High Impact Skills Development Program
in Artificial Intelligence, Data Science, and Blockchain

Project Title: Online Retail Segmentation.

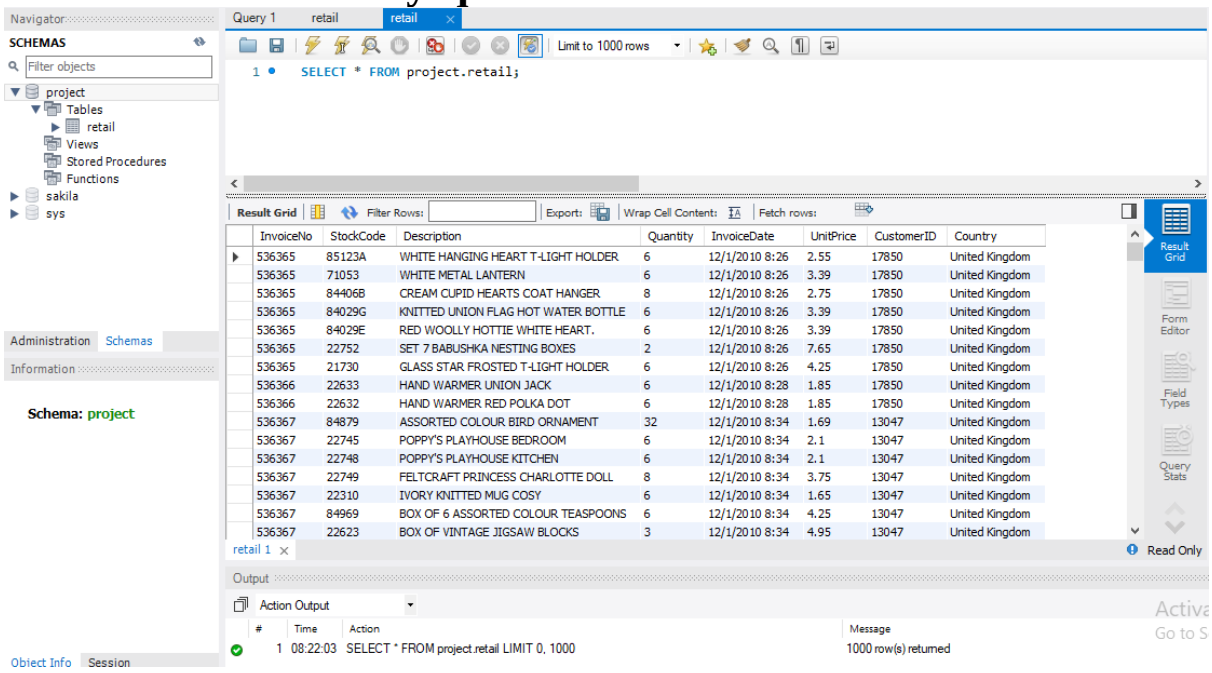
Name: Mureed Hussain Droty

Id: sk23201

Gmail: drotynew3492@gmail.com

Final Project Data Mining

1. Define meta data in mysql workbench:



2.

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

```
SELECT
    InvoiceNo,
    InvoiceDate,
    CustomerID,
    Country,
    SUM(Quantity * UnitPrice) AS OrderValue
FROM
    project.retail
GROUP BY
    InvoiceNo, InvoiceDate, CustomerID, Country
ORDER BY
    OrderValue DESC;
```

Result Grid

InvoiceNo	InvoiceDate	CustomerID	Country	OrderValue
541431	1/18/2011 10:01	12346	United Kingdom	77183.6
541220	1/14/2011 14:11	14156	EIRE	16774.719999999998
537659	12/7/2010 16:43	18102	United Kingdom	15885.489999999998
540815	1/11/2011 12:55	15749	United Kingdom	15160.900000000001
540689	1/11/2011 8:43	17450	United Kingdom	12797.52
541206	1/14/2011 12:24	14646	Netherlands	10389.060000000003
537657	12/7/2010 16:42	18102	United Kingdom	9639.119999999999
539731	12/21/2010 15:05	14646	Netherlands	8520.919999999998

