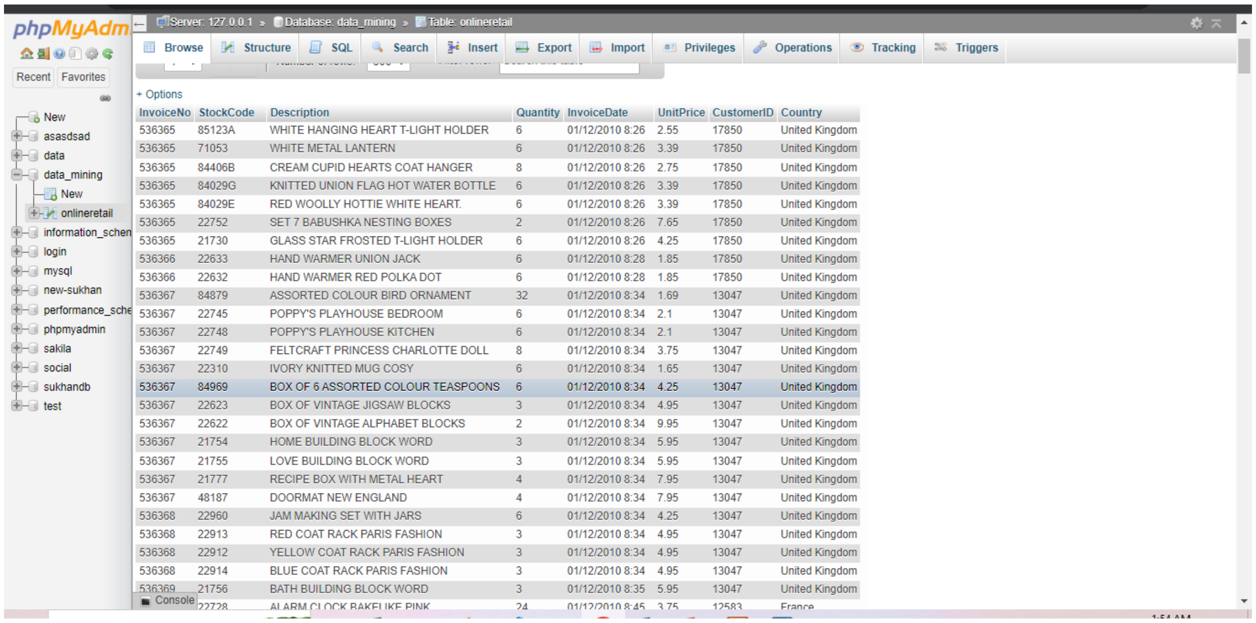
Data Mining Project

Project Title: Online Retail Segmentation.

Task1:

