Project Report: Analyzing Customer Behavior Using 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:

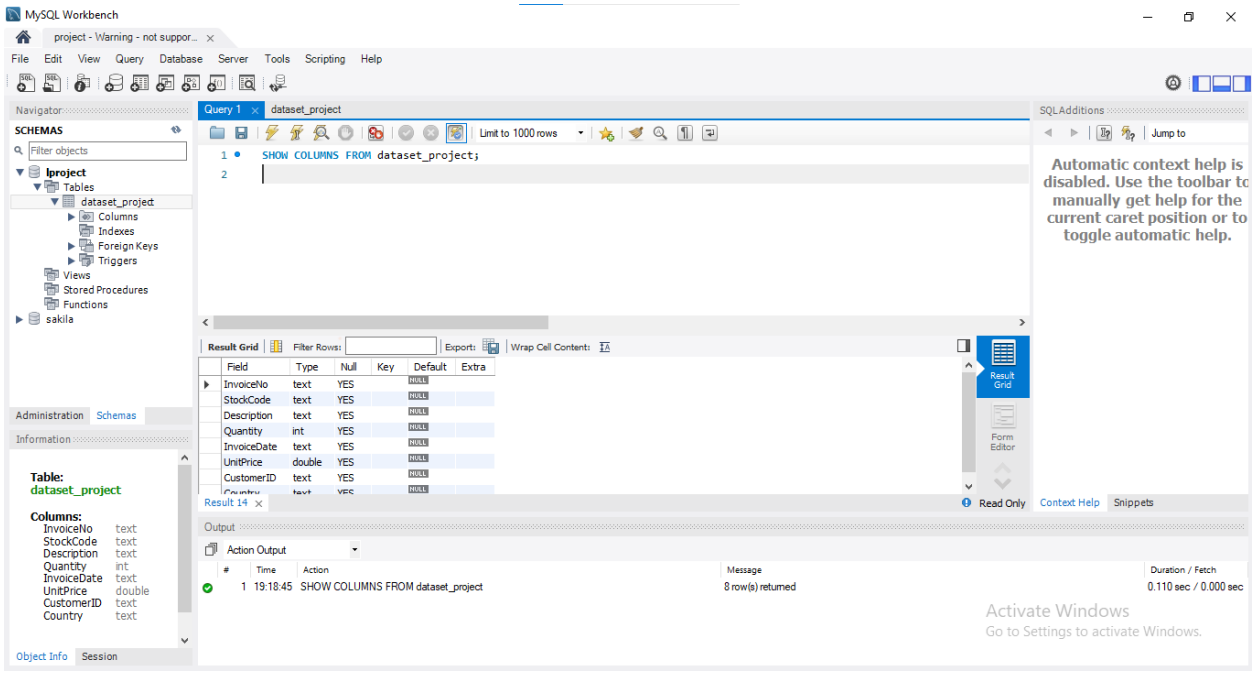
The dataset contains transaction records with fields such as InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

3. Beginner Queries:

3.1 Define Meta Data in MySQL Workbench

- **CODE**

```
SHOW COLUMNS FROM dataset_project;
```



- **Explanation:**

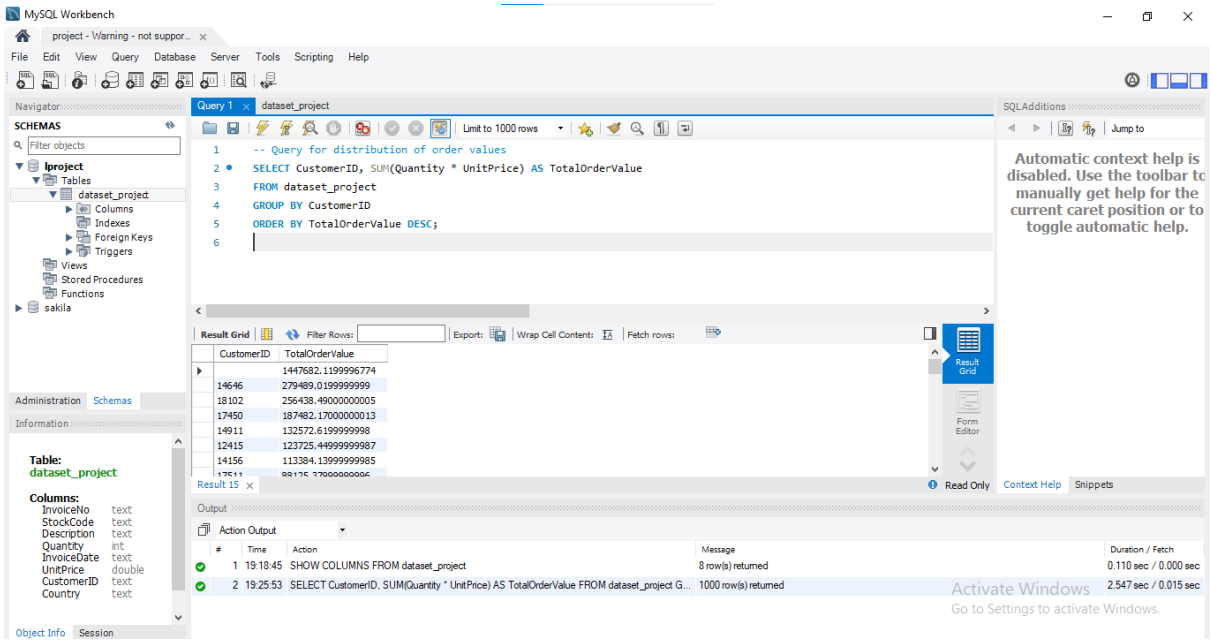
The query above displays the metadata of the dataset, listing the columns, their data types, and any constraints.

