Teradata Architecture Class Outline

The Teradata Architecture

- What is Parallel Processing?
- The Basics of a Single Computer
- Teradata Parallel Processes Data
- Parallel Architecture
- The Teradata Architecture
- All Teradata Tables are spread across ALL AMPS
- Teradata Systems can Add AMPs for Linear Scalability
- Understand that Teradata can scale to incredible size
- AMPs and Parsing Engines (PEs) live inside SMP Nodes
- Each Node is attached via a Network to a Disk Farm
- Two SMP Nodes Connected Become One MPP System
- There are Many Nodes in a Teradata Cabinet
- Inside a Teradata Node
- The Boardless BYNET and the Physical BYNET
- The Parsing Engine
- The AMPs Responsibilities

The Primary Index

- The Primary Index is defined when the table is CREATED
- A Unique Primary Index (UPI)
- Primary Index in the WHERE Clause - Single-AMP Retrieve
- Using EXPLAIN
- A Non-Unique Primary Index (NUPI)
- Primary Index in the WHERE Clause - Single-AMP Retrieve
- Using EXPLAIN in a NUPI Query
- A conceptual example of a Multi-Column Primary Index
- Primary Index in the WHERE Clause - Single-AMP Retrieve
- A conceptual example of a Table with NO PRIMARY INDEX
- A Full Table Scan is likely on a table with NO Primary Index
- An EXPLAIN that shows a Full Table Scan
- Table CREATE Examples with four different Primary Indexes
- What happens when you forget the Primary Index?
- Why create a table with No Primary Index (NoPI)?

Hashing of the Primary Index

- The Hashing Formula Facts
- The Hash Map Determines which AMP will own the Row
- The Hash Map Determines which AMP will own the Row
- Placing rows on the AMP
- Placing rows on the AMP Continued
- A Review of the Hashing Process
- Non-Unique Primary Indexes have Skewed Data
- The Uniqueness Value
- The Row Hash and Uniqueness Value make up the Row-ID
- A Row-ID Example for a Unique Primary Index
- A Row-ID Example for a Non-Unique Primary Index (NUPI)

Two Reasons why each AMP Sorts their rows by the Row-ID
AMPs sort their rows by Row-ID to Group like Data
AMPs sort their rows by Row-ID to do a Binary Search
Table CREATE Examples with four different Primary Indexes
Null Values all Hash to the Same AMP
A Unique Primary Index (UPI) Example
A Non-Unique Primary Index (NUPI) Example
A Multi-Column Primary Index Example
A No Primary Index (NoPI) Example

Teradata - The Cold Hard Facts

All Teradata Tables are spread across All AMPs
The Table Header and the Data Rows are Stored Separately
An AMP Stores the Rows of a Table inside a Data Block
To Read a Data Block, an AMP Moves the Block into Memory
Nothing is done on disk and everything is done in Memory
Most Taxing thing for an AMP is Moving Blocks into Memory
A Full Table Scan Means All AMPs must Read All Rows
The “Achilles Heel and slowest process is Block Transfer
Each Table has a Primary Index
A Query Using the Primary Index is a Single AMP Retrieve.
As Rows are added a Data Block will Eventually Split
A Full Table Scan Means All AMPs must Read All Blocks
A Primary Index Query uses a Single AMP and Single Block
Each AMP Can Have Many Blocks for a Single Table
A Full Table Scan Means All AMPs must Read All Blocks
Quiz – How Many Blocks Move into FSG Cache?
Answer – How Many Blocks Move into FSG Cache?
Quiz – How Many Blocks Move Using the Primary Index?

Answer – How Many Blocks Move Using the Primary Index?
Synchronized Scan (Sync Scan)
EXPLAIN Using a Synchronized Scan
Intelligent Memory (Teradata V14.10)
Teradata V14.10 Intelligent Memory Gives Data a Temperature
Data deemed VeryHot stays in each AMP's Intelligent Memory
Intelligent Memory Stays in Memory
What is the Goal of a Teradata Physical Database Design?

