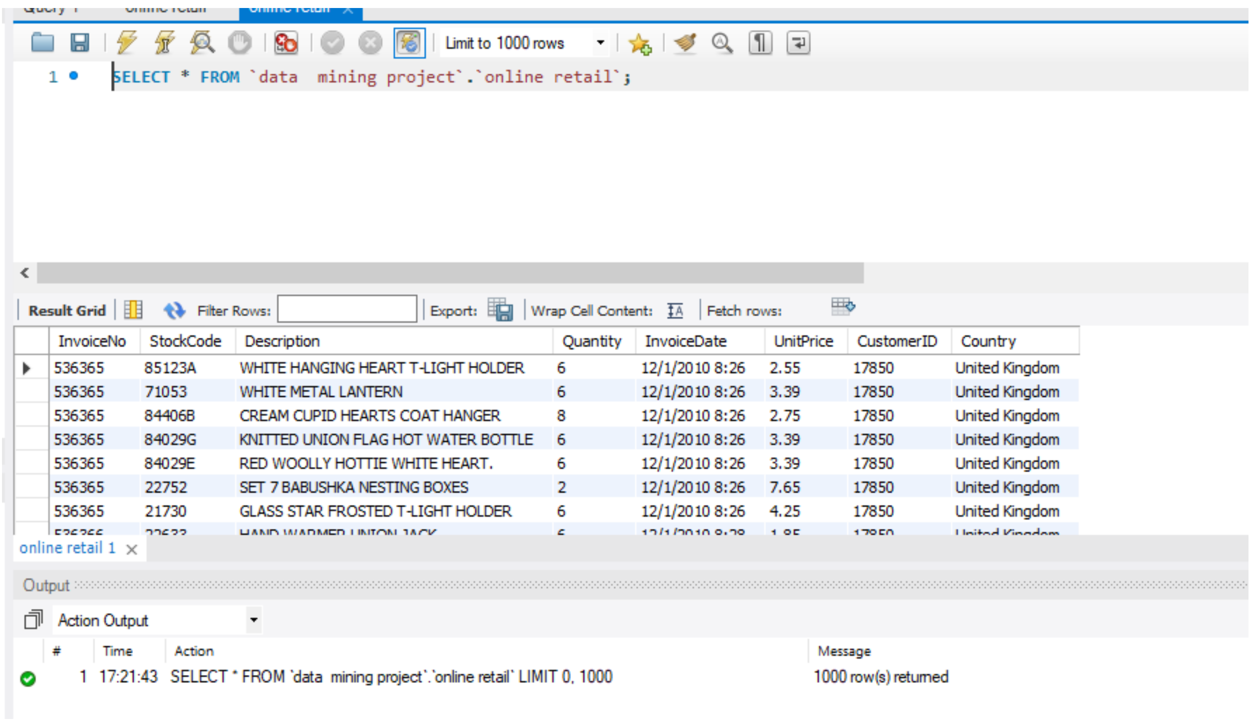
DATA MINING PROJECT

1-METADATA

a. Define metadata in my SQL workbench ?

