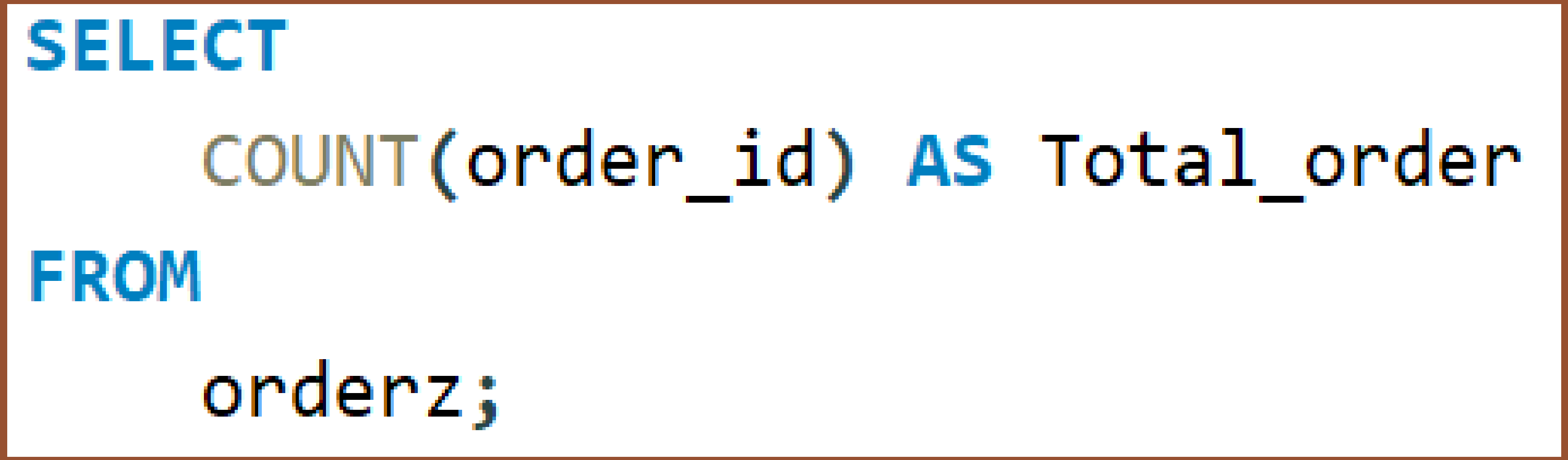
La Pizza

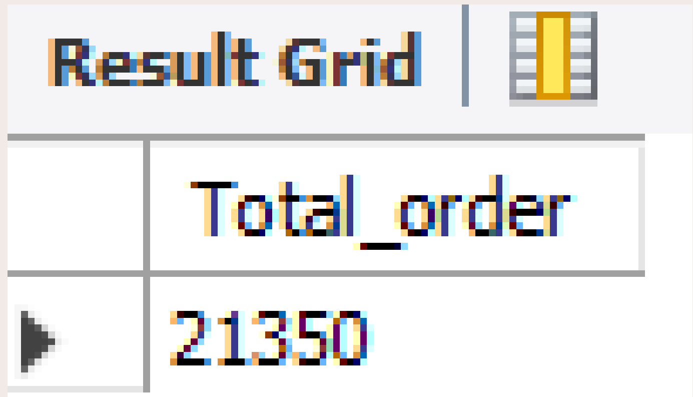
Introduction

This SQL project for a pizza store aims to analyze sales and ordering patterns to gain actionable insights. Key objectives include retrieving the total number of orders placed and calculating total revenue from pizza sales. We will identify the highest-priced pizza and the most common pizza size ordered to understand customer preferences. The project will list the top 5 most ordered pizza types and their quantities, and join necessary tables to find the total quantity of each pizza category ordered. We will analyze the distribution of orders by hour to identify peak times and determine the category-wise distribution of pizzas. By grouping orders by date, we will calculate the average number of pizzas ordered per day. Additionally, we will identify the top 3 most ordered pizza types based on revenue and calculate the percentage contribution of each type to total revenue. The analysis will include tracking cumulative revenue over time and determining the top 3 most ordered pizza types based on revenue for each category. Utilizing SQL, this project will provide the pizza store with valuable insights to optimize operations and improve market strategies.

Total number of orders placed

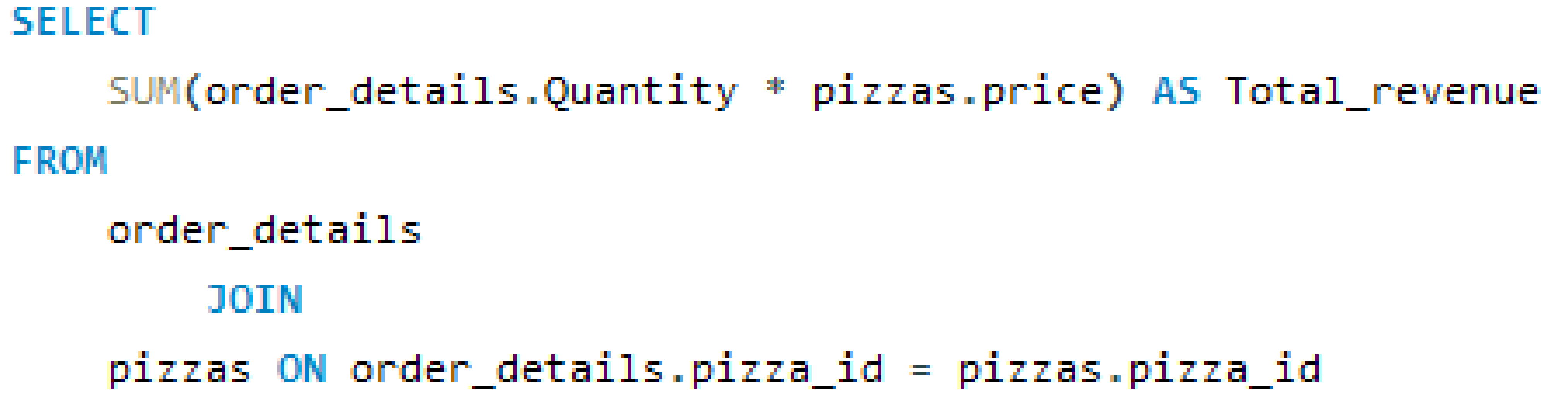


```
SELECT
    COUNT(order_id) AS Total_order
FROM
    orderz;
```

