**SQL FINAL PROJECT REPORT**

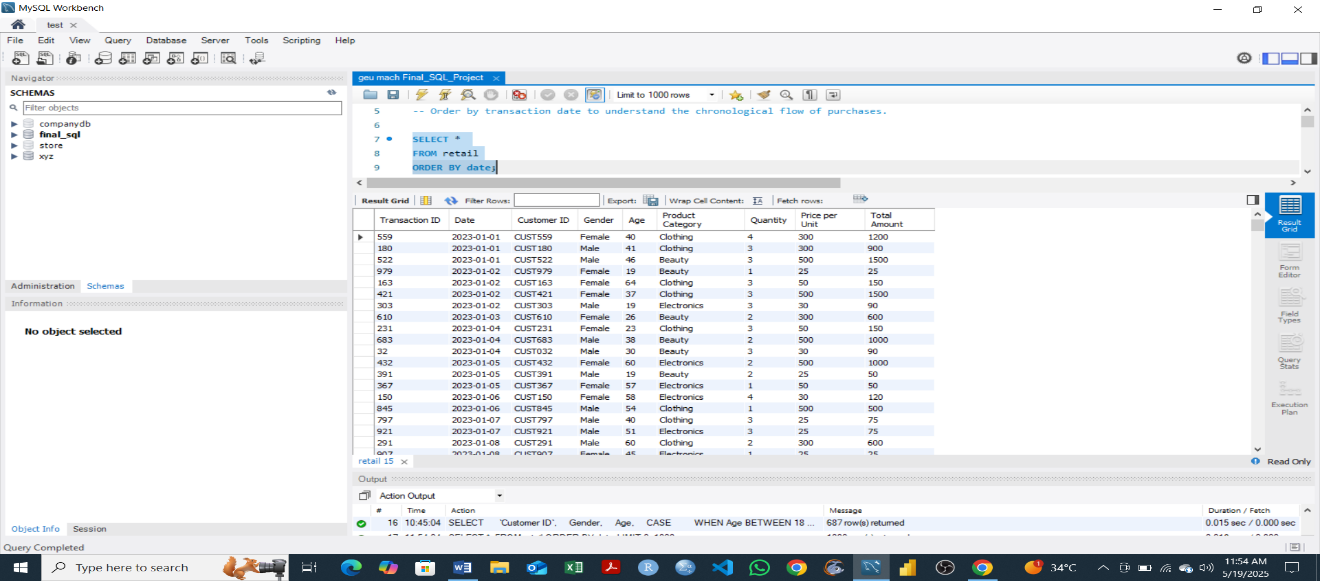
**-- 1. Retrieve all transactions with valid customer and product data.**

**-- Order by transaction date to understand the chronological flow of purchases.**

**SELECT \***

**FROM retail**

**ORDER BY date;**

****

* The output displays all transactions sorted chronologically by transaction date. This helps to track purchasing patterns over time, identify trends and analyze sales.

**-- 2. Clean the dataset by ensuring that numeric fields like Quantity,**

**-- Price per Unit, and Total Amount are properly formatted.**

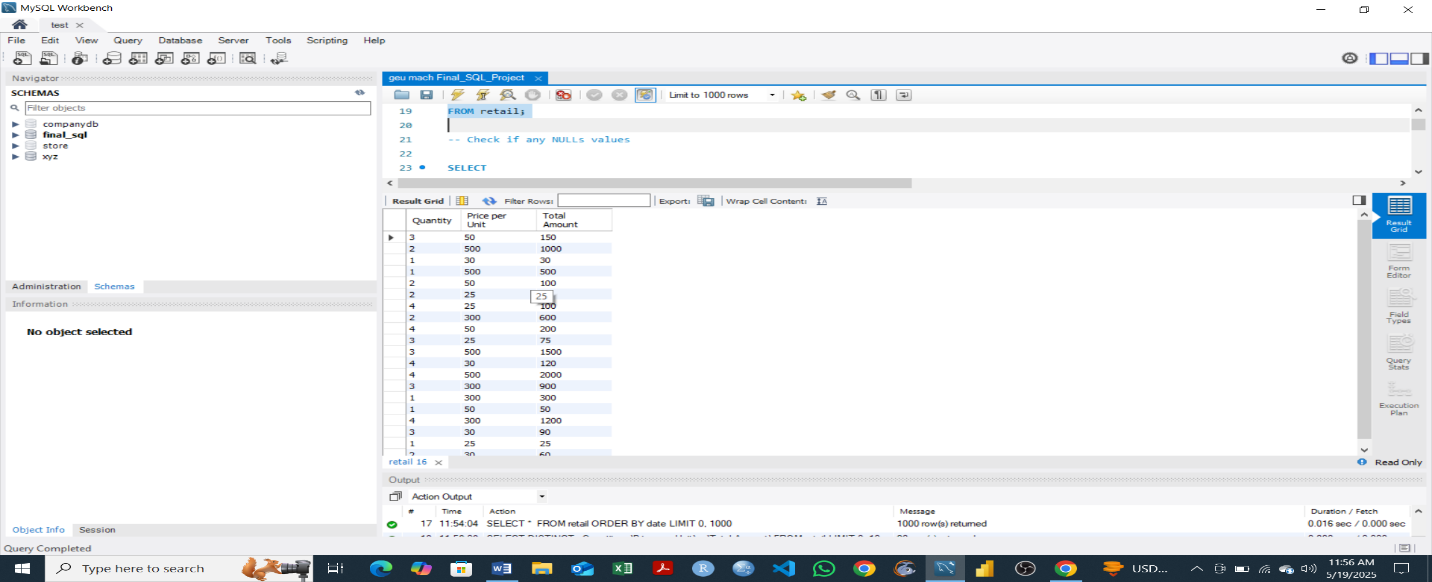
**-- Remove duplicates or null values if any exist.**

**SELECT DISTINCT**

**Quantity,**

**`Price per Unit`,**

**`Total Amount`**

**FROM retail;**

* The query extracts unique combinations of Quantity, Price per Unit, and Total Amount from the retail dataset. This helps identify and remove duplicate entries while ensuring numeric fields are properly formatted. It supports data cleaning by highlighting inconsistencies or anomalies

