zwykle wybiera plan oparty na indeksach.

Przetwarzanie kwerend działających na tabelach i indeksach podzielonych na partycje

Zanim zaczniesz lekturę tego punktu, możesz przejść do punktu „Tabele i indeksy podzielone na partycje” z rozdziału 14., aby lepiej zrozumieć tabele podzielone na partycje. W tym miejscu zakładamy, że masz podstawową wiedzę z tego zakresu.

Tabele i indeksy podzielone na partycje nie tylko ułatwiają zarządzanie systemem, ale też mogą prowadzić do wzrostu wydajności. Podział tabel na partycje zmienia przebieg operacji współbieżnych i pomaga ograniczyć zakres skanowania dużych tabel.

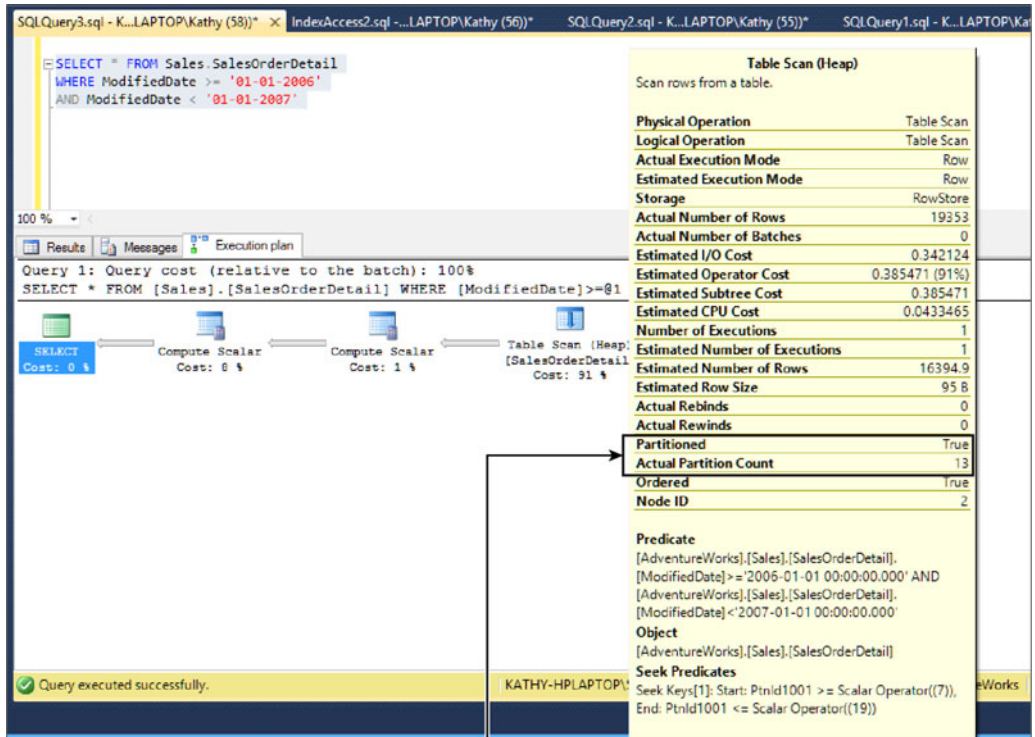
UWAGA Tabele i indeksy dzielone na partycje są obsługiwane tylko w edycjach Enterprise i Developer systemu SQL Server.

Operacje, na które wpływa podział na partycje

Podział tabel i indeksów na partycje wpływa na niektóre operacje w planach wykonywania kwerend. Tabele podzielone na partycje umożliwiają dodanie identyfikatora partycji indeksu do klucza indeksu klastrowanego, co pozwala podzielić tabelę na mniejsze jednostki. Tabela podzielona na partycje zawsze ma ukryte pole `PartitionID` powiązane z kluczem indeksu klastrowanego. Jeśli w klauzuli `WHERE` kwerendy używany jest klucz partycjonowania, zostaje on przekształcony na identyfikator z pola `PartitionID`, po czym wczytywane są tylko wiersze z odpowiednim identyfikatorem partycji. Jeżeli klucz partycjonowania nie jest używany w kwerendzie i wczytywanych jest wiele partycji, operatory współbieżne pomagają poprawić wydajność dzięki zastosowaniu dodatkowych procesów do wczytywania danych.

Operacje, na które wpływa podział na partycje, mają dodatkowe atrybuty w szczegółowym opisie. Na rysunku 13.21 przedstawiamy szczegóły dotyczące operatora z planu wykonywania kwerendy obsługującego operacje na partycjach. Widoczna jest też liczba partycji uwzględnionych przy przetwarzaniu kwerendy.

W tym przykładzie schemat i funkcja partycjonująca służą do podziału tabeli na czterdzieści dwie partycje odpowiadające miesiącom. Gdy optymalizator rozpoczyna przetwarzanie tej kwerendy, dodaje ukrytą kolumnę `PartitionID`, która reprezentuje numer partycji podzielonej tabeli. Kolumna jest uwzględniana jako pierwsza w operacjach `SEEK` i `SCAN`. Optymalizator kwerend szuka klucza indeksu klastrowanego i używa przy tym kolumny `PartitionID` jako pierwszej kolumny klucza złożonego.



Dodatkowe atrybuty związane z partycjami

Rysunek 13.21. Plan wykonywania kwerendy uwzględniający partycje

W tym przykładzie klucz złożony obejmuje kolumny PartitionID i LastModifiedDate. Trzeba też przeanalizować, które partycje należy wczytać, aby spełnić warunki określone w kwerendzie. Kolumna PartitionID jest używana w wyszukiwaniu na pierwszym poziomie i pozwala ograniczyć liczbę wierszy przetwarzanych w kwerendzie. Po ustaleniu odpowiednich partycji przeprowadzane jest normalne wyszukiwanie (operacja SEEK) i znajdowane są pasujące rekordy zwracane do kwerendy. W tym przykładzie uwzględniający partycje operator Table Scan wczytuje dane z tylko trzynastu z czterdziestu dwóch partycji, aby pobrać wszystkie dane potrzebne w kwerendzie.

Oto lista operatorów uwzględniających partycje i używających atrybutów partycji w planach wykonywania kwerend:

- Table Scan,
- Index Scan,
- Index Seek,
- Insert,
- Update,
- Delete,
- Merge.

Strategie współbieżnego wykonywania kwerend dla obiektów podzielonych na partycje

System SQL Server wykorzystuje współbieżność, aby poprawić wydajność kwerend używających tabel podzielonych na partycje. Jeśli liczba wątków jest mniejsza niż liczba partycji, każdy wątek jest przypisywany do innej partycji. Dlatego początkowo dla niektórych partycji nie jest przypisany żaden wątek. Gdy wątek zakończy przetwarzanie partycji, procesor kwerend przypisuje go do następnej partycji (aż do momentu ustawienia jednego wątku dla każdej partycji). Jest to jedyna sytuacja, gdy procesor kwerend przełącza wątek z jednej partycji do innej.

Jeżeli liczba wątków jest równa liczbie partycji, procesor kwerend przypisuje jeden wątek do każdej partycji. Wtedy wątek po zakończeniu pracy *nie jest przydzielany* innej partycji.

Gdy liczba partycji jest mniejsza od liczby wątków, procesor kwerend przypisuje każdej partycji więcej niż jeden wątek. W takiej sytuacji liczba wątków przydzielonych poszczególnym partycjom może być różna.

Zapamiętaj poniższe zalecenia. Przy dostępie do dużych ilości danych pomogą zapewnić wyższą wydajność kwerend działających na tabelach podzielonych na partycje.

- Wykorzystaj to, że w powszechnie dostępnych serwerach używane są procesory wielordzeniowe. System SQL Server może zastosować je do współbieżnego przetwarzania kwerend.
- Starannie zaplanuj podział tabel na partycje. Koniecznie przetestuj wpływ partycji na kwerendy, aby zminimalizować spadek wydajności.
- Pamiętaj, że optymalny podział na partycje pod kątem zarządzania danymi może wyglądać inaczej niż podział pod kątem przetwarzania kwerend.
- Pamiętaj, aby sprawdzić wydajność nieklastrowanego indeksu podzielonego na partycje i upewnić się, czy indeksy można wyrównać względem partycji.
- Upewnij się, że dla tabel podzielonych na partycje dostępny jest indeks klastrowany, aby procesor kwerend mógł wykorzystać optymalizacje z zakresu skanowania indeksów.
- Jeśli agregujesz dużą ilość danych z tabeli podzielonej na partycje, zadbaj o to, aby dla bazy tempdb dostępna była wystarczająca ilość miejsca.

UWAGA Więcej informacji o monitorowaniu pamięci zajmowanej przez bazę tempdb znajdziesz w artykule „Capacity Planning for tempdb” w dokumentacji Books Online.

Analizowanie wydajności kwerend w środowisku produkcyjnym

Wcześniej w tym rozdziale dowiedziałeś się, jak przeglądać graficzne plany wykonywania kwerend w programie SQL Server Management Studio. Jednak czasem przy monitorowaniu wydajności systemu będziesz musiał przeprowadzić bardziej szczegółowe analizy. Śledzenie przebiegu kwerend w środowisku produkcyjnym jest bardzo ważne w trakcie rozwiązywania występujących w nim problemów.

Zamiast wyświetlać plany na podstawie przykładowych lub testowych kwerend w programie SQL Server Management Studio, należy zbierać informacje, plany wykonywania i wskaźniki

wydajności dotyczące kwerend uruchamianych przez użytkowników w środowisku produkcyjnym. Zdarzenia rozszerzone pozwalają rejestrować plany wykonywania kwerend i wskaźniki wydajności na podstawie rzeczywistych kwerend działających w takim środowisku.

OSTRZEŻENIE Zachowaj ostrożność, jeśli rejestrujesz plany wykonywania kwerend za pomocą zdarzeń rozszerzonych. Ta operacja, jeśli zostanie nieprawidłowo przeprowadzona, może wymagać dużo zasobów i negatywnie wpływać na wydajność systemu. Więcej informacji na ten temat znajdziesz na stronie <http://connect.microsoft.com/SQLServer/feedback/details/732870/sqlserver-query-post-execution-showplan-performance-impact>.

W rozdziale 12., „Monitorowanie systemu SQL Server”, opisano monitorowanie zdarzeń systemowych za pomocą zdarzeń rozszerzonych. Tu poznasz inne techniki rejestrowania błędów, ostrzeżeń i planów wykonywania kwerend.

Aby rejestrować plany wykonywania kwerend w środowisku produkcyjnym, uruchom sesję zdarzeń rozszerzonych i włącz monitorowanie następujących zdarzeń:

- `query_post_compilation_showplan`,
- `query_post_execution_showplan`,
- `query_pre_execution_showplan`.

Zdarzenia te zwracają opis planu w formacie XML. Na podstawie tego opisu możesz określić etapy przetwarzania kwerendy. Ponadto można monitorować inne zdarzenia, aby uzyskać dodatkowe, szczegółowe informacje na temat etapów przetwarzania i problemów z wydajnością.

W trakcie zbierania danych koniecznie zastosuj w kryteriach monitorowania odpowiednie filtry, ponieważ pliki dziennika mogą się szybko rozrastać, a rejestrowanie planów w formacie XML wymaga wielu operacji na dysku w ramach ich zapisywania w tych plikach.

Właściwie zastosowane zdarzenia rozszerzone pozwalają uzyskać wiele szczegółowych informacji na temat wolno działających kwerend w środowisku produkcyjnym. Na podstawie tych informacji możesz dopracować instrukcje w języku T-SQL, tak aby działały wydajniej.

Systemowe widoki DMV

Jednym z zadań z zakresu dostrajania wydajności jest sprawdzanie wysokiego obciążenia procesora i niskiej wydajności spowodowanych rekompilacją kwerend. Są różne przyczyny rekompilacji: stosowanie parametrów, nieaktualne statystyki lub duża liczba zmian w danych.

Niektóre systemowe widoki DMV można wykorzystać do analizy problemów wynikających z rekompilacji planów wykonywania kwerend. Najważniejszy z tych widoków to `sys.dm_exec_cached_plans`. Ten widok wyświetla wiersz dla każdego planu zapisanego w pamięci podręcznej. Wspomniany widok razem z widokiem `sys.dm_exec_sql_text` można wykorzystać do pobrania kodu kwerendy w języku SQL zapisanego w pamięci podręcznej z planami. Oto przykładowa instrukcja tego rodzaju:

```
SELECT st.text, cp.plan_handle, cp.usecounts, cp.size_in_bytes,
       cp.cacheobjtype, cp.objtype
FROM sys.dm_exec_cached_plans cp
      CROSS APPLY sys.dm_exec_sql_text(cp.plan_handle) st
ORDER BY cp.usecounts DESC
```

Pobierane informacje znajdują się w następujących kolumnach.

- Kolumna `text` zawiera kod kwerendy w języku SQL, dla której wygenerowano plan.
- Kolumna `usecounts` określa, ile razy zastosowano dany plan. Ta liczba powinna być wysoka. Jeśli dla wielu kwerend jest ona niska, oznacza to, że system często musi przeprowadzać rekompilację.
- Kolumna `size_in_bytes` określa liczbę bajtów zajmowanych przez plan.
- Kolumna `cacheobjtype` określa typ obiektu w pamięci podręcznej (informuje, czy jest to skompilowany plan lub podobny obiekt).

Innym wartym zbadania widokiem DMV jest `sys.dm_exec_query_stats`. Za pomocą tego widoku można uzyskać dla kwerend zagregowane statystyki wydajności dotyczące wszystkich uruchomień każdej kwerendy. Poniższa kwerenda zwraca pięćdziesiąt kwerend z określonej bazy (np. AdventureWorks) powodujących największe obciążenie procesora:

-- 50 kwerend, które najbardziej obciążają procesor.

USE AdventureWorks

GO

SELECT TOP 50

```
DB_NAME(DB_ID()) AS [Database Name],
qs.total_worker_time / execution_count AS avg_worker_time,
SUBSTRING(st.TEXT, (qs.statement_start_offset / 2) + 1,
((CASE qs.statement_end_offset
  WHEN -1 THEN DATALENGTH(st.TEXT)
  ELSE qs.statement_end_offset
END - qs.statement_start_offset) / 2) + 1)
AS statement_text, *
FROM
  sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS st
ORDER BY avg_worker_time DESC;
```

Używanie systemowych widoków DMV do wykrywania rekompilacji pomoże zapewnić optymalną pracę systemu i zapobiec problemom w środowisku produkcyjnym.

Łączenie wszystkich elementów

Dostrajanie wydajności to proces badania pracy systemu i konkretnych kwerend z wykorzystaniem narzędzi, planów wykonywania kwerend oraz widoków DMV i funkcji DMF. Będziesz używał danych z wszystkich tych źródeł do ustalania, czy poszczególne kwerendy są wydajnie przetwarzane. Plany w formie graficznej i tekstowej zapewniają szczegółowe informacje na temat operatorów, przepływu danych i kroków przetwarzania. Na podstawie tych planów można ustalić, czy potrzebne są dodatkowe indeksy, zmiany w kodzie w języku T-SQL, zaktualizowane statystyki, dodatkowy sprzęt lub inne modyfikacje.

Najważniejszym czynnikiem wpływającym na powodzenie dostrajania wydajności środowiska jest ustalenie wyjściowych wartości wskaźników i ich ponowny pomiar po wprowadzeniu modyfikacji w systemie lub kodzie. Pomiar wydajności systemu za pomocą zdarzeń rozszerzonych pomaga wykryć problemy, zanim doprowadzą do sytuacji kryzysowej. Zapoznanie się z planami wykonywania kwerend w systemie testowym lub produkcyjnym pozwala znaleźć kwerendy o niskiej wydajności.

Najlepsi administratorzy baz danych mają bogatą wiedzę na temat wskaźników wydajności systemów: od poziomu obciążenia procesora przez wskaźniki operacji wejścia-wyjścia po wykorzystanie pamięci. Śledzenie tych wskaźników i wykrywanie nieoczekiwanych zmian w ich wartościach pomaga uniknąć przeciążenia systemu w momencie niespodziewanej zmiany wydajności kwerendy.

Podsumowanie

Wiesz już, jak czytać plany wykonywania kwerend, jak przetwarzane są kwerendy w systemie SQL Server, a także jak identyfikować usprawnienia wydajności. Poznałeś kroki, przez jakie kwerenda przechodzi w procesie kompilowania i wykonywania. Najpierw kwerenda jest parsowana, w wyniku czego generowane są operatory logiczne. Po etapie optymalizacji otrzymywane są operatory fizyczne. Trafiają one do silnika wykonawczego, gdzie są przetwarzane. W rozdziale omówiono też optymalizację opartą na kosztach i wpływ nowego mechanizmu szacowania kardynalności, wprowadzonego w systemie SQL Server 2014. Ponadto poznałeś wpływ statystyk na działanie tego mechanizmu. Wiesz już też, że do uzyskania poprawnych planów wykonywania kwerend potrzebne są aktualne statystyki.

Poznałeś wpływ różnych operatorów (m.in. operatorów dostępu do danych, Table Scan i Index Scan, oraz operatorów złączania, Hash Join, Nested Loops Join i Merge Join) z planów wykonywania kwerend na wydajność. Opisano tu również podział na partycje i ich wpływ na wydajność kwerend.

Identyfikowanie problematycznych kwerend powinno należeć do standardowych zadań w trakcie zarządzania systemem SQL Server. Microsoft i inni producenci udostępniają wiele narzędzi do monitorowania środowiska i pomiaru wydajności. Dowiedziałeś się, że zdarzenia rozszerzone pozwalają uzyskać wiele szczegółowych informacji przydatnych w trakcie rozwiązywania problemów z wydajnością. Ważne jest, aby wiedzieć, jakie wyniki są oczekiwane, a jakie wartości wskazują na problemy. Wiesz już też, że systemowe widoki DMV są pomocne przy wykrywaniu rekompilacji kwerend.

Skoro znasz już podstawy dostrajania wydajności kodu w języku SQL, możesz przejść do rozdziału 14., z którego dowiesz się, jak tworzyć indeksy pozwalające w poprawieniu wydajności kwerend.

Indeksowanie baz danych

ZAWARTOŚĆ ROZDZIAŁU:

- Związane z indeksami mechanizmy w systemie SQL Server 2014 i jego starszych wersjach.
- Sposób, w jaki podzielone na partycje tabele i indeksy ułatwiają zarządzanie bazami i zwiększają ich skalowalność.
- Tworzenie tabel i indeksów podzielonych na partycje.
- Konserwowanie i dostrajanie indeksów.

Jednym z najważniejszych zadań administratorów produkcyjnych baz danych jest dbanie o to, aby czas przetwarzania kwerend był zgodny z umową o poziomie świadczonych usług lub oczekiwaniami użytkowników. Jedną z najskuteczniejszych technik poprawiających wydajność kwerend jest tworzenie indeksów.

Wydajność kwerend zwykle mierzy się na podstawie czasu potrzebnego na ich wykonanie oraz ilości koniecznych zasobów i pracy. Długie i kosztowne kwerendy zajmują zasoby przez długi czas. Może to prowadzić do spowolnienia aplikacji i przekroczenia limitu czasu w trakcie generowania raportów lub wykonywania innych operacji na bazie danych.

Każdy administrator produkcyjnych baz danych powinien poznać związane z indeksami mechanizmy systemu SQL Server 2014 i dowiedzieć się, jak je stosować.

W rozdziale zamieszczony został przegląd związanych z indeksami mechanizmów z systemu SQL Server 2014. Poznasz tu m.in. najnowsze funkcje indeksów kolumnowych (klastrowanych i modyfikowalnych), które stanowią znaczne rozwinięcie mechanizmów wprowadzonych w systemie SQL Server 2012.

UWAGA W tym rozdziale nie omawiamy mechanizmów indeksów przeznaczonych dla zoptymalizowanych tabel przechowywanych w pamięci (np. indeksów z haszowaniem). Omówienie usprawnień indeksów dla takich tabel, w tym indeksów z haszowaniem, znajdziesz w rozdziale 9., „Technologia In-Memory OLTP”.

Nowe mechanizmy w indeksach w systemie SQL Server 2014

Indeks kolumnowy (ang. *columnstore index*) to rodzaj indeksu wprowadzony po raz pierwszy w wersji SQL Server 2012. Jest to oparty na kolumnach nieklastrowany indeks, który ma zwiększać wydajność kwerend w operacjach na dużej ilości danych, zwykle dostępnych w tabelach faktów w hurtowniach danych.

W systemie SQL Server 2014 wprowadzono dwie nowe funkcje indeksów kolumnowych; są to możliwość tworzenia klastrowanych indeksów tego rodzaju i możliwość aktualizowania istniejących klastrowanych indeksów kolumnowych.

W wersji SQL Server 2012, w której wprowadzono indeksy kolumnowe, można było tworzyć je jedynie w wersji nieklastrowanej. Ponadto takiego indeksu po utworzeniu nie można było modyfikować. W wersji SQL Server 2014 można tworzyć klastrowane indeksy kolumnowe (które są tabelami), a następnie je aktualizować.

W systemie SQL Server 2014 wprowadzono też znaczne usprawnienia w zarządzaniu indeksami w trybie online. W tej wersji można teraz odtwarzać indeksy dla pojedynczych partycji, co nie było dozwolone we wcześniejszych wersjach. To usprawnienie niezwykle przydaje się w firmach, które używają bardzo dużych tabel i muszą rozbijać proces konserwacji na kilka dni.

Inną nowością w systemie SQL Server 2014 jest możliwość wyświetlania informacji o indeksach kolumnowych w planach wykonywania kwerend zwracanych za pomocą zdarzeń z rodziny Showplan. Właściwości `EstimatedExecutionMode` i `ActualExecutionMode` przyjmują dwie wartości — `Batch` i `Row`. Właściwość `Storage` także przyjmuje dwie wartości — `RowStore` i `ColumnStore`.

Choć podstawowe mechanizmy działania indeksów nie zmieniły się, w kilku ostatnich wersjach systemu SQL Server wprowadzono wiele usprawnień. W tabeli 14.1 zamieszczono listę wielu powiązanych z indeksami funkcji dodanych we wcześniejszych wersjach systemu SQL Server i dostępnych także w wydaniu 2014.

UWAGA W tym rozdziale korzystamy z przykładowych baz danych AdventureWorks i AdventureWorksDW. Można je pobrać ze strony <http://msftdbprodsamples.codeplex.com/>.

Tabela 14.1. Zmiany w działaniu indeksów

Funkcja	Wprowadzono w wersji	Opis działania w pierwotnej wersji
Indeksy kolumnowe	SQL Server 2012	Są to oparte na kolumnach indeksy nieklastrowane, które mają zwiększać wydajność kwerend przetwarzających dużą ilość danych, występującą zwykle w tabelach faktów w hurtowniach danych. Dla tabeli można utworzyć tylko jeden indeks kolumnowy i nie można go modyfikować. Indeksy kolumnowe w niektórych kwerendach zapewniają wysoką wydajność. Dotyczy to zwłaszcza kwerend o dużej liczbie odczytów (takie kwerendy często stosuje się w hurtowniach danych). Indeksy kolumnowe mają pewne ograniczenia. Dla tabeli można utworzyć tylko jeden taki indeks. Ponadto takie indeksy można dzielić na partycje, ale wyłącznie w postaci wyrównanej względem partycji (partycje muszą być takie same jak dla tabeli). Niektóre typy danych (np. <code>binary</code> , <code>varbinary</code> , <code>text</code> , <code>image</code> , <code>varchar(max)</code> i <code>nvarchar(max)</code>) nie są obsługiwane.
Selektywne indeksy XML-owe	SQL Server 2012	Funkcję wprowadzono, aby umożliwić programistom projektowanie indeksów XML-owych opartych na konkretnej ścieżce. Jest to alternatywa dla indeksowania całych dokumentów XML.
Operacje na indeksach w trybie online	SQL Server 2012	Funkcja polega na wykonywaniu operacji na indeksach w trybie online dla typów LOB (<code>image</code> , <code>text</code> , <code>ntext</code> , <code>varchar(max)</code> , <code>nvarchar(max)</code>) i <code>xml</code> .
Obsługa do 15 000 partycji	SQL Server 2008	Limit 999 partycji tabeli został w wersjach SQL Server 2008 SP2 i SQL Server 2008 R2 SP1 podniesiony do 15 000.
Filtrowane indeksy i statystyki	SQL Server 2008	Można zastosować predykat do tworzenia filtrowanych indeksów i statystyk dla podzbioru wierszy tabeli. W wersjach starszych niż SQL Server 2008 indeksy i statystyki były tworzone dla wszystkich wierszy tabeli. W systemie SQL Server 2008 można dodać predykat <code>WHERE</code> w indeksie lub statystyce, aby ograniczyć liczbę uwzględnianych wierszy. Filtrowane indeksy i statystyki są przydatne zwłaszcza w kwerendach pobierających dane z dobrze zdefiniowanych podzbiorów albo kolumn z wartościami z różnych kategorii lub przedziałów.
Kompresja tabel i indeksów	SQL Server 2008	Dodano możliwość kompresji (na poziomie wierszy i stron) tabel, indeksów i widoków indeksowanych przechowywanych na dysku. Kompresję tabel i indeksów można skonfigurować niezależnie dla każdej partycji.

Tabela 14.1. Zmiany w działaniu indeksów — ciąg dalszy

Funkcja	Wprowadzono w wersji	Opis działania w pierwotnej wersji
Indeksy przestrzenne	SQL Server 2008	Dodano obsługę danych i indeksów przestrzennych. <i>Dane przestrzenne</i> to w tym kontekście dane reprezentujące obiekty geometryczne lub lokalizację fizyczną. System SQL Server obsługuje dwa rodzaje danych przestrzennych: geograficzne i geometryczne. <i>Kolumna przestrzenna</i> to kolumna tabeli zawierająca dane przestrzenne. <i>Indeks przestrzenny</i> to rodzaj indeksu rozszerzonego, który umożliwia indeksowanie kolumny przestrzennej. W systemie SQL Server do implementowania takich danych służy środowisko NET CLR. Szczegółowe informacje na ten temat zawiera artykuł „Working with Spatial Indexes (Database Engine)” w dokumentacji Books Online.
Tabele i indeksy podzielone na partycje	SQL Server 2005	W wersji SQL Server 2005 umożliwiono tworzenie tabel w wielu partycjach i indeksowanie każdej partycji. Pozwala to wydajniej zarządzać operacjami na dużych zbiorach danych (np. wczytywać lub usuwać nowe zbiory danych) dzięki indeksowaniu tylko nowych partycji — nie trzeba ponownie indeksować całej tabeli.
Operacje na indeksach w trybie online	SQL Server 2005	W systemie SQL Server 2005 dodano obsługę operacji na indeksach w trybie online, co zwiększa dostępność baz. Użytkownicy mogą nadal uruchamiać kwerendy dotyczące tabeli nawet wtedy, kiedy jej indeks jest właśnie budowany lub odtwarzany. Głównym zastosowaniem tej nowej funkcji jest modyfikowanie indeksów w godzinach pracy systemu. Do wykonywania operacji na indeksach w trybie online służy opcja <code>ONLINE = ON</code> dodawana do instrukcji <code>CREATE INDEX</code> , <code>ALTER INDEX</code> , <code>DROP INDEX</code> i <code>ALTER TABLE</code> .
Współbieżne operacje na indeksach	SQL Server 2005	Współbieżne operacje na indeksach to następna przydatna funkcja z systemu SQL Server 2005. Jest ona dostępna tylko w edycji Enterprise i w systemach działających na komputerach wieloprocessorowych. Głównym zastosowaniem tej funkcji jest ograniczanie ilości zasobów procesora zużywanych przez operacje na indeksach. Możliwe, że będziesz chciał jednocześnie wykonywać różne operacje na indeksach lub, co bardziej prawdopodobne, będziesz musiał umożliwić wykonywanie innych zadań w trakcie modyfikowania indeksu. Funkcja pozwala administratorom baz danych ustawić opcję <code>MAXDOP</code> dla operacji na indeksach. Jest to przydatne w dużych systemach, gdzie umożliwia ograniczenie maksymalnej liczby procesorów używanych w operacjach na indeksach. Jest to więc opcja <code>MAXDOP</code> przeznaczona dla operacji na indeksach. Przy jej ustawianiu trzeba uwzględnić opcję <code>MAXDOP</code> z poziomu serwera. Składnia tego mechanizmu to opcja <code>MAXDOP = n</code> , którą można dodać do instrukcji <code>CREATE INDEX</code> , <code>ALTER INDEX</code> , <code>DROP INDEX</code> (tylko dla indeksów klastrowanych), <code>ALTER TABLE ADD (indeks) CONSTRAINT</code> , <code>ALTER TABLE DROP (indeks klastrowany) CONSTRAINT</code> .

Tabela 14.1. Zmiany w działaniu indeksów — ciąg dalszy

Funkcja	Wprowadzono w wersji	Opis działania w pierwotnej wersji
Asynchroniczne aktualizowanie statystyk	SQL Server 2005	Można je włączyć za pomocą opcji SET AUTO_UPDATE_STATISTICS_ASYNC. Gdy opcja ta jest ustawiona, nieaktualne statystyki są umieszczane w kolejce i później automatycznie aktualizowane przez wątek roboczy. Kwerenda, która wygenerowała żądanie automatycznej aktualizacji, kontynuuje pracę jeszcze przed zaktualizowaniem statystyk. Asynchroniczna aktualizacja statystyk nie jest możliwa, jeśli w ramach tej samej transakcji wywoływane są instrukcje DDL (np. CREATE, ALTER lub DROP).
Indeksy pełnotekstowe	SQL Server 2005	Od wersji SQL Server 2005 przy wyszukiwaniu pełnotekstowym obsługiwane jest tworzenie indeksów dla kolumn z danymi w formacie XML. Funkcja ta została zaktualizowana tak, aby wykorzystywana w niej była technologia MSSearch 3.0, co pozwala dodatkowo zwiększyć wydajność indeksów pełnotekstowych. Oznacza to także, że dla każdego egzemplarza systemu SQL Server używany jest jeden egzemplarz technologii MSSearch.
Kolumny, które nie są kluczami, stosowane w indeksach nieklastrowanych	SQL Server 2005	W wersjach SQL Server 2005 i 2008 kolumny, które nie są kluczami, można stosować w indeksach nieklastrowanych. Ma to kilka zalet. Umożliwia kwerendom szybsze pobieranie danych, ponieważ teraz można wczytać wszystkie potrzebne informacje ze stron indeksu bez konieczności przeszukiwania tabeli w celu odczytu wiersza danych. W indeksach nieklastrowanych kolumny, które nie są kluczami, nie są uwzględniane w limicie liczby kolumn (16) lub długości klucza (900 bajtów). Składnia tej funkcji to INCLUDE (nazwa kolumny, ...) w instrukcji CREATE INDEX.
Zmiana precyzji blokad indeksów	SQL Server 2005	W systemie SQL Server 2005 instrukcje CREATE INDEX i ALTER INDEX języka T-SQL zostały wzbogacone o nowe opcje kontrolujące blokady dodawane w trakcie wykonywania operacji na indeksach. Opcje ALLOW_ROW_LOCKS i ALLOW_PAGE_LOCKS określają precyzję blokad stosowanych w czasie takich operacji.
Indeksy dla kolumn z danymi w formacie XML	SQL Server 2005	Indeks dla danych w formacie XML umożliwia silnikowi bazy danych wyszukiwanie elementów w tych danych bez konieczności każdorazowego ich wyodrębniania.
Usuwanie i odtwarzanie dużych indeksów	SQL Server 2005	W systemie SQL Server 2005 zmodyfikowano silnik bazodanowy, aby obsługiwał indeksy zajmujące ponad 128 ekstentów w bardziej skalowalny sposób. Proces usuwania lub odtwarzania takich indeksów jest dzielony na etapy logiczny i fizyczny. Na etapie logicznym strony są oznaczane jako nieprzydzielone. Po zatwierdzeniu transakcji następuje etap fizycznej dezalokacji stron. Jest ona wykonywana w tle w porcjach, co pozwala uniknąć ustawiania blokad na zbyt długi czas.

Tabela 14.1. Zmiany w działaniu indeksów — ciąg dalszy

Funkcja	Wprowadzono w wersji	Opis działania w pierwotnej wersji
Usprawnienia widoków indeksowanych	SQL Server 2005	Widoki indeksowane zostały ulepszone pod kilkoma względami. Obecnie mogą zawierać zagregowane dane skalarne i (z pewnymi ograniczeniami) funkcje zdefiniowane przez użytkownika. Ponadto optymalizator kwerend może lepiej dopasowywać kwerendy do widoków indeksowanych, gdy w kwerendzie używane są wyrażenia skalarne, zagregowane dane skalarne, funkcje zdefiniowane przez użytkownika, wyrażenia zwracające przedziały i warunki równoznaczności.
Magazyn wersji	SQL Server 2005	Magazyn wersji to podstawa platformy kontroli wersji wierszy używanej przez indeksowanie w trybie online, technologię MARS (ang. <i>Multiple Active Results Sets</i>), wyzwacze i nowe poziomy izolacji oparte na kontroli wersji wierszy.
DTA (ang. <i>Database Tuning Advisor</i>)	SQL Server 2005	DTA zastąpiło mechanizm ITW (ang. <i>Index Tuning Wizard</i>) z systemu SQL Server 2000. DTA udostępnia nowe funkcje, w tym dostrajanie z ograniczeniem czasowym, możliwość dostrajania indeksów z różnych baz danych, możliwość dostrajania bogatszego zestawu klas zdarzeń i wyzwaczy, dziennik dostrajania, analizy przyczynowo-skutkowe, lepszą kontrolę nad opcjami dostrajania, obsługę plików XML, obsługę partycji, przekazywanie procesu dostrajania do sprzętu o niższych parametrach i uruchamianie tego procesu przez właścicieli bazy danych.

Indeksy i tabele podzielone na partycje

W tym podrozdziale omawiamy proces tworzenia i używania indeksów dla tabel podzielonych na partycje, co ma ułatwić zarządzanie dużymi tabelami i ich skalowanie.

Omówienie indeksów

Projektowanie dobrego indeksu rozpoczyna się od zrozumienia korzyści, jakie indeks ma zapewniać. W książkach spis treści pomagają czytelnikom znaleźć interesujące ich punkty, rozdziały lub strony. Indeksy w systemie SQL Server pełnią tę samą funkcję — umożliwiają systemowi szybkie znajdowanie i pobieranie danych żądanych w kwerendzie.

Wyobraź sobie książkę o grubości pięciuset stron z wieloma punktami i rozdziałami, która jednak nie ma spisu treści. Aby znaleźć określony fragment książki, czytelnik musi przejrzeć wszystkie strony, aż natrafi na szukany tekst. A musisz powtórzyć ten proces dla wielu fragmentów. To bardzo pracochłonne zadanie.

To samo dotyczy tabel w bazach systemu SQL Server. Jeśli nie istnieje odpowiedni indeks, system SQL Server musi przejrzeć wszystkie strony zawierające dane z tabeli. Jeśli tabela zawiera dużo danych, to zadanie wymaga dużo czasu i zasobów. Dlatego indeksy są tak ważne.

Indeksy można pokategoryzować na kilka sposobów — na podstawie metod przechowywania danych, wewnętrznej struktury, przeznaczenia i rodzaju definicji. W poniższych podpunktach pokrótce opisujemy różne rodzaje indeksów.

Indeksy oparte na wierszach (indeksy wierszowe)

Indeksy oparte na wierszach (indeksy wierszowe) to tradycyjne indeksy, w których dane są przechowywane jako wiersze na stronach danych. W tej kategorii można wyróżnić indeksy klastrowane i nieklastrowane.

Indeksy klastrowane

Indeksy klastrowane przechowują dane z tabeli i sortują je na podstawie kolumn kluczy. Strony indeksu są powiązane ze sobą, zatem można w minimalnej liczbie operacji wejścia-wyjścia sekwencyjnie wczytać tabelę w kolejności wyznaczonej przez klucz klastrowania. Dla tabeli można utworzyć tylko jeden indeks klastrowany, ponieważ dane można posortować w tylko jednym porządku, a indeks klastrowany reprezentuje dane tabeli.

Klastrowanie danych poprawia wydajność ich sekwencyjnego odczytu. Jedna strona danych może zawierać od kilku do wielu wierszy danych znajdujących się według sortowania obok siebie.

Pola klucza klastrowania są przenoszone do wszystkich indeksów nieklastrowanych, gdzie pełnią funkcję referencji do wiersza z poziomu liści. Jeśli klucz indeksu klastrowanego jest duży, może wpływać na wielkość indeksów nieklastrowanych.

Indeks klastrowany jest tworzony domyślnie, gdy w definicji tabeli występuje klucz główny. Dobry indeks klastrowany ma te same cechy, co dobry klucz główny — wartości pól nie zmieniają się i są rosnące. Tego rodzaju klucz pomaga ograniczyć modyfikowanie stron przy dodawaniu nowych rekordów.

Indeksy nieklastrowane

Indeks nieklastrowany obejmuje wartości klucza indeksu i lokalizatory prowadzące do wierszy z danymi. Jeśli nie jest używany indeks klastrowany, tym lokalizatorem jest wskaźnik RowID prowadzący do wiersza. Gdy indeks klastrowany jest dostępny, lokalizator to wartość klucza indeksu klastrowanego dla danego wiersza.

Indeksy nieklastrowane można optymalizować, aby zapewnić obsługę większej liczby kwerend, skrócić czas przetwarzania kwerend i zmniejszyć wielkość indeksu. Poniżej opisujemy dwa najważniejsze rodzaje zoptymalizowanych indeksów nieklastrowanych.

Indeksy pokrywające

Indeksy pokrywające to indeksy, które obejmują wszystkie pola potrzebne w danej kwerendzie. Indeks nieklastrowany może zawierać na poziomie liści kolumny, które nie są kluczami, co pozwala zapewnić pokrycie kwerendy. Aby dodać takie kolumny, zastosuj klauzulę `INCLUDE` w instrukcji `CREATE INDEX`. Tego rodzaju indeksy poprawiają wydajność kwerend i ograniczają liczbę operacji wejścia-wyjścia, ponieważ kolumny potrzebne do wykonania kwerendy są dostępne w samym indeksie (mogą to być kolumny klucza lub zwykłe kolumny), dlatego nie trzeba wczytywać wierszy z danymi.

Dzięki klauzuli **INCLUDE** indeksy nieklastrowane dają więcej swobody, ponieważ można w nich uwzględnić kolumny zwykle niedozwolone w kluczu. Kolumny te nie są uwzględniane przy określaniu wielkości indeksu i liczby kolumn klucza.

Indeksy filtrowane

W *indeksach filtrowanych* używana jest klauzula **WHERE** do określenia, które wiersze należy uwzględnić w indeksie. Ponieważ indeksowana jest wtedy tylko część tabeli, można zapisać w indeksie mniejszy zbiór danych. Indeksy filtrowane zawsze są indeksami nieklastrowanymi, ponieważ obejmują podzbiór całego zbioru rekordów reprezentowanego w indeksie klastrowanym. Indeks filtrowany jest używany w planie wykonywania kwerendy, jeśli klauzula **WHERE** danej kwerendy wymaga tylko wierszy z klauzuli **WHERE** takiego indeksu.

Po co tworzyć indeks nieklastrowany dotyczący podzbioru danych tabeli? Dobrze zaprojektowany indeks filtrowany przyspiesza odczyt wybranych grup wierszy z tabeli, ponieważ wystarczy wczytać mniejszą liczbę stron indeksu. Przykładowo indeks kompletnej dużej tabeli też jest duży. Jeśli w tej tabeli występują mniejsze, dobrze zdefiniowane grupy często używanych wierszy, indeks filtrowany może pobierać tylko te wiersze. Pozwala to przyspieszyć pracę kwerend używających tych wierszy, ponieważ wystarczy wczytać mniej stron indeksu (indeks jest wtedy bardziej płaski i obejmuje mniej wierszy). Ponadto statystyki dla indeksów filtrowanych są bardziej wiarygodne, bo dotyczą mniejszych zbiorów danych i z większym prawdopodobieństwem dobrze je reprezentują. Lepsze statystyki i mniejsze zbiory danych prawie zawsze prowadzą do wydajniejszych planów wykonywania kwerend i wyższej wydajności.

Ponadto mniejsze indeksy są też korzystne podczas administrowania, ponieważ zajmują mniej miejsca i można je szybciej odbudować, a statystyki są dla nich generowane szybciej i dodatkowo są trafniejsze.

Indeksy oparte na kolumnach

Indeksy oparte na kolumnach to indeksy tworzone dla pojedynczych kolumn. Dwa podstawowe rodzaje takich indeksów to indeksy kolumnowe (wprowadzone po raz pierwszy w wersji SQL Server 2012) i indeksy XML-owe (indeksujące wartości w kolumnach z danymi w formacie XML).

Indeksy kolumnowe

Indeksy kolumnowe (ang. *columnstore index*) pojawiły się po raz pierwszy w systemie SQL Server 2012. W tego rodzaju indeksach dla każdej kolumny tworzony jest indeks wartości wierszy, po czym wszystkie indeksy są złączane i stanowią podstawowy magazyn danych tabeli. Indeksy te są oparte na silniku Vertipaq, który umożliwia znaczną kompresję danych i obsługuje duże zbiory danych. W systemie SQL Server 2012 indeksów tego rodzaju nie można modyfikować. Aby dodać wartości do indeksu, trzeba go ponownie utworzyć.

Nieklastrowane indeksy kolumnowe mają następujące cechy.

- Mogą dotyczyć podzbioru kolumn tabeli (klastrowanej lub w postaci sterty).
- Można je aktualizować tylko w wyniku ponownego utworzenia indeksu.
- Można je łączyć z innymi indeksami tabeli.
- Wymagają dodatkowego miejsca na kopie kolumn w indeksie, przechowywanych obok wartości wierszy.

W systemie SQL Server 2014 klastrowane indeksy kolumnowe można aktualizować, przy czym obowiązują pewne ograniczenia.

- Klastrowany indeks kolumnowy nie może obejmować nieklastrowanych indeksów — musi być jedynym indeksem tabeli.
- Tabeli zapisanej w postaci klastrowanego indeksu kolumnowego nie można używać przy replikacji.
- Dla tabel zapisanych w postaci klastrowanego indeksu kolumnowego nie można stosować dostępnej w systemie SQL Server funkcji rejestrowania zmian w danych.
- Tabela zapisana jako klastrowany indeks kolumnowy nie może mieć kolumn z danymi typu FILESTREAM.
- Tabele z klastrowanymi indeksami kolumnowymi są obsługiwane tylko w edycjach Enterprise, Developer i Evaluation systemu SQL Server 2014.
- W klastrowanym indeksie kolumnowym nie można utworzyć klucza głównego. Nie można też tworzyć ograniczeń integralności referencyjnej.

System SQL Server udostępnia dodatkową opcję kompresji dla indeksów kolumnowym (COLUMNSTORE_ARCHIVE), która pozwala zwiększyć poziom kompresji. Funkcja jest dostępna dla klastrowanych i nieklastrowanych indeksów kolumnowych. Powoduje ona dodatkową kompresję indeksu (zwykłego lub podzielonego na partycje). Jest to bardzo przydatne, gdy dane muszą zajmować jak najmniej miejsca, a jednocześnie można sobie pozwolić na dodatkowe obciążenie procesora związane z wypakowywaniem danych.

Klastrowany indeks kolumnowy obejmuje wszystkie pola tabeli, przy czym nie jest wymagany unikatowy klucz. Nie ma tu kolumny klucza, ponieważ w tym podejściu w indeksie nie jest potrzebny klucz prowadzący do wierszy. Po utworzeniu klastrowanego indeksu kolumnowego można używać standardowych instrukcji w języku T-SQL do wczytywania danych z tabeli (w tym do wczytywania masowego). Dla takich danych nie można tworzyć innych indeksów. Gdy dane są modyfikowane lub usuwane, zostają oznaczone jako przeznaczone do usunięcia, po czym w momencie odtwarzania indeksu zajmowane przez nie miejsce jest odzyskiwane.

W tradycyjnych tabelach wierszowych można utworzyć nieklastrowany indeks kolumnowy dla podzbioru kolumn. Dla tabel wierszowych można budować nieklastrowane indeksy kolumnowe i indeksy innych typów, przy czym indeksy kolumnowe nie są wtedy modyfikowalne (aby wprowadzić w nich zmiany, trzeba ponownie utworzyć cały indeks). Ponadto nieklastrowane indeksy kolumnowe nie mogą być używane jako indeksy oparte na kluczu głównym lub zewnętrznym, nie mogą być definiowane jako unikatowe, nie mogą obejmować kolumn typów SPARSE i FILESTREAM, a także nie przechowują statystyk.

Indeksy kolumnowe zaprojektowano do użytku w hurtowniach danych, a nie w aplikacjach OLTP. W hurtowniach tabele zwykle nie są (często) aktualizowane, wymagają zachowania integralności referencyjnej i często są dzielone na partycje. W niektórych kwerendach zastosowanie indeksów kolumnowych może prowadzić do spadku wydajności. Oto dobra wskazówka z obszaru wydajności: warto ponownie tworzyć indeksy kolumnowe dla partycji, które zostały niedawno zaktualizowane, zamiast dla całych tabel. Zwiększa to dostępność bazy i ogranicza problemy.

Tabele można przekształcać z formatu kolumnowego na wierszowy i z powrotem przez usuwanie i ponowne tworzenie indeksów. Jeśli usuniesz indeks kolumnowy, tabela kolumnowa zostanie zapisana w postaci sterty. Jeśli trzeba, można następnie utworzyć klastrowany indeks wierszowy. Aby przekształcić tabelę wierszową w kolumnową, usuń wszystkie indeksy, a następnie dodaj klastrowany indeks kolumnowy.

Zoptymalizowane indeksy przechowywane w pamięci

W systemie SQL Server 2014 wprowadzono nowe indeksy dla *zoptymalizowanych tabel przechowywanych w pamięci*. Indeks z *haszowaniem* działa w pamięci i służy do dostępu do danych w takich tabelach (technologia Hekaton). Ilość potrzebnej pamięci jest zależna od liczby kulek wygenerowanych w indeksie.

Zoptymalizowany nieklastrowany indeks przechowywany w pamięci sortuje dane tabeli przechowywanej w pamięci. Indeksy tego typu są tworzone tylko w instrukcjach `CREATE TABLE` i `CREATE INDEX` oraz są używane przy skanowaniu uporządkowanych przedziałów (przy odczycie dużych ilości danych zgodnie z kolejnością sortowania). Takie indeksy powstają w momencie wczytywania tabeli do pamięci i nie są utrwalane w fizycznej tabeli.

Inne rodzaje indeksów

W systemie SQL Server istnieje też kilka innych rodzajów indeksów przeznaczonych do konkretnych celów. W tym podpunkcie omawiamy podstawowe aspekty wybranych typów indeksów. Jeśli szukasz szczegółowych informacji związanych z indeksami, zapoznaj się z artykułami na ich temat dostępnymi w dokumentacji Books Online.

Indeksy XML-owe

Indeksy XML-owe służą do indeksowania wartości przechowywanych w kolumnach z danymi w formacie XML. Indeksy te wyodrębniają dane z takich kolumn i zachowują je, co zapewnia szybszy dostęp do danych w kwerendach w języku SQL. Kolumny z danymi w formacie XML mogą być bardzo duże, a wyodrębnianie poszczególnych porcji danych z tego formatu w czasie wykonywania programu może spowolnić rozbudowane kwerendy. Gdy stosujesz indeksy XML-owe, wyodrębnione z wyprzedzeniem dane zostają zachowane i w czasie wykonywania aplikacji dostęp do nich jest szybszy.

Istnieją dwa rodzaje indeksów XML-owych — główny i pomocniczy. Pierwszy indeks tworzony dla kolumny z danymi w formacie XML musi być *indeksem głównym*. Indeksuje on wszystkie znaczniki, wartości i ścieżki w takiej kolumnie. Dla każdego obiektu XML-owego indeks tworzy wiersz danych z każdym wyodrębnionym elementem. Liczba utworzonych wierszy odpowiada w przybliżeniu liczbie węzłów w XML-owym obiekcie. W poszczególnych wierszach zapisywane są: nazwa znacznika, wartość węzła, typ węzła, informacje o uporządkowaniu dokumentu, ścieżka i klucz główny (lub identyfikator RowID) tabeli bazowej.

Dla kolumn z danymi w formacie XML można też utworzyć *indeks pomocniczy*, który dodatkowo indeksuje wartości `PATH`, `VALUE` i `PROPERTY` z indeksu głównego.

Indeksy pełnotekstowe

Indeksy pełnotekstowe tworzy się dla mechanizmu wyszukiwania pełnotekstowego dostępnego w systemie SQL Server. Mechanizm umożliwia użytkownikom i aplikacjom wyszukiwanie danych tekstowych w tabelach tego systemu. Aby można było uwzględnić tabelę w wyszukiwaniu pełnotekstowym, trzeba utworzyć dla niej indeks pełnotekstowy.

Indeksować na potrzeby wyszukiwania pełnotekstowego można kolumny typów: `char`, `varchar`, `nchar`, `nvarchar`, `text`, `ntext`, `image`, `xml`, `varbinary(max)` i `FILESTREAM`. Wyszukiwanie pełnotekstowe to wyszukiwanie lingwistyczne słów i wyrażeń zapisanych w kolumnach tych typów. Każdy indeks pełnotekstowy dotyczy jednej lub kilku kolumn tabeli, a dla poszczególnych kolumn można ustawić inne języki używane w trakcie wyszukiwania.

Wyszukiwanie pełnotekstowe to opcjonalna funkcja systemu SQL Server, którą przed zastosowaniem trzeba włączyć. Wyszukiwanie pełnotekstowe to wyszukiwanie lingwistyczne słów i wyrażeń zgodnie z regułami języków zdefiniowanych dla poszczególnych kolumn. Indeksy pełnotekstowe pomagają przyspieszyć tę operację.

Indeksy przestrzenne

Indeks przestrzenny jest przeznaczony dla kolumn z danymi przestrzennymi, obejmujących wartości typu `GEOMETRY` lub `GEOGRAPHY`. Takie indeksy są używane przez operacje działające na danych przestrzennych, np. przez wbudowane metody związane z geografią (`STContains()`, `STDistance()`, `STEquals()` i `STIntersects()`). Aby kwerenda została wybrana przez optymalizator, tego typu metody muszą znajdować się w klauzuli `JOIN` lub `WHERE`.

Jak indeksy są wykorzystywane w systemie SQL Server?

Aby projektować dobre indeksy, trzeba też rozumieć, jak system SQL Server z nich korzysta. W tym systemie optymalizator kwerend określa najbardziej wydajny sposób wykonywania kwerend. Optymalizator ocenia różne plany wykonywania kwerend, a następnie wybiera plan o najniższym koszcie. W rozdziale 13., „Dostrajanie wydajności kodu w języku T-SQL”, opisano, jak indeksy są używane przy wykonywaniu kwerend i dostrajaniu wydajności.

Tworzenie indeksów

Na tym etapie powinieneś znać różne rodzaje indeksów i wiedzieć, jak są stosowane w planach wykonywania kwerend. Ta wiedza jest niezbędna, jeśli chcesz projektować i dostrajać indeksy w celu poprawy wydajności kwerend.

Indeksy są tworzone ręcznie za pomocą instrukcji w języku T-SQL lub przy użyciu interfejsu graficznego (np. programu SQL Server Management Studio). Wprowadzone w systemie SQL Server 2005 narzędzie DTA proponuje, jakie indeksy utworzyć, i generuje brakujące. Omówienie tego narzędzia znajdziesz dalej w tym rozdziale.

Klucz indeksu może obejmować jedno pole lub kilka pól tabeli. Indeksy mogą być klastrowane lub nieklastrowane. W indeksie klastrowanym na poziomie liści znajdują się strony danych zawierające wszystkie kolumny z każdego wiersza. W indeksie nieklastrowanym na poziomie liści dostępne są tylko dane z pól indeksu i kolumn zdefiniowanych w klauzuli `INCLUDE`. Za pomocą instrukcji `CREATE INDEX` domyślnie utworzysz indeks nieklastrowany.

Indeksy w zależności od definicji mogą zawierać unikatowe lub powtarzające się wartości. W indeksie unikatowym w żadnych dwóch wierszach w kluczu indeksu nie może znajdować się ten sam zbiór wartości. Unikatowość może być określona na podstawie wielu pól, a poszczególne pola mogą mieć wartość `NULL`, przy czym identyczne wiersze z wartościami `NULL` są traktowane jak powtórzenia. Za pomocą klucza głównego można utworzyć unikatowy indeks, który nie pozwala stosować wartości `NULL`. Ograniczenie unikatowości umożliwia utworzenie unikatowego indeksu nieklastrowanego. W każdej tabeli można utworzyć do 999 indeksów nieklastrowanych (włącznie z indeksami budowanymi na podstawie ograniczeń).

Indeksy można włączać i wyłączać. Gdy indeks jest włączony, użytkownicy mają do niego dostęp. Wyłączenie indeksu sprawia, że staje się niedostępny. Jeśli zatem indeks klastrowany jest wyłączony, cała tabela staje się niedostępna dla użytkowników. Wyłączone indeksy są niedostępne do momentu ich ponownego utworzenia.

Aby utworzyć indeks za pomocą instrukcji w języku T-SQL, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i połącz się z egzemplarzem systemu SQL Server.
2. Upewnij się, że w tym egzemplarzu zainstalowana jest baza AdventureWorks ze strony <http://msftdbprodsamples.codeplex.com/>. Istnieje kilka wersji tej bazy. Jeśli używasz wersji innej niż w tym przykładzie, zmień przedstawiony dalej kod, aby pasował do Twojej bazy AdventureWorks.
3. Otwórz nowe okno kwerendy i uruchom wybrane z przedstawionych poniżej instrukcji.

Zacznij od utworzenia kopii jednej z tabel. W tym przykładzie skopiuj tabelę HumanResources.Employee za pomocą poniższego skryptu, który eliminuje niepotrzebne pola:

```
SELECT BusinessEntityID, NationalIDNumber,
       LoginID, OrganizationLevel,
       JobTitle, BirthDate, MaritalStatus,
       Gender, HireDate, SalariedFlag,
       VacationHours, SickLeaveHours,
       CurrentFlag, rowguid, ModifiedDate
INTO HumanResources.EmployeeNew
FROM HumanResources.Employee
```

- Aby utworzyć indeks klastrowany dla zbudowanej wcześniej tabeli, użyj polecenia CREATE CLUSTERED INDEX języka T-SQL:

```
CREATE CLUSTERED INDEX cix_BusinessEntityID
ON HumanResources.EmployeeNew(BusinessEntityID)
```

- Aby utworzyć indeks nieklastrowany, wywołaj polecenie CREATE NONCLUSTERED INDEX języka T-SQL. Jest to domyślny rodzaj indeksu, dlatego można pominąć słowo NONCLUSTERED:

```
CREATE NONCLUSTERED INDEX idx_BirthDate
ON HumanResources.EmployeeNew (BirthDate)
```

lub:

```
CREATE INDEX idx_BirthDate
ON HumanResources.EmployeeNew (BirthDate)
```

- W celu utworzenia indeksu pokrywającego wywołaj polecenie CREATE NONCLUSTERED INDEX języka T-SQL z klauzulą INCLUDE:

```
CREATE NONCLUSTERED INDEX cidx_HireDate
ON HumanResources.EmployeeNew (HireDate)
INCLUDE (MaritalStatus, Gender, JobTitle)
```

- Aby utworzyć indeks filtrowany, wywołaj polecenie CREATE NONCLUSTERED INDEX języka T-SQL z klauzulą WHERE:

```
CREATE NONCLUSTERED INDEX idx_GenderFemale
ON HumanResources.EmployeeNew (Gender)
WHERE Gender = 'Female'
```

- W celu utworzenia klastrowanego indeksu kolumnowego najpierw trzeba usunąć wszystkie inne indeksy tabeli, a następnie wywołać polecenie CREATE COLUMNSTORE INDEX języka T-SQL:

```
CREATE CLUSTERED COLUMNSTORE INDEX idx_EmployeeNew ON HumanResources.
EmployeeNew;
```

- Aby przekształcić klastrowany indeks kolumnowy na tabelę wierszową, usuń indeks klastrowany:

```
DROP INDEX idx_EmployeeNew ON HumanResources.EmployeeNew;
```

- W celu utworzenia nieklastrowanego indeksu kolumnowego wywołaj polecenie CREATE COLUMNSTORE INDEX języka T-SQL:

```
CREATE NONCLUSTERED COLUMNSTORE INDEX idx_EmployeeNew  
ON HumanResources.Employee (FirstName, LastName, MaritalStatus, Gender);
```

Używanie tabel i indeksów podzielonych na partycje

W rozdziale 11. dowiedziałeś się, że dzielenie tabel na partycje pomaga zoptymalizować system. Podział tabeli na partycje pozwala rozbić jedną tabelę na wiele jednostek i umieścić każdą z nich w odrębnej grupie plików. Właściwie zastosowane partycje i indeksy pomagają w zarządzaniu dużą ilością danych i szybszym zwracaniu informacji do kwerend.

Partycje tworzy się po to, aby dzielić table na mniejsze jednostki i umożliwiać silnikowi kwerend systemu SQL Server stosowanie lepszych metod optymalizowania (w tym współbieżności i eliminowania partycji). W rozdziałach 11. i 13. opisano, jak silnik kwerend wykorzystuje partycje przy optymalizacji. Indeksy w połączeniu z partycjami dodatkowo wspomagają silnik kwerend, ponieważ zapewniają nowy poziom dostępu do danych, który pomaga identyfikować i znajdować potrzebne w kwerendzie wiersze.

Partycje obejmują w każdym wierszu nie tylko pola kluczy indeksu klastrowanego, ale też klucz partycjonowania. Wiersze w partycji są fizycznie grupowane w jednym miejscu na podstawie klucza partycjonowania. Indeksy tworzone dla tabel podzielonych na partycje można partycjonować za pomocą funkcji partycjonującej i schematu partycjonowania użytych dla tabeli lub za pomocą innej funkcji i innego schematu. Można też w ogóle nie dzielić takiego indeksu na partycje. Gdy indeks nieklastrowany jest dzielony na partycje za pomocą tego samego klucza partycjonowania, który zastosowano do bazowej tabeli (czyli według klucza indeksu klastrowanego), powstaje *indeks wyrównany względem partycji* (ang. *partition-aligned index*). Indeksy wyrównane względem partycji są potrzebne, gdy chcesz stosować technikę okna przesuwanego. Dzięki wyrównaniu partycji i używanego indeksu klastrowanego poszczególne partycje wraz z ich indeksami można przedstawiać i niezależnie konserwować.

Może się jednak okazać, że dodanie innego indeksu, niewyrównanego do partycji, pozwala poprawić wydajność systemu. Jeśli klucz partycjonowania jest stosowany w niewielkiej części kwerend, wydajność niektórych kwerend będzie niższa. W takiej sytuacji użycie indeksu nieklastrowanego i niewyrównanego do partycji powinno poprawić wydajność. Jeśli chcesz jednocześnie stosować indeks niewyrównany do partycji i przedstawiać partycje, przed przedstawianiem partycji musisz usunąć taki indeks, a po ich przedstawieniu — ponownie go utworzyć.

Klucz partycjonowania nie musi być używany w kluczu indeksu (chyba że indeks jest unikatowy).

Konserwacja indeksów

Bardzo ważnym zadaniem administratorów produkcyjnych baz danych jest monitorowanie stanu istniejących indeksów i ustalanie, gdzie potrzebne są nowe indeksy. Za każdym razem, gdy dane są wstawiane, aktualizowane lub usuwane w tabelach systemu SQL Server, indeksy są odpowiednio modyfikowane. W ramach tego procesu dane ze stron z poziomu liści są przenoszone, aby zachować porządek sortowania w indeksie. Może to prowadzić do fragmentacji indeksu.

W indeksach wierszowych po usunięciu lub zmodyfikowaniu wierszy można ponownie wykorzystać puste miejsce, jednak rozdzielanie stron może prowadzić do fragmentacji. Jeśli chodzi o indeksy kolumnowe, ważne jest, aby regularnie ponownie je tworzyć w celu odzyskania pamięci zwolnionej w wyniku usunięcia lub aktualizacji danych. Ponadto należy aktualizować nieklastrowane indeksy kolumnowe.

Z czasem strony mogą zacząć zawierać różną ilość danych. Niektóre strony będą stosunkowo puste, a inne — całkowicie wypełnione. Duża liczba w małym stopniu wypełnionych stron prowadzi do problemów z wydajnością, ponieważ w celu pobrania żądanych danych trzeba wczytać więcej stron.

Z kolei całkowicie wypełnione strony mogą zostać podzielone w momencie wstawiania lub aktualizowania danych. Przy podziale strony mniej więcej połowa danych jest przenoszona do nowo utworzonej strony. Takie stałe reorganizowanie zużywa zasoby i prowadzi do fragmentacji stron danych.

Celem jest zapisanie jak największej ilości danych na jak najmniejszej liczbie stron przy zachowaniu miejsca na dodawanie nowych danych, aby zapobiec zbyt częstym podziałom stron. Tę delikatną równowagę można zachować za pomocą precyzyjnego ustawienia współczynnika wypełnienia indeksu.

UWAGA Więcej informacji o ustawianiu współczynnika wypełnienia indeksu znajdziesz w dokumentacji Books Online na stronie [http://msdn.microsoft.com/en-us/library/ms177459\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms177459(v=SQL.110).aspx).

Monitorowanie poziomu fragmentacji indeksu

Do monitorowania fragmentacji indeksów (w tym indeksów kolumnowych) możesz wykorzystać widoki DMV z systemu SQL Server 2014. Jednym z najprzydatniejszych jest widok `sys.dm_db_index_physical_stats`, który udostępnia informacje o średnim poziomie fragmentacji każdego indeksu.

Oto przykładowa kwerenda pobierająca dane z widoku DMV `sys.dm_db_index_physical_stats`:

```
SELECT index_id, avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats
(
  DB_ID('AdventureWorks'),
  OBJECT_ID('AdventureWorks'),
  NULL, NULL, 'DETAILED'
) ORDER BY avg_fragmentation_in_percent desc
```

Na rysunku 14.1 przedstawiono wyniki wywołania tej kwerendy.

Na podstawie danych z tego widoku DMV można wykryć indeksy o wysokim poziomie fragmentacji. Takie indeksy należy zdefragmentować, aby uniknąć problemów z wydajnością. Indeksy o wysokim poziomie fragmentacji są zapisywane i używane przez system SQL Server 2014 w niewydajny sposób. Istnieją dwa rodzaje fragmentacji — wewnętrzna i zewnętrzna. *Fragmentacja zewnętrzna* polega na tym, że strony danych nie są zapisane w logicznej kolejności. *Fragmentacja wewnętrzna* oznacza, że na stronach znajduje się niewiele danych. Oba typy fragmentacji mogą prowadzić do wydłużenia czasu wykonywania kwerend. Dodatkowe kwerendy do widoków DMV pozwalają zidentyfikować konkretne indeksy wymagające defragmentacji.

```

SELECT index_id, avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats
(
    DB_ID('AdventureWorks'),
    OBJECT_ID('AdventureWorks'),
    NULL, NULL, 'DETAILED'
)
ORDER BY avg_fragmentation_in_percent desc

```

	index_id	avg_fragmentation_in_percent
1	1	100
2	1	87.5
3	1	85.7142857142857
4	3	75
5	2	75
6	1	66.6666666666667
7	2	66.6666666666667
8	1	66.6666666666667
9	1	66.6666666666667
10	2	66.6666666666667
11	1	66.6666666666667
12	2	66.6666666666667
13	4	66.6666666666667
14	3	66.6666666666667
15	2	66.6666666666667
16	5	66.6666666666667
17	1	60

Rysunek 14.1. Informacje o poziomie fragmentacji indeksów

W systemie SQL Server 2014 wprowadzono możliwość porządkowania i defragmentacji poszczególnych partycji indeksu. Pomaga to administratorom baz danych ograniczyć wpływ tych operacji na system i skrócić czas przestoju potrzebny na konserwację.

Porządkowanie indeksów

W ramach konserwacji bazy danych zawsze należy porządkować indeksy. Operację tę należy wykonywać regularnie, w zależności od poziomu fragmentacji indeksów spowodowanej zmianami w danych. Jeśli poziom fragmentacji indeksów jest wysoki, możesz zdefragmentować je za pomocą zmiany uporządkowania lub ponownego utworzenia.

- Reorganizacja indeksu:
 - powoduje zmianę uporządkowania i kompaktowanie stron z poziomu liści,
 - powoduje przeprowadzenie zmiany uporządkowania indeksów w trybie online, bez długotrwałych blokad,
 - dobrze sprawdza się dla indeksów o niskim poziomie fragmentacji.
- Ponowne utworzenie indeksu:
 - powoduje utworzenie nowego indeksu i usunięcie pierwotnego,
 - pozwala odzyskać miejsce na dysku,
 - powoduje zmianę uporządkowania i kompaktowanie wierszy na przyległych stronach,
 - w edycji Enterprise jest możliwe w trybie online,
 - lepiej nadaje się dla indeksów o wysokim poziomie fragmentacji.

W tabeli 14.2 pokazana jest ogólna składnia wykonywania operacji na indeksie tabeli DimCustomer.

Tabela 14.2. Składnia operacji na indeksie tabeli DimCustomer

Operacja	Składnia
Tworzenie indeksu	CREATE INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer (CustomerAlternateKey)
Reorganizacja indeksu	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REORGANIZE
Reorganizacja indeksu w jednej partycji	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REORGANIZE PARTITION = 1
Reorganizacja indeksu we wszystkich partycjach	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REORGANIZE PARTITION = ALL
Ponowne utworzenie indeksu	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REBUILD
Ponowne utworzenie indeksu w jednej partycji	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REBUILD PARTITION = 1
Ponowne utworzenie indeksu we wszystkich partycjach	ALTER INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer REBUILD PARTITION = ALL
Usuwanie indeksu	DROP INDEX IX_DimCustomer_CustomerAlternateKey ON DimCustomer

Z czasem poziom fragmentacji indeksu można znacznie wzrosnąć. Wybór, czy należy zreorganizować, czy ponownie utworzyć dany indeks, zależy od poziomu fragmentacji i czasu na konserwację. Zwykle akceptowalny poziom fragmentacji, po którego przekroczeniu warto ponownie utworzyć indeks, to 20 – 30%. Jeśli poziom fragmentacji jest niższy, prawdopodobnie wystarczy reorganizacja indeksu.

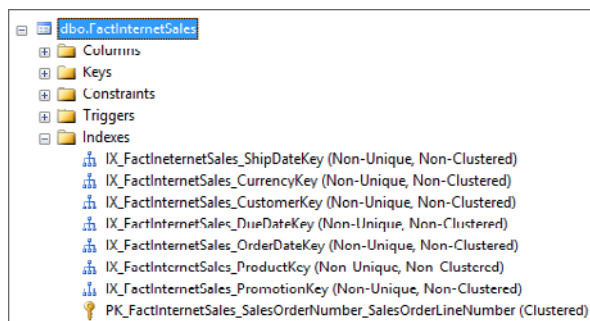
Dlaczego nie tworzyć za każdym razem indeksu od nowa? Można tak robić, jeśli czas na konserwację jest wystarczająco długi. Pamiętaj jednak, że ponowne tworzenie indeksu trwa dłużej. W tym czasie stosowane są blokady, a wszystkie operacje wstawiania, aktualizowania i usuwania danych muszą czekać na utworzenie indeksu. Jeśli używasz edycji Enterprise wersji SQL Server 2005 lub nowszej, możesz tworzyć indeksy w trybie online. Ten tryb umożliwia wstawianie, aktualizowanie i usuwanie danych w trakcie ponownego tworzenia indeksu. Ponadto dzięki dostępnej w systemie SQL Server 2014 funkcji ponownego tworzenia lub reorganizacji indeksu z jednej partycji można rozbić proces porządkowania tabeli na kilka dni i skrócić w ten sposób czas potrzebny każdego dnia na konserwację.

Zwiększanie wydajności kwerend za pomocą indeksów

W systemie SQL Server 2014 dostępnych jest kilka widoków DMV pomocnych przy dostrajaniu kwerend. Widoki te umożliwiają wyświetlenie statystyk dotyczących wykonywania poszczególnych kwerend. Można w ten sposób ustalić: liczbę wywołań kwerendy, liczbę operacji odczytu i zapisu, czas zajmowania procesora, statystyki używania indeksu itd.

Statystyki uzyskane za pomocą widoków DMV można zastosować do dostrojenia kwerendy przez refaktoryzację kodu w języku T-SQL i wykorzystanie współbieżności oraz istniejących indeksów. Na podstawie danych z widoków DMV można też wykryć brakujące indeksy, nieużywane indeksy, a także indeksy wymagające defragmentacji (co opisano wcześniej).

Przyjrzyj się przykładowo istniejącym indeksom tabeli FactInternetSales z bazy AdventureWorksDW. Na rysunku 14.2 widać, że ta tabela jest dobrze zindeksowana.



Rysunek 14.2. Indeksy przykładowej tabeli

Aby zrozumieć proces dostrajania kwerend, wykonaj poniższe kroki. Dzięki nim uzyskasz statystyki wykonywania kwerend dostępne za pomocą widoków DMV.

1. Usuń istniejące indeksy ProductKey i OrderDateKey z tabeli FactInternetSales:

```
USE [AdventureWorksDW]
GO
-- Usunięcie indeksu ProductKey.
IF EXISTS (SELECT * FROM sys.indexes
WHERE object_id = OBJECT_ID(N'[dbo].[FactInternetSales]') AND
name = N'IX_FactInternetSales_ProductKey')
DROP INDEX [IX_FactInternetSales_ProductKey] ON [dbo].[FactInternetSales]
GO
-- Usunięcie indeksu OrderDateKey.
IF EXISTS (SELECT * FROM sys.indexes WHERE object_id =
OBJECT_ID(N'[dbo].[FactInternetSales]')
AND name = N'IX_FactInternetSales_OrderDateKey')
DROP INDEX [IX_FactInternetSales_OrderDateKey] ON [dbo].[FactInternetSales]
GO
```

2. Wykonaj poniższy skrypt trzy razy:

```
/** Internet_ResellerProductSales */
SELECT
    D.[ProductKey],
    D.EnglishProductName,
    Color,
    Size,
    Style,
    ProductAlternateKey,
    sum(FI.[OrderQuantity]) InternetOrderQuantity,
    sum(FR.[OrderQuantity]) ResellerOrderQuantity,
    sum(FI.[SalesAmount]) InternetSalesAmount,
    sum(FR.[SalesAmount]) ResellerSalesAmount
FROM [FactInternetSales] FI
INNER JOIN DimProduct D
```

```

        ON FI.ProductKey = D.ProductKey
    INNER JOIN FactResellerSales FR
        ON FR.ProductKey = D.ProductKey
GROUP BY
    D.[ProductKey],
    D.EnglishProductName,
    Color,
    Size,
    Style,
    ProductAlternateKey

```

Na rysunku 14.3 przedstawiono wykonywany skrypt w języku T-SQL i uzyskane wyniki. Na Twoim komputerze wyniki mogą wyglądać inaczej (zależą one od dostępnych zasobów).

	ProductKey	EnglishProductName	Color	Size	Style	ProductAlternateKey	InternetOrderQuantity	ResellerOrderQuantity	InternetSalesAmount	ResellerSalesAmount
1	311	Road-150 Red, 44	Red	44	U	BK-R33R-44	21637	43836	77423027.99	94114226.232
2	329	Road-650 Red, 48	Red	48	U	BK-R50R-48	20956	76694	16408338.44	35681048.484
3	336	Road-650 Black, 62	Black	62	U	BK-R50B-62	1958	3850	1368834.2796	1614916.765
4	343	Road-650 Black, 52	Black	52	U	BK-R50B-52	19488	82432	15258909.12	38042608.9736
5	354	Mountain-200 Silver, 42	Silver	42	U	BK-M50S-42	45864	155264	95003588.5344	197596662.7048
6	361	Mountain-200 Black, 42	Black	42	U	BK-M50B-42	104188	401380	239110418.12	548919025.1657
7	368	Road-250 Red, 44	Red	44	U	BK-R25R-44	31680	108144	77405326.00	157664301.376
8	379	Road-250 Black, 58	Black	58	U	BK-R25B-58	15400	49362	32612515.00	64651041.00
9	386	Road-550-V8 Yellow, 42	Yellow	42	W	BK-R64V-42	39105	103332	43816761.45	69469483.808
10	472	Classic Vell, M	Blue	M	U	VL-C36AM	78844	414915	4498984.00	15445006.099
11	561	Touring-1000 Yellow, 46	Yellow	46	U	BK-T78V-46	46096	143276	108696090.72	174805806.6568
12	572	Touring-3000 Yellow, 62	Yellow	62	U	BK-T18Y-62	13460	39700	8884607.50	16721510.635
13	578	Touring-2000 Blue, 54	Blue	54	U	BK-T44U-54	26696	81136	31216785.00	58864778.6824
14	586	Touring-3000 Blue, 50	Blue	50	U	BK-T18U-50	12384	38064	9193262.40	15021537.8448
15	597	Mountain-500 Black, 42	Black	42	U	BK-M10B-42	7154	16611	3863088.46	5381864.334
16	604	Road-750 Black, 44	Black	44	U	BK-R19B-44	38160	106560	20606018.40	34524800.64
17	321	Road-650 Red, 58	Red	58	U	BK-R50R-58	7315	21285	5727571.85	9999565.29
18	326	Road-650 Red, 44	Red	44	U	BK-R50R-44	5830	15510	4075742.506	6505807.539
19	328	Road-650 Red, 48	Red	48	U	BK-R50R-48	5746	14586	4017018.2572	6118227.5154

Rysunek 14.3. Kod skryptu i uzyskane wyniki

3. Teraz uruchom poniższy skrypt, aby przeanalizować statystyki dotyczące wykonania przedstawionej wcześniej kwerendy:

```

SELECT TOP 10
    SUBSTRING(qt.TEXT, (qs.statement_start_offset/2)+1,
        ((CASE qs.statement_end_offset WHEN -1 THEN DATALENGTH(qt.TEXT)
        ELSE qs.statement_end_offset
        END - qs.statement_start_offset)/2)+1) QueryText,
    qs.last_execution_time,
    qs.execution_count ,
    qs.last_logical_reads,
    qs.last_logical_writes,
    qs.last_worker_time,

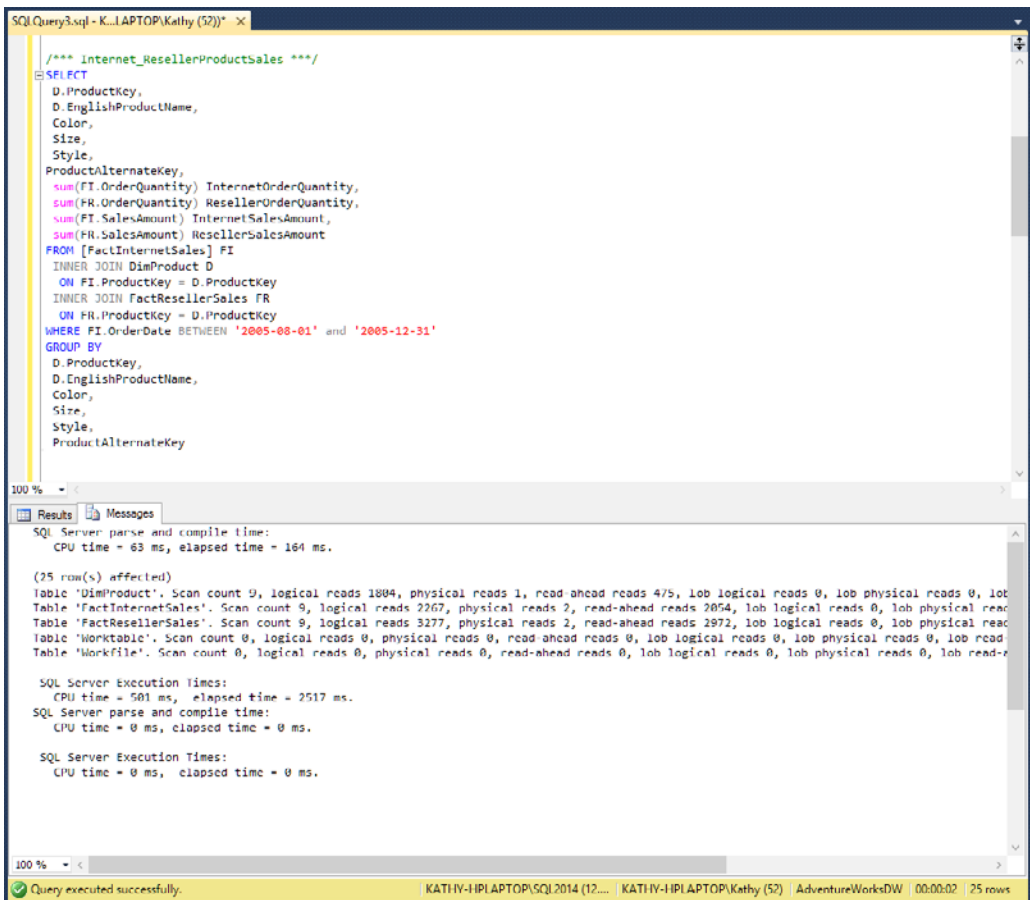
```

```

qs.total_logical_reads,
qs.total_logical_writes,
qs.total_worker_time,
qs.last_elapsed_time/1000000 last_elapsed_time_in_S,
qs.total_elapsed_time/1000000 total_elapsed_time_in_S,
qp.query_plan
FROM
sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) qp
ORDER BY
qs.last_execution_time DESC,
qs.total_logical_reads DESC

```

Na rysunku 14.4 można zobaczyć statystyki pobrane przede wszystkim z widoku DMV `sys.dm_exec_query_stats`.



Rysunek 14.4. Statystyki dotyczące wykonywania przedstawionej wcześniej kwerendy

Informacje z widoku DMV pokazują, że liczba odczytów była duża, a ponadto procesor przez długi czas przetwarzał kwerendę. Warto zachować te wyjściowe wartości. Dalej zobaczysz, jak uzyskać lepsze wyniki.

4. Pobierz dane z widoku DMV `sys.dm_db_missing_index_details`, aby zobaczyć, czy znajdują się w nim informacje o brakujących indeksach:

```
SELECT * FROM sys.dm_db_missing_index_details
```

Na rysunku 14.5 przedstawiono dane z widoku DMV `sys.dm_db_missing_index_details`. Widok ten to doskonałe źródło pozwalające szybko ustalić, czy należy dodać nowe indeksy.

```
SELECT
  DatabaseName = DB_NAME(database_id),
  [Number Indexes Missing] = count(*)
FROM
  sys.dm_db_missing_index_details
GROUP BY
  DB_NAME(database_id)
ORDER BY [Number Indexes Missing] DESC;
```

DatabaseName	Number Indexes Missing
AdventureWorksDW	2

Rysunek 14.5. Informacje o brakujących indeksach

Innym narzędziem do identyfikowania brakujących indeksów jest DTA. Narzędzie udostępnia kreator, który ułatwia przejście przez proces identyfikowania brakujących indeksów. DTA można uruchomić z poziomu menu *Tools* programu SQL Server Management Studio.

UWAGA Aby zidentyfikować potrzebne indeksy, możesz też przeanalizować tekst kwerend zapisany w widoku `sys.dm_exec_sql_text` (patrz rysunek 14.5).

5. W ramach dostrajania kwerend utwórz indeksy `ProductKey` i `OrderDateKey` dla tabeli `FactInternetSales`:

```
USE [AdventureWorksDW]
GO
IF EXISTS
  (SELECT * FROM sys.indexes
   WHERE object_id = OBJECT_ID(N'FactInternetSales')
   AND name = N'IX_FactInternetSales_OrderDateKey')
  DROP INDEX IX_FactInternetSales_OrderDateKey ON
  FactInternetSales
GO
IF NOT EXISTS
  (SELECT * FROM sys.indexes
   WHERE object_id = OBJECT_ID(N'[dbo].[FactInternetSales]') AND
   name = N'IX_FactInternetSales_ProductKey')

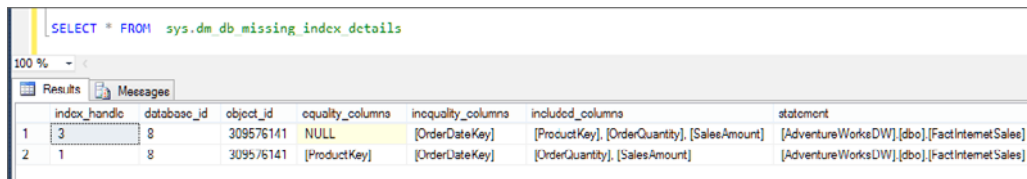
  CREATE NONCLUSTERED INDEX IX_FactInternetSales_ProductKey ON
  FactInternetSales
  (ProductKey ASC)
  WITH
  (PAD_INDEX = OFF,
   STATISTICS_NORECOMPUTE = OFF,
```

```

SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF,
ONLINE = OFF,
ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON
) ON [PRIMARY]
GO

```

6. Ponownie wywołaj trzy razy zdefiniowaną w kroku 2. kwerendę `Internet_Reseller` → `ProductSales`. Na rysunku 14.6 pokazano, że tym razem liczba odczytów znacznie spadła, co przełożyło się na skrócenie czasu wykonywania kwerendy.



	index_handle	database_id	object_id	equality_columns	inequality_columns	included_columns	statement
1	3	8	309576141	NULL	[OrderDateKey]	[ProductKey], [OrderQuantity], [SalesAmount]	[AdventureWorksDW].[dbo].[FactInternetSales]
2	1	8	309576141	[ProductKey]	[OrderDateKey]	[OrderQuantity], [SalesAmount]	[AdventureWorksDW].[dbo].[FactInternetSales]

Rysunek 14.6. Statystyki opisujące wykonywanie kwerendy po dodaniu indeksów

Narzędzie DTA

Jednym z narzędzi, które przyda się administratorom baz danych, jest DTA (ang. *Database Tuning Advisor*) wprowadzone w systemie SQL Server 2005. Wcześniej w tym rozdziale dowiedziałeś się, że DTA umożliwia analizowanie baz danych pod kątem brakujących indeksów. Wyświetla też inne rekomendacje związane z dostrajaniem wydajności, dotyczące np. partycji i widoków indeksowanych. Narzędzie DTA przyjmuje różne typy operacji:

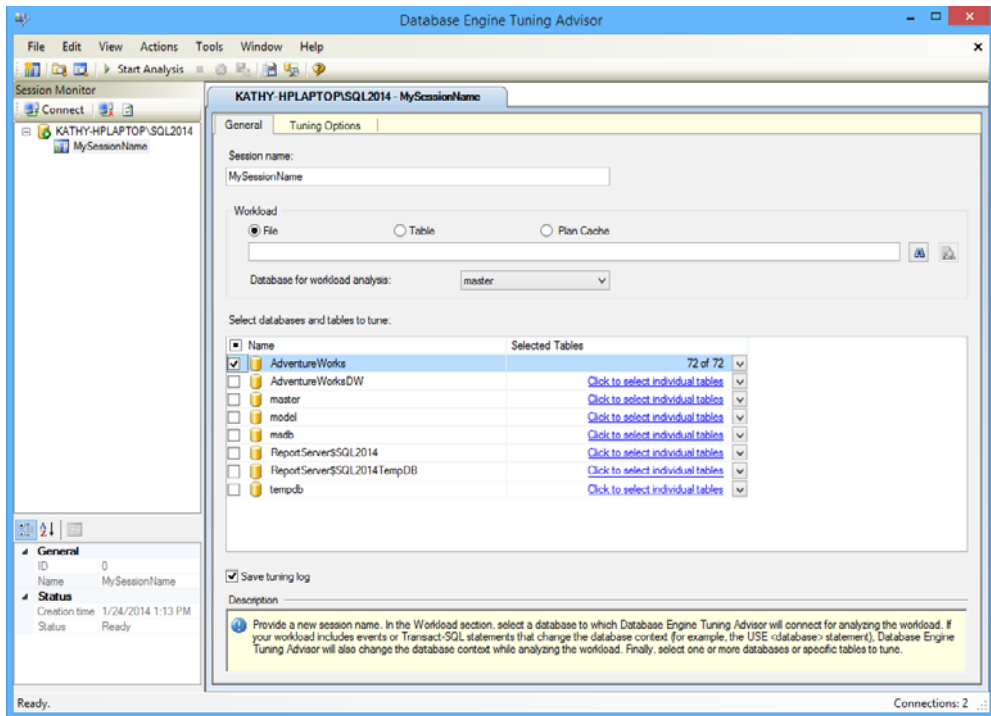
- pliki ze skryptami w języku SQL (*.sql),
- pliki ze śladem (*.trc),
- pliki XML (*.xml),
- tabele ze śladem,
- pamięć podręczną planów wykonywania kwerend.

Na rysunku 14.7 przedstawiono ekran wyboru operacji w narzędziu DTA. Dostępna jest m.in. opcja *Plan Cache* (czyli pamięć podręczna planów). Jedną z zalet narzędzia DTA, ważną dla administratorów baz danych i programistów systemu SQL Server, jest możliwość szybkiego generowania zaleceń pozwalających na poprawienie wydajności bazy danych. Nie trzeba przy tym znać schematu bazy, struktury danych, wzorców użytkowania, a nawet wewnętrznych mechanizmów optymalizatora kwerend z systemu SQL Server.

Od wersji SQL Server 2012 jako analizowanych operacji w DTA można używać pamięci podręcznej planów wykonywania kwerend. Ta nowa opcja sprawia, że nie trzeba dla analiz ręcznie generować operacji (np. plików ze śladem).

Koszt stosowania zbyt dużej liczby indeksów

Powiedzenie „co za dużo, to niezdrowo” jest prawdziwe także w kontekście indeksów. Zbyt duża liczba indeksów powoduje dodatkowe koszty związane z większą liczbą stron danych, które optymalizator kwerend musi sprawdzić. Ponadto zbyt liczne indeksy zajmują za dużo miejsca i wydłużają czas konserwacji.



Rysunek 14.7. Wybór operacji analizowanych przez narzędzie DTA

Jednak narzędzie DTA zwykle zaleca tworzenie dużej liczby indeksów; dzieje się tak zwłaszcza wtedy, gdy analizowane są operacje obejmujące dużą liczbę kwerend. Przyczyną jest to, że kwerendy są badane niezależnie od siebie. Dobrą praktyką jest stopniowe dodawanie indeksów, gdy są potrzebne. Zawsze należy przy tym mieć punkt odniesienia i na jego podstawie sprawdzać, czy nowy indeks poprawia wydajność kwerend.

System SQL Server 2014 udostępnia kilka widoków DMV z informacjami o używaniu indeksów. Oto te widoki.

- Widok `sys.dm_db_missing_index_details` zwraca szczegółowe informacje na temat brakującego indeksu.
- Widok `sys.dm_db_missing_index_columns` zwraca informacje o kolumnach tabeli, dla których brakuje indeksu.
- Widok `sys.dm_db_missing_index_groups` zwraca informacje o konkretnej grupie brakujących indeksów.
- Widok `sys.dm_db_missing_index_group_stats` zwraca podsumowanie informacji na temat grup brakujących indeksów.
- Widok `sys.dm_db_index_usage_stats` zwraca liczbę różnych typów operacji na indeksie i czas ostatniego wykonania każdej z nich.
- Widok `sys.dm_db_index_operational_stats` zwraca informacje o niskopoziomowych operacjach wejścia-wyjścia, blokadach, zamkach i metodach dostępu dla każdej partycji tabeli lub indeksu z bazy danych.
- Widok `sys.dm_db_index_physical_stats` zwraca informacje o wielkości i poziomie fragmentacji danych oraz indeksów określonej tabeli lub podanego widoku.

Aby np. uzyskać listę indeksów używanych i niestosowanych w kwerendach użytkowników, pobierz dane z widoku DMV `sys.dm_db_index_usage_stats`. Na podstawie listy używanych indeksów można pobrać ważne statystyki pomocne przy dostrajaniu indeksów. Zwracane informacje obejmują przykładowo wzorce dostępu do indeksu (operacje skanowania indeksu, przeszukiwania indeksu i wyszukiwania na podstawie klucza lub identyfikatora — ang. *bookmark lookup*). Pamiętaj, że ponowne utworzenie indeksu zmienia wyniki tej kwerendy, dlatego warto często ją uruchamiać.

Aby pobrać listę indeksów wykorzystywanych w kwerendach użytkownika, wywołaj poniższy skrypt:

```
SELECT
    SO.name Object_Name,
    SCHEMA_NAME(SO.schema_id) Schema_name,
    SI.name Index_name,
    SI.Type_Desc,
    US.user_seeks,
    US.user_scans,
    US.user_lookups,
    US.user_updates
FROM sys.objects AS SO
    JOIN sys.indexes AS SI
        ON SO.object_id = SI.object_id
    INNER JOIN sys.dm_db_index_usage_stats AS US
        ON SI.object_id = SI.object_id
        AND SI.index_id = SI.index_id
WHERE
    database_id=DB_ID('AdventureWorks')
    AND SO.type = 'u'
    AND SI.type IN (1, 2)
    AND (US.user_seeks > 0 OR US.user_scans > 0 OR US.user_lookups > 0 );
```

Jeśli chcesz wyświetlić listę indeksów niestosowanych w kwerendach użytkowników, wywołaj następujący skrypt:

```
SELECT
    SO.Name TableName,
    SI.name IndexName,
    SI.Type_Desc IndexType,
    US.user_updates
FROM sys.objects AS SO
    INNER JOIN sys.indexes AS SI
        ON SO.object_id = SI.object_id
    LEFT OUTER JOIN sys.dm_db_index_usage_stats AS US
        ON SI.object_id = US.object_id
        AND SI.index_id = US.index_id
WHERE
    database_id=DB_ID('AdventureWorks')
    AND SO.type = 'u'
    AND SI.type IN (1, 2)
    AND (US.index_id IS NULL)
    OR (US.user_seeks = 0 AND US.user_scans = 0 AND US.user_lookups = 0 );
```

Indeksy niewykorzystywane w kwerendach użytkowników należy usunąć, chyba że dodano je dla zadań o znaczeniu strategicznym wykonywanych w określonych momentach (np. w ramach miesięcznego lub kwartalnego pobierania danych i generowania raportów). Nieużywane indeksy zwiększają koszt operacji wstawiania, usuwania i aktualizowania, a także konserwacji.

Statystyki o używaniu indeksu są dostępne tylko tak długo, jak długo metadane indeksu (lub sterty) znajdują się w pamięci podręcznej metadanych. W momencie usuwania obiektu z pamięci podręcznej statystyki te są zerowane. Dzieje się tak np. wtedy, gdy usługa systemu SQL Server jest ponownie uruchamiana, gdy baza danych zostaje odłączona lub zamknięta lub gdy ustawiana jest właściwość `AUTO_CLOSE`.

Podsumowanie

W tym rozdziale poznałeś typy indeksów dostępne w systemie SQL Server, czyli indeksy wierszowe i indeksy kolumnowe, a także inne rodzaje indeksów (XML-owe, przestrzenne i pełnotekstowe).

Indeksy wierszowe to kombinacje kolumn zapisane w posortowanej kolejności i udostępniające wskaźnik do danych tabeli przechowywanych na poziomym liści. Indeks pokrywający to taki, który obejmuje wszystkie kolumny potrzebne do wykonania danej kwerendy.

Indeksy kolumnowe to oparte na kolumnach indeksy nieklastrowane przechowujące dane na podstawie wartości z danej kolumny. Takie indeksy zapewniają więcej korzyści niż zwykłe indeksy oparte na wierszach. Niektóre zalety indeksów kolumnowych to mniejsza wielkość i szybsze pobieranie danych. Od wersji SQL Server 2014 takie indeksy można modyfikować, a ponadto dostępna jest nowa opcja kompresji (`COLUMNSTORE_ARCHIVE`).

Ważnym aspektem indeksowania bazy jest optymalizacja dostępu do danych z wykorzystaniem partycji i indeksów oraz zaawansowanych technik indeksowania (np. indeksów filtrowanych i pokrywających). Reorganizowanie i ponowne tworzenie indeksów to ważne operacje z zakresu konserwacji pozwalające na zmniejszenie fragmentacji indeksów.

W ramach końcowego dopracowywania bazy możesz dostroić kwerendy za pomocą indeksów. Pomocne są przy tym widoki DMV. Warto też znaleźć indeksy niewykorzystywane w kwerendach użytkowników i usunąć je.

Po lekturze rozdziału 15. będziesz wiedział, jak za pomocą replikacji rozmieścić dane w różnych lokalizacjach.

Replikacja

ZAWARTOŚĆ ROZDZIAŁU:

- Różne typy replikacji.
- Modele replikacji.
- Konfigurowanie replikacji migawkowej.
- Konfigurowanie dystrybutora.
- Różnice pomiędzy replikacją migawkową, replikacją transakcyjną i ze skalaniem.
- Replikacja w trybie P2P.
- Monitorowanie replikacji.

Korporacje muszą rozsyłać dane do wielu działów i biur w różnych lokalizacjach. Mechanizmy replikacji z systemu SQL Server 2014 umożliwiają dystrybucję danych i obiektów bazodanowych do różnych baz danych z tego systemu, do baz innych producentów (takich jak Oracle), a także na urządzenia przenośne (smartfony, tablety i terminale w punktach sprzedaży). Replikacja obok przesyłania dzienników, tworzenia kopii lustrzanych bazy danych i klastrowania udostępnia mechanizmy, które zapewniają klientom równoważenie obciążenia, wysoką dostępność i skalowanie.

Rozdział ten stanowi wprowadzenie do procesu replikacji. Dowiesz się tu, jak zastosować prostą replikacją migawkową, a także na co powinieneś zwrócić uwagę przy konfigurowaniu replikacji transakcyjnej lub replikacji ze skalaniem.

Wprowadzenie do procesu replikacji

Replikacja w systemie SQL Server bardzo przypomina proces publikowania magazynu, dlatego przy omawianiu architektury replikacji wykorzystujemy tę analogię. Wyobraź sobie popularny magazyn. Duża grupa dziennikarzy pisze różne *artykuły*. Spośród nich redaktor wybiera teksty, które pojawią się w numerze w danym miesiącu. Wybrany zbiór artykułów trafia do *publikacji*. Po wydrukowaniu numer jest rozsyłany różnymi kanałami dystrybucji do *subskrybentów* z całego świata.

Przy omawianiu replikacji w systemie SQL Server używane są podobne określenia. Pulą, na podstawie której powstaje publikacja, jest *baza danych*. Każdy obiekt wybierany do publikacji to *artykuł*; może nim być tabela, procedura składowana lub inny obiekt bazodanowy. Nad magazynem pracuje redaktor, natomiast przy replikacji potrzebny jest *dystrybutor*, który udostępnia publikacje, śledzi status dostaw i rejestruje historię synchronizacji, aby zapewnić spójność danych.

W zależności od wybranego modelu replikacji publikowane artykuły mogą być zapisywane jako pliki w katalogu, do którego dostęp mają zarówno wydawca, jak i subskrybenci, lub jako rekordy w tabelach dystrybucyjnej bazy danych. Niezależnie od sposobu udostępniania publikacji w procesie replikacji zawsze potrzebny jest dystrybutor, który śledzi status dostaw. W zależności od pojemności serwera wydawcy baza danych dystrybutora może być zapisana po stronie wydawcy, po stronie subskrybenta lub na innym serwerze służącym wyłącznie do dystrybucji bazy danych.

Miedzy replikacją w systemie SQL Server a wydawaniem magazynu występują jednak różnice. Największa dotyczy funkcji kontrybutora, którą czasem pełni subskrybent. W niektórych modelach replikacji subskrybenci mogą aktualizować artykuły i przysyłać je do wydawcy lub innych subskrybentów. W modelu replikacji P2P każda jednostka pełni jednocześnie funkcję wydawcy i subskrybenta, dlatego zmiany wprowadzane w różnych bazach są replikowane na wielu serwerach.

Elementy w procesie replikacji

Wiesz już, jak proces replikacji wygląda. Pora zobaczyć, co różne pojęcia i funkcje znaczą w kontekście systemu SQL Server. Replikacja w tym systemie obejmuje kilka ważnych elementów pogrupowanych w następujące kategorie:

- role w procesie replikacji,
- dane replikacji,
- agenty replikacji,
- wewnętrzne komponenty replikacji (zadania z zakresu konserwacji).

Role w procesie replikacji

W procesie replikacji wyróżniane są trzy podstawowe role.

- **Wydawca.** To serwer (lub egzemplarz bazy danych), który jest źródłem publikowanych artykułów.
- **Dystrybutor.** To pośrednik w procesie wydawania i subskrybowania. W niektórych typach replikacji dystrybutor służy do przekazywania danych od wydawcy do subskrybenta. Dystrybutor przechowuje replikowane dane od wydawcy, a także rejestruje lokalizację

snapshotów (nazywanych też czasem migawkami). Dystrybutor to baza danych, która może znajdować się na serwerze wydawcy, na odrębnym specjalnym serwerze lub po stronie subskrybenta.

- **Subskrybent.** To serwer (lub egzemplarz bazy danych) będący jednostką docelową dla publikowanych artykułów. W niektórych modelach replikacji subskrybent może być jednocześnie wydawcą. Subskrybent może ponownie publikować artykuły. W modelu z subskrybentami aktualizującymi dane są one przesyłane do innego subskrybenta. Subskrybent może też ponownie publikować artykuły w modelu P2P (opisanym w punkcie „Typy replikacji”, dalej w tym rozdziale).

UWAGA Subskrypcja może działać w trybie wysyłania (ang. *push*) lub pobierania (ang. *pull*). Różnica dotyczy lokalizacji działania agenta. W subskrypcji z wysyłaniem agent działa po stronie wydawcy i wysyła zmiany do subskrybenta. W subskrypcji z pobieraniem agent jest uruchomiony po stronie subskrybenta i pobiera zmiany od wydawcy.

Dane replikacji

Oto trzy podstawowe komponenty danych replikacji.

- **Artykuł.** Jest to najmniejszy zbiór danych tworzony na potrzeby replikacji. Może obejmować tabelę, widok lub procedurę składowaną. Ponadto w każdym artykule można ustawić dodatkowe ograniczenia dotyczące wierszy i kolumn.
- **Publikacja.** To grupa publikowanych wspólnie artykułów. Zastosowanie publikacji umożliwia zarządzanie w trakcie replikacji logicznie pogrupowanymi artykułami. Dzięki temu nie trzeba zarządzać osobno każdym artykułem.
- **Subskrypcja.** To żądanie pobrania danych z jednej lub wielu publikacji. Można ustawić ograniczenia określające, jak i gdzie dane mają być udostępniane.

Agenty replikacji

Agenty replikacji to programy, które wykonują dużą część pracy związanej z replikacją. Często są uruchamiane za pomocą zadań narzędzia SQL Server Agent, ale można je też wywoływać ręcznie. W ramach replikacji tworzone są następujące zadania narzędzia SQL Server Agent.

- **Snapshot.** Agent jest uruchamiany przez zadanie narzędzia SQL Server Agent, które tworzy i stosuje snapshot w trzech typach replikacji: transakcyjnej, ze scalaniem i migawkowej. Te rodzaje replikacji są opisane szczegółowo w punkcie „Typy replikacji”, dalej w tym rozdziale. W replikacji transakcyjnej i ze scalaniem snapshot jest potrzebny tylko w momencie konfigurowania procesu replikacji i po dodaniu lub znacznym zmodyfikowaniu artykułu. W replikacji migawkowej ten agent jest uruchamiany przy każdej synchronizacji.
- **LogReader.** Agent jest uruchamiany przez zadanie narzędzia SQL Server Agent, które wczytuje dziennik transakcji po stronie wydawcy i rejestruje transakcje dotyczące każdego artykułu opublikowanego w udostępnianej bazie danych.
- **Distribution.** Agent jest uruchamiany przez zadanie narzędzia SQL Server Agent, które wczytuje transakcje zapisane w dystrybucyjnej bazie danych, a następnie stosuje je do baz danych subskrybentów w ramach replikacji transakcyjnej.

- **Merge.** Agent jest uruchamiany przez zadanie narzędzia SQL Server Agent, które przenosi zmiany z serwera wydawcy do serwera subskrybenta lub z serwera subskrybenta do serwera wydawcy, a także — jeśli trzeba — inicjuje proces rozwiązywania konfliktów. Ten agent jest używany w ramach replikacji ze scalaniem.
- **QueueReader.** Agent służy do wczytywania komunikatów zapisanych w kolejce systemu SQL Server lub kolejce komunikatów Microsoftu. Agent następnie stosuje te komunikaty po stronie wydawcy. Ten agent jest używany dla publikacji w ramach replikacji migawkowej lub transakcyjnej, co umożliwia aktualizowanie danych z wykorzystaniem kolejki (szczegółowe omówienie replikacji migawkowej i transakcyjnej zamieszczono w punkcie „Typy replikacji”, dalej w tym rozdziale).

Zadania związane z konserwacją w ramach replikacji

W ramach replikacji do przeprowadzania konserwacji używane są następujące zadania narzędzia SQL Server Agent.

- **History Cleanup.** Zadanie usuwa historię agenta replikacji zapisaną w dystrybucyjnej bazie danych. Jest uruchamiane co 10 minut.
- **Distribution Cleanup.** Zadanie usuwa transakcje z dystrybucyjnej bazy danych, gdy nie są już potrzebne. Jest uruchamiane co 10 minut.
- **Subscription Cleanup.** Zadanie określa moment wygasania snapshota i usuwa go. Jest uruchamiane każdego dnia o 1.00 w nocy.
- **Reinitialize Failed Subscriptions.** Zadanie wyszukuje nieudane subskrypcje i oznacza je jako przeznaczone do ponownego zainicjowania. Domyślnie to zadanie nie jest włączone, dlatego możesz albo uruchamiać je ręcznie, gdy jest potrzebne, albo utworzyć niestandardowy harmonogram.
- **Monitor.** Zadanie monitoruje pracę agentów i zapisuje w dzienniku zdarzeń systemu Windows nieudane kroki zadań. Jest uruchamiane co 10 minut.
- **Checkup.** Zadanie monitoruje działanie agentów replikacji po stronie dystrybutora. Sprawdzane są agenty uruchomione, które jednak nie zarejestrowały informacji w historii w danym przedziale czasu.

Typy replikacji

W systemie SQL Server dostępne są trzy typy replikacji:

- replikacja migawkowa,
- replikacja transakcyjna,
- replikacja ze scalaniem.

Replikacja w trybie P2P i replikacja z publikacją w bazach Oracle (Oracle Publishing) to odmiany tych trzech typów omówione w odpowiednich punktach.

Replikacja migawkowa

Jak wskazuje nazwa, replikacja migawkowa tworzy snapshot (nazywany też migawką) publikacji i udostępnia go subskrybentom. Gdy snapshot jest dodawany do bazy subskrybenta, artykuły w tej bazie (tabele, widoki i procedury składowane) są usuwane i ponownie tworzone. Replikacja

migawkowa to jednorazowa operacja. Nie ma tu ciągłego strumienia danych przesyłanego od wydawcy do subskrybenta. Dane z serwera wydawcy dostępne w momencie tworzenia snapshota są następnie przenoszone do subskrybenta.

Replikacja migawkowa najlepiej nadaje się do stosunkowo statycznych danych, gdy akceptowalne jest używanie nieaktualnych kopii danych w okresie między replikacjami lub gdy artykuły są niewielkie. Załóżmy, że potrzebne są tabele z kodami pocztowymi. Zwykle dobrze nadają się do zastosowania w replikacji migawkowej, ponieważ są statyczne.

OSTRZEŻENIE W czasie odświeżania snapshota artykuł jest niedostępny.

Replikacja transakcyjna

W tym podejściu zmiany są replikowane w czasie ich wprowadzania. Aby skonfigurować replikację transakcyjną, najpierw trzeba utworzyć snapshot publikacji i przesłać go do subskrybentów, aby otrzymać po ich stronie ten sam zbiór danych. Po utworzeniu snapshota agent LogReader wczytuje wszystkie transakcje dotyczące opublikowanego artykułu i rejestruje je w dystrybucyjnej bazie danych. Transakcje te są potem stosowane u każdego subskrybenta zgodnie z konfiguracją subskrypcji.

Replikacja transakcyjna umożliwia szybsze synchronizowanie danych z mniejszym opóźnieniem. W zależności od tego, jak jest skonfigurowana, synchronizacja może zachodzić niemal w czasie rzeczywistym. Dlatego to podejście jest przydatne, gdy przyrostowe zmiany mają być szybko wprowadzane po stronie subskrybenta.

Replikacja ze scalaniem

Replikację ze scalaniem stosuje się zwykle wtedy, gdy połączenie sieciowe między wydawcą a subskrybentem jest wolne lub przerywane. W tym modelu poszczególne jednostki mogą działać stosunkowo niezależnie i synchronizować zmiany w danych, gdy pracują w trybie online. Przy inicjowaniu replikacji potrzebny jest snapshot, a późniejsze zmiany są śledzone za pomocą wyzwalaczy.

Efektem ubocznym replikacji ze scalaniem jest możliwość występowania konfliktów w momencie synchronizowania zmian wprowadzonych w trybie offline. Takie konflikty są rozwiązywane automatycznie za pomocą agenta Merge, który stosuje technikę wybraną w trakcie tworzenia publikacji. Jeśli nie chcesz korzystać z automatycznego rozwiązywania konfliktów, możesz skonfigurować dla publikacji interaktywny odpowiednik tego procesu. W takiej sytuacji każdy konflikt trzeba rozwiązać ręcznie. Można wykorzystać do tego interfejs Interactive Resolver.

Inne „typy” replikacji

Następne dwie możliwości często uważa się za dodatkowe typy replikacji, jednak nimi nie są — to jedynie odmiany wcześniej opisanych podejść.

- **Replikacja w trybie P2P.** Jest to podtyp replikacji transakcyjnej. W tym modelu każdy wydawca jest właścicielem podzbioru wszystkich wierszy artykułu. Każda jednostka publikuje własne wiersze, a wszystkie pozostałe jednostki je subskrybują. Inne wiersze dana jednostka sama subskrybuje od pozostałych jednostek. Więcej o replikacji transakcyjnej w trybie P2P dowiesz się dalej w tym rozdziale.

- **Replikacja z publikacją w bazach Oracle.** W systemie SQL Server 2000 umożliwiono subskrypcję danych publikowanych w bazach Oracle. W tym trybie mogą działać replikacja migawkowa i transakcyjna; nie jest dostępny dla replikacji ze scalaniem.

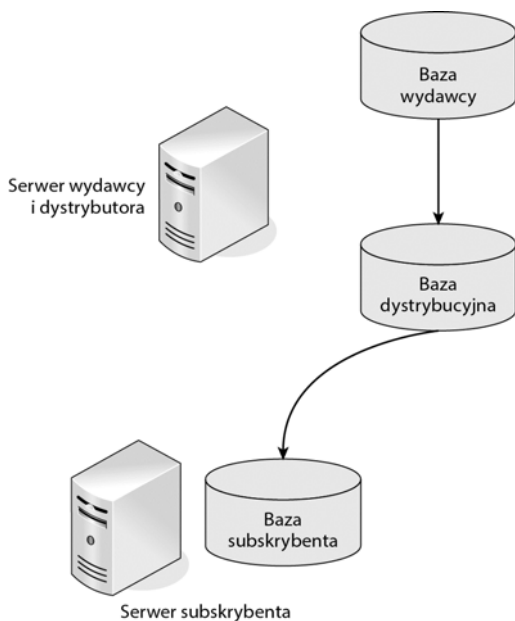
UWAGA W tym rozdziale nie omawiamy replikacji programowej z wykorzystaniem obiektów RMO (ang. *Replication Management Object*). Obiekty te tworzą model programowania replikacji w systemie SQL Server za pomocą kodu zarządzanego. Wszystkie kroki i procesy omówione w tym rozdziale można zaprogramować przy użyciu obiektów RMO. Aby lepiej je zrozumieć, zapoznaj się z przykładowymi aplikacjami i kodem z dokumentacji Books Online.

Modele replikacji

Replikację można skonfigurować na kilka różnych sposobów. W tym podrozdziale omawiamy najczęściej stosowane rozwiązania. Istnieją podstawowe cegiełki, z których można budować bardziej skomplikowane architektury.

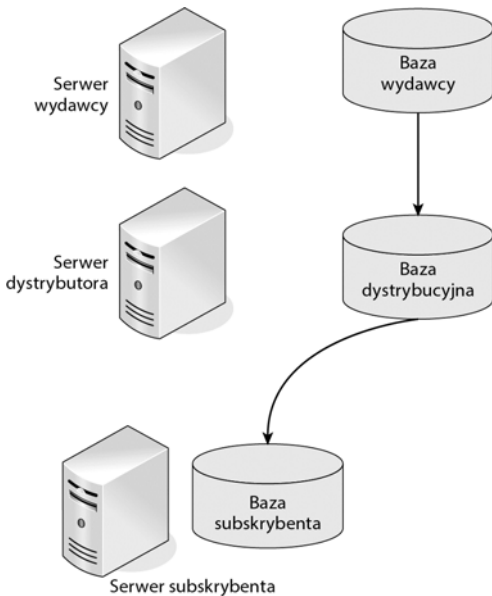
Jeden wydawca i dowolna liczba subskrybentów

Model z jednym wydawcą to najprostsza architektura stosowana wtedy, gdy istnieje jedna baza z publikacją i dowolna liczba baz subskrybentów. To rozwiązanie można zastosować, gdy potrzebny jest aktywny system rezerwowý lub trzeba rozsyłać dane z centralnego biura do wielu filii. Na rysunku 15.1 przedstawiono podstawową strukturę tej architektury, w której dystrybutor znajduje się na serwerze wydawcy.



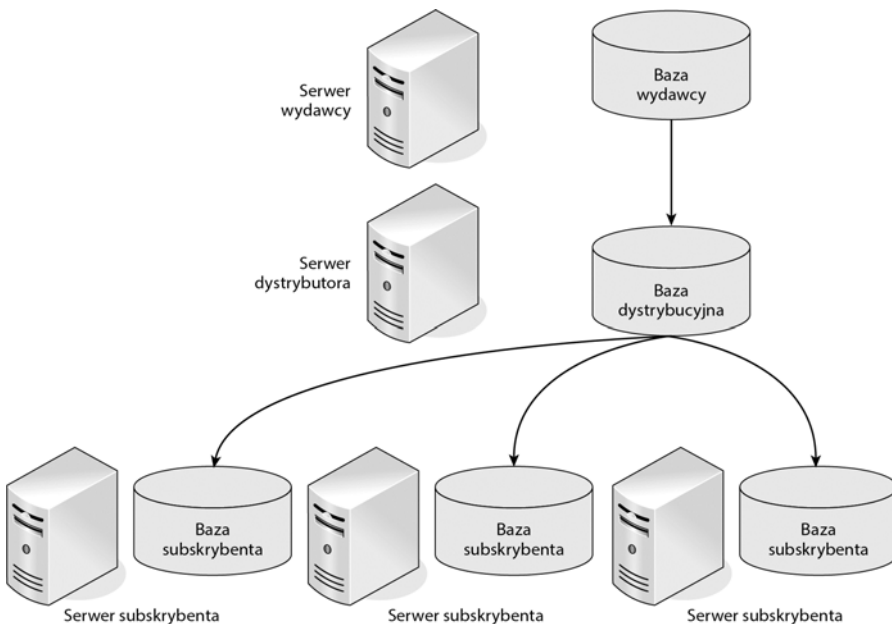
Rysunek 15.1. Architektura z jednym wydawcą

Na rysunku 15.2 pokazano bardziej skomplikowane rozwiązanie. Dystrybucyjna baza danych znajduje się na odrębnym serwerze. To podejście należy stosować, gdy dystrybutor musi być bardziej wydajny niż przy używaniu jednego serwera jako wydawcy i dystrybutora.



Rysunek 15.2. Wersja z dystrybutorem na odrębnym serwerze

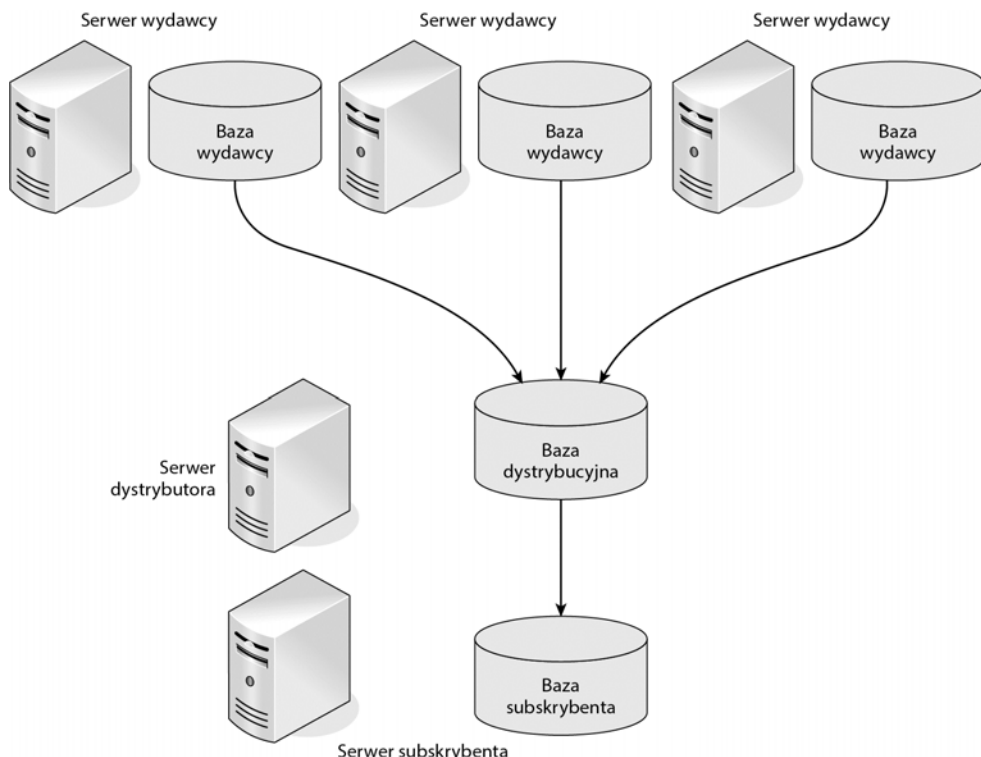
Na rysunku 15.3 przedstawiono następną wersję modelu z jednym wydawcą. Tu występuje wiele baz subskrybentów.



Rysunek 15.3. Wersja z wieloma subskrybentami

Wielu wydawców i jeden subskrybent

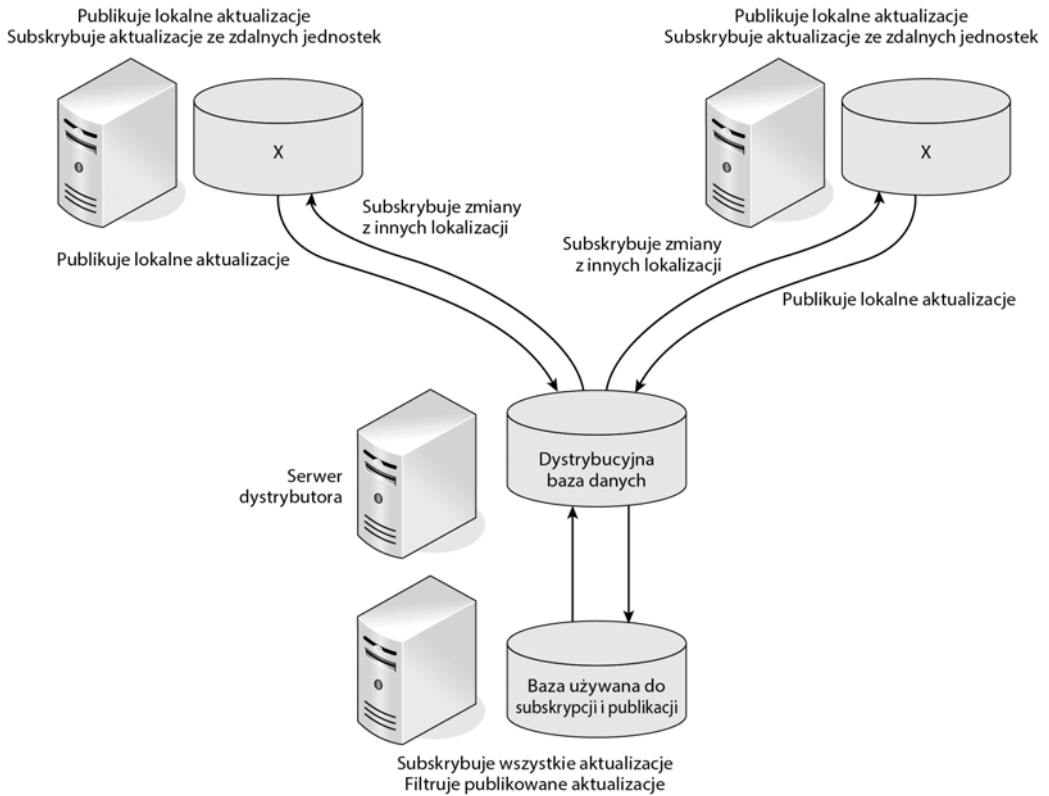
Aplikacje POS (ang. *Point of Service*) to dobry przykład modelu z wieloma wydawcami. Aplikacja POS ma wielu wydawców, ale tylko jednego subskrybenta. W tym podejściu często trzeba przysłać dane z wielu terminali POS do jednego centralnego systemu w sklepie lub głównym biurze, gdzie można skonsolidować poszczególne transakcje. Jest to model z wieloma wydawcami i jednym subskrybentem. Jego architekturę przedstawiono na rysunku 15.4.



Rysunek 15.4. Model z wieloma wydawcami i jednym subskrybentem

Wielu wydawców będących jednocześnie subskrybentami

Aplikacje do zarządzania relacjami z klientami (ang. *Customer Resource Management* — CRM) to dobry przykład zastosowania modelu z wieloma wydawcami będącymi jednocześnie subskrybentami. W aplikacjach CRM potrzebna może być książka adresowa z danymi kontaktowymi — lokalnie aktualizowana, ale synchronizowana między wszystkimi jednostkami. Aby uzyskać pożądany efekt, każda filia może publikować wprowadzone przez siebie modyfikacje, a także subskrybować zmiany dodane przez inne jednostki. Na rysunku 15.5 przedstawiono to rozwiązanie.



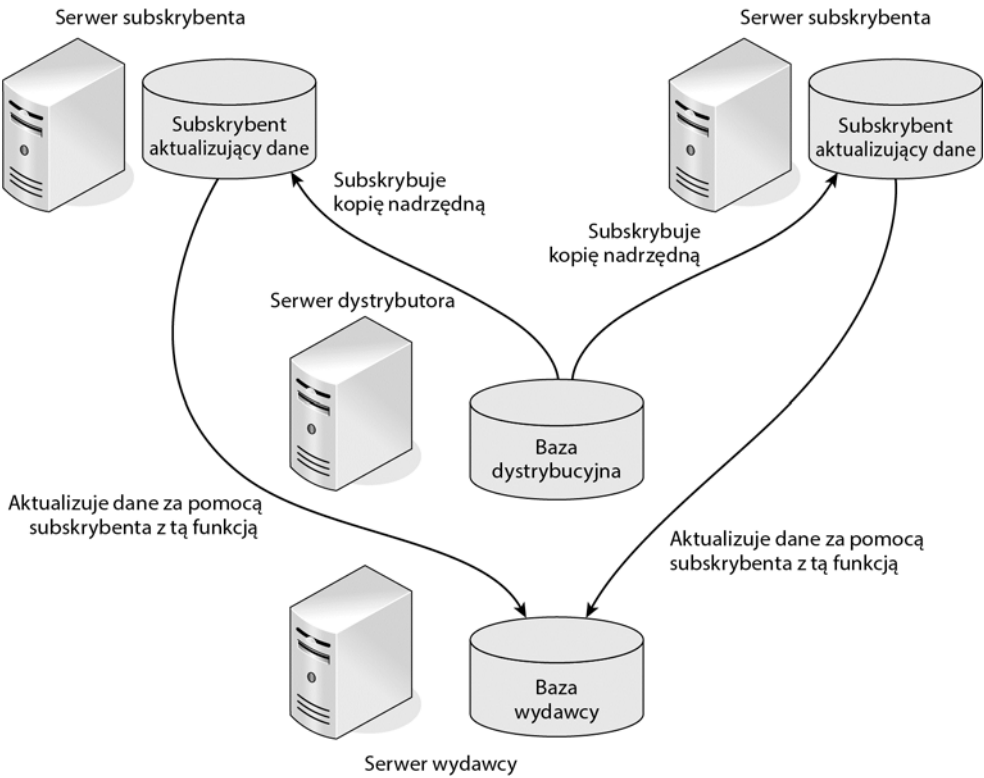
Rysunek 15.5. Subskrybenci mogą jednocześnie publikować dane

Subskrybent aktualizujący dane

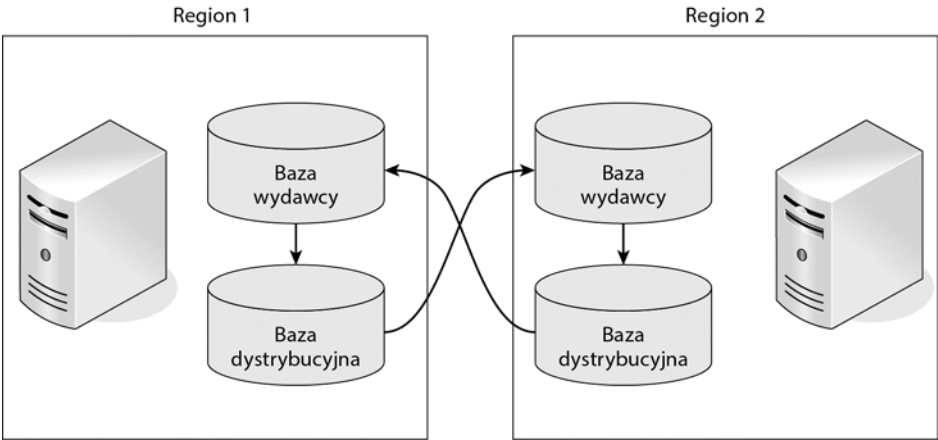
Aplikację CRM można też zaimplementować za pomocą modelu z subskrybentem aktualizującym dane. W tej architekturze nadrzędna kopia danych kontaktowych przechowywana w centralnej lokalizacji jest publikowana we wszystkich filiach. Wprowadzenie zmian w filii prowadzi do aktualizacji danych po stronie wydawcy za pomocą wbudowanego w proces replikacji mechanizmu subskrybenta aktualizującego dane. Na rysunku 15.6 przedstawiono to rozwiązanie.

Model P2P

Także architekturę P2P można opisać na przykładzie aplikacji CRM. W tym podejściu nie istnieje nadrzędna kopia danych. Każda jednostka zawiera własny zestaw wierszy i otrzymuje aktualizacje wprowadzone przez inne jednostki. Tę architekturę przedstawiono na rysunku 15.7.



Rysunek 15.6. Model z subskrybentami, którzy mogą aktualizować dane



Rysunek 15.7. Architektura P2P

Implementowanie replikacji

Skoro znasz już terminologię, typy i modele, pora zaimplementować replikację. Najpierw zobaczysz, jak skonfigurować replikację migawkową. Replikacja transakcyjna i replikacja ze scalaniem działają podobnie, jednak różnią się pewnymi elementami. Różnice te przedstawimy po omówieniu konfigurowania replikacji migawkowej.

UWAGA Warto przypomnieć, że bezpośrednio po zainstalowaniu systemu SQL Server wszystkie zewnętrzne protokoły są wyłączone. Aby komunikować się z systemem SQL Server, musisz uruchomić narzędzie SQL Server Configuration Manager, włączyć jeden z zewnętrznych protokołów (*Named Pipes* lub *TCP/IP*), a następnie ponownie uruchomić system SQL Server.

Konfigurowanie replikacji migawkowej

Żeby nie komplikować zadania, w tym przykładzie utworzysz tylko dwie nowe bazy danych — bazę wydawcy *Publisher* i bazę subskrybenta *Subscriber*. W tym przykładzie będziesz potrzebował schematu *sales* i utworzonej w nim tabeli *Cars*. Po opracowaniu tabeli wstaw do niej niewielki zbiór wierszy. Ten scenariusz opisuje producenta samochodów hybrydowych o niskim zużyciu paliwa. Są one sprzedawane w Stanach Zjednoczonych, Chinach i Szwecji. Zadanie polega na skonfigurowaniu replikacji migawkowej na serwerach bazodanowych w Stanach Zjednoczonych i Chinach, co ma umożliwić odświeżanie danych. Ponadto można skonfigurować replikację transakcyjną dla serwerów bazodanowych ze Stanów Zjednoczonych i Szwecji. Dane można także wykorzystać w ramach replikacji ze scalaniem.

Replikację można zaimplementować za pomocą kreatora z interfejsem graficznym lub przy użyciu skryptów. Jeśli dopiero poznajesz replikację, powinieneś zacząć od interfejsu graficznego i stron właściwości z programu SQL Server Management Studio. W trakcie korzystania z interfejsu graficznego możesz wygenerować skrypty w języku SQL, zapisać je w pliku i zmodyfikować dla innych komputerów.

Zanim utworzysz publikację i je zasubskrybujesz, musisz zaimplementować dystrybutor. Zaczynj więc od przygotowania dystrybutora (jest potrzebny we wszystkich typach replikacji).

Konfigurowanie dystrybucji

Dystrybutor obejmuje dystrybucyjną bazę danych (z historią replikacji, statusem i innymi ważnymi informacjami), a także współużytkowany katalog (służący do przechowywania, pobierania i odświeżania danych oraz artykułów).

Dystrybucyjną bazę danych możesz umieścić na serwerze wydawcy (jest to lokalizacja domyślna) lub na odrębnym serwerze. Jeśli liczba transakcji jest niewielka, serwer wydawcy dobrze się sprawdzi. Gdy liczba transakcji jest duża, zastosowanie odrębnego serwera pozwala uzyskać wyższą wydajność.

Aby rozpocząć konfigurowanie dystrybucji, musisz określić nazwę domeny i konto używane do uruchamiania różnych agentów replikacji (takich jak *Snapshot*, *Log Reader* i *Queue Reader*). W tym przykładzie możesz wykorzystać konto narzędzia SQL Server Agent, jednak w środowisku produkcyjnych z przyczyn bezpieczeństwa zaleca się stosowanie dedykowanego konta domenowego.

Poniżej opisujemy krok po kroku proces konfigurowania dystrybucji.

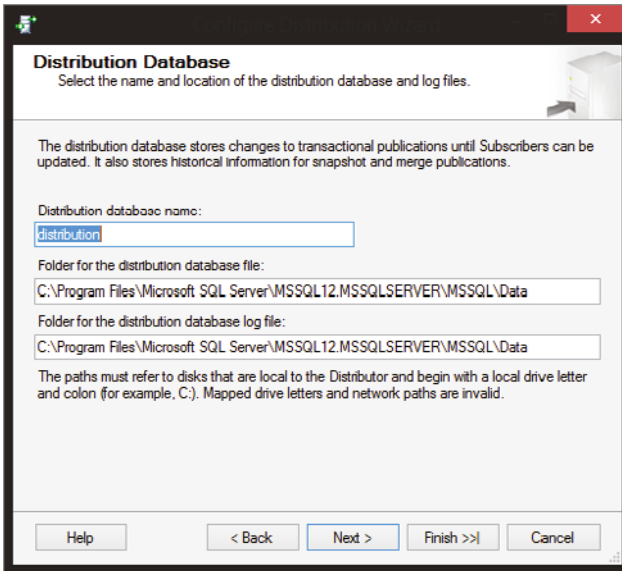
1. Połącz się z serwerem dystrybutora w programie SQL Server Management Studio. Rozwiń węzeł tego serwera w oknie *Object Explorer*.
2. Kliknij prawym przyciskiem myszy węzeł *Replication* i wybierz opcję *Configure Distribution*.

OSTRZEŻENIE Bezpośrednio po zainstalowaniu systemu SQL Server 2014 komponent Agent XPs jest wyłączony. To sprawia, że nie można uruchomić usługi SQL Server Agent. Jeśli ta usługa nie działa, przy próbie wykonania kroku 2. wystąpi błąd. Aby temu zapobiec, włącz komponent Agent XPs za pomocą poniższego skryptu, a następnie uruchom usługę SQL Server Agent (możesz to zrobić w programach SQL Server Management Studio lub SQL Server Configuration Manager):

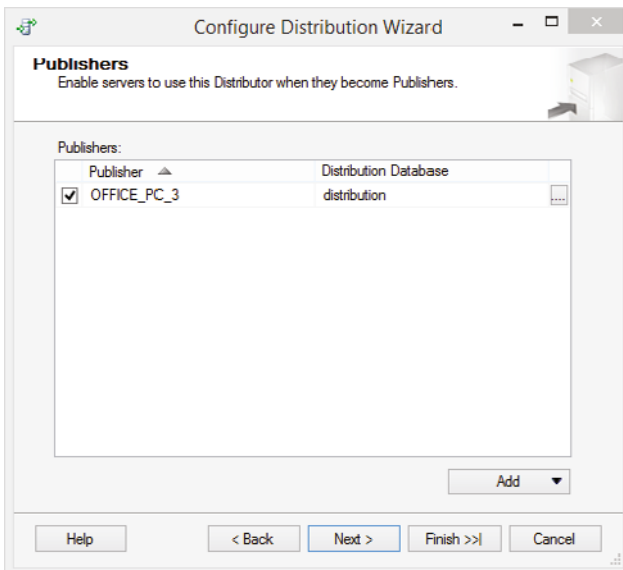
```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Agent XPs', 1;
GO
RECONFIGURE
GO
```

Po wykonaniu tego skryptu dla węzła SQL Server Agent w programie SQL Server Management Studio nie będzie się już pojawiać informacja *Agent XPs disabled*.

3. Na ekranie powitalnym kliknij przycisk *Next*. Pojawi się ekran *Distributor*, gdzie można ustawić serwer pełniący funkcję dystrybutora. W tym przykładzie użyj bieżącego komputera. W bardziej skomplikowanych architekturach możesz zastosować odrębny serwer.
4. Kliknij przycisk *Next*. Pojawi się strona *SQL Server Agent Start*. Tu można ustawić sposób uruchamiania narzędzia SQL Server Agent — automatyczny lub ręczny. Wybierz opcję automatycznego uruchamiania.
5. Kliknij przycisk *Next*. Pojawi się ekran *Snapshot Folder*. Tu możesz podać katalog, w którym zapisany zostanie snapshot. Jeśli chcesz stosować subskrypcję z pobieraniem, wprowadź ścieżkę sieciową. W tym przykładzie replikacja jest przeprowadzana lokalnie, dlatego ścieżka sieciowa nie jest konieczna.
6. Po wybraniu katalogu na snapshot przejdź do ekranu *Distribution Database*, przedstawionego na rysunku 15.8. Można tu podać nazwę dystrybucyjnej bazy danych (tu ta nazwa to *distribution*) i lokalizację katalogów przeznaczonych na pliki bazy i dzienników. Kliknij przycisk *Next*, aby przejść dalej.
7. Teraz pojawi się ekran *Publishers*, widoczny na rysunku 15.9. Można na nim ustawić listę serwerów, które mogą korzystać z dystrybucyjnej bazy danych.
8. Następnie na ekranie *Wizard Actions* (patrz rysunek 15.10) możesz skonfigurować, co kreator powinien zrobić. Kreator może skonfigurować dystrybucyjną bazę danych lub utworzyć skrypty. Jeśli chcesz wygenerować skrypty, zaznacz obie opcje.

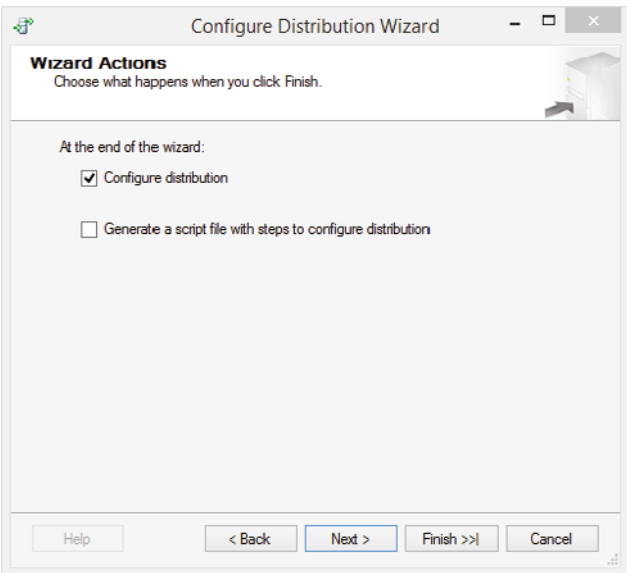


Rysunek 15.8. Tu możesz podać nazwę bazy i lokalizację katalogów przeznaczonych na pliki

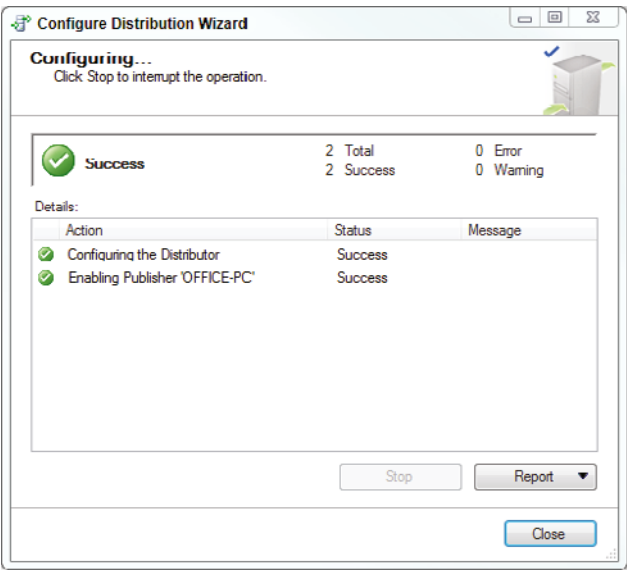


Rysunek 15.9. Tu możesz podać listę serwerów używających danej bazy

9. Następna strona kreatora wyświetla listę działań, jakie może podjąć system SQL Server. Zatwierdź tę listę i kliknij przycisk *Finish*. Jeśli lista nie jest prawidłowa, cofnij się i popraw błędne ustawienia. Następnie zobaczysz ekran z informacjami o postępach kreatora w wykonywaniu wybranych zadań.
10. Na rysunku 15.11 przedstawiono informacje o udanym wykonaniu operacji przez kreator. Jeśli wystąpiły błędy, zbadaj je i napraw, a dopiero potem ponownie uruchom kreator.

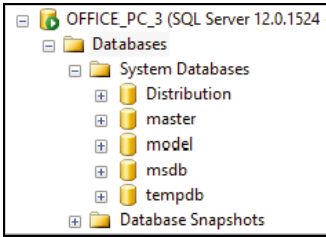


Rysunek 15.10. Określanie, jakie operacje kreator powinien wykonać

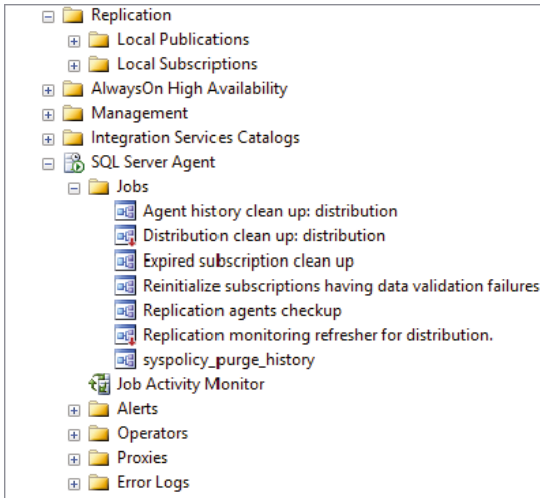


Rysunek 15.11. Informacje o udanym zakończeniu pracy przez kreator

Po wykonaniu zadań przez kreator możesz przejść do nowej dystrybucyjnej bazy danych w oknie *Object Explorer* w programie SQL Server Management Studio. Nowa baza pojawia się w węźle *System Databases*, co ilustruje rysunek 15.12. Ponadto możesz rozwinąć w tym oknie węzeł *SQL Server Agent* i otworzyć katalog *Jobs*. Zobaczysz wtedy, że na potrzeby replikacji utworzone zostały liczne nowe zadania (patrz rysunek 15.13).



Rysunek 15.12. Nowa dystrybucyjna baza danych w węźle System Databases



Rysunek 15.13. Zadania dodane na potrzeby replikacji

Implementowanie replikacji migawkowej

Po przygotowaniu dystrybutora można wykorzystać go do skonfigurowania replikacji dalej w tym rozdziale. Zanim jednak przejdziesz do konfigurowania replikacji migawkowej, musisz utworzyć bazy danych wydawcy i subskrybenta.

W tym przykładzie obie bazy znajdą się na jednym serwerze. Skrypt potrzebny do ich utworzenia znajdziesz w pliku *R15_01_Tworzenie_baz.sql* z przykładowym kodem. Możesz też dodać te bazy ręcznie w programie SQL Server Management Studio. Aby ręcznie utworzyć bazy, wykonaj następujące kroki.

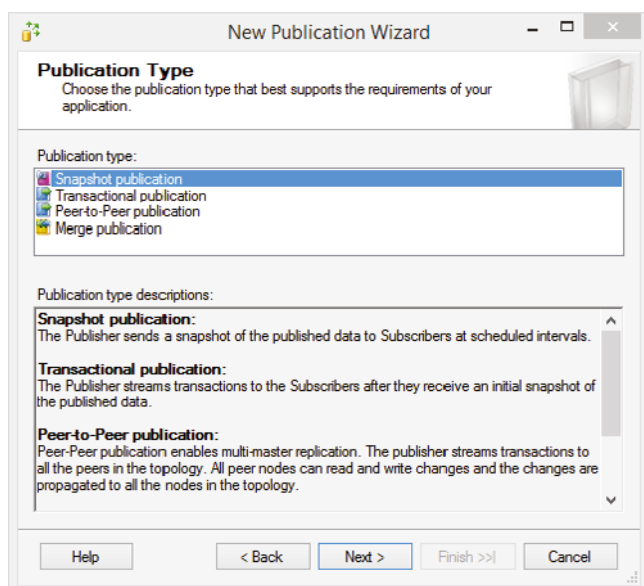
1. Kliknij węzeł *Databases* prawym przyciskiem myszy i wybierz opcję *New Database*.
2. Bazę danych wydawcy nazwij *Publisher*, a bazę subskrybenta — *Subscriber*. Dla pozostałych ustawień możesz pozostawić wartości domyślne.
3. Po utworzeniu bazy wydawcy musisz przygotować w niej tabele i zapisać w nich dane. W tym celu pobierz i uruchom skrypt z pliku *R15_02_Tworz_wczyt_Sales.sql*.

Po wykonaniu tych wstępnych kroków możesz przejść do publikowania snapshota, co opisano w następnym kroku.

Konfigurowanie publikacji snapshota

Najlepszą metodą konfigurowania publikacji snapshota jest wykorzystanie programu SQL Server Management Studio i wygenerowanie skryptów w końcowej części procesu. Wykonaj następujące kroki.

1. W programie SQL Server Management Studio połącz się z serwerem z bazą wydawcy i rozwiń węzeł *Replication*. Kliknij prawym przyciskiem myszy element *Local Publications* i wybierz opcję *New Publication*.
2. Pojawi się ekran powitalny kreatora publikacji. Kliknij przycisk *Next*. Na ekranie *Publication Database* wybierz bazę danych przeznaczoną na publikację. Ustaw bazę Publisher i kliknij przycisk *Next*.
3. Na rysunku 15.14 przedstawiono ekran *Publication Type*, na którym można wybrać typ replikacji. Tu wybierz *Snapshot publication* i kliknij przycisk *Next*, aby kontynuować.

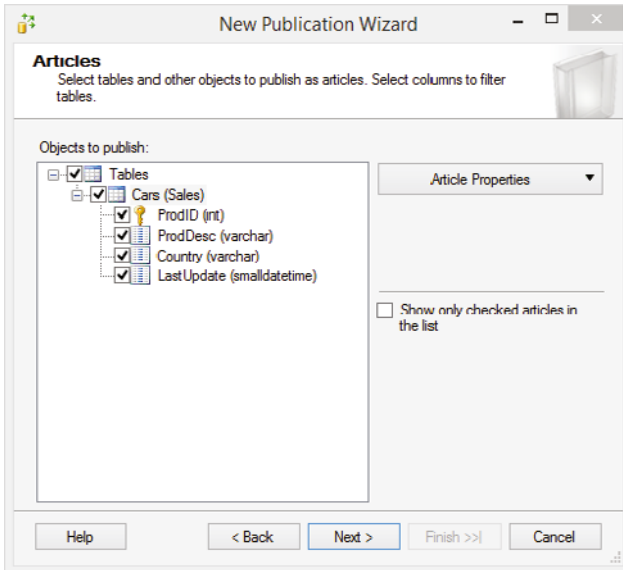


Rysunek 15.14. Ustawianie typu publikacji

4. Następny ekran to *Articles*. Można na nim wybrać tabele do danego artykułu. Domyślnie nie są wybrane żadne elementy, dlatego trzeba rozwinąć węzeł *Tables* i wskazać tabele przeznaczone do opublikowania. Po rozwinięciu węzła zaznacz tabelę *Cars* (utworzoną wcześniej dla publikacji) i jej elementy podrzędne, co można zobaczyć na rysunku 15.15. Widać na nim, że zaznaczona jest tabela *Cars* ze schematu *Sales*. Jeśli trzeba, możesz też wybrać kolumny tabeli. W tym celu usuń zaznaczenie pól obok nazw kolumn, które chcesz pominąć.

Możesz też ustawić właściwości publikowanych artykułów. Właściwości wpływają na sposób publikowania artykułu i przebieg synchronizacji z subskrybentami. Tu nie zmieniaj właściwości domyślnych, pamiętaj jednak, że w innych sytuacjach mogą być przydatne. Kliknij przycisk *Next*, aby kontynuować.

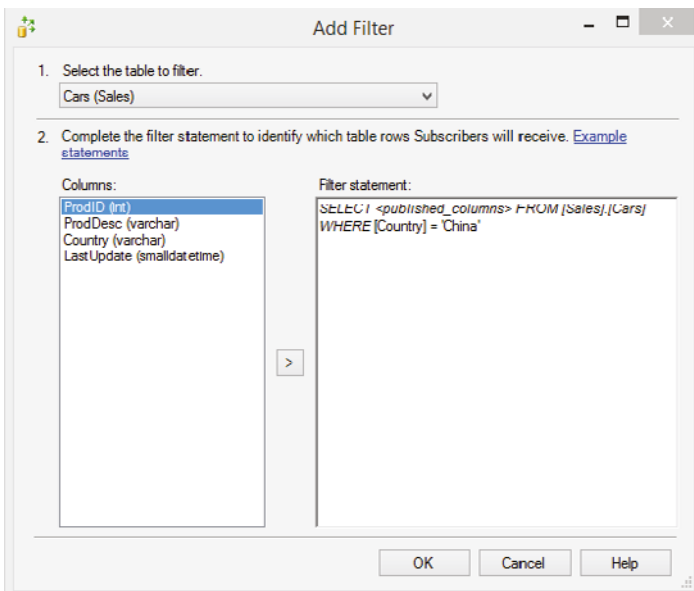
5. Na stronie *Filter Table Rows* kreator umożliwia przefiltrowanie wierszy. Kliknij przycisk *Add*, aby wyświetlić stronę *Add Filter*.



Rysunek 15.15. Wybieranie publikowanych tabel

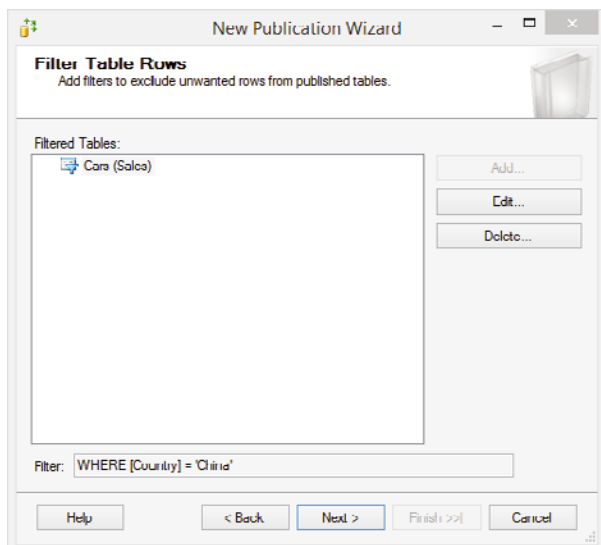
6. Ponieważ dane mają być replikowane w bazie w Chinach, trzeba przefiltrować samochody według państw. W tym celu dodaj klauzulę `WHERE` do instrukcji filtrującej, aby jej kod w języku SQL wyglądał tak (patrz rysunek 15.16):

```
SELECT <published_columns>
FROM [Sales].[Cars]
WHERE [Country] = 'China'
```



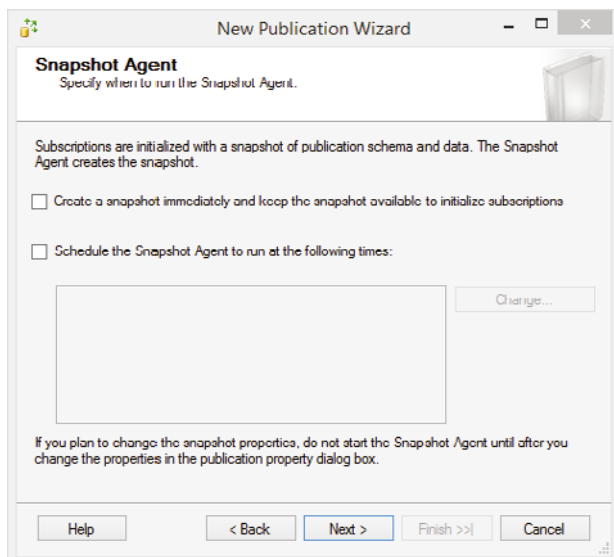
Rysunek 15.16. Filtrowanie publikowanych danych

Po kliknięciu przycisku *OK* filtr zostanie zastosowany. Spowoduje to powrót do poprzedniego ekranu, na którym filtr jest już teraz zdefiniowany, co przedstawiono na rysunku 15.17. Kliknij przycisk *Next*, aby kontynuować.



Rysunek 15.17. Ekran po zdefiniowaniu filtra

7. Na rysunku 15.18 widać ekran *Snapshot Agent*, gdzie można zdefiniować sposób generowania snapshota i czas wykonywania tej operacji w harmonogramie. Tu nie planuj wykonywania tej operacji ani na razie nie twórz snapshota. W tym przykładzie uruchomisz ten proces ręcznie za pomocą zadania narzędzia SQL Server Agent. Kliknij przycisk *Next*, aby kontynuować.



Rysunek 15.18. Konfigurowanie procesu generowania snapshota

8. Następny ekran to *Snapshot Agent Security*. Można tu określić konta używane do uruchamiania zadania agenta Snapshot i łączenia się z wydawcą. W różnych modelach replikacji stosowane są inne agenty. Kliknij przycisk *Security Settings*, aby skonfigurować konta potrzebne w danym modelu replikacji.

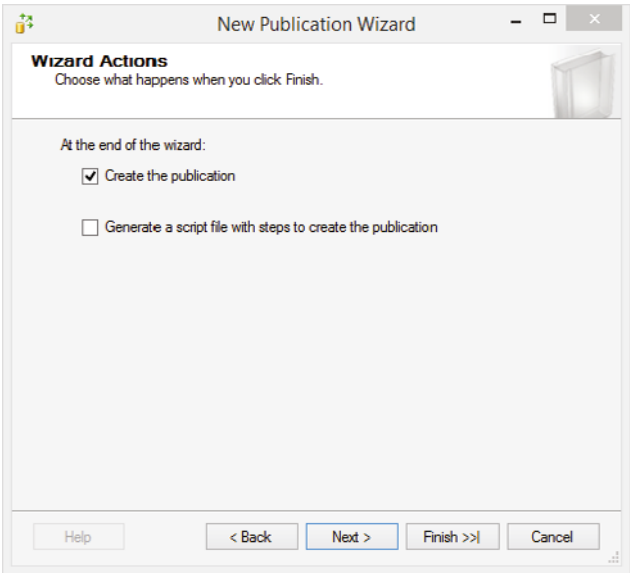
Dla replikacji warto utworzyć dedykowane konto domenowe z bezpiecznym hasłem, którego nie trzeba często zmieniać. Jeśli jednak jest to niemożliwe, możesz wykonywać zadania w kontekście konta narzędzia SQL Server Agent. Choć w tym przykładzie zastosujesz właśnie to rozwiązanie, lepiej zawsze korzystać z dedykowanego konta.

Jeśli zdecydujesz się zastosować dedykowane konto systemu Windows, musi być ono przypisane do roli db_owner dystrybucyjnej bazy danych i mieć uprawnienia do zapisu we współużytkowanym zasobie, na którym przechowywany jest snapshot.

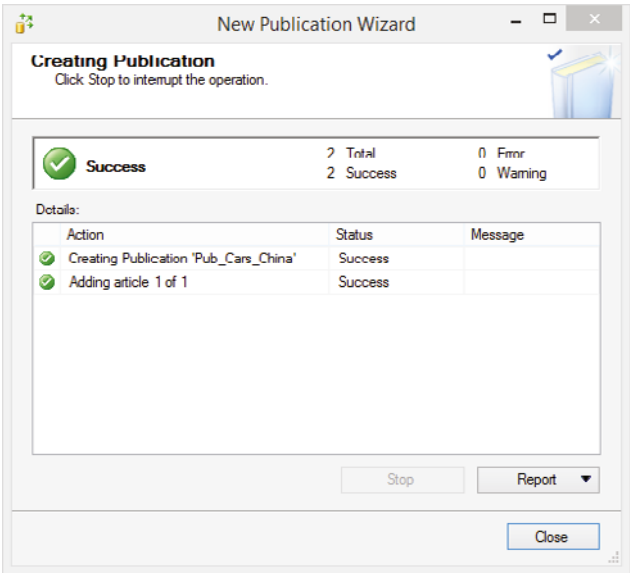
Wprowadź informacje o koncie na ekranie *Snapshot Agent Security* (patrz rysunek 15.19). Po skonfigurowaniu konta kliknij przycisk OK. Na stronie *Agent Security* kliknij przycisk *Next*.

Rysunek 15.19. Konfigurowanie konta

9. Na rysunku 15.20 przedstawiono ekran *Wizard Actions*, gdzie można określić, czy kreator ma utworzyć publikację, czy wygenerować tworzący ją skrypt. Na razie pozostaw ustawienia domyślne i utwórz publikację.
10. Na następnym ekranie należy zatwierdzić operacje kreatora. Można tu też przypisać nazwę publikacji. Wprowadź nazwę **Pub_Cars_China** w polu tekstowym *Publication Name*. Przyjrzyj się wyświetlonym operacjom. Jeśli zauważysz błędy, cofnij się i je popraw. Jeżeli wszystko wygląda prawidłowo, kliknij przycisk *Finish*.
11. Następny ekran (patrz rysunek 15.21) wyświetla potwierdzenie operacji. Tutaj te operacje to utworzenie publikacji i dodanie artykułów (w przykładzie jest tylko jeden artykuł). Pole stanu powinno informować o sukcesie i udanym utworzeniu nowej publikacji. Jeśli pojawiły się błędy, zbadaj je i popraw, a następnie ponownie spróbuj wykonać potrzebne operacje.



Rysunek 15.20. Na tej stronie możesz wygenerować skrypty tworzące publikację

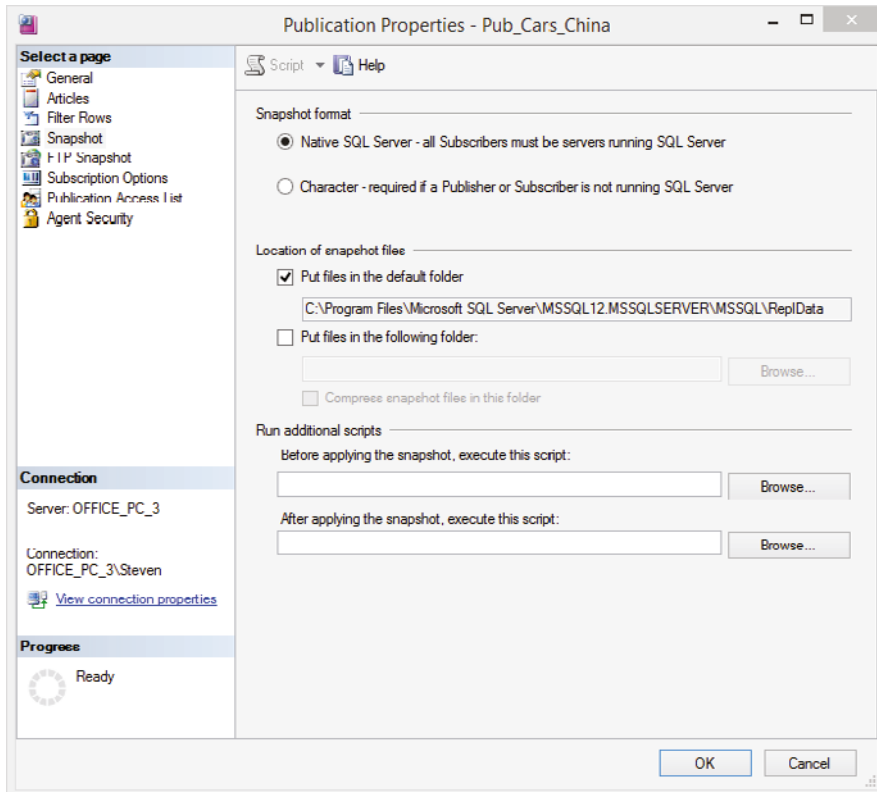


Rysunek 15.21. Ekran z potwierdzeniem operacji

Reszta procesu przebiega podobnie jak w trakcie tworzenia dystrybutora. Gdy klikniesz przycisk *Finish*, utworzona zostanie publikacja ze snapshotem. Publikację i wygenerowane zadania narzędzia SQL Server Agent możesz zobaczyć w oknie *Object Explorer*.

Na razie system nie utworzył żadnych plików ani katalogów w folderze przeznaczonym na snapshot. Także sam snapshot nie został wygenerowany, ponieważ nikt nie zasubskrybował publikacji. Aby system SQL Server wykonał dalsze operacje, trzeba zasubskrybować nową publikację. Gdy dodasz subskrybenta, system SQL Server utworzy wszystkie potrzebne pliki.

Znajdziesz je we współużytkowanym katalogu zdefiniowanym w trakcie konfigurowania dystrybutora. Domyślna lokalizacja tego katalogu to: `C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\ReplData`. Jeśli zdefiniowałeś inną lokalizację, zawsze możesz ją sprawdzić w oknie *Object Explorer* programu SQL Server Management Studio. W tym celu rozwiń węzeł *Replication/Local Publications* i wybierz odpowiednią publikację. Kliknij ją prawym przyciskiem myszy i wybierz opcję *Properties*. Na stronie *Snapshot* (patrz rysunek 15.22) zobaczysz sekcję *Location of snapshot files*, która informuje o lokalizacji potrzebnego katalogu.

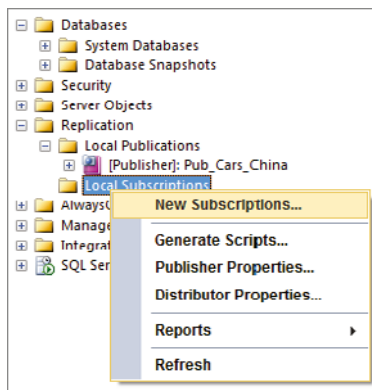


Rysunek 15.22. Sprawdzanie lokalizacji katalogu z plikami publikacji

Konfigurowanie subskrypcji publikacji ze snapshotem

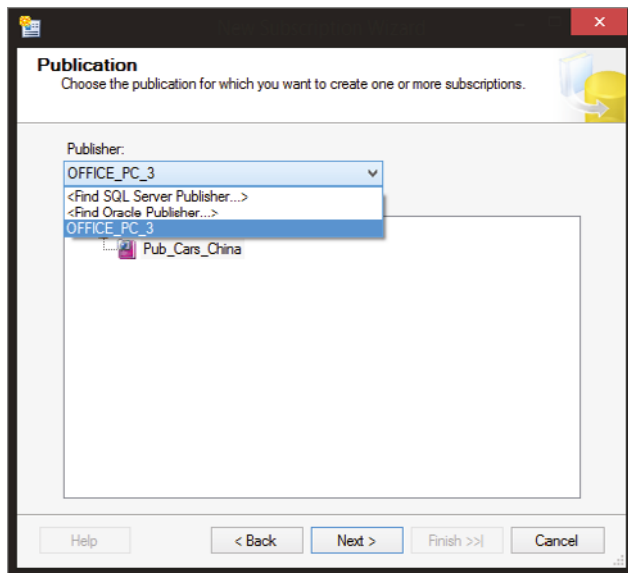
Po utworzeniu publikacji ze snapshotem możesz zasubskrybować ją w innej bazie danych lub na innym serwerze. Wykonaj następujące kroki.

1. Jeśli jeszcze nie nawiązałeś połączenia z serwerem subskrybenta, zrób to w programie SQL Server Management Studio. Następnie rozwiń węzeł *Replication* w oknie *Object Explorer*, kliknij prawym przyciskiem myszy węzeł *Local Subscriptions* i wybierz opcję *New Subscriptions*, co przedstawiono na rysunku 15.23.
2. Pojawi się ekran powitalny kreatora *New Subscription*. Kliknij przycisk *Next*, aby wybrać publikację, którą chcesz zasubskrybować.

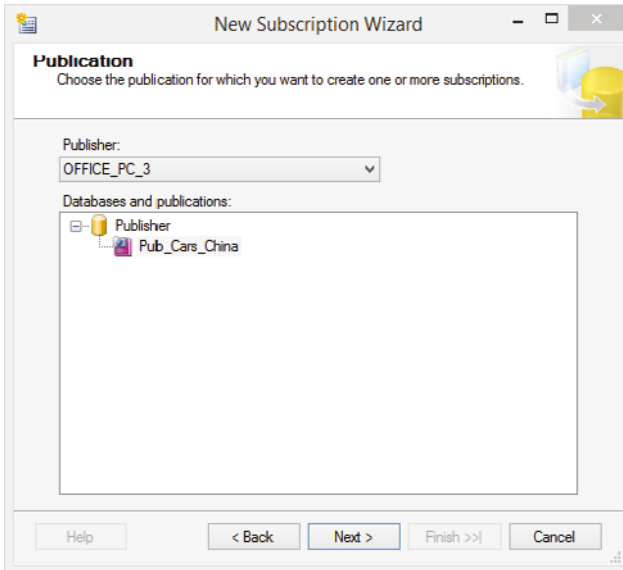


Rysunek 15.23. Tworzenie nowej subskrypcji

W tym przykładzie wydawca i subskrybent działają na tym samym serwerze, dlatego na stronie powinny znajdować się już dostępne publikacje. W praktyce zwykle wydawca i subskrybenci działają na innych serwerach, dlatego trzeba znaleźć serwer wydawcy. W tym celu wybierz opcję *<Find SQL Server Publisher...>* z listy rozwijanej widocznej na rysunku 15.24. Zobaczysz standardowe okno *Connect to Server* znane z programu SQL Server Management Studio. Aby nawiązać połączenie z serwerem, na którym wcześniej skonfigurowałeś publikację ze snapshotem, wybierz serwer wydawcy i kliknij przycisk *Connect*. Zobaczysz ekran *Publication* widoczny na rysunku 15.25. Rozwiń bazę danych *Publisher* i wybierz wcześniej utworzoną publikację *Pub_Cars_China*. Kliknij przycisk *Next*, aby kontynuować.

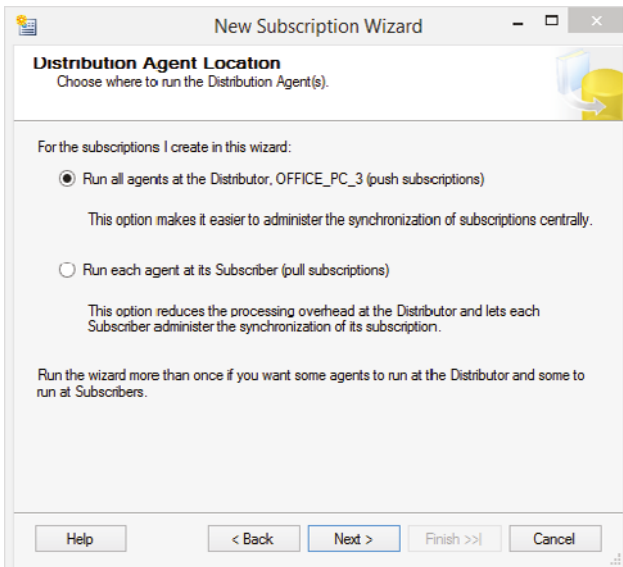


Rysunek 15.24. Wyszukiwanie serwera wydawcy



Rysunek 15.25. Wybieranie publikacji

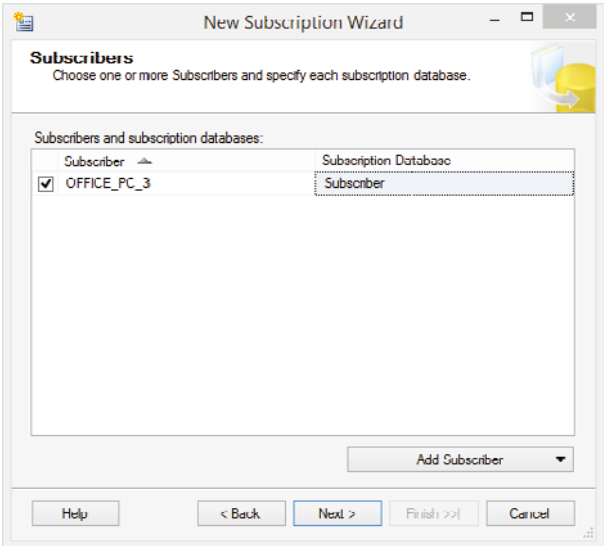
3. Następny ekran nosi nazwę *Distribution Agent Location* i jest przedstawiony na rysunku 15.26. Tu można określić, gdzie mają działać agenci dystrybucji. Można je uruchamiać po stronie dystrybutora, aby zastosować *subskrypcję z wysyłaniem*, albo po stronie subskrybenta, by zastosować *subskrypcję z pobieraniem*. Wybierz pierwszą z dostępnych opcji, *Run all agents at the Distributor*; spowoduje ona włączenie subskrypcji z wysyłaniem. Następnie kliknij przycisk *Next*.



Rysunek 15.26. Wybieranie modelu subskrypcji

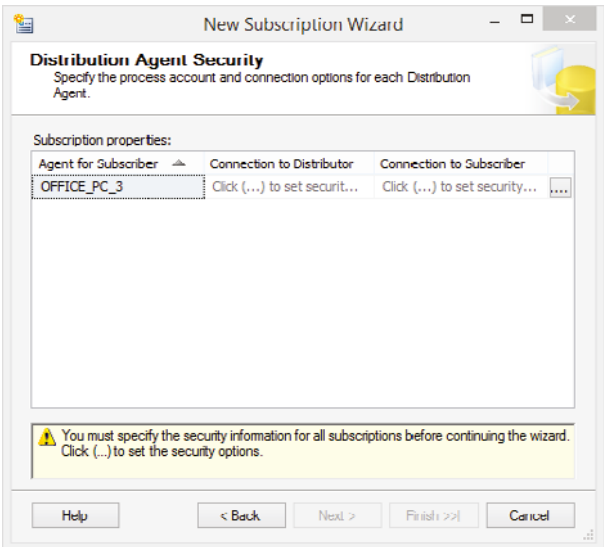
4. Następny ekran to *Subscribers*. Przedstawiono go na rysunku 15.27. Możesz tu ustawić właściwości subskrybenta. Wybierz serwer, na którym ma znajdować się subskrybent.

W tym przykładzie jest to serwer lokalny. Następnie wybierz bazę danych subskrybenta. Jeśli chcesz dodać kolejnych subskrybentów, użyj przycisku *Add Subscriber*. Następnie kliknij przycisk *Next*, aby przejść do kolejnego ekranu.

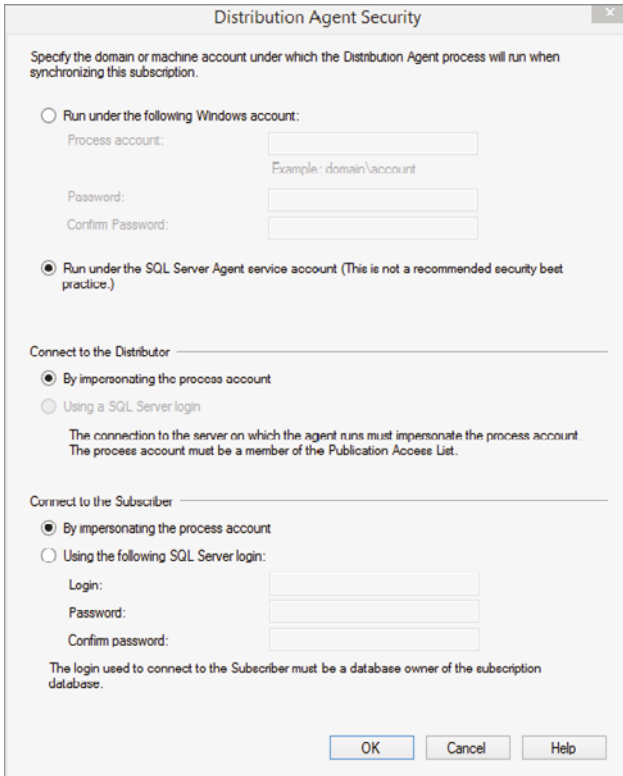


Rysunek 15.27. Ustawianie właściwości subskrybenta

5. Na rysunku 15.28 przedstawiono ekran *Distribution Agent Security*. Widoczna jest tu znana strona do konfigurowania zabezpieczeń agenta. Kliknij wielokropek (...), aby ustawić opcje zabezpieczeń. Jeśli chcesz ustawić inne konto domenowe dla zabezpieczeń agenta Distribution, podaj informacje o nim. W tym przykładzie wybierz opcje widoczne na rysunku 15.29.

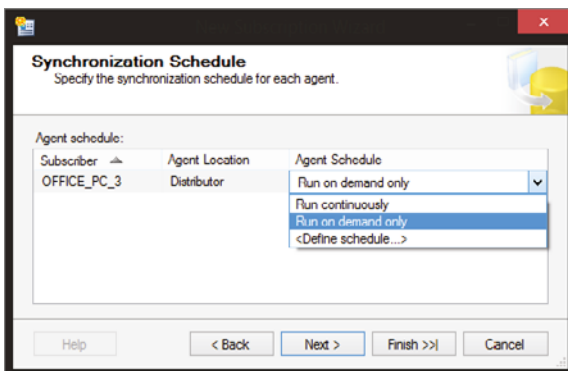


Rysunek 15.28. Ustawianie zabezpieczeń agenta dystrybucji



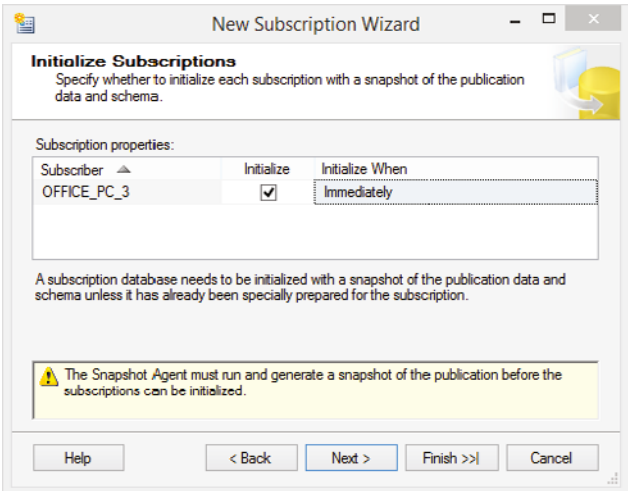
Rysunek 15.29. Właściwości zabezpieczeń

6. Określ harmonogram synchronizacji. Ponieważ to tylko pierwszy przykład testowy, zastosuj prostą synchronizację uruchamianą na żądanie. Z listy rozwijanej wybierz opcję *Run on demand only*, co pokazano na rysunku 15.30. Następnie kliknij przycisk *Next*, aby kontynuować.



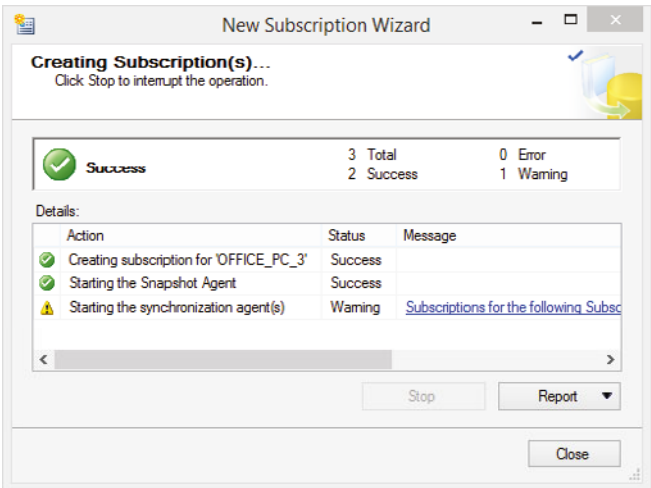
Rysunek 15.30. Synchronizacja będzie uruchamiana na żądanie

7. Na rysunku 15.31 znajduje się ekran *Initialize Subscriptions*. Można na nim określić moment inicjowania snapshota. Zaznacz pole, aby ustawić inicjowanie. Z listy rozwijanej wybierz opcję *Immediately*.



Rysunek 15.31. Konfigurowanie inicjowania snapshota

8. Pojawi się przedstawiony już wcześniej ekran *Wizard Actions* (patrz rysunek 15.20), gdzie można wybrać tworzenie subskrypcji lub generowanie skryptów. Ustaw wybrane opcje i kliknij przycisk *Next*, aby kontynuować.
9. Na ekranie z potwierdzeniem, który się pojawi, możesz się upewnić, że ustawienia są właściwe. Jeśli jest inaczej, cofnij się i popraw je przed wykonaniem wybranych w kreatorze operacji. Gdy będziesz zadowolony z ustawień, kliknij przycisk *Finish*.
10. Na następnym ekranie widać operacje i ich status. Powinieneś zobaczyć trzy operacje, a przy ostatniej z nich pojawi się ostrzeżenie. Gdy klikniesz komunikat dotyczący tej operacji, zobaczysz informację o tym, że snapshot jest niedostępny. Choć ustawiłeś natychmiastowe inicjowanie subskrypcji, musisz ręcznie uruchomić agenta Snapshot, aby utworzył snapshot. Ten ekran jest widoczny na rysunku 15.32. Jeśli wystąpią błędy, zbadaj je i popraw, a następnie ponownie uruchom kreator.