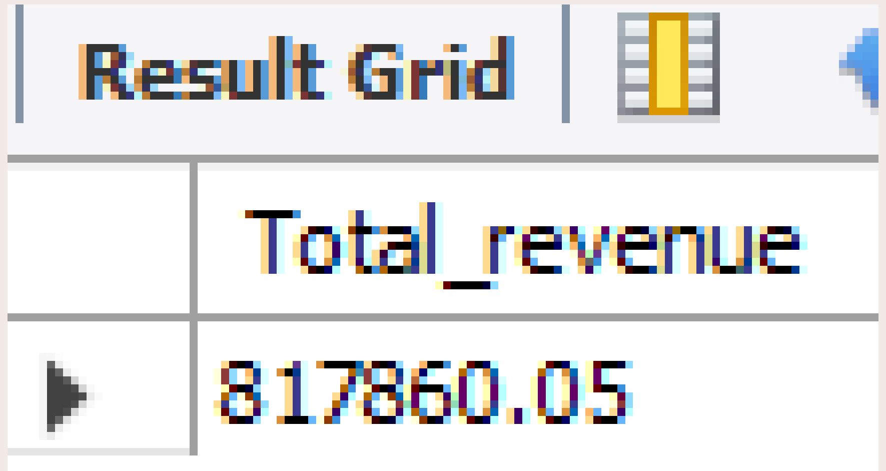


```
Total_order
21350
```

The total revenue generated from pizza sales

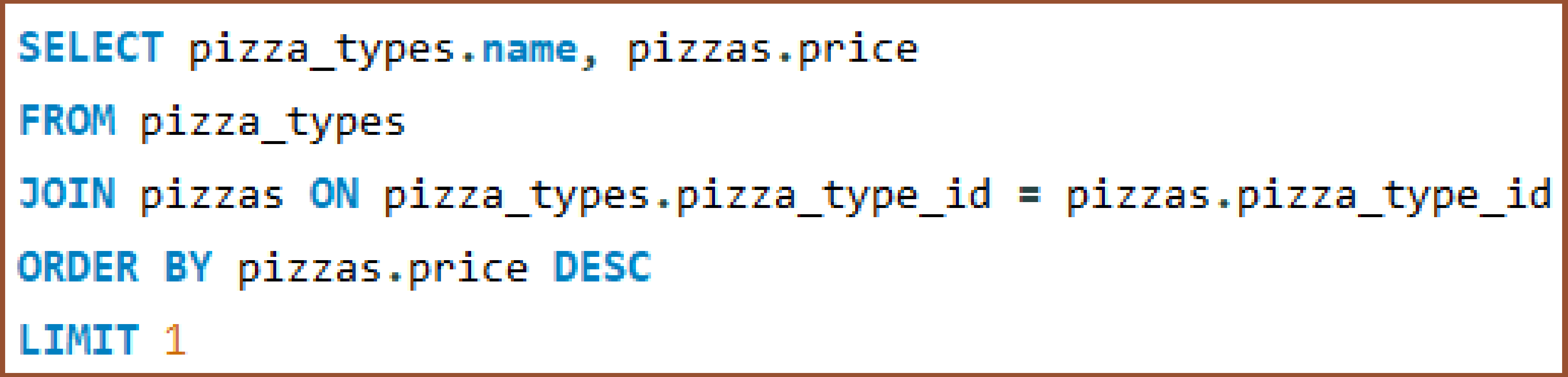


```sql
SELECT
    SUM(order_details.Quantity * pizzas.price) AS Total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

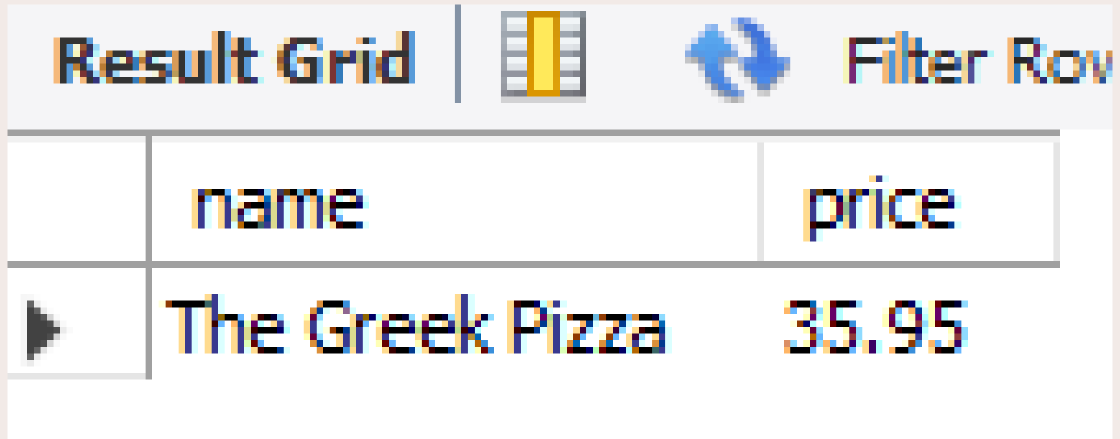


Total_revenue
817860.05

The highest-priced pizza

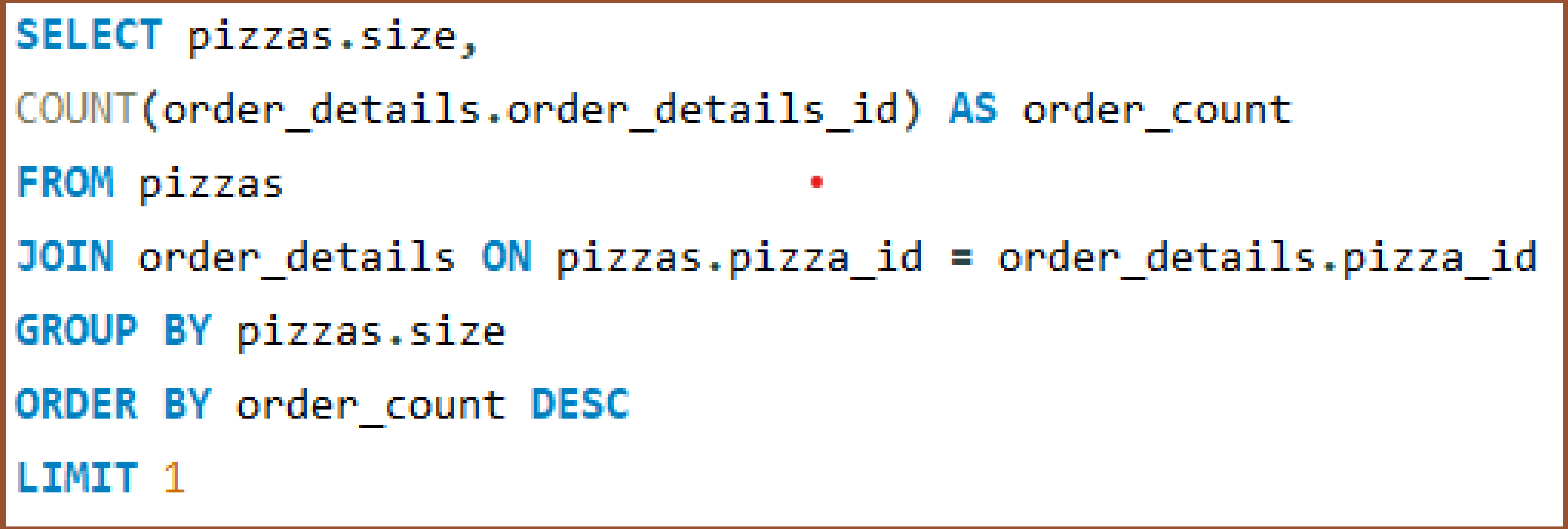


```sql
SELECT pizza_types.name, pizzas.price
FROM pizza_types
JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1
```

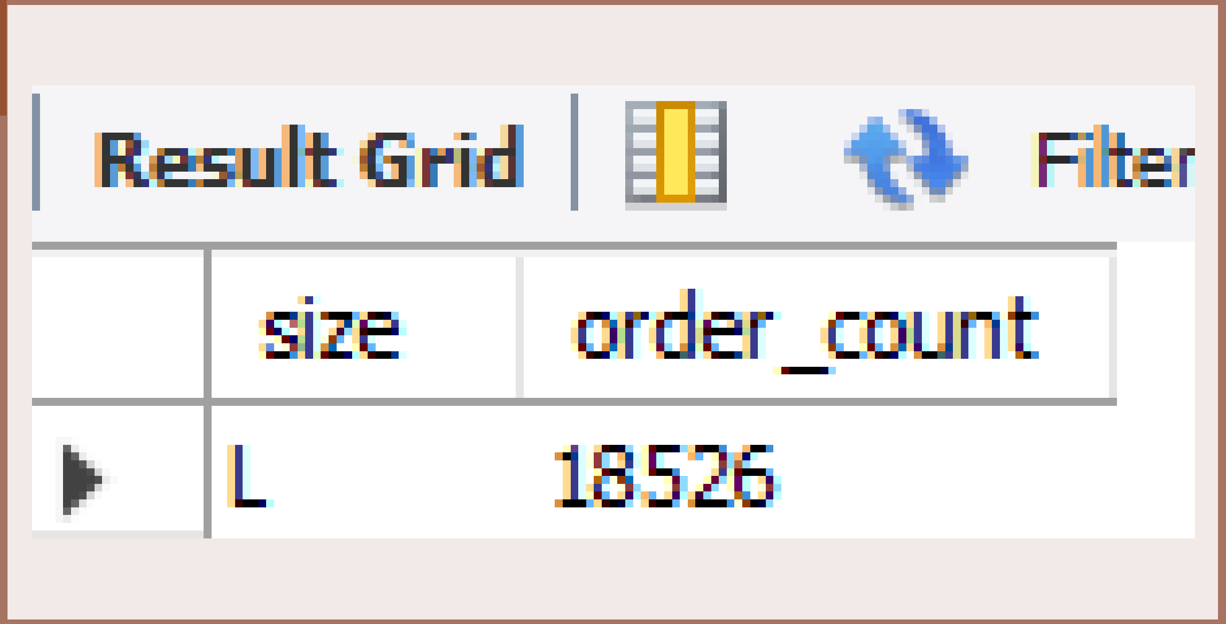


name	price
The Greek Pizza	35.95

The most common pizza size ordered

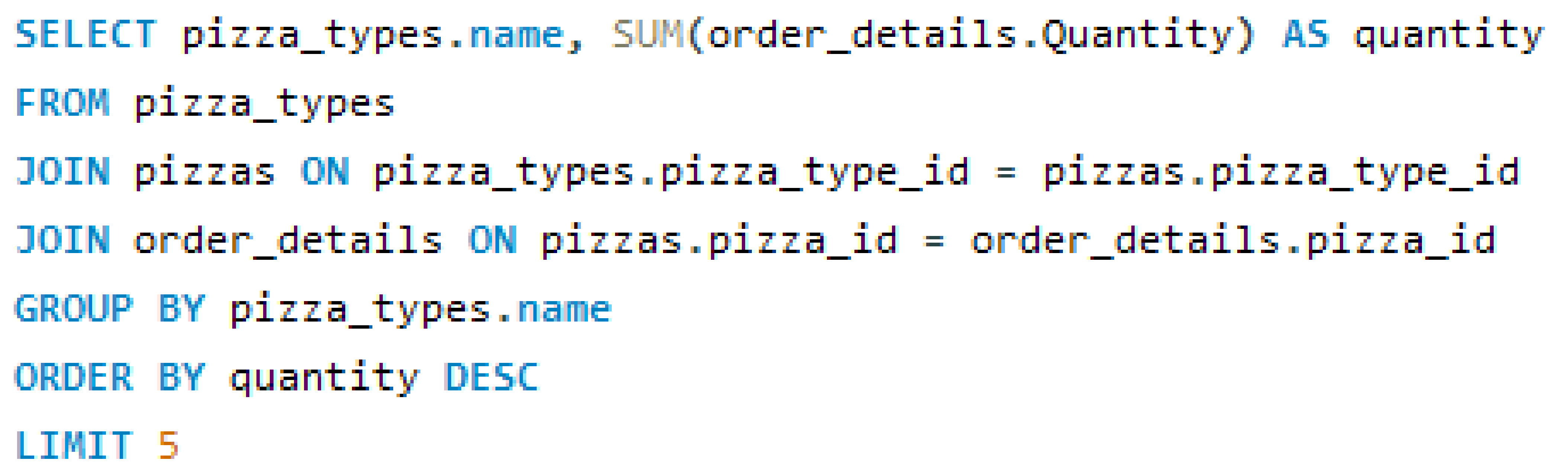