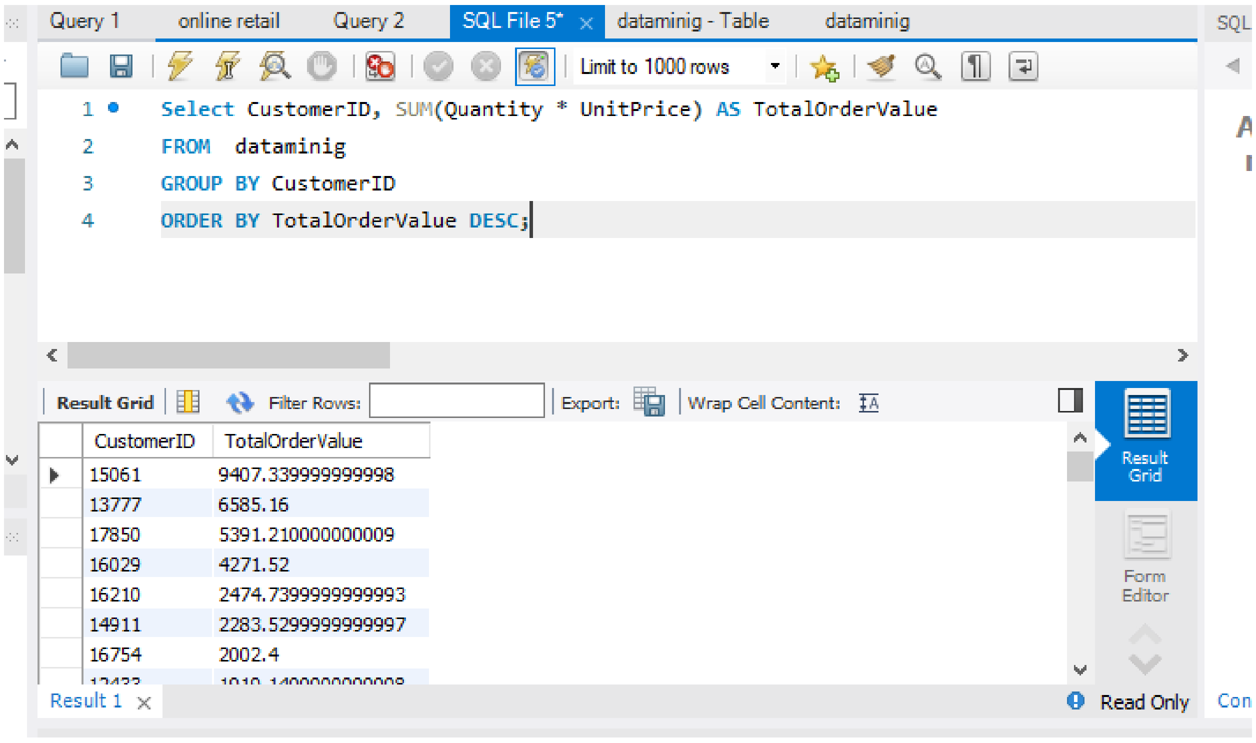


```
SELECT * FROM `data  mining project`.`online retail`;
```

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom

```
#   Time      Action                                                         Message
1   17:21:43  SELECT * FROM `data  mining project`.`online retail` LIMIT 0, 1000   1000 row(s) returned
```

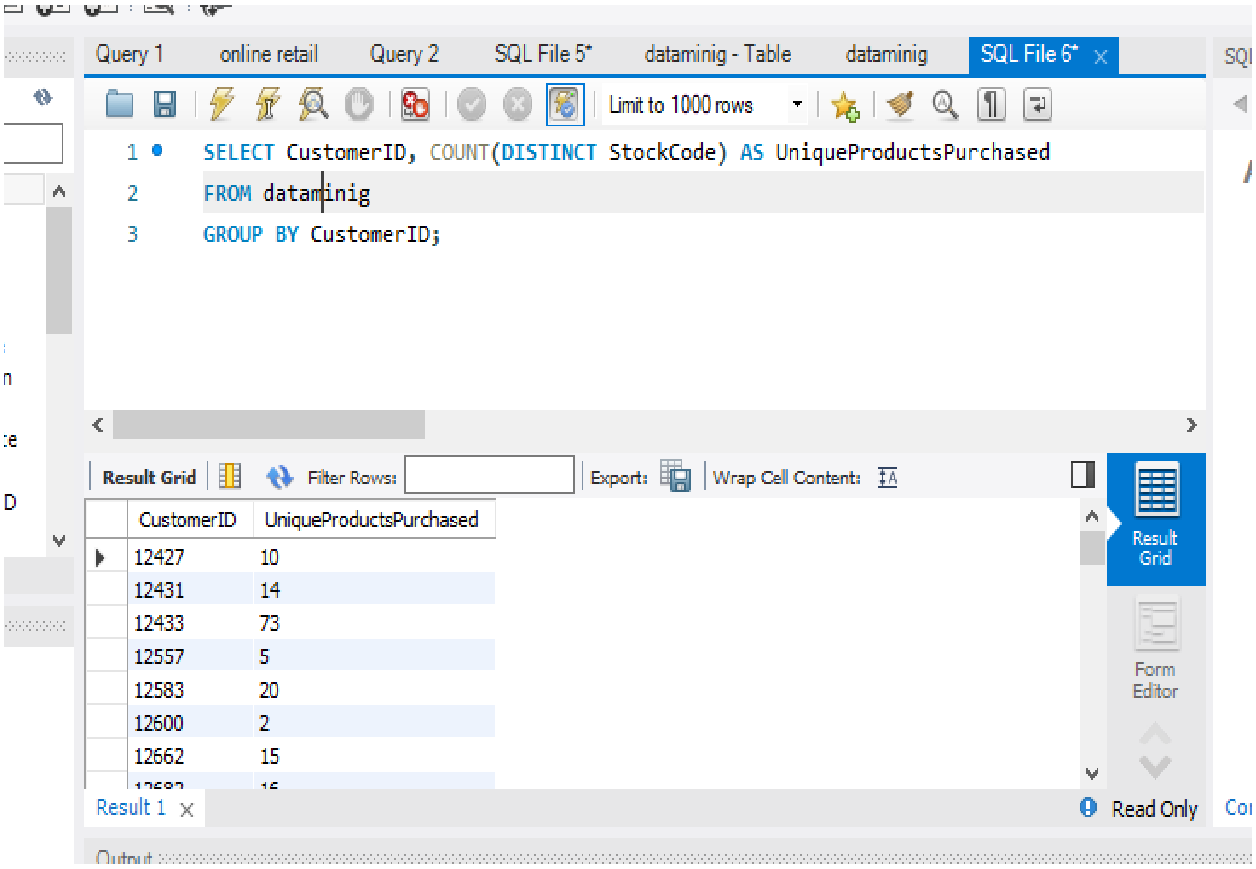
b. What is the distribution of order values across all customers in the dataset?



```sql
Select CustomerID, SUM(Quantity * UnitPrice) AS TotalOrderValue
FROM  dataminig
GROUP BY CustomerID
ORDER BY TotalOrderValue DESC;
```

