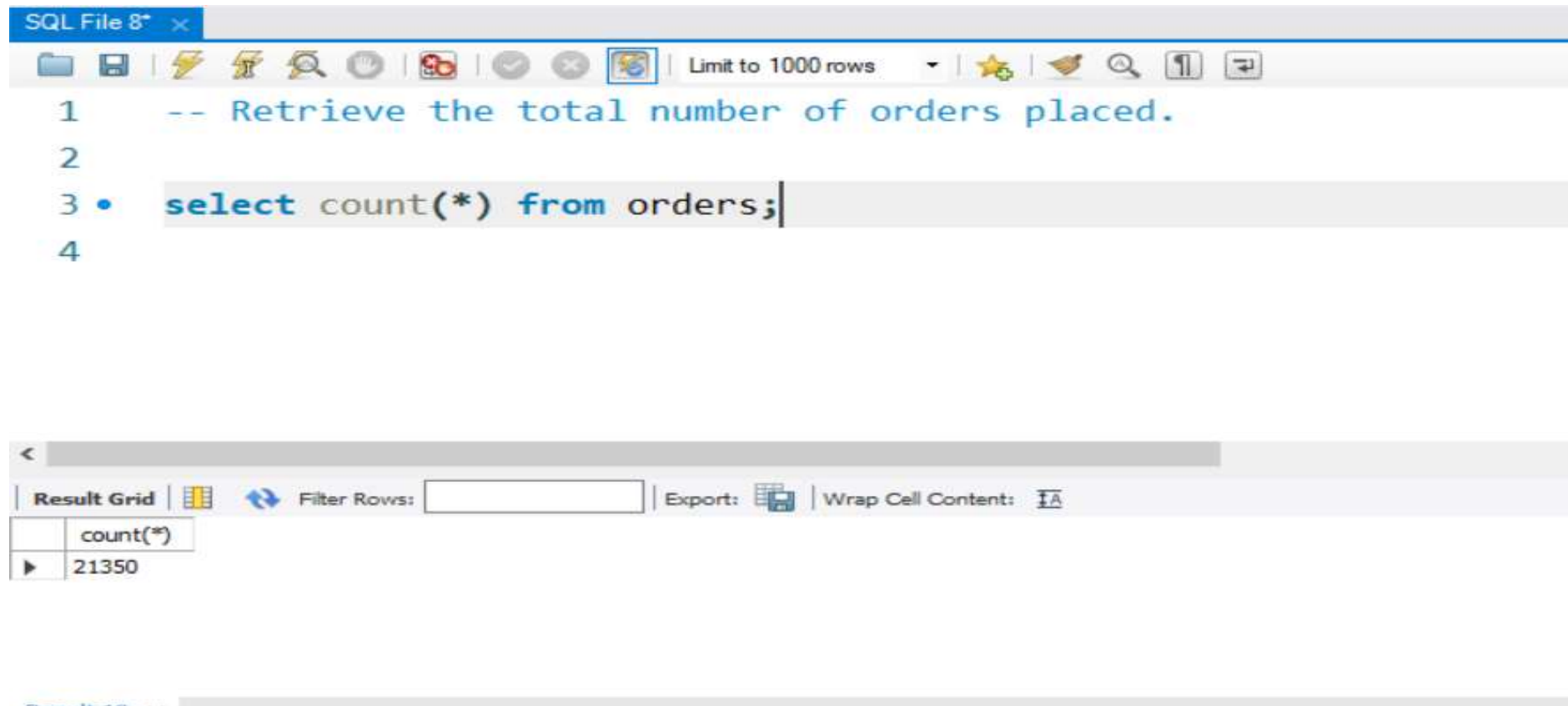




ANALYSING PIZZA **SALES DATA USING** **SQL**

#QUERY 1



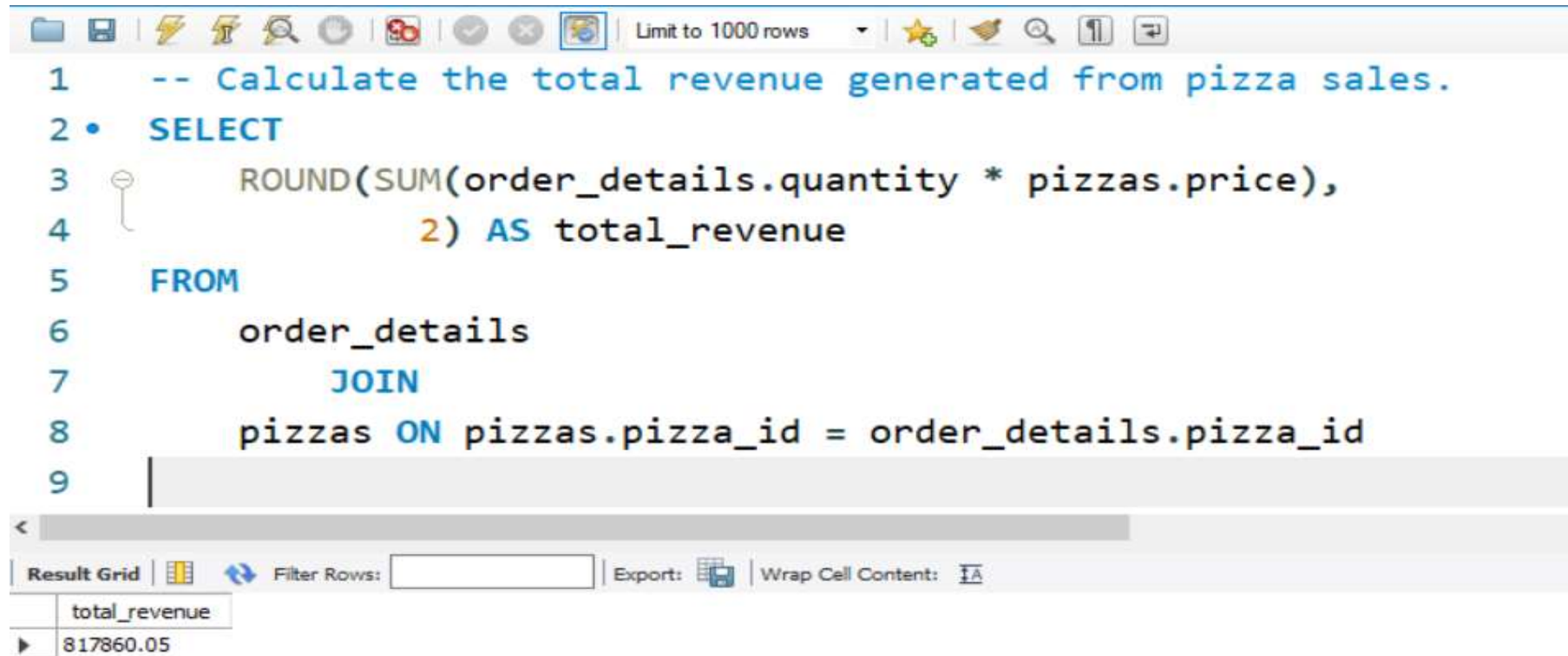
The screenshot shows a SQL IDE window titled "SQL File 8* x". The toolbar includes icons for file operations, execution, and settings. The query editor contains the following SQL code:

```
1  -- Retrieve the total number of orders placed.  
2  
3  • select count(*) from orders;  
4
```

Below the query editor, the "Result Grid" is displayed. It shows a single row with the column name "count(*)" and the value "21350". The toolbar for the result grid includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox.

count(*)
21350

#QUERY 2



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

```
1  -- Calculate the total revenue generated from pizza sales.
2  • SELECT
3      ROUND(SUM(order_details.quantity * pizzas.price),
4             2) AS total_revenue
5  FROM
6      order_details
7      JOIN
8      pizzas ON pizzas.pizza_id = order_details.pizza_id
9
```

Below the query editor, the results are displayed in a table:

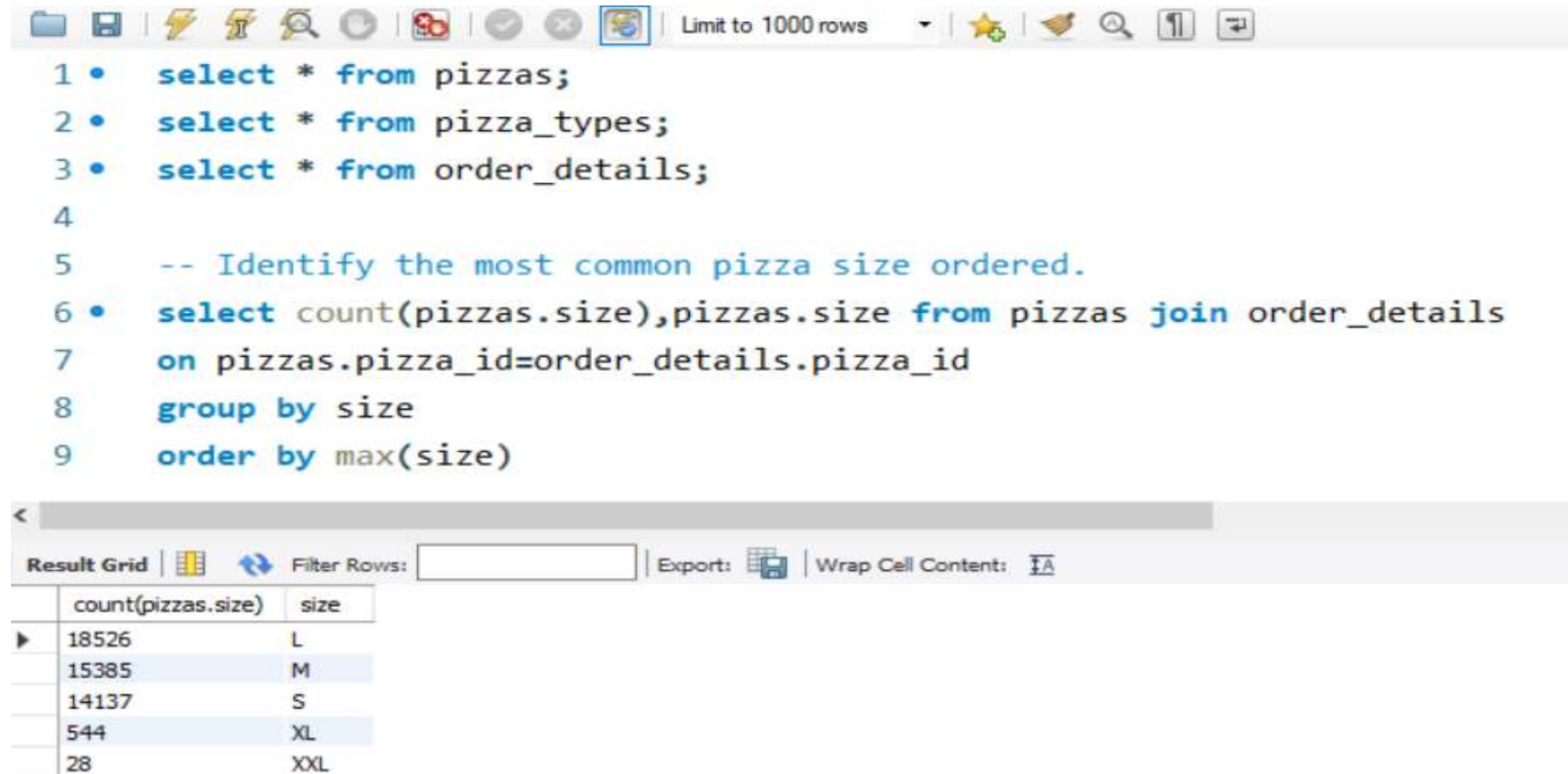
total_revenue
817860.05

#QUERY 3

```
1  -- Identify the highest-priced pizza.
2  • select name,price from pizza_types join pizzas
3    on pizza_types.pizza_type_id=pizzas.pizza_type_id
4    order by pizzas.price desc
5    limit 1
6  |
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	name	price				
▶	The Greek Pizza	35.95				

#QUERY 4



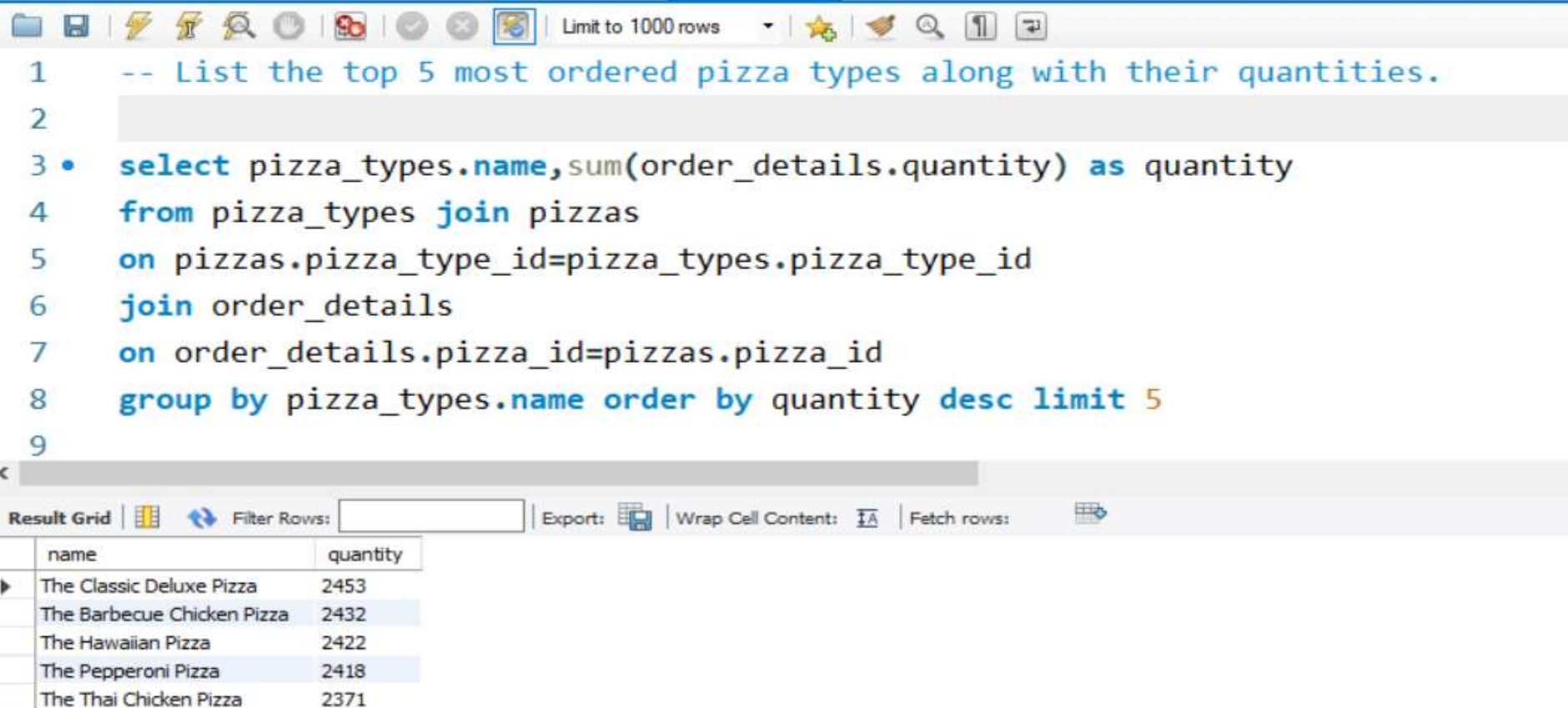
The screenshot shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query text is as follows:

```
1 • select * from pizzas;
2 • select * from pizza_types;
3 • select * from order_details;
4
5 -- Identify the most common pizza size ordered.
6 • select count(pizzas.size),pizzas.size from pizzas join order_details
