

The Teradata Database

Part 2: Teradata Fundamentals

Welcome to the second in the series of courses designed to provide you with an introduction to the Teradata Database.

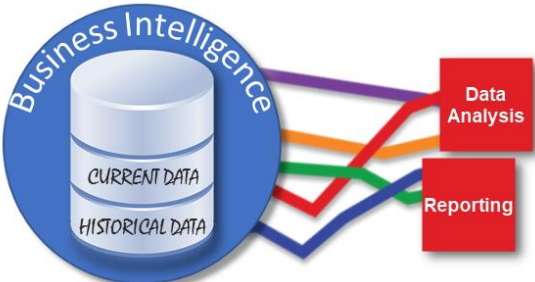
The first course, Database Basics, served as a refresher, or provided a foundation, to those who are new to the world of Databases. This course, Teradata Fundamentals builds on that foundation and transitions to exploring the Teradata Database.

By the end of this course you should be able to:

- Identify components of the Teradata Database architecture.
- Identify the differences between Primary Index and Primary Key.
- Identify the types, levels, and functionality of locking provided by Teradata.
- Recognize the functions and types of authentication in the Teradata Database.
- Identify tools and utilities used for data integration.

Module 1 - Teradata Database Architecture and Components

The Teradata Data Warehouse



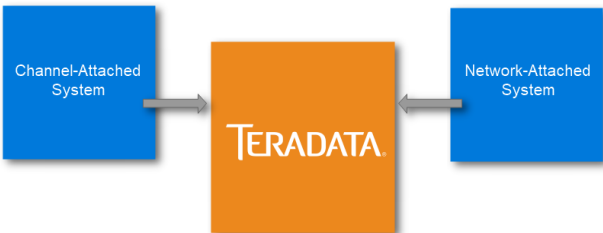
The Teradata Data Warehouse stores current, as well as historical data in one location. This enables effective data analysis and reporting. Because it is a central repository of integrated data, it is considered a core component of business intelligence.



The Teradata Database is designed to support high-performance, diverse queries as well as in-database analytics, and has sophisticated workload management so the data warehouse can adjust to the changing needs of the business. In addition, it supports, and enables all Teradata Data Warehouse solutions.

Unlike other relational databases, Teradata’s built in parallelism enables faster processing of queries.

How You Can Connect to Teradata



The client application submits an SQL request to the Teradata Database, receives the response, and submits the response to the user.

The Channel-attached system connects via a mainframe, where the Network-attached system connects via pc, tablet, phone, etc.

How is the Teradata Database used?

So how is the Teradata database used? The answer to this question is “endless ways”. Our Business Analytics Solutions take a “business-led, technology enabled” approach to problem solving and facilitating business growth to give organizations the insights needed to be successful and boost profitability.

Using the Teradata Database companies can tackle any business problem, achieving high-impact business outcomes by leveraging data and analytics to accelerate their time-to-value.

Some other examples of ways companies use the Teradata Database are:

- streamline operations,
- improve processes,
- manage marketing and promotions,
- prepare shipping labels,
- manage employee records, and
- track production, inventory and distribution.

Common Types of Databases

When we think of analytics we typically associate this with a function performed in the corporate business world, but analytics are on the rise everywhere.



For instance, I’m sure some of you watch basketball, well did you know that analytics has become a driving force in college basketball? The data is analyzed when making decisions about player recruitment, team building, game planning and strategy as well as many other things.



And how about social media, do you have a Twitter, Facebook, Snapchat or other social media account? All of these are highly dependent on big data analytics.

So, you can see that the Teradata Database can be used in many ways, some of which are obvious and some are more subtle and “behind the scenes”.

Features and Benefits of the Teradata Database

The features and benefits that you will see below are only a few of the many that make Teradata stand apart from the competition.

Single Version of Business

A design goal of the Teradata database was to provide a single data store for a variety of client architectures. This approach to data storage is known as the single version of the business and provides a single data store with a single copy of the corporation’s data. This is implemented through heterogeneous client access.

