

## DB2 SQL

### Course Summary

#### Description

In this course, students will learn the DB2 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 DB2 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 DB2 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

## DB2 SQL

### Course Outline

- I. **The Basics of SQL**
  - A. Introduction
  - B. Finding Your Current Schema
  - C. Setting Your Default SCHEMA
  - D. SELECT \* (All Columns) in a Table
  - E. SELECT Specific Columns in a Table
  - F. Commas in the Front or Back?
  - G. Place your Commas in front for better Debugging Capabilities
  - 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
  - K. Two Examples of ORDER BY using Different Techniques
  - L. Changing the ORDER BY to Descending Order
  - M. NULL Values sort First in Ascending Mode (Default)
  - N. NULL Values sort First in Descending Mode (DESC)
  - O. Major Sort vs. Minor Sorts
  - P. Multiple Sort Keys using Names vs. Numbers
  - Q. Sorts are Alphabetical, NOT Logical
  - R. Using A CASE Statement to Sort Logically
  - S. How to ALIAS a Column Name
  - T. A Missing Comma can by Mistake become an Alias
  - U. Comments using Double Dashes are Single Line Comments
  - V. Comments for Multi-Lines
  - W. Comments for Multi-Lines as Double Dashes per Line
  - X. Formatting Number Examples
  - Y. Formatting Date Example
  - C. Character Data needs Single Quotes in the WHERE Clause
  - D. Character Data needs Single Quotes, but Numbers Don't
  - E. Comparisons against a Null Value
  - F. NULL means UNKNOWN DATA so Equal (=) won't Work
  - G. Use IS NULL or IS NOT NULL when dealing with NULLs
  - H. NULL is UNKNOWN DATA so NOT Equal won't Work
  - I. Use IS NULL or IS NOT NULL when dealing with NULLs
  - J. Using Greater Than or Equal To (>=)
  - K. AND in the WHERE Clause
  - L. Troubleshooting AND
  - M. OR in the WHERE Clause
  - N. Troubleshooting Or
  - O. Troubleshooting Character Data
  - P. Using Different Columns in an AND Statement
  - Q. Quiz – How many rows will return?
  - R. Answer to Quiz – How many rows will return?
  - S. What is the Order of Precedence?
  - T. Using Parentheses to change the Order of Precedence
  - U. Using an IN List in place of OR
  - V. The IN List is an Excellent Technique
  - W. IN List vs. OR brings the same Results
  - X. The IN List Can Use Character Data
  - Y. Using a NOT IN List
  - Z. Null Values in a NOT IN List Bring Back No Rows
  - AA. A Technique for Handling Nulls with a NOT IN List
  - BB. BETWEEN is Inclusive
  - CC. NOT BETWEEN is Also Inclusive
  - DD. LIKE uses Wildcards Percent '%' and Underscore '\_'
  - EE. LIKE command Underscore is Wildcard for one Character
  - FF. LIKE Command Works Differently on Char Vs Varchar
- II. **The WHERE Clause**
  - A. The WHERE Clause limits Returning Rows
  - B. Double Quoted Aliases are for Reserved Words and Spaces

