Name: Arjamand Ali

Class: DSAI-Section-1

Email: arjamandalii@outlook.com

Github: https://github.com/arjamand/Online-Retail-Segmentation

Beginner Queries.

Task 1 : Define meta data in mysql workbench or any other SQL tool

1. **Query :**

```
-- Defining a new colunm and its meta data
ALTER TABLE online_retail
ADD COLUMN manufacturers VARCHAR(50) NOT NULL ;
```

Explanation :

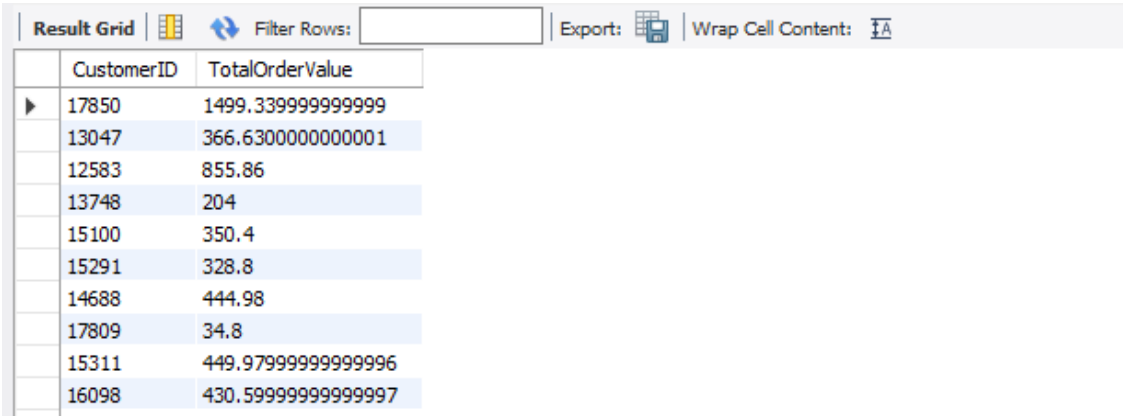
I added a new column called **`manufacturers`** to the **`online_retail`** table. This column stores manufacturer names and has a maximum length of 50 characters, ensuring that every entry has a value.

Task 2 - What is the distribution of order values across all customers in the dataset?

Query :

```
-- distribution of order values across all customers in the dataset
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM online_retail
GROUP BY CustomerID;
```

Output :



CustomerID	TotalOrderValue
17850	1499.339999999999
13047	366.6300000000001
12583	855.86
13748	204
15100	350.4
15291	328.8
14688	444.98
17809	34.8
15311	449.97999999999996
16098	430.59999999999997

Explanation :

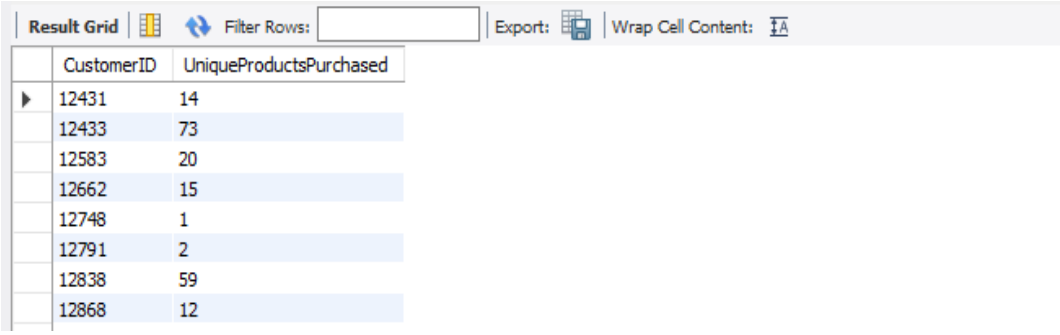
I determined how many distinct products each customer has bought. This metric shows the variety of products each customer engages with.

Task 3 : How many unique products has each customer purchased?

