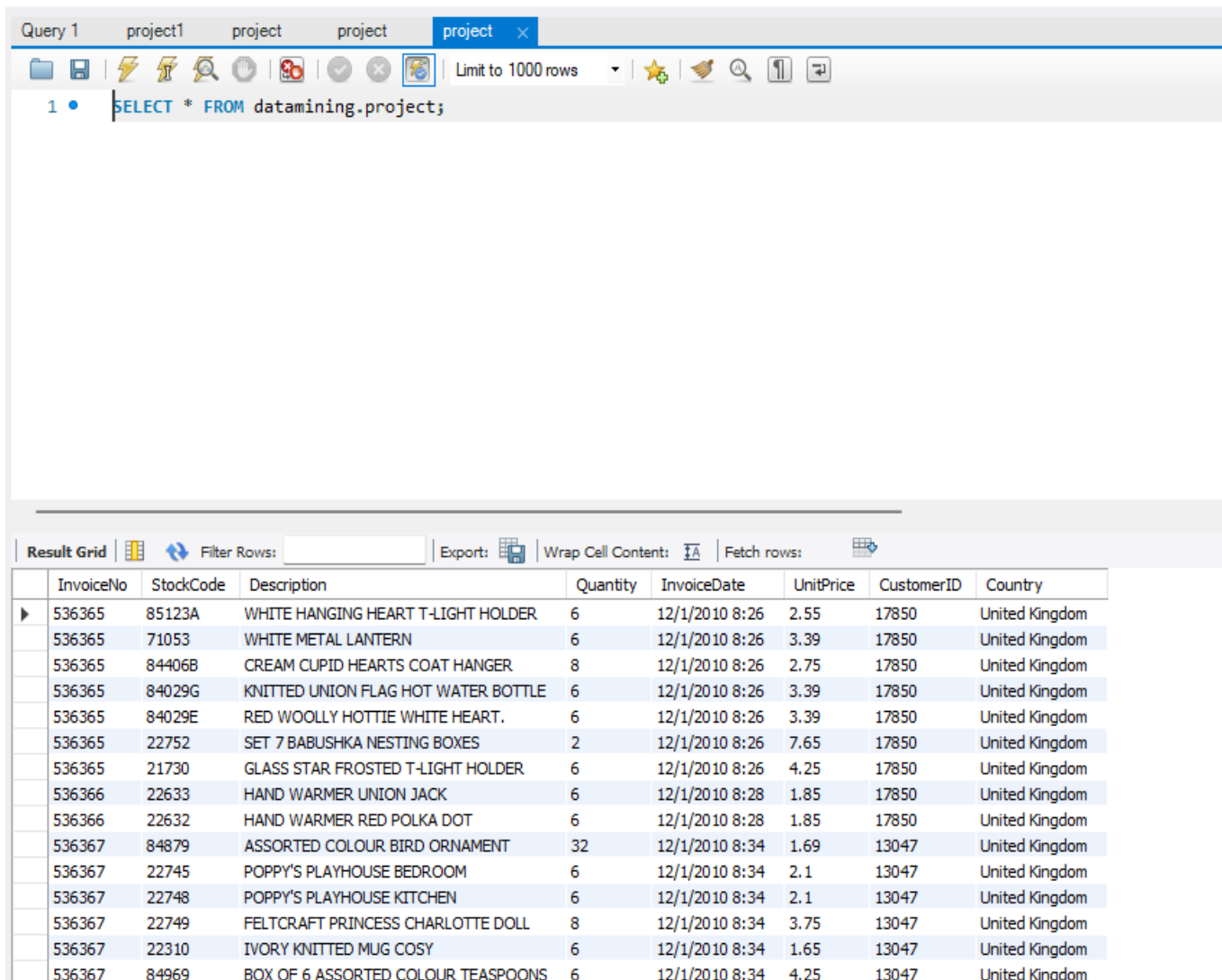


# Project Datamining

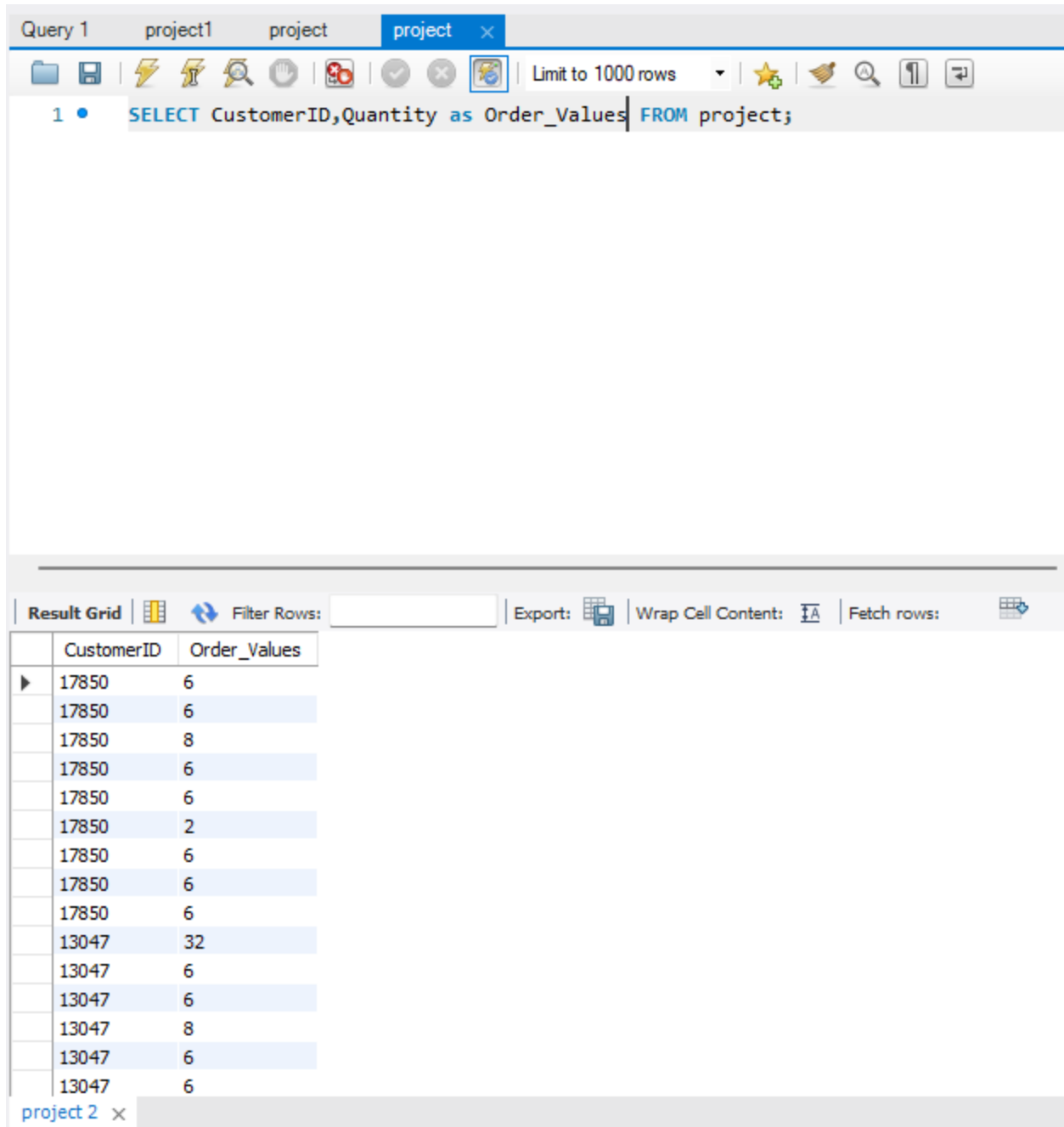


The screenshot shows a database query tool interface. At the top, there are tabs for 'Query 1', 'project1', 'project', and 'project'. Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The SQL query editor shows the query: `SELECT * FROM datamining.project;`. Below the query editor is a 'Result Grid' section with a toolbar containing 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The result grid displays a table with 9 columns: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country. The data is sorted by InvoiceNo in ascending order.

