

# WALMART SALES

*A MYSQL STUDY OF WALMART SALES TRENDS*



SlideShare

**Purposes of the Project:** The main goal of this project is to gain understanding from Walmart's Sales Data, exploring the various factors that influence sales across different Branches.

**About Data:** This project's data was obtained from the Kaggle Walmart Sales Forecasting Competition and it encompasses sales transactions from three Walmart branches situated in Mandalay, Yangon, and Naypyitaw, respectively. The data contains 17 columns and 1000 rows.



# GENERAL QUESTION ANALYSIS



- 1. How many unique cities does the data have?*
- 2. In which city is each branch situated?*



# 1 How many unique cities does the data have?



The screenshot shows a SQL query editor with a toolbar at the top. The toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and other utility icons. The query text is as follows:



```
1  /**How many unique cities does the data have?*/  
2  •  SELECT DISTINCT city FROM sales;  
3
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with the following data:

	city
▶	Yangon
	Naypyitaw
	Mandalay





## 2 In which city is each branch situated?



Limit to 1000 rows

```
1  /** In which city is each branch situated**/  
2 • SELECT DISTINCT city, branch FROM sales;  
3
```

<

**Result Grid** Filter Rows:  | Export:  | Wrap Cell Content: 

	city	branch
▶	Yangon	A
	Naypyitaw	C
	Mandalay	B

# PRODUCT QUESTION ANALYSIS



1. How many product lines are there in the dataset?
2. What is the most common payment method?
3. What is the most selling product line?
4. What is the total revenue by month?
5. Which month recorded the highest Cost of Goods Sold (COGS)?
6. Which product line generated the highest revenue?
7. Which city has the highest revenue?
8. Which product line incurred the highest VAT?
9. Retrieve each product line and add a column product category, indicating 'Good' or 'Bad,' based
10. Which branch sold more products than average product sold?
11. What is the most common product line by gender?
12. What is the average rating of each product line?

# How many unique product lines does the data have?



Limit to 1000 rows

```
1  /**How many unique product lines does the data have?*/  
2  • SELECT count(distinct product_line) FROM sales;  
3
```

Result Grid

	count(distinct product_line)
▶	6

Result 4 x

# What is the most common payment method?



The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, the SQL query is displayed in a monospaced font. The query is a SELECT statement that counts the frequency of different payment methods from a table named 'sales'. The results are shown in a table below the query. The table has two columns: 'payment' and 'frequency\_common\_method'. The first row of data shows 'Cash' as the payment method with a frequency of 344.

```
1  /**What is the most common payment method?*/  
2  •  SELECT payment, count(payment) AS  
3     frequency_common_method FROM sales GROUP BY payment  
4     ORDER BY payment LIMIT 1;
```

Result Grid

	payment	frequency_common_method
▶	Cash	344

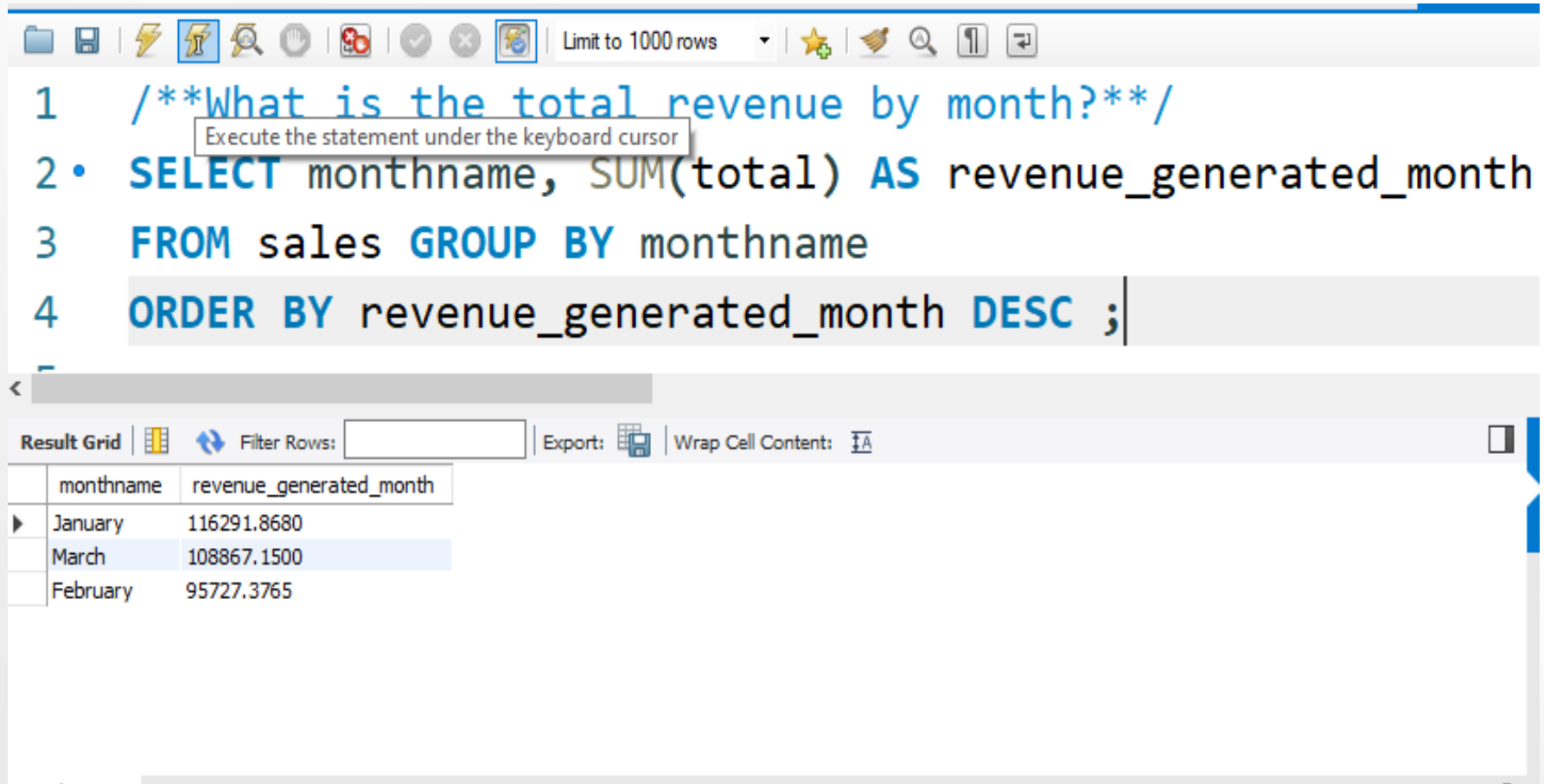


# What is the most selling product line?

```
1  /**What is the most selling product line?*/
2  • SELECT product_line, COUNT(product_line) AS cnt_pdct_line
3  FROM sales GROUP BY product_line ORDER BY cnt_pdct_line
4  DESC LIMIT 1;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
product_line	cnt_pdct_line			
Fashion accessories	178			

