

Kognitio Architecture and SQL

Course Summary

Description

In this course, students will learn the Kognitio Architecture and Kognitio SQL starting at the most basic level and going to the most advanced level with many examples.

Objectives

By the end of this course, students will have a deeper knowledge and understanding of the Kognitio Architecture and Kognitio SQL and how to write it.

Topics

- Basic SQL Functions
- The WHERE Clause
- Distinct Vs. Group By
- Aggregation Function
- Join Functions
- Date Functions
- OLAP Functions
- Temporary Tables
- Sub-query Functions
- Strings
- Interrogating the Data
- View Functions
- Set Operators
- Data Manipulation Language (DML)
- Statistical Aggregate Functions

Audience

This course is designed for anyone who has a desire to learn Kognitio SQL from beginners to an advanced audience. This course is completely customizable by the client.

Prerequisites

There are no prerequisites for this course.

Duration

Two to three days

Kognitio Architecture and SQL

Course Outline

- | | |
|--|--|
| <p>I. Introduction to the Kognitio Architecture</p> <ul style="list-style-type: none"> A. What is Parallel Processing? B. The Basics of a Single Computer C. Data in Memory is fast as Lightning D. Parallel Processing Of Data E. Kognitio is an In-Memory System F. Kognitio has Three Table Distribution Options G. Kognitio has Linear Scalability H. Nexus is Now Available for Kognitio <p>II. Kognitio Table Structures</p> <ul style="list-style-type: none"> A. Kognitio has Three Table Distribution Options B. A Table that is distributed via a Round Robin Technique C. Round Robin Technique is the Default D. Random Distribution E. A Table that is distributed by Hash F. Tables that join are excellent candidates for Hashed Tables G. Hash Distribution H. A Table that is distributed by Hash by Multiple Columns I. The Reasons for a Multi-Column HASHED Distribution Key J. Creating a Table that is replicated across all Nodes K. Replicated Distribution L. The Concept is all about the Joins M. Kognitio allows you to create Images N. Creating a Table Image to place a Table in Memory O. Partitioning an Image P. Partitioning an Image View Q. CREATE OR REPLACE TABLE IMAGE R. DEFRAG TABLE IMAGE <p>III. Nexus for Kognitio</p> <ul style="list-style-type: none"> A. Nexus is Available on the Cloud B. Nexus Queries Every Major System | <ul style="list-style-type: none"> C. How to Use Nexus D. Why is Nexus Special? Visualization and Automatic SQL E. Why is Nexus Special? Cross-System Joins F. Why is Nexus Special? The Amazing Hub System G. Why is Nexus Special? Save Answer Sets as Tables H. Why is Nexus Special? Automated Data Movement I. Why is Nexus Special? Nexus makes the Servers Talk Directly J. What Makes Nexus Special? The Garden of Analysis K. The Garden of Analysis Grouping Sets Tab L. The Garden of Analysis - Grouping Sets Answer Sets M. The Garden of Analysis – Join Tab N. The Garden of Analysis – Charts/Graphs Tab O. The Garden of Analysis – Dynamic Charts Tab P. The Garden of Analysis – Dashboard Tab Q. Getting to the Super Join Builder R. The Super Join Builder is the First Entry in the Menu S. The Super Join Builder Shows Tables Visually T. Using the Add Join Button U. What to Do When No Tables are Joinable? V. Drag a Joinable Object into the Super Join Builder W. You Will See the Add Custom Join Window X. Defining the Join Columns Y. Your Tables Will Appear Together Z. Select the Columns You Want on the Report AA. Check out the SQL Tab to See the SQL that has been built BB. SQL Tab CC. Hit Execute to get the Report inside the Super Join Builder |
|--|--|

