



# Greenplum Architecture Class Outline

## Introduction to the Greenplum Architecture

- What is Parallel Processing?

- 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

- Commodity Hardware Allows For One Segment Per CPU

- The Master Host

- The Segment's Responsibilities

- The Host's Plan is Either All Segments or a Single Segment

- A Table has Columns and Rows

- 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  
A Hash Distributed Table has A Distribution Key  
Picking A Distribution Key That Is Not Very Unique  
Random Distribution Uses a Round Robin Technique  
Tables Will Be Distributed Among All Segments  
The Default For Distribution Chooses the First Column  
Table are Either a Heap or Append-Only  
Tables are Stored in Either Row or Columnar Format  
Creating a Column Oriented Table  
Comparing Normal Table Vs. Columnar Tables  
Columnar can move just One Column Block Into Memory  
Segments on Distributions are Aligned to Rebuild a Row  
Columnar Tables Store Each Column in Separate Blocks  
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  
Why Create a Bitmap Index?  
Tables Can Be Partitioned  
A Table Partitioned By Range (Per Month)  
A Visual of a Partitioned Table by Range (Month)  
Tables Can Be Partitioned by Day  
Visualize a Partitioned Table by Day  
Creating a Partitioned Table Using a List  
Creating a Multi-Level Partitioned Table  
Changing a Table to a Partitioned Table

Not Null Constraints

Unique Constraints

Unique Constraints That Fail

Primary Key Constraints

A Primary Key Automatically Creates a Unique Index

Check Constraints

Creating an Automatic Number Called a Sequence

Multiple INSERT example Using a Sequence

## Hashing and Data Distribution

Distribution Keys Hashed on Unique Values Spread Evenly

Distribution Keys With Non-Unique Values Spread Unevenly

Best Practices for Choosing a Distribution Key

The Hash Map Determines which Segment owns the Row

The Hash Map Determines which Node will Own the Row

The Hash Map Determines which Node will Own the Row

The Hash Map Determines which Node will Own the Row

Hash Map Determines which Node will Own the Row

A Review of the Hashing Process

Non-Unique Distribution Keys have Skewed Data

Non-Unique Distribution Keys have Skewed Data

## The Technical Details

Greenplum Limitations

Every Segment has the Exact Same Tables

Tables are Distributed Across All Segments

The Table Header and the Data Rows are Stored Separately

Segments Store Rows inside a Data Block Called a Page

To Read a Data Block a Node Moves the Block into Memory

A Full Table Scan Means All Nodes Must Read All Rows

Rows are Organized inside a Page

Moving Data Blocks is Like Checking In Luggage

As Row-Based Tables Get Bigger, the Page Splits

Data Pages are Processed One at a Time Per Unit

Creating a Table that is a Heap

Heap Page

Creating a Table that has a Clustered Index

Clustered Index Page

The Row Offset Array is the Guidance System for Every Row

The Row Offset Array Provides Two Search Options (1 of 2)

The Row Offset Array Provides Two Search Options (2 of 2)

The Row Offset Array Helps With Inserts

B-Trees

The Building of a B-Tree for a Clustered Index (1 of 3)

The Building of a B-Tree for a Clustered Index (2 of 3)

The Building of a B-Tree for a Clustered Index (3 of 3)

When Do I Create a Clustered Index?

When Do I Create a Non Clustered Index?

B-Tree for Non Clustered Index on a Clustered Table (1 of 2)

B-Tree for Non Clustered Index on a Clustered Table (2 of 2)

Adding a Non Clustered Index To A

B-Tree for Non Clustered Index on a Heap Table (1 of 2)

B-Tree for Non Clustered Index on a Heap Table (2 of 2)

## Physical Database Design

The Four Stages of Modeling for Greenplum - Check out #4

The Logical Model

The Logical Model can be loaded inside Nexus

First, Second and Third Normal Form

Quiz – Choose that Normalization Technique

Answer to Quiz – Choose that Normalization Technique

Quiz – What Normalization is it Now?

Answer to Quiz – What Normalization is it Now?

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 Tell Us if the Column is Worthy

Data Demographics – Distinct Rows

Data Demographics – Distinct Rows Query

Data Demographics – Max Rows Null

Data Demographics – Max Rows Null Query

Data Demographics – Max Rows Per Value

Data Demographics – Max Rows Per Value

Data Demographics – Typical Rows Per Value

Typical Rows Per Value Query For Greenplum Systems

SQL to Get the Average Rows Per Value for a Column (Mean)

Data Demographics – Change Rating

Factors When Choosing Greenplum Indexes

Distribution Key Data Demographics Candidate Guidelines

Distribution key Access Considerations

Answer -Three Important distribution key Considerations

Step 1 is to Pick All Potential Distribution Key Columns

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

3<sup>rd</sup> 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

Horizontal Partitioning the New Way

Vertical Partitioning the Old Way

Columnar Tables Are the New Vertical Partitioning

## Nexus for Greenplum

Nexus is Available on the Cloud

Nexus Queries Every Major System

Setup of Nexus is as Easy as Pie

Setup of Nexus is as Easy as 1, 2, 3

Nexus Data Visualization

Nexus Data Visualization

Nexus Data Visualization Shows What Tables Can Be Joined

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

The 9 Tabs of the Super Join Builder – Columns Tab 2

Removing Columns From the Report in the Columns Tab

The 9 Tabs of the Super Join Builder – Sorting Tab 3

The 9 Tabs of the Super Join Builder – Joins Tab 4

The 9 Tabs of the Super Join Builder – Where Tab 5

Using the WHERE Tab For Additional WHERE or AND

The 9 Tabs of the Super Join Builder – SQL Tab 6 – check paragraph below

The 9 Tabs of the Super Join Builder – Answer Set Tab 7

The 9 Tabs of the Super Join Builder – Analytics Tab 9

Analytics Tab

Analytics Tab – OLAP Example

Analytics Tab – OLAP Example of SQL Generated

Analytics Tab – Grouping Sets Example

Analytics Tab – Grouping Sets Answer Set

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

The Database Mover Options Tab

Converting DDL Table Structures

Converting DDL Table Structures

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

Synchronizing Differences In the Results Tab

Hound Dog Compression

Hound Dog Compression On Greenplum