```
SELECT pizzas.size,
COUNT(order_details.order_details_id) AS order_count
FROM pizzas
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1
```

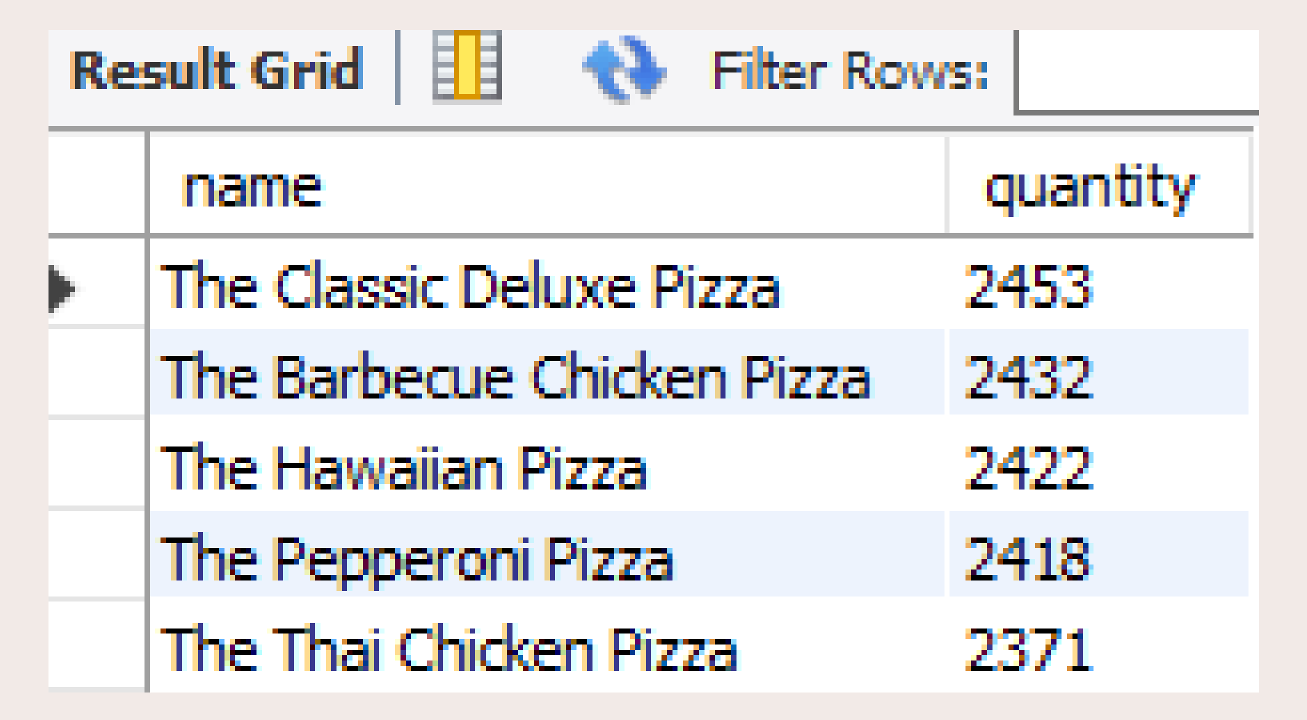


size	order_count
L	18526

Top 5 most ordered pizza types

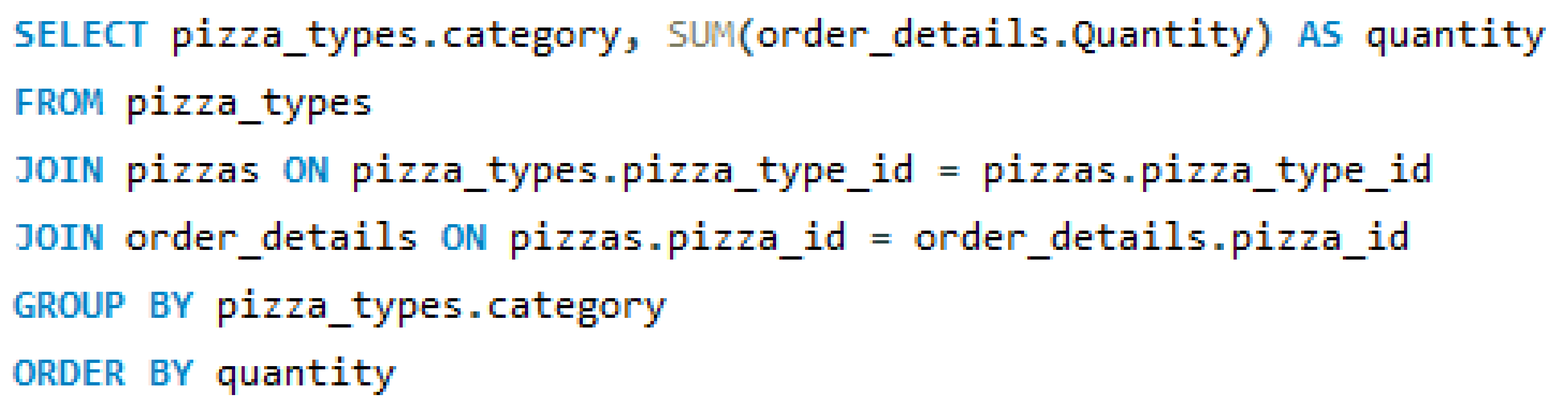