**-- Check if any NULLs values**

**SELECT**

**Quantity,**

**`Price per Unit`,**

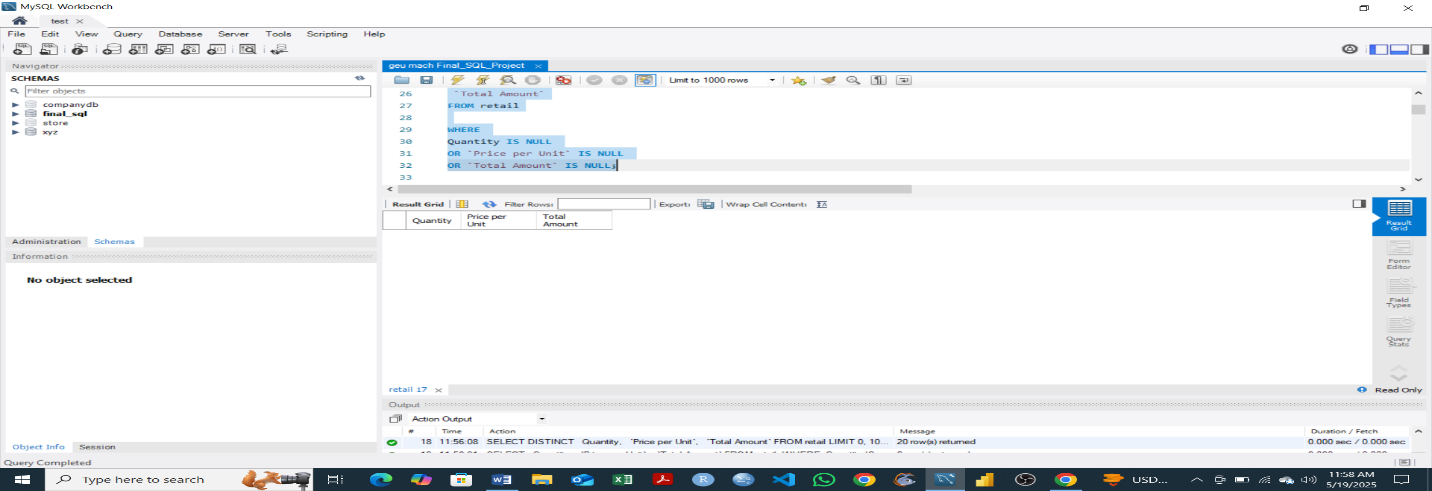
**`Total Amount`**

**FROM retail**

**WHERE Quantity IS NULL**

**OR `Price per Unit` IS NULL**

**OR `Total Amount` IS NULL;**

*  The query aimed to clean the retail dataset by checking for null values. The output confirms that there are no nulls in the specified fields, indicating that the dataset is clean.

**-- 3. Calculate the total and average revenue for each product category.**

**-- Which categories bring in the most and least revenue?**

**SELECT**

**`Product Category`,**

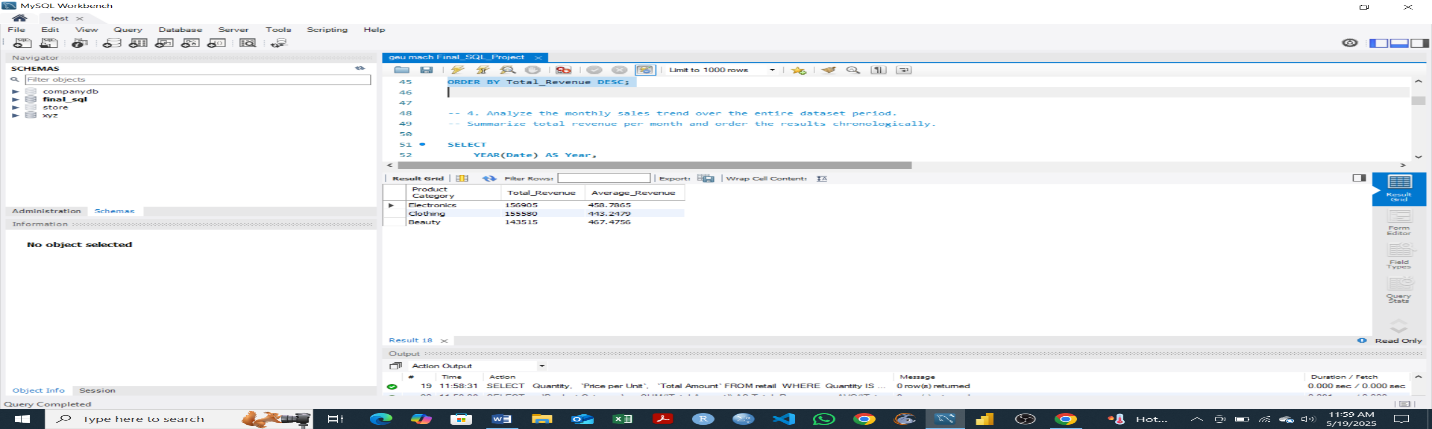
**SUM(`Total Amount`) AS Total\_Revenue,**

**AVG(`Total Amount`) AS Average\_Revenue**

**FROM retail**

**GROUP BY `Product Category`**

**ORDER BY Total\_Revenue DESC;**

****

* The query calculates total and average revenue for each product category in the dataset. The output displays electronics as top product after being grouped and sorted by total revenue from the highest to the lowest income.

