**The University Of Azad Jammu & Kashmir,**

**Muzaffarabad**

**Department of Software Engineering**

**LAB TASK 08**

**Database Systems**

**Course Code**: **CS-2204**

**Submitted To:**

Engr. Tahir Jahangir

**Submitted By:**

Ahmed Ali

**Roll No:**

2023-SE-41

**Session**: 2023-2027

Contents

[Lab Task – 8: SQL Server – Grouping & Aggregation 4](#_Toc207132772)

[📘 Task 01: Database and Table Setup 4](#_Toc207132773)

[📝 Task 02: Applying Basic Aggregate Functions 6](#_Toc207132774)

[📊 Task 03: GROUP BY Clause in Action 8](#_Toc207132775)

[📗 Task 04: Filtering Groups Using HAVING Clause 10](#_Toc207132776)

[✏️ Task 05: Conceptual Understanding of Grouping and Aggregation 12](#_Toc207132777)

[Summary: 14](#_Toc207132778)

# Lab Task – 8: SQL Server – Grouping & Aggregation

# 📘 Task 01: Database and Table Setup

**Objective:** Recreate database and tables used in previous labs to work with fresh aggregation queries.

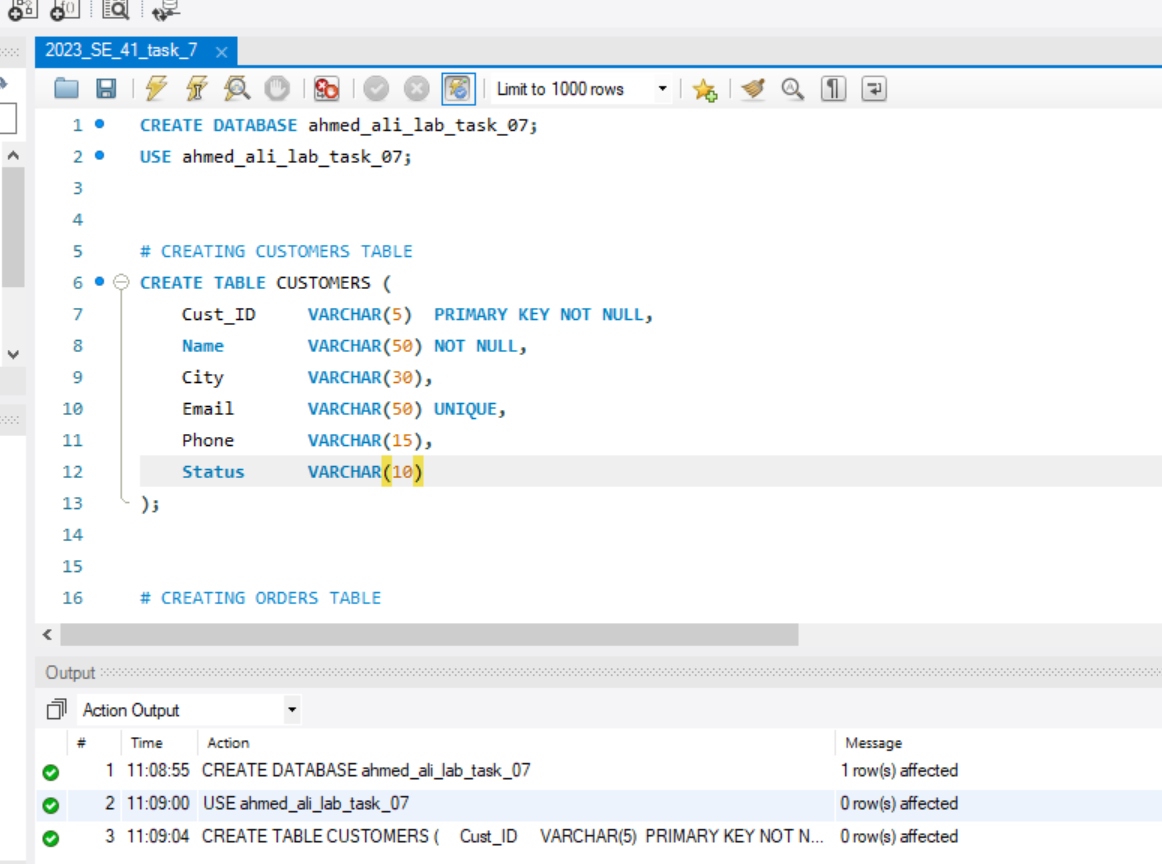
**Instructions:**

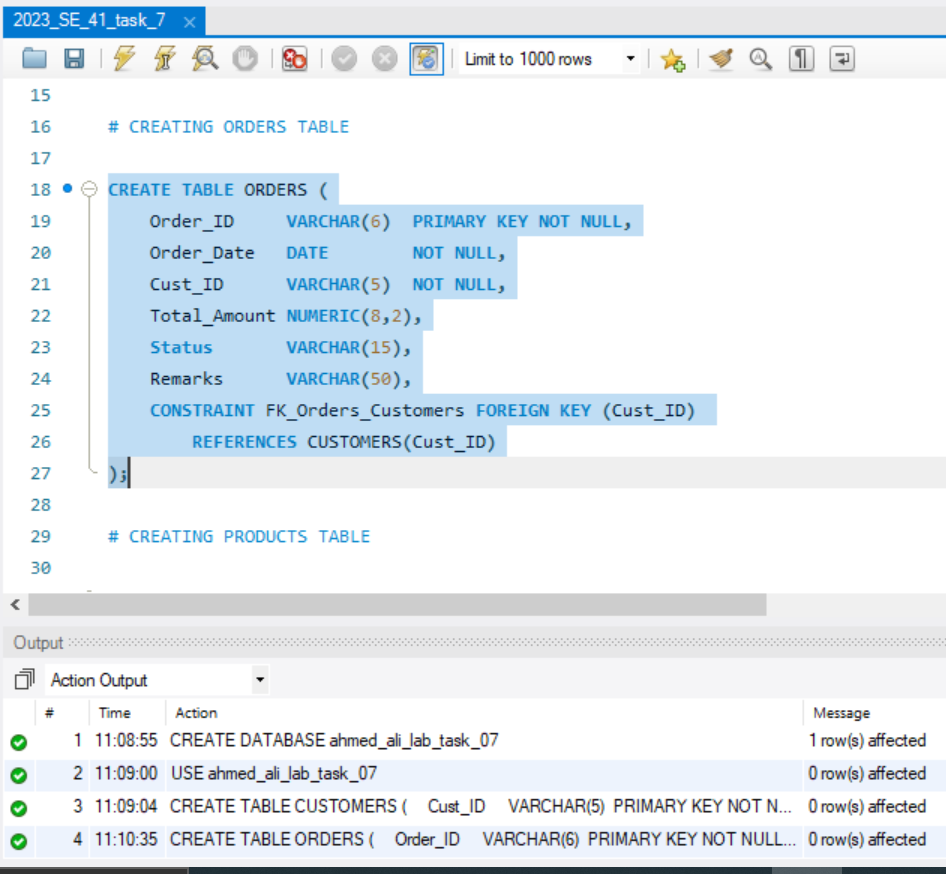
1. Create a new database using your name in the format:

