



# Vantage Administration Intermediate

---

Version 17.10.3

96021  
Student Guide

# Trademarks

The product or products described in this book are licensed products of Teradata Corporation or its affiliates.

Teradata, Applications-Within, Aster, BYNET, Claraview, DecisionCast, Gridscale, MyCommerce, QueryGrid, SQL-MapReduce, Teradata Decision Experts, "Teradata Labs" logo, Teradata ServiceConnect, Teradata Source Experts, WebAnalyst, and Xkoto are trademarks or registered trademarks of Teradata Corporation or its affiliates in the United States and other countries.

Adaptec and SCSISelect are trademarks or registered trademarks of Adaptec, Inc.

Amazon Web Services, AWS, [any other AWS Marks used in such materials] are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

AMD Opteron and Opteron are trademarks of Advanced Micro Devices, Inc.

Apache, Apache Avro, Apache Hadoop, Apache Hive, Hadoop, and the yellow elephant logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. Apple, Mac, and OS X all are registered trademarks of Apple Inc.

Axeda is a registered trademark of Axeda Corporation.

Axeda Agents, Axeda Applications, Axeda Policy Manager, Axeda Enterprise, Axeda Access, Axeda Software Management, Axeda Service, Axeda ServiceLink, and Firewall-Friendly are trademarks and Maximum Results and Maximum Support are servicemarks of Axeda Corporation.

CENTOS is a trademark of Red Hat, Inc., registered in the U.S. and other countries.

Cloudera, CDH, [any other Cloudera Marks used in such materials] are trademarks or registered trademarks of Cloudera Inc. in the United States, and in jurisdictions throughout the world.

Data Domain, EMC, PowerPath, SRDF, and Symmetrix are registered trademarks of EMC Corporation.

GoldenGate is a trademark of Oracle.

Hewlett-Packard and HP are registered trademarks of Hewlett-Packard Company.

Hortonworks, the Hortonworks logo and other Hortonworks trademarks are trademarks of Hortonworks Inc. in the United States and other countries.

Intel, Pentium, and XEON are registered trademarks of Intel Corporation.

IBM, CICS, RACF, Tivoli, and z/OS are registered trademarks of International Business Machines Corporation.

Linux is a registered trademark of Linus Torvalds.

LSI is a registered trademark of LSI Corporation.

Microsoft, Active Directory, Windows, Windows NT, and Windows Server are registered trademarks of Microsoft Corporation in the United States and other countries.

NetVault is a trademark or registered trademark of Dell Inc. in the United States and/or other countries.

Novell and SUSE are registered trademarks of Novell, Inc., in the United States and other countries.

Oracle, Java, and Solaris are registered trademarks of Oracle and/or its affiliates.

QLogic and SANbox are trademarks or registered trademarks of QLogic Corporation.

Quantum and the Quantum logo are trademarks of Quantum Corporation, registered in the U.S.A. and other countries.

Red Hat is a trademark of Red Hat, Inc., registered in the U.S. and other countries. Used under license.

SAP is the trademark or registered trademark of SAP AG in Germany and in several other countries.

SAS and SAS/C are trademarks or registered trademarks of SAS Institute Inc.

SPARC is a registered trademark of SPARC International, Inc.

Symantec, NetBackup, and VERITAS are trademarks or registered trademarks of Symantec Corporation or its affiliates in the United States and other countries.

Unicode is a registered trademark of Unicode, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

The information contained in this document is provided on an "as-is" basis, without warranty of any kind, either express or implied, including the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. In no event will Teradata Corporation be liable for any indirect, direct, special, incidental, or consequential damages, including lost profits or lost savings, even if expressly advised of the possibility of such damages.

The information contained in this document may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

Information contained in this document may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. Teradata Corporation may also make improvements or changes in the products or services described in this information at any time without notice.

# Vantage Administration Intermediate

Version 17.10.3

## Module 0 – Overview & Basics Recap

Course Description.....	0-2
Who Should Attend and Prerequisites .....	0-3
Teradata Certification Exams .....	0-4
Objectives .....	0-5
Space Metrics.....	0-6
Assigning Perm and Spool Limits .....	0-7
Giving One User to Another .....	0-9
Reserving Space for Spool.....	0-10
Views for Space Allocation Reporting .....	0-11
Creating and Using Account IDs .....	0-12
Account String Expansion (ASE) .....	0-13
ASE Accounting Example .....	0-14
Dynamically Changing an Account ID.....	0-15
Example – AMPUsage View .....	0-16
System Accounting Views.....	0-17
Profiles .....	0-18
Impact of Profiles on Users.....	0-19
Example – CREATE PROFILE.....	0-20
Miscellaneous SQL Functions .....	0-21
Access Rights Mechanisms: Summary .....	0-22
Privileges/Access Rights.....	0-23
CREATE USER – Automatic Rights .....	0-24
Implicit, Automatic, and Explicit Rights .....	0-25
Inheriting Access Rights .....	0-26
Access Rights and Views.....	0-27
System Views for Access Rights .....	0-29
What is a Role? .....	0-30
Access Rights Validation and Roles .....	0-31
System Views for Roles.....	0-32
Linux Managed Memory and FSG Cache .....	0-33
Node Memory .....	0-34
Why Teradata Intelligent Memory (TIM).....	0-35
Teradata Intelligent Memory (TIM): Benefits.....	0-36
Teradata Intelligent Memory (TIM): Implementation.....	0-37
Block and Row Terminology.....	0-38
Cylinder Full .....	0-39
Mini-Cylpack .....	0-40

## Module 1 – Logging

Objectives .....	1-2
Why Logging? .....	1-3
Logging Components.....	1-4
Access Logging – Features .....	1-5
Access Logging – Setup.....	1-6
Access Logging – Data Dictionary .....	1-7
Access Logging – Logging Statement .....	1-8
Access Logging – Example .....	1-9
System Usage Logging (ResUsage) – Features .....	1-11
ResUsage Logging – Setup .....	1-13
ResUsage Logging.....	1-14
ResUsage Logging – Data Dictionary Tables.....	1-15
ResUsage Logging – Data Dictionary Views .....	1-16
ResUsage Logging – Aggregation and Join.....	1-17
ResUsage Logging – Data Dictionary Macros .....	1-18
User Logon – Features .....	1-19
User Logon – Setup .....	1-20
User Logon – Host Logging Statement .....	1-21
User Logon – Data Dictionary .....	1-22
Query Logging – Features .....	1-23
Query Logging – Setup .....	1-24
Query Logging – Data Dictionary .....	1-25
Event Logging – Features .....	1-26
Crashdumps – Features .....	1-27
Workload Management Logging – Features.....	1-29
Space Logging – Features .....	1-30
Key Highlights .....	1-31



## Module 2 – Database Query Logging (DBQL) and PDCR

Objectives .....	2-2
Topics.....	2-3
Current Topic – Database Query Logging (DBQL) .....	2-4
How Well Are Your Queries Running?.....	2-5
Why You Need Database Query Logging (DBQL).....	2-6
What is Database Query Logging? .....	2-7
Query Logging – Features .....	2-8
Query Logging – Data Dictionary .....	2-9
BEGIN QUERY LOGGING .....	2-10
BEGIN QUERY LOGGING (WITH) .....	2-11
BEGIN QUERY LOGGING Examples (WITH) .....	2-12
BEGIN QUERY LOGGING (LIMIT).....	2-13
BEGIN QUERY LOGGING Examples (LIMIT).....	2-14
END / REPLACE QUERY LOGGING Statement .....	2-15
FLUSH QUERY LOGGING.....	2-16
DBQL Labs.....	2-17
Current Topic – Performance Data Collection and Reporting (PDCR) .....	2-18
How Well has Your System been Running for the Past Year?.....	2-19
Why You Need Performance Data Collection and Reporting (DBQL) .....	2-20
PDCR Purpose .....	2-21
PDCR Solution Components .....	2-22
PDCR Database Structure.....	2-23
PDCR Base Structures .....	2-24
PDCR Access and User Logons .....	2-25
Common DBC and PDCR View.....	2-26
PDCR Data Collection Schedule .....	2-27
Demo: Performance Data Collection Portlet – Credentials .....	2-28
Demo: Performance Data Collection Portlet – Jobs .....	2-29
Demo: Performance Data Collection Portlet .....	2-30
Demo: Alert Presets .....	2-31
Demo: Query Log Portlet.....	2-32
Demo: Application Queries Portlet.....	2-34
Performance Data Collection Portlet – Data Retention .....	2-35
Performance Data Collection Portlet – Adding Alerts .....	2-36
Proposed Alert Texts.....	2-37
Performance Data Collection Portlet – Job Run Details.....	2-38
Performance Data Collection Portlet – Job Drill Down .....	2-39
Performance Data Collection Portlet – View Job History .....	2-41
Summary .....	2-42

## Module 3 – Workload Management

Objectives .....	3-2
Topics.....	3-3
Current Topic – Workload Management Overview .....	3-4
Limited System Resource Challenge .....	3-5
Why Workload Management.....	3-6
What is a Workload?.....	3-7
What is Workload Management?.....	3-8
Workload Management Offerings .....	3-9
Current Topic – Mixed Workload Features .....	3-10
Virtual Partitions .....	3-11
Control Groups and Workloads .....	3-12
Workload Classification Criteria Types.....	3-13
Workload Classification.....	3-14
Pre-Execution Controls – Filters.....	3-15
Pre-Execution Controls – Arrival Rate Meter .....	3-16
Pre-Execution Controls – Throttles .....	3-17
State Matrix.....	3-18
During Execution Controls – Exceptions .....	3-19
Exception Actions .....	3-20
Workload Management Levels .....	3-21
Example: Query Management Architecture .....	3-22
Example: Workloads and Rules.....	3-23
Workload Management Offering Comparison .....	3-24
Current Topic – Viewpoint Portlets.....	3-25
Workload Management – Administration .....	3-26
Workload Designer – Workloads.....	3-27
Workload Designer Example.....	3-28
Workload Management – Monitoring and Reporting.....	3-29
Summary .....	3-30
Module 3: Demo and Labs.....	3-31

## Module 4 – AutoStats

Objectives .....	4-2
Why Automated Statistics.....	4-3
What is Automated Statistics .....	4-4
Automated Statistics: Features and Limitations .....	4-5
AutoStats Flow.....	4-6
AutoStats Concepts.....	4-7
AutoStats Concepts: DBQL/PDCR .....	4-8
AutoStats Approaches.....	4-9
AutoStats Recommendations .....	4-10
Summary .....	4-11
Lab Overview.....	4-12

## Module 5 – Backup and Recovery

Objectives .....	5-2
Why Backup and Recovery?.....	5-3
BAR Solutions .....	5-4
Archive and Recovery Utility (ARC) .....	5-5
Why Data Stream Architecture (DSA)? .....	5-6
Introduction to Data Stream Architecture.....	5-7
Components of a DSA Solution.....	5-8
DSA Components and Interfaces.....	5-9
Terminology – DSC Setup BAR Setup Portlet.....	5-10
BAR Setup Discover DSC Servers .....	5-11
BAR Setup Configure Nodes Under DSC Server.....	5-12
BAR Setup Configure Media Servers.....	5-13
BAR Setup Configure Fabrics .....	5-14
BAR Setup Configure Accounts.....	5-15
BAR Setup Configure Target Groups.....	5-16
BAR Operations – Check Backup Job Status.....	5-17
Verify AWS S3 Bucket for Backup Confirmation .....	5-18
Backup Options.....	5-19
Snapshot or Backup for Cloud Data Protection? .....	5-20
Cloud Business Continuity and Recovery Decision Matrix .....	5-21
BAR Job Workflow .....	5-22
Running a Backup Job .....	5-23
BAR Operations Portlet Create NEW JOB .....	5-24
Summary .....	5-25
Demo/Lab Overview.....	5-26

## Module 6 – MAPS

Objectives .....	6-2
Why MAPS.....	6-3
MAPS Benefits .....	6-4
MAPS Multiple Reconfigurations .....	6-5
MAPS Data Distribution.....	6-6
System Expansion and Table Migration .....	6-7
Move Tables: Process Flow .....	6-8
What is TDMaps? .....	6-9
System Defined Maps.....	6-10
Example 1: Query Behavior (1 of 2): Performing a Join with Tables in Different Maps .....	6-11
Example 1: Query Behavior (2 of 2): Performing a Join with Tables in Different Maps .....	6-12
Example 2: Primary Index Join: Tables in Different Maps .....	6-13
Small Tables.....	6-14
Small Tables Single AMP MAP .....	6-15
Sparse Maps .....	6-16
Sparse Maps: System Determines Which AMPs to Use .....	6-17
Sparse Maps – COLOCATION .....	6-18

Colocation Name .....	6-19
Foreign Tables .....	6-20
Moving Tables – New Map .....	6-21
ALTER TABLE – Possible Issues.....	6-23
GRANT MAP .....	6-24
Miscellaneous SQL and Hash Functions .....	6-25
Summary .....	6-26

## Module 7 – Viewpoint Overview

Objectives .....	7-2
Topics.....	7-3
Current Topic – Viewpoint Portal Basics .....	7-4
Teradata Viewpoint.....	7-5
Viewpoint Portal and Portlets .....	7-6
What it Looks Like After First Time Login.....	7-7
Add Portlets to the Current Page .....	7-8
Portlet Controls .....	7-9
Viewpoint Rewind .....	7-10
The Viewpoint Dashboard .....	7-11
Current Topic – Administration Portlets.....	7-12
Administration Portlets .....	7-13
Monitored Systems Portlet.....	7-14
Monitored Systems Portlet – General .....	7-15
Monitored Systems Portlet – Data Collectors.....	7-16
Monitored Systems Portlet – Monitor Rates.....	7-17
Monitored Systems Portlet – System Health .....	7-18
Current Topic – System and Query Monitoring Portlets .....	7-19
System Health Portlet .....	7-20
Metric Heatmap Portlet.....	7-21
Query Monitoring Portlets .....	7-22
Query Monitor Portlet – Summary View.....	7-23
Query Monitor Portlet – Session Details: Overview .....	7-24
Query Monitor Portlet – Session Details: Explain.....	7-25
My Queries Portlet.....	7-26
Query Groups Portlet .....	7-27
Query Spotlight Portlet .....	7-28
Summary .....	7-29
Lab 1: Viewpoint Overview.....	7-30
Lab 2: Rewind.....	7-31

## Module 8 – GSA, SZ, Restarts

Objectives .....	8-2
Why Global Space Accounting.....	8-3
Global Space Accounting Example .....	8-4
Why Global Space Accounting.....	8-5
Global Space Accounting .....	8-6
Global Space Accounting Example .....	8-7
Global Space Accounting Monitoring .....	8-8
Global Space Accounting Dictionary Tables.....	8-9
Why Teradata Secure Zones .....	8-10
Teradata Secure Zones.....	8-11
Teradata Secure Zones – Observations.....	8-12
Teradata System Emulation Tool (SET).....	8-13
Scheduled Restarts .....	8-14
Unscheduled Restarts: Drive Failures.....	8-16
Unscheduled Restarts: BYNET Failures .....	8-17
Unscheduled Restarts: Node, VPROC s/w Failures .....	8-18
Summary .....	8-19

## Appendix A: Class Instructions

The slide features a teal-colored curved shape on the left side, containing the Teradata logo and a silhouette of a person looking up. The background is a dark grey-blue.

**teradata.**

## **Module 0: Overview & Basics Recap**

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Course Description

The primary focus of this 5-day VILT course is to teach you about Teradata database administration intermediate.

The major topics covered in this course include:

1. Recap of Vantage Administration Basics
2. Identify various types, and features of logging
3. Explain the purpose for using the Database Query Log (DBQL) and Performance Data Collection and Reporting (PDCR)
4. Define Workload Management and explain concepts and features
5. Explain why Automated Statistics along with features and limitations
6. Explain why BAR (Backup, Archive & Recovery), DSA, and DSC Architecture
7. Why TDMaps and the various types of Maps (Contiguous and Sparse)
8. Describe the components of the Teradata Viewpoint, navigation of portal pages and portlets
9. Describe Global Space Accounting usage and benefits

## Who Should Attend and Prerequisites

### Who Should Attend

This course is designed for ...

- Teradata Professional Services Consultants
- Channel Partners
- Customers

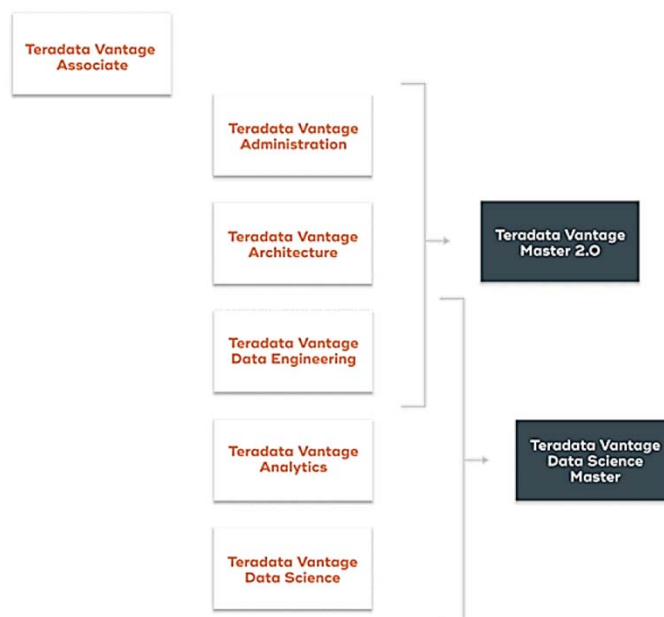
### Prerequisites

Required

- Teradata Vantage Masterclass → Course | ID: 0000095044
- Teradata Vantage Administration Basics → Teradata University: Curriculum | ID: 0000095918



# Teradata Certification Exams



Depending upon the tests that are completed, you can earn various Teradata Certified designations. The Teradata Certification exams require knowledge plus experience with Teradata release 16 and higher. This manual will help you prepare for these Teradata tests, but many of the test questions are scenario-based and Teradata experience is needed to answer these types of questions. The websites listed below provide much more information. The first website listed below can be used to understand the concepts and objectives that are on each exam. These objectives will give you a better idea of what to study.

Teradata Certification (TCPP) Website: [www.teradata.com/University/Certification](http://www.teradata.com/University/Certification)

Bypass Exams\_with\_Prior\_Teradata\_14\_Certification information:  
[www.teradata.com/University/Certification/Vantage-Certifications#BypassExams](http://www.teradata.com/University/Certification/Vantage-Certifications#BypassExams)

Registration Information: [www.pearsonvue.com/teradata](http://www.pearsonvue.com/teradata)

TCCP Contact Information: email: [certification.Teradata@Teradata.com](mailto:certification.Teradata@Teradata.com)  
 phone: 1-800-845-2273

Disclaimer: The Teradata Certification tests include questions from a mix of sources and require experience, especially on tests other than the Associate Exam.

## Objectives

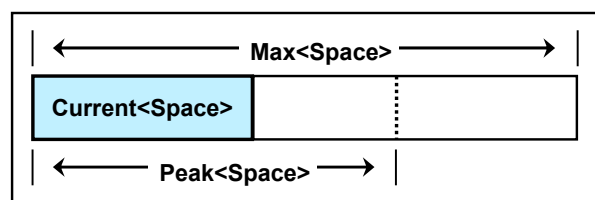
After completing this module, you will be able to:

- Explain key metrics related to Space Management
- Understand the difference in space management between creator and owner
- Impact of GIVE statement
- How Account Strings impact workload management and billing metrics
- Impact of Profile on Users
- Explain different types of Access Rights
- What is a Role and how does it get validated
- What constitutes Node Memory
- What is TIM and how it interacts with FSG Cache



# Space Metrics

Space	Space = Permanent	Space = Spool	Space = Temporary
Max<Space>	The maximum number of bytes available for table, index, permanent journal, stored procedure, and UDF storage in a database or user	A value used to limit the number of bytes the system will consume to create spool files (including volatile temporary tables) for a user	A value used to limit the number of bytes the system will use to store data for global temporary tables for a user
Current<Space>	The total number of bytes in use to store the tables, indexes, permanent journal, stored procedures, and UDFs contained in the database or user	The number of bytes currently in use for running transactions and/or volatile temporary tables	The number of bytes in use for global temporary tables
Peak<Space>	The largest number of bytes actually used to store data in this database/user since the value was last reset	The maximum number of spool bytes used for this user since the value was last reset	The maximum number of bytes used by global temporary tables for a user since the value was last reset



## Permanent Space

**MaxPerm** is the maximum number of bytes available for a table, subtables, join indexes, hash indexes, etc. in a system database or user. The number of bytes specified is divided by the number of AMP vprocs in the system. The result is recorded on each AMP vproc and may not be exceeded on that vproc.

**CurrentPerm** is the total number of bytes (including table headers) in use on the database to store the tables, subtables, etc. in a database or user. This value is maintained on each AMP vproc.

**PeakPerm** is the largest number of bytes ever actually used to store data in a database or user. This value is maintained on each AMP vproc.

## Spool Space

**MaxSpool** is a value used to limit the number of bytes the system will allocate to create spool files for a user.

One should limit the spool space which is allocated to reduce the impact of "runaway" transactions, such as accidental product joins.

**CurrentSpool** is the number of bytes in use for running transactions. This value is maintained on each AMP vproc for each user.

**PeakSpool** is the maximum number of bytes used by a transaction run for a user, since the value was last reset, by the ClearPeakDisk Macro (supplied in system user DBC).

## Temp Space

**MaxTemp** is a value used to limit the number of bytes the system will use to store data for global temporary tables for a user.

**CurrentTemp** is the number of bytes in use for global temporary tables. This value is maintained on each AMP vproc for each user.

**PeakTemp** is the maximum number of bytes used by global temporary tables for a user since the value was last reset by the ClearPeakDisk Macro (supplied in system user DBC).

## Questions

Q1 Why these metrics are important?.

A1 These are important since it helps to understand the current usage resulting in optimal space allocation. Let's say, the Peak Perm space usage is 70% of the allocated (max perm) space. If the peak perm over a certain period of time, does not exceed this 70% then, it leads to the question of whether additional space i.e., max perm was allocation and can this unutilized space be given to another database/user.

Q2: This brings to another question of how to know this peak perm parameter reflects the scenario in last 'x' months, 'y' weeks or 'z' years?.

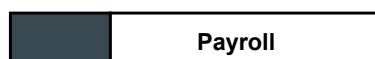
A2: To answer this question there is a macro to reset this PeakPerm value which is used to reset this after a certain period of time based on the customer requirements which would help in periodic monitoring and space assessment.

Q3: How to Reset Peak values

A3: The macro is called as the ClearPeakDisk Macro supplied in the user DBC.

Note: Space limits are enforced at the database/user level.

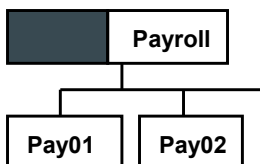
## Assigning Perm and Spool Limits



MaxPerm = 25E9, MaxSpool = 50E9 (via a Profile)

Payroll creates the following two users as children of itself.

```
CREATE USER Pay01 AS PASSWORD = abc123, PERM = 6E9, SPOOL = 35E9;
CREATE USER Pay02 AS PASSWORD = xyz456, PERM = 4E9;
```



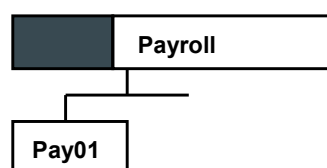
MaxPerm = 15E9

MaxSpool = 50E9

The maximum Spool value may not exceed that of the creator when a new user/database is created.



What is the maximum Spool limit of Pay02?



MaxPerm = 19E9

MaxSpool = 50E9

The Perm space from Pay02 is returned back to the immediate owner.

You can only define permanent and spool space limits at the database or user level, not at the table level.

In this example, Payroll has a Permanent space of 25GB and a Spool Space of 50GB. Payroll is creating two users and allocating permanent space and spool space for one of them. When you create databases or users, perm space limits are deducted from the available (unused) space of the immediate owner.

In this example the creator (Payroll) is assigned a SPOOL limit via a Profile, then the spool space limit cannot exceed that of the creator at the time you create a database or user.

The SPOOL and TEMPORARY limits for a new user are determined by the following order of preference (top to bottom).

- SPOOL limit in the PROFILE identified in the CREATE USER statement.
- SPOOL limit explicitly stated in the CREATE USER statement.
- SPOOL limit in the PROFILE of the CREATOR.

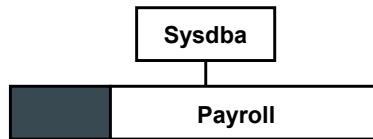
If the creator is not assigned to a PROFILE, then the SPOOL limit is limited to the immediate OWNER of the newly created user.

In this example

There is no statement of spool space for Pay02, its maximum Spool defaults to the limit of its creator which is 50 GB since the creator is assigned to a PROFILE.

The amount of maximum Perm increases and decreases as the owner creates and drops new users. The spool space figure remains constant even when the owner adds and drops users.

## Assigning Perm and Spool Limits (cont.)

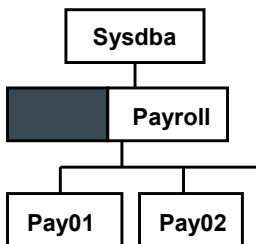


MaxPerm = 10E9, MaxSpool = 100E9 (via a Profile)

MaxPerm = 25E9, MaxSpool = 50E9 (via a Profile)

Sysdba creates the following two users as children of Payroll.

```
CREATE USER Pay01 FROM Payroll AS PASSWORD = abc123, PERM = 6E9, SPOOL = 75E9;
CREATE USER Pay02 FROM Payroll AS PASSWORD = xyz456, PERM = 4E9;
```



The Perm space for Pay01 and Pay02 is deducted from which user?

What is the maximum Spool limit of Pay01?

What is the maximum Spool limit of Pay02?

In this example, there are two users Payroll and Sysdba

A user, Payroll, has a 25 GB permanent space limit and a 50 GB spool space limit. Sysdba has a 100 GB spool space limit via a Profile.

Sysdba creates two new users, Pay01 and Pay02, as children of Payroll. After Sysdba creates the new users, Payroll's maximum Perm space drops to 15 GB because Pay01 is assigned 6 GB of maximum Perm space and Pay02 is assigned 4 GB of maximum Perm space.

Now Sysdba has a limit of 100 GB of maximum Spool (via a Profile), therefore, the Sysdba user can create new users and/or databases with a maximum of 100 GB of spool. When Sysdba creates user Pay01, it assigns 75 GB of maximum Spool which is more than Payroll – this is allowed.

Since there is no specification of spool space for Pay02, its maximum Spool defaults to the limit of its creator which is 100 GB.

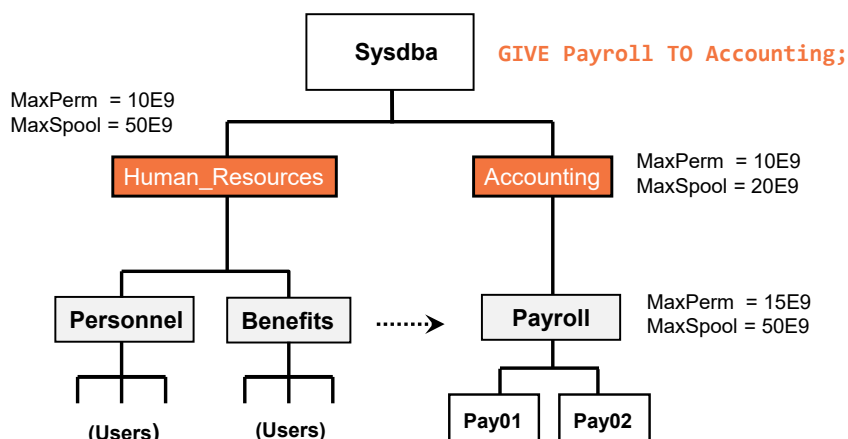
### Questions

Q1 Which are two situation when a child user/database can have more Spool and/or Temporary space than their immediate parent.

A1: When the creator has space allocated via Profile and creates users. The other example is when a user/database is moved (GIVE command) in the system.

## Giving One User to Another

- Payroll ownership is transferred (GIVE) to Accounting.
- All descendants (child users/databases, tables, views, etc.) of a “given” object remain with the “given” object.
- The GIVE command may also be used to move Permanent space from one database/user to another database/user.



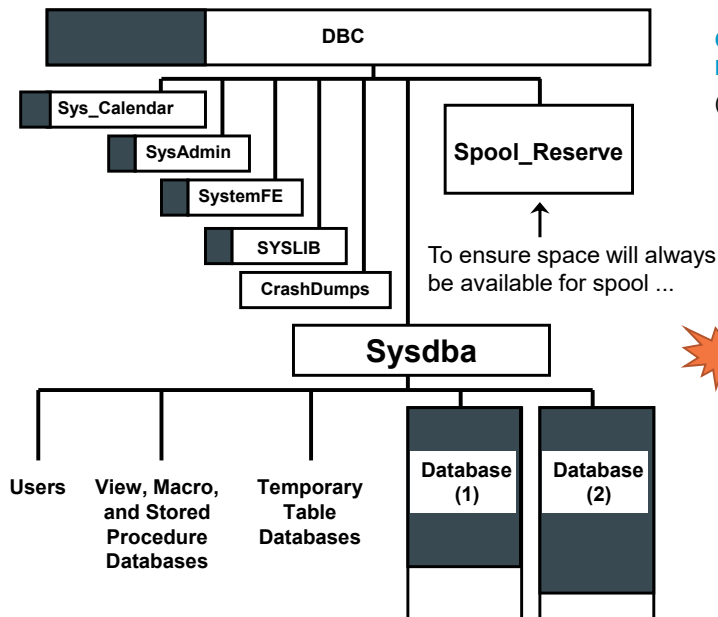
When you give an object to another object in the hierarchy, all space allocated to that object goes with it. If you drop the object, its perm space is credited to its immediate owner. When you give databases or users, all descendants of the given object remain descendants of the given object.

When you give a database/user to a new owner, the transferred database/owner retains the space limits assigned to them. The only change in the data dictionary for the transferred database/user is the immediate owner's new name.

However, If you drop a database/user, its perm space limit is credited to its immediate owner.

This example shows that using GIVE Statement how a database/user is moved from one database/user to another i.e., Payroll from Human\_Resources to Accounting.

## Reserving Space for Spool



```
CREATE DATABASE Spool_Resume AS
PERM = 40E9 *(HASHAMP()+1);
```

(This example reserves ~ 40 GB of space for spool on each AMP.)



Do not use the Spool\_Resume database to store tables.

User data tables/indexes may occupy up to 50 – 70% of the total disk space.

Remember that DBC tables, WAL, etc. also need space within the system.

Spool space serves as temporary storage for returned rows during transactions that users submit. To ensure that space is always available, you may want to set aside about 20 to 25% of total available space as spool space. To do this, you can create a special database called Spool\_Resume.

Decision support applications should reserve more of the total disk space as reserved spool space since their SQL statements generate larger spool files. Tactical queries/applications often use less spool space because their statements generate smaller spool files.

The above actions guarantee that data tables will never occupy more than 50% to 70% of the total disk space. There is no data stored in Spool\_Resume; therefore, this is space available for spool or temp.



## Views for Space Allocation Reporting

- **DBC.DiskSpace[V][X]**
  - AMP vproc information about disk space usage (including spool) for any database or account.
  - For the example it would show **Table All** – 100 MB

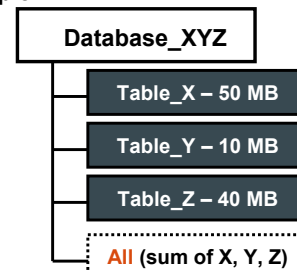
- **DBC.TableSize[V][X]**

- AMP vproc information about disk space usage (excluding spool) for any database, table or account
- For the example it would show
  - **Table\_X** – 50 MB
  - **Table\_Y** – 10 MB
  - **Table\_Z** – 40 MB

- **DBC.AllSpace[V][X]**

- AMP vproc information about disk space usage (including spool) for any database, table, or account.
- For the example it would show
  - **Table\_X** – 50 MB
  - **Table\_Y** – 10 MB
  - **Table\_Z** – 40 MB
  - **Table All** – 100 MB  
200 MB

Example:



Use the following system views to report current space allocation:

### DBC.DiskSpace[V][X]

This view gives AMP vproc information about disk space usage for any database or account. It gets this information from the ALL table.

### DBC.TableSize[V][X]

This view gives AMP vproc information about disk space usage (excluding spool) for any table or account.

### DBC.AllSpace[V][X]

This view gives AMP vproc information about disk space usage (including spool) for any database, table, or account.

Each of these views references the non-hashed DBC.DataBaseSpace table.

## Creating and Using Account IDs

<b>CREATE USER tfact07</b>	Username
<b>FROM Sysdba</b>	Name of the immediate owner in the hierarchy
<b>AS PERM = 1E9 *(HASHAMP()+1)</b>	Amount of Permanent space
<b>,SPOOL = 3E9 *(HASHAMP()+1)</b>	Maximum amount of Spool space
<b>,PASSWORD = Secure12</b>	Initial password
<b>,ACCOUNT = (</b>	
<b>'\$M_9038_&amp;D&amp;H',</b>	Default account – medium priority
<b>'\$H00EDUC&amp;D&amp;H',</b>	Opt. Account – high priority
<b>'\$M1\$LOAD&amp;D&amp;H',</b>	Opt. Account – medium priority
<b>'\$M_9038'</b>	Opt. account – medium priority
<b>);</b>	

A logon can optionally include an account ID; the first account ID is the default account ID.

- **\$xxxWORK&D&H** is the recommended account format where WORK is a 4-character workload type starting in the 5<sup>th</sup> position of an account id.
  - \$L (low), \$M (medium), \$H(high), and \$R(top/rush)
  - You may add the following substitution variables to a user's account string. The system resolves the variables at logon or at SQL statement execution time. Example:
    - **&D**                      Date (YYMMDD)
    - **&H**                      Hour (HH)

When you create a user, you can specify one or more account IDs that a new user can specify.

Account Strings may be used to track system CPU, I/O usage, or space usage. When the user logs on, the user can specify a valid account ID, or let the first account ID in the user row (from CREATE or MODIFY USER) become the default. You should determine an account ID scheme for ease of accounting and priorities.

In this example there are 4 Account IDs mentioned. Account IDs may begin with the characters \$L, \$M, \$H, or \$R to identify the priorities low, medium, high, and rush, respectively. The first Account ID becomes the default Account ID for the user.

### Using Account IDs with Logon

All logons require an Account ID. A user can submit an explicit account ID by including it in the logon string. It must be a valid ID specified in the last CREATE or MODIFY USER statement. If no ID is specified in the CREATE or MODIFY statements, it defaults to the ID of the immediate owner's database.

The Account ID's can be used to classify queries within the workload and accordingly, resources can be allocated. The Account ID's can also be used to group queries within the DBQL and identify cpu, i/o consumption based on the 4 character workload type for any chargeback mechanism

## Account String Expansion (ASE)

- ASE is used to provide more detailed utilization reports and user accounting data.
  - ASE increases the granularity of information returned with the DBC.AMPUsageV.
  - For queries that span multiple hours, the time will be accumulated in its entirety to the query's start hour.
- You may add the following substitution variables to a user's account string.
  - The system resolves the variables at logon or at SQL statement execution time.
  - |    |  |                              |
|----|--|------------------------------|
| &S | Session number                               | (SSSSSSSSS)                  |
| &D | Date   | (YYMMDD)                     |
| &H | Hour   | (HH)                         |
| &I | Logon hostid, session number, request number | (LLLL SSSSSSSSS RRR RRR RRR) |
| &L | Logon timestamp                              | (YYMMDDHHMISS.hh)            |
| &T | Time   | (HHMISS)                     |
- **\$xxxWORK&D&H** is the recommended account format where WORK is a 4-character workload type starting in the 5<sup>th</sup> position of an account id.
  - For \$L, \$M, \$H, and \$R in Resource Partition 0, then examples are:  
\$M00EDUC&D&H or \$M00TACT&D&H (the 00 is simply a placeholder)
  - If a two-character performance group name like M1 (or MD) is used, then examples are:  
\$M1\$TACT&D&H or \$MD\$LOAD&D&H

Account String Expansion (ASE) is an optional feature that enables you to use substitution variables in the account ID portion of the user's logon string. These variables enable you to include date and time information in the string. The variables are resolved at logon time or at actual SQL execution time. Account strings can be up to 128 characters. If, as a result of string expansion, you generate a string longer than 127 characters, the system truncates all characters to the right of position 127. Separation characters, such as colons in time fields and slashes in dates, are included in the character count.

### ASE Variables:

- &L The logon time stamp variable causes the logon time stamp to be inserted into the account string. The full logon time stamp consists of 15 characters. The value inserted into AMPUsage is established at logon time.
- &D The date variable causes the date to be inserted into the account string.
- &T The time variable inserts the time of day into the account string.
- &H The hour variable inserts the hour of the day into the account string. The inserted value consists of two characters and becomes truncated if you place &H to the right of position 29.
- &I The logon host ID/session number/request number variable inserts the logon host ID, the session number and the request number into the account string.
- &S The session number variable inserts the current session number into the account string.

The recommendation is to just use &D&H

## ASE Accounting Example

- Impact of using ASE variables

ASE Variable	Performance Impact	Data Capacity Impact
none	Negligible	1 row per account per AMP.
&D	Negligible	1 row per account per day per AMP.
&H	Negligible	1 row per account per hour per AMP (all days go into 1 hour).
&D&H	Negligible	1 row per account per hour per day per AMP.
&S&D&H	Negligible	1 row per account per session per hour per day per AMP.
&L	Negligible	1 row per logon (LAN) or session pool.
&T	Potentially Non-negligible	1 row per query per AMP.

- Perform the following tasks to extract accounting information:

Step 1. Create AMPUsageSum table.

Step 2. Populate AMPUsageSum table.

Step 3. Create Usage view.

Step 4. Create billing and resource usage reports.

### Example

Let's take an example of two existing users TFACT01 and TFACT02 being logged onto the system using an account string defined as &D&H.

The DBC.Acctg table contains a number of rows generated by the ASE feature and the users have an account string of '\$L00EDUC&D&H'

### Questions

Q1: You need to create a table, view, and a number of reports that provide billing and resource usage information based on the statistics collected by the AMPUsage view.

A1:

- Step 1. Create AmpUsageSum table -- Create a table to hold the collected statistics from the AMPUsage view ( to capture historical information )
- Step 2. Populate AMPUsageSum table -- Build the AmpUsageSum table, and use the INSERT command to populate it with row information from the DBC.AMPUsage view.
- Step 3. Create Usage view -- Use the CREATE statement to combine columns from the DBC.AMPUsage view and DBC.LogOnOff view into the Usage view.
- Step 4. Create billing and resource usage reports -- Once the view is completed, construct SQL statements to SELECT information from the Usage view to create billing and resource usage reports.

## Dynamically Changing an Account ID

- You can change your Account ID without logging off. This may be done to re-prioritize a query. This is also referred to as “nicing a query”.
- You can change Account IDs for the next SQL statement you run, or for all jobs for the remainder of the current session.
- To change Account IDs, use the SET SESSION ACCOUNT statement:

### Syntax:

For the next SQL statement: `SET SESSION ACCOUNT = 'Account_ID' FOR REQUEST;`

For the rest of the current session: `SET SESSION ACCOUNT = 'Account_ID' FOR SESSION;`

### Example:

For the rest of the session: `SET SESSION ACCOUNT = '$H00EDUC&D&H' FOR SESSION;`

Note: You can only use valid account ID's

You can dynamically change your Account ID without logging off and logging back on. One reason you may want to do this is to change your session's priority. This is also called “nicing” a query.

Example -- You could nice a query to a higher priority to run a business-critical job sooner than under its originally defined priority.

Use the SET SESSION ACCOUNT statement to change your performance group (account priority) for the next SQL query you run, or for all jobs for the remainder of the current session.

### Question

Q1 - Can I mention any Account ID and ‘nice’ my query?

A1 - The answer is no. The user can change the Account ID's from the list of Account ID's defined and not any random Account ID.

## Example – AMPUsage View

View CPU time and I/O totals for a single user.

```
SELECT      UserName (CHAR(10))
           ,Substring (AccountName FROM 1 FOR 8) AS "Acct_Str"
           ,Substring (AccountName FROM 9 FOR 6) AS "Date"
           ,Substring (AccountName FROM 15 FOR 2) AS "Hour"
           ,SUM (CPUTime) AS "CPU Time"
           ,SUM (DiskIO) AS "Disk IO"
FROM        DBC.AMPUsageV
WHERE       UserName = 'AdhocUsr'
GROUP BY    1, 2, 3, 4
ORDER BY    5 DESC;
```

- The Account ID is \$M00EDUC&D&H

<u>UserName</u>	<u>Acct Str</u>	<u>Date</u>	<u>Hour</u>	<u>CPU Time</u>	<u>Disk IO</u>
AdhocUsr	\$M00EDUC	180115	09	53.97	1,894,531
AdhocUsr	\$M00EDUC	180115	11	28.95	913,708
AdhocUsr	\$M00EDUC	180112	14	19.04	912,218
AdhocUsr	\$M00EDUC	180116	19	8.96	223,322

- To Reset counters for
  - ALL Rows / Selected Rows
  - Use the [DBC.ClearAccounting](#) macro
  - REPLACE MACRO [DBC.ClearAccounting](#) AS ( UPDATE Acctg SET CPU = 0, IO = 0 ALL; );

The SQL statement on this slide requests totals for CPU time and I/O for a user named student140. The totals are aggregates of all resources used across all AMP vprocs.

Note that the account ID has been expanded and the SUBSTRING function is used to display the expanded account string in a readable format.

## System Accounting Views

### ➤ DBC.AccountInfo[V][X]

- Returns each Account Name (Account ID) associated with a user (for users the requestor owns).
- Provides information about each user and the valid account codes associated with each. (X views – lists users and accounts the requestor owns or has modify rights to)

### ➤ DBC.AMPUsage[V][X]

- Provides information about I/O and AMP CPU usage by user and account
- AMPUsage views are views that use the DBC.Acctg table to provide accounting information by username and account.
- These views can be used to determine which users are consuming CPU and/or I/O resources.
- Information related to
  - CPUTime → Total number of AMP CPU seconds used (increments of 1/100 second).
  - DiskIO → Total number of logical disk I/O operations.
  - etc.,

### DBC.AccountInfo[V][X]

This view provide information about each user and the valid account codes associated with each user. When the requesting user indicates the [X] view, they can only see information about users that they own or have modify rights on.

The UserOrProfile column indicates whether the user is an actual user or a profile.

### DBC.AmpUsage[V][X]

This views use the underlying DBC.Acctg table to provide accounting information by username and account. This view provides CPU activity and logical I/O counts explicitly requested by the following two sources:

#### AMP database software

File system that is running in the context of an AMP worker task

This view can be used to determine which user or users are consuming CPU and I/O resources on a system.

The system requests I/Os to execute a step in the user's query. The DBC.AmpUsage views do not include I/Os the operating system performs for swapping or I/Os caused by parsing the user's query. The system charges a logical I/O even if the segment you request is cached and no physical I/O is done.

# Profiles

What access rights are used to support profiles?

- CREATE PROFILE – needed to create new profiles
- DROP PROFILE – needed to modify and drop profiles

Who is allowed to create and modify profiles?

- Initially, only DBC has the CREATE PROFILE and DROP PROFILE access rights.
  - As DBC, give the “profile” access rights to the database administrators (e.g., Sysdba).
- **GRANT CREATE PROFILE, DROP PROFILE TO Sysdba WITH GRANT OPTION;**

How are users associated with a profile?

- The CREATE PROFILE command is used to create a profile of desired attributes.
- **CREATE PROFILE Employee\_P AS ... ;**

The PROFILE option is used with CREATE USER and MODIFY USER commands to assign a user to a specific profile.

- **CREATE USER Emp01 AS ..., PROFILE = Employee\_P;**
- **MODIFY USER Emp02 AS PROFILE = Employee\_P;**

Profiles enable you to manage the following common parameters:

- Account strings, including ASE codes
- Default database
- Spool space
- Temporary space
- Query Band (15.10)
- Default Map (16.10)
- Password attributes, including:
  - Expiration
  - Composition (length, digits, and special characters)
  - Allowable logon attempts
  - Duration of user lockout (indefinite or elapsed time)
  - Reuse of passwords

The keyword PROFILE will give both the CREATE PROFILE and DROP PROFILE access rights → **GRANT PROFILE TO SYSDBA WITH GRANT OPTION;**

Note: The PROFILE privileges can only be granted to a user and not to a role or database. Permanent space needs to be defined as mandatory when creating a user.



## Impact of Profiles on Users

Assignment of a profile to a group of users is a way of ensuring that all members of a group operate with a common set of parameters

- Profile definitions apply to every assigned user, overriding specifications at the system or user level
- Any profile parameter can be NULL or NONE

All members inherit changed profile parameters and the impact on current users is as follows

- SPOOL and TEMPORARY space allocations are imposed immediately
- Password attributes take effect upon the next logon
- Database, Account IDs, and Query Band are considered at the next logon unless the member submits a SET statement

Order of Precedence for parameters:

1. Specify database, query band, or account ID at the session level
2. Specified parameters in a Profile
3. CREATE USER or MODIFY USER statements

All members inherit changed profile parameters. The impact is immediate, or in response to a SET SESSION statement, or upon next logon, depending on the parameter:

SPOOL and TEMP space allocations are imposed immediately. This will affect the current session of any member who is logged on at the time his or her user definition is modified.

Password attributes take effect upon next logon.

Account IDs, Query Band and a default database are considered at next logon unless the member submits a SET statement. For SET SESSION ACCOUNT case the account ID must agree with the assigned profile definition.

### Order of Precedence

With profiles, there are 3 ways of setting accounts and default database (2 for query band). The order of precedence (from high to low) is as follows:

The DATABASE statement is used to set the current default database or the SET SESSION ACCOUNT is used to set the account ID. However, a user can only specify a valid account ID. SET QUERY\_BAND can be used to specify the query band for the session or transaction. Note: If the values in the profile query band are not default, the transaction or session query band cannot contain the same names as the profile query band. Basically, these values cannot be changed (restricted).

Specify them in a profile and assign the profile to a user.

Specify accounts or default database through the CREATE USER/MODIFY USER statements.

## Example – CREATE PROFILE

```
CREATE PROFILE Employee_P AS
  ACCOUNT          = ('$M00EDUC&D&H', '$L00EDUC&D&H')
  ,DEFAULT DATABASE = HR_VM
  ,SPOOL           = 10E9
  ,TEMPORARY       = 1E9,
  ,PASSWORD        = (EXPIRE = 90, MINCHAR = 8, MAXLOGONATTEMPTS = 3,
                      LOCKEDUSEREXPIRE = 1440, REUSE = 365,
                      DIGITS = 'R', RESTRICTWORDS = 'Y', SPECCHAR = 'P')
  ,DEFAULT MAP     = TD_Map1
  ,QUERY_BAND      = 'BU=FIN;';

CREATE USER Emp01 AS PERM = 0, PASSWORD = emp01pass,
  PROFILE = Employee_P,
  SPOOL = 20E9,
  ACCOUNT = '$M00FACT&D&H';
```

What is the spool space limit for Emp01?

10E9

What is the default account code for Emp01?

\$M00EDUC&D&H

Assume this command is executed: **MODIFY USER Emp01 AS PROFILE = NULL;**

What is the spool space limit for Emp01?

20E9

What is the default account code for Emp01 for the current session?

\$M00EDUC&D&H

What is the default account code for Emp01 for a new session?

\$M00FACT&D&H

This example contains a simple example of creating a profile, assigning it to a user, and then removing it from the user with the MODIFY USER command.

The Spool space limit for Emp01 user is dependent on the profile hence the answer to the 1st question is 10e9.

There are two Account ID's defined for the user Emp01, one of them is via Profile and another is via creating a user. Which one takes precedence? The answer to this question is Profile always takes precedence hence the Account ID's for the user Emp01 is \$M00EDUC&D&H

Now we are modifying the user and mentioning the Profile as NULL. Observe the difference on the same questions. Please do note that the user has not logged off.

As we understood earlier, SPOOL Space gets reflected immediately on change hence now the spool space would be 20e9.

The tricky question is what is the default Account ID? , The answer to this is it continues to remain EDUC with Medium priority. One may ask why? Account ID will be reflected on next logon OR if explicitly set using the SET command. In this example it was not explicitly set hence continues to have value of old Account ID.

Answers to second set of three questions on this slide:

20E9

\$M00EDUC&D&H

\$M00FACT&D&H

## Miscellaneous SQL Functions

Example 1: As Emp01, identify the current user, role, profile, and database information.

```
SELECT USER, ROLE, PROFILE, DATABASE;
```

<u>User</u>	<u>Role</u>	<u>Profile</u>	<u>Database</u>
EMP01	HR_R	EMPLOYEE_P	HR_VM

Example 2: As Emp01, list the profile attributes

```
SELECT * FROM DBC.ProfileInfoVX;
```

<u>ProfileName</u>	<u>DefaultAccount</u>	<u>DefaultDB</u>	<u>SpoolSpace</u>	<u>TempSpace</u>	...
Employee_P	\$M00EDUC&D&H	HR_VM	10000000000	5000000000	...

Example 3: As Emp01, list account information.

```
SELECT * FROM DBC.AccountInfoVX;
```

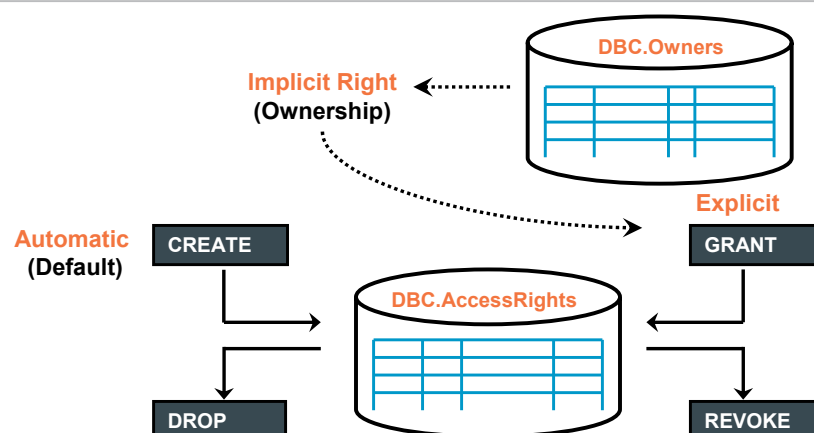
<u>Name</u>	<u>AccountName</u>	<u>UserOrProfile</u>
Employee_P	\$M00EDUC&D&H	Profile
Employee_P	\$L00EDUC&D&H	Profile
Emp01	\$M_&D&H	User

Example 4: As Emp01, list QueryBand information.

```
SELECT GetQueryBand();
```

GetQueryBand() =S> Group=HR; =P> TVSTEMPERATURE=NONE;
--

## Access Rights Mechanisms: Summary



### Inherited Rights

#### Automatic Rights

Automatic rights are privileges given to creators and, in the case of users and databases, their created objects. When a user submits a CREATE statement, new rows are inserted in the DBC.AccessRights table. All rights are automatically removed for an object when it is dropped.

#### Explicit Rights

Explicit rights are privileges conferred by using a GRANT statement. This statement inserts new rows into the DBC.AccessRights table. Explicit rights can be removed using the REVOKE statement.

#### Ownership Rights

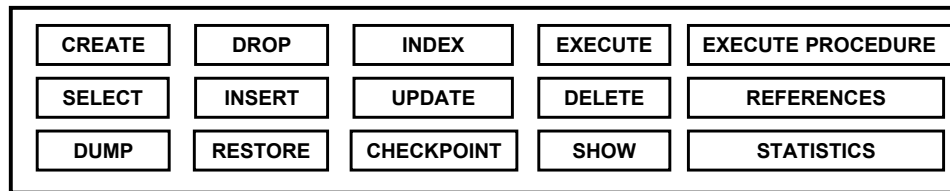
Owners (Parents) have the implicit right to grant rights on any or all of their owned objects (Children), either to themselves or to any other user or database. If an owner grants him or herself rights over any owned object, the parser will validate that GRANT statement even though the owner holds no other privileges. Ownership rights cannot be taken away unless ownership is transferred.

#### Inherited Rights

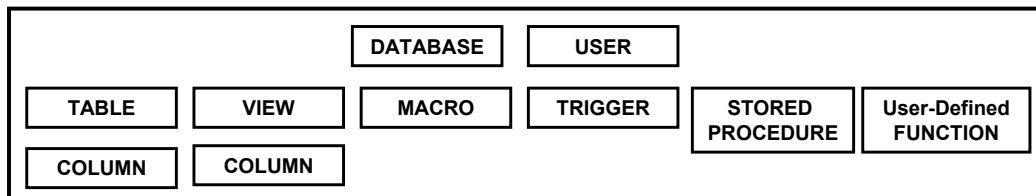
If an Owner (Parents) has the ALL option established, children will inherit the rights associated with the ALL flag. Inherited rights will be discussed in detail later in the module.

# Privileges/Access Rights

A privilege (or access right) is the right of a specific user to perform a specified operation.



On a specified Object



Note: Some access rights don't directly correspond to an SQL statement.

Your privileges (access rights) define the types of activities you can perform during a session. The top box identifies some of the common privileges that a user requires, but this is not an inclusive list of privileges.

Notes:

To use UPDATE or DELETE commands, you must also have the SELECT right on the object.

A column can only be specified with the SELECT, INSERT, UPDATE or REFERENCES access right.

SHOW privilege – this privilege enables you to have access to database object definitions and create text without having access to the data contained by the objects on which the privilege is granted. For example, SHOW permits a user to execute HELP and SHOW requests against an object while at the same time not being able to SELECT from it.

STATISTICS privilege – allows a user to collect or drop statistics on an object (e.g., table). STATISTICS does not grant users the wider capabilities associated with those privileges. STATISTICS can be granted at both the table and database levels.

## CREATE USER – Automatic Rights

By issuing a CREATE USER (or DATABASE) statement, the CREATOR causes Automatic privileges to be generated for both the created user or database AND the creator.

**SYSDBA**



**Accounting**

SYSDBA creates a new user named Accounting.

Both SYSDBA and Accounting are given the following privileges on the Accounting user.

CREATE Table	DROP Table	CREATE View	DROP View
CREATE Macro	DROP Macro	CREATE Trigger	DROP Trigger
SELECT	INSERT	UPDATE	DELETE
EXECUTE	DROP Procedure	DROP Function	DUMP
RESTORE	CHECKPOINT	CREATE Authorization	DROP Authorization
STATISTICS			

The creator (SYSDBA) is also given these 4 additional privileges on the Accounting user.

**CREATE Database**

**DROP Database**

**CREATE User**

**DROP User**

When you create a new user or database, the system automatically generates access rights for the created object and the creator of the object. The system inserts this rights information into the DBC.AccessRights table when you submit a CREATE request. You can remove these rights from the DBC.AccessRights table with the REVOKE statement.