High Availability

With its High Availability, there is no single point of failure. The Teradata database has the ability of making data available despite errors or failures because both hardware and software provide fault tolerance, some of which is mandatory and some of which is optional. For example, fallback tables are duplicate copies of primary tables. Each fallback row in a fallback table is stored on an amp different from the one to which the primary row hashes. This storage technique maintains availability should the system lose an amp and its associated disk storage in a cluster. In that event, the system would access data in the fallback rows.

Highly Scalable

It is highly scalable, which makes it easier for customers to expand the system as per the growth and need of the business.

Teradata Everywhere

Teradata Everywhere allows business to migrate easily across different deployment options and support hybrid architectures. The same Teradata software is available across all deployment options, enabling seamless application portability and scalability. These options include Teradata Cloud, Public Cloud, Teradata hardware, and Commodity hardware and any hybrid combination.

Unlimited Parallelism

Parallelism is at the very heart of the Teradata Database and is built into the Teradata Database from the ground up! There is virtually no part of the system where parallelism has not been built in. Unlimited Parallelism means that every request is broken down into components those components are all worked on at the same time (parallel).

Objects Within the Teradata Database



Let's review some of the Objects within the Teradata Database. When we say object, we mean things like tables, views, macros, triggers, stored procedures, or user-defined functions. Database objects are created and accessed using standard Structured Query Language or SQL.

Before we do that though, you will need to understand the concept of partitioning. Partitioning is a database process where very large tables are divided into multiple smaller parts. By splitting a large table into smaller, individual tables, queries that access only a fraction of the data can run faster because there is less data to scan.

As you might recall if you took the first course in this series, tables are made up of rows and columns.

To continue to expand on this information, and learn about more objects in the Teradata database, you should know about some additional terms.

Fallback

The concept of fallback is about protecting the customers' data against AMP failure. The data is protected by storing a second copy of each row of a table on an alternate AMP called the Fallback AMP.

Data Types

Data types specify the kind of values that will be stored in the column. Each column in a table is associated with a data type which is assigned when the table is added.

Partitioned Table

A column, or vertical, partitioned table is a physical design implementation option that allows sets of columns or a single column of a table to be stored in separate partitions. This can improve performance for some types of queries and also allows reduction on the I/O.

Set Table

Teradata does not allow duplicate rows in a table. Set Tables are the default in the Teradata Database. Multiset tables, however, allow duplicate rows.

View

For any database, there are a number of possible views that may be specified. Often referred to as a virtual table, the view doesn't actually store information itself, but just displays it from one or more existing or underlying tables. Views are used to control access to the underlying tables and simplify access to data. Some columns in a view do not exist in the underlying table. For example, it is possible to present calculations or data summaries that you cannot obtain from the table.

Macro

Macros reduce the number of keystrokes needed to perform a complex task. A familiar example is one or more sequel statements that can be executed at a single request.

Trigger

A trigger defines events that happens when an event occurs, such as a set of Structured Query Language (SQL) statements automatically "firing off" an action when an operation modifies a specified column or columns.

Stored Procedure

Stored procedures are a combination of pre-defined procedural statements, control declarations and sql statements that is a program that are stored in, and execute within, the Teradata Database.

Secondary Index

A Secondary Index allows optional ways for the system to access the rows of a table. It is unlike a Primary Index in that it has no influence on the way rows are distributed across the AMPs.

Join Index

A Join Index is defined to enable join queries to be resolved without accessing or joining the actual tables. They can be beneficial when queries frequently request a particular join.

User

A user can be thought of as a collection of tables, views, macros, triggers, stored procedures, join indexes, and access rights.

Object Management

In order to manage the data base objects, teradata needs to store information about all of the objects. This info is referred to as metadata.

Metadata can include:

- Means of creation of the data
- Purpose of the data
- Time and date of creation, along with the creator and author of the data
- Location on a computer network where the data was created
- Standards used and file size

Data Dictionary

The data dictionary is owned by system user DBC. The data dictionary is composed of tables and views. Data dictionary views provide access to the information in the tables.

