ONLINE ORDERS DETAILS

MYSQL PROJECT DESCRIPTION

Shahid Abbas

Email: shahidabbas2104514@gmail.com

Objective:

The objective of this project was to collect data from an online database and to cover each step of creating a data repository inside MySQL Workbench.
After creating a data repository,next step was to apply functions like basic aggregation,where,group by,having,Subqueries,CTEs and Window functions to find insights like

- Total Orders
- Total Revenue

- Orders and Revenue by Payment Method

- Top 3 Products by Sales Revenue

- Orders with Total Price > Average

- Products with Unit Price > Average

- Top-Selling Product by Revenue

- Countries with Revenue < Max Country Revenue

- Total Revenue per Product (CTE)

- Rank Products by Revenue (Window)

- Order Value vs Overall Average (Window)

Tools:

- MySQL

- Powerpoint (for project description)

Creating Database and Data table:

The first step in this project is to create a database. We can also use an already created database and add a new table in it.

CREATE DATABASE Online_Orders_Details;
USE Online_Orders_Details;

The next step is to create a table inside our database. Before creating a table make sure that the table has equal number of columns with same data type as in our .csv data file downloaded from kaggle or any other database.

CREATE TABLE Orders_Details
(Order_ID INT PRIMARY KEY,
Country Varchar(20),
Product_Category Varchar(50),
Product Varchar(20),
Unit_Price INT,
Quantity INT,
Total_Price INT,
Order_Purchased_Date DATE,
Order_Received_Date DATE,
Payment_Method Varchar(50),
Review_Rating INT);
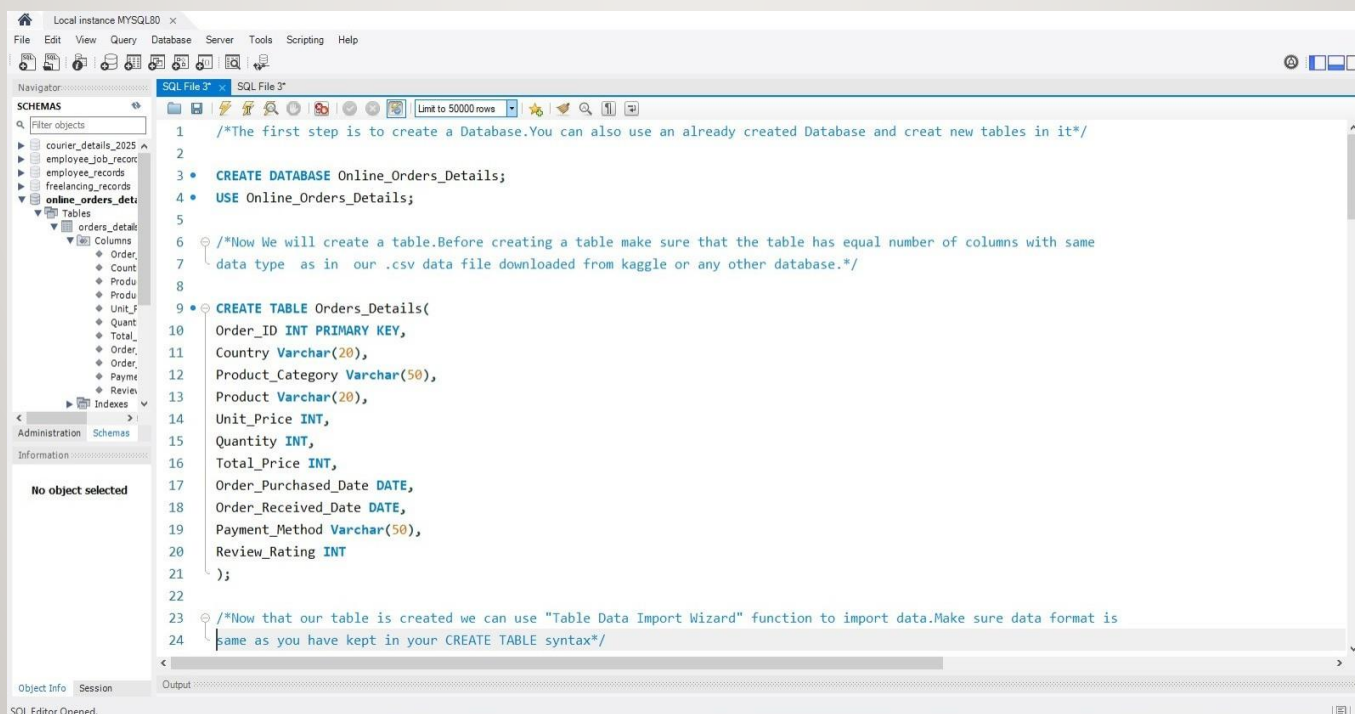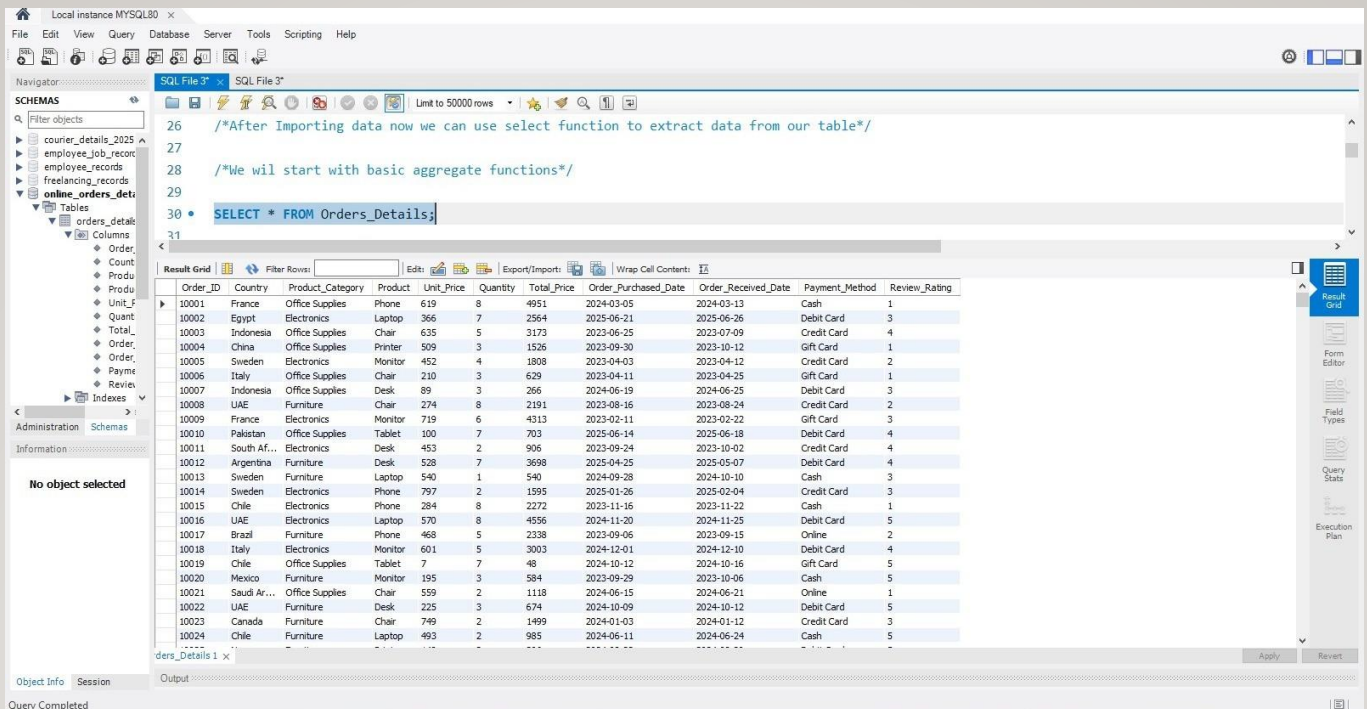
Now that our table is created we can use "Table Data Import Wizard" function to import data. Make sure data format is same as you have kept in your CREATE TABLE syntax

Preview Data:

In the following step we will use SELECT query to preview our entire data table containing all the entries in each column.

*SELECT * FROM Orders_Details;*

Total Orders:

In order to calculate the total number of orders placed we wil use following query.
SELECT COUNT(Order_ID) AS Total_Orders
FROM Orders_Details;

Total Revenue

Calculate the total revenue generated from all orders.

SELECT SUM(Total_Price) AS Total_Revenue FROM Orders_Details;

Average Unit Price:

In this step we will calculate average unit price from our orders.