7   on pizzas.pizza_id=order_details.pizza_id
8   group by size
9   order by max(size)
```

Below the query editor is the 'Result Grid' section, which includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'count(pizzas.size)' and 'size'.

count(pizzas.size)	size
18526	L
15385	M
14137	S
544	XL
28	XXL

#QUERY 5



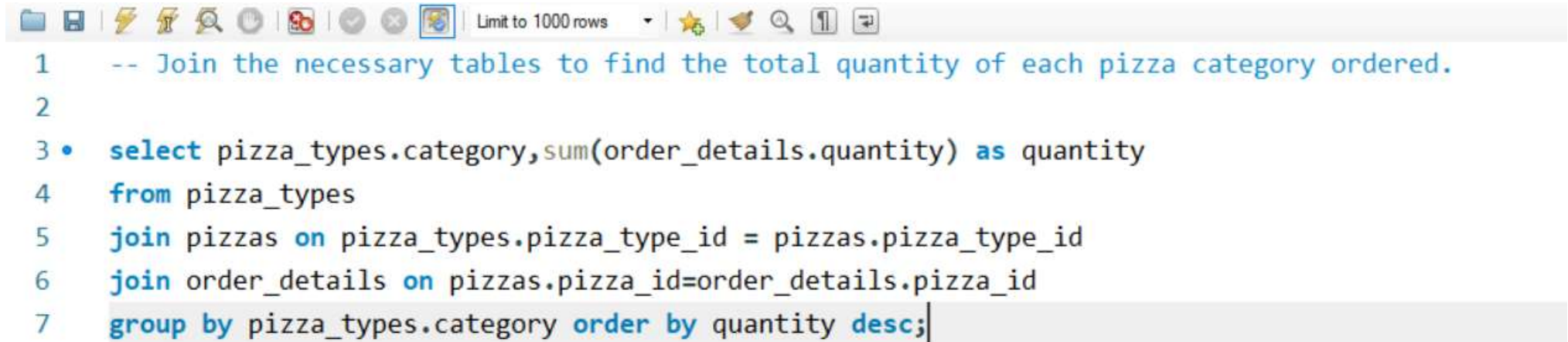
The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2
3  • select pizza_types.name, sum(order_details.quantity) as quantity
4  from pizza_types join pizzas
5  on pizzas.pizza_type_id=pizza_types.pizza_type_id
6  join order_details
7  on order_details.pizza_id=pizzas.pizza_id
8  group by pizza_types.name order by quantity desc limit 5
9
```