Rysunek 15.32. Ostatni ekran kreatora

Sprawdzanie poprawności replikacji migawkowej

Do tej pory proces konfiguracji był prosty. Jak jednak sprawdzić, czy rozwiązanie działa poprawnie? Oto wybrane testy, które możesz przeprowadzić, aby się upewnić, że wszystko funkcjonuje prawidłowo.

1. Nawiąż połączenie z serwerem wydawcy i sprawdź dostępne na nim rekordy. Wykorzystaj kod w języku SQL (plik *R15_03_Sprawdz_wydawcy.sql*):

```
use publisher;
select * from Sales.Cars;
```

ProdID	ProdDesc	Country	LastUpdate
1	ProEfficient Sedan	US	2014-05-18 09:18:00
2	ProEfficient Van	US	2014-05-18 09:18:00
3	JieNeng Crossover	China	2014-05-18 09:18:00
4	JieNeng Utility	China	2014-05-18 09:18:00
5	EuroEfficient Wagon	Sweden	2014-05-18 09:18:00
6	EuroEfficient Pickup	Sweden	2014-05-18 09:18:00

(6 row(s) affected)

2. Teraz nawiąż połączenie z serwerem subskrybenta. Użyj do tego poniższego kodu w języku SQL (plik *R15_04_Sprawdz_subskrybenta.sql*). Subskrypcja została już zainicjowana, dlatego pojawiają się tylko samochody przeznaczone na rynek chiński.

```
use subscriber;
select * from Sales.Cars;
```

ProdID	ProdDesc	Country	LastUpdate
3	JieNeng Crossover	China	2014-05-07 12:43:00
4	JieNeng Utility	China	2014-05-07 12:43:00

(2 row(s) affected)

3. Teraz założmy, że wprowadzono zmiany w samochodach przeznaczonych na rynek chiński i JieNeng Crossover został zastąpiony na serwerze wydawcy modelem JieNeng Crossover LE. Aby wprowadzić tę modyfikację, wywołaj następujący kod (plik *R15_05_Aktualiz_publikacji.sql*):

```
use publisher;
update Sales.Cars set proddesc = 'JieNeng Crossover LE' where prodid = 3;
```

4. Zaktualizowałeś rekordy po stronie wydawcy, dlatego musisz utworzyć nowy snapshot za pomocą zadania narzędzia SQL Server Agent. Możesz to zrobić za pomocą jednej z poniższych metod.

- W oknie *Object Explorer* rozwiń węzeł *SQL Server Agent/Jobs*. Powinieneś zobaczyć zadanie o nazwie w formacie <nazwa serwera>-Publisher-Pub_Cars_China-1. Kliknij je prawym przyciskiem myszy i wybierz opcję *Start Job* z menu kontekstowego.
- W oknie *Object Explorer* kliknij węzeł *Replication* prawym przyciskiem myszy i wybierz opcję *Launch Replication Monitor*. Korzystanie z monitora replikacji omawiamy w podrozdziale „Monitorowanie replikacji”, dalej w tym rozdziale. Monitor się otworzy, a ponieważ nawiązałeś połączenie z bieżącym serwerem, powinny pojawić się lokalne publikacje. W węźle *My Publications* zobaczysz serwer lokalny; tu jego nazwa to *OFFICE-PC*. Otwórz ten węzeł. Zobaczysz publikację skonfigurowaną w bazie *Publisher*. Jej nazwa to *[Publisher]: Pub_Cars_China*. Kliknij ją prawym przyciskiem myszy i wybierz opcję *Generate Snapshot*. To spowoduje utworzenie nowego snapshota, który jednak nie zostanie na razie zastosowany dla subskrybenta.

5. Ponieważ używana jest subskrypcja z wysyłaniem, przejdź do dystrybutora i uruchom zadanie odświeżające snapshot. Możesz w tym celu uruchomić określone zadanie lub wykorzystać monitor replikacji.
 - Aby uruchomić dane zadanie, otwórz okno *Object Explorer* z obiektami z serwera subskrybenta i rozwiń węzeł *SQL Server Agent/Jobs*. Znajdź zadanie o nazwie w formacie *OFFICE-PC-Publisher-Pub_Cars_Chine-OFFICE-PC-1*, kliknij je prawym przyciskiem myszy i z menu kontekstowego wybierz opcję *Start Job*.
 - W monitorze replikacji wybierz węzeł reprezentujący daną publikację. W prawym panelu otwórz zakładkę *Agent*, kliknij prawym przyciskiem myszy wiersz z zadaniem agenta Snapshot i wybierz opcję *Start Agent*.
6. Teraz sprawdź, czy dane rzeczywiście zostały odświeżone. W tym celu uruchom kod w języku SQL z kroku 2. (plik *R15_04_Sprawdz_subskrybenta.sql*):

```
use subscriber;
```

```
select * from Sales.Cars;
```

ProdID	ProdDesc	Country	LastUpdate
3	JieNeng Crossover LE	China	2014-05-24 12:43:00
4	JieNeng Utility	China	2014-05-24 12:43:00

(2 rows affected)

Implementowanie replikacji transakcyjnej i ze scalaniem

Procedura konfigurowania replikacji transakcyjnej lub ze scalaniem jest podobna do opisanej wcześniej replikacji migawkowej. Szczegółowe instrukcje znajdziesz wcześniej w tym rozdziale. W tym punkcie omawiamy dodatkowe operacje, wyjątki i kilka innych elementów wartych uwagi.

Replikacja transakcyjna

Typowa replikacja transakcyjna nie różni się zbytnio od replikacji migawkowej, jednak obejmuje ważny dodatkowy komponent — jest nim agent Log Reader. Agent śledzi wszystkie zmiany wprowadzone w artykule, dzięki czemu można je przesłać do subskrybenta. W efekcie obciążenie bazy dystrybucyjnej jest większe niż w replikacji migawkowej, dlatego trzeba je obserwować. Należy zwrócić uwagę przede wszystkim na plik dziennika w dystrybucyjnej bazie danych.

Jedną z wersji replikacji transakcyjnej jest odmiana z aktualizowaniem danych przez subskrybenta. W tym podejściu zmiany wprowadzone po stronie subskrybenta są wprowadzane po stronie wydawcy. Aby umożliwić obsługę tego mechanizmu, system SQL Server dodaje do publikowanych tabel kolumnę przeznaczoną na śledzenie zmian. Jest to kolumna *MSrep1_tran_version* pełniąca funkcję unikatowego identyfikatora. W tym modelu kod w poniższej postaci nie zadziała, ponieważ nie podano listy kolumn:

```
insert into Sales.Cars values (9,'English Car','UK', getdate());
```

Dlatego trzeba zmodyfikować kod aplikacji. Aby to zrobić, po nazwie tabeli należy podać w nawiasie listę kolumn odpowiadających poszczególnym wartościom:

```
insert into Sales.Cars (prodid, proddesc, country, lastupdate)
values (9,'English Car','UK', getdate());
```

W replikacji transakcyjnej z aktualizowaniem danych przez subskrybenta używany jest serwer połączony dla baz wydawcy i subskrybenta działających w systemie SQL Server. Serwer połączony

wykorzystuje technologię MS DTC (ang. *Microsoft Distributed Transaction Coordinator*) do koordynowania transakcji. Dlatego komponent MS DTC po stronie wydawcy musi akceptować zdalne połączenia.

Replikacja ze scalaniem

W replikacji ze scalaniem wszystkie artykuły muszą mieć kolumnę ROWGUIDCOL z unikatowym indeksem. Jeśli nie utworzysz takiej kolumny, system SQL Server ją doda. Podobnie jak w replikacji transakcyjnej z aktualizowaniem danych przez subskrybenta, instrukcja INSERT bez listy kolumn nie zadziała poprawnie.

Dla replikacji transakcyjnej i ze scalaniem używane są dodatkowe agenty, takie jak Log Reader i Queue Reader. Agenty te wymagają do działania konta domenowego. Może to być konto dedykowane lub współużytkowane z innymi agentami. To, które rozwiązanie wybierzesz, zależy od obowiązującej w Twojej firmie polityki bezpieczeństwa.

Replikacja w trybie P2P

W replikacji w trybie P2P każda jednostka jest jednocześnie wydawcą i subskrybentem. Ten model dobrze sprawdza się w sytuacjach, gdy aplikacje użytkowników muszą czytać lub modyfikować dane w bazach. Jest to ciekawe rozwiązanie, gdy potrzebne jest równoważenie obciążenia i zapewnienie wysokiej dostępności. Funkcja jest dostępna tylko w edycji Enterprise systemu SQL Server. Oracle nazywa ten rodzaj replikacji *multimaster*, natomiast w bazach DB2 jest to replikacja z *aktualizacją w dowolnym miejscu* (ang. *update anywhere*).

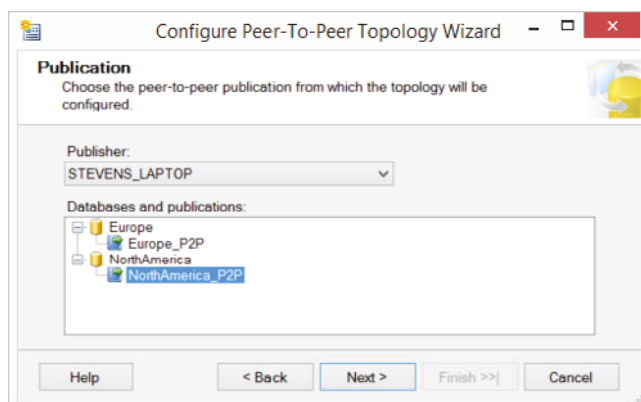
W trakcie analizowania i konfigurowania replikacji w trybie P2P uwzględnij następujące kwestie.

- Technika ta jest zaprojektowana pod kątem niewielkiej liczby baz danych. Zgodnie z ogólną regułą nie powinno ich być więcej niż dziesięć. Jeśli używasz większej liczby baz, mogą wystąpić problemy z wydajnością.
- Replikacja w trybie P2P obsługuje rozwiązywanie konfliktów, ale w inny sposób niż replikacja ze scalaniem. Od wersji SQL Server 2008 replikacja w trybie P2P po wykryciu konfliktu wywołuje błąd krytyczny. W efekcie działanie agenta Distribution kończy się niepowodzeniem. Dlatego powinieneś tak zaprojektować aplikację, aby uniknąć konfliktów.
- Replikacja w trybie P2P nie obsługuje filtrowania danych. Filtrowanie byłoby niezgodne z przeznaczeniem tego podejścia, ponieważ wszystkie jednostki mają być równorzędnymi partnerami zapewniającymi wysoką dostępność i równoważenie obciążenia.
- Aplikacje mogą rozdzielać operacje odczytu między wiele baz danych, a bazy zawsze są dostępne. Poszczególne węzły można dodawać lub (na czas konserwacji) usuwać.
- Replikacja w trybie P2P jest dostępna tylko w edycji Enterprise. Jednak podczas testów można z niej korzystać także w edycji Developer systemu SQL Server 2014.

Konfigurowanie replikacji w trybie P2P

Proces konfigurowania replikacji w trybie P2P przebiega podobnie jak konfigurowanie replikacji migawkowej, transakcyjnej lub ze scalaniem. Występują jednak pewne różnice. Zapoznasz się z nimi w następnych punktach w ramach przykładu ilustrującego konfigurację takiej replikacji.

1. Zaczynij od jednej bazy danych, utwórz jej kopię zapasową, a następnie odtwórz ją we wszystkich innych używanych jednostkach. Dzięki temu zaczniesz od spójnego punktu wyjścia.
2. Wszystkie węzły w tej architekturze wymagają dystrybutora. Dodaj go. Choć możesz wykorzystać jedną bazę dystrybucyjną na wszystkie publikacje, w systemach produkcyjnych nie jest to zalecane rozwiązanie. W omawianych tu przykładach na każdym serwerze wydawcy działa lokalna dystrybucyjna baza danych.
3. Po skonfigurowaniu dystrybutora utwórz publikację, tak jak w zwykłej replikacji transakcyjnej, jednak tym razem możesz wybrać nowy typ publikacji: *Peer-to-Peer*. Wybierz tę opcję w polu *Publication Type* na ekranie o tej samej nazwie (przedstawionym wcześniej na rysunku 15.14).
4. Po ustawieniu publikacji typu P2P przejdź do końca kreatora nowej publikacji. Proces ten przebiega podobnie jak przy tworzeniu opisanej wcześniej publikacji dla replikacji migawkowej.
5. Po utworzeniu publikacji typu P2P musisz opracować dodatkowe publikacje tego rodzaju dla innych równorzędnych jednostek z budowanej architektury. W tym przykładzie utwórz dwie odrębne publikacje typu P2P, o nazwach *NorthAmerica_P2P* i *Europe_P2P*. Możesz wykorzystać przykładowy kod z opisu replikacji migawkowej, aby utworzyć jedną tabelę replikowaną we wszystkich systemach.
6. W dalszej części procesu konfiguracji kliknij publikację prawym przyciskiem myszy i wybierz opcję *Configure Peer-to-Peer Topology*. Pojawi się okno dialogowe widoczne na rysunku 15.33. Wspomniana opcja jest dostępna tylko dla publikacji typu P2P. Kliknij przycisk *Next*, aby kontynuować.



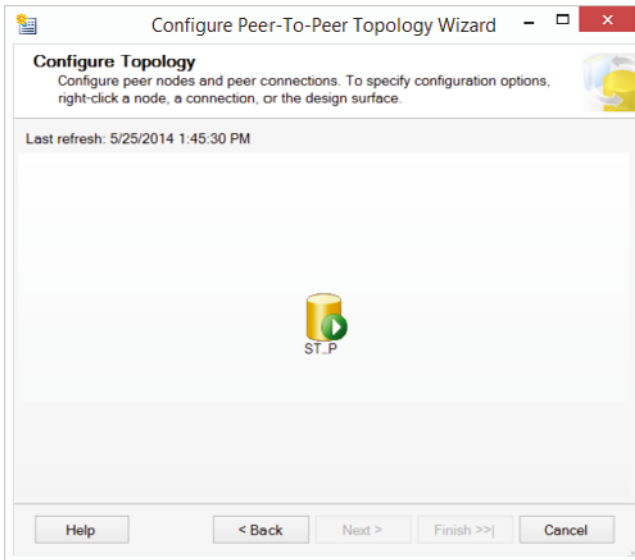
Rysunek 15.33. Początek procesu konfigurowania architektury P2P

Konfigurowanie architektury dla replikacji P2P

Po skonfigurowaniu replikacji w trybie P2P pojawi się ekran powitalny. Od tego miejsca możesz zacząć konfigurować architekturę P2P. Wykonaj następujące kroki.

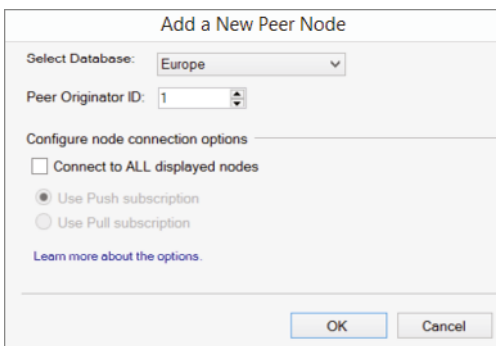
1. Wybierz publikację typu P2P używaną przy konfigurowaniu architektury. W tym przykładzie użyj publikacji lokalnej *NorthAmerica_P2P*.

- Po wybraniu publikacji NorthAmerica_P2P kliknij przycisk *Next*. Pojawi się strona *Configure Topology*, wyświetlająca obszar projektowy wyglądający tak, jak na rysunku 15.34. W celu dodania węzłów do architektury kliknij obszar projektowy prawym przyciskiem myszy i wybierz opcję *Add a New Peer Node*.



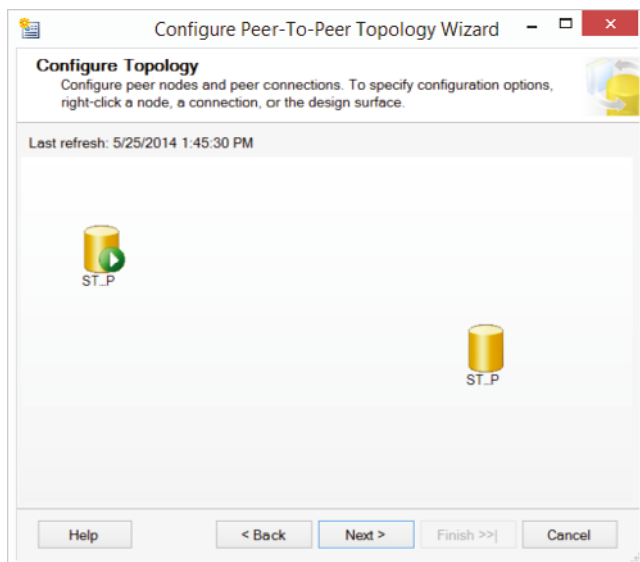
Rysunek 15.34. Obszar do projektowania architektury dla replikacji P2P

- Pojawi się standardowe okno dialogowe do łączenia z serwerami, znane z programu SQL Server Management Studio. Można tu określić, na jakim serwerze działa dana jednostka. W tym przykładzie wszystkie komponenty działają na tym samym serwerze, dlatego wybierz kropkę (.), aby wskazać serwer lokalny.
- Następnie zobaczysz okno dialogowe *Add a New Peer Node*, widoczne na rysunku 15.35. Wybierz bazę Europe, co pokazano na rysunku.



Rysunek 15.35. Dodawanie nowego węzła

- Kliknij przycisk *OK*. Węzeł zostanie dodany do obszaru projektowego, co widać na rysunku 15.36. W tym obszarze trudno określić, który węzeł jest który. Na szczęście po najechaniu kursorem nad każdy węzeł zobaczysz rozbudowane informacje.



Rysunek 15.36. Węzeł dodany do obszaru projektowego

6. Następny krok polega na połączeniu węzłów w architekturze. Aby połączyć dwa węzły, kliknij prawym przyciskiem myszy jeden z nich w obszarze projektowym i wybierz opcję *Connect to ALL displayed nodes* lub *Add a new peer connection*. W tym przykładzie wybierz *Connect to ALL displayed nodes*, ponieważ wyświetlone są tylko dwa węzły. To spowoduje dodanie dwukierunkowej strzałki między węzłami.
Ponieważ w tym przykładzie występują tylko dwa węzły, mogłeś uzyskać ten sam efekt szybciej, wybierając opcję *Connect to ALL displayed nodes* w oknie z rysunku 15.35.
7. Kliknij przycisk *Next*. Kreator przejdzie do strony *Log Reader Agent Security*. Ustaw informacje o zabezpieczeniach. W tym celu kliknij przycisk wielokropka (...) i wybierz opcję *Run under the SQL Server Agent service account*. Kliknij przycisk *Next*. Pojawi się strona *Distribution Agent Security*.
8. Na stronie *Distribution Agent Security* trzeba podać informacje o zabezpieczeniach dotyczące agentów *Distribution* z wszystkich węzłów z architektury. W tym przykładzie dla obu serwerów wybierz te same opcje: *Run under the SQL Server Agent service account* i *By impersonating the process account*, aby określić sposób łączenia się z dystrybutorem i subskrybentem (patrz rysunek 15.37). Potem kliknij przycisk *Next*, aby przejść do następnej strony.
9. Następna strona kreatora to *New Peer Initialization*. Można tu określić sposób inicjowania nowych jednostek. Tu bazy utworzono ręcznie i umieszczono w nich te same dane, dlatego możesz wybrać pierwszą opcję. Kliknij przycisk *Next*, aby przejść do następnej strony.
10. Pojawi się strona *Complete the Wizard*. Tu możesz przejrzeć wszystkie wybrane ustawienia. Jeśli są prawidłowe, kliknij przycisk *Finish*, aby wykonać skonfigurowane operacje.
11. Ostatnia strona kreatora wyświetla wykonywane operacje i informuje o ostrzeżeniach lub błędach. Jeśli któraś z operacji nie zakończyła się sukcesem, zbadaj i rozwiąż wszystkie problemy, a następnie ponownie uruchom kreator architektury.

Distribution Agent Security

Specify the domain or machine account under which the Distribution Agent process will run when synchronizing this subscription.

☐ Run under the following Windows account:

Process account: Example: domain/account

Password:

Confirm Password:

☒ Run under the SQL Server Agent service account (This is not a recommended security best practice.)

Connect to the Distributor

☒ By impersonating the process account

☐ Using a SQL Server login

The connection to the server on which the agent runs must impersonate the process account. The process account must be a member of the Publication Access List.

Connect to the Subscriber

☒ By impersonating the process account

☐ Using the following SQL Server login:

Login:

Password:

Confirm password:

The login used to connect to the Subscriber must be a database owner of the subscription database.

OK Cancel Help

Rysunek 15.37. Konfigurowanie agentów Distribution

W ten sposób ukończyłeś budowanie architektury P2P. Możesz w dowolnym momencie ją zmodyfikować. W tym celu ponownie uruchom kreator *Configure Topology Wizard* i dodaj nowe węzły, usuń istniejące lub zmień połączenia między nimi.

Replikacja obsługiwana za pomocą skryptów

Dla wielu użytkowników preferowanym narzędziem do zarządzania replikacją jest interfejs graficzny w postaci różnych kreatorów z programu SQL Server Management Studio. Jednak administrator bazy danych może też utworzyć skrypt zapisywany w systemie kontroli kodu źródłowego (takim jak Visual Source Safe lub Visual Studio Team Foundation Server).

Istnieją dwa sposoby przeprowadzania replikacji za pomocą skryptów. Można zastosować różne kreatory replikacji (umożliwiają one wygenerowanie skryptów) lub wykorzystać program SQL Server Management Studio i przygotować skrypty dla poszczególnych obiektów.

Skrypty możesz zapisać w systemie kontroli kodu źródłowego. Pozwala to na zarządzanie wersjami skryptów i stosowanie konfiguracji replikacji w systemach rozwojowych, testowych, kontroli jakości, przedprodukcyjnych i produkcyjnych.

Monitorowanie replikacji

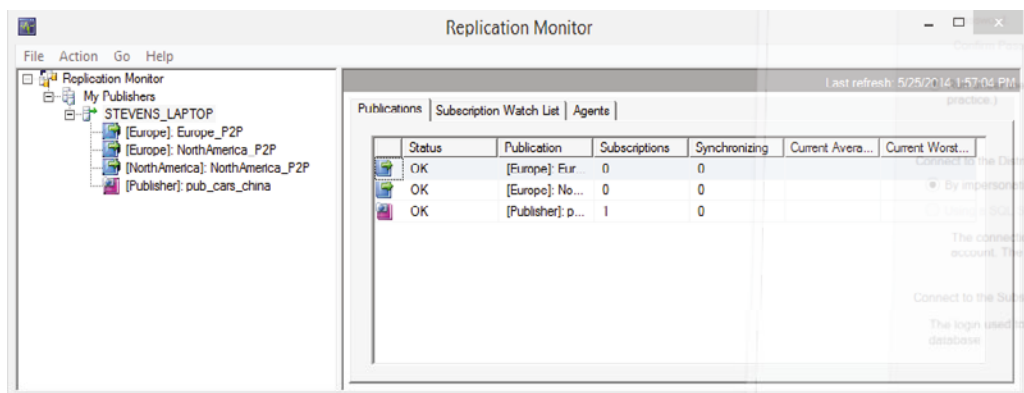
Do monitorowania replikacji służą trzy narzędzia: monitor replikacji, monitor wydajności i powiązane z replikacją widoki DMV. W tym podrozdziale szczegółowo omawiamy wszystkie te rozwiązania.

Monitor replikacji

Monitor replikacji (narzędzie Replication Monitor) jest wbudowany w program SQL Server Management Studio. Wyświetla stan replikacji i umożliwia wprowadzanie zmian w konfiguracji. Ponadto informuje o opóźnieniu i liczbie zaległych poleceń. Nie wyświetla wskaźników dotyczących transakcji i poleceń ani historii operacji z okresu przed wczytaniem bieżącego snapshota z danymi.

Monitor replikacji możesz uruchomić w programie SQL Server Management Studio.

1. Otwórz menu kontekstowe węzła *replication* lub jednego z serwerów z węzłów *Local Publications* albo *Local Subscriptions*.
2. Następnie wybierz opcję *Launch Replication Monitor*. Na rysunku 15.38 przedstawiono monitor replikacji działający na serwerze ze skonfigurowaną replikacją lokalną. Jeśli uruchomisz monitor replikacji na komputerze bez replikacji lokalnej, będziesz musiał wskazać serwery używane w architekturze replikacji, aby zobaczyć potrzebne informacje.

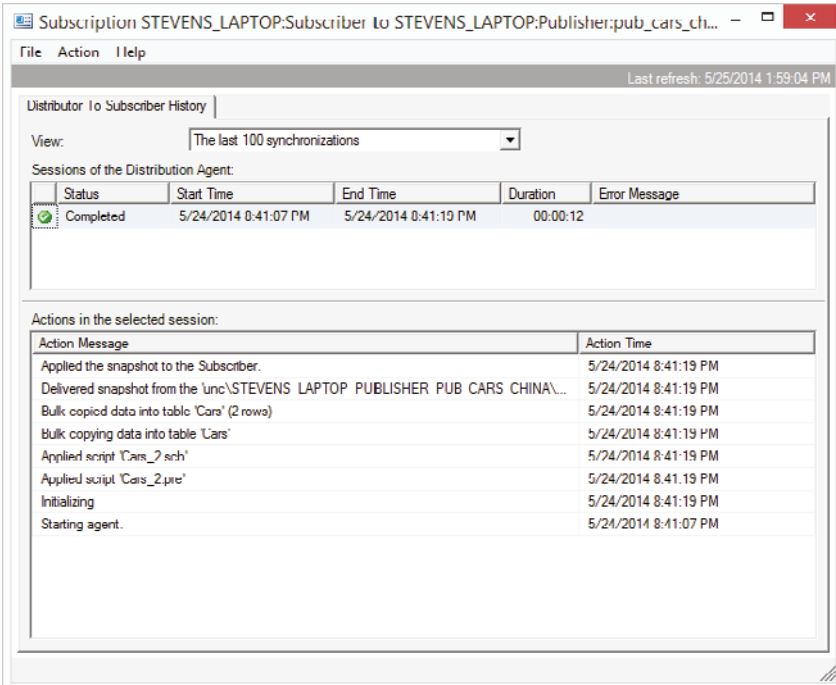


Rysunek 15.38. Monitor replikacji

Oto kilka najważniejszych operacji, które można wykonać po uruchomieniu monitora replikacji.

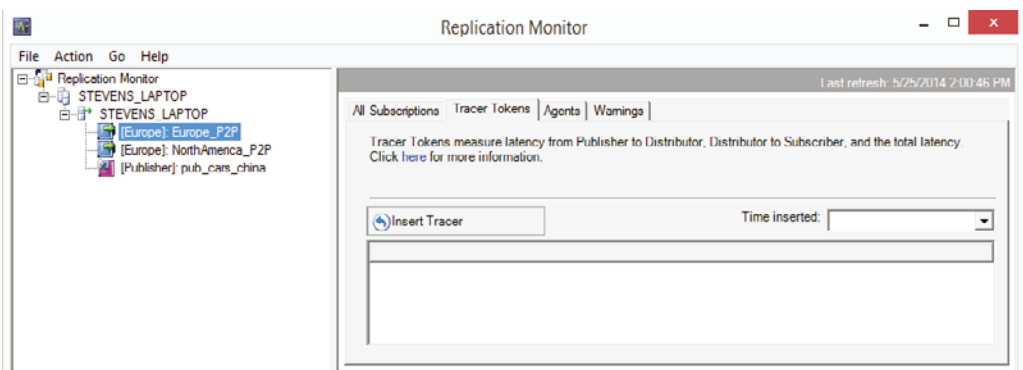
- Gdy wybierzesz główny węzeł w monitorze wydajności, możesz uruchomić widok dystrybutora z systemu SQL Server 2014. Na prawym panelu pojawią się opcje umożliwiające dodanie wydawcy (*Add Publisher*) lub przełączenie się do widoku dystrybutora (*Switch to Distributor View*). Jeśli wybierzesz drugą opcję, zobaczysz węzły powiązane z poszczególnymi dystrybutorami. Tu używany jest dystrybutor lokalny, dlatego widok się nie zmienia. W bardziej skomplikowanej konfiguracji, gdy używanych jest wiele strumieni replikacji i wielu dystrybutorów, jako jednostka nadrzędna dla publikacji i subskrypcji zostanie wybrany odpowiedni dystrybutor. Aby przełączyć się do poprzedniego widoku, ponownie zaznacz główny węzeł i wybierz opcję *Switch to Publisher Group View*.

- Aby sprawdzić stan subskrypcji (patrz rysunek 15.39), kliknij dwukrotnie dowolną pozycję w zakładce *All Subscriptions*. Pojawi się wtedy okno, w którym można wygodnie podejrzeć raporty i stan publikacji przeznaczonych do dystrybucji, dystrybucji dostępnych do subskrypcji i niewykonanych instrukcji. Jest to przyjazne dla użytkownika narzędzie, które zapewnia dobry przegląd wszystkich informacji o stanie replikacji.



Rysunek 15.39. Informacje o stanie subskrypcji

- Tokeny śledzące (ang. *tracer token*) umożliwiają pomiar wydajności replikacji. Zakładka *Tracer Tokens* jest przedstawiona na rysunku 15.40. Tokeny śledzące to fikcyjne rekordy, które monitor replikacji wykorzystuje do pomiaru wydajności danego modelu replikacji. Za ich pomocą możesz oszacować opóźnienie w transferze danych między wydawcą, dystrybutorem i subskrybentem.



Rysunek 15.40. Zakładka Tracer Tokens w monitorze replikacji

Monitor wydajności

W monitorze wydajności można śledzić kilka obiektów i liczników związanych z replikacją w systemie SQL Server. Oto one.

- **SQLServer: Replication Agent.** Obiekt obejmuje jeden licznik, *Running*, wyświetlający liczbę agentów replikacji działających na danym serwerze.
- **SQLServer: Replication Log Reader.** Wyświetla statystyki agenta Log Reader. Dostępne liczniki określają liczbę poleceń na sekundę, liczbę transakcji na sekundę i opóźnienie.
- **SQLServer: Replication Dist.** Wyświetla statystyki dotyczące dystrybutora. Dostępne są tu takie same liczniki jak dla obiektu *Replication Log Reader*.
- **SQLServer: Replication Merge.** Wyświetla statystyki dotyczące agenta Merge. Dostępne liczniki określają liczbę konfliktów na sekundę, liczbę pobranych zmian na sekundę i liczbę aktualizacji na sekundę.
- **SQLServer: Replication Snapshot.** Wyświetla statystyki dotyczące agenta Snapshot. Dostępne liczniki określają liczbę dostarczonych poleceń na sekundę i liczbę dostarczonych transakcji na sekundę.

Gdy zastanawiasz się, które liczniki wydajności wykorzystać, uwzględnij przede wszystkim typ replikacji. Dla replikacji migawkowej ważne są liczniki z grupy *Replication Snapshot*. Przy replikacji ze scalaniem uwzględnij liczniki z grupy *Replication Merge*. Jeśli stosujesz replikację transakcyjną, ważne będą liczniki z grup *Replication Log Reader* i *Replication Dist*.

Oto dodatkowe informacje na temat liczników, które warto monitorować dla replikacji transakcyjnej.

- **Obiekt: SQLServer: Replication Logreader, liczniki: Logreader: Delivered Cmds/Sec i Trans/Sec.** Te dwa liczniki są powielane dla każdej publikacji. Wyświetlają one liczbę poleceń i transakcji wczytanych przez agenta Log Reader na sekundę. Jeśli wartość tych liczników wzrasta, oznacza to wyższą liczbę transakcji po stronie wydawcy.
- **Obiekt: SQLServer: Replication Dist., liczniki: Dist: Delivered Cmds/Sec i Trans/Sec.** Te dwa liczniki są powielane dla każdej subskrypcji. Określają one, ile poleceń i transakcji jest dostarczanych na sekundę do subskrybenta. Jeśli te liczby są niższe niż wartości z liczników *Log Reader*, oznacza to, że polecenia kumulują się w dystrybucyjnej bazie danych. Wartości wyższe niż z liczników *Log Reader* i zbiór poleceń do wykonania mogą wskazywać na to, że proces replikacji nadrabia zaległości.

Widoki DMV związane z replikacją

Oto cztery związane z replikacją widoki DMV dostępne w każdej bazie systemu SQL Server.

- Widok `sys.dm_repl_articles` zawiera informacje o każdym opublikowanym artykule. Zwraca dane z opublikowanej bazy. Dla każdego obiektu opublikowanego w poszczególnych artykułach tworzony jest jeden wiersz. Oto składnia pozwalająca pobrać dane z tego widoku:

```
select
* from sys.dm_repl_articles
```

- Widok `sys.dm_repl_schemas` zawiera informacje o każdej opublikowanej tabeli i kolumnie. Zwraca dane z opublikowanej bazy. Dla każdej kolumny z każdego opublikowanego obiektu dostępny jest jeden wiersz. Oto składnia pozwalająca pobrać dane z tego widoku:

```
select
* from sys.dm_repl_schemas
```

- Widok `sys.dm_repl_tranhash` zawiera informacje o haszowaniu transakcji. Oto składnia pozwalająca pobrać dane z tego widoku:

```
select
* from sys.dm_repl_tranhash
```

- Widok `sys.dm_repl_traninfo` zawiera informacje o każdej transakcji dla replikacji transakcyjnej. Oto składnia pozwalająca pobrać dane z tego widoku:

```
select
* from sys.dm_repl_traninfo
```

Te widoki DMV wyświetlają informacje o tym, co jest opublikowane w poszczególnych bazach danych. Nie pomagają jednak w monitorowaniu przebiegu replikacji. Do tego służy systemowa procedura składowana `sp_replcounters` (opisujemy ją w następnym punkcie). Można też zapoznać się z zawartością dystrybucyjnej bazy danych. Znacznie łatwiej wywołać procedurę składowaną `sp_replcounters`, niż badać wewnętrzne mechanizmy replikacji w bazie MSDB lub w bazie dystrybucyjnej. Jednak czasem potrzebne są informacje niedostępne za pomocą tej procedury. Jeśli ich potrzebujesz, musisz poszukać ich w dystrybucyjnej bazie danych. Na razie jest to nieudokumentowany obszar. Dobrym punktem wyjścia jest kod źródłowy procedur składowanych związanych z replikacją. Wykorzystaj go do ustalenia, gdzie znajdują się potrzebne dane. Następnie je pobierz.

Procedura `sp_replcounters`

Systemowa procedura składowana `sp_replcounters` jest związana z replikacją i zwraca informacje o częstotliwości i opóźnieniu transakcji oraz o pierwszym i ostatnim numerze porządkowym z dziennika każdej publikacji z serwera. Procedurę należy wywołać na serwerze wydawcy. Jeśli uruchomisz ją na serwerze dystrybutora lub subskrybenta, nie otrzymasz żadnych informacji.

W tabeli 15.1 przedstawione są przykładowe dane wyjściowe z tej procedury. Wyniki rozdzielono na dwa wiersze, aby zwiększyć ich czytelność.

Tabela 15.1. Przykładowe dane wyjściowe z procedury `sp_replcounters`

Baza danych	Zreplikowane transakcje	Liczba transakcji na sekundę	Opóźnienie replikacji (w sekundach)
Publisher	178	129.0323	0.016
Pierwszy numer z dziennika	Ostatni numer z dziennika		
0x0002FACC003458780001	0x0002FACC003459A1003D		

Podsumowanie

Replikacja to ważna technologia przenoszenia danych między serwerami z systemem SQL Server. Istnieją różne rodzaje replikacji: migawkowa, transakcyjna i ze scalaniem. W ramach replikacji można też stosować różne architektury: z jednym wydawcą, z wieloma wydawcami, z subskrybentami aktualizującymi dane i P2P. Do monitorowania replikacji służą: Replication Monitor, monitor wydajności i odpowiednie liczniki z tego monitora. Ponadto widoki DMV i procedury składowane związane z replikacją pomagają w identyfikowaniu źródeł problemów.

Obok technologii AlwaysOn, przesyłania dzienników, kopii lustrzanych baz danych i klastrowania SQL Server 2014 udostępnia też wiele innych funkcji umożliwiających klientom równoważenie obciążenia, zapewnianie wysokiej dostępności, przywracanie stanu po awariach i skalowanie.

W rozdziale 16. poznasz klastry z systemem SQL Server 2014.

Klastrowanie w systemie SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Zapoznanie się z klastrowaniem i szczegółami działania tego mechanizmu.
- Analiza konfiguracji sprzętu i oprogramowania w celu zidentyfikowania najlepszego rozwiązania dla firmy.
- Konfigurowanie sprzętu.
- Konfigurowanie i klastrowanie systemu operacyjnego.
- Konfigurowanie systemu SQL Server w klastrze.
- Testowanie rozwiązania.
- Zarządzanie produkcyjnym klastrem z systemem SQL Server i rozwiązywanie problemów.

Korzystając z solidnej wiedzy z tego rozdziału, będziesz mógł z powodzeniem zainstalować i skonfigurować egzemplarz systemu SQL Server 2014 w klastrze oraz zarządzać takim egzemplarzem. Dowiesz się, jak działa klaster FCI (ang. *Failover Cluster Instance*), i ustalisz, czy jest on odpowiedni dla Twojej firmy. W wersji SQL Server 2012 nazwa tego mechanizmu to AlwaysOn Failover Cluster Instance; to ta sama technologia, co WSFC (ang. *Windows Server Failover Clustering*). Zobaczysz, kiedy lepiej zaktualizować starszą wersję klastra WSFC, a kiedy należy utworzyć nowy klaster. Dowiesz się też, jak zaplanować klastrowanie z uwzględnieniem sprzętu i systemu operacyjnego, a także jak zainstalować system SQL Server 2014 w klastrze WSFC. Nauczysz się też konserwować egzemplarz systemu SQL Server w klastrze i nim zarządzać.

Choć w systemie Windows 2012 R2 można tworzyć klastry obejmujące jednostki z różnych lokalizacji, w tym rozdziale koncentrujemy się głównie na klastrach działających w jednym miejscu. Omawiamy też najczęściej stosowane konfiguracje klastrów WSFC.

UWAGA Jeśli nie korzystasz z sieci SAN (ang. *Storage Area Network*) ani sprzętu pozwalającego wykonać opisywane tu przykłady na fizycznych urządzeniach, rozważ wykorzystanie maszyny wirtualnej Hyper-V i serwera plików w postaci systemu Windows Server 2012 skonfigurowanego jak udział plikowy SMB (ang. *Server Message Block*), aby pełnił funkcję współużytkowanych dysków. Więcej informacji znajdziesz w artykule „Install SQL Server with SMB Fileshare as a Storage Option” na stronie <http://msdn.microsoft.com/en-us/library/hh759341.aspx>.

Klastrowanie a sytuacja firmy

Wielu administratorów baz danych ma problemy z dobrym zrozumieniem klastrowania. Dalej znajdziesz przydatną roboczą definicję tego mechanizmu.

Microsoft Windows Failover Clustering Instance (FCI) to mechanizm zapewniający wysoką dostępność, zaprojektowany w celu zwiększenia czasu nieprzerwanej pracy egzemplarzy systemu SQL Server. Typowy *klast* obejmuje przynajmniej dwa fizyczne serwery nazywane *węzłami*. Zaleca się, aby serwery były identycznie skonfigurowane. Jeden serwer to *węzeł aktywny*. Egzemplarz systemu SQL Server z tego węzła wykonuje operacje produkcyjne. Drugi serwer to *węzeł pasywny*. System SQL Server jest na nim zainstalowany, ale nie pracuje. Jeśli egzemplarz z węzła aktywnego przestanie działać, węzeł pasywny przejmie jego funkcje i zacznie przetwarzać operacje produkcyjne. Przerwa potrzebna na przełączenie serwerów jest bardzo krótka. Ponadto w klastrze WSFC oba węzły mogą być aktywne. Wtedy po awarii dowolnego egzemplarza systemu SQL Server drugi może przejąć jego zadania. Często między węzłami działa współużytkowany magazyn danych. Pamiętaj, że klast z egzemplarzami systemu SQL Server przechowuje tylko jeden zbiór danych, dlatego nie chroni przed uszkodzeniem baz.

Jest to prosty opis, jednak może z niego wynikać wiele niejasnych założeń, dlatego z klastrowaniem wiąże się wiele nieporozumień. Jednym z najlepszych sposobów na dobre zrozumienie możliwości, jakie daje klastrowanie, jest przyjrzenie się szczegółom.

Co można uzyskać przy użyciu klastów?

Klastrowanie ma poprawiać dostępność sprzętu fizycznych serwerów, systemu operacyjnego i egzemplarzy systemu SQL Server, jednak nie dotyczy współużytkowanego magazynu danych. Jeśli któryś z komponentów zawiedzie, nastąpi przełączenie z niesprawnego egzemplarza systemu SQL Server na inny węzeł, który automatycznie przejmie wykonywanie zadań, aby maksymalnie skrócić przestój. Na zapleczu pasywny węzeł przejmuje wtedy kontrolę nad współużytkowanym magazynem danych. Węzeł pasywny uruchamia lokalne usługi systemu SQL Server i nawiązuje połączenie ze wspomnianym magazynem, w którym znajdują się wszystkie bazy danych (użytkowników i systemowe).

Zastosowanie klastów FCI pomaga też skrócić przestój przy konserwacji węzłów. Jeśli np. chcesz unowocześnić sprzęt fizycznego serwera lub zainstalować nowy pakiet poprawek w systemie operacyjnym, możesz to robić po kolei w poszczególnych węzłach. W tym celu wykonaj następujące kroki.

1. Zaktualizuj pasywny węzeł, na którym egzemplarz systemu SQL Server nie jest aktualnie uruchomiony.

2. Ręcznie przełącz aktywny węzeł na już zaktualizowany węzeł, który staje się aktywny.
3. Zaktualizuj obecnie pasywny węzeł.
4. Po aktualizacji, jeśli zechcesz, możesz przełączyć węzły. Ten mechanizm klastrów pomaga zmniejszyć łączny czas przestoju spowodowanych aktualizacją.

W trakcie aktualizacji musisz uważać, aby nie przełączyć klastra ręcznie na węzeł, który jeszcze nie został zaktualizowany, ponieważ system z niezaktualizowanymi plikami binarnymi nie pracuje stabilnie.

UWAGA Starsze klastry WSFC należy zastąpić klastrami WSFC z systemu Windows 2012 R2. Utwórz nowy klaster i przenieś do niego używane bazy danych.

Czego klastrowanie nie umożliwia?

Klastrowanie to tylko jeden z wielu ważnych i potrzebnych mechanizmów zapewniania wysokiej dostępności. Inne aspekty (np. dbanie o rezerwę wydajności wszystkich komponentów sprzętowych) są równie ważne. Bez rezerwy wydajności najbardziej zaawansowane klastry mogą nie spełniać swojej funkcji. Jeśli zaniedbasz niektóre aspekty, wysokie wydatki na klastrowanie nie będą dobrą inwestycją. Szczegółowe omówienie tej kwestii znajdziesz w podrozdziale „Przygotowanie do klastrowania”, dalej w tym rozdziale.

Niektórzy administratorzy baz danych uważają, że klastrowanie pozwala całkowicie wyeliminować przestoje. To nieprawda. Klastrowanie umożliwia skrócenie przestoju, ale ich nie eliminuje. Samo przełączanie węzłów powoduje przestój trwający od kilku sekund do kilku minut. Trzeba zatrzymać usługi systemu SQL Server w jednym węźle, a następnie uruchomić je w innym i przywrócić bazę danych.

Klastrowanie nie jest też mechanizmem ochrony danych, ponieważ w klastrze stosowany jest tylko jeden współużytkowany magazyn danych. Dla wielu administratorów baz danych jest to zaskakująca informacja. Dane trzeba więc chronić innymi sposobami — przez tworzenie kopii zapasowych, przysyłanie dzienników, tworzenie grup dostępności AlwaysOn lub kopii lustrzanych dysków. Wszystkie serwery w klastrze współużytkują te same dyski z bazą danych (choć nie korzystają z nich w tym samym momencie), dlatego jej uszkodzenie jest odczuwalne na wszystkich serwerach.

Klastrowanie nie jest też przydatne przy równoważeniu obciążenia. *Równoważenie obciążenia* ma miejsce wtedy, gdy wiele serwerów działa jak jeden, a obciążenie jest rozkładane między jednocześnie pracujące maszyny. Niektórzy administratorzy baz danych mogą sądzić, że klastrowanie zapewnia równoważenie obciążenia między węzłami. Tak się nie dzieje. Klastrowanie pomaga jedynie wydłużyć czas nieprzerwanej pracy egzemplarzy systemu SQL Server. Jeśli potrzebujesz równoważenia obciążenia, poszukaj innego rozwiązania. Może to być np. replikacja transakcyjna w trybie P2P, opisana w rozdziale 15. „Replikacja”.

Klastrowanie wymaga wersji Standard lub Datacenter systemu operacyjnego Windows Server 2012 R2 i edycji Standard, Enterprise lub Business Intelligence systemu SQL Server. Klastry zwykle działają w ramach centrum danych, jednak można je też tworzyć w odległych od siebie jednostkach (powstają wtedy tzw. *geoklastry*). Gdy chcesz utworzyć geoklaster, skonsultuj się z dostawcą magazynów pamięci, aby udostępnić je w oddalonych od siebie lokalizacjach i umożliwić synchronizację macierzy dyskowych. System SQL Server 2014 obsługuje też inny mechanizm — klastry oparte na podsięciach, obejmujące różne lokalizacje. Ograniczenia związane z podsięciami wyeliminowano w wersji SQL Server 2012.

UWAGA Klastrowanie wymaga zatrudnienia doświadczonych administratorów baz danych, dobrze przeszkolonych z zakresu sprzętu i oprogramowania. Specjaliści znający się na wysoce dostępnych systemach żądają zwykle wysokiego wynagrodzenia.

Choć system SQL Server obsługuje klastry, nie wszystkie aplikacje klienckie korzystające z tego systemu też mogą obsługiwać klastry. Przykładowo jeśli nawet przełączanie egzemplarzy systemu SQL Server przebiega stosunkowo płynnie, aplikacja kliencka może nie mieć wbudowanego mechanizmu ponownego nawiązywania połączenia. Użytkownicy takich aplikacji po przełączeniu egzemplarza systemu SQL Server muszą zamknąć dany program, a następnie ponownie go uruchomić. Może wiązać się to z utratą danych wyświetlanych na bieżącym ekranie.

Właściwe powody stosowania klastrowania dla systemu SQL Server 2014

Klastry mają zwiększać dostępność egzemplarzy systemu SQL Server, w tym wszystkich baz danych (użytkowników i systemowych), loginów i zadań. Jednak ma to sens tylko w następujących warunkach.

- Firma zatrudnia doświadczonych administratorów baz danych, którzy potrafią zainstalować i skonfigurować klaster z systemem SQL Server oraz zarządzać nim.
- Koszty (i problemy) związane z przestojami są wyższe niż cena zakupu sprzętu i oprogramowania na potrzeby klastra oraz zarządzania nim.
- Potrafisz chronić dane i zapewnić ich nadmiarowość. Pamiętaj, że klastry nie zabezpieczają danych.
- Jeśli klaster ma obejmować oddalone od siebie jednostki i zdalne centra danych, musisz mieć certyfikowany sprzęt i oprogramowanie od niezależnego dostawcy.
- Masz wszystkie niezbędne urządzenia peryferyjne (zasilanie awaryjne itd.) potrzebne do obsługi środowiska z wysoko dostępnym serwerem.

Jeśli wszystkie te warunki są spełnione, firma jest dobrym kandydatem do zainstalowania systemu SQL Server w klastrze i może do tego przystąpić. Jeżeli jednak podane kryteria nie są spełnione i organizacja nie zamierza tego zmieniać, prawdopodobnie lepiej będzie w niej zastosować jedną z opisanych poniżej metod zapewniania wysokiej dostępności.

Co zastosować zamiast klastrowania?

Klastry z systemem SQL Server to tylko jedna z technik pomagających zapewnić wysoką dostępność egzemplarzy tego systemu. Rozwiązania gwarantujące wysoką dostępność mogą obejmować kilka warstw, aby zapewniać długi czas nieprzerwanej pracy systemu. W tym punkcie pokrótce omawiamy inne możliwości. Zaczynamy od najtańszych i najłatwiejszych w implementacji rozwiązań, a następnie przechodzimy do droższych i trudniejszych we wdrożeniu metod.

Serwer zapasowy w trybie offline

Serwer zapasowy w trybie *offline* (ang. *cold backup*) wymaga przygotowania dodatkowego fizycznego serwera, na którym można uruchomić system SQL Server 2014, gdy nastąpi awaria serwera produkcyjnego. Na takim dodatkowym serwerze nie ma zainstalowanego systemu SQL

Server 2014 ani kopii zapasowych baz danych. Dlatego zainstalowanie systemu, przywrócenie baz danych i przekierowanie aplikacji na nowy serwer wymaga czasu. Ponadto utracone mogą zostać niektóre dane, jeśli nie uda się przywrócić dzienników z ostatnimi transakcjami z serwera produkcyjnego, na którym nastąpiła awaria, i bazę trzeba będzie przywrócić z najnowszej kopii zapasowej.

Jeśli przestój lub utrata danych nie stanowią problemu, używanie serwera zapasowego w trybie offline to najtańszy sposób na zapewnienie ciągłości pracy firmy po wystąpieniu awarii systemu SQL Server 2014.

Serwer zapasowy w trybie online

Główna różnica między serwerem zapasowym w trybie offline a *serwerem zapasowym w trybie online* (ang. *warm backup*) polega na tym, że w drugim z tych modeli dodatkowy serwer (w trybie online) ma zainstalowany system SQL Server 2014 i może być używany jako serwer rozwojowy z — niekoniecznie aktualnymi — produkcyjnymi bazami danych. To pozwala zaoszczędzić dużo czasu potrzebnego na instalowanie i konfigurowanie systemu oraz szybsze udostępnienie serwera w środowisku produkcyjnym niż przy korzystaniu z serwera zapasowego w trybie offline. Nadal trzeba przekierować aplikacje bazodanowe i zaktualizować dane. Ponadto może nastąpić utrata danych, jeśli nie uda się przywrócić dzienników z ostatnimi transakcjami z serwera, który uległ awarii.

Przesyłanie dzienników

Przy przesyłaniu dzienników (ang. *log shipping*) używane są dwa egzemplarze systemu SQL Server — główny (produkcyjny) i pomocniczy. Na serwerze pomocniczym także zainstalowany i skonfigurowany jest egzemplarz systemu SQL Server 2014. Główna różnica między serwerem zapasowym w trybie online a przesyłaniem dzienników polega na tym, że przesyłanie dzienników umożliwia nie tylko automatyczne przywrócenie kopii zapasowej bazy danych z serwera produkcyjnego na serwerze pomocniczym, ale też automatyczne przywrócenie przesłanych dzienników transakcji z bazy danych. Zmniejsza to zakres ręcznie wykonywanych zadań i ogranicza utratę danych, ponieważ w najgorszym przypadku utracone zostają informacje z jednego dziennika transakcji. Jeśli np. na serwerze dzienniki transakcji są tworzone co piętnaście minut, utrata danych dotyczy najwyżej tego okresu.

UWAGA Przesyłanie dzienników opisujemy szczegółowo w rozdziale 18., „Przesyłanie dzienników w systemie SQL Server 2014”.

Replikacja

Wielu ekspertów uznaje, że replikacja danych w systemie SQL Server jest mechanizmem zwiększania dostępności. My jednak uważamy inaczej. Choć replikacja doskonale nadaje się do przenoszenia danych między egzemplarzami systemu SQL Server, nie jest dobrym narzędziem do zapewniania wysokiej dostępności. Jest zbyt skomplikowana i oferuje za mało możliwości w zakresie łatwego replikowania całych baz danych, aby uzasadnione były próby korzystania z niej do przełączania się między serwerami w reakcji na awarię (chyba że świetnie znasz się na replikacji).

UWAGA Szczegółowe omówienie replikacji znajdziesz w rozdziale 15.

Kopie lustrzane baz danych

Kopie lustrzane baz danych są pod wieloma względami dobrą alternatywą dla klastrów z systemem SQL Server. Kopie te, podobnie jak klastry, można wykorzystać do automatycznego przełączania egzemplarza systemu SQL Server, na którym nastąpiła awaria, na serwer lustrzany. Technikę tę można stosować dla pojedynczych baz danych. Największa różnica między klastrowaniem a kopiami lustrzanymi polega na tym, że utworzenie dwóch kopii lustrzanych zapewnia ochronę danych. Przy klastrowaniu współużytkowany dysk może być źródłem awarii. Ponadto kopie lustrzane mogą działać w miejscach znacznie oddalonych od siebie. Technika ta jest mniej kosztowna od klastrowania, jej skonfigurowanie i zarządzanie nią wymaga mniej wiedzy, a w niektórych sytuacjach przełączanie serwerów po awarii można w pełni zautomatyzować (podobnie jak przy klastrowaniu). W niektórych sytuacjach do zapewnienia wysokiej dostępności lepiej wykorzystać kopie lustrzane, a nie klastrowanie.

UWAGA W wersji SQL Server 2012 kopie lustrzane baz danych dodano do listy przestarzałych rozwiązań. W systemie SQL Server 2014 kopie lustrzane nadal są dostępne, jednak mogą zniknąć w przyszłych wersjach. Technologia ta została zastąpiona grupami dostępności AlwaysOn.

Grupy dostępności AlwaysOn

Grupy dostępności AlwaysOn służą do zapewniania wysokiej dostępności i przywracania stanu po awariach. Zostały wprowadzone w systemie SQL Server 2012. Umożliwiają maksymalizację dostępności połączonych w grupy baz danych użytkowników. Aby zastosować tę technologię, należy skonfigurować jedną lub więcej grup dostępności. Każda grupa obejmuje zbiór baz danych użytkowników, który można przełączyć awaryjnie na inny serwer jako odrębną jednostkę. Wykorzystywane są przy tym klastry WSFC oraz nazwa i adres IP egzemplarza systemu SQL Server.

Grupa dostępności obejmuje zbiór do dziewięciu jednostek, na które można się awaryjnie przełączyć — *replik zapewniających dostępność*. Obsługa dziewięciu replik to nowość wprowadzona w systemie SQL Server 2014. W wersji SQL Server 2012 obowiązywało ograniczenie do pięciu replik. Każda replika obejmuje własną kopię wszystkich baz danych z grupy dostępności. Zarządzanie danymi w bazach może się odbywać synchronicznie lub asynchronicznie. Jedną z replik, *replika podstawowa*, zawiera nadrzędną kopię każdej bazy danych. Replika główna udostępnia te bazy danych (są to *podstawowe bazy danych*) użytkownikom do odczytu i zapisu. Dla każdej podstawowej bazy danych pozostałe repliki (*repliki pomocnicze*) przechowują kopie awaryjne, które mogą działać w trybie tylko do odczytu. Warto zauważyć, że w trybie synchronicznym może działać do trzech replik, a ponadto można ustawić do dwóch węzłów automatycznie włączanych w momencie awarii.

UWAGA Więcej informacji na temat takich grup można znaleźć w rozdziale 25., „Grupy dostępności AlwaysOn”.

Mechanizmy klastrowania udostępniane przez niezależnych producentów

Oprócz Microsoftu istnieją też niezależne firmy, które udostępniają mechanizmy zapewniające wysoką dostępność systemu SQL Server. Rozwiązania te są przeważnie droższe od klastrów WSFC (i równie skomplikowane), jednak niektóre mają dodatkowe funkcje i zalety w porównaniu z narzędziami Microsoftu.

Co wybrać?

Choć to krótkie wprowadzenie pozwala zapoznać się z dostępnymi rozwiązaniami, może nie wystarczyć do podjęcia właściwej decyzji. Jeśli najlepsze rozwiązanie nie jest oczywiste, powinieneś przeanalizować opisane tu możliwości i na tej podstawie określić, co sprawdzi się w Twojej firmie.

Klastrowanie — ogólne omówienie

Zanim wdrożysz klastry i zaczniesz nimi zarządzać, musisz poznać ich działanie, a także dostępne opcje konfiguracyjne.

Jak działa klastrowanie?

W tym punkcie poznasz potrzebne pojęcia, takie jak węzły aktywne i pasywne, współużytkowana macierz dyskowa, quorum oraz sieci publiczne i prywatne. Dowiesz się też, jak działa egzemplarz systemu SQL Server w klastrze i jak przebiega przełączanie awaryjne.

Węzły aktywne i pasywne

Klastrer WSFC w systemie Windows Server 2012 obsługuje do sześćdziesięciu czterech węzłów. Jednak w typowym klastrze działają tylko dwa węzły. Jeden egzemplarz systemu SQL Server 2014 może działać w danym momencie w tylko jednym węźle. Gdy w tym egzemplarzu wystąpi błąd, następuje awaryjne przełączenie do innego węzła. Gdy chcesz używać w klastrze wielu egzemplarzy systemu SQL Server, powinieneś rozważyć użycie trzech lub więcej fizycznych węzłów. Duże klastry omawiamy dalej w tym rozdziale.

W dwuwęzłowym klastrze WSFC z systemem SQL Server jeden z fizycznych węzłów jest aktywny, a drugi — pasywny. Nie ma znaczenia, który z fizycznych serwerów w klastrze pełni funkcję aktywnego, należy jednak przypisać do serwerów odpowiednie role. Pozwoli to uniknąć wątpliwości dotyczących tego, który serwer w danym momencie pełni daną funkcję.

Węzeł aktywny obejmuje działający egzemplarz systemu SQL Server, który korzysta z baz danych zlokalizowanych we współużytkowanej macierzy dyskowej.

W węźle pasywnym nie ma działającego egzemplarza systemu SQL Server. Gdy węzeł jest pasywny, nie korzysta z produkcyjnych baz danych, ale pozostaje w stanie gotowości. Gdy w węźle aktywnym wystąpią problemy i nastąpi awaryjne przełączenie serwerów, węzeł pasywny zacznie automatycznie obsługiwać produkcyjne bazy danych i przetwarzać żądania od użytkowników. W takiej sytuacji węzeł pasywny staje się aktywnym, a aktywny — pasywnym (lub uszkodzonym, jeśli awaria uniemożliwia jego dalszą pracę).

Współużytkowana macierz dyskowa

Niezależne egzemplarze systemu SQL Server zwykle przechowują bazy danych na dysku lokalnym lub niewspółużytkowanym. Egzemplarze działające w klastrze przechowują dane we współużytkowanej macierzy dyskowej. *Współużytkowana* oznacza tu to, że wszystkie węzły z klastra WSFC są fizycznie połączone z macierzą, przy czym dostęp do umieszczonych w niej baz danych ma tylko aktywny węzeł. Aby zachowana została integralność baz danych, węzły klastra nigdy nie mogą jednocześnie korzystać ze współużytkowanego dysku.

Współużytkowaną macierzą dyskową mogą być macierze iSCSI, Fiber Channel, SAS, RAID 1, RAID 5 lub RAID 10 pracujące w niezależnej jednostce lub sieci SAN. Taka macierz musi mieć przynajmniej dwie partycje logiczne; pierwsza przechowuje bazy danych egzemplarza systemu SQL Server z klastra, natomiast druga jest używana jako dysk kworum (jeśli jest potrzebny). Ponadto jeżeli w klastrze ma działać mechanizm MSDTC (ang. *Microsoft Distributed Transaction Coordinator*), potrzebna jest trzecia partycja logiczna. System SQL Server 2014 obsługuje woluminy CSV (ang. *Cluster Share Volumes*), w których wiele węzłów może wczytywać dane i zapisywać je w jednym pliku VHD (ang. *Virtual Hard Drive*).

W wersjach SQL Server 2012 i 2014 systemowe bazy danych (master, model, msdb i tempdb) i bazy użytkowników można instalować na serwerze plików opartym na protokole SMB (ang. *Server Message Block*). Dotyczy to zarówno niezależnych egzemplarzy systemu SQL Server, jak i egzemplarzy FCI działających w trybie AlwaysOn. Więcej informacji o używaniu protokołu SMB do obsługi współużytkowanego magazynu danych znajdziesz na stronie <http://msdn.microsoft.com/en-us/library/hh759341.aspx>.

Kworum

Gdy oba węzły klastra działają i pełnią swoje funkcje (aktywnego i pasywnego węzła), komunikują się ze sobą przez sieć. Gdy zmienisz konfigurację w węźle aktywnym, zostanie ona automatycznie (i szybko) przeniesiona do pasywnego węzła, co zapewnia synchronizację między nimi.

Łatwo jednak przewidzieć, że aktywny węzeł może ulec awarii przed przesłaniem zmian przez sieć i wprowadzeniem ich w węzeł pasywnym. W takiej sytuacji modyfikacje w węźle pasywnym nigdy nie są wprowadzane. W zależności od natury zmian może to prowadzić do problemów — w tym do awarii obu węzłów z klastra.

Aby zapobiec takim scenariuszom, w klastrze WSFC stosowane jest kworum. *Kworum dyskowe* to plik dziennika, przypominający dzienniki baz danych. Rejestrowane są w nim wszystkie zmiany wprowadzane w aktywnym węźle. Jeśli zatem zarejestrowana zmiana nie została wprowadzona w pasywnym węźle (ponieważ nastąpiła awaria i węzeł aktywny nie mógł przesłać informacji przez sieć), węzeł pasywny w momencie przejmowania funkcji węzła aktywnego może wczytać dziennik kworum i sprawdzić modyfikacje. Pozwala to wprowadzić zmiany w węźle pasywnym, zanim stanie się aktywny. Jeśli napęd z dziennikiem kworum zostanie uszkodzony, klaster może przestać działać.

Dlatego na każdy zasób kworum w klastrze przypada jeden głos, a na podstawie większości głosów (uwzględniane są tu dostępne zasoby kworum) ustalane jest, czy klaster ma kontynuować działanie w danym węźle. Zapobiega to próbie przejęcia kontroli nad tym samym egzemplarzem systemu SQL Server przez więcej niż jeden węzeł klastra. Zasoby kworum biorące udział w głosowaniu to węzły klastra lub, w określonych sytuacjach, *dysk monitora* lub *udział plikowy monitora*. Każdy zasób kworum uwzględniany w głosowaniu (z wyjątkiem udziału plikowego monitora) zawiera kopię konfiguracji klastra. Usługa klastrowania dba o to, aby wszystkie kopie konfiguracji zawsze były zsynchronizowane ze sobą.

Od systemu Windows 2012 w klastrach WSFC obsługiwane są kwora dynamiczne (są one domyślnie włączone). Ten mechanizm dostosowuje wymagane wartości kworum (na podstawie dostępnych zasobów uczestniczących w głosowaniu) i modyfikuje głosy w klastrze, gdy węzły są po kolei wyłączane. Przykładowo w klastrze z pięcioma węzłami cztery węzły mogą zostać wyłączone na potrzeby planowanej konserwacji. Pozostaje wtedy ostatni węzeł, na którym działa system SQL Server. Następnie węzły są ponownie włączane, liczba głosów jest przywracana

i odbywa się sprawdzanie kworum. Aby zmienić to domyślne działanie, wybierz opcję *Advanced quorum configuration and witness selection*. Jeśli chcesz sprawdzić bieżące głosy w klastrze, uruchom poniższe polecenie cmdlet Get-ClusterNode powłoki Windows PowerShell:

```
PS C:\> Get-ClusterNode | ft name, dynamicweight, nodeweight, state -AutoSize
```

Oto cztery tryby kworum obsługiwane w klastrach WSFC.

- **Większość węzłów.** Każdy dostępny i połączony węzeł może głosować. Klaster działa tylko przy większości głosów.
- **Większość węzłów i dysków.** Głos przysługuje każdemu węzłowi oraz wyznaczonemu dyskowi z magazynu klastra (dyskowi monitora), jeśli tylko dane jednostki są dostępne i połączone. Klaster działa tylko przy większości głosów.
- **Większość węzłów i udziałów plikowych.** Głos przysługuje każdemu węzłowi oraz wyznaczonemu udziałowi plikowemu utworzonemu przez administratora (udziałowi plikowemu monitora), jeśli tylko dane jednostki są dostępne i połączone. Klaster działa tylko przy większości głosów.
- **Bez sprawdzania większości — tylko dysk.** Do uzyskania kworum w klastrze wystarczy, jeśli jeden węzeł jest dostępny i połączony z konkretnym dyskiem z magazynu klastra. Tylko węzły połączone z tym dyskiem mogą dołączyć do klastra. Ten dysk to pojedynczy punkt krytyczny, dlatego należy zadbać o jego niezawodność. *Dysk kworum* to napęd logiczny we współużytkowanej macierzy dyskowej przeznaczony do przechowywania kworum. Zgodnie z najlepszymi praktykami powinien on mieć pojemność około 1 gigabajta i być odporny na błędy.

W tabeli 16.1 opisujemy klastry na podstawie liczby węzłów i innych cech oraz przedstawiamy tryby kworum zalecane standardowo dla poszczególnych konfiguracji.

Tabela 16.1. Zalecane tryby kworum dla klastrów

Opis klastra	Zalecany tryb kworum
Nieparzysta liczba węzłów	Większość węzłów
Klaster o parzystej liczbie węzłów (nie geoklaster)	Większość węzłów i dysków
Geoklaster o parzystej liczbie węzłów	Większość węzłów i udziałów plikowych
Parzysta liczba węzłów, bez współużytkowanego magazynu	Większość węzłów i udziałów plikowych

Po uruchomieniu klastra możesz zmienić tryb kworum na podstawie wymagań użytkowników i liczby węzłów w klastrze. W tym celu wybierz opcję *Configure Cluster Quorum Settings*.

Sieci publiczne i prywatne

Każdy węzeł w klastrze musi mieć przynajmniej dwie karty sieciowe, aby rozwiązanie było kompletne. Pierwsza z kart powinna być podłączona do sieci publicznej, a druga — do prywatnej sieci klastra.

- *Sieć publiczna* to sieć, z którą łączą się aplikacje klienckie. W ten sposób komunikują się one z egzemplarzem systemu SQL Server z klastra, używając adresu IP i nazwy tego egzemplarza. Aby zapewnić nadmiarowość i wysoką dostępność, w sieci publicznej warto zastosować dwie powiązane karty sieciowe.

- *Sieć prywatna* służy wyłącznie do komunikacji między węzłami klastra. Jest używana przede wszystkim do *komunikacji kontrolnej* (ang. *heartbeat communication*). Istnieją dwie formy takiej komunikacji.
 - **LooksAlive.** Sprawdza (domyślnie co 5 sekund), czy w dostępnym węźle działa usługa SQL Server.
 - **IsAlive.** Sprawdza za pomocą procedury `sp_server_diagnostics`, czy system SQL Server akceptuje połączenia.

Te mechanizmy wykrywania stanu pozwalają stwierdzić, czy węzeł przestał działać lub czy brakuje mu zasobów. Jeśli tak się stało, węzeł pasywny przejmuje zadania produkcyjne.

Egzemplarz systemu SQL Server

Ciekawe jest to, że aplikacje klienckie systemu SQL Server „nie muszą wiedzieć”, jak przełączać obiekt, z którym się komunikują, z węzła, gdzie wystąpiła awaria, na nowy aktywny węzeł. Nie wymagają też informacji o konkretnych węzłach klastra (np. nazwy NETBIOS lub adresu IP poszczególnych węzłów). Dzieje się tak, ponieważ każdy egzemplarz systemu SQL Server pracujący w klastrze ma określoną nazwę sieci i adres IP, używane przez aplikacje klienckie do łączenia się z takim egzemplarzem. Oznacza to, że aplikacje klienckie nie łączą się za pomocą nazwy lub adresu IP konkretnych węzłów, ale używają nazwy sieci lub adresu IP klastra. Te dane się nie zmieniają (wskazują zawsze na aktywny węzeł). Każdy egzemplarz systemu SQL Server należy do grupy Failover Cluster Resource Group, która zawiera takie zasoby jak (są one przełączane wspólnie):

- nazwa sieci egzemplarza systemu SQL Server,
- adres IP,
- jeden lub więcej współużytkowanych dysków,
- usługa silnika bazodanowego systemu SQL Server,
- usługa SQL Server Agent,
- usługi SQL Server Analysis Services (jeśli są zainstalowane w tej samej grupie),
- jeden udział plikowy (jeśli zainstalowany jest mechanizm FILESTREAM).

Jak działa przełączanie awaryjne?

Załóżmy, że jeden egzemplarz systemu SQL Server działa w aktywnym węźle klastra, a węzeł pasywny jest dostępny i w razie potrzeby może podjąć zadania. W tej sytuacji węzeł aktywny komunikuje się z bazą danych i z kworum ze współużytkowanej macierzy dyskowej. Ponieważ w danym momencie tylko jeden węzeł ma dostęp do tej macierzy, pasywny węzeł nie może komunikować się z bazą ani z kworum.

System SQL Server obsługuje woluminy CSV (jest to mechanizm dostępu do danych w środowisku rozproszonym, umożliwiający jednoczesny dostęp do współużytkowanego systemu plików, ale nie do konkretnych plików), zatem obsługa magazynu danych klastra jest stosunkowo prosta. Ponadto węzeł aktywny przesyła sygnały kontrolne w prywatnej sieci, a węzeł pasywny wykrywa te sygnały, dlatego może przejąć zadania po wystąpieniu awarii. Klienci komunikują się z aktywnym węzłem za pomocą nazwy sieci wirtualnej i adresu IP egzemplarza systemu SQL Server pracującego w klastrze, który wykonuje zadania produkcyjne.

Założmy, że aktywny węzeł przestaje działać z powodu awarii zasilania. Węzeł pasywny (śledzący sygnały kontrolne aktywnego węzła) wykrywa brak sygnałów kontrolnych. Po określonym czasie węzeł pasywny uznaje, że węzeł aktywny uległ awarii, i inicjuje przełączanie awaryjne.

W ramach przełączania węzeł pasywny (stający się aktywnym) przejmuje kontrolę nad współużytkowaną macierzą dyskową i odczytuje zasób quorum, aby wykryć niesynchronizowane zmiany w konfiguracji. Ten węzeł zaczyna też używać nazwy sieci wirtualnej i adresu IP egzemplarza systemu SQL Server pracującego w klastrze. Ponadto węzeł przejmuje kontrolę nad bazami danych — uruchamia system SQL Server i przywraca bazy.

Potrzebny na to czas zależy od wielu czynników, w tym od wydajności sprzętu i liczby transakcji, które trzeba wykonać lub wycofać w ramach przywracania bazy danych. Po zakończeniu procesu przywracania bazy nowy węzeł aktywny jest udostępniany w sieci. Używane są przy tym nazwa sieci wirtualnej i adres IP pracującego w klastrze egzemplarza systemu SQL Server, dzięki czemu aplikacje klienckie po krótkiej przerwie mogą ponownie nawiązać połączenie i zacząć używać danego egzemplarza systemu SQL Server 2014.

W systemie SQL Server 2012 wprowadzono przydatny mechanizm Flexible Failover Policy. Jest on dostępny także w wersji SQL Server 2014. Ponieważ do wykrywania stanu służy procedura składowana `sp_server_diagnostics`, administrator może ustawić różne poziomy, przy których inicjowane jest przełączanie awaryjne. Do ich ustawiania służą właściwości usługi SQL Server w narzędziu Failover Cluster Manager. Dostępne są różne poziomy. Przy ich użyciu można ustawić warunki, po których spełnieniu następuje przełączenie awaryjne. Oto te poziomy.

- **Poziom 0.** Brak automatycznego przełączania awaryjnego i ponownego uruchamiania systemu.
- **Poziom 1.** Przełączanie awaryjne lub ponowne uruchomienie, gdy serwer nie działa.
 - Usługa SQL Server nie działa.
- **Poziom 2.** Przełączanie awaryjne lub ponowne uruchomienie, gdy serwer nie odpowiada.
 - Usługa SQL Server nie działa.
 - Egzemplarz systemu SQL Server nie odpowiada.
- **Poziom 3. (domyślny).** Przełączanie awaryjne lub ponowne uruchomienie po wykryciu krytycznych błędów serwera.
 - Usługa SQL Server nie działa.
 - Egzemplarz systemu SQL Server nie odpowiada.
 - Funkcja `sp_server_diagnostics` zwraca błąd systemowy.
- **Poziom 4.** Przełączanie awaryjne lub ponowne uruchomienie po wykryciu umiarkowanych błędów serwera.
 - Usługa SQL Server nie działa.
 - Egzemplarz systemu SQL Server nie odpowiada.
 - Funkcja `sp_server_diagnostics` zwraca błąd systemowy.
 - Funkcja `sp_server_diagnostics` zwraca błąd zasobów.
- **Poziom 5.** Przełączanie awaryjne lub ponowne uruchomienie po wystąpieniu dowolnego z wymienionych warunków.
 - Usługa SQL Server nie działa.
 - Egzemplarz systemu SQL Server nie odpowiada.
 - Funkcja `sp_server_diagnostics` zwraca błąd systemowy.
 - Funkcja `sp_server_diagnostics` zwraca błąd zasobów.
 - Funkcja `sp_server_diagnostics` zwraca błąd przetwarzania kwerendy.

UWAGA Więcej informacji na temat mechanizmu Flexible Failover Policy znajdziesz na stronie <http://msdn.microsoft.com/en-us/library/ff878664.aspx>.

Rodzaje klastrów

Na razie poznałeś proste klastry o dwóch węzłach i jednym egzemplarzu. Jednak istnieje też wiele innych sposobów klastrowania systemu SQL Server. Dwie inne popularne możliwości to klastry aktywny/aktywny i klastry z wieloma węzłami. Ponadto na jednym serwerze można utworzyć klaster z wieloma egzemplarzami systemu SQL Server. W następnych podpunktach szczegółowo omawiamy te podejścia.

Klaster z wieloma egzemplarzami

Omawiane do tego miejsca przykłady dotyczyły *klastra z jednym egzemplarzem*. Taki klaster obejmuje dwa węzły, z których tylko na jednym działa aktywny egzemplarz systemu SQL Server 2014. Jeśli aktywny węzeł ulegnie awarii, węzeł pasywny przejmie jedyny egzemplarz systemu SQL Server 2014 i stanie się węzłem aktywnym.

Aby zmniejszyć koszty zakupu sprzętu, niektóre firmy konfiguruja *klaster z wieloma egzemplarzami*. Także taki klaster obejmuje dwa węzły, jednak działają w nim dwa egzemplarze systemu SQL Server — po jednym na każdy fizyczny węzeł.

Zaletą klastra z wieloma egzemplarzami jest to, że pozwala lepiej wykorzystać dostępny sprzęt. Oba węzły w klastrze są używane (w odróżnieniu od tylko jednego w klastrach z jednym egzemplarzem). Wadą jest to, że po przełączeniu awaryjnym oba egzemplarze systemu SQL Server muszą działać na jednym serwerze fizycznym, co może zmniejszyć wydajność obu egzemplarzy, ponieważ wymaga udostępnienia każdemu z nich wystarczającej ilości pamięci. Aby złagodzić ten problem, można zakupić fizyczne serwery z rezerwami zasobów. Pozwoli to spełnić wymagania obu egzemplarzy po przełączeniu awaryjnym.

Istnieje jednak duże prawdopodobieństwo, że po przełączeniu obu egzemplarzy na jeden węzeł nastąpi pewien spadek wydajności. Jeśli ponadto w klastrze typu aktywny/aktywny działają dwa egzemplarze systemu SQL Server, każdy z nich potrzebuje własnego dysku logicznego w współużytkowanej macierzy dyskowej. Takie egzemplarze nie mogą współużytkować dysków logicznych.

Jeśli więc chcesz ograniczyć koszty sprzętu i nie przeszkadza Ci możliwe spowolnienie aplikacji, zastosuj klaster z dwoma węzłami i dwoma egzemplarzami.

Klastry z wieloma węzłami

Jeśli w przyszłości zamierzasz dodać więcej egzemplarzy systemu SQL Server 2014 pracujących w klastrze, rozważ trzecią możliwość, czyli *klastry z wieloma węzłami*. Konserwatywnie skonfigurowany klaster z trzema węzłami (*wiele egzemplarzy z pasywną rezerwą*) to dobre rozwiązanie, które zapewnia nadmiarowość i nie powoduje spowolnienia aplikacji po awaryjnym przełączeniu egzemplarza na pasywny węzeł. Jeśli nie przeszkadza Ci złożoność dużych klastrów, możesz dodać jeszcze więcej węzłów.

Liczba fizycznych węzłów obsługiwanych w klastrach z systemem SQL Server zależy od używanej wersji oprogramowania, a także od stosowanej wersji systemu operacyjnego. Pamiętaj też, że używany model obsługi quorum zależy od tego, czy liczba węzłów w klastrze jest parzysta, czy nieparzysta.

Zakup właściwego oprogramowania

Jednym z powodów, dla których ważne jest przeanalizowanie wymagań z zakresu klastrowania, jest to, że bezpośrednio wpływają one na to, jakie oprogramowanie jest potrzebne (co z kolei ma wpływ na koszty jego zakupu). Oto dostępne możliwości.

- **SQL Server 2014 Standard (wersja 32- lub 64-bitowa).** Obsługuje klastry obejmujące do dwóch węzłów.
- **SQL Server 2014 BI (wersja 32- lub 64-bitowa).** Obsługuje klastry obejmujące do dwóch węzłów.
- **SQL Server 2014 Enterprise (wersja 32- lub 64-bitowa).** Obsługuje tyle węzłów, ile maksymalnie może obsłużyć system operacyjny (obecnie są to sześćdziesiąt cztery węzły).

Jeśli potrzebujesz klastra z tylko dwoma węzłami, możesz zmniejszyć koszty, kupując wersje Windows Server 2012 Standard i SQL Server 2014 Standard. Jeżeli chcesz używać więcej niż dwóch węzłów, koszty licencji znacznie wzrosną, ponieważ będziesz potrzebował edycji SQL Server 2014 Enterprise (wersja Windows Server 2012 Standard obsługuje do sześćdziesięciu czterech węzłów).

UWAGA System SQL Server 2014 jest obsługiwany w systemie operacyjnym Windows 2008 SP2 i nowszych. Aby uzyskać maksymalną wydajność i dostęp do najnowszych funkcji, użyj ostatniej wersji systemu operacyjnego Windows. W czasie, gdy powstawała ta książka, była to wersja Windows Server 2012 R2.

Liczba używanych węzłów

W klastrach z dwoma węzłami egzemplarz systemu SQL Server działa w aktywnym węźle, natomiast węzeł pasywny nie zawiera działającego egzemplarza, jest jednak gotowy na uruchomienie go w momencie awaryjnego przełączenia. Ta sama zasada obowiązuje w klastrach z wieloma węzłami.

Żałujemy, że używasz klastra z trzema węzłami. W tej sytuacji w dwóch węzłach aktywnych działają odrębne egzemplarze systemu SQL Server, a trzeci węzeł fizyczny jest węzłem pasywnym dla obu pozostałych. Jeśli w jednym z węzłów aktywnych nastąpi awaria, węzeł pasywny przejmie jego zadania. Można też ustawić preferowane węzły, aby określić sekwencję przełączania awaryjnego między węzłami.

Teraz wyobraź sobie klastr z ośmioma węzłami. W tej sytuacji jest siedem węzłów aktywnych i jeden pasywny. Jeśli jeden z siedmiu węzłów aktywnych zawiedzie, węzeł pasywny przejmie jego zadania. Gdy węzłów jest dużo, węzeł pasywny jest bardziej przydatny, ponieważ pomaga uniknąć przeciążenia aktywnych węzłów po awarii kilku z nich.

W konfiguracji aktywne/pasywne zaletą stosowania wielu węzłów jest to, że do obsługi przełączania awaryjnego potrzeba mniej sprzętu. Jeśli którykolwiek z siedmiu węzłów aktywnych zawiedzie, można go przełączyć na węzeł pasywny. W klastrze z dwoma węzłami 50% sprzętu jest używane do zapewniania nadmiarowości. W klastrze z ośmioma węzłami jest to tylko 12,5% sprzętu.

Aby ustalić, ile węzłów ma zawierać klastr, należy uwzględnić ograniczenia biznesowe: budżet, poziom doświadczenia pracowników i stopień awersji do złożoności. Niektóre firmy chcą uruchamiać w klastrach wiele różnych egzemplarzy systemu SQL Server, aby jednak zachować prostotę, wybierają kilka klastrów aktywny/pasywny o dwóch węzłach zamiast jednego klastra z wieloma węzłami.

Klastrowanie wielu egzemplarzy systemu SQL Server na tym samym serwerze

W przedstawionym wcześniej omówieniu rodzajów klastrow wspomniano, że jeden egzemplarz systemu SQL Server działa zwykle na jednym fizycznym serwerze. Nie jest to jednak wymagane. Edycja SQL Server Enterprise obsługuje do sześćdziesięciu czterech egzemplarzy tego systemu w klastrze. Jeśli współużytkowanym magazynem pamięci są udziały plikowe SMB, ograniczenie wynosi pięćdziesiąt egzemplarzy systemu SQL Server 2014. Skuteczność tego podejścia zależy od wymagań biznesowych, możliwości sprzętu, umów o gwarantowanym poziomie świadczenia usług i poziomie doświadczenia jednostki informatycznej zarządzającej danym rozwiązaniem.

Klasy mają zwiększać dostępność systemu. Dodanie wielu egzemplarzy systemu SQL Server do klastra zwiększa poziom złożoności, co może prowadzić do wzrostu ryzyka i liczby miejsc zagrożonych awarią. Jednak ze złożonością można sobie poradzić, jeśli dostępni są doświadczeni informatycy. Porozmawiaj z nimi, gdy podejmujesz decyzję.

Aktualizowanie klastrow z systemem SQL Server

Jeśli Twoja firma przypomina inne, prawdopodobnie używa w środowisku produkcyjnym klastrow ze starszymi wersjami systemu SQL Server. W takiej sytuacji należy zdecydować, czy aktualizować je do wersji SQL Server 2014. Oto dostępne możliwości.

- Rezygnacja z aktualizacji.
- Przeprowadzenie aktualizacji do wersji SQL Server 2014 w miejscu.
- Zbudowanie klastra od nowa z wersją SQL Server 2014. Jeśli używasz systemu Windows 2008 SP2 lub nowszego, możesz też pozostawić klaster WSFC bez zmian i zainstalować w nim nowy system SQL Server 2014 oraz przenieść do niego bazy danych.

W tym podrozdziale omawiamy wszystkie trzy możliwości.

Rezygnacja z aktualizacji

To łatwa decyzja. Takie rozwiązanie jest proste i nie wymaga ponoszenia kosztów. Sam fakt, że pojawiła się nowa wersja systemu SQL Server, nie oznacza, że musisz przeprowadzić aktualizację. Jeśli obecny klaster z systemem SQL Server działa zadowalająco, możliwe, że nie warto marnować środków i sił na jego aktualizację. Odpowiednio skonfigurowany klaster z systemem Windows 2012 R2 może stabilnie działać z wersjami SQL Server 2008 R2 i 2012.

Jednak system SQL Server 2014 zapewnia skalowalność, łatwość użytkowania, nowe możliwości, niezawodność i nowe funkcje. Możliwe, że zechcesz to wykorzystać. Jednak przed aktualizacją przeprowadź analizy i ustal, czy potrzebujesz nowych funkcji systemu SQL Server 2014. Jeśli nie, możesz zdecydować się na pozostawienie bieżącej wersji systemu SQL Server (pod warunkiem, że nadal jest wspierana przez Microsoft).

Aktualizowanie klastra do wersji SQL Server 2014 w miejscu

Przed zapoznaniem się z procesem aktualizacji klastra WSFC do wersji SQL Server 2014 sprawdź, z jakiego systemu operacyjnego korzystasz. Jeśli jest to system Windows Server 2008 SP2 lub nowszy, możesz przeprowadzić aktualizację w miejscu do wersji SQL Server 2014.

Potrzebujesz systemu operacyjnego Windows 2008 SP2 lub nowszego (najlepiej, jeśli będzie to Windows 2012 R2). Aktualizacja klastra WSFC w starszych wersjach nie jest możliwa. Niezbędne jest wtedy utworzenie klastra od nowa.

Jeśli używasz klastra WSFC z systemu Windows 2008 SP2 i chcesz zaktualizować system SQL Server do wersji 2014, możesz przeprowadzić aktualizację bez zatrzymywania środowiska (ang. *rolling upgrade*) z minimalnym czasem przestoju. W tym celu wykonaj opisane niżej kroki.

1. Ustal węzeł klastra, który chcesz zaktualizować jako pierwszy.
2. Przełącz awaryjnie wszystkie egzemplarze systemu SQL Server z tego węzła. W efekcie nie będzie na nim działał żaden egzemplarz i węzeł stanie się pasywny.
3. Zainstaluj prerekwizyty w każdym pasywnym węźle i zaktualizuj współużytkowane komponenty systemu SQL Server.
4. Jeśli trzeba, ponownie uruchom węzły pasywne.
5. Uruchom instalację systemu SQL Server 2014, aby przeprowadzić aktualizację w miejscu. Przed wprowadzeniem zmian powinieneś wykonać procedurę sprawdzania poprawności aktualizacji. Przygotuj środowisko testowe, uruchom narzędzie SQL Server 2014 Upgrade Advisor, zaktualizuj środowisko testowe do wersji SQL Server 2014, odtwórz w nim ślad ze środowiska produkcyjnego (zarejestrowany za pomocą programu SQL Server Profiler), przetestuj możliwość nawiązywania połączeń przez aplikacje i przeprowadź testy wydajności. Jeśli nie zastosujesz tej procedury, aktualizacja może się nie powieść. W efekcie konieczne może być wycofanie zmian, co w klastrach jest bardziej skomplikowane niż w środowiskach z niezależnym systemem SQL Server.
6. Po zmodyfikowaniu więcej niż połowy pasywnych węzłów proces aktualizacji automatycznie zainicjuje przełączanie awaryjne do zaktualizowanych węzłów. Generowana jest przy tym lista węzłów, które można ustawić jako właścicieli zasobów w trakcie aktualizowania pozostałych węzłów.

UWAGA Nie trzeba aktualizować wszystkich egzemplarzy systemu SQL Server. W klastrze WSFC jednocześnie mogą działać egzemplarze wersji SQL Server 2008, 2008 R2, 2012 i 2014.

Tworzenie klastra od nowa

Budowanie klastra od nowa jest dobrym pomysłem, jeśli spełniony jest dowolny z poniższych warunków.

- Musisz zaktualizować używany sprzęt (ponieważ jest za stary lub za mało wydajny).
- Na serwerze używana jest starsza wersja systemu operacyjnego Windows.
- Obecny klaster pracuje niestabilnie.
- Nie lubisz aktualizować oprogramowania w miejscu i wolisz instalować je od nowa.

Jeśli zdecydujesz się na budowanie klastra od nowa, musisz ustalić, czy chcesz zainstalować oprogramowanie w nowym, czy w starym sprzęcie. Nowy sprzęt pozwala tworzyć klaster i przeprowadzać testy we własnym tempie, podczas gdy w środowisku produkcyjnym wciąż działa dawny klaster. Pomaga to upewnić się o jakości rozwiązania, a jednocześnie powoduje mniejszy stres niż w sytuacji, gdy trzeba wykorzystać stary sprzęt i w krótkim czasie zbudować klaster od nowa.

Jeżeli nie możesz pozwolić sobie na zakup nowego sprzętu, musisz ustalić czas przestoju, przez który system będzie niedostępny z powodu budowania klastra. Może to zająć od czterech do dwunastu godzin (w zależności od wielkości i złożoności klastra). Oprócz czasu niedostępności klastra należy też uwzględnić ryzyko wystąpienia nieoczekiwanych problemów. Możliwe, że w połowie procesu aktualizacji popełnisz błąd w trakcie instalacji i będziesz musiał zacząć pracę od początku. Ponieważ trudno tu o pewność, powinieneś najpierw oszacować, ile czasu aktualizacja zajmie w optymalnych warunkach. Następnie pomnóż tę wartość przez dwa, aby oszacować czas przestoju. Dzięki temu użytkownicy będą odpowiednio przygotowani.

Niezależnie od tego, czy przy aktualizacji używasz nowego, czy starego sprzętu, musisz uwzględnić dwie dodatkowe kwestie.

- Czy zamierzasz ponownie wykorzystać nazwę i adres IP obecnego egzemplarza systemu SQL Server z klastra, czy chcesz je zmienić?
- W jaki sposób zamierzasz przenieść dane z poprzedniego klastra do nowego?

Klienci, które korzystają z obecnego klastra z systemem SQL Server, używają do tego nazwy i adresu IP egzemplarza tego systemu. Jeśli chcesz, aby klienci nadal używały tych samych danych, musisz ponownie wykorzystać je w nowym klastrze. Jest to najczęściej stosowane podejście, ponieważ zwykle łatwiej zmienić jedną nazwę i adres IP działającego w klastrze egzemplarza systemu SQL Server, niż zrekonfigurować dziesiątki (a czasem nawet setki) klientów używających tego systemu pracującego w klastrze WSFC.

Jeśli przy aktualizacji korzystasz ze starego sprzętu, ponowne użycie dawnej nazwy i adresu IP jest proste, ponieważ najpierw wyłączany jest stary klaster, a następnie zostaje włączony nowy. Dlatego nigdy nie dzieje się tak, że w dwóch klastrach używana jest ta sama nazwa i adres IP (co jest niedozwolone).

Jeżeli korzystasz z nowego sprzętu, na potrzeby testów musisz przypisać pracującemu w klastrze egzemplarzowi systemu SQL Server nazwę sieci wirtualnej i adres IP. Nie można jednak podać dawnych danych, ponieważ są obecnie używane. W takiej sytuacji trzeba zastosować tymczasową nazwę sieci i tymczasowy adres IP. Gdy będziesz gotowy zastąpić dawny klaster nowym, wykonaj następujące kroki.

1. Zarchiwizuj dane.
2. Usuń system SQL Server z dawnego klastra lub wyłącz ten klaster.
3. W nowym klastrze ustaw dla egzemplarza systemu SQL Server nazwę sieci wirtualnej i adres IP dawnego klastra.
4. Przywróć dane.

Sposób przenoszenia danych z dawnego klastra do nowego zależy zarówno od wielkości baz danych, jak i (w pewnym stopniu) od tego, czy używasz starego, czy nowego sprzętu.

Niezależnie od wybranego podejścia zacznij od zarchiwizowania wszystkich baz danych. Pamiętaj, aby ustalić wszystkie obiekty z systemowych baz danych (np. zadania systemu SQL Server, pakiety SSIS i loginy) i odtworzyć je w nowym klastrze WSFC. Jeśli korzystasz ze starego sprzętu, wystarczy utworzyć kopię zapasową baz danych użytkowników lub je odłączyć. Po zbudowaniu nowego klastra należy przywrócić bazy użytkowników lub ponownie je podłączyć.

Jeśli przenosisz dane na nowy sprzęt lub zmieniasz lokalizację plików baz danych, powinieneś najpierw utworzyć ich kopie zapasowe lub odłączyć bazy. Następnie przenieś dane do nowego klastra lub do nowej lokalizacji. Po utworzeniu nowego klastra przywróć bazy użytkowników lub ponownie je podłącz.

UWAGA Nie mamy tu miejsca na szczegółowe opisywanie kroków potrzebnych w każdym scenariuszu (np. przy zmianie liter napędu). Najważniejsze jest to, aby zaplanować wszystkie kroki i, jeśli to możliwe, przeprowadzić próby przed przełączeniem systemów.

Plan wycofywania zmian

Niezależnie od wybranej metody aktualizowania klastrów do wersji SQL Server 2014 potrzebny jest *plan wycofywania zmian*. Obejmuje on czynności wykonywane po nieudanej aktualizacji. Taki plan zwykle polega na ponownym zainstalowaniu systemu SQL Server oraz przywróceniu baz danych (systemowych i użytkowników). Prowadzi to do przynajmniej kilkugodzinnego przestoju.

Ponieważ każde środowisko wygląda inaczej, nie da się opracować jednego zestawu kroków pasującego do wszystkich planów wycofywania zmian. Dlatego w trakcie planowania aktualizacji zastanów się, co może się nie powieść, i ustal, jak zamierzasz przywrócić system po wystąpieniu problemów. To, czy zachowasz stanowisko, może zależeć od jakości planów wycofywania zmian. Największe ryzyko związane jest z aktualizacją systemu SQL Server w miejscu, ponieważ nie można wtedy anulować zmian. Jeśli zatem to możliwe, należy przeprowadzać aktualizację bez usuwania wcześniejszej wersji (i — oczywiście — koniecznie utworzyć kopie zapasowe wszystkich baz danych przed rozpoczęciem procesu aktualizacji).

Która metoda aktualizacji jest najlepsza?

Z uwagi na doświadczenia w trakcie aktualizacji zawsze staramy się tworzyć klastry od początku na nowym sprzęcie. Jest to najłatwiejsze, najszybsze, najmniej ryzykowne i najmniej stresujące rozwiązanie. Niestety, zdarza się, że z powodu czynników, nad którymi nie masz kontroli, nie możesz zastosować tego podejścia. W takiej sytuacji musisz dostosować się do sytuacji. Kluczem do udanej aktualizacji jest staranne zaplanowanie tej operacji, przeprowadzenie dokładnych testów i — oczywiście — opracowanie kompletnego planu wycofywania zmian.

Przygotowanie do klastrowania

Przy tworzeniu klastrów z systemem SQL Server 2014 ważne są szczegóły. Jeśli upewnisz się, że wszystkie kroki są wykonywane prawidłowo i we właściwej kolejności, instalowanie klastra będzie przebiegać płynnie oraz stosunkowo szybko i bezproblemowo. Jeżeli jednak nie lubisz czytać instrukcji i wolisz administrować komputerami metodą prób i błędów, przygotuj się na frustrację oraz długie i wielokrotne instalowanie klastra.

Najlepszą metodą na zapewnienie płynnej instalacji klastra jest opracowanie szczegółowego planu do poziomu poszczególnych ekranów. Przyznajemy, to nudne i żmudne zadanie, jednak zmusza do przemyślenia każdej opcji oraz jej wpływu na instalację i firmę (gdy system zacznie działać w środowisku produkcyjnym). Ponadto opracowany plan będzie przydatny, gdy będziesz ponownie budował klastry, i posłuży jako doskonała dokumentacja planu przywracania stanu po awarii.

Przygotowywanie infrastruktury

Zanim zaczniesz budować klastery z systemem SQL Server 2014, musisz się upewnić, że masz odpowiednią infrastrukturę sieciową. Poniżej znajdziesz listę kontrolną z elementami potrzebnymi przed rozpoczęciem instalowania takiego klastra. W wielu przypadkach za poszczególne elementy odpowiadają inne osoby z działu informatycznego, jednak zanim zaczniesz budować klastery, musisz sprawdzić, czy wszystkie komponenty są dostępne.

- W sieci musi działać przynajmniej jeden kontroler domen usługi Active Directory (a najlepiej dwa, aby zapewnić rezerwę).
- W sieci musi działać przynajmniej jeden serwer DNS (a najlepiej dwa, aby zapewnić rezerwę).
- W sieci muszą być dostępne porty przełącznika dla kart z sieci publicznej używanych przez węzły klastra. Skonfiguruj te porty tak, aby pasowały do ręcznie ustawionych opcji kart sieciowych używanych w węzłach klastra. System SQL Server 2014 obsługuje węzły z różnych podsieci (ta funkcja jest dostępna od wersji SQL Server 2012).
- Musisz zapewnić adresy IP dla wszystkich kart z sieci publicznej (zalecane rozwiązanie) lub zastosować protokół DHCP (ang. *Dynamic Host Configuration Protocol*).
- Musisz zdecydować, jak skonfigurować prywatną sieć do przesyłania sygnałów kontrolnych. Możesz zastosować bezpośrednie połączenie między kartami sieciowymi za pomocą skrętki (jest to możliwe tylko dla klastrów z dwoma węzłami) lub wykorzystać hub albo przełącznik. Powinien to być inny hub (lub przełącznik) niż ten używany dla sieci publicznej; pozwala to zapewnić rezerwę. Choć sieć do przesyłania sygnałów kontrolnych nie jest już wymagana, nadal zaleca się jej stosowanie.
- Musisz zapewnić adresy IP dla kart z sieci prywatnej. Zwykle należy wykorzystać podsieci, np. 10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255 lub 192.168.0.0 – 192.168.255.255. Pamiętaj, że dotyczy to sieci prywatnej dostępnej tylko dla węzłów klastra.
- Upewnij się, że dostępne jest odpowiednie zasilanie dla węzłów nowego klastra i współużytkowanej macierzy dyskowej.
- Upewnij się, że dla wszystkich węzłów klastra i współużytkowanej macierzy dyskowej dostępne jest rezerwowe zasilanie bateryjne.
- Jeśli jeszcze tego nie zrobiłeś, utwórz konto usługowe dla systemu SQL Server przeznaczone dla usług tego systemu pracujących w klastrze. Musi to być konto domenowe z niewygasającym hasłem.
- Ustal nazwę klastra WSFC i zapewnij dla niego adres IP. Ta nazwa będzie służyła do zarządzania klastrem po jego utworzeniu.
- Określ nazwę egzemplarza systemu SQL Server 2014 z klastra i zapewnij dla niego adres IP. Ta nazwa posłuży klientom do łączenia się z klastrem.

Opisane przygotowania okażą się istotne w trakcie instalacji klastra.

Przygotowywanie sprzętu

Wybór sprzętu związany jest z określonymi problemami, dlatego analiza i konfiguracja go wymaga często bardzo dużo czasu. Po części wynika to z bardzo bogatej oferty. Niektóre konfiguracje się sprawdzają, a niektóre nie. Niestety, nie istnieje kompletny poradnik, który pozwoli zawsze dokonać trafnego wyboru. Każdy producent sprzedaje inny sprzęt, a dostępna oferta ciągle się

zmienia. Ponadto pojawiają się nowe i zaktualizowane sterowniki. Jest to bardzo zmienna dziedzina bez prostych rozwiązań. Mimo to, poniżej piszemy, co powinieneś wiedzieć, aby podołać do wyboru odpowiedniego sprzętu dla klastra z systemem SQL Server 2014.

Znajdowanie drogi w gąszczu oprogramowania

W tym podpunkcie omawiamy podstawowy sprzęt potrzebny dla klastra z systemem SQL Server. Aby uprościć tekst, opisujemy tylko klastry typu aktywny/pasywny z dwoma węzłami (choć przedstawione zalecenia dotyczą też klastrów z wieloma węzłami). Ponadto niektóre klastry WSFC działają z dyskiem kworum, dlatego też należy go uwzględnić. Poniżej przedstawiamy sugerowane przez nas minimalne wymagania sprzętowe. Wymagania dla klastra z systemem SQL Server 2014 proponowane przez Microsoft są nieco niższe.

UWAGA Każdy węzeł w klastrze powinien być identyczny. Pozwala to uniknąć problemów z instalacją i zarządzaniem.

Oto minimalna specyfikacja węzłów z serwerami.

- Procesor dwurdzeniowy o szybkości 2 gigaherców lub wyższej, z 2 megabajtami pamięci L2 Cache (32- lub 64-bitowy).
- Przynajmniej 4 gigabajty pamięci RAM.
- Lokalny podłączony przez SCSI C: na aplikacje (z kopiami lustrzanymi w macierzy RAID 10, 5 lub 1), mający pojemność przynajmniej 9 gigabajtów.
- Napęd DVD podłączony przez SCSI.
- Połączenie SCSI dla lokalnego dysku i napędu DVD.
- Połączenie SCSI lub światłowodowe ze współużytkowaną macierzą dyskową, siecią SAN lub współużytkowanymi dyskami iSCSI.
- Rezerwowe źródło zasilania.
- Karta dla sieci prywatnej.
- Karta dla sieci publicznej.
- Mysz, klawiatura i monitor (mogą być współużytkowane).

Współużytkowana macierz dyskowa powinna być macierzą RAID 10, 5 lub 1 podłączoną przez szybką magistralę SCSI. W klastrach WSFC magistrala SCSI jest obsługiwana tylko w klastrach o dwóch węzłach. Jeśli chcesz używać w klastrze większej liczby węzłów, musisz zastosować interfejs iSCSI, udziały plikowe SMB 3.0, macierz dyskową podłączoną światłowodem lub sieć SAN z macierzą RAID 10, 5 albo 1 i odpowiednio szybkim połączeniem.

Jeśli dopiero uczysz się tworzyć klastry, spytaj dostawcę, jaki sprzęt proponuje. Pamiętaj, że w klastrze ma działać system SQL Server 2014, dlatego upewnij się, czy wybrany sprzęt poradzi sobie z prognozowanym obciążeniem w środowisku produkcyjnym.

Montaż sprzętu

Jeśli jesteś administratorem baz danych, możliwe, że nie będziesz zajmował się montażem sprzętu. Poniżej przedstawiamy jednak kroki, które zazwyczaj wykonuje się w trakcie przygotowywania sprzętu dla klastra.

1. Zainstaluj i skonfiguruj sprzęt dla każdego węzła w klastrze, tak jakby miał działać jako niezależny serwer. Proces ten obejmuje instalację najnowszych zatwierdzonych sterowników i firmware'u.
2. Po przygotowaniu sprzętu zainstaluj system operacyjny, najnowszy zestaw poprawek i dodatkowe potrzebne sterowniki. Następnie uruchom program Windows Update.
3. Skonfiguruj węzły pod kątem pracy w sieci publicznej. Aby ułatwić sobie identyfikację sieci używanej dla połączeń publicznych, nazwij ją **public network**.
4. Skonfiguruj prywatną sieć do przesyłania sygnałów kontrolnych. Aby ułatwić sobie identyfikację tej sieci, nazwij ją **private network**.
5. Zainstaluj i skonfiguruj współużytkowaną macierz dyskową, docelowe jednostki iSCSI lub światłowodową sieć SAN.
6. Zainstaluj i skonfiguruj w każdym węźle karty sieciowe iSCSI lub światłowodowe. Zainstaluj też najnowsze sterowniki. Konieczne może być też skonfigurowanie inicjatorów iSCSI, aby mogły łączyć się z dyskami.
7. Podłączaj po kolei każdy węzeł do współużytkowanej macierzy dyskowej, dysku iSCSI lub sieci SAN. Stosuj się do instrukcji dotyczących używanego sprzętu. Bardzo ważne jest, aby podłączać węzły pojedynczo. Za każdym razem włącz tylko jeden węzeł, podłącz go do współużytkowanej macierzy dyskowej lub sieci SAN i skonfiguruj. Po skonfigurowaniu węzła wyłącz go, włącz następny węzeł, skonfiguruj go itd. Jeśli nie zastosujesz tej procedury, narażasz się na uszkodzenie konfiguracji dysków w węzłach, co wymaga rozpoczęcia całego procesu od nowa.
8. Wykorzystaj program Disk Administrator do skonfigurowania i sformatowania dysków ze współużytkowanej macierzy dyskowej. Macierz ta musi obejmować przynajmniej dwa dyski logiczne. Pierwszy służy do przechowywania baz systemu SQL Server, a drugi pełni funkcję dysku kworum. Dysk przeznaczony na dane musi mieć wystarczającą pojemność, aby je pomieścić, a dysk kworum powinien mieć przynajmniej 500 megabajtów (jest to najmniejsza wielkość pozwalająca na wydajną pracę woluminów NTFS). W trakcie konfigurowania dysków za pomocą programu Disk Administrator w każdym węźle klastra należy zastosować tę samą literę napędu dla poszczególnych dysków z macierzy lub sieci SAN. Możesz np. we wszystkich węzłach ustawić literę F: dla dysku z danymi i literę Q: dla dysku kworum.
9. Po podłączeniu całego sprzętu upewnij się za pomocą przeglądarki zdarzeń systemu Windows, że nie występują problemy. Dopiero potem zacznij instalować usługi związane z klastrowaniem. Choć niektóre testy diagnostyczne sprzętu możesz przeprowadzić przed zainstalowaniem systemu operacyjnego, kompletne testy powinieneś wykonać dopiero po jego zainstalowaniu.
10. Upewnij się, że możesz skontaktować się za pomocą sygnału ping z węzłami klastra przez sieć publiczną i prywatną. Sprawdź też, czy możesz skontaktować się w ten sposób z kontrolerem domeny i z serwerem DNS. W ten sposób sprawdzisz, że są dostępne.
11. Zapisz plik na współużytkowanych dyskach jednego węzła; plik powinien być widoczny w drugim węźle.
12. Sprawdź, czy we wszystkich węzłach używane są takie same litery dysków i czy prowadzą do tych samych dysków.

Po skonfigurowaniu i przetestowaniu całego sprzętu możesz przejść do instalowania klastra WSFC.

Tworzenie klastra w systemie operacyjnym Windows Server 2012 R2

Przed zainstalowaniem w klastrze systemu SQL Server 2014 musisz najpierw zainstalować usługi WSFC z systemu Windows Server 2012 R2. Po ich zainstalowaniu i przetestowaniu możesz dodać do klastra system SQL Server. W tym podrozdziale omawiamy krok po kroku ogólny proces instalowania i konfigurowania klastra WSFC.

Co trzeba zrobić przed instalacją klastra w systemie Windows 2012 R2?

Aby zainstalować klaster WSFC, musisz wykonać szereg ważnych kroków. Jest to istotne zwłaszcza wtedy, gdy nie budowałeś węzłów klastra, ponieważ pozwoli Ci się upewnić przed rozpoczęciem instalacji samego klastra, że wszystko działa prawidłowo. Po wykonaniu tych kroków możesz zainstalować klaster WSFC z systemu Windows 2012 R2. Oto potrzebne operacje.

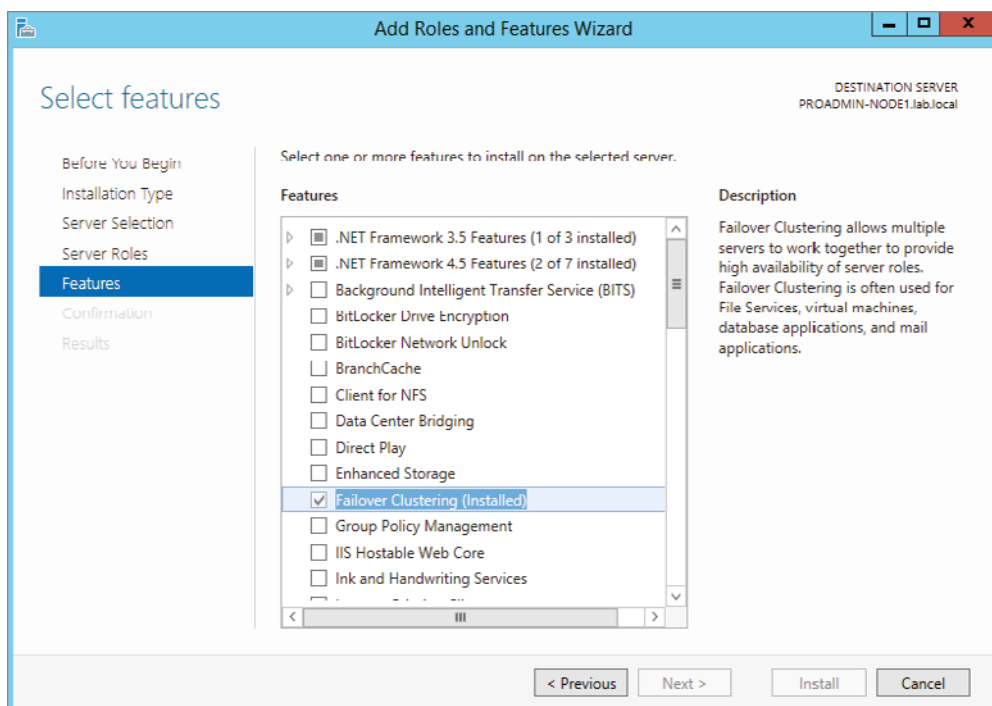
1. Upewnij się, że wszystkie węzły fizyczne pracują prawidłowo i są identycznie skonfigurowane (mają te same sterowniki, oprogramowanie i sprzęt).
2. Upewnij się, że żaden z fizycznych węzłów nie został skonfigurowany jako kontroler domeny i nie obsługuje krytycznych usług (takich jak usługa Exchange). Ponadto w środowisku powinno działać kilka kontrolerów domeny.
3. Sprawdź, czy wszystkie dyski działają w formacie NTFS i nie są skompresowane.
4. Upewnij się, że sieć publiczna i sieć prywatna są poprawnie zainstalowane oraz skonfigurowane.
5. Klaster WSFC z systemem SQL Server obsługuje dyski lokalne tylko na potrzeby instalacji plików bazy tempdb. Upewnij się, że we wszystkich węzłach klastra ścieżka do plików danych i plików dziennika tej bazy jest poprawna. Jeśli w trakcie przełączania awaryjnego okaże się, że katalogi z plikami bazy tempdb nie są dostępne w docelowym węźle, nie będzie można ich użyć w systemie SQL Server.
6. Upewnij się, że wyłączyłeś NetBIOS dla wszystkich kart z sieci prywatnej.
7. Sprawdź, czy na współużytkowanych dyskach nie ma udziałów sieciowych.
8. Jeśli zamierzasz stosować szyfrowanie w systemie SQL Server, we wszystkich węzłach klastra zainstaluj certyfikat serwera z pełną nazwą DNS egzemplarza systemu SQL Server pracującego w klastrze.
9. Sprawdź wszystkie dzienniki błędów, aby się upewnić, że nie występują żadne problemy. Jeśli pojawiły się jakieś usterki, napraw je przed kontynuowaniem instalowania klastra.
10. We wszystkich węzłach klastra dodaj konta usługowe systemu SQL Server i klastra do grupy lokalnych administratorów.
11. Upewnij się, że w węzłach klastra nie zainstalowano oprogramowania antywirusowego. Takie oprogramowanie może zmniejszyć dostępność klastra. Jeśli chcesz wykrywać wirusy w klastrze WSFC, skanuj węzły zdalnie, z poziomu innego komputera.

Trzeba sprawdzić wiele rzeczy, a każda z nich jest ważna. Jeśli coś pominiesz, każdy z opisanych elementów może negatywnie wpłynąć na poprawną instalację lub pracę klastra.

Instalowanie klastra WSFC z systemu Windows Server 2012 R2

Po przygotowaniu wszystkich fizycznych węzłów i współużytkowanej macierzy dyskowej, urządzeń iSCSI lub sieci SAN możesz przejść do instalowania klastra WSFC z systemu Windows 2012 R2. W tym punkcie opisujemy ten proces od początku do końca.

Na początku musisz włączyć obsługę klastrow we wszystkich węzłach z systemem Windows 2012 R2. W systemie 2008 zdecydowano się domyślnie wyłączyć obsługę klastrow, aby zmniejszyć liczbę działających komponentów. W celu włączenia obsługi klastrow wybierz opcję *Add Roles and Features Wizard* w narzędziu Server Manager. Zaznacz opcję *Failover Clustering*, co pokazano na rysunku 16.1, a kreator przeprowadzi instalację. Nie powoduje to utworzenia klastra z dostępnymi komputerami, ale prowadzi do zainstalowania na serwerze komponentów potrzebnych do zarządzania klastrem i obsługi przełączania awaryjnego. Ten krok trzeba powtórzyć na każdym fizycznym węźle użytym w klastrze WSFC.



Rysunek 16.1. Włączanie obsługi klastrow

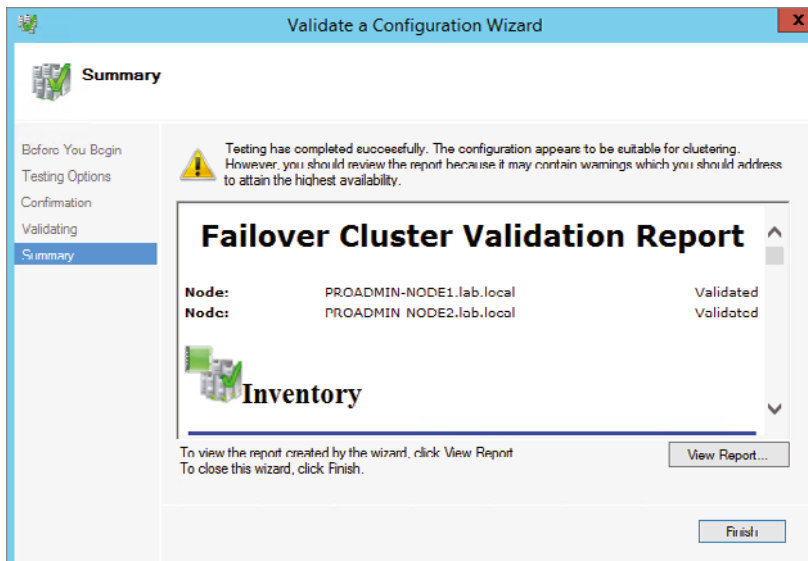
Sprawdzanie węzłów klastra WSFC

Przed utworzeniem klastra należy sprawdzić, czy w fizycznych węzłach używany jest obsługiwany sprzęt. Jest tak, jeśli węzły przechodzą testy sprawdzania poprawności klastra. Aby przeprowadzić taki test, otwórz narzędzie Failover Cluster Management i wybierz opcję *Validate a Configuration*. Pojawi się kreator *Validate a Cluster Configuration Wizard*, który przeprowadzi pełną diagnozę klastra przed zainstalowaniem komponentów.

Ponadto możesz uruchomić ten kreator po zainstalowaniu klastra, aby się upewnić, że wymagania sprzętowe są spełnione i wszystkie urządzenia działają prawidłowo.

UWAGA Jeśli klaster WSFC przejdzie sprawdzanie poprawności, wiadomo, że produkty Microsoftu będą go obsługiwać. Minęły już czasy, gdy trzeba było sprawdzać długą listę kontrolną, aby ustalić, czy klaster będzie działał w danej konfiguracji sprzętowej.

W kreatorze *Validate a Configuration Wizard* pojawia się pytanie o to, które fizyczne lub potencjalne węzły chcesz sprawdzić. Dodaj do listy wszystkie węzły i kliknij przycisk *Next*. Pojawi się wtedy pytanie o typy przeprowadzanych testów. Po wybraniu i uruchomieniu testów otrzymasz wizualne potwierdzenie po zakończeniu każdego ich etapu. Przedstawiono to na rysunku 16.2. Jeśli test zakończy się niepowodzeniem, nie instaluj klastra (choć możliwe jest zignorowanie testów i użycie węzłów w klastrze). Przejrzyj raport. Zobaczysz ocenę poszczególnych komponentów konfiguracji, co pozwoli określić, czy używany sprzęt jest odpowiedni.



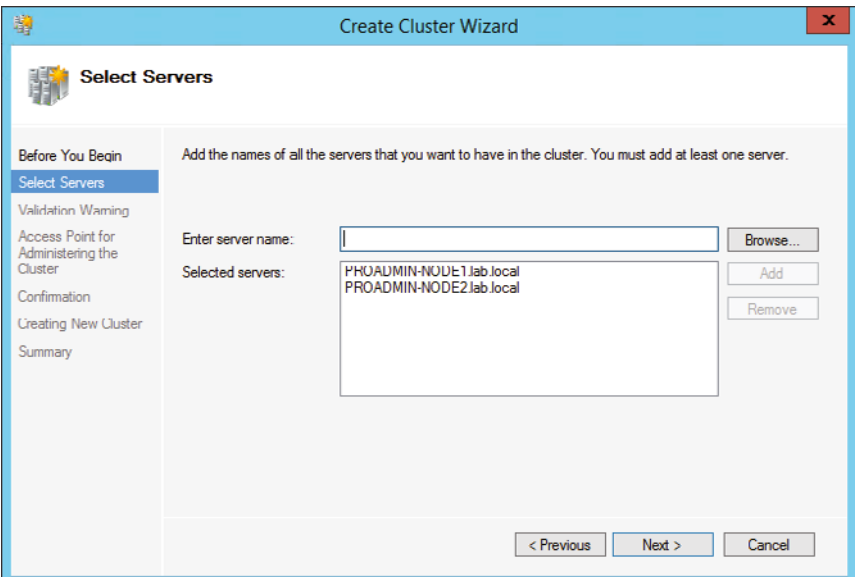
Rysunek 16.2. Wyniki testów

Aby wyświetlić raport, wystarczy kliknąć przycisk *View Report*. Zobaczysz wtedy szczegółowy raport w formacie HTML. Opisuje on wszystkie komponenty sprzętu wraz z oceną, czy nadają się do pracy w klastrze.

Instalowanie węzłów klastra

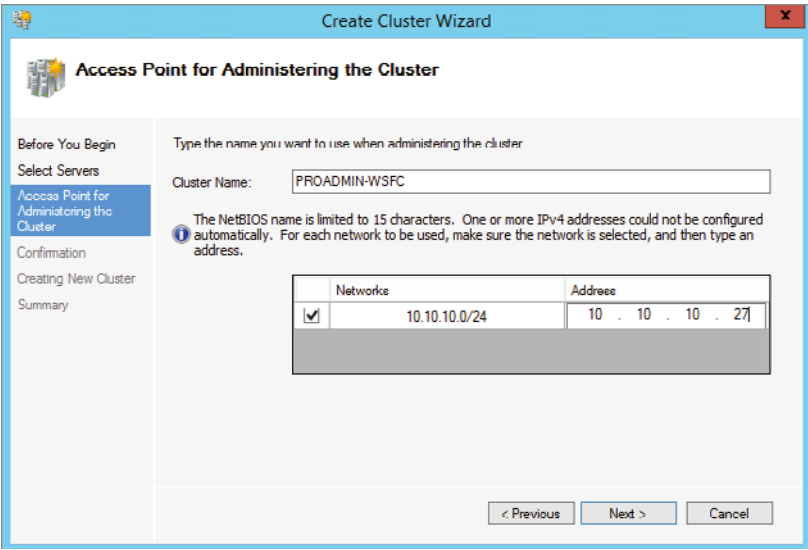
Klaster WSFC w systemie Windows 2012 R2 zostały znacznie uproszczone w porównaniu ze starszymi wersjami tego systemu. Teraz można dodać wszystkie węzły do klastra jednocześnie. Umożliwia to kreator, który wyświetla dwa pytania. Jeśli stosowałeś klaster WSFC ze starszych wersji systemu Windows, zdziwisz się, jak łatwe jest tworzenie klastrów w systemie Windows Server 2012 R2.

1. W narzędziu Failover Cluster Management wybierz opcję *Create a Cluster*. Pojawi się kreator *Create Cluster Wizard*. Na pierwszym ekranie po stronie powitalnej znajduje się pytanie o to, które fizyczne węzły mają znaleźć się w klastrze. Dodaj wszystkie fizyczne węzły i kliknij przycisk *Add*, co przedstawiono na rysunku 16.3. Jeśli będzie trzeba, w przyszłości będziesz mógł dodać więcej fizycznych węzłów.



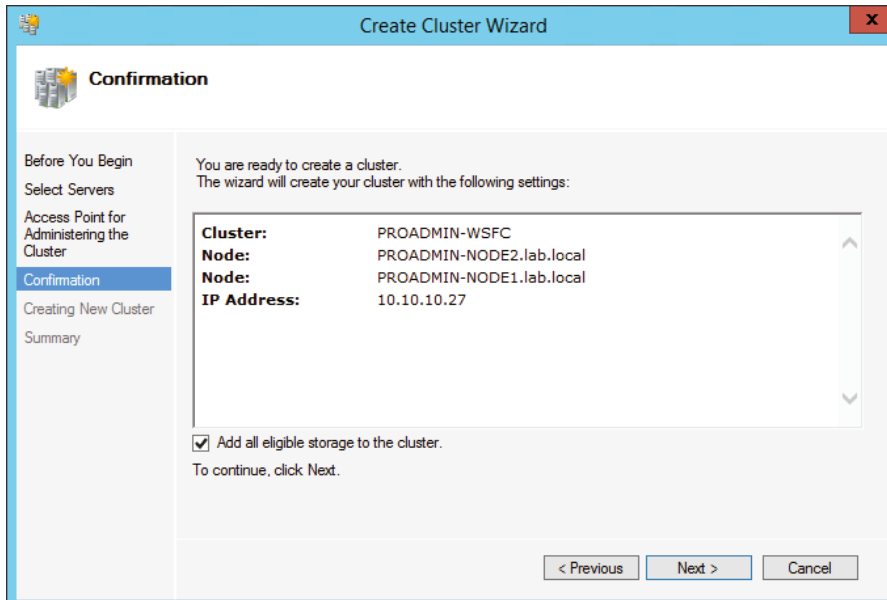
Rysunek 16.3. Dodawanie węzłów do klastra

2. Przypisz do klastra w systemie Windows adres IP i określ, której sieci klastr ma używać, co przedstawiono na rysunku 16.4. Na rysunku pokazano, że nazwą klastra w systemie Windows używaną do zarządzania nim jest PROADMIN-WSFC. Ponadto do klastra przypisywany jest adres IP; tu jest to 10.10.10.27.



Rysunek 16.4. Ustawianie nazwy używanej do zarządzania klastrem

Następnie kreator rozpoczyna tworzenie i konfigurowanie klastra (patrz rysunek 16.5). Gdy proces się zakończy, otrzymasz powiadomienie. Cały proces jest znacznie krótszy niż we wcześniejszych wersjach systemu Windows. W środowisku testowym zajmuje on mniej niż 2 minuty.



Rysunek 16.5. Uruchamianie tworzenia klastra

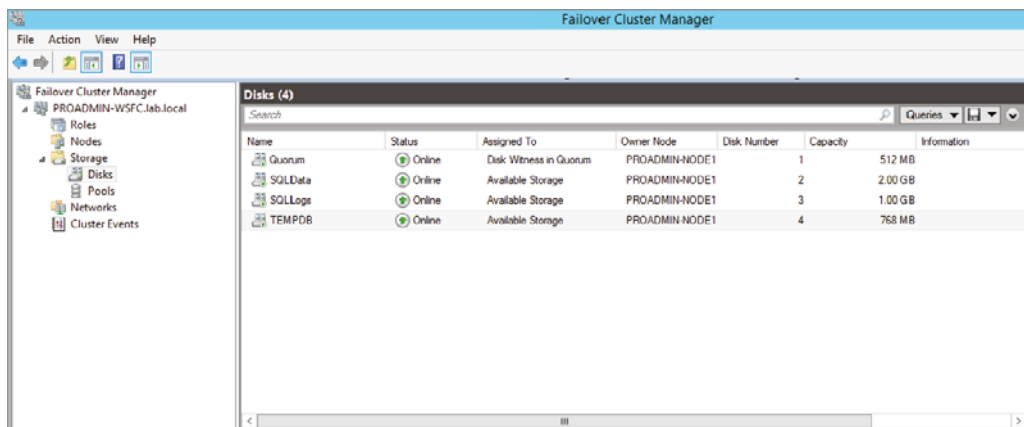
Przygotowywanie systemu Windows Server 2012 R2 do pracy w klastrze

Przed zainstalowaniem systemu SQL Server musisz wykonać prosty krok i przygotować klaster na dodanie tego systemu. W poprzednim punkcie dowiedziałeś się, jak utworzyć klaster w systemie Windows. Nie wskazałeś jednak temu systemowi dysków współużytkowanych. Wynika to z tego, że Microsoft stosuje minimalistyczne podejście do tworzenia klastrów. W trakcie budowania klastra trzeba ustawić tylko niezbędne elementy, a pozostałe potrzebne komponenty można dodać później. W końcowej części kreatora Create Cluster Wizard możesz dodać potrzebny magazyn pamięci, jednak nie jest to konieczne, by udało się utworzenie klastra. Jeśli nie określiłeś magazynu pamięci, wykonaj opisane poniżej kroki, aby dodać go do klastra.

Aby przygotować klaster do zainstalowania systemu SQL Server, wykonaj przedstawione poniżej operacje w narzędziu Failover Cluster Management.

1. W panelu nawigacji zaznacz węzeł *Storage/Disks*.
2. W panelu *Actions* wybierz opcję *Add Disk*.
3. Dodaj dyski, które mają być dostępne w klastrze WSFC.
4. Kliknij przycisk *OK*. Powinien pojawić się ekran podobny do tego z rysunku 16.6.

Jeśli nie widzisz współużytkowanych dysków, to albo zostały one już dodane jako zasób dyskowy, albo są niewidoczne w niektórych węzłach klastra. Może to oznaczać, że ustawiono niewłaściwe maski lub występują problemy z komunikacją. W nazwach dysków możesz określić kolejność ich dodawania, np.: „Dysk klastra 1”, „Dysk klastra 2” itd. Na tym etapie należy dodać do klastra wszystkie dostępne magazyny pamięci. Potem można przejść do instalowania systemu SQL Server.



Rysunek 16.6. Ekran z dostępnymi dyskami

Używanie koordynatora MSDTC w klastrze

Do koordynowania pracy systemu SQL Server w klastrze należy wykorzystać koordynator MSDTC (ang. *Microsoft Distributed Transaction Coordinator*), aby zapewnić wysoką dostępność tego systemu.

Windows 2003 obsługiwał tylko jeden koordynator MSDTC w klastrze. Dlatego wszystkie aplikacje z klastra musiały korzystać z tego koordynatora. Gdy jest on mocno obciążony, może spowalniać pracę systemu. W systemach Windows 2008 i nowszych w celu zapewnienia wyższej wydajności można zainstalować kilka egzemplarzy koordynatora MSDTC w jednym klastrze WSFC. Poniżej opisano, jak to zrobić. Pierwszy egzemplarz jest używany jako domyślny. Można go zmienić w konsoli zarządzania usługami składowymi (dcomcnfg).

1. Uruchom konsolę Usługi składowe z poziomu katalogu *Narzędzia administracyjne* lub za pomocą instrukcji dcomcnfg wywołanej w wierszu poleceń.
2. Rozwiń węzeł *Komputery*, a następnie kliknij prawym przyciskiem myszy węzeł *Mój komputer*.
3. Wybierz opcję *Właściwości*, otwórz zakładkę *MSDTC* i ustaw domyślny koordynator.

Jeśli istnieje kilka egzemplarzy koordynatora MSDTC, system SQL Server na podstawie podanych poniżej reguł określa, który z nich należy zastosować.

1. Egzemplarz koordynatora MSDTC zainstalowany w lokalnej grupie systemu SQL Server.
2. Egzemplarz koordynatora MSDTC ustawiony w odwzorowaniu. Aby utworzyć odwzorowanie, wywołaj poniższą instrukcję w wierszu poleceń:

```
msdct -tmMappingSet -name <NazwaOdwzorowania>
-service <NazwaUsługiSQLServer> -clusterResource <NazwaZasobuMSDTC>
```

Oto opis elementów z tego fragmentu kodu.

- <NazwaOdwzorowania> to dowolna nazwa identyfikująca odwzorowanie.
- <NazwaUsługiSQLServer> to nazwa usługi danego egzemplarza systemu SQL Server, np. MSSQLServer lub MSSQL\$<NazwaEgzemplarza>.
- <NazwaZasobuMSDTC> to nazwa koordynatora MSDTC z danego odwzorowania.

3. Domyślny egzemplarz koordynatora MSDTC z klastra.
4. Egzemplarz koordynatora MSDTC z węzła lokalnego.

System SQL Server automatycznie używa egzemplarza koordynatora MSDTC z grupy lokalnej (jeśli taki istnieje). W przeciwnym razie stosowany jest egzemplarz domyślny. Jeżeli egzemplarz z grupy lokalnej zawiedzie, trzeba nakazać systemowi SQL Server użycie innego egzemplarza. System nie stosuje automatycznie egzemplarza domyślnego. Aby utworzyć egzemplarz koordynatora MSDTC dla klastra, wykorzystaj narzędzie Failover Cluster Management i wykonaj następujące czynności.

1. Ustaw niewielki dysk współużytkowany do przechowywania pliku dziennika koordynatora MSDTC. Najlepiej umieścić taki plik na odrębnym dysku współużytkowanym lub w punkcie montowania, aby chronić go przed innymi usługami, które mogłyby uszkodzić dysk.
2. Jeśli koordynator MSDTC jest zainstalowany w grupie zasobów systemu SQL Server, może używać adresu IP i nazwy egzemplarza systemu SQL Server z tej grupy.
3. Domyślny egzemplarz koordynatora MSDTC należy zainstalować w odrębnej grupie zasobów klastra, aby uniknąć awaryjnego przełączania koordynatora, co z kolei prowadzi do problemów z innymi usługami z danej grupy zasobów. Gdy koordynator jest zainstalowany w odrębnej grupie zasobów, trzeba określić dla niego adres IP, nazwę sieciową i dysk współużytkowany.

Po utworzeniu koordynatorów MSDTC trzeba je podłączyć do sieci, aby umożliwić im dostęp do zasobów sieciowych, a także by pozwolić aplikacjom na dostęp do koordynatorów. W tym celu wykonaj następujące kroki.

1. Uruchom konsolę Usługi składowe z poziomu katalogu *Narzędzia administracyjne* lub za pomocą instrukcji `dcomcnfg` wywołanej w wierszu poleceń.
2. Rozwiń węzeł *Komputery*. Rozwiń węzeł *Koordynator transakcji rozproszonych*.
3. Kliknij prawym przyciskiem myszy właściwości danego egzemplarza koordynatora MSDTC.
4. W zakładce *Zabezpieczenia* zaznacz pola *Dostęp do usługi DTC przez sieć*, *Zezwalaj na przychodzące* i *Zezwalaj na wychodzące*, a następnie kliknij przycisk OK.

Dodawanie systemu SQL Server 2014 do klastra

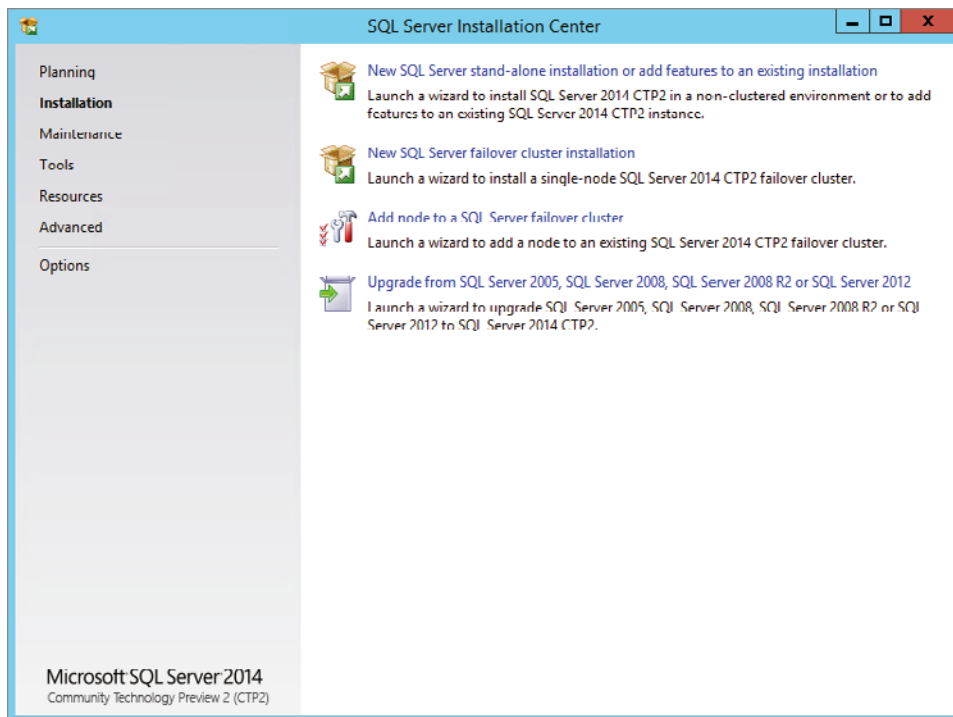
Instalowanie systemu SQL Server w klastrze WSFC to jeden z najprostszych kroków budowania klastra z tym systemem. Wystarczy zastosować program instalacyjny systemu SQL Server 2014, który wykonuje wszystkie skomplikowane operacje. Ty musisz tylko podjąć kilka ważnych decyzji, po czym możesz się rozluźnić i obserwować przebieg instalacji. Program instalacyjny sprawdza nawet to, czy wszystkie węzły są prawidłowo skonfigurowane, a jeśli nie są, przed rozpoczęciem instalacji proponuje rozwiązania. Pliki binarne systemu SQL Server 2014 są instalowane na dysku lokalnym każdego węzła, a systemowe bazy danych są zapisywane we wskazanej współużytkowanej macierzy dyskowej.

W następnym punkcie opisujemy krok po kroku, jak zainstalować system SQL Server 2014 w klastrze WSFC. Instrukcje dotyczą klastra typu aktywny/pasywny o dwóch węzłach. Jednak nawet wtedy, gdy budujesz klaster typu aktywny/aktywny o dwóch węzłach lub klaster z wieloma węzłami, możesz zastosować opisane tu kroki. Jedyną różnicą polega na tym, że trzeba zainstalować system SQL Server 2014 na wszystkich docelowych węzłach i ustawić dla każdego egzemplarza tego systemu inny dysk logiczny we współużytkowanej macierzy dyskowej.

Instrukcje instalowania systemu SQL Server w klastrze

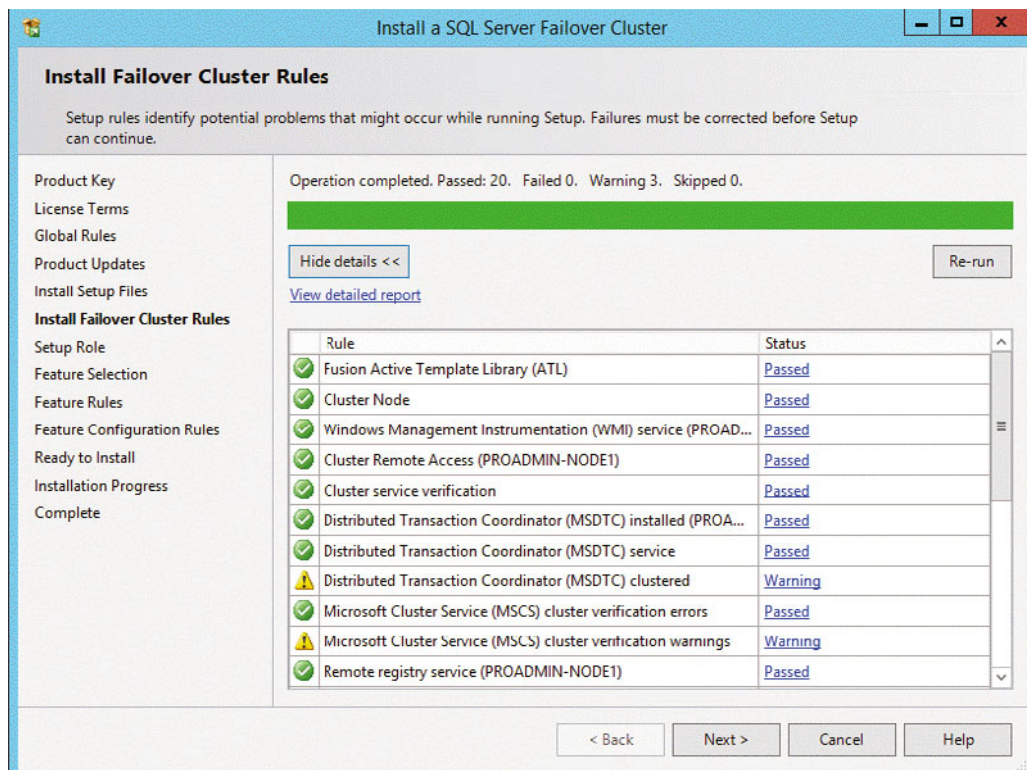
Aby rozpocząć instalowanie w klastrze systemu SQL Server, potrzebujesz nośnika instalacyjnego (płyty DVD lub obrazu ISO). Możesz zainstalować system bezpośrednio z nośnika lub przenieść pliki na aktywny węzeł klastra i uruchomić program instalacyjny z dysku. Proces instalacji systemu SQL Server przebiega tu tak samo jak w standardowej instalacji bez klastra (jej omówienie znajdziesz w rozdziale 2., „Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014”), dlatego w poniższych krokach podkreślamy różnice w stosunku do typowej instalacji.

1. Aby rozpocząć instalację, uruchom plik *Setup.exe*. Najpierw instalowane są wymagane komponenty, które opisano w rozdziale 2. Warto dodać te komponenty do wszystkich węzłów przed rozpoczęciem instalowania systemu w klastrze. Pozwala to przyspieszyć późniejszą pracę i umożliwia wizualne debugowanie rozwiązania, jeśli instalacja się nie powiedzie.
2. W oknie *SQL Server Installation Center* wybierz opcję *New SQL Server Failover Cluster Installation*. Na tym ekranie możesz też dodać nowe węzły do klastra po zakończeniu instalacji (patrz rysunek 16.7).



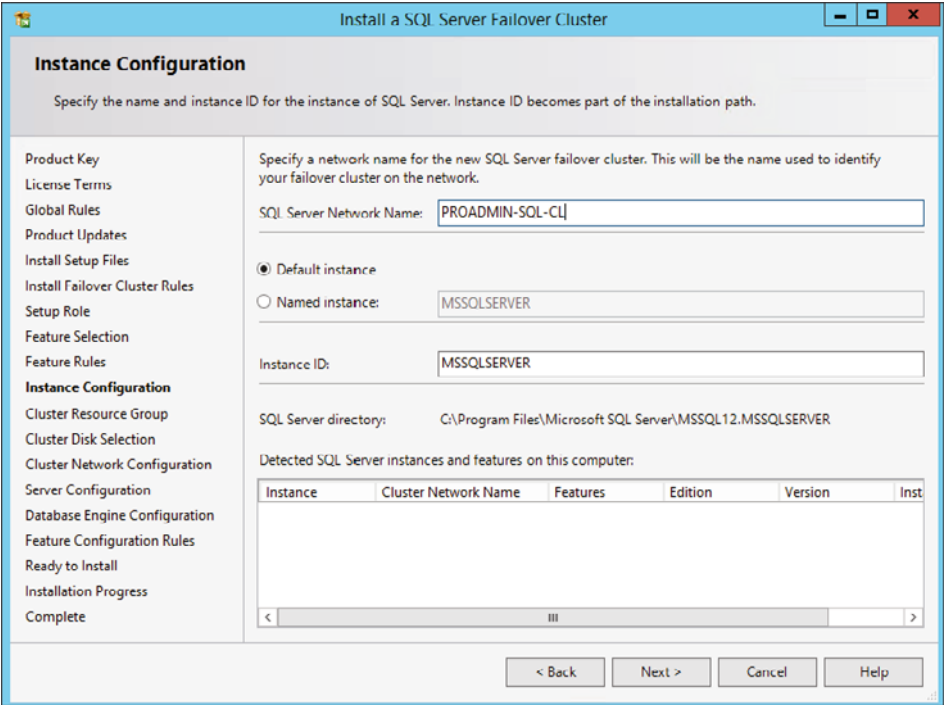
Rysunek 16.7. Uruchamianie instalacji systemu SQL Server w klastrze

Większość procesu instalowania systemu SQL Server w klastrze przebiega tak samo jak podczas instalowania niezależnego egzemplarza. Wyjątkiem jest kilka ekranów. Gdy dojdiesz do ekranu *Setup Support Rules*, zaznaczonych będzie wiele nowych reguł, co przedstawiono na rysunku 16.8. Przykładowo sprawdzane jest, czy w klastrze działa koordynator MSDTC. Choć system SQL Server może działać bez takiego koordynatora w klastrze, niektóre funkcje nie będą wtedy dostępne (np. transakcje rozproszone dla serwerów połączonych lub usługi SSIS).

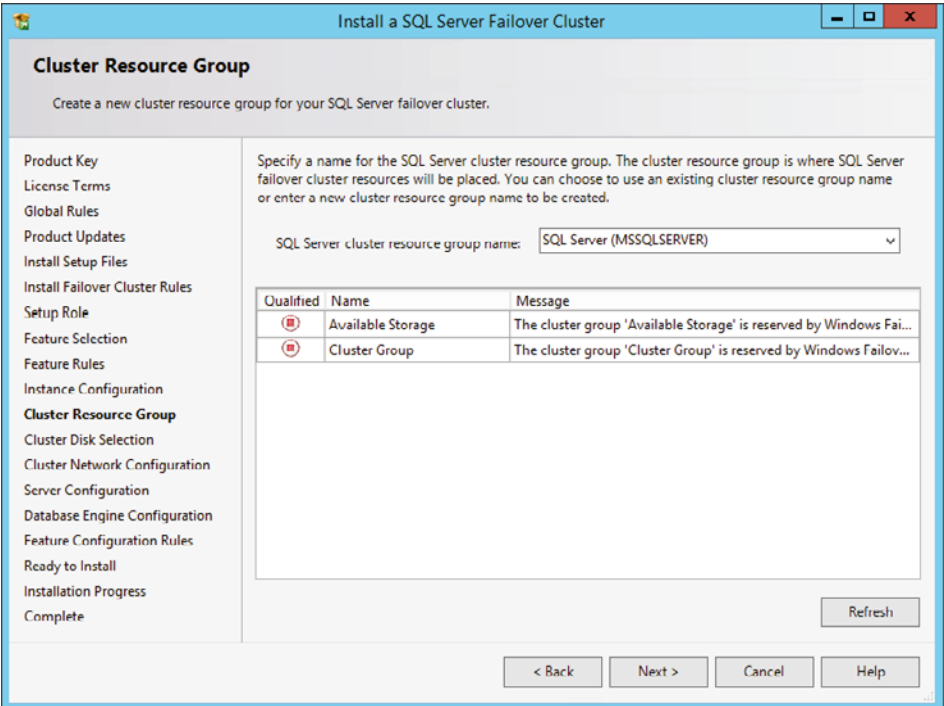


Rysunek 16.8. Przy instalowaniu systemu SQL Server w klastrze zaznaczone są dodatkowe reguły

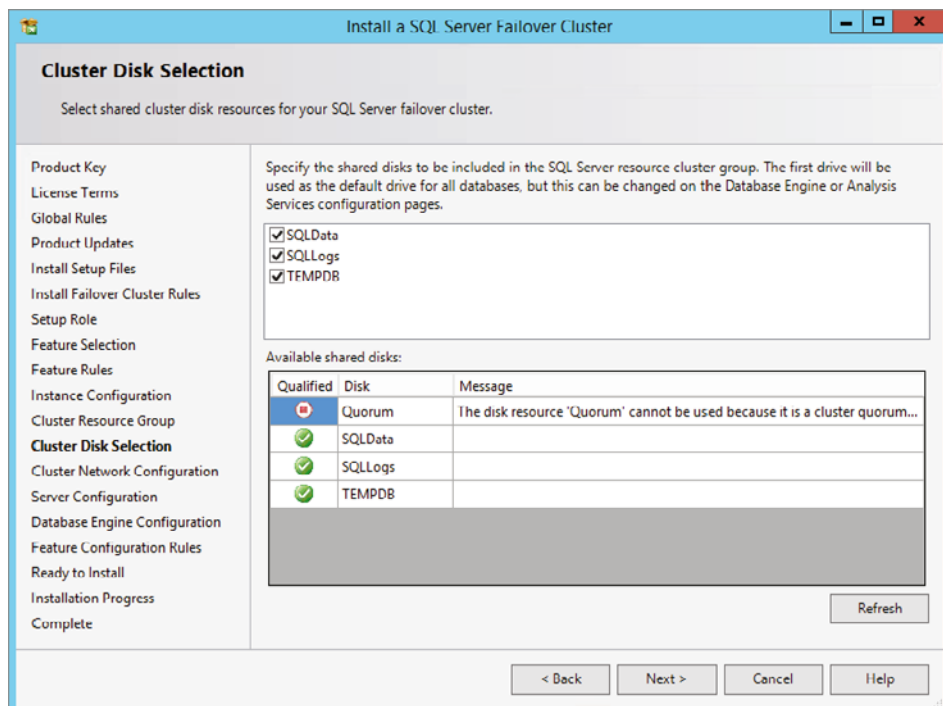
3. W oknie dialogowym *Instance Configuration* ustaw nazwę sieciową dla pracującego w klastrze egzemplarza systemu SQL Server, co przedstawiono na rysunku 16.9. Tej nazwy będą używać wszystkie aplikacje klienckie w łańcuchu znaków połączenia do łączenia się z systemem. Ponadto nazwa katalogu, do którego kopiowane są pliki programu, jest domyślnie generowana na podstawie ustawionej tu nazwy. Kliknij przycisk *Next*.
4. W oknie *Cluster Resource Group* ustaw grupę zasobów klastra, w której znajdą się wszystkie zasoby systemu SQL Server (np. nazwa sieciowa, adres IP i usługi systemu SQL Server). Przedstawiono to na rysunku 16.10. Kliknij przycisk *Next*.
5. Od wersji SQL Server 2008 można w trakcie instalacji ustawić kilka dysków klastra dostępnych dla tego systemu, co jest przydatnym udogodnieniem. W wersji SQL Server 2005 dyski trzeba było dodawać po zakończeniu instalacji. Ponadto nie było możliwe tworzenie kopii zapasowej poza dyskiem głównym systemu SQL Server, jeśli wcześniej nie ustawiono zależności. W systemie SQL Server 2014 wystarczy zaznaczyć, które zasoby dyskowe z klastra mają być dostępne dla systemu. Przedstawiono to na rysunku 16.11. Kliknij przycisk *Next*.
6. Po skonfigurowaniu dysków określ na stronie *Cluster Network Configuration* sieć, której system SQL Server ma używać do komunikowania się z aplikacjami klienckimi. Ponadto ustaw adres IP dla każdej sieci. Ten adres musisz wcześniej uzyskać od administratora sieci. Na rysunku 16.12 widać, że w systemie Windows 2012 R2 można używać protokołu DHCP. Kliknij przycisk *Next*.



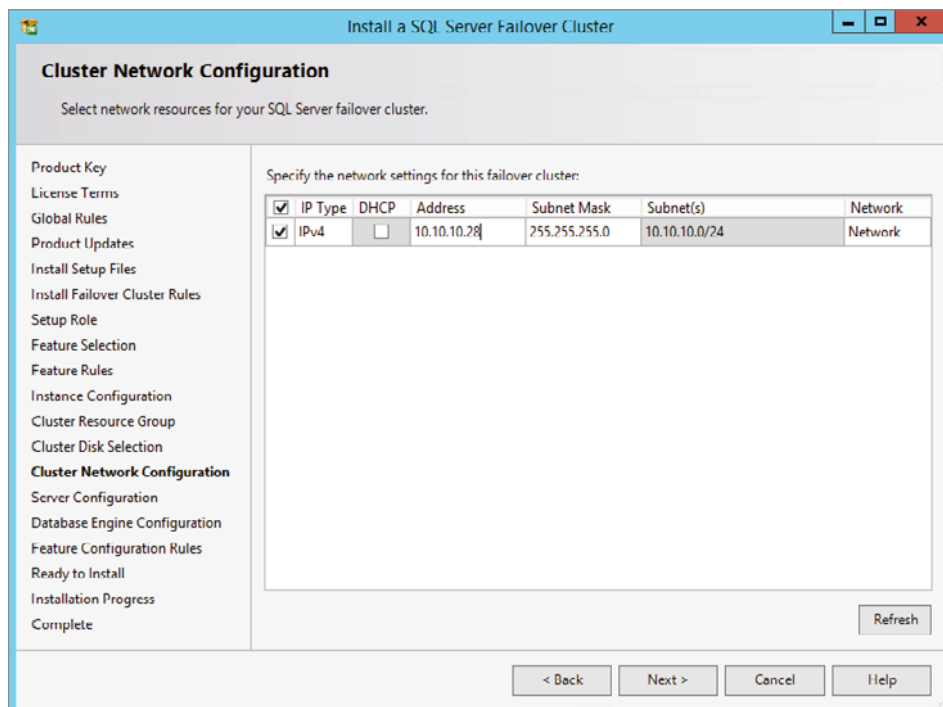
Rysunek 16.9. Ustawianie nazwy egzemplarza systemu SQL Server z klastra



Rysunek 16.10. Tworzenie grupy zasobów klastra dla systemu SQL Server

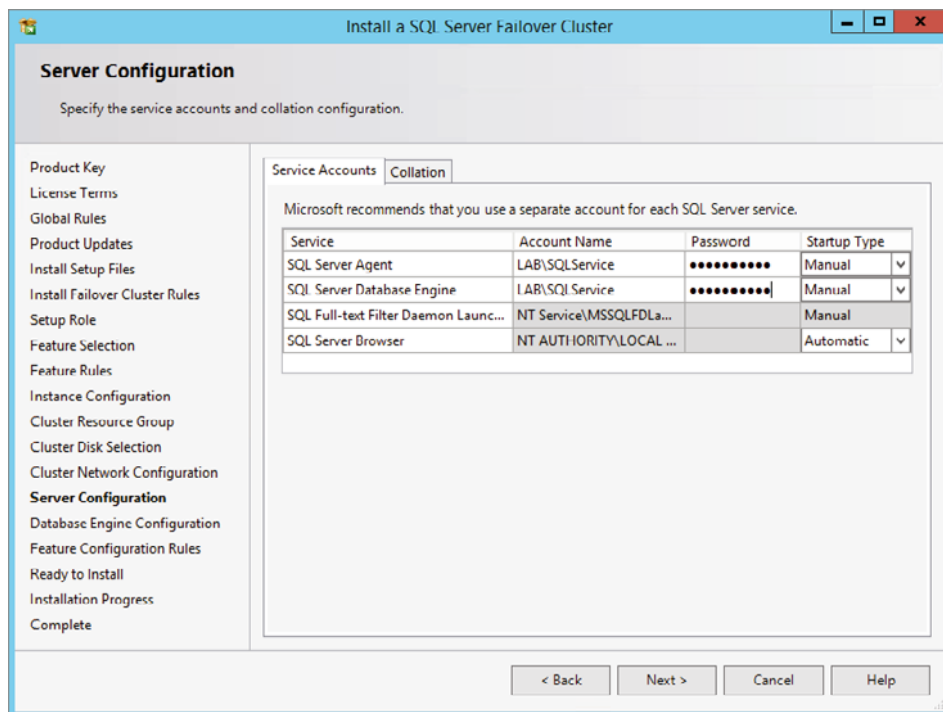


Rysunek 16.11. Ustawianie dysków klastra dostępnych dla systemu SQL Server



Rysunek 16.12. Konfigurowanie sieci

7. Na stronie *Server Configuration* ustaw konta usługowe i kolację, co pokazano na rysunku 16.13.



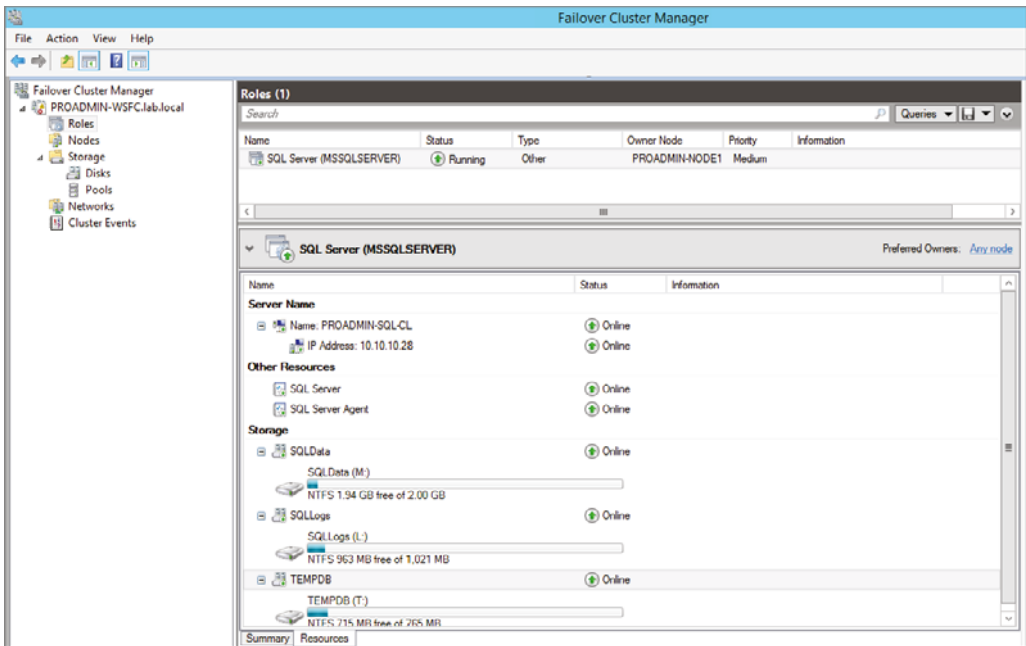
Rysunek 16.13. Konfigurowanie innych opcji serwera

Kontynuuj instalację. Pliki zostaną skopiowane do węzła klastra i zainstalowane. Ponadto wszystkie zasoby klastra zostaną utworzone i skonfigurowane. Po zakończeniu instalacji w danym węźle klastra przejdź do następnego węzła. Wybierz w nim opcję *Add Node*, aby dodać nowy węzeł do pracującego w klastrze egzemplarza systemu SQL Server. W klastrach z systemem SQL Server 2005 program instalacyjny kopiował pliki binarne tego systemu do wszystkich węzłów klastra. Od wersji SQL Server 2008 program instalacyjny dodaje do klastra węzeł, w którym jest wykonywany.

Aby dodać nowy węzeł w systemach SQL Server 2008 i nowszych, wykonaj następujące kroki.

1. Zaloguj się w węźle, który chcesz dodać do klastra.
2. Uruchom program instalacyjny w tym węźle.
3. Wybierz opcję *Add Node to a SQL Server Failover Cluster*.
4. W trakcie instalacji pojawi się pytanie o nazwę klastra, do którego chcesz dodać dany węzeł. Podaj odpowiednią nazwę.
5. Program instalacyjny dołączy dany węzeł do klastra. Jeśli chcesz dodać do klastra następne węzły, wykonaj te kroki dla każdego z nich.

Po zakończeniu instalowania klastra otwórz narzędzie Windows Failover Cluster Manager. Zainstalowany klaster z dwoma węzłami powinien wyglądać podobnie jak na rysunku 16.14.



Rysunek 16.14. Klastrer z dwoma węzłami

UWAGA Silnik relacyjny systemu SQL Server i usługi Analysis Services są w pełni zintegrowane z klastrami WSFC. Usługi SSIS mogą działać w klastrze tylko jako ogólny zasób, a usługi Reporting Services nie pracują w klastrach. Można wprowadzić przechowywać bazy danych usług Reporting Services w systemie działającym w klastrze, ale nie mogą one pracować jako fronton.

Po dodaniu do klastra wszystkich egzemplarzy systemu SQL Server grupa zasobów powinna obejmować następujące elementy:

- nazwę sieciową,
- adres IP,
- jeden lub więcej dysków współużytkowanych,
- usługę silnika bazodanowego systemu SQL Server,
- usługę SQL Server Agent,
- usługę SQL Server Analysis Services (jeśli jest zainstalowana); zgodnie z najlepszymi praktykami należy instalować ją w odrębnej grupie zasobów, aby uniknąć problemów związanych ze wzajemnym wpływem zasobów na siebie,
- jeden udział plikowy, jeśli zainstalowana jest funkcja FILESTREAM.

Po zakończeniu instalowania systemu SQL Server w klastrze zbadaj zależności zasobów, aby ustalić, których zasobów z grupy wymaga do działania dany zasób. Przykładowo usługa SQL Server Agent wymaga, aby dostępny był system SQL Server, a system SQL Server jest zależny od współużytkowanych dysków, nazwy sieciowej i adresu IP. Kompletny klastrer WSFC powinien wyglądać tak, jak na rysunku 16.14.

Instalowanie pakietów poprawek SP i aktualizacji typu Cumulative Update

Po zainstalowaniu systemu SQL Server i dodaniu go do klastra następny krok polega na zainstalowaniu dostępnych pakietów Service Pack (SP), które można pobrać za pomocą usługi Windows Update. Możesz przeprowadzić aktualizację bez zatrzymywania środowiska, co opisano w punkcie „Aktualizowanie klastra do wersji SQL Server 2014 w miejscu”. Instalowanie pakietów SP i aktualizacji typu Cumulative Update jest stosunkowo proste, ponieważ są one zgodne z klastrami.

Testuj, testuj i jeszcze raz testuj

Po dodaniu systemu SQL Server w węzłach do klastra musisz starannie przetestować instalację (tak jak wcześniej po zainstalowaniu klastra WSFC z systemem Windows 2012 R2). Poszukaj m.in. komunikatów o błędach w przeglądarce zdarzeń systemu Windows i sprawdź, czy możesz przełączyć awaryjnie system SQL Server z jednego węzła na drugi i z powrotem. Nie ograniczaj się do przetestowania klastra z systemem SQL Server. Sprawdź też, jak aplikacje klienckie reagują na przełączenie awaryjne systemu. Kolejne podpunkty poświęcone testom są podobne do wcześniejszych, jednak zostały zmodyfikowane o bardziej skomplikowane aspekty związane z niezbędnymi testami aplikacji klienckich.

Poniżej opisujemy serię testów, które możesz przeprowadzić, aby sprawdzić, czy klastrer z systemem SQL Server 2014 i jego aplikacje klienckie działają prawidłowo przy przełączaniu awaryjnym węzłów. Po wykonaniu każdego testu sprawdź, czy efekty są zgodne z oczekiwaniami (czyli czy przełączenie zakończyło się sukcesem). Zbadaj też pliki dzienników systemu Windows pod kątem problemów. Jeśli w trakcie któregoś testu wykryjesz usterki, rozwiąż je przed przejściem do dalszych testów.

Przygotowania do testów

Podobnie jak w testach klastra omówionych wcześniej, wybierz stację roboczą z narzędziem Failover Cluster Management, którą wykorzystasz do interakcji z klastrem.

W ramach przygotowań musisz przetestować każdą aplikację kliencką, w której będzie używany system SQL Server z klastra. Pozwoli Ci to ocenić przebieg przełączania awaryjnego. Niektóre aplikacje klienckie w momencie przełączania ponownie nawiązują połączenie. Inne nie mają wbudowanego mechanizmu ponownego nawiązywania połączenia i przestają działać. Powinieneś na początku ustalić sposób działania każdej aplikacji klienckiej.

W tym celu zacznij od zrobienia listy wszystkich aplikacji klienckich. Mogą ich być dziesiątki. Każdą z nich trzeba zrekonfigurować, aby używała nazwy sieciowej i adresu IP nowego egzemplarza systemu SQL Server z klastra. Ponadto aby aplikacje klienckie działały poprawnie, musisz przywrócić lub zainstalować odpowiednie bazy danych w nowym klastrze. Jest to konieczne, jeśli chcesz utworzyć wysoce dostępny klastrer.

Gdy każda aplikacja kliencka z powodzeniem nawiązała połączenie z systemem SQL Server, możesz przystąpić do testów. Pamiętaj, że sprawdzasz wiele aspektów, w tym klastrer w systemie Windows 2012 R2, system SQL Server pracujący w klastrze i aplikacje klienckie.

Przenoszenie grup zasobów między węzłami

Najłatwiejszym testem jest użycie narzędzia Failover Cluster Management i ręczne przeniesienie grupy zasobów systemu SQL Server z węzła aktywnego do pasywnego i z powrotem. W tym celu wykonaj następujące kroki.

1. Przejdź do grupy zasobów dla systemu SQL Server. Kliknij ją prawym przyciskiem myszy i wybierz opcję *Move this service or application to another node*. Określ, gdzie chcesz przenieść daną grupę zasobów. To zainicjuje przenoszenie grupy z węzła aktywnego do wskazanego węzła pasywnego.
2. Następnie sprawdź narzędzie Failover Cluster Management i każdą z aplikacji klienckich. Wszystkie elementy powinny działać tak, jakby nie nastąpiło przełączenie awaryjne. Narzędzie Failover Cluster Management prawdopodobnie łatwo przejdzie ten test. Z aplikacjami klienckimi może być inaczej. Musisz sprawdzić, czy każda z nich ponownie nawiązała połączenie. Jeśli tak się nie stało, ustal przyczynę; nie zawsze jest to łatwe. W większości aplikacji klienckich przyczyną zaprzestania działania po przełączeniu awaryjnym jest brak mechanizmu ponownego nawiązywania połączenia. Może być tak, że mogą one nawiązać połączenie dopiero po zamknięciu i ponownym uruchomieniu.
3. Jeśli grupę zasobów z powodzeniem przeniesiono z węzła aktywnego do pasywnego, za pomocą tej samej procedury przenieś grupę ponownie do pierwotnego węzła. Tak jak wcześniej, sprawdź następnie narzędzie Failover Cluster Management i aplikacje klienckie oraz przejrzyj dzienniki zdarzeń pod kątem problemów. Jeśli w klastrze WSFC lub systemie SQL Server wykryjesz usterki spowodowane testem przełączania awaryjnego, musisz je rozwiązać przed przejściem do dalszych etapów. Jeżeli problem dotyczy aplikacji klienckiej, możesz kontynuować testy i spróbować naprawić usterkę później. Zwykle jest tak, że aplikacje klienckie, które nie radzą sobie z pierwszym testem, mają problemy także we wszystkich późniejszych próbach.

Ręczne przełączanie awaryjne węzłów przez ich wyłączenie

Aby sprawdzić, czy następuje przełączanie awaryjne węzłów i czy każdy węzeł może przejąć kontrolę nad systemem SQL Server, możesz ręcznie wyłączać węzły.

1. Wyłącz aktywny węzeł. Następnie obserwuj przebieg przełączania awaryjnego w narzędziu Failover Cluster Management i aplikacjach klienckich. Tak jak wcześniej, zwróć uwagę na problemy.
2. Następnie włącz wyłączony wcześniej węzeł i poczekaj, aż wszystkie komponenty wznowią pracę. Wtedy wyłącz obecnie aktywny węzeł. Ponownie przyjrzyj się przebiegowi przełączania awaryjnego w narzędziu Failover Cluster Management i aplikacjach klienckich oraz zwróć uwagę na problemy.
3. Po zakończeniu przełączania ponownie włącz wyłączony węzeł.

Ręczne przełączanie awaryjne węzłów przez zerwanie połączenia z siecią publiczną

Możesz ręcznie przełączyć węzły, zrywając połączenie z siecią publiczną.

1. Wyłącz połączenie z siecią publiczną w aktywnym węźle. Spowoduje to przełączenie awaryjne, które możesz obserwować w narzędziu Failover Cluster Management i aplikacjach klienckich. Zwróć uwagę na problemy.
2. Następnie ponownie włącz połączenie z siecią publiczną i odłącz od tej sieci obecnie aktywny węzeł. To spowoduje przełączenie awaryjne do obecnie pasywnego węzła. Obserwuj przebieg tego procesu w narzędziu Failover Cluster Management i aplikacjach klienckich. Zwróć uwagę na problemy.
3. Po zakończeniu testu ponownie włącz połączenie w węźle.

Ręczne przełączanie awaryjne węzłów przez zerwanie połączenia z macierzą współużytkowaną

Trzeci test ręcznego przełączania awaryjnego możesz przeprowadzić przez zerwanie połączenia z macierzą współużytkowaną.

1. Zerwij połączenie ze współużytkowaną macierzą dyskową w aktywnym węźle. Spowoduje to przełączanie awaryjne, które możesz obserwować w narzędziu Failover Cluster Management i aplikacjach klienckich. Zwróć uwagę na problemy.
2. Następnie ponownie włącz połączenie w obecnie pasywnym węźle i odłącz od macierzy aktualnie aktywny węzeł. Obserwuj przebieg przełączania awaryjnego w narzędziu Failover Cluster Management i aplikacjach klienckich. Zwróć uwagę na problemy.
3. Po zakończeniu testu ponownie nawiąż połączenie.

UWAGA Jeśli w którymś z testów natrafisz na problemy, rozwiąż je przed przejściem do dalszych zadań.

Zarządzanie klastrem i monitorowanie go

Po uruchomieniu i przetestowaniu systemu SQL Server w klastrze możesz włączyć go w środowisku produkcyjnym. Może to wymagać utworzenia nowych baz danych, przeniesienia baz ze starszych serwerów do klastra, skonfigurowania zadań systemu SQL Server itd. W większości sytuacji zarządzanie systemem SQL Server w klastrze przebiega tak samo jak zarządzanie niezależnym egzemplarzem tego systemu.

Należy jednak pamiętać, że przy dostępie do klastra z poziomu narzędzi administracyjnych systemu SQL Server 2014 (takich jak SQL Server Management Studio) trzeba podać nazwę sieciową i adres IP egzemplarza pracującego w klastrze. Gdy natomiast korzystasz z narzędzi systemu operacyjnego (np. monitora wydajności), musisz podać nazwę sieciową lub adres IP monitorowanego węzła (zwykle jest to węzeł aktywny).

Jeśli jesteś administratorem baz danych, prawdopodobnie zarządzasz systemem SQL Server 2014 za pomocą programu SQL Server Management Studio. Czasem jednak trzeba uzyskać dostęp do poszczególnych węzłów klastra. Gdy masz łatwy dostęp do klastra, zawsze możesz zalogować się w węzeł lokalnie. Jeżeli wolisz zdalnie zarządzać systemem, możesz uzyskać dostęp do poszczególnych węzłów za pomocą pulpitu zdalnego i usług terminalowych.

Gdy administrator zaczyna zarządzać pierwszym klastrem WSFC, może się zastanawiać, gdzie działa system SQL Server. Pamiętaj, że klaster z systemem SQL Server obejmuje węzeł aktywny, gdzie ten system jest uruchomiony, i węzeł pasywny, na którym ten system nie pracuje. Egzemplarz systemu SQL Server zawsze działa tylko w aktywnym węźle. Gdy zatem bezpośrednio sprawdzasz węzły, zwykle interesuje Cię węzeł aktywny. Jeśli nie wiesz, który węzeł pełni tę funkcję, możesz to ustalić za pomocą narzędzia Failover Cluster Management.

Gdy zalogujesz się w aktywnym węźle (lub połączysz z nim za pomocą zdalnego pulpitu) i otworzysz program Eksplorator Windows (jest to rutynowa czynność), uzyskasz dostęp do współużytkowanych dysków z danymi systemu SQL Server. Jeśli jednak zalogujesz się w węzeł pasywnym, dyski będą niewidoczne. Powodem jest to, że w danym momencie dostęp do dysków jest możliwy tylko z jednego węzła z systemem SQL Server.

Jeśli używasz klastra za pomocą zdalnego pulpitu i usług terminalowych, pamiętaj o kilku nietypowych zjawiskach. Gdy np. korzystasz ze zdalnego pulpitu i łączysz się z klastrem za pomocą nazwy sieciowej i adresu IP systemu SQL Server, nawiążesz połączenie z aktywnym węzłem, tak jakbyś łączył się bezpośrednio z tym węzłem przy użyciu jego nazwy sieciowej i adresu IP. Jeżeli jednak po przełączeniu awaryjnym uzyskasz w zdalnym pulpicie dostęp do klastra przy użyciu nazwy sieciowej i adresu IP systemu SQL Server, mogą wystąpić problemy, zwłaszcza jeśli korzystasz z programu Eksplorator plików z systemu Windows. Przykładowo dyski z danymi mogą wyglądać na niedostępne (choć ostatecznie uzyskasz do nich dostęp). Aby rozwiązać ten problem, po przełączeniu awaryjnym węzłów wyloguj się ze zdalnego pulpitu i ponownie go włącz.

Rozwiązywanie problemów z klastrem

Rozwiązywanie problemów związanych z klastrami wymaga odporności, cierpliwości i doświadczenia, a także umowy z działem pomocy technicznej Microsoftu. Dzieje się tak, ponieważ klastrowanie bywa skomplikowane i wymaga znajomości sprzętu, współużytkowanych macierzy dyskowych, sterowników sprzętu, systemu operacyjnego, usług klastrowania i systemu SQL Server. Przyczyną problemów może być każdy z tych czynników, a precyzyjne ustalenie źródła usterki jest często trudne.

Inny czynnik, sprawiający, że rozwiązywanie problemów z klastrami jest trudne, to mało zrozumiałe, a czasem w ogóle niedostępne informacje zwrotne w postaci komunikatów lub wpisów w dzienniku. A gdy już uzyskasz takie informacje, trudno na ich podstawie zidentyfikować i rozwiązać usterkę.

Jeśli używasz klastra WSFC, pomyśl o wykupieniu pomocy technicznej w Microsoftzie. To dobra inwestycja, która z pewnością się spłaci. Wielokrotnie korzystaliśmy z pomocy technicznej Microsoftu i w większości sytuacji otrzymaliśmy przydatne wskazówki. Nie musisz automatycznie dzwonić do pracowników pomocy technicznej, gdy tylko wykryjesz problemy. Zawsze staraj się samodzielnie zidentyfikować źródło usterki i ją rozwiązać. Jednak czasem, zwłaszcza gdy klaster nie działa i potrzebujesz pomocy w jego ponownym uruchomieniu, musisz pogodzić się z tym, że samodzielnie nie naprawisz problemu i potrzebujesz konsultacji.

W następnym punkcie przedstawiamy ogólne wskazówki, które pomogą Ci zacząć pracę nad identyfikowaniem i rozwiązywaniem problemów dotyczących klastrów.

Jak podejść do rozwiązywania problemów z klastrami WSFC?

W tym rozdziale w omówieniu instalowania klastrów podkreślono znaczenie przeprowadzania testów po każdym wykonanym zadaniu. Do następnego kroku można przejść tylko wtedy, jeśli wszystko działa poprawnie. To metodyczne podejście pomaga zidentyfikować źródło problemów bezpośrednio po ich wystąpieniu.

Jeśli np. wszystko działa poprawnie, a po wykonaniu określonej czynności testy kończą się niepowodzeniem, można przyjąć, że wykonane operacje bezpośrednio lub pośrednio spowodowały problem. Dzięki temu identyfikacja źródła usterki jest prostsza. Jeśli nie przeprowadzasz regularnych testów i wykryjesz problem dopiero po wykonaniu wielu zadań, zidentyfikowanie jego źródła będzie znacznie trudniejsze.

Najlepszym sposobem rozwiązywania problemów jest częste wykonywanie testów. Pomocne jest też opracowanie szczegółowego planu instalacji, który pozwala się upewnić, że wykonano wszystkie potrzebne kroki (w tym testy w odpowiednich momentach).

Właściwe rozwiązanie już za pierwszym razem

Możesz uniknąć wielu kłopotów z rozwiązywaniem problemów, jeśli uda Ci się im zapobiec. Oto opis, jak to zrobić.

