Teradata Basics

Course Summary

Description

In this course, students will learn the basics of Teradata architecture with a focus on what's important to know from an IT and Developer perspective.

Topics

- The Teradata Architecture
- The Primary Index
- Hashing of the Primary Index
- Teradata - The Cold Hard Facts
- Inside the AMPs Disk
- Partition Primary Index (PPI) Tables
- Columnar Tables
- Space
- The User Environment
- Secondary Indexes
- Temporal Tables Create Functions
- How Joins Work Internally
- Join Indexes
- Collect Statistics
- Temporary Tables
- Teradata Load Utilities Introduction

Audience

This course is designed for IT and Developers to help them understand the basics of Teradata Architecture.

Prerequisites

There are no prerequisites for this course.

Duration

One to two days

Teradata Basics

Course Outline

I. The Teradata Architecture

A. What is Parallel Processing?
B. The Basics of a Single Computer
C. Teradata Parallel Processes Data
D. Parallel Architecture
E. The Teradata Architecture
F. All Teradata Tables are spread across ALL AMPS
G. Teradata Systems can Add AMPs for Linear Scalability
H. Understand that Teradata can scale to incredible size
I. AMPs and Parsing Engines (PEs) live inside SMP Nodes
J. Each Node is attached via a Network to a Disk Farm
K. Two SMP Nodes Connected Become One MPP System
L. There are Many Nodes in a Teradata Cabinet
M. Inside a Teradata Node
N. The Boardless BYNET and the Physical BYNET
O. The Parsing Engine
P. The AMPs Responsibilities
Q. This is the Visual You Want to Know in order to Understand Teradata

II. The Primary Index

A. The Primary Index is defined when the table is CREATED
B. A Unique Primary Index (UPI)
C. Primary Index in the WHERE Clause - Single-AMP Retrieve
D. Using EXPLAIN
E. A Non-Unique Primary Index (NUPI)
F. Primary Index in the WHERE Clause - Single-AMP Retrieve
G. Using EXPLAIN in a NUPI Query
H. A conceptual example of a Multi-Column Primary Index
I. Primary Index in the WHERE Clause - Single-AMP Retrieve
J. A conceptual example of a Table with NO PRIMARY INDEX
K. A Full Table Scan is likely on a table with NO Primary Index
L. An EXPLAIN that shows a Full Table Scan
M. Table CREATE Examples with four different Primary Indexes
N. What happens when you forget the Primary Index?
O. Why create a table with No Primary Index (NoPI)?

III. Hashing of the Primary Index

A. The Hashing Formula Facts
B. The Hash Map determines which AMP will own the Row
C. The Hash Map determines which AMP will own the Row
D. Placing rows on the AMP
E. Placing rows on the AMP Continued
F. A Review of the Hashing Process
G. Non-Unique Primary Indexes have Skewed Data
H. The Uniqueness Value
I. The Row Hash and Uniqueness Value make up the Row-ID
J. A Row-ID Example for a Unique Primary Index
K. A Row-ID Example for a Non-Unique Primary Index (NUPI)
L. Two Reasons why each AMP Sorts their rows by the Row-ID
M. AMPs sort their rows by Row-ID to Group like Data
N. AMPs sort their rows by Row-ID to do a Binary Search
O. Table CREATE Examples with four different Primary Indexes
P. Null Values all Hash to the Same AMP
Q. A Unique Primary Index (UPI) Example
R. A Non-Unique Primary Index (NUPI) Example
S. A Multi-Column Primary Index Example
T. A No Primary Index (NoPI) Example

IV. The Primary Index

A. All Teradata Tables are spread across All AMPs
B. The Table Header and the Data Rows are Stored Separately
C. An AMP Stores the Rows of a Table inside a Data Block
D. To Read a Data Block, an AMP Moves the Block into Memory
E. Nothing is done on disk and everything is done in Memory
F. Most Taxing thing for an AMP is Moving Blocks into Memory
G. A Full Table Scan Means All AMPs must Read All Rows

