Greenplum Architecture

Introduction to the Greenplum Architecture

- The Basics of a Single Computer
- Data in Memory is Fast as Lightning
- Parallel Processing Of Data
- Symmetric Multi-Processing (SMP) Server
- Commodity Hardware Servers are Configured for Greenplum
- The Segment's Responsibilities The Host's Plan is Either All Segments or a Single Segment
- Greenplum has Linear Scalability
- The Architecture of A Greenplum Data Warehouse
- Nexus is Now Available For Greenplum

Greenplum Table Structures

- The Concepts of Greenplum Tables
- Tables are Either Distributed by Hash or Random
- Random Distribution Uses a Round Robin Technique
- Table are Either a Heap or Append-Only
- Tables are Stored in Either Row or Columnar Format
- Comparing Normal Table Vs. Columnar Tables
- Segments on Distributions are Aligned to Rebuild a Row
- Visualize the Data – Rows vs. Columns
- Table Rows are Either Sorted or Unsorted
- Creating a Clustered Index in Order to Physically Sort Rows
- Physically Ordered Tables Are Faster on Certain Queries
- Another Way to Create a Clustered Table
- Creating a B-Tree Index and then Running Analyze
- Creating a Bitmap Index
- Tables Can Be Partitioned
- Creating a Partitioned Table Using a List
- Creating a Multi-Level Partitioned Table
- Not Null and Unique Constraints

- When Do I Create a Non Clustered Index?
- B-Tree for Non Clustered Index on a Clustered Table
- Adding a Non Clustered Index To A
- B-Tree for Non Clustered Index on a Heap Table

Physical Database Design

- The Four Stages of Modeling for Greenplum - Check out #4
- The Logical Model
- First, Second and Third Normal Form
- The Employee_Table and Department_Table can be Joined
- The Employee_Table and Department_Table Join SQL
- The Extended Logical Model Template
- User Access is of Great Importance
- User Access in Layman's Terms
- User Access for Joins in Layman's Terms
- The Nexus Shows Users the Table's Distribution Key
- Data Demographics:
    - Distinct Rows
    - Distinct Rows Query
    - Max Rows Null
    - Max Rows Null Query
    - Max Rows Per Value
    - Max Rows Per Value
    - Typical Rows Per Value
    - Change Rating
- Typical Rows Per Value Query For Greenplum Systems
- SQL to Get the Average Rows Per Value for a Column (Mean)
- Factors When Choosing Greenplum Indexes
- Distribution Key Data Demographics Candidate Guidelines
- Distribution key Access Considerations
- Step 1 is to Pick All Potential Distribution Key Columns
- Step 2 is to Pick All Potential Secondary Indexes

- Answer to 2nd Step to Picking Potential Secondary Indexes
- Choose the Distribution Key and Secondary Indexes
- 3rd Step is to Picking your Indexes
- Our Index Picks

Denormalization

- Denormalization
- Derived Data
- Repeating Groups
- Pre-Joining Tables
- Storing Summary Data with a Trigger
- Summary Tables or Data Marts the Old Way
- Horizontal Partitioning the Old Way and the New Way
- Vertical Partitioning the Old Way
- Columnar Tables Are the New Vertical Partitioning

Nexus for Greenplum

- Nexus Queries Every Major System
- Nexus Data Visualization
- Nexus is Doing a Five-Table Join
- Nexus Generates the SQL Automatically
- Nexus Delivers the Report
- Cross-System Joins From Teradata, Oracle and SQL Server
- The Tabs of the Super Join Builder
- The 9 Tabs of the Super Join Builder
    - Objects Tab 1- Selecting Columns in the Objects Tab
    - Columns Tab 2- Removing Columns From the Report in the Columns Tab
    - Sorting Tab 3
    - Joins Tab 4
    - Where Tab 5- Using the WHERE Tab For Additional WHERE or AND
    - SQL Tab 6 – check paragraph below
    - Answer Set Tab 7
    - Analytics Tab 9

