|  |  |
| --- | --- |
|  |  |
|  |
|  |

**High Impact Skills Development Program for Gilgit Baltistan**

**Data Mining Module Project**

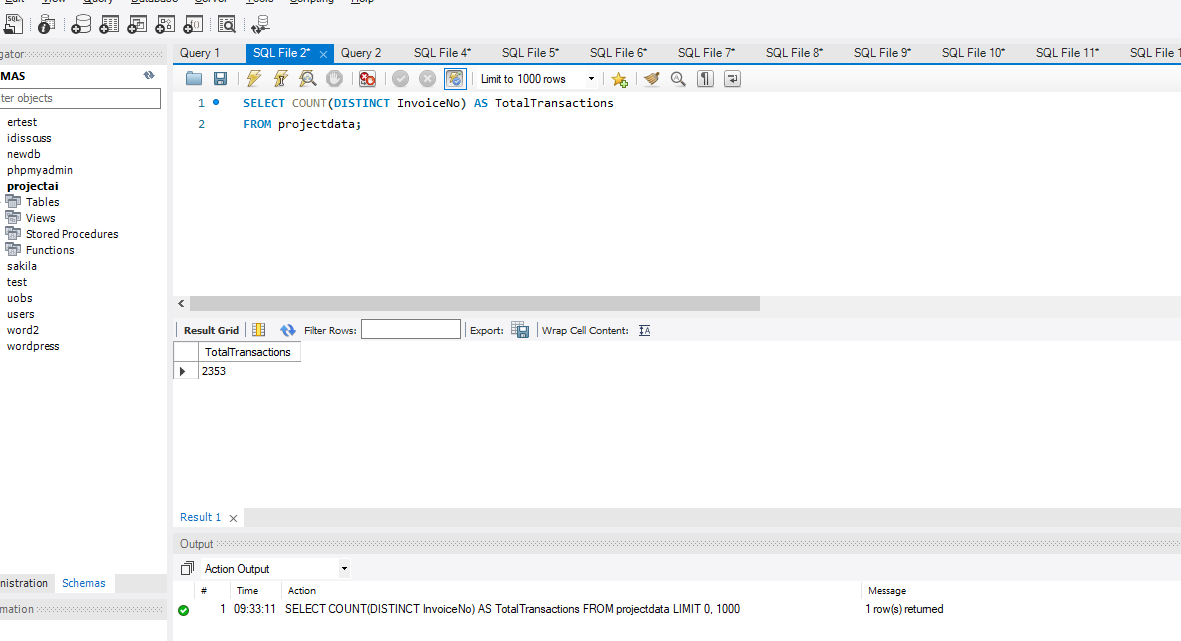
**Project Title:** Online Retail Segmentation.

**Learning Objectives:**

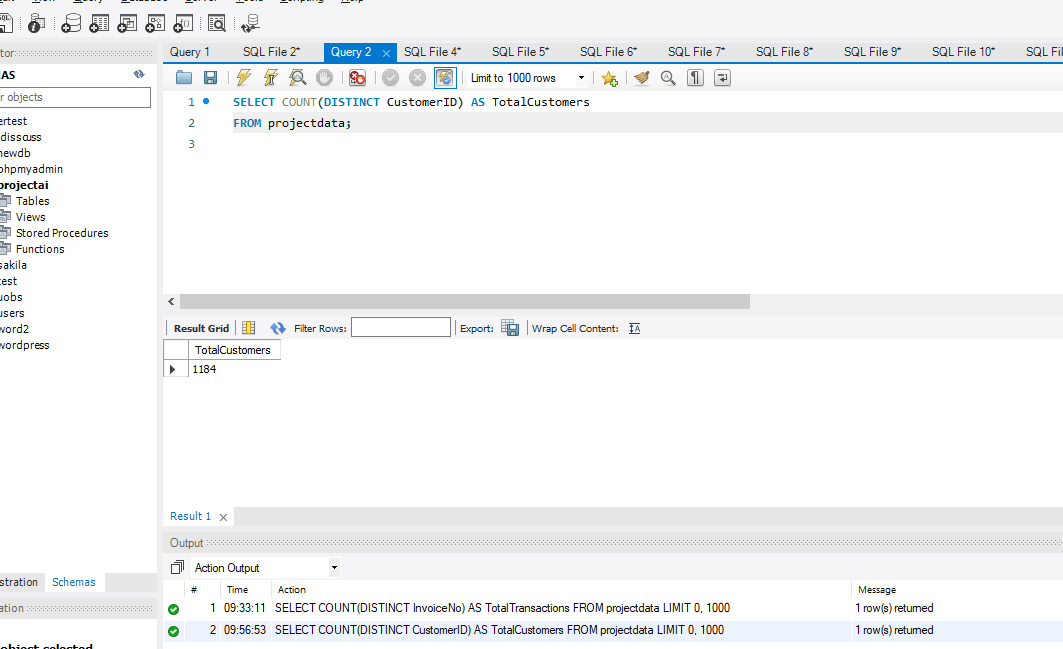
1. Understanding the fundamentals of Data Mining

2. Learn how to use SQL in data mining

3. Learn how to implement mining concepts in MySQL Workbench

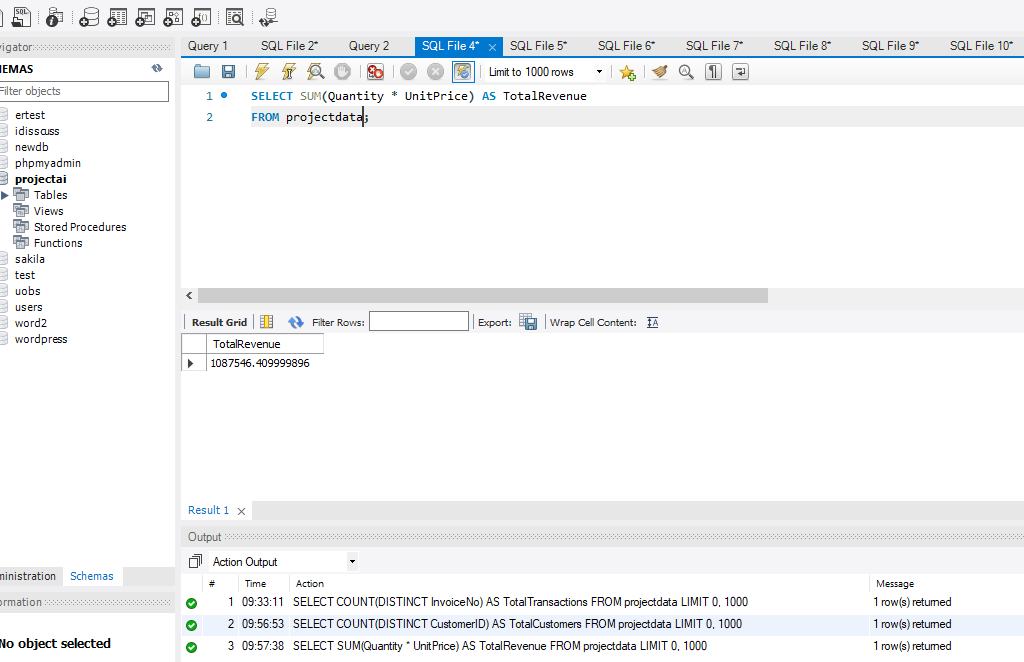


**1. Total Number of Transactions:**

**Description**: This query determines the overall count of unique transactions (invoices) contained within the dataset, giving an insight into the total volume of business interactions.