Teradata Basics

Course Outline (cont'd)

H. The "Achilles Heel and slowest process is Block Transfer
I. Each Table has a Primary Index
J. A Query Using the Primary Index is a Single AMP Retrieve.
K. As Rows are added a Data Block will Eventually Split
L. A Full Table Scan Means All AMPs must Read All Blocks
M. A Primary Index Query uses a Single AMP and Single Block
N. Each AMP Can Have Many Blocks for a Single Table
O. A Full Table Scan Means All AMPs must Read All Blocks
P. Quiz – How Many Blocks Move into FSG Cache?

Q. Answer – How Many Blocks Move into FSG Cache?
R. Quiz – How Many Blocks Move Using the Primary Index?
S. Answer – How Many Blocks Move Using the Primary Index?
T. Synchronized Scan (Sync Scan)
U. EXPLAIN Using a Synchronized Scan
V. Intelligent Memory (Teradata V14.10)
W. Teradata V14.10 Intelligent Memory Gives Data a Temperature
X. Data deemed VeryHot stays in each AMP's Intelligent Memory
Y. Intelligent Memory Stays in Memory
Z. What is the Goal of a Teradata Physical Database Design?

V. Inside the AMPs Disk

A. Rows are Stored in Data Blocks which are stored in Cylinders
B. An AMP's rows are stored inside a Data Block in a Cylinder
C. An AMP's Master Index is used to find the Right Cylinder
D. The Row Reference Array (RRA) Does the Binary Search?
E. A Block Splits into Two Blocks at Maximum Block Size
F. Data Blocks Maximum Block Size has Changed (V14.10)
G. The New Block Split with Teradata V14.10
H. The Block Split with Even More Detail in Teradata V14.10
I. Teradata V14.10 Block Split Defaults
J. There is One Master Index and Thousands of Cylinder Indexes
K. Blocks Continue to Split as Tables Grow Larger

L. FYI – Some Advanced Information about Data Block Headers
M. A top down view of Cylinders
N. There are Hot, Warm, and Cold Cylinders
O. Cylinders are used for Perm, Spool, Temp, and Journals
P. Each AMP has Their Own Master Index
Q. Each Cylinder on an AMP has a Cylinder Index

R. Quiz – What Two Things Does and AMP Read?

S. Answer – What Two Things Does and AMP Read?
T. Quiz – How Many Row Reference Arrays do you see?
U. Answer – How Many Row Reference Arrays do you see?
V. Quiz – How Many Row Reference Arrays are there Now?
W. Answer – How Many Row Reference Arrays do you see?
X. Quiz – How Many Row Reference Arrays in Total?
Y. Answer – How Many Row Reference Arrays in Total?
Z. Quiz – How Many Cylinder Indexes are here?

AA. Answer – How Many Cylinder Indexes are here?

BB. A More Detailed Illustration of the Master Index

CC. A Real-World View of the Master Index
DD. An Even More Realistic View of an AMP's Master Index
EE. The Cylinder Index
FF. An Even More Realistic View of a Cylinder Index
GG. How a Query using the Primary Index works
HH. How the AMPs Do a Full Table Scan
II. How an AMP Reads Using a Primary Index

Teradata Basics

Course Outline (cont'd)

