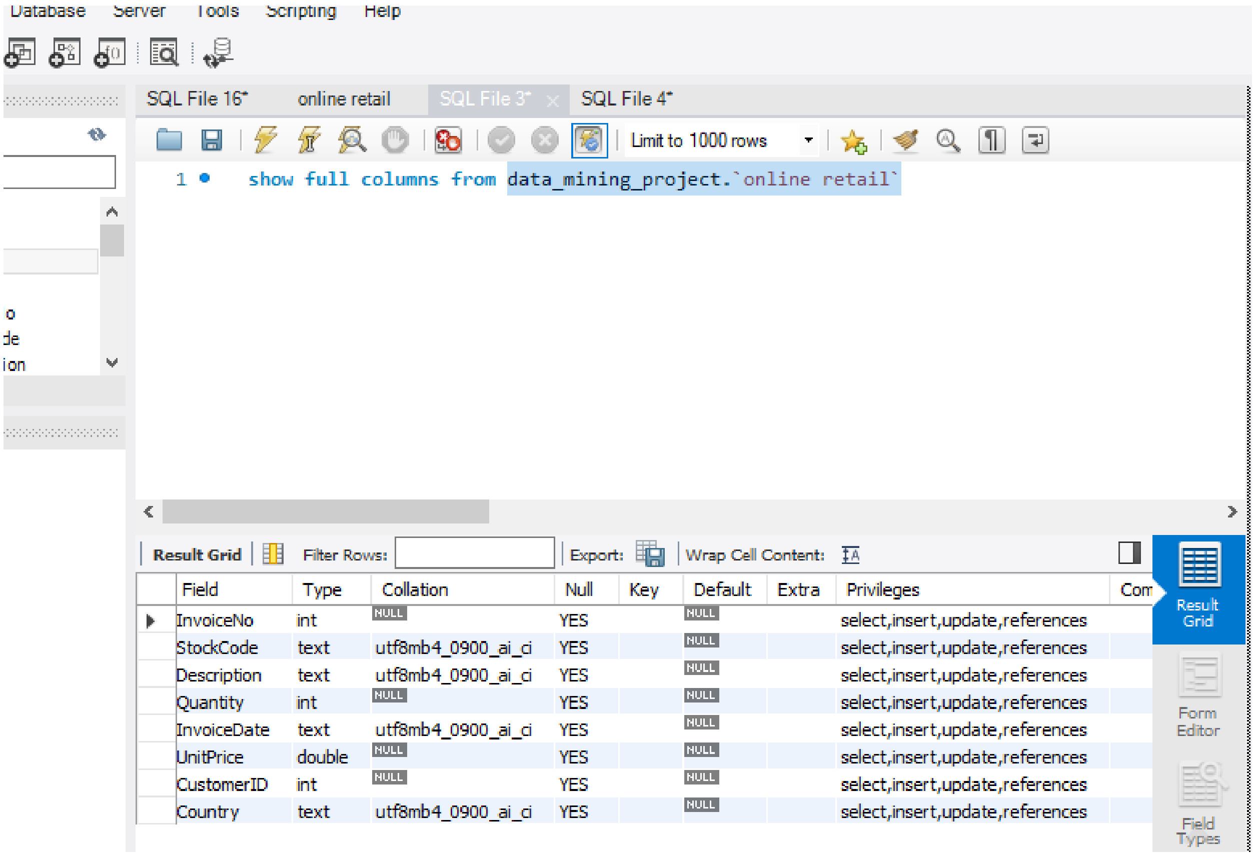
Task 1: Define Meta data in MYSQL workbench?

