Name: -Suyash Navanath Shinde Roll No: -D2223034
Course: - Database Management System Lab Batch: - B
Division: - TY CSF 2

Assignment No.: 4

Aim: Database Querying – Simple queries, Queries using aggregate functions, GROUP BY and HAVING clauses. (https://learnsql.com/blog/examples-of-sql-group-by/)

A. Write a Group-by query for one/two columns in Manufacturing industry / **Hospital**/ Company table
B. Write a Having clause query for Manufacturing industry / **Hospital**/ Company table
C. Write a queries to make use of aggregate functions Count(),Sum(),Avg(),Min(),Max()

Software required: MySQL

Theory: In a database management system (DBMS), the GROUP BY clause and the HAVING clause are used in conjunction with the SELECT statement to perform advanced data analysis and filtering on groups of rows. Here's a brief introduction to each clause:

GROUP BY Clause:

> The GROUP BY clause is used to group rows in a result set based on one or more columns. It is commonly used in combination with aggregate functions like COUNT, SUM, AVG, MAX, or MIN to perform calculations on groups of data. The result of a GROUP BY query is a set of rows where each row represents a unique combination of values in the specified column(s) and the aggregate function(s) are applied to the corresponding groups.

For example, if you have a "Sales" table with columns like "Product", "Category", and "Revenue", you can use the GROUP BY clause to calculate the total revenue per product category. The query might look like this:

```
SELECT Category, SUM(Revenue) AS TotalRevenue
FROM Sales
GROUP BY Category;
```

HAVING Clause:

> The HAVING clause is used to filter the results of a GROUP BY query based on specified conditions. It allows you to apply filtering criteria to the grouped data after the aggregation has taken place. This clause operates similarly to the WHERE clause, but while the WHERE clause filters individual rows, the HAVING clause filters groups of rows.

Continuing with the previous example, if you want to retrieve only the product categories with total revenue greater than a certain value, you can use the HAVING clause. For instance:

SELECT Category, SUM(Revenue) AS TotalRevenue
FROM Sales
GROUP BY Category
HAVING SUM(Revenue) > 100000;

This query will return the product categories with a total revenue greater than 100,000.

SQL Commands:

A. Write a Group-by query for one/two columns in Manufacturing industry / **Hospital**/ Company table

Syntax:

SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;

Example:

Suppose we have a "Manufacturing" table, and we want to group the companies by their "Location."

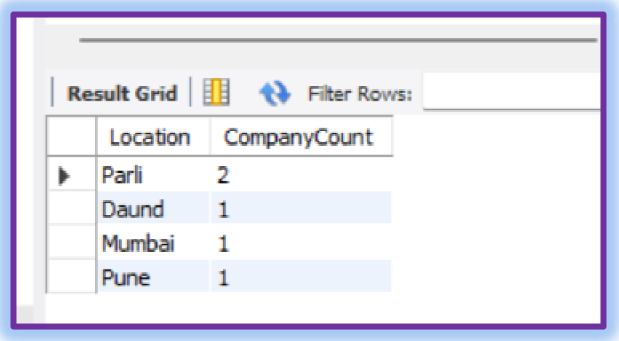
```
USE mydatabase;
-- Create a Manufacturing table
CREATE TABLE Manufacturing (
    CompanyID INT PRIMARY KEY,
    Name VARCHAR(255),
    Location VARCHAR(100)
);

-- Insert dummy data into the Manufacturing table
INSERT INTO Manufacturing (CompanyID, Name, Location)
VALUES
    (1, 'Company Rohit', 'Parli'),
    (2, 'Company Dhanuja', 'Daund'),
    (3, 'Company Prakhar', 'Mumbai'),
    (4, 'Company Pranjali', 'Pune'),
    (5, 'Company Chaitanya', 'Parli');


-- Group the companies by location and count how many companies are in each location
SELECT Location, COUNT(*) AS CompanyCount
FROM Manufacturing
GROUP BY Location;
```

This query will group the manufacturing companies by their location and count how many companies are in each location.

OUTPUT:



Location	CompanyCount
Parli	2
Daund	1
Mumbai	1
Pune	1

B. Write a Having clause query for Manufacturing industry / **Hospital**/ Company table

Syntax:

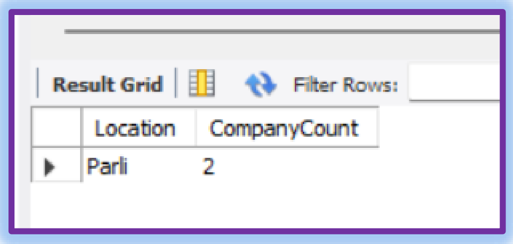
SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
HAVING *condition;*

Example:
Using the same "Manufacturing" table, let's say we want to find locations with more than three manufacturing companies.

```
-- Group the companies by location and filter out locations with more than three manufacturing companies
SELECT Location, COUNT(*) AS CompanyCount
FROM Manufacturing
GROUP BY Location
HAVING COUNT(*) >= 2;
```

This query groups the companies by location and filters out the locations with more than three manufacturing companies.