Result 2

Read Only

- **How many unique products has each customer purchased?**

```
SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS UniqueProductsPurchased
FROM
    project.retail
GROUP BY
    CustomerID
ORDER BY
    UniqueProductsPurchased asc;
```

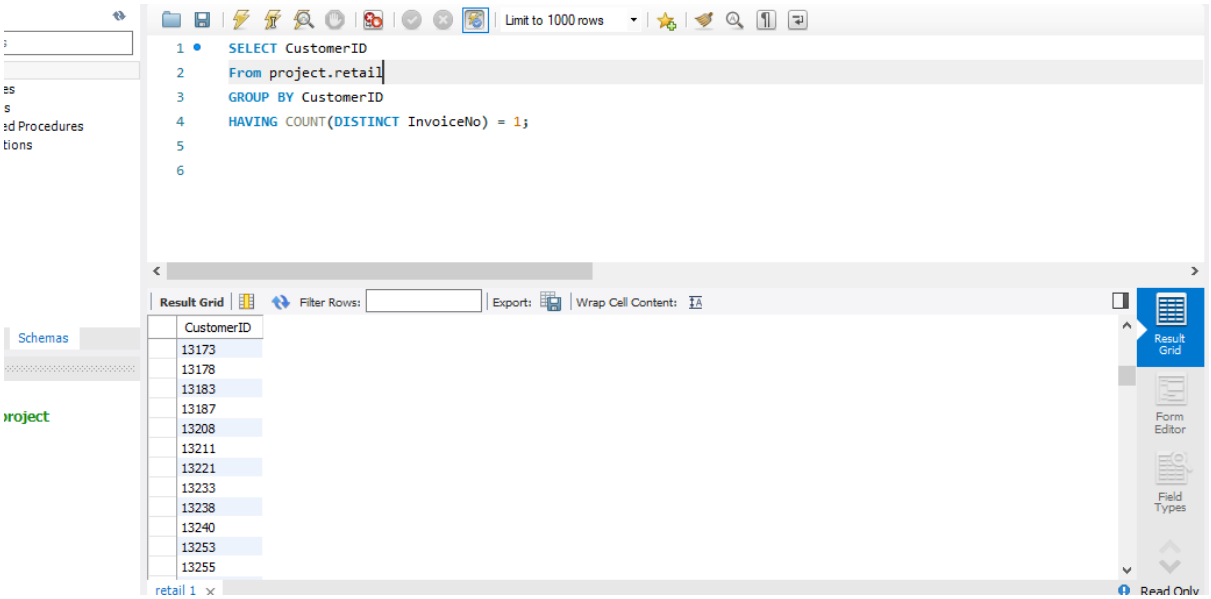
Result Grid

CustomerID	UniqueProductsPurchased
16949	5
17025	5
17430	5
17446	5
17940	5
18032	5
18050	5
18062	5
12464	6
12939	6
13027	6
13255	6