	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
	536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom
	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	12/1/2010 8:34	2.1	13047	United Kingdom
	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	12/1/2010 8:34	2.1	13047	United Kingdom
	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12/1/2010 8:34	3.75	13047	United Kingdom
	536367	22310	IVORY KNITTED MUG COSY	6	12/1/2010 8:34	1.65	13047	United Kingdom
	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12/1/2010 8:34	4.25	13047	United Kingdom

1. In above picture the meta data of the customer is shown.

# Project Datamining



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and settings. The query text is: `SELECT CustomerID, Quantity as Order_Values FROM project;`. Below the query, the results are displayed in a table with two columns: CustomerID and Order\_Values. The results show multiple rows for different customer IDs, with the number of orders (Quantity) listed for each. The table is titled 'Result Grid' and includes options for filtering, exporting, and wrapping cell content.

CustomerID	Order_Values
17850	6
17850	6
17850	8
17850	6
17850	6
17850	2
17850	6
17850	6
17850	6
13047	32
13047	6
13047	6
13047	8
13047	6
13047	6

2. In this picture the number of orders made by a customer is shown.

# Project Datamining

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

```
1 • SELECT customerid, COUNT(DISTINCT stockcode) AS num_unique_products
2 FROM datamining.project
3 GROUP BY customerid
4 ORDER BY num_unique_products DESC;
```

The results grid displays the following data:

customerid	num_unique_products
3323	3323
14911	1131
12748	1081
17841	973
14606	683
14298	661
13089	484
14156	478
14769	477
15311	445

3. In the above picture, the number of unique products purchased by a customer is shown.

# Project Datamining

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

```
1 • SELECT customerid AS Customers, count(distinct quantity) as Num_Of_Purchases FROM datamining.project where quantity  
2 group by customerid;
```

The result grid displays the following data:

Customers	Num_Of_Purchases
12348	1
12350	1
12352	1
12354	1
12359	1
12361	1
12362	1
12365	1
12370	1

The interface includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' status indicator at the bottom right.

4. In this picture the number of customers who made a single purchase.

# Project Datamining

The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

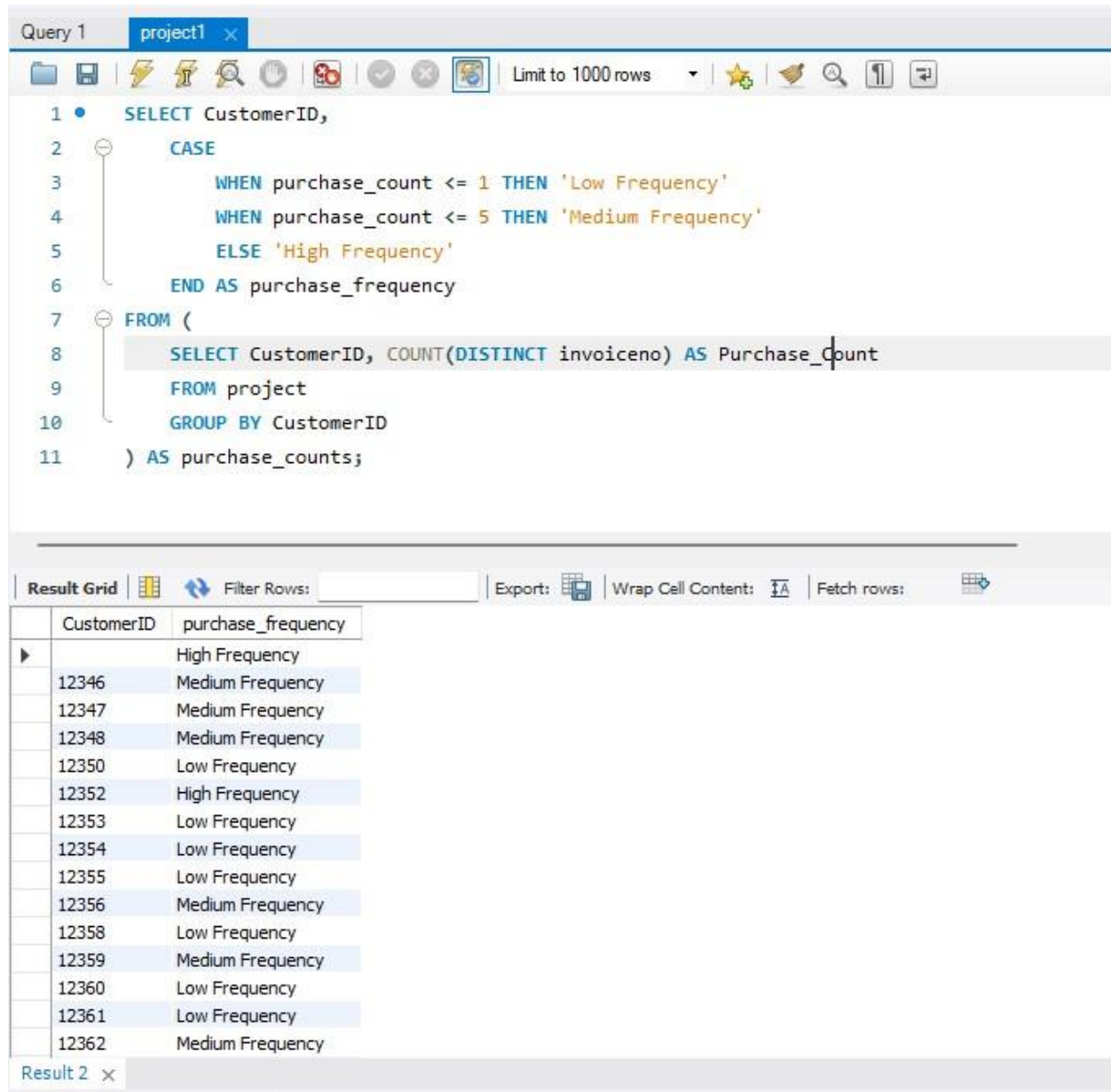
```
1 • SELECT od1.StockCode AS product1, od2.StockCode AS product2, COUNT(*) AS purchase_count
2 FROM project1 od1
3 JOIN project1 od2 ON od1.InvoiceNo = od2.InvoiceNo AND od1.StockCode < od2.StockCode
4 GROUP BY od1.StockCode, od2.StockCode
5 ORDER BY purchase_count DESC
6 LIMIT 10;
```