- Upewnij się, że cały sprzęt z fizycznych węzłów i współużytkowanej macierzy dyskowej przechodzi testy poprawności klastra.
- Upewnij się, że używasz najnowszych sterowników dla sprzętu i pakietów poprawek dla oprogramowania.
- Przygotuj szczegółowy plan instalacji, który będzie pełnił funkcję przewodnika w trakcie instalacji i — jeśli trzeba — przywracania stanu po awarii.
- Jak najlepiej poznaj klastry WSFC, zanim zaczniesz instalację. Przyczyną wielu problemów z klastrami są ludzie, ponieważ wykonują zadania metodą prób i błędów, zamiast dokładnie ustalić, co należy zrobić.
- Przygotuj opis procedur dla klastra WSFC. Uwzględnij w nich stan, konfigurację i instrukcje dotyczące działania klastra, aby umożliwić zbudowanie spójnego i stabilnego systemu.

Zbieranie informacji

Gdy chcesz zidentyfikować problem, często potrzebnych jest dużo informacji. Niestety, możliwe, że potrzebne dane nie istnieją, znajdują się w wielu różnych miejscach lub są mylące. Aby jednak rozwiązać problem, należy zebrać jak najwięcej informacji.

Gdy chcesz poradzić sobie z tymi trudnościami, wykorzystaj poniższe wskazówki i zasoby przy zbieraniu informacji potrzebnych do rozwiązywania problemów z klastrami.

- **Ustal, jakich efektów oczekujesz.** Jeśli spodziewasz się konkretnych wyników i ich nie uzyskujesz, dokładnie określ, co się powinno stać i co naprawdę się przydarzyło. Musisz ustalić, czym różnią się oczekiwania od rzeczywistości.
- **Ustal, co stało się bezpośrednio przed wystąpieniem problemu.** Jest to znacznie łatwiejsze, gdy przeprowadzasz testy po każdej zmianie, co opisano wcześniej.
- **Ustal, czy problem jest powtarzalny.** Nie wszystkie problemy można łatwo odtworzyć, jeśli jednak Ci się to uda, możesz uzyskać przydatne informacje.
- **Zapisz wszystkie komunikaty o błędach pojawiające się na ekranie.** Koniecznie rób zrzuty ekranu z wszystkimi komunikatami. Niektórzy administratorzy baz danych automatycznie klikają przycisk OK po zobaczeniu komunikatów o błędach, nie rejestrując ich treści. Często dokładny tekst komunikatu jest pomocny, jeśli chcesz dowiedzieć się czegoś na temat danego błędu w internecie.
- **Przejrzyj dzienniki.** W zależności od miejsca w procesie konfigurowania klastra możesz przejrzeć różne dzienniki. Są to m.in. trzy dzienniki systemu operacyjnego: dziennik klastra (`c:\windows\cluster\cluster.log`), dziennik instalacji systemu SQL Server 2014 (`%ProgramFiles%\Microsoft SQL Server\120\Setup Bootstrap\LOG\Summary.txt`) i pliki dziennika systemu SQL Server 2014. W dziennikach mogą znajdować się rozmaite komunikaty o błędach. Po wykryciu błędu możesz poszukać w internecie informacji o tym, czy ktoś już na niego natrafił, a także wskazówek dotyczących rozwiązania problemu.
- **Poszukaj w internecie informacji o danym komunikacie o błędzie.** Jeśli wyświetlane komunikaty o błędach nie są oczywiste (a czy kiedykolwiek są?), poszukaj ich w internecie, w tym na grupach dyskusyjnych.

Im więcej informacji na temat problemu uzyskasz, tym łatwiej będzie rozwiązać problem.

Rozwiązywanie problemów

Wiele problemów z klastrami wynika ze złożoności używanego w nich oprogramowania. Czasem łatwiej odbudować klastery od nowa, włącznie z ponowną instalacją systemu operacyjnego. Jest tak zwłaszcza wtedy, gdy użytkownik próbował zainstalować klastery WSFC lub system SQL Server w klastrze, a proces instalacji został przerwany bez usunięcia częściowo zainstalowanych komponentów.

Gdy stworzysz nowy klastery WSFC, odbudowanie go od nowa w celu rozwiązania problemu jest zwykle możliwe, ponieważ nie występują ograniczenia czasowe. Załóżmy jednak, że używasz klastra z systemem SQL Server w środowisku produkcyjnym i nagle wszystkie węzły przestają pracować. Ponieważ wtedy zależy Ci na czasie, warto skontaktować się z Microsoftem.

Współpraca z Microsoftem

Używanie systemu SQL Server w klastrze bez umowy z pomocą techniczną Microsoftu jest jak prowadzenie samochodu bez ubezpieczenia. Możesz to robić, jeśli jednak wystąpią nieoczekiwane problemy, będziesz żałował, że nie wykupiłeś polisy.

Istnieją dwa podstawowe powody kontaktowania się z pomocą techniczną Microsoftu przy wystąpieniu problemów z klastrem. Pierwszy to wykrycie mniej poważnych trudności, z którymi użytkownik nie potrafi sobie sam poradzić. Wtedy klientowi zostaje przydzielony inżynier, który przez kilka dni pomaga w rozwiązywaniu problemu. Ten proces może obejmować używanie narzędzi diagnostycznych w celu zebrania informacji o klastrze, co pozwoli inżynierowi ustalić przyczynę problemów.

Drugi powód to brak pomysłu na szybkie ponowne uruchomienie klastra produkcyjnego, który przestał działać. W tej sytuacji warto jak najszybciej skontaktować się z pomocą techniczną Microsoftu, aby firma zaczęła proces obsługi zgłoszenia. Ponadto powinienś wyraźnie poinformować pierwszą osobę, z którą rozmawiasz, a także przydzielonego inżyniera, że nastąpiło przerwanie pracy klastra produkcyjnego. Oznacza to zgłoszenie *sytuacji krytycznej*. Dla Microsoftu jest to informacja, że dany problem jest priorytetowy, dlatego należy mu poświęcić wyjątkową uwagę. Gdy zgłaszasz sytuację krytyczną, rozmówca musi to zweryfikować, ponieważ zespół pomocy technicznej podejmuje wtedy serię działań. Jeśli jednak klastery produkcyjne nie działają, koniecznie poinformuj o tym, że problem jest poważny. Uzyskasz wtedy natychmiastową pomoc, która doprowadzi do rozwiązania problemów z klastrem WSFC.

Podsumowanie

W tym rozdziale opisaliśmy tylko niewielką część tego, co administrator baz danych powinien wiedzieć na temat klastrów w systemie operacyjnym Windows Server R2 z systemem SQL Server 2014. Ważne jest, aby zrozumiał podstawy instalowania, konfigurowania, testowania, monitorowania, rozwiązywania problemów i konserwowania klastrów WSFC z systemem SQL Server 2014.

Aby z powodzeniem zainstalować klastery WSFC, zacznij od ustalenia potrzebnego sprzętu i sprawdzenia, czy spełnia on wymagania związane z pracą w klastrze. Wykonaj szczegółowe instrukcje z tego rozdziału, aby skonfigurować sprzęt dla klastra. Wszystkie kroki należy wykonać w opisany tu sposób. Odstępstwa mogą prowadzić do problemów z instalacją klastra.

Należy przeprowadzić sprawdzanie poprawności klastra, aby ustalić, czy dana konfiguracja jest prawidłowa. Jeśli wykryjesz błędy, rozwiąż je przed przejściem do dalszych zadań. Jeżeli wszystko działa poprawnie, możesz przystąpić do tworzenia klastra.

Ustal, ile koordynatorów MSDTC potrzebujesz, i zainstaluj je. Za pomocą programu instalacyjnego systemu SQL Server 2014 zainstaluj egzemplarz tego systemu w klastrze. W węźle może pracować kilka egzemplarzy tego systemu. Ponadto można utworzyć wiele węzłów z działającymi egzemplarzami systemu SQL Server, wspomaganych przez węzeł pasywny (a także bez węzła tego rodzaju). Aby dodać kolejne egzemplarze systemu SQL Server w klastrze, uruchom program instalacyjny systemu SQL Server 2014.

Klastry można łączyć ze sobą, aby zapewnić wysoką dostępność środowiska i mechanizm przywracania stanu po awarii. W tym celu należy zastosować *geograficznie rozproszony klastery*, obejmujący węzły z dwóch centrów danych. Węzły te mogą nawet działać w odmiennych podsięciach. Jeśli chcesz dowiedzieć się więcej o różnych konfiguracjach klastrów WSFC, zapoznaj się z artykułami na ten temat, dokumentacją Books Online poświęconą systemowi SQL Server 2014 i dodatkowymi informacjami z witryny Microsoftu oraz innych stron.

W rozdziale 17. przeczytasz o tworzeniu kopii zapasowych i przywracaniu stanu w systemie SQL Server 2014.

Kopie zapasowe i przywracanie stanu

ZAWARTOŚĆ ROZDZIAŁU:

- Różne rodzaje awarii i ich przyczyny.
- Planowanie reakcji na awarię.
- Jak działają kopie zapasowe?
- Wybór odpowiedniej konfiguracji tworzenia kopii zapasowych w danym środowisku.
- Przywracanie baz danych po wystąpieniu problemów.

Dane to zasób o znaczeniu strategicznym. Przez lata firmy zbierają informacje o klientach, stanie magazynu, zakupach, finansach i produktach. Dane mają służyć do ulepszenia poziomu obsługi klienta, a także przydają się podczas podejmowania ważnych decyzji. Przestoje są nieakceptowalne i mogą okazać się dla firmy kosztowne. Bez baz danych dom maklerski nie może przyjmować zleceń zakupu akcji, a linie lotnicze nie mogą sprzedawać biletów. Każda godzina braku dostępu do bazy może oznaczać dla firmy utratę milionowych zysków. Aby zapewnić ciągłość pracy, organizacje stosują rozwiązania gwarantujące wysoką dostępność (np. klastry z przełączaniem awaryjnym, kopie lustrzane baz danych, replikację i przesyłanie dzienników). Dzięki temu nawet po awarii serwera z bazą danych firma funkcjonuje, wykorzystując serwer rezerowy. Wymienione wcześniej techniki omawiamy w innych rozdziałach tej książki.

Ponadto bazy można chronić za pomocą zapewniających odporność na błędy i wysoką dostępność technologii przechowywania danych, takich jak macierze RAID. Jeśli nawet firma stosuje takie technologie, potrzebuje kopii zapasowych, aby możliwe było przywrócenie stanu po uszkodzeniu lub utracie danych, a także w ramach przywracania baz po katastrofie, gdy nastąpi całkowita awaria głównego centrum danych.

Choć rozwiązania zapewniające wysoką dostępność mają gwarantować ciągły dostęp do danych, plan tworzenia kopii zapasowej bazy jest niezbędny, jeśli chcesz chronić dane firmy. Jeżeli istnieje problem z danymi i bazy nie da się naprawić, administrator może wykorzystać kopię zapasową, aby przywrócić bazę w poprawnym stanie. Ponadto dobra strategia tworzenia kopii zapasowej pozwala ograniczyć utratę danych przy wystąpieniu określonych błędów napotykanym w codziennej pracy.

W tym rozdziale najpierw omawiamy proces tworzenia kopii zapasowych i przywracania stanu. Następnie przedstawiamy projektowanie i opracowanie planu tych czynności. Opisujemy też archiwizację danych i planowanie przywracania stanu po katastrofie.

Usprawnienia związane z tworzeniem kopii zapasowej i przywracaniem stanu

System SQL Server 2014 zawiera następujące usprawnienia związane z tworzeniem kopii zapasowej i przywracaniem stanu.

- Tworzenie kopii zapasowych baz SQL Server z zapisem na adres URL.
- Zarządzane tworzenie kopii zapasowych baz SQL Server z zapisem w platformie Windows Azure.
- Szyfrowanie kopii zapasowych.

Tworzenie kopii zapasowych z zapisem na adres URL

Mechanizm wprowadzono w wersji SQL Server 2012 SP1 CU2. Początkowo był on obsługiwany w języku T-SQL, powłoce Windows PowerShell i dla obiektów SMO (ang. *Server Management Object*). W systemie SQL Server 2014 można za pomocą programu SQL Server Management Studio tworzyć kopie zapasowe i przywracać bazy z wykorzystaniem usługi Windows Azure Blob Storage. Nowa opcja jest dostępna zarówno w zadaniu Backup, jak i w planach konserwacji.

Zarządzane tworzenie kopii zapasowych z zapisem w platformie Windows Azure

Mechanizm jest oparty na technologii opisanej w poprzednim punkcie. Jest to udostępniana przez system SQL Server usługa przeznaczona do zarządzania tworzeniem kopii zapasowych baz i dzienników oraz planowania tych operacji. Obecnie obsługiwane jest tylko tworzenie kopii zapasowych w platformie Windows Azure. Technologię tę można skonfigurować na poziomie bazy danych lub egzemplarza, co pozwala zarówno na precyzyjną kontrolę ustawień dla baz danych, jak i na automatyzację zadań na poziomie egzemplarza. Zarządzanie tworzeniem kopii zapasowych z zapisem w platformie Windows Azure można skonfigurować w egzemplarzach systemu SQL Server działających lokalnie, a także w egzemplarzach pracujących na maszynach wirtualnych z tą platformą (zalecane jest drugie z tych podejść).

Szyfrowanie kopii zapasowych

Obecnie możesz zaszyfrować plik z kopią zapasową w trakcie jej tworzenia. System SQL Server obsługuje kilka algorytmów szyfrowania, w tym AES 128, AES 192, AES 256 i Triple DES. Do szyfrowania potrzebny jest certyfikat lub klucz asymetryczny.

Przegląd procesu tworzenia kopii zapasowej i przywracania stanu

Zanim będziesz mógł opracować skuteczny plan tworzenia kopii zapasowej i przywracania stanu, musisz zrozumieć przebieg tych operacji na poziomie mechanicznym. System SQL Server udostępnia kilka procesów tworzenia kopii zapasowych i przywracania stanu. Należy je dostosować do potrzeb organizacji. W tym podrozdziale wyjaśniamy przebieg omawianych operacji i pomagamy wybrać optymalny plan.

Jak przebiega tworzenie kopii zapasowej?

Tworzenie kopii zapasowej bazy danych to procedura, która chroni inwestycje firmy i pozwala zmniejszyć ilość utraconych danych. Polega ona na zapisaniu kopii danych i dziennika transakcji w formie obrazu na dysku lub taśmach. Dane te pochodzą z określonego momentu. System SQL Server udostępnia rozbudowany proces tworzenia kopii zapasowych, którego elementy można stosować osobno lub razem w celu przygotowania optymalnej strategii dostosowanej do firmy. System ten może tworzyć kopie zapasowe w czasie, gdy baza jest aktywna i dostępna dla użytkowników.

Można wykorzystać następujące rodzaje kopii zapasowych.

- **Pełna kopia.** Obejmuje wszystkie dane z bazy, włącznie z dziennikiem transakcji. Przy użyciu takiej kopii można przywrócić bazę do stanu z momentu, w którym kopia została utworzona. Jest to podstawowy typ kopii zapasowych, często niezbędny przed zastosowaniem pozostałych technik. Gdy przywracasz stan bazy na podstawie pełnej kopii, przywracane są wszystkie jej pliki (jednak bez zależności), po czym baza jest udostępniana i ma spójny transakcyjnie stan.
- **Częściowa kopia zapasowa.** Jest to sposób na utworzenie kopii tylko tych części bazy, które się zmieniły. Zmniejsza to wielkość kopii zapasowej i czas jej tworzenia oraz przywracania stanu. W tym podejściu tworzona jest kopia głównej grupy plików oraz grup plików do odczytu i zapisu. Aby zastosować tę technikę, musisz pogrupować modyfikowane tabele w grupy plików, a statyczne i historyczne tabele umieścić w odrębnych grupach plików. Grupy plików z danymi historycznymi działają w trybie odczytu i zapisu lub w trybie tylko do odczytu. Technika ta przyspiesza proces tworzenia kopii zapasowej baz z dużymi obszarami przeznaczonymi tylko do odczytu. W bazie mogą znajdować się dane archiwalne, które się nie zmieniają, dlatego nie wymagają każdorazowego zapisu w kopii zapasowej. Zmniejsza to ilość danych, które trzeba umieścić w kopii zapasowej.
- **Kopia zapasowa plików i grup plików.** Technika ta polega na kopiowaniu wybranych plików lub grup plików z bazy danych. Metodę zwykle stosuje się w dużych bazach danych, w których utworzenie pełnej kopii jest niemożliwe. Jeśli kopia zapasowa obejmuje pliki lub grupy plików działające w trybie odczytu i zapisu, potrzebna jest kopia

dziennika transakcji. Trudności w tym podejściu sprawia zarządzanie kopiami zapasowymi plików, grup plików i dzienników transakcji, ponieważ w dużych bazach znajduje się wiele plików i ich grup. Ponadto proces przywracania stanu przy stosowaniu tej techniki wymaga więcej kroków.

UWAGA W trakcie tworzenia kopii zapasowej plików lub grup plików tabelę i wszystkie jej indeksy trzeba zapisać w tej samej kopii. System SQL Server sprawdza tę regułę i zgłasza błąd, jeśli nie jest przestrzegana. Aby można było wykorzystać kopie zapasowe plików i grup plików, musisz odpowiednio zaplanować lokalizację indeksów.

- **Różnicowe kopie zapasowe.** Obejmują wszystkie dane, które zmieniły się od czasu utworzenia ostatniej pełnej kopii zapasowej. Proces tworzenia kopii zapasowej w systemie SQL Server 2014 wykrywa wszystkie zmodyfikowane ekstenty i opracowuje ich kopię zapasową. Kopie różnicowe zawierają wszystkie zmiany wprowadzone od zapisania ostatniej kopii zapasowej. Jeśli stworzysz pełną kopię zapasową w niedzielę w nocy, kopia różnicowa zapisana w poniedziałek w nocy obejmuje wszystkie zmiany wprowadzone od niedzielnej nocy. Jeśli we wtorek w nocy utworzysz nową kopię różnicową, też będzie ona zawierać wszystkie modyfikacje wprowadzone od niedzielnej nocy. Przy przywracaniu danych należy użyć ostatniej pełnej kopii zapasowej i najnowszej kopii różnicowej. Następnie trzeba przywrócić kopie zapasowe dzienników transakcji wykonane po ostatniej kopii różnicowej. Pozwala to przyspieszyć przywracanie stanu. To, czy kopie różnicowe będą przydatne, zależy od tego, jaki odsetek wierszy jest modyfikowany między utworzeniem kolejnych pełnych kopii zapasowych. Gdy zmieniają się prawie wszystkie wiersze, wielkość kopii różnicowej jest niemal równa wielkości pełnej kopii zapasowej. W takiej sytuacji często lepiej utworzyć kolejną pełną kopię, a następnie zacząć tworzenie nowych kopii różnicowych. Kopie różnicowe są też przydatne wtedy, gdy grupa wierszy często się zmienia. Pamiętaj, że kopia zapasowa dziennika transakcji obejmuje wszystkie zmiany. Z kolei w kopii różnicowej zapisana jest tylko ostatnia modyfikacja każdego wiersza. Wyobraź sobie, że baza danych zawiera wartości akcji 100 spółek. Wartości te są aktualizowane co minutę. Każdy wiersz jest więc aktualizowany 1440 razy dziennie. Załóżmy, że niedzielnej nocy tworzona jest pełna kopia zapasowa, a w tygodniu zapisywane są kopie zapasowe dziennika transakcji. Jeśli trzeba będzie przywrócić dane na podstawie wszystkich dzienników transakcji w piątkowy wieczór, konieczne będzie odtworzenie wszystkich zmian z każdego wiersza. W tym przykładzie każdy wiersz jest aktualizowany 7200 razy (1440 razy 5 dni w tygodniu). Dla 100 spółek oznacza to konieczność odtworzenia 720 000 transakcji. Gdyby zamiast tego zastosować kopie różnicowe, wystarczyłoby zastąpić tylko 100 wierszy. W kopii różnicowej zachowywana jest tylko najnowsza wersja danych. W niektórych sytuacjach jest to bardzo wygodne.
- **Częściowe kopie różnicowe.** To podejście działa tak samo jak tworzenie różnicowej kopii zapasowej, ale uwzględnia tylko dane z częściowej kopii zapasowej. Zapisywane są tu wszystkie ekstenty zmodyfikowane od czasu utworzenia ostatniej częściowej kopii zapasowej. Aby przywrócić stan na podstawie częściowej kopii różnicowej, potrzebna jest podstawowa częściowa kopia zapasowa.
- **Różnicowe kopie zapasowe plików.** Są to kopie tych ekstentów plików lub grup plików, które zostały zmodyfikowane od czasu ostatniego utworzenia kopii zapasowych określonych plików lub grup plików. Dla plików i grup plików działających w trybie odczytu i zapisu potrzebna jest też kopia zapasowa dziennika transakcji. Po przywróceniu kopii różnicowej

zapasowej pliku należy przywrócić także dziennik transakcji. Stosowanie kopii zapasowych plików i różnicowych kopii zapasowych plików prowadzi do wzrostu złożoności procedur przywracania stanu. Ponadto przywrócenie kompletnej bazy danych może wtedy zajmować więcej czasu.

- **Tylko kopiowanie.** Podejście to można zastosować dla bazy danych lub dziennika transakcji. Proces obejmujący tylko kopiowanie nie jest uwzględniany w standardowych procedurach tworzenia kopii zapasowych i przywracania stanu. Zwykle tworzenie pełnej kopii zapasowej powoduje wyzerowanie różnicowych kopii zapasowych, natomiast podejście z samym kopiowaniem nie wpływa na różnicowe kopie zapasowe — nadal zawierają one wszystkie zmiany od momentu utworzenia ostatniej pełnej kopii. Metoda z samym kopiowaniem dziennika transakcji nie powoduje opróżnienia dziennika i nie wpływa na następne normalne tworzenie kopii zapasowej takiego dziennika. Podejście z samym kopiowaniem jest przydatne, gdy chcesz utworzyć kopię bazy danych na potrzeby testów lub programowania i nie zakłócać przy tym procesu przywracania stanu.

W systemie SQL Server dziennik transakcji jest głównym komponentem, który pozwala zachować zgodność transakcji z modelem ACID (ang. *atomicity, consistency, isolation, durability*, czyli atomowość, spójność, izolacja i trwałość). System SQL Server używa protokołu WAL (ang. *Write Ahead Logging*), co oznacza, że rekordy dziennika transakcji są zapisywane na stabilnym nośniku przed zapisaniem danych na dysku i przed wysłaniem przez system SQL Server potwierdzenia o trwałym zatwierdzeniu danych. *Stabilny nośnik* oznacza tu zwykle bezpośrednio podłączony dysk twardy, jednak może to być dowolne urządzenie gwarantujące, że po wyłączeniu zasilania dane nie znikną.

Nawet przy korzystaniu z bezpośrednio podłączonych systemów takie rozwiązanie może sprawiać trudności. Zarówno dyski, jak i kontrolery macierzy RAID (nawet najprostsze) mają pamięć podręczną zapisywanych danych, która musi być chroniona przed zapisem lub podtrzymywana bateryjnie. Także w zewnętrznych systemach magazynowania danych, takich jak sieci SAN, trzeba sprawdzić, czy pamięć podręczna ma zasilanie bateryjne i czy nawet po wystąpieniu awarii zasilania zachowana zostanie spójność rekordów dziennika.

Obecnie popularne stają się urządzenia pamięci masowej oparte na pamięci flash. Przyjmują one wiele postaci. Jeśli korzystasz z takich urządzeń, upewnij się, że albo gwarantują zapis po wystąpieniu awarii zasilania, albo są używane w miejscach, gdzie działanie pamięci podręcznej zapisywanych danych nie ma dużego znaczenia (np. dla bazy tempdb, której dane i tak są usuwane w momencie ponownego uruchamiania systemu). W szybkich systemach, w których wymagana jest jak najwyższa wydajność zapisu dziennika transakcji, coraz częściej umieszcza się takie dzienniki na nośnikach opartych na pamięci flash. Choć zapewnia to wysoką wydajność, trzeba też zagwarantować trwałość danych przy awarii zasilania.

Silnik bazodanowy systemu SQL Server wymaga zachowania spójnego stanu dziennika transakcji po ponownym uruchomieniu systemu. Jeśli dziennik jest niespójny, silnik bazodanowy uzna bazę za uszkodzoną, ponieważ nie będzie można stwierdzić, czy dane są poprawne.

Ponadto przy modyfikowaniu danych system SQL Server generuje numery LSN (ang. *log sequence number*), używane przy ponownym uruchomieniu do sprawdzania spójności danych, gdy trzeba przywrócić bazę. Numery LSN są potrzebne przy przywracaniu dziennika transakcji. System SQL Server używa ich do ustalenia kolejności operacji z każdego dziennika. Jeśli np. kopia zapasowa dziennika transakcji z łańcucha dzienników jest niedostępna, powstaje *zerwany łańcuch dzienników*. Uniemożliwia to przywrócenie dziennika transakcji spoza danego punktu. Tworzenie kopii zapasowej dzienników transakcji od punktu przywracania jest niezbędnym elementem strategii zarządzania kopiami zapasowymi.

Kopiowanie baz danych

Czasem administrator nie chce tworzyć kopii zapasowej bazy, a jedynie zamierza skopiować dane. W tym punkcie opisujemy kilka metod kopiowania baz danych.

Odlaczanie i podłączanie

Odlaczanie i podłączanie to doskonały sposób przenoszenia bazy danych na nowy serwer. Technikę można też wykorzystać do utworzenia kopii bazy. Mechanizm ten powoduje odłączenie bazy danych lub zamknięcie systemu SQL Server. Pliki po odłączeniu można skopiować w nowe miejsce.

UWAGA Obecnie procedura `sp_attach_db` jest uważana za przestarzałą. Choć w systemie SQL Server 2014 nadal jest dostępna, w skryptach należy ją zastąpić nową instrukcją `CREATE DATABASE FOR DETACH`. Zauważ, że procedura `sp_detach_db` nie jest przestarzała. Dlatego choć istnieje instrukcja `ALTER DATABASE SET OFFLINE` (powiązana z instrukcją `CREATE DATABASE FOR DETACH`), wciąż można stosować procedurę `sp_detach_db`.

Aby utworzyć kopię zapasową za pomocą tej techniki, odłącz bazę za pomocą systemowej procedury składowanej `sp_detach_db` lub instrukcji `ALTER DATABASE SET OFFLINE`, co ilustrują dwa poniższe fragmenty kodu.

Oto przykład zastosowania procedury `sp_detach_db` (plik *KodR17.sql*):

```
EXEC master.dbo.sp_detach_db @dbname = N'ch17_samples'
```

Instrukcję `ALTER DATABASE` stosuje się tak:

```
ALTER DATABASE ch17_samples SET OFFLINE
```

Aby przywrócić dane, możesz albo utworzyć nową bazę za pomocą instrukcji `CREATE DATABASE`, albo — jeśli masz już bazę i chcesz ją zaktualizować za pomocą nowej kopii z innego serwera — zastąpić pliki i zastosować polecenie `ALTER DATABASE SET ONLINE`.

Oto przykład z instrukcją `CREATE DATABASE` (plik *KodR17.sql*):

```
CREATE DATABASE [ch17_samples] ON
( FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\
  MSSQL\DATA\ch17_samples.mdf' ),
( FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\
  MSSQL\DATA\ch17_samples_log.ldf' )
FOR ATTACH;
```

UWAGA Powinieneś zmienić użytą ścieżkę, tak aby prowadziła do katalogu, w którym znajdują się pliki z danymi. W tym przykładzie zastosowano ścieżkę domyślną.

Oto przykładowy kod z instrukcją `ALTER DATABASE SET ONLINE`:

```
ALTER DATABASE ch17_samples SET ONLINE;
```

UWAGA Jeśli dołączasz zaszyfrowaną bazę danych, właściciel bazy musi odszyfrować klucz główny bazy za pomocą następującego polecenia:

```
OPEN MASTER KEY DECRYPTION BY PASSWORD = '<hasło>'
```

Microsoft zaleca włączenie automatycznego deszyfrowania klucza głównego. Służy do tego następująca instrukcja:

```
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY
```

BCP

Druga technika tworzenia kopii zapasowych jest oparta na programie BCP (ang. *Bulk Copy Program*). Można wywołać go za pomocą prostego skryptu wsadowego i wyeksportować wszystkie dane z bazy. To rozwiązanie nie uwzględnia wszystkich danych z bazy, jest jednak przydatne do kopiowania konkretnych informacji.

Jeśli chcesz wyeksportować z tabeli więcej niż 2 miliardy wierszy, *musisz* użyć programu BCP z systemu SQL Server 2008 lub jego nowszej wersji. Błąd w starszych wersjach (SQL Server 2005 i dawniejszych) uniemożliwiał eksportowanie i importowanie tak dużych zbiorów danych, przy czym program nie zgłaszał wtedy żadnych błędów.

Oto dwa przykładowe wywołania programu BCP w wierszu poleceń. Pokazują one, jak wyeksportować (OUT) i zaimportować (IN) tabelę *People* z serwera *Seattle_SQL* na serwer *London_SQL*:

```
BCP address OUT address.dat -SSeattle_SQL -T -c -dPeople
```

```
BCP address IN address.dat -SLondon_SQL -T -c -dPeople
```

Kreator skryptów

Program SQL Server Management Studio udostępnia kreator skryptów, który umożliwia generowanie skryptów dotyczących określonych obiektów bazy danych. Jest to doskonały sposób na uzyskanie skryptów potrzebnych do odtworzenia bazy danych. W ten sposób można też sprawdzać ostatnio wprowadzone zmiany w bazie.

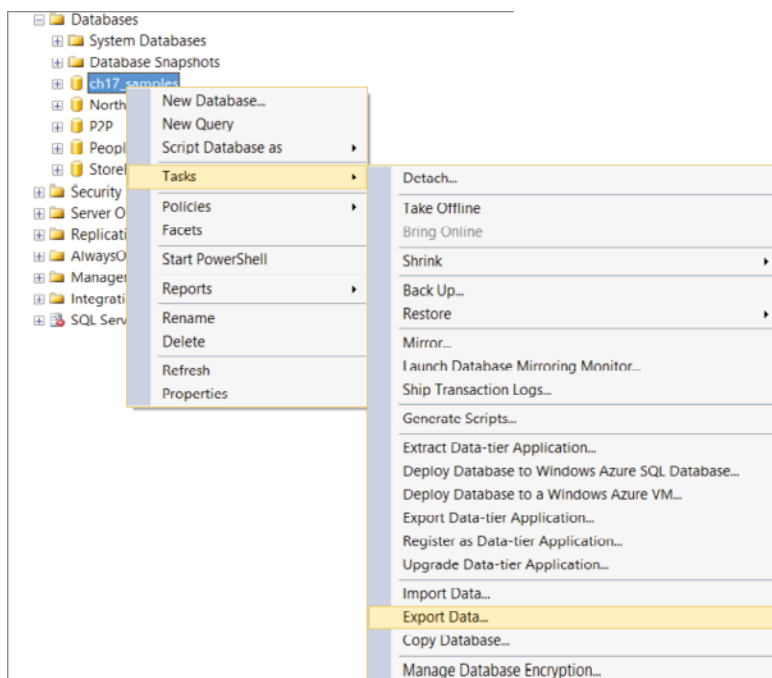
Zastosuj tę technikę razem z systemem kontroli wersji kodu źródłowego, aby określić różnicę między wcześniej wygenerowanymi skryptami a nowymi skryptami. Pozwoli to wykryć zmiany w obiektach bazy danych.

Kreator importu i eksportu

Program SQL Server Management Studio udostępnia mechanizmy importu i eksportu. Są one dostępne za pomocą opcji *Tasks* w menu kontekstowym bazy danych, co przedstawiono na rysunku 17.1.

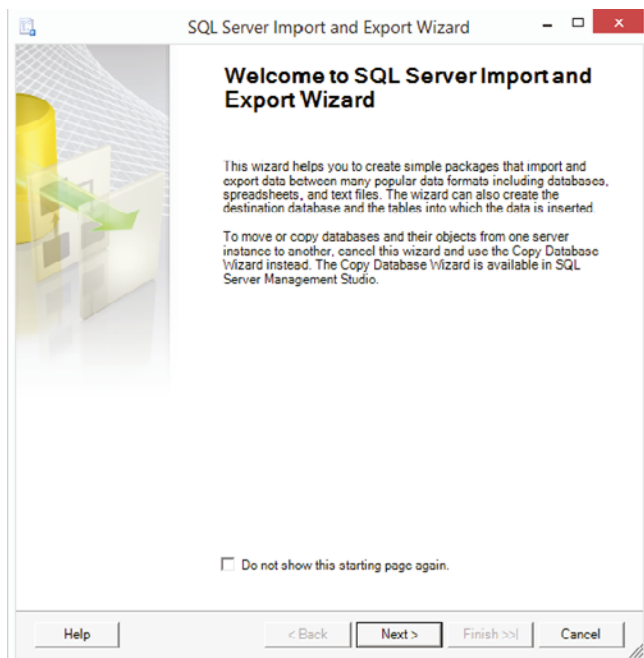
Możliwy jest tu zarówno eksport, jak i import danych. W poniższym przykładzie pokazano, jak wyeksportować dane ze źródłowej bazy.

UWAGA Przykładowe tabele potrzebne w tym ćwiczeniu można wygenerować za pomocą pliku *TabelaR17.sql*. Jest on dostępny do pobrania w witrynie wydawnictwa Helion.



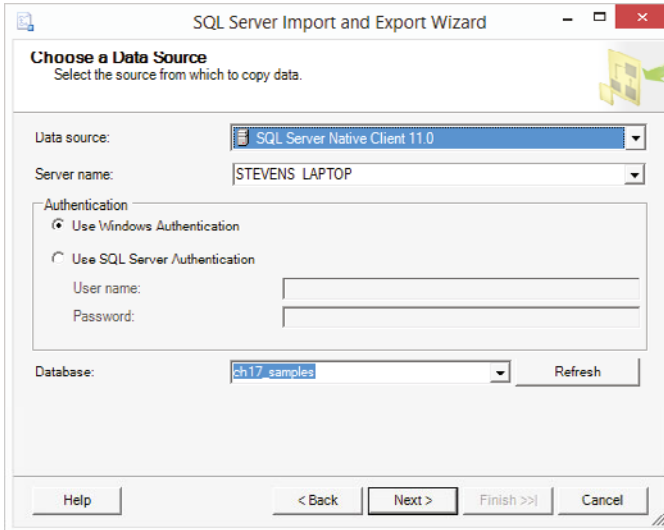
Rysunek 17.1. Opcje importu i eksportu

1. Wybierz opcję *Export Data*, aby uruchomić kreator *Import and Export Wizard*. Pojawi się ekran powitalny kreatora widoczny na rysunku 17.2.



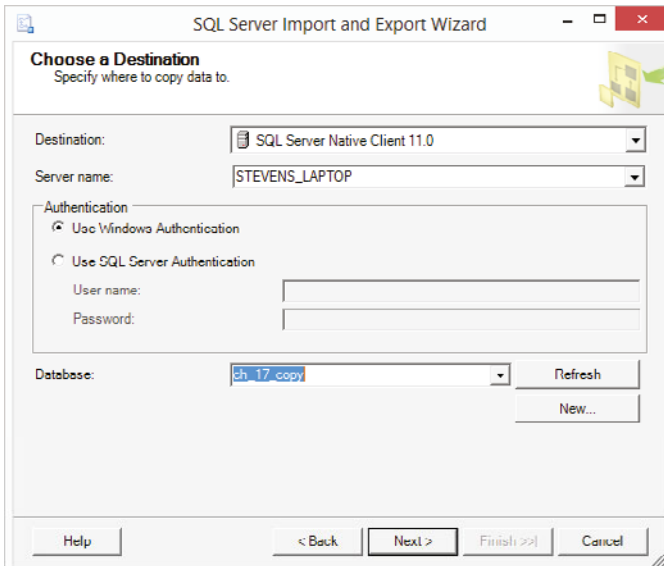
Rysunek 17.2. Ekran powitalny kreatora importu i eksportu

2. Kliknij przycisk *Next* na ekranie powitalnym. Pojawi się strona *Choose a Data Source* przedstawiona na rysunku 17.3. Wprowadź nazwę serwera i używaną na nim metodę uwierzytelniania. Następnie wybierz bazę, z której chcesz wyeksportować dane. Tu jest to baza `ch17_samples`, co ilustruje ten sam rysunek.



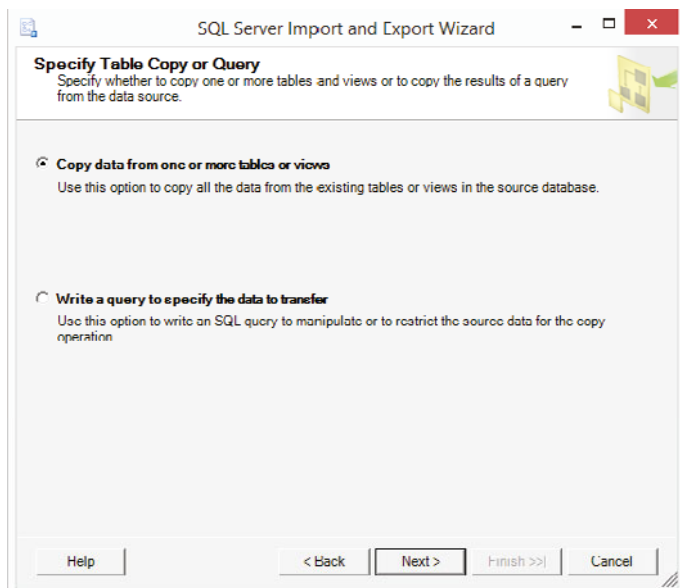
Rysunek 17.3. Wybieranie źródła danych

3. Następną stroną kreatora to *Choose a Destination*. Przedstawiono ją na rysunku 17.4. Wprowadź tu nazwę serwera, używaną dla niego metodę uwierzytelniania i nazwę bazy. Kreator standardowo wybiera domyślną bazę danych loginu używanego do uwierzytelniania się na danym serwerze. Kliknij przycisk *Next*.



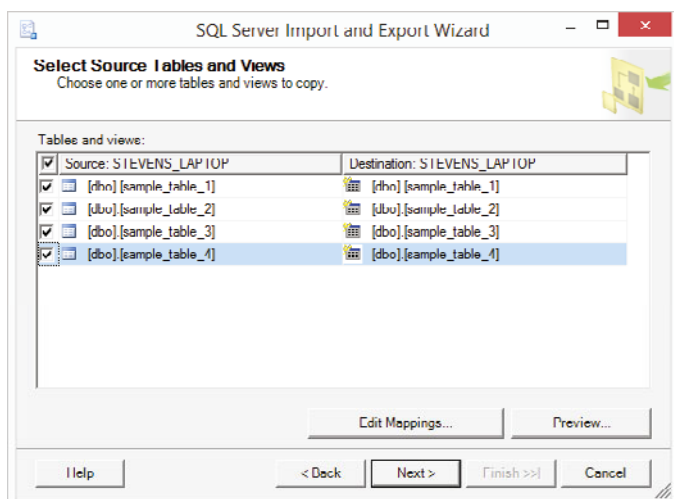
Rysunek 17.4. Określanie docelowej lokalizacji danych

- Na następnej stronie (patrz rysunek 17.5) wybierz opcję *Copy data from one or more tables or views*, aby skopiować tabelę, lub opcję *Write a query to specify the data to transfer*, by napisać kwerendę pobierającą dane. W tym przykładzie wybierz pierwszą opcję, co pozwoli skopiować dane z tabel lub widoków. Następnie kliknij przycisk *Next*.



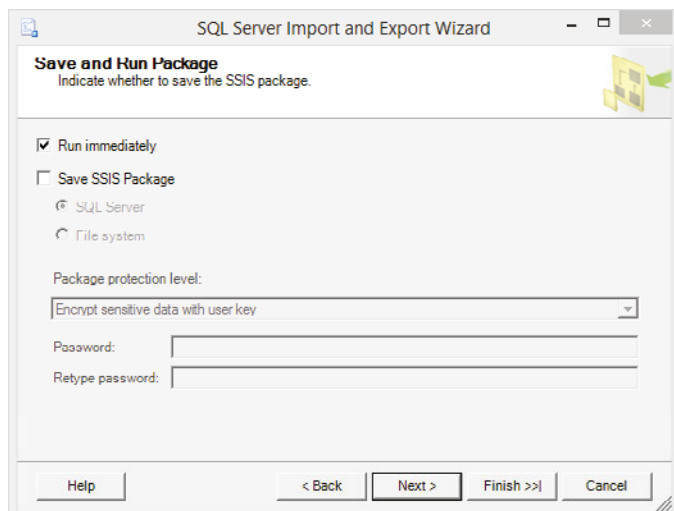
Rysunek 17.5. Możesz skopiować tabele lub widoki albo napisać kwerendę pobierającą dane

- Na ekranie *Select Source Tables and Views* (widocznym na rysunku 17.6) kreator przedstawia listę wszystkich dostępnych tabel i widoków. W przykładowej bazie danych dostępnych jest tylko kilka tabel, co przedstawiono na wspomnianym rysunku. Zaznacz tabele i widoki, które chcesz wyeksportować, a następnie kliknij przycisk *Next*.



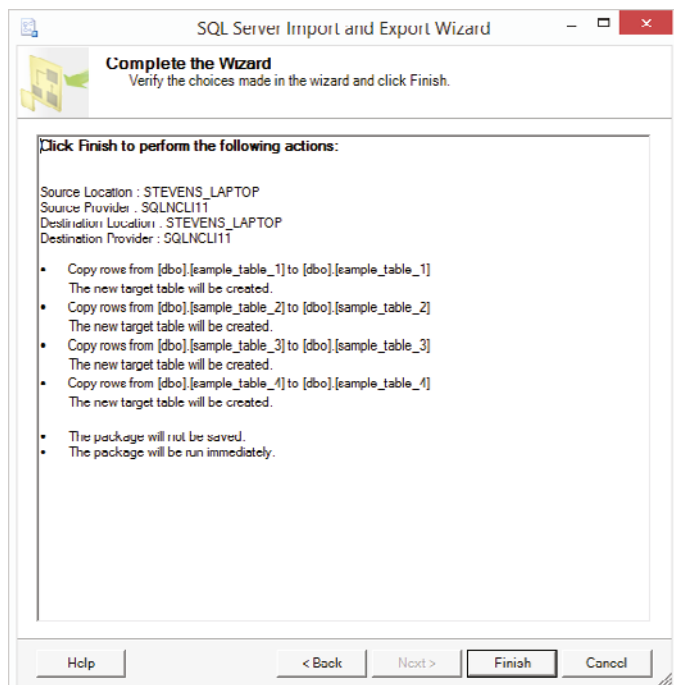
Rysunek 17.6. Wybieranie kopiowanych tabel

6. Na ekranie *Save and Run Package* (przedstawionym na rysunku 17.7) możesz teraz od razu uruchomić pakiet lub zapisać pakiet SSIS. Jeśli zdecydujesz się zachować pakiet, określ, czy chcesz go umieścić w systemie SQL Server czy w systemie plików. Możesz też ustawić poziom ochrony zawartości pakietu. Kliknij przycisk *Next*.



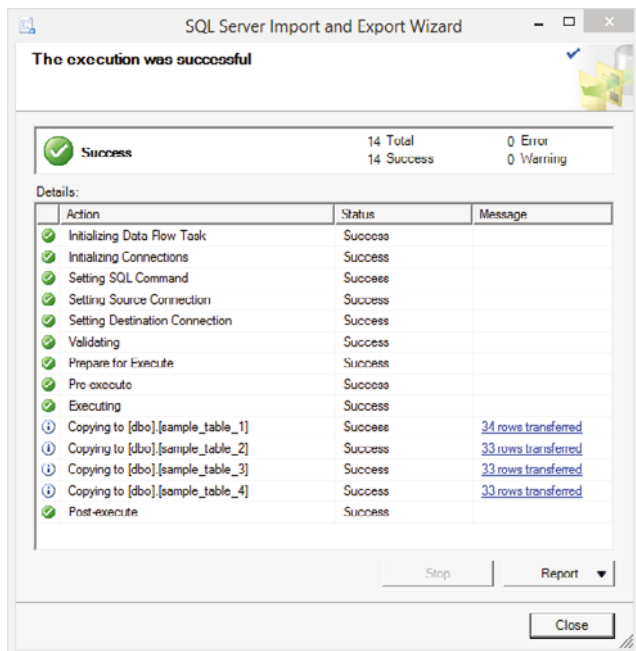
Rysunek 17.7. Jeśli chcesz, możesz zapisać pakiet

7. Na ekranie *Complete the Wizard* (patrz rysunek 17.8) widoczne są ustawione wcześniej opcje.



Rysunek 17.8. Przegląd ustawionych opcji

8. Kliknij przycisk *Finish*, aby wyeksportować dane zgodnie z ustawionymi w kreatorze opcjami. Tu kreator uruchamia pakiet SSIS, aby wyeksportować wybrane tabele ze źródłowego serwera, tworzy nowe tabele na docelowym serwerze i importuje dane do nowo utworzonych tabel. Następnie kreator informuje o statusie wykonywania pakietu. Na rysunku 17.9 pokazano, że wszystkie operacje zostały wykonane z powodzeniem.



Rysunek 17.9. Eksportowanie zakończyło się powodzeniem

Warto zauważyć, że jeśli wystąpią błędy, a Ty cofniesz się, aby wprowadzić zmiany i ponownie uruchomić pakiet, możesz natrafić na dodatkowe problemy, ponieważ kreator nie cofa zmian, jeżeli w trakcie wykonywania zadań pojawiają się kłopoty.

Błędy mogą wystąpić np. wtedy, gdy tabele zostaną utworzone, ale importowanie danych się nie powiedzie z powodu przepełnienia dziennika transakcji. W tej sytuacji można albo ręcznie usunąć tabele, albo wyświetlić utworzone tabele w docelowej bazie danych i ponownie uruchomić pakiet. Inne częste błędy związane są z przycinaniem danych, różnicami w typach danych i tożsamością rekordów przy wstawianiu.

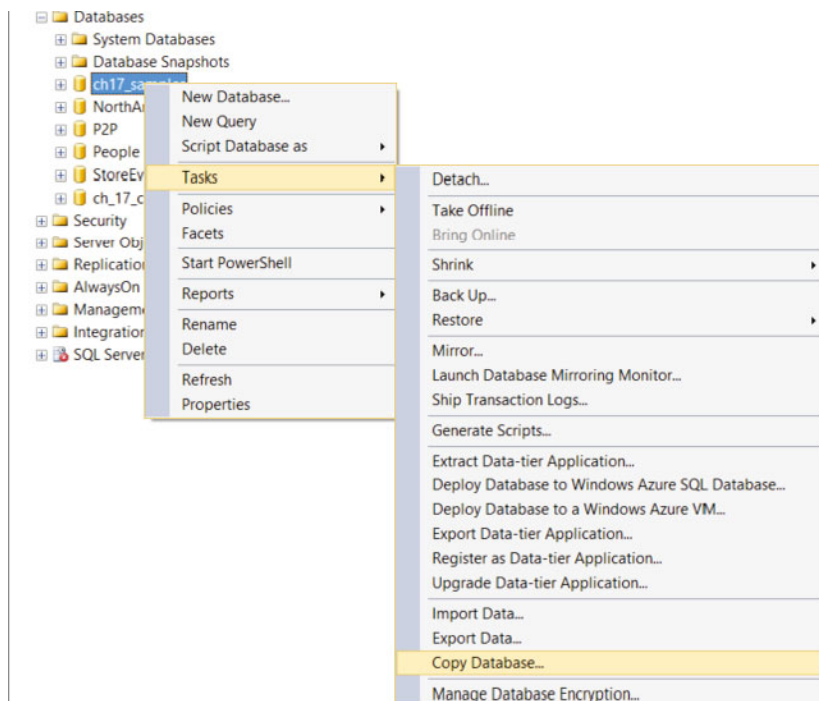
UWAGA Pakiet do działania nie wymaga uruchomionej usługi SQL Server Agent.

Aplikacje warstwy danych

Jeśli w bazie danych używane są mechanizmy zgodne z aplikacjami warstwy danych (ang. *Data Tier Application* — DAC), można utworzyć taką aplikację za pomocą kreatora z programu SQL Server Management Studio. Aplikacja DAC obejmuje sam schemat bazy (bez danych), co jest przydatne przy przenoszeniu schematów między środowiskami rozwojowym, testowym, kontroli jakości i produkcyjnym. Szczegółowe informacje o stosowaniu aplikacji DAC znajdziesz w poświęconych im częściach dokumentacji Books Online.

Kreator kopiowania baz danych

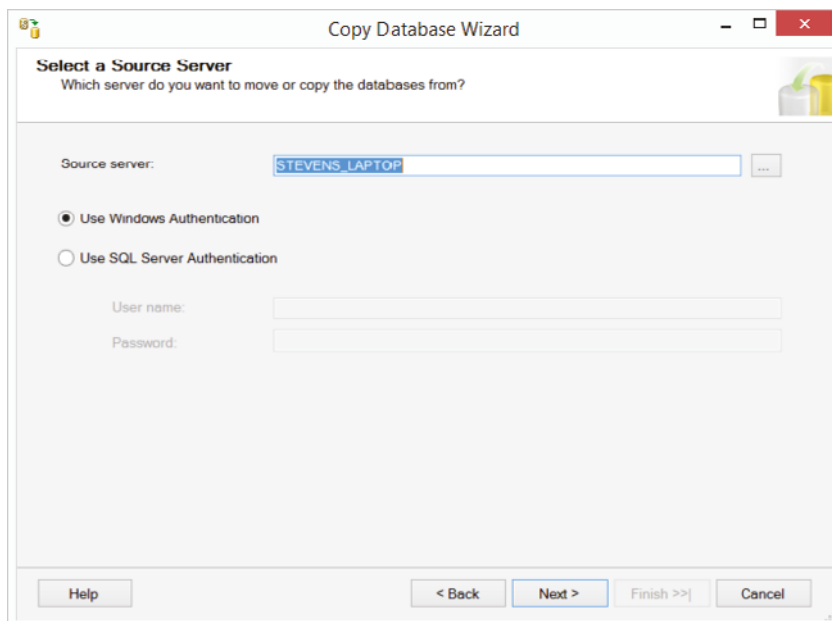
Program SQL Server Management Studio udostępnia kreator kopiowania baz danych. Można w nim wykorzystać mechanizm dołączania i odłączania baz lub obiekty SMO, aby utworzyć kopię bazy i przenieść ją na inny serwer. Jeśli chcesz zastosować dołączanie i odłączanie, dana baza musi być wyłączona. Technika oparta na obiektach SMO pozwala skopiować bazę bez jej wyłączania. Kreator kopiowania baz danych możesz uruchomić za pomocą opcji *Tasks* w menu kontekstowym bazy danych (opcja *Copy Database* na rysunku 17.10).

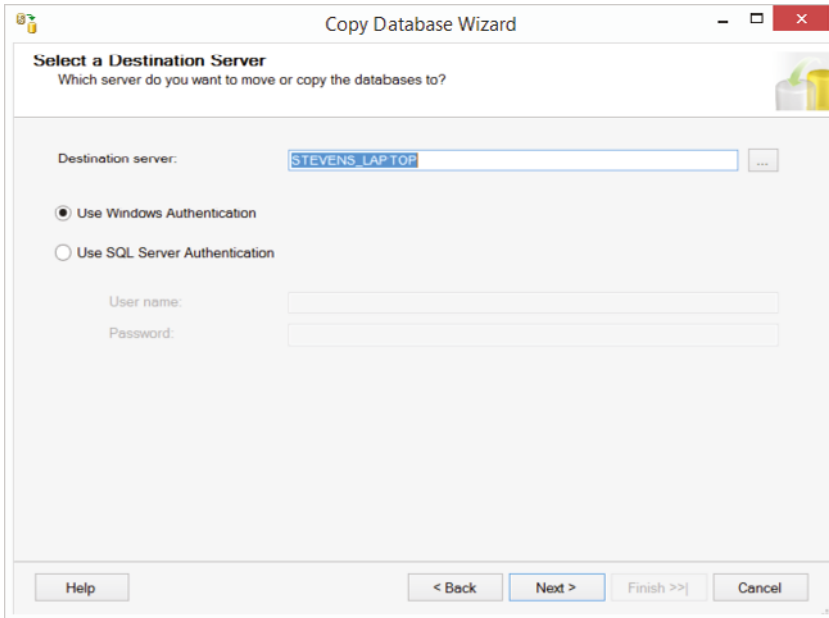


Rysunek 17.10. Aby uruchomić kreator kopiowania baz, wybierz opcję *Tasks/Copy Database*

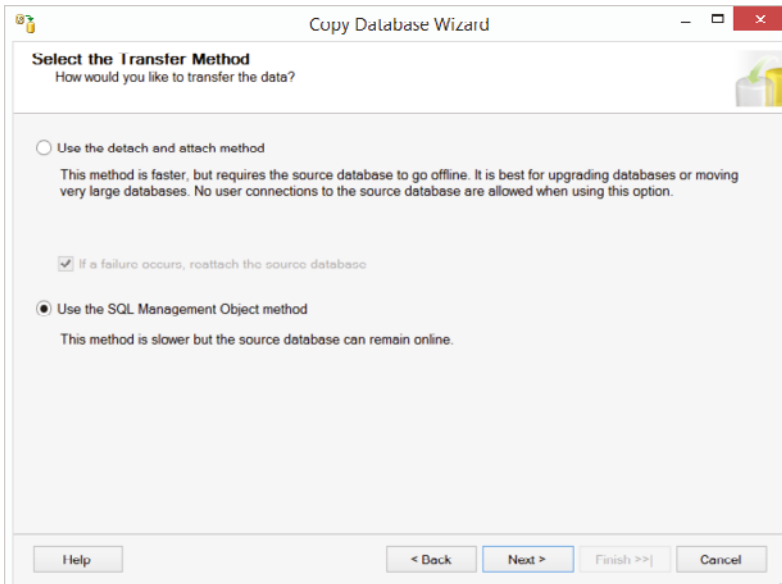
W poniższych krokach pokazano, jak używać tego kreatora.

1. Otwórz kreator kopiowania baz danych. Pojawi się ekran powitalny widoczny na rysunku 17.11. Kliknij przycisk *Next*.
2. Wybierz serwer źródłowy, co ilustruje rysunek 17.12. Kliknij przycisk *Next*.
3. Na stronie *Select a Destination Server* (patrz rysunek 17.13) wybierz serwer docelowy. Kreator standardowo używa domyślnej bazy danych loginu stosowanego do uwierzytelniania się na serwerze. Kliknij przycisk *Next*.
4. Na przedstawionej na rysunku 17.14 stronie *Select the Transfer Method* wybierz, w jaki sposób kreator ma przysyłać dane. Dostępne opcje to dołączanie i odłączanie oraz zastosowanie obiektów SMO. W tym przykładzie wybierz drugą z tych metod, a następnie kliknij przycisk *Next*.

**Rysunek 17.11.** Ekran powitalny kreatora**Rysunek 17.12.** Wybieranie serwera źródłowego

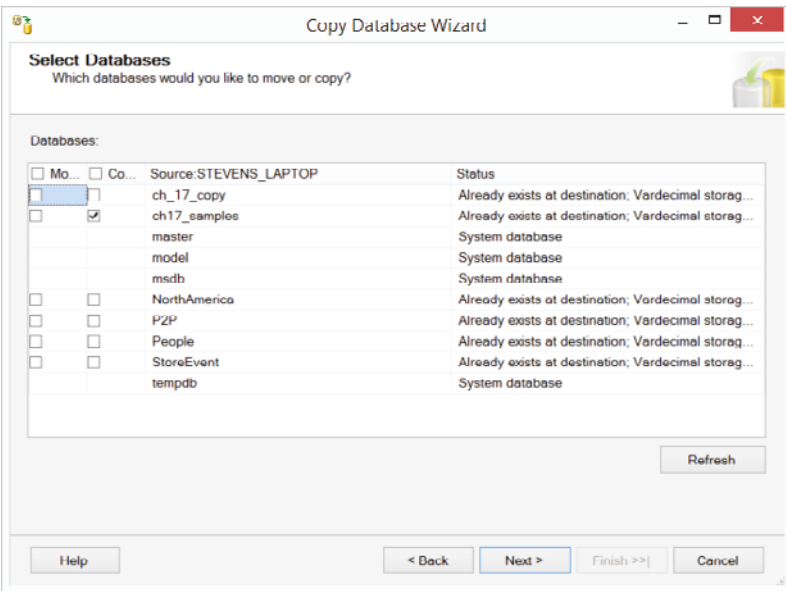


Rysunek 17.13. Ustawianie serwera docelowego



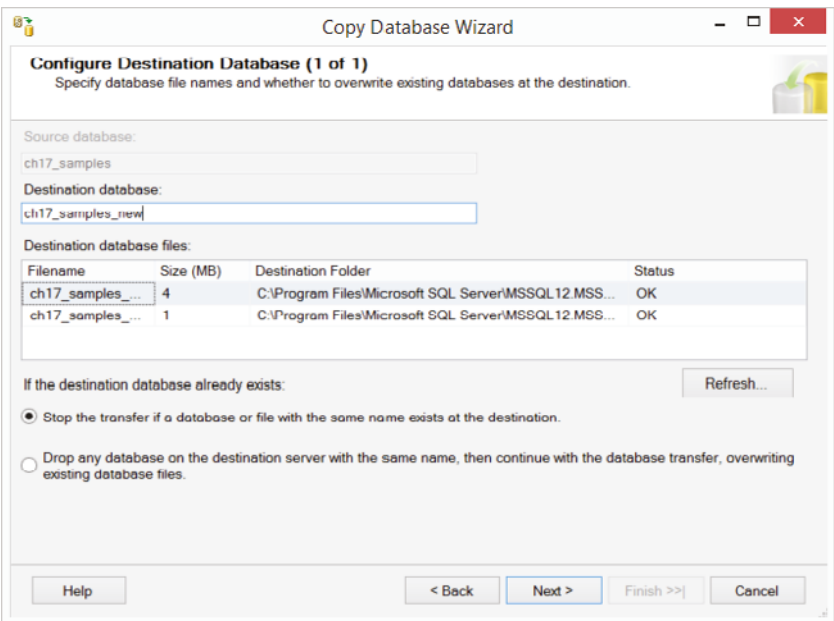
Rysunek 17.14. Wybieranie sposobu przesyłania danych

5. Na stronie *Select Databases* (patrz rysunek 17.15) wybierz kopiowanie bazy danych. Możesz albo skopiować, albo przenieść zaznaczone bazy. Tę opcję można ustawić dla każdej bazy osobno, co pozwala przenieść niektóre bazy i skopiować inne. Na rysunku 17.15 ustawiono kopiowanie bazy `ch17_samples`. Kliknij przycisk *Next*.



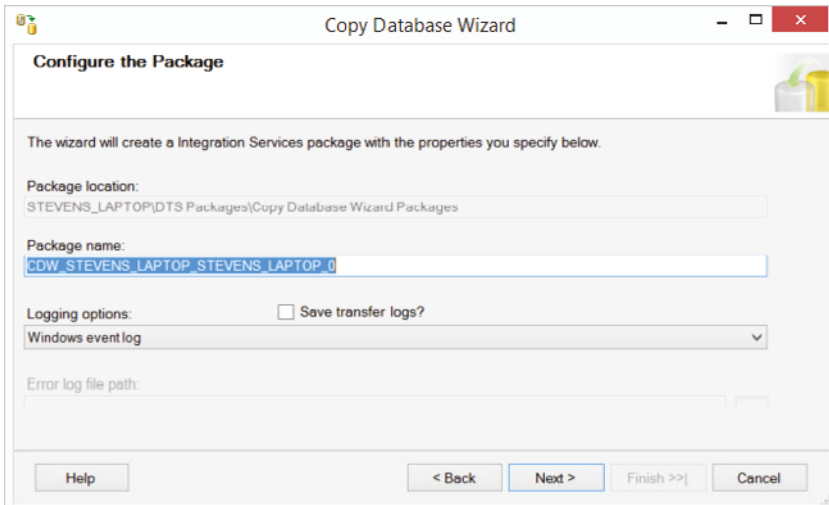
Rysunek 17.15. Wybieranie kopiowanych lub przenoszonych baz

6. Na stronie *Configure Destination Database* (patrz rysunek 17.16) podaj nazwę nowej bazy danych i określ, jakie pliki mają zostać utworzone. Domyślnie wartości te są automatycznie ustawiane na podstawie informacji ze źródłowej bazy danych. Na tej stronie dostępne są też inne opcje, które pozwalają na określenie, co zrobić, jeśli docelowa baza już istnieje. Kliknij przycisk *Next*.



Rysunek 17.16. Określanie ustawień docelowej bazy danych

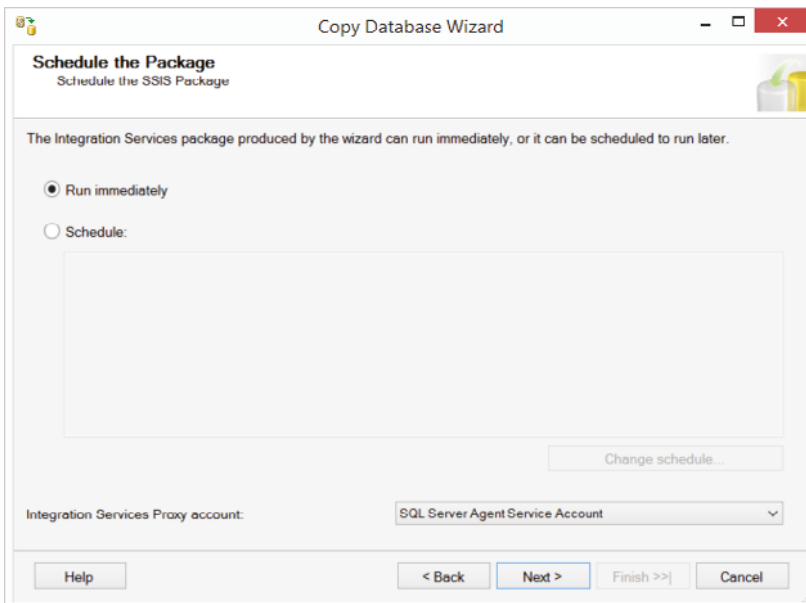
7. Skonfiguruj pakiet SSIS na potrzeby kopiowania bazy danych, co przedstawiono na rysunku 17.17. Kliknij przycisk *Next*.



The screenshot shows the 'Configure the Package' window of the 'Copy Database Wizard'. The window title is 'Copy Database Wizard'. The main heading is 'Configure the Package'. Below the heading, it says 'The wizard will create a Integration Services package with the properties you specify below.' The 'Package location' is 'STEVENS_LAPTOP\DTS Packages\Copy Database Wizard Packages'. The 'Package name' is 'CDW_STEVENS_LAPTOP_STEVENS_LAPTOP_0'. The 'Logging options' section has a checkbox for 'Save transfer logs?' which is unchecked, and a dropdown menu set to 'Windows event log'. The 'Error log file path' is empty. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

Rysunek 17.17. Konfigurowanie pakietu SSIS

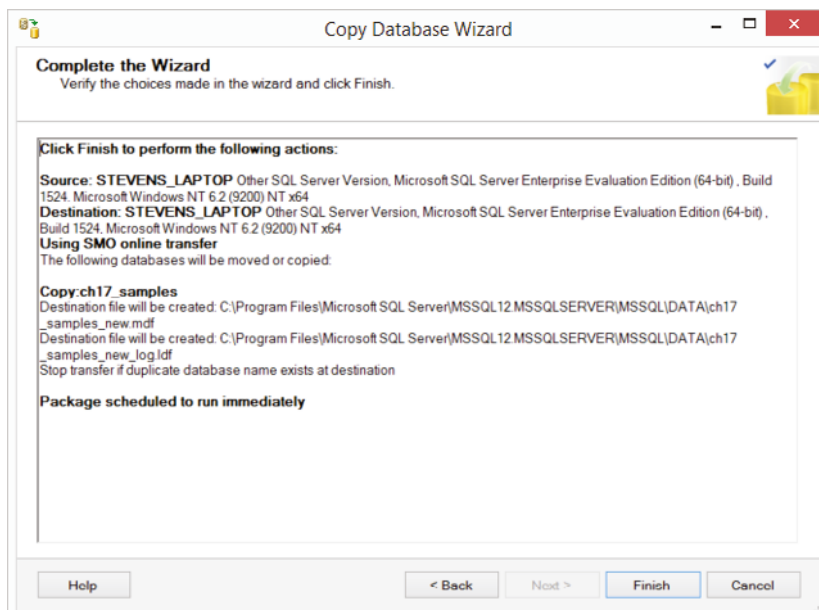
8. Teraz jesteś gotowy, aby zaplanować uruchamianie pakietu. Możesz wywołać pakiet natychmiast lub zaplanować jego późniejsze wykonanie. Na ekranie widocznym na rysunku 17.18 wybrana jest opcja, która powoduje natychmiastowe uruchomienie pakietu. Kliknij przycisk *Next*.



The screenshot shows the 'Schedule the Package' window of the 'Copy Database Wizard'. The window title is 'Copy Database Wizard'. The main heading is 'Schedule the Package' with the subtitle 'Schedule the SSIS Package'. Below the heading, it says 'The Integration Services package produced by the wizard can run immediately, or it can be scheduled to run later.' There are two radio buttons: 'Run immediately' (selected) and 'Schedule:'. Below the 'Schedule:' radio button is a large empty rectangular box. To the right of this box is a button labeled 'Change schedule...'. At the bottom, the 'Integration Services Proxy account:' is set to 'SQL Server Agent Service Account'. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

Rysunek 17.18. Planowanie wykonywania pakietu SSIS

9. Na stronie *Complete the Wizard* (patrz rysunek 17.19) możesz przed uruchomieniem pakietu sprawdzić, czy ustawione są właściwe opcje.



Rysunek 17.19. Przegląd ustawionych opcji

10. Kliknij przycisk *Finish*, aby uruchomić pakiet i skopiować wybrane bazy oraz obiekty z serwera źródłowego do docelowego. W trakcie pracy kreator wyświetla status każdego kroku. Po zakończeniu działania kreator informuje o wynikach wykonania pakietu.

UWAGA Upewnij się, że na docelowym serwerze działa usługa SQL Server Agent. Jeśli nawet pakiet jest wykonywany natychmiast, zostaje uruchomiony za pomocą zadania usługi SQL Server Agent z docelowego serwera. Ten kreator różni się więc od kreatora importu i eksportu, który nie potrzebuje usługi SQL Server Agent do natychmiastowego wykonywania wygenerowanego pakietu.

Kompresja kopii zapasowej

Mechanizm kompresji kopii zapasowej umożliwia zapisanie plików kopii w skompresowanej postaci, a to skraca czas tworzenia kopii zapasowej i zmniejsza wielkość plików. Mniejsza liczba operacji wejścia-wyjścia (wynikająca z mniejszych plików) jest uzyskiwana kosztem większego obciążenia procesora spowodowanego kompresją.

Zwykle kompresja kopii zapasowej prowadzi do znacznego skrócenia czasu tworzenia takiej kopii. To, czy kompresja jest przydatna, zależy od dostępności czasu procesora i zasobów wejścia-wyjścia, a także stopnia kompresji. Aby ocenić swój system, należy w monitorze wydajności sprawdzić następujące liczniki.

- Liczniki monitora wydajności systemu Windows dotyczące dysków.
- Liczniki systemu SQL Server:

- SQLServer: Backup Device: Device Throughput Bytes/sec,
- SQLServer: Databases: Backup/Restore Throughput/sec.

UWAGA Więcej informacji na temat monitorowania liczników wydajności znajdziesz w rozdziałach 12., „Monitorowanie systemu SQL Server” i 13., „Dostrajanie wydajności kodu w języku T-SQL”.

Od wersji SQL Server 7 dla kopii zapasowych używany był format MTF (ang. *Microsoft Tape Format*). Ten sam format jest stosowany dla kopii zapasowych systemu operacyjnego Windows. Dzięki temu kopie zapasowe systemu SQL Server można było zapisywać na tym samym nośniku, co kopie zapasowe systemu Windows. Było to wygodne zwłaszcza dla użytkowników korzystających z jednego napędu taśmowego.

Od wersji SQL Server 2008 skompresowane kopie zapasowe są zapisywane w innym formacie, niezgodnym z kopiami zapasowymi systemu Windows. Prowadzi to do pewnych ograniczeń związanych z kompresją kopii zapasowych. Oto te ograniczenia.

- Skompresowanych i nieskompresowanych kopii zapasowych nie można przechowywać w tej samej grupie nośników.
- Wcześniejsze wersje systemu SQL Server nie mogą wczytywać skompresowanych kopii zapasowych.
- Kopie zapasowe systemu Windows N i skompresowane kopie zapasowe nie mogą być przechowywane w tej samej grupie nośników.

Jeśli naruszysz jedno z tych ograniczeń, system SQL Server zwróci błąd.

Nowa opcja konfiguracyjna z poziomu serwera pozwala określić, czy kopie zapasowe mają być domyślnie kompresowane. Opcję można ustawić za pomocą procedury `sp_configure` lub w programie SQL Server Management Studio. Domyślnie kompresja kopii zapasowych jest wyłączona. Aby zmienić to ustawienie, użyj opcji `with compression` (lub `with no_compression`) w poleceniu `backup` w języku T-SQL. Możesz też wykorzystać okno dialogowe *Backup* w programie SQL Server Management Studio lub kreator planu konserwacji bazy danych.

Poziom kompresji zależy od kilku czynników.

- Skompresowanych baz danych nie da się dodatkowo skompresować w kopii zapasowej.
- Zaszyfrowane dane są kompresowane w mniejszym stopniu niż niezaszyfrowane.
- Tekstowe typy danych są kompresowane w większym stopniu niż inne typy.
- Wyższy poziom kompresji można uzyskać, gdy na stronie znajduje się wiele wierszy o identycznych wartościach w danej kolumnie.

Modele odzyskiwania

Zrozumienie modeli odzyskiwania jest niezbędne, kiedy chcesz opracować skuteczną strategię tworzenia kopii zapasowych. Używany model wpływa na to, w jaki sposób system SQL Server zarządza dziennikami transakcji. Model należy dostosować do planu tworzenia kopii zapasowej i przywracania bazy danych.

W modelu *FULL* w dzienniku transakcji rejestrowane są wszystkie zmiany w danych, co pozwala zastosować dowolny mechanizm odzyskiwania bazy i zapewnia najwyższą ochronę danych, przy czym dziennik transakcji zajmuje wtedy najwięcej pamięci. Ten model można zastosować dla wszystkich operacji tworzenia kopii zapasowej. Umożliwia on odzyskanie stanu z danego punktu w czasie i tworzenie kopii zapasowej dziennika transakcji. Model *FULL* należy stosować dla większości produkcyjnych systemów OLTP i aplikacji o znaczeniu strategicznym, w których niezbędne jest ograniczenie utraty danych.

UWAGA Pierwszą kopię zapasową dziennika transakcji można utworzyć dopiero po zapisaniu pełnej kopii zapasowej bazy danych.

Model *BULK_LOGGED* związany jest z rejestrowaniem minimalnej ilości danych dotyczących wybranych operacji na bazie (w tym operacji importu masowego). Oto niektóre z tych operacji:

- BCP,
- BULK INSERT,
- SELECT INTO,
- CREATE INDEX,
- ALTER INDEX REBUILD,
- DBCC DBREINDEX.

W tym modelu zamiast wszystkich zmian związanych z tymi operacjami rejestrowane są tylko alokacje ekstentów i oznaczone zostają zmodyfikowane eksteny. Przyspiesza to wykonywanie operacji (ponieważ rejestrowana jest minimalna ilość informacji), jednak grozi utratą danych w trakcie ich odzyskiwania. Tworzona jest kopia zapasowa dziennika transakcji, po czym system sprawdza eksteny oznaczone jako zmodyfikowane i kopiuje je z plików danych do kopii zapasowej dziennika. Ponadto po operacjach wykonywanych w modelu *BULK_LOGGED* nie jest możliwe odzyskiwanie stanu od punktu w czasie za pomocą kopii zapasowej dziennika transakcji.

Oto przykładowy scenariusz, w którym dla bazy stosowany jest model *BULK_LOGGED*. Pełną kopię zapasową utworzono o 13.00. Kopia zapasowa dziennika transakcji została zapisana o 13.30. Następnie wykonano operację z rejestracją minimalnych informacji. O 14.00 nastąpiła awaria fizycznych dysków z plikami danych. Nie da się przywrócić bazy do stanu z 14.00, ponieważ kopia zapasowa dziennika transakcji wymaga dostępu do plików danych w celu pobrania modyfikacji wprowadzonych przez operacje z rejestracją minimalnych informacji. Nie można więc przywrócić kopii zapasowej końcówki dziennika. Dane zostają utracone i można przywrócić bazę do stanu z 13.30.

Przeprowadzanie operacji masowych w tym trybie grozi utratą danych. Są one bezpieczne dopiero po utworzeniu kompletnej kopii zapasowej dziennika, która jest zapisywana po zakończeniu danej operacji. W niektórych operacjach można zminimalizować utratę danych dzięki zastosowaniu krótszych transakcji, co pozwala tworzyć kopie zapasowe dziennika transakcji w trakcie wykonywania operacji z rejestracją minimalnych informacji i bezpośrednio po ich zakończeniu. Opisany model często stosowany jest wtedy, gdy administrator wykonuje operacje z rejestrowaniem minimalnych informacji, a po ich zakończeniu wraca do trybu tworzenia pełnych kopii zapasowych. Pomaga to zwiększyć wydajność operacji masowych. Ten model często używany jest też w systemach OLAP i w bazach stosowanych do generowania raportów, w których co noc wykonuje się operacje masowego wczytywania danych. Po wczytaniu danych można utworzyć kopię zapasową, a w ciągu dnia baza nie jest modyfikowana. Jeśli zatem z powodu awarii nastąpi utrata danych, można je przywrócić za pomocą kopii zapasowej.

W modelu *SIMPLE*, podobnie jak w modelu *BULK_LOGGED*, zapisywana jest niewielka ilość informacji, przy czym rekordy z dziennika transakcji są przechowywane tylko do momentu utworzenia następnego punktu kontrolnego, kiedy to niezatwierdzone zmiany są wprowadzane w plikach danych. Następnie w ramach tworzenia punktu kontrolnego dziennik transakcji jest przycinany. W tym modelu nie są zapisywane kopie zapasowe dziennika transakcji, co uniemożliwia przywrócenie stanu do określonego punktu w czasie.

Ten model odzyskiwania jest stosowany zwykle na serwerach rozwojowych i testowych, gdzie utrata danych jest akceptowalna, ponieważ zawsze można wczytać je ponownie. Czasem ten model wykorzystuje się w systemach OLAP i bazach używanych do generowania raportów, w których w nocy wczytywane są dane, po czym następuje utworzenie pełnej lub różnicowej kopii zapasowej. Jeśli w tym modelu w bazie wystąpią problemy w trakcie wczytywania danych, cały proces trzeba zacząć od początku (chyba że w trakcie wczytywania utworzono pełną lub różnicową kopię zapasową). Jeśli ponadto administrator bazy danych przełączy bazę z innego modelu przywracania na model *SIMPLE*, zerwana zostanie ciągłość dziennika transakcji, ponieważ ten tryb powoduje przycinanie dziennika. W czasie, gdy baza pracuje w tym modelu, istnieje większe zagrożenie utratą danych.

UWAGA W modelu *SIMPLE* nie można stosować replikacji transakcyjnej, przysyłać dzienników ani tworzyć kopii lustrzanych, ponieważ nie istnieje trwały magazyn transakcji.

Wybór modelu odzyskiwania

To, który model jest najlepszy w danej sytuacji, zależy od akceptacji utraty danych, standardowo wykonywanych operacji odczytu i zapisu oraz znaczenia bazy w codziennym funkcjonowaniu firmy. Oto rekomendacje dotyczące poszczególnych modeli.

Model FULL

Ten model stosuj dla baz danych o znaczeniu strategicznym, aby zminimalizować utratę danych dzięki rejestrowaniu wszystkich operacji w dzienniku. Jeśli pliki danych zostaną uszkodzone, można wykorzystać dziennik transakcji z niezapisanymi operacjami i przywrócić bazę do danego punktu w czasie. Dlatego w produkcyjnych systemach OLTP zwykle stosuje się właśnie model *FULL* (chyba że dane są modyfikowane tylko w nocy, co czasem ma miejsce w systemach OLAP i bazach używanych do generowania raportów).

Model BULK_LOGGED

Ten model możesz wykorzystać, aby zwiększyć wydajność operacji masowych (dzięki temu, że rejestrowanych jest bardzo niewiele informacji). Możesz np. wykonywać w nocy operacje masowe, a następnie ponownie włączać model *FULL*. W modelu *BULK_LOGGED* zachowywany jest pełen dziennik (tak jak w modelu *FULL*), przy czym nie dotyczy to operacji masowych. Dlatego możesz stosować ten model na stałe, choć grozi to utratą danych. O ile nie są wykonywane operacje masowe, administrator może utworzyć kopię zapasową dziennika transakcji. Jednak często administrator może nie zdawać sobie sprawy z tego, że z powodu wykonania operacji masowej przywrócenie stanu z wykorzystaniem kopii zapasowej końcówki dziennika transakcji nie jest wykonalne.

Aby zabezpieczyć się przed wykonywaniem przez kogoś operacji masowych bez tworzenia kopii zapasowej i ograniczyć ryzyko utraty danych, powinieneś włączać model `BULK_LOGGED` tylko na czas wykonywania takich operacji. Ten model można stosować na stałe w systemach OLAP i bazach używanych do generowania raportów, w których w ciągu dnia dane nie są modyfikowane. W tej sytuacji wystarczy utworzyć kopię zapasową bazy po conocnym wczytywaniu danych, aby ograniczyć ryzyko utraty informacji. W ciągu dnia dane nie są zagrożone, ponieważ się nie zmieniają. Ponadto czasem utrata części danych jest akceptowalna, ponieważ bazy z systemu OLAP lub bazy do generowania raportów można, jeśli trzeba, wczytać ze źródła danych z systemu OLTP.

Model SIMPLE

W tym modelu dziennik transakcji nie jest zachowywany — zostaje przycięty w procesie tworzenia punktu kontrolnego. Dlatego nie trzeba zarządzać dziennikiem transakcji. Model jest często stosowany w systemach rozwojowych, przeznaczonych tylko do odczytu, i testowych, w których kopie zapasowe dziennika transakcji nie są potrzebne. Jeśli w takich systemach dane zostaną utracone, można wczytać nową wersję danych ze źródła danych ze środowiska OLTP. Jeżeli administrator bazy danych włączy ten model odzyskiwania zamiast innego, ciągłość dziennika transakcji zostanie naruszona, ponieważ nie można utworzyć jego kopii zapasowej. W tym modelu nie jest możliwe przywracanie bazy do stanu z określonego punktu w czasie, ponieważ administrator nie może utworzyć kopii zapasowej dziennika. Dlatego przy odzyskiwaniu trzeba zastosować pełne i różnicowe kopie zapasowe.

Przełączanie modeli odzyskiwania

System SQL Server zapewnia pełną swobodę w zakresie przełączania modeli odzyskiwania. Warto jednak znać ograniczenia związane ze zmianą tych modeli, ponieważ może ona prowadzić do utraty danych. Ograniczenia związane z przełączaniem modeli odzyskiwania opisujemy niżej.

- **Zmiana z modelu FULL na BULK_LOGGED.** Jeśli wykonywane mogą być operacje powodujące rejestrowanie minimalnych informacji, zaleca się utworzenie przynajmniej kopii zapasowej dziennika transakcji, co pozwoli administratorowi przywrócić dane z ostatniego dziennika transakcji (jeżeli końcówka dziennika transakcji jest niedostępna). Aby zmienić model odzyskiwania w ten sposób, wywołaj następujące polecenie:

```
ALTER DATABASE ch17_samples SET RECOVERY BULK_LOGGED
```

- **Zmiana z modelu FULL na SIMPLE.** Ponieważ przy tej zmianie może nastąpić zerwanie ciągłości dziennika transakcji, zaleca się, aby wcześniej przynajmniej utworzyć kopię zapasową tego dziennika. Ta zmiana modelu powoduje, że nie jest możliwe tworzenie kopii zapasowej dziennika transakcji ani odzyskiwanie stanu do określonego punktu w czasie. Aby przełączyć model, wywołaj poniższą instrukcję:

```
ALTER DATABASE ch17_samples SET RECOVERY SIMPLE
```

- **Zmiana z modelu BULK_LOGGED na FULL.** Ponieważ możliwe, że wykonywane były operacje z rejestrowaniem minimalnych informacji, to aby ograniczyć utratę danych, w sytuacji gdy niedostępna jest końcówka dziennika transakcji, po zmianie modelu zaleca się utworzenie kopii zapasowej dziennika transakcji. Aby wykonać tę zmianę modeli, wywołaj poniższą instrukcję:

```
ALTER DATABASE ch17_samples SET RECOVERY FULL
```

- **Zmiana z modelu BULK_LOGGED na SIMPLE.** W modelu SIMPLE występuje zwiększone ryzyko utraty danych z powodu awarii bazy, dlatego zaleca się, aby przed przełączeniem modeli utworzyć przynajmniej kopię zapasową dziennika transakcji. Aby przeprowadzić tę zmianę modeli, wywołaj następujące polecenie:

```
ALTER DATABASE ch17_samples SET RECOVERY SIMPLE
```

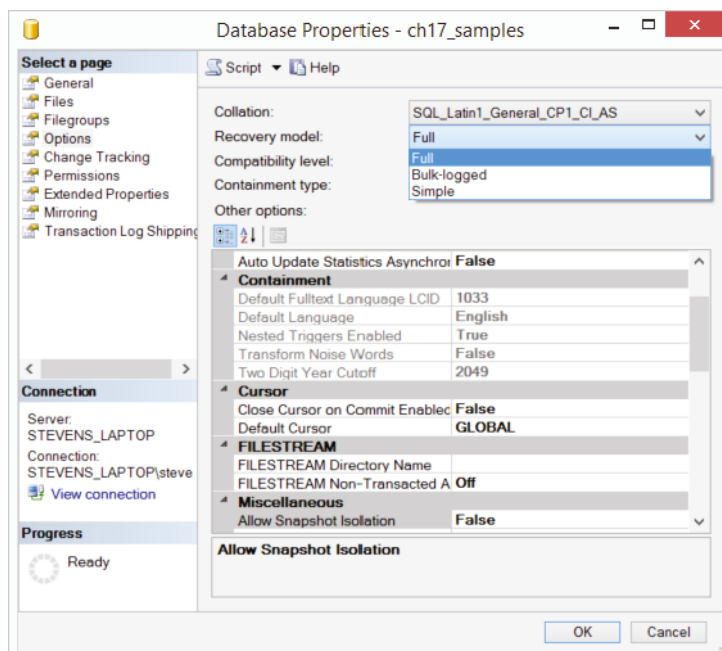
- **Zmiana modelu SIMPLE na FULL.** Aby umożliwić obsługę kopii zapasowej dziennika transakcji w modelu FULL, po przełączeniu trzeba utworzyć kopię zapasową: pełną, różnicową, pliku lub grupy plików. Żeby wykonać tę zmianę modeli, uruchom następującą instrukcję:

```
ALTER DATABASE ch17_samples SET RECOVERY FULL
```

- **Zmiana modelu SIMPLE na BULK_LOGGED.** Aby umożliwić stosowanie kopii zapasowych dziennika transakcji w modelu BULK_LOGGED, po przełączeniu modelu trzeba utworzyć kopię zapasową: pełną, różnicową, pliku lub grupy plików. W celu zmiany modeli wywołaj następującą instrukcję:

```
ALTER DATABASE ch17_samples SET RECOVERY BULK_LOGGED
```

Model odzyskiwania jest ustawiany dla każdej bazy. Zmienić model można też w programie SQL Server Management Studio. W tym celu należy otworzyć właściwości bazy danych i wybrać zakładkę *Options*, co pokazano na rysunku 17.20.



Rysunek 17.20. Zmienianie modelu odzyskiwania w programie SQL Server Management Studio

Tworzenie tabel z historią kopii zapasowych

System SQL Server przechowuje historię kopii zapasowych w bazie msdb. Używana jest do tego grupa tabel, na których podstawie można zidentyfikować kopie zapasowe dostępne dla baz. System SQL Server wyświetla dostępne kopie zapasowe w oknie dialogowym *Restore*. Oto używane tabele.

- **Backupfile.** Zawiera wiersz dla każdego pliku danych i dziennika, dla którego istnieje kopia zapasowa.
- **Backupfilegroup.** Zawiera wiersz dla każdej grupy plików w zestawie kopii zapasowych.
- **Backupmediafamily.** Zawiera wiersz dla każdej rodziny nośników.
- **Backupmediaset.** Zawiera wiersz dla każdego zestawu nośników.
- **Backupset.** Zawiera wiersz dla każdego zestawu kopii zapasowych.

UWAGA *Zestaw nośników* (ang. *media set*) to uporządkowana kolekcja wszystkich taśm lub dysków ze wszystkich urządzeń uczestniczących w tworzeniu kopii zapasowej. *Rodzina nośników* (ang. *media family*) to kolekcja wszystkich nośników z kopiami zapasowymi w jednym urządzeniu biorącym udział w tworzeniu kopii zapasowej. *Nośnik z kopią zapasową* to urządzenie taśmowe lub dyskowe używane do zapisu kopii zapasowej.

Poniższe trzy instrukcje zwracają informacje z tabel z historią kopii zapasowych.

- **RESTORE FILELISTONLY.** Zwraca listę plików bazy danych i dziennika z zestawu kopii zapasowych z tabeli backupfile:
`RESTORE FILELISTONLY FROM ch17_samples_Backup`
- **RESTORE HEADERONLY.** Zwraca wszystkie informacje nagłówkowe dla wszystkich zestawów kopii zapasowych z urządzenia z tabeli backupset:
`RESTORE HEADERONLY FROM ch17_samples_Backup`
- **RESTORE LABELONLY.** Zwraca informacje o nośniku z kopią zapasową z urządzenia z tabeli backupmediaset:
`RESTORE LABELONLY FROM ch17_samples_Backup`

UWAGA Instrukcje **RESTORE HEADERONLY**, **RESTORE FILELISTONLY**, **RESTORE VERIFYONLY** i **RESTORE LABELONLY** nie przywracają żadnych danych, dlatego można je bezpiecznie wywoływać. Polecenia te jedynie zwracają informacje, co jest nieintuicyjne. Pamiętaj, aby zwracać uwagę na kontekst wywoływania instrukcji **RESTORE**.

Uprawnienia potrzebne do tworzenia kopii zapasowych i przywracania stanu

System SQL Server udostępnia precyzyjne uprawnienia do tworzenia kopii zapasowych i przywracania stanu baz danych. Użytkownikowi lub grupie z uwierzytelnieniami systemu Windows albo SQL Server można przyznać uprawnienia do tych operacji. Aby użytkownik mógł utworzyć kopię zapasową bazy danych, musi mieć przynajmniej następujące uprawnienia:

- **rola serwera** — none,
- **rola bazy danych** — db_backupoperator.

Na potrzeby przywracania bazy danych użytkownik musi mieć przynajmniej następujące uprawnienia:

- **rola serwera** — dbcreator,
- **rola bazy danych** — db_owner.

Tworzenie kopii zapasowych systemowych baz danych

Systemowe bazy danych w systemie SQL Server są niezbędne do jego działania. Bazy te nie są często modyfikowane, ale zawierają ważne informacje, które trzeba zapisać w kopii zapasowej. Po utworzeniu nowego egzemplarza systemu SQL Server przygotuj plan tworzenia kopii zapasowych, który obejmuje wykonanie pełnej kopii wszystkich systemowych baz danych oprócz bazy tempdb. System SQL Server przy każdym uruchomieniu tworzy bazę tempdb od nowa, ponieważ nie zawiera ona żadnych danych, które trzeba przywrócić. Tworzenie kopii zapasowej systemowych baz danych trwa tylko kilka minut, dlatego nie ma powodów, dla których można pominąć to zadanie. Często planuje się wykonywanie takich kopii co noc. Dodatkowo warto tworzyć kopie zapasowe na lokalnym dysku twardym przed wprowadzeniem zmian i po ich wykonaniu. Pozwala to zapewnić bezpieczeństwo do czasu wykonania następnej nocy kolejnej standardowej kopii zapasowej.

Baza master

Baza master zawiera informacje o loginach i metadane na temat każdej bazy z danego egzemplarza systemu SQL Server. Ponadto znajdują się tu informacje o konfiguracji tego egzemplarza.

Baza jest modyfikowana po wykonaniu każdej z poniższych operacji:

- dodaniu, usunięciu lub zmodyfikowaniu ustawienia z poziomu bazy danych,
- dodaniu lub usunięciu bazy użytkownika,
- dodaniu lub usunięciu pliku albo grupy plików w bazie użytkownika,
- dodaniu, usunięciu lub zmodyfikowaniu zabezpieczeń dla loginu,
- zmodyfikowaniu konfiguracji systemu SQL Server na poziomie serwera,
- dodaniu lub usunięciu logicznego urządzenia z kopią zapasową,
- skonfigurowaniu kwerend rozproszonych lub procedur RPC,
- dodaniu, zmodyfikowaniu lub usunięciu loginu z serwera połączanego lub loginu zdalnego.

Choć zmiany tego rodzaju są wykonywane rzadko, po ich wprowadzeniu rozważ utworzenie pełnej kopii zapasowej bazy danych. Jeśli jej nie utworzysz, narażasz się na utratę modyfikacji w momencie przywrócenia starszej kopii zapasowej bazy master. Ponadto warto prewencyjnie tworzyć kopie zapasowe tej bazy przed dodaniem pakietów SP lub zainstalowaniem poprawek i po wykonaniu takiej operacji.

Baza msdb

Baza msdb zawiera zadania narzędzia SQL Server Agent, zadania tworzenia kopii zapasowej, harmonogramy, operatorów oraz historię tworzenia kopii zapasowych i przywracania. Może też obejmować pakiety SSIS i inne elementy. Gdy dodasz nowe zadanie lub pakiet SSIS, a baza msdb ulegnie awarii, wcześniejsza kopia zapasowa nie będzie zawierać nowych elementów i trzeba będzie je ponownie utworzyć.

Baza tempdb

Nie należy tworzyć kopii zapasowej bazy tempdb. Ponieważ ta baza jest tworzona od nowa przy każdym uruchomieniu systemu SQL Server, nie zawiera danych, które trzeba przywracać.

Baza model

Baza model jest modyfikowana jeszcze rzadziej niż pozostałe systemowe bazy danych. Jest to szablonowa baza używana przy tworzeniu nowych baz danych. Jeśli chcesz, aby w każdej nowej bazie danych znajdował się określony obiekt (np. procedura składowana lub tabela), umieść go w bazie model. Po dodaniu takich obiektów należy utworzyć kopię zapasową tej bazy. Jeśli tego nie zrobisz, modyfikacje zostaną utracone i bazę model trzeba będzie zbudować od nowa. Ponadto warto zachować skrypty z wszystkimi zmianami wprowadzanymi w bazie model. Jest to dodatkowa warstwa zabezpieczeń.

Kopie zapasowe indeksów pełnotekstowych

Wyszukiwanie pełnotekstowe umożliwia szybkie przeszukiwanie nieustrukturyzowanych danych za pomocą słów kluczowych opartych na słowach z określonego języka. Technika służy głównie do przeszukiwania pól z danymi typów char, nchar, varchar i nvarchar. Przed przeszukiwaniem trzeba utworzyć indeks pełnotekstowy. Służy do tego proces zapewniania indeksu — dane są rozbijane na słowa kluczowe, które zostają umieszczone w indeksie. Każdy indeks pełnotekstowy jest zapisywany w katalogu pełnotekstowym, a sam katalog zostaje zachowany w grupie plików.

W systemie SQL Server 2014 pełna kopia zapasowa obejmuje indeksy pełnotekstowe. W systemie SQL Server 2005 indeksy pełnotekstowe były częścią katalogu zapisanego w grupie plików; miały określoną ścieżkę fizyczną i były traktowane jak zwykły plik bazy danych. System SQL Server 2014 traktuje cały katalog jak obiekt wirtualny obejmujący kolekcję indeksów pełnotekstowych. W kontekście tworzenia kopii zapasowych takie indeksy nie różnią się od innych indeksów. Aby utworzyć kopię zapasową wszystkich indeksów pełnotekstowych, należy ustalić, w jakich plikach są one zapisane, a następnie wykonać kopię tych plików (lub grup plików).

Kopie zapasowe indeksów pełnotekstowych są w pełni zintegrowane z ogólną architekturą tworzenia kopii zapasowych. Indeksy pełnotekstowe możesz umieścić w odrębnych grupach plików i w grupie głównej albo pozostawić je w grupie plików bazowej tabeli. Dzięki temu zarządzanie takimi indeksami jest łatwiejsze niż we wcześniejszych wersjach systemu.

Sprawdzanie poprawności obrazów z kopią zapasową

W każdym mechanizmie tworzenia kopii zapasowych bardzo istotne jest sprawdzenie poprawności obrazów z kopią zapasową. Administrator bazy danych może sumiennie tworzyć kopie zapasowe, jeśli jednak baza danych jest uszkodzona, każda jej kopia też będzie nieużyteczna. Dlatego okresowo przywracaj bazy z zapisanych kopii, aby określić ich poprawność. Ponadto sprawdzaj spójność baz danych w celu zbadania ich struktury. Za pomocą instrukcji RESTORE VERIFYONLY z języka T-SQL można sprawdzić poprawność obrazu z kopią zapasową. Ta instrukcja nie przywraca kopii zapasowej, ale sprawdza jej stan, wykonując takie operacje jak:

- sprawdzanie, czy zbiór kopii zapasowej jest kompletny i możliwy do odczytania,
- sprawdzanie wybranych pól nagłówkowych na stronach bazy danych,
- sprawdzanie sum kontrolnych, jeśli kopię zapasową utworzono z opcją WITH CHECKSUMS,
- sprawdzanie, czy docelowe urządzenia mają wystarczająco dużo wolnego miejsca.

Polecenie RESTORE VERIFYONLY nie daje jednak pełnej gwarancji, że kopię zapasową można będzie przywrócić. Dlatego trzeba losowo przywracać kopie zapasowe na serwerze testowym.

Polecenie `RESTORE VERIFYONLY` to tylko dodatkowy poziom sprawdzania poprawności. Oto składnia tej instrukcji:

```
RESTORE VERIFYONLY FROM <nazwa_urządzenia_z_kopią_zapasową>
```

Oto przykładowy komunikat z wynikiem:

The backup set on file 1 is valid.

UWAGA Polecenie `RESTORE VERIFYONLY` nie działa dla snapshotów baz danych. Jeśli planujesz przywrócić bazę na podstawie snapshota, do sprawdzenia jego stanu użyj instrukcji `DBCC CHECKDB`.

Aby zapewnić wyższą niezawodność i zabezpieczyć się przed sytuacją, w której niesprawne urządzenie uniemożliwia przywrócenie kopii zapasowej, użyj kopii lustrzanych z rezerwowym zestawem kopii zapasowych. Możesz wykorzystać do tego dysk lub taśmę. Obowiązują przy tym następujące ograniczenia.

- Urządzenia z kopiami zapasowymi muszą być identyczne.
- Aby utworzyć nową kopię zapasową (lub rozbudować istniejącą), lustrzany zestaw kopii zapasowych musi być nienaruszony. W przeciwnym razie nie można użyć zestawu nośników.
- Przy przywracaniu bazy z zestawu nośników używane może być tylko jedno z urządzeń z kopiami lustrzanymi.
- Jeśli jednak kopia lustrzana zestawu nośników jest uszkodzona, nie można tworzyć dodatkowych kopii lustrzanych dla tego zestawu.

Poniższa instrukcja tworzy kopię lustrzaną kopii zapasowej bazy `ch17_samples` (plik *KodR17.sql*):

```
BACKUP DATABASE ch17_samples
TO DISK = 'x:\ch17_samples\backup\ch17_samples.bak'
MIRROR TO DISK = 'y:\ch17_samples\backup_mirror\ch17_samples.bak'
WITH FORMAT, MEDIANAME='ch17_sample_mirror'
```

Przebieg przywracania

Przywracanie pozwala odtworzyć bazę danych po awarii. Jest to ważna funkcja transakcyjnych systemów zarządzania relacyjnymi bazami danych. Gdy administrator przywraca bazę danych, odbywa się to w trzech etapach. Oto one:

- kopiowanie,
- ponawianie operacji,
- wycofywanie operacji.

Na *etapie kopiowania* na dysku tworzony i inicjowany jest obraz bazy danych, po czym odtwarzana jest pełna kopia zapasowa. Następnie można wykorzystać kopię różnicową lub dziennika transakcji. Służy do tego instrukcja `RESTORE` języka T-SQL.

PRZYWRACANIE A ODZYSKIWANIE

Pojęcia przywracanie (ang. *restore*) i odzyskiwanie (ang. *recovery*) są często mylone i błędnie używane. Przywracanie to operacja wykonywana po wywołaniu polecenia `RESTORE` języka T-SQL w celu odtworzenia bazy danych.

Odzyskiwanie związane jest z doprowadzaniem bazy do spójnego stanu. Zatwierdzone transakcje są wtedy wykonywane na dysku (etap ponawiania operacji), a rozpoczęte, ale niezatwierdzone transakcje zostają anulowane (etap wycofywania operacji). W efekcie w bazie uwzględniane są tylko zatwierdzone transakcje, zatem znajduje się ona w spójnym stanie.

Przy każdym uruchomieniu serwera następuje automatyczne odzyskiwanie. W momencie rozruchu systemu SQL Server nie wiadomo, dlaczego ostatnim razem zakończył on pracę. Możliwe, że został poprawnie zamknięty, ale mógł też przestać działać z powodu przerwy w zasilaniu. Jeśli przyczyną był brak zasilania, niektóre rozpoczęte transakcje mogły nie zostać dokończone. W ramach odzyskiwania zatwierdzone transakcje są ponawiane, a niezatwierdzone — wycofywane.

Ostatni etap w procesie przywracania to właśnie odzyskiwanie. Polecenie `RESTORE` pozwala określić, kiedy należy wykonać odzyskiwanie. Jeśli ten etap nie został wykonany, można przywrócić bazę na podstawie różnicowej kopii zapasowej lub kontynuować przywracanie dzienników transakcji. Po odzyskiwaniu nie można przywracać danych na podstawie dziennika, a baza danych przechodzi w tryb online.

Po przywróceniu pełnej kopii zapasowej oraz kopii różnicowej i dziennika transakcji administrator bazy danych uruchamia odzyskiwanie. System SQL Server wykonuje wtedy ponawianie i wycofywanie operacji. Na *etapie ponawiania operacji* wykonywane są wszystkie zatwierdzone transakcje z dziennika, które nie zostały uwzględnione w plikach z danymi. Protokół WAL gwarantuje, że zatwierdzone transakcje zostały zapisane na stabilnym nośniku z dziennikiem transakcji. Następnie, na *etapie ponawiania operacji*, system SQL Server sprawdza rekordy z dziennika transakcji i wprowadza odpowiednie modyfikacje w plikach danych bazy.

Czas trwania tego etapu zależy od ilości modyfikacji w danych wprowadzonych w systemie SQL Server. To z kolei zależy od tego, co system robił w momencie awarii i jaki interwał odzyskiwania jest ustawiony. Jeśli np. system SQL Server właśnie skończył aktualizowanie 10 milionów wierszy tabeli i zatwierdził transakcję, po czym nieoczekiwanie zakończył pracę, w trakcie odzyskiwania będzie musiał ponownie zmodyfikować wszystkie te dane. Interwał odzyskiwania wpływa na to, ile niezatwierdzonych stron jest przechowywanych w pamięci przed zapisaniem ich na stabilnym nośniku w ramach procesu tworzenia punktu kontrolnego. Domyślnie ten interwał jest ustawiony na 0. Przy tej wartości system SQL Server przechowuje niezapisane operacje z niecałej ostatniej minuty. Przy tym ustawieniu przed udostępnieniem bazy użytkownikom trzeba ponowić niewiele operacji. Im wyższa wartość interwału odzyskiwania, tym dłużej będzie ono trwało.

Po *etapie ponawiania operacji* następuje *etap wycofywania operacji*. Polega on na wycofaniu niedokończonych transakcji. Ten etap może trwać dość długo, a zależy to od zakresu operacji i długości transakcji wykonywanych w momencie zamknięcia systemu. Jeśli w tej chwili administrator bazy danych właśnie usuwał 10 milionów wierszy, system SQL Server w ramach odzyskiwania musi wycofać wszystkie te zmiany. System w trakcie wycofywania operacji udostępnia bazę użytkownikom, choć działa ona wtedy wolniej.

Przygotowania do odzyskiwania

Ważnymi zadaniami administratora baz danych, które pozwalają ograniczyć ryzyko i zakres utraty danych, są tworzenie kopii zapasowej i planowanie odzyskiwania. Należy opracować plan tworzenia kopii zapasowej, który zminimalizuje utratę danych i będzie możliwy do zrealizowania w czasie dostępnym na konserwację. Wybierz odpowiednie mechanizmy tworzenia kopii zapasowej dostępne w systemie SQL Server, aby przygotować pożądaną plan. Powinien on być zgodny z wymaganiami dotyczącymi akceptowalnej utraty danych i zapewnienia ciągłości pracy firmy. Ponadto opracuj procedurę tworzenia kopii zapasowych i codziennie sprawdzaj, czy działa prawidłowo. Powinieneś m.in. sprawdzać, czy możliwe jest poprawne przywrócenie kopii zapasowej bazy danych.

Możliwe, że firma regularnie tworzy kopie zapasowe i zakłada, iż posiada kopie potrzebne do przywrócenia bazy. Okazuje się jednak, że baza była uszkodzona i niektóre z najnowszych kopii zapasowych nie nadają się do przywrócenia. Takie sytuacje mogą pozostać niewykryte przez wiele miesięcy do momentu, gdy ktoś musi przywrócić bazę danych i zauważy, że jest to niemożliwe. Aby zmniejszyć ryzyko wystąpienia takiej sytuacji, sprawdzaj spójność wszystkich baz danych i opracuj proces testowania możliwości przywracania kopii zapasowych. Ponadto zapisuj kopie zapasowe baz danych w zdalnej lokalizacji, aby chronić dane przed lokalną katastrofą, a jednocześnie przechowuj lokalnie najnowsze kopie zapasowe, co pozwoli szybko je przywrócić.

Innym istotnym zadaniem jest planowanie odzyskiwania systemu po katastrofie. Jeśli centrum danych firmy zostanie całkowicie zniszczone, powinieneś szybko zbudować nowe centrum danych. Zadbaj o to, aby jak najbardziej ograniczyć utratę danych i zakłócenia w pracy organizacji. Zespół powinien co pewien czas symulować awarię centrum danych i w ramach ćwiczeń budować nowe centrum danych zgodnie z opracowanym planem.

Wymagania związane z odzyskiwaniem

Planowanie tworzenia kopii zapasowych należy zaczynać z myślą o ostatecznym celu — wymaganiach związanych z odzyskiwaniem systemu. Oto kilka aspektów, które warto rozważyć.

- Możliwe, że dostępna musi być tylko część bazy danych. W takiej sytuacji możesz przywracać dane fragment po fragmencie, zwłaszcza jeśli baza jest duża i jej pełne przywrócenie wymaga dużo czasu. Ustal, które części bazy muszą być dostępne, i rozmieść dane w grupach plików, co pozwoli Ci zacząć przywracanie od najważniejszych grup. Dane archiwalne i używane do generowania raportów są mniej istotne, dlatego można je przywrócić na końcu.
- Firma może zainstalować nowszy sprzęt z rezerwami mocy i zapewniającą wysoką dostępność macierz RAID, aby ograniczyć czas przestoju. Można też wykorzystać więcej wydajniejszych urządzeń z kopiami zapasowymi, co pozwoli na szybkie przywracanie baz danych.
- Ustal, jak łatwe będzie odtwarzanie utraconych danych z każdej bazy. W niektórych bazach można łatwo odtworzyć dane po pobraniu ich z innego systemu lub wczytaniu ze zwykłych plików. W bazach wspomagających podejmowanie decyzji do pobierania danych zwykle używa się narzędzi ETL (ang. *extract, transfer, load*). Gdy danych nie da się przywrócić, można wykorzystać takie narzędzia do ponownego wczytania informacji.
- Jaki jest akceptowalny czas przestoju po awarii napędu, np. dysku? Ponieważ dyski stają się coraz tańsze, większość firm umieszcza bazy danych na odpornych na błędy macierzach dyskowych. Ogranicza to ryzyko utraty danych w wyniku awarii jednego dysku.

Przykładowo w macierzach RAID 5 utrata jednego dysku prowadzi tylko do odczuwalnego spadku wydajności. Dopiero awaria drugiego dysku w tej samej macierzy skutkuje utratą danych. Aby ograniczyć ryzyko, można umieścić w macierzy dodatkowe dyski i podpisać umowę o gwarantowanym poziomie usług z dostawcą sprzętu, który będzie dostarczał i instalował potrzebne dyski. Oto inny często powtarzający się scenariusz: dział firmy używa bazy w środowisku ze sprzętem o niskiej jakości. Po pewnym czasie baza zyskuje strategiczne znaczenie w tym dziale, jednak administrator nic o niej nie wie. Administrator powinien podjąć próbę zlokalizowania wszystkich baz w firmie i przygotowania planu ich odzyskiwania.

- Ustal, które bazy są zależne od innych baz i wymagają przywrócenia obu baz, aby użytkownicy mogli wykonywać codzienne zadania. Określ, czy baza wymaga połączeń z serwerami połączonymi, aplikacjami zewnętrznymi lub jednostką mainframe.
- Określ dostępny sprzęt, który można wykorzystać przy odbudowie centrum danych. Ustal, gdzie się znajduje.
- Ustal, którzy pracownicy są potrzebni przy tworzeniu kopii zapasowych, przywracaniu stanu i odzyskiwaniu systemu po katastrofie. Te osoby muszą zrozumieć procedury odzyskiwania systemu i znać swoją w nich rolę. Zapisz godziny pracy wszystkich niezbędnych pracowników, ich numery kontaktowe i zadania. Sprawdź też, jak kontaktować się z tymi osobami. Określ hierarchię tych osób i dowiedz się, czy ich zastępcy mają wiedzę potrzebną do tworzenia kopii zapasowych, przywracania stanu i odzyskiwania systemu po katastrofie w sytuacji, gdy poszczególni pracownicy są niedostępni. Poznaj poziom umiejętności personelu. Ustal, czy potrzebne są dodatkowe szkolenia. Jeśli tak, znajdź kursy, które będą przydatne.
- Udokumentuj wszystkie informacje na temat zadań usług SQL Server Agent, serwerów połączonych i loginów, które mogą okazać się potrzebne przy przywracaniu bazy na innym serwerze.

Wzorce użytkowania danych

W ramach opracowywania planu odzyskiwania systemu należy przeanalizować sposoby użytkowania danych w typowych scenariuszach. Dla każdej bazy ustal, jak często dane są modyfikowane. Inne strategie tworzenia kopii zapasowych stosuje się dla baz, do których dane są wczytywane raz dziennie, a inne dla baz przeznaczonych tylko do odczytu lub modyfikowanych co minutę. Oddziel zmieniane tabele od tabel używanych tylko do odczytu. Tabele obu rodzajów można umieścić w innych grupach plików i wykorzystać te grupy w planie tworzenia kopii zapasowych.

Określ wzorce użytkowania baz danych w ciągu dnia, aby ustalić strategię tworzenia kopii zapasowych. Przykładowo w czasie wysokiej aktywności administrator może zaplanować częste tworzenie różnicowych kopii zapasowych lub kopii dziennika transakcji. Pełną kopię zapasową należy wykonywać wtedy, gdy obciążenie jest niższe.

Sprawdź, ile miejsca na dysku zajmuje dziennik transakcji w godzinach maksymalnego obciążenia. Zbadaj też wydajność dziennika. W godzinach szczytu dziennik transakcji może zapełniać przeznaczony na niego dysk. Ponadto liczba dysków przydzielonych na dziennik transakcji może być za mała, co prowadzi do spadku jego wydajności w godzinach maksymalnego obciążenia. Wybór modelu odzyskiwania wpływa zarówno na ilość miejsca zajmowanego na dysku, jak i na wydajność systemu.

W bazie działającej w modelu FULL pomyśl o zastosowaniu modelu BULK_LOGGED na czas wykonywania operacji masowych. Pozwoli to poprawić wydajność, ponieważ w drugim z tych modeli rejestrowana jest minimalna ilość informacji o transakcjach. Przed rozpoczęciem operacji masowych powinienś przynajmniej utworzyć różnicową kopię zapasową lub kopię dziennika transakcji, aby zabezpieczyć się przed awarią dysku z danymi i brakiem dostępu do końcówki dziennika transakcji.

Pomyśl też, jak baza ma być używana. Jeśli jest to baza o znaczeniu strategicznym, zapewnij rezerwowe zasoby sprzętowe. Zaczynj od systemu magazynowania danych zapewniającego nadmiarowość, np. zastosuj macierz RAID 10, a następnie dołącz dodatkowy sprzęt, jeśli możesz sobie na to pozwolić. Najbardziej zaawansowanym rozwiązaniem jest tu kompletny rezerwowy aktywny system w klastrze z przełączaniem awaryjnym. Określ, jaka utrata danych jest dla firmy akceptowalna. Zaplanuj tworzenie kopii zapasowych dziennika transakcji w taki sposób, aby spełnić wymagania organizacji. Ponadto stosuj model FULL, aby można było odtworzyć stan z momentu wystąpienia awarii.

Czas na konserwację

Czasem strategia tworzenia kopii zapasowej zależy od ilości czasu dostępnego na konserwację, kiedy to trzeba przeprowadzić defragmentację bazy danych, zapisać kopie zapasowe, zaktualizować statystyki i wykonać inne zadania. Aby zwiększyć komfort pracy użytkowników, firmy muszą szybciej pobierać dane i zapewniać klientom lepszy dostęp do informacji. W efekcie użytkownicy stają się bardziej zależni od danych. Jednoczesne zapewnienie wygody klientom, zmniejszenie ryzyka utraty danych i umożliwienie szybkiego przywrócenia stanu po awarii systemu bazodanowego nie jest łatwe.

Zadaniem administratora baz danych jest określenie najlepszej strategii tworzenia kopii zapasowych, dostosowanej do wymagań biznesowych firmy. Czas na konserwację jest zwykle ograniczony. W systemie SQL Server dostępne są różne mechanizmy tworzenia kopii zapasowych, które można wykorzystać w celu spełnienia wymogów firmy. Poniżej wymienione są wybrane trudności związane z projektowaniem strategii tworzenia kopii zapasowych.

- Czas na tworzenie kopii zapasowych w wysoce dostępnej bazie o znaczeniu strategicznym może być krótki. Firmy często mają podpisane umowy o poziomie świadczonych usług i muszą zakończyć konserwację w określonym czasie, po którym użytkownicy wracają do korzystania z systemu. Jeśli tworzenie kopii potrwa dłużej, może opóźnić inne prace nad bazą i sprawić, że nie uda się zakończyć konserwacji przed zalogowaniem użytkowników do systemu. W efekcie firma może utracić okazje biznesowe.
- Czasem w trakcie konserwacji trzeba utworzyć kopie zapasowe dla wielu baz danych. Możesz spróbować przyspieszyć prace i tworzyć kopie zapasowe jednocześnie na wszystkich dostępnych nośnikach z kopiami zapasowymi, jeśli serwer bazodanowy ma wystarczające zasoby.
- Rozrastająca się baza danych powoduje, że czas potrzebny na jej konserwację się wydłuża. Aby sobie z tym poradzić, można dodać nowe urządzenia na kopie zapasowe, zastosować szybsze serwery bazodanowe i szybsze urządzenia wejścia-wyjścia. W niektórych sytuacjach czas na konserwację da się wydłużyć, jednak często jest to niemożliwe. Rozważ też tworzenie kopii zapasowych w sieci SAN, aby przyspieszyć ten proces.
- Na serwerze bazodanowym w bazach zwykle wykonywane są też inne operacje (np. testy spójności, defragmentacja, aktualizowanie statystyk i wczytywanie danych). Gdy baza staje się większa, operacje te się wydłużają.

- W czasie przeznaczonym na konserwację należy też zaktualizować oprogramowanie, zainstalować poprawki i pakiety SP oraz zaktualizować strukturę bazy danych.
- Czasem konieczne jest też zmodyfikowanie katalogów pełnotekstowych.
- Ponieważ coraz więcej firm docenia korzyści stosowania systemów wspomagających podejmowanie decyzji (takich jak usługi SQL Server Analysis Services), w czasie konserwacji konieczne może być zmodyfikowanie baz danych używanych w takich systemach.

Jeśli baza jest niewielka, można tworzyć jej pełną kopię zapasową każdej nocy, a opisane wymagania zostaną spełnione. Gdy jednak baza stanie się większa, może to okazać się niemożliwe. Wtedy należy wykonywać pełną kopię zapasową co weekend, a co noc tworzyć różnicową kopię zapasową. Dla większych baz danych rozważ umieszczenie danych przeznaczonych tylko do odczytu oraz danych do odczytu i zapisu w różnych grupach plików. Pozwoli to tworzyć pełne kopie zapasowe dla części bazy co weekend i różnicowe kopie dla tych części co noc. Jeśli baza wciąż będzie się rozrastać, pomyśl o conocnym tworzeniu kopii zapasowych poszczególnych plików.

Inne rozwiązania zapewniające wysoką dostępność

Jeśli w bazie danych zastosowano mechanizmy zapewniające wysoką dostępność, takie jak AlwaysOn, klastry z przełączaniem awaryjnym, przesyłanie dzienników lub kopie lustrzane, przy tworzeniu kopii zapasowych należy uwzględnić pewne dodatkowe aspekty.

- Jeżeli używasz technologii AlwaysOn (wprowadzono ją w wersji SQL Server 2012), możesz zmodyfikować plany tworzenia kopii zapasowych. Jednym z powodów budowania replik pomocniczych jest przeniesienie na nie z repliki głównej obciążenia związanego z operacjami wejścia-wyjścia wykonywanymi w czasie tworzenia kopii zapasowej. W tym modelu kopie zapasowe tworzy się nie na podstawie repliki głównej lub zwykłych replik pomocniczych, ale przy użyciu specjalnej, przeznaczonej do tego repliki pomocniczej. Aby wskazać tę replikę, użyj ustawienia `BACKUP_PRIORITY` dla grupy dostępności.

UWAGA Więcej informacji o technologii AlwaysOn podajemy w rozdziale 25., „Grupy dostępności AlwaysOn”.

- Przesyłanie dzienników pozwala utworzyć kopie zapasowe dzienników transakcji. Nie należy wtedy tworzyć kopii dzienników innymi metodami, ponieważ zakłóca to łańcuch dzienników i uniemożliwia ich przywracanie na serwerze rezerwowym. Jeśli tak się stanie, trzeba zrekonfigurować proces przesyłania dzienników.
- Jeśli przy korzystaniu z kopii lustrzanych serwer lustrzany przestanie działać, na serwerze głównym należy umieścić w kolejce dziennika transakcji wszystkie modyfikacje, które potem zostaną przeniesione na serwer lustrzany w dzienniku. Dziennika transakcji nie można przyciąć bardziej niż do miejsca, od którego zaczynają się modyfikacje nieprzesłane na serwer lustrzany.
- Klaster z przełączaniem awaryjnym pracuje jak pojedynczy egzemplarz systemu SQL Server, dlatego nie wymaga specjalnego traktowania. Jeśli jednak klaster jest zintegrowany z przesyłaniem dzienników lub kopiami lustrzanymi, należy uwzględnić opisane powyżej uwagi.

- Zastosowanie replikacji (w dowolnej postaci) wymaga opracowania szczegółowego planu odzyskiwania kopii zapasowej, obejmującego synchronizację między bazą źródłową, dystrybucyjną bazą danych i subskrybentami. Replikacja może powodować dodatkowe komplikacje w planie tworzenia kopii zapasowej i odzyskiwania danych. Choć możesz odzyskać zreplikowaną bazę danych na podstawie kopii zapasowej, udane wykonanie tej operacji wymaga spełnienia pewnych warunków. Zastanów się, czy używanie kopii zapasowych to właściwe rozwiązanie przy odzyskiwaniu baz danych subskrybentów. Jeśli bazy są niewielkie, łatwiejsze i bardziej niezawodne może być odtwarzanie bazy i przenoszenie do subskrybentów nowego snapshota z serwera wydawcy.

UWAGA W replikacji transakcyjnej awaria subskrybenta powoduje, że dziennika transakcji nie można przyciąć poza miejscem, od którego modyfikacje nie zostały zreplikowane na serwerze subskrybenta.

Opracowywanie i wykonywanie planu tworzenia kopii zapasowych

System SQL Server udostępnia trzy metody planowania i wykonywania procesu tworzenia kopii zapasowych. Możesz wykorzystać graficzny interfejs z programu SQL Server Management Studio, plany konserwacji bazy danych i instrukcje tworzenia kopii zapasowych z języka T-SQL. W tym podrozdziale omawiamy wszystkie trzy podejścia.

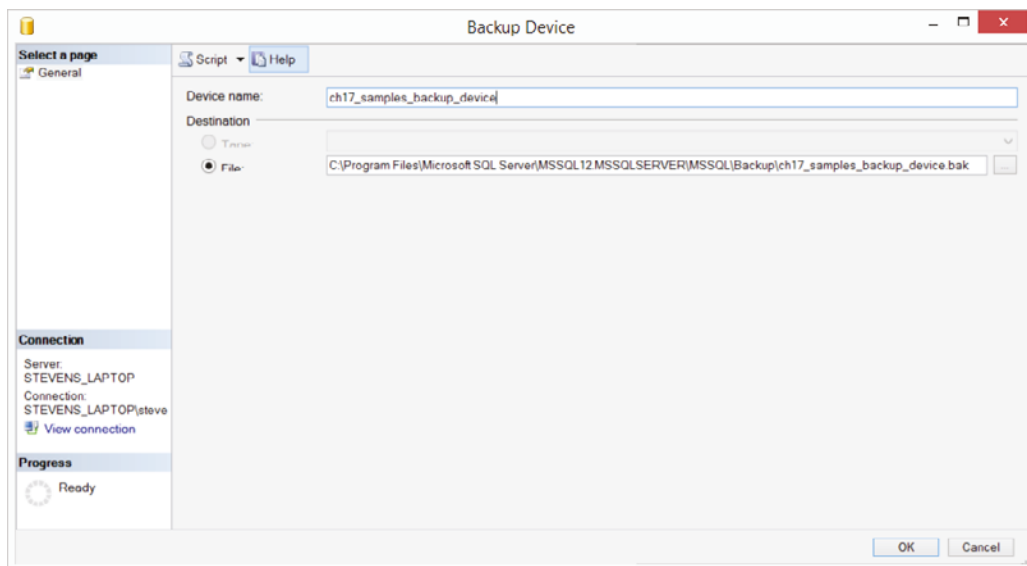
Używanie programu SQL Server Management Studio

Program SQL Server Management Studio udostępnia mechanizmy zarządzania kopiami zapasowymi, pozwalające administratorom baz danych opracować wykonywany zgodnie z harmonogramem plan konserwacji lub bezpośrednio utworzyć kopię zapasową. Zanim zaczniesz, określ docelową lokalizację obrazu z kopią zapasową. Możesz podać ścieżkę do katalogu i nazwę pliku albo odrębne urządzenie z kopiami zapasowymi.

Jeśli używasz urządzeń z kopiami zapasowymi, najpierw musisz utworzyć logiczne urządzenie i określić, gdzie system SQL Server ma kopiować obraz z kopią zapasową. W programie SQL Server Management Studio wybierz opcję *Server Objects/Backup Devices/New Backup Device*. Pojawi się okno dialogowe widoczne na rysunku 17.21. Znajdują się w nim dwa rodzaje docelowych lokalizacji.

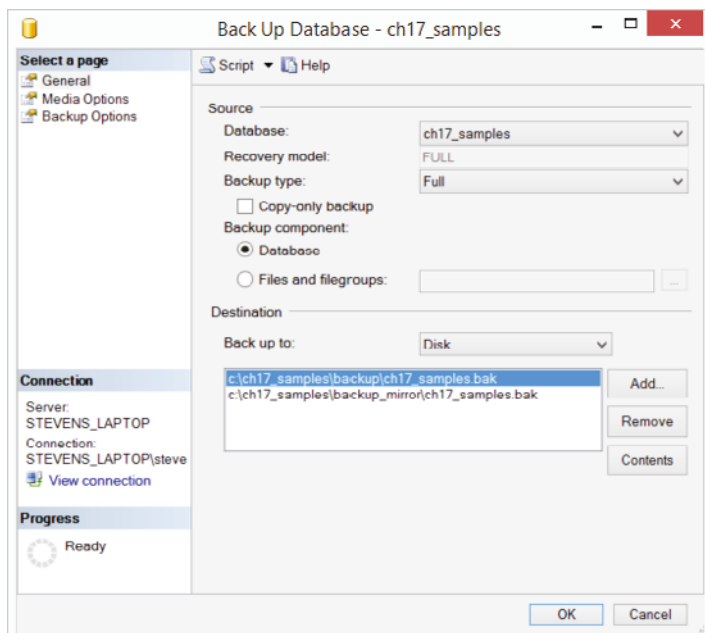
- **Tape.** Wymaga, aby na serwerze bazodanowym dostępny był lokalny napęd taśmowy.
- **File.** Wymaga podania poprawnej lokalizacji docelowej na dysku.

Przy tworzeniu kopii zapasowej na dysku nie musisz podawać urządzenia, ponieważ lokalizacja jest zapisana na stałe. Wystarczy określić unikatowe nazwy plików z kopiami zapasowymi. W takich nazwach podaj nazwę bazy danych, typ kopii zapasowej oraz datę i czas, co zapewni ich niepowtarzalność. To podejście daje więcej swobody niż korzystanie z urządzeń z kopiami zapasowymi. Aby wykonać kopię zapasową bazy w programie SQL Server Management Studio, wykonaj następujące kroki.



Rysunek 17.21. Określanie lokalizacji kopii zapasowych

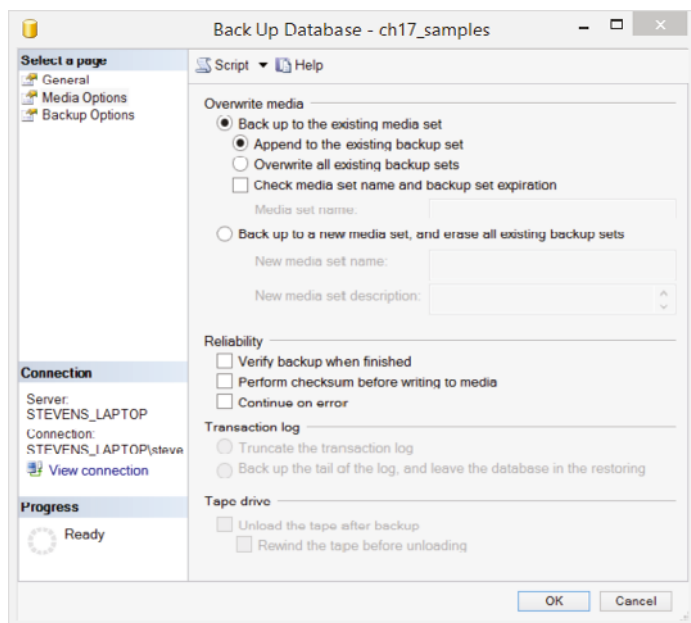
1. Wybierz bazę, której kopię zapasową chcesz utworzyć. Kliknij tę bazę prawym przyciskiem myszy i wybierz opcję *Tasks/Backup*. Otworzy się zakładka *General* w oknie dialogowym *Back Up Database*, co przedstawiono na rysunku 17.22.



Rysunek 17.22. W tym oknie dialogowym można utworzyć kopię zapasową bazy

2. W sekcji *Source* w tym oknie dialogowym ustaw następujące opcje.
 - **Database.** Podaj bazę, której kopię zapasową chcesz utworzyć.

- **Recovery model.** To pole jest nieaktywne, ponieważ nie można go zmienić. Używany jest model FULL. W modelu SIMPLE nie można wykonać kopii zapasowej dziennika transakcji, ponieważ jest on przycinany w ramach tworzenia punktu kontrolnego. Ponadto kopie zapasowe plików i ich grup można tworzyć wyłącznie dla danych w trybie tylko do odczytu.
 - **Backup type.** Dostępne opcje to *Full*, *Differential* i *Transaction Log*.
 - **Copy-only backup.** Umożliwia utworzenie kopii zapasowej, która nie wpływa na łańcuch dzienników transakcji i nie powoduje ich przycinania.
 - **Backup component.** Dostępne są następujące opcje.
 - **Database.** Tworzy kopię zapasową bazy danych.
 - **Files and filegroups.** Tworzy kopie zapasowe plików lub ich grup. Jeśli wybierzesz tę opcję, pojawi się okno dialogowe, w którym można wybrać pliki lub grupy plików.
3. W sekcji *Destination* możesz wskazać typ lokalizacji docelowej — *Disk* lub *URL* (tę drugą opcję wprowadzono w systemie SQL Server 2014). Na rysunku 17.22 widoczna jest lista dwóch plików. Zostały one utworzone w przykładach, wcześniej w tym rozdziale. Możesz dodać nowe lokalizacje i usunąć wymienione.
 4. Po lewej stronie ekranu wybierz zakładkę *Media Options*. Pojawi się ekran widoczny na rysunku 17.23.



Rysunek 17.23. Konfigurowanie nośników

5. W sekcji *Overwrite media* możesz ustawić tworzenie kopii zapasowych w istniejącym zestawie nośników. Wtedy należy skonfigurować następujące opcje.
 - **Append to the existing backup set.** Powoduje zachowanie istniejących kopii zapasowych i dodanie nowych w danym zestawie nośników. Jest to ustawienie domyślne.

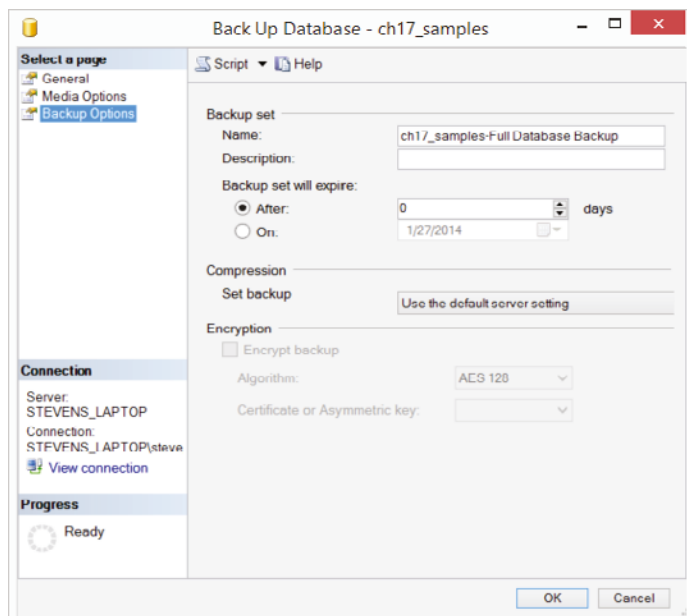
- **Overwrite all existing backup sets.** Powoduje usunięcie wszystkich istniejących kopii zapasowych i zastąpienie ich nową kopią. W efekcie wszystkie istniejące kopie zostają zastąpione, chyba że zaznaczone jest pole *Check media set name and backup set expiration*.

Możesz też ustawić tworzenie kopii zapasowych w nowym zestawie nośników i skasowanie wszystkich istniejących zestawów kopii zapasowych. Spowoduje to skasowanie wszystkich kopii z danego nośnika i utworzenie nowego zestawu nośników zgodnie z podanymi ustawieniami.

6. W sekcji *Reliability* znajdują się trzy pola wyboru. Warto je zaznaczyć, ponieważ kopia zapasowa nie jest przydatna, jeśli nie można jej odzyskać. Zaznacz te pola.
 - **Verify backup when finished.** Po zakończeniu tworzenia kopii zapasowej system SQL Server sprawdzi, czy wszystkie woluminy można odczytać.
 - **Perform checksum before writing to media.** System SQL Server określi sumę kontrolną przed zapisem kopii na nośniku. Tę sumę można potem wykorzystać w trakcie odzyskiwania, aby się upewnić, że nikt nie zmodyfikował kopii zapasowej. Operacja ta prowadzi do spadku wydajności.
 - **Continue on error.** Tworzenie kopii zapasowej powinno być kontynuowane nawet po natrafieniu na błąd (np. błąd sumy kontrolnej lub uszkodzoną stronę).
7. W sekcji *Transaction log* znajdują się opcje istotne tylko dla kopii zapasowych dzienników transakcji. Jeśli tworzysz takie kopie, zaznacz odpowiednie opcje.
 - **Truncate the transaction log.** Przy tworzeniu standardowych kopii zapasowych dzienników transakcji często przycina się dzienniki po zapisaniu kopii.
 - **Back up the tail of the log, and leave the database in the restoring state.** Opcja jest przydatna, gdy pliki danych bazy są nieodstępne. Możliwe, że fizyczne dyski z tymi plikami zostały uszkodzone, ale dziennik transakcji zapisany na innym fizycznym dysku nadal jest dostępny. Wtedy w trakcie odzyskiwania bazy danych zastosuj kopię zapasową dziennika transakcji, aby odzyskać dane do punktu wystąpienia awarii.
8. Sekcja *Tape drive* zawiera pola wyboru związane z obsługą taśmy. Dostępne są tu dwie opcje.
 - *Unload the tape after backup.*
 - *Rewind the tape before unloading.*
9. Z listy widocznej po lewej stronie okna wybierz zakładkę *Backup Options*. Pojawi się strona widoczna na rysunku 17.24.

W sekcji *Backup set* ustaw następujące opcje.

 - **Name.** Ustaw nazwę zestawu kopii zapasowych, aby ułatwić jego identyfikację. Ta nazwa odróżnia daną kopię zapasową od innych z tego samego urządzenia.
 - **Description.** Podaj opcjonalny opis dla danego zestawu nośników.
 - **Backup set will expire.** Skonfiguruj opcje w tym miejscu zgodnie z firmową polityką przechowywania danych. W ten sposób zabezpieczysz się przed zastąpieniem zestawu kopii zapasowych przez używany w systemie SQL Server proces tworzenia takich kopii.
 - **After.** To pole określa liczbę dni (od 0 do 99 999), po których upływie dany zestaw zostanie zastąpiony. Wartość domyślna to 0, co oznacza, że dany zestaw nigdy nie wygaśnie. Możesz zmienić wartość domyślną używaną na serwerze. W tym celu



Rysunek 17.24. Tu można ustawić opcje związane z kopiami zapasowymi

wybrać opcję *Properties* egzemplarza systemu SQL Server, otwórz zakładkę *Database Settings* i zmień domyślny czas zachowywania nośnika z kopiami zapasowymi (opcja *Default back media retention (in days)*).

➤ **On.** Określa datę wygaśnięcia danego zestawu kopii zapasowych.

10. W sekcji *Compression* wybierz jedną z trzech opcji kompresji kopii zapasowych.

➤ *Use the default server setting* (użyj domyślnych ustawień serwera).

➤ *Compress backup* (kompresuj kopie zapasowe).

➤ *Do not compress backup* (nie kompresuj kopii zapasowych).

11. W sekcji *Encryption* możesz ustawić opcje szyfrowania kopii zapasowych. Aby były dostępne, trzeba spełnić następujące warunki.

➤ Utworzyć klucz główny bazy danych.

➤ Utworzyć certyfikat lub klucz asymetryczny na potrzeby szyfrowania kopii zapasowej.

12. Kliknij przycisk **OK**, a proces tworzenia kopii zapasowej zostanie uruchomiony.

Plany konserwacji bazy danych

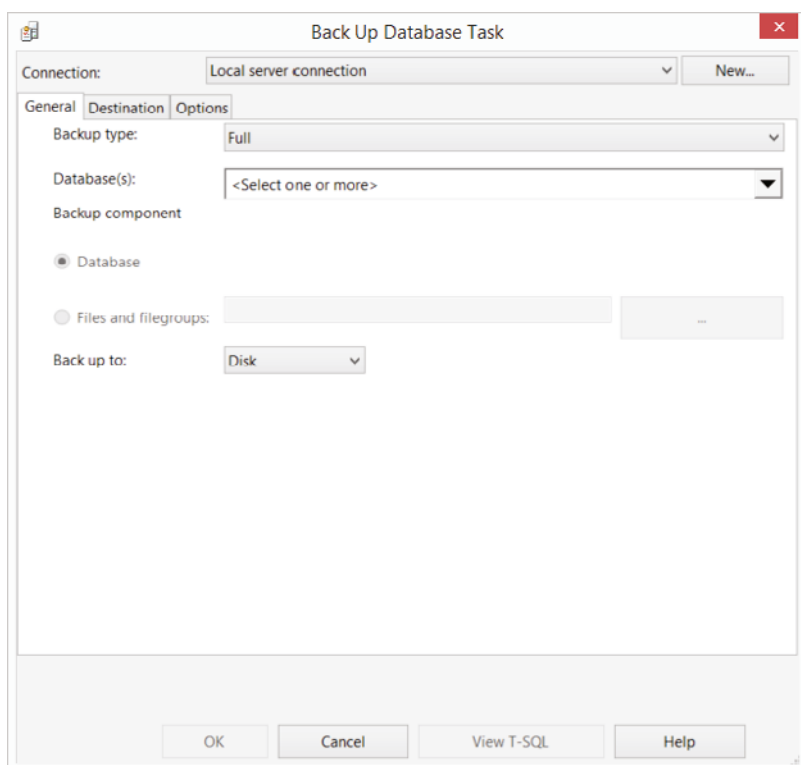
Innym sposobem tworzenia kopii zapasowych jest utworzenie planów konserwacji każdej bazy danych, umieszczenie ich realizowania w harmonogramie i ustawienie systemu SQL Server, aby wysyłał e-mailem raport o przebiegu tworzenia kopii zapasowej.

Plan konserwacji bazy danych ma być łatwy w (wielokrotnym) użyciu. Można w nim ująć wiele standardowych zadań z zakresu konserwacji i pogrupować je, wykonywać zgodnie z planem oraz otrzymać raport o przebiegu przeprowadzonych operacji. Plan możesz utworzyć ręcznie lub za pomocą kreatora, który wyświetla serię okien dialogowych.

Aby utworzyć plany konserwacji w programie SQL Server Management Studio, otwórz węzeł *Management/Maintenance Plans*, a następnie kliknij go prawym przyciskiem myszy i wybierz opcję *New Maintenance Plan*. Po ustawieniu nazwy planu konserwacji przejdziesz do ekranu projektowego, gdzie można wykonać następujące kroki.

UWAGA Jeśli trzeba, otwórz okno narzędziowe *Maintenance Plan Tasks* (domyślnie jest ono ukryte). W tym celu wciśnij kombinację *Ctrl+Alt+X* lub wybierz opcję *View/Toolbox*.

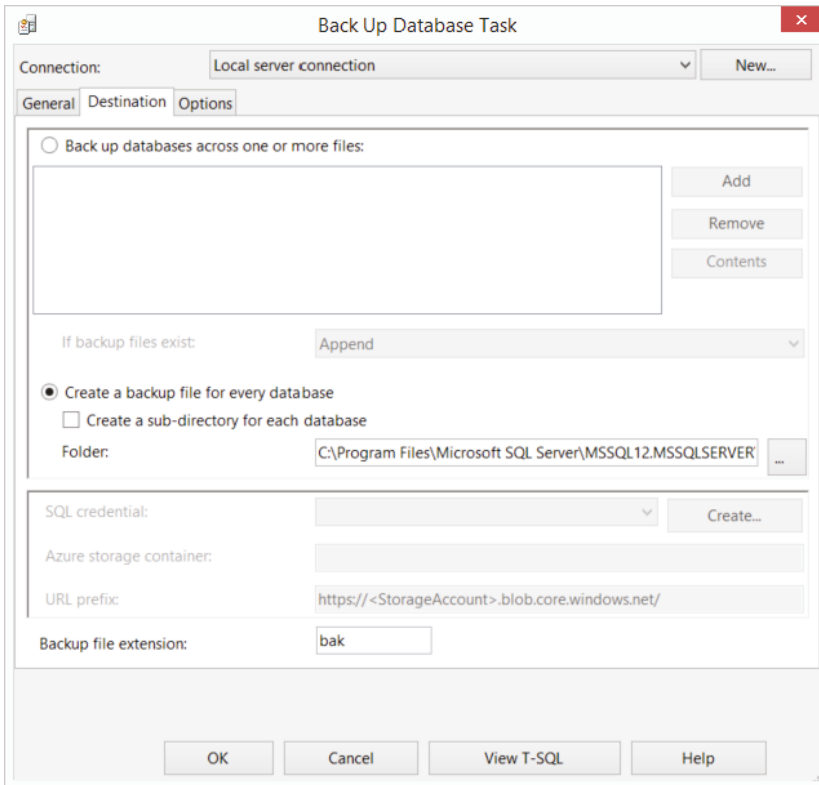
1. Wybierz zadanie *Back Up Database Task* i przeciągnij je do obszaru projektowego.
2. Kliknij zadanie *Back Up Database Task* prawym przyciskiem myszy i wybierz opcję *Edit*. Pojawi się okno dialogowe *Back Up Database Task* widoczne na rysunku 17.25.



Rysunek 17.25. Okno do konfigurowania zadania tworzenia kopii zapasowej

3. W polu *Connection* wybierz opcję *Local server connection* lub, jeśli dany plan ma tworzyć kopie zapasowe baz danych z innego serwera, ustaw opcję *New connection* i podaj informacje o połączeniu.
4. W polu *Database(s)* wybierz bazę *ch17_samples*. Możesz ustawić tu więcej baz, jeśli tworzenie ich kopii zapasowych ma przebiegać identycznie.
5. W polu *Backup component* wybierz opcję *Database* lub *Files and filegroups*. Jeśli ustawisz opcję *Files and filegroups*, podaj pliki lub grupy plików, dla których chcesz tworzyć kopie zapasowe.

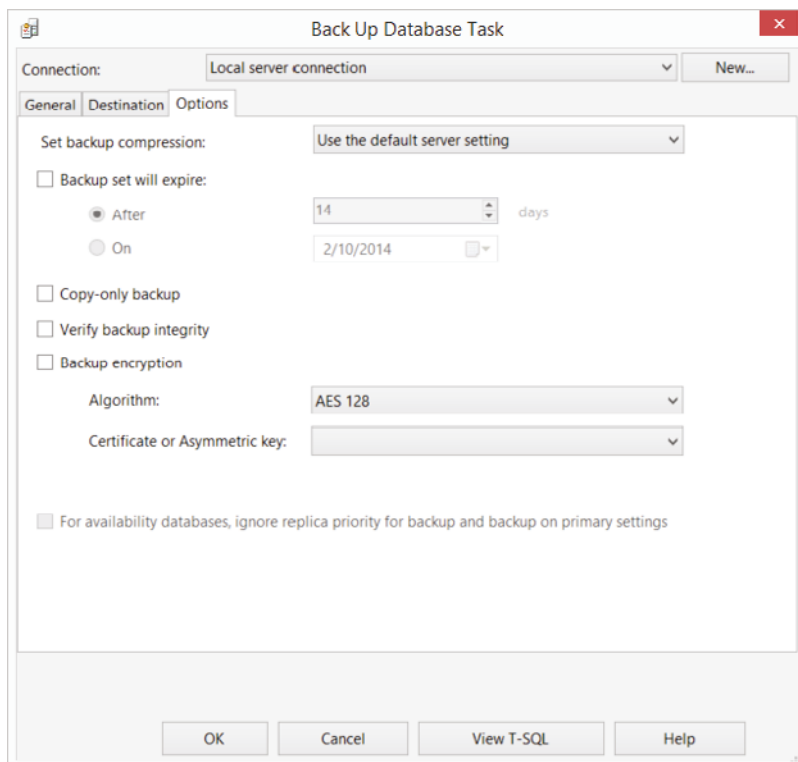
6. W polu *Back up to* wybierz opcję *Disk, Tape* lub *URL*.
7. Otwórz zakładkę *Destination*, co pokazano na rysunku 17.26. W zakładce *Destination* możesz wybrać pozycję z listy zapisanych na stałe plików, aby określić, gdzie ma zostać zapisana kopia zapasowa bazy. Plan konserwacji może też utworzyć dla każdej bazy plik z automatycznie wygenerowaną nazwą.



Rysunek 17.26. Konfigurowanie docelowej lokalizacji kopii zapasowych

- a. Jeśli wybrałeś opcję *Back up databases across one or more files*, wykonaj następujące kroki.
 1. Kliknij przycisk *Add*, aby określić lokalizację kopii zapasowej. Jeżeli używasz dysku, podaj pełną ścieżkę lub taśmowe urządzenie z kopiami zapasowymi. Dla taśm podaj lokalizację taśmy lub taśmowe urządzenie z kopiami zapasowymi. Możesz wskazać kilka plików lub urządzeń. Wtedy wszystkie bazy będą zapisywane we wskazanych urządzeniach (system SQL Server obsługuje do sześćdziesięciu czterech urządzeń z kopiami zapasowymi).
 2. W polu *If backup files exist* określ, czy system ma dodać dane do istniejącego pliku z kopią zapasową, czy je zastąpić. Domyślnie dane są dodawane (opcja *Append*).
- b. Jeśli wybrałeś opcję *Create a backup file for every database*, wykonaj następujące kroki.
 1. Wybierz katalog główny przeznaczony na kopie zapasowe.
 2. Ustaw rozszerzenie plików z kopiami zapasowymi.

8. Jeśli w zakładce *General* wybrałeś *URL* jako lokalizację, dostępne będą opcje *SQL credential*, *Azure storage container* i *URL prefix*. W tych polach możesz podać dane uwierzytelniające systemu SQL Server na potrzeby kontenera Azure Storage. Otwórz zakładkę *Options*, co przedstawiono na rysunku 17.27.



Rysunek 17.27. Ustawianie różnych opcji kopii zapasowych

9. W zakładce *Options* określ w polu *Set backup compression*, czy kopia zapasowa ma być kompresowana.
10. Możesz też określić datę wygasania zestawu kopii zapasowych. Zapobiega to ich przypadkowemu zastąpieniu.
11. Opcja *Copy-only Backup* umożliwia utworzenie kopii niezależnej od standardowych kopii zapasowych. W ten sposób można zbudować kompletną kopię bazy danych bez uwzględniania łańcucha kopii zapasowych i dzienników. W omawianym przykładzie nie zaznaczaj tej opcji.
12. Zaznacz pole *Verify backup integrity* (jest to zalecana praktyka).
13. Jeśli zaznaczysz pole *Backup encryption*, będziesz mógł wybrać algorytm szyfrowania kopii zapasowej (pole *Algorithm*) i określić certyfikat lub klucz asymetryczny (pole *Certificate or Asymmetric key*).
14. Kliknij przycisk *OK*. To spowoduje zamknięcie okna dialogowego *Back Up Database Task* i powrót do okna projektowego.

15. Kliknij przycisk *Reporting and Logging* na pasku narzędzi okna *Maintenance Plan [Design]* i określ, w jakiej postaci chcesz otrzymywać raport z przebiegu tworzenia kopii zapasowych. Jeśli chcesz użyć e-maili, skonfiguruj usługę Database Mail i operatora w usłudze SQL Server Agent. Kliknij przycisk OK.
16. Kliknij przycisk *Schedule* i ustaw harmonogram wykonywania danego planu konserwacji.

UWAGA Gdy w ramach planu konserwacji dla każdej bazy tworzony jest plik z kopią zapasową, pliki mają nazwy w formacie *ch17_samples_backup_2014_01_24_090243_2394128* (jest to nazwa dla bazy *ch17_samples*). Obejmuje ona typ kopii zapasowej oraz datę i godzinę jej utworzenia. Jest to wygodne, ponieważ na podstawie nazwy można ustalić czas utworzenia kopii zapasowej i jej rodzaj. Wadą jest to, że powstaje jeden plik z kopią zapasową, co nie pozwala uzyskać wzrostu wydajności możliwego dzięki zapisaniu kopii w kilku plikach.

Do procesu tworzenia kopii zapasowych innych baz danych, dla których obowiązują odmienne wymagania, możesz dodać nowe zadania. Przykładowo pierwsze zadanie *Back Up Database Task* może wykonywać pełne kopie zapasowe kilku baz danych, drugie może tworzyć kopie różnicowe, a trzecie — kopie grup plików. Wszystkie te zadania można uruchamiać zgodnie z tym samym planem.

UWAGA Wcześniej w tym rozdziale dowiedziałeś się, jak podzielić bazy danych na kategorie według potrzeb związanych z tworzeniem kopii zapasowych i przywracaniem stanu. Możesz utworzyć plany konserwacji dostosowane do poszczególnych kategorii. Następnie, gdy na serwerze pojawi się nowa baza, wystarczy określić jej kategorię i dodać do odpowiedniego planu.

Gdy plan konserwacji jest gotowy, zostaje automatycznie uwzględniony w harmonogramie jako zadanie usługi SQL Server Agent.

Używanie instrukcji tworzenia kopii zapasowych z języka T-SQL

Wszystkie polecenia i funkcje tworzenia kopii zapasowych używane w programie SQL Server Management Studio są bezpośrednio dostępne w języku T-SQL. Pełny opis składni instrukcji BACKUP znajdziesz w dokumentacji Books Online na stronie [http://msdn.microsoft.com/en-us/library/ms186865\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms186865(v=sql.120).aspx). Oto przykładowe polecenia.

- Tworzenie logicznego urządzenia z kopiami zapasowymi przeznaczonego na kopie bazy *ch17_samples*:

```
EXEC sp_addumpdevice 'disk', 'ch17_samples_Backup',
'C:\BACKUP\ch17_samples.bak';
```

UWAGA Jeśli używasz innych liter napędów, zmień ścieżkę z nazwą pliku. Jeżeli korzystasz z udziału plikowego, powinieneś podać nazwę UNC zamiast litery współużytkowanego napędu. Oto przykład:

```
\\myserver\myshare\Backup\ch17_samples.bak
```

- Tworzenie pełnej kopii zapasowej bazy ch17_samples:
BACKUP DATABASE ch17_samples TO ch17_samples_Backup;
- Tworzenie pełnej kopii różnicowej:
BACKUP DATABASE ch17_samples TO ch17_samples_Backup WITH DIFFERENTIAL;
- Tworzenie kopii zapasowej końcówki dziennika transakcji. Kopie tego rodzaju są używane po awarii bazy danych, gdy dostępne są dzienniki transakcji:
BACKUP LOG ch17_samples TO tailLogBackup WITH NORECOVERY;

UWAGA Ta instrukcja wymaga utworzenia nowego urządzenia z kopiami zapasowymi tailLogBackup.

- Określanie nazwy pliku z kopią zapasową bazy ch17_samples:
DECLARE @devname varchar(256)
SELECT @devname = 'C:\BACKUP\ch17_samples_Full_' + REPLACE
(REPLACE(CONVERT(Varchar(40), GETDATE(), 120), '-', '_'), ':', '_') + '.bak';
- Tworzenie pełnej kopii zapasowej bazy ch17_samples:
BACKUP DATABASE ch17_samples TO DISK = @devname;
- Określanie nazwy pliku z różnicową kopią zapasową bazy ch17_samples:
DECLARE @devname varchar(256)
SELECT @devname = 'C:\BACKUP\ch17_samples_Differential_' + REPLACE
(REPLACE(CONVERT(Varchar(40), GETDATE(), 120), '-', '_'), ':', '_') + '.bak';
- Tworzenie pełnej różnicowej kopii zapasowej:
BACKUP DATABASE ch17_samples TO DISK = @devname WITH DIFFERENTIAL;
- Określanie nazwy pliku z kopią zapasową bazy ch17_samples:
DECLARE @devname varchar(256)
SELECT @devname = 'C:\BACKUP\ch17_samples_Log_' + REPLACE
(REPLACE(CONVERT(Varchar(40), GETDATE(), 120), '-', '_'), ':', '_') + '.bak';
- Tworzenie standardowej kopii zapasowej dziennika transakcji:
BACKUP LOG ch17_samples TO DISK = @devname;

Gdy używasz plików na dysku, umieść każdą kopię zapasową w odrębnym pliku i odpowiednio nazwij poszczególne pliki. Nazwy powinny obejmować unikatową nazwę bazy (z członem określającym serwer, jeśli na kilku serwerach znajdują się identycznie nazwane bazy), typ kopii zapasowej i datę. We wcześniej pokazanych przykładach używane są daty w formacie rrrr_mm_dd gg_mm_ss. Nieraz tworzone są procedury składowane i funkcje zdefiniowane przez użytkownika, które przyjmują odpowiednie parametry i zwracają nazwy plików z kopiami zapasowymi.

UWAGA Nie używaj zmapowanych liter napędów w nazwach plików z kopią zapasową. Jeśli zapisujesz kopie zapasowe w plikach w udziale plikowym, podaj nazwę UNC. Zmapowane litery napędów mogą się zmieniać w zależności od tego, kto jest zalogowany na fizycznym serwerze. Wprawdzie możesz utworzyć trwale zmapowane litery napędów, jednak zaleca się używanie nazw UNC.

Zarządzanie kopiami zapasowymi

Zarządzanie kopiami zapasowymi to następne ważne zadanie administratora baz danych. Im lepszy jest proces zarządzania kopiami, tym szybciej i dokładniej można je zidentyfikować i przywrócić. Sumienne wykonywanie kopii zapasowych nie daje korzyści, jeśli nie można zidentyfikować kopii zapasowych lub, co gorsza, zostaną one utracone lub zastąpione.

Poniższe wskazówki pomogą Ci opracować program zarządzania kopiami zapasowymi.

- Twórz opisowe nazwy kopii zapasowych, aby zapobiec zastąpieniu takich kopii lub ich przeniesieniu w niewłaściwe miejsce. Możesz zastosować schemat podobny do przedstawionego wcześniej: <nazwa_serwera>_<nazwa_bazy>_<rok>_<miesiąc>_<dzień>.bck.
- Określ politykę przechowywania kopii, aby zapobiec zastąpieniu kopii zapasowej na taśmie lub dysku. Przy tworzeniu reguł uwzględnij politykę firmy, regulacje prawne, koszty, dostępną przestrzeń i logistykę.
- Taśmy się zużywają, dlatego określ, ile razy można je zastosować przed wyrzuceniem. Podnosi to koszty, ale zużyta taśma może uniemożliwić przywrócenie systemu.
- Przygotuj miejsce, w którym można łatwo uporządkować kopie zapasowe, znaleźć je i uzyskać do nich dostęp. Jeśli nie ma Ciebie w pracy i ktoś inny zajmuje się przywracaniem systemu, ta osoba musi poprawnie zidentyfikować potrzebne kopie zapasowe. Ponadto powinien przechowywać dodatkowe kopie zapasowe poza firmą, gdzie będą chronione przed lokalnymi katastrofami. Lokalizacja poza firmą powinna być dostępna 24 godziny na dobę. Lokalnie przechowuj nowsze kopie zapasowe, co umożliwi Ci szybkie przywrócenie systemu.
- Lokalizację z kopiami zapasowymi należy zabezpieczyć, tak aby nieupoważnione osoby nie miały dostępu do poufnych danych. Dla najbardziej poufnych informacji zastosuj obsługiwane przez system SQL Server szyfrowanie na poziomie kolumn.
- Utwórz kopie zapasowe kluczy używanych przy szyfrowaniu baz danych. Kopie zapasowe kluczy trzeba utworzyć ponownie po zmianie kont (zwykle są to konta usługowe lub konta komputera). Klucze i certyfikaty *trzeba* zachować, ponieważ w przeciwnym razie nie będzie można przywrócić bazy i uzyskać dostępu do danych.
- Przygotuj procedury logistyczne, które pozwolą szybko przenosić kopie zapasowe do lokalizacji poza firmą. W ten sposób zabezpieczysz kopie przed katastrofą.

Wydajność tworzenia i przywracania kopii zapasowych

System SQL Server obsługuje do sześćdziesięciu czterech urządzeń z kopiami zapasowymi i może używać ich równolegle przy tworzeniu i przywracaniu kopii zapasowych, co przyspiesza te operacje. Dla urządzeń z kopiami zapasowymi i baz danych można też zastosować inne kontrolery, co dodatkowo zwiększa wydajność. Dla urządzeń dyskowych warto zastosować macierz RAID, aby zapewnić odporność na błędy i poprawić wydajność. Użycie macierzy RAID 5 dla dysków z kopiami zapasowymi to zły pomysł, ponieważ obliczanie parzystości obniża liczbę operacji wejścia-wyjścia, a tym samym spowalnia tworzenie kopii. Aby przyspieszyć zapis, warto zastosować macierz RAID 10, zwłaszcza gdy kontroler może rozdzielać zapisywane dane między obie kopie lustrzane. Pozwala to znacznie przyspieszyć zapis. Poproś dostawcę sprzętu, by polecił odpowiednie urządzenia.

Kopie zapasowe często zapisuje się na dużych i powolnych dyskach, ponieważ są tańsze. Bezpośrednio wpływa to na wydajność zapisu kopii zapasowych na tych dyskach. Trzeba się z tym pogodzić, ponieważ bardzo nieliczne firmy są gotowe na wydawanie dużych sum na wydajne podsystemy dyskowe przeznaczone na kopie zapasowe.

Połączenie różnych kopii zapasowych — pełnych, różnicowych i dzienników transakcji — pomaga poprawić wydajność, ponieważ zmniejsza ilość danych, które trzeba wczytać z bazy i zapisać w urządzeniu z kopiami zapasowymi. Jeśli codziennie będziesz tworzył pełną kopię zapasową bazy zajmującej 5 terabajtów, codziennie zapisać trzeba będzie aż tyle danych. Jeżeli każdego dnia modyfikowana jest tylko niewielka część bazy, wykonywanie pełnej kopii zapasowej raz w tygodniu (w weekend lub w innym czasie, gdy obciążenie jest niskie) i codziennych kopii różnicowych pozwoli znacznie ograniczyć ilość wczytywanych i zapisywanych danych.

Jeśli zapisujesz kopie zapasowe na urządzeniu sieciowym lub zewnętrznym serwerze, niska przepustowość sieci może stanowić problem. Najlepiej używać dla kopii zapasowych dedykowanej szybkiej sieci, która obsługuje wymagania związane z tworzeniem i przywracaniem kopii zapasowych.

Odzyskiwanie baz danych

Odzyskiwanie to proces przywracania bazy i zapewniania jej spójnego stanu. W tym podrozdziale omawiamy różne metody odzyskiwania baz, wykorzystujące program SQL Server Management Studio i język T-SQL. Dowiesz się też, jak przywrócić systemowe bazy danych.

Proces przywracania

Administrator baz danych musi dbać, aby kopie zapasowe były regularnie wykonywane, i sprawdzać, czy umożliwiają przywrócenie baz. Każdą serię kopii zapasowych należy opisać i zachować, aby umożliwić ich szybką identyfikację w celu przywrócenia bazy. Procedury przywracania obejmują następujące zadania:

- przywracanie pełnej bazy danych,
- przywracanie dziennika transakcji,
- przywracanie części bazy danych,
- przywracanie plików i ich grup,
- przywracanie snapshota bazy danych,
- przywracanie tabel z danymi historycznymi.

Pełne przywracanie bazy danych

Pełne przywracanie bazy danych obejmuje kompletny obraz wszystkich danych z wszystkich plików i wystarczającą część dziennika transakcji, aby można było poprawnie odtworzyć zatwierdzone i niezatwierdzone transakcje. Pełne przywracanie może być podstawą do przywracania kopii różnicowych i dziennika transakcji w celu odtworzenia stanu bazy z określonego punktu w czasie. Określ, czy w trakcie pełnego przywracania należy zastąpić bieżącą bazę, czy baza powinna zostać aktywna i czy przywracane mają być dodatkowe kopie zapasowe (np. kopie różnicowe i dzienników transakcji). Należy też zastosować klauzulę **WITH MOVE**, jeśli chcesz zmienić nazwę plików bazy danych lub przenieść je do innego katalogu.

Po wykonaniu pełnego przywracania odtwórz wszystkie pozostałe kopie zapasowe — różnicowe i dzienników transakcji. Zaletą tego podejścia jest mniejsza liczba kroków przy przywracaniu bazy. Jest to jednak powolny proces, wymagający zarezerwowania czasu na konserwację.

UWAGA W wersjach starszych od SQL Server 2005 konieczne było zainicjowanie pliku przez wypełnienie go zerami w celu zastąpienia istniejących danych przy wykonywaniu następujących operacji: tworzenia bazy danych, dodawania plików, dzienników lub danych do istniejącej bazy, powiększania istniejącego pliku, przywracania bazy lub grupy plików. Dla dużych baz inicjowanie plików zajmowało sporo czasu. Od wersji SQL Server 2005 można stosować natychmiastowe inicjowanie plików, jeśli konto usługowe systemu SQL Server ma przyznane uprawnienie `SE_MANAGE_VOLUME_NAME` systemu Windows (aby je przyznać, przypisz konto do zasady bezpieczeństwa Wykonuj zadania konserwacji woluminów; ang. *Perform Volume Maintenance Tasks*). Natychmiastowe inicjowanie plików skraca czas tworzenia bazy danych i wykonywania innych zadań. Technika ta działa tylko dla plików z danymi (nie jest dostępna dla plików dzienników transakcji).

Obraz pełnej różnicowej kopii zapasowej zawiera wszystkie ekstenty zmodyfikowane od momentu utworzenia ostatniej pełnej kopii. Zwykle kopia różnicowa jest mniejsza i działa szybciej niż pełna kopia (pod warunkiem, że liczba modyfikacji nie jest bardzo duża). Przywracanie kopii różnicowej odbywa się zwykle jako uzupełnienie przywracania pełnej kopii. Najpierw należy wtedy przywrócić pełną kopię zapasową, potem baza jest ustawiana w tryb `NORECOVERY`, a następnie można przywrócić kopię różnicową.

Przywracanie dzienników transakcji

Wcześniej wspomniano, że w bazach o znaczeniu strategicznym można zmniejszyć ryzyko utraty danych dzięki okresowemu tworzeniu kopii zapasowych dzienników transakcji. *Przywracanie dzienników transakcji* wymaga pełnej kopii zapasowej bazy danych, kopii zapasowej pliku lub kopii zapasowej grupy plików. Po odtworzeniu takiej kopii należy przywrócić kopie różnicowe, a następnie kopie dzienników transakcji w odpowiedniej kolejności (od najstarszego). Pozwoli to odtworzyć bazę do stanu z określonego punktu w czasie. Można przywrócić wszystkie dzienniki transakcji lub zatrzymać się w wybranym miejscu. Aby np. przywrócić bazę do punktu sprzed wystąpienia określonego błędu, zastosuj jedną z poniższych opcji przywracania dziennika transakcji:

- `WITH STOPAT` — pozwala zatrzymać przywracanie transakcji w określonym czasie,
- `WITH STOPATMARK` — pozwala zatrzymać przywracanie po napotkaniu oznaczonej transakcji,
- `WITH STOPBEFOREMARK` — pozwala zatrzymać przywracanie przed oznaczoną transakcją.

Do oznaczania transakcji w dzienniku służy opcja `WITH MARK` używana w poleceniu `BEGIN TRANSACTION`. Przy włączonym oznaczaniu po zatwierdzeniu transakcji do tabeli `logmarkhistory` w bazie `msdb` wstawiany jest wiersz.

Zwykle przy przywracaniu do punktu w czasie trzeba określić dokładny moment. Możliwe, że proces wsadowy spowodował problemy i chcesz przywrócić bazę do punktu bezpośrednio sprzed rozpoczęcia wykonywania tego procesu. Kiedy dokładnie to było? Trudno to ustalić, chyba że używałeś narzędzia do odczytu dzienników. Dlatego przydatne są *znaczniki dziennika*

(ang. *logmark*). Do pierwszej transakcji z procesu wsadowego dodaj znacznik z nazwą unikatową dla tego procesu. Dzięki temu gdy zechcesz przywrócić bazę do momentu rozpoczęcia procesu, będziesz mógł użyć znacznika. To proste.

Oto przykładowa sekwencja przywracania dziennika transakcji.

1. Przywracanie pełnej bazy danych w trybie NORECOVERY.
2. Przywracanie różnicowych kopii zapasowych w trybie NORECOVERY.
3. Przywracanie każdego dziennika transakcji w trybie NORECOVERY. Możesz też zastosować klauzulę STOP, aby przywrócić bazę do określonego punktu w czasie.
4. Jeśli dostępna jest końcówka dziennika transakcji, też ją przywróć. Następnie przestaw bazę do trybu RECOVERY.

UWAGA Po przywróceniu bazy danych w ten sposób nie można przywracać innych kopii bez rozpoczęcia całego procesu od początku.

Częściowe przywracanie bazy danych

Częściowa kopia zapasowa obejmuje główną grupę plików, wszystkie grupy plików przeznaczone do odczytu i zapisu oraz określone grupy plików przeznaczone tylko do odczytu. Grupa plików działa w trybie tylko do odczytu, jeśli została tak ustawiona przed utworzeniem jej ostatniej kopii zapasowej. *Przywracanie częściowe* bazy działającej w trybie tylko do odczytu obejmuje tylko główną grupę plików. To podejście stosuje się zwykle dla baz, w których używane są grupy plików tylko do odczytu i, co ważniejsze, duże grupy tego rodzaju można zarchiwizować, aby zwolnić miejsce na dysku.

Obraz częściowej różnicowej kopii zapasowej obejmuje zmiany z głównej grupy plików i modyfikacje z grup przeznaczonych do odczytu i zapisu. Przywracanie częściowej różnicowej kopii zapasowej wymaga użycia obrazu częściowej kopii zapasowej.

Przywracanie plików lub grup plików

Ten proces można nazwać *przywracaniem fragmentarycznym*. Najpierw przywracana jest główna grupa plików, do czego używane jest słowo kluczowe PARTIAL. Następnie należy przywrócić pozostałe grupy plików. Każdą grupę plików (jeśli jest spójna) można udostępnić już w trakcie przywracania pozostałych grup. Dzięki temu administrator może szybciej udostępniać fragmenty bazy bez konieczności oczekiwania na przywrócenie jej w całości.

Poniżej pokazano, jak przywrócić bazę grupy plików po grupie plików. Należy zacząć od głównej grupy plików:

```
RESTORE DATABASE ch17_samples FILEGROUP='PRIMARY' FROM ch17_samples_Backup WITH PARTIAL, NORECOVERY;
```

Następnie przywracana jest grupa FILEGROUP 'ch17_samples_fg_rw' (przeznaczona do odczytu i zapisu):

```
RESTORE DATABASE ch17_samples FILEGROUP='ch17_samples_fg_rw' FROM ch17_samples_Backup WITH NORECOVERY;
```

W ostatnim kroku należy przywrócić grupę FILEGROUP 'ch17_samples_fg_ro'. Jest to grupa tylko do odczytu, dlatego nie wymaga dzienników transakcji:

```
RESTORE DATABASE ch17_samples FILEGROUP='ch17_samples_fg_ro' FROM
ch17_samples_Backup WITH RECOVERY;
```

UWAGA Kopie zapasowe plików i grup plików wymagają, aby stosowany był model FULL lub BULK_LOGGED, co pozwala tworzyć kopie zapasowe dzienników transakcji. Nie jest to konieczne dla baz działających w trybie tylko do odczytu.

Przedstawiony przykładowy kod jest odpowiedni w sytuacjach, gdy na jednym urządzeniu umieszczonych jest wiele kopii zapasowych. Jeśli każda kopia zapasowa zapisana jest w odrębnym pliku, w poleceniach RESTORE należy podawać nazwy plików.

Przywracanie snapshotów baz danych

System SQL Server obsługuje tworzenie snapshotów baz danych. Polega to na utworzeniu przeznaczonej tylko do odczytu kopii bazy z określonego momentu. Plik snapshota początkowo nie zawiera danych, ponieważ stosowana jest tu technologia kopiowania przy pierwszym zapisie. Gdy baza jest modyfikowana, w momencie pierwszej zmiany wartości dawna wartość jest umieszczana w pliku snapshota.

Technika nie jest przeznaczona do przywracania bazy po awariach nośników. Zwykle stosują się ją wtedy, gdy trzeba wprowadzić serię zmian w bazie, a następnie przywrócić pierwotną wersję danych. Jeśli zmian jest niewiele, cały proces przebiega szybko. Ta metoda często jest stosowana w trakcie testów. Można wtedy wprowadzić zmiany w ramach testów, a następnie przywrócić pierwotną wersję bazy. Przywracanie bazy w tym modelu powoduje odtworzenie stanu z momentu wykonania snapshota.

Snapshoty mają ograniczenia. Duże obiekty binarne, grupy plików tylko do odczytu lub skompresowane, pliki offline i używanie kilku snapshotów uniemożliwiają przywracanie baz w ten sposób. Ponadto przywrócenie bazy na podstawie snapshota narusza istniejący wcześniej łańcuch kopii zapasowych. To oznacza, że trzeba następnie utworzyć pełną kopię zapasową (lub kopię zapasową pliku), a dopiero potem można zacząć wykonywać kopie zapasowe dzienników.

Aby utworzyć snapshot bazy danych, zastosuj następującą składnię:

```
CREATE DATABASE ch17_samples_dbss9AM ON (NAME = ch17_samples_Data
, FILENAME = 'C:\Program Files\Microsoft SQL Server
\MSSQL12.MSSQLSERVER\MSSQL\Data\ch17_samples_data')
AS Snapshot of ch17_samples
```

W celu przywrócenia bazy na podstawie snapshota wywołaj poniższy kod:

```
USE MASTER
RESTORE DATABASE ch17_samples
FROM DATABASE_SNAPSHOT='ch17_samples_dbss9AM'
```

Od systemu SQL Server 2005 znacznie usprawniono strukturę raportów z poziomu stron. Błędy stron są obecnie zapisywane w tabeli suspect_pages tabeli msdb. Administrator może teraz nie tylko rejestrować błędy stron, ale też przywracać podejrzane strony. System SQL Server może przywracać strony w czasie, gdy baza pracuje w trybie online i jest dostępna (dotyczy to także plików i grup plików zawierających podejrzane strony). W starszych wersjach systemu SQL Server podejrzane strony można przywracać tylko w trybie offline.

Przywracanie działa tylko dla stron danych. Nie są obsługiwane strony map alokacji, indeksów pełnotekstowych, dzienników transakcji, a także strony ładowania (ang. *boot page*) baz danych i plików. Ponadto przywracanie stron nie działa w modelu SIMPLE.

Proces przywracania stron przebiega podobnie jak dla plików, przy czym trzeba podać numery odtwarzanych stron. Oto używana składnia:

```
RESTORE DATABASE <nazwa_bazy>
PAGE = '<plik:strona>,...'
FROM <plik lub urządzenie z kopią zapasową>
WITH NORECOVERY
```

Należy przywrócić różnicową kopię zapasową, a następnie kopie zapasowe dzienników transakcji, używając przy tym klauzuli `NORECOVERY`. Następnie trzeba utworzyć normalną kopię zapasową dziennika i przywrócić ją:

```
BACKUP LOG <nazwa_bazy_danych> TO <nazwa_pliku>
RESTORE LOG <nazwa_bazy_danych> FROM <nazwa_pliku> WITH RECOVERY
```

Podejrzane strony można ustalić przy użyciu tabeli `suspect_pages` z bazy `msdb`, dziennika błędów systemu SQL Server, śladów zdarzeń systemu SQL Server, poleceń `DBCC` (ang. *Database Console Command*) i dostawcy WMI (ang. *Windows Management Instrumentation*) dla zdarzeń serwera. Przywracanie stron bywa bardzo przydatne. Administrator może szybko przywrócić strony bez konieczności odtwarzania całej bazy danych. Jest to korzystne zwłaszcza po awarii sprzętu (np. przy przejściowych problemach z kontrolerem lub dyskiem).

Przywracanie tabel metadanych

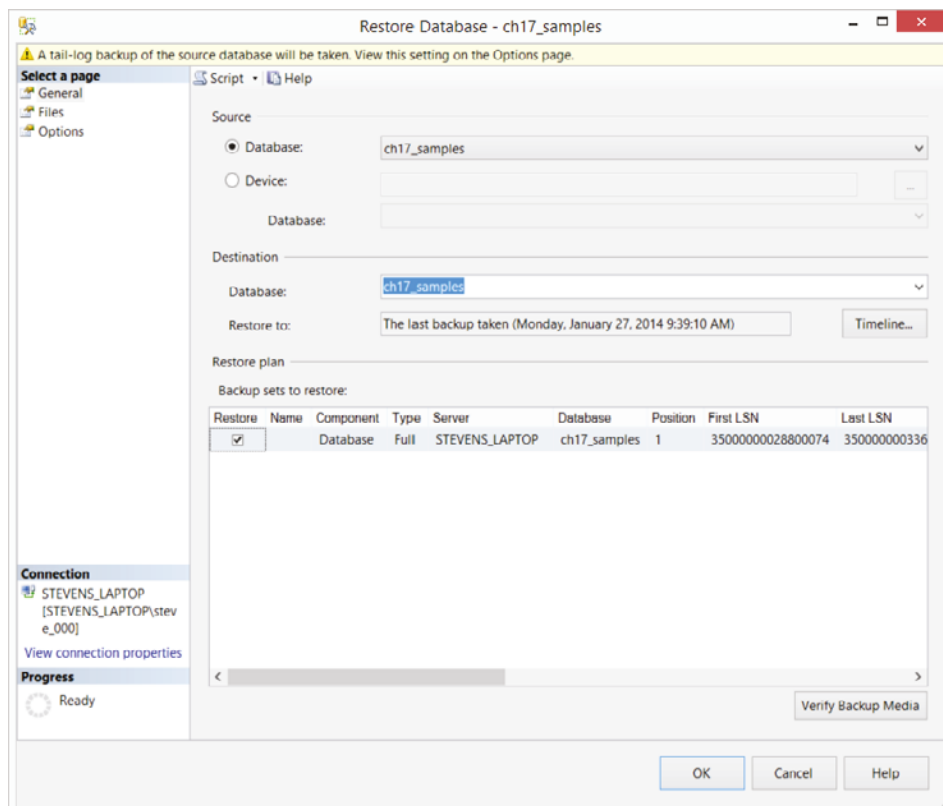
Baza `msdb` zawiera tabele z metadanymi dotyczącymi przywracania. Tabele są przywracane razem z bazą `msdb`. Poniższa lista zawiera opis tych tabel i ich zawartości.

- Tabela `dbo.restorefile` zawiera wiersz dla każdego przywróconego pliku (w tym dla plików przywróconych pośrednio za pomocą grupy plików).
- Tabela `dbo.restorefilegroup` zawiera wiersz dla każdej przywróconej grupy plików.
- Tabela `dbo.restorehistory` zawiera wiersz dla każdej operacji przywracania.

Przywracanie baz za pomocą programu SQL Server Management Studio

Aby przywrócić bazę danych w programie SQL Server Management Studio, wykonaj następujące kroki.