```
SELECT pizza_types.name, SUM(order_details.Quantity) AS quantity
FROM pizza_types
JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
```

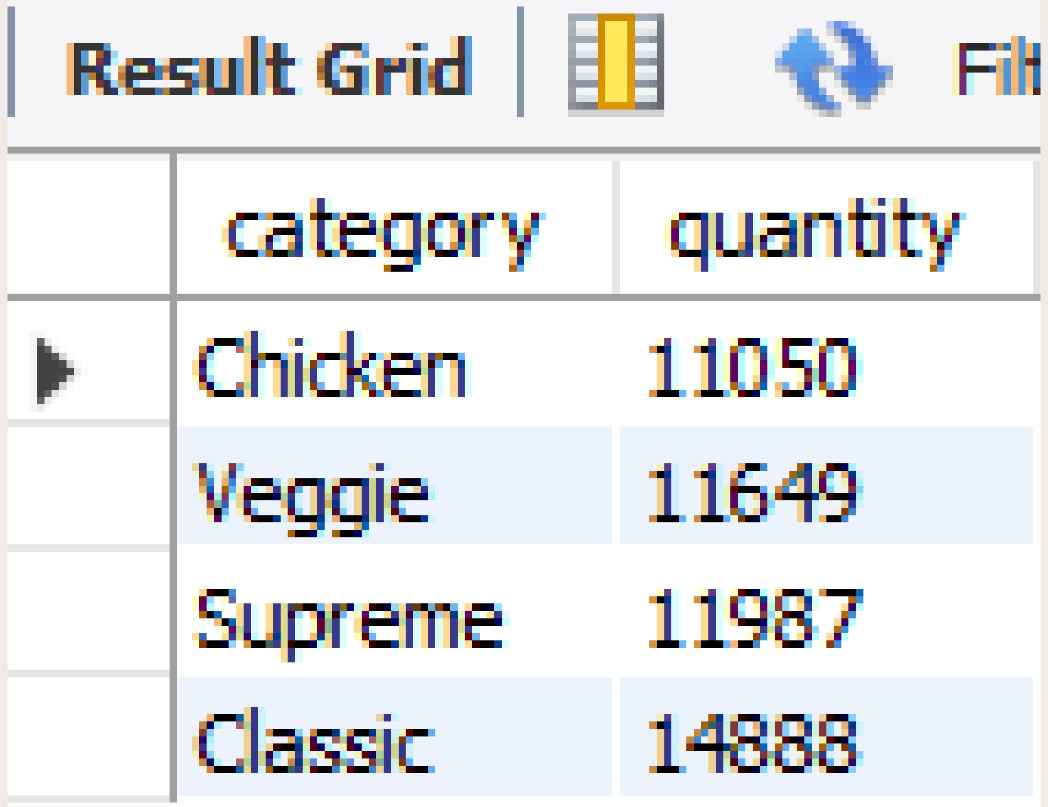


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Total quantity of each pizza category ordered

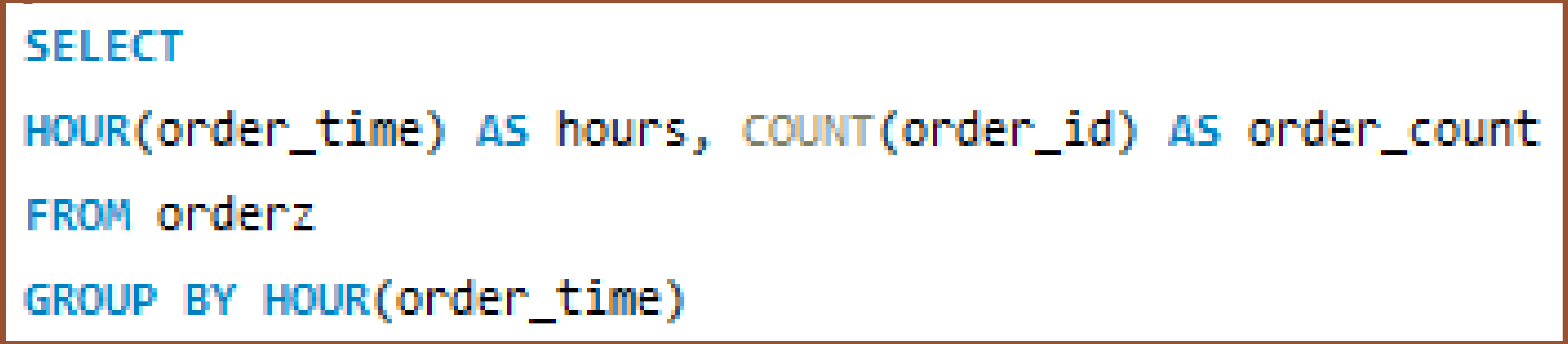


```
SELECT pizza_types.category, SUM(order_details.Quantity) AS quantity
FROM pizza_types
JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity
```

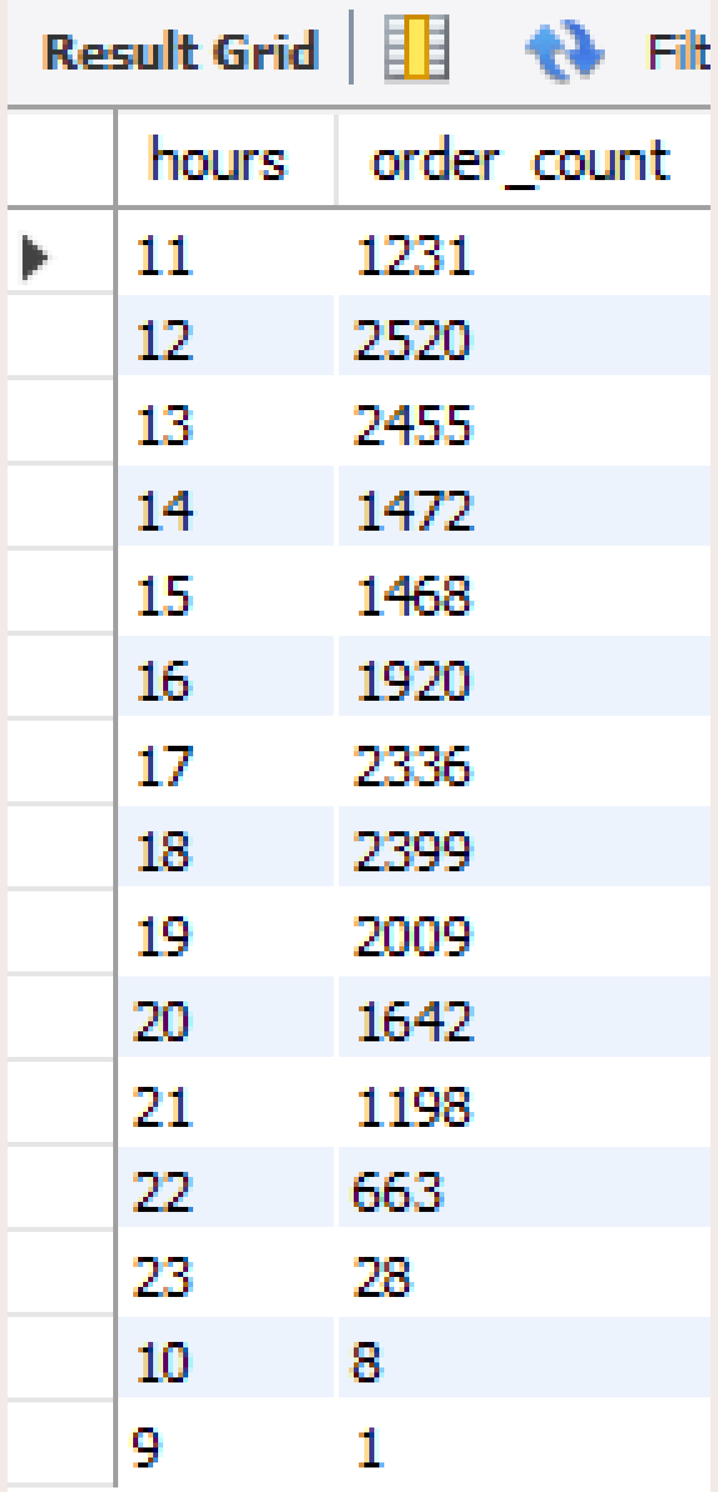


category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

The distribution of orders by hour of the day.

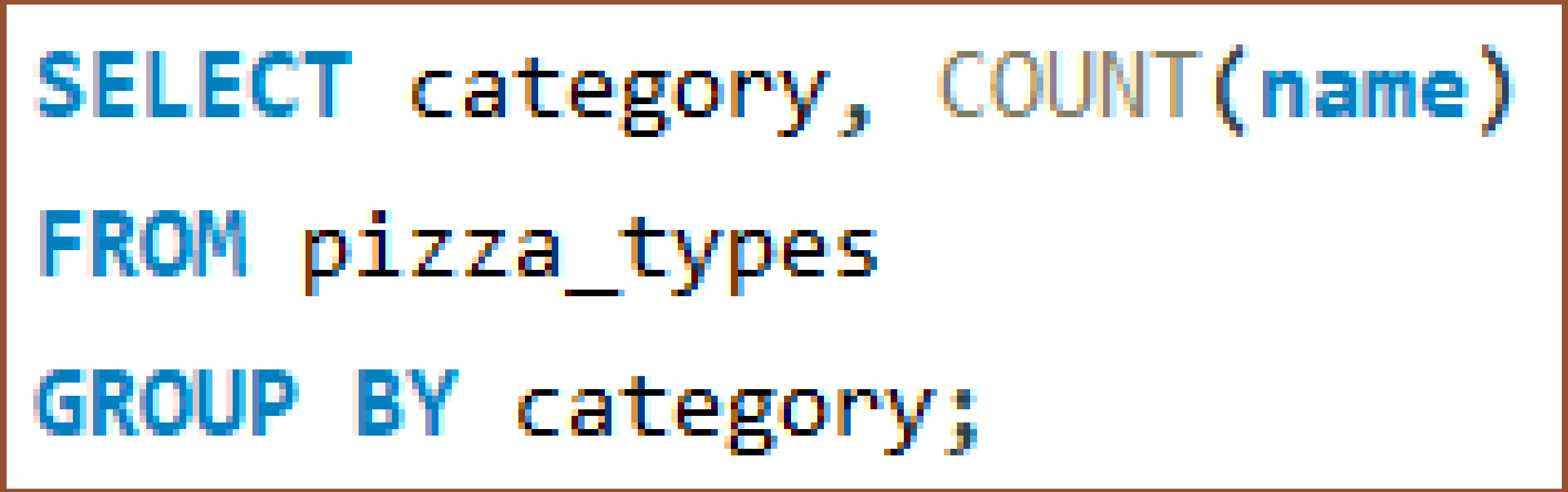