# What is the total revenue by month?



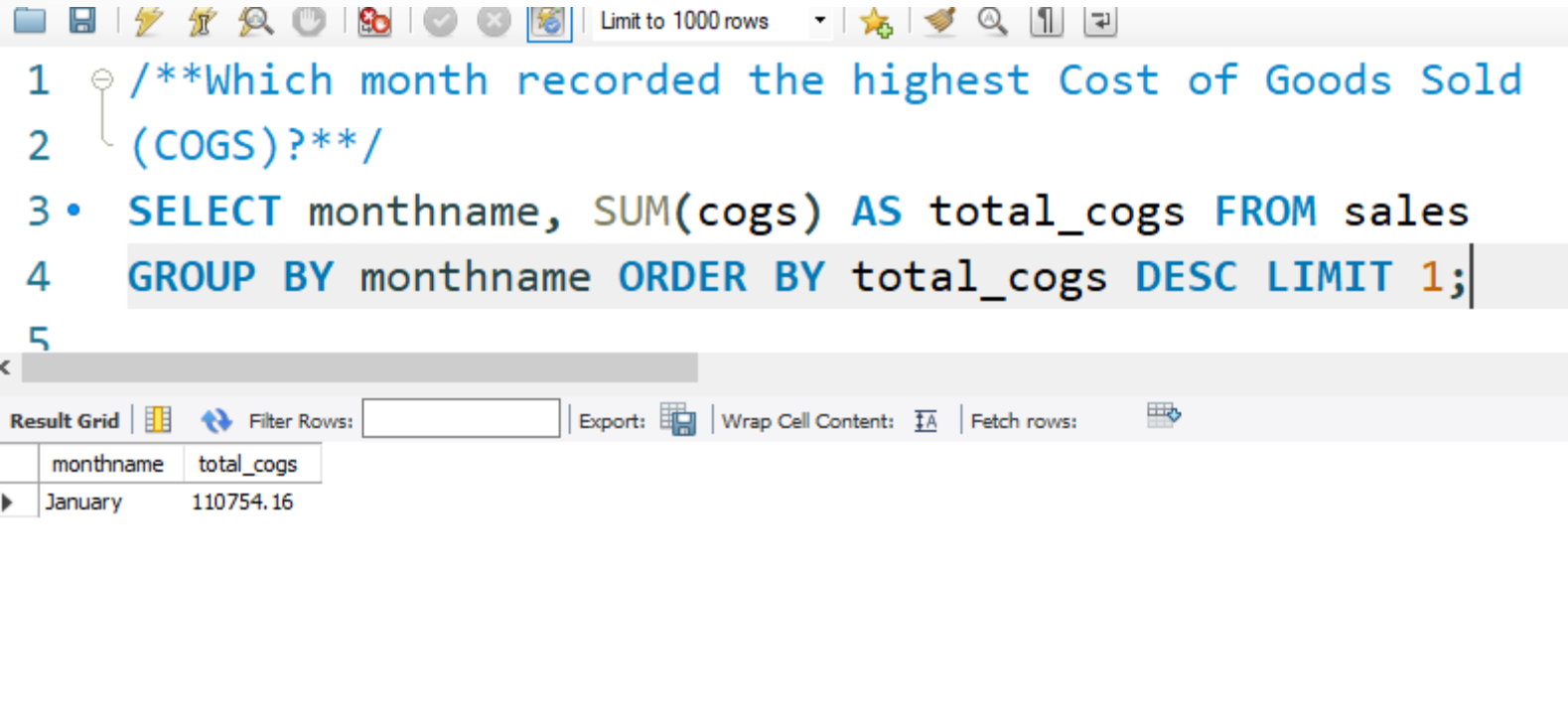
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The query editor contains the following SQL code:

```
1  /**What is the total revenue by month?*/  
2  • SELECT monthname, SUM(total) AS revenue_generated_month  
3  FROM sales GROUP BY monthname  
4  ORDER BY revenue_generated_month DESC ;
```

A tooltip "Execute the statement under the keyboard cursor" is visible over the first line of the query. Below the query editor, the "Result Grid" tab is active, displaying the following data:

monthname	revenue_generated_month
January	116291.8680
March	108867.1500
February	95727.3765

# Which month recorded the highest Cost of Goods Sold (COGS)?



The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and navigation. Below the toolbar, the SQL query is entered in a text area. The query is as follows:


```
1 /**Which month recorded the highest Cost of Goods Sold
2 (COGS)?**/
3 • SELECT monthname, SUM(cogs) AS total_cogs FROM sales
4 GROUP BY monthname ORDER BY total_cogs DESC LIMIT 1;
5
```

Below the query editor, there is a section for the results. It includes a "Result Grid" tab, a "Filter Rows" input field, and buttons for "Export", "Wrap Cell Content", and "Fetch rows". The results are displayed in a table with two columns: "monthname" and "total\_cogs". The table shows one row for "January" with a "total\_cogs" value of 110754.16.

monthname	total_cogs
January	110754.16






# Which product line generated the highest revenue?



Limit to 1000 rows

```
1  /**Which product line generated the highest revenue?*/  
2  • SELECT product_line, SUM(total) AS  
3  revenue_generated_product_line FROM sales GROUP BY  
4  product_line ORDER BY revenue_generated_product_line  
5  DESC LIMIT 1;
```

<

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	product_line	revenue_generated_product_line
▶	Food and beverages	56144.8440

# Which city has the highest revenue?



Limit to 1000 rows

```
1  /**Which city has the highest revenue?*/  
2  • SELECT city, SUM(total) AS total_revenue FROM sales  
3  GROUP BY city ORDER BY total_revenue DESC LIMIT 1;  
4
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



	city	total_revenue
▶	Naypyitaw	110490.7755



# Which product line incurred the highest VAT?



Limit to 1000 rows

```
1  /**Which product line incurred the highest VAT?*/
2  •  SELECT product_line, SUM(tax_pct) as VAT FROM sales
3  GROUP BY product_line ORDER BY VAT DESC LIMIT 1;
```

Result Grid

Filter Rows:

Export: Wrap Cell Content: Fetch rows:

product_line	VAT
Food and beverages	2673.5640

Retrieve each product line and add a column product category, indicating 'Good' or 'Bad,' based on whether its sales are above the average.

```
1  /**Retrieve each product line and add a column product_category,  
2   indicating 'Good' or 'Bad,' based on whether  
3   its sales are above the average.**/  
4  •  SELECT product_line,  
5     CASE WHEN total > (SELECT AVG(total) FROM sales)  
6     THEN 'Good' ELSE 'Bad' END AS product_category FROM sales;
```














Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

	product_line	product_category
▶	Food and beverages	Good
	Health and beauty	Bad
	Sports and travel	Bad
	Health and beauty	Good
	Fashion accessories	Bad
	Sports and travel	Good
	Home and lifestyle	Bad
	Fashion accessories	Good
	Sports and travel	Good
	Food and beverages	Bad
	Fashion accessories	Bad
	Electronic accessories	Good
	Fashion accessories	Bad

# Which branch sold more products than average product sold?








12\*   SQL File 11\*   home work sql class\*   SQL File 12\*   SQL File 13\*   SQL File 17\*   SQL File 13\*   walmart\_sales sql\*   sales