CustomerID	TotalOrderValue
15061	9407.339999999998
13777	6585.16
17850	5391.210000000009
16029	4271.52
16210	2474.7399999999993
14911	2283.5299999999997
16754	2002.4
12433	1919.1400000000008

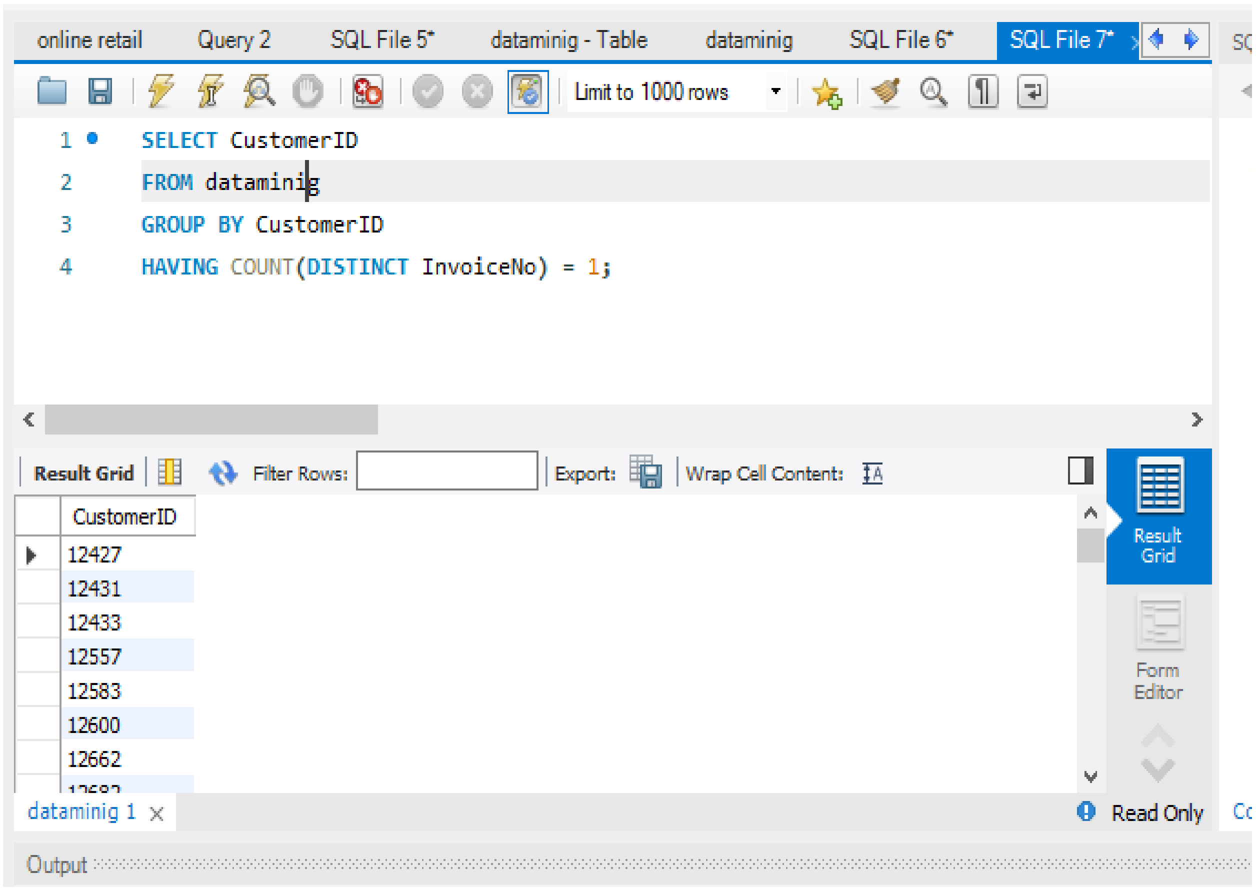
- c. How many unique products has each customer purchased?



```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductsPurchased
FROM dataminig
GROUP BY CustomerID;
```