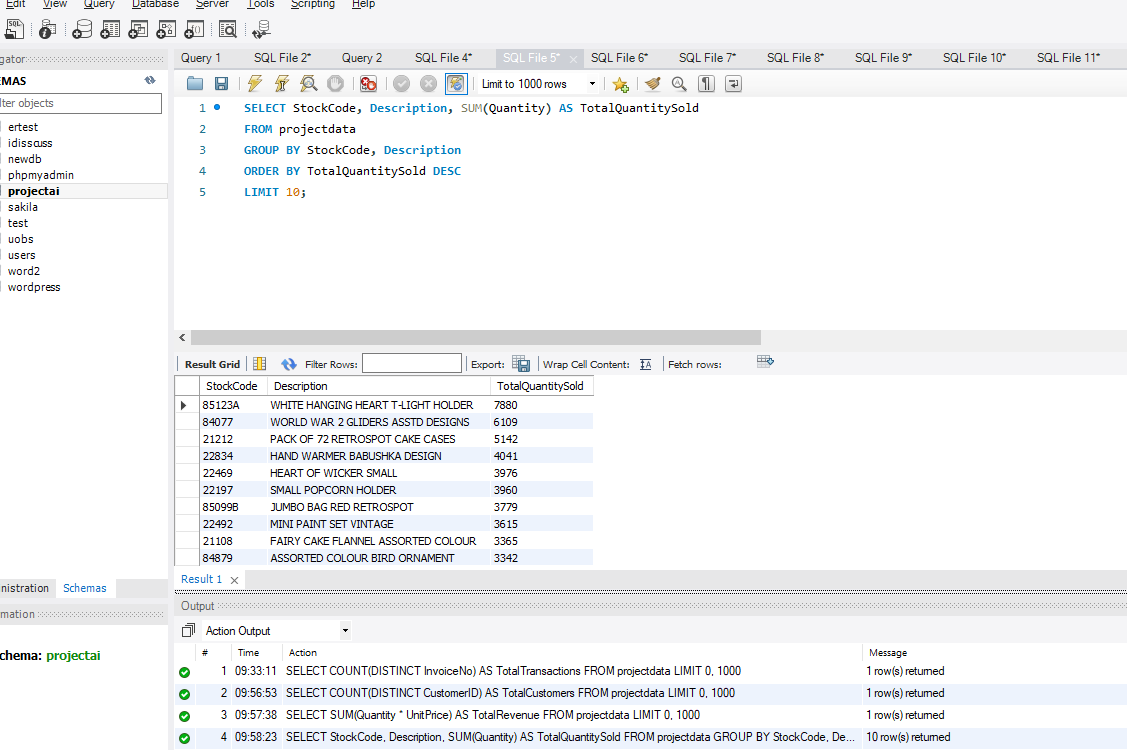
**2. Total Number of Customers:**

**Description**: This query calculates the count of distinct customers represented in the dataset, providing an understanding of the total customer base.



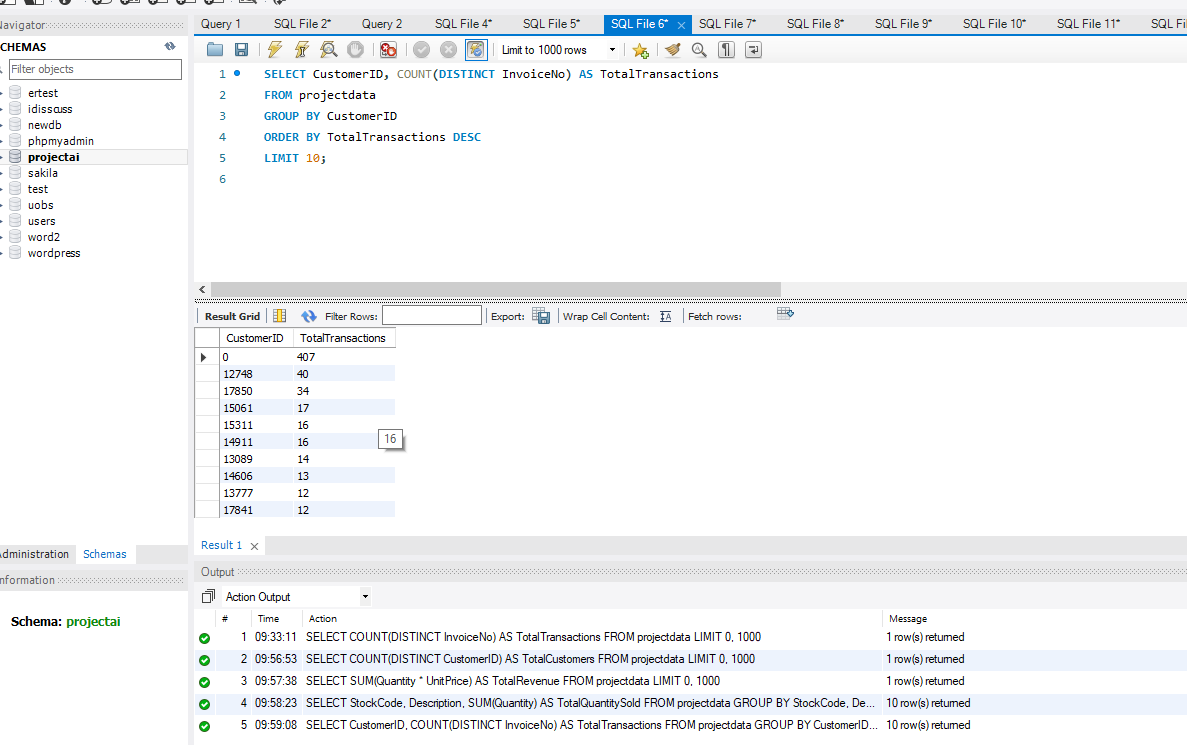
**3. Total Revenue:**

**Description**: This query computes the cumulative revenue generated from all transactions by multiplying the quantity of each product sold by its corresponding unit price, giving a holistic view of financial performance.



