Amazon Redshift Architecture Class Outline

What is Columnar?

- What is Parallel Processing?
- The Basics of a Single Computer
- Data in Memory is Fast as Lightning
- Parallel Processing of Data
- A Table has Columns and Rows
- Each Parallel Process Organizes the Rows inside a Data Block
- Moving Data Blocks is Like Checking in Luggage
- Facts That Are Disturbing
- Why Columnar?
- Row Based Blocks vs. Columnar Based Blocks
- As Row-Based Tables Get Bigger, the Blocks Split
- Data Blocks Are Processed One at a Time Per Unit
- Columnar Tables Store Each Column in Separate Blocks
- Visualize the Data – Rows vs. Columns
- Row Based Blocks Can Waste Memory Space and Resources

The Architecture of Redshift
Redshift has Linear Scalability
Distribution Styles
Distribution Key Where the Data is Unique
Another Way to Create A Table
Distribution Key Where the Data is Non-Unique
Distribution Key is ALL
Even Distribution Key
Matching Distribution Keys for Co-Location of Joins
Big Table / Small Table Joins
Fact and Dimension Table Distribution Key Designs
Improving Performance By Defining a Sort Key
Sort Keys Help Group By, Order By and Window Functions
Each Block Comes With Metadata
How Data Might Look On A Slice
Question – How Many Blocks Move Into Memory?
Answer – How Many Blocks Move Into Memory?
Quiz – Master that Query With the Metadata
Answer to Quiz – Master that Query With the Metadata
Creating Three Tables with Different Sort Key Strategies
A Table with a Distribution Key and a Single-Sortkey
A Normal Sort Key Example
Creating a Table with an Interleaved Sort Key
Interleaved Vs. a Normal Sort Key
The ANALYZE Command Collects Statistics
Redshift Automatically ANALYZES Some Create Statements
What is a Vacuum?
When is a Good Time to Vacuum?
The VACUUM Command Grooms a Table

Database Limits
Creating a Database
Creating a User
Dropping a User
Inserting into a Table
Renaming a Table or a Column
Adding and Dropping a Column to a Table

Best Practices for Table Design
Converting Table Structures to Redshift
Converting Table Structures to Redshift Finale
Best Practices for Designing Tables
Choose the Best Sort Key
Each Block Comes with Metadata
Creating a Sort Key
Sort Keys Help Group By, Order By and Window Functions
Choose a Great Distribution Key
Distribution Key Where the Data is Unique
Matching Distribution Keys for Co-Location of Joins
Big Table / Small Table Joins
Define Primary Key and Foreign Key Constraints
Primary Key and Foreign Key Examples
Use the Smallest Column Size When Creating Tables
Use Date/Time Data Types for Date Columns
Specify Redundant Predicates on the Sort Column
Setting the Statement_Timeout to Abort Long Queries

System Tables
Redshift Has System Tables that Log to Disk (Prefix STL)

Redshift Has System Tables that are Virtual (STV Prefix)
Redshift Has System Catalog Tables Visible to Users
Amazon Redshift System Tables
Trouble Shooting Catalog Table pg_table_def
Seeing the System Tables in your Nexus Tree
Catalog Table pg_table_def
Checking Tables for Skew (Poor Distribution)
Checking All Statements That Used the Analyze Command
Checking Tables for Skew (Poor Distribution)
Checking for Details About the Last Copy Operation
Checking When a Table Has Last Been Analyzed
Checking for Column Information on a Table
System tables for troubleshooting data loads
Determining Whether a Query is Writing to Disk
Showing Alert events
Showing the Last Queries Run on the System
Showing Queries that Last More than One Second
Listing Queries From Longest to Shortest for a Particular Day
Reporting Queries with High CPU Time
Reporting Queries of Nested Loops Returning Many Rows
Finding Queries Aborted Because of a Monitoring Rule
The Number of MB blocks used by each column in a Table
Checking if a Table is Distributed Over All Slices
List Schemas and Tables in a Database from the PG Catalog
A View to See the State of the system Queues for Workloads
SELECT From the WLM_QUEUE_STATE_VW View
WLM_QUEUE_STATE_VW View Definitions
A View Showing the State of Current Queries and Queues
WLM_QUERY_STATE_VW View Definitions