```sql
SELECT
HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM orderz
GROUP BY HOUR(order_time)
```

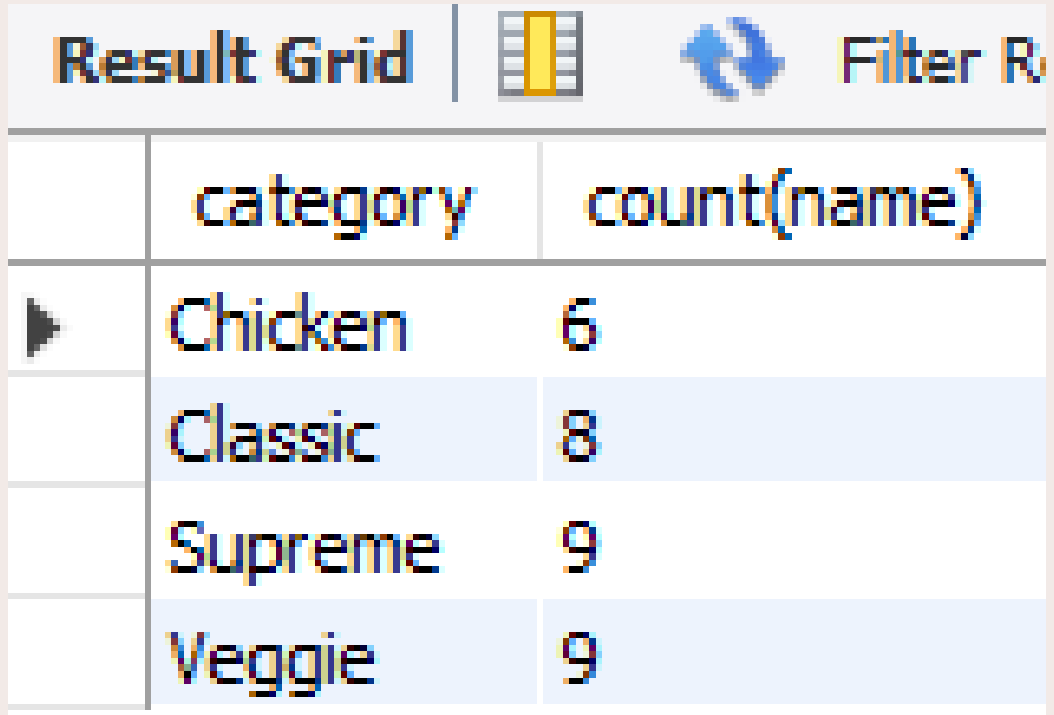


hours	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Category-wise distribution of pizzas

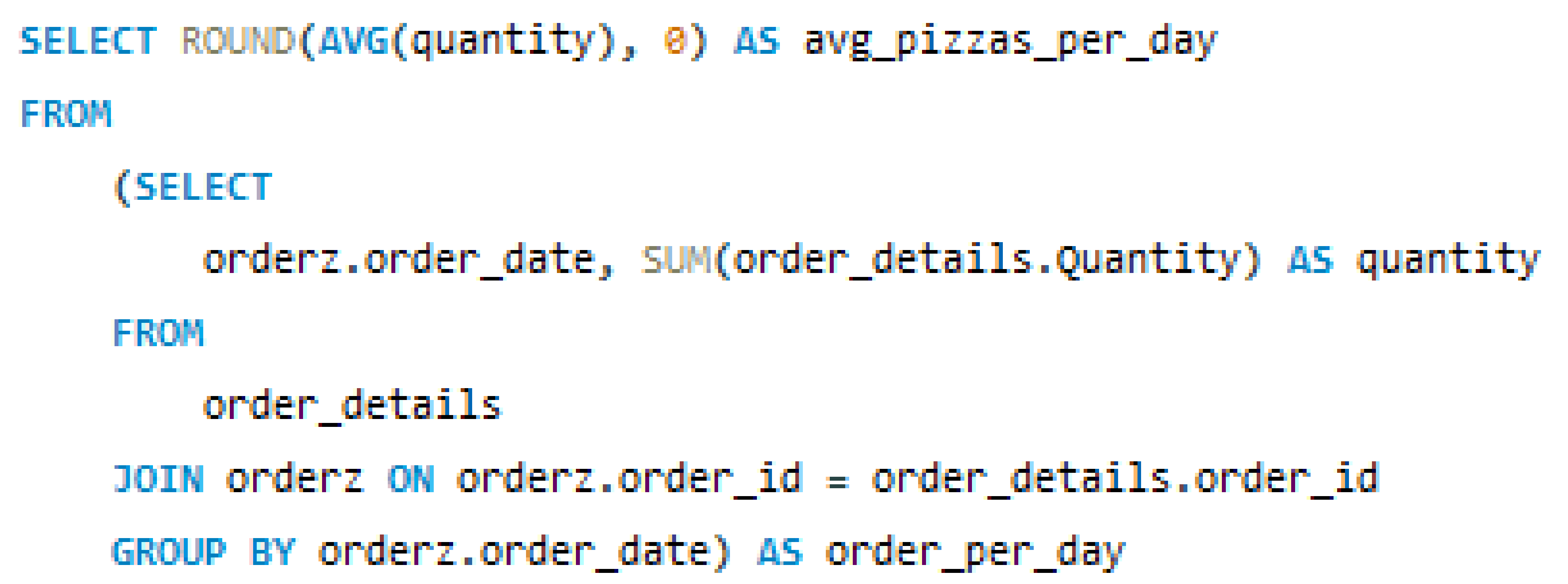


```
SELECT category, COUNT(name)
FROM pizza_types
GROUP BY category;
```

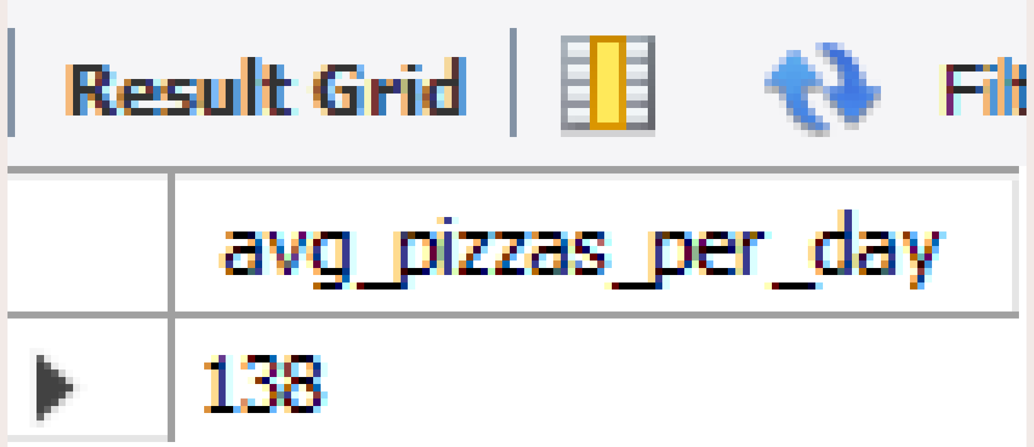


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Average number of pizzas ordered per day

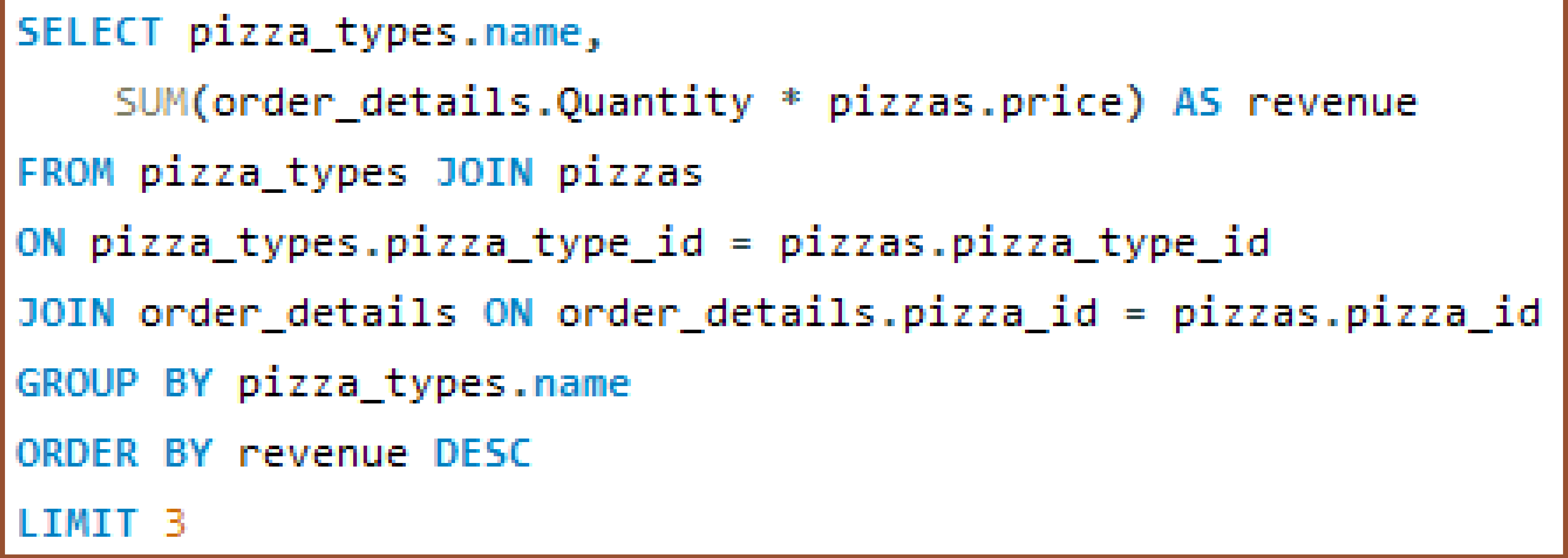