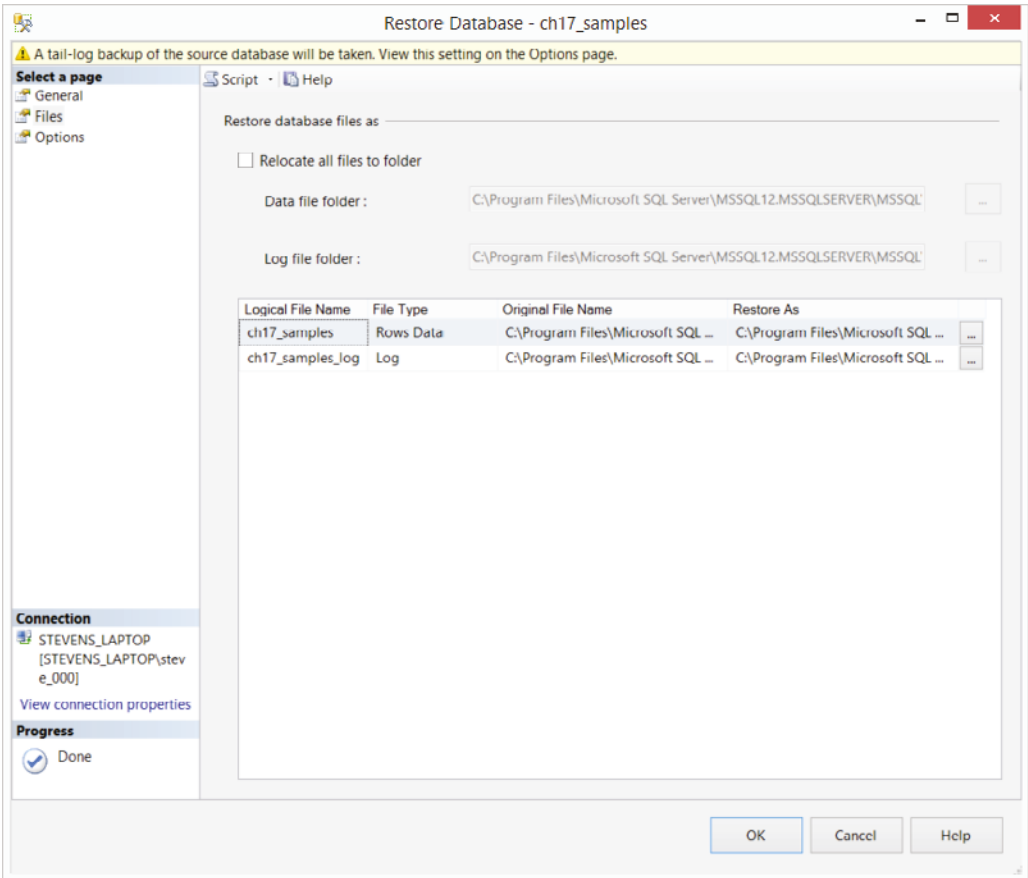
1. Otwórz katalog *Database*, kliknij prawym przyciskiem myszy wybraną bazę i wybierz opcję *Tasks/Restore/Database*. W widocznym na rysunku 17.28 oknie dialogowym *Restore Database* dostępne są funkcje przywracania. Domyślnie otwarta jest zakładka *General*.
2. W sekcji *Source* tego okna dialogowego wybierz jedną z opcji.
 - **Database.** Podaj nazwę przywracanej bazy danych. Ta informacja jest pobierana z tabel z metadanymi kopii zapasowych z bazy `msdb`.
 - **Device.** Podaj urządzenie lub plik z przywracaną kopią zapasową. Tę opcję można wykorzystać do przywrócenia bazy w innym egzemplarzu systemu SQL Server 2014, a także gdy w bazie `msdb` nie ma odpowiednich informacji o kopiach zapasowych.
3. W sekcji *Destination* okna dialogowego *Restore Database* dostępne są następujące ustawienia.
 - **Database.** Tu podaj nazwę istniejącej bazy lub typ nazwy.



Rysunek 17.28. Okno z funkcjami przywracania

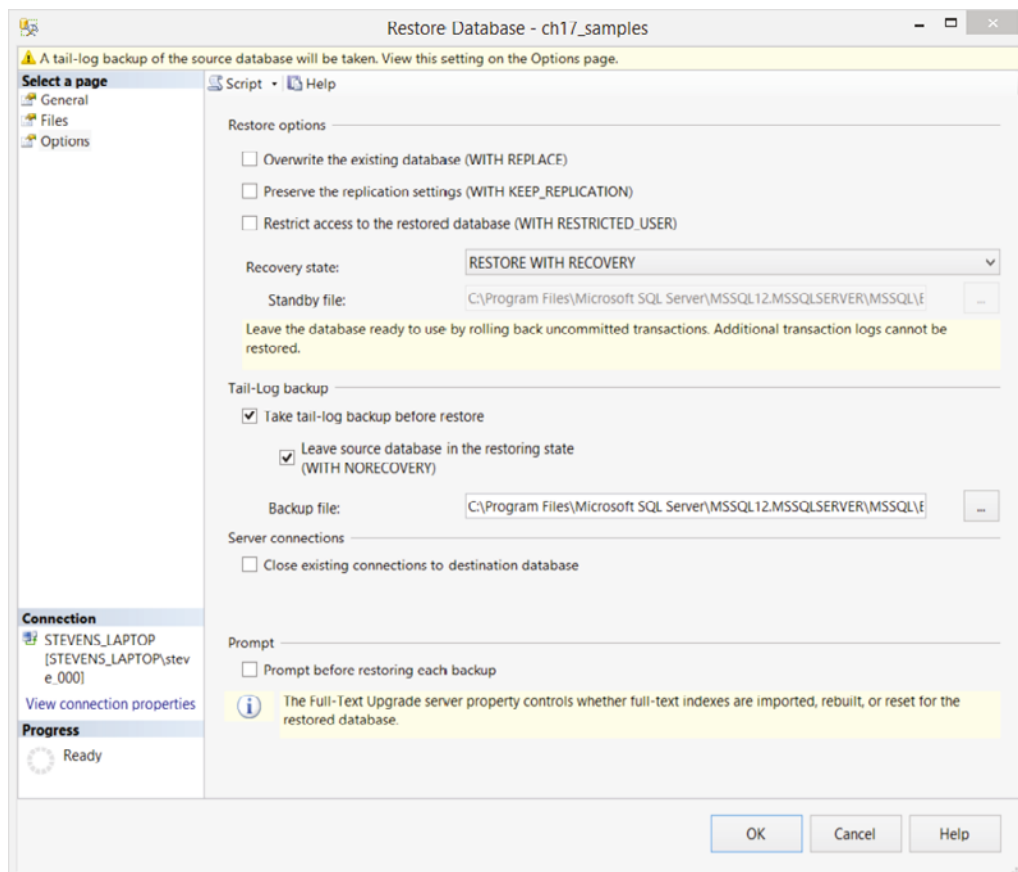
- **Restore to.** Domyślnie używana jest ostatnia kopia zapasowa bazy, jednak za pomocą przycisku *Timeline* można wyświetlić i wybrać inny punkt w czasie, aby przywrócić bazę do stanu z tego momentu. Taki punkt stosuje się zwykle wtedy, gdy nastąpiła błędna modyfikacja danych przez użytkownika lub aplikację i można ustalić jej moment. Wtedy należy przywrócić bazę do czasu sprzed wystąpienia błędu. Technika jest niedostępna w modelu SIMPLE, ponieważ dziennik transakcji jest w nim przycinany.
- 4. Następnie wybierz przywracane zestawy kopii zapasowych z listy widocznej na dole opisywanego okna dialogowego. Gdy ustawisz źródło przywracania, pojawią się zestawy kopii zapasowych dostępnych dla danej bazy. Możesz też wybrać, które zestawy chcesz przywrócić.
- 5. Otwórz zakładkę *Files*. Pojawią się opcje widoczne na rysunku 17.29.
- **Restore database files as.** Tu można ustawić przywrócenie bazy w innym katalogu i z nową nazwą pliku. Jeśli w tym samym katalogu jest tworzona nowa kopia bazy, trzeba zmienić nazwę pliku przywracanej wersji. Jest to odpowiednik opcji *MOVE* z polecenia *Restore Database*. Jeżeli w takiej sytuacji nie zmienisz nazw plików, system SQL Server wygeneruje następujący błąd:

```
Restore failed for Server 'STEVEN'S_LAPTOP'.
(Microsoft.SqlServer.SmoExtended)System.Data.SqlClient.SqlError:
Exclusive access could not be obtained because the database is in
use. (Microsoft.SqlServer.Smo)
```



Rysunek 17.29. Konfigurowanie plików przywracanej bazy

6. Teraz w oknie dialogowym *Restore Database* otwórz po lewej stronie zakładkę *Options*. Pojawi się okno dialogowe widoczne na rysunku 17.30.
7. W sekcji *Restore options* widocznego okna dostępne są następujące opcje.
 - **Overwrite the existing database.** Zaznacz to pole, jeśli w danym egzemplarzu systemu SQL Server znajduje się baza, którą chcesz przywrócić. Jeżeli zaznaczysz to pole, istniejąca baza zostanie zastąpiona. Jest to odpowiednik opcji `REPLACE` w poleceniu `Restore Database`.
 - **Preserve the replication settings.** Zaznacz to pole, gdy przywracasz bazę danych wydawcy. Jest to odpowiednik opcji `PRESERVE_REPLICATION` w poleceniu `Restore Database`.
 - **Restrict access to the restored database.** Zaznacz to pole, jeśli chcesz wykonać dodatkowe operacje na bazie lub sprawdzić jej poprawność przed udostępnieniem danych użytkownikom. Gdy użyjesz tej opcji, dostęp do bazy będzie możliwy tylko za pomocą kont `db_owner`, `dbcreator` i `sysadmin`. To ustawienie jest odpowiednikiem opcji `RESTRICTED_USER` w poleceniu `Restore Database`.



Rysunek 17.30. Zakładka z opcjami przywracania bazy

8. W sekcji *Recovery state* dostępne są następujące opcje.

- **RESTORE WITH RECOVERY.** Jest to ustawienie domyślne. Powoduje, że po przywróceniu bazy danych nie można przywracać innych obrazów z kopiami zapasowymi, a baza jest udostępniana użytkownikom. Jeśli chcesz przywrócić dodatkowe obrazy kopii zapasowych (gdy np. po przywróceniu pełnej bazy danych trzeba odtworzyć dzienniki transakcji), odzyskiwanie należy wykonać na samym końcu, ponieważ po tej operacji w celu przywrócenia dalszych kopii zapasowych cały proces trzeba zacząć od początku. Jest to odpowiednik opcji `WITH RECOVERY` w poleceniu `Restore Database`.
- **RESTORE WITH NORECOVERY.** Przy tym ustawieniu po przywróceniu obrazu z kopią zapasową baza nie jest odzyskiwana, co pozwala uwzględnić inne obrazy z kopią zapasową (np. z kopią różnicową lub dziennika transakcji). Ponadto w trybie `NORECOVERY` baza nie jest dostępna dla użytkowników. Ten stan jest używany na serwerach lustrzanych, a także na serwerach pomocniczych przy przysyłaniu dzienników. To ustawienie jest odpowiednikiem opcji `WITH NORECOVERY` w poleceniu `Restore Database`.

➤ **RESTORE WITH STANDBY.** Przy tym ustawieniu po przywróceniu obrazu z kopią zapasową można przywracać dodatkowe kopie zapasowe, a użytkownicy mają dostęp do bazy, jednak w trybie tylko do odczytu. Aby zachować spójność danych w tym stanie, wycofywane i niezatwierdzone transakcje są zapisywane w pliku rezerwowym, co umożliwia ich późniejsze zatwierdzenie. Możliwe, że chcesz uwzględnić dodatkowe obrazy z kopią zapasową i sprawdzać poprawność danych przed każdą operacją przywracania. Tę opcję często stosuje się na serwerach pomocniczych przy przesyłaniu dzienników, aby umożliwić użytkownikom dostęp do danych na potrzeby generowania raportów. To ustawienie jest odpowiednikiem opcji **WITH STANDBY** w poleceniu **Restore Database**.

9. W sekcji *Tail-Log Backup* w zakładce *Options* można ustawić opcję, która przed przywróceniem bazy powoduje utworzenie kopii zapasowej końcówki dziennika. Jeśli zaznaczysz tę opcję, możesz pozostawić źródłową bazę danych w stanie przywracania po wykonaniu kopii zapasowej końcówki dziennika (w przygotowaniu do przywracania bazy). Dostępna jest też opcja *Backup File*, która pozwala ustawić lokalizację kopii zapasowej końcówki dziennika.
10. W sekcji *Server Connections* w zakładce *Options* można ustawić zamykanie istniejących połączeń z docelową bazą przed rozpoczęciem operacji przywracania.
11. W sekcji *Prompt* w zakładce *Options* można zaznaczyć opcję *Prompt before restoring each backup*, która umożliwi zamianę taśm z zestawami kopii zapasowych.

Polecenie RESTORE z języka T-SQL

Wszystkie polecenia i funkcje przywracania z programu SQL Server Management Studio są dostępne także bezpośrednio w języku T-SQL. Aby np. w prosty sposób przywrócić pełną kopię zapasową bazy danych, użyj następującej składni:

```
RESTORE DATABASE [ch17_samples] FROM DISK = '
C:\Program
Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup\ch17_samples.bak'
```

Poniżej pokazany jest bardziej skomplikowany przykład, z wykorzystaniem pełnej kopii, różnicowej kopii i kopii dziennika transakcji oraz opcji **STOPAT**. Opcja ta umożliwia administratorom zakończenie przywracania w określonym punkcie w czasie, przed momentem, w którym modyfikacje danych spowodowały błąd. Zaleca się stosowanie opcji **STOPAT** dla wszystkich kopii zapasowych dzienników transakcji. Jeśli wskazany punkt znajduje się w kopii zapasowej dziennika transakcji, przywracanie zatrzymuje się w określonym miejscu. Jeżeli przywrócone zostaną nowsze dane, cały proces trzeba zacząć od początku.

-- Przywracanie pełnej kopii zapasowej bazy danych.

```
RESTORE DATABASE ch17_samples FROM ch17_samples_Full_Backup
WITH NORECOVERY;
```

-- Przywracanie różnicowej kopii zapasowej bazy danych.

```
RESTORE DATABASE ch17_samples FROM ch17_samples_Diff_Backup
WITH NORECOVERY;
```

-- Przywracanie dzienników transakcji z opcją **STOPAT**, aby zatrzymać proces w określonym punkcie.

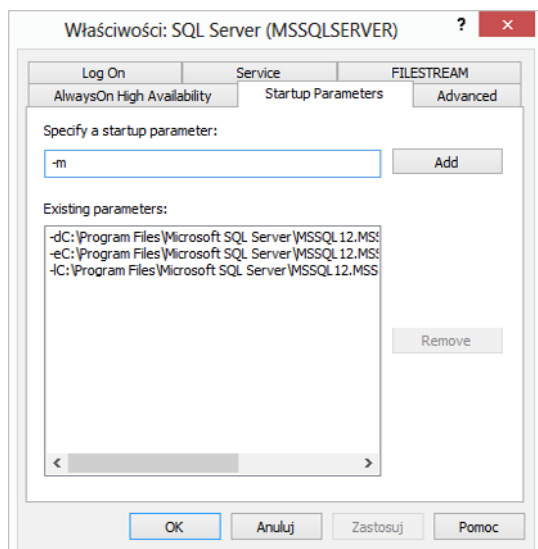
```
RESTORE LOG ch17_samples
FROM ch17_samples_Log1
WITH NORECOVERY, STOPAT = 'Feb 1, 12:00 AM';
RESTORE LOG ch17_samples
FROM ch17_samples_Log2
WITH RECOVERY, STOPAT = 'Feb 1, 2012 12:00 AM';
```

UWAGA Kopie baz danych, dla których włączone jest szyfrowanie, są automatycznie szyfrowane za pomocą tego samego klucza, co sama baza. Przy przywracaniu takich kopii dostępny musi być klucz szyfrowania. Bez klucza dostęp do danych jest niemożliwy. Klucze szyfrowania muszą być przechowywane tak długo, jak długo zachowywane są kopie zapasowe.

Przywracanie systemowych baz danych

Przyczyny uszkodzenia bazy master wpływają na to, jak przebiegać będzie jej przywracanie. Jeśli awaria wymaga zainstalowania nowego egzemplarza systemu SQL Server i dostępna jest świeża kopia zapasowa bazy master, wykonaj następujące kroki.

1. Zainstaluj nowy egzemplarz systemu SQL Server.
2. Uruchom ten egzemplarz.
3. Zainstaluj pakiety SP i inne poprawki.
4. Zatrzymaj usługę SQL Server Agent. Jeśli tego nie zrobisz, może ona zająć jedyne połączenie w trybie z jednym użytkownikiem. Ponadto wyłącz inne usługi, które używają nowego egzemplarza systemu SQL Server, ponieważ także one mogą zająć jedyne dostępne połączenie.
5. Uruchom egzemplarz systemu SQL Server w trybie z jednym użytkownikiem. Można to zrobić na kilka sposobów: za pomocą narzędzia SQL Server Configuration Manager, uruchamiając plik binarny systemu SQL Server z wiersza poleceń, a także skorzystać z usługi SQL Server w programie Usługi systemu Windows. Każda z tych metod wymaga dodania parametru `-m`, co powoduje ustawienie trybu z jednym użytkownikiem, i ponownego uruchomienia systemu. Zalecana technika to użycie programu SQL Server Configuration Manager i znalezienie odpowiedniego egzemplarza systemu SQL Server w węźle *SQL Server Services*. Zatrzymaj znaną usługę SQL Server, a następnie w oknie dialogowym *Właściwości* dodaj parametr `-m` dla tej usługi i ponownie ją uruchom (patrz rysunek 17.31).



Rysunek 17.31. Ustawianie parametru `-m` dla usługi

6. Użyj narzędzia SQLCMD lub innego programu administracyjnego, aby zalogować się do egzemplarza systemu SQL Server za pomocą konta administratora systemu. Przywróć bazę master przy użyciu następującego polecenia:

```
RESTORE DATABASE [MASTER] FROM DISK =
N'C:\Program
Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup\master.bak'
```
7. Jeśli system SQL Server nie rozpoczął pracy, ponieważ baza master jest uszkodzona, a aktualna kopia zapasowa tej bazy nie jest dostępna, trzeba odbudować tę bazę. Uruchom program *setup.exe* systemu SQL Server, aby naprawić systemowe bazy danych.
8. Jeśli aktualna kopia zapasowa bazy master jest dostępna, to po odbudowaniu baz i uruchomieniu systemu SQL Server ustaw egzemplarz systemu SQL Server w tryb z jednym użytkownikiem i przywróć tę bazę. Jeżeli aktualna kopia zapasowa bazy master nie istnieje, wszelkie modyfikacje tej bazy (np. w zabezpieczeniach loginów, punktach końcowych lub serwerach połączonych) zostaną utracone i trzeba je będzie wprowadzić jeszcze raz.
9. Program *setup.exe* w ramach odbudowywania systemowych baz danych tworzy też bazy msdb i model. Jeśli aktualne kopie tych tabel są dostępne, przywróć je. W przeciwnym razie trzeba ponownie wprowadzić wszystkie zmiany z tych tabel. Oto składnia pozwalająca odbudować bazę master:

```
start /wait setup.exe /qn INSTANCENAME=<nazwa_egzemplarza> REINSTALL=SQL_Engine
```
10. W ostatnim kroku podłącz bazy danych użytkowników.

Jeśli uszkodzone są tylko bazy model i msdb, możesz przywrócić je za pomocą aktualnej kopii zapasowej. Jeżeli jest niedostępna, musisz uruchomić program *setup.exe*, który odbuduje wszystkie systemowe bazy danych. Zwykle bazy model i msdb są zapisane na tym samym dysku, co baza master. Dlatego w wyniku uszkodzenia macierzy dyskowej prawdopodobnie wszystkie trzy wymienione bazy zostaną utracone. Aby ograniczyć ryzyko, pomyśl o zastosowaniu macierzy RAID do przechowywania tych baz. Bazy tempdb nie trzeba przywracać, ponieważ jest automatycznie odtwarzana przez system SQL Server przy uruchamianiu. Jest to baza o znaczeniu strategicznym i pojedynczy punkt podatności na awarie egzemplarza systemu SQL Server. Dlatego powinna działać w macierzy dyskowej odpornej na błędy.

Archiwizowanie danych

Archiwizowanie wielu danych z dużych tabel czasem sprawia trudności. Przykładowo pobranie milionów wierszy z tabeli zawierającej miliard wierszy, skopiowanie ich, a następnie usunięcie to długi proces, który może doprowadzić do tworzenia blokad tabeli i zmniejszenia współbieżności, co jest nieakceptowalne, chyba że nikt nie używa danej tabeli. Dlatego często po kolei usuwa się niewielkie zbiory wierszy. Pozwala to uniknąć spadku współbieżności, ponieważ przy mniejszej liczbie wierszy czasem wystarczy blokadę stron i można wykorzystać indeks przyspieszający dostęp do danych.

Wydałym sposobem archiwizowania dużych ilości danych jest zastosowanie partycji z oknem przesuwającym. Metoda ma dwie odmiany — z użyciem partycjonowania tabel w systemie SQL Server lub z wykorzystaniem widoków partycjonowanych.

Podział tabel w systemie SQL Server na partycje

System SQL Server obsługuje podział tabel na partycje. Tabelę można podzielić aż na 15 000 części, z których każda znajduje się w odrębnej grupie plików. Dla każdej z tych grup można niezależnie utworzyć kopię zapasową. Poszczególne grupy plików czasem zapisuje się w innych miejscach. Przykładowo aktualne dane znajdują się na szybkich dyskach (mogą to być nawet dyski SSD), natomiast starsze i archiwalne informacje są przenoszone na wolniejsze i większe dyski, a gdy stają się niepotrzebne, zostają usunięte. Usuwanie partycji to bardzo szybka operacja, która niemal nie wpływa na kwerendy dotyczące aktualnych danych.

Oto przykładowy podział na partycje; każda z nich zawiera dane z jednego miesiąca. Gdy pojawiają się dane z następnego miesiąca, tworzona jest nowa pusta partycja. Co miesiąc najstarsza partycja jest usuwana z tabeli i przenoszona do tabeli z danymi archiwalnymi. Oto podstawowe kroki potrzebne do utworzenia partycji.

1. Utwórz funkcję partycjonującą, która określa sposób podziału danych na partycje.
2. Przygotuj schemat partycjonowania, który odwzoruje poszczególne elementy na grupy plików.
3. Utwórz tabele za pomocą schematu partycjonowania.

W poniższym przykładowym kodzie pokazano, jak utworzyć tabelę z partycjami z wykorzystaniem okna przesuwanego obejmującego dane z jednego miesiąca:

-- Tworzenie funkcji partycjonującej.

```
CREATE PARTITION FUNCTION [OrderDateRangePFN](datetime)
AS RANGE RIGHT
FOR VALUES (N'2014-01-01 00:00:00'
, N'2014-02-01 00:00:00'
, N'2014-03-01 00:00:00'
, N'2014-04-01 00:00:00');
```

-- Tworzenie schematu partycjonowania.

```
CREATE PARTITION SCHEME [OrderDatePScheme]
AS PARTITION [OrderDateRangePFN]
TO ([filegroup1], [filegroup2], [filegroup3], [filegroup4], [filegroup5]);
-- Tworzenie podzielonej na partycje tabeli SalesOrderHeader.
```

```
CREATE TABLE [dbo].[SalesOrderHeader](
    [SalesOrderID] [int] NULL,
    [RevisionNumber] [tinyint] NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [DueDate] [datetime] NOT NULL,
    [ShipDate] [datetime] NULL,
    [Status] [tinyint] NOT NULL
) ON [OrderDatePScheme]([OrderDate]);
```

W tym przykładzie każda partycja jest zapisana w innej grupie plików. Podział tabeli na partycje i jej scalanie wymaga przenoszenia danych. Jeśli partycje znajdują się w tej samej grupie plików, można szybko rozdzielać i scalać dane bez blokowania tabeli lub zmniejszania poziomu współbieżności. W takiej sytuacji przełączenie lub scalanie partycji wymaga tylko zmiany schematu i nie zajmuje dużo czasu. Aby obsługa partycji była szybka, trzeba też spełnić kilka innych warunków, jednak najważniejsze jest właśnie umieszczenie partycji w jednej grupie plików.

Widoki podzielone na partycje

Technika była dostępna już w starszych wersjach systemu SQL Server. Polega na wykorzystaniu podzielonego na partycje widoku do grupowania niezależnych identycznych tabel (np. nowej tabeli dla danych z każdego miesiąca). Oto potrzebna procedura.

1. Utwórz identyczne tabele z ograniczeniem CHECK, aby ograniczyć dane, które mogą znajdować się w każdej z nich.
2. Utwórz widok łączący wszystkie te tabele.
3. Wczytaj dane za pomocą widoku podzielonego na partycje. SQL Server uwzględni ograniczenia, aby wstawić dane z właściwej tabeli.
4. Przed rozpoczęciem następnego okresu utwórz nową tabelę z ograniczeniem opartym na datach i dodaj ją do definicji widoku. Następnie wczytaj aktualne dane za pomocą widoku.
5. W trakcie archiwizacji usuń najstarszą tabelę z definicji widoku, a następnie ją zarchiwizuj. Każdą tabelę można umieścić w odrębnej grupie plików i osobnej kopii zapasowej.

W tej technice nie ma ograniczenia do 15 000 partycji, jednak wymaga więcej pracy, ponieważ każda tabela jest niezależna i trzeba nią osobno zarządzać.

Podsumowanie

Tworzenie kopii zapasowych i odzyskiwanie baz to ostatnia linia obrony dla firmy, która chce odtworzyć dane w sytuacji, gdy wszystko inne zawiodło. Mechanizmy tworzenia kopii zapasowych i przywracania stanu muszą gwarantować, że można będzie odzyskać zbierane przez wiele lat informacje o klientach, wzorce kupowania, dane finansowe i dane z inwentaryzacji.

SQL Server 2014 to system zarządzania relacyjnymi bazami danych, który jest skalowalny i zapewnia wysoką dostępność. Jest używany w największych bazach danych z bardzo dużą liczbą równoległych użytkowników korzystających z aplikacji o znaczeniu strategicznym. Ważne mechanizmy tworzenia kopii zapasowych i przywracania stanu pozwalają na łatwiejsze zarządzanie dużymi bazami danych.

Jeśli uważnie przeczytałeś ten rozdział, wiesz już, jak opracować solidny plan tworzenia kopii zapasowych danych firmy, a także masz gotowe dokumenty z planem odzyskiwania baz.

W rozdziale 18. dowiesz się, jak działa przesyłanie dzienników w systemie SQL Server 2014.

Przesyłanie dzienników w systemie SQL Server 2014

ZAWARTOŚĆ ROZDZIAŁU:

- Wprowadzenie do przesyłania dzienników.
- Praktyczne zastosowanie tej techniki.
- Omówienie architektury przesyłania dzienników.
- Wprowadzenie do wdrażania przesyłania dzienników.
- Zmiany ról przy przesyłaniu dzienników.
- Monitorowanie systemu i rozwiązywanie problemów.

Przesyłanie dzienników to tania, wydajna i prosta technika z systemu SQL Server, wprowadzona już w jednej z wczesnych wersji tego narzędzia. Bardzo pomaga zachować ciągłość funkcjonowania firmy. Przy przesyłaniu dzienników dziennik transakcji z bazy danych z jednego systemu SQL Server jest kopiowany i przywracany w pomocniczym systemie, gdzie jest stosowany, by zapewnić wysoką dostępność, a także podczas generowania raportów i odzyskiwania systemu po katastrofie. Od wersji SQL Server 2005 przesyłanie dzienników pomaga w zagwarantowaniu ciągłości pracy firmy. Ponadto nadal jest jednym z mechanizmów zapewniania wysokiej dostępności za pomocą aktywnego serwera rezerwowego. Na serwerze pomocniczym przesłane dzienniki są też wykorzystywane przy przełączaniu awaryjnym.

W tym rozdziale omawiamy architekturę przesyłania dzienników i scenariusze stosowania tej techniki. Wyjaśniamy ponadto, jak skonfigurować przesyłanie dzienników, i prezentujemy różne sytuacje związane ze zmianą ról serwerów (głównego i pomocniczych). Tu dowiesz się też, jak rozwiązywać problemy z przesyłaniem dzienników i jak zintegrować ten mechanizm z innymi technikami zapewniania wysokiej dostępności.

Scenariusze stosowania przesyłania dzienników

Przy przesyłaniu dzienników wykorzystywane są dostępne w systemie SQL Server mechanizmy tworzenia kopii zapasowych dzienników transakcji i ich przywracania. Dwa systemy SQL Server uczestniczące w przesyłaniu dzienników mogą być zlokalizowane obok siebie (aby zapewniać wysoką dostępność) lub w odległych miejscach (na potrzeby odzyskiwania systemu po katastrofie). Jedyne ograniczenie dotyczące odległości związane jest z tym, aby między serwerami istniało połączenie umożliwiające pomocniczemu systemowi SQL Server skopiowanie dziennika transakcji i przywrócenie go. Przesyłanie dzienników jest stosowane w trzech różnych scenariuszach. Oto one.

- **Tworzenie aktywnego serwera rezerwowego.** Kopia zapasowa bazy danych jest przechowywana w tej samej fizycznej lokalizacji jako zabezpieczenie przed awarią głównego serwera.
- **Mechanizm odzyskiwania systemu po katastrofie.** Dwa serwery znajdują się w różnych lokalizacjach. Jest to ochrona przed sytuacją, gdy w lokalnej jednostce, gdzie pracuje główny serwer, wydarzy się katastrofa.
- **System do generowania raportów.** Serwer pomocniczy jest wykorzystywany do generowania raportów.

Przesyłanie dzienników w celu utworzenia aktywnego serwera rezerwowego

Aktywny serwer rezerwowy wymaga utworzenia pełnej kopii zapasowej i okresowego wykonywania kopii zapasowych dzienników transakcji na serwerze głównym. Potem te kopie zapasowe są przywracane (w odpowiedniej kolejności) na serwerze rezerwowym. Między operacjami przywracania kopii serwer rezerwowy pracuje w trybie tylko do odczytu. Gdy serwer rezerwowy trzeba udostępnić użytkownikom, zaległe kopie zapasowe dzienników transakcji (w tym kopia zapasowa aktywnego dziennika transakcji z serwera głównego) są na nim przywracane i odzyskiwana jest baza danych. Przesyłanie dzienników często stosuje się przy tworzeniu aktywnego serwera rezerwowego, gdy serwer pomocniczy jest zlokalizowany blisko serwera głównego. Jeśli serwer główny zostanie wyłączony (planowo lub nieoczekiwanie), serwer pomocniczy może przejąć jego zadania i zapewnić ciągłość pracy firmy. Następnie administrator może zdecydować o ponownym przełączeniu systemu na serwer główny, gdy ten stanie się dostępny.

Skonfigurowanie aktywnego serwera rezerwowego korzystającego z przesyłania dzienników jest proste, ponieważ używane są solidne kopie zapasowe dzienników transakcji, mechanizm kopiowania plików z systemu operacyjnego i przywracanie dzienników transakcji. W większości scenariuszy należy zaplanować częste wykonywanie zadania przesyłania dzienników, aby serwer pomocniczy był dobrze zsynchronizowany z serwerem głównym. Pomaga to skrócić czas zmiany ról serwerów i ograniczyć utratę danych. Aby jeszcze lepiej chronić dane, w sytuacji gdy dostępna jest aktywna część dziennika transakcji z serwera głównego, na serwerze pomocniczym można przywrócić bazę do momentu wystąpienia awarii na serwerze głównym.

Jednak w niektórych sytuacjach aktywna część dziennika transakcji jest niedostępna, ponieważ obszar pamięci z dziennikiem transakcji jest uszkodzony lub pliki dziennika transakcji nie dotarły do serwera pomocniczego. Przy zmianie ról serwerów odzyskiwane są wszystkie przesyłane dzienniki transakcji i aktywna część dziennika transakcji. Ponadto po zmianie ról trzeba przekierować użytkowników na serwer pomocniczy, ponieważ w tym podejściu (inaczej niż w klastrach WSFC i innych mechanizmach systemu SQL Server zapewniających wysoką dostępność) nie odbywa się to automatycznie.

Czasem przy przełączaniu awaryjnym przesyłanie dzienników stosuje się zamiast klastrów WSFC, ponieważ jest to mniej kosztowne rozwiązanie. Klasytry wymagają np. współużytkowanego systemu dyskowego, a możliwe, że firma go nie posiada. Przy przesyłaniu dzienników nie ma tego rodzaju wymagań sprzętowych, dlatego obecny sprzęt używany w organizacji często można wykorzystać przy przesyłaniu dzienników, nawet jeśli nie jest odpowiedni do pracy w klastrze. Ponadto przy przesyłaniu dzienników bazy główna i pomocnicza znajdują się na odrębnych serwerach. Jest to środowisko bez współużytkowanych elementów. W klastrach WSFC potrzebny jest współużytkowany system dyskowy z jedną kopią bazy danych. Taki system jest narażony na awarie.

UWAGA Przełączanie awaryjne przy przesyłaniu dzienników to (inaczej niż w klastrach) proces zawsze wykonywany ręcznie. Oznacza to, że trzeba zainicjować przełączanie awaryjne, monitorować stan tego procesu i ustawić nowy serwer główny w łańcuchach połączeń w aplikacjach klienckich. W klastrach WSFC z systemem SQL Server monitorowanie i przełączanie awaryjne odbywa się automatycznie. Gdy przy przesyłaniu dzienników ustalisz, że trzeba przełączyć system awaryjnie na serwer pomocniczy, możesz wykonać operacje ręcznie lub automatycznie. Jeśli chcesz zastosować podejście automatyczne, musisz utworzyć odpowiednie skrypty. Przykładowe skrypty tego rodzaju przedstawiono dalej w tym rozdziale.

Inna różnica między przesyłaniem dzienników a klastrami polega na tym, że klasytry chronią cały egzemplarz systemu SQL Server. W klastrach uwzględniane są wszystkie bazy danych z egzemplarza. Przesyłanie dzienników działa dla poszczególnych baz użytkowników. Jeśli musisz zapewnić wysoką dostępność wszystkich baz z serwera, możesz zastosować albo klasytry, albo przesyłanie dzienników. Jeżeli jednak wybierzesz przesyłanie dzienników, musisz skonfigurować je dla każdej bazy z serwera. Gdy chcesz zapewnić wysoką dostępność tylko wybranych baz z serwera, przesyłanie dzienników jest dobrym rozwiązaniem.

Przesyłanie dzienników jako mechanizm odzyskiwania systemu po katastrofie

Jeśli nawet firma stosuje już rozwiązanie zapewniające wysoką dostępność (klastry WSFC lub przesyłanie dzienników), warto skonfigurować także mechanizm obejmujący zdalną lokalizację. Gdy przesyłasz dzienniki na serwer pomocniczy w zdalnej lokalizacji, możesz zabezpieczyć firmę przed awarią sieci zasilania lub lokalną katastrofą.

Jeśli nie można użyć dzienników transakcji z katalogu z kopią zapasową lub aktywnego dziennika transakcji (gdy np. z powodu awarii zasilania nie da się uruchomić serwera głównego), możesz połączyć aktywny dziennik transakcji serwera głównego z plikami dzienników transakcji za pomocą programu do analiz takich dzienników. Pozwala to zidentyfikować niewykonane transakcje i wykonać je ręcznie. Jednak plan zapewniania dostępności i odzyskiwania systemu nie powinien zależeć od takich programów. Pliki dzienników transakcji zapisane przez zadanie tworzące kopie zapasowe należy zarchiwizować, aby umożliwić odzyskanie systemu z serwera głównego, gdy użytkownik błędnie zmodyfikuje dane lub nastąpi podobne wydarzenie.

Ponadto archiwizowanie dzienników transakcji razem z pełnymi kopiami zapasowymi bazy zapewnia inną możliwość odzyskania systemu po katastrofie. Kiedy chcesz kontrolować czas usuwania plików dzienników transakcji, tak aby mechanizm tworzenia kopii zapasowych w systemie operacyjnym mógł je zapisać zgodnie ze swoim harmonogramem, ustaw opcję

Delete Files Older Than na czas dłuższy niż okres używany w tym programie. Opcję znajdziesz na ekranie *Transaction Log Backup Settings*. Jeśli np. w systemie operacyjnym tworzenie kopii zapasowych jest ustawione na każdą noc, tak skonfiguruj opcję *Delete Files Older Than*, aby pliki zostały zachowane do czasu zakończenia zapisywania wspomnianych kopii.

Problemem w tym scenariuszu jest to, że sieć musi być wystarczająco wydajna, aby umożliwić przesyłanie dużych dzienników do zdalnej lokalizacji. Ponadto gdy wydarzy się katastrofa, może się okazać, że niektóre przesyłane właśnie pliki nie dotrą na serwer pomocniczy. Jeśli nawet sieć jest wystarczająco szybka, by pozwolić na płynny transfer dzienników, w trakcie katastrofy wydajność sieci czasem jest ograniczana przez inne czynniki. Może to doprowadzić do utraty danych.

Bazy danych są synchronizowane za pomocą dzienników transakcji. W trakcie katastrofy dane z dziennika transakcji są zagrożone. W aplikacjach o znaczeniu strategicznym, w których należy zminimalizować utratę danych, konieczne może być wybranie innego rozwiązania. Przykładowo za pomocą kopii lustrzanych lub grup dostępności można synchronizować transakcje bezpośrednio po ich zatwierdzeniu, bez oczekiwania na utworzenie kopii zapasowej dziennika transakcji. Gdy korzystasz z przesyłania dzienników, musisz liczyć się z utratą większej ilości danych z powodu brakującego pliku dziennika lub niedostępności katalogu z kopią zapasową.

Przesyłanie dzienników jako mechanizm tworzenia baz potrzebnych do generowania raportów

Spśród trzech scenariuszy, w których stosuje się przesyłanie dzienników, przy tworzeniu baz do generowania raportów technika ta jest najmniej przydatna. Ma jednak pewne zalety. Przesyłanie dzienników jest tanie, nie wymaga kosztownego sprzętu oraz jest proste w implementacji i obsłudze. Dlatego w niektórych sytuacjach można wykorzystać bazę z serwera pomocniczego do generowania raportów (pod warunkiem, że tryb przywracania ustawiony dla tej bazy to *STANDBY*). Używanie tego serwera do generowania raportów związane jest jednak z pewnymi wadami.

Proces przywracania wymaga dostępu do bazy w trybie na wyłączność. Gdy użytkownicy generują raporty, proces przywracania nie odtworzy danych. Zadanie będzie wtedy oczekiwać na następny okres, w którym zaplanowane jest przywracanie. Może to spowodować wysyłanie alertów o nieaktualnych danych na serwerze pomocniczym. Ponadto w momencie zmiany ról część dzienników transakcji może być jeszcze nieuwzględniona, ponieważ generowanie raportów nie umożliwiło ich przywrócenia. Wydłuża to czas przełączania ról z powodu konieczności przywrócenia dzienników.

Możesz też tak skonfigurować przesyłanie dzienników, aby użytkownicy byli rozłączani, gdy system chce przywrócić dzienniki. Jednak to rozwiązanie spowoduje, że długie raporty pozostaną nieukończone. Jeśli np. przywracanie odbywa się co 10 minut, a ukończenie raportu wymaga 30 minut, raport nigdy nie zostanie wygenerowany, ponieważ mechanizm przesyłania dzienników co 10 minut przerwie ten proces. Aby zwiększyć prawdopodobieństwo wygenerowania całego raportu, należy ustawić dłuższe odstępy między operacjami przywracania. To z kolei sprawia, że serwer pomocniczy ma większe opóźnienie w stosunku do serwera głównego. Ponadto dane w raportach nie są wtedy w pełni aktualne, a schematu bazy z serwera pomocniczego nie można zoptymalizować pod kątem raportów, ponieważ baza działa w trybie tylko do odczytu. Jeśli np. w bazie do generowania raportów przydatne są określone indeksy, trzeba je utworzyć w bazie z serwera głównego, gdzie te indeksy mogą obniżyć wydajność.

Dlatego przesyłanie dzienników na potrzeby generowania raportów związane jest z pewnymi problemami i w niektórych środowiskach nie jest dobrym rozwiązaniem. Jeśli raporty są generowane rzadko, to — pod warunkiem, że organizacja godzi się z pewnymi trudnościami

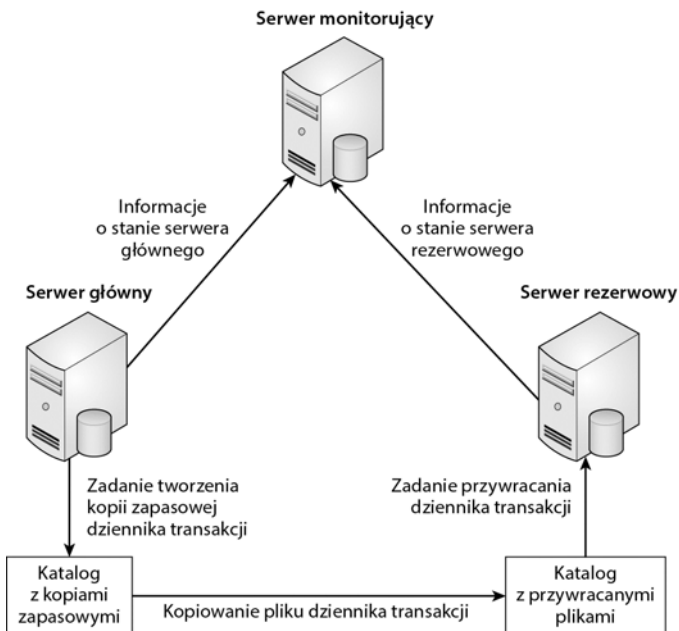
— można zastosować przesyłanie dzienników. Jednak z raportami lepiej współdziałają grupy dostępności, pozwalające na generowanie raportów za pomocą baz danych z replik (więcej informacji o grupach dostępności zamieszczono w rozdziale 25.). Możesz też zastosować replikację transakcyjną, co zapewnia współbieżność, precyzję i synchronizację niemal w czasie rzeczywistym, a także dodatkowo umożliwia modyfikowanie schematu bazy danych. Omówienie replikacji transakcyjnej znajdziesz w rozdziale 15.

UWAGA Konieczność dostępu do bazy w trybie na wyłączność może okazać się poważną wadą mechanizmu przesyłania dzienników. Jeśli przywracasz dzienniki co godzinę, przed wykonaniem tej operacji trzeba zamknąć wszystkie procesy generujące raporty. Jest to niewygodne dla użytkowników i może powodować, że nie będą stosować przesyłania dzienników na potrzeby tworzenia raportów. Jeśli jednak opisane ograniczenia nie przeszkadzają, technika ta może okazać się przydatna.

Architektura mechanizmu przesyłania dzienników

Na rysunku 18.1 przedstawiono podstawową architekturę mechanizmu przesyłania dzienników. Architektura ta wymaga trzech serwerów:

- serwera głównego,
- serwera pomocniczego (czyli serwera rezerwowego),
- serwera monitorującego (który jest opcjonalny).



Rysunek 18.1. Architektura przesyłania dzienników

Serwer główny

Serwer główny to serwer produkcyjny, z którym użytkownicy nawiązują połączenie i w którym wykonują operacje. Na tym serwerze działa system SQL Server 2014, który trzeba zabezpieczyć na wypadek awarii sprzętu, błędów oprogramowania, katastrof naturalnych lub błędów spowodowanych przez użytkowników (takich jak przypadkowe skasowanie danych). Możesz skonfigurować serwer pomocniczy w taki sposób, aby w reakcji na błędy użytkowników po pewnym czasie przywracał dziennik. Przy przesyłaniu dzienników serwer główny uruchamia zadanie narzędzia SQL Server Agent, które zapisuje kopię zapasową dziennika transakcji w pliku. Aby przesyłanie dzienników działało prawidłowo, na serwerze trzeba zastosować właściwy model odzyskiwania (BULK_LOGGED lub FULL). Odpowiedni model można ustawić we właściwościach bazy danych w programie SQL Server Management Studio.

Serwer pomocniczy

Serwer pomocniczy zawiera rezerwowy egzemplarz systemu SQL Server z kopią bazy z serwera głównego. W serwerze pomocniczym używa się narzędzia SQL Server Agent do kopiowania pliku dziennika transakcji z katalogu z kopią zapasową (gdzie plik został umieszczony przez serwer główny) i przywracania kopii zapasowej dziennika. Skonfigurowane są tu dwa zadania narzędzia SQL Server Agent — jedno kopiuje plik z dziennikiem transakcji ze współużytkowanego katalogu z kopią zapasową, a drugie przywraca dziennik transakcji. Serwer ten powinien mieć wydajność podobną do serwera głównego, aby po przełączeniu awaryjnym komfort pracy użytkowników został zachowany. Przy przesyłaniu dzienników można stosować kilka serwerów pomocniczych. Jeden może być używany jako aktywny serwer rezerwowy, inny do generowania raportów, a w jeszcze innym można przywracać dziennik transakcji z opóźnieniem.

Dla pomocniczej bazy danych możesz ustawić opcje odzyskiwania **STANDBY** lub **NORECOVERY**.

- **STANDBY.** Zapewnia użytkownikom dostęp do bazy w trybie tylko do odczytu między operacjami przywracania dzienników transakcji. To oznacza, że zadanie wymagające dostępu w tym trybie można obsługiwać za pomocą serwera pomocniczego. Aby jednak przywracanie dzienników transakcji zakończyło się powodzeniem, w czasie tej operacji odczyt jest niemożliwy.
- **NORECOVERY.** Przy tej opcji baza nie jest dostępna nawet w trybie tylko do odczytu.

Serwer monitorujący

Serwer monitorujący przy przesyłaniu dzienników jest opcjonalny, ale warto go stosować. Powinien to być odrębny serwer fizyczny, co pozwala uniknąć powstania pojedynczego punktu podatności na awarię na serwerach głównym lub pomocniczym. Na serwerze monitorującym może działać dowolna wersja systemu SQL Server (w tym SQL Server Express). Gdy serwer monitorujący jest używany, zarządza zadaniami zbierającymi informacje (np. o czasie utworzenia ostatniej kopii zapasowej dziennika transakcji na serwerze głównym, czasie ostatniego przywracania dzienników na serwerze pomocniczym i czasie między tymi operacjami). Serwer monitorujący może też wysyłać alerty na pager lub na adres e-mail operatora, gdy przekroczone zostaną ustalone limity czasu. Jeden serwer może monitorować wiele środowisk, w których używane jest przesyłanie dzienników.

Zaleca się stosowanie odrębnego fizycznego serwera monitorującego. Jeśli serwer monitorujący działa na jednej maszynie razem z serwerem głównym lub pomocniczym, awaria takiego komputera powoduje jednocześnie utratę serwera monitorującego. Przesyłanie dzienników działa także bez serwera monitorującego; wtedy do monitorowania można wykorzystać widoki DMV.

Proces przesyłania dzienników

Na używanych serwerach narzędzie SQL Server Agent wykonuje procesy odpowiedzialne za przesyłanie dzienników. Poniżej opisane są trzy podstawowe procesy.

- **Tworzenie kopii zapasowej dziennika transakcji na serwerze głównym.** Zadanie narzędzia SQL Server Agent na serwerze głównym tworzy kopie zapasowe dziennika transakcji w odstępach czasu ustawionych przez użytkownika. Są one zapisywane w pliku w katalogu przeznaczonym na kopie zapasowe. Domyślnie do nazw plików dodawany jest czas (np. *nazwa_bazy_rrrrmmddggmmss.trn*), co zapewnia ich unikatowość. Domyślna nazwa wspomnianego zadania to *LSBackup_nazwa_bazy_danych*. To zadanie uruchamia polecenie systemu operacyjnego, które tworzy kopię zapasową dziennika transakcji:

```
"C:\Program Files\Microsoft SQL Server\120\Tools\Binn\sqllogship.exe"  
-Backup 0E5D9AA6-D054-45C9-9C6B-33301DD934E2 -server SQLServer1
```

- **Kopiowanie dziennika transakcji na serwer pomocniczy.** Zadanie narzędzia SQL Server Agent na serwerze pomocniczym wykorzystuje nazwę UNC lub dysk współużytkowany, aby uzyskać dostęp do katalogu z kopiami zapasowymi z serwera głównego i skopiować plik dziennika transakcji do lokalnego katalogu z serwera pomocniczego. Domyślnie zadanie kopiowania nosi nazwę *LSCopy_nazwa_serwera_nazwa_bazy_danych* i wykonuje polecenie systemu operacyjnego, które kopiuje plik dziennika transakcji:

```
"C:\Program Files\Microsoft SQL Server\120\Tools\Binn\sqllogship.exe"  
-Copy F2305BFA-B9E3-4B1C-885D-3069D0D11998 -server SQLServer1\SQLServer2
```

- **Przywracanie dziennika transakcji na serwerze pomocniczym.** Zadanie narzędzia SQL Server Agent na serwerze pomocniczym przywraca dziennik transakcji na tym serwerze. Aby możliwe było przywrócenie dziennika, baza musi pracować w trybie *STANDBY* lub *NORECOVERY*. Domyślna nazwa tego zadania to *LSRestore_nazwa_serwera_nazwa_bazy_danych*. Wykonuje ono polecenie systemu operacyjnego, które przywraca plik dziennika transakcji:

```
"C:\Program Files\Microsoft SQL Server\120\Tools\Binn\sqllogship.exe"  
-Restore F2305BFA-B9E3-4B1C-885D-3069D0D11998 -server SQLServer1\SQLServer2
```

UWAGA Większość obiektów związanych z przesyłaniem dzienników znajduje się w bazie *msdb*. Informacje na temat tych obiektów znajdziesz w dokumentacji Books Online poświęconej systemowi SQL Server 2014.

Wymagania systemowe

Serwery używane przy przesyłaniu dzienników muszą spełniać minimalne wymagania sprzętowe systemu SQL Server 2014 (znajdziesz je w dokumentacji Books Online poświęconej temu systemowi). Ponadto potrzebne są określone komponenty w infrastrukturze. W następnych punktach opisujemy te komponenty i wyjaśniamy, jak je zapewnić.

Sieć

Systemy SQL Server używane przy przesyłaniu dzienników muszą pracować w sieci. Serwer główny potrzebuje dostępu do katalogu przeznaczonego na kopie zapasowe, a serwer pomocniczy musi mieć możliwość skopiowania plików dzienników transakcji z tego katalogu do folderu

lokalnego. Ponadto serwer monitorujący potrzebuje połączenia z serwerami głównym i pomocniczym. Aby usprawnić kopiowanie plików dzienników transakcji w środowisku używającym omawianej techniki, umieść serwery w odrębnym segmencie sieci i zastosuj dodatkowe karty sieciowe do obsługi przesyłania dzienników. Przesyłanie dzienników może działać w każdej stosunkowo szybkiej sieci. W wolnych sieciach transfer plików trwa dłużej, a serwer pomocniczy jest aktualizowany z większym opóźnieniem.

Serwery o identycznych możliwościach

Serwery główny i pomocniczy powinny mieć identyczne parametry, aby po przełączeniu awaryjnym serwer pomocniczy mógł przejąć zadania i zapewnić tę samą wydajność oraz taki sam komfort pracy użytkownikom. Ponadto niektóre firmy muszą spełnić warunki umowy o gwarantowanym poziomie świadczonych usług. Taka umowa może wymagać, aby po przełączeniu awaryjnym wydajność systemu była taka sama jak normalnie. Wtedy serwer pomocniczy musi mieć te same parametry, co serwer główny. Niektóre firmy używają wolniejszych serwerów pomocniczych. Trzeba ustalić wymagania organizacji i odpowiednio dostosować do nich serwer pomocniczy.

Przechowywanie danych

Klasy WSFC wymagają infrastruktury ze współużytkowanym dyskiem. Przy przesyłaniu dzienników nie jest to konieczne ani nawet zalecane. Aby dysk z danymi nie stał się pojedynczym punktem podatności na awarię, serwery główny i pomocniczy nie powinny korzystać z tego samego systemu dyskowego. Gdy omawiana technika ma służyć do odzyskiwania systemu po katastrofie, serwery główny i pomocniczy są umieszczone daleko od siebie, dlatego zwykle nie korzystają z tego samego systemu dyskowego. Ustal, ile pamięci potrzeba na kopie zapasowe dzienników transakcji, aby uniknąć problemu zbyt małej ilości miejsca na dysku.

Ponadto przy szacowaniu wymaganej wydajności systemów dyskowych uwzględnij operacje wejścia-wyjścia wykonywane przy przesyłaniu dzienników. Możesz wykorzystać narzędzie SQLIO Disk Subsystem Benchmark do ustalenia wydajności wejścia-wyjścia w danym systemie dyskowym. Narzędzie to jest dostępne na stronie <http://www.microsoft.com/en-us/download/details.aspx?id=20163>.

Gdy ustawiasz przesyłanie dzienników za pomocą programu SQL Server Management Studio, konto usługi SQL Server i narzędzia SQL Server Agent (lub jednostka pośrednicząca uruchamiająca zadanie tworzenia kopii zapasowej) muszą mieć uprawnienia do odczytu i zapisu katalogu z kopiami zapasowymi. Jeśli to możliwe, ten katalog powinien znajdować się w odpornym na błędy systemie dyskowym, tak aby awaria dysku nie spowodowała utraty wszystkich plików dzienników transakcji.

Oprogramowanie

Przesyłanie dzienników jest obsługiwane przez następujące edycje systemu SQL Server:

- SQL Server 2014 Enterprise Edition,
- SQL Server 2014 Standard Edition,
- SQL Server 2014 Business Intelligence Edition.

Dla serwera monitorującego można zastosować dowolną edycję (w tym SQL Server Express).

Serwery używane przy przesyłaniu dzienników muszą mieć identyczne ustawienia uwzględniania wielkości znaków. Ponadto bazy danych muszą działać w trybie odzyskiwania FULL lub BULK_LOGGED.

Wdrażanie przesyłania dzienników

Zanim rozpoczniesz proces wdrażania przesyłania dzienników, musisz przygotować wstępną konfigurację. Następnie możesz wybrać sposób wdrażania: za pomocą programu SQL Server Management Studio lub przy użyciu skryptów w języku T-SQL. Administratorzy zwykle do konfigurowania przesyłania dzienników używają programu SQL Server Management Studio, a następnie generują skrypty w języku T-SQL, co pozwala później ponowić proces. W tym miejscu omawiamy obie procedury.

Wstępne konfigurowanie

Przed wdrożeniem przesyłania dzienników trzeba przygotować określone katalogi używane do kopiowania plików dzienników transakcji. Należy też skonfigurować system SQL Server na potrzeby procesu przesyłania dzienników. W celu skonfigurowania katalogów współużytkowanych wykonaj następujące kroki.

1. Najpierw utwórz katalog na kopie zapasowe, z którego korzystać będzie serwer główny. Udostępnij ten katalog i upewnij się, że także serwer pomocniczy ma do niego dostęp. Możesz np. wykorzystać katalog `c:\primaryBackupLog`, współużytkowany za pomocą nazwy UNC `\\primaryserver\primaryBackupLog`. Nazwa UNC jest stosowana przy dostępie do udziału plikowego z poziomu serwera zdalnego. Dostęp do udziału lokalnego można uzyskać za pomocą nazwy UNC lub litery katalogu. Konto narzędzia SQL Server Agent z serwera głównego musi mieć uprawnienia do odczytu i zapisu do opisywanego katalogu. Na serwerze pomocniczym konto narzędzia SQL Server Agent lub pośrednika wykonującego zadania potrzebuje uprawnień do odczytu do tego katalogu. Ponadto konto narzędzia SQL Server Agent musi mieć uprawnienia do wykonywania rozszerzonych procedur składowanych związanych z przesyłaniem dzienników.
2. Następnie utwórz na serwerze pomocniczym katalog docelowy, np. `c:\secondaryBackupDest`. Konto narzędzia SQL Server Agent lub konto pośrednika wykonującego zadania na tym serwerze musi mieć uprawnienia do odczytu i zapisu do tego katalogu. Ponadto potrzebne jest uprawnienie do wykonywania rozszerzonych procedur składowanych związanych z przesyłaniem dzienników.

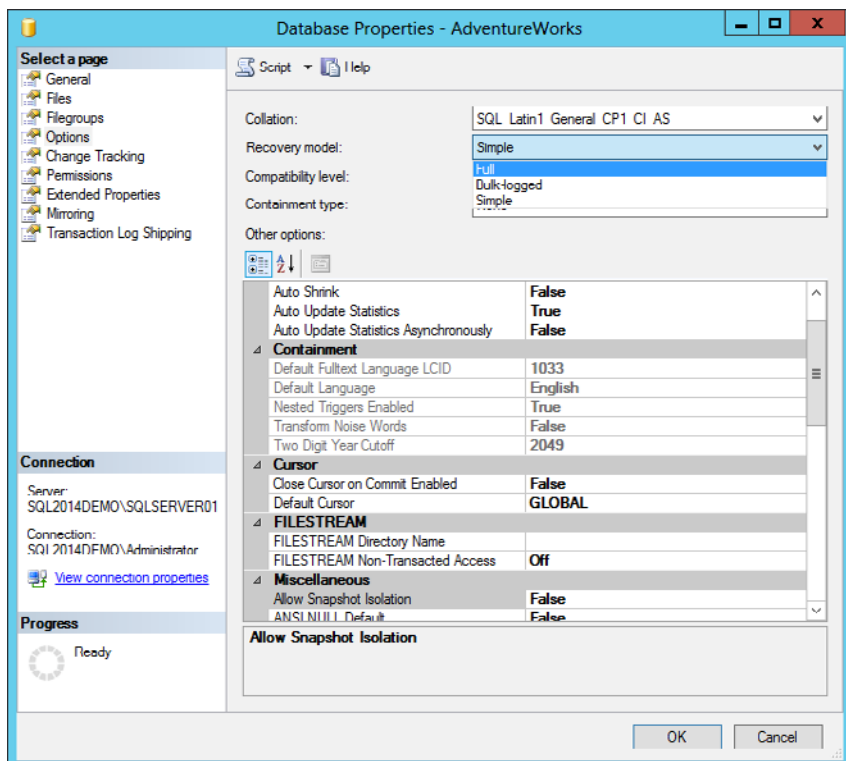
UWAGA Przy przesyłaniu dzienników używane są rozszerzone procedury składowane. Dla narzędzia SQL Server Agent na serwerach głównym i pomocniczym trzeba włączyć obsługę takich procedur (domyślnie jest ona wyłączona). W tym celu wywołaj następujący kod:

```
EXECUTE sp_configure "Agent XPs", 1;
GO

Reconfigure;
GO
```

- Model odzyskiwania dla bazy, w której używane jest przysyłanie dzienników, ustaw na FULL lub BULK_LOGGED. Możesz to zrobić na dwa sposoby — w programie SQL Server Management Studio lub za pomocą polecenia w języku T-SQL.

Aby zastosować program SQL Server Management Studio, otwórz okno *Database Properties* i kliknij zakładkę *Options*. Na liście rozwijanej *Recovery model* wybierz model odzyskiwania (patrz rysunek 18.2).



Rysunek 18.2. Ustawianie modelu odzyskiwania

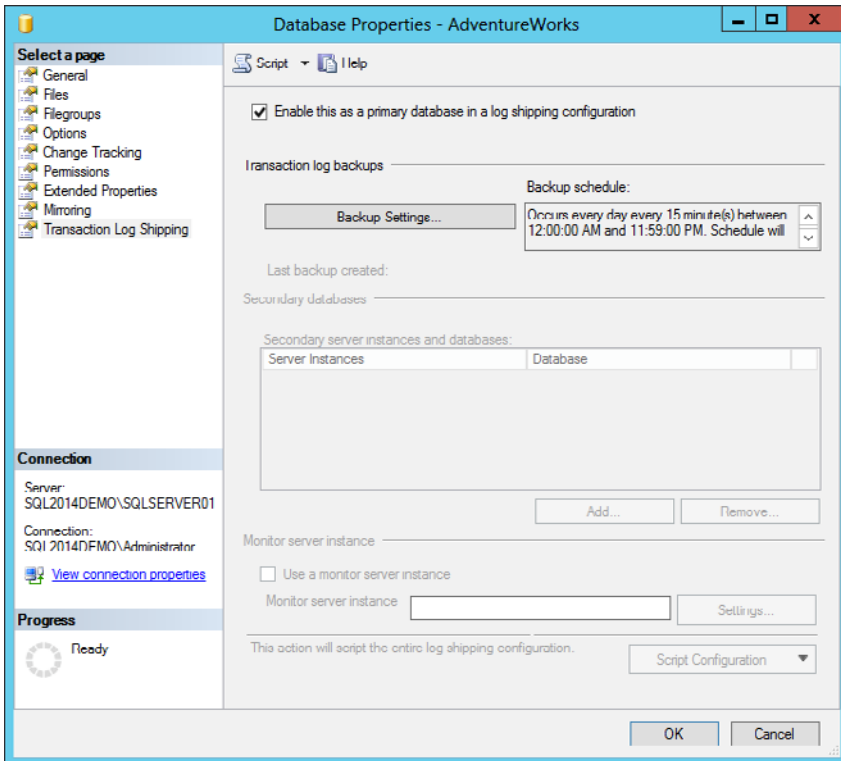
Jeśli chcesz zastosować polecenie w języku T-SQL, otwórz okno kwerend języka SQL i wpisz instrukcję ALTER DATABASE, aby zmienić model odzyskiwania. Przykładowo w celu ustawienia modelu dla bazy AdventureWorks na Full, wywołaj następujący kod w języku T-SQL:

```
USE master;
GO
ALTER DATABASE AdventureWorks
SET RECOVERY FULL;
GO
```

Wdrażanie przysyłania dzienników za pomocą programu SQL Server Management Studio

Aby włączyć przysyłanie dzienników w programie SQL Server Management Studio, wykonaj następujące kroki.

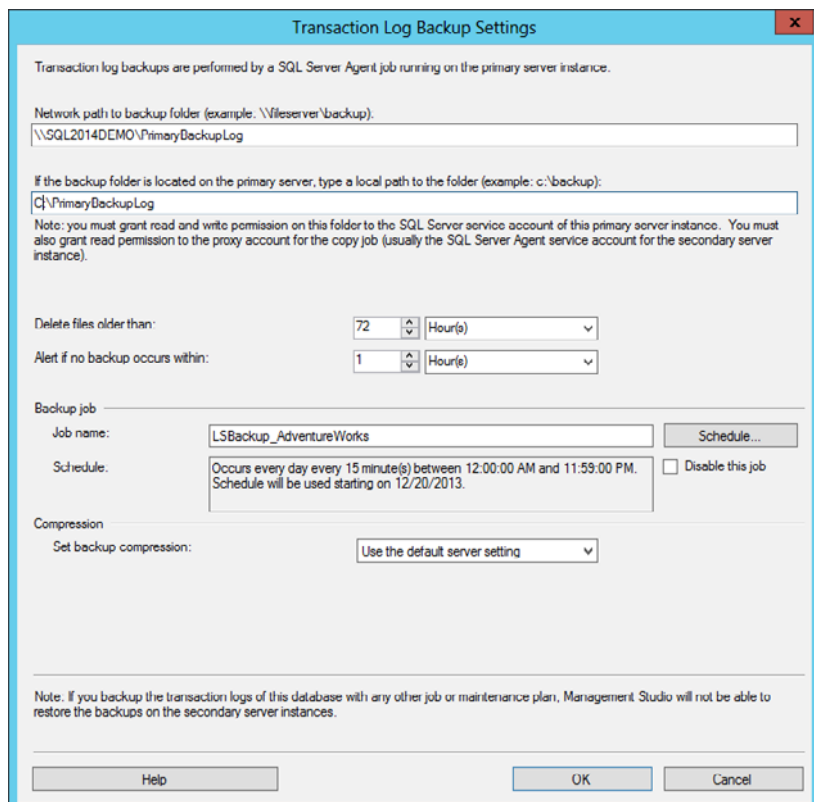
1. Najpierw wybierz skonfigurowaną bazę danych i otwórz okno z jej właściwościami. Wybierz zakładkę *Transaction Log Shipping*. Kliknij pole wyboru *Enable this as a primary database in a log shipping configuration*, co przedstawiono na rysunku 18.3.



Rysunek 18.3. Ustawianie głównej bazy danych

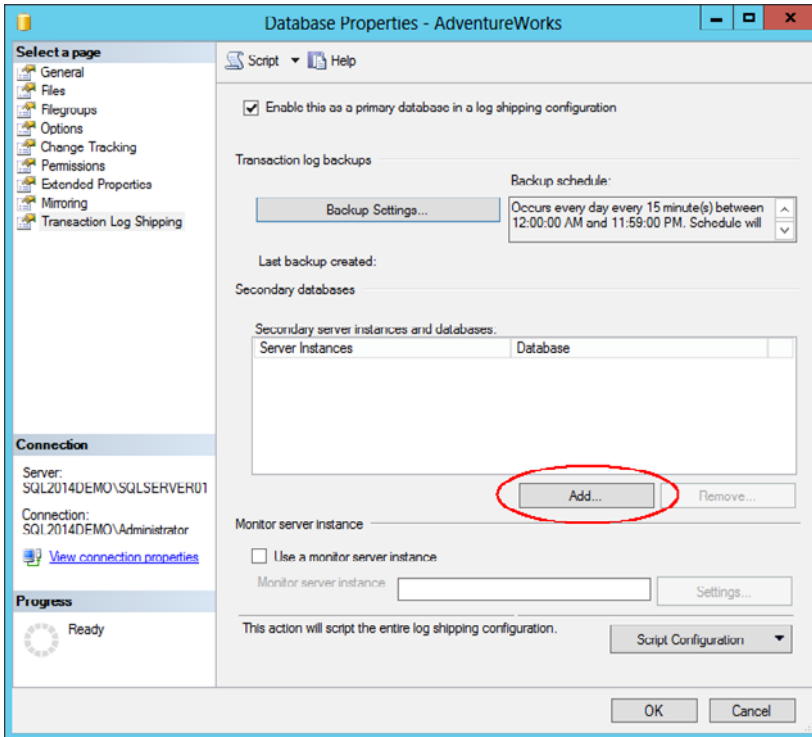
2. Następnie kliknij przycisk *Backup Settings*. Pojawi się okno dialogowe *Transaction Log Backup Settings*, przedstawione na rysunku 18.4.
3. Tu należy podać ścieżkę sieciową do katalogu z kopiami zapasowymi, a także ścieżkę lokalną, jeśli ten katalog znajduje się bezpośrednio na serwerze głównym (wtedy przy przesyłaniu dzienników używana jest ścieżka lokalna). Pamiętaj, że usługa SQL Server i konto narzędzia SQL Server Agent lub pośrednika wykonującego zadania tworzenia kopii zapasowej wymagają uprawnień do odczytu i zapisu do tego katalogu. Jeśli to możliwe, katalog ten należy umieścić w systemie dyskowym odpornym na błędy. Wtedy awaria dysku nie spowoduje utraty plików dzienników transakcji.

Zwykle można usunąć kopie zapasowe dzienników transakcji, które zostały już przywrócone i są starsze niż wartość podana w polu *Delete files older than*. Pozwala to kontrolować wielkość katalogu ze starszymi plikami kopii zapasowych. Aby jednak zapewnić dodatkowy poziom ochrony, w sytuacji gdy firma potrzebuje odzyskać stan z określonego punktu w czasie, nie usuwaj tych plików do czasu utworzenia przez system operacyjny ich kopii na innym dysku (przy czym aby przywrócić dzienniki transakcji, dostępna musi być też pełna kopia zapasowa bazy). Wartość domyślna wspomnianego ustawienia to 72 godziny.

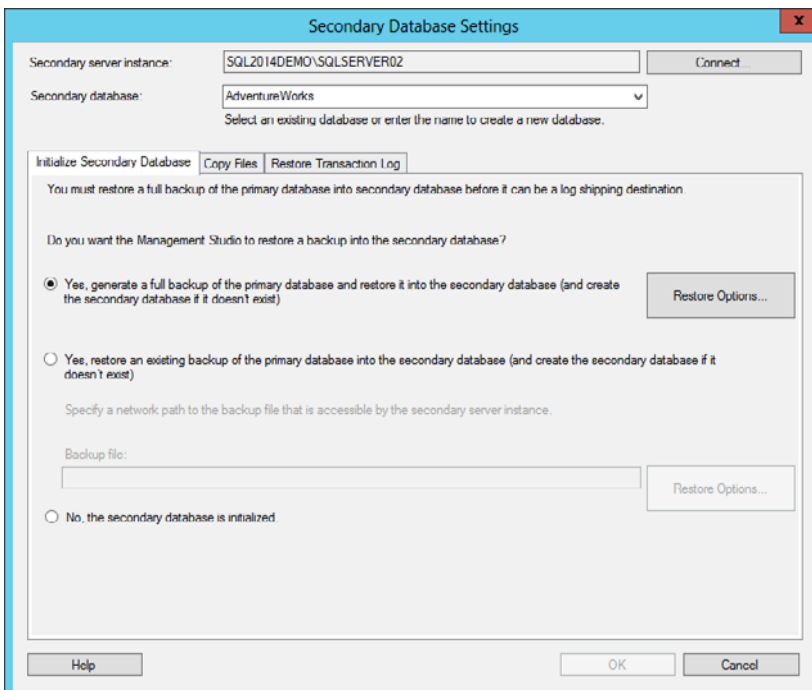


Rysunek 18.4. Ustawienia kopii zapasowych dziennika transakcji

4. W polu *Alert if no backup occurs within* ustaw wartość dostosowaną do wymagań firmy. Czas między tworzeniem kolejnych kopii zapasowych zależy od ilości danych, z których utratą firma się godzi, i od wagi tych danych. Ten czas wpływa na zgłaszanie alertów. Jeśli np. firma potrzebuje wysoce dostępnego serwera pomocniczego, a kopia zapasowa dziennika transakcji jest wykonywana co kilka minut, wspomnianą wartość należy ustawić tak, aby alert był wysyłany, gdy zadanie nie zostanie uruchomione po tych kilku minutach. Wartość domyślna to 1 godzina.
5. Kliknij przycisk *Schedule*, aby otworzyć stronę *Job Schedule Properties* i skonfigurować harmonogram wykonywania zadania tworzącego kopię zapasową dziennika transakcji. Ważnym ustawieniem jest tu *Occurs every*. Domyślnie ma ono wartość 15 minut. Aby zapewnić wyższą dostępność, można uruchamiać zadanie co minutę. Należy przy tym zadbać o to, aby poprzednie zadanie tworzenia kopii zapasowej zdążyło zakończyć pracę. Ta wartość określa, w jakim stopniu serwery główny i pomocniczy są zsynchronizowane. Po ustawieniu odpowiedniej wartości kliknij przycisk *OK* na stronie *Job Schedule Properties*. Następnie kliknij przycisk *OK* w oknie dialogowym *Log Backup Settings*, aby wrócić do zakładki *Transaction Log Shipping* w oknie *Database Properties*.
6. Kliknij przycisk *Add*, aby wyświetlić okno dialogowe *Secondary Database Settings*, w którym można skonfigurować serwer pomocniczy (rezerwowy). Okno jest pokazane na rysunku 18.5. Kliknij przycisk *Connect* i wybierz egzemplarz serwera pomocniczego. Następnie podaj nazwę istniejącej lub nowej bazy danych. W zakładce *Initialize Secondary Database*, widocznej na rysunku 18.6, dostępne są trzy opcje.



Rysunek 18.5. Kliknij przycisk Add

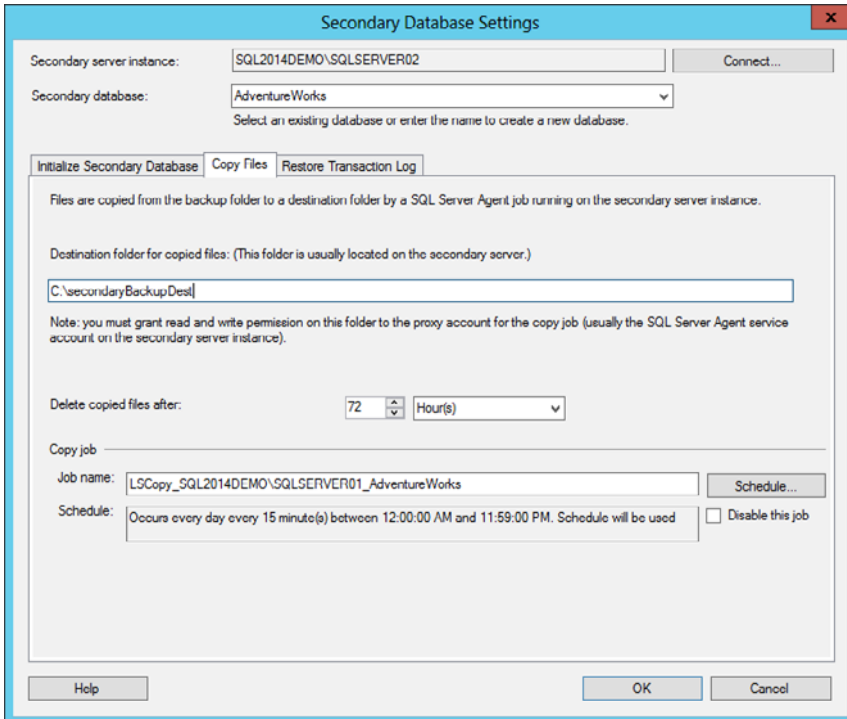


Rysunek 18.6. Inicjowanie bazy danych na serwerze pomocniczym

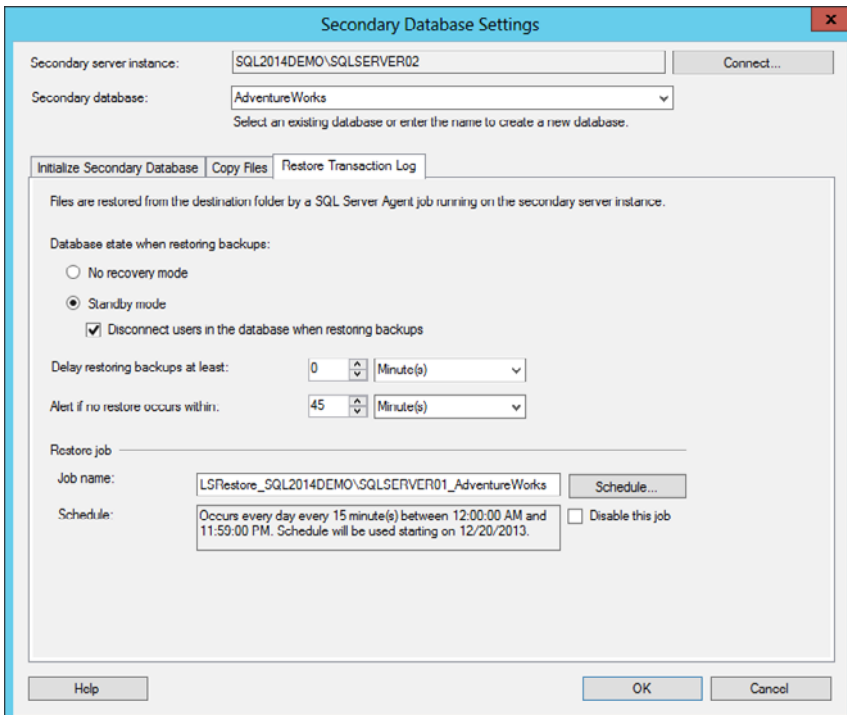
Dostępne opcje są odpowiedziami na pytanie, czy chcesz, aby program Management Studio przywracał kopie zapasowe w pomocniczej bazie danych.

- **Yes, generate a full backup of the primary database and restore it into the secondary database (and create the secondary database if it doesn't exist).** To ustawienie pozwala ustawić katalogi na pliki danych i pliki dziennika. Jeśli ich nie podasz, użyte zostaną lokalizacje domyślne.
- **Yes, restore an existing backup of the primary database into the secondary database (and create the secondary database if it doesn't exist).** Ta opcja pozwala ustawić katalogi na pliki danych i pliki dziennika. Należy też określić lokalizację sieciową z plikami kopii zapasowych, które chcesz przywrócić.
- **No, the secondary database is initialized.** Ta opcja oznacza, że baza została już utworzona. Przed rozpoczęciem odzyskiwania danych trzeba udostępnić dzienniki transakcji, aby cały mechanizm mógł zadziałać. W tym podejściu numery LSN dzienników z serwerów głównego i pomocniczego muszą do siebie pasować. Ponadto pomocnicza baza musi działać w trybie NORECOVERY lub STANDBY, aby możliwe było przywrócenie dodatkowych plików dzienników transakcji.

7. Następnie przywróć pomocniczą bazę danych na podstawie kopii z serwera głównego. Ścieżka do pliku i nazwa nowej bazy danych będą takie same jak dla bazy głównej. Nie da się zmienić nazwy pliku, można natomiast zmodyfikować ścieżkę (w tym celu kliknij przycisk *Restore Options*). W przykładach z tego podrozdziału bazy główna i pomocnicza znajdują się na tym samym serwerze z systemem Windows, ale pracują w innych egzemplarzach systemu SQL Server 2014. Pomocnicza baza AdventureWorks została utworzona za pomocą tych samych nazw, co główna baza AdventureWorks, ale znajduje się w innym katalogu.
8. W zakładce *Copy Files*, przedstawionej na rysunku 18.7, podaj w polu tekstowym *Destination folder for copied files* docelowy katalog (np. *c:\secondaryBackupDest*). Opcja *Delete copied files after* pozwala kontrolować wielkość katalogu po przywróceniu dziennika transakcji w bazie z serwera pomocniczego. Usuwane są wszystkie pliki starsze niż podana wartość. Domyślnie ta wartość to 72 godziny.
9. Kliknij przycisk *Schedule*, aby skonfigurować harmonogram wykonywania zadania kopiującego pliki dziennika transakcji. Ważnym ustawieniem jest *Occurs every*; jego domyślna wartość to 15 minut. Możesz skrócić ten czas, aby serwer pomocniczy był szybciej synchronizowany z serwerem głównym. Po zmianie ustawień kliknij przycisk *OK*, aby wrócić do strony *Secondary Database Settings*.
10. Otwórz zakładkę *Restore Transaction Log* (patrz rysunek 18.8).
Przy polu *Database state when restoring backups* dostępne są dwie opcje.
 - **No recovery mode.** Baza pomocnicza jest ustawiana na tryb NORECOVERY, co umożliwia serwerowi przywrócenie dodatkowych kopii zapasowych dzienników transakcji, ale nie daje użytkownikom dostępu do bazy.
 - **Standby mode.** Baza pomocnicza umożliwia wykonywanie na bazie operacji w trybie tylko do odczytu (takich jak generowanie raportów). Jednak, co opisano już wcześniej, proces przywracania wymaga dostępu do pomocniczej bazy w trybie na wyłączność. Jeśli użytkownicy będą korzystać z bazy, proces przywracania nie zadziała.



Rysunek 18.7. Zarządzanie kopiowanymi plikami



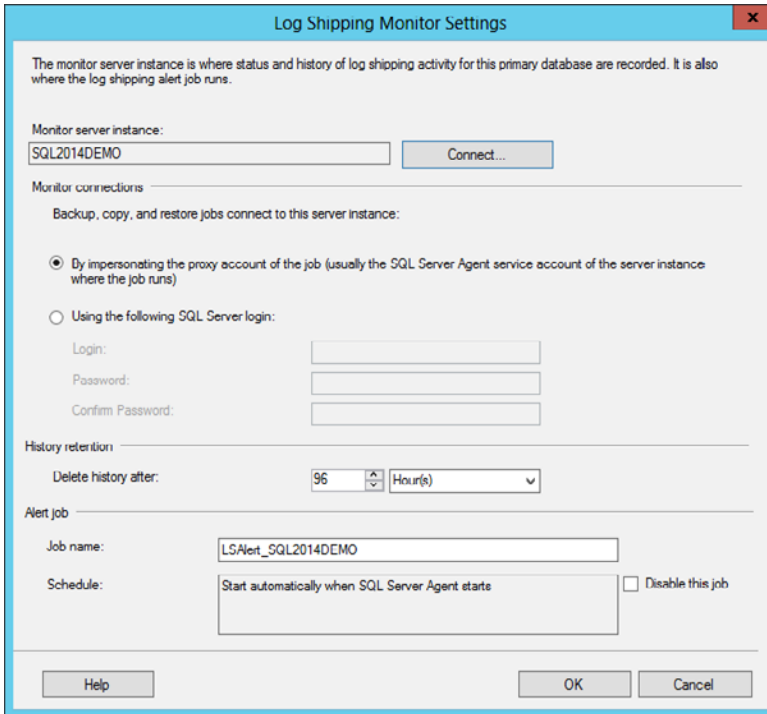
Rysunek 18.8. Konfigurowanie przywracania dzienników transakcji

Domyślną wartością ustawienia *Delay restoring backups at least* jest 0 minut. To ustawienie zwykle zmienia się wtedy, gdy firma chce opóźnić synchronizację pomocniczej bazy danych na wypadek, gdyby w głównej bazie dane zostały uszkodzone lub przypadkowo usunięte. Takie opóźnienie zapobiega przywróceniu dziennika transakcji z niepożądanymi operacjami.

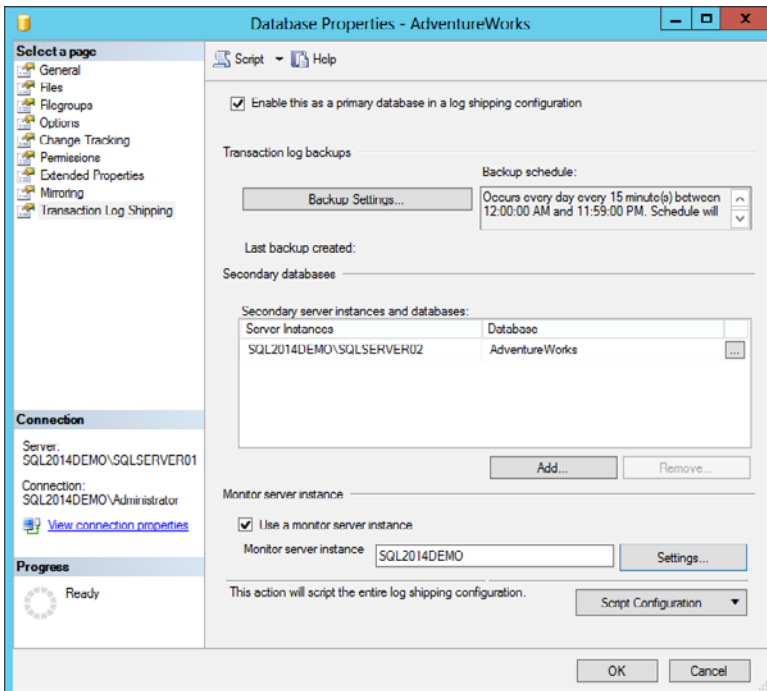
Ustawienie *Alert if no restore occurs within* domyślnie ma wartość 45 minut. Należy je dostosować do wymagań firmy.

UWAGA Alert może świadczyć o wystąpieniu w pomocniczej bazie poważnego błędu, który uniemożliwia przywrócenie dodatkowych dzienników transakcji. W takiej sytuacji przejrzyj historię pracy zadania przywracającego dzienniki. Domyślna nazwa tego zadania to `LSRestore_Nazwa_serwera_Nazwa_głównej_bazy`. Jest to jedno z zadań narzędzia SQL Server Agent z serwera pomocniczego. Ponadto poszukaj dodatkowych informacji w przeglądarce zdarzeń systemu Windows. Możesz też skopiować i wkleić polecenia z tego zadania do okna poleceń języka SQL, aby uzyskać dodatkowe informacje o błędach pomocne przy diagnozowaniu problemu.

11. Po ustawieniu opcji kliknij przycisk *OK* na stronie *Secondary Database Settings*. Aby dodać kolejny serwer pomocniczy, kliknij przycisk *Add* i ponownie wykonaj opisane kroki.
12. Jeśli chcesz dodać serwer monitorujący, w zakładce *Transaction Log Shipping* w oknie właściwości głównej bazy danych kliknij przycisk *Use a Monitor Server Instance*. Następnie kliknij przycisk *Settings*. Zalecamy utworzenie serwera monitorującego na odrębnym komputerze, tak aby awaria serwera głównego lub pomocniczego nie spowodowała wyłączenia monitorowania.
13. Na stronie *Log Shipping Monitor Settings*, widocznej na rysunku 18.9, kliknij przycisk *Connect*, a następnie wybierz *Monitor server instance*, aby dodać serwer monitorujący w danym środowisku z przesyłaniem dzienników.
14. Używane konto musi mieć uprawnienia administratora systemu do serwera pomocniczego. W polu *By impersonating the proxy account of the job* lub *Using the following SQL Server login* możesz określić, jak zadania tworzące kopie zapasowe, kopiujące pliki i przywracające dane mają łączyć się z systemem przy aktualizowaniu informacji o zadaniach w bazie msdb. Jeśli stosujesz zabezpieczenia zintegrowane, zadania powinny przy nawiązywaniu połączenia podawać się za konto pośrednika narzędzia SQL Server Agent uruchamiającego zadania. Można też zastosować login z systemu SQL Server.
Pole *Delete history after* określa, ile danych historycznych jest przechowywanych w bazie msdb. Domyślnie zachowywane są dane z 96 godzin. Czas przechowywania informacji zależy od wymagań firmy i ilości miejsca na dysku. W większości sytuacji wartość domyślna jest odpowiednia, chyba że planujesz w przyszłości przeprowadzać analizy danych — wtedy należy wprowadzić zmiany.
15. Po wykonaniu wszystkich kroków kompletna konfiguracja przesyłania dzienników powinna wyglądać tak, jak na rysunku 18.10. Kliknij przycisk *OK* w oknie *Database Properties*, aby zakończyć konfigurowanie przesyłania dzienników.



Rysunek 18.9. Ustawianie serwera monitorującego



Rysunek 18.10. Skonfigurowany mechanizm przesyłania dzienników

UWAGA Kompletny skrypt ustawiający przesyłanie dzienników w przedstawiony tu sposób znajdziesz w pliku *R18_PrzesylanieDziennikow.sql* dostępnym do pobrania w witrynie wydawnictwa Helion.

Wdrażanie przesyłania dzienników za pomocą instrukcji języka T-SQL

W celu skonfigurowania przesyłania dzienników można też wykorzystać polecenia języka T-SQL. Jeśli konfigurujesz ten mechanizm za pomocą programu SQL Server Management Studio, możesz zachować wygenerowany skrypt z instrukcjami, co pozwala szybko ponownie skonfigurować serwer w czasie odzyskiwania systemu po katastrofie i uniknąć przy tym błędów ludzkich. Poniższe polecenia języka T-SQL odpowiadają krokom wykonanym wcześniej w programie SQL Server Management Studio.

Na serwerze głównym wywołaj następujące procedury składowane.

- `master.dbo.sp_add_log_shipping_primary_database`. Powoduje skonfigurowanie głównej bazy danych pod kątem przesyłania dzienników i ustawia zadanie tworzenia kopii zapasowych dzienników transakcji.
- `msdb.dbo.sp_add_schedule`. Tworzy harmonogram na potrzeby przesyłania dzienników.
- `msdb.dbo.sp_attach_schedule`. Dodaje zadanie przesyłania dzienników do harmonogramu.
- `msdb.dbo.sp_update_job`. Aktywuje zadanie tworzenia kopii zapasowych dzienników transakcji.
- `master.dbo.sp_add_log_shipping_alert_job`. Tworzy zadanie zgłaszające alerty i dodaje jego identyfikator do tabeli `log_shipping_monitor_alert`. Ta procedura składowana umożliwia powiadamianie o alertach.

Na serwerze pomocniczym wywołaj następujące procedury składowane.

- `master.dbo.sp_add_log_shipping_secondary_database`. Konfiguruje podstawowe informacje, dodaje łącza służące do lokalnego i zdalnego monitorowania oraz tworzy zadania kopiujące i przywracające dane na serwerze pomocniczym dla podanej głównej bazy danych.
- `msdb.dbo.sp_add_schedule`. Tworzy harmonogram dla zadania kopiowania dzienników.
- `msdb.dbo.sp_attach_schedule`. Dodaje zadanie kopiowania dzienników do harmonogramu.
- `msdb.dbo.sp_add_schedule`. Tworzy harmonogram dla zadania przywracania dzienników.
- `msdb.dbo.sp_attach_schedule`. Dodaje zadanie przywracania dzienników do harmonogramu.
- `master.dbo.sp_add_log_shipping_secondary_database`. Konfiguruje pomocniczą bazę danych pod kątem przesyłania dzienników.
- `msdb.dbo.sp_update_job`. Aktywuje zadanie kopiowania dzienników.
- `msdb.dbo.sp_update_job`. Aktywuje zadanie przywracania dzienników.

Następnie wróć do serwera głównego i wykonaj następującą procedurę składowaną.

- `master.dbo.sp_add_log_shipping_primary_secondary`. Dodaje na serwerze głównym wpis dotyczący serwera pomocniczego.

Monitorowanie systemu i rozwiązywanie problemów

Dla przesyłania dzienników opracowane zostały mechanizmy monitorowania, pozwalające śledzić postęp zadań tworzących kopie zapasowe, kopiujących dzienniki i przywracających je. Ponadto monitorowanie pomaga stwierdzić, czy te zadania działają w zsynchronizowany sposób na serwerze pomocniczym. Oznaką problemów jest np. brak postępów lub niepowodzenie zadania (obie te sytuacje omawiamy dalej w tym podrozdziale).

Istnieją dwa sposoby monitorowania postępów w przesyłaniu dzienników — raport Transaction Log Shipping Status w programie SQL Server Management Studio i procedura składowana `master.dbo.sp_help_log_shipping_monitor`. Każda z tych metod pomaga ustalić, czy serwer pomocniczy jest zsynchronizowany z serwerem głównym, a także jakie jest przesunięcie w czasie między nimi.

W programie SQL Server Management Studio można obserwować operacje w formie graficznej. Zastosowanie procedury składowanej umożliwia ustawienie wielokrotnie wykonywanego zadania narzędzia SQL Server Agent, które monitoruje przesyłanie dzienników i wysyła alerty. Każde z tych podejść pozwala stwierdzić, że zadanie nie robi postępów, a także ustalić ostatnią kopię zapasową dziennika transakcji oraz ostatni plik skopiowany i przywrócony na serwerze pomocniczym.

Aby zwracać informacje o wykrytych błędach, przy przesyłaniu dzienników uruchamiane są zadania zgłaszające alerty. Zadania sprawdzają, czy nie zostały przekroczone ustawione limity czasu. Używana jest przy tym procedura składowana `sys.sp_check_log_shipping_monitor_alert`. Po przekroczeniu limitu czasu procedura zgłasza alert ustawiany jako stan w procesie monitorowania przesyłania dzienników. Możesz zmodyfikować te zadania, aby przechwytywać alerty i wysyłać powiadomienia o nich za pomocą narzędzia SQL Server Agent.

Jeśli używany jest serwer monitorujący, przechowuje on alerty i informuje o tworzeniu kopii zapasowych, kopiowaniu plików i przywracaniu dzienników transakcji. Gdy taki serwer jest niedostępny, serwer główny zarządza zadaniami zgłaszającymi alerty o tworzeniu kopii zapasowych, a serwer pomocniczy odpowiada za zadania zgłaszające alerty o kopiowaniu plików i przywracaniu dzienników transakcji. Jeżeli serwer monitorujący jest dostępny, serwery główny i pomocniczy nie uruchamiają zadań związanych z alertami.

Poniżej pokazany jest przykładowy błąd zgłaszany, gdy proces tworzenia kopii zapasowej trwa dłużej niż ustawiony limit 30 minut:

```
Executed as user: NT AUTHORITY\SYSTEM. The log shipping primary database
SQLServer1.AdventureWorks has backup threshold of 30 minutes and has not
performed a backup log operation for 60 minutes. Check agent log and log shipping
monitor information. [SQLSTATE 42000](Error 14420). This step failed.
```

Następny przykładowy błąd jest zgłaszany, gdy proces przywracania dziennika transakcji przekroczył limit 30 minut:

```
Executed as user: NT AUTHORITY\SYSTEM. The log shipping secondary database
SQLServer2.AdventureWorks has restore threshold of 30 minutes and is out of
sync. No restore was performed for 60 minutes. Restored latency is 15 minutes.
Check agent log and log shipping monitor information. [SQLSTATE 42000]
(Error 14421).
The step failed.
```

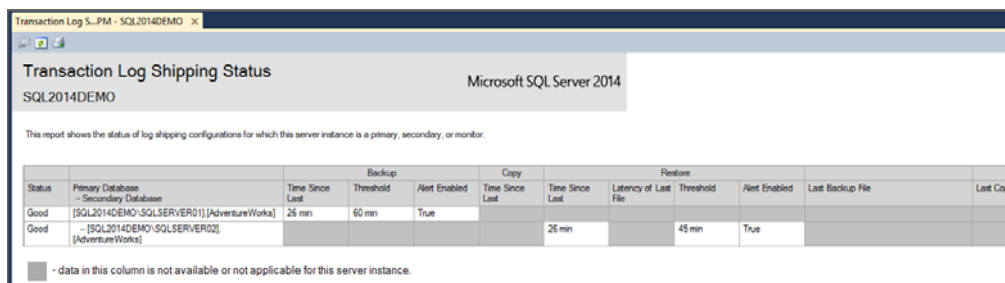
Można też ustawić alerty dla błędów 14420 i 14421, aby po ich wystąpieniu narzędzie SQL Server Agent przesyłało powiadomienia do operatorów.

Monitorowanie z wykorzystaniem programu SQL Server Management Studio

Raport Transaction Log Shipping Status wyświetla informacje z mechanizmu monitorowania w programie SQL Server Management Studio. Ten raport uruchamia procedurę składowaną `sp_help_log_shipping_monitor`. Gdy działa ona na serwerze głównym, informuje o szczegółach tworzenia kopii zapasowej dzienników transakcji. Jeśli jest wykonywana na serwerze pomocniczym, wyświetla informacje na temat kopiowania i przywracania dzienników. Gdy używany jest serwer monitorujący, generowany jest na nim raport zbiorczy z informacjami o tworzeniu kopii zapasowych, kopiowaniu plików i przywracaniu dzienników. Aby wyświetlić raport Transaction Log Shipping Status, wykonaj następujące kroki.

1. Nawiąż połączenie z serwerem głównym, pomocniczym lub monitorującym w programie SQL Server Management Studio. Najlepiej użyj serwera monitorującego, ponieważ może on wygenerować raport zbiorczy na temat przesyłania dzienników.
2. Jeśli okno *Object Explorer* jest niewidoczne, wybierz opcję *View/Object Explorer*.
3. Kliknij prawym przyciskiem myszy węzeł serwera w oknie *Object Explorer* i wybierz opcję *Reports/Standard Reports/Transaction Log Shipping Status*.

Na rysunku 18.11 przedstawiono przykładowy raport Transaction Log Shipping Status wygenerowany na serwerze monitorującym. Widoczne są tu informacje na temat wszystkich operacji z procesu przesyłania dzienników wraz z alertami.



Status	Primary Database - Secondary Database	Backup			Copy		Restore			Last Backup File	Last Copy
		Time Since Last	Threshold	Alert Enabled	Time Since Last	Time Since Last	Latency of Last File	Threshold	Alert Enabled		
Good	[SQL2014DEMO-SQLSERVER01].[AdventureWorks]	26 min	60 min	True							
Good	[SQL2014DEMO-SQLSERVER02].[AdventureWorks]					26 min		45 min	True		

■ data in this column is not available or not applicable for this server instance.

Rysunek 18.11. Raport Transaction Log Shipping Status

Monitorowanie z wykorzystaniem procedur składowanych

Jeśli wywołasz procedurę składowaną `sp_help_log_shipping_monitor` z bazy master w oknie kwerend języka SQL, uzyskasz informacje o stanie przesyłania dzienników (są to dane podobne do danych w raporcie Transaction Log Shipping Status). Uruchomienie tej procedury na serwerze głównym spowoduje wyświetlenie informacji na temat zadania tworzącego kopie zapasowe dzienników transakcji. Na serwerze pomocniczym zwracane są dane o zadaniach kopiujących i przywracających takie dzienniki. Na serwerze monitorującym procedura wyświetla zbiorcze informacje na temat tworzenia kopii zapasowych, kopiowania plików i przywracania dzienników transakcji. Dzieje się tak, ponieważ serwer monitorujący ma dostęp do wszystkich procesów związanych z przesyłaniem dzienników.

Dodatkowe informacje o przebiegu przesyłania dzienników i przydatne przy rozwiązywaniu problemów można pobrać z odpowiednich tabel za pomocą procedur składowanych z bazy msdb. Więcej informacji na ten temat zawiera dokumentacja Books Online poświęcona systemowi SQL Server 2014.

Rozwiązywanie problemów

Jak wcześniej wspomniano, przesyłanie dzienników obejmuje trzy podstawowe operacje: tworzenie kopii zapasowej dziennika transakcji, kopiowanie plików i przywracanie dziennika transakcji. Rozwiązywanie problemów wymaga ustalenia, która z tych operacji się nie powiodła. Możesz wykorzystać do tego mechanizmy monitorowania przesyłania dzienników.

Załóżmy, że zgodnie z informacjami o przywracaniu operacja ta w ostatnich 60 minutach nie została wykonana. Należy na serwerze pomocniczym sprawdzić historię zadania za pomocą narzędzia SQL Server Agent i przeglądarki zdarzeń systemu Windows, aby ustalić komunikat o błędzie. Jeśli nie powiodło się zadanie kopiowania plików, przyczyną może być awaria sieci. Jeżeli zawiodło zadanie przywracania dziennika transakcji, powodem może być niedostępny serwer lub korzystanie z bazy przez użytkowników (gdy pracuje ona w trybie STANDBY).

OSTRZEŻENIE Pamiętaj, że zmiana modelu odzyskiwania na Simple uniemożliwia przesyłanie dzienników, ponieważ dziennik transakcji jest wtedy przycinany. W takiej sytuacji należy zmienić konfigurację przesyłania dzienników. Jeśli zapisałeś skrypty konfiguracyjne przesyłania dzienników, rekonfiguracja nie powinna sprawiać trudności. Ponadto przy przesyłaniu dzienników nie należy tworzyć poza tym mechanizmem kopii zapasowych dzienników transakcji, ponieważ łańcuch dzienników na serwerach głównym i pomocniczym będą się wtedy różnić między sobą. Gdy ponadto pliki dzienników transakcji są duże, ich kopiowanie i przywracanie trwa dłużej, przez co opóźnienie serwera pomocniczego względem głównego jest większe.

Zarządzanie zmianą ról

Aby zachować ciągłość pracy firmy, rozwiązanie zapewniające wysoką dostępność musi umożliwiać płynną zmianę ról serwerów głównego i pomocniczego. Przy przesyłaniu dzienników na serwerze pomocniczym dostępne muszą być pewne zależności, ponieważ ta operacja odbywa się na poziomie baz danych. Obiekty spoza baz uczestniczących w przesyłaniu dzienników nie są uwzględniane w tym procesie. Przykładowo loginy systemu SQL Server znajdują się w bazie master, a zadania systemu SQL Server — w bazie msdb. Dlatego tymi i innymi komponentami trzeba zarządzać za pomocą innych procedur, aby umożliwić użytkownikom nawiązanie połączenia z serwerem pomocniczym, gdy przejmie on rolę serwera głównego. Ponadto musisz opracować sposób przekierowywania aplikacji klienckich na nowy serwer główny. Przykłady opisujemy w punkcie „Przekierowywanie klientów na serwer pomocniczy”, dalej w tym rozdziale.

Synchronizowanie zależności

Przy przesyłaniu dzienników wprowadzane są zmiany z bazy obsługiwanej za pomocą tego mechanizmu, natomiast zewnętrzne zależności nie są uwzględniane. Ponadto przesyłanie dzienników nie można stosować do transferu systemowych baz danych. Przesyłanie dzienników nie synchronizuje nowych użytkowników baz danych, a także nowych loginów, zadań i zależności z zewnętrznymi bazami danych.

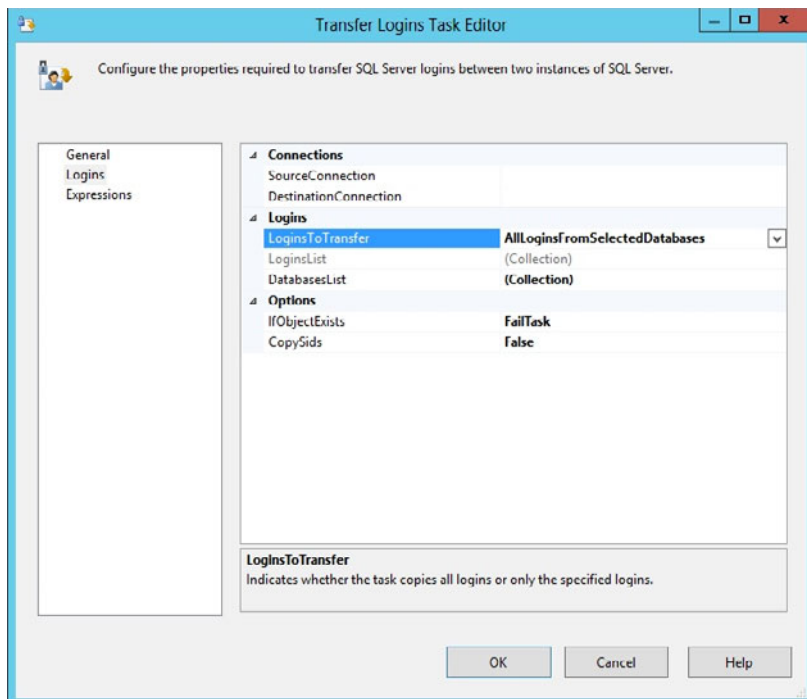
W systemie SQL Server 2014 do obsługi zależności związanych z loginem służy mechanizm tworzenia niezależnych baz danych (ang. *contained database*). Pozwala on zapisać dane uwierzytelniające użytkownika w bazie danych i przeprowadzać uwierzytelnianie bezpośrednio w bazie danych, co znosi zależność od loginu. W nowych systemach warto rozważyć wykorzystanie tej właściwości i uniknięcie zależności od loginów z serwera.

Gdy niezależne bazy danych nie są stosowane, po przełączeniu awaryjnym w bazie z pomocniczego serwera nie ma loginów użytkowników próbujących się zalogować. Ponadto niedostępne są zadania skonfigurowane na serwerze głównym. Jeśli serwer główny używa serwerów połączonych do uzyskiwania dostępu do zdalnego systemu SQL Server, praca bazy danych będzie niemożliwa, ponieważ nie będzie można znaleźć potrzebnych serwerów połączonych. Dlatego należy zidentyfikować zależności baz, dla których używane jest przysyłanie dzienników, i opracować plan udostępniania niezbędnych zasobów po przełączeniu awaryjnym. W dalszych podpunktach opisujemy standardowe zależności związane z przysyłaniem dzienników i sposoby ich obsługi.

Loginy i użytkownicy baz danych

W trakcie przygotowywania procesu synchronizowania loginów systemu SQL Server z serwerem pomocniczym i bazą danych powinieneś zacząć od utworzenia zadania narzędzia SQL Server Agent, które będzie regularnie uruchamiane. Przy planowanym przełączeniu awaryjnym te zadania należy wykonać przed przełączeniem, aby zaktualizować serwer pomocniczy za pomocą najświeższych informacji. Wykonaj następujące kroki.

1. Przygotuj pakiet SSIS do przenoszenia loginów. Użyj pakietu SQL Server Data Tools 2014 i utwórz nowy projekt typu *Integration Services*.
2. W oknie *Solution Explorer* nazwij projekt SSIS *Transfer Logins* i zmień nazwę pakietu SSIS na *Transfer Logins*.
3. Kliknij okno narzędzi i przeciągnij do pakietu zadanie *Transfer Logins Task*.
4. Kliknij zadanie *Transfer Logins Task* prawym przyciskiem myszy i wybierz opcję *Edit*.
5. Kliknij opcję *Logins*. Pojawi się okno dialogowe widoczne na rysunku 18.12.



Rysunek 18.12. Przenoszenie loginów

6. Dla właściwości *SourceConnection* ustaw nowe połączenie z serwerem głównym.
7. Dla właściwości *DestinationConnection* ustaw nowe połączenie z serwerem pomocniczym.
8. Dla właściwości *LoginsToTransfer* ustaw opcję *AllLoginsFromSelectedDatabases*.
9. Dla właściwości *DatabaseList* ustaw bazę danych, dla której stosowane jest przesyłanie dzienników.
10. W sekcji *Options* we właściwości *IfObjectExists* określ, co zrobić, jeśli dany login istnieje. Dostępne opcje to: *FailTask* (niepowodzenie zadania), *Override* (zastępowanie) i *Skip* (pomijanie). Jeżeli na serwerze pomocniczym działają inne bazy danych, mogą pojawić się zduplikowane loginy.
11. Zapisz pakiet i wybierz opcję *Build/Build SSIS Transfer Logins*, aby go skompilować.
12. W programie SQL Server Management Studio nawiąż połączenie z usługami Integration Services z serwera głównego.
13. W węźle *Stored Packages* wybierz węzeł *MSDB*. Kliknij go prawym przyciskiem myszy i wybierz opcję *Import Package*.

Następne zadanie ustawia loginy na serwerze pomocniczym po odzyskaniu bazy pomocniczej. Konto usługi SQL Server Agent lub konto pośrednika musi mieć rolę administratora systemu, aby możliwe było uruchomienie tego zadania. Oto kroki potrzebne do zsynchronizowania loginów z serwerów głównego i pomocniczego.

1. Utwórz nowe zadanie agenta SQL Server Agent na serwerze głównym i nazwij je *Sync Secondary Server Access Information*.
2. Utwórz krok o nazwie *BCP Syslogins* o następujących cechach.
 - **Type** — *Operating system*.
 - **Run As** — *SQL Server Agent Service Account* (to konto musi mieć uprawnienia administratora systemu, aby wykonać poniższe polecenie; ponadto kontu trzeba przyznać uprawnienia do odczytu i zapisu katalogu, z którego kopiowane są dane).
 - **Command** — `BCP Master.sys.syslogins out c:\login1\syslogins.dat /N /S <Nazwa_serwera> -T.`
3. Utwórz krok o nazwie *Copy Syslogins* o następujących cechach.
 - **Type** — *Operating system*.
 - **Run As** — *SQL Server Agent Service Account* (to konto musi mieć uprawnienia do odczytu i zapisu katalogu docelowego).
 - **Command** — `COPY c:\login1\syslogins.dat \\SecondaryServer\login2.`
4. Utwórz krok o nazwie *Transfer Logins* o następujących cechach.
 - **Type** — *Operating system*.
 - **Run As** — *SQL Server Agent Service Account* (to konto musi mieć uprawnienia administratora systemu, aby możliwe było wykonanie poniższego polecenia).
 - **Command** — `DTEXEC /sq Transfer Logins /Ser <Nazwa_serwera>.`
5. Utwórz krok o nazwie *Resolve Logins* o następujących cechach.
 - **Type** — *Transact-SQL script (T-SQL)*.
 - **Command** — `EXEC sp_resolve_logins @dest_db = '<Nazwa_bazy>', @dest_path = 'c:\login2\', @filename = 'syslogins.dat'.`

UWAGA Gdy synchronizujesz loginy w sposób opisany w tym punkcie, powinieneś regularnie wykonywać opisane zadanie narzędzia SQL Server Agent, aby uniknąć niespójnych loginów w momencie awarii serwera głównego.

Zadania narzędzia SQL Server Agent

Do synchronizowania zadań między serwerami głównym i pomocniczym możesz wykorzystać zadanie *Transfer Jobs Task* usługi Integration Services.

1. Utwórz pakiet za pomocą narzędzia SQL Server 2014 Data Tools i wybierz zadanie *Transfer Jobs Task*.
2. Podaj informacje o połączeniu i zadania, które należy przesłać.
3. Skompiluj i zaimportuj pakiet do systemu SQL Server lub jako pakiet SSIS z systemu plików.
4. W ostatnim kroku zaplanuj wykonywanie pakietu za pomocą zadania narzędzia SQL Server Agent, co pozwoli okresowo synchronizować zmiany z serwerem pomocniczym. Częstotliwość uruchamiania pakietu zależy od tego, jak często zmieniają się zadania, które wymagają synchronizacji.

Inne zależności baz danych

Zrób listę wszystkich zależności bazy, dla której stosowane jest przysyłanie dzienników. Oto przykładowe zależności: serwery połączone, zaszyfrowane dane, certyfikaty, zdefiniowane przez użytkownika komunikaty o błędach, powiadomienia o zdarzeniach, zdarzenia WMI, rozszerzone procedury składowane, konfiguracja serwera, silnik pełnotekstowy, uprawnienia do obiektów, ustawienia replikacji, aplikacje narzędzia Service Broker, procedury uruchomieniowe, wyzwalacze, procedury środowiska CLR i ustawienia usługi Database Mail. Przygotuj plan synchronizowania tych zależności z serwerem pomocniczym.

Przełączanie ról między serwerami głównym i pomocniczym

Gdy na serwerze głównym następuje awaria, serwer pomocniczy przejmuje jego funkcje. Jest to *przełączenie ról*. Istnieją dwa typy przełączania ról — planowane przełączanie awaryjne i nieplanowane przełączanie awaryjne. *Planowane przełączanie awaryjne* występuje zazwyczaj wtedy, gdy administrator chce wykonać prace konserwacyjne. Planuje się je zwykle na okres ograniczonej aktywności firmy lub w czasie przeznaczonym na konserwację. Administrator może wtedy np. zainstalować poprawki SP na serwerze głównym. *Nieplanowane przełączanie awaryjne* ma miejsce wtedy, gdy serwer główny staje się niedostępny i administrator przełącza role w celu zachowania ciągłości pracy firmy.

Planowane przełączanie awaryjne

Przy planowanym przełączaniu ustal czas, gdy serwer główny nie pracuje lub jest obciążony w niewielkim stopniu. Możliwe, że serwer pomocniczy nie przywrócił jeszcze wszystkich dzienników transakcji z serwera głównego. Te dzienniki mogą być dopiero kopiowane przez zadanie narzędzia SQL Server Agent. Ponadto aktywny dziennik transakcji może zawierać rekordy, których nie umieszczono jeszcze w kopii zapasowej.

Zanim zaczniesz planowane przełączanie awaryjne, musisz w pełni zsynchronizować serwer pomocniczy i przywrócić na nim aktywny dziennik transakcji. W tym celu wykonaj następujące kroki.

1. Zatrzymaj i wyłącz zadanie tworzące kopie zapasowe dzienników transakcji na serwerze głównym.
2. Wywołaj zadania kopiujące i przywracające dzienniki transakcji, aby przywrócić pozostałe dzienniki. Przy użyciu narzędzi do monitorowania przesyłania dzienników lub do generowania raportów na ten temat sprawdź, czy cały zestaw kopii zapasowych dzienników transakcji został skopiowany na serwer pomocniczy i przywrócony na nim. Można też ręcznie skopiować wszystkie zaległe kopie zapasowe dzienników transakcji z serwera głównego do katalogu na serwerze pomocniczym. Następnie należy po kolei przywrócić każdy dziennik transakcji na serwerze pomocniczym.
3. Zatrzymaj i wyłącz zadania kopiujące i przywracające dzienniki na serwerze pomocniczym.
4. Wywołaj zadanie *Sync Secondary Server Access Information*, aby zsynchronizować zależności bazy danych. Następnie wyłącz to zadanie.
5. Utwórz kopię zapasową aktywnego dziennika transakcji i przenieś ją z serwera głównego na pomocniczy. Użyj opcji *NORECOVERY*:

```
USE MASTER;
BACKUP LOG <Nazwa_bazy_danych> TO DISK =
'C:\primaryBackupLog\<Nazwa_bazy_danych>.trn'
WITH NORECOVERY;
```

 Pozwala to zrealizować dwa cele.
 - Utworzyć kopię zapasową aktywnego dziennika transakcji z serwera głównego, co pozwala przywrócić ten dziennik na serwerze pomocniczym i zsynchronizować pomocniczą bazę danych.
 - Ustawić bazę z dawnego serwera głównego na tryb *NORECOVERY*, aby umożliwić zastosowanie dzienników transakcji z nowego serwera głównego bez inicjowania bazy za pomocą przywracania. Jest to możliwe, ponieważ w tym trybie łańcuch dzienników nie jest naruszany.
6. Skopiuj kopię zapasową dziennika na serwer pomocniczy. Na tym serwerze przywróć końcówkę dziennika transakcji, a następnie odzyskaj bazę danych:

```
RESTORE LOG <Nazwa_bazy_danych>
FROM DISK = 'c:\secondaryBackupDest\Nazwa_bazy_danych.trn' WITH RECOVERY;
```
7. Jeśli aktywny dziennik transakcji jest niedostępny, bazę można odzyskać bez niego:

```
RESTORE DATABASE <Nazwa_bazy_danych> WITH RECOVERY;
```
8. Na nowym serwerze głównym wykonaj zadanie *Resolve Logins*, aby zsynchronizować loginy. Baza na serwerze pomocniczym stanie się teraz bazą serwera głównego i będzie można zacząć wprowadzać w niej zmiany.
9. Przekieruj wszystkie aplikacje do nowego serwera głównego.
10. Skonfiguruj przesyłanie dzienników z nowego serwera głównego na serwer pomocniczy. Serwer pomocniczy (wcześniejszy serwer główny) działa już w trybie *NORECOVERY*. W trakcie konfigurowania przesyłania dzienników w oknie dialogowym *Secondary Database Settings* wybierz opcję *No, the secondary database is initialized*.

11. Gdy skończysz konfigurowanie przesyłania dzienników, nowy serwer główny uruchomi zadanie tworzenia kopii zapasowej dziennika transakcji, a serwer pomocniczy skopiuje i przywróci pliki dziennika. Ustaw i włącz wszystkie zadania narzędzia SQL Server Agent, które synchronizowały dane z dawnym serwerem głównym (np. do synchronizowania loginów i użytkowników baz danych między serwerami).

Nieplanowane przełączanie awaryjne

Gdy serwer główny nieoczekiwanie stanie się niedostępny, może nastąpić utrata części danych. Możliwe, że nie da się utworzyć kopii zapasowej aktywnego dziennika transakcji lub niektóre kopie zapasowe są niedostępne. Dlatego przy nieplanowanym przełączaniu awaryjnym trzeba sprawdzić, czy ostatnie kopie dziennika zostały przywrócone. Należy posłużyć się mechanizmami monitorowania przesyłania dzienników i raportami na ten temat. Jeśli kopia zapasowa aktywnego dziennika transakcji jest dostępna, trzeba ją przywrócić, aby zsynchronizować serwer pomocniczy z głównym do punktu wystąpienia awarii. Następnie trzeba przywrócić pomocniczą bazę danych za pomocą instrukcji z opcją RECOVERY.

Po ocenie uszkodzenia i naprawieniu dawnego serwera głównego często trzeba zmienić konfigurację przesyłania dzienników, ponieważ aktywny dziennik transakcji jest niedostępny lub łańcuch dzienników został przerwany. W trakcie konfigurowania przesyłania dzienników w oknie dialogowym *Secondary Database Settings* możesz albo ustawić przywracanie z wcześniejszej kopii zapasowej bazy danych, albo wygenerować nową kopię zapasową bazy. Możesz też przełączyć rolę, aby przywrócić dawnemu serwerowi głównemu jego pierwotne funkcje.

Przełączanie między serwerami głównym i pomocniczym

Przy planowanym przełączaniu awaryjnym w ramach zmiany ról należy odpowiednio wykorzystać zadania związane z przesyłaniem dzienników, dostępne na serwerach głównym i pomocniczym. Aby przełączyć się między serwerami głównym i pomocniczym, wykonaj następujące kroki.

1. Zatrzymaj i wyłącz zadanie tworzące kopię zapasową dziennika transakcji na serwerze głównym.
2. Sprawdź, czy wszystkie kopie zapasowe dzienników transakcji zostały skopiowane i przywrócone. Możesz to zrobić ręcznie lub za pomocą zadań narzędzia SQL Server Agent.
3. Wykonaj zadanie *Sync Logins*.
4. Zatrzymaj i wyłącz zadania kopiowania i przywracania dzienników transakcji na serwerze pomocniczym.
5. Utwórz kopię zapasową aktywnego dziennika transakcji na serwerze głównym; użyj opcji NORECOVERY.
6. Przywróć kopię zapasową aktywnego dziennika transakcji na serwerze pomocniczym.
7. Przywróć bazę danych na serwerze pomocniczym; użyj opcji RECOVERY.
8. Uruchom zadanie *Resolve Logins*.
9. Włącz tworzenie kopii zapasowych dziennika transakcji na nowym serwerze głównym, co pozwoli przysyłać dzienniki na serwer pomocniczy.
10. Włącz zadania kopiowania i przywracania dzienników transakcji na serwerze pomocniczym.
11. Włącz synchronizację loginów i użytkowników baz danych.
12. Włącz pozostałe zadania narzędzia SQL Server Agent.

Przekierowywanie klientów na serwer pomocniczy

Po zmianie ról trzeba klientów przekierować na serwer pomocniczy i zminimalizować przy tym zakłócenia w pracy użytkowników. Przesyłanie dzienników nie ma wbudowanego mechanizmu przekierowywania klientów, dlatego trzeba zastosować inne rozwiązanie. To, jakie podejście wybierzesz, zależy od infrastruktury i tego, kto ma nad nią kontrolę, a także od liczby modyfikowanych klientów, wymaganej dostępności aplikacji (możliwe, że obowiązuje umowa o wymaganych poziomie świadczonych usług) i aktywności tych aplikacji. W najlepszej sytuacji użytkownicy odczują krótkie zakłócenie w momencie przekierowywania aplikacji klienckich. W kolejnych podpunktach opisujemy kilka standardowych sposobów przekierowywania klientów na serwer pomocniczy.

Kod aplikacji

Możesz napisać aplikację z obsługą przełączania awaryjnego, która będzie łączyć się z serwerem pomocniczym za pomocą automatycznego ponawiania prób lub po ręcznej zmianie nazwy serwera. Aplikacja powinna najpierw próbować połączyć się z serwerem głównym. Jeśli ten będzie niedostępny i próby ponownego nawiązania połączenia zakończą się niepowodzeniem, aplikacja ma łączyć się z serwerem pomocniczym (jeśli został ustawiony jako główny).

Po przywróceniu pomocnicza baza danych czasem nie jest jeszcze gotowa do obsługi żądań użytkowników. Możliwe, że najpierw trzeba uruchomić kilka zadań (np. zadanie *Resolve Logins*). Aby rozwiązać ten problem, bazę należy uruchomić w trybie jednego użytkownika. Pozwala to zapobiec łączeniu się z bazą innych osób w czasie, gdy administrator wykonuje potrzebne zadania. Dlatego aplikacja musi radzić sobie z sytuacjami, w których serwer główny nie jest już dostępny, a serwer pomocniczy jeszcze nie jest gotowy.

Równoważenie obciążenia sieci

Zastosuj mechanizmy równoważenia obciążenia sieci, takie jak NLB (ang. *Network Load Balancing*) z systemu Windows lub rozwiązanie sprzętowe. W tym podejściu aplikacja łączy się za pomocą nazwy lub adresu IP mechanizmu równoważenia obciążenia, a mechanizm kieruje aplikację do serwera bazodanowego. Dlatego przy przełączaniu awaryjnym aplikacja nadal łączy się za pomocą nazwy lub adresu IP mechanizmu równoważenia obciążenia, a ten mechanizm należy zaktualizować (ręcznie lub przy użyciu skryptu) i określić w nim nazwę sieciową oraz adres IP nowego serwera głównego. Powoduje to przekierowanie klientów. Mechanizmy NLB są dostępne w niektórych wersjach systemu Windows. Ich konfigurowanie jest proste. Taki mechanizm może kierować aplikację do serwera głównego.

Serwery DNS

Serwer DNS przekształca nazwy na adresy IP i może służyć do kierowania klientów do nowego serwera głównego. Jeśli masz dostęp do serwera DNS, po przełączeniu awaryjnym możesz zmodyfikować adres IP, aby przekierować aplikacje klienckie. Użyj do tego skryptu lub narzędzia do zarządzania usługami DNS w systemie Windows. Serwer DNS działa jak odsyłacz dla aplikacji klienckich, ponieważ mogą one używać tej samej nazwy w połączeniach, natomiast zmiany na serwerze DNS powodują przekierowywanie żądań do bazy danych na nowy serwer główny.

Aliasy dla klientów systemu SQL Server

Aliasy w systemie SQL Server to następna technika umożliwiająca przekierowywanie klientów na nowy serwer główny. Aby skonfigurować aliasy, otwórz program SQL Server Configuration Manager i wybierz węzeł *SQL Native Client Configuration*. Metoda ta jest mniej przydatna, jeśli duża liczba aplikacji klienckich bezpośrednio łączy się z serwerem bazodanowym, ponieważ wtedy trzeba utworzyć alias na każdym komputerze klienckim. Czasem jest to niewykonalne. Łatwiej zastosować tę technikę, gdy aplikacje klienckie łączą się z serwerem WWW lub serwerem aplikacji, który sam łączy się z serwerem bazodanowym. Wtedy alias można ustawić na serwerze WWW lub serwerze aplikacji, a wszystkie klienty zostaną przekierowane na nowy serwer główny.

Plan tworzenia kopii zapasowych baz danych

Niezależnie od stosowania rozwiązania zapewniającego wysoką dostępność zaleca się opracowanie planu tworzenia kopii zapasowych baz danych, aby chronić informacje przed uszkodzeniem i błędami użytkowników. Pomocnicza baza danych to za mało, ponieważ uszkodzenie głównej bazy danych może spowodować błędy także w bazie pomocniczej. Przy pracy nad planem tworzenia kopii zapasowych można wybrać dwa rozwiązania — z pełną kopią i z kopią różnicową.

Pełna kopia zapasowa bazy danych obejmuje wszystkie dane oraz dzienniki transakcji i jest zapisywana na urządzeniu z kopiami zapasowymi, którym zwykle jest napęd taśmowy lub dyskowy. Ta kopia jest używana po nieodwracalnej awarii, co pozwala przywrócić bazę do punktu w czasie, w którym po raz ostatni zapisano jej kopię zapasową. Pełne przywracanie bazy danych jest też niezbędnym punktem wyjścia do przywracania kopii różnicowych i dzienników transakcji. Pełną kopię zapasową należy przechowywać poza firmą, aby baza nie została utracona z powodu katastrofy.

Różnicowa kopia zapasowa bazy danych obejmuje wszystkie ekstenty zmodyfikowane od czasu utworzenia ostatniej pełnej kopii. *Ekstent* to jednostka alokacji pamięci, obejmująca osiem stron bazy danych. System SQL Server używa ekstentów do alokacji pamięci na obiekty bazy danych. Kopie różnicowe są mniejsze od pełnych, chyba że baza jest aktywnie używana i od czasu utworzenia ostatniej pełnej kopii wszystkie ekstenty zostały zmodyfikowane. Aby przywrócić bazę na podstawie kopii różnicowej, wcześniej trzeba utworzyć pełną kopię.

Tworzenie kopii zapasowych (pełnych lub różnicowych) bazy danych nie narusza przesyłania dzienników, pod warunkiem, że nie są wykonywane żadne operacje zmieniające dziennik transakcji. Możliwa jest jednak odwrotna sytuacja, czyli wpływ przesyłania dzienników na plany tworzenia i odzyskiwania kopii zapasowych. Oto przykłady.

- Gdy stosujesz przesyłanie dzienników, nie twórz poza tym mechanizmem kopii zapasowych dziennika transakcji, ponieważ naruszy to łańcuch dzienników. System SQL Server nie zapobiega utworzeniu dodatkowych zadań generujących kopie zapasowe dzienników transakcji w bazie, dla której stosowane jest przesyłanie dzienników.
- Jeśli utworzenie kopii zapasowej dziennika transakcji powoduje przycięcie tego dziennika, nastąpi naruszenie łańcucha dzienników. Przesyłanie dzienników przestanie wtedy działać.
- Jeśli ustawisz dla bazy model odzyskiwania SIMPLE, dziennik transakcji zostanie przycięty przez system SQL Server i przesyłanie dzienników przestanie działać.

Integrowanie przesyłania dzienników z innymi rozwiązaniami zapewniającymi wysoką dostępność

Przesyłanie dzienników można stosować razem z innymi rozwiązaniami Microsoftu zapewniającymi wysoką dostępność. Wtedy ten mechanizm pozwala odzyskać bazę po katastrofie, a dodatkowe rozwiązanie gwarantuje wysoką dostępność bazy. Wyjątkiem są tu grupy dostępności, które jednocześnie zapewniają wysoką dostępność, jak i umożliwiają odzyskiwanie baz po katastrofie. Przesyłanie dzienników można zintegrować z trzema podstawowymi rozwiązaniami.

- **Z kopiami lustrzanymi danych.** Tworzenie zdalnej jednostki używanej, gdy kopie lustrzane staną się niedostępne.
- **Z klastrami WSFC.** Tworzenie zdalnej jednostki używanej do odzyskiwania baz po katastrofie, gdy lokalny klaster WSFC stanie się niedostępny.
- **Z replikacją.** Tworzenie wysoce dostępnego wydawcy w replikacji.

Kopie lustrzane danych w systemie SQL Server 2014

Przesyłanie dzienników najlepiej współdziała z kopiami lustrzanymi danych z systemu SQL Server 2014, gdy firma lokalnie tworzy kopie lustrzane, a następnie przesyła dzienniki z głównego serwera do zdalnej jednostki, używanej do odzyskiwania baz po katastrofie. Pamiętaj jednak, że przesyłanie dzienników nie powoduje automatycznego przełączania ról związanych z kopiami lustrzanymi. Dlatego musisz ręcznie włączyć wcześniejszą kopię lustrzaną (ustawianą teraz jako serwer główny), aby rozpocząć przesyłanie dzienników transakcji z niej na serwer pomocniczy. Wymaga to skonfigurowania przesyłania dzienników z nowego serwera głównego.

Klastry WSFC

Przesyłanie dzienników bazy danych z klastra WSFC do zdalnej lokalizacji używanej na potrzeby odzyskiwania systemu po katastrofie jest najbardziej przydatne po wystąpieniu poważnych lokalnych problemów. Jeśli w trakcie odzyskiwania systemu po katastrofie klaster WSFC jest niedostępny, dzięki przesłanym dziennikom firma może kontynuować pracę z wykorzystaniem zdalnej jednostki. Jeśli firma nie utworzyła klastra obejmującego odległe lokalizacje, węzły klastra znajdują się blisko siebie i w trakcie katastrofy mogą wszystkie zostać uszkodzone.

Gdy stosujesz przesyłanie dzienników w klastrze, katalog na kopie zapasowe i katalog używany przez narzędzie BCP powinny być ustawione jako zasób klastra. Należy je umieścić w jednej grupie klastra z egzemplarzem systemu SQL Server obejmującym bazę, dla której włączone jest przesyłanie dzienników. Dzięki temu przy przełączaniu awaryjnym w klastrze katalog z kopią zapasową też zostanie uwzględniony, natomiast egzemplarz systemu SQL Server używany do przesyłania dzienników pozostanie dostępny dla drugiego węzła z klastra WSFC. W klastrze WSFC pliki z konfiguracją i danymi potrzebne drugiemu węzłowi także należy skonfigurować jako zasób klastra w grupie klastra z egzemplarzem systemu SQL Server używanym przy przesyłaniu dzienników. Jeśli np. zechcesz uruchamiać pakiet SSIS Transfer Logins Task za pomocą pliku, zamiast zapisywać go w systemie SQL Server, powinieneś dodać odpowiedni katalog jako zasób klastra, aby był dostępny dla wszystkich węzłów.

W celu ustawienia katalogu jako zasobu klastra wykonaj następujące kroki.

1. Najpierw utwórz katalog.
2. Następnie z poziomu administratora klastra utwórz nowy zasób i wybierz opcję *Shared Folder*. Koniecznie umieść ten zasób w jednej grupie z egzemplarzem systemu SQL Server z bazą, dla której włączone jest przysyłanie dzienników.
3. Teraz w klastrze WSFC możesz ustawić w systemie SQL Server zależność od tego katalogu. Jeśli katalog nie jest jeszcze dostępny, system SQL Server będzie czekał na ten zasób.

UWAGA Przy ustawianiu zasobów jako zależności uwzględnij, czy dany zasób musi być dostępny przed uruchomieniem systemu SQL Server. Jeśli np. katalog z kopiami zapasowymi przeznaczony na pliki dzienników transakcji jest niedostępny, system SQL Server nie musi na niego czekać, ponieważ baza już działa i udostępnienie jej użytkownikom jest ważniejsze niż tworzenie kopii zapasowych dzienników transakcji.

Replikacja w systemie SQL Server 2014

Gdy w replikacji system SQL Server 2014 pełniący funkcję wydawcy jednocześnie przechowuje wszystkie dane, może stać się on pojedynczym punktem podatności na awarię. Wszyscy subskrybenci łączą się z dystrybutorem, który następnie nawiązuje połączenie z wydawcą, aby otrzymać aktualizację danych. Jeśli wydawca zawiedzie, replikacja zatrzyma się do momentu ponownego jego uruchomienia. Przesyłanie dzienników można w tym modelu wykorzystać do zapewniania wysokiej dostępności, aby wydawca nie stał się pojedynczym punktem podatności na awarię. Jest to możliwe w replikacji transakcyjnej i ze skalaniem.

Przesyłanie dzienników ma chronić wydawcę. Serwer główny, który jest jednocześnie wydawcą, przesyła dziennik transakcji na serwer pomocniczy.

Przy przełączaniu ról wykonaj kroki opisane w punktach „Przełączanie ról między serwerami głównym i pomocniczym” oraz „Przekierowywanie klientów na serwer pomocniczy”.

Ponadto przy stosowaniu replikacji transakcyjnej z przysyłaniem dzienników uwzględnij następujące kwestie.

- Aby replikacja działała po przełączeniu ról, konfiguracja serwerów głównego i pomocniczego musi być identyczna.
- By w replikacji transakcyjnej zapobiec konieczności przysyłania danych na serwer pomocniczy przez subskrybentów, na serwerze głównym należy zastosować parametr `sync with backup` w poleceniu tworzącym kopie zapasowe dzienników transakcji na potrzeby przysyłania dzienników. Wtedy transakcja jest replikowana po stronie subskrybentów dopiero po wykonaniu kopii zapasowej dziennika transakcji. Wprowadza to opóźnienie. Subskrybenci nie działają wtedy w czasie rzeczywistym, ponieważ replikacja musi czekać na proces przysyłania dzienników. Aby ograniczyć opóźnienie, skróć czas między tworzeniem kolejnych kopii zapasowych dziennika transakcji. Jeśli nie zastosujesz parametru `sync with backup`, może się zdarzyć, że dane subskrybenta nie będą dopasowane do danych wydawcy przy przełączaniu ról serwerów uczestniczących w przysyłaniu dzienników. Możliwa jest wtedy utrata danych.
- Przy replikacji ze skalaniem po przełączeniu ról wydawca może zsynchronizować zmiany utracone w trakcie przełączania. W tym celu należy przeprowadzić replikację ze skalaniem z każdym z subskrybentów. Częściej stosowane rozwiązanie zapewniające wysoką

dostępność i zapobiegające powstaniu pojedynczego punktu podatności na awarię na serwerze wydawcy to skonfigurowanie klastra WSFC. Wtedy po awarii można przełączyć replikację do drugiego węzła klastra.

Kończenie przesyłania dzienników

Przed usunięciem bazy, dla której stosowane jest przełączanie dzienników, trzeba usunąć elementy związane z tym mechanizmem. Oznacza to usunięcie wszystkich harmonogramów, zadań, historii i informacji o błędach. Istnieją dwa sposoby na zakończenie przesyłania dzienników — wykorzystanie programu SQL Server 2014 Management Studio lub użycie języka T-SQL. Możliwe, że przed wyłączeniem przesyłania dzienników zechcesz zapisać konfigurację tego mechanizmu w skrypcie. Pozwoli to w przyszłości szybko ponownie uruchomić przesyłanie dzienników.

Kończenie przesyłania dzienników w programie SQL Server Management Studio

Aby zakończyć przesyłanie dzienników za pomocą programu SQL Server Management Studio, wykonaj następujące kroki.

1. Nawiąż połączenie z serwerem głównym w programie SQL Server Management Studio.
2. Otwórz właściwości bazy danych z serwera głównego.
3. Na liście *Select a Page* wybierz opcję *Transaction Log Shipping*.
4. Usuń zaznaczenie pola *Enable this as a primary database in a log shipping configuration* i kliknij przycisk OK.
5. Jeśli trzeba, usuń serwer pomocniczy z właściwości bazy danych z serwera głównego. W sekcji *Secondary Databases* wybierz egzemplarz serwera pomocniczego i kliknij przycisk *Remove*.
6. Aby usunąć egzemplarz serwera monitorującego, wyłącz zaznaczenie pola *Use a monitor server instance*.

Kończenie przesyłania dzienników za pomocą języka T-SQL

Aby zakończyć przesyłanie dzienników za pomocą języka T-SQL, wywołaj poniższe polecenie na serwerze głównym:

```
Use Master;
sp_delete_log_shipping_primary_secondary @primary_database, @secondary_server,
@secondary_database;
```

Instrukcja usuwa z tabeli `msdb.dbo.log_shipping_primary_secondaries` na serwerze głównym informacje o serwerze pomocniczym.

Na serwerze pomocniczym wywołaj poniższe polecenie:

```
Use Master;
sp_delete_log_shipping_secondary_database @secondary_database;
```

Instrukcja wywołuje procedurę składowaną `sys.sp_delete_log_shipping_secondary_database_internal`, która usuwa z serwera pomocniczego informacje o nim, a także wykonywane na nim zadania.

Wróć na serwer główny i wywołaj następujące polecenie:

```
Use Master;  
sp_delete_log_shipping_primary_database @database;
```

Instrukcja usuwa z serwera głównego informacje i zadania związane z przesyłaniem dzienników, usuwa informacje o serwerze monitorującym i sam serwer, a także kasuje tabelę `msdb.dbo.log_shipping_primary_databases`.

Następnie, jeśli trzeba, można usunąć pomocniczą bazę danych.

Wydajność przesyłania dzienników

- Wydajność przesyłania dzienników jest bardzo ważna, jeśli przełączanie awaryjne ma być szybkie i spełniać wymagania związane z aktywnym serwerem zapasowym (opisane np. w umowie o gwarantowanym poziomie usług). Dlatego trzeba dopracować i zoptymalizować wymienione poniżej ważne kwestie, aby obsłużyć dodatkowe obciążenie spowodowane przesyłaniem dzienników. Potrzebna jest sieć do kopiowania plików dzienników transakcji z serwera głównego na pomocniczy.
- Urządzenia wejścia-wyjścia obsługujące udziały plikowe na serwerze głównym (gdzie tworzone są kopie zapasowe dzienników transakcji) i pomocniczym (który odbiera dzienniki transakcji) muszą być wydajne. Rozważ umieszczenie katalogu na kopie zapasowe dzienników transakcji na innym dysku niż ten, na którym znajdują się pliki bazy danych. Ponadto kompresja kopii zapasowych przyspiesza wykonywanie operacji wejścia-wyjścia. W ramach codziennych zadań administracyjnych śledź liczniki wydajności operacji wejścia-wyjścia pod kątem zatorów. Przykładowo średni czas w sekundach na odczyt i średni czas w sekundach na zapis powinien wynosić mniej niż 10 milisekund.
- Aby serwer pomocniczy był szybciej zsynchronizowany z serwerem głównym, w miarę możliwości staraj się stosować krótsze transakcje w bazie danych. Dziennik z danymi transakcjami jest przesyłany dopiero po ich zatwierdzeniu w głównej bazie danych.
- Zadania administracyjne na bazie danych (np. defragmentację indeksu) przeprowadzaj w okresie obniżonej aktywności systemu. W zależności od poziomu fragmentacji plik dziennika transakcji może być duży, co wpływa na długość tworzenia kopii zapasowej, jej kopiowania i przywracania, a to przekłada się na opóźnienie w synchronizacji.
- Aby po przełączeniu awaryjnym wydajność systemu została zachowana, serwery główny i pomocniczy powinny mieć identyczną konfigurację sprzętu i oprogramowania.

UWAGA Edycje Standard i Enterprise systemu SQL Server 2012 obsługują kompresję kopii zapasowych. Przy przesyłaniu dzienników używane są domyślne ustawienia kompresji kopii zapasowych stosowane na serwerze. To, czy kopia zapasowa dziennika zostanie skompresowana, zależy od ustawienia *Backup Compression*.

Wymienione techniki pozwalają zsynchronizować serwer pomocniczy z głównym oraz zapewnić oczekiwaną przez użytkowników wydajność po zmianie ról. Warto też oddzielić katalog na

kopie zapasowe od katalogu bazy danych i w ramach codziennych zadań administracyjnych obserwować liczniki wydajności operacji wejścia-wyjścia. Ponadto trzeba śledzić przepustowość sieci, aby mieć pewność, że możliwe będzie szybkie przesyłanie dzienników transakcji.

Aktualizacja przesyłania dzienników do wersji z systemu SQL Server 2014

Przy aktualizowaniu istniejących mechanizmów przesyłania dzienników do wersji z systemu SQL Server 2014 musisz wykonać określone kroki, aby zapobiec błędom w konfiguracji na serwerach głównym i pomocniczym. Istnieją trzy standardowe techniki aktualizowania przesyłania dzienników: podejście z minimalnym przestojem, metoda ze zwykłym przestojem i sposób z instalacją przesyłania dzienników. Wszystkie te podejścia omawiamy w następnych punktach.

Podejście z minimalnym przestojem

Technika wymaga rozpoczęcia prac od aktualizacji serwera pomocniczego w miejscu za pomocą programu instalacyjnego systemu SQL Server 2014. Nie uszkadza to mechanizmu przesyłania dzienników, ponieważ dzienniki transakcji z wersji SQL Server 2008, 2008 R2 i 2012 można stosować także w wydaniu SQL Server 2014. Następnie należy przeprowadzić planowane przełączanie awaryjne ze standardowymi procedurami przełączania ról i skierować użytkowników na nowy serwer główny (czyli dawny serwer pomocniczy). Po przełączeniu ról i przekierowaniu użytkowników wykonaj następujące kroki.

1. Zatrzymaj i wyłącz zadania związane z przesyłaniem dzienników. Świeżo zaktualizowany serwer główny z systemem SQL Server 2014 nie może przysyłać dzienników do dawnego serwera głównego z systemem SQL Server 2008, 2008 R2 lub 2012.
2. Przeprowadź aktualizację w miejscu na dawnym serwerze głównym za pomocą programu instalacyjnego systemu SQL Server 2014. Pozostaw ten serwer jako pomocniczy.
3. Skonfiguruj w systemie SQL Server 2014 przesyłanie dzienników z nowego serwera głównego na serwer pomocniczy. Wykorzystaj te same współużytkowane katalogi, które stosowano przy przesyłaniu dzienników w systemie SQL Server 2008, 2008 R2 lub 2012. Ponadto w zakładce *Initialize Secondary Database* w oknie dialogowym *Secondary Database Settings* zaznacz opcję *No, the secondary database is initialized*. Jeśli chcesz, możesz ponownie przełączyć role serwerów.

Podejście ze standardowym przestojem

W tej metodzie serwery główny i pomocniczy nie są dostępne w trakcie aktualizacji, ponieważ oba są aktualizowane w miejscu za pomocą programu instalacyjnego. Technikę możesz stosować, gdy masz zaplanowany przestój na aktualizację serwerów z systemem SQL Server 2008, 2008 R2 lub 2012. Jest to prostsze podejście, ponieważ nie wymaga przełączania awaryjnego.

1. Sprawdź, czy serwer pomocniczy jest zsynchronizowany z serwerem głównym. W tym celu przywróć wszystkie kopie zapasowe dzienników transakcji i aktywnego dziennika transakcji z serwera głównego.

2. Zatrzymaj i wyłącz zadania związane z przesyłaniem dzienników. Następnie przeprowadź aktualizację w miejscu na serwerze pomocniczym, po czym zaktualizuj serwer główny.
3. Zmień konfigurację przesyłania dzienników w systemie SQL Server 2014. Wykorzystaj przy tym używane wcześniej współużytkowane serwery. Ponadto w zakładce *Initialize Secondary Database* w oknie dialogowym *Secondary Database Settings* zaznacz opcję *No, the secondary database is initialized*. W systemie SQL Server 2014 rozpocznie się przesyłanie dzienników. Nie wymaga to tworzenia kopii zapasowej bazy danych i przywracania jej na serwerze pomocniczym.

Podejście z instalowaniem przesyłania dzienników

Stosowanie tej metody jest uzasadnione, gdy bazy danych, dla których działa przesyłanie dzienników, są niewielkie i można szybko utworzyć ich kopie zapasowe, a następnie skopiować je i przywrócić na serwerze pomocniczym, zaktualizowanym już do wersji SQL Server 2014. Technika jest podobna do podejścia ze standardowym przestojem, jednak po aktualizacji, w trakcie konfigurowania przesyłania dzienników w systemie SQL Server 2014, w zakładce *Initialize Secondary Database* wybierz opcję *Yes, generate a full backup of the primary database*. Mechanizm przesyłania dzienników wykona wtedy kopię zapasową bazy na serwerze głównym i przywróci tę kopię na serwerze pomocniczym, aby rozpocząć przesyłanie dzienników.

Podsumowanie

Przesyłanie dzienników to proste, niedrogie i solidne rozwiązanie z systemu SQL Server zapewniające wysoką dostępność. Jest ono stosowane już od dawna. Aby zastosować je jako mechanizm odzyskiwania systemu po katastrofie, utwórz serwer pomocniczy w zdalnej lokalizacji. Pozwoli to zapewnić ciągłość pracy firmy po wystąpieniu lokalnej katastrofy, awarii zasilania lub problemów z siecią. Dzienniki można przysyłać do dowolnego miejsca na świecie. Jedynym ograniczeniem jest przepustowość sieci; musi ona umożliwiać szybki transfer plików dzienników transakcji. Przesyłanie dzienników można też zastosować na potrzeby generowania raportów, gdy firma potrzebuje danych z określonego punktu w czasie, a nie najnowszych informacji. Ponadto przesyłanie dzienników wykorzystuje się zamiast klastrów WSFC do zapewniania wysokiej dostępności; podejście to jest tańsze od klastrów, ponieważ nie wymaga współużytkowanego systemu dyskowego.

Główna trudność związana z przesyłaniem dzienników polega na tym, że nie zapewniają automatycznego przekierowywania klientów. Dlatego operator musi o to zadbać. Ponadto przełączanie ról wymaga interwencji użytkownika (trzeba się upewnić, że wszystkie dzienniki transakcji zostały przywrócone na serwerze pomocniczym). Także ponowne przełączenie ról wymaga ręcznego wykonania określonych zadań. Innym problemem jest to, że na poziomie baz danych użytkowników nie są kopiowane używane na serwerze głównym nowe loginy, zadania narzędzia SQL Server Agent, zadania usług Integration Services ani serwery połączone. Jednak za pomocą zadań usług Integration Services i narzędzia SQL Server Agent można przenieść potrzebne informacje.

Niezależnie od zagrożeń i wyzwań wiele firm z powodzeniem używa przesyłania dzienników i cały proces (włącznie z przełączaniem ról, synchronizacją loginów i przekierowywaniem klientów) obsługuje za pomocą skryptów. Pozwala to na szybkie przełączanie awaryjne z minimalnym czasem przestoju potrzebnym na ręczne wykonywanie zadań.

Ponadto ponieważ przesyłanie dzienników obejmuje przynajmniej dwa fizycznie niezależne serwery, które (inaczej niż w klastrach WSFC) nie współużytkują żadnych komponentów, można najpierw zainstalować poprawki na jednym serwerze, potem przełączyć role i dodać poprawki do drugiego serwera.

Kopie lustrzane baz danych to niezawodne rozwiązanie zapewniające wysoką dostępność. Działa ono w edycjach Standard i Enterprise systemu SQL Server 2014. Może obsługiwać automatyczne przełączanie awaryjne, a także synchroniczny i asynchroniczny transfer danych (szczegółowe informacje na ten temat podano w rozdziale 19.). Obecnie kopie lustrzane częściej stosuje się w edycji Standard, ponieważ edycja Enterprise obsługuje oferujące większe możliwości grupy dostępności.

Grupy dostępności to następna technika zapewniania wysokiej dostępności, która w przyszłości może zastąpić przesyłanie dzienników. Jednak przesyłanie dzienników to sprawdzone rozwiązanie. Jest ono stosowane w wielu firmach i w licznych strategicznych aplikacjach, gdzie zapewnia wysoką dostępność i możliwość odzyskania systemu po katastrofie.

Kopie lustrzane baz danych

ZAWARTOŚĆ ROZDZIAŁU:

- Wprowadzenie do kopii lustrzanych baz danych.
- Architektura kopii lustrzanych baz danych.
- Tryby działania kopii lustrzanych.
- Instalowanie, konfigurowanie i monitorowanie.
- Rozwiązywanie problemów z kopiami lustrzanymi.
- Scenariusze zapewniania dostępności bazy danych.

Kopie lustrzane baz danych to niezawodne rozwiązanie zapewniające wysoką dostępność. Jest ono obsługiwane w edycjach Standard i Enterprise systemu SQL Server 2014. Obecnie częściej stosuje się je w edycji Standard, ponieważ wersja Enterprise obsługuje grupy dostępności, które oferują więcej możliwości.

UWAGA W wersji SQL Server 2014 kopie lustrzane baz danych zostały uznane za przestarzałe. Nadal są one obsługiwane, ale w przyszłości zostaną usunięte. Pełną listę funkcji silnika bazodanowego systemu SQL Server 2014 uznanych za przestarzałe zamieszczono na stronie <http://msdn.microsoft.com/en-us/library/ms143729%28v=sql.120%29.aspx>.

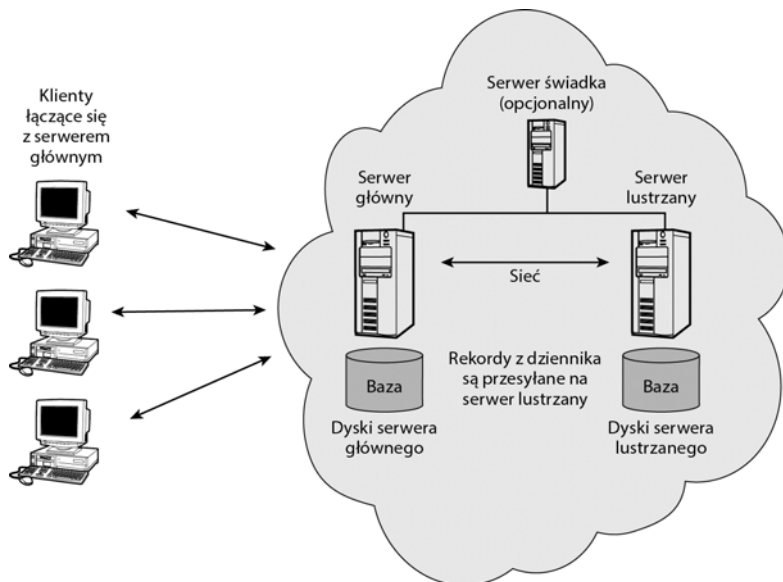
Maksymalizacja dostępności bazy danych to priorytetowe zadanie większości administratorów. Trudno opisać, co przeżywa administrator, gdy baza jest niedostępna. Nie da się jej szybko przywrócić, a jednocześnie trzeba odpowiadać na trudne pytania zadawane przez menedżera. W tym rozdziale pokazujemy, że w takich sytuacjach kopie lustrzane baz danych pozwalają czasem wyjść z opresji.

Kopie lustrzane baz danych pomagają przywrócić dostęp do bazy. Wymaga to automatycznego lub ręcznego przełączenia awaryjnego na kopię lustrzaną. Jest to następny mechanizm zapewniania wysokiej dostępności obsługiwany w systemie SQL Server 2014. W tym rozdziale wyjaśniamy zagadnienia związane z kopiami lustrzanymi baz danych, pokazujemy, jak zarządzać bazami, dla których używany jest ten mechanizm, i przedstawiamy przykłady ilustrujące stosowanie tej techniki. Opisujemy też snapshoty baz danych; można je wykorzystać do wczytywania baz, dla których tworzone są kopie lustrzane.

Wprowadzenie do kopii lustrzanych baz danych

Kopie lustrzane to mechanizm zapewniający wysoką dostępność. Działa na poziomie baz danych i stosuje się go dla poszczególnych baz. Aby zmaksymalizować dostępność bazy, należy zminimalizować planowane i nieplanowane przestoje. Planowane przestoje są częste. Wynikają np. z konieczności wprowadzenia zmian w systemie produkcyjnym, aktualizowania sprzętu lub oprogramowania (instalowania poprawek bezpieczeństwa lub pakietów SP), zmieniania konfiguracji bazy lub aktualizowania magazynu danych. Wszystkie te operacje wymagają krótkiego (jeśli wszystko przebiega zgodnie z planem) zablokowania dostępu do bazy lub serwera. Nieplanowane przestoje są powodowane przez awarie sprzętu (np. dysków), brak zasilania, błędy ludzkie lub naturalne katastrofy. Każdy z tych czynników może spowodować brak dostępu do serwera produkcyjnego lub do centrum danych.

Na rysunku 19.1 przedstawiono różne aspekty kopii lustrzanych, omówione szczegółowo dalej w tym rozdziale.



Rysunek 19.1. Architektura kopii lustrzanych

W omawianej technologii tworzone są dwie kopie bazy danych. Znajdują się one na odrębnych egzemplarzach systemu SQL Server, zwykle na innych komputerach. Można używać różnych egzemplarzy systemu SQL Server 2014 na tym samym komputerze, jednak nie zapewnia to wysokiej dostępności; to podejście jest zwykle stosowane podczas testów.

W danym momencie tylko jedna kopia bazy jest dostępna dla klientów. Jest to *główna baza danych*. System SQL Server przechowujący główną bazę danych to *serwer główny*. Kopie lustrzane są tworzone za pomocą przesyłania i wprowadzania rekordów z dziennika bazy danych do jej kopii. Kopia bazy to *kopia lustrzana*. System SQL Server z kopią lustrzaną to *serwer lustrzany*.

Serwery główny i lustrzany są *partnerami w sesji* tworzenia kopii lustrzanych. Mechanizm tworzenia takich kopii wprowadza każdą modyfikację (spowodowaną operacjami DML, DDL itd.) z głównej bazy danych w bazie lustrzanej. Dotyczy to także fizycznych i logicznych zmian w bazie, wprowadzanych np. w plikach i indeksach. Dany serwer może być serwerem głównym dla jednej bazy danych i serwerem lustrzanym dla innej bazy. Przy automatycznym przełączaniu awaryjnym potrzebna jest trzecia jednostka — *serwer świadka*. Serwer świadka opisujemy dalej w tym rozdziale, w punkcie „Tryb wysokiego bezpieczeństwa z automatycznym przełączaniem awaryjnym”.

Tworzenie kopii lustrzanych baz pomaga zminimalizować planowane i nieplanowane przestoje na kilka sposobów.

- Umożliwia automatyczne lub ręczne przełączanie awaryjne na lustrzane bazy danych.
- Zapewnia, że lustrzana baza danych jest aktualną kopią głównej bazy. Można do tego używać trybów synchronicznego lub asynchronicznego. Tryby tworzenia kopii lustrzanych omawiamy w następnym podrozdziale, „Tryby działania kopii lustrzanych”.
- Umożliwia przechowywanie lustrzanej bazy w zdalnym centrum danych, co może być przydatne przy odzyskiwaniu systemu po katastrofie.

UWAGA Nie można tworzyć kopii lustrzanych baz master, msdb, tempdb i model. Możesz jednak utworzyć kopie lustrzane wielu baz z danego egzemplarza systemu SQL Server. Tworzenie kopii lustrzanych dla kopii nie jest dozwolone. Ponadto główna baza danych nie może mieć więcej niż jednego partnera w postaci kopii lustrzanej.

Tryby działania kopii lustrzanych

Aby lustrzana baza danych była aktualna, mechanizm tworzenia kopii lustrzanych przesyła strumień rekordów dziennika do bazy lustrzanej i je w niej wprowadza. Należy określić, który *tryb działania* jest ustawiony dla mechanizmu tworzenia kopii lustrzanych.

Sesja tworzenia kopii lustrzanych może działać w trzech trybach. Używany tryb zależy od ustawienia bezpieczeństwa transakcji i tego, czy dostępny jest serwer świadka. Tryby działania kopii lustrzanych opisane są w tabeli 19.1.

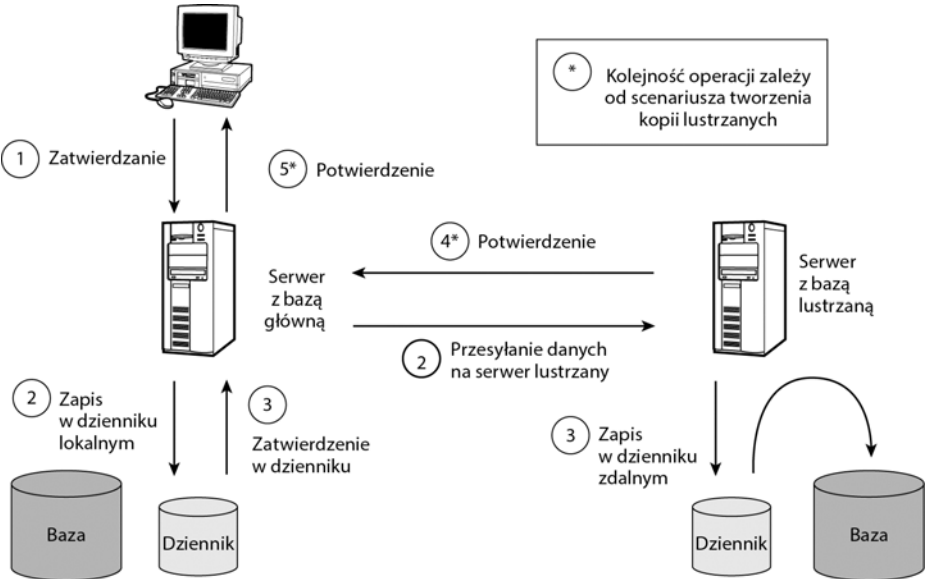
W trakcie konfigurowania kopii lustrzanych dostępne są dwie opcje — SAFETY FULL i SAFETY OFF. Gdy wybierasz używaną opcję, musisz zdecydować, czy chcesz, aby bazy główna i lustrzana zawsze były zsynchronizowane (opcja SAFETY FULL), czy dopuszczasz możliwość utraty części danych po awarii serwera głównego (opcja SAFETY OFF). Opcje te można ustawić w instrukcji ALTER DATABASE przy włączaniu tworzenia kopii lustrzanych.

Jeśli wybierzesz opcję SAFETY FULL, tworzenie kopii lustrzanych działa w *trybie wysokiego bezpieczeństwa* (jest to *tryb synchronicznego tworzenia kopii zapasowych*). Gdy serwer główny zapisuje rekordy dziennika głównej bazy na dysku (czyli przenosi bufor dziennika na dysk),

Tabela 19.1. Tryby działania kopii lustrzanych

Tryb działania	Bezpieczeństwo transakcji	Mechanizm transferu	Wymagane kworum	Serwer świadka	Sposób przełączania
Wysoka wydajność	OFF	Asynchroniczny	Nie	Nie dotyczy	Wymuszone przełączanie (z możliwą utratą danych); jest to operacja ręczna
Wysokie bezpieczeństwo bez automatycznego przełączania	FULL	Synchroniczny	Tak	Nie	Ręczne lub wymuszone
Wysokie bezpieczeństwo z automatycznym przełączaniem	FULL	Synchroniczny	Tak	Tak	Automatyczne lub ręczne

jednocześnie wysyła bufor dziennika na serwer lustrzany. Następnie serwer główny czeka na odpowiedź serwera lustrzanego. Serwer lustrzany wysyła potwierdzenie, gdy zapisze przesłane rekordy dziennika w swoim dzienniku transakcji. Potwierdzenie jest wtedy zgłaszane klientowi. Transfer synchroniczny gwarantuje, że wszystkie transakcje z dziennika transakcji lustrzanej bazy danych będą zsynchronizowane z dziennikiem transakcji głównej bazy danych. Dzięki temu transakcje są bezpiecznie przenoszone. Na rysunku 19.2 przedstawiono sekwencję zdarzeń wykonywanych przy ustawieniu SAFETY FULL.



Rysunek 19.2. Przebieg synchronizowania serwerów głównego i lustrzanego

Ustawienie SAFETY FULL chroni przed utratą danych i gwarantuje, że serwery główny i lustrzany są zsynchronizowane, jeśli transakcja została zatwierdzona. Ten model związany jest ze spadkiem wydajności, ponieważ transakcja zostaje zatwierdzona dopiero po zapisaniu jej w dzienniku transakcji na serwerze lustrzanym. Prowadzi to do niewielkiego, ale zauważalnego wydłużenia

czasu reakcji na działania użytkownika, a także zmniejszenia liczby przetwarzanych transakcji, ponieważ serwer główny musi czekać na to, aż serwer lustrzany potwierdzi zapisanie transakcji w dzienniku.

Wielkość opóźnienia zależy od wielu czynników, m.in. od szybkości serwera, obciążenia bazy danych, szybkości sieci, architektury aplikacji, wydajności dysku itd. Aplikacje generujące dużą liczbę niewielkich transakcji mają większy wpływ na czas reakcji niż programy generujące pojedyncze długie transakcje. Dzieje się tak, ponieważ każda transakcja wymaga oczekiwania na potwierdzenie od serwera lustrzanego, a czas oczekiwania bardziej wydłużają krótkie transakcje.

Wybór opcji SAFETY OFF powoduje tworzenie kopii lustrzanych w trybie wysokiej wydajności (jest to tryb asynchroniczny). W tym trybie dzienniki są przesyłane w taki sam sposób jak w trybie synchronicznym, ale serwer główny nie musi czekać z zatwierdzeniem transakcji na to, że serwer lustrzany potwierdzi zapisanie bufora dziennika na dysku. Bezpośrednio po kroku 3. z rysunku 19.2 transakcja jest zatwierdzana na serwerze głównym. Baza danych staje się zsynchronizowana po wykonaniu na serwerze lustrzanym zaległych operacji z serwera głównego. Ponieważ serwer lustrzany cały czas musi „doganiać” serwer główny, może to prowadzić do utraty danych przy nieoczekiwanej awarii serwera głównego. Jednak dotyczy to tylko danych niewysłanych na serwer lustrzany. W tym trybie wpływ tworzenia kopii lustrzanych na czas reakcji i liczbę przetwarzanych transakcji jest minimalny, ponieważ nie trzeba czekać z zatwierdzeniem transakcji na otrzymanie potwierdzenia z serwera lustrzanego.

Zwróć uwagę na trzy ważne pojęcia związane z kopiami lustrzanymi baz danych.

- **Kolejka rekordów do wysłania.** Jeśli rekordów z dziennika z serwera głównego na lustrzany nie można przesyłać tak szybko, jak są generowane, na serwerze głównym powstaje *kolejka rekordów do wysłania*. Kolejka ta nie wymaga dodatkowej pamięci. Jest w całości przechowywana w dzienniku transakcji serwera głównego. Obejmuje część dziennika, która nie została jeszcze przesłana na serwer lustrzany.
- **Kolejka rekordów do wprowadzenia.** Jeśli na serwerze lustrzanym nie można wprowadzać rekordów z taką szybkością, z jaką są nadsyłane, powstaje na tym serwerze *kolejka rekordów do wprowadzenia*. Kolejka (podobnie jak kolejka rekordów do wysłania) nie wymaga dodatkowej pamięci. Znajduje się w całości w dzienniku transakcji serwera lustrzanego. Obejmuje część zapisanego dziennika, którą trzeba wprowadzić w lustrzanej bazie danych. Zwykle do wprowadzania rekordów używany jest jeden wątek, jednak w edycji SQL Server Enterprise obsługiwane jest *wprowadzanie równoległe*, pozwalające zastosować jeden wątek na każde cztery jednostki obliczeniowe (co odpowiada czterem rdzeniom).
- **Kompresja strumienia rekordów.** Przy transferze danych między partnerami stosowana jest *kompresja strumienia rekordów*. Pozwala to zmniejszyć obciążenie sieci i uzyskać kompresję na poziomie przynajmniej 12,5%. Serwer główny kompresuje dane przed ich wysłaniem na serwer lustrzany, a serwer lustrzany wypakowuje dane przed ich zastosowaniem. Kompresja i wypakowywanie powodują niewielki wzrost obciążenia procesora na serwerach głównym i lustrzanym, natomiast pozwalają ograniczyć obciążenie sieci. Jest to przydatne zwłaszcza wtedy, gdy w ciągu dnia wprowadzanych jest bardzo dużo zmian w bazie, ponieważ można wtedy zmniejszyć ilość zasobów sieciowych potrzebnych do tworzenia kopii lustrzanych.

Przykłady tworzenia kopii lustrzanych baz danych

Skoro wiesz już, od czego zależy bezpieczeństwo transakcji, przyjrzyj się przykładowi, aby lepiej zrozumieć tryby działania i inne aspekty kopii lustrzanych. W tym przykładzie będziesz potrzebował trzech egzemplarzy systemu SQL Server: serwera głównego, serwera lustrzanego i serwera świadka. Zastosuj tryb wysokiego bezpieczeństwa z automatycznym przełączaniem awaryjnym, co pokazano na rysunku 19.3. W tym przykładzie zakładamy, że wszystkie trzy egzemplarze systemu SQL Server działają w sieci i korzystają z tego samego konta domeny. Jest to konto Administrator mające dostęp do wszystkich używanych egzemplarzy systemu SQL Server.

Serwer główny

1. Tworzenie certyfikatu na serwerze głównym
2. Tworzenie punktu końcowego na serwerze głównym
7. Tworzenie loginu i przyznawanie uprawnień CONNECT na serwerze głównym
10. Tworzenie kopii zapasowej bazy AdventureWorks na serwerze głównym
12. Wstawianie i modyfikowanie danych na serwerze głównym
13. Tworzenie kopii zapasowej dziennika transakcji na serwerze głównym
15. Konfigurowanie serwera głównego

Serwer lustrzany

3. Tworzenie certyfikatu na serwerze lustrzanym
4. Tworzenie punktu końcowego na serwerze lustrzanym
8. Tworzenie loginu i przyznawanie uprawnień CONNECT na serwerze lustrzanym
11. Przywracanie bazy AdventureWorks na serwerze lustrzanym
14. Konfigurowanie serwera lustrzanego
16. Przywracanie dziennika transakcji na serwerze lustrzanym

Serwer świadka

5. Tworzenie certyfikatu na serwerze świadka
6. Tworzenie punktu końcowego na serwerze świadka
9. Tworzenie loginu i przyznawanie uprawnień CONNECT na serwerze świadka
17. Konfigurowanie serwera świadka

Rysunek 19.3. Operacje wykonywane przy tworzeniu kopii lustrzanych bazy danych

Przygotowywanie punktów końcowych

Aby partnerzy w procesie tworzenia kopii lustrzanych baz mogli się połączyć, muszą sobie ufać. Zaufane połączenie jest tworzone za pomocą punktów końcowych TCP (ang. *Transmission Control Protocol*). Dlatego na każdym z serwerów trzeba utworzyć punkt końcowy przy użyciu instrukcji CREATE ENDPOINT języka T-SQL i z wykorzystaniem polecenia GRANT CONNECT ON ENDPOINT przyznać uprawnienia do łączenia się z punktami końcowymi. Punkty końcowe działają tu w sposób opisany w rozdziale 8., „Zabezpieczanie egzemplarzy baz danych” i obowiązują tu te same reguły. Jedyna różnica polega na tym, że zamiast tworzyć punkt końcowy Service Brokera, należy przygotować punkt końcowy dla kopii lustrzanych baz. Obowiązują przy tym te same zasady bezpieczeństwa — można zastosować uwierzytelnianie z systemu Windows lub certyfikaty.

W tym przykładzie używamy certyfikatów, dzięki czemu nauczysz się korzystać z nich do uwierzytelniania przesyłu danych. Uwierzytelnianie z systemu Windows łatwo skonfigurować, jednak tu nie omawiamy tego zagadnienia.

UWAGA Wszystkie przykładowe skrypty z tego rozdziału znajdziesz na poświęconej tej książce stronie w witrynie wydawnictwa Helion. Przed uruchomieniem tych skryptów wprowadź w nich niezbędne zmiany.

1. Aby możliwe było tworzenie kopii lustrzanych, trzeba włączyć protokół TCP/IP. Użyj narzędzia SQL Server Configuration Manager, aby sprawdzić, czy ten protokół jest włączony.
2. Następnie na serwerach głównym, lustrzanym i świadka utwórz w wybranym miejscu nowy katalog przeznaczony na przykładowe dane.

UWAGA W skryptach z tego rozdziału <twój katalog> oznacza katalog utworzony w tym punkcie. Musisz zmienić to określenie na właściwą ścieżkę do tego katalogu.

3. Następnie na każdym serwerze utwórz certyfikat. Nawiąż połączenie z serwerem głównym i uruchom skrypt *CertSerwerGlowny.sql* przedstawiony na listingu 19.1.

Listing 19.1. Plik CertSerwerGlowny.sql

```
USE MASTER
GO
IF NOT EXISTS(SELECT 1 FROM sys.symmetric_keys where name =
'##MS_DatabaseMasterKey##')
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '23%&weq^yzYu3000!'
GO

IF NOT EXISTS (SELECT 1 FROM sys.databases where
[is_master_key_encrypted_by_server] = 1)
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY
GO

IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name = 'PrincipalServerCert')
CREATE CERTIFICATE PrincipalServerCert
WITH SUBJECT = 'Principal Server Certificate'
GO
```

Można zastosować uwierzytelnianie z systemu Windows lub certyfikaty. Tu używamy certyfikatów. Dla uproszczenia są to certyfikaty wygenerowane przez system SQL Server.

```
BACKUP CERTIFICATE PrincipalServerCert TO FILE = '<Twój katalog>
\PrincipalServerCert.cer'
```

Instrukcja `BACKUP CERTIFICATE` powoduje utworzenie kopii zapasowej certyfikatu dla danego klucza prywatnego.

4. Teraz utwórz punkt końcowy na serwerze głównym. Połącz się z serwerem głównym i wywołaj skrypt *PunktKoncowySerwerGlowny.sql*.

-- Sprawdzenie, czy istnieje punkt końcowy związany z kopiami lustrzanymi.

```
IF NOT EXISTS(SELECT * FROM sys.endpoints WHERE type = 4)
CREATE ENDPOINT DBMirrorEndPoint
STATE = STARTED AS TCP (LISTENER_PORT = port_num)
FOR DATABASE_MIRRORING ( AUTHENTICATION = CERTIFICATE PrincipalServerCert,
                        ENCRYPTION = REQUIRED
                        ,ROLE = ALL
                        )
```

Ten kod tworzy punkt końcowy DBMirrorEndPoint. Do uwierzytelniania używany jest certyfikat PrincipalServerCert. Ponadto ustawiono opcję ROLE=ALL, co oznacza, że dany serwer może działać jako serwer główny, lustrzany lub świadka. Jeśli chcesz, aby dany serwer pełnił wyłącznie funkcje świadka, zastosuj wartość WITNESS. Dostępna jest też wartość PARTNER, która oznacza, że dana jednostka może działać jako serwer główny lub lustrzany, ale już nie jako serwer świadka.

5. Teraz utwórz certyfikaty na serwerach lustrzanym i świadka. Połącz się z serwerem lustrzanym i wywołaj skrypt *CertSerwerLustrz.sql*.

```
USE MASTER
GO
IF NOT EXISTS (SELECT 1 FROM sys.symmetric_keys where name =
'##MS_DatabaseMasterKey##')
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '23%&weq^yzYu3000!'
GO

IF NOT EXISTS (select 1 from sys.databases where
[is_master_key_encrypted_by_server] = 1)
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY
GO

IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name = 'MirrorServerCert')
CREATE CERTIFICATE MirrorServerCert
WITH SUBJECT = 'Mirror Server Certificate'
GO

BACKUP CERTIFICATE MirrorServerCert TO FILE = '<Twój katalog>\MirrorServerCert
.cer'
```

6. Następnie, przy nawiązaniem połączeniu z serwerem lustrzanym, wywołaj skrypt *PunktKoncowySerwerLustrz.sql*.

-- Sprawdzenie, czy istnieje punkt końcowy związany z kopiami lustrzanymi.

```
IF NOT EXISTS (SELECT * FROM sys.endpoints WHERE type = 4)
CREATE ENDPOINT DBMirrorEndPoint
STATE=STARTED AS TCP (LISTENER_PORT = port_num)
FOR DATABASE_MIRRORING ( AUTHENTICATION = CERTIFICATE MirrorServerCert,
                        ENCRYPTION = REQUIRED
                        ,ROLE = ALL
                        )
```

7. Nawiąż połączenie z serwerem świadka i wywołaj skrypt *CertSerwerMonitor.sql*.

```
USE MASTER
GO
IF NOT EXISTS (SELECT 1 FROM sys.symmetric_keys where name =
'##MS_DatabaseMasterKey##')
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '23%&weq^yzYu3000!'
GO
```

```
IF NOT EXISTS (select 1 from sys.databases where
[is_master_key_encrypted_by_server] = 1)
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY
GO
```

```
IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name = 'WitnessServerCert')
CREATE CERTIFICATE WitnessServerCert
WITH SUBJECT = 'Witness Server Certificate'
GO
```

```
BACKUP CERTIFICATE WitnessServerCert
TO FILE = '<Twój katalog>\WitnessServerCert.cer'
```

8. Przy nawiązaniem połączeniu z serwerem świadka wywołaj skrypt *PunktKoncowySerwerSwiadka.sql*.

-- Sprawdzenie, czy istnieje punkt końcowy związany z kopiami lustrzanymi.

```
IF NOT EXISTS (SELECT * FROM sys.endpoints WHERE type = 4)
CREATE ENDPOINT DBMirrorEndPoint
STATE=STARTED AS TCP (LISTENER_PORT = port_num)
FOR DATABASE_MIRRORING
(AUTHENTICATION = CERTIFICATE WitnessServerCert,
ENCRYPTION = REQUIRED
,ROLE = ALL
)
```

Ponieważ wszyscy partnerzy mają komunikować się ze sobą, każdy z nich potrzebuje uprawnień do łączenia się z pozostałymi. Utwórz więc na każdym serwerze loginy i powiąż je z certyfikatami z obu pozostałych serwerów. Musisz też przyznać tym loginom uprawnienia CONNECT do punktu końcowego.

