



PIZZA SALES ANALYSIS USING SQL

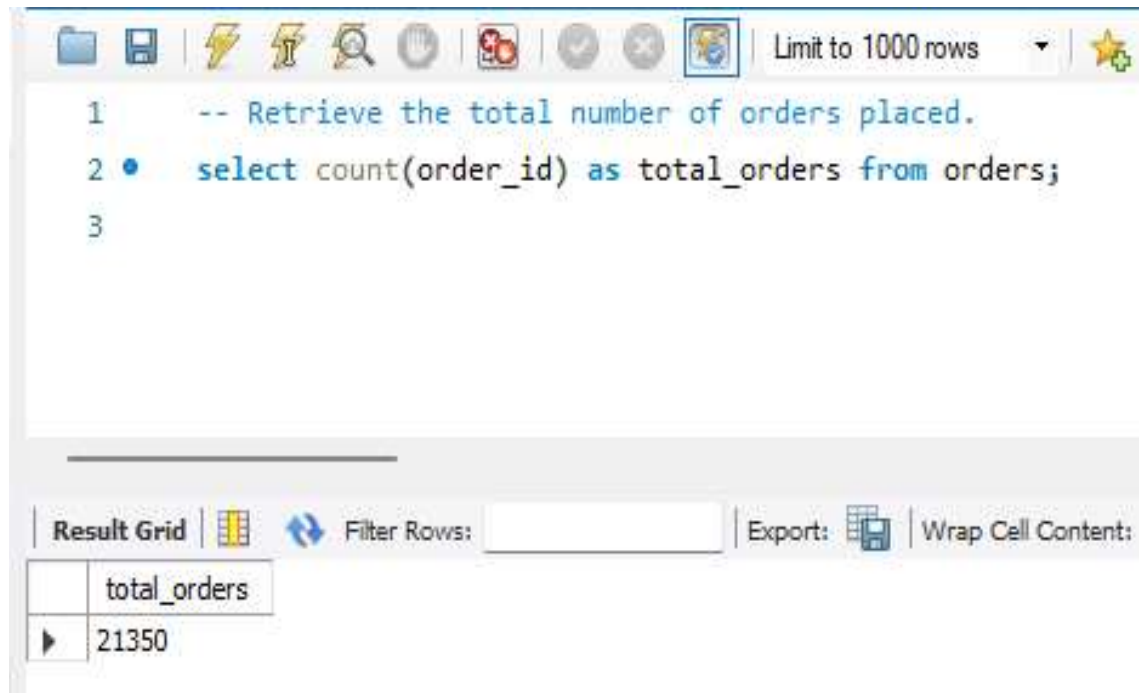
- By Rashi Chandel

Project Overview

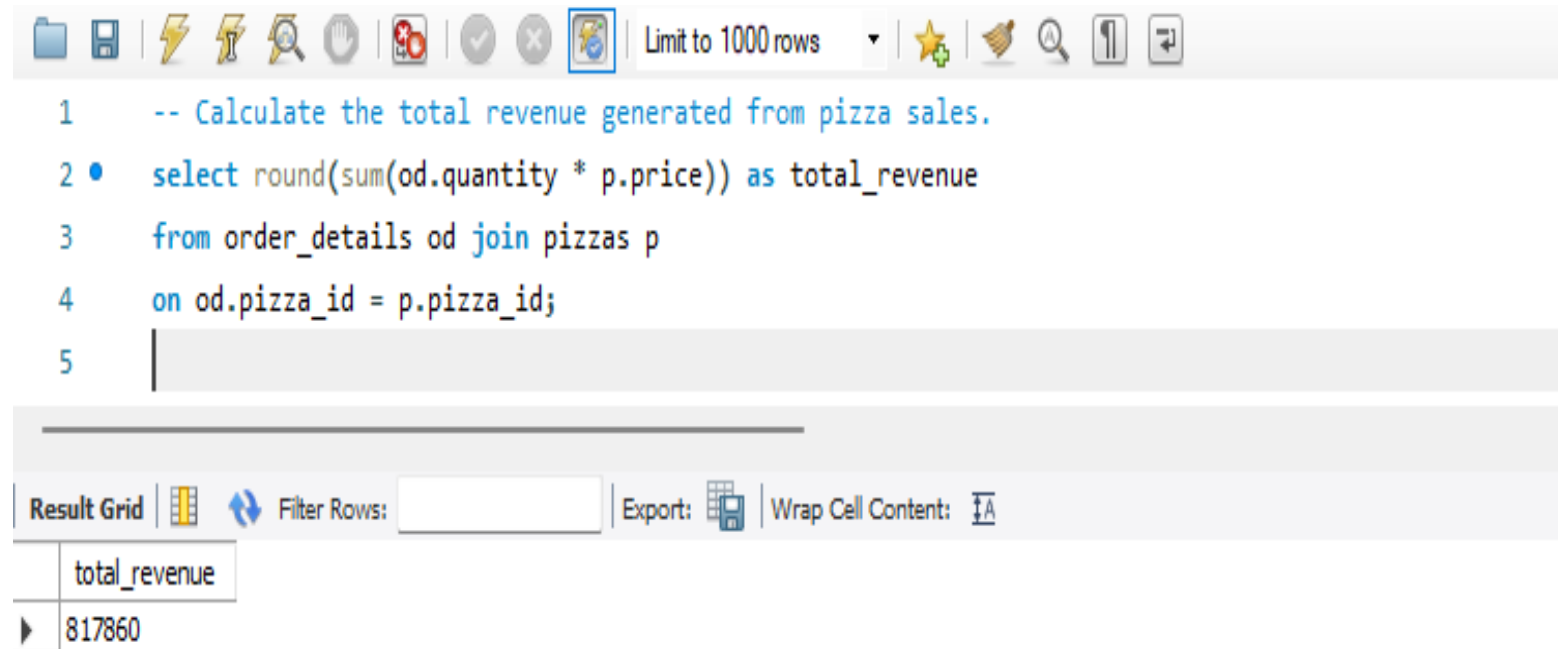
Objective: To analyze pizza sales data to understand sales patterns, revenue, customer preferences, and trends over time.

