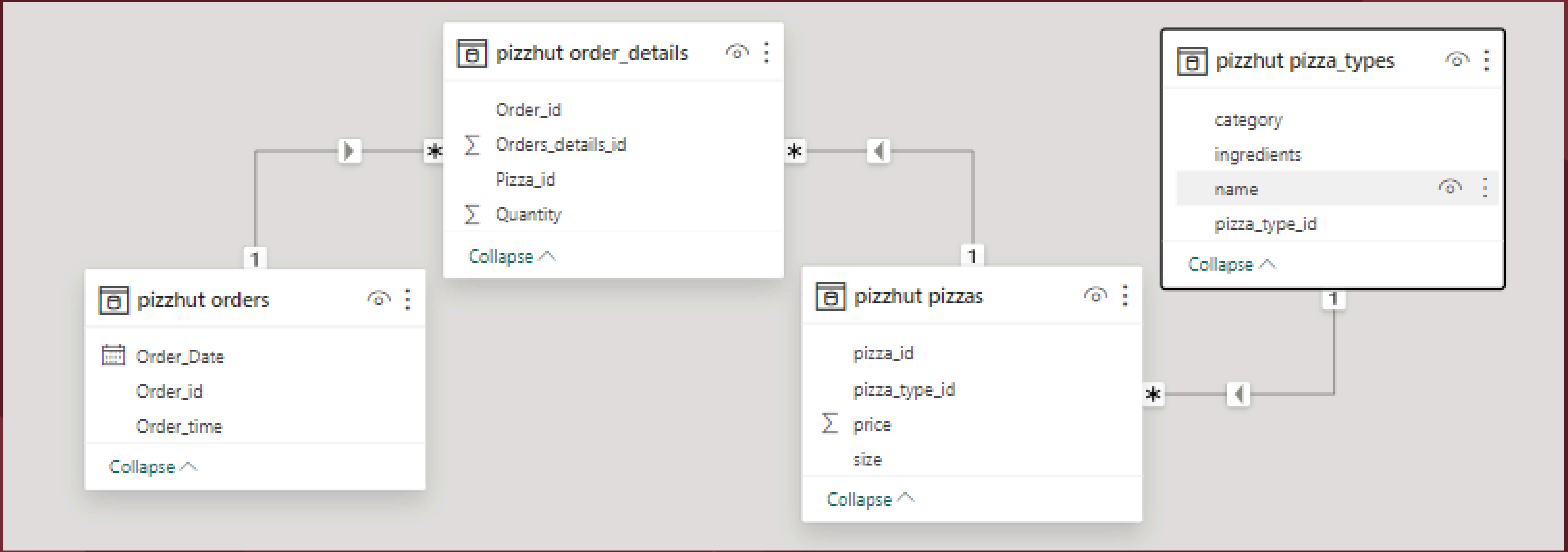
SQL SALES REPORT ON PIZZA

INTRODUCTION

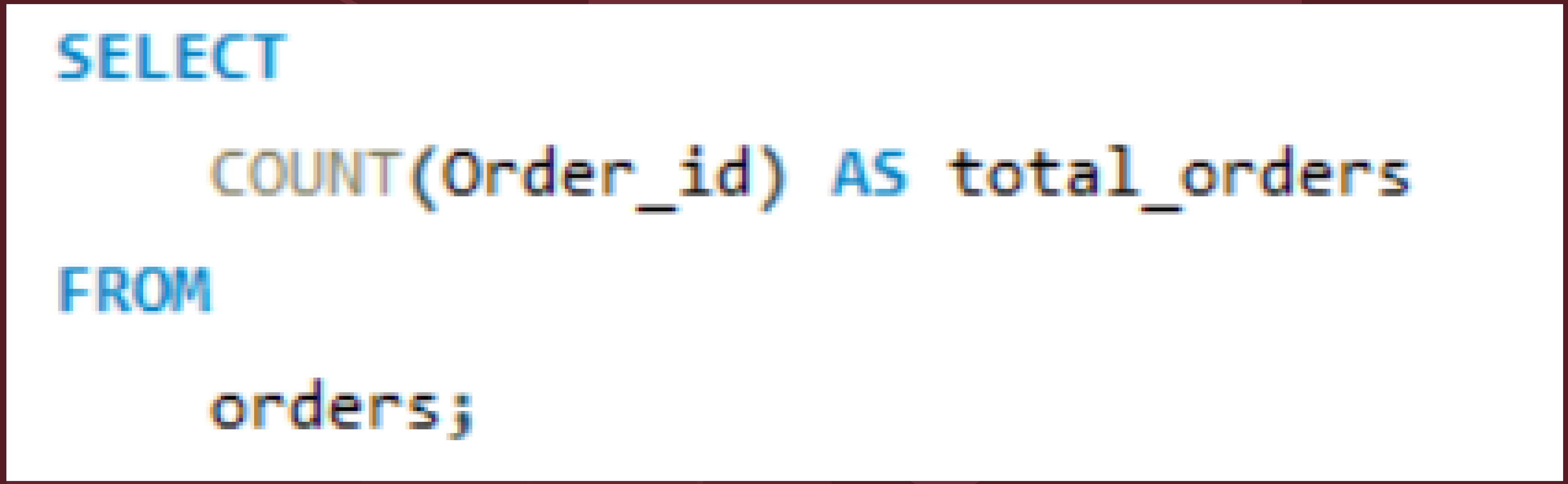
Welcome to Our Pizza Sales Report Using SQL. In this project, I have utilized SQL Queries to solve questions related to Pizza Sales. The questions are given below.

