

Snowflake Architecture and SQL

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: SQL-701

Duration: 5 days

Overview

Course Description

In this private, online or onsite Snowflake Architecture and SQL training course, attendees learn SQL starting at the most basic level and going to the most advanced level, including working with date functions, non-structured data (JSON), advanced analytics, User Defined Functions (UDF), and Stored Procedures. Students learn how to create Snowflake tables and views, performance tune Snowflake, and more. This class also prepares the learners for [Snowflake Certification](#).

Skills Gained

- Understand data ingestion
- Implement stored procedures and UDFs
- Master advanced SQL skills
- Share data
- Perform advanced analytics

Prerequisites

This class presumes that attendees have prior SQL SELECT experience and that at least the first two chapters will be review.

Training Materials

All Snowflake training students receive comprehensive courseware.

Software Requirements

All attendees will need access to a Snowflake SQL environment with Ascendient Learning's sample data loaded.

Audience

Course Details

Outline

- Getting Started with a Snowflake Trial Account
 - How to Create a Starter Account on Snowflake
 - Click on the START FOR FREE Menu Item
 - Start Your 30-Day Free Trial
 - Choose Your Edition of Snowflake and Cloud Provider
 - You're Now Signed Up – An Email Will Arrive Soon
 - Take the Snowflake Email and CLICK TO ACTIVATE
 - Click the Snowflake Link to Enter a Username and Password
 - Welcome to Snowflake Message Comes Up
 - Snowflake Browser and Query Tool
 - Snowflake Worksheet to Create and Run Queries
 - Script to Create the Database, Schema, Tables, and Views
 - Choose the COURSE OUTLINE Button for Snowflake
 - Press on Download Snowflake Class Database Script Button
 - Copy All of the Text From Your SnowflakeClass.Txt Script
 - Run Your Script on the Snowflake Website – First Way
 - Copy Your Script to the Snowflake Website – Alternative Way
 - Run Your Script on the Snowflake Website – Alternative Way
 - Your Systems Tree will Contain the Nexus Database
 - The SQL_CLASS Schema Contains Your Tables
 - Set Your Database to NEXUS and Schema to SQL_CLASS

- Download Nexus at CoffingDW.com
- Nexus Server Migrates All Data Warehouses to Snowflake
- Nexus Joins Data Across Platforms with the Super Join Builder
- What is Snowflake?
 - Snowflake IPO Largest in History
 - Advertising in the Computer Industry Pays off Each Year
 - Scaling Compute and Storage on Public Clouds
 - Customer Retention is Gold in the Computer Industry
 - High-powered Investors are Critical
 - Who are the Biggest Losers in the IPO?
 - Sharing Data Between Customers and Providers is Amazing
 - Standard SQL Makes Application Migration Easy
 - Hiring a Proven CEO can make the Difference
 - Snowflake Architecture
 - Snowflake Tables are Immutable
 - Elasticity
 - Worker Nodes and Worker Processes
 - Min-Max Based Pruning
 - Snowflake Execution Engine
 - Fault Tolerance
 - Semi-Structured Data
 - Snowflake Editions
 - Snowflake Pricing
 - Snowflake Virtual Warehouse Sizes
 - Snowflake Storage Pricing
- Configuring and Managing the Snowflake Warehouse
 - Five System Defined Snowflake Roles
 - Create a Snowflake Virtual Data Warehouse
 - Create a Role
 - Create a User and Grant a Role
 - Scaling Up or Down the Virtual Data Warehouse
 - Scaling Up the Virtual Warehouse
 - Scaling Out the Virtual Data Warehouse