Inside the AMPs Disk

Rows are Stored in Data Blocks which are stored in Cylinders
An AMP's rows are stored inside a Data Block in a Cylinder
An AMP's Master Index is used to find the Right Cylinder
The Row Reference Array (RRA) Does the Binary Search?
A Block Splits into Two Blocks at Maximum Block Size
Data Blocks Maximum Block Size has Changed (V14.10)
The New Block Split with Teradata V14.10
The Block Split with Even More Detail in Teradata V14.10
Teradata V14.10 Block Split Defaults
There is One Master Index and Thousands of Cylinder Indexes
Blocks Continue to Split as Tables Grow Larger
FYI – Some Advanced Information about Data Block Headers
A top down view of Cylinders
There are Hot, Warm, and Cold Cylinders
Cylinders are used for Perm, Spool, Temp, and Journals
Each AMP has Their Own Master Index
Each Cylinder on an AMP has a Cylinder Index
Quiz – What Two Things Does and AMP Read?
Answer – What Two Things Does and AMP Read?

Quiz – How Many Row Reference Arrays do you see?
Answer – How Many Row Reference Arrays do you see?
Quiz – How Many Row Reference Arrays are there Now?
Answer – How Many Row Reference Arrays do you see?
Quiz – How Many Row Reference Arrays in Total?
Answer – How Many Row Reference Arrays in Total?
Quiz – How Many Cylinder Indexes are here?
Answer – How Many Cylinder Indexes are here?
A More Detailed Illustration of the Master Index
A Real-World View of the Master Index
An Even More Realistic View of an AMP's Master Index
The Cylinder Index
An Even More Realistic View of a Cylinder Index
How a Query using the Primary Index works
How the AMPs Do a Full Table Scan
How an AMP Reads Using a Primary Index

Partition Primary Index (PPI) Tables
The Concept behind Partitioning a Table
Creating a PPI Table with Simple Partitioning
A Visual Display of Simple Partitioning
An SQL Example that explains Simple Partitioning
Creating a PPI Table with RANGE_N Partitioning per Month
A Visual of One Year of Data with Range_N per Month
An SQL Example explaining Range_N Partitioning per Month
A Partition # and Row-ID = Row Key
An AMP Stores its Rows Sorted in only Two Different Ways
Creating a PPI Table with RANGE_N Partitioning per Day
A Visual of Range_N Partitioning Per Day

An SQL Example that explains Range_N Partitioning per Day
Creating a PPI Table with RANGE_N Partitioning per Week
A Visual of Range_N Partitioning Per Week
SQL Example that explains Range_N Partitioning per Week
A Clever Range_N Option
Creating a PPI Table with CASE_N
A Visual of Case_N Partitioning
An SQL Example that explains CASE_N Partitioning
How many partitions do you see?
Number of PPI Partitions Allowed
How many partitions do you see?
NO CASE and UNKNOWN Partitions Together
A Visual of Case_N Partitioning
Combining Older Data and Newer Data in PPI
A Visual for Combining Older Data and Newer Data in PPI
The SQL on Combining Older Data and Newer Data in PPI
Multi-Level Partitioning Combining Range_N and Case_N
A Visual of Multi-Level Partitioning
The SQL on a Multi-Level Partitioned Primary Index
NON-Unique Primary Indexes (NUPI) in PPI
PPI Table with a Unique Primary Index (UPI)
Tricks for Non-Unique Primary Indexes (NUPI)
Character Based PPI for RANGE_N
A Visual for Character-Based PPI for RANGE_N
The SQL on Character-Based PPI for RANGE_N
Character-Based PPI for CASE_N
Dates and Character-Based Multi-Level PPI
TIMESTAMP Partitioning
Using CURRENT_DATE to define a PPI

ALTER to CURRENT_DATE the next year
ALTER to CURRENT_DATE with Save
Altering a PPI Table to Add or Drop Partitions
Deleting a Partition
Deleting a Partition and saving its contents
Using the PARTITION Keyword in your SQL
SQL for RANGE_N
SQL for CASE_N

Columnar Tables

Columnar Tables have NO Primary Index
This is NOT a NoPI Table
NoPI Tables Spread rows across all-AMPs Evenly
NoPI Tables used as Staging Tables for Data Loads
NoPI Table Capabilities
NoPI Table Restrictions
What does a Columnar Table look like?
Comparing Normal Table vs. Columnar Tables
Columnar Table Fundamentals
Example of Columnar CREATE Statement
Columnar can move just One Container to Memory
Containers on AMPs match up perfectly to rebuild a Row
Indexes can be used on Columns (Containers)
Indexes can be used on Columns (Containers)
Visualize a Columnar Table
Single-Column vs. Multi-Column Containers
Comparing Normal Table vs. Columnar Tables
Columnar Row Hybrid CREATE Statement
Columnar Row Hybrid Example