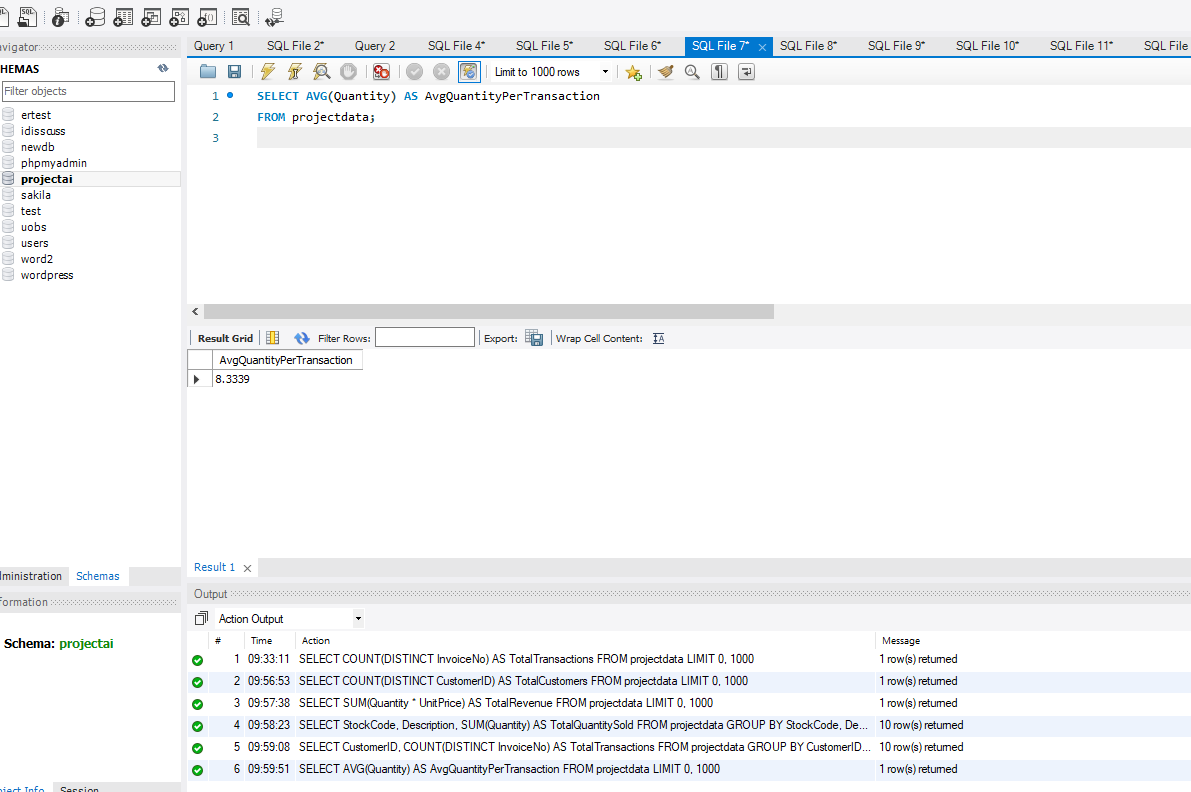
**4. Average Order Value:**

**Description**: This query calculates the mean value of each order by finding the average product of quantity and unit price across all transactions, aiding in assessing typical purchase sizes.



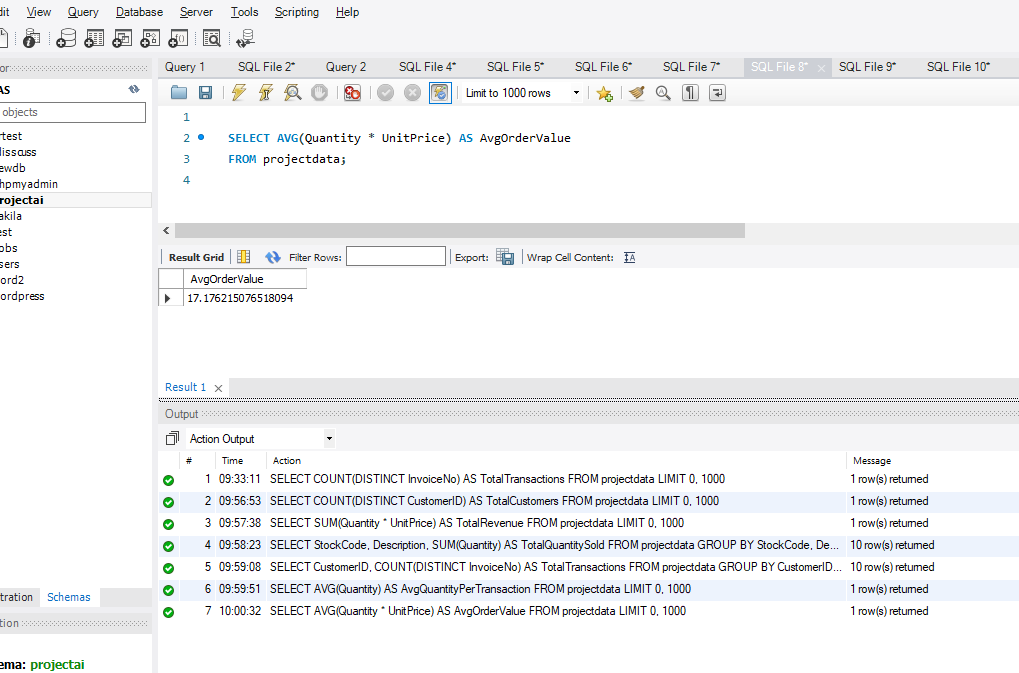
**5. Top 5 Selling Products:**

Description: This query identifies and lists the five most frequently purchased products based on the aggregate quantity of each item sold, revealing popular items.



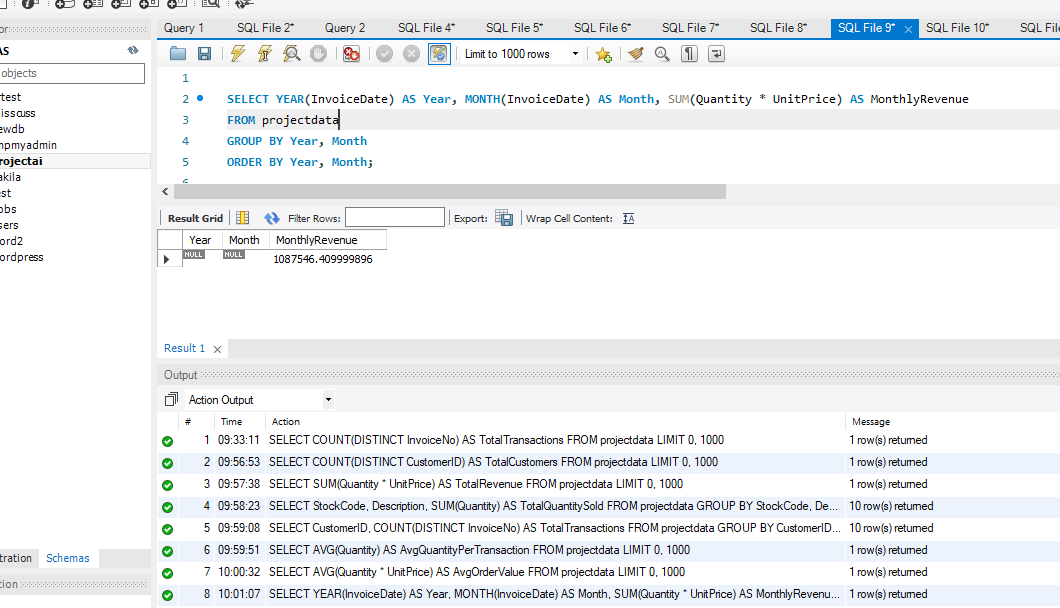
**6. Most Active Customers (Based on Transactions):**

Description: This query highlights the five customers with the highest transaction counts, showcasing those who have engaged in the most purchase activities.



