# Project Report: Online Retail Data Analysis Using SQL

---

## 1. Introduction

### Objective:

The primary goal of this project is to analyze an online retail dataset using SQL to extract valuable insights into customer behavior, product affinity, purchase patterns, and customer segmentation. By leveraging SQL queries, we aim to identify key customer segments, understand purchasing behavior, and determine opportunities for enhancing marketing strategies and improving customer retention.

### Dataset Overview:

The dataset consists of transactional data from an online retail store, containing details such as invoice numbers, product codes, descriptions, quantities, invoice dates, unit prices, customer IDs, and countries. This data was analyzed using SQL queries executed in MySQL Workbench.

---

## 2. Tools and Technologies Used

- **SQL Tool:** MySQL Workbench
- **Database Management System:** MySQL
- **Programming Language:** SQL
- **Data Source:** Online Retail dataset

---

## 3. SQL Queries Executed and Analysis

### Beginner Queries:

➢ **Define Metadata in SQL Tool**

Created a table structure (`online_retail`) to store data using SQL.

```
7    -- Beginner Queries
8
9    -- Query 1: Define Metadata in SQL Tool
10 • CREATE TABLE online_retail (
11       InvoiceNo VARCHAR(10),
12       StockCode VARCHAR(10),
13       Description VARCHAR(255),
14       Quantity INT,
15       InvoiceDate DATETIME,
16       UnitPrice DECIMAL(10, 2),
17       CustomerID VARCHAR(10),
18       Country VARCHAR(100)
19   );
```

➢ **Distribution of Order Values Across All Customers**

Calculated the total order value for each customer to determine top spenders.

```sql
-- Query 2: Distribution of Order Values Across All Customers
SELECT
    CustomerID,
    SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM online_retail
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC;
```

Filter Rows: Export: Wrap Cell Content:

| # | CustomerID | TotalOrderValue |
|---|---|---|
| 1 | 16029 | 3702.12 |
| 2 | 16210 | 2474.7399999999993 |
| 3 | 12433 | 1919.1400000000008 |
| 4 | 17511 | 1825.74 |
| 5 | 17850 | 1499.339999999999 |
| 6 | 13408 | 1024.6800000000003 |
| 7 | 15485 | 950.09 |
| 8 | 12583 | 855.86 |
| 9 | 13694 | 842.12 |

- **Unique Products Purchased by Each Customer**

  Identified the number of unique products purchased by each customer.

```sql
--  Query3: Unique Products Purchased by Each Customer
SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS UniqueProductsPurchased
FROM
    online_retail
GROUP BY
    CustomerID
ORDER BY
    UniqueProductsPurchased DESC;
```

Filter Rows: Export: Wrap Cell Content:

| # | CustomerID | UniqueProductsPurchas |
|---|---|---|
| 1 | 17968 | 74 |
| 2 | 12433 | 73 |
| 3 | 14729 | 69 |
| 4 | 15862 | 64 |

- **Most Commonly Purchased Products Together**

  Determined which products are most frequently bought together.

```sql
-- Query 5: Most Commonly Purchased Products Together
SELECT
    a.StockCode AS ProductA,
    b.StockCode AS ProductB,
    COUNT(*) AS TimesPurchasedTogether
FROM online_retail a
JOIN online_retail b ON a.InvoiceNo = b.InvoiceNo AND a.StockCode < b.StockCode
GROUP BY a.StockCode, b.StockCode
ORDER BY TimesPurchasedTogether DESC
LIMIT 10;
```

Filter Rows: Export: Wrap Cell Content: Fetch rows:

| # | ProductA | ProductB | TimesPurchasedTogethe |
|---|---|---|---|
| 1 | 21448 | 85049E | 15 |
| 2 | 21448 | 22273 | 15 |
| 3 | 21448 | 22749 | 15 |
| 4 | 21448 | 22243 | 15 |
| 5 | 21448 | 21738 | 15 |
| 6 | 21448 | 22077 | 15 |
| 7 | 22197 | 22866 | 11 |
| 8 | 21448 | 22902 | 10 |
| 9 | 21448 | 21708 | 10 |

