ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Aster Data Basics

Course Summary

Description

In this course, participants will learn the basics of Aster Data with a focus on what's most important from a user's perspective.

Topics

- The Aster Data Architecture
- Fact and Dimension Tables
- How Aster Processes Data
- Four Options for Aster Data Table Design
- How Joins Work Inside the Aster Engine
- Temporary and Analytic Tables
- Aster Modeling Rules
- Tera-Tom's Top Tips
- Indexes
- Aster Windows Functions
- SQL-MapReduce
- Time and Date
- How Rows are Handled
- The Fundamental SQL Commands That Work on Aster

Audience

This course is designed for all users of Aster Data to help them understand the basics of Aster Data.

Prerequisites

There are no prerequisites for this class.

Duration

1-2 days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Aster Data Basics

Course Outline

I. The Aster Data Architecture
- A. What is Parallel Processing?
- B. Aster Data is a Parallel Processing System
- C. Each vworker holds a Portion of Every Table
- D. The Rows of a Table are Spread Across All vworkers
- E. The Aster Data Architecture
- F. The Queen Node
- G. The Worker Node
- H. The Loader Node
- I. The Backup Node
- J. The Aster Architecture Interconnect
- K. Backup and Loader Nodes Do Not use the Interconnect
- L. The Aster Architecture has Spare Nodes
- M. The Aster Architecture Allows Flexibility based on Need
- N. Aster Data Provides Four Fundamental Hardware Strengths
- O. Replication Failover
- P. Data is Compressed on Data Transfers
- Q. Aster Utilizes Dual Optimizers
- R. Aster Allows a Hybrid of SQL and MapReduceMapReduce History
- S. What is MapReduce?
- T. What is SQL-MR?
- U. Sessionize – An Example of SQL-MR
- V. Support for Mixed Workload Management and Prioritization

II. Fact and Dimension Tables
- A. Aster Tables are defined as Fact or Dimension when Created
- B. Fact Table
- C. A More Detailed Look at the Fact Table Distribution
- D. Dimension Table are Replicated
- E. A Dimension Table is often Replicated across vworkers
- F. Aster Data has Fact and Dimension Tables
- G. Aster Tables are defined as Fact or Dimension when Created
- H. Fact and Dimension Tables can be Hashed by the same Key
- I. Distribution Key Rules
- J. Aster Data Uses a Hash Formula
- K. The Hash Map Determines which vworker will own the Row
- L. The Hash Formula, Hash Map and vworker
- M. Placing rows on the vworker
- N. Placing rows on the vworker Continued
- O. A Review of the Hashing Process
- P. Like Data Hashes to the Same vworker
- Q. Distribution Key Data Types
- R. Run ANALYZE to COLLECT STATISTICS on a Table
- S. Some Examples of ANALYZE
- T. What Columns to Analyze

III. How Aster Processes Data
- A. When a Table is Created, a Table Header is Created
- B. Every vworker has the Exact Same Tables
- C. All Aster Tables are spread across All vworkers
- D. The Table Header and the Data Rows are Stored Separately
- E. A vworker Stores the Rows of a Table inside a Data Block
- F. To Read Rows, a vworker Moves the Data Block into Memory
- G. A Full Table Scan Means All vworkers must Read All Rows
- H. The “Achilles Heel”, or Slowest Process, is Block Transfer
- I. Each Table has a Distribution Key
- J. A Query Using the Distribution Key uses a Single vworker.
- K. As Rows are Added, a Data Block will Eventually Split
- L. A Full Table Scan Means All vworkers Read All Blocks
- M. Distribution Key Query uses One vworker
- N. Each vworker Can Have Many Blocks for a Single Table
- O. A Full Table Scan Means All vworkers Read All Blocks
- P. Quiz – How Many Blocks Move into vworker Memory?
- Q. Answer – How Many Blocks Move into vworker Memory?
- R. Quiz – How Many Blocks Move Using the Distribution Key?
- S. Answer-How Many Blocks Move Using the Distribution Key?

