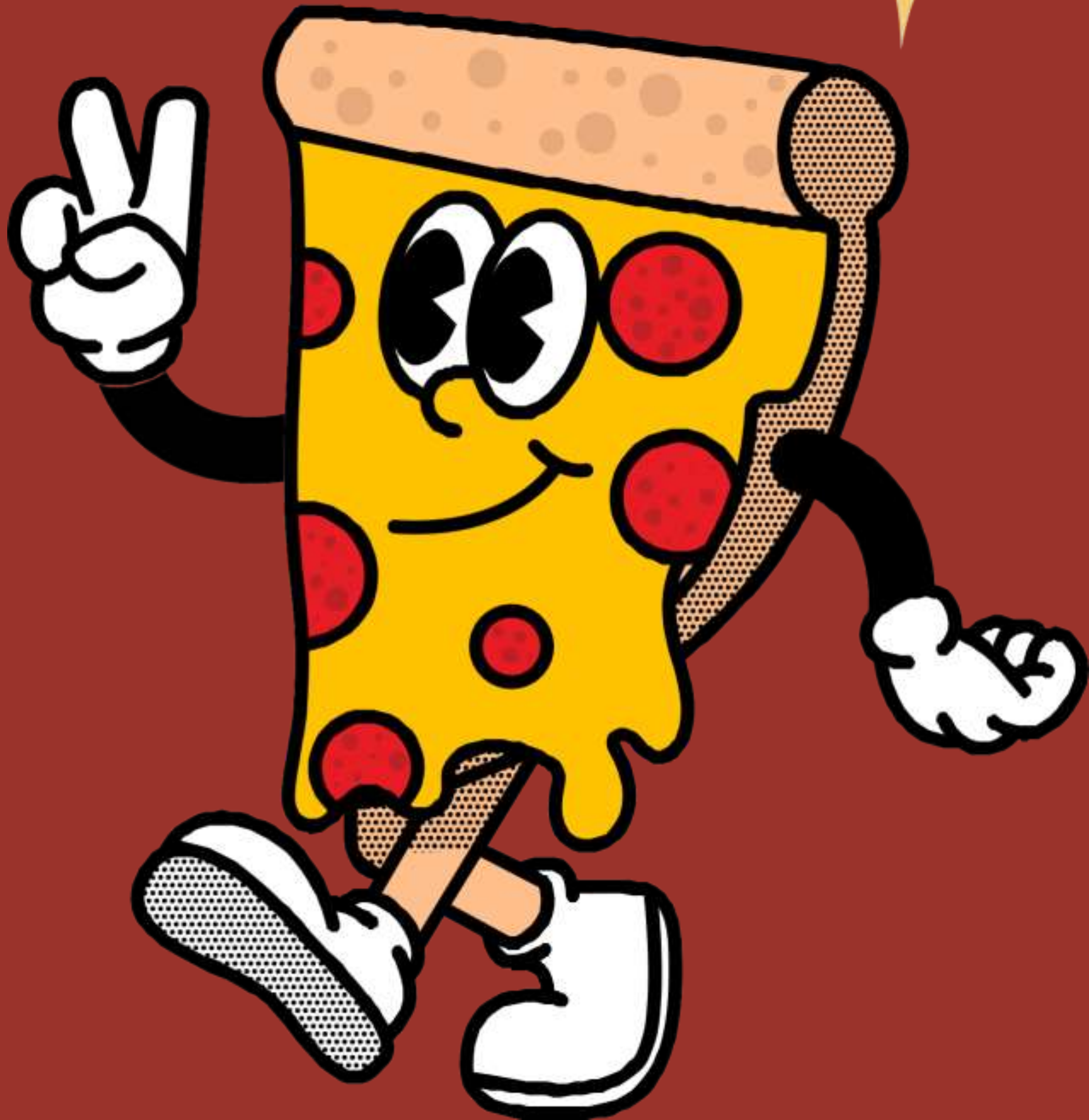
PIZZA SALES ANALYSIS

Using SQL

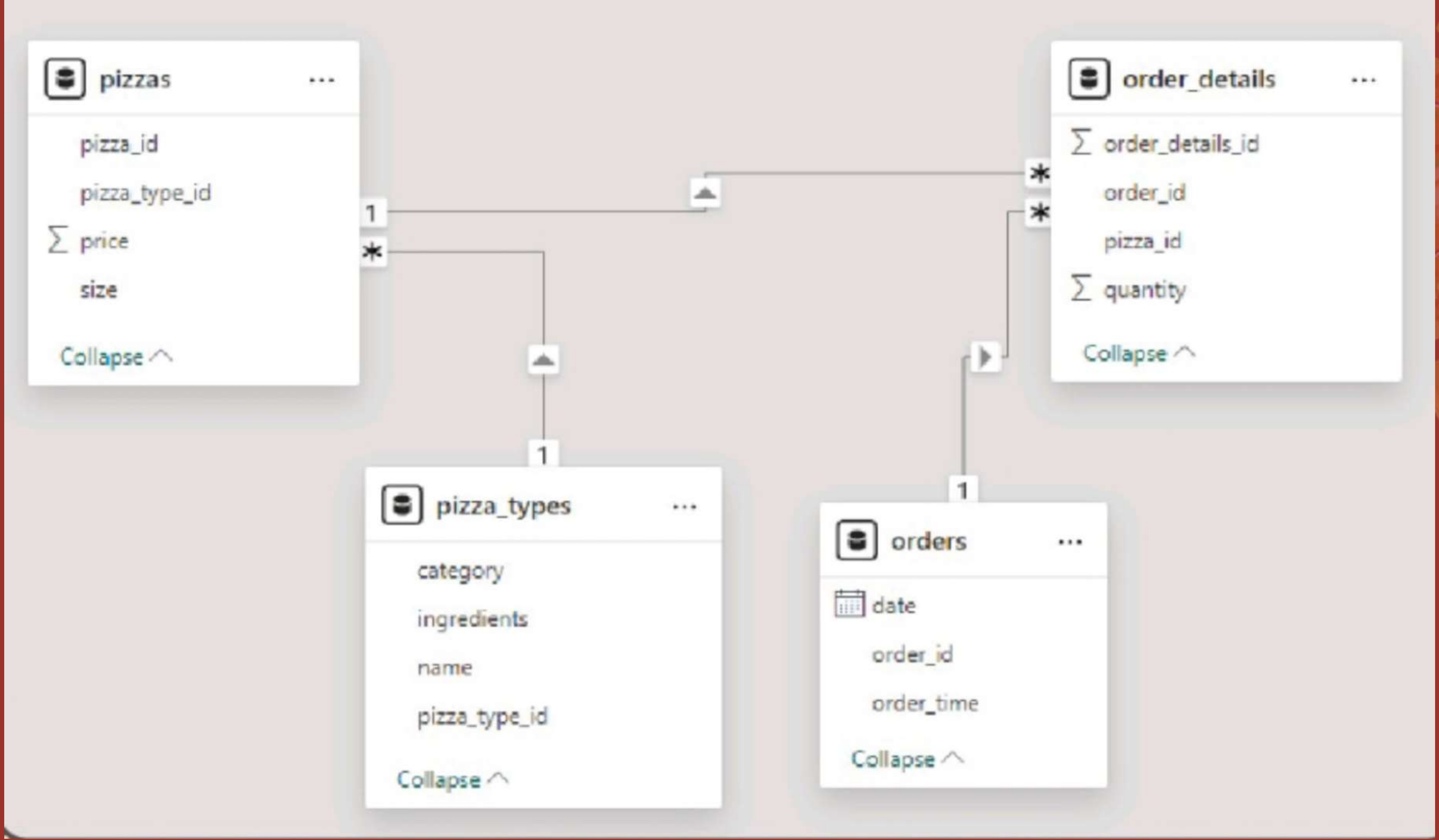


By:**DIVYADHARSHINI**

HELLO EVERYONE!