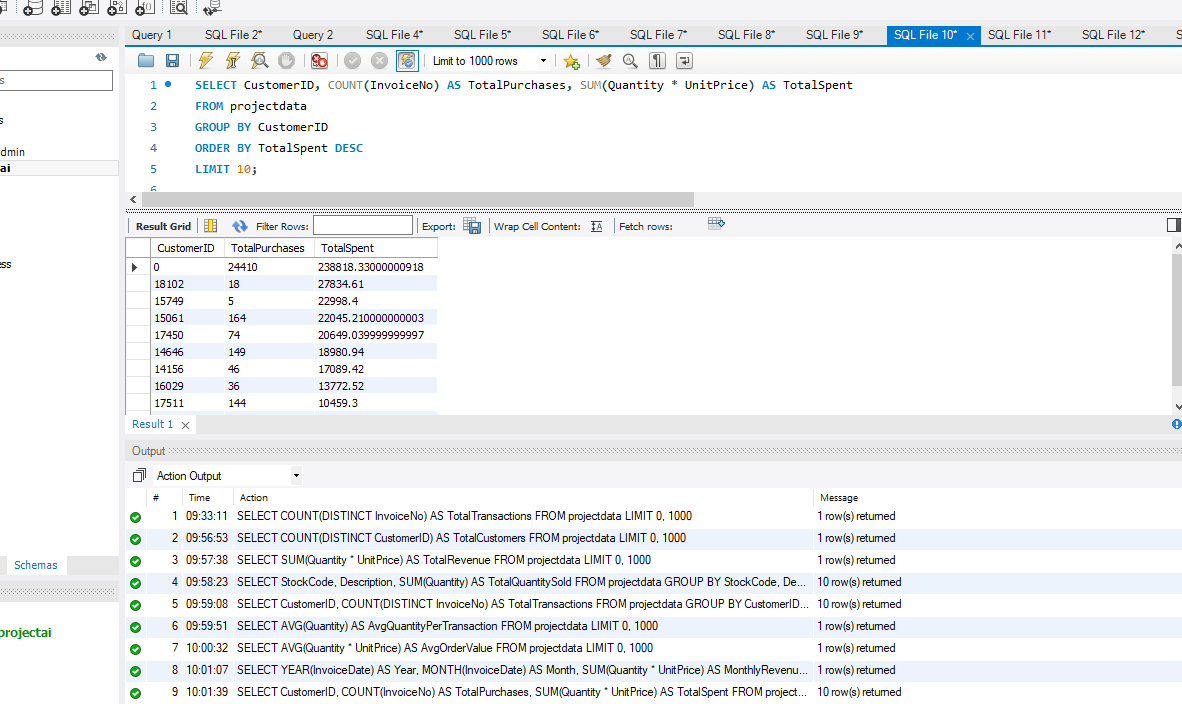
**7. Total Revenue by Country:**

Description: This query aggregates the revenue generated from transactions in each country, delivering insights into the distribution of earnings across different geographic regions.



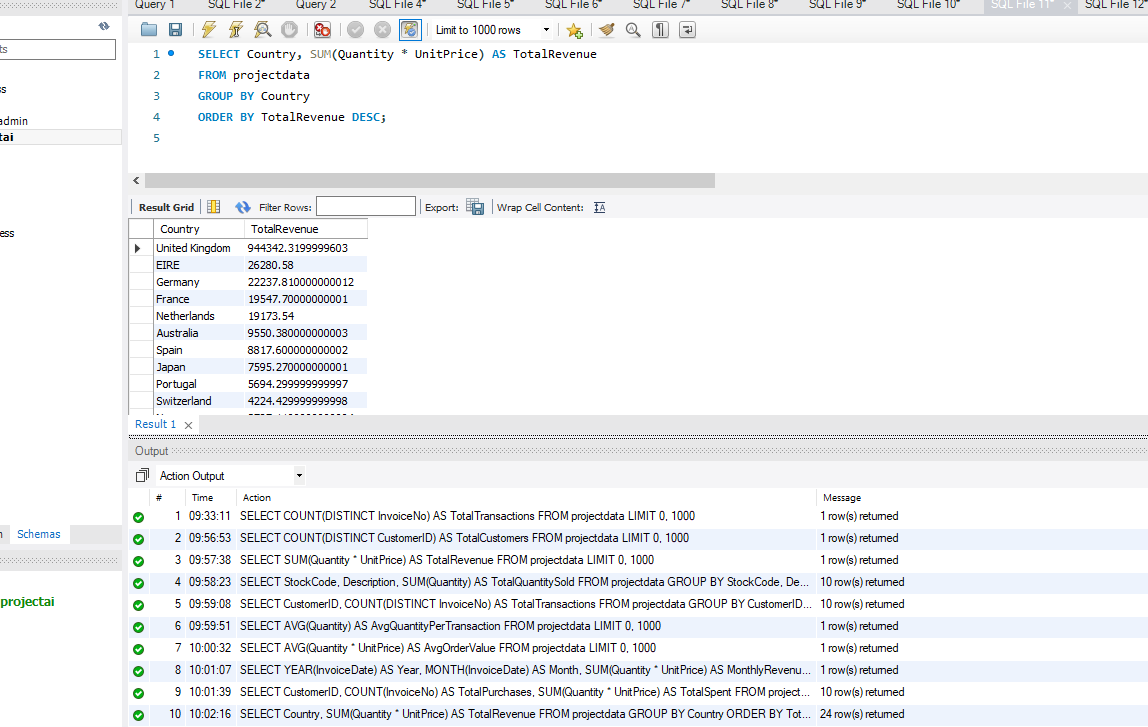
**8. Monthly Revenue for a Specific Year:**

Description: This query analyzes the revenue earned each month for a designated year by summing up revenue (quantity \* unit price) for each month, facilitating a temporal understanding of performance.



