Ingoude Company

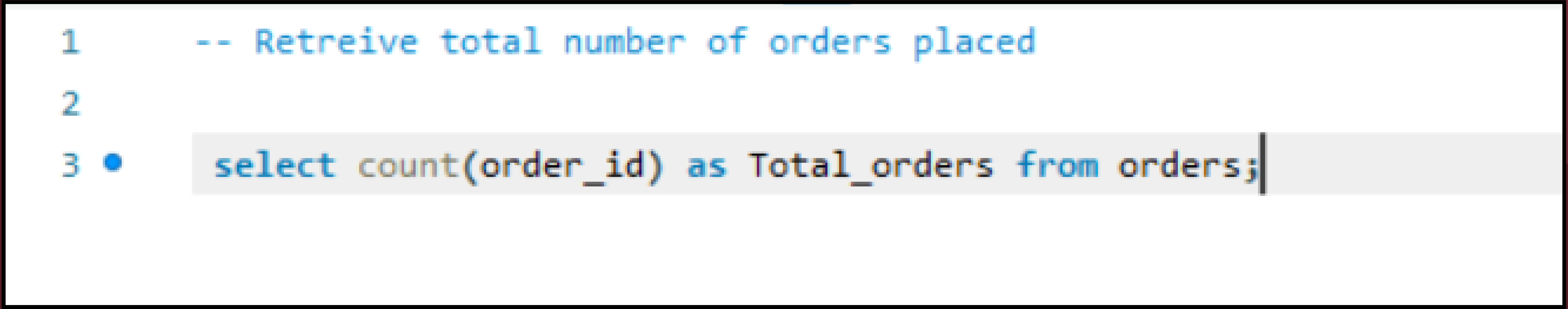
HELLO!

I am Kaushik Kamat, a fourth-year B.Tech (CSE) student. This is a project where I have applied SQL to address different queries related to pizza sales from PIZZA HUT , leading to valuable insights.

INTRODUCTION

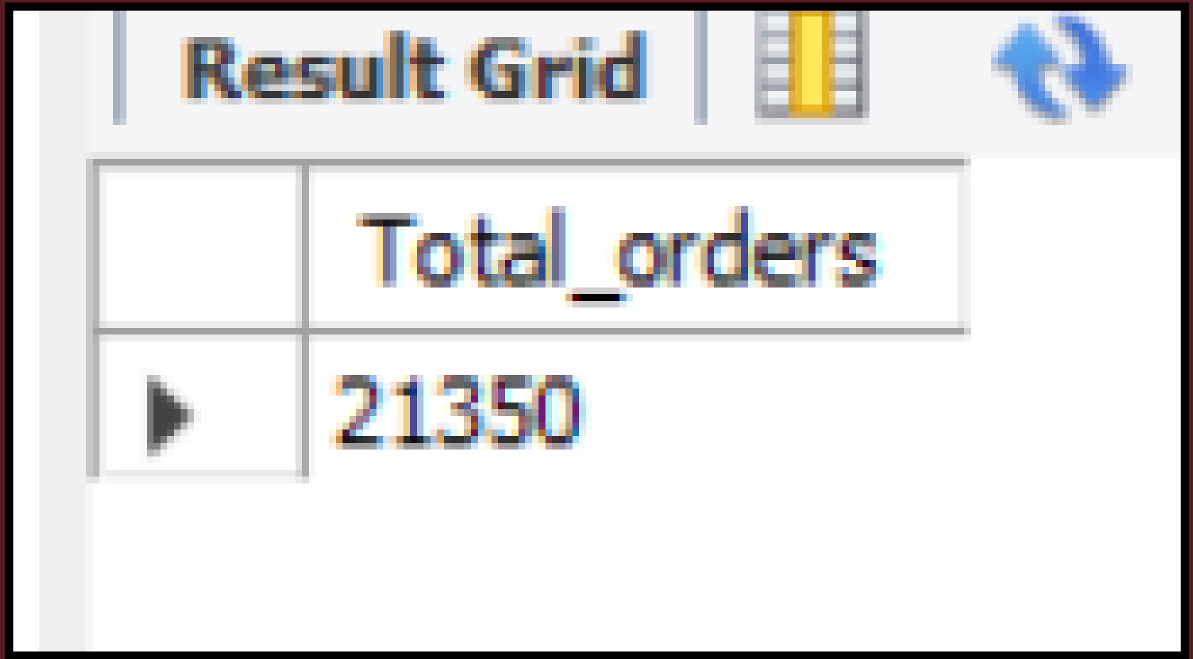
Welcome to our presentation on Using SQL Queries to Solve Pizza Sales Queries. In today's data-driven world, businesses rely heavily on data analysis to make informed decisions. For a pizza restaurant, understanding sales trends, customer preferences, and inventory needs is crucial for optimizing operations and maximizing profits. In this presentation, we will explore how SQL (Structured Query Language) can be leveraged to analyze and solve various sales-related queries in a pizza business

1:RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



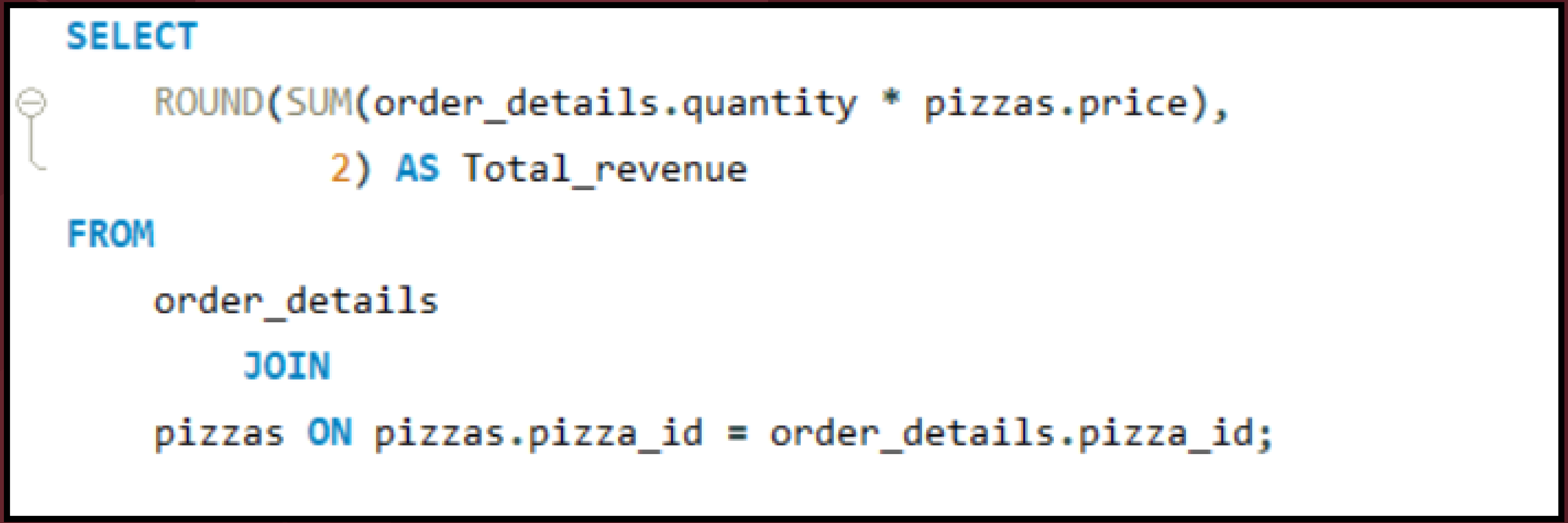
```sql
-- Retreive total number of orders placed

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

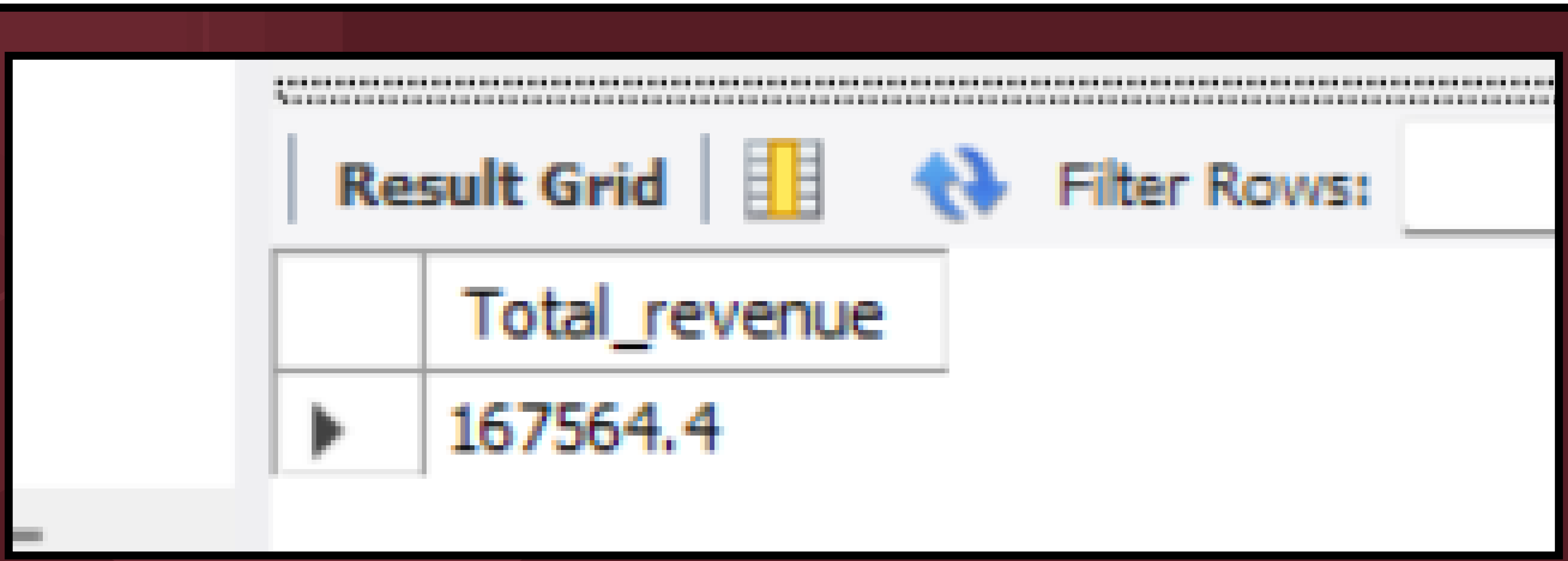


Total_orders
21350

2:CALCULATE THE TOTAL REVENUE GENREATED FROM PIZZA SALES

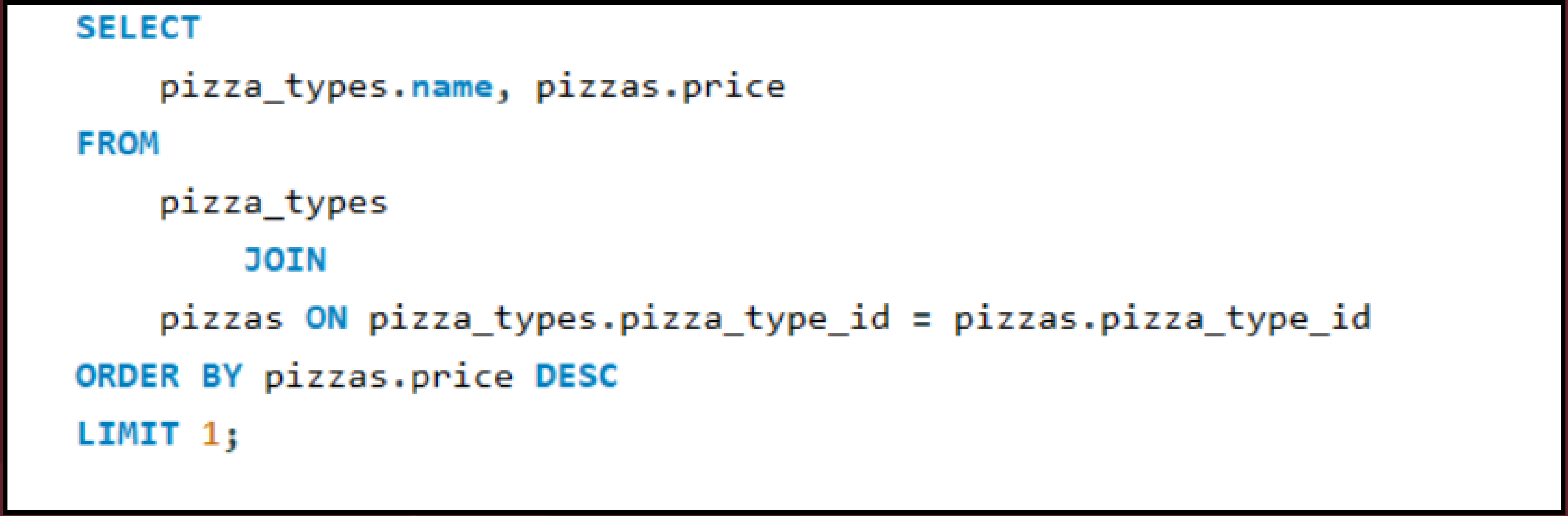


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

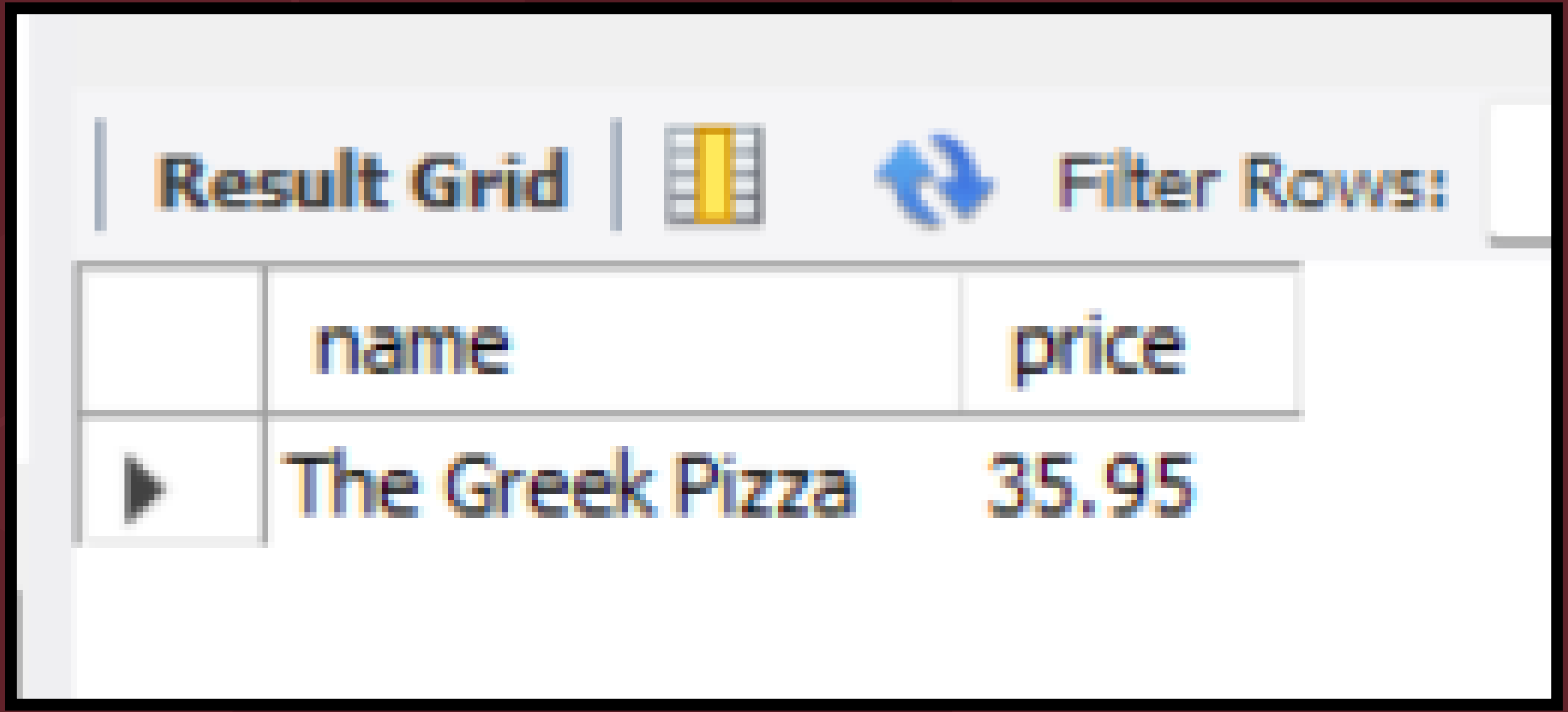