My name is Divyadharshini Singh and I developed an SQL project utilizing MySQL database software to conduct a comprehensive analysis of pizza sales data. The project involved working with four distinct datasets to extract key insights into various aspects of sales performance. factors driving success in the pizza industry . By addressing 11 critical questions, I was able to identify trends in sales, analyze customer behavior, and assess overall business performance, providing a deeper understanding of the driving success in the pizza industry.

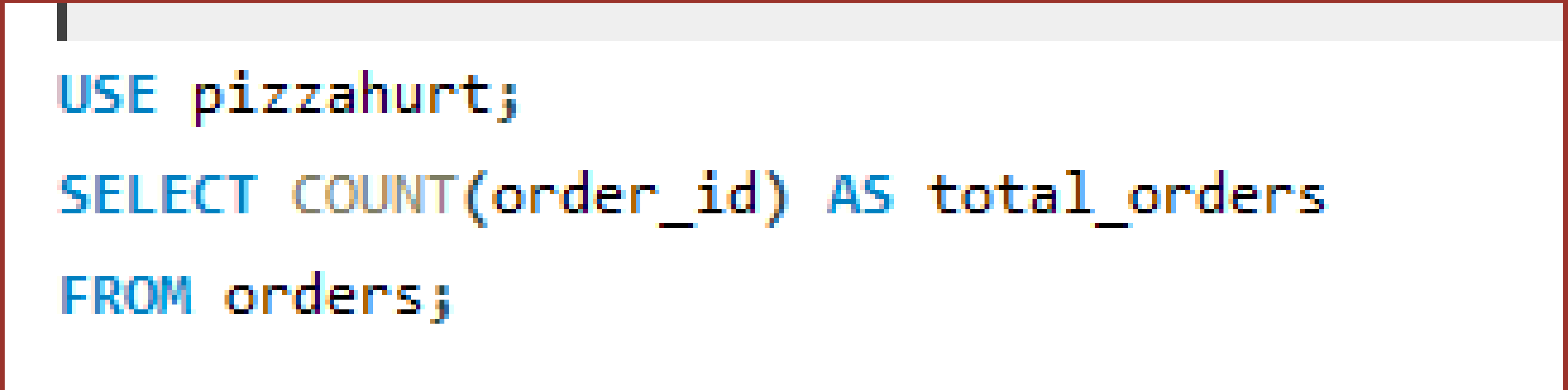
“SCHEMA”


pizzas
pizza_id
pizza_type_id
∑ price
size
Collapse
order_details
∑ order_details_id
order_id
pizza_id
∑ quantity
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
orders
date
order_id
order_time
Collapse
1
*
1
*
*
1

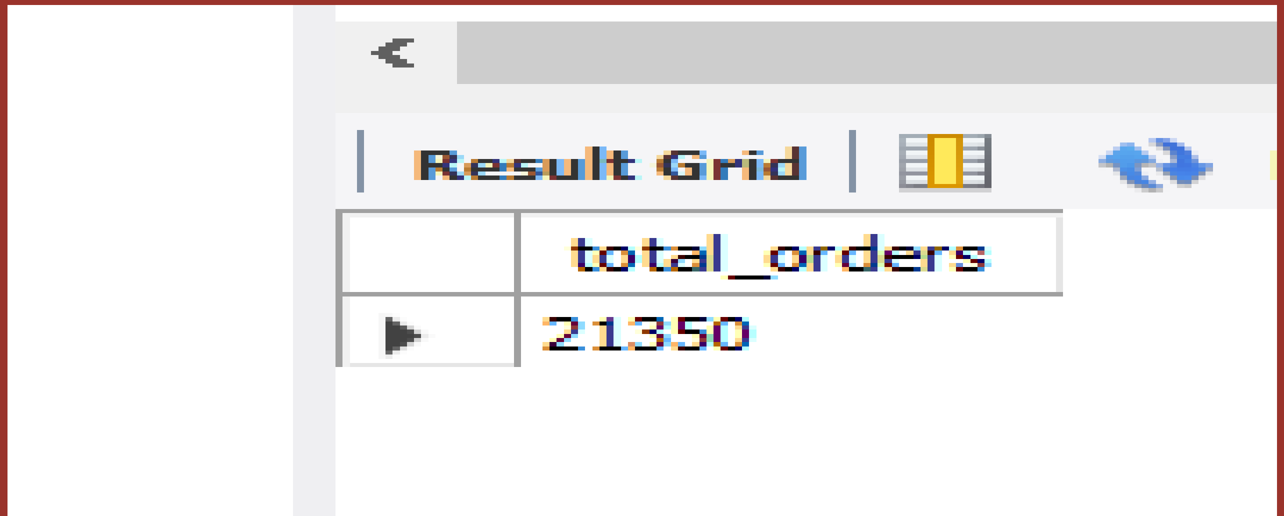
“QUESTIONS”

1.Retrieve the total number of orders placed.
2.Calculate the total revenue generated from pizza sales.
3.Identify the highest-priced pizza.
4.Identify the most common pizza size ordered.
5.List the top 5 most ordered pizza types along with their quantities.
6.Join the necessary tables to find the total quantity of each pizza category ordered.
7.Determine the distribution of orders by hour of the day.
8.Join relevant tables to find the category-wise distribution of pizzas.
9.Group the orders by date and calculate the average number of pizzas ordered per day.
10.Determine the top 3 most ordered pizza types based on revenue.
11.Calculate the percentage contribution of each pizza type to total revenue.

1. Retrieve the total number of orders placed

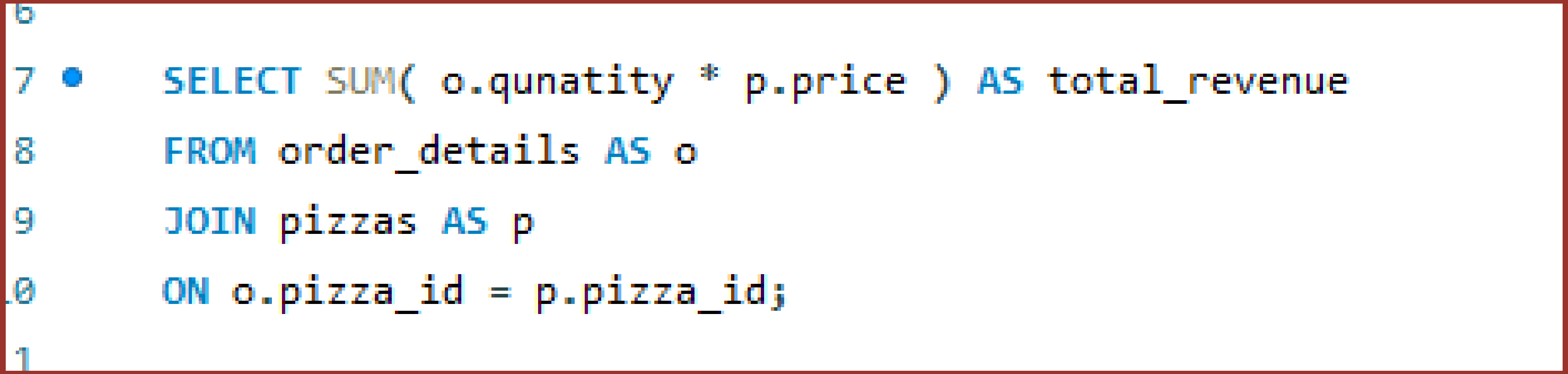


```
USE pizzahurt;
SELECT COUNT(order_id) AS total_orders
FROM orders;
```