CustomerID	UniqueProductsPurchased
12427	10
12431	14
12433	73
12557	5
12583	20
12600	2
12662	15

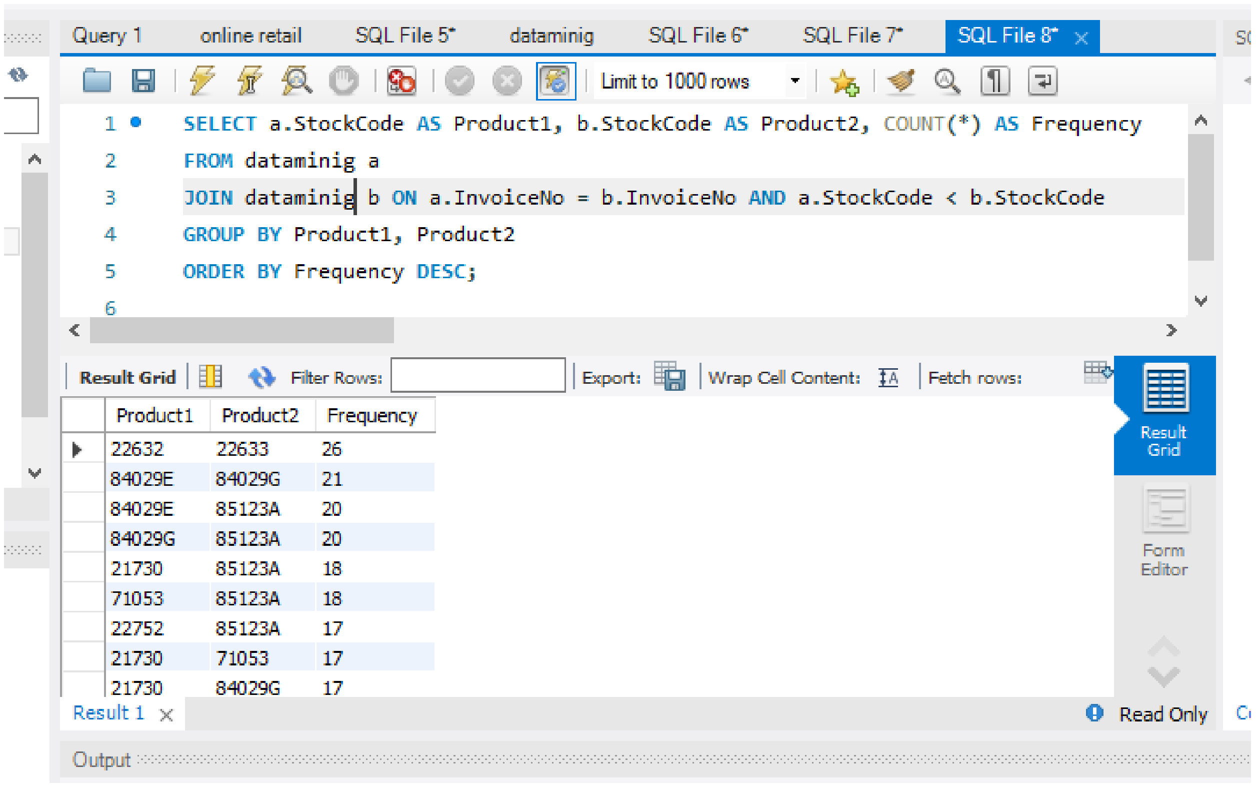
d. Which customers have only made a single purchase from the company ?



