

**Databricks SQL**

**Course Number:** SQL-712
**Duration:** 3 days

**Overview**

This Databricks course teaches attendees the fundamentals of Databricks SQL, including how to run SQL statements, use analytics and window functions, and sort answer sets. Participants also learn advanced aggregation techniques, including using the different Databricks temporary options, date functions, and advanced aggregation techniques.

**Prerequisites**

All students must have prior SQL SELECT experience.

**Materials**

All Databricks SQL training students receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will write applications using the Databricks service running on the cloud.

**Objectives**

* Work with the Databricks database to run SQL statements
* Learn a wide variety of Databricks analytics and window functions
* Use advanced Order By methods to sort the answer sets retrieved
* Fully understand and use Joins and Subqueries
* Interrogate the data using Case, Coalesce, and Decode
* Be able to use all of the different Databricks temporary options fully
* Create tables and views on the Databricks system
* Utilize the many different Databricks date functions
* Use Aggregation and advanced Aggregation techniques
* Handle and manipulate Strings
* Learn and write SQL for advanced Statistical Aggregate Functions

**Outline**

* Introduction to SQL
	+ Introduction
	+ 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 First in Ascending Mode (Default)
	+ Order By with Nulls Last
	+ Order By with Nulls First
	+ 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
	+ Quiz – Can you Add a Minor Sort?
	+ Answer – Can you Add a Minor Sort?
	+ Order By Decode
	+ Quiz – Can you Add Two Minor Sorts Using Decode?
	+ Answer – Can you Add Two Minor Sorts Using Decode?
	+ How to ALIAS a Column name
	+ Using an Alias in the ORDER BY Clause
	+ A Missing Comma can by Mistake become an Alias
	+ Comments using Double Dashes are Single Line Comments
	+ Comments for Multi-Lines
	+ Comments for Multi-Lines As Double Dashes Per Line
* 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
	+ Is NULL
	+ IS Not Null
	+ 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
	+ Troubleshooting Character Data Continued
	+ Quiz – How many rows will return?
	+ Answer to Quiz – How many rows will return?
	+ 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
	+ Technique 2 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
	+ LIKE uses Wildcards Percent ‘%’ and Underscore ‘\_’
	+ Another Example of UPPER and LOWER
	+ Using LIKE for all Cases with Lower and Upper
	+ Using ILIKE Handle Case Issues
	+ LIKE command Underscore is Wildcard for one Character
	+ Finding Anyone Whose name End in 'Y'
	+ Escape Character in the LIKE Command changes Wildcards
	+ Escape Characters Turn off Wildcards in the LIKE Command
	+ The REPLACE Function
* Distinct, Group By and Top
	+ The Distinct Command
	+ Distinct vs. GROUP BY
	+ Quiz – How many rows come back from the Distinct?
	+ Answer – How many rows come back from the Distinct?
	+ Top Command
	+ Top Command and Order By
* Aggregation
	+ Quiz – You calculate the Answer Set in your Mind
	+ Quiz 2  – Calculate the Answer Set in your Mind
	+ Answer - Quiz 2  – Calculate the Answer Set in your Mind
	+ There are Five Aggregates
	+ Quiz – How many rows come back?
	+ Answer – How many rows come back?
	+ Casting a Data Type
	+ Troubleshooting Aggregates
	+ GROUP BY Delivers One Row Per Group
	+ GROUP BY dept\_no or GROUP BY Column Number
	+ 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
	+ GROUP BY GROUPING SETS
	+ GROUP BY ROLLUP
	+ GROUP BY ROLLUP Answer Set
	+ GROUP BY CUBE
	+ GROUP BY CUBE Answer Set
* Joining Tables
	+ Nexus Builds Your Join SQL Automatically
	+ A Two-Table Join Using Traditional Syntax
	+ Two-Table join using Traditional 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
	+ Quiz – Can You Finish the Join Syntax?
	+ Quiz – Can You Find the Error?