Aster Data Basics

Course Outline (cont'd)

IV. Four Options for Aster Data Table Design
- A. There are Four Options to Aster Table Design
- B. Straight up Distribute by Hash
- C. Straight up Distribute by Hash - Problems
- D. Straight up Distribute by Replication
- E. Partition the Table with Logical Partitioning
- F. This Partitioned Table Sorts Rows by Month of Order_Date
- G. An All vworkers Retrieve By Way of a Single Partition
- H. You can Partition a Table by Range or by List
- I. A Partitioned By List Example with Three Tactical Queries
- J. Aster Data Multi-Level Partitioning
- K. Aster Allows for Multi-Level Partitioning
- L. SQL Commands for Logical Partitioning as One Table
- M. What Partitions are on my Table?
- N. What does a Columnar Table look like?
- O. A Comparison of Data for Normal Vs. Columnar
- P. A Columnar Table is best for Queries with Few Columns
- Q. Quiz – How Many Blocks Move to vworker Memory?
- R. Answer – How Many Containers Move to vworker Memory?
- S. When to use a Columnar Table

V. How Joins Work Inside the Aster Engine
- A. Aster Join Quiz
- B. Aster Join Quiz Answer
- C. The Joining of Two Tables
- D. Aster Moves Joining Rows to the Same vworker
- E. Because of the Join Rule – Dimension Table are Replicated
- F. The Two Different Philosophies for Table Join Design
- G. What Could You Do If Two Tables Joined 1000 Times a Day?
- H. Fact and Dimension Tables can be Hashed by the same Key
- I. Joining Two Tables with the same PK/FK Distribution Key
- J. A Join With Co-Location
- K. A Performance Tuning Technique for Large Joins
- L. The Joining of Two Tables with an Additional WHERE Clause
- M. Aster Performs Joins Using Three Different Methods
- N. The Hash Join
- O. The Merge Join
- P. Nested Loop Joins

VI. Temporary and Analytic Tables
- A. Aster has Three Types of Data
- B. Create a Permanent Table Using Create Table AS (CTAS)
- C. Create a Logically Partitioned Table and Populate It
- D. Create a Temporary Table with using Create Table AS (CTAS)
- E. A Temporary Table in Action
- F. A Temporary Table That Uses an Insert/SelectCreate an Analytic Table Using an Insert/Select
- G. Create an Analytic Table Using CREATE TABLE AS (CTAS)
- H. Operations that Invalidate an Analytic Table
- I. If an Analytic Table is Invalid
- J. Tera-Tom History

VII. Aster Modeling Rules
- A. Modeling Rules for Aster Data
- B. Three Principles that Govern the Modeling Rules
- C. Modeling Rule 1 – Dimensionalize your Model
- D. A Dimensional Model is called a "Star Schema"
- E. To Read a Data Block, a vworker Moves the Block to Memory
- F. A Dimensional Model Moves Less Mass into Memory
- G. Which Move From Disk to Memory Would You Choose?
- H. Vworkers transfer their Fact Table into Memory in Parallel
- I. Modeling Rule 2 – Use Columnar
- J. Which Move From Disk to Memory Would You Choose?
- K. Let's Discuss Modeling and Joins at the Simplest Level
- L. Let's Discuss Modeling and Joins at the Simplest Level
- M. Let's Discuss Joins at the Simplest Level
- N. Modeling Rule 3 – Distribute your Tables Based on Joins
- O. The Two Different Philosophies for Table Join Design

Aster Data Basics

Course Outline (cont'd)

P. Facts are Hashed and most often the Dimension is Replicated
Q. Fact and Dimension Tables can be Hashed by the same Key
R. Joining Two Tables with the same PK/FK Primary Index
S. A Join With No Redistribution or Duplication
T. Aster Hates Joining Tables with a Different Distribution Key
U. Aster Hates to Redistribute by Hash to Join Tables
V. Modeling Rule 4 – Replicate Dimension Tables
W. Modeling Rule 5 – Partition Your Tables
X. Modeling Rule 6 – Make Fact Tables Skinny
Y. Modeling Rule 6 – Make Fact Tables Skinny Example
Z. Modeling Rule 7 – Index Your Tables
AA. The B-Tree Index
BB. Which Columns Might You Create an Index?
CC. Answer - Which Columns Might You Create an Index?
DD. Modeling Rule 8 – Denormalize based on Your Environment.
EE. Modeling Rule 8 – Denormalize based on Your Environment.

VIII. Tera-Tom's Top Tips

A. Tera-Tom's Top Tips
B. Tera-Tom's Top Tips # 2
C. Tera-Tom's Top Tips #3
D. Tera-Tom's Top Tips # 3 Rewritten
E. Tera-Tom's Top Tips #4
F. When the GROUP BY Column is NOT the Distribution Key
G. Example of GROUP BY Column is NOT the Distribution Key
H. Tera-Tom's Top Tips #5
I. Tera-Tom's Top Tips #6 – Use EXPLAIN
J. Query Plan and Estimates
K. Explain Plan Showing a Hash Join
L. Explain Plan Showing a Merge Join
M. Explain Plan Showing a Nested Loop Join

IX. Indexes

A. There are Only Three Types of Scans
B. Guidelines for Indexes
C. An Index Syntax Example
D. The B-Tree Index
E. Which Columns Might You Create an Index?
F. Answer - Which Columns Might You Create an Index?
G. A Visual of an Index (Conceptually)
H. A Query Using an Index Uses All vworkers
I. Multicolumn indexes
J. A NUSI BITMAP Theory
K. A NUSI Bitmap in Action
L. Indexes on Expressions
M. Indexes on Extracts of Dates
N. GiST Indexes
O. Five Operational Tips for Efficient Indexing
P. REINDEX
Q. createCompressedIndexOnCompressedTableBy Default Flag

X. Aster Windows Functions

A. Cumulative Sum
B. Cumulative Sum - Major and Minor Sort Key(s)
C. The ANSI CSUM – Getting a Sequential Number
D. The ANSI OLAP – Reset with a PARTITION BY Statement
E. PARTITION BY only Resets a Single OLAP not ALL of them
F. ANSI Moving Sum is Current Row and Preceding n Rows
G. How ANSI Moving SUM Handles the Sort
H. Quiz – How is that Total Calculated?
I. Answer to Quiz – How is that Total Calculated?
J. Moving SUM every 3-rows vs. a Continuous Sum
K. Moving Average
L. Quiz – How is that Total Calculated?
M. Answer to Quiz – How is that Total Calculated?
N. Quiz – How is that 4th Row Calculated?
O. Answer to Quiz – How is that 4th Row Calculated?
P. Partition By Resets an ANSI OLAP
Q. Moving Average Using BETWEEN
R. Moving Difference using ANSI Syntax
S. Moving Difference using ANSI Syntax with Partition By
T. RANK Defaults to Ascending Order
U. Getting RANK to Sort in DESC Order
V. You can use Window Functions in Expressions
W. RANK() OVER and PARTITION BY
X. DENSE_RANK() OVER
Y. PERCENT_RANK() OVER
Z. PERCENT_RANK() OVER with 14 rows in Calculation

Aster Data Basics

Course Outline (cont'd)

AA. PERCENT_RANK() OVER with 21 rows in Calculation
BB. RANK With ORDER BY SUM()
CC. COUNT OVER for a Sequential Number
DD. Quiz – What caused the COUNT OVER to Reset?
EE. Answer to Quiz – What caused the COUNT OVER to Reset?
FF. The MAX OVER Command
GG. MAX OVER with PARTITION BY Reset
HH. The MIN OVER Command
II. Quiz – Fill in the Blank
JJ. Answer to Quiz – Fill in the Blank
KK. The Row_Number Command
LL. Quiz – How did the Row_Number Reset?
MM. Quiz – How did the Row_Number Reset?
NN. NTILE
OO. NTILE Using a Value of 10
PP. NTILE With a Partition
QQ. CUME_DIST
RR. CUME_DIST With a Partition
SS. LEAD
TT. LEAD With Partitioning
UU. LAG
VV. LAG with Partitioning
WW. FIRST_VALUE
XX. FIRST_VALUE After Sorting by the Highest Value
YY. FIRST_VALUE with Partitioning
ZZ. LAST_VALUE
AAA. NTH_VALUE
BBB. NTH_VALUE With Partition
CCC. SUM(SUM(n))

XI. SQL-MapReduce

A. MapReduce History
B. What is MapReduce?
C. What is SQL-MapReduce?
D. SQL-MapReduce Input
E. SQL-MapReduce Output
F. Subtle SQL-MapReduce Processing
G. Aster Data Provides an Analytic Foundation
H. Path Analysis
I. Text Analysis
J. Statistical Analysis
K. Segmentation (Data Mining)
L. Graph Analysis
M. Transformation of Data
N. Sessionize
O. Tokenize
P. SQL-MapReduce Function… nPath
Q. nPath SELECT Clause
R. nPath ON Clause
S. nPath PARTITION BY Expression
T. nPath DIMENSION Expression
U. nPath ORDER BY Expression
V. nPath MODE Clause has Overlapping or NonOverlapping
W. nPath PATTERN Clause
X. Pattern Operators
Y. Pattern Operators Order of Precedence
Z. Matching Patterns Which Repeat
AA. nPath SYMBOLS Clause
BB. nPath RESULTS Clause
CC. Adding an Aggregate to nPath Results
DD. Adding an Aggregate to nPath Results (Continued)
EE. SQL-MapReduce Examples - Use Regular SQL
FF. SQL-MapReduce Examples - Create Objects
GG. SQL-MapReduce Examples - Subquery
HH. SQL-MapReduce Examples - Query as Input
II. SQL-MapReduce Examples - Nesting Functions
JJ. SQL-MapReduce Examples - Functions in Derived Tables
KK. SQL-MapReduce Examples - SMAVG
LL. SQL-MapReduce Examples - Pack Functio
MM. SQL-MapReduce Examples - Pack Function (Continued)
NN. SQL-MapReduce Examples - Pivot Columns

XII. Time and Date

A. Date, Time, and Timestamp Keywords
B. Add or Subtract Days from a date
C. The to_char command
D. A Summary of Math Operations on Date
E. Using a Math Operation to find your Age in Years
F. Find What Day of the week you were Born
G. Date Related Functions
H. The EXTRACT Command
I. EXTRACT from DATES and TIME
J. EXTRACT with DATE and TIME Literals
K. EXTRACT of the Month on Aggregate Queries
L. A Side Title example with Reserved Words as an Alias
M. Implied Extract of Day, Month and Year
N. DATE_PART Function
O. DATE_TRUNC Function
P. DATE_TRUNC Function using TIME
Q. Aster NOW() Function

Aster Data Basics

Course Outline (cont'd)

XIII. How Rows are Handled

A. Aster Rowid, CreateXid, DeleteXid
B. An Update of Multiple Rows
C. A Delete Example
D. A Delete Example Query
E. An Insert Example
F. An Insert Example Query

XIV. The Fundamental SQL Commands That Work on Aster

A. BETWEEN is Inclusive
B. BETWEEN Works for Character Data
C. LIKE uses Wildcards Percent '%' and Underscore '_'
D. LIKE command Underscore is Wildcard for one Character
E. GROUP BY Vs. DISTINCT – Good Advice
F. The Five Aggregates of Aster Data
G. GROUP BY when Aggregates and Normal Columns Mix
H. GROUP BY Delivers one row per Group
I. GROUP BY Dept_No or GROUP BY 1 the same thing
J. Limiting Rows and Improving Performance with WHERE
K. WHERE Clause in Aggregation limits unneeded Calculations
L. Keyword HAVING tests Aggregates after they are Totaled
M. Keyword HAVING is like an Extra WHERE Clause for Totals
N. Getting the Average Values per Column
O. Getting the Average Values per Column
P. Average Values per Column for All Columns in a Table
Q. A two-table join using Non-ANSI Syntax
R. A two-table join using Non-ANSI Syntax with Table Alias
S. Aliases and Fully Qualifying Columns
T. A two-table join using ANSI Syntax
U. Both Queries have the same Results and Performance
V. Quiz – Can You Finish the Join Syntax?
W. Answer to Quiz – Can You Finish the Join Syntax?
X. Quiz – Can You Find the Error?
Y. Answer to Quiz – Can You Find the Error?
Z. Quiz – Which rows from both tables Won't Return?
AA. Answer to Quiz – Which rows from both tables Won't Return?
BB. LEFT OUTER JOIN
CC. LEFT OUTER JOIN Brings Back All Rows in the Left Table
DD. RIGHT OUTER JOIN
EE. RIGHT OUTER JOIN Brings Back All Rows in the RIGHT Table
FF. FULL OUTER JOIN
GG. FULL OUTER JOIN Brings Back All Rows in All Tables
HH. Which Tables are the Left and which are the Right?
II. Answer - Which Tables are the Left and which are the Right?
JJ. INNER JOIN with Additional AND Clause
KK. ANSI INNER JOIN with Additional AND Clause
LL. ANSI INNER JOIN with Additional WHERE Clause
MM. OUTER JOIN with Additional WHERE Clause
NN. OUTER JOIN with Additional AND Clause
OO. Results from OUTER JOIN with Additional AND Clause
PP. Quiz – Why is this considered an INNER JOIN?
QQ. The DREADED Product Join 357
RR. Result Set of the DREADED Product Join
SS. The Horrifying Cartesian Product Join
TT. The ANSI Cartesian Join will ERROR
UU. Quiz – Do these Joins Return the Same Answer Set?
VV. Answer – Do these Joins Return the Same Answer Set?
WW. How would you Join these two tables?
XX. How would you Join these two tables? You Can't Yet!
YY. An Associative Table is a Bridge that Joins Two Tables
ZZ. Quiz – Can you Write the 3-Table Join?
AAA. Answer to Quiz – Can you Write the 3-Table Join?
BBB. Quiz – Can you Write the 3-Table Join to ANSI Syntax?
CCC. Answer – Can you Write the 3-Table Join to ANSI Syntax?
DDD. Quiz – Can you Place the ON Clauses at the End?
EEE. Answer – Can you Place the ON Clauses at the End?