VI. Partition Primary Index (PPI) Tables

A. The Concept behind Partitioning a Table
B. Creating a PPI Table with Simple Partitioning
C. A Visual Display of Simple Partitioning
D. An SQL Example that explains Simple Partitioning
E. Creating a PPI Table with RANGE_N Partitioning per Month
F. A Visual of One Year of Data with Range_N per Month
G. An SQL Example explaining Range_N Partitioning per Month
H. A Partition # and Row-ID = Row Key
I. An AMP Stores its Rows Sorted in only Two Different Ways
J. Creating a PPI Table with RANGE_N Partitioning per Day
K. A Visual of Range_N Partitioning Per Day
L. An SQL Example that explains Range_N Partitioning per Day
M. Creating a PPI Table with RANGE_N Partitioning per Week
N. A Visual of Range_N Partitioning Per Week
O. SQL Example that explains Range_N Partitioning per Week
P. A Clever Range_N Option
Q. Creating a PPI Table with CASE_N
R. A Visual of Case_N Partitioning
S. An SQL Example that explains CASE_N Partitioning
T. How many partitions do you see?
U. Number of PPI Partitions Allowed
V. How many partitions do you see?
W. NO CASE and UNKNOWN Partitions Together
X. A Visual of Case_N Partitioning
Y. Combining Older Data and Newer Data in PPI
Z. A Visual for Combining Older Data and Newer Data in PPI
AA. The SQL on Combining Older Data and Newer Data in PPI
BB. Multi-Level Partitioning Combining Range_N and Case_N
CC. A Visual of Multi-Level Partitioning
DD. The SQL on a Multi-Level Partitioned Primary Index
EE. NON-Unique Primary Indexes (NUPI) in PPI
FF. PPI Table with a Unique Primary Index (UPI)
GG. Tricks for Non-Unique Primary Indexes (NUPI)
HH. Character Based PPI for RANGE_N
II. A Visual for Character-Based PPI for RANGE_N
JJ. The SQL on Character-Based PPI for RANGE_N
KK. Character-Based PPI for CASE_N
LL. Dates and Character-Based Multi-Level PPI
MM. TIMESTAMP Partitioning
NN. Using CURRENT_DATE to define a PPI
OO. ALTER to CURRENT_DATE the next year
PP. ALTER to CURRENT_DATE with Save
QQ. Altering a PPI Table to Add or Drop Partitions
RR. Deleting a Partition
SS. Deleting a Partition and saving its contents
TT. Using the PARTITION Keyword in your SQL
UU. SQL for RANGE_N
VV. SQL for CASE_N

VII. Columnar Tables

A. Columnar Tables have NO Primary Index
B. This is NOT a NoPI Table
C. NoPI Tables Spread rows across all-AMPs Evenly
D. NoPI Tables used as Staging Tables for Data Loads
E. NoPI Table Capabilities
F. NoPI Table Restrictions
G. What does a Columnar Table look like?
H. Comparing Normal Table vs. Columnar Tables
I. Columnar Table Fundamentals
J. Example of Columnar CREATE Statement
K. Columnar can move just One Container to Memory
L. Container on AMPs match up perfectly to rebuild a Row
M. Indexes can be used on Columns (Containers)
N. Indexes can be used on Columns (Containers)
O. Visualize a Columnar Table
P. Single-Column vs. Multi-Column Containers
Q. Comparing Normal Table vs. Columnar Tables
R. Columnar Row Hybrid CREATE Statement
S. Columnar Row Hybrid Example
T. Columnar Row Hybrid Query Example
U. Review of Row-Based Partition Primary Index (PPI)
V. Visual of Row Partitioning (PPI Tables) by Month
W. CREATE Statement for both Row and Column Partition

Teradata Basics

Course Outline (cont'd)

X. Visual of Row Partitioning (PPI Tables) and Columnar
Y. How to Load into a Columnar Table
Z. Columnar NO AUTO COMPRESS
AA. Auto Compress in Columnar Tables
BB. Auto Compress Techniques in Columnar Tables
CC. When and When NOT to use Columnar Tables
DD. Did you know?