- Configuring a Warehouse
- The Life of a Query
- Resource Monitor
- Create Resource Monitor Screen
- Creating a Resource Monitor with SQL
- Brilliant Features of Snowflake
 - Snowflake Caching
 - The Life of a Query with Caching
 - Time Travel Using
 - Restoring Data Using Time Travel Feature (Best Option)
 - Restoring Data Using Time Travel Feature (Bad Option)
 - CREATE TABLE with Time Travel Days
 - Account Usage
 - Drop and Undrop
 - Rename a Table
 - SHOW TABLES With LIKE
 - Table Types
 - TABLE STORAGE METRICS QUERY
 - Creating a Transient Schema
 - Zero-Copy Cloning
 - Cloning a Temporary Table
 - Creating Temporary and Transient Tables
 - Fail-Safe
- Performance Tuning with Cluster Keys
 - What is Parallel Processing?
 - The Basics of a Single Computer
 - Data in Memory is Fast as Lightning
 - Parallel Processing Of Data
 - Snowflake has Linear Scalability
 - Snowflake Clustering For Performance Tuning
 - Creating a Table with a Cluster Key
 - Altering a Table to Create or Drop a Cluster Key
 - Joining Tables Can Have the Same Cluster Keys for Speed

- The Emp_Tbl CREATE Statement with Eight Rows
- Snowflake Quiz –Put the Rows on the Proper Micro-Partition
- The First Row is Now on the Proper Micro-Partition
- Place the Remaining Rows on the Proper Micro-Partition
- The Rows Are Now on the Proper Micro-Partition
- Snowflake Slices will use Columnar Storage
- Snowflake Micro-Partitions Create Metadata for each Block
- Emp_Tbl CREATE Statement with a Cluster Key of DeptNo
- Snowflake Quiz –Place the Rows on the Micro-Partition
- The First Row is on the Proper Micro-Partition
- Place the Remaining Rows on the Proper Micro-Partitions
- The Rows are on the Proper Micro-Partitions
- Snowflake will use Columnar Storage
- Each Block Comes with Metadata
- Snowflake Architecture: Data Storage
- Min-Max Based Pruning
- Data Sharing
 - Data Sharing
 - Implementing Data Sharing
 - Investigating Data Sharing – SHOW SHARES
 - SEE SHARE DETAILS with the DESC Command
 - CREATE a Database for the SHARE
 - Sharing All Tables in a Database and Schema
 - Investigating Data Sharing – SHOW SHARES
 - SEE SHARE DETAILS with the DESC Command
 - Data Sharing for Non-Snowflake Account Users
 - Steps We Need to Take to Share with Non-Snowflake Users
 - Creating a Reader Account
 - Viewing Managed Accounts
 - Sharing Data with the Managed Account
 - Troubleshooting Sharing Data with the Managed Account
 - Gathering the URL for Logging Into a Reader Account
 - Using the URL to Login to your Snowflake Reader Account

- SHOW SHARES for Reader Account
- SEE SHARE DETAILS with the DESC Command
- CREATE a Database and Warehouse for the SHARE
- Create a User and Grant a Role for Reader Account Users
- Snowflake Tasks
 - Creating a Task
 - Creating a Task Using CRON
 - Creating a Task Using CRON Using Minute and Hour
 - Tasks Using CRON Using Minute, Hour, and Day of Week
 - Creating a Tree of Tasks
- Information Schema System Catalog
 - APPLICABLE_ROLES
 - COLUMNS
 - COLUMNS EXAMPLE
 - DATABASES
 - ENABLED_ROLES
 - EXTERNAL_TABLES
 - FILE_FORMATS
 - FUNCTIONS
 - INFORMATION_SCHEMA_CATALOG_NAME
 - LOAD_HISTORY
 - OBJECT_PRIVILEGES
 - PACKAGES
 - PIPES
 - PROCEDURES
 - REFERENTIAL_CONSTRAINTS
 - REPLICATION_DATABASES
 - SCHEMATA
 - SEQUENCES
 - STAGES
 - TABLE_CONSTRAINTS
 - TABLE_PRIVILEGES
 - TABLE_STORAGE_METRICS

