MENTORNESS ARTICLE

TASK 1

BATCH NAME : MIP-DA-02

MASTERING SQL AGGREGATION:

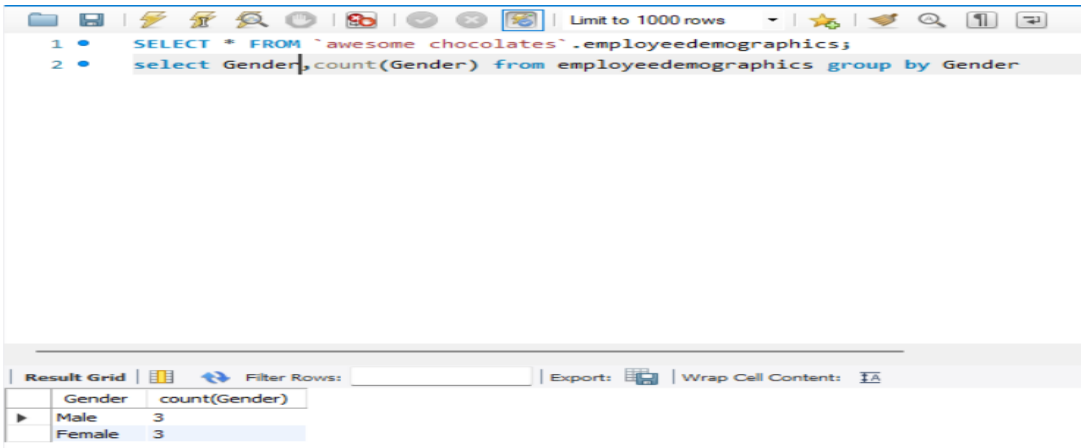
GROUP BY, HAVING CLAUSE, AGGREGATE FUNCTIONS

INTRODUCTION:

SQL, or Structured Query Language, is a very powerful tool in the field of data management and analysis. SQL is a relational database that enables users to retrieve , manipulate and control data easily. In this article, we'll dive deep into SQL Aggregation : Group by, Having Clause & Aggregate Functions. This queries empowers you to unlock the potential of data and make informed decisions. Let's dive in!

1.GROUP BY:

Group by statement is similar to distinct in the select statement. In that it's going to show the unique values in the column the difference is if we say distinct (Gender) , what's going to be returned is the very first unique values of female and the very first unique value of male, and if we say GROUP BY Gender the it's only going to return two values, but we have all the values of male rolled in one row and all values of male rolled in another. Here is an Example below:



In the above example we can see, we have a 'employeedemographics' table where we have a column called Gender. When we select Gender, count(Gender) group by Gender , we see two values Male and Female and there count i.e 3.

Similarly, we can write many complex queries using group by statement.

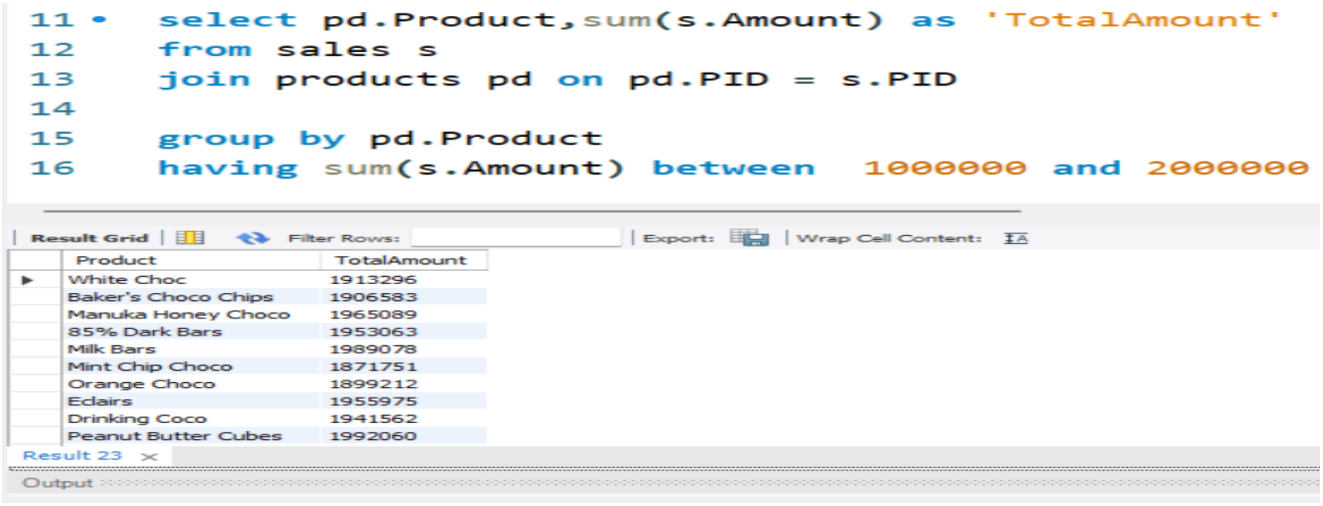
2.HAVING CLAUSE:

The HAVING CLAUSE in SQL is a tool used for filtering data based on some conditions. The HAVING CLAUSE is used together with GROUP BY CLAUSE to filter group of rows. It allow us to set some conditions to aggregated values such as sum, count, etc.