- Nexus Data Movement
- Moving a Single Table To a Different System
- The Single Table Data Movement Screen
- Moving an Entire Database To a Different System
- The Database Mover Screen and Options Tab
- Converting DDL Table Structures
- Compare and Synchronize
- Compare Two Different Databases From Different Systems
- Comparisons Down to the Column Level
- The Results Tab
- View Differences
- Synchronizing Differences In the Results Tab
- Hound Dog Compression

The Basics of SQL

- Introduction
- SELECT * (All Columns) in a Table
- Fully Qualifying a Database, Schema and Table
- SELECT Specific Columns in a Table
- Sort the Data with the ORDER BY Keyword
- ORDER BY Defaults to Ascending
- Use the Name or the Number in your ORDER BY Statement
- Two Examples of ORDER BY using Different Techniques
- Changing the ORDER BY to Descending Order
- NULL Values sort First in Ascending Mode (Default)
- NULL Values sort Last in Descending Mode (DESC)
- Major Sort vs. Minor Sorts
- Multiple Sort Keys using Names vs. Numbers
- Sorts are Alphabetical, NOT Logical
- Using A CASE Statement to Sort Logically
- How to ALIAS a Column Name
- A Missing Comma can by Mistake become an Alias

- Comments using Double Dashes are Single Line Comments
- Comments for Multi-Lines

The WHERE Clause

- The WHERE Clause limits Returning Rows
- Double Quoted Aliases are for Reserved Words and Spaces
- Character Data needs Single Quotes in the WHERE Clause
- Comparisons against a Null Value
- Use IS NULL or IS NOT NULL when dealing with NULLs
- Using Greater Than or Equal To (>=)
- AND in the WHERE Clause
- OR in the WHERE Clause
- Troubleshooting Character Data
- Using Different Columns in an AND Statement
- What is the Order of Precedence?
- Using Parentheses to change the Order of Precedence
- Using an IN List in place of OR
- IN List vs. OR brings the same Results
- Using a NOT IN List
- Null Values in a NOT IN List Bring Back No Rows
- A Technique for Handling Nulls with a NOT IN List
- BETWEEN is Inclusive
- NOT BETWEEN is Also Inclusive
- LIKE uses Wildcards Percent ‘%’ and Underscore ‘_’
- LIKE command Underscore is Wildcard for one Character
- ilike
- LIKE Command Works Differently on Char Vs Varchar
- Troubleshooting LIKE Command on Character Data
- Introducing the TRIM Command
- Introducing the RTRIM Command
- Numbers are Right Justified and Character Data is Left
- A Visual of CHARACTER Data vs. VARCHAR Data

- Use the TRIM command to remove spaces on CHAR Data
- Escape Character in the LIKE Command changes Wildcards
- Escape Characters Turn off Wildcards in the LIKE Command
- Introducing the RTRIM Command
- An example of Data with Left and Right Justification
- A Visual of CHARACTER Data vs. VARCHAR Data
- RTRIM command Removes Trailing spaces on CHAR Data
- Using Like with an AND/OR Clause to Find Letters

Distinct vs. Group By

- The Distinct Command
- Distinct vs. GROUP BY

Aggregation

- The 3 Rules of Aggregation
- There are Five Aggregates
- Troubleshooting Aggregates
- GROUP BY when Aggregates and Normal Columns Mix
- GROUP BY delivers one row per Group
- GROUP BY Dept_No or GROUP BY 1 the same thing
- Limiting Rows and Improving Performance with WHERE
- WHERE Clause in Aggregation limits unneeded Calculations
- Keyword HAVING tests Aggregates after they are Totaled
- Aggregates Return Null on Empty Tables
- Keyword HAVING is like an Extra WHERE Clause for Totals
- Keyword HAVING tests Aggregates after they are Totaled
- Getting the Average Values Per Column
- Average Values Per Column For all Columns in a Table
- Three types of Advanced Grouping
- Group By Grouping Sets/Rollup
- GROUP BY Cube