**-- 4. Analyze the monthly sales trend over the entire dataset period.**

**-- Summarize total revenue per month and order the results chronologically.**

**SELECT**

**YEAR(Date) AS Year,**

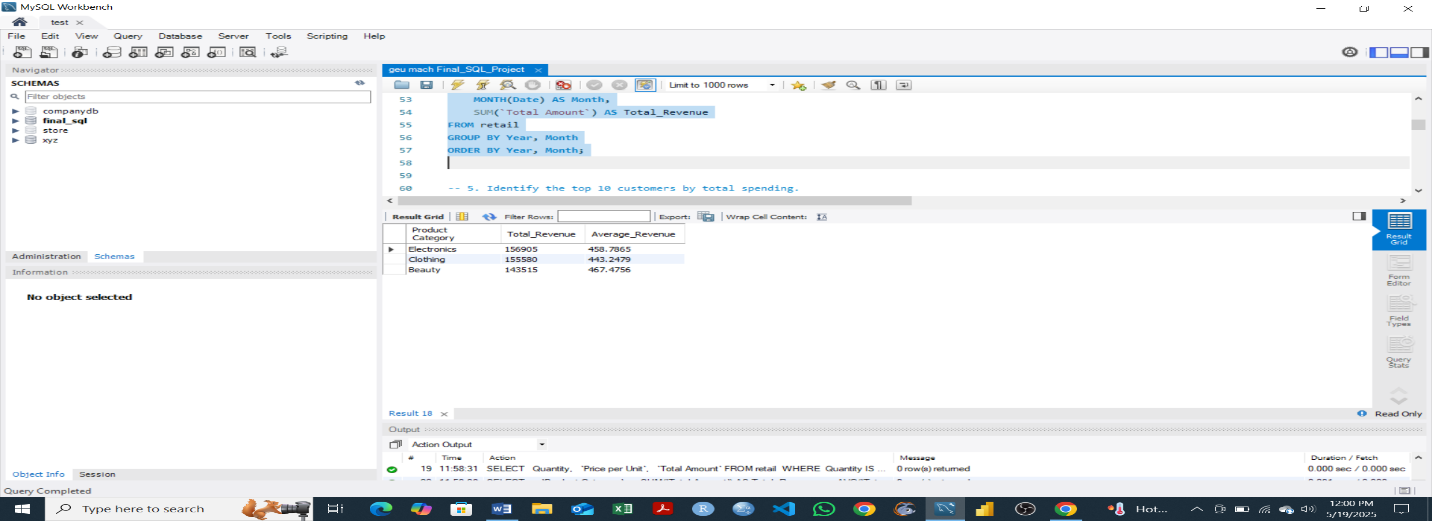
**MONTH(Date) AS Month,**

**SUM(`Total Amount`) AS Total\_Revenue**

**FROM retail**

**GROUP BY Year, Month**

**ORDER BY Year, Month;**

****

* This query analyzes monthly sales trends by summarizing total revenue for each month and year. The output displays that electronics have the highest total revenue and average revenue.

**-- 5. Identify the top 10 customers by total spending.**

**-- Rank customers based on how much they’ve spent across all transactions.**

**SELECT**

**`Customer ID`,**

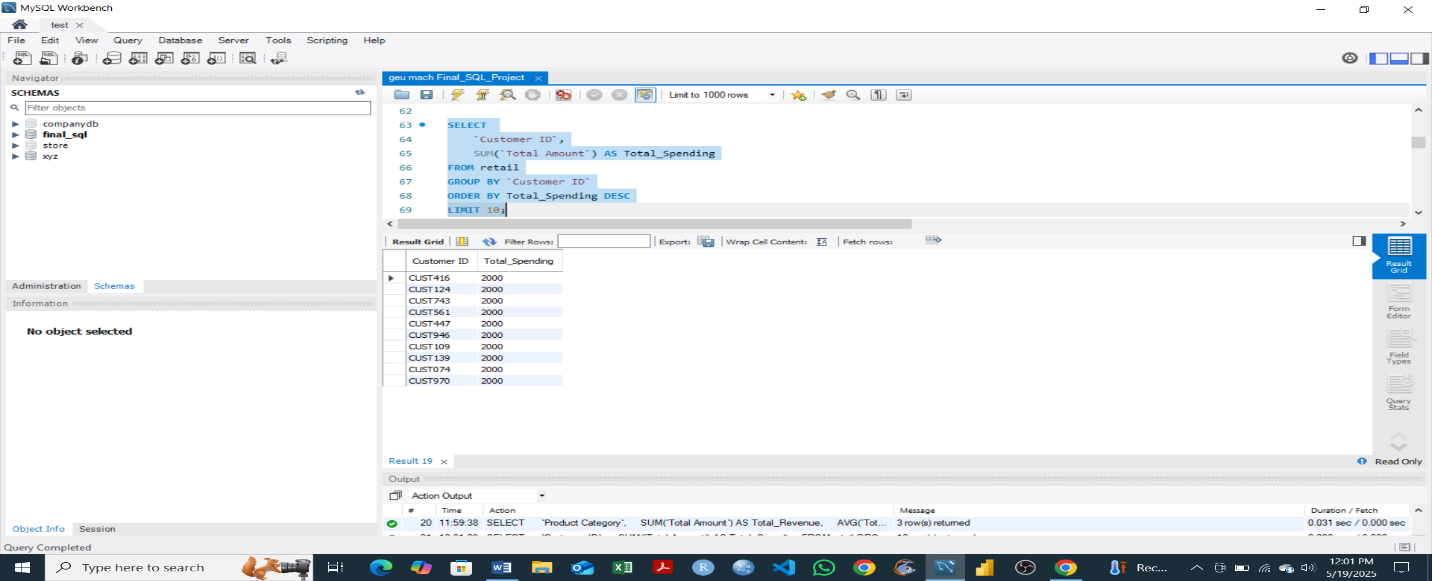
**SUM(`Total Amount`) AS Total\_Spending**

**FROM retail**

**GROUP BY `Customer ID`**

**ORDER BY Total\_Spending DESC**

**LIMIT 10;**

****

* The query identifies the top 10 highest-spending customers by calculating the total amount each has spent across all transactions. The output displays that the top 10 customers spend a total revenue 2000 each.