**9. Customer Purchasing Patterns:**

Description: This query presents a comprehensive overview of customer behavior by calculating the total transaction count, average item quantity per transaction, and average price per transaction for each customer.



**10. Customer with Highest Total Spending:**

Description: This query identifies the customer who holds the highest aggregate spending across all transactions, pinpointing significant contributors to revenue. These descriptions aim to provide a clear and concise understanding of the insights that each SQL query seeks to uncover from the dataset.

**DEFINE META DATA IN MYSQL WORKBENCH**

In MySQL Workbench, "metadata" refers to information about the structure and properties of database objects, such as tables, columns, indexes, and relationships. Metadata provides essential details that help you understand and manage your database schema and its components. MySQL Workbench is a graphical tool that allows you to interact with your MySQL databases, and it provides various ways to access and view metadata.

**1. Schema Navigator:**

The Schema Navigator panel in MySQL Workbench displays a hierarchical view of your database's metadata. You can expand and collapse objects like schemas, tables, columns, indexes, and more to see their properties and relationships.

**2. Object Information:**

When you select a specific database object (e.g., a table) in the Schema Navigator or the SQL Editor, the Object Information panel provides details about that object, including its columns, indexes, and foreign keys.

**3. SQL Editor and Autocompletion:**

As you write SQL queries or statements in the SQL Editor, MySQL Workbench offers autocompletion suggestions that are based on the metadata of your database objects. This can help you write accurate queries and references.

**4. Table Designer:**

When you design or modify a table using the Table Designer feature, MySQL Workbench uses metadata to show you the current structure of the table. You can add or modify columns, indexes, and relationships through the user interface.

**5. Reverse Engineering:**

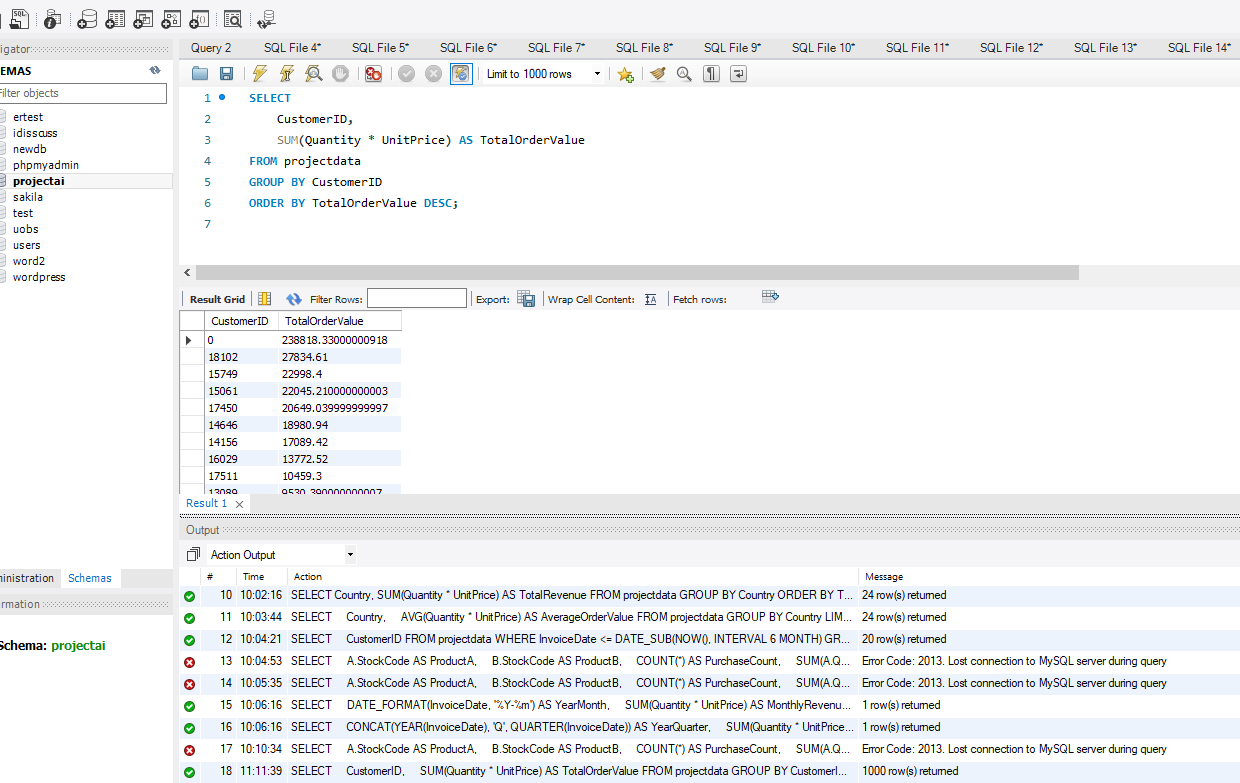
You can use the "Reverse Engineer" feature in MySQL Workbench to create a visual representation of your existing database structure based on its metadata. This is helpful for understanding complex schemas.

**6. Data Export and Import:**

When exporting or importing data, MySQL Workbench relies on metadata to map columns and tables between source and target databases.