VIII. Space
A. When your System Arrives, there is only User named DBC
B. USER DBC
C. First Assignment is to create another User just under DBC
D. USER DBC
E. Perm and Spool Space
F. Perm Space is for Permanent Tables
G. Spool Space is work space that builds a User's Answer Sets
H. Spool Space is in an AMP's Memory and on its Disk
I. Users are Assigned Spool Space Limits
J. What is the Purpose of Spool Limits?
K. Why did my query Abort and say "Out of Spool"?
L. How can Skewed Data cause me to run "Out of Spool"?
M. Why did my Join cause me to run "Out of Spool"?
N. Finding out how much Space you have
O. Space per AMP on all tables in a Database shows Skew
P. What does my system look like when it first arrives?
Q. DBC owns all the PERM Space in the system on day one
R. DBC's First Assignment is Spool Space
S. DBC's 2nd Assignment is to CREATE Users and Databases
T. The Teradata Hierarchy Begins
U. The Teradata Hierarchy Continues
V. Differences between PERM and SPOOL
W. Databases, Users, and Views
X. What are Similarities between a DATABASE and a USER?
Y. What is the Difference between a DATABASE and a USER?
Z. Objects that take up PERM Space
AA. A Series of Quizzes on Adding and Subtracting Space
BB. Answer 1 to Quiz on Space
CC. Space Transfer Quiz
DD. Answer to Space Transfer Quiz
EE. Drop Space Quiz
FF. Answers to Drop Space Quiz

IX. The User Environment
A. DBC is the only user when the system first arrives
B. DBC will Create Databases and Give them Space
C. DBC will create some initial Users
D. A Typical Teradata Environment
E. What are Similarities between a DATABASE and a USER?
F. Roles
G. Create a Role and then Assign that Role Its Access Rights
H. Create a User and Assign them a Default Role
I. Granting Access Rights
J. There are Three Types of Access Rights
K. Description of the Three Types of Access Rights
L. Profiles
M. Creating a Profile and a User
N. ProfileInfoVX, RoleMembers, RoleInfo and UserRoleRights
O. Accounts and their Associated Priorities
P. Creating a User with Multiple Account Priorities
Q. Account String Expansion (ASE)
R. The DBC.AMPUsage View
S. Teradata TASM provides a User Traffic System
T. Teradata Viewpoint

X. Secondary Indexes
A. Creating a Unique Secondary Index (USI)
B. What is in a Unique Secondary Index (USI) Subtable?
C. A Unique Secondary Index (USI) Subtable is hashed
D. How the Parsing Engine uses the USI Subtable
E. A USI is a Two-AMP Operation
F. Creating a Non-Unique Secondary Index (NUSI)
G. What is in a Unique Secondary Index (USI) Subtable?
H. Non-Unique Secondary Index (NUSI) Subtable is AMP Local
I. How the Parsing Engine uses the NUSI Subtable

Teradata Basics

Course Outline (cont'd)

Creating a Value-Ordered NUSI
- J. The Hash Map Determines which AMP will own the Row
- K. A Unique Primary Index Spreads the Data Evenly
- L. Quiz – Answer the Tough USI Questions
- M. Answer to Quiz – Answer the Tough USI Questions
- N. A Picture with a Base Table, USI, and NUSI Subtable
- O. Quiz – Tough Questions on the USI and NUSI Subtables
- P. Answer – Tough Questions on the USI and NUSI Subtables
- Q. A Query Using an USI Only Moves Two Blocks
- R. A Query Using A NUSI Always Uses All AMPs
- S. Two Non-Unique Secondary Indexes (NUSI) on a Table
- T. A NUSI BITMAP Query (1 of 3)
- U. A NUSI BITMAP Theory (2 of 3)
- V. A NUSI Bitmap in Action (3 of 3)
- W. A Brilliant Technique for a Unique Secondary Index
- X. The USI for Partitioned Tables Points to the Row Key
- Y. A Brilliant Technique for a Non-Unique Secondary Index
- Z. The NUSI for Partitioned Tables Points to the Row Key
- AA. How the PE Decides on the NUSI or the Full Table Scan
- BB. The Bigger Quiz
- CC. The Bigger Quiz Answers
- DD. Multiple Choice DBA
- EE. Multiple Choice DBA
- FF. What are the Big Four Tactical Queries?
- GG. What are the Big Four Tactical Queries?

