Jennifer DeVore

13 March 2020

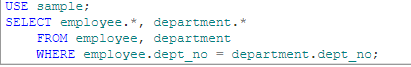
CIS 435L

**Unit 2 Lab Assignment 1: Retrieve data from two or more tables**

**Terms:**

1. **Implicit Syntax vs. Explicit Syntax**

According to W3Computing, “The implicit join syntax is “old-style” syntax, where each join operation is defined implicitly via the WHERE clause, using the so-called join columns (W3Computing, 2020).”



(W3Computing, 2020).

Explicit syntax actually uses the word JOIN.



(W3Computing, 2020).

1. **Table alias**

This is a temporary name for a table, or a column that allows for more readability, but only for the query (W3Schools, 2020).



(W3Schools, 2020).

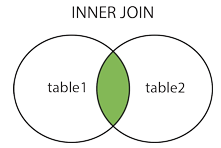


(W3Schools, 2020).

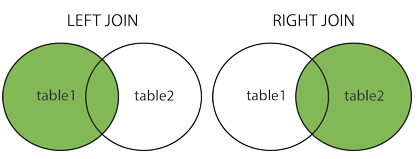
1. **Qualified Column name**

According to IBM, “A qualifier for a column name can be a table name, a view name, an alias name, a synonym, or a correlation name. Whether a column name can be qualified depends, like its meaning, on its context (IBM, 2020).” The cases that allow for a qualified column name are in some COMMENT and LABEL statements, when used for an INSERT statement, in the portions of assignment for UPDATE or MERGE statements.

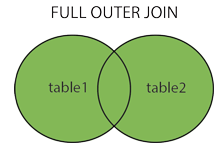
1. **Inner JOIN vs Outer JOIN vs FULL JOIN**



(W3Schools, SQL Joins, 2020). The inner join is the one in the diagram above and shows how the inner join works. The inner join only returns records that are a commonality of both tables being joined.



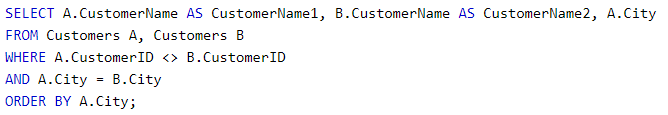
(W3Schools, SQL Joins, 2020). The image above shows how the left outer join and right outer join work. The left returns all of table one (the “left” table) and the commonality records shared from table 2. The right join does the same thing, just flip-flopped.



(W3Schools, SQL Joins, 2020). The full outer join joins both table and one completely – the complete records from one and the complete records from two.

1. **Self-join**

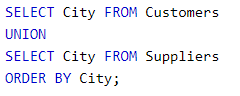
SQL Self-join is doing the same thing as any other join, except it is joining the table with itself.



(W3Schools, SQL Self Join, 2020).

1. **UNION**

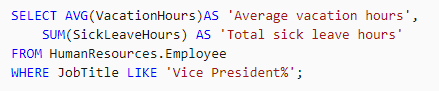
According to W3Schools, “The UNION operator is used to combine the result-set of two or more SELECT statements. Each SELECT statement within UNION must have the same number of columns. The columns must also have similar data types. The columns in each SELECT statement must also be in the same order (W3Schools, SQL UNION Operator, 2020).” Union will not return duplicate values within the tables unless you use UNION ALL.



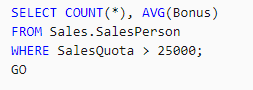
(W3Schools, SQL UNION Operator, 2020).

1. **Aggregate function (with two examples)**

According to Microsoft, “An aggregate function performs a calculation on a set of values and returns a single value. Except for COUNT, aggregate functions ignore null values. Aggregate functions are often used with the GROUP BY clause of the SELECT statement (Microsoft, 2020).”



(Microsoft, 2020). This example shows two aggregate functions – average and sum. This example takes all of the VacactionHours in the HumanResources.Employee table and calculates the average. This also adds up all of the SickLeaveHours in the same table.



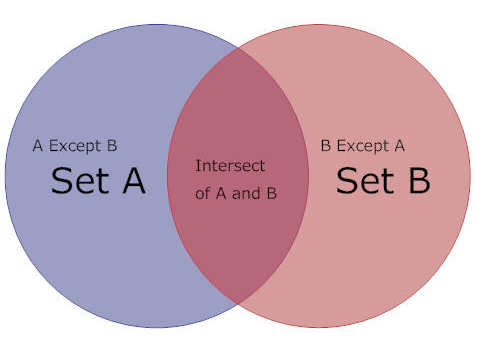
(Microsoft, 2020). This example also shows two aggregate functions – count and average. This example counts each record within the Sales.SalesPerson table where the SalesQuota is greater than 25k and averages the bonuses in the same table.

1. **Summary query**

According to QueryTool, “Summary queries (also called Group-By queries) are used to summarise the contents of a table (QueryTool, 2020).”

1. **INTERSECT vs EXCEPT**

According to Sheldon, “The INTERSECT operator returns all rows that are in both result sets. The EXCEPT operator returns the rows that are only in the first result set but not in the second (Sheldon, 2011).”