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

CustomerID
12427
12431
12433
12557
12583
12600
12662

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

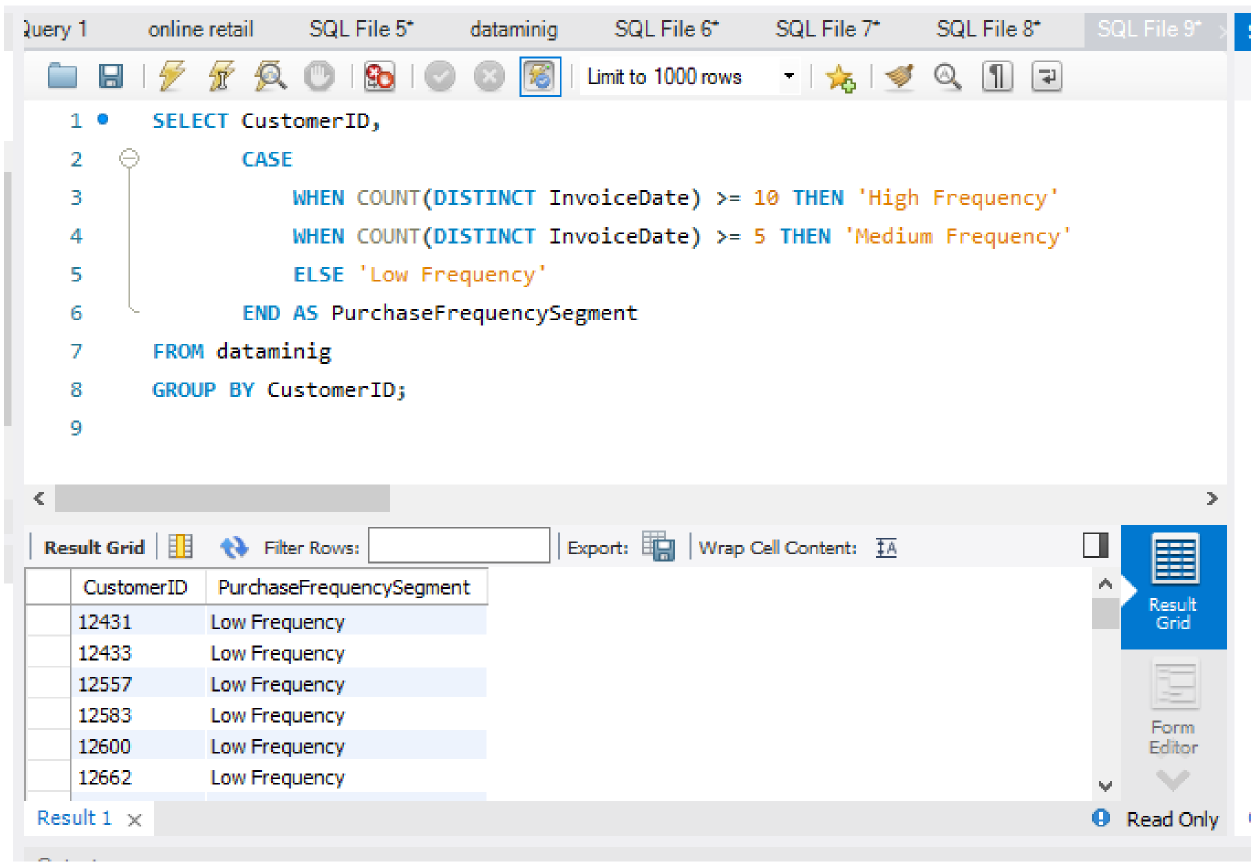


```sql
SELECT a.StockCode AS Product1, b.StockCode AS Product2, COUNT(*) AS Frequency
FROM dataminig a
JOIN dataminig b ON a.InvoiceNo = b.InvoiceNo AND a.StockCode < b.StockCode
GROUP BY Product1, Product2
ORDER BY Frequency DESC;
```

Product1	Product2	Frequency
22632	22633	26
84029E	84029G	21
84029E	85123A	20
84029G	85123A	20
21730	85123A	18
71053	85123A	18
22752	85123A	17
21730	71053	17
21730	84029G	17