Compression

- Compression Types
- Byte Dictionary Compression
- Delta Encoding
- LZO Encoding
- Mostly Encoding
- Runlength encoding
- Text255 and Text 32k Encodings
- ANALYZE COMPRESSION
- Copy

Temporary Tables

- Create Table Syntax
- Basic Temporary Table Examples
- More Advanced Temporary Table Examples
- Advanced Temporary Table Examples
- Table Limits and CTAS
- Performing a Deep Copy
- Deep Copy Using the Original DDL
- Deep Copy Using a CTAS
- Deep Copy Using a Create Table LIKE
- Deep Copy by Creating a Temp Table and Truncating Original
- CREATING A Derived Table
- The Three Components of a Derived Table
- Naming the Derived Table
- Aliasing the Column Names in The Derived Table
- Visualize This Derived Table
- Most Derived Tables are Used To Join To Other Tables

Multiple Ways to Alias the Columns in a Derived Table
Our Join Example with a Different Column Aliasing Style
Column Aliasing Can Default For Normal Columns
CREATING A Derived Table using the WITH Command
Our Join Example With The WITH Syntax
WITH Statement That Uses a SELECT *
A WITH Clause That Produces Two Tables
The Same Derived Query shown Three Different Ways
Quiz - Answer the Questions
Answer to Quiz - Answer the Questions
Clever Tricks on Aliasing Columns in a Derived Table
A Derived Table lives only for the lifetime of a single query
An Example of Two Derived Tables in a Single Query
Connecting To Redshift Via Nexus
Connecting To Redshift Via Nexus
Connecting To Redshift Via Nexus
Connecting To Redshift Via Nexus

Explain

Three Ways to Run an EXPLAIN
EXPLAIN – Steps, Segments and Streams
EXPLAIN Terms For Scans and Joins
EXPLAIN Terms For Aggregation and Sorts
EXPLAIN Terms For Set Operators and Miscellaneous Terms
EXPLAIN Terms For Set Operators and Miscellaneous Terms
EXPLAIN Example and the Cost
EXPLAIN Example and the Rows
EXPLAIN Example and the Width
Simple EXPLAIN Example and the Costs

Look for These Keywords to Track Data Movement
EXPLAIN Join Example Using DS_BCAST_INNER
EXPLAIN Join Example Using DS_DIST_NONE
EXPLAIN Showing DS_DIST_NONE Visually
EXPLAIN With a Warning
EXPLAIN For Ordered Analytics Such as CSUM
EXPLAIN For Scalar Aggregate Functions
EXPLAIN For Hash Aggregate Functions
EXPLAIN Using Limit, Merge and Sort
EXPLAIN Using a WHERE Clause Filter
EXPLAIN Using the Keyword Distinct
EXPLAIN for Subqueries

User Defined Functions
Creating a User Defined Scalar Function
Function Syntax
Creating a Simple Function
Creating a Function That Shows the Sunday Date of the Week
Create a Flight_Table that Holds Longitude and Latitude
A Function Example for Measuring Distance in Miles
A Function Example for Measuring Distance
Create a Flight_Table that Holds Longitude and Latitude
A Function Example for Comparing Two Numbers
A Function Example Using Multiple Tables
SQL that Utilizes Two User Defined Functions (UDFs)
Function Volatility
Amazon Redshift Vs. Python Data Types
Privileges

Workload Management

- Create the WLM_QUEUE_STATE_VW View
- SELECT From the WLM_QUEUE_STATE_VW View
- WLM_QUEUE_STATE_VW View Definitions
- Create the WLM_QUERY_STATE_VW View
- WLM_QUERY_STATE_VW View Definitions
- Open Up Two Sessions in your Nexus
- SELECT From our WLM_QUERY_STATE_VW View
- Run a Long-Running Query in Tab
- In Tab Run These Two Queries
- After Setup of Four Queues
- How to use the SET command to Place a Query in a Queue
- Checking which Queue the Query is Executing In?
- How to Reset the Query Group
- Creating and Altering a Group
- Admin User Can Still SET to a Different Queue if they Want
- Overriding the Concurrency Level