

Restaurant Analysis— MySQL



About

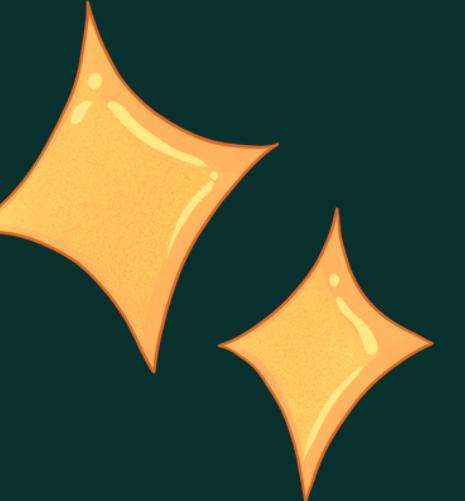
This project is a part of Maven Analytics Guided Project solely focussed on MySQL skills

TOOLS USED

- Maven Analytics Platform for Data
- MySQL Workbench for SQL Queries
- Canva for presentation



DATASETS



There are 2 datasets in this project that the candidates are required to use.

- 1) Menu_Items : menu_item_id, item_name, category, price
- 2) order_details: order_details_id, order_id, order_date, order_time, item_id



Analysis

Task 1 : View the Menu Items Table and write a query to find the total number of items on the menu.

```
1 • SELECT count(distinct(menu_item_id)) as total_menu_items  
2   from menu_items;
```

Result Grid	
	total_menu_items
▶	32

There are a total of 32 items that are listed on the menu.

Analysis

Task 2 : What are the least and most expensive items on the menu?

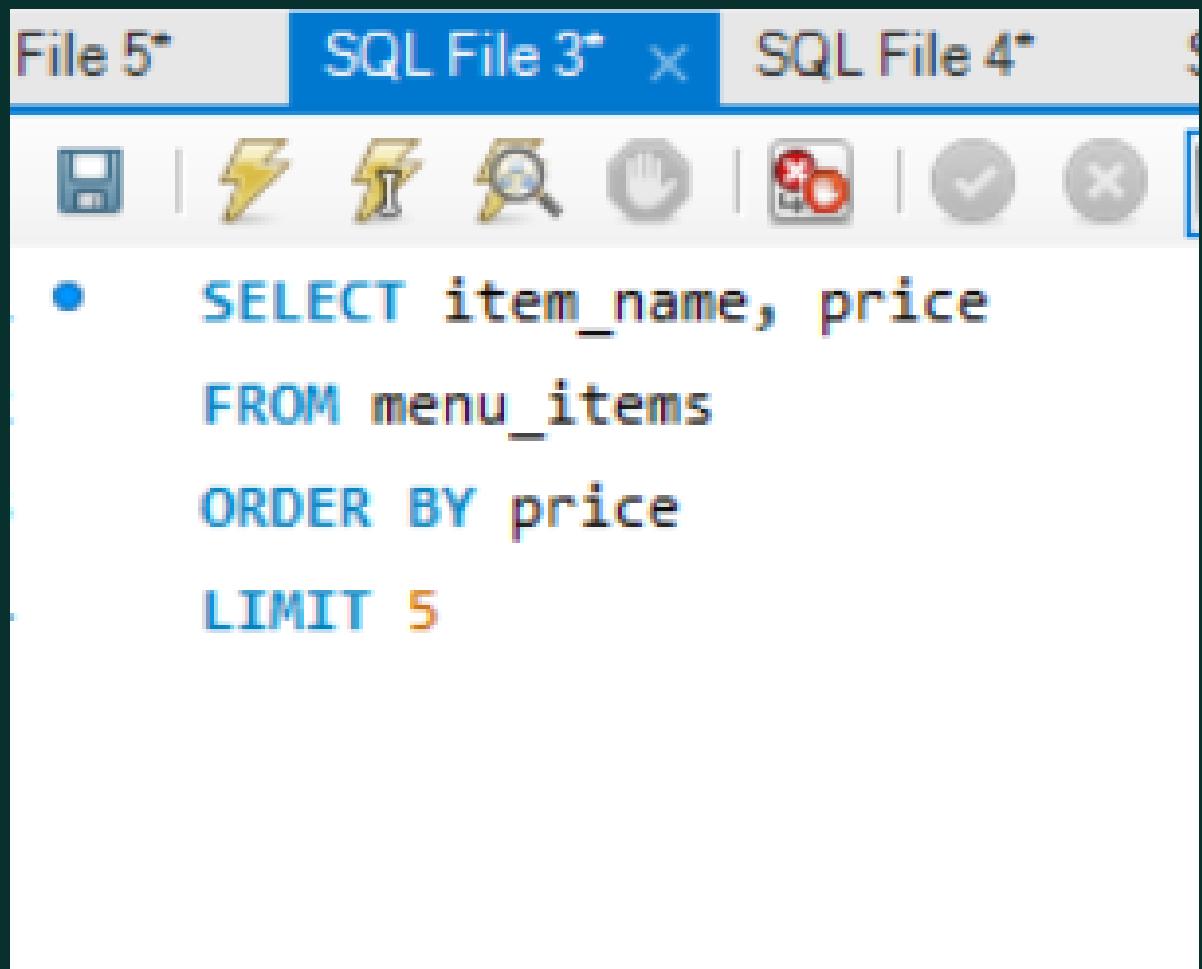
```
1 •   SELECT item_name, price  
2       FROM menu_items  
3       ORDER BY price DESC  
4       LIMIT 5
```

	item_name	price
▶	Shrimp Scampi	19.95
	Meat Lasagna	17.95
	Korean Beef Bowl	17.95
	Pork Ramen	17.95
	Spaghetti & Meatballs	17.95