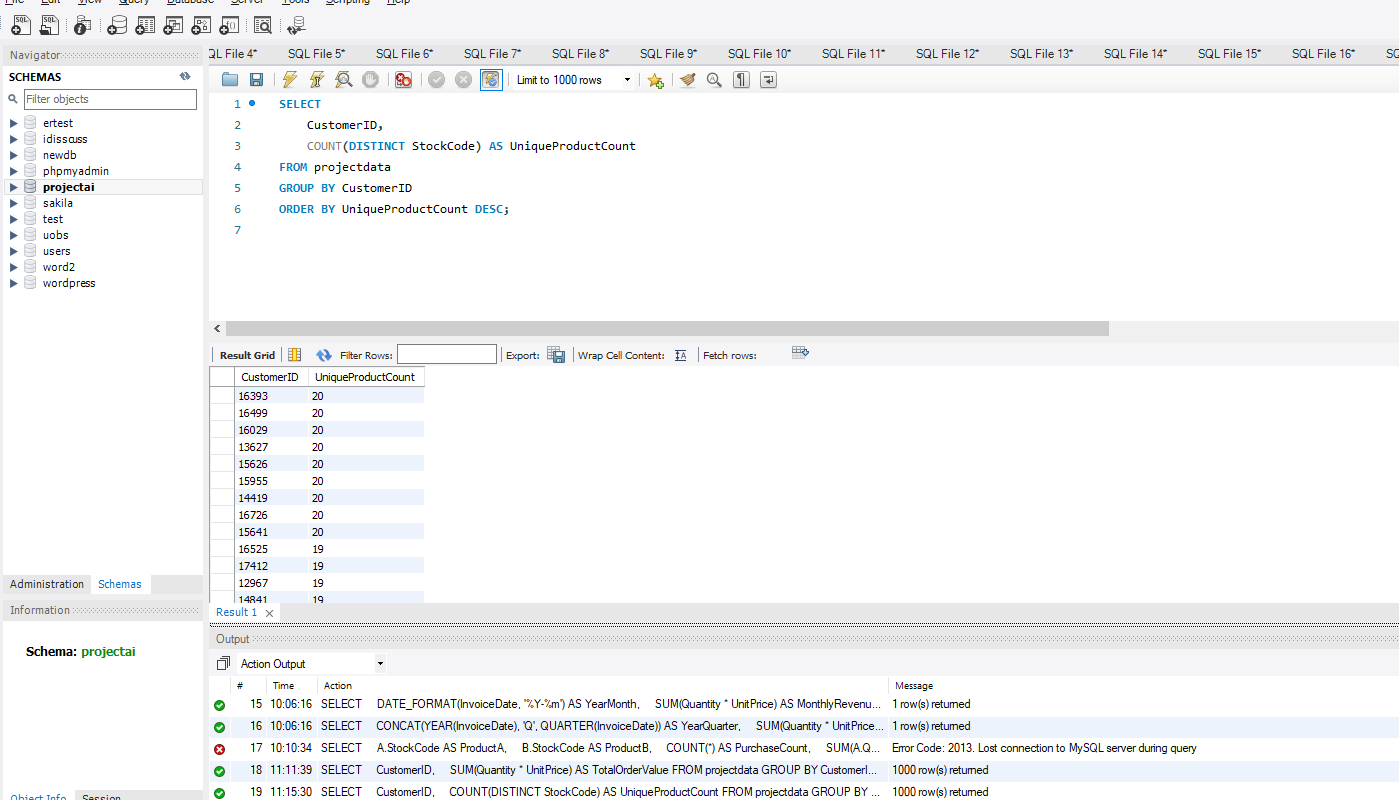
**7. Database Documentation:**

MySQL Workbench allows you to generate documentation for your database schema, including metadata details such as tables, columns, indexes, and relationships.



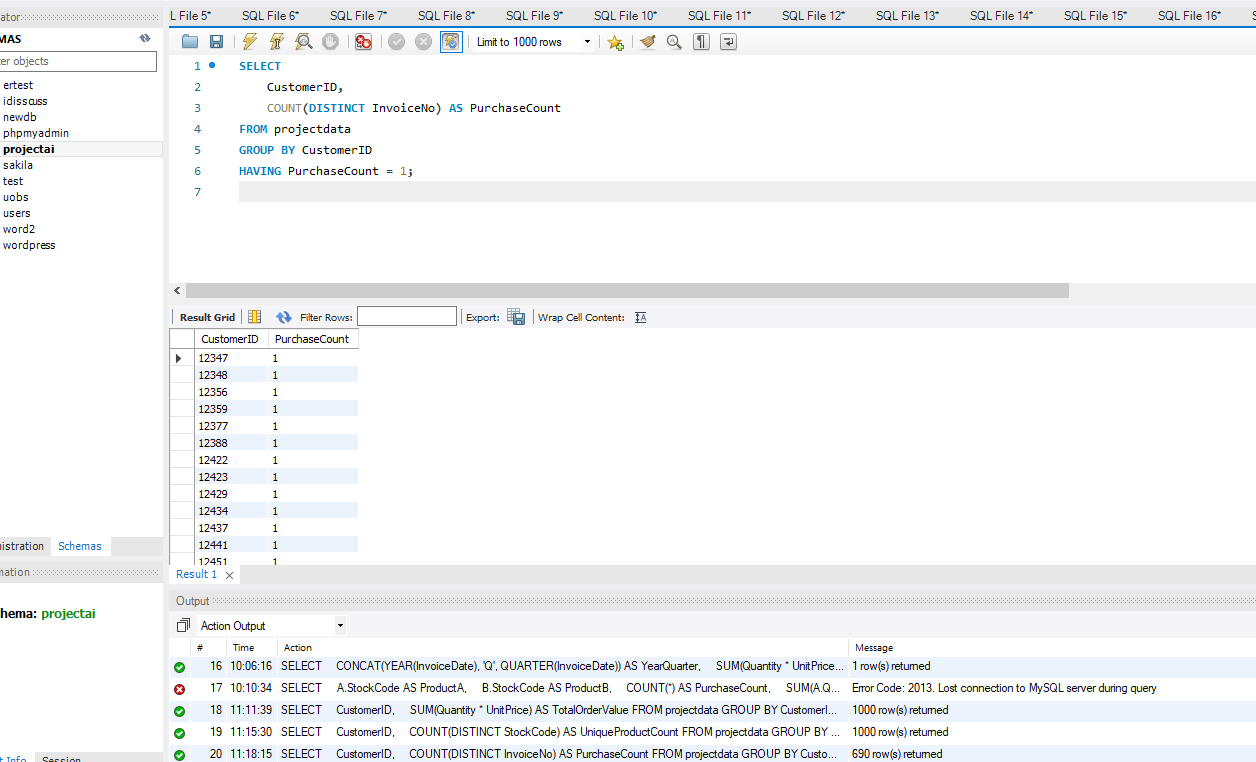
**Description (Distribution of Order Values):**

This SQL query calculates the total order value for each customer in the projectdata table by summing up the product of Quantity and UnitPrice for all transactions associated with each customer. The result set displays customers along with their corresponding total order values, ordered in descending order. This analysis provides insights into the distribution of order values across different customers, helping to identify high-value customers.