- TABLES
- USAGE_PRIVILEGES
- VIEWS
- Systems Functions
 - ALL_USER_NAMES and CURRENT_IP_ADDRESS
 - CURRENT_ACCOUNT, CURRENT_AVAILABLE_ROLES
 - CURRENT_CLIENT and CURRENT_DATE
 - CURRENT_DATABASE and CURRENT_SCHEMA
 - CURRENT_SESSION and CURRENT_STATEMENT
 - CURRENT_TIME and CURRENT_TIMESTAMP
 - CURRENT_VERSION and CURRENT_WAREHOUSE
 - CURRENT_TRANSACTION and CURRENT_USER
 - CURRENT_REGION and CURRENT_ROLE
- Data Ingestion
 - Loading Data
 - What is a Staging Area?
 - Snowflake has Four Types of Stages
 - Snowflake Internal Stages
 - Data File Details
 - User Stage
 - Loading User Stage Data with the Snowflake Web Interface
 - Go to the Databases Tab to See Your Databases
 - Click on the Table You Want to Load
 - The Table Columns Appear – Choose Load Table
 - Load Data Screen Appears
 - Choose Your Source Files
 - Choose Your File Format
 - Create the File Format
 - Create the File Format – Press Load
 - Load Results
 - Table Stage
 - Download a Flat File From Our Website for Load Exercise
 - Choose the COURSE OUTLINE Button for Snowflake

- Press on Download Snowflake Class Database Script Button
- Copying Data Into a Table Using the Table Stage
- The PUT Command will not work on Snowflake Browser
- Query a Table Residing in the Table Stage
- Using the Copy Command with File Format Options
- Named Stages
- Create Three Internal Named Stages Using SQL
- Creating Three Internal Named Stages with Nexus
- Using an Internal Named Stage to Load Data with PUT
- Create Named Stage Using the Snowflake Browser Tool
- Importing Data From External Stages
- Example of Copy Command from an External Stage
- Example of Copy Command Using Pattern
- Pattern Results from Loading Multiple Files
- Copying Only Some Columns Into a Table
- Functions for Transforming Data During a Load
- Transformation Example Using CASE and Implicit CAST
- Transformation Example To Populate Only Two Columns
- Using the ON_ERROR Options
- Creating and Describing a File Format
- Altering or Changing Attributes in a File Format
- DESCRIBING our File Format to Confirm Attribute Changes
- Important Copy Options – Validation Mode
- Validation Mode RETURN_ERRORS
- Saving Load Error Rows in a Table
- Another Technique to Save Load Error Rows in a Table
- SPLIT_PART Function for Easy Reading of Errors
- Important Copy Options – Size Limit
- Example of Using the Size_Limit Option
- Important Copy Options – RETURNED_FAILED_ONLY
- RETURNED_FAILED_ONLY Example
- Important Copy Options – TRUNCATECOLUMNS
- TRUNCATECOLUMNS Example

- Important Copy Options – FORCE
- FORCE Example
- An Example of a JSON File
- Creating a JSON Stage and File Format
- Querying the JSON Table
- Tricks and Tips to Query the JSON Table
- Creating a Table with Flattened Data
- LOAD_HISTORY
- STAGES View in Information_Schema
- Nexus Server Migrates All Data Warehouses to Snowflake
- Introduction to SQL
 - Introduction
 - SELECTING Current Information
 - Setting Your Default DATABASE and SCHEMA
 - SELECT * (All Columns) in a Table
 - SELECT Specific Columns in a Table
 - Commas in the Front or Back?
 - Place your Commas in front for better Debugging Capabilities
 - Sort the Data with the ORDER BY Keyword
 - Use a Column Name or Number in an ORDER BY Statement
 - Two Examples of ORDER BY using Different Techniques
 - Changing the ORDER BY to Descending Order
 - NULL Values Sort Last in Ascending Mode (Default)
 - Using the Nulls First Command
 - NULL Values Sort First in Descending Mode (DESC)
 - Using the Nulls Last Command
 - Major Sort vs. Minor Sort
 - Multiple Sort Keys using Names vs. Numbers
 - An Order By That Uses an Expression
 - Sorts are Alphabetical, NOT Logical
 - Using A Valued CASE Statement to Sort Logically
 - Using A Searched CASE Statement to Sort Logically
 - Order By Decode