(Sheldon, 2011). In the diagram above you can see how Intersect and Except work. Intersect returns the results that are shared in common. If you want to return table A and the parts that are not in common with B, you’d use A EXCEPT B. If you wanted B and the parts that are not in common with A you’d use B Except A.

1. **HAVING**

According to Guru99, “The HAVING clause is used to restrict the results returned by the GROUP BY clause (Guru99, 2020).”

|  |  |
| --- | --- |
| **Exercise 1** | **Determine how many products you can find in the Production.Product table.**  ----------- 504  (1 row(s) affected) |
| **Exercise 2** | **Write a query that retrieves the number of products in the Production.Product table that are included in a subcategory.**  ***Hint:*** the rows that have NULL in column ProductSubcategoryID are considered not a part of any subcategory.  HasSubCategoryID ---------------- 295  (1 row(s) affected)  Notice that the result has a column name. Also take a look at the message you receive from SQL Server. |
| **Exercise 3** | **Determine how many Products reside in each SubCategory.**  The answer to this is retrievable if you write a query that uses the COUNT aggregate function combined with a GROUP BY clause.  The column ProductSubcategoryID is a candidate for building groups of rows when querying the Production.Product table. Your result set should look something like the result below:  Notice the column alias for the second column.  ProductSubcategoryID CountedProducts -------------------- --------------- NULL                 209 1                    32 2                    43   (38 row(s) affected) |
| **Exercise 4** | **Write a query that lists the country and province names stored in AdventureWorks2012 sample database.**  In the Person schema, you will find the CountryRegion and StateProvince tables. Join them and produce a result set similar to the following. Notice that there is no particular sort order in the result set.  Country                        Province ------------------------------ ------------------------ Canada                         Alberta United States                  Alaska United States                  Alabama ………  (181 row(s) affected) |
| **Exercise 5** | **Continue to work with the previous query and add a filter to list only the countries Germany and Canada.**  Also, notice the sort order and column headings of the result set. Your result set should look similar to the following:  Country                        Province ------------------------------ ------------------------ Canada                         Alberta Canada                         British Columbia ……… Germany                        Brandenburg Germany                        Hamburg  (20 row(s) affected |
| **Exercise 6** | **Write a query that retrieves the average commission percent for the SalesPerson table.** |
| **Exercise 7** | **Write a query that retrieves a count of all Male Employees. Name the column.**  Hint: Use a HumanResources table. |
| **Exercise 8** | **Write a query that retrieves the highest List Price and lowest List Price of any Product in the Production.Product table.**  Column names should be Max List Price and Min List Price.  Min List Price cannot be 0.00. |
| **Exercise 9** | **Identify the possible causes of error(s) and rectify.**  Consider the following SQL query:  SELECT ProductID, LineTotal AS 'Total' FROM Sales.SalesOrderDetail Group By Cube(LineTotal)  Once executed, the preceding query generates errors. Identify the possible causes of such error(s) and rectify.    ProductID                     Total ----------------------------  ------------------------ 707                           157772.394392 708                           160869.517836 ……… 718                           395182.699300 …… (267 row(s) affected |

# References

Guru99. (2020, March 9). *MySQL GROUP BY and HAVING Clause with Examples*. Retrieved from Guru99: https://www.guru99.com/group-by.html

IBM. (2020, March 13). *Qualified Column Names*. Retrieved from IBM: https://www.ibm.com/support/knowledgecenter/SSEPEK\_11.0.0/sqlref/src/tpc/db2z\_qualifiedcolumnnames.html

Microsoft. (2020, March 12). *Aggregate Functions (Transact-SQL)*. Retrieved from Microsoft: https://docs.microsoft.com/en-us/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15

QueryTool. (2020, March 13). *Summary or Group-by Queries*. Retrieved from QueryTool: http://www.querytool.com/help/230.htm

Sheldon, R. (2011, February 28). *The EXCEPT and INTERSECT Operators in SQL Server*. Retrieved from Redgate Hub: https://www.red-gate.com/simple-talk/sql/performance/the-except-and-intersect-operators-in-sql-server/

W3Computing. (2020, March 13). *SQL JOIN – JOIN Operator (Explicit join syntax, Implicit join syntax)*. Retrieved from W3Computing: https://www.w3computing.com/sqlserver2012/join-operator-explicit-implicit-syntax/

W3Schools. (2020, March 13). *SQL Aliases*. Retrieved from W3Schools: https://www.w3schools.com/sql/sql\_alias.asp

W3Schools. (2020, March 13). *SQL Joins*. Retrieved from W3Schools: https://www.w3schools.com/sql/sql\_join.asp

W3Schools. (2020, March 13). *SQL Self Join*. Retrieved from W3Schools: https://www.w3schools.com/sql/sql\_join\_self.asp

W3Schools. (2020, March 13). *SQL UNION Operator*. Retrieved from W3Schools: https://www.w3schools.com/sql/sql\_union.asp