Below the query editor is the 'Result Grid' tab, which displays the results of the query. The grid has four columns: 'product1', 'product2', and 'purchase\_count'. The results are sorted by 'purchase\_count' in descending order. The first row shows product1 '22726' and product2 '22727' with a purchase count of 67. The last row shows product1 '22745' and product2 '22748' with a purchase count of 43. On the right side of the grid, there are buttons for 'Result Grid', 'Form Editor', and 'Read Only'.

product1	product2	purchase_count
22726	22727	67
22086	22910	65
22632	22633	48
84029E	84029G	46
22865	22866	45
22727	22730	44
21733	85123A	44
22745	22748	43

5. In this picture the products that are most commonly purchased together by customers are shown.

# Project Datamining



Query 1 project1 x

Limit to 1000 rows

```
1 • SELECT CustomerID,  
2 CASE  
3 WHEN purchase_count <= 1 THEN 'Low Frequency'  
4 WHEN purchase_count <= 5 THEN 'Medium Frequency'  
5 ELSE 'High Frequency'  
6 END AS purchase_frequency  
7 FROM (  
8 SELECT CustomerID, COUNT(DISTINCT invoiceno) AS Purchase_Count  
9 FROM project  
10 GROUP BY CustomerID  
11 ) AS purchase_counts;
```

Result Grid

Filter Rows: Export: Wrap Cell Content: Fetch rows:

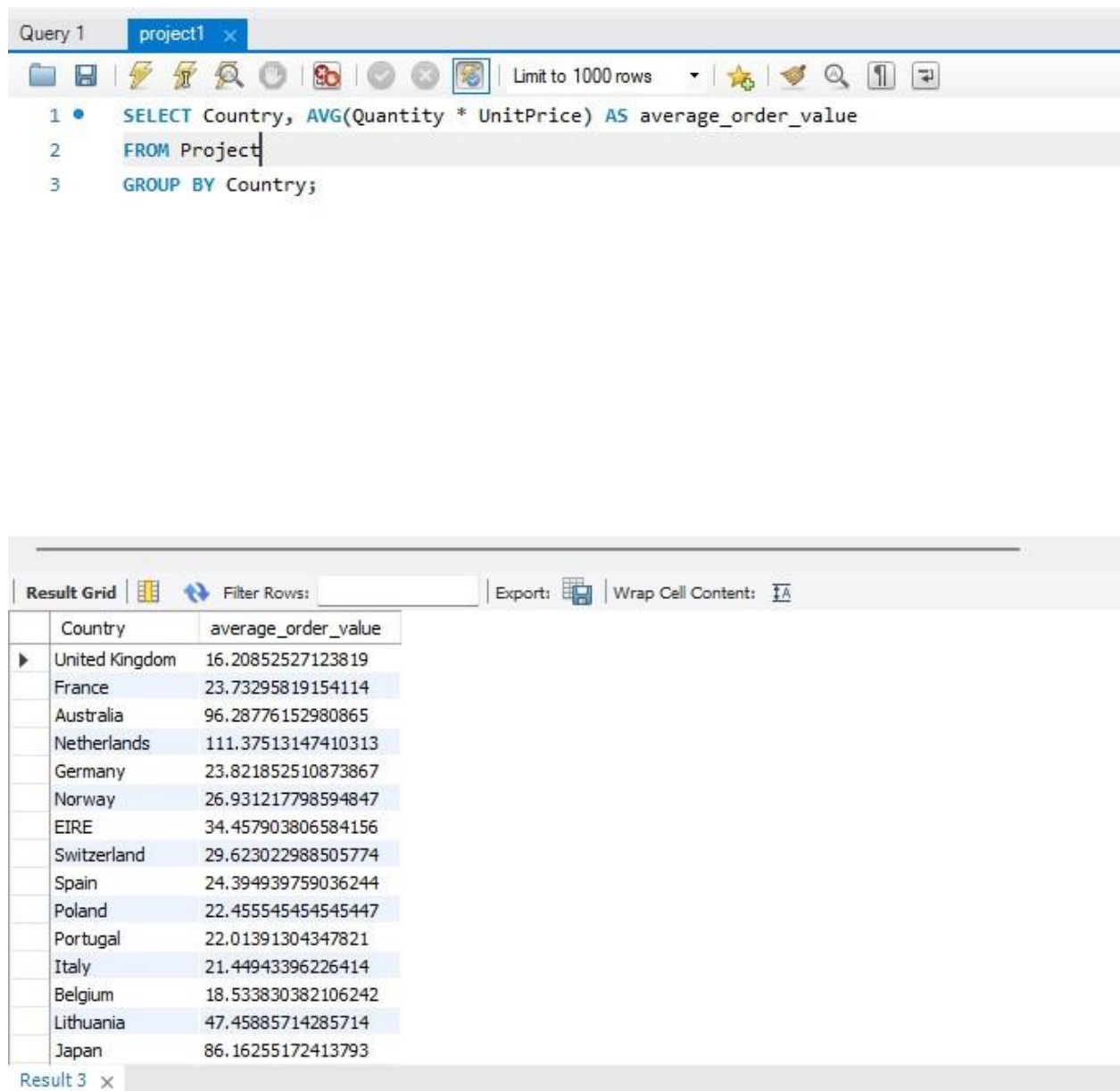
CustomerID	purchase_frequency
	High Frequency
12346	Medium Frequency
12347	Medium Frequency
12348	Medium Frequency
12350	Low Frequency
12352	High Frequency
12353	Low Frequency
12354	Low Frequency
12355	Low Frequency
12356	Medium Frequency
12358	Low Frequency
12359	Medium Frequency
12360	Low Frequency
12361	Low Frequency
12362	Medium Frequency

Result 2 x

## Customer Segmentation by Purchase Frequency

1. In this picture customers are grouped based on their purchase frequency.

# Project Datamining



Query 1 project1 x

Limit to 1000 rows

```
1 • SELECT Country, AVG(Quantity * UnitPrice) AS average_order_value
2 FROM Project
3 GROUP BY Country;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	Country	average_order_value
▶	United Kingdom	16.20852527123819
	France	23.73295819154114
	Australia	96.28776152980865
	Netherlands	111.37513147410313
	Germany	23.821852510873867
	Norway	26.931217798594847
	EIRE	34.457903806584156
	Switzerland	29.623022988505774
	Spain	24.394939759036244
	Poland	22.455545454545447
	Portugal	22.01391304347821
	Italy	21.44943396226414
	Belgium	18.533830382106242
	Lithuania	47.45885714285714
	Japan	86.16255172413793

Result 3 x

## Average Order Value by Country

2. In this picture the avg for purchase is shown for each country.

# Project Datamining

The screenshot shows a SQL query editor window titled "Query 1" with a tab labeled "project1". The query is as follows:

```
1 • SELECT CustomerID
2 FROM project
3 WHERE InvoiceDate <= DATE_SUB(NOW(), INTERVAL 6 MONTH)
4 GROUP BY CustomerID;
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, a "Wrap Cell Content:" checkbox, and a "Fetch rows:" button. The result grid displays a table with the following data:

CustomerID
17850
13047
12583
13748
15100
15291
14688
17809

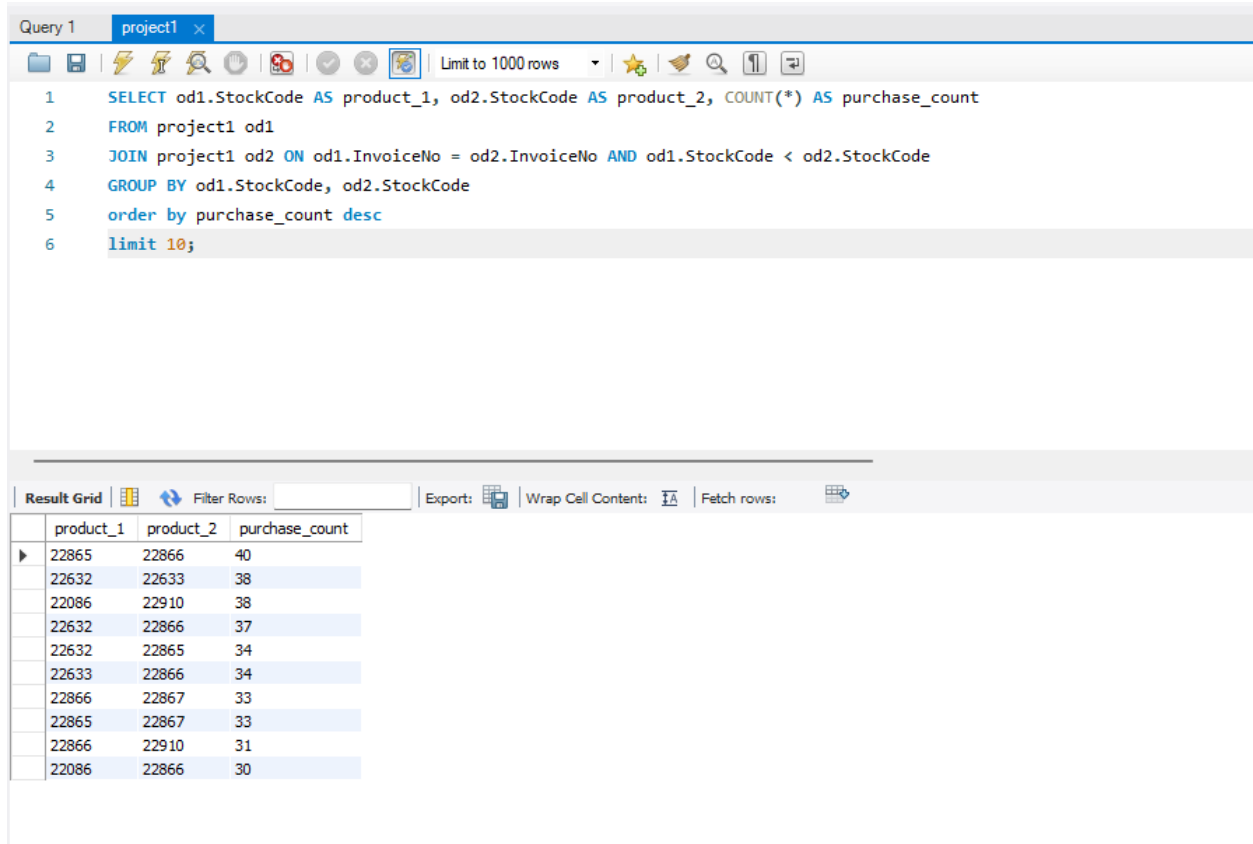
The bottom of the window shows a tab labeled "project 4".