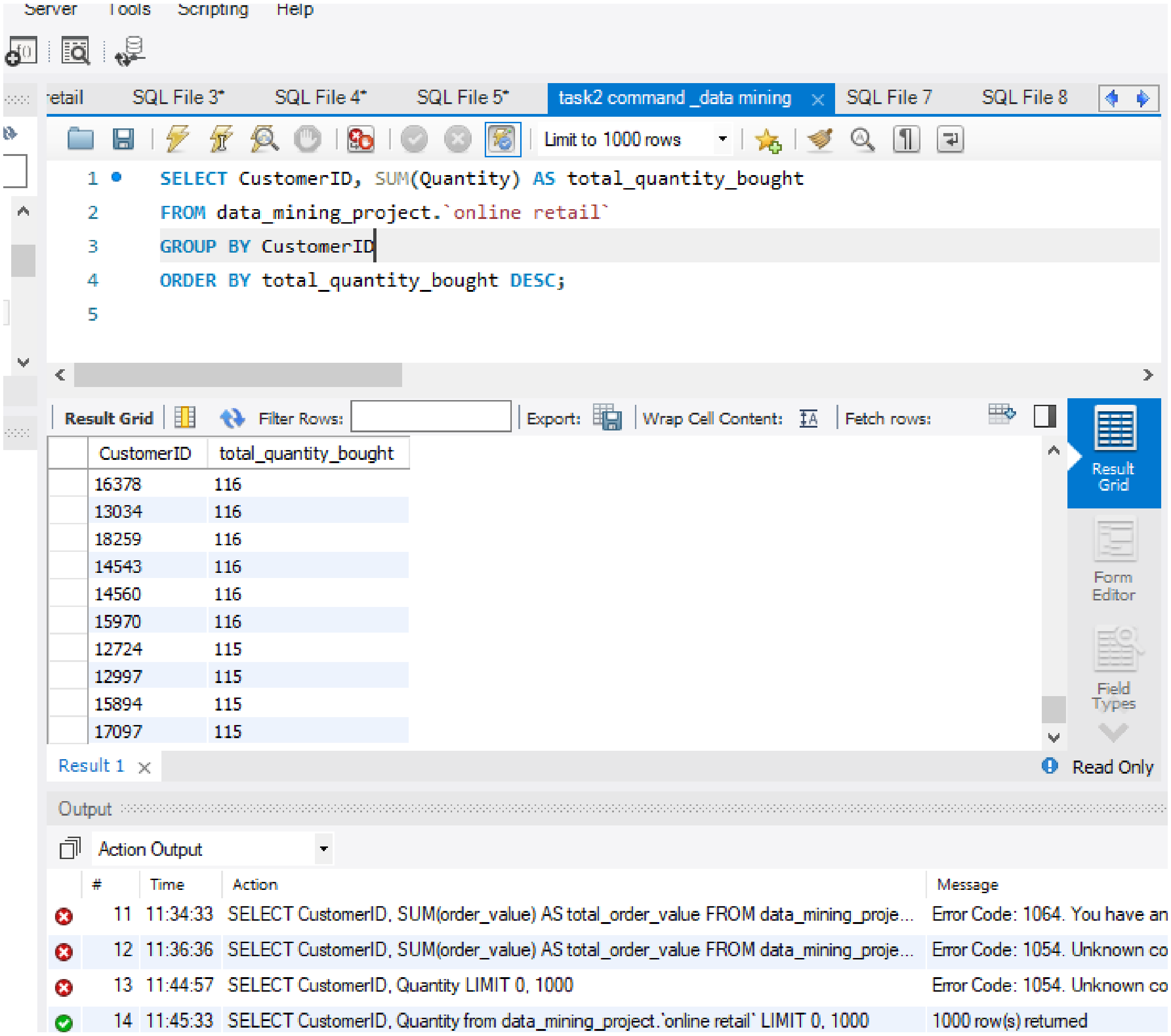


```
show full columns from data_mining_project.`online retail`
```

	Field	Type	Collation	Null	Key	Default	Extra	Privileges	Com
▶	InvoiceNo	int	NULL	YES		NULL		select,insert,update,references	
	StockCode	text	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
	Description	text	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
	Quantity	int	NULL	YES		NULL		select,insert,update,references	
	InvoiceDate	text	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
	UnitPrice	double	NULL	YES		NULL		select,insert,update,references	
	CustomerID	int	NULL	YES		NULL		select,insert,update,references	
	Country	text	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	

Task 2: What is the distribution of order values across all customers in the datasets?

Ans:



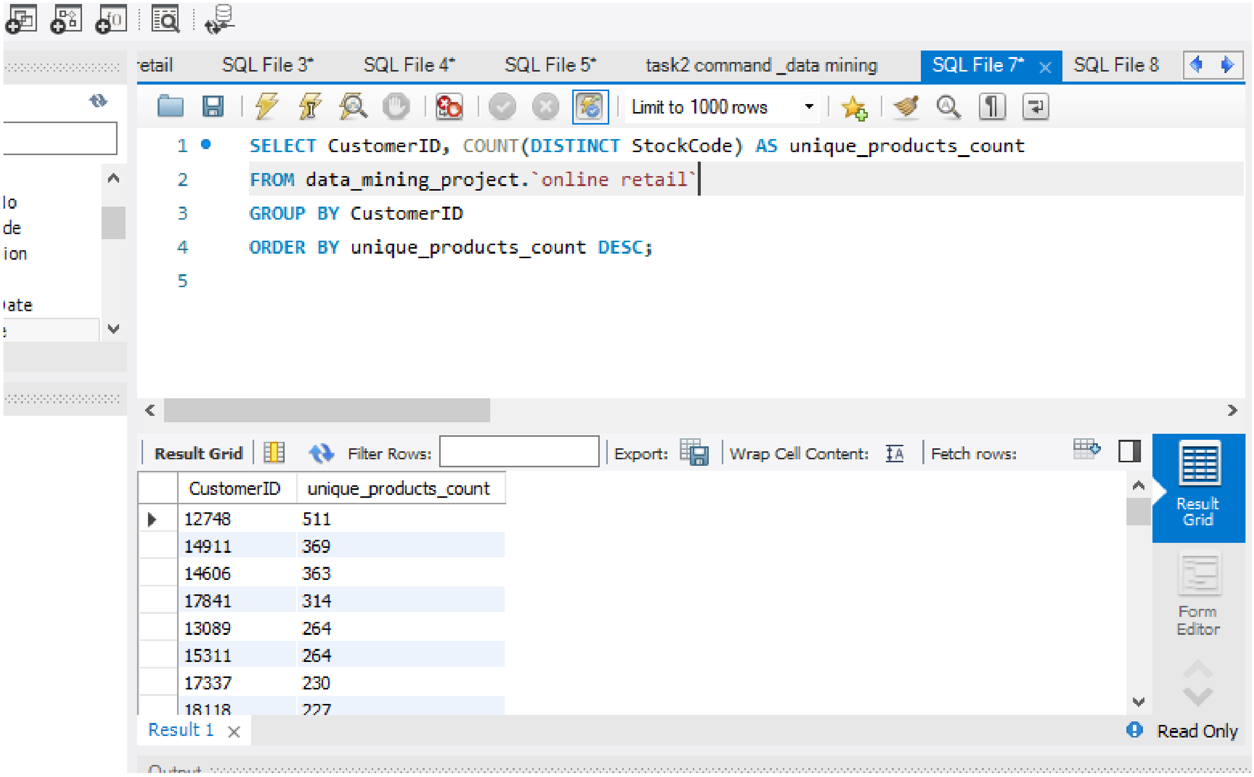
```sql
SELECT CustomerID, SUM(Quantity) AS total_quantity_bought
FROM data_mining_project.`online retail`
GROUP BY CustomerID
ORDER BY total_quantity_bought DESC;
```

CustomerID	total_quantity_bought
16378	116
13034	116
18259	116
14543	116
14560	116
15970	116
12724	115
12997	115
15894	115
17097	115

	#	Time	Action	Message
☒	11	11:34:33	SELECT CustomerID, SUM(order_value) AS total_order_value FROM data_mining_proje...	Error Code: 1064. You have an
☒	12	11:36:36	SELECT CustomerID, SUM(order_value) AS total_order_value FROM data_mining_proje...	Error Code: 1054. Unknown co
☒	13	11:44:57	SELECT CustomerID, Quantity LIMIT 0, 1000	Error Code: 1054. Unknown co
✓	14	11:45:33	SELECT CustomerID, Quantity from data_mining_project.`online retail` LIMIT 0, 1000	1000 row(s) returned