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|------|---|------------------------------|--|
| DD. | The Report is delivered inside the Super Join Builder | CCC. | How to join Kognitio, Oracle and SQL Server Tables |
| EE. | Let's Join Two Tables Again | DDD. | The Kognitio Table is now in the Super Join Builder |
| FF. | The Tabs of the Super Join Builder Philosophy – One Query | EEE. | Drag the Joining Oracle Table to the Super Join Builder |
| GG. | The Tabs of the Super Join Builder – Objects Tab | FFF. | Defining the Join Columns |
| HH. | The Tabs of the Super Join Builder – Columns Tab) | GGG. | Choose the Columns You Want on Your Report |
| II. | The Tabs of the Super Join Builder – Sorting Tab | HHH. | Let's Add a SQL Server Table to our Teradata and Oracle Join |
| JJ. | The Tabs of the Super Join Builder – Joins Tab | III. | Defining the Join Columns |
| KK. | The Tabs of the Super Join Builder – SQL Tab | JJJ. | All Three Tables are now in the Super Join Builder |
| LL. | The Tabs of the Super Join Builder – Metadata Tab | KKK. | Change the Hub and Run the Join on Oracle |
| MM. | The Tabs of the Super Join Builder – Analytics Tab | LLL. | Change the Hub and Run the Join on SQL Server |
| NN. | The Tabs of the SJB – Analytics Tab – OLAP Screen | MMM. | Simply Amazing - Change the Hub to the Garden of Analysis |
| OO. | Getting a Simple CSUM in the Analytics Tab – OLAP | NNN. | Have the Answer Set Saved Automatically to Any System |
| PP. | Getting a Simple CSUM – The SQL Automatically Generated | OOO. | Saving the Answer Set to an Oracle or SQL Server System |
| QQ. | The Answer Set of the CSUM | PPP. | Saving the Answer Set to a Kognitio System |
| RR. | Getting all of the OLAP functions in the Analytics Tab | QQQ. | Saving the Answer Set to a Teradata System |
| SS. | A Five Table Join Using the Menu | | |
| TT. | The First Table is placed in the Super Join Builder | IV. The Basics of SQL | |
| UU. | Using the Add Join Cascading Menu | A. | Introduction |
| VV. | All Five Tables Are In the Super Join Builder | B. | Setting the Default Schema |
| WW. | A Five Table Join Two Steps (Cube) | C. | SELECT * (All Columns) in a Table |
| XX. | Choose Cube with Columns from the Left Top of the Table | D. | Fully Qualifying a Database, Schema and Table |
| YY. | All Tables are Cubed (Joined Together Instantly) | E. | SELECT Specific Columns in a Table |
| ZZ. | Choose Cube and then Choose Your Columns | F. | Commas in the Front or Back? |
| AAA. | Create Cube - Tables Are Joined Without Columns Selected | G. | Place your Commas in front for better Debugging Capabilities |
| BBB. | Create Cube – Select the Columns You Want on the Report | H. | Sort the Data with the ORDER BY Keyword |
| | | I. | ORDER BY Defaults to Ascending |
| | | J. | Use the Name or the Number in your ORDER BY Statement |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|----|---|-----|---|
| K. | Two Examples of ORDER BY using Different Techniques | M. | OR in the WHERE Clause |
| L. | Changing the ORDER BY to Descending Order | N. | Troubleshooting Or |
| M. | NULL Values sort Last in Ascending Mode (Default) | O. | Troubleshooting Character Data |
| N. | NULL Values sort First in Descending Mode (DESC) | P. | Using Different Columns in an AND Statement |
| O. | Major Sort vs. Minor Sorts | Q. | Quiz – How many rows will return? |
| P. | Multiple Sort Keys using Names vs. Numbers | R. | Answer to Quiz – How many rows will return? |
| Q. | Sorts are Alphabetical, NOT Logical | S. | What is the Order of Precedence? |
| R. | Using A CASE Statement to Sort Logically | T. | Using Parentheses to change the Order of Precedence |
| S. | How to ALIAS a Column Name | U. | Using an IN List in place of OR |
| T. | A Missing Comma can by Mistake become an Alias | V. | The IN List is an Excellent Technique |
| U. | Comments using Double Dashes are Single Line Comments | W. | IN List vs. OR brings the same Results |
| V. | Comments for Multi-Lines | X. | The IN List Can Use Character Data |
| W. | Comments for Multi-Lines as Double Dashes per Line | Y. | Using a NOT IN List |
| X. | A Great Technique for Comments to Look for SQL Errors | Z. | Null Values in a NOT IN List Bring Back No Rows |
| V. | The WHERE Clause | AA. | A Technique for Handling Nulls with a NOT IN List |
| A. | The WHERE Clause limits Returning Rows | BB. | BETWEEN is Inclusive |
| B. | Double Quoted Aliases are for Reserved Words and Spaces | CC. | NOT BETWEEN is Also Inclusive |
| C. | Character Data needs Single Quotes in the WHERE Clause | DD. | LIKE uses Wildcards Percent '%' and Underscore '_' |
| D. | Character Data needs Single Quotes, but Numbers Don't | EE. | LIKE command Underscore is Wildcard for one Character |
| E. | Comparisons against a Null Value | FF. | The ilike Command |
| F. | NULL means Unknown Data so Equal (=) won't return rows | GG. | LIKE Command Works Differently on Char Vs Varchar |
| G. | Use IS NULL or IS NOT NULL when dealing with NULLs | HH. | Troubleshooting LIKE Command on Character Data |
| H. | NULL is UNKNOWN DATA so NOT Equal won't Work | II. | Introducing the TRIM Command |
| I. | Use IS NULL or IS NOT NULL when dealing with NULLs | JJ. | Introducing the RTRIM Command |
| J. | Using Greater Than or Equal To (>=) | KK. | Quiz – What Data is Left Justified and what is Right? |
| K. | AND in the WHERE Clause | LL. | Numbers are Right Justified and Character Data is Left |