Basic:

Retrieve the total number of orders placed



Calculate the total revenue generated from pizza sales.



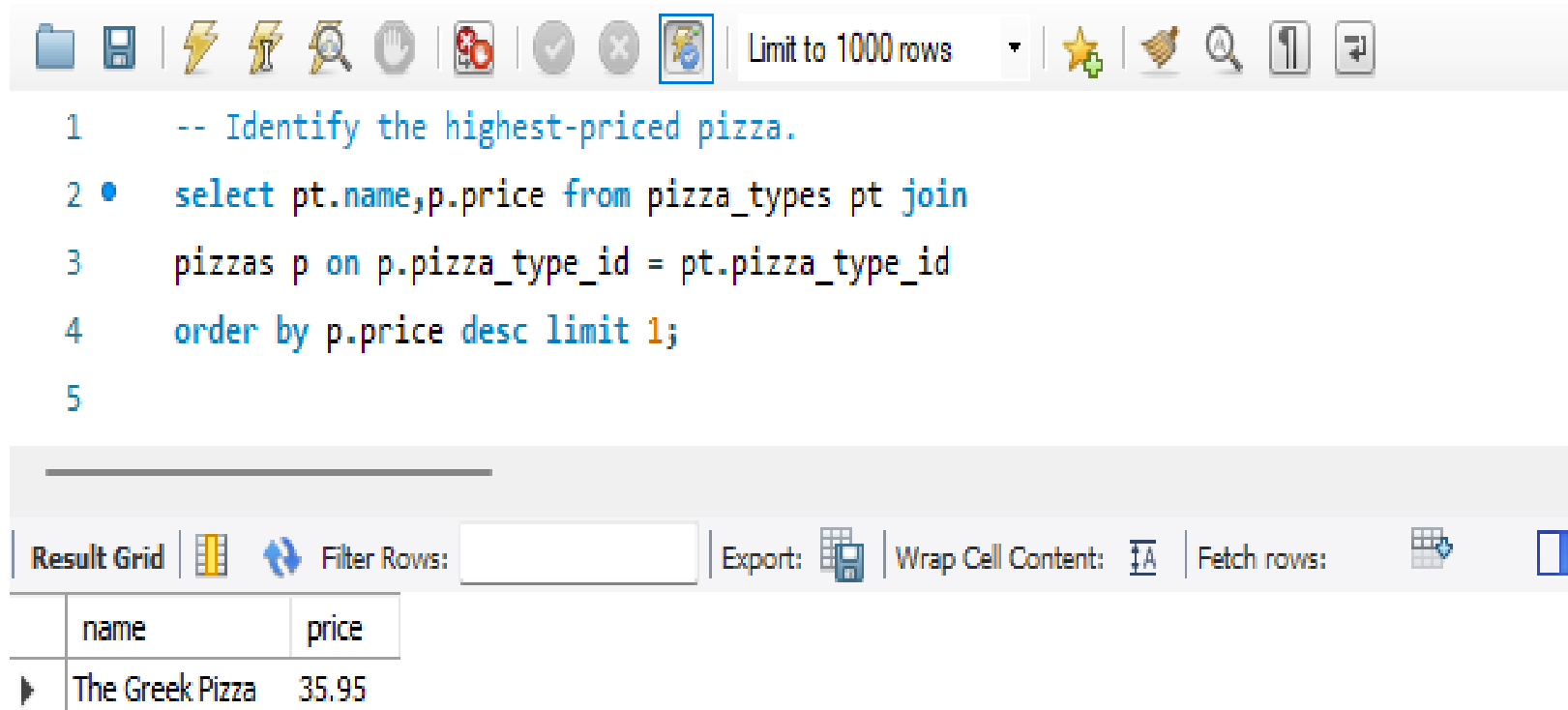
The screenshot shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a dropdown menu set to "Limit to 1000 rows". The query text is as follows:

```
1  -- Calculate the total revenue generated from pizza sales.  
2  • select round(sum(od.quantity * p.price)) as total_revenue  
3  from order_details od join pizzas p  
4  on od.pizza_id = p.pizza_id;  
5
```

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 result grid displays a single row with the column header "total_revenue" and the value "817860".

total_revenue
817860

Identify the highest-priced pizza.