### Example

In this example, user SYSDBA logs on to the system and creates a new user called Accounting.

Both SYSDBA and Accounting have the 21 privileges (shown on this slide) written into the DBC.AccessRights table:

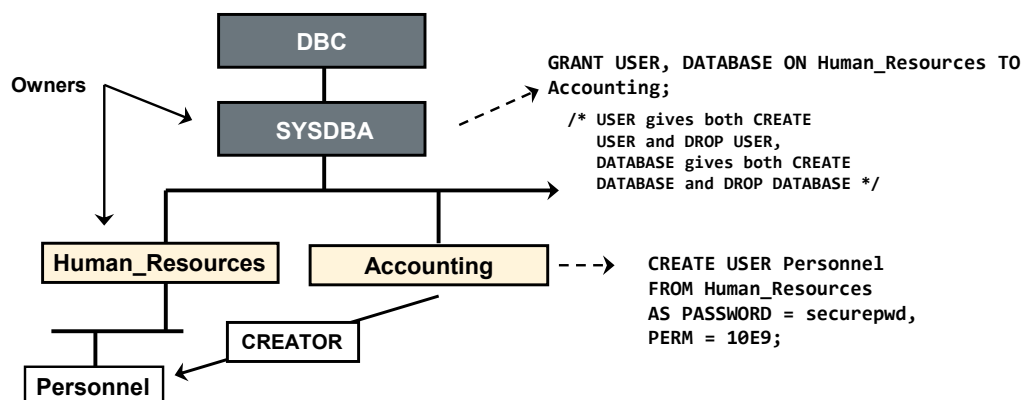
As the creator, SYSDBA is also given these 21 privileges PLUS the 4 following rights on Accounting:

- CREATE Database/User
- DROP Database/User

These access rights that are given to a new user and the creator are the same starting with Teradata 14.0 through Teradata 16.20.

The same number of access rights are also established if you are creating a new database.

## Implicit, Automatic, and Explicit Rights



Where does the 10E9 space for Personnel come from?

How many automatic access rights are created for Personnel?

How many automatic access rights are created for Human\_Resources?

How many automatic access rights are created for Accounting?

Implicit rights belong to the owners of objects. Owners do not require rows in the DBC.AccessRights table to grant privileges on owned objects. Ownership rights cannot be “revoked.” An owner has the implicit right to GRANT privileges over any owned object.

When you submit a CREATE statement, the system automatically adds new rows to the DBC.AccessRights table. You can remove automatic rights with the REVOKE or DROP statements.

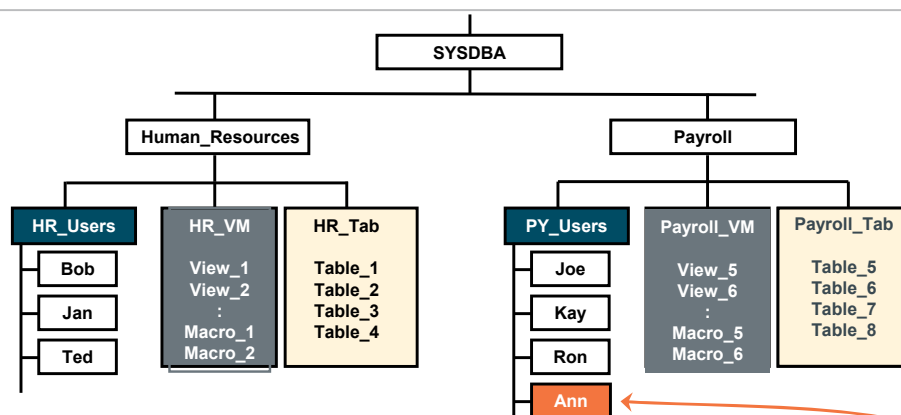
GRANT and REVOKE statements control explicit rights. The GRANT statement adds new rows to the DBC.AccessRights table. The REVOKE statement removes them.

In the example, Accounting is the creator. The system automatically inserts rows for access rights in DBC.AccessRights for the creator (Accounting) and for the created user (Personnel). These rights can be revoked.

The user named Personnel is the created object. The user Personnel automatically receives all but four access rights on itself. These rights are inserted automatically in DBC.AccessRights. These rights can be revoked.

Assume that Human\_Resources is a user and is the immediate owner. The system does not insert any rows in the Data Dictionary for Human Resources. If Human\_Resources is a user, then Human\_Resources has the owner’s implicit right to grant itself rights over Personnel. You cannot revoke the right to GRANT (or re-GRANT) rights over owned objects.

## Inheriting Access Rights



```
GRANT SELECT ON Payroll_Tab TO Payroll_VM WITH GRANT OPTION;
GRANT SELECT, EXECUTE ON Payroll_VM TO ALL PY_Users;
```

```
CREATE USER Ann FROM PY_Users AS PERM = 0, PASSWORD = temp;
```

Ann "inherits" the SELECT and EXECUTE access rights for the database Payroll\_VM.

You may inherit access rights by the placement of your user in the hierarchy. As an administrator, you can set up access rights so that any new object added to an existing user or database inherits specific access rights. Doing so saves time since you do not need to submit a GRANT statement each time you add a new user.

The immediate owner (user or database) of a view or table that is referenced by another must have the right on the referenced object that is specified (SELECT, EXECUTE, etc.) and must have that right with the GRANT option.

### Example

The example here captures a user inheriting access rights.

The user Human\_Resources logs on to the system and grants the SELECT and EXECUTE privileges to user HR\_Users and all of its current and future descendants on the database HR\_VM.

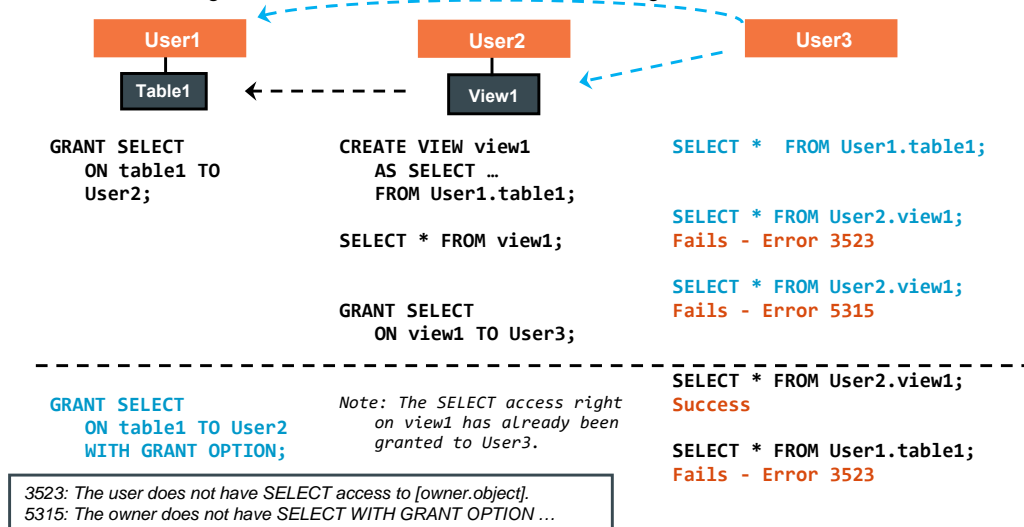
The user Payroll also logs on to the system and grants the SELECT and EXECUTE privileges to user PY\_Users and all of its current and future descendants on the database Payroll\_VM.

Later, Payroll creates a new user called Ann from the space owned by user PY\_Users. Ann inherits the SELECT and EXECUTE privileges on database Payroll\_VM database.



## Access Rights and Views

- View names are fully expanded (resolved) at creation time.
- The system checks access rights at creation time, and validates them again at execution time.



Views may be nested up to 64 levels. View names are fully expanded (resolved) at creation time. This slide shows an example of a nested view.

The system checks access rights at creation time, and validates them again at execution time. Any database referenced by the view requires access rights on all objects accessed by the view.

You can create a view with the intention of read access only, or for controlled UPDATES use. For read access, the SELECT right is needed. For updates, the UPDATE right is needed.

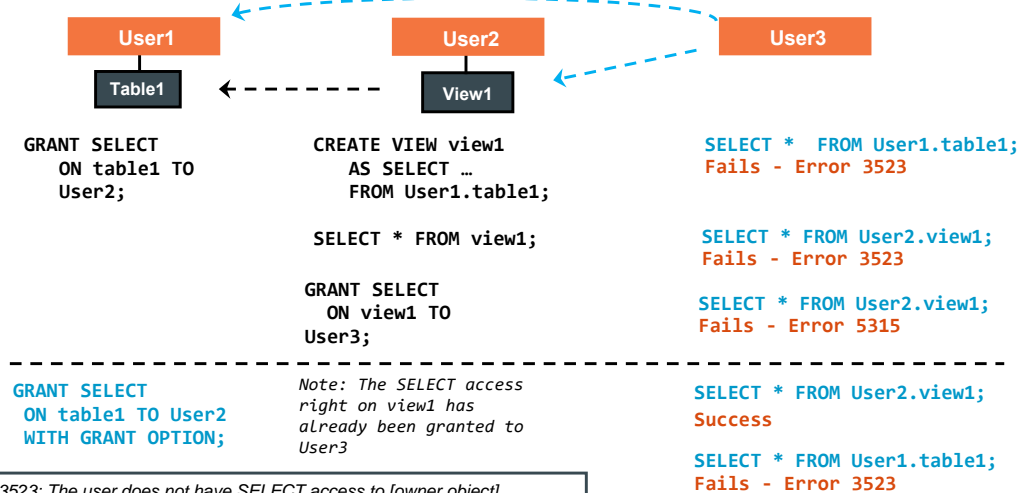
For other users to access a view, the owner must grant the appropriate rights on the view and must have the appropriate rights WITH GRANT OPTION.

The system verifies that the creator has the appropriate right on the objects being referenced when a view is created. It also verifies that the creator has the rights needed to execute the statements defined in a macro. To grant to another user any privilege on a view or macro that references objects owned by a third user, the owner of the view or macro must have the appropriate rights with GRANT OPTION.

Teradata also verifies that the appropriate privileges exist on the objects being referenced for any user who attempts to access a view or execute a macro. This ensures that a change to a referenced object does not result in a violation of access rights when the view or macro referencing that object is invoked.

# Access Rights and Views

- View names are fully expanded (resolved) at creation time.
- The system checks access rights at creation time, and validates them again at execution time.



3523: The user does not have SELECT access to [owner.object].  
5315: The owner does not have SELECT WITH GRANT OPTION ...

## System Views for Access Rights

### ➤ **DBC.AllRights[V][X]**

- Provides information about all rights that have been automatically or explicitly granted
- Lists all rights granted to users in the system

### ➤ **DBC.UserRights[V]**

- AMP Provides information about all rights the user has acquired, either automatically or explicitly
- Lists all rights granted to the current user

### ➤ **DBC.UserGrantedRights[V]**

- Provides information about rights that the current user explicitly has granted to other users

There are three system views you can use to obtain information about access rights. (These views access the DBC.AccessRights table to obtain needed information.) They are:

- DBC.AllRights[V][X]
- DBC.UserRights[V]
- DBC.UserGrantedRights[V]

### **DBC.AllRights[V][X]**

The DBC.AllRights[X] views provide information about all rights that have been automatically or explicitly granted.

### **DBC.UserRights[V]**

This view provides information about all rights that the user has acquired, either automatically or explicitly.

### **DBC.UserGrantedRights[V]**

This view provides information about rights that the current user has explicitly granted to other users.

## What is a Role?

A Role is an administration/security feature that can help simplify the management of access rights.

What is a "role"?

- A role is simply a collection of access rights. Rights are first granted to a role and the role is then granted to users.
- A DBA can create different roles for different job functions and responsibilities.

What are the advantages of using "roles"?

- Simplify access rights management by allowing grants and revokes of multiple rights with one request.
- The number of access rights in the DBC.AccessRights table is reduced.
- Improves performance and reduces dictionary contention for DDL, especially CREATE USER.

A role can be viewed as a pseudo-user with privileges on a number of database objects. A database administrator can create different roles for different job functions and responsibilities, grant specific privileges on database objects to these roles, and then grant these roles to users.

### Advantages of Roles

- **Simplify access rights administration** – a database administrator can grant rights on database objects to a role and have these rights automatically applied to all users assigned to that role. When a user's function within his organization changes, it is easier to change his/her role than deleting old rights and granting new rights that go along with the new function.
- **Reduce disk space usage** – maintaining rights on a role level rather than on an individual level makes the size of the DBC.AccessRights table is much smaller. Instead of inserting one row per user per right on a database object, one row per role per right is placed in the DBC.AccessRights table.
- **Better performance** – roles can improve performance and reduces dictionary contention for DDL. If roles are fully utilized on a system, roles will reduce the size of the AccessRights table and improve the performance of DDL commands that do full-table scans of this table.
  - Faster DROP/DELETE USER/DATABASE, DROP TABLE/VIEW/MACRO due to shorter scans of the DBC.AccessRights table.
  - Faster CREATE USER, DATABASE – remove a copy of hierarchical inherited rights.
  - Less dictionary contention during DDL operations because the commands use less time.

## Access Rights Validation and Roles

The sequence of access to validate roles is

1. If the user has a current role, check the `DBC.AccessRights` table for the required right at the role level.
2. Retrieve all roles nested within the current role from the `DBC.RoleGrants` table. For each nested role, check the `DBC.AccessRights` table for the required right.
3. Check the `DBC.AccessRights` table for the required right at the individual level.
4. Check if the required right is a `PUBLIC` right.

Default sequence of checking access rights is:

- 1) If the user has a current role, search the AccessRights table for RoleId-ObjectId pair entry for the required right.
- 2) If not found, retrieve all roles (e.g., ALL) or nested roles associated within the current role from the RoleGrants table. For each role or nested role, search the AccessRights table for RoleId-ObjectId pair entry for the required right.
- 3) If the access right is not yet found, search the AccessRights table for a UserId-ObjectId pair entry for the required right. In this step, the system will check for rights at the database/user level and at the object (e.g., table, view) level.
- 4) If not yet found, check if the right is a Public right.

## System Views for Roles

Views	Description
DBC.RoleInfo[V][X]	<ul style="list-style-type: none"><li>Provides information about roles</li><li>List role names that exist in the system</li></ul>
DBC.RoleMembers[V][X]	<ul style="list-style-type: none"><li>Provides information about roles and its members</li></ul>
DBC.AllRoleRights[V]	<ul style="list-style-type: none"><li>Lists all rights granted to roles in the system</li></ul>
UserRoleRights[V]	<ul style="list-style-type: none"><li>Lists all rights granted to the enabled roles of the user.</li></ul>

There are three system views you can use to obtain information about access rights. (These views access the DBC.AccessRights table to obtain needed information.) They are:

- DBC.AllRights[V][X]
- DBC.UserRights[V]
- DBC.UserGrantedRights[V]

### DBC.AllRights[V][X]

The DBC.AllRights[X] views provide information about all rights that have been automatically or explicitly granted.

### DBC.UserRights[V]

This view provides information about all rights that the user has acquired, either automatically or explicitly.

### DBC.UserGrantedRights[V]

This view provides information about rights that the current user has explicitly granted to other users.

## Linux Managed Memory and FSG Cache

**Memory managed by Linux** is referred to as “free memory”.

- Teradata Vprocs
  - AMP – includes AMP worker tasks
  - PE – Session control, Parser, Optimizer, Dispatcher
  - PDE (Parallel Database Extensions) – messaging, FSG space management, etc.
  - GTW (Gateway) – Logon Security, Session Context, Connection to Client
  - TVS (Teradata Virtual Storage) – manages Teradata Virtual Storage
  - .....
- Administrative and/or user programs such as:
  - kernel resources and administrative program text and data
  - message buffers (ex., TCP/IP)

**Memory managed by PDE** is called FSG cache.

- When Teradata needs to read a database block, it checks FSG Cache first.
  - Master and Cylinder Indexes
  - Permanent data blocks
  - Spool data blocks
  - Journal blocks; Transient Journal and/or Permanent Journals
  - Synchronized scan (sync scan) data blocks

Memory managed and used by Linux and the vprocs is sometimes called “free memory”. The main code (on a TPA node) that uses free memory is the operating system and Teradata vprocs.

### A brief description of Teradata Vprocs:

- AMP Access module processors perform database functions, such as executing database queries. Each AMP owns a portion of the overall database storage.
- GTW Gateway vprocs provide a socket interface to Teradata on Windows and Linux systems. On MP-RAS systems, the same functionality is provided by gateway software running directly on the system nodes within the PDE vproc.
- Node (or Base) PDE vproc - the node vproc handles PDE and operating system functions not directly related to AMP and PE work. Node vprocs cannot be externally manipulated, and do not appear in the output of the Vproc Manager utility.
- PE Parsing engines perform session control, query parsing, security validation, query optimization, and query dispatch.
- TVS Manages Vantage Advanced SQL Engine storage. AMPs acquire their portions of database storage through the TVS vproc
- Etc.,

When Teradata needs to read a database block, it checks FSG Cache first.

## Node Memory

Example → IntelliFlex node 2 PEs, 36 AMPs (120 AWTs per AMP), 512GB of memory

### Memory

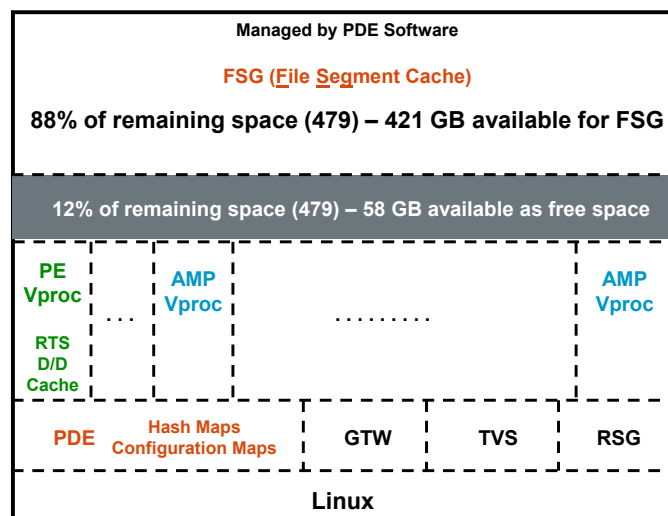
- Linux, Device Drivers, and space for vprocs ≈ 33 GB

	512 GB
-	33 GB
	479 GB

- FSG Cache 88%
- FSG Cache ≈ 421 GB
- Free Memory ≈ 58 GB

- Examples of objects that are memory resident:

- Hash Maps
- Configuration Maps
- Master Indexes
- RTS – Request-to-Steps Cache
- D/D – Data Dictionary Cache



Ex. 512 GB Memory

This example considers an IntelliFlex™ 2.1 node with 512 GB of memory executing the Vantage Advanced SQL Engine. This example considers 2 PEs, 36 AMPs (with 120 AWTs per AMP), PDE, GTW, RSG, 2 TVS, and a NODE vproc in this node. This means that memory will have to be allocated for the vprocs.

In this example

- The operating system, device drivers, and Teradata vprocs will initially use approximately 33 GB of memory of the 512 GB. Practical experience (for most environments) indicates that the operating system (e.g., Linux) may need more than this initial allocation.
- The amount of memory that an AMP uses can be determined/verified by using the "ctl" utility "hardware" command. For example, each AMP (of 36) has an average of 11.69 GB of memory. Therefore, 36 AMPs x 11.69 = 421 GB of FSG cache.
- Parsing Engines and especially AMPs will typically use more than their initial allocation of memory (120 MB). Redistribution buffers for an AMP may use an additional (or more) 280 MB of memory for a total of 400 MB of memory per AMP. Therefore, PDE is not assigned all of the remaining memory for FSG cache, but a percentage of the remaining memory. The default of 88% for FSG Cache Percent works for most systems with 512 GB of memory per node. In this example, 88% of GB (512 - 33) = 421 GB of FSG cache.

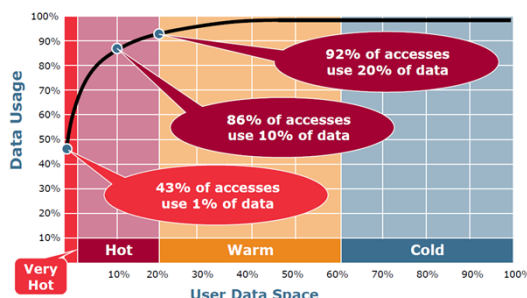
Q1 What are Redistribution Buffers for AMP?

A1 These determine the buffer size for AMP-level hashed row redistributions, as used by load utilities (MultiLoad and FastLoad), and archive/recovery operations. This field also determines the size of the buffers used to redistribute USI rows when creating an index



## Why Teradata Intelligent Memory (TIM)

- Complexity & 3V's of Data
  - With additional storage of near real-time and tactical use of data warehouses with big data sources, faster query performance is still expected and can be quite challenging.
  - In-memory databases try to solve the performance issue by storing all data in memory which makes it expensive
- Multi-Temperature Data
  - All data does not have the same value and is not used in the same way
  - Blindly storing all data in memory in a brute-force attempt to meet data warehouse performance requirements



As companies simultaneously expand near real-time tactical use of data warehouses, and store more data from traditional and new big data sources, faster query performance is imperative. But with the volume, complexity, variety, and variability of data—as well as uses—growing exponentially, delivering rapid query performance can be challenging.

In-memory databases try to solve the performance issue by storing all data in memory. However, this drives the cost of the system up and limits the amount of data that can be stored and made available to the business.

## Teradata Intelligent Memory (TIM): Benefits

### Teradata® Intelligent Memory

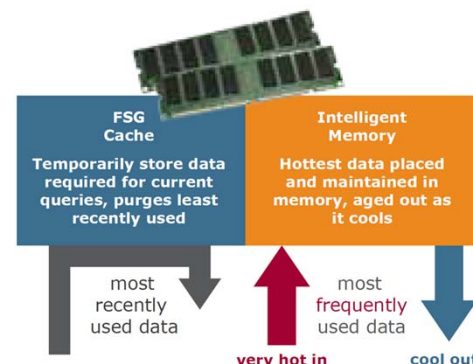
- Maximizes the value of system memory by ensuring that the most frequently used data is kept in memory
- Provides a new extended memory space
- Improves query performance
- Is a smarter approach than in-memory databases
- Teradata Intelligent Memory does not require the manual help of Database Administrators (DBA's)

Teradata Intelligent Memory is the only memory technology that automatically and transparently puts the hottest, most frequently used data in memory.

# Teradata Intelligent Memory (TIM): Implementation

## Teradata® Intelligent Memory

- Teradata Database continuously tracks the temperature (or the relative access frequency) of all data in the database. \
- The most frequently used data is identified internally within the database on a “very hot” data list.
- Whenever data on the very hot data list is read from disk during query processing, a copy is kept in Teradata Intelligent Memory
- When data in Intelligent Memory is needed by another query, Teradata Database automatically looks to Intelligent Memory, eliminating the need for Disk I/O
- Users get the answers they need with the performance they want because Intelligent Memory reduces the I/O needed in the system and increases performance and throughput



Teradata Intelligent Memory is the only memory technology that automatically and transparently puts the hottest, most frequently used data in memory.

Teradata Database FSG cache (i.e., FileSegment cache, the primary internal cache within Teradata Database) is an extremely efficient short-term database cache.

Teradata File System keeps several classes of data likely to be needed in the immediate future available in FSG cache (e.g., temporary tables, spool intermediate results, indexes).

This mechanism achieves high "cache hit rates" however it is available for very short periods of time.

Teradata Intelligent Memory partners with FSG cache with a focus on long-term data usage. Its extended memory area works alongside existing FSG cache and provides a more stable, temperature based collection of data which will satisfy many queries over an extended period of time. Teradata Intelligent Memory works with FSG cache to ensure that only one copy of data is kept in memory at any given time.

When data is accessed from either FSG cache or Teradata Intelligent Memory, those accesses are automatically tracked along with disk I/O accesses to maintain accurate temperature measurements for all data in the system.

## Block and Row Terminology

### Data Block Size (DATABLOCKSIZE)

- Effectively the largest block size and impacts when a block split will occur.
  - Default `DBSControl PermDBSize` may use 512 sectors
  - Table level attribute DATABLOCKSIZE has a max of 2047 sectors (~ 1 MB)

### Small and Large Data Blocks

- Small Blocks – established if table DATABLOCKSIZE  $\leq$  255 sectors
- Large Blocks – established if DATABLOCKSIZE  $>$  255 sectors.

### Large Row

- The largest fixed-length row that allows multiple rows/block; less than  $\frac{1}{2}$  of the block size

### Oversized Row

- A row that requires its own Data Block (one I/O per row):
- A fixed length row that is larger than a Large Row; greater than  $\frac{1}{2}$  of the block size

Tables supporting analytic or decision support queries generally have their block size set very large to accommodate more rows per block and reduce the number of block I/Os needed to do full table scans. Tables involved in online applications and heavy data maintenance generally have smaller block sizes.

Extremely large rows, called Oversized Rows, can be costly. Each Oversized row requires its own block and costs one I/O every time it is touched. Oversized rows are common in non-relational data models and often appear in poor relational data models.

# Cylinder Full

Cylinder Full means there is no block big enough on the Free Block List. The File System does either of the following:

1. Cylinder **Migrate** to an **adjacent** cylinder

- Checks logically adjacent cylinders for fullness. If it finds space, it moves a maximum of 10 data blocks from the full cylinder to an adjacent one.
- Cylinder Defragmentation
  - If the total cylinder free space  $\geq 25\%$ (default) of the cylinder size, then the cylinder is defragmented.
  - Defragmentation collects all free sectors at the end of a new cylinder by moving all the data blocks to the top of the new cylinder.
- Mini-Cylpack
  - if the number of free cylinders falls below a threshold (default is 10), then a "Mini-Cylpack" is performed to pack data together to free up a cylinder and place it on the free cylinder list.

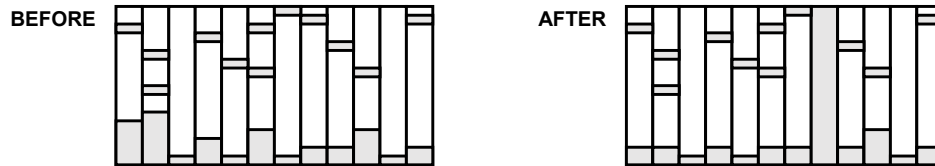
2. Cylinder **Migrate** to a **new** Cylinder

- Looks for a free cylinder, allocates one, and moves a maximum of 10 data blocks from the congested cylinder to a new one.

A **Cylinder Full** condition occurs when there is no block on the Free Block List that has enough sectors to accommodate additional data during an INSERT or UPDATE. If this condition occurs, the File System goes through the steps outlined on this slide which results in a **Cylinder Migrate** to an existing adjacent cylinder or to a new cylinder. As part of this process, the file system software may also choose to perform a **Cylinder Defragmentation** or a **Mini Cylinder Pack (Mini-Cylpack)** operation.

- A **Mini-Cylpack** is a background process that occurs automatically when the number of free (or available) cylinders falls below a threshold. The mini-Cylpack process is the mechanism that Teradata uses to rearrange data blocks to free cylinders. This process involves moving data blocks from a data cylinder to the logically preceding data cylinder until a whole cylinder becomes empty.
- Mini-Cylpack is an indication that the system does not have enough free space to handle its current workload.

## Mini-Cylpack



When the system falls below a DBS Control specified threshold (e.g., 10 free cylinders), the Mini-Cylpack process

- Moves data blocks from the data cylinder(s) to logically preceding data cylinder(s) until enough cylinders are emptied to go above the threshold.
  - Spool cylinders are never cylpacked.
  - Mini-Cylpacks indicate that the system does not have space to handle its current workload.
  - Excessive Cylpacks indicate too little disk space and/or spool utilization during data maintenance.

The **Mini-Cylpack** is the mechanism that Teradata uses to rearrange data blocks to free cylinders. The process involves moving data blocks from a data cylinder to the logically preceding data cylinder until a whole cylinder becomes empty.

- A Mini-Cylpack is an indication that the system does not have enough free space to handle its current workload.
- Spool cylinders are never “Cylpacked”.

Teradata has a **Free Space** (a percentage) parameter that can be set to control how much free space is left in a cylinder during loading and the use of the Ferret PackDisk utility. This parameter is not used with **mini-cylpacks**.

- This parameter should be set low (probably 0%) for tables which are used solely for analytic or decision support processing as there is little data maintenance involved.
- In cases where there is moderate data maintenance on tables (batch or some tactical or OLTP), the **Free Space** parameter should be set at approximately 10 – 15%.
- If heavy data maintenance is to be done on tables (tactical or OLTP), the **Free Space** parameter may have to be set higher (20%).



## Module 1: Logging

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

- Identify various types of logging
- Explain features, setup and the various dictionary tables involved with each logging option





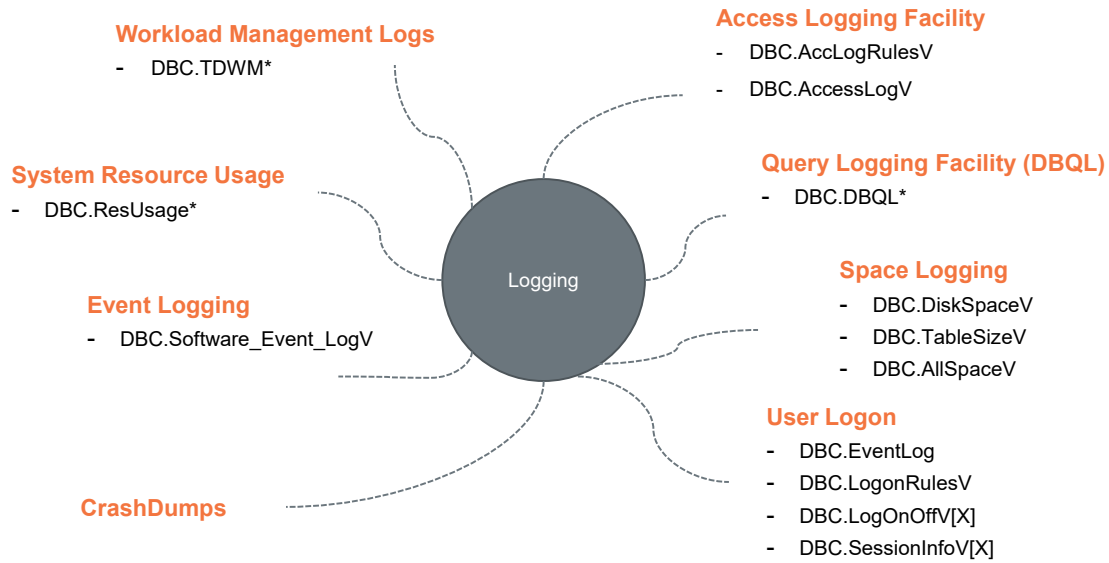
## Why Logging?

Logging is important

- To perform access and security audit analysis
- Used for historical query activity and workload analysis
- Measure component performance
- Analyze performance degradation and improvement
- Space allocation for databases and tables
- Capture software errors and system events
- Analyze the query execution plan
- Capacity Planning



# Logging Components



Note: Query Logging and Workload Management Logs is covered in detail as part of separate modules

DBC.tablesizeV

## Access Logging – Features

An administrator can use the Access Logging facility

- Access and security audit analysis
- Monitor data access requests (via access rights checks) and log entries for requests that are granted and/or denied
- Optionally capture the SQL text along with the access right check

Any User who has been granted EXECUTE permission on below macro can start and end monitoring

Example:

**GRANT EXECUTE ON DBC.AccLogRule TO SecAdmin;**

This allows "SecAdmin" to execute the



### **BEGIN LOGGING**

- Starts the monitoring of data access requests

### **END LOGGING**

- Ends the monitoring of data access requests

# Access Logging – Setup

**Step 1****Install Access Logging on the system**

- a. Run the DIP script DIPACC to install the DBC.AccLogRule macro
- b. Restart the database to activate the code

**Step 2****Grant Privileges to a security administrator**

- a. CREATE USER SecAdmin AS PASSWORD=\*\*\*\*\*, PERM=0, SPOOL=5E9;
- b. GRANT EXECUTE ON DBC.AccLogRule TO SecAdmin;

**Step 3****Define access logging rules as security admin user**

- a. BEGIN LOGGING .....
- b. END LOGGING .....

## Access Logging – Data Dictionary

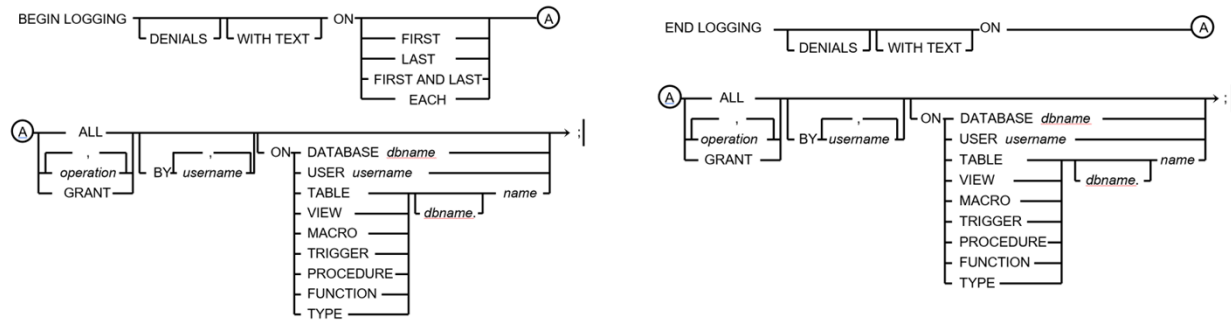
	DBC Table	DBC View
Execution of <b>BEGIN LOGGING, END LOGGING</b> statements causes rows (representing the rules) to be added or updated in	<ul style="list-style-type: none"><li>DBC.AccLogRuleTbl</li></ul>	<ul style="list-style-type: none"><li>DBC.AccLogRules[V]</li></ul>
Based on the rules, access of specified objects or SQL statements cause entries to be placed in	<ul style="list-style-type: none"><li>DBC.AccLogTbl</li></ul>	<ul style="list-style-type: none"><li>DBC.AccessLog[V]</li></ul>

To delete entries in the DBC.AccLogTbl, use the DBC.DeleteAccessLog[V][X] views.

- **DELETE FROM DBC.DeleteAccessLogV;**
  - Deletes entries older than 30 days
- **DELETE FROM DBC.DeleteAccessLogV WHERE LOGDATE < (CURRENT\_DATE – 90);**
  - Deletes entries older than 90 days

Note: Deleting entries is more applicable when PDCR is not enabled.

# Access Logging – Logging Statement



**Operation** Any function for which an access right can be granted (e.g., GRANT)

**BY** *username* – implies all users, if not specified

**ON** *object-name* – implies all entities, if not specified. Valid object-names are:

<b>DATABASE</b>	database_name	<b>USER</b>	user_name	<b>TABLE</b>	table_name	<b>VIEW</b>
	view_name					
<b>MACRO</b>	macro_name	<b>PROCEDURE</b>	procedure_name	<b>TRIGGER</b>	trigger_name	
<b>FUNCTION</b>	function_name					

## Access Logging – Example (1 of 2)

1. Log All Denied Select requests on Table PD.Employee with text details
2. Log First Successful Insert requests on Table PD.Employee with text details
3. Log First and Last Successful Delete requests on Table PD.Employee with text details
4. Log First Update requests on Table PD.Employee with text details
5. Log Last Denied Update requests on Table PD.Employee with text details

BEGIN LOGGING	DENIALS	WITH TEXT	ON EACH	SELECT ON TABLE PD.Employee;	1
BEGIN LOGGING		WITH TEXT	ON FIRST	INSERT ON TABLE PD.Employee;	2
BEGIN LOGGING		WITH TEXT	ON FIRST AND LAST	DELETE ON TABLE PD.Employee;	3
BEGIN LOGGING		WITH TEXT	ON FIRST	UPDATE ON TABLE PD.Employee;	} 4
BEGIN LOGGING	DENIALS	WITH TEXT	ON LAST	UPDATE ON TABLE PD.Employee;	

Note: This is an academic example, not intended as a best practice example.

## Access Logging – Example (2 of 2)

```

SELECT      UserName          (CHAR (6))          AS "User//Name"
            ,DatabaseName      (CHAR (6))          AS "Dbase//Name"
            ,TVMName           (CHAR (10))         AS "TVM//Name"
            ,AcrSelect
            ,AcrInsert
            ,AcrDelete
            ,AcrUpdate
FROM      DBC.AccLogRulesV
WHERE     DatabaseName = 'PD';

```

			1	2	3	4
User Name	Dbase Name	TVM Name	SEL	INS	DEL	UPD
All	PD	Employee	E-	F +	B +	FL=

Position #1 = How often to log requests (F, L, B, E, blank = First, Last, Both, Each, None)

Position #2 = How often to log denials (F, L, B, E, blank = First, Last, Both, Each, None)

Position #3 = How often to save text (+ All entries, - Denials, = All Specified)

NOTE: There is a nominal overhead of having access logging is up to 5%



## System Usage Logging (ResUsage) – Features

ResUsage data has the following features:

- Historical system-level performance information
- Data is logged into various ResUsage tables at a specified logging interval
- Two Teradata subsystems work in conjunction with other subsystems to gather data:
  - Parallel Database Extension (PDE)
  - Resource Sampling Subsystem (RSS) which writes the collected data to ResUsage tables

ResUsage data gathering is a two-phase process that encompasses data collection and data logging. The ResUsage facility consists of a set of tables, views, and macros to access system metrics.

Two Teradata subsystems work in conjunction with other subsystems to gather ResUsage data:

- Parallel Database Extension (PDE)
- Resource Sampling Subsystem (RSS)

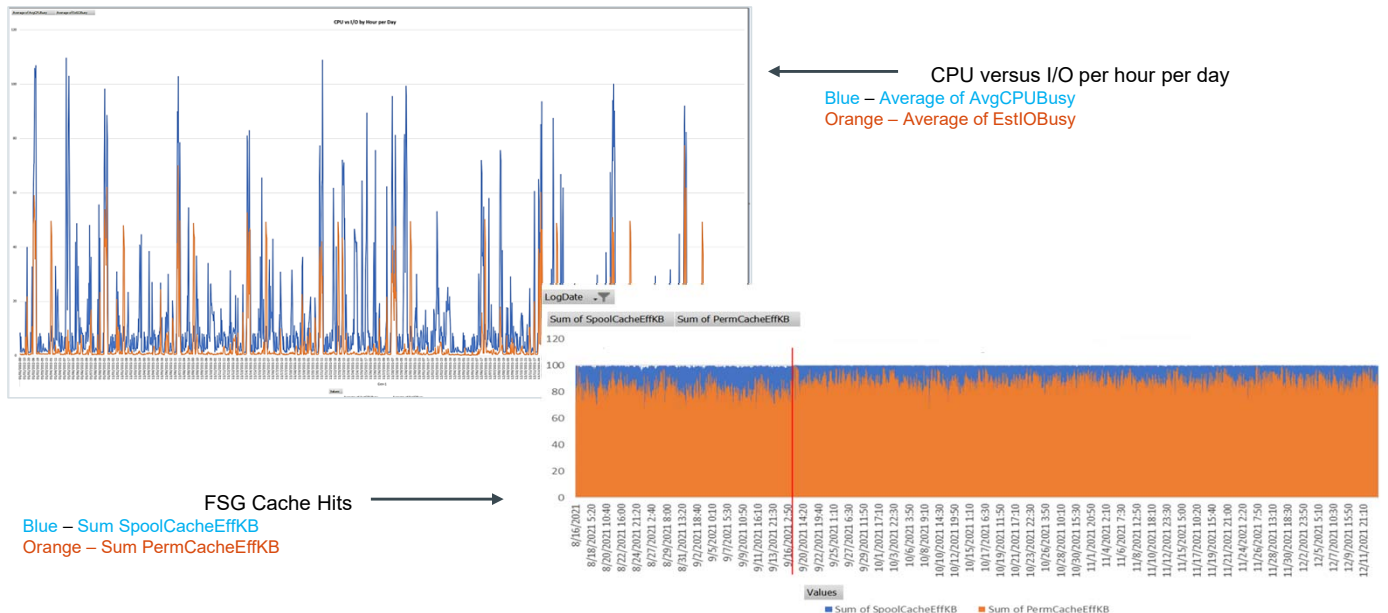
**Data Collection** - during the data collection phase, both PDE and RSS gather information from the operating system and from Teradata Database. This data is temporarily stored in shared data collection buffers. Data collection continues for a user-specified period of time called the collect interval.

**Data Logging** - in the logging phase, RSS writes all gathered data to ResUsage tables and reinitializes the shared data collection buffers for the next log interval.

**Collection Costs** - recording information in the DBC.ResUsage table requires disk space and processing time. Despite the additional resources used in performance monitoring, there are benefits to understanding how your system resources are being used.

**Cost** - the collection of ResUsage data incurs associated system overhead costs in three areas: I/O capacity, User DBC Perm Space and CPU utilization. Space is usually the largest cost.

# System Usage Logging (ResUsage) – Features



CPU versus I/O per Hour per Day

## ResUsage Logging – Setup

### > Remote Console

TDSystem > Operator Console Utility

Utility cnscons started in processor 0, partition 20.

Input Supervisor Command:

get resource

Rate Information:

PMPC Resource Collection Rate = 6  
Node Logging Rate = 60

Input Supervisor Command:

get logtable all

spma table's logging is enabled

ipma table's logging is enabled

svpr table's logging is enabled

ivpr table's logging is enabled

scpu table's logging is enabled

sldv table's logging is enabled

spdsk table's logging is enabled

### > Remote Console

TDSystem > Operator Console Utility

set resource logging 1200

Input Supervisor Command:

get resource

Rate Information:

PMPC Resource Collection Rate = 6  
Node Logging Rate = 1200

Input Supervisor Command:

set resource collection 600

Input Supervisor Command:

get resource

Rate Information:

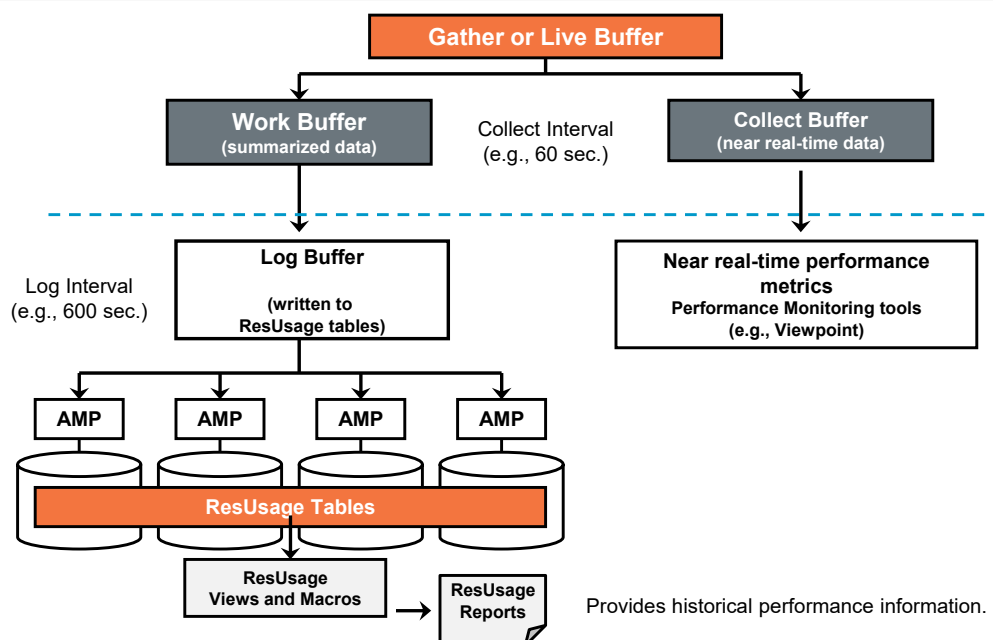
PMPC Resource Collection Rate = 600  
Node Logging Rate = 1200

Input Supervisor Command:

Within RemoteConsole of the Viewpoint there is Operator Console which allows you to enter supervisor commands

- run the command "get resource" to see the defined RSS rates
- run the command "get logtable all" to see the status of all RSS tables
- run the command "set logtable all on" to activate all the RSS tables,

## ResUsage Logging



ResUsage information is gathered in three ways, depending on the nature of the data being collected.

- Counted – the number of times an event happened. The “gather or live” buffer is updated.
- Time monitored – determines how much time was spent in a particular state. The “gather or live” buffer is updated at each state change.
- Tracked data – uses a snap shot of a queue length at the collect period. This information goes directly into the collect buffer.

### Collection and Log Buffers

Statistical information is stored in the gather buffer and holds it there until the set collect interval. At that time, the utility moves the data to the work and collect buffers. The collected data is referred to as PM/PC (Performance Monitoring/Production Control) data.

### Resource Usage Logging

When you initiate data logging, the system collects a variety of statistics for a period you specify. Teradata stores performance data in the DBC.ResUsage tables. Teradata uses nine tables to gather resource utilization data for a specified time period, and stores this information by node or vproc. The system does not automatically collect resource utilization data. Consequently, you must activate resource collection and logging to gather performance data

### Real-time Performance Monitoring

With Viewpoint, you can view and capture data by looking directly into the collect buffer to see what is happening real-time on the system.

## ResUsage Logging – Data Dictionary Tables

	DBC Table
<b>Node Data</b> <ul style="list-style-type: none"> <li>- System-wide node information</li> <li>- System-wide internal node information (generally not enabled)</li> </ul>	<ul style="list-style-type: none"> <li>• DBC.ResUsageSpma</li> <li>• DBC.ResUsageIpma</li> </ul>
CPU Data - Information specific to the CPUs in a node	<ul style="list-style-type: none"> <li>• DBC.ResUsageScpu</li> </ul>
<b>Vproc Data</b> <ul style="list-style-type: none"> <li>- Data specific to each virtual processor</li> <li>- System-wide internal vproc information</li> </ul>	<ul style="list-style-type: none"> <li>• DBC.ResUsageSvpr</li> <li>• DBC.ResUsageLvpr</li> </ul>
Host and Lan Data - Host channel and LAN traffic	<ul style="list-style-type: none"> <li>• DBC.ResUsageShst</li> </ul>
Logical Device Data - Information specific to disk I/O	<ul style="list-style-type: none"> <li>• DBC.ResUsageSldv</li> </ul>
AWT - Collects and reports statistics about the AWTs	<ul style="list-style-type: none"> <li>• DBC.ResUsageSawt</li> </ul>
<b>DISK Data</b> <ul style="list-style-type: none"> <li>- Provides AMP-level Pdisk statistics</li> <li>- Provides AMP-level Vdisk statistics</li> </ul>	<ul style="list-style-type: none"> <li>• DBC.ResUsageSpdsk</li> <li>• DBC.ResUsageSvdsk</li> </ul>
<b>Workload Data</b> <ul style="list-style-type: none"> <li>- Priority Scheduler details &amp; information</li> </ul>	<ul style="list-style-type: none"> <li>• DBC.ResUsageSps</li> </ul>

## ResUsage Logging – Data Dictionary Views

	DBC Views	DBC Tables
View of general system information	• DBC.ResGeneralInfoView	• ResUsageSpma
View of CPU usage by AMP	• DBC.ResCPUUsageByAMPView	• ResUsageSvpr
View of CPU usage by PE	• DBC.ResCPUUsageByPEView	• ResUsageSvpr
View of Host Channel and LAN activity	• DBC.ResShstGroupView	• ResUsageShst
View of disk activity with Node Groups	• DBC.ResSldvGroupView	• ResUsageSldv
View AMP virtual disk activity	• DBC.ResVdskGroupView	• ResUsageSvds

Each row in the ResUsage tables represents activity during one logging period; the same is true of each row in the views. The views derive values from data in ResUsage tables.

### **ResGeneralInfoView**

The ResGeneralInfoView provides an overview of system operation. Contains data from ResUsageSpma covering CPUs, disks, and BYNET information.

### **ResCPUUsageByAMPView**

Contains data from ResUsageSvpr detailing the ways the CPUs are used by the AMPs.

### **ResCPUUsageByPEView**

Contains data from ResUsageSvpr detailing the ways the CPUs are used by the PEs.

### **ResShstGroupView**

The ResShstGroupView is based on the ResUsageShst table.

### **ResSldvGroupView**

The ResSldvGroupView is based on the ResUsageSldv table.

### **ResSvdsGroupView**

The ResSvdsGroupView is based on the ResUsageSvds table.

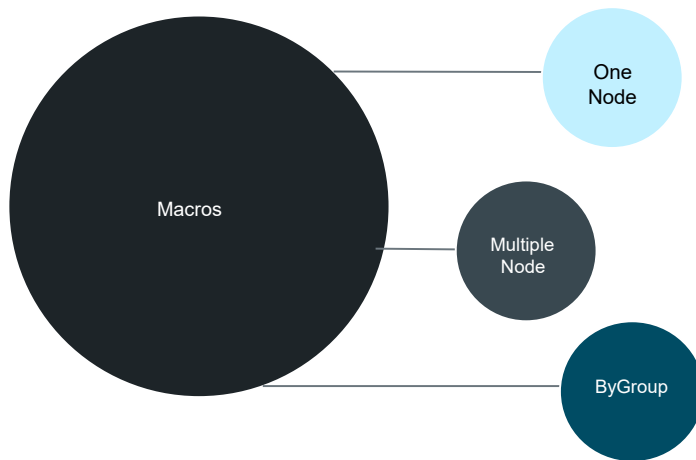
## ResUsage Logging – Aggregation and Join

Join Columns

	IPMA	IVPR	SAWT	SCPU	SHST	SLDV	SPDSK	SPMA	SPS	SCDSK	SVPR
TheDate	x	x	x	x	x	x	x	x	x	x	x
TheTime	x	x	x	x	x	x	x	x	x	x	x
NodeID	x	x	x	x	x	x	x	x	x	x	x
VprId		x	x		x					x	x
CPUID				x							
HstID					x						
LdviD						x					
PDiskDeviceID							x				
WdID									x		

Aggregation Level

## ResUsage Logging – Data Dictionary Macros



- They are primarily used for Single node system
- One-node versions eliminate redundant report columns
  - NodeID columns that focus
  - cross-node load balancing)
- They can be used as <FromNode> , <ToNode>
- Multiple-node macro to report on just one node by supplying equal FromNode and ToNode parameters
- Co-existing nodes are nodes of different model types in the same configurations hence the nodes may become bottlenecks in the throughput of the system as a whole
- These macros were developed to provide the system user with a summary of the performance data based on node groupings (identified by DBA when the system is configured)



# User Logon – Features

An administration via the Logon processing feature to

- Prevent unauthorized persons from gaining access to the RDBMS and its resources
- Permit legitimate users access to only those resources they are authorized to use



## Level 1

### Physical Security

- Physical security pertains to the actual building or computer room in which the Teradata system resides
- The system's owner designs and implements physical security



## Level 2

### Host Logon Processing

- Host logon processing is the first level of access control and allows or disallows connection between the host and database systems
- It involves a host ID and password. This level controls access to the host system



## Level 3

### Database Processing

- Database logon processing is another level of access control and determines access to the Teradata system
- This level performs user authentication and controls access to the Teradata system itself

## User Logon – Setup

An administrator can ...

- Grant or deny users permission to logon to Teradata from specific client connections
- Give users permission to logon to Teradata from specific host connections using a NULL password
- This is specifically for Host logon processing

Example:

**GRANT EXECUTE ON DBC.LogonRule TO SecAdmin;**

This allows "SecAdmin" to execute the



**GRANT LOGON**

- Gives users permission to logon to Teradata from specific client connections

**REVOKE LOGON**

- Denies users permission to logon to Teradata from client system(s)

Logging Slide 1-21

host_ID	Host number from configuration data. The database console is host number "0" (zero). ALL is represented as "1024"
AS DEFAULT	Changes the default for the specified host
username	You can specify up to 25, but not "DBC"
WITH NULL	When used in conjunction with a TDP exit or with single sign-on (LDAP)
PASSWORD	Overrides the system default that a password is required

## User Logon – Data Dictionary

Level 2

	DBC Table	DBC.View
Execution of <a href="#">GRANT LOGON</a> or <a href="#">REVOKE LOGON</a> statements causes rows (representing the rules) to be added or updated in	<ul style="list-style-type: none"> <li>DBC.LogonRuleTbl</li> </ul>	<ul style="list-style-type: none"> <li>DBC.LogonRules[V]</li> </ul>

Level 3

	DBC Table	DBC.View
Provides information about logon attempts (successful or unsuccessful) and logoffs	<ul style="list-style-type: none"> <li>DBC.EventLog</li> </ul>	<ul style="list-style-type: none"> <li>DBC.LogOnOff[V][X]</li> </ul>
Provides information about the current user or all users currently logged on, their session source, the logon date, and connect time	<ul style="list-style-type: none"> <li>DBC.SessionTbl</li> </ul>	<ul style="list-style-type: none"> <li>DBC.SessionInfo[V][X]</li> </ul>

## Query Logging – Features

An administration user can use the Query Logging facility

- Analyze and Log historical query information
- To cache and eventually store query information in multiple Teradata Data Dictionary tables as the queries are executed
- Track SQL for query performance and workload analysis

Any User who has been granted EXECUTE permission on below macro can start and end monitoring

- Initially, only DBC and SystemFE users are allowed to issue BEGIN/END QUERY LOGGING statements

Example:

**GRANT EXECUTE ON DBC.DBQLAccessMacro TO SecDBA;**

This allows "SecDBA" to execute the



**BEGIN QUERY LOGGING** → Starts query logging

**END QUERY LOGGING** → Ends query logging

# Query Logging – Setup

**Step 1****Install Access Logging on the system**

- a. Run the DIP script DIPACC to install the DBC.AccLogRule macro
- b. Restart the database to activate the code

**Step 2****Grant Privileges to a security administrator**

- a. CREATE USER SecDBA AS PASSWORD=\*\*\*\*\*, PERM=0, SPOOL=5E9;
- b. GRANT EXECUTE ON DBC.DBQLAccessMacro TO SysDBA;

**Step 3****Define access logging rules**

- a. BEGIN QUERY LOGGING .....
- b. END QUERY LOGGING .....

## Query Logging – Data Dictionary

	DBC Table	DBC View
Execution of <b>BEGIN QUERY LOGGING, END QUERY LOGGING / REPLACE QUERY LOGGING</b> statements causes rows (representing the rules) to be added or updated in	• DBC.DBQLRuleTbl	• DBC.DBQLRules[V]
Stores default rows (key table)	• DBC.DBQLLogTbl	• DBC.QryLog[V]
One row per step	• DBC.DBQLStepTbl	• DBC.QryLogSteps[V]
One row per object referenced in query	• DBC.DBQLObjTbl	• DBC.QryLogObjects[V]
Stores full SQL text – multiple rows may be needed	• DBC.DBQLSqlTbl	• DBC.QryLogSQL[V]
Queries meeting Summary or Threshold rules	• DBC.DBQLSummaryTbl	• DBC.QryLogSummary[V]
Stores EXPLAIN steps of query	• DBC.DBQLExplainTbl	• DBC.QryLogExplain[V]
Stores Optimizer query plan for the logged query as an XML document	• DBC.DBQLXMLTbl	• DBC.QryLogXMLV
.....	.....	.....