- Define meta data in mysql workbench

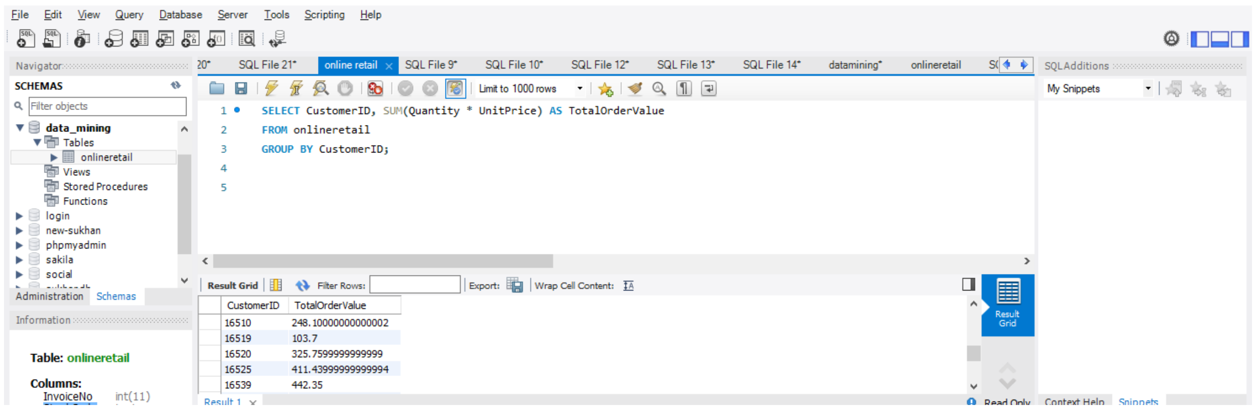


Server: 127.0.0.1 » Database: data_mining » Table: onlineretail

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01/12/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	01/12/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01/12/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01/12/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01/12/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	01/12/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	01/12/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	01/12/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	01/12/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	01/12/2010 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	01/12/2010 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	01/12/2010 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	01/12/2010 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	01/12/2010 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	01/12/2010 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	01/12/2010 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	01/12/2010 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLOCK WORD	3	01/12/2010 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLOCK WORD	3	01/12/2010 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH METAL HEART	4	01/12/2010 8:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW ENGLAND	4	01/12/2010 8:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	01/12/2010 8:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	01/12/2010 8:34	4.95	13047	United Kingdom
536368	22912	YELLOW COAT RACK PARIS FASHION	3	01/12/2010 8:34	4.95	13047	United Kingdom
536368	22914	BLUE COAT RACK PARIS FASHION	3	01/12/2010 8:34	4.95	13047	United Kingdom
536369	21756	BATH BUILDING BLOCK WORD	3	01/12/2010 8:35	5.95	13047	United Kingdom
[illegible]	22728	ALARM CLOCK BAKELIKE PINK	24	01/12/2010 8:45	3.75	12583	France

Task2:

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



```sql
SELECT CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM onlineretail
GROUP BY CustomerID;
```

CustomerID	TotalOrderValue
16510	248.10000000000002
16519	103.7
16520	325.7599999999999
16525	411.43999999999994
16539	442.35

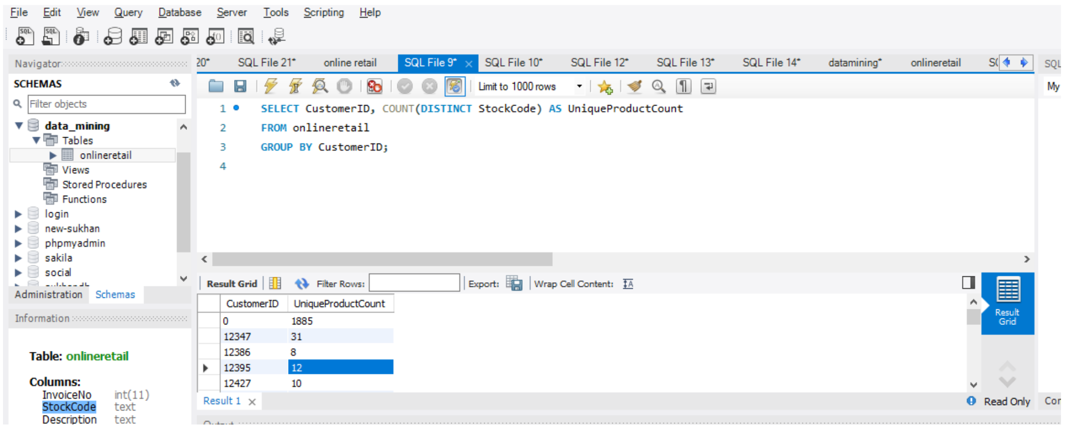
Table: onlineretail

Columns:

InvoiceNo int(11)

Task3:

- How many unique products has each customer purchased?