**DB2 SQL****Course Outline (cont'd)**

- GG. LIKE Command on Character Data Auto Trims
  - HH. Quiz – What Data is Left Justified and what is Right?
  - II. Numbers are Right Justified and Character Data is Left
  - JJ. Answer – What Data is Left Justified and what is Right?
  - KK. An Example of Data with Left and Right Justification
  - LL. A Visual of CHARACTER Data vs. VARCHAR Data
  - MM. Use the TRIM command to remove spaces on CHAR Data
  - NN. Escape Character in the LIKE Command changes Wildcards
  - OO. Escape Characters Turn off Wildcards in the LIKE Command
  - PP. Quiz – Turn off that Wildcard
  - QQ. ANSWER – To Find that Wildcard
  - RR. The Distinct Command
  - SS. Distinct vs. GROUP BY
  - TT. Quiz – How many rows come back from the Distinct?
  - UU. Answer – How many rows come back from the Distinct?
  - VV. The FETCH Clause
  - WW. The FETCH Clause with an ORDER BY Clause
- III. Aggregation**
- A. Quiz – You calculate the Answer Set in your own Mind
  - B. Answer – You calculate the Answer Set in your own Mind
  - C. Quiz – You calculate the Answer Set in your own Mind
  - D. Answer – You calculate the Answer Set in your own Mind
  - E. The 3 Rules of Aggregation
  - F. There are Five Aggregates
  - G. Quiz – How many rows come back?
  - H. Answer – How many rows come back?
  - I. Troubleshooting Aggregates
  - J. GROUP BY delivers one row per Group
  - K. GROUP BY Dept\_No Works  
GROUP BY 1 Fails
  - L. Limiting Rows and Improving Performance with WHERE
  - M. WHERE Clause in Aggregation limits unneeded Calculations
  - N. Keyword HAVING tests Aggregates after they are totaled
  - O. Keyword HAVING is like an Extra WHERE Clause for Totals
  - P. Keyword HAVING tests Aggregates after they are totaled
  - Q. Getting the Average Values per Column
  - R. Average Values per Column for all Columns in a Table
  - S. GROUP BY GROUPING SETS Command
  - T. GROUP BY Grouping Sets
  - U. GROUP BY ROLLUP Command
  - V. GROUP BY Rollup Result Set
  - W. GROUP BY CUBE Command
  - X. GROUP BY CUBE Result Set
  - Y. Quiz - GROUP BY GROUPING SETS Challenge
  - Z. Answer To Quiz - GROUP BY GROUPING SETS Challenge
- IV. Join Functions**
- A. A Two-Table Join Using Traditional Syntax
  - B. A two-table join using Non-ANSI Syntax with Table Alias
  - C. You Can Fully Qualify All Columns
  - D. A two-table join using ANSI Syntax
  - E. Both Queries have the same Results and Performance
  - F. Quiz – Can You Finish the Join Syntax?
  - G. Answer to Quiz – Can You Finish the Join Syntax?
  - H. Quiz – Can You Find the Error?
  - I. Answer to Quiz – Can You Find the Error?
  - J. Super Quiz – Can You Find the Difficult Error?