1. Najpierw skopiuj na dwa pozostałe serwery certyfikaty utworzone za pomocą przedstawionych wcześniej skryptów z instrukcją BACKUP CERTIFICATE. Przykładowo certyfikat *PrincipalServerCert.cer* z serwera głównego przenieś z utworzonego wcześniej katalogu do analogicznych katalogów na serwerach świadka i lustrzanym.
2. Nawiąż połączenie z serwerem głównym i wywołaj skrypt *LoginUprawSerwGlowny.sql*.

```
USE MASTER
GO
```

-- Umożliwia łączenie się serwera lustrzanego.

```
IF NOT EXISTS (SELECT 1 FROM sys.syslogins WHERE name = 'MirrorServerUser')
CREATE LOGIN MirrorServerUser WITH PASSWORD = '32sdgsgy^%$!'
IF NOT EXISTS (SELECT 1 FROM sys.sysusers WHERE name = 'MirrorServerUser')
CREATE USER MirrorServerUser;
```

```
IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name = 'MirrorDBCertPub')
CREATE CERTIFICATE MirrorDBCertPub AUTHORIZATION MirrorServerUser
FROM FILE = '<Twój katalog>\MirrorServerCert.cer'
```

```
GRANT CONNECT ON ENDPOINT::DBMirrorEndPoint TO MirrorServerUser
GO
```

-- Umożliwia łączenie się serwera świadka.

```
IF NOT EXISTS (SELECT 1 FROM sys.syslogins WHERE name = 'WitnessServerUser')
CREATE LOGIN WitnessServerUser WITH PASSWORD = '32sdgsgy^%$!'
IF NOT EXISTS (SELECT 1 FROM sys.sysusers WHERE name = 'WitnessServerUser')
CREATE USER WitnessServerUser;
```

```
IF NOT EXISTS (SELECT 1 FROM sys.certificates WHERE name = 'WitnessDBCertPub')
CREATE CERTIFICATE WitnessDBCertPub AUTHORIZATION WitnessServerUser
FROM FILE = '<Twój katalog>\WitnessServerCert.cer'
```

```
GRANT CONNECT ON ENDPOINT::DBMirrorEndPoint TO WitnessServerUser
GO
```

Skrypt ten tworzy na serwerze głównym dwóch użytkowników — `MirrorServerUser` i `WitnessServerUser`. Użytkownicy są wiązani z certyfikatami z serwerów lustrzanego i świadka. Następnie przyznawane są uprawnienia `CONNECT` do punktu końcowego. Dzięki temu serwery lustrzane i świadek mogą łączyć się z punktem końcowym z serwera głównego. Analogiczne kroki należy wykonać na serwerach lustrzanym i świadka.

3. Następnie nawiąż połączenie z serwerem lustrzanym i wywołaj skrypt *LoginUprawSerwLustrz.sql*. Na końcu połącz się z serwerem świadka i uruchom skrypt *LoginUprawSerwMonitor.sql*.

Skonfigurowałeś już punkty końcowe na wszystkich serwerach z wykorzystaniem certyfikatów. Jeśli chcesz zastosować uwierzytelnianie z systemu Windows, konfigurowanie punktów końcowych będzie łatwiejsze. Wystarczy na każdym serwerze wywołać instrukcje podobne do poniższych. Polecenia te należy wykonać na serwerze głównym, a kod dla pozostałych serwerów znajdziesz w plikach *LoginUprawSerwerLustrz.sql* i *LoginUprawSerwerMonitor.sql*.

```
IF NOT EXISTS (SELECT * FROM sys.endpoints WHERE type = 4)
CREATE ENDPOINT DBMirrorEndPoint
STATE = STARTED AS TCP (LISTENER_PORT = 5022)
FOR DATABASE_MIRRORING (AUTHENTICATION = WINDOWS, ROLE = ALL)
```

```
GRANT CONNECT ON ENDPOINT::DBMirrorEndPoint TO
[MyDomain\MirrorServerServiceAccount]
GO
```

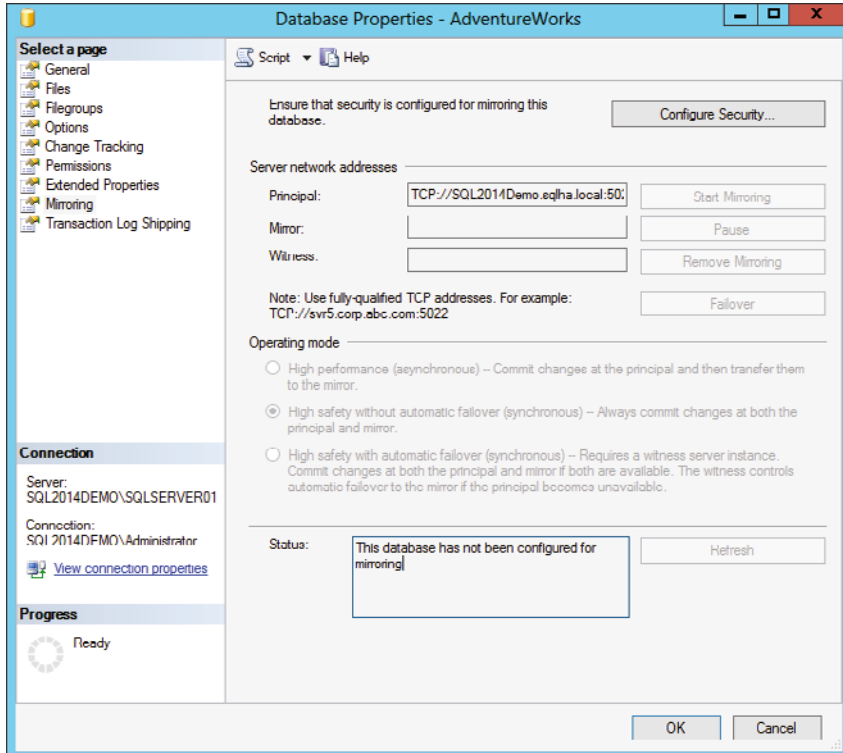
```
GRANT CONNECT ON ENDPOINT::DBMirrorEndPoint TO
[MyDomain\WitnessServerServiceAccount]
GO
```

Oczywiście musisz odpowiednio zmienić loginy. Przy uwierzytelnianiu z systemu Windows każdy serwer do nawiązywania połączeń z pozostałymi serwerami używa konta usługowego, w którym działa. Dlatego trzeba przyznać kontom usługowym systemu SQL Server 2014 uprawnienia `CONNECT` do punktów końcowych.

UWAGA Na serwerze potrzebny jest tylko jeden punkt końcowy związany z kopiami lustrzanymi (nawet wtedy, kiedy kopie tworzone są dla wielu baz). Koniecznie użyj portu, który nie jest zajmowany przez inne punkty końcowe. Wybierz dowolny dostępny port z przedziału od 1024 do 32767.

Nie zmieniaj (za pomocą instrukcji `ALTER ENDPOINT`) konfiguracji punktu końcowego używanego aktualnie do tworzenia kopii lustrzanych. Egzemplarze serwera korzystają ze swoich punktów końcowych do ustalenia stanu innych systemów. Po zmianie konfiguracji punktów końcowych może nastąpić ponowne uruchomienie systemu, co przez inne egzemplarze jest odbierane jako błąd. Jest to ważne zwłaszcza w trybie wysokiego bezpieczeństwa z automatycznym przełączaniem awaryjnym, ponieważ w tym podejściu zmiana konfiguracji punktu końcowego partnera może prowadzić do przełączenia awaryjnego.

Możesz też skonfigurować tworzenie kopii lustrzanych z uwierzytelnianiem z systemu Windows w programie SQL Server Management Studio. Kliknij prawym przyciskiem myszy bazę danych, dla której chcesz tworzyć kopie lustrzane, a następnie wybierz opcję *Tasks/Mirror*. Pojawi się okno widoczne na rysunku 19.4. Kliknij przycisk *Configure Security*. Uruchomiony zostanie kreator, który przeprowadzi Cię przez kroki potrzebne do skonfigurowania tworzenia kopii lustrzanych. Kreator próbuje użyć portu 5022 jako domyślnego portu TCP; jeśli chcesz, możesz ustawić inny port.



Rysunek 19.4. Kliknij przycisk *Configure Security*

Przygotowywanie bazy danych do tworzenia kopii lustrzanych

Zanim skonfigurujesz tworzenie kopii lustrzanych, musisz przygotować bazę danych.

1. W tym przykładzie użyj bazy AdventureWorks. Trzeba dla niej ustawić model odzyskiwania FULL:
 - Model odzyskiwania bazy danych należy ustawić na FULL,
 - aby możliwe było tworzenie kopii lustrzanych.

```
ALTER DATABASE AdventureWorks SET RECOVERY FULL
```
2. Nawiąż połączenie z serwerem głównym i uruchom skrypt *KopiaZapasowaBazy.sql*:
 - Tworzenie pełnej kopii zapasowej bazy danych.

```
BACKUP DATABASE [AdventureWorks] TO DISK = N'<Twój katalog>\AdventureWorks.bak'
```

```
WITH FORMAT, INIT,
NAME = N'AdventureWorks-Full Database Backup',STATS = 10
GO
```

Za pomocą tego skryptu utwórz pełną kopię zapasową bazy AdventureWorks.

3. Następnie przywróć bazę na serwerze lustrzanym. W instrukcji RESTORE DATABASE użyj opcji NORECOVERY. Nawiąż połączenie z serwerem lustrzanym i uruchom skrypt *PrzywracanieBazy.sql*. Ten skrypt wymaga umieszczenia kopii zapasowej bazy z serwera głównego w odpowiednim katalogu na serwerze lustrzanym.

- Jeśli ścieżka do bazy lustrzanej jest inna niż ścieżka do
- bazy głównej (np. używane są inne litery napędu),
- przy tworzeniu bazy lustrzanej trzeba w operacji RESTORE
- zastosować klauzulę MOVE. Omówienie tej klauzuli zawiera dokumentacja Books Online.

```
RESTORE DATABASE [AdventureWorks]
FROM DISK = ' <Twój katalog>\AdventureWorks.bak'
WITH NORECOVERY
,MOVE N'AdventureWorks_Data' TO N' <Twój katalog>\AdventureWorks_Data.mdf'
,MOVE N'AdventureWorks_Log' TO N' <Twój katalog>\AdventureWorks_Log.LDF'
```

Kod ten przywraca bazę AdventureWorks w trybie NORECOVERY na serwerze lustrzanym. Opcja NORECOVERY jest tu niezbędna. Baza jest już gotowa do tworzenia kopii lustrzanych. Teraz powinieneś dowiedzieć się, jak zapewnić początkową synchronizację między serwerami głównym i lustrzanym.

UWAGA Aby można było uruchomić sesję tworzenia kopii lustrzanych, nazwy baz na serwerach głównym i lustrzanym muszą być identyczne. Zanim utworzysz kopię zapasową głównej bazy danych, upewnij się, że używany jest dla niej model odzyskiwania FULL.

Początkowa synchronizacja serwerów głównego i lustrzanego

Wcześniej utworzyłeś kopię zapasową bazy AdventureWorks i przywróciłeś ją na potrzeby tworzenia kopii lustrzanych. Oczywiście tworzenie kopii lustrzanej bazy tak małej jak AdventureWorks nie odzwierciedla rzeczywistych sytuacji. W praktyce tworzone są kopie lustrzane baz o wielkości setek tysięcy megabajtów, dlatego (w zależności od wielkości bazy i odległości między serwerami) kopiowanie bazy i jej przywracanie na serwerze lustrzanym może zajmować dużo czasu.

W tym czasie w bazie głównej wygenerowanych może zostać wiele dzienników transakcji. Zanim włączysz tworzenie kopii lustrzanych, musisz skopiować i przywrócić wszystkie dzienniki transakcji na serwerze lustrzanym. Użyj przy tym opcji NORECOVERY. Jeśli nie chcesz kopiować dzienników transakcji i przywracać ich na serwerze lustrzanym, zatrzymaj wykonywanie kopii zapasowych dzienników transakcji (gdy używasz do tego zadania narzędzia SQL Server Agent, po prostu wyłącz to zadanie) do momentu przywrócenia bazy na serwerze lustrzanym i uruchomienia sesji tworzenia kopii lustrzanych (inicjowanie sesji jest opisane w punkcie „Uruchamianie sesji tworzenia kopii lustrzanych”). Po uruchomieniu takiej sesji możesz wznowić tworzenie kopii zapasowych dzienników transakcji na serwerze głównym. Należy zauważyć, że w tym podejściu plik dziennika transakcji rośnie, ponieważ nie są tworzone kopie zapasowe dziennika. Dlatego upewnij się, że na dysku jest wystarczająca ilość wolnego miejsca.

Przed uruchomieniem sesji tworzenia kopii zapasowych trzeba zsynchronizować obie bazy (główną i lustrzaną). Dlatego w pewnym momencie trzeba zatrzymać tworzenie kopii zapasowych dzienników transakcji na serwerze głównym. Określ, na jakim etapie chcesz to zrobić — przed utworzeniem pełnej kopii zapasowej bazy danych i jej przywróceniem na serwerze lustrzanym czy po wykonaniu takiej kopii. W drugim podejściu przed uruchomieniem sesji tworzenia kopii lustrzanych musisz skopiować i przywrócić wszystkie dzienniki transakcji na serwerze lustrzanym.

Gdy kopie lustrzane są tworzone dla dużej bazy, wykonywanie kopii zapasowych i ich przywracanie na serwerze lustrzanym trwa stosunkowo długo. Aby zminimalizować spadek wydajności serwera głównego, postaraj się zaplanować przygotowania do tworzenia kopii lustrzanych na okres niskiej aktywności systemu. W wysoce aktywnym systemie możesz wydłużyć czas między tworzeniem kopii zapasowych dzienników transakcji. Dzięki temu rzadziej trzeba będzie kopiować i przywracać dzienniki na serwerze lustrzanym, jednak zwiększa to ryzyko utraty danych w wyniku awarii serwera głównego. Jeśli aktywność bazy jest bardzo niska, możesz zatrzymać wykonywanie kopii zapasowych dzienników transakcji na serwerze głównym, utworzyć kopię zapasową bazy na tym serwerze, przywrócić bazę na serwerze lustrzanym, uruchomić sesję tworzenia kopii lustrzanych, a następnie wznowić tworzenie kopii zapasowych dzienników transakcji na serwerze głównym.

Uruchamianie sesji tworzenia kopii lustrzanych

Po przygotowaniu serwerów do tworzenia kopii lustrzanych bazy trzeba wykonać kilka zadań w bazie AdventureWorks. Dopiero potem można uruchomić sesję tworzenia kopii lustrzanych. Wykonaj zatem następujące kroki.

1. Nawiąż połączenie z serwerem głównym i wywołaj skrypt *ModyfikacjaDanych.sql*. Skrypt modyfikuje dane w tabeli `AdventureWorks.person.address`.
2. Nawiąż połączenie z serwerem głównym i uruchom skrypt *KopiaDzienSerwGlowny.sql*:

```
USE MASTER
GO
BACKUP LOG AdventureWorks TO DISK = '<Twój katalog>\AdventureWorks.trn'
```

To powoduje utworzenie kopii zapasowej dziennika transakcji bazy AdventureWorks. Na tym etapie bazy AdventureWorks z serwerów głównego i lustrzanego *nie są zsynchronizowane*. Zobacz, co się stanie, gdy spróbujesz uruchomić sesję tworzenia kopii lustrzanych z udziałem tych dwóch baz.

3. Nawiąż połączenie z serwerem lustrzanym i wywołaj skrypt *UstawSerwLustrzany.sql*. Ten kod zadziała prawidłowo.

```
USE MASTER
GO
ALTER DATABASE AdventureWorks
SET PARTNER = 'TCP://YourPrincipalServer:5022'
```

4. Teraz nawiąż połączenie z serwerem głównym i wywołaj skrypt *UstawSerwGlowny.sql*.

```
USE MASTER
GO
ALTER DATABASE AdventureWorks
SET PARTNER = 'TCP://YourMirrorServer:5023'
```

Możesz albo zapisywać numery portów używane przy tworzeniu kopii lustrzanych, albo wywołać zapytanie do widoku `sys.tcp_endpoints`, aby ustalić stosowane porty.

Próba wykonania tego skryptu zakończy się niepowodzeniem i wyświetleniem następującego komunikatu:

Msg 1412, Level 16, State 0, Line 3

The remote copy of database "AdventureWorks" has not been rolled forward to a point

in time that is encompassed in the local copy of the database log.

5. Jest to informacja, że baza na serwerze lustrzanym nie jest wystarczająco aktualna, aby można było rozpocząć sesję tworzenia kopii lustrzanych. Trzeba więc przywrócić na serwerze lustrzanym kopię zapasową dziennika transakcji utworzoną na serwerze głównym. Należy użyć trybu NORECOVERY, aby zsynchronizować serwery lustrzany i główny. Nawiąż połączenie z serwerem lustrzanym i wywołaj skrypt *PrzywracanieDzienSerwLustrz.sql*.

```
USE MASTER
GO
RESTORE LOG AdventureWorks
FROM DISK = '<Twój katalog>\AdventureWorks.trn'
WITH NORECOVERY
```

Skrypt ten wymaga skopiowania dziennika transakcji na dysk serwera lustrzanego. Jeśli umieściłeś dziennik transakcji w innym miejscu, podaj odpowiednią lokalizację, a następnie uruchom ten skrypt. Teraz serwery główny i lustrzany są zsynchronizowane.

6. Nawiąż połączenie z serwerem lustrzanym i ponownie wywołaj skrypt *UstawSerwLustrzany.sql*.
7. Następnie połącz się z serwerem głównym i wywołaj skrypt *UstawSerwGlowny.sql*. Teraz operacja powinna zakończyć się powodzeniem, ponieważ dziennik transakcji jest już przywrócony. W ten sposób sesja tworzenia kopii lustrzanych została uruchomiona.

Przy wykonywaniu skryptów *UstawSerwGlowny.sql* i *UstawSerwLustrzany.sql* możesz natrafić na następujące błędy.

```
Database mirroring connection error 4 'An error occurred
while receiving data: '64(The specified network name
is no longer available.)'.' for TCP://YourMirrorServer:5022'.
Error: 1443, Severity: 16, State: 2.
```

Możliwe, że zaporą na serwerze lustrzanym lub głównym blokuje połączenie z podanym portem. Użyj opcji *Ustawienia zaawansowane* w narzędziu Zapora systemu Windows, aby sprawdzić, czy numery portów używane przy tworzeniu kopii lustrzanych nie są blokowane.

Tryb wysokiego bezpieczeństwa bez automatycznego przełączania awaryjnego

Gdy uruchomisz sesję tworzenia kopii lustrzanych, bezpieczeństwo transakcji jest domyślnie ustawione na FULL. Dlatego sesja tworzenia kopii lustrzanych zawsze jest uruchamiana w trybie wysokiego bezpieczeństwa *bez* automatycznego przełączania awaryjnego. W tym trybie nie jest używany serwer świadka, zatem automatyczne przełączanie awaryjne jest niemożliwe. Ponieważ nie ma serwera świadka, serwer główny nie potrzebuje kworum do udostępniania bazy danych. Jeśli serwer główny utraci kworum z powodu awarii serwera lustrzanego, nadal będzie mógł udostępniać główną bazę danych i zapisywać w kolejce transakcje przeznaczone do przywrócenia na serwerze lustrzanym. Transakcje z tej kolejki są przesyłane na serwer lustrzany wtedy, gdy ten stanie się dostępny.

Z następnego punktu dowiesz się, jak zmienić ten tryb na model z automatycznym przełączaniem awaryjnym.

Tryb wysokiego bezpieczeństwa z automatycznym przełączaniem awaryjnym

Automatyczne przełączanie awaryjne oznacza, że gdy główna baza danych (lub zawierający ją serwer) przestanie działać, nastąpi przełączenie na serwer lustrzany, który przejmie rolę serwera głównego i zacznie udostępniać bazę danych. To podejście wymaga zastosowania trzeciej jednostki — serwera świadka. Serwer świadka jest używany przez serwer lustrzany przy ustalaniu, czy serwer główny rzeczywiście przestał działać. Jeśli spełniony jest warunek „dwa z trzech działają”, następuje automatyczne przełączenie awaryjne. Gdy zastosowany jest serwer świadka, użytkownik nie musi nic robić, aby nastąpiło automatyczne przełączenie na serwer lustrzany.

Serwer świadka

Jeśli wybrałeś opcję `SAFETY FULL`, możesz skonfigurować serwer świadka (patrz rysunek 19.1). Od dostępności tego serwera zależy, czy w trybie wysokiego bezpieczeństwa obsługiwane jest automatyczne przywracanie awaryjne po wystąpieniu problemów z główną bazą danych.

Automatyczne przełączanie awaryjne następuje, gdy:

- serwery świadka i lustrzany są połączone z serwerem głównym, kiedy ten ulegnie awarii (stanie się niedostępny),
- bezpieczeństwo transakcji jest ustawione na `FULL`,
- serwery są zsynchronizowane.

Aby w pełni wykorzystać możliwości automatycznego przełączania awaryjnego przy tworzeniu kopii lustrzanych, potrzebujesz odrębnego egzemplarza systemu SQL Server (niezależnego od serwerów głównego i lustrzanego). Ten sam serwer świadka może uczestniczyć w wielu równoległych sesjach tworzenia kopii lustrzanych.

Wróćmy do przykładu. Aby utworzyć serwer świadka, wykonaj następujące kroki.

1. Nawiąż połączenie z serwerem głównym i wykonaj skrypt `UstawSerwMonitor.sql`.

```
USE MASTER
GO
ALTER DATABASE AdventureWorks
SET WITNESS = 'TCP://YourWitnessServer:5024'
```

W celu wykonania tego skryptu trzeba połączyć się z serwerem głównym. W trakcie działania skryptu serwery główny i lustrzany muszą działać. Ten kod wystarczy do utworzenia serwera świadka obsługującego automatyczne przełączanie awaryjne. Na serwerze świadka może działać dowolna edycja systemu SQL Server — od Express po Enterprise. Na serwerze świadka można wykonywać różne operacje na bazach danych; jego rola nie ogranicza się do roli „świadka”.

UWAGA Serwer świadka przy tworzeniu kopii lustrzanych jest opcjonalny, przy czym jest niezbędny, jeśli chcesz stosować automatyczne przełączanie awaryjne.

Kworum

Gdy włączasz serwer świadka, wpływa on na *kworum* w sesji tworzenia kopii lustrzanych z automatycznym przełączaniem awaryjnym. Kworum wyznacza relację między dwoma połączonymi egzemplarzami (lub większą ich liczbą) w sesji tworzenia kopii lustrzanych i pomaga określić, który egzemplarz powinien pełnić funkcję serwera głównego.

W procesie tworzenia kopii lustrzanych kworum może działać w trzech trybach. Oto one.

- **Pełny** — obaj partnerzy i serwer świadka są połączeni ze sobą.
- **Partner z partnerem** — partnerzy są połączeni ze sobą, ale nie z serwerem świadka.
- **Serwer świadka z partnerem** — serwer świadka jest połączony z jednym z partnerów.

Baza w sesji tworzenia kopii zapasowych musi pracować w jednym z tych trybów, aby mogła działać jako główna baza danych.

Tryb wysokiej wydajności

Gdy uruchamiasz sesję tworzenia kopii lustrzanych, domyślnie używane jest ustawienie `SAFETY ON`. Aby więc aktywować tryb wysokiej wydajności, musisz zastosować ustawienie `SAFETY OFF`:

```
USE Master
ALTER DATABASE AdventureWorks SET PARTNER SAFETY OFF
```

W tym trybie tworzenie kopii lustrzanych ma tylko minimalny wpływ na szybkość przetwarzania transakcji i czas reakcji systemu. Przesyłanie dzienników na serwer lustrzany odbywa się tak samo jak w trybie wysokiego bezpieczeństwa, ponieważ jednak serwer główny nie musi czekać na zapisanie dziennika na dysku serwera lustrzanego, nieoczekiwana awaria serwera głównego może prowadzić do utraty danych.

Możesz ustawić dla serwera świadka tryb wysokiej wydajności, ponieważ jednak w tym trybie automatyczne przełączanie awaryjne jest niemożliwe, serwer świadka nie daje żadnych korzyści. Jeśli używasz trybu wysokiej wydajności, pominiń serwer świadka. Aby (w celu zastosowania trybu wysokiej wydajności) usunąć taki serwer, wywołaj przedstawione poniżej polecenie:

```
USE Master
ALTER DATABASE AdventureWorks SET WITNESS OFF
```

Jeśli zastosujesz serwer świadka w sesji w trybie wysokiej wydajności, ustalanie kworum będzie przebiegać tak.

- Gdy serwer lustrzany zakończy pracę, serwer główny musi być połączony z serwerem świadka. W przeciwnym razie serwer główny wyłączy dostęp do bazy do czasu ponownego podłączenia serwera świadka lub lustrzanego.
- Gdy serwer główny przestanie działać, wymuszenie przełączenia awaryjnego na serwer lustrzany wymaga, aby serwer lustrzany był podłączony do serwera świadka.

W tym modelu jedyny sposób na przełączenie awaryjne na serwer lustrzany to wywołanie na tym serwerze pokazanej poniżej instrukcji po odłączeniu serwera głównego od sesji. Jest to tzw. *wymuszone przełączenie awaryjne*. Opis różnych sposobów przełączania awaryjnego zamieszczono w tabeli 19.1.

```
USE MASTER
ALTER DATABASE AdventureWorks SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS
```

Wymuszone przełączenie awaryjne powoduje natychmiastowe odzyskanie bazy na serwerze lustrzanym, co może wiązać się z utratą danych. Ten tryb najlepiej stosować do przesyłania danych na duże odległości (np. do zdalnej lokalizacji używanej do odzyskiwania systemu po katastrofie) lub do tworzenia kopii lustrzanej aktywnej bazy danych, gdy akceptowalna jest utrata części danych. Tryb wysokiej wydajności można zastosować np. do tworzenia kopii lustrzanej hurtowni danych. Można też wykorzystać snapshot bazy danych (patrz podrozdział „Snapshoty baz danych”, dalej w tym rozdziale) na serwerze lustrzanym do generowania raportów na podstawie kopii lustrzanej hurtowni danych.

Kopie lustrzane i różne edycje systemu SQL Server 2014

Poznałeś już tryby pracy mechanizmu tworzenia kopii lustrzanych. Warto wiedzieć, które tryby są dostępne w poszczególnych edycjach systemu SQL Server. W tabeli 19.2 pokazano, które aspekty kopii lustrzanych są dostępne w różnych edycjach.

Tabela 19.2. Funkcje dostępne w poszczególnych edycjach systemu SQL Server 2014

Funkcje kopii lustrzanych	Edycja Enterprise	Edycja Developer	Edycja Standard	Edycja Business Intelligence	Edycja SQL Server Express
Partnerzy (serwer główny lub lustrzany)	✓	✓	✓	✓	
Serwer świadka	✓	✓	✓	✓	✓
SAFETY FULL	✓	✓	✓	✓	
SAFETY OFF	✓	✓			
Dostępność w trakcie wycofywania po przełączeniu awaryjnym	✓	✓	✓	✓	
Równoległe wprowadzanie rekordów	✓	✓			

Edycja SQL Server Express może być używana jedynie na serwerze świadka. Do stosowania niektórych innych funkcji, np. trybu wysokiej wydajności, niezbędne są edycje Enterprise lub Developer.

System SQL Server 2014 może używać wielu wątków, by przetwarzać kolejkę rekordów do wprowadzenia na serwerze lustrzanym. Jeśli serwer lustrzany ma mniej niż pięć procesorów, system SQL Server używa tylko jednego wątku. Równoległe wprowadzanie rekordów na serwerze lustrzanym jest zoptymalizowane tak, aby używany był jeden wątek na cztery procesory. Funkcja jest dostępna tylko w edycjach Enterprise i Developer.

Zmiana ról w mechanizmie tworzenia kopii lustrzanych

Może zauważyłeś, że jeśli spróbujesz wywołać zapytanie do bazy AdventureWorks z serwera lustrzanego, pojawi się następujący błąd:

Msg 954, Level 14, State 1, Line 1

The database "AdventureWorks" cannot be opened. It is acting as a mirror database.

Nie można uzyskać dostępu do bazy z serwera lustrzanego, jak więc przełączyć role partnerów? Są trzy sposoby przełączania awaryjnego systemu na serwer lustrzany:

- automatyczne przełączanie awaryjne,
- ręczne przełączanie awaryjne,
- wymuszone przełączanie awaryjne.

Sposób przełączania awaryjnego zależy od ustawienia bezpieczeństwa transakcji (FULL lub OFF), a także od dostępności serwera świadka.

Automatyczne przełączanie awaryjne

Automatyczne przełączanie awaryjne to funkcja działająca w trybie wysokiej dostępności (ustawienie SAFETY FULL i działający serwer świadka). Gdy na serwerze głównym zdarzy się awaria, inicjowane jest przełączanie awaryjne. Ponieważ tworzenie kopii lustrzanych działa w trybie wysokiej dostępności i używany jest serwer świadka, może nastąpić automatyczne przełączenie awaryjne. Oto zdarzenia zachodzące w tej sytuacji.

1. **Wystąpienie awarii.** Baza główna staje się niedostępna. Powodem może być awaria zasilania, usterka sprzętu, problemy z pamięcią lub inne.
2. **Wykrycie awarii.** Problemy są wykrywane przez serwery lustrzane i świadka. Obaj partnerzy i serwer świadka nieustannie wysyłają sobie sygnały kontrolne. Oczywiście nie są to zwykłe sygnały ping, tylko sygnały pozwalające ustalić dostępność systemu SQL Server, głównej bazy danych itd. Dla tych sygnałów określony jest limit czasu oczekiwania. Domyślnie w momencie konfigurowania sesji tworzenia kopii lustrzanych jest on ustawiany na 10 sekund. Możesz go zmienić za pomocą przedstawionej wcześniej instrukcji ALTER DATABASE.

Jeśli serwer główny nie odpowiada na sygnał kontrolny w określonym czasie, przyjmuje się, że serwer nie działa. Oznacza to wykrycie awarii. Powinieneś pozostawić domyślny limit czasu 10 sekund, a przynajmniej nie ustawiać go na niższą wartość, ponieważ przy dużym obciążeniu i tymczasowych problemach z siecią może nastąpić błędne wykrycie awarii oraz przełączanie awaryjne systemu z jednego serwera na drugi i z powrotem.
3. **Wykonanie wszystkich zaległych transakcji na serwerze lustrzanym.** Do tego momentu baza danych na serwerze lustrzanym działa w trybie przywracania i nieustannie odtwarza transakcje z dziennika. Po wykryciu awarii serwer lustrzany musi odzyskać bazę. Wymaga to wykonania zaległych transakcji z dziennika.
4. **Podjęcie decyzji o przełączeniu awaryjnym.** Na tym etapie serwer lustrzany kontaktuje się z serwerem świadka w celu uzyskania kworum i określa, czy bazę należy przełączyć awaryjnie na serwer lustrzany. W trybie wysokiego bezpieczeństwa z automatycznym przełączaniem awaryjnym wymaga to działającego serwera świadka. Podejmowanie decyzji trwa około sekundy, dlatego jeśli serwer główny wznowi pracę przed ukończeniem kroku 3., przełączanie awaryjne zostaje anulowane.
5. **Serwer lustrzany staje się serwerem głównym.** W czasie podejmowania decyzji o przełączeniu awaryjnym wykonywanie transakcji jest kontynuowane. Gdy serwery świadka i lustrzany utworzą kworum potrzebne do podjęcia tej decyzji, baza zostaje ostatecznie odzyskana. Serwer lustrzany przejmuje wtedy rolę serwera głównego i następuje odzyskanie bazy (proces ten obejmuje ustawienie odpowiedniego stanu bazy, anulowanie bieżących transakcji systemowych i rozpoczęcie wycofywania transakcji użytkowników). Baza staje się dostępna dla klientów i można wznowić jej standardowe działanie.

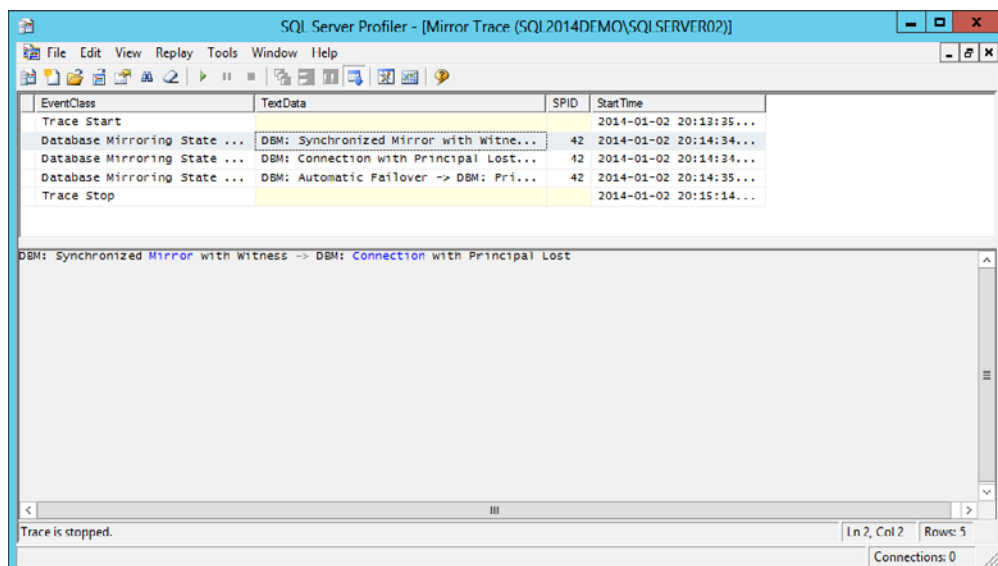
6. **Wycofywanie operacji.** Dostępne mogą być niezatwierdzone transakcje (transakcje przesłane na serwer lustrzany w czasie, gdy dawny serwer główny jeszcze działał, ale niezatwierdzone w dzienniku nowego serwera głównego); są one wycofywane.

Czas przełączania awaryjnego w tym trybie jest przeważnie krótki (zwykle jest to kilka sekund), jednak zależy to od etapu wykonywania zaległych transakcji. Jeśli serwer lustrzany wykonał wszystkie zaległe transakcje przed awarią serwera głównego, ten etap nie wprowadza opóźnień. Czas potrzebny na wykonanie zaległych transakcji zależy od długości kolejki i szybkości przetwarzania transakcji. Dopóki wartość kolumny `mirroring_state` wskazuje na brak synchronizacji, dopóty nie można wykonać przełączenia awaryjnego. Dostępne są liczniki wydajności, które pozwalają oszacować czas potrzebny na przetworzenie rekordów z kolejki zaległych transakcji na serwerze lustrzanym.

Aby zmierzyć rzeczywisty czas przełączania, możesz użyć programu SQL Server Profiler do śledzenia zdarzeń. Po przeczytaniu rozdziału 13., „Dostrajanie wydajności kodu w języku T-SQL”, dowiesz się więcej o analizowaniu śladów w tym programie. Tu możesz wykorzystać go do pomiaru długości trwania przełączania awaryjnego.

1. Uruchom program SQL Server Profiler.
2. Nawiąż połączenie z serwerem lustrzanym.
3. Wybierz zdarzenie *Database Mirroring State Change* z grupy *Database*.
4. Wybierz kolumny `TextData` i `StartTime`.
5. Uruchom śledzenie.
6. Zatrzymaj usługę systemu SQL Server na serwerze głównym. Po krótkim czasie nastąpi automatyczne przełączenie.

Ciekawe są dwie kolumny ze śladu. `TextData` udostępnia opis zdarzenia zmiany stanu w mechanizmie kopii lustrzanych. `StartTime` określa czas zajścia tego zdarzenia. Na rysunku 19.5 przedstawiono ślad tych zdarzeń w programie SQL Server Profiler.



Rysunek 19.5. Pomiar czasu trwania przełączania awaryjnego

Zdarzenie *Synchronized Mirror with Witness* polega na tym, że serwer lustrzany informuje serwer świadka o utracie połączenia z serwerem głównym (krok 4. na rysunku 19.2). Wtedy następuje przełączenie bazy na serwer lustrzany, który przejmuje rolę serwera głównego. To oznacza, że serwer główny działa bez partnera. Jeśli nastąpi awaria, baza stanie się niedostępna. Przy przełączaniu w dzienniku błędów systemu SQL Server pojawia się komunikat podobny do poniższego:

```
The mirrored database "AdventureWorks" is changing roles from "MIRROR" to "PRINCIPAL" due to Auto Failover.
```

Z kolumny `StartTime` można się dowiedzieć, że automatyczne przełączanie awaryjne zajęło około 7 sekund.

Czas trwania przełączania awaryjnego zależy od rodzaju awarii i obciążenia bazy danych. Przy wysokim obciążeniu proces ten trwa dłużej niż przy braku aktywności. Po przełączeniu w dzienniku błędów systemu SQL Server znajdziesz komunikat podobny do poniższego:

```
The mirrored database "AdventureWorks" is changing roles from "MIRROR" to "PRINCIPAL" due to Failover from partner.
Starting up database 'AdventureWorks'
Analysis of database 'AdventureWorks' (9) is 81% complete (approximately 0 seconds remain).
Analysis of database 'AdventureWorks' (9) is 100% complete (approximately 0 seconds remain).
Recovery of database 'AdventureWorks' (9) is 0% complete (approximately 30 seconds remain). Phase 2 of 3.
Recovery of database 'AdventureWorks' (9) is 16% complete (approximately 17 seconds remain). Phase 2 of 3.
13 transactions rolled forward in database 'AdventureWorks' (9).
Recovery of database 'AdventureWorks' (9) is 16% complete (approximately 17 seconds remain). Phase 3 of 3.
Recovery of database 'AdventureWorks' (9) is 100% complete (approximately 0 seconds remain). Phase 3 of 3.
```

Dodatkowe kroki wykonywane w czasie analiz i odzyskiwania bazy danych powodują, że ręczne przełączanie awaryjne trwa dłużej.

Po przełączeniu awaryjnym trzeba przekierować klienty na nowy serwer główny. Opisano to w podrozdziale „Przygotowanie serwera lustrzanego do przełączenia awaryjnego”, dalej w tym rozdziale. Dowiesz się tam także, jakie inne zadania musisz wykonać na serwerze lustrzanym, aby przygotować go do przełączenia awaryjnego i przejęcia obsługi bazy danych.

Ręczne przełączanie awaryjne

Przy ręcznym przełączaniu awaryjnym decydujesz się zmienić role partnerów. Serwer lustrzany staje się głównym, a główny przyjmuje rolę lustrzanego. W celu ręcznego przełączenia awaryjnego wywołaj następujące polecenie:

```
ALTER DATABASE AdventureWorks SET PARTNER FAILOVER
```

UWAGA Przy ręcznym przełączaniu awaryjnym należy zastosować opcję `SAFETY FULL`. Nie ma wtedy znaczenia, czy serwer świadka jest skonfigurowany.

Aby wykonać przełączenie awaryjne, należy wywołać to polecenie na serwerze głównym. Ponadto wartość kolumny `mirroring_state` musi wskazywać na to, że serwery są zsynchronizowane.

W przeciwnym razie przy próbie wywołania na serwerze głównym polecenia przełączenia pojawi się następujący komunikat:

Msg 1422, Level 16, State 2, Line 1

The mirror server instance is not caught up to the recent changes to database "AdventureWorks". Unable to fail over.

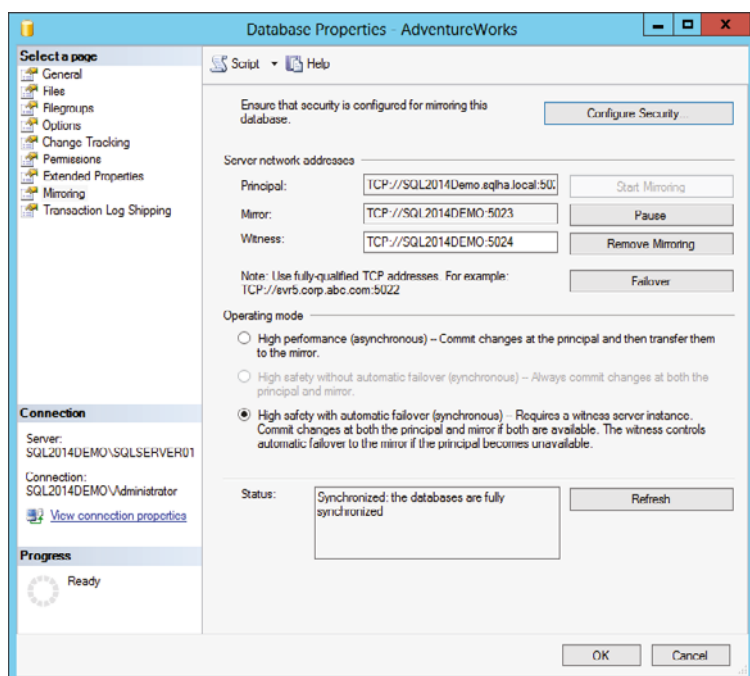
Teraz możesz ręcznie uruchomić przełączanie awaryjne serwerów używanych przy tworzeniu kopii lustrzanych.

1. Otwórz skrypt *KopieLustrzanePolecenia.sql*. Gdy wcześniej przeprowadzałeś automatyczne przełączanie awaryjne, zatrzymałeś usługę systemu SQL Server na serwerze głównym. Przed ręcznym przełączeniem awaryjnym ponownie uruchom tę usługę, ponieważ tym razem oba serwery muszą być aktywne.
2. Aby zobaczyć, co dzieje się na zapleczu, uruchom program SQL Server Profiler, nawiąż połączenie z serwerem lustrzanym i wybierz zdarzenie *Database Mirroring State Change* z grupy *Database*.
3. Następnie uruchom śledzenie. Uwzględnij dwie kolumny — *TextData* i *StartTime*.
4. Aby obserwować zdarzenia z serwera głównego, uruchom następny egzemplarz programu SQL Server Profiler i nawiąż połączenie z tym serwerem.
5. Teraz połącz się z serwerem głównym i wywołaj polecenie:

```
ALTER DATABASE AdventureWorks SET PARTNER FAILOVER
```

Właśnie ręcznie wymusiłeś przełączenie awaryjne z serwera głównego na lustrzany. Aby ręcznie wykonać przełączenie awaryjne, możesz też wykorzystać program SQL Server Management Studio.

1. Kliknij główną bazę danych prawym przyciskiem myszy i wybierz opcję *Tasks/Mirror*. Pojawi się okno dialogowe widoczne na rysunku 19.6.



Rysunek 19.6. Tu możesz ręcznie przełączyć serwery

2. Kliknij przycisk *Failover*. Zobaczysz okno, w którym możesz zatwierdzić przełączenie serwerów.
3. Kliknij przycisk *OK*.

Ręczne przełączanie awaryjne stosuje się przy planowanych przestojach, migracjach i aktualizacjach, co opisujemy w podrozdziale „Przygotowywanie serwera lustrzanego do przełączenia awaryjnego”, dalej w tym rozdziale.

Wymuszone przełączenie awaryjne

Do wymuszania przełączenia awaryjnego służy poniższa instrukcja; należy ją wywołać na serwerze lustrzanym:

```
ALTER DATABASE AdventureWorks SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS
```

OSTRZEŻENIE Zachowaj ostrożność przy stosowaniu tej instrukcji, ponieważ może ona prowadzić do utraty danych.

W momencie uruchomienia tego polecenia serwer lustrzany nie może mieć połączenia z serwerem głównym. W przeciwnym razie przełączenie awaryjne będzie niemożliwe. Jeśli serwer główny działa i serwer lustrzany może z nim nawiązać połączenie, po wywołaniu przedstawionej instrukcji pojawi się następujący komunikat o błędzie:

```
Msg 1455, Level 16, State 2, Line 1
```

```
The database mirroring service cannot be forced for database "AdventureWorks" because the database is not in the correct state to become the principal database.
```

Teraz spróbuj wykonać to ćwiczenie z wykorzystaniem serwerów.

1. Ponieważ wcześniej skonfigurowałeś tworzenie kopii lustrzanych w trybie pełnego bezpieczeństwa z automatycznym przełączaniem awaryjnym, musisz to zmienić. Otwórz skrypt *KopieLustrzanePolecenia.sql* i wywołaj poniższą instrukcję na serwerach głównym lub lustrzanym:

```
ALTER DATABASE AdventureWorks SET WITNESS OFF
```

2. Następnie uruchom poniższe polecenie na serwerze głównym:

```
ALTER DATABASE AdventureWorks SET SAFETY OFF
```

3. Teraz baza AdventureWorks działa z ustawieniem SAFETY OFF i bez serwera świadka, dlatego można wymusić przełączenie awaryjne. Trzeba zasymulować sytuację, w której serwer lustrzany nie może uzyskać quorum z serwerem głównym (czyli nie może się z nim połączyć). Aby to zrobić, zatrzymaj usługę systemu SQL Server na serwerze głównym, a następnie wywołaj na serwerze lustrzanym poniższe polecenie:

```
ALTER DATABASE AdventureWorks SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS
```

To polecenie powoduje odzyskanie bazy AdventureWorks i udostępnienie jej na serwerze lustrzanym. Wartość kolumny `mirroring_state` (z widoku `sys.database_mirroring`) nie ma tu znaczenia, ponieważ jest to wymuszone przełączenie awaryjne, które może prowadzić do utraty danych.

Zauważ, że gdy ponownie uruchomisz pierwotny serwer główny (na którym zatrzymałeś usługę systemu SQL Server), sesja tworzenia kopii lustrzanych zostanie zawieszona. Możesz nawiązać połączenie z serwerem głównym lub lustrzanym, aby sprawdzić zawartość widoku `sys.database_`

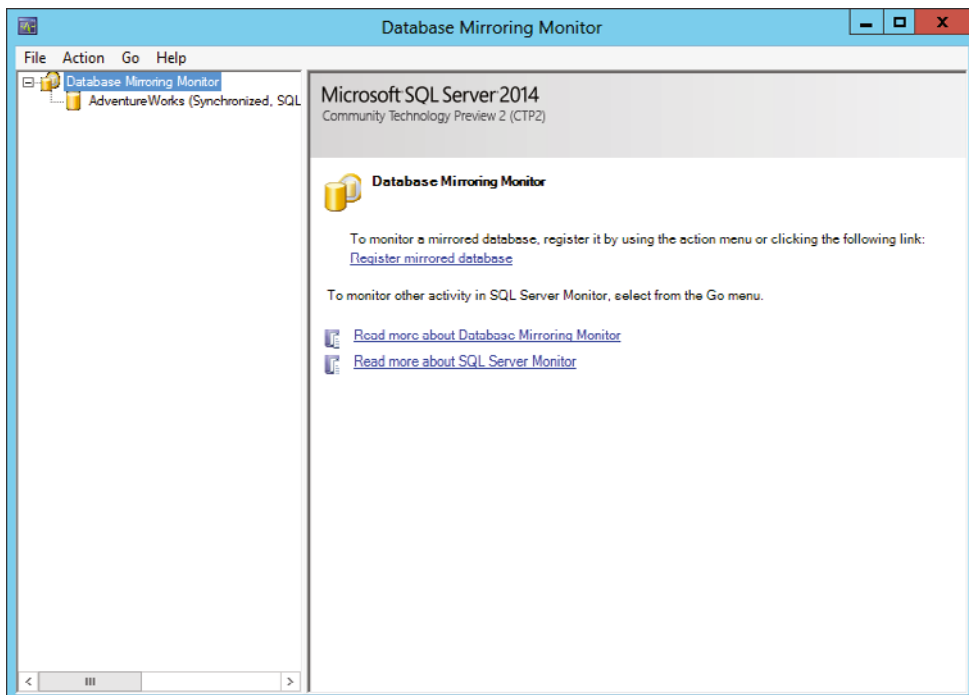
→mirroring za pomocą skryptu *KopieLustrzaneWidok.sql* i wyświetlić stan mechanizmu tworzenia kopii lustrzanych. Aby wznowić sesję, wywołaj poniższe polecenie ze skryptu *KopieLustrzaneWidok.sql* (możesz to zrobić na serwerze głównym lub lustrzanym):

```
ALTER DATABASE AdventureWorks SET PARTNER RESUME
```

Używanie monitora tworzenia kopii lustrzanych

Monitor tworzenia kopii lustrzanych (narzędzie Database Mirroring Monitor) z systemu SQL Server 2014 śledzi przebieg procesu omawianego w tym rozdziale. Ułatwia to pracę administratorom baz danych. Aby uzyskać dostęp do tego narzędzia, w programie SQL Server Management Studio kliknij prawym przyciskiem myszy w oknie *Object Explorer* dowolną bazę użytkownika z systemu SQL Server 2014 i wybierz opcję *Tasks/Launch Database Mirroring Monitor*. Za pomocą tego narzędzia możesz śledzić wszystkie sesje tworzenia kopii lustrzanych.

Bazę danych, dla której tworzone są kopie lustrzane, trzeba zarejestrować w monitorze. W tym celu wybierz opcję *Action/Register Mirrored Database*, a następnie wykonaj instrukcje z kreatora. W kreatorze możesz podać nazwę serwera głównego lub lustrzanego, a narzędzie samo wykryje drugiego partnera. Na rysunku 19.7 przedstawiono monitor z zarejestrowaną bazą danych.



Rysunek 19.7. Baza danych zarejestrowana w monitorze tworzenia kopii lustrzanych

Za pomocą tego interfejsu możesz śledzić najważniejsze liczniki wymienione w poprzednim punkcie. Możesz też ustawić alerty, aby narzędzie wysyłało powiadomienia lub wykonywało odpowiednie operacje po osiągnięciu przez licznik określonych wartości progowych. To zagadnienie omawiamy dalej.

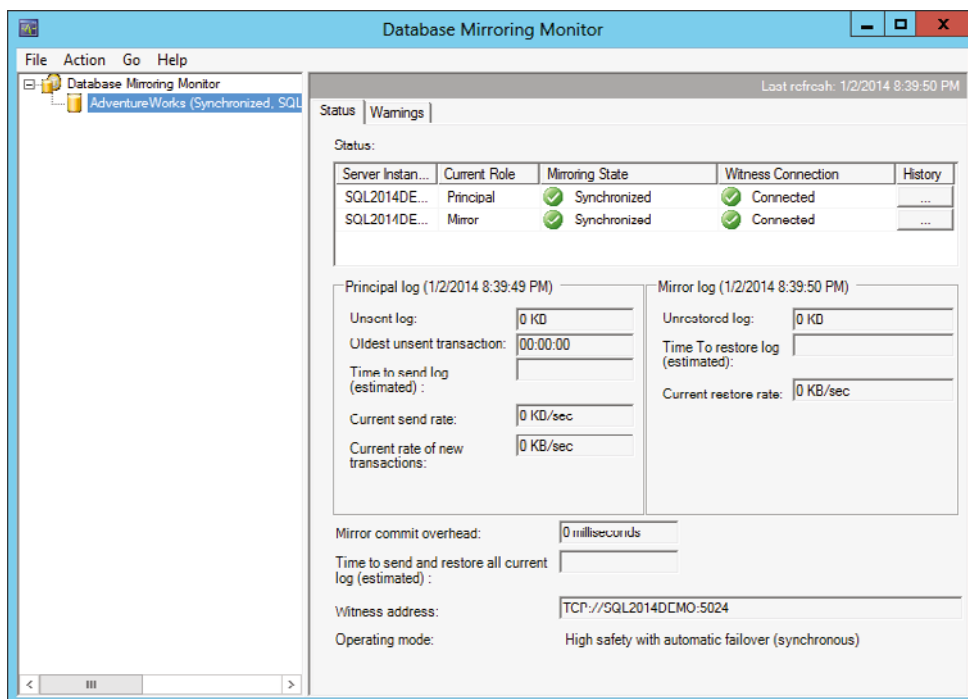
Gdy konfigurujesz tworzenie kopii lustrzanych w programie SQL Server Management Studio, generowane jest zadanie *Database Mirroring Monitor* narzędzia SQL Server Agent. Domyślnie jest ono uruchamiane co minutę i odświeża odpowiednie liczniki. Uzyskane dane trafiają do tabeli `msdb.dbo.dbm_monitor_data`. Jeśli chcesz, możesz zmienić harmonogram wykonywania tego zadania. Gdy konfigurujesz tworzenie kopii lustrzanych za pomocą przedstawionych wcześniej skryptów, możesz utworzyć odświeżające liczniki zadanie przy użyciu następującego polecenia:

```
sp_dbmmonitoraddmonitoring [ update_period ]
```

Wartość parametru `[update_period]` to domyślnie minuta. Możesz ustawić tu wartość z przedziału od 1 do 120 (oznacza ona czas w minutach). Jeśli nie utworzysz opisanego zadania, wciśnij klawisz *F5* w narzędziu Database Mirroring Monitor. Spowoduje to wywołanie procedury składowanej `sp_dbmmonitorresults`, która odświeża dane (dodaje wiersz z nowymi odczytami) w tabeli `msdb.dbo.dbm_monitor_data`. Ta procedura wywołuje inną procedurę składowaną z bazy `msdb`, `sp_dbmmonitorupdate`, która aktualizuje tabelę ze stanem i oblicza wskaźniki wydajności wyświetlane w omawianym narzędziu. Jeśli wciśniesz klawisz *F5* więcej niż raz w przeciągu 15 sekund, dane w tabeli nie zostaną odświeżone.

Przyjrzyj się informacjom z zakładki *Status* widocznym na rysunku 19.8. Dane z tego obszaru (nazwy serwerów, ich role, stan tworzenia kopii lustrzanych i stan połączenia serwera świadka) pochodzą z widoku `sys.database_mirroring`. Jeśli klikniesz przycisk *History*, pojawią się dane historyczne o stanie tworzenia kopii lustrzanych i inne wskaźniki wydajności. Te dane historyczne domyślnie są przechowywane w tabeli `msdb.dbo.dbm_monitor_data` przez 7 dni (168 godzin). Jeśli chcesz zmienić ten czas, użyj procedury składowanej `sp_dbmmonitorchangealert`:

```
EXEC sp_dbmmonitorchangealert AdventureWorks, 5, 8, 1
```



Rysunek 19.8. Informacje o stanie mechanizmu tworzenia kopii lustrzanych

W tym przykładzie czas przechowywania danych dla bazy AdventureWorks jest ustawiany na 8 godzin, 5 to wartość parametru `alter_id`, a 1 to wartość parametru `enabled`. Opis tej procedury składowanej znajdziesz w dokumentacji Books Online systemu SQL Server 2014.

Oto szczegółowe omówienie różnych liczników dostępnych w zakładce *Status*.

➤ **Principal log**

- **Unsent log.** Jest to wartość licznika *Log Send Queue KB* z serwera głównego. W polu *Unsent log* pojawia się ostatnia wczytana wartość; jeśli chcesz ją porównać z danymi z monitora wydajności, zwróć w nim uwagę na ostatni odczyt. Możesz uruchomić skrypt *ModyfikacjaDanych.sql*. Jednak wcześniej wstrzymaj sesję tworzenia kopii zapasowych. W tym celu wywołaj następujące polecenie:

```
ALTER DATABASE AdventureWorks SET PARTNER SUSPEND
```

Zobaczysz, że wartość w tym polu zacznie rosnąć wraz z akumulowaniem się niewysłanych rekordów dziennika.

- **Oldest unsent transaction.** Pole wyświetla wiek (w formacie gg:mm:ss) najstarszej niewysłanej transakcji z kolejki. Pozwala to określić, jak bardzo serwer lustrzany jest opóźniony względem głównego.
- **Time to send log (estimated).** Pole określa szacunkowy czas potrzebny serwerowi głównemu na wysłanie dziennika z kolejki do serwera lustrzanego. Ponieważ częstotliwość wykonywanych transakcji się zmienia, w tym polu pojawia się tylko szacunkowa wartość. Ten licznik określa też przybliżony czas potrzebny do ręcznego przełączenia serwerów. Jeśli wstrzymasz tworzenie kopii lustrzanych, w tym polu pojawi się wartość *Infinite*. Oznacza ona, że ponieważ na serwer lustrzany nie są wysyłane żadne dzienniki transakcji, ten serwer nigdy nie wykona zaległych transakcji.
- **Current send rate.** Licznik określa, z jaką szybkością (w kilobajtach na sekundę) dziennik transakcji jest przesyłany na serwer lustrzany. Odpowiada on licznikowi *Log Bytes Sent/Sec* w monitorze wydajności. Różnica jest taka, że tu wartość jest podawana w kilobajtach, a w monitorze wydajności — w bajtach. Gdy w polu *Time to Send Log* wyświetlana jest wartość *Infinite*, ten licznik pokazuje 0 kilobajtów na sekundę, ponieważ na serwer lustrzany nie są wysyłane żadne dane.
- **Current rate of new transaction.** Pole określa liczbę nowych transakcji na sekundę. Odpowiada ono licznikowi *Transaction/sec* obiektu *Database* w monitorze wydajności.

➤ **Mirror Log**

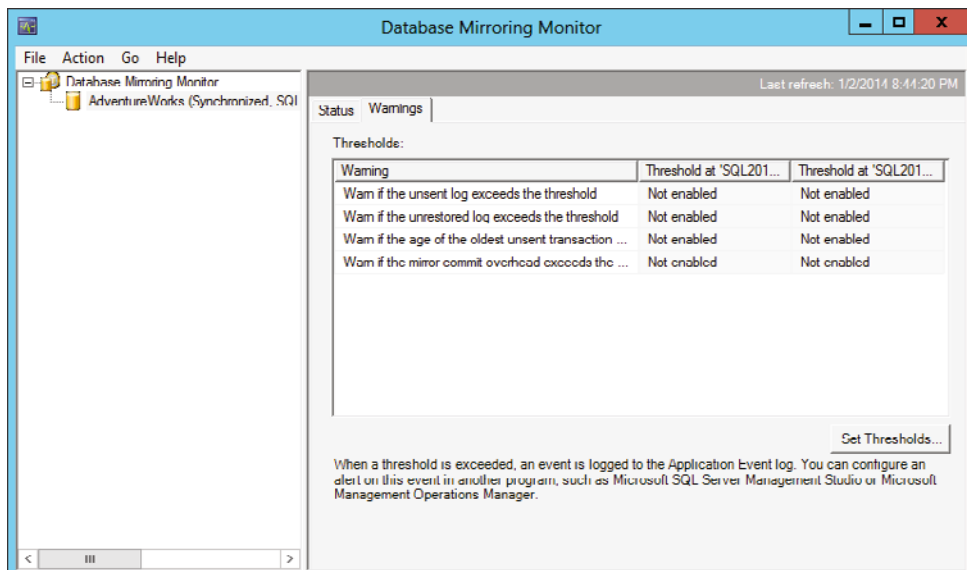
- **Unrestored log.** Licznik dotyczy serwera lustrzanego. Określa, ile kilobajtów danych znajduje się w kolejce rekordów do wprowadzenia. Odpowiada on licznikowi *Redo Queue KB* z serwera lustrzanego. Jeśli licznik *Unrestored log* ma wartość 0, serwer lustrzany wykonał już wszystkie transakcje z serwera głównego i — jeśli będzie trzeba — może natychmiast przejąć jego zadania.
- **Time to restore log.** Licznik podaje szacunkowy czas (w minutach) potrzebny serwerowi lustrzanemu na odtworzenie transakcji oczekujących w kolejce rekordów do wprowadzenia (to zagadnienie zostało już opisane wcześniej). Jest to jednocześnie szacunkowy czas, po jakim można przełączyć się na serwer lustrzany.
- **Current restore rate.** Jest to szybkość wprowadzania rekordów z dziennika transakcji w lustrzanej bazie danych, mierzona w kilobajtach na sekundę.
- **Mirror committed overhead.** Licznik mierzy opóźnienie (w milisekundach) związane z oczekiwaniem na potwierdzenie z serwera lustrzanego. Jest to odpowiednik licznika

Transaction Delay na serwerze głównym. Licznik *Mirror committed overhead* ma znaczenie tylko w trybie wysokiego bezpieczeństwa. W trybie wysokiej wydajności wartość licznika to 0, ponieważ serwer główny zapisuje dziennik transakcji na dysku bez oczekiwania na serwer lustrzany.

- **Time to send and restore all current log (estimated).** Licznik wyświetla czas potrzebny na przesłanie i wprowadzenie wszystkich zatwierdzonych rekordów z aktualnego dziennika serwera głównego. Ten czas może być krótszy niż suma wartości liczników *Time to send log (estimated)* i *Time to send log (estimated)*, ponieważ wysyłanie i wprowadzanie rekordów może się odbywać równolegle. Ta szacunkowa wartość określa czas potrzebny na wysłanie i wprowadzenie nowych transakcji zatwierdzonych na serwerze głównym z uwzględnieniem zaległych danych z kolejki rekordów do wysłania.
- **Witness address.** Jest to pełna nazwa domenowa serwera świadka, włącznie z numerem portu przypisanym do punktu końcowego.
- **Operating mode.** Określa tryb działania sesji tworzenia kopii lustrzanych. Przyjmuje jedną z następujących wartości:
 - *High performance (asynchronous),*
 - *High safety without automatic failover (synchronous),*
 - *High safety with automatic failover (synchronous).*

Ustawianie wartości progowych liczników i wysyłanie alertów

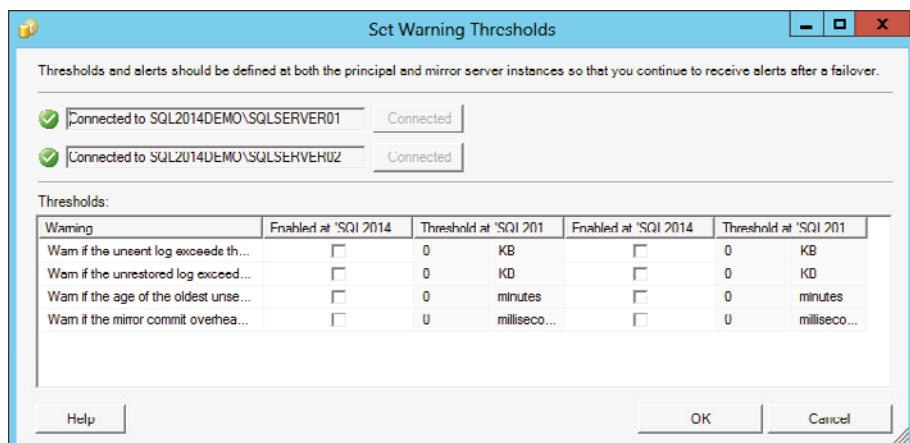
Możesz ustawić ostrzegawcze wartości progowe związane z tworzeniem kopii lustrzanych, aby otrzymywać alerty po ich przekroczeniu. Na rysunku 19.9 przedstawiono zakładkę *Warnings* narzędzia Database Mirroring Monitor.



Rysunek 19.9. Okno do ustawiania alertów

Poniżej przedstawione są instrukcje ustawiania wartości progowych dla różnych ostrzeżeń.

1. Najpierw kliknij przycisk *Set Thresholds* w zakładce *Warnings*. Pojawi się okno dialogowe widoczne na rysunku 19.10. Tu możesz ustawić wartości progowe liczników z serwerów głównego i lustrzanego. Wartości mogą być dla obu serwerów identyczne lub różne.



Rysunek 19.10. W tym miejscu możesz ustawić poziomy progowe liczników

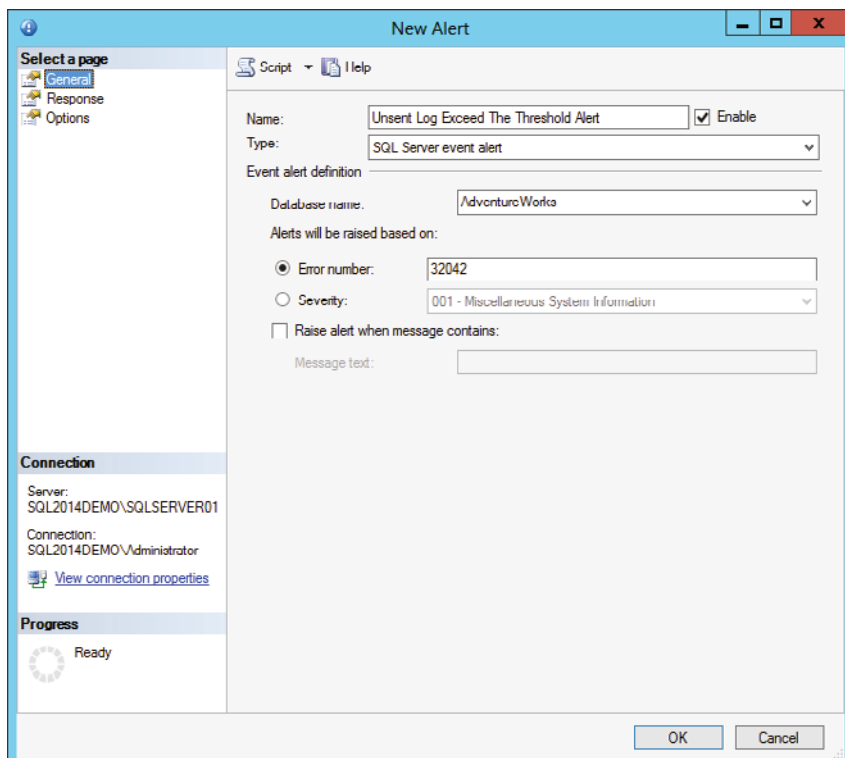
2. W tym przykładzie zaznacz pole wyboru przy pierwszym ostrzeżeniu, *Warn if the unent log exceeds the threshold*, i ustaw wartość progową na 100 kilobajtów, ale tylko dla serwera głównego.
3. Następnie w katalogu *Alert* narzędzia SQL Server Agent w programie SQL Server Management Studio dodaj nowy alert. Pojawi się okno dialogowe podobne do tego z rysunku 19.11. Wpisz nazwę alertu i ustaw bazę danych, której ma dotyczyć. Tu jest to baza *AdventureWorks*.
4. System SQL Server zgłasza błąd po wykryciu przekroczenia ustawionej wartości progowej. Jako numer błędu podaj **32042**. W punkcie „Using Warning Thresholds and Alerts on Mirroring Performance Metrics” dokumentacji Books Online poświęconej systemowi SQL Server znajdziesz numery błędów powiązane z innymi zdarzeniami.
5. Następnie kliknij zakładkę *Response* w lewym panelu i podaj informacje o operatorze, który ma otrzymać alert po przekroczeniu danej wartości progowej.

Aby przetestować ten alert, na serwerze głównym wstrzymaj za pomocą poniższej instrukcji tworzenie kopii lustrzanych bazy:

```
ALTER DATABASE AdventureWorks SET PARTNER SUSPEND
```

Diennik przestanie być wysyłany na serwer lustrzany, a wartość licznika *Unsent Log* zacznie rosnąć. Gdy dotrze do poziomu 100 kilobajtów, powinien pojawić się alert.

UWAGA Tworzenie kopii lustrzanych baz możesz monitorować w programie SQL Server Profiler. Użyj do tego zdarzenia *Database Mirroring State Change* z kategorii *Database*. To zdarzenie rejestruje wszystkie zachodzące na serwerze zmiany stanu mechanizmu tworzenia kopii lustrzanych (stan informuje o wstrzymaniu tworzenia kopii, synchronizacji w toku, zakończeniu synchronizacji itd.).



Rysunek 19.11. Konfigurowanie alertu

Przygotowywanie serwera lustrzanego do przełączenia awaryjnego

Kopie lustrzane służą do tego, aby po przełączeniu awaryjnym serwer lustrzany mógł przejąć całe obciążenie bazy. Jeśli serwer lustrzany ma w pełni przejąć funkcje serwera głównego, musisz wprowadzić odpowiednie zmiany w konfiguracji. Tworzenie kopii lustrzanych baz danych pozwala przełączać awaryjnie same bazy — nie jest tworzona kopia lustrzana całego egzemplarza systemu SQL Server. Jeśli chcesz umożliwić przełączanie awaryjne całych egzemplarzy, pomyśl o zastosowaniu klastrów WSFC, co opisano w rozdziale 16., „Service Broker w systemie SQL Server 2014”.

W trakcie przygotowywania serwera lustrzanego do przełączania awaryjnego należy zacząć od sprzętu i oprogramowania.

Sprzęt, oprogramowanie i konfiguracja serwera

Jeśli serwer lustrzany ma obsługiwać to samo obciążenie, co serwer główny, oba serwery powinny mieć identyczny sprzęt (procesory, pamięć RAM, dyski i komponenty sieciowe). Jeżeli jednak serwer główny to 32-rdzeniowy 64-bitowy komputer z 256 gigabajtami pamięci RAM, używanie identycznego serwera lustrzanego jest kosztowne, gdy awarie serwera głównego są

rzadkie i serwer lustrzany zwykle nie jest wykorzystywany. Dla mniej istotnych aplikacji można zastosować wolniejszy serwer, używany tylko tymczasowo po przełączeniu awaryjnym. Jednak dla mało ważnych aplikacji zwykle także serwer główny nie musi być bardzo wydajny.

Ponadto tak wydajny sprzęt jest używany przy dużym obciążeniu, dlatego po przełączeniu awaryjnym serwer lustrzany musi sobie poradzić z przetwarzaniem choćby przez krótki czas. Uwzględnij też koszty przestoju firmy w porównaniu do kosztów sprzętu, a następnie podejmij decyzję. Serwer lustrzany możesz też wykorzystywać do wykonywania dodatkowych zadań. Dzięki temu nawet w roli serwera lustrzanego sprzęt będzie przydatny. Oczywiście musisz wszystko odpowiednio zaplanować, aby właściwie wykorzystać serwer lustrzany. Jeśli obciążenie z serwera bazodanowego można obsłużyć za pomocą serwerów wirtualnych, do tworzenia kopii lustrzanych można zastosować także takie serwery.

Upewnij się, że na obu serwerach używane są te same wersje systemu operacyjnego z identycznymi pakietami SP i poprawkami. W trakcie aktualizowania aktywnego systemu (co opisano w punkcie „Dostępność bazy danych w trakcie planowanego przestoju”, dalej w tym rozdziale) pakiety SP i poprawki mogą tymczasowo różnić się od siebie.

Na serwerach obu partnerów używana musi być ta sama edycja systemu SQL Server. Serwer świadka może mieć zainstalowaną inną edycję. Serwer świadka może być mniej wydajny, ponieważ nie obsługuje obciążenia, a jest używany jedynie do wyznaczania quorum. Przy automatycznym przełączeniu awaryjnym serwer świadka koniecznie musi być dostępny. W tabeli 19.2 z punktu „Kopie lustrzane i różne edycje systemu SQL Server 2014” znajdziesz szczegółowe informacje o tym, które funkcje kopii lustrzanych są dostępne w poszczególnych edycjach.

Upewnij się też, czy na serwerach obu partnerów używana jest identyczna struktura katalogów z systemem SQL Server i plikami bazy danych. Jeśli dodasz plik bazy danych do woluminu lub katalogu, który nie istnieje na serwerze lustrzanym, sesja tworzenia kopii lustrzanych zostanie natychmiast wstrzymana.

Konfiguracja systemu SQL Server na serwerach głównym i lustrzanym powinna być identyczna. Dotyczy to np. wielkości pliku tempdb, opcji śledzenia, parametrów uruchomieniowych, ustawień pamięci i poziomu współbieżności.

Wszystkie loginy systemu SQL Server z serwera głównego muszą być dostępne także na serwerze lustrzanym. W przeciwnym razie po przełączeniu awaryjnym aplikacja nie będzie mogła nawiązać połączenia z serwerem. Możesz wykorzystać usługi Integration Services systemu SQL Server 2014 i zadanie *Transfer Logins Task* do skopiowania loginów i haseł z jednego serwera na drugi. Więcej informacji na ten temat zamieszczono w rozdziale 18., „Przesyłanie dzienników w systemie SQL Server 2014”. Ponadto należy ustawić dla tych loginów uprawnienia do bazy danych. Jeśli przenosisz loginy do innej domeny, musisz dopasować ich identyfikatory SID, czyli unikatowe nazwy używane przez system Windows do identyfikowania użytkowników i grup.

Na serwerze głównym znajdować się może także wiele innych obiektów potrzebnych dla aplikacji. Są to np. zadania narzędzia SQL Server Agent, pakietu usług Integration Services, definicje serwerów połączonych, plany konserwacji, powiązane bazy danych, ustawienia usługi SQL Server Mail i Database Mail lub ustawienia koordynatora DTC. Także wszystkie te obiekty trzeba przenieść na serwer lustrzany.

Jeśli używasz uwierzytelniania z systemu SQL Server, musisz ustawić loginy na nowym serwerze głównym po przełączeniu awaryjnym. Możesz wykorzystać do tego procedurę składowaną `sp_change_users_login`.

Koniecznie opracuj procedurę, która po zmodyfikowaniu konfiguracji na serwerze głównym pomoże odtworzyć zmiany na serwerze lustrzanym.

Ponadto w systemie SQL Server 2014 można zarządzać loginami za pomocą niezależnych baz danych. Pozwalają one zapisać dane uwierzytelniające użytkowników bezpośrednio w bazie i przeprowadzać uwierzytelnianie z poziomu bazy, dzięki czemu nie trzeba martwić się zależnością od loginów. W nowych instalacjach pomyśl o zastosowaniu tego rozwiązania. Pozwoli Ci to pozbyć się zależności od loginów.

Po skonfigurowaniu serwera lustrzanego przeprowadź przełączenie awaryjne bazy danych i przez pewien czas używaj aplikacji na nowym serwerze głównym. Jest to najlepszy sposób na upewnienie się, że wszystkie ustawienia są prawidłowe. Zaplanuj testy przełączania awaryjnego na okres niskiej aktywności systemu i poinformuj o nich zainteresowane osoby.

Dostępność bazy danych w trakcie planowanego przestoju

W tym punkcie omawiamy kroki potrzebne przy *aktualizowaniu aktywnego serwera*. Polega to na aktualizowaniu oprogramowania i sprzętu w czasie, gdy baza danych ciągle jest dostępna dla aplikacji. Proces aktualizacji aktywnego serwera zależy od konfiguracji bezpieczeństwa transakcji w sesji tworzenia kopii lustrzanych (opcje SAFETY FULL lub SAFETY OFF).

Aktualizacja aktywnego serwera przy ustawieniu SAFETY FULL

Jeśli w sesji tworzenia kopii lustrzanych używane jest ustawienie SAFETY FULL, w trakcie aktualizowania oprogramowania lub sprzętu wykonaj następujące operacje.

1. Najpierw wprowadź zmiany w sprzęcie i oprogramowaniu serwera lustrzanego. Jeśli wymaga to ponownego uruchomienia systemu SQL Server 2014 lub całego serwera, zrób to. Gdy serwer wznowi pracę, sesja tworzenia kopii lustrzanych zostanie automatycznie uruchomiona i serwer lustrzany zacznie synchronizację danych z serwerem głównym. W czasie, gdy lustrzana baza danych jest wyłączona, dostępny jest serwer główny. Dlatego jeśli używasz serwera świadka, zadбай o to, aby pracował w tym okresie. W przeciwnym razie serwer główny będzie jedynym działającym serwerem i nie będzie mógł udostępniać bazy, ponieważ nie utworzy kworum.
2. Po zsynchronizowaniu serwera lustrzanego z głównym przeprowadź przełączenie awaryjne za pomocą następującego polecenia:
`ALTER DATABASE <nazwa_bazy_danych> SET PARTNER FAILOVER`
 Teraz aplikacja nawiąże połączenie z nowym serwerem głównym. W punkcie „Przekierowywanie klientów na serwer lustrzany”, dalej w tym rozdziale, opisujemy przekierowywanie aplikacji do lustrzanej bazy danych. Transakcje aktywne w momencie przełączania awaryjnego zostaną wycofane. Jeśli jest to nieakceptowalne, możesz na krótki czas przełączania awaryjnego zatrzymać aplikacje, a po przełączeniu ponownie je uruchomić.
3. Teraz przeprowadź aktualizację sprzętu lub oprogramowania na dawnym serwerze głównym. Gdy zakończysz aktualizowanie i baza na dawnym serwerze głównym stanie się dostępna, ten komputer przejmie rolę serwera lustrzanego. Automatycznie uruchomiona zostanie sesja tworzenia kopii lustrzanych i rozpocznie się synchronizacja.
4. Jeśli używasz serwera świadka, przeprowadź na nim aktualizację sprzętu lub oprogramowania.
5. W tym momencie dawny serwer główny pełni funkcję serwera lustrzanego. Po zakończeniu aktualizacji możesz przełączyć bazę z powrotem na dawny serwer główny. Jeśli obie maszyny są identyczne, nie musisz przełączać serwerów; dzięki temu unikniesz tymczasowego zatrzymywania aplikacji. Jeżeli jednak używane komputery różnią się od siebie, warto przełączyć bazę z powrotem na pierwotny serwer główny.

Aktualizowanie aktywnego serwera przy ustawieniu SAFETY OFF

Jeśli do tworzenia kopii lustrzanych używana jest opcja SAFETY OFF, możesz zastosować technikę aktualizowania aktywnego serwera w następujący sposób.

1. Najpierw wprowadź zmiany w sprzęcie lub oprogramowaniu na serwerze lustrzanym. Szczegóły tej operacji są opisane w poprzednim podpunkcie.
2. Na serwerze głównym zmień ustawienie na SAFETY FULL za pomocą następującej instrukcji:

```
ALTER DATABASE <nazwa_bazy_danych> SET SAFETY FULL
```

UWAGA Zaplanuj to zadanie na czas niskiej aktywności systemu, aby ograniczyć czas synchronizacji serwera lustrzanego.

3. Po zsynchronizowaniu serwera lustrzanego przeprowadź przełączenie awaryjne.
4. Zaktualizuj sprzęt i oprogramowanie na dawnym serwerze głównym. Gdy ten serwer wznowi działanie, przejmie rolę serwera lustrzanego i zacznie synchronizację z nowym serwerem głównym.
5. Po zakończeniu synchronizacji możesz przełączyć bazę z powrotem na dawny serwer główny.

Jeśli kopie lustrzane stosujesz tylko do tworzenia dodatkowych kopii baz danych, możesz zrezygnować z przełączania awaryjnego w czasie planowanego przestoju, ponieważ niektóre ustawienia na serwerze lustrzanym mogą być wtedy nieprawidłowe (opisano to w punkcie „Sprzęt, oprogramowanie i konfiguracja serwera”, wcześniej w tym rozdziale). W takiej sytuacji trzeba wyłączyć serwer główny na czas aktualizacji; baza jest wtedy chwilowo niedostępna.

Konfigurowanie zadań narzędzia SQL Server Agent na serwerze lustrzanym

Aby uzyskać identyczną konfigurację na serwerach głównym i lustrzanym, musisz zsynchronizować zadania narzędzia SQL Server Agent na serwerze lustrzanym. W bazie lustrzanej te zadania nie powinny być uruchamiane. Oto zalecane rozwiązania związane z tym zagadnieniem.

- Umieść w krokach zadania sprawdzanie stanu bazy. Wtedy można uruchamiać wybrane kroki tylko dla bazy głównej.
- Włącz wykrywanie zmiany stanu bazy z lustrzanej na główną. Należy wtedy wykonać zadanie narzędzia SQL Server Agent włączające wszystkie inne zadania, które mają być uruchamiane. Przy wykryciu zmiany stanu bazy z głównej na lustrzaną należy zatrzymać lub wyłączyć te zadania.

UWAGA Możesz też wyłączyć zadania i włączać je ręcznie, gdy baza stanie się bazą główną. Jednak ręczne zarządzanie zadaniami przez administratora jest niewygodne, dlatego nie jest to dobre rozwiązanie.

Przekierowywanie klientów na serwer lustrzany

W systemie SQL Server 2014 po nawiązaniu przy użyciu sterowników ADO.NET lub SQL Server Native Client połączenia z bazą danych, dla której tworzone są kopie lustrzane, można wykorzystać możliwości tych sterowników do automatycznego przekierowywania połączeń po przełączeniu awaryjnym. Należy przy tym określić w łańcuchu znaków połączenia początkowy serwer główny i główną bazę, a także opcjonalnie serwer partnera.

Łańcuch znaków połączenia można zapisać na kilka sposobów. Oto przykładowy łańcuch, w którym ServerA to serwer główny, ServerB to serwer lustrzany, a AdventureWorks to baza danych:

```
"Data Source=ServerA;Failover Partner=ServerB;Initial Catalog=AdventureWorks;
Integrated Security=True;"
```

Serwer partnera z tego łańcucha znaków połączenia jest używany jako alternatywny serwer, gdy nie powiedzie się próba nawiązania połączenia z pierwotnym serwerem głównym. Jeśli uda się nawiązać połączenie z pierwotnym serwerem głównym, partner nie jest używany, jednak sterowniki zachowują jego nazwę, pobraną z serwera głównego do pamięci podręcznej po stronie klienta.

Żałujemy, że klient nawiązał połączenie z serwerem głównym i nastąpiło przełączenie awaryjne (automatyczne, ręczne lub wymuszone). Gdy aplikacja ponownie spróbuje użyć połączenia, sterownik ADO.NET lub SQL Server Native Client wykryje błąd połączenia z dawnym serwerem głównym i automatycznie spróbuje połączyć się z nowym serwerem głównym przy użyciu zapisanej nazwy partnera. Jeśli to się powiedzie i nowy serwer główny określi nowy serwer lustrzany, sterowniki pobiorą nazwę nowego partnera i zapiszą ją w pamięci podręcznej klienta. Jeżeli klient nie zdoła połączyć się z alternatywnym serwerem, sterowniki będą na zmianę próbowały nawiązać połączenie z każdym z serwerów do czasu przekroczenia limitu czasu logowania się.

Obsługa klientów wbudowana w sterowniki ADO.NET i SQL Server Native Client ma tę zaletę, że nie trzeba modyfikować kodu aplikacji ani umieszczać w niej specjalnych kodów do obsługi przełączania awaryjnego baz z kopiami lustrzanymi.

Jeśli nie stosujesz automatycznego przekierowywania z wykorzystaniem sterowników ADO.NET lub SQL Server Native Client, możesz wykorzystać inne techniki umożliwiające aplikacjom przełączenie serwera. Przykładowo mechanizm równoważenia obciążenia w sieci (ang. *Network Load Balancing* — NLB) pozwala przekierowywać połączenia z jednego serwera na drugi w sposób niezauważalny dla aplikacji klienckiej, używającej nazwy serwera wirtualnego obsługującego NLB. Po przełączeniu awaryjnym wystarczy zrekonfigurować mechanizm NLB, aby kierował wszystkie aplikacje klienckie używające danej bazy do nowego serwera głównego. Można wykorzystać przy tym zdarzenia zmiany stanu baz uzyskane za pomocą usługi WMI i na ich podstawie ustawiać konfigurację mechanizmu NLB.

Możesz też samodzielnie napisać kod do przekierowywania aplikacji i ponawiania prób połączenia się. Ponadto usługi DNS umożliwiają przekształcanie nazw na adresy IP, co pozwala przekierowywać aplikacje klienckie na nowy serwer główny. Jeśli administrator ma dostęp do serwera DNS (za pomocą skryptów lub przy użyciu narzędzia do zarządzania usługami DNS w systemie Windows) i po przełączeniu awaryjnym może zmodyfikować docelowy adres IP, aby przekierować aplikacje, ten serwer może działać jak odsyłacz dla aplikacji, które cały czas nawiązują połączenie przy użyciu tej samej nazwy. Po modyfikacjach w usłudze DNS będzie ona kierować żądania do bazy danych na nowy serwer główny.

Tworzenie kopii lustrzanych baz i inne metody zapewniania wysokiej dostępności

Tworzenie kopii lustrzanych baz danych to tylko jedna z wielu metod zapewniania wysokiej dostępności systemu SQL Server. Istnieje przynajmniej pięć takich rozwiązań. Oczywiście możliwości niektórych częściowo się pokrywają, a poszczególne techniki mają określone wady i zalety.

- **Tworzenie kopii lustrzanych.** W tym kontekście zastosuj tryb wysokiego bezpieczeństwa z serwerem świadka.
- **Egzemplarze pracujące w klastrze.** Jest to typowa metoda zapewniania wysokiej dostępności. Polega na tworzeniu klastrów WSFC (obejmujących dwa lub więcej węzłów), w których działa system SQL Server. Szczegółowe omówienie klastrów zamieściliśmy w rozdziale 16., „Klastrowanie w systemie SQL Server 2014”.
- **Replikacja transakcyjna.** Na potrzeby dalszych porównań uwzględniana jest architektura z odrębnym serwerem dystrybucyjnym i jednym serwerem subskrybenta, używanym jako rezerwa w przypadku awarii wydawcy.
- **Przesyłanie dzienników.** System SQL Server 2014 ma wbudowany mechanizm przesyłania dzienników. Jego szczegółowy opis znajdziesz w rozdziale 18., „Przesyłanie dzienników w systemie SQL Server 2014”.
- **Grupy dostępności.** W systemie SQL Server 2014 dostępne są grupy dostępności. Ich omówienie znajdziesz w rozdziale 25., „Grupy dostępności AlwaysOn”.

Poniżej porównujemy kopie lustrzane baz danych z pozostałymi technologiami.

Kopie lustrzane baz danych a klastry

Największą różnicą między kopiami lustrzanymi baz danych a klastrami WSFC jest poziom, na którym zapewniana jest nadmiarowość. Kopie lustrzane zapewniają ochronę na poziomie baz danych, natomiast klastry zabezpieczają cały egzemplarz systemu SQL Server.

Jeśli aplikacja korzysta z wielu baz danych, prawdopodobnie lepiej będzie zbudować klastr zamiast stosować kopie lustrzane. Jeżeli chcesz zapewnić dostępność tylko jednej bazy, kopie lustrzane będą dobrym rozwiązaniem, mającym wiele zalet w porównaniu z klastrami (np. łatwiej je skonfigurować).

Kopie lustrzane, w odróżnieniu od klastrów, nie wymagają współużytkowanego magazynu danych, który staje się pojedynczym punktem podatności na awarię. Ponadto kopie lustrzane pozwalają szybciej niż klastry udostępnić rezerwową bazę danych oraz dobrze współdziałają ze sterownikami ADO.NET i SQL Server Native Client w zakresie przekierowywania klientów.

Inna ważna różnica dotyczy tego, że przy tworzeniu kopii lustrzanych serwery główny i lustrzany to odrębne egzemplarze systemu SQL Server o różnych nazwach. W klastrze egzemplarz systemu SQL Server ma jedną wirtualną nazwę i adres IP. Pozostają one takie same niezależnie od tego, w którym węzle ten egzemplarz działa.

Możesz tworzyć kopie lustrzane baz w ramach klastra, aby przygotować aktywny serwer rezerwowy z bazą w klastrze z systemem SQL Server 2014. Pamiętaj jednak, że ponieważ przełączanie awaryjne w klastrze może trwać dłużej niż limit czasu w mechanizmie tworzenia kopii lustrzanych, sesja działająca w trybie wysokiej dostępności może potraktować przełączanie awaryjne w klastrze jak awarię serwera głównego, co spowoduje zmianę stanu kopii lustrzanych.

Aby zwiększyć limit czasu w mechanizmie tworzenia kopii lustrzanych, wywołaj następującą instrukcję:

```
ALTER DATABASE <nazwa_bazy> SET PARTNER TIMEOUT <czas_w_sekundach>
```

Kopie lustrzane baz danych a replikacja transakcyjna

Obie te metody wczytują dziennik transakcji ze źródłowego serwera. Inny jest jednak mechanizm synchronizowania serwerów, a przy tworzeniu kopii lustrzanych następuje bezpośrednie zainicjowanie operacji wejścia-wyjścia na pliku dziennika transakcji w celu przeniesienia rekordów dziennika.

Replikację transakcyjną można stosować dla więcej niż jednego subskrybenta, natomiast przy tworzeniu kopii lustrzanych występuje dwóch partnerów. W bazie subskrybenta dane są dostępne niemal w czasie rzeczywistym, natomiast z kopii lustrzanej bazy nie można wczytać danych do czasu utworzenia snapshota (czyli statycznego obrazu bazy z określonego punktu w czasie).

Kopie lustrzane baz danych a przesyłanie dzienników

Obie te techniki są oparte na przesyłaniu i przywracaniu rekordów z dziennika. Przy tworzeniu kopii lustrzanych baza lustrzana cały czas pracuje w trybie odzyskiwania, dlatego nie można kierować do niej zapytań. Przy przesyłaniu dzienników baza działa w trybie oczekiwania, dlatego można pobierać z niej dane (pod warunkiem, że w danym momencie nie odbywa się przywracanie dziennika). Ponadto przesyłanie dzienników obsługuje model odzyskiwania BULK_LOGGED, a dla kopii lustrzanych dostępny jest tylko model FULL.

W trybie wysokiej wydajności może nastąpić utrata danych, gdy serwer główny przestanie działać, a baza na serwerze lustrzanym zostanie odzyskana w wyniku wymuszonego przełączenia awaryjnego. Jeśli przesyłasz dzienniki z dawnego serwera głównego i dziennik transakcji na tym serwerze nie jest uszkodzony, możesz utworzyć kopię zapasową końcówki dziennika, aby zapisać ostatni zestaw rekordów z dziennika transakcji. Jeżeli w rezerwowej bazie danych przywrócono już pozostałe kopie zapasowe dzienników transakcji, można wprowadzić kopię zapasową końcówki dziennika, aby nie utracić żadnych danych z dawnego serwera głównego. Następnie można porównać dane z serwera rezerwowego ze zdalną bazą danych i ewentualnie skopiować brakujące dane na zdalny serwer.

Możesz połączyć te dwie techniki i za pomocą przesyłania dzienników przenosić rekordy do zdalnej lokalizacji na potrzeby odzyskiwania systemu po katastrofie, a w centrum danych stosować kopie lustrzane w celu zapewniania wysokiej dostępności baz.

Kopie lustrzane baz danych a grupy dostępności

Obie te metody polegają na przenoszeniu rekordów z dzienników transakcji i ich przywracaniu. Grupy dostępności to mechanizm zapewniania wysokiej dostępności wprowadzony w systemie SQL Server 2012. Działa on podobnie jak kopie lustrzane, oferuje jednak następujące dodatkowe możliwości.

- Grupy dostępności wykorzystują klastry WSFC do używania wirtualnej nazwy i adresu IP przy nawiązywaniu połączeń z klientami. Jednak, inaczej niż w klastrach, nie wymagają współużytkowanych dysków; zamiast nich używane są odrębne dyski, tak jak przy tworzeniu kopii lustrzanych.

- W ramach grup dostępności można przełączać awaryjnie dowolną liczbę baz danych. Kopie lustrzane działają dla pojedynczych baz.
- Grupy dostępności obsługują do ośmiu partnerów w trybach synchronicznym i asynchronicznym. Kopie lustrzane działają dla dwóch partnerów.
- Grupy dostępności obsługują przeznaczone tylko do odczytu synchronizowane repliki baz danych, które można wykorzystać do generowania raportów lub jako kopie zapasowe. Przy tworzeniu kopii lustrzanych baza lustrzana jest dostępna tylko jako snapshot.

Snapshoty baz danych

Prawdopodobnie zauważyłeś już, że lustrzana baza danych działa w trybie NORECOVERY, dlatego nie można kierować do niej zapytań. Jeśli chcesz wczytywać dane z lustrzanej bazy w celu generowania raportów, system SQL Server 2014 (edycje Enterprise i Developer) udostępnia wprowadzone w wersji SQL Server 2005 *snapshoty baz danych*. Snapshot to przeznaczony tylko do odczytu statyczny obraz źródłowej bazy danych z określonego punktu w czasie. Snapshoty są przydatne do odczytu danych z bazy lustrzanej. Może istnieć wiele snapshotów, jednak zawsze działają one jako baza w tym samym egzemplarzu systemu SQL Server. Każdy snapshot jest spójny ze względu na transakcje z bazą źródłową (z jej stanem z momentu utworzenia snapshota). Snapshot istnieje do czasu bezpośredniego usunięcia go przez właściciela danej bazy.

Za pomocą tej techniki można utworzyć snapshot bazy lustrzanej, a następnie wczytywać z niego dane, tak jak ze zwykłych baz. Snapshot działa na poziomie stron danych. Zanim strona źródłowej bazy zostanie zmodyfikowana pierwszy raz po utworzeniu snapshota, zostaje skopiowana ze źródłowej bazy do pliku snapshota. Tak działa proces *kopiowania przy zapisie*. W snapshotcie zapisywana jest pierwotna strona, co pozwala zachować rekordy z danymi w stanie z momentu utworzenia snapshota. Późniejsze aktualizacje rekordów ze zmodyfikowanej strony źródłowej bazy nie wpływają na dane w snapshotcie. Dlatego snapshot zachowuje pierwotne strony ze wszystkimi rekordami, które zostały zmodyfikowane od momentu jego utworzenia. Jeśli nawet zmodyfikujesz źródłowe dane, snapshot zachowa ich stan z momentu, w którym został utworzony. Więcej informacji znajdziesz w artykule „Database Mirroring and Database Snapshots” w dokumentacji Books Online poświęconej systemowi SQL Server 2014.

W poniższym przykładzie pokazujemy, jak utworzyć snapshot bazy AdventureWorks:

```
CREATE DATABASE AdventureWorks_Snapshot ON
(NAME = AdventureWorks, FILENAME = '<Twój katalog>\ADW_Mirroring_
snapshot_Data1.SS')
AS SNAPSHOT OF AdventureWorks
```

W pierwotnej bazie wciąż wprowadzane będą zmiany, dlatego jeśli zechcesz umieścić w snapshotcie nowsze informacje, dodane po jego utworzeniu, musisz usunąć snapshot i ponownie go utworzyć. Usuwanie snapshotów odbywa się tak samo jak kasowanie baz danych:

```
DROP DATABASE AdventureWorks_Snapshot
```

Generowanie snapshota negatywnie wpływa na wydajność serwera lustrzanego. Jeśli chcesz tworzyć snapshoty dla wielu baz na serwerze, zbadaj skutki tego rozwiązania. Najważniejsze jest to, że zbyt duża liczba snapshotów może spowolnić wprowadzanie rekordów i zwiększyć opóźnienie bazy lustrzanej względem głównej, co przekłada się na dłuższy czas przełączania awaryjnego.

Ponadto przygotuj na dysku miejsce równe wielkości źródłowej bazy danych, ponieważ gdy dane w źródłowej bazie będą się zmieniać, do pliku snapshota kopiowane będą pierwotne strony, co doprowadzi do wzrostu jego objętości. Inną techniką stosowaną na potrzeby generowania raportów jest replikacja dostępna w systemie SQL Server.

Podsumowanie

Kopie lustrzane baz danych to mechanizm tworzenia rezerwowych baz za pomocą przenoszenia dzienników. Rekordy z dziennika transakcji są przesyłane na dziennik na serwerze lustrzanym, gdy bufor dziennika zostanie zapisany na dysku serwera głównego.

Tworzenie kopii lustrzanych może się odbywać w trybie wysokiej wydajności lub wysokiego bezpieczeństwa. Gdy w trybie wysokiej wydajności używany jest serwer świadka, po awarii serwera głównego serwer lustrzany automatycznie staje się głównym i następuje odzyskanie bazy danych.

Aby zapewnić optymalną wydajność rozwiązania, należy ustalić szybkość generowania rekordów w dzienniku przez aplikacje, liczbę równoległych połączeń i wielkość transakcji.

W środowisku, w którym tworzone są kopie lustrzane, ważna jest przepustowość sieci. Gdy dostępna jest sieć o wysokiej przepustowości i niskim opóźnieniu, kopie lustrzane są solidnym mechanizmem zapewniającym wysoką dostępność przy planowanych i nieplanowanych przestojach.

W rozdziale 20. poznasz wiele różnorodnych zadań administracyjnych związanych z zarządzaniem usługami SSIS (ang. *SQL Server Integration Services*).

Zarządzanie usługami Integration Services i dostrajanie ich wydajności

ZAWARTOŚĆ ROZDZIAŁU:

- Wprowadzenie do usług SSIS.
- Instalowanie i konfigurowanie pakietów SSIS.
- Zabezpieczanie usług SSIS i zarządzanie nimi.
- Pułapki związane z wydajnością usług SSIS.

W tym rozdziale kontynuujemy omawianie tego, w jaki sposób system SQL Server 2014 zmienia sposób pracy administratorów baz danych. Dowiesz się tu, jak sprawniej zarządzać komponentami BI (ang. *Business Intelligence*) w systemie SQL Server. Zestaw narzędzi z obszaru BI obejmuje w wersji SQL Server 2014 usługi SSIS (ang. *SQL Server Integration Services*), SSAS (ang. *SQL Server Analysis Services*) i SSRS (ang. *SQL Server Reporting Services*).

Tu przedstawiamy wiele zróżnicowanych zadań administracyjnych związanych z zarządzaniem usługami SSIS. Zaczynamy od przeglądu tych usług, co pozwoli lepiej zrozumieć zmienne aspekty wymagające uwagi administratora. Gdy zapoznasz się z architekturą usług SSIS, skupisz się na zarządzaniu nimi. Poznasz ich konfigurację, dzienniki zdarzeń i techniki monitorowania. Dalej opisujemy różne zadania administracyjne dotyczące pakietów SSIS (jest to wykonywalny komponent usług SSIS), w tym ich tworzenie, wykonywanie i instalowanie oraz zarządzanie nimi. W ostatnim fragmencie dowiesz się, jak zabezpieczać komponenty usług SSIS.

UWAGA Szczegółowe omówienie usług SSIS znajdziesz w książce *Professional Microsoft SQL Server 2014 Integration Services* (Brian Knight i inni, wydawnictwo Wrox, Indianapolis, 2014).

Przegląd usług SSIS

Najważniejszym oferowanym przez Microsoft komponentem z obszaru BI są usługi SSIS. Służą one przede wszystkim do przenoszenia i porządkowania danych. Bez tych operacji pozostałe komponenty nie mogłyby działać lub zwracałyby błędne dane.

Usługi SSIS zapewniają zaawansowane rozwiązania z zakresu integracji danych i przepływu informacji, obejmujące pobieranie, przekształcanie i wczytywanie danych z różnych źródeł do docelowych lokalizacji. Usługi te oferują rozmaite narzędzia i kreatory pomagające przy tworzeniu procesów przepływu informacji oraz operacji przepływu danych, potrzebnych przy zarządzaniu skomplikowanymi rozwiązaniami z zakresu przenoszenia informacji.

Zastosowania usług SSIS

Przed szczegółowym omówieniem komponentów usług SSIS warto przedstawić częste scenariusze biznesowe, w których stosuje się rozwiązania z tego obszaru. Oto typowe zastosowania usług SSIS:

- archiwizowanie danych (eksportowanie ich),
- wczytywanie nowych danych (importowanie ich),
- przenoszenie danych z jednego źródła do innego,
- porządkowanie lub przekształcanie nieprawidłowych danych,
- wykonywanie zadań administratora, np. usuwanie dawnych plików lub indeksowanie baz.

Często trzeba połączyć dane z różnych źródeł przechowywane w kilku systemach. W tej sytuacji usługi SSIS nawiązują połączenie z każdym źródłem, pobierają dane i scalają je w jeden zbiór. Obecnie w architekturze systemów informatycznych jest to coraz częściej stosowana technika, ponieważ firmy archiwizują informacje niepotrzebne do codziennej działalności, ale przydatne do analizowania trendów biznesowych lub wymagane ze względów prawnych. Możliwe też, że różne jednostki organizacji korzystają z odmiennych technologii przechowywania danych lub różnych schematów do reprezentowania tych samych informacji. W takich sytuacjach usługi SSIS pozwalają ujednolicić dane. Usługi te mogą obsługiwać wiele źródeł danych i przekształcać informacje: zmieniać typy danych, rozdzielać lub scalać kolumny i wyszukiwać bardzo przydatne w takich scenariuszach opisowe dane.

Innym często wykonywanym zadaniem jest zapełnianie i konserwowanie hurtowni danych oraz tematycznych hurtowni danych. W tym kontekście ilość danych jest zwykle bardzo duża, a czas dostępny na pobieranie, przekształcanie i wczytywanie danych jest często krótki. Usługi SSIS obsługują masowe wczytywanie danych bezpośrednio z plików z danymi bez zależności strukturalnych z systemu SQL Server. Dostępny jest też komponent, który pozwala masowo wczytywać dane do systemu SQL Server. Ważną funkcją w kontekście dużych ilości danych i skomplikowanych transformacji jest możliwość wznowiania operacji. Usługi SSIS obsługują punkty kontrolne, co umożliwia ponowne uruchamianie pakietu z poziomu zadania lub kontenera w ramach przepływu sterowania, a także pozwala na sprawną obsługę różnego rodzaju błędów, które mogą wystąpić w trakcie wykonywania skomplikowanych zadań wczytywania danych.

W hurtowniach danych bazy często nie są znormalizowane. Dlatego przy wczytywaniu danych z hurtowni ważna jest możliwość przenoszenia do docelowej lokalizacji danych z wielu różnych tabel lub plików. Pakiety SSIS mogą scalać dane w jeden zbiór i wczytywać je do docelowej tabeli w ramach jednego procesu, bez konieczności zapisywania danych na każdym etapie.

Często użytkownicy chcą zarządzać danymi historycznymi z hurtowni danych lub dzielić je na partycje, aby sprawdzić stan operacji z danego punktu w czasie. Zarządzanie danymi historycznymi związane jest ze skomplikowanymi aktualizacjami. W usługach SSIS pomocny jest wtedy kreator Slowly Changing Dimension Wizard. Ten kreator dynamicznie tworzy i konfiguruje zestaw zadań przekształcających dane, który pozwala zarządzać wstawianiem i aktualizowaniem rekordów, aktualizowaniem powiązanych rekordów i dodawaniem do tabel nowych kolumn przydatnych do zarządzania danymi historycznymi.

Firmy nieraz pobierają dane spoza własnych systemów i muszą sprawdzić jakość tych informacji oraz ustandaryzować je i uporządkować przed wczytaniem do swojego systemu. Usługi SSIS obsługują większość źródeł danych: od niejednorodnych baz danych po pliki z danymi bez zależności strukturalnych. Firmy muszą porządkować dane, gdy w różnych jednostkach stosowane są odmienne standardy i formaty lub po zakupie informacji (np. danych kontaktowych). Czasem formaty różnią się, ponieważ platforma źródłowa jest inna od docelowej. Usługi SSIS udostępniają bogaty zestaw zadań przekształcających dane, który pozwala przeprowadzać różne operacje porządkowania, modyfikowania i wzbogacania informacji. Możesz np. zastąpić wybrane wartości albo pobrać opisy na podstawie kodów z wykorzystaniem dokładnego lub przybliżonego dopasowywania. Za pomocą transformacji grupujących można zidentyfikować powtarzające się rekordy i usunąć je przed wczytaniem danych do docelowej lokalizacji.

Często potrzebna jest możliwość dynamicznego modyfikowania transformacji danych w trakcie ich wprowadzania. Dane trzeba wtedy obsługiwać w inny sposób na podstawie ich wartości, a nawet w zależności od wartości sumarycznych lub liczby wystąpień elementów w danym zbiorze rekordów. Usługi SSIS udostępniają rozbudowany zestaw transformacji przydatnych przy podziale lub scalaniu danych na podstawie wartości, agregowaniu danych, przeprowadzaniu obliczeń na różnych składnikach zbioru danych lub wczytywaniu poszczególnych fragmentów danych do odmiennych lokalizacji. Kontenery SSIS obsługują np. przetwarzanie wyrażeń, przechodzenie po zbiorach informacji i wykonywanie w ramach przepływu informacji zadań na podstawie wyników obliczeń.

Ponadto niektóre funkcje administracyjne wymagają automatyzacji. Usługi SSIS obejmują cały zestaw zadań związanych z funkcjami administracyjnymi. Przy ich użyciu można kopiować obiekty systemu SQL Server lub usprawnić masowe wczytywanie danych. Ponadto w usługach SSIS dostępny jest enumerator obiektów SMO, który przechodzi w pętli po serwerach używanych w firmowym środowisku i wykonuje operacje administracyjne na każdym z nich. Można też zaplanować wykonywanie pakietów SSIS i rozwiązań wykorzystujących usługi SSIS za pomocą zadań narzędzia SQL Server Agent.

Główne elementy usług SSIS

W systemie SQL Server używane są dwa modele — model z instalowaniem pakietów i model z instalowaniem projektów. Wpływają one na sposób pracy programistów i administratorów. Model jest wybierany przy tworzeniu projektu. Później można go zmienić, ale nie należy robić tego pochopnie. Choć w danym środowisku można stosować oba modele, staraj się posługiwać modelem z instalowaniem projektów, ponieważ pozwala na wykorzystanie wszystkich funkcji z wersji SQL Server 2014.

Do wersji SQL Server 2012 administratorzy i programiści mogli stosować wyłącznie model z instalowaniem pakietów. W tym modelu trzeba zainstalować pakiet samodzielnie, a projekt pakietu to tylko kontener, który w bardzo niewielkim stopniu wspomaga pracę. Pakiety uruchamiane w tym modelu trzeba zainstalować w bazie msdb lub w systemie plików serwera.

W tym modelu dostępna jest też usługa SSIS, która monitoruje wykonywanie pakietów. Pakiety można konfigurować z zewnątrz w czasie ich wykonywania za pomocą plików konfiguracyjnych lub wpisów w tabeli konfiguracji.

W modelu z instalowaniem projektów, wprowadzonym w wersji SQL Server 2012, projekt zawierający pakiety ma więcej zadań niż we wcześniejszych wydaniach tego systemu. W tym modelu do projektu lub pakietu można w czasie ich wykonywania przekazać parametry, aby zmienić konfigurację. Projekty są tu bardziej istotne, ponieważ można zainstalować cały projekt, natomiast nie da się instalować poszczególnych pakietów. Zainstalowane pakiety zostają dodane do katalogu usług SSIS i można je wywoływać za pomocą języka T-SQL i w powłoce Windows PowerShell.

W wersji SQL Server 2014 powinienś stosować nowy model z instalowaniem projektów, ponieważ zapewnia bardziej rozbudowaną infrastrukturę do konfigurowania pakietów, zarządzania nimi i rejestrowania zdarzeń. Model z instalowaniem pakietów jest dostępny w celu zachowania zgodności wstecz, jednak nie należy go używać, gdy nie jest to konieczne.

Model z instalowaniem pakietów obejmuje wiele ważnych elementów, w tym usługę, silnik i komponenty wykonawcze, model obiektowy oraz silnik i komponenty przepływu danych. W kolejnych podpunktach omawiamy wszystkie elementy. W wersji SQL Server 2014 są one takie same jak wcześniej.

Usługa SSIS

Usługa SSIS to element architektury odpowiedzialny za zarządzanie przechowywaniem pakietów i monitorowanie ich w trakcie działania. Głównym zadaniem usługi jest zapisywanie dostawców danych w pamięci podręcznej, śledzenie tego, które pakiety są wykonywane, a także monitorowanie pakietów przechowywanych w ich magazynie. Usługa jest używana tylko w modelu z instalowaniem pakietów.

Silnik i komponenty wykonawcze usług SSIS

Silnik wykonawczy usług SSIS jest używany w obu modelach. Odpowiada za zapisywanie układu i projektu pakietów oraz uruchamianie pakietów. Zapewnia też obsługę dodatkowych funkcji, takich jak transakcje, punkty przerywania, konfiguracja, połączenia, obsługa zdarzeń i rejestrowanie zdarzeń. Pliki wykonywalne tego silnika to pakiety, kontenery i zadania. W usługach SSIS występują trzy domyślne stany: sukces, ukończenie i niepowodzenie.

Model obiektowy w usługach SSIS

Dostęp do narzędzi, programów uruchamianych z wiersza poleceń i niestandardowych aplikacji z modelu obiektowego usług SSIS zapewnia zarządzany interfejs API. Choć nie omawiamy tu szczegółowo modelu obiektowego, jest on ważnym elementem usług SSIS.

Silnik i komponenty przepływu danych w usługach SSIS

W przepływie sterowania w pakiecie SSIS zadanie Data Flow Task służy do tworzenia egzemplarzy silnika przepływu danych. Silnik odpowiada za przenoszenie w pamięci danych z lokalizacji źródłowych do docelowych. Ponadto silnik wykonuje żądane transformacje w celu zmodyfikowania danych zgodnie z żądaniami użytkownika. Silnik przepływu danych obejmuje trzy podstawowe elementy.

- *Lokalizacje źródłowe* umożliwiają łączenie się z różnymi źródłami danych (takimi jak tabele lub widoki bazy danych, pliki, arkusze kalkulacyjne, a nawet pliki XML) i pobieranie z nich informacji.
- *Lokalizacje docelowe* służą do wstawiania informacji do równie różnorodnych miejsc.
- *Transformacje* umożliwiają modyfikowanie źródłowych danych przed ich wczytaniem do lokalizacji docelowej. Możliwe jest przy tym wyszukiwanie, scalanie, przestawianie, przekształcanie i obliczanie danych.

Zarządzanie projektem i kontrola zmian

W obszarze BI różnice między programistami i administratorami ulegają zatarciu. Obie te grupy powinny ściśle współpracować. Spójny obraz prac jest możliwy przy użyciu narzędzia SSDT (ang. *SQL Server Data Tools*); jego wcześniejsza nazwa to BIDS (ang. *Business Intelligence Development Studio*). Narzędzie SSDT było w przeszłości powłoką środowiska Visual Studio, dodawaną opcjonalnie w trakcie instalowania systemu SQL Server. W wersji SQL Server 2014 narzędzie SSDT nie jest dostępne w programie instalacyjnym tego systemu i trzeba je pobrać osobno. Rozwiązania i projekty usług SSIS tworzy się właśnie w narzędziu SSDT.

Za konfigurowanie rozwiązań i projektów w SSDT zwykle odpowiadają programiści, którzy jednak konsultują z administratorami różne ustawienia. Do zarządzania usługami SSIS używany jest głównie program SQL Server Management Studio. Przenoszenie rozwiązań SSIS z jednego środowiska do drugiego nieraz wymaga modyfikacji danych w pakietach i skonfigurowania elementów używanych przez te pakiety. Te elementy to np. ustawienia z konfiguracji pakietu, pliki XML, pliki konfiguracyjne i źródła danych. Wszystkie omawiamy dalej w tym rozdziale.

Gdy programista kliknij przycisk *Save* lub zapisze pakiet w narzędziu SSDT, dawna wersja pakietu zostanie natychmiast zastąpiona w systemie plików. Aby temu zapobiec, należy zintegrować środowisko używane do programowania usług SSIS z systemem kontroli wersji, takim jak TFS (ang. *Team Foundation Server*). Dzięki takiej integracji po zapisaniu pakietu zawsze można wrócić do jego wcześniejszej wersji. Obsługiwane są wszystkie systemy kontroli wersji integrujące się ze środowiskiem Visual Studio.

Zarządzanie usługą SSIS

Skoro znasz już elementy usług SSIS, pora przyjrzeć się różnym aspektom zarządzania tymi usługami. Znajdziesz tu informacje potrzebne do wygodnego posługiwania się poszczególnymi komponentami. Zaczniemy od omówienia usługi SSIS i zagadnień związanych z jej konfiguracją. Dalej dowiesz się, jak dostosować właściwości usługi SSIS za pomocą przystawki Usługi (Services) z systemu Windows albo w programie SQL Server Configuration Manager. Zobaczysz też, jak skonfigurować zaporę systemu Windows, a także jak zarządzać dziennikami zdarzeń i monitorowaniem wydajności oraz konfiguracją tych mechanizmów.

Wprowadzenie do usługi SSIS

Usługa SSIS to usługa systemu Windows służąca do zarządzania pakietami SSIS dodanymi w modelu z instalowaniem pakietów. Usługa jest dostępna w programie SQL Server Management Studio i zapewnia następujące możliwości:

- uruchamianie i zatrzymywanie lokalnych oraz zdalnych pakietów,
- monitorowanie lokalnych i zdalnych pakietów,
- importowanie i eksportowanie pakietów z różnych źródeł,
- zarządzanie magazynem pakietów,
- modyfikowanie katalogów magazynu,
- zatrzymywanie działających pakietów w momencie wyłączania usługi,
- wyświetlanie zawartości dziennika zdarzeń systemu Windows,
- nawiązywanie połączeń z wieloma egzemplarzami serwera usług SSIS.

Warto wyjaśnić, że do projektowania lub wykonywania pakietów nie potrzebujesz tej usługi. Jej główną funkcją jest zarządzanie pakietami w programie SQL Server Management Studio. Korzyścią płynącą z używania tej usługi jest to, że narzędzie do projektowania pakietów SSIS w SSDT może ją wykorzystać do zapisywania obiektów w pamięci podręcznej. Zwiększa to wydajność tego narzędzia.

Konfiguracja

Konfigurowanie usługi SSIS obejmuje otwieranie i modyfikowanie pliku XML z jej konfiguracją, ustawianie właściwości usługi za pomocą przystawki Usługi systemu Windows lub programu SQL Server Configuration Manager i konfigurowanie zapory systemu Windows, tak aby przepuszczała usługę Integration Services.

Plik XML z konfiguracją

Plik *MsDtsSrvr.ini.xml* z konfiguracją usługi SSIS znajduje się domyślnie w katalogu *<napęd z systemem SQL Server>\Program Files\Microsoft SQL Server\120\DTS\Binn*. Aby przenieść ten plik w inne miejsce, zmień klucz `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\120\SSIS\ServiceConfigFile` w rejestrze. Ten plik obejmuje ustawienia do określania, czy działające pakiety mają kończyć pracę po zamknięciu usługi, listę katalogów głównych wyświetlanych w oknie *Object Explorer* programu SQL Server Management Studio i ustawienia wpływające na to, którymi katalogami w systemie plików zarządza ta usługa.

Zmianę w konfiguracji trzeba wprowadzić np. przy nawiązywaniu połączenia z nazwanym egzemplarzem systemu SQL Server. Poniżej pokazana jest modyfikacja wprowadzana dla egzemplarza o nazwie *NazwaSerwera\NazwaEgzemplarza*. W domyślnej konfiguracji usługa zawsze prowadzi do katalogu, dlatego przy stosowaniu klastra lub egzemplarza nazwanego trzeba wprowadzić zmiany w konfiguracji.

```
<?xml version="1.0" encoding="utf-8"?>
<DtsServiceConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <StopExecutingPackagesOnShutdown>true</StopExecutingPackagesOnShutdown>
  <TopLevelFolders>
    <Folder xsi:type="SqlServerFolder">
      <Name>MSDB</Name>
      <ServerName>NazwaSerwera\NazwaEgzemplarza</ServerName>
    </Folder>
    <Folder xsi:type="FileSystemFolder">
      <Name>FileSystem</Name>
      <StorePath>..\Packages</StorePath>
    </Folder>
  </TopLevelFolders>
</DtsServiceConfiguration>
```

Inne sytuacje, w których często wprowadza się zmiany w pliku z konfiguracją, to dodawanie ścieżek do pakietów, gdy ścieżki te są inne niż domyślna ścieżka usług SSIS (czyli `C:\Program Files\Microsoft SQL Server\120\DTS\Packages`), lub tworzenie scentralizowanej struktury katalogów dla wielu serwerów, gdy plik z konfiguracją jest przechowywany w centralnym udziale plikowym.

Tworzenie centralnego serwera usług SSIS

Korporacje często używają tak wielu pakietów, że usługi SSIS są oddzielane od systemu SQL Server i umieszczane na odrębnym serwerze. W tym modelu także trzeba mieć licencję na serwer (tak jak przy używaniu systemu SQL Server). Zaletą tego podejścia jest to, że pakiety SSIS nie zajmują pamięci systemu SQL Server, gdy są wczytywane, a ponadto dostępne jest jedno miejsce do zarządzania nimi. Wady to konieczność opłacenia licencji na serwer i dodatkowa warstwa oprogramowania w trakcie debugowania pakietów. Gdy używasz dedykowanego serwera, możesz łatwo umożliwić obsługę większej liczby pakietów — wystarczy dodać więcej pamięci do centralnego serwera. Jednak odrębny serwer może też negatywnie wpływać na wydajność, ponieważ wszystkie zdalne dane trzeba najpierw skopiować przez sieć; dopiero potem trafiają one do buforów używanych przy przepływie danych.

Aby utworzyć centralny serwer usług SSIS, wystarczy zmodyfikować zawartość pliku `MsDtsSrvr.ini.xml` i ponownie uruchomić usługę. Usługa może wczytywać nazwy UNC w formacie `\\NazwaSerwera\Udział` i może być powiązana z wieloma zdalnymi serwerami. Nie zaleca się stosowania zmapowanych napędów, ponieważ konto, które uruchamia usługę SSIS, musi znać dany napęd, dlatego może powstać niepotrzebna zależność od określonego konta. Aby dodać nowe lokalizacje, wystarczy skopiować węzeł `<Folder>` w pliku XML z konfiguracją oraz zastąpić wartości węzłów `<Name>` i `<ServerName>` nowymi danymi.

Możesz zaplanować wykonywanie pakietów za pomocą narzędzia SQL Server Agent lub systemu, takiego jak Harmonogram zadań (Task Scheduler) z systemu Windows. Skoro już opłaciłeś licencję na system SQL Server, lepiej zainstalować go i wykorzystać narzędzie SQL Server Agent, ponieważ oferuje znacznie więcej możliwości. Ponadto w systemie SQL Server możesz zapisać tabele z konfiguracją i tabele używane do rejestrowania zdarzeń, co pozwoli scentralizować przetwarzanie tych elementów.

Po każdej zmianie w pliku konfiguracyjnym musisz zatrzymać i uruchomić usługę SSIS, co opisujemy w dalszych punktach. Skoro już wiesz, jak zmodyfikować plik `MsDtsSrvr.ini.xml` z konfiguracją usługi SSIS, warto zapoznać się z ustawianiem właściwości tej usługi.

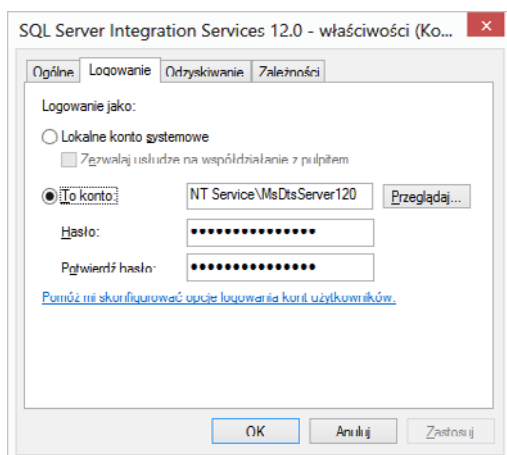
Ustawianie właściwości usługi za pomocą przystawki Usługi systemu Windows

Usługa SSIS (podobnie jak każda inna usługa systemu Windows) ma właściwości, które określają, jak należy ją uruchamiać. W przystawce Usługi możesz:

- ustawić tryb uruchamiania (*Automatyczny, Ręczny, Wylączony*),
- zażądać, aby usługa została uruchomiona, zatrzymana lub ponownie uruchomiona,
- określić, jak komputer ma reagować na awarię usługi,
- wyświetlić lub zmodyfikować listę zależnych usług (domyślnie nie ma takich usług).

Aby wyświetlić i zmodyfikować właściwości usługi SSIS w przystawce Usługi systemu Windows, wykonaj następujące kroki.

1. Otwórz przystawkę Usługi z listy *Panel sterowania/Narzędzia administracyjne* (lub *System i konserwacja/Narzędzia administracyjne* w widoku kategorii).
2. Znajdź na liście usługę *SQL Server Integration Services 12.0* i kliknij ją prawym przyciskiem myszy.
3. Wybierz opcję *Właściwości*, aby wyświetlić obecne ustawienia.
4. W zakładce *Ogólne* możesz wyświetlić i zmienić *Typ uruchomienia* (*Automatyczny*, *Ręczny* lub *Wyłączony*). W trybach *Ręczny* i *Wyłączony* możesz zmienić stan usługi za pomocą przycisków *Uruchom*, *Zatrzymaj* i *Wznów*.
5. W zakładce *Logowanie* (przedstawionej na rysunku 20.1) możesz wyświetlić i zmienić konto używane do uruchamiania i wykonywania usługi. Domyślnie używane jest konto NT Service\MsDtsServer120, jednak większość osób używa domenowego konta usługowego.



Rysunek 20.1. Zakładka Logowanie w oknie właściwości usługi SSIS

6. W zakładce *Odzyskiwanie* możesz ustawić, jak serwer ma reagować na awarię usługi. W polach *Pierwszy błąd*, *Drugi błąd* i *Kolejne błędy* dostępne są opcje: *Nie podejmuj żadnej akcji* (jest to opcja domyślna), *Uruchom usługę ponownie*, *Uruchom program* i *Uruchom ponownie komputer*. Możesz też nakazać usłudze, aby resetowała liczbę błędów po upływie określonej liczby dni.
7. W zakładce *Zależności* możesz zmodyfikować listę usług, od których usługa SSIS jest zależna, a także wyświetlić listę usług zależnych od SSIS. Domyślnie obie te listy są puste.

Ustawianie właściwości usługi za pomocą programu SQL Server Configuration Manager

W programie SQL Server Configuration Manager, podobnie jak w przystawce Usługi systemu Windows, można skonfigurować niewielką liczbę właściwości usługi SSIS. Tu dostępne są ustawienia logowania używane przez usługę i opcje określające sposób jej uruchamiania.

Aby wyświetlić i zmodyfikować właściwości usługi SSIS w programie SQL Server Configuration Manager, wykonaj następujące kroki.

1. Otwórz program SQL Server Configuration Manager (*Wszystkie programy/Microsoft SQL Server 2014/Configuration Tools*).
2. Na liście usług widocznej po prawej stronie kliknij prawym przyciskiem myszy pozycję *SQL Server Integration Services* i wybierz opcję *Właściwości*.
3. W zakładce *Log On* możesz wyświetlić lub zmodyfikować konto używane do uruchamiania i wykonywania usługi. Domyślnie używane jest konto `NT Service\MsDtsServer120`. Dostępne tu opcje są podobne do tych z rysunku 20.1.
4. W zakładce *Service* możesz zmienić sposób uruchamiania usługi (dostępne opcje to *Automatic*, *Disabled* i *Manual*).

Wiesz już, jak konfigurować właściwości usługi SSIS za pomocą przystawki Usługi systemu Windows i programu SQL Server Configuration Manager. Dalej zobaczysz, jak zmodyfikować ustawienia zapory systemu Windows, aby zezwalała na dostęp do tej usługi.

Dzienniki zdarzeń

Usługi SSIS rejestrują w dziennikach zdarzenia zgłaszane przez pakiety w trakcie ich działania. Dostawcy dzienników usług SSIS potrafią dodawać rekordy do plików tekstowych, narzędzia SQL Server Profiler, systemu SQL Server, dziennika zdarzeń systemu Windows i plików XML. Aby rejestrowanie działało, trzeba je włączyć w pakietach i zadaniach usług SSIS. Rejestrować zdarzenia można na poziomie pakietu, kontenera i zadania. Możesz też ustawić różne dzienniki dla pakietów, kontenerów i zadań.

Aby rejestrować zgłoszone zdarzenia, trzeba wybrać dostawcę i dodać dziennik dla pakietu. Dzienniki można tworzyć tylko na poziomie pakietu, a zadanie lub kontener musi używać jednego z dzienników utworzonych dla danego pakietu. Po skonfigurowaniu dzienniki pakietu można wyświetlać za pomocą przeglądarki zdarzeń z systemu Windows i w programie SQL Server Management Studio.

Aby wyświetlić dzienniki zdarzeń usług SSIS w przeglądarce zdarzeń systemu Windows, wykonaj następujące kroki.

1. Otwórz przeglądarkę zdarzeń za pomocą opcji *Panel sterowania/Narzędzia administracyjne/Podgląd zdarzeń* (lub *System i konserwacja/Narzędzia administracyjne/Podgląd zdarzeń* w widoku kategorii).
2. W oknie dialogowym *Podgląd zdarzeń* kliknij węzeł *Dzienniki systemu Windows/Aplikacja*.
3. Gdy wyświetli się panel *Aplikacja*, znajdź w kolumnie *Źródło* wartości *SQLISService120* i *SQLISPackage120*. Wpisy z wartością *SQLISPackage120* są generowane na potrzeby dzienników pakietów, natomiast pozycje z wartością *SQLISService120* to proste komunikaty usługi SSIS.
4. Kliknij daną pozycję prawym przyciskiem myszy i wybierz opcję *Właściwości zdarzenia*, aby wyświetlić szczegółowe informacje.

W celu wyświetlenia zdarzeń w programie SQL Server Management Studio wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z docelowym serwerem z usługami SSIS.
2. W oknie *Object Explorer* kliknij prawym przyciskiem myszy węzeł *Integration Services* i wybierz opcję *View Logs*.

3. W sekcji *Select Logs* wybierz pozycję *SQL Server Integration Services*.
4. Gdy w górnym panelu klikniesz zdarzenie, w dolnym zobaczysz szczegółowe informacje na jego temat.

Monitorowanie aktywności

W ramach monitorowania wydajności usług SSIS należy skonfigurować rejestrowanie liczników wydajności. Liczniki pozwalają wyświetlić i ustalić zasoby używane w trakcie wykonywania pakietów SSIS. Rejestrowanie dotyczy wykorzystania zasobów w czasie, gdy pakiety wykonują zadania związane z przepływem danych.

Skoncentruj się na przydatnych licznikach. Oto wybrane (są to liczniki z poziomu serwera, należące do kategorii *SQLServer:SSIS Pipeline 12.0*).

- **Rows Read.** Określa liczbę wierszy wczytanych z wszystkich źródeł danych w trakcie wykonywania pakietu.
- **Buffers in Use.** Wyświetla liczbę buforów potoków (pól pamięci) używanych w potoku pakietu.
- **Buffers Spooled.** Określa liczbę buforów używanych do obsługi przepływu danych.

Licznik *Buffers Spooled* jest istotny, ponieważ to dobry wskaźnik informujący o braku fizycznej lub wirtualnej pamięci w trakcie obsługi przepływu danych. W niektórych sytuacjach zastosowanie buforów zamiast zapisu danych na dysku powoduje, że pakiet kończy pracę po 20 minutach zamiast po 20 godzinach. Przy każdym zapisie danych z bufora na dysk trzeba zapisać 10 megabajtów danych.

Te liczniki wydajności możesz wykorzystać np. do upewnienia się, że serwer, na którym uruchamiane są pakiety SSIS, ma wystarczająco dużo pamięci. Jednym ze źródeł zatorów w transformacjach są operacje wejścia-wyjścia wykonywane, gdy dane trzeba zapisać na dysku. Usługi SSIS są tak zaprojektowane, aby optymalnie wykorzystywały zasoby systemowe przy przekształcaniu danych przenoszonych z lokalizacji źródłowej do docelowej. Usługi m.in. starają się wykonywać operacje w pamięci, zamiast zapisywać dane na dysku (co wymaga operacji wejścia-wyjścia i prowadzi do spadku wydajności). Gdy w trakcie przeprowadzania transformacji przez pakiety SSIS używana jest tylko pamięć, wartość licznika *Buffers Spooled* jest równa 0. Jeśli zauważysz, że ten licznik często ma wyższą wartość, oznacza to, że na serwerze, na którym działają pakiety SSIS, warto zainstalować dodatkową pamięć.

Narzędzie SQL Server Profiler umożliwia analizowanie operacji na danych i planów wykonywania zapytań wygenerowanych dla różnych zadań w przepływie danych. Na podstawie uzyskanych informacji możesz dopracować indeksy lub zastosować inne techniki optymalizacji źródeł danych używanych przez usługi SSIS.

Zarządzanie pakietami SSIS w modelu z instalowaniem pakietów

Poznałeś już różne aspekty zarządzania usługą SSIS. W tym podrozdziale znajdziesz przegląd elementów pakietów SSIS, a także dowiesz się, jak tworzyć takie pakiety. Dalej omawiamy zarządzanie gotowymi pakietami SSIS. Gdy już nauczysz się tworzyć pakiety i nimi zarządzać, będziesz mógł przejść do instalowania, wykonywania i planowania uruchamiania pakietów SSIS oraz opartych na nich rozwiązań.

Zarządzanie pakietami za pomocą programu SQL Server Management Studio

W podrozdziale „Zarządzanie usługą SSIS” napisano, że do zarządzania pakietami służy głównie program SQL Server Management Studio połączony z usługą SSIS. Po nawiązaniu połączenia z tą usługą pojawiają się dwa główne katalogi — *Running Packages* i *Stored Packages*. Wyświetlane pakiety są przechowywane w tabeli `sysssispackages` w bazie `msdb` lub w katalogu systemu plików ustawionym w pliku konfiguracyjnym usługi SSIS.

Program SQL Server Management Studio służy głównie do monitorowania działających pakietów i zarządzania pakietami przechowywanymi w środowisku usług SSIS.

Informacje na temat obecnie wykonywanych pakietów zawiera katalog *Running Packages*. Dane o tych pakietach są widoczne na ekranie *Summary*. Aby wyświetlić informacje o wybranym pakiecie, należy kliknąć jego nazwę w katalogu *Running Packages*. Jeśli chcesz zatrzymać pakiet z dostępnej w tym katalogu listy, kliknij nazwę pakietu prawym przyciskiem myszy i wybierz opcję *Stop*.

Możesz zmienić sposób przechowywania pakietów. W tym celu dodaj niestandardowe katalogi lub skopiuj pakiety z jednego miejsca w inne za pomocą narzędzi przeznaczonych do importowania i eksportowania elementów. Aby skonfigurować katalogi logiczne wyświetlane w programie SQL Server Management Studio w katalogu *MSDB*, kliknij dany katalog prawym przyciskiem myszy. Inną możliwością to modyfikacja tabeli `sysssispackagefolders` w bazie `msdb`. W tej tabeli katalogi główne mają wartość `null` w kolumnie `parentfolderid`. Dodanie wiersza do tej tabeli powoduje utworzenie katalogu logicznego. Do określania hierarchii katalogów służą kolumny `folderid` i `parentfolderid`. Ponadto możesz skonfigurować domyślne katalogi z systemu plików, które program SQL Server Management Studio ma wyświetlać. To zagadnienie opisano w podpunkcie „Plik XML z konfiguracją”, wcześniej w tym rozdziale. Importowanie i eksportowanie pakietów omawiamy w punkcie „Instalowanie”, dalej w tym rozdziale.

Oto główne zadania związane z zarządzaniem pakietów, wykonywane za pomocą programu SQL Server Management Studio.

- Tworzenie nowych katalogów wyświetlanych w oknie *Object Explorer* i obejmujących pakiety zapisane w systemie plików lub w systemie SQL Server (w tabeli `sysssispackages` bazy `msdb`).
- Importowanie i eksportowanie pakietów.
- Wykonywanie pakietów.
- Usuwanie pakietów.
- Zmienianie nazw pakietów.

Zarządzanie pakietami za pomocą narzędzia DTUtil

Zarządzać pakietami można nie tylko za pomocą programu SQL Server Management Studio, ale też przy użyciu uruchamianego z wiersza poleceń narzędzia *DTUtil*. Główną przyczyną stosowania tego narzędzia jest to, że pozwala ono zarządzać pakietami za pomocą programów szeregujących i plików wsadowych. *DTUtil* (podobnie jak program SQL Server Management Studio) umożliwia kopiowanie, usuwanie, przenoszenie i podpisywanie pakietów, a także sprawdzanie, czy na danym serwerze znajdują się określone pakiety.

Gdy korzystasz z tego narzędzia, możesz dodać opcje `/SQL`, `/FILE` i `/DTS`, aby wskazać lokalizację pakietów, którymi chcesz zarządzać. Za pomocą opcji (parametrów) określa się sposób działania

tego narzędzia. Opcje zaczynają się od ukośnika (/) lub znaku minus (-) i można je umieścić w wierszu poleceń w dowolnej kolejności.

Generowane są kody wyjścia, które informują o problemach ze składnią lub argumentami albo niewłaściwej kombinacji opcji. Gdy wszystko działa poprawnie, DTUtil zwraca kod wyjścia 0 i wyświetla komunikat *The Operation Completed Successfully*. Oto inne kody wyjścia:

- 1: Failed (niepowodzenie),
- 4: Cannot locate package (nie można znaleźć pakietu),
- 5: Cannot load package (nie można wczytać pakietu),
- 6: Cannot resolve the command (niezrozumiała instrukcja).

W trakcie tworzenia instrukcji zwróć uwagę na następujące zasady składniowe.

- Wartościami opcji muszą być łańcuchy znaków. Jeśli w łańcuchu występują spacje, trzeba go ująć w cudzysłów.
- Aby zastosować w łańcuchu znaków apostrof, należy ująć łańcuch w cudzysłów.
- Wielkość znaków w instrukcjach nie ma znaczenia; wyjątkiem od tej reguły są hasła.

Jednym z zastosowań narzędzia DTUtil jest tworzenie nowych identyfikatorów dla skopiowanych pakietów. Warto przypomnieć, że skopiowany pakiet przejmuje nazwę i identyfikator pierwotnego pakietu. Możesz wywołać narzędzie DTUtil z opcją /I [D Regenerate], aby utworzyć identyfikator pakietu. W niektórych sytuacjach pozwala to rozwiązać problemy z pakietem.

UWAGA Aby zaktualizować grupę pakietów w jednym wywołaniu narzędzia DTUtil, utwórz plik wsadowy, który wyszukuje w danym katalogu wszystkie pliki *.dtsx* (są to pliki pakietów) i za pomocą omawianego narzędzia tworzy nowe identyfikatory. Jeśli chcesz wykonać potrzebną instrukcję w pliku wsadowym, zastosuj poniższą składnię w katalogu z projektem SSIS:

```
for %%f in (<Ścieżka_do_pliku>\*.dtsx) do dtutil.exe /i /File %%f
```

Gdy zrozumiesz działanie programu DTUtil, uzyskasz cenną broń w arsenale narzędzi do zarządzania pakietami.

Importowanie i eksportowanie pakietów

Administratorzy powinni też poznać metody przenoszenia pakietów między różnymi lokalizacjami i zmieniania formatu pakietów. Mechanizmy importowania i eksportowania pozwalają dodawać pakiety i kopiować je z jednego miejsca w inne. Ponadto można zmienić format pakietów, a także sposób ich przechowywania (np. z katalogów systemu plików na bazę msdb z systemu SQL Server).

Aby w programie SQL Server Management Studio zaimportować pakiet za pomocą usług SSIS, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z serwerem z usługami SSIS.
2. W oknie Object Explorer rozwiń katalog *Stored Packages* i jego foldery podrzędne, aby znaleźć katalog, do którego chcesz zaimportować pakiet.
3. Kliknij prawym przyciskiem myszy docelowy katalog i wybierz opcję *Import Package*.

4. W oknie dialogowym *Import Package* wybierz lokalizację pakietu (dostępne opcje to *SQL Server*, *File System* i *SSIS Package Store*).
5. Jeśli wybrałeś opcję *SQL Server*, określ serwer, sposób uwierzytelniania, nazwę użytkownika i hasło. Gdy lokalizacja to *SSIS Package Store*, podaj serwer.
6. W tym samym oknie kliknij przycisk *Browse* (widoczny obok pola *Package path*) i wybierz importowany pakiet. Na tym ekranie możesz też ustawić nazwę pakietu używaną w nowej lokalizacji i określić poziom ochrony pakietu.

Eksportowanie pakietów przebiega podobnie. Istotną różnicą jest to, że należy kliknąć prawym przyciskiem myszy eksportowany pakiet i wybrać opcję *Export*, zamiast klikać prawym przyciskiem myszy docelowy katalog i wybierać opcję *Import*. Operacje importowania i eksportowania można też wykonywać za pomocą narzędzia DTUtil.

Instalowanie

Pakiety SSIS i rozwiązania oparte na usługach SSIS utworzone na komputerach lokalnych lub serwerach używanych do programowania trzeba zainstalować na potrzeby testów na serwerach produkcyjnych. Zwykle proces instalacji rozpoczyna się od sprawdzenia w SSDT, czy pakiety działają poprawnie.

Pakiety i rozwiązania możesz instalować za pomocą kilku technik. Oto one.

- Utworzenie instalatora pakietu i użycie kreatora Package Installer Wizard.
- Użycie narzędzi importowania i eksportowania pakietów w programie SQL Server Management Studio.
- Zapisanie lub przeniesienie kopii pakietów w systemie plików.
- Użycie narzędzia DTUtil.

Często modyfikacje, jakie trzeba wprowadzić w rozwiązaniu, wpływają na metodę instalowania i używane narzędzia. Jeśli np. chcesz zmienić jeden pakiet w rozwiązaniu obejmującym trzydzieści pakietów, łatwiej będzie użyć narzędzia importowania w programie SQL Server Management Studio albo zapisać lub przenieść kopie pakietów w systemie plików, niż instalować całe rozwiązanie za pomocą narzędzia instalacyjnego i kreatora Package Installation Wizard.

Wymienione cztery metody można podzielić na podejścia automatyczne i ręczne. Używanie narzędzia instalacyjnego i kreatora Package Installation Wizard to metoda automatyczna, natomiast pozostałe sposoby to techniki ręczne. W następnych podpunktach szczegółowo omawiamy wszystkie dostępne metody.

Zautomatyzowane instalowanie pakietów

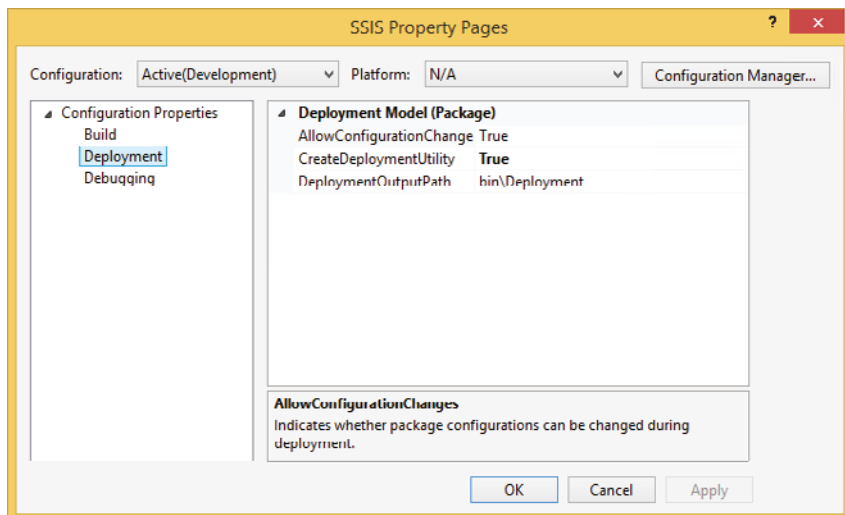
Do instalowania pakietów często używa się narzędzia instalacyjnego. To narzędzie jest dostępne tylko w modelu z instalowaniem pakietów. Narzędzie umieszcza pakiet SSIS, konfigurację, pakiety i pliki pomocnicze w specjalnym katalogu instalacyjnym znajdującym się w katalogu *bin* projektu SSIS. Ponadto generowany jest specjalny plik wykonywalny o nazwie *NazwaProjektu.SSISDeploymentManifest*, który zostaje umieszczony w katalogu instalacyjnym. Po utworzeniu tych elementów można uruchomić plik manifestu, aby zainstalować pakiety.

Metoda obejmuje dwa odrębne kroki. Najpierw należy utworzyć instalator obejmujący wszystkie potrzebne pliki. Następnie trzeba użyć kreatora Package Installer Wizard, aby zainstalować te pliki na docelowym serwerze.

Używanie narzędzia instalacyjnego

Poniżej pokazujemy, jak użyć narzędzia instalacyjnego do zainstalowania rozwiązania SSIS.

1. Otwórz rozwiązanie SSIS w programie SQL Server Data Tools.
2. Kliknij rozwiązanie lub projekt (węzeł z najwyższego poziomu) prawym przyciskiem myszy w oknie *Solution Explorer* i wybierz opcję *Properties*.
3. W oknie dialogowym *[Rozwiązanie/Nazwa projektu] Property Pages* otwórz sekcję *Deployment*.
4. W tej sekcji ustaw wartość opcji *CreateDeploymentUtility* na *True*, co przedstawiono na rysunku 20.2.



Rysunek 20.2. Używanie narzędzia instalacyjnego

5. Możesz też wybrać zmiany w konfiguracji. W tym celu ustaw opcję *AllowConfigurationChanges* na *True*. Opcja pozwala na aktualizację konfiguracji najważniejszych elementów pakietu zależnych od komputera lub środowiska. Te elementy to np. nazwa serwera lub katalog bazy danych (są to właściwości dostępne w menedżerze połączeń z bazą danych).
6. Zbuduj projekt w standardowy sposób. W procesie budowania utworzony zostanie plik *NazwaProjektu.SSISDeploymentManifest*, a pakiety zostaną skopiowane do katalogu *bin/Deployment* (lub innego katalogu ustawionego we właściwości *DeploymentOutputPath* w sekcji *Deployment* w oknie *Property Pages*).

Ponieważ omawiane narzędzie kopiuje wszystkie pliki rozwiązania, możesz też zainstalować dodatkowe pliki (np. *Readme*). Wystarczy umieścić je w katalogu *Miscellaneous* projektu SSIS.

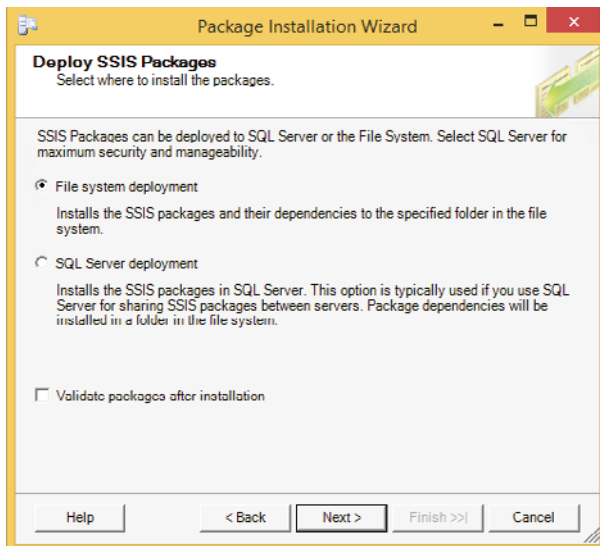
Używanie kreatora Package Installer Wizard

Po utworzeniu za pomocą narzędzia instalacyjnego pliku *.SSISDeploymentManifest* możesz zainstalować pakiety. Użyj w tym celu kreatora Package Installation Wizard. Aby go uruchomić, kliknij plik *.SSISDeploymentManifest*. Kreator wywołuje program *DTSInstall.exe* i kopiuje pakiety oraz pliki konfiguracyjne do wskazanego miejsca.

Kreator Package Installation Wizard udostępnia przydatne możliwości, które są niedostępne lub niewygodne w użyciu w ręcznych metodach instalacji. Możesz np. wybrać sposób instalacji — oparty na systemie plików lub na systemie SQL Server. Zależności w postaci plików zawsze są instalowane w systemie plików. Innym ważnym (i przydatnym) udogodnieniem jest możliwość modyfikowania konfiguracji używanej na docelowym serwerze. Dzięki temu można zmienić w kreatorze wartości właściwości (np. nazwę serwera).

Oto kroki, które trzeba wykonać, aby z powodzeniem zainstalować pakiety za pomocą kreatora Package Installation Wizard.

1. W Eksploratorze plików systemu Windows znajdź ścieżkę do katalogu z plikiem *.SSISDeploymentManifest* (zwykle jest to podkatalog */bin/Deployment* w katalogu rozwiązania lub projektu).
2. Po utworzeniu plików w katalogu *Deployment* skopiuj go wraz z zawartością na docelowy serwer.
3. Na docelowym serwerze otwórz katalog *Deployment* i kliknij dwukrotnie plik *.SSISDeploymentManifest*. Uruchomi się kreator Package Installation Wizard (program *DTSInstall.exe*).
4. Na stronie *Deploy SSIS Packages* określ, czy pakiety chcesz zainstalować w systemie plików, czy w systemie SQL Server (patrz rysunek 20.3). Opcjonalnie możesz też sprawdzić poprawność pakietów przed zainstalowaniem.



Rysunek 20.3. Używanie kreatora Package Installation Wizard

5. Na stronie *Select Installation Folder* możesz podać ścieżkę do katalogu (przy instalacji w systemie plików) lub nazwę serwera i odpowiednie dane uwierzytelniające (przy instalacji w systemie SQL Server).
6. Przy instalacji w systemie SQL Server na stronie *Select Installation Folder* podaj ścieżkę do katalogu przeznaczonego na zależności, które trzeba zapisać w systemie plików. Jeśli zaakceptujesz w tym miejscu ustawienia domyślne, będziesz mógł później zmienić lokalizację plików z konfiguracją pakietu.

7. Jeśli pakiet obejmuje pliki konfiguracyjne i w trakcie tworzenia manifestu ustawiłeś opcję `AllowConfigurationChanges` na wartość `True`, pojawi się opcjonalna strona *Configure Packages*. Można na niej zaktualizować konfigurację.
8. Jeżeli zażądałeś sprawdzania poprawności pakietów, pojawi się opcjonalna strona *Packages Validation*, na której można przejrzeć wyniki tego procesu.

Instalowanie pakietów za pomocą ich importowania lub eksportowania

Wcześniej w tym rozdziale dowiedziałeś się, jak importować i eksportować pakiety. Pozwala to dodać pakiety lub skopiować je z jednego miejsca w inne. Można też zmienić format pakietów i sposób ich przechowywania. Jednym z oczywistych zastosowań tego mechanizmu jest instalowanie pakietów po ich utworzeniu i przetestowaniu.

Ciekawą zaletą tego podejścia jest możliwość zmiany sposobu przechowywania (np. z katalogów z systemu plików na bazę `msdb` z systemu SQL Server). Jest to przydatne przy odzyskiwaniu systemu po katastrofie, ponieważ można zapisać pakiety w różnych formatach i miejscach, co jest dodatkowym zabezpieczeniem rozwiązań SSIS.

Instalowanie pakietów za pomocą ich zapisywania lub przenoszenia

Prawdopodobnie najprostszym sposobem instalowania pakietów jest skopiowanie ich z katalogu *bin* projektu utworzonego w środowisku Visual Studio na docelowy serwer. Metoda nie oferuje żadnych dodatkowych możliwości, jednak dobrze sprawdza się dla niewielkich rozwiązań SSIS. Ważną wadą tego podejścia jest brak możliwości instalowania pakietów w systemie SQL Server.

Instalowanie pakietów za pomocą narzędzia DTUtil

Narzędzie DTUtil (podobnie jak program SQL Server Management Studio) umożliwia kopiowanie i przenoszenie pakietów. Wcześniej w rozdziale wspomniano, że zaletą narzędzia DTUtil jest możliwość wykonywania utworzonych instrukcji zgodnie z harmonogramem lub w późniejszym terminie. Pozwala to zaplanować instalację pakietów na innym serwerze z wykorzystaniem poleceń `COPY` lub `MOVE` tego narzędzia, które przeniosą zmodyfikowane pakiety na docelowy serwer.

Poniższy przykład stanowi ilustrację użycia polecenia `COPY` narzędzia DTUtil do instalowania pakietów:

```
dtutil /DTS srcPackage.dtsx /COPY SQL;destPackage
```

Zarządzanie pakietami SSIS w modelu z instalacją projektów

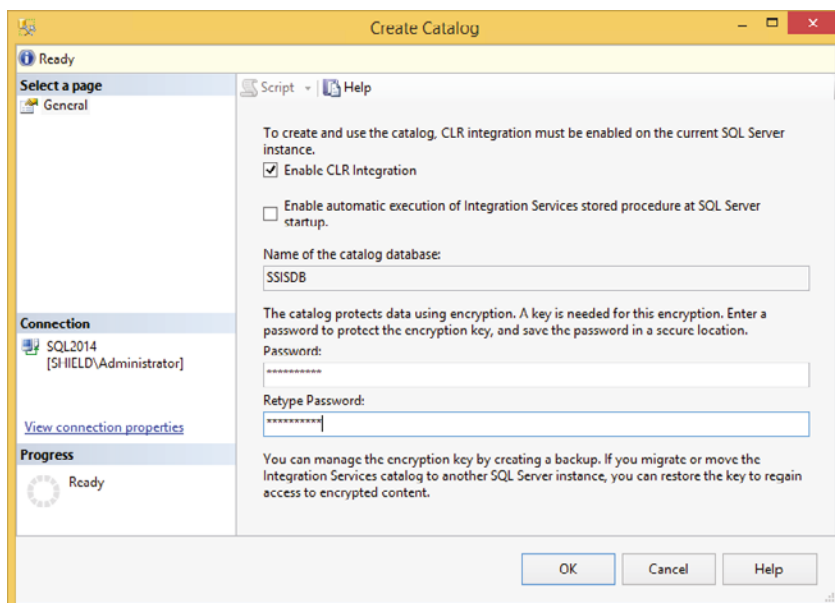
Do tego miejsca koncentrowaliśmy się na modelu z instalacją pakietów. Było to jedyne podejście dostępne w wersjach SQL Server 2005 i 2008. W wersjach SQL Server 2012 i 2014 można zarządzać pakietami w projekcie (który obejmuje grupę pakietów) oraz instalować je w ten sposób. Większość programistów jest przyzwyczajona do pracy z wykorzystaniem pakietów. W tym podrozdziale pokazujemy, jak skonfigurować katalog SSIS, a następnie zainstalować w nim pakiety.

Konfigurowanie katalogu SSIS

W modelu z instalowaniem pakietów można umieścić je w bazie msdb lub w systemie plików. W modelu z instalowaniem projektów obsługiwany jest tylko katalog SSIS; znajduje się on w egzemplarzu bazy danych. Dlatego w tym podejściu usługa SSIS nie jest potrzebna. Pakiety są uruchamiane w kontekście egzemplarza bazy danych, a gdy silnik bazodanowy przestanie działać, usługi SSIS także zawiodą lub nastąpi ich przełączenie awaryjne w klastrze.

Przed zainstalowaniem projektu musisz przygotować katalog SSIS. W tym celu otwórz program SQL Server Management Studio i wykonaj następujące kroki.

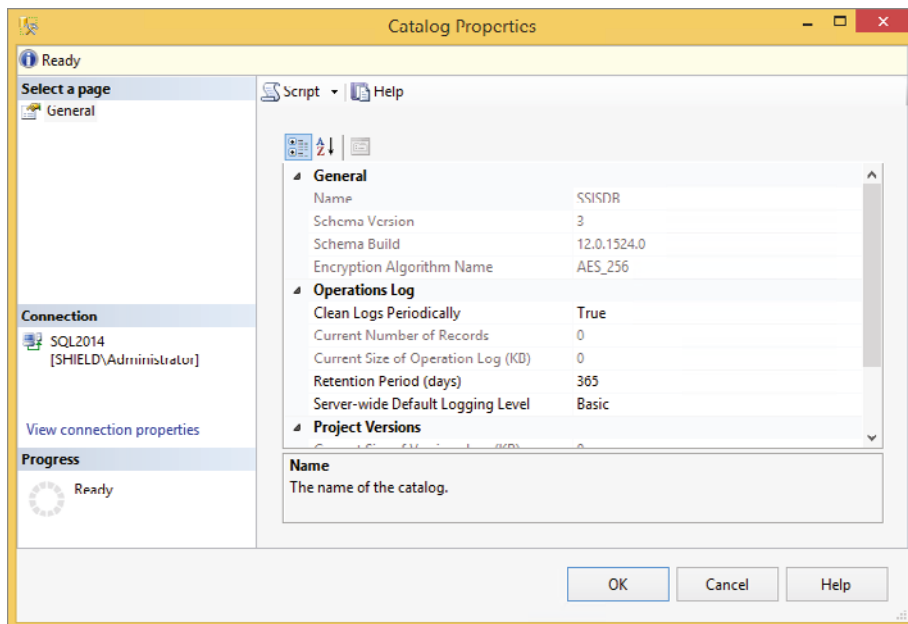
1. Nawiąż połączenie z silnikiem bazodanowym w programie SQL Server Management Studio.
2. Kliknij prawym przyciskiem myszy węzeł *Integration Services Catalogs* i wybierz opcję *Create Catalog*. Pojawi się okno dialogowe *Create Catalog* przedstawione na rysunku 20.4. W tym oknie dialogowym należy wpisać hasło, co pozwoli wygenerować klucz na potrzeby szyfrowania. Można tu także włączyć obsługę środowiska CLR, aby wykonywać pakiety za pomocą kodu w języku T-SQL. Kliknij przycisk OK, aby utworzyć katalog.



Rysunek 20.4. Tworzenie katalogu SSIS

Utworzenie katalogu skutkuje wygenerowaniem bazy SSISDB. Dla egzemplarza bazy danych można utworzyć tylko jeden katalog. Aby pobrać metadane na temat pakietów, użyj zapytań do bazy SSISDB. Możesz też wczytać wybrane tabele z tej bazy, by uzyskać informacje o ostatnich błędach pakietów; te dane są pobierane z dzienników.

Po utworzeniu katalog należy skonfigurować. W tym celu kliknij utworzony wcześniej katalog prawym przyciskiem myszy i wybierz opcję *Properties*. Pojawi się okno dialogowe *Catalog Properties*, przedstawione na rysunku 20.5. W tym oknie wybierz poziom rejestrowania zdarzeń i liczby dni, przez które dzienniki mają być przechowywane. Dostępne poziomy rejestrowania to *Basic* (jest to poziom domyślny), *Verbose*, *Performance* i *None*. W trybach *Verbose* i *Performance* generowane są komunikaty o problemach z wydajnością pakietów SSIS. Te dwie opcje stosuje się tylko tymczasowo na potrzeby diagnozowania pakietów.

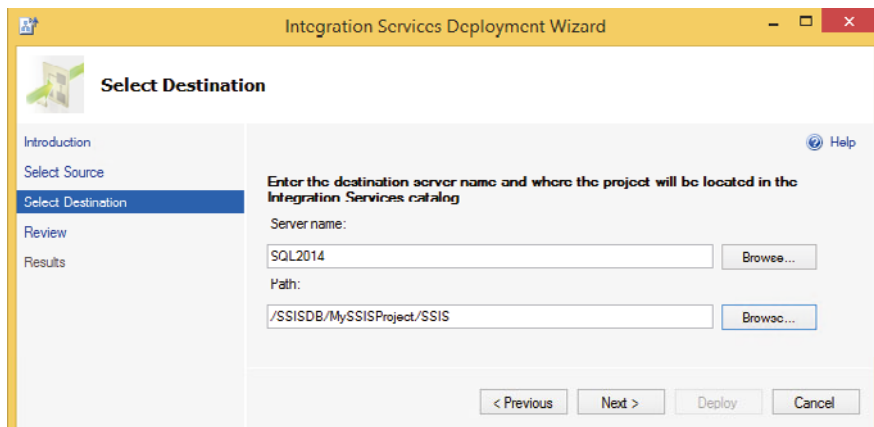


Rysunek 20.5. Konfigurowanie katalogu SSIS

Ponadto możesz wybrać, ile wersji projektu ma być zachowywanych. To pozwala cofnąć się do wcześniejszej wersji projektu. Domyślnie zachowywanych jest dziesięć wersji pakietów.

Instalowanie pakietów

Najprostszy sposób instalowania pakietów to zastosowanie modelu z instalacją projektów i kreatora Integration Services Deployment Wizard. Kreator można uruchomić w narzędziu SSDT. W tym celu kliknij projekt prawym przyciskiem myszy i wybierz opcję *Deploy*. Możesz też wybrać kreator z grupy narzędzi *SQL Server 2014/Integration Services*. Kreator w ramach instalacji wyświetla kilka pytań. Na pierwszym ekranie (patrz rysunek 20.6) należy określić docelowy serwer i ścieżkę.



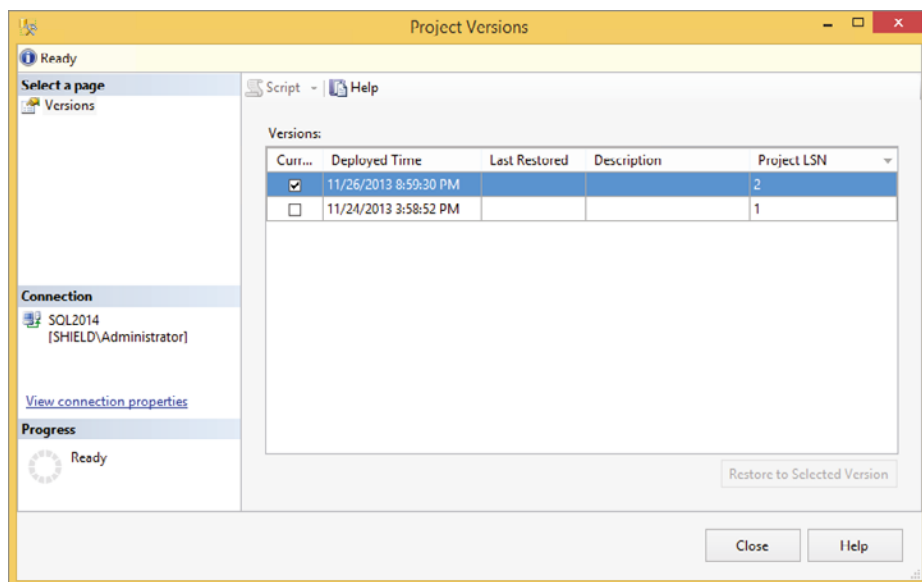
Rysunek 20.6. Instalowanie projektu

Jeśli potrzebny katalog nie istnieje, kliknij przycisk *Browse* i utwórz go. Aby zainstalować projekt, trzeba przygotować katalog. Pełni on funkcję kontenera na jeden lub więcej projektów i pomaga zarządzać konfiguracją środowiska.

Po ustawieniu poprawnej ścieżki kliknij przycisk *Next*, aby zainstalować pakiet. W Eksploratorze plików znajdziesz teraz plik *.ispac* katalogu *Deployment*. Jeśli prześlesz ten plik do administratora, będzie on mógł kliknąć go dwukrotnie, aby uruchomić ten sam kreator. Wspomniany plik zawiera wszystkie potrzebne w projekcie informacje o pakietach i konfiguracji.

WYCOFYWANIE ZMIAN W PROJEKCIE

Wyobraź sobie, że programista przypadkowo wysłał administratorowi niewłaściwy plik projektu do zainstalowania. Jest godzina 3.00 w nocy i administrator nie potrafi przywrócić stanu środowiska. W systemie SQL Server 2014 można łatwo wycofać zmiany w projekcie. Wystarczy kliknąć projekt prawym przyciskiem myszy w programie SQL Server Management Studio i wybrać opcję *Versions*. W oknie dialogowym *Project Versions* (przedstawionym na rysunku 20.7) wybierz starszą wersję projektu i kliknij przycisk *Restore to selected version*, aby ją przywrócić. Nie stosuj jednak tego mechanizmu jako podstawowego systemu kontroli wersji kodu źródłowego. Technika ta służy wyłącznie do wycofywania zmian na serwerze i nie pozwala naprawić błędów programistycznych, które pojawiły się przed zainstalowaniem projektu.



Rysunek 20.7. Wycofywanie zmian w projekcie

Konfigurowanie pakietów

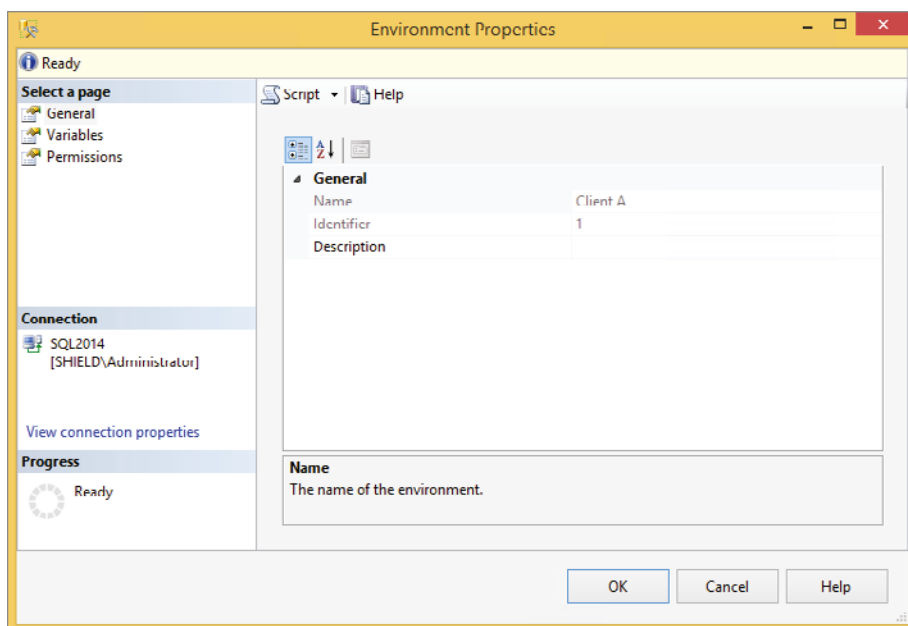
Następny krok po instalacji polega na skonfigurowaniu pakietu pod kątem danego środowiska. Możliwe, że programista ustawił w pakietach nazwy i hasła z serwera rozwojowego. W modelu z instalacją pakietów używane są pliki konfiguracyjne, natomiast w modelu z instalacją projektów stosuje się parametry. Programista odpowiada za przygotowanie parametrów i skonfigurowanie

pakietów tak, aby z nich korzystały. Dzięki temu administrator może zmienić wartości parametrów, jeśli zechce zmodyfikować konfigurację pakietu.

Środowiska

Środowiska (ang. *environment*) pozwalają na wprowadzanie znacznych zmian w konfiguracji pakietów. Możliwe jest ustawianie innych zmiennych dla różnych klientów z tego samego serwera lub używanie odmiennych zmiennych w systemach rozwojowym i produkcyjnym. Środowiska obejmują kolekcje zmiennych, w których można zapisać konfigurację projektu lub pakietu. Aby zastosować tę technikę, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i rozwiń węzeł *Integration Services Catalogs*.
2. W tym węźle rozwiń wybrany katalog, kliknij prawym przyciskiem myszy węzeł *Environments* i wybierz opcję *Create Environment*.
3. W tym ćwiczeniu utwórz dwa środowiska: *Client A* i *Client B*. Po ich dodaniu kliknij wybrane środowisko prawym przyciskiem myszy i wybierz opcję *Properties*. Pojawi się okno dialogowe *Environment Properties*, przedstawione na rysunku 20.8.



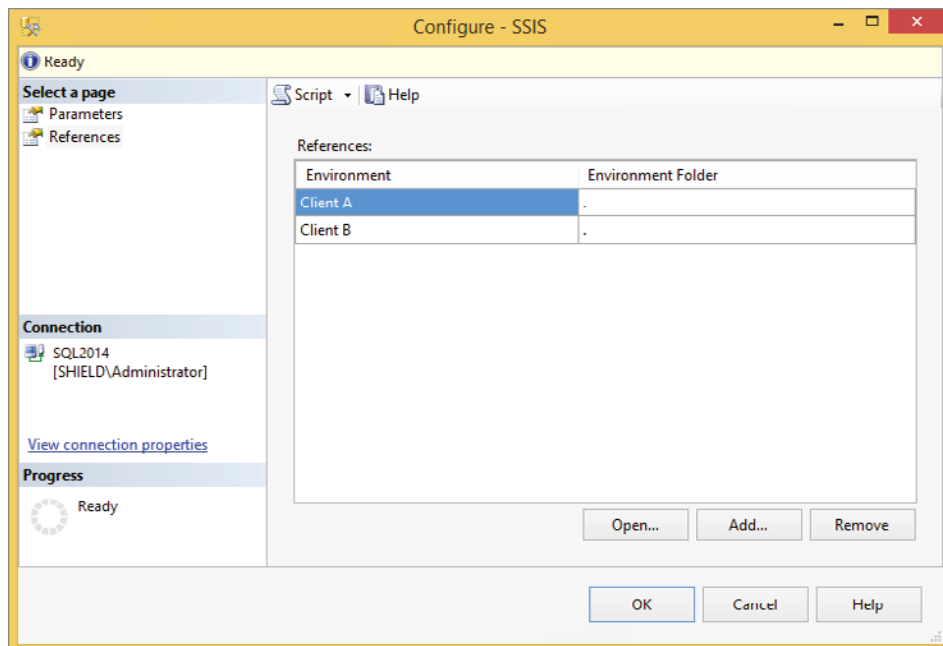
Rysunek 20.8. Konfigurowanie środowiska

4. Otwórz zakładkę *Variables* i utwórz nową zmienną tekstową *ServerName*. Przypisz do niej wartość *ServerA*. Powtórz ten krok dla środowiska *Client B*, jednak tu ustaw wartość *ServerB*. Jeśli chcesz zaszyfrować tę wartość, zaznacz opcję *Sensitive*.

Używanie środowisk

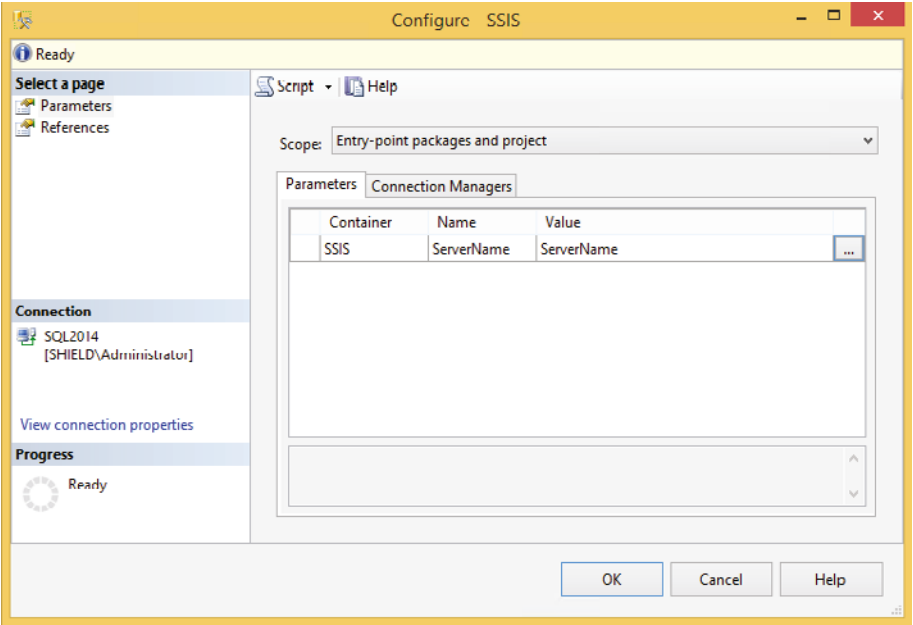
Aby wykorzystać środowisko, najpierw trzeba umożliwić dostęp do niego w projekcie. Wykonaj więc następujące kroki.

1. Kliknij projekt prawym przyciskiem myszy i wybierz opcję *Configure*.
2. Na stronie *References* dodaj referencję do środowiska, które chcesz zastosować dla danego projektu. W tym celu kliknij przycisk *Add* widoczny na rysunku 20.9.

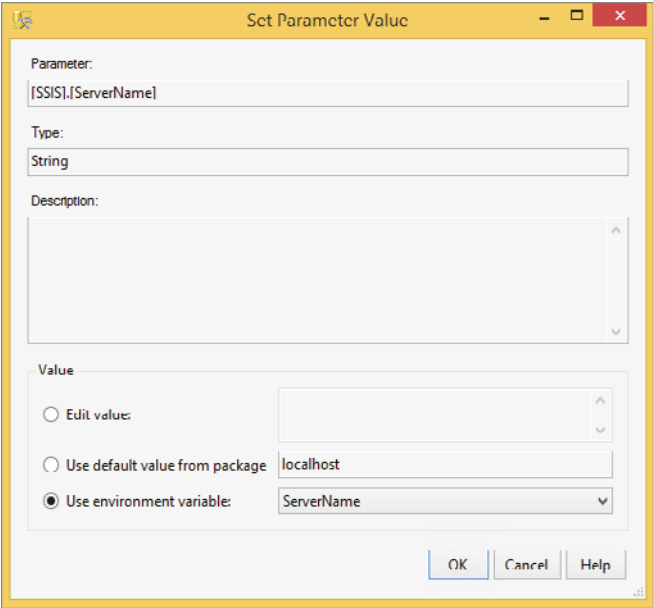


Rysunek 20.9. Dodawanie środowiska dla projektu

3. Po utworzeniu referencji możesz ponownie skonfigurować pakiet, aby go użyć. Na ekranie *Configure Project* otwórz stronę *Parameters*, co przedstawiono na rysunku 20.10. Znajduje się tu lista parametrów i menedżerów połączeń, które można łatwo skonfigurować poza pakietem. Na rysunku 20.10 parametr *ServerName* jest podkreślony. To oznacza, że jest on powiązany z utworzoną w poprzednim punkcie zmienną środowiska.
4. Aby zmienić wartość parametru, kliknij widoczny obok parametru przycisk wielokropka. Pojawi się okno dialogowe *Set Parameter Value*, widoczne na rysunku 20.11. Na tym ekranie można ustawić wartość parametru w polu *Edit Value*. Inne rozwiązanie to ustawienie wartości ze środowiska. W tym celu należy wybrać zmienną środowiska z listy rozwijanej.



Rysunek 20.10. Konfigurowanie parametrów pakietu



Rysunek 20.11. Określanie wartości parametru

Wykonywanie i planowanie uruchamiania pakietów

Wiesz już, jak tworzyć rozwiązania SSIS, zarządzać nimi i instalować je. W tym podrozdziale koncentrujemy się na wykonywaniu tych rozwiązań i planowaniu ich uruchamiania. Pakiety można wykonywać za pomocą różnych narzędzi (to samo dotyczy także innych zadań administracyjnych związanych z pakietami i rozwiązaniami). Pakiety uruchamiane są za pomocą następujących narzędzi:

- SQL Server Data Tools,
- SQL Server Import and Export Wizard (uruchamiane w programie SQL Server Management Studio),
- uruchamiane z wiersza poleceń narzędzie DTEXec,
- narzędzie DTEXecUI,
- Execute Package Utility,
- zadania narzędzia SQL Server Agent,
- język T-SQL (w modelu z instalacją projektów).

Używane narzędzia zależą np. od etapu cyklu życia danego pakietu. I tak w trakcie rozwijania pakietu sensownym wyborem jest SSIS Designer w narzędziu SSDT, ponieważ dostępne są tu mechanizmy pomocne przy programowaniu (np. zmiany koloru tła zadań informujące o postępie w wykonywaniu pakietów).

Uruchamianie pakietów w narzędziu SSDT

Początkowo pakiety zwykle wykonuje się w narzędziu SSDT, ponieważ jest to środowisko programistyczne służące do tworzenia rozwiązań SSIS. W SSDT wystarczy kliknąć pakiet prawym przyciskiem myszy i wybrać opcję *Execute Package*, wcisnąć klawisz *F5* lub wybrać przycisk *Start* na pasku narzędzi. Najlepszą techniką jest kliknięcie pakietu prawym przyciskiem myszy i wybranie opcji *Execute Package*. Chroni to przed uruchomieniem innych pakietów, gdy rozwiązanie jest niewłaściwie skonfigurowane.