- How to ALIAS a Column Name
- Using an Alias in the WHERE and ORDER BY Clause
- Using an Alias in the ORDER BY Clause with Decode
- A Missing Comma Can Become an Alias by Mistake
- Comments using Double Dashes are Single Line Comments
- Comments for Multi-Lines
- Comments are a Great Technique for Finding SQL Errors
- The WHERE Clause
 - The WHERE Clause limits Returning Rows
 - Numbers Don't Need Single Quotes
 - Not Equal
 - Searching for NULL Values Using Equality Returns Nothing
 - Use IS NULL or IS NOT NULL when dealing with NULLs
 - Using Greater Than Or Equal To (>=)
 - AND in the WHERE Clause
 - Troubleshooting AND
 - OR in the WHERE Clause
 - Troubleshooting OR
 - WHY OR must utilize the Column Name Each Time
 - 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
 - The IN List is an Excellent Technique
 - IN List vs. OR Brings the Same Results
 - The IN List Can Use Character Data
 - Using a NOT IN List
 - Null Values in a NOT IN List Return No Rows
 - A Technique for Handling Nulls with a NOT IN List
 - The BETWEEN Statement is Inclusive
 - The NOT BETWEEN Statement is also Inclusive
 - The BETWEEN Statement Works for Character Data

- The Like Command Wildcards are Percent and Underscore
- LIKE command Underscore is Wildcard for one Character
- CASE Matters with the LIKE Command
- Using LIKE for all Cases with Lower and Upper
- Another Example of UPPER and LOWER
- LIKE Command to Find Multiple Characters
- LIKE Command to Find Either Character
- Using ILIKE Handle Case Issues
- Finding Anyone Whose Name Ends in 'Y'
- Escape Character
- Time Travel
- Distinct, Group By, Top, and Pivot
 - The Distinct Command
 - DISTINCT vs. GROUP BY
 - Top Command
 - Top Command and Order By
 - Top Command and Order By Plus Nulls Last
 - The FETCH Clause
 - Sample and Tablesample
 - TOP vs. SAMPLE
 - Two Forms of Data Sampling
 - The Pivot Command
- Aggregation
 - Casting a Data Type
 - Troubleshooting Aggregates
 - GROUP BY Delivers One Row Per Group
 - GROUP BY DEPT_NO or GROUP BY 1 are Equivalent
 - Limiting Rows and Improving Performance with WHERE
 - WHERE Clause in Aggregation limits unneeded Calculations
 - Keyword HAVING tests Aggregates after they are Totaled
 - Keyword HAVING is like an Extra WHERE Clause for Totals
 - ANY_VALUE
 - COUNT_IF

- GROUP BY GROUPING SETS
- GROUP BY ROLLUP
- GROUP BY CUBE
- Joining Tables
 - A Two-Table Join Using Traditional Syntax
 - A Join using Traditional Syntax with Table Alias
 - You Can Fully Qualify All Columns
 - A 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 are Right?
 - Answer - Which Tables are the Left and Which are Right?
 - INNER JOIN
 - ANSI INNER JOIN
 - OUTER JOIN
 - The DREADED Product Join
 - Cartesian Product Join
 - The CROSS JOIN
 - The SELF JOIN
 - An Associative Table is a Bridge that Joins Two Tables
 - The Five-Table Join – Logical Insurance Model
- Date Functions
 - CURRENT_DATE
 - CURRENT_DATE and CURRENT_TIMESTAMP
 - Current_Timestamp and Local_Timestamp With Precision
 - CURRENT_TIME vs. LOCALTIME With Precision
 - Add or Subtract Days from a Date
 - The ADD_MONTHS Command
 - ADD_MONTHS to Add a Year to a Date
 - ADD_MONTHS to Add Five Years to a Date
 - Incrementing Date Values Using the Dateadd Function