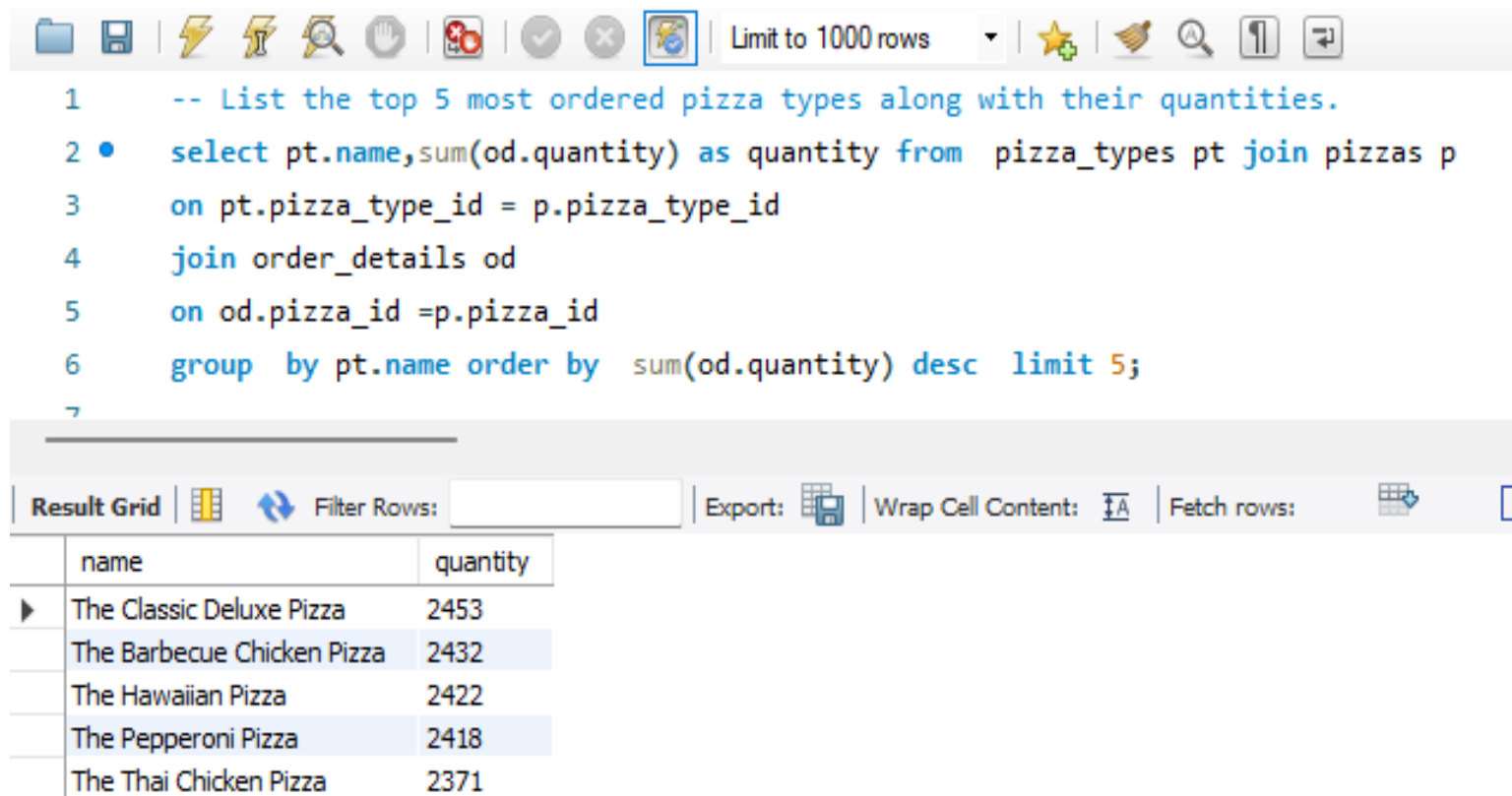
The screenshot shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a dropdown menu set to "Limit to 1000 rows". The query is as follows:

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

Below the query editor, the results are displayed in a table with columns "name" and "price". The first row shows "The Greek Pizza" with a price of 35.95.

	name	price
▶	The Greek Pizza	35.95

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



The screenshot shows a SQL query editor window with a toolbar at the top containing icons for file operations, execution, and search. The query text is as follows:

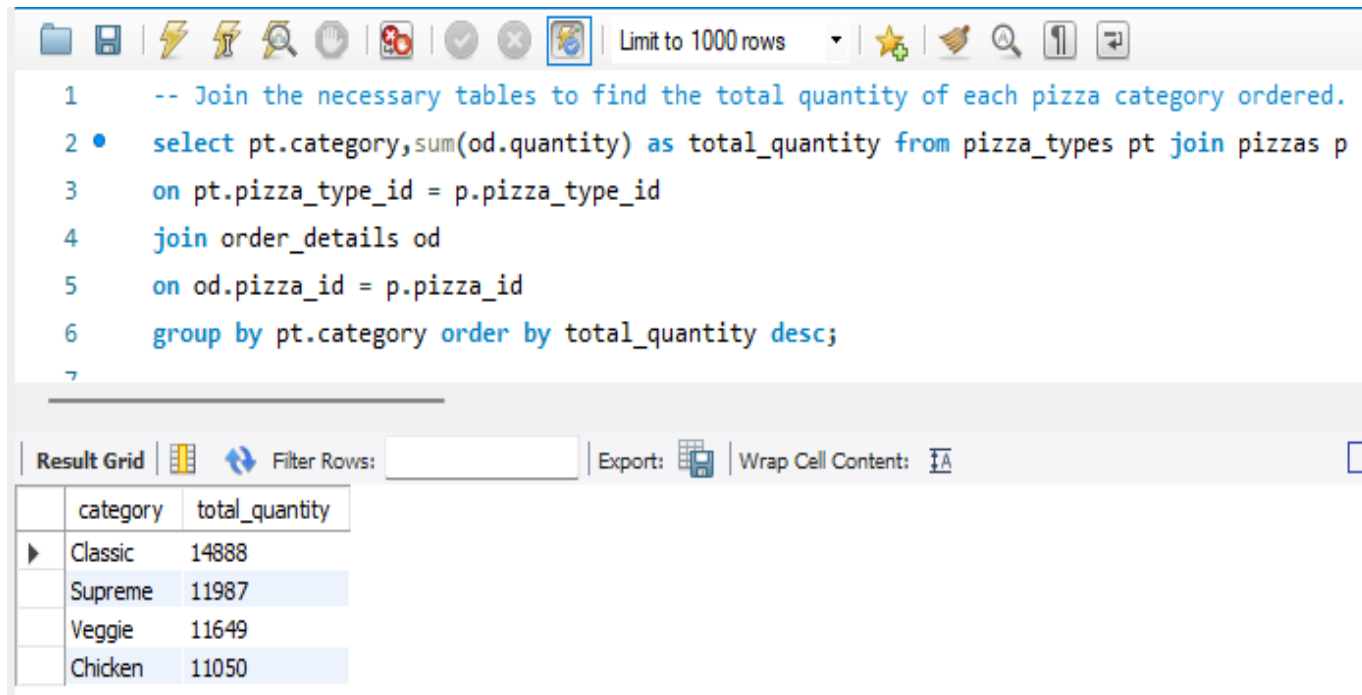
```
1  -- List the top 5 most ordered pizza types along with their quantities.
2  • select pt.name, sum(od.quantity) as quantity from pizza_types pt join pizzas p
3     on pt.pizza_type_id = p.pizza_type_id
4     join order_details od
5     on od.pizza_id = p.pizza_id
6     group by pt.name order by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

Below the query editor 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 results are displayed 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

Intermediate:

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



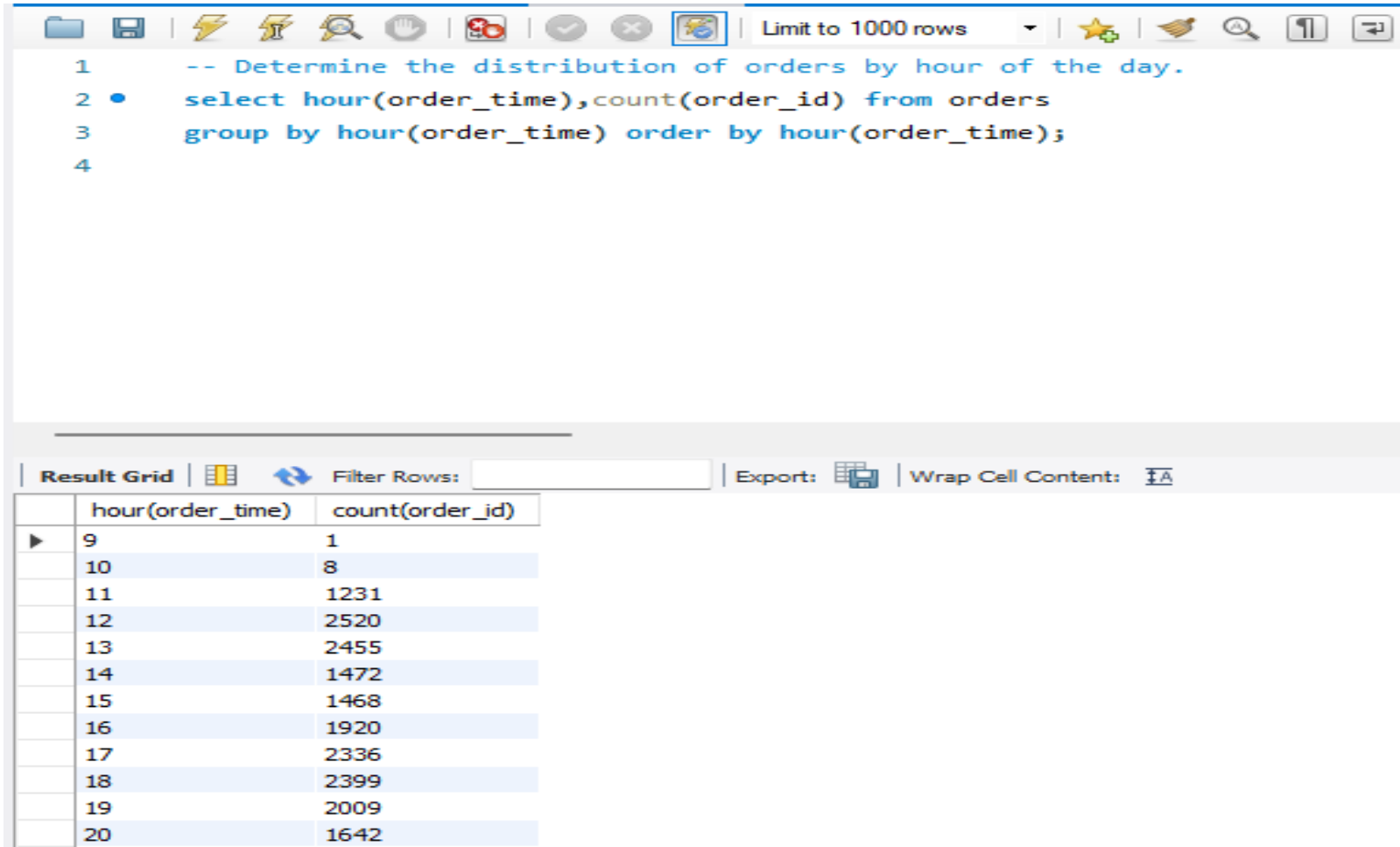
The screenshot shows a SQL query editor window. The query is as follows:

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

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 displays the following data:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.



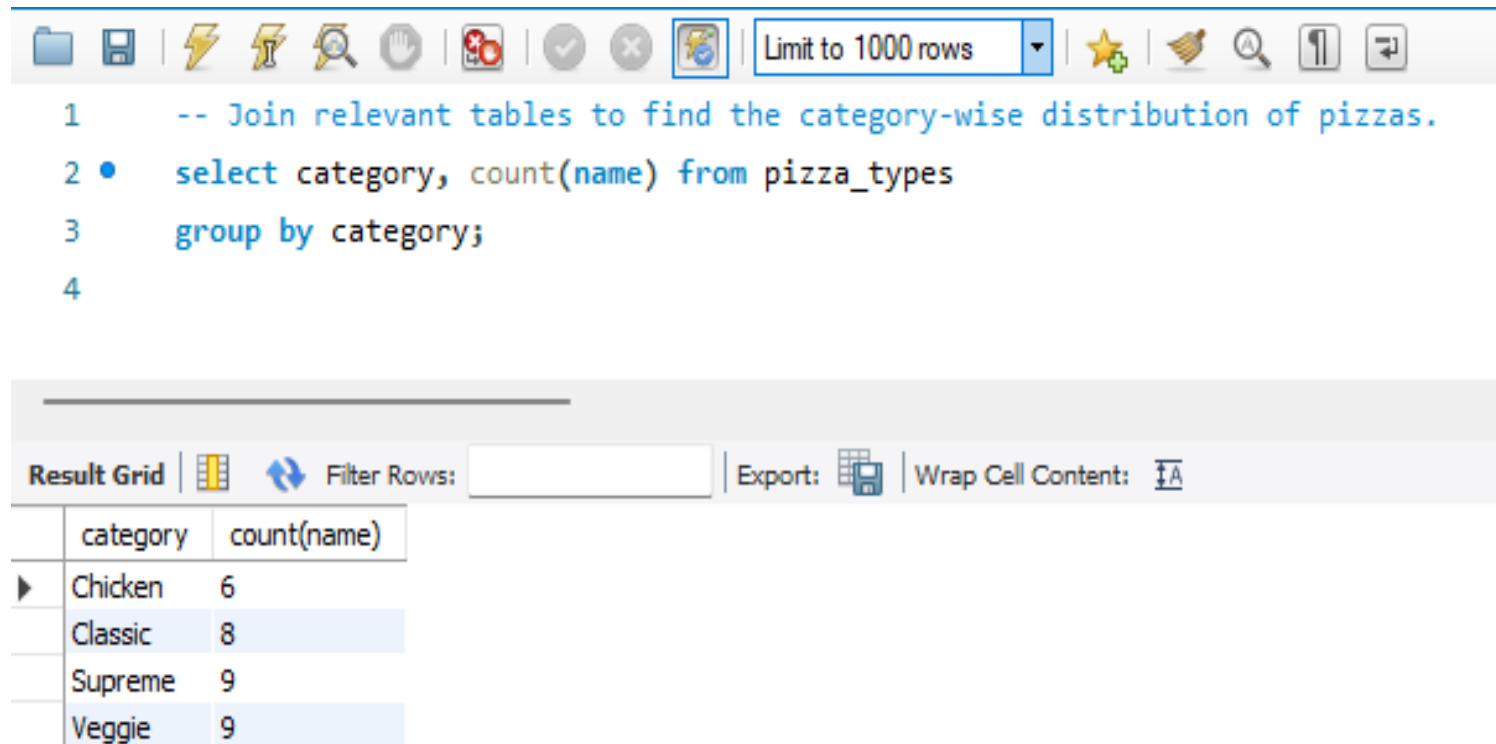
The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL query is entered in a text area. The query is as follows:

```
1  -- Determine the distribution of orders by hour of the day.  
2  • select hour(order_time), count(order_id) from orders  
3     group by hour(order_time) order by hour(order_time);  
4
```

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 displays the following data:

	hour(order_time)	count(order_id)
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

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



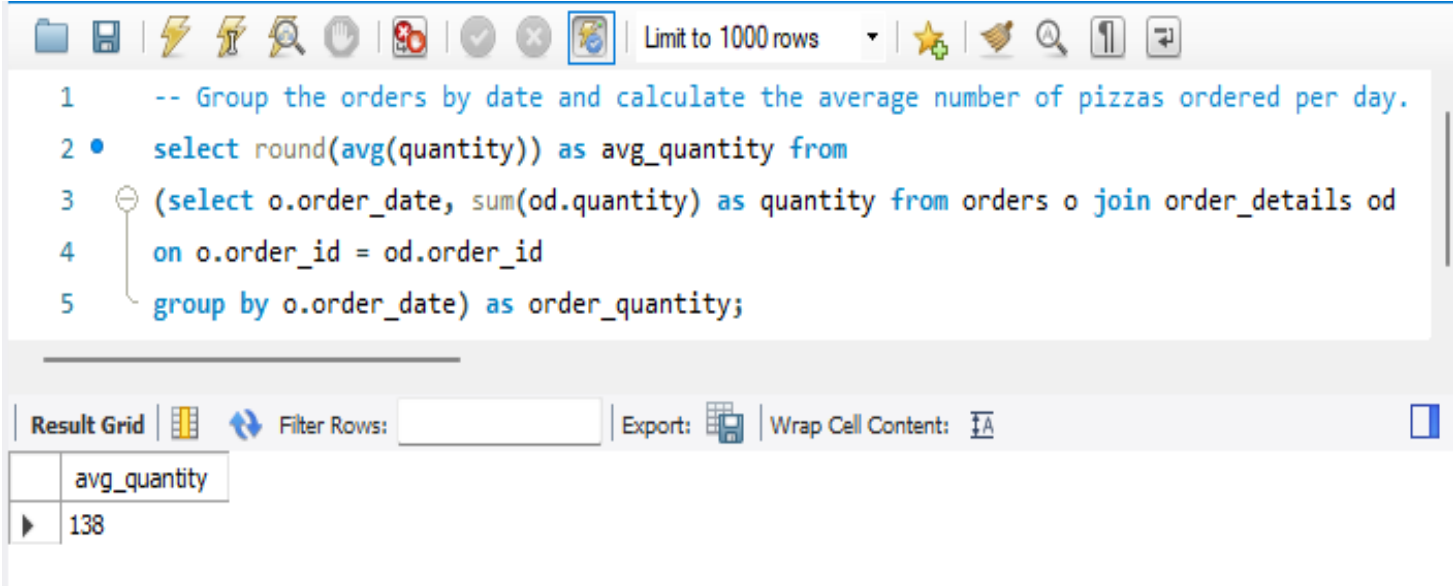
The screenshot shows a SQL query editor interface. The top toolbar includes icons for file operations, execution, and a dropdown menu set to "Limit to 1000 rows". The query text is as follows:

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

Below the query editor, the "Result Grid" tab is active, displaying the results of the query in a table. The table has two columns: "category" and "count(name)". The results are:

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

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



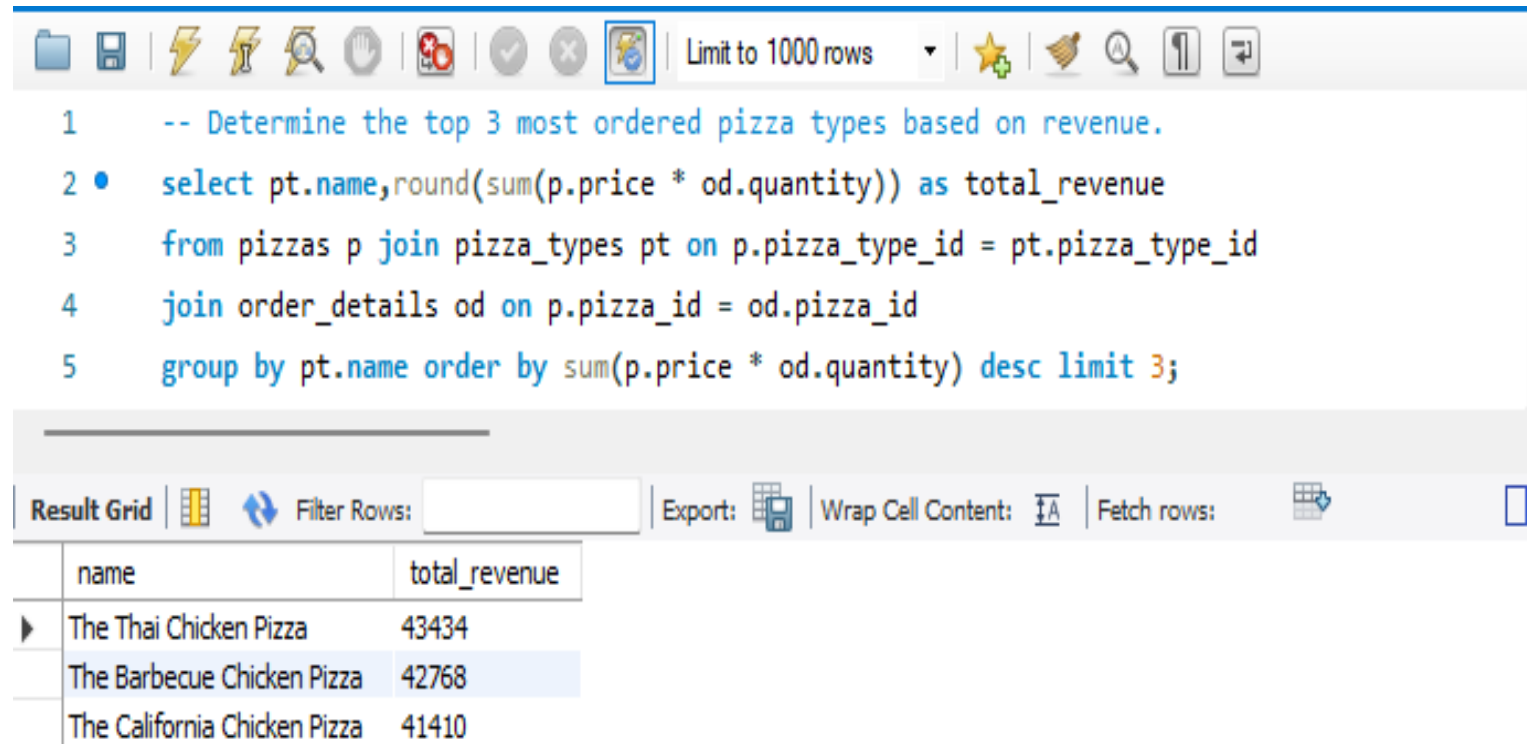
The screenshot shows a SQL query editor window. The query is as follows:

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

Below the query editor, there is a toolbar with icons for saving, running, and other functions. The text "Limit to 1000 rows" is visible. Below the toolbar, there is a section for "Result Grid" with a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox. The result grid shows a single row with the column "avg_quantity" and the value "138".

avg_quantity
138

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



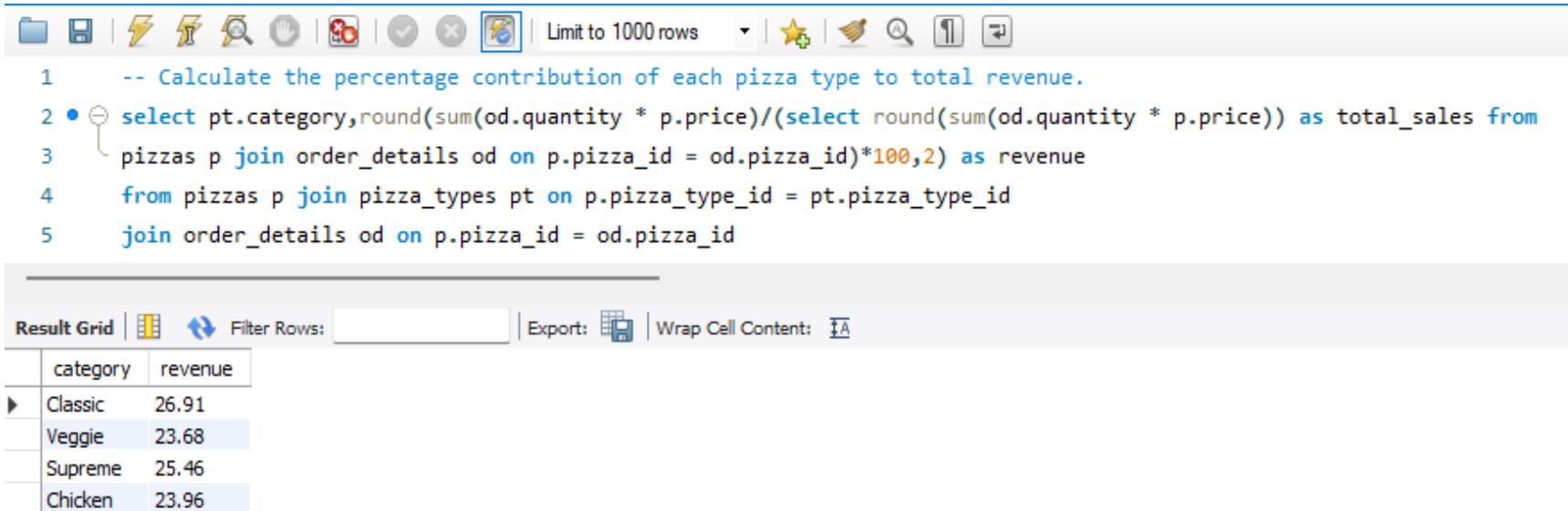
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search, along with a dropdown menu set to "Limit to 1000 rows". The SQL editor contains a query to find the top 3 pizza types by revenue. Below the editor, the "Result Grid" tab is active, displaying a table with the query results. The table has two columns: "name" and "total_revenue". The results are sorted in descending order of revenue, showing "The Thai Chicken Pizza" as the top performer, followed by "The Barbecue Chicken Pizza" and "The California Chicken Pizza".

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

name	total_revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Advanced:

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



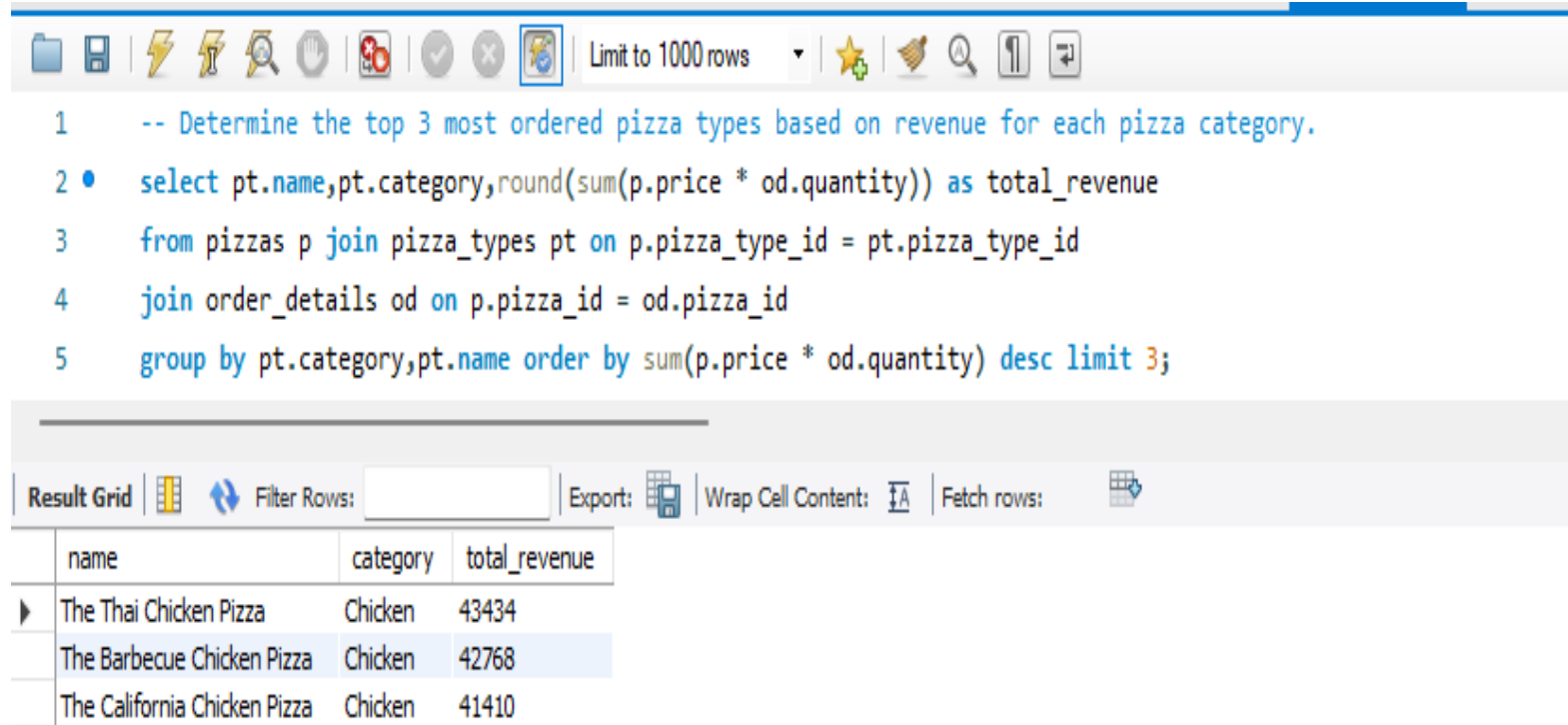
The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2  • select pt.category, round(sum(od.quantity * p.price) / (select round(sum(od.quantity * p.price)) as total_sales from
3  pizzas p join order_details od on p.pizza_id = od.pizza_id) * 100, 2) as revenue
4  from pizzas p join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
5  join order_details od on p.pizza_id = od.pizza_id
```

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 result grid displays the following data:

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Determine the top 3 most ordered pizza types based on revenue for each pizza category.



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

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
2  • select pt.name,pt.category,round(sum(p.price * od.quantity)) as total_revenue  
3  from pizzas p join pizza_types pt on p.pizza_type_id = pt.pizza_type_id  
4  join order_details od on p.pizza_id = od.pizza_id  
5  group by pt.category,pt.name order by sum(p.price * od.quantity) desc limit 3;
```

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

	name	category	total_revenue
▶	The Thai Chicken Pizza	Chicken	43434
	The Barbecue Chicken Pizza	Chicken	42768
	The California Chicken Pizza	Chicken	41410