

Task 1: Define Meta data in MYSQL workbench?

The screenshot shows the MySQL Workbench interface. The SQL editor contains the command: `show full columns from data_mining_project.`online retail``. The results are displayed in the 'Result Grid' tab, showing the following table metadata:

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
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:

Server Tools Scripting Help

task2 command_data mining

```

1 • SELECT CustomerID, SUM(Quantity) AS total_quantity_bought
2 FROM data_mining_project.`online retail`
3 GROUP BY CustomerID
4 ORDER BY total_quantity_bought DESC;
5

```

Limit to 1000 rows

Result Grid

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

Result 1

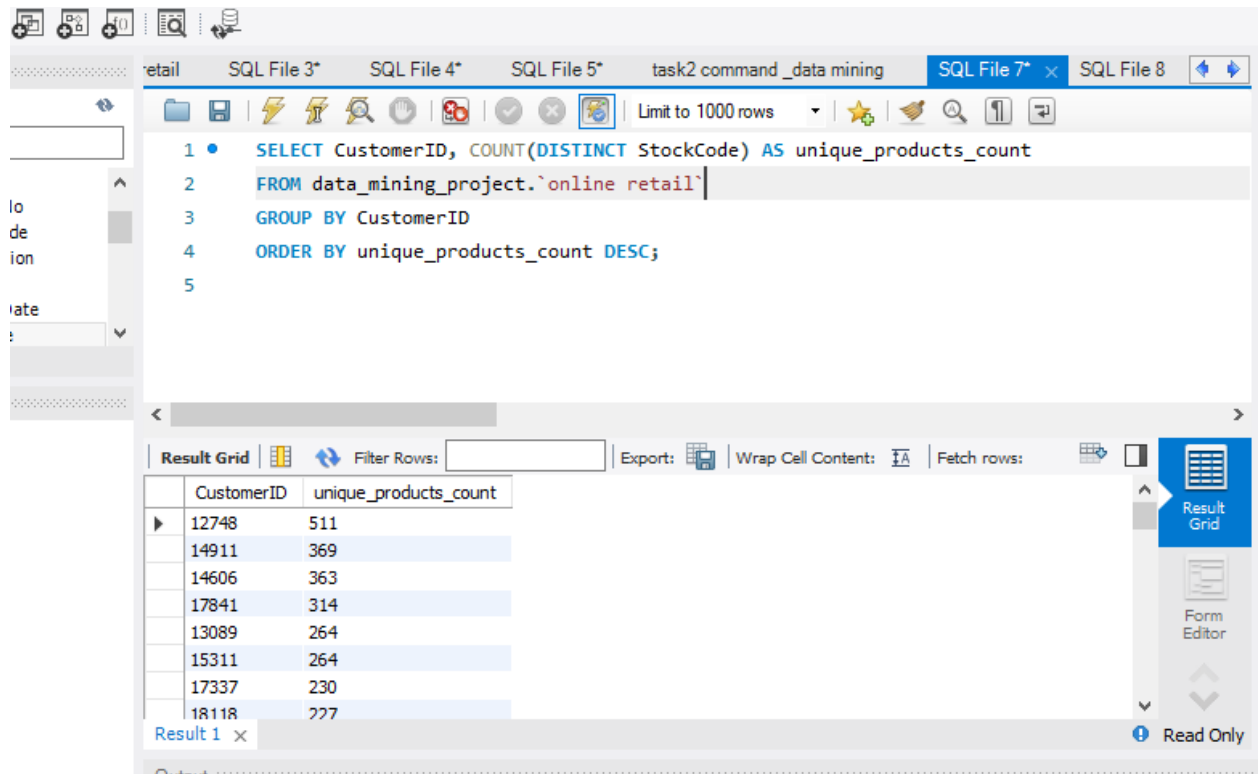
Output

Action Output

#	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:



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
1 • SELECT CustomerID, COUNT(DISTINCT StockCode) AS unique_products_count
2 FROM data_mining_project.`online retail`
3 GROUP BY CustomerID
4 ORDER BY unique_products_count DESC;
5
```

The results grid displays the following data:

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

The interface includes a toolbar with various icons, a sidebar with a tree view, and a bottom status bar indicating 'Read Only'.

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

Database Server Tools Scripting Help

SQL File 3* SQL File 4* SQL File 5* task2 command_data mining task 3 SQL File 8* x

Limit to 1000 rows

```

1 • SELECT CustomerID, COUNT(*) AS purchase_count
2   FROM data_mining_project.`online retail`
3  GROUP BY CustomerID
4  HAVING purchase_count = 1;
5

```

Result Grid

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

Result 1 x

Output

Action Output

#	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 colour
✓ 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: Which 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:

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is as follows:

```
1 • SELECT CustomerID,  
2     COUNT(*) AS purchase_frequency,  
3     CASE  
4         WHEN COUNT(*) <= 20 THEN 'Low'  
5         WHEN COUNT(*) <= 482 THEN 'Medium'  
6         WHEN COUNT(*) <= 693 THEN 'High'  
7         ELSE 'High'  
8     END AS frequency_range  
9 FROM data_mining_project.`online retail`Quantity  
10 GROUP BY CustomerID  
11 ORDER BY purchase_frequency DESC;
```

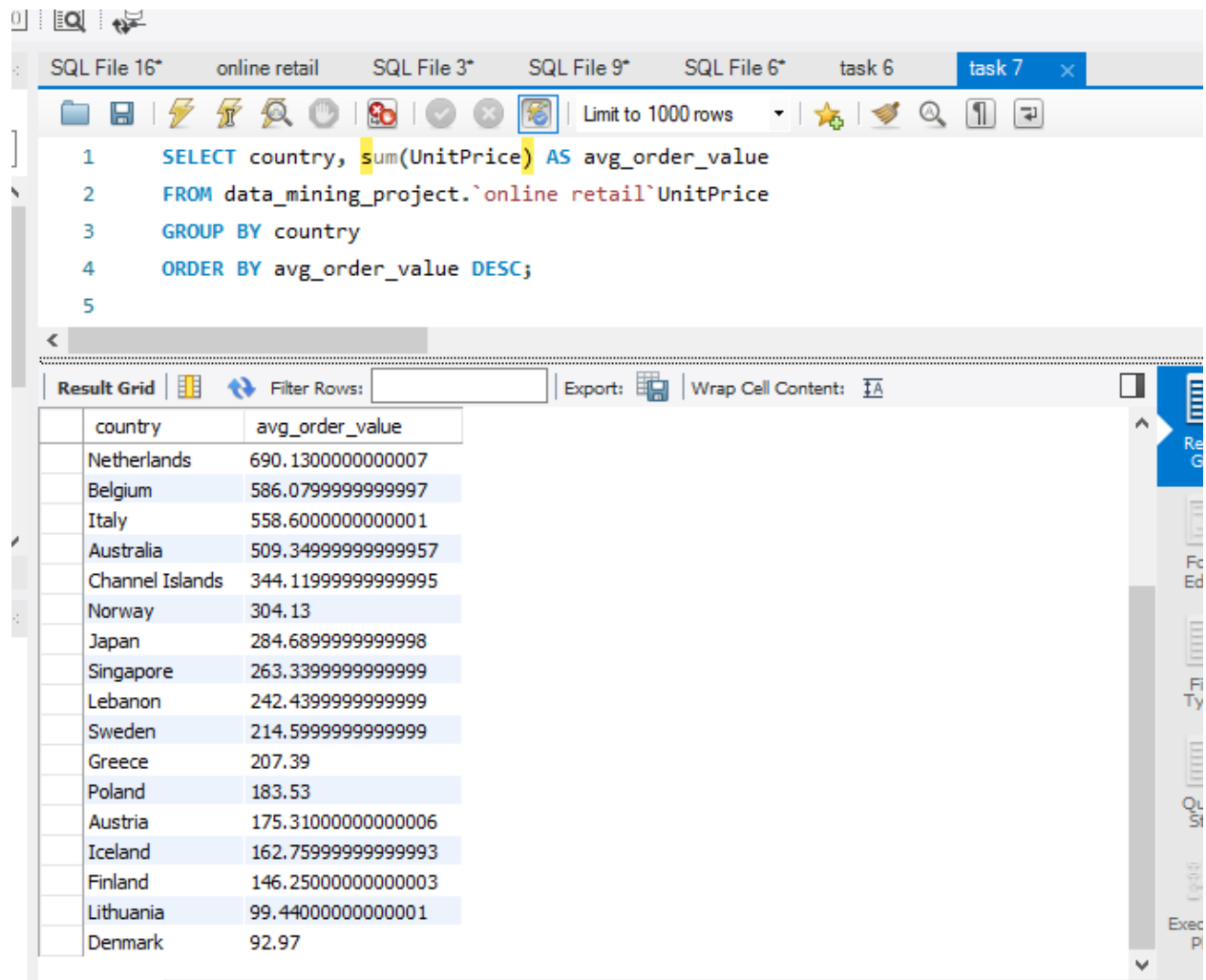
The result grid displays the following data:

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

Result Grid			
Filter Rows:			
Export:			
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.



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 SELECT country, sum(UnitPrice) AS avg_order_value
2 FROM data_mining_project.`online retail`UnitPrice
3 GROUP BY country
4 ORDER BY avg_order_value DESC;
5
```

