Project Report: Exploring Customer Insights Through SQL Queries

1. Introduction:

This report details the analysis of customer behavior using SQL queries on a provided dataset. Through the utilization of SQL, we explore various aspects of customer interactions and purchasing patterns to uncover valuable insights.

2. Dataset Overview:

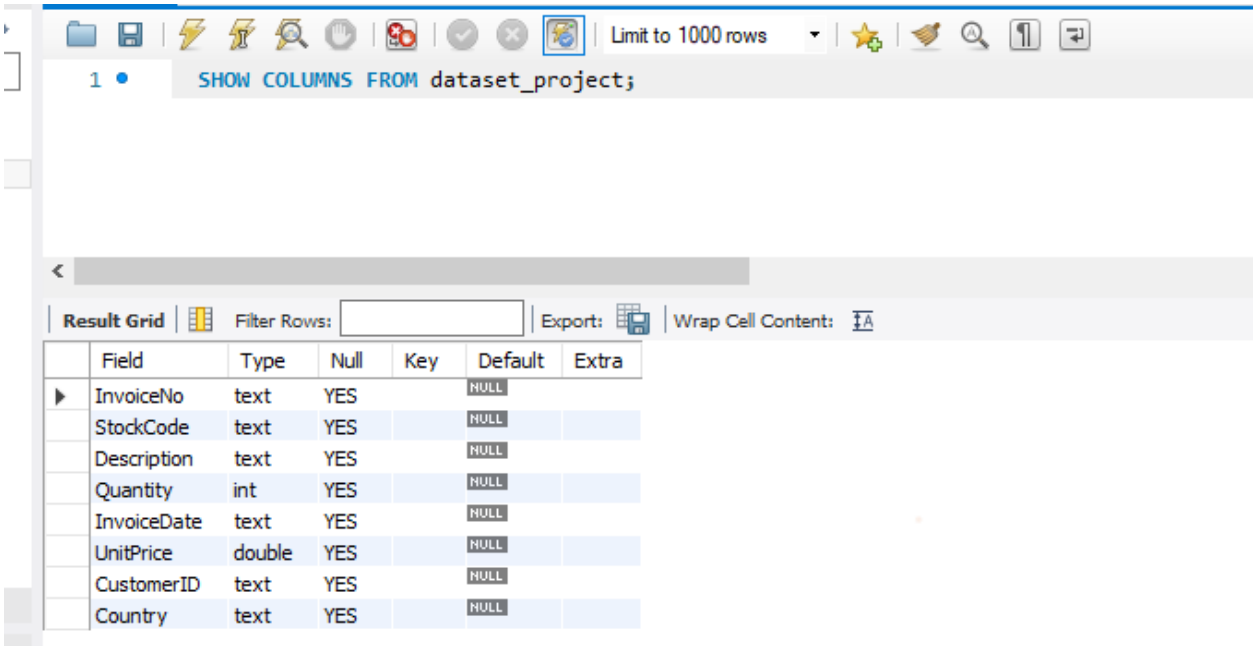
The dataset contains transaction records with fields such as InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

3. Beginner Queries:

3.1 Define Meta Data in MySQL Workbench

- **CODE**

  SHOW COLUMNS FROM dataset_project;
- **Output Screenshot:**



Field	Type	Null	Key	Default	Extra
InvoiceNo	text	YES		NULL	
StockCode	text	YES		NULL	
Description	text	YES		NULL	
Quantity	int	YES		NULL	
InvoiceDate	text	YES		NULL	
UnitPrice	double	YES		NULL	
CustomerID	text	YES		NULL	
Country	text	YES		NULL	