Columnar Row Hybrid Query Example
Review of Row-Based Partition Primary Index (PPI)
Visual of Row Partitioning (PPI Tables) by Month
CREATE Statement for both Row and Column Partition
Visual of Row Partitioning (PPI Tables) and Columnar
How to Load into a Columnar Table
Columnar NO AUTO COMPRESS
Auto Compress in Columnar Tables
Auto Compress Techniques in Columnar Tables
When and When NOT to use Columnar Tables

Space

When your System Arrives, there is only User named DBC
USER DBC
First Assignment is to create another User just under DBC
USER DBC
Perm and Spool Space
Perm Space is for Permanent Tables
Spool Space is work space that builds a User's Answer Sets
Spool Space is in an AMP's Memory and on its Disk
Users are Assigned Spool Space Limits
What is the Purpose of Spool Limits?
Why did my query Abort and say "Out of Spool"?
How can Skewed Data cause me to run "Out of Spool"?
Why did my Join cause me to run "Out of Spool"?
Finding out how much Space you have
Space per AMP on all tables in a Database shows Skew
What does my system look like when it first arrives?
DBC owns all the PERM Space in the system on day one

DBC's First Assignment is Spool Space
DBC's 2nd Assignment is to CREATE Users and Databases
The Teradata Hierarchy Begins
The Teradata Hierarchy Continues
Differences between PERM and SPOOL
Databases, Users, and Views
What are Similarities between a DATABASE and a USER?
What is the Difference between a DATABASE and a USER?
Objects that take up PERM Space
A Series of Quizzes on Adding and Subtracting Space
Answer 1 to Quiz on Space
Space Transfer Quiz
Answer to Space Transfer Quiz
Drop Space Quiz
Answers to Drop Space Quiz

The User Environment
DBC is the only user when the system first arrives
DBC will Create Databases and Give them Space
DBC will create some initial Users
A Typical Teradata Environment
What are Similarities between a DATABASE and a USER?
Roles
Create a Role and then Assign that Role Its Access Rights
Create a User and Assign them a Default Role
Granting Access Rights
There are Three Types of Access Rights
Description of the Three Types of Access Rights
Profiles

Creating a Profile and a User
ProfileInfoVX, RoleMembers, RoleInfo and UserRoleRights
Accounts and their Associated Priorities
Creating a User with Multiple Account Priorities
Account String Expansion (ASE)
The DBC.AMPUsage View
Teradata TASM provides a User Traffic System
Teradata Viewpoint

Secondary Indexes

Creating a Unique Secondary Index (USI)
What is in a Unique Secondary Index (USI) Subtable?
A Unique Secondary Index (USI) Subtable is hashed
How the Parsing Engine uses the USI Subtable
A USI is a Two-AMP Operation
Creating a Non-Unique Secondary Index (NUSI)
What is in a Unique Secondary Index (USI) Subtable?
Non-Unique Secondary Index (NUSI) Subtable is AMP Local
How the Parsing Engine uses the NUSI Subtable
Creating a Value-Ordered NUSI
The Hash Map Determines which AMP will own the Row
A Unique Primary Index Spreads the Data Evenly
Quiz – Answer the Tough USI Questions
Answer to Quiz – Answer the Tough USI Questions
A Picture with a Base Table, USI, and NUSI Subtable
Quiz – Tough Questions on the USI and NUSI Subtables
Answer – Tough Questions on the USI and NUSI Subtables
A Query Using an USI Only Moves Two Blocks
A Query Using A NUSI Always Uses All AMPs

Two Non-Unique Secondary Indexes (NUSI) on a Table
A NUSI BITMAP Query (1 of 3)
A NUSI BITMAP Theory (2 of 3)
A NUSI Bitmap in Action (3 of 3)
A Brilliant Technique for a Unique Secondary Index
The USI for Partitioned Tables Points to the Row Key
A Brilliant Technique for a Non-Unique Secondary Index
The NUSI for Partitioned Tables Points to the Row Key
How the PE Decides on the NUSI or the Full Table Scan
The Bigger Quiz
The Bigger Quiz Answers
Multiple Choice DBA
What are the Big Four Tactical Queries?