The output here shown is the results for the most expensive items

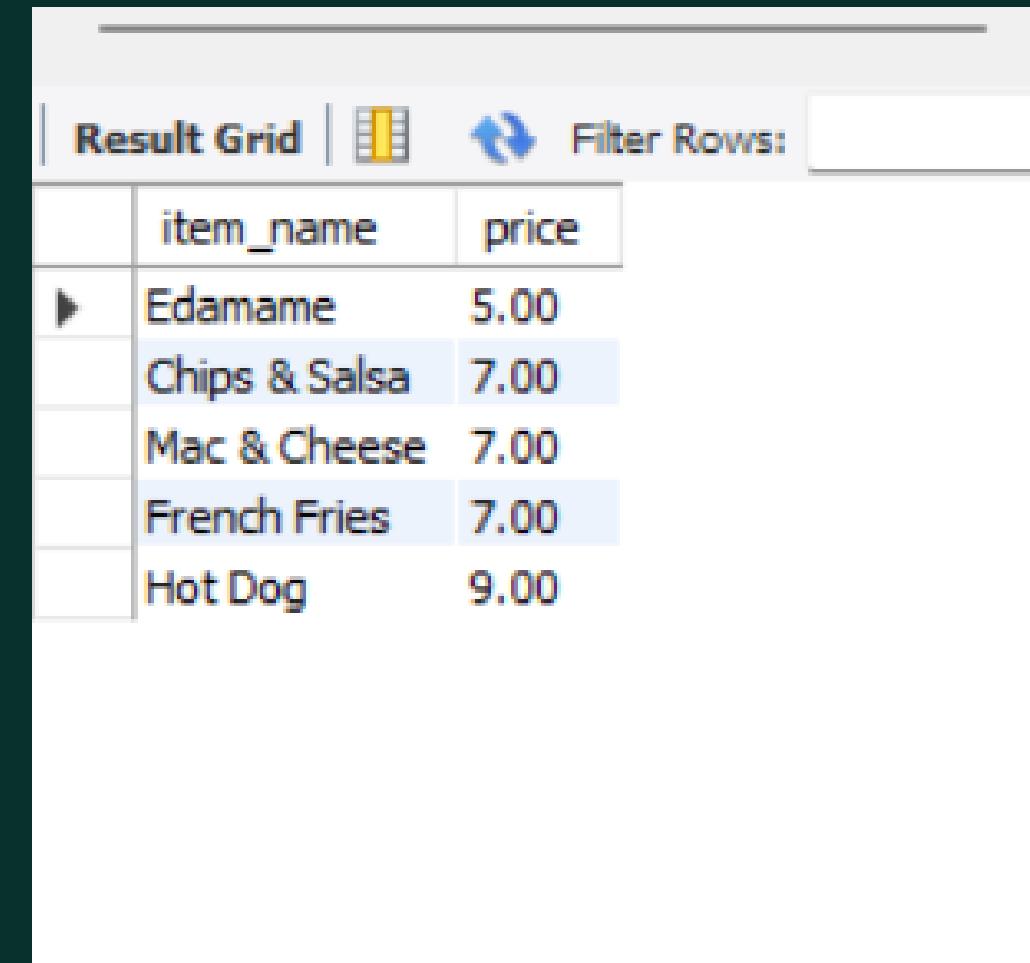
Analysis

Task 2 : What are the least and most expensive items on the menu?



The screenshot shows the MySQL Workbench interface. In the top bar, there are tabs for "File 5*", "SQL File 3*", and "SQL File 4*". Below the tabs is a toolbar with various icons. The main area contains a SQL query:

```
• SELECT item_name, price  
  FROM menu_items  
  ORDER BY price  
  LIMIT 5
```