Uruchamianie pakietów za pomocą kreatora Import and Export Wizard

Kreator Import and Export Wizard służy do kopiowania danych między dwoma lokalizacjami. Mogą to być egzemplarze systemu SQL Server, a także system SQL Server i plik z danymi bez zależności strukturalnych. W tle to narzędzie tworzy bardzo prosty pakiet SSIS.

Przy używaniu narzędzia Import and Export Wizard w programie SQL Server Management Studio można natychmiast uruchomić pakiet. Pozwala to przenieść pakiet i wykonać go w jednym kroku.

Uruchamianie pakietów za pomocą narzędzia DTExec

Głównym zastosowaniem narzędzia DTExec jest uruchamianie pakietów z wiersza poleceń, za pomocą skryptów lub na podstawie harmonogramów. Narzędzie obsługuje wszystkie mechanizmy związane z konfigurowaniem i wykonywaniem pakietów. Możliwe jest wczytywanie i uruchamianie pakietów z systemu SQL Server, usługi SSIS i systemu plików.

W instrukcjach narzędzia DTExec trzeba przestrzegać następujących zasad.

- Wszystkie opcje instrukcji muszą zaczynać się od ukośnika (/) lub znaku minus (-).
- Argumenty obejmujące spacje trzeba ujmować w cudzysłów.
- Wartości obejmujące apostrof trzeba ujmować w cudzysłów.

Oto ogólna składnia instrukcji DTExec:

`Dtexec /opcja wartość`

Poniżej znajduje się przykładowe polecenie uruchamiające pakiet *CaptureDataLineage.dtsx*. Opcja /FILE pozwala wskazać pakiet przechowywany w systemie plików w modelu z instalowaniem pakietów. Opcja /CONNECTION umożliwia zmianę menedżera połączenia w momencie wykonywania pakietu.

```
Dtexec /FILE "C:\Program Files\Microsoft SQL Server\120\Samples\Integration
Services\Package Samples\CaptureDataLineage
Sample\CaptureDataLineage\CaptureDataLineage.dtsx " /CONNECTION
" (local).AdventureWorks "; "\ "Data Source=(local);Initial
Catalog=AdventureWorks;Provider=SQLNCLI.1;Integrated Security=SSPI;Auto
Translate=False;\ " " /REPORTING
E
```

Gdy uruchamiasz pakiet za pomocą narzędzia DTExec, możesz otrzymać następujące kody wyjścia:

- 0: Successful execution (udane wykonanie),
- 1: Failed (niepowodzenie),
- 3: Canceled by User (anulowane przez użytkownika),
- 4: Unable to Find Package (nie można znaleźć pakietu),
- 5: Unable to Load Package (nie można wczytać pakietu),
- 6: Syntax Not Correct (nieprawidłowa składnia).

Istnieje wiele opcji zmieniających sposób wykonywania pakietu. Przykładowo opcja /Decrypt pozwala ustawić hasło w celu zabezpieczenia informacji z pakietu, a opcja /Set służy do przypisywania wartości do zmiennych pakietu SSIS w czasie jego wykonywania. Opcje są przetwarzane zgodnie z kolejnością ich występowania. Gdy używasz opcji /Set i /ConfigFile, podane wartości także są przetwarzane w takiej kolejności. Wielkość znaków w opcjach i argumentach nie ma znaczenia (wyjątkiem są hasła).

Uruchamianie pakietów za pomocą narzędzia DTExecUI (w modelu z instalowaniem pakietów)

W programie DTExecUI (jest to graficzny odpowiednik narzędzia DTExec) można skonfigurować różne opcje potrzebne przy uruchamianiu pakietów. Dzięki kreatorowi używanemu przez to narzędzie do zbierania szczegółowych informacji o wykonywaniu pakietów możesz lepiej zrozumieć liczne opcje i poznać składnię uruchamiania pakietów. W celu zastosowania tego kreatora wykonaj następujące kroki.

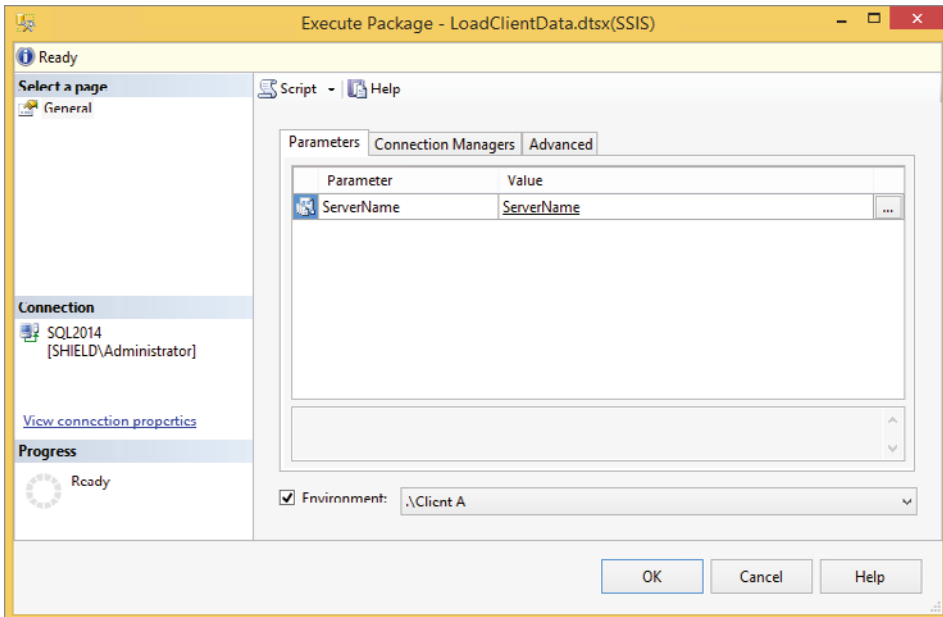
1. Uruchom narzędzie DTExecUI. W tym celu kliknij dwukrotnie plik z rozszerzeniem *.dtsx* lub kliknij pakiet prawym przyciskiem myszy w programie SQL Server Management Studio i wybierz opcję *Run*.
2. Wybierz potrzebne opcje po lewej stronie ekranu i ustaw ich wartości w głównym panelu. Na ostatniej stronie wyświetlana jest instrukcja wiersza poleceń potrzebna do uruchomienia pakietu z ustawionymi opcjami.
3. Po przejściu przez różne strony i zapoznaniu się z instrukcją wiersza poleceń kliknij przycisk *Execute*. To spowoduje przesłanie przez narzędzie DTExecUI instrukcji wiersza poleceń do silnika usług SSIS. Zachowaj ostrożność przy stosowaniu tego narzędzia w środowisku 64-bitowym, ponieważ jest ono przeznaczone dla systemów 32-bitowych. W środowisku 64-bitowym należy stosować 64-bitową wersję narzędzia DTExec w wierszu poleceń lub narzędzie SQL Server Agent.

Głównym powodem, dla którego warto zapoznać się z narzędziami DTExec i DTExecUI jest to, że są one przydatne do testowania pakietów i sprawdzania poprawności instrukcji wiersza poleceń uruchamianych później według harmonogramu za pomocą narzędzia SQL Server Agent.

Uruchamianie pakietów za pomocą narzędzia Execute Package (w modelu z instalowaniem projektów)

W modelu z instalowaniem projektów wykonywanie i konfigurowanie pakietów jest jeszcze łatwiejsze. Aby uruchomić pakiety w tym modelu, wykonaj następujące kroki.

1. Kliknij pakiet prawym przyciskiem myszy w programie SQL Server Management Studio i wybierz opcję *Execute*. Pojawi się okno dialogowe *Execute Package*, widoczne na rysunku 20.12.



Rysunek 20.12. Okno narzędzia Execute Package

2. Jeśli dla pakietu używane są zmienne środowiska, zaznacz pole opcji *Environment* i wybierz środowisko, które chcesz zastosować przy wykonywaniu pakietu. Pozwala to skonfigurować pakiet, tak aby używał utworzonej wcześniej kolekcji zmiennych ze środowiska.
3. Jeżeli chcesz zmienić menedżer połączenia, użyj zakładki *Connection Managers*. Umożliwia ona zmianę połączenia na potrzeby jednego wykonania pakietu. W zakładce *Advanced* możesz ustawić poziomy rejestrowania zdarzeń i tryb 32-bitowy.
4. Kliknij przycisk OK. Pakiet zostanie uruchomiony asynchronicznie, co oznacza, że będzie wykonywany w tle za pomocą języka T-SQL.
5. Aby wyświetlić informacje o stanie wykonywania pakietu, otwórz raport operacyjny. W tym celu kliknij katalog SSIS prawym przyciskiem myszy i wybierz opcję *Reports/Standard Reports/Integration Services Dashboard*. Następnie możesz przyrzeć się raportowi *Overview* z informacjami o wykonywaniu pakietu. Ten raport, przedstawiony na rysunku 20.13, pozwala zobaczyć, jakie parametry zostały przekazane do pakietu, a także zapoznać się z informacjami na temat wydajności.

Microsoft SQL Server 2014

Overview

on SQL2014 at 11/24/2013 4:16:23 PM

This report provides an overview of the package tasks and parameters, including execution or validation information.

[View Messages](#)

[View Performance](#)

Execution Information

Operation ID	6	Duration (sec)	6.875
Package	MySSISProject\SSIS\LoadClientData.dtsx	Start Time	11/24/2013 4:16:12 PM
Environment	-	End Time	11/24/2013 4:16:19 PM
Status	Succeeded	Caller	SHIELD\Administrator

Execution Overview

[Filter:](#) Result: All (3 more)

Result	Duration (sec)	Package Name	Task Name	Execution Path
Succeeded	5.484	LoadClientData.dtsx	LoadClientData	LoadClientData
Succeeded	0	LoadClientData.dtsx	FSYS Archive File	LoadClientData\FSYS Archive File
Succeeded	5.437	LoadClientData.dtsx	SQL Update Client	LoadClientData\SQL Update

Parameters Used

Name	Value	Data Type
CALLER_INFO		String
DUMP_EVENT_CODE	0	String
DUMP_ON_ERROR	False	Boolean
DUMP_ON_EVENT	False	Boolean
LOGGING_LEVEL	1	Int32

Rysunek 20.13. Raport z informacjami o wykonywaniu pakietu

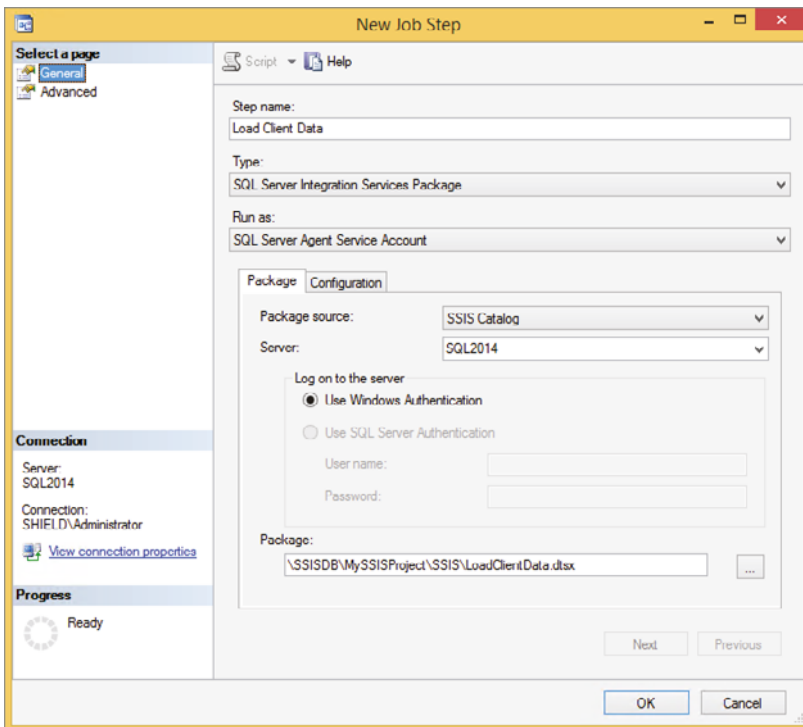
Planowanie wykonywania za pomocą narzędzia SQL Server Agent

Możliwość automatyzacji wykonywania pakietów SSIS jest potrzebna. Choć istnieje wiele programów do automatycznego wykonywania zadań zgodnie z harmonogramem, tu zobaczysz, jak zautomatyzować uruchamianie pakietów za pomocą narzędzia SQL Server Agent.

Zacznij od utworzenia zadania. Dodaj do niego przynajmniej jeden krok typu *SQL Server Integration Services Package*. Możesz też skonfigurować inne opcje, np. wysłanie na e-mail powiadomień o zakończeniu, powodzeniu lub niepowodzeniu zadania. Ponadto warto skonfigurować alerty, aby wysłać powiadomienia po wystąpieniu określonych zdarzeń systemu SQL Server, warunków związanych z wydajnością lub zdarzeń WMI. Wiele tego typu aspektów administrator może skonfigurować za pomocą środowisk.

Aby ustawić uruchamianie pakietu przez narzędzie SQL Server Agent, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z systemem SQL Server.
2. W oknie *Object Explorer* rozwiń węzeł *SQL Server Agent*.
3. W węźle *SQL Server Agent* kliknij prawym przyciskiem myszy katalog *Jobs* i wybierz opcję *New Job*.
4. W zakładce *General* okna dialogowego *New Job* podaj nazwę, właściciela, kategorię i opis zadania.
5. W zakładce *Steps* okna dialogowego *New Job* kliknij przycisk *New* (znajdziesz go w dolnej części okna).
6. W oknie dialogowym *New Job Step*, pokazanym na rysunku 20.14, podaj nazwę kroku i wybierz typ *SQL Server Integration Services Package*. Ponadto w sekcjach dotyczących usług SSIS podaj informacje potrzebne do uruchamiania pakietów. Sekcje zawierają opcje bardzo podobne do tych z narzędzia DTEXECUI i okna dialogowego *Package Execution*. Źródło pakietu (opcja *Package source*) w modelu z instalowaniem projektów ustawiane jest na katalog SSIS, a w modelu z instalowaniem pakietów — na system SQL Server, system plików lub magazyn pakietów SSIS. Następnie podaj pakiet, który ma być wykonywany. W zakładce *Command line* możesz podejrzeć instrukcję wiersza poleceń używaną przez narzędzie SQL Server Agent do uruchamiania pakietu. Możesz porównać tę instrukcję z poleceniem wygenerowanym przez narzędzie DTEXECUI w czasie, gdy testowałeś wykonywanie pakietu.



Rysunek 20.14. Konfigurowanie kroku związanego z wykonywaniem pakietu

7. W zakładce *Advanced* okna dialogowego *New Job Step* możesz określić operacje uruchamiane po udanym wykonaniu kroku, liczbę ponawianych prób, odstęp między kolejnymi próbami i czynności wykonywane po niepowodzeniu kroku. Gdy klikniesz przycisk *OK*, aby zaakceptować konfigurację kroku, na stronie *Page* okna dialogowego *New Job* zobaczysz nowy krok. Po dodaniu dodatkowych kroków możesz zmienić ich kolejność.
8. Po zaakceptowaniu konfiguracji kroku możesz w oknie dialogowym *New Job* opcjonalnie ustawić harmonogram wykonywania pakietu, alerty, powiadomienia i docelowe serwery.

Uruchamianie pakietów za pomocą języka T-SQL

Aby uruchomić pakiet w języku T-SQL, możesz wykorzystać procedury składowane ze schematu `catalog`. Najpierw za pomocą procedury składowanej `catalog.create_execution` należy utworzyć jednostkę wykonawczą (ang. *execution*). To powoduje wygenerowanie identyfikatora GUID. Na podstawie identyfikatora można wywołać pakiet za pomocą procedury składowanej `catalog.start_execution`. Ponadto przy użyciu procedury składowanej `catalog.set_execution_parameter_value` można ustawiać parametry pakietu. Kompletną jednostkę wykonawczą przedstawia poniższy fragment kodu:

```
Declare @execution_id bigint
EXEC [SSISDB].[catalog].[create_execution] @package_name=N'2-OtherFeature.dtsx',
@execution_id=@execution_id OUTPUT,
@folder_name=N'EDW', @project_name=N'ExpeditionDenali',
@use32bitruntime=False, @reference_id=NULL
Select @execution_id
DECLARE @var0 sql_variant = N'localhost'
EXEC [SSISDB].[catalog].[set_execution_parameter_value] @execution_id,
@object_type=30, @parameter_name=N'ServerName',
@parameter_value=@var0
DECLARE @var1 smallint = 1
EXEC [SSISDB].[catalog].[set_execution_parameter_value] @execution_id,
@object_type=50, @parameter_name=N'LOGGING_LEVEL',
@parameter_value=@var1
EXEC [SSISDB].[catalog].[start_execution] @execution_id
GO
```

Zabezpieczanie usług SSIS

Opanowałeś już większość istotnych zadań administracyjnych związanych z pakietami. Wiesz już, jak tworzyć rozwiązania SSIS, zarządzać nimi, instalować je i wykonywać. Ponadto zapoznałeś się z podstawowymi zadaniami administracyjnymi dotyczącymi usługi SSIS. W tym podrozdziale omawiamy zabezpieczenia dostępne dla usług SSIS.

Przegląd zabezpieczeń usług SSIS

Usługi SSIS (podobnie jak cały system SQL Server) wykorzystują warstwy zabezpieczeń oparte na różnych mechanizmach. Pozwala to chronić zarówno kod pakietów, jak i mechanizmy zarządzania nimi i ich wykonywania. W modelu z instalowaniem pakietów zabezpieczenia usług SSIS są używane po stronie klienta i po stronie serwera. Stosowane są tu następujące rozwiązania.

- Poziomy ochrony pozwalające zaszyfrować lub usunąć z pakietów poufne informacje.
- Poziomy ochrony za pomocą hasła pozwalające chronić wszystkie dane lub tylko poufne informacje.
- Ograniczanie dostępu do pakietów za pomocą ról.
- Blokowanie dostępu do lokalizacji, w których przechowywane są pakiety.
- Podpisywanie pakietów za pomocą certyfikatów.

W pakietach SSIS jako poufne traktowane są np. hasła i łańcuchy znaków połączenia. Nie można zdefiniować, jakie dane usługi SSIS mają traktować jako poufne (chyba że zastosujesz niestandardowe zadanie).

W usługach SSIS za poufne uznaje się następujące informacje:

- hasło w łańcuchu znaków połączenia (przy ustawieniu Sensitive) lub całe łańcuchy znaków połączenia (przy ustawieniu All),
- wygenerowane przez zadania węzły XML oznaczone przez SSIS jako poufne,
- zmienne oznaczone przez SSIS jako poufne.

W modelu z instalowaniem projektów sytuacja jest prostsza. Gdy instalujesz pakiet w bazie danych z katalogiem, za szyfrowanie odpowiada ta baza. Następnie można zabezpieczyć pakiet za pomocą ról.

Zabezpieczenia w modelu z instalowaniem pakietów

Istnieją dwa podstawowe sposoby zabezpieczania pakietów SSIS — ustawienie poziomu ochrony i skonfigurowanie odpowiednich ról dotyczących baz danych związanych z usługami SSIS. W następnych podpunktach omawiamy oba te mechanizmy.

Poziomy ochrony pakietu

Wiele firm przechowuje w pakietach SSIS poufne informacje, dlatego chce kontrolować, gdzie znajdują się te dane. Pakiety czasem zawierają hasła, które w rękach nieodpowiednich osób mogą prowadzić do wykradzenia tajnych danych.

W usługach SSIS do zabezpieczenia się przed takimi problemami służą poziomy ochrony pakietów. Do kontrolowania informacji o lokalizacji danych (np. łańcuchów znaków połączeń) służą poziomy EncryptSensitive. Ponadto przy stosowaniu poziomów EncryptAll ustawiane są hasła. Dzięki temu otwierać i wykonywać pakiety mogą tylko osoby znające właściwe hasła.

Oto poziomy ochrony pakietów dostępne w usługach SSIS:

- DontSaveSensitive,
- EncryptAllWithUserKey i EncryptSensitiveWithUserKey,
- EncryptAllWithPassword i EncryptSensitiveWithPassword,
- ServerStorage (wykorzystanie magazynu serwera przy szyfrowaniu; dostępne tylko dla pakietów z magazynu systemu SQL Server).

Poziomy ochrony pakietów początkowo ustawiane są za pomocą narzędzia SSDT. Później można je zmienić po zainstalowaniu albo w trakcie importowania lub eksportowania pakietów za pomocą programu SQL Server Management Studio. Ponadto poziomy można modyfikować przy kopiowaniu pakietów z narzędzia SSDT w inne miejsce. Jest to dobry kompromis między

programistami a administratorami, ponieważ programista może ustawić poziomy pod kątem prac nad kodem, a administratorzy mogą zmodyfikować je z uwzględnieniem standardów bezpieczeństwa ze środowiska produkcyjnego.

Role dotyczące baz danych w modelu z instalowaniem pakietów

Jeśli instalujesz pakiety w systemie SQL Server (w bazie msdb), powinieneś je chronić. Zabezpieczeniami są tu, podobnie jak w innych bazach, role. Aby kontrolować dostęp do pakietów, można ustawić trzy role dotyczące bazy msdb: `db_dtsadmin`, `db_dtsltduser` i `db_dtsoperator`.

Role są przypisywane pakietom w programie SQL Server Management Studio. Informacje o przyznanych rolach są zapisane w bazie msdb w kolumnach `readerrole`, `writerrole` i `ownersid` tabeli `sysssispackages`. Kolumna `readerrole` informuje o rolach mających dostęp w trybie odczytu do danego pakietu, kolumna `writerrole` informuje o rolach z dostępem w trybie zapisu, a kolumna `ownersid` zawiera dane o roli użytej przy tworzeniu pakietu.

Aby przypisać role zapewniające dostęp do odczytu i do zapisu pakietom w modelu z instalowaniem pakietów, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z serwerem usług SSIS.
2. W oknie *Object Explorer* wybierz katalog *Stored Packages* i rozwiń odpowiedni podkatalog.
3. W podkatalogu kliknij wybrany pakiet prawym przyciskiem myszy, aby przypisać role.
4. W oknie dialogowym *Packages Roles* wybierz rolę z uprawnieniami do odczytu z listy *Reader Role* i rolę z uprawnieniami do zapisu z listy *Writer Role*.

W modelu z instalowaniem projektów potrzebne kroki są niemal identyczne. Jedyna różnica polega na tym, że należy skonfigurować projekt zamiast pakietów. Kliknij projekt prawym przyciskiem myszy, wybierz opcję *Properties*, a następnie przejdź do zakładki *Permissions*. Ponieważ wszystkie pakiety w tym modelu są przechowywane w egzemplarzu bazy danych, możesz uruchamiać je za pomocą loginów z uwierzytelnianiem z systemu SQL Server.

Jeśli domyślnie przypisane do ról uprawnienia do wykonywania i aktualizowania pakietów nie spełniają Twoich potrzeb z obszaru bezpieczeństwa, utwórz role zdefiniowane przez użytkownika. Aby zdefiniować takie role, nawiąż połączenie z egzemplarzem systemu SQL Server i rozwiń węzeł *Roles* bazy msdb. Kliknij ten węzeł prawym przyciskiem myszy i wybierz opcję *New Database Role*. Po dodaniu nowej roli dotyczącej bazy msdb musisz ponownie uruchomić daną usługę, aby zastosować tę rolę.

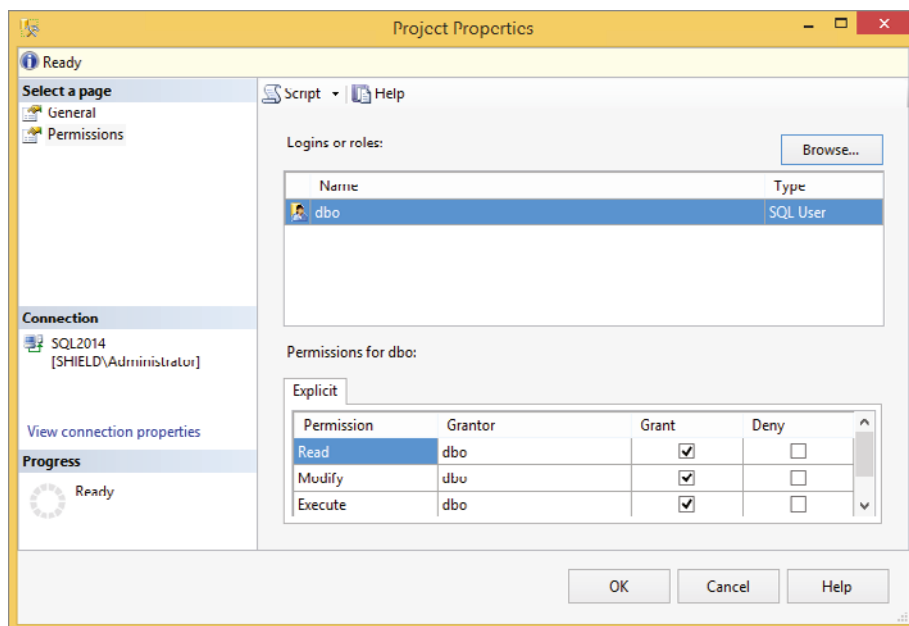
Role dotyczące baz danych pomagają skonfigurować tabelę `sysssispackages` w bazie msdb z wykorzystaniem opcji umożliwiających dostęp w trybie do odczytu i do zapisu do konkretnych pakietów. Pozwala to zapewnić bezpieczeństwo na poziomie serwera, bazy danych i tabeli. Zabezpieczenia omawiane w tym punkcie są dostępne tylko wtedy, gdy pakiety zostały zapisane w systemie SQL Server (w bazie msdb).

Role dotyczące baz danych w modelu z instalowaniem projektów

Jeśli instalujesz pakiety za pomocą modelu z instalowaniem projektów w katalogu usług SSIS, możesz zabezpieczyć pakiety na poziomie projektu. W tym celu wykonaj następujące kroki.

1. Aby skonfigurować uprawnienia, kliknij projekt prawym przyciskiem myszy w programie SQL Server Management Studio i wybierz opcję *Properties*.

- Otwórz stronę *Permissions* (przedstawioną na rysunku 20.15) i kliknij przycisk *Browse*, aby przyznać uprawnienia nowej roli lub użytkownikowi systemu SQL Server zapisanemu w bazie SSISDB. Warto zauważyć, że domyślnie każdy użytkownik z bazy SSISDB mający rolę *dbo* automatycznie posiada wszystkie uprawnienia do projektu.



Rysunek 20.15. Ustawianie uprawnień do projektu

Możesz przyznać użytkownikowi uprawnienia do odczytu, jeśli chcesz, aby mógł tylko wyświetlić pakiet i sprawdzić metadane pakietu z projektu. Uprawnienia do wykonywania pozwalają użytkownikowi (lub osobom z przypisaną daną rolą) uruchamiać pakiet z projektu. Uprawnienia do modyfikowania służą do zastępowania projektu i konfigurowania go.

Podsumowanie

Znasz już liczne funkcje administracyjne związane z usługami SSIS. Istnieją dwa tryby instalowania pakietów SSIS: model z instalowaniem pakietów i model z instalowaniem projektów. Drugi z tych modeli pozwala zainstalować cały projekt i udostępnić większość nowych mechanizmów usług SSIS (np. obsługę parametrów). Model z instalowaniem pakietów działa podobnie jak w systemach SQL Server 2005 i 2008. W modelu z instalowaniem projektów pakiety można łatwo skonfigurować za pomocą środowisk, które pozwalają uruchamiać pakiety z zestawem zmiennych dostosowanych do danego użytkownika.

W rozdziale 21., „Zarządzanie usługami Analysis Services i dostrajanie ich wydajności”, poznasz podobne funkcje administracyjne dotyczące usług Analysis Services.

Zarządzanie usługami Analysis Services i dostrajanie ich wydajności

ZAWARTOŚĆ ROZDZIAŁU:

- Omówienie dwóch typów serwerów SSAS.
- Zarządzanie właściwościami serwerów SSAS.
- Dostrajanie wydajności usług SSAS.

Wiesz już, jak zarządzać usługami SSIS. Ten rozdział to kontynuacja omawiania zarządzania narzędziami BI. Dowiesz się tu, jak administratorzy mogą zarządzać usługami SSAS. Koncentrujemy się na codziennych zadaniach administratorów bazy danych, a nie na różnych szczegółowych operacjach wykonywanych przez programistów.

Rozdział zaczynamy od krótkiego przeglądu usług SSAS, zatem będziesz miał okazję do zapoznania się z aspektami zarządzania tymi usługami i ich optymalizowania. Dalej opisujemy zarządzanie serwerem SSAS. Przedstawiamy m.in. ustawienia serwera i wymagane usługi. Dowiesz się też, jak wykonywać różne zadania administracyjne (np. przenosić bazy SSAS ze środowiska rozwojowego do produkcyjnego lub tworzyć kopie zapasowe takich baz) za pomocą skryptów. Następnie omawiamy zarządzanie bazami SSAS (instalowanie, tworzenie kopii zapasowych, przywracanie i synchronizację) oraz monitorowanie wydajności usług SSAS.

Dalej przedstawiamy zarządzanie przechowywaniem danych (w tym tryby przechowywania i konfigurowanie partycji), a także projektowanie agregacji. Następnie zapoznasz się z zarządzaniem zadaniami przetwarzania, które służą do łączenia projektów z danymi.

UWAGA Omawianie stosowania usług SSAS z perspektywy programisty wykracza poza zakres tej książki, dlatego nie opisujemy tego zagadnienia. Więcej na ten temat dowiesz się z pozycji *Professional Microsoft SQL Server Analysis Services 2012 with MDX and DAX* (Sivakumar Harinath, Wiley, 2012, Indianapolis).

Przegląd usług SSAS

Aby lepiej zrozumieć różne istotne aspekty, którymi muszą zarządzać administratorzy, warto zacząć od krótkiego przeglądu usług SSAS. Główną ich zaletą dla firm jest udostępnianie przydatnych, ważnych i aktualnych informacji, które trudno uzyskać z innych źródeł (systemów ERP, systemów rachunkowości, systemów CRM, systemów zarządzania łańcuchem dostaw itd.). Jeśli usłyszysz określenie *samoobsługowe raporty*, rozmówcy prawdopodobnie używają systemu OLAP (ang. *Online Analytical Processing*). Usługi SSAS obsługują właśnie takie systemy.

Usługi SSAS działają w dwóch trybach — MOLAP (ang. *Multidimensional OLAP*) i w modelu tabelowym. Gdy instalujesz egzemplarz SSAS, musisz zdecydować, z którego trybu będziesz korzystać.

W trybie MOLAP punktem wyjścia jest schemat gwiazdy, na którym oparte są kostki. Tryb ten umożliwia drążenie danych i zapewnia maksymalną skalowalność. Aktualizowanie danych w tym modelu oferuje dodatkowe możliwości, takie jak przyrostowe wczytywanie informacji.

UWAGA Więcej informacji o modelach danych znajdziesz w książce *The Data Warehousing Toolkit: The Definitive Guide to Dimensional Modeling* (Ralph Kimball, Wiley, 2013, Indianapolis).

Model tabelowy jest łatwiejszy do opanowania i mniej kosztowny. Punktem wyjścia są tu dane (z hurtownią lub bez niej), na ich podstawie budowana jest kostka. To podejście jest bardzo proste w użyciu. Użytkownicy mogą budować kostki za pomocą Excela lub środowiska Visual Studio.

Niezależnie od tego, który model stosujesz, na zapleczu przy użyciu silnika SSAS tworzona jest kostka. Wybór trybu zależy przede wszystkim od sposobu aktualizowania danych. W modelu tabelowym w momencie aktualizowania kostki cała kostka lub partycja zostają opróżnione i ponownie zapełnione. W trybie MOLAP można aktualizować poszczególne partycje w całości lub przyrostowo.

W obu modelach ważne jest zoptymalizowanie silnika OLAP w taki sposób, aby możliwe było błyskawiczne pobieranie danych. Oto istotne korzyści stosowania opisywanej technologii.

- Jednolity dostęp do współużytkowanych danych z wielu źródeł z precyzyjnie ustawionymi zabezpieczeniami i możliwością zapisu danych.
- Błyskawiczne i wygodne zapisywanie oraz agregowanie bardzo dużych ilości danych.
- Wielowymiarowe widoki danych wykraczające poza tradycyjne widoki z wierszami i kolumnami.
- Zaawansowane obliczenia, które oferują większe możliwości i wyższą wydajność niż w silniku RDBMS.
- Zaawansowane techniki drążenia danych pomagające przewidywać przyszłe zdarzenia na podstawie danych historycznych z bazy.

Jaka jest rola administratora baz danych w kontekście usług SSAS? W tabeli 21.1 przedstawione są tradycyjne zadania administratorów baz danych i ich odpowiedniki w usługach SSAS.

Przedstawiliśmy podstawy trybów MOLAP i tabelowego; w następnych punktach lepiej zapoznasz się z każdym z tych modeli.

Tabela 21.1. Zadania administratora baz danych w obszarze usług SSAS

Zadania administratora baz z systemu SQL Server	Zadania w usługach SSAS
Tworzenie loginów	Tworzenie ról
Tworzenie indeksów	Konfigurowanie agregacji
Podział tabel na partycje	Podział grup faktów na partycje
Tworzenie kopii zapasowych baz danych	Tworzenie kopii zapasowych baz danych SSAS

Elementy modelu MOLAP

W usługach SSAS w modelu MOLAP tworzone są kostki. Kostka łączy tabele wymiarów i tabele faktów w jeden widok, po którym można się poruszać. Użytkownicy mogą zatem samodzielnie przeprowadzać analizy. Oto elementy występujące w modelu MOLAP.

- **Widok źródła danych.** Centralnym elementem modelu MOLAP jest logiczny schemat reprezentujący dane ze źródła w zrozumiałym i standardowym sposób. Schemat to widok źródła danych (ang. *data source view* — DSV). Izoluje on kostkę od zmian wprowadzanych w używanych źródłach danych.
- **Model wymiarów.** Model ten określa projekt kostki. Obejmuje fakty (potrzebne użytkownikom do wyciągania wniosków biznesowych) i wymiary (służą do ograniczania faktów do przydatnych kombinacji wartości).
- **Obliczenia (wyrażenia).** Często kostkę trzeba wzbogacić o dodatkowe obliczenia, aby zapewnić oczekiwaną wartość biznesową analiz. Obliczenia w modelu MOLAP są oparte na fragmentach kodu w języku MDX (ang. *Multi Dimensional Expression*). Język MDX jest dla kostek tym, czym język SQL dla baz danych. Za pomocą języka MDX pobierane są z kostki dane potrzebne do obsługi różnych żądań od użytkowników.
- **Standardowe i abstrakcyjne modele.** Istnieje wiele dodatkowych mechanizmów, które wzbogacają analizy użytkowników i ułatwiają generowanie raportów oraz poruszanie się po kostce. Do modelu oprócz obliczeń często dodaje się też inne funkcje, które nie występują w źródłach danych. W modelu MOLAP te funkcje to: tłumaczenia, aliasy nazw baz danych, widoki pozwalające uniknąć przeciążenia danymi i wskaźniki KPI (ang. *Key Performance Indicator*) umożliwiające szybkie sprowadzenie danych do zrozumiałych miar.
- **Konfiguracja administracyjna.** Po zaprojektowaniu i utworzeniu kostki istotne stają się administracyjne aspekty modelu MOLAP. W takim modelu często wykonywane są zadania administracyjne, takie jak konfigurowanie zabezpieczeń w kostce lub projektowanie schematu podziału na partycje, który zwiększy wydajność obsługi kwerend i przetwarzania danych.

Elementy modelu tabelowego

Model tabelowy obejmuje komponenty bardzo podobne do komponentów modelu MOLAP, ale w uproszczonej postaci. Użytkownik nie zaczyna tu od projektowania schematu, ale od importowania danych. Następnie na podstawie tych danych tworzy model. Dane mogą pochodzić z różnych źródeł: z programu Excel, zwykłych plików i niemal dowolnego źródła zgodnego z interfejsami OLE DB i ODBC. Oto wybrane elementy występujące w modelu tabelowym.

- **Połączenia.** Lista połączeń z danymi potrzebnych do utworzenia kostki.

- **Tabele.** Dane, z których budowana jest kostka.
- **Role.** Mechanizm, za pomocą którego administrator może zabezpieczyć kostkę i zapisać w niej dane.

Elementy architektury serwera SSAS

Po omówieniu podstaw modeli MOLAP i tabelowego pora przyjrzeć się elementom serwera SSAS. Serwer SSAS (aplikacja *msmdsrv.exe*) działa jako usługa systemu Windows i obejmuje procesor kwerend (w języku MDX, a także kwerend DMX — ang. *Data Mining Extensions* — używanych przy drążeniu danych), odbiornik XMLA oraz mechanizm XLMA (ang. *XML for Analysis*). Na poniższej liście dokładniej omawiamy te komponenty.

- **Procesor kwerend.** Procesor kwerend parsuje i przetwarza instrukcje, podobnie jak silnik przetwarzania kwerend w systemie SQL Server. Procesor odpowiada też za zapisywanie obiektów w pamięci podręcznej, przechowywanie obiektów z modelu MOLAP i ich danych, wykonywanie obliczeń, obsługę zasobów serwera i zarządzanie transakcjami.
- **XLMA.** To natywny protokół używany do komunikacji w usługach SSAS. Jest oparty na protokole SOAP. Wszystkie aplikacje klienckie muszą korzystać z protokołu XLMA do komunikowania się z usługami SSAS. Ważne jest to, że klienci, które chcą komunikować się z usługami SSAS, nie wymagają instalacji komponentu klienckiego, co było konieczne w starszych wersjach usług SSAS (np. w usługach PivotTable Services). Ponieważ XLMA jest oparty na protokole SOAP, działa optymalnie w środowiskach bez stałych połączeń i bezstanowych, wymagających dostępu wydajnego ze względu na czas i zasoby. W usługach SSAS oprócz tego protokołu dostępne są też rozszerzenia umożliwiające zarządzanie metadanymi, zarządzanie sesją i tworzenie blokad. Istnieją dwie metody przesyłania komunikatów XMLA do usług SSAS. Domyślnie używany jest protokół TCP/IP, a alternatywą jest wykorzystanie protokołu HTTP.
- **Odbiornik XMLA.** Odbiornik ułatwia i obsługuje komunikację między różnymi klientami oraz serwerem SSAS. Konfiguracja portu tego odbiornika jest zapisana w pliku *msmdsrv.ini* (domyślnie znajduje się on w katalogu *C:/Program Files/Microsoft SQL Server/MSAS12.0/MSSQLSERVER/OLAP/Config*). Wartość 0 w znaczniku <Port> w tym pliku oznacza, że serwer SSAS odbiera dane w portach TCP/IP 2383 (domyślny egzemplarz serwera SSAS) i 2382 (pozostałe egzemplarze).

Dla nazwanych egzemplarzy serwera SSAS można używać portów innych niż domyślne. Narzędzie SQL Server Browser śledzi porty wykorzystywane przez poszczególne egzemplarze i przekierowuje klienty, które podały samą nazwę egzemplarza (bez numeru portu). Zastosuj zaporę, aby ograniczyć użytkownikom dostęp do portów serwera SSAS z poziomu internetu.

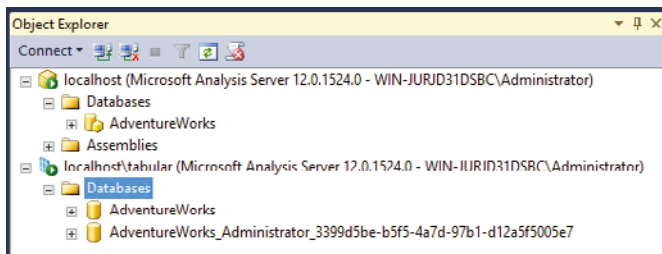
UWAGA Usługi SSAS czasem wymagają dużo pamięci RAM i zasobów wejścia-wyjścia. Jeśli są nieprawidłowo skonfigurowane, mogą pozbawić inne narzędzia (takie jak system SQL Server) zasobów. W dużych systemach (obejmujących setki gigabajtów danych) warto rozważyć utworzenie odizolowanego egzemplarza serwera SSAS na odrębnej fizycznej maszynie. Obserwuj zużycie pamięci, aby ustalić, czy należy przenieść serwer SSAS na osobny komputer.

Zarządzanie serwerem SSAS

W tym podrozdziale omawiamy ważne zadania administracyjne związane z egzemplarzami serwera SSAS. Dla usług SSAS używane są dwa narzędzia — SSDT do programowania i SQL Server Management Studio do administrowania. Tu najpierw omawiamy ustawienia konfiguracyjne z programu SQL Server Management Studio, a następnie szczegółowo opisujemy usługi potrzebne do działania serwera SSAS. W końcowej części znajdziesz wprowadzenie do języka ASSL (ang. *Analysis Services Scripting Language*) i jego zastosowań przy wykonywaniu zadań administracyjnych.

Zauważ, że gdy nawiążesz połączenie z serwerem SSAS w programie SQL Server Management Studio, ikony serwera wyglądają inaczej w zależności od typu egzemplarza. Należy połączyć się za pomocą uwierzytelniania z systemu Windows.

W oknie dialogowym widocznym na rysunku 21.1 znajdują się dwa egzemplarze serwera SSAS. Pierwszy (*localhost*) używa tradycyjnego modelu MOLAP. Drugi (*localhost\tabular*) to egzemplarz z modelem tabelowym. Trzeci typ (jego omawianie wykracza poza zakres tej książki) służy do obsługi arkuszy PowerPivot w programie SharePoint; dla tego typu trzeba używać nazwy POWERPivot; jego działanie bardzo przypomina działanie egzemplarzy z modelem tabelowym.



Rysunek 21.1. Różne typy serwerów SSAS w oknie Object Explorer

Zauważ, że na rysunku 21.1 w egzemplarzu *localhost\tabular* widoczne są dwie podobne bazy danych. Dzieje się tak, ponieważ w momencie otwarcia narzędzi programistycznych tworzony jest tymczasowy obszar roboczy związany z modyfikowaniem kostki w środowisku SSDT. Nazwa bazy danych używana dla tego obszaru roboczego kończy się nazwą użytkownika i identyfikatorem GUID. Dotyczy to tylko modelu tabelowego, ponieważ model MOLAP jest rozwijany w trybie offline i dopiero potem instalowany na serwerze.

Właściwości serwera

Właściwości serwera opisane w tym punkcie służą do konfigurowania działania egzemplarza serwera SSAS. Aby przejrzeć i zmodyfikować właściwości serwera, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio.
2. Nawiąż połączenie z serwerem SSAS za pomocą okna *Object Explorer*.
3. Kliknij serwer (węzeł główny) prawym przyciskiem myszy i wybierz opcję *Properties*.

W oknie *Properties* znajdują się dziesiątki właściwości, które pomogą dostosować dany egzemplarz serwera SSAS. Dodatkowe właściwości można wyświetlić za pomocą pola wyboru *Show Advanced Properties*. W tym punkcie omawiamy wybrane z ważnych właściwości serwera SSAS.

Właściwości dzienników

Jeśli starasz się ustalić, dlaczego przetwarzanie kwerendy trwa dłużej, niż oczekiwałeś, właściwości z tej grupy pomogą Ci uzyskać odpowiedź. Właściwości dzienników określają, kiedy i gdzie zdarzenia są rejestrowane. Ta grupa pozwala skonfigurować np. rejestrowanie błędów, rejestrowanie wyjątków, narzędzie Flight Recorder (opisane dalej w tym rozdziale), rejestrowanie kwerend i śledzenie zdarzeń.

Przykładowe właściwości z tej grupy to `QueryLog/QueryLogConnectionString` i `QueryLog/QueryLogTableName`, określające, gdzie rejestrowane są kwerendy (w jakiej bazie i w której tabeli). Właściwość `QueryLog/QueryLogConnectionString` określa egzemplarz systemu SQL Server i bazę danych, gdzie zapisany jest dziennik kwerend usług SSAS. Domyślnie po dodaniu tej właściwości i ustawieniu właściwości `CreateQueryLogTable` na `True` system SQL Server rozpoczyna rejestrowanie w tabeli co dziesiątej kwerendy. Jeśli egzemplarz serwera SSAS jest aktywny, rozważ zmniejszenie tej wartości do co setnej kwerendy.

Dziennik kwerend możesz wykorzystać później do dostrojenia systemu SQL Server za pomocą narzędzia Usage Based Optimization. Dziennik ten pomaga dopracować kostkę SSAS na podstawie wywoływanych w przeszłości kwerend.

Właściwości dotyczące pamięci

Właściwości z tej grupy określają, jak serwer wykorzystuje pamięć systemową. Właściwość `LowMemoryLimit` reprezentuje progową procentową wartość całej fizycznej pamięci. Przekroczenie tej wartości sprawia, że serwer zaczyna przywracać nieużywaną pamięć w celu jej zwolnienia. Domyślne ustawienie to 65% całej fizycznej pamięci. Właściwość `TotalMemoryLimit` określa, jaki procent całej fizycznej pamięci komputera można udostępnić serwerowi SSAS. Domyślnie jest to 80%.

Przy ustawieniach domyślnych serwer SSAS zajmuje od 65 do 80% pamięci komputera i nie zwraca jej, jeśli używa mniej niż 65%.

W modelu tabelowym właściwość `VertipaqMemoryLimit` określa, od jakiego poziomu serwer SSAS umożliwia stronicowanie. Stronicowanie jest możliwe tylko wtedy, gdy właściwość `VertipaqPagingPolicy` jest ustawiona na 1 (jest to właściwość zaawansowana, dostępna tylko wtedy, jeśli pole *Show Advanced Properties* jest zaznaczone). Domyślnie stronicowanie jest włączone.

Właściwości sieci

Właściwości z tej kategorii służą do kontrolowania używanych przez serwer zasobów sieciowych. Najważniejsze są ustawienia określające protokół używany przez odbiornik (IPv4 lub IPv6) oraz to, czy serwer zezwala na stosowanie formatu Binary XML w żądaniach i odpowiedziach. W systemach Windows Server 2008 R2 i nowszych oraz Windows 7 i nowszych domyślnie używany jest protokół IPv6.

Właściwości OLAP

Właściwości te kontrolują, w jaki sposób serwer przetwarza obiekty (kostki, wymiary i agregacje). Do tej kategorii należą też właściwości określające, jak serwer ma przetwarzać kwerendy. Niektóre z tych właściwości są przydatne do symulowania scenariuszy testowych. Możesz np. ustawić właściwości `IndexUseEnabled`, `UseDataSlice` i `AggregationUseEnabled`, aby sprawdzić różne scenariusze obsługi kwerend i ustalić, czy wybrane optymalizacje pozwalają poprawić wydajność.

Właściwości związane z bezpieczeństwem

Właściwości te służą do kontrolowania obsługi uprawnień przez serwer. Przykładową właściwością z tej grupy jest `RequireClientAuthentication`, określająca, czy klienci łączące się z serwerem wymagają uwierzytelniania. Jeśli ustawisz właściwość `BuiltInAdminsAreServerAdmins` na `False`, lokalni administratorzy komputera nie otrzymają pośrednio uprawnień administratora do egzemplarza serwera SSAS. Za pomocą tej właściwości można też przyznać podwyższone uprawnienia do serwera SSAS lokalnym administratorom i kontom usługowym.

Wymagane usługi

Usługi systemu Windows wymagane dla serwera SSAS to SQL Server Analysis Services, SQL Server Agent (tylko wtedy, kiedy chcesz uruchamiać zadania według harmonogramu) i SQL Server Browser. Usługa SQL Server Browser obsługuje mechanizm przekierowań serwera SSAS, używany, gdy klienci łączą się z nazwanymi egzemplarzami tego serwera.

Dla usług należy stosować konta o minimalnych uprawnieniach wystarczających do ich prawidłowego działania. Zwykle potrzebne jest konto z uprawnieniami do sieci, któremu należy przyznać uprawnienia dostępu do zdalnych zasobów (takie konto trzeba powiązać z daną usługą).

Język ASSL

Pomyśl, ile zadań administracyjnych można zautomatyzować za pomocą wbudowanego języka skryptowego. ASSL to język służący do automatyzowania takich zadań na serwerze SSAS. Język jest oparty na XML-u i umożliwia aplikacjom klienckim pobieranie danych z serwera SSAS.

ASSL jest podzielony na dwie odrębne części. Pierwsza z nich definiuje obiekty i właściwości będące częścią serwera, w tym obiekty używane przy tworzeniu rozwiązań (fakty i wymiary). Druga część służy do zgłaszania żądań wykonania przez serwer określonych zadań, np. przetwarzania obiektów lub operacji wsadowych.

Skoncentrujemy się teraz na aspektach języka ASSL pomocnych przy zarządzaniu serwerem SSAS. Zapoznamy Cię z przykładami, które ilustrują używanie tego języka do przetwarzania obiektów. W ten sposób można wypełnić obiekty danymi, co pozwala użytkownikom wykorzystać te obiekty w analizach biznesowych. Niektóre z obsługiwanych obiektów to kostki, bazy danych, wymiary i partycje. Do uruchamiania przetwarzania za pomocą tego języka służy instrukcja `Process`.

Oto przykładowy skrypt, który przetwarza wymiar `Employee` bazy `AdventureWorks`:

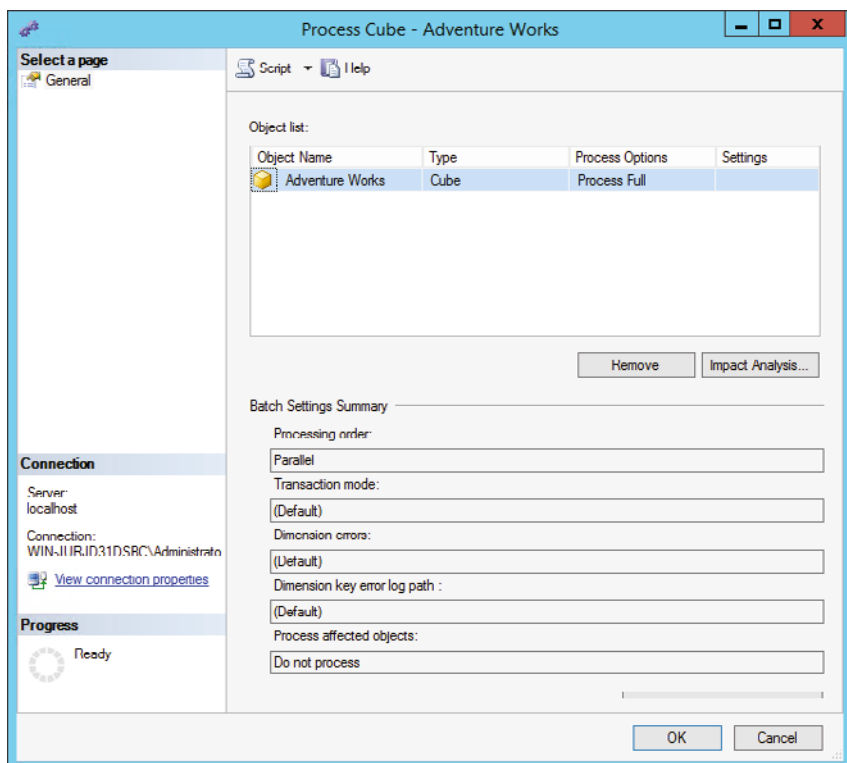
```
<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Parallel>
    <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
      xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2/2"
      xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/engine/100/100"
      xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/engine/200/200"
      xmlns:ddl300="http://schemas.microsoft.com/analysiservices/2011/engine/300"
```

```

xmlns:ddl300_300="http://schemas.microsoft.com/analysisservices/2011/engine/
300/300"
xmlns:ddl400="http://schemas.microsoft.com/analysisservices/2012/engine/400"
xmlns:ddl400_400="http://schemas.microsoft.com/analysisservices/2012/engine/400/
400">
<Object>
  <DatabaseID>AdventureWorks</DatabaseID>
  <DimensionID>Dim_Employee</DimensionID>
</Object>
<Type>ProcessUpdate</Type>
<WriteBackTableCreation>UseExisting</WriteBackTableCreation>
</Process>
</Parallel>
</Batch>

```

Wiele operacji wykonywanych w programie SQL Server Management Studio można zapisać w formie skryptu. Możesz np. wygenerować przedstawiony tu skrypt. W tym celu kliknij prawym przyciskiem myszy kostkę AdventureWorks i wybierz opcję *Process*. Pojawi się okno dialogowe *Process Cube* widoczne na rysunku 21.2. W tym oknie kliknij przycisk *Script* (widoczny w górnej części okna, pod paskiem tytułu), a następnie wybierz miejsce, w którym chcesz zapisać skrypt.



Rysunek 21.2. W tym oknie możesz wygenerować skrypt

Opcje widoczne na tym ekranie na razie nie są istotne. Omawiamy je w punkcie „Przetwarzanie obiektów SSAS”, dalej w tym rozdziale.

Zarządzanie bazami z serwera SSAS

Skoro zapoznałeś się już z serwerem SSAS, pora przejść do zadań administracyjnych wykonywanych w bazach danych instalowanych i uruchamianych na takich serwerach. Głównym zadaniem w obszarze zarządzania bazami z serwera SSAS jest ich instalowanie na serwerze, przetwarzanie obiektów SSAS, wykonywanie operacji związanych z odzyskiwaniem systemu po katastrofie (tworzenie i odzyskiwanie kopii zapasowych) oraz synchronizowanie baz danych w celu ich kopiowania.

Instalowanie baz na serwerach SSAS

Bez zainstalowanych baz uruchamianie serwera SSAS nie ma sensu. Po zainstalowaniu bazy można wprowadzać na serwerze zmiany w jej projekcie.

Przy wykonywaniu zadań administracyjnych możesz albo wykorzystać program SQL Server Management Studio i bezpośrednio wprowadzać zmiany w bazie (w *trybie online*), albo zastosować narzędzie SSDT i modyfikować bazy w ramach ich budowania oraz instalowania (w *trybie offline*). Dostępne są następujące techniki:

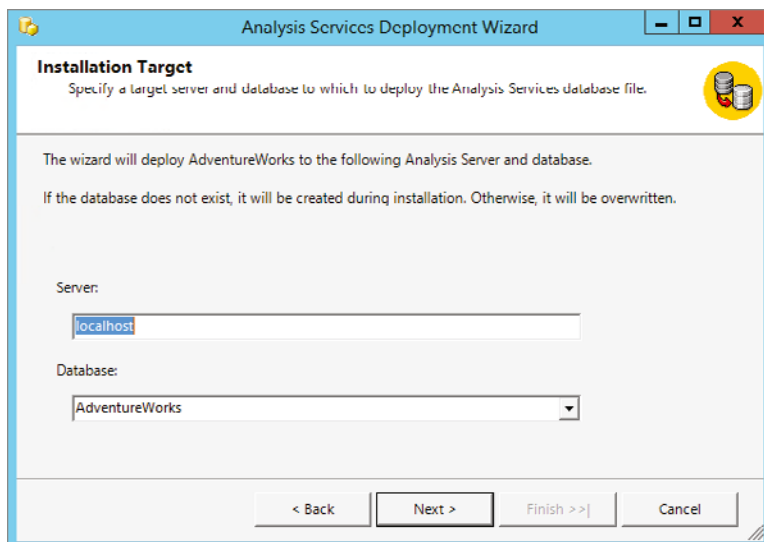
- wprowadzanie zmian bezpośrednio w narzędziu SSDT,
- zapisywanie zmian w skryptach i ich wykonywanie w programie SQL Server Management Studio,
- stopniowe wprowadzanie zmian przy użyciu kreatora Deployment Wizard,
- przetwarzanie zmian za pomocą kreatora Synchronize Database Wizard.

Niektóre z tych technik są przydatne tylko w określonych warunkach, dlatego nie poświęcamy im wiele miejsca. Najbardziej przydatna i kompletna metoda instalowania baz danych polega na użyciu kreatora Deployment Wizard. Drugim dobrym wyborem jest kreator Synchronize Database Wizard.

Główną zaletą kreatora Deployment Wizard jest to, że jako jedyny umożliwia umieszczenie definicji projektu bazy danych w środowisku produkcyjnym (lub dowolnym innym), a także pozwala zachować wiele ustawień (np. związanych z bezpieczeństwem i partycjami) z konfiguracji produkcyjnej bazy danych. Jest to ważne, ponieważ bezpośrednia instalacja z poziomu narzędzia SSDT i skrypty z programu SQL Server Management Studio nie umożliwiają zachowania istniejących ustawień.

Poniżej opisujemy działanie kreatora Deployment Wizard, co pomoże zrozumieć wartość tego narzędzia w zakresie instalowania baz danych.

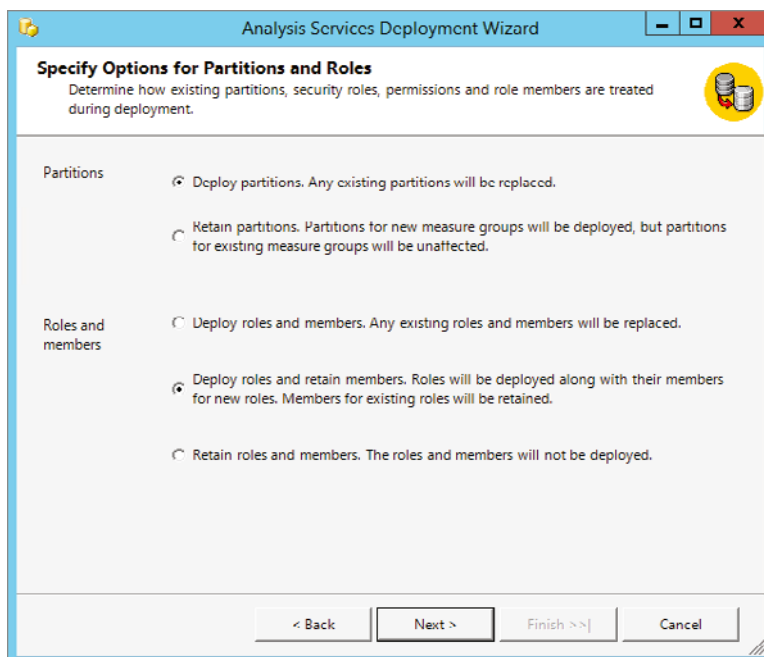
1. Otwórz menu *Start*, wybierz kategorię *Microsoft SQL Server 2014/Analysis Services* i uruchom kreator Deployment Wizard.
2. Na stronie *Specify Source Analysis Services Database* wpisz pełną ścieżkę do bazy danych SSAS. Ten plik (o rozszerzeniu *.asdatabase*) powinien zostać udostępniony przez programistę rozwiązań SSAS. Plik bazy może też znajdować się w katalogu projektu SSAS. Plik zawiera wszystkie metadane opisujące kostkę, zabezpieczenia i partycje, natomiast nie obejmuje samych danych.
3. Na stronie *Installation Target* (patrz rysunek 21.3) wskaż serwer, na którym należy zainstalować bazę danych. Podaj jej nazwę. Domyślnie używana jest nazwa pliku bazy danych; jeśli Ci ona nie odpowiada, zmień ją.



Rysunek 21.3. Ustawianie docelowego serwera

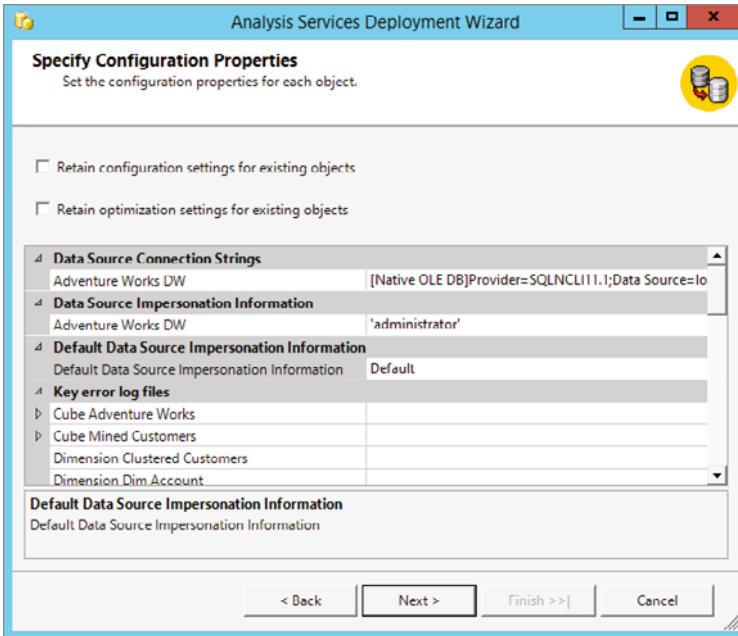
4. Na stronie *Specify Options for Partitions and Roles*, pokazanej na rysunku 21.4, określ, które opcje konfiguracyjne (*Partitions* lub *Roles and members*) mają zostać zachowane w docelowej bazie danych.

Wskazane opcje nie zostaną zastąpione. Ten ekran jest przydatny zwłaszcza wtedy, gdy wprowadzasz zmiany w rolach i partycjach w programie SQL Server Management Studio, a nie chcesz, aby pliki programisty zmodyfikowały ustawione przez Ciebie opcje.



Rysunek 21.4. Określanie, które ustawienia mają zostać zachowane

- Na stronie *Specify Configuration Properties* (patrz rysunek 21.5) określ, które ustawienia z pliku konfiguracyjnego (.configsettings) należy zastosować w docelowej bazie danych. Te ustawienia pozwalają np. zmienić łańcuchy znaków połączenia, aby wskazywały na produkcyjne źródła danych, a nie na źródła używane w trakcie programowania i testów. Pola *Retain...* w górnej części strony umożliwiają zarządzanie aktualizowaniem wcześniejszych instalacji, ponieważ wyłączają zmianę ustawień konfiguracyjnych i związanych z optymalizacją. Na tej stronie możesz też zmienić źródło danych i docelową lokalizację, w której zapisywane będą fizyczne pliki.

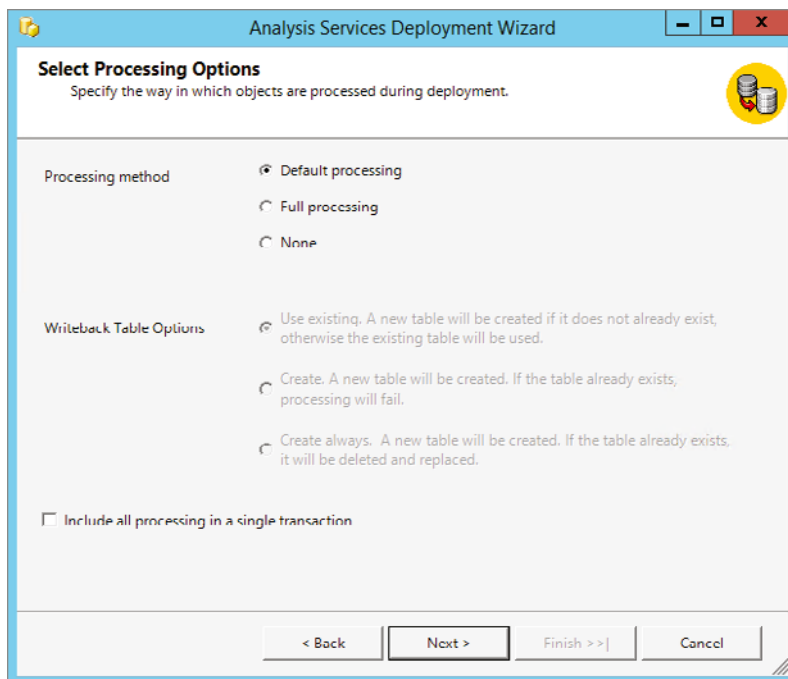


Rysunek 21.5. Określanie, które ustawienia mają zostać wprowadzane lub zachowane w docelowej bazie

- Na stronie *Select Processing Options*, przedstawionej na rysunku 21.6, ustaw metody przetwarzania i zmień opcje zapisu danych w tabeli wybrane przez programistę. Aby instalacja przebiegała sprawnie, możesz ustawić opcję powodującą przetwarzanie wszystkich operacji w jednej transakcji, co pozwala wycofać zmiany, jeśli na jednym z etapów instalacji wystąpią błędy. Opcja *Default processing* umożliwia usługom SSAS przegląd wprowadzanych modyfikacji i ustalenie optymalnej metody przetwarzania.
- Na stronie *Confirm Deployment* możesz zapisać cały proces instalacji w skrypcie. Jest to przydatne, jeśli użytkownik, który uruchomił kreator Deployment Wizard, nie ma uprawnień do zainstalowania bazy lub gdy proces instalacji trzeba zaplanować na inną godzinę, aby nie zakłócał innych zadań.

Przetwarzanie obiektów SSAS

Wiesz już, jak zainstalować bazy SSAS. Teraz trzeba zacząć ich przetwarzanie w celu dodania do nich danych. Jeśli ponadto kostki trzeba zaktualizować, aby uwzględnić w nich zmiany wprowadzone po instalacji, należy ponownie uruchomić przetwarzanie. Gdy w źródłach danych pojawią się zmiany, także potrzebne jest (minimalne) przyrostowe przetwarzanie kostki.



Rysunek 21.6. Konfigurowanie opcji przetwarzania

Pozwala to zagwarantować, że dane w rozwiązaniu SSAS będą aktualne. Programiści często budują i testują projekty lokalnie. Lokalny schemat z narzędzia SSDT trzeba najpierw zainstalować na serwerze, a dopiero potem można przejść do etapu przetwarzania.

Przetwarzanie w modelu MOLAP

Obiekty w modelu MOLAP, które wymagają przetwarzania, to: grupy faktów, partycje, wymiary, kostki, modele drażenia danych, struktury drażenia danych i bazy. Przetwarzanie odbywa się hierarchicznie; gdy przetwarzany jest obiekt obejmujący inne obiekty, także one zostają przetworzone.

Wyobraź sobie bazę, która obejmuje jedną lub więcej kostek zawierających jeden lub więcej wymiarów. Gdy uruchomisz przetwarzanie bazy, operacja ta będzie dotyczyć także wszystkich umieszczonych w niej kostek i wszystkich wymiarów zawartych lub używanych w tych kostkach. W modelu tabelowym trzeba ponownie przetworzyć tabelę w całości, a nie przyrostowo.

W następnych podpunktach wyjaśniamy, jak przetwarzać poszczególne obiekty. Jeśli używasz bazy SSAS o wielkości mniejszej niż 20 gigabajtów, możesz przetworzyć całą bazę i udostępnić ją, zanim użytkownicy będą potrzebować danych. W tym celu należy przeprowadzić pełne przetwarzanie bazy (czyli usunąć ją i ponownie załadować), co jest znacznie prostszą operacją niż przetwarzanie przyrostowe.

Przetwarzanie wymiarów

Usługi SSAS przetwarzają wymiary za pomocą kwerend, które zwracają dane ze źródłowych tabel do wymiarów. Te dane są porządkowane w hierarchie zapisywane w plikach odwzorowań, obejmujących listę wszystkich hierarchicznych ścieżek do każdego wymiaru. Przetwarzanie

wymiarów można zoptymalizować za pomocą odpowiednich indeksów klucza głównego lub innych atrybutów. Gdy przetwarzasz obiekty selektywnie, przed przystąpieniem do przetwarzania kostek lub partycji najpierw wykonaj tę operację dla wymiarów.

Przetwarzanie kostek

Kostka obejmuje grupy faktów i partycje. W połączeniu z wymiarami zapewniają one definicję danych kostki. W ramach przetwarzania kostki wywoływane są kwerendy pobierające elementy tabeli faktów i powiązane wartości. W efekcie każda ścieżka w hierarchii wymiarów przyjmuje określoną wartość.

Przetwarzanie partycji

Celem podziału na partycje w SSAS (podobnie jak przy partycjonowaniu baz danych) jest przyspieszenie obsługi kwerend i wykonywania zadań przez administratora. Proces ten polega na podziale dużych zbiorów danych na mniejsze pliki (zwykle obejmujące dane z określonego przedziału czasu). Istotną cechą takiego przetwarzania jest konieczność oszacowania wolnego miejsca na dysku i uwzględnienia ograniczeń struktur danych SSAS. Podział na partycje jest bardzo istotny, jeśli chcesz zapewnić szybką obsługę kwerend i wydajne przetwarzanie obiektów.

Ponowne przetwarzanie

Po zainstalowaniu bazy SSAS wiele zdarzeń sprawia, że trzeba ponownie przetworzyć wybrane lub wszystkie obiekty z bazy danych. Jest to konieczne np. po strukturalnych zmianach obiektów lub schematów, modyfikacjach w projektach agregacji lub aktualizacji danych obiektu.

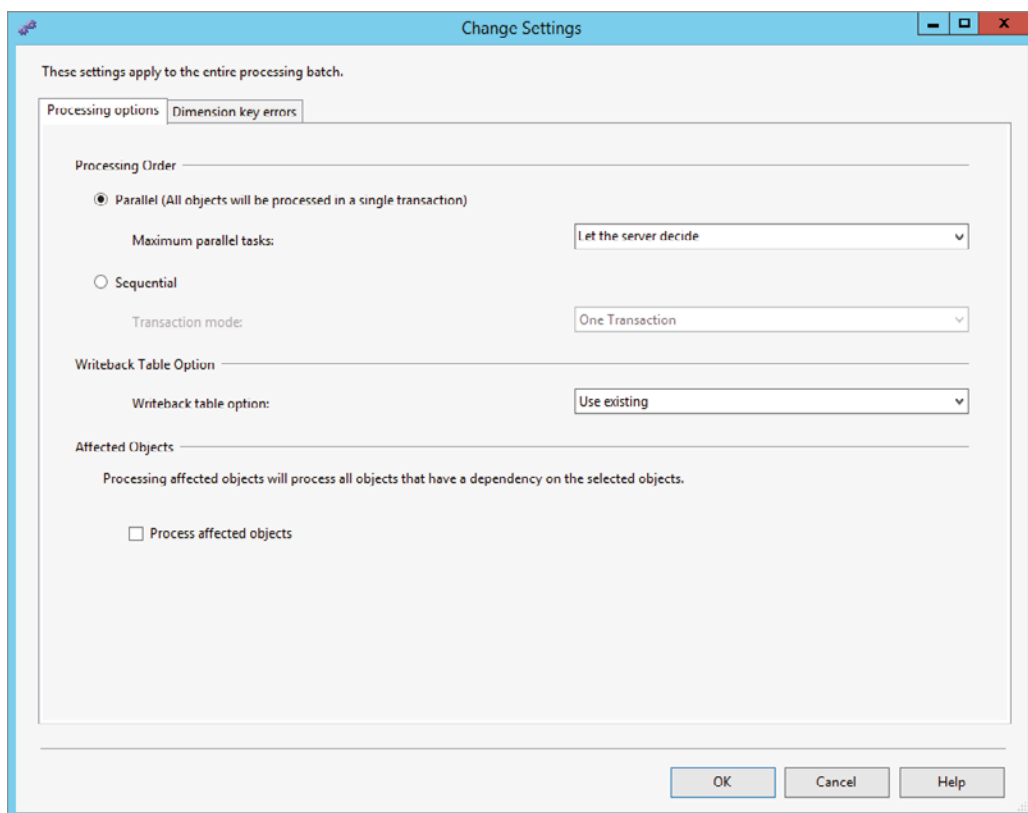
Uruchamianie przetwarzania

Aby rozpocząć przetwarzanie obiektów SSAS, można wykorzystać program SQL Server Management Studio lub SSDT albo wywołać skrypt XMLA. Ponadto można zastosować obiekty AMO (ang. *Analysis Management Objects*) do uruchamiania zadań przetwarzania za pomocą narzędzia programistycznych.

UWAGA Gdy używasz tych metod przy ręcznym przetwarzaniu, w momencie zatwierdzania obiektów SSAS nie są one dostępne przy obsłudze żądań użytkowników. Dzieje się tak, ponieważ na etapie zatwierdzania obiektów SSAS tworzona jest blokada na wyłączność. Żądania użytkowników nie są wtedy odrzucane, tylko umieszczane w kolejce do czasu udanego zatwierdzenia obiektów. Zamiast ręcznie przetwarzać kostkę, można też proaktywnie zapisać ją w pamięci podręcznej. Jest to zaawansowana technika, powodująca stopniowe zbieranie nowych danych i wczytywanie ich do nowej kostki z pamięci podręcznej; w tym samym czasie kwerendy są obsługiwane za pomocą pierwotnej kostki. Gdy przetwarzanie zostanie zakończone, nowa kostka jest udostępniana użytkownikom, a stara zostaje usunięta. W tym podejściu kostka pozostaje dostępna w trakcie jej przetwarzania, a zmiany są automatycznie udostępniane użytkownikom po krótkiej chwili od wprowadzenia modyfikacji w hurtowni danych.

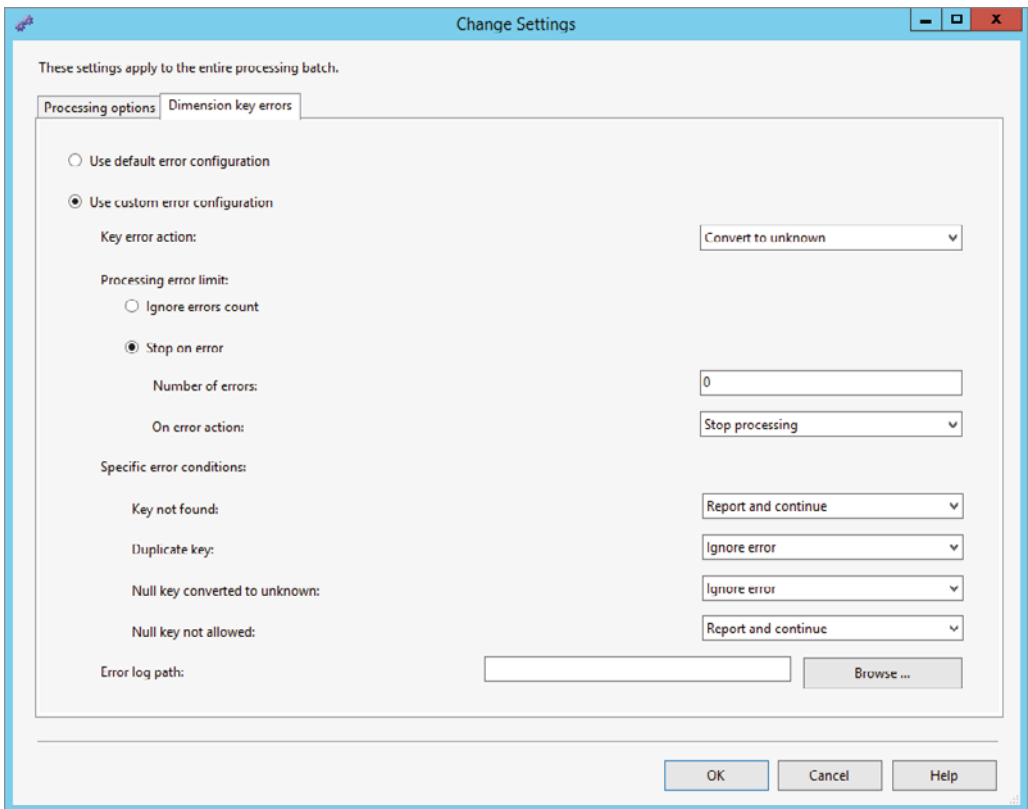
Aby uruchomić przetwarzanie bazy SSAS w programie SQL Server Management Studio, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z serwerem SSAS.
2. Kliknij bazę SSAS prawym przyciskiem myszy i wybierz opcję *Process*. Pojawi się okno dialogowe *Process Database*, w którym możesz skonfigurować proces przetwarzania (patrz rysunek 21.2).
3. Kliknij przycisk *Impact Analysis*, aby ocenić wpływ przetwarzania na powiązane obiekty. Aby np. w pełni przetworzyć wymiar, trzeba także wykonać tę operację dla powiązanych z nim grup faktów.
4. Kliknij przycisk *Change Settings*, by skonfigurować opcje przetwarzania (np. kolejność operacji). Służący do tego ekran jest widoczny na rysunku 21.7. Przy określaniu kolejności operacji można ustawić równoległe przetwarzanie obiektów w jednej transakcji albo sekwencyjne przetwarzanie obiektów w jednej lub wielu transakcjach. Ważną opcją jest *Process affected objects*, która kontroluje, czy należy przetworzyć także wszystkie obiekty zależne od danej bazy. W hurtowniach danych często stosuje się współużytkowane wymiary. Są one stosowane w całej firmie, co ułatwia ich konserwację oraz zwiększa jednolitość danych. W tym podejściu ustawienie *Process affected objects* może mieć istotny wpływ na systemy, jeśli wymusi ponowne przetwarzanie licznych innych baz danych, w których występują określone współużytkowane wymiary.



Rysunek 21.7. Ustawianie opcji przetwarzania

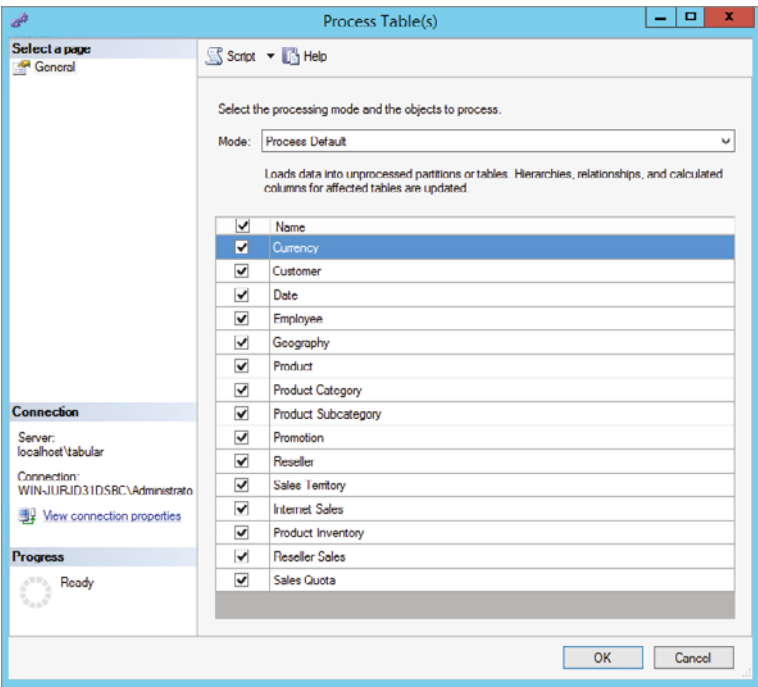
5. Możesz też skonfigurować zaawansowaną obsługę błędów kluczy wymiarów. W tym celu otwórz zakładkę *Dimension key errors*, widoczną na rysunku 21.8. Możesz np. zastosować niestandardową konfigurację obsługi błędów, aby przekształcać błędy kluczy na nieznanne rekordy, zamiast zatrzymywać przetwarzanie. Ponadto możesz ustawić limit błędów i określić, jakie operacje należy wykonać po jego przekroczeniu. Dostępne są też opcje do ustawiania obsługi konkretnych błędów (takich jak *Key not found* lub *Duplicate key*). Możliwości to zgłoszenie błędu i kontynuowanie przetwarzania, zignorowanie błędu i kontynuowanie przetwarzania oraz zgłoszenie błędu i zatrzymanie przetwarzania. Występowanie takich błędów zwykle oznacza problemy z danymi. Należy wtedy wrócić do etapu operacji ETL i naprawić usterki, ponieważ używanie technik ustawianych na tej stronie tylko ukrywa problemy.



Rysunek 21.8. Konfigurowanie obsługi błędów

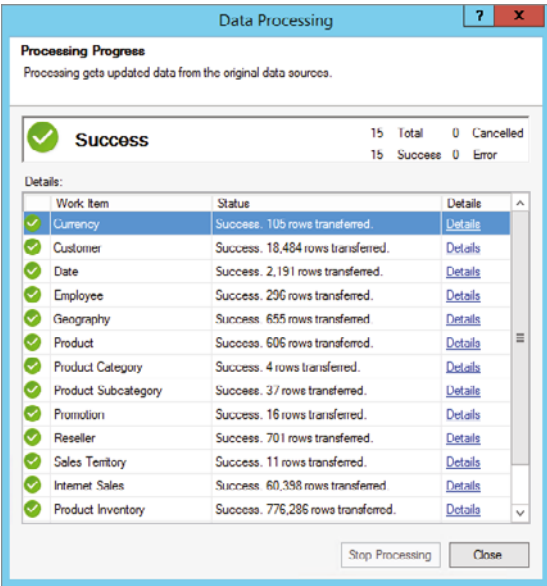
Przetwarzanie w modelu tabelowym

Choć uruchamianie przetwarzania w modelach tabelowym i MOLAP przebiega tak samo, na zapleczu wykonywane są inne operacje. W celu włączenia przetwarzania modelu tabelowego kliknij prawym przyciskiem myszy wybrany obiekt i wybierz opcję *Process in Management Studio*. Pojawi się wtedy okno widoczne na rysunku 21.9. W ten sposób możesz uzyskać dostęp do pierwotnych źródeł danych (systemu SQL Server, programów Access lub Excel itd.) oraz usunąć i wczytać tabele SSAS.



Rysunek 21.9. Przetwarzanie tabel w modelu tabelowym

Gdy klikniesz przycisk **OK**, rozpocznie się aktualizowanie tabel. Na rysunku 21.10 przedstawiono ekran z liczbą wierszy, które zostały zaktualizowane w poszczególnych tabelach. Jeśli źródła danych zostały przeniesione (np. bazę Accessa umieszczono w innym katalogu), możesz zmienić prowadzące do nich ścieżki w węźle *Data Sources* w programie SQL Server Management Studio.



Rysunek 21.10. Informacje na temat aktualizacji danych

Tworzenie i przywracanie kopii zapasowych baz SSAS

Tworzenie i przywracanie kopii zapasowych to zadania często wykonywane przez każdego administratora baz danych. Kopia zapasowa bazy SSAS obejmuje stan bazy i jej obiektów z określonego punktu w czasie. Taka kopia jest zapisywana w systemie plików w pliku o rozszerzeniu *.abf*. W ramach odzyskiwania na serwerze na podstawie pliku z kopią zapasową przywracany jest stan bazy i jej obiektów. Tworzenie i przywracanie kopii zapasowych jest więc przydatne przy odzyskiwaniu danych po wystąpieniu problemów z bazą na serwerze. Technika ta pozwala też przeprowadzić kontrolę stanu bazy danych. Choć dla baz SSAS można utworzyć kopie zapasowe w postaci zwykłych plików fizycznych, kroki opisane w tym punkcie powodują utworzenie jednego niezawodnego pliku, który jest zaszyfrowany i skompresowany.

OSTRZEŻENIE W opisywanych tu kopiach zapasowych umieszczana jest tylko zawartość baz SSAS, a nie źródła danych używane do zapełnienia tych baz. Dlatego obok kopii zapasowych baz SSAS powinieneś tworzyć zwykłe kopie zapasowe źródeł danych (w bazie lub systemie plików), aby uzyskać rzeczywisty stan zarówno obiektów SSAS, jak i ich źródeł z tego samego lub podobnego czasu.

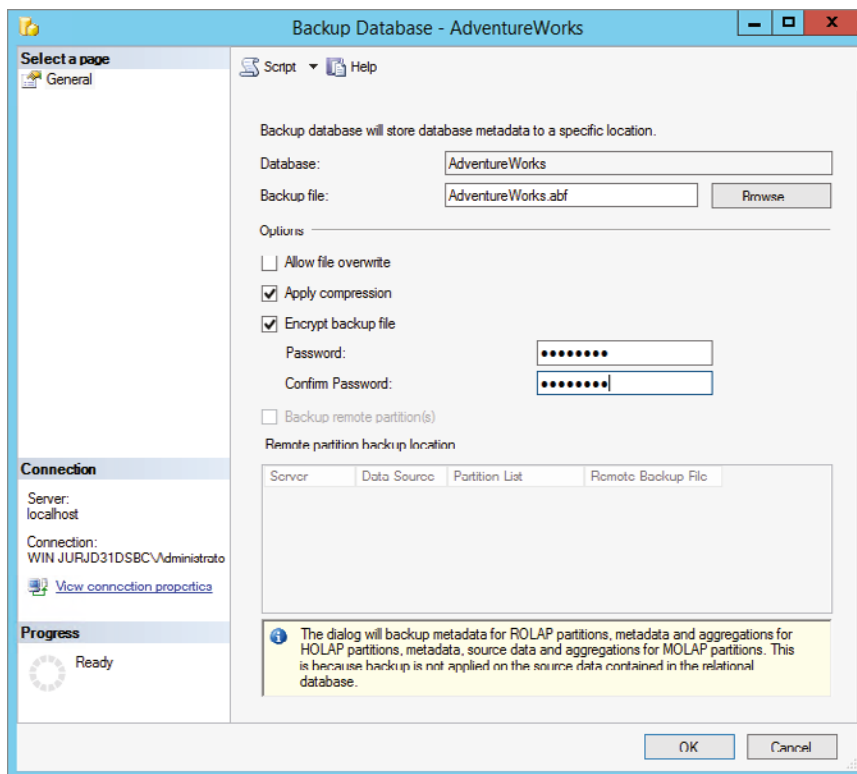
Informacje zapisane w kopiach zapasowych zależą do sposobu przechowywania bazy danych. Szczegółowy komunikat wyświetlany w dolnej części okna dialogowego *Backup Database* informuje o różnych obiektach umieszczonych w kopii zapasowej. Oto obiekty, które można zapisać w kopii zapasowej:

- metadane definiujące wszystkie obiekty,
- obliczone agregacje,
- źródła danych używane do zapełniania obiektów.

Poniżej opisujemy, jak tworzyć kopie zapasowe baz SSAS. Możesz wykorzystać program SQL Server Management Studio do pomocy przy konfigurowaniu zadań, a następnie zapisać operacje w formie skryptu, co pozwala realizować je według harmonogramu. Wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z serwerem SSAS.
2. Kliknij bazę SSAS prawym przyciskiem myszy i wybierz opcję *Backup*. Pojawi się okno dialogowe *Backup Database*, przedstawione na rysunku 21.11. W tym miejscu możesz skonfigurować proces tworzenia kopii zapasowej, np. włączyć kompresję, określić lokalizację pliku z kopią zapasową i ustawić szyfrowanie. Sposoby przechowywania baz omawiamy dalej. W dolnej części okna dialogowego dokładnie opisane jest, jakie dane znajdują się w kopii zapasowej. W kopii zapisywane są tylko dostępne w bazie elementy związane z usługami SSAS (partycje, metadane, dane źródłowe i agregacje). To, jakie elementy są dostępne, zależy od sposobu przechowywania bazy.
3. Możesz też zapisać proces tworzenia kopii w skrypcie. W tym celu kliknij przycisk *Script* widoczny w górnej części okna dialogowego. Wynikowy skrypt wygląda tak, jak poniżej (włącznie z ukrytym tu hasłem):

```
<Backup xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Object>
    <DatabaseID>AdventureWorks</DatabaseID>
  </Object>
  <File>AdventureWorks.abf</File>
  <Password>hasło</Password>
</Backup>
```



Rysunek 21.11. Konfigurowanie procesu tworzenia kopii zapasowych

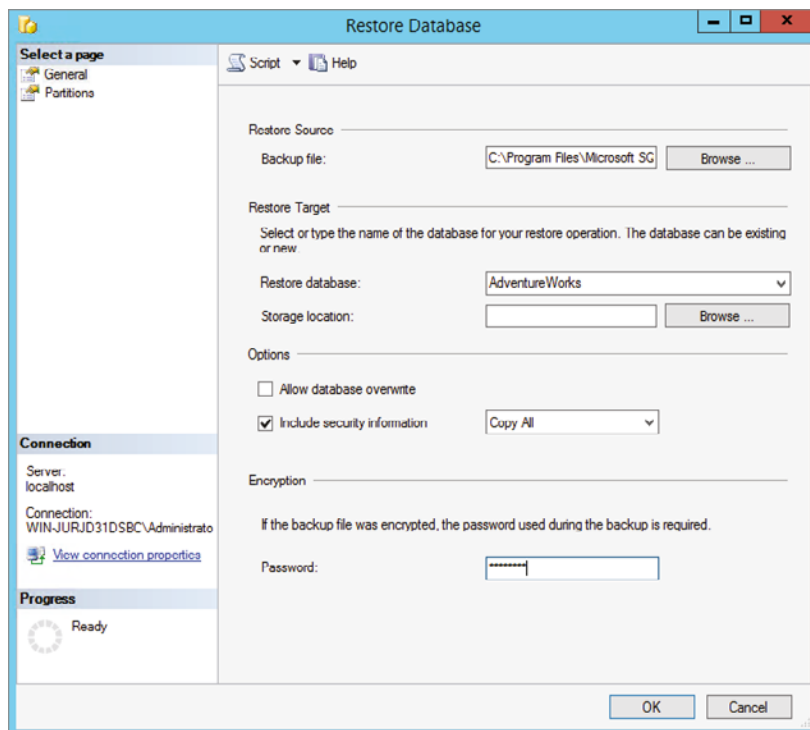
4. Ten skrypt jest automatycznie otwierany w nowym oknie kwerend XMLA w programie SQL Server Management Studio. Możesz uruchomić ten skrypt (tak jak kod w językach MDX i T-SQL), skopiować go na inny serwer lub zachować w celu późniejszego użycia.

Po utworzeniu kopii zapasowej bazy SSAS pora przejść do odzyskiwania takich baz. W ramach odzyskiwania utworzony wcześniej plik kopii zapasowej (o rozszerzeniu *.abf*) jest przywracany jako baza SSAS. W tym procesie dostępnych jest kilka możliwości:

- wykorzystanie pierwotnej nazwy bazy (lub podanie nowej nazwy),
- zastąpienie istniejącej bazy,
- wykorzystanie istniejących informacji o zabezpieczeniach (lub ich pominięcie),
- zmiana katalogu przywracania dla poszczególnych partycji.

W celu odzyskania bazy danych wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z serwerem SSAS.
2. Kliknij bazę SSAS prawym przyciskiem myszy i wybierz opcję *Restore*. Pojawi się okno dialogowe *Restore Database*, pokazane na rysunku 21.12. W tym miejscu możesz skonfigurować przywracanie, np. ustawić uwzględnianie zabezpieczeń lub zastępowanie istniejącej bazy.



Rysunek 21.12. Konfigurowanie przywracania bazy danych

- Możesz też zapisać proces przywracania w skrypcie. W tym celu kliknij przycisk *Script* widoczny w górnej części przedstawionego okna dialogowego. Wynikowy skrypt wygląda tak, jak poniżej (także on obejmuje hasło, co nie jest dobrym rozwiązaniem):

```
<Restore xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <File>C:\Program Files\Microsoft SQL Server\MSAS12.MSSQLSERVER\OLAP\Backup\
    AdventureWorks.abf</File>
  <DatabaseName>AdventureWorks</DatabaseName>
  <Password>hasło</Password>
</Restore>
```

- Skrypt jest automatycznie otwierany w nowym oknie kwerend XMLA w programie SQL Server Management Studio. Możesz uruchomić ten skrypt (tak jak kod w językach MDX i T-SQL), skopiować go na inny serwer lub zachować w celu późniejszego użycia.

Możesz też kliknąć widoczną w lewej części okna dialogowego zakładkę *Partitions* i zmienić miejsce przechowywania poszczególnych partycji.

Synchronizowanie baz SSAS

Innym ważnym zadaniem jest synchronizowanie baz SSAS z różnych serwerów. Zwykle odbywa się to przy przenoszeniu bazy z serwera testowego lub serwera działu kontroli jakości na serwer produkcyjny. Ten mechanizm jest przydatny, ponieważ użytkownicy w trakcie synchronizacji cały czas mogą przeglądać dane z kostek ze środowiska produkcyjnego. Po zakończeniu synchronizacji i przetwarzania użytkownicy są automatycznie kierowani do zsynchronizowanej kopii bazy danych, a starsza wersja bazy jest usuwana z serwera.

Proces ten przebiega inaczej niż instalacja, ponieważ w ramach instalacji zwykle przetwarzane są wymiary i kostki. Niektóre techniki przetwarzania obiektów SSAS wymagają wyłączenia kostki, co sprawia, że do czasu zakończenia przetwarzania jest ona niedostępna dla użytkowników.

Proces synchronizacji (podobnie jak wiele innych zadań dotyczących baz danych) można uruchomić bezpośrednio w kreatorze — tu jest to kreator Synchronize Database Wizard — albo zapisać w skrypcie w celu późniejszego wywołania lub wykonywania według harmonogramu.

Aby zsynchronizować bazy SSAS z różnych serwerów, wykonaj następujące kroki.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z docelowym serwerem SSAS.
2. Na docelowym serwerze kliknij katalog z bazą prawym przyciskiem myszy i wybierz opcję *Synchronize*.
3. Na stronie *Select Databases to Synchronize* ustaw źródłowy serwer i bazę danych. Docelowym serwerem jest ten, na którym uruchomiłeś synchronizację.
4. Jeśli używane są partycje, na stronie *Specify Locations for Local Partitions* jako katalog źródłowy jest wyświetlany katalog z serwera z partycją lokalną. Jako katalog docelowy możesz ustawić ten, w którym chcesz umieścić zsynchronizowaną bazę danych.
5. Jeżeli stosujesz partycje, na stronie *Specify Locations for Remote Partitions* możesz zmienić katalog i serwer docelowy, aby określić, gdzie chcesz umieścić zsynchronizowaną bazę danych. Jeśli ponadto w danej lokalizacji znajdują się zdalne partycje, które trzeba uwzględnić przy synchronizacji, zaznacz opcję *Sync*.
6. Na stronie *Synchronization Options* ustaw opcje związane z zabezpieczeniami i określ, czy należy zastosować kompresję. Opcje zabezpieczeń pozwalają skopiować wszystkie definicje zabezpieczeń i informacje o użytkownikach przypisanych do ról, skopiować definicje zabezpieczeń i pominąć informacje o użytkownikach przypisanych do ról lub zignorować wszystkie dane.
7. Na stronie *Select Synchronization Method* możesz albo natychmiast wywołać synchronizację, albo wygenerować skrypt na potrzeby późniejszego wykonywania tej operacji według harmonogramu.

Przetwarzanie to jeden z obszarów, w których granice między programowaniem a administrowaniem produkcyjną bazą danych się zacierają. To samo dotyczy dostrajania wydajności kostki. Programista zwykle przeprowadza podstawowe dostrajanie, a administrator odpowiada za długoterminowe optymalizowanie wydajności za pomocą agregacji.

Monitorowanie i dostrajanie wydajności usług SSAS

Aby z powodzeniem korzystać z usług SSAS, trzeba nieustannie monitorować obsługę kwerend i przebieg innych procesów oraz wprowadzać odpowiednie poprawki w celu zwiększenia ich wydajności. Podstawowe narzędzia służące do wykonywania tych zadań to SQL Server Profiler, liczniki wydajności i Flight Recorder.

Monitorowanie zdarzeń SSAS

Istnieją dwa sposoby monitorowania systemu SQL Server i usług SSAS; są to narzędzie SQL Server Profiler i zdarzenia rozszerzone. Narzędzie SQL Server Profiler ma w przyszłości zostać usunięte z systemu SQL Server, dlatego lepiej stosować zdarzenia rozszerzone. Obie techniki pozwalają przechwytywać te same zdarzenia.

W rozdziale 12. szczegółowo opisano używanie narzędzia SQL Server Profiler. Tu dowiesz się, jak stosować je do monitorowania zdarzeń usług SSAS. W narzędziu SQL Server Profiler można podejrzeć, co serwer robi w trakcie przetwarzania obiektów i obsługi kwerend. Równie ważna jest możliwość zarejestrowania danych uzyskanych w ramach profilowania. Można je zapisać w tabeli bazy danych lub pliku, aby później przejrzeć albo odtworzyć w celu lepszego zrozumienia przebiegu operacji. Ponadto możesz wyświetlić zdarzenia obok liczników wydajności komputera lub usług SSAS, aby wykryć czynniki wpływające na wydajność systemu.

Należy się tu skupić na śledzeniu aktywności serwera SSAS i badaniu wydajności kwerend MDX przesyłanych na serwer w celu obsługi żądań użytkowników. Oto przydatne kategorie zdarzeń.

- Zdarzenia związane z poleceniami, pomagające zrozumieć rodzaje instrukcji wywoływanych w celu wykonywania zadań.
- Zdarzenia związane z wykrywaniem danych, dotyczące żądań metadanych o obiektach serwera. Są to np. zdarzenia z grupy Discovery Server State lub zdarzenia wykrywania otwartych połączeń.
- Zdarzenia związane z błędami i ostrzeżeniami, pokazujące, jakie błędy lub ostrzeżenia zostały zgłoszone przez dany egzemplarz serwera SSAS.
- Zdarzenia związane z kwerendami, przechwytyjące kwerendy przesłane na serwer SSAS.

Ponieważ ślad obejmuje wiele szczegółowych danych, użyj przycisku *Column Filter*, aby wyświetlić informacje dotyczące operacji z konkretnej bazy SSAS. Ślad można też odtworzyć na innym serwerze podczas analizy skalowania.

Używanie narzędzia Flight Recorder do analiz następnych

Administratorzy często są sfrustrowani, gdy nie mogą wykryć przyczyny problemu. Nierzad dodatkowym utrudnieniem jest niemożność spowodowania zgłoszonych usterek, gdy administrator próbuje odtworzyć, co się stało, by ustalić, co zrobić, żeby uniknąć kłopotów.

Za pomocą narzędzia Flight Recorder można odtworzyć warunki, które doprowadziły do zgłoszonych problemów. Flight Recorder działa podobnie jak magnetofon i rejestruje aktywność serwera SSAS. Przy każdym uruchomieniu serwera automatycznie rozpoczynany jest nowy plik śladu. Ten rejestrator jest automatycznie włączony. Skonfigurować go możesz za pomocą właściwości serwera SSAS.

Aby wykorzystać plik śladu zapisany przez narzędzie Flight Recorder i odtworzyć operacje z serwera, wykonaj następujące kroki.

1. Uruchom program SQL Server Profiler i otwórz plik śladu utworzony przez narzędzie Flight Recorder. Ten plik domyślnie ma nazwę *FlightRecorderCurrent.trc* i znajduje się w katalogu *C:/Program Files/Microsoft SQL Server/MSAS12.MSSQLSERVER/OLAP/Log*.
2. Wybierz opcję *Replay/Start* z paska narzędzi.
3. W oknie dialogowym *Connect to Server* wprowadź nazwę serwera i dane uwierzytelniające.
4. W oknie dialogowym *Replay Configuration* ustaw opcje odtwarzania. Można np. odtworzyć instrukcje przekazane na serwer w określonym czasie.

Ten sposób odtwarzania jest przydatny, ponieważ usługi SSAS uruchamiają instrukcje zapisane w śladzie. Oczywiście przy rozwiązywaniu problemów warto uwzględnić też inne czynniki, np. liczbę otwartych połączeń, a nawet liczbę sesji istniejących w momencie wystąpienia kłopotów. W trakcie odtwarzania śladów zarejestrowanych przez narzędzie Flight Recorder te aspekty są symulowane.

UWAGA Możesz też rejestrować wywoływane wyrażenia MDX poza plikiem śladu i uruchamiać je za pomocą zadania narzędzia SQL Server Agent. W trybie MOLAP powoduje to aktywowanie pamięci podręcznej usług SSAS i przeniesienie danych z dysku do pamięci (jeśli jest dostępna).

Podsumowanie

W tym rozdziale zapoznałeś się z wieloma informacjami na temat różnych funkcji administracyjnych związanych z usługami SSAS. W SSAS istnieją dwa modele — MOLAP i tabelowy. Zarządzanie w obu tych podejściach przebiega podobnie, przy czym w modelu tabelowym przy przetwarzaniu trzeba zaktualizować całą tabelę. Wiesz już, jak za pomocą kreatora Analysis Services Deployment Wizard instalować nowe bazy danych, a także jak modyfikować istniejące bazy. Dowiedziałeś się też, jak wykonywać i przywracać kopie zapasowe baz SSAS za pomocą interfejsu graficznego oraz skryptów.

Po opanowaniu administrowania usługami SSAS możesz przejść do rozdziału 22., gdzie opisujemy, jak zarządzać usługami SSRS (ang. *SQL Server Reporting Services*) w systemie SQL Server 2014.

Zarządzanie usługami Reporting Services w systemie SQL Server

ZAWARTOŚĆ ROZDZIAŁU:

- Konfigurowanie nowego serwera SSRS za pomocą narzędzia SQL Server Configuration Manager.
- Poruszanie się po serwerze SSRS i jego właściwości.
- Zarządzanie raportami i generowanie ich za pomocą narzędzia Report Manager.

Po zapoznaniu się z tym rozdziałem przekonasz się, że usługi SSRS (ang. *SQL Server Reporting Services*) to jedno z najprostszych w konfiguracji narzędzi spośród produktów oferowanych przez Microsoft. Jest tak dzięki łatwemu w użyciu narzędziu SQL Server Configuration Manager i rozbudowanym (a przy tym zrozumiałym) właściwościom serwera SSRS, do których dostęp można uzyskać w programie SQL Server Management Studio.

Aby zrozumieć działanie usług SSRS, musisz zobaczyć, jak budowane i generowane są raporty. Dlatego rozdział obejmuje krótki przegląd narzędzia Report Builder (jest to przydatny mechanizm do generowania raportów przez administratorów baz danych). Dalej zaczynają się prawdziwe atrakcje. Zobaczysz, jak używać usług SSRS za pomocą narzędzia Report Manager. Narzędzie to zestaw stron WWW udostępnianych razem z usługami SSRS. Za pomocą tych stron użytkownicy mogą łatwo generować raporty, a administratorzy — zarządzać zasobami serwera, planować ich wykorzystanie i maksymalizować ich dostępność.

Narzędzie Configuration Manager usług SSRS

Do konfigurowania usług SSRS służy odrębne narzędzie, niezależne od programu SQL Server Configuration Manager. Zazwyczaj egzemplarz serwera SSRS jest konfigurowany już w trakcie instalowania systemu SQL Server. Poniżej wymieniamy powody, dla których konieczna może być zmiana konfiguracji.

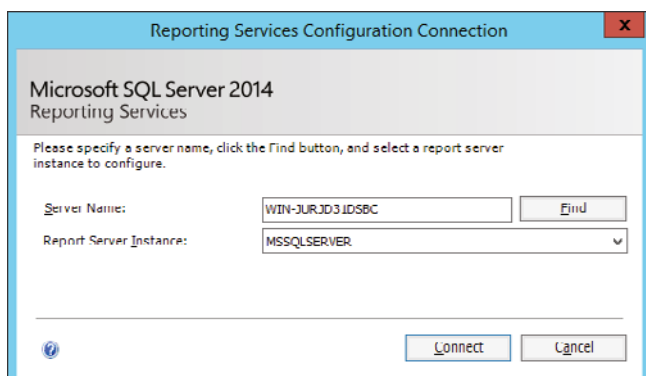
- Chcesz zmienić określoną wartość, np. konto usługowe używane do uruchamiania usług SSRS.
- W trakcie instalacji systemu SQL Server nie zainstalowałeś usług SSRS.
- Chcesz zainstalować dodatkowe egzemplarze serwera SSRS.
- Chcesz dostosować egzemplarz serwera SSRS do większego obciążenia.
- Instalujesz serwer SSRS po wcześniejszym zainstalowaniu silnika bazodanowego.

Te i inne zadania można wykonać za pomocą narzędzia Reporting Services Configuration Manager. Aby je uruchomić, wykonaj następujące kroki.

1. W celu uruchomienia tego programu wybierz opcję *Start/Microsoft SQL Server 2014/Configuration Tools/Reporting Services Configuration Manager*.

UWAGA Narzędzie Reporting Services Configuration Manager jest dostosowane do wersji systemu SQL Server. Narzędzie z systemu SQL Server 2014 można stosować wyłącznie do zarządzania usługami SSRS z tego systemu. Aby zarządzać serwerem SSRS ze starszych wersji systemu SQL Server, użyj dostępnego w nich narzędzia Reporting Services Configuration Manager.

2. Gdy narzędzie się uruchomi, wybierz serwer SSRS i egzemplarz, który chcesz skonfigurować, co pokazano na rysunku 22.1.

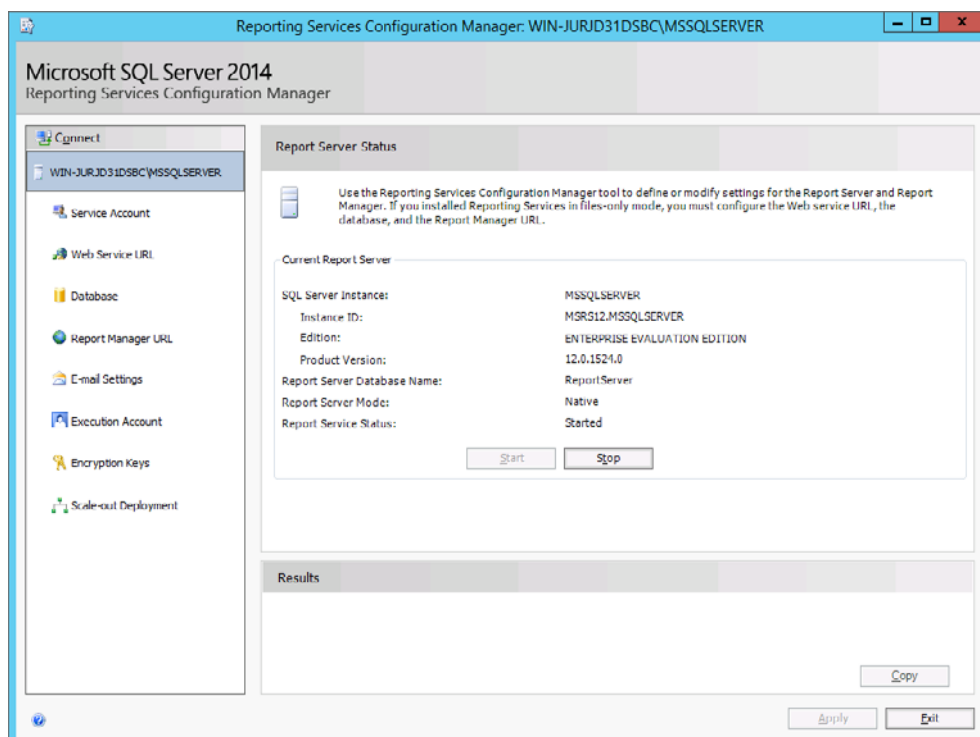


Rysunek 22.1. Wybieranie konfigurowanego serwera

Zwykle konfigurowany jest komputer, na którym działa narzędzie Reporting Services Configuration Manager, jednak za pomocą tego programu można też zarządzać wieloma serwerami SSRS z systemu SQL Server 2014.

3. Po wybraniu serwera kliknij przycisk *Connect*, aby otworzyć główne okno narzędzia.

Ekran powitalny, widoczny na rysunku 22.2, wyświetla podstawowe informacje na temat konfigurowanego serwera. Widoczne są tu nazwa i identyfikator egzemplarz, a także edycja i numer wersji. Pokazana jest też nazwa bazy zawierającej elementy związane z raportami. Na rysunku 22.2 używana jest domyślna baza ReportServer.



Rysunek 22.2. Ekran powitalny narzędzia Reporting Services Configuration Manager

Z informacji wynika, że usługa działa w trybie *Native*. Usługi SSRS mogą też działać w trybie *SharePoint Integrated*. Od wersji SQL Server 2012 usługi SSRS pracujące w trybie *SharePoint Integrated* należy konfigurować za pomocą konsoli platformy SharePoint, a nie w programie Reporting Services Configuration Manager.

W trybie *SharePoint Integrated* to platforma SharePoint zarządza wszystkimi aspektami raportów. Raporty są przesyłane do programu SharePoint i w nim przechowywane. SharePoint kontroluje także zabezpieczenia i subskrypcje. Tylko w tym trybie dostępne są alerty dotyczące danych. Są one podobne do subskrypcji, ponieważ też powodują wysyłanie informacji do użytkowników, przy czym dzieje się to tylko po wystąpieniu określonych warunków. W tym trybie dla raportów dostępne są też mechanizmy z bibliotek SharePointa, np. kontrola wersji i alerty.

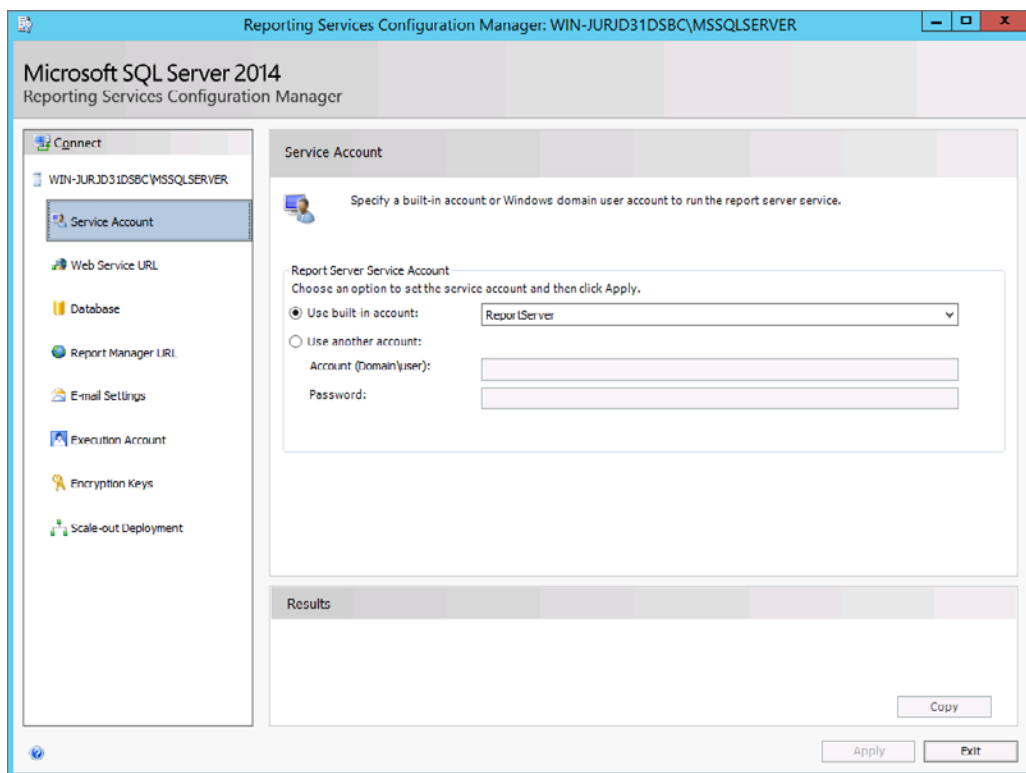
Jednak używanie trybu *SharePoint Integrated* ma pewne wady. Nie można wtedy używać niestandardowych rozszerzeń zabezpieczeń, zarządzać raportami za pomocą narzędzia Report Manager i stosować narzędzia Reporting Services Configuration Manager.

UWAGA Omawianie konfigurowania usług SSRS za pomocą konsoli SharePointa wykracza poza zakres tego rozdziału. Więcej informacji na ten temat znajdziesz na stronie <http://msdn.microsoft.com/en-us/library/bb326356.aspx>.

Czasem po wprowadzeniu zmian trzeba zrestartować usługi SSRS. W tym celu zatrzymaj i uruchom usługę za pomocą przycisków widocznych na rysunku 22.2.

Konto usługowe

Do uruchamiania usług SSRS służy konto usługowe. Początkowo jest ono konfigurowane w trakcie instalowania systemu SQL Server. Później można je zaktualizować lub zmodyfikować na stronie *Service Account*, co przedstawiono na rysunku 22.3. Z tego konta korzystają usługa sieciowa Report Server, narzędzie Report Manager i zadania wykonywane w tle.



Rysunek 22.3. Konfigurowanie konta usługowego

Jeśli zdecydujesz się zmienić konto, masz dwie możliwości.

- **Wykorzystanie konta wbudowanego.** W systemie Windows Server 2008 i starszych wersjach można użyć wybranych kont wbudowanych, np. *Network Service*, *Local System* lub *Local Service*. Microsoft zaleca stosowanie konta *Network Service*.

Od wersji Windows Server 2008 R2 system SQL Server instaluje wszystkie usługi (w tym SSRS) w taki sposób, aby używały kont wirtualnych. *Konto wirtualne* jest kontem lokalnym na serwerze, na którym używa się konta usługowego systemu Windows. SQL Server zarządza wszystkimi ustawieniami takich kont. Na rysunku 22.3 widać, że instalator użył konta wirtualnego *ReportServer*. Gdy zainstalowanych jest kilka egzemplarzy serwera SSRS, w nazwach kont wirtualnych występują nazwy egzemplarzy.

- **Konkretnie konto.** Zamiast konta wbudowanego można wybrać określone konto użytkownika systemu Windows. Może to być konto specyficzne dla komputera i wprowadzone w formacie <nazwa_komputera>\<nazwa_użytkownika>. Można też wykorzystać konto domenowe w formacie <domena>\<użytkownik>. Unikaj stosowania konta administratora domeny. To konto ma większe uprawnienia, niż są potrzebne do uruchamiania usług SSRS, i naraża system na włamania. Trzeba wybrać konto, które już istnieje. Narzędzie Reporting Services Configuration Manager nie może tworzyć nowych kont. Może jednak przyznać potrzebne uprawnienia wybranemu kontu.

Zanim skonfigurujesz konto, zapoznaj się z kilkoma ograniczeniami.

- Nazwa konta nie może mieć więcej niż 20 znaków.
- Jeśli podasz nazwę konta użytkownika z domeny, a stosowany jest protokół Kerberos, musisz zarejestrować nazwę SPN (ang. *Service Principal Name*) dla identyfikatora używanego na serwerze generującym raporty.

OSTRZEŻENIE Uruchamianie systemu SQL Server na serwerze, który pełni funkcję kontrolera domeny, nie jest zalecane. Czasem jednak używana jest taka konfiguracja — zwłaszcza na prostych komputerach używanych do programowania, kiedy to jedna maszyna odpowiada za wszystkie zadania. W takich sytuacjach wbudowane konta usługowe (np. *Local Service* lub *Network Service*) nie mogą być używane dla usług SSRS.

Po zmianie konta usługowego pojawia się monit o utworzenie kopii zapasowej kluczy szyfrowania. Powinieneś być na to przygotowany, ponieważ koniecznie należy wykonać kopię zapasową tych kluczy. Klucze szyfrowania omawiamy dalej w tym rozdziale. Na razie przygotuj się na to, aby po zobaczeniu monitu utworzyć ich kopię zapasową.

Adres URL usługi sieciowej

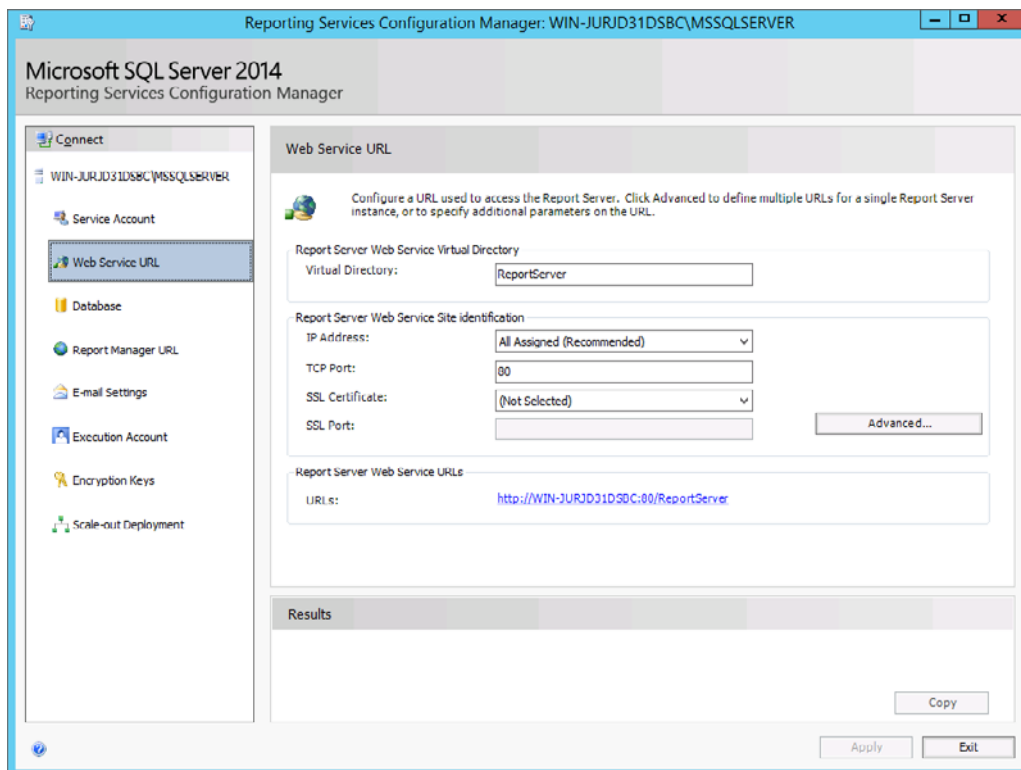
Jedną z ważnych zalet usług SSRS jest możliwość integrowania ich z innymi aplikacjami. Raporty można wyświetlać za pomocą odnośnika na stronie WWW. Można je też umieszczać w kontrolce *Report Viewer* w aplikacjach ASP.NET lub dla systemu Windows. Wszystko to jest możliwe, ponieważ usługi SSRS udostępniają swoje funkcje za pośrednictwem usługi sieciowej.

Usługa sieciowa to program, który jako interfejs wykorzystuje adres URL (ang. *Uniform Resource Locator*; jest to wymyślne określenie adresu HTTP). Gdy odpowiednio sformatowane polecenie `http` zostanie skierowane na adres usługi sieciowej, usługa przetworzy żądanie i zwróci dane, zwykle w postaci strony HTML.

Usługi sieciowe nie tylko zwracają dane. Przy użyciu usług sieciowych SSRS możesz przysyłać raporty, generować je, eksportować dane do pliku, sprawdzać poprawność parametrów itd.

Wszystkie te operacje wymagają użycia aplikacji komunikującej się z usługą sieciową. Aplikacje Microsoftu umożliwiające interakcje z usługami sieciowymi to np. *Report Builder*, *SQL Server Data Tools (SSDT)* i *Report Manager*. Ponadto możesz w jednym z wielu języków platformy .NET pisać własne aplikacje komunikujące się z usługami sieciowymi.

Aby zmodyfikować adres URL usługi sieciowej w danym egzemplarzu serwera SSRS, otwórz stronę *Web Service URL*, pokazaną na rysunku 22.4. Domyślny adres URL obejmuje nazwę serwera, numer portu i nazwę katalogu wirtualnego *ReportServer*. Wszystkie te człony można zmienić.



Rysunek 22.4. Ustawianie adresu URL usługi sieciowej

Na serwerze SSRS można udostępnić adresy URL wielu usług sieciowych. Aby to zrobić, kliknij przycisk *Advanced* i dodaj dla serwera nowe adresy URL. Usługi SSRS obsługują bezpieczne połączenia za pomocą protokołu SSL (ang. *Secure Socket Layer*). By utworzyć takie połączenie, wystarczy podać certyfikat SSL i numer portu.

Jednym z głównych miejsc, gdzie prawdopodobnie będziesz używał adresów URL usług sieciowych, jest narzędzie SSDT. Przy jego użyciu możesz umieszczać raporty na serwerze. W tym celu najpierw otwórz stronę *Properties* projektu raportu. Jedną z właściwości jest docelowy serwer; w tej właściwości należy podać adres URL usługi sieciowej.

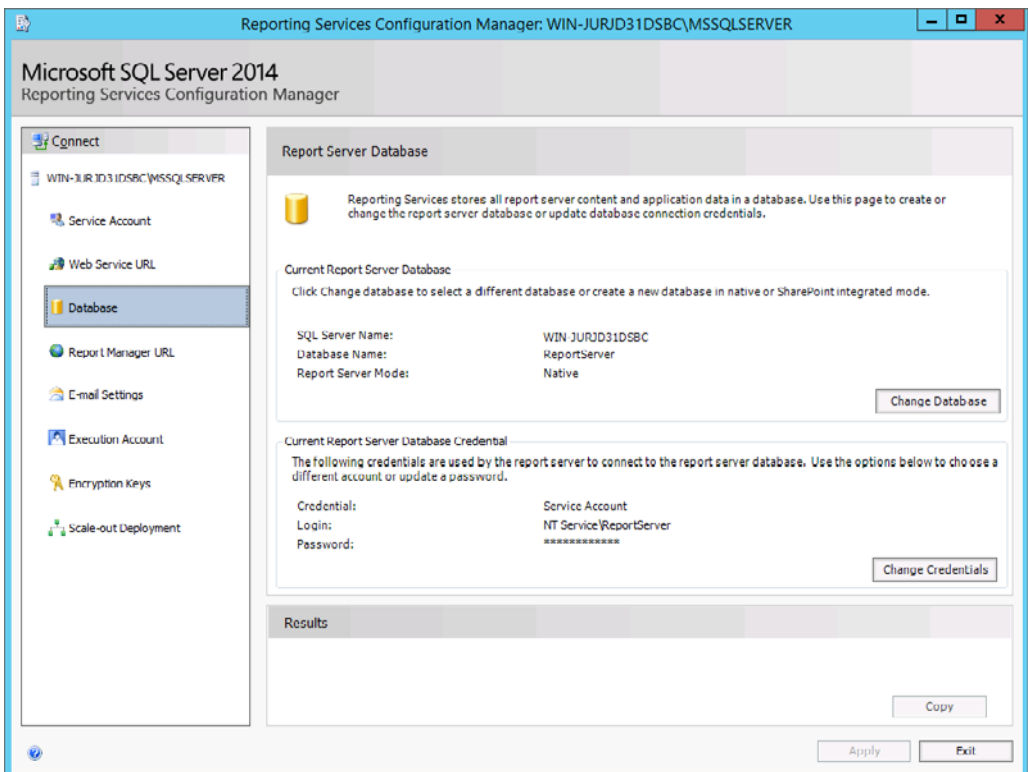
Na rysunku 22.5 przedstawiono przykładowe dane wyświetlane po przejściu pod adres URL usługi sieciowej w przeglądarce. Widoczna jest tu prosta lista raportów z odnośnikami, które umożliwiają otwieranie katalogów. Jest to dobry sposób na sprawdzenie stosowanych w aplikacji adresów URL usługi sieciowej, ale taki interfejs jest bardzo niewygodny dla przeciętnego użytkownika. Na szczęście Microsoft udostępnia kompletne rozwiązanie dla użytkowników, którzy chcą komunikować się z serwerem SSRS. Jest to narzędzie Report Manager. Omawiamy je szczegółowo w podrozdziale „Report Manager”, dalej w tym rozdziale.



Rysunek 22.5. Lista raportów wyświetlona za pomocą adresu URL usługi sieciowej

Bazy SSRS

Usługi SSRS wymagają do działania dwóch baz. Możesz utworzyć je w trakcie instalowania serwera SSRS lub później, za pomocą narzędzia Reporting Services Configuration Manager. Domyślnie bazy te mają nazwy ReportServer i ReportServerTempDB, przy czym można to zmienić w trakcie ich tworzenia. Na rysunku 22.6 przedstawiono ekran konfiguracyjny służący do konfigurowania tych baz.



Rysunek 22.6. Tu możesz skonfigurować bazy usług SSRS

Bazy są wyświetlane w programie SQL Server Management Studio obok innych baz użytkowników.

Baza ReportServer obejmuje wszystkie raporty, a także informacje o źródłach danych, loginach, subskrypcjach itd. Dane poufne są zaszyfrowane za pomocą kluczy szyfrowania. To zagadnienie opisujemy w punkcie „Klucze szyfrowania”, dalej w tym rozdziale.

Baza ReportServerTempDB zawiera tymczasowe informacje z serwera SSRS, np. raporty z pamięci podręcznej. Jeśli raport (np. poranny raport o stanie działań) jest wyświetlany przez wielu użytkowników, można go wygenerować i zapisać w pamięci podręcznej, dzięki czemu ze źródłowej bazy trzeba korzystać tylko raz. Użytkownicy wyświetlający ten raport zobaczą wersję z pamięci podręcznej — nie trzeba go za każdym razem generować od nowa. Gdy upłynie czas przechowywania raportu w pamięci podręcznej, dane zostaną usunięte z bazy ReportServerTempDB.

Te dwie bazy różnią się w istotny sposób. Ważne jest, aby utworzyć kopię zapasową bazy SSRS, ponieważ zawiera ona wszystkie informacje o raportach przechowywanych w danym egzemplarzu serwera SSRS. Jeśli trzeba będzie przywrócić bazę, będziesz mógł to zrobić w standardowy sposób. Po przywróceniu bazy trzeba przywrócić także klucze szyfrowania, aby można było odczytać zaszyfrowane dane.

Z bazą ReportServerTempDB jest inaczej. Wszystkie dane z tej tymczasowej bazy można usunąć i nie spowoduje to trwałego uszkodzenia danego egzemplarza serwera SSRS. Konieczne jest jednak wtedy ponowne wygenerowanie i umieszczenie w pamięci podręcznej wszystkich raportów z włączonym zapisywaniem w tej pamięci, ponieważ zostaje ona utracona w momencie usunięcia bazy ReportServerTempDB.

Jeśli chcesz umożliwić odzyskanie danych po katastrofie, masz dwie możliwości. Pierwsza to utworzenie kopii zapasowych baz ReportServerTempDB i ReportServer. Jednak tymczasowa baza zwykle rośnie do dużych rozmiarów, dlatego jej kopia zapasowa czasem zajmuje dużo miejsca, a jej wygenerowanie wymaga dużo czasu. Dlatego możesz zdecydować się napisać skrypt tworzący bazę ReportServerTempDB. W momencie odzyskiwania należy uruchomić ten skrypt i — jeśli trzeba — wygenerować raporty, które mają być dostępne w pamięci podręcznej. W celu utworzenia takiego skryptu kliknij bazę ReportServerTempDB prawym przyciskiem myszy, wybierz opcję *Script Database As/Create To/File* i zapisz wygenerowane dane wyjściowe.

ReportServer i ReportServerTempDB to nazwy domyślne tych baz. Jeśli na tej samej maszynie zainstalowanych jest kilka egzemplarzy serwera SSRS, do tych domyślnych nazw dołączane jest podkreślenie i nazwa egzemplarza. Warto zachować te nazwy, ponieważ większość informatyków zajmujących się systemem SQL Server jest do nich przyzwyczajona. Jeśli jednak trzeba, możesz zmienić te nazwy.

W celu zmodyfikowania bazy kliknij przycisk *Change Database*. Pojawi się wtedy kreator udostępniający dwie opcje — nawiązanie połączenia z istniejącą bazą lub utworzenie nowej bazy. Gdy wybierzesz jedną z tych możliwości, kreator przeprowadzi Cię przez serię ekranów wspólnych dla obu opcji. Należy wybrać serwer dla bazy, podać nazwę bazy, dane uwierzytelniające itd. Po zakończeniu tego procesu serwer SSRS będzie używał bazy wskazanej w kreatorze.

Choć bazy ReportServer i ReportServerTempDB nieraz znajdują się w tym samym systemie SQL Server, w którym działa serwer SSRS, nie jest to konieczne. Bazy danych używane do generowania raportów możesz umieścić na odrębnym komputerze, a na innej maszynie uruchomić tylko serwer SSRS.

Ta możliwość jest często wykorzystywana przy skalowaniu. W tym scenariuszu można zastosować dwa rozwiązania. W pierwszym bazy ReportServer i ReportServerTempDB znajdują się na serwerze, na którym działa silnik bazodanowy systemu SQL Server. Następnie należy utworzyć dwa serwery, na których działa tylko SSRS. Oba powinny używać baz ReportServer i ReportServerTempDB z pierwszego serwera.

W drugim rozwiązaniu używane są tylko dwa serwery. Na pierwszym działają obie bazy i serwer SSRS. Drugi obejmuje tylko serwer SSRS, który używa baz ReportServer i ReportServerTempDB z pierwszego serwera.

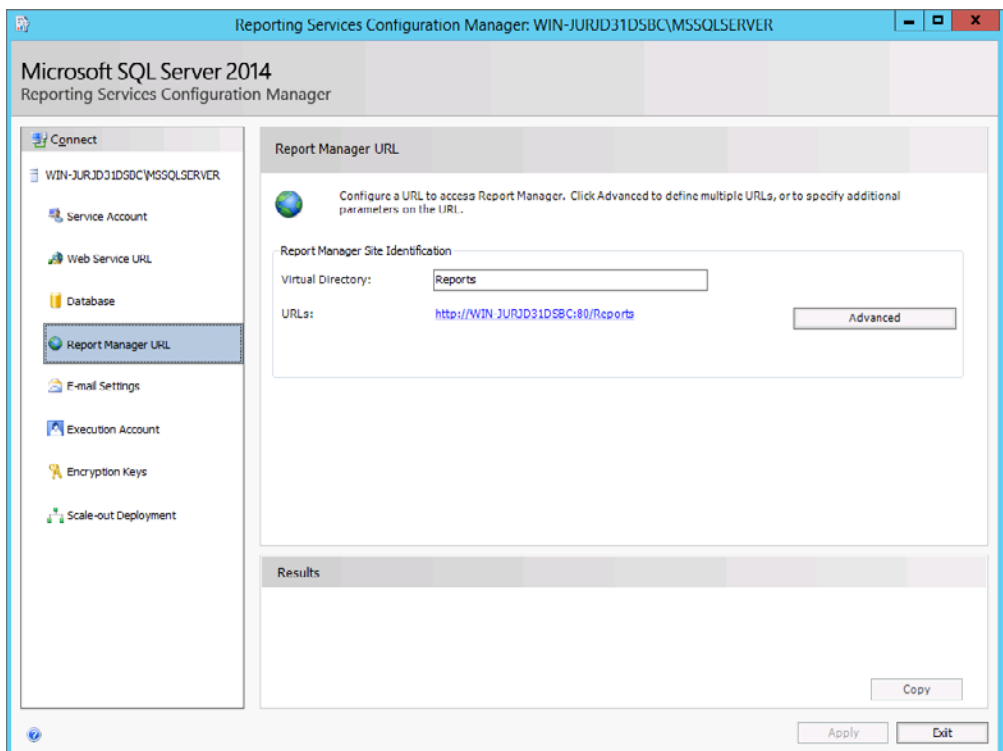
Choć są to najczęściej stosowane konfiguracje, przy skalowaniu możesz zastosować więcej niż dwa serwery SSRS. Należy wtedy tak skonfigurować dodatkowe serwery, aby korzystały z baz ReportServer i ReportServerTempDB z centralnego serwera. Więcej informacji na ten temat znajdziesz w punkcie „Skalowanie systemu”, dalej w tym rozdziale.

Jeśli w środowisku z jednym serwerem zdecydujesz się zainstalować i skonfigurować usługi SSRS w ramach instalacji systemu SQL Server, zwykle nie będziesz musiał modyfikować danych widocznych na rysunku 22.6. Warto jednak wiedzieć, jak używane są bazy danych do generowania raportów, a także jak można je skonfigurować na serwerze.

Adres URL narzędzia Report Manager

Dla usług SSRS dostępny jest interfejs Report Manager, który umożliwia użytkownikom przesyłanie, konfigurowanie i generowanie raportów. Za pomocą tego narzędzia można ustawić zabezpieczenia dla konkretnego użytkownika lub grupy. Uprawnienia mogą być różne: od bardzo ograniczonych, umożliwiających tylko dostęp do wybranych raportów, po rozbudowane, zapewniające pełną kontrolę nad serwerem do generowania raportów.

Adresem URL narzędzia Report Manager jest zwykle nazwa serwera, po której następuje numer domyślnego portu (80) i nazwa katalogu wirtualnego (*Reports*). Ten adres URL można zmienić na stronie *Report Manager URL*, pokazanej na rysunku 22.7.



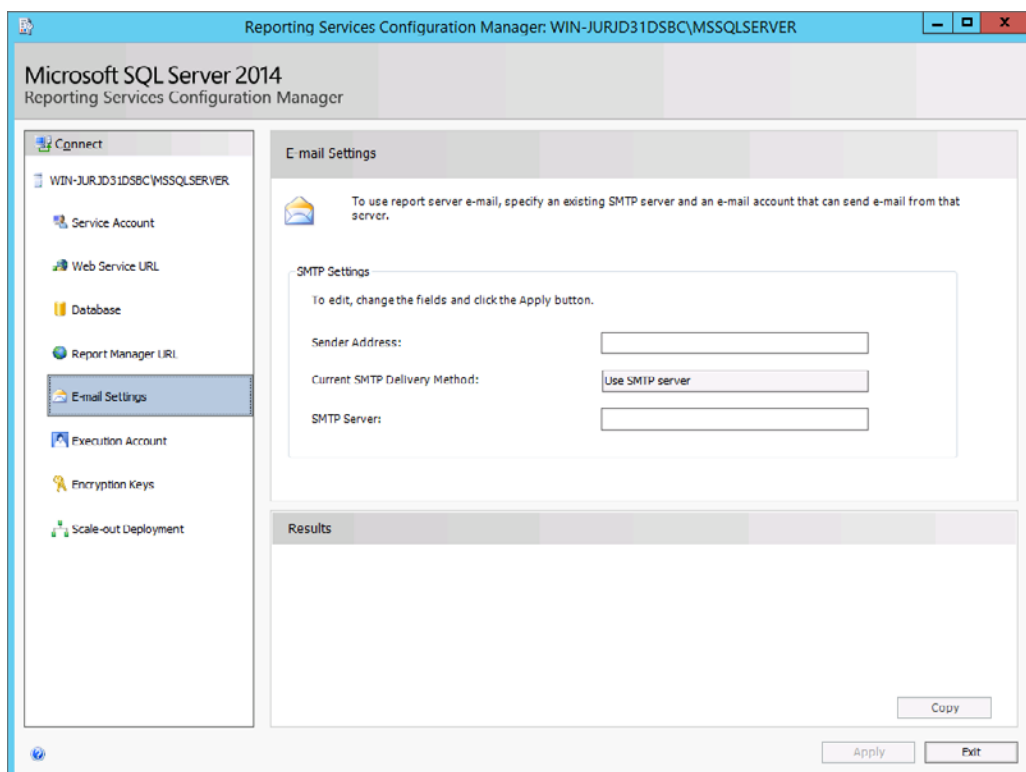
Rysunek 22.7. Tu możesz ustawić adres URL narzędzia Report Manager

Jeśli chcesz utworzyć więcej adresów URL, to (podobnie jak w przypadku adresów URL usług sieciowych) kliknij przycisk *Advanced* w narzędziu Report Services Configuration Manager. Możliwe, że chcesz zastosować jeden standardowy adres URL (HTTP) do użytku wewnętrznego i użyć protokołu SSL (adres HTTPS) do użytku zewnętrznego.

Report Manager to rozbudowane narzędzie. Omawiamy je szczegółowo w punkcie „Report Manager”, dalej w tym rozdziale.

Ustawienia poczty elektronicznej

Serwer SSRS umożliwia też skonfigurowanie subskrypcji raportów, które następnie są przysyłane na adres e-mail. Użytkownicy mogą zażądać automatycznego generowania raportów i przysyłania ich pocztą elektroniczną. Aby było to możliwe, serwer SSRS musi mieć dostęp do konta e-mailowego. Na rysunku 22.8 przedstawiono stronę *E-Mail Settings*, na której wpisywane są informacje związane z e-mailami.

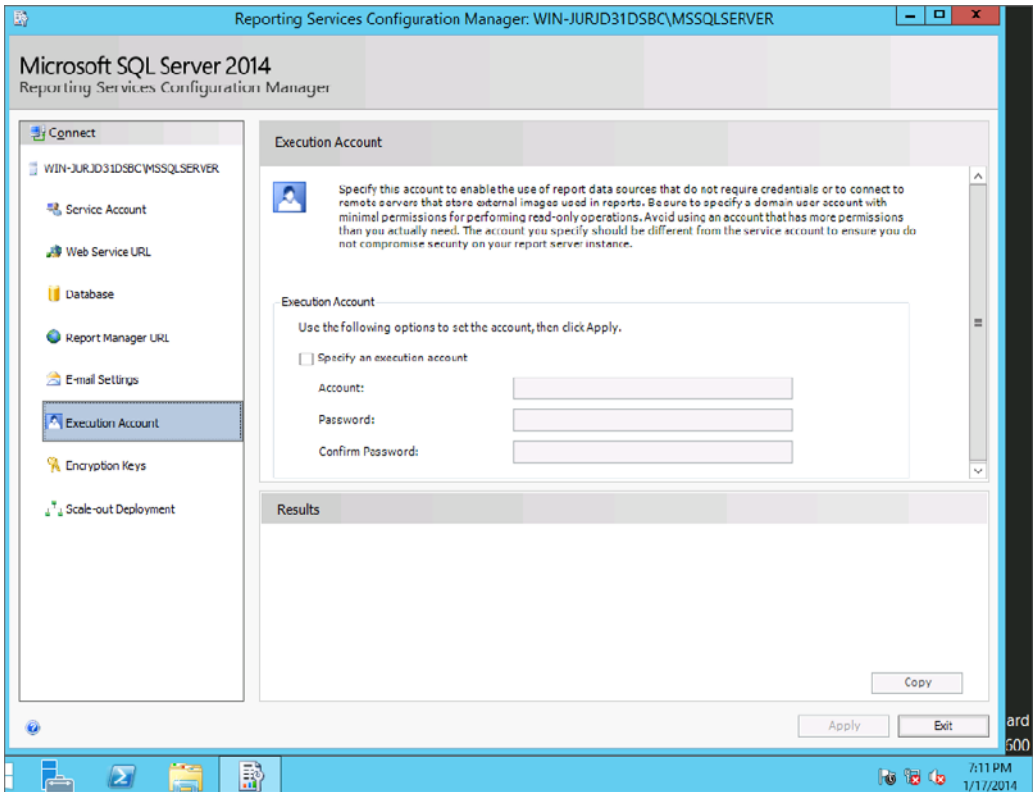


Rysunek 22.8. Konfigurowanie przesyłania raportów na adres e-mail

Podobnie jak w większości kont używanych przez serwery, zadбай o to, aby konto do wysyłania e-maili miało niewygasające hasło. Ponadto przypisz do niego uprawnienia do wysyłania różnych typów załączników generowanych przez usługi SSRS. Zachowaj jednak ostrożność. Niektóre raporty są dość duże i mogą przepełnić skrzynkę pocztową. Pomyśl o generowaniu raportów i zapisywaniu ich w centralnym repozytorium oraz przesyłaniu e-mailem samych odnośników do tych raportów.

Konto wykonawcze

Czasem raport wymaga danych, ale dane uwierzytelniające niezbędne do pobrania tych danych nie są w raporcie zapisane. Usługi SSRS wymagają danych uwierzytelniających do pobierania źródłowych informacji. Na stronie *Execution Account* narzędzia Configuration Manager (przedstawionej na rysunku 22.9) możesz wybrać konto domenowe, które chcesz zastosować. To konto stanie się kontem wykonawczym. Wybrane konto domenowe powinno mieć minimalne uprawnienia potrzebne do wykonywania jego zadań. Należy przyznać mu np. uprawnienia tylko do odczytu do danych źródłowych wykorzystywanych w raportach.



Rysunek 22.9. Ustawianie konta wykonawczego

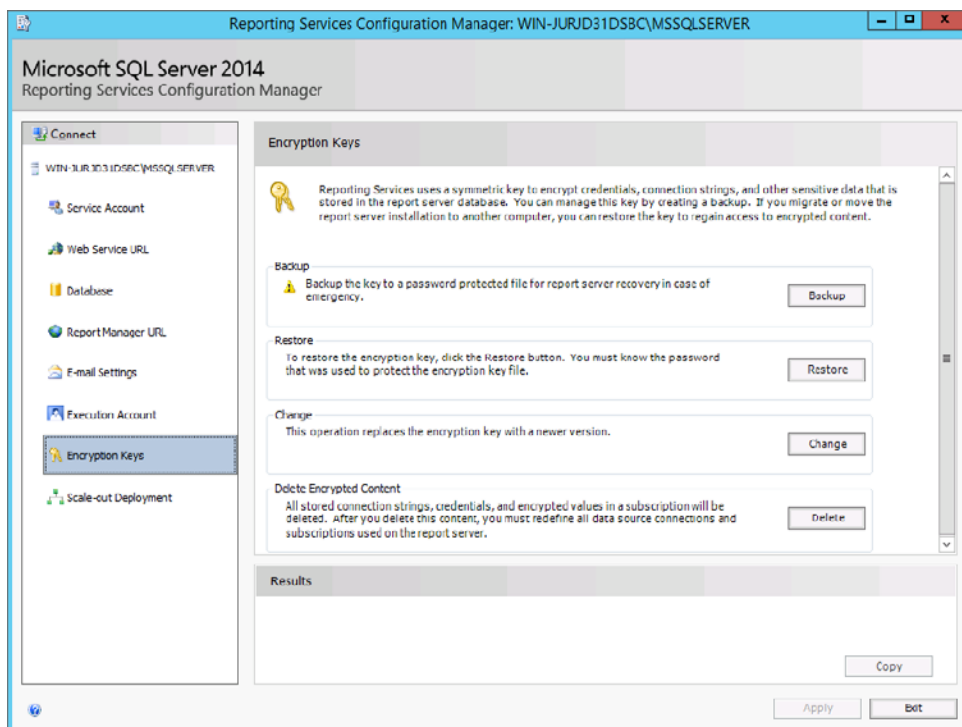
W raportach można też wyświetlać rysunki przechowywane poza samym raportem. Dane uwierzytelniające konta wykonawczego muszą umożliwiać dostęp do katalogów z tymi rysunkami.

UWAGA Nie pomył konta wykonawczego z kontem usługowym. Konto usługowe jest używane do wykonywania usługi SSRS. Konto wykonawcze służy do dostępu do danych lub rysunków, gdy inne dane uwierzytelniające nie są dostępne.

Konto domenowe używane dla konta wykonawczego powinno być inne niż dla konta usługowego. Konto wykonawcze wymaga odmiennych (i zwykle mniejszych) uprawnień niż konto usługowe. Zastosowanie różnych kont minimalizuje zagrożenia związane z bezpieczeństwem.

Klucze szyfrowania

Usługi SSRS wymagają wielu poufnych informacji do wykonywania zadań. Dane uwierzytelniające, łańcuchy znaków połączenia i podobne informacje trzeba przechowywać w bezpieczny sposób. Tego typu poufne dane są zapisywane w bazie usług SSRS. Jednak przed zapisaniem takie informacje są szyfrowane przy użyciu klucza. Do zarządzania kluczami szyfrowania służy strona *Encryption Keys*, przedstawiona na rysunku 22.10.



Rysunek 22.10. Zarządzanie kluczami szyfrowania

Bardzo ważne jest, aby utworzyć kopię zapasową kluczy szyfrowania. W czasie przywracania bazy usług SSRS (z powodu awarii lub jako ułatwienie przy przenoszeniu egzemplarza na nowy serwer) trzeba przywrócić także klucze szyfrowania. Jeśli tego nie zrobisz, wszystkie zapisane poufne informacje staną się niedostępne i będziesz musiał ręcznie je odtworzyć, co jest żmudnym zadaniem.

Po przywróceniu bazy usług SSRS możesz przywrócić klucze szyfrowania w oknie dialogowym widocznym na rysunku 22.10. Gdy już przywrócisz klucze, serwer SSRS będzie mógł poprawnie odszyfrować zapisane dane uwierzytelniające i przywrócić pełną funkcjonalność serwera.

Jest wiele sytuacji, w których trzeba zmienić dane uwierzytelniające. Możliwe, że utraciłeś kopię zapasową kluczy szyfrowania, co stanowi zagrożenie dla bezpieczeństwa. Możliwe też, że firmowe reguły wymagają okresowego modyfikowania wszystkich kluczy szyfrowania. Po kliknięciu przycisku *Change* widocznego na rysunku 22.10 możesz utworzyć nowy klucz. Oczywiście po jego zmianie koniecznie utwórz kopię zapasową.

Czasem trzeba usunąć poufne dane zapisane w bazie usług SSRS. W tym celu kliknij przycisk *Delete*. Spowoduje to usunięcie wszystkich poufnych informacji zapisanych przez usługi SSRS.

Skalowanie systemu

W pewnym momencie obciążenie serwera SSRS może stać się zbyt duże, aby jedna maszyna mogła sobie z nim poradzić. Microsoft w takich sytuacjach udostępnia mechanizm skalowania systemu. Pozwala on na przetwarzanie raportów za pomocą wielu serwerów. Wszystkie te serwery korzystają ze wspólnej bazy usług SSRS.

Zanim zaczniesz planować budowanie środowiska umożliwiającego skalowanie, zapoznaj się z wymaganiami wstępnymi. Mechanizm ten jest dostępny tylko w edycji Enterprise. Choć można go skonfigurować także w edycjach Developer i Evaluation, jest w nich przeznaczony do nauki, programowania i testów, a nie do stosowania w środowisku produkcyjnym. Edycje Standard, Workgroup i Express systemu SQL Server nie obsługują skalowania systemów z serwerami SSRS.

Wszystkie serwery w farmie używanej do skalowania muszą korzystać z wersji o tym samym numerze oraz mieć zainstalowane te same aktualizacje i pakiety SP. Ponadto muszą działać w tej samej domenie lub w domenie zaufanej. Na tych serwerach trzeba też korzystać z jednakowych trybów uwierzytelniania. Jeśli utworzyłeś niestandardowe rozszerzenia związane z uwierzytelnianiem, trzeba je zastosować na każdym z serwerów.

UWAGA Choć nie jest to wymagane, oba serwery powinny mieć identyczną (lub jak najbardziej zbliżoną) fizyczną konfigurację.

Po spełnieniu tych podstawowych wymagań pora skonfigurować serwery. Tu prezentujemy najczęstszy scenariusz. Pierwszy komputer obejmuje serwer SSRS oraz zainstalowane bazy ReportServer i ReportServerTempDB. Na drugim działa tylko serwer SSRS.

Wykonaj następujące kroki.

1. Skonfiguruj pierwszy serwer SSRS w sposób opisany w tym rozdziale.
2. Następnie zainstaluj serwer SSRS na drugim komputerze. Wprawdzie możesz zainstalować kilka egzemplarzy serwera SSRS na jednej maszynie i skonfigurować je na potrzeby skalowania, rozwiązanie to nie daje żadnych korzyści. Gdy dojdiesz do etapu instalowania serwera SSRS, wybierz opcję *Install but do not configure server*.
3. Po zakończeniu instalacji uruchom narzędzie Reporting Services Configuration Manager i w oknie dialogowym, które się pojawi, wybierz nowy serwer. Przejdź do strony *Database* i wskaż bazę usług SSRS z pierwszego komputera.
4. Następnie przejdź do stron *Report Server Web Service URL* i *Report Manager URL*, aby skonfigurować adresy URL. Jednak na razie ich nie testuj, ponieważ nie będą dostępne do momentu dodania serwera do rozwiązania zapewniającego skalowalność (zrobisz to w następnym kroku).
5. Zamknij połączenie na drugim serwerze SSRS, ponownie uruchom narzędzie konfiguracyjne i przejdź do pierwotnego serwera. Otwórz stronę *Scale-out Deployment*. Teraz powinieneś zobaczyć oba serwery. Status pierwszego z nich to *Joined*. Dla nowego serwera powinien być wyświetlony status *Waiting to Join*.
6. Wybierz nowy serwer, a następnie kliknij przycisk *Add Server*. Po dołączeniu serwer możesz sprawdzić. W tym celu wybierz adres URL usługi sieciowej lub narzędzia Report Manager określony w trakcie konfigurowania nowego serwera.

Usługi SSRS można też instalować w klastrach z równoważeniem obciążenia sieci. Wtedy trzeba skonfigurować także kilka dodatkowych elementów. Więcej informacji na ten temat zawiera artykuł „Configure a Report Server on a Network Load Balancing Cluster” w dokumentacji Books Online.

Dziennik wykonywania raportów

W tym podrozdziale dowiesz się, co zawiera dziennik wykonywania raportów. Zacznij od następujących kroków.

1. Otwórz program SQL Server Management Studio i nawiąż połączenie z silnikiem bazodanowym, w którym przechowywana jest baza ReportServer. Jest to domyślna nazwa tej bazy i tu posługujemy się właśnie nią, jeśli jednak zmieniłeś tę nazwę w trakcie instalowania serwera SSRS lub za pomocą narzędzia Reporting Services Configuration Manager, wybierz odpowiednią bazę.
2. W węźle *Views* widoczne są trzy widoki bezpośrednio powiązane z rejestrowaniem zdarzeń: *ExecutionLog*, *ExecutionLog2* i *ExecutionLog3*. Każdy z nich zawiera te same podstawowe informacje: identyfikator raportu, egzemplarz serwera, na którym wykonano raport, czas generowania raportu, autora raportu itd. W drugiej i trzeciej wersji tego widoku znajdują się też dodatkowe informacje.

Możesz napisać w SQL-u instrukcję, która pobiera dane z widoku *ExecutionLog*. Instrukcja będzie podstawą raportu, który utworzysz w następnym punkcie. Wspomniany widok zawiera większość potrzebnych informacji. Jedyny brakujący element to nazwy raportów. Można je pobrać z tabeli *dbo.Catalog*. W listingu 22.1 (plik *Listing22-1.sql*) przedstawiono zapytanie zwracające potrzebne informacje.

Listing 22.1. Skrypt w języku SQL wyświetlający podstawowe informacje z dziennika błędów

```
SELECT [InstanceName]
, C.[Path] AS [ReportFolder]
, C.[Name] AS [ReportName]
, [UserName]
, CASE [RequestType]
  WHEN 0 THEN 'Interactive Report'
  ELSE 'Subscription Report'
END AS [ReportType]
, [TimeStart]
, [TimeEnd]
, [TimeDataRetrieval] AS [TimeDataRetrievalMilliseconds]
, [TimeProcessing] AS [TimeProcessingMilliseconds]
, [TimeRendering] AS [TimeRenderingMilliseconds]
, CASE [Source]
  WHEN 1 THEN 'Live'
  WHEN 2 THEN 'Cache'
  WHEN 3 THEN 'Snapshot'
  WHEN 4 THEN 'History'
  WHEN 5 THEN 'AdHoc'
  WHEN 6 THEN 'Session'
  WHEN 7 THEN 'RDCE'
  ELSE 'Other'
END AS [ReportSource]
```

```

, [Status]
, [ByteCount]
, [RowCount]
FROM [ReportServer].[dbo].[ExecutionLog] E
JOIN [ReportServer].[dbo].[Catalog] C ON E.ReportID = C.ItemID

```

Możesz sprawdzić, jakie jeszcze informacje są dostępne.

Report Manager

Raport po wygenerowaniu można przekazać na serwer SSRS, gdzie zarządza się raportem. Służy do tego narzędzie Report Manager.

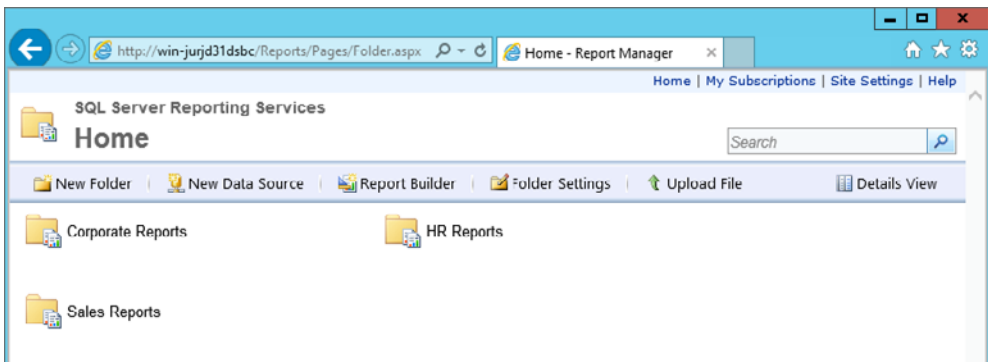
Report Manager to interfejs WWW, za pomocą którego zarówno informatycy, jak i użytkownicy końcowi mogą zarządzać kolekcją raportów i je generować. Aby przejść do tego narzędzia, otwórz Internet Explorer (lub inną przeglądarkę internetową) i wybierz adres URL ustawiony na stronie *Report Manager URL* w programie Reporting Services Configuration Manager. Następnie możesz wykonywać różne operacje. Oto trzy główne typy zadań.

- W SSDT możesz porządkować raporty w projektach. Następnie możesz zainstalować raporty z narzędzia SSDT na serwerze SSRS przy użyciu adresu URL usługi sieciowej ustawionego w programie Reporting Services Configuration Manager.
- Możesz zapisać raporty z narzędzia Report Builder bezpośrednio na serwerze SSRS za pomocą adresu URL usługi sieciowej.
- Report Manager umożliwia przysyłanie raportów na dysk.

Używanie narzędzia Report Manager

Report Manager ma pewną dziwną cechę — aby wszystkie funkcje działały prawidłowo, trzeba go uruchomić w trybie administratora. Znajdź w menu *Start* ikonę programu Internet Explorer, kliknij ją prawym przyciskiem myszy i wybierz opcję *Uruchom jako administrator*.

Gdy uruchomisz Internet Explorer, wpisz adres URL narzędzia Report Manager. Ekran powinien wyglądać podobnie do tego z rysunku 22.11 (nie martw się, jeśli zauważysz jakieś rozbieżności).

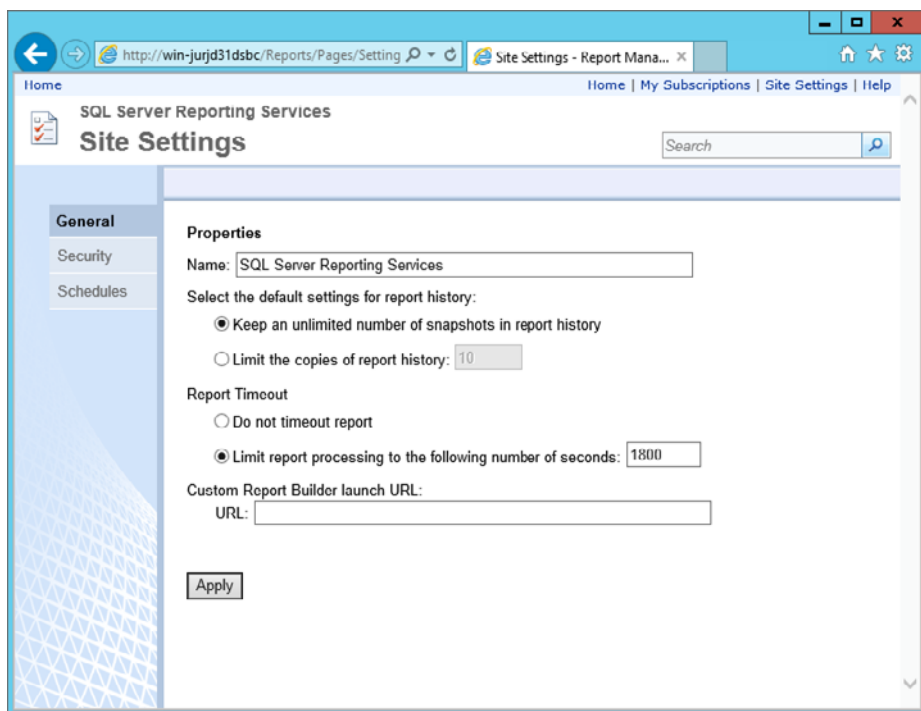


Rysunek 22.11. Ekran główny narzędzia Report Manager

Menu nawigacyjne

W prawym górnym rogu rysunku 22.11 widoczne jest menu nawigacyjne. Opcja *Home* pozwala przejść z powrotem na stronę główną, widoczną na rysunku 22.11. Opcja *My Subscriptions* powoduje przejście do prywatnego obszaru z raportami użytkownika. Jeśli subskrypcje są wyłączone, ta opcja nie jest widoczna.

Opcja *Site Settings* jest przeznaczona tylko dla administratorów serwera SSRS i nie jest widoczna dla nikogo oprócz ustawionych administratorów. Na rysunku 22.12 pokazano, że opcje dostępne na docelowej stronie to podzbiór właściwości, które można ustawić w programie SQL Server Management Studio. Opcja *Help* umożliwia wyświetlenie podstawowej pomocy narzędzia Report Manager.

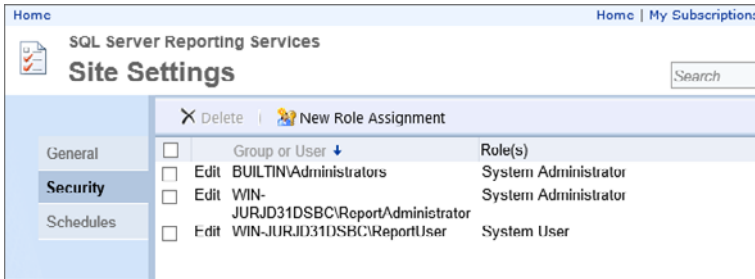


Rysunek 22.12. Strona z ustawieniami administracyjnymi

Na stronie *Site Settings* zmien nazwę serwera na określenie dopasowane do środowiska (jeśli jeszcze tego nie zrobiłeś). Zwykle używana jest nazwa firmy, jednak możesz też ustawić nazwę działu. Jeśli używasz serwera testowego lub rozwojowego, też uwzględnij to w nazwie.

Po zmianie nazwy serwera (lub innych właściwości) trzeba kliknąć widoczny na rysunku 22.12 przycisk *Apply*. Dotyczy to nie tylko tej strony, ale wszystkich stron narzędzia Report Manager. Gdziekolwiek wprowadzisz zmiany, musisz kliknąć przycisk *Apply*. W przeciwnym razie zmiany zostaną utracone.

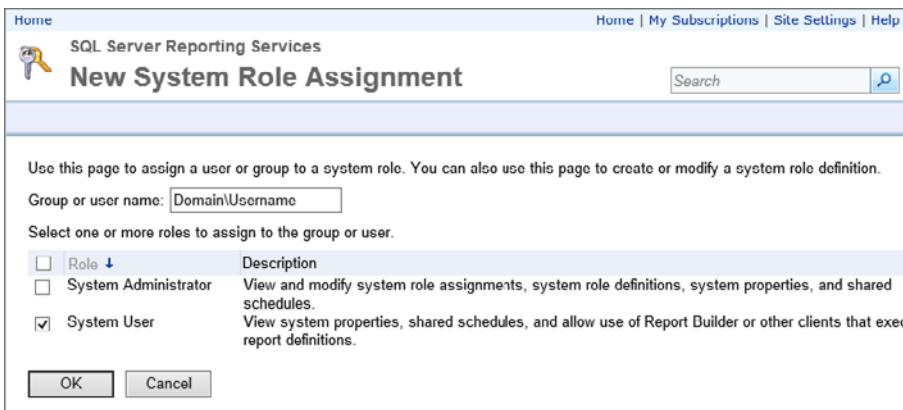
Do tej pory omówiono zakładkę *General* strony *Site Settings* (menu z zakładkami jest widoczne w lewej części ekranu). Następną ważną zakładką to *Security*. Pozwala ona dodawać i edytować dane użytkowników lub ich grup. Na ekranie przedstawionym na rysunku 22.13 widoczne są dane dwóch użytkowników z danego serwera i przypisane im role.



Rysunek 22.13. Zarządzanie użytkownikami

Aby dodać nowego użytkownika, kliknij przycisk *New Role Assignment* (nie jest on widoczny na części ekranu pokazanej na rysunku 22.13). Pojawi się strona *New System Role Assignment*, pokazana na rysunku 22.14. Zacznij od wprowadzenia nazwy użytkownika lub grupy z usługi Active Directory, a następnie zaznacz odpowiednią rolę. Dostępne są dwie opcje.

- **System Administrator.** Zachowaj ostrożność przy przyznawaniu tej roli. Administratorzy systemu mogą zmieniać dowolne właściwości, dodawać i usuwać użytkowników oraz raporty itd. Tę rolę należy zarezerwować dla administratorów baz danych i podobnych specjalistów.
- **System User.** Jest to prosta rola. Umożliwia użytkownikom wyświetlanie (ale już nie zmienianie) właściwości systemu i współużytkowanych harmonogramów, a także uruchamianie narzędzia Report Builder.

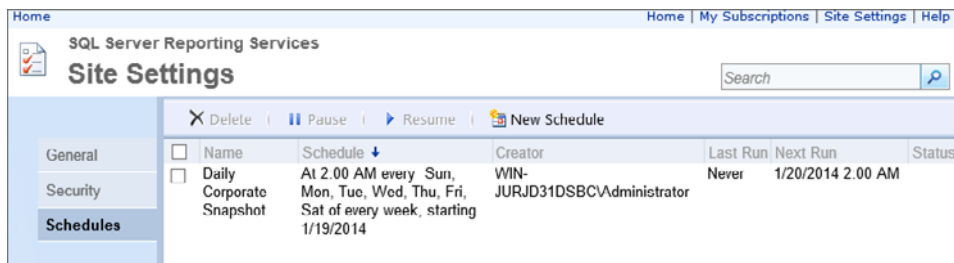


Rysunek 22.14. Przypisywanie ról użytkownikom

Stronę *Schedules* można wyświetlić za pomocą zakładki o tej samej nazwie, widocznej w lewej części strony *Site Settings*, co przedstawiono na rysunku 22.13. Strona *Schedules* służy do tworzenia współużytkowanych harmonogramów zapisywania snapshotów raportów. Załóżmy, że używane są trzy raporty z ważnymi danymi finansowymi firmy. Raporty można generować niezależnie od siebie kilka razy w ciągu dnia.

Dla inspekcji należy każdego dnia uruchamiać wszystkie trzy raporty o tej samej porze i zapisywać snapshoty. W tym celu zacznij od przygotowania współużytkowanego harmonogramu. Wybierz opcję *New Schedule*. Określ, o jakiej porze dnia i w jakie dni raporty mają być generowane, a także od kiedy należy rozpocząć ten proces i (opcjonalnie) kiedy ma on zostać zakończony.

Na rysunku 22.15 przedstawiono harmonogram o nazwie *Daily Corporate Snapshot*. Zapamiętaj go; dalej w rozdziale zobaczysz, jak go użyć.

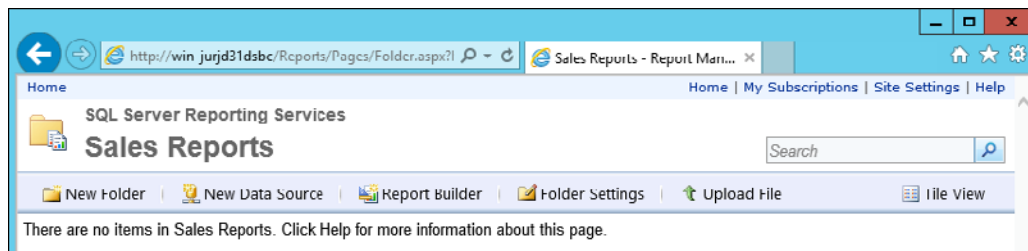


Rysunek 22.15. Skonfigurowany harmonogram generowania raportów

Menu zadań

Wróć do ekranu *Home* (patrz rysunek 22.11). W górnej jego części znajduje się duże menu zadań. Pierwsza opcja od lewej to *New Folder*. Report Manager umożliwia porządkowanie raportów w logicznych katalogach, podobnych do katalogów z dysku twardego. Gdy przeniesiesz raport do katalogu, w drugim wierszu tytułu raportu pojawi się nazwa katalogu.

Na rysunku 22.16 widoczna jest nowa nazwa serwera SSRS i informacja o przejściu do katalogu *Sales Reports*. Na dole znajduje się komunikat informujący, że ten katalog nie zawiera żadnych raportów. Wyświetlanie tego komunikatu to dobre rozwiązanie; gdyby nie były widoczne żadne elementy, użytkownik zastanawiałby się, dlaczego tak się dzieje.



Rysunek 22.16. Komunikat informuje, że katalog Sales Reports jest pusty

Na stronie *Home* obok opcji *New Folder* widoczna jest opcja *New Data Source*. W trakcie tworzenia raportu w narzędziu Report Builder należy zacząć od utworzenia zbioru danych. Gdy stworzysz zbiór danych, najpierw należy podać źródło danych.

Można tu wykorzystać współużytkowane źródło danych. Aby je utworzyć, użyj opcji *New Data Source* w narzędziu Report Manager. Następnie wskaż to źródło w narzędziu Report Builder.

Współużytkowane źródła danych ułatwiają zarządzanie systemem. Jeśli grupa raportów korzysta z tej samej bazy danych i baza ta zostanie przeniesiona na nowy serwer, połączenie wystarczy zaktualizować raz; nie trzeba tego robić dla każdego raportu. Współużytkowane źródła danych ułatwiają także tworzenie raportów. Możesz skonfigurować testowy serwer SSRS ze źródłem danych powiązanym z testową bazą danych. Gdy raport przejdzie wszystkie testy, można go przenieść na serwer produkcyjny ze współużytkowanym źródłem danych o tej samej nazwie, ale powiązanym z produkcyjną bazą danych. Wtedy nie trzeba aktualizować raportu.

Choć tworzenie źródła danych jest proste, wymaga pewnej wiedzy na temat działania usług SSRS, a także o tym, jak raport będzie używany. Na stronie *New Data Source* (patrz rysunek 22.17) widać, że należy zacząć od ustawienia dla źródła danych opisowej nazwy. Możesz też dodać opcjonalny opis, aby lepiej wyjaśnić przeznaczenie źródła danych.

Rysunek 22.17. Konfigurowanie nowego źródła danych

Ekran *Home* (patrz rysunek 22.11) jest wyświetlany w trybie kafelków. Większość użytkowników używa narzędzia Report Manager właśnie w tym trybie. Dostępny jest też tryb szczegółów; opisujemy go dalej. Aby zablokować wyświetlanie źródła danych, zaznacz pole *Hide in tile view*, widoczne na rysunku 22.17. Jest to często stosowana technika. Zmniejsza ona liczbę elementów wyświetlanych w narzędziu Report Manager, co upraszcza zawartość okna.

Niżej na stronie widocznej na rysunku 22.17 znajduje się opcja *Enable this data source*. Pozwala ona wyłączyć i włączyć określone źródło danych.

Następna opcja to typ źródła danych. Usługi SSRS zapewniają dostęp nie tylko do systemu SQL Server, ale też do różnych innych źródeł danych. Usługi SSAS, SQL Server Azure, OLE DB i Oracle to tylko kilka spośród licznych źródeł dostępnych w usługach SSRS.

Format łańcucha znaków połączenia zależy od typu źródła danych. Gdy źródłem jest system SQL Server, łańcuch znaków połączenia zaczyna się od członu `Data Source=<nazwa_serwera>`. Jeśli nie jest używany egzemplarz domyślny, należy też podać nazwę egzemplarza systemu SQL Server. Dalej następuje średnik, człon `Initial Catalog=` i nazwa bazy, z której pobierane będą dane. Zwróć uwagę na spacje w określeniach `Data Source` i `Initial Catalog` — są one konieczne.

Aby nawiązać połączenie z przechowywanym na serwerze testowym dziennikiem związanym z wykonywaniem raportów, użyj następującego łańcucha znaków połączenia:

`Data Source=WIN-JURJD31DSBC;Initial Catalog=ReportServer`

Dalej znajduje się sekcja *Connect using*. Obejmuje ona cztery opcje. Mają one duży wpływ na możliwe sposoby używania raportów na serwerze.

- ***Credentials supplied by the user running the report.*** Gdy ustawiona jest ta opcja, użytkownicy przy każdym wykonywaniu raportu widzą prośbę o wprowadzenie danych uwierzytelniających. Można określić tekst prośby i ustawić wykorzystywanie podanych informacji jako danych uwierzytelniających systemu Windows. Dla większości użytkowników konieczność wprowadzania danych uwierzytelniających przy każdym wykonywaniu raportu jest irytująca. Dlatego tę opcję stosuje się rzadko.
Jest ona jednak przydatna np. w jednostce, w której duża liczba pracowników tymczasowych ma pracować przez krótki czas. Konfigurowanie w usłudze Active Directory kont dla wszystkich tych osób to niepraktyczne rozwiązanie. Opcja staje się jeszcze bardziej przydatna, gdy pracownicy korzystają z tego samego komputera (np. na hali produkcyjnej, gdzie przeprowadzana jest konserwacja). Można wtedy wygenerować zestaw identyfikatorów i przydzielić je grupie pracowników. Gdy osoby odpowiedzialne za konserwację wykonują raport, powinny wprowadzić wspólne dla nich dane uwierzytelniające. W raporcie należy wtedy wyświetlać tylko dane przeznaczone dla tej grupy pracowników. Także kierownicy zmian powinni widzieć tylko przeznaczone dla nich informacje itd.
Przy tej metodzie uwierzytelniania nie można stosować nienadzorowanego wykonywania raportów. Więcej o nienadzorowanym wykonywaniu raportów dowiesz się dalej.
- ***Credentials stored securely in the report server.*** Gdy wybrana jest ta opcja, należy wprowadzić określony zestaw danych uwierzytelniających używany do wykonywania raportów. Możesz tu wykorzystać identyfikator z systemu SQL Server lub usługi Active Directory. Jeśli używasz identyfikatora z usługi Active Directory, powinieneś zaznaczyć także opcję *Use as Windows credentials when connecting to the data source*.
Dla źródła danych autorem żądania jest nie użytkownik systemu Windows, ale identyfikator. Jest to wygodne, gdy nie są potrzebne zabezpieczenia danych specyficzne dla użytkownika. Jeśli jednak w źródle danych potrzebne są informacje o użytkowniku systemu Windows, można zaznaczyć opcję *Impersonate the authenticated user after a connection has been made to the data source* (nie jest ona widoczna na rysunku 22.17). Usługa SSRS przekazuje identyfikator użytkownika systemu Windows do źródła danych, wtedy źródło może zwracać dane na podstawie tego identyfikatora (lub przynależności do grupy z usługi Active Directory).
Jest to jedna z dwóch opcji, które umożliwiają wykonywanie raportów w trybie nienadzorowanym.
- ***Windows integrated security.*** Gdy zaznaczona jest ta opcja (na rysunku 22.17 jej nie widać), usługa SSRS może automatycznie wykrywać dane uwierzytelniające z systemu Windows użytkownika korzystającego z tej usługi i przekazywać je do źródła danych. Ta opcja nie umożliwia wykonywania raportów w trybie nienadzorowanym.
- ***Credentials not required.*** Na rysunku 22.17 ta opcja nie jest widoczna. Zdarzają się sytuacje, w których dane uwierzytelniające nie są potrzebne (ani nawet możliwe do zastosowania). Dotyczy to np. raportów generowanych na podstawie plików XML. W XML-u nie występuje uwierzytelnianie. Ten tryb umożliwia wykonywanie raportów w trybie nienadzorowanym.