- Incrementing Time Values Using the Dateadd Function
- Dateadd Function And Add_Months Function are Different
- Formatting a Date
- The TO_CHAR Command to Format Dollar Signs
- The TO_CHAR Command for Formatting Numbers
- The EXTRACT Command
- MONTHNAME
- EXTRACT from DATES and TIME
- EXTRACT from DATES and TIME Optional Syntax
- Another Option for Extracting Portions of Dates and Times
- The DATE_PART Function
- Using DATE_PART to Extract
- Implied Extract of Day, Month, and Year using TO_CHAR
- The DATE_PART Function Using Day of Week (DOW)
- Day of Week and a CASE Statement
- Day of Week and DECODE
- Great Date Functions to Know
- Week of Year and Year of Week
- First Day and Last Day Functions
- DATEDIFF
- Using CASE and Extract to Reformat Dates
- Using CAST and SUBSTRING to Reformat Dates
- The Date_Trunc Function
- DATE_TRUNC Command With Time
- DATE_TRUNC Command With Dates
- LAST_DAY
- LAST_DAY
- NEXT_DAY
- PREVIOUS_DAY
- MONTHS_BETWEEN
- TIME_SLICE
- TO_TIMESTAMP
- Using Day, Month, and Year intervals

- The Basics of a Simple Interval
- Temporary Tables
 - Derived Query Examples with Three Different Techniques
 - Most Derived Tables Are Used To Join To Other Tables
 - The Three Components of a Derived Table
 - Visualize This Derived Table
 - An Example of Two Derived Tables Using WITH
 - WITH RECURSIVE
 - Creating a Temporary Table
 - CREATE a Temporary TABLE using LIKE
 - Creating a Temporary Table using a CTAS
 - CREATE Temporary Table AS (CTAS) with Specific Columns
 - CREATE a Temporary Table AS (CTAS) Using a Join
 - Cloning a Temporary Table
 - Dropping a Temporary Table
- Sub-query Functions
 - An IN List is much like a Subquery
 - An IN List Never has Duplicates – Just like a Subquery
 - An IN List Ignores Duplicates
 - The Subquery
 - The Three Steps of How a Basic Subquery Works
 - These are Equivalent Queries
 - The Answer Set from the Subquery
 - Answer to Quiz- Answer the Difficult Question
 - Should you use a Subquery or 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
 - NOT IN Subquery Returns Nothing when NULLs are Present
 - Fixing a NOT IN Subquery with Null Values
 - How the Double Parameter Subquery Works
 - More on how the Double Parameter Subquery Works
 - IN is equivalent to =ANY

- Using a Correlated Exists
- How a Correlated Exists Matches Up
- The Correlated NOT Exists
- Analytic and Window Functions
 - ROW_NUMBER
 - Using a Derived Table
 - RANK
 - Dense_Rank
 - RANK vs. DENSE_RANK
 - Getting RANK to Sort in DESC Order
 - RANK() OVER, PARTITION BY, and QUALIFY
 - Using a Derived Table
 - DENSE_RANK() OVER and PARTITION BY
 - PERCENT_RANK() OVER with 14 rows in Calculation
 - PERCENT_RANK() OVER with 21 rows in Calculation
 - PERCENT_RANK and PARTITION BY
 - Cumulative Sum
 - Reset with a PARTITION BY Statement
 - Totals and Subtotals through Partition By
 - Moving SUM every 3-rows vs. a Continuous Average
 - Partition By Resets the Calculations
 - Moving Average
 - The Moving Window is Current Row and Preceding n
 - How Moving Average Handles the Order By
 - Moving Average every 3-rows Vs. a Continuous Average
 - The Partition By Statement
 - Partition By Resets an ANSI OLAP
 - Moving Difference
 - Moving Difference with Partition By
 - Finding a Value of a Column in the Next Row with MIN
 - Finding a Next Row Value with MIN and PARTITION BY
 - Finding Multiple Values of a Column in Upcoming Rows
 - Finding The Next Date using MAX

- COUNT OVER for a Sequential Number
- COUNT OVER using ROWS UNBOUNDED PRECEDING
- The MAX OVER Command
- MAX OVER with PARTITION BY Reset
- The MIN OVER Command
- The MIN OVER Command with PARTITION BY
- Finding Gaps Between Dates
- CSUM For Each Product_ID For the First Three Days
- FIRST_VALUE
- Using FIRST_VALUE
- Last_Value
- First and Last Value Difference
- Using LEAD
- Using LAG
- CUME_DIST
- CURRENT ROW AND UNBOUNDED FOLLOWING
- Different Windowing Options
- How Ntile Works
- Ntile
- Using Quantiles (Partitions of Four)
- Using Deciles (Partitions of Ten)
- MEDIAN
- PERCENTILE_CONT Function Description and Syntax
- Result Information About PERCENTILE_CONT
- PERCENTILE_CONT
- LISTAGG
- RATIO_TO_REPORT
- ANY_VALUE
- MODE
- Width_Bucket
- COUNT_IF
- Strings
 - UPPER and lower Functions

- The LENGTH Command Counts Characters
- LENGTH Does Not Work on Fixed Length Columns
- OCTET_LENGTH
- The TRIM Command Trims Leading and Trailing Spaces
- The RTRIM and LTRIM Command Trims Spaces
- Concatenation
- Concat and Concat_WS for Concatenation
- The SUBSTR and SUBSTRING Commands
- How SUBSTR Works with NO ENDING POSITION
- Using SUBSTR and CHAR_LENGTH Together
- The POSITION Command finds a Letters Position
- The POSITION Command is brilliant with SUBSTR
- CHARINDEX Finds a Letter's Position in a String
- CHARINDEX Command is Brilliant with SUBSTRING
- The CHARINDEX Command Using a Literal
- LPAD and RPAD
- The REPLACE Function
- REGEXP
- SOUNDEX Function to Find a Sound
- The REVERSE String Function
- The RIGHT Function
- The LEFT and RIGHT Functions
- The ASCII Function
- Interrogating the Data
 - The COALESCE Command
 - COALESCE is Equivalent to this CASE Statement
 - Some Great CAST (Convert And Store) Examples
 - A Rounding Example Using CAST
 - CAST will Round Values Up or Down
 - Valued Case vs. Searched Case
 - Combining Searched Case and Valued Case
 - The DECODE Command
 - DECODE

- A Trick for getting a Horizontal Case
- Put a CASE in the ORDER BY
- Using A Searched CASE Statement to Sort Logically
- Order By DECODE
- CASE Challenge
- View Functions
 - The Fundamentals of Views
 - Creating a Simple View to Restrict Sensitive Columns
 - Creating a Simple View to Restrict Rows
 - Creating a View to Join Tables Together
 - Sometimes We Create Views for Formatting
 - Basic Rules for Views
 - How to Modify a View
 - Creating a Secure View to Restrict DDL Viewing
 - The Exception to the ORDER BY Rule inside a View
 - Another Exception to the ORDER BY Rule is TOP
 - Derived Columns in a View Must Have a Column Alias
 - The Standard Way Most Aliasing is Done
 - Another Way to Alias Columns in a View CREATE
 - What Happens When a View Column gets Aliased Twice?
- Set Operators
 - Rules of Set Operators
 - INTERSECT Explained Logically
 - Rule 1 - Equal Number of Columns in Both SELECT Lists
 - Rule 2 - Top Query Handles all Aliases
 - Rule 3 - The Bottom Query does the ORDER BY
 - Intersect Challenge
 - Answer - Intersect Challenge
 - UNION Vs. UNION ALL
 - Using UNION ALL and Literals
 - Using UNION ALL for Speed in Merging Data Sets
 - Great Trick: Place your Set Operator in a Derived Table
 - A Great Example of how EXCEPT and MINUS work