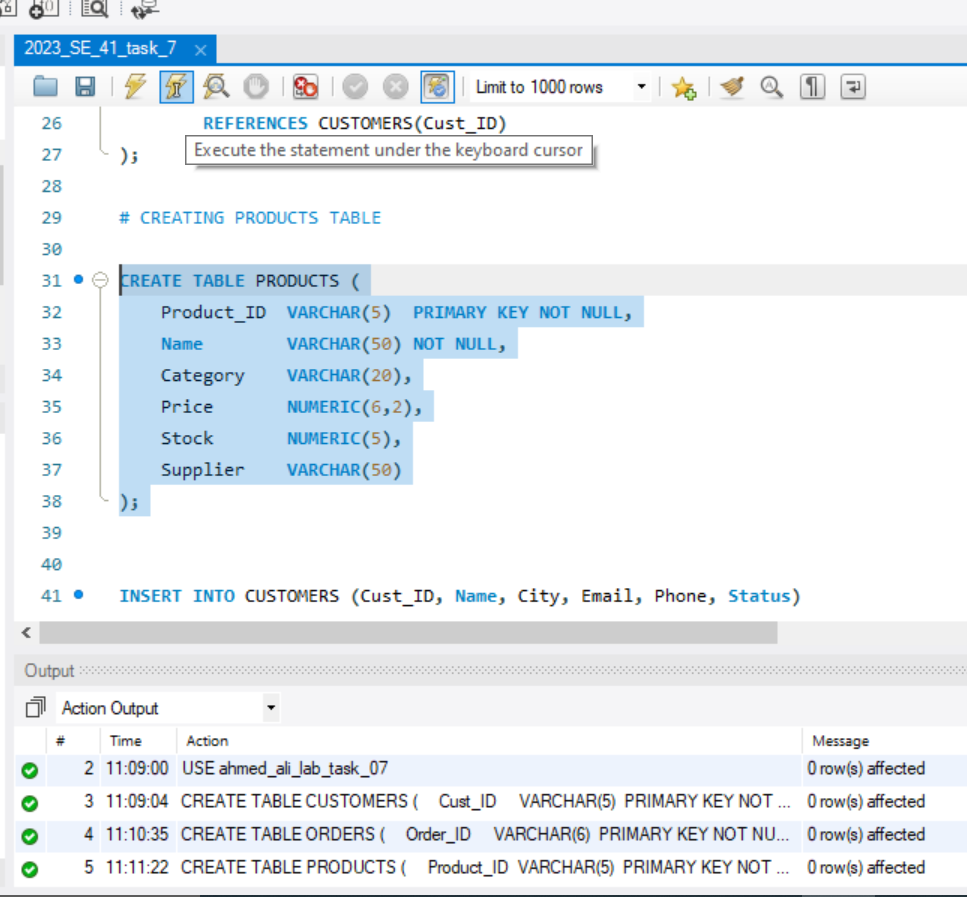
your\_full\_name\_lab\_task\_08

*Example:* bushra\_batool\_lab\_task\_08

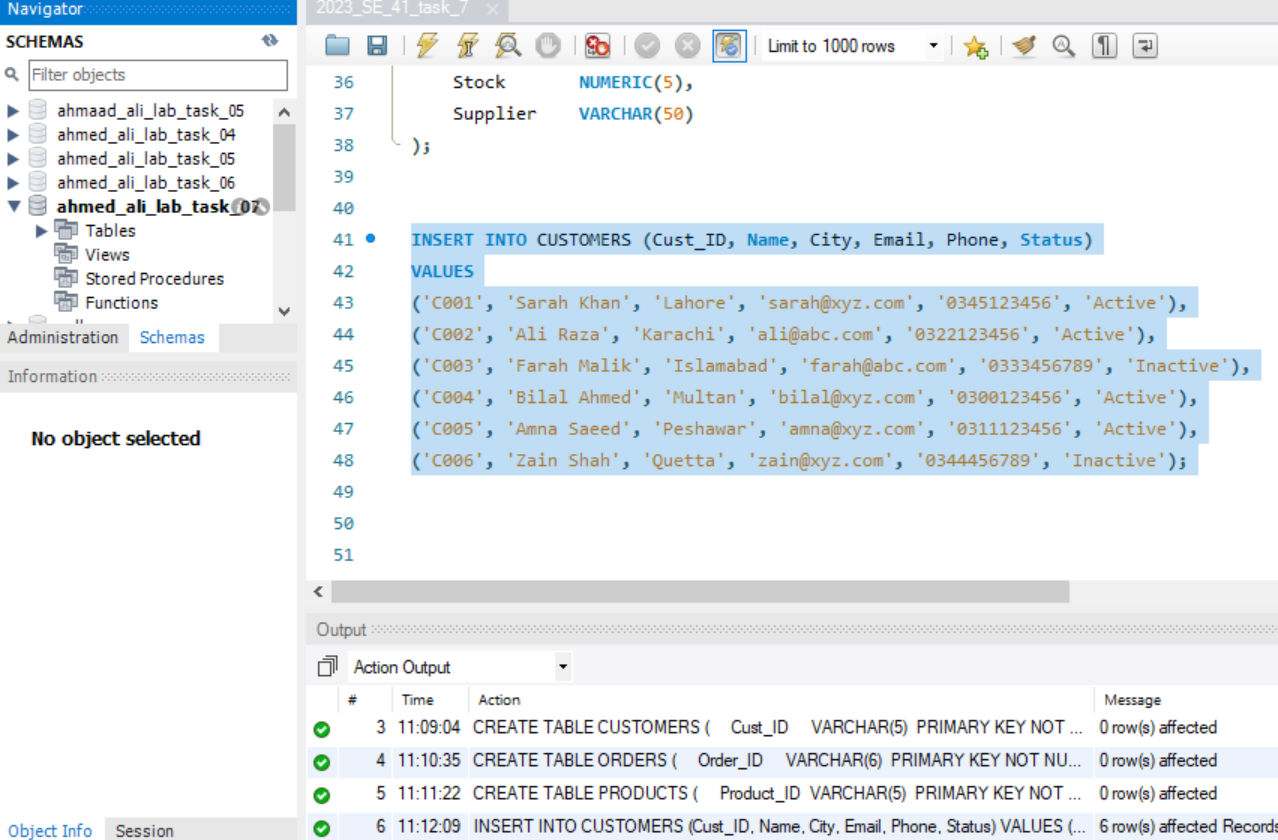
1. Recreate the following tables and insert the provided records: o **CUSTOMERS** o **ORDERS** o **PRODUCTS**