**-- 6. Calculate the average transaction value for each customer.**

**-- How much does each customer spend per transaction on average?**

**SELECT**

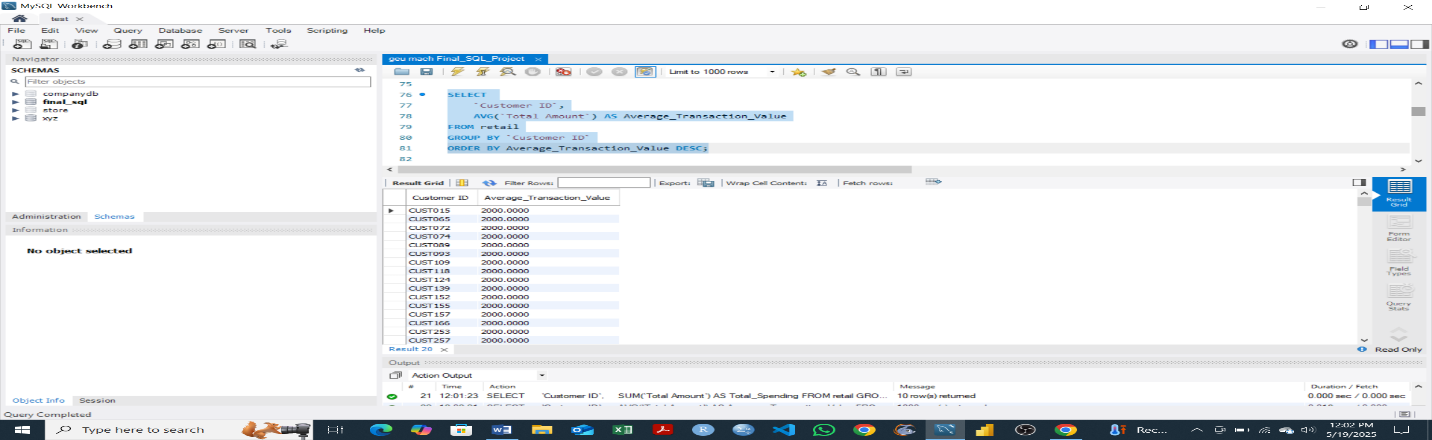
**`Customer ID`,**

**AVG(`Total Amount`) AS Average\_Transaction\_Value**

**FROM retail**

**GROUP BY `Customer ID`**

**ORDER BY Average\_Transaction\_Value DESC;**

****

* The query calculates the average transaction value for each customer by averaging the total amount spent per purchase. It ranks customers from highest to lowest average spend.

**-- 7. Group customers by gender and age brackets (e.g., 18–25, 26–35, 36–50, etc.).**

**-- Summarize total revenue and transaction count for each group.**

**SELECT**

**`Customer ID`,**

**Gender,**

**Age,**

**CASE**

**WHEN Age BETWEEN 18 AND 25 THEN '18-25'**

**WHEN Age BETWEEN 26 AND 35 THEN '26-35'**

**WHEN Age BETWEEN 36 AND 50 THEN '36-50'**

**ELSE '51+'**

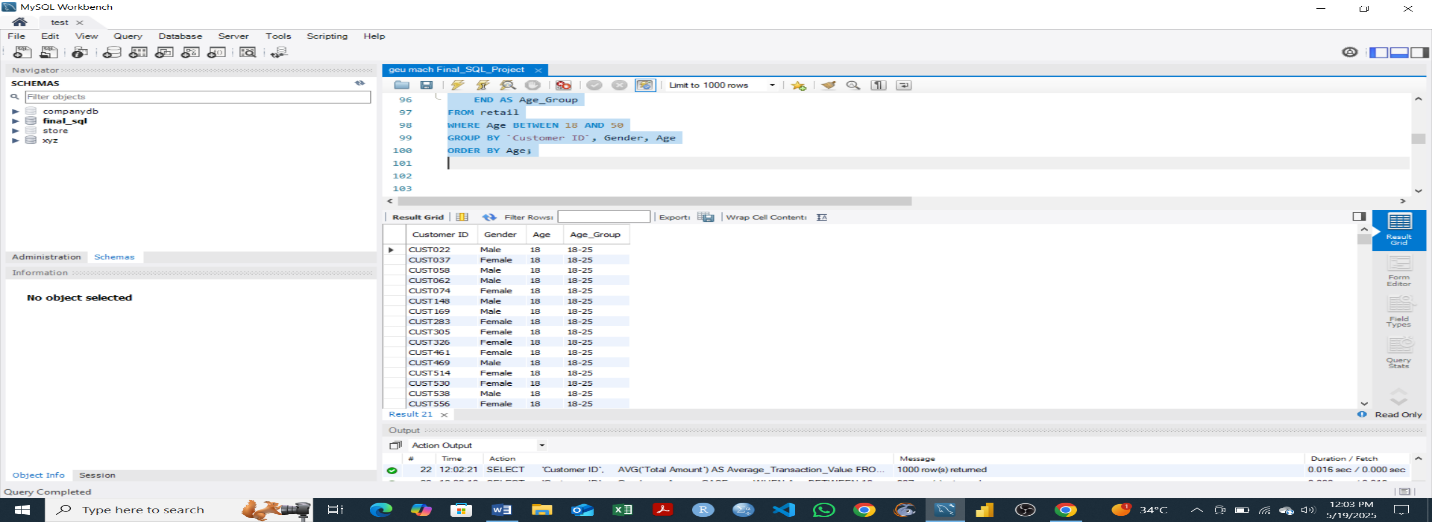
**END AS Age\_Group**

**FROM retail**

**WHERE Age BETWEEN 18 AND 50**

**GROUP BY `Customer ID`, Gender, Age**

**ORDER BY Age;**

****

* This query groups customers by gender and age brackets (18–25, 26–35, 36–50). It shows that customers in the age bracket of 18-25 have high total revenue and transaction count.