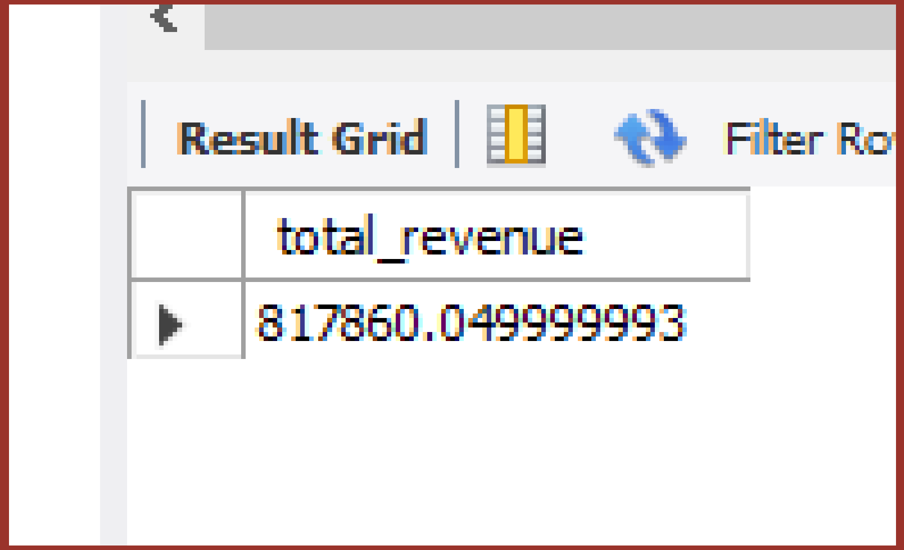


```
total_orders
21350
```

2. Calculate the total revenue generated from pizza sales.

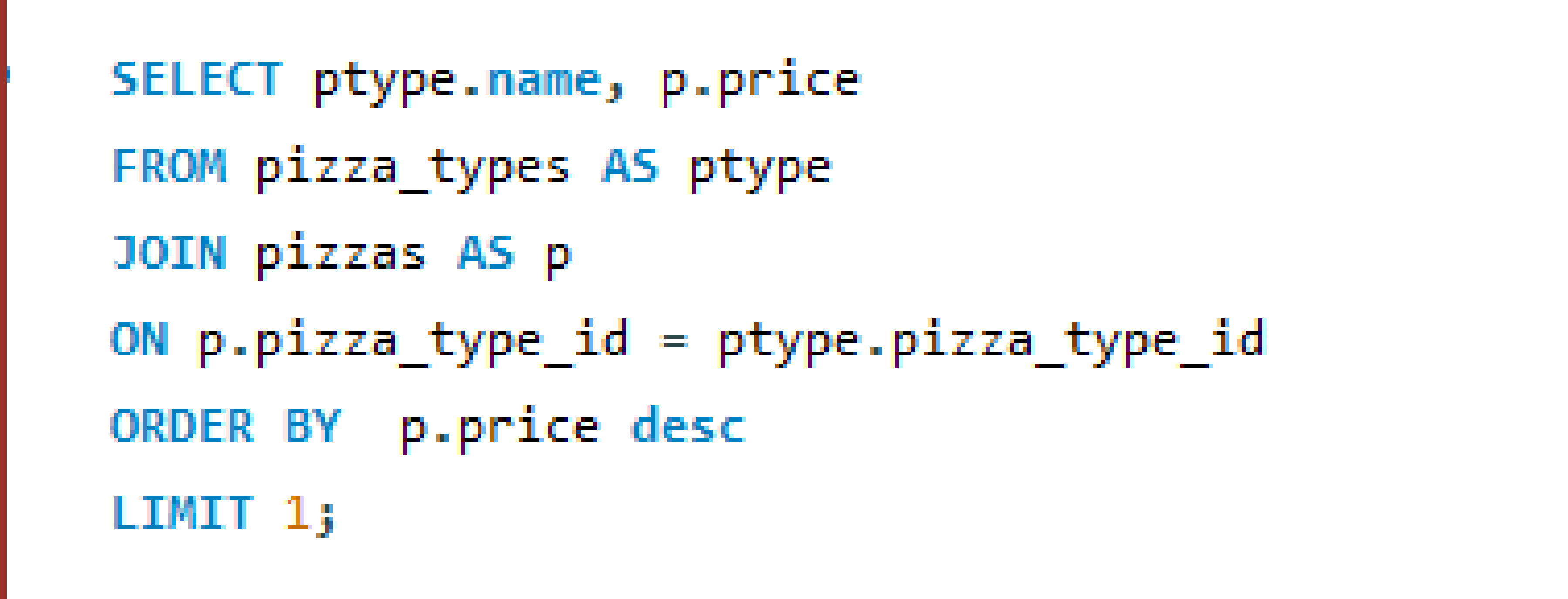


```
SELECT SUM( o.qunatity * p.price ) AS total_revenue
FROM order_details AS o
JOIN pizzas AS p
ON o.pizza_id = p.pizza_id;
```

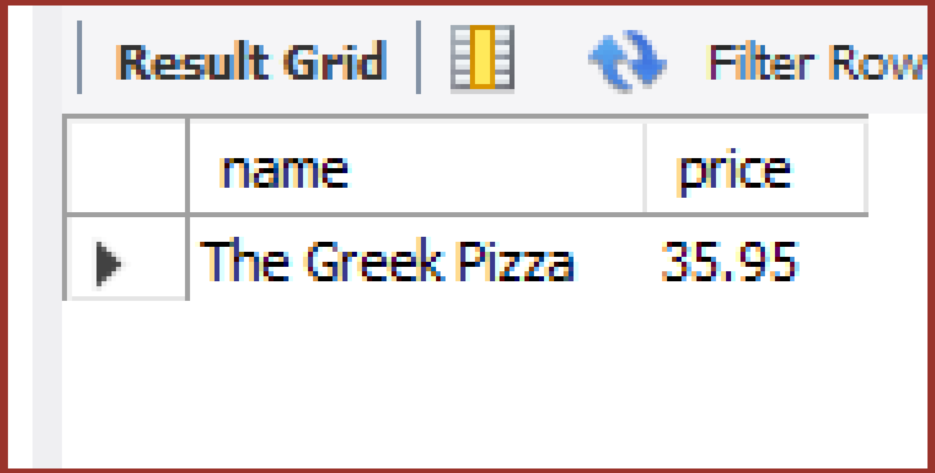


```
total_revenue
817860.049999993
```

3. Identify the highest-priced pizza.

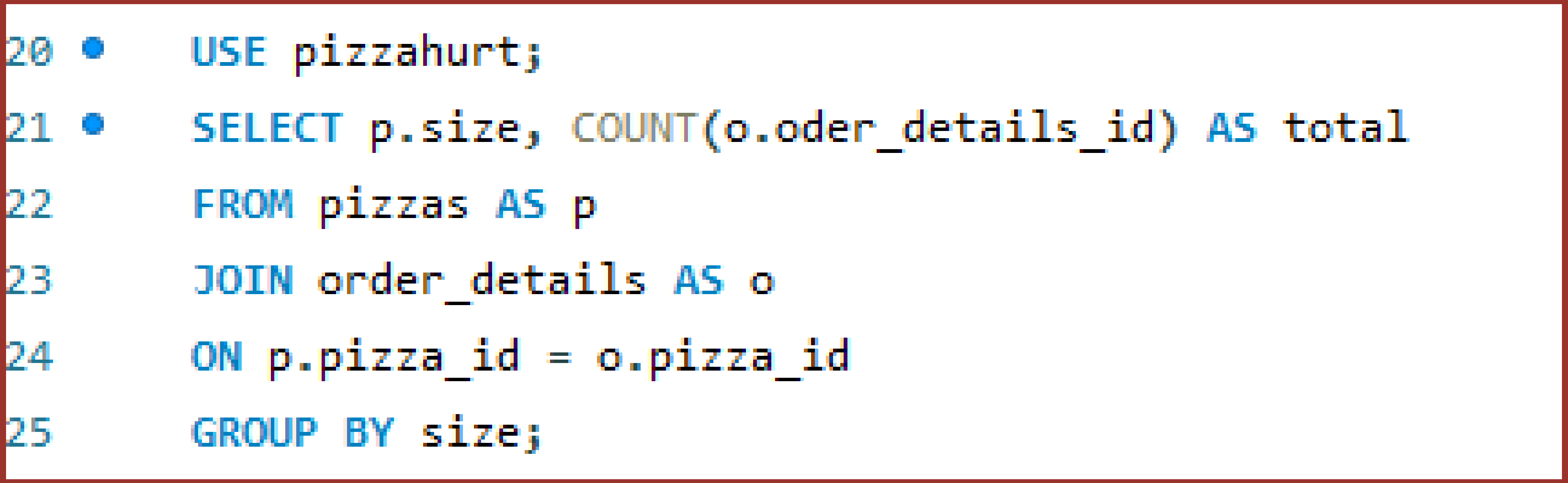


```sql
SELECT ptype.name, p.price
FROM pizza_types AS ptype
JOIN pizzas AS p
ON p.pizza_type_id = ptype.pizza_type_id
ORDER BY  p.price desc
LIMIT 1;
```