## Customer Churn Analysis

3. In this picture the customer who haven't made any purchase till last 6 months.



# Project Datamining



The screenshot displays a SQL query editor window titled "Query 1" with a tab labeled "project1". The query is as follows:

```
1 SELECT od1.StockCode AS product_1, od2.StockCode AS product_2, COUNT(*) AS purchase_count
2 FROM project1 od1
3 JOIN project1 od2 ON od1.InvoiceNo = od2.InvoiceNo AND od1.StockCode < od2.StockCode
4 GROUP BY od1.StockCode, od2.StockCode
5 order by purchase_count desc
6 limit 10;
```

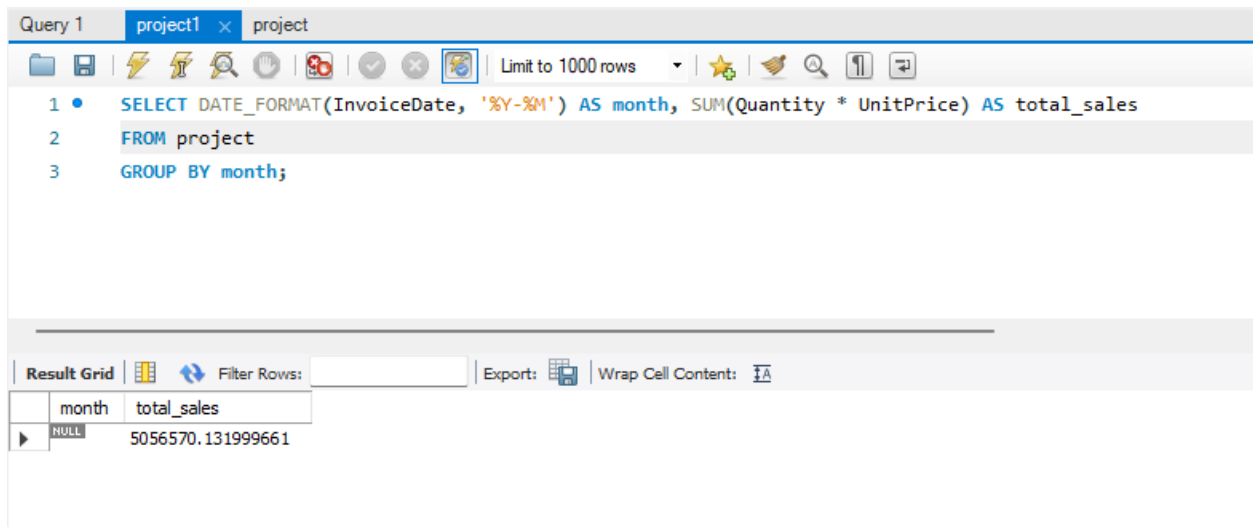
Below the query editor, the "Result Grid" shows the top 10 results of the query. The grid has columns for "product\_1", "product\_2", and "purchase\_count".

	product_1	product_2	purchase_count
▶	22865	22866	40
	22632	22633	38
	22086	22910	38
	22632	22866	37
	22632	22865	34
	22633	22866	34
	22866	22867	33
	22865	22867	33
	22866	22910	31
	22086	22866	30

## Product Affinity Analysis

4. In this picture which products are often purchased together by the correlation between product purchases is shown.

# Project Datamining



The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query 1', 'project1', and 'project'. Below the tabs is a toolbar with various icons for file operations, execution, and formatting. The main area contains a SQL query:

```
1 • SELECT DATE_FORMAT(InvoiceDate, '%Y-%M') AS month, SUM(Quantity * UnitPrice) AS total_sales
2 FROM project
3 GROUP BY month;
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid itself shows a table with two columns: 'month' and 'total\_sales'. The first row has a 'NULL' value for 'month' and a value of '5056570.131999661' for 'total\_sales'.

month	total_sales
NULL	5056570.131999661

## Time-based Analysis

**5.**In this picture the trends in customer behavior over time, such as monthly or quarterly sales patterns is shown.