Luis Valderrama

April 24, 2021

IT FDN 130 A

Module 03

Selecting and Aggregating Data

# Intro

During the course of the third module, I learned the SQL statements’ main clauses available to query the database as well as the commonly used functions to aggregate data. This document addresses concepts and practical uses of Aggregate Functions, Clauses and the primary difference between WHERE and HAVING clauses, and their practical use as demonstrated through the database created for this assignment, Assignment03DB\_LuisValderrama.

# Aggregate Functions

*“In database management, an aggregate function or aggregation function is a function where the values of multiple rows are grouped together to form a single summary value”,* <https://en.wikipedia.org/wiki/Aggregate_function>*,* (2021) (external site). The commonly used function include:

* **AVG** - calculates the average of an attribute in a defined group
* **COUNT** - counts the number of elements in a defined group
* **MAX** - finds the maximum value in a defined group
* **MIN** - finds the minimum value in a defined group
* **SUM** - calculates the sum of the attribute in a defined group

There are other functions offered by the RDMS MS SQL. [SQL Aggregate Functions in Microsoft.com](https://docs.microsoft.com/en-us/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15%23:~:text=An%20aggregate%20function%20performs%20a,All%20aggregate%20functions%20are%20deterministic.), (2018) (external site). The aggregate functions perform in a way, similar to a pivot table and some common functions found in MS Excel which allows the user to filter the data, and perform calculations as well as other functions. **The data analyst would use** **aggregate functions to** **return results from a table’s attribute(s) requiring arithmetical or statistical computations**. The example below was created using MAX as one of the elements of the script in order to return the maximum number of inventories by a specific category and within a specific date range. (Figure 1.1)

Select Products.ProductName, Max(Inventories.Count) As [MaxAmountInInventory]

From Products

Join Inventories

On Products.ProductID = Inventories.ProductID

Where InventoryDate Between '01/01/2017' And '02/28/2017'

And CategoryID = 8

Group By ProductName

Order By ProductName Asc;

go

***Figure 1.1: Practical application of the Aggregate Function, ‘MAX’.***

# SQL CLAUSES or Statements

*“Clauses are in-built functions available to us in SQL. With the help of clauses, we can deal with data easily stored in the table. Clauses help us filter and analyze data quickly.* ***When we have large amounts of data stored in the database, we use Clauses to query and get data required by the user.”*** [*https://data-flair.training/blogs/clause-in-sql/*](https://data-flair.training/blogs/clause-in-sql/)*.* (2019) (external site). The list below are commonly used clauses that are used in multiple DML statements.

* **SELECT** - used to list one or more columns to be displayed
* **FROM** - used to list one or more tables
* **ORDER BY** - used to sort the results-set in ascending or descending order
* **GROUP BY** - used to groups rows that have the same values into summary rows
* **TOP** - used to specify the number of records to return
* **WHERE** - Defines the condition to be met for the rows to be returned. In other words, it is used as a Boolean filter
* **HAVING** – Specifies one or more predicates for groups and/or aggregates to meet

As noted above WHERE and HAVING are very similar in the fact that both clauses are used to filter records. However, **the primary difference between the two clauses is the fact HAVING** **can be used and is typically used with aggregate functions** **while WHERE cannot be used with aggregate functions.** The example below created for my assignment displays the use of both, WHERE and HAVING. The HAVING statement is being used with the aggregate function MAX and greater or equal to (>=), operator with the objective of returning the maximum number of inventories for a specific category, and restricting the maximum count to 10 or higher. (Figure 2.1)

Select Products.ProductName, Max(Inventories.Count) As [MaxAmountInInventory]

From Products

Join Inventories

On Products.ProductID = Inventories.ProductID

Where InventoryDate Between '01/01/2017' And '02/28/2017'

And CategoryID = 8

Group By ProductName

Having Max(Inventories.Count) >= 10

Order By ProductName Asc;

go

***Figure 2.1: Practical use of ‘HAVING’ clause utilized with the aggregate function ‘MAX’ and operator ‘Greater or Equal to’.***

## Group-By with Aggregate Functions

*“The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause”*. <https://www.tutorialspoint.com/sql/sql-group-by.htm>. (2021) (external site). **The data analyst would need to use GROUP BY combined with an aggregate function to return results requiring arithmetic or statistical calculations grouped by a defined segment(s)** as the GROUP BY statement is often used with aggregate functions such as COUNT (), MAX (), MIN (), SUM (), AVG () to group the result-set by one or more columns. The example presented below displays the use of GROUP BY to summarize the data by CategoryName, specifically, Seafood. The GROUP BY statement is being used with the aggregate function MAX and the operator greater or equal to with the objective of returning the maximum number of inventories for the Seafood category, and restricting the maximum count to 10 or higher. (Figure 2.2)

Select Categories.CategoryName, Products.ProductName, Max(Inventories.Count) As [MaxAmountInInventory]

From Products

Join Inventories

On Products.ProductID = Inventories.ProductID

Join Categories

On Categories.CategoryID = Products.CategoryID

Where InventoryDate Between '01/01/2017' And '02/28/2017'

And CategoryName = 'Seafood'

Group By CategoryName, ProductName

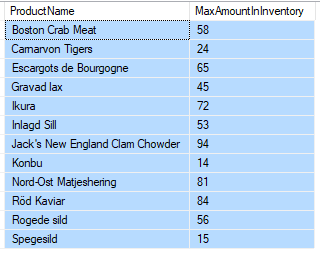
Having Max(Inventories.Count) >= 10

Order By ProductName Asc;

go

***Figure 2.2: Practical use of ‘GROUP BY’ clause utilized with the aggregate function ‘MAX’ and operator ‘Greater or Equal to’.***

As shown below on (figure 2.3), the results from the script presented above on (figure 2.2) displays the maximum inventory count for the Seafood category between January and February of 2017 which has been grouped by the product name.



***Figure 2.3: Results of ‘GROUP BY’ clause utilized with the aggregate function ‘MAX’ and operator ‘Greater or Equal to’.***

# Summary

To recap, the third module taught me the concepts of SQL statements’ main clauses and aggregate functions available in MS SQL to query the database. Reinforcing the fact, the data analyst would: 1) use aggregate functions to return results from a table’s attribute(s) requiring arithmetical or statistical computations, 2) use clauses to query and get data required by the user, and 3) need to use GROUP BY clause combined with an aggregate function to return results requiring arithmetic or statistical calculations which is then grouped by a defined segment(s). Additionally, the assignment gave me the opportunity to understand the primary difference between the WHERE and HAVING clauses. Lastly, the DB created for this assignment gave me the opportunity to put in practice some of the aggregate functions, clauses such as GROUP BY, HAVING, WHERE, MAX, as well as Operators.