- Using UNION to be same as GROUP BY GROUPING SETS
- USING Multiple SET Operators in a Single Request
- Changing the Order of Precedence with Parentheses
- Creating Tables
 - Table Types
 - SHOW TABLES With LIKE
 - TABLE STORAGE METRICS QUERY
 - Snowflake Data Types
 - More Snowflake Data Types
 - Show Databases and Table DDL Commands
 - Finding Constraints
 - The Basics of Creating a Table
 - Creating a Table
 - Creating Temporary and Transient Tables
 - Snowflake Clustering
 - Creating a Table with a Cluster Key
 - Creating a Table with a Cluster Key Function
 - Creating a Table with a Multi-Cluster Key
 - Altering a Table to Create or Drop a Cluster Key
 - Joining Tables Can Have the Same Cluster Keys for Speed
 - Creating Tables with a Primary Key/Foreign Key Relationship
 - Primary Key Does Not Enforce Uniqueness
 - A Table with a NOT NULL Constraint
 - AUTOINCREMENT
 - Restoring Data Using Time Travel Feature
 - Fail Safe
 - CREATE TABLE with Time Travel Days
 - Create Table LIKE
 - CREATE a Temporary TABLE using LIKE
 - CREATE TABLE AS (CTAS) Populates the Table With Data
 - CREATE TABLE AS (CTAS) Can Choose Certain Columns
 - CREATE a Temporary Table (CTAS) with Specific Columns
 - CREATE a Temporary Table AS (CTAS) Using a Join

- Create a Table IF NOT EXISTS
- A Table with a NOT NULL Constraint
- Create a Table with a Column Default Value
- Creating a Transient Schema
- Zero-Copy Cloning
- Cloning a Temporary Table
- CREATE TABLE AS (CTAS) Populates the Table With Data
- CREATE TABLE AS (CTAS) Can Choose Specific Columns
- CREATE Temporary Table AS (CTAS) with Specific Columns
- CREATE a Temporary Table AS (CTAS) Using a Join
- Inserts, Updates, and Deletes
 - INSERT Syntax
 - INSERT Example with Multiple Rows
 - Inserting Null Values into a Table
 - INSERT/SELECT Command
 - INSERT/SELECT to Build a Data Mart
 - UPDATE Examples
 - Example of Subquery UPDATE Command
 - MERGE
 - Drop and Undrop a Table
 - Drop and Undrop a Schema
 - Drop and Undrop a Database
 - Rename a Table
 - Deleting Rows in a Table
- Unstructured Data: JSON and XML
 - JSON and the Internet of Things
 - JSON Syntax Explanation
 - Building a Complex Object
 - Continuing to Build a Complex Object
 - Creating a Table and Inserting JSON Data
 - Querying the JSON Table
 - Performing a Flatten to Display Rows Like a Typical Table
 - Creating a View on a Flattened Table

- Further Building a Complex Object
- Creating a Table and Inserting JSON Data
- Querying the JSON Table
- Creating a JSON Table
- Inserting Two Objects Into a JSON Table
- How to Query a JSON Variant Data Type
- Inserting 10 Objects Into a JSON Table
- JSON Technique to CREATE a Table and INSERT Data
- How to Query JSON USING \$1
- How to Format JSON Data
- Querying Portions of Semi-Structured Data
- Querying Semi-Structured Data With a WHERE Clause
- Using the FLATTEN Function to Parse Arrays
- Using the FLATTEN Function to Parse Nested Arrays
- Extracting Values by Path Using the GET_PATH Function
- Creating an XML Table
- Insert Into An XML Table
- XMLGET and FLATTEN Function For XML Data
- PARSE_JSON Semi-Structured Data Function
- Describing a Table that Contains a Variant Data Type
- Inserting a Wide Variety of Data Into a Variant Data Type
- Using the TYPEOF Function
- Using the STRIP_NULL_VALUE Function
- User-Defined Functions (UDFs)
 - CREATE Function Syntax
 - Two Simple Functions with Hard-Coded Values
 - You Can Use a Function in a Join
 - You Can Use a Function in WHERE Clause
 - You Can Use a Function in the ORDER BY Clause
 - Creating a function with an Input Argument
 - Create a Function to get Factorial Numbers
 - Creating a function to Get Fibonacci Numbers
 - Creating a function with the Keyword TABLE