Below the query editor is the 'Result Grid' section, which displays the results of the query. It includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' toggle, and a 'Fetch rows' button. The results are shown in a table with two columns: 'name' and 'quantity'.

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

#QUERY 6



Limit to 1000 rows

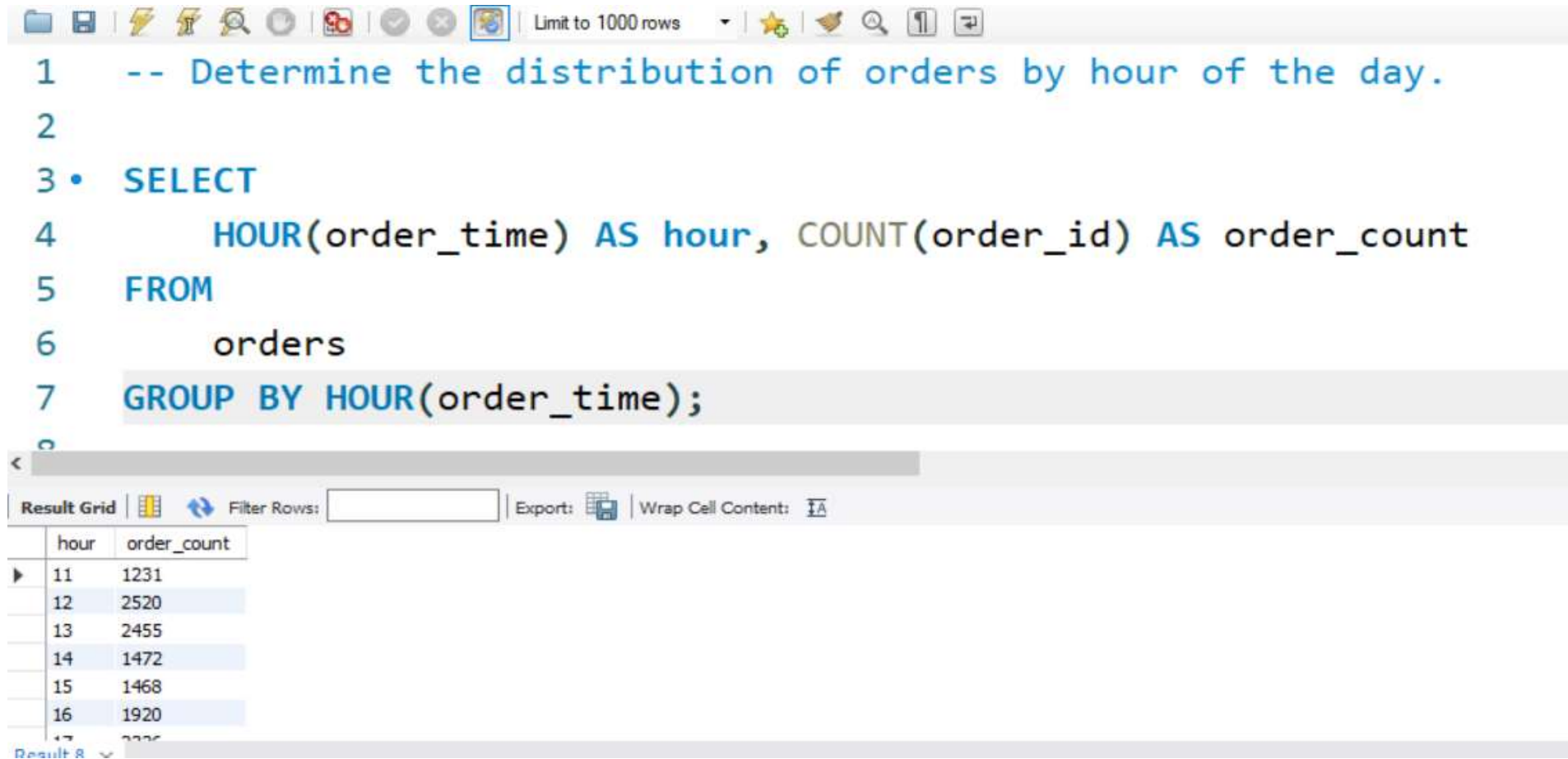
```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3  • select pizza_types.category, sum(order_details.quantity) as quantity
4  from pizza_types