Tables and views are reserved for use by the system which contains the metadata about the database objects such as privileges, system events, and system usage. For example,

Tables store:

- Indexes
- Constraints
- table creation (date and time).

Views store:

- View or macros text
- creation time details
- access privileges

Components

Now that we have reviewed some of the objects within the Teradata Database, let's take a look at the major components that make up the database.

Node

A node is a term for a general-purpose processing unit under the control of a single operating system. It serves as the hardware platform upon which the database software operates. The basic building block for a Teradata system, the node is where the processing occurs for the database.

Clique

A clique is a set of Teradata nodes that share a common set of disk arrays. Cabling a subset of nodes to the same disk arrays creates a clique. **Add in info from email related Cliques**

Virtual Storage

Teradata Virtual Storage (TVS) is designed to allow the Teradata Database to make use of new storage technologies (specifically SSD or Solid State Drives). Teradata virtual storage is software that automatically looks at access patterns and will store data that is accessed more frequently on the faster

devices, for example SSD and data that is accessed less frequently on slower disk drives, for example HDD – Hard Disk Drives.

BYNET

BYNET handles the internal communication of the Teradata Database. All communication between Parsing Engines and AMPs are done via the Bynet.

Hot Standby Node

A hot standby node is included in the customers data warehouse to improve availability and maintain performance in case of a node failure. As a member of a clique, it does not normally participate in Teradata operations but can be brought in to compensate for the loss of a node in the clique.

Additional Options

In addition to normal database **operation** there are options to ensure data integrity and protection you also have an option for back up and recovery.

Archive/Recovery (Teradata ARC)

This is a means of archiving data to tape and disk and restoring data to the Teradata Database.

Teradata Data-Stream Architecture (DSA)

This has similar functionality to ARC, but generally provides improved performance.

Virtual Components

The following items would be considered virtual components which simply means that they eliminate dependency on specialized physical components (virtual processors-Vprocs). The virtual components are a set of software processes that run on a node under Teradata. Now let's take a look at the major components that make up the database.

Parsing Engine

The Parsing Engine, or PE performs session control, query parsing, security validation, query optimization, and query dispatch. The Parsing engine interprets sequel requests, receives input records and passes data. To do that it sends the messages through the Message Passing Layer to the AMP. An AMP is responsible for the storage, maintenance, and retrieval of the data under its control.

Access Module Processor (AMP)

The Access Module Processor, more commonly known as the AMP is responsible for managing a portion of the database. Each AMP controls some portion of each table on the system. AMPs do all of the physical work associated with generating an answer set including, sorting, aggregating, formatting, and converting

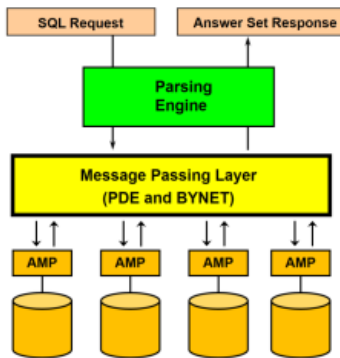
Virtual Disk Space

Virtual Disk space is associated with an AMP. Tables and data rows are stored in this space. A virtual disk is usually assigned to two or more disk drives in a disk array.

More About Virtual Components

Message Parsing Layer

Message Parsing Layer



The message parsing layer or communications layer is responsible for carrying messages between **virtual processors (ampAMPs, and PEs)**, making Teradata parallelism possible, and merging answer sets back to the PE. It is a combination of the PDE, BYNET software, and BYNET Hardware for MPP systems.