| L. | Troubleshooting AND | MM. | Answer – What Data is Left Justified and what is Right? |
| | | NN. | An example of Data with Left and Right Justification |
| | | OO. | A Visual of CHARACTER Data vs. VARCHAR Data |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|-------------|--|--------------|--|
| PP. | Escape Character in the LIKE Command changes Wildcards | M. | Limiting Rows and Improving Performance with WHERE |
| QQ. | Escape Characters Turn off Wildcards in the LIKE Command | N. | WHERE Clause in Aggregation limits unneeded Calculations |
| RR. | Quiz – Turn off that Wildcard | O. | Keyword HAVING tests Aggregates after they are totaled |
| SS. | ANSWER – To Find that Wildcard | P. | Keyword HAVING is like an Extra WHERE Clause for Totals |
| TT. | Using ILIKE with an AND Clause to Find Multiple Letters | Q. | Keyword HAVING tests Aggregates after they are totaled |
| UU. | Using ILIKE with an OR Clause to Find Either Letters | R. | Getting the Average Values per Column |
| VV. | Keywords | | |
| VI. | Distinct, Group By and TOP | VIII. | Join Functions |
| A. | The Distinct Command | A. | A Two-Table Join Using Traditional Syntax |
| B. | Distinct vs. GROUP BY | B. | A two-table join using Non-ANSI Syntax with Table Alias |
| C. | Quiz – How many rows come back from the Distinct? | C. | You Can Fully Qualify All Columns |
| D. | Answer – How many rows come back from the Distinct? | D. | A two-table join using ANSI Syntax |
| E. | TOP Command | E. | Both Queries have the same Results and Performance |
| F. | TOP Command with an ORDER BY Statement | F. | Quiz – Can You Finish the Join Syntax? |
| G. | Just Place the TOP n in front of any Query | G. | Answer to Quiz – Can You Finish the Join Syntax? |
| VII. | Aggregation | H. | Quiz – Can You Find the Error? |
| A. | Quiz – You calculate the Answer Set in your own Mind | I. | Answer to Quiz – Can You Find the Error? |
| B. | Answer – You calculate the Answer Set in your own Mind | J. | Super Quiz – Can You Find the Difficult Error? |
| C. | Quiz – You calculate the Answer Set in your own Mind | K. | Answer to Super Quiz – Can You Find the Difficult Error? |
| D. | Answer – You calculate the Answer Set in your own Mind | L. | Quiz – Which rows from both tables won't return? |
| E. | The 3 Rules of Aggregation | M. | Answer to Quiz – Which rows from both tables won't return? |
| F. | There are Five Aggregates | N. | LEFT OUTER JOIN |
| G. | Quiz – How many rows come back? | O. | LEFT OUTER JOIN Results |
| H. | Answer – How many rows come back? | P. | RIGHT OUTER JOIN |
| I. | Troubleshooting Aggregates | Q. | RIGHT OUTER JOIN Example and Results |
| J. | GROUP BY when Aggregates and Normal Columns Mix | R. | FULL OUTER JOIN |
| K. | GROUP BY delivers one row per Group | S. | FULL OUTER JOIN Results |
| L. | GROUP BY Dept_No or GROUP BY 1 the same thing | T. | Which Tables are the Left and which Tables are Right? |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|-----|---|---------------------------|---|
| U. | Answer - Which Tables are the Left and which are the Right? | VV. | Quiz – Can you write the 3-Table Join to ANSI Syntax? |
| V. | INNER JOIN with Additional AND Clause | WW. | Answer – Can you write the 3-Table Join to ANSI Syntax? |
| W. | ANSI INNER JOIN with Additional AND Clause | XX. | Quiz – Can you Place the ON Clauses at the End? |
| X. | ANSI INNER JOIN with Additional WHERE Clause | YY. | Answer – Can you Place the ON Clauses at the End? |
| Y. | OUTER JOIN with Additional WHERE Clause | ZZ. | The 5-Table Join – Logical Insurance Model |
| Z. | OUTER JOIN with Additional AND Clause | AAA. | Quiz - Write a Five Table Join Using ANSI Syntax |
| AA. | OUTER JOIN with Additional AND Clause Results | BBB. | Answer - Write a Five Table Join Using ANSI Syntax |
| BB. | Quiz – Why is this considered an INNER JOIN? | CCC. | Quiz - Write a Five Table Join Using Non-ANSI Syntax |
| CC. | Evaluation Order for Outer Queries | DDD. | Answer - Write a Five Table Join Using Non-ANSI Syntax |
| DD. | The DREADED Product Join | EEE. | Quiz –Re-Write this putting the ON clauses at the END |
| EE. | The DREADED Product Join Results | FFF. | Answer –Re-Write this putting the ON clauses at the END |
| FF. | The Horrifying Cartesian Product Join | | |
| GG. | The ANSI Cartesian Join will ERROR | IX. Date Functions | |
| HH. | Quiz – Do these Joins Return the Same Answer Set? | A. | Current_Date |
| II. | Answer – Do these Joins Return the Same Answer Set? | B. | Current_Date and Current_Time |
| JJ. | The CROSS JOIN | C. | Current_Date and Current_Timestamp |
| KK. | The CROSS JOIN Answer Set | D. | Current_Timestamp with Milliseconds |
| LL. | The Self Join | E. | Current_Timestamp with Microseconds |
| MM. | The Self Join with ANSI Syntax | F. | Current_Timestamp and SYSDATE are Synonyms |
| NN. | Quiz – Will both queries bring back the same Answer Set? | G. | The Now Function |
| OO. | Answer – Will both queries bring back the same Answer Set? | H. | Adding Days, Weeks and Months to a Date |
| PP. | Quiz – Will both queries bring back the same Answer Set? | I. | Add or Subtract Days from a date |
| QQ. | Answer – Will both queries bring back the same Answer Set? | J. | The EXTRACT Command |
| RR. | How would you join these two tables? | K. | EXTRACT from DATES and TIME |
| SS. | An Associative Table is a Bridge that Joins Two Tables | L. | EXTRACT of the Month on Aggregate Queries |
| TT. | Quiz – Can you write the 3-Table Join? | M. | Deriving a Timestamp from a Date and Time |