```
SELECT ROUND(AVG(quantity), 0) AS avg_pizzas_per_day
FROM
    (SELECT
        orderz.order_date, SUM(order_details.Quantity) AS quantity
    FROM
        order_details
    JOIN orderz ON orderz.order_id = order_details.order_id
    GROUP BY orderz.order_date) AS order_per_day
```

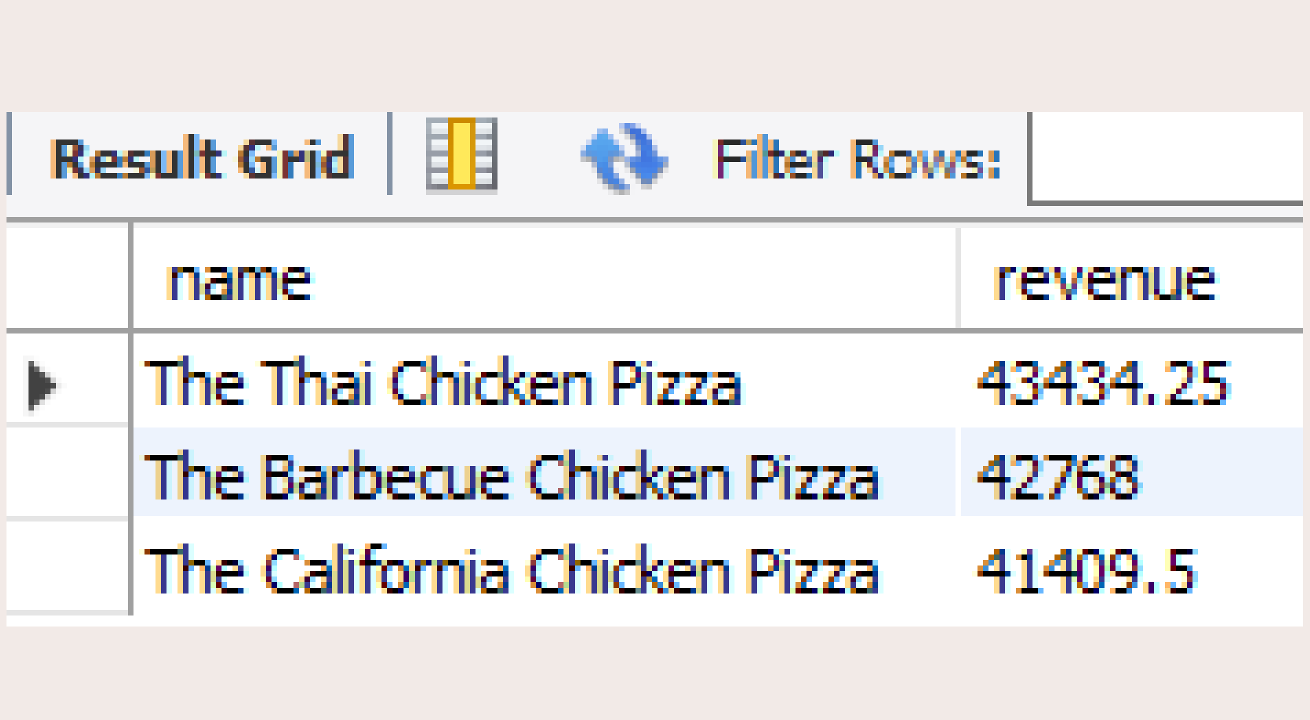


avg_pizzas_per_day
138

Top 3 most ordered pizza types based on revenue

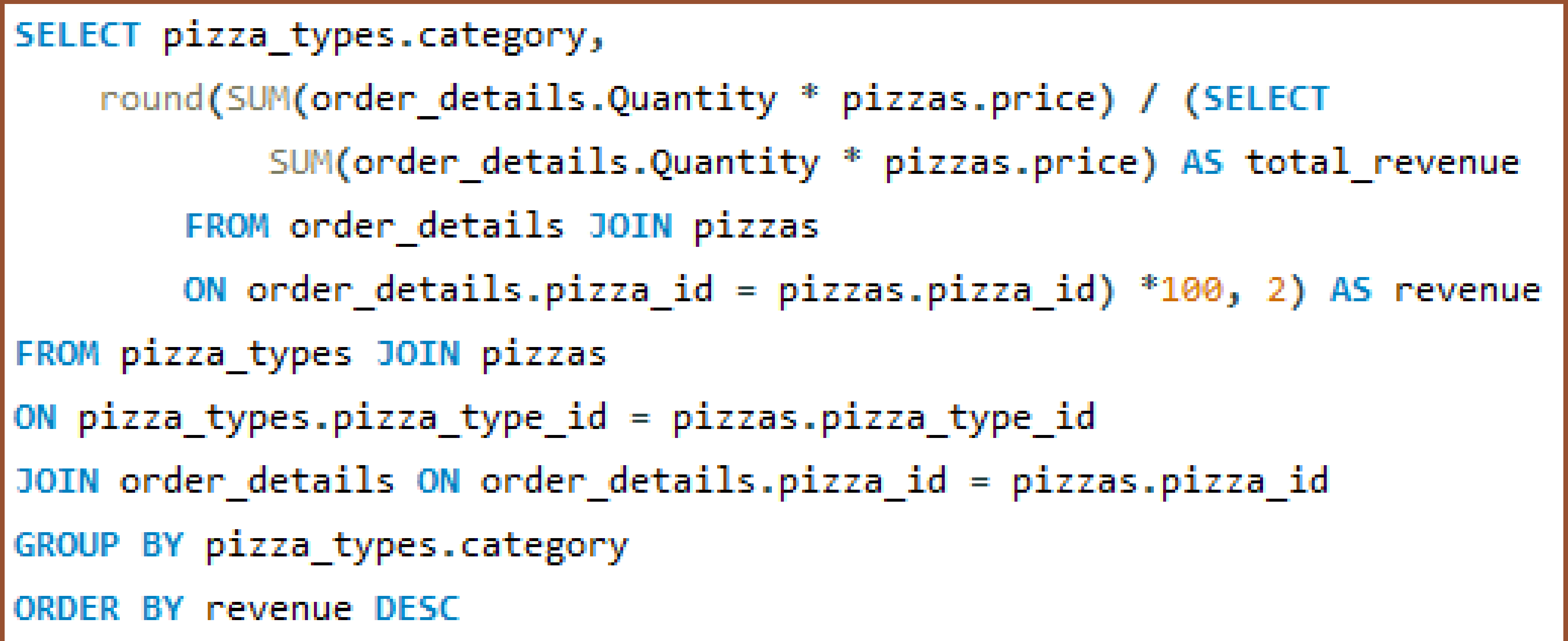


```sql
SELECT pizza_types.name,
    SUM(order_details.Quantity * pizzas.price) AS revenue
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```