**-- 8. Compare the number of one-time buyers versus repeat buyers.**

**-- Group customers by purchase frequency to determine repeat behavior.**

**SELECT**

**Purchase\_Count,**

**COUNT(\*) AS Number\_of\_Customers**

**FROM (**

**SELECT**

**`Customer ID`,**

**COUNT(\*) AS Purchase\_Count**

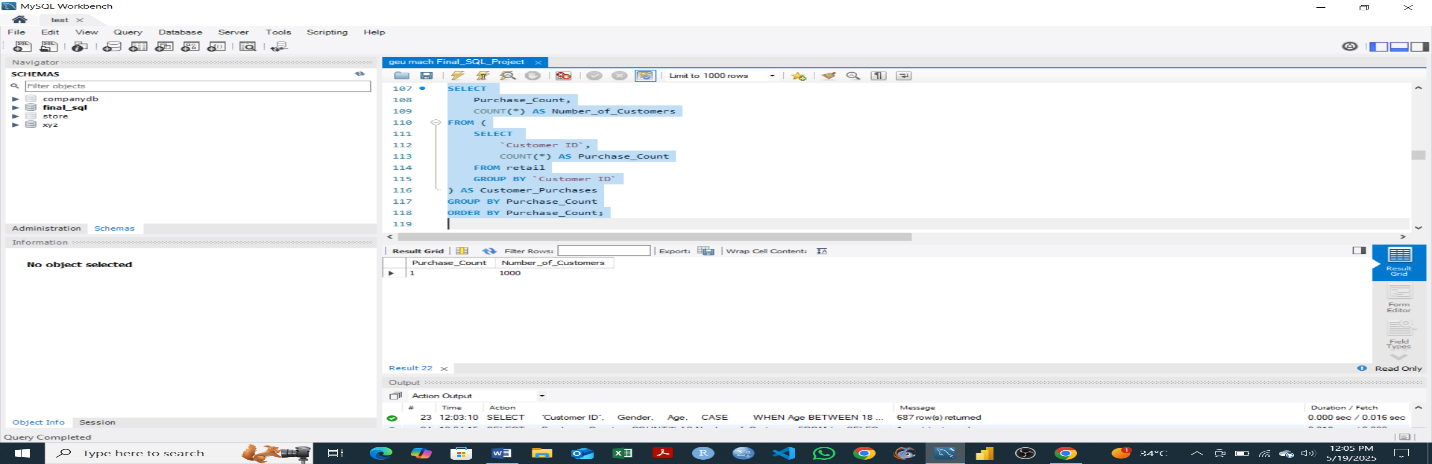
**FROM retail**

**GROUP BY `Customer ID`**

**) AS Customer\_Purchases**

**GROUP BY Purchase\_Count**

**ORDER BY Purchase\_Count;**

****

* This query analyzes customer purchase frequency by grouping them based on how many times they've made a transaction. The output shows that all the customers did a transaction once.

**-- 9. Identify inactive customers who have not made a purchase in the last 6 months.**

**-- Use the most recent date in the dataset as the reference point.**

**SELECT**

**`Customer ID`,**

**MAX(Date) AS Last\_Purchase\_Date**

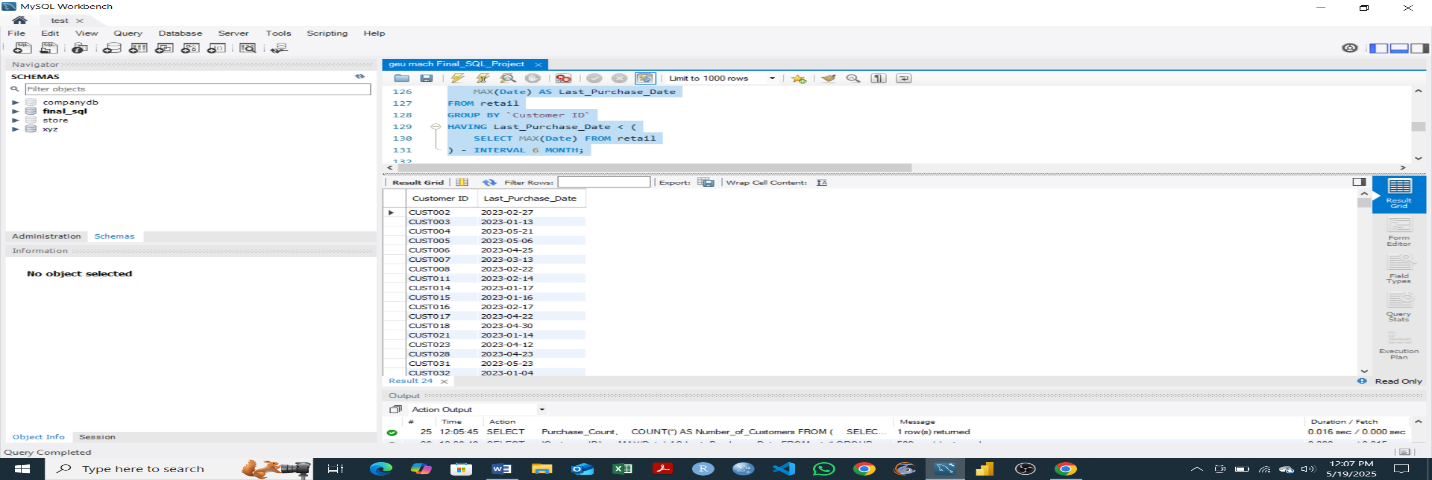
**FROM retail**

**GROUP BY `Customer ID`**

**HAVING Last\_Purchase\_Date < (**

**SELECT MAX(Date) FROM retail**

**) - INTERVAL 6 MONTH;**

****

* The query identifies inactive customers by selecting those who have not made a purchase in the past 6 months, using the most recent transaction date as a reference.