```
Total_revenue
167564.4
```

3:IDENTIFY THE HIGHEST PRICE PIZZA

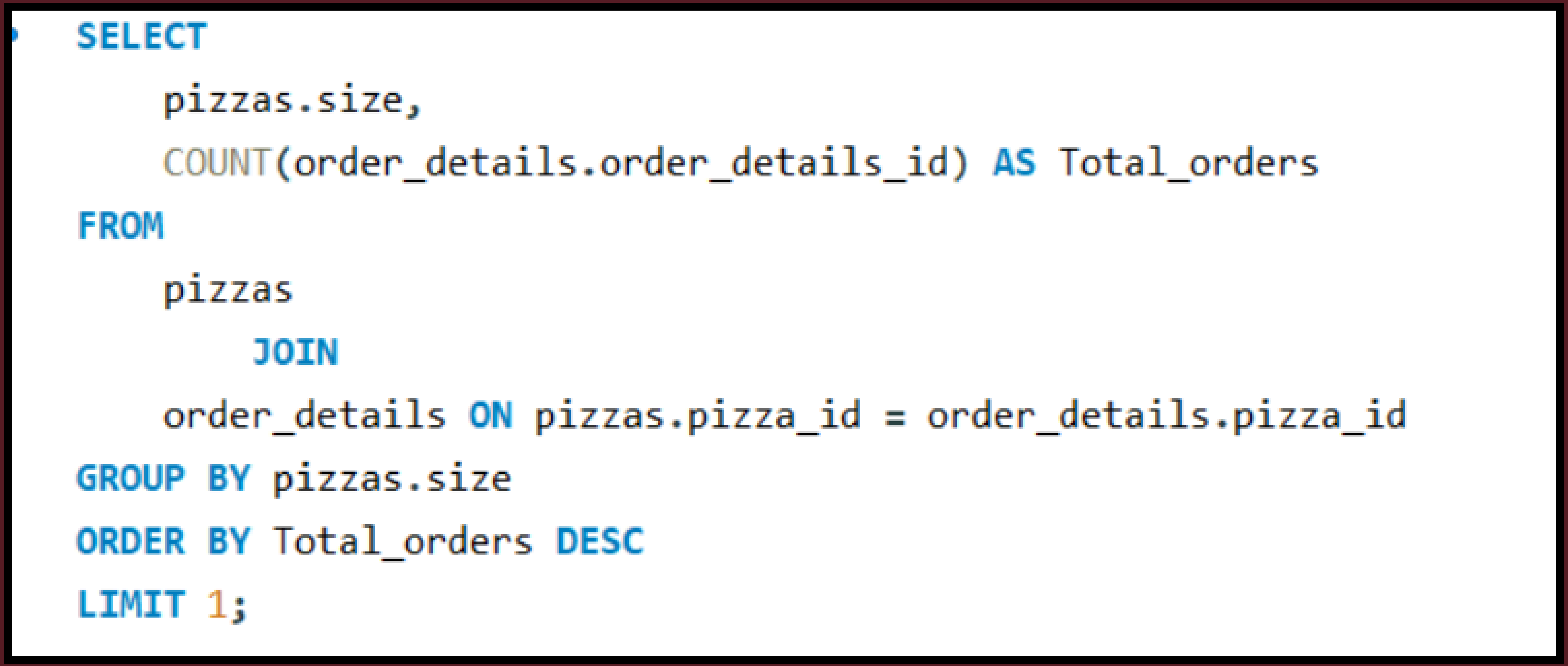


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

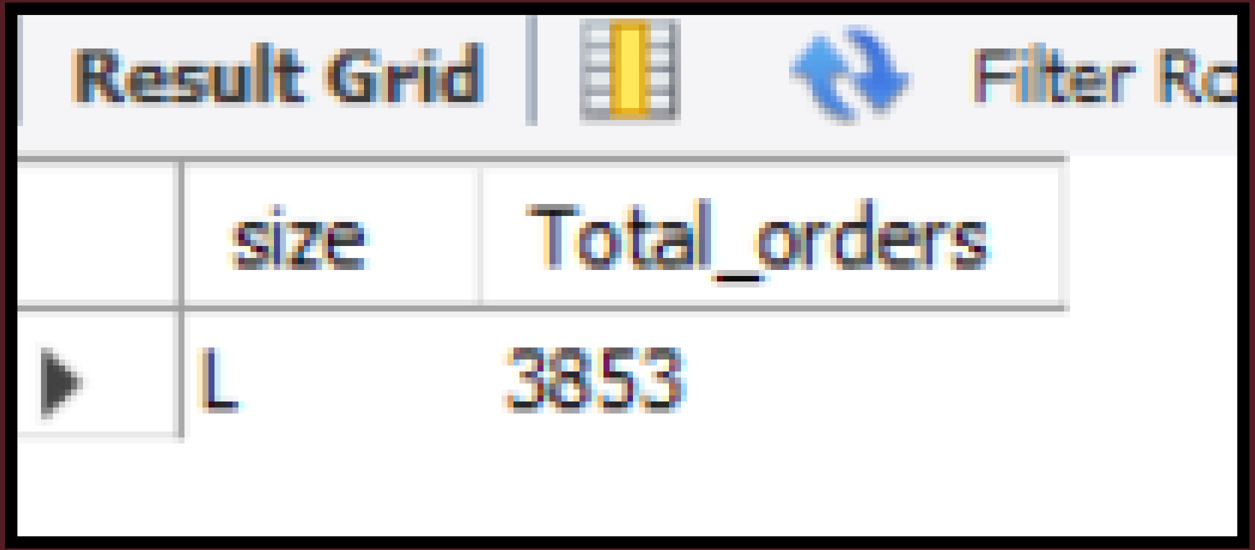


name	price
The Greek Pizza	35.95

4:IDENTIFY MOST COMMON PIZZA SIZE ORDERS

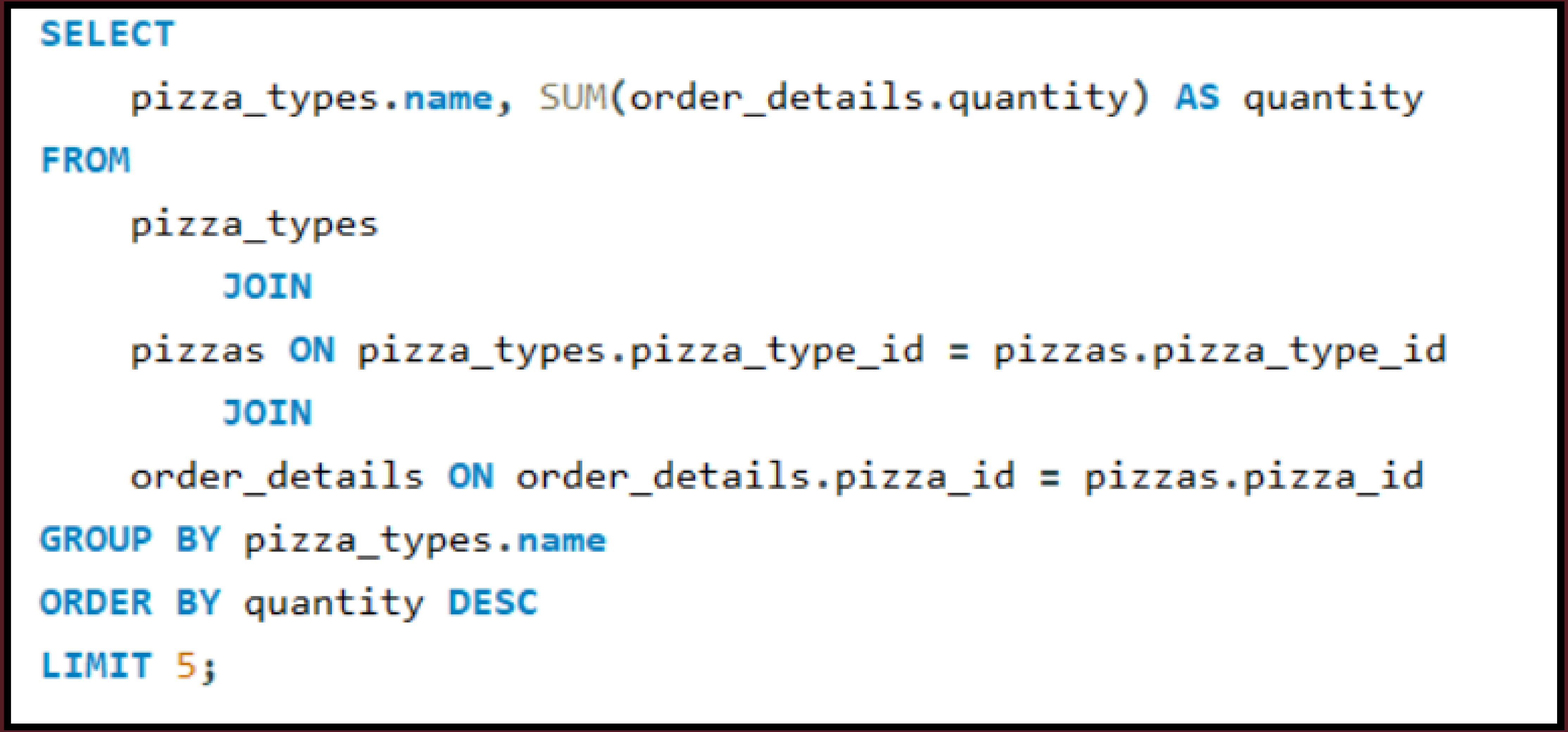


```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Total_orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_orders DESC
LIMIT 1;
```

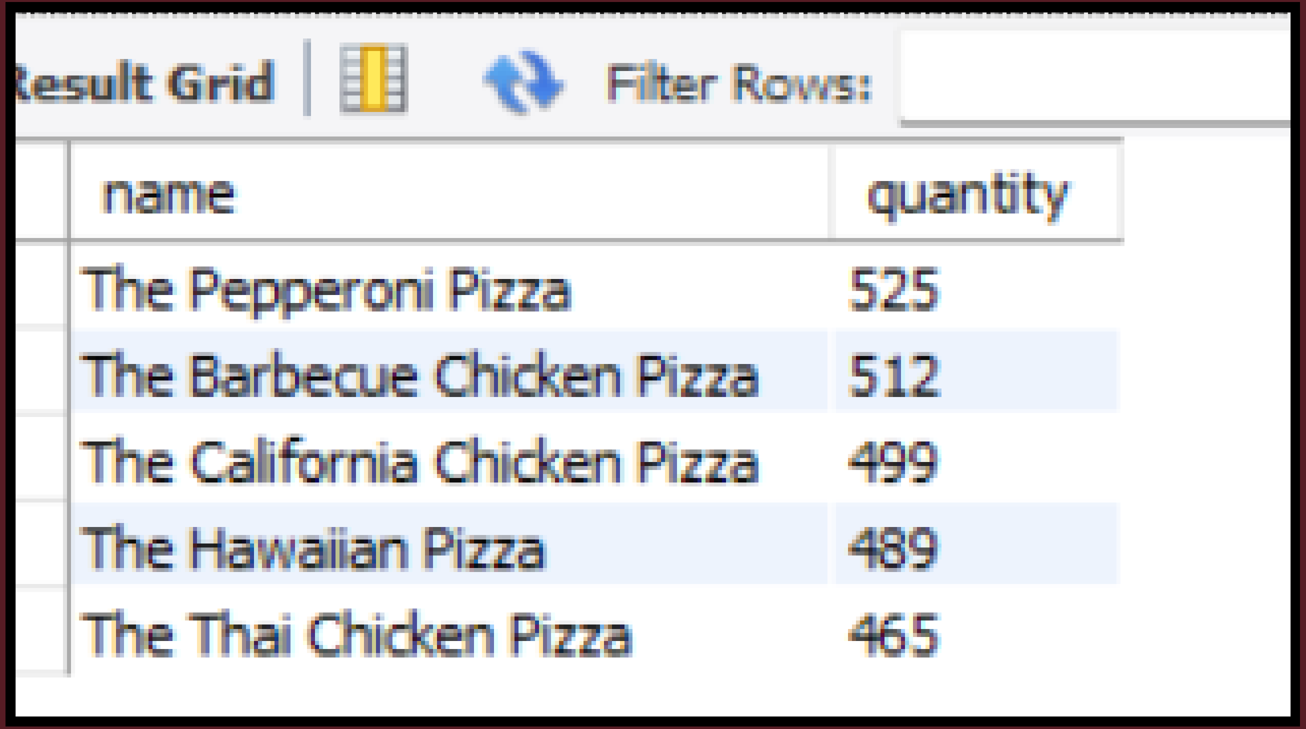