| UU. | Answer to quiz – Can you write the 3-Table Join? | N. | Formatting Dates and Dollar Amounts |
| | | O. | TO_CHAR Example that is Impressive |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|-----------|--|-----|---|
| P. | TO_CHAR Example that is Amazing | P. | Answer to Quiz – Write the Extreme Subquery |
| Q. | TO_CHAR Example to get Seconds since Midnight | Q. | Quiz- Write the Subquery with an Aggregate |
| R. | TO_CHAR Example that is ahead of its Time | R. | Answer to Quiz- Write the Subquery with an Aggregate |
| S. | TO_DATE | S. | Quiz- Write the Correlated Subquery |
| T. | TO_TIME | T. | Answer to Quiz- Write the Correlated Subquery |
| U. | TO_TIMESTAMP | U. | The Basics of a Correlated Subquery |
| V. | Using CASE and Extract to reformat Dates | V. | The Top Query always runs first in a Correlated Subquery |
| W. | Using CAST and SUBSTRING to reformat Dates | W. | Correlated Subquery Example vs. a Join with a Derived Table |
| X. | Using the DAYOFWEEK and the DECODE Function | X. | Quiz- A Second Chance to Write a Correlated Subquery |
| Y. | Intervals | Y. | Answer - A Second Chance to Write a Correlated Subquery |
| Z. | More Interval Examples | Z. | Quiz- A Third Chance to Write a Correlated Subquery |
| AA. | TO_CHAR Details | AA. | Answer - A Third Chance to Write a Correlated Subquery |
| BB. | TO_CHAR Details Continued | BB. | Quiz- Last Chance to Write a Correlated Subquery |
| CC. | TO_CHAR, TO_DATE, TO_TIME and TO_TIMESTAMP | CC. | Answer – Last Chance to Write a Correlated Subquery |
| X. | Sub-query Functions | DD. | Quiz – Write the Extreme Correlated Subquery |
| A. | An IN List is much like a Subquery | EE. | Answer To Quiz – Write the Extreme Correlated Subquery |
| B. | An IN List Never has Duplicates – Just like a Subquery | FF. | Quiz- Write the NOT Subquery |
| C. | An IN List Ignores Duplicates | GG. | Answer to Quiz- Write the NOT Subquery |
| D. | The Subquery | HH. | Quiz- Write the Subquery using a WHERE Clause |
| E. | The Three Steps of How a Basic Subquery Works | II. | Answer - Write the Subquery using a WHERE Clause |
| F. | These are Equivalent Queries | JJ. | Quiz- Write the Subquery with Two Parameters |
| G. | The Final Answer Set from the Subquery | KK. | Answer to Quiz- Write the Subquery with Two Parameters |
| H. | Quiz- Answer the Difficult Question | LL. | How the Double Parameter Subquery Works |
| I. | Answer to Quiz- Answer the Difficult Question | MM. | More on how the Double Parameter Subquery Works |
| J. | Should you use a Subquery of a Join? | NN. | Quiz – Write the Triple Subquery |
| K. | Quiz- Write the Subquery | | |
| L. | Answer to Quiz- Write the Subquery | | |
| M. | Quiz- Write the More Difficult Subquery | | |
| N. | Answer to Quiz- Write the More Difficult Subquery | | |
| O. | Quiz – Write the Extreme Subquery | | |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|---------------------------|--|-----|--|
| OO. | Answer to Quiz – Write the Triple Subquery | S. | Reset with a PARTITION BY Statement |
| PP. | Quiz – How many rows return on a NOT IN with a NULL? | T. | PARTITION BY only Resets a Single OLAP not ALL of them |
| QQ. | Answer – How many rows return on a NOT IN with a NULL? | U. | Moving SUM |
| RR. | How to handle a NOT IN with Potential NULL Values | V. | ANSI Moving Window is Current Row and Preceding n Rows |
| SS. | IN is equivalent to =ANY | W. | How ANSI Moving SUM Handles the Sort |
| TT. | Using a Correlated Exists | X. | Quiz – How is that Total Calculated? |
| UU. | How a Correlated Exists matches up | Y. | Answer to Quiz – How is that Total Calculated? |
| VV. | The Correlated NOT Exists | Z. | Moving SUM every 3-rows Vs a Continuous Average |
| WW. | Quiz – How many rows come back from this NOT Exists? | AA. | PARTITION BY Resets an ANSI OLAP |
| XX. | Answer – How many rows come back from this NOT Exists? | BB. | Moving Average |
| XI. OLAP Functions | | CC. | Moving Average with a Moving Window of 3 |
| A. | The Row_Number Command | DD. | The Moving Window is Current Row and Preceding |
| B. | Using a Derived Table and Row_Number | EE. | How Moving Average Handles the Sort |
| C. | Finding the First Occurrence | FF. | Quiz – How is that Total Calculated? |
| D. | Finding the Last Occurrence | GG. | Answer to Quiz – How is that Total Calculated? |
| E. | Quiz – How did the Row_Number Reset? | HH. | Quiz – How is that 4th Row Calculated? |
| F. | Answer – How did the Row_Number Reset? | II. | Answer to Quiz – How is that 4th Row Calculated? |
| G. | RANK Defaults to Ascending Order | JJ. | Moving Average every 3-rows Vs a Continuous Average |
| H. | Getting RANK to Sort in DESC Order | KK. | PARTITION BY Resets an ANSI OLAP |
| I. | RANK OVER and PARTITION BY | LL. | Moving Difference |
| J. | RANK and DENSE RANK | MM. | Moving Difference using ANSI Syntax with Partition By |
| K. | CSUM | NN. | PERCENT_RANK OVER |
| L. | CSUM – The Sort Explained | OO. | PERCENT_RANK OVER with 14 rows in Calculation |
| M. | CSUM – Rows Unbounded Preceding Explained | PP. | PERCENT_RANK OVER with 21 rows in Calculation |
| N. | CSUM – Making Sense of the Data | QQ. | COUNT OVER for a Sequential Number |
| O. | CSUM – Making Even More Sense of the Data | RR. | Troubleshooting COUNT OVER |
| P. | CSUM – The Major and Minor Sort Key(s) | | |
| Q. | The ANSI CSUM – Getting a Sequential Number | | |
| R. | Troubleshooting the ANSI OLAP on a GROUP BY | | |