```
1  /**Which branch sold more products than average product sold?*/
2  •  SELECT branch, SUM(quantity) AS quantity
3     FROM sales GROUP BY branch HAVING SUM(quantity) > AVG(quantity) ORDER BY quantity
4     DESC limit 1;
```

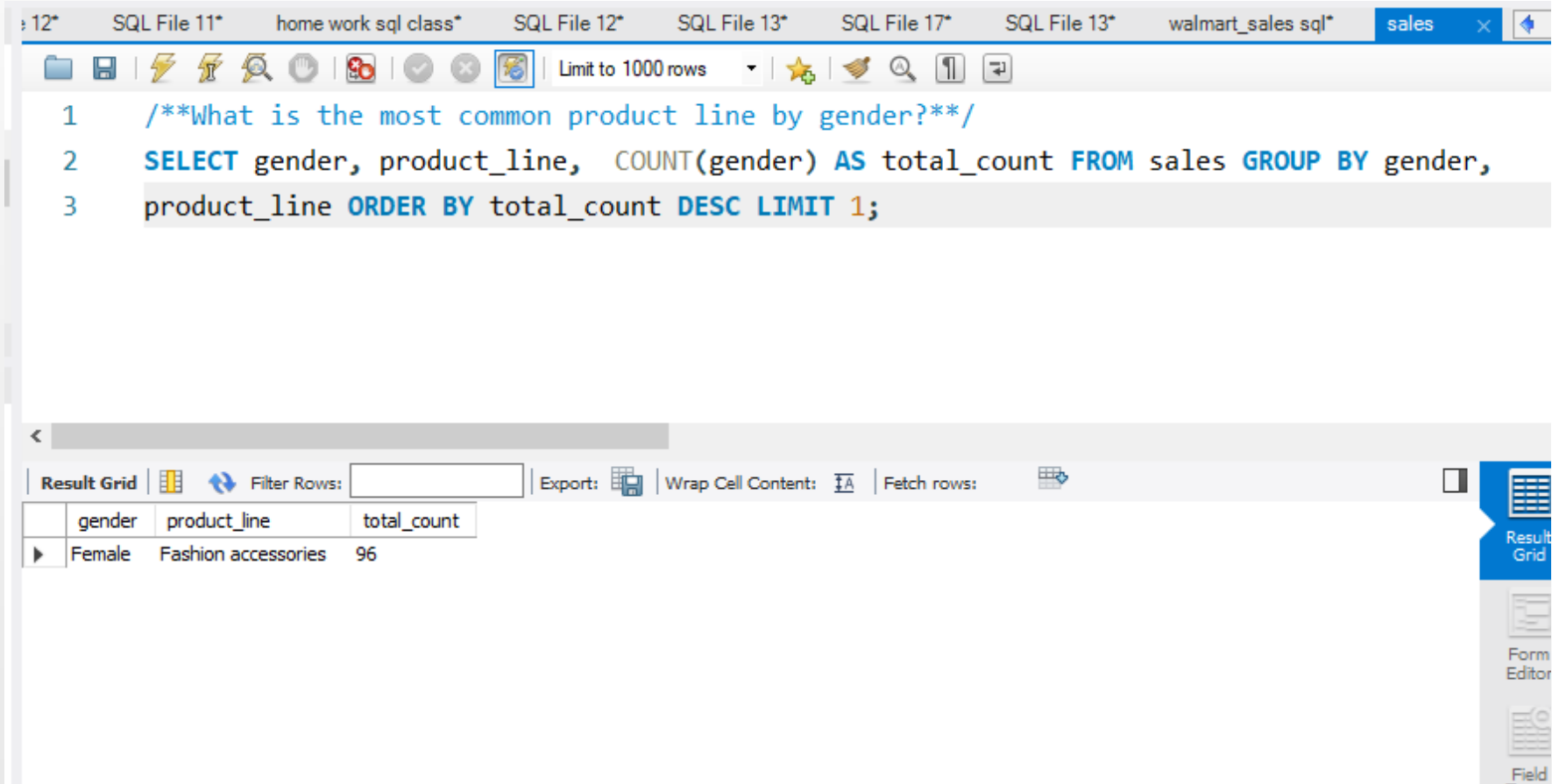
<

Result Grid   Filter Rows:

Export:  Wrap Cell Content:  Fetch rows: 

	branch	quantity
▶	A	1849

# What is the most common product line by gender?



The screenshot shows a SQL IDE interface with multiple tabs. The active tab is 'sales'. The query editor contains the following SQL code:

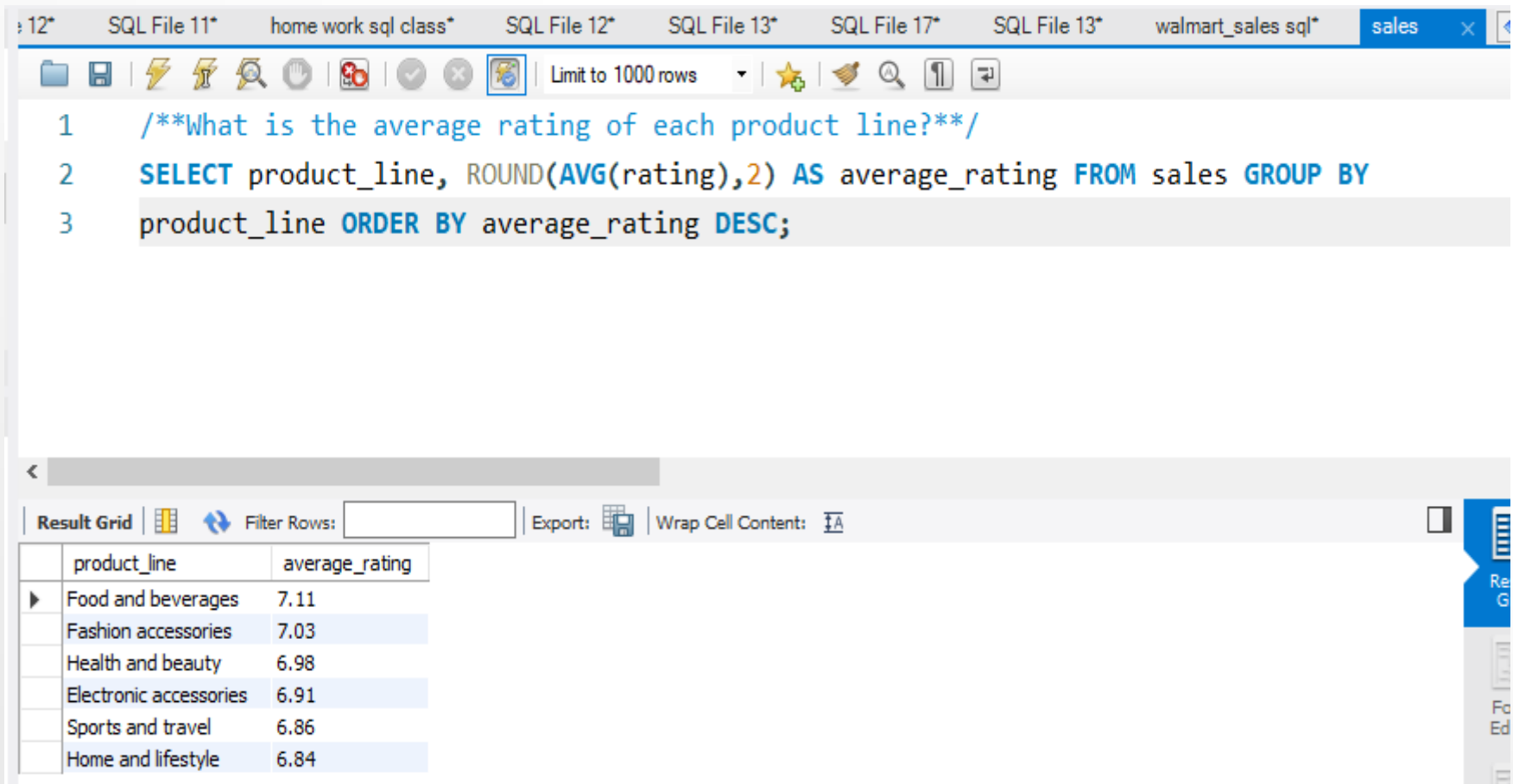
```
1  /**What is the most common product line by gender?*/  
2  SELECT gender, product_line, COUNT(gender) AS total_count FROM sales GROUP BY gender,  
3  product_line ORDER BY total_count DESC LIMIT 1;
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query. The grid has three columns: 'gender', 'product\_line', and 'total\_count'. The first row of data is:

gender	product_line	total_count
Female	Fashion accessories	96

The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field'.

# What is the average rating of each product line?



The screenshot shows a SQL IDE interface with multiple tabs at the top: 'SQL File 11\*', 'home work sql class\*', 'SQL File 12\*', 'SQL File 13\*', 'SQL File 17\*', 'SQL File 13\*', 'walmart\_sales sql\*', and 'sales'. The 'sales' tab is active. The query editor displays the following SQL code:

```
1  /**What is the average rating of each product line?*/  
2  SELECT product_line, ROUND(AVG(rating),2) AS average_rating FROM sales GROUP BY  
3  product_line ORDER BY average_rating DESC;
```

Below the query editor, the 'Result Grid' tab is selected, showing the results of the query. The results are displayed in a table with two columns: 'product\_line' and 'average\_rating'.

product_line	average_rating
Food and beverages	7.11
Fashion accessories	7.03
Health and beauty	6.98
Electronic accessories	6.91
Sports and travel	6.86
Home and lifestyle	6.84