SELECT AVG(Unit_Price) AS Avg_Unit_Price
FROM Orders_Details;
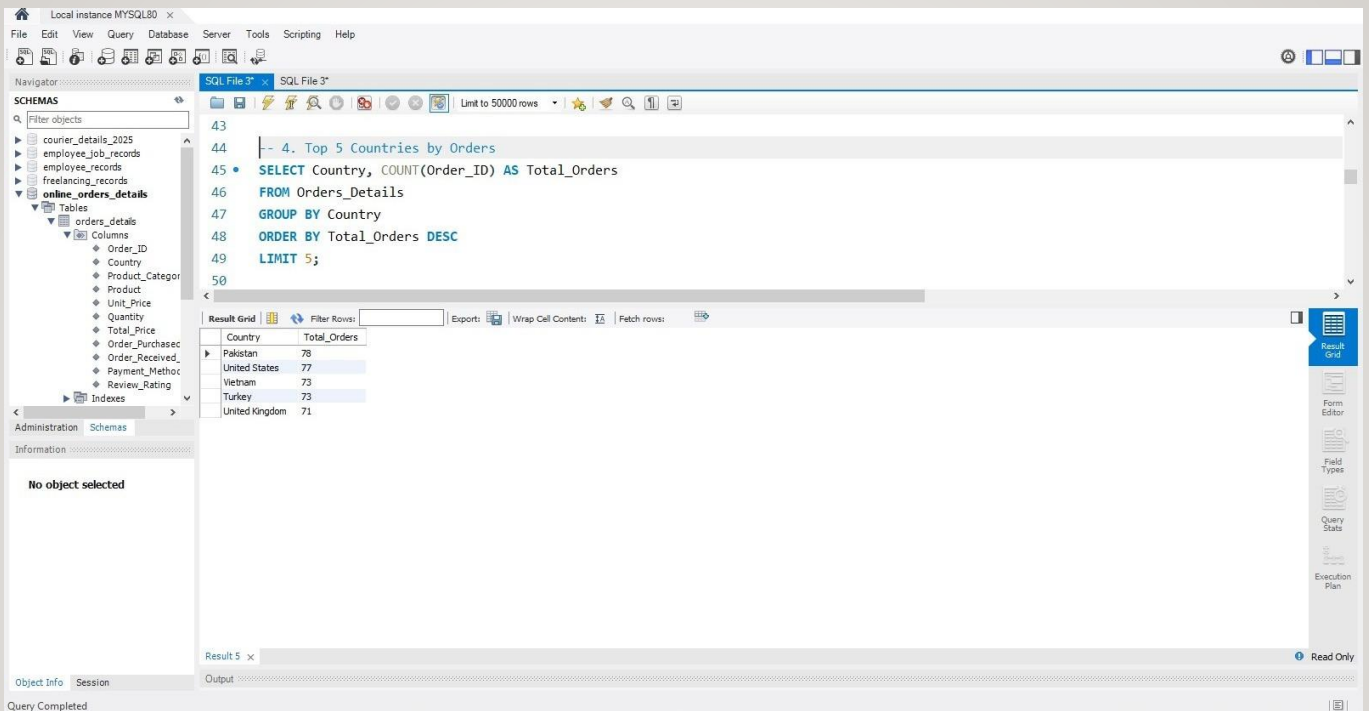
Top 5 Countries by Orders:

In this query we will find out top 5 countries with the most amount of orders placed.This query will employ Group by,Order by and Limit function for the first time in this project.

SELECT Country, COUNT(Order_ID) AS Total_Orders
FROM Orders_Details
GROUP BY Country
ORDER BY Total_Orders DESC
LIMIT 5;
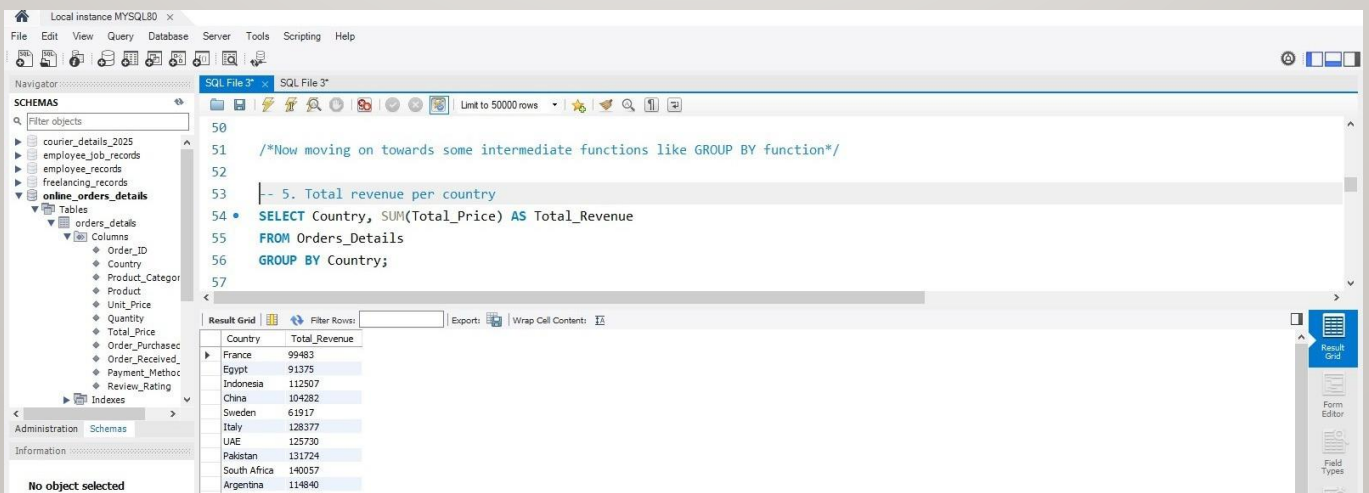
Total Revenue per Country:

In this step we will find out the total revenue made in each country.
SELECT Country, SUM(Total_Price) AS Total_Revenue
FROM Orders_Details
GROUP BY Country;
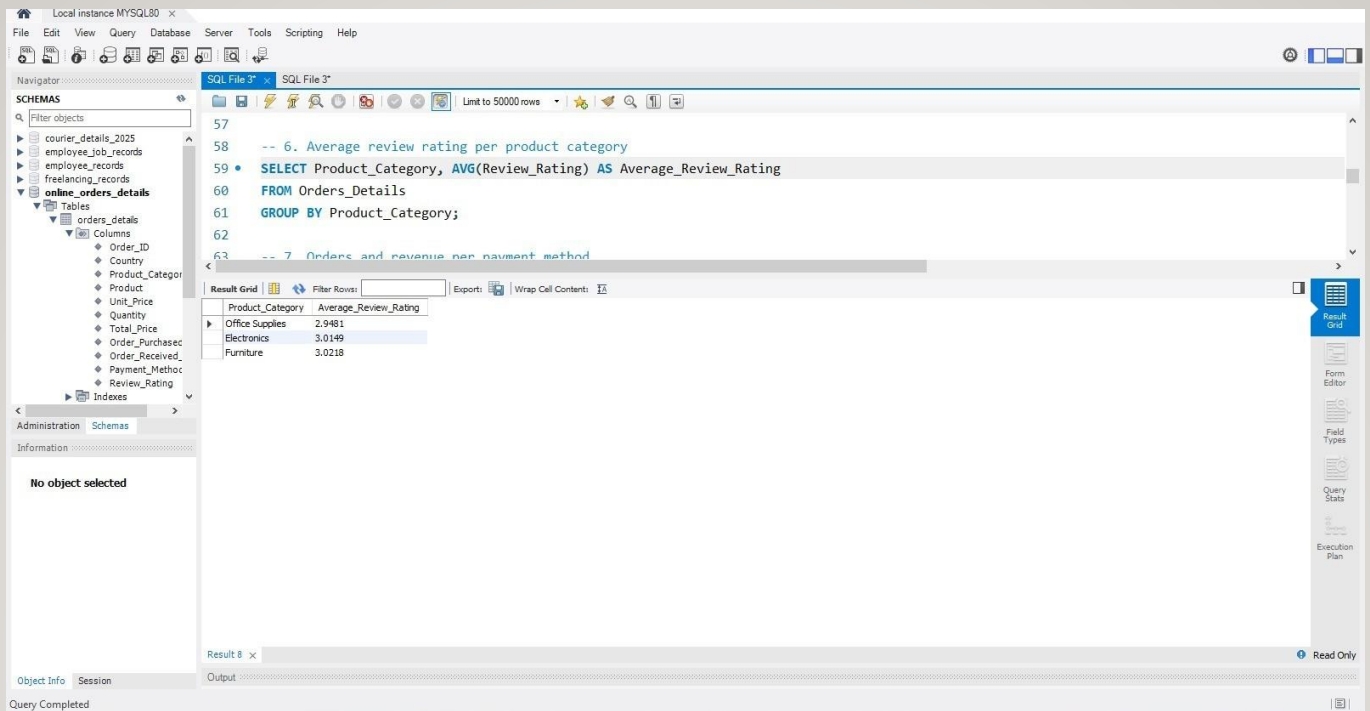
Average Review Rating by Product Category:

This query gives us average review rating for each product category.

SELECT Product_Category, AVG(Review_Rating) AS Average_Review_Rating

FROM Orders_Details

GROUP BY Product_Category;

Orders and Revenue by Payment Method:

Analyze orders and revenue grouped by payment method.

SELECT Payment_Method, COUNT(Order_ID) AS Total_Orders, SUM(Total_Price) AS Total_Revenue FROM Orders_Details

GROUP BY Payment_Method;

Top 3 Products by Sales Revenue:

This query lists the top three products ranked by total sales revenue.

