

Course duration

- 3 days

Course Benefits

- Learn to gain a deeper knowledge and understanding of the Greenplum Architecture and SQL and how to write it.

Course Outline

1. Introduction to the Greenplum Architecture
 1. What is Parallel Processing?
 2. The Basics of a Single Computer
 3. Data in Memory is Fast as Lightning
 4. Parallel Processing Of Data
 5. Symmetric Multi-Processing (SMP) Server
 6. Commodity Hardware Servers are Configured for Greenplum
 7. Commodity Hardware Allows For One Segment Per CPU
 8. The Master Host
 9. The Segment's Responsibilities
 10. The Host's Plan is Either All Segments or a Single Segment
 11. A Table has Columns and Rows
 12. Greenplum has Linear Scalability
 13. The Architecture of A Greenplum Data Warehouse
 14. Nexus is Now Available For Greenplum
2. Greenplum Table Structures
 1. The Concepts of Greenplum Tables
 2. Tables are Either Distributed by Hash or Random
 3. A Hash Distributed Table has A Distribution Key
 4. Picking A Distribution Key That Is Not Very Unique
 5. Random Distribution Uses a Round Robin Technique
 6. Tables Will Be Distributed Among All Segments
 7. The Default For Distribution Chooses the First Column
 8. Table are Either a Heap or Append-Only
 9. Tables are Stored in Either Row or Columnar Format
 10. Creating a Column Oriented Table
 11. Comparing Normal Table Vs. Columnar Tables
 12. Columnar can move just One Column Block Into Memory
 13. Segments on Distributions are Aligned to Rebuild a Row
 14. Columnar Tables Store Each Column in Separate Blocks
 15. Visualize the Data - Rows vs. Columns

16. Table Rows are Either Sorted or Unsorted
17. Creating a Clustered Index in Order to Physically Sort Rows
18. Physically Ordered Tables Are Faster on Certain Queries
19. Another Way to Create a Clustered Table
20. Creating a B-Tree Index and then Running Analyze
21. Creating a Bitmap Index
22. Why Create a Bitmap Index?
23. Tables Can Be Partitioned
24. A Table Partitioned By Range (Per Month)
25. A Visual of a Partitioned Table by Range (Month)
26. Tables Can Be Partitioned by Day
27. Visualize a Partitioned Table by Day
28. Creating a Partitioned Table Using a List
29. Creating a Multi-Level Partitioned Table
30. Changing a Table to a Partitioned Table
31. Not Null Constraints
32. Unique Constraints
33. Unique Constraints That Fail
34. Primary Key Constraints
35. A Primary Key Automatically Creates a Unique Index
36. Check Constraints
37. Creating an Automatic Number Called a Sequence
38. Multiple INSERT example Using a Sequence
3. Hashing and Data Distribution
 1. Distribution Keys Hashed on Unique Values Spread Evenly
 2. Distribution Keys With Non-Unique Values Spread Unevenly
 3. Best Practices for Choosing a Distribution Key
 4. The Hash Maps
 5. A Review of the Hashing Process
 6. Non-Unique Distribution Keys have Skewed Data
4. The Technical Details
 1. Greenplum Limitations
 2. Every Segment has the Exact Same Tables
 3. Tables are Distributed Across All Segments
 4. The Table Header and the Data Rows are Stored Separately
 5. Segments Store Rows inside a Data Block Called a Page
 6. To Read a Data Block a Node Moves the Block into Memory
 7. A Full Table Scan Means All Nodes Must Read All Rows
 8. Rows are Organized inside a Page
 9. Moving Data Blocks is Like Checking In Luggage
 10. As Row-Based Tables Get Bigger, the Page Splits
 11. Data Pages are Processed One at a Time Per Unit
 12. Creating a Table that is a Heap
 13. Heap Page
 14. Creating a Table that has a Clustered Index
 15. Clustered Index Page
 16. The Row Offset Array is the Guidance System for Every Row