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

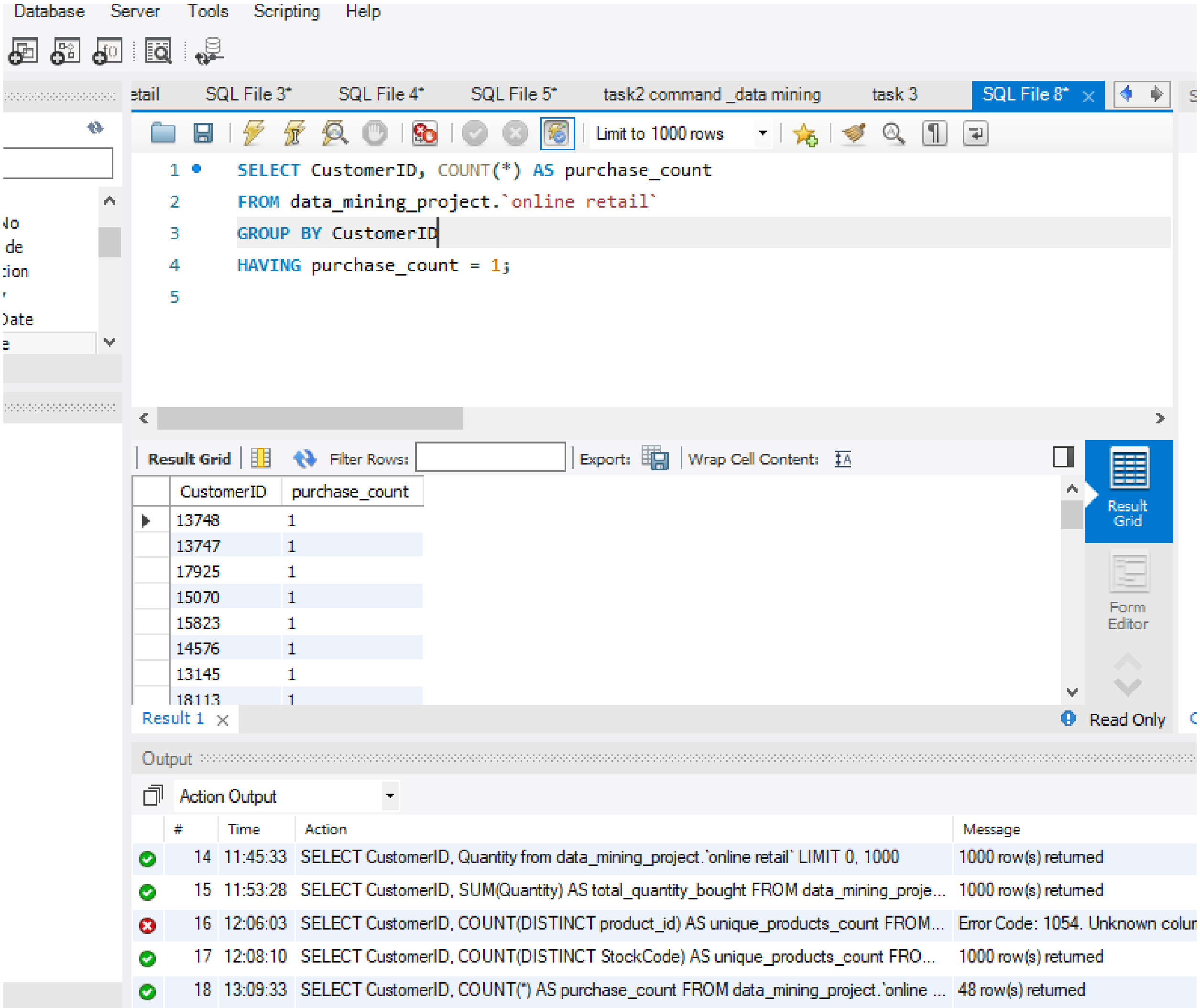
Ans:



```
SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_count
FROM data_mining_project.`online retail`
GROUP BY CustomerID
ORDER BY unique_products_count DESC;
```

CustomerID	unique_products_count
12748	511
14911	369
14606	363
17841	314
13089	264
15311	264
17337	230
18118	227

Task 4: which customer have only made a single purchase from the company?



```sql
SELECT CustomerID, COUNT(*) AS purchase_count
FROM data_mining_project.`online retail`
GROUP BY CustomerID
HAVING purchase_count = 1;
```

CustomerID	purchase_count
13748	1
13747	1
17925	1
15070	1
15823	1
14576	1
13145	1
18113	1

```
#   Time      Action                                                                       Message
14  11:45:33  SELECT CustomerID, Quantity from data_mining_project.`online retail` LIMIT 0, 1000   1000 row(s) returned
15  11:53:28  SELECT CustomerID, SUM(Quantity) AS total_quantity_bought FROM data_mining_proje...  1000 row(s) returned
16  12:06:03  SELECT CustomerID, COUNT(DISTINCT product_id) AS unique_products_count FROM...     Error Code: 1054. Unknown colur
17  12:08:10  SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_count FRO...      1000 row(s) returned
18  13:09:33  SELECT CustomerID, COUNT(*) AS purchase_count FROM data_mining_project.`online ...  48 row(s) returned
```