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:**



- **Explanation**:

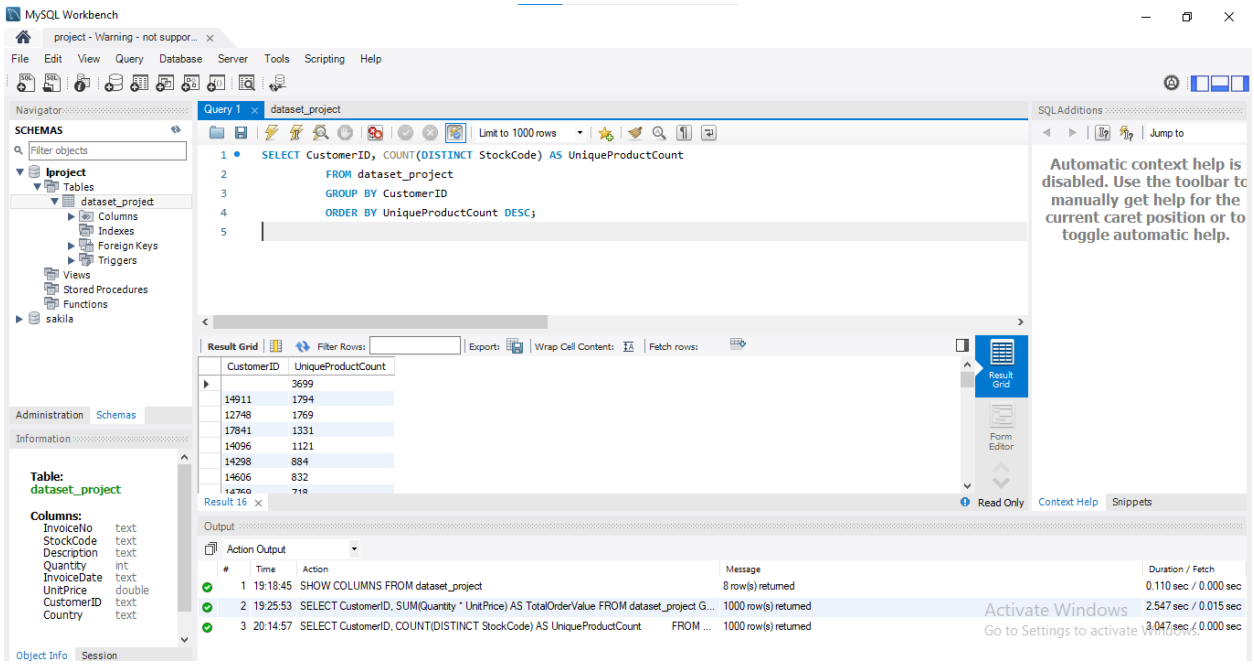
This query calculates the total order value for each customer, providing insight into the distribution of purchasing behavior.