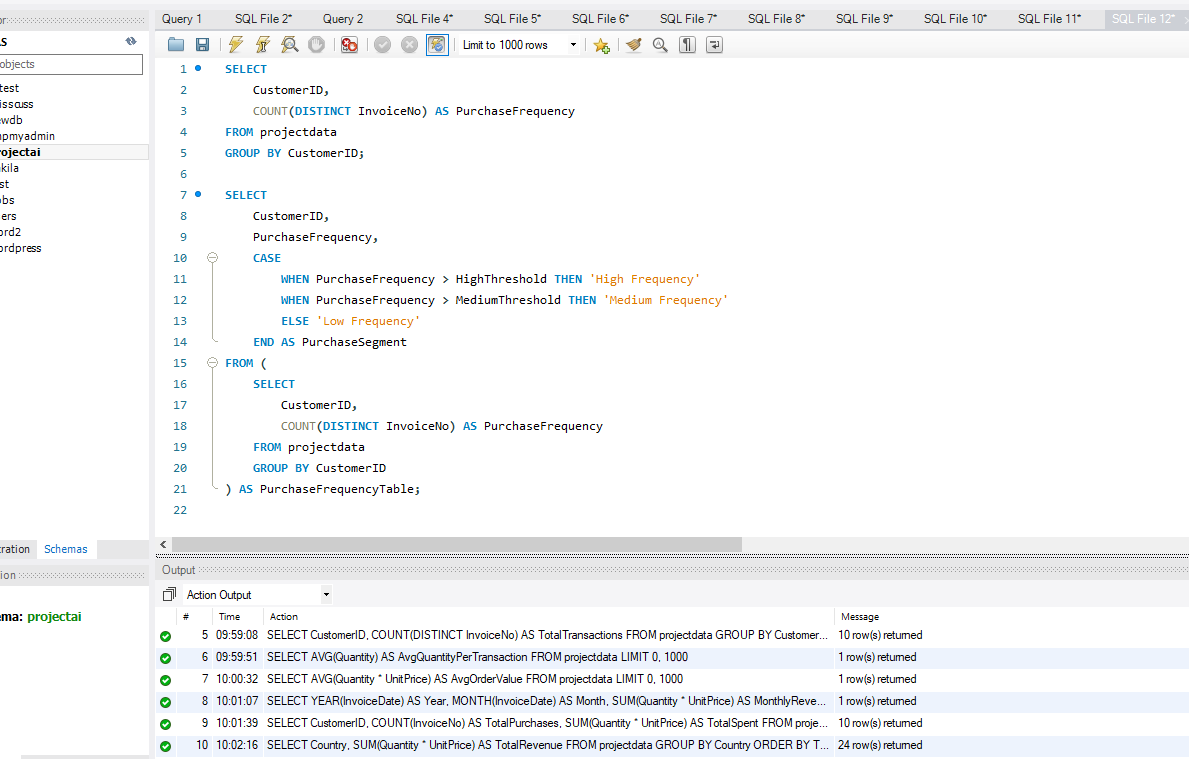
**Description (Number of Unique Products Purchased):**

This SQL query calculates the number of unique products purchased by each customer in the projectdata table. It counts the distinct StockCode values associated with each customer's transactions. The result set displays customers along with the count of unique products they've purchased, ordered in descending order. This analysis helps identify customers with diverse purchase behaviors.



**Description (Customers with Single Purchase):**

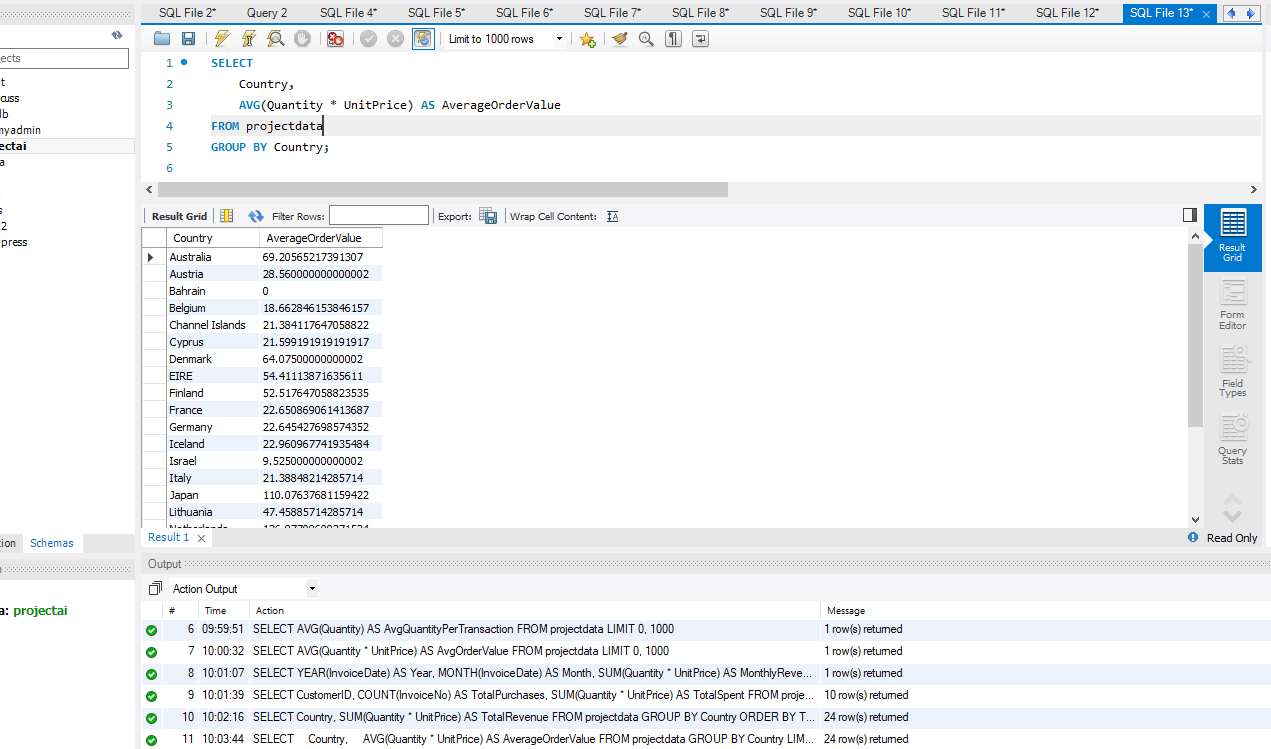
This SQL query identifies customers who have made only a single purchase from the company. It calculates the count of distinct invoices (InvoiceNo) for each customer in the projectdata table and filters the results to include only those customers with a purchase count of 1. The result set displays customers who have interacted with the company just once.



