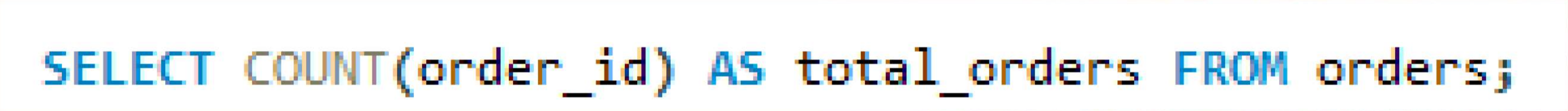
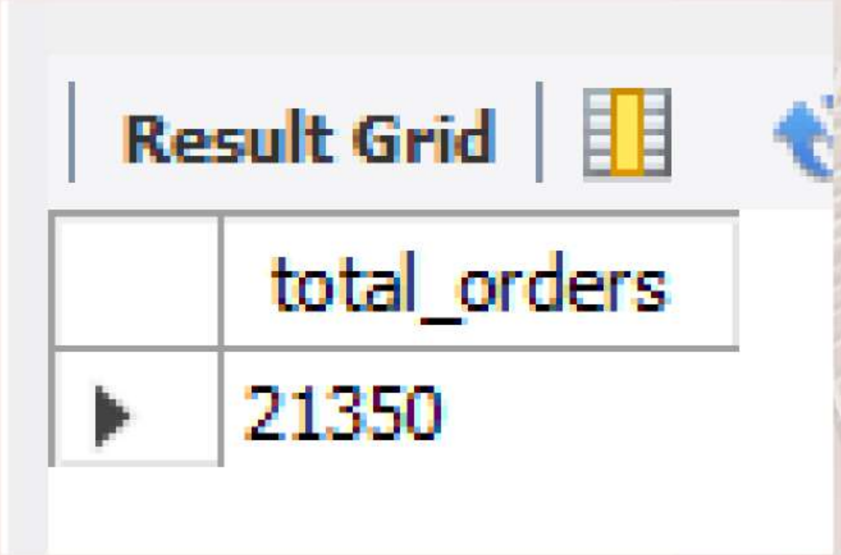
RAHUL KUMAR

PIZZA HUT BY USING SQL

Retrieve the total number of orders placed.

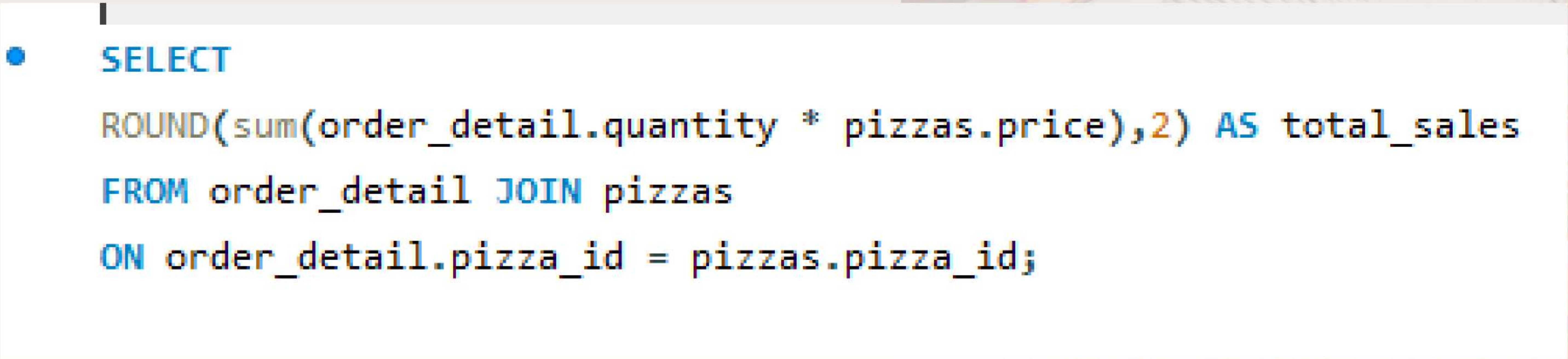


```sql
SELECT COUNT(order_id) AS total_orders FROM orders;
```