**DB2 SQL****Course Outline (cont'd)**

- |     |   |           |  |
|-----|---|-----------|--|
| K.  | Answer to Super Quiz – Can You Find the Difficult Error?    | NN.       | Quiz – Will both queries bring back the same Answer Set?   |
| L.  | Quiz – Which rows from both tables won't return?            | OO.       | Answer – Will both queries bring back the same Answer Set? |
| M.  | Answer to Quiz – Which rows from both tables Won't Return?  | PP.       | Quiz – Will both queries bring back the same Answer Set?   |
| N.  | LEFT OUTER JOIN   | QQ.       | Answer – Will both queries bring back the same Answer Set? |
| O.  | LEFT OUTER JOIN Results                                     | RR.       | How would you join these two tables?                       |
| P.  | RIGHT OUTER JOIN  | SS.       | An Associative Table is a Bridge that Joins Two Tables     |
| Q.  | RIGHT OUTER JOIN Example and Results                        | TT.       | Quiz – Can you write the 3-Table Join?                     |
| R.  | FULL OUTER JOIN   | UU.       | Answer to quiz – Can you write the 3-Table Join?           |
| S.  | FULL OUTER JOIN Results                                     | VV.       | Quiz – Can you write the 3-Table Join to ANSI Syntax?      |
| T.  | Which Tables are the Left and which Tables are Right?       | WW.       | Answer – Can you write the 3-Table Join to ANSI Syntax?    |
| U.  | Answer - Which Tables are the Left and which are the Right? | XX.       | Quiz – Can you Place the ON Clauses at the End?            |
| V.  | INNER JOIN with Additional AND Clause                       | YY.       | Answer – Can you Place the ON Clauses at the End?          |
| W.  | ANSI INNER JOIN with Additional AND Clause                  | ZZ.       | The 5-Table Join – Logical Insurance Model                 |
| X.  | ANSI INNER JOIN with Additional WHERE Clause                | AAA.      | Quiz - Write a Five Table Join Using ANSI Syntax           |
| Y.  | OUTER JOIN with Additional WHERE Clause                     | BBB.      | Answer - Write a Five Table Join Using ANSI Syntax         |
| Z.  | OUTER JOIN with Additional AND Clause                       | CCC.      | Quiz - Write a Five Table Join Using Non-ANSI Syntax       |
| AA. | OUTER JOIN with Additional AND Clause Results               | DDD.      | Answer - Write a Five Table Join Using Non-ANSI Syntax     |
| BB. | Quiz – Why is this considered an INNER JOIN?                | EEE.      | Quiz –Re-Write this putting the ON clauses at the END      |
| CC. | Evaluation Order for Outer Queries                          | FFF.      | Answer –Re-Write this putting the ON clauses at the END    |
| DD. | The DREADED Product Join                                    | GGG.      | The Nexus Query Chameleon Writes the SQL for Users         |
| EE. | The DREADED Product Join Results                            |           |  |
| FF. | The Horrifying Cartesian product join                       |           |  |
| GG. | The ANSI Cartesian Join will ERROR                          |           |  |
| HH. | Quiz – Do these Joins Return the Same Answer Set?           |           |  |
| II. | Answer – Do these Joins Return the Same Answer Set?         | <b>V.</b> | <b>Using Nexus for DB2</b>                                 |
| JJ. | The CROSS JOIN  | A.        | Nexus is Available on the Cloud                            |
| KK. | The CROSS JOIN Answer Set                                   | B.        | Nexus Queries Every Major System                           |
| LL. | The Self Join   | C.        | How to Use Nexus   |
| MM. | The Self Join with ANSI Syntax                              | D.        | Why is Nexus Special? Visualization and Automatic SQL      |

**DB2 SQL****Course Outline (cont'd)**

- |     |   |      |   |
|-----|---|------|---|
| E.  | Why is Nexus Special? Cross-System Joins                    | FF.  | The Tabs of the Super Join Builder Philosophy – One Query |
| F.  | Why is Nexus Special? The Amazing Hub System                | GG.  | The Tabs of the Super Join Builder – Objects Tab          |
| G.  | Why is Nexus Special? Save Answer Sets as Tables            | HH.  | The Tabs of the Super Join Builder – Columns Tab)         |
| H.  | Why is Nexus Special? Automated Data Movement               | II.  | The Tabs of the Super Join Builder – Sorting Tab          |
| I.  | Why is Nexus Special? Nexus makes the Servers Talk Directly | JJ.  | The Tabs of the Super Join Builder – Joins Tab            |
| J.  | What Makes Nexus Special? The Garden of Analysis            | KK.  | The Tabs of the Super Join Builder – SQL Tab              |
| K.  | The Garden of Analysis Grouping Sets Tab                    | LL.  | The Tabs of the Super Join Builder – Metadata Tab         |
| L.  | The Garden of Analysis - Grouping Sets Answer Sets          | MM.  | The Tabs of the Super Join Builder – Analytics Tab        |
| M.  | The Garden of Analysis – Join Tab                           | NN.  | The Tabs of the SJB – Analytics Tab – OLAP Screen         |
| N.  | The Garden of Analysis – Charts/Graphs Tab                  | OO.  | Getting a Simple CSUM in the Analytics Tab – OLAP         |
| O.  | The Garden of Analysis – Dynamic Charts Tab                 | PP.  | Getting a Simple CSUM – The SQL Automatically Generated   |
| P.  | The Garden of Analysis – Dashboard Tab                      | QQ.  | The Answer Set of the CSUM                                |
| Q.  | Getting to the Super Join Builder                           | RR.  | Getting all of the OLAP functions in the Analytics Tab    |
| R.  | The Super Join Builder is the First Entry in the Menu       | SS.  | A Five Table Join Using the Menu                          |
| S.  | The Super Join Builder Shows Tables Visually                | TT.  | The First Table is placed in the Super Join Builder       |
| T.  | Using the Add Join Button                                   | UU.  | Using the Add Join Cascading Menu                         |
| U.  | What to Do When No Tables are Joinable?                     | VV.  | All Five Tables Are In the Super Join Builder             |
| V.  | Drag a Joinable Object into the Super Join Builder          | WW.  | A Five Table Join Two Steps (Cube)                        |
| W.  | You will see the Add Custom Join Window                     | XX.  | Choose Cube with Columns from the Left Top of the Table   |
| X.  | Defining the Join Columns                                   | YY.  | All Tables are Cubed (Joined Together Instantly)          |
| Y.  | Your Tables Will Appear Together                            | ZZ.  | Choose Cube and then Choose Your Columns                  |
| Z.  | Select the Columns You Want on the Report                   | AAA. | Create Cube - Tables Are Joined Without Columns Selected  |
| AA. | Check out the SQL Tab to See the SQL that has been built    | BBB. | Create Cube – Select the Columns You Want on the Report   |
| BB. | SQL Tab   | CCC. | How to join DB2, Oracle and SQL Server Tables             |
| CC. | Hit Execute to get the Report inside the Super Join Builder | DDD. | The DB2 Table is now in the Super Join Builder            |
| DD. | The Report is delivered inside the Super Join Builder       |      |   |
| EE. | Let's Join Two Tables Again                                 |      |   |