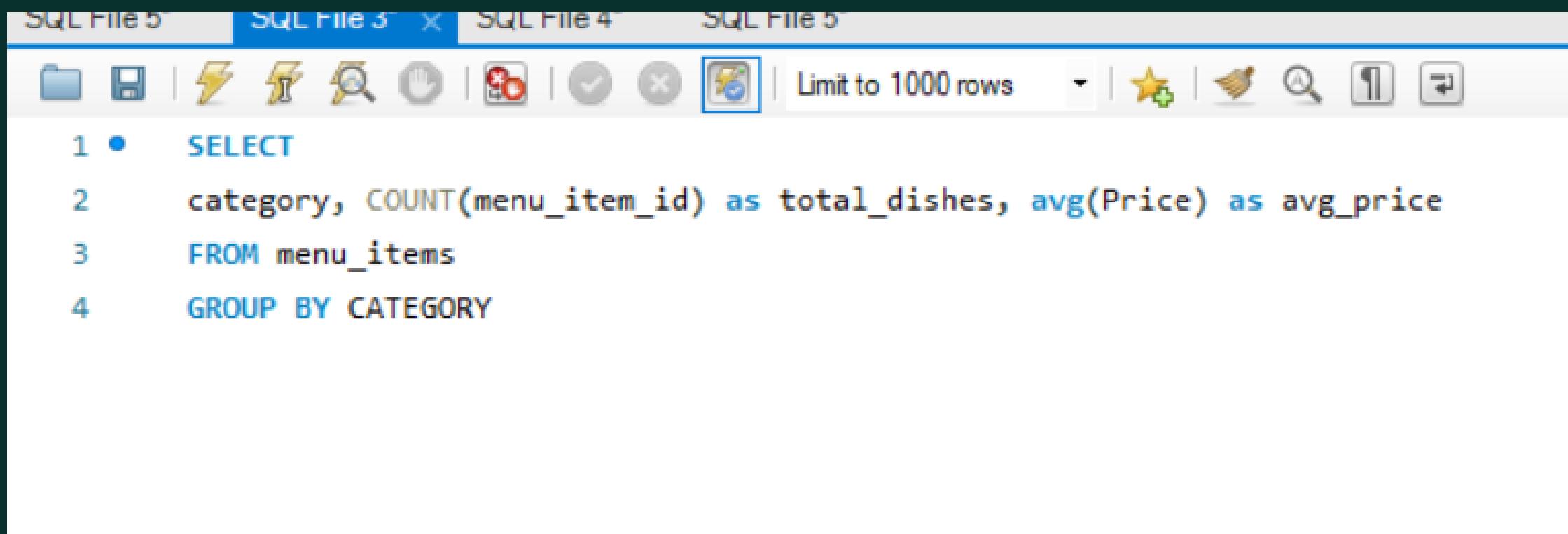
The screenshot shows the "Result Grid" window in MySQL Workbench. The grid displays the results of the executed SQL query. The columns are labeled "item_name" and "price". The data is as follows:

	item_name	price
▶	Edamame	5.00
	Chips & Salsa	7.00
	Mac & Cheese	7.00
	French Fries	7.00
	Hot Dog	9.00