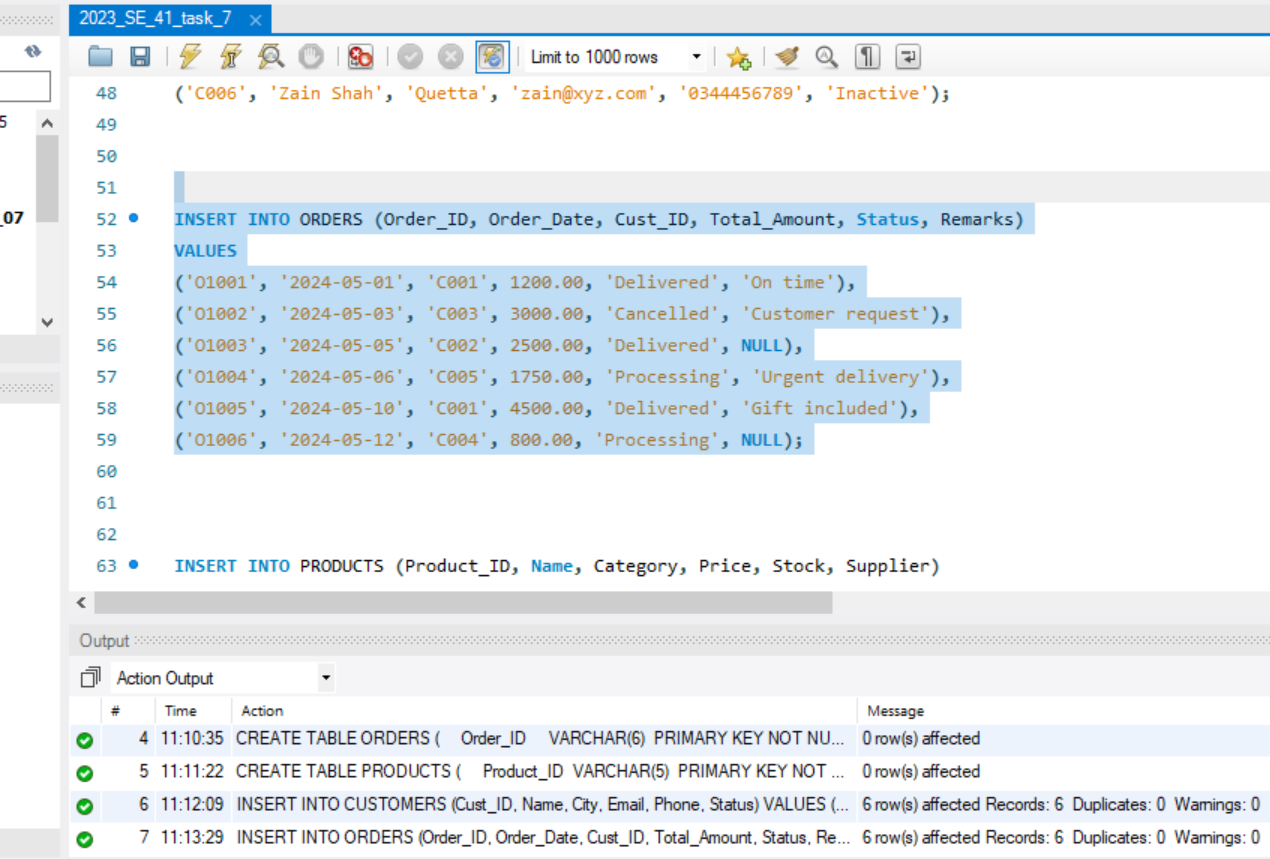


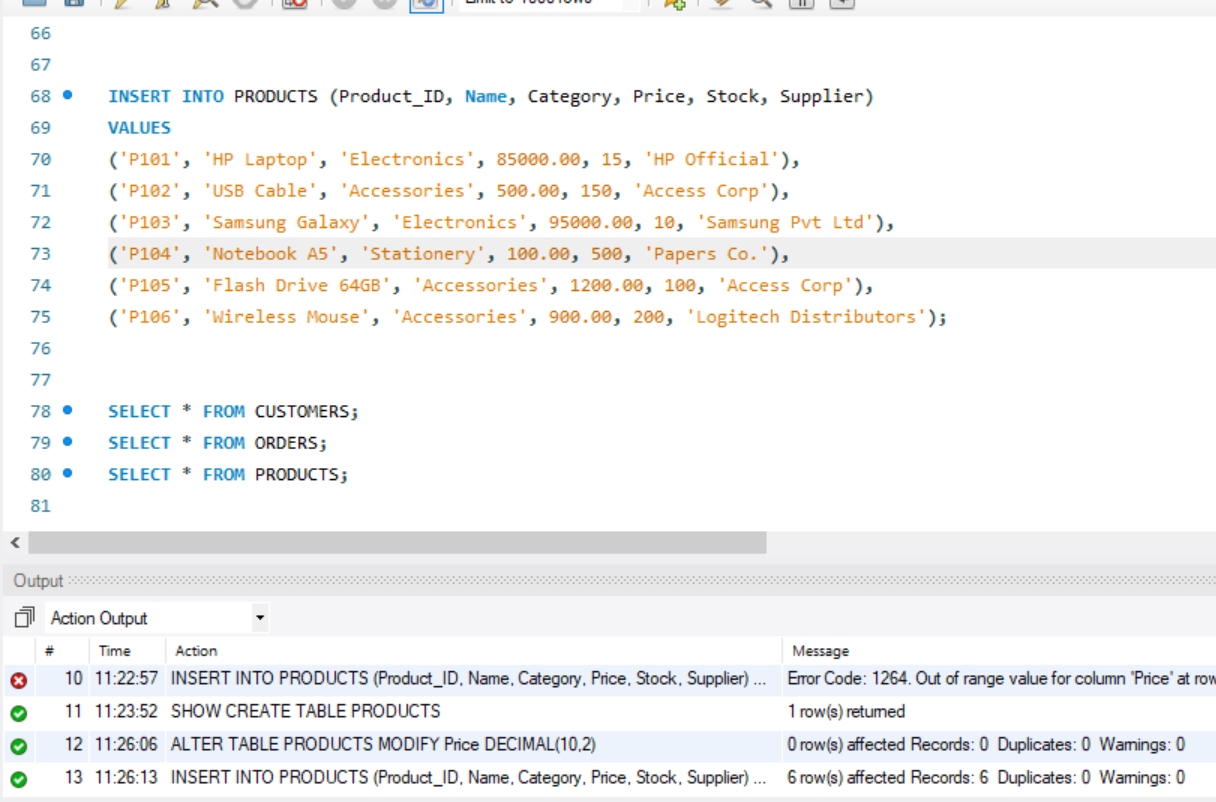
****



**Data Insertion:**

****

****



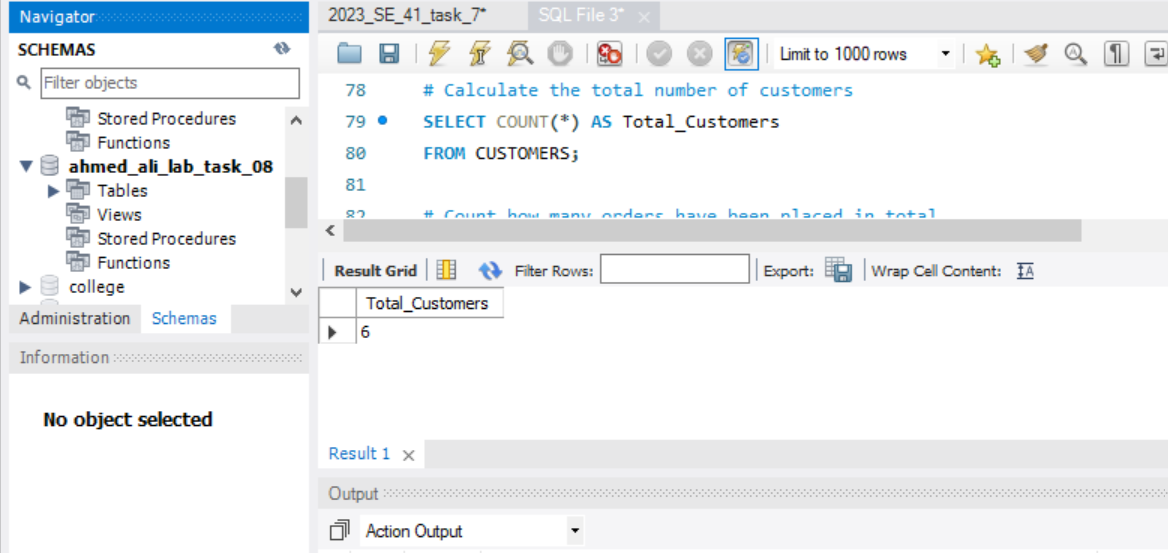
# 📝 Task 02: Applying Basic Aggregate Functions

**Objective:** Use aggregate