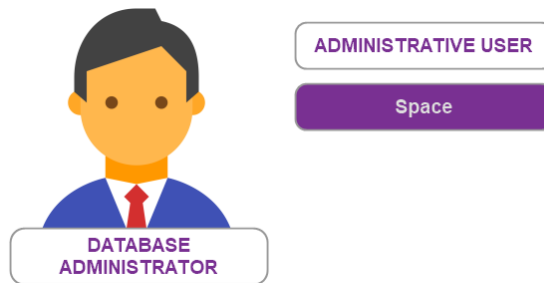
Parallel Database Extensions (PDE)

PDE is a software interface layer that lies between the operating system and Teradata Database. PDE supports the parallelism that gives Teradata Database its speed and linear scalability. PDE provides Teradata Database with the ability to:

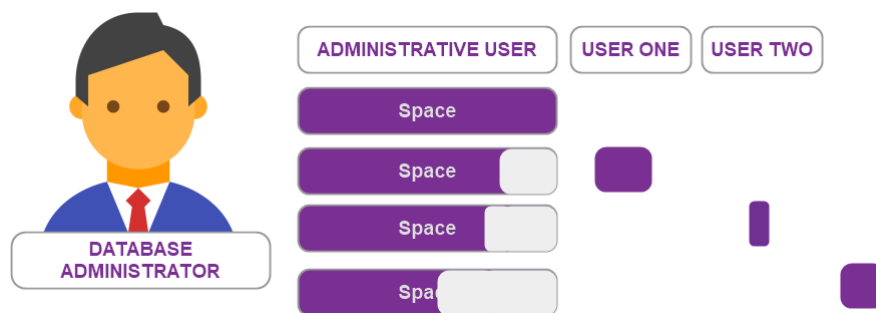
- Run in a parallel environment
- Execute vprocs
- Manage and prioritize Teradata Database workloads
- Consistently manage memory, I/O, and messaging system interface across multiple OS platforms

How Teradata uses Space

The way the Teradata Database performs space management is different from most other database implementations.



Before defining application users and databases, the Database Administrator creates a special administrative user and commonly assigns most of the space in the system to that user.



As the administrative user creates additional users and databases, space assigned to those objects will be subtracted from the administrative user's space. As these users and databases create their own objects, they will give up some of their space to these new users and databases.

Most database vendors do not handle space this way. Once space is allocated to a table, it cannot be made available again without the Database Administrator having to perform a re-organization and re-allocate the space and partition the data. Teradata's approach to space management is flexible, dynamic and requires minimal involvement of the Database Administrator.

Types of Space

There are three types of space within the Teradata Database:

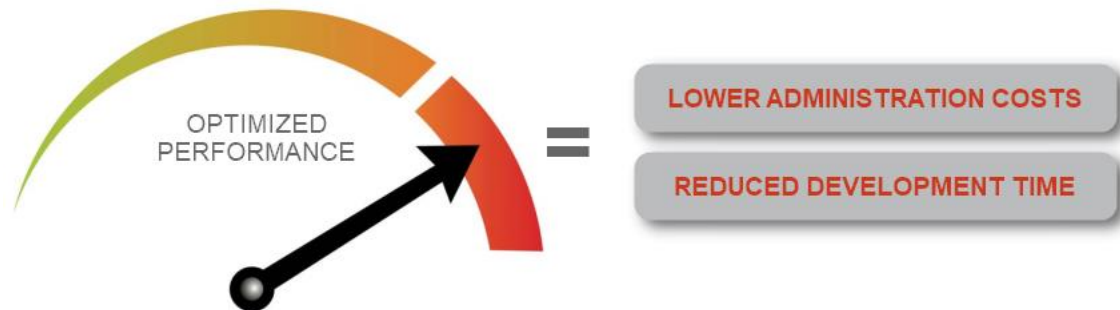
- Permanent Space, also known as Perm space is used for storing the data rows of tables and is the maximum storage assigned to a user and/or database. It is not pre-allocated or reserved ahead of time, it is available on-demand and is deducted from the owner's Perm Space.
- Spool Space is unused Perm Space used to hold intermediate query results or formatted answer sets for queries. Once the query is complete, the Spool Space is released. All databases have an upper limit of Spool Space. Theoretically, a user could use all of the unallocated space in the system for their query.
- Temporary Space is sometimes referred to as Temp Space and is used for global temporary tables. Tables created in Temp Space will survive a restart and remain available to the user until their session is terminated.