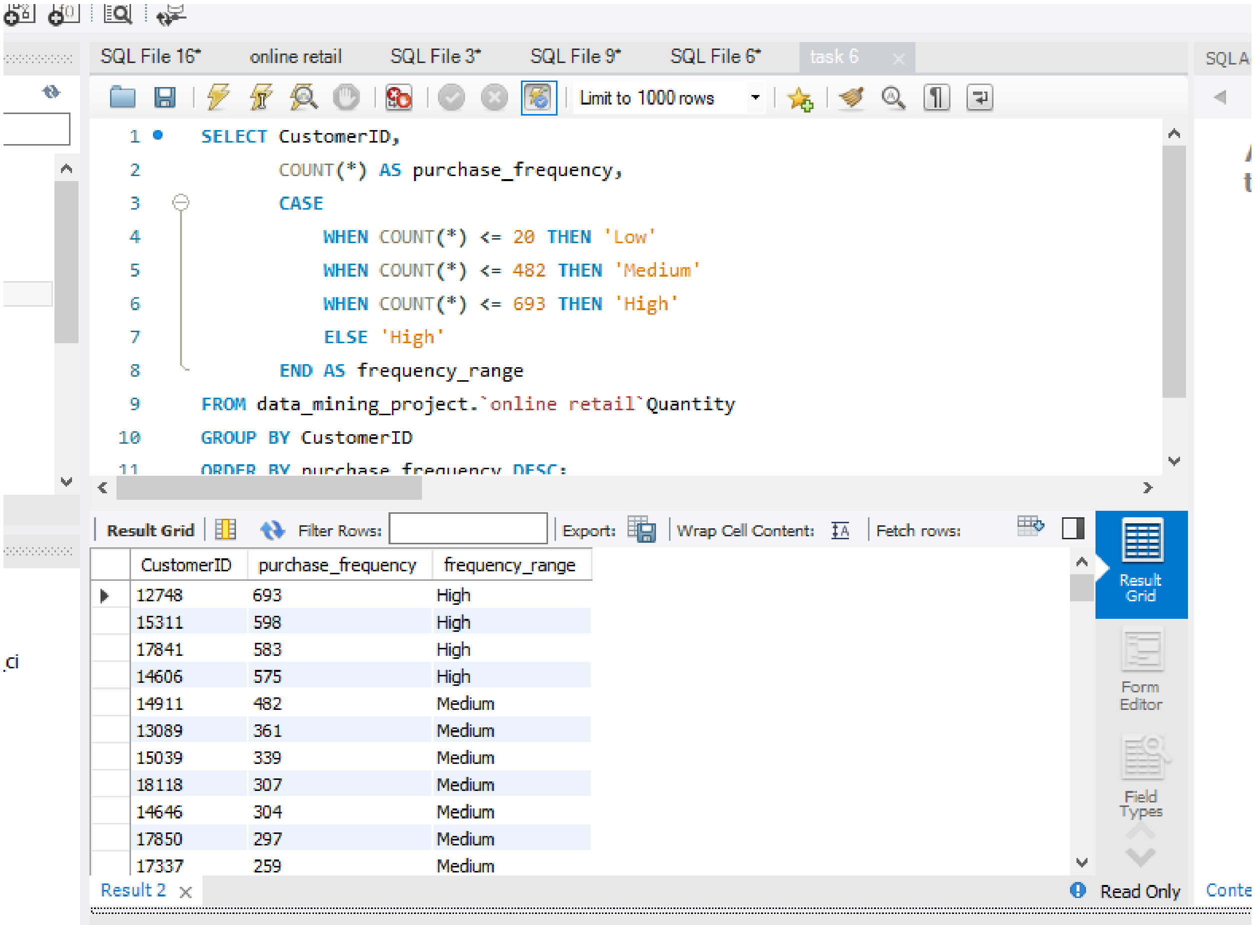
Task 5: Whish product are most commonly purchased together by the customers in the datasets?

Advance Queries

1. Customer Segmentation by Purchase Frequency

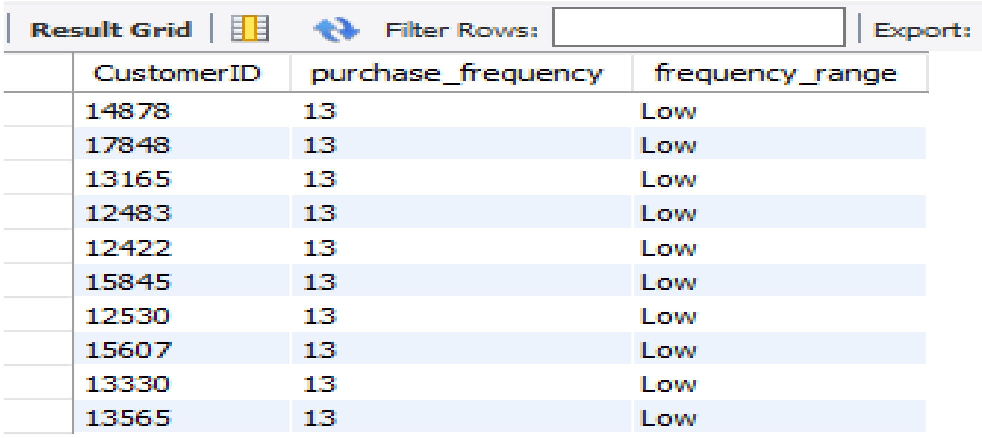
Group customers into segments based on their purchase frequency, such as high, medium, and low frequency customers. This can help you identify your most loyal customers and those who need more attention.