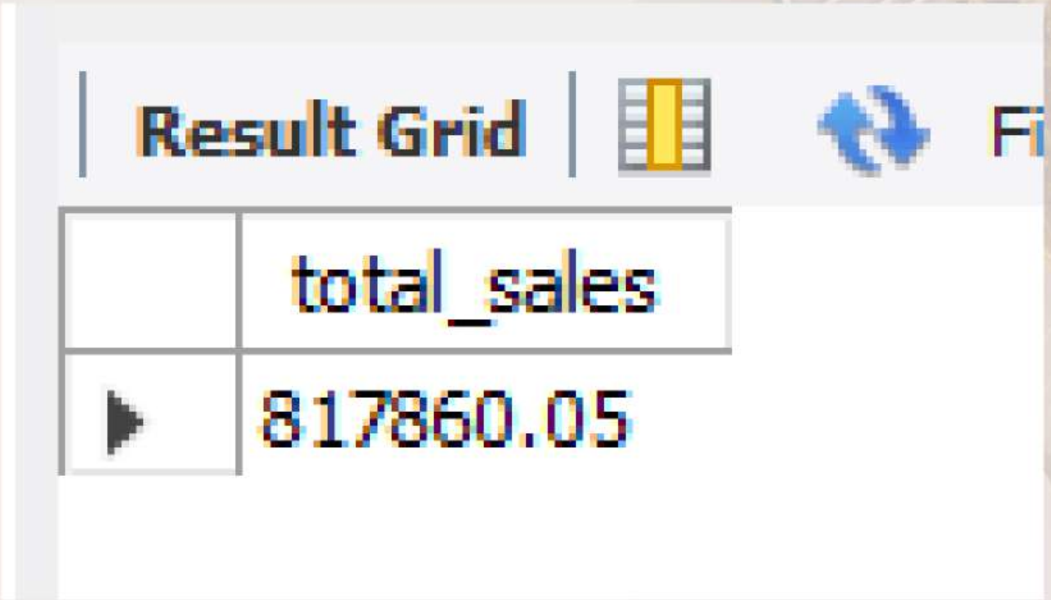


total_orders
21350

Calculate the total revenue generated from pizza sales

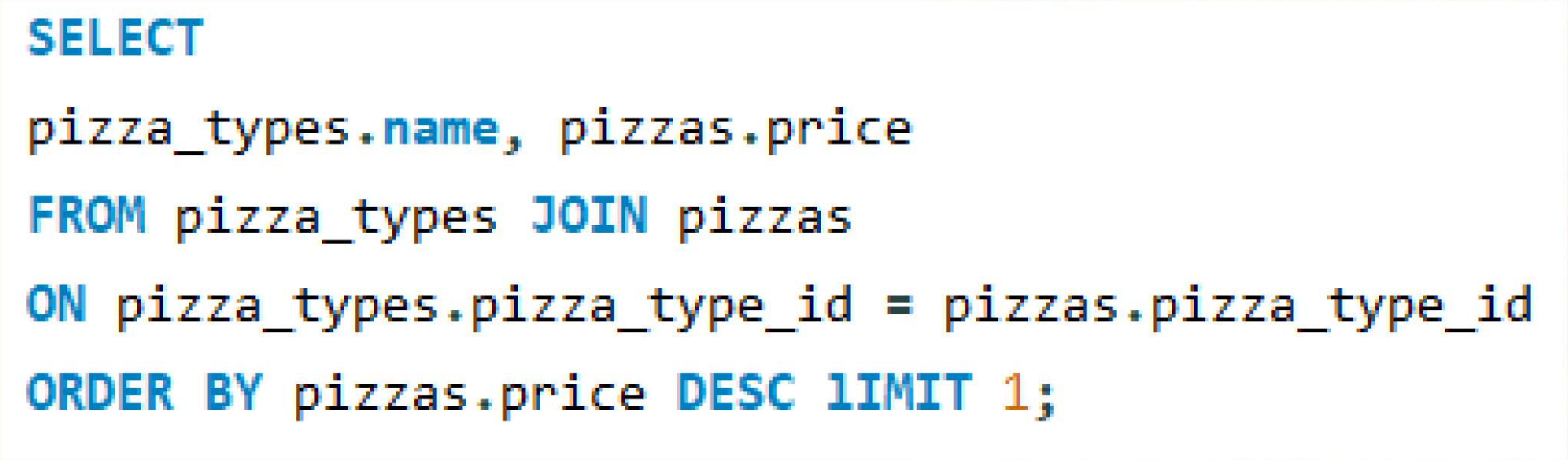


```sql
SELECT
ROUND(sum(order_detail.quantity * pizzas.price),2) AS total_sales
FROM order_detail JOIN pizzas
ON order_detail.pizza_id = pizzas.pizza_id;
```

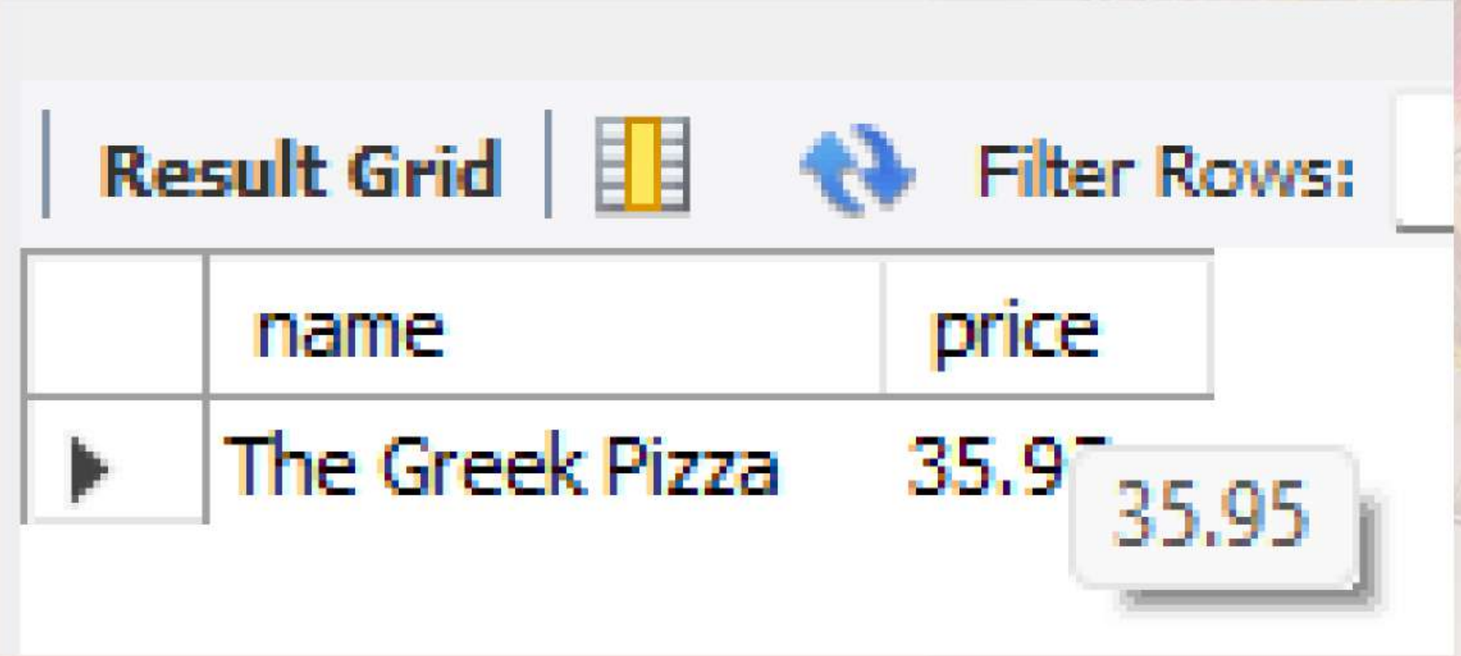


total_sales
817860.05

Identify the highest-priced pizza.

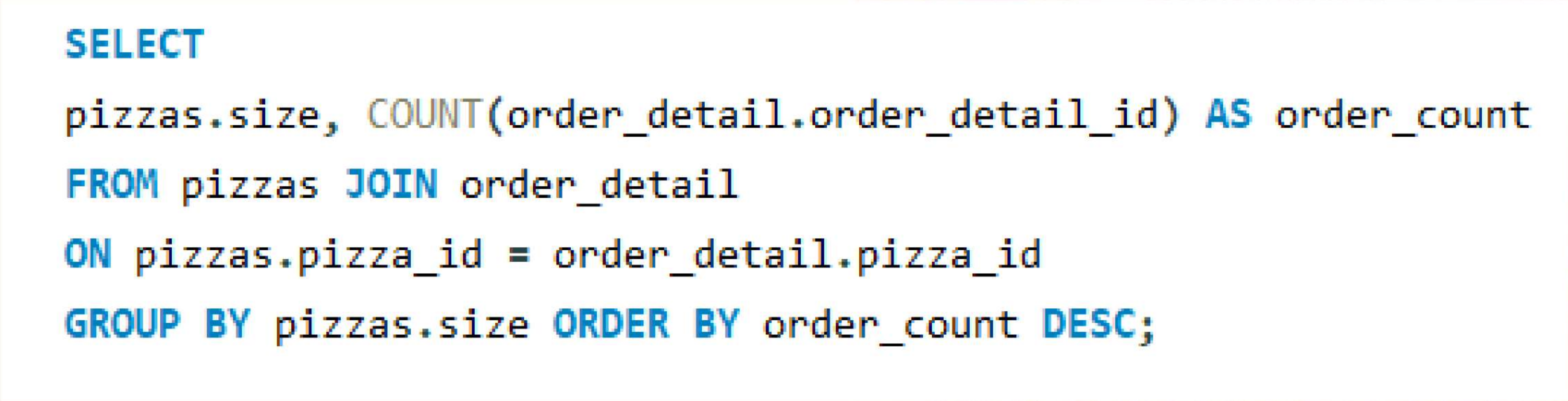


```sql
SELECT
pizza_types.name, pizzas.price
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC lIMIT 1;
```