Query :

```
# How many unique products has each customer purchased?
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductsPurchased
FROM online_retail
GROUP BY CustomerID;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	UniqueProductsPurchased
12431	14
12433	73
12583	20
12662	15
12748	1
12791	2
12838	59
12868	12

Explanation :

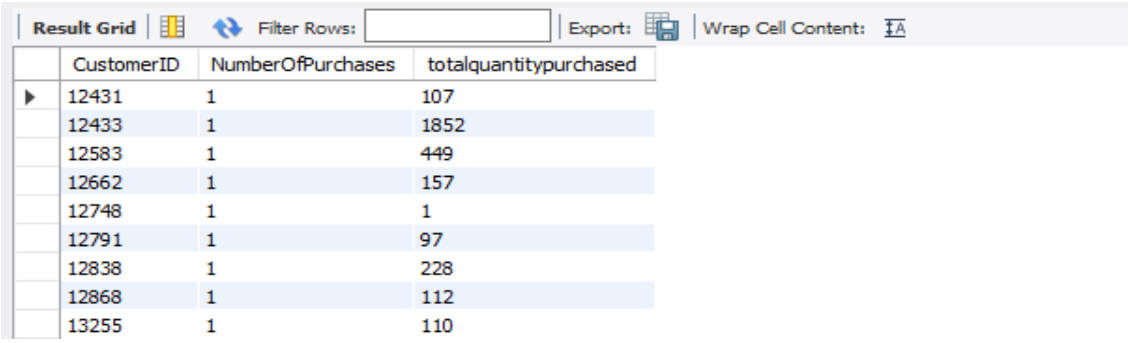
I determined how many distinct products each customer has bought. This metric shows the variety of products each customer engages with.

Task 4 : Which customers have only made a single purchase from the company?

Query :

```
# Which customers have only made a single purchase from the company?
SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS NumberOfPurchases, SUM(Quantity) AS totalquantitypurchased
FROM online_retail
GROUP BY CustomerID
HAVING COUNT(DISTINCT InvoiceNo) = 1;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	NumberOfPurchases	totalquantitypurchased
12431	1	107
12433	1	1852
12583	1	449
12662	1	157
12748	1	1
12791	1	97
12838	1	228
12868	1	112
13255	1	110

Explanation :

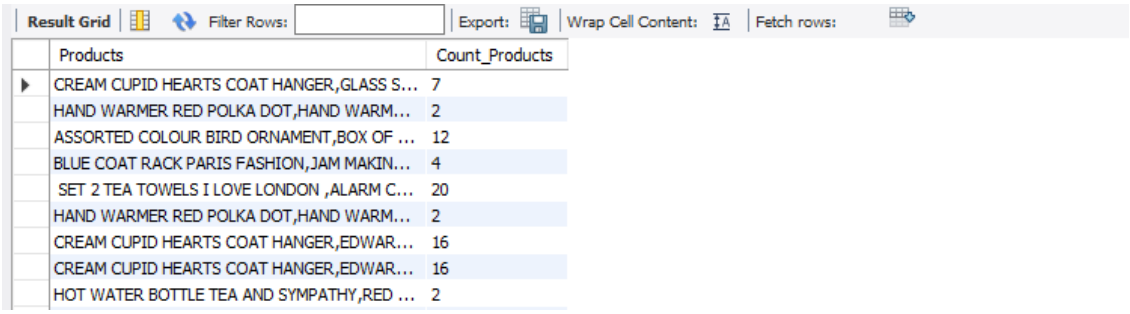
I identified customers who have made only a single purchase. For these customers, I also calculated the total quantity purchased. This helps in recognizing one-time buyers who may need targeted marketing to encourage further purchases.

Task 5 : Which products are most commonly purchased together by customers in the dataset?