functions to derive summarized data from individual columns.

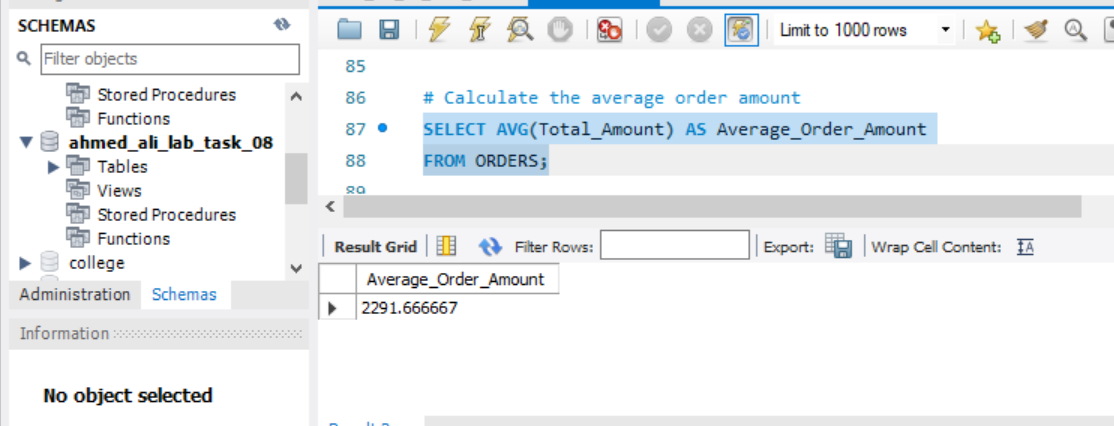
**Instructions:**

Write SQL queries to:

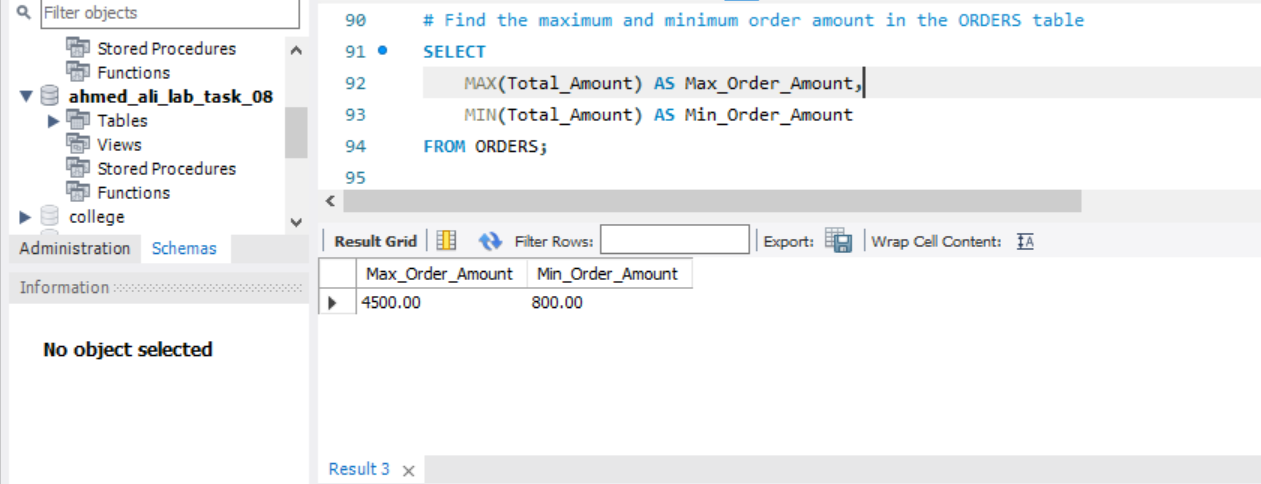
1. **Calculate the total number of customers.**