- **Advanced Queries:**

- **Customer Segmentation by Purchase Frequency**

  Segmented customers based on their purchase frequency into High, Medium, and Low categories.

```sql
-- 1. Customer Segmentation by Purchase Frequency
SELECT
    CustomerID,
    COUNT(DISTINCT InvoiceNo) AS PurchaseFrequency,
    CASE
        WHEN COUNT(DISTINCT InvoiceNo) > 20 THEN 'High'
        WHEN COUNT(DISTINCT InvoiceNo) BETWEEN 5 AND 20 THEN 'Medium'
        ELSE 'Low'
    END AS FrequencySegment
FROM
    online_retail
GROUP BY
    CustomerID
ORDER BY
    PurchaseFrequency DESC;
```

Filter Rows: Export: Wrap Cell Content:

| # | CustomerID | PurchaseFrequenc | FrequencySegmen |
|---|---|---|---|
| 1 | 17850 | 10 | Medium |
| 2 | 14849 | 3 | Low |
| 3 | 13047 | 3 | Low |

- **Average Order Value by Country**

  Computed the average order value for each country to identify the most valuable markets.

```sql
-- 2. Average Order Value by Country
SELECT
    Country,
    AVG(TotalOrderValue) AS AverageOrderValue
FROM (
    SELECT
        Country,
        InvoiceNo,
        SUM(Quantity * UnitPrice) AS TotalOrderValue
    FROM online_retail
    GROUP BY Country, InvoiceNo
) AS OrderValues
GROUP BY Country
ORDER BY AverageOrderValue DESC;
```

Filter Rows: Export: Wrap Cell Content:

| # | Country | AverageOrderValue |
|---|---|---|
| 1 | Norway | 1919.1400000000008 |
| 2 | France | 855.86 |
| 3 | United Kingdom | 369.8485915492959 |
| 4 | Australia | 358.25 |
| 5 | Germany | 261.48 |
| 6 | Netherlands | 192.60000000000002 |

- **Time-Based Analysis**

  Analyzed sales patterns by month and quarter to identify seasonal trends.

- **Monthly Sales Pattern:**

```sql
-- 5. Time-Based Analysis
-- Monthly Sales Pattern:
SELECT
    DATE_FORMAT(STR_TO_DATE(InvoiceDate, '%m/%d/%Y'), '%Y-%m') AS Month,
    SUM(Quantity * UnitPrice) AS TotalSales
FROM
    online_retail
GROUP BY
    Month
ORDER BY
    Month;
```

Filter Rows: Export: Wrap Cell Content:

| # | Month | TotalSales |
|---|---|---|
| 1 | 2010-12 | 29846.580000000133 |

- Quarterly Sales Pattern:

```sql
-- Quarterly Sales Pattern:
SELECT
    CONCAT(YEAR(STR_TO_DATE(InvoiceDate, '%m/%d/%Y')), '-Q', QUARTER(STR_TO_DATE(InvoiceDate, '%m/%d/%Y'))) AS Quarter,
    SUM(Quantity * UnitPrice) AS TotalSales
FROM
    online_retail
GROUP BY
    Quarter
ORDER BY
    Quarter;
```

Filter Rows: Export: Wrap Cell Content:

| # | Quarter | TotalSales |
|---|---|---|
| 1 | 2010-Q4 | 29846.580000000133 |

### 4. Key Findings and Insights

- **Top Spenders:** Identified the highest-value customers based on total spending, allowing targeted promotions and loyalty programs.
- **Customer Segmentation:** Segmented customers into high, medium, and low-frequency groups to tailor marketing efforts accordingly.
- **Product Affinity:** Highlighted commonly co-purchased products, providing insights for cross-selling and bundling opportunities.
- **Customer Churn:** Identified customers at risk of churning, enabling proactive retention strategies.
- **Seasonal Trends:** Monthly and quarterly sales patterns provided data for inventory management and sales forecasting.

---

### 5. Conclusion and Future Work

This analysis of the online retail dataset using SQL has provided valuable insights into customer behavior, product preferences, and sales patterns. The findings can help guide strategic decisions, optimize marketing efforts, and enhance customer retention strategies.

**Future Enhancements:**

- Utilize Business Intelligence (BI) tools for advanced data visualization.
- Develop predictive models using machine learning to forecast customer behavior.
- Extend customer segmentation using additional demographic and behavioral data.

---

### 6. GitHub Profile

- **GitHub Profile**: https://github.com/ayankarim101