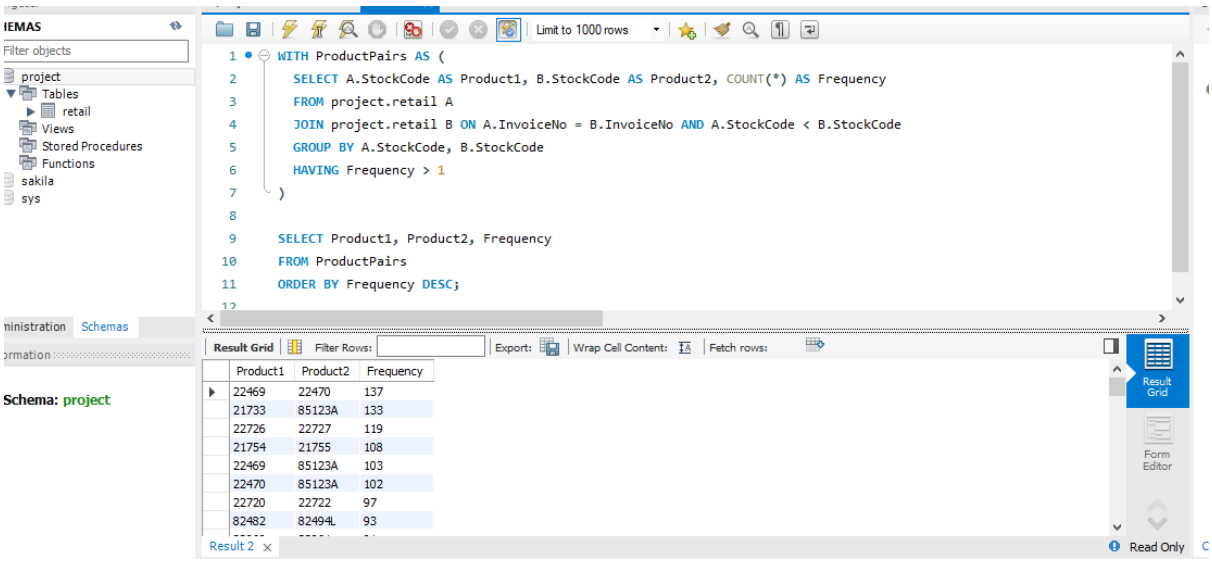
Result 3

Output

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

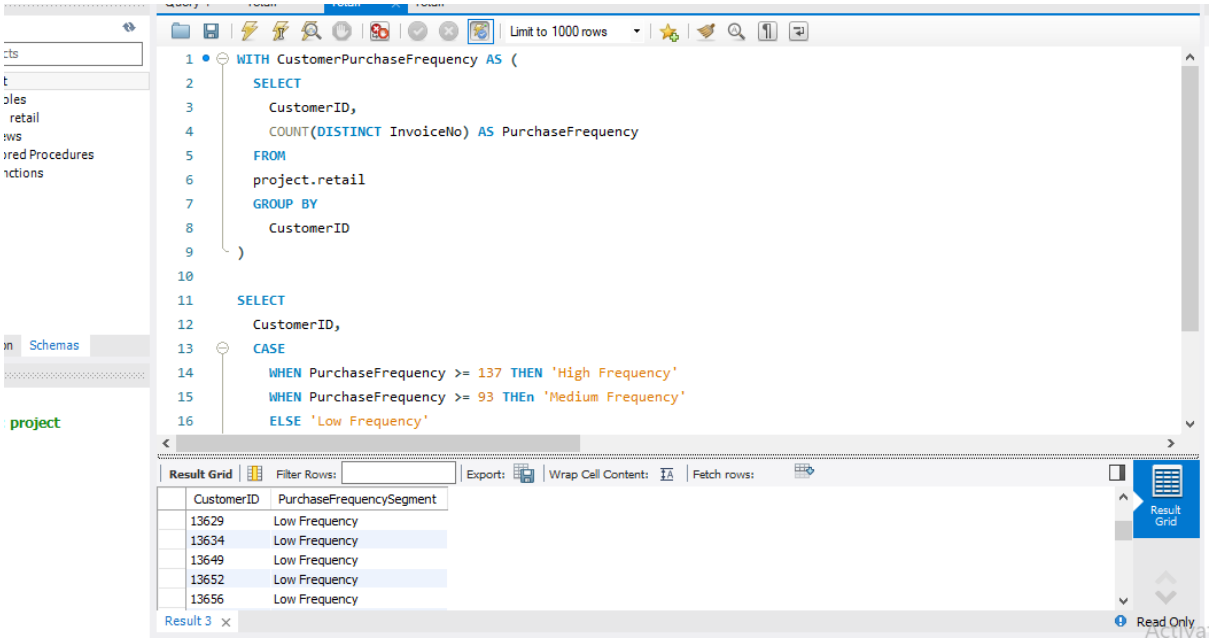


- **Which products are most commonly purchased together by customers in the dataset?**