```
size   Total_orders
L      3853
```

5:LIST THE TOP '5' MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITITIES

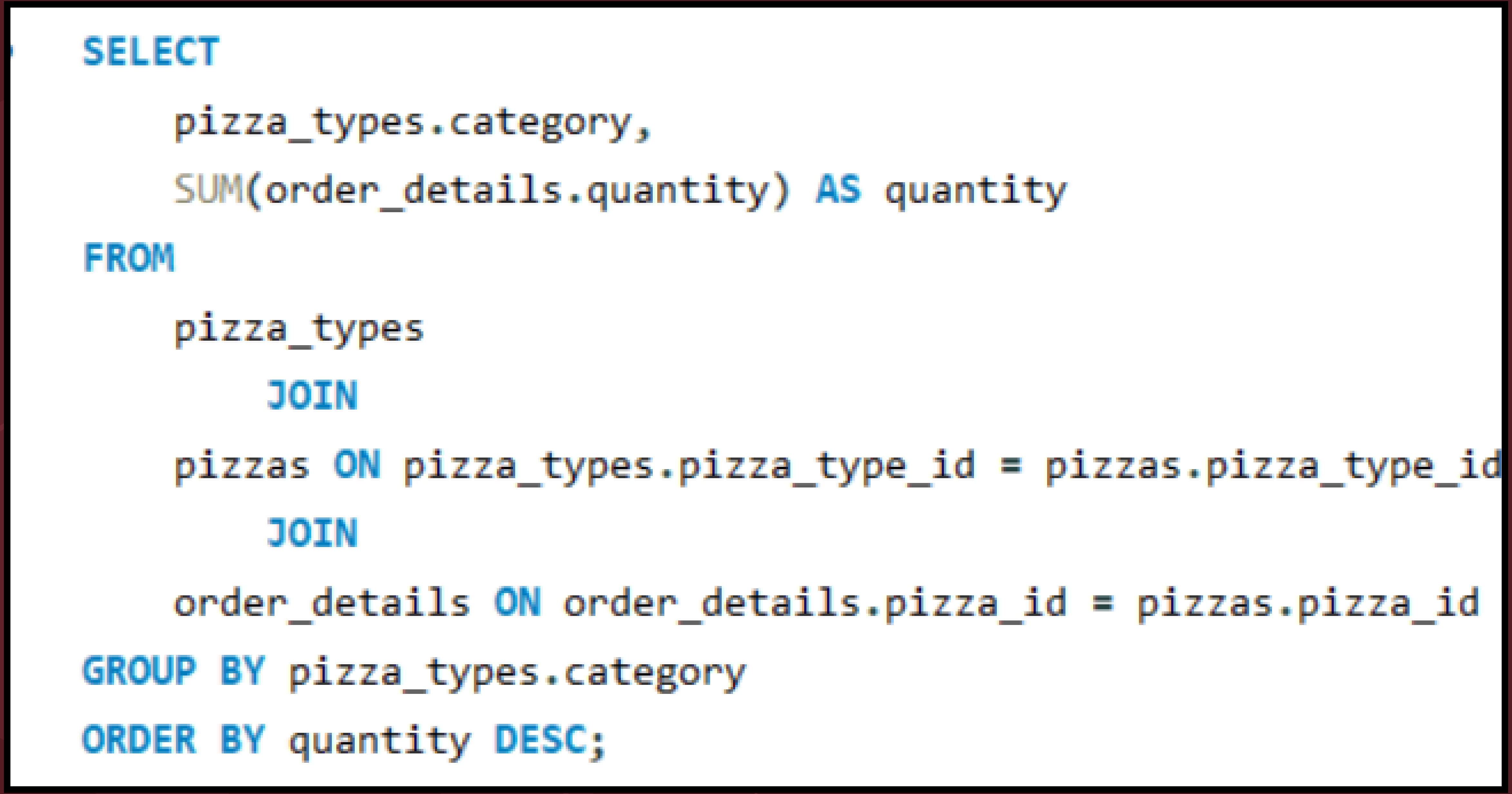


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

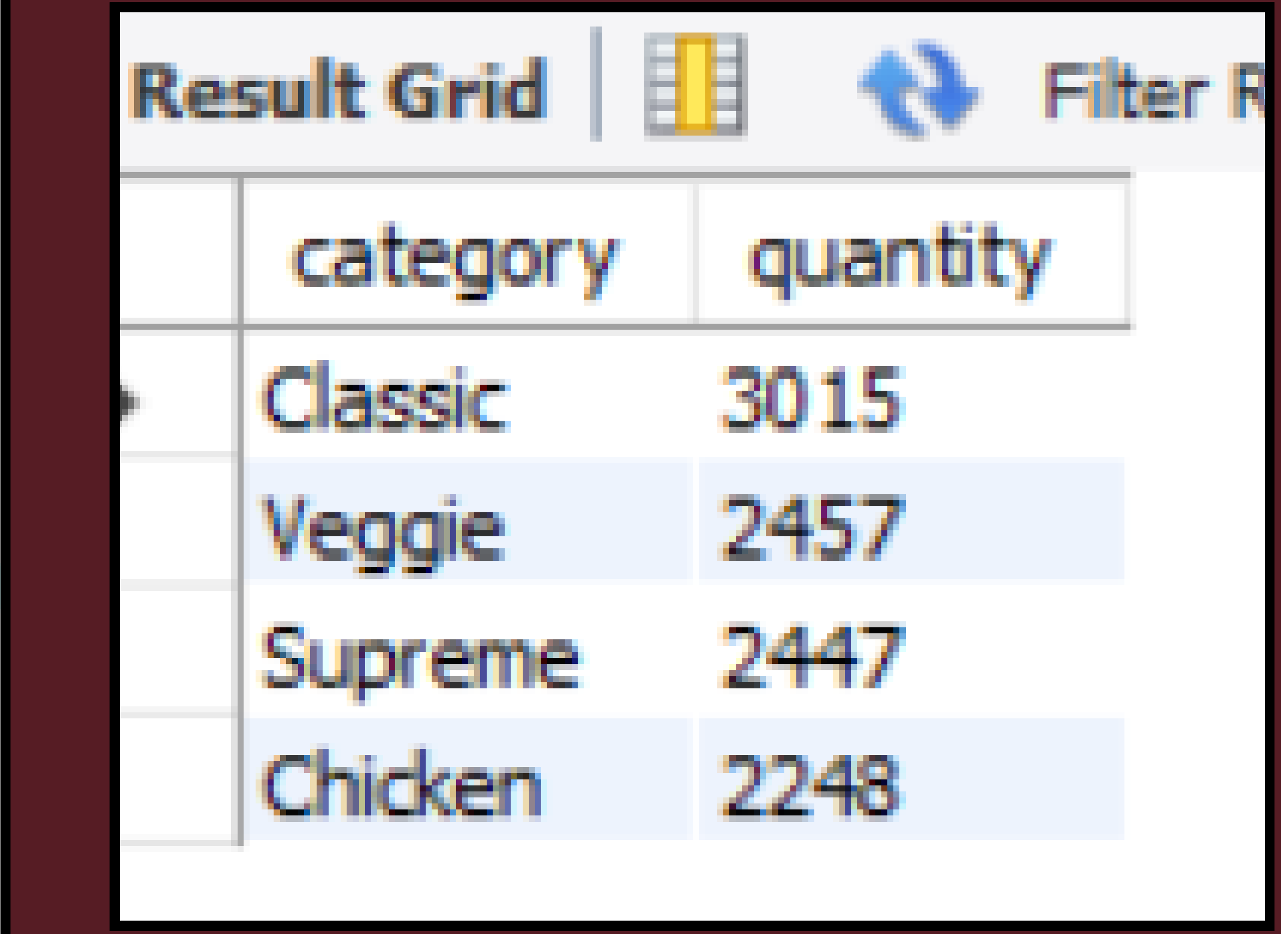


name	quantity
The Pepperoni Pizza	525
The Barbecue Chicken Pizza	512
The California Chicken Pizza	499
The Hawaiian Pizza	489
The Thai Chicken Pizza	465

6: FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

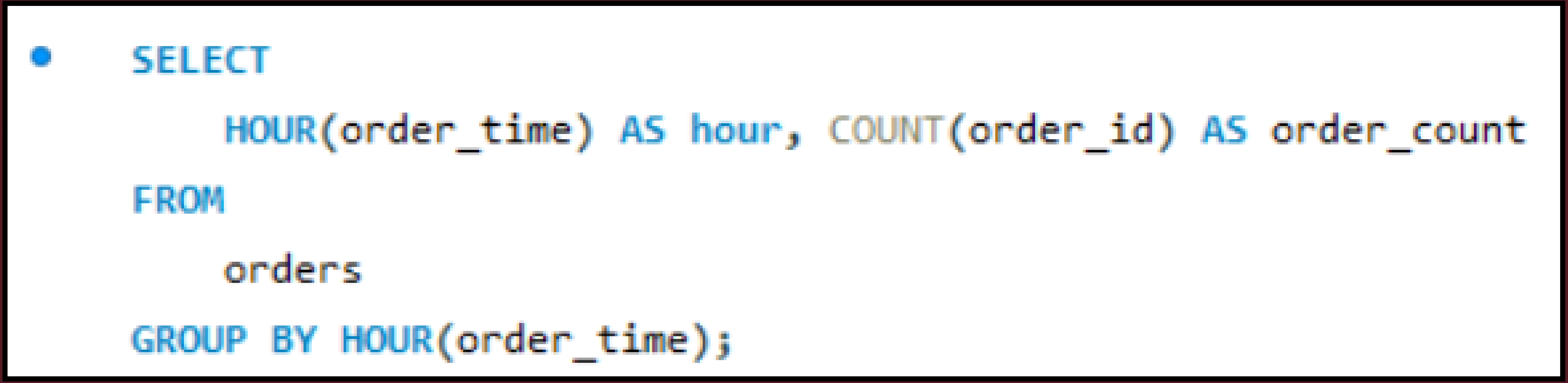


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

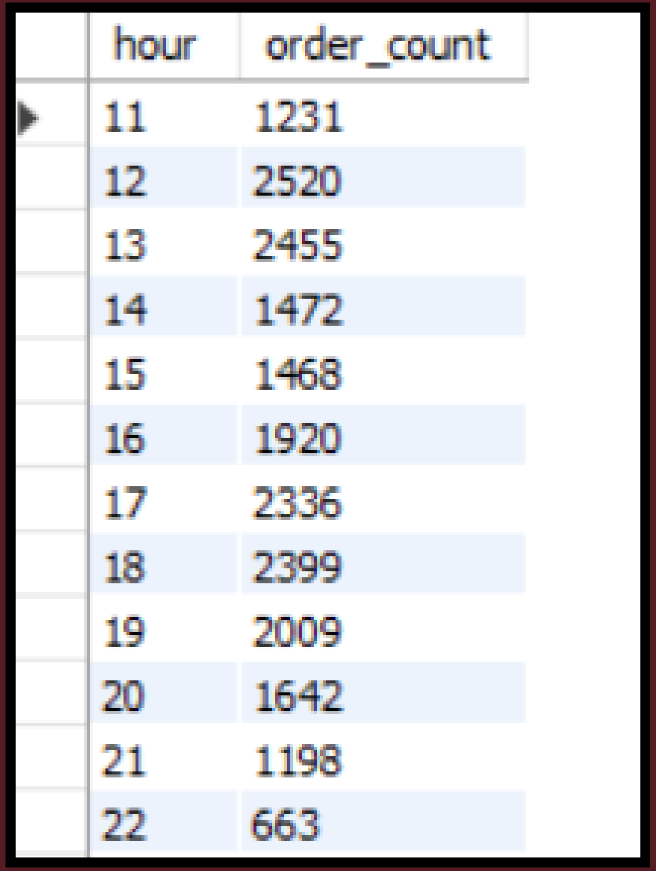


category	quantity
Classic	3015
Veggie	2457
Supreme	2447
Chicken	2248

7:DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

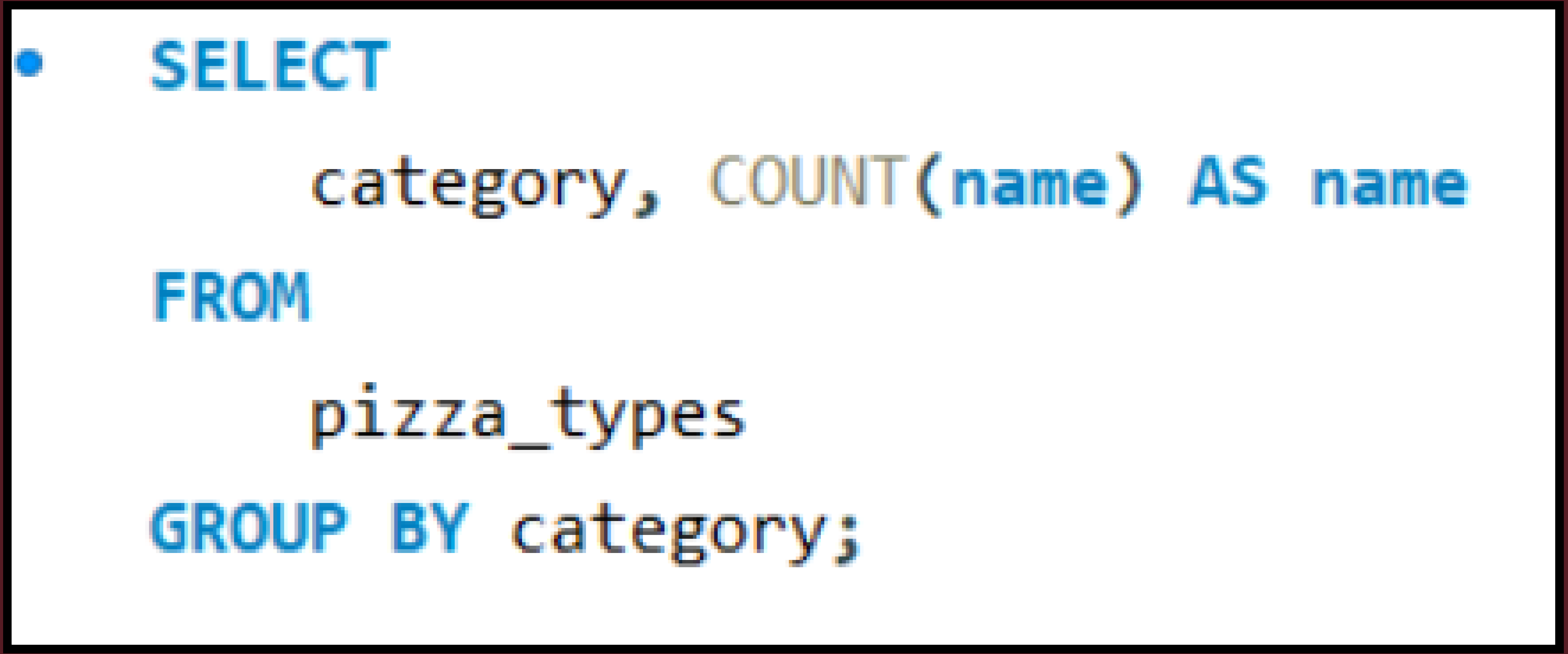


