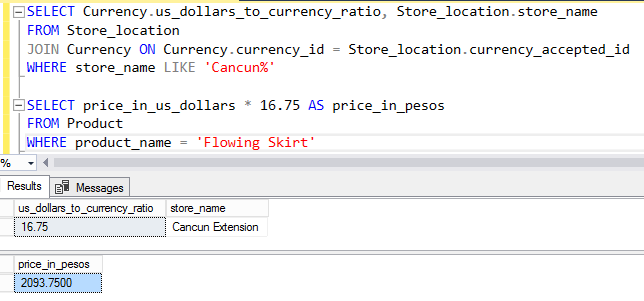
This submission template is a convenient document for you to provide the screenshots and explanations for Lab 5. This submission template is intended to be used in conjunction with the Lab 5 Instructions document. The instructions document illustrates how to correctly execute each SQL construct, explains important theoretical and practical details, and contains the complete set of instructions on how to complete this lab.

**Name**: Scott Kaeneman

**Date: 6/11/2018**

**Section One – Expressions and Value Manipulation**

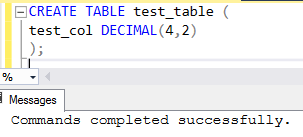
3. Two independent queries



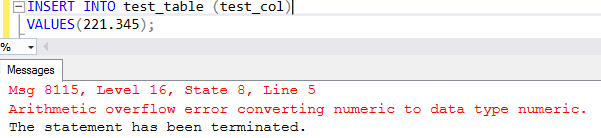
5. Datatype size limitation

a. Explanation: The “Decimal” data type in MS SQL Server “Allows numbers from -10^38 +1 to 10^38 –1” (<https://www.w3schools.com/sql/sql_datatypes.asp>). the decimal data type has both “precision” and “scale.” Precision is the total number of digits that can be used, whereas scale refers to the digits to the right of the decimal point. For instance ‘myDecimal DECIMAL(4, 2)’ would create a column called myDecimal that allowed for 4 digits total, with up to 2 of those being at the right of the decimal point.

b. Table creation



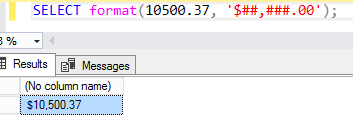
c. Insertion attempt



7. Expression and operator precedence explanation: An expression is essentially comprised of operands such as the numbers 1 or 2 and operators that allow for some type of data manipulation such as multiplication or division. The order of operations is important and needs to be followed or else an incorrect answer can be derived. For instance, in the example 10 - 8 / 4 the division must first be completed and then the subtraction.

9. Datatype precedence explanation: Data type precedence means that there is a certain order when it comes to data types and that those with a lower precedence will be converted into the higher data type precedence when there is some form of operation conducted on an expression that contains two or more data types that have different precedence. For example, a decimal and a float value multiplied together would yield a float value for the answer.

11. Currency formatting



13. SQL clients explanation

Different SQL clients display the same results differently due to the fact that each client was coded by different development teams at separate companies, or in the case of PostgreSQL by its open source developers. Each team had to decide how best to handle and present the data stored in the database so there will be slight variations on how this is accomplished across different SQL clients.

**Section Two – Subqueries**

15. Subquery explanation

a. Subquery superiority explanation

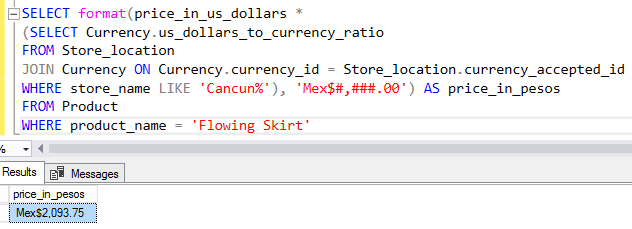
A subquery is more efficient than 2 separate queries due to the fact that it can be run dynamically so the value in one table can change without having to manually update the second query.

b. Uncorrelated subquery explanation

A subquery is uncorrelated which means that it can be run independently of the outer query. A good test to see if the subquery is executing properly is to run it without running the outer query. The SQL Engine will execute the subquery separately from the outer query. The subquery results are then available to the outer query.

16. Two queries combination

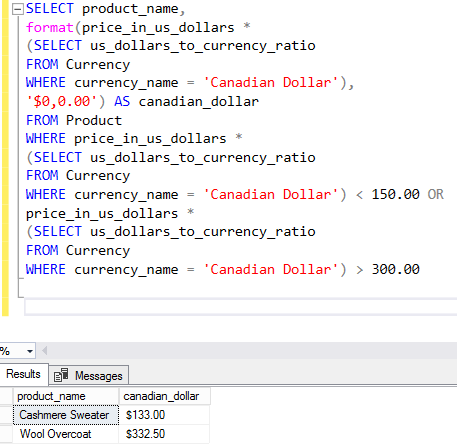
a. Screenshot



b. Explanation

A subquery is used to join the Currency table and the Store\_location table on the currencies. The subquery is then multiplied by the price\_in\_us\_dollars and formatted with the Mexican peso currency. This approach is more streamlined and dynamic instead of having 2 independent queries.