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;
```

- **Output Screenshot:**



- **Explanation:**

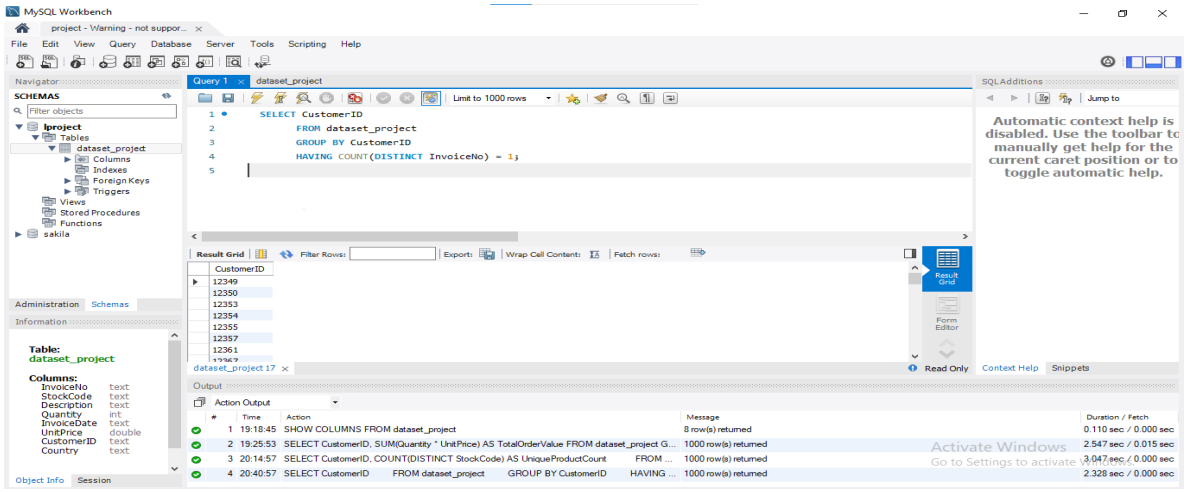
By counting distinct StockCodes for each customer, this query reveals the number of unique products purchased by each customer.

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.

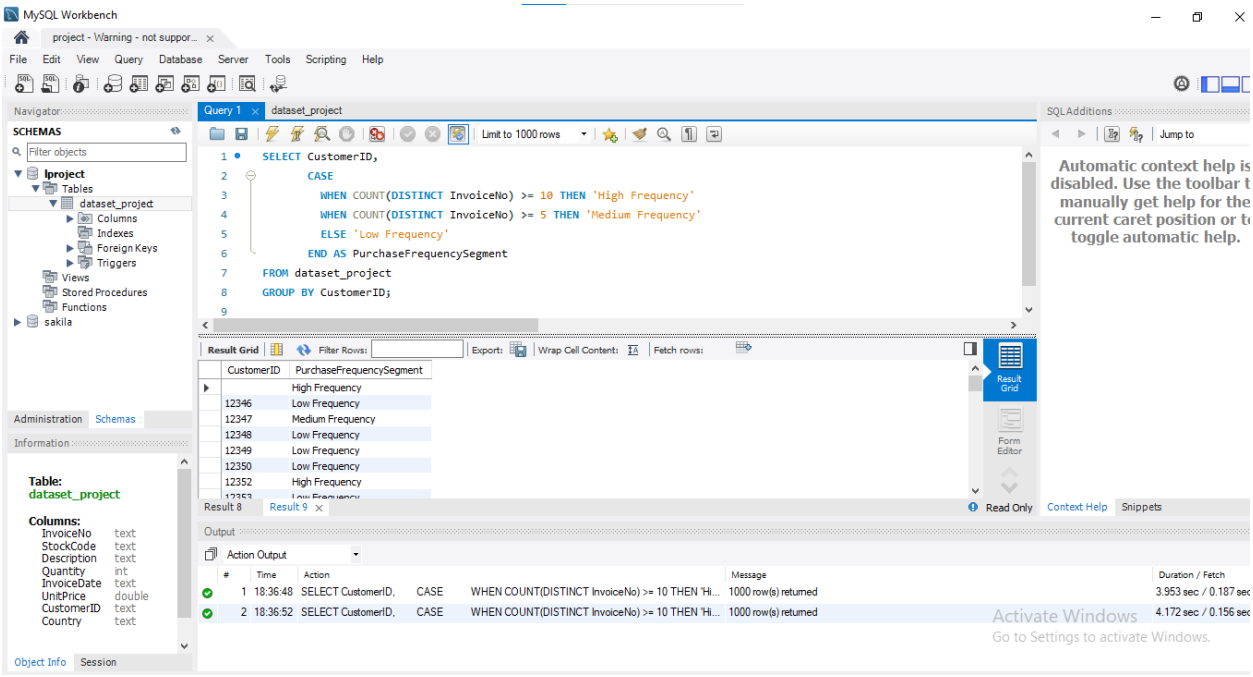
4. Advanced Queries:

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**