XI. Temporal Tables Create Functions
- A. Three types of Temporal Tables
- B. CREATING a Bi-Temporal Table
- C. PERIOD Data Types
- D. Bi-Temporal Data Type Standards
- E. Bi-Temporal Example – Tera-Tom buys!
- F. A Look at the Temporal Results
- G. Bi-Temporal Example – Tera-Tom Sells!
- H. Bi-Temporal Example – How the data looks!
- I. Normal SQL for Bi-Temporal Tables
- J. NONSEQUENCED SQL for Temporal Tables
- K. AS OF SQL for Temporal Tables
- L. NONSEQUENCED for Both
- M. Creating Views for Temporal Tables
- N. Bi-Temporal Example – Socrates is DELETED!
- O. Bi-Temporal Results – Socrates is DELETED

XII. How Joins Work Internally
- A. Teradata Join Quiz
- B. Teradata Join Quiz Answer
- C. The Joining of Two Tables
- D. Teradata Moves Joining Rows to the Same AMP
- E. Imagine Joining Two NoPI Tables that have No Primary Index
- F. Both Tables are redistributed to Join Rows on the Same AMP
- G. How do you join if One Table is Big and One Table is Small?
- H. Duplicate the Small Table on Every AMP (like a mirror)
- I. What Could You Do If Two Tables Joined 1000 Times a Day?
- J. Joining Two Tables with the same PK/FK Primary Index
- K. A Join with No Redistribution or Duplication
- L. A Performance Tuning Technique for Large Joins
- M. The Joining of Two Tables with an Additional WHERE Clause
- N. An Example of the Fastest Join Possible
- O. Using a Simple Volatile Table
- P. A Volatile Table with a Primary Index
- Q. Using a Simple Global Temporary Table
- R. Two Brilliant Techniques for Global Temporary Tables
- S. The Joining of Two Tables Using a Global Temporary Table
- T. Quiz – How Much Data Moves Across the BYNET?
- U. Answer – How Much Data Moves Across the BYNET?
- V. Teradata V14.10 Join Feature PRPD

XIII. Join Indexes
- A. Creating a Multi-Table Join Index
- B. Visual of a Join Index
- C. Outer Join Multi-Table Join Index
- D. Visual of a Left Outer Join Index
- E. Compressed Multi-Table Join Index
- F. A Visual of a Compressed Multi-Table Join Index
- G. Creating a Single-Table Join Index
- H. Conceptual of a Single Table Join Index on an AMP
- I. Single Table Join Index Great For LIKE Clause

Teradata Basics

Course Outline (cont'd)

J. Single Table Join Index with Value Ordered NUSI
K. Aggregate Join Indexes
L. Compressed Single-Table Join Index
M. Aggregate Join Index
N. New Aggregate Join Index (Teradata V14.10)
O. Sparse Join Index
P. A Global Multi-Table Join Index
Q. Creating a Hash Index
R. Join Index Details