Ans:



```
SELECT CustomerID,
       COUNT(*) AS purchase_frequency,
       CASE
           WHEN COUNT(*) <= 20 THEN 'Low'
           WHEN COUNT(*) <= 482 THEN 'Medium'
           WHEN COUNT(*) <= 693 THEN 'High'
           ELSE 'High'
       END AS frequency_range
FROM data_mining_project.`online retail`Quantity
GROUP BY CustomerID
ORDER BY purchase_frequency DESC;
```

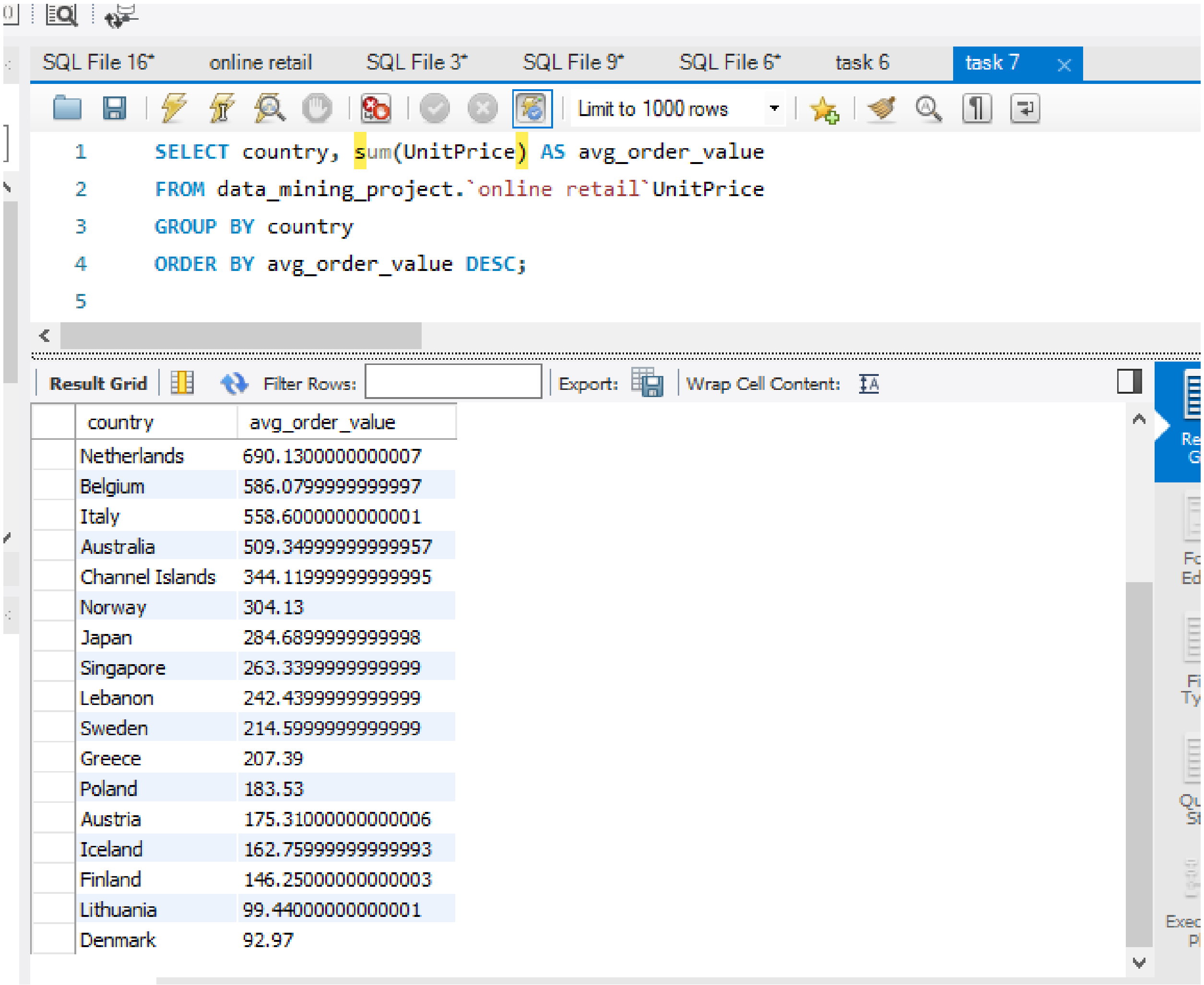
CustomerID	purchase_frequency	frequency_range
12748	693	High
15311	598	High
17841	583	High
14606	575	High
14911	482	Medium
13089	361	Medium
15039	339	Medium
18118	307	Medium
14646	304	Medium
17850	297	Medium
17337	259	Medium