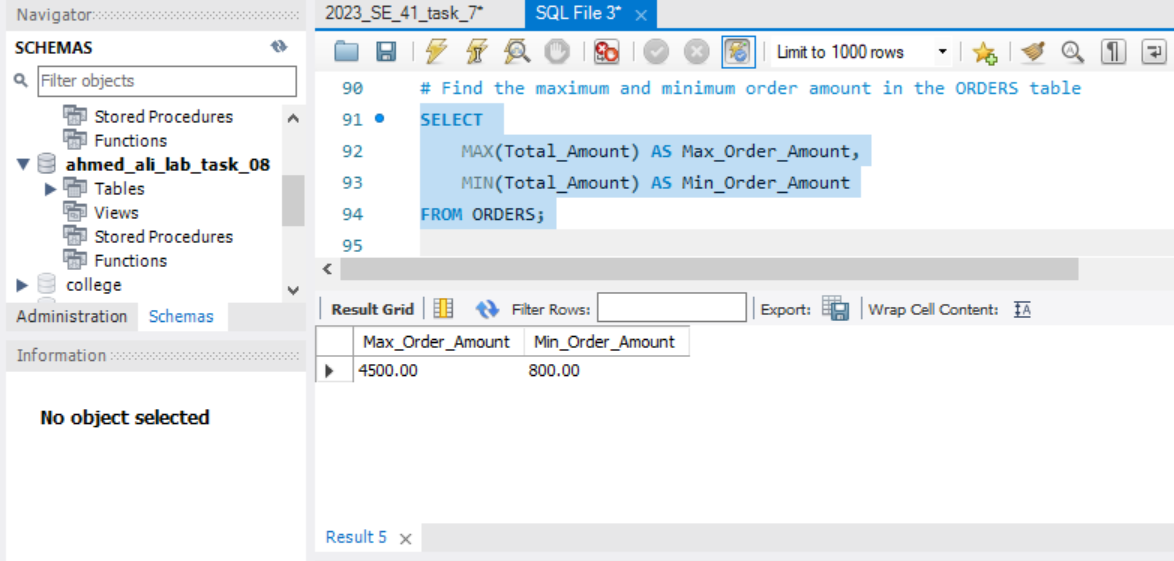
1. **Count how many orders have been placed in total.**



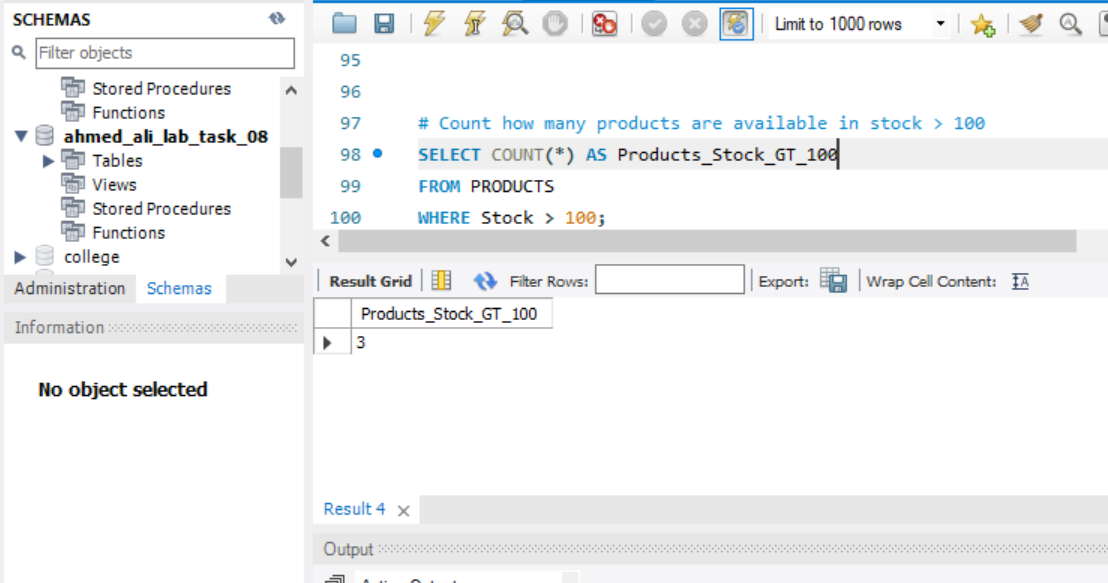
1. **Calculate the average order amount.**