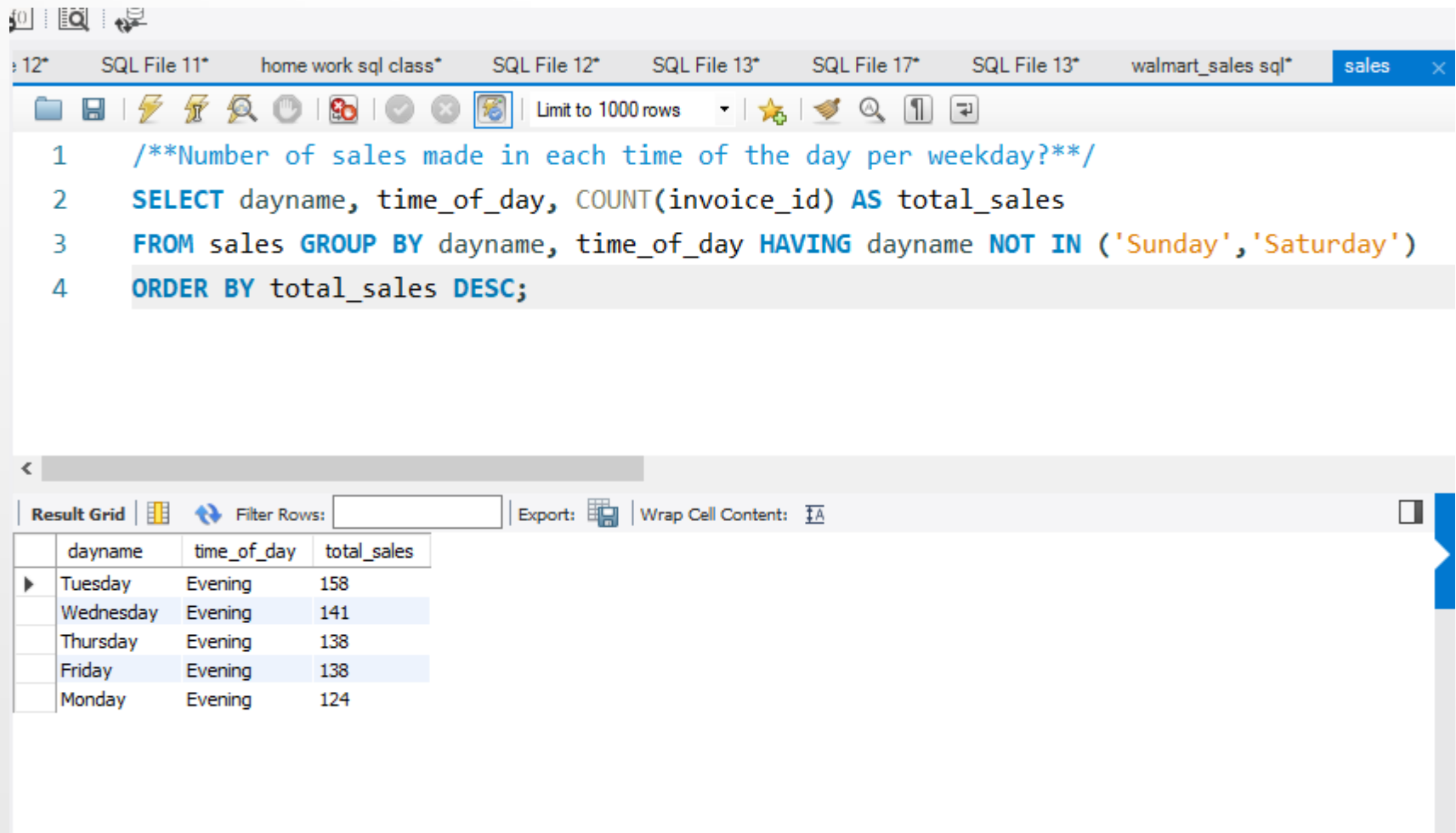


# Sales Question Analysis



1. Number of sales made in each time of the day per weekday
2. Identify the customer type that generates the highest revenue.
3. Which city has the largest tax percent/ VAT (Value Added Tax)?
4. Which customer type pays the most VAT?

# Number of sales made in each time of the day per weekday?



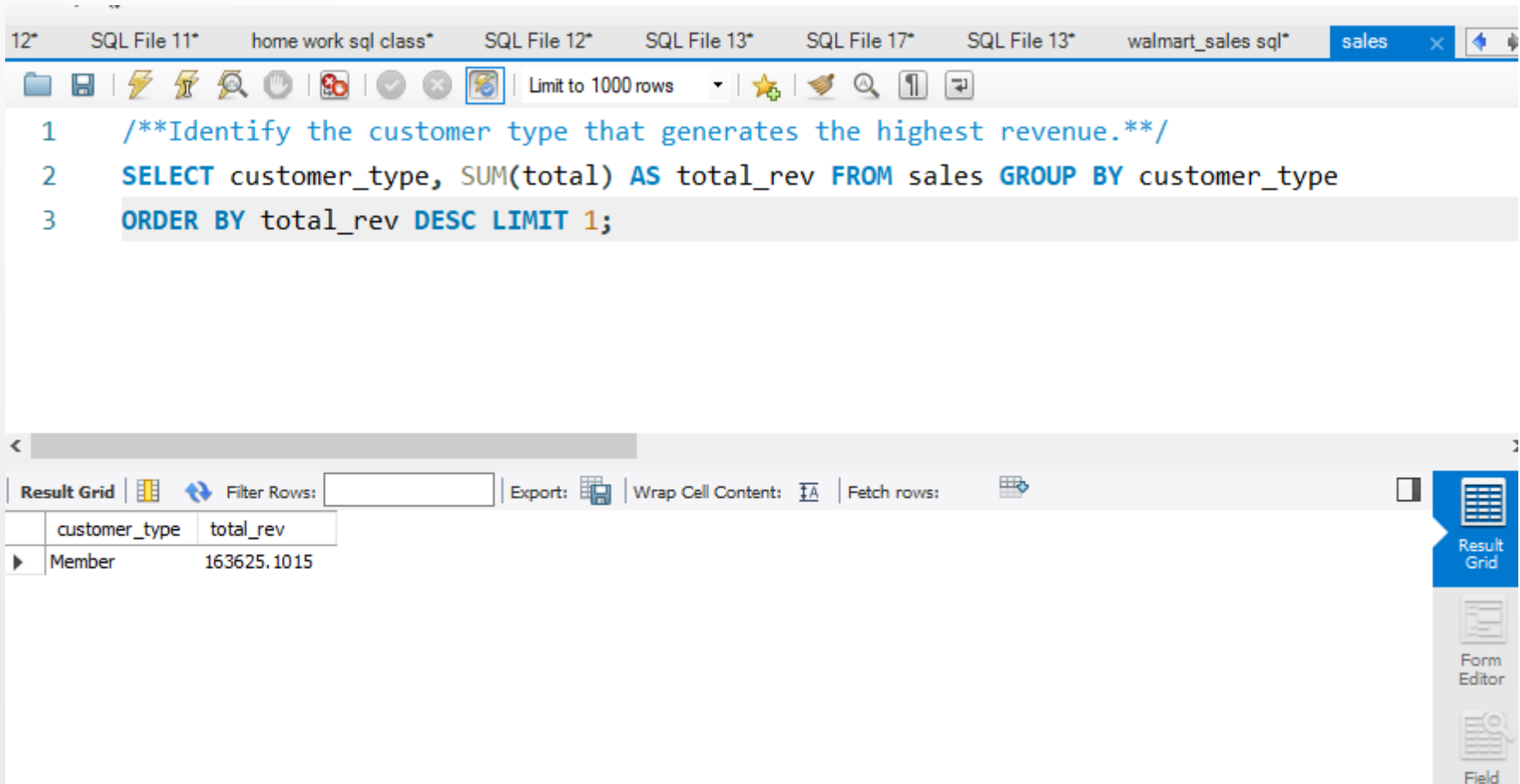
The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1  /**Number of sales made in each time of the day per weekday?*/  
2  SELECT dayname, time_of_day, COUNT(invoice_id) AS total_sales  
3  FROM sales GROUP BY dayname, time_of_day HAVING dayname NOT IN ('Sunday','Saturday')  
4  ORDER BY total_sales DESC;
```

The result grid displays the following data:

	dayname	time_of_day	total_sales
▶	Tuesday	Evening	158
	Wednesday	Evening	141
	Thursday	Evening	138
	Friday	Evening	138
	Monday	Evening	124

Identify the customer type that generates the highest revenue.



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

```
1  /**Identify the customer type that generates the highest revenue.**/  
2  SELECT customer_type, SUM(total) AS total_rev FROM sales GROUP BY customer_type  
3  ORDER BY total_rev DESC LIMIT 1;
```

The results pane displays the following data:

customer_type	total_rev
Member	163625.1015

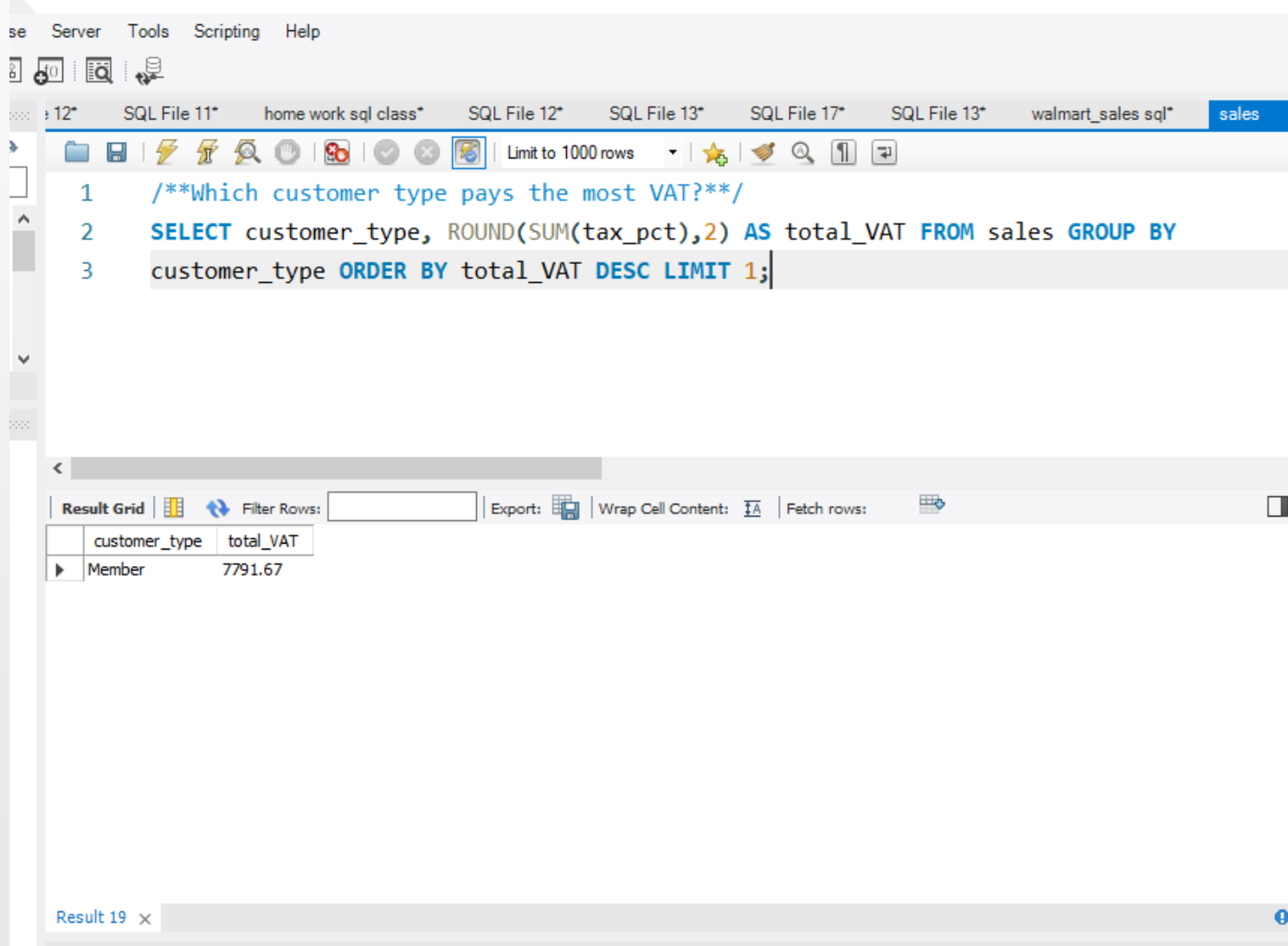
The IDE interface includes a toolbar with various icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button on the right side.

# Which city has the largest tax percent/ VAT (Value Added Tax)?