```
● SELECT
      HOUR(order_time) AS hour, COUNT(order_id) AS order_count
  FROM
      orders
  GROUP BY HOUR(order_time);
```

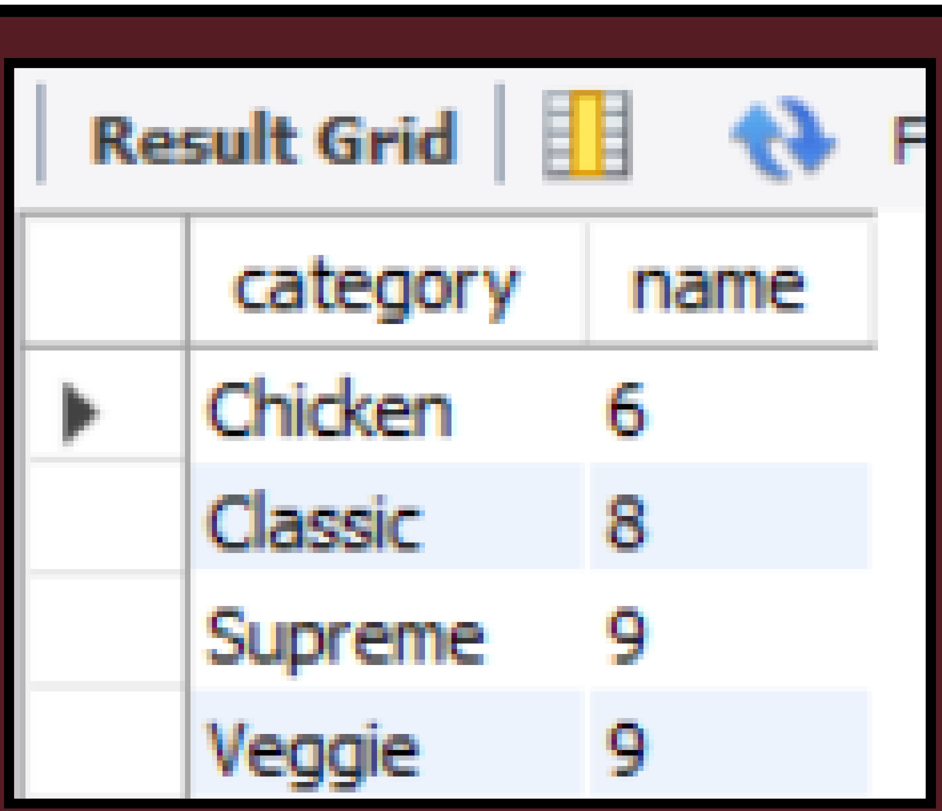


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

8:FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

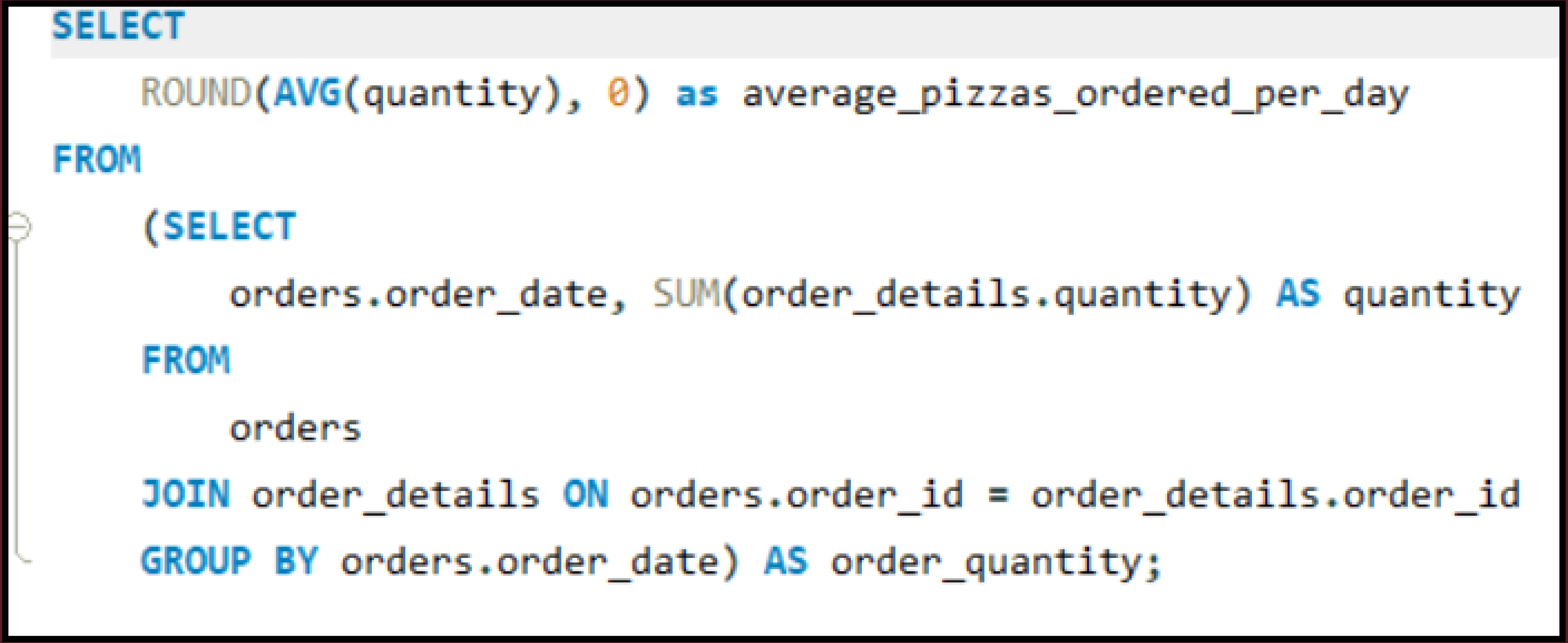