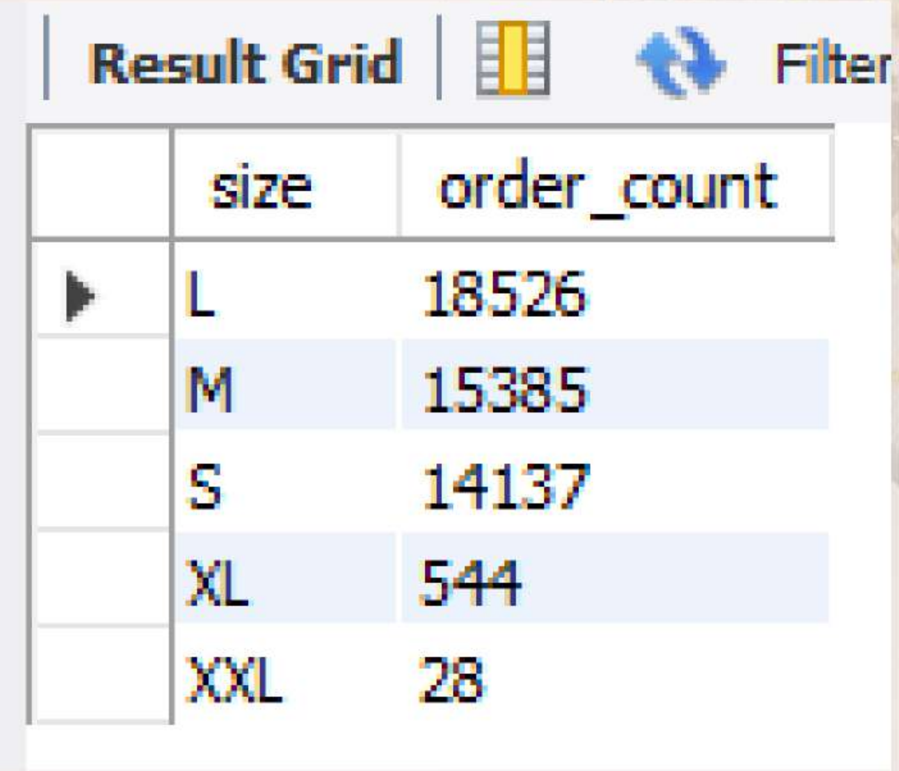


	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

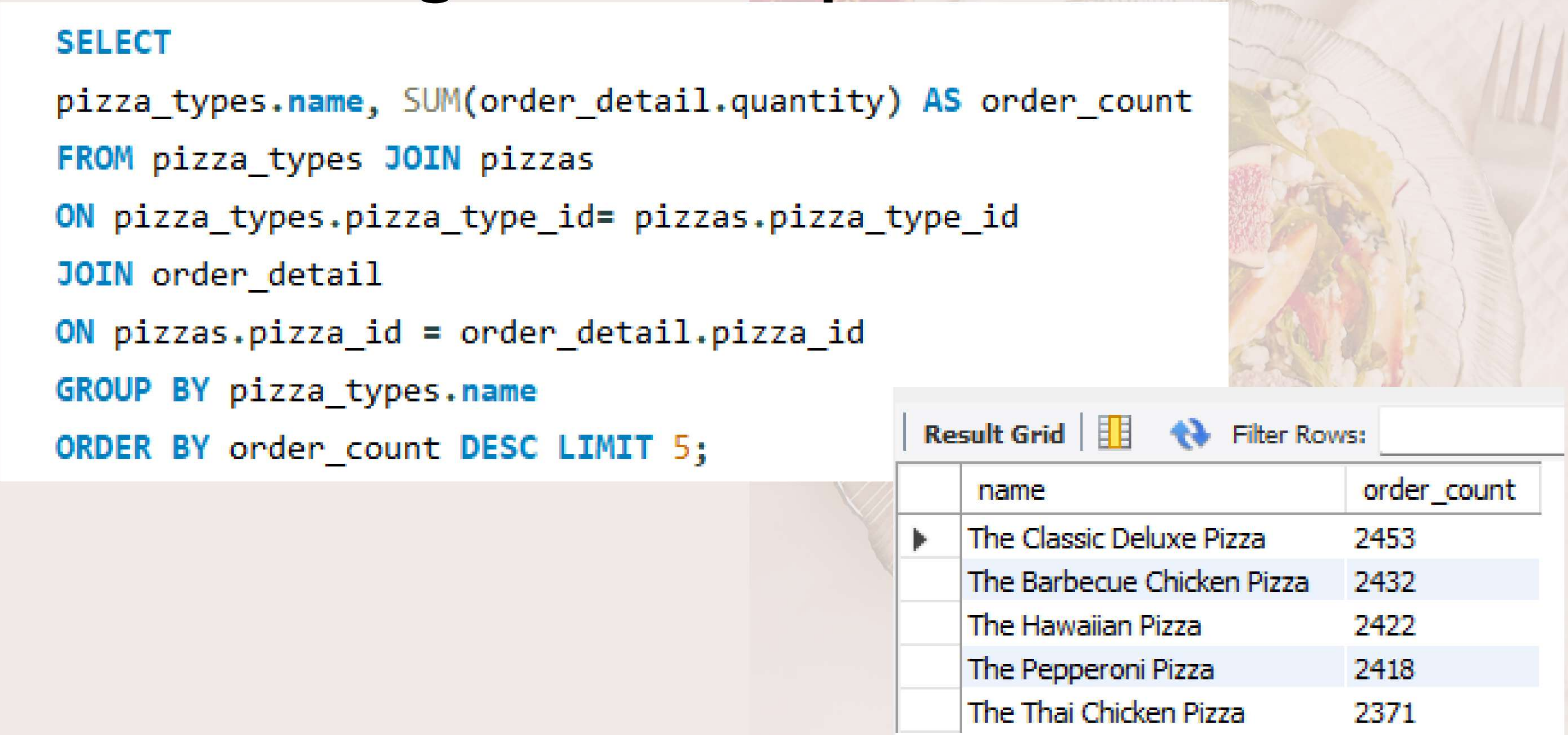


```
SELECT
pizzas.size, COUNT(order_detail.order_detail_id) AS order_count
FROM pizzas JOIN order_detail
ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizzas.size ORDER BY order_count DESC;
```



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

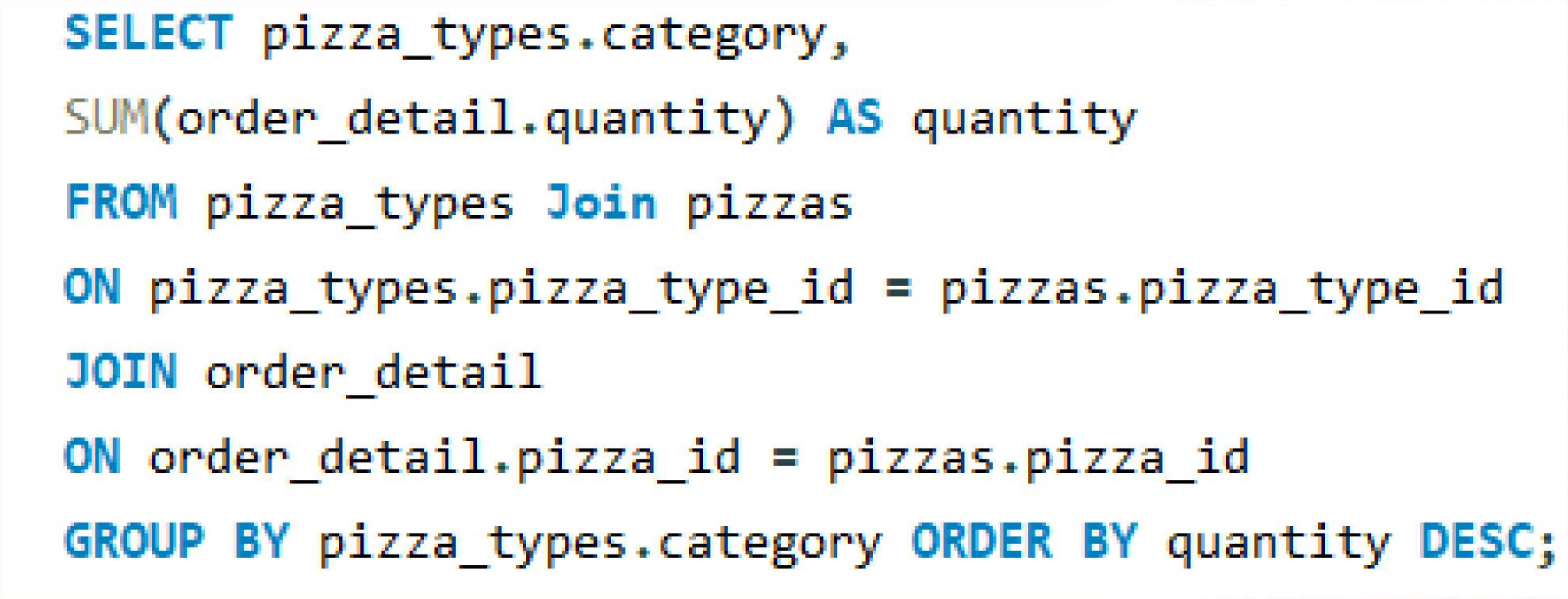
List the top 5 most ordered pizza types along with their quantities.