Query :

```
# Which products are most commonly purchased together by customers in the dataset?
SELECT GROUP_CONCAT(DISTINCT Description) AS Products,
       COUNT(*) AS Count_Products
FROM online_retail
GROUP BY InvoiceNo
HAVING COUNT(*) > 1
LIMIT 25;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Products	Count_Products
CREAM CUPID HEARTS COAT HANGER,GLASS S...	7
HAND WARMER RED POLKA DOT,HAND WARM...	2
ASSORTED COLOUR BIRD ORNAMENT,BOX OF ...	12
BLUE COAT RACK PARIS FASHION,JAM MAKIN...	4
SET 2 TEA TOWELS I LOVE LONDON ,ALARM C...	20
HAND WARMER RED POLKA DOT,HAND WARM...	2
CREAM CUPID HEARTS COAT HANGER,EDWAR...	16
CREAM CUPID HEARTS COAT HANGER,EDWAR...	16
HOT WATER BOTTLE TEA AND SYMPATHY,RED ...	2

Explanation :

I analyzed which products are most frequently bought together in a single invoice. This helps identify common product pairings and can guide product bundling strategies.

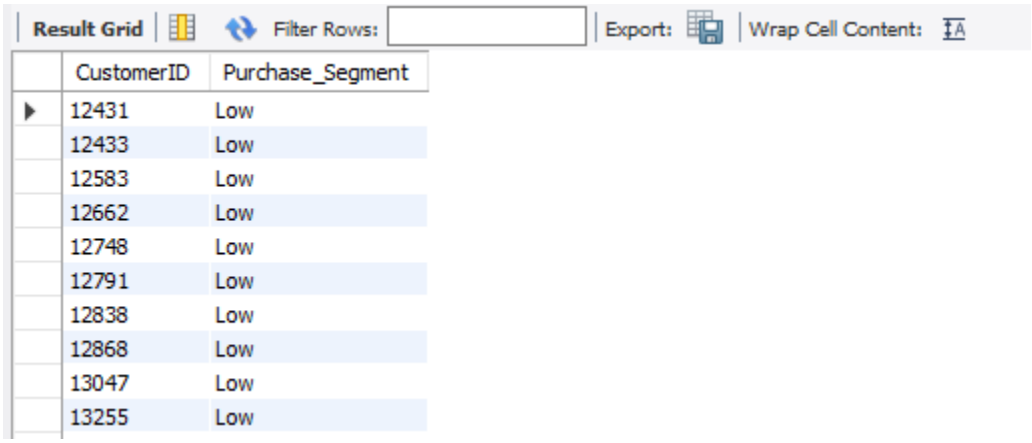
Advance Queries

Task 1 : Group customers into segments based on their purchase frequency, such as high, medium, and low frequency customers. This can help you identify your most loyal customers and those who need more attention.

Queries :

```
# 1-Customer Segmentation by Purchase Frequency
SELECT CustomerID,
      CASE
          WHEN COUNT(DISTINCT InvoiceNo) >= 10 THEN 'High'
          WHEN COUNT(DISTINCT InvoiceNo) >= 5 THEN 'Medium'
          ELSE 'Low'
      END AS Purchase_Segment
FROM online_retail
GROUP BY CustomerID;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	Purchase_Segment
12431	Low
12433	Low
12583	Low
12662	Low
12748	Low
12791	Low
12838	Low
12868	Low
13047	Low
13255	Low

Explanation :

I categorized customers into three segments:

- **High**: Customers with 10 or more distinct invoices.
- **Medium**: Customers with 5 to 9 distinct invoices.
- **Low**: Customers with fewer than 5 distinct invoices.

- **Purpose**: This segmentation helps in identifying loyal customers who frequently purchase and those who may require additional engagement to increase their purchasing frequency.

Task 2 : Calculate the average order value for each country to identify where your most valuable customers are located.