Note: Query Logging is covered in detail as part of separate module

## Event Logging – Features

This Event logging feature is

- Used by Teradata Customer service (SystemFE user)
- Automatically inserts rows in response to all events (hardware and software)

	DBC View	SystemFE.Macros
Returns all events (hardware and software) for the set period of time.	• DBC.Software_Event_LogV	• ListSoftware_Event_Log
Returns all software restart events and logon-enabled events for the set period of time.	• DBC.Software_Event_LogV	• ListRestart_Logon_Events
Returns all software restart events for a set period of time.	• DBC.Software_Event_LogV	• AllRestarts
Returns all disk events for a set period of time.		• DiskEvent
.....	• .....	• .....

The DBC.Software\_Event\_LogV view is based on the table DBC.Sw\_Event\_Log



## Crashdumps – Features

When there is an unscheduled TPA reset, a crashdump is generated (PDE dump).

- If you encounter any problems that result in a Teradata crashdump being generated, contact Teradata Services

Type	Description
Operating System (OS) dump	<ul style="list-style-type: none"> <li>- An OS (varies with platform) dump is the unprocessed contents of memory at the time of an operating system crash</li> <li>- An OS dump is the result of a single node OS crash and contains the entire contents of the memory of a node. It may also be referred to as a system dump, kernel dump, or node dump. On Linux, an OS dump is called a kernel panic dump</li> <li>- An OS dump is independent of Teradata or any particular application</li> </ul>
Teradata Crashdump	<ul style="list-style-type: none"> <li>- A Teradata crashdump also known as a PDE dump is a result of an unrecoverable Teradata error</li> <li>- When Teradata crashes, PDE on each node writes selective data to its own PDE dump directory, called a raw PDE dump. How much and which of this data is actually written also depends on the type of error</li> </ul>
Snapshot dump	<ul style="list-style-type: none"> <li>- A snapshot dump occurs when a process fails on a node and the failure does not cause a restart. PDE writes selective data to the dump directory, and then the database may abort or retry the request</li> </ul>

When there is an unscheduled TPA reset, a crashdump is generated (PDE dump).

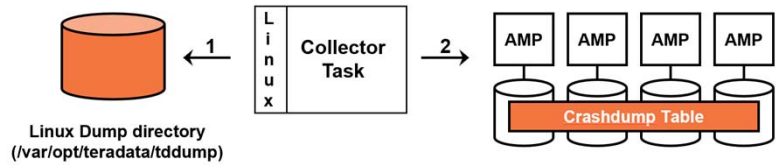
### PDE DUMP

A PDE dump is a selective dump of system memory; including only information that might be needed to analyze a problem within PDE or TPA (the only one currently is the Teradata Database). It can also contain pages read in from swap space that are not in memory at the time of the dump. Exact contents vary depending on the cause of the dump.

PDE dumps, being selective, vary in size depending on the system configuration and the cause of the crash. PDE dumps are much smaller than memory. PDE dumps are taken in parallel on all nodes.

# Crashdumps – Features

- A full crashdump is a system-wide entity in a table format
- All nodes work in parallel to write their piece of the crash system at approximately the same time



- PDE first writes the data in a raw PDE dump and this information is saved (or can be saved) into
  - o A table in the DBC.Crashdumps database
  - o Flat files (called stream files) which are compressed binary Fastload format, which you can either move to another system or upload directly to the Teradata Support Center
- Typically, a Teradata crashdump is smaller than the sum of the memory for all the nodes in the database configuration
- The size of raw PDE dumps still in “dump” format (before the crashdumps are saved in table format) depends on various factors

Type of error	#of Vprocs	Memory	#of Process running (when Vantage crashed)
---------------	------------	--------	---

- The DBC.Crashdumps database, should be large enough to hold 4 to 5 crashdumps (size of raw PDE dumps can vary from 10MB –to- >=4GB)

## Workload Management Logging – Features

The Viewpoint **Workload Monitor** portlet tracks workload management activity

- The following Data Dictionary tables and views in the database DBC contain TASM information

	DBC Table	DBC.View
Information on the following events <ul style="list-style-type: none"> <li>- Ruleset activations, including times when activations and optimizations are made</li> <li>- State changes, including times and details</li> <li>- When a request is moved between workloads due to an exception or manual (DBA) intervention</li> </ul>	DBC.TDWMEventLog	DBC.QryLogEventsV
Information on <ul style="list-style-type: none"> <li>- Data about requests in a workload, summarized over a collection period</li> <li>- Users can define the logging interval in Viewpoint Workload Designer</li> </ul>	DBC.TDWMSummaryLog	DBC.QryLogTDWMSumV
Information on <ul style="list-style-type: none"> <li>- Workload and system events and the values that lead up to a state change</li> </ul>	DBC.TDWMEventHistory	DBC.QryLogEventHisV
Information on <ul style="list-style-type: none"> <li>- A row for each exception that TASM detects for a request, including filter, throttle, and workload exceptions</li> <li>- There could be several rows for one request</li> </ul>	DBC.TDWMExceptionLog	DBC.QryLogExceptionsV

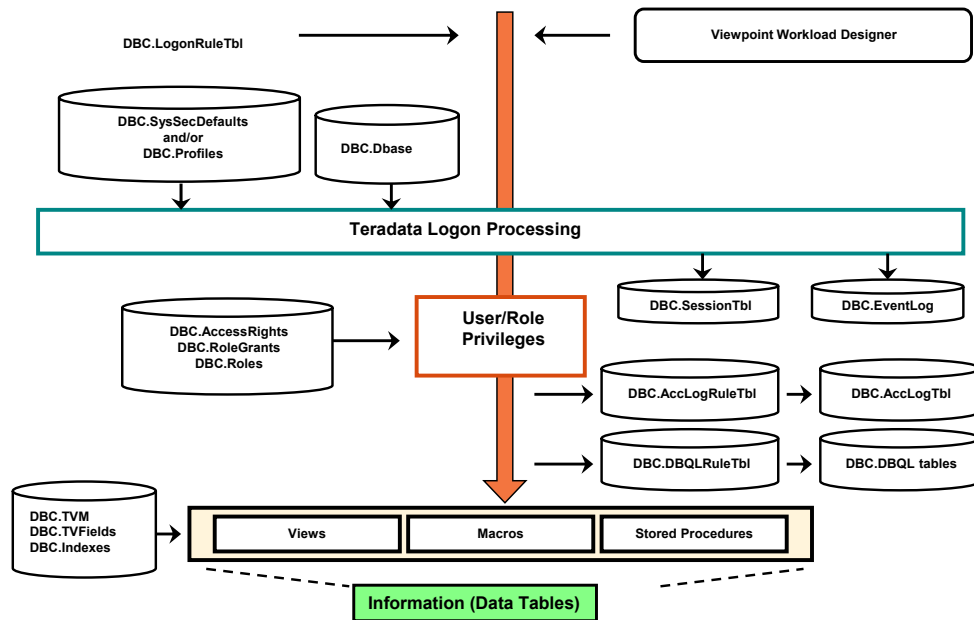
Note: Workload Management is covered in detail as part of separate module

## Space Logging – Features

### Identifying Space Issues by Querying System Views

	DBC View
To capture AMP information about disk space usage at the database or user level <ul style="list-style-type: none"><li>- Can also report Spool Space usage</li><li>- MaxSpool and MaxTemp, one gets the space limits for individual user space</li></ul>	<ul style="list-style-type: none"><li>• DBC.DiskSpaceV[X]</li></ul>
This provides AMP information about disk space usage at the table level	<ul style="list-style-type: none"><li>• DBC.TableSpaceV</li></ul>
This provides space usage information at the object level and the database/user level <ul style="list-style-type: none"><li>- Table, join index, permanent journal, or stored procedures</li></ul>	<ul style="list-style-type: none"><li>• DBC.AllSpaceV</li></ul>

# Key Highlights





## Module 2: Database Query Logging (DBQL) and PDCR

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

- Explain the purpose for using the Database Query Log (DBQL)
- Identify the tables and views that make up the DBQL facility
- Describe the SQL syntax for query logging and its collection options.
- Find information about current query logging rules
- Explain the purpose for using Performance Data Collection and Reporting (PDCR)
- Describe the PDCR Database Structure
- Explain the purpose for the Performance Data Collection portlet in Viewpoint



## Topics

- Database Query Logging (DBQL)
  - What are the benefits of DBQL?
  - What is DBQL?
  - What Objects are used in DBQL?
  - How is logging Initiated?
  - SQL Syntax for query logging and its options
  - Viewing the current logging rules
- Performance Data Collection and Reporting (PDCR)
  - What is PDCR?
  - What's included in the PDCR Database Structure
  - Viewpoint's Performance and Data Collection portlet





## Current Topic – Database Query Logging (DBQL)

- Database Query Logging (DBQL)
  - What are the benefits of DBQL?
  - What is DBQL?
  - What Objects are used in DBQL?
  - How is logging Initiated?
  - SQL Syntax for query logging and its options
  - Viewing the current logging rules
- Performance Data Collection and Reporting (PDCR)
  - What is PDCR?
  - What's included in the PDCR Database Structure
  - Viewpoint's Performance and Data Collection portlet



## How Well Are Your Queries Running?

Query Analysis needs the following understanding

- **Where** can you find query counts and response times?
- **How** do you know if queries are using Non-Unique Secondary Indexes (NUSIs)?
- **How** do you know which Aggregate Join Indexes (AJIs) are used most?
- **How** do you determine which step in the query consumed the most CPU and what types of operation it was performing?
- **How** do you find which applications are accessing your largest transaction tables?



Query Analysis means you have to address some questions...

- Where can you find query/request counts and response times?
- How do you know if queries are using Non-Unique Secondary Indexes (NUSIs)?
- How do you know which Aggregate Join Indexes (AJIs) are used most?
- How do you determine which step in query consumed the most CPU and what type of operation it was performing?
- How do you find which applications are accessing your largest transaction tables?

## Why You Need Database Query Logging (DBQL)

With DBQL you can

- **Capture** query/request counts, response times, and various resource consumption metrics
- **Discover** potential application improvements
- **Validate** whether NUSIs are being used
- **Validate** which AJIs are used most versus resource consumption
- **Identify** potential opportunities for new JI/SI



Here are just some of the benefits that DBQL provides. With DBQL you can...

- Capture query/request counts, response times, and various resource consumption metrics for Query Analysis
- Validate whether NUSIs are being used, allowing you to delete unused NUSIs
- Validate which AJIs are used most, allowing you to make cost-benefit analysis on them, comparing usage against size and time to build
- Discover potential application improvements

## What is Database Query Logging?

The Database Query Log (DBQL) is a feature that logs historical query information

- Based on the rules you specify, captures information in predefined Data Dictionary tables
- Periodically caches and eventually stores information in multiple Data Dictionary tables
- With DBQL you can track SQL for query and workload analysis



The Database Query Log (DBQL) is a feature that you can employ to log query processing activity for later analysis. Query counts and response times can be charted and SQL text and processing steps can be compared to fine-tune your applications for optimum performance.

DBQL provides a series of predefined tables that can store, based on rules you specify, historical records of queries and their duration, performance, and target activity. DBQL is flexible enough to log information on the variety of SQL requests that run on Teradata, from short transactions to longer-running analysis and mining queries. You begin and end collection for a user or group of users and/or one or a list of accounts. DBQL is streamlined for collection efficiency and provides more detail about the query than the Access Log. On the other hand, because DBQL caches data in memory before writing it periodically to disk, there is some delay in seeing the logged data, and it is possible to lose data that is still in the cache on a restart.

## Query Logging – Features

- Any User who has been granted EXECUTE permission on the **DBC.DBQLAccessMacro** macro can start and end monitoring

- Initially, only DBC and SystemFE users are allowed to issue BEGIN/END QUERY LOGGING statements

Example:

**GRANT EXECUTE ON DBC.DBQLAccessMacro TO SecDBA;**

This allows "SecDBA" to execute the



<b>BEGIN QUERY LOGGING</b>	→ Starts query logging
<b>END QUERY LOGGING</b>	→ Ends query logging
<b>REPLACE QUERY LOGGING</b>	→ Replace query logging rule set

- By default, one row per query is logged that contains user id information and some performance metrics for that query
- DBSControl option (**DBQLFlushRate**)
  - Sets the frequency (default is 10 minutes) for writing DBQL cache entries to underlying dictionary tables

The DBQL logs are a series of system tables created in database DBC during the Teradata Database installation process. The suite of DBQL components includes a security macro **DBC.DBQLAccessMacro** and a view for each table, which are created in database DBC by the DIP utility during installation.

Granting the EXECUTE privilege to a user allows that user to issue the BEGIN QUERY LOGGING to create the logging rules.

You define rules that identify for which users and how much data to log for their queries. For instance, you can log DBQL performance metrics, such as CPU and IO consumption, during a session invoked by a specific user under a specific account. This rule can also be qualified so that only queries that exceed a specified time threshold are logged and those queries that execute in less than the threshold time are simply counted.

DBSControl option (**DBQLFlushRate**) – determines the frequency (default is 10 minutes) for writing DBQL cache entries to DBQL dictionary tables.

## Query Logging – Data Dictionary

	DBC Table	DBC View
Execution of <b>BEGIN QUERY LOGGING, END QUERY LOGGING / REPLACE QUERY LOGGING</b> statements causes rows (representing the rules) to be added or updated in	• DBC.DBQLRuleTbl	• DBC.DBQLRules[V]
Stores default rows (key table)	• DBC.DBQLLogTbl	• DBC.QryLogV
One row per step	• DBC.DBQLStepTbl	• DBC.QryLogStepsV
One row per object referenced in query	• DBC.DBQLObjTbl	• DBC.QryLogObjectsV
Stores full SQL text – multiple rows may be needed	• DBC.DBQLSqlTbl	• DBC.QryLogSQLV
Queries meeting Summary or Threshold rules	• DBC.DBQLSummaryTbl	• DBC.QryLogSummaryV
Stores EXPLAIN steps of query	• DBC.DBQLExplainTbl	• DBC.QryLogExplainV
Stores Optimizer query plan for the logged query as an XML document	• DBC.DBQLXMLTbl	• DBC.QryLogXMLV
TDWM events that could affect system performance		• DBC.QryLogEventsV
Query exceptions as defined by Workload Definitions (WD)		• DBC.QryLogExceptionsV

Like other system tables, the predefined DBQL logs are created as relational tables in database DBC during normal Teradata database installation. However, while most system tables are populated automatically, you can choose whether you want to populate the DBQL tables.

If you choose not to use the feature, the tables remain empty. If you want to use the feature, simply submit a BEGIN/END QUERY LOGGING statement, with or without options, to control the start, magnitude, and end of logging activity.

The DBC.DBQLRuleTbl table stores the rules resulting from each BEGIN QUERY LOGGING statement. One row exists for each set of specifications, which are made up of user and/or account plus any options or limits set for the user.

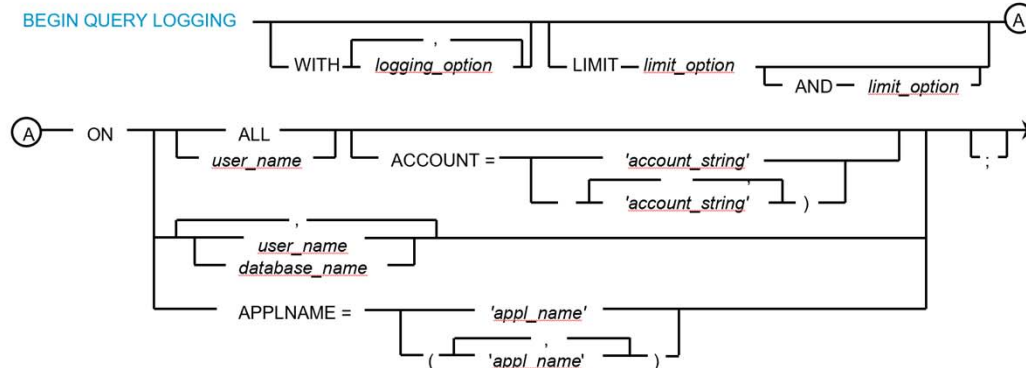
The DBC.DBQLRules[V] views are used to display DBQL rules that are in effect.

The key or foundation table in DBQL is the DBC.DBQLLogTbl table that holds the default rows. When you specify options that result in more information being captured, a default row is still generated in this table plus one or more additional logs (tables) will get rows.

Exceptions to this are when you use the SUMMARY option or a query completes within a THRESHOLD. In these cases, default rows won't be placed into DBC.DBQLLogTbl.

This slide summarizes some of the key tables and views used by DBQL to hold query data.

## BEGIN QUERY LOGGING



- A default row placed in the DBQLogTbl and contains the following information
  - Username, account string (expanded), time stamp information
  - Unique ID for process, session, and client (host) connection
  - First 200 characters of SQL statement

The DBQL facility is controlled by the Teradata SQL statements `BEGIN QUERY LOGGING` and `END QUERY LOGGING`. There are numerous collection options using the `WITH` and `LIMIT` options.

### BEGIN QUERY LOGGING

When submitted by a user with `EXECUTE` privileges on `DBC.DBQLAccessMacro`, enables logging for the named users and/or accounts. For active sessions, logging begins when the next query is received. (Teradata recommends a maximum of 100 user/account pairs per statement.)

When you do not specify a `LIMIT` option, one default row of query-level information is logged in `DBQLogTbl` for each query processed during a session that is initiated by any user for whom a query logging rule exists.

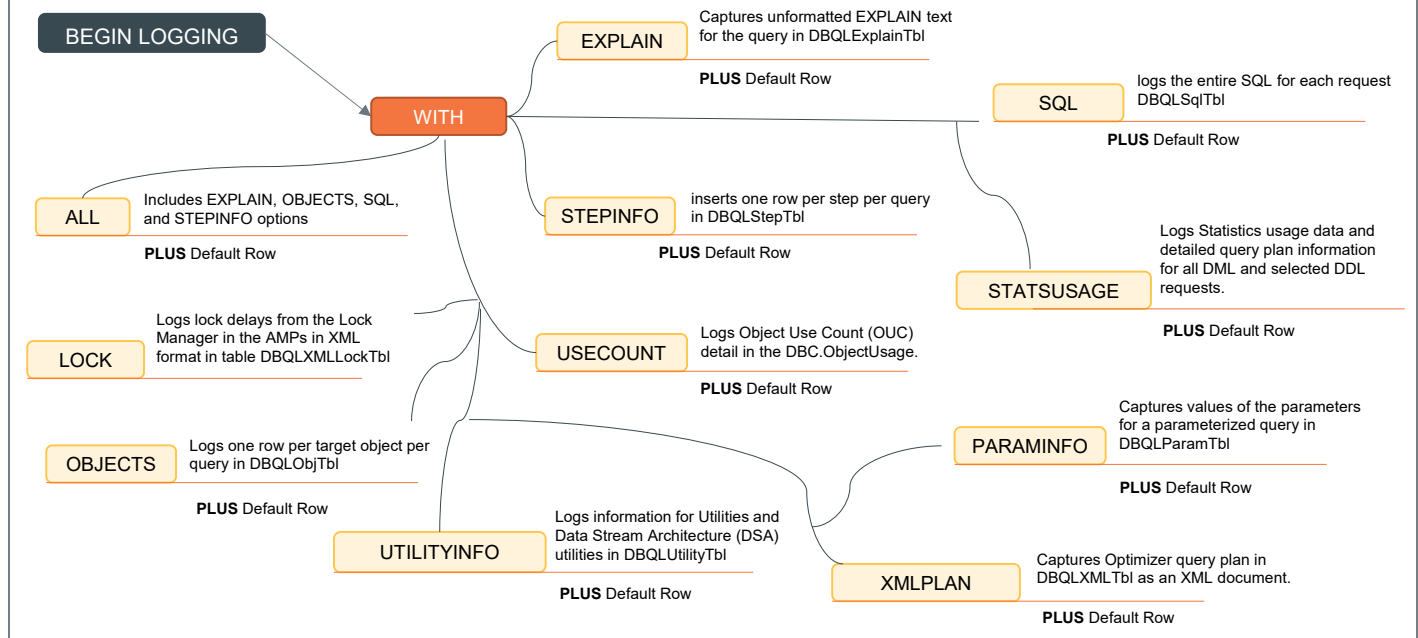
Default rows are stored in `DBQLogTbl`, the foundation of the DBQL feature. If you specify options that result in more detailed information, a default row is still generated in `DBQLogTbl` (except with the `SUMMARY` option or a query that completes within the limit specified with the `THRESHOLD` option), plus one or more additional logs are populated with one or more additional rows.

Examples of valid application names (Teradata 13.0 feature) include `FASTLOAD`, `FASTEXP`, `MULTLOAD`, `ARC`, etc. You can determine the application name used in the `LogonSource` string for a running application by querying `DBC.SessionInfo` as follows:

```
SELECT DISTINCT (LogonSource) FROM DBC.SessionInfo;
```



# BEGIN QUERY LOGGING (WITH)



**ALL** - logs all information for EXPLAIN, OBJECTS, SQL, & STEPINFO and the default row.

**EXPLAIN** - this option generates and logs the unformatted EXPLAIN text for each query. It does not generate EXPLAIN text for queries preceded by the EXPLAIN modifier. This option logs one or more rows into DBC.DBQLExplainTbl.

**OBJECTS** - object data is useful for analyzing queries that make heavy use of join indexes and indexed access. Inserts one row per target object per query in DBC.DBQLObjTbl.

**SQL** - this option logs the entire SQL statement in the DBC.DBQLSqlTbl table. Large statements can cause multiple rows to be written in order to log the full query text.

**STEPINFO** - use this option selectively. Although step data is useful for analyzing queries, this option can generate many rows. Inserts one row per step per query in the DBC.DBQLStepTbl.

Both PARAMINFO and UTILITYINFO option are 15.0 features. A parameterized query uses placeholders for parameters, and the parameter values are supplied in a separate statement at execution time. Because this type of query distinguishes between code and data, it prevents attackers from changing the intent of a query by inserting SQL commands.

Show Parameters logs the values for parameterized queries in a new 15.0 Data Dictionary table, DBC.DBQLParamTbl. For each parameter, DBC.DBQLParamTbl logs the:  
Name, Type, Position, Value

Values can have these data types:

- Standard Teradata data types and their ANSI equivalents
- BLOB/CLOB of up to 64 KB
- UDTs



DBQL - Show Parameters is off by default, but can be enabled with the PARAMINFO option for the BEGIN/REPLACE QUERY LOGGING statements. For example:

**Begin Query Logging with PARAMINFO on user01;**

Logs lock delays from the Lock Manager in the AMPs in XML format in table DBQLXMLLockTbl. This feature expands the Blocked party to include all Waiting Transactions, up to 30 transactions in each AMP.

There is a need to clarify the 2631 error as well as things like Deadlocks and Timeouts using actual Transactional/Session Data from the users. The goal is not necessarily to substitute for the error 2631 or other Contention error/failure. By the time we get into those situations it would have been too late. Rather, this feature supplements those errors. This additional information could be used to forecast and/or explain the length of lock waiting time, deadlocks and timeout due to Database Lock contention.

**XMLPLAN** - logs the query plan generated by the Optimizer for all SQL DML requests as an XML document in system table DBC.DBQLXMLTbl. Because the XML document includes the query and step text, you generally do not need to specify the EXPLAIN and SQL options if you specify XMLPLAN. You should also specify a value of 0 for SQLTEXT to avoid redundant logging when you specify XMLPLAN. This option logs one or more rows into DBC.DBQLXMLTbl.

**STATSUSAGE** - logs statistics usage data and detailed query plan information for all DML and selected DDL requests as an XML document in DBC.DBQLXMLTbl. If you specify both XMLPLAN and STATSUSAGE, Teradata logs the collected data into a single integrated document.

**USECOUNT** - improvements to Object Use Count (OUC) by adding support for statistics access usage and providing detailed usage context via Update, Delete, and Insert (UDI) object usage counts. Reduce risk of lock contention on popular dictionary tables, via new table DBC.ObjectUsage.

**UTILITYINFO, PARAMINFO (15.0)** - capture information for load, export, and Data Stream Architecture (DSA) utilities at the job level or capture parameter values in parameterized query requests.

**LOCK=n (14.10, 15.10)** - logs any Lock contention into DBQLXMLLockTbl

**NONE** - you use this option to turn off logging for any of the following items with NONE:

- account:user pair or account:user list
- application name or application name list
- user name list
- ALL:account name or account name list or ALL (which specifies all accounts)

## BEGIN QUERY LOGGING Examples (WITH)

1. Creates a system rule to log default query, SQL, and OBJECT information for all users/accounts
  - `BEGIN QUERY LOGGING WITH SQL, OBJECTS ON ALL;`
2. Creates two additional rules to disable query logging for tacticaluser1 and tacticaluser2
  - `BEGIN QUERY LOGGING WITH NONE ON tacticaluser1, tacticaluser2;`
3. Creates two rules for a specific user – each rule identifies a specific account id
  - `BEGIN QUERY LOGGING ON tfact03 ACCOUNT = ('$L00EDUC&D&H', '$M00EDUC&D&H');`
4. Creates a rule to enable logging for any TPT Load or Update job
  - `BEGIN QUERY LOGGING ON APPLNAME = ('TPTLOAD', 'TPTUPD');`
5. Creates a rule to log default query information and full SQL on all users and accounts
  - `BEGIN QUERY LOGGING WITH SQL, OBJECTS ON ALL;`

### Hierarchy of Applying Database Query Logging Rules

Database Query Logging works from a hierarchical foundation that allows BEGIN QUERY LOGGING requests to be submitted for individual users even if a rule exists for ALL users.

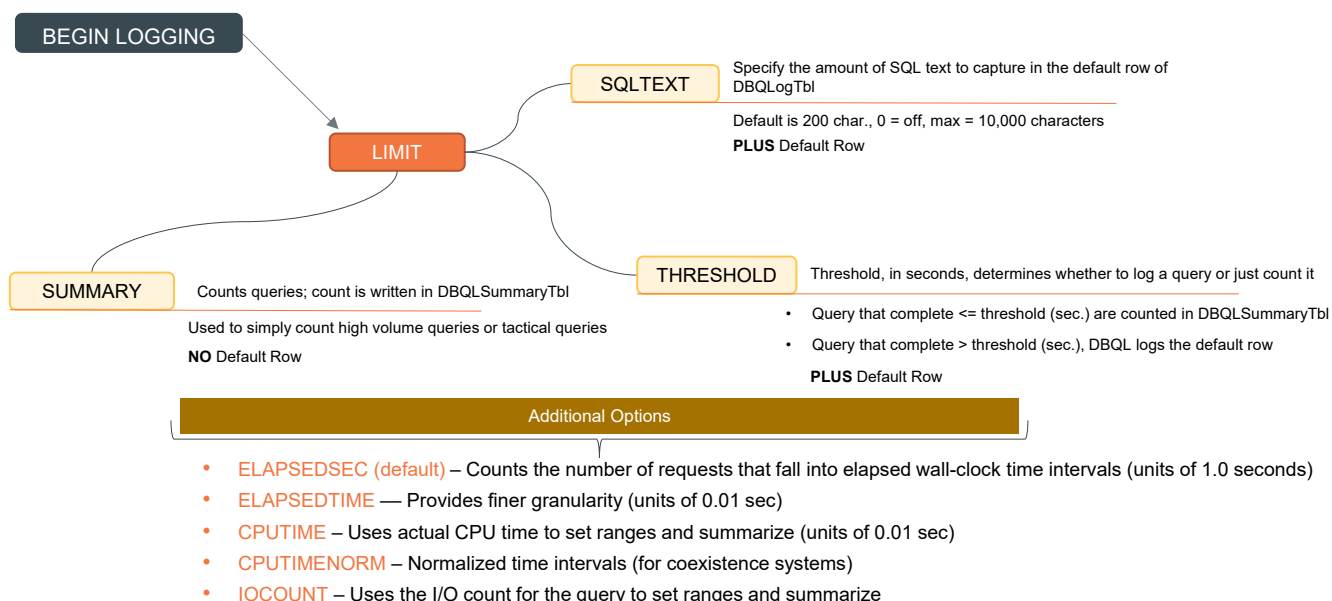
However, if a rule exists for a specific account:user pair, you must submit an appropriate END QUERY LOGGING request to delete the rule before you can issue a new rule for that account:user pair. Teradata Database applies the rules in the following order:

Order	Type of Rule
1	A rule based on an application name.
2	A rule for this specific user and specific account.
3	A rule for this specific user and any account.
4	A rule for all users and this specific account.
5	A rule for all users and any account.

DBQL first searches for a rule based on an application name. If no such rule exists, DBQL then looks for a rule specific to the user and account, and so on down the hierarchy.

For example, you can submit a BEGIN QUERY LOGGING request for default logging on ALL users, and DBQL can also be enabled for *user1* with objects and steps. If *user1* logs on and executes queries, DBQL collects objects and steps. When users other than *user1* log on and execute queries, DBQL only logs default row information for them.

# BEGIN QUERY LOGGING (LIMIT)



**SQLTEXT** - use this option if you want to capture less than or more than the first 200 characters in the default row. The maximum limit is 10,000 characters. If you specify the option keyword but not a value, up to 10,000 characters are logged in DBQLLogTbl.

To store the complete statement regardless of length, specify the SQL option, as many rows as needed to contain the full text will be logged in DBQLSqlTbl. If you do this, define LIMIT SQLTEXT=0 to avoid redundant logging in both the default row and DBQLSqlTbl.

**SUMMARY** - SUMMARY is useful for tracking voluminous short queries, such as for tactical queries, because it does not grow the DBQLLogTbl. It simply counts queries based on specified time differentials in DBQLSummaryTbl. The SUMMARY option is unique in that it:

- Does not generate default rows in DBQLLogTbl
- Summary information is flushed at system-controlled intervals of 10 minutes
- If no data has been collected for a summary category in a 10-minute interval, no rows will be written for it.

**THRESHOLD** - THRESHOLD also is useful for short, high-volume queries, but in addition to incrementing a count for qualifying queries, THRESHOLD logs a default row for any query that exceeds the specified time. This enables you to examine the processing timestamps and the query structure. You can combine THRESHOLD with SQLTEXT if you want to capture more than the first 200 characters of a query that runs longer than THRESHOLD seconds for identification of the longer running queries.

## BEGIN QUERY LOGGING Examples (LIMIT)

1. Capture full SQL text in the SQL table, but does not capture the default first 200 characters of a query in the default row
  - `BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL`
2. If a query runs for less than or equal to 60 seconds of wall clock time, increment the count and greater than 1 minute, log a default row
  - `BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT THRESHOLD = 60 ON ACCOUNT = 'acctname';`
3. Default Summary option only counts running queries based on the elapsed time
  - 3 values (in sec.) are required and 4 count intervals are logged (<=5, <=60, <=600, >600)
  - Summary limit cannot be used with any other limits
  - `BEGIN QUERY LOGGING LIMIT SUMMARY = 5, 60, 600 ON ALL;`
4. If a query uses less than or equal to 1 CPU second, increment the count in the Summary table and uses more than 1 CPU second, log a default row, SQL text, and OBJECT details
  - `BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT THRESHOLD = 100 CPUTIME ON ...;`
5. Summary example counting queries based on actual CPU time
  - 3 values are required. 4 count intervals are logged (<=50, <=100, <=200, >200)
  - 50 is .5 CPU seconds, 100 is 1 CPU seconds, and 200 is 2 CPU seconds
  - `BEGIN QUERY LOGGING LIMIT SUMMARY = 50,100,200 CPUTIME ON ACCOUNT='$H00TACT&D&H'`

SQLTEXT: option to control the number of SQL statement characters to log

- 200 characters of SQL logged in default row
- SQLTEXT values range from 0 to 10,000
- "SQLTEXT" without a value logs 10,000 characters

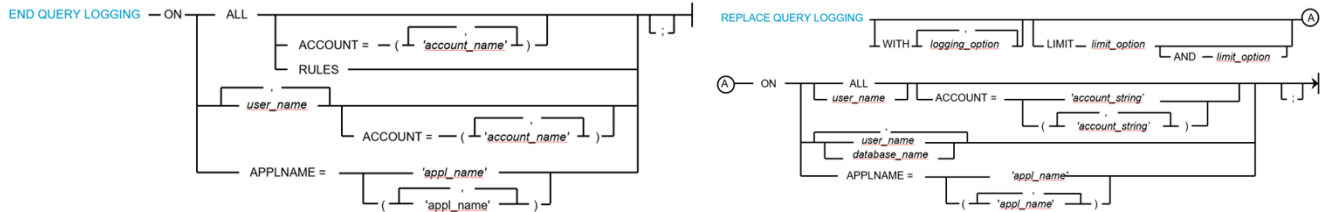
THRESHOLD: option to limit the queries logged by elapsed time

- THRESHOLD values are specified in seconds.
- "THRESHOLD" without a value results in a 5 second value.
- Queries greater than Threshold value generate a default row.
- Maximum THRESHOLD value is 32,767 seconds.

### Notes about Threshold logging

- You can use THRESHOLD logging along with the SQL, STEPINFO, and EXPLAIN options.
- With threshold logging, DBQL cannot log to separate Explain and XML tables, even for those queries taking longer than the specified criteria. SQL, STEPINFO, and OBJECTS can be logged during threshold logging, even for those queries taking longer than the specified clock seconds.
- By filtering logging with THRESHOLD will lower the performance overhead of detailed logging of unnecessary queries.
- For example, log queries that run > 1 CPU second into DBQLLogTbl, DBQLSqlTBL, DBQLStepTbl and DBQLObjTbl and summarize all other queries in the DBQLSummaryTbl:

## END / REPLACE QUERY LOGGING Statement



- **END QUERY LOGGING** statement will cause DBQL cache to be written to the tables except for the Summary cache
  - **END QUERY LOGGING ON ALL RULES;** → Removes all rules.
  - **END QUERY LOGGING ON tfact01;** → You can end logging for a specific user If a list of users was given in the BEGIN statement,
  - **END QUERY LOGGING ON tfact03 ACCOUNT='\$L00EDUC&D&H';** → You can end logging for a specific user If a list of users was given in the BEGIN statement
- **REPLACE QUERY LOGGING** allows you to replace a logging rule set without using BEGIN / END
  - This will help prevent missing logging data between the end logging and begin logging sequence of statements
  - This statement also avoids the unnecessary flushing of DBQL caches
  - Replaces an existing rule. If no rule exists, it will be created

... **ON ALL** – to stop logging query information for all users specified by a rule created by a BEGIN QUERY LOGGING ON ALL statement.

...**user\_name** – the name of a specific user or set of users for whom logging of SQL query information is to be stopped.

...**account\_name** – the name of one or more specific accounts for which logging of SQL query information is to be stopped.

**Account names must be enclosed by LEFT and RIGHT PARENTHESIS characters. When you specify a list of accounts, each account name must be delimited by APOSTROPHE characters and separated by COMMA characters.**

When you enable or disable query logging, the change has an immediate effect on active sessions where the user or account being logged appears within the first 100 names you specify in the user and/or account list of a single BEGIN/END QUERY LOGGING statement. For users listed beyond the first 100, the user must log off from the Teradata Database and restart the session.

When you disable logging (submit an END QUERY LOGGING statement) for an active session (a query for that session is in process) and data is already cached, the following occurs:

- The data is committed immediately
- One or more DBQL rows are written (but may be incomplete); the cache is flushed
- Subsequent queries during that session are not logged

The REPLACE QUERY LOGGING statement allows the customer to modify their query logging for active users without having to end query logging and begin query logging. This will help prevent missing logging data between the end logging and begin logging sequence of statements.

This statement can also be used to avoid the end query logging and begin query logging statement pairs and unnecessary flushing of DBQL caches.

The REPLACE QUERY LOGGING statement will replace an existing rule or will create a new rule if one did not exist.

### **Additional Enhancement**

An additional enhancement is that TypeOfUse column in the DBQLObjTbl table is populated.

The TypeOfUse column contains the following numeric values:

- 1 = Found in the resolver
- 2 = Accessed during query processing
- 4 = Found in a conditional context
- 8 = Found in inner join condition
- 16 = Found in outer join condition
- 32 = Found in a sum node
- 64 = Found in a full outer join condition

# FLUSH QUERY LOGGING

## FLUSH QUERY LOGGING WITH *flush\_option*;

- This SQL statement is valid in Teradata session mode only
- Requires the EXECUTE privilege on DBC.DBQLAccessMacro.

### Example:

1. To flush all DBQL cache, but not flush any of the TDWM cache

- **FLUSH QUERY LOGGING WITH ALLDBQL;**

- ALL
- ALLDBQL
- ALLTDWM
- DEFAULT
- EXPLAIN
- LOCK
- OBJECTS
- SQL
- STATUSAGE
- STEPINFO
- SUMMARY
- TDWMEVENT
- TDWMEXCEPTION
- TDWMHISTORY
- XMLPLAN

To flush one, several, or all database query log (DBQL) caches or Teradata Database Workload Management (TDWM) caches to disk. Note: This SQL statement is only valid in Teradata session mode and requires the EXECUTE privilege on DBC.DBQLAccessMacro.

```
FLUSH QUERY LOGGING WITH flush_option;
```

Where *flush\_option* (14.10 feature) is which database query logging or Teradata database workload management cache to flush: (ALL, ALLDBQL, ALLTDWM, DEFAULT, EXPLAIN, LOCK, OBJECTS, SQL, STATUSAGE, STEPINFO, SUMMARY, TDWMEVENT, TDWMEXCEPTION, TDWMHISTORY or XMLPLAN).

- ALL is mutually exclusive to all other options.
- ALLDBQL is mutually exclusive to all other DBQL options.
- ALLTDWM is mutually exclusive to all other TDWM options.

## Examples

This example flushes the caches for all DBQL and TDWM logs.

```
FLUSH QUERY LOGGING WITH ALL;
```

This example flushes all the caches of DBQL logs but does not flush any of the TDWM caches.

```
FLUSH QUERY LOGGING WITH ALLDBQL;
```



### What's included in this lab activity:

- Analyze queries that you submit using DBQL data
- Discover if a query can be made more efficient with secondary index access
- Identify which step for a specific query is consuming the most CPU and IO resources and discover whether a re-write of the query changes the CPU and IO consumption
- Identify if a query is skewed and determine whether addressing the distribution issue can improve the query
- Find how many records are read from an Amazon S3 bucket for a NOS query



## Current Topic – Performance Data Collection and Reporting (PDCR)

teradata.

- Database Query Logging (DBQL)
  - What are the benefits of DBQL?
  - What is DBQL?
  - What Objects are used in DBQL?
  - How is logging Initiated?
  - SQL Syntax for query logging and its options
  - Viewing the current logging rules
- Performance Data Collection and Reporting (PDCR)
  - What are the benefits of PDCR?
  - What is PDCR?
  - What's included in the PDCR Database Structure
  - Viewpoint's Performance and Data Collection portlet



# How Well has Your System been Running for the Past Year?

teradata.

Performance issues requires addressing following

- **What** Vantage resources are required for Performance analysis?
- **Is there** a single repository providing a historical view capturing all information?
- **How** do I find a list of high-frequency queries along with their cumulative resource consumption?



Performance issues means you have to address some questions...

- What Vantage sources are needed for the purpose of analysis and management of System Performance?
- Is there a single repository that I can use to perform Performance Analysis?
- How do I find a list of high frequency queries along with their cumulative resource consumption?

# Why You Need Performance Data Collection and Reporting (DBQL)

teradata.

With PDCR you can

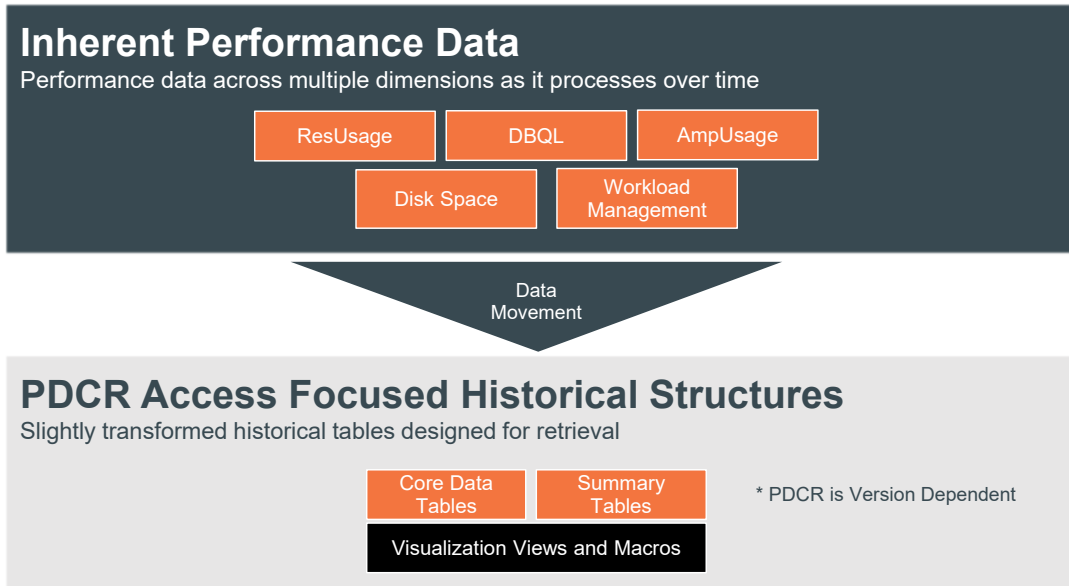
- **Perform** Trend analysis across historical information
- **Analyze** shifting resource consumption across query demographics
- **Identify** capacity trends to avoid running out of system resources
- **Proactive** performance improvement
- **Effective** Space Management



Here are just some of the benefits that PDCR provides. With PDCR you can...

- Gain and understand the right information to tune data warehouse applications, and capture performance data in a single repository
- Identify capacity trends to avoid running out of system resources
- Become more pro-active in data warehouse performance
- Clear up disk in DBC

## PDCR Purpose



The Performance Data Collection and Reporting (PDCR) database is designed to collect data from various Teradata sources for the purpose of analysis and management of Teradata Performance. The data collected for Performance Management is stored primarily in a database called PDCRDATA. This database contains most of the supplemental tables that store historical data and provide reference data that assists the reporting and analysis efforts.

The PDCRDATA tables are loaded and purged using macros contained in an administration database named PDCRADM.

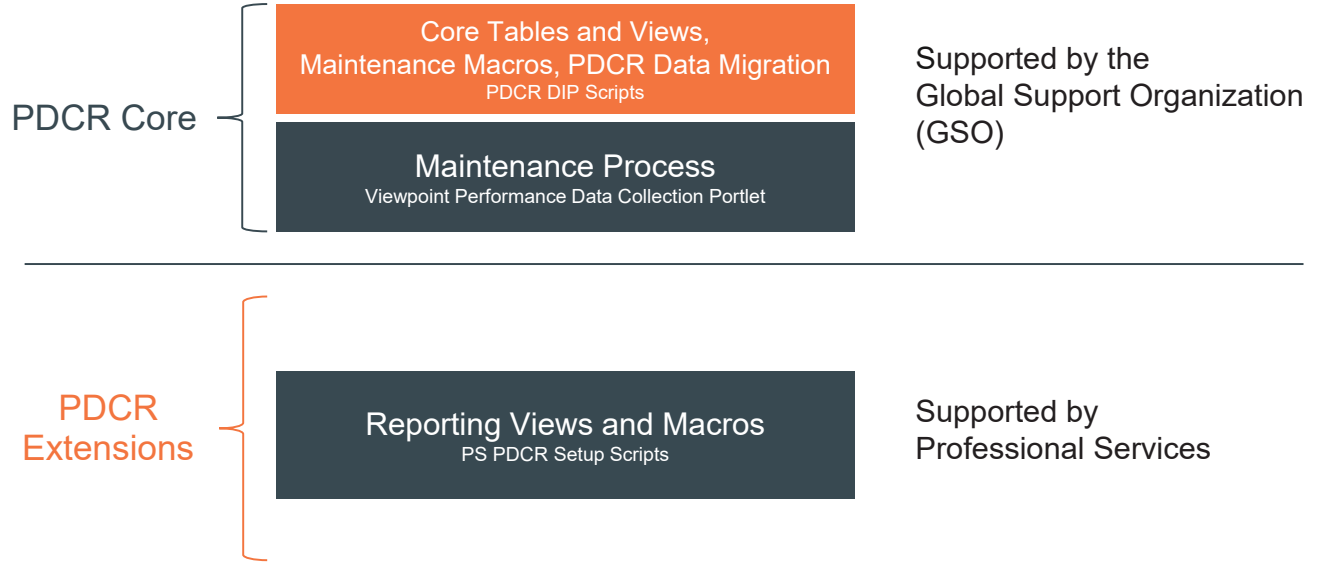
The reporting macros and views are contained in the database PDCRINFO database. This is the information access layer. PDCRINFO needs select with grant rights to two databases (PDCRDATA and dbcmngr) to use its one-to-one views, report views and macros. An analytical user would need both select and execute rights on PDCRINFO.

The PDCR databases are created as children of the PDCRADM user.

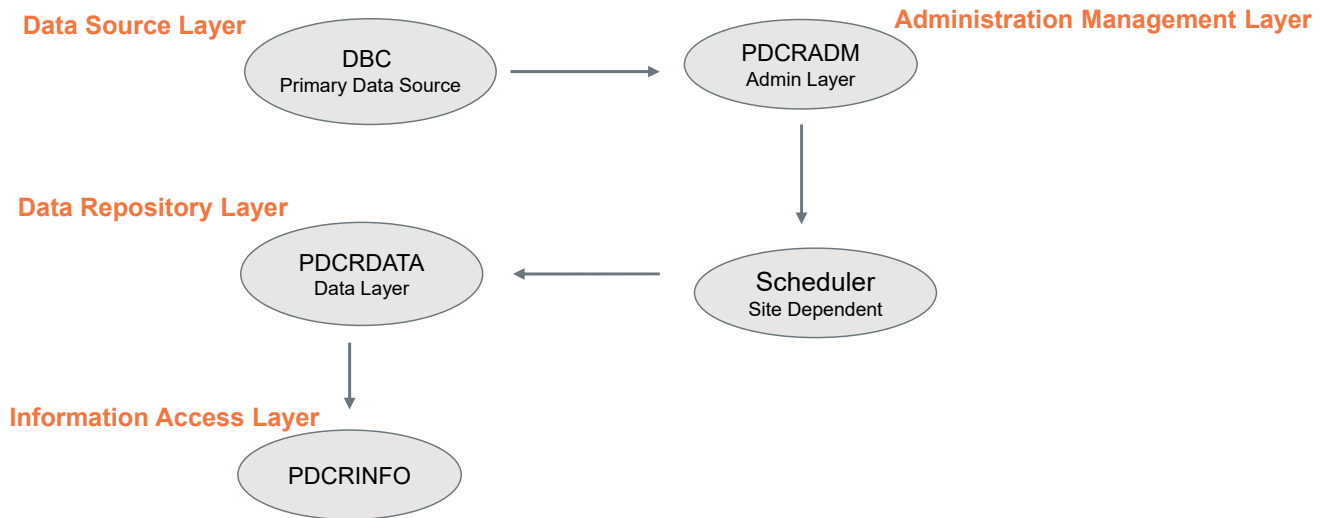
The Performance Data Collection and Reporting Database is the foundation for a historical data collection practice. It is used to design and implement comprehensive Performance Management analysis and reporting.

The PDCRDATA database contains several historical data collection tables, temporary tables, reference tables, and lookup tables. The following diagrams outline the structure of the database as it relates to data sources, views, macros, and tables.

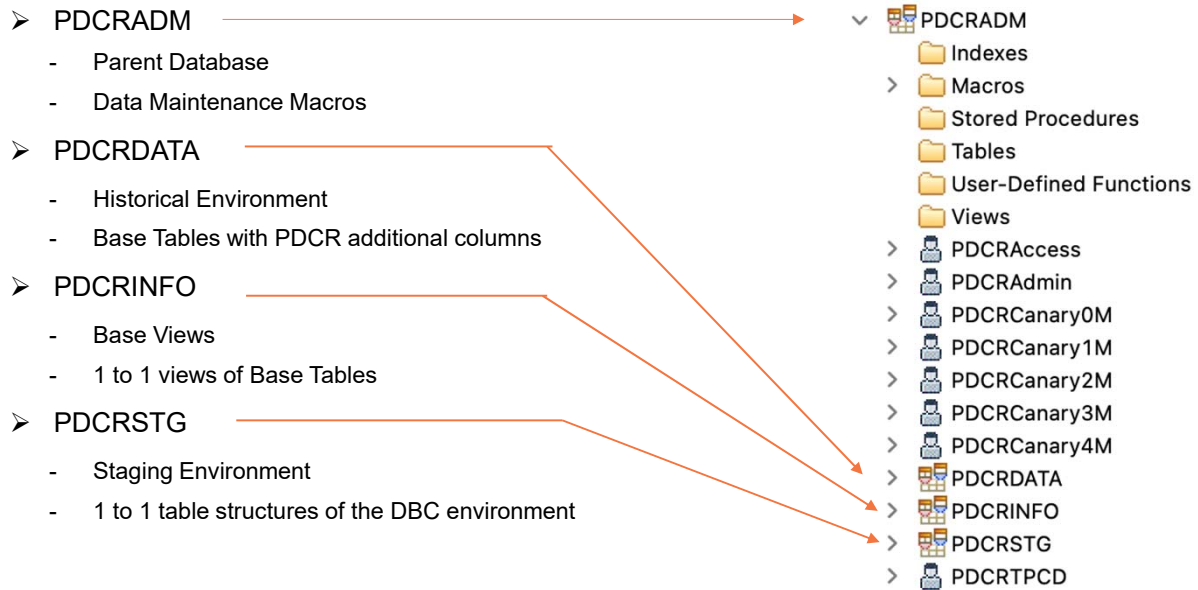
## PDCR Solution Components



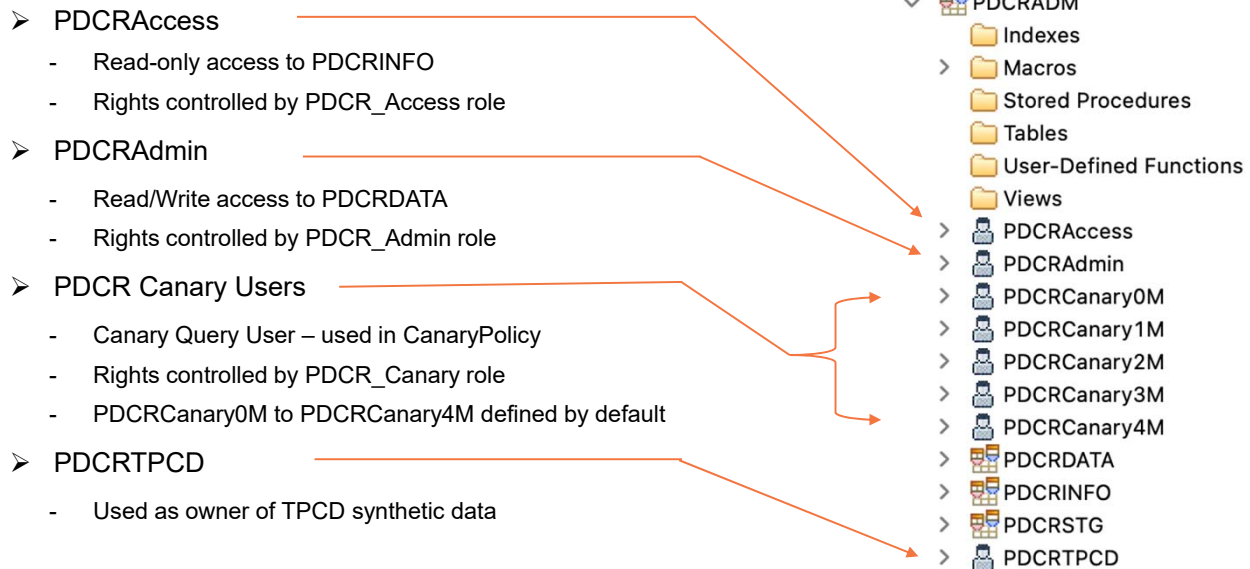
## PDCR Database Structure



## PDCR Base Structures



## PDCR Access and User Logons





## Common DBC and PDCR View

DBC Table	DBC View Referencing DBC Table	PDCR View Referencing DBC Table	PDCR View Referencing Archive Table in PDCRDATA
DBC.AccessLog	DBC.AccessLogV		PDCRINFO.AccessLog_Hst
DBC.DataBaseSpace	DBC.DiskSpaceV		PDCRINFO.DatabaseSpace_Hst
DBC.DataBaseSpace	DBC.DiskSpaceV		PDCRINFO.SpoolSpace_Hst
DBC.DataBaseSpace	DBC.TableSizeV		PDCRINFO.TableSpace_Hst
DBC.DBQLogTbl	DBC.QryLogV	PDCRINFO.DBQLogTbl_DBC	PDCRINFO.DBQLogTbl_Hst
DBC.DBQLSummaryTbl	DBC.QryLogSummaryV	PDCRINFO.DBQLSummaryTbl_DBC	PDCRINFO.DBQLSummaryTbl_Hst
DBC.DBQLUtilityTbl	DBC.QryLogUtilityV	PDCRINFO.DBQLUtilityTbl_DBC	PDCRINFO.DBQLUtilityTbl_Hst
DBC.EventLog	DBC.LogOnOffV		PDCRINFO.LogOnOff_Hst
DBC.ResUsageSawt	DBC.ResSvprView	PDCRINFO.ResUsageSvpr_DBC	PDCRINFO.ResUsageSvpr_Hst
DBC.ResUsageShst	DBC.ResShstView	PDCRINFO.ResUsageShst_DBC	PDCRINFO.ResUsageShst_Hst
DBC.ResUsageSpma	DBC.ResSpmaView	PDCRINFO.ResUsageSpma_DBC	PDCRINFO.ResUsageSpma_Hst
DBC.ResUsageSvds	DBC.ResSvdsView	PDCRINFO.ResUsageSvds_DBC	PDCRINFO.ResUsageSvds_Hst
DBC.ResUsageSvpr	DBC.ResSawtView	PDCRINFO.ResUsageSawt_DBC	PDCRINFO.ResUsageSawt_Hst
DBC.TDWMEventLog	DBC.QryLogEventsV		PDCRINFO.TDWMEventLog_Hst
DBC.TDWMSummaryLog	DBC.QryLogTdwmsumV		PDCRINFO.TDWMSummaryLog_Hst

Common DBC tables along with their corresponding DBC and PDCR views.