- Joining a Function with the Keyword TABLE
- JavaScript UDF with a Try Catch Block
- Using a JavaScript UDF with a Try Catch Block for Validation
- JavaScript UDF with IF, ELSE IF, and ELSE Try Catch Block
- DDL and DML Operations are Not Supported in Functions
- An EXAMPLE of a CASE Statement Using Two Functions
- Functions With Duplicate Names But Different Arguments
- You ALTER a Function for Three Reasons
- Examples of Altering a Function
- SHOW User Functions
- Information_Schema Functions
- GET DDL Function
- Conversion Functions and TRY_CAST
- Semi-Structured Data Functions
- Functions for Array and Object Creation and Manipulation
- Functions for Semi-structured Extraction and Conversion
- Functions for Semi-structured Extraction and Type Predicates
- Stored Procedures
 - Why Use Snowflake Stored Procedures?
 - Stored Procedure Syntax Using Javascript
 - Create and Execute Your First Javascript Stored Procedure
 - Create and Execute with an Argument Using Backticks
 - A Stored Procedure to INSERT Using Language SQL
 - SQL Stored Procedure INSERT Using Input Parameters
 - Creating a Snowflake Stored Procedure that Updates
 - Stored Procedure UPDATE Using Subquery
 - Stored Procedure Delete Using an Input Parameter
 - Snowflake Stored Procedure Control Structures
 - A Stored Procedure With IF and END IF Logic
 - A Stored Procedure With IF, ELSEIF, and ELSE Logic
 - A Stored Procedure With Simple Valued Case Logic
 - A Stored Procedure With Searched Case Logic
 - Using Loops in Stored Procedures

- Using a WHILE Loop
- Using a FOR Loop
- Using a FOR Reverse Loop
- Stored Procedure Workshop
- Creating a Complex Javascript Stored Procedure
- Inserting a Row Into a Table Using Javascript
- Updating a Row Into a Table Using Javascript
- Deleting a Row Into a Table Using Javascript
- DDL that Applies to Stored Procedures
- Describe Procedure
- Procedure Definition from Information_Schema
- Procedure Definition from GET_DDL
- Procedure Definition from GET_DDL using TRUE Keyword
- Alter Procedure
- Renaming a Procedure
- Dropping a Procedure
- Snowflake Data Types you must Convert to Javascript
- Stored Procedure to Get Methods from an Object
- Stored Procedure Javascript Methods Part 1
- Stored Procedure JavaScript Methods
- Stored Procedure Javascript Methods Part 2
- Stored Procedure to Get Methods from a Result Set Object
- Stored Procedure Javascript Result Set Methods
- Make Your Arguments in Upper Case
- Best Practice – Make Your Arguments in Upper Case
- How to Catch and Error using Try/Catch
- Using a WHILE Loop
- Creating a Stored Procedure to Get Fibonacci Numbers
- Create a Stored Procedure to get Factorial Numbers
- Using a WHILE Loop Example Result Set
- Line Continuation for Long SQL Statements
- Loading Rows From a Join Into a Table
- Using Stored Procedures to Produce Dynamic SQL

- Dynamic SQL Example to Truncate a Table
- Dynamic SQL Store Procedure that Calls Another Procedure
- Stored Procedures Privileges
- Information_Schema for a Database and a Schema
- A Stored Procedure to Show Privileges
- Finding Details About Procedures
- Granting and Revoking Usage on a Procedure to Another Role
- Statistical Aggregate Functions
 - The Stats Table
 - The KURTOSIS Function
 - The SKEW Function
 - The STDDEV_POP Function
 - STDDEV_POP Example
 - The VAR_POP Function
 - The VAR_SAMP Function
 - The CORR Function
 - Another CORR Example so you can Compare
 - The VARIANCE Function
 - The COVAR_POP Function
 - The COVAR_SAMP Function
 - Another COVAR_SAMP Example so you can Compare
 - 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
 - APPROX_COUNT_DISTINCT
- Mathematical Functions

- Numeric Manipulation Functions
- Conclusion