Temporal Tables Create Functions
Three types of Temporal Tables
CREATING a Bi-Temporal Table
PERIOD Data Types
Bi-Temporal Data Type Standards
Bi-Temporal Example – Tera-Tom buys!
A Look at the Temporal Results
Bi-Temporal Example – Tera-Tom Sells!
Bi-Temporal Example – How the data looks!
Normal SQL for Bi-Temporal Tables
NONSEQUENCED SQL for Temporal Tables
AS OF SQL for Temporal Tables
NONSEQUENCED for Both
Creating Views for Temporal Tables
Bi-Temporal Example – Socrates is DELETED!

Bi-Temporal Results – Socrates is DELETED

How Joins Work Internally

Teradata Join Quiz
Teradata Join Quiz Answer
The Joining of Two Tables
Teradata Moves Joining Rows to the Same AMP
Imagine Joining Two NoPI Tables that have No Primary Index
Both Tables are redistributed to Join Rows on the Same AMP
How do you join if One Table is Big and One Table is Small?
Duplicate the Small Table on Every AMP (like a mirror)
What Could You Do If Two Tables Joined 1000 Times a Day?
Joining Two Tables with the same PK/FK Primary Index
A Join with No Redistribution or Duplication
A Performance Tuning Technique for Large Joins
The Joining of Two Tables with an Additional WHERE Clause
An Example of the Fastest Join Possible
Using a Simple Volatile Table
A Volatile Table with a Primary Index
Using a Simple Global Temporary Table
Two Brilliant Techniques for Global Temporary Tables
The Joining of Two Tables Using a Global Temporary Table
Quiz – How Much Data Moves Across the BYNET?
Answer – How Much Data Moves Across the BYNET?
Teradata V14.10 Join Feature PRPD

Join Indexes

Creating a Multi-Table Join Index
Visual of a Join Index

Outer Join Multi-Table Join Index
Visual of a Left Outer Join Index
Compressed Multi-Table Join Index
A Visual of a Compressed Multi-Table Join Index
Creating a Single-Table Join Index
Conceptual of a Single Table Join Index on an AMP
Single Table Join Index Great For LIKE Clause
Single Table Join Index with Value Ordered NUSI
Aggregate Join Indexes
Compressed Single-Table Join Index
Aggregate Join Index
New Aggregate Join Index (Teradata V14.10)
Sparse Join Index
A Global Multi-Table Join Index
Creating a Hash Index
Join Index Details

Collect Statistics
The Teradata Parsing Engine (Optimizer) is Cost Based
The Purpose of Collect Statistics
When Teradata Collects Statistics it creates a Histogram
The Interval of the Collect Statistics Histogram
Histogram Quiz
Answers to Histogram Quiz
What to COLLECT STATISTICS On?
Why Collect Statistics?
How do you know if Statistics were collected on a Table?
A Huge Hint that No Statistics Have Been Collected
The Basic Syntax for COLLECT STATISTICS

COLLECT STATISTICS Examples for a better Understanding
The New Teradata V14 Way to Collect Statistics
Where Does Teradata Keep the Collected Statistics?
The Official Syntax for COLLECT STATISTICS
How to Recollect STATISTICS on a Table
Teradata Always Does a Random AMP Sample
Random Sample is kept in the Table Header in FSG Cache
Multiple Random AMP Samplings
How a Random AMP gets a Table Row count
Random AMP Estimates for NUSI Secondary Indexes
USI Random AMP Samples are Not Considered
There's No Random AMP Estimate for Non-Indexed Columns
The PE's Plan if No Statistics Were Collected?
Stale Statistics Detection and Extrapolation
Extrapolation for Future Dates
How to Copy a Table with Data and the Statistics?
How to Copy a Table with NO Data and the Statistics?
COLLECT STATISTICS Directly From another Table
When to COLLECT STATISTICS Using only a SAMPLE
Examples of COLLECT STATISTICS Using only a SAMPLE
Examples of COLLECT STATISTICS For V14
How to Collect Statistics on a PPI Table on the Partition
Teradata V12 and V13 Statistics Enhancements
Teradata V14 Statistics Enhancements
Teradata V14 Summary Statistics
Teradata V14 MaxValueLength
Teradata V14 MaxIntervals
Teradata V14 Sample N Percent
Teradata V14.10 Statistics Collection Improvements