## PDCR Data Collection Schedule

Category	Job	Description / What is Archived
Every 10 Minutes Job	Session	Runs every 10 minutes to archive data about user session duration and session counts
Hourly Job	Accounting	Runs every hour to archive data about AMP usage
Daily Jobs	Info Tables	User information, including profiles, groups, and space owners
	Logon/Logoff	Data collected automatically by the system for each user logon or logoff
	TDWM	Logs for workload management functions:
		DBC.TDWMSummaryLog
		DBC.TDWMEventLog
		DBC.TDWMEExceptionLog
		DBC.TDWMEEventHistory
	DBQL	Database query logs, depending on the type of logging you enable
	Disk Space	Data collected automatically on space allocation for databases and tables
	Resource Usage	System-level performance data from the DBC.ResUsage tables
	Access Log	Security data
	Statistics	No data; this job refreshes statistics on PDCR tables
Weekly Job	Maintenance	Runs once a week to check retention values and clean up Data Dictionary logs if needed

# Demo: Performance Data Collection Portlet – Credentials

teradata.

Portlet: **Performance Data Collection** > Button: **Edit Jobs** > Tab: **JOBS** >  
Button: **CREDENTIALS**

⚡ Performance Data Collection

Edit Jobs

JOBS RETENTION PERIODS ALERTS

☒ Enable jobs

Time Zone

Select the time zone that all jobs will be based on

Time zone: (GMT-08:00) Pacific Time (US & Canada)

Every 10 Minutes Job

☒ Enable Session

Starts on the 0 minutes mark

Hourly Job

☒ Enable Accounting

Starts at 00 of every hour

Daily Jobs

☒ Enable Info Tables

☒ Enable Disk Space

☒ Enable Logon/Logoff

☒ Enable Resource Usage

☐

SAVE CANCEL

CREDENTIALS

Credentials

\* Username: PDCRAAdmin

\* Password: \*\*\*\*\*

Account string:

Authentication mechanism: DEFAULT

TEST

SAVE CANCEL

If the PDCRAAdmin Password is changed, you will need to update it here

The **Edit Jobs** view allows you to enable jobs and set the start time for jobs. You access this view by clicking **Edit Jobs** in the summary view.

## Credentials

Credentials are commonly a user name and a password. The **Performance Data Collection** portlet validates credentials to confirm that users have the right, or permissions, to execute jobs against the Teradata Database. For example, if you want to set retention periods or alerts for jobs, you need to have credentials that the system considers valid.

## Demo: Performance Data Collection Portlet – Jobs

Portlet: **Performance Data Collection** > Button: **Edit Jobs** > Tab: **JOBS**

### PDCR\_TenMin.job (Every 10 Minutes)

- SessionInfo

### PDCR\_Hourly.job (Every Hour)

- Acctg

### PDCR\_Daily.job (Every Day, 00:05)

- InfoTables, DBQL, LogonOff, ResUsage, DiskSpace, TDWM, AccessLog, Statistics

### PDCR\_Weekly.job (Sunday, 04:15)

- Purge data according to PDCR\_Table\_Retention
- Adjust PPI ranges according to PDCR\_Table\_Retention
  - Runs every week but only makes adjustments if necessary

Performance Data Collection 1017.10

Edit Jobs

**JOBS** RETENTION PERIODS ALERTS

☒ Enable jobs **Check to enable jobs** CREDENTIALS

**Time Zone**

Select the time zone that all jobs will be based on

Time zone: (GMT-05:00) Eastern Time (US & Canada) **Change Time Zone Appropriately**

**Every 10 Minutes Job**

☒ Enable Session  
Starts on the 0 minutes mark

**Hourly Job**

☒ Enable Accounting  
Starts at 00 of every hour

**Daily Jobs**

☒ Enable Info Tables  
☒ Enable Disk Space  
☒ Enable Logon/Logoff  
☒ Enable Resource Usage  
☒ Enable TDWM  
☒ Enable Access Log

**Check Jobs you wish to enable and their start time**

SAVE CANCEL

The **Edit Jobs** view has three tabs that allow you to edit all aspects of a job:

#### Jobs:

Shows the start times set for each job type (hourly, daily, weekly) and the time zone used to interpret the times specified in the portlet, such as job start times.

#### Retention Periods:

Shows the number of days data is retained before being purged for each job type and table within a job type.

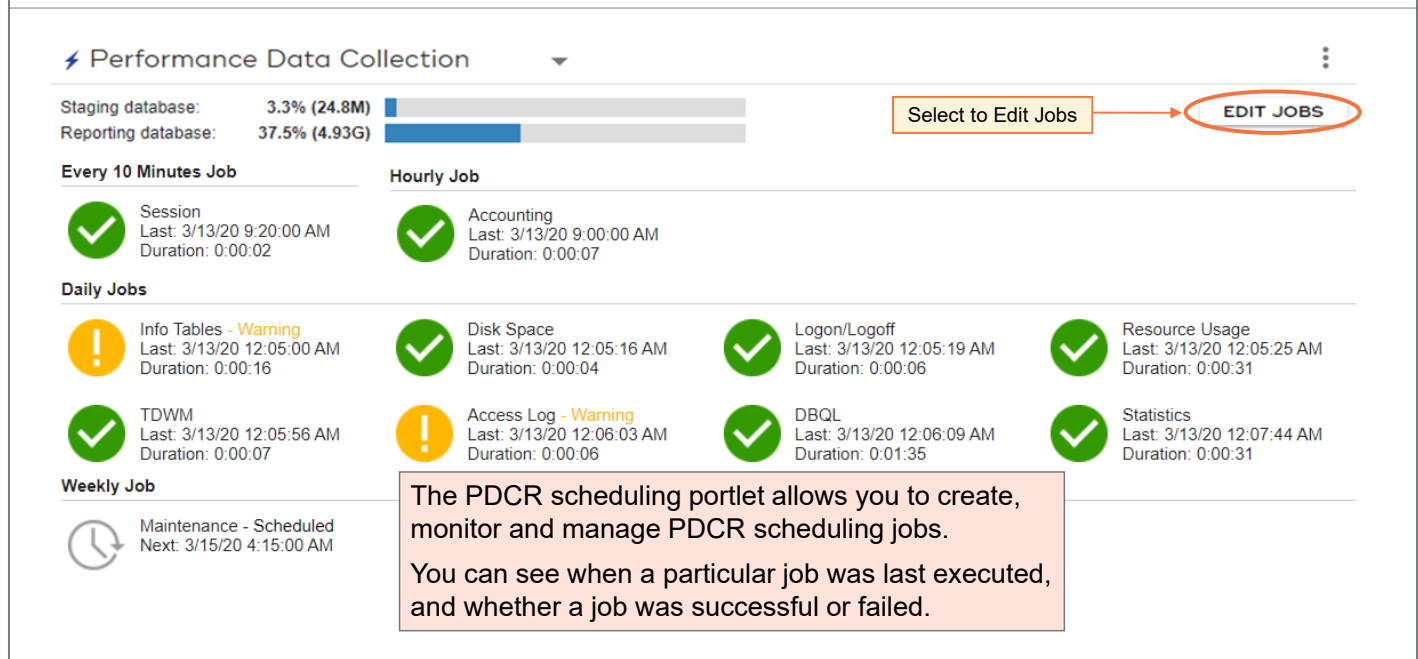
#### Alerts:

Shows created alerts in a summary table. This tab also has an **Actions** list, which contains options for adding job alerts and space alerts.

#### Jobs Tab

The **Jobs** tab on the **Edit Jobs** view displays options you can set for jobs and a button to access the **Credentials** dialog box. If you want to run all of the jobs, select the **Enable jobs** check box. Your selections on this tab include changing the time zone, selecting job categories, and setting job start times.

## Demo: Performance Data Collection Portlet



The **Performance Data Collection** portlet uses jobs to collect performance data from Teradata database sources. This data can then be used to generate performance analysis reports.

The following are just a few of the features of **Performance Data Collection**:

- Collect data automatically within specific time frames. For example, Session data is collected every 10 minutes, and Accounting data is collected every hour.
- Create alerts to notify when a job fails or a database threshold has been met.

The **Performance Data Collection** portlet displays data in two views. The summary view shows a summary of all jobs and current information for any jobs that are enabled. This top-level view for the portlet refreshes every 30 seconds. The details view displays detailed information on a selected job, such as status and information about the tables of data included in the job. This view shows completed information and is not refreshed unless the selected job is currently in progress.

Supported versions of Teradata Database are 15.0 and later. Aster and Hadoop systems are not supported.

## Demo: Alert Presets

Portlet: **Administration** > Button: **Alert Setup** > Setup Options: **Alert Presets** >  
Preset Options: **Action Sets** > Button : **Add new action set**

Setup the Action Sets for PDCR specific alerts

- Action Set Name → PDCR\_FAILURE
- Actions → Email recipients
- Click the Apply button to save your changes

Create the following Action Sets:

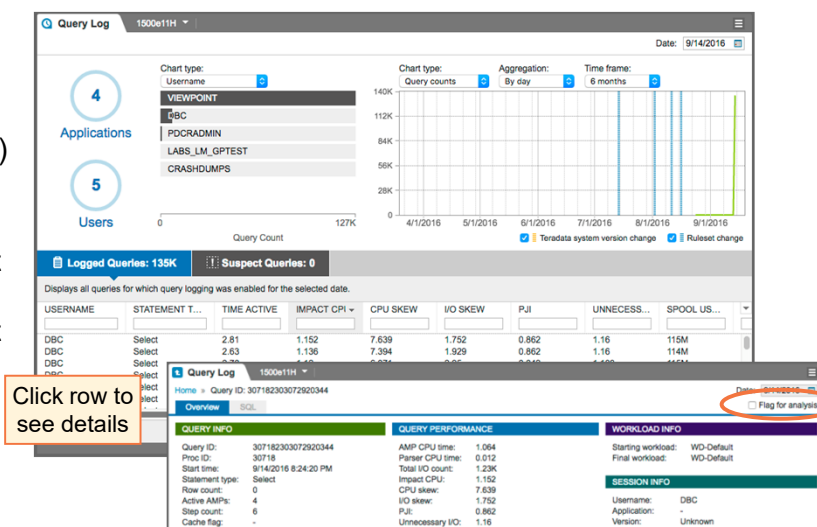
- PDCR\_FAILURE (example shown)
- PDCR\_WARNING

The screenshot shows the 'Alert Setup' portlet with the 'Action Sets' tab selected. The 'Action Set Name' is 'PDCR\_FAILURE'. The 'Times' section has checkboxes for 'Core', 'Evening', and 'Weekend', all of which are checked. The 'Actions' section has a checkbox for 'Include in Alert Viewer' which is checked. Below this, there is a checkbox for 'Email recipients (separate multiple recipients with semicolons)' which is checked. There are radio buttons for 'Bcc' and 'To', with 'To' selected. There are input fields for 'User', 'Role', and 'Email'. The 'Email' field contains 'teradata@ba.email'. At the bottom, there are checkboxes for 'SNAPP', 'SQL', 'BTEQ', and 'Run a program', all of which are unchecked. At the bottom right, there are 'APPLY' and 'RESET' buttons.

## Demo: Query Log Portlet (1 of 2)

View key reports based on the historical DBQL data in the PDCRDATA.DBQLLogTbl\_Hst table

- Must have query logging (DBQL) feature enabled
- Review queries designated as suspect based on thresholds set for the Query Log data collector in the Monitored Systems portlet
- Access information
  - System-wide
  - Application specific
  - User specific



The **Query Log** portlet enables Teradata Database Administrators to view key reports based on the historical DBQL data in the PDCRDATA.DBQLLogTbl\_Hst table in Teradata Database. If the number of DBQL rows exceeds 10 million per day, you can specify the name of a database view in the **Monitored Systems** portlet to load a subset of the query log data to make sure only the most important data is collected.

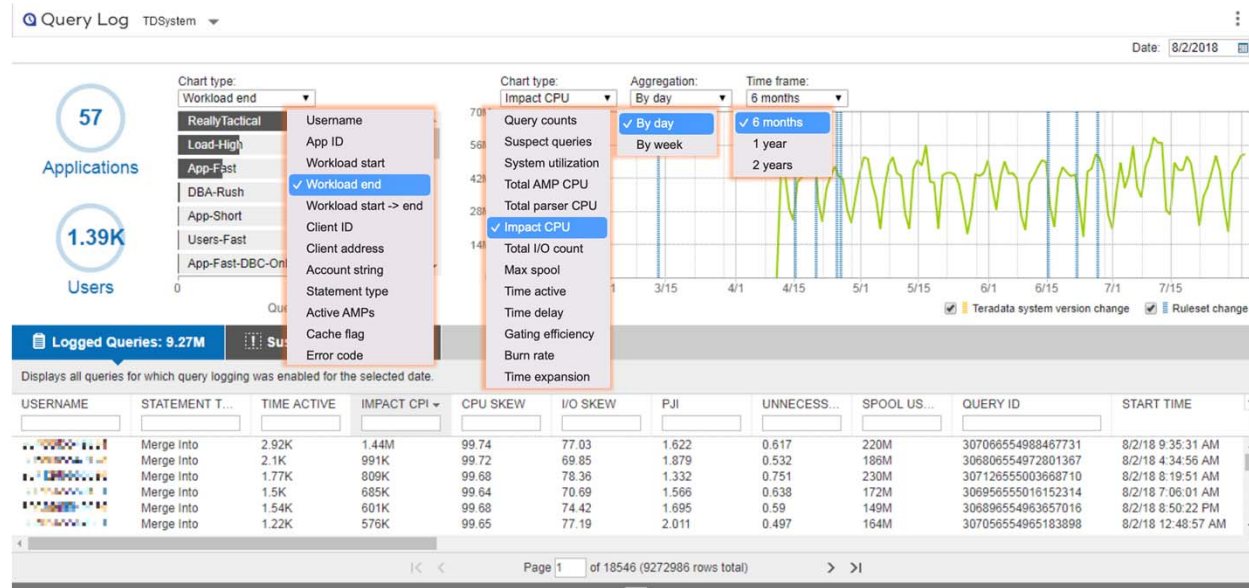
You can review queries designated as suspect based on thresholds set for the Query Log data collector in the **Monitored Systems** portlet, and access system-wide, application specific, and user specific information. You can view key metrics for any date for which query log data has been collected and retained by Viewpoint, and drill down to see details about specific queries. You can also view the trends of key performance indicators, aggregated daily or weekly and going back over a time frame from the last six months to two years.

The **Bar Chart** displays a visual representation of the numbers of queries that fall into each category for the selected metric. When you hover over each bar, a balloon displays the actual number of queries the bar represents. The horizontal **Query Count** line under the chart indicates the number of queries represented by the full width of the chart.

The **Trend Chart** displays a visual representation of the magnitude of the selected metric over the selected time period. The horizontal axis represents the date, and the vertical axis represents the value of the metric selected for display. The green trend line plots the metric values that occurred during the selected time frame, until the day before the current date. As you move your mouse pointer over any point on the trend line, a balloon displays the actual magnitude of the selected metric on that date.

You can view query details by clicking any of the queries listed in the **Query Log** portlet tables. Query details are displayed on an **Overview** and a **SQL** tab. If applicable, **Query Band** and **Errors** tabs are also displayed.

## Demo: Query Log Portlet (2 of 2)



### Configuring the Trend Chart

You can configure and display a trend chart that displays aggregate values for key metrics over a selected time period. You can choose to view the values aggregated by day or by week. You can also choose whether to display the values from the past six months, one year, or two years, through the day before the current date.

1. Select the metric you want to display from the **Chart type** list.
2. Select the aggregation period from the **Aggregation** list.
3. Select the time period display from the **Time frame** list.

The **Logged Queries** tab provides key metrics for queries logged on the selected date.

The **Suspect Queries** tab displays information for all logged queries that are designated as suspect. Suspect queries are those whose values surpass thresholds defined for the Query Log data collector in the **Monitored Systems** portlet.

### Application and User Queries Details View

You can drill down on any of the applications listed in the **Applications** view. You can also drill down on any of the users listed in the **Users** view.

When you click a row for an application or user, details for that application or user are displayed on three tabs: **Overview**, **Queries**, and **Trends**.

If you are viewing application details, you can also select the application version of the information you want to view. Application version is always included when a query is logged with the "version" query band.



## Demo: Application Queries Portlet

View key information about the queries submitted by applications

- Must have query logging (DBQL) feature enabled
- Review application queries designated as suspect based on thresholds set by a Viewpoint Administrator

The screenshot shows the 'Application Queries' portlet. The top part is a table with columns: APPLICATION, VERSION, SYSTEM, QUERY CO..., SUSPECT..., SYSTEM U..., TOTAL AMP..., TOTAL PAR..., and IMPACT C... A red box highlights a row with the text 'Click row to see details'. Below the table is a detailed view of a specific query, showing tabs for Overview, SQL, and Query Band. The Overview tab is active, displaying query information, performance metrics, workload info, and session info. A red circle highlights a 'Flag for analysis' button in the top right corner of the detailed view.

APPLICATION	VERSION	SYSTEM	QUERY CO...	SUSPECT...	SYSTEM U...	TOTAL AMP...	TOTAL PAR...	IMPACT C...
PDCR	15.11.00.04	1500e11H	508	0	0.016	1,548	12.38	2.96
PDCR	15.11.00.05	1500e11H	322	0	0.011	1,056	8.676	1.856
Viewpoint	15.11.00.04	1500e11H	19.6K	0	0.247	45.36	168.1	98.59
Viewpoint	16.00.00.00.Dev	1500e11H	76.3K	0	0.247	183.6	665.9	365.7
Viewpoint	15.11.00.05	1500e11H	13.2K	0	0.247	31.82	111.9	67.1
Viewpoint	15.11.00.02	1500e11H	18K	0	0.247	43.97	116.5	75.54

QUERY INFO		QUERY PERFORMANCE		WORKLOAD INFO	
Query ID:	307192303072914798	AMP CPU time:	0.152	Starting workload:	WD-Default
Proc ID:	30719	Parser CPU time:	0.212	Final workload:	WD-Default
Start time:	9/14/2016 5:48:39 AM	Total I/O count:	718		
Statement type:	Insert	Impact CPU:	0.544		
Row count:	0	CPU skew:	72.06		
Active AMPs:	4	I/O skew:	19.14		
Step count:	19	P/J:	0.212		
Cache flag:	A	Unnecessary I/O:	4.724		

The **Application Queries** portlet enables you to view key information about the queries submitted by applications, based on the historical DBQL data in Teradata Database. You can review application queries designated as suspect based on thresholds set by a Viewpoint Administrator. You can view key metrics for any date for which query log data has been collected and retained by Viewpoint for the applications you have access to, and drill down to see details about specific queries. You can also view the trends of key performance indicators, aggregated daily or weekly and going back over a time frame from the last six months to two years, specific to an application and application version.

### Application Queries Details View

When you click an application listed in the **Application Queries** view, the system and application are listed at the top, and you can select the version of the application from the **Version** list.

The **Overview** tab includes a **Summary** section that lists key metric values and a **Counts** section that displays bar charts representing query counts for a selected metric.

The **Queries** tab enables you to view two subtabs:

- The **Logged Queries** tab provides key metrics for queries logged on the selected date.
- The **Suspect Queries** tab displays information for all logged queries that are designated as suspect. Suspect queries are those whose values surpass thresholds defined for the Query Log data collector in the **Monitored Systems** portlet.

# Performance Data Collection Portlet – Data Retention

teradata.

Portlet: **Performance Data Collection** > Button: **Edit Jobs** > Tab: **RETENTION PERIODS**

Data Retention periods are managed in the Data Collection Portlet. The DCRDATA.PDCR\_Table\_Retention table is the underlying table that stores and controls the retention periods settings.

The values in the Data Collection Portlet will override existing values in the underlying table during the week maintenance job.

Performance Data Collection td17.10

Edit Jobs

JOB RETENTION PERIODS ALERTS

Resource Usage	ResUsageSldv_Hst	90
Resource Usage	ResUsageSpdsk_Hst	90
Resource Usage	ResUsageSpma_Hst	1461
Resource Usage	ResUsageSps_Hst	180
Resource Usage	ResUsageSum10_Hst	1461
Resource Usage	ResUsageSvdk_Hst	180
Resource Usage	ResUsageSvpr_Hst	90
Session	MonitorSession_Hst	400
Session	TDWMThrottleStats_Hst	90
Session	TDWMUtilityStats_Hst	90
TDWM	TDWMEventHistory_Hst	400
TDWM	TDWMEventLog_Hst	400
TDWM	TDWMEExceptionLog_Hst	400
TDWM	TDWMSummaryLog_Hst	400

RESTORE DEFAULTS

SAVE CANCEL

Make Data Retention Period changes in days as Required

Please note that changes are not written and reflected until the Maintenance Job has been executed

## Retention Periods Tab

The **Retention Periods** tab on the **Edit Jobs** view displays retention information for all jobs and the tables associated with those jobs. A retention period is the number of days a table retains its data before that data is purged.

The **Weekly Job** purges data that exceeds the retention limits set for each table. Data is purged from a table on a First-In-First-Out basis.

The **Cleanup Schedule** in the **Monitored Systems** portlet allows you to schedule table cleanup to occur after **Performance Data Collection** daily jobs have completed successfully. This makes sure the data is archived to the **Performance Data Collection** database before the cleanup occurs.

# Performance Data Collection Portlet – Adding Alerts

teradata.

Portlet: **Performance Data Collection** > Button: **Edit Jobs** > Tab: **ALERTS** >  
Button: **Actions**

The screenshot shows the 'Performance Data Collection' portlet in the 'ALERTS' tab. The main area lists existing alerts with columns for 'Enabled', 'Alert Name', and 'Type'. Two alerts are listed: 'PDCR\_FAILURE\_10\_Minute\_Job' and 'PDCR\_STAGING\_LOW\_SPACE'. To the right, there are two side panels: 'Add Space Alert' and 'Edit Job Alert'. Annotations with arrows point to specific UI elements: 'Click Actions Button to add Alert' points to the 'Actions' dropdown; 'Click Pen to edit Alert' points to the edit icon; 'Click Trash Can to delete Alert' points to the delete icon; 'Add Job Alert' and 'Add Space Alert' point to their respective buttons. A text box on the left explains the purpose of alerts.

Alerts are used to **notify** interested parties when a job **fails to complete** successfully or when the **space** for a database reaches a specific **threshold**

Click **Actions** Button to add Alert

Click **Pen** to edit Alert

Click **Trash Can** to delete Alert

Add Job Alert

Add Space Alert

SAVE CANCEL

**Add Space Alert**

Create an alert if a database space threshold is met.

Alert name: PDCR\_SPACE\_PDCRC ☐ Enable alert

Severity: Medium

**Alert Rules**

Database threshold ☒ Reporting database ☐ Staging database

Alert Action

Do not run twice in 0 minutes

Message:

**Edit Job Alert**

Create an alert if a job does not complete successfully.

Alert name: PDCR\_FAILURE\_10\_M ☒ Enable alert

Severity: High

**Jobs:**

Every 10 Minutes Job ☒ Session ☐ Hourly Job ☐ Accounting

Daily Jobs ☐ Info Tables ☐ Disk Space ☐ Logon/Logoff ☐ Resource Usage ☐ TDWM ☐ Access Log ☐ DBQL ☐ Statistics

Weekly Job ☐ Maintenance

Alert Action

## Alerts Tab

Alerts are used to notify interested parties when a job fails to complete successfully or when the space for a database reaches a specific threshold.

The **Alerts** tab in the **Edit Jobs** view displays a summary of existing alerts, including their enabled status and their type: job or space. In this tab, you can edit, delete, and add alerts.

Alerts created in the **Performance Data Collection** portlet can be displayed in the **Alert Viewer** portlet. This portlet displays alert type, severity, source, duration, error messages, and the action taken when the alert is triggered. The option to view **Performance Data Collection** alerts in the **Alert Viewer** portlet is set in the **Alert Setup** portlet.

## Proposed Alert Texts

Alert Name	Jobs	Alert Action	Alert Text
PDCR_FAILURE_10_Minute_Job	Session	PDCR_FAILURE	VIEWPOINT SYSTEM ALERT: (System Name) PDCR Failure with the 10 minute Job
PDCR_FAILURE_Daily_Job	Info Tables, Disk Space, Logon/Logoff, Resource Usage, TDWM, Access Log, DBQL, Statistics	PDCR_FAILURE	VIEWPOINT SYSTEM ALERT: (System Name) PDCR Failure with the Daily Job
PDCR_FAILURE_Weekly_Maint_Job	Maintenance Process	PDCR_FAILURE	VIEWPOINT SYSTEM ALERT: (System Name) PDCR Failure with the Weekly Job
PDCR_FAILURE_Hourly_Maint_Job	Accounting	PDCR_FAILURE	VIEWPOINT SYSTEM ALERT: (System Name) PDCR Failure with the Hourly Job
PDCR_SPACE_PDCRDATA	Space Monitoring	PDCR_WARNING	VIEWPOINT SYSTEM ALERT: (System Name) PDCRDATA Database Space is greater than 80%
PDCR_SPACE_PDCRSTG	Space Monitoring	PDCR_WARNING	VIEWPOINT SYSTEM ALERT: (System Name) PDCRSTG Database Space is greater than 80%

NOTE: Modify Alerts accordingly to distinguish between different systems, if Viewpoint is monitoring multiple systems.

teradata.

Click on icons to drill into details

# Performance Data Collection Portlet – Job Drill Down (1 of 2)

Portlet: **Performance Data Collection** > Button: **Any Job Icon**> Tab: **TABLE DETAILS** >  
Button: **Actions**

Performance Data Collection

Info Tables Job | Daily

Run job failed:  
[Teradata Database] [TeraJDBC 16.20.00.01] [Error 2802] [SQLState 23000] Duplicate row error in PDCRDATA.UserInfo.

Status: **Failed**  
Start time: 2/28/20 12:05:01 AM  
Stop time: 2/28/20 12:05:13 AM  
Duration: 0:00:12  
Total row count: 0

Run Job  
View Job History

TABLE DETAILS SQL LOG

TABLE	START	DURATION	STATUS	ROW COUNT
TASMMNames_Hst	2/28/20 12:05:02 AM	0:00:11	Failed	
Userinfo	2/28/20 12:05:02 AM	5:26:33	In progress	
GroupInfo	2/28/20 12:05:02 AM	5:26:33	In progress	
SpaceOwnerInfo	2/28/20 12:05:02 AM	5:26:33	In progress	

Last run's results

By selecting a given job, you can drill down to view the last execution, rerun a job, and view the job history

4 rows total

# Performance Data Collection Portlet – Job Drill Down (2 of 2)

teradata.

Portlet: **Performance Data Collection** > Button: **Any Job Icon**> Tab: **SQL LOG**

Performance Data Collection

Info Tables Job | Daily

Run job failed:  
[Teradata Database] [TeraJDBC 16.20.00.01] [Error 2802] [SQLState 23000] Duplicate row error in PDCRDATA.User

Status: **Failed**  
Start time: 2/28/20 5:35:58 AM  
Stop time: 2/28/20 5:36:10 AM  
Duration: 0:00:11  
Total row count: 0

START	DURATION	SQL	PARAMETERS	ERROR CO...
2/28/20 5:36:10 AM	0:00:00	INSERT INTO PDCR	'LoadInfoTbls', 'PDCRDATA', 'In	
2/28/20 5:36:00 AM	0:00:10	EXECUTE PDCRAD		2802
2/28/20 5:35:59 AM	0:00:00	INSERT INTO PDCR	'Loadinfotbls', 'PDCRDATA', 'TA	
2/28/20 5:35:59 AM	0:00:00	INSERT INTO PDCR	'Loadinfotbls', 'PD	
2/28/20 5:35:59 AM	0:00:00	INSERT INTO PDCR	'Loadinfotbls', 'PD	

Click row to drill into SQL details

By selecting a given job, you can drill down to **view the last execution**, rerun a job, and view the job history

Performance Data Collection 1510411C

Resource Usage Job | Daily | SQL Details

Start: 7/18/16 12:06:32 AM Duration: 0:00:00

SQL:

```
SELECT COUNT(*)  
FROM TABLE ( TDWM.TDWMInquire()) INQ  
WHERE RuleSetID <= 0 ;
```

Parameters:

Error message [Error 7504]:  
### Error querying database. Cause: java.sql.SQLException: [Teradata Database] [TeraJDBC 16.00.00.12] [Error 7504] [SQLState HY000] in UDF/XSP/UDM tdwm.TDWMInquire: SQLSTATE U0004: ### The error may exist in com/teradata/viewpoint/csp/pdc/rda/mysql/ResourceUsageJob-Teradata.xml ### The error may involve defaultParameterMap ### The error occurred while setting parameters ### SQL: SELECT COUNT(\*) FROM TABLE (TDWM.TDWMInquire()) INQ WHERE RuleSetID <= 0 ; ### Cause: java.sql.SQLException: [Teradata Database] [TeraJDBC 16.00.00.12] [Error 7504] [SQLState HY000] in UDF/XSP/UDM tdwm.TDWMInquire: SQLSTATE U0004: ; uncategorized SQLException for SQL ; SQL state [HY000]; error code [7504]; [Teradata Database] [TeraJDBC 16.00.00.12] [Error 7504] [SQLState HY000] in UDF/XSP/UDM tdwm.TDWMInquire: SQLSTATE U0004: ; nested exception is java.sql.SQLException: [Teradata Database] [TeraJDBC 16.00.00.12] [Error 7504] [SQLState HY000] in UDF/XSP/UDM tdwm.TDWMInquire: SQLSTATE U0004:

# Performance Data Collection Portlet – View Job History

teradata.

Portlet: **Performance Data Collection** > Button: **Any Job Icon**

Info Tables Job History		Daily		Last 14 days ▾	
START ▾	DURATION	STATUS	ROW COUNT	STOP TIME	▾
2/28/20 5:35:58 AM	0:00:02	In progress	0		
2/28/20 12:05:01 AM	0:00:12	Failed	0	2/28/20 12:05:13 AM	← Last run's result
2/27/20 12:05:01 AM	0:00:12	Failed	0	2/27/20 12:05:12 AM	
2/26/20 12:05:01 AM	0:00:12	Failed	0	2/26/20 12:05:13 AM	
2/25/20 12:05:00 AM	0:00:12	Failed	0	2/25/20 12:05:12 AM	
2/24/20 12:05:01 AM	0:00:12	Failed	0	2/24/20 12:05:13 AM	
2/23/20 12:05:01 AM	0:00:12	Failed	0	2/23/20 12:05:12 AM	
2/22/20 12:06:18 AM	0:00:12	Failed	0	2/22/20 12:06:31 AM	

By selecting a given job, you can drill down to view the last execution, rerun a job, and [view the job history](#)



## Summary

In this module we covered how to:

- Explain the purpose for using the Database Query Log (DBQL)
- Identify the tables and views that make up the DBQL facility
- Describe the SQL syntax for query logging and its collection options.
- Find information about current query logging rules
- Explain the purpose for using Performance Data Collection and Reporting (PDCR)
- Describe the PDCR Database Structure
- Explain the purpose for the Performance Data Collection portlet in Viewpoint

This slide summarizes some important concepts regarding this module.



## Module 3: Workload Management

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

- Define Workload Management and explain its benefits
- Discuss the concepts and features of Workload Management
- Explain the purpose for the Workload Management portlets in Viewpoint



## Topics

- Workload Management Overview
- Workload Management Features
- Viewpoint Portlets



## Current Topic – Workload Management Overview

- **Workload Management Overview**
  - Limited System Resource Challenge
  - Why Workload Management
  - What is a Workload?
  - What is Workload Management?
  - There are Two Workload Management Offerings
- Workload Management Features
- Viewpoint Portlets



## Limited System Resource Challenge

Resources Challenges needs to manage below

- **How** to ensure that high-priority work always has access to the necessary systems resources?
- **Prevent** lower-priority work from taking resources
- **Protect** your high-priority work on degraded systems
- **Prioritize** query appropriately
- **Minimize** flood of requests from a single application
- **Automatically** deal with a runaway query



Limited System Resources means you have to address some questions...

- How do you ensure that high priority work always has access to the systems resources that it needs?
- How do you stop lower priority work from taking resources away from high priority work?
- How do you protect your high priority work when the system's health condition becomes critical?
- How do you make sure that a query always runs with the appropriate priority?
- How do you automatically deal with a runaway query?

## Why Workload Management

Workload Management provides the following benefits:

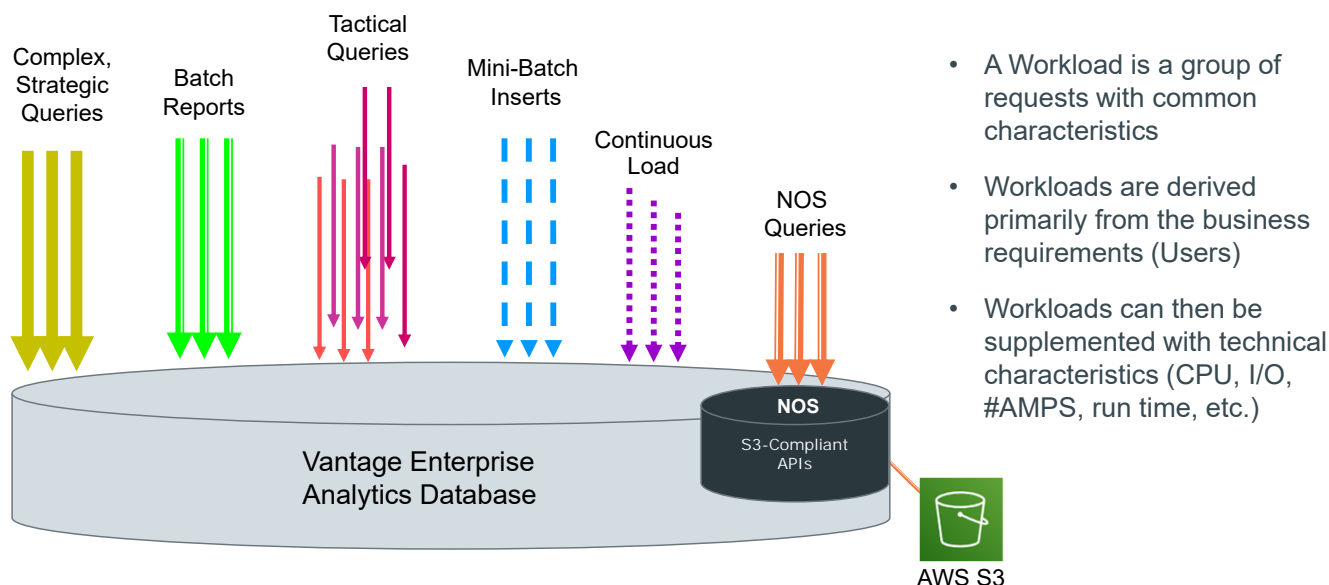
- **Stabilizes** response times of important work
- **Prioritizes** and protects known, proven work from poorly written requests
- **Prioritizes** work based on business operations and rhythms
- **Manages** system **resources** by type of work or department
- **Techniques** to manage **unexpected** situations



Workload Management provides the following benefits:

- Stabilizes response times of important work
- Prioritizes and protects known, proven work from poorly written requests
- Prioritizes work based on business operations and rhythms
- Manages system resources by type of work or department
- Manages resource use
- Offers techniques to manage unexpected situations

# What is a Workload?



A workload represents a portion of the queries that are running on a system. A Workload Definition (WD) is a workload grouping and its operating rules to assist in managing queries. It consists of:

**Classification Criteria:** criteria to determine which queries belong to the workload. This criteria defines characteristics which are detectable prior to query execution. This is also known as the "who", "where", and "what" criteria of a query. For example, "who" may be an account name, "where" is the database tables being accessed, and "what" may be the type of statement (UPDATE) being executed.

**Exception Criteria:** criteria to specify "abnormal" behavior for queries in this workload. These criteria are only detectable after a query has begun execution. If the exception criteria are met, the request is subject to the specified exception action which may be to lower the priority or abort the query.

**Operating Periods:** a description of hours of the day and/or days of the week (or month). Directives may be specified for exception handling and Priority Scheduler settings can be changed for each operating period.



## What is Workload Management?

Workload Management is a goal-oriented, workload-centric, automated management of mixed workloads.

With Workload Management users create rules that can:

- Limit sessions, utilities, and requests – delaying or rejecting them when limits are reached
- Divide system resources among different types of requests
- Send alerts or change the workload of active requests that exceed limits
- Monitor time of day and system resources and react in predefined ways



The Workload infrastructure is a Goal-Oriented, Automatic Management and Advisement technology in support of performance tuning, workload management, capacity planning, configuration and system health management. Workload Management features greatly improves system management capabilities, with a key focus being to reduce the effort required by DBAs, application developers and support engineers through automation. In addition, Workload Management provides many more system management monitoring and analysis capabilities than previously available to Teradata users. Business-driven Service Level Goals can be specified and monitored against for a quick-and-easy evaluation of performance when using the new monitoring capabilities. Users of Workload Management will realize improved response time consistency and predictability of their workloads.

## Workload Management Offerings

### Teradata Active System Management (TASM)

- Performs full workload management
- Gives administrators the ability to prioritize workloads, tune performance, and monitor and manage workload and system health
- Automates tasks that were previously labor-intensive for application DBAs and operational DBAs

### Teradata Integrated Workload Management (TIWM)

- Provides basic workload management capabilities to customers without full TASM. TIWM offers a subset of TASM's capabilities

Teradata Integrated  
Workload Management  
(TIWM)



Teradata Active  
System Management  
(TASM)



Benefits of Workload Management include:

- Support business operations priority decisions
- Stabilize response times of the critical work
- Increase throughput
- Protect known, proven work from impact by unknown, adhoc unpredictable queries
- Give priority to proven, good performing queries
- Automatically manage prioritization based on processing periods or system health conditions

## Current Topic – Mixed Workload Features

- Workload Management Overview
- **Workload Management Features**
  - Virtual Partitions
  - Workload Classification Criteria Types
  - Workload Management Methods – TIWM/TASM
  - Filters
  - Arrival Rate Meter
  - Throttles
  - State Matrix
  - Exceptions
  - Query Management Architecture
  - Workload Management Offering Comparison
- Viewpoint Portlets

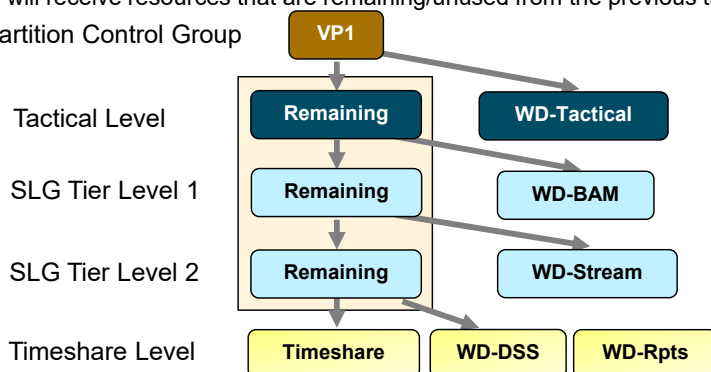


## Virtual Partitions

- Virtual Partitions are intended for sites supporting multiple **geographic entities** or **business units**
- By default, a single Virtual Partition exists named **Standard** and **maximum 10** can be defined.
- The recommendation is to **start with one** Virtual Partition
- Resources **“flow”** from the top tier to the bottom tier

- Each tier will receive resources that are remaining/unused from the previous tier

Virtual Partition Control Group



The first level in the priority hierarchy that the administrator can interact with is the virtual partition level. All platforms default to one virtual partition, and TASM offers up to 10 virtual partitions.

A virtual partition represents a collection of workloads. A single virtual partition exists for user work by default, but up to 10 can be defined with TASM.

A single virtual partition is expected to be adequate to support most priority setups. Multiple virtual partitions are intended for platforms supporting several distinct business units or geographic entities that require strict separation.

A single Virtual Partition exists for user work by default, but up to 10 may be defined, if needed. Due to improvements in Priority Scheduler capabilities, a single Virtual Partition is expected to be adequate to support most priority setups. Multiple virtual partitions are intended for platforms supporting several distinct business units or geographic entities that require strict separation.

Virtual Partitions provide the ability to manage resources for groups of Workloads dedicated to specific divisions of the business. When a new Virtual Partition is defined, the administrator will be prompted to define a percentage of the Teradata Database resources that will be targeted for each, from the Viewpoint Workload Designer screens. This percent will be taken out of the percent of resources that flow down through the User Control Group.

Once defined, each of these Virtual Partitions can contain their own Control Group hierarchies. Each Virtual Partition hierarchy can include all allowable priority levels from Tactical to Timeshare.

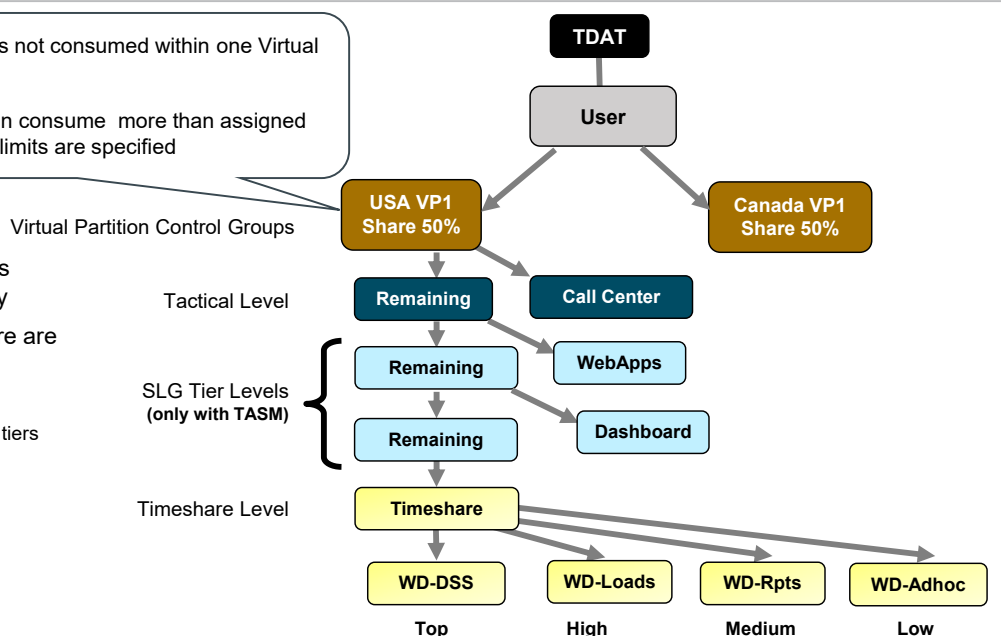
The share percent given to a Virtual Partition and will determine how the CPU is initially allocated across multiple Virtual Partitions. If there are spare cycles not able to be used within one Virtual Partition, another Virtual Partition will be able to consume more than its defined percent specifies.

For example, things like space accounting execute in the internal Virtual Partition, as do other activities not associated directly with user work, but that are important to get done. In addition, some internal activities that used to run in Performance Groups L and R in SLES 10 will now run in the internal Virtual Partition.

## Control Groups and Workloads

- If there are spare resources not consumed within one Virtual Partition
- Another Virtual Partition can consume more than assigned share percent unless hard limits are specified

- The Hierarchy of Control Groups determines the Workload priority
- Under each Virtual Partition there are levels (tiers) of service:
  - a) Tactical Tier (Tier #0)
  - b) 5 Service Level Goals (SLG) tiers
  - c) Time Share tier (last Tier)



### Tactical Recommendations

- Only assign Workloads that support highly-tuned, very short queries to Tactical.
- Do not assign Workloads that support load utilities into Tactical.
- Rely on reasonable exception thresholds to demote queries with non-tactical characteristics.
- Monitor tactical exceptions regularly, adjust the exception thresholds, when needed.
- Don't increase the AMP worker task reserve count above zero unless a shortage of AMP worker tasks is impacting the tactical performance.

If specifying reserved AWTs, set the reserved count for the worst case AMP usage by all tactical query Workloads combined, including tactical Workloads across all Virtual Partitions.

Avoid placing tactical Workloads in Virtual Partitions with an inadequate percent of resources.

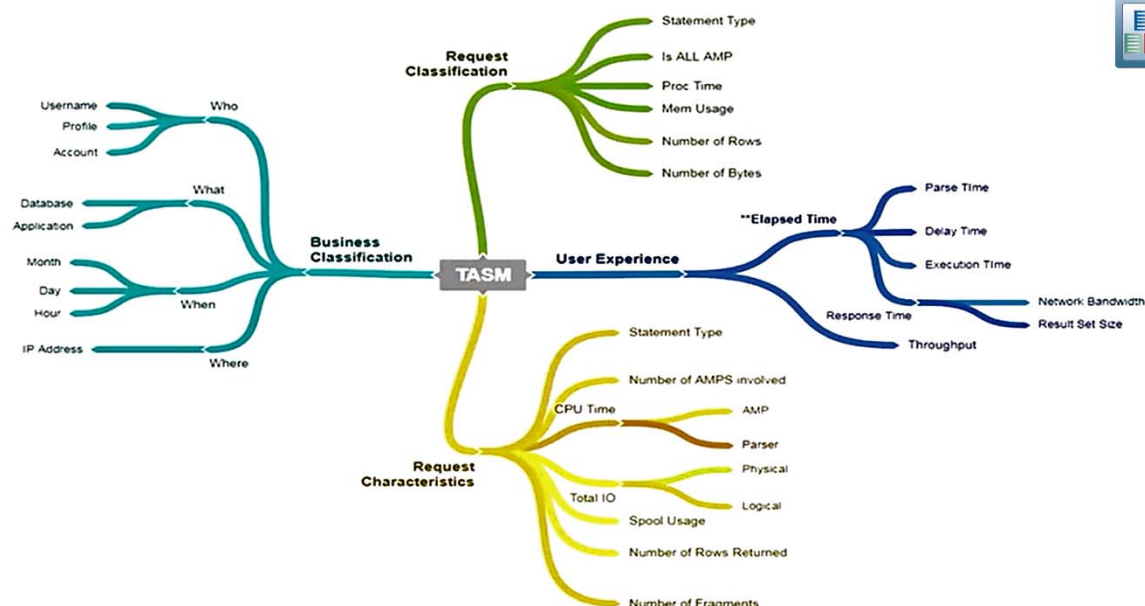
### SLG Tier Recommendations

- SLG Tiers are intended for high priority work that is associated with response time expectations.
- Use only a single SLG Tier if only a few workloads fall into this category.
- If a large number of Workloads with widely-varying priorities fall into this category, place the Workloads with the more critical service level expectations on the higher SLG Tiers.
- If more than one SLG Tier supports Workloads, attempt to define smaller Workload Share Percents on higher Tiers, to allow a more predictable level of resources.

### Timeshare Recommendations

- If SLG Tiers are not used, place Workloads in the Top Access Level that are very high priority.
- If a penalty box Workload exists when migrating to the new Priority Scheduler, add that Workload into the Timeshare Low Access Level.
- For more predictable priority differentiation in Timeshare, keep the decay option turned off.

## Workload Classification Criteria Types



Workload Management now allows queries to be classified using multiple criteria, whereas prior to Workload Management only Account ID was used.

We will start with workload classification or classification criteria which consists of the following:

- Request Source – username, account name, account string, profile, application, client IP, client ID
- Target – database, table, macro, view, or stored procedure  
Subcriteria: Full Table Scan, Join Type, Min Step Row Count, Max Step Row Count, or Min Step Time
- Query Characteristics – Statement type, AMP Limits, Step Row Count, Final Row Count, Estimated Processing Time, Min Step Time, Join Type, or Full Table Scan
- QueryBand – User-define metadata about the query
- Utility – FastLoad, FastExport, MultiLoad, or Backup Utilities

## Workload Classification

- In Workload Management every query is assigned to a Workload
- Queries are assigned to a Workload via their classification criteria
- If a query is not classified into a user-defined Workload, it will be assigned to a default Workload, called **WD\_Default**
- Classification determines the Workload's priority and operation rules it must adhere to

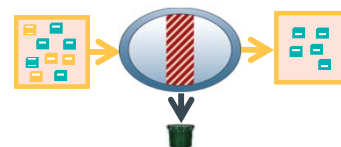


## Pre-Execution Controls – Filters

- Filters are applied system-wide and **reject a query** before the query starts running
  - Based on the classification criteria. The classification criteria are the same as workloads
  - Based on predefined system states (environment / health conditions)
- Use Filters carefully after an evaluation period in “**Warning Only**” mode
- System Bypass** privileges can be applied based on username, account name, account string, or profile

### Example

- Enable a filter that rejects long-running requests when the system is in a degraded state
- A request accesses a large, detailed table using an unconstrained product join
- A request uses a full-table scan on very large, detailed data tables, such as call detail data or deep history data, and on Native Object Store tables



NOTE: Use filters with **caution** and apply them selectively. It may be inappropriate to reject requests in an ad hoc environment.

Filters are system-wide and allows the DBA to reject queries before they begin running based on the classification criteria.

If it is determined that a certain type of request should never run during the day, for example, system filter rules are able to enforce that.

In order to restrict the impact and scope of a filter to a selective number of deserving queries, the administrator can apply a variety of qualifying criteria. These are the same criteria choices that can be used for workload classification purposes.

Filter rules need to be used with caution and forethought, and applied very selectively, as rejecting queries is a strong action to take and may be considered as inappropriate in an ad hoc environment. “Warning Only” option can be used for testing as queries are logged but not rejected (can be used for testing new filters).

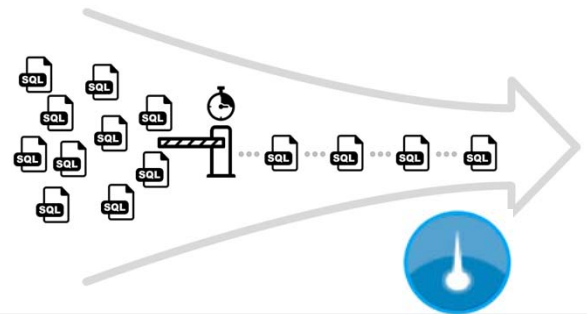
System Bypass privileges can be applied based on username, account name, account string, or profile

## Pre-Execution Controls – Arrival Rate Meter

- ARM (**TASM ONLY**) rules provides
  - A means to regulate the flow of SQL requests admitted by TASM into the system
  - This rule specifies the max number of SQL requests per time unit that TASM will admit into the system

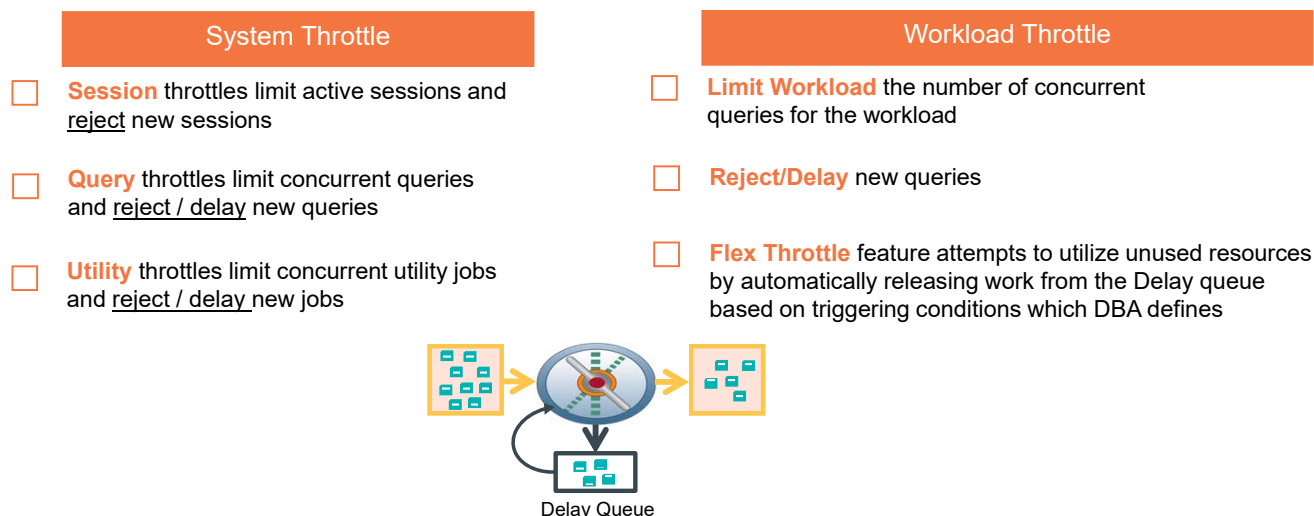
### Example

- Flood of requests from a specific application monopolizes the front of the throttle delay queue
- Complex requests which consume lot of resources to allow only a few requests each hour during weekdays and a higher limit on weekends.
- Certain applications which generate a very large number of table scans



## Pre-Execution Controls – Throttles

- Throttles can be used to **limit** the number of **active queries**
- System Bypass** privileges can be applied based on username, account name, account string, or profile



Controlling the number of concurrent request is by far the most popular use of throttles at Vantage sites today. When a throttle rule is active, a counter is used to keep track of the number of requests that are active at any point in time among the queries under control of that rule. When a new query is ready to begin execution, the counter is compared against the limit specified within the rule. If the counter is below the limit, the query runs immediately; if the counter is equal to or above the limit, the query is either rejected or placed in a delay queue. Most often throttles are set up to delay queries, rather than reject them.

Once a query which has been delayed is released from the delay queue and begins running, it can never be returned to the delay queue. Throttles exhibit control before a query begins to execute, and there is no mechanism in place to pull back a query after it has been released from the delay queue.

Starting in Teradata 15.10 there is a new option to order the delay queue by workload priority. A priority value is calculated for each workload based on the workload management method assigned to the workload. Requests in the delay queue are ordered from high to low based on the workload value. Ties are ordered by start time. If the option to order the delay queue by workload priority is not selected, the queue is ordered by query start time. In that case queries are released from the delay queue in first-in-first-out (FIFO) order if all applicable throttles are within limits.

Two types of throttles are available; system and workload throttles.

System throttles include session throttles which are used to limit the number of active sessions and reject any new sessions and the user must resubmit the query. Query throttles limit the number of concurrent queries and will reject or delay new queries. And last utility throttles limit the number of concurrent utility jobs and will reject or delay new utility jobs.

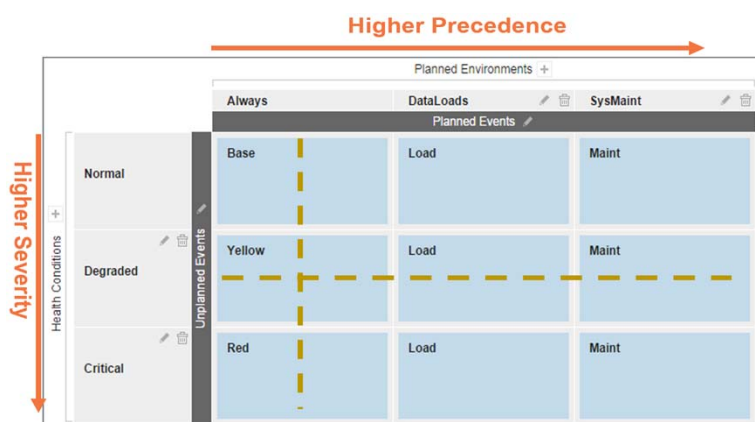
The other type of throttles is the workload throttle which is used to limit the number of concurrent queries for the workload and will reject or delay any new queries for that workload.