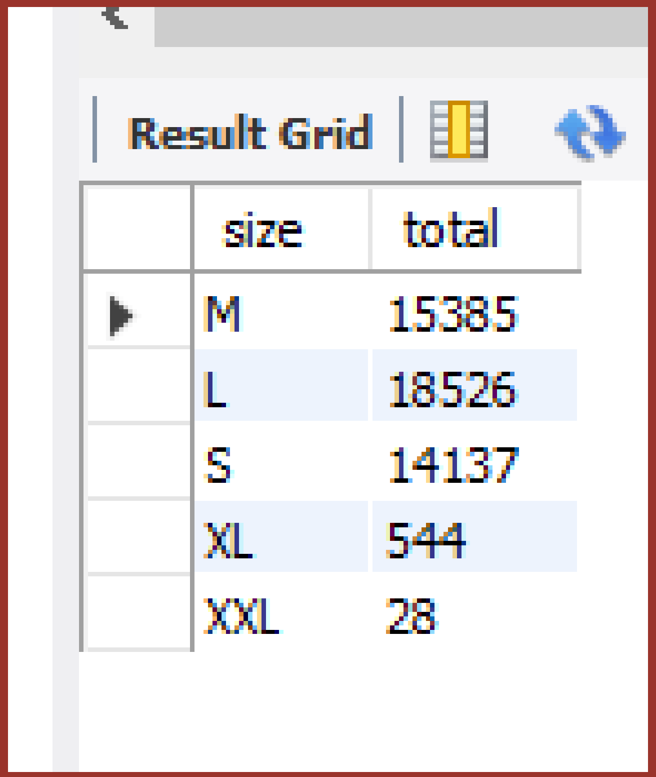


name	price
The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

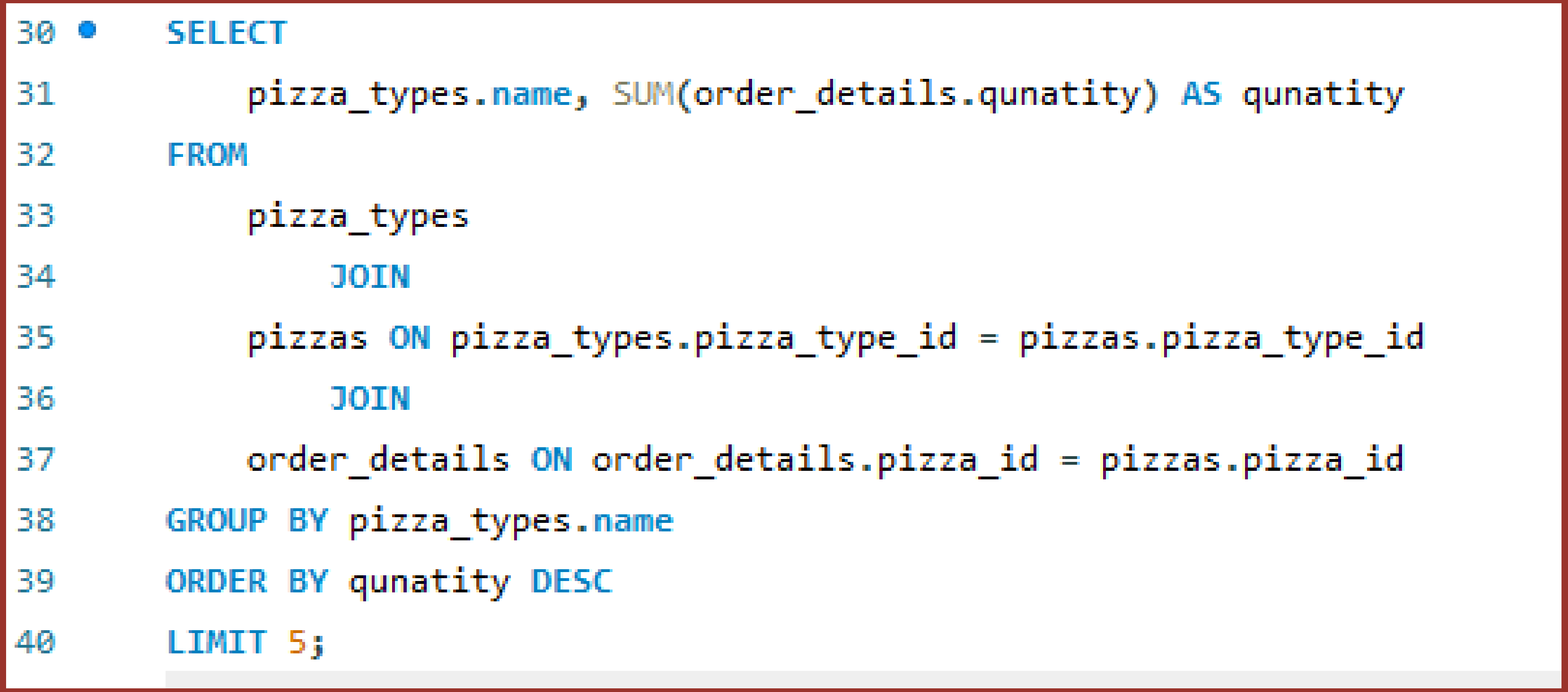


```
USE pizzahurt;
SELECT p.size, COUNT(o.oder_details_id) AS total
FROM pizzas AS p
JOIN order_details AS o
ON p.pizza_id = o.pizza_id
GROUP BY size;
```