Join Functions

- Redistribution

- Big Table Small Table Join Strategy
- Duplication of the Smaller Table across All-Distributions
- If the Join Condition is the Distribution Key no Movement
- Matching Rows That Are On The Same Node Naturally
- Strategy 1 of 4 – The Merge Join
- Strategy 2 of 4 – The Hash Join
- Strategy 3 of 4 – The Nested Join
- Strategy 4 of 4 – The Product Join
- A Two-Table Join Using Traditional Syntax
- A two-table join using Non-ANSI Syntax with Table Alias
- You Can Fully Qualify All Columns
- A two-table join using ANSI Syntax
- Both Queries have the same Results and Performance
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- Which Tables are the Left and which Tables are Right?
- INNER JOIN with Additional AND Clause
- ANSI INNER JOIN with Additional AND and WHERE Clause
- OUTER JOIN with Additional WHERE and AND Clause
- Evaluation Order for Outer Queries
- The DREADED Product Join
- The Horrifying Cartesian Product Join
- The ANSI Cartesian Join will ERROR
- The CROSS JOIN
- The Self Join
- The Self Join with ANSI Syntax
- How would you Join these two tables?
- An Associative Table is a Bridge that Joins Two Tables
- The 5-Table Join – Logical Insurance Model
- The Nexus Query Chameleon Writes the SQL for Users.

Date Function

- Current_Date
- Current_Date, Current_Time, and Current_Timestamp
- Current_Time vs. LocalTime With Precision
- Local_Time and Local_Timestamp With Precision
- Now() and Timeofday() Functions
- Adding A Week to a Date
- Add or Subtract Days from a date
- Formatting Dates and Dollar Amounts
- The EXTRACT Command
- EXTRACT Command on the Century
- Date_part Command
- Date_Trunc Command With Time/Dates
- The AGE Command
- Epoch
- Using Intervals
- Interval Arithmetic Results
- A Complex Time Interval example using CAST
- The OVERLAPS Command
- Using Both CAST and CONVERT in Literal Values
- A Better Technique for YEAR, MONTH, and DAY Functions

Conversions and Formatting

- Postgres Conversion Functions
- To_Char command Examples
- Formatting A Date with To_Char
- To_Number
- To_Date
- To_Timestamp

Sub-query Functions

- An IN List is much like a Subquery
- The Subquery

- The Three Steps of How a Basic Subquery Works
- These are Equivalent Queries
- The Final Answer Set from the Subquery
- Should you use a Subquery of a Join?
- The Basics of a Correlated Subquery
- The Top Query always runs first in a Correlated Subquery
- Correlated Subquery Example vs. a Join with a Derived Table
- How to handle a NOT IN with Potential NULL Values
- IN is equivalent to =ANY
- Using a Correlated Exists
- How a Correlated Exists matches up
- The Correlated NOT Exists

OLAP Functions

- CSUM
- The ANSI CSUM
- Troubleshooting The ANSI OLAP on a GROUP BY
- Reset with a PARTITION BY Statement
- PARTITION BY only Resets a Single OLAP not ALL of them
- Moving SUM
- How ANSI Moving SUM Handles the Sort
- Moving SUM every 3-rows Vs a Continuous Average
- Partition By Resets an ANSI OLAP
- Both the Greenplum Moving Average and ANSI Version
- Moving Average
- The Moving Window is Current Row and Preceding
- How Moving Average Handles the Sort
- Moving Average every 3-rows Vs a Continuous Average
- Partition By Resets an ANSI OLAP
- Moving Difference using ANSI Syntax with Partition By
- RANK Defaults to Ascending Order
- Getting RANK to Sort in DESC Order

Temporary Tables

- Troubleshooting Concatenation

Interrogating the Data

- The NULLIF Command
- The COALESCE Command – Fill In the Answers
- COALESCE is Equivalent to This CASE Statement
- The COALESCE Command
- The Basics of CAST (Convert and Store)
- A Rounding Example
- Some Great CAST (Convert And STore) example
- Using an ELSE in the Case Statement
- Using an ELSE as a Safety Net
- Rules For a Valued Case Statement
- Rules for a Searched Case Statement
- Valued Case Vs. A Searched Case
- The CASE Challenge
- Combining Searched Case and Valued Case
- A Trick for getting a Horizontal Case
- Nested Case