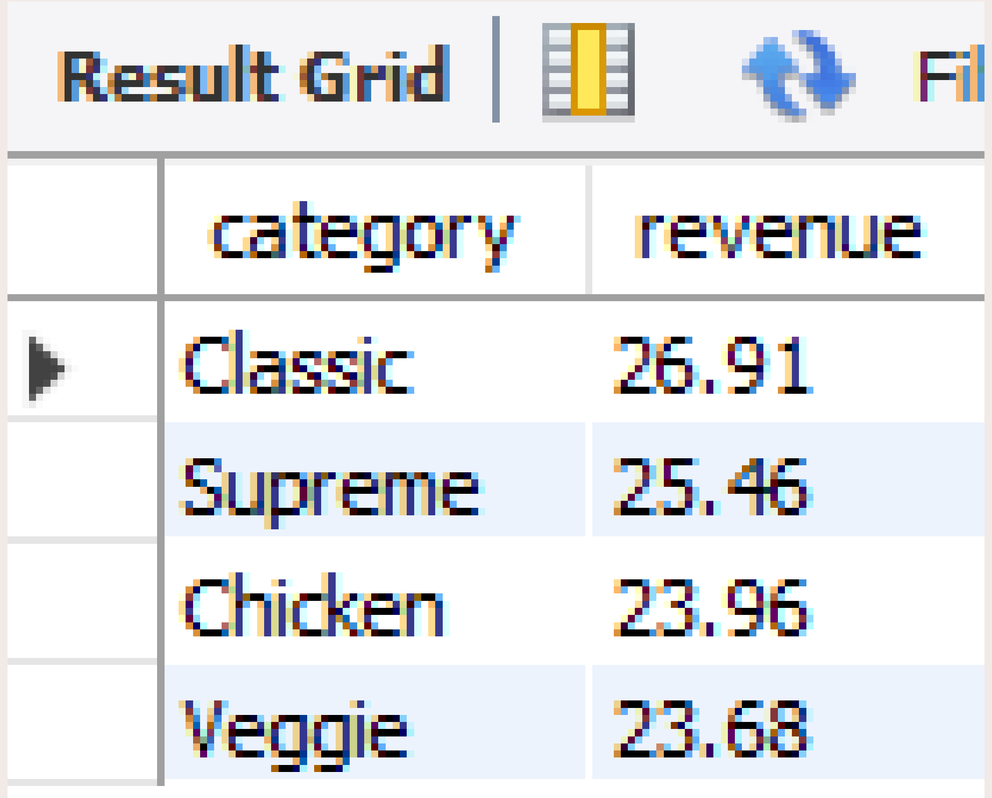


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

The percentage contribution of each pizza type to total revenue

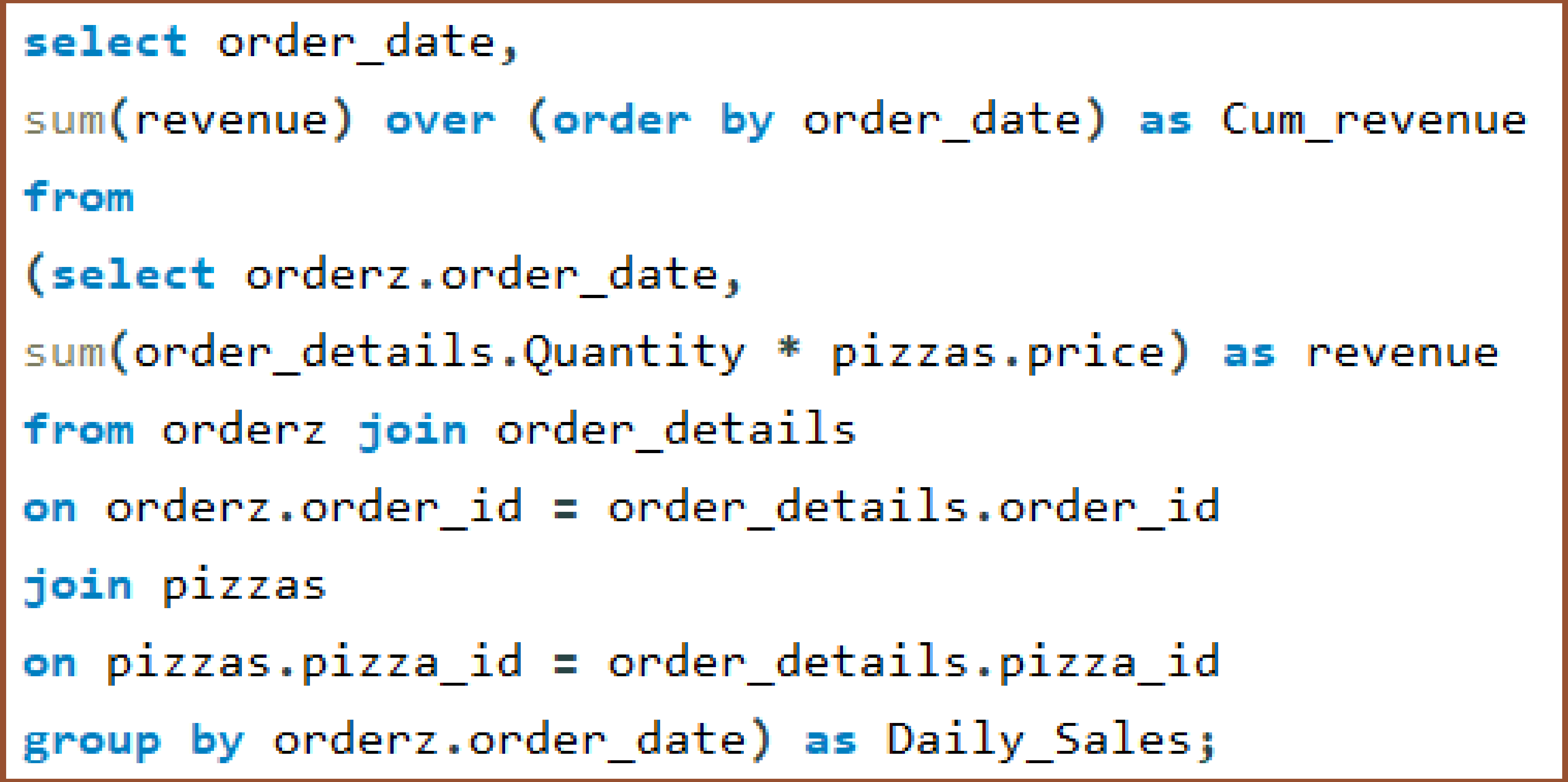


```
SELECT pizza_types.category,
    round(SUM(order_details.Quantity * pizzas.price) / (SELECT
            SUM(order_details.Quantity * pizzas.price) AS total_revenue
        FROM order_details JOIN pizzas
        ON order_details.pizza_id = pizzas.pizza_id) *100, 2) AS revenue
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
```