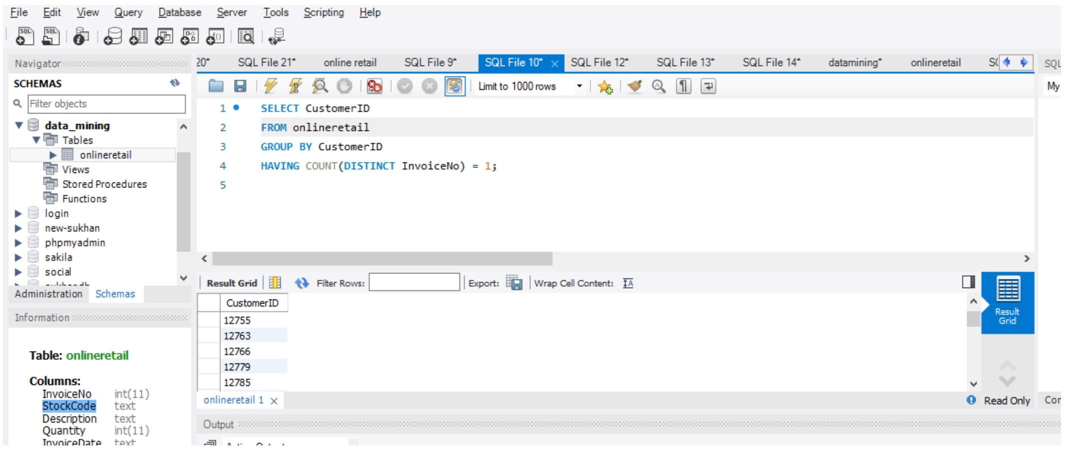


```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductCount
FROM onlineretail
GROUP BY CustomerID;
```

CustomerID	UniqueProductCount
0	1885
12347	31
12386	8
12395	12
12427	10

Task4:

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

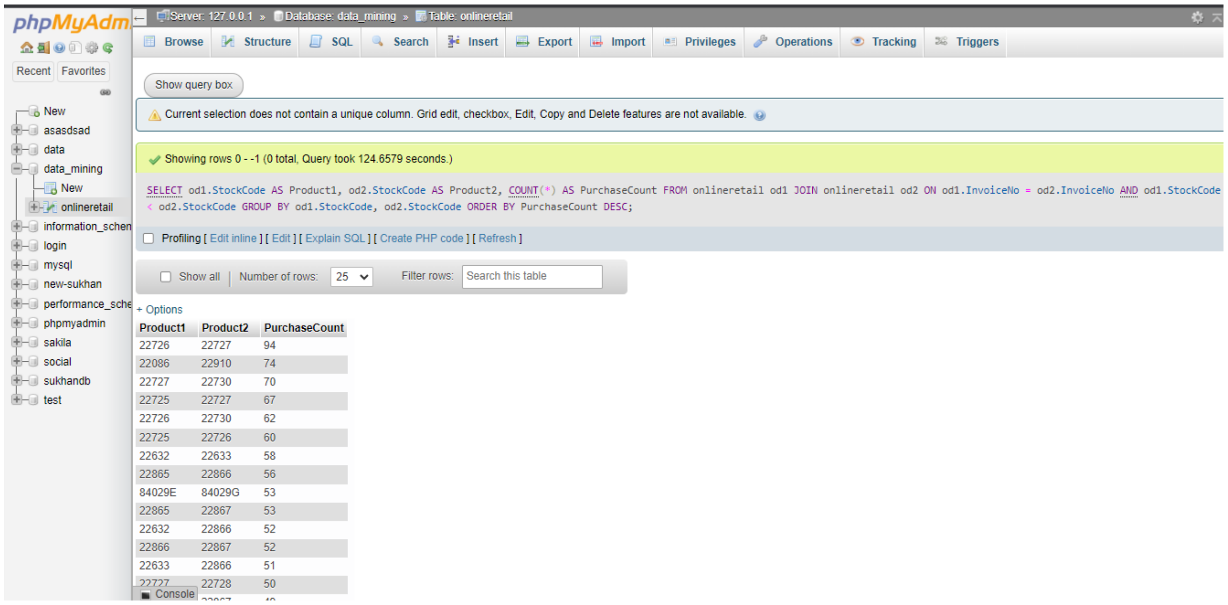


```
SELECT CustomerID
FROM onlineretail
GROUP BY CustomerID
HAVING COUNT(DISTINCT InvoiceNo) = 1;
```