1. **Find the maximum and minimum order amount in the ORDERS table.**



1. **Count how many products are available in stock > 100.**

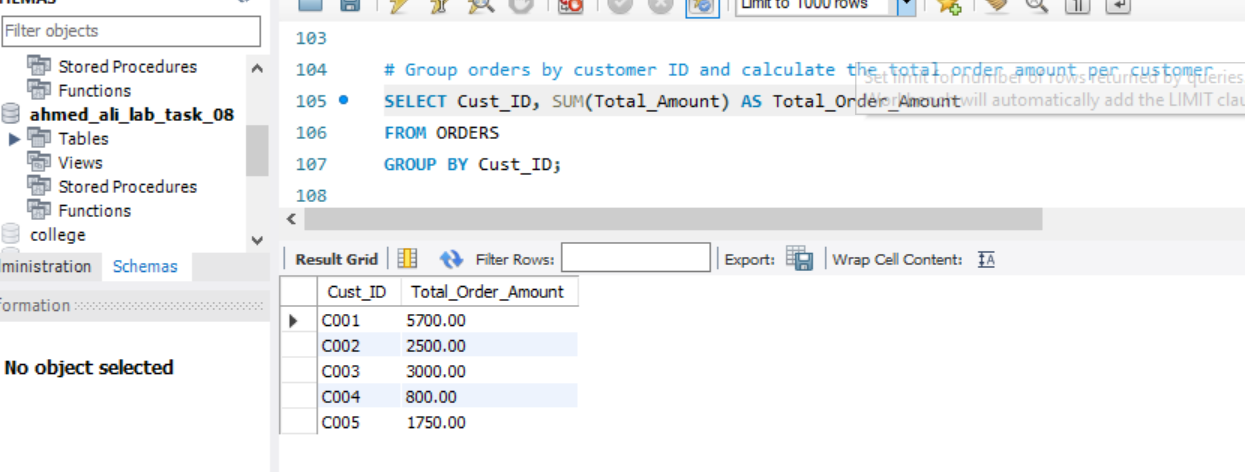


# 📊 Task 03: GROUP BY Clause in Action

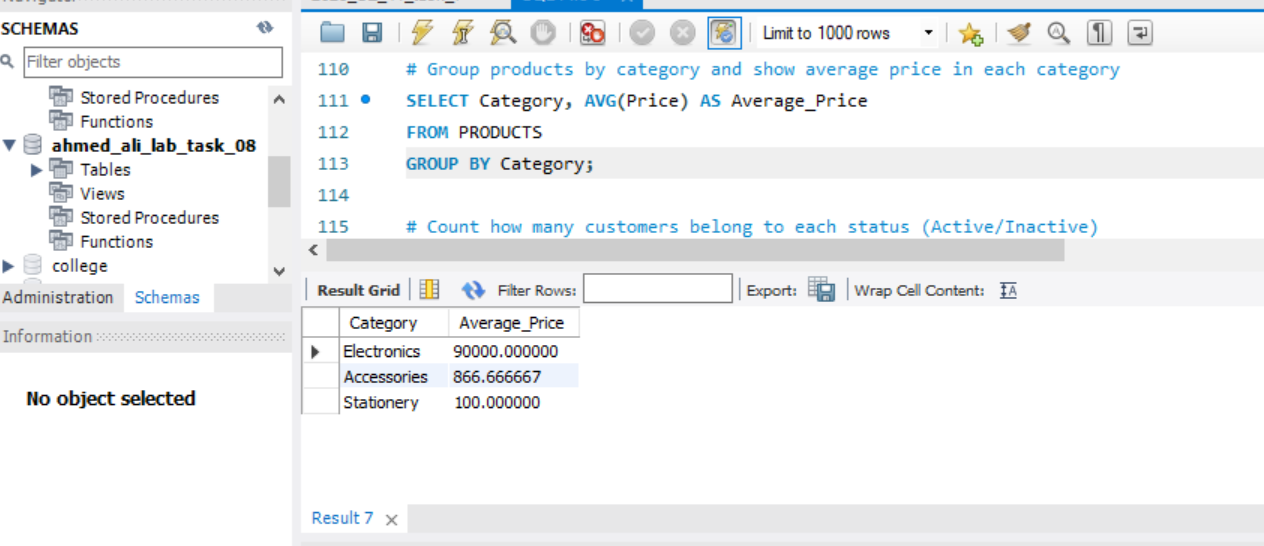
**Objective:** Apply GROUP BY to organize results based on common field values.

**Instructions:**

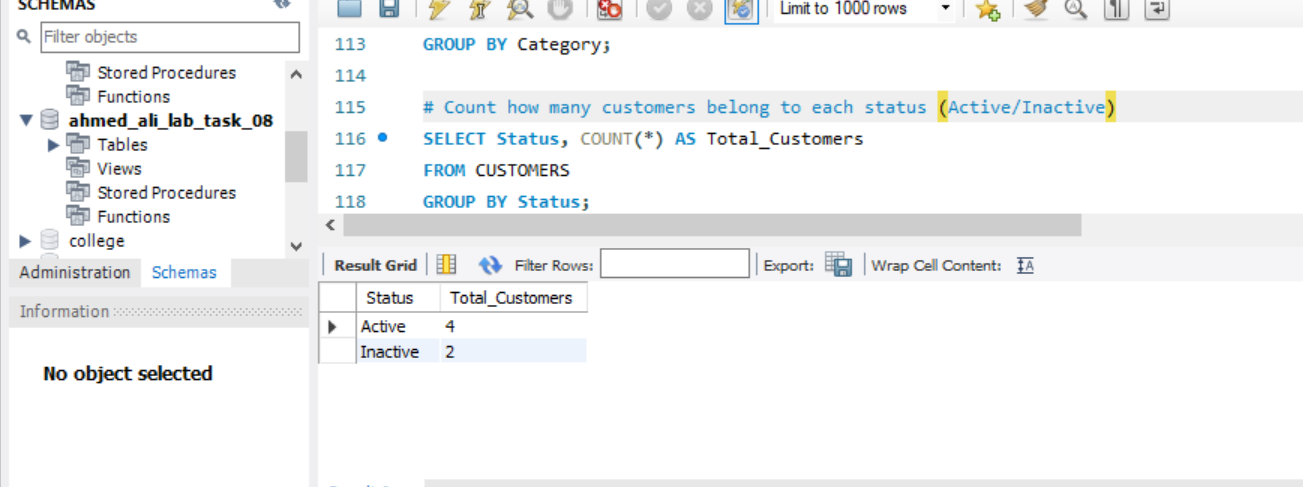
1. **Group orders by customer ID and calculate the total order amount per customer.**



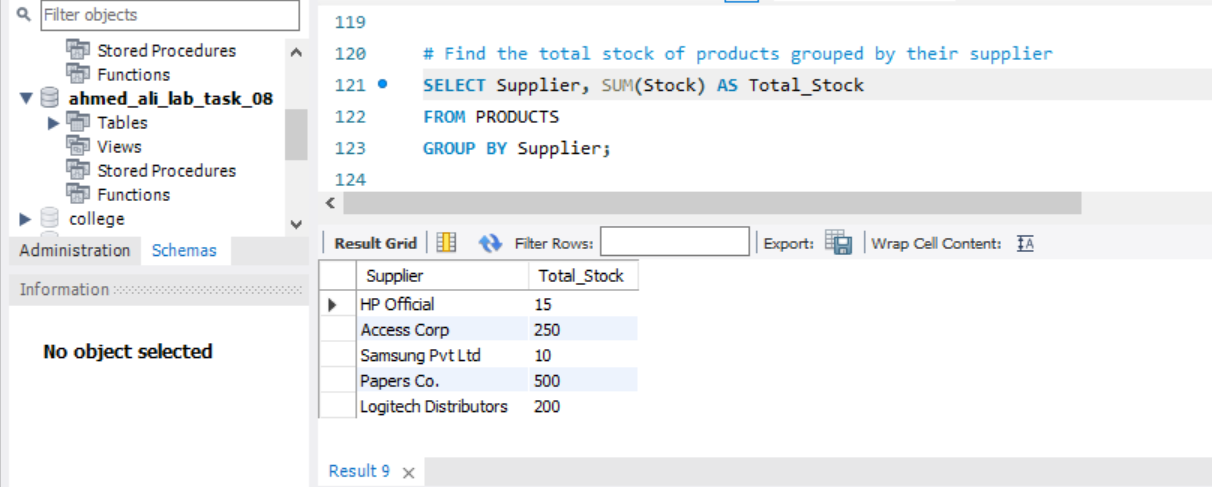
1. **Group products by category and show average price in each category.**