- **Explanation:**

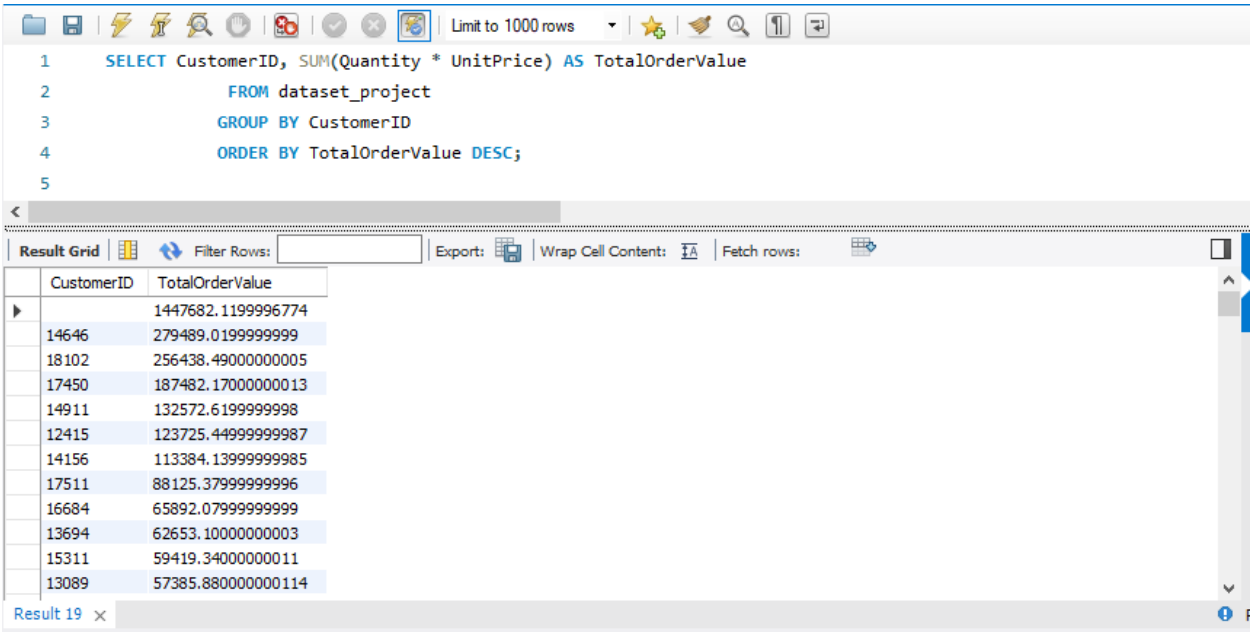
The query above displays the metadata of the dataset, listing the columns, their data types, and any constraints.

3.2 Distribution of Order Values Across Customers

- **Code**:

```
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM dataset_project
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC;
```

- **Output Screenshot:**



CustomerID	TotalOrderValue
	1447682.1199996774
14646	279489.0199999999
18102	256438.49000000005
17450	187482.17000000013
14911	132572.6199999998
12415	123725.44999999987
14156	113384.13999999985
17511	88125.37999999996
16684	65892.07999999999
13694	62653.10000000003
15311	59419.34000000011
13089	57385.880000000114

- **Explanation**:

This query calculates the total order value for each customer, providing insight into the distribution of purchasing behavior.

3.3 Unique Products Purchased by Each Customer

- **Code**

```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductCount
FROM dataset_project
GROUP BY CustomerID
ORDER BY UniqueProductCount DESC;
```

```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductCount
        FROM dataset_project
        GROUP BY CustomerID
        ORDER BY UniqueProductCount DESC;
```

CustomerID	UniqueProductCount
	3699
14911	1794
12748	1769
17841	1331
14096	1121
14298	884
14606	832
14769	718
14156	716
14646	703
13089	636
14456	580

- **Explanation:**

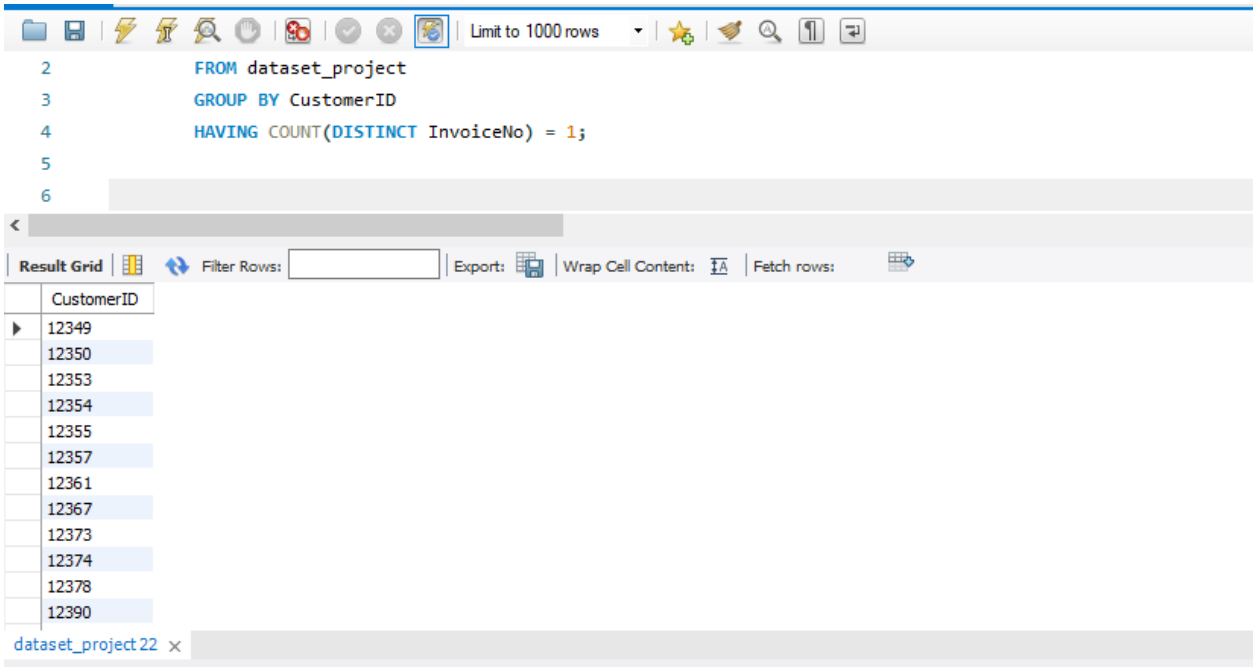
By counting distinct StockCodes for each customer, this query reveals the number of unique products purchased by each customer.