	+ Super Quiz – Can You Find the Difficult Error?
	+ Super Quiz – Can You Find the Difficult Error?
	+ Quiz – Which Rows from Both Tables Won’t Return?
	+ Left Outer Join
	+ Left Outer Join Results
	+ Right Outer Join
	+ Right Outer Join Example and Results
	+ Full Outer Join
	+ Full Outer Join Results
	+ Which Tables are Left Tables and Which are Right?
	+ Answer - Which Tables are Left Tables and Which are Right?
	+ INNER JOIN with Additional AND Clause
	+ ANSI INNER JOIN with Additional AND Clause
	+ ANSI INNER JOIN with Additional WHERE Clause
	+ OUTER JOIN with Additional WHERE Clause
	+ OUTER JOIN with Additional AND Clause
	+ The DREADED Product Join
	+ The DREADED Product Join Results
	+ Cartesian Product Join with Traditional Syntax
	+ Cartesian Product Join with ANSI Syntax
	+ The CROSS JOIN
	+ The CROSS JOIN Answer Set
	+ The Self Join
	+ The Self Join with ANSI Syntax
	+ An Associative Table is a Bridge that Joins Two Tables
	+ Quiz – Can you Write the 3-Table Join?
	+ Answer to Quiz – Can you Write the 3-Table Join?
	+ Quiz – Can you Write the 3-Table Join Using ANSI Syntax?
	+ Answer – Can you Write the 3-Table Join to ANSI Syntax?
	+ Quiz – Can you Place the ON Clauses at the End?
	+ Answer – Can you Place the ON Clauses at the End?
	+ The 5-Table Join – Logical Insurance Model
	+ Quiz - Write a Five Table Join Using ANSI Syntax
	+ Answer - Write a Five Table Join Using ANSI Syntax
	+ Quiz - Write a Five Table Join Using Traditional Syntax
	+ Answer - Write a Five Table Join Using Non-ANSI Syntax
	+ Quiz –Re-Write this putting the ON clauses at the END
* Date Functions
	+ Migrate Any Database to Databricks and Vice Versa
	+ Current\_Date
	+ Current\_Date, Current\_Timestamp, and Current\_Timezone
	+ Now() Function
	+ Add or Subtract From a Date
	+ Date Function
	+ To\_Date Function
	+ To\_Timestamp Function
	+ Add or Subtract Days From a Date
	+ Subtract Two Dates for a Difference in Days
	+ Subtract Two Dates for a Difference in Days
	+ MONTHS\_BETWEEN
	+ The ADD\_MONTHS Command
	+ Using the ADD\_MONTHS Command to Add 1 Year
	+ Using the ADD\_MONTHS Command to Add 5 Years
	+ The EXTRACT Command
	+ The EXTRACT Command
	+ EXTRACT from DATES and TIME
	+ Day, Month, Year, DayofMonth, DayofWeek, and DayofYear
	+ Using CASE and Extract to Reformat Dates
	+ Using CAST and SUBSTRING to Reformat Dates
	+ The Date\_Part Function
	+ Date\_Format Function
	+ More Date\_Format Examples
	+ Datediff Example
	+ Dateadd
	+ Incrementing Time Values Using the Dateadd Function
	+ Date\_Sub Function
	+ The Date\_Trunc Function
	+ Date\_Trunc Command With Time
	+ Date\_Trunc Command With Dates
	+ Last\_Day
	+ Advanced Tricks for Month
	+ Clever Tricks for Month
	+ Make\_Date
	+ Make\_Timestamp
	+ Using Day, Month, and Year intervals
	+ The Basics of a Simple Interval
	+ Determining if the Current\_Date is a Leap Year
	+ Determining if the Current\_Timestamp is a Leap Year
	+ Make\_Interval
	+ Try\_Divide Function
	+ Chapter 7 – Analytic and Window Functions
	+ Nexus Gives You Databricks Analytics for Free
	+ ROW\_NUMBER
	+ Quiz – How did the Row\_Number Reset?
	+ Answer – How did the Row\_Number Reset?
	+ QUALIFY
	+ Top Two Students Per class\_code Using a Derived Table
	+ RANK
	+ Dense\_Rank
	+ Getting RANK to Sort in DESC Order
	+ RANK() OVER and PARTITION BY
	+ RANK() OVER, PARTITION BY, and QUALIFY
	+ RANK() OVER and a Derived Table
	+ RANK() OVER and a WITH Derived Table
	+ RANK vs. DENSE\_RANK
	+ DENSE\_RANK() OVER and PARTITION BY
	+ PERCENT\_RANK() OVER with 14 rows in Calculation
	+ PERCENT\_RANK() OVER with 21 rows in Calculation
	+ PERCENT\_RANK() OVER and PARTITION BY
	+ Cumulative Sum
	+ Cumulative Sum with CAST
	+ Cumulative Sum – The Sort Explained
	+ Cumulative Sum – Rows Unbounded Preceding Explained
	+ Cumulative Sum – Making Sense of the Data
	+ Cumulative Sum – Major and Minor Sort Keys
	+ Reset with a PARTITION BY Statement
	+ Totals and Subtotals through Partition By