The output here shown is the results for the least expensive items

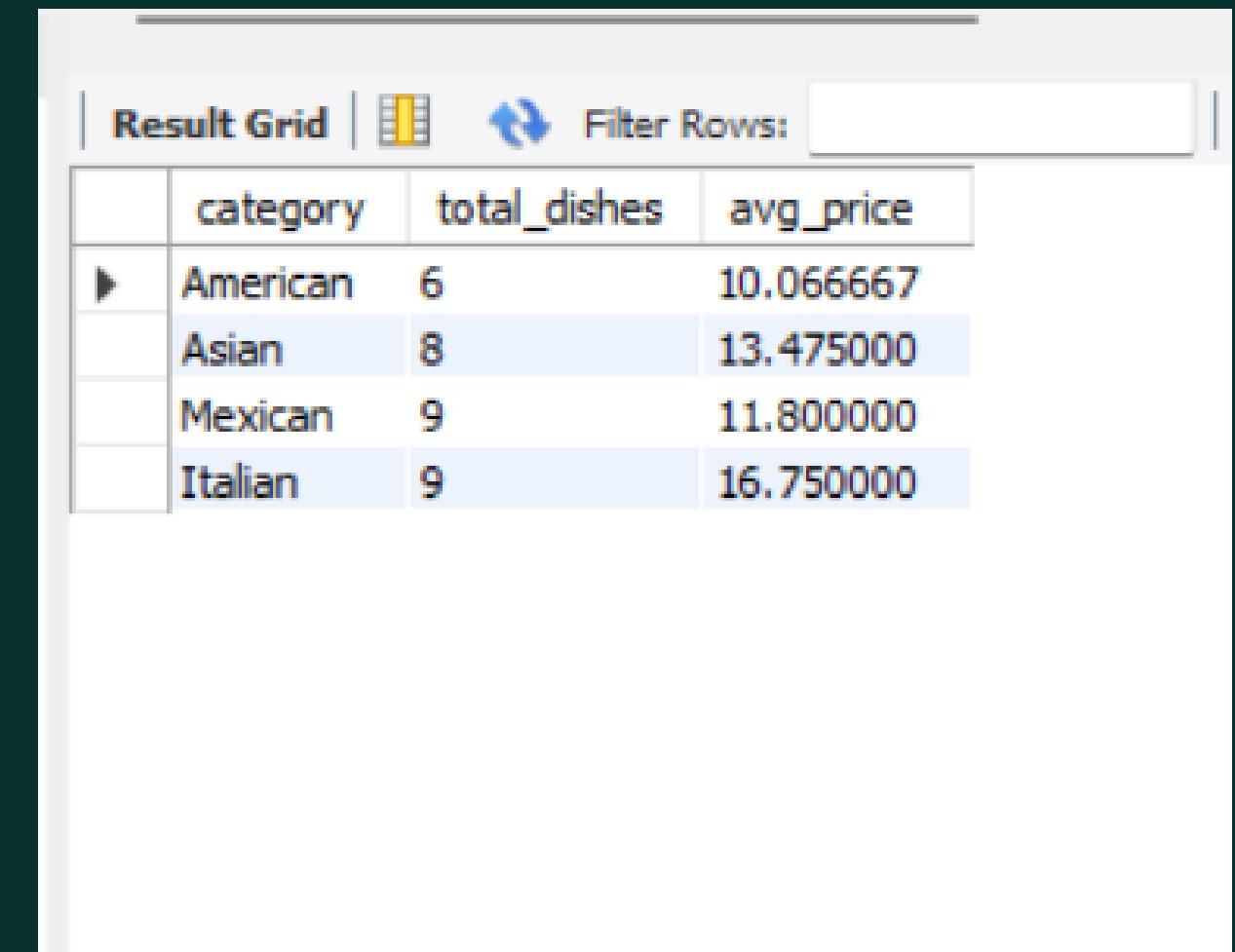
Analysis

Task 3 : How many dishes are in each Category? What is the Average Price?



The screenshot shows a MySQL Workbench interface. On the left, there are tabs for 'SQL File 5' (selected), 'SQL File 3', 'SQL File 4', and 'SQL File 5'. Below the tabs is a toolbar with various icons. The main area contains a SQL query:

```
1 • SELECT
2     category, COUNT(menu_item_id) as total_dishes, avg(Price) as avg_price
3     FROM menu_items
4     GROUP BY CATEGORY
```



The screenshot shows the 'Result Grid' tab in MySQL Workbench. The results of the query are displayed in a table:

	category	total_dishes	avg_price
▶	American	6	10.066667
	Asian	8	13.475000
	Mexican	9	11.800000
	Italian	9	16.750000

The output shows that the Average Price of Italian Dishes are the highest.

Analysis

Task 4 : Which orders has the most number of Items?

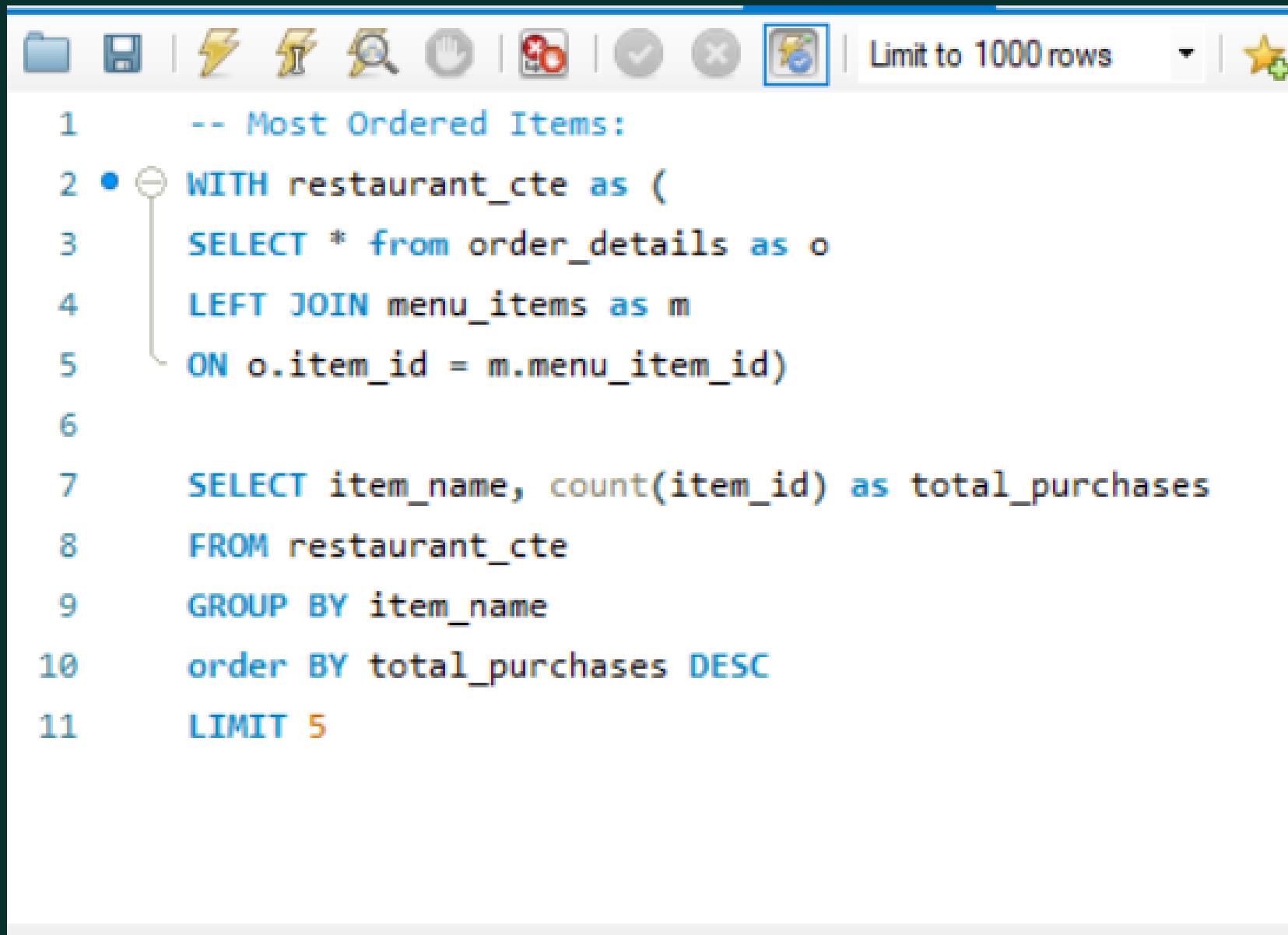
```
1 •   SELECT  
2       order_id, count(item_id) as total_items  
3   FROM order_details  
4   GROUP BY order_id  
5   ORDER BY total_items DESC  
6   LIMIT 10
```

