ONLINE RETAIL SEGMENTATION.

BEGINNER QUERIES.

DEFINE META DATA IN MYSQL WORKBENCH

WHAT IS THE DISTRIBUTION OF ORDER VALUES ACROSS ALL CUSTOMERS IN THE DATASET?

HOW MANY UNIQUE PRODUCTS HAS EACH CUSTOMER PURCHASED?

WHICH CUSTOMERS HAVE ONLY MADE A SINGLE PURCHASE FROM THE COMPANY?

WHICH PRODUCTS ARE MOST COMMONLY PURCHASED TOGETHER BY CUSTOMERS IN THE DATASET?

Define meta data in mysql workbench

Query 1 | SQL File 4

Limit to 1000 rows

```
Metadata is information about other data.
 It describes attributes, context,
 and details of primary data, helping
 with organization, management, and
 understanding. There are types like
 descriptive, structural, administrative,
 technical, rights, and preservation metadata.
 Metadata is crucial for efficient searching,
 data management, and interoperability between systems.
```

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

```sql
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM  `online retail.`.`online retail.`
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC;
```

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

CustomerID	TotalOrderValue
18102	27834.61
15061	19950.660000000007
16029	13202.52
17511	10573.219999999998
14646	8591.879999999997
13089	7738.670000000001
14911	7737.939999999999
12415	7011.379999999997
16210	7000.639999999999
13777	6961.78
13081	5894.419999999997

How many unique products has each customer purchased?

```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductCount
FROM `online retail.`.`online retail.`
GROUP BY CustomerID
```

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

CustomerID	UniqueProductCount
12347	31
12348	13
12370	90
12377	43
12383	37
12386	10
12395	29
12413	24
12415	46
12417	11
12423	16
12427	10
12429	20
12431	21
12433	79
12441	11
12471	57
12472	70
12476	2
12481	21
12483	13
12494	4
12515	16
12540	20
12551	10
12557	5

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

```
SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS PurchaseCount
FROM `online retail.`.`online retail.`
GROUP BY CustomerID
HAVING PurchaseCount = 1;
```

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

CustomerID	PurchaseCount
12347	1
12348	1
12377	1
12383	1
12413	1
12415	1
12417	1
12423	1
12427	1
12429	1
12441	1
12472	1
12476	1

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

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

Result Grid

Product1	Product2	Frequency
22726	22727	82
22086	22910	71
21733	85123A	61
22469	22470	59
21485	22111	57
22111	22112	56
82482	82494L	56
22470	85123A	52
22727	22730	51
22112	22835	51

Result 1

Advance Queries

- **Customer Segmentation by Purchase Frequency**
- 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.
- **2. Average Order Value by Country**
- Calculate the average order value for each country to identify where your most valuable customers are located.
- **3. Customer Churn Analysis**
- Identify customers who haven't made a purchase in a specific period (e.g., last 6 months) to assess churn.
- **4. Product Affinity Analysis**
- Determine which products are often purchased together by calculating the correlation between product purchases.
- **5. Time-based Analysis**
- Explore trends in customer behavior over time, such as monthly or quarterly sales patterns.
-

Customer Segmentation by Purchase Frequency

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.

```sql
SELECT CustomerID,
       CASE
           WHEN PurchaseCount > 5 THEN 'High Frequency'
           WHEN PurchaseCount > 2 THEN 'Medium Frequency'
           ELSE 'Low Frequency'
       END AS PurchaseFrequencySegment
FROM (
    SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS PurchaseCount
    FROM `online retail.`.`online retail.`
    GROUP BY CustomerID
```

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

CustomerID	PurchaseFrequencySegment
12921	Medium Frequency
12928	Low Frequency
12931	Low Frequency
12942	Low Frequency
12944	Low Frequency
12947	Low Frequency
12957	Low Frequency
12963	Low Frequency
12967	Low Frequency
12971	High Frequency

2. Average Order Value by Country

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

```sql
SELECT Country, AVG(TotalOrderValue) AS AverageOrderValue
FROM (
    SELECT Country, InvoiceNo, SUM(Quantity * UnitPrice) AS TotalOrderValue
    FROM `online retail.`.`online retail.`
    GROUP BY Country, InvoiceNo
) AS CountryOrderValues
GROUP BY Country
ORDER BY AverageOrderValue DESC;
```

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

Country	AverageOrderValue
Netherlands	2928.1599999999994
Japan	2568.3566666666666
Norway	1893.5600000000004
Australia	1637.4459999999995
Cyprus	1590.82
Sweden	1460.62
Denmark	1281.5000000000002
Finland	892.8000000000001
Iceland	711.79
Switzerland	652.4599999999998
EIRE	587.592
Germany	508.03800000000007
France	472.26040000000006
Spain	460.9325

3. Customer Churn Analysis

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

```
SELECT CustomerID
FROM `online retail.`.`online retail.`
GROUP BY CustomerID
HAVING MAX(InvoiceDate) <= DATE_SUB(CURDATE(), INTERVAL 6 MONTH);
```

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

CustomerID
17850
13047
12583
13748
15100
15291
14688
17809
15311
16098
18074
17420
16029
16250

. Time-based Analysis

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

```sql
SELECT
    DATE_FORMAT(InvoiceDate, '%Y-%m') AS Month,
    SUM(TotalOrderValue) AS TotalSales
FROM (
    SELECT
        InvoiceDate,
        SUM(Quantity * UnitPrice) AS TotalOrderValue
    FROM
        `online retail.`.`online retail.`
    GROUP BY
        InvoiceNo, InvoiceDate
) AS InvoiceTotals
GROUP BY
    Month
ORDER BY
    Month;
```

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

Month	TotalSales
NULL	647876.6700000007

4. Product Affinity Analysis

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

```sql
SELECT
    p1.StockCode AS Product1,
    p2.StockCode AS Product2,
    COUNT(DISTINCT p1.InvoiceNo) AS CoPurchaseCount
FROM
    `online retail.`.`online retail.` p1
JOIN
   `online retail.`.`online retail.` p2 ON p1.InvoiceNo = p2.InvoiceNo AND p1.StockCode < p2.StockCode
GROUP BY
    Product1, Product2
HAVING
    CoPurchaseCount > 10  -- Adjust the threshold as needed
ORDER BY
    CoPurchaseCount DESC
LIMIT
    10;
```

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

Export recordset to an external file

Product1	Product2	CoPurchaseCount
22086	22910	60
22469	22470	58
21733	85123A	57
22726	22727	57
21485	22111	53
82482	82494L	52
22111	22112	51
22112	22835	48
84029E	84029G	47
22470	85123A	46

Github project link

https://github.com/atheralizair/datamining