## DB2 SQL

### Course Outline (cont'd)

- EEE. Drag the Joining Oracle Table to the Super Join Builder
  - FFF. Defining the Join Columns
  - GGG. Choose the Columns You Want on Your Report
  - HHH. Let's Add a SQL Server Table to our DB2 and Oracle Join
  - III. Defining the Join Columns
  - JJJ. All Three Tables are now in the Super Join Builder
  - KKK. Change the Hub and Run the Join on Oracle
  - LLL. Change the Hub and Run the Join on SQL Server
  - MMM. Simply Amazing - Change the Hub to the Garden of Analysis
  - NNN. Have the Answer Set Saved Automatically to Any System
  - OOO. Saving the Answer Set to an Oracle or SQL Server System
  - PPP. Saving the Answer Set to a DB2 System
  - QQQ. Saving the Answer Set to a Teradata System
- VI. Date Functions**
- A. Getting the System Date
  - B. Extracting From a Timestamp
  - C. The EXTRACT Command
  - D. Using the EXTRACT Command to Extract Month, Day, Year
  - E. Extracting From a Date Column
  - F. Extracting the Date and Time from the Timestamp
  - G. Formatting Dates Example
  - H. Formatting Date Standards
  - I. Adding and Subtracting Days from a Date
  - J. Adding Years, Months, Days, Hours and Seconds
  - K. Using the Add\_Months Command
  - L. Adding Years to a Date
  - M. Add Five Years to a Date
  - N. Converting Character Data to a Date or Time
  - O. Timestamp DAYOFWEEK, DAYNAME and MONTHNAME
  - P. Finding Orders That Happened on a Friday
  - Q. NEXT\_DAY Command Finds a Future Day of the Week
  - R. Finding the Last Day of a Month
  - S. Finding the Last Day of the Previous Month
  - T. Getting the First Day of the Month
  - U. Finding the Number of Days between Two Dates
  - V. Resetting the Microseconds Back to Zero
  - W. Turning Date and Time into Characters
  - X. Converting Character Data to a Timestamp
  - Y. Finding Differences between Timestamps
  - Z. Differences between Timestamps Fractions of a Second
  - AA. Find Differences between Timestamp Seconds and Minutes
  - BB. Find Differences between Timestamp Hours and Days
  - CC. Find Differences between Timestamp Weeks and Months
  - DD. Find Differences between Timestamp Quarters and Years
  - EE. Formatting Dates
  - FF. Formatting Dates Example
  - GG. Formatting Timestamp Example
- VII. OLAP Functions**
- A. The Row\_Number Command
  - B. Quiz – How did the Row\_Number Reset?
  - C. Using a Derived Table and Row\_Number
  - D. Ordered Analytics OVER
  - E. RANK and DENSE RANK
  - F. RANK Defaults to Ascending Order
  - G. Getting RANK to Sort in DESC Order
  - H. RANK OVER and PARTITION BY
  - I. Finding Gaps between Dates
  - J. CSUM – Rows Unbounded Preceding Explained