In addition to these three types, it is important to understand database space. Database Space is the total amount of space available in the database to create and store objects that need permanent space. All space assigned to the User/Database is equally divided among all the of AMPs in the system. For example, if a database was created with 100 GB of permanent space and we had 10 AMPs in our system. Each AMP would get a limit of 10GB.

Module 2 – Data Distribution and Access

Data Distribution

The Teradata Database uses hashing to dynamically distribute data across all AMPs. Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string.



The Teradata Database's hashed data distribution scheme provides optimized performance with minimal tuning and no manual reorganizations, resulting in lower administration costs and reduced development time.

More About Hashing

For PIs, Teradata Database generates a row hash by hashing the values of the PI columns. The row hash and a sequence number, which is assigned to distinguish between rows with the same row hash within a table, are collectively called a row identifier and uniquely identify each row in a table.

What an Index is and How it is Used

The Teradata Database has several types of indexes. Depending on the type of Index, Teradata may use it to distribute data, retrieve data efficiently, or enhance performance.

Following are the types:

- **Primary Index (PI)**
 - A Primary Index (PI) is defined when the table is created, and is there for two major reasons:
 - To determine data distribution
 - Enhance performance
- **No Primary Index (NOPI)**
 - A No Primary Index (NOPI) table is simply a table without a primary index. NoPI tables are typically used as staging tables for the initial data load.
- **Secondary Index (SI)**
 - A Secondary Index (SI) can be used to enforce uniqueness on a column. The SI also provides an alternate path to access data.
- **Join Index (JI)**
 - A Join Index (JI) is an optional index which may be created by the user to improve performance of complex queries.

Relationship Between Primary Indexes and Primary Keys

Below shows the relationship between primary keys and primary indexes.

Primary Key:

- Relational modeling convention which ensures each row to be uniquely identified.
- Consists of more or more columns
- May not be null and must be unique.
- Logical concept of data modeling.

Primary Index:

- Teradata convention which determines how the row will be stored and accessed.
- Consists of more or more columns
- May be unique or non-unique.
- Provides a physical access path and is also a mechanism for determining where the data is stored.

A primary key is a logical database concept. It may not be the best column, or columns, to choose as a primary index for a table.

How the Teradata Database Uses Indexes

So how does the Teradata database use indexes? And how can those uses benefit the customer? Let's look at several ways.



To improve performance – How? Accessing data with equality on Primary Index will result in a one AMP operation, which is extremely fast and efficient. Similar improvements can be seen with the use of other indexes



A good choice of Primary Index will ensure even data distribution.



Indexes can provide an easier and faster way of accessing and locating data and thus reducing unwanted inputs and outputs.



The use of UPI (Unique Primary Index) and USI (Unique Secondary Index) ensures effective maintenance of uniqueness on the table.



The Partition Primary Index feature can dramatically improve the performance of complex Range-based queries.

Index Types

FUPI (Unique Primary Index)

Guarantees uniform distribution of rows, has no duplicate values, and is made up of one or more columns

NUPI (non unique primary index)

Can cause skewed data, **may-allow**s have duplicate values, and is made up of one or more columns

USI (Unique Secondary Index)

Not involved in data distribution, has no dup values, enables data access via an alternate path

NUSI (non unique secondary index)

One or more columns, may have dup values and is used to improve access

PPI (Partition Primary Index)

Can dramatically improve the performance of complex Range-based queries. Partitioned Primary Index (PPI) is an indexing mechanism that is useful in improving the performance of certain queries. When rows are inserted into a table, they are stored in an AMP and arranged by their row hash order. When a table is defined with PPI, the rows are sorted by their partition number. Within each partition, they are arranged by their row hash. Rows are assigned to a partition based on the partition expression defined.

NOPI

No Primary Index: Support faster loads and inserts into tables with no primary index defined.

Types, Levels, and Functionality of Locking

Locks are automatically acquired during the processing of a request and released at the termination of the request. In addition, users can specify locks.