5  join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
6  join order_details on pizzas.pizza_id=order_details.pizza_id
7  group by pizza_types.category order by quantity desc;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

#QUERY 7



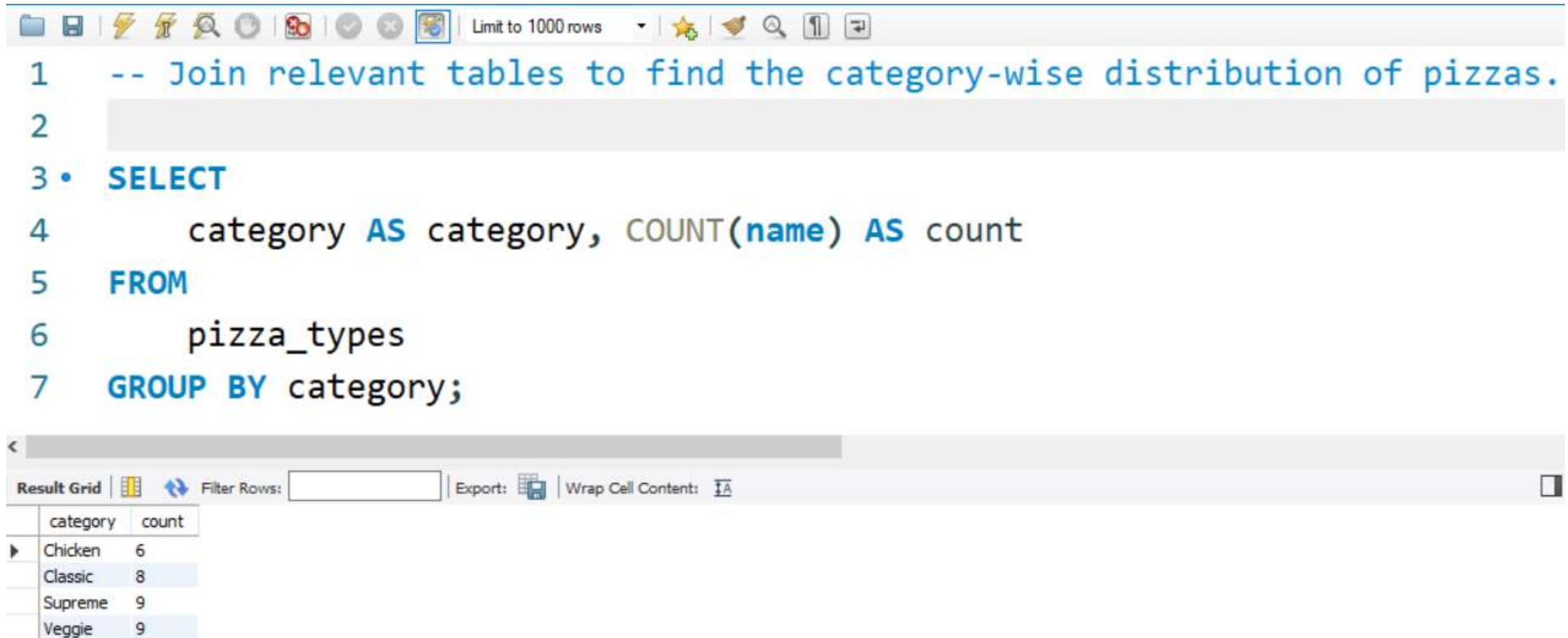
```
1  -- Determine the distribution of orders by hour of the day.
2
3 • SELECT
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5 FROM
6     orders
7 GROUP BY HOUR(order_time);
8
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2226