And the same Bypass privileges discussed with filters can also be applied to throttles.

The Flex Throttle feature attempts to utilize unused resources by automatically releasing work from the Delay queue based on triggering conditions a DBA defines.

# State Matrix

The State Matrix primarily consists of two dimensions



## Planned Environment

- The kind of work the system is expected to perform
- Indicative of planned time periods or operating windows for particularly critical applications

## Health Condition (TASM ONLY)

- Unplanned events that include
  - System performance and availability considerations, such as AMPs in flow control
  - %ges of nodes down at system startup



Generally, workloads do not generate consistent demand, nor do they maintain the same level of importance throughout the day/week/month/year. For example, suppose there are two workloads: A query workload and a load workload. Perhaps the load workload is more important during the night and the query workload is more important during the day. Or perhaps there are tactical workloads and strategic workloads, and when the system is somehow degraded, it is more important to assure tactical workload demands are met, at the expense of the strategic work. Or finally, a year-end accounting workload may take precedence over all other workloads when present. The State Matrix allows a transition to a different working value set to support these changing needs.

The State Matrix allows a simple way to enforce gross-level management rules amidst these types of situations. In TASM it is a two-dimensional matrix, with Operating Environments and System Conditions represented, with the intersection of any Operating Environment and System Condition pair being associated with a State with different rule set working values. Multiple Operating Environment and System Condition pairs can be associated with a single State.

- State – identifies a set of Working Values and can be associated with one or more intersections of a Health Condition and Planned Environment
- Current State – the intersection of the current Health Condition and Planned Environment

## During Execution Controls – Exceptions

Exception Rules are

- Used to detect inappropriate queries in a workload
- Automated for Tactical Workloads
- Available for Non-Tactical Workloads (TASM ONLY)

Take Action **NOW**

Unqualified Criteria

☐ Must exist for a period of time

- I/O Count & Physical Bytes
- Sum & Node CPU Time
- Spool Usage Bytes
- Maximum Spool Rows
- Blocked & Elapsed Time
- Number of Amps

**WAIT** Before Action

Qualified Criteria

☐ Must exist for a period of time

☐ Length of time (measured in CPU seconds) must exist before an action is triggered

- I/O Skew Difference & Percent
- CPU Skew Difference & Percent
- CPU Disk Ratio



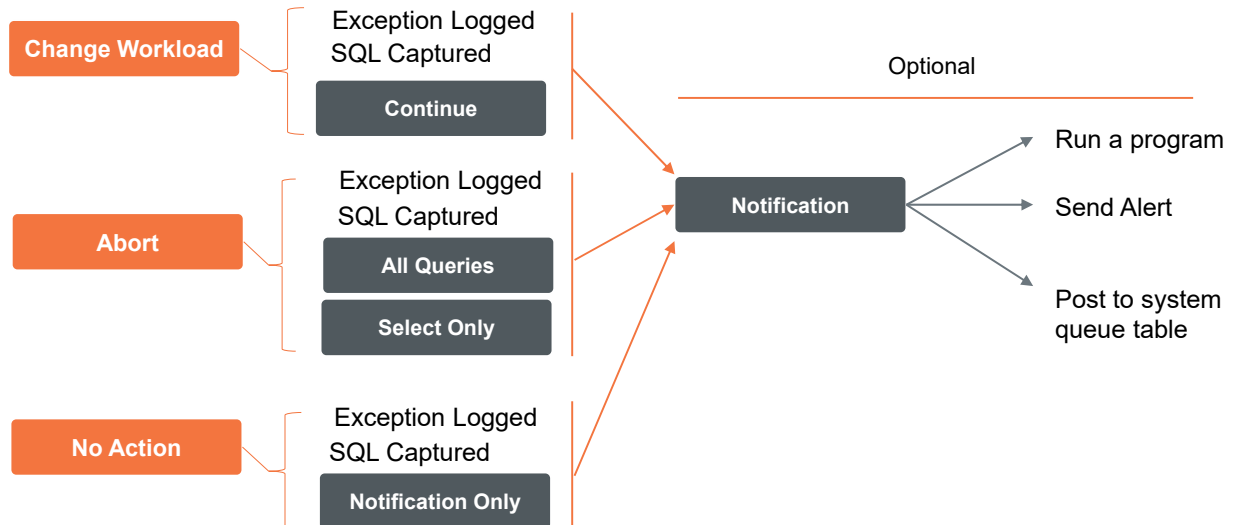
Workload Management has additional functionality to monitor queries during execution for adherence to specified criteria. Prior to Workload Management only CPU accumulations was able to be monitored.

It allows TASM to recognize atypical query processing conditions not intended for that workload so the priority scheduler can perform actions. The qualification criteria prevents false triggers.

The atypical query processing exception is defined via exceptions criteria such as max spool rows, I/O count, spool size, block time, response time, number of amps, cpu time, tactical cpu usage threshold (per node), tactical I/O physical bytes (per node), and/or I/O physical bytes.

## Exception Actions

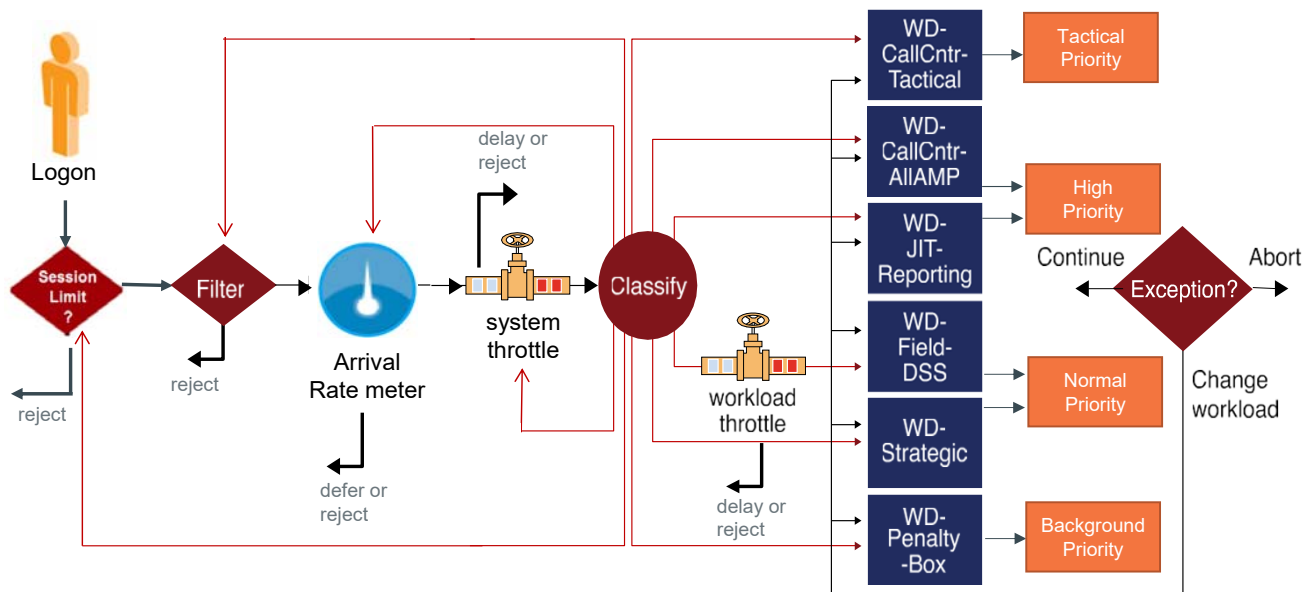
An Exception Action specifies **what to do** when an Exception condition is detected



Workload Management now allows a variety of actions to be taken when an exception condition is detected. Prior to Workload Management only demotion to another allocation group was possible.

The actions that can be taken by the priority scheduler include: abort or stop the query, abort select only queries, change the workload definition to a different work load, send a notification only as an alert or run a program or post an entry in the Qtable for processing by other applications.

## Workload Management Levels

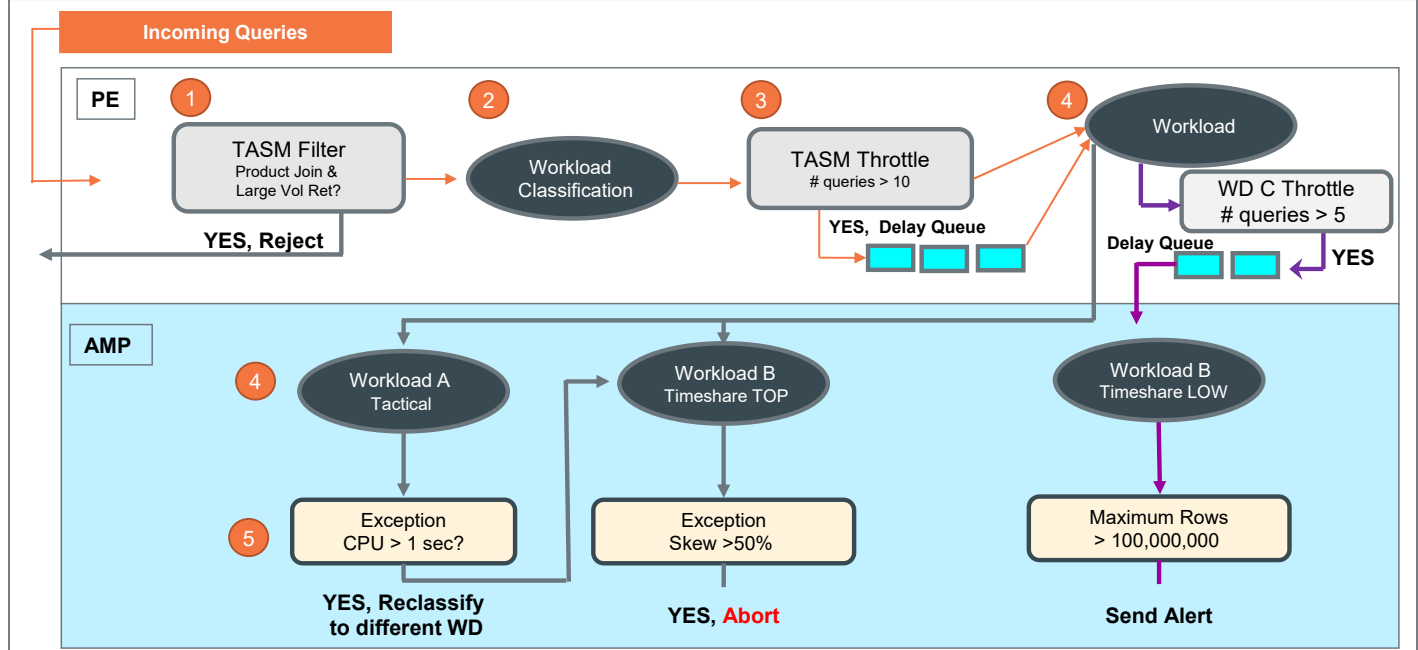


There are seven different methods of management offered, as illustrated below:

- Methods regulated prior to the query beginning execution
- Session Limits can reject Logons
- Filters can reject requests from ever executing
- Arrival Rate Meter can defer requests by managing the number of requests admitted into the system per specified time period
- System Throttles can pace requests by managing concurrency levels at the system level.
- Classification determines which workload's regulation rules a request is subject to
- Workload-level Throttles can pace the requests within a particular workload by managing that workload's concurrency level
- Methods regulated during query execution
- Priority Management regulates the amount of CPU and I/O resources of individual requests as defined by its workload rules
- Exception Management can detect unexpected situations and automatically act such as to change the workload the request is subject to or to send a notification



## Example: Query Management Architecture



The TDWM rules you create are stored in tables in the Advanced SQL Engine. Unless otherwise specified, every logon and every query in every Advanced SQL Engine session is checked against the enabled TDWM rules. That includes SQL queries from any supported Advanced SQL Engine interface, such as BTEQ, CLIV2, ODBC, and JDBC.

TDWM rules are loaded into the Dispatcher components of the Advanced SQL Engine. When a Vantage client application issues a request to the Advanced SQL Engine, the request is examined and checked by TDWM functions in the Dispatcher before being forwarded to the AMPs to execute the request against the user database.

Query Management analyzes the incoming requests and compares the requests against the active rules to see if the requests should be accepted, rejected, or delayed.

Queries that do not pass Filter Rules are rejected

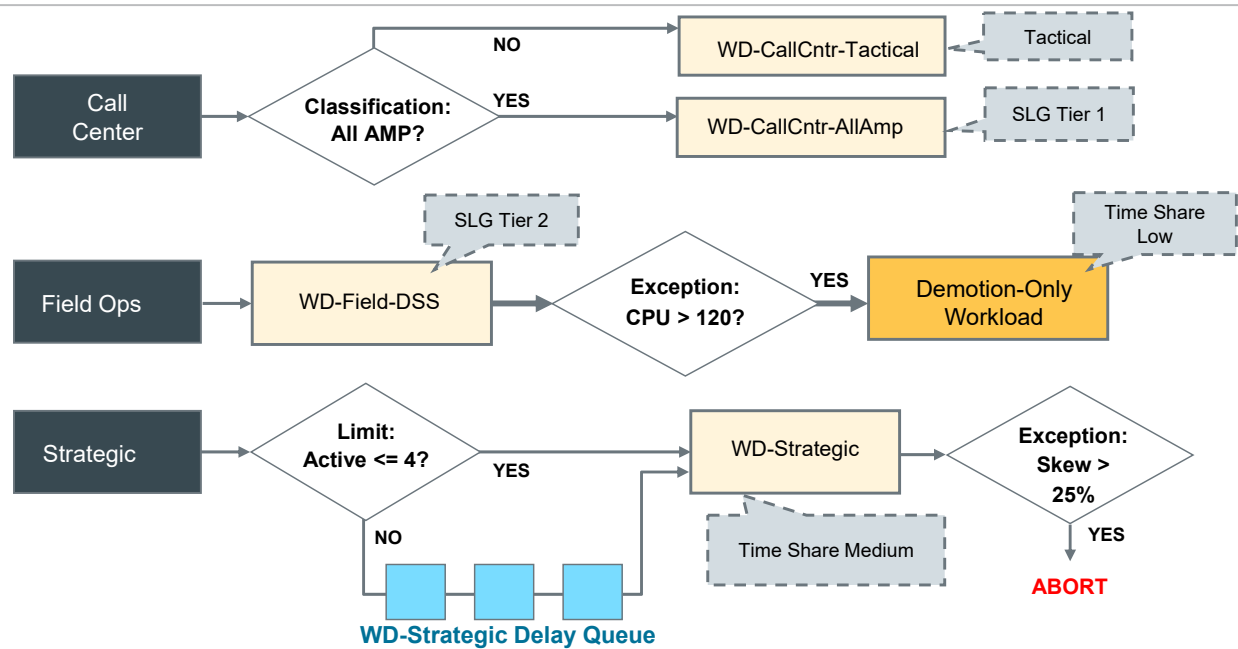
Queries that pass Filter rules are classified into a Workload Definition.

Queries that do not pass Throttle Rules can be delayed or rejected. Additional throttles can also be applied at the Workload Definition level.

As queries execute within their assigned workload, they will be monitored against any exception rules.








Violations of exception rules can invoke several actions from changing workloads, abort the query, send alert or run a program.

## Example: Workloads and Rules



The facing page illustrates an example of creating five workload definitions to handle a mix of queries.

## Workload Management Offering Comparison

Key Features		TIWM	TASM
	<b>Workload Classification</b>	Source, Target, Query Characteristics, QueryBand, Utility	Source, Target, Query Characteristics, QueryBand, Utility
	<b>Virtual Partitions</b>	One Partition	<b>Multiple Virtual Partitions</b>
	<b>Prioritization</b>	Tactical and Timeshare	Tactical, <b>SLG Tiers</b> , and Timeshare
	<b>Resource Management</b>	CPU and I/O	CPU and I/O
	<b>Filters, Meters &amp; Throttles</b>	Filters, Workload and System Throttles; Flex Throttles	Filters, <b>Arrival Rate Meters</b> , Workload and System Throttles; Flex Throttles
	<b>State Matrix</b>	Planned Environments	by Planned Environment and by <b>Health Conditions</b>
	<b>Exceptions</b>	Tactical and Timeshare Decay	Tactical, Timeshare Decay, and <b>Workload Exceptions</b>

Teradata offers Workload Management bundled with all the platforms and it is called Teradata Integrated Workload Management. Teradata also offers advanced Workload Management on some platforms and it is called TASM.

With release 14.0/SLES11 all Teradata Platforms (that support Teradata 14.0 and beyond on SLES 11) will include “Teradata Integrated Workload Management” and TASM will be available on some platforms. See the Workload Management Support Matrix in the TASM 14.0 OCI for details.

This chart highlights the differences between the offerings for release 14.0 for SLES11. The key new features for SLES11 are highlighted in bold:

- All platforms now get full workload classification
- All platforms default to one virtual partition and TASM offers up to 10 virtual partitions
- Prioritization methods have been enhanced with all platforms utilizing Tactical and Timeshare methods while TASM adding SLG Tiers method
- All platforms utilize now also utilize workload throttles in addition to system throttles and filters
- TASM offers sophisticated operating period and health conditions state matrix
- Finally, all platforms utilize Tactical exceptions and Timeshare Decay and TASM adds Workload Exceptions

## Current Topic – Viewpoint Portlets

- Workload Management Overview
- Workload Management Features
- **Viewpoint Portlets**
  - **Workload Management - Administration**
  - **Workload Management – Monitoring and Reporting**



# Workload Management – Administration

The screenshot displays the 'Workload Designer' interface for 'Team1'. The top navigation bar includes tabs for 'General', 'States', 'Sessions', 'Filters', 'Meters', 'Throttles', 'Workloads', and 'Exceptions'. The 'General' tab is active, showing sub-tabs for 'GENERAL', 'BYPASS', 'LIMITS/RESERVES', and 'OTHER'. The 'GENERAL' sub-tab is selected, showing a form with a 'Name: \*' field containing 'FirstConfig' and a 'Description:' field. The 'FirstConfig' label is highlighted in an orange box.

Workload Management Administration

Administration of Workload Management is administered using the Viewpoint Porlet Workload Designer

## Workload Designer – Workloads

Workload Designer TDT5B FactoryRuleSet

General States Sessions Filters Throttles **Workloads** Exceptions

Workloads SLG Summary Evaluation Order Virtual Partitions Partition Resources Workload Distribution Console Utilities

Use workloads to group queries that have common characteristics or common requirements.

Workloads + Virtual Partition: (All) State: Base

ENABL...	NAME	METHOD	WORKLOAD THROTTLE
<input checked="" type="checkbox"/>	H-WD	Timeshare	-
<input checked="" type="checkbox"/>	L-WD	Timeshare	-
<input checked="" type="checkbox"/>	M-WD	Timeshare	-
<input checked="" type="checkbox"/>	T-WD	Timeshare	-
<input checked="" type="checkbox"/>	WD-Default	Timeshare	-

WD-Default is a system workload used for any queries that do not classify to any previously defined workloads.  
It cannot be disabled, modified or deleted.

Save Reset

Viewpoint's Workload Designer can be used to create and modify workload definitions.

One workload that is always present is named WD\_Default. The WD-Default workload definition is the default workload. It is automatically created and is used as a “No-Home WD”. Queries that do not match the characteristics of any other workload definition will run in this WD.

Note: The WD-Default definition cannot be deleted or edited.

## Workload Designer Example

Viewpoint's Workload Designer example.

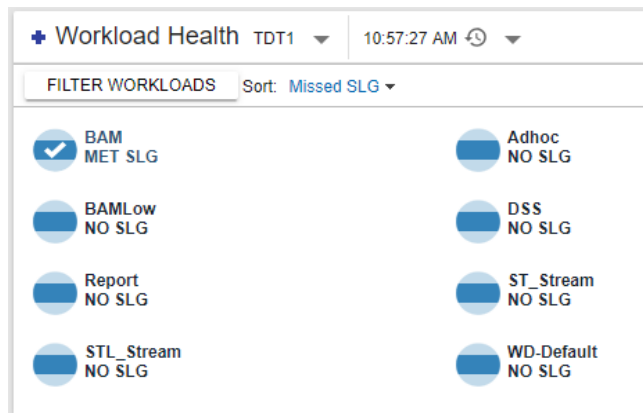
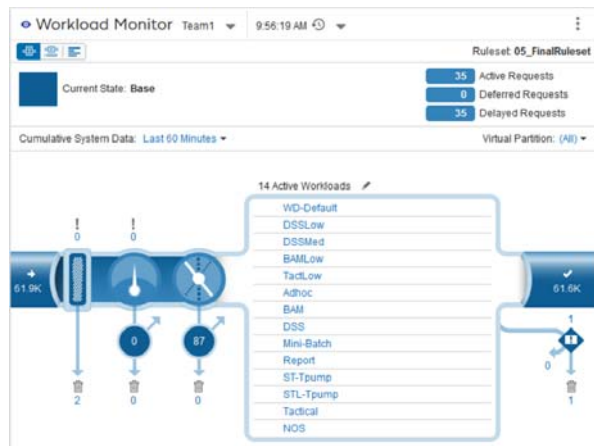
The screenshot shows the Viewpoint Workload Designer interface. The top navigation bar includes tabs for General, States, Sessions, Filters, Throttles (selected), Workloads, and Exceptions. The left sidebar lists various throttle categories: System Throttles, Virtual Partition Throttles, Workload Group Throttles, and Workloads with Throttles. The main content area displays the 'Throttles' tab, which includes sub-tabs for Throttle Limits by State, Resource Limits, and Resource Limits by State. A 'New Throttle' dialog box is open, showing the 'General' tab. The dialog includes fields for Name (set to 'Limit\_Queries') and Description. The 'Rule Type' section has three radio buttons: 'Collective: one queue for all matched queries', 'Individual: one queue for each DB object' (selected), and 'Member: one queue for each user'. There is also a checkbox for 'Disable manual Release or Abort'.

This example illustrates the starting point of creating a throttle.

Throttles can be used to control the number of queries that can execute simultaneously and can be defined at the [System](#), [Virtual Partition](#), or [Workload](#) level.

Viewpoint's Workload Designer is required to manage sessions, filters, throttles, workload definitions, and exceptions.

# Workload Management – Monitoring and Reporting



Workloads can be monitored on a real-time basis or reported at a summary and historical level.

Workload Management provides for monitoring and reporting by workloads.



## Summary

In this module we covered how to:

- Define Workload Management and explain its benefits
- Discuss the concepts and features of Workload Management
- Explain the purpose for the Workload Management portlets in Viewpoint



## Lab Overview

teradata.

Demo 1: Walkthrough of a Ruleset

Demo 2: Walkthrough of a Filter rule

Lab 1: Impact of Errored and Aborted Queries by User / Workload

Lab 2: CPU Consumption by Workloads

Lab 3: Concurrency Queries by User / Workload

Lab 4: Review Exception Criteria using Workload exception logs

Lab 5: Execute a query that breaks the Filter rule criteria and observe the output

Check your understanding of the concepts discussed in this module by completing the review questions as directed by your instructor.



## Module 4: AutoStats

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

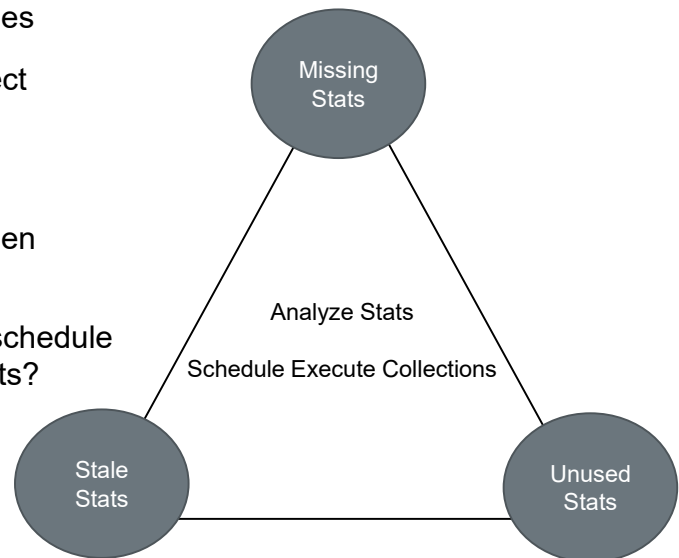
- Explain why Automated Statistics
- Explain what is Automated Statistics (AutoStats)
- Define Features and Limitations
- Describe Concepts and Approaches



## Why Automated Statistics

Managing Statistics has the following challenges

- Which columns and indexes should we collect statistics on?
- How often do we need to refresh them?
- How do we know if the statistics we have been collecting are still being used?
- Are there any tools that can help organize, schedule recurring collections, and monitor their results?



## What is Automated Statistics

Automated Statistics Management (AutoStats) is

- A new feature that automates and provides intelligence to database administrative tasks related to Optimizer Statistics Collections
- A Viewpoint portlet named **Stats Manager** with a user interface
- The portlet calls DB-supplied stored procedures and related metadata is stored in the **TDStats** database

### Use-Cases

- A new system or application where little or no query tuning has taken place
- DBA's and Application programmers may have limited knowledge of statistics collection
- Ease of administrative tasks in existing systems and/or assist in tuning activities

Although AutoStats DBS features can be used apart from the portlet, it is assumed that the majority of Teradata customers will use Stats Manager.

## Automated Statistics: Features and Limitations

Automate and provide intelligence to DBA tasks related to Optimizer Statistics Collections:

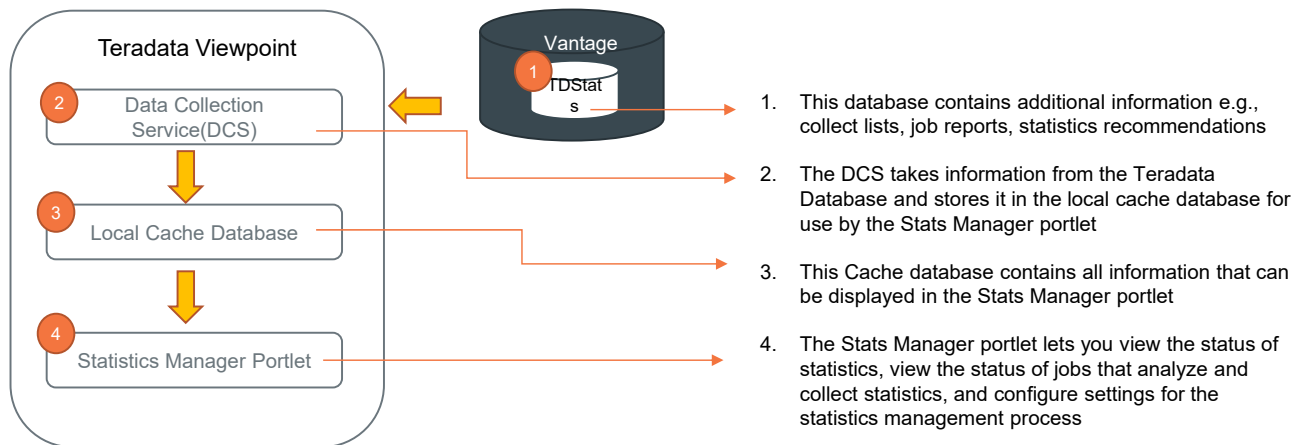
- ✓ View all statistics on a system
- ✓ Identify and collect **missing** (new) statistics needed for accurate query optimization
- ✓ Detect **stale** statistics and promptly refresh them
- ✓ Identify and remove **unused** statistics from ongoing maintenance
- ✓ Prioritize the list of pending collections such that important and stale statistics are given precedence
- ✓ User-defined statistics must be manually collected once before they can be automated with Stats Manager
- Repository
  - ✓ A system-supplied database named TDSTATS that stores metadata for all stats collections
  - ✓ Created by DIPSTATS

Automated Statistics is unsuitable for

- ✗ Complex Data types e.g., GeoSpatial
- ✗ Temporary tables
- ✗ Extract – Transform tables

Open API – in the form of SQL external stored procedures (XSPs) that perform important stats management operations.

## AutoStats Flow



### TDStats

This database contains additional information related to statistics management, such as the statistics that are automated, object scope lists for jobs, collect lists, job reports, and statistic recommendations. The TDStats database is updated when a job runs, a job completes, or when an action, such as automating a statistic, occurs. TDStats is initially configured with only minimal permanent space that is sufficient for automating only a very small number of statistics. Users should configure additional space using the SQL MODIFY DATABASE statement or via the Space Usage Viewpoint portlet.

Do not specify a spool size limit for TDStats

### Teradata Viewpoint Data Collection Service (DCS)

The DCS takes information from the Teradata Database and stores it in the local cache database for use by the Stats Manager portlet. The DCS is also used for job scheduling and initiating jobs. If the DCS is not running, scheduled jobs will not run.

### Local Cache Database

This database contains all information that can be displayed in the Stats Manager portlet. The database is updated when a job completes, when an action such as Automate is performed, or when the Stats Manager collector runs. The local cache database makes data available quickly without impacting the Teradata Database.

### Stats Manager portlet

The Stats Manager portlet lets you view the status of statistics, view the status of jobs that analyze and collect statistics, and configure settings for the statistics management process.



## AutoStats Concepts

### Analyze

- An analyze job evaluates statistics usage, identifies stale statistics, and generates recommendations
- Running an analyze job with the Use DBQL option could be resource intensive

### Collect

- A collect job submits COLLECT STATISTICS statements to the Vantage database which are approved for automation
- Schedule, Scope, and Priority can be controlled for individual COLLECT STATISTICS statements

### Automate

- Automate allows one to approve statistics for collection by COLLECT jobs
- Automation can happen on a single object or table, multiple objects or tables entire databases, or entire systems



Open API – in the form of SQL external stored procedures (XSPs) that perform important stats management operations.

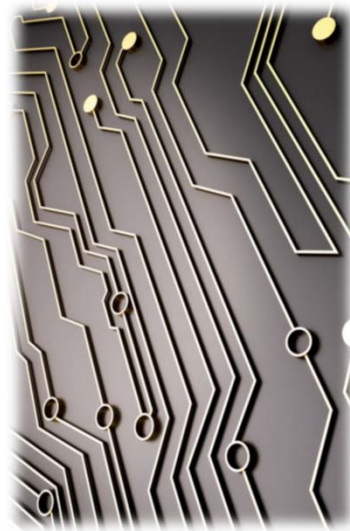
## AutoStats Concepts: DBQL/PDCR

An analyze job requires query plan data generated by STATSUSAGE and XMLPLAN (optional) to

- Evaluate Stats Usage
- Staleness
- Missing Stats to Collect

An analyze job requires enables query logging with the USECOUNT option to

- Evaluate Stats to determine if they should be collected
- Evaluate Stats to determine if they should be deactivated

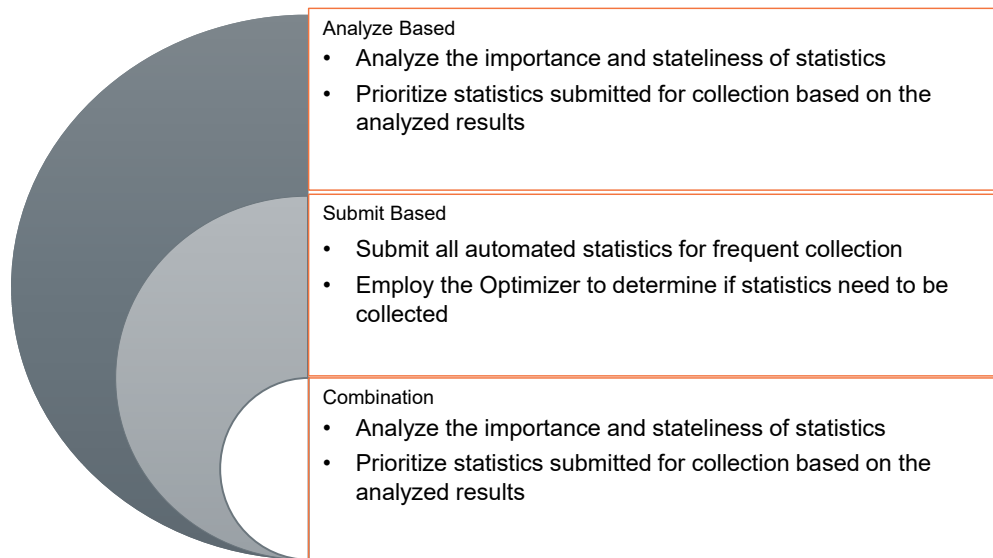


Many systems routinely archive their active query logs into another database. If Analyze Jobs are not run frequently, it may be necessary to specify a Log database other than DBC which is done as part of the job's Evaluation Method Settings.

Analyze logged queries from the Performance Data Collection and Reporting (PDCR) tool to focus the analysis on problematic queries that are in need of performance tuning.

Analyze Jobs can specify the Log database as 'PDCRDATA', in which case the analysis will read the query log data generated by the PDCR tool. This can be very useful in those cases where PDCR was previously used to identify and isolate 'bad' queries.

## AutoStats Approaches



### Submit-Based Approach

The submit-based approach employs the Optimizer to determine if a statistic needs to be collected. You can enable collect thresholds to prevent repeat collection of statistics that were recently collected or have not changed significantly.

Following is an overview of the tasks employed in this approach: Change the Collect age threshold and Data change threshold on the Edit Collect Settings dialog box to skip the collection of statistics that were recently collected or have not changed significantly.

Run the collect jobs frequently. When a job is run, the Optimizer determines which statistics to collect, based on the threshold settings.

Review the Collect Job report for a list of collected and skipped statistics.

Verify that all the automated statistics are submitted for collection. In the Statistics by Database view, examine the Max Submit Age metric for databases or Submit Age metric for objects. If the value is greater than the expected frequency, check the following:

- Statistics are associated with a collect job
- Collect job is scheduled to run with the expected frequency
- End state of the collect job is Complete or Partially Complete. If the job is Partially Complete and not scheduled to resume, the job duration may be too short.

### Analyze-Based Approach

Use the analyze-based approach to evaluate the importance and staleness of statistics, and to prioritize the statistics submitted for collection.

During the collection process, statistics most in need of collection are submitted first. This makes sure

that if a collect job does not complete in the allotted duration, then the higher priority statistics are more likely to have been collected.

If using the analyze-based approach, make sure the collect lists are automatically generated.

You can prioritize individual statistics by setting the user importance.

Following is an overview of the tasks employed in this approach: Enable Teradata Database query logging features (Requires DBQL option with STATSUSAGE. XMLPLAN is optional.) at least one week prior to running analyze jobs.

Create an analyze job. Analyze all statistics on the system and select the Use DBQL option.

Schedule the analyze job to run at least once a week, so data changes are captured. Access the Analyze job report once the analyze job has run. Look for stale statistics. Verify that all stale statistics are scheduled for collection again in the near future.

From the Statistics tab, do the following:

- Review the STALE STATS column to see which statistics are considered stale.
- For collect jobs, make sure STALE STATS is nearing zero. If the value is not nearing zero, increase the duration of the collect job or schedule the collect job to run again in the near future

## AutoStats Recommendations

### WHERE

- Summary Tables
- PARTITION Keyword
- USI / UPI in non-equality predicates
- Most NUPI
- All NUSI
- Non-Indexed columns used in predicates
- Full Statistics always need to be collected on
  - Relevant columns and indexes on small tables (< 100 records per AMP)

### WHEN

- Newly created populated tables
- Recollect Stats after data is added
- After a significant percentage of change to the table or partition
  - ~10% rows change
  - ~7% for high volumes of non-unique values change
- After an upgrade or migration

## Summary

In this module, you learned:

- Explain why Automated Statistics
- Explain what is Automated Statistics (AutoStats)
- Define Features and Limitations
- Describe Concepts and Approaches



## Lab Overview

teradata.

In the following labs, you will:

- Lab 01 - Create an Analyze Job
- Lab 02 – Approve Recommendations based on Analysis
- Lab 03 – Create a Collect Job



## Module 5: Backup and Recovery

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.



## Objectives

After completing this module, you will be able to:

- Explain why BAR (Backup, Archive & Recovery)?
- Discuss Teradata Backup and Restore (BAR)
- Explain what is TARA (arcmain)?
- Explain what is DSA?
- Explain the DSC Architecture
- Define features and limitations
- Explain concepts and approaches



## Why Backup and Recovery?

BAR (Backup, Archive, and Recovery) solutions provide data resilience that can help users overcome data loss caused by user error, system hardware failure, application failure, and natural disasters. A BAR solution can also satisfy long-term storage requirements imposed by corporate and government-imposed regulations.

The solutions should be configured to meet your budgets and the following two goals:

- **Recovery Time Objective (RTO)** - The service-level timeframe within which data must be restored to avoid unacceptable consequences. It is the maximum amount of time users can be without access to their application data following a disaster.
- **Recovery Point Objective (RPO)** - A point in time by which data must be restored to be acceptable to the data warehouse users. It can be defined as the amount of acceptable data loss if a recovery needs to be done.

## BAR Solutions

To archive, restore, and recover data in Teradata Database, you can use either:

- Teradata Data Stream Architecture, which is accessible via the Viewpoint BAR Operations portlet
- Teradata Archive and Recovery utility (ARC) available till 16.20

These programs can co-exist at a customer site; however, only the program that created the archive can read it and restore it.

Teradata Data Stream Architecture cannot restore an archive created by ARC and vice versa.



Protect your strategic business intelligence with a complete Teradata Backup and Restore solution. Users get data resilience that helps overcome data loss while also satisfying long-term storage requirements imposed by corporate and government regulations.

**COMPLETE DATA PROTECTION:** You won't sweat about data loss with a Teradata Backup and Restore solution, and you'll experience rapid recovery with maximum data integrity.

**REDUCE ERRORS:** Reduce operator errors through our automated back-up process.

**MULTI-SITE SYNCHRONIZATION:** Synchronize data between multiple sites and systems with ease.

**INTEGRATED SOLUTION:** Tightly integrated hardware and software solution from industry leading vendors.

## Archive and Recovery Utility (ARC)

### Major tasks or functions of the ARC facility include:

- **Archive** – captures user data on portable storage media
- **Restore** – restores data from portable storage media
- **Copy** – transfer archived data to another system or optionally back to same system
- **Recovery** – recovers changes to data from permanent journal tables

### ARC provides additional data protection for these situations:

- Loss of an AMP's Vdisk for no fallback tables
- Loss of multiple Vdisks (AMPs) in the same cluster
- Accidentally dropped tables, views or macros
- Failed batch processes or miscellaneous user errors
- Disaster recovery
- **ARC is simpler and faster than loading (TPT Load) data back into individual tables**

### Common uses for ARC:

- Archive a database, individual table, or selected partitions of a row partitioned table
- Restore a database, individual table, or selected partitions of a row partitioned table
- Copy an archived database, table, or selected partitions of a row partitioned table to a Teradata Database on a different system

### How Teradata ARC Works

Teradata ARC (available till 16.20) creates files when databases, individual data tables, selected partitions of primary partition index (PPI) tables, or permanent journal tables from the Teradata Database are archived. When those objects are restored to the Teradata Database, the Teradata ARC files are also restored.

Teradata ARC also includes recovery with rollback and roll forward functions for data tables defined with a journal option. Checkpoint these journals with a synchronization point across all AMPs, and delete selected portions of the journals.

<https://docs.teradata.com/r/Teradata-Archive/Recovery-Utility-Reference/March-2019/Introduction/What-is-Teradata-ARC/Teradata-ARC-Specific-Terminology>

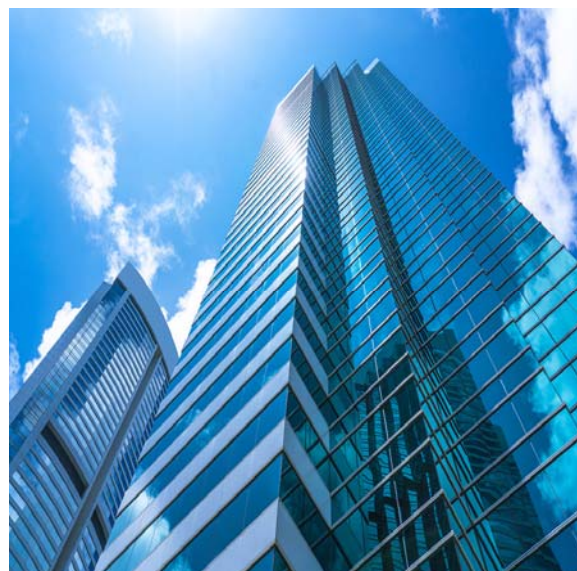
## Why Data Stream Architecture (DSA)?

The **Data Stream Architecture (DSA)** is a complete re-architecture of the current BAR solutions.

- **Improve Performance**
  - Actual throughput will vary and be dependent on BAR infrastructure
  - Remove software bottleneck and move to BAR HW infrastructure
- **More consistent data alignment per stream**
  - Better deduplication with the disk storage device
- **Improve usability with a more robust front-end**
  - Utilizes web interface GUI
  - Flexible for future enhancements
  - Integrated with the Viewpoint Portlet
- **Improve security**
  - Client server communication over the SSL channel
  - Uses Viewpoint user access control
- **Sets up Teradata BAR Solution for future enhancements**

## Introduction to Data Stream Architecture

- Teradata® Data Stream Architecture (DSA) enables you to back up and restore Teradata system data.
- DSA is optimized for Teradata MPP Architecture
- It integrates with the **Teradata® Viewpoint** portlets: BAR Setup and BAR Operations and which provide user interfaces to Teradata DSA
- This integration uses **Viewpoint** account management features and enhances usability.
- Teradata DSA also provides a command-line utility that you can use to configure, initiate, and monitor backup and restore jobs



Teradata® Data Stream Architecture (DSA) enables you to back up and restore Teradata system data. DSA is optimized for Teradata MPP Architecture.

It integrates with the Teradata® Viewpoint portlets: BAR Setup and BAR Operations. The portlets provide user interfaces to Teradata DSA that are similar to other Teradata ecosystem components. This integration uses Viewpoint account management features and enhances usability.

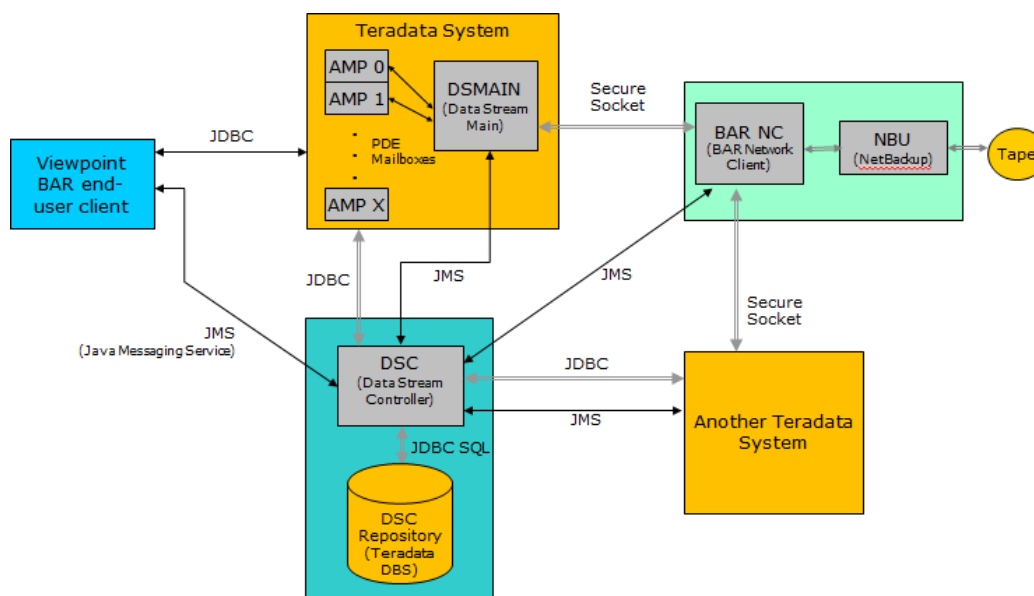
Teradata DSA also provides a command-line utility that you can use to configure, initiate, and monitor backup and restore jobs.

## Components of a DSA Solution

- Teradata Data Stream Architecture (DSA) is assembled for each installation. Installations typically use only one or the other option.
- Teradata DSA provides two different parallel processing options to choose from: Data Stream Extensions (DSE) or Data Stream Utility (DSU).
- ❑ **DSE** supports third-party backup targets such as Veritas NetBackup, IBM Spectrum Protect (DSA 16.20.50 and later).
- ❑ **DSU** supports basic backup targets, including disk file systems, Dell PowerProtect DD Boost storage devices, Amazon S3 (including Snowball), Azure Blob storage, and GCP Storage, but not third-party backup applications.
- ❑ **DSC** is a small Postgres database that controls DSA/BAR operations. It is a required part of all BAR configurations. DSA components DSC or DSU are often installed on Teradata Multi-purpose Servers (TMS). Many components can also be installed in VMware virtual machines.



## DSA Components and Interfaces



### Data Stream Controller (DSC)

The Data Stream Controller (DSC) component controls all DSA BAR operations throughout the BAR enterprise environment. DSC manages and controls the DSC repository. The DSC repository is a Teradata Database instance that contains all the DSA BAR job metadata and relevant activity logs. DSC also discovers configuration changes of other components in the DSA environment and sends information to Viewpoint portlets when the data needs to be displayed. DSC is notified of all requested BAR operations and manages resources to ensure optimal system backup and restore job performance.

Teradata Extension for NetBackup extends the capabilities of NetBackup to include backing up and restoring Teradata Databases. It supports parallel backups and restores that are coordinated across multiple hosts contained in a single Teradata Database. Features include:

- Full integration with the NetBackup master server and Media Manager
- Support for parallel job launch using a single point of control
- Backup and restore of databases and tables
- Client operation monitoring through the NetBackup Client Job Monitor
- Server monitoring is also available through the NetBackup master server
- Job launch, which is supported through these options:
  - Immediate launch through the Teradata TARA GUI
  - Scheduled launch through the NetBackup scheduler
  - Command-line launch

### DSA Network Client - BARNC

The DSA Network Client controls the data path from DSMain to the BAR storage device and verifies authentication from Teradata Database. The DSA Network Client includes the third-party backup application access module that interfaces with the third-party backup application. DSA Network Client is packaged as ClientHandler software. This module opens the connection to the appropriate third-party backup application programming interface (API). DSA Network Client is sometimes referred to as



Backup Archive Restore Network Client (BARNC).

### **DSA Graphical User Interface**

Teradata DSA utilizes web interface technology to allow the user to perform BAR tasks. DSA is integrated with two Teradata Viewpoint portlets, BAR Setup and BAR Operations, providing a GUI front end to Teradata DSA. This is similar to other Teradata ecosystem components. This integration leverages Viewpoint's user role administration features and enhances usability. In addition, Teradata DSA provides a command-line utility interface that can also be used to configure, initiate, and monitor backup and restore jobs.

### **DSA Command-Line Interface**

DSA command-line interface (CLI) provides an alternative to the BAR portlets in performing BAR administrative and operational tasks. The DSA CLI allows backup, restore and analyze job creation, facilitates BAR job runs through a time-based job scheduler, and imports and exports DSA configurations and jobs. DSA CLI requires XML files for BAR operations and setup tasks. A BAR administrator can create DSA XML files containing appropriate commands and metadata to successfully utilize DSA's CLI capabilities. Jobs created via the DSA GUI interface are converted to XML files. The user can also create a job XML file with an editor as long as it follows DSA conventions. Consequently, either of these XML files can then be invoked through the DSA CLI.

[https://docs.teradata.com/viewer/attachment/UMz6TTR~F6D\\_p4VSDc4lnA/kCV2b2Ud4BWEHGK~r8~NYQ](https://docs.teradata.com/viewer/attachment/UMz6TTR~F6D_p4VSDc4lnA/kCV2b2Ud4BWEHGK~r8~NYQ)

## Terminology – DSC Setup

### BAR Setup Portlet

teradata.

- DSC (Data Stream Controller)
- Systems
- Media Servers
- Backup solutions; such as Disk File System (including Spectrum Protect), NetBackup, DD (Data Domain) Boost, Amazon S3, Azure Blob
- Hardware and software groups to use as targets for backup and restore operations
- Automatic backup schedule for the DSC repository
- Beginning with DSA 16.10, nodes are configured through auto-discovery.

DD Boost storage units are tied to media servers during the target group configuration. It is important that the storage units entered for your configuration coincide with the storage units intended for the media server and device configuration mapped as a target.

# BAR Setup

## Discover DSC Servers

The screenshot shows the 'BAR Setup' web interface. The title bar says 'BAR Setup' and 'Configure the hardware and software to use when backing up your Teradata systems.' The left sidebar has a 'DSC Servers' tab and a 'Categories' list with 'General' selected. The main content area is titled 'General System Details' and contains the following fields:

- Broker IP/Host: 10.0.3.240
- Broker Port: 61616
- Broker Connectivity: TCP
- DSC Server Name: 10.0.3.240
- ☒ Enable DSC server
- ☐ Security Management
- BAR Logging:
  - Data Stream Controller: Info
  - BAR Network Client: Info
  - Delete Retired Jobs: After 0 days

At the bottom right, there are 'DISCOVER SERVERS' (with a green checkmark), 'APPLY', and 'RESET' buttons.

- After deploying a Teradata Data Stream Controller (DSC) VM, you must do the following before using the software:
  - Initialize the DSC VM
  - Configure and initialize ClientHandler on Vantage nodes
- SSH to DSC VM using Public IP Address
- Run dsu-init
  - `/opt/teradata/client/16.20/dsa/commandline/dsu-init -v vp private-DNS`  
where vp private-DNS is the private-DNS of the Viewpoint VM to be used with DSC  
Automated scripts initialize DSC on the VM without further input.
- The script sets the repository target group
- DSC is ready to use. The script sets the repository target group
- The repository backup will be stored on the local machine at `/var/opt/teradata/dsu/pathRepo`
- SSH to Vantage PDN node using Public IP Address
- Run the following command from each Vantage node
  - `/opt/teradata/client/16.20/dsa/barnc-init`
  - Please provide tdactivemq running server dns name (This is probably the same as DSC running server): DSC Server Private IP
  - Please provide BarNC web service host name (This is dns name for first tpa node): Private IPv4 DNS of PDN node

# BAR Setup

## Configure Nodes Under DSC Server



17

BAR Setup

Configure the hardware and software to use when backing up your Teradata systems.

DSC Servers

+

Categories

Systems

+

Setup

Nodes

Node Details

10.0.3.240

General

Systems an...

Media Servers

Fabrics

Backup Solut...

Target Groups

Repository B...

Alerts

TDSystem

System Details

Nodes

IR Support

TDSystem00...

Node Name

TDSystem001-01

IP Address:

10.0.3.123

IP Address:

10.0.3.141

IP Address:

127.0.0.1

Stream Limit For This Node

For each node:

480

For each job on a node:

24

# BAR Setup

## Configure Media Servers

7

BAR Setup

Configure the hardware and software to use when backing up your Teradata systems.

?

DSC Servers +

Categories

Media Servers +

Media Server Details

10.0.3.151

General

10.0.3.151

Media Server Name

10.0.3.151

Systems and...

SMP001-01...

BAR NC Port \*

15401

Media Servers

SMP001-02...

Fabrics

Backup Solut...

Target Groups

Repository B...

Alerts

IP Address \*

10.0.3.151

Network Mask \*

255.255.255.0

+

APPLY

RESET

# BAR Setup

## Configure Fabrics

7

BAR Setup

Configure the hardware and software to use when backing up your Teradata systems.

DSC Servers +

Categories

Fabrics +

Fabric Details

10.0.3.240

General

Systems and...

Media Servers

Fabrics

Backup Solut...

Target Groups

Repository B...

Alerts

dscfabric

Fabric Name: dscfabric

☒ Enable fabric

System Name: TDSYSTEM

Nodes +

\* Node: tdsystem001-01

\* Media server: smp001-01\_mec

\* IP Address: 10.0.3.123

APPLY

RESET

# BAR Setup

## Configure Accounts

The screenshot shows the 'BAR Setup' window with the 'Accounts' tab selected. The 'AWS S3 Storage Details' section is active, displaying the following configuration:

- Account Name: dsc-bsaccount
- Access Type: ☒ Key authentication, ☐ IAM Role
- Account Id: \* WZX3P4XN2GF3W6EI
- Account Key: \* [Redacted]
- Regions: +
- \* Region: us-east-1
- \* Bucket: dscs3bucket
- \* Prefix: folder1/
- \* Prefix: folder2/
- \* Storage Units: 24
- \* Storage Units: 24

Buttons at the bottom: APPLY, RESET

As the throughput in AWS is based folder specific, we will create multiple folders for performance benefits. It's independent throughput for individual folder.

Ex. Assume there's a limit of reading only 10 objects/second from a folder, multiple folders will allow  $n \times x$  times of performance benefits.

- On BAR Setup portlet → Navigate to Backup Solutions → AWS S3
- Accounts +
- Account Name: Give any name
- Access Type: Key Authentication
- Account ID: AWS Access Key ID (Received in email for AWS account credentials)
- Account Key: Secret Access Key (Received in email for AWS account credentials)
- Enter AWS Region: (example format: eu-central-1)
- Bucket: Enter the Bucket Name
- Add 2 prefixes for 2 folders created under S3 bucket with storage units 24 each
  - /dbbackup1
  - /dbbackup2

# BAR Setup

## Configure Target Groups

7

BAR Setup

Configure the hardware and software to use when backing up your Teradata systems.

DSC Servers +

Categories

Target Groups

Remote Gro... +

10.0.3.240

General

Systems and...

Media Servers

Fabrics

Backup Solut...

Target Groups

Repository B...

Alerts

Remote Gro...

Restore Gro...

Replication ...

default...