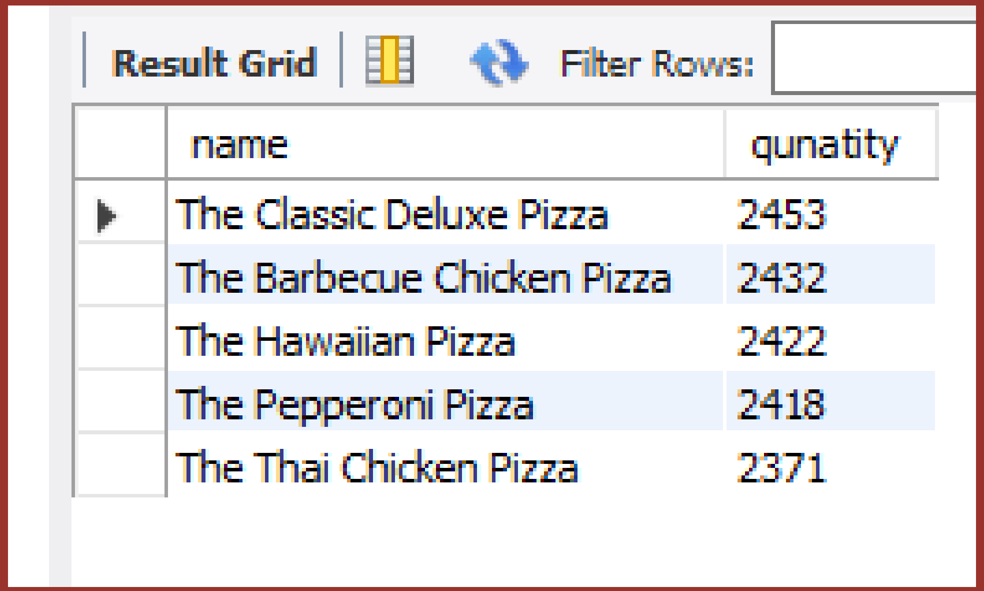


size	total
M	15385
L	18526
S	14137
XL	544
XXL	28

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

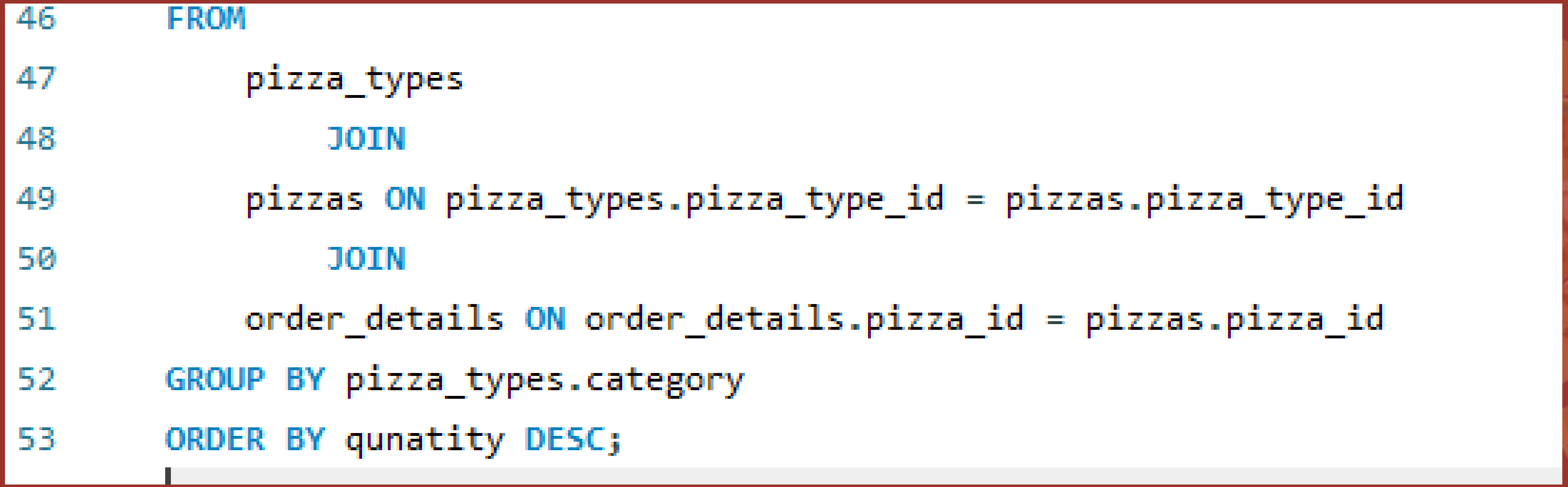


```sql
SELECT
    pizza_types.name, SUM(order_details.qunatity) AS qunatity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY qunatity DESC
LIMIT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

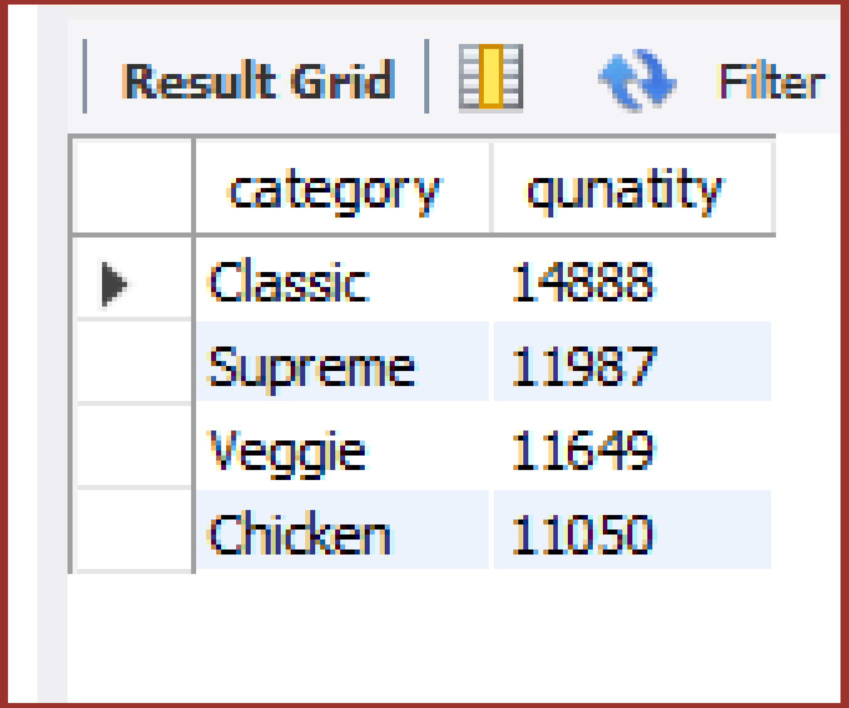


```
  name                          qunatity
▶ The Classic Deluxe Pizza      2453
  The Barbecue Chicken Pizza    2432
  The Hawaiian Pizza            2422
  The Pepperoni Pizza           2418
  The Thai Chicken Pizza        2371
```

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

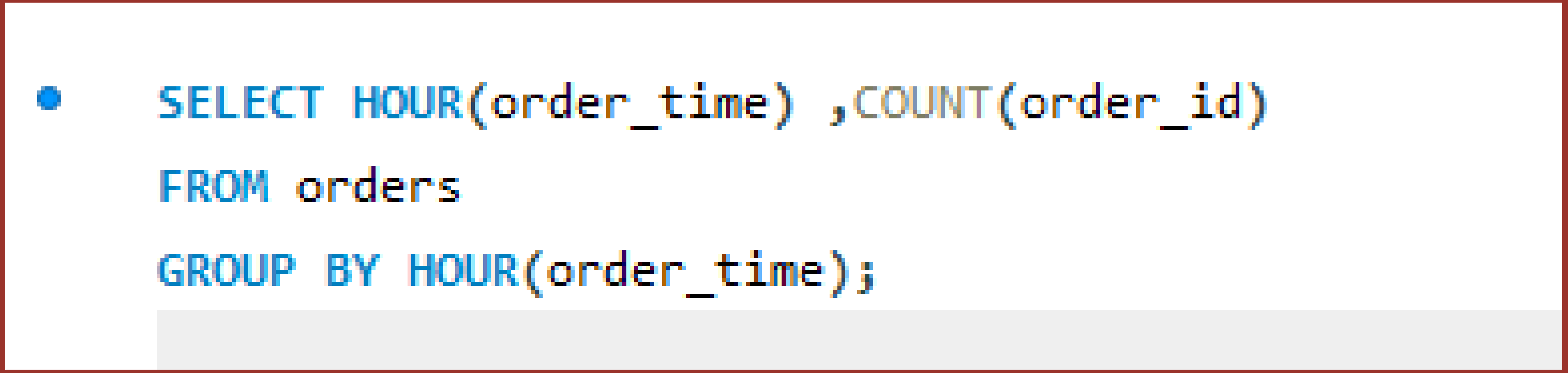