Result 8

#QUERY 8



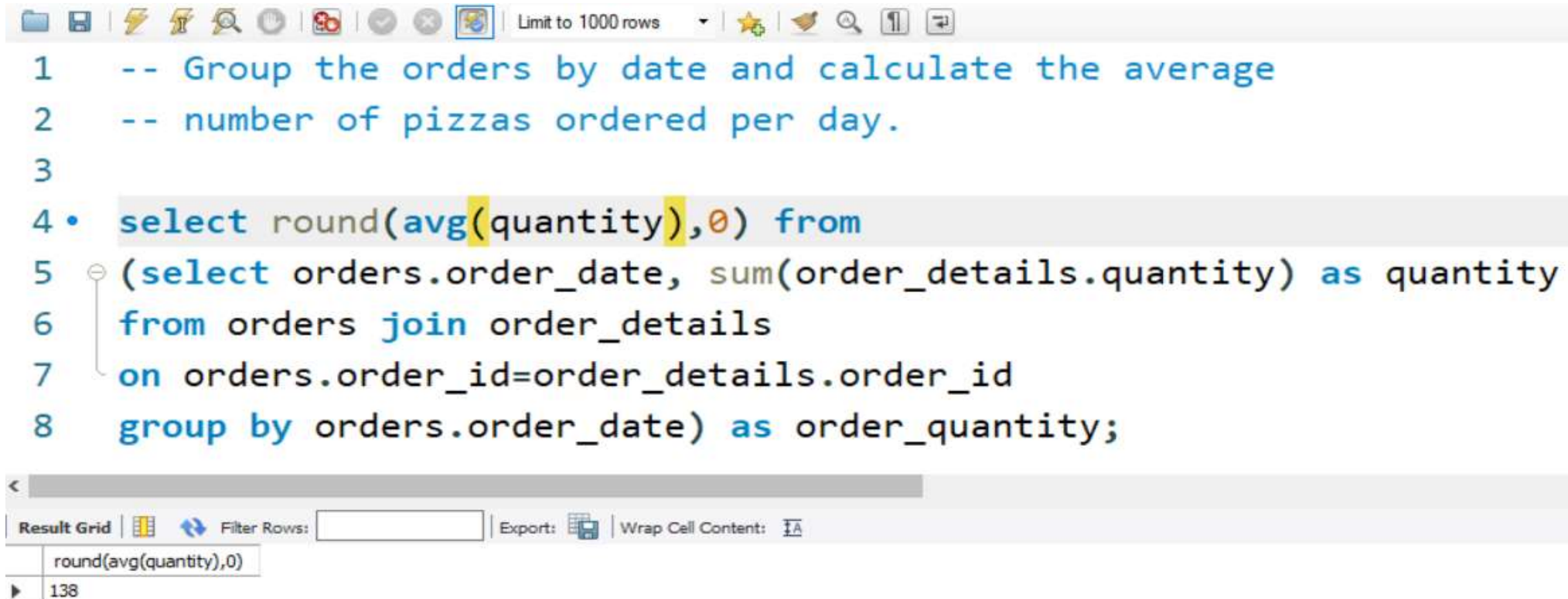
The screenshot shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and settings, along with a 'Limit to 1000 rows' dropdown. The query text is as follows:

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.
2
3 • SELECT
4     category AS category, COUNT(name) AS count
5 FROM
6     pizza_types
7 GROUP BY category;
```

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

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

#QUERY 9

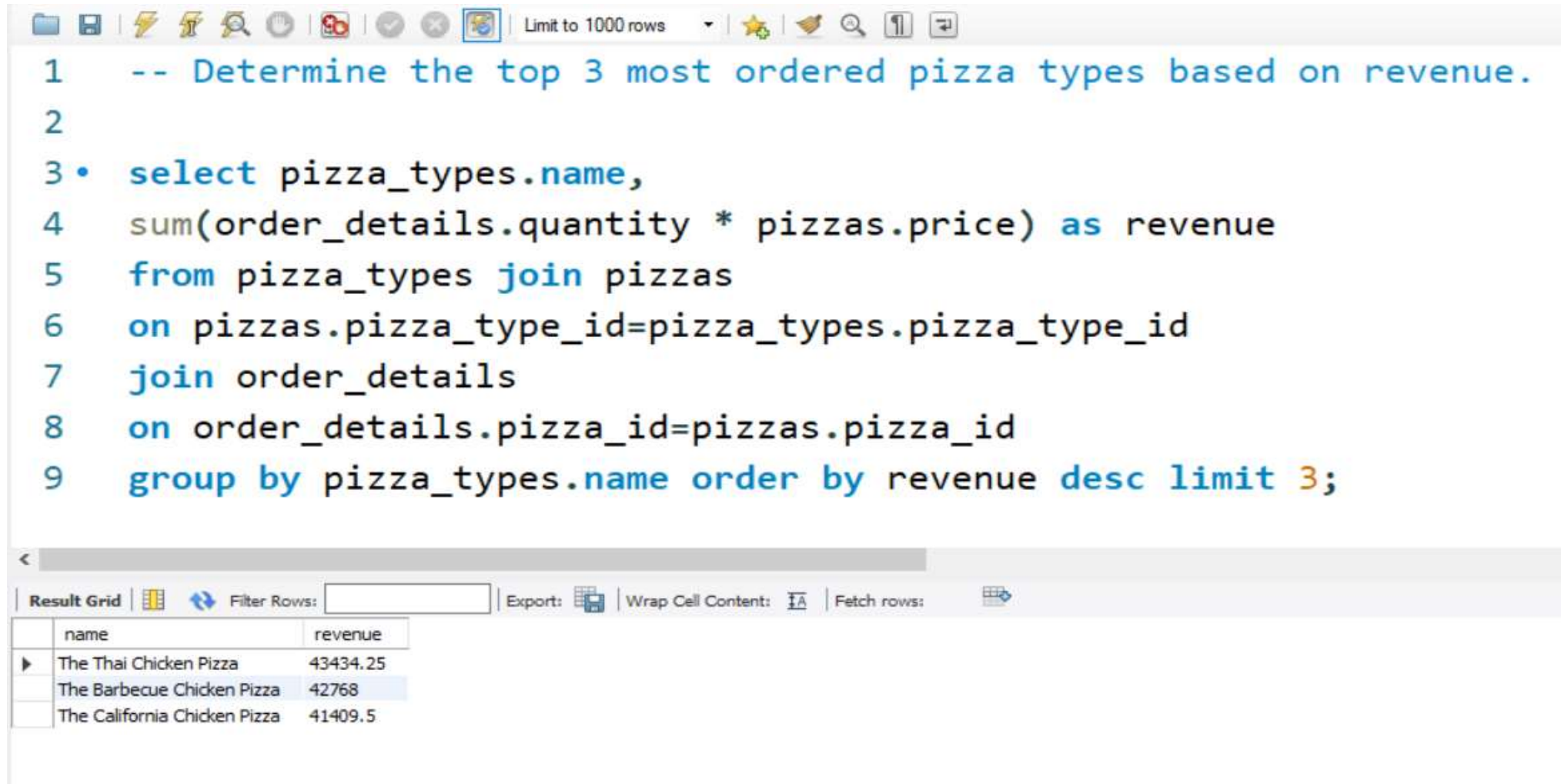


```
1  -- Group the orders by date and calculate the average
2  -- number of pizzas ordered per day.
3
4  • select round(avg(quantity),0) from
5    (select orders.order_date, sum(order_details.quantity) as quantity
6     from orders join order_details
7     on orders.order_id=order_details.order_id
8     group by orders.order_date) as order_quantity;
```

The screenshot shows a SQL query editor interface. The query is a subquery wrapped in a select statement. The subquery joins the 'orders' and 'order_details' tables on 'order_id' and groups the results by 'order_date'. The outer query calculates the average quantity of pizzas ordered per day, rounded to the nearest integer. The interface includes a toolbar at the top with various icons and a 'Limit to 1000 rows' dropdown. Below the query, there is a 'Result Grid' section with a 'Filter Rows' input, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid shows a single row with the value '138'.

round(avg(quantity),0)
138

#QUERY 10



```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3 • select pizza_types.name,
4    sum(order_details.quantity * pizzas.price) as revenue
5  from pizza_types join pizzas
6  on pizzas.pizza_type_id=pizza_types.pizza_type_id
7  join order_details
8  on order_details.pizza_id=pizzas.pizza_id
9  group by pizza_types.name order by revenue desc limit 3;
```

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

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5