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|-------------|---|-----|--|
| SS. | Quiz – What caused the COUNT OVER to Reset? | C. | Naming the Derived Table |
| TT. | Answer to Quiz – What caused the COUNT OVER to Reset? | D. | Aliasing the Column Names in the Derived Table |
| UU. | The MAX OVER Command | E. | Multiple Ways to Alias the Columns in a Derived Table |
| VV. | MAX OVER with PARTITION BY Reset | F. | CREATING a Derived Table using the WITH Command |
| WW. | Troubleshooting MAX OVER | G. | The Same Derived Query shown Three Different Ways |
| XX. | The MIN OVER Command | H. | Most Derived Tables Are Used To Join To Other Tables |
| YY. | Troubleshooting MIN OVER | I. | The Three Components of a Derived Table |
| ZZ. | Finding a Value of a Column in the Next Row with MIN | J. | Visualize This Derived Table |
| AAA. | Quiz – Fill in the Blank | K. | A Derived Table and CAST Statements |
| BBB. | Answer – Fill in the Blank | L. | A Derived example Using the WITH Syntax |
| CCC. | Ordered Analytics OVER | M. | Quiz - Answer the Questions |
| DDD. | CURRENT ROW AND UNBOUNDED FOLLOWING | N. | Answer to Quiz - Answer the Questions |
| EEE. | Different Windowing Options | O. | Clever Tricks on Aliasing Columns in a Derived Table |
| FFF. | The CSUM for Each Product_Id and the Next Start Date | P. | An example of Two Derived Tables in a Single Query |
| GGG. | How Ntile Works | Q. | MULTIPLE Derived Tables using the WITH Command |
| HHH. | Ntile | R. | Finding the First Occurrence |
| III. | Ntile Percentile | S. | Finding the Last Occurrence |
| JJJ. | Another Ntile example | T. | Three Steps to Creating a Temporary Table |
| KKK. | Using Quantiles (Partitions of Four) | U. | Two Versions of Creating a Temporary Table |
| LLL. | NTILE Using a Value of 10 | V. | ON COMMIT DELETE ROWS is the Kognitio Default |
| MMM. | NTILE - Tertiles with a PARTITION BY | W. | ON COMMIT DELETE ROWS |
| NNN. | FIRST_VALUE | X. | Important Temporary Table Information |
| OOO. | FIRST_VALUE after Sorting by the Highest Value | Y. | How to Use the ON COMMIT DELETE ROWS Option |
| PPP. | FIRST_VALUE with Partitioning | Z. | Create Table AS |
| QQQ. | LAST_VALUE | AA. | Creating a Temporary Table Using a CTAS that Joins Multiple Tables |
| RRR. | Using LEAD | BB. | Create Table LIKE |
| SSS. | Using LEAD With and Offset of 2 | | |
| TTT. | LEAD | | |
| UUU. | LEAD With Partitioning | | |
| VVV. | Using LAG | | |
| WWW. | Using LAG with an Offset of 2 | | |
| XXX. | LAG | | |
| YYY. | LAG with Partitioning | | |
| ZZZ. | SUM (SUM(n)) | | |
| XII. | Temporary Tables | | |
| A. | There are Two Types of Temporary Tables | | |
| B. | CREATING A Derived Table | | |