	+ Moving Sum
	+ Moving SUM every 3-rows Vs. a Continuous Average
	+ Partition By Resets the Calculations
	+ Moving Average
	+ The Moving Window is Current Row and Preceding
	+ How Moving Average Handles the Order By
	+ Quiz – How is that Total Calculated?
	+ Answer to Quiz – How is that Total Calculated?
	+ Quiz – How is that 4th Row Calculated?
	+ Answer to Quiz – How is that 4th Row Calculated?
	+ 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
	+ 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 The Next Date using MAX
	+ Finding Multiple Values of a Column in Upcoming Rows
	+ 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
	+ Different Windowing Options
	+ How Ntile Works
	+ Ntile in DESC Mode
	+ Ntile
	+ Ntile Percentile
	+ Another Ntile Example
	+ Using Quantiles (Partitions of Four)
	+ NTILE With a Partition
	+ NTILE With a Qualify Statement
	+ Using FIRST\_VALUE
	+ FIRST\_VALUE With Partitioning
	+ Daily\_Sales Minus FIRST\_VALUE With Partitioning
	+ FIRST\_VALUE With Partitioning
	+ FIRST\_VALUE After Sorting by the Highest Value
	+ FIRST\_VALUE with Partitioning
	+ Using LAST\_VALUE
	+ LAST\_VALUE – Current Row
	+ First\_Value Review
	+ Last\_Value Can Be Confusing
	+ Last\_Value Now Makes Sense
	+ Last\_Value With Partitioning
	+ Last\_Value And First\_Value with Partitioning
	+ First and Last Value Difference Between Today's Daily\_Sales
	+ Using LEAD
	+ Using LEAD with a PARTITION Statement
	+ Using LEAD With an Offset of 2
	+ Using LEAD With an Offset of 2 and a PARTITION
	+ Using LAG
	+ Using LAG with a PARTITION Statement
	+ Using Two LAG Statements
	+ Using LAG With an Offset of 2
	+ Using LAG With an Offset of 2 and a PARTITION
	+ CUME\_DIST and Qualify
	+ CUME\_DIST With Ties
	+ CUME\_DIST and Partition By
	+ CUME\_DIST With a Partition on the Sales\_Table
	+ CURRENT ROW AND UNBOUNDED FOLLOWING
	+ Different Windowing Options
	+ MEDIAN Example
	+ MEDIAN with Partitioning and a WHERE Clause
	+ MEDIAN with Partitioning
	+ PERCENTILE\_CONT Function Description and Syntax
	+ Final Result Information About PERCENTILE\_CONT
	+ PERCENTILE\_DISC Function Arguments
	+ PERCENTILE\_CONT Example
	+ PERCENTILE\_CONT Example with Percentage Change
	+ PERCENTILE\_CONT With PARTITION Example
	+ PERCENTILE\_DISC Function Description and Syntax
	+ PERCENTILE\_DISC Example
	+ PERCENTILE\_DISC Example with Percentage Change
	+ PERCENTILE\_DISC With PARTITION Example
* Temporary Tables
	+ CREATING A Derived Table
	+ Naming the Derived Table
	+ Aliasing the Column names in the Derived Table
	+ CREATING A Derived Table using the WITH Command
	+ 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
	+ Our Join Example Using The WITH Syntax
	+ An Example of Two Derived Tables in a Single Query
* Subqueries
	+ 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 Final Answer Set from the Subquery
	+ Quiz- Answer the Difficult Question
	+ Answer to Quiz - Answer the Difficult Question
	+ Should you use a Subquery or a Join?
	+ Quiz - Write the Subquery
	+ Answer to Quiz- Write the Subquery
	+ Quiz - Write the More Difficult Subquery
	+ Quiz – Write the Extreme Subquery
	+ Quiz - Write the Subquery with an Aggregate
	+ Quiz- Write the Correlated Subquery
	+ 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
	+ Answer - A Second Chance to Write a Correlated Subquery
	+ Answer - A Third Chance to Write a Correlated Subquery
	+ Answer – Last Chance to Write a Correlated Subquery
	+ Answer To Quiz – Write the Extreme Correlated Subquery
	+ NOT IN Subquery Returns Nothing when nulls are Present
	+ Fixing a NOT IN Subquery with Null Values
	+ Quiz- Write the NOT Subquery
	+ Quiz - Write the Subquery using a WHERE Clause
	+ Quiz- Write the Subquery with Two Parameters
	+ How the Double Parameter Subquery Works
	+ More on how the Double Parameter Subquery Works
	+ Another Example of a Double Parameter Subquery
	+ Quiz – Write the Triple Subquery
	+ Using a Correlated Exists