**-- 10. Perform RFM (Recency, Frequency, Monetary) analysis for customer segmentation.**

**-- Recency: Days since last purchase; Frequency: Number of purchases; Monetary: Total amount spent.**

**-- Step 1: Calculate RFM metrics for each customer**

**SELECT**

**`Customer ID`,**

**-- Recency: Days since last purchase**

**DATEDIFF(**

**(SELECT MAX(Date) FROM retail),**

**MAX(Date)**

**) AS Recency,**

**-- Frequency: Number of purchases**

**COUNT(\*) AS Frequency,**

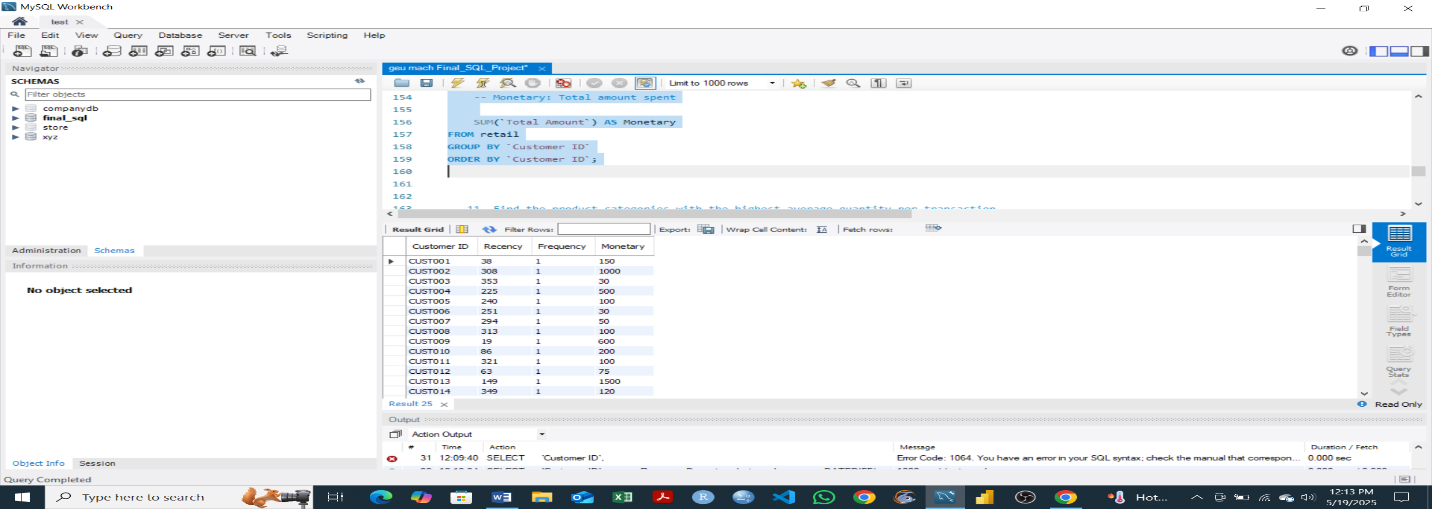
**-- Monetary: Total amount spent**

**SUM(`Total Amount`) AS Monetary**

**FROM retail**

**GROUP BY `Customer ID`**

**ORDER BY `Customer ID`;**

****

* This query performs an RFM (Recency, Frequency, Monetary) analysis for each customer. It calculates how recently a customer made a purchase, how often they buy, and how much they spend.

**-- 11. Find the product categories with the highest average quantity per transaction.**

**-- Which product types are purchased in bulk?**

**SELECT**

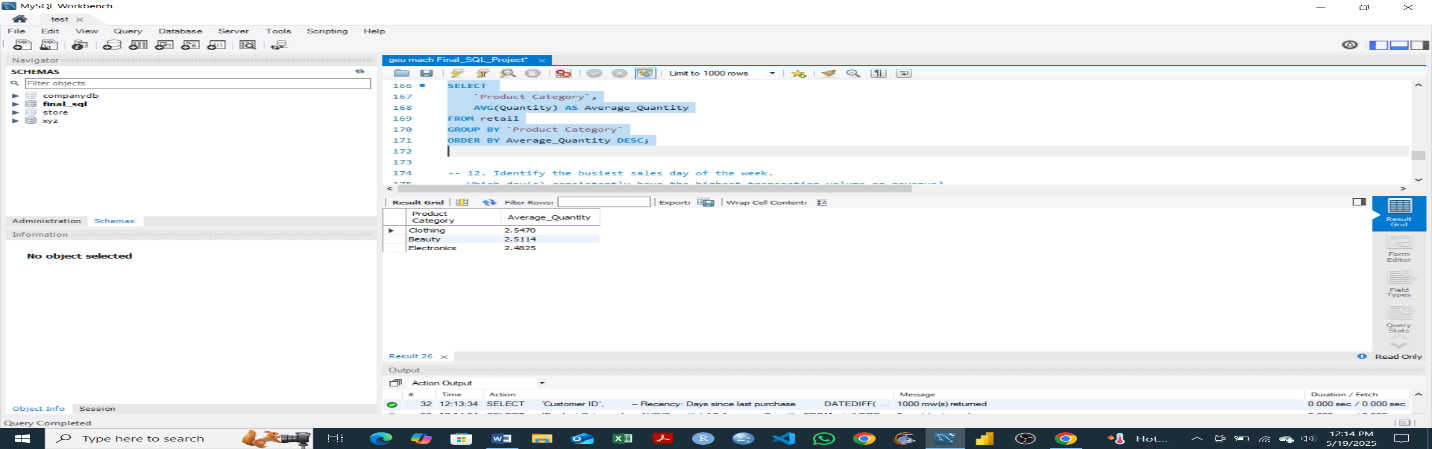
**`Product Category`,**

**AVG(Quantity) AS Average\_Quantity**

**FROM retail**

**GROUP BY `Product Category`**

**ORDER BY Average\_Quantity DESC;**

****

* The output shows product categories with the highest average quantity per transaction, revealing items that were purchased in bulk.