```sql
SELECT
    category, COUNT(name) AS name
FROM
    pizza_types
GROUP BY category;
```

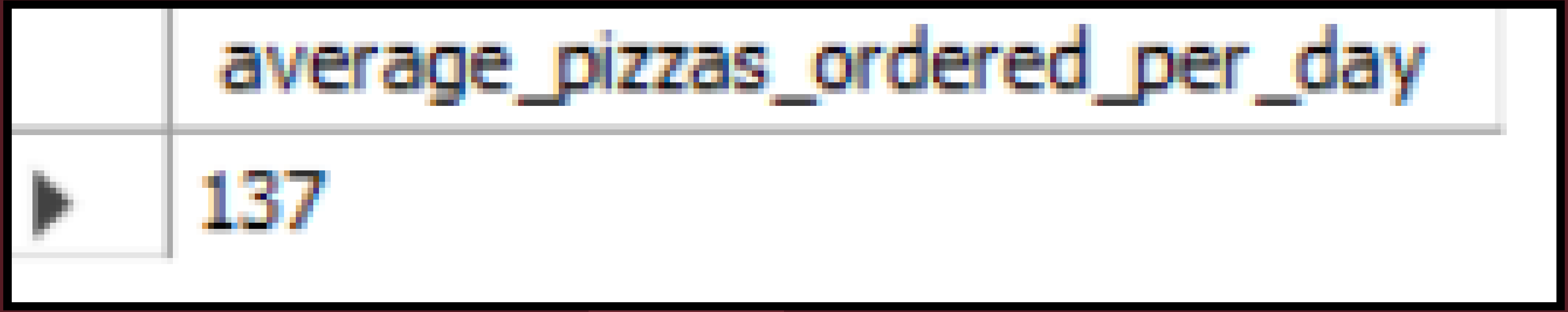


	category	name
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

9:GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

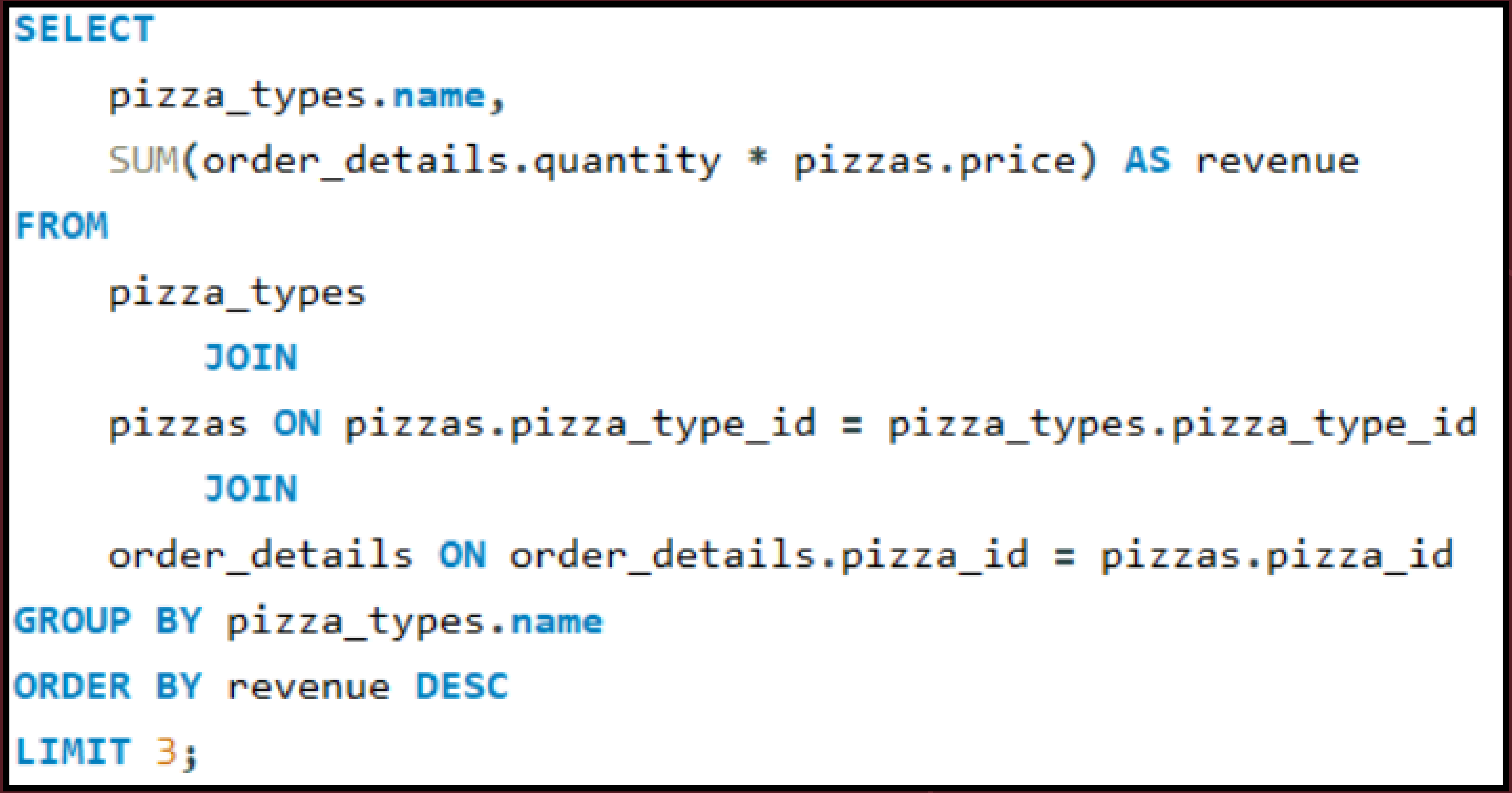


```sql
SELECT
    ROUND(AVG(quantity), 0) as average_pizzas_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;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