```
    FROM
        pizza_types
            JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
            JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY pizza_types.category
    ORDER BY qunatity DESC;
```

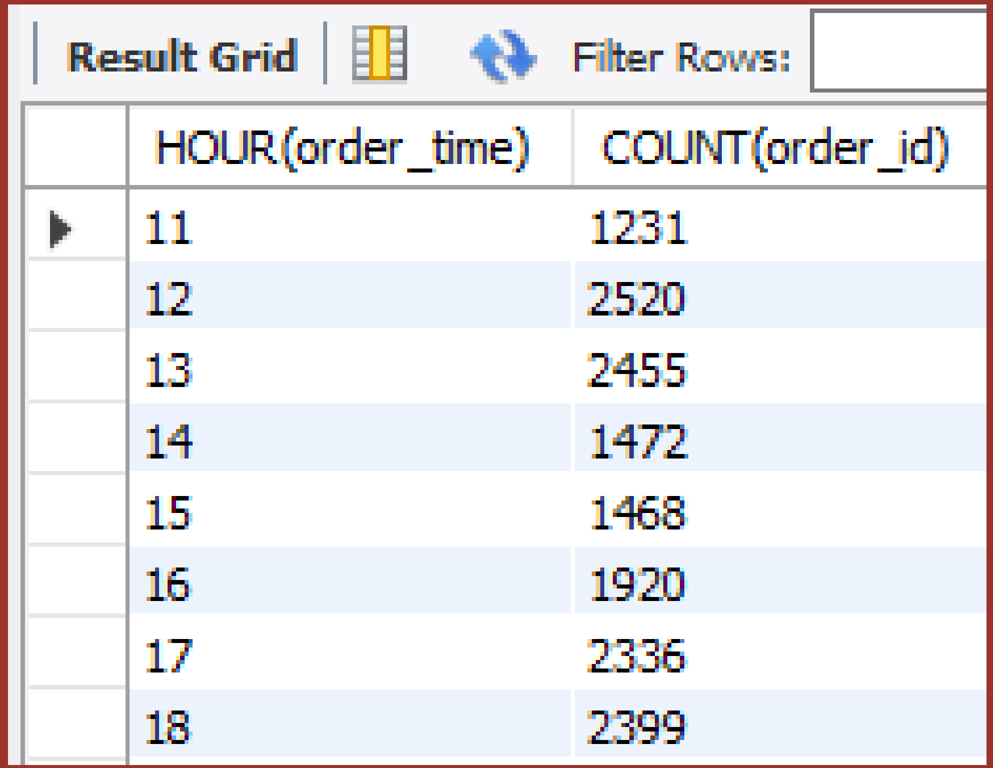


category	qunatity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7. Determine the distribution of orders by hour of the day.

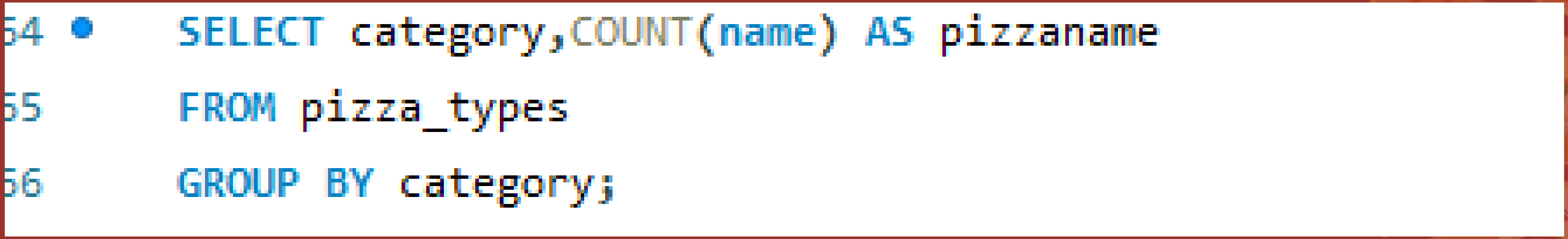


```
SELECT HOUR(order_time) ,COUNT(order_id)
FROM orders
GROUP BY HOUR(order_time);
```

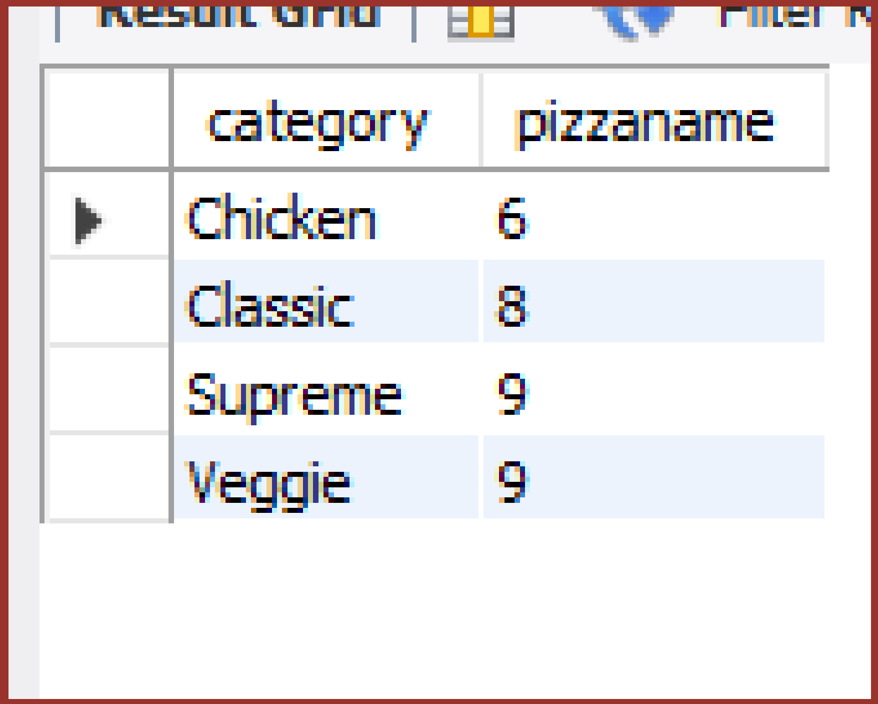


HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

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

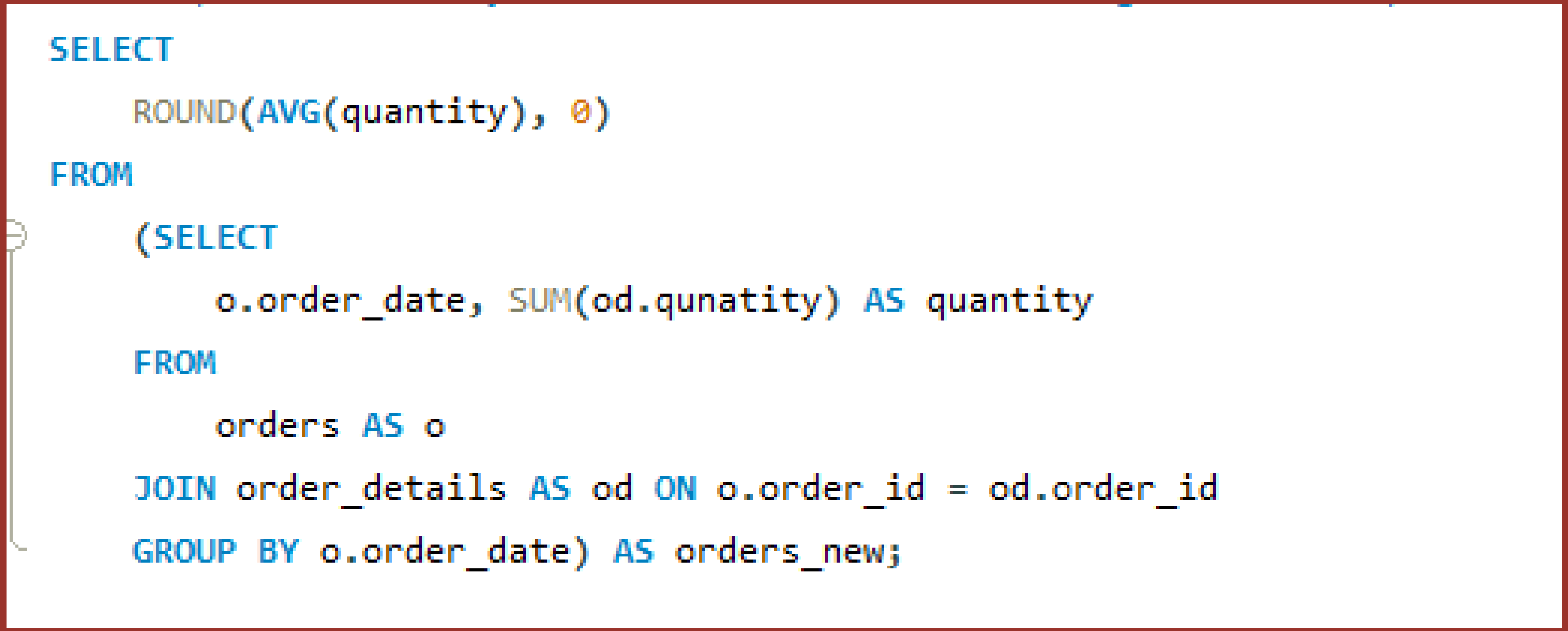