OUTPUT:



Location	CompanyCount
Parli	2

C. Write a queries to make use of aggregate functions Count(),Sum(),Avg(),Min(),Max()

Syntax:

SELECT COUNT(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT AVG(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT SUM(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT COUNT(ProductID)
FROM Products;

SELECT AVG(Price)
FROM Products;

SELECT SUM(Quantity)
FROM OrderDetails;

Example:

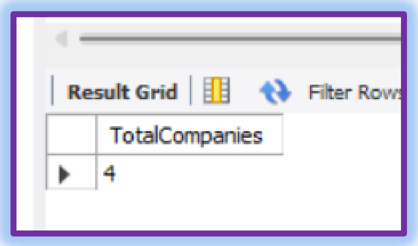
```
  USE mydatabase;
-- Create a Company table
CREATE TABLE Company (
    CompanyID INT PRIMARY KEY,
    Name VARCHAR(255),
    Revenue DECIMAL(10, 2),
    Profit DECIMAL(10, 2)
);
```

```
-- Insert dummy data into the Company table
INSERT INTO Company (CompanyID, Name, Revenue, Profit)
VALUES
    (1, 'Company X', 10000.50, 2500.75),
    (2, 'Company Y', 8000.25, 1800.50),
    (3, 'Company Z', 12000.75, 3000.25),
    (4, 'Company W', 9500.00, 2200.80);
```

- Count: To count the number of manufacturing companies.

```
-- Count the number of companies
SELECT COUNT(*) AS TotalCompanies FROM Manufacturing;
```

OUTPUT:

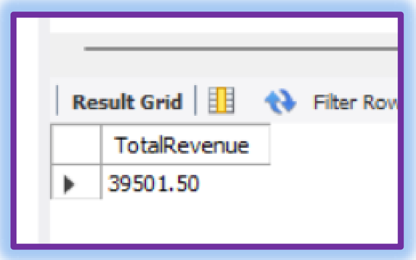


```
TotalCompanies
4
```

- Sum: To calculate the total revenue of all companies.

```
-- Calculate the total revenue of all companies
SELECT SUM(Revenue) AS TotalRevenue FROM Company;
```

OUTPUT:

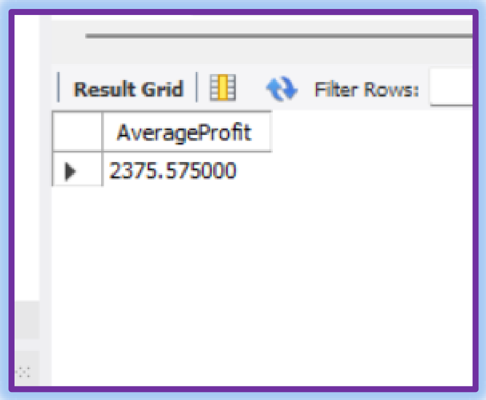


```
TotalRevenue
39501.50
```

- Avg: To calculate the average age of employees in a hospital.

```
-- Calculate the average profit of all companies
SELECT AVG(Profit) AS AverageProfit FROM Company;
```

OUTPUT:

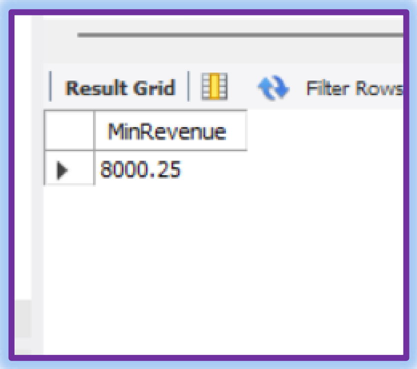


	AverageProfit
▶	2375.575000

- Min: To find the minimum capacity among hospitals.

```
-- Find the minimum revenue among companies
SELECT MIN(Revenue) AS MinRevenue FROM Company;
```

OUTPUT:

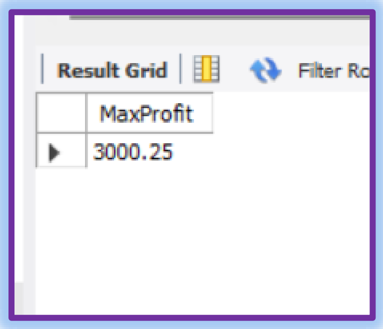


	MinRevenue
▶	8000.25

- Max: To find the maximum profit in the manufacturing industry.

```
-- Find the maximum profit among companies
SELECT MAX(Profit) AS MaxProfit FROM Company;
```

OUTPUT:



	MaxProfit
▶	3000.25

Conclusion: In summary, the GROUP BY clause helps to group rows based on specific columns, while the HAVING clause allows you to filter the grouped data based on aggregate conditions. Together, they provide powerful tools for data analysis and summarization in DBMS.

FAQs: (Answer all FAQs using suitable examples)

I. What is the difference between the WHERE clause and the HAVING clause?

II. Can I use the GROUP BY clause without any aggregate functions?

III. Can I use the HAVING clause without the GROUP BY clause?

IV. Can I include columns in the SELECT statement that are not part of the GROUP BY clause?

V. Can I use multiple aggregate functions in the HAVING clause?

Additional problem statements:

Write an SQL query to analyze sales data by grouping it by product category and cI.alculate the total revenue for each category. Display only the categories with total revenue greater than $10,000.

II. Develop an SQL query to retrieve the department names and the count of employees in each department. Display only the departments that have more than 50 employees.

III. Create an SQL query to group the products by their suppliers and calculate the average stock quantity for each supplier. Display only the suppliers whose average stock quantity exceeds 500 units.

IV. Write an SQL query to group customers by their age range (e.g., 18-25, 26-35, etc.) and calculate the count of customers in each age range. Display only the age ranges with more than 100 customers.

V. Develop an SQL query to analyze exam scores by grouping them by the subject and calculate the average score for each subject. Display only the subjects with an average score above 80.