	order_id	total_items
▶	2675	14
	443	14
	1957	14
	3473	14
	330	14
	440	14
	4305	14
	1274	13
	2126	13
	1734	13

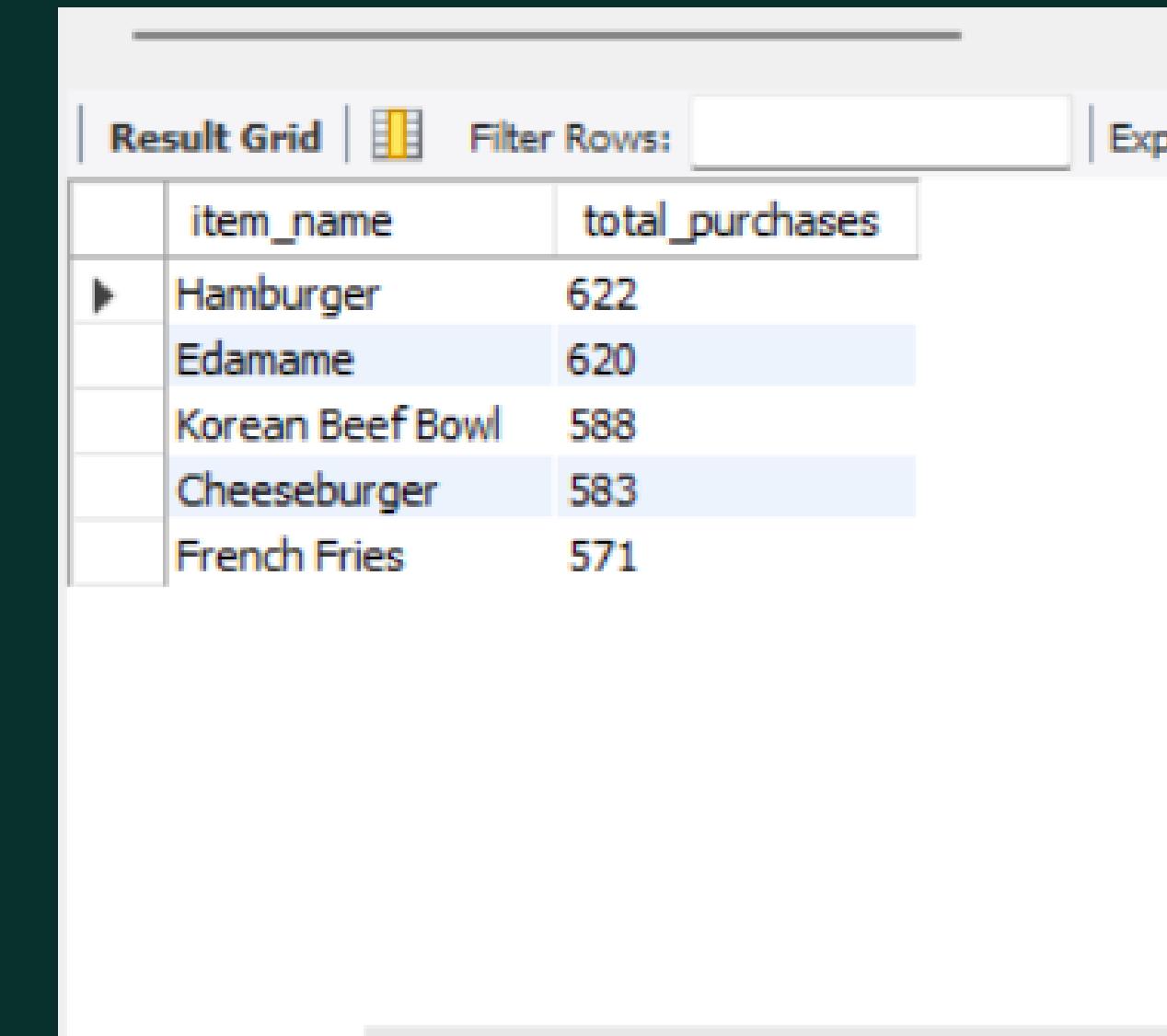
The output shows that the highest number of items were purchased is 14.