Advance Queries

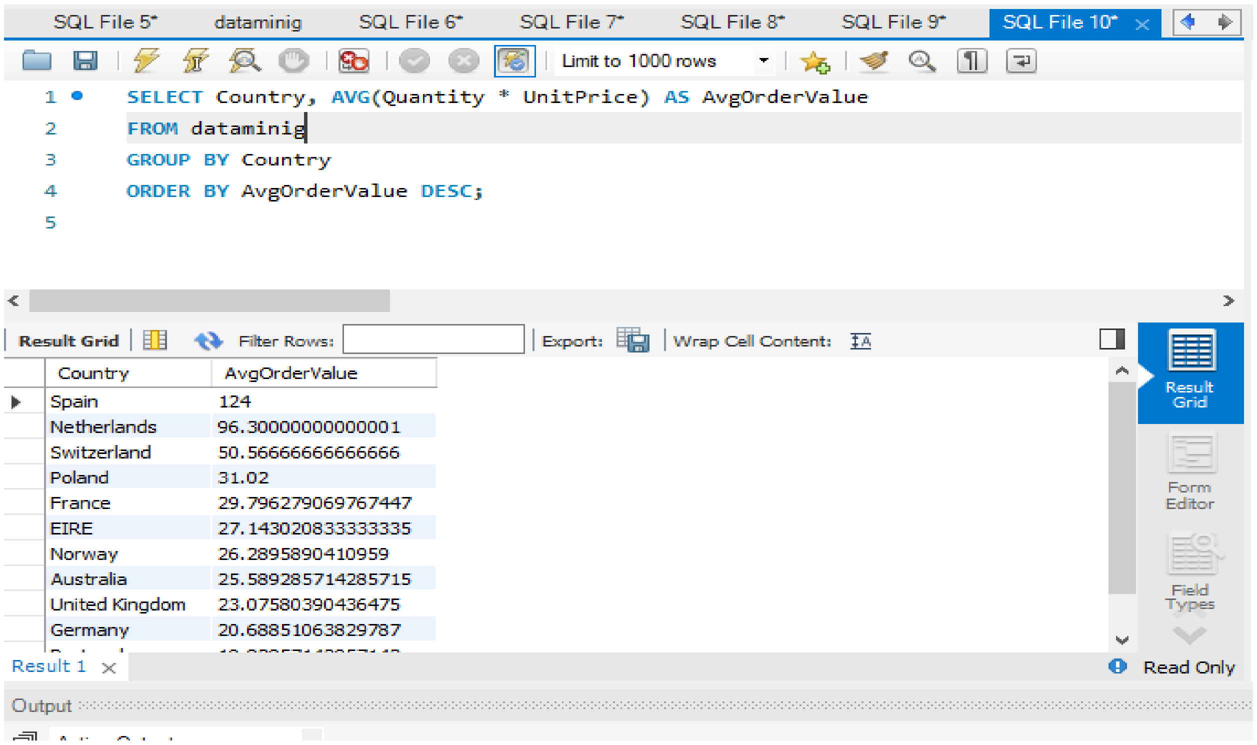
1. Customer Segmentation by Purchase Frequency



```sql
SELECT CustomerID,
        CASE
            WHEN COUNT(DISTINCT InvoiceDate) >= 10 THEN 'High Frequency'
            WHEN COUNT(DISTINCT InvoiceDate) >= 5 THEN 'Medium Frequency'
            ELSE 'Low Frequency'
        END AS PurchaseFrequencySegment
FROM dataminig
GROUP BY CustomerID;
```

CustomerID	PurchaseFrequencySegment
12431	Low Frequency
12433	Low Frequency
12557	Low Frequency
12583	Low Frequency
12600	Low Frequency
12662	Low Frequency