1. **Count how many customers belong to each status (Active/Inactive).**



1. **Find the total stock of products grouped by their supplier.**

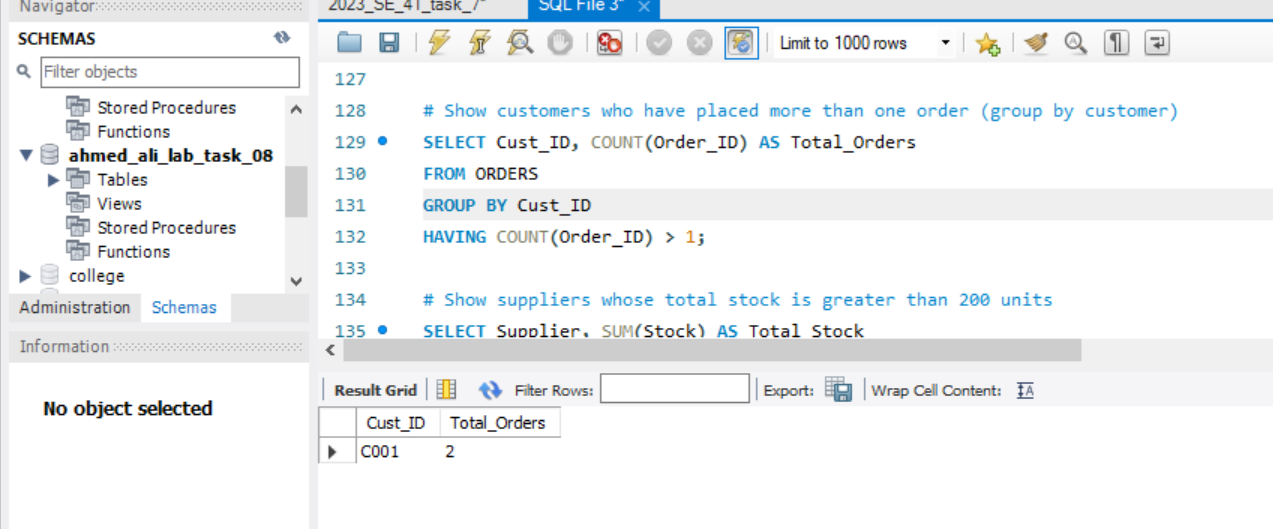


# 📗 Task 04: Filtering Groups Using HAVING Clause

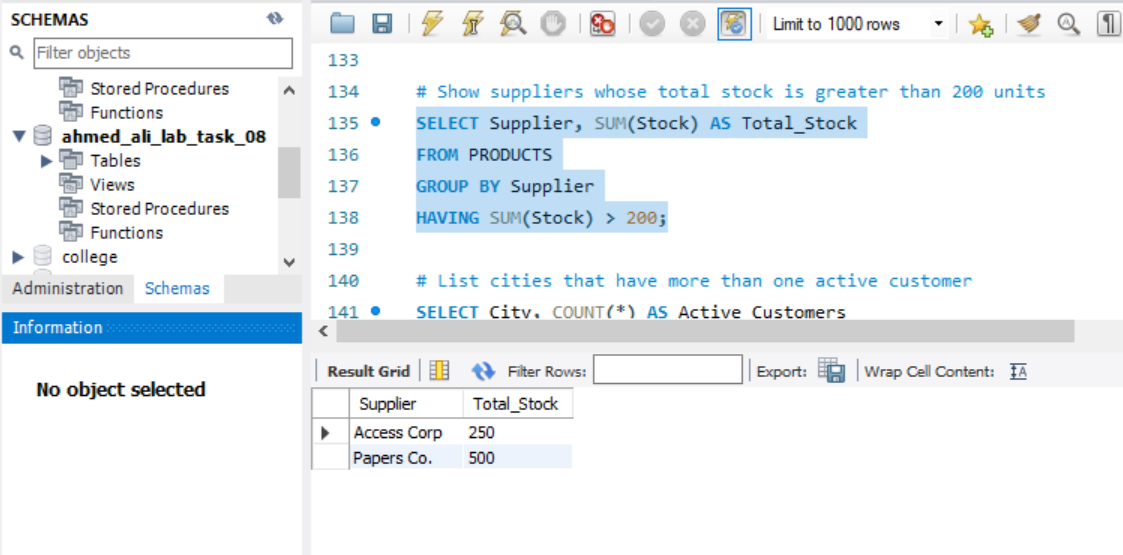
**Objective:** Learn how to filter grouped results using HAVING.

**Instructions:**

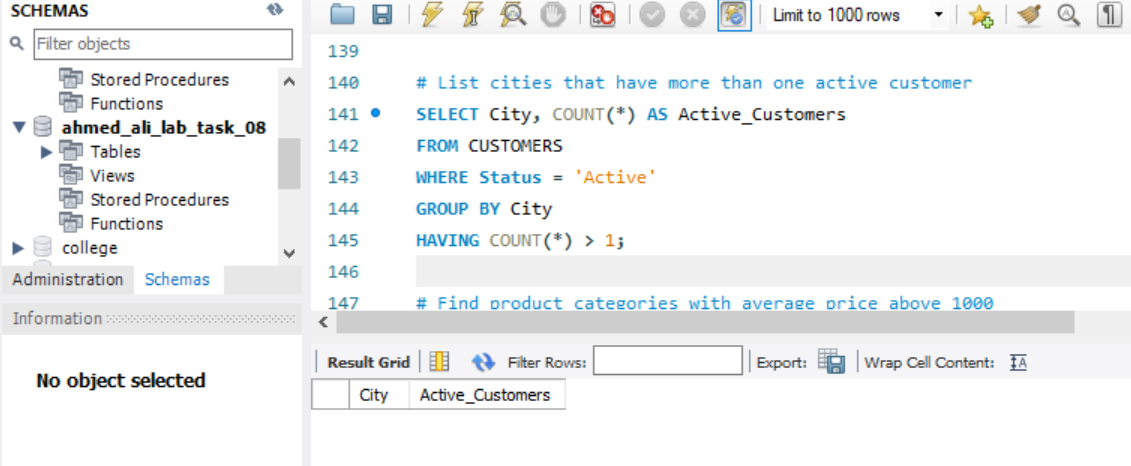
1. **Show customers who have placed more than one order (group by customer).**