17. The Building of a B-Tree for a Clustered Index
18. When Do I Create a Non Clustered Index?
19. B-Tree for Non Clustered Index on a Clustered Table
20. Adding a Non Clustered Index To A
21. B-Tree for Non Clustered Index on a Heap Table
5. Physical Database Design
 1. The Four Stages of Modeling for Greenplum- Check out #4
 2. The Logical Model
 3. First, Second and Third Normal Form
 4. The Employee_Table and Department_Table can be Joined
 5. The Employee_Table and Department_Table Join SQL
 6. The Extended Logical Model Template
 7. User Access is of Great Importance
 8. User Access in Layman's Terms
 9. User Access for Joins in Layman's Terms
 10. The Nexus Shows Users the Table's Distribution Key
 11. Data Demographics Tell Us if the Column is Worthy
 12. Data Demographics
 13. Typical Rows Per Value Query For Greenplum Systems
 14. SQL to Get the Average Rows Per Value for a Column (Mean)
 15. Data Demographics - Change Rating
 16. Factors When Choosing Greenplum Indexes
 17. Distribution Key Data Demographics Candidate Guidelines
 18. Distribution key Access Considerations
 19. Choose the Distribution Key and Secondary Indexes
 20. Our Index Picks
6. Denormalization
 1. Denormalization
 2. Derived Data
 3. Repeating Groups
 4. Pre-Joining Tables
 5. Storing Summary Data with a Trigger
 6. Summary Tables or Data Marts the Old Way
 7. Horizontal Partitioning
 8. Vertical Partitioning the Old Way
 9. Columnar Tables Are the New Vertical Partitioning
7. Nexus for Greenplum
 1. Nexus is Available on the Cloud
 2. Nexus Queries Every Major System
 3. Setup of Nexus is as Easy as Pie
 4. Setup of Nexus is as Easy as 1, 2, 3
 5. Nexus Data Visualization
 6. Nexus Data Visualization Shows What Tables Can Be Joined
 7. Nexus is Doing a Five-Table Join
 8. Nexus Generates the SQL Automatically
 9. Nexus Delivers the Report
 10. Cross-System Joins From Teradata, Oracle and SQL Server

11. The Tabs of the Super Join Builder
 12. The 9 Tabs of the Super Join Builder - Objects Tab 1
 13. Selecting Columns in the Objects Tab
 14. The 9 Tabs of the Super Join Builder
 15. Using the WHERE Tab For Additional WHERE or AND
 16. Analytics Tab
 17. Nexus Data Movement
 18. Moving a Single Table To a Different System
 19. The Single Table Data Movement Screen
 20. Moving an Entire Database To a Different System
 21. The Database Mover Screen
 22. The Database Mover Options Tab
 23. Converting DDL Table Structures
 24. Compare and Synchronize
 25. Compare Two Different Databases From Different Systems
 26. Comparisons Down to the Column Level
 27. The Results Tab
 28. View Differences
 29. Synchronizing Differences In the Results Tab
 30. Synchronizing Differences In the Results Tab
 31. Hound Dog Compression
8. The Basics of SQL
 1. Introduction
 2. SELECT * (All Columns) in a Table
 3. Fully Qualifying a Database, Schema and Table
 4. SELECT Specific Columns in a Table
 5. Commas in the Front or Back?
 6. Place your Commas in front for better Debugging Capabilities
 7. Sort the Data with the ORDER BY Keyword
 8. ORDER BY Defaults to Ascending
 9. Use the Name or the Number in your ORDER BY Statement
 10. Two Examples of ORDER BY using Different Techniques
 11. Changing the ORDER BY to Descending Order
 12. NULL Values sort First in Ascending Mode (Default)
 13. NULL Values sort Last in Descending Mode (DESC)
 14. Major Sort vs. Minor Sorts
 15. Multiple Sort Keys using Names vs. Numbers
 16. Sorts are Alphabetical, NOT Logical
 17. Using A CASE Statement to Sort Logically
 18. How to ALIAS a Column Name
 19. A Missing Comma can by Mistake become an Alias
 20. Comments using Double Dashes are Single Line Comments
 21. Comments for Multi-Lines
 22. Comments for Multi-Lines As Double Dashes Per Line
 23. A Great Technique for Comments to Look for SQL Errors
 9. The WHERE Clause
 1. The WHERE Clause limits Returning Rows

2. Double Quoted Aliases are for Reserved Words and Spaces
 3. Character Data needs Single Quotes in the WHERE Clause
 4. Character Data needs Single Quotes, but Numbers Don't
 5. Comparisons against a Null Value
 6. NULL means UNKNOWN DATA so Equal (=) won't Work
 7. Use IS NULL or IS NOT NULL when dealing with NULLs
 8. NULL is UNKNOWN DATA so NOT Equal won't Work
 9. Use IS NULL or IS NOT NULL when dealing with NULLs
 10. Using Greater Than or Equal To (>=)
 11. AND in the WHERE Clause
 12. Troubleshooting AND
 13. OR in the WHERE Clause
 14. Troubleshooting Or
 15. Troubleshooting Character Data
 16. Using Different Columns in an AND Statement
 17. What is the Order of Precedence?
 18. Using Parentheses to change the Order of Precedence
 19. Using an IN List in place of OR
 20. The IN List is an Excellent Technique
 21. IN List vs. OR brings the same Results
 22. The IN List Can Use Character Data
 23. Using a NOT IN List
 24. Null Values in a NOT IN List Bring Back No Rows
 25. A Technique for Handling Nulls with a NOT IN List
 26. BETWEEN is Inclusive
 27. NOT BETWEEN is Also Inclusive
 28. LIKE uses Wildcards Percent '%' and Underscore '_'
 29. LIKE command Underscore is Wildcard for one Character
 30. ilike
 31. LIKE Command Works Differently on Char Vs Varchar
 32. Troubleshooting LIKE Command on Character Data
 33. Introducing the TRIM Command
 34. Introducing the RTRIM Command
 35. Numbers are Right Justified and Character Data is Left
 36. Answer - What Data is Left Justified and What is Right?
 37. An example of Data with Left and Right Justification
 38. A Visual of CHARACTER Data vs. VARCHAR Data
 39. Use the TRIM command to remove spaces on CHAR Data
 40. Escape Character in the LIKE Command changes Wildcards
 41. Escape Characters Turn off Wildcards in the LIKE Command
 42. Introducing the RTRIM Command
 43. An example of Data with Left and Right Justification
 44. A Visual of CHARACTER Data vs. VARCHAR Data
 45. RTRIM command Removes Trailing spaces on CHAR Data
 46. Using Like with an AND Clause to Find Multiple Letters
 47. Using Like with an OR Clause to Find Either Letters
10. Distinct vs. Group By

1. The Distinct Command
2. Distinct vs. GROUP BY

11. Aggregation

1. The 3 Rules of Aggregation
2. There are Five Aggregates
3. Troubleshooting Aggregates
4. GROUP BY when Aggregates and Normal Columns Mix
5. GROUP BY delivers one row per Group
6. GROUP BY Dept_No or GROUP BY 1 the same thing
7. Limiting Rows and Improving Performance with WHERE
8. WHERE Clause in Aggregation limits unneeded Calculations
9. Keyword HAVING tests Aggregates after they are Totaled
10. Aggregates Return Null on Empty Tables
11. Keyword HAVING is like an Extra WHERE Clause for Totals
12. Keyword HAVING tests Aggregates after they are Totaled
13. Getting the Average Values Per Column
14. Average Values Per Column For all Columns in a Table
15. Three types of Advanced Grouping
16. Group By Grouping Sets
17. Group By Rollup
18. GROUP BY

12. Join Functions

1. Redistribution
2. Big Table Small Table Join Strategy
3. Duplication of the Smaller Table across All-Distributions
4. If the Join Condition is the Distribution Key no Movement
5. Matching Rows That Are On The Same Node Naturally
6. Strategy 1 of 4 - The Merge Join
7. Strategy 2 of 4 - The Hash Join
8. Strategy 3 of 4 - The Nested Join
9. Strategy 4 of 4 - The Product Join
10. A Two-Table Join Using Traditional Syntax
11. A two-table join using Non-ANSI Syntax with Table Alias
12. You Can Fully Qualify All Columns
13. A two-table join using ANSI Syntax
14. Both Queries have the same Results and Performance
15. LEFT OUTER JOIN
16. LEFT OUTER JOIN Results
17. RIGHT OUTER JOIN
18. RIGHT OUTER JOIN Example and Results
19. FULL OUTER JOIN
20. FULL OUTER JOIN Results
21. Which Tables are the Left and which Tables are Right?
22. INNER JOIN with Additional AND Clause
23. ANSI INNER JOIN with Additional AND Clause
24. ANSI INNER JOIN with Additional WHERE Clause
25. OUTER JOIN with Additional WHERE Clause

26. OUTER JOIN with Additional AND Clause
 27. OUTER JOIN with Additional AND Clause Results
 28. Evaluation Order for Outer Queries
 29. The DREADED Product Join
 30. The DREADED Product Join Results
 31. The Horrifying Cartesian Product Join
 32. The ANSI Cartesian Join will ERROR
 33. Answer - Do these Joins Return the Same Answer Set?
 34. The CROSS JOIN
 35. The CROSS JOIN Answer Set
 36. The SelfJoin
 37. The SelfJoin with ANSI Syntax
 38. The Nexus Query Chameleon Writes the SQL for Users
13. Date Function
1. Current_Date
 2. Current_Date, Current_Time, and Current_Timestamp
 3. Current_Time vs. LocalTime With Precision
 4. Local_Time and Local_Timestamp With Precision
 5. Now() and Timeofday() Functions
 6. Adding A Week to a Date
 7. Add or Subtract Days from a date
 8. Formatting Dates and Dollar Amounts
 9. The EXTRACT Command
 10. EXTRACT from DATES and TIME
 11. EXTRACT Command on the Century
 12. EXTRACT Command for the Decade, DOW and DOY
 13. EXTRACT Microseconds, Milliseconds and Millennium
 14. EXTRACT of the Month on Aggregate Queries
 15. Date_part Command
 16. Date_Trunc Command With Time
 17. Date_Trunc Command With Dates
 18. The AGE Command
 19. AGE Challenge
 20. AGE Challenge Results
 21. Epoch
 22. Using Intervals
 23. More Interval Examples
 24. Interval Arithmetic Results
 25. A Complex Time Interval example using CAST
 26. The OVERLAPS Command
 27. An OVERLAPS example that Returns No Rows
 28. The OVERLAPS Command using TIME
 29. Using Both CAST and CONVERT in Literal Values
 30. A Better Technique for YEAR, MONTH, and DAY Functions
14. Conversions and Formatting
1. Postgres Conversion Functions
 2. Postgres Conversion Function Templates

3. Postgres Conversion Function Templates Continued
4. To_Char command Examples
5. Formatting A Date with To_Char
6. Formatting A Date With To_Char Continued
7. To_Number
8. To_Number Examples
9. To_Date
10. To_Timestamp
15. Sub-query Functions
 1. An IN List is much like a Subquery
 2. An IN List Never has Duplicates - Just like a Subquery
 3. An IN List Ignores Duplicates
 4. The Subquery
 5. The Three Steps of How a Basic Subquery Works
 6. These are Equivalent Queries
 7. The Final Answer Set from the Subquery
 8. Should you use a Subquery of a Join?
 9. The Basics of a Correlated Subquery
 10. The Top Query always runs first in a Correlated Subquery
 11. Correlated Subquery Example vs. a Join with a Derived Table
 12. How to handle a NOT IN with PotentialNULL Values
 13. IN is equivalent to =ANY
 14. Using a Correlated Exists
 15. How a Correlated Exists matches up
 16. The Correlated NOT Exists
 17. The Correlated NOT Exists Answer Set
16. OLAP Functions
 1. CSUM
 2. The ANSI CSUM - Getting a Sequential Number
 3. Troubleshooting The ANSI OLAP on a GROUP BY
 4. Reset with a PARTITION BY Statement
 5. PARTITION BY only Resets a Single OLAP not ALL of them
 6. Moving SUM
 7. ANSI Moving Window is Current Row and Preceding n Rows
 8. How ANSI Moving SUM Handles the Sort
 9. Moving SUM every 3-rows Vs a Continuous Average
 10. Partition By Resets an ANSI OLAP
 11. Both the Greenplum Moving Average and ANSI Version
 12. Moving Average
 13. The Moving Window is Current Row and Preceding
 14. How Moving Average Handles the Sort
 15. Moving Average every 3-rows Vs a Continuous Average
 16. Partition By Resets an ANSI OLAP
 17. Moving Difference using ANSI Syntax with Partition By
 18. RANK Defaults to Ascending Order
 19. Getting RANK to Sort in DESC Order
 20. RANK() OVER and PARTITION BY

21. RANK and DENSE RANK
 22. PERCENT_RANK() OVER
 23. COUNT OVER for a Sequential Number
 24. Troubleshooting COUNT OVER
 25. The MAX OVER Command
 26. MAX OVERwith PARTITION BY Reset
 27. Troubleshooting MAX OVER
 28. The MIN OVER Command
 29. Troubleshooting MIN OVER
 30. Finding a Value of a Column in the Next Row with MIN
 31. The Row_Number Command
 32. Using a Derived Table and Row_Number
 33. Ordered Analytics OVER
 34. CURRENT ROW AND UNBOUNDED FOLLOWING
 35. Different Windowing Options
 36. The CSUM For Each Product_Id and the Next Start Date
 37. How Ntile Works
 38. Ntile
 39. Ntile Continued
 40. Ntile Percentile
 41. Another Ntile example
 42. Using Tertiles (Partitions of Four)
 43. NTILE
 44. NTILE Using a Value of 10
 45. NTILE With a Partition
 46. Using FIRST_VALUE
 47. FIRST_VALUE
 48. FIRST_VALUE After Sorting by the Highest Value
 49. FIRST_VALUE with Partitioning
 50. Using LAST_VALUE
 51. LAST_VALUE
 52. Using LEAD
 53. Using LEAD With and Offset of 2
 54. LEAD
 55. LEAD With Partitioning
 56. Using LAG
 57. Using LAG With an Offset of 2
 58. LAG
 59. LAG with Partitioning
 60. CUME_DIST
 61. CUME_DIST With a Partition
 62. SUM(SUM(n))
17. Temporary Tables
 1. There are Two Types of Temporary Tables
 2. CREATING A Derived Table
 3. Naming the Derived Table
 4. Aliasing the Column Names in The Derived Table