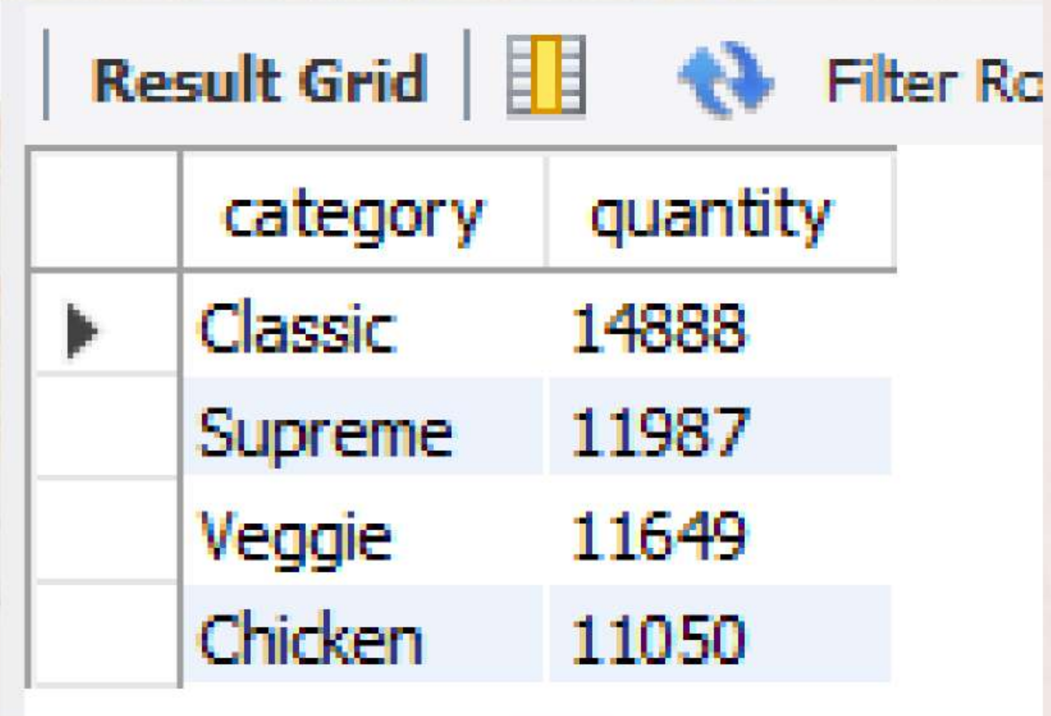
```
SELECT
pizza_types.name, SUM(order_detail.quantity) AS order_count
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id= pizzas.pizza_type_id
JOIN order_detail
ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizza_types.name
ORDER BY order_count DESC LIMIT 5;
```

name	order_count
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.

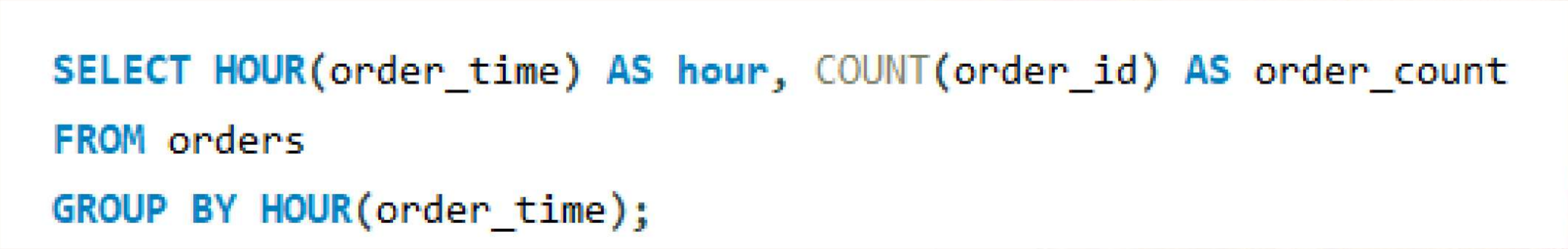


```
SELECT pizza_types.category,
SUM(order_detail.quantity) AS quantity
FROM pizza_types Join pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_detail
ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category ORDER BY quantity DESC;
```