XIV. Collect Statistics

A. The Teradata Parsing Engine (Optimizer) is Cost Based
B. The Purpose of Collect Statistics
C. When Teradata Collects Statistics it creates a Histogram
D. The Interval of the Collect Statistics Histogram
E. Histogram Quiz
F. Answers to Histogram Quiz
G. What to COLLECT STATISTICS On?
H. Why Collect Statistics?
I. How do you know if Statistics were collected on a Table?
J. A Huge Hint that No Statistics Have Been Collected
K. The Basic Syntax for COLLECT STATISTICS
L. COLLECT STATISTICS Examples for a better Understanding
M. The New Teradata V14 Way to Collect Statistics
N. Where Does Teradata Keep the Collected Statistics?
O. The Official Syntax for COLLECT STATISTICS
P. How to Recollect STATISTICS on a Table
Q. Teradata Always Does a Random AMP Sample
R. Random Sample is kept in the Table Header in FSG Cache
S. Multiple Random AMP Samplings
T. How a Random AMP gets a Table Row count
U. Random AMP Estimates for NUSI Secondary Indexes
V. USI Random AMP Samples are Not Considered
W. There's No Random AMP Estimate for Non-Indexed Columns
X. The PE's Plan if No Statistics Were Collected?
Y. Stale Statistics Detection and Extrapolation
Z. Extrapolation for Future Dates
AA. How to Copy a Table with Data and the Statistics?
BB. How to Copy a Table with NO Data and the Statistics?
CC. COLLECT STATISTICS Directly From another Table
DD. When to COLLECT STATISTICS Using only a SAMPLE
EE. Examples of COLLECT STATISTICS Using only a SAMPLE
FF. Examples of COLLECT STATISTICS For V14
GG. How to Collect Statistics on a PPI Table on the Partition
HH. Teradata V12 and V13 Statistics Enhancements
II. Teradata V14 Statistics Enhancements
JJ. Teradata V14 Summary Statistics
KK. Teradata V14 MaxValueLength
LL. Teradata V14 MaxIntervals
MM. Teradata V14 Sample N Percent
NN. Teradata V14.10 Statistics Collection Improvements
OO. Teradata V14.10 Statistics Collection Improvements
PP. Teradata V14.10 AutoStats feature
QQ. Teradata Statistics Wizard

XV. Temporary Tables

A. There are three types of Temporary Tables
B. CREATING A Derived Table
C. Naming the Derived Table
D. Aliasing the Column Names in the Derived Table
E. Most Derived Tables Are Used To Join To Other Tables
F. Multiple Ways to Alias the Columns in a Derived Table
G. Our Join Example with a Different Column Aliasing Style
H. Column Aliasing Can Default for Normal Columns
I. CREATING A Derived Table using the WITH Command
J. Our Join Example With the WITH Syntax
K. The Same Derived Query shown Three Different Ways
L. Quiz - Answer the Questions
M. Answer to Quiz - Answer the Questions
N. Clever Tricks on Aliasing Columns in a Derived Table
O. A Derived Table lives only for the lifetime of a single query
P. An Example of Two Derived Tables in a Single Query
Q. WITH RECURSIVE Derived Table
R. Defining the WITH Recursive Derived Table
S. Looping Through the Recursive Derived Table

Teradata Basics

Course Outline (cont'd)

T. Looping Through a Second Time
U. Looping Through a Third Time
V. Looping Through and Adding Nothing Ends the Loop
W. Looping Through the WITH Recursive Derived Table
X. Creating a Volatile Table
Y. You Populate a Volatile Table with an INSERT/SELECT
Z. The Three Steps to Use a Volatile Table
AA. Why Would You Use the ON COMMIT DELETE ROWS?
BB. The HELP Volatile Table Command Shows your Volatiles
CC. A Volatile Table with a Primary Index
DD. The Joining of Two Tables Using a Volatile Table
EE. You Can Collect Statistics on Volatile Tables
FF. The New Teradata V14 Way to Collect Statistics
GG. Four Examples of Creating a Volatile Table Quickly
HH. Four Advanced Examples of Creating a Volatile Table Quickly
II. Creating Partitioned Primary Index (PPI) Volatile Tables
JJ. Using a Volatile Table to Get Rid of Duplicate Rows
KK. Using a Simple Global Temporary Table
LL. Two Brilliant Techniques for Global Temporary Tables
MM. The Joining of Two Tables Using a Global Temporary Table
NN. CREATING A Global Temporary Table

