



SQL PROJECT

Pizza sales Analysis

We would like to offer you a stylish and reasonable presentation that
will help you to promote your business



Introduction

Hello, my name is Chaitanya Kohad. In today's competitive food industry, understanding sales patterns and customer preferences is crucial for business success. This project focuses on analyzing pizza sales data to uncover insights that can drive better decision-making and optimize sales strategies. Utilizing SQL, a powerful language for managing and manipulating relational databases, this analysis aims to identify trends, provide actionable insights, and ultimately enhance pizza sales.

Objective

- A comprehensive understanding of sales patterns and customer preferences.
- Identification of top-selling products and key sales periods.
- Recommendations for inventory management to reduce waste and stockouts.
- Marketing strategies to boost sales during off-peak periods



QUESTIONS

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

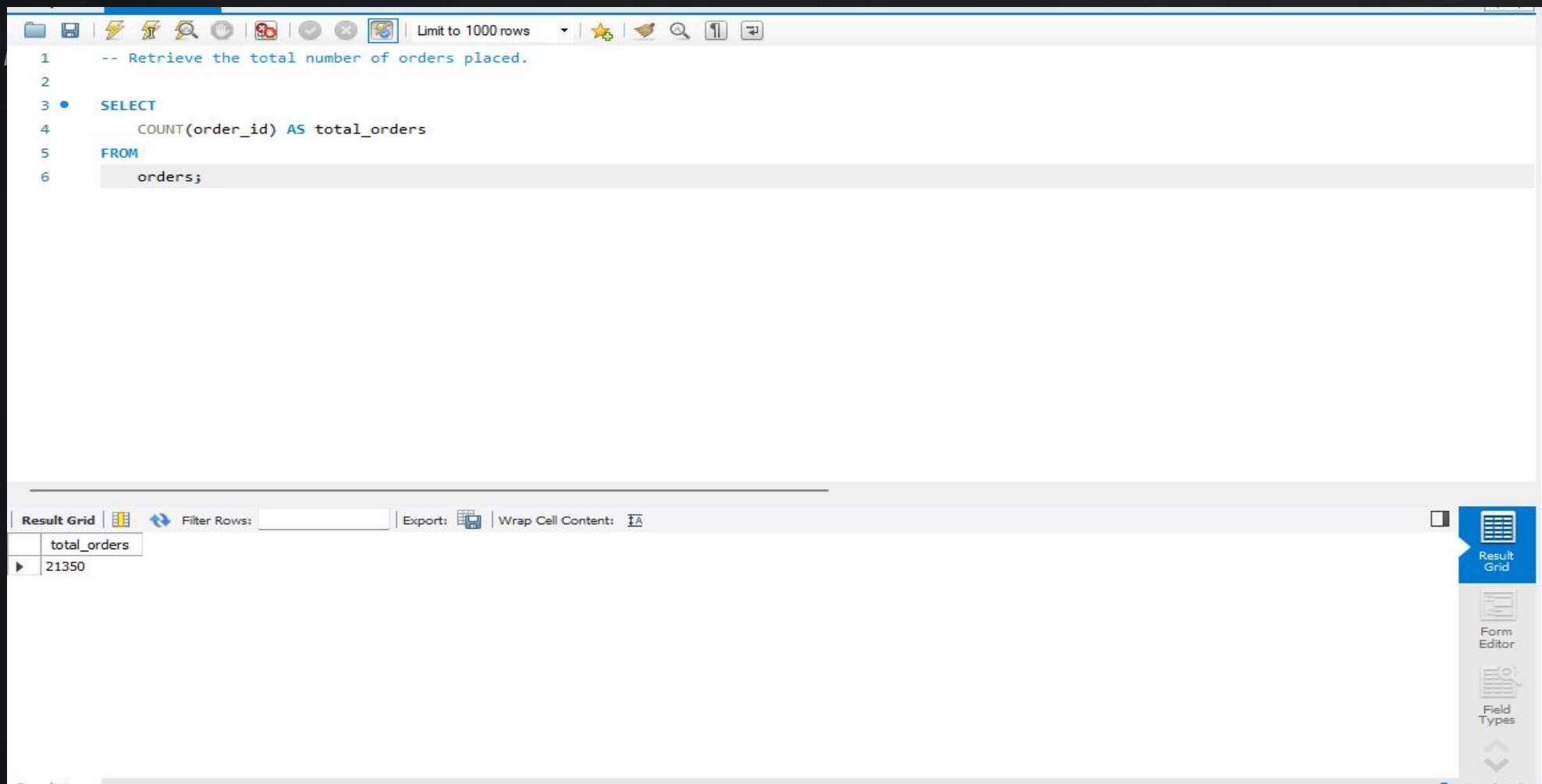
Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.



BASIC

- Retrieve the total number of orders placed.



The screenshot shows a SQL query editor interface. The query is as follows:

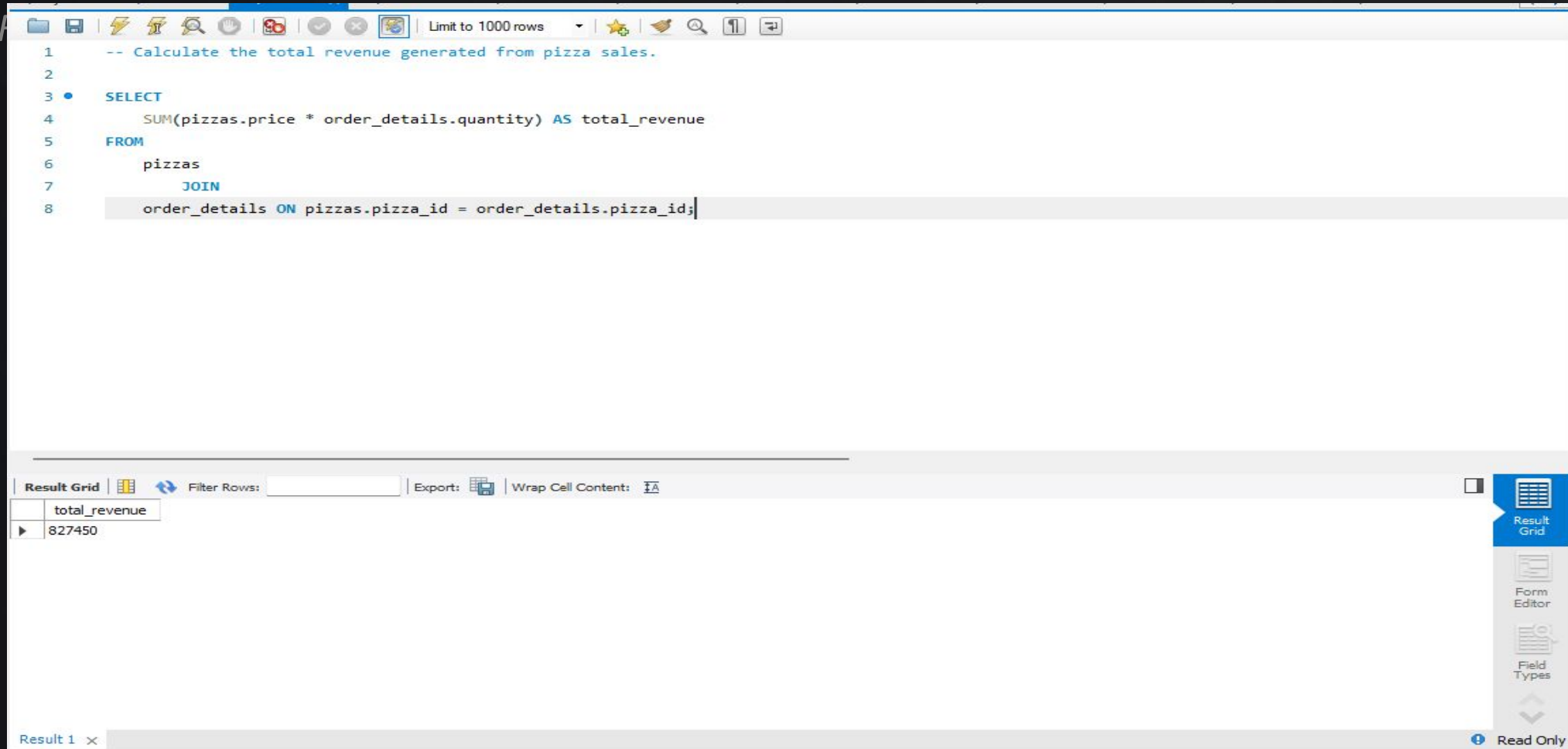
```
1  -- Retrieve the total number of orders placed.
2
3  • SELECT
4      COUNT(order_id) AS total_orders
5  FROM
6      orders;
```

Below the query editor, the result grid is displayed, showing a single row with the value 21350 for the column total_orders.

total_orders
21350

The interface includes a toolbar at the top with various icons and a 'Limit to 1000 rows' dropdown. The bottom right corner features a sidebar with options for 'Result Grid', 'Form Editor', and 'Field Types'.

- Calculate the total revenue generated from pizza sales.



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and settings. A dropdown menu indicates "Limit to 1000 rows". The query is as follows:

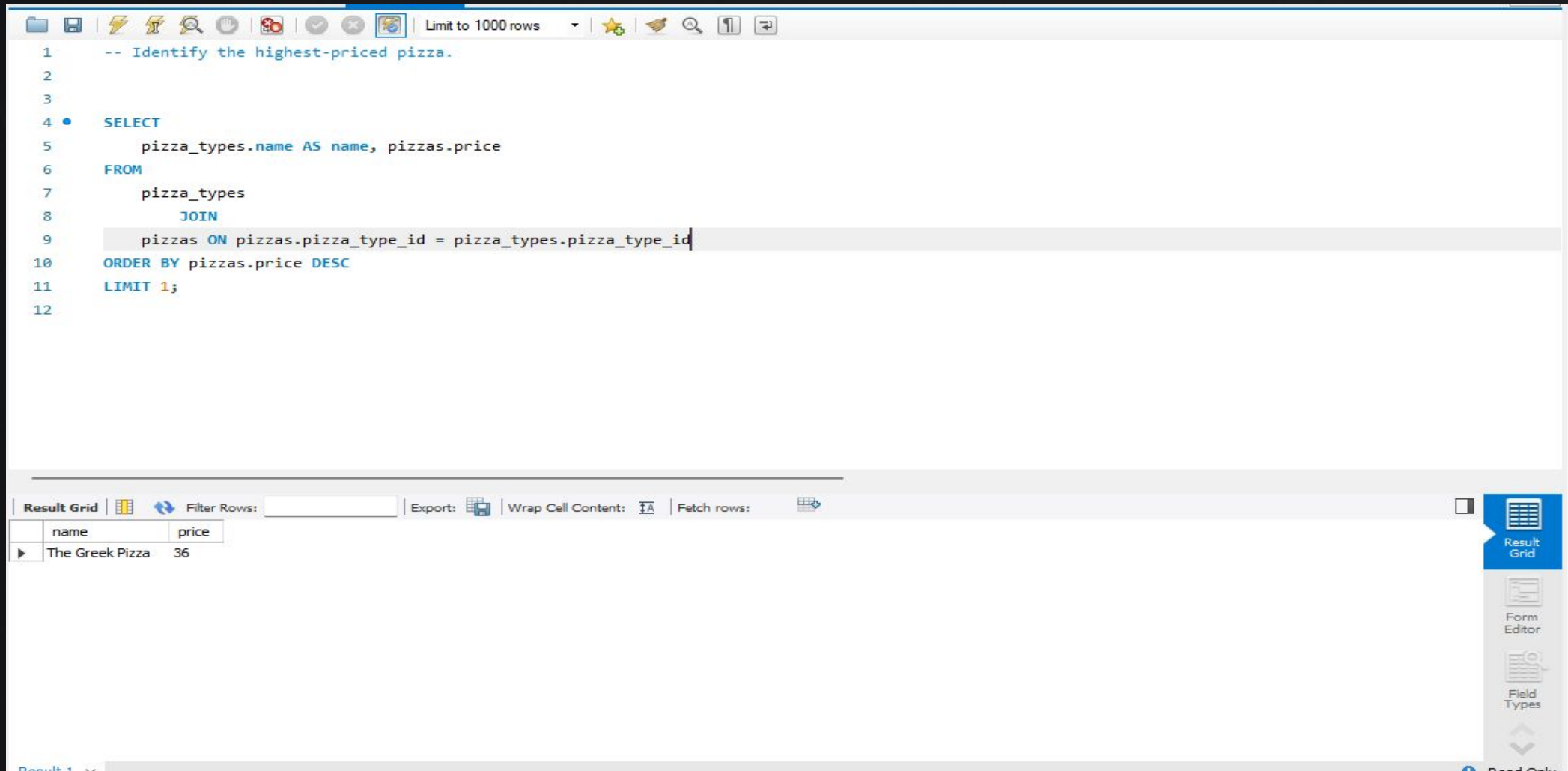
```
1  -- Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4      SUM(pizzas.price * order_details.quantity) AS total_revenue
5  FROM
6      pizzas
7      JOIN
8      order_details ON pizzas.pizza_id = order_details.pizza_id;
```

Below the query editor, the "Result Grid" tab is active, displaying the following data:

total_revenue
827450

On the right side of the interface, there is a vertical toolbar with icons for "Result Grid" (selected), "Form Editor", and "Field Types". At the bottom right, a "Read Only" status is indicated.

- Identify the highest-priced pizza.



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 the following query:

```
1  -- Identify the highest-priced pizza.
2
3
4  • SELECT
5      pizza_types.name AS name, pizzas.price
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10 ORDER BY pizzas.price DESC
11 LIMIT 1;
12
```

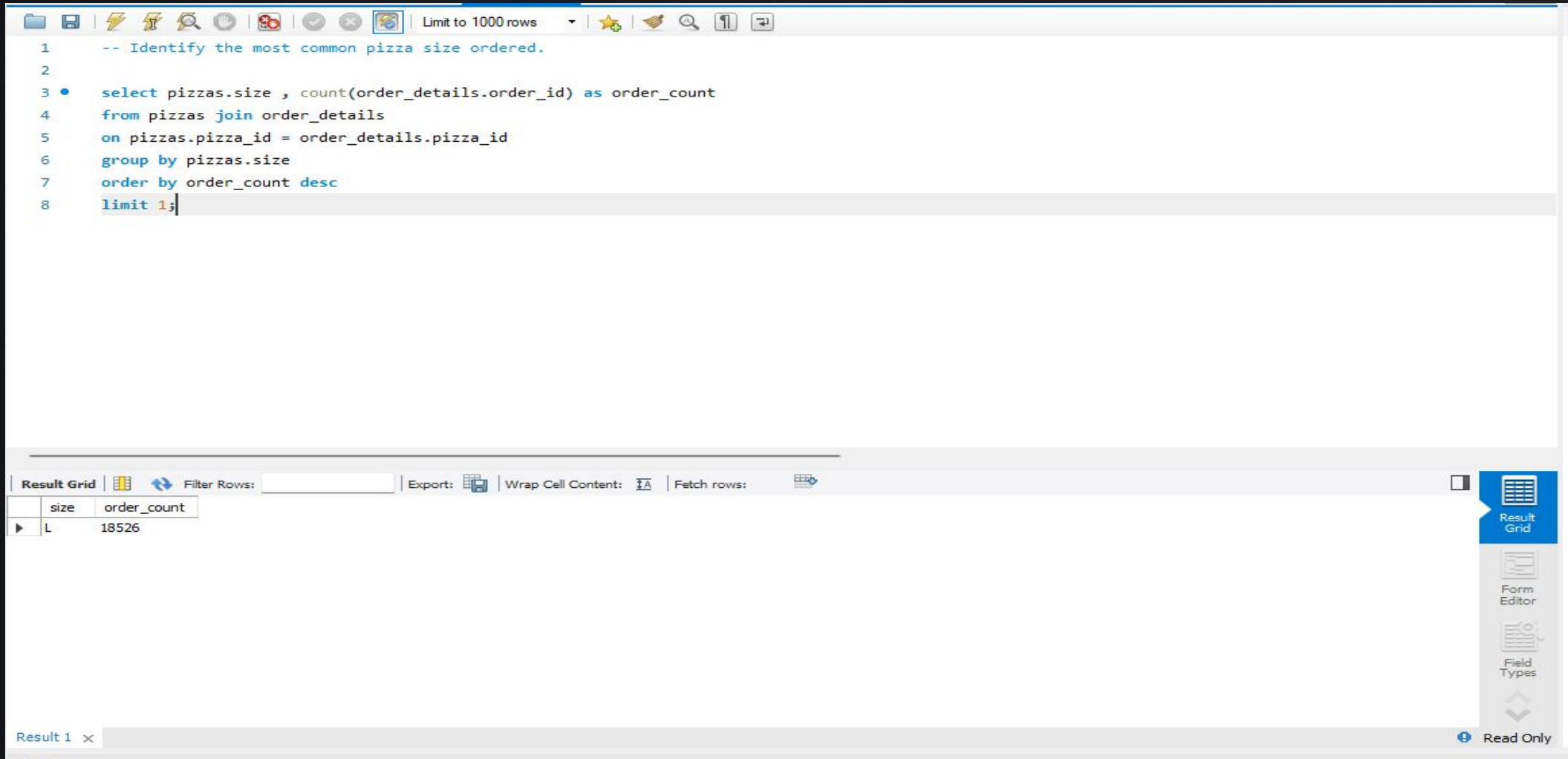
The bottom toolbar includes 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The 'Result Grid' is active, displaying the following data:

	name	price
▶	The Greek Pizza	36

On the right side of the IDE, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom right, there is a 'Read Only' indicator.



- Identify the most common pizza size ordered.



```
1  -- Identify the most common pizza size ordered.
2
3  •  select pizzas.size , count(order_details.order_id) as order_count
4     from pizzas join order_details
5     on pizzas.pizza_id = order_details.pizza_id
6     group by pizzas.size
7     order by order_count desc
8     limit 1;
```

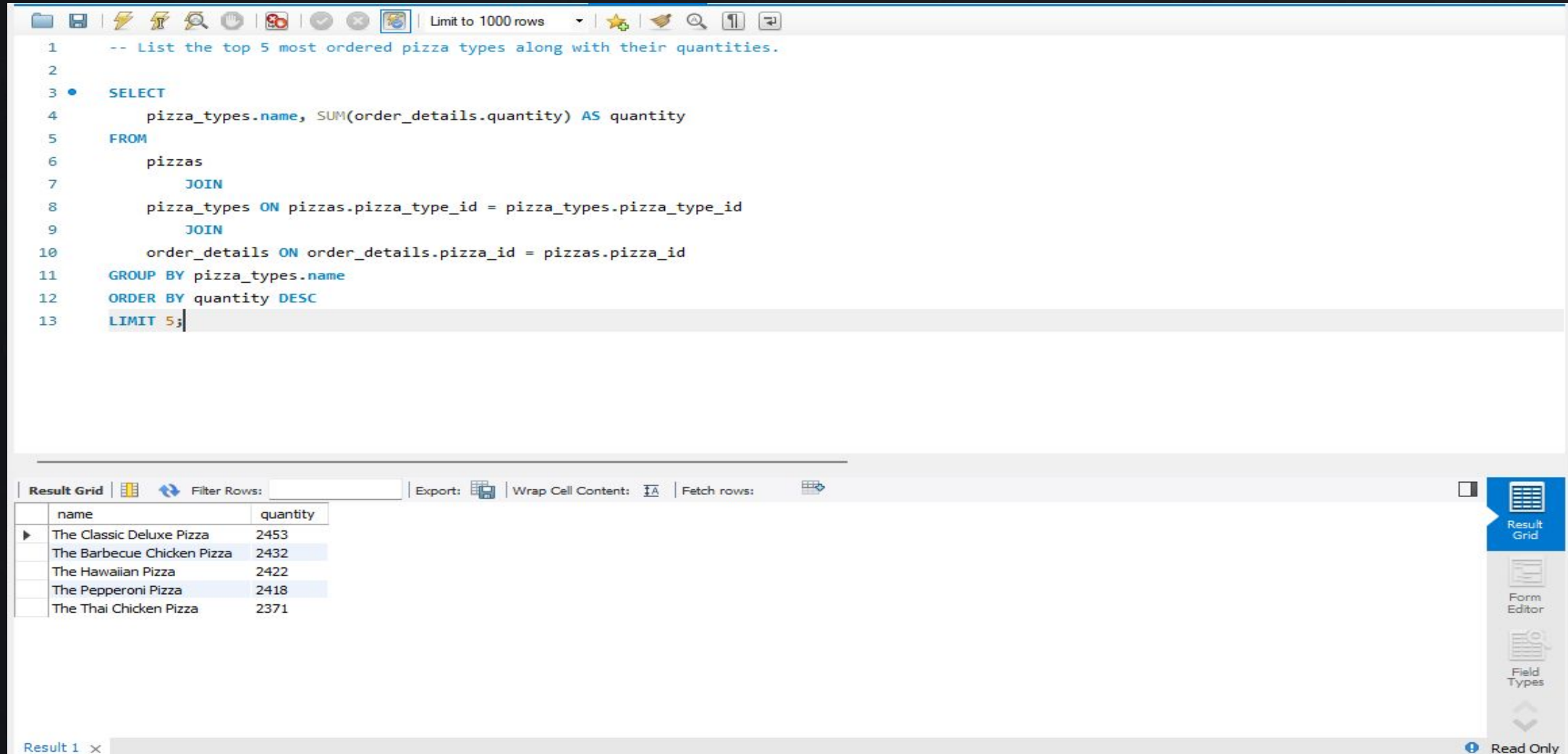
Result Grid

	size	order_count
▶	L	18526

Result 1 x

Read Only

- List the top 5 most ordered pizza types along with their quantities.



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 1000 rows' dropdown. The SQL query is as follows:

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2
3  • SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizzas
7      JOIN
8      pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9      JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY quantity DESC
13 LIMIT 5;
```

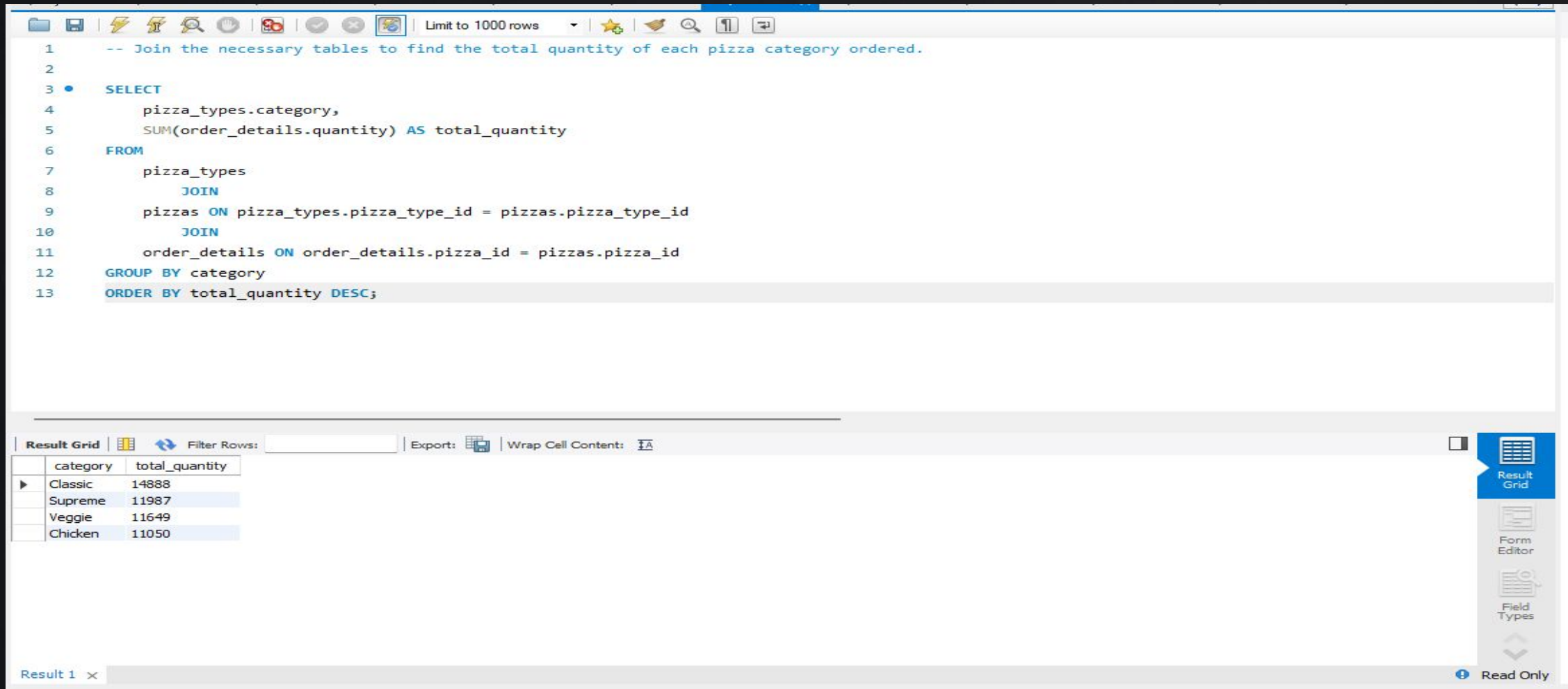
Below the query editor, the 'Result Grid' tab is active, displaying the following data:

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar indicates 'Result 1' and 'Read Only'.

Intermediate

- Join the necessary tables to find the total quantity of each pizza category ordered.



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

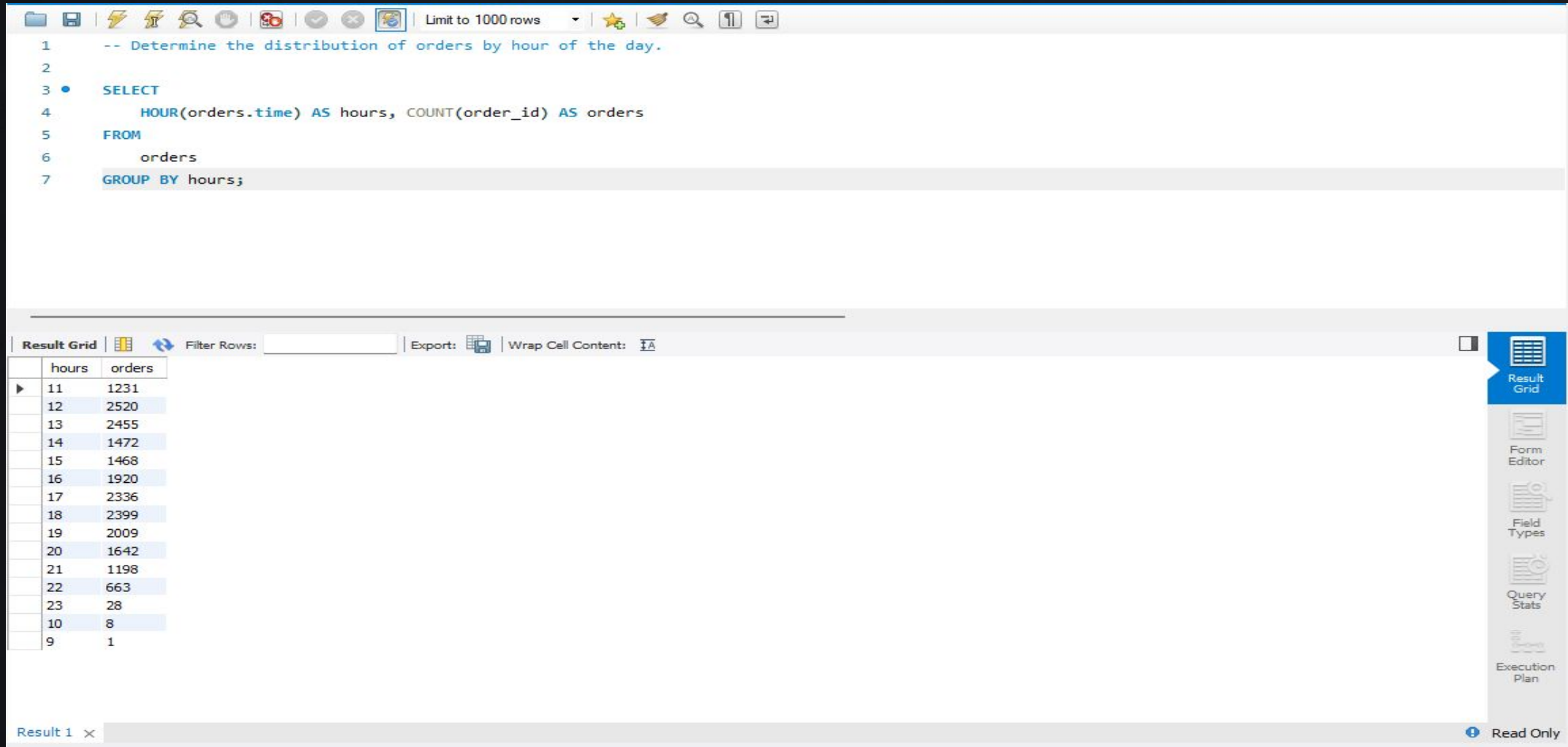
```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3  •  SELECT
4      pizza_types.category,
5      SUM(order_details.quantity) AS total_quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY category
13 ORDER BY total_quantity DESC;
```

Below the editor, the 'Result Grid' tab is active, displaying the query results in a table:

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

On the right side of the IDE, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom right, there is a 'Read Only' status indicator.

- Determine the distribution of orders by hour of the day.



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area:

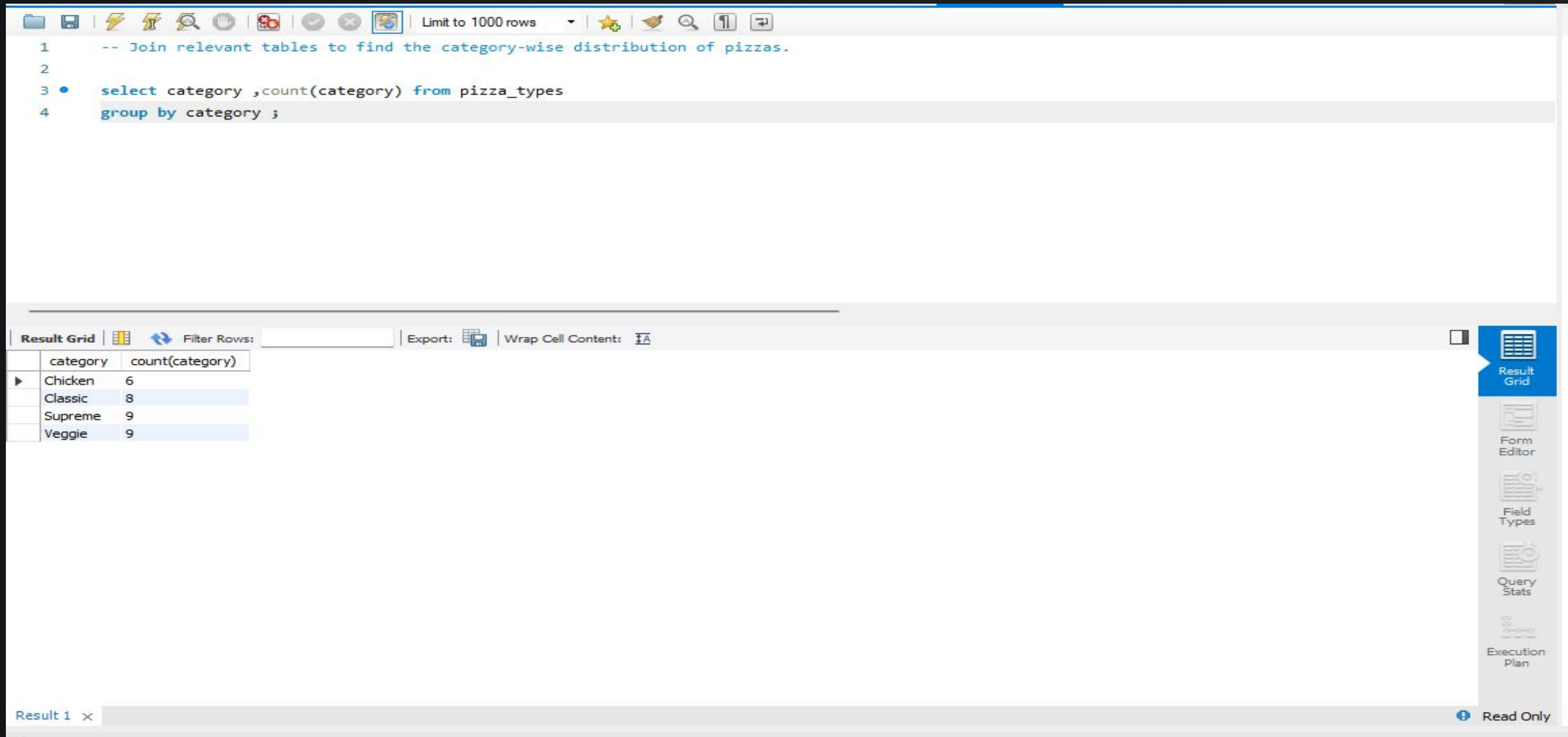
```
1  -- Determine the distribution of orders by hour of the day.  
2  
3  •  SELECT  
4      HOUR(orders.time) AS hours, COUNT(order_id) AS orders  
5  FROM  
6      orders  
7  GROUP BY hours;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query in a table. The table has two columns: "hours" and "orders". The data is as follows:

hours	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

On the right side of the IDE, there is a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", "Query Stats", and "Execution Plan". At the bottom right, there is a "Read Only" status indicator.

- Join relevant tables to find the category-wise distribution of pizzas.



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

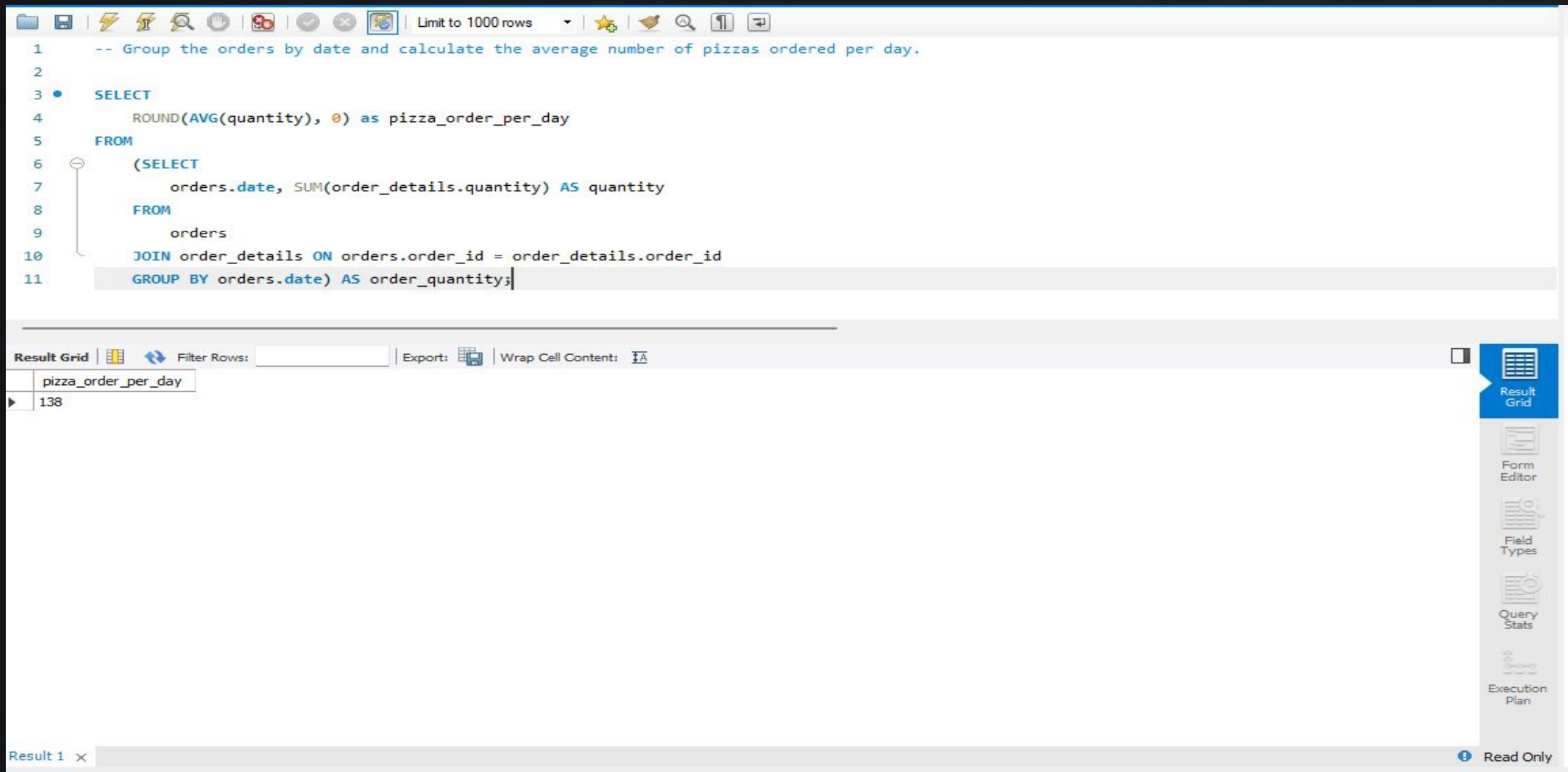
```
1  -- Join relevant tables to find the category-wise distribution of pizzas.
2
3  •  select category ,count(category) from pizza_types
4     group by category ;
```

Below the query editor, the 'Result Grid' tab is active, displaying the following data:

	category	count(category)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom status bar shows 'Result 1' and 'Read Only'.

- Group the orders by date and calculate the average number of pizzas ordered per day.



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

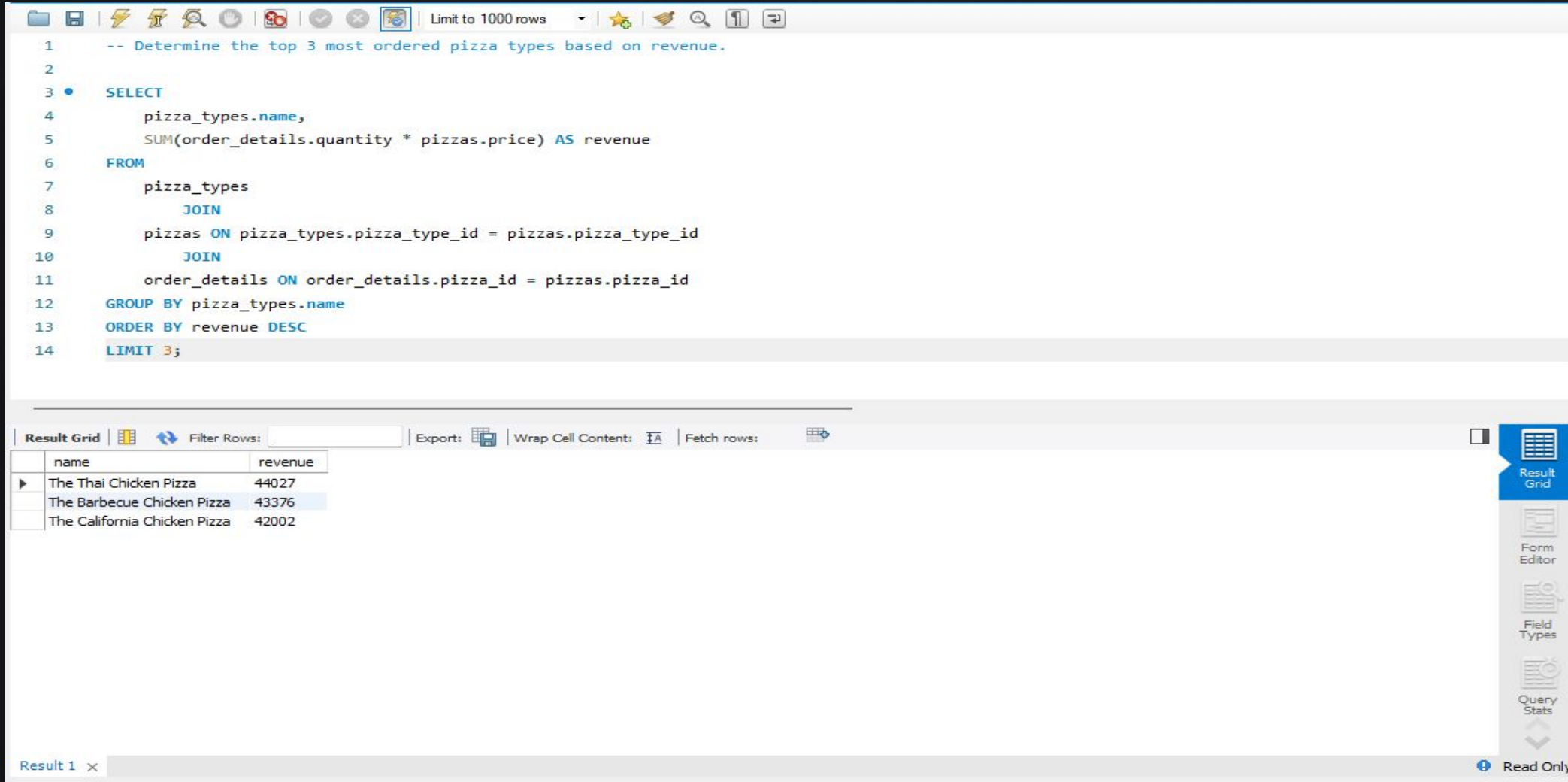
```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3  • SELECT
4      ROUND(AVG(quantity), 0) as pizza_order_per_day
5  FROM
6      (SELECT
7          orders.date, SUM(order_details.quantity) AS quantity
8      FROM
9          orders
10     JOIN order_details ON orders.order_id = order_details.order_id
11     GROUP BY orders.date) AS order_quantity;
```

Below the editor, the 'Result Grid' tab is active, showing a single row of results:

pizza_order_per_day
138

The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom status bar indicates 'Result 1' and 'Read Only'.

- Determine the top 3 most ordered pizza types based on revenue.



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 the following query:

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3  •  SELECT
4      pizza_types.name,
5      SUM(order_details.quantity * pizzas.price) AS revenue
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

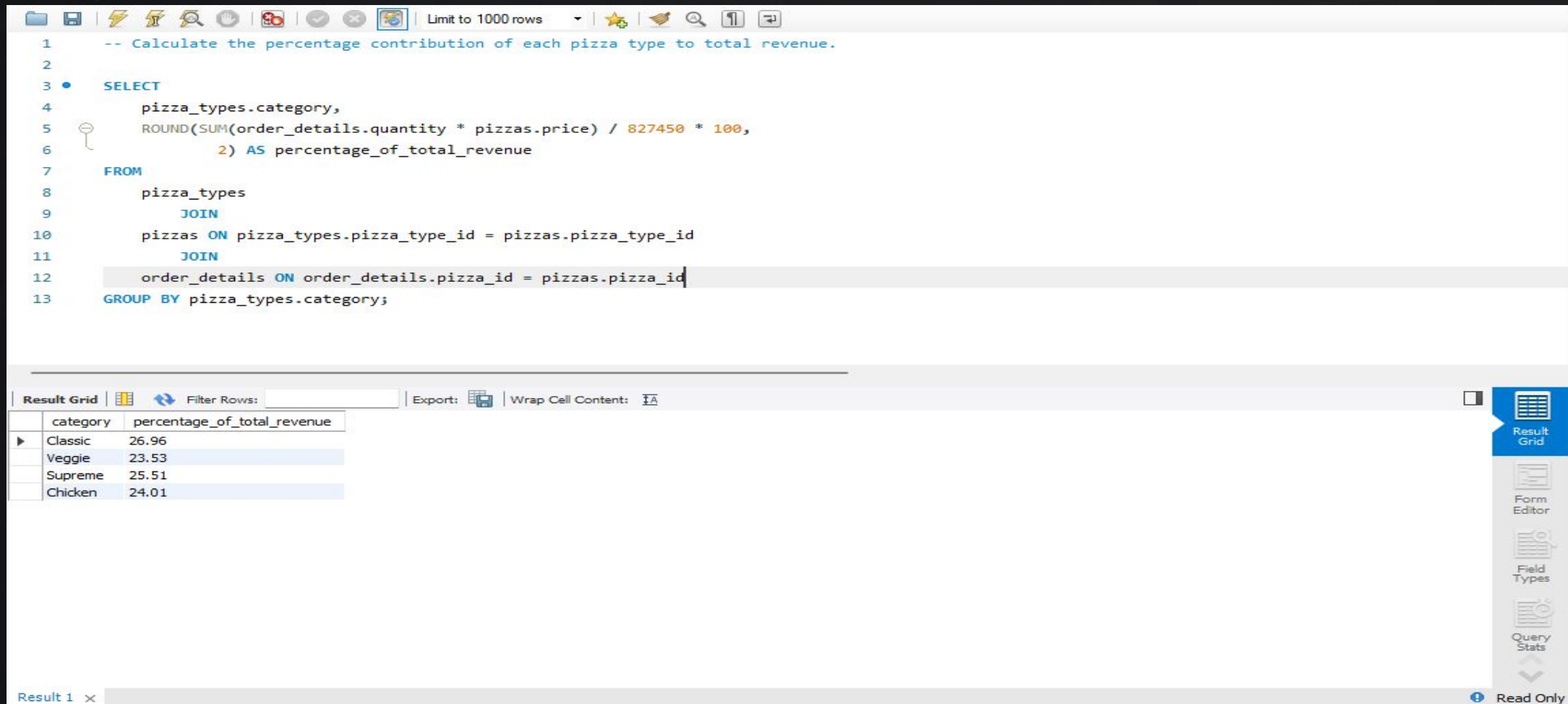
Below the editor, the 'Result Grid' tab is active, displaying the query results in a table:

	name	revenue
▶	The Thai Chicken Pizza	44027
	The Barbecue Chicken Pizza	43376
	The California Chicken Pizza	42002

The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The bottom status bar indicates 'Result 1' and 'Read Only'.

Advanced

- Calculate the percentage contribution of each pizza type to total revenue.



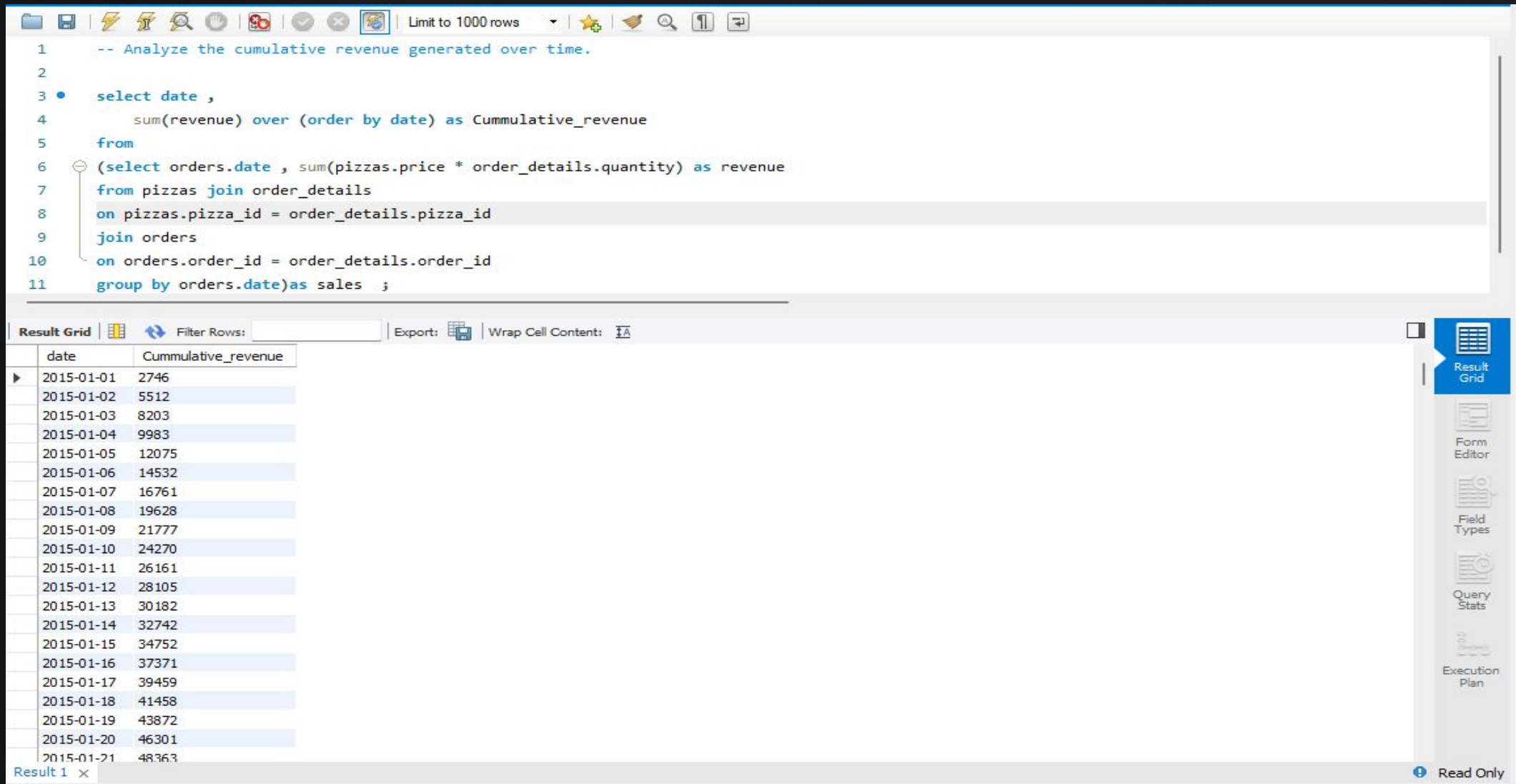
```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3  • SELECT
4      pizza_types.category,
5      ROUND(SUM(order_details.quantity * pizzas.price) / 827450 * 100,
6            2) AS percentage_of_total_revenue
7  FROM
8      pizza_types
9      JOIN
10     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11     JOIN
12     order_details ON order_details.pizza_id = pizzas.pizza_id
13 GROUP BY pizza_types.category;
```

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

	category	percentage_of_total_revenue
▶	Classic	26.96
	Veggie	23.53
	Supreme	25.51
	Chicken	24.01

Result 1 x | Read Only

- Analyze the cumulative revenue generated over time.



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

```
1  -- Analyze the cumulative revenue generated over time.
2
3  • select date ,
4      sum(revenue) over (order by date) as Cumulative_revenue
5  from
6  (select orders.date , sum(pizzas.price * order_details.quantity) as revenue
7   from pizzas join order_details
8   on pizzas.pizza_id = order_details.pizza_id
9   join orders
10  on orders.order_id = order_details.order_id
11  group by orders.date)as sales ;
```

Below the editor is the 'Result Grid' tab, which shows the query results in a table with two columns: 'date' and 'Cumulative_revenue'. The results show a daily increase in revenue from 2015-01-01 to 2015-01-21. The right sidebar contains icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom status bar indicates 'Read Only'.

date	Cumulative_revenue
2015-01-01	2746
2015-01-02	5512
2015-01-03	8203
2015-01-04	9983
2015-01-05	12075
2015-01-06	14532
2015-01-07	16761
2015-01-08	19628
2015-01-09	21777
2015-01-10	24270
2015-01-11	26161
2015-01-12	28105
2015-01-13	30182
2015-01-14	32742
2015-01-15	34752
2015-01-16	37371
2015-01-17	39459
2015-01-18	41458
2015-01-19	43872
2015-01-20	46301
2015-01-21	48363



THANK
YOU