Kognitio Architecture and SQL

Course Outline (cont'd)

XIII. Strings

- A. The LENGTH Command Counts Characters
- B. The LENGTH Command – Spaces can Count too
- C. The LENGTH Command Counts Trailing Spaces
- D. The LENGTH Command and TRIM
- E. UPPER and LOWER Commands
- F. Using the LOWER Command
- G. Using the UPPER Command
- H. Non-Letters are Unaffected by UPPER and LOWER
- I. The CHARACTERS Command Counts Characters
- J. The CHARACTERS Command and Character Data
- K. The CHARACTERS and TRIM Commands
- L. LENGTH, CHARACTER_LENGTH and OCTET_LENGTH
- M. The TRIM Command trims both Leading and Trailing Spaces
- N. How to TRIM only the Trailing Spaces
- O. Concatenation
- P. A Visual of the TRIM Command Using Concatenation
- Q. Trim and Trailing is Case Sensitive
- R. How to TRIM Trailing Letters
- S. The SUBSTRING Command
- T. SUBSTRING and SUBSTR are equal, but use different syntax
- U. How SUBSTRING Works with NO ENDING POSITION
- V. Using SUBSTRING to move backwards
- W. How SUBSTRING Works with a Starting Position of -1
- X. How SUBSTRING Works with an Ending Position of 0
- Y. An example using SUBSTRING, TRIM and CHAR Together
- Z. The POSITION Command finds a Letters Position

- AA. Concatenation
- BB. Concatenation and SUBSTRING
- CC. Four Concatenations Together
- DD. Troubleshooting Concatenation