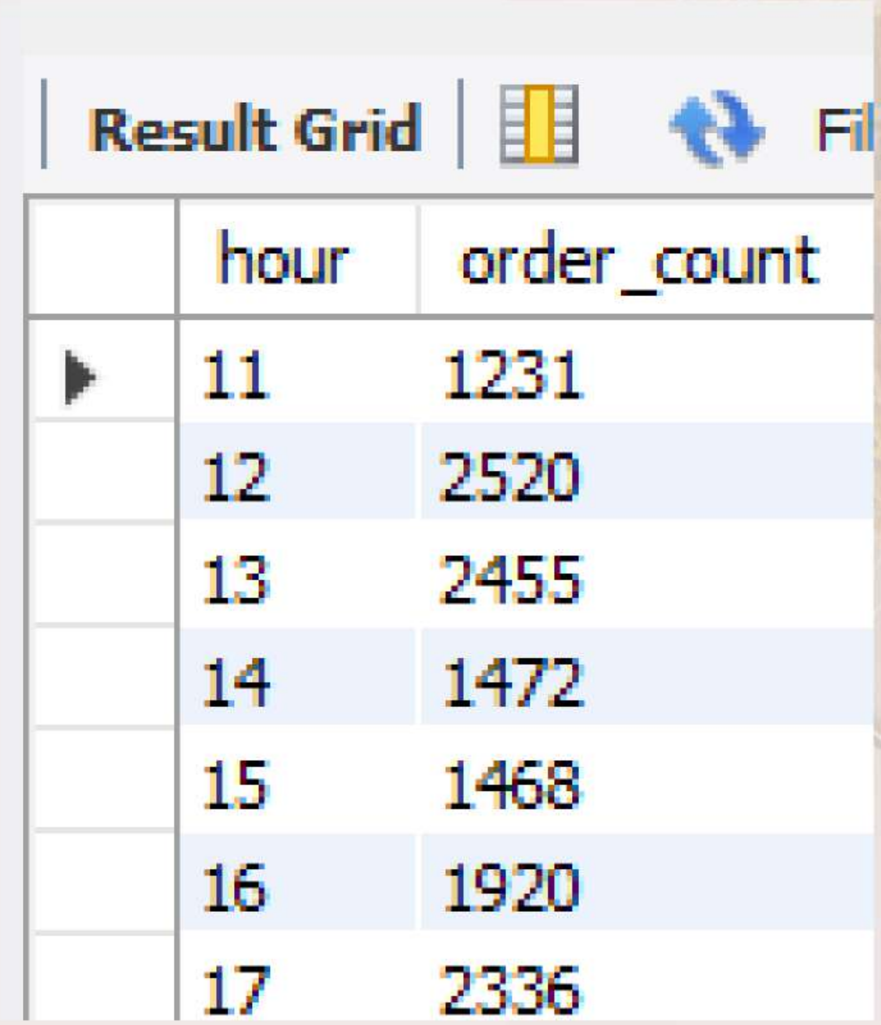


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

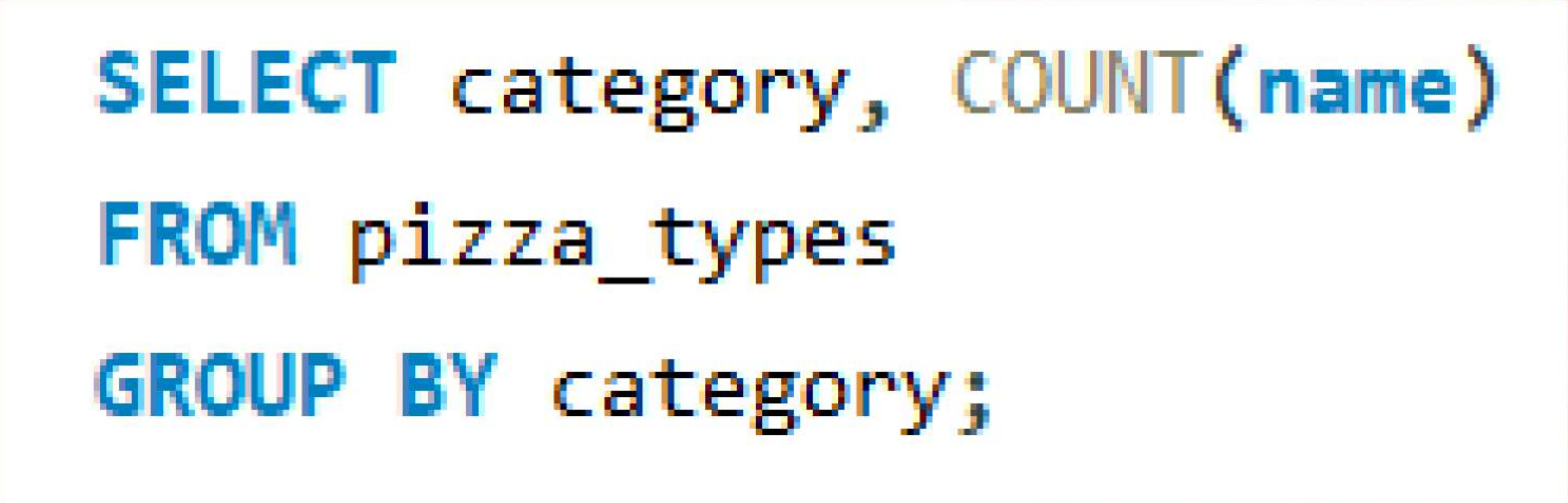


```sql
SELECT HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM orders
GROUP BY HOUR(order_time);
```

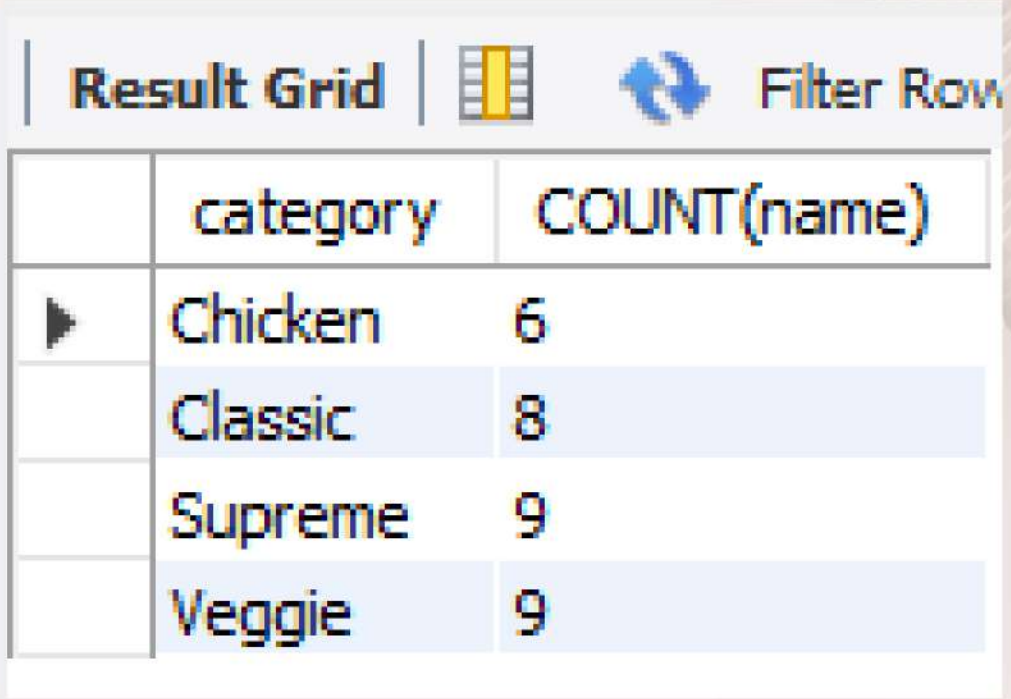


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

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

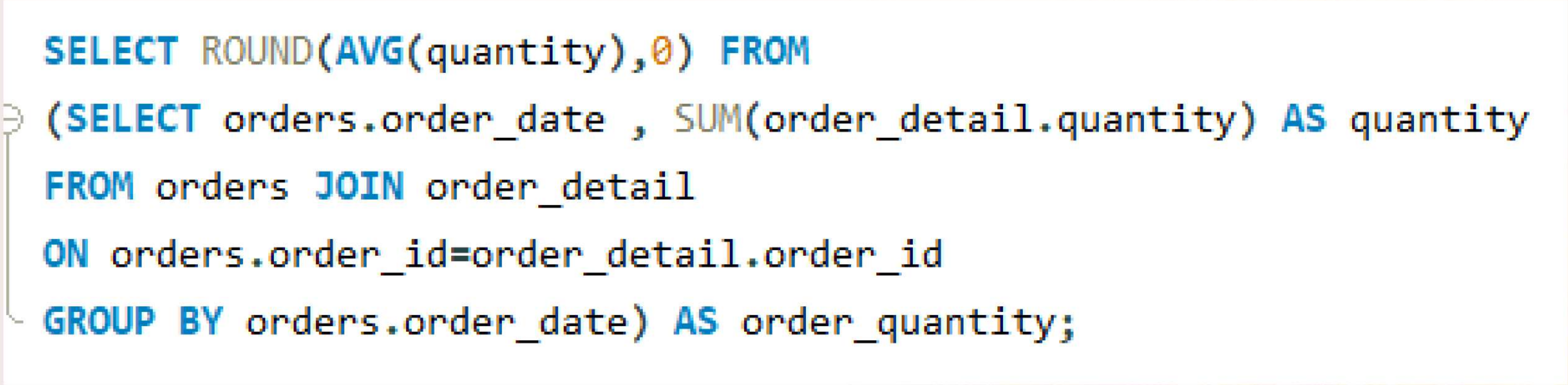


```sql
SELECT category, COUNT(name)
FROM pizza_types
GROUP BY category;
```