18. More and less expensive product use case



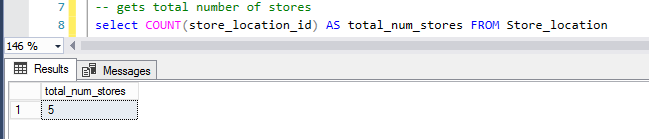
20. More complex use case

a. Part identification

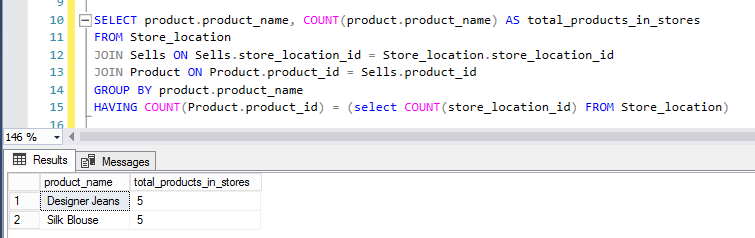
Marcus wants a product only if it is available in all store locations, so check each store to ensure it has the product. this will require looking at the Product and Store\_location tables. The sizing options of each product also need to be shown.

b. Independent queries

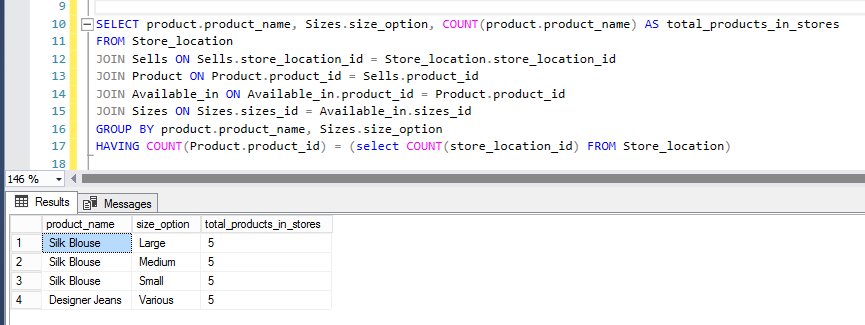
total number of stores



where to find the product that is sold in all locations



c. Full query



d. Explanation

I first got a count of how many stores there were, then made sure that a clothing item was in each of those stores, then got the sizes.

22. FROM clause solution

a. Screenshot

b. Explanation

24. Correlated vs. uncorrelated explanation

A correlated subquery references at a minimum one table in the outer query whereas an uncorrelated subquery is completely independent of the outer query and can be run on its own.

25. EXISTS solution

a. Screenshot

b. Explanation of changes

c. Explanation of better choice



Your lab submission will be evaluated according to the following rubric.

|  |  |  |
| --- | --- | --- |
|  | **Letter Grade** | **Qualities Demonstrated by the Lab Submission** |
| **Correctness, completeness, and constitution**  **Measures the correctness and completeness of the results, and the quality of the constitution of the SQL constructs** | A+ ➔ 100 | The results and explanations are entirely complete and correct for all steps. There are absolutely no technical or other errors present. There is no known way to improve the logic and makeup of any of the SQL constructs. |
| A ➔ 96 | One insignificant technical or other error is present, but otherwise the results and explanations are entirely complete and correct for all steps. Excluding the insignificant error, there is no known way to improve the makeup of any of the SQL constructs. |
| A- ➔ 92 | One or two consequential technical or other errors are present, but otherwise the results and explanations are entirely complete and correct for all steps. Excluding the one or two errors, there is no known way to improve the makeup of any of the SQL constructs. |
| B+ ➔ 88 | A few steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of most SQL constructs are sound. |
| B ➔ 85 | A few steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of most SQL constructs are sound. |
| B- ➔ 82 | About ¼ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least ¾ of the SQL constructs are sound. |
| C+ ➔ 78 | About ¼ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of at least ¾ of the SQL constructs are sound. |
| C ➔ 75 | About half of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least half of the SQL constructs are sound. |
| C- ➔ 72 | About half of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of at least half of the SQL constructs are sound. |
| D ➔ 67 | About ¾ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least ¼ of the SQL constructs are sound |
| F ➔ 0 | All or almost all of the steps have incomplete or incorrect results or explanations. The logic and makeup of all or almost all of the SQL constructs are unsound. |