**DB2 SQL****Course Outline (cont'd)**

- |     |  |      |  |
|-----|--|------|--|
| K.  | CSUM – Making Sense of the Data                        | KK.  | MAX OVER with PARTITION BY Reset                     |
| L.  | CSUM – Making Even More Sense of the Data              | LL.  | MAX OVER without Rows Unbounded Preceding            |
| M.  | CSUM – The Major and Minor Sort Key(s)                 | MM.  | The MIN OVER Command                                 |
| N.  | The ANSI CSUM – Getting a Sequential Number            | NN.  | MIN OVER without Rows Unbounded Preceding            |
| O.  | Reset with a PARTITION BY Statement                    | OO.  | MIN OVER Using PARTITION BY to Reset                 |
| P.  | PARTITION BY only Resets a Single OLAP not ALL of them | PP.  | Finding a Value of a Column in the Next Row with MIN |
| Q.  | CURRENT ROW AND UNBOUNDED FOLLOWING                    | QQ.  | The CSUM for Each Product_Id and the Next Start Date |
| R.  | Different Windowing Options                            | RR.  | Quiz – Fill in the Blank                             |
| S.  | Moving Sum has a Moving Window                         | SS.  | Answer – Fill in the Blank                           |
| T.  | How ANSI Moving SUM Handles the Sort                   | TT.  | Using FIRST_VALUE                                    |
| U.  | Quiz – How is that Total Calculated?                   | UU.  | FIRST_VALUE  |
| V.  | Answer to Quiz – How is that Total Calculated?         | VV.  | FIRST_VALUE after Sorting by the Highest Value       |
| W.  | Moving SUM every 3-rows Vs a Continuous Average        | WW.  | FIRST_VALUE with Partitioning                        |
| X.  | PARTITION BY Resets an ANSI OLAP                       | XX.  | FIRST_VALUE Combined with Row_Number                 |
| Y.  | The Moving Window is Current Row and Preceding         | YY.  | FIRST_VALUE and Row_Number with Different Sort       |
| Z.  | Moving Average   | ZZ.  | Using LAST_VALUE                                     |
| AA. | Moving Average Using a CAST Statement                  | AAA. | LAST_VALUE   |
| BB. | Moving Average every 3-rows Vs a Continuous Average    | BBB. | Using LAG and LEAD                                   |
| CC. | PARTITION BY Resets an ANSI OLAP                       | CCC. | LEAD   |
| DD. | Moving Difference                                      | DDD. | LEAD   |
| EE. | Moving Difference using ANSI Syntax with Partition By  | EEE. | LEAD With Partitioning                               |
| FF. | COUNT OVER for a Sequential Number                     | FFF. | LEAD to Find the First Occurrence                    |
| GG. | COUNT OVER without Rows Unbounded Preceding            | GGG. | Using LEAD   |
| HH. | Quiz – What caused the COUNT OVER to Reset?            | HHH. | Using LEAD with an Offset of 2                       |
| II. | Answer to Quiz – What caused the COUNT OVER to Reset?  | III. | Using LAG  |
| JJ. | The MAX OVER Command                                   | JJJ. | Using LAG with an Offset of 2                        |
|     |  | KKK. | LAG  |
|     |  | LLL. | LAG with Partitioning                                |
|     |  | MMM. | SUM (SUM (n))  |
- VIII. Temporary Tables**
- |    |  |
|----|--|
| A. | There are two types of Temporary Tables              |
| B. | CREATING A Derived Table                             |
| C. | Creating Multiple Derived Tables in the WITH Command |
| D. | Creating Multiple Derived Tables in the WITH Command |