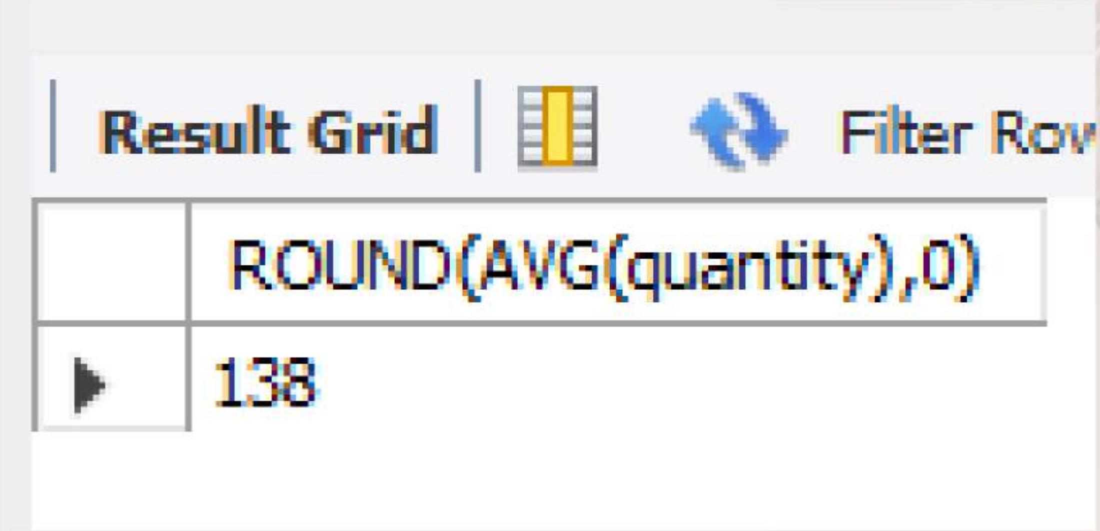


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

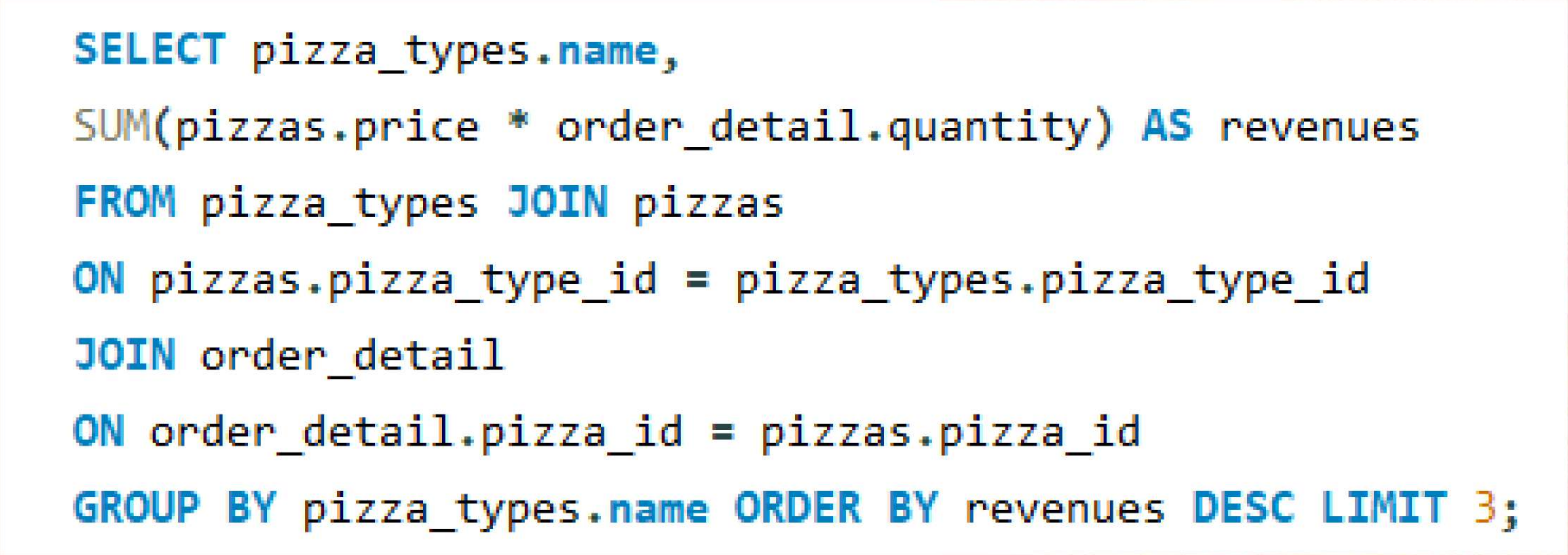


```
SELECT ROUND(AVG(quantity),0) FROM
(SELECT orders.order_date , SUM(order_detail.quantity) AS quantity
FROM orders JOIN order_detail
ON orders.order_id=order_detail.order_id
GROUP BY orders.order_date) AS order_quantity;
```