📄

📁

Remote Group Details

Target Group Name \* dscdemo-tg ☒ Enable Target Group

Solution Type AWS S3

Account Name 1nodeAWSaccount

Region us-east-1

Targets +

\* BAR Media Server: SMP001-01\_media

\* Bucket: 1node demo

\* Prefix: folder1/ \* Storage Units: 24 --

\* Prefix: folder2/ \* Storage Units: 24 --

APPLY RESET



## BAR Operations – Check Backup Job Status

**BAR Operations** 10.0.3.240

**Backup Job Status** [View History](#) [View Save Sets](#) [Delete Save Sets](#)

Job Name: my1stDSCjob  
 Source System: TDSysSystem  
 Target Group: dscdemo-tg  
 Lock Status: NO\_LOCK

Complete - 100% Elapsed Time: 00:02:02 [View Phase Log](#)

**LOG**

END	OBJECT NAME	OBJECT TYPE	PHASE	STATUS	PARENT
5/2/23 4:18:45 PM	Accounts	TABLE	DATA	COMPLETE	TRAINED
5/2/23 4:18:45 PM	Accounts	TABLE	DATA	COMPLETE	TRAINED
5/2/23 4:18:45 PM	Accounts	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	Accounts_VD	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	admissions_test	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	admissions_train	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	ALLCHANNEL_EVE	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	attrib	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	attrib1	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	attrib3	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	attrib7	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	attrib9	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	ATTRIBUTION_CON	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	ATTRIBUTION_MOD	TABLE	DATA	COMPLETE	TRNG_1
5/2/23 4:18:45 PM	ATTRIBUTION_MOD	TABLE	DATA	COMPLETE	TRNG_1

Page 1 of 14 (1332 rows total)

## Verify AWS S3 Bucket for Backup Confirmation

them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	my1stDSCjob_10_1_file_dict_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file11_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file12_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file13_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file14_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file15_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file16_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	my1stDSCjob_10_1_file17_data_1683058634824/	Folder	-	-	-
<input type="checkbox"/>	mv1stDSCjob_10_1_file18_data_1683058634824/	Folder	-	-	-

## Backup Options

The incremental backup feature allows three types of backups: full, delta, and cumulative.

- **Full backup:**
  - The type of backup which is a must for the first time, also acts as a base backup for the other two types of backup
- **Delta backup:**
  - Data that changed since last backup
  - Shortest time, least storage
  - Increases restore time
- **Cumulative backup:**
  - Data that has changed since last full backup
  - Consolidates changes
  - Shorter restore time

### Example of a Backup Strategy

Day	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Backup Type	F	D	D	C	D	D	D

The incremental backup feature allows three types of backups: full, delta, and cumulative.

### Backup Types

The first backup must always be a Full backup. The full backup is the baseline for all further backups.

**Full:** A full backup archives all data from the specified objects. This backup takes the longest time to complete and uses the most backup storage space. However, a full backup has the shortest restore time, since all data required to restore the objects will be contained within a single backup image.

**Delta:** A delta backup archives only the data that has changed since the last backup operation. This backup will complete in the shortest time and use the least storage space. However, a delta backup increases the time to restore the database, as it may add many backup images that must be processed before a set of objects can be fully restored.

**Cumulative:** A cumulative backup archives the data that has changed since the last full backup. This backup type consolidates changes from multiple delta backups or cumulative backups before a full backup is run. A cumulative backup has a shorter database restore time than a series of delta backups, and it takes less time and space than a full backup.

## Snapshot or Backup for Cloud Data Protection?

- *You need both!*
- **Snapshots are ideal for:**
  - *Fast*, full system backup (only) for data protection
  - Full system copy for Disaster Recovery purposes
- **Backups are ideal for:**
  - Object-level backup (i.e., at the individual table or database level)
  - More frequent backup of databases / tables that change frequently
  - Partial system backup / restore for the movement and restore of individual objects between multiple Teradata systems (classic Test / Dev / Prod environment)
    - *Example* – only a single database / table is being sent to Test / Dev from Prod for sandbox or development use

Snapshot or Backup for Cloud Data Protection?

# Cloud Business Continuity and Recovery Decision Matrix

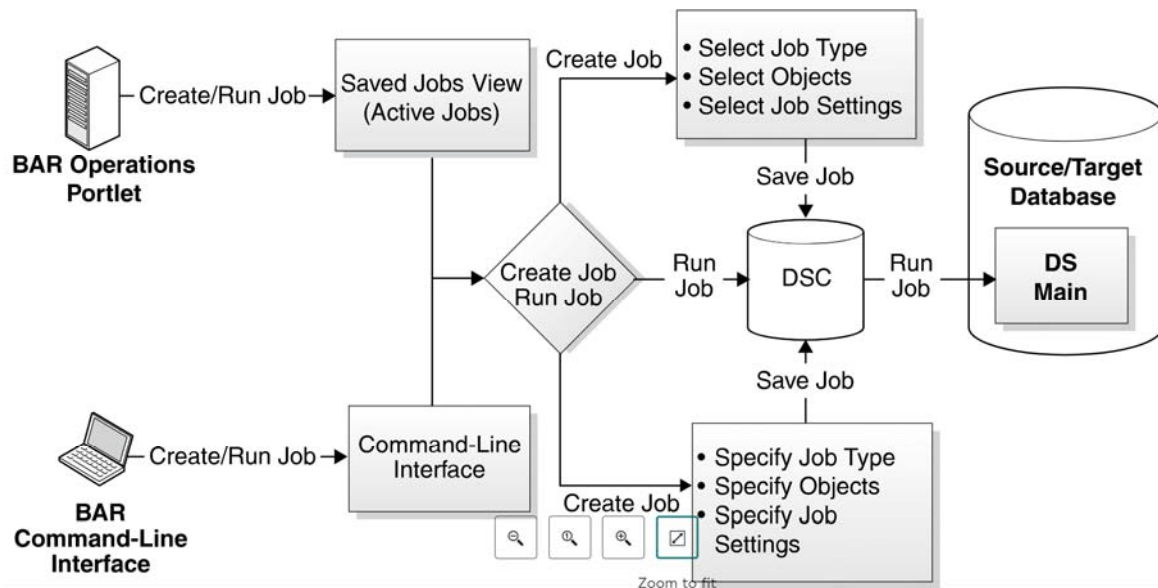
teradata.

BCR Offering	Solution	Technology Used	Data Protection and DR Scope	Standard or Custom Solution	CSP Availability	Target Recovery System	RTO/RPO	Cost (Low, Medium, High)	DR Solution Temperature (Cold, Warm, Hot)	Multi-Cloud / Hybrid Support
BaaS (Standard Backup)	Backup as a service solution using native Teradata technology.	DSA (Data Stream Architecture); DSC (Data Stream Controller)	Data Protection Full Database recovery or Partial recovery (Object Level) of the Database	StandardaaS solution	AWS: Yes Azure: Yes GCP: Yes	Supports Heterogenous target system for both storage and compute.	RTO: Days to Weeks RPO: Depends upon backup schedule	Low	Cold solution.	Not-MultiCloud Not-Hybrid
BaaS (Snapshot Backup)	Backup as a service solution using disk snapshot technology provided by CSPs.	CSP Disk Snapshot	Full Database recovery Only	StandardaaS solution	AWS: Yes Azure: Yes GCP: Yes	Supports heterogenous target for compute VM size only. Number of nodes and storage needs to be homogenous.	RTO: Days to Weeks RPO: Depends upon backup snapshot ? schedule	Low	Cold solution.	Not-MultiCloud Not-Hybrid
DRaaS (Using Snapshot)	Provides an active DR environment.	CSP Native technology (Disk Snapshot)	Disaster Recovery, Full Database only.	StandardaaS solution	AWS: Yes Azure: Planned in Q4 2022 GCP: GA in Q1 2023	Supports heterogenous target for compute VM size only. No. of nodes and storage needs to be homogenous.	RTO: <24 Hours RPO: <24 Hours? Should be even less than 24 hours.	Medium	Warm Solution	Not-MultiCloud Not-Hybrid
Data Mover Solution for DR	Custom replication solution for DR using Data Mover.	Teradata Data Mover (DM)	Disaster Recovery, Full Database or Partial objects.	Custom Solution: Requires a professional service engagement	AWS: Yes Azure: Yes GCP: Yes	Supports Heterogenous target system for both storage and compute.	RTO: Custom typically 4-24 hours RPO: Custom (Typically 4-24 hours)	High	Warm/Hot Solution	MultiCloud Hybrid (Both Ways): Manual failover and fallback.

Note that customer may choose a combination of solutions mentioned above to meet their BCR requirements. For example, they may choose BaaS Standard for regular backup and Data Mover for moving subset of data to a secondary system or may use combination of DRaaS for Disaster Recovery and BaaS Standard to meet their backup needs.

Restore times can vary greatly based on amount of data and source/target system configuration; any AMP or node count mismatch will increase restore times by 4x-6x (smaller system to larger system) to 6x-10x (larger to smaller system) restore, due to need to rehash/redistribute data.

## BAR Job Workflow



The BAR Operations portlet or DSA command-line interface communicates with the DSC when a user creates or runs a backup, restore, or analyze job. The DSC controls the job flow by sending the job processing instructions to the appropriate DSA component. The DSC receives job status information from the DSA component and also notifies the Teradata Database and other client applications of any action taken on a specific job. The job definition is stored in the DSC repository.

<https://docs.teradata.com/viewer/attachment/RJwrXYTj6GIJLW9AInWpMw/u3OsObFEs2j8mKCjMulAn>  
g

## Running a Backup Job



### What We Need?

- Web Browser
- Viewpoint server up and running
- BAR Setup portlet configuration complete
- AWS S3 bucket and permission to write to the bucket



### Steps

- Connect to Viewpoint
- Add "BAR Operations" portlet
- Create a new backup job
- Run backup job



### Outcome


- Data from Teradata instance saved as files in S3 bucket

Although AutoStats DBS features can be used apart from the portlet, it is assumed that the majority of Teradata customers will use Stats Manager.



# BAR Operations Portlet

## Create NEW JOB

teradata.

 **teradata.** VIEWPOINT Home


REWIND ON ☐ OFF

 **BAR Operations** 10.0.3.240 

SAVED JOBS JOB HISTORY

NEW JOB Active Jobs ▾

0	0	0	0	0	0	0	0	▾
All	Complete	Running	Failed	Queued	Aborted	Aborting	Warning	
JOB NAME ▴	TYPE	STATUS	START	END	▾			
<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>				

 No Data Available



## Summary

In this module, you learned:

- Explain why BAR (Backup, Archive & Recovery)?
- Discuss Teradata Backup and Restore (BAR)
- Explain what is TARA (arcmain)?
- Explain what is DSA?
- Explain the DSC Architecture
- Define features and limitations
- Explain concepts and approaches



## Demo/Lab Overview

In the following demo/labs, you will:

- Demo 01 - Create a Backup Job
- Demo 02 – Run a Backup Job
- Demo 03 – View Job Reports



## Module 6: MAPS

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

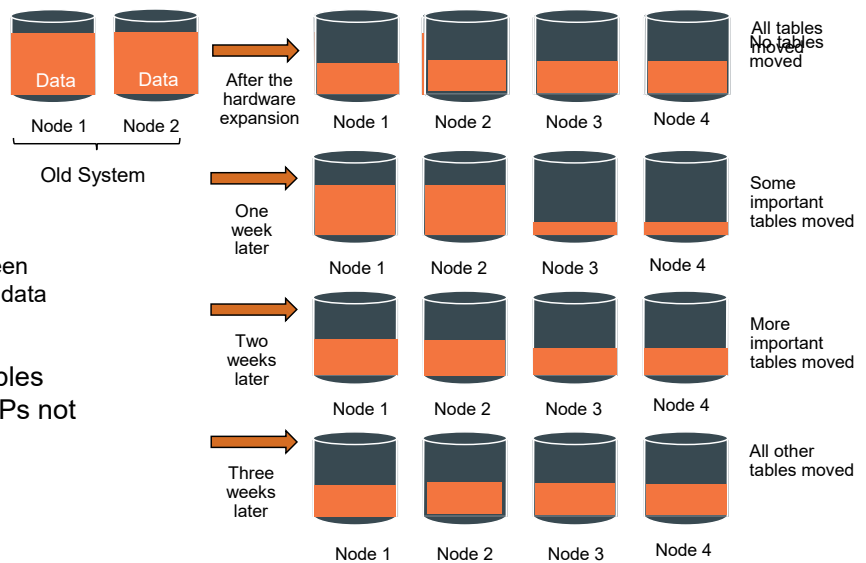
- Why TDMaps and its benefits
- Explain the two types of Maps (Contiguous and Sparse)



## Why MAPS

Whenever the system was expanded there was a need to

- Move every table at the same time
  - The system would not be available during that time
  - Unavailability could range between hours to days depending on the data volume
- Observe performance for small tables since could get degraded with AMPs not having any records.



After you add AMPs or nodes to a system, if the Teradata Database MAPS architecture is enabled, the Reconfiguration utility (Reconfig) creates a new map that reflects the expanded system. A DBA can then choose to move table rows to the new AMPs using either Reconfig or maps. There are several advantages to using maps instead of Reconfig:

- Reconfig requires more planned downtime.
- Reconfig requires you to move all tables in the system immediately to evenly distribute rows between AMPs. However, if you use MAPS, you can move tables into the new configuration gradually while the system is online. Some tables can be in the old map and some tables can be in the new, expanded map.

MAPS feature allows for skipping the table redistribution portion of the reconfiguration process, leaving tables on the old AMPs. Decoupling the movement of data from the hardware expansion shortens the planned downtime to just the time required to connect the new nodes and update a few system files.

Tables can then be redistributed across the new nodes and their AMPs at the convenience of the administrator, and while the system is online. By doing so, data is accessible more quickly and there is less disruption to the business following a hardware expansion.

Prior to MAPS, the old map that pointed rows to AMPs where they were located was discarded after the rows had all been successfully relocated. With MAPS, the old map is retained to manage access to tables that are still in their original location. A new map is created in readiness for the movement of tables to the new nodes, and is immediately available for spools. Thus, a single Teradata system can use multiple maps at the same time.

## MAPS Benefits

- Expansion window outage is significantly reduced
  - You do not have to move every table
  - You do not have to move every table at the same time
- Greater availability of data for end-user processing
  - Ability to read data while tables are being moved to the new nodes
- Spreading table rows across the new nodes can be done at the convenience of the administrator
- Improves performance for **small tables**
  - Without MAPS, access to small tables can be inefficient because some AMPs do not have rows
  - All-AMP access still has to access the AMPs that don't have any rows

The Teradata MAPS architecture (or simply MAPS) allows tables to be assigned a subset of AMPs in the system. As part of MAPS, the Teradata Database has been enhanced to support multiple different “maps” at the same time. Each map definition points to a range or set of AMPs that support the data for tables assigned to that map. MAPS provide two key benefits:

- Increased availability during system expansion
- Greater efficiency when accessing rows from small tables.

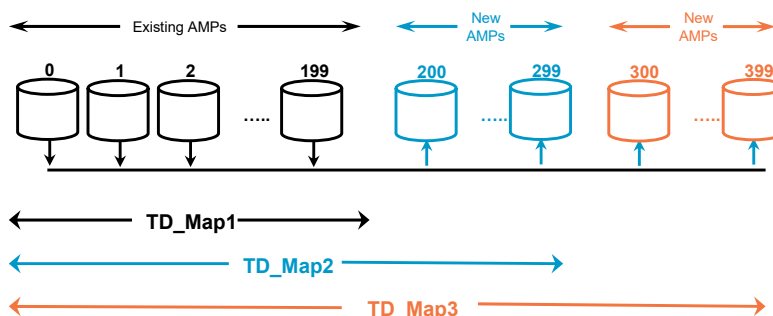
Every table uses a map that specifies which AMPs store the rows of the table. For tables with fallback, the primary and fallback copies of the row are stored on different AMPs in the map.

You can use maps to optimize the placement of tables on the AMPs in your system or to redistribute table rows after a system expansion. Before doing this, it is important to consider some general concepts. Teradata uses two types of maps to track which rows of a table belong on which AMP: Contiguous maps include all AMPs within a specified range. By default, every system has one contiguous map that includes every AMP in the system. Teradata Database creates contiguous maps during a system initialization, configuration, or reconfiguration.

Sparse maps are smaller maps that contain only a limited number of AMPs. By default, each system that enables Teradata Database MAPS Architecture has a 1-AMP sparse map and an n-AMP sparse map, where n is the number of nodes in the system.

Secondary index tables use the same map as their base (indexed) table. Tables and join indexes are assigned a map either explicitly or by default.

## MAPS Multiple Reconfigurations



Contiguous maps use every AMP between a range of AMPs. The Config/Reconfig process creates new Hash Maps called Contiguous Maps.

- Original system TD\_Map1 contains AMPs 0 – 199
- After 1st expansion, TD\_Map2 contains AMPs 0 – 299
- After 2nd expansion, TD\_Map3 contains AMPs 0 – 399

A contiguous map allows for a range of AMPs to act in combination as the repository for the rows of a set of tables. When a table is accessed, only the AMPs in the table's contiguous map are involved. Prior to MAPS, the Teradata Database had one contiguous map for primary data with a range of AMPs from zero to N-1, where N is the number of AMPs in the system. With just a single contiguous map, queries that executed on the platform were guaranteed to have all the AMPs in the system available when they run. The parallelism is maximized, stable, and predictable.

The key advantage to having more than one contiguous map is to allow some (or all) tables to stay in the original map after a system expansion. This reduces the downtime associated with a hardware expansion because it postpones the effort of moving rows from every table in the old map to the new map.

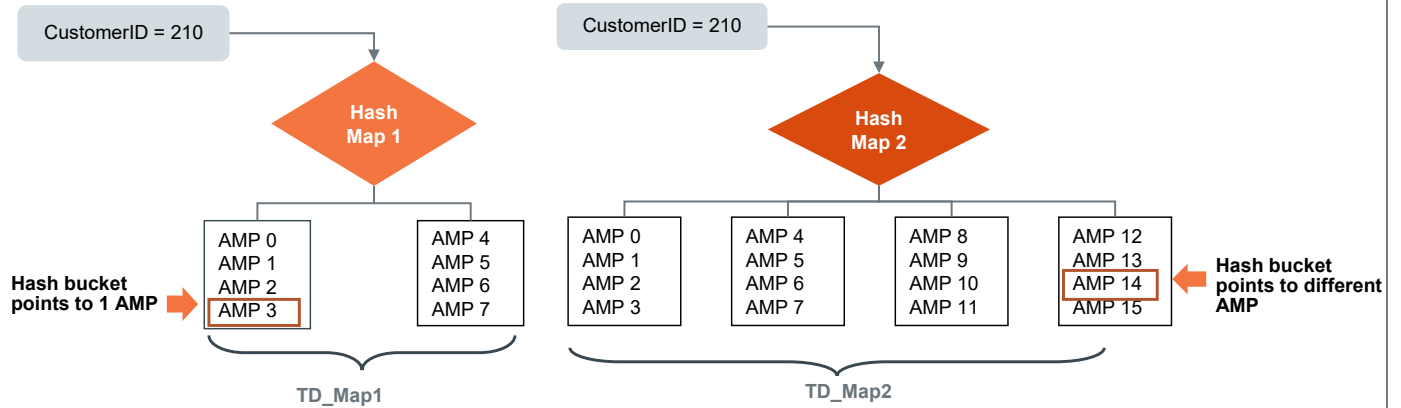
Whether or not all tables are eventually moved to the new map is now up to the individual site. The movement of tables can be scheduled by the administrator, and performed at convenient times and run as background, low-priority work. Having more than one contiguous map for user tables is best viewed as a temporary state of a few weeks, or a month or two, until such time as the movement of all user tables is complete and all rows are contained within the newest, more parallel-rich contiguous map.

If a site undergoes multiple hardware expansions and the administrator has fallen behind in moving tables to the most current contiguous map, it is possible to end up having user tables located in several, even many different contiguous maps. This situation is best avoided, as it will make query tuning more difficult and could add confusion when analyzing system metrics. Optimal performance is experienced when all units of parallelism are fully available to each query that executes.

# MAPS Data Distribution

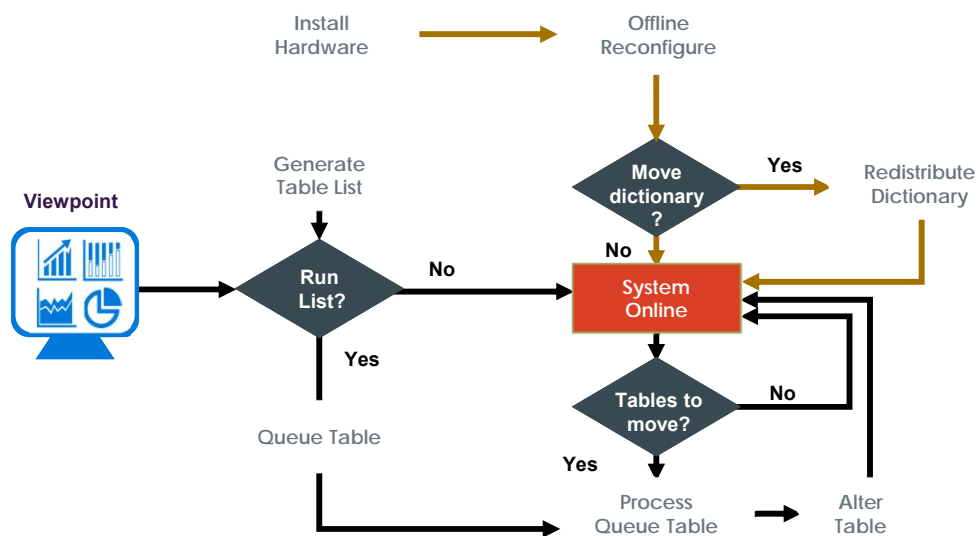
When a table is moved to a different map

- The hashing algorithm used in the new map will usually assign the row to a different AMP





## System Expansion and Table Migration



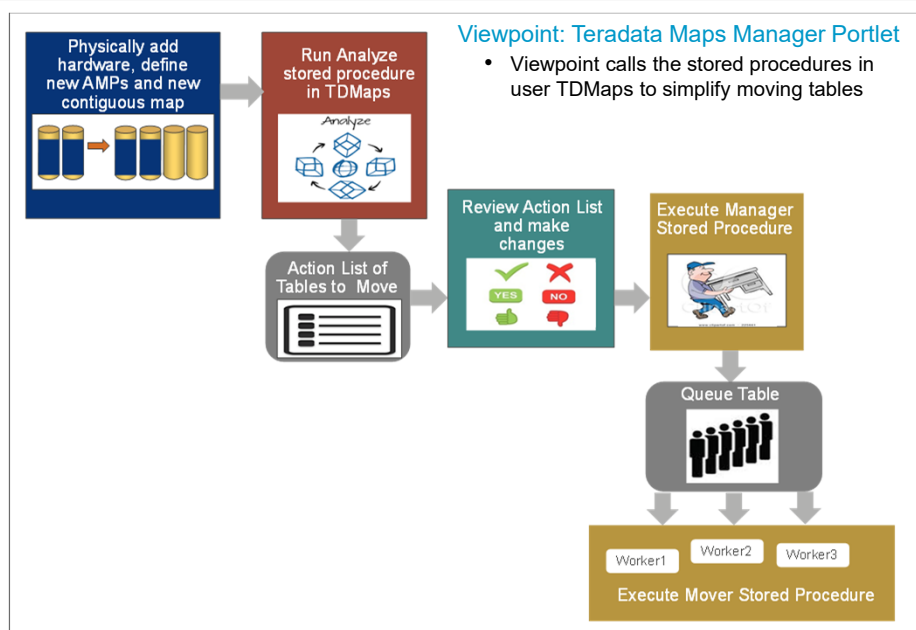
Do you move the DD tables? There is no compelling need to expand this map and spread the data dictionary tables across a larger number of AMPs at the time you expand hardware. You might, for example, be thinking about getting more parallelism from queries that access the DBC. If you do wish to modify TD\_DataDictionaryMap to include all AMPs, it is recommended that you specify that at the start of the Reconfig job, when prompted. But even if you don't decide until after the Reconfig that you want to expand the data dictionary map, you can re-run the Reconfig utility for the sole purpose of expanding TD\_DataDictionaryMap and redistributing the data dictionary tables and objects.

Note: You can only expand the data dictionary map and move the contents of the data dictionary during a Reconfig. DBC table can never be moved manually under the control of the administrator.

If you choose to expand TD\_DataDictionaryMap and move the DBC tables during a Reconfig, whether during the system expansion or at a later time, all of the data in the data dictionary must be redistributed successfully to the larger version of the map before the system can come back up. There is only a single dictionary map, which means all DBC tables must be spread across that map's AMPs as soon as the map has been modified. Unlike user tables, DBC tables cannot be moved after the Reconfig by means of an ALTER TABLE, at the convenience of the administrator.

A reason to leaving TD\_DataDictionaryMap as is coming into the Reconfig: the time to redistribute the data in the dictionary will reduce availability of the data. DBC tables are not the only objects in the dictionary map. Database modules, such as the parser and the optimizer, issue express requests to access data in the DBC tables, and those express requests are mostly composed of streamlined single-AMP requests. Additional AMPs would provide no performance benefit for those types of queries.

## Move Tables: Process Flow



Use the Teradata Viewpoint portlet MAPS Manager to determine if tables are using optimal maps and, if necessary, switch them to more suitable maps on a flexible schedule.

There are several advantages of using Viewpoint, and it is recommended that the Analysis job be your starting point for the movement of tables across maps. Before a Mover job can be initiated, an Analysis job must evaluate each table undergoing analysis at the same time. The Analysis job may suggest groupings of tables from among those submitted with the recommendation that they be moved together. This might be the case if tables perform primary index joins. The Analyze job also identifies candidate small tables suitable for sparse maps.

An Analysis job produces an action list that becomes input to the subsequent Mover job. This list is materialized in the database as a table named TDMaps.ActionsTbl. The size of the action list depends on how many tables and/or databases were submitted to the Analysis job.

After the Analysis job completes, the administrator can review the recommendations that have been made and make modifications. A single Mover job is then executed, and typically includes all the tables in the action list. The Mover job calls two different procedures:  
A Manager procedure, which inserts all the table names in the actions table to be included in the move into a Mover queue table.

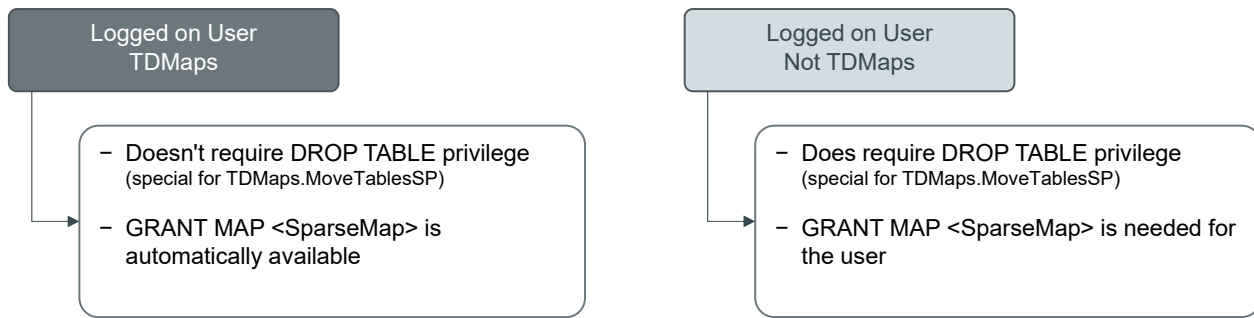
TDMaps.MoveTablesSP procedure, which consumes from the queue table and initiates an ALTER TABLE for each table in the queue.

More than one table can be moved to the new map by calling more than one TDMaps.MoveTablesSP procedure. If using Viewpoint, this is done by defining multiple “workers

## What is TDMaps?

TDMaps is a user created for the MAPS feature

- This contains stored procedures to help the movement of tables
- Reconfig won't be used to move tables, how will a site submit ALTER TABLEs for all of their tables?
  - This can be done using Stored Procedure `TDMaps.MoveTablesSP` which can be utilized



Within the TDMaps system user, there are many SQL stored procedures for automating system expansion and map assignments. The procedures in TDMaps fall into two main categories: Advisor and Mover.

Advisor procedures analyze tables to recommend the best map for each table. The recommendations can optionally be customized and used as input for the Mover.

Mover procedures redistribute the data for a specified list of tables based on new maps. Mover can use multiple worker sessions to achieve the desired level of concurrency. One manager session schedules the execution of groups of related actions and coordinates the worker sessions.

The ALTER TABLE command has been enhanced for use with maps so that when it sees the map name clause, it performs differently from the traditional ALTER TABLE. The `TDMaps.MoveTablesSP` procedure that is also used by the Viewpoint portlet to perform table moves is basically a packaging of the enhanced ALTER TABLE.

Although they are both manual approaches, one difference between calling the procedure and submitting the ALTER TABLE is that the procedure includes a summary-level statistics collection request against that table after it has been moved. When using the ALTER TABLE command outside of the procedure, remember to recollect summary statistics on the table when the ALTER completes. Another difference is the requirement for the DROP TABLE privilege on the table being moved to the new map. If the procedure is being executed by user TDMaps, the DROP TABLE privilege is not needed. If the procedure is being executed by another user, that user needs the DROP TABLE privilege just as if that user issued the ALTER TABLE MAP statement themselves.

## System Defined Maps

- **TD\_GlobalMap**
  - Always defined and includes every AMP
  - Only unhashed tables (e.g., DatabaseSpace) reside in this map
- **TD\_DataDictionaryMap**
  - Always defined, all tables in DBC reside in this map
  - All stored procedures, UDFs, and XSPs reside in this map
- **TD\_Map1**
  - Can be dropped – however, all tables must be removed from a Map, before the Map itself can be dropped
  - All other tables reside in this map upon upgrade to 16.xx or sysinit
- **TD\_Map2...**
  - Added by Config/Reconfig when expanding a system
- **DIPMAPS creates two Sparse maps – a 1 AMP and #AMPs = #Nodes**
  - TD\_1AmpSparseMap\_nNodes, ex. TD\_1AmpSparseMap\_4Nodes
  - TD\_nAmpSparseMap\_nNodes, ex. TD\_4AmpSparseMap\_4Nodes

Note: Every table in the system is assigned to a map.

When the MAPS architecture first becomes available after a software upgrade to Teradata 16.10, several changes will be noticeable, even if none of the options of the MAPS architecture is being currently utilized. The following default maps will have been created for you:

- TD\_GlobalMap – this contiguous map includes all of the AMPs in the system; it represents a global view of the system across all of the installed hardware.
- TD\_DataDictionaryMap – this contiguous map contains DBC tables as well as SQL procedures, external procedures, and UDFs.
- TD\_Map1 – this contiguous map contains all other tables, such as user tables and join indexes.
- TD1AmpSparseMap\_nNodes – a one-AMP sparse map with TD\_Map1 as the parent contiguous map. n is the number of nodes in the system.
- TDnAmpSparseMap\_nNodes – an n-AMP sparse map with TD\_Map1 as the parent contiguous map. n is the number of nodes in the system. Note that this map is only created if there are two or more nodes in the system.

Initially, all three of the above contiguous maps have the same scope that includes all the AMPs.

When no additional maps have been created on the system beyond the ones mentioned above, all newly created tables will go into TD\_Map1. Every time new hardware is added, a new contiguous map TD\_MapK (K is a number chosen by the system) is created and TD\_GlobalMap is changed to cover the complete scope of AMPs available to Teradata Database.

Contiguous maps follow a defined naming convention.

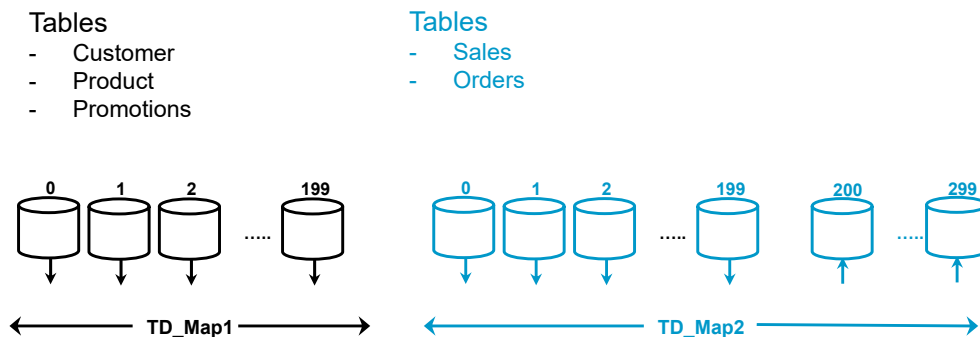
## Example 1: Query Behavior (1 of 2): Performing a Join with Tables in Different Maps

teradata.

Rows from two different tables must be on the same map to perform a join

The optimizer may read a table from one map and write to a spool file in a different map

Optimizer will apply cost algorithms to determine which map to use



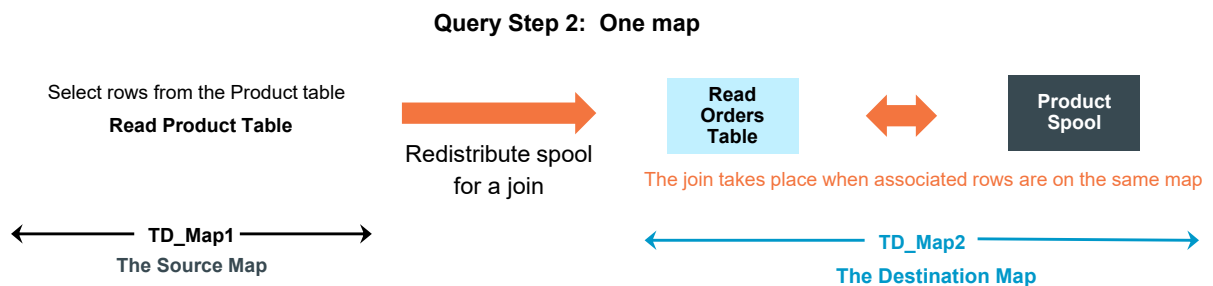
So now you've started moving your tables into TD\_Map2. And there may be a period of time - perhaps several weeks, perhaps several months - where a certain number of queries are going to be joining tables that are in different map: you may have one of the tables in TD\_Map1 and another in TD\_Map2.

The Optimizer is going to have to make sure that rows to be joined are in the same map before it can perform that join. It has to be using the same hash map for the spool and the table being joined to.

So, it's very possible that the Optimizer is going to read a table in 1 map build a spool file, and write that spool file in another map, sort of like the example I showed you in the previous slide. Costing algorithm is going to be used by the Optimizer to determine the most efficient way to get the two tables that are going to be joined to get their rows onto a single map in order to perform the join efficiently.

## Example 1: Query Behavior (2 of 2): Performing a Join with Tables in Different Maps

- Explain text will show a “source map” and a “destination map”
- When tables-to-be joined reside in different maps, the optimizer will consider:
  - Duplicating the small table into the map of the larger table
  - Redistributing one or both tables to one map



So you might see something like this happening - where step one in a query plan is to read the Product table which is in TD\_Map1 create a spool file and redistribute that spool file to all of the AMPs in TD\_Map2 in preparation for a join to the second table, which is the Orders table. So again, the spool or the table that are being joined have to be on the same map in order to make the join happen.

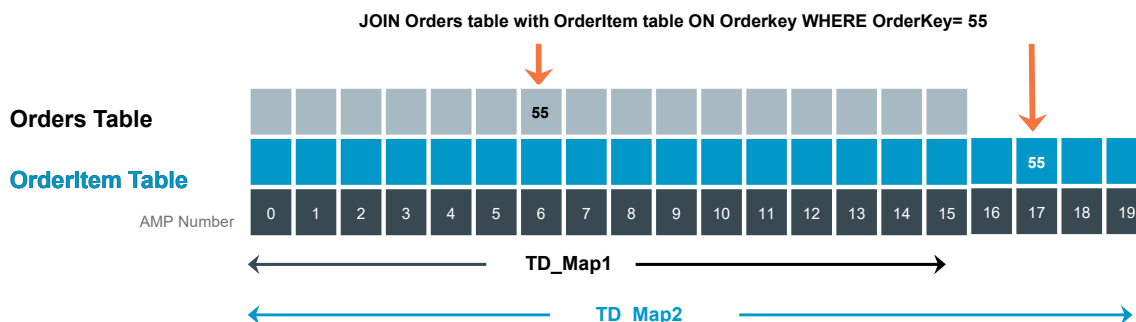
Some of the things the Optimizer will consider are the same kinds of things it considers in the pre-map plans: that is duplicating small tables or redistributing one or both tables. And the map it chooses to redistribute to will be determined by the costing algorithms.

The nice thing though is you can see this in the explain text. It very clearly will tell you the source map and the destination map if more than 1 map is involved in a given step. So there will be no confusion in your mind about how the Optimizer has been making its decisions when it comes to situations like I'm describing on this slide.

## Example 2: Primary Index Join: Tables in Different Maps

Be cautious moving tables that participate in primary index joins

- Best to keep tables with the same primary index (PI) on the same map
- If not, joining will require row redistribution and PI joins will not be possible
- Move them in the same time frame to preserve AMP-local primary index join



You need to be a little bit careful when you start moving tables if those tables have primary index join relationships. In other words, they share the primary index domain.

For instance, let's take an Orders table and an OrderItem table and both share Orderkey as their primary index. One of course uses it as a NUPI and the other has as primary index. When both of those tables resided in TD\_Map1, the associated rows, the rows that share the same primary index value, would both end up on the same AMP. So when we join the rows between Orders and OrderItems, it was an AMP local activity.

As soon as we move one of those tables into TD\_Map2 - and in this example we are moving the OrderItem, the larger one - to TD\_Map2, it gets its rows rehashed and replaced on different AMPs based on the same Orderkey value but it's going to pick a different AMP now because it's a different hash map.

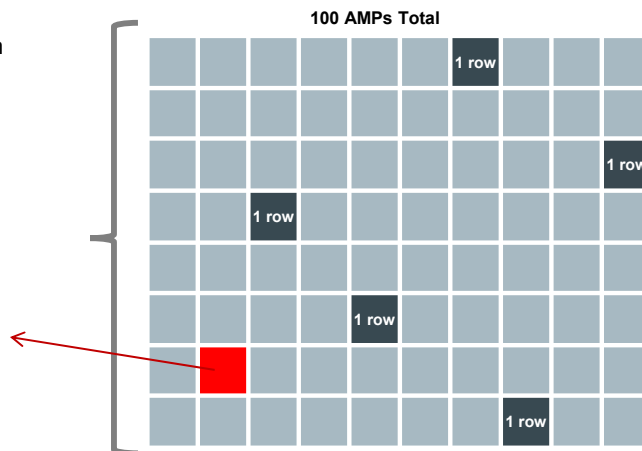
So now in our example, the Orderkey value of 55 for the OrderItem table is now on AMP 17. So we no longer can do an AMP local join between these 2 tables. So now we have introduced the need for the Optimizer to do either a redistribution or duplication; and that's exactly what will happen here.

And it will do it in a most efficient way possible. However, if you are moving tables, it usually is a good idea to identify the tables that have this kind of relationship and move them in the same Mover job.

In fact, the Viewpoint Maps Manager portlet will help you identify such situations; it will use as an input the step-level output from the DBQL step table and it will see when those primary next joins are done and will feed that information into the Analyze. And it will make recommendations to move those types of tables in the same Mover job.

## Small Tables

- A Promotion table has 5 records in a 100 AMP system
- The query accesses this table as **an all-AMP step**
- Each of the 100 AMP's must
  - Get an AMP Worker Tasks
  - Acquire Memory
  - Read the table header
  - Issue at least 1 physical I/O
- If any AMP is congested, then this could impact the completion time of the Promotion table access step



So let's take a look at why you might want to move small tables into a sparse map.

Consider a table, like a Promotion table, where you have 5 rows. If you are scanning that Promotion table, you are going to be engaging every single AMP in your configuration whether or not that AMP owns one of those 5 promotion rows. And that's going to be an all-AMP step. Every AMP is actually going to do a lot of different things - Now, these are all very, very small activities but have to be done on every AMP. Every AMP has to process the BYNET message, has to get an AMP worker task, has to read the table header, gets a little memory. It has to do at least one physical I/O even though it's not going to find anything. So all of this can add up if every AMP in the configuration is doing this unnecessary work when it is really only 5 AMPs that are being productive.

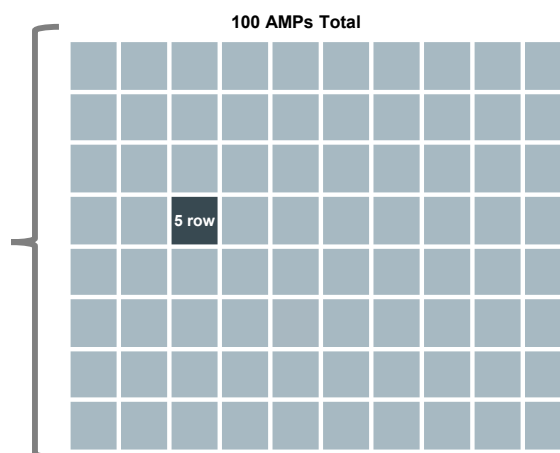
The other thing is if it got a busy system, it's possible that one AMP or maybe several AMPs are getting close to being congested - Maybe they are out of AMP tasks temporally, and they might cause a delay in getting the completion of that step done.

So, any time you can reduce the number of AMPs involved in the process, the more efficient it's going to be and the more elapsed-time consistent it's going to be.



## Small Tables Single AMP MAP

- A Promotion table has 5 records in a 100 AMP system
- The query access only a single AMP
- Single-AMP queries may qualify for TASM Tactical priority
- Each of the 99 AMP's free to service other work
  - More available AMP Worker Tasks system wide
  - Reduced demand on CPU, I/O
  - Less locking contention
  - Reduced AMP-level congestion



So if you can put - and the MAPS feature lets you do this - if we can put all of those 5 rows in one AMP, then doing a scan of a Promotion table becomes a single-AMP operation, and all of the other AMPs then are free to do other work, free to support other activities that are going on at the same time as less locking contention, certainly less congestion potential on other AMPs in the system. And it's likely to be a more consistent experience if you're repetitively accessing something like the Promotion table.

So, I/O is going to be a big reduction here because now only single AMP - the AMP with that table is going to do the I/O. And chances are, all of those 5 row are in 1 data block.

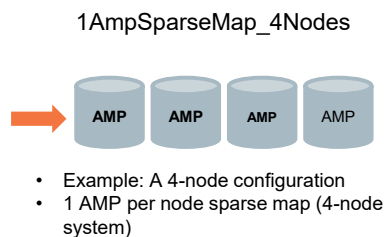
In addition, if you set up your Workload Management where a single-AMP steps or queries that only access a single AMP are given a special priority - and some people do this - they get a tactical priority for their single-AMP requests then access to the Promotion table all of a sudden becomes a tactical request and this can further accelerate the consistency and the elapsed time involving accessing that table, or similar small tables.

# Sparse Maps

**Sparse** maps include some of the AMPs within a range of AMPs

- Good candidates for sparse maps:
  - Tables with fewer rows than AMPs
  - High-usage small tables, such as dimension tables
- Monitor / Observe whether a small table continues to stay small over time

Table Size  
≤ 128KB x 4



## Benefits

- No need to wait for a hardware expansion. This feature can be used immediately
- More consistency, shorter time, for repetitive small table access
- Saves system resources when accessing small tables
- The larger the configuration, the greater the benefit

So how do you know which tables then are appropriate to move into the sparse maps?

Well, certainly any tables that have fewer rows than AMPs is a good candidate. You can probably think of a few of those in your own environment.

And among the tables that are small, you really want focus on the tables that are frequently accessed. It's not going to be much benefit to you to move tables in the sparse map if nobody ever touches it anyway.

In fact, before you start moving tables to sparse maps, it's not a bad idea to clean up your environment a little bit maybe delete those small tables that people put out there over the years for experimental reasons, or applications that are no longer in that form anymore. So, think about doing some cleanup work before you move your tables.

In addition, we are going to give you some guidelines on what tables are appropriate for sparse maps - There's a map sizing view that's available to you that you can run at any time, and it will tell all the tables that are appropriate for the single AMP or multi-AMP sparse maps. It uses these criteria that I am showing you on this slide. Single-AMP sparse maps are appropriate for tables that are 128KB or less. And this is minus the table header. What this view does is it strips off the table header overhead and just looks at the perm data size minus table header. If that perm size is less than 128KB, that's a good choice to move that table to a single-AMP sparse map.

For the multi-AMP sparse maps, it's a sort of the same guideline, but just take that 128KB size and multiply it by the number of AMPs that multi-AMP sparse map covers. On a 4-node system, that would be 4 AMPs; so you just take tables that are 128KB times 4. Tables that are smaller than that would be a great fit for that multi-AMP sparse map.

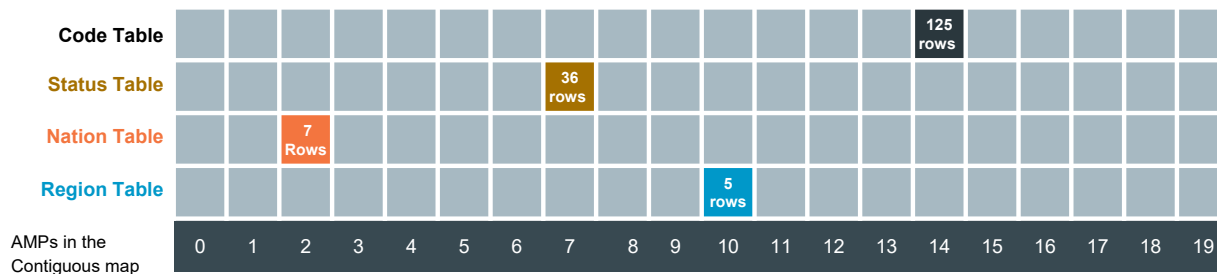
I just want to give you one word of warning here on empty tables. It may be tempting to move all of your

empty tables into sparse maps. However, some of these empty tables you need to be careful with because they may be used as staging tables in ETL processes. If you were to move them, for example into a single AMP sparse map, and you run your ETL processes that night, you may find that they are staging for a million rows or more. And that's going to severely impact that ETL job. So be vigilant about empty tables.

And the other tip I want to leave with you here is that some tables are small today, but they might be bigger tomorrow. Maybe Promotions today for us has 5 rows - maybe we only have 5 promotions - but our company is growing and next month or three months out maybe we've got 4,000 promotions or 10,000 promotions. So occasionally you need to review these small tables. It is very easy to do - Viewpoint will do it for you, or the table sizing view I mentioned will do it for you as well. But make sure that's in your mind somewhere when start working with sparse maps.

## Sparse Maps: System Determines Which AMPs to Use

- Many tables can be moved into the same sparse map
- The system will determine which AMPs will support each table's rows
- The database attempts to balance AMP assignments across nodes
  - Spreading out table-to-AMP assignments reduces pressure on any single AMP



Four very small tables in the same single-AMP sparse map

So we got a single-AMP sparse map. Let's say we've got 100 small tables. How do we control which AMPs those tables are going to be targeted to when they are moved into the sparse map? Do we need to create an individual sparse map for each of these tables? Well - No. In fact you don't even have to think about which AMP these tables are being placed in because the Database is going to do this for you.

A single-AMP sparse map can be home for hundreds of small tables. The system will automatically decide for you where your small table is going to reside, which AMP is going to be a recipient of its rows when you move it to, for example, a single-AMP sparse map.

And when the system does this, it attempts to spread these tables across different AMPs. Not only different AMPs but different nodes. So the first table is going to go on 1 AMP on node A. And second table you move into a sparse map is going to go on another AMP on node B, and the third table you move into the same sparse map is going to go on some AMP on node C, and so forth.

And then we go back and fill in other AMPs as needed. So if you got lots of AMPs in your configuration, you may be just fine with one single-AMP sparse map. In fact most of you will be. And the same as too for your multi-AMP sparse maps. The system will spread among different AMPs reducing pressure on any single one AMP and really minimizing the chance that there's going to be a bottleneck.

## Sparse Maps – COLOCATION

- **COLOCATION** determines which AMPs by hashing name using parent contiguous map
- AMPs not chosen until CREATE TABLE



```
CREATE TABLE Nation,
MAP=TD_1AMPSparseMap_2Nodes
COLOCATE USING Co_NationRegion
  (NationKey INTEGER,
   Name      VARCHAR(10),
   RegionKey INTEGER)
UNIQUE PRIMARY INDEX (NationKey);
```

```
CREATE TABLE Region,
MAP=TD_1AMPSparseMap_2Nodes
COLOCATE USING Co_NationRegion
  (RegionKey INTEGER,
   Name      VARCHAR(10))
UNIQUE PRIMARY INDEX (RegionKey);
```

A contiguous map covers a range of AMPs, usually across some number of nodes in the system. A sparse map is the opposite; it covers just a single AMP, or under some conditions a small number of AMPs. A sparse map does not exist in isolation within its own dedicated AMP or AMPs, but rather it is defined within a parent contiguous map. The rows of a table targeted to a sparse map coexist with the rows of tables that belong to the parent contiguous map.

A sparse map is intended as a destination for the rows of “small tables”. A small table that would be suitable for a single-AMP sparse map is one that contains 128KB or less, after removing table header overhead. If the table is being considered for a multi-AMP sparse map, the upper limit of suitability is the number of AMPs within the multi-AMP sparse map times 128KB.

For example, if the permanent space of a small table minus table header overhead is 100KB, then that table is suitable for a single-AMP sparse map. If the perm space of a table (minus table header overhead) is between 500KB, it would be suitable for a 4-AMP sparse map ( $128\text{KB} * 4 \text{ AMPs} = 512\text{KB}$ ), and so forth.

Accessing tables with small amounts of data, where some AMPs have no rows, can be inefficient on large systems. Unnecessary resources are spent trying to find rows on AMPs that do not carry them. If that large system is expanded, the number of empty AMPs supporting a small table increases, driving up this overhead. Using the default single-AMP sparse map for a small table allows all-row access to become a single-AMP retrieve step that only runs on the one AMP where all the data is located. While particularly helpful for very large systems with thousands of AMPs, moderate-sized or smaller systems may also benefit from sparse maps, although to a lesser degree.

## Colocation Name

### Sparse Tables and the Colocation Name

- Tables in sparse maps reside on a subset of AMPs in the parent contiguous map
- MAPS introduces the *Colocation Name* parameter, the purpose of which is twofold:
  - It is the mechanism used to determine which AMPs a table utilizing a sparse map will reside on
  - Can be used to insure that two tables with the same sparse map are colocated on the same AMPs
- The *Colocation Name* parameter is specified as part of a CREATE TABLE statement and can have any value
- The *Colocation Name* can only be specified for tables that utilize sparse maps
- To view the *Colocation* name, either SHOW TABLE or use the Colocation column in DBC.TablesV
  - To list the *Colocation* names in a system,

```
SELECT DISTINCT(ColocationName) FROM DBC.TablesV ORDER BY 1;
```

The user may or may not specify a colocation name at the time the table is created or moved. If one is not specified, the system constructs one from the table's database name, underscore, and the table's name. If you issue a SHOW TABLE command, you can see both the sparse map name and the colocation name associated with the table.

### Selecting a Colocation Name

If you have small tables that require colocation, then it is recommended that you come up with a meaningful colocation name to distinctly represent those tables. If you do not have a need for collocating multiple tables, let the system generate the colocation name for you and simply ignore the colocation naming option.

The colocation name is put through the hashing algorithm and the resulting hash bucket is used to identify which AMP(s) will house that small table. This process that selects an AMP or AMPs only considers AMPs that make up the parent contiguous map.

To view all colocation names in a system,

```
SELECT DISTINCT(ColocationName) FROM DBC.TablesV ORDER BY 1;
```

## Foreign Tables

- MAP can be defined for a foreign table

```
CREATE FOREIGN TABLE Temp,  
EXTERNAL SECURITY DEFINER TRUSTED CEPH_AUTH2,  
MAP = My1AMP_SPARSE  
USING ( LOCATION ('<some path>')  
STOREDAS ('PARQUET'))  
NO PRIMARY INDEX PARTITION BY COLUMN;
```

- Multiple foreign tables can use the same multi-AMP sparse map

- Each foreign table will be assigned a different set of AMPs on each of the nodes in the configuration
    - There is no need to create a different sparse map for each foreign table

- MAP clause would be honored only if

- When a map is created at the user level and tables are created with the same map and with the same user
  - When a map is created at dbc/any user level and granted the map to executing user and tables created with that user will honor the map

- MAP clause if not honored will get overridden with the default map

If you only have a single foreign and you expect high concurrency against it, it may be better to define multiple copies of the foreign table, each with a different table name, with each using the same sparse map. That will spread out the AMPs used by the foreign tables and ensure that users will not be creating hot spots on the same subset of AMPs.

## Moving Tables – New Map (1 of 2)

**ALTER TABLE** used to move a table to a new map.

- `ALTER TABLE Table1, MAP=TD_Map2;`
- Similar performance (maybe faster) than Reconfig

### ALTER TABLE

- Old AMPs:
  - Read all rows and send all rows
  - Insert old rows
- New AMPs:
  - Insert new rows

### Reconfig

- Old AMPs:
  - Read all rows and send some rows to new AMPs
- New AMPs:
  - Insert new rows
- Old AMPs:
  - Delete the rows sent

If you decide that a table should use a different map, either issue an **ALTER TABLE** request to name a different map for the table or use the Mover stored procedure, which is provided with the TDMaps user that we will explain later in this module.

The form of an **ALTER TABLE** to move a table to another map leaves the original table in its entirety in the source map and performs an **INSERT-SELECT** into a second instance of the table that resides in the destination map. At the end of a successful **INSERT**, this new **ALTER TABLE** updates the map name for the table and instructs the file system to drop the source data. From that point on, the version of the table residing in the new map is used.

The key benefit of this map-specific **ALTER TABLE** is that all of the data in the source table can continue to be accessed while the move is taking place, as is true of any **INSERT-SELECT**. A read lock is placed on the source table while rows are being inserted into the destination table, but is then updated very briefly to an exclusive lock at the end of processing. A secondary benefit is the move can be aborted and the rollback is a quick drop of the data in the new map.

One tradeoff, however, is that two versions of the table are in existence while the **INSERT-SELECT** is executing. Twice as much space is required compared to the traditional **ALTER TABLE** which makes changes one cylinder at a time.



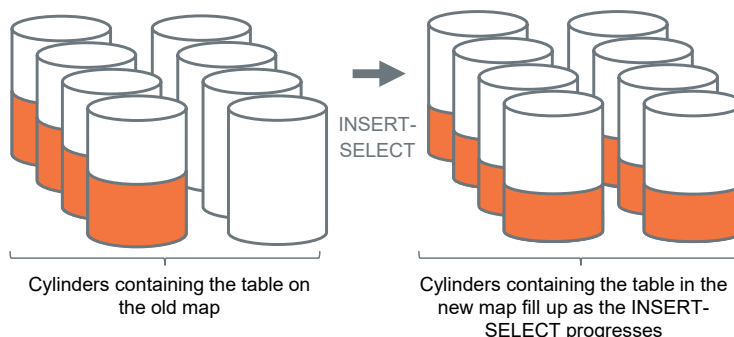
## Moving Tables – New Map (2 of 2)

When the “Maps” clause appears in the ALTER TABLE

- The operation is converted into an INSERT-SELECT

- **Read** lock on the source table
- Upgraded to **Exclusive** lock at transaction end

- Data is accessible from the source table as the ALTER TABLE proceeds
- When complete, updates the table name and deletes the source table
- A new “Mover” Stored Procedure packages the ALTER TABLE and includes the Summary statistics collection when the move is complete



So how do you move your tables then, from TD\_Map1 to into the new larger map, TD\_Map2. This is done by means of a new version of the ALTER TABLE.