SELECT Product, SUM(Total_Price) AS Total_Sales_Revenue FROM Orders_Details
GROUP BY Product
ORDER BY Total_Sales_Revenue DESC LIMIT 3;
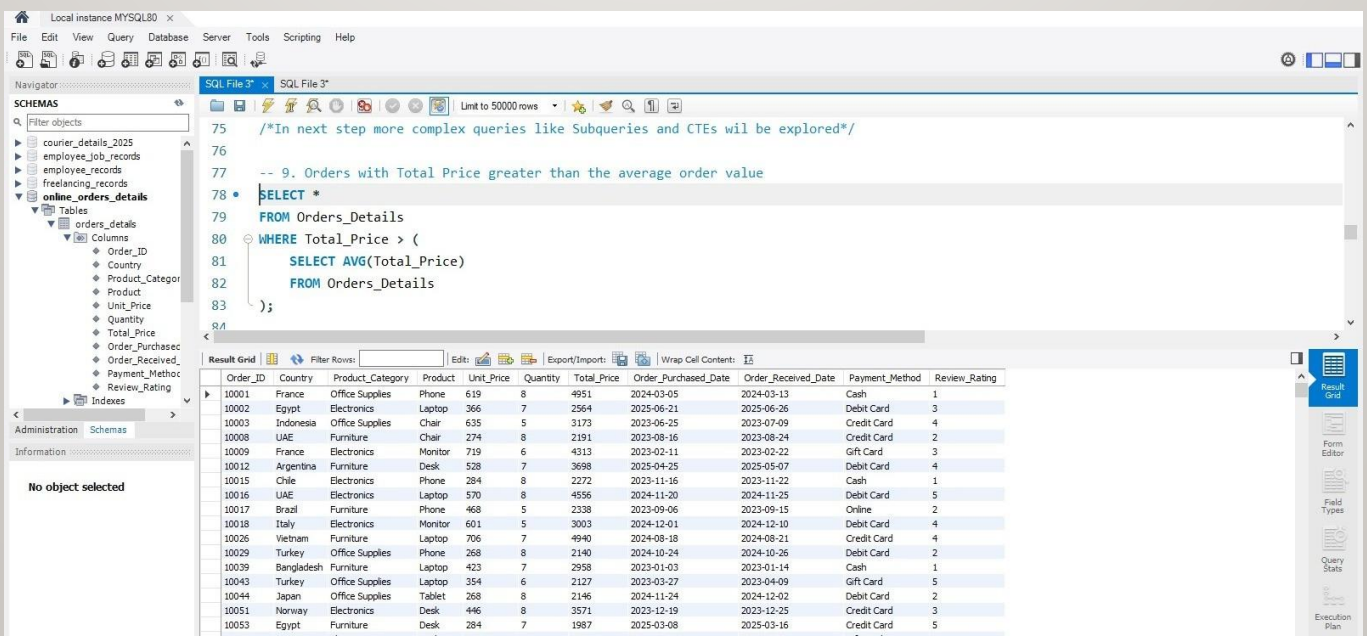
Subqueries and CTEs

In next steps more complex queries like Subqueries and CTEs wil be explored.

Orders with Total Price greater than the average order value:

SELECT *
FROM Orders_Details
WHERE Total_Price > (
    SELECT AVG(Total_Price)
    FROM Orders_Details);

Products with Unit Price greater than the average Unit Price:

SELECT Product, Unit_Price
FROM Orders_Details
WHERE Unit_Price > (
    SELECT AVG(Unit_Price)
    FROM Orders_Details
);

Top-selling product by revenue (highest Total_Revenue):

SELECT Product, SUM(Total_Price) AS Total_Revenue
FROM Orders_Details
GROUP BY Product
ORDER BY Total_Revenue DESC
LIMIT 1;

Show Countries with Revenue less than the Maximum Country Revenue:

SELECT Country, SUM(Total_Price) AS Total_Revenue

FROM Orders_Details
GROUP BY Country
HAVING Total_Revenue < (
    SELECT MAX(Total_Revenue)
    FROM (
     SELECT Country, SUM(Total_Price) AS
     Total_Revenue FROM Orders_Details
     GROUP BY Country
) AS X
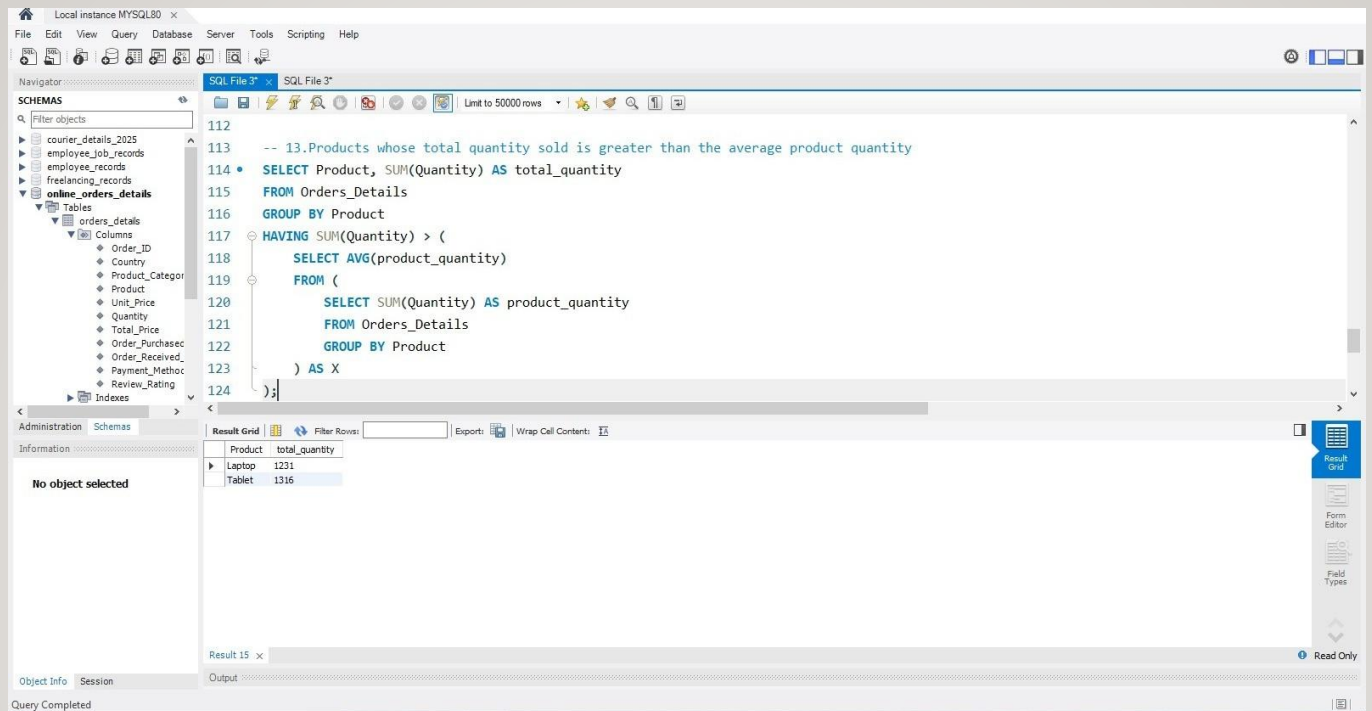
Products with Total Quantity > Average:

Find products with total quantity sold greater than the average product quantity.

```
SELECT Product, SUM(Quantity) AS total_quantity FROM Orders_Details
GROUP BY Product
HAVING SUM(Quantity) > (
        SELECT AVG(product_quantity) FROM (
                SELECT SUM(Quantity) AS product_quantity FROM Orders_Details
                GROUP BY Product
        ) AS X
);
```
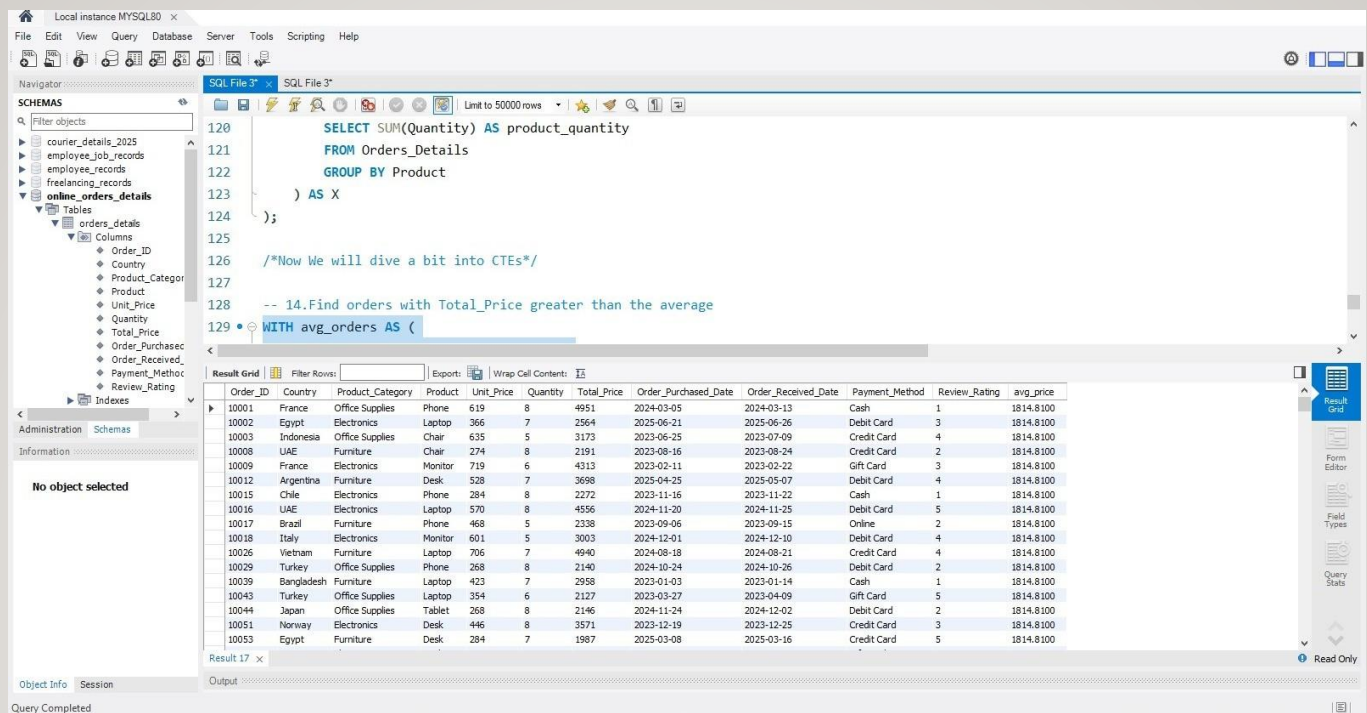
Orders with Total Price > Average (CTE):