**Description:**

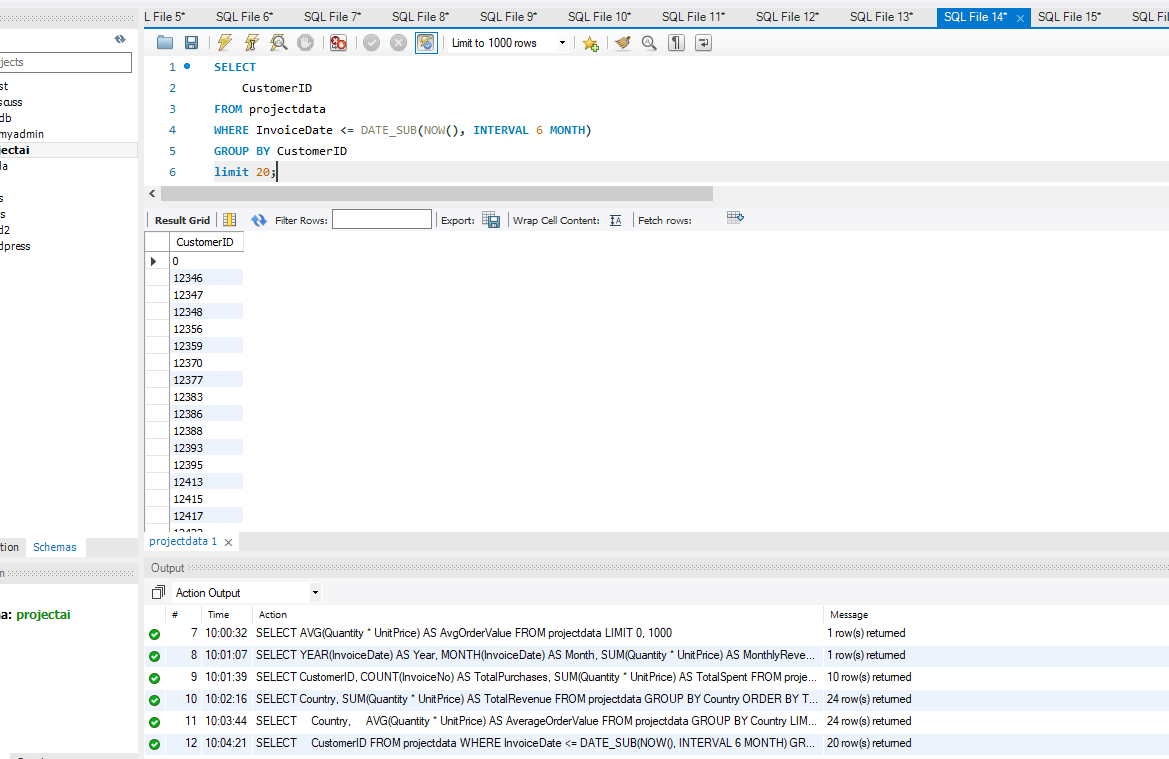
This SQL query calculates the purchase frequency for each customer in the **projectdata** table. It counts the distinct invoices (transactions) associated with each customer and groups the results by **CustomerID**. The result set includes each customer's ID along with their corresponding purchase frequency, giving insights into how frequently each customer makes purchases.

This SQL query performs customer segmentation based on purchase frequency using predefined thresholds. It first calculates the purchase frequency for each customer similarly to the previous query. Then, it defines customer segments ("High Frequency," "Medium Frequency," or "Low Frequency") based on their purchase behavior. Customers are classified into segments based on the calculated purchase frequency and the provided thresholds (**HighThreshold** and **MediumThreshold**). The result set includes each customer's ID, their purchase frequency, and the segment they belong to.



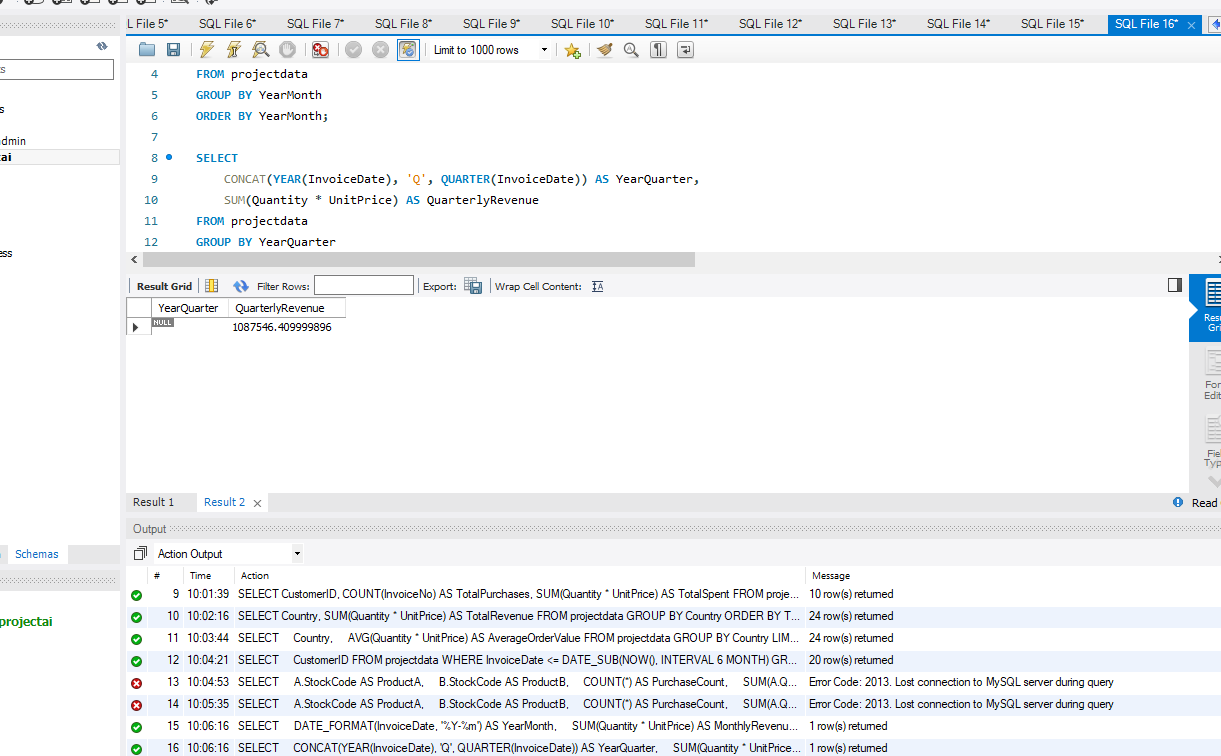
**Description:**

This SQL query calculates the average order value for each country in the **projectdata** table. It groups the data by the Country column and computes the average order value for transactions originating from each country. The average order value is determined by multiplying the quantity of each product (**Quantity**) by its corresponding unit price (**UnitPrice**) for every transaction, and then taking the average for each country. The result set includes each country along with its corresponding calculated average order value. This analysis provides insights into the spending patterns of customers from different countries, helping you identify which countries have higher or lower average order values.



**Description:**

This SQL query identifies customers who haven't made a purchase in the last 6 months based on the **InvoiceDate** column in the **projectdata** table. The query retrieves the **CustomerID** for those customers whose most recent transaction (invoice) occurred more than 6 months ago from the current date. The GROUP BY **CustomerID** groups the results by customer, ensuring that each customer appears only once in the result set. The LIMIT 20 restricts the output to the first 20 inactive customers found. This analysis helps assess customer churn by pinpointing customers who might need re-engagement efforts or attention to prevent attrition.



**Description (Monthly Revenue Trend):**

This SQL query calculates and presents a monthly revenue trend using the **projectdata** table. It groups the data by the year and month (**YearMonth**) derived from the **InvoiceDate**. For each month, the query calculates the total revenue by summing up the product of Quantity and **UnitPrice** for all transactions. The result set displays the revenue achieved for each month, providing insights into the revenue distribution over time.

**Description (Quarterly Revenue Trend):**

This SQL query presents a quarterly revenue trend using the **projectdata** table. It groups the data by the year and quarter (**YearQuarter**) derived from the **InvoiceDate**. For each quarter, the query calculates the total revenue by summing up the product of Quantity and **UnitPrice** for all transactions. The result set displays the revenue achieved for each quarter, allowing you to discern patterns and fluctuations in revenue over specific time intervals.