## DB2 SQL

### Course Outline (cont'd)

- E. The Same Derived Query shown Three Different Ways
  - F. Most Derived Tables Are Used To Join To Other Tables
  - G. The Three Components of a Derived Table
  - H. Visualize This Derived Table
  - I. Our Join Example with A Different Column Aliasing Style
  - J. Column Aliasing Can Default For Normal Columns
  - K. Our Join Example With the WITH Syntax
  - L. Quiz - Answer the Questions
  - M. Answer to Quiz - Answer the Questions
  - N. Clever Tricks on Aliasing Columns in a Derived Table
  - O. An Example of Two Derived Tables in a Single Query
  - P. Example of Two Derived Tables in a Single WITH Statement
  - Q. WITH RECURSIVE Derived Table Hierarchy
  - R. WITH RECURSIVE Derived Table Query
  - S. WITH RECURSIVE Derived Table Definition
  - T. WITH RECURSIVE Derived Table Seeding
  - U. WITH RECURSIVE Derived Table Looping
  - V. WITH RECURSIVE Derived Table Looping in Slow Motion
  - W. WITH RECURSIVE Derived Table Looping Continued
  - X. WITH RECURSIVE Derived Table Ends the Looping
  - Y. WITH RECURSIVE Derived Table Definition
  - Z. WITH RECURSIVE Final Answer Set
  - AA. Creating and Populating a Global Temporary Table
  - BB. Global Temporary Table Definitions Persist
  - CC. ON COMMIT DELETE ROWS Example
  - DD. Creating and Populating a Global Temporary Table
  - EE. Creating a Global Temporary Table Using a CTAS
  - FF. Creating a Global Temporary Table Using a CTAS Join
  - GG. A Global Temp Table That Populates Some of the Rows
  - HH. A Temporary Table with Some of the Columns
- IX. Sub-query Functions**
- A. An IN List is much like a Subquery
  - B. An IN List Never has Duplicates – Just like a Subquery
  - C. The Subquery
  - D. The Three Steps of How a Basic Subquery Works
  - E. These are Equivalent Queries
  - F. The Final Answer Set from the Subquery
  - G. Quiz- Answer the Difficult Question
  - H. Answer to Quiz- Answer the Difficult Question
  - I. Should you use a Subquery or a Join?
  - J. Quiz- Write the Subquery
  - K. Answer to Quiz- Write the Subquery
  - L. Quiz- Write the More Difficult Subquery
  - M. Answer to Quiz- Write the More Difficult Subquery
  - N. Quiz – Write the Extreme Subquery
  - O. Answer to Quiz- Write the Extreme Subquery
  - P. Quiz- Write the Subquery with an Aggregate
  - Q. Answer to Quiz- Write the Subquery with an Aggregate
  - R. Quiz- Write the Correlated Subquery
  - S. Answer to Quiz- Write the Correlated Subquery