Queries :

```
# Query 3: Average Order Value by Country
SELECT Country,
AVG(Order_Values) AS Avg_Order_Values
FROM (
    SELECT Country, InvoiceNo, SUM(Quantity * UnitPrice) AS Order_Values
    FROM online_retail
    GROUP BY Country, InvoiceNo
) AS total_Orders
GROUP BY Country
ORDER BY Avg_Order_Values DESC;
```

Output :

```
# Query 2: Average Order Value by Country
SELECT Country,
       AVG(Order_Values) AS Avg_Order_Values
FROM (
    SELECT Country,
           InvoiceNo,
           SUM(Quantity * UnitPrice) AS Order_Values
    FROM online_retail
    GROUP BY Country, InvoiceNo
) AS total_Orders
GROUP BY Country
ORDER BY Avg_Order_Values DESC;
```

Explanation :

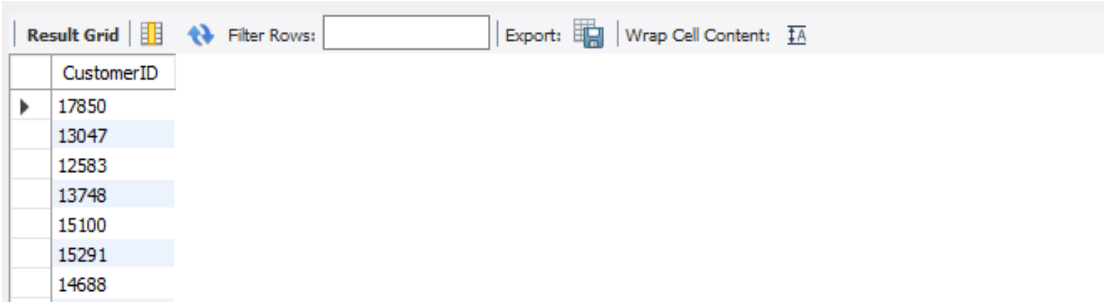
I calculated the average order value for each country by first summing the total order values per invoice and then averaging these values by country. This helps identify which countries have the highest average spending and where my most valuable customers are located.

Task 3 : Identify customers who haven't made a purchase in a specific period (e.g., last 6 months) to assess churn.

Query :

```
# Query 3: Customer analysis
SELECT CustomerID
FROM online_retail
WHERE CustomerID IS NOT NULL
GROUP BY CustomerID
HAVING MAX(InvoiceDate) < DATE_SUB(NOW(), INTERVAL 6 MONTH);
```

Output :



Explanation :

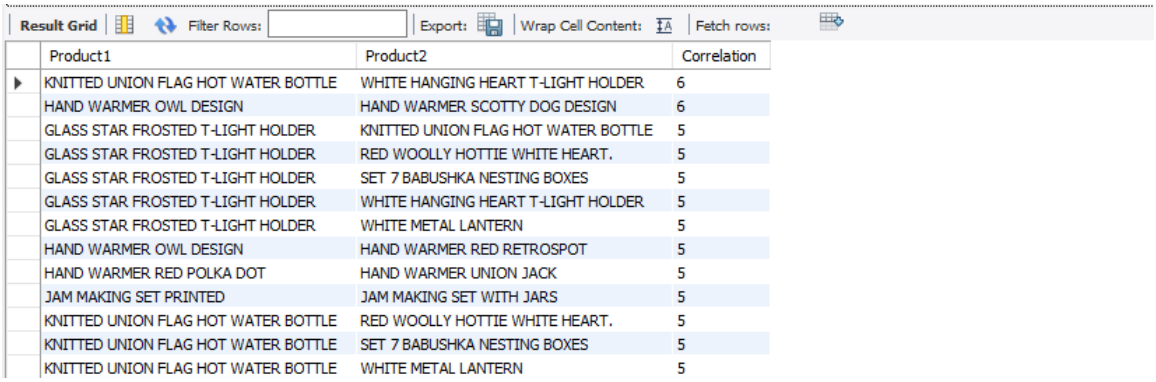
I filtered customers based on the most recent **`InvoiceDate`**. If their last purchase date was more than 6 months ago, they were classified as inactive. This analysis is crucial for assessing customer retention and identifying those who may need targeted re-engagement efforts.

Task 4 : Determine which products are often purchased together by calculating the correlation between product purchases.