There is a key difference between WHERE CLAUSE AND HAVING CLAUSE. WHERE CLAUSE filters data before the data aggregations but the HAVING CLAUSE filters data after the data aggregation.

WHERE CLAUSE filter individual rows based on column values but the HAVING CLAUSE filters group of rows based on aggregate function results.

Here is an example below:



```
11 •   select pd.Product,sum(s.Amount) as 'TotalAmount'
12     from sales s
13     join products pd on pd.PID = s.PID
14
15     group by pd.Product
16     having sum(s.Amount) between  1000000 and 2000000
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Product	TotalAmount
White Choc	1913296
Baker's Choco Chips	1906583
Manuka Honey Choco	1965089
85% Dark Bars	1953063
Milk Bars	1989078
Mint Chip Choco	1871751
Orange Choco	1899212
Eclairs	1955975
Drinking Coco	1941562
Peanut Butter Cubes	1992060

Result 23 | Output

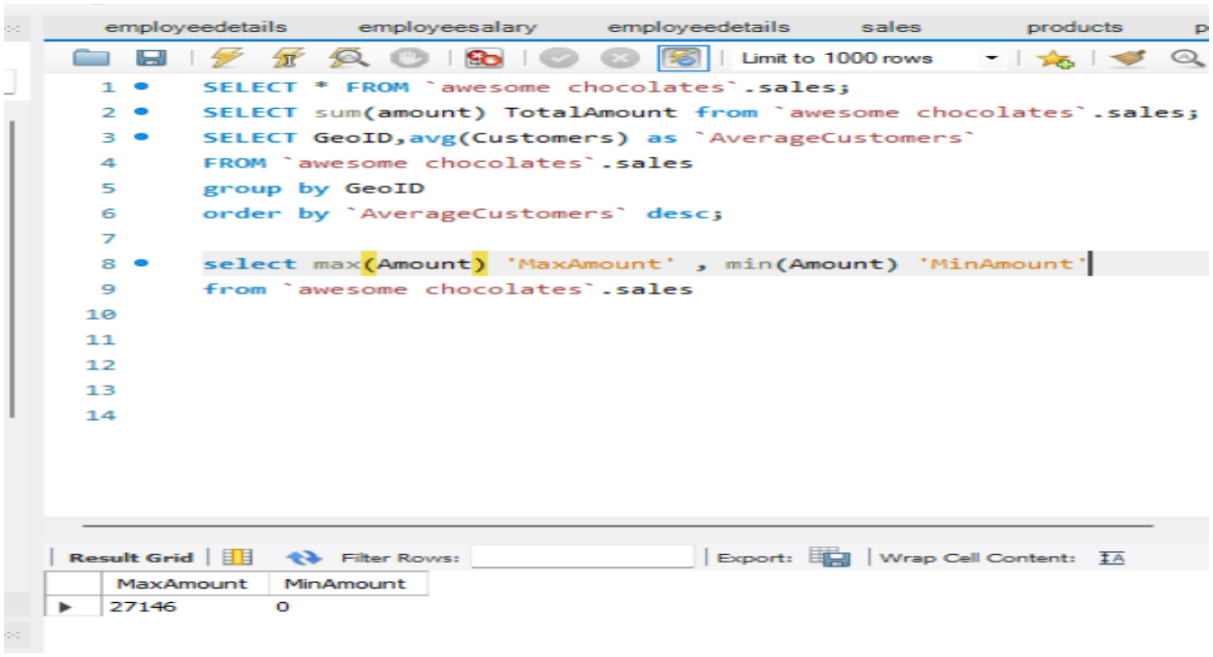
In the above given example, we grouped by product table and we want to find the sum of amount, since sum is an aggregate function, we cannot use WHERE CLAUSE, for that HAVING CLAUSE is used to filter sum of amount based on different products.

3.AGGREGATE FUNCTIONS:

Aggregate functions are very powerful tools for summarising data in database. It performs calculations on a set of values and give a single result. Commonly used aggregate functions are SUM,COUNT,MAX,MIN,AVG. Each function have their own purpose of summarising and analysing data.

Aggregate function often used with GROUP BY clause to perform calculation on group or rows.

Here is an example below of different aggregate functions:



In the above example we can see, SUM,MAX,MIN,AVG being used along with GROUP BY STATEMENT.

CONCLUSION:

In conclusion, the "HAVING" clause in SQL provides a powerful tool for filtering grouped data based on aggregate conditions, enabling more precise analysis and reporting. When combined with the "GROUP BY" statement and aggregation functions, it allows for insightful summaries of data subsets meeting specific criteria. By leveraging these SQL features effectively, analysts and developers can extract valuable insights and make informed decisions from their datasets.