CustomerID	purchase_frequency	frequency_range
14878	13	Low
17848	13	Low
13165	13	Low
12483	13	Low
12422	13	Low
15845	13	Low
12530	13	Low
15607	13	Low
13330	13	Low
13565	13	Low

2. Average Order Value by Country

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



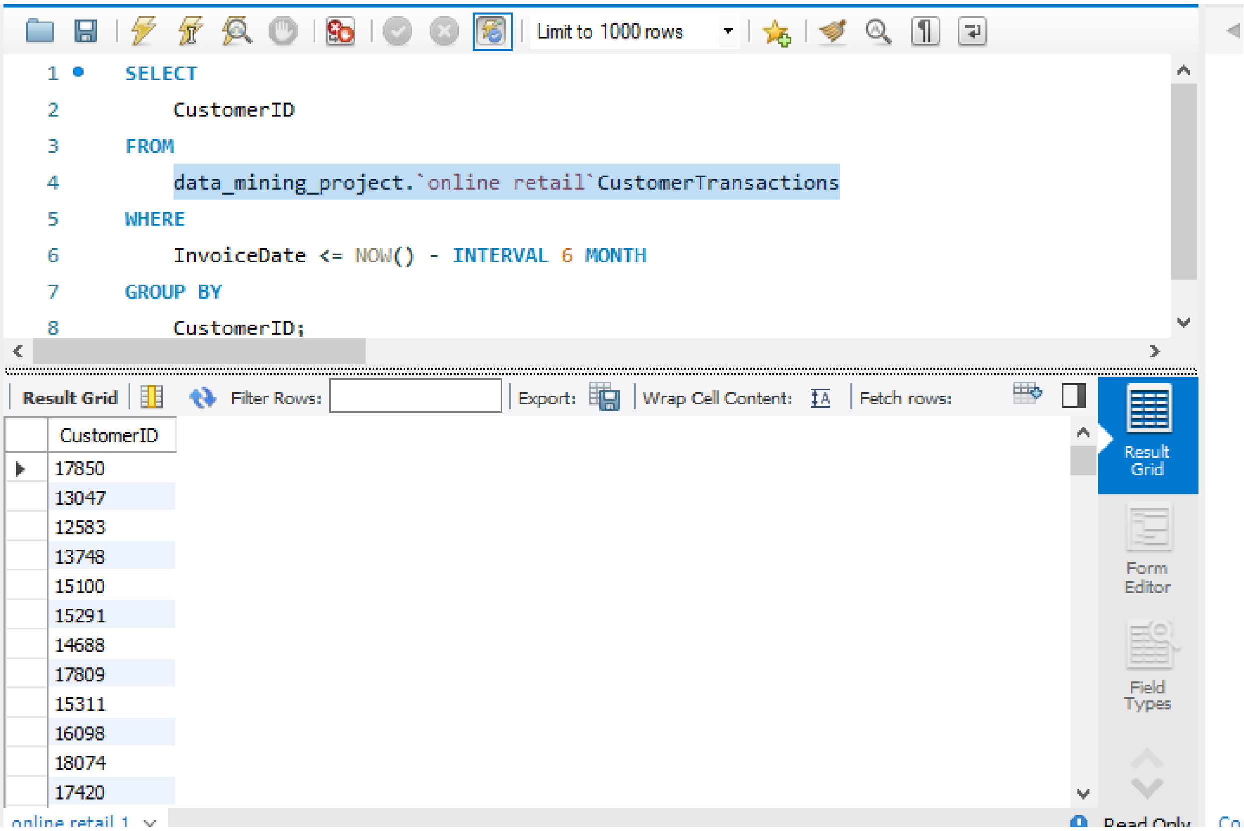
```sql
SELECT country, sum(UnitPrice) AS avg_order_value
FROM data_mining_project.`online retail`UnitPrice
GROUP BY country
ORDER BY avg_order_value DESC;
```