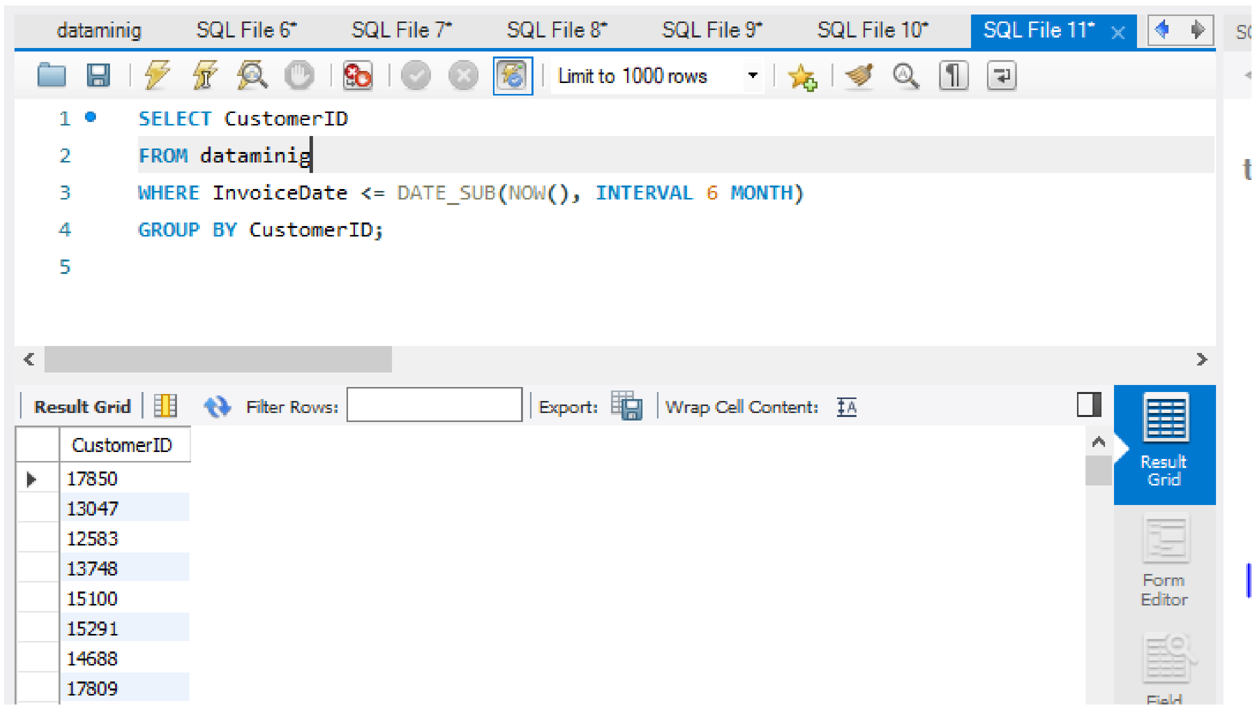
2. Average Order Value by Country



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

Country	AvgOrderValue
Spain	124
Netherlands	96.30000000000001
Switzerland	50.56666666666666
Poland	31.02
France	29.796279069767447
EIRE	27.143020833333335
Norway	26.2895890410959
Australia	25.589285714285715
United Kingdom	23.07580390436475
Germany	20.68851063829787

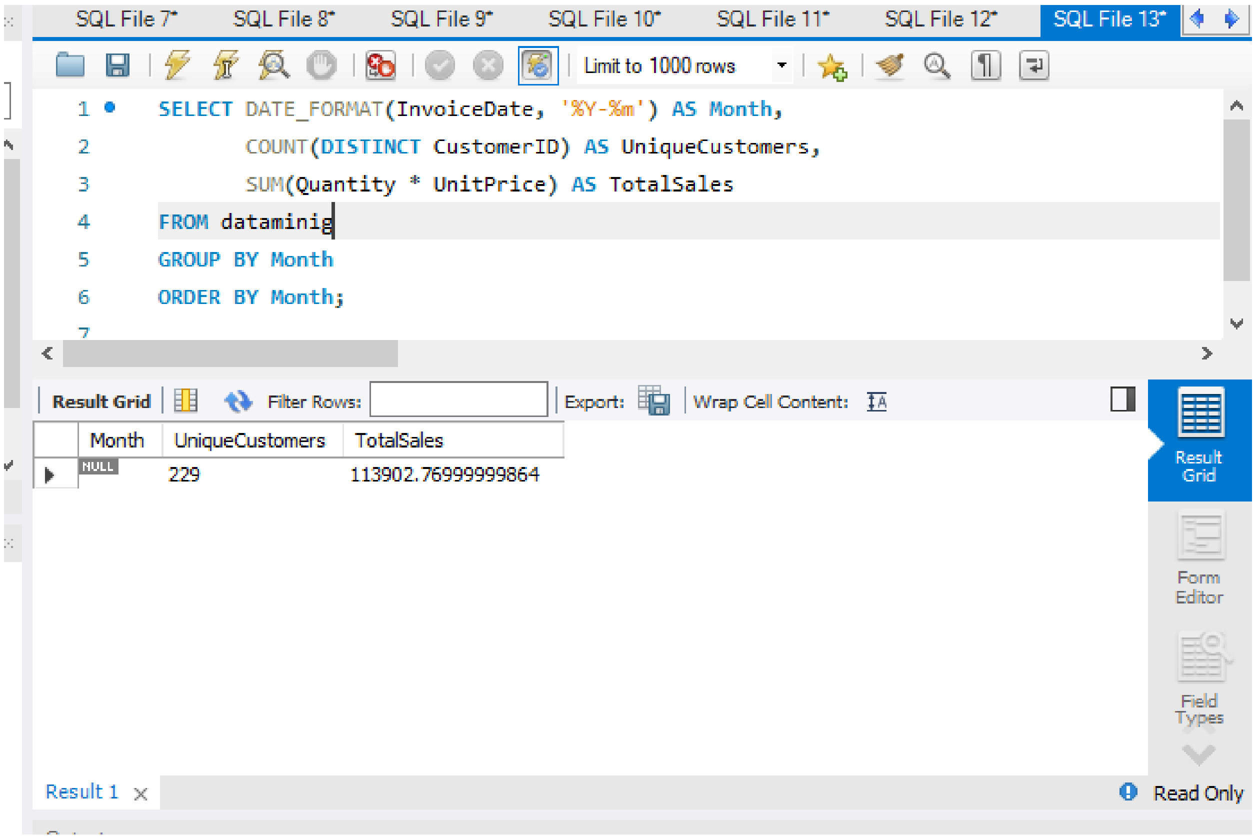
3. Customer Churn Analysis



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

```
CustomerID
17850
13047
12583
13748
15100
15291
14688
17809
```