Analysis

Task 5 : What were the most and least ordered Items



```
1 -- Most Ordered Items:
2 • WITH restaurant_cte as (
3   SELECT * from order_details as o
4   LEFT JOIN menu_items as m
5   ON o.item_id = m.menu_item_id)
6
7   SELECT item_name, count(item_id) as total_purchases
8   FROM restaurant_cte
9   GROUP BY item_name
10  ORDER BY total_purchases DESC
11  LIMIT 5
```

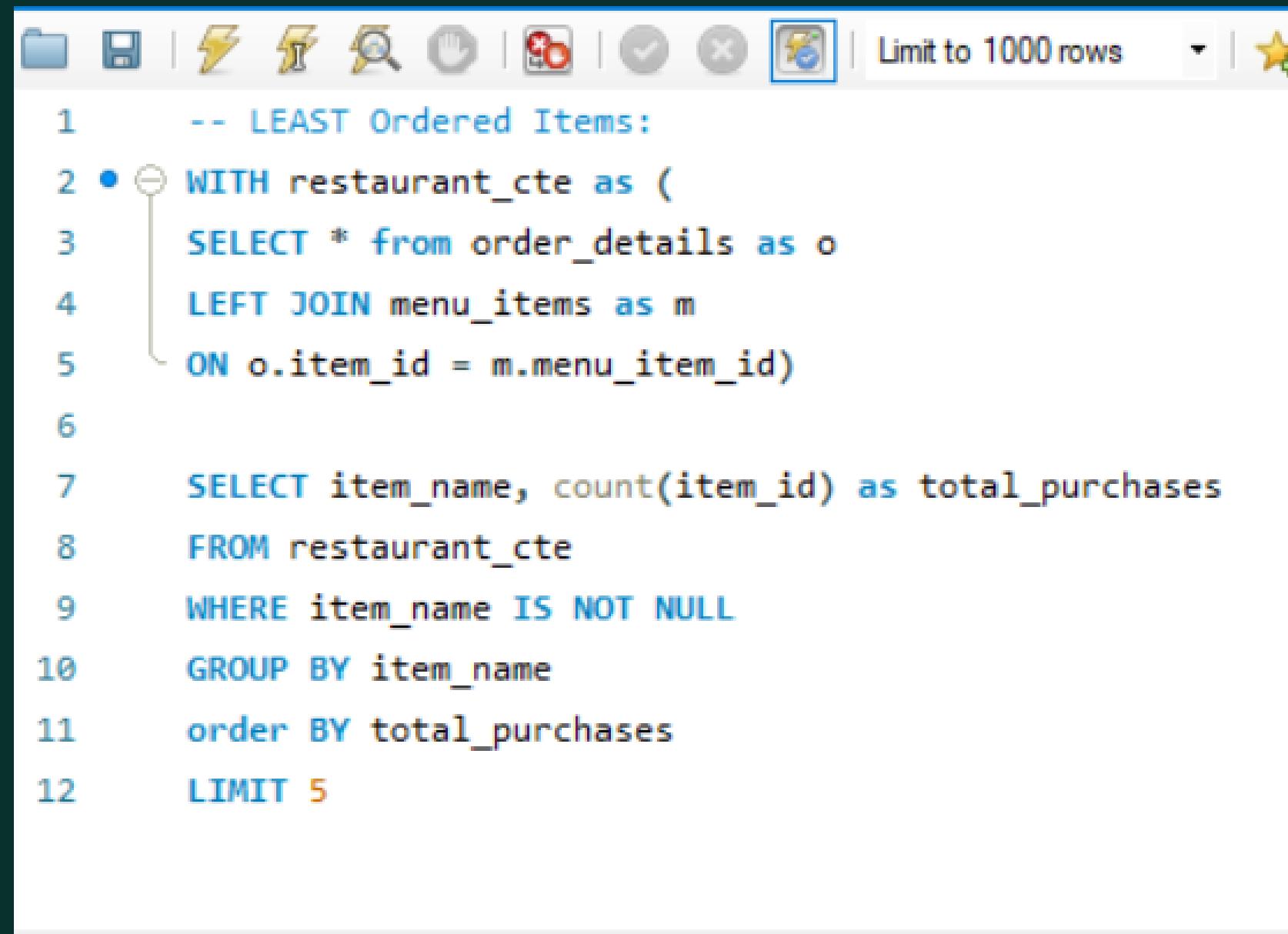


	item_name	total_purchases
▶	Hamburger	622
	Edamame	620
	Korean Beef Bowl	588
	Cheeseburger	583
	French Fries	571

The most ordered items was Hamburger with count of 622.

Analysis

Task 5 : What were the most and least ordered Items



```
1 -- LEAST Ordered Items:
2 • WITH restaurant_cte as (
3     SELECT * from order_details as o
4     LEFT JOIN menu_items as m
5     ON o.item_id = m.menu_item_id)
6
7     SELECT item_name, count(item_id) as total_purchases
8     FROM restaurant_cte
9     WHERE item_name IS NOT NULL
10    GROUP BY item_name
11    order BY total_purchases
12    LIMIT 5
```



	item_name	total_purchases
▶	Chicken Tacos	123
	Potstickers	205
	Cheese Lasagna	207
	Steak Tacos	214
	Cheese Quesadillas	233

The least ordered items were Chicken Tacos with count of only 123.