Query :

```
SELECT
    p1.Description AS Product1,
    p2.Description AS Product2,
    COUNT(DISTINCT o1.InvoiceNo) AS Correlation
FROM online_retail o1
JOIN online_retail o2
    ON o1.InvoiceNo = o2.InvoiceNo
    AND o1.Description < o2.Description
JOIN online_retail p1
    ON o1.InvoiceNo = p1.InvoiceNo
    AND p1.Description = o1.Description
JOIN online_retail p2
    ON o2.InvoiceNo = p2.InvoiceNo
    AND p2.Description = o2.Description
GROUP BY Product1, Product2
ORDER BY Correlation DESC;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Product1	Product2	Correlation
KNITTED UNION FLAG HOT WATER BOTTLE	WHITE HANGING HEART T-LIGHT HOLDER	6
HAND WARMER OWL DESIGN	HAND WARMER SCOTTY DOG DESIGN	6
GLASS STAR FROSTED T-LIGHT HOLDER	KNITTED UNION FLAG HOT WATER BOTTLE	5
GLASS STAR FROSTED T-LIGHT HOLDER	RED WOOLLY HOTTIE WHITE HEART.	5
GLASS STAR FROSTED T-LIGHT HOLDER	SET 7 BABUSHKA NESTING BOXES	5
GLASS STAR FROSTED T-LIGHT HOLDER	WHITE HANGING HEART T-LIGHT HOLDER	5
GLASS STAR FROSTED T-LIGHT HOLDER	WHITE METAL LANTERN	5
HAND WARMER OWL DESIGN	HAND WARMER RED RETROSPOT	5
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK	5
JAM MAKING SET PRINTED	JAM MAKING SET WITH JARS	5
KNITTED UNION FLAG HOT WATER BOTTLE	RED WOOLLY HOTTIE WHITE HEART.	5
KNITTED UNION FLAG HOT WATER BOTTLE	SET 7 BABUSHKA NESTING BOXES	5
KNITTED UNION FLAG HOT WATER BOTTLE	WHITE METAL LANTERN	5

Explanation :

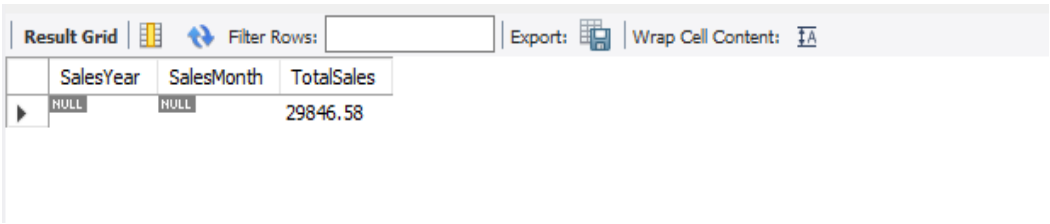
I joined the `online_retail` table with itself to find pairs of products bought together in the same invoice. By grouping and counting these pairs, I determined which products are most commonly bought together. This insight helps in developing cross-selling strategies and optimizing product placements.

Task 5 : Explore trends in customer behavior over time, such as monthly or quarterly sales patterns.

Query :

```
108     # Query 5: Time-based Analysis
109 •   SELECT YEAR(InvoiceDate) AS SalesYear,
110            MONTH(InvoiceDate) AS SalesMonth,
111            SUM(T_Price) AS TotalSales
112     FROM (
113         SELECT InvoiceDate, SUM(Quantity * UnitPrice) AS T_Price
114         FROM online_retail
115         GROUP BY InvoiceDate
116     ) AS T_Invoice
117     GROUP BY SalesYear, SalesMonth
118     ORDER BY SalesYear, SalesMonth;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SalesYear	SalesMonth	TotalSales
NULL	NULL	29846.58

Explanation :

I aggregated sales data by year and month, summing up the total sales **(`SUM(Quantity * UnitPrice)`)** for each period. This analysis reveals sales patterns and trends, helping to understand seasonal effects and evaluate the performance across different time frames.