5. Multiple Ways to Alias the Columns in a Derived Table
6. CREATING A Derived Table using the WITH Command
7. The Same Derived Query shown Three Different Ways
8. Most Derived Tables Are Used To Join To Other Tables
9. The Three Components of a Derived Table
10. Visualize This Derived Table
11. A Derived Table and CAST Statements
12. A Derived example Using The WITH Syntax
13. Clever Tricks on Aliasing Columns in a Derived Table
14. An example of Two Derived Tables in a Single Query
15. MULTIPLE Derived Tables using the WITH Command
16. Three Steps to Creating a Temporary Table
17. Three Versions of Creating a Temporary Table
18. ON COMMIT PRESERVE ROWS is the Greenplum Default
19. ON COMMIT DELETE ROWS
20. How to Use the ON COMMIT DELETE ROWS Option
21. ON COMMIT DROP
22. How to Use the ON COMMIT DROP Option
23. Create Table AS
24. Create Table LIKE
25. Creating a Clustered Index on a Temporary Table
18. Substrings and Positioning Functions
 1. The CHARACTERS Command Counts Characters
 2. The CHARACTERS Command and Char(20) Data
 3. CHARACTER_LENGTH and OCTET_LENGTH
 4. The TRIM Command trims both Leading and Trailing Spaces
 5. Trim Combined with the CHARACTERS Command
 6. How to TRIM only the Trailing Spaces
 7. A Visual of the TRIM Command Using Concatenation
 8. Trim and Trailing is Case Sensitive
 9. How to TRIM Trailing Letters
 10. The SUBSTRING Command
 11. SUBSTRING and SUBSTR are equal, but use different syntax
 12. How SUBSTRING Works with NO ENDING POSITION
 13. Using SUBSTRING to move Backwards
 14. How SUBSTRING Works with a Starting Position of -1
 15. How SUBSTRING Works with an Ending Position of 0
 16. An example using SUBSTRING, TRIM and CHAR Together
 17. The POSITION Command finds a Letters Position
 18. Concatenation
 19. Concatenation and SUBSTRING
 20. Four Concatenations Together
 21. Troubleshooting Concatenation
19. Interrogating the Data
 1. The NULLIF Command
 2. The COALESCE Command - Fill In the Answers
 3. The COALESCE Answer Set

4. COALESCE is Equivalent to This CASE Statement
5. The COALESCE Command
6. The COALESCE Answer Set
7. The Basics of CAST (Convert and Store)
8. Some Great CAST (Convert and Store) Examples
9. Some Great CAST (Convert and Store) Examples
10. A Rounding Example
11. Some Great CAST (Convert And STore) example
12. Using an ELSE in the Case Statement
13. Using an ELSE as a Safety Net
14. Rules For a Valued Case Statement
15. Rules for a Searched Case Statement
16. Valued Case Vs. A Searched Case
17. The CASE Challenge
18. The CASE Challenge Answer
19. Combining Searched Case and Valued Case
20. A Trick for getting a Horizontal Case
21. Nested Case
20. Set Operators Functions
 1. Rules of Set Operators
 2. Rules of Set Operators
 3. INTERSECT Explained Logically
 4. INTERSECT Explained Logically
 5. UNION Explained Logically
 6. UNION Explained Logically
 7. UNION ALL Explained Logically
 8. UNION ALL Explained Logically
 9. EXCEPT Explained Logically
 10. EXCEPT Explained Logically
 11. An Equal Amount of Columns in both SELECT List
 12. Columns in the SELECT list should be from the same Domain
 13. The Top Query handles all Aliases
 14. The Bottom Query does the ORDER BY (a Number)
 15. Great Trick: Place your Set Operator in a Derived Table
 16. UNION Vs UNION ALL
 17. Using UNION ALL and Literals
 18. A Great example of how EXCEPT works
 19. USING Multiple SET Operators in a Single Request
 20. Changing the Order of Precedence with Parentheses
 21. Using UNION ALL for speed in Merging Data Sets
21. View Functions
 1. The Fundamentals of Views
 2. Creating a Simple View to Restrict Sensitive Columns
 3. Creating a Simple View to Restrict Rows
 4. Basic Rules for Views
 5. Exception to the ORDER BY Rule inside a View
 6. Views sometimes CREATED for Formatting

7. Creating a View to Join Tables Together
8. Another Way to Alias Columns in a View CREATE
9. The Standard Way Most Aliasing is Done
10. What Happens When Both Aliasing Options Are Present
11. Resolving Aliasing Problems in a View CREATE
12. Answer to Resolving Aliasing Problems in a View CREATE
13. Aggregates on View Aggregates
14. Altering A Table
15. Altering A Table After a View has been Created
16. A View that Errors After An ALTER