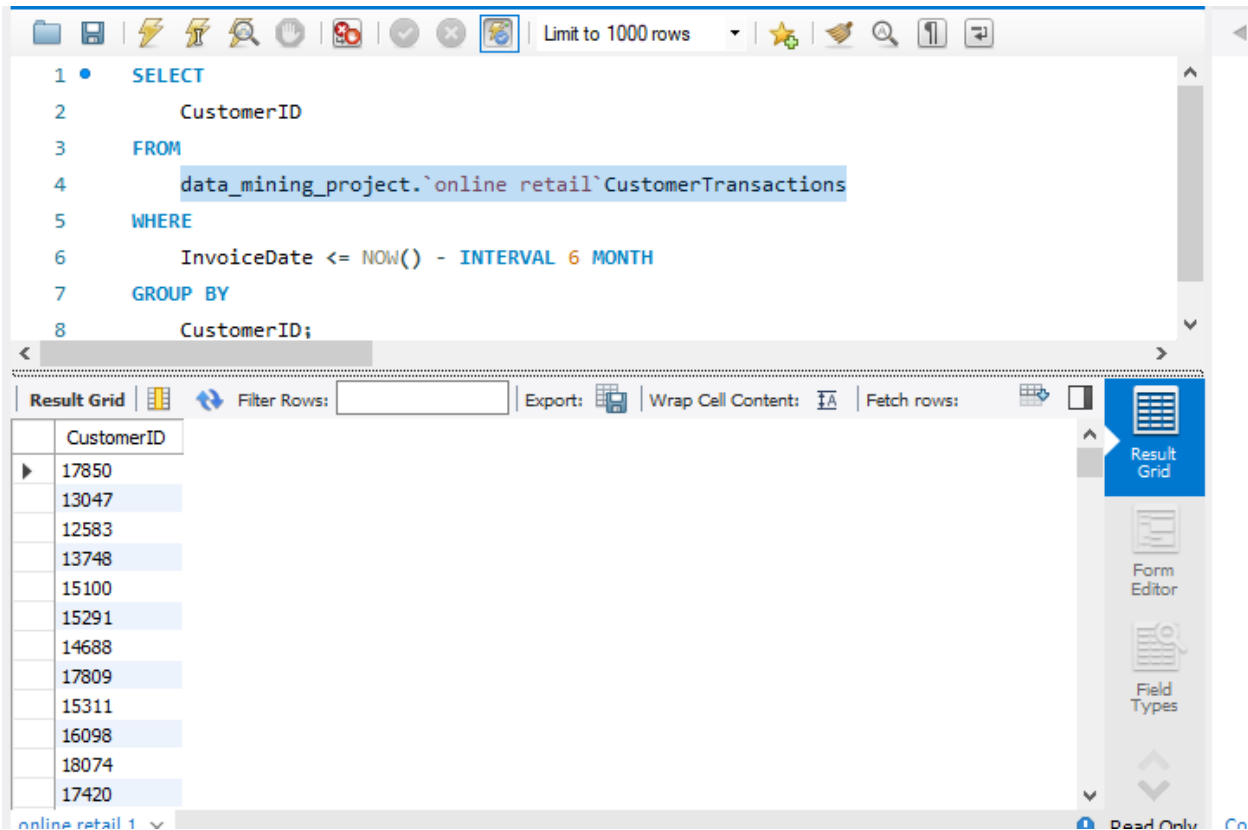
The results grid displays the following data:

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	263.33999999999999
Lebanon	242.43999999999999
Sweden	214.59999999999999
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:



The screenshot shows a SQL query editor with the following query:

```
1 SELECT
2   CustomerID
3 FROM
4   data_mining_project.`online retail`CustomerTransactions
5 WHERE
6   InvoiceDate <= NOW() - INTERVAL 6 MONTH
7 GROUP BY
8   CustomerID;
```

Below the query editor, the 'Result Grid' is displayed, showing a list of CustomerIDs. The first few rows are highlighted in blue:

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:



```
1 • SELECT
2     YEAR(InvoiceDate) AS Year,
3     MONTH(InvoiceDate) AS Month,
4     SUM(UnitPrice) AS TotalSales
5 FROM
6     data_mining_project.`online retail`transactions
7 GROUP BY
8     Year, Month;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Result Grid
Year	Month	TotalSales				
NULL	NULL	169574.2000000241				