Advance Queries

1. **Customer Segmentation by Purchase Frequency:**



2. Average Order Value by Country

```sql
SELECT
  Country,
  AVG(TotalOrderValue) AS AverageOrderValue
FROM (
  SELECT
    Country,
    InvoiceNo,
    SUM(UnitPrice * Quantity) AS TotalOrderValue
  FROM
    project.retail  -- Replace with your actual table name
  GROUP BY
    Country, InvoiceNo
) AS OrderValues
GROUP BY
  Country
ORDER BY
  AverageOrderValue DESC;
```

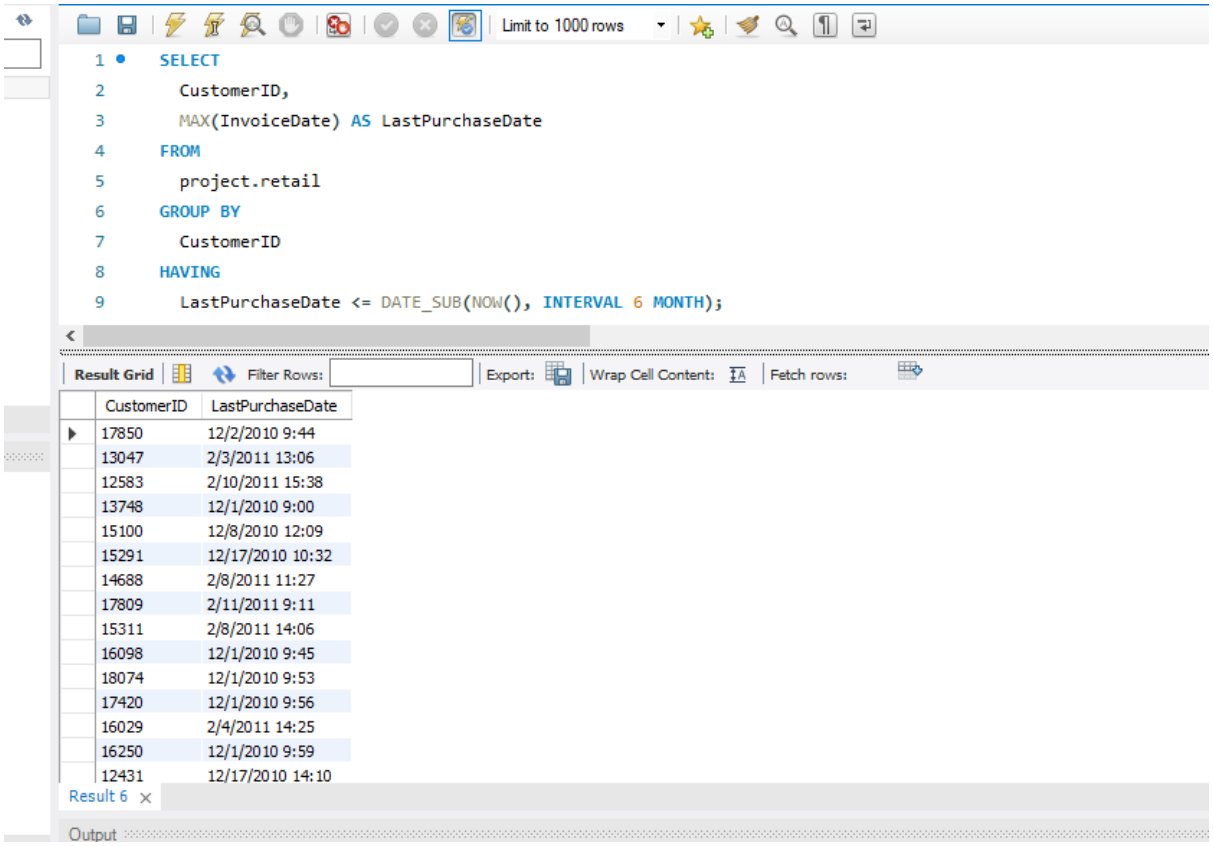
```sql
SELECT
    Country,
    AVG(TotalOrderValue) AS AverageOrderValue
FROM (
    SELECT
```