3.4 Customers with Single Purchase

- Code

```
SELECT CustomerID
FROM dataset_project
GROUP BY CustomerID
HAVING COUNT(DISTINCT InvoiceNo) = 1;
```

- **Output Screenshot:**



- **Explanation**

  This query identifies customers who have made only a single purchase from the company.

3.5 Most Commonly Purchased Products Together

- **Code**

```
SELECT A.StockCode AS Product1, B.StockCode AS Product2, COUNT(*) AS Frequency
FROM dataset_project A
JOIN dataset_project B ON A.InvoiceNo = B.InvoiceNo AND A.StockCode < B.StockCode
GROUP BY Product1, Product2
ORDER BY Frequency DESC
Limit 10;
```

- **Explanation**

This query identifies products that are frequently purchased together by customers, revealing potential product affinities.

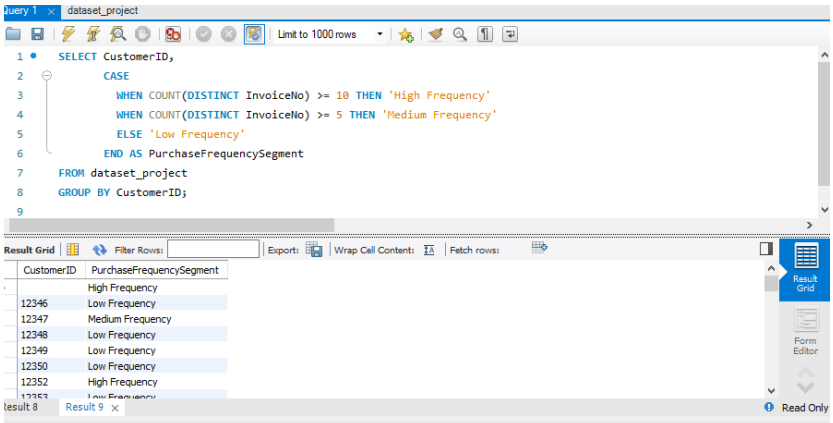
4. Advanced Queries:

4.1 Customer Segmentation by Purchase Frequency**

- **Code**

```
SELECT CustomerID,
    CASE
      WHEN COUNT(DISTINCT InvoiceNo) >= 10 THEN 'High Frequency'
      WHEN COUNT(DISTINCT InvoiceNo) >= 5 THEN 'Medium Frequency'
      ELSE 'Low Frequency'
    END AS PurchaseFrequencySegment
FROM dataset_project
GROUP BY CustomerID;
```

- **Output Screenshot:**



- **Explanation**

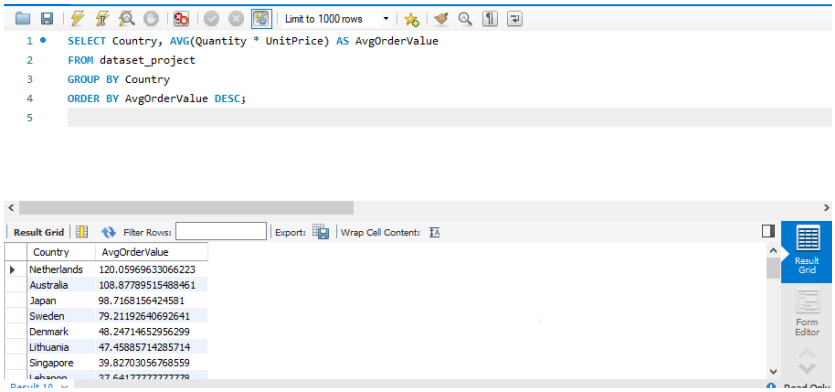
This query categorizes customers into segments based on their purchase frequency, allowing for targeted marketing and engagement strategies.