**-- 12. Identify the busiest sales day of the week.**

**-- Which day(s) consistently have the highest transaction volume or revenue?**

**SELECT**

**DAYNAME(Date) AS Day\_of\_Week,**

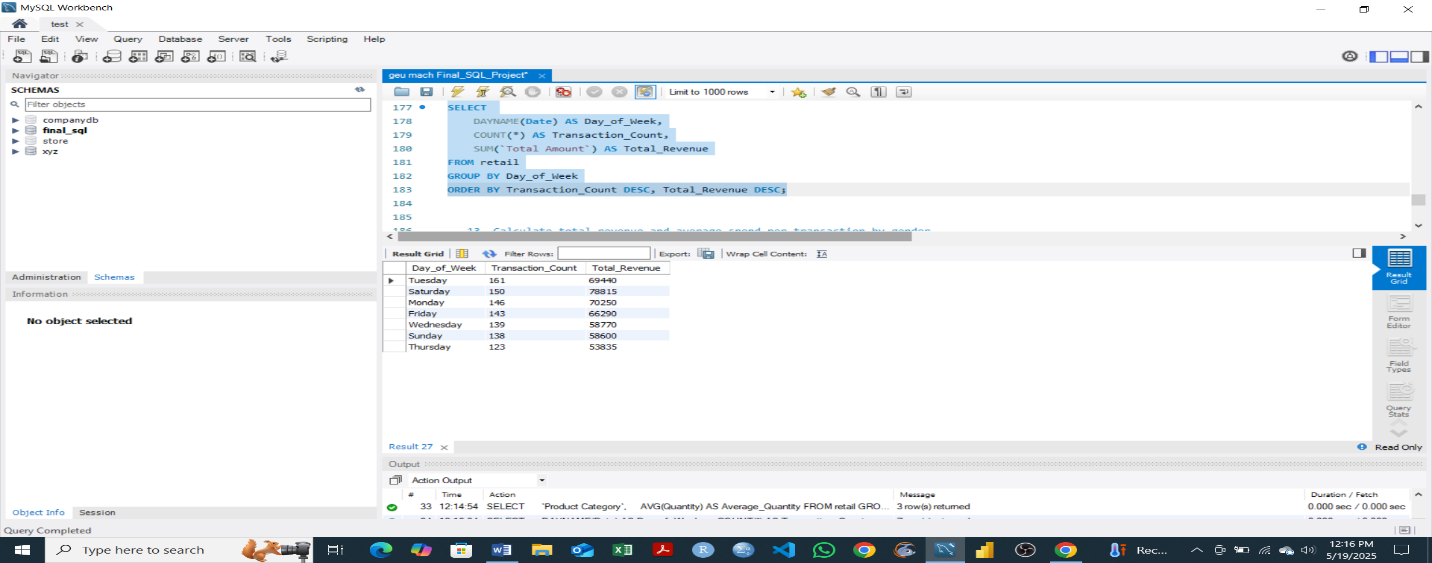
**COUNT(\*) AS Transaction\_Count,**

**SUM(`Total Amount`) AS Total\_Revenue**

**FROM retail**

**GROUP BY Day\_of\_Week**

**ORDER BY Transaction\_Count DESC, Total\_Revenue DESC;**

****

* The output displays days of the week with the highest number of transactions in a chronological order from busiest day to the day with least number of transaction. It shows that Tuesday is the busiest day of transactions.

**-- 13. Calculate total revenue and average spend per transaction by gender.**

**-- Are there differences in spending patterns across genders?**

**SELECT**

**Gender,**

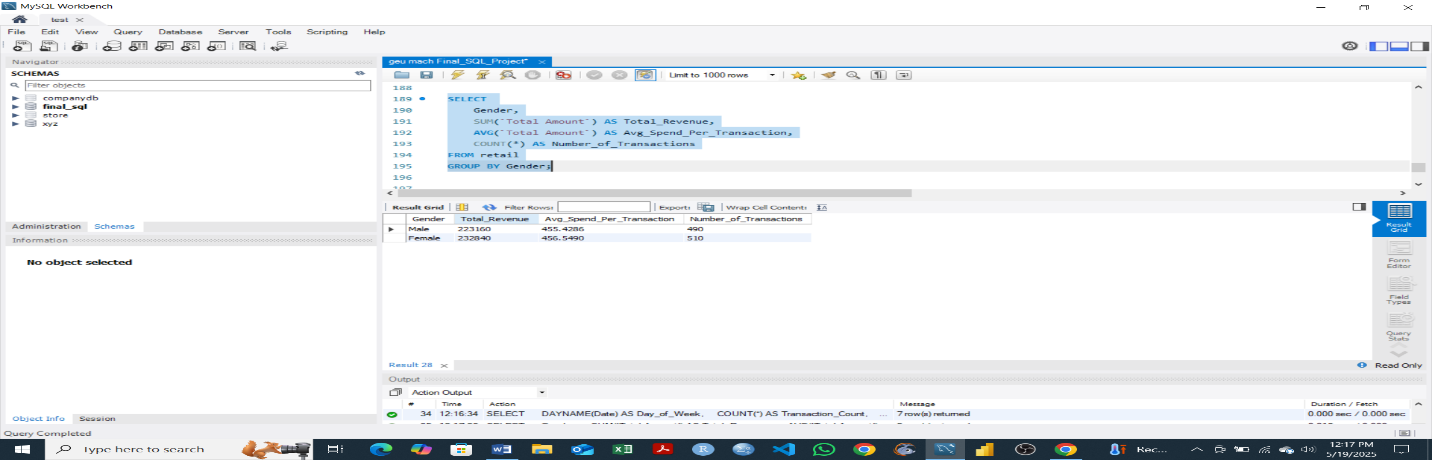
**SUM(`Total Amount`) AS Total\_Revenue,**

**AVG(`Total Amount`) AS Avg\_Spend\_Per\_Transaction,**

**COUNT(\*) AS Number\_of\_Transactions**

**FROM retail**

**GROUP BY Gender;**

****

* The output shows total revenue, average spend per transaction, and number of transactions by gender from the retail dataset. Therefore, indicating that the male gender has the highest total revenue, average spend per transaction, and number of transactions.