**DB2 SQL****Course Outline (cont'd)**

- |     |   |            |   |
|-----|---|------------|---|
| T.  | The Basics of a Correlated Subquery                         | TT.        | How a Correlated Exists matches up                      |
| U.  | The Top Query always runs first in a Correlated Subquery    | UU.        | The Correlated NOT Exists                               |
| V.  | Correlated Subquery Example vs. a Join with a Derived Table | <b>X.</b>  | <b>Strings</b>  |
| W.  | Quiz- A Second Chance to Write a Correlated Subquery        | A.         | The LENGTH Command Counts Characters                    |
| X.  | Answer - A Second Chance to Write a Correlated Subquery     | B.         | The LENGTH Command – Spaces can Count too               |
| Y.  | Quiz- A Third Chance to Write a Correlated Subquery         | C.         | The LENGTH Command and Character Data                   |
| Z.  | Answer - A Third Chance to Write a Correlated Subquery      | D.         | The LENGTH Needs a TRIM                                 |
| AA. | Quiz- Last Chance to Write a Correlated Subquery            | E.         | The TRIM Command trims both Leading and Trailing Spaces |
| BB. | Answer – Last Chance to Write a Correlated Subquery         | F.         | A Visual of the TRIM Command Using Concatenation        |
| CC. | Quiz – Write the Extreme Correlated Subquery                | G.         | Trim and Trailing is Case Sensitive                     |
| DD. | Answer To Quiz – Write the Extreme Correlated Subquery      | H.         | How to TRIM Trailing Letters                            |
| EE. | Quiz- Write the NOT Subquery                                | I.         | The SUBSTRING Command                                   |
| FF. | Answer to Quiz- Write the NOT Subquery                      | J.         | How SUBSTRING Works with NO ENDING POSITION             |
| GG. | Quiz- Write the Subquery using a WHERE Clause               | K.         | An Example using SUBSTRING, TRIM and CHAR Together      |
| HH. | Answer - Write the Subquery using a WHERE Clause            | L.         | Concatenation   |
| II. | Quiz- Write the Subquery with Two Parameters                | M.         | Concatenation and SUBSTRING                             |
| JJ. | Answer to Quiz- Write the Subquery with Two Parameters      | N.         | Four Concatenations Together                            |
| KK. | How the Double Parameter Subquery Works                     | O.         | UPPER and LOWER Commands                                |
| LL. | More on how the Double Parameter Subquery Works             | P.         | LPAD and RPAD   |
| MM. | Quiz – Write the Triple Subquery                            | Q.         | SOUNDEX   |
| NN. | Answer to Quiz – Write the Triple Subquery                  | <b>XI.</b> | <b>Interrogating the Data</b>                           |
| OO. | Quiz – How many rows return on a NOT IN with a NULL?        | A.         | Using the LOWER Command                                 |
| PP. | Answer – How many rows return on a NOT IN with a NULL?      | B.         | Using the UPPER Command                                 |
| QQ. | How to handle a NOT IN with potential NULL Values           | C.         | Non-Letters are Unaffected by UPPER and LOWER           |
| RR. | IN is equivalent to =ANY                                    | D.         | Quiz – Fill in the Answers for the NULLIF Command       |
| SS. | Using a Correlated Exists                                   | E.         | The COALESCE Command                                    |
|     |   | F.         | The COALESCE Answer Set                                 |
|     |   | G.         | The COALESCE Command – Fill In the Answers              |
|     |   | H.         | The COALESCE Answer Set                                 |
|     |   | I.         | COALESCE is Equivalent to This CASE Statement           |
|     |   | J.         | The Basics of CAST (Convert and Store)                  |
|     |   | K.         | Some Great CAST (Convert and Store) Examples            |

**DB2 SQL****Course Outline (cont'd)**

- L. A Rounding Example
  - M. Quiz - The Basics of the CASE Statements
  - N. Answer to Quiz - The Basics of the CASE Statements
  - O. Using an ELSE in the Case Statement
  - P. Using an ELSE as a Safety Net
  - Q. Rules for a Valued Case Statement
  - R. Rules for a Searched Case Statement
  - S. Valued Case Vs. A Searched Case
  - T. Quiz - Valued Case Statement
  - U. Answer - Valued Case Statement
  - V. Quiz - Searched Case Statement
  - W. Answer - Searched Case Statement
  - X. The CASE Challenge
  - Y. The CASE Challenge Answer
  - Z. Combining Searched Case and Valued Case
  - AA. A Trick for getting a Horizontal Case
  - BB. Nested Case
  - CC. Put a CASE in the ORDER BY
- XII. View Functions**
- A. The Fundamentals of Views
  - B. Creating a Simple View to Restrict Sensitive Columns
  - C. You SELECT From a View
  - D. Creating a Simple View to Restrict Rows
  - E. A View Provides Security for Columns and Rows
  - F. Basic Rules for Views
  - G. How to Modify a View
  - H. An Exception to the ORDER BY Rule inside a View
  - I. Views Are Sometimes CREATED for Formatting
  - J. Creating a View to Join Tables Together
  - K. How to Alias Columns in a View CREATE
  - L. The Standard Way Most Aliasing is done
- XIII. Set Operators Functions**
- A. Rules of Set Operators
  - B. INTERSECT Explained Logically
  - C. UNION Explained Logically
  - D. UNION ALL Explained Logically
  - E. EXCEPT Explained Logically
  - F. Minus 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. A Great Example of how MINUS works
  - N. USING Multiple SET Operators in a Single Request
  - O. Changing the Order of Precedence with Parentheses
  - P. Using UNION ALL for speed in Merging Data Sets
- XIV. Table Create and Data Types**
- A. The Basics of Creating a Table
  - B. Creating a Table with NOT NULL Constraints
  - C. Creating a Table with a UNIQUE Constraint
  - D. Creating a Unique Index
  - E. Creating a Clustered Index
- M. What Happens When Both Aliasing Options Are Present
  - N. Resolving Aliasing Problems in a View CREATE
  - O. Answer to Resolving Aliasing Problems in a View CREATE
  - P. Creating a View with a Local Check
  - Q. Aggregates on View Aggregates
  - R. Altering a Table after a View Has Been Created
  - S. A View that Errors after an ALTER