CustomerID
12755
12763
12766
12779
12785

Task5:

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



Server: 127.0.0.1 » Database: data_mining » Table: onlineretail

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - -1 (0 total, Query took 124.6579 seconds.)

```
SELECT od1.StockCode AS Product1, od2.StockCode AS Product2, COUNT(*) AS PurchaseCount FROM onlineretail od1 JOIN onlineretail od2 ON od1.InvoiceNo = od2.InvoiceNo AND od1.StockCode
< od2.StockCode GROUP BY od1.StockCode, od2.StockCode ORDER BY PurchaseCount DESC;
```

Number of rows: 25

Product1	Product2	PurchaseCount
22726	22727	94
22086	22910	74
22727	22730	70
22725	22727	67
22726	22730	62
22725	22726	60
22632	22633	58
22865	22866	56
84029E	84029G	53
22865	22867	53
22632	22866	52
22866	22867	52
22633	22866	51
22727	22728	50

I took this screenshot from localhost phpmyadmin because I was facing an issue in mysql workbench server.

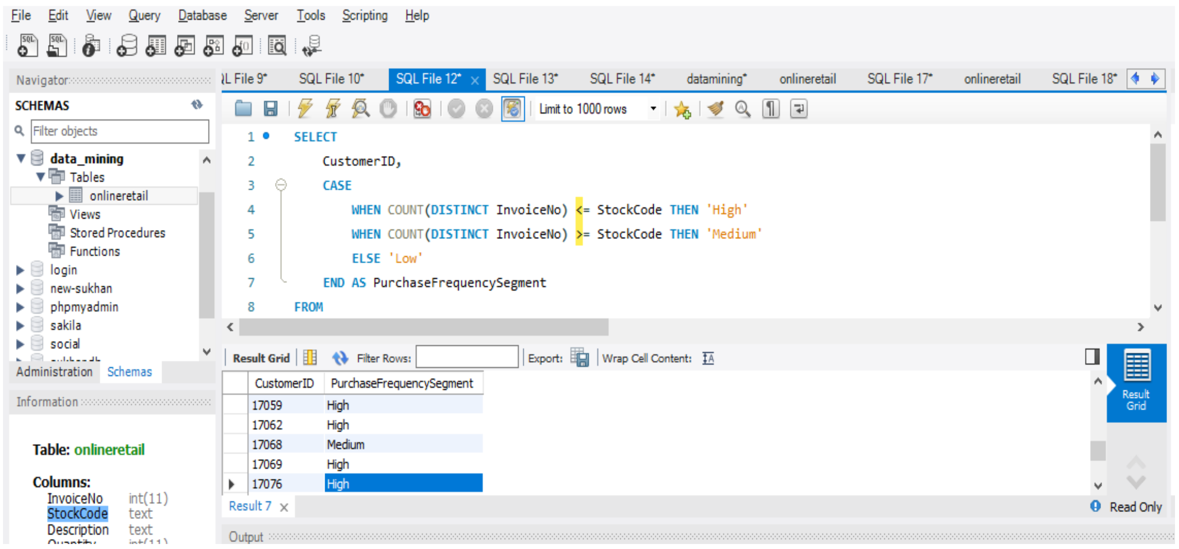
Advance Queries

1. 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.