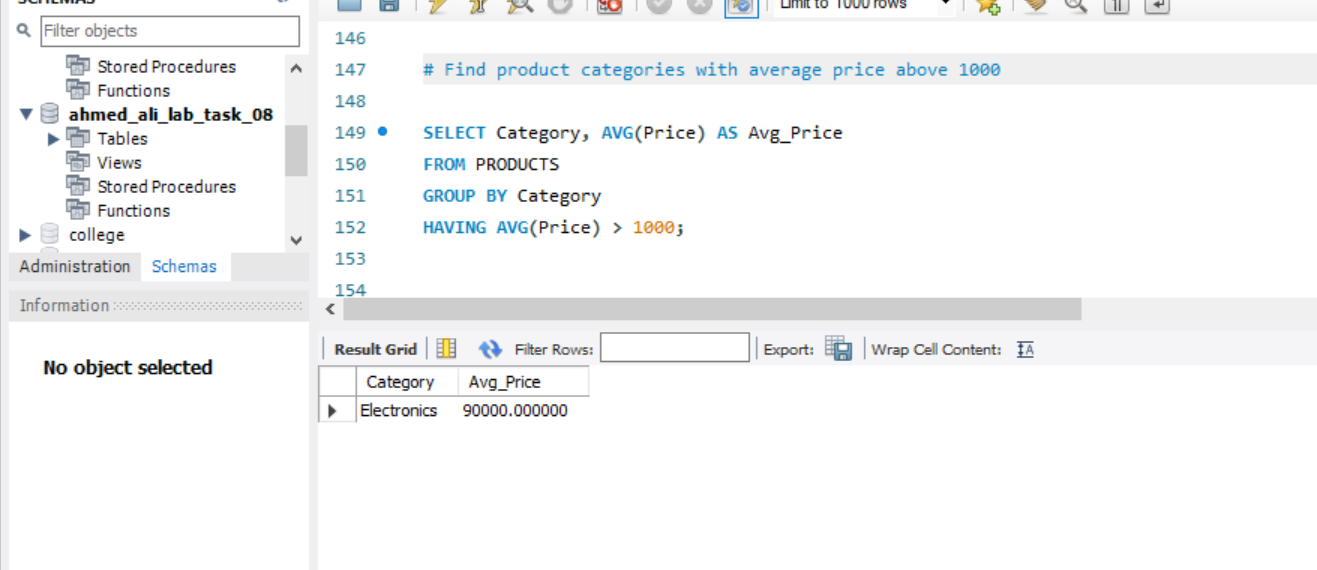
1. **Show suppliers whose total stock is greater than 200 units.**



1. **List cities that have more than one active customer.**



1. **Find product categories with average price above 1000.**



# ✏️ Task 05: Conceptual Understanding of Grouping and Aggregation

**Objective:** Assess student understanding of aggregate functions and grouping.

**Instructions:**

Answer the following theoretical questions in your lab documentation:

1. **What are aggregate functions in SQL? List and briefly explain five of them.**

Aggregate functions perform a calculation on a set of values and return a single summarized value.  
They are mostly used with **GROUP BY** to analyze grouped data.

**Five common aggregate functions:**

| **Function** | **Description** | **Example** |
| --- | --- | --- |
| **COUNT()** | Returns the number of rows. | SELECT COUNT(\*) FROM CUSTOMERS; |
| **SUM()** | Returns the total sum of a numeric column. | SELECT SUM(Total\_Amount) FROM ORDERS; |
| **AVG()** | Returns the average value of a numeric column. | SELECT AVG(Price) FROM PRODUCTS; |
| **MAX()** | Returns the highest value in a column. | SELECT MAX(Total\_Amount) FROM ORDERS; |
| **MIN()** | Returns the lowest value in a column. | SELECT MIN(Price) FROM PRODUCTS; |

1. **What is the difference between WHERE and HAVING clause? Present in tabular form with examples.**

| **Aspect** | **WHERE Clause** | **HAVING Clause** |
| --- | --- | --- |
| **Purpose** | Filters rows before grouping/aggregation. | Filters groups after aggregation. |
| **Use with Aggregate Functions** | Cannot be used directly with aggregates. | Specifically designed to filter aggregate results. |
| **Execution Order** | Applied **before GROUP BY**. | Applied **after GROUP BY**. |
| **Example** | SELECT \* FROM ORDERS WHERE Status = 'Delivered'; | SELECT Cust\_ID, SUM(Total\_Amount) FROM ORDERS GROUP BY Cust\_ID HAVING SUM(Total\_Amount) > 3000; |

1. **Why is it necessary to use GROUP BY when working with aggregate functions?**

 **GROUP BY** is necessary when we want aggregate results for **subsets of data**, not for the entire table.

 Without GROUP BY, aggregate functions compute a single value for the entire table.

 With GROUP BY, the table is divided into groups, and aggregate functions are applied to each group separately.

1. **Explain with example how HAVING helps filter grouped results.**

 The HAVING clause is like a **WHERE clause for groups**.

 It filters groups after aggregation is performed.

SELECT Cust\_ID, COUNT(Order\_ID) AS Total\_Orders

FROM ORDERS

GROUP BY Cust\_ID

HAVING COUNT(Order\_ID) > 1;

 Groups orders by customer.

 Then shows only customers with **more than one order**.

 Without HAVING, we would get results for all customers, even those with a single order.

1. **What will happen if an aggregate function is used without GROUP BY?**

If no GROUP BY is used, the aggregate function works on the **entire table** and returns a **single summarized value**.

SELECT AVG(Total\_Amount) FROM ORDERS;

Returns the average order amount for all orders in the table.

* If GROUP BY was used, it would return average per customer (or per group).

Summary:

In this lab, I learned how aggregate functions such as COUNT, SUM, AVG, MAX, and MIN are used to summarize data in SQL. I explored how the GROUP BY clause allows us to organize data into groups and apply aggregates to each group separately, while the HAVING clause is used to filter grouped results based on aggregate conditions. I also understood the key difference between WHERE (which filters rows before grouping) and HAVING (which filters groups after aggregation). Finally, I realized that when aggregate functions are used without GROUP BY, they return a single summarized value for the entire table. Overall, these concepts are essential for data analysis and reporting in SQL.