Analysis

Task 6: What are the top 5 orders that spent the most money?

SPENT MOST:

```
1  -- Top 5 orders that spent the most money
2 • WITH restaurant_cte as (
3   SELECT * from order_details as o
4   LEFT JOIN menu_items as m
5   ON o.item_id = m.menu_item_id)
6
7   SELECT order_id, SUM(price) as money_spent
8   FROM restaurant_cte
9   GROUP BY order_id
10  ORDER BY money_spent DESC
11  LIMIT 5
```

	order_id	money_spent
▶	440	192.15
	2075	191.05
	1957	190.10
	330	189.70
	2675	185.10

Analysis

Task 6: What are the top 5 orders that spent the most money?

SPENT LEAST:

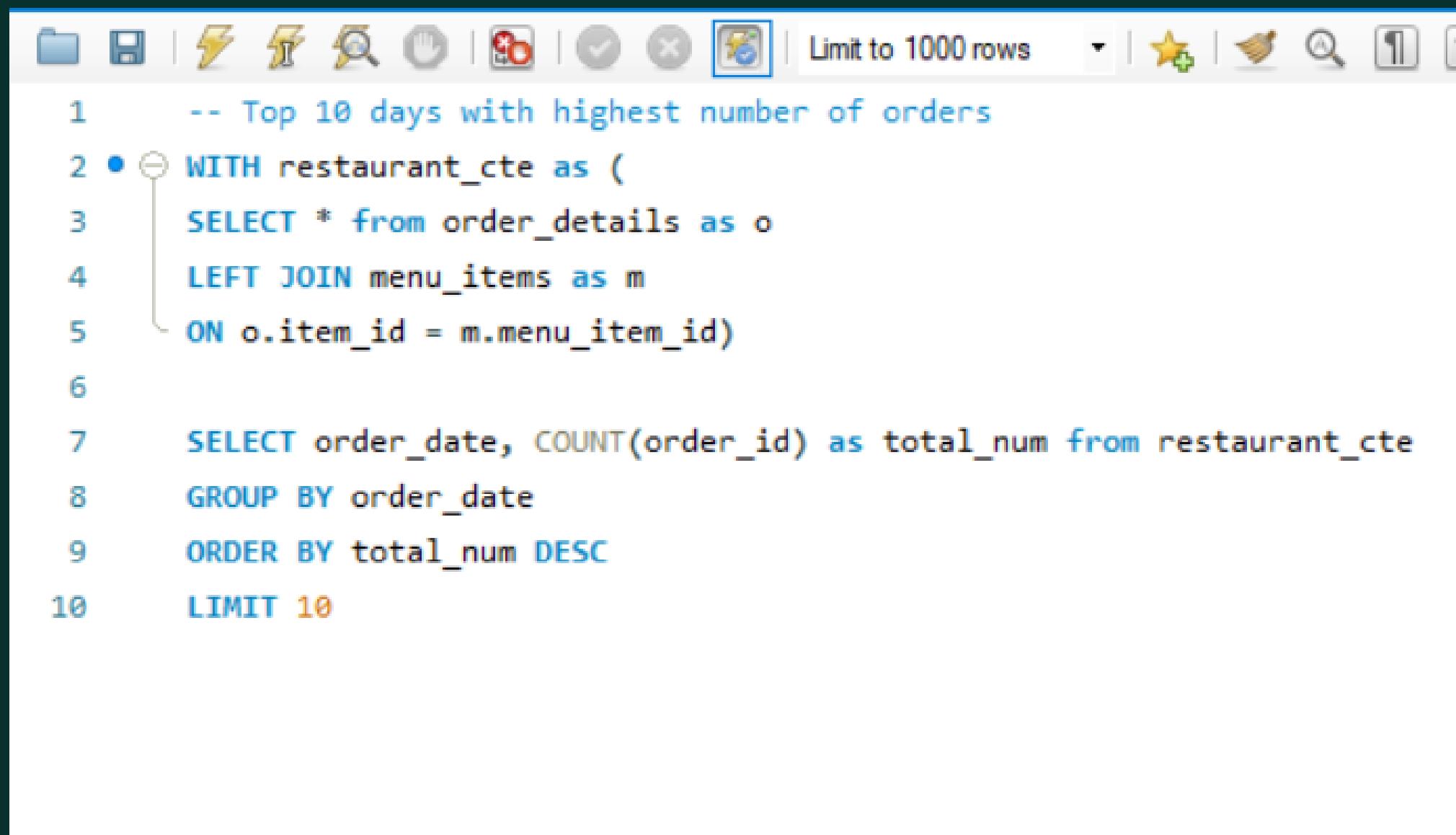
```
1 -- Top 5 orders that spent the least money
2 • WITH restaurant_cte as (
3     SELECT * from order_details as o
4     LEFT JOIN menu_items as m
5     ON o.item_id = m.menu_item_id)
6
7     SELECT order_id, SUM(price) as money_spent
8     FROM restaurant_cte
9     GROUP BY order_id
10    HAVING SUM(price) is NOT NULL
11    ORDER BY money_spent ASC
12    LIMIT 5
```

The screenshot shows a results grid titled "Result Grid" with a "Filter Rows:" input field. The grid has two columns: "order_id" and "money_spent". There are five rows of data, each representing an order with a total spending of 5.00.