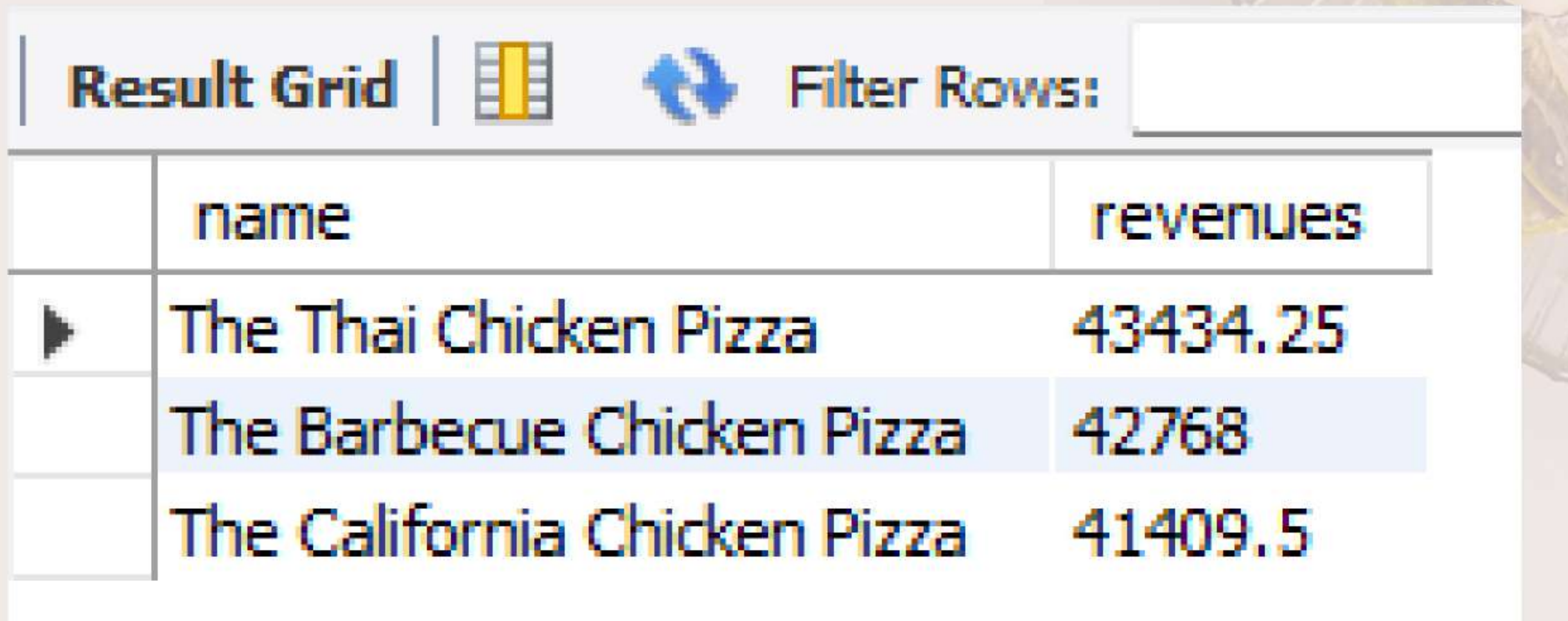


```
ROUND(AVG(quantity),0)
138
```

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

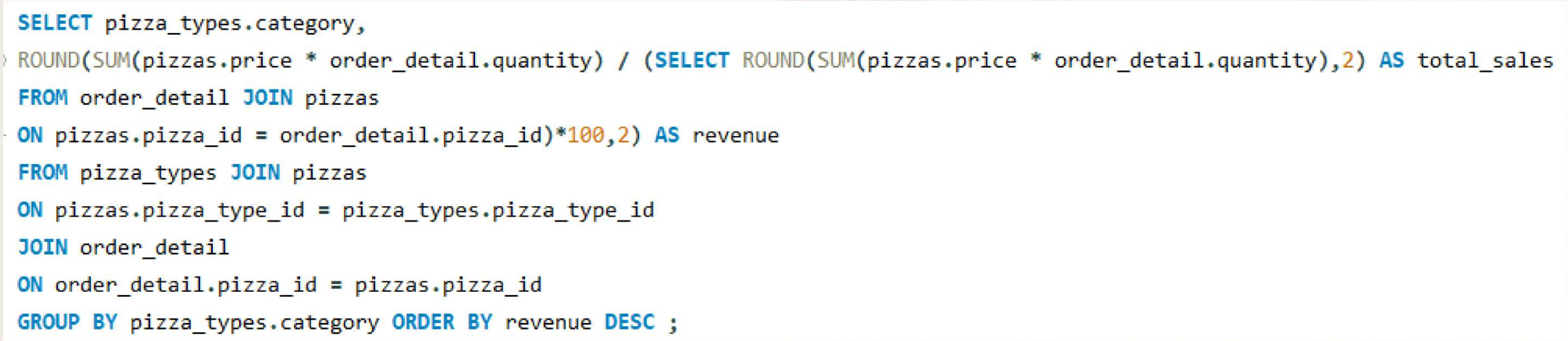


```
SELECT pizza_types.name,
SUM(pizzas.price * order_detail.quantity) AS revenues
FROM pizza_types JOIN pizzas
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_detail
ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name ORDER BY revenues DESC LIMIT 3;
```