```
SELECT category,COUNT(name) AS pizzaname
FROM pizza_types
GROUP BY category;
```

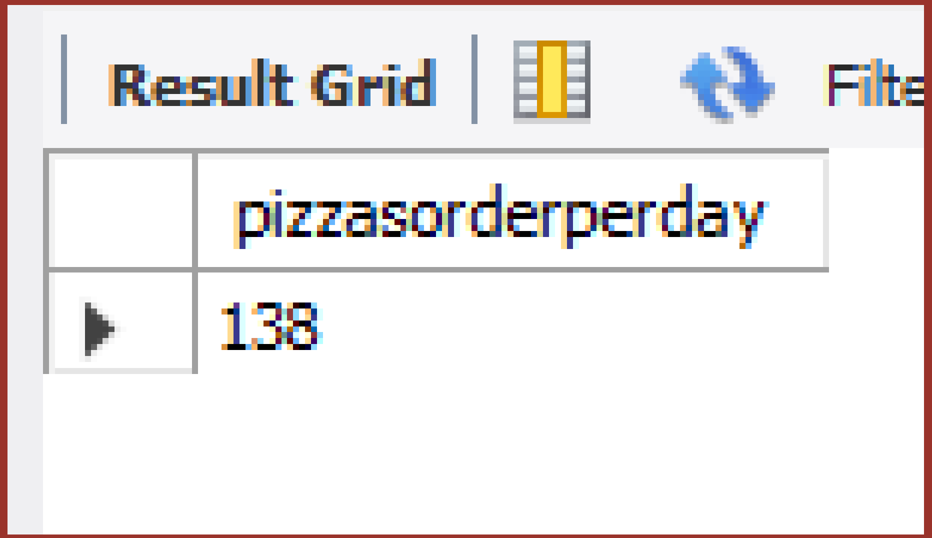


category	pizzaname
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

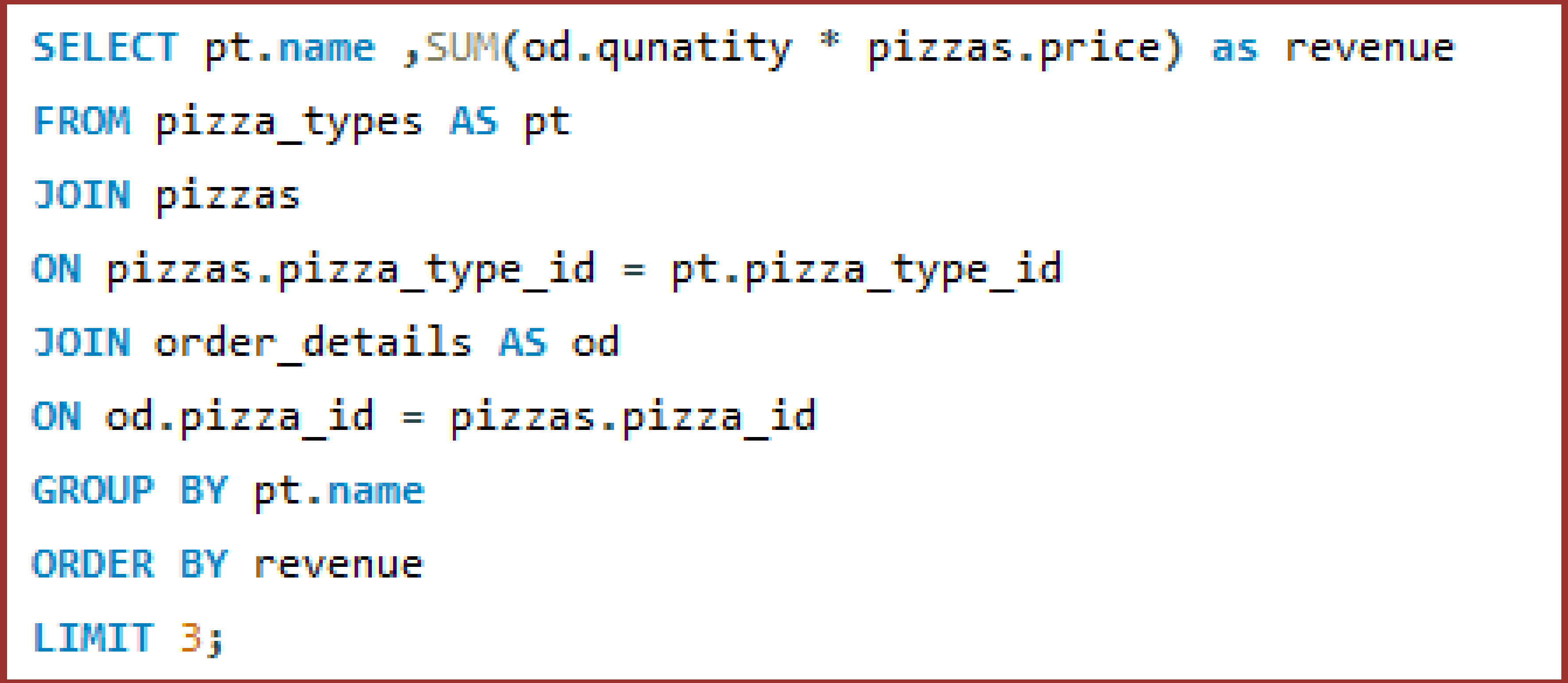