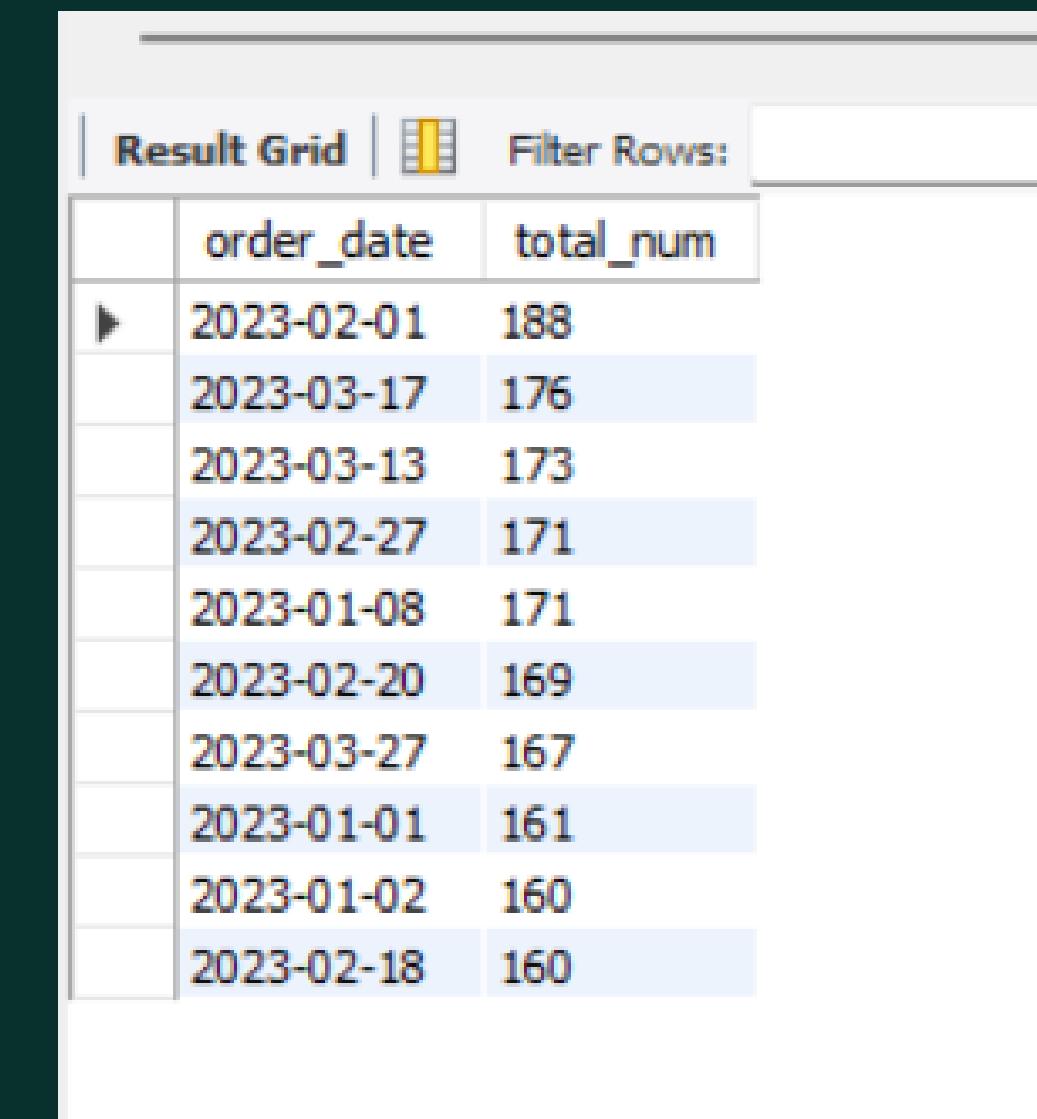
	order_id	money_spent
▶	4812	5.00
	4827	5.00
	4885	5.00
	4960	5.00
	5007	5.00

Analysis

Task 7: What are the highest performing days?



```
1 -- Top 10 days with highest number of orders
2 • WITH restaurant_cte as (
3   SELECT * from order_details as o
4   LEFT JOIN menu_items as m
5   ON o.item_id = m.menu_item_id)
6
7   SELECT order_date, COUNT(order_id) as total_num from restaurant_cte
8   GROUP BY order_date
9   ORDER BY total_num DESC
10  LIMIT 10
```



	order_date	total_num
▶	2023-02-01	188
	2023-03-17	176
	2023-03-13	173
	2023-02-27	171
	2023-01-08	171
	2023-02-20	169
	2023-03-27	167
	2023-01-01	161
	2023-01-02	160
	2023-02-18	160



THANK YOU