22. Table Create and Data Types

1. Greenplum Has Only Two Distribution Policies
2. Creating a Table With A Single Column Distribution Key
3. The Default Table Storage is a Heap
4. Creating a Table With a Multi-Column Distribution Key
5. Creating a Table With Random Distribution
6. Creating a Table With No Distribution Key
7. Guidelines for Partitioning a Table
8. Creating a Partitioned Table Using a Range
9. A Visual of One Year of Data with Range Partitioning
10. Creating a Partitioned Table Using a Range Per Day
11. A Visual of One Year of Data with Range Per Day
12. Creating a Partitioned Table Using a List
13. Creating a Multi-Level Partitioned Table
14. Changing a Table to a Partitioned Table
15. Not Null Constraints
16. Unique Constraints
17. Primary Key Constraints
18. Check Constraints
19. Append Only Tables
20. Storage is Either Row, Column, or a Combination of Both
21. Column-Orientated Tables
22. CREATE INDEX Syntax
23. CREATE INDEX Syntax
24. Create Table LIKE
25. Greenplum Data Types

23. Data Manipulation Language (DML)

1. INSERT Syntax # 1
2. INSERT example with Syntax 1
3. INSERT Syntax # 2
4. INSERT example with Syntax 2
5. INSERT example with Syntax 3
6. INSERT/SELECT Command
7. INSERT/SELECT example using All Columns (*)
8. INSERT/SELECT example with Less Columns
9. Two UPDATE Examples
10. Subquery UPDATE Command Syntax example of Subquery UPDATE Command

11. Join UPDATE Command Syntax example of an UPDATE Join Command
12. Fast UPDATE
13. The DELETE Command Basic Syntax
14. DELETE and TRUNCATE Examples
15. To DELETE or to TRUNCATE
16. Subquery and Join DELETE Command Syntax
17. Example of Subquery DELETE Command
24. ANALYZE and VACUUM
 1. ANALYZE
 2. ANALYZE Options
 3. What Columns Should You Analyze?
 4. Why Analyze?
 5. VACUUM
 6. VACUUM Options
25. Greenplum Explain
 1. How to See an EXPLAIN Plan
 2. The Eight Rules to Reading an EXPLAIN Plan
 3. Interpreting Keywords in an EXPLAIN Plan
 4. Interpreting an EXPLAIN Plan
 5. A Single Segment Retrieve – The Fastest Query
 6. EXPLAIN With an ORDER BY Statement
 7. EXPLAIN ANALYZE
 8. EXPLAIN With a Range Query on a Table Partitioned By Day
 9. EXPLAIN That Uses a B-Tree Index Scan
 10. EXPLAIN That Uses a Bitmap Scan
 11. EXPLAIN With a Simple Subquery
 12. EXPLAIN With a Columnar Query
 13. EXPLAIN With a Clustered Index
 14. The Most Important Concept for Joins is the Distribution Key
 15. EXPLAIN With Join that has to Move Data
 16. EXPLAIN With Join that has to Move Data
 17. Changing the Join Query Changes the EXPLAIN Plan
 18. Analyzing the Tables Structures For a 3-Table Join
 19. An EXPLAIN For a 3-Table Join
 20. Explain of a Derived Table vs. a Correlated Subquery
 21. Explain of The Correlated Subquery
 22. Explain of The Derived Table
26. Statistical Aggregate Functions
 1. The Stats Table
 2. The STDDEV_POP Function
 3. A STDDEV_POP Example
 4. The STDDEV_SAMP Function
 5. A STDDEV_SAMP Example
 6. The VAR_POP Function
 7. A VAR_POP Example
 8. The VAR_SAMP Function
 9. A VAR_SAMP Example

- 10. The VARIANCE Function
- 11. A VARIANCE Example
- 12. The CORR Function
- 13. A CORR Example
- 14. A REGR_SYY Example
- 15. Using GROUP BY

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.