In this query we will usese a CTE to select orders with a total price above the average.

```sql
WITH avg_orders AS (
    SELECT AVG(Total_Price) AS avg_price FROM Orders_Details
)
SELECT *
FROM Orders_Details, avg_orders WHERE Total_Price > avg_price;
```


Total Revenue per Product:

```sql
WITH product_revenue AS (
    SELECT Product, SUM(Total_Price) AS total_revenue FROM Orders_Details
    GROUP BY Product
)
SELECT Product, total_revenue FROM product_revenue;
```
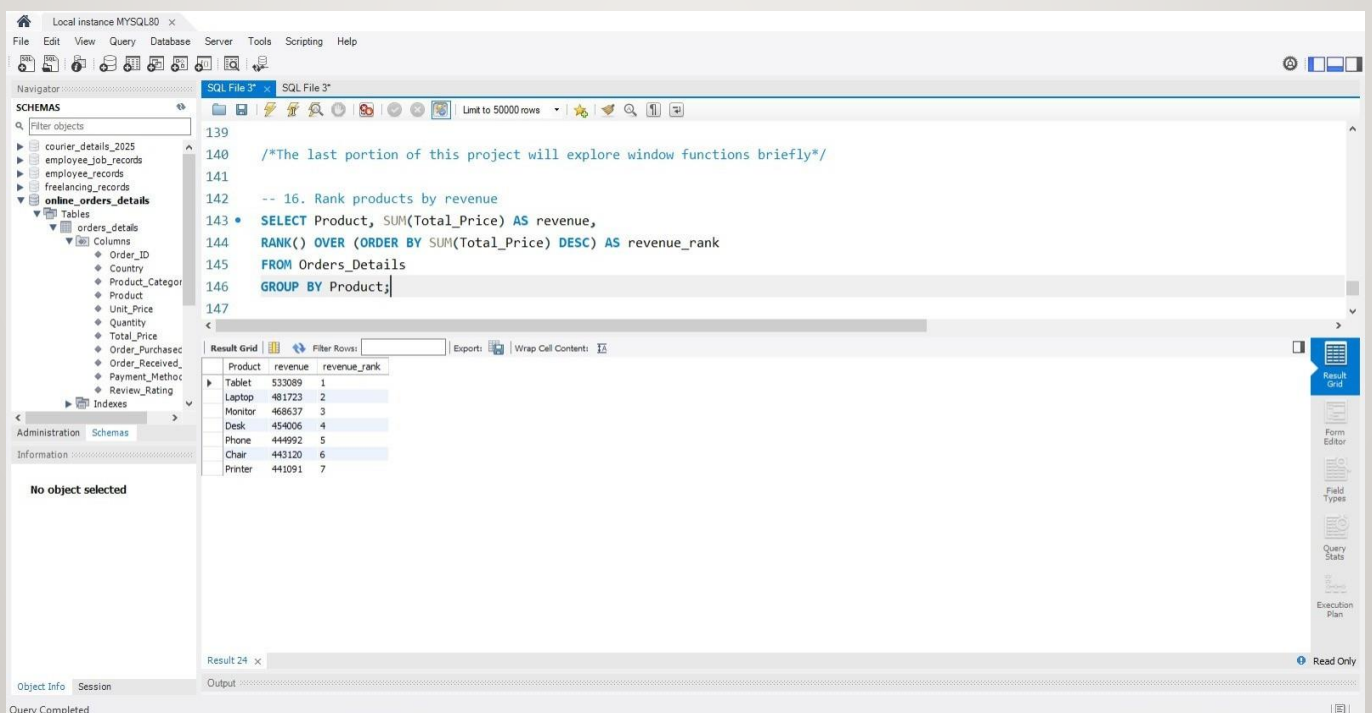
Window Functions