```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        o.order_date, SUM(od.qunatity) AS quantity
    FROM
        orders AS o
    JOIN order_details AS od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS orders_new;
```

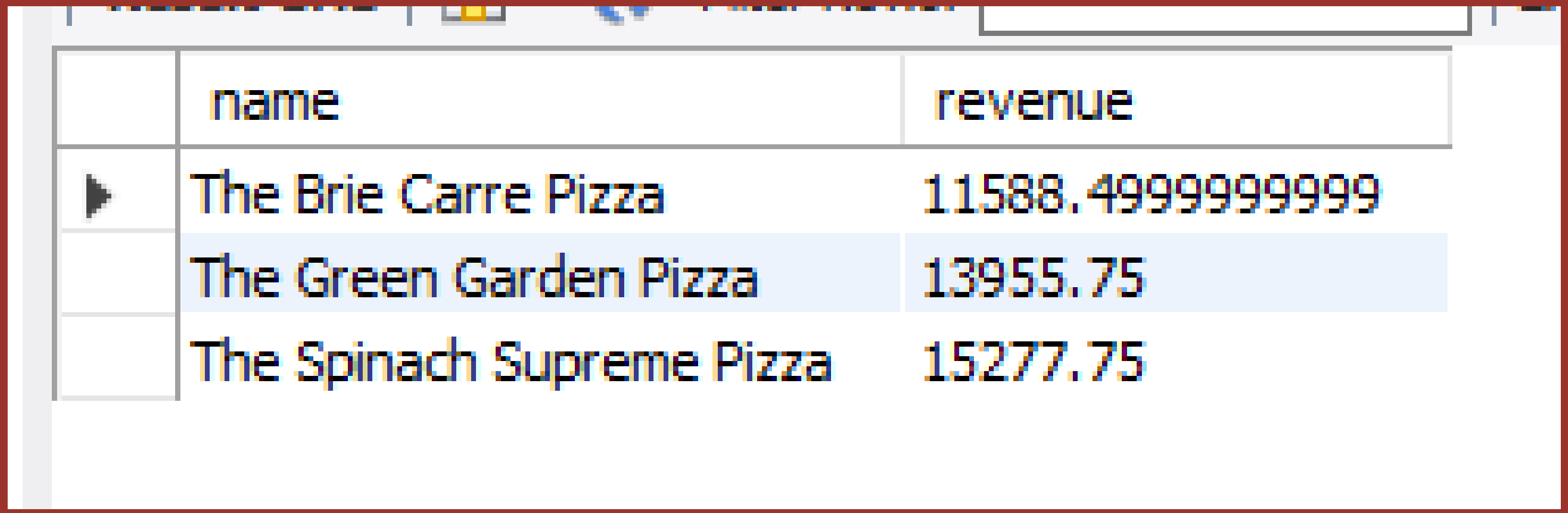


	pizzasorderperday
▶	138

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

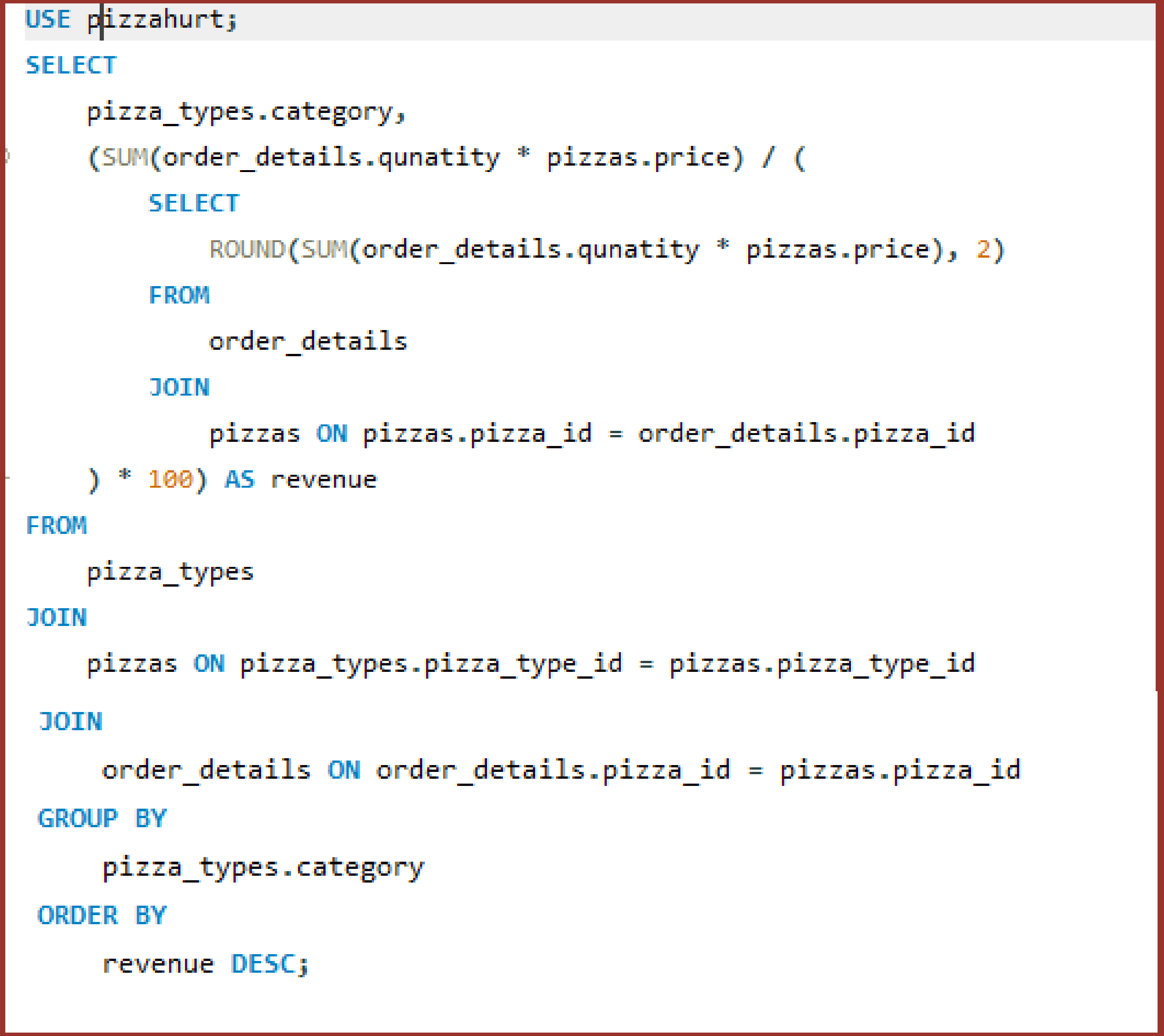


```
SELECT pt.name ,SUM(od.qunatity * pizzas.price) as revenue
FROM pizza_types AS pt
JOIN pizzas
ON pizzas.pizza_type_id = pt.pizza_type_id
JOIN order_details AS od
ON od.pizza_id = pizzas.pizza_id
GROUP BY pt.name
ORDER BY revenue
LIMIT 3;
```

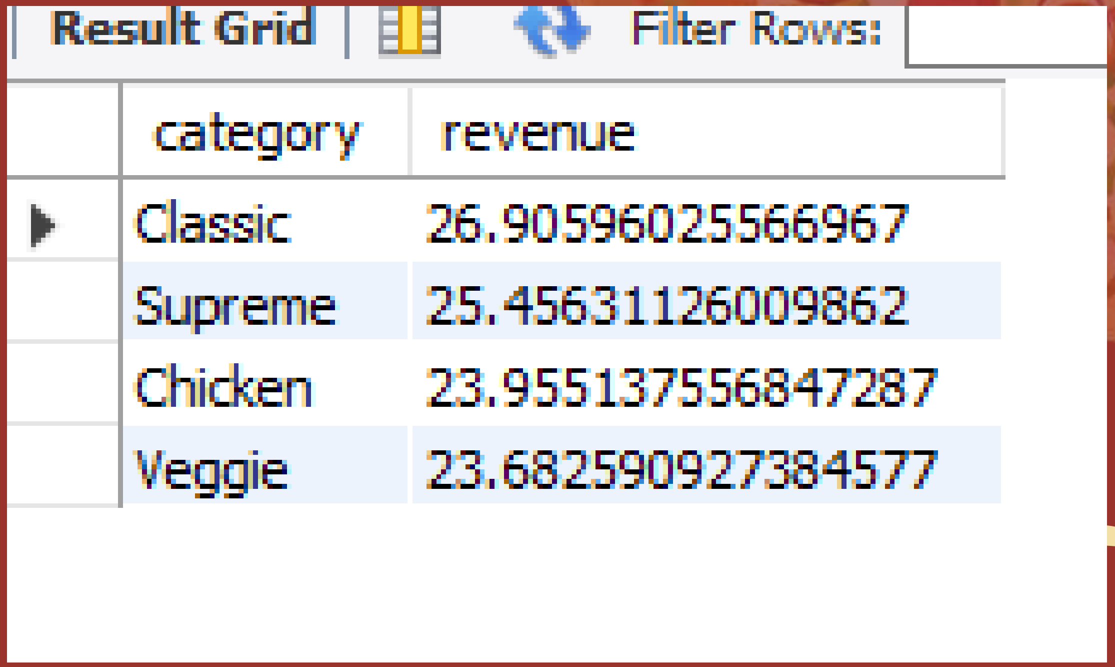


name	revenue
The Brie Carre Pizza	11588.4999999999
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75

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



```
USE pizzahurt;
SELECT
    pizza_types.category,
    (SUM(order_details.qunatity * pizzas.price) / (
        SELECT
            ROUND(SUM(order_details.qunatity * pizzas.price), 2)
        FROM
            order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id
    ) * 100) 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	26.9059602556967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

THANK YoU