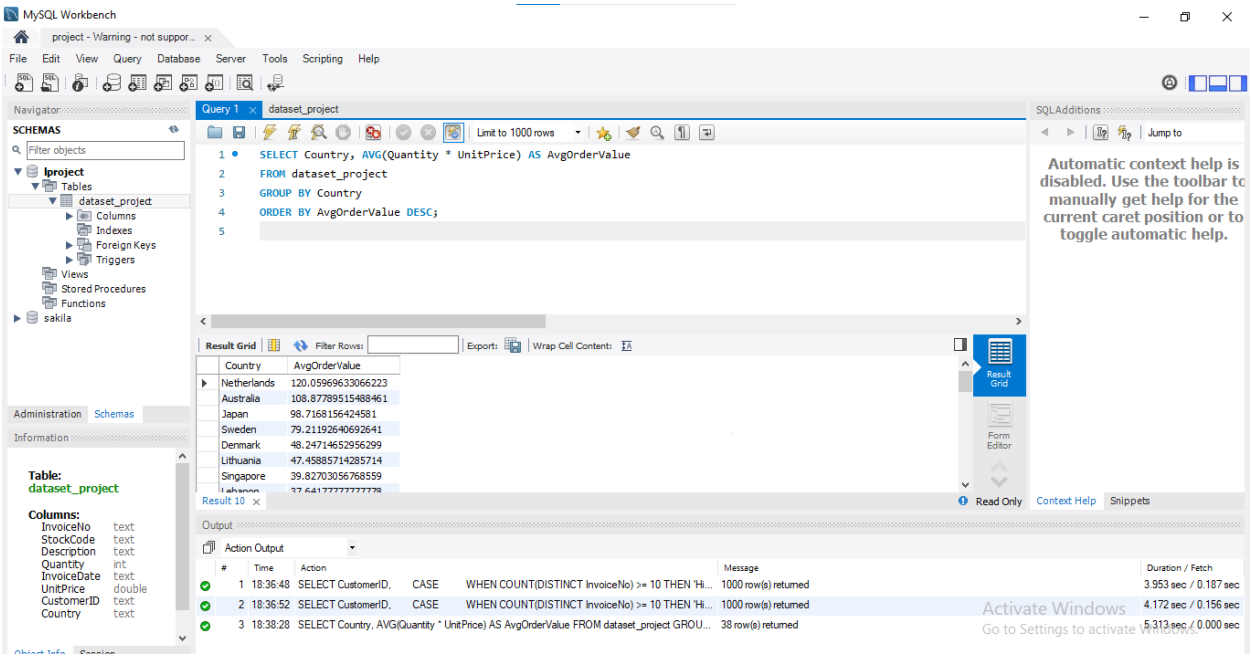
This query categorizes customers into segments based on their purchase frequency, allowing for targeted marketing and engagement strategies.

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:**



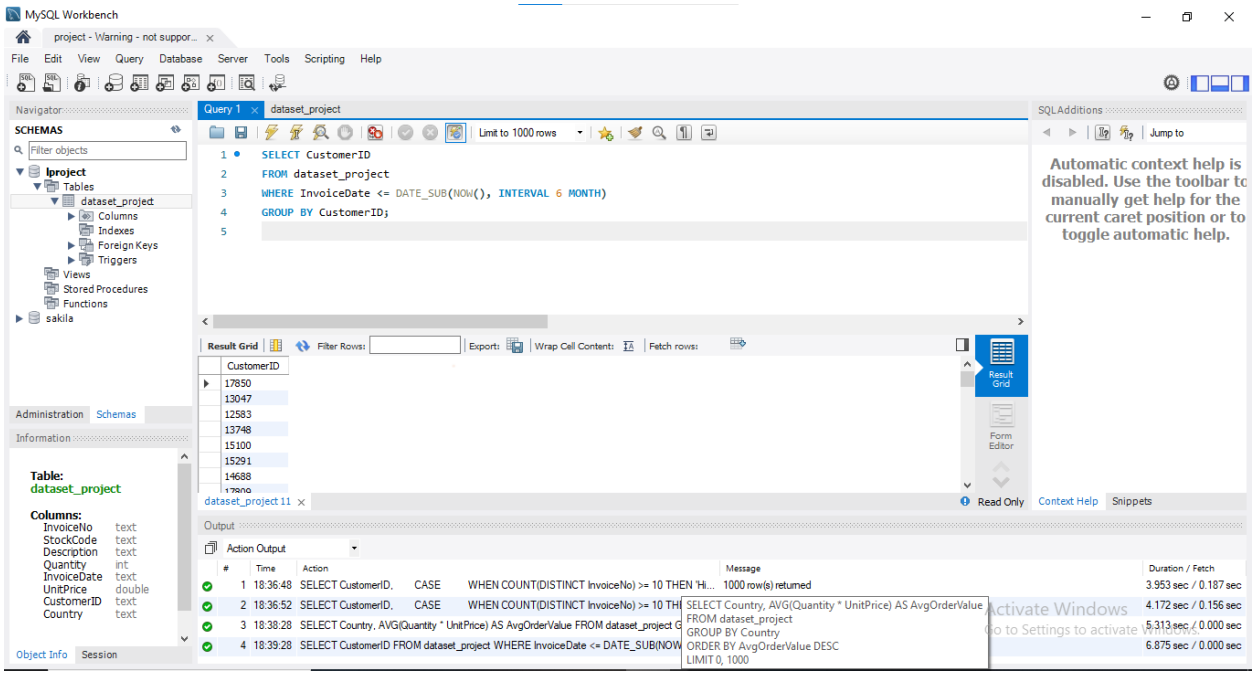
- **Explanation**

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 Product Affinity Analysis

- **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;
```