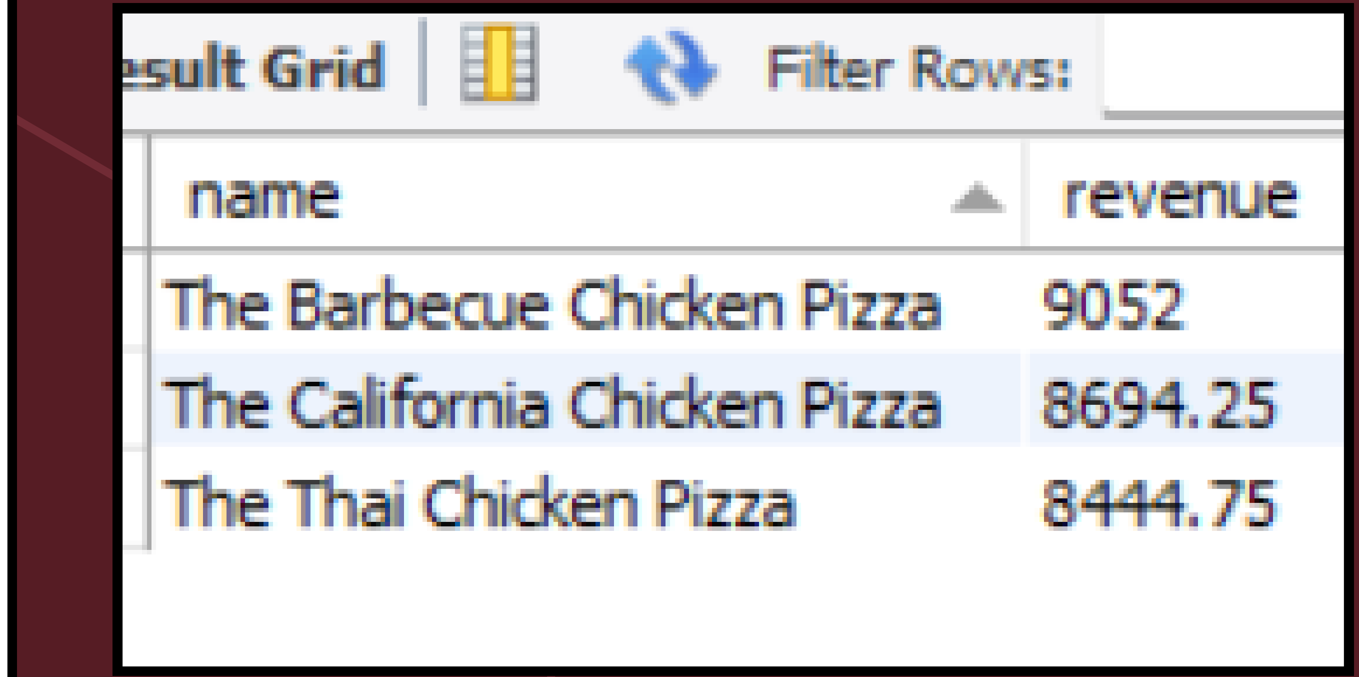


average_pizzas_ordered_per_day
137

10:DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

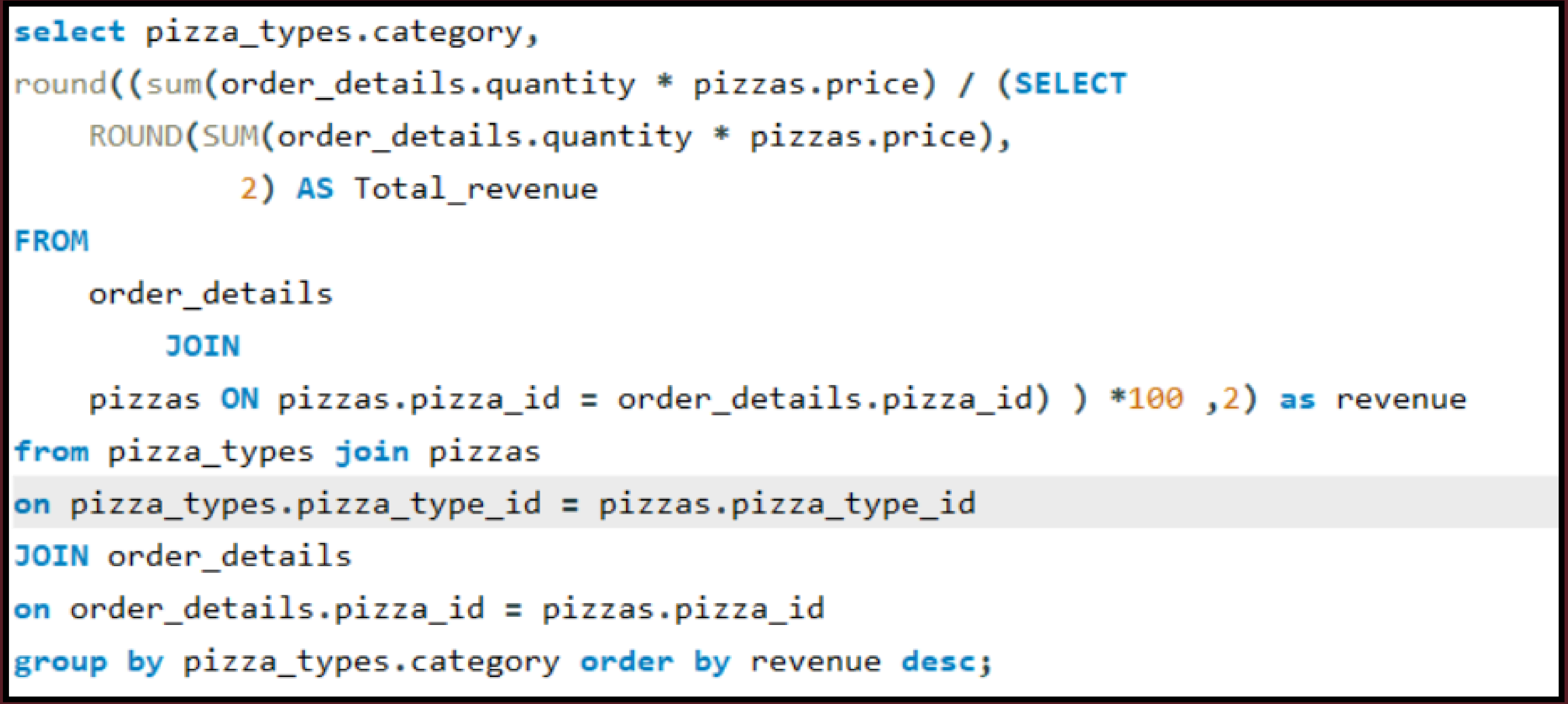


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

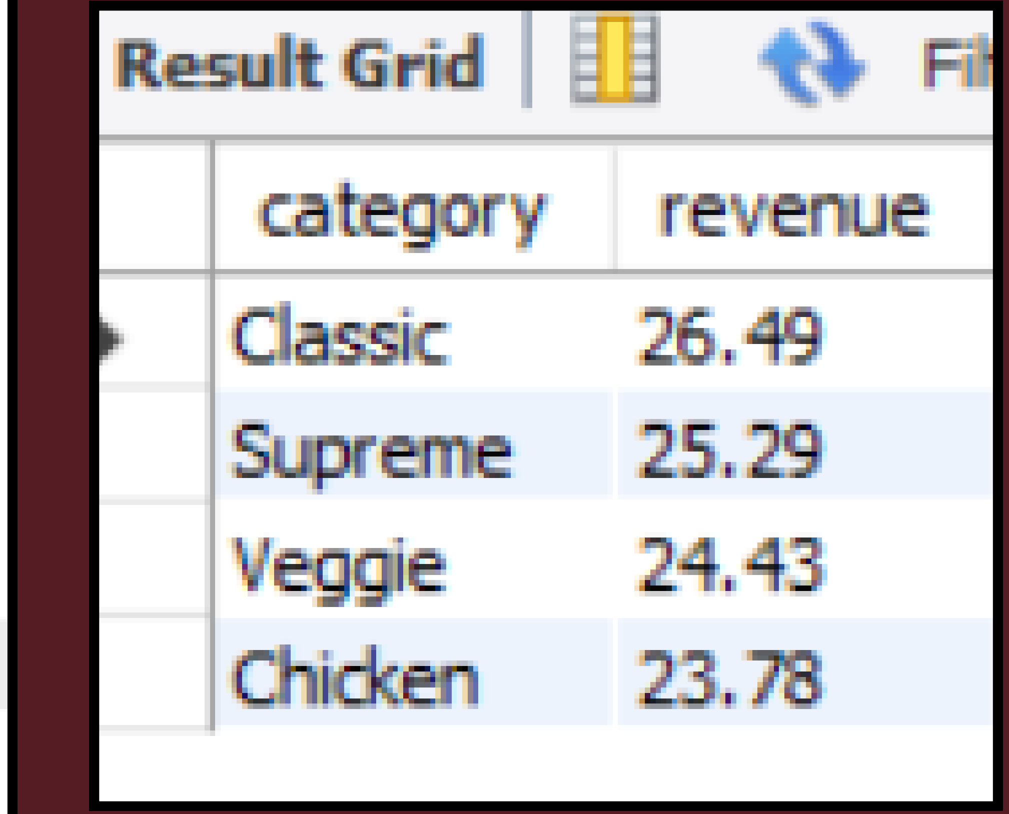


name	revenue
The Barbecue Chicken Pizza	9052
The California Chicken Pizza	8694.25
The Thai Chicken Pizza	8444.75

11: CACULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

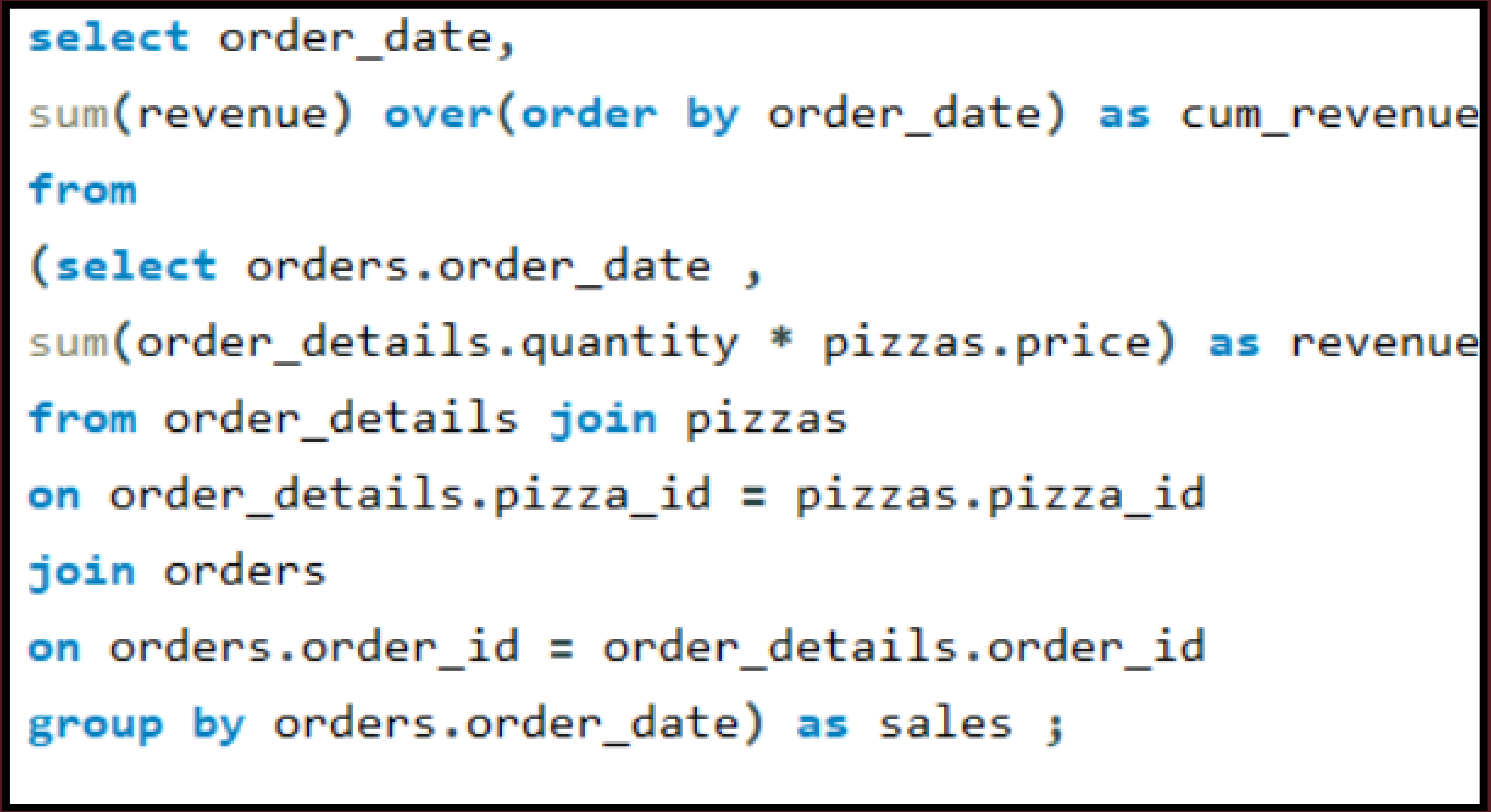


```
select pizza_types.category,
round((sum(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) ) *100 ,2) as revenue
from pizza_types 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;
```