	+ How a Correlated Exists Matches Up
	+ The Correlated NOT Exists
* Strings
	+ UPPER and lower  Functions
	+ The Length Command Counts Characters
	+ LENGTH and TRIM Work on Fixed Length Columns
	+ The Char\_Length Command Counts Characters
	+ CHAR\_LENGTH and OCTET\_LENGTH
	+ The TRIM Command trims both Leading and Trailing Spaces
	+ The RTRIM and LTRIM Command Trims Spaces
	+ TRIM can also TRIM Characters
	+ 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
	+ The CHARINDEX Command is brilliant with SUBSTRING
	+ The CHARINDEX Command Using a Literal
	+ LPAD and RPAD
	+ The REPLACE Function
	+ The ASCII Function
	+ The Reverse String Function
	+ The RIGHT Function
	+ The LEFT and RIGHT Functions
	+ REGEXP Example for Whitespace Character
	+ REGEXP Example for Non-Whitespace
	+ REGEXP Example for [xyz]
	+ REGEXP Example Start of a String
	+ REGEXP Example End of a String
	+ REGEXP Example Matching Within a Range
	+ REGEXP\_REPLACE
	+ REGEXP\_REPLACE Example
	+ Another REGEXP\_REPLACE Example
	+ REGEXP\_LIKE
	+ RLIKE
	+ SOUNDEX Function to Find a Sound
* Interrogating the Data
	+ Quiz – Fill in the Answers for the NULLIF Command
	+ Answer – Fill in the Answers for the NULLIF Command
	+ COALESCE in a Real-World Example
	+ 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
	+ Decode
	+ A Trick for getting a Horizontal Case
	+ Put a Valued CASE in the ORDER BY
	+ Put a Searched CASE in the ORDER BY
	+ Put a Decode in the ORDER BY
	+ Extreme CASE Challenge
* Views
	+ 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
	+ Basic Rules for Views
	+ How to Modify a View
	+ The Exception to the ORDER BY Rule inside a View
	+ Derived Columns in a View Should Contain 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
	+ Quiz - Intersect Explained Logically
	+ Quiz - Union Explained Logically
	+ Quiz - Union ALL Explained Logically
	+ Quiz - Except Explained Logically
	+ Quiz - Testing Your Knowledge
	+ An Equal Number of Columns in both SELECT List
	+ The Top Query handles all Aliases
	+ 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 works
	+ USING Multiple SET Operators in a Single Request
	+ Changing the Order of Precedence with Parentheses
* Creating Tables
	+ Create Table Syntax
	+ Data Types
	+ Create Table Examples
	+ Best Practices for Partitioned Tables
	+ Describe Detail Tablename
	+ Not Null Constraint
	+ Create a Table IF NOT EXISTS
	+ Create Table AS (CTAS) Populates the Table With Data
	+ Create Table AS (CTAS) can Choose Certain Columns
* Data Manipulation Language (DML)
	+ INSERT Syntax # 1
	+ INSERT Syntax # 2
	+ INSERT Example with Multiple Rows
	+ Above we have inserted multiple rows and placed null values in some of them.
	+ INSERT/SELECT Command
	+ INSERT/SELECT to Build a Data Mart
	+ UPDATE Examples
	+ Deleting Rows in a Table
* Statistical Aggregate Functions
	+ The Stats Table
	+ The KURTOSIS Function
	+ The STDDEV\_POP Function
	+ The STDDEV\_SAMP Function
	+ The VAR\_POP Function
	+ The VAR\_SAMP Function
	+ A VAR\_SAMP Example
	+ The CORR Function
	+ A CORR Example
	+ Another CORR Example so you can Compare
	+ The VARIANCE Function
	+ The COVAR\_POP Function
	+ The COVAR\_SAMP Function
	+ The REGR\_INTERCEPT Function
	+ Another REGR\_INTERCEPT Example so you can Compare
	+ The REGR\_SLOPE Function
	+ A REGR\_SLOPE Example
	+ NOT IN Subquery Returns Nothing when nulls are Present
	+ The REGR\_AVGX Function
	+ A REGR\_AVGX Example
	+ Another REGR\_AVGX  Example so you can Compare
	+ The REGR\_AVGY   Function
	+ A REGR\_AVGY Example
	+ Quiz- Write the Subquery with Two Parameters
	+ The REGR\_COUNT Function
	+ A REGR\_COUNT Example
	+ The REGR\_R2 Function
	+ A REGR\_R2 Example
	+ The REGR\_SXX Function
	+ Answer to Quiz – Write the Triple Subquery
	+ The REGR\_SXY Function
	+ A REGR\_SXY Example
	+ The REGR\_SYY Function
	+ A REGR\_SYY Example
	+ Using GROUP BY
	+ APPROX\_COUNT\_DISTINCT
* Mathematical Functions
	+ Numeric Manipulation Functions
* Conclusion