And this ALTER TABLE, under the covers, is actually an INSERT-SELECT, which is really great for moving tables in this context because you can read the data in the source table while the target table is being loaded so availability of your data increases using this technique.

When it's complete, the ALTER TABLE will drop the source table and do whatever it has to update the dictionary and make sure that we now know that this is the table and its new instance; and it's now on TD\_Map2.

There is a "Mover" Stored Procedure we have created for you that we recommend you use - I recommend you use - because it makes things a little easier and it does some of the few extra things for you - you don't have to all of your privileges exactly the way you would have to have under if you did the ALTER TABLE directly.

So it's easier with the privileges. And the Stored Procedure, the Mover Stored Procedure collects summary statistics for you When the move is complete so your statistics will be up to date.

## ALTER TABLE – Possible Issues

- Join/Hash Index with ROWID will be marked invalid after NoPI table is ALTERed
- Table that has permanent journaling enabled will cause the ALTER will fail
- Not enough free permanent space to ALTER table (space issues)
  - Depends on the number of workers
  - ALTER TABLE uses INSERT-SELECT of the entire table so 2x space is used.
    - CP and NoPI tables use 3x space (spool is needed to generate new RowIDs)
  - Administrator must allocate space, if necessary, before starting the job
  - If there isn't enough space available
    1. Then offline RECONFIG is a choice where the table is added to the reconfig order table
      - Not recommended
    2. View with UNION ALL
      - For a very large table with partitioning based on time, consider moving only the current partition into the new system-default map, and then using a UNION in a view to connect the history partitions which remain in the older map

Special consideration should be given to join indexes when used on columnar and “No Primary Index” (NoPI) tables if the join index contains the ROWID derived column of a base table. These join indexes are invalidated when the base table is moved to a new map.

### Space Issues

For columnar and No Primary Index (NoPI) tables, three times the space is required because an intermediate spool is used to sort the rows. This sort is required so that new row IDs can be generated for those types of tables. An extra step can be seen in the Explain text for the INSERT-SELECT in cases such as this.

The table below delineates the tradeoffs involved in using Reconfig versus the Teradata Viewpoint Maps Manager portlet Mover job (or manual ALTER TABLE) when considering how to get a very large table to a new contiguous map:

### Space requirements

Reconfig: Doesn't require any additional space, a practical choice when there is a shortage of free space on old AMPs.

Viewpoint Mover / ALTER TABLE: Requires free space on each AMP (including the old AMPs) for the new table rows, total space needs will be double the current table size.

### Availability

Reconfig: Table data, as well as the entire system, is unavailable during the Reconfig process.

Viewpoint Mover / ALTER TABLE: Table data is available during the move.

## GRANT MAP

- Before the MAP word can be used in a SQL statement, the map must first have been granted
- An exception is if the map is the default map (TD\_Map1 or after 1st expansion TD\_Map2)
  - How can a customer restrict/force map usage for a user or group of users?
  - The answer is: A User/Database/Profile can have an optional default map that can be specified and used to force specific map usage
- Creator of the sparse map automatically gets GRANT MAP
- You can Grant Map to a Role or User
  - `GRANT MAP TD_1AmpSparseMap_1Node to GL_ACCESS_R;`
- DIPMAPS calls TDMaps.CreateInitialSparseMaps to create two sparse maps
  - The TDMaps user automatically gets the GRANT MAP privilege on these created sparse maps

Before you can add the clause “MAP = map-name” to a CREATE or an ALTER command, you must have been granted the map. However, this grant is not required when the map that you specify in your request is your default map. So, if you are only creating and moving tables within the scope of your own default map, then you do not have to be concerned with having new privileges established.

Your default map is determined first by the map-name carried in your Profile, if one exists. Secondly, it is determined by the map-name associated with your User record, if one exists. If neither Profile nor User carries a map-name, then the system-default map established by the Reconfig utility becomes your default map.

In addition, MAPS feature introduces two privileges: CREATE MAP and DROP MAP. These privileges must be granted for an administrator to be able to create maps and drop maps. These privileges can be revoked.

Additionally, before the MAP clause can be used in a SQL statement such as CREATE TABLE/DATABASE/USER/ PROFILE, the map itself must first be granted via a GRANT MAP statement unless the map specified is the same as the map that would be chosen by default. User tables, join indexes, and hash indexes always have a map, but specifying the map is optional. If the map is not specified, a default map is chosen.

For example, if TD\_Map2 is your default map, you can issue SQL without either naming a map or being granted that map. The table is created in your default map, TD\_Map2.

## Miscellaneous SQL and Hash Functions

```
SHOW MAP TD_Map1;
```

```
CREATE MAP TD_Map1
FROM TD_Map1
CONTIGUOUS
AMP BETWEEN 0 AND 15;
```

```
SHOW MAP TD_1AmpSparseMap_1Node;
```

```
CREATE MAP TD_1AmpSparseMap_1Node
FROM TD_Map1
SPARSE
AMPCOUNT = 1;
```

```
SELECT MapName FROM DBC.MapsV WHERE SystemDefault = 'Y'; →
```

MapName  
TD\_MAP1

```
SELECT HASHAMP(MAP = TD_MAP1) + 1 AS #AMPS; OR
SELECT HASHAMP (HASHBUCKET (HASHROW (0)) MAP = TD_MAP1) + 1 AS #AMPS;
```

#AMPS  
16

```
SELECT HASHAMP (HASHBUCKET (HASHROW (0))
MAP = TD_1AmpSparseMap_1Node
COLOCATE USING Co_NationRegion) AS PrimaryAMP#;
```

PrimaryAMP#  
2

```
SELECT HASHBAKAMP (HASHBUCKET (HASHROW (0))
MAP = TD_1AmpSparseMap_1Node
COLOCATE USING Co_NationRegion) AS FallBackAMP#;
```

FallBackAMP#  
3

There are two different designations that may be assigned to a contiguous map, one at the system level and one at the user level. Both these designations use the term “default” and are explained below:

**System-default:** Reconfig sets what is called the “system-default” map. This is the new, larger contiguous map that is created by Reconfig. SystemDefault is the name of the column in the DBC.Maps table.

**Default:** An assigned, specified “default” map is chosen by the database based on the Profile→User→System hierarchy.

### Recommendation: Rely on the System-Default Map

It is recommended that the system-default map be used as much as possible as a means of associating new tables with maps, rather than coding specific maps for a given set of users. This guarantees that new tables will be located in the most expansive map and benefit from the highest level of parallelism. In addition, when the system expands, any map entries in Database, Profile, or User records will have to be revisited.

### HASHAMP / HASHBAKAMP Hash-Related Functions

You can specify an existing contiguous or sparse map for the table. Optionally, you can specify a colocation name so that tables can share a sparse map. When two tables that share the same sparse map are joined on a primary index or primary AMP index, the tables reside on the same AMPs to avoid a redistribution of the rows.

## Summary

Why is the MAPS feature useful?

- Eliminates the extended downtime for Reconfig
- Improves performance for small tables by putting them in Sparse Maps

TDMaps

- Contains stored procedures to help the movement of tables since Reconfig will not be moving the tables
- Viewpoint: Teradata Maps Manager Portlet
  - Viewpoint calls the stored procedures in user TDMaps to simplify moving tables

Additional considerations comparing Reconfig to Viewpoint Mover / ALTER TABLE:

### Restart support

Reconfig: In case of failure, can restart and continue from the cylinder where the job left off.

Viewpoint Mover / ALTER TABLE: No restart or rollback, will drop the target table, source still available, can restart the job.

### Control over execution time

Reconfig: If runs longer than expected, can't stop the job once it starts.

Viewpoint Mover / ALTER TABLE: Can abort the job and immediately access the source data.

### Special caveats

Reconfig: Will prolong system down time.

Viewpoint Mover / ALTER TABLE: Not a viable choice if adequate free space is not available.

If neither of those two options satisfies your needs, there is a third option: Split the table in two and use a UNION within a view. With a very large table that is partitioned by date, you have the option of only moving the current partition of that table to the new contiguous map, and leaving the majority of the partitions in the old map. Over time, more and more current data will be inserted into the version of the table in the new, more expansive map. Also, over time, older partitions in the old map can be dropped when no longer needed and more recent partitions in the old map can be moved to the new map; eventually, the table on the old map will be empty and can be dropped.



## Module 7: Viewpoint Overview

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

- Describe the components of the Teradata Viewpoint
- Describe how to navigate portal pages and portlets
- Configure Viewpoint to support one or more Teradata systems
- Setup data collectors for system monitoring
- Use portlets to identify and resolve potential system resource issues
- Leverage query monitoring portlets to recognize and analyze problem queries



## Topics

- Viewpoint Portal Basics
- Administration Portlets
- System and Query Monitoring Portlets





## Current Topic – Viewpoint Portal Basics

- **Viewpoint Portal Basics**
  - Overview
  - Portlet Controls
  - Rewind
  - Dashboard
- Administration Portlets
- System and Query Monitoring Portlets



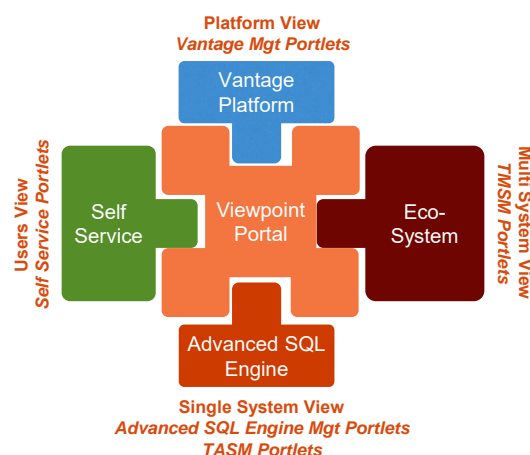
# Teradata Viewpoint

## Viewpoint is the cornerstone of Vantage Systems Monitoring and Management.

- Provides systems management via a web browser.
- Provides a single operational view (SOV) for a multi-system environment.
- Highly customizable and can be personalized.

## Viewpoint 22.10 Supported Browsers

- Mozilla Firefox 96+
- Google Chrome 97+
- Apple Safari 12.1.1+
- Microsoft Edge 97+
- Google Chrome for Android 70+
- Safari for iOS 9.3.5+



Teradata Viewpoint is intended as a Teradata customer's Single Operational View (SOV), meaning it supports various Teradata systems including Teradata Vantage Advanced SQL Engine, Teradata Vantage Machine Learning Engine, Teradata Hadoop System, and Teradata QueryGrid.

It provides web-based interface (a set of portals) for a wide range of capabilities and features, such as monitoring, management, alerting, and others. It serves both system administrators as well as business users. It also serves as user interface for other Teradata products, for example Teradata Data Lab.

Teradata Viewpoint provides systems management via a web browser which is extensible to Teradata end users and management, allowing them to understand the state of the system and make intelligent decisions about their work day.

Administrators can use Teradata Viewpoint to determine system status, trends, and individual query status. By observing trends in system usage, system administrators are better able to plan project implementations, batch jobs, and maintenance to avoid peak periods of use. Business users can use Teradata Viewpoint to quickly access the status of reports and queries and drill down into details.

The Data Collection Server (DCS) is a component of the Viewpoint Framework and is responsible for retrieving performance and monitoring data for Teradata. Most of the data collected by the DCS is transient in Teradata so the DCS extracts it and persists it for historical analysis. The data is collected by communicating via JDBC to Teradata's SQL, Monitor and Console partitions. The DCS is capable of collecting data from multiple Teradata instances as well as Aster and Hadoop.

## Viewpoint Portal and Portlets

- A Viewpoint portal represents a Web location that can be accessed or logged onto
- Each *portal page* is a virtual workspace where you decide which *portlets* to add/ display and how to arrange them on the page
  - Viewpoint continually updates the information displayed on the portal page

### System Overview

- Alert Viewer
- Canary Response Times
- Elastic Performance
- Hadoop Services
- Productivity
- Space Usage
- System Health
- Today's Statistics
- TVS Monitor
- Viewpoint Monitoring

### Node Overview

- Node Monitor
- Node Resources

### Workload Management

- Workload Designer
- Workload Health
- Workload Monitor

### Utilities

- Remote Console\*
- SQL Scratchpad
- Viewpoint Calendar

### Applications

- Data Labs
- MAPS Manager
- QueryGrid
- Stats Manager
- Performance Data Collection

### Session Management

- Application Queries
- Completed Queries
- Lock Viewer
- My Queries
- Query Groups
- Query Log
- Query Monitor
- Query Spotlight

### Trends

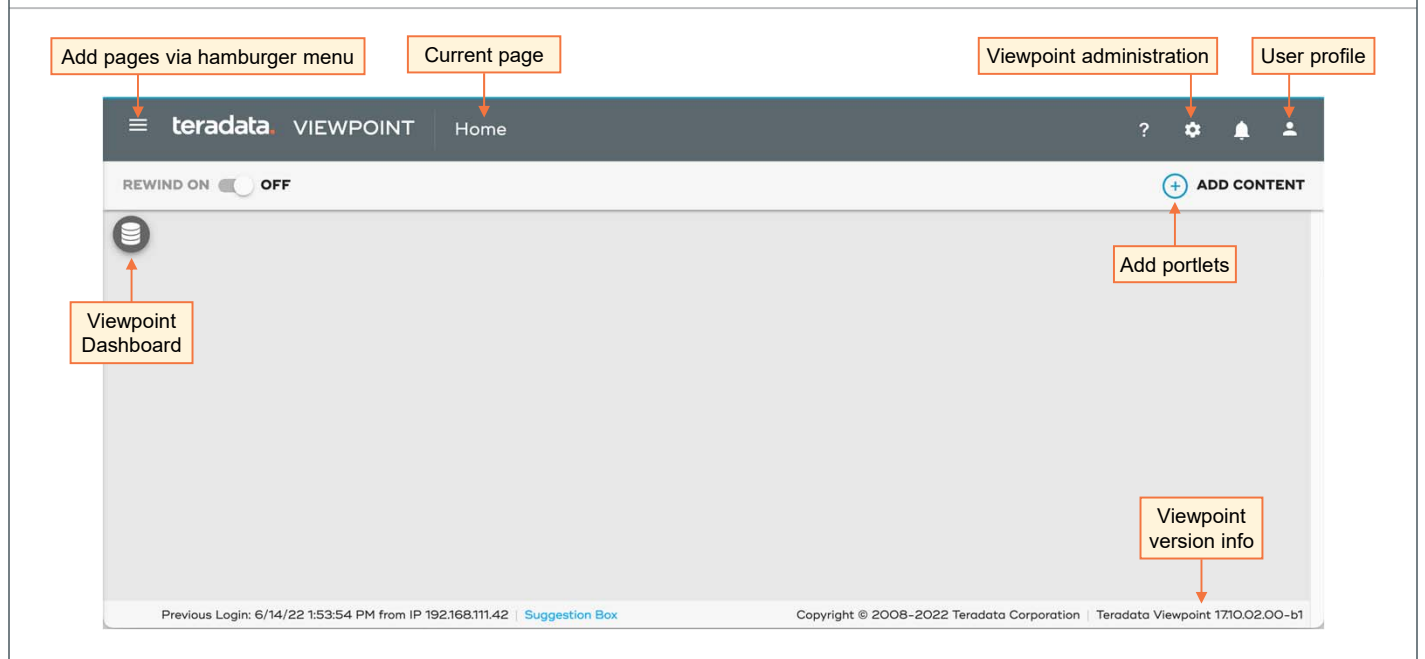
- Metric Heatmap
- Metrics Analysis

### Others

- Unity, Ecosystem Manager
- Data Mover
- DSA / BAR

\*Remote Console is disabled by default in Portlet Library

## What it Looks Like After First Time Login



### Admin Menu

Available only to users with permissions to Viewpoint administrative portlets, this menu is the central location for configuration and management of Viewpoint.

### Portal Pages

Users may use one or many “pages” to display portlets in Viewpoint. These pages may be created by the user or shared from other users. When a page is selected, it will be displayed and all portlets on that page will be continuously updated. This is referred to as the active page. Portal Pages may be added by clicking the Add Page menu.

### Add Content Menu

This menu lists all available portlets by type. Users can choose to add one or more portlets to their portal page. Users will only see portlets that have been assigned to their role or user.

### Suggestion Box

If there are other features missing that are important to you, report them using the Suggestion Box link, located at the bottom of the Teradata Viewpoint portal.

## Add Portlets to the Current Page

**2. Click Add to add Portlets**

**1. Select portlets to be added to a page by clicking on its name**  
(Clicking twice on a portlet will put two of the same portlet on your page)

**ADD CONTENT**

**New portlets are added to a Page via the Add Content**

### Add Content

One can add one or more portlets in an operation including multiple instances of the same portlet if desired. There is also a search option at the top to assist in finding the right portlet. Portlet category groupings assist in search and also understanding of portlet relationships. Lastly, notice that all portlets have a unique representative icon.

# Portlet Controls

The screenshot shows the 'Query Monitor' portlet interface. Key controls are highlighted with orange boxes and arrows:

- Portlet Name:** Points to the 'Query Monitor' title.
- Time Stamp:** Points to the '3:44:01 PM' timestamp.
- Pause/Resume:** Points to the circular refresh icon.
- Portlet Menu:** Points to the three-dot menu icon in the top right.
- Table Actions:** Points to the 'Table Actions' dropdown menu, which includes options like 'Settings', 'Maximize', 'Share', 'Help', and 'Remove'.
- Information Balloons:** Points to a 'Metrics Analysis' graph showing 'AWT (Factory)' with a value of '2.656'.
- Export...**: A callout box explains: 'Export data displayed in the portlet table to a comma-separated values (CSV) file for further analysis and formatting.'
- Adjust height:** Points to the 'Adjust height' button at the bottom right.
- Restore height:** Points to the 'Restore height' button at the bottom right.

The main table displays query statistics with columns: SESSION ID, STATE, USERNAME, ACCOUNT, and ΔCPU. It shows 33 rows total.

## Portlet Controls

The Teradata Viewpoint Administrator controls which portlets and what features you are authorized to use. If a portlet or feature is not available, check with the Teradata Viewpoint administrator for permission. Each portlet instance has its own settings and controls, as well as the features common to all portlets. Not all portlets have all controls.

## Time Stamp

Displays the last time data was retrieved from the Teradata server. The time is based on the time zone set in the Profile portlet.

## Pause, Resume

Pauses or resumes refreshing data for a portlet.

## Restore Height

Resets the portlet to its default height. The button appears in the bottom right of the portlet frame after you change the default height of the portlet.

## Information Balloons

Displays detailed information when you hover over graphs and other objects in the portlet.

## Table Actions

Provides controls for managing information displayed in the table.

## Exporting Table Data

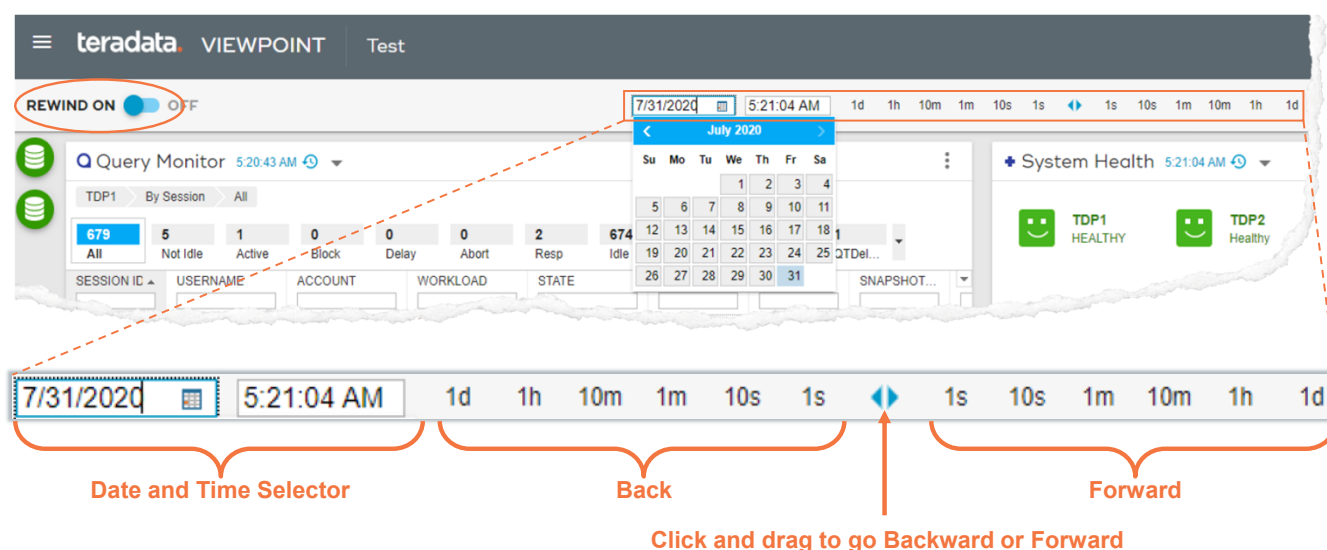
You can export data to a .csv file for further analysis and formatting. The exported .csv file contains all available data up to one million rows.

If filters are used, only filtered data is exported for most portlets. If a portlet displays more than one

million rows of data, then use the filters to display the data you want before exporting it.

## Viewpoint Rewind

Rewind, replay, and fast forward Viewpoint portlets to review Teradata operations at past points in time.



The rewind feature allows you to view data that corresponds to dates and times in the past and compare it to data for a different date and time. You can rewind the data for some or all portlets on a portal page to a previous point in time, such as when a job failed. Rewinding portlet data is useful for identifying and resolving issues.

You can rewind data as far back as data is available. The rewind feature is not available for portlets that have portlet-specific methods for reviewing data over time.

Using the rewind toolbar, you can enter a specific date and time as well as scroll through the data in increments of seconds, minutes, hours, or days. All portlets on the page that are participating in rewind activities display data that corresponds to the selected rewind date and time each time a selection is made.

Three ways to rewind:

- Enter a specific date and/or time
- Drag left/right to scroll back/forward through time in increments
- Clickable increments include: 1 sec, 10 sec, 1 min, 10 min, 1 hour, 1 day

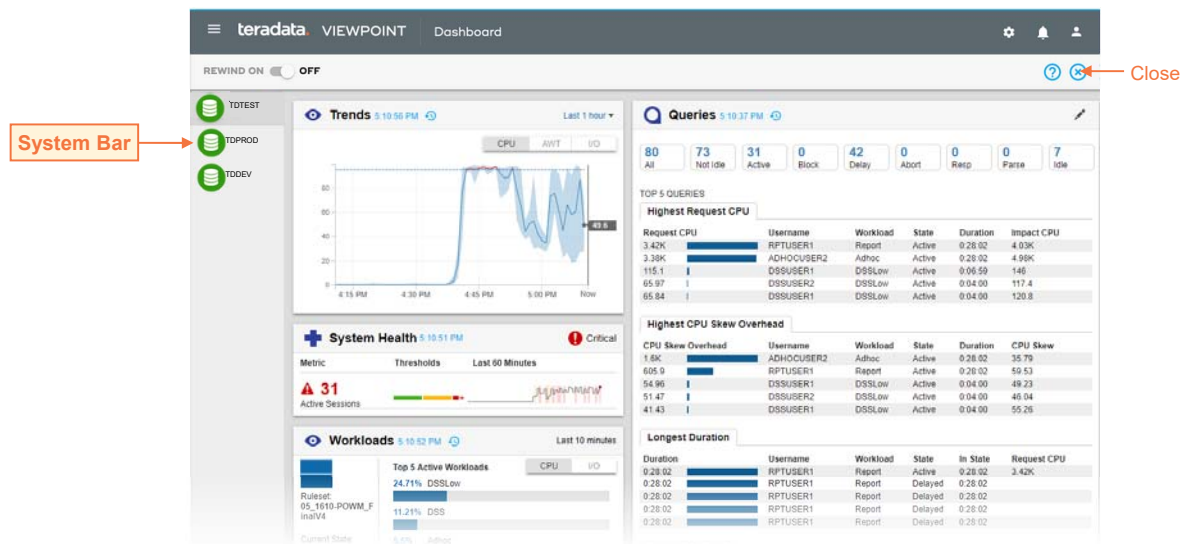
Portlet Control Pause, Resume

- Pauses or resumes refreshing data for a portlet.
- If rewind is on, you can select to Unlink from Rewind or Link to Rewind.
- If rewind is off, you can select Refresh to bring the most recent data collection from the Viewpoint server to the portlet.



# The Viewpoint Dashboard

The **DASHBOARD** provides access to the most commonly used information about a system including **System Health**, **Workloads**, **Queries**, and **Alerts**.



Viewpoint Dashboard allows you to view the current state of all Teradata Database, Aster, and Hadoop systems. It provides a high-level view to monitor overall system activity allowing you to rapidly address issues.

When expanded, the Dashboard initially shows an overview for the selected system. For this at-a-glance system overview, there are 5 main content areas:

## Trends

Monitor system metrics and analyze system resource usage trends.

## System Health

View the current state of monitored systems and key metrics against defined thresholds and then investigate metrics exceeding healthy thresholds.

## Workloads

Monitor the flow of queries to identify any unusual conditions related to workload performance.

## Queries

View and manage queries running on Teradata Database and Teradata Aster systems so you can target problem queries and take action to correct problems.

## Alerts

View alerts triggered for systems.

## Hadoop Services

View service status and monitor key performance indicators on a Hadoop system.

If the Viewpoint Dashboard is not available, the Teradata Viewpoint Administrator can assign

permissions.

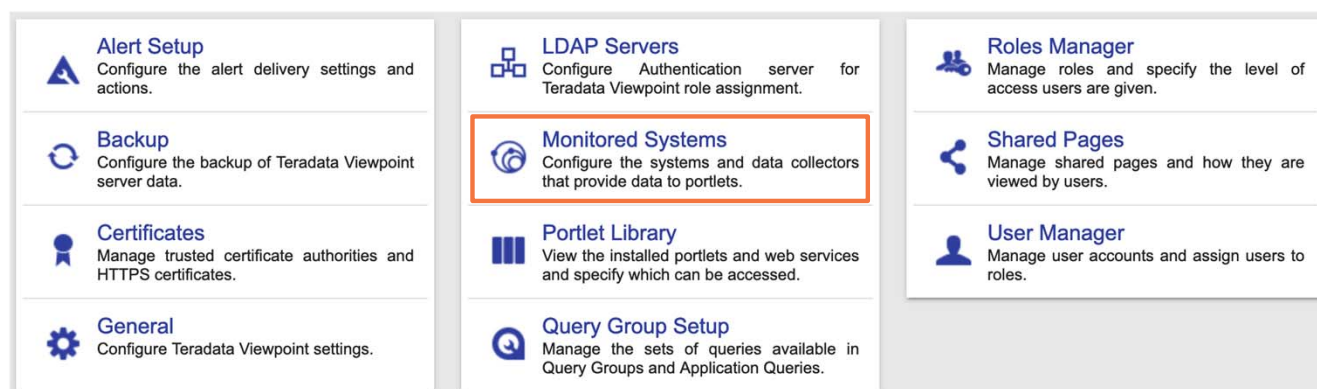
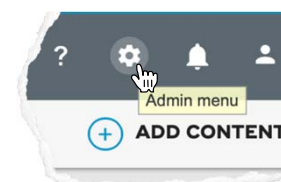
## Current Topic – Administration Portlets

- Viewpoint Portal Basics
- **Administration Portlets**
  - **General Settings**
  - **Monitored System**
- System and Query Monitoring Portlets



## Administration Portlets

The Teradata Viewpoint administrative portlets allow the Viewpoint Administrator to **configure access** to Vantage and Viewpoint resources. The Administrative portlets are available from the **Admin Portlet** button.



You can access these portlets from the Teradata Viewpoint portal page if your role has permission.

- Alert Setup: Configure delivery settings and actions that are triggered when the alert service generates an alert.
- Backup: Configure the backup of Teradata Viewpoint server data.
- Certificates: Manage trusted certificate authorities and HTTPS certificates.
- General: Configure Teradata Viewpoint settings.
- LDAP Servers: Configure the LDAP (Lightweight Directory Access Protocol) servers for Teradata Viewpoint to authenticate users and assign user roles.
- Monitored Systems: Add the systems and configure the data collectors that provide data to portlets.
- Portlet Library: View the installed portlets and specify which portlets can be enabled.
- Query Group Setup: Manage the sets of queries available to users in the Query Groups and Application Queries portlets.
- Roles Manager: Manage roles and specify the level of access users are given.
- Server Management: Configure Server Management options.
- Shared Pages: Manage shared pages and how they are viewed by users.
- User Manager: Manage user accounts and assign users to roles.

## Monitored Systems Portlet

The **MONITORED SYSTEMS** portlet allows the Teradata Viewpoint Administrator to **add, configure, enable, and disable systems**, as well as view the amount of disk space used and set a threshold for a disk usage alert.

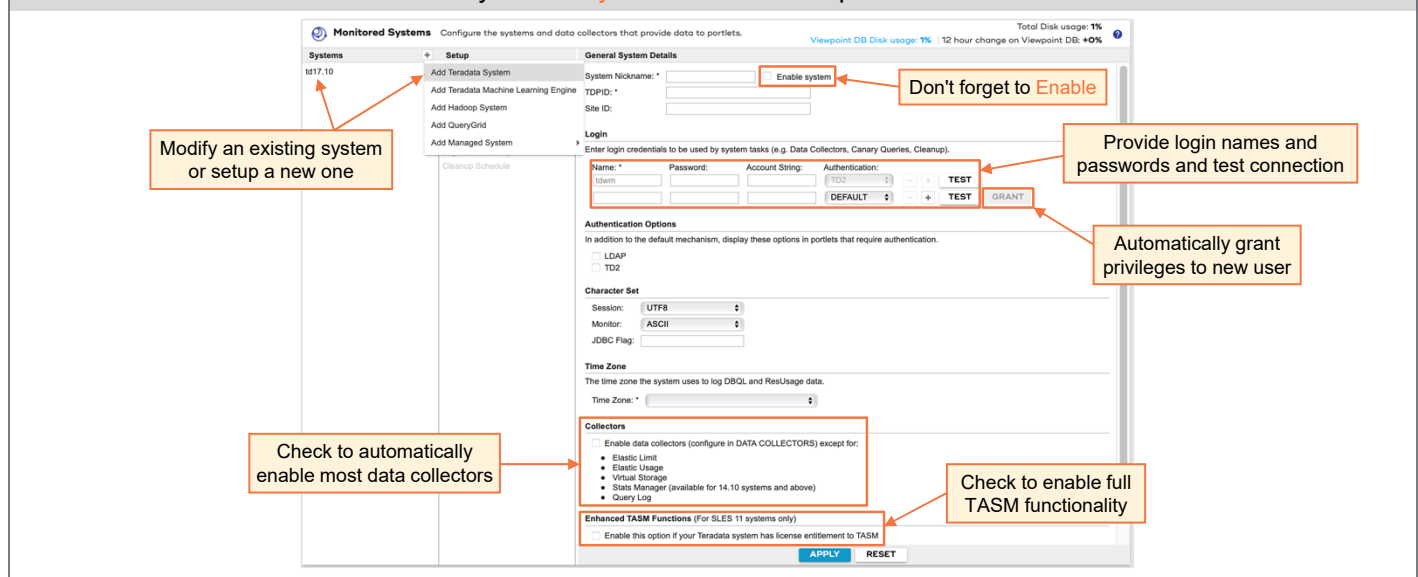
- General – Configure the system nickname, **TDPID, login names, passwords**, and account strings (optional)
- Data Collectors – Enable, disable, and **configure data collectors** to capture and retain portlet, disk usage, and resource data
- System Health – Enable **metrics for the SYSTEM HEALTH** portlet. Configure degraded and critical thresholds for each metric
- Canary Queries – Configure canary queries used to **test** Advanced SQL Engine **response** times
- Alerts – Add, delete, copy, and **configure alerts**
- Monitor Rates – Set Advanced SQL Engine **internal sample rates** for Sessions, Node logging, and Vproc logging
- Log Table Clean Up – Select system log tables to **clean up**
- Clean Up Schedule – **Schedule cleanup** of system log tables

The **MONITORED SYSTEMS** portlet allows the Teradata Viewpoint Administrator to add, configure, enable, and disable Vantage systems using specific dialog boxes:

- General - Configure the system nickname, TDPID, login names, passwords (hidden), and account strings (optional). Test the connection to Advanced SQL Engine, and add or delete login names.
- Data Collectors - Enable, disable, and configure data collectors to capture and retain portlet, disk usage, and resource data.
- System Health - Enable metrics for the **SYSTEM HEALTH** portlet. Configure degraded and critical thresholds for each metric.
- Canary Queries Configure canary queries used to test Advanced SQL Engine response times. System Heartbeat - canary query cannot be removed.
- Alerts - Add, delete, copy, and configure alerts, or migrate existing Teradata Manager alerts.
- Monitor Rates - Set Advanced SQL Engine internal sample rates for sessions, node logging, and vproc logging.
- Log Table Clean Up - Select system log tables to clean up.
- Clean Up Schedule - Schedule clean up of system log tables.

# Monitored Systems Portlet – General

Button:  (Viewpoint Administration) > Portlet: Monitored Systems   
Systems: System Name > Setup: General



The screenshot shows the 'Monitored Systems' portlet in the 'Setup: General' tab. The interface includes a 'Systems' list on the left, a 'General System Details' section, a 'Login' section, 'Authentication Options', 'Character Set', 'Time Zone', 'Collectors', and 'Enhanced TASM Functions'. Annotations with arrows point to various fields and checkboxes:

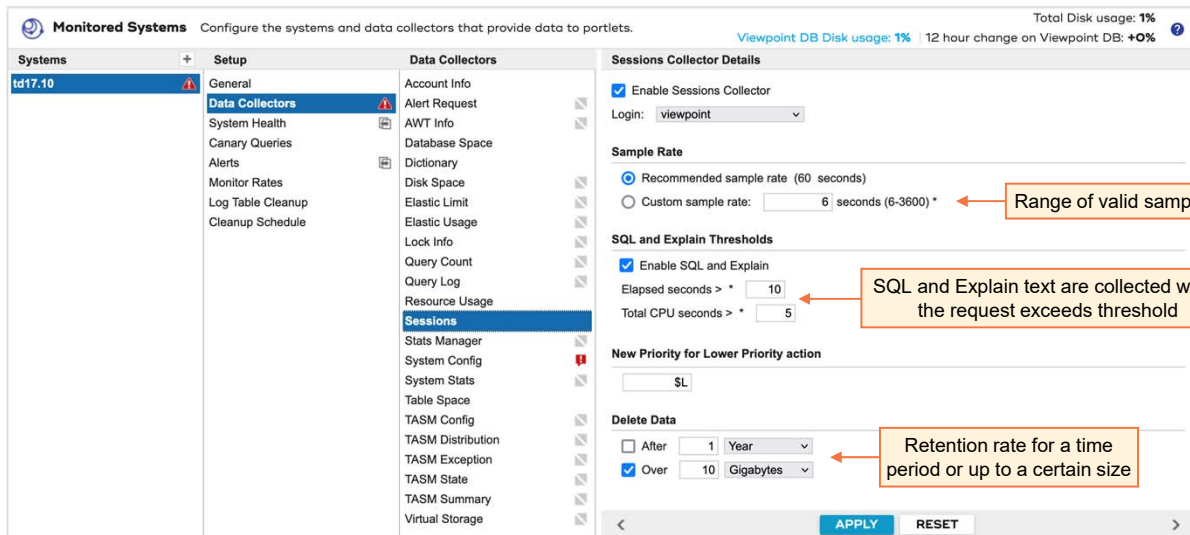
- Modify an existing system or setup a new one:** Points to the 'Systems' list on the left.
- Don't forget to Enable:** Points to the 'Enable system' checkbox.
- Provide login names and passwords and test connection:** Points to the 'Name', 'Password', 'Account String', and 'Authentication' fields, as well as the 'TEST' button.
- Automatically grant privileges to new user:** Points to the 'GRANT' button.
- Check to automatically enable most data collectors:** Points to the 'Enable data collectors' checkbox under the 'Collectors' section.
- Check to enable full TASM functionality:** Points to the 'Enable this option' checkbox under the 'Enhanced TASM Functions' section.

## Adding an Advanced SQL Engine

- Next to Systems, select +, then select Add Teradata System.
- Under General System Details, enter a system nickname, up to 30 characters.
- [Optional] Select the Enable system check box to start collecting data.
- Enter the TDPID of the Advanced SQL Engine if not shown.
- [Optional] Enter the Site ID assigned by Teradata Customer Services.  
A site ID is required when using Elastic Performance on Demand.
- Enter a login name and password.  
Specify an account string and authentication, add more logons, test the logon settings, and grant access to an Advanced SQL Engine. Add any logins used for Viewpoint data collection to the system-level bypass. To define system-level bypass settings, use the Workload Designer portlet.
- [Optional] Click Test to verify that the login settings are correct. If you receive an error, verify that the login credentials are valid and the host can be reached.
- [Optional] Under Authentication Options, specify which authentication mechanisms are available in the portlets.
- [Optional] Under Character Set, select default character sets and enter a JDBC Flag value.
- Under Time Zone, select the time zone the system uses to log DBQL and ResUsage data.
- [Optional] Under Collectors, select Enable Data Collectors to begin collecting data. The data collectors can be enabled and configured individually in Data Collectors.
- [Optional] Under Enhanced TASM Functions, select Enable this option if your Teradata has license entitlement to TASM. The workload management features are available with your TASM license on SLES11 or higher.
- Select Apply.

# Monitored Systems Portlet – Data Collectors

Button:  (Viewpoint Administration) > Portlet: Monitored Systems   
Systems: System Name > Setup: Data Collectors > Data Collectors: Sessions



**Monitored Systems** Configure the systems and data collectors that provide data to portlets. Viewpoint DB Disk usage: 1% 12 hour change on Viewpoint DB: +0% Total Disk usage: 1%

Systems	Setup	Data Collectors	Sessions Collector Details
td17.10	<ul style="list-style-type: none"> <li>General</li> <li><b>Data Collectors</b></li> <li>System Health</li> <li>Canary Queries</li> <li>Alerts</li> <li>Monitor Rates</li> <li>Log Table Cleanup</li> <li>Cleanup Schedule</li> </ul>	<ul style="list-style-type: none"> <li>Account Info</li> <li>Alert Request</li> <li>AWT Info</li> <li>Database Space</li> <li>Dictionary</li> <li>Disk Space</li> <li>Elastic Limit</li> <li>Elastic Usage</li> <li>Lock Info</li> <li>Query Count</li> <li>Query Log</li> <li>Resource Usage</li> <li><b>Sessions</b></li> <li>Stats Manager</li> <li>System Config</li> <li>System Stats</li> <li>Table Space</li> <li>TASM Config</li> <li>TASM Distribution</li> <li>TASM Exception</li> <li>TASM State</li> <li>TASM Summary</li> <li>Virtual Storage</li> </ul>	<p><b>Sessions Collector Details</b></p> <p><input checked="" type="checkbox"/> Enable Sessions Collector</p> <p>Login: viewpoint</p> <p><b>Sample Rate</b></p> <p><input checked="" type="radio"/> Recommended sample rate (60 seconds)</p> <p><input type="radio"/> Custom sample rate: 6 seconds (6-3600) *</p> <p><b>SQL and Explain Thresholds</b></p> <p><input checked="" type="checkbox"/> Enable SQL and Explain</p> <p>Elapsed seconds &gt; * 10</p> <p>Total CPU seconds &gt; * 5</p> <p><b>New Priority for Lower Priority action</b></p> <p>SL</p> <p><b>Delete Data</b></p> <p><input type="checkbox"/> After 1 Year</p> <p><input checked="" type="checkbox"/> Over 10 Gigabytes</p>

APPLY RESET

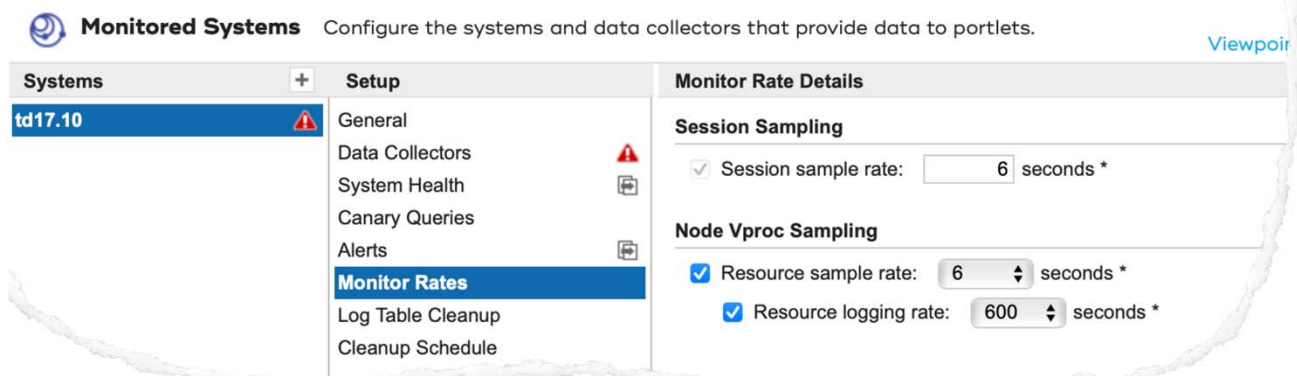
Data Collectors are used to monitor systems. After a system has been configured in Teradata Viewpoint, data collectors can be configured to monitor the system. Data collectors gather information from different sources and make the data available to Teradata Viewpoint portlets. Each data collector has a sample rate, or frequency, used to collect data from the system and a retention rate used to keep the collected data for a time period or up to a certain size.

Enable, disable, and configure data collectors to capture and retain portlet, disk usage, and resource data.

ResUsage tables must be enabled to

## Monitored Systems Portlet – Monitor Rates

Button:  (Viewpoint Administration) > Portlet: **Monitored Systems**   
 Systems: **System Name** > Setup: **Data Collectors** > Data Collectors: **Monitor Rates**



**Monitored Systems** Configure the systems and data collectors that provide data to portlets. Viewpoint

Systems	Setup	Monitor Rate Details
td17.10	General Data Collectors System Health Canary Queries Alerts <b>Monitor Rates</b> Log Table Cleanup Cleanup Schedule	<b>Session Sampling</b> <input checked="" type="checkbox"/> Session sample rate: 6 seconds * <b>Node Vproc Sampling</b> <input checked="" type="checkbox"/> Resource sample rate: 6 seconds * <input checked="" type="checkbox"/> Resource logging rate: 600 seconds *

ResUsage tables must be enabled using *Control GDO Editor (ctl)* utility or the database commands in *Database Window (DBW)*

### Monitor Rates

After a Teradata system is configured in Viewpoint, you can set internal sample rates for database sessions, node/vproc collection, and node/vproc logging. These rates determine how often PM/API data are refreshed. Note the following when setting the monitoring rate: The rates are based on the numbers set in the database.

The sample rates for node/vproc collection and node/vproc logging must be an integer divisor of 3600.

The logging rate is independent of the sample rate.


### Setting Monitor Rates

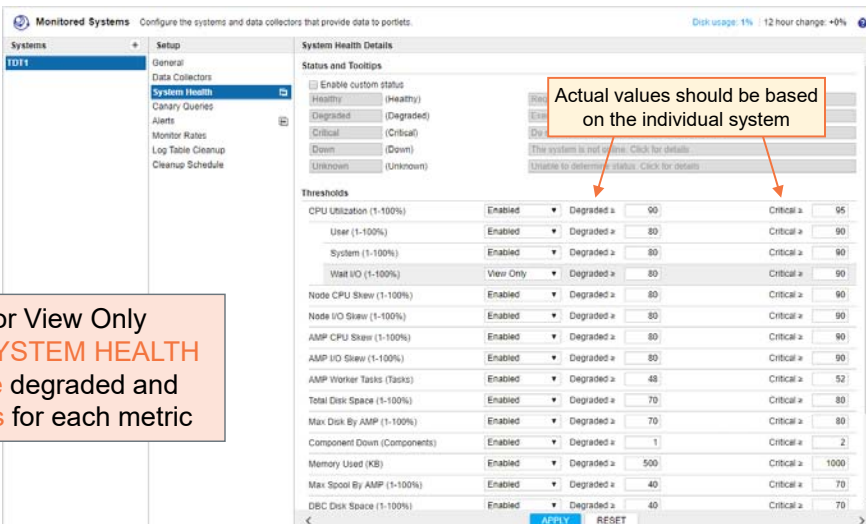
You can set internal sample rates for database sessions, system statistics, and resource usage. The sample rate you set becomes the recommended sample rate for each collector. Enabling the monitoring and logging process allows each collector to begin collecting data.

- From the Systems list, select the name of the system you want to update.
- From the Setup list, select Monitor Rates.
- Under Session Sampling, select the Session sample rate check box and enter the session sample rate in the box for the Sessions collector.
- Under Node Vproc Collection, select the Resource collection rate check box and select the collection rate for the System Stats collector.
- Under Node Vproc Logging, select the ResUsage logging rate check box and select the logging rate for the Resource Usage collector.
- Click Apply.



# Monitored Systems Portlet – System Health

Button:  (Viewpoint Administration) > Portlet: Monitored Systems   
Systems: System Name > Setup: System Health



Actual values should be based on the individual system

Enable, Disable or View Only metrics for the SYSTEM HEALTH portlet. Configure degraded and critical thresholds for each metric

You can customize system status and tooltips and configure metrics and thresholds. The thresholds are settings for the data collected by canary queries and the disk space, sessions, and system statistics data collectors.

For Vantage Advanced SQL Engines, the system status, tooltips, metrics, and thresholds appear in the System Health and Productivity portlets.

Enable metrics for the SYSTEM HEALTH portlet. Configure degraded and critical thresholds for each metric

- |           |  |
|-----------|--|
| Enabled   | Makes the metric visible in the System Health portlet. Uses the threshold values in the system status calculation.     |
| Disabled  | Omits the metric in the System Health portlet. Does not use threshold values in the system status calculation.         |
| View Only | Makes the metric visible in the System Health portlet. Does not use threshold values in the system status calculation. |

If CPU skew or I/O skew metrics are Enabled, do one of the following:

- Select the CPU skew or I/O skew qualification check box and specify a threshold.
- If the system CPU is under the defined threshold, the skew metric is excluded from the system health calculation. The metric displays in the System Health portlet as View Only.
- Clear the CPU skew or I/O skew qualification check box to include the skew metric in the system health calculation, regardless of system CPU.

## Current Topic – System and Query Monitoring Portlets

teradata.

- Viewpoint Portal Basics
- Administration Portlets
- **System and Query Monitoring Portlets**
  - **System Health**
  - **Metric Heatmap**
  - **Query Monitor**
  - **My Queries**
  - **Query Groups**
  - **Query Spotlight**



## System Health Portlet

Key performance indicator (KPI) display of system performance and state

Short periods of “Critical” are acceptable and desired.

Extended periods will be investigated by DBAs

Selecting an icon in the summary display, drills down to the detailed display



The System Health portlet monitors the status and key metrics of all monitored systems against defined thresholds. This portlet reports status with customized states or the following default states: healthy, degraded, critical, down, or unknown. The portlet also uses customized or default tooltips. These states and tooltips enable you to investigate metrics exceeding healthy thresholds.

The Teradata Viewpoint Administrator enables system health metrics, configures the degraded and critical thresholds, and customizes the states and tooltips for your system in the Monitored Systems portlet. The metrics and thresholds are usually selected to highlight an unusual load on the system that has the potential to impact overall performance.

If your system does not use the default states and tooltips, the Teradata Viewpoint Administrator customized the states and tooltips. You can customize the icon style using the Icons tab in the Settings view.

The System Health view displays a summary of states for one or more systems, using colored text and icons to represent system health. From this view, click anywhere on a status icon or text to investigate metrics for that system in the System Health details view.

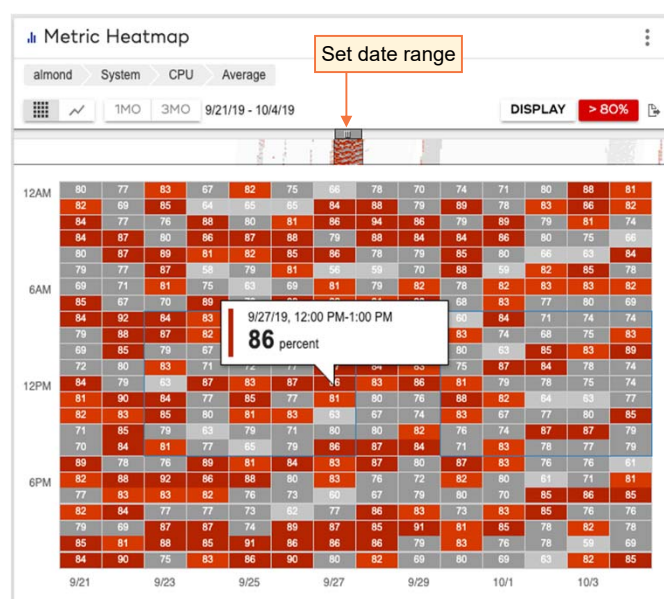
The System Health details view displays detailed statistics and information about each metric to evaluate the overall health of a system. If the Teradata Viewpoint Administrator defines metrics as View Only, these metrics are displayed with a gray background and are not used in determining system health. If any connection issue happens with the system because of a wrong configuration, then a relevant error message appears with the system status.

Note that “critical” means the system is running at 98-100% saturation. In some sites, this is exactly what they want so “Critical” might mean “great server utilization – we are getting our money’s worth”.

## Metric Heatmap Portlet

The Metric Heatmap portlet allows you to monitor single system metrics and resource usage trends within a specified time frame to help identify periods when a system is either over-utilized or under-utilized.

- System metrics over time
- Analyze hotspots
- Selectable metrics
- Helps find time slots to run resource-intensive jobs
- Adjustable duration and history
- Adjustable color thresholds



The Metric Heatmap portlet allows you to monitor single system metrics and resource usage trends within a specified time frame to help identify periods when a system is either over-utilized or under-utilized. You can use this information to determine when to schedule resource-intensive jobs with minimal impact to other users. Data displays in either a heatmap or graph view.

The Heatmap View displays a heatmap of variations and trends in system resource usage. The heatmap plots days on the horizontal axis and hours along the vertical axis. Customize the view using the menus, toolbar, and date-range slider to select metrics and date ranges.

The Toolbar contains the time frame list. Select a time frame from the list to change the graph. The time frame start and end dates are displayed on the toolbar next to the time frame buttons.

The Work Week and Shift Indicator highlights times when most users are normally accessing system resources with a box around the work week and work shift hours. Specify work day and shift hours using the Settings view.

The Cell contains data about an hour on a specific date in the heatmap. Hover over any cell to display an information balloon containing the date, hour, and metric value for the 1-hour block.  
Setting Cell Values and Shading in the Heatmap View.

You can display cell values and set metric thresholds for applying cell shading for each system monitored. The shading provides a visual overview by highlighting interesting values.

Values under the defined range appear in a very light shade of gray, while values after the range have a very dark shade. Values within the defined range appear in three shades of gray.  
Setting the Threshold Value in the Heatmap View

The threshold button displays the current threshold value. Use the threshold button to access the

threshold slider and adjust exception highlighting to show when the system is over-used or under-used. The threshold button and slider only appear when the metric displayed is a percentage.

## Query Monitoring Portlets

There are several portlets that can be used for monitoring queries. Each has a different purpose.

### Query Groups

**Specific Groupings of Users (single system)**

**My Queries**

**Viewpoint User's Queries (multiple systems)**

### Query Monitor

**All Queries or Sessions (single system)**

**Query Spotlight**

**ID Problem Queries (single system)**

The Query Monitor, Query Groups and My Queries portlet allows you to view information about queries running in Teradata Database. One can see the progress of submitted queries and target longer-running queries so you can optimize them. After you have identified a problem query, you can take action to correct the problem by changing the priority or workload, releasing the query, or aborting the query or session. You can take these actions for a single query or session, or multiple queries or sessions at a time. The session view provides a table listing the sessions, account strings, users, or utilities running on the database.

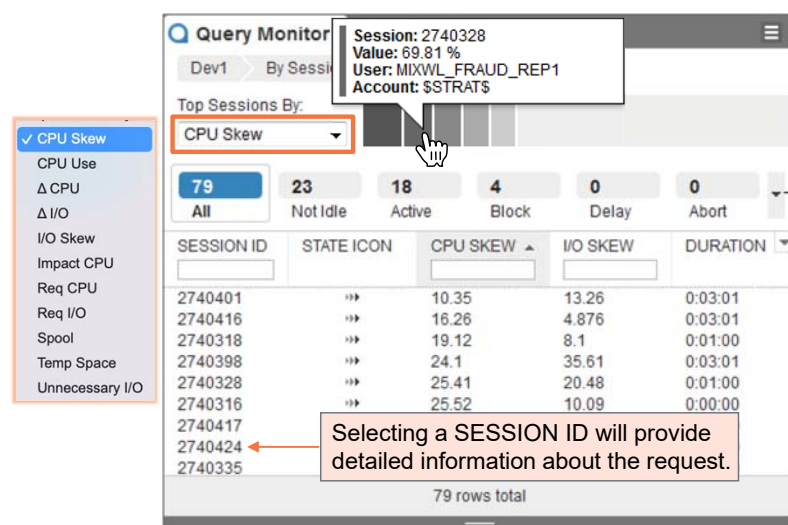
The Query Monitor portlet allows you to view information about queries running in Teradata Database and Teradata Aster systems as well as in Teradata Machine Learning Engine and Teradata QueryGrid so you can target problem queries.

The My Queries portlet allows you to view and manage your queries across multiple Teradata Database systems and in Teradata QueryGrid.

The Query Groups portlet allows you to view information about session queries running on a Teradata Database system or in Teradata QueryGrid that match defined criteria.

The Query Spotlight portlet allows you to view completed queries that exceeded the thresholds you set. You can view individual queries of interest as a line graph over time and against a variety of metrics. The Snapshot tab displays the query data at any point in time.

## Query Monitor Portlet – Summary View



The **Query Monitor** portlet allows you to view information about queries in the:

- Teradata Database
- Teradata QueryGrid

Provides reports grouped by:

- Account String
- Session
- User
- Utility
- VProc

The Query Monitor view shows a list of sessions running on a system and allows you to categorize the displayed sessions, such as by account string, session, user, and so on.

### Selection Menus

Enables you to select the system and report that defines the sessions displayed.

### Top Sessions Graph

Displays the sessions with the largest value compared with the total number of sessions currently running in a Teradata Database.

### State Filter Bar

Displays a count of the sessions in each state. Click any state in the bar to change the displayed data in the summary table to show only the sessions running a query in the selected state.