XVI. Teradata Load Utilities Introduction

A. The Teradata Utilities
B. Block Level Utilities
C. Row Level Utilities
D. Fast Path Inserts Using Insert/Select
E. Fast Path Deletes
F. Freespace Percent
G. Referential Integrity and Load Utility Solutions
H. Teradata has a No Primary Index Table called a NoPI Table
I. This is NOT Necessarily a NoPI Table
J. NoPI Tables Spread rows across all-AMPs Evenly
K. NoPI Tables used as Staging Tables for Data Loads
L. NoPI Table Capabilities
M. NoPI Table Restrictions
N. Why Would a NoPI Table have a Row-ID?
O. BTEQ – Batch Teradata Query Tool
P. How to Logon to BTEQ in Interactive Mode
Q. Runing Queries in BTEQ in Interactive Mode
R. BTEQ Commands vs BTEQ SQL Statements
S. WITH BY Command for Subtotals
T. WITH Command for a Grand Total
U. WITH and WITH BY Together for Subtotals and Grand Totals
V. How to Logon to BTEQ in a SCRIPT
W. Running Queries in BTEQ through a Batch Script
X. Running a BTEQ Batch Script through the Command Prompt
Y. Running a BTEQ Batch Script through the Run Command
Z. Using Nexus to Build Your BTEQ Scripts
AA. Using Nexus to Build Your BTEQ Scripts
BB. FastLoad
CC. Block Level tility Limits
DD. FastLoad has Two Phases
EE. FastLoad Phase 1
FF. FastLoad Phase 2
GG. A Sample FastLoad Script Created by Nexus SmartScript
HH. Executing the FastLoad Script
II. The Nexus SmartScript Easily Builds Your Utilities
JJ. The Nexus SmartScript FastLoad Builder
KK. Create and Execute Your FastLoad Scripts with Nexus
LL. MultiLoad
MM. Block Level Utility Limits
NN. MultiLoad has Five Phases
OO. MultiLoad has IMPORT and DELETE Tasks
PP. A Sample MultiLoad Script Created by Nexus SmartScript
QQ. TPump
RR. TPump is NOT a Block Level Utility and has No Limits
SS. Limitations of TPump
TT. A Sample TPump Script Created by Nexus SmartScript
UU. FastExport
VV. New Rules for Block Utilities
WW. A Sample FastExport Script Created by Nexus SmartScript
XX. FastExport by Default places Null Indicators in Output
YY. A Sample FastExport Script Created by Nexus SmartScript

Teradata Basics

Course Outline (cont'd)

ZZ. What is TPT?
AAA. TPT Producers Create Streams and Consumers Write Them
BBB. The Four Major Operators of TPT
CCC. TPT can read from multiple source files in Parallel
DDD. TPT can have more Operators than Consumers
EEE. TPT Operators and their Equivalent Load Utility
FFF. How to Run a TPT Script

XVII. Top SQL Commands Cheat Sheet

A. SELECT All Columns from a Table and Sort
B. Select Specific Columns and Limiting the Rows
C. Changing your Default Database
D. Keywords that describe you
E. Select TOP Rows in a Rank Order
F. A Sample number of rows
G. Getting a Sample Percentage of rows
H. Find Information about a Database
I. Find information about a Table
J. Using Aggregates
K. Performing a Join
L. Performing a Join using ANSI Syntax
M. Using Date, Time and Timestamp
N. Using Date Functions
O. Using the System Calendar
P. Using the System Calendar in a Query
Q. Formatting Data
R. Using Rank
S. Using a Derived Table
T. Using a Subquery
U. Correlated Subquery
V. Using Substring
W. Basic CASE Statement
X. Advanced CASE Statement
Y. Using an Access Lock in your SQL
Z. Collect Statistics
AA. CREATING a Volatile Table with a Primary Index
BB. CREATING a Volatile Table that is Partitioned (PPI)
CC. CREATING a Volatile Table that is deleted after the Query
DD. Finding the Typical Rows per Value for specific column
EE. Finding out how much Space you have
FF. How much Space you have Per AMP
GG. Finding your Space
HH. Finding Space Skew in Tables in a Database
II. Finding the Number of rows per AMP for a Column
JJ. Finding Account Information
KK. Ordered Analytics