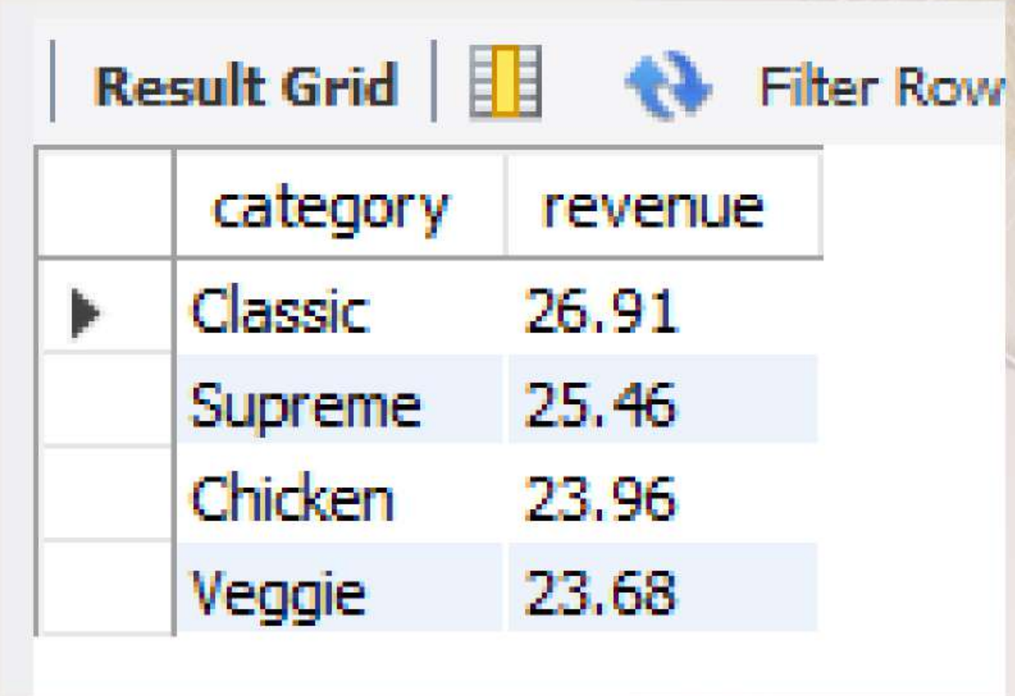


name	revenues
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue

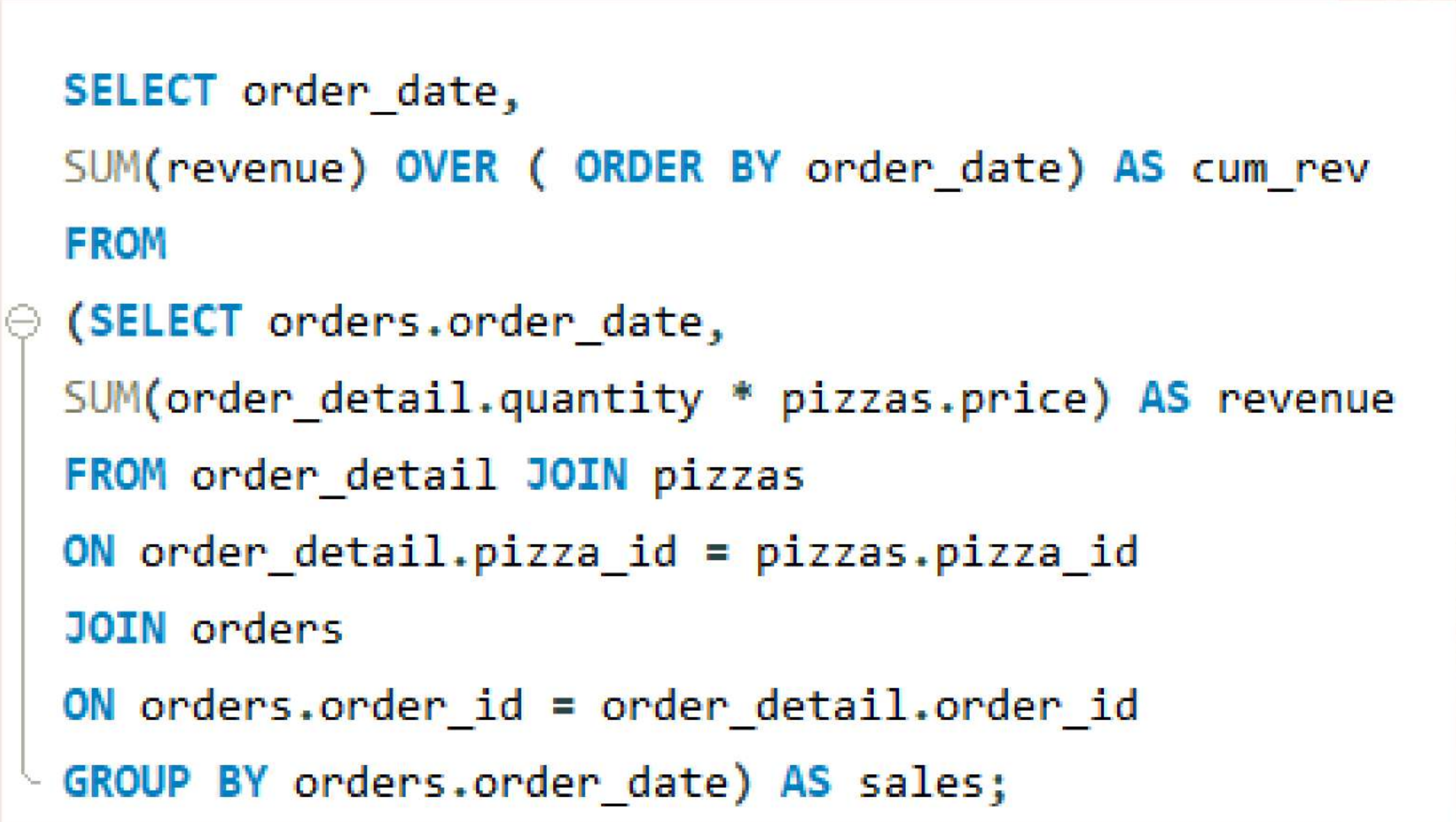


```
SELECT pizza_types.category,
ROUND(SUM(pizzas.price * order_detail.quantity) / (SELECT ROUND(SUM(pizzas.price * order_detail.quantity),2) AS total_sales
FROM order_detail JOIN pizzas
ON pizzas.pizza_id = order_detail.pizza_id)*100,2) AS revenue
FROM pizza_types JOIN pizzas
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_detail
ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category ORDER BY revenue DESC ;
```

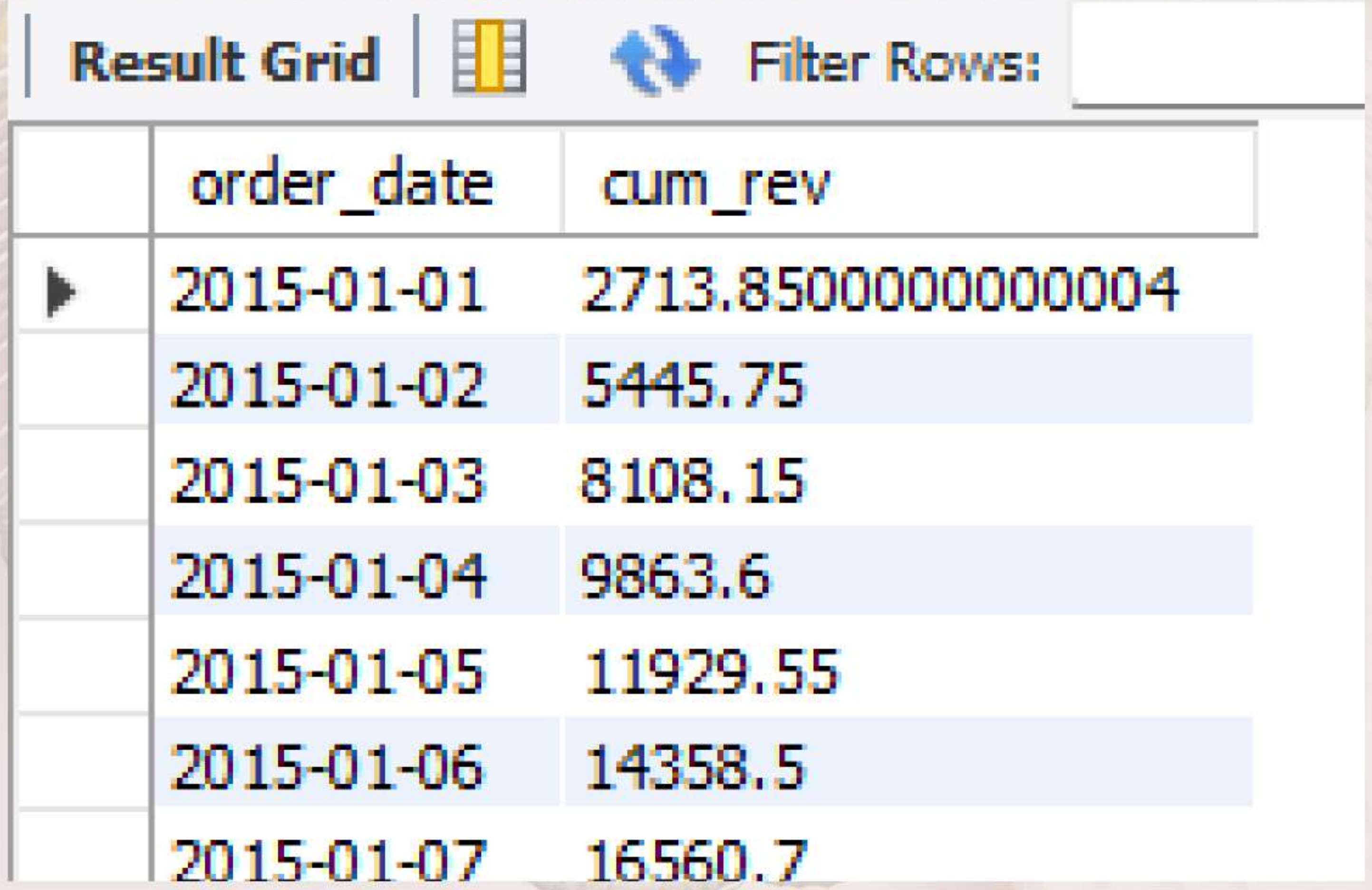


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.



```
SELECT order_date,
SUM(revenue) OVER ( ORDER BY order_date) AS cum_rev
FROM
(SELECT orders.order_date,
SUM(order_detail.quantity * pizzas.price) AS revenue
FROM order_detail JOIN pizzas
ON order_detail.pizza_id = pizzas.pizza_id
JOIN orders
ON orders.order_id = order_detail.order_id
GROUP BY orders.order_date) AS sales;
```



order_date	cum_rev
2015-01-01	2713.8500000000004
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