Teradata V14.10 Statistics Collection Improvements
Teradata V14.10 AutoStats feature
Teradata Statistics Wizard

Temporary Tables

There are three types of Temporary Tables
CREATING A Derived Table
Naming the Derived Table
Aliasing the Column Names in the Derived Table
Most Derived Tables Are Used To Join To Other Tables
Multiple Ways to Alias the Columns in a Derived Table
Our Join Example with a Different Column Aliasing Style
Column Aliasing Can Default for Normal Columns
CREATING A Derived Table using the WITH Command
Our Join Example With the WITH Syntax
The Same Derived Query shown Three Different Ways
Quiz - Answer the Questions
Answer to Quiz - Answer the Questions
Clever Tricks on Aliasing Columns in a Derived Table
A Derived Table lives only for the lifetime of a single query
An Example of Two Derived Tables in a Single Query
WITH RECURSIVE Derived Table
Defining the WITH Recursive Derived Table
Looping Through the Recursive Derived Table
Looping Through a Second Time
Looping Through a Third Time
Looping Through and Adding Nothing Ends the Loop
Looping Through the WITH Recursive Derived Table
Creating a Volatile Table

You Populate a Volatile Table with an INSERT/SELECT
The Three Steps to Use a Volatile Table
Why Would You Use the ON COMMIT DELETE ROWS?
The HELP Volatile Table Command Shows your Volatiles
A Volatile Table with a Primary Index
The Joining of Two Tables Using a Volatile Table
You Can Collect Statistics on Volatile Tables
The New Teradata V14 Way to Collect Statistics
Four Examples of Creating a Volatile Table Quickly
Four Advanced Examples of Creating a Volatile Table Quickly
Creating Partitioned Primary Index (PPI) Volatile Tables
Using a Volatile Table to Get Rid of Duplicate Rows
Using a Simple Global Temporary Table
Two Brilliant Techniques for Global Temporary Tables
The Joining of Two Tables Using a Global Temporary Table
CREATING A Global Temporary Table

Teradata Load Utilities Introduction
The Teradata Utilities
Block Level Utilities
Row Level Utilities
Fast Path Inserts Using Insert/Select
Fast Path Deletes
Freespace Percent
Referential Integrity and Load Utility Solutions
Teradata has a No Primary Index Table called a NoPI Table
This is NOT Necessarily a NoPI Table
NoPI Tables Spread rows across all-AMPs Evenly
NoPI Tables used as Staging Tables for Data Loads

NoPI Table Capabilities
NoPI Table Restrictions
Why Would a NoPI Table have a Row-ID?
BTEQ – Batch Teradata Query Tool
How to Logon to BTEQ in Interactive Mode
Running Queries in BTEQ in Interactive Mode
BTEQ Commands vs BTEQ SQL Statements
WITH BY Command for Subtotals
WITH Command for a Grand Total
WITH and WITH BY Together for Subtotals and Grand Totals
How to Logon to BTEQ in a SCRIPT
Running Queries in BTEQ through a Batch Script
Running a BTEQ Batch Script through the Command Prompt
Running a BTEQ Batch Script through the Run Command
Using Nexus to Build Your BTEQ Scripts
Using Nexus to Build Your BTEQ Scripts
FastLoad
Block Level Utility Limits
FastLoad has Two Phases
FastLoad Phase 1
FastLoad Phase 2
A Sample FastLoad Script Created by Nexus SmartScript
Executing the FastLoad Script
The Nexus SmartScript Easily Builds Your Utilities
The Nexus SmartScript FastLoad Builder
Create and Execute Your FastLoad Scripts with Nexus
MultiLoad
Block Level Utility Limits
MultiLoad has Five Phases

MultiLoad has IMPORT and DELETE Tasks
A Sample MultiLoad Script Created by Nexus SmartScript
TPump
TPump is NOT a Block Level Utility and has No Limits
Limitations of TPump
A Sample TPump Script Created by Nexus SmartScript
FastExport
New Rules for Block Utilities
A Sample FastExport Script Created by Nexus SmartScript
FastExport by Default places Null Indicators in Output
A Sample FastExport Script Created by Nexus SmartScript
What is TPT?
TPT Producers Create Streams and Consumers Write Them
The Four Major Operators of TPT
TPT can read from multiple source files in Parallel
TPT can have more Operators than Consumers
TPT Operators and their Equivalent Load Utility
How to Run a TPT Script