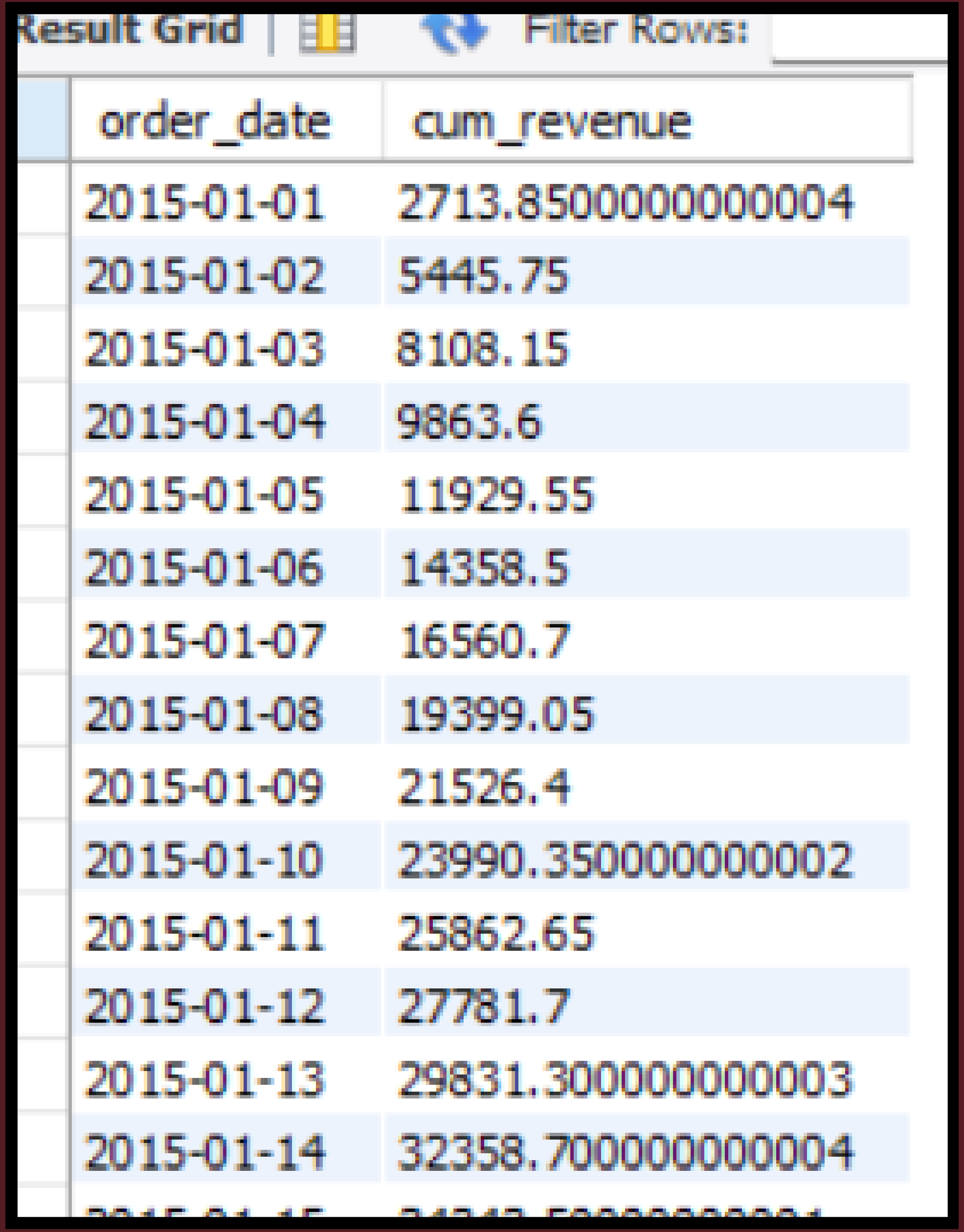


category	revenue
Classic	26.49
Supreme	25.29
Veggie	24.43
Chicken	23.78

12:ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

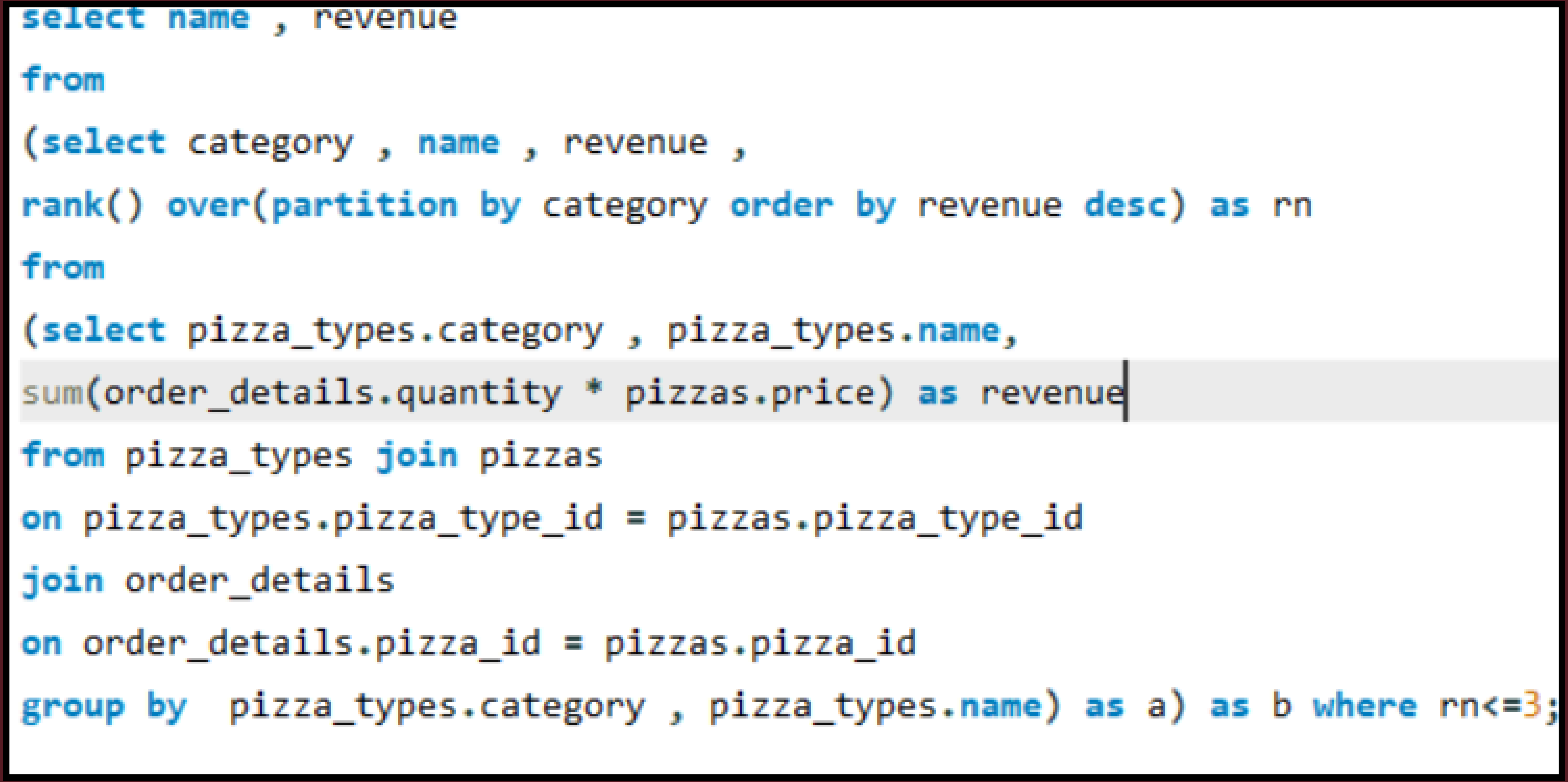


```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date ,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales ;
```

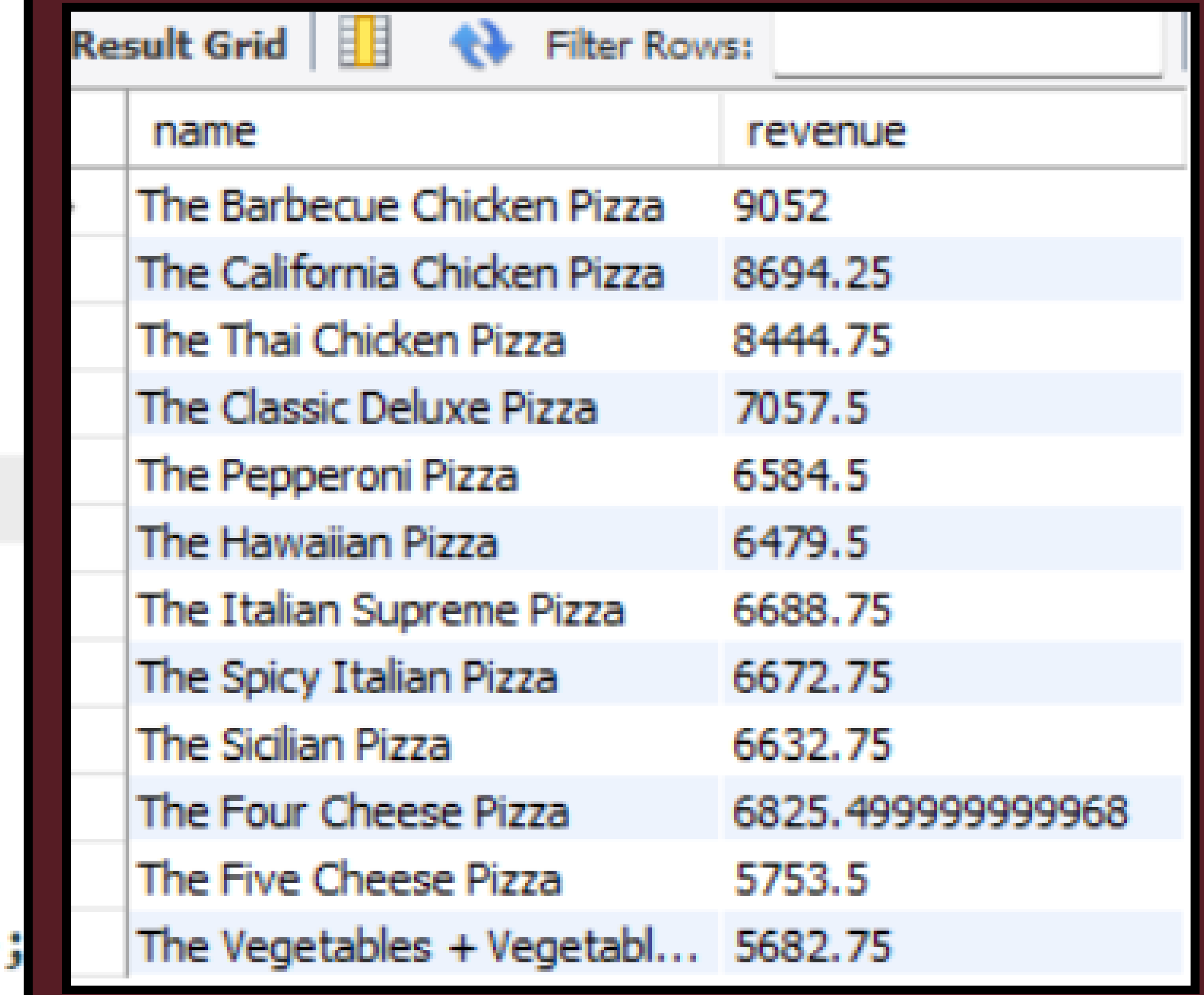


order_date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

13:TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY



```
select name , revenue
from
(select category , name , revenue ,
rank() over(partition by category order by revenue desc) as rn
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 rn<=3;
```



name	revenue
The Barbecue Chicken Pizza	9052
The California Chicken Pizza	8694.25
The Thai Chicken Pizza	8444.75
The Classic Deluxe Pizza	7057.5
The Pepperoni Pizza	6584.5
The Hawaiian Pizza	6479.5
The Italian Supreme Pizza	6688.75
The Spicy Italian Pizza	6672.75
The Sicilian Pizza	6632.75
The Four Cheese Pizza	6825.499999999968
The Five Cheese Pizza	5753.5
The Vegetables + Vegetabl...	5682.75

THANK YOU !