```
1  /**Which city has the largest tax percent/ VAT (Value Added Tax)?**/  
2  • SELECT city, ROUND(SUM(tax_pct), 2) AS total_tax_pct FROM sales GROUP BY city  
3  ORDER BY total_tax_pct DESC LIMIT 1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
city	total_tax_pct				
Naypyitaw	5261.47				

# Which customer type pays the most VAT?



The screenshot shows a SQL IDE interface with a menu bar (File, Server, Tools, Scripting, Help) and a toolbar. The main window displays a SQL query in a file named 'sales'. The query is as follows:

```
1  /**Which customer type pays the most VAT?*/  
2  SELECT customer_type, ROUND(SUM(tax_pct),2) AS total_VAT FROM sales GROUP BY  
3  customer_type ORDER BY total_VAT DESC LIMIT 1;
```

Below the query editor, the 'Result Grid' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'customer\_type' and 'total\_VAT'. The table contains one row with the value 'Member' for 'customer\_type' and '7791.67' for 'total\_VAT'.

customer_type	total_VAT
Member	7791.67

The bottom status bar indicates 'Result 19' and a close button.



# CUSTOMER QUESTION ANALYSIS



1. How many unique customer types does the data have?
2. How many unique payment methods does the data have?
3. Which is the most common customer type?
4. What is the gender of most of the customers?
5. What is the gender distribution per branch?
6. Which time of the day do customers give most ratings?
7. Which time of the day do customers give most ratings per branch?
8. Which day of the week has the best avg ratings?
9. Which day of the week has the best average ratings per branch?

# How many unique customer types does the data have?

```
1  /**How many unique customer types does the data have**/  
2  SELECT DISTINCT customer_type FROM sales;
```

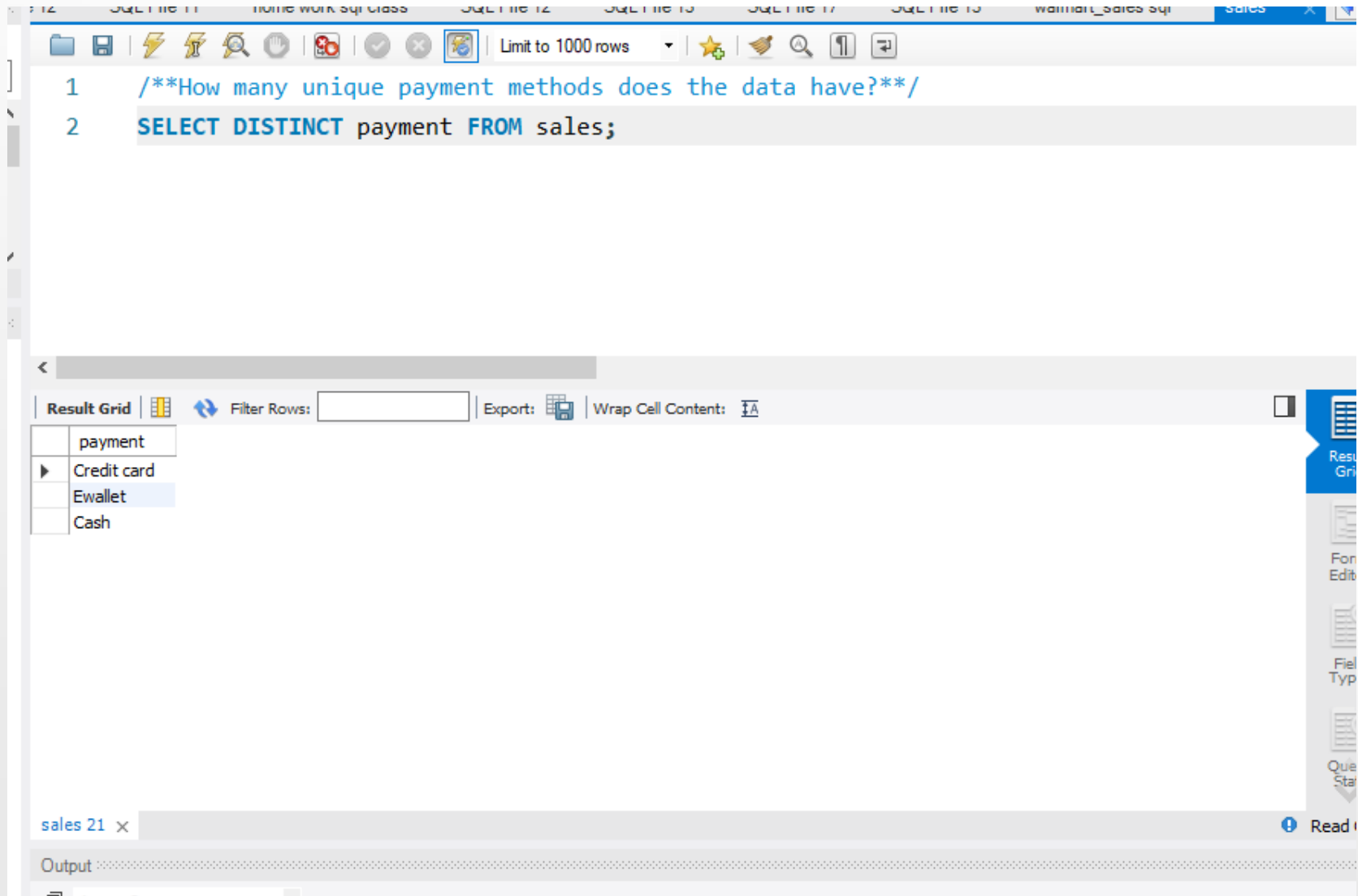
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	customer_type
▶	Normal
	Member

sales 20 x

Output

# How many unique payment types does the data have?



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains two lines of code:

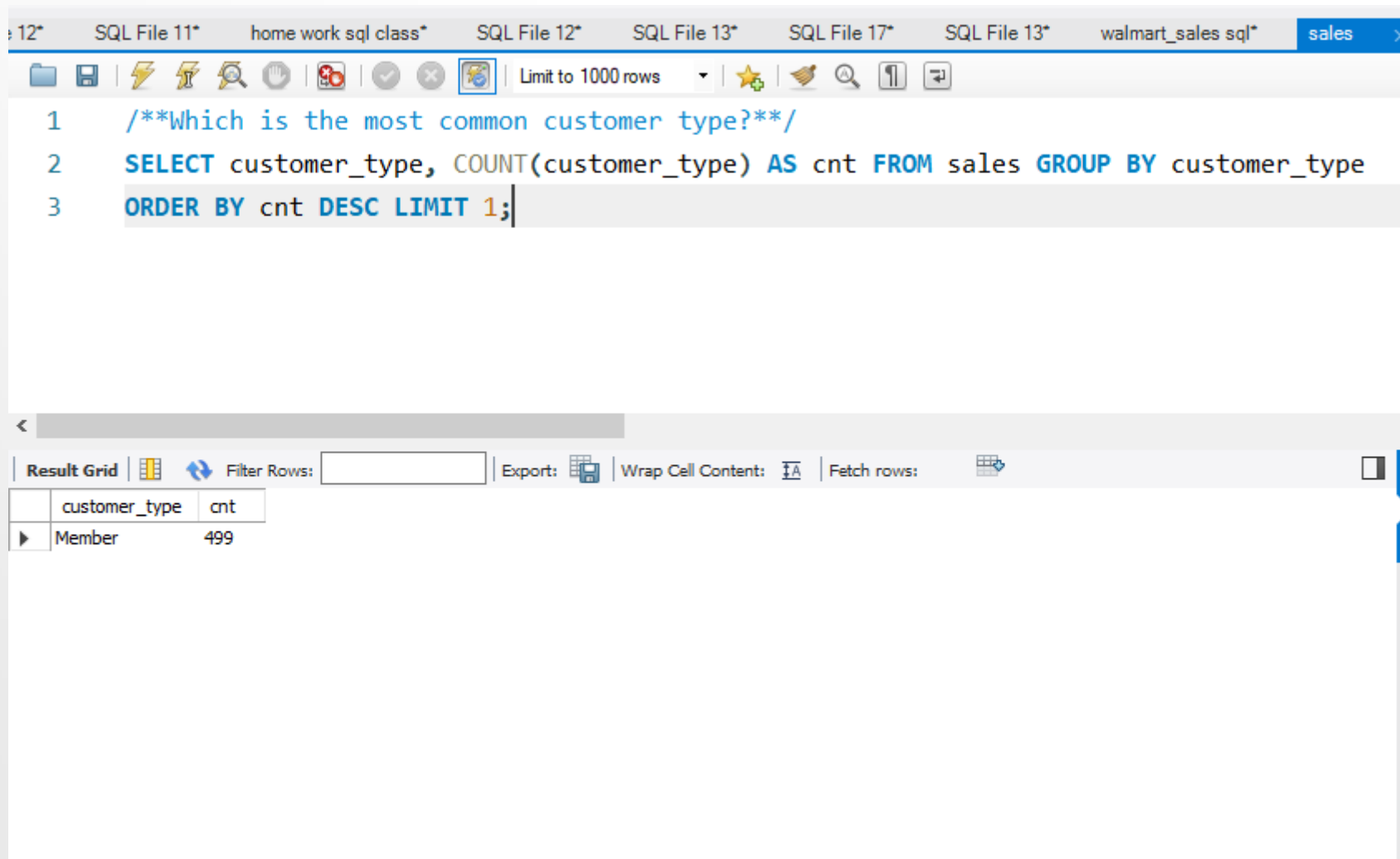
```
1  /**How many unique payment methods does the data have?*/  
2  SELECT DISTINCT payment FROM sales;
```

Below the editor, the 'Result Grid' tab is active. It shows a table with the following data:

payment
Credit card
Ewallet
Cash

The right sidebar contains icons for 'Result Grid', 'Format Edit', 'Field Type', and 'Query Status'. At the bottom, there is an 'Output' pane and a status bar showing 'sales 21' and a 'Read' button.

# Which is the most common customer type?



The screenshot shows a SQL IDE interface with multiple tabs at the top: '12\*', 'SQL File 11\*', 'home work sql class\*', 'SQL File 12\*', 'SQL File 13\*', 'SQL File 17\*', 'SQL File 13\*', 'walmart\_sales sql\*', and 'sales'. The 'sales' tab is active. The toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and other utility icons. The SQL editor contains the following query:

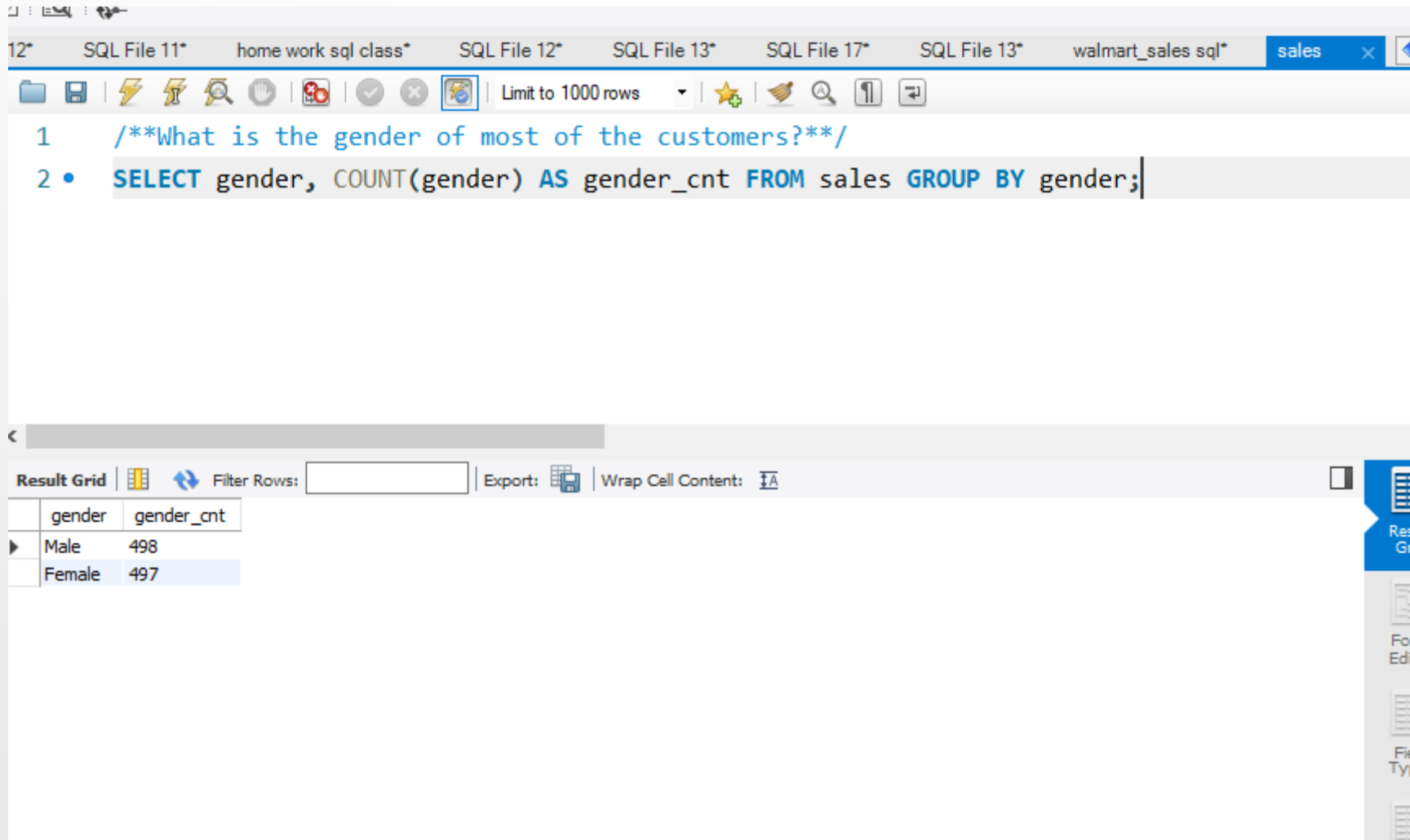
```
1  /**Which is the most common customer type?*/  
2  SELECT customer_type, COUNT(customer_type) AS cnt FROM sales GROUP BY customer_type  
3  ORDER BY cnt DESC LIMIT 1;
```

Below the editor, the 'Result Grid' tab is selected. It shows a single row of results:

	customer_type	cnt
▶	Member	499

The interface also includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button.

# What is the gender of most of the customers?



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, a search icon, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL code:

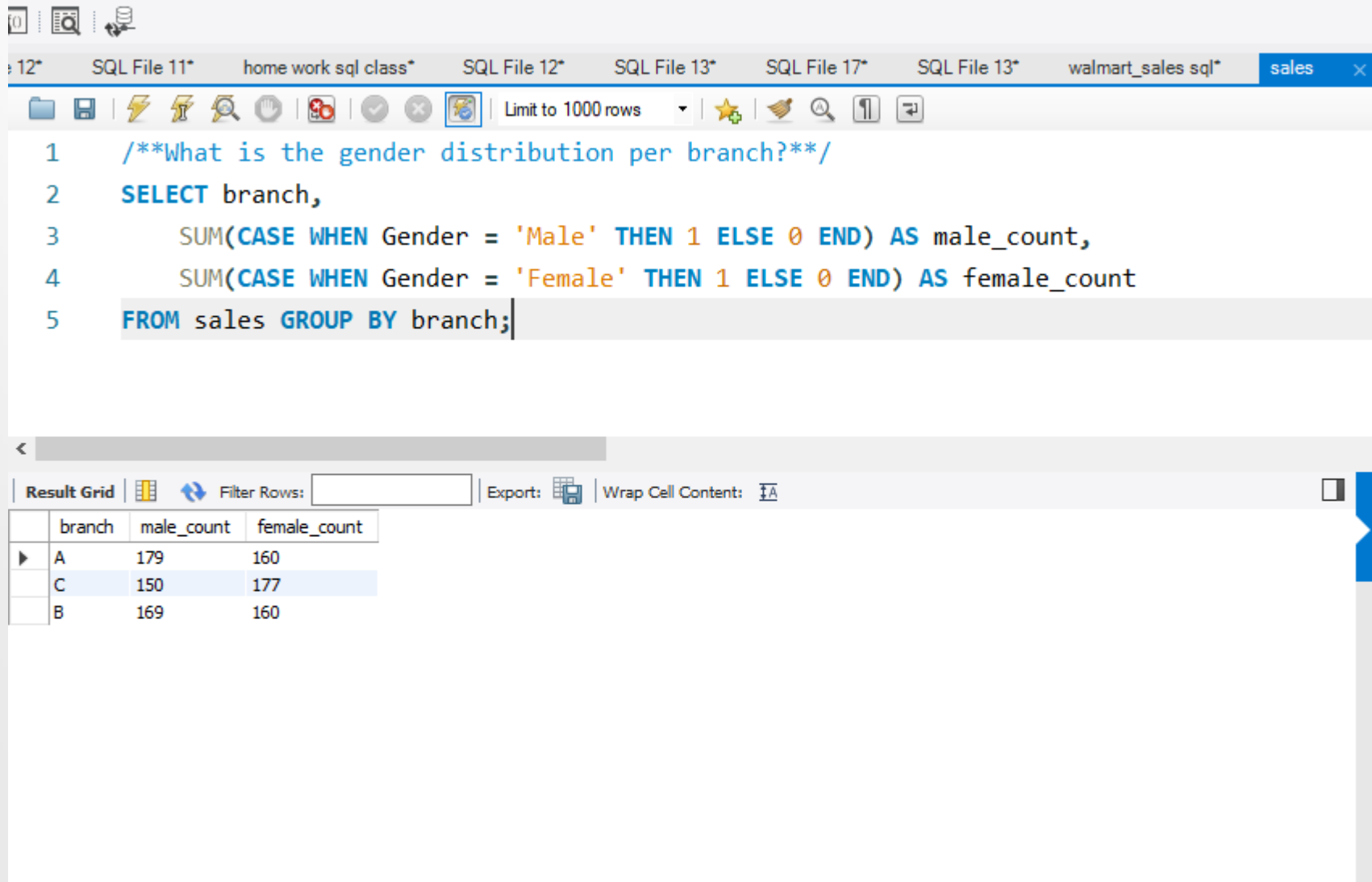
```
1  /**What is the gender of most of the customers?*/  
2  •  SELECT gender, COUNT(gender) AS gender_cnt FROM sales GROUP BY gender;
```

Below the query editor, the 'Result Grid' is displayed. It has a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are shown in a table with two columns: 'gender' and 'gender\_cnt'.

gender	gender_cnt
Male	498
Female	497



# What is the gender distribution per branch?



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1  /**What is the gender distribution per branch?*/  
2  SELECT branch,  
3         SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0 END) AS male_count,  
4         SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) AS female_count  
5  FROM sales GROUP BY branch;
```

Below the editor, the 'Result Grid' tab is active, displaying the query results in a table. The table has four columns: 'branch', 'male\_count', and 'female\_count'. The results are as follows:

	branch	male_count	female_count
▶	A	179	160
	C	150	177
	B	169	160

# Which time of the day do customers give most ratings?

12\* SQL File 11\* home work sql class\* SQL File 12\* SQL File 13\* SQL File 17\* SQL File 13\* walmart\_sales sql\* sales

Limit to 1000 rows

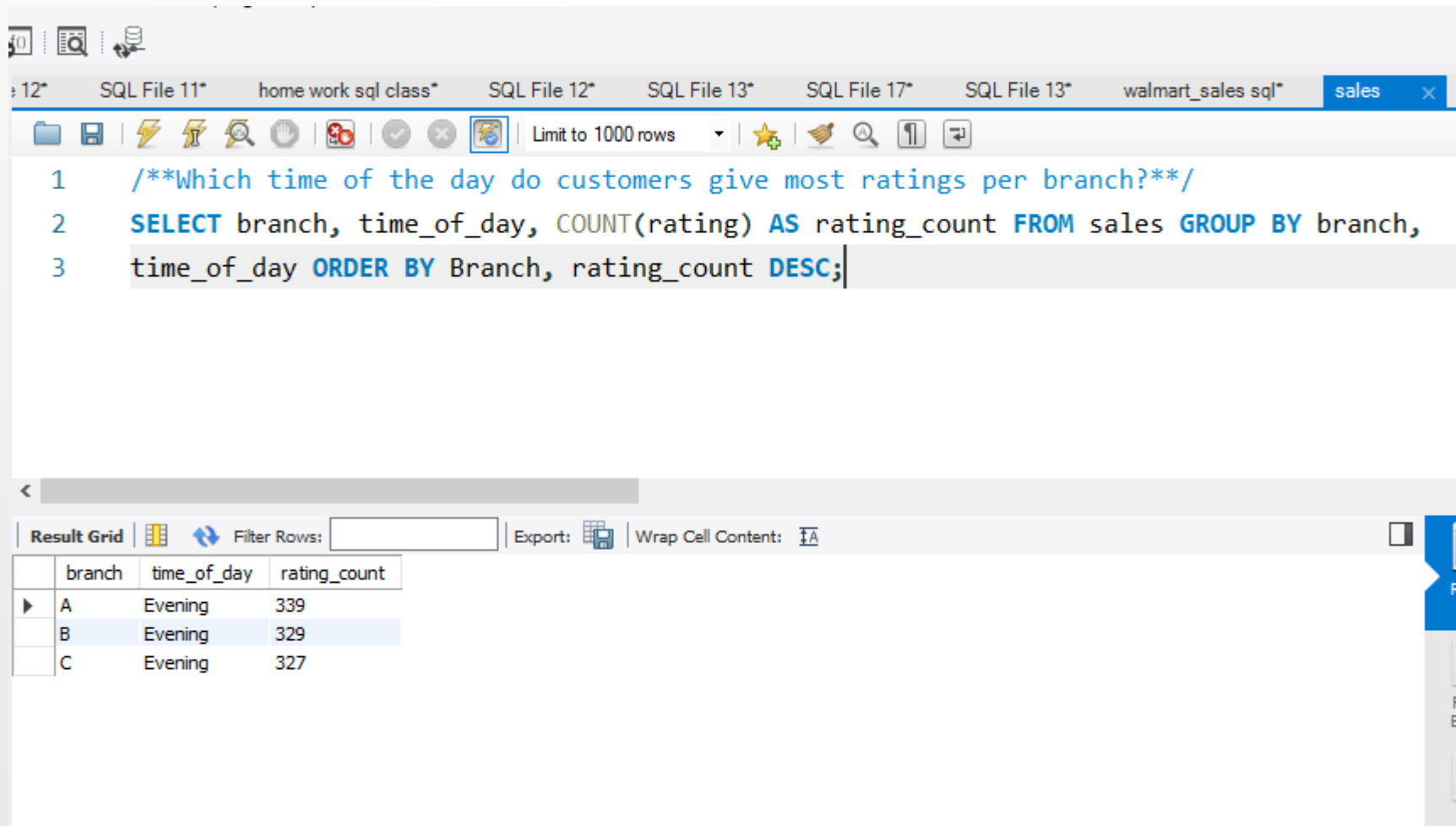
```
1  /**Which time of the day do customers give most ratings?*/  
2  SELECT SUM(rating) AS sum_of_rating , time_of_day FROM sales GROUP BY time_of_day  
3  ORDER BY sum_of_rating DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	sum_of_rating	time_of_day
►	6922.7	Evening