country	avg_order_value
Netherlands	690.1300000000007
Belgium	586.0799999999997
Italy	558.6000000000001
Australia	509.34999999999957
Channel Islands	344.11999999999995
Norway	304.13
Japan	284.6899999999998
Singapore	263.3399999999999
Lebanon	242.4399999999999
Sweden	214.5999999999999
Greece	207.39
Poland	183.53
Austria	175.31000000000006
Iceland	162.75999999999993
Finland	146.25000000000003
Lithuania	99.44000000000001
Denmark	92.97

3. Customer Churn Analysis

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

Ans:



```sql
SELECT
    CustomerID
FROM
    data_mining_project.`online retail`CustomerTransactions
WHERE
    InvoiceDate <= NOW() - INTERVAL 6 MONTH
GROUP BY
    CustomerID;
```

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

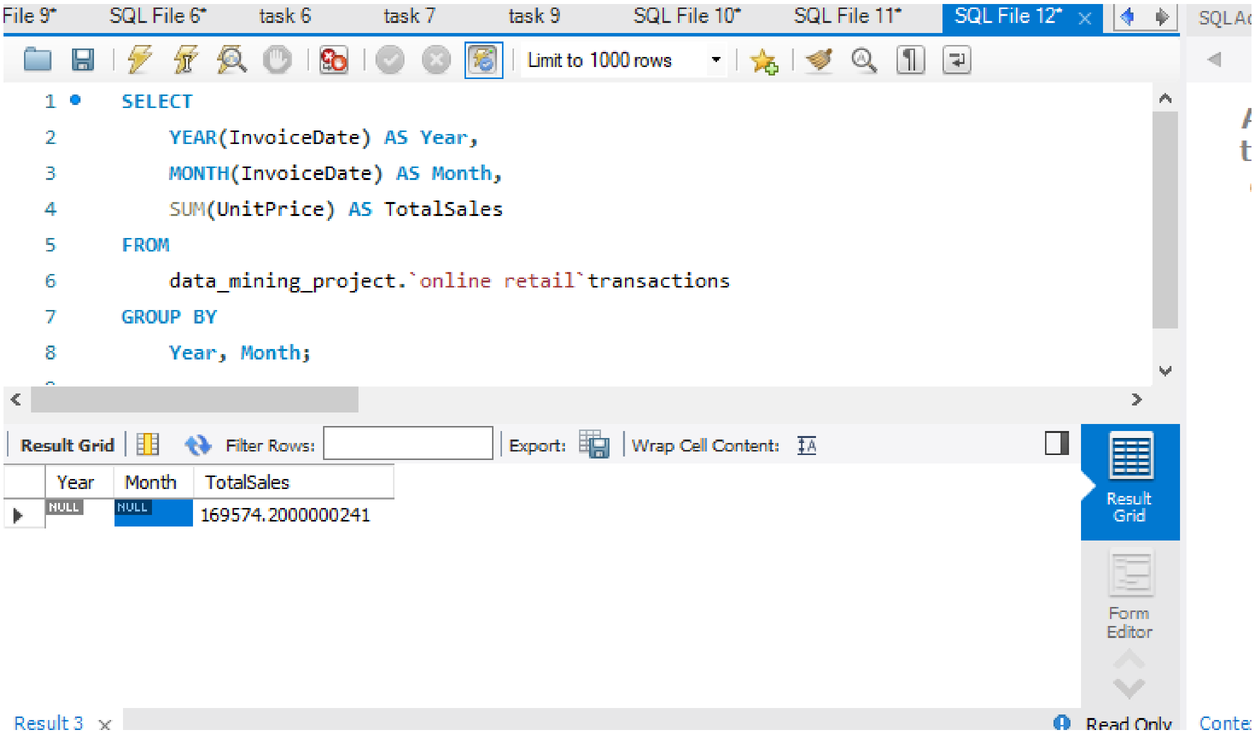
4. Product Affinity Analysis

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

Ans:



```
SELECT
    YEAR(InvoiceDate) AS Year,
    MONTH(InvoiceDate) AS Month,
    SUM(UnitPrice) AS TotalSales
FROM
    data_mining_project.`online retail`transactions
GROUP BY
    Year, Month;
```

Year	Month	TotalSales
NULL	NULL	169574.2000000241