Aster Data Basics

Course Outline (cont'd)

FFF. The 5-Table Join – Logical Insurance Model
GGG. Quiz - Write a Five Table Join Using ANSI Syntax
HHH. Answer - Write a Five Table Join Using ANSI Syntax
III. Quiz - Write a Five Table Join Using ANSI Syntax

JJJ. Answer - Write a Five Table Join Using ANSI Syntax
KKK. Quiz - Write a Five Table Join Using Non-ANSI Syntax
LLL. Answer - Write a Five Table Join Using Non-ANSI Syntax
MMM. Quiz –Re-Write this putting the ON clauses at the END
NNN. Answer –Re-Write this putting the ON clauses at the END
OOO. The Nexus Query Chameleon Writes the SQL for Users.
PPP. An IN List is much like a Subquery
QQQ. An IN List Never has Duplicates – Just like a Subquery
RRR. An IN List Ignores Duplicates
SSS. The Subquery
TTT. How a Basic Subquery Works
UUU. The Final Answer Set from the Subquery
VVV. Quiz- Answer the Difficult Question
WWW. Answer to Quiz- Answer the Difficult Question
XXX. Should you use a Subquery or a Join?

YYY. Quiz- Write the Subquery
ZZZ. Answer to Quiz- Write the Subquery
AAAA. Quiz- Write the More Difficult Subquery

BBBB. Answer to Quiz- Write the More Difficult Subquery
CCCC. Quiz- Write the Subquery with an Aggregate
DDDD. Answer to Quiz- Write the Subquery with an Aggregate
EEEE. Quiz – Write the Triple Subquery
FFFF. Answer to Quiz – Write the Triple Subquery
GGGG. CHARACTER_LENGTH AND OCTET_LENGTH

HHHH. The TRIM Command trims both Leading and Trailing Spaces
IIII. Trim and Trailing is Case Sensitive
JJJJ. Trim and Trailing works if Case right
KKKK. The SUBSTRING Command
LLLL. How SUBSTRING Works with NO ENDING POSITION
MMMM. Using SUBSTRING to move Backwards

NNNN. How SUBSTRING Works with a Starting Position of -1
OOOO. How SUBSTRING Works with an Ending Position of 0
PPPP. An Example using SUBSTRING, TRIM and CHAR Together
QQQQ. SUBSTRING and SUBSTR are equal, but use different syntax
RRRR. The POSITION Command finds a Letters Position
SSSS. Concatenation
TTTT. The Basics of CAST (Convert and Store)
UUUU. Some Great CAST (Convert and Store) Examples
VVVV. Some Great CAST (Convert and Store) Examples
WWWW. Combining Searched Case and Valued Case
XXXX. A Trick for getting a Horizontal Case
YYYY. Nested Case
ZZZZ. Put a CASE in the ORDER BY