XIV. Interrogating the Data

- A. Quiz – What would the Answer be?
- B. Answer to Quiz – What would the Answer be?
- C. The NULLIF Command
- D. Quiz – Fill in the Answers for the NULLIF Command
- E. Answer– Fill in the Answers for the NULLIF Command
- F. The COALESCE Command – Fill In the Answers
- G. The COALESCE Answer Set
- H. The COALESCE Command
- I. The COALESCE Answer Set
- J. The COALESCE Quiz
- K. Answer - The COALESCE Quiz
- L. The Basics of CAST (Convert and Store)
- M. A CAST (Convert and Store) example
- N. Quiz - The Basics of the CASE Statements
- O. Answer to Quiz - The Basics of the CASE Statements
- P. Using an ELSE in the Case Statement
- Q. Using an ELSE as a Safety Net
- R. Rules for a Valued Case Statement
- S. Rules for a Searched Case Statement
- T. Valued Case Vs. A Searched Case
- U. Quiz - Valued Case Statement
- V. Answer - Valued Case Statement
- W. Quiz - Searched Case Statement
- X. Answer - Searched Case Statement
- Y. The CASE Challenge
- Z. The CASE Challenge Answer
- AA. Combining Searched Case and Valued Case

Kognitio Architecture and SQL

Course Outline (cont'd)

- BB. A Trick for getting a Horizontal Case
- CC. Put a CASE in the ORDER BY
- DD. Nested Case
- XV. Set Operators Functions**
 - A. Rules of Set Operators
 - B. INTERSECT Explained Logically
 - C. UNION Explained Logically
 - D. UNION ALL Explained Logically
 - E. EXCEPT/MINUS Explained Logically
 - F. EXCEPT Explained Logically
 - G. An Equal Amount of Columns in both SELECT List
 - H. Columns in the SELECT list should be from the same Domain
 - I. The Top Query handles all Aliases
 - J. The Bottom Query does the ORDER BY
 - K. Great Trick: Place your Set Operator in a Derived Table
 - L. UNION Vs UNION ALL
 - M. Using UNION ALL and Literals
 - N. A Great example of how EXCEPT works
 - O. Quiz – Build that Query
 - P. Answer To Quiz – Build that Query
 - Q. USING Multiple SET Operators in a Single Request
 - R. Changing the Order of Precedence with Parentheses
 - S. Using UNION ALL for speed in Merging Data Sets
- XVI. View Functions**
 - A. The Fundamentals of Views
 - B. Creating a Simple View to Restrict Sensitive Columns
 - C. Creating a Simple View to Restrict Rows
 - D. Basic Rules for Views
 - E. Exception to the ORDER BY Rule inside a View
 - F. Views sometimes CREATED for Formatting
- G. Creating a View to Join Tables Together
- H. Another Way to Alias Columns in a View CREATE
- I. The Standard Way Most Aliasing is done
- J. What Happens When Both Aliasing Options Are Present
- K. Resolving Aliasing Problems in a View CREATE
- L. Answer to Resolving Aliasing Problems in a View CREATE
- M. Aggregates on View Aggregates
- XVII. Table Create and Data Types**
 - A. Kognitio Has Three Table Distribution Options
 - B. A Table that is distributed via a Round Robin Technique
 - C. Round Robin Technique is the Default
 - D. A Table that is distributed by Hash
 - E. Tables that join are excellent candidates for Hashed Tables
 - F. A Table that is distributed by Hash by Multiple Columns
 - G. The Reasons for a Multi-Column HASHED Distribution Key
 - H. Creating a Table that is replicated across all Nodes
 - I. The Concept is all about the Joins
 - J. Creating a Table with Primary Key
 - K. Creating a Table with a UNIQUE constraint
 - L. How to create tables with Referential Integrity
 - M. Not Null Constraints
 - N. Creating a Table with Default Values
 - O. Creating a Table with a CHECK Constraint
 - P. Creating a Global Temporary Table
 - Q. Important Temporary Table Information
 - R. Creating a Table Image to place a Table in Memory

Kognitio Architecture and SQL

Course Outline (cont'd)