Code:

```
SELECT
  CustomerID,
  CASE
    WHEN COUNT(DISTINCT InvoiceNo) >= StockCode THEN 'High'
    WHEN COUNT(DISTINCT InvoiceNo) >= StockCode THEN 'Medium'
    ELSE 'Low'
  END AS PurchaseFrequencySegment
FROM
  onlineretail
GROUP BY
  CustomerID;
```



```
SELECT
    CustomerID,
    CASE
        WHEN COUNT(DISTINCT InvoiceNo) <= StockCode THEN 'High'
        WHEN COUNT(DISTINCT InvoiceNo) >= StockCode THEN 'Medium'
        ELSE 'Low'
    END AS PurchaseFrequencySegment
FROM
```

CustomerID	PurchaseFrequencySegment
17059	High
17062	High
17068	Medium
17069	High
17076	High

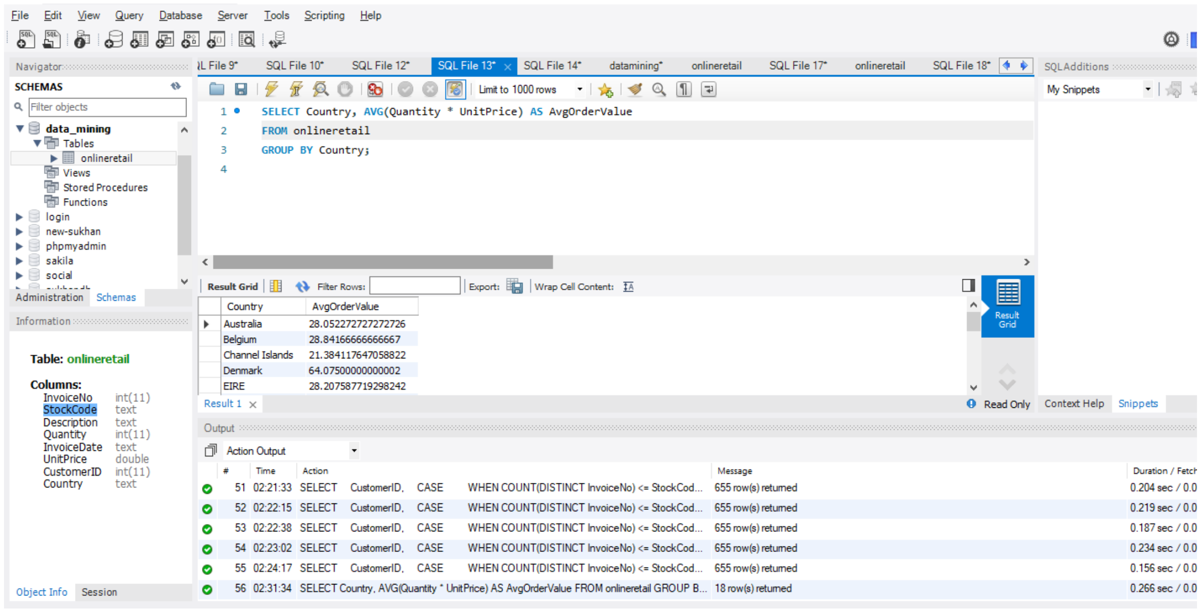
Table: onlineretail

Columns:

InvoiceNo	int(11)
StockCode	text
Description	text

2. Average Order Value by Country

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



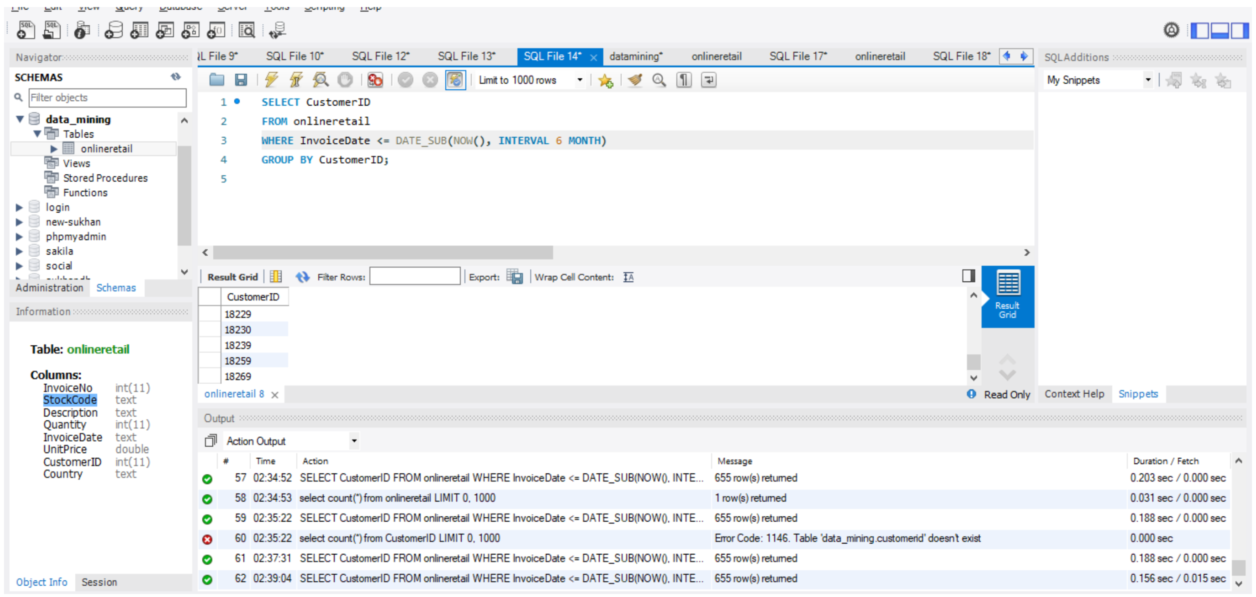
```sql
SELECT Country, AVG(Quantity * UnitPrice) AS AvgOrderValue
FROM onlineretail
GROUP BY Country;
```

Country	AvgOrderValue
Australia	28.052272727272726
Belgium	28.84166666666667
Channel Islands	21.384117647058822
Denmark	64.07500000000002
EIRE	28.207587719298242

```
#   Time      Action                                                                          Message              Duration / Fetch
51  02:21:33  SELECT  CustomerID,  CASE   WHEN COUNT(DISTINCT InvoiceNo) <= StockCod...        655 row(s) returned  0.204 sec / 0.0
52  02:22:15  SELECT  CustomerID,  CASE   WHEN COUNT(DISTINCT InvoiceNo) <= StockCod...        655 row(s) returned  0.219 sec / 0.0
53  02:22:38  SELECT  CustomerID,  CASE   WHEN COUNT(DISTINCT InvoiceNo) <= StockCod...        655 row(s) returned  0.187 sec / 0.0
54  02:23:02  SELECT  CustomerID,  CASE   WHEN COUNT(DISTINCT InvoiceNo) <= StockCod...        655 row(s) returned  0.234 sec / 0.0
55  02:24:17  SELECT  CustomerID,  CASE   WHEN COUNT(DISTINCT InvoiceNo) <= StockCod...        655 row(s) returned  0.156 sec / 0.0
56  02:31:34  SELECT Country, AVG(Quantity * UnitPrice) AS AvgOrderValue FROM onlineretail GROUP B...  18 row(s) returned  0.266 sec / 0.0
```