**DB2 SQL****Course Outline (cont'd)**

- F. Creating a Partitioned Table for a Range on a Date Column
  - G. Creating a Partitioned Table for a Number
  - H. Declaring a Global Temporary Table
  - I. Declaring a Global Temporary Table that Persists Using Like
  - J. Creating a Table with an XML Data Type
  - K. Creating a Table with a CHECK Constraint
  - L. Creating a Table with Default Values
  - M. Creating a Table with Multiple Constraints
  - N. Defining Primary Keys
  - O. Defining a Primary Key after the Table Has Been Created
  - P. Defining a Foreign Key after the Table Has Been Created
  - Q. Creating a Table with an Identity Column that is Unique
  - R. Creating a Table with an Identity Column that is Non-Unique
  - S. Creating a Sequence
  - T. Altering a Table to Add a Column
  - U. Altering a Table to Drop a Column
  - V. Renaming a Table
  - W. Dropping a Table
  - X. Creating a Table Using a CTAS or a LIKE
  - Y. Creating a Table Using a CTAS Join
  - Z. Data Types
- XV. Data Manipulation Language (DML)**
- A. INSERT Syntax # 1
  - B. INSERT example with Syntax 1
  - C. INSERT Syntax # 2
  - D. INSERT example with Syntax 2
  - E. INSERT/SELECT Command
  - F. INSERT/SELECT example using All Columns (\*)
  - G. INSERT/SELECT example with Less Columns
  - H. Two UPDATE Examples
  - I. Subquery UPDATE Command Syntax
  - J. Example of Subquery UPDATE Command
  - K. The DELETE Command Basic Syntax
  - L. Example of Subquery DELETE Command
  - M. Example of Subquery DELETE That Gets Rid of Null Values
- XVI. Statistical Aggregate Functions**
- A. Numeric Manipulation Functions
  - B. The Stats Table
  - C. The VARIANCE Function
  - D. A VARIANCE Example
  - E. The CORR Function
  - F. A CORR Example
  - G. Another CORR Example so you can compare
  - H. The REGR\_INTERCEPT Function
  - I. A REGR\_INTERCEPT Example
  - J. Another REGR\_INTERCEPT Example so you can compare
  - K. The REGR\_SLOPE Function
  - L. A REGR\_SLOPE Example
  - M. Another REGR\_SLOPE Example so you can compare
  - N. The REGR\_AVGX Function
  - O. A REGR\_AVGX Example
  - P. Another REGR\_AVGX Example so you can compare
  - Q. The REGR\_AVGY Function
  - R. A REGR\_AVGY Example
  - S. Another REGR\_AVGY Example so you can compare
  - T. The REGR\_COUNT Function
  - U. A REGR\_COUNT Example
  - V. The REGR\_R2 Function
  - W. A REGR\_R2 Example
  - X. The REGR\_SXX Function
  - Y. A REGR\_SXX Example
  - Z. The REGR\_SXY Function
  - AA. A REGR\_SXY Example
  - BB. The REGR\_SYY Function
  - CC. A REGR\_SYY Example