There are four types of Locks in Teradata:

- Exclusive, which prevents any other type of concurrent access
- Write which Prevents other Reads, Writes, and Exclusives
- Read which prevents other Writes and Exclusives
- Access which prevents Exclusive only

Locks may be applied at four levels:

- The database level applies to all Tables/Views in the Database
- Table/View level applies to all Rows in the Table/View
- Partition level which applies to all rows within the specified partition(s)
- the Row hash level. A row hash is an algorithm which determines which AMP gets the row

The Rules of Locking

Locking requests are queued behind all outstanding incompatible lock request for the same object. There is no time-out for SQL Requests waiting for a lock.

LOCK REQUEST	LOCK LEVEL HELD				
	NONE	ACCESS	READ	WRITE	EXCLUSIVE
ACCESS	Granted	Granted	Granted	Granted	Queued
READ	Granted	Granted	Granted	Queued	Queued
WRITE	Granted	Granted	Queued	Queued	Queued
EXCLUSIVE	Granted	Queued	Queued	Queued	Queued

Module 3 - Security and Privacy

User Security

There are some key concepts about the Teradata Database user security that you should be aware of. Let's take a look at them.

Users

A user is almost the same as a database except that a user can actually log on to the Teradata Database. To accomplish this, a user must have a password.

Roles

A Role is an administration/security concept which can help simplify database administration. A Role is simply a collection of Access Rights or privileges.

When multiple users require access rights to the same database objects, a "fictitious" user can be set up and be assigned access rights to those objects. Then that role can be assigned to the multiple users.

External Roles

External roles are different from internal roles. External roles are mapped to users on the directory server for example LDAP.

Privileges

Privileges sometimes referred to as access rights define the type of activities you can perform during a session. There are four types of privileges. Automatic Privileges which are privileges given to creators and, in the case of users and databases, their created objects. Explicit privileges are privileges assigned by granting access rights to specific objects. Owners of objects have the implicit right to grant privileges for any, or all, of their owned objects either to themselves or to any other user or database are given ownership privileges. Ownership rights cannot be taken away unless ownership is transferred. Finally, Inherited privileges are passed on indirectly to a user based on its relationship to another user or role to which the privileges were granted directly.

Profiles

Profiles define system attributes such as account (id)s, default database, spool space allocation, temporary space allocation, password attributes, and QueryBand.

A profile is a set of common user attributes that can be applied to a group of users. Using profiles is very helpful for the Database Administrator (DBA) since profiles make changing parameters for a group of users a single step. Change the profile and all users assigned that profile will be changed! Without the profile, each user in the group would need to be changed individually

Authentication

Authentication is the process by which the user identity in the logon is verified. There are two categories of authentication that are available. Authentication by Teradata Database requires that the user and its privileges are defined in the database and external Authentication which allows Teradata Database users to be authenticated by an agent running on the same network as Teradata Database and its clients.

Middle-tier Application

Middle-tier applications may stand between end users and Teradata Database, accepting requests from users, processing them and eventually returning results to the users. The middle-tier application logs on

to the database, is authenticated as a permanent database user, and establishes a connection pool. The application is then responsible for authenticating the individual application end users.

Privacy Mechanisms



The Teradata Secure Zones feature allows you to create one or more exclusive database hierarchies, called zones, within a single Teradata Database system. Access to the data in each zone and the database administration is handled separately from the overall Teradata Database system and from other zones.

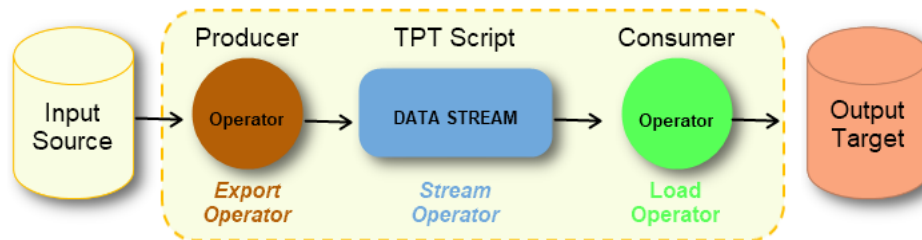
Secure Zones are useful in situations where the access to data must be tightly controlled and restricted.