Do dyspozycji masz dwie podstawowe metody wykonywania raportów. W pierwszej raport jest wykonywany na żądanie użytkownika. Zwykle jest ono zgłaszane w narzędziu Report Manager, ale czasem raporty są też wyświetlane za pomocą aplikacji (np. witryn ASP.NET). Wtedy usługi

SSRS mogą określić użytkownika i przekazać jego dane uwierzytelniające do źródła danych. Jest to tryb *wykonywania nadzorowanego*.

W drugim podejściu to usługi SSRS generują raport. Dzieje się tak, gdy raport lub snapshot są wykonywane według harmonogramu. Tak działa tryb *wykonywania nienadzorowanego*.

Tryby te są powiązane z metodami nawiązywania połączenia. Aby zastosować tryb nienadzorowany, dane uwierzytelniające źródła danych muszą być przechowywane na serwerze lub muszą być zbędne.

Po wprowadzeniu informacji na ekranie *New Data Source* kliknij przycisk OK, aby zachować współużytkowane źródło danych.

Takie same opcje można ustawić dla poszczególnych raportów. W następnym punkcie dowiesz się, gdzie to zrobić. Opcje połączeń są identyczne dla źródeł danych współużytkowanych i przeznaczonych dla konkretnych raportów.

Trzecim elementem menu zadań widocznym w górnej części rysunku 22.11 jest *Report Builder*. Ta opcja powoduje uruchomienie narzędzia Report Builder. Jeśli nie jest zainstalowane, wybranie tej opcji spowoduje jego zainstalowanie.

Opcja *Folder Settings* umożliwia ustawienie zabezpieczeń dla bieżącego katalogu. Przebieg tej operacji przypomina tę samą operację na stronach z ustawieniami witryny (patrz rysunki 22.13 i 22.14), dotyczy jednak bieżącego katalogu i domyślnie także jego podkatalogów. Dostępne są tu nieco inne opcje zabezpieczeń, co pokazano na rysunku 22.18.

Role	Description
<input type="checkbox"/> Browser	May view folders, reports and subscribe to reports.
<input type="checkbox"/> Content Manager	May manage content in the Report Server. This includes folders, reports and resources.
<input type="checkbox"/> My Reports	May publish reports and linked reports; manage folders, reports and resources in a users My Reports folder.
<input type="checkbox"/> Publisher	May publish reports and linked reports to the Report Server.
<input type="checkbox"/> Report Builder	May view report definitions.

Rysunek 22.18. Ustawienia katalogu

Microsoft wykonał dobrą robotę, tworząc ekran widoczny na rysunku 22.18. Widoczne są tu uprawnienia każdej z ról. Nie wymaga to dodatkowych wyjaśnień. Użytkownikowi — jeśli trzeba — można przypisać kilka ról. Podobnie jak przy zabezpieczaniu innych elementów, najlepiej zacząć od minimalnych uprawnień (*Browser*) i zwiększać je, jeśli trzeba.

Report Builder oprócz zapisywania raportów bezpośrednio na serwerze umożliwia zachowywanie ich na twardym dysku. Jest to doskonałe rozwiązanie dla raportów, które są generowane raz, a potem usuwane.

Niektóre firmy stosują ściśle zabezpieczenia serwerów używanych do generowania raportów. Użytkownicy, którzy chcą generować raporty, muszą zapisywać je lokalnie, a następnie przysyłać kompletne raporty (w postaci plików *.RDL*) do działu informatycznego w celu sprawdzenia poprawności rozwiązania.

Aby przesłać plik do narzędzia Report Manager, wykorzystaj opcję *Upload File* dostępną w górnej części strony *Home*. Pojawi się wtedy prosta strona WWW, na której użytkownik może przesłać raport. Przy aktualizowaniu istniejącego raportu należy zaznaczyć opcję *Overwrite*. Jest to mechanizm chroniący przed przypadkowym zastąpieniem istniejących raportów.

W omówieniu definiowania źródła danych pojawiła się opcja *Hide in tile view* (patrz rysunek 22.17). Choć widok kafelków to najczęściej stosowany sposób przeglądania raportów, dostępny jest też widok szczegółów. Aby go użyć, wybierz ostatnią opcję z menu zadań — *Details View*.

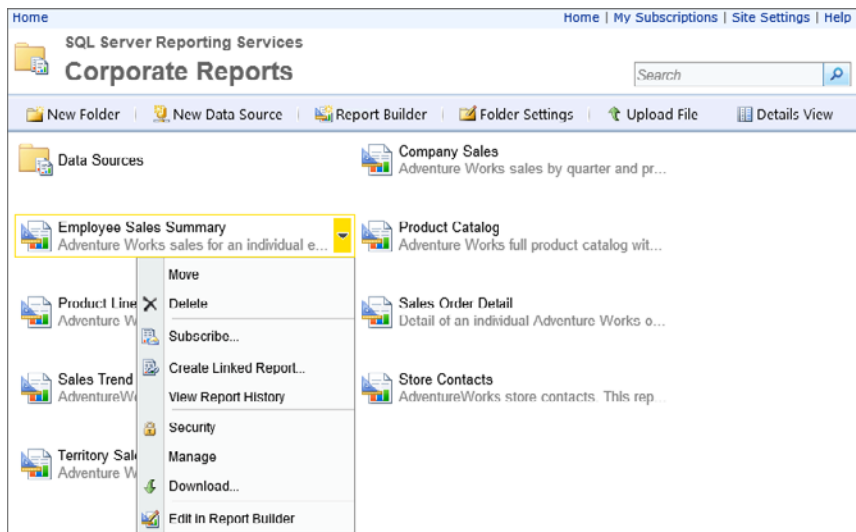
Widok *Details View* udostępnia dodatkowe informacje, np. datę ostatniego wykonania raportu, datę ostatniej jego modyfikacji i autora modyfikacji. Wymienione są też wszystkie obiekty z katalogu, również ukryte. W ten sposób można uzyskać dostęp do wcześniej ukrytych elementów.

Ponadto po lewej stronie paska zadań pojawiają się dwie dodatkowe opcje — *Delete* i *Move*. Gdy zaznaczysz obiekty, te przyciski staną się aktywne i umożliwią trwałe usunięcie wybranych obiektów lub przeniesienie ich do innego katalogu z tej samej witryny narzędzia Report Manager.

Zobaczyłeś już, jak zarządzać narzędziem Report Manager, a także używanymi w nim katalogami, źródłami danych i raportami. W następnym punkcie dokładniej omawiamy zarządzanie raportami.

Zarządzanie raportami

Niezależnie od używanego widoku zacznij od najechania kursorem na raport. Wokół raportu pojawi się wtedy żółta ramka, a po prawej stronie zobaczysz przycisk. Gdy klikniesz ten przycisk, otworzy się menu zarządzania raportem, pokazane na rysunku 22.19.



Rysunek 22.19. Menu zarządzania raportem

Niektóre opcje są oczywiste. *Move* i *Delete* działają podobnie jak w widoku szczegółów. Opcja *Download* umożliwia zapisanie raportu lub źródła danych na dysku twardym. Opcja *Edit in Report Builder* uruchamia narzędzie Report Builder i wczytuje w nim wybrany raport gotowy do edycji.

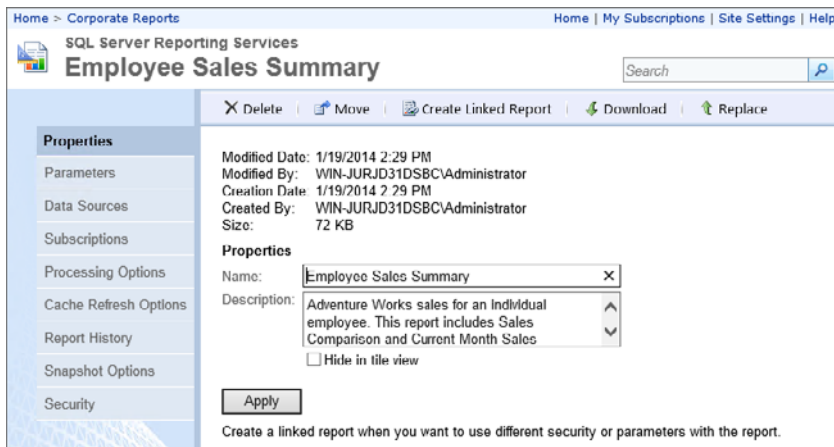
Do zarządzania raportem służy przede wszystkim opcja *Manage*, z którą zapoznasz się wkrótce. *Subscribe*, *View Report History* i *Security* to skróty do stron używanych do zarządzania raportami; ich omówienie znajdziesz dalej.

Na końcu znajduje się opcja *Create Linked Report*. Tworzy ona odnośnik podobny do skrótów z systemu Windows. Powstaje w ten sposób nowy wpis dla raportu, przy czym sam raport nie jest kopiowany. Na stronach zarządzania możesz potem zmienić wszystkie właściwości raportu, co opisujemy dalej. Pozwala to uruchamiać ten sam raport z dwoma różnymi zestawami parametrów, harmonogramów, snapshotów itd.

Tworzenie skrótu do raportu jest proste. Po kliknięciu wspomnianej wcześniej opcji wystarczy określić nazwę skrótu i docelowy katalog. Domyślnie używany jest katalog bieżący, jednak można to zmienić. Następnie należy zmienić odpowiednie właściwości na stronach zarządzania, co opisujemy poniżej.

Właściwości

Gdy klikniesz opcję *Manage*, pojawi się strona *Properties*, pokazana na rysunku 22.20. Powinieneś znać już większość właściwości i instrukcji widocznych w menu zadań, ponieważ występują one także w innych miejscach. Jedyne nowe polecenie to *Replace*; jest to skrót do instrukcji *Upload*. Różnica między nimi polega na tym, że dla instrukcji *Replace* nie jest wyświetlane pole wyboru *Overwrite if exists*. Serwer SSRS uznaje, że jeśli używasz tego polecenia, można bezpiecznie przyjąć, że chcesz zastąpić istniejący raport, dlatego nie wymaga potwierdzenia tego.



Rysunek 22.20. Właściwości raportu

UWAGA Na wszystkich stronach zarządzania należy po wprowadzeniu zmian kliknąć przycisk *Apply*. Jeśli przejdziesz do innej strony bez kliknięcia tego przycisku, modyfikacje zostaną utracone.

Lista widoczna w lewej części ekranu pozwala przejść ze strony *Properties* do następujących stron:

- *Parameters*,
- *Data Sources*,
- *Subscriptions*,

- *Processing Options,*
- *Cache Refresh Options,*
- *Report History,*
- *Snapshot Options,*
- *Security.*

Warto przyjrzeć się każdej z tych stron.

Parametry

Druga strona w obszarze zarządzania, *Parameters*, pozwala przejrzeć wszystkie parametry raportu. Znajdują się tu zarówno ukryte, jak i widoczne parametry. Przy pierwszym otwarciu tej strony widoczne są domyślne wartości ustawione w narzędziach SSDT lub Report Builder. Zmiana wartości na tej stronie prowadzi do zmiany ustawień domyślnych. Możesz też ukryć parametry przed użytkownikami lub je wyświetlić, zmienić tekst pojawiający się przed parametrami, a także zobaczyć typy danych parametrów. Nie można natomiast zmieniać zbioru danych, z którego pochodzą wartości parametru, tworzyć parametrów i ich usuwać.

Źródła danych

Data Sources to następna strona na liście widocznej w lewej części obszaru zarządzania. Przedstawiono ją na rysunku 22.21. W górnej części możesz zaznaczyć opcję *A shared data source* (więcej informacji o konfigurowaniu współużytkowanych źródeł danych znajdziesz wcześniej w tym rozdziale). Gdy ustawisz tę opcję, będziesz mógł kliknąć przycisk *Browse*. Następnie wystarczy znaleźć w strukturze katalogów skonfigurowane wcześniej źródło danych.

The screenshot displays the 'Employee Sales Summary' report configuration interface. On the left, a navigation pane lists various settings: Properties, Parameters, **Data Sources**, Subscriptions, Processing Options, Cache Refresh Options, Report History, Snapshot Options, and Security. The main content area is titled 'AdventureWorks' and contains the following configuration options:

- ☐ **A shared data source**: This option is currently unselected. Below it is a text field showing the path '/Corporate Reports/Data Sources/AdventureWorks' and a 'Browse' button.
- ☒ **A custom data source**: This option is selected. It includes:
 - Data source type**: A dropdown menu set to 'Microsoft SQL Server'.
 - Connection string**: An empty text area with up and down arrow icons on the right.
 - Connect using:**
 - ☒ **Credentials supplied by the user running the report**: This sub-option is selected. It includes a checkbox 'Display the following text to prompt user for a user name and password:' which is checked. Below this is a text box containing the prompt 'Type or enter a user name and password to access the data source'. There is also an unchecked checkbox 'Use as Windows credentials when connecting to the data source'.
 - ☐ **Credentials stored securely in the report server**: This sub-option is unselected. It includes text boxes for 'User name:' and 'Password:', and an unchecked checkbox 'Use as Windows credentials when connecting to the data source'.

Rysunek 22.21. Strona Data Sources

Inna opcja z rysunku 22.21, *Custom data source*, pozwala utworzyć niestandardowe źródło danych specyficzne dla danego raportu. W tym podejściu opcje działają tak samo jak dla współużytkowanego źródła danych.

Sekcja *Connect Using* (też widoczna na rysunku 22.21) informuje źródło danych o tym, w jaki sposób przekazywane są dane uwierzytelniające. Aby generować raport według harmonogramu, należy zastosować dane uwierzytelniające umożliwiające wykonywanie nienadzorowane. Należy wtedy zaznaczyć opcję *Credentials stored securely in the report server*. Gdy jest ona wybrana, można podać dane uwierzytelniające systemu SQL Server lub systemu Windows.

UWAGA Po wprowadzeniu zmian zawsze warto je przetestować, dlatego korzystaj z przycisku *Test Connection*.

Subskrypcje

Subskrypcje to przydatne narzędzie na serwerze SSRS. Umożliwiają one generowanie raportów i dostarczanie użytkownikom bez konieczności interwencji z ich strony. Jest to idealne rozwiązanie dla długo wykonywanych raportów lub gdy raport trzeba generować regularnie (np. codziennie).

Gdy po raz pierwszy otworzysz stronę *Subscriptions*, na liście nie będzie żadnych subskrypcji. Dostępne są dwie opcje tworzenia subskrypcji — *New Subscription* i *New Data-Driven Subscription*.

New Subscription

Nowe subskrypcje pozwalają przysyłać dane w określonym czasie. Jeśli wybierzesz tę opcję, będziesz musiał określić sposób i czas dostarczania danych, co przedstawiono na rysunku 22.22.

The screenshot shows the 'Subscription: Employee Sales Summary' configuration page in SQL Server Reporting Services. The page has a header with navigation links: Home, My Subscriptions, Site Settings, and Help. Below the header, there's a search bar. The main content area is divided into several sections:

- Report Delivery Options:** Includes a dropdown menu for 'Delivered by:' with the option 'Choose a method of delivery' selected.
- Subscription Processing Options:** Includes a section for 'Run the subscription:' with two radio buttons:
 - ☒ When the scheduled report run is complete. This option has a 'Select Schedule' button next to it, and the schedule is set to 'At 8:00 AM every Mon of every week, starting 1/19/2014'.
 - ☐ On a shared schedule. This option has a 'Select a shared schedule' dropdown menu.
- Report Parameter Values:** Includes a section for 'Report Month' with a dropdown menu set to 'July' and a checkbox for 'Use Default'.

Rysunek 22.22. Konfigurowanie nowych subskrypcji

Jeśli na stronie *Reporting Services Configuration Manager's E-mail* podałeś adres e-mail, możesz wybrać jedną z dwóch metod przysyłania danych, czyli na e-mail i za pomocą udziału plikowego systemu Windows. Możesz podać podstawowe informacje na temat dostarczania raportu — jaki

ma być jego format, do kogo powinien trafić itd. Dla udziału plikowego należy podać ścieżkę do katalogu, w którym trzeba zapisać raport, określić format raportu i ustawić dane uwierzytelniające umożliwiające dostęp do tego udziału.

W dolnej części strony *New Subscription* należy ustawić czas wykonywania raportu. Pierwsza opcja umożliwia skonfigurowanie niestandardowego harmonogramu. Druga opcja pozwala wykorzystać współużytkowany harmonogram. Współużytkowane harmonogramy można przygotować w sekcji *Site Settings*, co opisano w punkcie „Menu nawigacyjne”, wcześniej w tym rozdziale.

Nowe subskrypcje sterowane danymi

Drugą opcją są subskrypcje sterowane danymi. Jest to nieco mylące określenie, ponieważ nie chodzi tu o dane źródłowe, tylko o informacje używane do ustawiania opcji raportu w momencie jego wykonywania. Przykładowe opcje raportu uzyskiwane na podstawie danych dotyczą odbiorców raportu, sposobu jego dostarczania i formatu.

Zanim utworzysz subskrypcję sterowaną danymi, musisz przygotować tabelę przeznaczoną na potrzebne dane. Tabela nie ma ustalonego formatu — możesz samodzielnie określić liczbę kolumn, nazwy kolumn i ich wielkość.

Aby utworzyć tabelę, wykonaj następujące kroki.

1. Zaczynaj od utworzenia prostej tabeli przeznaczonej na dane, które można wykorzystać do wysyłania e-maili lub zapisywania raportów na podstawie ścieżki z systemu Windows. W listingu 22.2 (plik *Listing22-2.sql*) przedstawiono kod w języku T-SQL tworzący tę tabelę. Tabelę możesz przechowywać w dowolnym miejscu. W tym przykładzie dla tabeli tworzona jest nowa baza danych ReportSubscriptions.

Listing 22.2. Tworzenie tabeli przeznaczonej na informacje o subskrypcjach

```
CREATE TABLE [dbo].[SubscriptionInfo] (
    [SubscriptionInfoID] [int] NOT NULL PRIMARY KEY
    , [SubscriberName] [nvarchar] (50) NOT NULL
    , [EmailAddress] [nvarchar] (256) NOT NULL
    , [Path] [nvarchar] (256) NOT NULL
    , [FileName] [nvarchar] (256) NOT NULL
    , [Format] [nvarchar] (20) NOT NULL
    , [Comment] [nvarchar] (256) NOT NULL
    , [ReportNameFilter] [nvarchar] (200) NOT NULL
) ON [PRIMARY]
GO
```

1. Następnie umieść w tabeli dane. Zrobisz to z pomocą kodu z listingu 22.3 (plik *Listing22-3.sql*).

Listing 22.3. Zapisywanie informacji o subskrypcjach w tabeli SubscriptionInfo

```
INSERT INTO [dbo].[SubscriptionInfo]
( [SubscriptionInfoID]
, [SubscriberName]
, [EmailAddress]
, [Path]
, [FileName]
, [Format]
, [Comment]
```

```

, [ReportNameFilter]
)
VALUES ('1'
, 'Adam Nowak'
, 'adres@domena.com'
, '\\JURJD31DSBC\FileShare'
, 'Plik Adama'
, 'IMAGE'
, 'Adamie, to Twój raport.'
, 'Sprzedaż wg pracowników')
, ('2'
, 'Marek Kowalski'
, 'adres@domena.com'
, '\\JURJD31DSBC\FileShare'
, 'Dane Marka'
, 'MHTML'
, 'Marku, to Twój raport.'
, 'Sprzedaż wg pracowników'
)
, ('3'
, 'Hans Kloss'
, 'adres@gmail.com'
, '\\JURJD31DSBC\FileShare'
, 'Materiały '
, 'PDF'
, 'Bruner znowu knuje.'
, 'Sprzedaż wg pracowników'
);

```

Gdy dane są już gotowe, możesz skonfigurować subskrypcję sterowaną danymi. Na stronie subskrypcji wybierz opcję *New Data-Driven Subscription*. Teraz wykonaj kroki potrzebne do ustawienia subskrypcji.

3. Pierwszy krok jest przedstawiony na rysunku 22.23. Po określeniu nazwy subskrypcji należy ustawić sposób dostarczania danych. Możesz wybrać e-mail lub udział plikowy z systemu Windows. Wcześniej dla zwykłych subskrypcji (niesterowanych danymi) zastosowano e-mail, dlatego tu wybierz udział plikowy z systemu Windows. Ostatnia opcja pozwala określić źródło danych.

Home > Corporate Reports SQL Server Reporting Services Home | My Subscriptions | Site Settings | Help

Subscription: Employee Sales Summary

Step 1 - Create a data-driven subscription: Employee Sales Summary

Provide a description for this subscription, then choose a delivery extension and data source to use.

Description:

Specify how recipients are notified: Windows File Share

Specify a data source that contains recipient information:

☐ Specify a shared data source

☒ Specify for this subscription only

< Back Next > Cancel Finish

Rysunek 22.23. Konfigurowanie subskrypcji sterowanych danymi

4. Ponieważ źródło danych wybrane w poprzednim kroku dotyczy tylko tej subskrypcji, należy podać informacje o połączeniu, co pokazano na rysunku 22.24. Jeśli wybierzesz współużytkowane źródło danych, na tym etapie możesz wybrać istniejące źródło tego rodzaju.

Rysunek 22.24. Podawanie informacji o połączeniu

UWAGA Jedną z niedogodności jest to, że za każdym razem, gdy wykonujesz ten krok, musisz podać hasło. Nie jest ono zachowywane, dlatego jeśli później zechcesz zmodyfikować subskrypcję, przygotuj się do ponownego wpisania hasła.

5. Teraz wpisz odpowiednią kwerendę w SQL-u, aby zastosować ją do ustawionej wcześniej bazy (patrz rysunek 22.25). Oprócz wpisywania kwerendy możesz też zastąpić domyślne ustawienie limitu czasu. Jeśli kwerenda wymaga dużo czasu, ustaw dłuższy limit. Ponadto zawsze używaj przycisku *Validate* (nie jest widoczny na rysunku), aby mieć pewność, że wprowadzona kwerenda w SQL-u jest prawidłowa.

Ponieważ ekran widoczny na rysunku 22.5 nie pozwala pokazać kompletnej kwerendy, przedstawiamy ją na listingu 22.4 (plik *Listing22-4.sql*). Gdy uruchomisz tę kwerendę, zwróci następujące informacje na temat udziału plikowego: docelową lokalizację wygenerowanego pliku, nazwę tego pliku i jego format. Do nazwy pliku dołączany jest człon z datą. Zwykle warto stosować mechanizm chroniący przed próbą tworzenia w docelowym udziale plikowym powtarzających się plików. Kod z listingu 22.4 dołącza do nazwy pliku liczbę w formacie RRRRMMDD.GGMMSS. Przykładowo wartość 20140131.011221 to 12 minut i 21 sekund po 1.00 dnia 31 stycznia 2014 roku.

Home > Corporate Reports SQL Server Reporting Services

Home | My Subscriptions | Site Settings | Help

Subscription: Employee Sales Summary

Search

Step 3 - Create a data-driven subscription: Employee Sales Summary

Specify a command or query that returns a list of recipients and optionally returns fields used to vary delivery settings and report parameter values for each recipient:

```
SELECT [SubscriberName]
, [Path]
, [FileName] + '' +
CAST(
  (YEAR(GetDate()) * 10000
  + Month(GetDate()) * 100
  + Day(GetDate())
  + CAST(DATEPART(hh, GetDate()) AS decimal) / 100
  + CAST(DATEPART(mi, GetDate()) AS DECIMAL) / 10000
  + CAST(DATEPART(ss, GetDate()) AS DECIMAL) / 1000000)
AS nvarchar(20)) AS [FileName]
```

The delivery extension settings and report parameter values can use field values returned by the command or query. If there are field values that map to these settings, include the fields in your command or query.

The delivery extension has the following settings: FILENAME, PATH, RENDER_FORMAT, WRITEMODE, FILEEXTN, USERNAME, PASSWORD

The report takes the following parameters: ReportMonth, ReportYear, EmployeeID, ShowDescription

Specify a time-out for this command: seconds

Rysunek 22.25. Wprowadzanie kwerendy

Listing 22.4. Kwerenda w subskrypcji sterowanej danymi

```
SELECT [SubscriberName]
, [Path]
, [FileName] + ' ' +
CAST(

  (YEAR(GetDate()) * 10000
  + Month(GetDate()) * 100
  + Day(GetDate())
  + CAST(DATEPART(hh, GetDate()) AS decimal) / 100
  + CAST(DATEPART(mi, GetDate()) AS DECIMAL) / 10000
  + CAST(DATEPART(ss, GetDate()) AS DECIMAL) / 1000000)
AS nvarchar(20)) AS [FileName]
, [Format]
, [Comment]
FROM [dbo].SubscriptionInfo
WHERE [ReportNameFilter] = 'Sprzedaż wg pracowników'
```

Dołączenie daty do nazwy pliku to tylko jedna z metod tworzenia unikatowych nazw. Możesz też zastosować identyfikator GUID lub aktualizować tabelę z danymi o subskrypcjach za pomocą innego procesu, aby za każdym razem otrzymać nową nazwę pliku. Jeszcze inne rozwiązanie polega na używaniu innego procesu do usuwania wszystkich plików z udziału przed zaplanowaną godziną wykonywania raportu.

- Następnie należy podać informacje związane z wybraną metodą generowania raportów. Na rysunku 22.26 przedstawiono ekran dla udziału plikowego z systemu Windows. Można tu podać statyczne wartości (tak jak w polach z nazwą użytkownika i hasłem), a także wykorzystać dane z kwerendy pokazanej w kroku 3. Nazwa pliku, ścieżka i format są pobierane z kwerendy (patrz rysunek 22.26). Można też pominąć niektóre wartości; tu dotyczy to pól z trybem zapisu i rozszerzeniem pliku.

Step 4 - Create a data-driven subscription: Employee Sales Summary

Specify delivery extension settings for Report Server FileShare

File name

☐ Specify a static value:

☒ Get the value from the database:

Path

☐ Specify a static value:

☒ Get the value from the database:

Render Format

☐ Specify a static value:

☒ Get the value from the database:

Write mode

☐ Specify a static value:

☐ Get the value from the database:

☒ No value

File Extension

☐ Specify a static value:

☐ Get the value from the database:

☒ No value

User name

☒ Specify a static value:

☐ Get the value from the database:

Password

☒ Specify a static value:

Rysunek 22.26. Konfigurowanie udziału plikowego, w którym zapisywany jest raport

Ścieżka musi mieć format UNC: \\NAZWASERWERA\\NAZWAUDZIAŁU. W tym przykładzie ścieżka UNC do udziału plikowego (zapisana w tabeli; patrz listing 22.3) to \\WIN-JURJD31DSBC\\FileShare. Jeśli jako sposób dostarczania wybierzesz e-mail, w tym kroku zobaczysz pola związane z wysyłaniem e-maili. Dane z tabeli można pobierać w taki sam sposób jak dla udziału plikowego z systemu Windows (patrz rysunek 22.26).

- Jeśli w raporcie używane są parametry, należy powiązać z nimi wartości z kwerendy w języku SQL. Jest to skuteczna technika pozwalająca na wielokrotne wykorzystanie raportu.

Można np. wygenerować dla każdego menedżera raport z listą pracowników i łączną liczbą godzin przepracowanych w każdym tygodniu. W tabeli z informacjami o subskrypcji należy zapisać nazwisko menedżera, jego adres e-mail i nazwę działu. W subskrypcji sterowanej danymi można powiązać nazwę działu z parametrem raportu, dzięki czemu w raporcie przesyłanym e-mailem do danego menedżera znajdą się tylko pracownicy z określonego działu.

Na rysunku 22.27 przedstawiono ekran przeznaczony do wiązania parametrów z danymi. Możesz podać tu statyczne wartości, zaakceptować ustawienia domyślne lub wykorzystać dane zapisane w bazie (co pokazano w konfiguracji subskrypcji na rysunku 22.26).

- Następnie należy określić czas przesyłania raportu. Przedstawiono to na rysunku 22.28. Jeśli wybierzesz pierwszą opcję (*When the report data is updated on the report server*), usługi SSRS uruchomią przesyłanie danych w ramach subskrypcji w momencie zaktualizowania snapshota raportu. Jest to przydatne, gdy dane są aktualizowane rzadko i nieregularnie.

Home > Corporate Reports SQL Server Reporting Services

Subscription: Employee Sales Summary

Step 5 - Create a data-driven subscription: Employee Sales Summary

Specify report parameter values for Employee Sales Summary

Report Month

☒ Specify a static value: January ☐ Use Default

☐ Get the value from the database: Choose a field

Report Year

☒ Specify a static value: 2014 ☐ Use Default

☐ Get the value from the database: Choose a field

Employee

☒ Specify a static value: 283 ☐ Use Default

☐ Get the value from the database: Choose a field

ShowDescription

☒ Specify a static value: True ☒ False ☐ Use Default

☐ Get the value from the database: Choose a field

Rysunek 22.27. Tu możesz powiązać parametry z danymi

Home > Corporate Reports SQL Server Reporting Services

Subscription: Employee Sales Summary

Step 6 - Create a data-driven subscription: Employee Sales Summary

Specify when the subscription is processed.

Notify recipients:

☐ When the report data is updated on the report server

☒ On a schedule created for this subscription

☐ On a shared schedule: Select a shared schedule

< Back Next > Cancel Finish

Rysunek 22.28. Ustawianie czasu przesyłania raportu

Pozostałe dwie opcje powodują przesyłanie danych według harmonogramu. Ostatnia opcja umożliwia wykorzystanie istniejącego harmonogramu (patrz informacje o sekcji *Site Settings* w punkcie „Menu nawigacyjne”, wcześniej w tym rozdziale). Gdy wybrane są pierwsza lub ostatnia opcja, przycisk *Finish* jest aktywny i można zakończyć pracę.

Jeśli ustawiona jest środkowa opcja, trzeba jeszcze skonfigurować harmonogram dla danej subskrypcji.

9. W ostatnim kroku konfigurowania subskrypcji sterowanych danymi wystarczy podać informacje określające, kiedy i jak często raport ma być wykonywany. Na rysunku 22.29 pokazano, że użytkownik ma dużą swobodę w tym zakresie.

Step 7 - Create a data-driven subscription: Employee Sales Summary

Use the following schedule to determine when the subscription is processed.

Schedule details

Choose whether to run the report on an hourly, daily, weekly, monthly, or one time basis.

All times are expressed in (GMT -05:00) Eastern Standard Time.

☐ Hour
☒ Day
☐ Week
☐ Month
☐ Once

Daily Schedule

☒ On the following days:
☒ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat
☐ Every weekday
☐ Repeat after this number of days:

Start time: : ☒ A.M. ☐ P.M.

Start and end dates

Specify the date to start and optionally end this schedule.

Begin running this schedule on:

☐ Stop this schedule on:

Rysunek 22.29. Konfigurowanie harmonogramu przetwarzania subskrypcji

Cały proces wymagał dużo pracy, ale efekty są tego warte. Subskrypcje sterowane danymi są przydatne, gdy chcesz wielokrotnie wykonywać ten sam raport, używając za każdym razem innych opcji i wartości parametrów. Na rysunku 22.30 przedstawiono listę subskrypcji utworzonych w tym punkcie.

Home > Corporate Reports

SQL Server Reporting Services

Employee Sales Summary

Home | My Subscriptions | Site Settings | Help

Search

	Description	Trigger	Last Run	Status
<input type="checkbox"/>	Employee Sales Summary Data Driven Subscription	Timed Subscription		New Subscription

☐ Edit

Properties
 Parameters
 Data Sources
Subscriptions
 Processing Options
 Cache Refresh Options
 Report History
 Snapshot Options
 Security

Rysunek 22.30. Gotowa subskrypcja na liście

Opcje przetwarzania

Na rysunku 22.31 przedstawiamy wiele różnych opcji przetwarzania. Dotyczą one głównie wyboru między stosowaniem pamięci podręcznej i snapshotów. Pamięć podręczna służy do przechowywania tymczasowych kopii raportów, natomiast snapshoty można zapisać na określony czas. Użytkownicy mają dostęp do wcześniejszych snapshotów raportów, ale raporty

z pamięci podręcznej po wygaśnięciu zostają utracone. Oczywiście możesz zrezygnować z obu tych rozwiązań (służy do tego pierwsza opcja z rysunku 22.31). Gdy nie używasz ani pamięci podręcznej, ani snapshotów, przy każdym wywołaniu raportu usługi SSRS komunikują się z bazą w celu pobrania danych.

Rysunek 22.31. Opcje przetwarzania

Jeśli stosowana jest pamięć podręczna, to w momencie wywołania raportu usługi SSRS najpierw sprawdzają, czy jest on dostępny w pamięci podręcznej. Jeżeli tak, zwracana jest wersja z pamięci podręcznej i nie trzeba komunikować się ze źródłową bazą danych. W przeciwnym razie usługi generują raport i dodają go do pamięci podręcznej. Różnica między dwoma dostępnymi opcjami związanymi z używaniem pamięci podręcznej polega na tym, kiedy ta pamięć wygasa, czy po określonym czasie, czy według ustalonego harmonogramu.

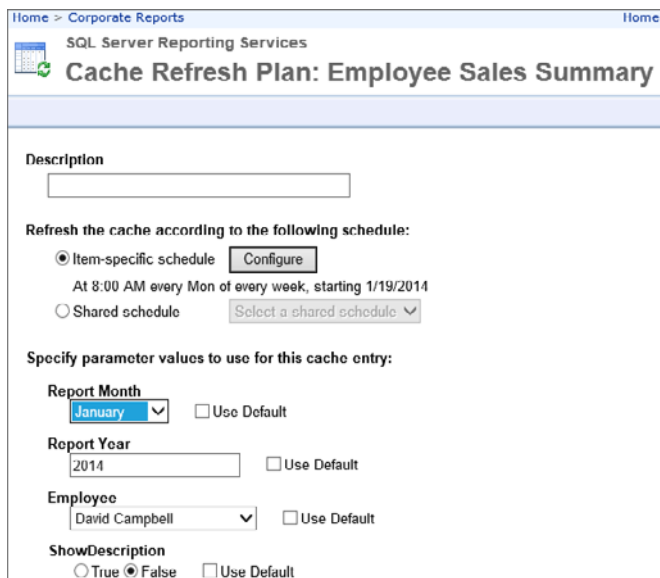
Druga opcja widoczna na rysunku 22.31 umożliwia generowanie raportu na podstawie snapshota. Snapshoty to przechowywane długoterminowo kopie raportu i powiązanych z nim danych. Jeśli wybierzesz tę opcję, będziesz mógł ustawić harmonogram generowania snapshotów.

Ostatnia opcja pozwala zmienić standardowe ustawienia limitu czasu dla danego raportu.

Opcje odświeżania pamięci podręcznej

Raport może trafić do pamięci podręcznej na dwa sposoby. Pierwszy opisano pokrótce w poprzednim punkcie (patrz „Opcje przetwarzania”). Gdy raport jest wykonywany i jest przeznaczony do zapisywania w pamięci podręcznej, ale się w niej nie znajduje, po wygenerowaniu zostaje do niej dodany.

Drugi sposób polega na skonfigurowaniu harmonogramu odświeżania pamięci podręcznej. Przedstawiono to na rysunku 22.32. Gdy stworzysz nowy plan odświeżania, otwierana jest prosta strona WWW, na której możesz określić, czy chcesz utworzyć nowy harmonogram, czy wykorzystać istniejący.



Home > Corporate Reports

SQL Server Reporting Services

Cache Refresh Plan: Employee Sales Summary

Description

Refresh the cache according to the following schedule:

☒ Item-specific schedule [Configure](#)

At 8:00 AM every Mon of every week, starting 1/19/2014

☐ Shared schedule [Select a shared schedule](#)

Specify parameter values to use for this cache entry:

Report Month: [January](#) ☐ Use Default

Report Year: [2014](#) ☐ Use Default

Employee: [David Campbell](#) ☐ Use Default

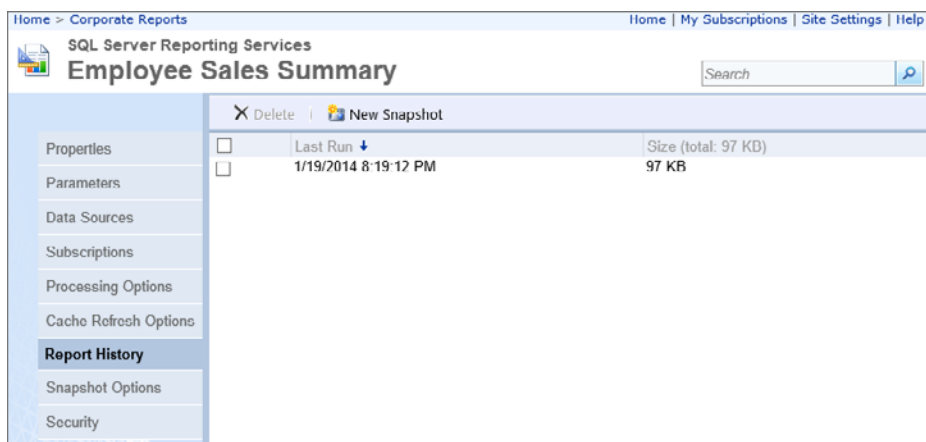
ShowDescription: ☐ True ☒ False ☐ Use Default

Rysunek 22.32. Tworzenie harmonogramu odświeżania pamięci podręcznej

Używanie harmonogramu odświeżania pamięci podręcznej to dobry sposób na zagwarantowanie, że raporty w tej pamięci są aktualne przy jednoczesnym przyspieszeniu udostępniania ich użytkownikom.

Historia raportu

Aby wyświetlić dawne snapshoty raportu, otwórz stronę *Report History*, widoczną na rysunku 22.33.



Home > Corporate Reports

SQL Server Reporting Services

Employee Sales Summary

Search

[Delete](#) [New Snapshot](#)

	Last Run	Size (total: 97 KB)
<input type="checkbox"/>	1/19/2014 8:19:12 PM	97 KB

Properties

Parameters

Data Sources

Subscriptions

Processing Options

Cache Refresh Options

Report History

Snapshot Options

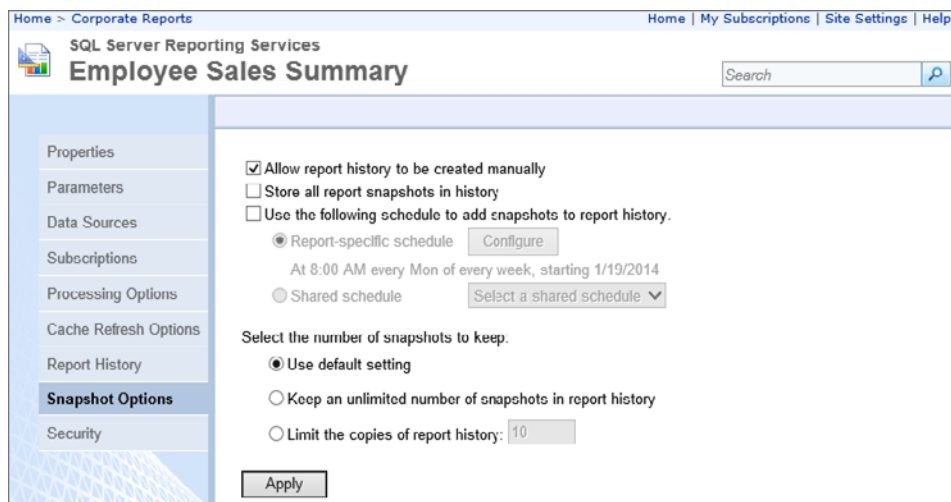
Security

Rysunek 22.33. Historia raportu

Na tej stronie znajduje się lista wszystkich snapshotów raportu. Aby wyświetlić snapshot, kliknij datę i czas w kolumnie *Last run* (patrz rysunek 22.33). Możesz też wygenerować nowy snapshot; w tym celu kliknij przycisk *New Snapshot*. Spowoduje to natychmiastowe wykonanie raportu i zapisanie go jako snapshot.

Opcje snapshotów

Gdy stosujesz snapshoty, musisz nimi zarządzać. Na stronie *Snapshot Options* (patrz rysunek 22.34) można utworzyć harmonogram automatycznego generowania nowych snapshotów. Ponadto można określić, ile snapshotów należy zachować.



Rysunek 22.34. Strona do zarządzania snapshotami

Zabezpieczenia

Security to ostatnia ze stron używanych do zarządzania raportami. Pozwala ona precyzyjnie skonfigurować dostęp do raportu. Okna dialogowe i operacje są tu takie same jak dla katalogów (patrz rysunek 22.18).

Podsumowanie

Usługi SSRS oferują bogaty zestaw narzędzi, który pozwala zarządzać zarówno serwerem SSRS, jak i zapisanymi na nim raportami. Narzędzie Reporting Services Configuration Manager umożliwia konfigurowanie najważniejszych ustawień, np. baz danych potrzebnych usługom SSRS, kluczy szyfrowania, adresów URL itd.

Za pomocą właściwości w programie SQL Server Management Studio można precyzyjnie skonfigurować dany egzemplarz serwera SSRS. Dostępne są tu właściwości związane z limitem czasu wykonywania raportów, rejestrowaniem zdarzeń, historią i zabezpieczeniami.

Report Manager to narzędzie służące do zarządzania raportami i ich wykonywania. Możesz skonfigurować raporty w taki sposób, aby były wykonywane zgodnie z harmonogramem i umieszczane w pamięci podręcznej w celu ponownego wykorzystania. Możesz też zapisywać raporty w formie snapshotów.

Dzięki zdobytej tu wiedzy administratorzy będą dobrze przygotowani do zarządzania usługami SSRS.

Po zapoznaniu się z mechanizmami BI z systemu SQL Server 2014 możesz przejść do rozdziału 23., gdzie opisana została integracja systemu SQL Server z platformą SharePoint.

Integracja systemu SQL Server 2014 z platformą SharePoint 2013

ZAWARTOŚĆ ROZDZIAŁU:

- Zaawansowane funkcje z obszaru raportów i samoobsługowych mechanizmów BI.
- Obsługa platformy SharePoint przez administratorów baz danych.
- Zarządzanie danymi na platformie SharePoint.

Po wprowadzeniu nowych funkcji opisanych w poprzednich rozdziałach Microsoft może wykorzystać wcześniejsze usprawnienia platformy SharePoint, aby zwiększyć możliwości systemu SQL Server 2014 w zakresie BI, wydajności i integracji z narzędziami biurowymi. Użytkownicy domagają się szybszego dostępu do zmieniających się danych, a narzędzia BI z systemu SQL Server 2014 to zapewniają, a ponadto oferują nowe ulepszenia związane z integracją tego systemu z platformą SharePoint 2013.

W platformie SharePoint 2013 wykorzystano możliwości i skalowalność systemu SQL Server, aby udostępniać informacje, dane konfiguracyjne oraz metadane na temat użytkowników i zabezpieczeń. Głównym obszarem integracji z platformą SharePoint 2013 są bazy danych używane przez aplikacje usługowe. W tym rozdziale szczegółowo omawiamy mechanizmy BI systemu SQL Server, które komunikują się z platformą SharePoint i wymagają jej do pełnego wykorzystania swoich możliwości.

Komponenty uczestniczące w integracji

W kontekście integrowania systemu SQL Server i platformy SharePoint ważne są aplikacje usługowe z tej platformy, które komunikują się z produktami lub funkcjami ze środowiska SQL Server. W tej integracji uczestniczy wiele komponentów, w tym aplikacje usługowe, mechanizmy systemu SQL Server i funkcje usług Reporting Services (np. Power View). W następnych punktach szczegółowo omawiamy każdy z tych elementów.

PowerPivot

PowerPivot to bezpłatny i wszechstronny samoobsługowy dodatek do Excela. Służy do pracy z dużymi ilościami danych w Excelu i doskonale nadaje się dla specjalistów BI oraz administratorów baz danych zajmujących się danymi o wydajności i innych licznikami. PowerPivot zapewnia w Excelu liczne możliwości modelu tabelowego usług SSAS z zakresu sortowania danych i manipulowania nimi. Możliwości tego dodatku w połączeniu z SharePointem są znacznie większe. Służy on wtedy do obsługi *osobistych mechanizmów BI* (lub *zespołowych mechanizmów BI*). PowerPivot zapewnia użytkownikom prosty, znany interfejs pozwalający dzielić dane bez konieczności umieszczania ich w hurtowni danych. Nawet gdy celem jest współpraca nad danymi w SharePointie, dokumenty dodatku PowerPivot są tworzone w Excelu. W tym środowisku dane są importowane i kompresowane za pomocą silnika xVelocity, po czym tworzone są relacje, wskaźniki KPI, diagramy i wykresy.

Po zakończeniu tworzenia dokumentu dodatku PowerPivot jest on zapisywany jako plik Excela 2010 lub 2013 (tak samo jak każdy inny skoroszyt Excela). Na tym etapie skoroszyt można przesłać do witryny SharePointa, aby mogły z niego korzystać inne osoby z firmy.

Używanie Excela z SharePointem

Konfigurowanie dodatku PowerPivot pod kątem SharePointa związane jest z pewnymi wymogami. Aplikacja Excel Services musi wyświetlać skoroszyt w przeglądarce, aby użytkownicy mogli go używać. Ponadto potrzebne są usługi SSAS, aby można było uruchamiać interaktywne kwerendy z wykorzystaniem skoroszytu dodatku PowerPivot. Usługi SSAS to główna różnica między wersjami dodatku PowerPivot — kliencką i z SharePointa. Na klienckiej stacji roboczej usługi SSAS działają w kontekście Excela. W SharePointie potrzebny jest odrębny egzemplarz usług SSAS.

Instalowanie

Przed zainstalowaniem dodatku PowerPivot należy przygotować egzemplarz usług SSAS o nazwie *PowerPivot*. Dodatek najłatwiej zainstalować i uruchomić w nowej farmie serwerów SharePointa, jednak można też dodać go w działającej farmie. Wtedy należy określić, że instalacja jest wykonywana w istniejącej farmie.

UWAGA Dodatek PowerPivot dla SharePointa to funkcja przeznaczona dla korporacji, wymagająca, aby serwer został dołączony do domeny. Wymagana jest też platforma SharePoint 2010 SP1 lub SharePoint 2013.

Aby zainstalować dodatek PowerPivot dla SharePointa, wykonaj następujące kroki.

1. Uruchom kreator instalacji platformy SharePoint 2010 SP1 lub SharePoint 2013 (edycja Enterprise lub Enterprise Evaluation).

2. Na jednym z ostatnich ekranów kreatora ustaw późniejszą konfigurację farmy. W tym celu usuń zaznaczenie pola z informacją, że po zakończeniu instalacji uruchomiony ma zostać kreator Product Configuration Wizard. To pozwala zainstalować dodatek PowerPivot i wszystkie potrzebne obiekty bazy danych, a następnie uruchomić kreator Product Configuration Wizard za pomocą instalatora systemu SQL Server.
3. Uruchom kreator SQL Server 2014 Installation Wizard.
4. Na początkowym ekranie wybierz opcję *SQL Server PowerPivot for SharePoint*. Możesz też uwzględnić silnik relacyjnej bazy danych w instalacji. Wtedy silnik bazodanowy jest instalowany dla egzemplarza o nazwie *PowerPivot*. Technikę tę stosuje się w celu konsolidacji baz danych lub gdy silnik bazodanowy nie jest dostępny. Jeśli silnik jest już zainstalowany, nie musisz dodawać nowego. Pamiętaj, gdzie zapisywane będą bazy dla SharePointa, innych egzemplarzy i egzemplarza *PowerPivot*. Będziesz potrzebował tych informacji w trakcie konfigurowania farmy.
5. Ten krok dotyczy *tylko wersji SharePoint 2013*. Pobierz i zainstaluj program *spPowerPivot.msi*, aby umożliwić odświeżanie danych po stronie serwera i zarządzanie dodatkiem. W wersji SharePoint 2010 program *spPowerPivot.msi* nie jest obsługiwany, dlatego nie trzeba go instalować.

Konfigurowanie SharePointa

Po wykonaniu niezbędnych kroków instalacji systemu SQL Server w celu utworzenia egzemplarza dodatku PowerPivot (co opisano w poprzednim punkcie) i po skonfigurowaniu farmy należy utworzyć nową aplikację usługową. Przed dodaniem tej aplikacji trzeba uruchomić jeszcze jeden kreator konfiguracji.

1. Na serwerze, na którym zainstalowałeś dodatek SQL Server PowerPivot for SharePoint, otwórz menu *Start* i przejdź do grupy *Microsoft SQL Server 2014/Configuration Tools*.
2. Wybierz opcję dostosowaną do używanej wersji platformy SharePoint. Dla wersji SharePoint 2010 wybierz *PowerPivot for SharePoint Configuration*, a dla wersji SharePoint 2013 kliknij *PowerPivot for SharePoint 2013 Configuration*.
3. Kreator konfiguracji zbada system i na podstawie obecnej instalacji ustali, jakie kroki trzeba wykonać. Kliknij wszystkie pozycje na lewym panelu i uzupełnij wymagane pola na prawym panelu.
4. Kliknij przycisk *Validate* w prawym dolnym rogu ekranu.
5. Napraw wszystkie błędy wykryte w trakcie sprawdzania poprawności i ponownie kliknij wspomniany przycisk.
6. Kliknij przycisk *Run* (widoczny obok przycisku *Validate*). Uruchomiona zostanie seria poleceń powłoki PowerShell systemu Windows. Spowoduje to skonfigurowanie usługi Secure Store Service, zainstalowanie plików rozwiązania i utworzenie pierwszej aplikacji usługowej dodatku PowerPivot.

W celu utworzenia dodatkowych aplikacji usługowych dodatku PowerPivot wykonaj w oknie *Central Administration* następujące kroki.

1. W oknie *Central Administration* kliknij opcję *Manage Service Applications*. Pojawi się lista aplikacji usługowych.
2. Z menu *New* w lewym górnym rogu ekranu wybierz opcję *SQL Server PowerPivot Service Application*.

3. Aktywuj dodatek PowerPivot w kolekcji witryn. W sekcji *Site Collection Features* w ustawieniach witryny wybierz opcję *Activate* obok tekstu *PowerPivot Feature Integration for Site Collection*.

Ponadto musisz skonfigurować nienadzorowane konto usługowe do obsługi odświeżania danych i podobnych zadań. Zabezpieczenia tego konta trzeba dostosować do danej organizacji. Traktuj je jak zwykłe konto aplikacji, które musi mieć możliwość wczytywania informacji ze źródła danych. Źródłem danych prawdopodobnie jest egzemplarz *PowerPivot* serwera usług Reporting Services. Aby rozpocząć pracę, przygotuj w witrynie lokalizację na skoroszyty dodatku PowerPivot (jest to *galeria dodatku PowerPivot*). W celu dodania nowej lokalizacji w wersji SharePoint 2010 utwórz witrynę, a następnie wybierz opcję *Site Actions/More Options* i poszukaj pozycji *PowerPivot Gallery*. W wersji SharePoint 2013 opcja *PowerPivot Gallery* jest dostępna na ekranie *Add an App*.

Od tego momentu można zarządzać odświeżaniem danych w dodatku PowerPivot w SharePoincie, ustawiać czas przechowywania danych w historii oraz monitorować wydajność i użytkowanie skoroszytów w oknie *Central Administration*.

UWAGA Gdy po raz pierwszy otworzysz galerię dodatku PowerPivot, może pojawić się monit dotyczący instalacji wtyczki Silverlight. Jeśli polityka firmy na to pozwala, zainstaluj tę wtyczkę.

Usługi Reporting Services

Usługi Reporting Services są integralnym elementem zestawu narzędzi BI Microsoftu. W systemie SQL Server 2014 integracja usług Reporting Services odbywa się za pomocą powiązanej z nimi aplikacji usługowej SharePointa.

Architektura systemu SQL Server 2014

Choć usługi Reporting Services można uruchamiać także w trybie natywnym, po zintegrowaniu zostają one powiązane z pozostałymi usługami SharePointa. Najważniejszym wymogiem przy konfigurowaniu usług Reporting Services 2014 w trybie integracji z SharePointem jest używanie wersji SharePoint 2010 SP1 lub SharePoint 2013. Na stronie *Feature Selection* w trakcie instalacji systemu SQL Server trzeba zaznaczyć dwie opcje, czyli *Reporting Services — SharePoint* i *Reporting Services Add-In for SharePoint Products*. Po zakończeniu instalacji dalsze operacje są wykonywane w SharePoincie.

Podobnie jak przy konfigurowaniu innych aplikacji usługowych, takich jak Excel Services, kliknij w oknie *Central Administration* SharePointa przycisk *New* w lewym górnym rogu. Następnie wybierz opcję *SQL Server Reporting Services Service Application*. Wszystkie ustawienia związane z konfiguracją usług Reporting Services znajdują się w oknie *Central Administration*. Można tu ustawić pulę aplikacji, konto zabezpieczeń i lokalizację bazy usług Reporting Services.

To podejście ma wiele zalet. Jak wcześniej wspomniano, lokalizację baz usług Reporting Services ustawia się w oknie *Central Administration* SharePointa. Jest to wygodne, ponieważ bazy można łatwo umieścić na dowolnym serwerze (także na tym, który zawiera bazy z danymi SharePointa) bez konieczności wychodzenia poza platformę SharePoint. Aplikacje usługowe to programy

działające w SharePointcie. Łatwo można skonfigurować wiele aplikacji usługowych dla usług Reporting Services i korzystać z nich w farmie serwerów. W tej sytuacji każda aplikacja korzysta z własnego zestawu baz danych z zaplecza (bazy nie są współużytkowane).

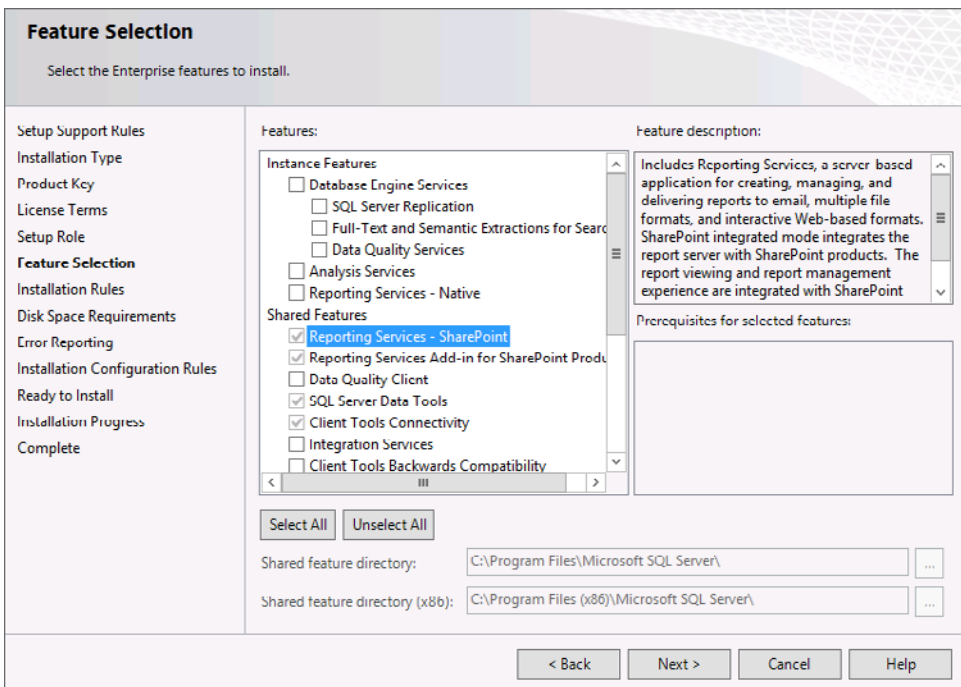
Ponieważ w trybie integracji z platformą SharePoint wszystkie elementy usług Reporting Services działają w tej platformie, zarządzanie usługami jest prostsze. Także skalowanie można przeprowadzić po stronie SharePointa, zamiast osobno skalować usługi Reporting Services i sam SharePoint.

Instalacja

Od wersji SQL Server 2012 instalowanie i konfigurowanie usług Reporting Services w SharePointcie zostało usprawnione. Te poprawki są dostępne także w wersji SQL Server 2014. Konfigurowanie usług SQL Server Reporting Services (SSRS) i zarządzanie nimi odbywa się teraz w oknie *Central Administration* SharePointa. Konieczne jest jednak uruchomienie instalatora systemu SQL Server i dodanie dwóch współużytkowanych funkcji, czyli *ReportingServices — SharePoint* i *Reporting Services Add-in for SharePoint Products*.

Aby zainstalować i skonfigurować usługi Reporting Services w SharePointcie, wykonaj następujące kroki.

1. Uruchom instalację systemu SQL Server i na ekranie *Setup Role* wybierz opcję *SQL Server Feature Installation*.
2. Na następnym ekranie (*Feature Selection*) zaznacz pola wyboru obok pozycji *ReportingServices — SharePoint* i *Reporting Services Add-in for SharePoint Products* w sekcji *Shared Features* (patrz rysunek 23.1).



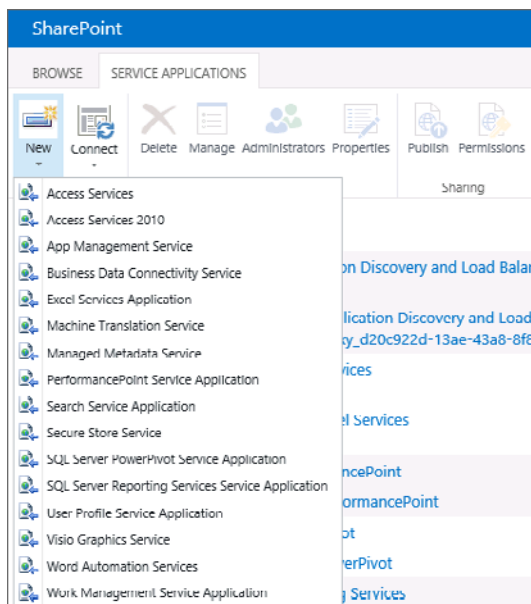
Rysunek 23.1. Opcje potrzebne do zintegrowania usług Reporting Services z SharePointem

3. Zaakceptuj pozostałe domyślne ustawienia i kliknij przycisk *Install* na ostatniej stronie kreatora.
4. Po zakończeniu instalacji uruchom za pomocą menu *Start* powłokę *Management Shell* platformy SharePoint 2013.
5. W oknie poleceń powłoki PowerShell uruchom instrukcję `Install-SPRSService`, aby zainstalować usługę SharePointa (patrz rysunek 23.2). Jeśli nie pojawiły się żadne komunikaty, oznacza to, że polecenie zostało wykonane z powodzeniem.



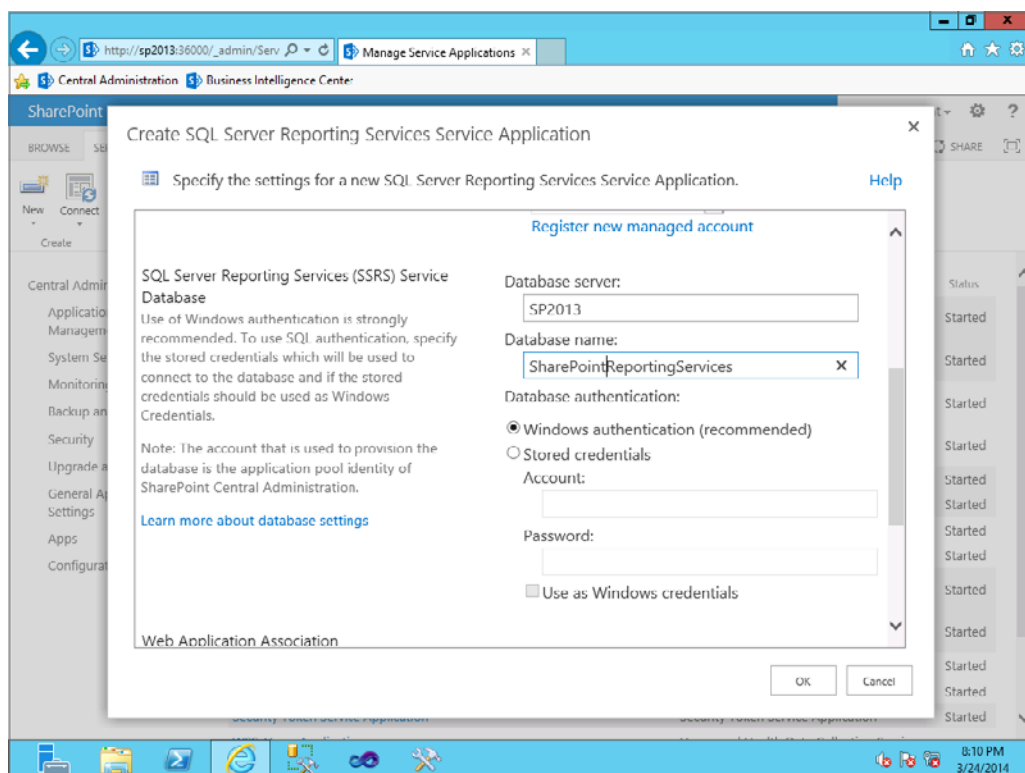
Rysunek 23.2. Wywołaj odpowiednie instrukcje w powłoce

6. W oknie poleceń powłoki PowerShell wywołaj instrukcję `Install-SPRSServiceProxy`, aby zainstalować usługę pośrednika (patrz rysunek 23.2). Jeśli nie pojawiły się żadne komunikaty, oznacza to, że polecenie zostało wykonane z powodzeniem.
7. Otwórz okno *Central Administration* SharePointa.
8. Kliknij przycisk *Manage Service Applications*, aby wyświetlić listę aplikacji usługowych.
9. W menu *New* w lewym górnym rogu ekranu wybierz opcję *SQL Server Reporting Services Service Application*, co pokazano na rysunku 23.3.



Rysunek 23.3. Tworzenie nowej aplikacji usługowej

10. Na wyświetlonej stronie podaj odpowiednią pulę aplikacji i informacje o serwerze bazodanowym (patrz rysunek 23.4).



Rysunek 23.4. Określanie ustawień nowej aplikacji bazodanowej

11. W sekcji *Web Application Association* zaznacz pola obok aplikacji sieciowych, które mają mieć dostęp do aplikacji powiązanej z usługami Reporting Services.
12. Kliknij przycisk OK.

Power View

Power View to rozszerzenie usług Reporting Services. Współdziała z SharePointem i umożliwia interaktywne generowanie jednorazowych raportów. Rozszerzenie Power View jest obsługiwane w edycjach Developer, Enterprise i Business Intelligence systemu SQL Server 2014. Ponadto wymaga edycji Enterprise platformy SharePoint 2010 lub 2013 i włączonego dodatku Reporting Services dla SharePointa. Raporty Power View można tworzyć na podstawie modelu tabelowego i wielowymiarowych kostek.

Czym jest rozszerzenie Power View?

Rozszerzenie Power View to innowacyjne narzędzie do korzystania z danych za pomocą stron generowanych przez wtyczkę Silverlight. Przy jego użyciu raporty są tworzone i zapełniane na podstawie skoroszytów dodatku PowerPivot zapisanych w galerii w SharePointcie lub za pomocą połączeń z wielowymiarowymi kostkami i modelami tabelowymi z usług SSAS 2014.

Takie raporty nie mają zastępować tradycyjnych raportów usług Reporting Services i narzędzia Report Builder. Power View służy do szybkiej eksploracji danych i poruszania się po nich w celu szukania powiązań. Pliki generowane przez narzędzia powiązane z usługami Reporting Services mają format RDL (ang. *Report Definition File*) i nie mogą być edytowane ani wyświetlane w rozszerzeniu Power View. Ponieważ to rozszerzenie jest używane tylko w SharePointcie, nie we wszystkich sytuacjach można je zastosować. Każda z dostępnych technologii ma swoje zastosowania w środowisku generowania raportów.

Warstwa prezentacji

Usługi Reporting Services generują tradycyjne raporty, natomiast program Power View wyświetla prezentacje w trybie aktywnym, co nie wymaga generowania podglądu raportów. Dane zawsze są udostępniane „na żywo”. Dodatkową korzyścią jest możliwość wyeksportowania raportu z programu Power View do PowerPointa. W tym podejściu każdy raport staje się odrębnym slajdem. Widok pełnoekranowy i slajdy PowerPointa działają bardzo podobnie. W obu przypadkach możliwa jest interakcja z danymi (np. stosowanie filtrów dodanych w trakcie budowania raportu) i używanie różnych wizualizacji. W tym stanie dane są interaktywne, ale nie można dodawać nowych filtrów ani wizualizacji.

Oto wybrane wizualizacje dostępne w programie Power View:

- tabele,
- wykresy,
- wykresy bąbelkowe,
- wykresy punktowe,
- karty,
- kafelki.

Tworzenie

Przy inicjowaniu tworzenia raportów w programie Power View można wykorzystać elementy modelu tabelowego, pliki *.xlsx*, a także pliki połączeń prowadzące do modelu tabelowego lub kostek wielowymiarowych z usług SSAS. Cały proces odbywa się w bibliotece dokumentów SharePointa lub galerii dodatku PowerPivot. Gdy klikniesz strzałkę obok odpowiedniego elementu, pojawi się opcja tworzenia raportu programu Power View. Gdy wybierzesz tę opcję, pojawi się okno projektowe programu Power View. Możesz też edytować istniejące raporty. W tym celu kliknij strzałkę obok raportu i wybierz opcję *Edit in Power View*.

Gdy zapisujesz raporty programu Power View, mają one format RDLX. Dostępna jest też opcja zapisywania powiązanych rysunków. Jeśli w trakcie eksportowania raportu do PowerPointa nie użyjesz tej opcji, w prezentacji pojawią się puste miejsca na rysunki.

Odświeżanie danych

Zadaniem specjalistów systemu SQL Server jest umożliwianie szybkiego i wydajnego pobierania danych. Pozwala to zapewnić efektywną pracę środowiska z danymi i korzystających z niego aplikacji. Wcześniej w tym rozdziale zapoznałeś się już z mechanizmami przydatnymi administratorom baz danych (np. z dodatkiem PowerPivot i programem Power View). Tu omawiamy dodatkowe narzędzia. W tym podrozdziale koncentrujemy się na zarządzaniu

odświeżaniem danych w raportach i SharePointcie. Istnieje wiele źródeł danych, dlatego najlepiej skoncentrować się na danych przechowywanych zwykle poza systemem SQL Server. Oto przykładowe źródła:

- usługi Excela,
- usługi PerformancePoint,
- usługi Visio,
- dodatek PowerPivot.

Używanie połączeń z danymi w Excelu

Każdy skoroszyt Excela korzystający z zewnętrznych danych obejmuje połączenie ze źródłem danych. Połączenia zawierają wszystkie elementy potrzebne do komunikowania się z zewnętrznym źródłem i pobierania z niego danych. Oto niezbędne elementy:

- *łańcuch znaków połączenia*, określający, z którym serwerem należy się połączyć i jak to zrobić,
- *kwerenda*, czyli łańcuch znaków określający pobierane dane,
- inne informacje potrzebne do pobrania danych (np. używane konto, tryb pośrednika itd.),
- połączenia zagnieżdżone i odnośniki do połączeń.

Skoroszyty Excela mogą zawierać zagnieżdżone połączenia i odnośniki do zewnętrznych połączeń. Połączenia zagnieżdżone są przechowywane wewnątrz skoroszytu. Połączenia zewnętrzne mają postać plików ODC (ang. *Office Data Connection*), które można wskazywać w skoroszytcie.

Połączenia zagnieżdżone i odnośniki do połączeń działają w ten sam sposób. W obu przypadkach należy podać wszystkie parametry potrzebne do nawiązania połączenia z danymi. Zewnętrzne pliki połączeń można przechowywać w centralnym magazynie i zabezpieczać. Można też nimi zarządzać i korzystać z nich wielokrotnie. Jest to dobre rozwiązanie, gdy planujesz ogólny mechanizm obsługujący łączenie się dużej grupy użytkowników z zewnętrznymi danymi. Więcej informacji na ten temat znajdziesz na stronie <http://technet.microsoft.com/en-us/library/ff604007.aspx#section4>.

Dla pojedynczych połączeń w skoroszytcie można utworzyć zagnieżdżoną kopię informacji o połączeniu i odnośnik do zewnętrznego pliku połączenia. Połączenie można tak skonfigurować, aby zawsze używało zewnętrznego pliku połączenia do odświeżania danych z zewnętrznych źródeł. Jeśli w tym podejściu nie można pobrać zewnętrznego pliku połączenia lub nie może on nawiązać połączenia ze źródłem danych, skoroszyt nie może uzyskać dostępu do nowych danych. Skutki takiej sytuacji zależą od ustawień połączenia. Jeśli nie jest ono skonfigurowane tak, że może używać tylko zewnętrznego pliku połączenia, Excel próbuje zastosować zagnieżdżoną kopię połączenia. Jeżeli to się nie powiedzie, Excel spróbuje wykorzystać plik połączenia do nawiązania komunikacji z zewnętrznym źródłem danych.

Aby zwiększyć bezpieczeństwo, możesz tak skonfigurować usługi Excela, aby umożliwiała używanie tylko plików połączeń. Wtedy w skoroszytach wczytanych na serwer SharePointa wszystkie połączenia zagnieżdżone są ignorowane. Próba nawiązania połączenia jest podejmowana tylko wtedy, gdy dostępny jest odnośnik do poprawnego pliku połączenia, uznanego za zaufany przez administratora serwera. Więcej informacji na ten temat zawiera strona <http://technet.microsoft.com/en-us/library/ff604007.aspx#section4>.

OSTRZEŻENIE Usługi Excela domyślnie są niedostępne w SharePoincie. Musisz je włączyć i skonfigurować.

Dostawcy danych

Dostawcy danych to sterowniki używane przez aplikacje (np. Excel i usługi Excela) do łączenia się z określonymi źródłami danych. Przykładowo specjalny dostawca MSOLAP umożliwia połączenie się z usługami Microsoft SQL Server 2008 Analysis Services (SSAS). Dostawcę należy ustawić w łańcuchu znaków połączenia.

Dostawcy danych obsługują kwerendy, przetwarzanie łańcuchów znaków połączeń i inne operacje związane z połączeniami. Tych zadań nie wykonują usługi Excela — nie mogą one kontrolować pracy dostawców danych.

Każdego dostawcę danych używanego w usługach Excela trzeba bezpośrednio ustawić jako zaufanego. Na stronie <http://technet.microsoft.com/en-us/library/ff191200.aspx> dowiesz się, jak dodać nowego dostawcę danych do listy zaufanych dostawców.

W usługach Excela domyślnie wielu popularnych dostawców danych jest ustawionych jako zaufani. Zwykle nie trzeba dodawać nowych dostawców. Przeważnie ustawia się ich tylko w niestandardowych rozwiązaniach.

Uwierzytelnianie dostępu do zewnętrznych danych

Serwery bazodanowe wymagają, aby użytkownik był *uwierzytelniony* (czyli by udowodnił serwerowi swoją tożsamość). Następnym krokiem jest *autoryzacja* (czyli określenie przez serwer, jakie operacje dany użytkownik może wykonywać).

Uwierzytelnianie jest niezbędne, aby serwer mógł przeprowadzić autoryzację, a także do działania zabezpieczeń chroniących dane przed dostępem nieuprawnionych osób.

Usługi Excela obsługują następujące metody uwierzytelniania.

- **Uwierzytelnianie z systemu Windows.** Usługi Excela mogą używać zintegrowanego uwierzytelniania z systemu Windows i próbować łączyć się ze źródłem danych na podstawie udostępnionych przez system Windows danych użytkownika, który wyświetlił skoroszyt.
- **Aplikacja Secure Store Service (SSS).** Usługi Excela mogą korzystać z danych uwierzytelniających powiązanych z aplikacją docelową w narzędziu SSS.
- **Brak.** Usługi Excela wykorzystują nienadzorowane konto usługowe i przekazują łańcuch znaków połączenia do źródła danych.

Metodę uwierzytelniania należy ustawić w Excelu. Jest ona właściwością połączenia z zewnętrznymi danymi. Domyślnie stosowane jest uwierzytelnianie z systemu Windows.

Zintegrowane uwierzytelnianie z systemu Windows

Jeśli wybierzesz opcję uwierzytelniania z systemu Windows, usługi Excela spróbują przekazać do zewnętrznego źródła danych pobraną z systemu Windows tożsamość użytkownika wyświetlającego skoroszyt. Gdy źródło danych obsługuje zintegrowane uwierzytelnianie i jest zlokalizowane na serwerze innym niż ten, na którym działają usługi obliczeniowe Excela, stosowany jest mechanizm delegowania z wykorzystaniem protokołu Kerberos.

W większości środowisk korporacyjnych usługi obliczeniowe Excela i źródła danych znajdują się na różnych komputerach. To oznacza, że do nawiązywania połączeń z uwierzytelnianiem z systemu Windows niezbędne jest delegowanie z wykorzystaniem protokołu Kerberos (zalecane jest delegowanie ograniczone). Więcej informacji o tym, jak skonfigurować delegowanie ograniczone z wykorzystaniem protokołu Kerberos dla usług Excela znajdziesz na stronie <http://technet.microsoft.com/en-us/library/ff829837.aspx>.

Secure Store Service (SSS)

SSS to aplikacja usługowa z platformy SharePoint Server 2010 używana do przechowywania w bazie zaszyfrowanych danych uwierzytelniających stosowanych do uwierzytelniania programów. Usługi Excela korzystają z narzędzia SSS do przechowywania i pobierania informacji używanych do uwierzytelniania w zewnętrznych źródłach danych.

Jeśli wybrałeś narzędzie SSS, musisz podać identyfikator aplikacji docelowej. Ustawiona aplikacja docelowa jest używana przy wyszukiwaniu i pobieraniu odpowiedniego zestawu danych uwierzytelniających. Dla każdej aplikacji docelowej można określić zestaw uprawnień, tak aby tylko wybrani użytkownicy (lub ich grupy) mogli korzystać z zapisanych danych uwierzytelniających.

Usługi Excela na podstawie identyfikatora aplikacji mogą pobrać z bazy narzędzia SSS dane uwierzytelniające użytkownika, który korzysta ze skróty (za pomocą przeglądarki lub usług sieciowych Excela). Następnie usługi Excela mogą wykorzystać dane uwierzytelniające do uwierzytelnienia się w źródle danych i pobrania informacji.

Omówienie stosowania narzędzia SSS z usługami Excela zawiera strona <http://technet.microsoft.com/en-us/library/ff191191.aspx>.

Brak

Jeśli wybierzesz tę metodę, nie następuje pobieranie danych uwierzytelniających ani nie są wykonywane żadne specjalne operacje w związku z uwierzytelnianiem połączenia. Usługi Excela nie próbują delegować danych uwierzytelniających lub pobierać danych zapisanych dla użytkownika w bazie narzędzia SSS. Zamiast tego usługi Excela wykorzystują nienadzorowane konto usługowe i przekazują łańcuch znaków połączenia do dostawcy danych, który obsługuje uwierzytelnianie.

Łańcuch znaków połączenia może obejmować nazwę użytkownika i hasło potrzebne do łączenia się ze źródłem danych albo na podstawie informacji z systemu Windows określać tożsamość użytkownika lub komputera chcącego nawiązać połączenie. W obu przypadkach najpierw ustawiane jest nienadzorowane konto, a później następuje próba połączenia się ze źródłem danych. Łańcuch znaków połączenia i dostawca określają metodę autoryzacji. Ponadto przy autoryzacji można wykorzystać dane uwierzytelniające z łańcucha znaków połączenia albo ustaloną za pomocą systemu Windows tożsamość używanego nienadzorowanego konta.

Zabezpieczenia w usługach Excela i dane zewnętrzne

Usługi Excela zarządzają skrótytami i połączeniami z zewnętrznymi danymi za pomocą kilku mechanizmów. Oto one.

- **Zaufane lokalizacje plików.** Są to podane przez administratora lokalizacje, z których usługi Excela mogą wczytywać skrótyty.
- **Zaufane biblioteki połączeń.** Są to biblioteki połączeń z danymi z SharePointa, które zostały bezpośrednio ustawione przez administratora jako zaufane. Usługi Excela mogą wczytywać pliki połączeń z tych bibliotek.

- **Zaufani dostawcy danych.** Są to dostawcy danych ustawieni jako zaufani przez administratora.
- **Nienadzorowane konto usługowe.** Jest to konto o niskich uprawnieniach, z którego usługi Excela mogą korzystać przy łączeniu się z danymi.

Zaufane lokalizacje plików

Usługi Excela wczytują skoroszyty tylko z zaufanych lokalizacji. *Zaufana lokalizacja plików* to lokalizacja na serwerze SharePointa, sieciowy udział plikowy lub adres katalogu sieciowego, bezpośrednio ustawione przez administratora jako miejsce, z którego można wczytywać skoroszyty. Podane w ten sposób katalogi są dodawane do wewnętrznej listy usług Excela (do *listy zaufanych lokalizacji plików*).

W zaufanych lokalizacjach można ustawić zestaw ograniczeń dotyczących wczytywanych skoroszytów. Wszystkie skoroszyty pobierane z zaufanej lokalizacji są zgodne z tymi ustawieniami. Oto krótka lista ustawień wpływających na zewnętrzne dane.

- **Allow External Data.** Określa sposób dostępu do zewnętrznych danych.
 - **None** (jest to ustawienie domyślne). Akceptowane są tylko pliki połączeń z zaufanej biblioteki połączeń SharePointa.
- **Warn on Refresh.** Określa, czy przy odświeżaniu danych wyświetlane mają być ostrzeżenia.
- **Stop When Refresh on Open Fails.** Określa, czy wczytywanie skoroszytu ma się zakończyć niepowodzeniem, jeśli w momencie jego otwierania nie można odświeżyć zewnętrznych danych. Ta opcja jest używana, gdy skoroszyt przechowuje w pamięci podręcznej wyniki modyfikowane w zależności od osoby, która wyświetla dany skoroszyt. Celem jest ukrycie wyników z pamięci podręcznej i upewnienie się, że przeglądający skoroszyt użytkownik widzi tylko przeznaczone dla niego dane. Opcja sprawia, że jeśli dla skoroszytu ustawione jest odświeżanie danych przy otwieraniu, a danych nie da się odświeżyć, skoroszyt nie jest wyświetlany.
- **External Data Cache Lifetime.** Określa czas wygasania pamięci podręcznej z zewnętrznymi danymi. Dane są współużytkowane na serwerze przez wiele osób, co pozwala poprawić skalowalność i wydajność rozwiązania. Czas przechowywania danych w pamięci podręcznej można zmienić. Jest to przydatne w sytuacji, gdy liczbę wywołań kwerendy należy ograniczyć do minimum, ponieważ wykonywanie danej kwerendy wymaga dużo czasu. W takim scenariuszu dane są modyfikowane raz na dzień, tydzień lub miesiąc, a nie co minutę lub co godzinę.

Zaufana biblioteka połączeń i połączenia zarządzane

Biblioteka połączeń z danymi to biblioteka z SharePointa przeznaczona do przechowywania plików połączeń używanych w aplikacjach z pakietu Office (takich jak Excel lub Visio). Usługi Excela wczytują pliki połączeń tylko z zaufanych bibliotek połączeń z SharePointa. *Zaufana biblioteka połączeń z danymi* to biblioteka bezpośrednio dodana przez administratora do wewnętrznej listy zaufanych lokalizacji. Biblioteki połączeń umożliwiają scentralizowane zarządzanie połączeniami, zabezpieczanie ich, przechowywanie i wielokrotne używanie.

- **Zarządzanie połączeniami.** Ponieważ skoroszyty zawierają odnośnik do pliku z biblioteki połączeń, to gdy coś w połączeniu się zmieni (np. nazwa serwera lub identyfikator aplikacji w narzędziu SSS), wystarczy zmodyfikować jeden plik zamiast wielu skoroszytów. W skoroszytach zmiany są wprowadzane automatycznie, gdy skoroszyt korzysta z danego pliku połączenia przy odświeżaniu danych w Excelu lub usługach Excela.

- **Zabezpieczanie połączeń.** Biblioteka połączeń w SharePointcie obsługuje wszystkie uprawnienia SharePointa (w tym uprawnienia dla katalogów i dla elementów). Dzięki temu biblioteka połączeń może stać się chronionym i ściśle kontrolowanym magazynem połączeń. Użytkownikom można przyznać dostęp do niej w trybie tylko do odczytu. Użytkownicy mogą zatem używać połączeń, ale nie mają uprawnień do ich dodawania. Jeśli zastosujesz listy ACL (ang. *Access Control List*) dla biblioteki połączeń i pozwolisz dodawać połączenia tylko zaufanym osobom, biblioteka stanie się magazynem zaufanych połączeń.

Możesz tak skonfigurować usługi Excela, aby wczytywały pliki połączeń tylko z bibliotek bezpośrednio ustawionych przez administratora jako zaufane i blokowały połączenia zagnieżdżone. W takiej konfiguracji usługi Excela wykorzystują bibliotekę połączeń jako dodatkową warstwę zabezpieczeń połączeń z danymi.

Bibliotekę połączeń można stosować razem z nową rolą *Viewer* z systemu SharePoint Server, aby umożliwić odświeżanie skoroszytów wyświetlanych w przeglądarce przez usługi Excela. Gdy stosowana jest ta rola, użytkownicy nie mają dostępu do zawartości plików połączeń z poziomu aplikacji klienckiej (np. z Excela). Chroni to zawartość pliku połączenia, a przy tym umożliwia odświeżanie skoroszytów na serwerze.

- **Przechowywanie połączeń.** Przechowywanie połączeń z danymi to następna ważna funkcja bibliotek dokumentów. Połączenia z danymi są zapisywane w bibliotece, tak jak dokumenty lub rysunki, oraz są udostępniane usługom i raportom z farmy (zgodnie z ich uprawnieniami).
- **Wielokrotne używanie połączeń.** Użytkownicy mogą ponownie wykorzystać połączenia utworzone przez inne osoby i generować różne raporty na podstawie tego samego źródła danych. Możliwe, że dział informatyczny lub ekspert BI tworzy połączenia, a pozostałe osoby mogą z nich korzystać bez konieczności poznawania dostawców danych, nazw serwerów lub kwestii związanych z uwierzytelnianiem. Lokalizację biblioteki można nawet opublikować w aplikacjach klienckich z pakietu Office, dzięki czemu w Excelu i innych aplikacjach klienckich używających biblioteki wyświetlane są połączenia z danymi.

Zaufani dostawcy danych

Usługi Excela używają tylko tych zewnętrznych dostawców danych, którzy znajdują się na wewnętrznej liście zaufanych dostawców. Jest to mechanizm zabezpieczający chroniący serwer przed używaniem dostawców, którym administrator nie ufa.

Nienadzorowane konto usługowe

Usługi Excela używają konta o wysokich uprawnieniach. Ponieważ usługi te nie mają kontroli nad dostawcą danych i nie przetwarzają bezpośrednio łańcuchów znaków połączeń specyficznych dla dostawców, korzystanie z tego konta przy dostępie do danych powoduje zagrożenie bezpieczeństwa. Aby ograniczyć ryzyko, usługi Excela mogą korzystać z *nienadzorowanego konta usługowego*. Jest to konto o niskich uprawnieniach używane przez usługi Excela w następujących warunkach.

- Przy próbie nawiązania połączenia, gdy wybrana jest opcja bez uwierzytelniania.
- Gdy używane jest narzędzie SSS, a zapisane dane uwierzytelniające nie są danymi z systemu Windows.
- Jeśli wybrana jest opcja bez uwierzytelniania, a konto nienadzorowane nie zapewnia dostępu do źródła danych, usługi Excela korzystają z tego konta i używają informacji z łańcucha znaków połączenia do nawiązania komunikacji ze źródłem danych.

- Jeśli wybrana jest opcja bez uwierzytelniania i konto nienadzorowane zapewnia dostęp do źródła danych, komunikacja jest nawiązywana za pomocą danych uwierzytelniających nienadzorowanego konta usługowego. Zachowaj ostrożność w trakcie projektowania rozwiązań, w których to konto jest celowo używane do łączenia się z danymi. Konto to może być stosowane przez wszystkie skrośzty z serwera. Każdy użytkownik może utworzyć skrośzt niewymagający uwierzytelniania i za pomocą usług Excela wyświetlić dane. W niektórych sytuacjach jest to pożądane rozwiązanie. Jednak lepiej stosować hasła użytkowników i grup oraz zarządzać nimi za pomocą narzędzia SSS.
- Gdy używane jest narzędzie SSS i przechowywane dane uwierzytelniające nie są danymi z systemu Windows, usługi Excela korzystają z nienadzorowanego konta usługowego i próbują nawiązać połączenie ze źródłem danych za pomocą zapisanych danych.
- Jeśli używane jest uwierzytelnianie z systemu Windows (lub narzędzie SSS i dane uwierzytelniające z systemu Windows), nienadzorowane konto usługowe nie jest stosowane. Zamiast niego usługi Excela posługują się tożsamością określoną za pomocą systemu Windows i na tej podstawie próbują połączyć się ze źródłem danych.

Połączenia danych z narzędzia PerformancePoint

PerformancePoint to wbudowane w SharePointa narzędzie do tworzenia arkuszy wyników i paneli administracyjnych. Choć to narzędzie (w odróżnieniu od dodatku PowerPivot i usług Reporting Services) nie wymaga instalowania specjalnych komponentów systemu SQL Server, często komunikuje się z systemem SQL Server. Źródłami danych mogą być bazy systemu SQL Server, wielowymiarowe kostki usług Analysis Services lub modele tabelowe.

W narzędziu PerformancePoint trzeba utworzyć połączenia ze źródłami danych, które mają być używane w panelu administracyjnym. Wszystkie dane w narzędziu PerformancePoint pochodzą z zewnątrz (są przechowywane w repozytoriach poza tym narzędziem). Po nawiązaniu połączenia z danymi można je wykorzystać w różnych mechanizmach tego narzędzia.

PerformancePoint obsługuje tabelowe źródła danych (listy SharePointa, usługi Excela, tabele systemu SQL Server i skrośzty Excela), a także źródła danych z usług Analysis Services. Ponadto obsługiwany jest dodatek PowerPivot Excela.

Tabelowe źródła danych

Użytkownik może utworzyć połączenie z listami SharePointa, usługami Excela, tabelami z systemu SQL Server i skrośztyami Excela. Te źródła umożliwiają podejście próbkowania danych w narzędziu Dashboard Designer i ustawienie właściwości danych w zależności od tego, jak mają być interpretowane w narzędziu PerformancePoint. Możesz np. określić, które zbiory danych mają być traktowane jak wymiary lub fakty. Możesz też ustawić pomijanie danych, których nie potrzebujesz. Jeśli zdecydujesz się traktować określone wartości jako fakty, możesz określić, w jaki sposób mają być one agregowane w usługach PerformancePoint. Możesz też stosować zbiory danych z wartościami reprezentującymi czas i używać funkcji Time Intelligence do ustawiania parametrów związanych z czasem oraz tworzenia filtrów w panelu administracyjnym.

Listy SharePointa

W usługach PerformancePoint można używać danych z list SharePointa. W tym celu w narzędziu Dashboard Designer należy utworzyć odpowiednie źródło danych. Dane z list SharePointa można tu tylko czytywać (nie są one przeznaczone do modyfikowania).

Zmiany na listach SharePointa są wprowadzane tylko w samym SharePointcie. Użytkownicy mogą łączyć się z dowolnymi listami SharePointa.

Usługi Excela

Dane z plików Excela publikowane za pomocą usług Excela w witrynie SharePointa można wykorzystać w usługach PerformancePoint. Wymaga to utworzenia źródła danych dla usług Excela. W usługach PerformancePoint można wtedy tylko wczytywać opublikowane dane. Używane wartości parametrów są modyfikowane w narzędziu Dashboard Designer. Jeśli używasz parametrów z usług Excela do obliczania wskaźników KPI, możesz łatwo wprowadzić dodatkowe zmiany. Usługi PerformancePoint obsługują komponenty usług Excela, takie jak zakresy nazwane, tabele i parametry.

Tabele z systemu SQL Server

Możesz utworzyć połączenie z bazą z systemu SQL Server i wykorzystać dane w usługach PerformancePoint. W tych usługach obsługiwane są tabele i widoki.

Skoroszyty Excela

W usługach PerformancePoint źródłem danych może być przechowywana w nich zawartość pliku Excela. Wymaga to utworzenia połączenia ze źródłem danych w postaci skoroszytu Excela i zaznaczenia potrzebnych danych. Pierwotny plik Excela jest wtedy niezależny od kopii jego danych z usług PerformancePoint. W usługach PerformancePoint 2010 źródłem danych mogą być skoroszyty z Excela 2007 i 2010.

Usługi Analysis Services

Aby w usługach PerformancePoint wykorzystać dane z wielowymiarowych kostek lub modeli tabelowych z usług SQL Server Analysis Services, należy utworzyć połączenie z tym źródłem. Usługi PerformancePoint umożliwiają powiązanie wymiaru czasu i pożądanego poziomu szczegółów z wewnętrznymi mechanizmami analizy czasowej z tych usług.

Dodatek PowerPivot dla Excela

W usługach PerformancePoint można wykorzystać model z dodatku PowerPivot jako źródło danych używane w panelu administracyjnym. Aby zastosować dodatek PowerPivot w ten sposób, trzeba aktywować usługi PerformancePoint w farmie z platformą SharePoint Server i zainstalować dodatek PowerPivot dla SharePointa. Po utworzeniu modelu za pomocą dodatku PowerPivot dla Excela trzeba przesłać lub opublikować plik serwera w witrynie SharePointa z włączoną obsługą usług PowerPivot. W narzędziu Dashboard Designer utwórz połączenie ze źródłem danych za pomocą szablonu dla usług Analysis Services.

Odświeżanie danych w usługach Visio

Usługa Visio obsługuje połączenia ze źródłami danych, takimi jak listy SharePointa, skoroszyty Excela przechowywane w farmie, bazy danych (np. z systemu SQL Server) i źródła niestandardowe. Aby kontrolować dostęp do źródeł danych, należy bezpośrednio zdefiniować zaufanych dostawców danych i skonfigurować ich na liście takich dostawców.

Gdy usługi Visio wczytują rysunki w formacie VDW, usługa sprawdza zapisane w pliku rysunku informacje o połączeniu, aby ustalić, czy używany dostawca danych znajduje się na liście zaufanych jednostek. Jeśli tak jest, następuje próba nawiązania połączenia za pomocą tego dostawcy. W przeciwnym razie żądanie nawiązania połączenia jest ignorowane.

Po tym jak administrator skonfiguruje usługi Visio i umożliwi łączenie się z określonym źródłem danych, trzeba skonfigurować dodatkowe zabezpieczenia (zależne od rodzaju źródła danych). Usługi Visio współdziałają z następującymi źródłami danych:

- skoroszytami Excela przechowywanymi w SharePointcie z włączonymi usługami Excela,
- listami SharePointa,
- bazami danych (np. z systemu SQL Server),
- niestandardowymi dostawcami danych,
- rysunkami VDW połączonymi z listami SharePointa.

UWAGA Opublikowane rysunki Visio można powiązać z listami SharePointa z tej samej farmy, w której rysunki się znajdują. Użytkownik przeglądający takie rysunki musi mieć dostęp zarówno do nich, jak i do powiązanych list SharePointa. Za zarządzanie niezbędnymi uprawnieniami i danymi uwierzytelniającymi odpowiada SharePoint.

Dane z Visio można odświeżać w SharePointcie na kilka sposobów. Najczęściej stosowane dwie techniki (wykorzystanie usług Excela i systemu SQL Server) opisujemy w następnych podpunktach.

Rysunki VDW powiązane z usługami Excela

Opublikowane rysunki Visio można powiązać ze skoroszytami Excela przechowywanymi w tej samej farmie. Wymaga to uruchomionych i poprawnie skonfigurowanych usług Excela. Aby wyświetlić rysunki, użytkownik musi mieć dostęp zarówno do nich, jak i do powiązanego z nimi skoroszytu Excela. Za zarządzanie niezbędnymi uprawnieniami i danymi uwierzytelniającymi odpowiada SharePoint.

Rysunki VDW powiązane z bazami z systemu SQL Server

Gdy opublikowany rysunek VDW jest powiązany z bazą z systemu SQL Server, usługi Visio przy nawiązywaniu połączenia między usługą Visio Graphics i bazą danych używają dodatkowych opcji konfiguracyjnych dotyczących zabezpieczeń.

Metody uwierzytelniania w usługach Visio

Visio obsługuje następujące metody uwierzytelniania.

- **Zintegrowane uwierzytelnianie z systemu Windows.** W tym modelu zabezpieczeń usługa Visio Graphics wykorzystuje tożsamość osoby oglądającej rysunek do uwierzytelniania w bazie danych. Ten model z ograniczonym delegowaniem opartym na protokole Kerberos zwiększa bezpieczeństwo w większym stopniu niż pozostałe metody uwierzytelniania. Taka konfiguracja wymaga włączenia ograniczonego delegowania z wykorzystaniem protokołu Kerberos na serwerze aplikacji, na którym działa usługa Visio Graphics, i na serwerze bazodanowym. Baza danych może wymagać dodatkowych zmian w konfiguracji, aby obsługiwała uwierzytelnianie oparte na protokole Kerberos.

- **Narzędzie SSS.** W tym modelu usługa Visio Graphics używa narzędzia SSS do powiązania danych uwierzytelniających użytkownika z innymi danymi, które zapewniają dostęp do bazy danych. Narzędzie SSS obsługuje wiązanie danych uwierzytelniających z użytkownikami i grupami na podstawie zintegrowanego uwierzytelniania z systemu Windows i innych mechanizmów uwierzytelniania (np. z systemu SQL Server). Dzięki temu administratorzy mogą definiować powiązania jeden do jednego, wiele do jednego i wiele do wielu. Ten model można stosować tylko dla rysunków, które korzystają z połączeń z plików ODC (ang. *Office Data Connection*). Plik ODC określa w narzędziu SSS docelową aplikację, dla której można dodać powiązania danych uwierzytelniających. Pliki ODC tworzy się w Excelu.
- **Nienadzorowane konto usługowe.** Aby ułatwić konfigurowanie rozwiązań, usługa Visio Graphics udostępnia specjalny mechanizm pozwalający administratorowi powiązać wszystkich użytkowników z jednym kontem dla docelowej aplikacji z systemu SSS. To powiązane konto (*nienadzorowane konto usługowe*) musi być kontem domenowym z systemu Windows mającym niskie uprawnienia i zapewniającym dostęp do baz danych. Usługa Visio Graphics posługuje się tym kontem, gdy łączy się z bazą danych, a nie jest ustawiona inna metoda uwierzytelniania. To podejście nie pozwala na wykonywanie spersonalizowanych kwerend w bazie danych i nie umożliwia inspekcji wywołań kierowanych do bazy. Jest to domyślna metoda uwierzytelniania używana przy łączeniu się z bazami z systemu SQL Server. Jeśli dla rysunku VDW nie jest ustawiony plik ODC z inną metodą uwierzytelniania, usługi Visio łączą się z bazą z systemu SQL Server za pomocą danych uwierzytelniających przypisanych do konta nienadzorowanego.

W farmie SharePointa obejmującej wiele rysunków Visio zwykle stosowane są różne metody. W trakcie konfigurowania zabezpieczeń aplikacji Visio uwzględnij następujące cechy różnych metod uwierzytelniania.

- Usługi Visio umożliwiają używanie plików ODC i nienadzorowanego konta usługowego w tej samej farmie. Dla rysunków VDW, które są powiązane z danymi z systemu SQL Server, ale nie używają plików ODC, niezbędne (i zawsze używane) jest konto nienadzorowane.
- Jeśli stosowane jest uwierzytelnianie z systemu Windows i próba uwierzytelnienia w źródle danych się nie powiedzie, usługi Visio nie będą próbowały wyświetlić rysunku za pomocą nienadzorowanego konta usługowego.
- Zintegrowane uwierzytelnianie z systemu Windows można stosować razem z narzędziem SSS. W tym celu dla rysunków, które wymagają specjalnych danych uwierzytelniających, należy ustawić plik ODC z docelową aplikacją dla narzędzia SSS.

Odświeżanie danych w dodatku PowerPivot

Odświeżanie danych w dodatku PowerPivot to wykonywana na serwerze według harmonogramu operacja, która obejmuje wywoływanie kwerend skierowanych do zewnętrznych źródeł danych w celu zaktualizowania zagnieżdżonych danych dodatku PowerPivot w skoroszybie Excela z biblioteki.

Odświeżanie danych to wbudowany mechanizm dodatku PowerPivot w SharePointcie, jednak stosowanie jej wymaga uruchomienia określonych usług i zadań zegara w farmie SharePointa. Aby odświeżanie danych zakończyło się powodzeniem, często trzeba wykonać dodatkowe zadania administracyjne (np. zainstalować dostawców danych i sprawdzić uprawnienia baz danych).

Po skonfigurowaniu środowiska serwera i uprawnień można odświeżyć dane. W celu wykonywania tej operacji SharePoint tworzy harmonogram dla skoroszytu dodatku PowerPivot. Harmonogram określa, jak często dane mają być odświeżane. Za przygotowanie harmonogramu odpowiada zwykle właściciel lub autor skoroszytu, który opublikował plik w SharePointcie. Ta osoba tworzy harmonogramy odświeżania danych dla należących do niej skoroszytów i zarządza tymi harmonogramami. W następnych podpunktach przedstawiamy kroki, które trzeba wykonać, aby z powodzeniem odświeżyć dane dodatku PowerPivot.

Krok 1. Skonfigurowanie narzędzia SSS i wygenerowanie klucza głównego

Odświeżanie danych dodatku PowerPivot wymaga, aby narzędzie SSS udostępniało dane uwierzytelniające potrzebne w zadaniu odświeżania i nawiązywało połączenie z zewnętrznymi źródłami danych, dla których używane są te informacje.

Jeśli zainstalowałeś dodatek PowerPivot dla SharePointa za pomocą opcji *New Server*, narzędzie SSS jest już skonfigurowane. Wszystkie inne scenariusze instalacji wymagają ręcznego utworzenia i skonfigurowania aplikacji usługowej oraz wygenerowania głównego klucza szyfrowania dla narzędzia SSS. W tym celu należy wykonać następujące kroki.

1. W oknie *Central Administration* w sekcji *Application Management* wybierz opcję *Manage service applications*.
2. Na wstążce *Service Applications* wybierz opcję *Create/New*.
3. Wybierz opcję *Secure Store Service*.
4. Na stronie *Create Secure Store Application* wprowadź nazwę aplikacji.
5. W sekcji *Database* określ egzemplarz systemu SQL Server, w którym działa baza dla danej aplikacji usługowej. Domyślnie używany jest egzemplarz silnika bazodanowego systemu SQL Server, w którym znajdują się bazy z konfiguracją farmy.
6. W sekcji *Database Name* wprowadź nazwę bazy aplikacji usługowej. Domyślnie stosowana nazwa to *Secure_Store_Service_DB_<guid>*. Jest ona powiązana z domyślną nazwą aplikacji usługowej. Jeśli wprowadzisz unikatową nazwę aplikacji usługowej, zastosuj odpowiednią nazwę dla bazy danych, aby wspólnie nimi zarządzać.
7. W sekcji *Database Authentication* wartość domyślna to *Windows Authentication*. Jeśli wybierzesz opcję *SQL Server Authentication*, poproś administratora SharePointa o wyjaśnienie, jak zastosować ten typ uwierzytelniania w farmie.
8. W sekcji *Application Pool* wybierz opcję *Create new application pool*. Podaj tu opisową nazwę, która pomoże innym administratorom serwera ustalić, w jaki sposób pula aplikacji jest używana.
9. Wybierz konto zabezpieczeń dla puli aplikacji. Określ zarządzane konto, które chcesz zastosować. Powinno to być konto użytkownika domeny.
10. W pozostałych miejscach pozostaw wartości domyślne, a następnie kliknij przycisk *OK*. Nowa aplikacja usługowa pojawi się obok innych usług zarządzanych na liście aplikacji z danej farmy.
11. Kliknij na liście aplikację *Secure Store Service*.
12. Wybierz opcję *Manage* ze wstążki *Service Applications*.
13. W sekcji *Key Management* wybierz opcję *Generate New Key*.

14. Wprowadź i zatwierdź hasło. Posłuży ono do dodawania nowych aplikacji usługowych do narzędzia SSS.
15. Kliknij przycisk OK.

UWAGA Aby korzystać z mechanizmu rejestrowania operacji narzędzia SSS na potrzeby inspekcji (jest to przydatne przy rozwiązywaniu problemów), trzeba go włączyć.

Krok 2. Wyłączanie niepotrzebnych metod uwierzytelniania

Mechanizm odświeżania danych dodatku PowerPivot udostępnia trzy opcje uwierzytelniania w harmonogramie. Gdy właściciel skoroszytu tworzy harmonogram odświeżania danych, wybiera jedną z tych opcji i określa tym samym konto używane przez zadanie odświeżania. Administrator może ustalić, które z tych opcji mają być dostępne dla właściciela harmonogramu.

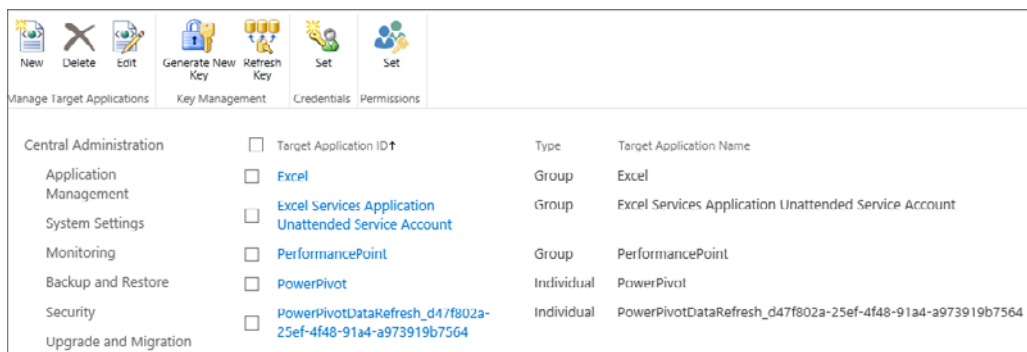
- **Opcja 1.** *Use the data refresh account configured by the administrator.* Opcja zawsze jest dostępna na stronie z definicją harmonogramu, jednak działa tylko wtedy, gdy skonfigurowane jest nienadzorowane konto do odświeżania danych.
- **Opcja 2.** *Connect using the following Windows user credentials.* Jest ona zaznaczona na rysunku 23.5. Opcja zawsze jest dostępna, jednak działa tylko wtedy, gdy na stronie z konfiguracją aplikacji usługowej włączona jest opcja *Allow Users to Enter the Custom Windows Credentials*. Ta ostatnia opcja domyślnie jest włączona, jednak można ją wyłączyć, jeśli jej stosowanie powoduje więcej problemów, niż przynosi korzyści.

Rysunek 23.5. Dostępne opcje uwierzytelniania

- **Opcja 3.** *Connect using the credentials saved in Secure Store Service....* Ta opcja także jest zawsze dostępna na stronie, jednak działa tylko wtedy, gdy właściciel harmonogramu określi odpowiednią aplikację docelową. Administrator musi wcześniej utworzyć aplikację docelową, a następnie podać jej nazwę osobie, która tworzy harmonogram odświeżania danych. Opcja do działania wymaga skonfigurowanej aplikacji usługowej i blokuje inne możliwości (np. opcję 2.).

Aplikacja usługowa dla dodatku PowerPivot obejmuje opcję, która umożliwia właścicielom harmonogramu podanie nazwy dowolnego użytkownika z systemu Windows i jego hasła, aby uruchomić zadanie odświeżania danych. Jest to opcja widoczna na rysunku 23.1 (opisana wcześniej jako opcja 2.).

Opcja uwierzytelniania zaznaczona na rysunku 23.5 jest domyślnie włączona. Przy tym ustawieniu usługa systemowa programu PowerPivot generuje w narzędziu SSS docelową aplikację, aby zapisać nazwę użytkownika i hasło wprowadzone przez właściciela harmonogramu. Nazwa wygenerowanej aplikacji docelowej jest tworzona według schematu `PowerPivotData↪Refresh_<guid>`. Przedstawiono to na rysunku 23.6. Dla każdego zestawu danych uwierzytelniających z systemu Windows tworzona jest jedna docelowa aplikacja.



Rysunek 23.6. Docelowa aplikacja wygenerowana dla odświeżania danych

Główną zaletą stosowania tej opcji uwierzytelniania jest łatwość korzystania z niej. Nie trzeba wykonywać wcześniej wielu zadań, ponieważ docelowe aplikacje są generowane automatycznie. Ponadto odświeżanie danych z wykorzystaniem danych uwierzytelniających właściciela harmonogramu upraszcza późniejszą obsługę uprawnień. Ten użytkownik zwykle ma uprawnienia do docelowej bazy danych, ponieważ prawdopodobnie jest autorem skoroszytu. Gdy odświeżanie danych jest przeprowadzane za pomocą ustalonej przez system Windows tożsamości tego użytkownika, wszystkie połączenia wykorzystujące bieżącego użytkownika automatycznie będą działać.

Wadą są ograniczone możliwości w zakresie zarządzania. Choć docelowe aplikacje są tworzone automatycznie, nie są automatycznie usuwane ani aktualizowane, gdy zmienia się informacje o koncie. Polityka wygasania haseł może sprawić, że aplikacje docelowe staną się nieaktualne. Zadania odświeżania danych, które korzystają z nieaktualnych danych uwierzytelniających, przestaną wtedy działać. Jeśli skonfigurowane są alerty, administrator dostanie komunikat na e-mail. Wtedy właściciel harmonogramu powinien zaktualizować dane uwierzytelniające i podać w harmonogramie nową nazwę użytkownika oraz odpowiednie hasło. Wtedy wygenerowana zostanie nowa aplikacja docelowa. Gdy użytkownicy z czasem będą dodawać i poprawiać dane uwierzytelniające w harmonogramach, w systemie może pojawić się duża liczba automatycznie wygenerowanych aplikacji docelowych.

Obecnie nie istnieje sposób na ustalenie, które z aplikacji docelowych są aktywne lub nieaktywne. Ponadto nie ma możliwości sprawdzenia, z którymi harmonogramami powiązana jest dana aplikacja docelowa. Zwykle nie należy usuwać aplikacji docelowych, ponieważ może to sprawić, że istniejące harmonogramy odświeżania danych przestaną działać. Usunięcie aktywnej aplikacji docelowej powoduje, że odświeżanie danych kończy się niepowodzeniem. Na stronie historii odświeżania danych skoroszytu pojawia się wtedy komunikat *Target Application Not Found*.

Jeśli zdecydujesz się wyłączyć omawianą opcję uwierzytelniania, będziesz mógł bezpiecznie usunąć wszystkie docelowe aplikacje wygenerowane na potrzeby odświeżania danych dodatku PowerPivot.

Krok 3. Tworzenie docelowych aplikacji z danymi uwierzytelniającymi używanymi przy odświeżaniu danych

Gdy skonfigurowane jest narzędzie SSS, administratorzy SharePointa mogą tworzyć aplikacje docelowe, aby udostępniać zapisane dane uwierzytelniające na potrzeby odświeżania danych. Używane może być przy tym nienadzorowane konto do odświeżania danych dodatku PowerPivot lub inne konto służące do uruchamiania zadań lub łączenia się z zewnętrznymi źródłami danych.

Aby stosować niektóre opcje uwierzytelniania, trzeba utworzyć aplikację docelową. Niezbędne są aplikacja docelowa dla nienadzorowanego konta służącego do odświeżania danych dodatku PowerPivot i dodatkowe dane uwierzytelniające, które mogą być potrzebne przy odświeżaniu danych.

Krok 4. Instalowanie dostawców danych używanych do importowania danych dodatku PowerPivot

Odświeżanie danych przebiega tak samo jak importowanie pierwotnych informacji. Dlatego ci sami dostawcy danych, którzy posłużyli do importowania informacji w aplikacji klienckiej dodatku PowerPivot, muszą być zainstalowani na serwerze aplikacji dodatku PowerPivot.

Dostawców danych na serwerze Windows mogą instalować lokalni administratorzy. Jeśli instalujesz dodatkowe sterowniki, umieść je na każdym komputerze z farmy SharePointa, na którym działa dodatek PowerPivot dla SharePointa. Jeśli w farmie działa kilka serwerów aplikacji dodatku PowerPivot, dostawców musisz zainstalować na każdym z nich.

UWAGA Pamiętaj, że serwery SharePoint to aplikacje 64-bitowe. Dlatego koniecznie zainstaluj 64-bitowe wersje dostawców używanych w operacjach odświeżania danych.

Krok 5. Przyznawanie uprawnień do tworzenia harmonogramów i dostępu do zewnętrznych źródeł danych

Właściciele i autorzy skoroszytów muszą mieć uprawnienie `Contribute`, aby mogli utworzyć harmonogram odświeżania danych w skoroszycie. Przy tym poziomie uprawnień użytkownicy mogą otwierać i edytować strony konfiguracji odświeżania danych ze skoroszytu, aby ustawić potrzebne dane uwierzytelniające i informacje o harmonogramie.

Oprócz uprawnień SharePointa czasem trzeba też sprawdzić uprawnienia do zewnętrznych źródeł danych, aby się upewnić, że konta używane przy odświeżaniu mają wystarczające uprawnienia do danych. Przy ustalaniu potrzebnych uprawnień trzeba przeprowadzić staranną analizę. Uprawnienia, które należy przyznać, zależą od łańcucha znaków połączenia ze skoroszytu i od tożsamości użytkownika używanej do uruchamiania zadania odświeżania danych. W ramach analiz należy zastanowić się nad następującymi pytaniami.

- **Dlaczego istniejące łańcuchy znaków połączenia ze skoroszytu dodatku PowerPivot mają znacznie przy odświeżaniu danych?** Gdy dane są odświeżane, serwer wysyła żądanie nawiązania połączenia z zewnętrznym źródłem danych. Używany jest przy tym łańcuch znaków połączenia utworzony w trakcie importowania danych. Parametry określające lokalizację serwera, nazwę bazy i dane uwierzytelniające są ponownie używane przy odświeżaniu danych, aby uzyskać dostęp do tego samego źródła. Łańcucha znaków połączenia i jego budowy nie można zmienić na potrzeby odświeżania danych. Jest on ponownie używany w pierwotnej postaci. Czasem, gdy do łączenia się ze źródłem danych nie jest stosowane uwierzytelnianie z systemu Windows, w łańcuchu znaków połączenia można zmienić nazwę użytkownika i hasło.

W większości skoroszytów domyślną metodą uwierzytelniania jest używanie zaufanych połączeń lub zintegrowanego uwierzytelniania z systemu Windows. Łańcuch znaków połączenia obejmuje wtedy człony `SSPI=IntegratedSecurity` lub `SSPI=Trusted Connection`. Gdy łańcuch znaków połączenia jest wykorzystywany do odświeżania danych, stosowane jest konto użyte do uruchomienia zadania odświeżania. Dlatego

to konto musi mieć uprawnienia do odczytu zewnętrznego źródła danych używanego za pomocą zaufanego połączenia.

- **Czy włączyłeś nienadzorowane konto do odświeżania danych dodatku PowerPivot?** Jeśli tak, powinieneś przyznać temu kontu uprawnienia do odczytu źródeł używanych przy odświeżaniu danych. Te uprawnienia są potrzebne, ponieważ w skoroszytcie z ustawionymi domyślnymi opcjami uwierzytelniania odświeżanie danych odbywa się przy użyciu nienadzorowanego konta. To konto potrzebuje uprawnień do odczytu źródeł danych aktywnie używanych w firmie (chyba że właściciel harmonogramu ustawi dane uwierzytelniające w łańcuchu znaków połączenia).
- **Czy używasz 2. opcji uwierzytelniania, umożliwiającej właścicielowi harmonogramu wprowadzenie nazwy użytkownika z systemu Windows i hasła?** Użytkownicy, którzy tworzą skoroszyty dodatku PowerPivot, zwykle mają niezbędne uprawnienia, ponieważ to oni zaimportowali dane. Jeśli te osoby później konfiguruje odświeżanie danych z wykorzystaniem własnej tożsamości z systemu Windows, ich konto (mające uprawnienia do odpowiedniej bazy) posłuży do pobierania danych w trakcie ich odświeżania. Istniejące uprawnienia są wtedy wystarczające.
- **Czy używasz 3. opcji uwierzytelniania, kiedy to aplikacja docelowa w narzędziu SSS udostępnia tożsamość użytkownika na potrzeby uruchamiania zadań odświeżania danych?** Konto służące do uruchamiania zadań odświeżania danych potrzebuje uprawnień do odczytu z tych samych powodów, co opisane wcześniej konto nienadzorowane.

Krok 6. Umożliwianie aktualizacji skoroszytu na potrzeby odświeżania danych

Skoroszytów utworzonych za pomocą dodatku PowerPivot dla Excela i systemu SQL Server 2008 R2 domyślnie nie można skonfigurować pod kątem odświeżania danych według harmonogramu w dodatku PowerPivot dla SharePointa i systemie SQL Server 2014. Jeśli w środowisku SharePointa używasz starszych i nowszych wersji skoroszytów dodatku PowerPivot, musisz najpierw zaktualizować skoroszyty z wersji SQL Server 2008 R2; dopiero potem będziesz mógł planować automatyczne odświeżanie ich danych na serwerze.

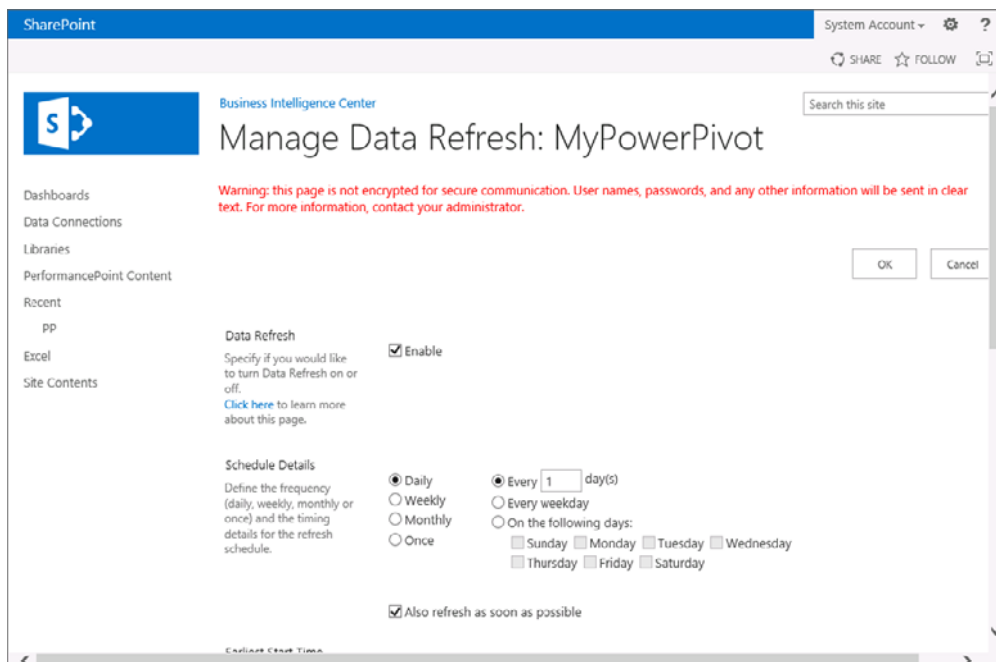
Krok 7. Sprawdzanie konfiguracji mechanizmu odświeżania danych

Aby sprawdzić konfigurację odświeżania danych, musisz opublikować skoroszyt dodatku PowerPivot w witrynie SharePointa. Potrzebne są uprawnienia `Contribute` do skoroszytu i uprawnienia dostępu do źródeł uwzględnianych w harmonogramie odświeżania danych.

Aby zaplanować odświeżanie danych ze skoroszytu według harmonogramu, wykonaj następujące kroki.

1. W SharePoincie otwórz bibliotekę zawierającą dany arkusz dodatku PowerPivot.
2. Zaznacz ten skoroszyt i kliknij skierowaną w dół strzałkę, aby wyświetlić listę poleceń. Zauważ, że konieczne może być przełączenie się z widoku galerii do widoku wszystkich dokumentów.
3. Wybierz opcję *Manage PowerPivot Data Refresh*.
4. W pierwszej sekcji (*Data Refresh*) zaznacz pole wyboru obok opcji *Enable*.
5. W sekcji *Schedule Details* ustaw częstotliwość odświeżania danych, co pokazano na rysunku 23.7. Jeśli skoroszyt używa kilku źródeł danych, w ten sposób ustawisz domyślny

harmonogram. Harmonogramy dla poszczególnych źródeł będzie można zmodyfikować później.



Rysunek 23.7. Konfigurowanie częstotliwości odświeżania danych

6. System może wysyłać e-maile, gdy odświeżanie danych zakończy się niepowodzeniem. W tym celu w polu *Email Notifications* wprowadź dane powiadamianych użytkowników.
7. Na podstawie informacji z podpunktu „Krok 2. Wyłączanie niepotrzebnych metod uwierzytelniania” ustaw opcje uwierzytelniania na potrzeby odświeżania danych.
8. Jeśli chcesz ustawić niestandardowy harmonogram dla źródła danych ze skoroszytu, zrób to w sekcji *Data Sources*, pokazanej na rysunku 23.8.
9. Kliknij przycisk *OK*.

Po utworzeniu harmonogramu zaznacz widoczną na rysunku 23.7 opcję *Also refresh as soon as possible*, aby natychmiast uruchomić odświeżanie danych. Następnie wyświetl stronę z historią odświeżania danych ze skoroszytu, aby sprawdzić, czy ten proces zakończył się powodzeniem. Skonfigurowane zadanie zegara *PowerPivot Data Refresh* jest uruchamiane co minutę. Wygenerowanie potwierdzenia o udanym odświeżeniu danych zajmuje przynajmniej tyle samo czasu.

Koniecznie sprawdź działanie wszystkich opcji uwierzytelniania, które chcesz udostępniać. Jeśli np. skonfigurowałeś nienadzorowane konto do odświeżania danych z dodatku PowerPivot, sprawdź, czy proces kończy się powodzeniem, gdy stosowane jest to konto.

Jeżeli odświeżanie danych się nie powiodło, poszukaj rozwiązań na stronie *Troubleshooting PowerPivot Data Refresh* w serwisie wiki TechNet (<http://technet.microsoft.com>).

SharePoint System Account

E-mail Notifications
Specify e-mail address of the users to be notified in the event of data refresh failures.

LEVIATHAN administrator

Credentials
Provide the credentials that will be used to refresh data on your behalf.

☒ Use the data refresh account configured by the administrator
☐ Connect using the following Windows user credentials
☐ Connect using the credentials saved in Secure Store Service (SSS) to log on to the data source. Enter the ID used to look up the credentials in the SSS ID box

Data Sources
Select which data sources should be automatically refreshed.

View: [Collapse All](#) | [Expand All](#)

Refresh	Data Source
<input checked="" type="checkbox"/>	AdventureWorks

OK Cancel

Rysunek 23.8. Ustawianie niestandardowego harmonogramu

Modyfikowanie konfiguracji odświeżania danych

Każda aplikacja usługowa dodatku PowerPivot zawiera ustawienia, które wpływają na odświeżanie danych. W tym podpunkcie omawiamy dwa podstawowe sposoby modyfikowania tych ustawień.

Zmiana harmonogramu zadania zegara PowerPivot Data Refresh

Planowane odświeżanie danych jest uruchamiane przez zadanie zegara *PowerPivot Data Refresh*, które co minutę sprawdza informacje o harmonogramie w bazie danych aplikacji usługowej dodatku PowerPivot. Gdy planowane odświeżanie danych ma się rozpocząć, zadanie zegara dodaje żądanie do kolejki dostępnego serwera z dodatkiem PowerPivot.

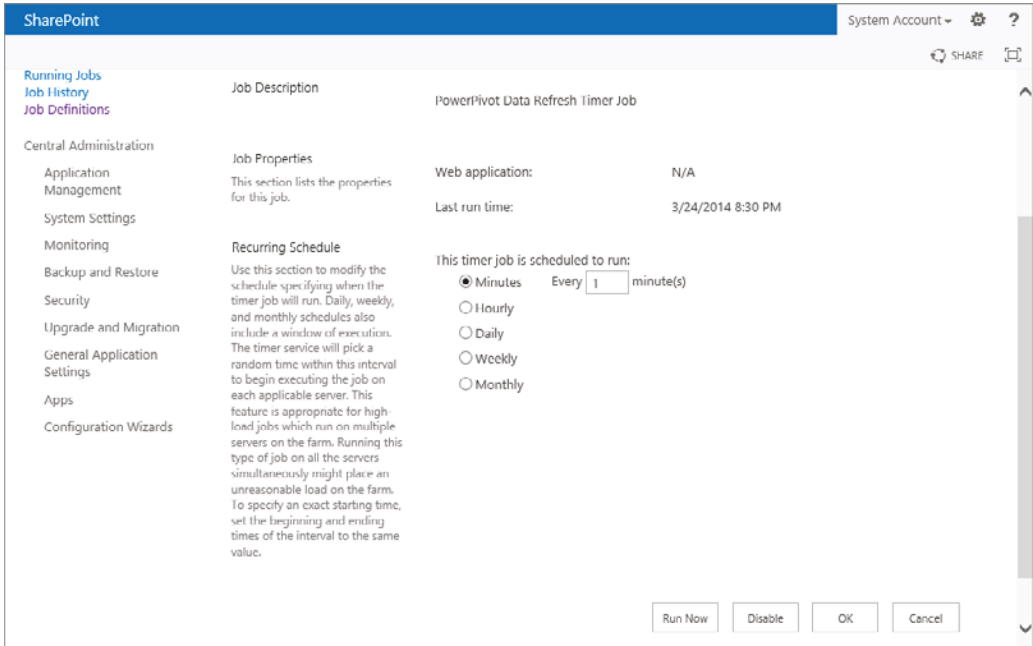
Możesz zwiększyć czas między operacjami sprawdzania harmonogramu lub wyłączyć zadanie zegara, aby tymczasowo zatrzymać odświeżanie danych na czas rozwiązywania problemów.

Ustawienie domyślne to minuta (to najniższa wartość, jaką można ustawić). Jest to zalecany czas, ponieważ zapewnia najbardziej przewidywalne wyniki dla harmonogramów uruchamianych w arbitralnych porach dnia. Jeśli np. użytkownik zaplanował odświeżanie danych na 16.15, a zadanie zegara sprawdza harmonogramy co minutę, żądanie odświeżenia zostanie wykryte o 16.15 i obsłużone w przeciągu kilku minut.

Jeśli zwiększysz odstęp między operacjami sprawdzania harmonogramu w taki sposób, aby były one wykonywane rzadko (np. raz dziennie o północy), wszystkie zaplanowane operacje odświeżania danych trafią do kolejki w tym samym momencie, co może doprowadzić do przeciążenia serwera i odebrania zasobów systemowych innym aplikacjom. Jeżeli liczba operacji odświeżania w harmonogramie jest duża, kolejka może stać się tak długa, że nie wszystkie zadania uda się zakończyć. Żądania odświeżenia danych z końca kolejki mogą wtedy zostać usunięte, jeśli nie uda się ich wykonać w określonym przedziale czasu.

Aby skonfigurować zadanie zegara, wykonaj następujące kroki.

1. Wybierz opcję *Monitoring* w oknie *Central Administration*.
2. Wybierz opcję *Review Job Definitions*.
3. Zaznacz zadanie *PowerPivot Data Refresh Timer Job*.
4. Zmodyfikuj częstotliwość sprawdzania informacji o harmonogramie odświeżania danych przez zadanie zegara. Można określić odstępy między operacjami sprawdzania (w minutach) lub ustawić wykonywanie tej operacji co godzinę, codziennie, co tydzień lub co miesiąc. Domyślne ustawienia przedstawiono na rysunku 23.9.



Rysunek 23.9. Ustawianie częstotliwości sprawdzania harmonogramu

Wyłączanie zadania zegara PowerPivot Data Refresh

Zadanie *PowerPivot Data Refresh* to zadania zegara działające na poziomie farmy. Można je włączyć lub wyłączyć dla wszystkich egzemplarzy dodatku PowerPivot w farmie. Nie jest ono powiązane z daną aplikacją sieciową lub aplikacją usługową dodatku PowerPivot. Nie możesz też go wyłączyć na wybranych serwerach, aby wymusić odświeżenie danych na innych serwerach z farmy.

Jeśli wyłączysz zadanie zegara *PowerPivot Data Refresh*, ządania znajdujące się już w kolejce zostaną wykonane, ale żadne nowe zadania nie będą do niej dodawane do momentu ponownego włączenia zadania. Żądania, których wykonanie zostało zaplanowane na późniejszy termin, nie będą przetwarzane.

Wyłączenie zadania zegara nie wpływa na dostępność funkcji na stronach aplikacji. Nie można usunąć lub ukryć funkcji odświeżania danych w aplikacjach sieciowych. Użytkownicy z uprawnieniami *Contribute* i wyższymi mogą tworzyć nowe harmonogramy odświeżania danych, nawet jeśli zadanie zegara jest trwale wyłączone.

Aby wyłączyć zadanie zegara, wykonaj następujące kroki.

1. Wybierz opcję *Monitoring* w oknie *Central Administration*.
2. Wybierz opcję *Review Job Definitions*.
3. Zaznacz zadanie *PowerPivot Data Refresh Timer Job*.
4. Przewiń ekran w dół i kliknij przycisk *Disable* widoczny na rysunku 23.9.

Podsumowanie

SharePoint jest integralną częścią mechanizmów BI w systemie SQL Server 2014. Zainstalowanie i skonfigurowanie dodatku PowerPivot oraz używanie go razem z usługami Reporting Services działającymi jako usługa współużytkowana pozwala zwiększyć możliwości SharePointa. Nowe funkcje, takie jak Power View, bez wątpienia zwiększają atrakcyjność SharePointa dla osób, które zaczynają korzystać z nowego formatu usług analitycznych, BISM (ang. *BI Semantic Model*), i używają dodatku PowerPivot. SharePoint w połączeniu z systemem SQL Server 2014 jest jeszcze lepszy, łatwiejszy w stosowaniu i bardziej „inteligentny”.

Po zapoznaniu się z integracją mechanizmów BI z SharePointem możesz przejść do rozdziału 24. Znajdziesz w nim wprowadzenie do baz SQL Database z platformy Azure i związanych z nimi zadań administracyjnych.

Administrowanie bazami SQL Database i ich konfigurowanie

ZAWARTOŚĆ ROZDZIAŁU:

- Konfigurowanie baz Windows Azure SQL Database.
- Zarządzanie serwerem i bazami danych.
- Zadania administracyjne.

W tym rozdziale zapoznasz się z administrowaniem bazami Windows Azure SQL Database i ich konfigurowaniem. Zakładamy, że masz już konto w platformie Windows Azure i znasz tę platformę. SQL Database to usługa typu PaaS (ang. *Platform as a Service*, czyli platforma jako usługa), udostępniana w ramach platformy Windows Azure. SQL Database to transakcyjna baza danych oparta na systemie SQL Server. Oferuje ona wiele funkcji systemu SQL Server znanych i cenionych przez użytkowników. Jednak inaczej niż w systemie SQL Server nie trzeba zajmować się instalowaniem i konserwacją sprzętu oraz oprogramowania używanego przez bazy SQL Database. Baza jest udostępniana jako usługa i działa w centrum danych Microsoftu. Firma Microsoft nie tylko dba o fizyczną konserwację sprzętu i oprogramowania, ale też oferuje wbudowane funkcje zapewniania wysokiej dostępności, co pozwala skoncentrować się na ważnych zadaniach, takich jak projektowanie i tworzenie baz danych.

Wprowadzenie do technologii Windows Azure SQL Database

SQL Database to transakcyjna i relacyjna baza danych Microsoftu przeznaczona do pracy w chmurze. Jest ona oparta na systemie SQL Server 2012. Obsługuje wiele funkcji systemu SQL Server, w tym tabele, klucze główne, procedury składowane, widoki itd.

SQL Database (podobnie jak system SQL Server) udostępnia interfejs w języku T-SQL służący do pobierania danych z tabel, dlatego aplikacje bazodanowe mogą korzystać z baz SQL Database w taki sam sposób, w jaki stosują system SQL Server. Jedyna różnica polega na tym, że SQL Database to baza udostępniana jako usługa, co oznacza, że administrowanie nią odbywa się inaczej.

W bazie SQL Database obowiązuje podział na logiczne i fizyczne aspekty administrowania. Przykładowo nadal możesz zarządzać bazami danych, loginami i użytkownikami, jednak to Microsoft obsługuje administrowanie elementami fizycznymi, takimi jak napędy i pamięć lub fizyczne serwery.

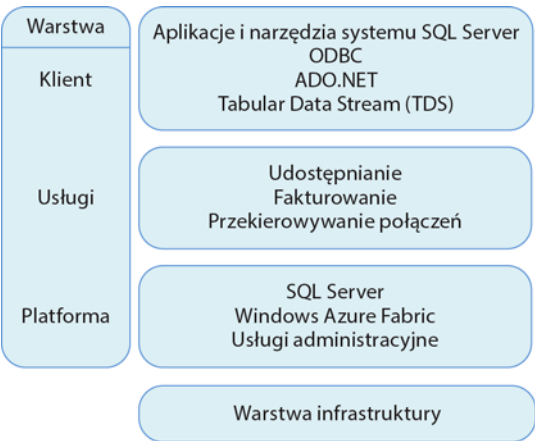
Mimo podziału administrowania na aspekty logiczne i fizyczne, SQL jest tu obsługiwany w standardowy sposób. SQL Database, jak sama nazwa wskazuje, jest bazą SQL-ową. Jest to relacyjna i transakcyjna baza, obejmująca większość obiektów i funkcji z systemu SQL Server. SQL Database oferuje więcej ciekawych mechanizmów niż jej lokalnie instalowany kuzyn, system SQL Server. SQL Database za darmo udostępnia wbudowane funkcje zapewniania wysokiej dostępności i odzyskiwania systemu po katastrofie, których obsługa w lokalnym środowisku jest kosztowna.

Architektura bazy SQL Database

Architektura bazy SQL Database obejmuje wymienione poniżej cztery odrębne warstwy abstrakcyjne, które wspólnie tworzą relacyjną bazę danych przeznaczoną do pracy w chmurze. Oto te warstwy:

- klient,
- usługi,
- platforma,
- infrastruktura.

Na rysunku 24.1 pokazano, że cztery warstwy architektury umożliwiają bazie SQL Database współpracę z aplikacjami o otwartym dostępie do kodu źródłowego niezależnych producentów, a także z wieloma znanymi technologiami Microsoftu.



Rysunek 24.1. Architektura bazy SQL Database

Ta warstwowa architektura jest podobna do architektury lokalnie instalowanego systemu SQL Server. Wyjątkiem jest warstwa usług. To specjalna warstwa bazy SQL Database, która udostępnia wiele specyficznych dla niej funkcji platformy bazodanowej, które opisano w dalszych punktach.

Warstwa klientów

Warstwa klientów działa najbliżej rozwijanych aplikacji. Aplikacja korzysta z tej warstwy do komunikowania się z bazą SQL Database. Warstwa klientów może działać lokalnie lub w platformie Windows Azure. Baza SQL Database korzysta z tego samego interfejsu TDS, co system SQL Server. Dzięki temu programiści mogą korzystać ze znanych im narzędzi i bibliotek do rozwijania pracujących w chmurze aplikacji klienckich. Warstwa zapewnia dostęp do danych za pomocą dostawcy ADO.NET i innych dostawców. Pozwala to manipulować danymi za pomocą standardowych instrukcji w języku T-SQL i znanych technologii.

Warstwa usług

Warstwa usług łączy warstwę klientów i warstwę platformy. Odpowiada za następujące aspekty.

- **Udostępnianie.** Tworzenie i udostępnianie baz danych za pomocą portalu platformy Azure lub narzędzia SQL Server Management Studio.
- **Fakturowanie i pomiary.** Obsługa pomiarów i fakturowania na podstawie używanych usług dla poszczególnych kont z platformy Azure.
- **Przekierowywanie połączeń.** Obsługa wszystkich połączeń między aplikacjami a fizycznymi serwerami, na których znajdują się dane.

Warstwa platformy

Warstwa obejmuje fizyczne serwery i usługi używane w warstwie usług. To w warstwie platformy działa wiele egzemplarzy systemu SQL Server.

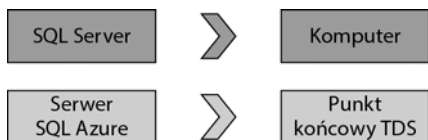
Najważniejszym elementem tej warstwy jest *system SQL Database Fabric* — rozproszony system obliczeniowy instalowany w każdym systemie SQL Server, składający się ze ściśle zintegrowanych sieci, serwerów i magazynów danych. SQL Database Fabric zapewnia automatyczne przełączanie awaryjne, równoważenie obciążenia i automatyczną replikację baz między serwerami.

Warstwa infrastruktury

Warstwa infrastruktury obejmuje fizyczne zarządzanie przez informatyków sprzętem i systemami operacyjnymi obsługującymi warstwę usług.

Różnice między środowiskami

Aby zrozumieć architekturę bazy SQL Database, trzeba pamiętać o zadaniach warstwy usług. Połączenia nie prowadzą tu bezpośrednio do fizycznych egzemplarzy systemu SQL Server. Gdy łączysz się ze standardowym systemem tego typu, nawiązywane jest połączenie z fizycznym serwerem. W bazie SQL Database połączenia przechodzą przez punkt końcowy TDS i warstwę usług, która kieruje je do ukrytego za nią fizycznego serwera. Na rysunku 24.2 przedstawiono różnice między połączeniami z lokalnie zainstalowanym systemem a połączeniami z bazą SQL Database.



Rysunek 24.2. Połączenia ze zwykłym systemem SQL Server i bazą SQL Database

Te różnice są widoczne np. w nazwach serwerów w lokalnie instalowanym systemie SQL Server i w bazach SQL Database. Gdy łączysz się z lokalnym systemem SQL Server, używaną nazwą jest zwykle nazwa fizycznego serwera. W bazach SQL Database używane są nazwy FQDN (ang. *Fully Qualified DNS Name*) w formacie `serwer.database.windows.net`.

Człon określający serwer w nazwie FQDN to dziesięć losowych znaków. Przy łączeniu się z bazą SQL Database trzeba podać całą nazwę.

Konfigurowanie baz SQL Database

Po utworzeniu konta w platformie Azure możesz zacząć pracę z bazą SQL Database. Najpierw musisz utworzyć serwer SQL Azure i bazę danych oraz poznać różne sposoby korzystania z baz SQL Database. Jednym z tych sposobów jest zastosowanie narzędzia Azure Management Portal. To sieciowe narzędzie administracyjne umożliwia zarządzanie wszystkimi aspektami platformy Azure. Oto niektóre elementy tej platformy.

- **Usługi obliczeniowe.** Maszyny wirtualne, usługi w chmurze, witryny internetowe, usługi mobilne.
- **Usługi obsługi danych.** Przechowywanie danych, baza SQL Database, kopie zapasowe, pamięć podręczna, HDInsight, Hyper-V Recovery Manager.
- **Usługi aplikacji.** Media Services, Active Directory, Multi-Factor Authentication, Service Bus, Notification Hubs, BizTalk Services, Scheduler, Visual Studio Online.
- **Usługi sieciowe.** Virtual Network, Traffic Manager.

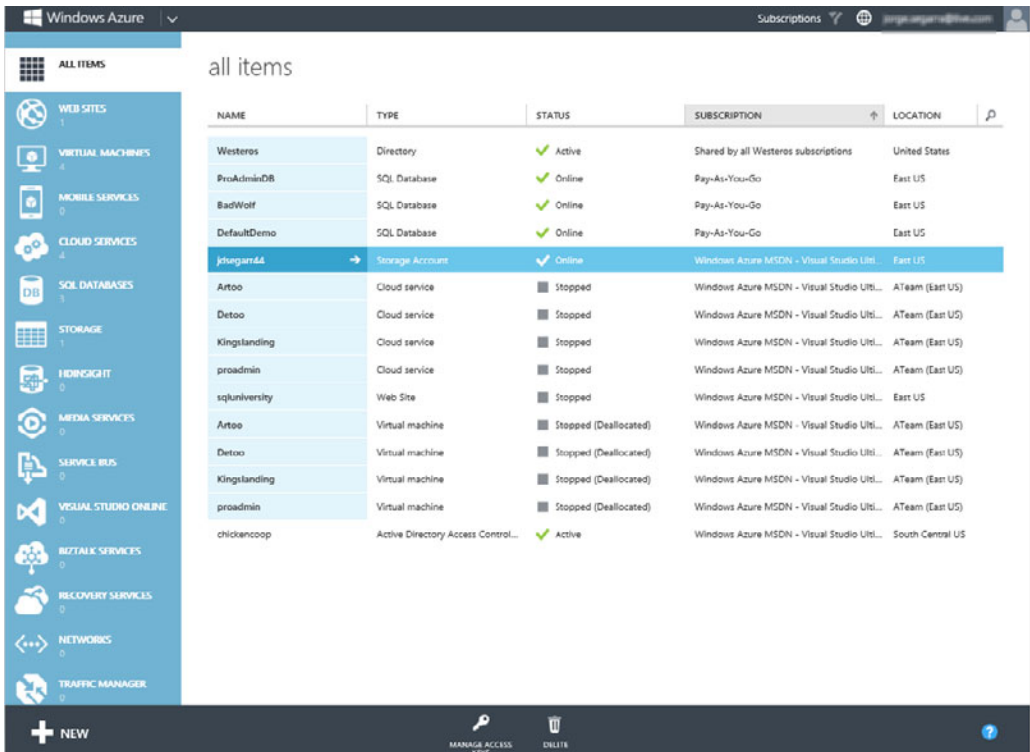
W następnych punktach przedstawiamy proces tworzenia i konfigurowania bazy SQL Database.

Udostępnianie serwera i bazy danych

Najprostszym sposobem na udostępnienie nowego serwera jest użycie narzędzia Azure Management Portal, pokazanego na rysunku 24.3. Aby je otworzyć, wpisz adres <http://manage.windowsazure.com>.

Za pomocą narzędzia Azure Management Portal można zarządzać wszystkimi aspektami subskrypcji bazy SQL Database. Na rysunku 24.3 widać, że dla każdego konta można ustawić wiele subskrypcji. Każda subskrypcja może obejmować wiele serwerów, a na każdym z nich można zapisać wiele baz danych.

Aby utworzyć nową bazę SQL Database, wystarczy kliknąć przycisk *New* na pasku narzędzi w dolnej części głównej strony. W menu *New* wybierz opcję *Data Services/SQL Database/Quick Create*, aby szybko utworzyć bazę danych, serwer bazodanowy i login. Pozostałe opcje konfiguracyjne (wielkość bazy danych, wersję itd.) możesz ustawić później. Inna opcja, *Custom Create*, powoduje uruchomienia kreatora tworzenia baz SQL Database.

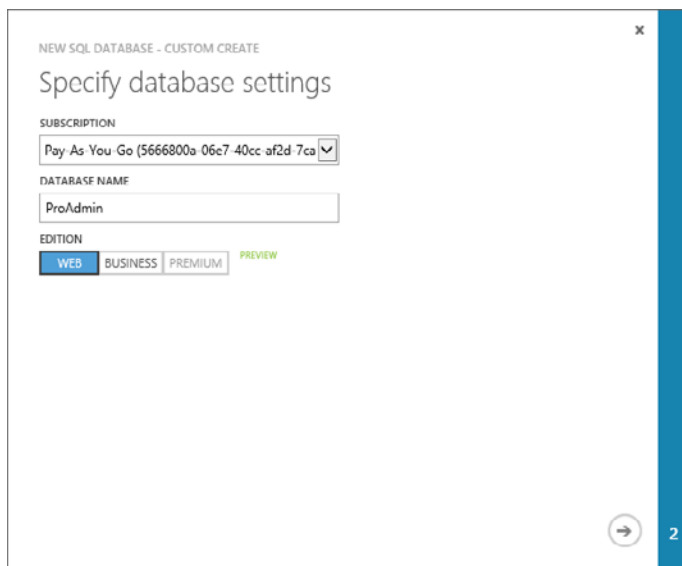


Rysunek 24.3. Narzędzie Azure Management Portal

Przyjrzymy się teraz temu kreatorowi. Kliknij przycisk *New* na dolnym pasku narzędzi, a następnie wybierz opcję *Data Services/SQL Database/Custom Create*.

1. Pierwszy krok w kreatorze polega na podaniu podstawowych informacji, które można wykorzystać do utworzenia bazy danych i serwera (patrz rysunek 24.4). Na tej stronie należy podać nazwę bazy, wersję (Web, Business lub Premium), wielkość i kolację, a także serwer SQL Database, na którym baza ma działać. Oto krótki przegląd różnic między wersjami.
 - **Web.** Obsługuje bazy o wielkości 1 lub 5 gigabajtów. Korzysta ze współużytkowanych zasobów i ma wbudowany mechanizm tworzenia replik w centrum danych.
 - **Business.** Obsługuje bazy o wielkości 10, 20, 30, 40, 50, 100 i 150 gigabajtów. Także korzysta ze współużytkowanych zasobów i ma wbudowany mechanizm tworzenia replik w centrum danych.
 - **Premium.** Wersja jest oparta na edycjach Web i Business, jednak otrzymuje stałe zasoby, które nie są współużytkowane z innymi bazami. Można zaktualizować używaną bazę do wersji Premium, aby zapewnić sobie bardziej przewidywalną wydajność niż w edycjach Web i Business. Proces aktualizowania bazy zajmuje od kilku minut do kilku godzin (zależy to od wielkości bazy).

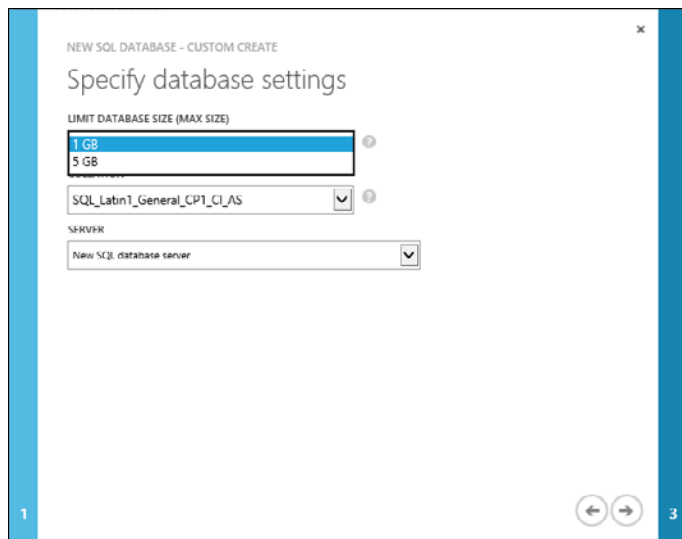
Utwórz bazę o nazwie ProAdminDB. Po wybraniu ustawień kliknij strzałkę w lewym górnym rogu ekranu, aby kontynuować.



Rysunek 24.4. Tworzenie bazy danych w kreatorze

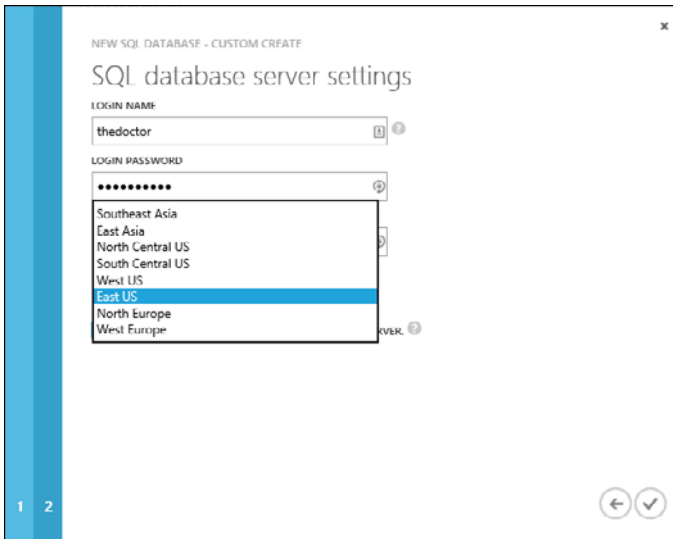
2. Następnie określ wielkość bazy danych. Dostępne rozmiary zależą od wybranej w poprzednim kroku wersji bazy. Możesz tu także zmienić kolację (domyślnie używana to `SQL_Latin1_General_CP1_CI_AS`) na dowolną z obsługiwanych. Ich lista pojawia się w menu rozwijanym. Wybierz też serwer, na którym baza ma działać. Jeśli już utworzyłeś serwer, będzie on dostępny do wyboru.

Ponadto możesz utworzyć nowy serwer baz SQL Database, co pokazano na rysunku 24.5. W trakcie dodawania nowego serwera można określić, w jakim centrum danych ma działać. Nazwa serwera jest generowana automatycznie. Wybierz opcję *New SQL Database server* i kliknij skierowaną w prawo strzałkę, aby kontynuować.



Rysunek 24.5. Tu możesz ustawić wielkość bazy, kolację i serwer

3. Na stronie *SQL Database server settings*, pokazanej na rysunku 24.6, ustaw serwer bazy SQL Database. Jest to logiczny kontener mieszczący takie bazy. Podaj też login i hasło dla bazy. Powstanie w ten sposób główne konto tworzonej bazy SQL Database. Dodany login to odpowiednik konta administracyjnego sa z lokalnie instalowanych systemów SQL Server.



Rysunek 24.6. Konfigurowanie serwera dla bazy SQL Database

W ostatnim kroku wybierz region, w którym dany serwer baz SQL Database ma działać. Najlepiej umieścić serwer i bazę w obszarze najbliższym użytkownikom i aplikacjom z nich korzystającym. Wybierz region z listy rozwijanej widocznej na rysunku 24.6.

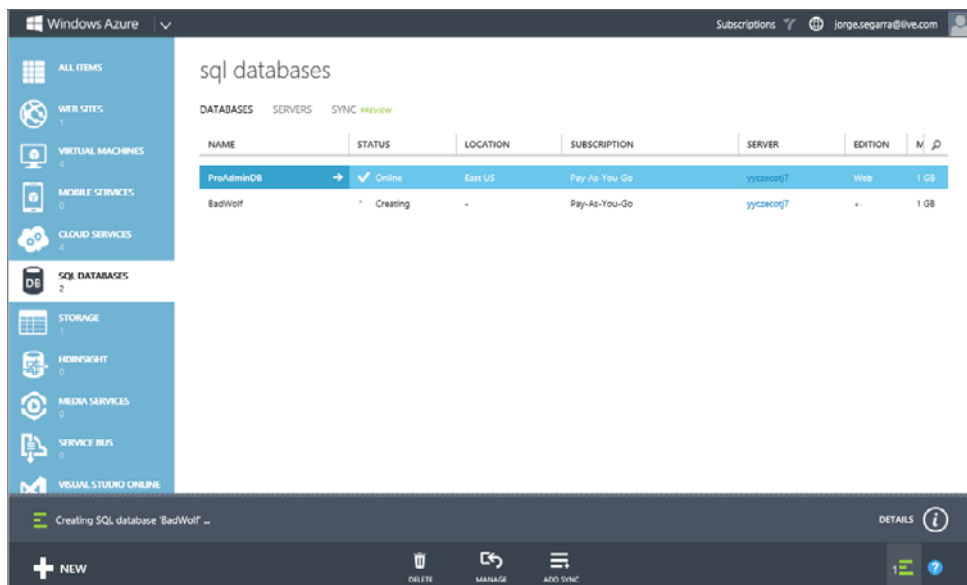
W niektórych sytuacjach firma może chcieć umieścić serwer i bazę w innym regionie niż ten, w którym znajdują się użytkownicy i aplikacje. Załóżmy, że firma z Australii wybrała centrum danych w Azji, ponieważ podejrzewała, że najbliższa taka jednostka zapewni im najwyższą wydajność. Wkrótce jednak okazało się, że wyższą wydajność zapewnia centrum danych *South Central US*, ponieważ łączy między Australią i Stanami Zjednoczonymi zapewniają dużo większą przepustowość niż łączy między Australią i Azją. Choć należy uwzględnić takie kwestie, zazwyczaj lepiej początkowo ustawić najbliższe centrum danych i przeprowadzić dokładne testy.

Utworzyłeś już serwer i bazę SQL Database. Jednak zanim zaczniesz uczyć się z nich korzystać, zapoznaj się z istotnymi informacjami.

Nazwa serwera i bazy danych

Po utworzeniu serwera w środkowej części sekcji *sql databases* portalu w zakładce *DATABASES* wyświetlana jest lista wszystkich baz SQL Database z danej subskrypcji platformy Azure. Lista ta obejmuje właśnie utworzoną bazę i nazwę serwera, na którym ona działa. Przedstawiono to na rysunku 24.7.

Jeśli masz jakieś wątpliwości dotyczące subskrypcji, kliknij jej nazwę na pasku narzędzi w prawej górnej części ekranu. Jeżeli masz kilka subskrypcji, w środkowej części okna wyświetlana jest lista wszystkich baz SQL Database powiązanych z wybraną subskrypcją. Lista *DATABASES* wyświetlana po kliknięciu zakładki *DATABASES* obejmuje następujące kolumny.



Rysunek 24.7. Lista utworzonych baz SQL Database

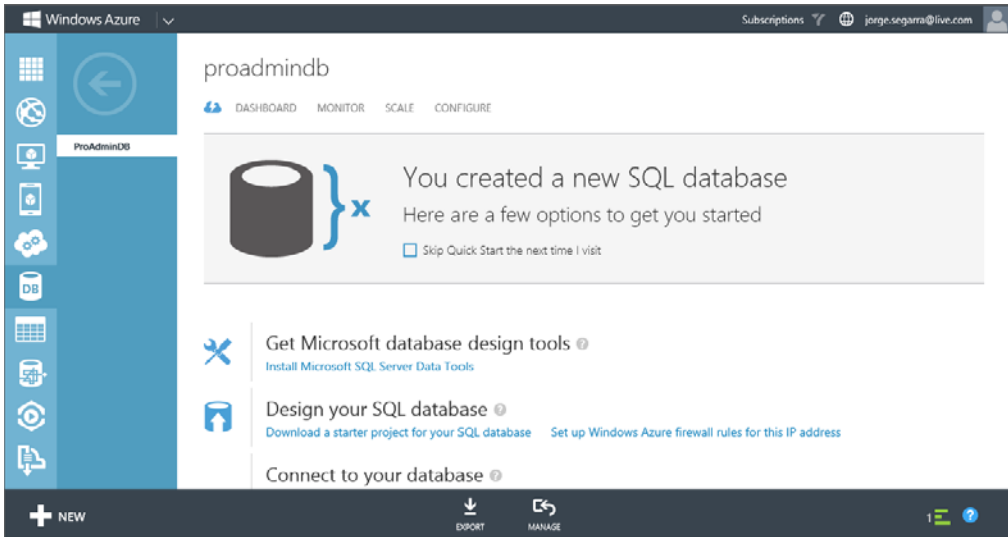
- **Name.** Nazwa bazy danych.
- **Status.** Status bazy (czy jest dostępna).
- **Location.** Lokalizacja centrum danych.
- **Subscription.** Nazwa subskrypcji powiązanej z bazą.
- **Server.** Nazwa serwera baz SQL Database.
- **Edition.** Wersja bazy SQL Database (ta kolumna określa też, czy następuje zmiana prostszej edycji na Premium lub na odwrót).
- **Max Size.** Maksymalna wielkość bazy.

Przyjrzyjmy się nowej bazie SQL Database. Kliknij strzałkę obok nazwy bazy danych, aby wyświetlić szczegółowe informacje na jej temat. Gdy klikniesz odnośnik po raz pierwszy, zobaczysz menu *Quick Start*, widoczne na rysunku 24.8. Zapewnia ono wygodny dostęp do odnośników pozwalających pobrać narzędzia SQL Server Data Tools i projekt startowy bazy SQL Database, skonfigurować zasady zapory dla używanego adresu IP lub połączyć bazę SQL Database z narzędziami programistycznymi i językami programowania. Znajdziesz tu także nazwę FQDN serwera baz SQL Database.

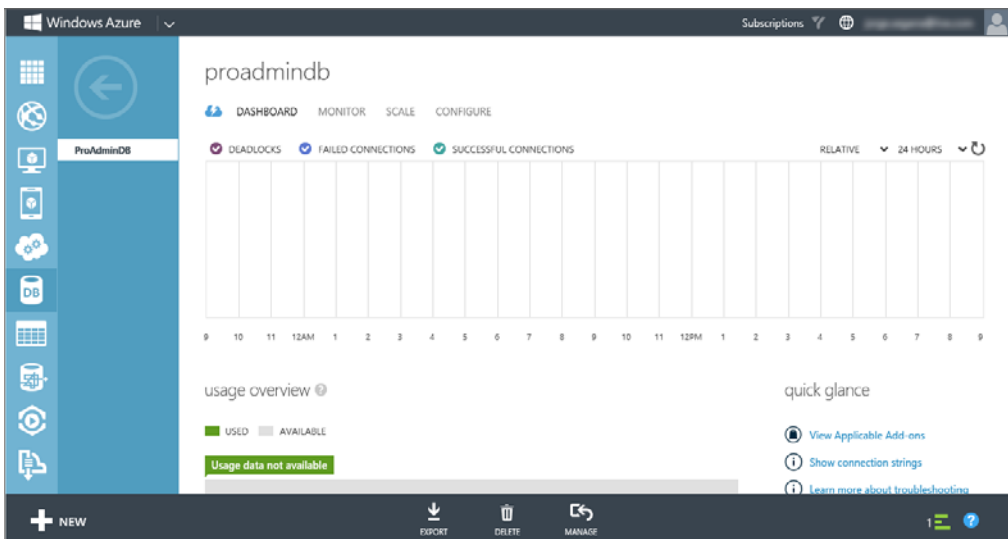
Zauważ, że pierwszy człon nazwy FQDN to nazwa serwera. W tym kontekście nazwa serwera baz SQL Database to wartość wyświetlana w kolumnie serwera na stronie z bazami SQL Database, przedstawionej na rysunku 24.7. Gdy nazwa serwera jest podawana w łańcuchu znaków połączenia, należy użyć nazwy FQDN:

`nazwaserwera.bazadanych.windows.net`

Nazwa FQDN będzie potrzebna przy nawiązywaniu połączenia z bazą SQL Database. Dlatego zaznacz pełną nazwę serwera baz SQL Database i skopiuj ją do schowka. Nazwę FQDN znajdziesz też w sekcji *Quick Glance* w widoku *DASHBOARD*, co przedstawiono na rysunku 24.9. Aby wyświetlić ten widok dla wybranej bazy, kliknij jej nazwę na stronie *sql databases* (patrz rysunek 24.7), a następnie kliknij odnośnik *Dashboard* na stronie *Quick Start*.



Rysunek 24.8. Strona nowej bazy SQL Database



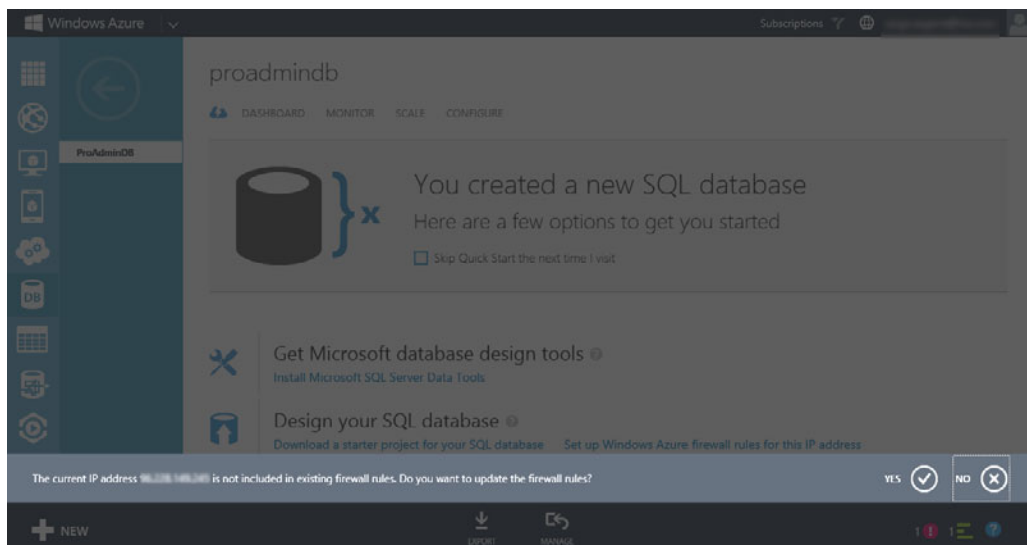
Rysunek 24.9. Widok DASHBOARD

By wyświetlić widok szczegółów serwera baz SQL Database, kliknij zakładkę *Servers* (patrz rysunek 24.7), a następnie kliknij strzałkę obok nazwy serwera. Pojawi się ekran podobny do strony *Quick Start* baz danych. Znajdziesz tu odnośniki prowadzące do często wykonywanych zadań, np. tworzenia nowej bazy lub zarządzania samym serwerem.

Choć użytkownik nie łączy się z fizycznym komputerem, serwer baz SQL Database działa podobnie jak instalowany lokalnie system SQL Server. Oznacza to, że taki serwer obejmuje logiczną grupę baz danych i działa jak centralna jednostka administracyjna dla wielu baz danych. Dlatego możesz utworzyć liczne obiekty znane z instalowanych lokalnie baz danych (np. tabele, widoki, procedury składowane i indeksy). Za pomocą myszy zaznacz pełną nazwę serwera i skopiuj ją do schowka (skrót *Ctrl+C*), ponieważ wkrótce będzie potrzebna.

Gdy stworzysz serwer baz SQL Database, automatycznie generowana jest baza master. Zawiera ona informacje o konfiguracji i zabezpieczeniach baz danych. Aby uzyskać dostęp do tej bazy i nią zarządzać, kliknij ikonę *Manage* na dolnym pasku nawigacyjnym *DETAILS* na stronie *sql databases* (patrz rysunek 24.7).

UWAGA Na platformie Windows Azure można dość dobrze skonfigurować zaporę. Jeśli w trakcie używania portalu lub konfigurowania ustawień wymagane jest przejście poza zaporę, portal automatycznie wyświetla w dolnej części ekranu pokazane na rysunku 24.10 pytanie o to, czy chcesz zapewnić dostęp obecnemu adresowi IP. Pozwala to szybko skonfigurować ustawienia zabezpieczeń na własny użytek, co umożliwia kontynuowanie wprowadzania innych ustawień.



Rysunek 24.10. Dodawanie używanego adresu IP do zasad zapory

Na stronie *Quick Start* bazy SQL Database (patrz rysunek 24.8) można zarządzać zasadami zapory. W tym celu kliknij odnośnik *Set up Windows Azure firewalls rules for this IP address* w sekcji *Design your SQL database*. Za pomocą tego odnośnika możesz szybko przyznać uprawnienia do przekraczania zapory używanemu adresowi IP i sieci. Gdy klikniesz ten odnośnik, pojawi się pytanie o to, czy chcesz dodać dany adres IP do zasad zapory i przejść do konfigurowania zaawansowanych opcji.

Bardziej szczegółowe zasady zapory możesz ustawić w zakładce *DASHBOARD* w sekcji *Quick Glance*. Kliknij odnośnik *Manage allowed IP addresses* na ekranie przedstawionym na rysunku 24.9 (sam odnośnik jest tu niewidoczny). Zaletą stosowania tej opcji jest możliwość tworzenia, modyfikowania i usuwania całych zakresów adresów IP zamiast pojedynczych adresów. Ta opcja doskonale nadaje się do konfigurowania bazy w organizacji, w której komputery o różnych adresach IP mają mieć dostęp do danej bazy SQL Database. Zasady zapory to pierwszy poziom zabezpieczeń baz SQL Database. Są one niezbędne do ochrony danych. Więcej o konfigurowaniu zasad zapory platformy Azure dowiesz się dalej w tym rozdziale.

Opcje z wersji Premium

W czasie, gdy powstawała ta książka, dostępne były trzy wersje baz danych. W zależności od edycji były bazy różnej wielkości. Dwie z dostępnych wersji to *Web* i *Business*; nie różnią się one niczym oprócz wielkości baz. Ich funkcje są takie same.

Trzecia wersja, *Premium*, to rozwinięcie pozostałych edycji. Istniejące bazy *Web* i *Business* można rozbudować do wersji *Premium*. Aktualizacja bazy do wersji *Premium* pozwala uzyskać wyższą i bardziej przewidywalną wydajność. Bazy *Premium* doskonale sprawdzają się dla aplikacji, które stale potrzebują zasobów i wymagają wysokiej współbieżności oraz przewidywalnego opóźnienia. W zależności od potrzeb możesz zmieniać wersję baz SQL Database z niższych na *Premium* i na odwrót. Wielkość bazy decyduje, czy zmiana wersji zajmie kilka minut, czy kilka godzin.

Ograniczanie dostępu do zasobów i równoważenie obciążenia

Ograniczanie dostępu do zasobów (ang. *throttling*) to mechanizm baz SQL Database zapewniający, że aplikacje lub kod (procedury składowane lub kod w języku T-SQL) jednego subskrybenta nie zajmą wszystkich zasobów. Ponieważ baza SQL Database działa na zapleczu i ma zapewniać wysoką wydajność, używany jest mechanizm *równoważenia obciążenia*. Pozwala to zagwarantować, że serwer nie pracuje cały czas w trybie ograniczania dostępu do zasobów. Aby w pełni zrozumieć mechanizmy ograniczania dostępu do zasobów i równoważenia obciążenia, warto przyjrzeć się temu, jak (i gdzie) tworzone są nowe bazy SQL Database.

Technologia SQL Database ma zapewniać dostępność baz danych subskrybentów przez 99,9 czasu (jest to obecnie stosowany poziom). Tak wysoką dostępność uzyskuje się kilkoma metodami.

Po pierwsze, Microsoft wykorzystuje standardowy sprzęt, który po awarii można szybko i łatwo zastąpić. Po drugie, co ważniejsze, Microsoft stosuje automatyczne zarządzanie replikami baz danych. Gdy stworzysz bazę, powstają trzy jej egzemplarze: wersja główna i dwie pomocnicze repliki. Wszystkie znajdują się w tym samym centrum danych. Bazy zawsze są automatycznie synchronizowane bez konieczności podejmowania działań przez użytkownika.

Użytkownik, jego aplikacje i inne jednostki nie mają bezpośredniego dostępu do pomocniczych baz danych, jednak te bazy są dostępne i potrzebne. Jeśli z jakiegoś powodu główna baza stanie się niedostępna, serwer baz SQL Database wyłączy ją, wybierze jedną z pomocniczych baz danych, ustawi ją jako główną, a następnie utworzy dodatkową bazę pomocniczą i ją zaktualizuje. Wszystko to odbywa się automatycznie na zapleczu. Trzeba jednak zająć się zerwanymi połączeniami z bazą danych. W tym kontekście najlepiej dodać do aplikacji mechanizm ponawiania prób nawiązania połączenia i wykonania instrukcji. Dzięki temu jeśli główna baza danych przestanie działać, aplikacja będzie mogła ponownie spróbować nawiązać połączenie. Zanim do tego dojdzie, nowy serwer główny będzie już gotowy do przyjęcia nadsyłanych żądań.

UWAGA W praktyce zawsze należy dodać do aplikacji mechanizm ponawiania prób nawiązania połączenia i wykonania instrukcji. Nie chcesz przecież, aby użytkownicy otrzymywali komunikaty z informacją, że nie można ukończyć wykonywanej operacji. Każda aplikacja powinna automatycznie ponawiać próby w reakcji na zwrócenie określonych błędów z systemu SQL Server. To samo dotyczy platformy Windows Azure i baz SQL Database.

Platforma Azure jest *środowiskiem współużytkowanym*, co oznacza, że współużytkujesz zasoby serwera razem z innymi subskrybentami. Ponadto na każdym serwerze działają też inne procesy niezbędne do ciągłej i płynnej pracy baz SQL Database. Aby uniknąć odebrania przez bazę niezbędnych zasobów innym bazom lub samemu serwerowi baz SQL Database, Microsoft utworzył komponent Engine Throttling. Jego zadaniem jest dbałość o stan komputera. Ten komponent gwarantuje, że wszystkie procesy mają potrzebne im do płynnej i wydajnej pracy zasoby, a także żadna jednostka nie zużywa więcej zasobów, niż jest to konieczne, i limit wykorzystania zasobów nie został przekroczony.

Jeśli limit (np. obciążenia procesora lub wielkości dziennika) został przekroczony, komponent Engine Throttling podejmuje niezbędne działania naprawcze. Polegają one na zrywaniu połączeń, blokowaniu operacji odczytu lub zapisu przez określony czas (np. 10 sekund), a nawet trwałym blokowaniu tych zadań, jeśli źródło problemu wciąż powoduje kłopoty. Inne czynniki, które mogą spowodować ograniczanie dostępu do zasobów, to zadania wymagające wielu operacji wejścia-wyjścia (np. konserwacja indeksu).

Gdy dodawane są nowe bazy, mechanizm równoważenia obciążenia określa lokalizację nowej bazy głównej i baz pomocniczych na podstawie obecnego obciążenia komputerów w centrum danych.

Lokalizacja baz przez pewien czas może być odpowiednia, jednak nie da się przewidzieć obciążenia baz innych subskrybentów. Dlatego mechanizm równoważenia obciążenia ma za zadanie zapewniać wystarczającą wydajność baz danych. Jeśli dana maszyna jest przeciążona, mechanizm równoważenia obciążenia może automatycznie przenieść bazę na mniej obciążony serwer. Operacja odbywa się w sposób nieodczuwalny dla aplikacji.

Aby zminimalizować ograniczanie dostępu do zasobów, można zaktualizować bazę do wersji Premium. Zapewnia ona stabilny poziom wydajności, ponieważ gwarantuje rezerwy mocy i izolowane zasoby. Warto zauważyć, że aktualizacja bazy do tej wersji związana jest z wyższymi kosztami. Aby je ograniczyć, możesz aktualizować bazy do tej wersji tylko na okres, w którym aplikacja lub baza wymagają przewidywalnej wydajności. W okresach niższego obciążenia możesz używać niższej wersji, co pozwoli zmniejszyć koszty.

UWAGA Szczegółowe informacje na temat unikania problemów z połączeniami z aplikacjami znajdziesz pod adresem <http://social.technet.microsoft.com/wiki/contents/articles/1541.windows-azure-sql-database-connectionmanagement.aspx>.

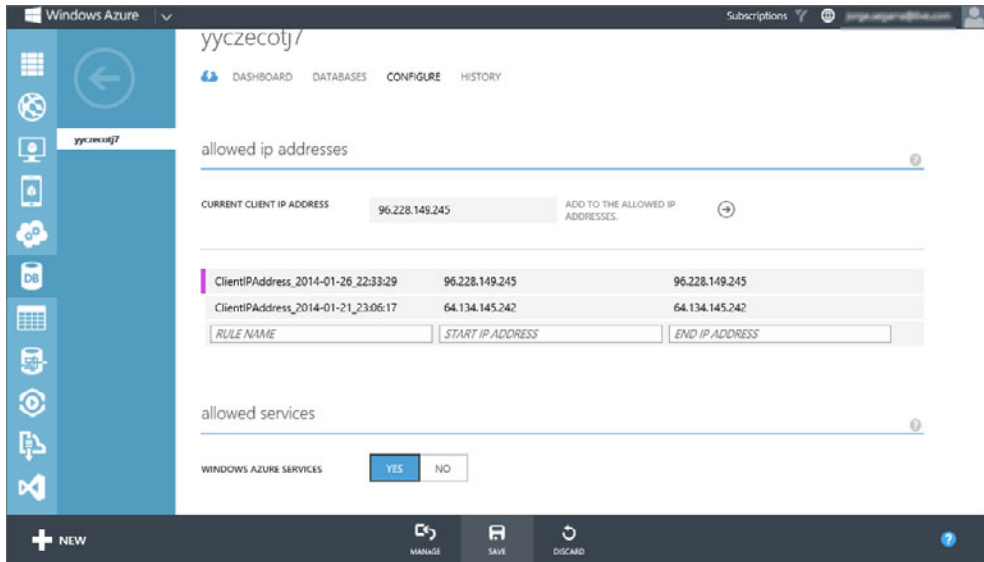
Konfigurowanie zapory dla baz SQL Database

Wiesz już, jak utworzyć serwer i bazy danych, a także znasz działania zaplecza baz SQL Database mające na celu zapewnienie ich wysokiej dostępności i wydajności. Jednak wszystko to będzie nieprzydatne, jeśli nie będziesz mógł połączyć się z bazami danych. W tym obszarze istotna staje się zaporą baz SQL Database.

Możliwe, że napisałeś wspaniałą aplikację, zaprojektowałeś świetną bazę danych i ustawiłeś odpowiedni łańcuch znaków połączenia. Jeśli jednak nie zdefiniowałeś odpowiednich zasad zapory, wszystko to okaże się nieprzydatne. Zapora baz SQL Database to pierwszy poziom zabezpieczeń, który pomaga chronić dane i zapobiegać niepożądanemu dostępowi do serwera baz SQL Database. Do czasu określenia komputerów o odpowiednich uprawnieniach dostęp do serwera baz SQL Database jest w całości blokowany. Żądania nawiązania połączenia przychodzące z internetu, a nawet z samej platformy Azure nie dotrą do serwera baz SQL Database, chyba że określisz, kto ma mieć dostęp do serwera.

Zasady zapory są oparte na adresach IP. Można zdefiniować tylko akceptowalne adresy IP lub ich zakresy. Zasady zapory definiuje się w sekcji *Configure* serwera baz danych. Aby otworzyć tę stronę, kliknij odnośnik *Set up Windows Azure firewall rules for this IP address* widoczny na rysunku 24.8.

W celu skonfigurowania zasad zapory kliknij odnośnik *Manage allowed IP addresses* w sekcji *Quick Glance* w widoku *DASHBOARD* konfigurowanej bazy danych. Pojawi się ekran widoczny na rysunku 24.11. W sekcji *allowed ip addresses* znajdują się trzy pola na dane wyjściowe: *RULE NAME*, *START IP ADDRESS* i *END IP ADDRESS*.



Rysunek 24.11. Ustawianie adresów IP mających dostęp do serwera

Aby szybko dodać nową zasadę zapory dotyczącą obecnie używanego adresu IP, wystarczy kliknąć odnośnik *ADD TO THE ALLOWED IP ADDRESSES*. Jest on widoczny obok tego adresu, co ilustruje rysunek 24.11. Możesz też podać zakres adresów IP, aby umożliwić dostęp dla wielu adresów. Po dodaniu odpowiednich reguł kliknij przycisk *Save* w dolnej części ekranu, by zatwierdzić zmiany.