Result Grid  
Format  
Edit  
Field Type  
Query Status

# Which time of the day do customers give most ratings per branch?



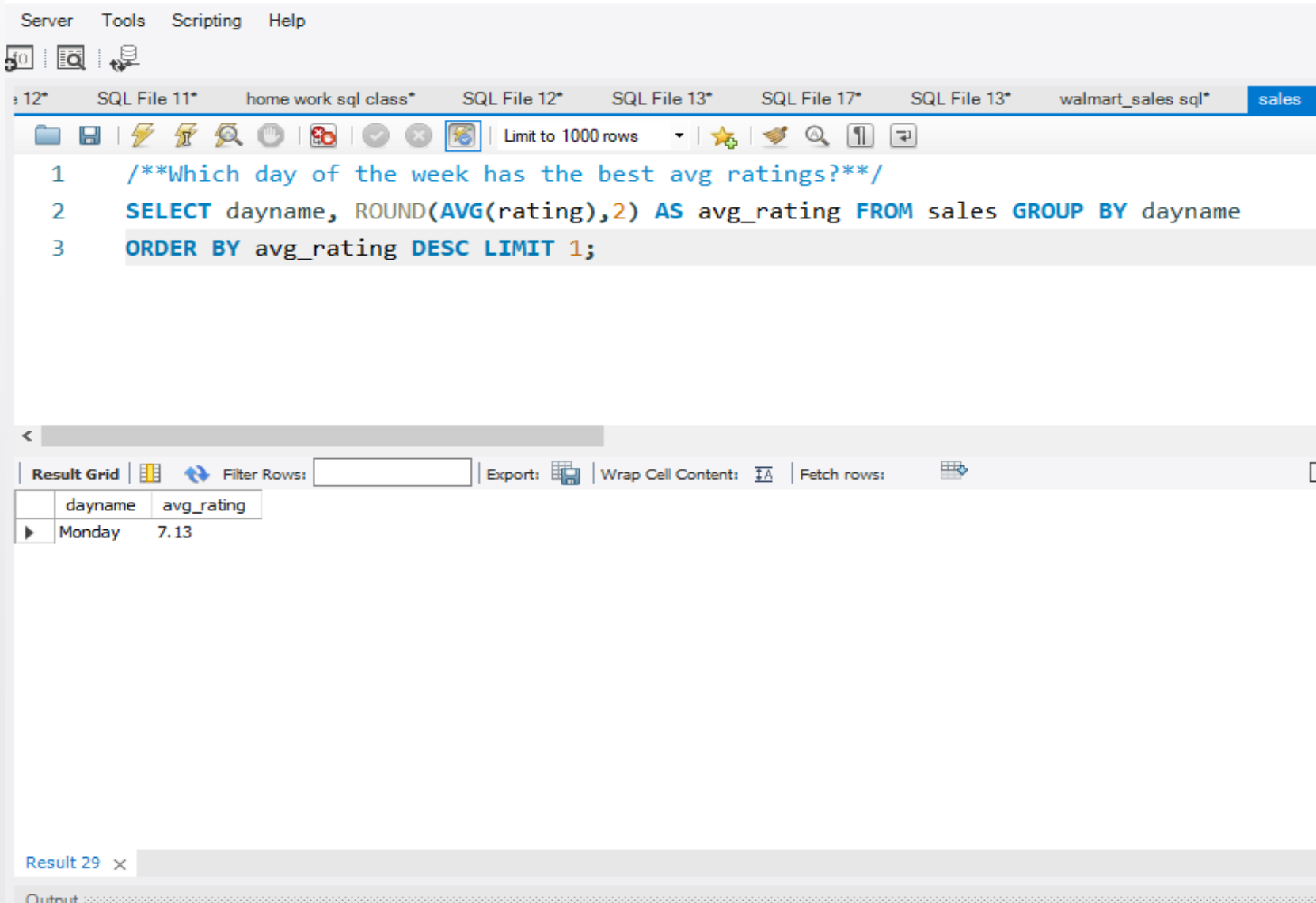
The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1  /**Which time of the day do customers give most ratings per branch?*/  
2  SELECT branch, time_of_day, COUNT(rating) AS rating_count FROM sales GROUP BY branch,  
3  time_of_day ORDER BY Branch, rating_count DESC;
```

The results grid displays the following data:

	branch	time_of_day	rating_count
▶	A	Evening	339
	B	Evening	329
	C	Evening	327

# Which day of the week has the best avg ratings?



The screenshot shows a SQL IDE interface with a menu bar (Server, Tools, Scripting, Help) and a toolbar. The file explorer shows several SQL files, with 'walmart\_sales.sql' selected. The query editor contains the following SQL code:

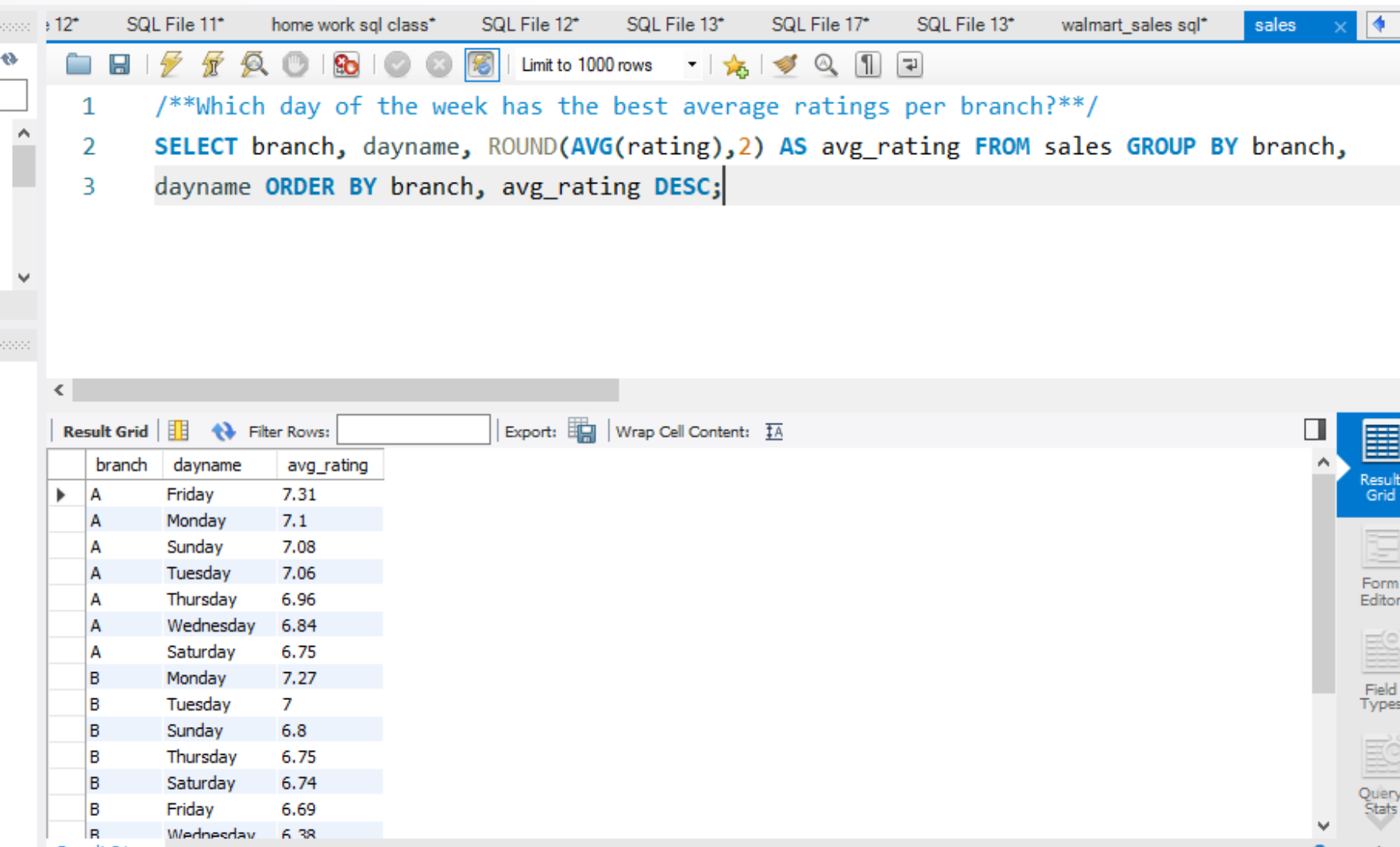
```
1  /**Which day of the week has the best avg ratings?*/  
2  SELECT dayname, ROUND(AVG(rating),2) AS avg_rating FROM sales GROUP BY dayname  
3  ORDER BY avg_rating DESC LIMIT 1;
```

The results pane shows the following table:

dayname	avg_rating
Monday	7.13

The interface also includes a 'Result Grid' tab, a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. The bottom status bar shows 'Result 29' and 'Output'.

# Which day of the week has the best average ratings per branch?



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

```
1  /**Which day of the week has the best average ratings per branch?*/  
2  SELECT branch, dayname, ROUND(AVG(rating),2) AS avg_rating FROM sales GROUP BY branch,  
3  dayname ORDER BY branch, avg_rating DESC;
```

The result grid displays the following data:

	branch	dayname	avg_rating
▶	A	Friday	7.31
	A	Monday	7.1
	A	Sunday	7.08
	A	Tuesday	7.06
	A	Thursday	6.96
	A	Wednesday	6.84
	A	Saturday	6.75
	B	Monday	7.27
	B	Tuesday	7
	B	Sunday	6.8
	B	Thursday	6.75
	B	Saturday	6.74
	B	Friday	6.69
	B	Wednesday	6.38

# **ANALYST LIST:**



**Product Analysis:** Perform an analysis on the data to gain insights into different product lines, determine the top-performing product lines, and identify areas for improvement in other product lines.

**Sales Analysis:** The objective of this analysis is to address the inquiry regarding the sales trends of the product. The outcomes of this analysis can assist in evaluating the efficiency of each applied sales strategy in the business and determining necessary modifications to increase sales.

**Customer Analysis:** This analysis is focused on identifying various customer segments, understanding purchasing trends, and evaluating the profitability associated with each of these customer segments.





***THANK YOU***

***BY: Ruposi Chatterjee  
([chatterjeeruposi@gmail.com](mailto:chatterjeeruposi@gmail.com))***