- **Explanation**

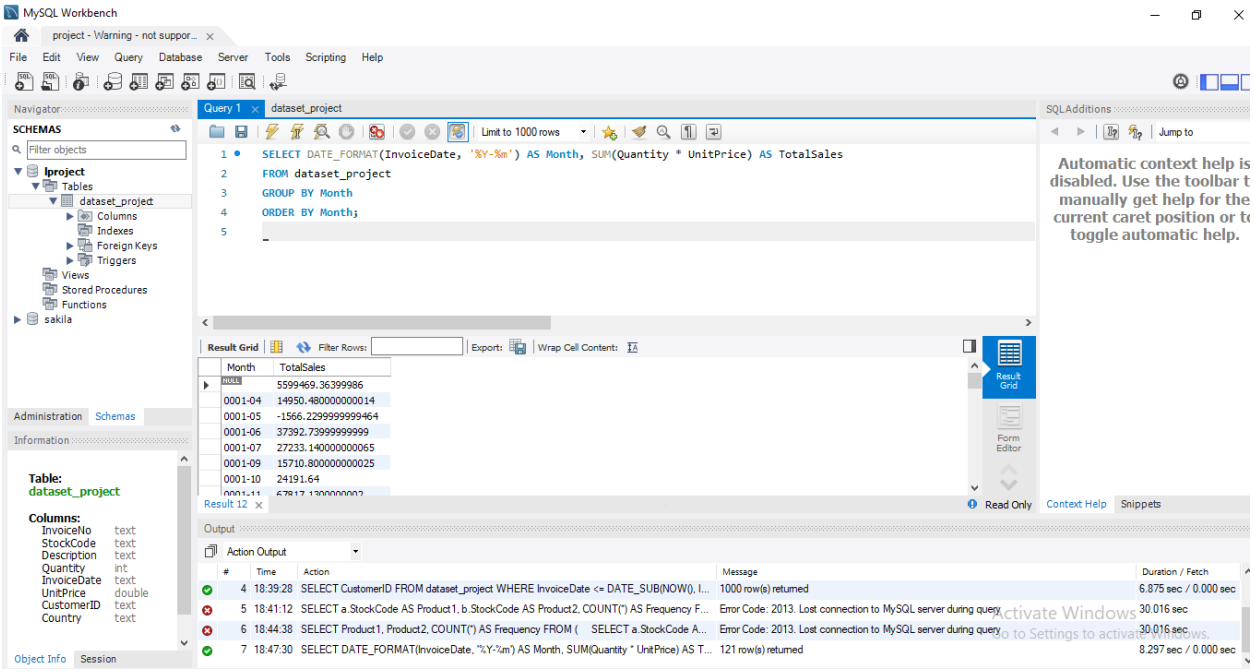
This query determines which products are often purchased together by customers, aiding in cross-selling strategies.

4.5 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:

Incorporating both beginner and advanced queries, our analysis has revealed multifaceted insights into customer behavior within the dataset. The basic queries provided initial glimpses into order values, product preferences, and customer engagement. Building upon this foundation, the advanced queries delved deeper, uncovering customer segmentation, geographic trends, churn risk, product affinities, and temporal patterns.

These insights equip us with the knowledge to tailor strategies for diverse customer segments, optimize marketing efforts, and enhance customer satisfaction. The synergy between basic and advanced queries has transformed raw data into actionable intelligence, driving data-driven decision-making and fostering growth for our business.

6. GitHub and Profile Links:

-GitHub Profile-
Repository: