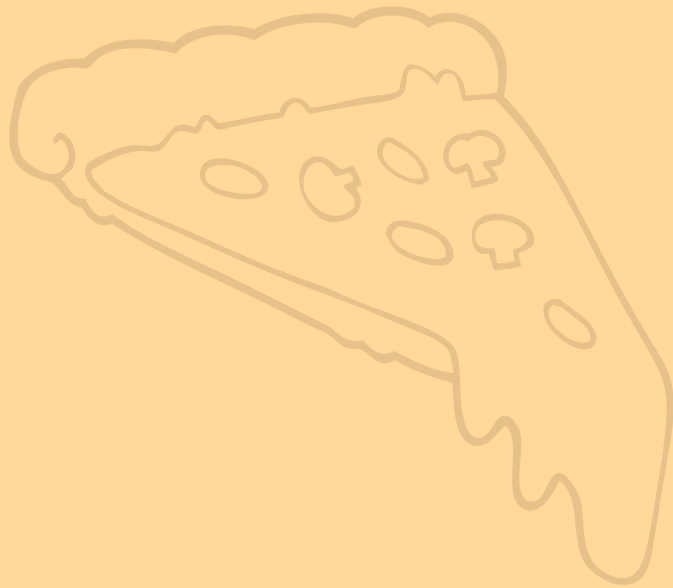
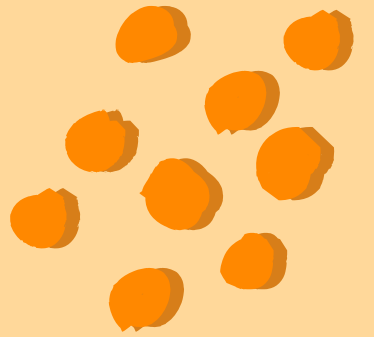


# SQL PROJECT ON PIZZA

Delicious Pizza for Everyone!



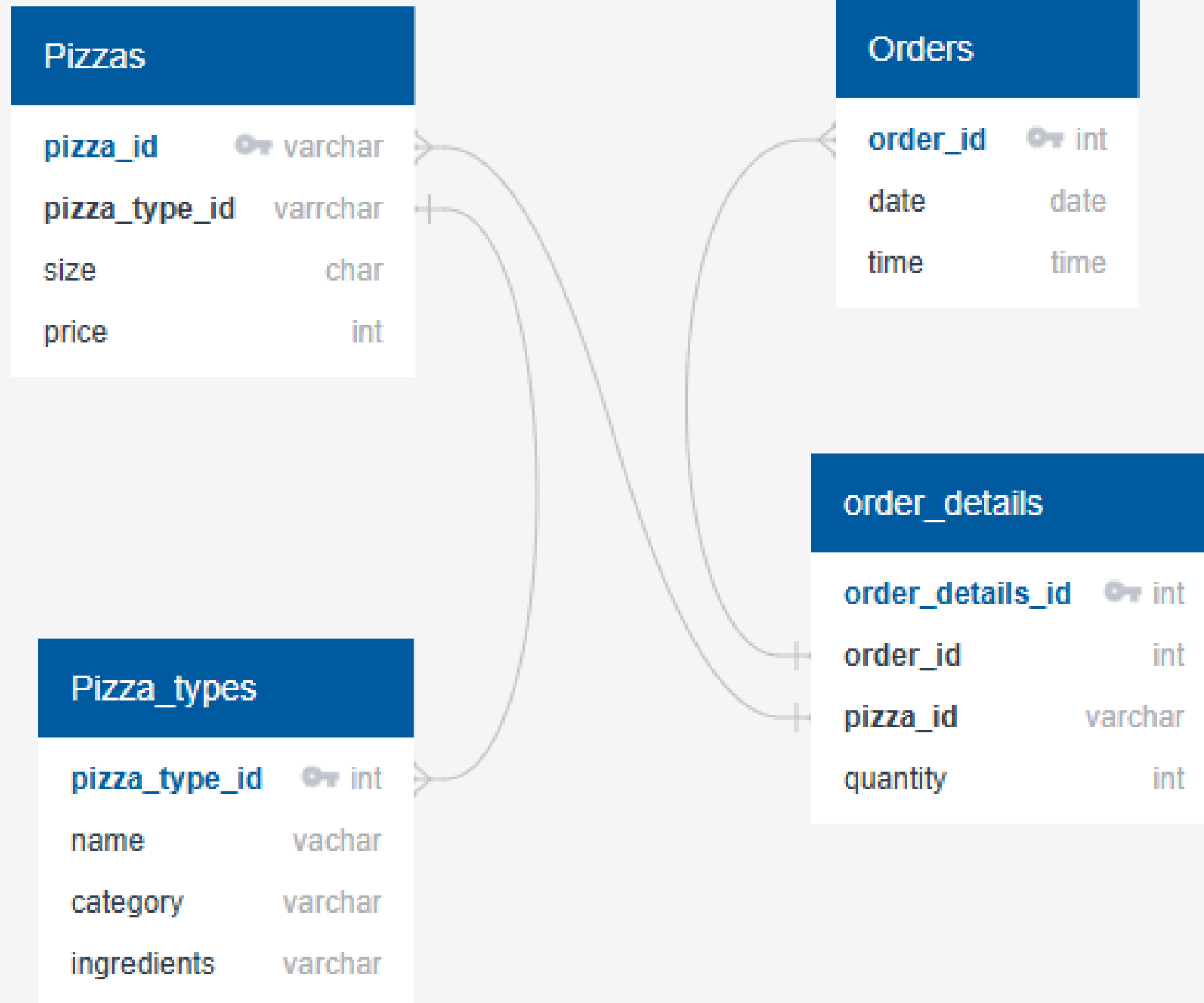
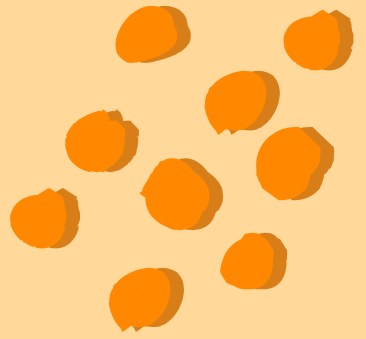


# Hello!

My name is Zain and I have utilized SQL to solve questions related to Pizza Sales.



# Schema of database

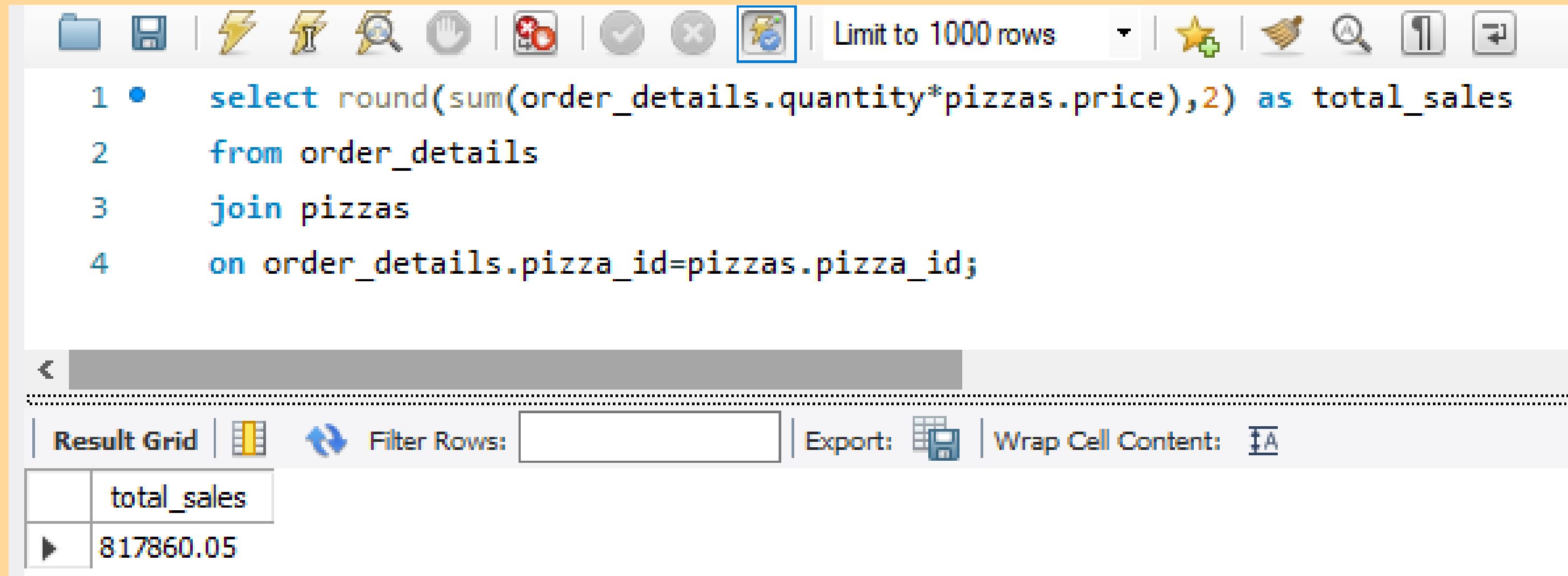


# Retrieve the total number of orders placed

```
select count(order_id) as total_oders  
from orders;
```

Result Grid	
	total_oders
▶	21350

# Calculate the total revenue generated from pizza sales



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

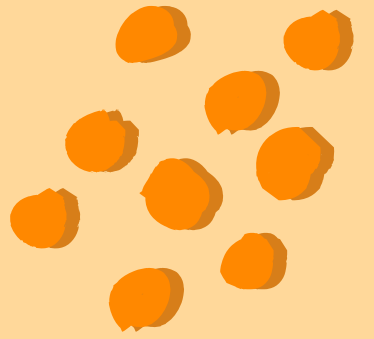
```
1 • select round(sum(order_details.quantity*pizzas.price),2) as total_sales
2   from order_details
3   join pizzas
4   on order_details.pizza_id=pizzas.pizza_id;
```

Below the query editor, there is a toolbar with various icons for file operations, execution, and settings. The 'Limit to 1000 rows' option is visible. Below the toolbar, there is a section for 'Result Grid' with a table showing the result of the query.

	total_sales
▶	817860.05



# Identify the highest-priced pizza



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

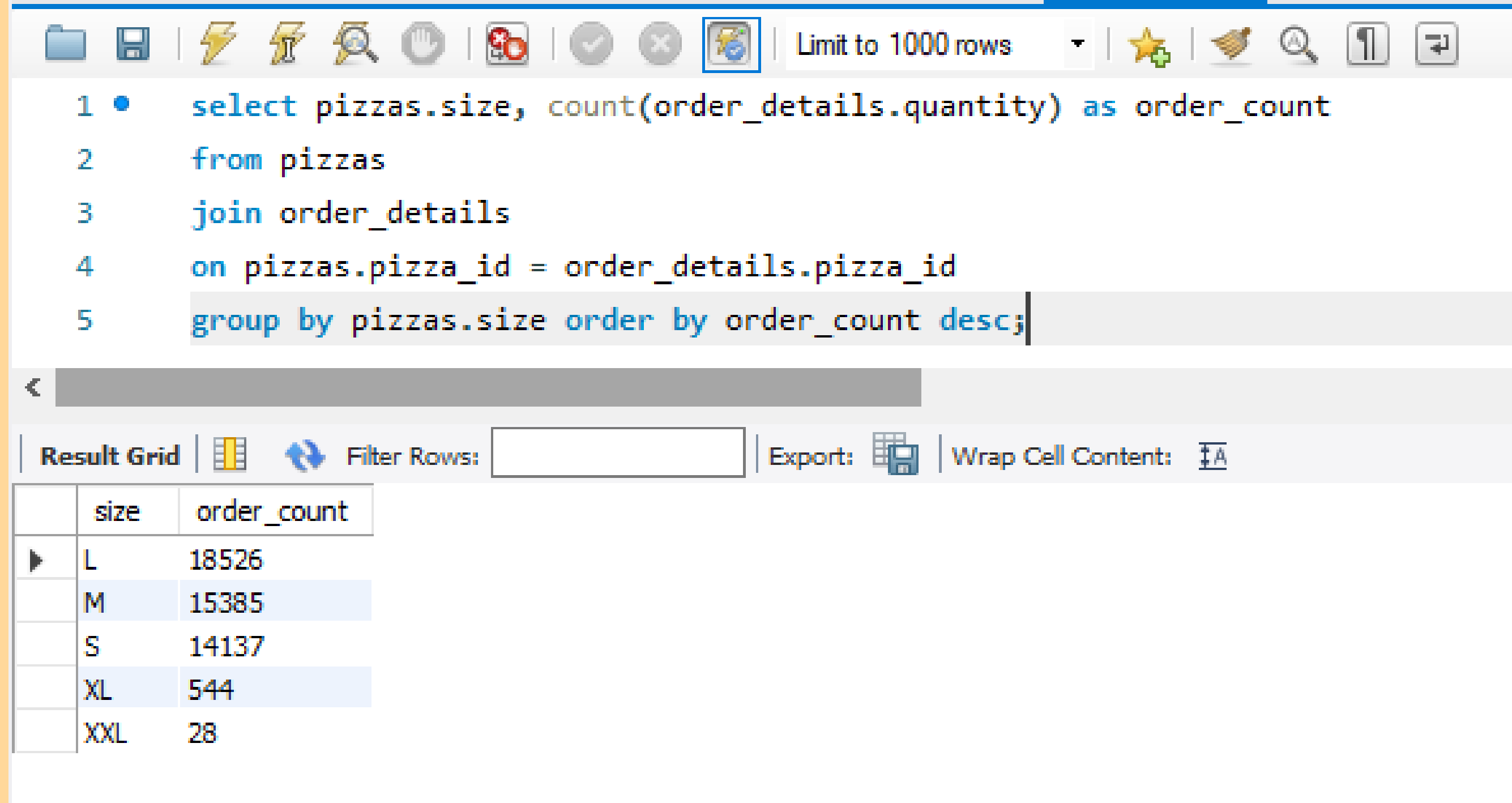
```
2  from pizza_types
3  join pizzas
4  on pizza_types.pizza_type_id= pizzas.pizza_type_id
5  order by pizzas.price desc
6  limit 1;
```

Below the query editor is a control bar with options for "Result Grid", "Filter Rows" (with an input field), "Export" (with a download icon), "Wrap Cell Content" (with a text icon), and "Fetch rows" (with a refresh icon). Below this bar is a table displaying the query results:

	name	price
▶	The Greek Pizza	35.95



# Identify the most common pizza size ordered



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

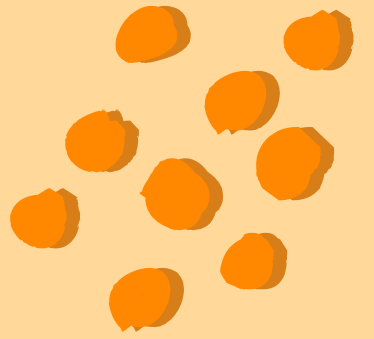
```
1 • select pizzas.size, count(order_details.quantity) as order_count
2   from pizzas
3  join order_details
4  on pizzas.pizza_id = order_details.pizza_id
5  group by pizzas.size order by order_count desc;
```

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:

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



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



```
3  join pizzas
4  on pizza_types.pizza_type_id=pizzas.pizza_type_id
5  join order_details
6  on order_details.pizza_id = pizzas.pizza_id
7  group by pizza_types.name
```

<

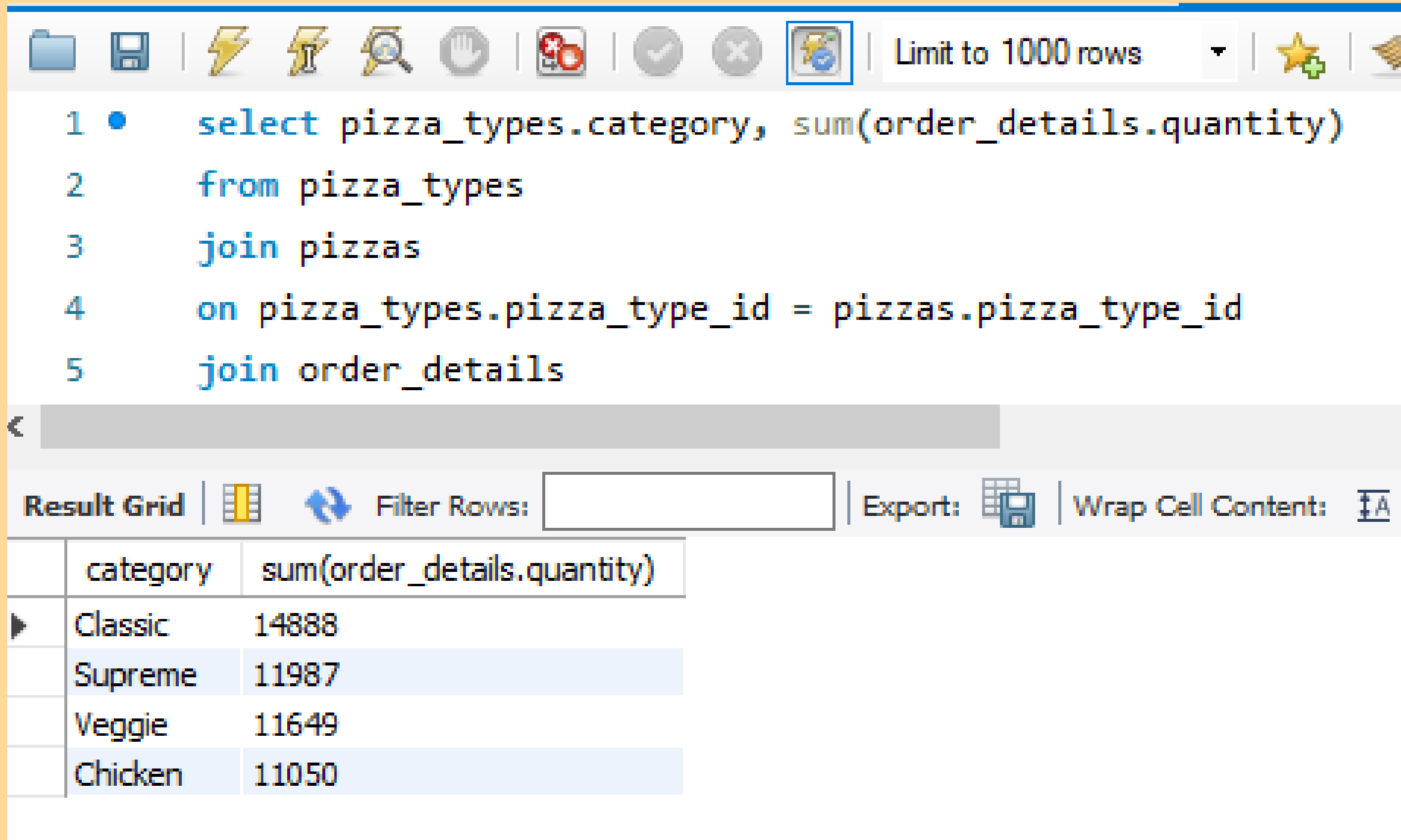
Result Grid | Filter Rows:  | Export: | Wrap Cell Content: | F

	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





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



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

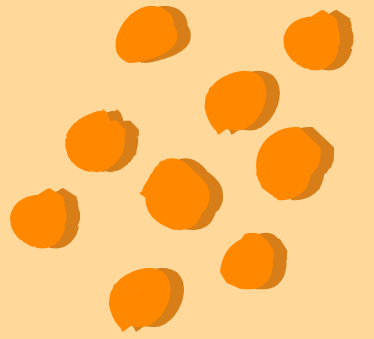
```
1 • select pizza_types.category, sum(order_details.quantity)
2   from pizza_types
3  join pizzas
4 on pizza_types.pizza_type_id = pizzas.pizza_type_id
5  join order_details
```

Below the query editor is a "Result Grid" section with a "Filter Rows" input field and "Export" and "Wrap Cell Content" options. The result grid displays the following data:

	category	sum(order_details.quantity)
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# Determine the distribution of orders by hour of the day



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

```
1 • select hour(orders.time) as hour, count(order_id) as order_count
2   from orders
3   group by hour(orders.time);
```

Below the query editor is a result grid. The grid has two columns: 'hour' and 'order\_count'. The data is as follows:

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

The interface includes a toolbar at the top with various icons for file operations, execution, and viewing. A dropdown menu shows 'Limit to 1000 rows'. Below the query editor, there are options for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.



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

```
1 • select pizza_types.category, count(name)
2   from pizza_types
3  group by category;
```

Result Grid



Filter Rows:

Export:

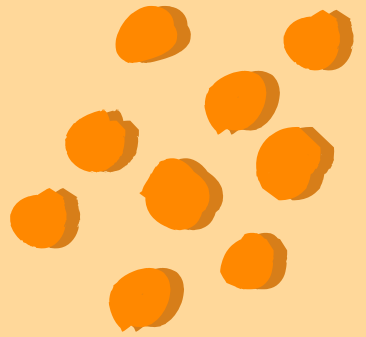


Wrap Cell Content:

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

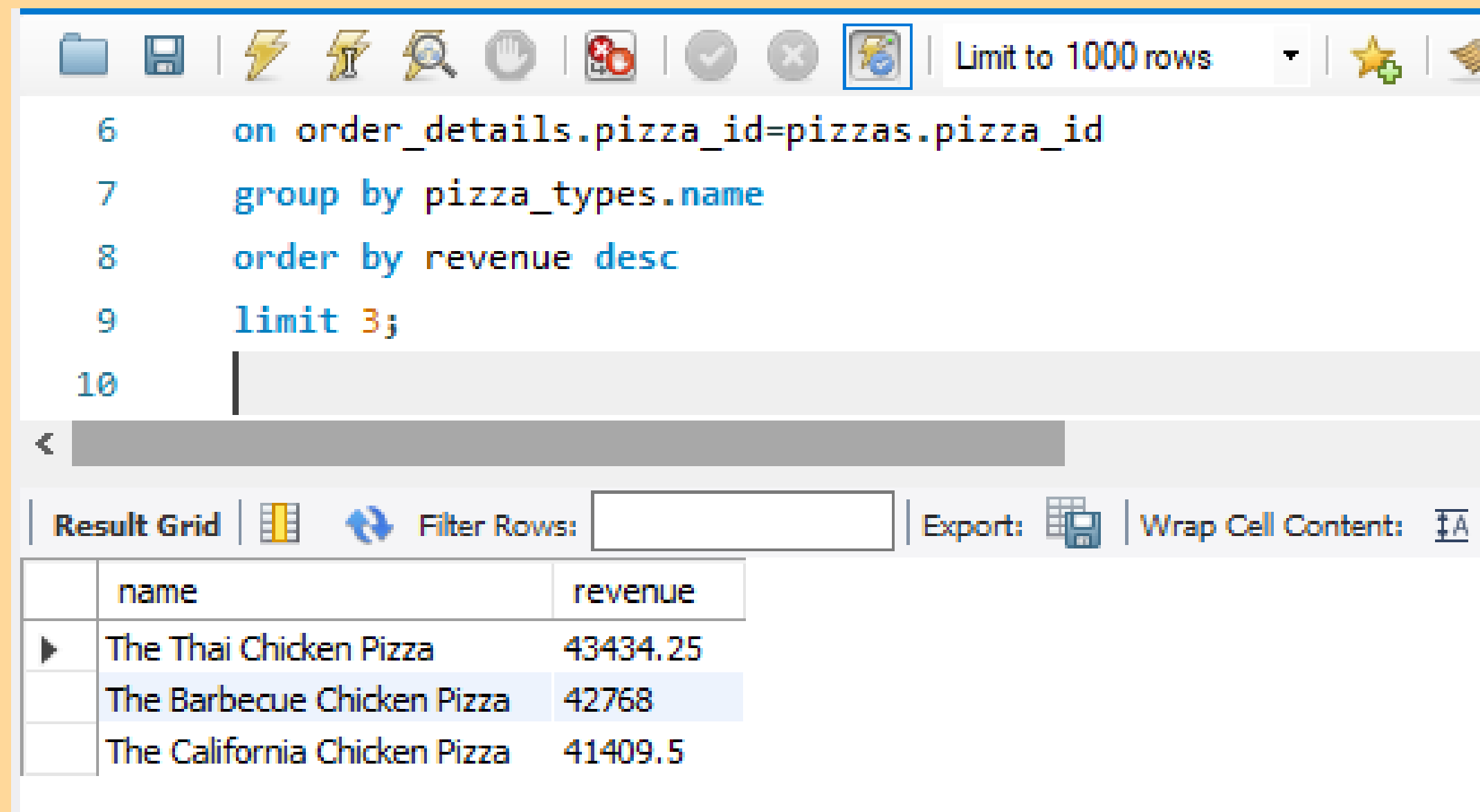
```
3  (select orders.date, sum(order_details.quantity) as quantity
4  from orders
5  join order_details
6  on orders.order_id=order_details.order_id
7  group by orders.date) as order_quantity;
```

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:

	Avg_pizaa_per_day
▶	138



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



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

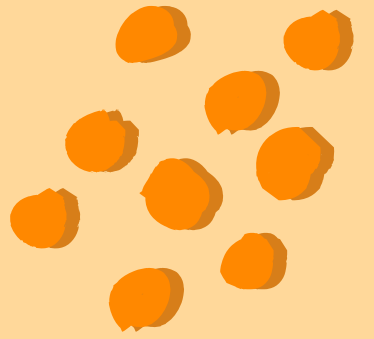
```
6 on order_details.pizza_id=pizzas.pizza_id
7 group by pizza_types.name
8 order by revenue desc
9 limit 3;
10
```

Below the query editor is a "Result Grid" section with a "Filter Rows" input field and an "Export" button. The results are displayed in a table with two columns: "name" and "revenue".

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



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



SQL Query Editor Interface

```
1 • select pizza_types.category,  
2   round((sum(order_details.quantity*pizzas.price) / (select round(sum(order_details.quantity*pizzas.price),2)  
3   as total_sales  
4   from order_details  
5   join pizzas
```

Result Grid




	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



# Analyze the cumulative revenue generated over time

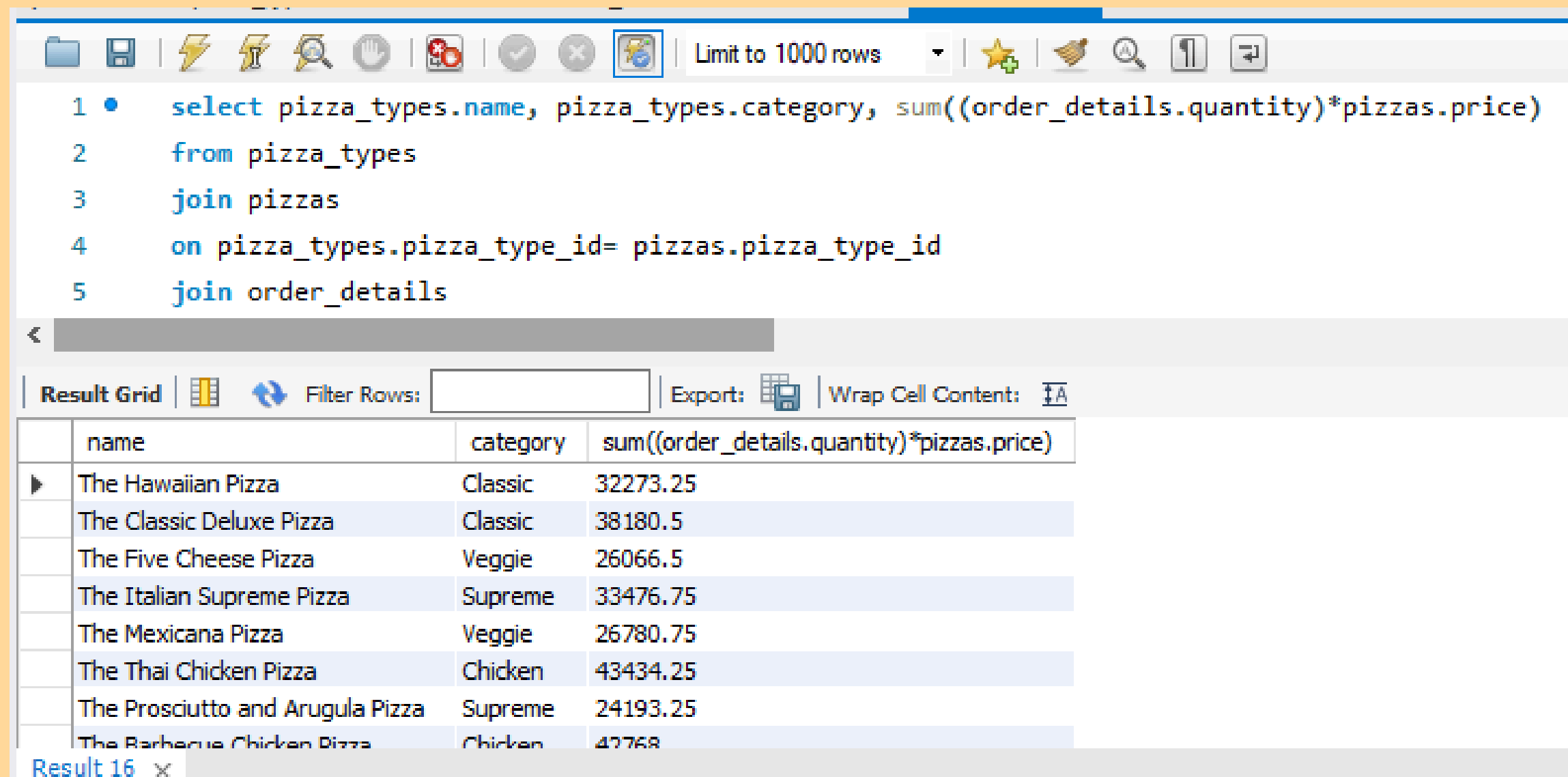
```
1 • select date,  
2   sum(revenue) over (order by date) as cum_revenue  
3   from  
4   (select orders.date,  
5    round(sum(order_details.quantity * pizzas.price),2) as revenue
```

<

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

	date	cum_revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7

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



```
1 • select pizza_types.name, pizza_types.category, sum((order_details.quantity)*pizzas.price)
2   from pizza_types
3  join pizzas
4 on pizza_types.pizza_type_id= pizzas.pizza_type_id
5  join order_details
```

Result Grid

	name	category	sum((order_details.quantity)*pizzas.price)
▶	The Hawaiian Pizza	Classic	32273.25
	The Classic Deluxe Pizza	Classic	38180.5
	The Five Cheese Pizza	Veggie	26066.5
	The Italian Supreme Pizza	Supreme	33476.75
	The Mexicana Pizza	Veggie	26780.75
	The Thai Chicken Pizza	Chicken	43434.25
	The Prosciutto and Arugula Pizza	Supreme	24193.25
	The Barbecue Chicken Pizza	Chicken	42768

Result 16 x