- | | | | |
|--|--|--|---|
| S. | CREATE OR REPLACE TABLE
IMAGE | C. | The Eight Rules to Reading an
EXPLAIN Plan |
| T. | DEFRAG TABLE IMAGE | D. | Interpreting Keywords in an
EXPLAIN Plan |
| U. | Not Null Constraints | E. | Interpreting an EXPLAIN Plan |
| V. | Unique Constraints | F. | A Single Segment Retrieve – The
Fastest Query |
| W. | Primary Key Constraints | G. | EXPLAIN With an ORDER BY
Statement |
| X. | Check Constraints | H. | EXPLAIN ANALYZE |
| Y. | Create Table AS WITH DATA or
WITH NO DATA | I. | EXPLAIN With a Range Query on
a Table Partitioned By Day |
| Z. | Another Version of Create Table
AS | J. | EXPLAIN That Uses a B-Tree
Index Scan |
| AA. | CREATE Table FOR and FROM | K. | EXPLAIN That Uses a Bitmap
Scan |
| BB. | Create Table LIKE | L. | EXPLAIN With a Simple
Subquery |
| CC. | String Data Types | M. | EXPLAIN With a Columnar Query |
| DD. | Numeric Data Types | N. | EXPLAIN With a Clustered Index |
| EE. | Date, Time and Timestamp Data
Types | O. | The Most Important Concept for
Joins is the Distribution Key |
| XVIII. Data Manipulation Language (DML) | | P. | EXPLAIN With Join that has to
Move Data |
| A. | INSERT Syntax # 1 | Q. | Changing the Join Query
Changes the EXPLAIN Plan |
| B. | INSERT example with Syntax 1 | R. | Analyzing the Tables Structures
for a 3-Table Join |
| C. | INSERT Syntax # 2 | S. | An EXPLAIN For a 3-Table Join |
| D. | INSERT example with Syntax 2 | T. | Explain of a Derived Table vs. a
Correlated Subquery |
| E. | INSERT example with Syntax 3 | U. | Explain of the Correlated
Subquery |
| F. | INSERT/SELECT Command | V. | Explain of the Derived Table |
| G. | INSERT/SELECT example using
All Columns (*) | XX. Statistical Aggregate Functions | |
| H. | INSERT/SELECT example with
Less Columns | A. | The Stats Table |
| I. | Two UPDATE Examples | B. | Numeric Manipulation Functions |
| J. | Subquery UPDATE Command
Syntax | C. | Ceiling Gets the Smallest Integer
Not Smaller Than X |
| K. | Example of Subquery UPDATE
Command | D. | Floor Finds the Largest Integer
Not Greater Than X |
| L. | Join UPDATE Command Syntax | E. | The Round Function and
Precision |
| M. | Example of an UPDATE Join
Command | F. | The STDDEV_POP Function |
| N. | DELETE and TRUNCATE
Examples | G. | A STDDEV_POP Example |
| O. | To DELETE or to TRUNCATE | H. | The STDDEV_SAMP Function |
| P. | Subquery and Join DELETE
Command Syntax | I. | A STDDEV_SAMP Example
The VAR_POP Function |
| Q. | Example of Subquery DELETE
Command | | |
| XIX. Kognitio Explain | | | |
| A. | How to See an EXPLAIN Plan | | |
| B. | Seeing an EXPLAIN Plan with
Nexus | | |

Kognitio Architecture and SQL

Course Outline (cont'd)

J.	A VAR_POP Example	Z.	A REGR_SLOPE Example
K.	The VAR_SAMP Function	AA.	Another REGR_SLOPE Example so you can compare
L.	A VAR_SAMP Example	BB.	The REGR_AVGX Function
M.	The CORR Function	CC.	A REGR_AVGX Example
N.	A CORR Example	DD.	Another REGR_AVGX Example so you can compare
O.	Another CORR Example so you can compare	EE.	The REGR_AVGY Function
P.	The COVAR_POP Function	FF.	A REGR_AVGY Example
Q.	A COVAR_POP Example	GG.	Another COVAR_POP Example so you can compare
R.	Another COVAR_POP Example so you can compare	HH.	The REGR_COUNT Function
S.	The COVAR_SAMP Function	II.	A REGR_COUNT Example
T.	A COVAR_SAMP Example	JJ.	The REGR_R2 Function
U.	Another COVAR_SAMP Example so you can compare	KK.	A REGR_R2 Example
V.	The REGR_INTERCEPT Function	LL.	The REGR_SXX Function
W.	A REGR_INTERCEPT Example	MM.	A REGR_SXX Example
X.	Another REGR_INTERCEPT Example so you can compare	NN.	The REGR_SXY Function
Y.	The REGR_SLOPE Function	OO.	A REGR_SXY Example
		PP.	The REGR_SYY Function
		QQ.	A REGR_SYY Example
		RR.	Using GROUP BY