Rank Products by Revenue:

In this query we will rank products based on revenue using a window function.

WITH product_sales AS (

    SELECT Product, SUM(Total_Price) AS revenue

        FROM Orders_Details

        GROUP BY Product

)

SELECT Product, revenue,

      RANK() OVER (ORDER BY revenue DESC) AS revenue_rank

FROM product_sales;
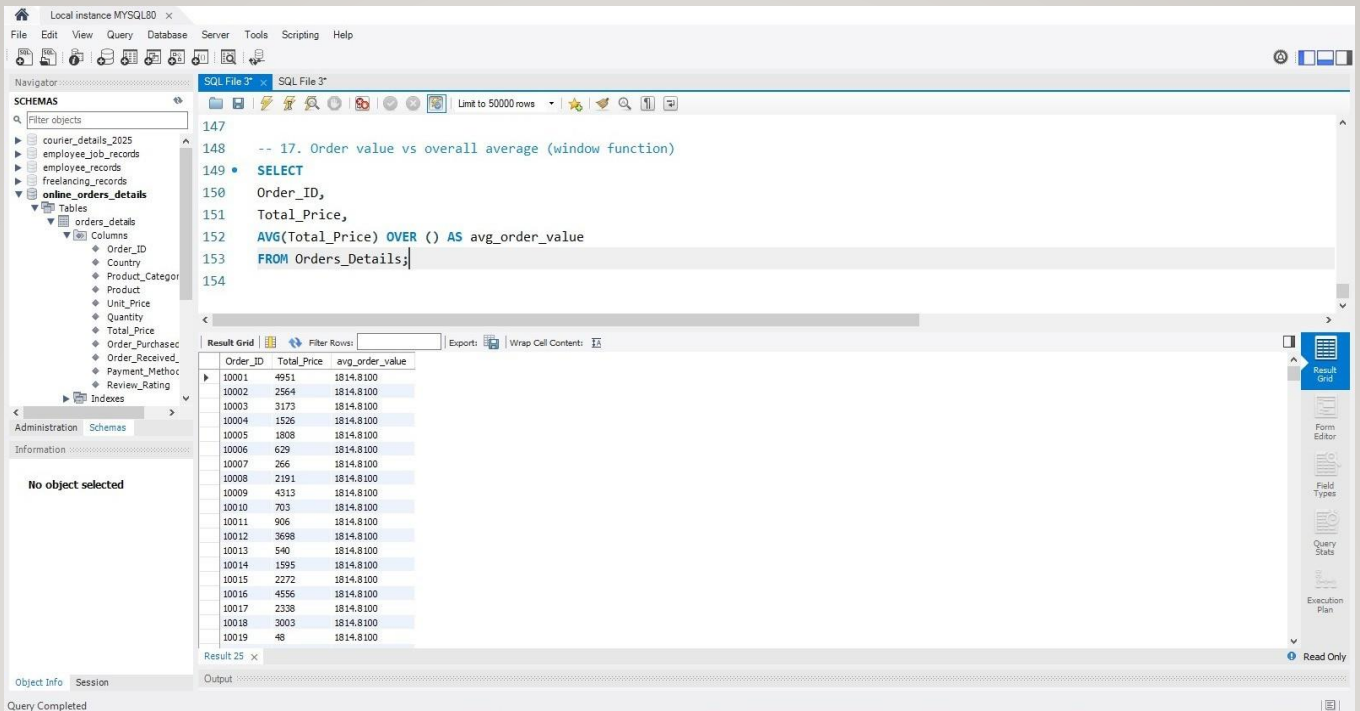
Order Value vs Overall Average:

This query compares each order value against the overall average using a window function.

SELECT Order_ID, Total_Price, AVG(Total_Price) OVER () AS avg_order_value
FROM Orders_Details;

For collaboration,feedback or suggestions regarding this project feel free to mail me at shahidabbas2104514@gmail.com