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

Country	AverageOrderValue
Netherlands	3691.6369999999997
Japan	2697.176
Greece	2661.24
Singapore	2053.07
Lebanon	1693.8800000000003
Sweden	1566.024
Norway	1373.8400000000001
EIRE	1301.34
Denmark	1281.5000000000002
Cyprus	1033.0700000000002
Spain	906.5440000000001
Switzerland	889.9257142857142
Australia	687.3673333333332
Portugal	649.568
Italy	642.8019999999999
Finland	593.94
Iceland	593.5899999999999
France	539.0147368421054
Channel Isl...	519.5550000000001
Germany	503.9799999999999

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,
  MAX(InvoiceDate) AS LastPurchaseDate
FROM
  project.retail
GROUP BY
  CustomerID
HAVING
  LastPurchaseDate <= DATE_SUB(NOW(), INTERVAL 6 MONTH);
```

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

CustomerID	LastPurchaseDate
17850	12/2/2010 9:44
13047	2/3/2011 13:06
12583	2/10/2011 15:38
13748	12/1/2010 9:00
15100	12/8/2010 12:09
15291	12/17/2010 10:32
14688	2/8/2011 11:27
17809	2/11/2011 9:11
15311	2/8/2011 14:06
16098	12/1/2010 9:45
18074	12/1/2010 9:53
17420	12/1/2010 9:56
16029	2/4/2011 14:25
16250	12/1/2010 9:59
12431	12/17/2010 14:10

Result 6

Output

4. Product Affinity Analysis

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

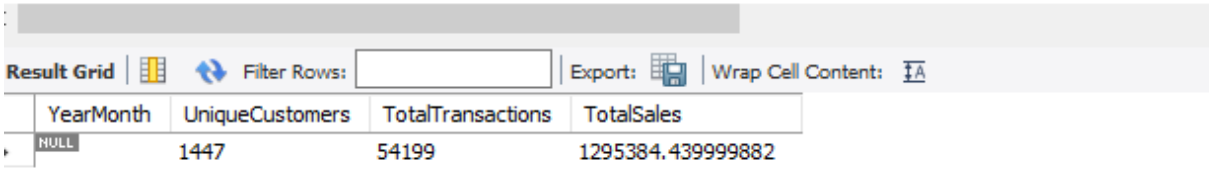
```sql
WITH ProductPairs AS (
  SELECT
    A.StockCode AS Product1,
    B.StockCode AS Product2,
    COUNT(*) AS Frequency
  FROM
    project.retail AS A
    JOIN project.retail AS B ON A.InvoiceNo = B.InvoiceNo AND A.StockCode < B.StockCode
  GROUP BY
    A.StockCode, B.StockCode
),

ProductPurchaseCounts AS (
  SELECT
    StockCode,
    COUNT(*) AS TotalPurchases
  FROM
    project.retail
  GROUP BY
    StockCode
)
```

```sql
ProductPurchaseCounts AS (
  SELECT
    StockCode,
    COUNT(*) AS TotalPurchases
  FROM
    project.retail
  GROUP BY
    StockCode
)

SELECT
  PP.Product1,
  PP.Product2,
  PP.Frequency,
  PP.Frequency / (PP1.TotalPurchases * PP2.TotalPurchases) AS Correlation
FROM
  ProductPairs AS PP
  JOIN ProductPurchaseCounts AS PP1 ON PP.Product1 = PP1.StockCode
  JOIN ProductPurchaseCounts AS PP2 ON PP.Product2 = PP2.StockCode
ORDER BY
  Correlation DESC;
```

5. 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 YearMonth, COUNT(DISTINCT CustomerID)
 AS UniqueCustomers, COUNT(InvoiceNo)
 AS TotalTransactions, SUM(UnitPrice * Quantity) AS TotalSales
FROM project.retail
GROUP BY YearMonth
ORDER BY YearMonth;

```



YearMonth	UniqueCustomers	TotalTransactions	TotalSales
NULL	1447	54199	1295384.439999882

Best of luck