Set Operators Functions

- Rules of Set Operators
- INTERSECT Explained Logically
- UNION Explained Logically
- UNION ALL Explained Logically
- EXCEPT Explained Logically
- An Equal Amount of Columns in both SELECT List
- Columns in the SELECT list should be from the same Domain
- The Top Query handles all Aliases
- The Bottom Query does the ORDER BY (a Number)
- Great Trick: Place your Set Operator in a Derived Table
- UNION Vs UNION ALL
- Using UNION ALL and Literals

- A Great example of how EXCEPT works
- USING Multiple SET Operators in a Single Request
- Changing the Order of Precedence with Parentheses
- Using UNION ALL for speed in Merging Data Sets

View Functions

- The Fundamentals of Views
- Creating a Simple View to Restrict Sensitive Columns/Rows
- Basic Rules for Views
- Exception to the ORDER BY Rule inside a View
- Views sometimes CREATED for Formatting
- Creating a View to Join Tables Together
- Another Way to Alias Columns in a View CREATE
- The Standard Way Most Aliasing is Done
- What Happens When Both Aliasing Options Are Present
- Resolving Aliasing Problems in a View CREATE
- Answer to Resolving Aliasing Problems in a View CREATE
- Aggregates on View Aggregates
- Altering A Table
- A View that Errors After An ALTER

Table Create and Data Types

- Greenplum Has Only Two Distribution Policies
- Creating a Table With A Single Column Distribution Key
- The Default Table Storage is a Heap
- Creating a Table With a Multi-Column Distribution Key
- Creating a Table With Random Distribution
- Creating a Table With No Distribution Key
- Guidelines for Partitioning a Table
- Creating a Partitioned Table Using a Range
- A Visual of One Year of Data with Range Partitioning
- Creating a Partitioned Table Using a Range Per Day
- Creating a Partitioned Table Using a List

- Creating a Multi-Level Partitioned Table
- Not Null Constraints
- Unique Constraints
- Primary Key Constraints
- Check Constraints
- Append Only Tables
- Column-Orientated Tables
- CREATE INDEX Syntax
- Create Table LIKE
- Greenplum Data Types

Data Manipulation Language (DML)

- INSERT Syntax # 1
- INSERT Syntax # 2
- INSERT example with Syntax 3
- INSERT/SELECT Command
- Two UPDATE Examples
- Subquery UPDATE Command Syntax
- Join UPDATE Command Syntax
- Fast UPDATE
- The DELETE Command Basic Syntax
- To DELETE or to TRUNCATE
- Subquery and Join DELETE Command Syntax

ANALYZE and VACUUM

- ANALYZE
- What Columns Should You Analyze?
- Why Analyze?
- VACUUM

Greenplum Explain

- How to See an EXPLAIN Plan
- The Eight Rules to Reading an EXPLAIN Plan
- Interpreting Keywords in an EXPLAIN Plan

- Interpreting an EXPLAIN Plan
- A Single Segment Retrieve – The Fastest Query
- EXPLAIN With an ORDER BY Statement
- EXPLAIN ANALYZE
- EXPLAIN With a Range Query on a Table Partitioned By Day
- EXPLAIN That Uses a B-Tree Index Scan
- EXPLAIN That Uses a Bitmap Scan
- EXPLAIN With a Simple Subquery
- EXPLAIN With a Columnar Query
- EXPLAIN With a Clustered Index
- EXPLAIN With Join that has to Move Data
- EXPLAIN With Join that has to Move Data
- Changing the Join Query Changes the EXPLAIN Plan
- Analyzing the Tables Structures For a 3-Table Join
- An EXPLAIN For a 3-Table Join
- Explain of a Derived Table vs. a Correlated Subquery

Statistical Aggregate Functions

- The Stats Table
- Above, is the Stats_Table data in which we will use in our statistical examples
- The STDDEV_POP Function
- The STDDEV_SAMP Function
- The VAR_POP Function
- The VAR_SAMP Function
- The VARIANCE Function
- The CORR Function
- The COVAR_POP Function
- The COVAR_SAMP Function
- The REGR_INTERCEPT Function
- The REGR_SLOPE Function
- The REGR_AVGX Function
- The REGR_AVGY Function

- The REGR_COUNT Function
- The REGR_R2 Function
- The REGR_SXX Function
- The REGR_SXY Function
- The REGR_SYY Function
- Using GROUP BY