Row Level Security, or RLS is used to control user access by table row and by SQL operation. RLS Access Rules are based on the comparison of the RLS access capabilities of each user and the RLS access requirements for each row.

Module 4 - Teradata Tools and Utilities

Data Integration Tools

Teradata Parallel Transporter, or TPT is client software that performs data extractions, data transformations, and data loading functions in a scalable, parallel processing environment. Jobs in TPT are run using "operators" which define the type of activity to be performed.



The two key operator types in TPT are producer and consumer:

- Producer Operators extract data from a data source by reading the data and writing it to the data stream.
- Consumer Operators read the data from the data stream and writes or loads it to the target which might be Teradata tables or an external data store.

Other Operators:

- The Load Operator is a consumer-type operator that is typically used for initial loading of Teradata tables. The Load operator inserts the data it consumes from the data stream(s) into individual rows of the target table. Inserting is the only operation supported by the Load operator.
- The Update Operator is a consumer-type operator that can be used to insert new rows into the database like the Load operator, it can also perform updates, deletes and inserts to the target tables. Its primary function is to perform high volume maintenance transactions against multiple Teradata tables.
- The Export Operator is a producer-type operator that is designed to extract large volumes of data at high speed from Teradata tables using block transfers over multiple sessions.
- The Stream Operator is a consumer operator, which performs high-speed parallel Inserts, Updates, and Deletes in near-real-time to one or more empty or preexisting Teradata Database tables without locking the target tables.

NOTE: Batch/Basic Teradata Query Language, BTEQ is a general-purpose command-based utility for submitting Sequel requests to the Teradata Database and deliver the response in the format required. BTEQ can be used for both exporting and importing data.

UDA Tools

Listener

Listener is an enterprise wide platform for ingesting high volume real-time streams of data. It is used to capture and control large volumes of real-time streams from any source such as websites, social feeds, systemj and application logs.

Legacy Tools

FastLoad

This utility provides high-performance data loading from client files into empty tables.

MultiLoad

The multiload utility provides high-performance data maintenance, including inserts, updates, and deletions to existing tables.

Teradata Parallel Data Pump (TPump)

Provides continuous updates of tables; performs insert, update, and delete operations or a combination of those operations on multiple tables using the same source feed.

Administration and Monitoring Tools

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 so they can make intelligent decisions about their work day. Teradata Viewpoint also enables database and system administrators as well as business users to monitor and manage Teradata Database systems from anywhere using a standard web browser.

The Teradata Studio family consists of two client tools, Teradata Studio and Teradata Studio Express, used for information discovery, database exploration, and database administration. The tools are intended for two different work-based roles. Teradata Studio targeted for DBAs, is a client-based graphical user interface for performing database administration, query development, and management tasks on Teradata Databases, Teradata Aster Databases, and Hadoop systems.

Teradata Studio Express is the base level product and is for business Users and Sequel Developers. It is an information discovery tool for retrieving and displaying data from Teradata Database systems. Its main use is for running Sequel statements, displaying result set data, and storing Sequel execution history.

A Teradata Database Management Tools and Utilities

More About Teradata Studio

In addition to those already mentioned, Teradata Studio allows you to:

Build and edit SQL requests visually using SQL Query Builder

Generate and run COLLECT STATISTICS requests

Copy database objects to another database or system

Move space between databases

TASM (Teradata Active Systems Management)

TASM is a set of products that interact with each other to facilitate automation in four key areas:

Workload management

Performance tuning

Capacity planning

Performance monitoring

Teradata Unity

Unity's powerful features directly address and overcome the challenges organizations face in a multi-system environment. It automates and simplifies tasks enabling corporations to manage multiple active Teradata systems so they appear as one ecosystem.