**-- 14. Find the top 5 most frequently purchased product categories.**

**-- Based on number of transactions involving each category.**

**SELECT**

**`Product Category`,**

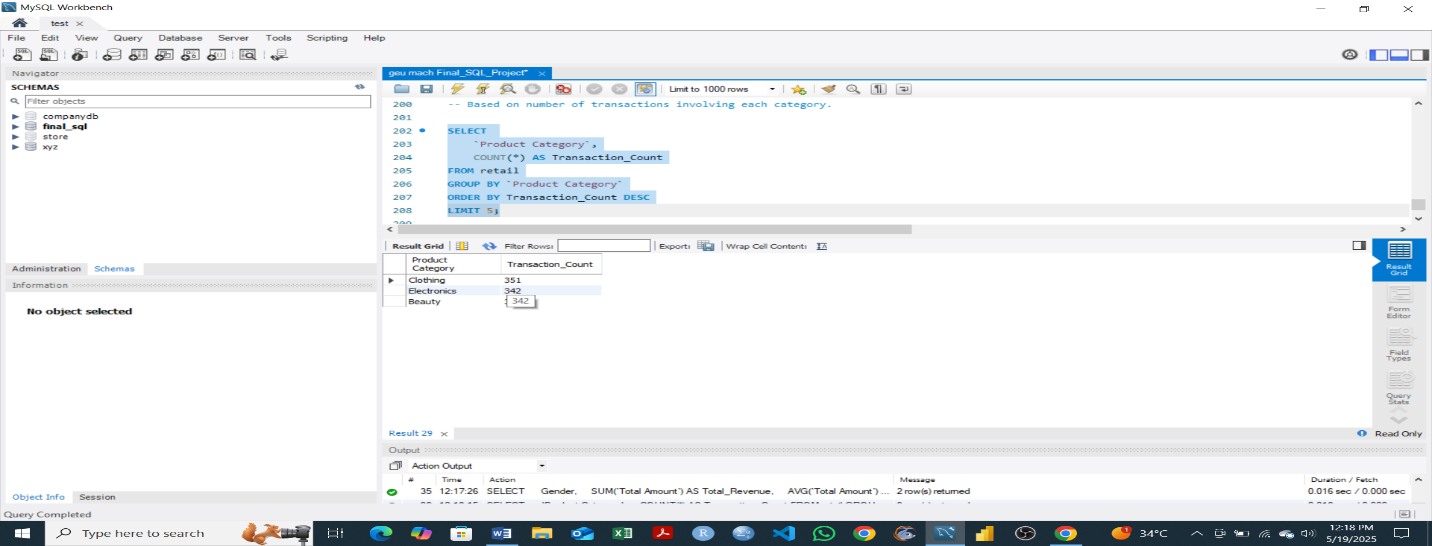
**COUNT(\*) AS Transaction\_Count**

**FROM retail**

**GROUP BY `Product Category`**

**ORDER BY Transaction\_Count DESC**

**LIMIT 5;**

****

* This output displays the top 5 product categories with the highest number of transactions in the retail dataset. This was achieved using the filtering and limit SQL functions.

**-- 15. Determine the percentage of total revenue contributed by each age group.**

**-- Which customer age brackets are most valuable to the business?**

**-- Step 1: Categorize customers into age brackets and calculate revenue per group**

**SELECT**

**CASE**

**WHEN Age BETWEEN 18 AND 25 THEN '18-25'**

**WHEN Age BETWEEN 26 AND 35 THEN '26-35'**

**WHEN Age BETWEEN 36 AND 50 THEN '36-50'**

**WHEN Age > 50 THEN '51+'**

**ELSE 'Unknown'**

**END AS Age\_Group,**

**SUM(`Total Amount`) AS Group\_Revenue,**

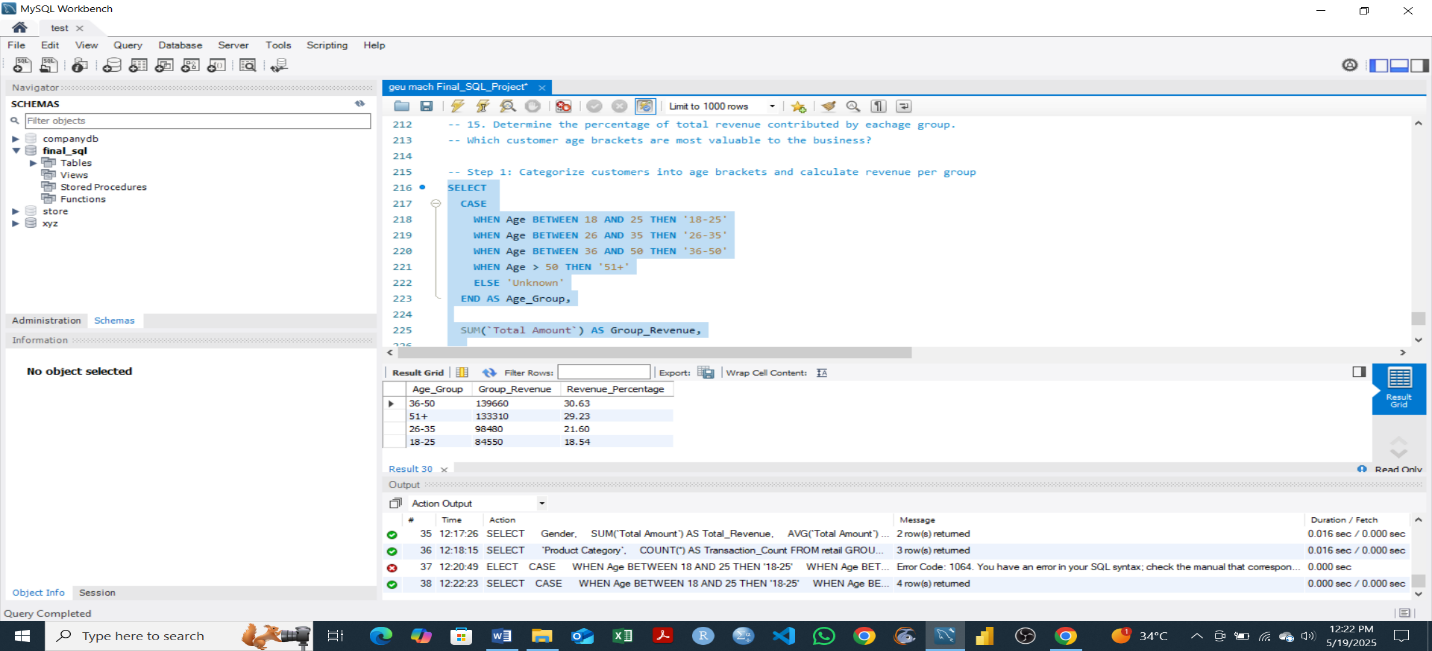
**ROUND(SUM(`Total Amount`) \* 100.0 /**

**(SELECT SUM(`Total Amount`) FROM retail), 2) AS Revenue\_Percentage**

**FROM retail**

**GROUP BY Age\_Group**

**ORDER BY Revenue\_Percentage DESC;**

****

* The output shows customers categorized into age brackets and each age group’s contribution to total revenue. It shows that customers in the age bracket of 35-50 years have the highest revenue percentage.