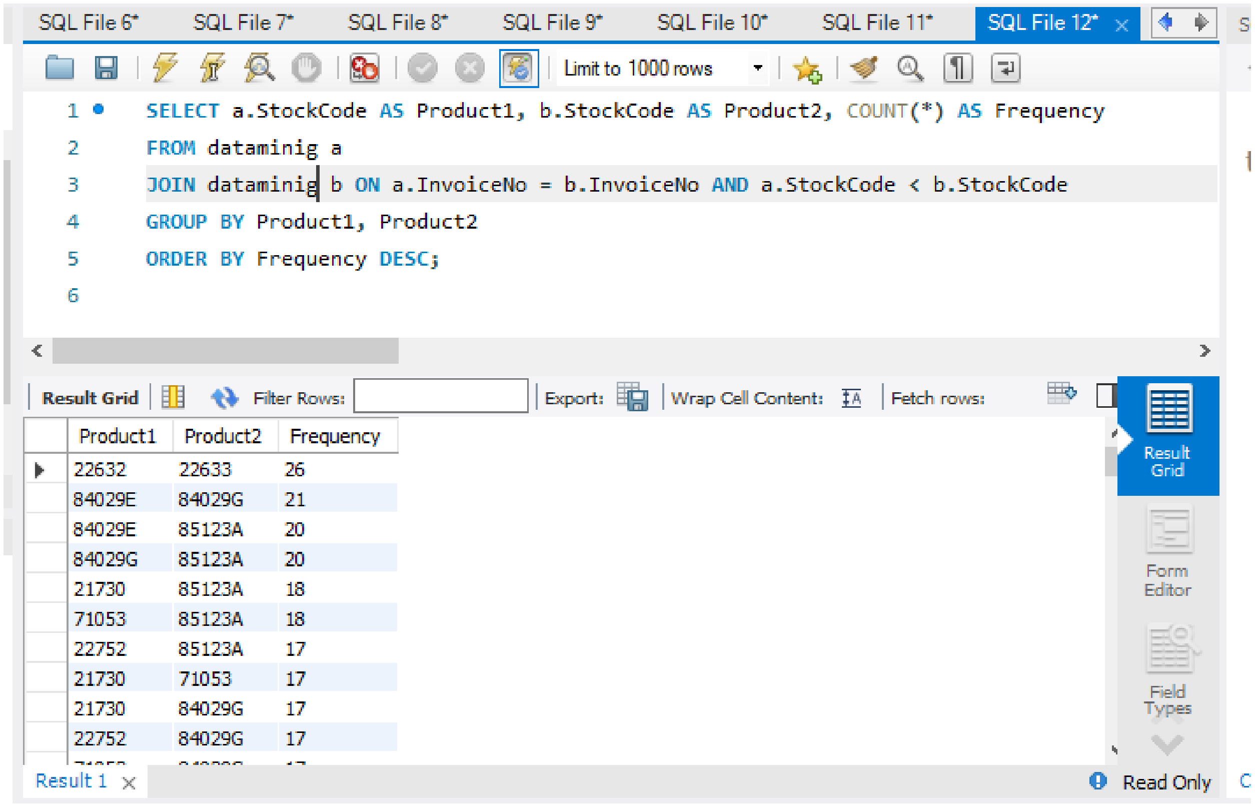
4. Product Affinity Analysis



```sql
SELECT DATE_FORMAT(InvoiceDate, '%Y-%m') AS Month,
       COUNT(DISTINCT CustomerID) AS UniqueCustomers,
       SUM(Quantity * UnitPrice) AS TotalSales
FROM dataminig
GROUP BY Month
ORDER BY Month;
```

Month	UniqueCustomers	TotalSales
NULL	229	113902.76999999864

5. Time-based Analysis



```
SELECT a.StockCode AS Product1, b.StockCode AS Product2, COUNT(*) AS Frequency
FROM dataminig a
JOIN dataminig b ON a.InvoiceNo = b.InvoiceNo AND a.StockCode < b.StockCode
GROUP BY Product1, Product2
ORDER BY Frequency DESC;
```

Product1	Product2	Frequency
22632	22633	26
84029E	84029G	21
84029E	85123A	20
84029G	85123A	20
21730	85123A	18
71053	85123A	18
22752	85123A	17
21730	71053	17
21730	84029G	17
22752	84029G	17