4.2 Average Order Value by Country

- **Code**

```
SELECT Country, AVG(Quantity * UnitPrice) AS AvgOrderValue
FROM dataset_project
GROUP BY Country
ORDER BY AvgOrderValue DESC;
```

- **Output Screenshot:**



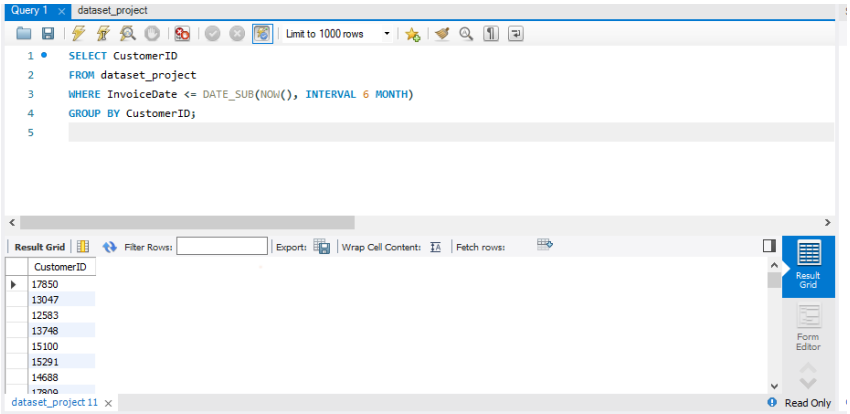
- **Explanation**

This query calculates the average order value for each country, helping to identify regions with the most valuable customers.

4.3 Customer Churn Analysis

- **Code**

```
SELECT CustomerID
FROM dataset_project
WHERE InvoiceDate < DATE_SUB(NOW(), INTERVAL 6 MONTH)
GROUP BY CustomerID;
```



- **Explanation**

By identifying customers who haven't made a purchase in the last 6 months, this query helps assess customer churn.

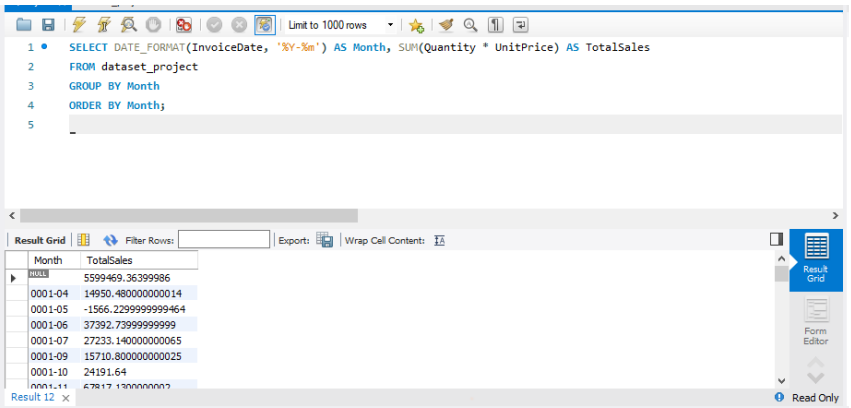
4.4 Time-based Analysis

- **Code**

```
SELECT DATE_FORMAT(InvoiceDate, '%Y-%m') AS Month, SUM(Quantity) AS TotalQuantity
FROM dataset_project
```

GROUP BY Month
ORDER BY Month;

- **Output Screenshot**



- **Explanation**

This query explores trends in customer behavior over time, revealing monthly sales patterns.

5. Conclusion:

In conclusion, the execution of both beginner and advanced queries has provided valuable insights into customer behavior and interactions within the dataset. The beginner queries revealed essential aspects such as order value distribution, unique product preferences, and customer engagement. These insights serve as the groundwork for understanding customer preferences.

Moving to the advanced queries, we gained a deeper understanding of customer segmentation, geographical preferences, churn behavior, product affinities, and time-based trends. These insights enable data-driven decision-making and strategic planning to enhance customer engagement, optimize resources, and boost business growth.

Together, the combination of both beginner and advanced queries has provided a comprehensive perspective on customer behavior, enabling us to tailor strategies and initiatives that resonate with our customers and drive positive outcomes for the business.