3. Customer Churn Analysis

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



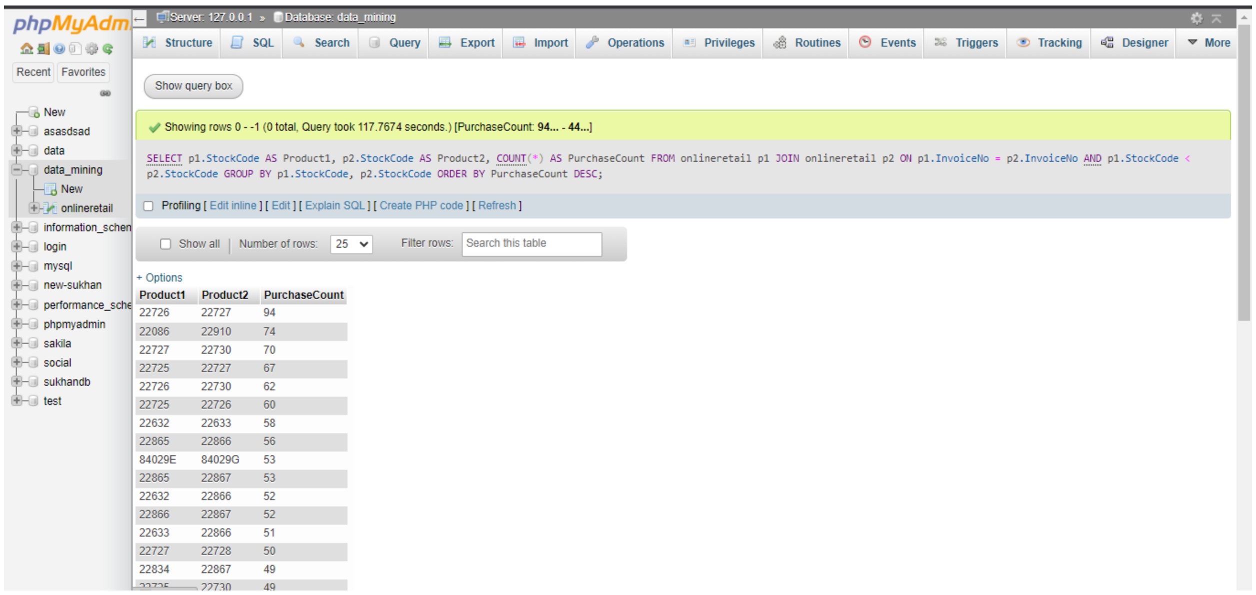
```sql
SELECT CustomerID
FROM onlineretail
WHERE InvoiceDate <= DATE_SUB(NOW(), INTERVAL 6 MONTH)
GROUP BY CustomerID;
```

CustomerID
18229
18230
18239
18259
18269

```
#   Time      Action                                                                        Message                                                  Duration / Fetch
57  02:34:52  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.203 sec / 0.000 sec
58  02:34:53  select count(*) from onlineretail LIMIT 0, 1000                                    1 row(s) returned                                        0.031 sec / 0.000 sec
59  02:35:22  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.188 sec / 0.000 sec
60  02:35:22  select count(*) from CustomerID LIMIT 0, 1000                                      Error Code: 1146. Table 'data_mining.customerid' doesn't exist  0.000 sec
61  02:37:31  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.188 sec / 0.000 sec
62  02:39:04  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.156 sec / 0.015 sec
```