### Filters

Shows only rows that match your filter criteria. Summary Table Displays summary information about sessions, account strings, users, or utilities in columns. The current view is configured in the Configure Columns dialog box and Settings view. The view is refreshed when new data is collected. Click a row in the table to see details.

### Table Actions

For Teradata Database systems, table actions include:

- Clear Filters removes any content in the filter boxes.
- Configure Columns allows you to choose the columns to display and set thresholds.
- Abort, Change Priority, Change Workload, and Release Query allow you to manage the selected queries and sessions.
- Export creates a .csv file containing all available data. If filters are used, only filtered data is exported.



# Query Monitor Portlet – Session Details: Overview

Portlet: **Query Monitor** > Select SESSION ID > Tab: **Overview**

Query Monitor 10:55:25 AM

TDT1 By Session All

Session: 3747

OVERVIEW SQL EXPLAIN SKEW QUERY BAND

QUERY INFO	WORKLOAD INFO
State: Active Time in state: 0:02:00 Total duration: 0:04:00 Spool space: 49,773,457,000 Hot AMP spool: 2,702,163,970 Spool skew: 7,9007506 Temp space: 0 Request CPU: 2350.5 Request I/O: 1,986,571 PJI: 1.18 Unnecessary I/O: 0.85	Name: Adhoc Method: Timeshare CPU decay: Level 0 Classification mode: Auto Virtual partition: Standard I/O decay: Level 0
SNAPSHOT INFO	SESSION INFO
CPU use: 37.4% Impact CPU: 2430.56 Snapshot CPU: 9.4% Snapshot I/O: 21.6%	User: ADHOCUSER2 Account: ADHOC_&S Partition: DBC/SQL Requests: 2 Source: (TCP/IP) b327 172.32.0.50 TDAWS 15612 EC2-USER ADHOC_B TQ 01 LSS

Last sample period

The **Overview** tab provides detailed information about key metrics for the selected session and its queries.

The details view displays statistics and information about the selected session. This view can be accessed by clicking on a session row in the summary view.

When viewing a request, you can see detailed information from the following tabs:

- The Overview tab provides detailed information about key metrics for the selected session and its queries.
- The SQL tab displays the SQL for the selected query and allows you to export the SQL. If the SQL is from a stored procedure, the name of the stored procedure and the default database name at the time the SP was compiled displays. Otherwise, the default database name of the session at the start of the request displays. The tab is available only when a query is active, blocked, or delayed and meets thresholds established by the Teradata Viewpoint Administrator.
- The Explain tab displays an abbreviated version of the Step statistics and Explain text
- The Skew tab displays details about the level of skew in the query or session.
- The Query Band tab displays the profile, session, and transaction query bands for the selected query. This information is read-only. The tab is available only when a query band is included in the query.
- When a query is blocked, use the Blocked By tab to identify the other queries blocking the selected query.
- When a query is holding locks that are blocking other queries from completing, use the Blocking tab to gauge the impact of this blocking.
- The Defer tab displays details about the rules that are deferring a query.
- The Delay tab displays details about all rules that are delaying a query.



## Query Monitor Portlet – Session Details: Explain

Portlet: **Query Monitor** > Select SESSION ID > Tab: **Explain**

The screenshot shows the Query Monitor interface with the 'Explain' tab selected. It displays details for session 33032, including a list of steps. Step 23 is highlighted with a black number box, indicating it is completed. Step 24 is highlighted with a pulsating blue number box, indicating it is active. The interface shows estimated and actual times and rows for each step, along with a detailed description of the SQL operations being performed.

- The Explain tab displays an abbreviated version of the Step statistics and Explain text.
- The tab is available only when a query meets certain thresholds.
- Each Explain step is uniquely identified with a number.
- If a query uses incremental planning, appears with two gray bars next to the running steps.
- When all steps are generated, appears with *three* black bars.

The Explain tab displays an abbreviated version of the Step statistics and Explain text that result from an Explain request in an SQL session and allows you to export Explain data. The tab is available only when a query meets certain thresholds established by the Teradata Viewpoint Administrator. The information is read-only. Each Explain step is uniquely identified with a number, where the background color of the number box indicates status. If a query uses incremental planning, appears with two gray bars next to the running steps, indicating that the total number of steps can change as additional steps are generated. When all steps are generated, appears with *three* black bars next to the completed steps.

If you have the correct permissions, use to retrieve the latest Explain steps from the database.

Step Information	Description
Step Number	<ul style="list-style-type: none"> <li>• Completed steps are at the top of the list and indicated by a black number box.</li> <li>• Active steps are indicated by a pulsating number box (flashes blue).</li> <li>• Steps to run are at the bottom of the list and indicated by a white number box.</li> </ul>
Confidence Level Indicator Icon	No confidence in the estimate Low confidence in the estimate High confidence in the estimate High confidence in the estimate due to a join index
Estimated Time	Estimated execution time for the step
Estimated Rows	Estimated number of rows for the step
Actual Time	Actual CPU time consumed by the step, or blank if the step has not run
Actual Rows	Actual number of rows for the step, or blank if the step has not run

## My Queries Portlet

**My Queries** operates similarly to the **Query Monitor** portlet.

It allows a user to view their queries **across multiple systems**.

SESSION ID	SYSTEM	START	...	IN STATE	STATE
518799	Factory			2:31:00	Idle
518813	Factory			2:20:00	Idle
519091	Factory			0:16:00	Idle
519104	Factory Backup			0:12:00	Idle

**Profile** Configure settings for your Viewpoint user account and associate Teradata Database accounts.

Details

PERSONAL INFO **TERADATA ACCOUNTS**

Add an account to associate your Teradata database user with your Teradata Viewpoint profile. View queries for these data:

**ADD ACCOUNT**

SYSTEM	USERNAME	ACCOUNT STRING	AUTHENTICATION
ACCT	ADMIN		DEFAULT
PROD	BOB		DEFAULT
SKATE	DIANE		DEFAULT
TEST	JOE		DEFAULT

Systems are defined in the user's Viewpoint profile.

The My Queries portlet allows you to view and manage your queries across multiple Teradata Vantage Advanced SQL engines. You can see if queries are queued or blocked, and you can see their impact on system resources.

Use the My Queries portlet to view information about queries in either the summary view or the details view. The summary view contains a table with one row allocated to each of the sessions logged on under one or more user names. Select a row in the summary view to see additional session and query information in the details view. Use the SQL, Explain, Blocked By, or Query Band tab in the details view to display information for the selected session.

The PREFERENCES view allows you to select one or more systems, and then select one or more users per system to monitor. From this view, you can also select a format for the SQL that appears in the query details view.

## Query Groups Portlet

**Query Groups** operates similarly to the **Query Monitor** portlet.

It allows a user to view **groups of queries** belonging to groups defined by an admin using **Query Group Setup**.

The screenshot shows the 'Query Groups' portlet interface. At the top, there's a header with the title 'Query Groups', a timestamp '4:14:04 PM', and a menu icon. Below the header, there are tabs for 'Dev1' and 'TestGroup'. A summary bar shows counts for different states: 70 All, 35 Not Idle, 27 Active, 3 Block, 2 Delay, and 0 Abort. Below this is a table with columns: SESSION ID, STATE, ΔCPU, CPU SKEW, and DURATION. The table lists 10 rows of session data. At the bottom, it indicates '70 rows total'.

SESSION ID	STATE	ΔCPU	CPU SKEW	DURATION
2740331	ACTIVE	58.22	7.751	0:05:00
2740327	ACTIVE	55.71	25.4	0:01:00
2740322	ACTIVE	17.3	58.89	0:02:00
2740409	IDLE	14		0:00:00
2740317	ACTIVE	8.211	61.11	0:01:00
2740313	ACTIVE	3.531	62.58	0:05:00
2740335	ACTIVE	3.082	94.44	0:04:00
2740398	ACTIVE	2.781	61.11	0:01:00
2740417	ACTIVE	2.766	60.42	0:01:00
2740416	ACTIVE	2.424	64.81	0:03:00

The Query Groups portlet allows you to view information about session queries running on a Teradata Database system or in Teradata QueryGrid that match defined criteria.

A query group is any combination of one or more Teradata Database users, account strings, query bands, and workloads. For example, a query group that contains a group of users, could be configured for you to view queries based on the country of the user.

Queries appear in a group because they match certain criteria or characteristics. You can only view groups that have been assigned to you. If you identify a problem query, you can correct the problem by changing the priority or workload, releasing the query, or aborting the query or session. Take these actions for one query or session, or multiple queries or sessions at a time.

The summary view contains a table with one row allocated to each group running on the database. It contains all the information matching the criteria set for that group.

The details view contains session and query information.

The Query Groups portlet allows you to filter queries by group for all sessions. You can set thresholds for certain columns, and when the threshold is exceeded, the information is highlighted in the summary table.

The Settings view allows you to change display options.

You can monitor multiple Teradata Database systems by opening more than one instance of the Query Groups portlet.

# Query Spotlight Portlet

Query Spotlight almond

Last 1 month

2/9/2020 1:00 PM - 3/9/2020 1:00 PM

USERNAME	SESSION ID	REQ CPU	IMPACT CPU	REQ CPU S...	PJI	REQ I/O	UNNECESS...	REQ I/O SK...	DURATION	SPOOL
TOST_ADWI_DSS_99490		430K	505K	14.71	4.310	99.7M	0.232	14.72	3:25:18	12.5T
TOST_ADWI_DSS_113432		393K	489K	19.81	5.909	66.4M	0.199	21.22	2:58:25	8.09T
TOST_ADWI_DSS_99494		343K	479K	28.38	6.203	55.3M	0.191	29	3:21:17	6.81T
TOST_ADWI_DSS_122402		387K	478K	18.85	6.123	63.2M	0.193	20.01	3:10:21	7.83T
TOST_ADWI_DSS_108553		364K	470K	22.94	6.218	58.5M	0.191	22.54	3:10:19	7.25T
TOST_ADWI_DSS_127242		373K	457K	18.2	6.151	60.7M	0.193	19.42	2:51:49	7.56T
TOST_ADWI_DSS_104328		365K	454K	19.6	6.028	60.6M	0.195	20.33	2:54:16	7.47T
TOST_ADWI_DSS_136536		370K	451K	18.05	6.098	60.6M	0.194	18.55	3:05:15	7.46T
TOST_ADWI_DSS_131922		369K	451K	18.19	6.015	61.3M	0.199	18.13	3:18:19	7.47T
TOST_ADWI_DSS_140937		369K	448K	17.59	6.08	60.7M	0.194	19.09	2:59:21	7.56T
TOST_ADWI_DSS_117917		382K	447K	19.03	6.011	60.2M	0.195	19.46	3:01:19	7.47
TOST_ADWI_DSS_109549		321K	392K	11.27	4.314	74.4M	0.232	12.88	2:25:15	9.31T
TOST_ADWI_DSS_133183		269K	399K	12.4	4.25	83.1M	0.235	12.11	2:02:10	7.87T
TOST_ADWI_DSS_137902		252K	285K	11.59	4.281	58.9M	0.234	10.88	1:53:07	7.34T
TOST_ADWI_DSS_124239		221K	248K	10.86	4.238	52.1M	0.236	11.46	1:42:07	6.49T
TOST_ADWI_DSS_142573		213K	234K	8.945	4.157	51.3M	0.241	8.183	1:27:05	6.49T
TOST_ADWI_DSS_119555		210K	232K	9.404	4.227	48.7M	0.237	9.52	1:34:08	6.27
TOST_ADWI_DSS_129143		184K	205K	10.54	4.277	42.9M	0.234	9.907	1:24:21	5.34T
TOST_ADWI_DSS_106218		166K	179K	7.223	4.122	40.2M	0.243	7.67	1:13:03	4.99T
TOST_ADWI_DSS_115427		117K	127K	7.438	4.208	27.8M	0.238	7.187	0:53:05	3.47T
TOST_ADWI_DSS_113310		68.8K	109K	36.57	0.932	73.9M	1.073	35.98	2:27:14	9.28T
TOST_ADWI_DSS_104199		66.2K	103K	35.8	0.93	71.3M	1.076	35.08	2:33:14	8.92T
TOST_ADWI_DSS_127155		65.6K	98.5K	33.44	0.932	70.3M	1.072	32.12	2:35:30	8.81T
TOST_ADWI_DSS_117803		62.7K	95.3K	34.15	0.929	67.5M	1.077	32.03	2:40:16	8.48T
TOST_ADWI_DSS_140625		59.6K	89.5K	33.48	0.928	64.3M	1.08	34.07	2:38:15	8.05T
TOST_ADWI_DSS_122343		58.9K	85.9K	31.45	0.919	64.2M	1.091	31.83	2:42:15	8.05T
TOST_ADWI_DSS_136486		56.5K	84.3K	32.49	0.922	61.7M	1.084	30.51	2:42:15	7.85T
TOST_ADWI_DSS_131578		58.3K	82.5K	29.36	0.942	61.8M	1.091	28.3	2:51:15	7.78T
TOST_ADWI_DSS_108486		50.8K	74.1K	31.91	0.954	52.9M	1.048	32.89	2:48:18	6.67T
TOST_ADWI_DSS_99492		34.9K	72.8K	52.14	0.944	38.5M	1.09	54.96	2:31:11	4.8T
TOST_ADWI_OJAF_2839558		2.91K	67.4K	96.68	43.29	87.2K	0.023	30.33	0:13:07	2.93
TOST_ADWI_OJAF_2860811		2.98K	68.2K	96.63	42.37	86.8K	0.024	31.02	0:13:00	4.73

Page 1 of 30 (1933 rows total)

**Query Spotlight** displays all queries run within a set time period that has exceeded predefined thresholds

- Sorted by Impact CPU as a default
- By default, only shows queries that exceed the Standard Performance Thresholds

Users can filter this portlet to determine which queries require performance improvement and to judge the overall system impact

- The top impact queries should be investigated for tuning improvement
- Users with high impact queries should review explain plans for improvement areas

Time range can be extended and rewind can be used

The Query Spotlight portlet allows you to view completed queries that exceeded the thresholds you set. You can view individual queries of interest as a line graph over time and against a variety of metrics. The Snapshot tab displays the query data at any point in time.

Use Query Spotlight to:

- View general information about a query.
- Inspect a query as a line graph using the date-range slider and time marker.
- Set a date and time in the rewind toolbar. This feature is useful for rewinding data in surrounding portlets to a point of time specified within a query. The Query Spotlight portlet can set time and date but it does not respond to changes made in the rewind toolbar.
- The Settings view allows you to choose the query criteria that filter which metrics appear in the views.
- You can monitor multiple systems by opening additional instances of the Query Spotlight portlet. Rewind control is given over to the last instance of Query Spotlight to use the time marker.

## Summary

In this module, you learned how to:

- Describe the components of the Teradata Viewpoint
- Describe how to navigate portal pages and portlets
- Configure Viewpoint to support one or more Teradata systems
- Setup data collectors for system monitoring
- Use portlets to identify and resolve potential system resource issues
- Leverage query monitoring portlets to recognize and analyze problem queries



## Lab 1: Viewpoint Overview

For this exercise, you will log on to the Teradata Viewpoint portal and create a portal page.

1. Open a browser and enter the URL of the Teradata Viewpoint portal  
Log on to the portal using the username and password provided by the instructor
2. Create a new portal page and name it *Start\_Page*. Add the following portlets to this page: System Health, Query Monitor, Remote Console, and Space Usage
3. For the Query Monitor portlet, open "Settings" and add the "Top Session Graph". Open "Configure Columns" and add "Req CPU", "Spool" and "Impact CPU"
4. For the Space Usage portlet, apply a filter to only show databases with a "Current Perm %" over 10%  
How many databases were returned?  
Which one has the highest Max Perm?  
Which is the biggest table in this database?  
Check the "By Vproc" view to see how many AMPs are defined on this system and the Allocated Perm space per AMP.

The Query Monitor Settings view provides the following tabs that allow you to change display options and customize the way query and session information displays in the portlet.

### Display

For Teradata Database systems, choose whether or not to display the top sessions graph and whether to format the SQL to display in the SQL tab.

### Criteria

For Teradata Database systems, set the criteria values used to filter sessions in the My Criteria view. If no thresholds are specified, all sessions display in the view. After setting the criteria value and enabling a threshold, any query exceeding the criteria value appears in the sessions table in the My Criteria view.

### Skew

For Teradata Database systems, set the CPU skew threshold and CPU limit for sessions that are displayed in the By Vproc > By Skewed AMP view.



## Lab 2: Rewind

teradata.

In this lab, you will utilize Rewind to examine resource intensive queries.

**Dashboard:**

Open the Dashboard for the training system and switch on *Rewind*.

**Trends:**

Set the *Trends* time frame to 1 week.

Visually spot the time when the simulation was running and rewind to it.

Check the CPU, AWT and IO graphs.

**System Health:**

Check the *Degraded* or *Critical* metrics at this point in time.

**Queries:**

Look at the details for some of the Top 5 Queries and check the Overview, Explain, and SQL tab.

**Thank you.**

**teradata.**

©2023 Teradata





## Module 8: GSA, SZ, Restarts

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Objectives

After completing this module, you will be able to:

- Describe Global Space Accounting usage and benefits
- Explain how Secure Zones help with Multi-Tenancy
- Identify scenarios in which Unscheduled Restarts are possible
- Use tools available to emulate Production environments



## Why Global Space Accounting

- Database and user space limits are set at the AMP level to total space divided by the number of AMPs in the system.
- Space limits are hard limits and managed at the AMP level
- Transactions where usage exceeded the hard limit even on a single AMP, such as long-running load jobs, had to be resubmitted after increasing the space
- When source and target system configurations are different, then the number of AMP's configuration is different thereby space allocation per AMP is different

### Missing SLAs and wasted resources

The space accounting and space management infrastructure has been enhanced to manage the space limits at a global level (system level) and augment the AMP-level space management.

Global space accounting does “need based” space allocations to AMPs and maintains overall limit. Limits may no longer be at AMP-level.

- This is controlled by a new “skew” limit defined on space.
- A “soft limit” is defined to allow space usage to exceed system level max values.

The SQL CREATE USER/DATABASE command is extended to specify a SKEW limit percentage that allows maximum AMP space usage to be above the per-AMP quota. This can help load jobs complete without getting a per-AMP limit error.

The Skew factor and soft limits can be cumulative. For example, consider the following example:

- Database or global limit: 800 GB
- Soft limit: 10 percent (set with DBS Control)
- Per-AMP limit on a 4-AMP system: 200 GB
- Skew factor: 25 percent

With these numbers, the global limit can exceed the 800 GB limit by 10 percent (to 880 GB). In addition, an AMP can exceed the 200 GB limit by 25% (to 250 GB) if the total across all AMPs does not exceed 880 GB. Therefore, each of the AMPs can also receive an additional 20 GB because of the global soft limit. The cumulative effects can increase the maximum possible space allotment to an AMP to  $250 + 20 = 270$  GB, as long as the total space used across all AMPs does not exceed 880. An error message is given if either of the following limits are exceeded: the global limit of 880; the per-AMP limit of 270.

## Global Space Accounting Example

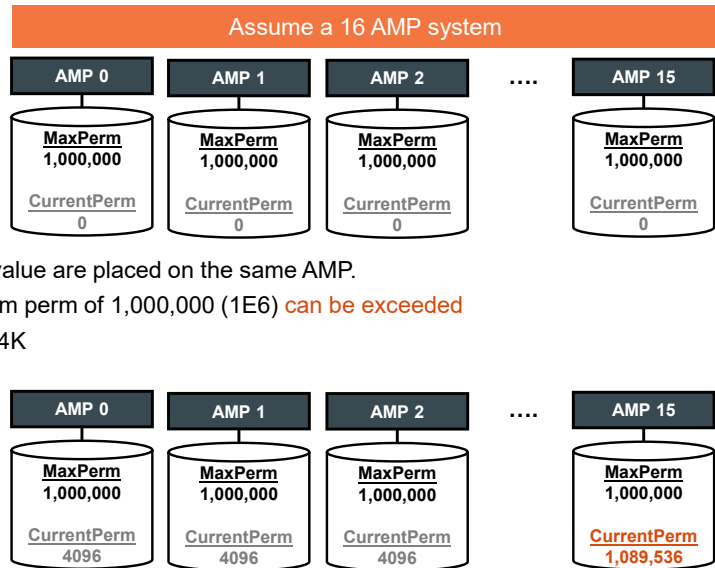
```
CREATE DATABASE DBNew AS PERM = 16E6;
```

### SKEW = 20 PERCENT

- The default is the 16e6 total space for a database is equally distributed per AMP
- Hence, the maximum perm is 1,000,000 (1E6)

A new table is created and records having the same PI value are placed on the same AMP.

- With a **SKEW = 20 PERCENT**, the per-AMP maximum perm of 1,000,000 (1E6) **can be exceeded**
- The other AMPs only have the table header which is 4K



The following example assumes a 4 AMP system:

```
CREATE DATABASE database2 AS PERM = 40000 SKEW = 10 PERCENT;
CREATE MULTISET TABLE database2.t1 (a INT, b INT);
SELECT AllocatedPerm, MaxPerm FROM DBC.GlobalDBSpaceV
WHERE Databasename = 'database2';
AllocatedPerm      MaxPerm
36000      40000

SELECT Vproc, CurrentPerm, AllocatedPerm, MaxPerm
FROM DBC.DiskSpaceV WHERE Databasename='database2' ORDER BY 1;
Vproc      CurrentPerm      AllocatedPerm      MaxPerm
0          4096      9000      10000
1          4096      9000      10000
2          4096      9000      10000
3          4096      9000      10000
```

Internal details: the system level space entry in DBC.GlobalDBSpace tries to maintain a reserve space in its AllocatedPerm column (the default is 10% as defined in DBS Control internal field #459. Initially the reserve amount was 4000 (40000 – 36000), but when an AMP crosses its AllocatedPerm, the reserve AllocatedPerm at global level drops to below 10% of its max limit. At this point, a de-allocation cycle is triggered in the background which collects all the unused AllocatedPerm from all the AMPs into global pool which decreases the AllocatedPerm at AMP level.

# Why Global Space Accounting

## Permanent Space Skew

- Ideally, data should be distributed equally amongst all AMPs, and they process equal amounts of data
- There could be asymmetry in data and certain values may have higher frequency compared to others
- In such cases the Teradata hashing algorithm responsible for data distribution on AMPs allocates varying data packets to each AMP
- This Data skewness results in asymmetric permanent space utilization on each AMP resulting in Permanent Space Skew

## Spool Space Skew

- To perform any operation between data rows they must be on the same AMP and to do that data is redistributed in a spool
- Redistributed data in the spool may be skewed making some AMPs work on more rows compared to others, causing processing skew
- Processing skew results in asymmetric spool space utilization on each AMP causing Spool Space Skew

## Temporary Space Skew

- Temporary space which is used for Global temporary tables could get skewed if data loaded in them is skewed resulting in Temporary Space Skew

# Global Space Accounting

## AMP Level

The per-AMP space quota is the maximum permissible AMP-level space (default)

To define explicitly for AMP soft limit we have a parameter named SKEW FACTOR

There are 3 DBS Control Parameter values which is at the AMP level

- Default<Space>SkewLimitPercent where Space is Perm/Spool/Temp
- This is used for users/databases for which the explicit skew limit clause is not specified or specified as DEFAULT

## Global Level

This maintains both an overall system limit and AMP limits

There is a DBS Control parameter for Global soft limit named GlobalSpaceSoftLimitPercent

- This is the percentage of space limit (for permanent, spool, and temporary) over the maximum value that the system permits for any database/user



- ❑ Skew limits cannot solve space problems, especially when actual use is close to the limits
- ❑ If there is not enough space, it is best to increase the space limits instead of allowing more skew

## Global Space Accounting Example

- A system soft limit can be established (DBS Control) as well as a SKEW factor at the database/user level
- These two limits (Global level and AMP level) can be cumulative

For example:

Database or global limit → 800 GB	Number of Amps = 4
Global Soft Limit → 10% [ DBS Control field:: GlobalSpaceSoftLimitPercent ]	AMP Level Skew factor → 25 percent

- Total Space = 800 GB 
- 4 AMP's =  $800/4 = 200$  GB 
- Global Space Limit =  $800 \text{ GB} * 10\% = 80\text{GB}$ , hence total 880 GB
- Skew Factor per AMPs = 25%
  - $200 \text{ GB} * 25\% = 50\text{GB}$ , hence a total of 250 GB
  - $80 \text{ GB} / 4 = 20\text{GB}$  hence max possible space allocation  $250 + 20 = 270 \text{ GB}$

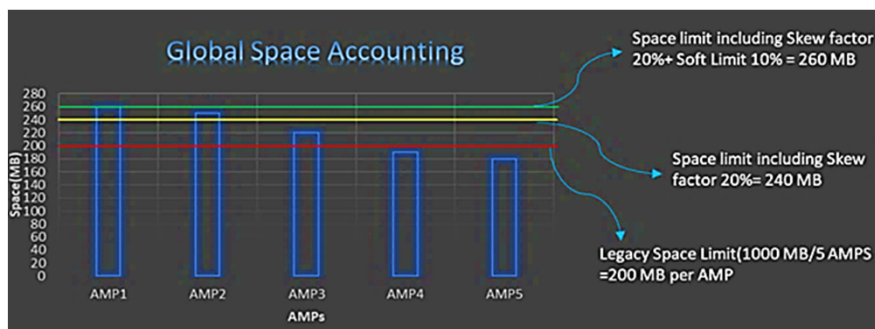
The user will receive an error message if either of the following limits is exceeded:

- Global Space limit of 880 GB
- Per-AMP limit of 270 GB

# Global Space Accounting Monitoring

Is it possible to monitor Global Space Accounting?

- There are no direct Viewpoint alerts related to Global space accounting
- The soft limits would need to be converted to normal space alerts
  - Soft limit alert messages can be seen from /var/log/messages
  - If your soft limit is 80%, set the alert within Viewpoint to 70-75% to warn you that you are getting close
- INFO: Teradata: 2900 # Alert code 9780: The Allocated PERM space for the database (0 411) exceeds the global soft limit



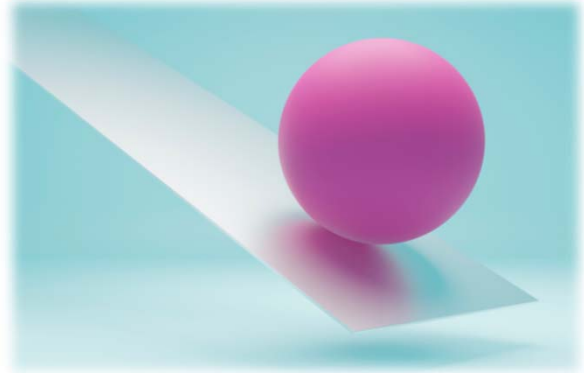
Global Space Accounting Visualization (academic)



## Global Space Accounting Dictionary Tables

PermSkew, SpoolSkew, and TempSkew additional columns

- These limit columns capture the permissible skew limit for the various spaces
- Added to the following Dictionary tables
- DBC.GlobalDBSpaceV[X]
- DBC.DiskSpaceV[X]
- DBC.AllSpaceV[X]



## Why Teradata Secure Zones

Are you thinking of the following requirements/needs:

- Regulatory requirements to protect your data hence thinking about a separate database
- Build a separate database, implement redundant infrastructure, and then manage data and users

**Teradata Secure Zones** provides you

- Virtual Database environment
- Isolation of users and administrators from each other as if they were running on physically segregated databases
- Users within a virtual database have no access or visibility to objects within other databases
- Data within a virtual database is secure from being accessed by the database or zone administrator
- Shared Infrastructure without creating a separate database for each regulatory requirement
- Co-hosting otherwise separated applications and leveraging a larger resource pool

### Secure Zones Elements:

**Zone** - 'zone' is a secure area which restricts unauthorized outside users from accessing data and objects within the zone. A zone can be created by a database administrator who has the 'Create Zone' privilege.

**Zone Creator** – a Zone creator creates a zone and assigns a user or a database as the zone root. Zone creator will be prevented from accessing the objects or data within the zone he creates.

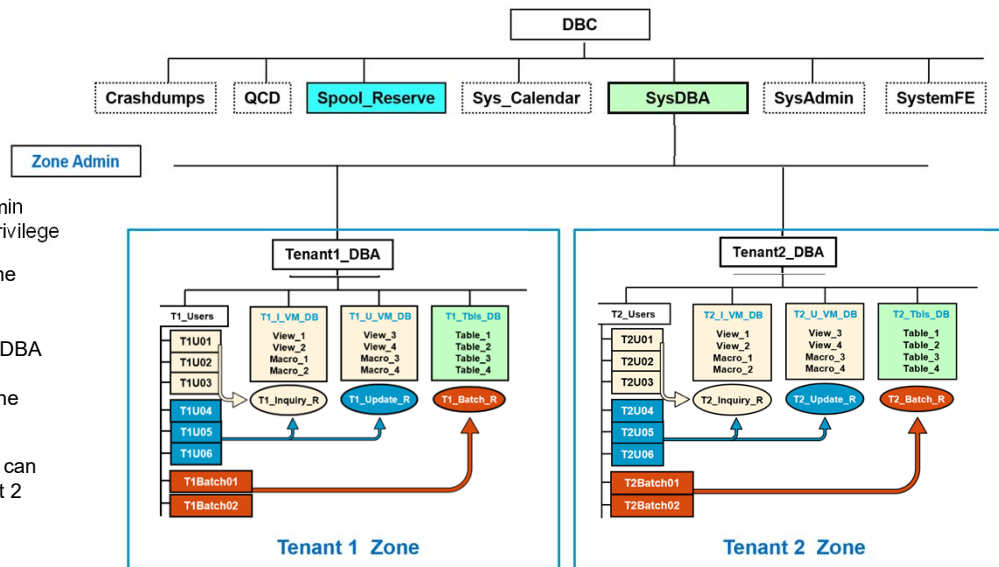
Rules concerning a zone creator are as follows:

- Only the zone creator can add a root and primary DBA to the zone.
- Only the zone creator can delink the root and primary DBA from the zone.
- Zone creator may grant zone access to users or roles outside the zone. Note: Zone grants do not allow grantees to access database objects within the zone. The zone's primary DBA must also grant Discretionary Access Control (DAC) privileges to zone guests before access is permitted.
- Zone creator is also responsible for revoking zone access to a zone.
- Zone creator must be a user having 'Create Zone' and 'Drop Zone' privileges and additionally have all privileges on the user which will become a root of the zone.
- Zone creator cannot be dropped until the zone is dropped.

**Zone Root** - Zone root is a database or a user on which a zone is created

# Teradata Secure Zones

- Create a dedicated ZoneAdmin user with the Create Zone Privilege
- Create Two Zones using Zone Admin user
- Create each Zone's primary DBA
- Each Zone DBA will create the entire database hierarchy
- No user from Tenant 1 Zone can access any data from Tenant 2 Zone



## Teradata Secure Zones – Observations

When the query against Data Dictionary tables is sent to the Teradata system

- The SQL is effectively changed to view DBC.TablesV\_SZ (and not DBC.TablesV as coded)
- It is this view that uses the new table DBC.Zones to filter the returned data
- This change only happens for non-DBC users and if accessing views
- If user DBC runs the above query, then the SQL is not modified, and rows are returned from all zones

### Query Logging

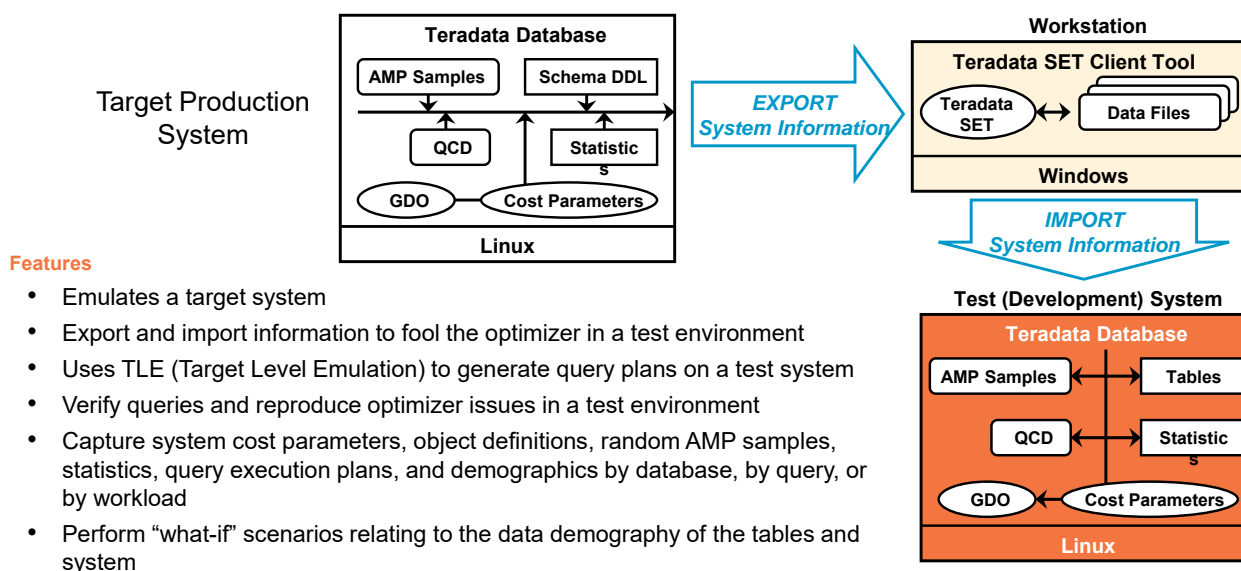
- These rules are unique to the zone of the user (assuming it is not DBC) executing the BEGIN / REPLACE QUERY LOGGING command

### Workload Management

- All zones are on the same physical Teradata platform, they are all subject to the same set of rules



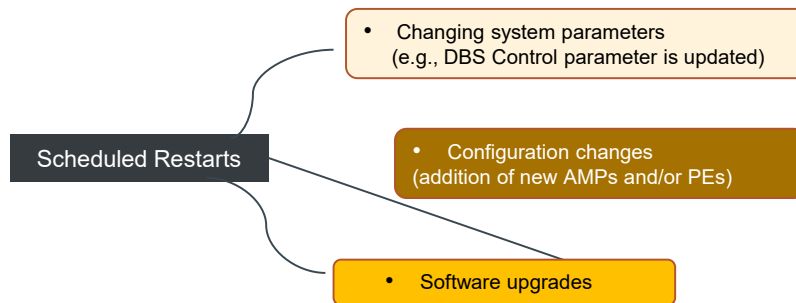
## Teradata System Emulation Tool (SET)



The Teradata System Emulation Tool simplifies the task of emulating a target system by providing the ability to export and import all of the information necessary to fake out the optimizer in a test environment. This information can be used along with Teradata's Target Level Emulation feature to generate query plans on the test system as if they were run on the target system. This feature is useful for verifying queries and reproducing optimizer related issues in a test environment.

Teradata SET allows the user to capture system cost parameters, object definitions, random AMP samples, statistics, query execution plans and demographics by database, by query or by workload. This tool does not export user data. Upon import the user can customize or edit object definitions, random AMP samples, statistics and cost parameters. The Customize feature allows the user to perform "what-if" scenarios relating to the data demography of the tables and system performance parameters. Teradata SET also has an option to log SQL statements. The user can view the log directly from the Teradata SET window to troubleshoot any failures that occur during export or import operations.

## Scheduled Restarts



### Restart Processes

- Spool cylinders are returned to the free cylinder list (unused cylinder pool)
- Before logons are enabled, uncommitted work is rolled back
  - 1<sup>st</sup> Tables are re-locked for background recovery
  - 2<sup>nd</sup> Logons are enabled in cold start

There are two types of restarts on a Teradata Database:

- Scheduled restarts
- Unscheduled restarts

### **Scheduled Restart/User-initiated Restart**

In a scheduled or user-initiated restart, use the RESTART command from either the DBW Supervisor window or from vprocmanager to restart the system.

### **Unscheduled Restart/Automatic Restart**

In an unscheduled restart or automatic restart, the system reboots without user input.

This slide provides examples of when you might need to perform scheduled restarts, and under what conditions you might encounter unscheduled restarts.

## Scheduled Restarts

Restart Teradata with	Use this command	Options
Command-line	<code>tpareset &lt;comment&gt;</code>	<code>-f, -x, -y, -d, -l, -Q, -P</code>
Vprocmanager	<code>restart</code>	<code>cold, coldwait</code>
DB Console - Supervisor	<code>restart tpa &lt;comment&gt;</code>	<code>cold, coldwait</code>

Example:

- View any restarts that have occurred in the last 90 days
  - `EXEC ALLRESTARTS (DATE - 90,);`
- The “tpatrace” command may also be used to see information about restarts
  - `tpatrace 3` (shows last three restarts)
- Restart Teradata
  - `tpareset -f` < Change of system parameters >

This slide shows the windows and utilities from which you can restart the Teradata Database system, the necessary commands and available restart options.

### -f option

The -f option, used with the tpareset command, forces all TPA nodes to participate in the tpareset regardless of their current state, without rebooting Linux.

### -x option

The -x option allows you to shut down the Teradata Database on the entire system without shutting down the operating system. This option does not automatically restart Teradata.

### -y option

The -y option automatically answers yes to the confirmation prompt.

### -d option

Specifies that a DBS dump be taken before doing the restart.

### -l delay option

Specifies the delay interval in seconds to wait for other nodes to join the TPA configuration. This parameter controls how long a node will wait during the BYNET configuration phase of PDE initialization for other nodes to reach that point before continuing on without them.

Status VCOLD full restart, but transaction recovery will be deferred.

Status COLDWAIT full restart, but database startup will be held up until transaction recovery is complete.

## Unscheduled Restarts: Drive Failures

Scenario 1	<p><b>Failure:</b> One drive in a drive group (assume RAID 1 or RAID 6)</p> <p><b>Result:</b> No TPA reset</p> <p><b>Resolution:</b> Replace drive – Array Controllers automatically rebuild the drive</p>
Scenario 2	<p><b>Failure:</b> Two drives in a RAID 1 drive group or three drives in a RAID 6 drive group</p> <p><b>Result:</b></p> <ul style="list-style-type: none"> <li>– TPA reset (1-5 minutes)</li> <li>– AMP taken offline and marked as Fatal</li> <li>– Fallback tables OK</li> <li>– Non-fallback tables partially available</li> </ul> <p><b>Resolution:</b></p> <ul style="list-style-type: none"> <li>– Replace the two drives</li> <li>– Reformat LUNs or Volumes in the drive group</li> <li>– Perform a table rebuild</li> <li>– Restore non-fallback tables</li> </ul>
Scenario 3	<p><b>Failure:</b> Two drives in two different drive groups (RAID 1) associated with AMPs in the same cluster – 2 AMPs fail in a cluster</p> <p><b>Result:</b> Machine halts</p> <p><b>Resolution:</b> Restore User DBC and tables</p>

### Drive Failure

When a drive fails, there may be a loss of data. Tables with fallback protection continue to be 100% available. Tables without fallback protection will only be partially available.

An AMP will attempt 5 retries to a disk array before determining that it cannot access the array or its associated Vdisk.



## Unscheduled Restarts: BYNET Failures

### Scenario 1

**Failure:** One BYNET fails

**Result:**

- No TPA reset
- All traffic auto-switched to remaining BYNET
- Impact on system performance

**Resolution:** Repair BYNET

### Scenario 2

**Failure:** Both BYNETs fail

**Result:** Teradata halts and is not available

**Resolution:** Repair BYNETs

### **BYNET Failure**

If a BYNET fails, processing will resume on the other BYNET. Performance will be impacted.

## Unscheduled Restarts: Node, VPROC s/w Failures

### Node Failure

**Failure:** Node Fails (e.g., Linux hangs, 2 power supplies fail, memory fails, etc.)

**Result:**

- TPA restart (1 - 5 minutes) and vprocs migrate to other nodes in clique
- Possible Linux reboot (3 - 15 minutes)

**Resolution:**

- Repair node and reboot operating system
- Restart Teradata to allow node to rejoin Teradata configuration

### VPROC s/w Failure

**Failure:** AMP or PE Vproc fails

**Result:** TPA restart (1 - 5 minutes) and vprocs may be marked offline

**Resolution:** If necessary, run Scandisk, Checktable, and Rebuild utilities

### Node Failure

This slide describes a node failure.

### VPROC Failure

- PE VPROC** This type of failure produces very little impact on system performance. It reduces the maximum number of sessions that can be active at one time, however, and logons may take longer when a PE vproc is down.
- AMP VPROC** If a single vproc fails in one or more clusters, the system can continue servicing users with the other AMP vprocs. However, the performance level drops causing a slow down in performance and response time.

If two or more AMP vprocs fail in a single cluster, it halts the database system. All processing stops until the administrator brings at least one of the down vprocs back on-line.

### SWS Failure

This slide describes an SWS failure.

## Summary

In this module, you learned:

- Describe Global Space Accounting usage and benefits
- Explain how Secure Zones help with Multi-Tenancy
- Identify scenarios in which Unscheduled Restarts are possible
- Use tools available to emulate Production environments



## Appendix A: Class Instructions

Vantage Administration Intermediate

Copyright © 2007–2023 by Teradata. All Rights Reserved.

## Course Materials

The primary course material for this 5-day VILT (Virtual Instructor Led Training) course is:

- Electronic copy (PDF file) of the Student Manual

The course materials that are provided on a USB flash drive are listed on this slide. These materials are provided at the beginning of the class.

The Student Manual and the Lab Workbook have been created as PDF files which can be viewed using Adobe® Reader®.

These PDF files were created using Adobe Acrobat® and commenting has been enabled for both files. This allows you to use Adobe® Reader® Comment and Markup tools to place your own notes and comments within the files.

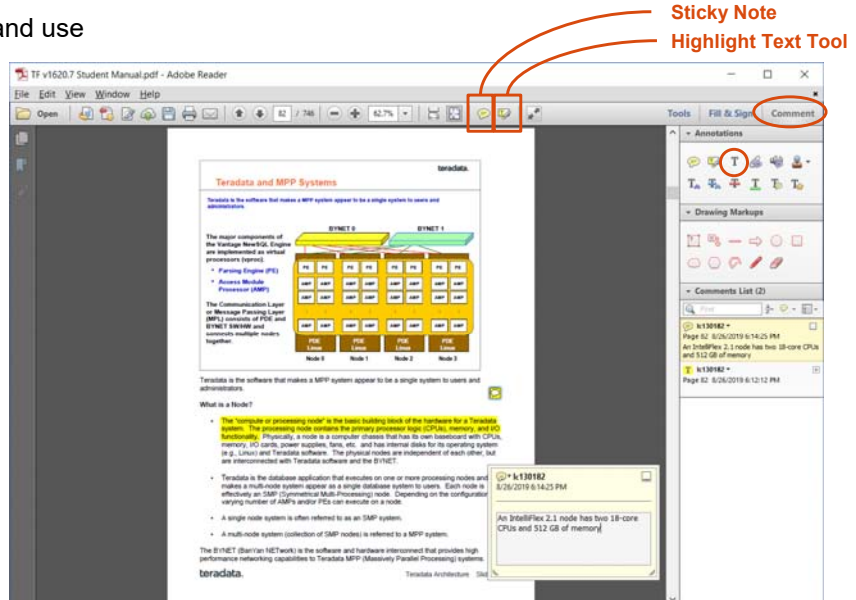
### View and search a PDF

In the Adobe Reader toolbar, use the Zoom tools and the Magnification menu to enlarge or reduce the page. Use the options on the View menu to change the page display. There are various options in the Tools menu to provide you with more ways to adjust the page for better viewing (Tools > Select & Zoom).

## PDF Comment and Markup Tools

You can add comments, highlighting, and use other markup tools. Examples are:

- Sticky Note
- Highlight Text Tool
- Add Text



These course materials have "commenting" enabled. Therefore, you can make comments in these files using the commenting and markup tools. Of the many commenting and markup tools that are available, you may find it easier to use the following tools (highlighted on this slide).

- Add Sticky Note
- Highlight Text Tool
- Add Text

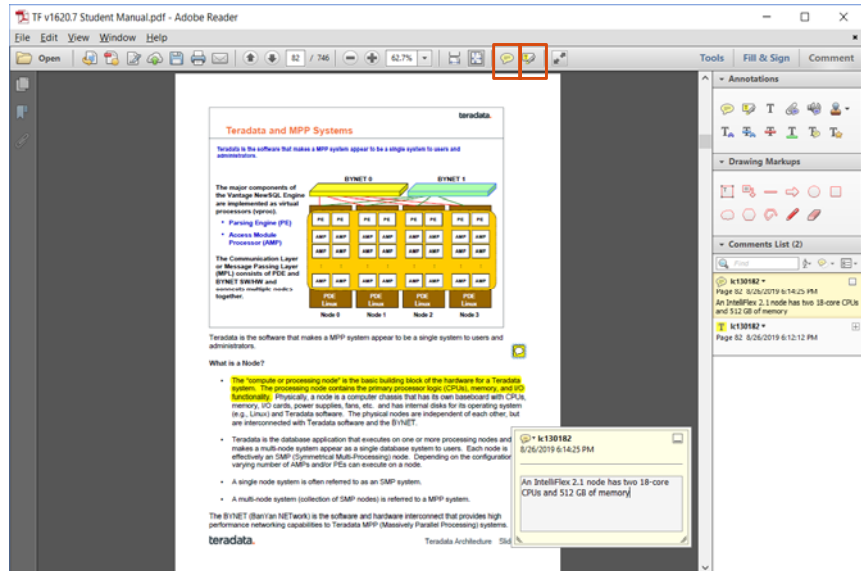
Comments can include both notes and drawings (if you have the time during class). You can enter a text message using the Sticky Note tool. You can use a drawing tool to add a line, circle, or other shape and then type a note in the associated pop-up note.

You can enable the Comment & Markup Toolbar or you can simply select the tools using the pull-down menus

After you add a note or comment, it stays selected until you click elsewhere on the page. A selected comment is highlighted by a blue halo to help you find the markup on the page.

## Example of Highlighter and Sticky Note Tools

After including highlights and comments in your PDF file, a recommendation is to use the Save AS option and save the updated version with a new name – maybe include your initials.



This slide illustrates an example of using the Highlighter and Sticky Note tools.

Select a commenting or markup tool.

Choose Tools > Comment & Markup > Highlighter or Sticky Note (or another tool)

Note: After you make an initial comment, the tool changes back to the Select tool so that you can move, resize, or edit your comment. (The Pencil, Highlight Text, and Line tools stay selected.)

To keep a commenting tool selected so you can add multiple comments without reselecting the tool, do the following:

Select the tool you want to use (but don't use it yet).

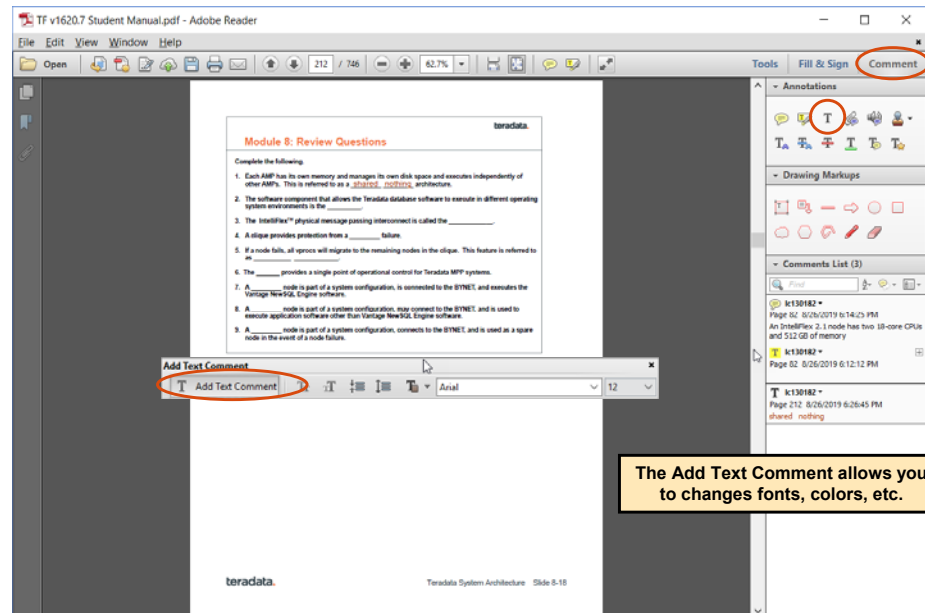
Choose View > Toolbars > Properties Bar.

Select Keep Tool Selected.

You can change the font of a text in a sticky note. Open the sticky note, choose View > Toolbars > Properties Bar, select the text in a note, and then change the font size in the Properties Bar.

## Example of Add Text Tool

The Add Text Comment tool can be used to add text at any location in the PDF file.

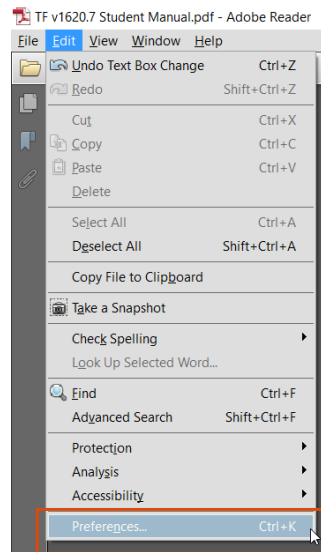


The Add Text Toolbar may be useful when completing review questions as shown on this slide. You already have the answer to one of hundreds of questions in this course.

After making notes and comments, save your changes. You may want to save your changes to a different PDF file name in order to preserve the original PDF file.

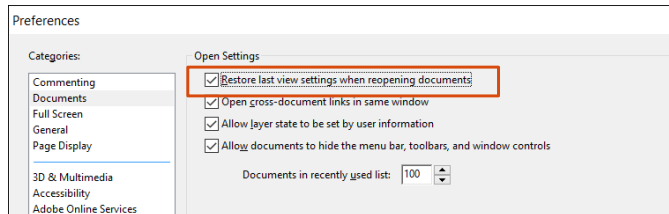


## Open PDF File at Last Page Viewed



You may want to set your preferences so you can open your PDF documents at the last page that was viewed before closing the document.

**Edit > Preferences > Documents**



If you are reading a 1500+ page PDF manual and you get to page 500, you may want to start on page 500 the next time you open the PDF file. The default is to return you to the beginning of the PDF file. There are multiple options that can be used to open your PDF file at the last page you viewed.

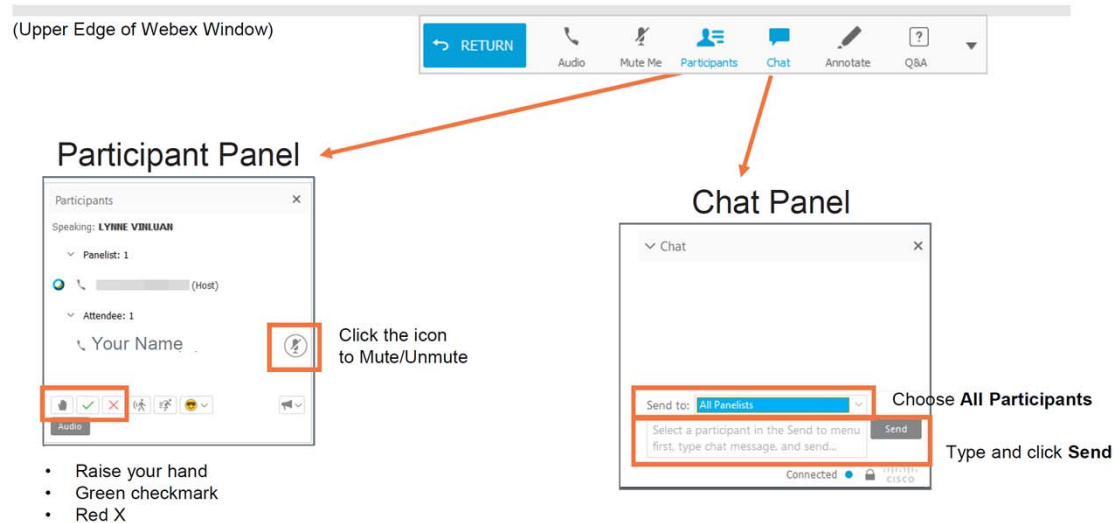
One option is to set a bookmark and select the bookmark when you reopen the file.

Another option is set your preferences so you open every PDF document at the last page that was viewed before closing it.

The next time you reopen your PDF file, you will be able to continue reading from exactly where you were when you closed the PDF document.

This setting applies to all of the PDF files you that you open.

# Conferencing Tool Controls



Let's take a few moments to get familiar with the features of the conferencing tool we will be using today. To see your Control Panel, move your cursor to the top of your Webex Training Center screen, and it will pull down.

The Participants icon is a toggle. When it is highlighted in blue, you can see the names of everyone in this session somewhere on your screen. If you have two monitors, it may be on your other monitor. Point your cursor to your name in the Participants panel, and you will see a red or gray microphone next to it. This is another toggle, where you can mute and unmute yourself. Red is muted, click again to make it gray to unmute and speak up.

At the bottom of the Participants panel, below everyone's name, is a set of icons that your instructor may be asking you to use. You can raise and lower your hand, you can give it a try now. During the labs, we will be asking you to click either the green checkmark if you are done with that lab, or the red X if you need more time. This is helpful to us so that we know if we can move ahead to the next section.

Some of us have been using the Chat panel already. When you have a question, use the drop-down box by the Send To: Prompt and choose "All Participants" so that our instructor does not have to repeat the question. Then type your message in the box below it and click Send.

## Audio or Video Issues?



If you lose **Audio**, click the Audio icon to reconnect.  
You do not have to leave the Webex session.

If your screen "freezes," send a chat to your Host.

Depending on your internet connection, the state of Webex Training Center, or other factors, we have found that some attendees have trouble with audio. If that is the case, click on the Audio icon on the Control Panel to reconnect your audio.

This lets you stay connected in Webex and not have to leave the session completely to get back in.

On rare occasions, your screen may seem to freeze up and get out of sync. If that is the case, send a chat message to me, your Host. We can address it then.

## Virtual Classroom Ground Rules

- Join Class on time. Attendance will be taken (after 15 minutes you would be considered No Show)
- For Course credit you must attend the entire class and perform labs where applicable
- Mute your line (WebEx OR on your phone) while listening
- Do not put this call On Hold
- To get the most out of this class, participate in activities and labs



Now to make this the best possible learning experience for everyone, here are a few ground rules for this session.

- Attendance is taken, after 15 minutes you will be considered a no-show.
- To get course credit, you need to attend the entire class.
- Quick reminder to ensure your line is muted when you are listening to others and unmute to contribute a question or comment.
- Also, do not put this call on hold, to avoid having elevator music that some providers use.
- If you already have a second monitor set up, you may find it helpful in this training, especially when doing the labs.
- And most of all, please participate in the activities and labs to get the most out of your time in this class.