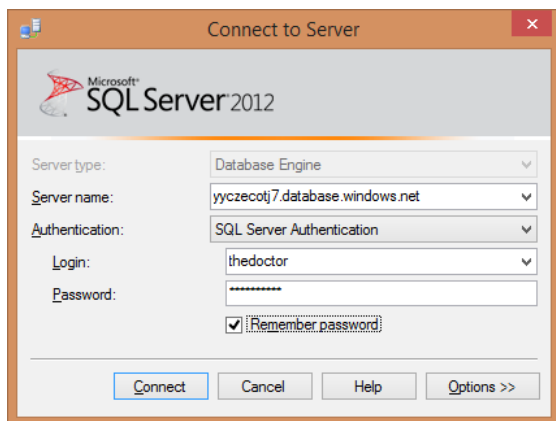
Zauważ, że na rysunku 24.11 widoczna jest też sekcja *allowed services*. Umożliwia ona kontrolowanie, czy inne usługi platformy Windows Azure mogą łączyć się z daną bazą SQL Database. Domyślnie jest to możliwe (opcja *YES*), możesz jednak ustawić opcję *NO* i zastosować niestandardowe zasady, aby pozwolić usługom na dostęp do bazy.

Nawiązywanie połączenia z bazami SQL Database

Gdy już utworzyłeś bazy danych i zdefiniowałeś zasady zapory, możesz nawiązać połączenie z serwerem baz SQL Database. W tym celu wykonaj następujące kroki.

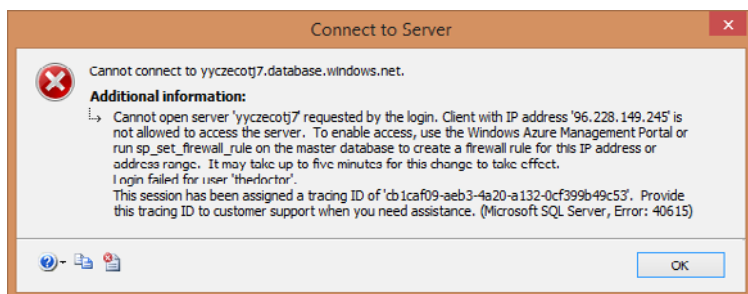
UWAGA W przykładach z tego punktu pokazujemy, jak nawiązać połączenie w programie SQL Server Management Studio, a nie przy użyciu aplikacji. Krótkie omówienie łańcuchów znaków połączenia znajdziesz wcześniej w tym rozdziale.

1. Otwórz program SQL Server Management Studio. Nawiązanie połączenia z bazą SQL Database wymaga użycia programu SQL Server Management Studio z wersji SQL Server 2008 lub nowszej. Gdy pojawi się okno dialogowe *Connect to Server*, widoczne na rysunku 24.12, wprowadź nazwę FQDN serwera. Nadal powinna się ona znajdować w schowku. Jeśli tak nie jest, wróć do strony Azure Management Portal, ponownie skopiuj nazwę, a następnie wklej ją w polu *Server name*.



Rysunek 24.12. Wprowadź nazwę FQDN serwera

2. Bazy SQL Database obsługują tylko uwierzytelnianie z systemu SQL Server, dlatego wybierz je i wprowadź login oraz hasło ustawione w oknie z rysunku 24.6. Jeśli w zaporze bazy SQL Database podałeś odpowiedni adres IP, powinieneś z powodzeniem się połączyć. Jeżeli jednak zasady zapory nie są poprawnie skonfigurowane, pojawi się komunikat o błędzie widoczny na rysunku 24.13.

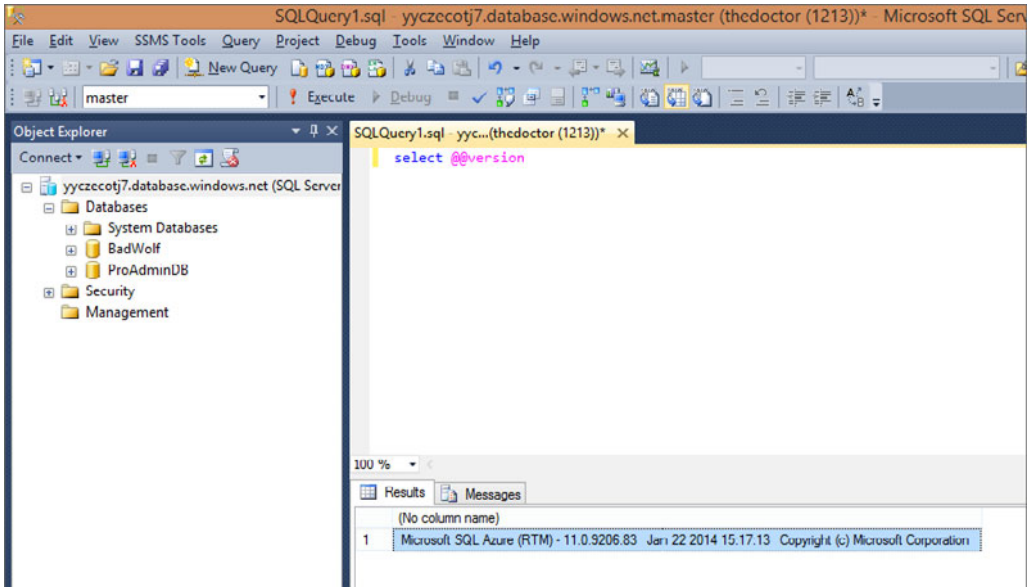


Rysunek 24.13. Komunikat o błędzie wyświetlany po nieudanej próbie połączenia się z bazą

3. W ramach krótkiego ćwiczenia otwórz nowe okno kwerendy (w tym celu wybierz skrót *Ctrl+N* lub kliknij serwer prawym przyciskiem myszy w oknie *Object Explorer* i wybierz opcję *New Query* z menu kontekstowego) i upewnij się, że możesz połączyć się z bazą master. W oknie kwerendy wprowadź następującą instrukcję:

```
SELECT @@VERSION
```

Na rysunku 24.14 widoczne jest okno *Results* wyświetlające wersję baz SQL Database udostępnianą w czasie, gdy powstawała ta książka.



Rysunek 24.14. Sprawdzanie, czy można nawiązać połączenie z bazą master

W ten sposób nawiązałeś połączenie z serwerem baz SQL Database w programie SQL Server Management Studio. Teraz możesz przejść do nauki logicznego administrowania bazami SQL Database. Dowiesz się m.in., jak tworzyć loginy i konta użytkowników.

Administrowanie bazami SQL Database

O bazach SQL Database często błędnie myśli się, że znacznie różnią się od baz z systemu SQL Server. W rzeczywistości obie te technologie wykorzystują ten sam model autoryzacji z kontami użytkowników i rolami tworzonymi w każdej bazie i przypisywanymi do loginów. W systemie SQL Server występują stałe role z poziomu serwera, np. serveradmin, securityadmin i dbcreated, których nie ma w bazach SQL Database. Nie są one tam potrzebne, ponieważ w bazach SQL Database administrowanie dotyczy aspektów logicznych. Zamiast tego dostępne są role loginmanager (służy do tworzenia loginów) i dbmanager (jest przeznaczona do tworzenia baz danych i zarządzania nimi). Te role można przypisać do użytkowników tylko w bazie master.

Tworzenie loginów i kont użytkowników

W bazie SQL Database dostępne są te same obiekty zabezpieczeń (ang. *security principals*), co przy uwierzytelnianiu w systemie SQL Server. Możesz je wykorzystać do autoryzacji użytkowników i zabezpieczania danych. W bazie SQL Database loginy służą do uwierzytelniania dostępu do bazy na poziomie serwera. Konta użytkowników baz są przeznaczone do przyznawania dostępu na poziomie baz, a role służą do grupowania kont użytkowników i przyznawania im dostępu także na poziomie baz danych.

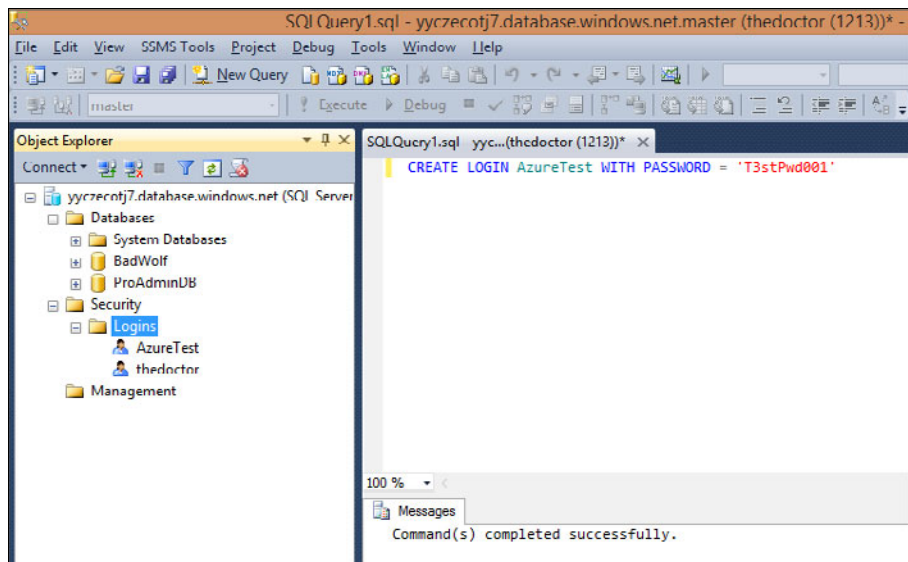
Tworzenie nowego loginu

Tworzenie loginu bardzo przypomina tę operację w systemie SQL Server. Różnicą jest to, że nie można tworzyć loginów na podstawie danych uwierzytelniających z systemu Windows. Dlatego dla wszystkich loginów używane jest uwierzytelnianie z systemu SQL Server. Poniżej opisujemy kroki potrzebne do utworzenia loginu.

1. W programie SQL Server Management Studio otwórz nowe okno kwerendy i nawiąż połączenie z bazą master za pomocą konta administratora. W oknie kwerendy wpisz i wywołaj następujące polecenie:

```
CREATE LOGIN AzureTest WITH PASSWORD = 'T3stPwd001'
```

To spowoduje utworzenie nowego loginu, AzureTest, widocznego na rysunku 24.15.



Rysunek 24.15. Tworzenie nowego loginu

2. Choć na rysunku 24.15 utworzony login jest pokazany w programie SQL Server Management Studio, możesz też wywołać kwerendę do tabeli `sys.sql_logins`, aby wyświetlić wszystkie loginy z serwera:

```
SELECT * FROM sys.sql_logins
```

Po utworzeniu loginów nie można się jeszcze zalogować. Najpierw trzeba utworzyć konta użytkowników, co omawiamy w następnym punkcie.

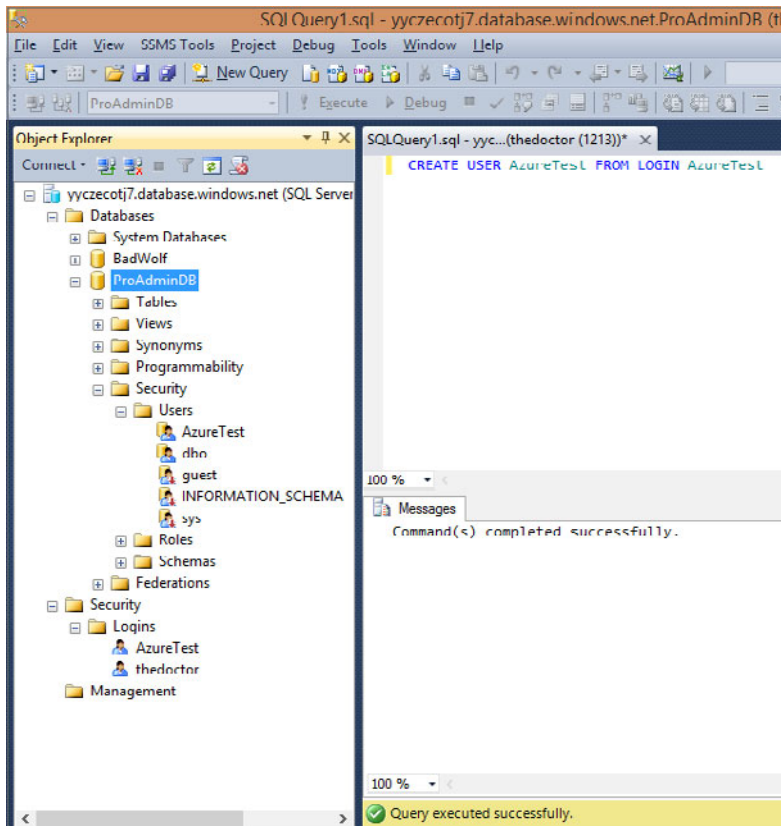
OSTRZEŻENIE Jeśli spróbujesz utworzyć login w bazie użytkownika, wystąpi błąd informujący, że loginy trzeba tworzyć w bazie master. Jeśli hasło nie jest wystarczająco skomplikowane, zobaczysz komunikat o błędzie z informacją, że sprawdzanie poprawności hasła zakończyło się niepowodzeniem, ponieważ hasło nie spełnia zasad systemu Windows. Zgodnie z tymi zasadami hasła mogą obejmować duże i małe litery alfabetu łacińskiego (od A do Z), cyfry z systemu dziesiętnego (od 0 do 9) i znaki niealfanumeryczne. W bazach SQL Database nie można wyłączyć zasad sprawdzania haseł.

Tworzenie nowego konta użytkownika

Po utworzeniu loginu należy dodać konto użytkownika. Aby przygotować takie konto, nawiąż połączenie z określoną bazą użytkownika za pomocą konta administracyjnego i wywołaj poniższe polecenie:

```
CREATE USER AzureTest FROM LOGIN AzureTest
```

Na rysunku 24.16 pokazano wywołane polecenie i uzyskane wyniki. Widać tu, że w bazie ProAdminDB zostało utworzone konto użytkownika.



Rysunek 24.16. Tworzenie nowego konta użytkownika

Jeśli spróbujesz utworzyć konto użytkownika bez wcześniejszego przygotowania loginu, pojawi się komunikat z informacją, że dla użytkownika nie jest ustawiony poprawny login. W bazach SQL Database nie jest dozwolone tworzenie użytkowników bez loginów.

Przypisywanie uprawnień dostępu

Następny krok w procesie administrowania polega na przypisaniu uprawnień dostępu nowo utworzonemu kontu użytkownika. Aby zapewnić kontu AzureTest nieograniczony dostęp do wybranej bazy użytkownika, trzeba przypisać konto do grupy db_owner, co pokazano poniżej:

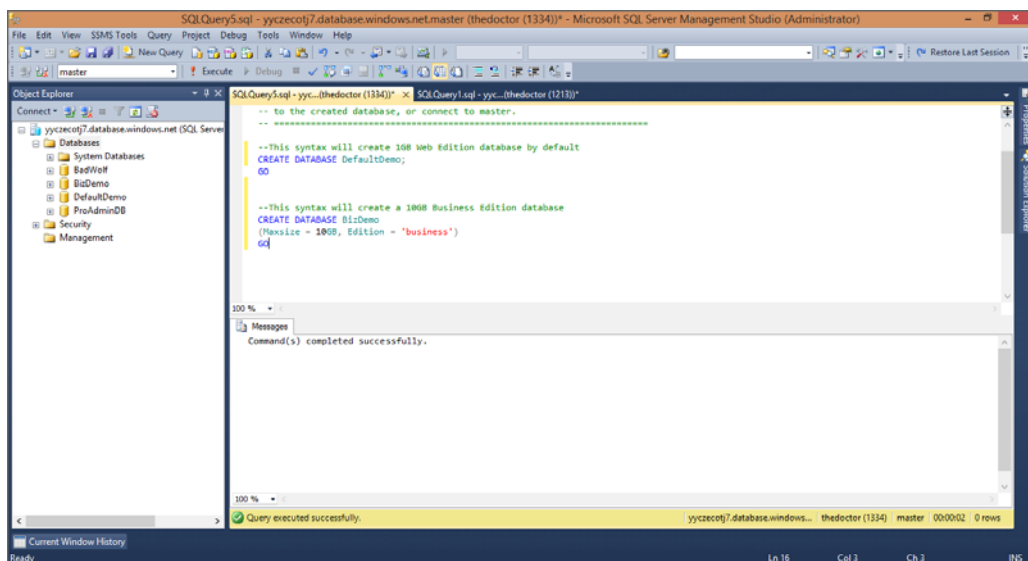
```
EXEC sp_addrolemember 'db_owner', 'AzureTest'
```

Teraz użytkownik AzureTest może tworzyć tabele, widoki, procedury składowane i inne obiekty. Jednak (podobnie jak w systemie SQL Server) można też precyzyjnie konfigurować uprawnienia. Możesz przyznawać i wycofywać uprawnienia insert, update, create i delete w taki sam sposób, jak w systemie SQL Server. Konta użytkowników w systemie SQL Server są automatycznie przypisywane do roli public. W bazach SQL Database jest inaczej, ponieważ ze względów bezpieczeństwa wiązanie roli public z kontami użytkowników jest niedozwolone. Dlatego do konta użytkownika trzeba przypisać określone uprawnienia dostępu.

Praca z bazami SQL Database

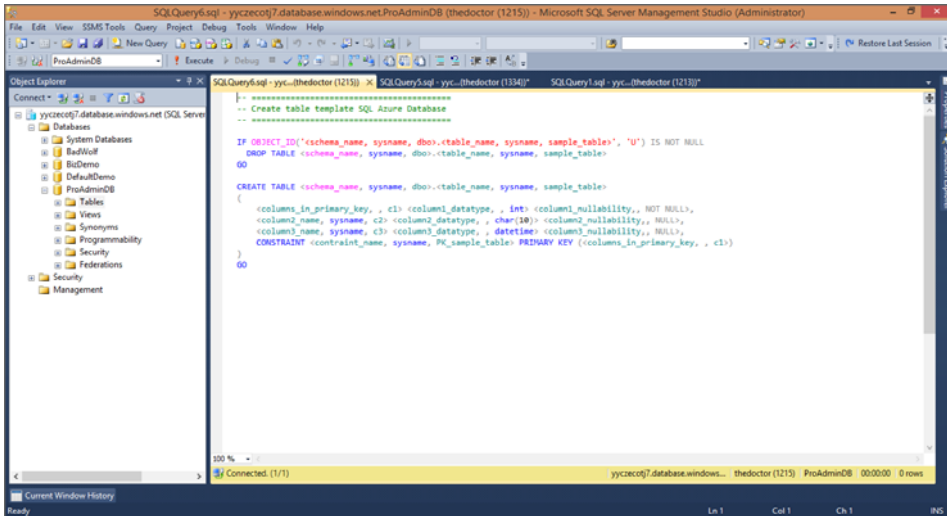
Program SQL Server Management Studio umożliwia wygodne i łatwe tworzenie baz danych oraz tabel, a także wykonywanie wielu zadań z zakresu administracji i konserwacji przy użyciu doskonałego interfejsu użytkownika. Jednak niektóre mechanizmy nie są automatycznie dostępne. Można je wykorzystać za pomocą odpowiedniego kodu. Wykonaj zatem następujące kroki.

1. Kliknij prawym przyciskiem myszy węzeł *Database* w oknie *Object Explorer* i wybierz z menu kontekstowego opcję *New Database*. Pojawi się okno kwerend z szablonem tworzenia nowych baz danych.
2. Zauważ, że w oknie edytora kwerend znajduje się szablonowy kod z instrukcją DDL tworzącą nową bazę danych. Na rysunku 24.17 można zobaczyć kilka przykładowych poleceń. Wykorzystać je można do utworzenia baz danych za pomocą różnych opcji. Określ nazwę bazy, jej wersję i wielkość. Jeśli pominiesz wersję i wielkość, domyślnie wygenerowana zostanie baza Web o pojemności 1 gigabajta.



Rysunek 24.17. Instrukcje tworzące nowe bazy danych

Możesz też napisać kod do generowania tabel i widoków oraz ustawiania uprawnień. Na rysunku 24.18 pokazano efekt kliknięcia prawym przyciskiem myszy węzła *Tables* w oknie *Object Explorer* i wybrania opcji *New Table* z menu kontekstowego.



Rysunek 24.18. Polecenie tworzące nową tabelę

Choć w interfejsie użytkownika nie ma niektórych funkcji, nie oznacza to, że nie istnieją. Aby wykonać większość potrzebnych zadań, wystarczy napisać odpowiedni kod (używając standardowej składni języka T-SQL).

Tworzenie kopii zapasowych baz SQL Database

Jednymi z najczęstszych pytań potencjalnych użytkowników platformy Azure są pytania dotyczące kopii zapasowych. Obecnie kopie zapasowe tworzone w lokalnie zainstalowanym SQL Server nie są w bazach SQL Database dostępne. Zamiast tego używana jest funkcja Database Copy, która umożliwia wykonanie spójnej transakcyjnie kopii głównej bazy SQL Database w innej bazie. Oto składnia tego mechanizmu:

```
CREATE DATABASE DB2 AS COPY OF DB1
```

Ta sama składnia pozwala kopiować bazy między serwerami, przy czym użytkownik musi mieć wtedy na obu serwerach uprawnienia administratora. Jednak administrowanie dwoma serwerami wymaga ponoszenia opłat za dwie bazy danych. Ponadto trudniej wtedy tworzyć codziennie kopie zapasowe i przechowywać historię operacji.

Okno Object Explorer dla baz SQL Database

Gdy w programie SQL Server Management Studio nawiążesz połączenie z bazą SQL Database z platformy Azure, zauważysz brak kilku węzłów, takich jak *Server Objects*, *Replication* i *SQL Server Agent*. W następnych podpunktach wyjaśniamy, dlaczego te węzły (na razie) nie są dostępne i czego możesz spodziewać się w przyszłości.

Server Objects

Przyjrzyj się węzłom podrzędnym z węzła *Server Objects*. Są to:

- *Backup Devices*,
- *SOAP Endpoints*,

- *Linked Servers*,
- *Triggers*.

Oto powody, dla których te elementy nie są dostępne dla baz SQL Database.

- *Backup Devices*. Po pierwsze, ponieważ użytkownicy nie mają dostępu do fizycznego sprzętu, na którym działa baza SQL Database, nie trzeba tworzyć urządzenia przeznaczonego na kopie zapasowe (i tak będzie ono niedostępne). Po drugie, kopie zapasowe baz można tworzyć tylko w opisany wcześniej sposób. Dopóki Microsoft nie udostępni dodatkowych mechanizmów generowania kopii zapasowych, węzeł *Backup Devices* będzie niepotrzebny.
- *SOAP Endpoints*. Punkty końcowe SOAP zostały wprowadzone w wersji SQL Server 2005 i umożliwiają tworzenie usług sieciowych, które zapewniają dostęp do bazy danych przez protokół HTTP. Microsoft poinformował, że w przyszłych wersjach systemu SQL Server ten mechanizm zostanie wycofany. W bazach SQL Database istnieje lepsza technika niż tworzenie punktów końcowych SOAP — protokół OData. Za pomocą tego protokołu można udostępnić dane z bazy SQL Database w kilku kliknięciach myszą. Usługa OData Service dla baz SQL Database to proste i niewymagające pisania kodu rozwiązanie do udostępniania punktów końcowych przez otwarty protokół HTTP.
- *Linked Servers*. Ten węzeł zwykle służy do obsługi kwerend rozproszonych. W bazach SQL Database takie kwerendy nie są obsługiwane, jednak dostępne jest lepsze rozwiązanie, dzięki któremu kwerendy rozproszone odejdą w przeszłość. Pod koniec 2010 roku Microsoft zaprezentował technologię SQL Database Federation, która umożliwia tworzenie partycji i podział danych (ang. *sharding*) w celu zwiększenia skalowalności i wydajności bazy danych.

UWAGA Dzielenie danych to proces podziału logicznej bazy danych aplikacji na mniejsze porcje danych i rozmieszczania tych porcji w wielu fizycznych bazach danych na potrzeby skalowania aplikacji. W tym podejściu tabele bazy danych są dzielone na wiersze i rozdzielane między różne bazy. Taki podział można przeprowadzić bez przestoju, a aplikacje klienckie mają wtedy ciągły dostęp do danych.

W serwisie MSDN dostępne są wpisy na blogach i artykuły, w których szczegółowo opisana jest technologia SQL Database Federation. Serwery połączone nie są w niej potrzebne.

- *Triggers*. Stosowanie wyzwalaczy w środowisku współużytkowanym powoduje różne komplikacje, np. zagrożenia z zakresu bezpieczeństwa i problemy z wydajnością.

Replikacja

Dzięki usługom SQL Database Data Sync Services replikacja w bazach SQL Database nie jest potrzebna. Jak wcześniej wspomniano, dla każdej bazy tworzone są trzy kopie: główna i dwie repliki. Bazy SQL Database odpowiadają za ich synchronizację oraz automatycznie zapewniają wysoką dostępność i nadmiarowość.

Usługi SQL Database Data Sync Services działają znacznie lepiej oraz są łatwiejsze do skonfigurowania i prostsze w użyciu niż replikacja. Są oparte na platformie Sync Framework, dlatego stanowią znacznie bogatsze rozwiązanie do synchronizowania i przenoszenia danych. Obsługują m.in. zarządzanie konfliktami i generowanie raportów o stanie.

Jeśli chcesz replikować dane z jednej bazy w innej, usługi SQL Database Data Sync Services udostępniają łatwy w użyciu oparty na kreatorze interfejs, który zapewnia wielokierunkową synchronizację danych między dwoma (głównymi) bazami SQL Database lub między bazą SQL Database i bazą z zainstalowanego lokalnie systemu SQL Server.

SQL Server Agent

Dla baz SQL Database narzędzie SQL Server Agent nie jest dostępne, jednak w wielu artykułach w internecie wyjaśnione jest, jak zasymulować jego działanie za pomocą mechanizmu ról roboczych platformy Windows Azure.

SQL Server Agent to usługa systemu Windows, która wykonuje zaplanowane przez administratora operacje (*zadania*) i umożliwia generowanie alertów. Role robocze (ang. *worker role*) platformy Windows Azure działają jak usługi systemu Windows, tylko w chmurze. *Role robocze* służą do wykonywania operacji w tle i długotrwałych lub przerywanych zadań.

Ważną zaletą ról roboczych jest to, że ich struktura bardzo przypomina strukturę usług systemu Windows. Możliwe jest np. uruchamianie, zatrzymywanie i konfigurowanie ich w sposób typowy dla tych usług. Pod adresem <http://blogs.msdn.com/b/sqlazure/archive/2010/07/30/10044271.aspx> znajdziesz doskonały wpis na ten temat.

Innym rozwiązaniem jest użycie zainstalowanego lokalnie systemu SQL Server i narzędzia SQL Server Agent do wykonywania skryptów i zadań powłoki PowerShell systemu Windows. Pozwalają one zarządzać bazą SQL Database przy użyciu cmdletów platformy Windows Azure. Więcej o cmdletach dowiesz się ze strony <http://msdn.microsoft.com/en-us/library/windowsazure/jj554332.aspx>.

Czego brakuje w bazach SQL Database?

Gdy zaczynasz używać baz SQL Database lub nawet jeśli korzystasz z nich od pewnego czasu, z pewnością nieraz będziesz się zastanawiał, dlaczego nie udostępniają one określonych funkcji (np. wyszukiwania pełnotekstowego). Kiedy porównasz bazy SQL Database z systemem SQL Server funkcja po funkcji, szybko dostrzeżesz, że w bazach SQL Database niektóre mechanizmy są niedostępne.

Nie wyjaśniono publicznie powodu niedostępności niektórych narzędzi. Jednak w tym podrozdziale przedstawiamy informacje na temat wybranych brakujących mechanizmów.

Środowisko SQLCLR (ang. *SQL Common Language Runtime*) jest częściowo obsługiwane. Przykładowo XML-owe i przestrzenne typy danych (są dostępne w bazach SQL Database) to typy ze środowiska CLR. Bazy SQL Database nie obsługują jednak możliwości tworzenia komponentów z kodem zarządzanym (patrz rozdział 7., „Integrowanie systemu SQL Server ze środowiskiem CLR”) ani instalowania takich komponentów. Nie wiadomo, czy i kiedy te funkcje zostaną dodane.

Microsoft ciężko pracuje nad udostępnieniem szyfrowania, jednak wymaga to rozwiązania pewnych problemów. Poważną trudnością jest obsługa wielu poziomów szyfrowania w środowisku współużytkowanym. Pamiętaj, że bazy SQL Database działają w środowisku współużytkowanym, dlatego są zapisywane w różnych miejscach. Przykładowo bazy Twoja i współpracownika mogą działać na tym samym serwerze. W tym scenariuszu występują problemy, gdy Ty używasz jednego poziomu szyfrowania, a Twój współpracownik innego.

To tylko kilka przykładów brakujących funkcji. Dla poszczególnych mechanizmów wyjaśnienie niedostępności jest inne. Zauważ np., że w platformie Azure nie są na razie dostępne narzędzia z obszaru BI (takie jak usługi Reporting Services, Analysis Services i Integration Services). Microsoft pracuje nad nimi, jednak na razie nie wiadomo, kiedy i w jaki sposób zostaną udostępnione.

Wniosek z tego jest taki, że niektóre funkcje się nie pojawiają, ponieważ ich dodawanie nie ma sensu. Jeśli chodzi o inne mechanizmy, należy zachować cierpliwość.

Podsumowanie

Rozdział miał zapewnić solidne wprowadzenie do konfigurowania baz SQL Database i administrowania nimi. Opisano tutaj tworzenie serwera baz SQL Database, tworzenie baz danych i mechanizmy działające na zapleczu, które zapewniają wysoką dostępność i przełączanie awaryjne potrzebne w rozwiązaniach opartych na chmurze.

Logiczne administrowanie bazami SQL Database to ważny aspekt przy tworzeniu serwerów baz SQL Database i samych baz. Choć istnieją różnice między instalowanym lokalnie systemem SQL Server a bazami SQL Database, bazy SQL Database można traktować jak działający w chmurze system SQL Server, o równie dużych i różnorodnych możliwościach.

W rozdziale 25. omawiamy następną ciekawą funkcję systemu SQL Server, przydatną przy generowaniu raportów i zapewnianiu wysokiej dostępności. Są to grupy dostępności AlwaysOn.

Grupy dostępności AlwaysOn

ZAWARTOŚĆ ROZDZIAŁU:

- Przedstawienie wszystkich aspektów grup dostępności i ich specyfiki.
- Analiza konfiguracji sprzętu i oprogramowania potrzebna do ustalenia ustawień optymalnych dla firmy.
- Konfigurowanie sprzętu i uruchamianie systemu operacyjnego w klastrze.
- Konfigurowanie grup dostępności z systemem SQL Server 2014.
- Administrowanie produkcyjną grupą dostępności z systemem SQL Server 2014 i rozwiązywanie problemów.
- Testowanie i monitorowanie grup dostępności oraz dostrajanie ich wydajności.

Grupy dostępności to funkcja technologii AlwaysOn z systemu SQL Server 2014, oferująca rozszerzenia zapewniające wysoką dostępność. Znacznie usprawnia ona mechanizmy tworzenia kopii lustrzanych dostępne od wersji SQL Server 2005. W ramach technologii AlwaysOn system SQL Server 2014 oferuje grupy dostępności, które zapewniają możliwości podobne do możliwości kopii lustrzanych. Oprócz wspomnianych rozszerzeń możliwa jest obsługa do dziewięciu partnerów w mechanizmie tworzenia replik, obsługa wielu baz w jednej grupie i tworzenie serwerów pomocniczych z replikami przeznaczonymi do odczytu. Serwery główne i pomocnicze można łączyć w trybach asynchronicznych i synchronicznych. Nie są przy tym potrzebne współużytkowane dyski. Dzięki temu serwery można instalować w odległych fizycznie lokalizacjach.

Gdy bazy danych są od siebie zależne, można połączyć je w grupę dostępności, aby były przełączane awaryjnie wspólnie. Jeśli np. utworzysz serwery z replikami do odczytu, będziesz mógł przenieść operacje związane z generowaniem raportów lub samym odczytem na replikę pomocniczą. Pozwala to zmniejszyć obciążenie repliki głównej, zapewnić skalowanie obciążenia w środowisku produkcyjnym oraz szybciej przetwarzać zapytania. Ponadto możesz tworzyć kopie

zapasowe przy użyciu repliki pomocniczej, co ogranicza obciążenie związane z tym zadaniem na serwerze głównym. Grupy dostępności obsługują wbudowane szyfrowanie i kompresję przesyłanych danych, przełączanie awaryjne (automatyczne, ręczne i wymuszone), konfigurowalne zasady przełączania awaryjnego, a także automatyczne przekierowywanie aplikacji z wykorzystaniem nazwy sieciowej odbiornika grupy dostępności.

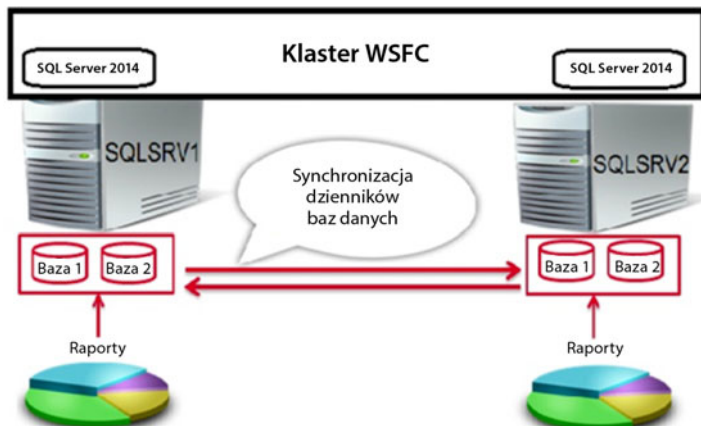
Grupy dostępności zapewniają obsługę wielu funkcji i komponentów systemu SQL Server. Oto one:

- rejestrowanie zmian w danych,
- śledzenie zmian,
- niezależne bazy danych,
- szyfrowanie baz danych,
- snapshoty baz danych,
- formaty FileStream i FileTable,
- wyszukiwanie pełnotekstowe,
- dodatek RBS (ang. *Remote Blob Storage*),
- replikacja,
- Service Broker,
- SQL Server Agent.

W tym rozdziale szczegółowo omawiamy grupy dostępności AlwaysOn.

Architektura

Grupa dostępności to zbiór obejmujący do dziewięciu serwerów z systemem SQL Server ze skonfigurowanym przełączaniem awaryjnym zestawu baz danych użytkowników (*baz wysokiej dostępności*). Każdy zestaw baz z zapewnioną dostępnością jest przechowywany w *replce wysokiej dostępności*, a każda taka replika ma określoną rolę. Każda replika wysokiej dostępności ma też ustalony *tryb dostępności*. Na rysunku 25.1 szczegółowo przedstawiamy architekturę grup dostępności, a w dalszych punktach omawiamy jej elementy.



Rysunek 25.1. Architektura grup dostępności

Repliki i role w grupach dostępności

Wcześniej wspomniano, że bazy wysokiej dostępności są przechowywane w replikach wysokiej dostępności. Istnieją dwa rodzaje takich replik.

- Pojedyncza replika główna zapewnia klientom dostępność głównych baz danych w trybie do odczytu i zapisu. Ponadto przesyła rekordy z dzienników transakcji każdej bazy głównej do wszystkich replik pomocniczych.
- Repliki pomocnicze (od 1 do 8) przechowują zestaw pomocniczych baz danych. Każda replika pomocnicza wprowadza rekordy z dzienników transakcji we własnym zestawie pomocniczych baz danych i może być używana przy przełączaniu awaryjnym w grupie dostępności.

Opcjonalnie można skonfigurować repliki pomocnicze zapewniające dostęp do pomocniczych baz danych w trybie tylko do odczytu. Repliki pomocnicze mogą też służyć do tworzenia kopii zapasowych pomocniczych baz danych.

Do każdej repliki wysokiej dostępności przypisywana jest początkowa rola — *rola główna* lub *pomocnicza* — dziedziczona przez bazy wysokiej dostępności po danej replice. Rola danej repliki określa, czy przechowywane w niej bazy są przeznaczone do odczytu i zapisu, czy tylko do odczytu. Replika główna pełni rolę główną i przechowuje bazy dostępne do odczytu i zapisu (są to *główne bazy danych*). Przynajmniej jedna replika pomocnicza pełni rolę pomocniczą. Repliki pomocnicze przechowują bazy działające w trybie tylko do odczytu (*pomocnicze bazy danych*).

Aby replika mogła działać w grupie dostępności, musi być zainstalowana w klastrze WSFC. Jeśli zatem chcesz utworzyć grupę dostępności, musisz najpierw dodać do klastra WSFC każdy serwer z grupy. W klastrze WSFC grupa dostępności jest zasobem klastra z własną nazwą sieciową i adresem IP. Nazwa i adres są używane przez aplikacje klienckie do łączenia się z grupą dostępności. Główna różnica między systemem SQL Server pracującym w klastrze a grupą dostępności jest taka, że grupa nie wymaga współużytkowanego dysku. W tej technologii serwery nie korzystają z tych samych dysków.

Tryby dostępności

Każda replika wysokiej dostępności ma właściwość określającą tryb dostępności. Dotyczy on sposobu obsługi rekordów transakcji i określa, jak aktualne są dane w replikach pomocniczych w porównaniu z danymi z repliki głównej. Oto dwa stosowane tryby dostępności.

- **Tryb z asynchronicznym zatwierdzaniem.** Replika główna zatwierdza transakcje bez oczekiwania na potwierdzenie zapisania dziennika transakcji przez replikę pomocniczą z włączonym asynchronicznym zatwierdzaniem. Ten tryb pozwala zminimalizować opóźnienie transakcji w bazach z repliki pomocniczej, ale sprawia, że mogą być one nieaktualne w porównaniu z bazami z repliki głównej. Zwiększa to prawdopodobieństwo awarii i utraty danych.
- **Tryb z zatwierdzaniem synchronicznym.** Replika główna oczekuje na potwierdzenie zapisania dziennika transakcji przez repliki pomocnicze z włączonym synchronicznym potwierdzaniem. Ten tryb gwarantuje, że gdy dana baza z repliki pomocniczej jest zsynchronizowana z bazą z repliki głównej, zatwierdzone transakcje są w pełni zabezpieczone. Ta ochrona jest uzyskiwana kosztem zwiększenia opóźnienia w obsłudze żądań. Wydłuża to czas przetwarzania transakcji użytkowników, ponieważ opóźnienie wpływa na ogólny czas reakcji bazy danych.

Obsługiwane tryby przełączania awaryjnego

W trakcie przełączania awaryjnego repliki główne i pomocnicze zamieniają się rolami. Jedna z replik pomocniczych staje się repliką główną, a główna staje się repliką pomocniczą. Działające w grupie dostępności bazy z nowej repliki głównej przechodzą w tryb online i są udostępniane do odczytu oraz zapisu. Dawna replika główna po przełączeniu awaryjnym staje się repliką pomocniczą i zaczyna odbierać dane o transakcjach z nowej repliki głównej. Jeśli przełączenie awaryjne zostało spowodowane awarią dawnej repliki głównej, np. wskutek jej tymczasowej niedostępności, to po wznowieniu pracy ta replika automatycznie dołącza do grupy dostępności jako replika pomocnicza.

Istnieją trzy tryby przełączania awaryjnego: automatyczny, ręczny i wymuszony (może on prowadzić do utraty danych). To, które z nich są obsługiwane, zależy od używanego trybu dostępności. Przedstawiamy to w tabeli 25.1.

Tabela 25.1. Obsługiwane tryby przełączania awaryjnego

Tryb dostępności	Ustawienie trybu przełączania awaryjnego	Obsługiwany sposób przełączania awaryjnego do repliki pomocniczej
Synchroniczny	Ręczny	Ręczny (bez utraty danych)
Synchroniczny	Automatyczny	Automatyczny lub ręczny (bez utraty danych)
Asynchroniczny	Ręczny	Ręczny (z możliwą utratą danych)

Poniżej szczegółowo opisujemy wszystkie tryby przełączania awaryjnego wymienione w tabeli 25.1.

- **Ręczne przełączanie awaryjne bez utraty danych.** Ręczne przełączanie awaryjne ma miejsce wtedy, gdy administrator wywoła odpowiednie polecenie. W efekcie zsynchronizowana replika pomocnicza staje się repliką główną (z gwarantowaną ochroną danych), natomiast replika główna staje się repliką pomocniczą. Administrator może wskazać replikę pomocniczą przejmującą rolę repliki głównej. Ręczne przełączanie awaryjne wymaga tego, aby replika główna i docelowa replika pomocnicza używały trybu z zatwierdzaniem synchronicznym. Ponadto replika pomocnicza musi być zsynchronizowana.
- **Automatyczne przełączanie awaryjne bez utraty danych.** Automatyczne przełączanie awaryjne jest wykonywane w reakcji na błąd i powoduje zmianę zsynchronizowanej repliki pomocniczej w replikę główną (z gwarantowaną ochroną danych). Gdy dawna replika główna wznowia pracę, staje się repliką pomocniczą. Automatyczne przełączanie awaryjne wymaga, aby replika główna i docelowa replika pomocnicza używały trybu z zatwierdzaniem synchronicznym. Ponadto włączony musi być automatyczny tryb przełączania awaryjnego, a replika pomocnicza musi być zsynchronizowana.
- **Ręczne przełączanie awaryjne z możliwą utratą danych.** W trybie z zatwierdzaniem asynchronicznym możliwe jest tylko ręczne wymuszone przełączanie awaryjne z możliwą utratą danych. Wymuszone przełączanie awaryjne należy stosować tylko przy odzyskiwaniu systemu po katastrofie, ponieważ może prowadzić do utraty danych. Gdy docelowa replika pomocnicza nie jest zsynchronizowana z repliką główną, możliwe jest tylko ręczne przełączanie awaryjne (nawet w trybie z zatwierdzaniem synchronicznym).

Umożliwianie dostępu do replik pomocniczych w trybie tylko do odczytu

Grupy dostępności mają przede wszystkim zapewniać wysoką dostępność i umożliwiać odzyskiwanie danych po katastrofie. Ponadto repliki pomocnicze można tak skonfigurować, aby umożliwiały dostęp do baz w trybie tylko do odczytu. W takim scenariuszu operacje wymagające samego odczytu można wykonywać w bazach z replik pomocniczych i zoptymalizować replikę główną, tak aby obsługiwała operacje o znaczeniu strategicznym wymagające dostępu do baz z możliwością odczytu i zapisu.

Jeśli np. chcesz generować raporty, to zamiast uruchamiać je w replice głównej i przeciążać ją, możesz wykonywać je w jednej z replik pomocniczych. Ponadto możesz tworzyć kopie zapasowe baz danych przy użyciu repliki pomocniczej. Obsługiwane są wtedy pełne kopie zapasowe oraz kopie plików i grup plików. Nie są natomiast obsługiwane różnicowe kopie zapasowe. W replikach pomocniczych można też tworzyć kopie zapasowe dzienników transakcji. Służą one tylko do kopiowania danych. Szczegółowe omówienie tego zagadnienia podajemy w podrozdziale „Tworzenie kopii zapasowych w replice pomocniczej”, dalej w tym rozdziale.

Oto lista możliwości replik pomocniczych udostępnianych w trybie tylko do odczytu.

- Dane w replice pomocniczej są synchronizowane niemal w czasie rzeczywistym względem repliki głównej. Zależy to od opóźnień w przesyłaniu danych siecią między replikami główną a pomocniczą, czasu wprowadzania zmian w replice pomocniczej, blokad baz pomocniczych i procesów wykonywanych na serwerze pomocniczym.
- Możliwość dostępu w trybie tylko do odczytu dotyczy wszystkich baz z grupy dostępności.
- Aby znacznie ograniczyć blokady w replikach pomocniczych działających w trybie tylko do odczytu, bazy w tych replikach domyślnie działają w trybie izolacji snapshota. Dzięki temu operacje odczytu nie blokują dziennika wprowadzanych transakcji, służącego do aktualizowania danych. Tryb izolacji snapshota powoduje dodawanie 14-bajtowego identyfikatora wiersza, co pozwala zachować informacje o wersji wiersza, a także zapisywanie ostatnich zatwierdzonych wierszy w bazie tempdb w trakcie wprowadzania aktualizacji w bazach danych w replice.
- Aby zoptymalizować zadania wymagające tylko odczytu, w bazie z repliki głównej należy utworzyć indeks lub statystyki potrzebne w replice pomocniczej i przesłać je do wszystkich replik przez przeniesienie danych z dzienników transakcji. Do obsługi zadań wymagających tylko odczytu system SQL Server może tworzyć tymczasowe statystyki przechowywane w bazie tempdb. Te statystyki zostają utracone w momencie ponownego uruchomienia egzemplarza systemu SQL Server lub w momencie zmiany repliki pomocniczej w główną w trakcie przełączania awaryjnego.
- Grupy dostępności, podobnie jak kopie lustrzane, obsługują automatyczne naprawianie stron. Każda replika próbuje automatycznie naprawić uszkodzone strony lokalnej bazy danych. W tym celu reaguje na określone typy błędów uniemożliwiające odczyt stron. Jeśli replika pomocnicza nie może odczytać strony, żąda jej świeżej kopii z repliki głównej. Jeżeli replika główna nie może odczytać strony, rozsyła żądanie świeżej kopii do wszystkich replik pomocniczych i pobiera stronę z pierwszej repliki, która odpowie. Po udanym odebraniu odpowiedzi można zastąpić nieczytelną stronę kopią, co zwykle rozwiązuje problem.
- Aplikacje klienckie mogą łączyć się z replikami przeznaczonymi tylko do odczytu. Połączenie z repliką pomocniczą, która obsługuje dostęp tylko do odczytu, można nawiązać na dwa sposoby — bezpośrednio przez określenie nazwy egzemplarza systemu

SQL Server w łańcuchu znaków połączenia lub przy użyciu nazwy odbiornika grupy dostępności (następuje wtedy przekierowywanie operacji wymagających samego odczytu do następnej dostępnej repliki pomocniczej działającej w trybie tylko do odczytu). Szczegółowe omówienie tego tematu znajdziesz w punkcie „Łączenie się klientów z repliką pomocniczą”, dalej w tym rozdziale. Dostęp do repliki pomocniczej jest zależny od właściwości grupy dostępności, co przedstawiono w tabeli 25.2.

Tabela 25.2. Ustawienia replik pomocniczych

Replika pomocnicza umożliwia odczyt?	Opis
Nie (opcja No)	Replika pomocnicza nie umożliwia odczytu. Wszystkie próby nawiązania połączenia z repliką pomocniczą są blokowane.
Tylko w celu odczytu (opcja Read-intent only)	Replika pomocnicza umożliwia odczyt, ale akceptuje wyłącznie połączenia w trybie READONLY. Tryb READONLY to właściwość połączeń dostępna w nowym protokole TDS (ang. <i>Tabular Data Stream</i>). Starsze aplikacje klienckie i programy, które nie mają ustawionego trybu READONLY, nie mogą nawiązać połączenia z repliką pomocniczą dostępną tylko do odczytu. Aby połączyć się z taką repliką, trzeba umieścić w łańcuchu znaków człon <code>ApplicationIntent = ReadOnly</code> .
Tak (opcja Yes)	Replika pomocnicza umożliwia odczyt. Akceptuje wtedy wszystkie połączenia, także te bez ustawionego trybu READONLY w łańcuchu znaków połączenia. To umożliwia starszym aplikacjom klienckim łączenie się z replikami pomocniczymi na potrzeby odczytu. Obsługiwane są wtedy tylko polecenia odczytu. Instrukcje, które próbują tworzyć lub modyfikować dane, powodują wyświetlenie następującego komunikatu o błędzie: <code>Msg 3906, Level 16, State 1, Server SQLHA2, Line 1 Failed to update database "AdventureWorks" because the database is read-only.</code>

UWAGA W grupie dostępności nie można umieszczać baz master, msdb i model. Obsługiwane są tylko bazy użytkowników.

Przykład ilustrujący grupę dostępności

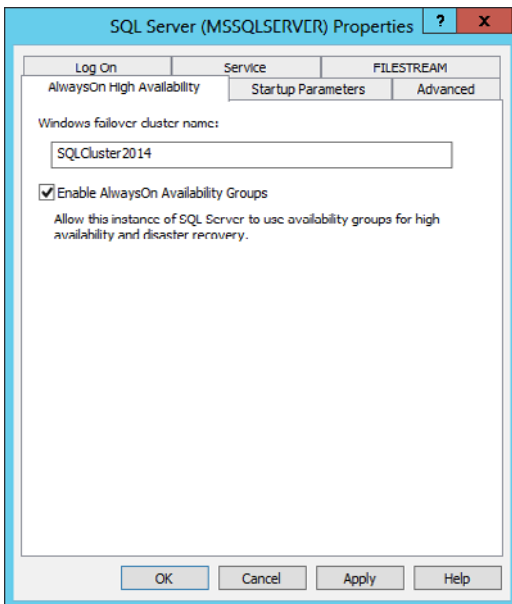
Aby poznać grupy dostępności, warto wykonać ćwiczenie pokazujące, jak krok po kroku utworzyć i skonfigurować taką grupę. Zaczniemy od skonfigurowania nowej grupy dostępności, a następnie dodamy nową replikę. Potem zobaczysz, jak utworzyć nową bazę wysokiej dostępności i jak nawiązać połączenie z grupą dostępności. Po zainstalowaniu grupy dostępności można wykonać próbne przełączenie awaryjne.

UWAGA Wszystkie przykładowe skrypty z tego rozdziału znajdziesz w powiązanim z nim katalogu w archiwum, które można pobrać z witryny wydawnictwa Helion. Przed uruchomieniem skryptów trzeba wprowadzić kilka modyfikacji.

Konfigurowanie nowej grupy dostępności

Wcześniej wspomniano, że przed zainstalowaniem grupy dostępności trzeba utworzyć klastr WSFC. Poniżej w początkowych krokach opisano tworzenie klastra WSFC w systemie Windows 2012 R2. W dalszych krokach przedstawiony jest cały proces konfigurowania nowej grupy dostępności.

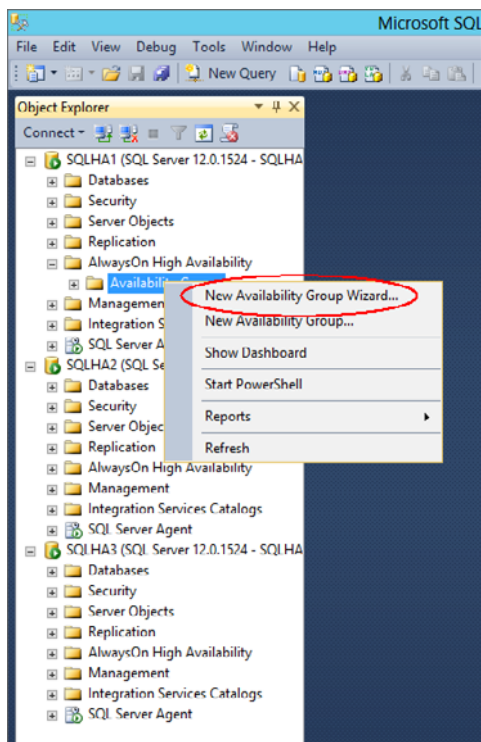
1. Zaczynij od zainstalowania klastra WSFC z systemem Windows 2012 R2 na każdym serwerze, który ma działać w grupie dostępności. W rozdziale 16., „Klastrowanie w systemie SQL Server 2014”, możesz się dowiedzieć, jak to zrobić.
2. Zainstaluj system SQL Server 2014 jako niezależny egzemplarz na każdym serwerze, który ma działać w grupie dostępności. Opis instalacji znajdziesz w rozdziale 2., „Najlepsze praktyki związane z instalowaniem systemu SQL Server 2014”.
3. Następnie w każdym systemie SQL Server, który ma działać w grupie dostępności, otwórz program SQL Server Configuration Manager, wybierz opcję *SQL Server Services*, a następnie wyświetl właściwości systemu SQL Server. W zakładce *AlwaysOn High Availability* zaznacz opcję *Enable AlwaysOn Availability Groups*, tak jak na rysunku 25.2. Następnie należy ponownie uruchomić system SQL Server, aby nowe ustawienie zostało uwzględnione.



Rysunek 25.2. Włączanie obsługi grup dostępności

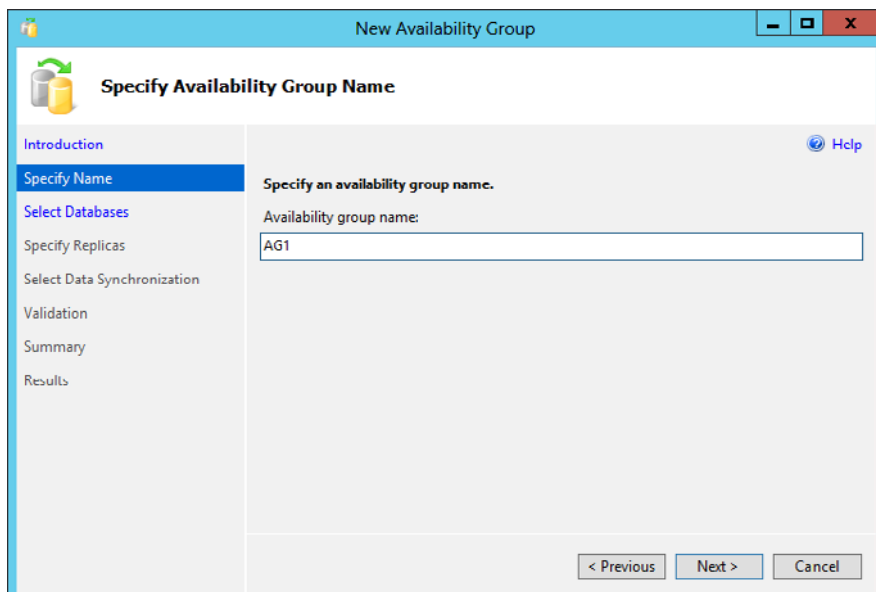
Teraz możesz przystąpić do tworzenia grupy dostępności. Wykonaj następujące kroki.

1. Wybierz egzemplarz systemu SQL Server, który ma być repliką główną. W tym egzemplarzu muszą znajdować się bazy używane w grupie dostępności. Tu tę funkcję pełni egzemplarz SQLHA1 z bazą AdventureWorks.
2. W programie SQL Server Management Studio rozwiń węzeł egzemplarza systemu SQL Server (tu jest nim SQLHA1), kliknij odpowiedni węzeł prawym przyciskiem myszy i wybierz opcję *New Availability Group Wizard*, co przedstawiono na rysunku 25.3.



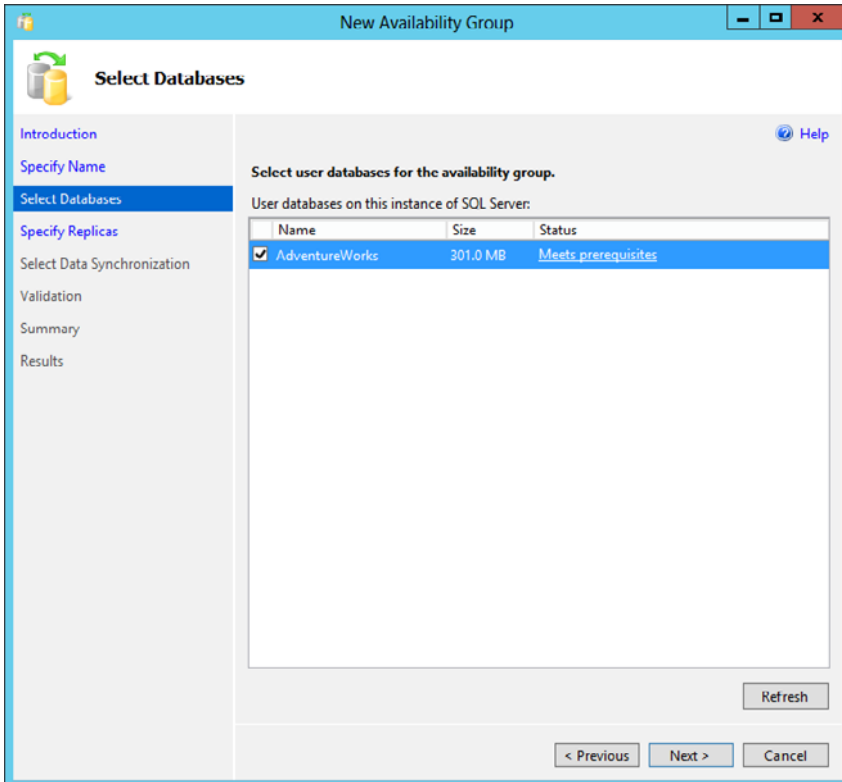
Rysunek 25.3. Uruchamianie kreatora grupy dostępności

3. Na stronie *Introduction* kliknij przycisk *Next*. Na stronie *Specify Availability Group Name* ustaw nazwę grupy dostępności. Na rysunku 25.4 jest to nazwa *AG1*.



Rysunek 25.4. Ustawianie nazwy grupy dostępności

4. Kliknij przycisk *Next*. Na stronie *Select Databases*, pokazanej na rysunku 25.5, zaznacz pola wyboru przy bazach danych (np. przy bazie AdventureWorks). Na liście pojawiają się bazy spełniające następujące warunki wstępne.
 - Baza musi być bazą użytkownika. W grupach dostępności nie można używać systemowych baz danych.
 - Baza musi być przeznaczona do odczytu i zapisu. W grupach dostępności nie można używać baz tylko do odczytu.
 - Baza musi być wielodostępna.
 - Nie może być używana opcja AUTO_CLOSE.
 - Używany musi być model odzyskiwania FULL.
 - Dostępna musi być pełna kopia zapasowa bazy.
 - Baza musi działać w egzemplarzu systemu SQL Server, w którym tworzona jest grupa dostępności, i być dostępna dla tego egzemplarza.
 - Baza nie może należeć do innej grupy dostępności.
 - Baza nie może być używana w mechanizmie tworzenia kopii lustrzanych.



Rysunek 25.5. Ustawianie baz dostępnych w grupie dostępności

5. Kliknij przycisk *Next*. W zakładce *Replicas* na stronie *Specify Replicas* (przedstawionej na rysunku 25.6) wybierz repliki pomocnicze, która mają działać w grupie dostępności. W tabeli 25.3 opisane są wszystkie opcje dostępne na tej stronie. Po ustawieniu opcji kliknij przycisk *Next*.

Specify Replicas

Introduction

Specify Name

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Specify an instance of SQL Server to host a secondary replica.

Replicas

Endpoints

Backup Preferences

Listener

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Secondary
SQLHA1	Primary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
SQLHA2	Secondary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
SQLIA3	Secondary	<input type="checkbox"/>	<input type="checkbox"/>	Read-intent only

Add Replica...

Add Azure Replica...

Remove Replica

Summary for the replica hosted by SQLHA3

Replica mode: Asynchronous commit

This replica will use asynchronous-commit availability mode and support only forced failover (with possible data loss).

Readable secondary: Read-intent only

In the secondary role, this availability replica will only allow read-intent connections.

< Previous

Next >

Cancel

Rysunek 25.6. Ustawianie replik pomocniczych

Tabela 25.3. Opis opcji replik dostępnych w 5. kroku

Początkowa rola	Określanie początkowych replik głównych i pomocniczych
Automatic Failover (Up to 2)	Jeśli chcesz skonfigurować automatyczne przełączanie awaryjne, ustaw do dwóch replik wysokiej dostępności używanych w charakterze partnerów przy automatycznym przełączaniu awaryjnym. Jednym z wybranych partnerów musi być początkowa replika główna. Dla obu wybranych replik należy ustawić tryb dostępności z zatwierdzaniem synchronicznym. Przy automatycznym przełączaniu awaryjnym obsługiwane są tylko dwie repliki.
Synchronous Commit (Up to 3)	Tu można ustawić dodatkowe repliki w trybie synchronicznego zatwierdzania, używane przy planowanym ręcznym przełączaniu awaryjnym. W tym trybie mogą działać tylko trzy repliki. Nie zaznaczaj pola w tej kolumnie, jeśli replika ma używać trybu dostępności z zatwierdzaniem asynchronicznym. Wtedy replika będzie obsługiwać tylko wymuszane ręczne przełączanie awaryjne (z możliwą utratą danych).

Tabela 25.3. Opis opcji replik dostępnych w 5. Kroku — ciąg dalszy

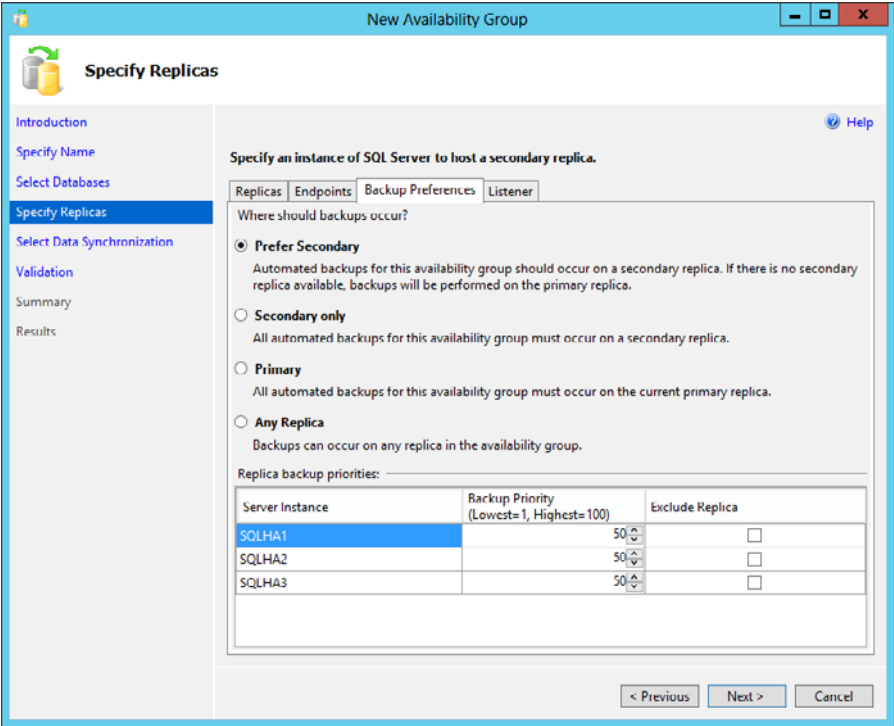
Początkowa rola	Określanie początkowych replik głównych i pomocniczych
<i>Readable Secondary</i>	<p>No. Nie są dozwolone bezpośrednie połączenia z pomocniczymi bazami z tej repliki. Te bazy nie są dostępne do odczytu. Jest to ustawienie domyślne.</p> <p>Read-intent only. Z bazami pomocniczymi z danej repliki można się łączyć tylko w trybie READONLY. Wszystkie te bazy są dostępne do odczytu.</p> <p>Yes. Wszystkie połączenia z bazami pomocniczymi z danej repliki są dozwolone, ale tylko na potrzeby odczytu. Wszystkie te bazy są dostępne do odczytu.</p>

6. W zakładce *Backup Preferences* na stronie *Specify Replicas* (patrz rysunek 25.7) określ, gdzie mają być tworzone kopie zapasowe. Oto dostępne opcje.
 - **Prefer Secondary.** Określa, że kopie zapasowe mają być wykonywane w replice pomocniczej, chyba że dostępna jest tylko replika główna (wtedy kopie zapasowe są tworzone w niej). Jest to ustawienie domyślne.
 - **Secondary only.** Określa, że nigdy nie należy wykonywać kopii zapasowych w replice głównej. Jeśli dostępna jest tylko replika główna, kopie zapasowe w ogóle nie są tworzone.
 - **Primary.** Określa, że kopie zapasowe zawsze mają być wykonywane w replice głównej. Opcja umożliwia tworzenie różnicowych kopii zapasowych, które nie zawsze są obsługiwane, gdy kopie są wykonywane w replice pomocniczej.
 - **Any Replica.** Określa, że zadanie wykonujące kopie zapasowe ma ignorować rolę replik wysokiej dostępności, gdy wybiera replikę, w której tworzona jest taka kopia. Zadanie może uwzględniać wtedy inne czynniki: ustawienie *Backup Priority* poszczególnych replik, ich stan oraz status połączenia.
7. W zakładce *Listener* na stronie *Specify Replicas* (patrz rysunek 25.8) wybierz opcję *Create an availability group listener*. Podaj nazwę odbiornika DNS. Jest to nazwa sieciowa, której aplikacje klienckie będą używać do łączenia się z daną grupą dostępności.

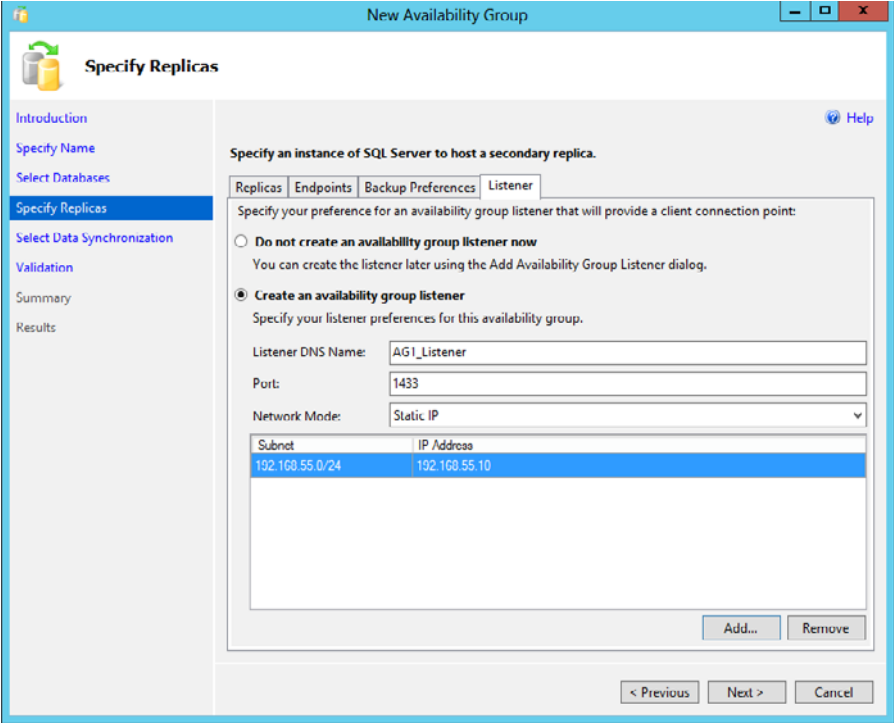
System SQL Server domyślnie oczekuje na żądania w porcie 1433. Jeśli zmieniłeś tę wartość, podaj ją na pokazanej stronie.

Dla odbiornika grupy dostępności można wykorzystać protokół DHCP lub ustawić statyczny adres IP. Tu podany został statyczny adres IP.

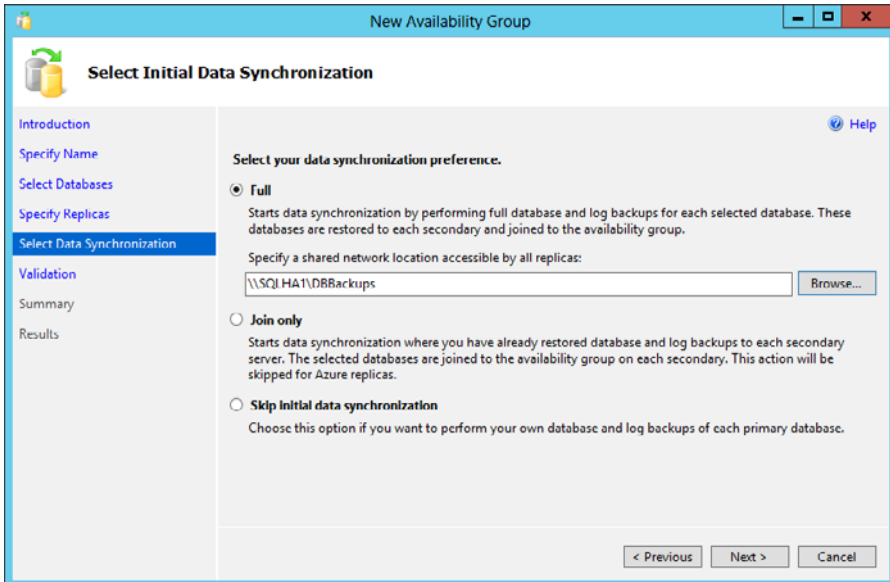
Jeśli chcesz, możesz pominąć ten krok i później utworzyć odbiornik grupy dostępności. W tym przykładzie używamy odbiornika AG1_Listener i portu 1433. Kliknij przycisk *Next*.
8. Na stronie *Select Initial Data Synchronization*, przedstawionej na rysunku 25.9, wybierz opcję *Full*. W polu *Specify a shared network location accessible by all replicas* ustaw katalog współużytkowany dostępny dla wszystkich replik (np. `\\SQLHA1\DBBackups`). Ponadto możesz wybrać opcję *Skip initial data synchronization* i ręcznie przywracać dane w każdej replice pomocniczej.



Rysunek 25.7. Ustawianie miejsca tworzenia kopii zapasowych

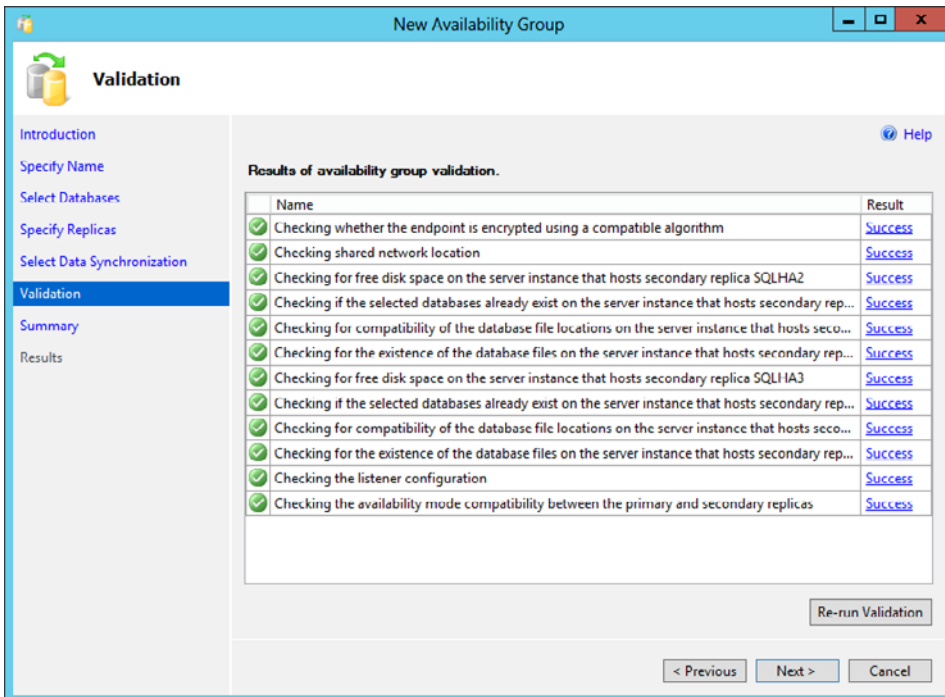


Rysunek 25.8. Konfigurowanie odbiornika



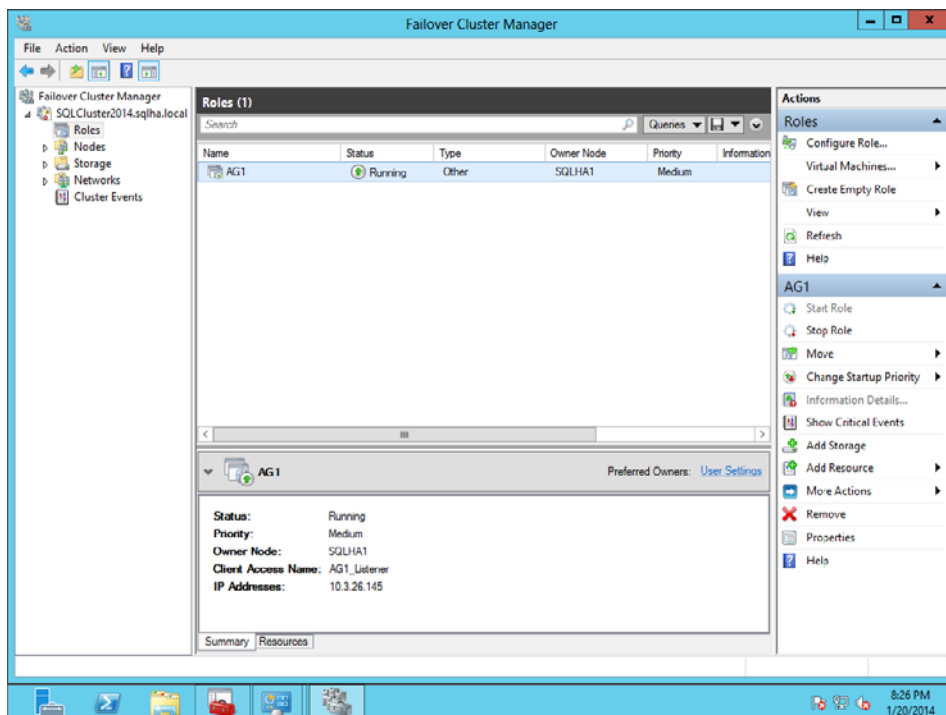
Rysunek 25.9. Konfigurowanie synchronizacji danych

9. Kliknij przycisk *Next*, aby przeprowadzić sprawdzanie poprawności, co przedstawiono na rysunku 25.10. Następnie ponownie kliknij przycisk *Next*. Na stronie *Summary* kliknij przycisk *Finish*, aby zakończyć tworzenie grupy dostępności AG1.



Rysunek 25.10. Sprawdzanie, czy grupa dostępności jest prawidłowo skonfigurowana

10. Po zakończeniu pracy otwórz narzędzie Failover Cluster Manager. Powinna być w nim widoczna utworzona grupa dostępności (patrz rysunek 25.11).



Rysunek 25.11. Nowa grupa dostępności w narzędziu Failover Cluster Manager

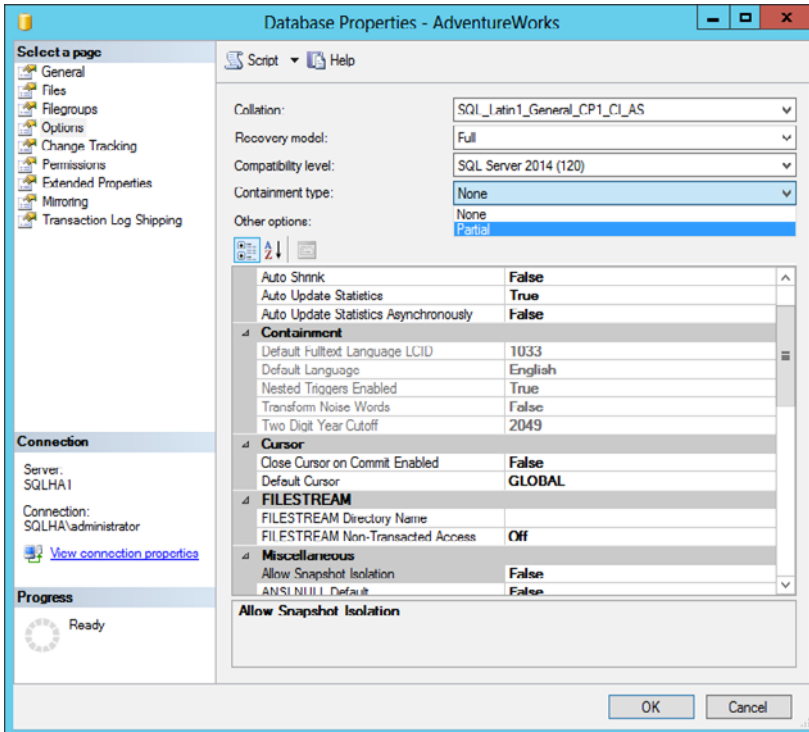
Gdy do repliki głównej dodawany jest login z systemu SQL Server, ten sam login musi znaleźć się w każdej replice pomocniczej, aby został uwierzytelniony po przełączeniu awaryjnym. Grupy dostępności nie obsługują baz systemowych, w których zapisane są loginy. Jednak w systemie SQL Server 2014 loginy można umieścić w bazach użytkowników, używając niezależnych baz danych (ang. *contained database*). W tym podejściu wszystkie utworzone loginy są synchronizowane w grupach dostępności jako dane. W częściowo niezależnych bazach danych (opcja *Partial*; w tym trybie działa baza z rysunku 25.12) powiązania między kontami użytkowników a loginami można skonfigurować i obsługiwać na poziomie baz danych, zamiast przy użyciu systemowych baz SQL Servera.

UWAGA Więcej informacji na temat niezależnych baz danych znajdziesz w rozdziale 4., „Zarządzanie silnikiem bazodanowym i rozwiązywanie związanych z nim problemów”.

Konfigurowanie istniejących grup dostępności

Teraz grupa dostępności jest już zainstalowana i można jej używać do zapewniania wysokiej dostępności, przywracania danych po katastrofie i generowania raportów. Jeśli wystąpi jeden z wymienionych poniżej warunków, mogą pojawić się nowe wymagania.

- Aplikacja zostanie zaktualizowana i utworzona będzie nowa baza zależna od innych baz wysokiej dostępności z grupy dostępności.



Rysunek 25.12. Ustawianie częściowo niezależnej bazy danych

- Firma zwiększy wymagania z obszaru generowania raportów i będzie musiała zainstalować dodatkową bazę tylko do odczytu, aby odciążyć replikę główną.
- Określona została lokalizacja sprzętu używanego do odzyskiwania danych po katastrofie. W tej lokalizacji należy zainstalować replikę.
- Kopie zapasowe są wykonywane w replice głównej i negatywnie wpływają na jej wydajność, dlatego trzeba zainstalować replikę używaną do tworzenia kopii zapasowych.
- Kilka baz danych z grupy dostępności zostało skonsolidowanych i trzeba usunąć jedną z nich.
- Replika ma zostać usunięta, ponieważ nie jest już potrzebna lub została zastąpiona inną repliką, działającą na szybszym sprzęcie.

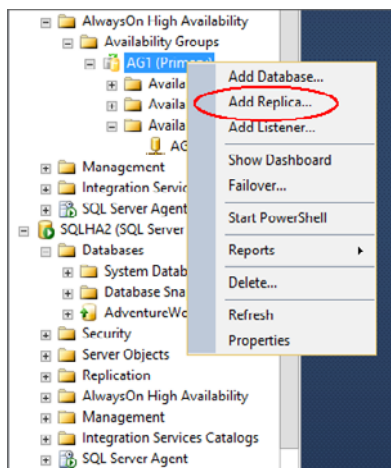
W takich sytuacjach grupy dostępności zapewniają dużą swobodę. Można zacząć od grupy obejmującej przynajmniej dwie repliki, a następnie zainstalować dodatkowe repliki i bazy wysokiej dostępności, gdy wymagania się zmieniają. W tym punkcie opisujemy, jak zainstalować dodatkowe repliki (do dziewięciu) lub dodatkowe bazy wysokiej dostępności. Gdy dane repliki i bazy wysokiej dostępności nie są już potrzebne w grupie dostępności, można je usunąć. W środowisku produkcyjnym operacje te należy wykonywać w czasie przeznaczonym na konserwację, a nie w trakcie zwykłej pracy systemu.

Dodawanie i usuwanie replik

Za każdym razem, gdy zechcesz dodać replikę, wykonaj następujące kroki.

1. Dodaj nową replikę do klastra w programie Failover Cluster Manager. Z rozdziału 16. „Klastrowanie w systemie SQL Server 2014” dowiesz się, jak to zrobić.

2. Włącz obsługę grup dostępności AlwaysOn w dodanym egzemplarzu systemu SQL Server (patrz rysunek 25.2).
3. W węźle *Availability Groups* w programie SQL Server Management Studio kliknij prawym przyciskiem myszy odpowiednią grupę dostępności (np. AG1) i wybierz opcję *Add Replica*, co przedstawiono na rysunku 25.13. Następnie za pomocą kreatora dodaj nową replikę.



Rysunek 25.13. Dodawanie repliki do istniejącej grupy dostępności

Aby usunąć replikę, kliknij ją prawym przyciskiem myszy i wybierz opcję *Delete*.

W celu usunięcia grupy dostępności kliknij ją prawym przyciskiem myszy i wybierz opcję *Delete*.

Dodawanie replik w platformie Azure

W systemie SQL Server 2014 kreator Add Azure Replica Wizard pozwala dodać lub utworzyć nową replikę w grupie dostępności jako maszynę wirtualną platformy IaaS (ang. *Infrastructure as a Service*) Windows Azure. Ta maszyna wirtualna jest automatycznie tworzona przez kreator z programu SQL Server Management Studio jako część hybrydowego środowiska informatycznego. Mechanizm ten umożliwia udostępnienie repliki w platformie Azure dla zadań wymagających tylko odczytu, tworzenia kopii zapasowych i odzyskiwania danych po katastrofie. Nie jest przy tym konieczne używanie lokalnie instalowanego sprzętu. Aby dodać maszynę wirtualną z repliką w platformie Azure, trzeba spełnić następujące warunki.

- Należy uruchomić kreator w egzemplarzu systemu, w którym działa obecna replika główna.
- Grupa dostępności musi obejmować zainstalowane lokalnie repliki wysokiej dostępności, a środowisko sieciowe musi mieć lokalnie działającą podsieć połączoną siecią VPN z platformą Windows Azure.
- Klienci używające odbiornika grupy dostępności muszą być podłączone do internetu, jeśli mają zachować połączenie z odbiornikiem po przełączeniu awaryjnym grupy dostępności na replikę z platformy Windows Azure.

Gdy te wymagania wstępne są spełnione, można uruchomić kreator Add Azure Replica Wizard w programie SQL Server Management Studio za pomocą opcji ze strony *Specify Replicas*. Strona jest dostępna w kreatorze New Availability Group i w kreatorze Add Replica to Availability Group. Po uruchomieniu kreatora Add Azure Replica Wizard wykonaj następujące kroki.

1. Kliknij przycisk *Download*, aby pobrać certyfikat odpowiadający subskrypcji platformy Windows Azure. Będzie on potrzebny do zarządzania repliką z maszyny wirtualnej z tej platformy.
2. Następnie musisz zalogować się do konta platformy Windows Azure, w którym chcesz utworzyć maszynę wirtualną z systemem SQL Server. Następnie kreator zainstaluje na lokalnym serwerze certyfikat potrzebny do zarządzania systemem. Jeśli pobierzesz kilka certyfikatów, możesz kliknąć przycisk ... i wybrać odpowiedni certyfikat.
3. Aby następnie nawiązać połączenie z kontem, kliknij przycisk *Connect*. Po nawiązaniu połączenia listy rozwijane w kreatorze zostaną wypełnione parametrami z platformy Windows Azure (m.in. *Image*, *VM Size*, *Virtual Network* i *Virtual Network Subnet*), opisanymi w tabeli 25.4.

Tabela 25.4. Opis parametrów z platformy Windows Azure

Parametr	Opis
Image	Wybierz z listy obraz z wersją systemu SQL Server dla platformy Windows Azure.
VM Size	Określ wielkość maszyny wirtualnej z platformy Windows Azure.
VM Name	Wybierz nazwę DNS dla maszyny wirtualnej z systemem SQL Server z platformy Windows Azure.
VM Username	Ustaw domyślną nazwę administratora maszyny wirtualnej z platformy Windows Azure.
VM Administrator Password (i Confirm Password)	Ustaw domyślne hasło administratora maszyny wirtualnej z platformy Windows Azure.
Virtual Network	Ustaw sieć wirtualną, która umożliwi maszynie wirtualnej z platformy Windows Azure dostęp do replik zainstalowanych lokalnie.
Virtual Network Subnet	Ustaw podsieć sieci wirtualnej, która umożliwi maszynie wirtualnej z platformy Windows Azure dostęp do replik zainstalowanych lokalnie.
Domain	Dodaj maszynę wirtualną z platformy Windows Azure do domeny usług Active Directory razem z innymi członkami grupy dostępności.
Domain Username	Podaj nazwę użytkownika z usługi Active Directory, który ma uprawnienia do dołączania maszyny wirtualnej z platformy Windows Azure do domeny.
Password	Podaj hasło z usługi Active Directory należące do użytkownika, który ma uprawnienia do dołączania maszyny wirtualnej z platformy Windows Azure do domeny.

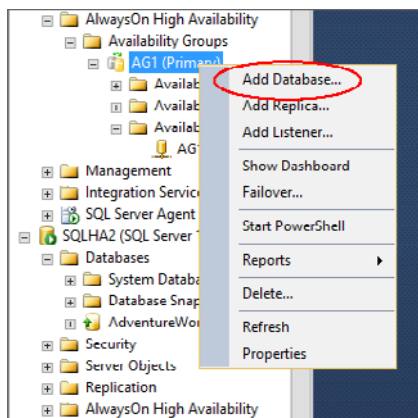
4. Skonfiguruj maszynę wirtualną z platformy Windows Azure. W tej maszynie ma działać nowa replika pomocnicza.
5. Następnie kliknij przycisk *OK*, aby zatwierdzić ustawienia i wyjść z kreatora Add Azure Replica Wizard.
6. Po powrocie na stronę *Specify Replicas* dokończ konfigurowanie nowej repliki.

Po zakończeniu pracy kreatora Availability Group Wizard lub Add Replica to Availability Group Wizard proces konfiguracji wykona wszystkie niezbędne operacje w platformie Windows Azure, aby utworzyć nową maszynę wirtualną, dołączyć ją do domeny usług Active Directory, dodać do klastra WSFC, włączyć obsługę grup dostępności AlwaysOn i dodać nową replikę z platformy Windows Azure do grupy dostępności.

Dodawanie i usuwanie baz danych

Za każdym razem, gdy zechcesz dodać bazę wysokiej dostępności, wykonaj następujące kroki.

1. W programie SQL Server Management Studio wybierz katalog *Availability Groups*, kliknij prawym przyciskiem myszy grupę dostępności (np. AG1) i wybierz opcję *Add Database*, co przedstawiono na rysunku 25.14.



Rysunek 25.14. Dodawanie baz do grup dostępności

2. Wykonaj operacje w kreatorze, aby dodać nową bazę.

UWAGA Tworzenie kopii zapasowych dużych baz danych i kopiowanie plików może negatywnie wpływać na wydajność środowiska produkcyjnego, dlatego operacje te warto wykonywać w czasie przeznaczonym na konserwację.

Aby usunąć bazę z grupy dostępności, kliknij tę bazę prawym przyciskiem myszy i wybierz opcję *Delete*.

Przełączanie awaryjne w grupach dostępności

Dostępne są dwa tryby przełączania awaryjnego — automatyczny i ręczny. Przebieg przełączania awaryjnego w grupach dostępności zależy od trybu ustawionego dla replik.

Przy automatycznym przełączaniu awaryjnym w trybie synchronicznym awaria repliki głównej prowadzi do przejęcia jej zadań przez replikę pomocniczą bez konieczności interwencji użytkownika. Aplikacje klienckie łączące się z odbiornikiem grupy dostępności (np. z odbiornikiem AG1_Listener) automatycznie łączą się z nową repliką główną, jeśli są skonfigurowane tak, aby próbować ponownie nawiązać połączenie po utraceniu.

Przy ręcznym przełączaniu awaryjnym w trybie synchronicznym przełączenie się do repliki pomocniczej wymaga ręcznej interwencji. Aby ręcznie przełączyć repliki, wykonaj w programie SQL Server Management Studio następujące kroki.

1. Nawiąż połączenie z egzemplarzem systemu SQL Server zawierającym replikę pomocniczą, która ma stać się repliką główną, i rozwiń drzewo serwera.
2. Rozwiń katalogi *AlwaysOn High Availability* i *Availability Groups*.
3. Kliknij prawym przyciskiem myszy grupę dostępności, w której chcesz przełączyć repliki, i wybierz opcję *Failover*.

Przełączenie awaryjne można też wykonać za pomocą kodu w języku T-SQL. Poniższy fragment kodu powoduje przełączenie do lokalnej repliki pomocniczej w grupie dostępności AG1:

```
ALTER AVAILABILITY GROUP AG1 FAILOVER;
```

W celu wymuszenia przełączenia awaryjnego z możliwą utratą danych wywołaj poniższą instrukcję w języku T-SQL:

```
ALTER AVAILABILITY GROUP AG1 FORCE_FAILOVER_ALLOW_DATA_LOSS;
```

Ponadto możesz przełączyć awaryjnie repliki w powłoce PowerShell systemu Windows. Poniższy przykład ilustruje wymuszone przełączenie awaryjne (z możliwą utratą danych) w grupie dostępności AG1. Przełączenie następuje do repliki pomocniczej z danego egzemplarza systemu SQL Server:

```
Switch-SqlAvailabilityGroup -Path
SQLSERVER:\Sql\SecondaryServer\InstanceName\AvailabilityGroups\
AG1 -AllowDataLoss
```

Po wymuszonym przełączeniu awaryjnym docelowa replika pomocnicza staje się nową repliką główną, a wszystkie pomocnicze bazy danych są zawieszane. Aby wznowić pracę zawieszonych baz danych, trzeba zmienić konfigurację quorum klastra WSFC i dostosować tryb dostępności w grupie. Przyjrzyj się następującym scenariuszom.

- Jeśli przełączenie awaryjne odbyło się poza zestawem replik uwzględnianym przy przełączaniu automatycznym w grupie dostępności, dostosuj głosy w quorum odpowiadające węzłom z klastra WSFC, aby odzwierciedlić nową konfigurację grupy dostępności (ustawianie quorum jest opisane w rozdziale 16., „Klastrowanie w systemie SQL Server 2014”).
- Jeżeli przełączenie awaryjne odbyło się poza zestawem replik z ustawionym zatwierdzaniem synchronicznym, rozważ zmianę trybu dostępności i trybu przełączania awaryjnego w nowej replce głównej oraz w pozostałych replikach pomocniczych. Ustaw w nich pożądaną konfigurację zatwierdzania synchronicznego i automatycznego przełączania awaryjnego.

W obu sytuacjach musisz ręcznie wznowić pracę każdej zawieszonej bazy danych. Po wznowieniu działania pomocnicze bazy inicjują synchronizację danych z powiązaną bazą główną. Gdy wcześniejsza replika główna stanie się dostępna, przyjmie funkcje repliki pomocniczej i od razu zawiesi pracę swoich baz danych (teraz już pomocniczych). W trybie z zatwierdzaniem asynchronicznym dla tych baz danych mogło się zebrać wiele niewysłanych rekordów dziennika. Wznowienie pracy nowej bazy pomocniczej powoduje odrzucenie niewysłanych rekordów dziennika i wycofanie zmian, które nie zostały wprowadzone w obecnej replce głównej.

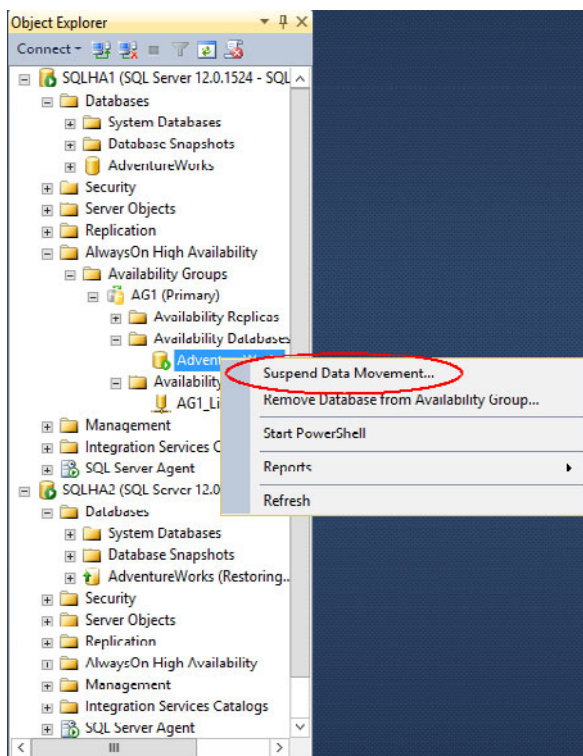
Jeśli replika wysokiej dostępności, w której nastąpiła awaria, nie wznowi pracy lub robi to zbyt późno, aby można było ciągle odraczać przycinanie dziennika transakcji nowej bazy głównej, pomyśl o usunięciu problematycznej repliki z grupy dostępności.

Zawieszanie pracy baz wysokiej dostępności

Gdy następuje spadek wydajności, możesz zdecydować się zawiesić pracę pomocniczych baz wysokiej dostępności. Do zawieszonych baz nie są przesyłane rekordy dziennika transakcji. W tym czasie dziennik transakcji w replce głównej staje się coraz większy i nie można go przyciąć. Jeśli zawieszysz pomocniczą bazę z pomocniczej repliki, zawieszona zostanie tylko ta lokalna baza. Jeżeli zawieszysz bazę w replce głównej, rekordy z dziennika transakcji nie będą przesyłane do żadnej bazy z replik pomocniczych. Zawieszenie pomocniczej bazy powoduje ustawienie jej stanu na **SUSPENDED**. Taka baza staje się nieaktualna względem głównej bazy danych. Główna baza wciąż jest dostępna, przy czym jeśli używana jest tylko jedna replika pomocnicza i zostanie ona zawieszona, dane z repliki głównej będą narażone.

W celu zawieszenia bazy za pomocą programu SQL Server Management Studio wykonaj następujące kroki.

1. W programie SQL Server Management Studio nawiąż połączenie z egzemplarzem systemu SQL Server, w którym działa replika wysokiej dostępności z bazą przeznaczoną do zawieszenia.
2. Rozwiń katalogi *AlwaysOn High Availability* i *Availability Groups*.
3. Rozwiń katalog *Availability Databases*, kliknij prawym przyciskiem myszy zawieszaną bazę i wybierz opcję *Suspend Data Movement*, co pokazano na rysunku 25.15.



Rysunek 25.15. Zawieszanie pracy bazy danych

4. W oknie dialogowym *Suspend Data Movement* kliknij przycisk **OK**.

Aby zawiesić pracę bazy za pomocą kodu w języku T-SQL, wykonaj następujące kroki.

1. Nawiąż połączenie z egzemplarzem systemu SQL Server, w którym działa replika z bazą danych przeznaczoną do zawieszenia.
2. Zawieś pomocniczą bazę danych za pomocą następującej instrukcji ALTER DATABASE:
ALTER DATABASE nazwa_bazy_danych SET HADR SUSPEND;

Ponadto możesz zawiesić bazę wysokiej dostępności przy użyciu polecenia w powłoce PowerShell systemu Windows.

1. Za pomocą instrukcji cd przejdź do egzemplarza systemu SQL Server, w którym działa replika z bazą danych przeznaczoną do zawieszenia.
2. Za pomocą cmdletu Suspend-SqlAvailabilityDatabase zawieś wybraną bazę z grupy dostępności.

Wznawianie pracy baz wysokiej dostępności

Gdy baza danych była zawieszona, to po wznowieniu pracy przez replikę jej początkowy stan to SYNCHRONIZING. Utrzymuje się on do momentu zsynchronizowania danych. Główna baza danych ostatecznie wznowia pracę wszystkich powiązanych baz pomocniczych, które zostały zawieszone w wyniku zawieszenia bazy głównej.

Aby wznowić pracę bazy w programie SQL Server Management Studio, wykonaj następujące kroki.

1. W programie SQL Server Management Studio nawiąż połączenie z egzemplarzem systemu SQL Server zawierającym replikę z bazą, która ma wznowić działanie.
2. Rozwiń katalogi *AlwaysOn High Availability* i *Availability Groups*.
3. Rozwiń katalog *Availability Databases*, kliknij odpowiednią bazę prawym przyciskiem myszy i wybierz opcję *Resume Data Movement*.
4. W oknie dialogowym *Resume Data Movement* kliknij przycisk OK.

W celu wznowienia pracy bazy za pomocą kodu w języku T-SQL wykonaj następujące kroki.

1. Nawiąż połączenie z egzemplarzem systemu SQL Server zawierającym bazę, której pracę chcesz wznowić.
2. Wznów pracę pomocniczej bazy danych za pomocą następującej instrukcji ALTER DATABASE:

```
ALTER DATABASE nazwa_bazy_danych SET HADR RESUME;
```

Ponadto możesz wznowić pracę baz wysokiej dostępności przy użyciu instrukcji powłoki PowerShell systemu Windows.

1. Za pomocą polecenia cd przejdź do egzemplarza systemu SQL Server zawierającego replikę z bazą, której pracę chcesz wznowić.
2. Użyj cmdletu Resume-SqlAvailabilityDatabase, aby wznowić pracę bazy w grupie dostępności.

Połączenia aplikacji klienckich

Do obsługi przełączania awaryjnego w aplikacjach klienckich łączących się z repliką główną można stosować dwa rozwiązania — nazwę sieciową ustawioną dla odbiornika grupy dostępności lub łańcuchy znaków połączenia z mechanizmu tworzenia kopii lustrzanych.

Gdy budujesz i konfigurujesz grupę dostępności, dodajesz też odbiornik, co powoduje ustawienie nazwy sieciowej i adresu IP w klastrze WSFC (patrz rysunek 25.8). Gdy w wyniku awarii replika główna stanie się niedostępna, replika pomocnicza stanie się nową repliką główną. W łańcuchu znaków połączenia używana jest nazwa sieciowa, która pozwala automatycznie przekierowywać aplikacje klienckie do nowej repliki głównej.

Aby umożliwić klientom łączenie się z repliką główną danej grupy dostępności, możesz wykorzystać dowolną z dwóch wspomnianych metod. Szczegółowo omawiamy je w dwóch następnych podpunktach.

Nazwa sieciowa odbiornika grupy dostępności

Jeśli przy przełączaniu awaryjnym replik w grupie dostępności aplikacje klienckie używają w łańcuchach znaków połączenia nazwy sieciowej, nazwa ta pozwala nawiązać połączenie z nową repliką główną. Dla każdej grupy dostępności trzeba utworzyć unikatową w domenie nazwę sieciową. Przy tworzeniu nazwy sieciowej przypisywany jest do niej adres IP. Tylko protokół TCP umożliwia stosowanie nazwy sieciowej i opcjonalnie adresu IP przy łączeniu się z grupą dostępności. W łańcuchach znaków połączenia używanych w aplikacjach klienckich do łączenia się z bazami z grupy dostępności należy stosować nazwę sieciową zamiast nazwy serwera. Dzięki temu aplikacje mogą łączyć się bezpośrednio z obecną repliką główną. Oto przykładowe łańcuchy znaków połączenia używane w aplikacjach klienckich:

```
Server=tcp:MynetworkName;Database=AdventureWorks;IntegratedSecurity=SSPI
Server=tcp:MynetworkName,1433;Database=AdventureWorks;IntegratedSecurity=SSPI
```

Łańcuchy znaków połączenia z mechanizmu tworzenia kopii lustrzanych

Załóżmy, że chcesz zastąpić mechanizm tworzenia kopii lustrzanych grupami dostępności. Jeśli grupa dostępności zawiera tylko dwie repliki i nie umożliwia dostępu w trybie odczytu do repliki pomocniczej, aplikacje klienckie mogą łączyć się z repliką główną przy użyciu łańcuchów znaków połączenia z mechanizmu tworzenia kopii lustrzanych.

To podejście jest przydatne, gdy istniejące aplikacje klienckie używające systemu z tworzeniem kopii lustrzanych baz danych mają zacząć korzystać z grup dostępności (przy czym grupa dostępności musi wtedy obejmować tylko dwie repliki). Zanim w tym podejściu dodasz nowe repliki, musisz utworzyć nazwę sieciową odbiornika i ustawić ją w aplikacjach klienckich.

Gdy używasz łańcuchów znaków z mechanizmu tworzenia kopii lustrzanych, klienci mogą korzystać z dostawców danych SQL Server Native Client lub .NET Framework Data Provider for SQL Server. Łańcuch znaków połączenia podawany przez aplikację kliencką musi obejmować przynajmniej nazwę jednego egzemplarza systemu SQL Server (*początkowego partnera głównego*), co pozwala zidentyfikować egzemplarz początkowo przechowujący główną replikę wysokiej dostępności, z którą aplikacja ma się łączyć. Opcjonalnie w łańcuchu znaków połączenia można też podać nazwę repliki pomocniczej (czyli nazwę partnera rezerwowego), co pozwala określić egzemplarz początkowo przechowujący replikę pomocniczą. Oto przykład:

```
"Data Source=ServerA;Failover Partner=ServerB;Initial Catalog=AdventureWorks;
Integrated Security=True;"
```

Aktywna replika pomocnicza udostępniana w trybie tylko do odczytu

W tym podrozdziale omawiamy oferowane w grupach dostępności funkcje replik pomocniczych związane z wykonywaniem raportów i obsługą kwerend wymagających operacji tylko do odczytu. Grupy dostępności umożliwiają tworzenie replik pomocniczych udostępnianych w trybie tylko do odczytu. Repliki pomocnicze mają przede wszystkim zapewniać wysoką dostępność i umożliwiać odzyskiwanie danych po katastrofie, jednak można też tak je skonfigurować, aby obsługiwały używane do generowania raportów kwerendy tylko do odczytu, co pozwala odciążyć replikę główną. Wtedy replika główna może lepiej wykonywać zadania o znaczeniu strategicznym, a raporty wymagające tylko odczytu danych są generowane za pomocą repliki pomocniczej.

Nowością w systemie SQL Server 2014 jest to, że z replik pomocniczych udostępnianych do odczytu można korzystać także po ich odłączeniu od repliki głównej i po utracie kworum w klastrze. Replika pomocnicza udostępniana w trybie tylko do odczytu jest aktualizowana niemal w czasie rzeczywistym. Jej opóźnienie względem repliki głównej jest mierzone w sekundach. Wynika ono z czasu potrzebnego na synchronizację dzienników transakcji i na zaktualizowanie danych przez wątki wprowadzające rekordy z tych dzienników w każdej replice pomocniczej. Dopiero po zakończeniu tej aktualizacji kwerendy mogą pobierać dane.

Dostęp w trybie tylko do odczytu

Gdy umożliwisz dostęp do repliki pomocniczej w trybie tylko do odczytu, z wszystkich baz z grupy dostępności będzie można korzystać w ten sposób. Ustawienie, które określa sposób dostępu do replik pomocniczych, jest widoczne na rysunku 25.6 w kolumnie *Readable Secondary*. Dostęp tylko do odczytu nie oznacza, że pomocnicze bazy danych działają w trybie tylko do odczytu. Chodzi o to, że połączenia użytkowników z bazą pomocniczą mają dostęp do danych w takim trybie. Choć nie można zapisywać danych w bazach pomocniczych, możliwy jest zapis w bazach spoza grupy dostępności, w tym w bazach systemowych (takich jak tempdb). Dany egzemplarz systemu SQL Server może jednocześnie obejmować kilka replik wysokiej dostępności i bazy danych spoza grup dostępności.

UWAGA Opis opcji z kolumny *Readable Secondary* ustawianych dla replik pomocniczych znajdziesz w tabeli 25.3.

W tabeli 25.5 pokazano, że początkowa replika główna, SQLHA1, obsługuje połączenia w trybie odczytu i zapisu, natomiast replika pomocnicza, SQLHA2, obsługuje wszystkie bezpośrednie połączenia, w tym połączenia bez ustawionej właściwości *ApplicationIntent*. W momencie przełączenia awaryjnego repliki zamieniają się rolami. Replika SQLHA2, przejmująca rolę główną, obsługuje wtedy połączenia w trybie odczytu i zapisu, natomiast replika SQLHA1, ustawiana jako pomocnicza, odrzuca wszystkie połączenia bezpośrednie.

Tabela 25.5. Ustawienia opcji *Readable Secondary*

Nazwa repliki	Początkowa rola	Opcja <i>Readable Secondary</i>
SQLHA1	Główna	No
SQLHA2	Pomocnicza	Yes

Łączenie się klientów z repliką pomocniczą

W momencie przełączenia awaryjnego w grupie dostępności istniejące połączenia z grupą są zrywane, po czym aplikacje klienckie muszą ponownie nawiązać połączenie. Repliki zmieniają wtedy role — replika pomocnicza staje się repliką główną. W tym momencie inna replika pomocnicza powinna zostać udostępniona tylko do odczytu na potrzeby generowania raportów. Aby było to możliwe, aplikacje klienckie generujące raporty muszą mieć możliwość połączenia się z następną repliką pomocniczą udostępnianą w trybie tylko do odczytu. Łączenie się z repliką pomocniczą udostępnianą w trybie tylko do odczytu odbywa się na dwa sposoby — można bezpośrednio podać nazwę egzemplarza systemu SQL Server w łańcuchu znaków połączenia lub użyć nazwy odbiornika grupy dostępności i wykorzystać przekierowywanie operacji odczytu, aby nawiązać połączenie z następną repliką pomocniczą udostępnianą tylko do odczytu.

Przekierowywanie operacji odczytu polega na tym, że połączenia do odbiornika grupy dostępności trafiają do repliki pomocniczej udostępnianej dla zadań wymagających samego odczytu.

Technika działa tylko wtedy, gdy w łańcuchu znaków połączenia podany jest odbiornik grupy dostępności. Przychodzące połączenia z nazwą tego odbiornika mogą być automatycznie kierowane do repliki udostępnianej tylko do odczytu, jeśli spełnione są następujące warunki.

- Właściwość `ApplicationIntent` w przychodzącym połączeniu jest ustawiona na wartość `ReadOnly`.
- W replice pomocniczej ustawiona jest obsługa połączeń wymagających tylko odczytu.
- Za pomocą polecenia `CREATE` lub `ALTER AVAILABILITY GROUP` języka T-SQL została ustawiona opcja `READ_ONLY_ROUTING_URL` każdej repliki (w ramach ustawiania opcji repliki o roli `SECONDARY_ROLE`). Przed przejściem do konfigurowania listy przekierowań trzeba ustawić tę opcję.
- Za pomocą polecenia `CREATE AVAILABILITY GROUP` lub `ALTER AVAILABILITY GROUP` języka T-SQL została ustawiona opcja `READ_ONLY_ROUTING_LIST` każdej repliki (w ramach ustawiania opcji repliki o roli `PRIMARY_ROLE`).
- W opcji `READ_ONLY_ROUTING_LIST` można podać jedną lub więcej docelowych replik, do których połączenia mają być kierowane. Można skonfigurować wiele docelowych replik, a przekierowywanie będzie się odbywać zgodnie z ich kolejnością na liście.

Poniższe przykłady ilustrują modyfikowanie istniejącej grupy dostępności do obsługi przekierowywania połączeń wymagających tylko odczytu:

```
ALTER AVAILABILITY GROUP [AG1]
    MODIFY REPLICA ON
N'SQLHA1' WITH
(SECONDARY_ROLE (READ_ONLY_ROUTING_URL = N'TCP://SQLHA1. contoso.com:1433'));
ALTER AVAILABILITY GROUP [AG1]
    MODIFY REPLICA ON
N'SQLHA2' WITH
(SECONDARY_ROLE (READ_ONLY_ROUTING_URL = N'TCP://SQLHA2. contoso.com:1433'));
ALTER AVAILABILITY GROUP [AG1]
    MODIFY REPLICA ON
N'SQLHA1' WITH
(PRIMARY_ROLE (READ_ONLY_ROUTING_LIST=('SQLHA2','SQLHA1')));
```

Następnie na podstawie wartości parametru `ApplicationIntent` (`ReadWrite` lub `ReadOnly`) ustawionej w łańcuchu znaków połączenia aplikacja kliencka będzie przekierowywana do replik umożliwiających zapis i odczyt lub do replik tylko do odczytu. W tym przykładzie parametr `ApplicationIntent` ma wartość `ReadOnly`, dlatego połączenie jest kierowane do repliki pomocniczej przeznaczonej tylko do odczytu:


```
Server=tcp:AGListener,1433;Database=AdventureWorks;
IntegratedSecurity=SSPI;ApplicationIntent=ReadOnly
```

Wydajność

Aby zapewnić współbieżność i zapobiec blokowaniu się przez operacje odczytu i zapisu, w bazach z replik pomocniczych domyślnie włączona jest izolacja snapshotów. Gdy udostępniane są repliki pomocnicze działające w trybie tylko do odczytu, w głównych bazach danych dodawane jest 14 bajtów dla usuwanych, modyfikowanych i wstawianych wierszy, co pozwala zachować wskaźniki do wersji wierszy w bazach z repliki pomocniczej. Struktura określająca wersje wierszy jest kopiowana do pomocniczych baz danych. Zapewnia to możliwość sprawdzenia wersji wierszy potrzebnych dla operacji wymagających samego odczytu. Kontrola wersji wierszy zwiększa ilość miejsca zajmowanego w replikach głównych i pomocniczych. Ponadto może nastąpić podział stron, ponieważ wiersze zawierają po 14 dodatkowych bajtów.

Dla zadań wymagających tylko odczytu często potrzebne są indeksy obsługujące kwerendy z samym odczytem. Ponieważ w bazach pomocniczych przeznaczonych tylko do odczytu nie można tworzyć nowych indeksów, trzeba to zrobić w replice głównej i przenieść indeksy do replik pomocniczych.

Statystyki dotyczące kolumn tabel i widoków indeksowanych są przenoszone z repliki głównej do replik pomocniczych w wyniku przesyłania rekordów z dzienników transakcji. W zadaniach wymagających odczytu w replikach pomocniczych potrzebne mogą być dodatkowe statystyki. Dlatego system SQL Server może w każdej replice pomocniczej generować statystyki tymczasowe zapisywane w bazie tempdb. Jeśli chcesz sam tworzyć statystyki, musisz zrobić to w replice głównej. W trakcie tworzenia statystyk pamiętaj o następujących kwestiach.

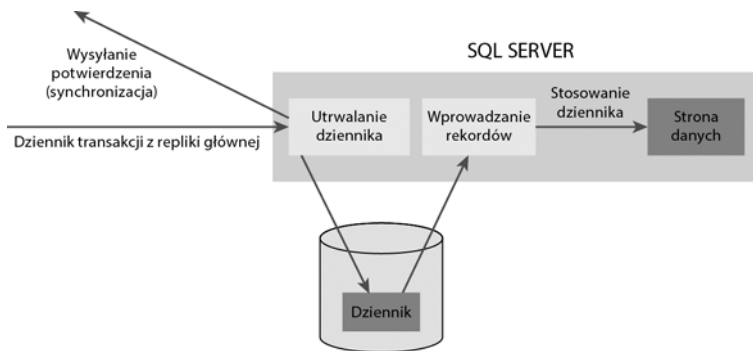
- Do usuwania tymczasowych statystyk służy instrukcja `DROP STATISTICS` języka T-SQL.
- Do monitorowania statystyk służą widoki `sys.stats` i `sys.stats_columns` z katalogu. Widok `sys.stats` zawiera kolumnę o nazwie `is_temporary`, określającą, czy statystyki są trwałe, czy tymczasowe.
- Statystyki tymczasowe są automatycznie usuwane w momencie ponownego uruchomienia repliki pomocniczej (ponieważ baza tempdb jest wtedy zerowana) lub przełączenia awaryjnego i zmiany ról replik.

Repliki pomocnicze przeznaczone tylko do odczytu zwykle są nieaktualne względem repliki głównej (najczęściej mają opóźnienie mierzone w sekundach). Wpływają na to następujące czynniki.

- Czas trwania transakcji w bazie danych.
- Obciążenie repliki pomocniczej. Gdy replika jest obciążona obsługą zadań wymagających odczytu, przenoszenie rekordów z dzienników transakcji może zostać spowolnione.
- Czas transferu danych w sieci między replikami główną a pomocniczą.
- Gdy zadania wymagają wielu operacji wejścia-wyjścia (np. przy przenoszeniu danych do replik pomocniczych), może nastąpić kopiowanie indeksu lub kopiowanie masowe.

Ponadto kwerendy kierowane do repliki pomocniczej „widzą” dane z repliki głównej dopiero po tym, jak wątek wprowadzający przesłane rekordy wprowadzi te dane w replice pomocniczej. Oto kroki potrzebne do tego, aby dane trafiły do kwerendy w replice pomocniczej (patrz rysunek 25.16).

1. Dane docierają do repliki pomocniczej, gdzie zostają utrwalone w dzienniku transakcji.
2. Wysyłane jest potwierdzenie do repliki głównej.



Rysunek 25.16. Modyfikowanie danych w replice pomocniczej

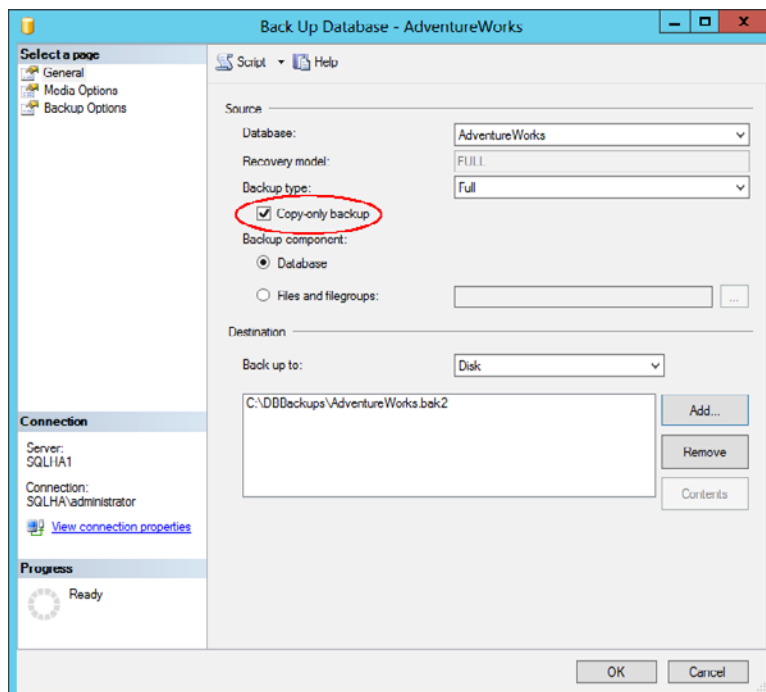
3. Wątek wprowadzający przesłane rekordy wprowadza zmiany z dziennika transakcji na stronach danych.
4. Kwerenda skierowana do repliki pomocniczej ma dostęp do nowych danych.

Tworzenie kopii zapasowych w replice pomocniczej

Jedną z funkcji grup dostępności w systemie SQL Server 2014 jest możliwość wykonywania kopii zapasowych w dowolnej replice wysokiej dostępności — głównej lub pomocniczej. Dlatego możesz przenieść tworzenie kopii zapasowych (co obciąża procesor i wymaga wielu operacji wejścia-wyjścia) z repliki głównej do repliki pomocniczej. Więcej informacji o udostępnianiu replik pomocniczych znajdziesz w punkcie „Umożliwianie dostępu do replik pomocniczych w trybie tylko do odczytu”, wcześniej w tym rozdziale. Dzięki przeniesieniu procesu tworzenia kopii zapasowych do repliki pomocniczej, w replice głównej można bez zakłóceń wykonywać zadania o znaczeniu strategicznym, niezbędne do działania firmy. Ponadto kopie zapasowe można zapisywać w replikach działających synchronicznie lub asynchronicznie. Kopie zapasowe dzienników transakcji z replik można łączyć, aby uzyskać jeden łańcuch dzienników transakcji.

Na rysunku 25.17 pokazano, że w replikach pomocniczych można tworzyć pełne kopie zapasowe przeznaczone tylko do kopiowania. Mogą to być kopie baz danych, plików i grup plików. Oto cechy takich kopii.

- *Pełna kopia zapasowa tylko do kopiowania* to kopia zapasowa z systemu SQL Server niezależna od sekwencji standardowych kopii zapasowych. Nie wpływa ona na procedury tworzenia i przywracania kopii zapasowych bazy danych.
- Pełne kopie zapasowe tylko do kopiowania są obsługiwane we wszystkich trybach odzyskiwania danych.
- Pełne kopie zapasowe tylko do kopiowania nie mogą być używane jako podstawa do tworzenia kopii różnicowych ani jako same kopie różnicowe. Nie wpływają też na proces wykonywania podstawy dla kopii różnicowych. W replikach pomocniczych różnicowe kopie zapasowe nie są obsługiwane.
- Kopię zapasową w replice pomocniczej można wykonać tylko wtedy, gdy replika pomocnicza może komunikować się z repliką główną i jest z nią zsynchronizowana (lub synchronizacja jest w toku).



Rysunek 25.17. Konfigurowanie tworzenia kopii zapasowych w replice pomocniczej

Aby określić, gdzie mogą działać zadania wykonujące kopie zapasowe, ustaw w replice głównej lub pomocniczej następujące opcje w instrukcjach `CREATE AVAILABILITY GROUP` lub `ALTER AVAILABILITY GROUP` języka T-SQL.

- Ustaw opcję `AUTOMATED_BACKUP_PREFERENCE`, aby określić, czy kopie zapasowe mają być wykonywane zgodnie z ustawieniami z rysunku 25.7.
- System SQL Server nie wymusza przestrzegania opcji `AUTOMATED_BACKUP_PREFERENCE`. Trzeba sprawdzić zwróconą wartość i uwzględnić ją w logice zadania wykonującego kopie zapasowe, aby uzyskać oczekiwany efekt.
- Jeśli nie chcesz wykonywać kopii zapasowych w którejś z replik pomocniczych, możesz ustawić opcję `BACKUP_PRIORITY = 0`. Wtedy w tej replice nie będą tworzone kopie zapasowe.
- Na poziomie repliki możesz określić kolejność replik pomocniczych preferowanych przy wykonywaniu kopii zapasowych. W tym celu ustaw opcję `BACKUP_PRIORITY` na wartość z przedziału od 1 do 100 (100 oznacza najwyższy priorytet).

Analizowanie metadanych z replik, w których wykonywane są kopie zapasowe

Czy chcesz wiedzieć, gdzie wykonywane są kopie zapasowe? Umożliwi Ci to sprawdzenie, czy dana replika ma wystarczającą ilość wolnego miejsca i wydajność, aby tworzyć kopie zapasowe baz danych. Jeśli chcesz ustalić preferowaną lokalizację wykonywania kopii zapasowych baz danych, możesz to łatwo zrobić za pomocą poniższej kwerendy:

```
SELECT automated_backup_preference, automated_backup_preference_desc
FROM sys.availability_groups;
```

W tabeli 25.6 znajdziesz opis kolumn `automated_backup_preference` i `automated_backup_preference_desc`, co pomoże Ci zinterpretować zwracane przez tę kwerendę wartości.

Tabela 25.6. Wartości kolumn `automated_backup_preference` i `automated_backup_preference_desc`

Kolumna	Wartość
<code>automated_backup_preference</code>	0 = preferowane jest wykonywanie kopii zapasowych w replice głównej. 1 = preferowane jest wykonywanie kopii zapasowych w replice pomocniczej. 2 = preferowane jest wykonywanie kopii zapasowych w replice pomocniczej, jeśli jednak żadna taka replika nie jest dostępna, dozwolone jest tworzenie kopii zapasowych w replice głównej. 3 = nie ma preferencji dotyczących wykonywania kopii zapasowych w replikach głównej lub pomocniczej.
<code>automated_backup_preference_desc</code>	PRIMARY SECONDARY_ONLY SECONDARY NONE

Możesz łatwo określić priorytet (w zakresie wykonywania kopii zapasowych) danej repliki wysokiej dostępności względem innych replik. W tym celu wywołaj poniższą kwerendę. Pole `backup_priority` reprezentuje priorytet danej repliki względem innych replik z tej samej grupy dostępności. Wartość jest podawana jako liczba całkowita z przedziału od 0 do 100.

```
SELECT backup_priority FROM sys.availability_replicas;
```

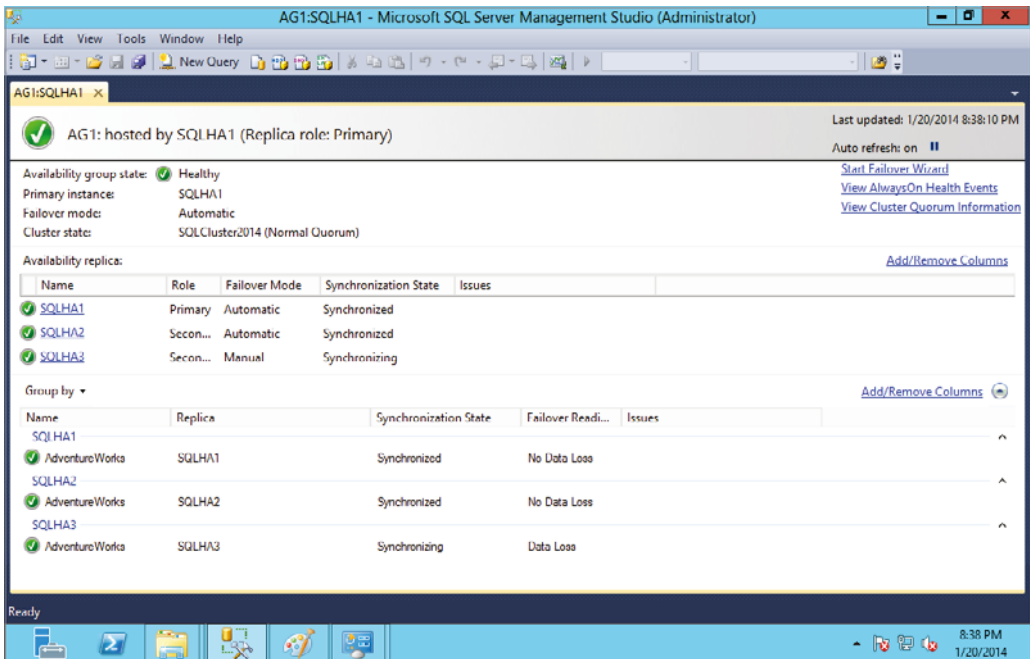
Panel kontrolny grup dostępności AlwaysOn

Aby monitorować grupy dostępności AlwaysOn, repliki wysokiej dostępności i bazy wysokiej dostępności w systemie SQL Server 2014, możesz wykorzystać panel kontrolny lub widoki DMV. W panelu kontrolnym możesz określić stan i wydajność każdej grupy dostępności. Aby otworzyć ten panel kontrolny, wykonaj następujące kroki.

1. W programie SQL Server Management Studio nawiąż połączenie z egzemplarzem systemu SQL Server, w którym chcesz uruchomić panel kontrolny grupy dostępności. Możesz tu wybrać replikę główną lub pomocniczą.
2. Rozwiń katalogi *AlwaysOn High Availability* i *Availability Groups*.
3. Kliknij prawym przyciskiem myszy nazwę grupy dostępności (tu jest to AG1), a następnie wybierz opcję *Show Dashboard*. Na rysunku 25.18 przedstawiono otwarty panel kontrolny.

W tabeli 25.7 opisane są najważniejsze elementy panelu kontrolnego.

Ponadto zamiast używać panelu kontrolnego grupy dostępności AlwaysOn, możesz pobrać opisane informacje za pomocą widoku DMV `sys.dm_hadr_availability_replica_states`.



Rysunek 25.18. Panel kontrolny grupy dostępności

Tabela 25.7. Ważne elementy panelu kontrolnego

Element	Opis
<i>Availability Group State</i>	Wyświetla stan grupy dostępności.
<i>Primary Instance</i>	Wyświetla nazwę egzemplarza systemu SQL Server obejmującego replikę główną z grupy dostępności.
<i>Failover Mode</i>	Wyświetla tryb przełączania awaryjnego ustawiony dla repliki. Oto możliwe wartości. Automatic — określa, że dwie repliki działają w trybie automatycznego przełączania awaryjnego. Manual — informuje, że żadna replika nie działa w trybie automatycznego przełączania awaryjnego.
<i>Role</i>	Obecna rola danej repliki dostępności. Używane wartości to <i>Primary</i> i <i>Secondary</i> .
<i>Synchronization State</i>	Określa, czy replika pomocnicza jest aktualnie zsynchronizowana z repliką główną. Oto używane wartości. Not Synchronized . Baza danych nie jest zsynchronizowana lub nie została jeszcze dodana do grupy dostępności. Synchronized . Baza jest zsynchronizowana z bazą główną z obecnej repliki głównej (jeśli ta jest dostępna) lub z ostatniej repliki głównej. NULL (Unknown) . Ta wartość pojawia się, gdy lokalny egzemplarz systemu SQL Server nie może skomunikować się z klastrem WSFC (oznacza to, że lokalny węzeł nie uczestniczy w kworum klastra WSFC). Synchronizing . Baza jest synchronizowana z bazą główną z obecnej repliki głównej (jeśli ta jest dostępna) lub z ostatniej repliki głównej.

Monitorowanie i rozwiązywanie problemów

Podobnie jak w każdym rozwiązaniu, musisz regularnie monitorować grupę dostępności, co pozwoli wykryć ewentualne problemy, które mogą uniemożliwić synchronizację danych w replikach pomocniczych, lub zauważyć nieudane przełączenie awaryjne. Jeśli chcesz monitorować grupy dostępności, repliki wysokiej dostępności i bazy wysokiej dostępności za pomocą programu SQL Server Management Studio, wykonaj następujące kroki.

1. W programie SQL Server Management Studio nawiąż połączenie z egzemplarzem systemu SQL Server, w którym chcesz monitorować grupę dostępności, i kliknij nazwę tego egzemplarza.
2. Rozwiń katalogi *AlwaysOn High Availability* i *Availability Groups*.
3. W panelu *Details* okna *Object Explorer* pojawi się lista wszystkich grup dostępności, z których repliki znajdują się w danym egzemplarzu systemu SQL Server. W kolumnie *Server Instance (Primary)* dla każdej grupy podana jest nazwa egzemplarza systemu zawierającego replikę główną. Aby wyświetlić więcej informacji na temat danej grupy dostępności, zaznacz ją w oknie *Object Explorer*. W panelu *Details* pojawią się wtedy repliki i bazy wysokiej dostępności z wybranej grupy.

Do szczegółowego monitorowania i rozwiązywania problemów użyj widoków DMV podanych w tabeli 25.8. Pomagają one w identyfikowaniu i rozwiązywaniu problemów z grupami dostępności oraz kontrolowaniu takich grup.

Tabela 25.8. Widoki przydatne przy monitorowaniu i rozwiązywaniu problemów z grupami dostępności

Widok DMV	Opis
sys.availability_groups	Zwraca wiersz dla każdej grupy dostępności, z której lokalny egzemplarz systemu SQL Server zawiera replikę wysokiej dostępności.
sys.dm_hadr_availability_group_states	Zwraca wiersz dla każdej grupy dostępności, z której lokalny egzemplarz systemu SQL Server zawiera replikę wysokiej dostępności.
sys.availability_replicas	Zwraca wiersz dla każdej repliki wysokiej dostępności z wszystkich grup dostępności, z których lokalny egzemplarz systemu SQL Server zawiera repliki.
sys.dm_hadr_availability_replica_states	Zwraca wiersze ze stanem każdej lokalnej repliki wysokiej dostępności, a także każdej zdalnej repliki z tych samych grup dostępności.
sys.dm_hadr_database_replica_states	Zwraca wiersz dla każdej bazy danych z grup dostępności, z których repliki znajdują się w lokalnym egzemplarzu systemu SQL Server.
sys.dm_hadr_database_replica_cluster_states	Zwraca wiersz ze szczegółowymi informacjami o stanie baz wysokiej dostępności z każdej grupy dostępności z danego klastra WSFC.
sys.availability_group_listener_ip_addresses	Zwraca wiersz dla każdego dostępnego adresu IP odbiornika grupy dostępności.
sys.availability_group_listeners	Dla danej grupy dostępności zwraca 0 wierszy (co oznacza, że z daną grupą dostępności nie jest powiązana żadna nazwa sieciowa) lub zwraca wiersz dla każdego odbiornika danej grupy dostępności skonfigurowanego w klastrze WSFC.
sys.dm_tcp_listener_states	Zwraca wiersze z dynamicznie zmieniającymi się informacjami o stanie każdego odbiornika TCP.
sys.fn_hadr_is_primary_replica	Zwraca 1, jeśli dana baza działa w replce głównej. W przeciwnym razie zwraca 0.

Podsumowanie

Grupy dostępności AlwaysOn to solidne rozwiązanie zapewniające wysoką dostępność. W grupie działa do dziewięciu partnerów. To rozwiązanie obsługuje tryby synchroniczny i asynchroniczny, nie wymaga współużytkowanej infrastruktury dyskowej i pozwala zastosować nazwę sieciową, aby niezauważalnie przełączać aplikacje klienckie między replikami. Za pomocą replik pomocniczych możesz wykonywać kopie zapasowe i operacje w trybie tylko do odczytu. Pozwala to odciążyć replikę główną. Grupy dostępności mają najlepsze cechy klastrów WSFC i mechanizmu tworzenia kopii lustrzanych z systemu SQL Server. Udostępniają też inne funkcje, dzięki czemu zapewniają zestaw mechanizmów obsługujących korporacyjne aplikacje o znaczeniu strategicznym.

Skorowidz

.NET, 221
.NET CLR Exceptions, 229
.NET CLR Loading, 229
.NET CLR Memory, 229

A

Activity Monitor, 132
administrator
 baz danych, 37
 BI, 38
 produkcyjnej bazy danych, 291
 programista, 289
administrowanie bazami SQL Database, 795, 809
adres URL
 narzędzia Report Manager, 741
 URL usługi sieciowej, 737
agenty replikacji, 475
akcje, 385
aktualizacja
 aktywnego serwera, 672, 673
 katalogu pełnotekstowego, 109
 klastrow, 524, 527
 systemu, 91, 94, 105
 typu Cumulative Update, 544
 usług Reporting Services, 109
aktualizowanie, 67
 do wersji 64-bitowej, 110
 przesyłania dzienników, 639
 w miejscu, 94, 97
aktywna replika pomocnicza, 839
aktywność dysków, 360
alerty, 166, 317, 668
 dotyczące wydajności, 169
 dotyczące zdarzeń, 170
aliasy dla klientów, 634
alokacja plików, 318
AlwaysOn, 817, 844
analizowanie
 metadanych, 843
 operacji, 470
 wydajności kwerend, 444
analizy
 biznesowe, 38
 następcze, 731
aplikacja Secure Store Service, 778
aplikacje POS, 480
aplikacje warstwy danych, 563
APS, Analytic Platform System, 29
architektura
 bazy SQL Database, 796
 grup dostępności, 818
 kopii lustrzanych, 644
 mechanizmu SQL Trace, 375, 376
 NUMA, 296, 334
 P2P, 481, 482
 przesyłania dzienników, 611
 serwera SSAS, 714
 systemu, 35, 39, 772
 WSSRA, 288
 x64, 294
archiwizowanie danych, 606
ARM, Analyze, Report, Migrate, 249
asynchroniczne
 aktualizowanie statystyk, 453
 przesyłanie komunikatów, 189
ATA, Advanced Technology Attachment, 303
automatyczne przełączanie awaryjne, 657

automatyzowanie
 pracy, 147
 systemu, 156
 autoryzacja dostępu, 237

B

baza danych
 rozmieszczanie plików, 313
 baza danych
 master, 42, 576
 MDW, 267, 292
 model, 43, 84, 577
 msdb, 43, 576
 resource, 41
 tempdb, 42, 83, 314, 577
 SQL Database, 796, 803
 SSRS, 739
 bazy danych
 częściowo niezależne, 120
 systemowe, 122
 BCP, Bulk Copy Program, 557
 bezpieczeństwo, 231
 BI, Business Intelligence, 36, 679
 BISM, BI Semantic Model, 794
 blokada
 indeksu, 453
 partycji tabeli, 321
 błąd strony, 298, 300
 błąd strony miękkiej, 300
 BPE, Buffer Pool Extension, 339
 bufor, 339

C

cele monitorowania, 354
 centralne serwery zarządzania, 139
 chmura, 36, 795
 CLR, Common Language Runtime, 219
 CTE, Common Table Expression, 83
 czas trwania przełączania awaryjnego, 661
 częstotliwość odświeżania danych, 791

D

DAC, Data Tier Application, 563
 dane
 replikacji, 475
 uwierzytelniające, 174, 789
 XML, 453
 zewnętrzne, 779

DAS, Direct-Attached Storage, 61
 definicja wydajności, 287
 definiowanie obciążenia, 312
 DMO, Dynamic Management Objects, 44, 113
 DMV, 231
 dodatek PowerPivot, 783
 dodawanie
 baz danych, 834
 replik, 831
 replik w platformie Azure, 832
 systemu do klastra, 537
 środowiska dla projektu, 699
 węzła, 503
 wiersza, 254
 domeny aplikacji, 221
 domyślny ślad, 357, 371, 373
 dostawcy danych, 778
 dostęp
 do danych, 427
 do pamięci, 343
 do replik pomocniczych, 821
 do serwera, 807
 do zewnętrznych danych, 778, 789
 tylko do odczytu, 839
 dostępność, 583
 dostrajanie
 pamięci, 341
 wydajności, 280
 wydajności kodu, 415
 wydajności kwerend, 420
 drzewo parsowania, 417
 DSS, Decision Support System, 289
 DTA, Database Tuning Advisor, 320, 454, 468
 DW, Data Warehousing, 289
 dynamicznie dodawane procesory, 334
 dysk, 62, 304, 360, 362
 magnetyczny, 302
 opóźnienia, 306
 pamięć podręczna, 304
 RAID, 307
 SSD, 62, 305
 dystrybutor, 474, 483
 działanie
 indeksów, 451–454
 klastrowania, 517
 dzielenie partycji, 329
 dziennik, 211, 410, 607
 błędów, 131, 410, 746
 transakcji, 39, 571
 wykonywania raportów, 746
 zdarzeń, 411, 687

E

edycje systemu
 Business Intelligence Edition, 53
 Enterprise Edition, 52
 Standard Edition, 53
 egzemplarz systemu, 520
 ekran
 Database Engine Configuration, 69
 New Session, 397, 398, 399
 Server Properties, 126
 eksportowanie pakietów, 690
 ekstenty, 318
 dedykowane, 318
 niejednorodne, 318
 Excel
 dodatek PowerPivot, 770, 783

F

filtr, 386
 filtrowanie
 obiektów, 131
 publikowanych danych, 489
 filtry sesji zdarzeń, 394
 fizyczna przestrzeń adresowa, 298
 formaty planów, 423
 fragmentacja indeksu, 462
 funkcja
 AlwaysOn Availability Groups, 36
 DMF, 400
 PowerPivot for SharePoint, 80
 Resource Governor, 261
 suser_name(), 247
 sys.dm_exec_sql_text, 138
 funkcje, 38, 243
 DMF, 357
 kopii lustrzanych, 659
 nieobsługiwane, 106
 partycjonujące, 322, 323
 przestarzałe, 108
 przywracania, 600
 replik pomocniczych, 839

G

GAM, Global Allocation Map, 318
 generowanie
 klucza, 786
 planów wykonywania, 418
 planów wykonywania kwerend, 423

 raportów, 610
 snapshota, 490
 tabel, 812
 widoków, 812
 gniazdo, 296
 grupa
 plików, 324
 konwersacji, 202
 operacyjna, 346
 serwerów, 139
 grupy dostępności, 241, 676, 817
 AlwaysOn, 516
 architektura, 818
 dodawanie baz, 834
 konfigurowanie, 823
 monitorowanie, 846
 panel kontrolny, 844, 845
 przełączanie awaryjne, 834
 repliki, 819
 role, 819
 rozwiązywanie problemów, 846
 sprawdzanie konfiguracji, 829
 ustawianie baz, 825, 834
 włączanie obsługi, 823

H

harmonogram, 162
 odświeżania pamięci podręcznej, 766
 przetwarzania subskrypcji, 764
 historia
 kopii zapasowych, 574
 raportu, 766

I

IAM, Index Allocation Map, 250, 318
 IDE, Integrated Drive Electronics, 303
 identyfikowanie przyczyn zakleszczenia, 372
 ilość pamięci, 343
 implementowanie replikacji, 483
 migawkowej, 487
 transakcyjnej, 500
 importowanie
 danych, 789
 pakietów, 690
 indeksowanie baz danych, 449
 indeksy, 253
 filtrowane, 451, 456
 klastrowane, 429, 455

indeksy

- nieklastrowane, 432, 453, 455
- nieużywane, 406
- oparte na kolumnach, 37, 450, 456
- oparte na przedziałach, 256
- pełnotekstowe, 453, 458, 577
- podzielone na partycje, 442, 452, 454, 461
- pokrywające, 455
- przechowywane w pamięci, 458
- przestrzenne, 452, 459
- wierszowe, 455
- XML-owe, 451, 458
- z haszowaniem, 253

informacje

- o blokadach, 404
- o brakujących indeksach, 468
- o kopiach lustrzanych, 666
- o podłączonych użytkownikach, 408
- o połączeniu, 760
- o poziomie fragmentacji indeksów, 463
- o procedurze składowanej, 278
- o skanowaniu tabeli, 431
- o stanie subskrypcji, 507
- o tabeli, 276
- o zerowaniu plików, 286

inicjowanie plików

- plików bazy danych, 83, 284
- snapshota, 498

In-Memory OLTP, 249

- indeksy, 253
- rekordy, 252
- struktury tabel, 252
- wirtualizacja, 258

inspekcja działań użytkowników, 372

instalacje

- nienadzorowane, 70
- nadzorowane, 67
- równoległe, 66

instalowanie

- funkcji PowerPivot, 80
- klastra, 531, 532
- pakietów, 691, 696
 - eksportowanie, 694
 - importowanie, 694
 - narzędzie DTUtil, 694
 - przenoszenie, 694
 - zapisywanie, 694
- systemu, 66
- usług, 76
- węzłów klastra, 533

instrukcja

- cmpxchg16b, 258
- DELETE, 441
- DENY, 237
- GRANT, 237
- RESTORE, 603
- REVOKE, 237

instrukcje

- do testowania wydajności, 425
- instalowania systemu, 538
- SQL, 290
- tworzenia kopii zapasowych, 592
- wdrażania przesyłania dzienników, 624

integracja

- z CLR, 222, 231
- z SharePoint, 769

izolacja snapshotów, 841

J

jednostki

- docelowe, 387
- pośredniczące, 175

język

- ASSL, 717
- SQL, 290
- T-SQL, 38, 136, 157, 222, 415

K

kardynalność, 418, 419

katalog SSIS, 695

kategorie zadań, 158

klastry

- dodawanie systemu, 537
- grupy zasobów, 540
- instalacja systemu, 538
- instalowanie węzłów, 533
- monitorowanie, 546
- problemy, 547
- ustawianie dysków, 541
- używanie koordynatora MSDTC, 536
- Windows Server 2012 R2, 532, 535
- WSFC, 532, 533, 614, 635
- z wieloma egzemplarzami, 522, 524
- z dwoma węzłami, 522, 543

klastrowanie, 511

- przygotowywanie infrastruktury, 528
- przygotowywanie sprzętu, 528
- w Windows Server 2012 R2, 531

- klucz główny, 208, 786
- klucze szyfrowania, 744
- kolacja, collation, 58, 63
 - systemu SQL Server, 64
 - systemu Windows, 64
- kolejki, 197
- kolejki komunikatów, 192
- kolejność sortowania, 64
- koligacja, 336
- komórki
 - MLC, 305
 - SLC, 305
- komponenty
 - CLR, 223
 - przepływu danych, 682
 - systemu, 109
- kompresja, 37, 330
 - danych, 325, 331, 332
 - indeksów, 451
 - kopii zapasowej, 569
 - przedrostkowa, 328
 - słownikowa, 328
 - stron, 328
 - tabel, 451
 - wierszy, 326
- komunikat o błędzie, 808
- konfigurowanie
 - alertu, 670
 - baz MDW, 267
 - baz SQL Database, 798
 - dystrybucji, 483
 - grupy dostępności, 823
 - harmonogramu przetwarzania subskrypcji, 764
 - inicjowania snapshota, 498
 - istniejących grup dostępności, 830
 - katalogu SSIS, 695, 696
 - konta usługowego, 736
 - narzędzia SQL Server Agent, 176
 - narzędzia SSS, 786
 - nośników, 586
 - nowego alertu, 317
 - nowych subskrypcji, 757
 - obsługi błędów, 725
 - odbiornika, 828
 - odświeżania danych, 792
 - opcji serwera, 542
 - pakietów, 697
 - pakietu SSIS, 568
 - parametrów pakietu, 700
 - plików przywracanej bazy, 601
 - po instalacji, 81
 - publikacji snapshota, 488
 - replikacji migawkowej, 483
 - replikacji w trybie P2P, 501
 - Service Brokera, 193
 - serwera, 279
 - serwera lustrzanego, 670
 - SharePointa, 771
 - sieci, 541
 - subskrypcji, 493, 759
 - synchronizacji danych, 829
 - systemu, 70, 126, 282
 - tworzenia kopii zapasowych, 843
 - udziału plikowego, 762
 - usługi SSIS, 684
 - ustawień systemu, 85
 - właściwości planu konserwacji, 149
 - zadania, 151, 155
 - zapory, 806
 - źródła danych, 751
- konserwacja, 582
 - bazy danych, 589
 - indeksów, 461
- konto
 - domenowe, 743
 - SA, 85
 - usługowe, 65, 171
 - wykonawcze, 743
- kontrakty, 196
- kontrola
 - wersji wierszy, 83
 - zmian, 683
- kontrolery dysków, 63
- kończenie przesyłania dzienników, 637
- koordynator MSDTC, 536
- kopia, 556
 - częściowa, 553
 - pełna, 553
 - zapasowa, 86
 - zapasowa bazy SSAS, 727
- kopie lustrzane, 643
 - architektura, 644
 - automatyczne przełączanie awaryjne, 657, 660
 - baz danych, 516
 - funkcje, 659
 - danych, 635
 - grupy dostępności, 676
 - klastry, 675
 - przesyłanie dzienników, 676
 - punkty końcowe, 648

kopie lustrzane
 replikacja transakcyjna, 676
 ręczne przełączanie awaryjne, 662
 synchronizacja serwerów, 654
 tryb wysokiego bezpieczeństwa, 656
 tryby działania, 645
 uruchamianie sesji, 655
 używanie monitora, 665
 wymuszone przełączenie awaryjne, 664
 zmiana ról, 659
 kopie zapasowe, 551, 827
 indeksów pełnotekstowych, 577
 różnicowe, 554
 różnicowe częściowe, 554
 kostka SSAS, 38
 koszt stosowania indeksów, 469
 kreator
 bazy SQL Database, 798–801
 Deployment Wizard, 719–721
 grupy dostępności, 824
 Import and Export Wizard, 557, 701
 konfiguracji bazy MDW, 267
 kopiowania baz danych, 563–569
 New Session Wizard, 391–396
 Package Installer Wizard, 692, 693
 planów konserwacji, 148
 skryptów, 557
 kroki zadań, 156, 159
 kubełek, 255
 kwerendy
 o niskiej wydajności, 421
 oczekujące, 407
 usług, 157
 kworum, 518, 657

L

licencje, 55
 na rdzeń procesora, 55
 na serwer, 56
 liczba
 dysków, 303
 indeksów, 469
 węzłów, 523
 licznik
 Bajty prywatne, 368
 Bajty wirtualne, 368
 Buffer Cache Hit Ratio, 368
 CLR, 228
 Czas procesora, 359

 Czas uprzywilejowany, 359
 Długość kolejki procesora, 359
 Dostępna pamięć, 367
 Free Pages, 368
 Page Life Expectancy, 368
 zasobów procesora, 359
 Zestaw roboczy, 368
 liczniki dysków
 fizycznych, 361
 logicznych, 361
 lista
 baz SQL Database, 802
 dostępnych raportów, 412
 raportów, 421
 SharePointa, 783
 zdarzeń, 398
 login, 236, 238, 628
 lokalizacja kopii zapasowych, 585, 590
 lokalność danych, 343

Ł

łańcuchy uprawnień, 244–247
 łączenie operatorów dostępu, 434

M

macierz RAID, 62, 307, 363
 sprzętowa, 364
 magazyn wersji, 454
 maksymalna ilość pamięci, 343
 maska koligacji, 336
 MAXDOP, 336
 mechanizm
 klastrowania, 516
 odświeżania danych, 790
 SQL Trace, 375
 szacowania kardynalności, 419
 menedżer
 pamięci wirtualnej, 298
 zadań, 291
 metadane, 843
 metody zapewniania wysokiej dostępności, 675
 miejsce tworzenia kopii zapasowych, 828
 minimalny przestój, 639
 model
 BULK_LOGGED, 572
 FULL, 572
 MOLAP, 713
 P2P, 481
 SIMPLE, 573

- modele
 - abstrakcyjne, 713
 - odzyskiwania, 570, 572
 - replikacji, 478
 - standardowe, 713
 - tabelowe, 713
 - tabelaryczne SSAS, 38
- moduł, 382
- modyfikowanie
 - danych, 440
 - sesji, 399
- monitor
 - tworzenia kopii lustrzanych, 665
 - wydajności, 291, 356, 358, 508
- monitorowanie, 340
 - aktywności, 688
 - długości kolejek, 365
 - dysków, 365
 - dziennika błędów, 410
 - dzienników, 410
 - dzienników zdarzeń, 411
 - ilości dostępnej pamięci, 367
 - kompresji danych, 331
 - liczników, 365
 - odczytu i zapisu, 365
 - opóźnień, 364
 - pracy, 350
 - procesów, 136
 - przepustowości, 363, 364
 - replikacji, 506
 - systemu, 353, 400, 625
 - procedury składowane, 626
 - transferu, 365
 - użytkowania pamięci, 367
 - wydajności, 228
 - zdarzeń, 371
- zdarzeń SSAS, 731

N

- narzędzia
 - administracyjne, 114
 - do monitorowania wydajności, 369
 - do rozwiązywania problemów, 121
 - konfiguracyjne, 114
 - Microsoft SQL Server Data Tools, 226
 - monitorujące, 356
- narzędzie
 - Activity Monitor, 132
 - Activity Monitor w SQL Server Management Studio, 357

- APS, 29
- ARM, 249, 266
- Azure Management Portal, 798, 799
- BCP, 557
- Configuration Manager, 734
- Database Mirroring Monitor, 668
- Distributed Replay, 378
- DRU, 105
- DTA, 468, 469, 470
- DTEExec, 702
- DTEExecUI, 702
- DTUtil, 689, 694
- Execute Package, 703
- Failover Cluster Manager, 830
- Flight Recorder, 731
- Logman, 369
- PDW, 29
- PerformancePoint, 782
- Relog, 370
- Report Manager, 741, 747
- SQL Server 2014 Upgrade Advisor, 98
- SQL Server Agent, 156, 171, 673, 704
- SQL Server Backup to URL, 37
- SQL Server Backup to Windows Azure, 37
- SQL Server Configuration Manager, 86
- SQL Server Management Studio, 126
- SQLDiag.exe, 142
- SQLDumper.exe, 141
- SSDT, 701, 719, 738
- SSDT-BI, 38
- SSS, 785, 786
- System Center Advisor, 411
- WMI, 170

- natychmiastowe inicjowanie plików, 284
- nawiązywanie połączenia, 807
- nazwa FQDN serwera, 808
- nienadzorowane konto usługowe, 781, 785
- nieudana instalacja, 88
- niezależna baza danych, 831
- niezależny użytkownik, 236
- niezawodność, 232
- nowe funkcje, 36, 92
- nowe subskrypcje, 758

O

- obciążenie, 312
- obiekty
 - DMF, 45
 - DMO, 44
 - SSAS, 721

- obiekty
 - typu pred_compare, 386
 - typu pred_source, 387
 - zdarzeń rozszerzonych, 382
- obliczenia, 713
- obraz z kopią zapasową, 577
- obsługa
 - błędów, 725
 - grup dostępności, 823
 - przekierowywania połączeń, 840
 - techniczna, 141
 - wejścia-wyjścia, 312
- odbieranie komunikatów, 206
- odbiornik
 - grupy dostępności, 838
 - XMLA, 714
- odświeżanie
 - danych, 776, 784–790
 - pamięci podręcznej, 765
- odtworzenie śladu, 377
- odzyskiwanie, 570, 580
 - baz danych, 595
 - systemu, 609
- ograniczanie dostępu do zasobów, 805
- okno
 - Installation Center, 67
 - New Alert, 167
 - Object Explorer, 225, 813
 - Object Explorer Details, 126
 - projektanta planów konserwacji, 154
 - Ready to Install, 69
- określanie
 - celów monitorowania, 354
 - punktu odniesienia, 355
- OLAP, Online Analytical Processing, 712
- OLTP, Online Transaction Processing, 36, 249
- opcje
 - generowania raportów, 152
 - importu i eksportu, 558
 - kopii zapasowych, 591
 - przetwarzania, 724
 - przywracania bazy, 602
 - replik, 826
 - snapshotów, 767
 - sprzętowe, 58
 - systemu, 81
 - śledzenia, 140, 287
 - uruchomieniowe, 117
 - uwierzytelniania, 787

- operacje
 - DBCC, 83
 - na indeksach, 452
 - na indeksie, 464
 - wejścia-wyjścia, 300, 312
- operator awaryjny, 164
- operatory
 - dostępu, 427, 434
 - sortowania, 83
 - złączeń, 435
- operatorzy, 163
- opóźnienia dysków twardych, 303
- opóźnione utrwalanie, 37
- oprogramowanie, 63
- optymalizator kwerend, 110, 419
- optymalizowanie
 - aplikacji, 311
 - sprzętu serwera, 292
 - systemu, 311

P

- PaaS, Platform as a Service, 795
- pakiety, 382
 - poprawek i aktualizacje, 85
 - sieciowe, 83
 - SSIS, 38, 568, 688, 694
- pamięć, 59, 82, 258, 297, 339
 - bliska, 343
 - daleka, 343
 - DAS, 61
 - DIMM, 297
 - dyskowa, 60
 - fizyczna, 297
 - flash, 305
 - NAND, 305
 - NOR, 305
 - podręczna, 295
 - RAM, 297
 - zajęta, 409
- panel kontrolny grupy dostępności, 845
- parametry
 - uruchomieniowe, 116
 - wiersza poleceń, 71
- parsowanie, 417
- partycje, 37, 320, 329, 362, 454, 606
- PDW, Parallel Data Warehouse, 29
- PerformancePoint
 - źródła danych, 782
- pętla zagnieżdżona, 435

- PFS, Page Free Space, 318
- plan
 - konserwacji bazy danych, 147, 148, 589
 - odświeżania danych, 792
 - tworzenia kopii zapasowych, 584, 634
 - wykonywania kwerend, 409, 423–426, 437, 440, 443
 - zasilania, 282
- planowanie systemu, 58
- platforma
 - .NET, 221
 - SharePoint 2013, 769
 - Windows Azure, 833
- plik
 - CertSerwerGlowny.sql, 649
 - dostępDoIndeksu1.sql, 428
 - dostępDoIndeksu2.sql, 430
 - dostępDoIndeksu3.sql, 431
 - dostępDoIndeksu4.sql, 433
 - Instalacja-Sql2014.ps1, 73
 - konfiguracyjny, 71
 - logman_create_io.cmd, 370
 - nieuzywaneIndeksy.sql, 407
 - oczekiwanieNaPamiec.sql, 407
 - planyWykonywanychKwerend.sql, 409
 - stronicowania, 299
 - tworzKolejki.sql, 380
 - tworzPowiadomOZdarz.sql, 380
 - uzywaneIndeksy.sql, 405
 - wykonywaneKwerendy.sql, 402
 - wykorzystPamieci.sql, 409
 - wykorzystZasobow.sql, 402
 - XEdlugieKwerendy.sql, 389
 - XEnodyfSesji.sql, 389
 - zlaczenie.sql, 435
 - zlaczenie3.sql, 439
- pliki bazy danych, 39
- podejście równoległe, 97
- podział
 - na partycje, 442, 606
 - tabel i indeksów, 320, 606
- polecenia
 - systemu operacyjnego, 156
 - usług SSAS, 157
- połączenia
 - aplikacji klienckich, 837
 - zarządzane, 780
- połączenie
 - DAC, 121
 - z bazą master, 809
 - ze źródłem danych, 777
 - pomiar obciążenia pamięci, 341
 - poprawki SP, 544
 - porty TCP/IP, 85
 - porządkowanie indeksów, 463
 - POS, Point of Service, 480
 - Power View, 775
 - PowerPivot, 770, 783
 - odświeżanie danych, 785
 - PowerPivot Data Refresh, 792
 - powiadomienia
 - dla kwerend, 83
 - o zadaniach, 161
 - o zdarzeniach, 83, 372, 378
 - operatorów, 164
 - poziomy ochrony pakietu, 707
 - predykat, 386
 - priorytety, 201
 - problemy
 - z aktualizacją, 102
 - z klastrami, 547
 - procedura
 - sp_configure, 130
 - sp_replicounters, 509
 - sp_who, 137
 - procedury składowane, 243, 263
 - natywnie kompilowane, 263
 - proces
 - przesyłania dzienników, 613
 - przetwarzania kwerend, 417
 - przywracania, 595
 - procesor, 58, 258, 294, 333
 - czas, 359
 - kwerend, 714
 - procesy
 - BI, 279
 - ETL, 38
 - produkcyjne bazy danych, 36, 291
 - projektant planów konserwacji, 153
 - przebieg szeregowania, 336
 - przechowywanie
 - danych, 305, 614
 - połączeń, 781
 - przekazywanie zdarzeń, 183
 - przekierowywanie
 - klientów, 633, 674
 - operacji odczytu, 840
 - połączeń, 840
 - przełączenie
 - awaryjne, 520, 670, 820, 834
 - automatyczne, 660, 820
 - nieplanowane, 632

przłączanie

awaryjne

planowane, 630

ręczne, 662, 820

węzłów, 545

wymuszone, 664

kontekstu, 335

między serwerami, 632

modeli odzyskiwania, 573

ról, 630

przenoszenie

grup, 544

loginów, 628

przestawianie partycji, 329

przestój, 672

przestój standardowy, 639

przesyłanie dzienników, 515, 607–610, 676

aktualizacja, 639

kończenie, 637

scenariusze, 608

wdrażanie, 615

wydajność, 638

wymagania systemowe, 613

przetwarzanie

kostek, 723

kwerend, 416, 442

obiektów SSAS, 721

partycji, 723

ponowne, 723

w modelu MOLAP, 722

w modelu tabelowym, 725

wymiarów, 722

przypisywanie

ról użytkownikom, 749

uprawnień, 811

przyspieszanie zapisu, 304

przystawka Usługi systemu Windows, 685

przywracanie, 578

bazy danych, 596, 597, 729

dzienników transakcji, 596

grup plików, 597

kopii zapasowych, 727

plików, 597

snapshotów, 598

stanu, 551, 553, 575

systemowych baz danych, 604

tabel metadanych, 599

publikacja ze snapshotem, 493

pula

BPE, 339, 340

buforów, 409

zasobów, 345, 348

punkty końcowe, 238

R

RAID, 62, 307, 363

RAID 0, 307

RAID 1, 307

RAID 10, 308

RAID 5, 308, 595

RAID 6, 308

raport

Contention Analysis, 276

Stored Procedure Based, 277

Usage Analysis, 274, 275

raporty, 124

o blokadach, 136

o bazach danych, 124

o błędach, 88

standardowe, 411

z poziomu serwera, 124

rdzeń, 296

reagowanie na alerty, 170

reindeksacja, 83

rejestrowanie

danych, 159

danych o krokach zadań, 159

śladu, 378

rekordy, 252

relacje z klientami, 480

replika pomocnicza

dostęp tylko do odczytu, 839

kopie zapasowe, 842

połączenie, 840

wydajność, 841

replikacja, 473, 515, 636

agenty, 475

dane replikacji, 475

implementowanie, 483

migawkowa, 476, 499

modele, 478

monitor, 506

P2P, 502

procedura sp_replcounters, 509

role, 474

skrypty, 505

transakcyjna, 477, 676

- transakcyjna ze skalaniem, 500
- w trybie P2P, 477, 501
- widoki DMV, 508
- z publikacją, 478
- zadania, 476
- ze skalaniem, 477
- repliki, 819, 831
- repliki pomocnicze, 821, 839
- Report Manager, 747
 - historia raportu, 766
 - menu nawigacyjne, 748
 - menu zadań, 750
 - New Subscription, 757
 - opcje przetwarzania, 764
 - strona Data Sources, 756
 - zabezpieczenia, 767
 - zarządzanie raportami, 754
- Resource Governor, 37, 344
 - grupy operacyjne, 346
 - klasyfikowanie, 347
 - monitorowanie pracy, 350
 - pule zasobów, 345
- rodzaje
 - indeksów, 458
 - klastrów, 522
 - pamięci flash, 305
 - uwierzytelniania, 234
- rola
 - SQLAgentReaderRole, 172
 - SQLAgentUserRole, 172
- role
 - bazy danych, 575, 708
 - serwera, 575
 - użytkownika, 241
 - wbudowane, 242
 - w grupach dostępności, 819
- rozmieszczanie plików, 313
- rozszerzenia puli buforów, 339
- rozszerzenie Power View, 775
- rozwiązywanie problemów, 121
- równoległe przetwarzanie kwerend, 321
- równoważenie obciążenia, 805
- równoważenie obciążenia sieci, 633
- różnicowe kopie zapasowe plików, 554
- rysunki VDW, 784

S

- SAN, Storage Area Network, 61
- SAS, Serial Attached SCSI, 303
- SCA, System Center Advisor, 411

- skalanie partycji, 329
- schematy, 43
 - bazy danych, 290
 - partycjonowania, 324
- SCSI, Small Computer Systems Interface, 303
- Secure Store Service, 779
- Server Objects, 813
- Service Broker, 189, 191, 192
 - grupy konwersacji, 202
 - kolejki, 197
 - kontrakty, 196
 - odbieranie komunikatów, 206
 - priorytety, 201
 - przesyłanie komunikatów, 207
 - punkty końcowe, 209
 - trasy, 199
 - typy komunikatów, 195
 - usługi, 199
 - ustawianie stanu, 193
 - wysyłanie komunikatów, 203
 - zapisywanie danych, 211
 - zewnętrzna aktywacja, 209
- serwer
 - docelowy, 185
 - DNS, 633
 - główny, 612
 - lustrzany, 670, 673
 - monitorujący, 612, 623
 - nadrzędny, 185
 - pomocniczy, 612
 - rezerwowy, 608
 - SQL Database, 799, 801
 - SSAS, 711, 715
 - elementy architektury, 714
 - instalowanie baz, 719
 - język ASSL, 717
 - właściwości, 715
 - wymagane usługi, 717
 - zarządzanie, 715
 - zarządzanie bazami, 719
 - SSRS, 747
 - świadka, 657
 - usług SSIS, 685
 - zapasowy w trybie offline, 514
 - zapasowy w trybie online, 515
- sesja
 - system_health, 374
 - zdarzeń, 388
 - zdarzeń rozszerzonych, 391
- SGAM, Shared Global Allocation Map, 318

- sieci, 301
 - prywatne, 519
 - publiczne, 519
 - SAN, 61
- silnik
 - bazodanowy, 113
 - OLAP, 712
 - RDBMS, 712
- skalowalność, 231
- skalowanie systemu, 745
- skanowanie
 - indeksów klastrowanych, 429
 - indeksów nieklastrowanych, 432
 - tabeli, 428
- składnia operacji na indeksie, 464
- skrypty, 157
- skrypty powłoki PowerShell, 72
- snapshot, 490, 493, 598, 677, 767
- sortowanie, 64
- spójność pamięci podręcznej, 334
- sprawdzanie
 - połączeń, 135
 - poprawności obrazów, 577
 - poprawności replikacji migawkowej, 499
 - systemu, 81
 - węzłów, 532
- sprzęt serwera, 292
- SQL Database
 - administrowanie, 795, 809
 - konfigurowanie, 798
 - kopie zapasowe, 813
 - nawiazywanie połączenia, 807
 - niedostępność narzędzi, 815
 - replikacja, 814
 - warstwa infrastruktury, 797
 - warstwa klientów, 797
 - warstwa platformy, 797
 - warstwa usług, 797
- SQL Server 2014, 29, 35
- SQL Server Agent, 704, 815
 - jednostki pośredniczące, 173
 - kategoria Advanced, 177
 - kategoria Alert System, 178
 - kategoria Connection, 180
 - kategoria General, 176
 - kategoria History, 180
 - kategoria Job System, 179
 - konfiguracja, 176
 - konfigurowanie zadań, 673
 - podsystemy, 173
 - zabezpieczenia, 171
 - zadania, 156, 630
- SQL Server Configuration Manager, 114, 686
 - właściwości usługi, 686
- SQL Server Data Tools, 226
- SQL Server Management Studio, 123, 126, 292, 584, 599, 808
 - ekran Server Properties, 126
 - konfigurowanie systemu, 126
 - kończenie przesyłania dzienników, 637
 - monitorowanie systemu, 626
 - Object Explorer Details, 126
 - przesyłanie dzienników, 616
 - raporty, 124
 - zakładka Advanced, 129
 - zakładka Connections, 129
 - zakładka Database Setting, 129
 - zakładka Memory, 127
 - zakładka Permissions, 130
 - zakładka Processors, 127
 - zakładka Security, 129
 - zarządzanie pakietami, 689
- SQL Server Native Client, 40
- SQL Server Profiler, 356, 371, 375
- SQL Server Upgrade Advisor, 98
- SQL Trace, 357, 371, 374
- SQLAgentOperatorRole, 172
- SQLCLR, 219
- SQLOS, 342
- SSAS, SQL Server Analysis Services, 679
- SSD, Solid State Drive, 62
- SSDT, SQL Server Data Tools, 683
- SSRS, SQL Server Reporting Services, 679
- standardowe raporty, 357, 413
- statystyki oczekiwania, 404
- stosowanie klastrowania, 514
- strony, 318
- subskrybent, 475, 478, 480
- subskrybent aktualizujący dane, 481
- subskrypcja, 764
- subskrypcja sterowana danymi, 759
- synchronizowanie
 - baz SSAS, 729
 - loginów, 236
 - serwerów, 654
 - zależności, 627
- synonimy, 44
- system
 - Center Advisor, 357
 - SQLOS, 342
 - wielordzeniowy, 296
 - zgłaszania problemów, 322

systemowe
 bazy danych, 41
 procedury składowane, 357
 szablon sesji zdarzeń, 393
 szacowanie
 kardynalności, 419
 oszczędności miejsca, 329
 szeregowanie zadań, 334
 szyfrowane kopie zapasowe, 37, 553

Ś

śledzenie, 340
 środowisko
 CLR, 52, 219–222
 produkcyjne, 444
 SQLCLR, 815
 systemu, 35
 Visual Studio, 224
 wirtualne, 56

T

tabele
 metadanych, 599
 podzielone na partycje, 454
 z historią kopii, 574
 zoptymalizowane, 250, 260
 tabelowe źródła danych, 782
 technologia
 AlwaysOn, 817
 Hyper-Threading, 295
 In-Memory OLTP, 249
 Resource Governor, 345
 SQLCLR, 221
 testy, 544
 po aktualizacji, 110
 wydajności, 284, 425
 tokeny, 181
 transakcje OLTP, 36
 trasy, 199
 tryb
 DirectQuery, 79
 DirectQuery with In-Memory, 79
 EXECUTE AS OWNER, 264
 HOLAP, 78
 In-Memory, 79
 In-Memory with DirectQuery, 79
 MOLAP, 78, 712

NORECOVERY, 677
 ROLAP, 78
 tabelaryczny, 78
 tylko do odczytu, 821, 839
 UDM, 78
 uprzywilejowany, 335
 tryby
 dostępności, 819
 działania kopii lustrzanych, 645
 kworum, 519
 przełączania awaryjnego, 820
 wysokiego bezpieczeństwa, 656
 wysokiej wydajności, 658
 tworzenie
 aplikacji usługowej, 774
 certyfikatów, 208
 funkcji partycjonującej, 322
 grup plików, 324
 grupy zasobów, 540
 indeksów, 459
 klastra, 525, 531
 komponentów CLR, 223
 konta użytkownika, 811
 kopii lustrzanych, 648, 653, 669, 838
 kopii zapasowej, 86, 154, 552, 575, 728, 813, 842
 loginu, 810
 obciążenia, 372
 planu, 153
 punktu odniesienia, 372
 schematu partycjonowania, 324
 serwera rezerwowego, 608
 SQL Database, 798
 subskrypcji, 494
 tabeli, 813
 typy
 danych, 45, 47
 binarne, 49
 dokładne liczbowe, 48
 przybliżone liczbowe, 48
 systemowe, 51
 z datą i czasem, 49
 znakowe, 47
 kompresji, 330
 komunikatów, 195
 LOB, 83
 przełączania awaryjnego, 820
 publikacji, 488
 replikacji, 476
 serwerów SSAS, 715

U

umożliwianie aktualizacji skoroszytu, 790
 uprawnienia, 575

- do bazy danych, 243
- do tabel i widoków, 243
- na serwerze, 237
- podsystemów, 173
- z poziomu egzemplarza, 239, 240, 241

 uruchamianie

- narzędzia DRU, 105
- narzędzia Upgrade Advisor, 101
- pakietów
 - język T-SQL, 706
 - kreator, 701
 - narzędzie DTEExec, 702
 - narzędzie DTEExecUI, 702
 - narzędzie Execute Package, 703
 - narzędzie SSDT, 701
- przetwarzania, 723

 uruchomieniowe procedury składowane, 118
 usługa, 65, 199

- SSIS, 29, 38
- SSRS, 38
- typu PaaS, 795

 usługi

- Analysis Services, 76, 87, 711, 783
- Excels, 779, 783
- Integration Services, 679
- PerformancePoint, 783
- Reporting Services, 87, 733, 772
- SSAS, 711
 - dostrajanie wydajności, 730
 - monitorowanie, 730
- SSIS, 680
 - główne elementy, 681
 - komponenty przepływu danych, 682
 - komponenty wykonawcze, 682
 - konfiguracja, 684
 - serwer, 685
 - właściwości, 685
 - zabezpieczenia, 706
 - zarządzanie, 683
 - zastosowania, 680
- SSRS
 - Configuration Manager, 734
 - konto usługowe, 736
 - Report Manager, 741
- Visio, 784
 - metody uwierzytelniania, 784
 - odświeżanie danych, 784

WMI, 184
 usprawnienia widoków indeksowanych, 454
 ustawianie

- alertów, 668
- katalogu, 753
- modelu odzyskiwania, 616
- nowej sesji, 399
- pamięci, 82
- planu zasilania, 283
- poczty elektronicznej, 742
- replik pomocniczych, 822, 826

 ustawienie

- Cost Threshold for Parallelism, 337
- MAXDOP, 336
- SAFETY FULL, 672
- SAFETY OFF, 673

 usuwanie

- baz danych, 834
- dużych indeksów, 453
- instalacji silnika bazodanowego, 88
- instalacji systemu, 87
- instalacji usług, 87
- replik, 831

 uwierzytelnianie, 234

- dostępu, 778
- w SQL Server, 234
- w Windows, 235

 użytkowanie pamięci, 366
 użytkownik, 236, 289, 628
 używanie

- indeksów, 405, 461
- narzędzia Upgrade Advisor, 99
- programu SQLDiag.exe, 144
- Service Brokera, 203
- sesji zdarzeń rozszerzonych, 391
- tabel, 461
- technologii Resource Governor, 349
- usług WMI, 184

V

Visual Studio, 224
 VMM, Virtual Memory Manager, 298

W

warstwa

- infrastruktury, 797
- klientów, 797
- platformy, 797

- prezentacji, 776
- usług, 797
- wątek, 296
- wątek lekki, 335
- wdrażanie przesyłania dzienników, 616, 624
- wersje kodu systemu, 93
- węzły, 504
 - aktywne, 517
 - pasywne, 517
- widok
 - DASHBOARD, 803
 - sys.dm_exec_connections, 137
 - sys.dm_xe_packages, 383
 - źródła danych, 713
- widoki
 - DMV, 231, 263, 316, 400, 846
 - systemowe, 445
 - podzielone na partycje, 607
 - z katalogu, 390
- wielkość znaków, 64
- wiersz poleceń, 70
- Windows Azure, 37
- Windows Azure SQL Database, 795
- Windows System Monitor, 228
- wirtualizacja, 258
- właściwości
 - dotyczące pamięci, 716
 - dzienników, 716
 - OLAP, 716
 - raportu, 755
 - serwera SSAS, 715
 - sieci, 716
 - zabezpieczeń, 497
 - zadania, 155
 - związane z bezpieczeństwem, 717
- włączanie
 - In-Memory OLTP, 250
 - parametrów uruchomieniowych, 116
- WMI, Windows Management Instrumentation, 170
- wolna przestrzeń, 408
- współbieżne wykonywanie kwerend, 444
- współużytkowana macierz dyskowa, 517
- wybieranie
 - modelu odzyskiwania, 572
 - modelu subskrypcji, 495
 - pamięci, 61
 - publikacji, 495
 - publikowanych tabel, 489
 - raportów, 274
 - trybu usług, 77

- wydajność, 168, 228, 231, 289, 291
 - dysku, 366
 - kodu, 415
 - kwerend, 420, 444, 464
 - przesyłania dzienników, 638
 - serwera, 279
- wydawca, 474, 478
- wykonywanie
 - kwerendy, 401
 - pakietów usług SSIS, 157
- wykorzystanie liczników pamięci, 369
- wyłączanie zegara, 793
- wymagania sprzętowe, 59
- wymiana certyfikatów, 208
- wyrażenia CTE, 83
- wysoka dostępność, 583, 635
- wysyłanie
 - alertów, 668
 - komunikatów, 203
- wyszukiwanie serwera wydawcy, 494
- wyświetlanie kwerend, 407
- wyzwalacze, 83
- wznawianie pracy bazy danych, 837
- wzorce użytkowania danych, 290, 581

X

- XLMA, 714

Z

- zabezpieczanie
 - egzemplarzy baz danych, 233
 - połączeń, 781
 - usług SSIS, 706
- zabezpieczenia, 171
 - agenta dystrybucji, 496
 - na poziomie wierszy, 247
 - przy integracji, 227
- zadania, 156
- zadania powłoki PowerShell, 157
- zakleszczenie, 372
- zależności baz danych, 630
- zapis
 - na adres URL, 552
 - w dzienniku, 211
 - w Windows Azure, 552
- zapisywanie środowiska, 104
- zapora, 804
- zapora baz SQL Database, 806

zarządzane tworzenie kopii, 552

zarządzanie

bazami, 719

klastrem, 546

kluczami szyfrowania, 744

kopiami zapasowymi, 594

pakietami, 689

pakietami SSIS, 688, 694

pamięcią, 261

połączeniami, 781

projektem, 683

raportami, 754

relacjami z klientami, 480

serwerem SSAS, 715

silnikiem bazodanowym, 113

sprzętem, 293

usługami

Analysis Services, 711

Integration Services, 679

Reporting Services, 733

SIS, 683

użytkownikami, 749

wieloma serwerami, 138, 180, 185

zmianą ról, 627

zasady zapory, 804

zatory w pamięci, 369

zaufane

biblioteki połączeń, 780

lokalizacje plików, 780

zaufani dostawcy danych, 781

zawieszanie pracy bazy danych, 836

zdarzenia, 170, 183, 383

rozszerzone, 292, 356, 371, 381, 385

rozszerzone, 230

SSAS, 731

systemu, 166

zgłaszanie problemów, 322

zgodność wstecz, 106

złączanie

przez scalanie, 438

w pętli, 435

z haszowaniem, 437

zmiana ról, 627

Ż

źródła danych, 756

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>