4. Product Affinity Analysis

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



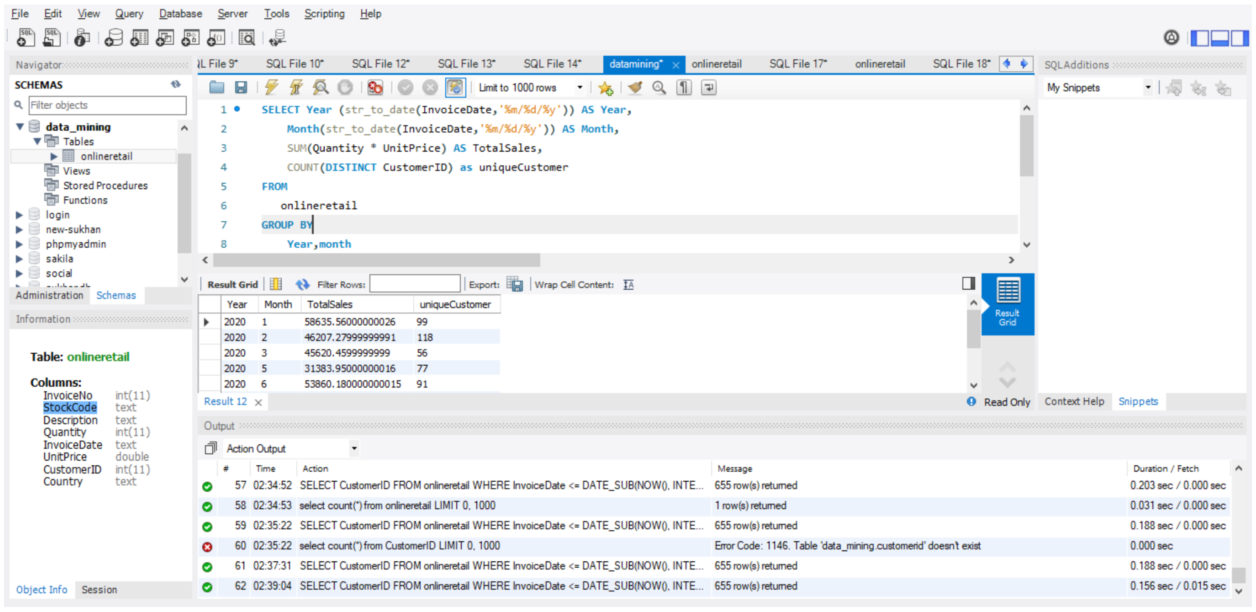
Showing rows 0 - -1 (0 total, Query took 117.7674 seconds.) [PurchaseCount: 94... - 44...]

```
SELECT p1.StockCode AS Product1, p2.StockCode AS Product2, COUNT(*) AS PurchaseCount FROM onlineretail p1 JOIN onlineretail p2 ON p1.InvoiceNo = p2.InvoiceNo AND p1.StockCode <
p2.StockCode GROUP BY p1.StockCode, p2.StockCode ORDER BY PurchaseCount DESC;
```

Product1	Product2	PurchaseCount
22726	22727	94
22086	22910	74
22727	22730	70
22725	22727	67
22726	22730	62
22725	22726	60
22632	22633	58
22865	22866	56
84029E	84029G	53
22865	22867	53
22632	22866	52
22866	22867	52
22633	22866	51
22727	22728	50
22834	22867	49
[illegible]	22730	49

5. Time-based Analysis

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



```
SELECT Year (str_to_date(InvoiceDate,'%m/%d/%y')) AS Year,
    Month(str_to_date(InvoiceDate,'%m/%d/%y')) AS Month,
    SUM(Quantity * UnitPrice) AS TotalSales,
    COUNT(DISTINCT CustomerID) as uniqueCustomer
FROM
   onlineretail
GROUP BY
    Year,month
```

Year	Month	TotalSales	uniqueCustomer
2020	1	58635.56000000026	99
2020	2	46207.27999999991	118
2020	3	45620.4599999999	56
2020	5	31383.95000000016	77
2020	6	53860.180000000015	91

Table: onlineretail

Columns:

InvoiceNo	int(11)
StockCode	text
Description	text
Quantity	int(11)
InvoiceDate	text
UnitPrice	double
CustomerID	int(11)
Country	text

```
#   Time      Action                                                                    Message                                                  Duration / Fetch
57  02:34:52  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.203 sec / 0.000 sec
58  02:34:53  select count(*) from onlineretail LIMIT 0, 1000                               1 row(s) returned                                        0.031 sec / 0.000 sec
59  02:35:22  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.188 sec / 0.000 sec
60  02:35:22  select count(*) from CustomerID LIMIT 0, 1000                                 Error Code: 1146. Table 'data_mining.customerid' doesn't exist  0.000 sec
61  02:37:31  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.188 sec / 0.000 sec
62  02:39:04  SELECT CustomerID FROM onlineretail WHERE InvoiceDate <= DATE_SUB(NOW(), INTE...  655 row(s) returned                                      0.156 sec / 0.015 sec
```

The End