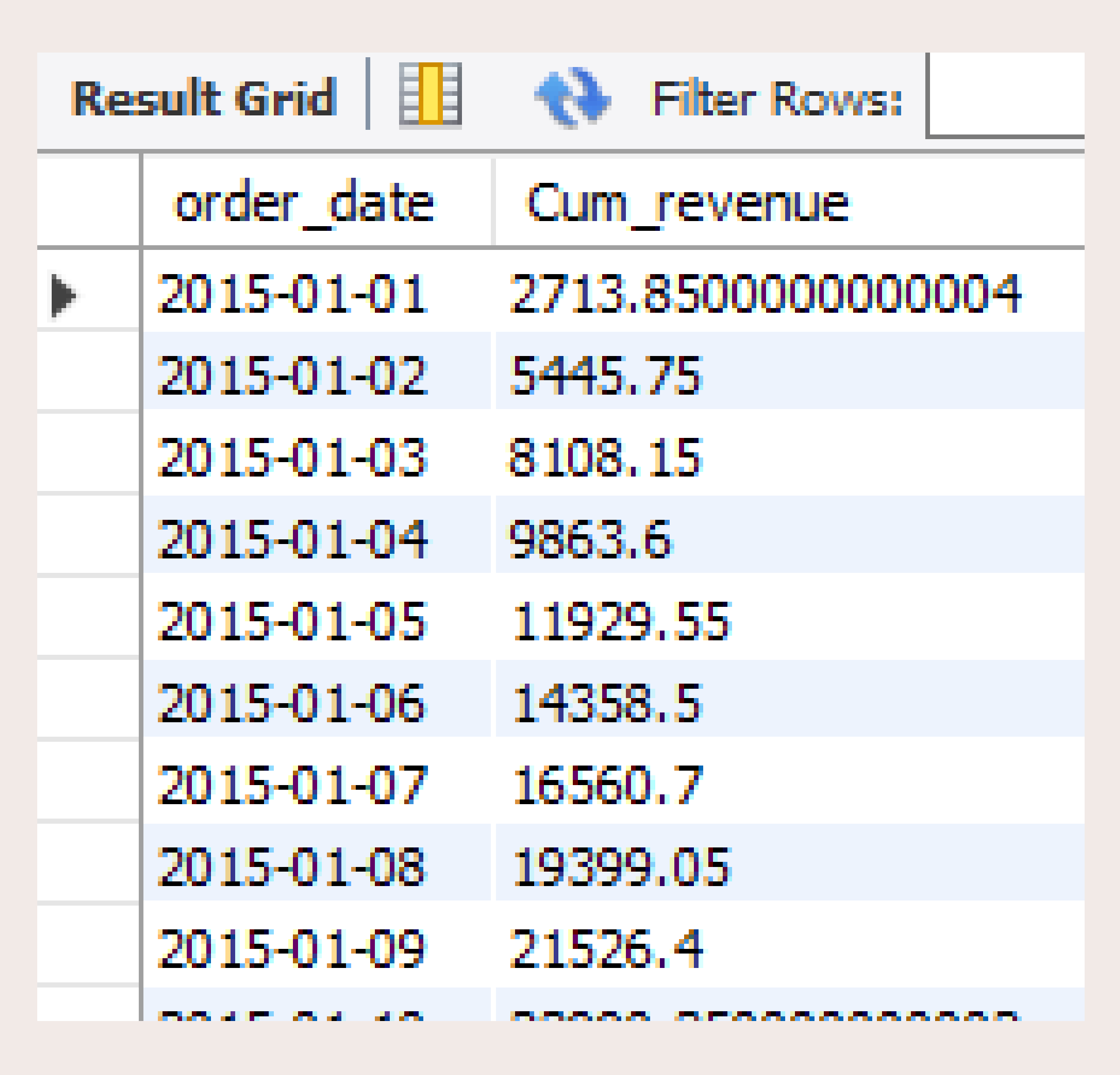


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

The cumulative revenue generated over time

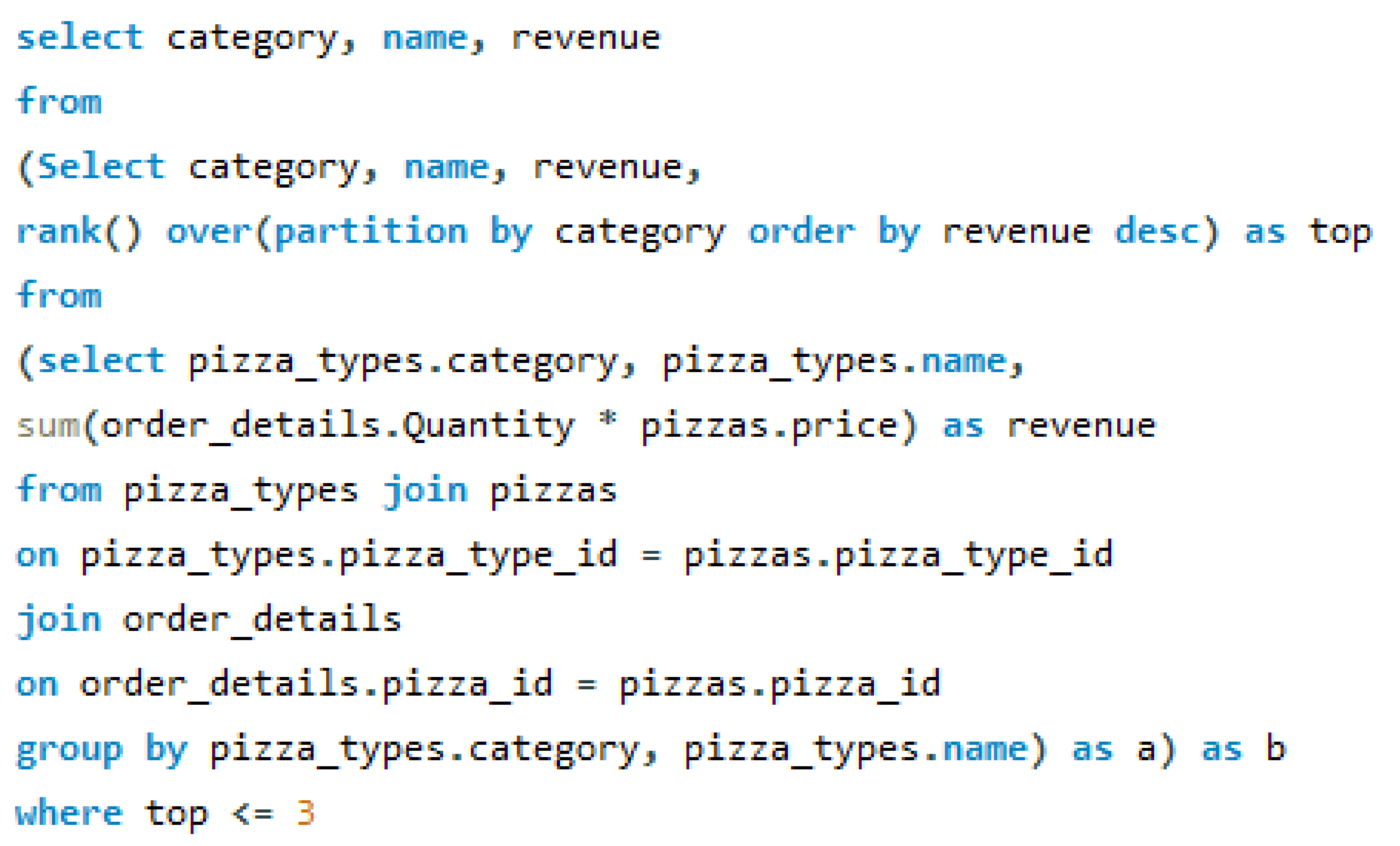


```
select order_date,
sum(revenue) over (order by order_date) as Cum_revenue
from
(select orderz.order_date,
sum(order_details.Quantity * pizzas.price) as revenue
from orderz join order_details
on orderz.order_id = order_details.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orderz.order_date) as Daily_Sales;
```



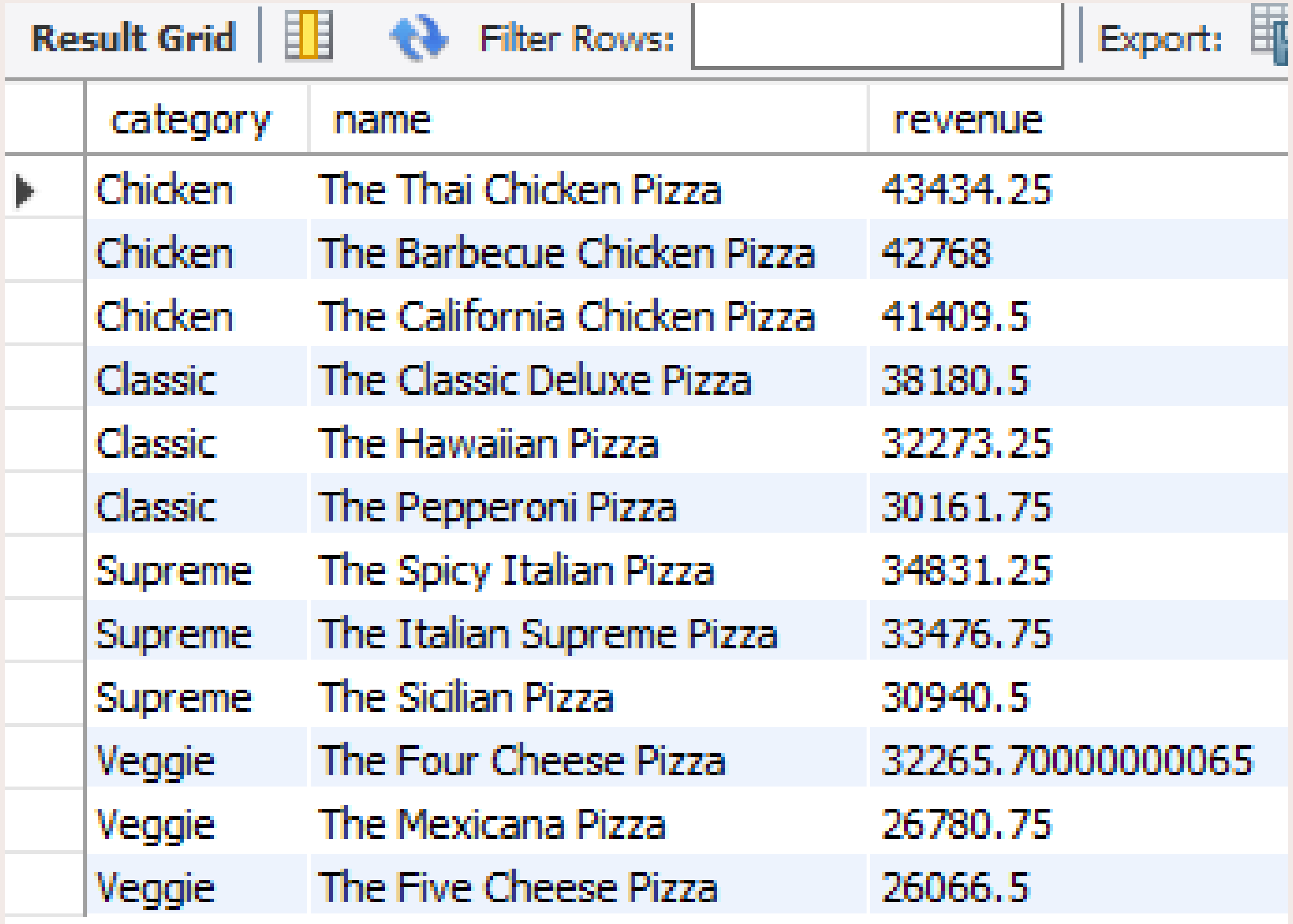
order_date	Cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4

Top 3 most ordered pizza types based on revenue for each pizza category



```
select category, name, revenue
from
(Select category, name, revenue,
rank() over(partition by category order by revenue desc) as top
from
(select pizza_types.category, pizza_types.name,
sum(order_details.Quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where top <= 3
```

Top 3 most ordered pizza types based on revenue for each pizza category



category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Thank You