1) Retrieve the total number of orders placed.
2) Calculate the total Revenue generated from the pizza sales
3) Identified the highest price pizza
4) identified 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
7) determine the distribution of orders by an hour of the day
8) Join the relevant tables to find the category-wise distribution of pizzas.
9) Group by order by date and calculate the average number of pizzas ordered by per day
10) Determine the top 3 most ordered pizzas type based on revenue
11) Calculate the percentage contribution of each pizza type (name or category) to total revenue timestamp 58:00
12) Analyze the cumulative revenue generated over time
13) Determine the top 3 most ordered pizzas types based on revenue for each pizza category

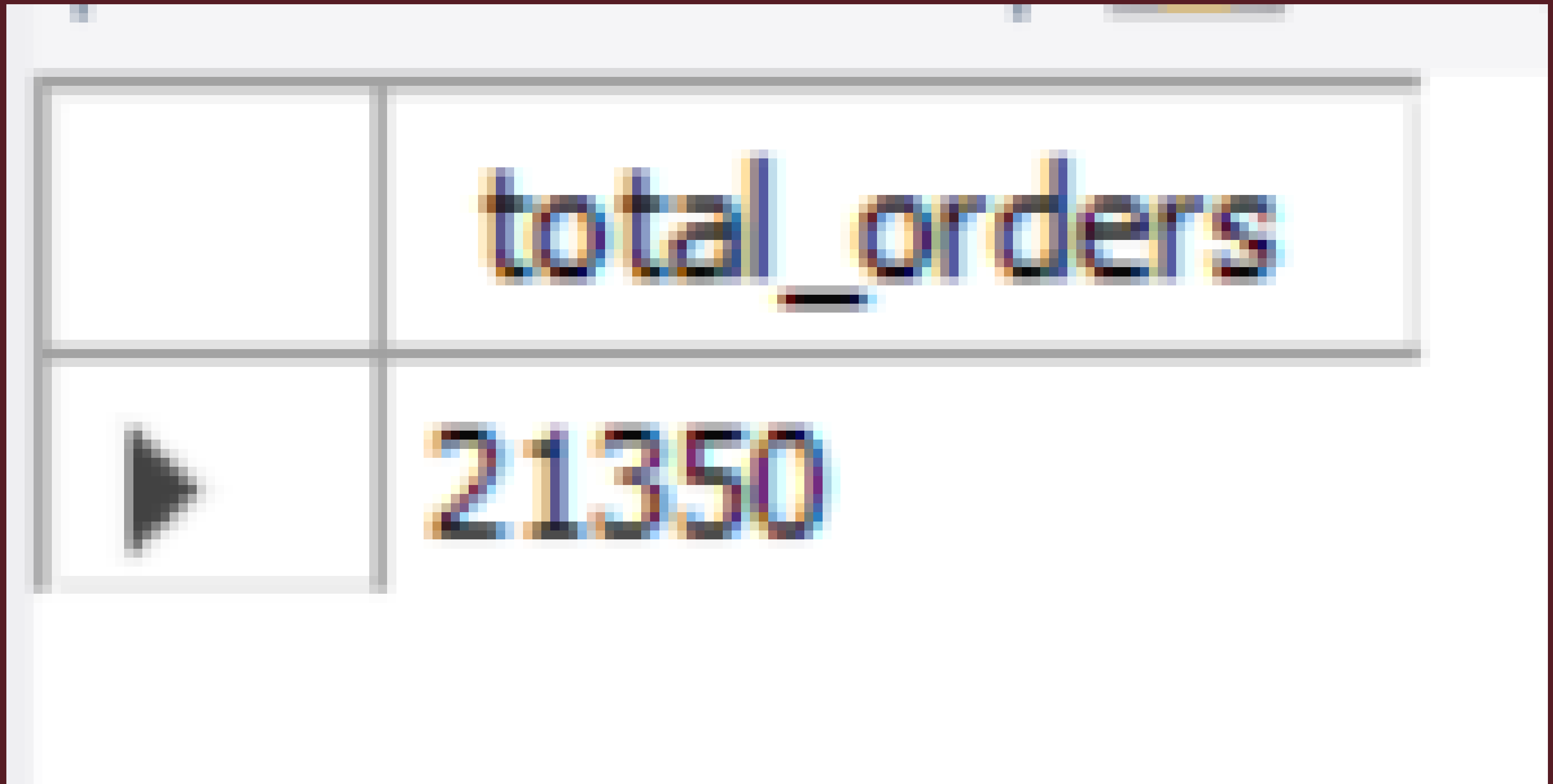
DATABASE SCHEMA



1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

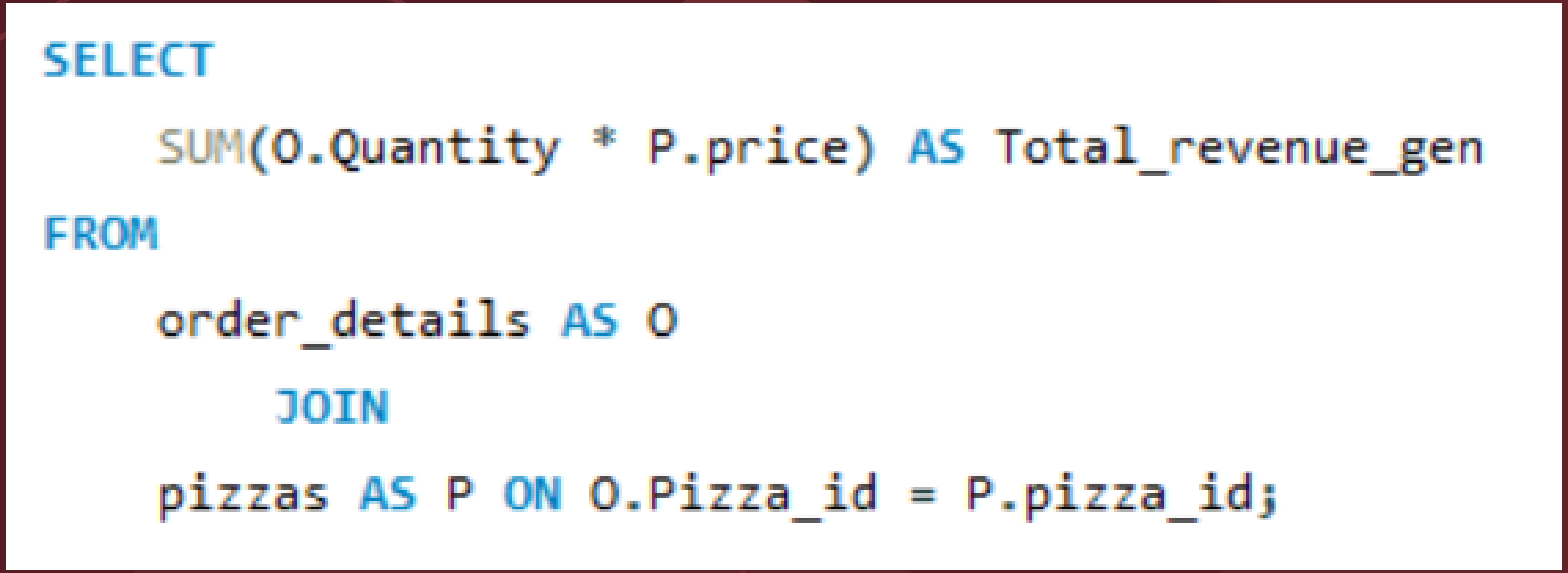


```sql
SELECT
    COUNT(Order_id) AS total_orders
FROM
    orders;
```

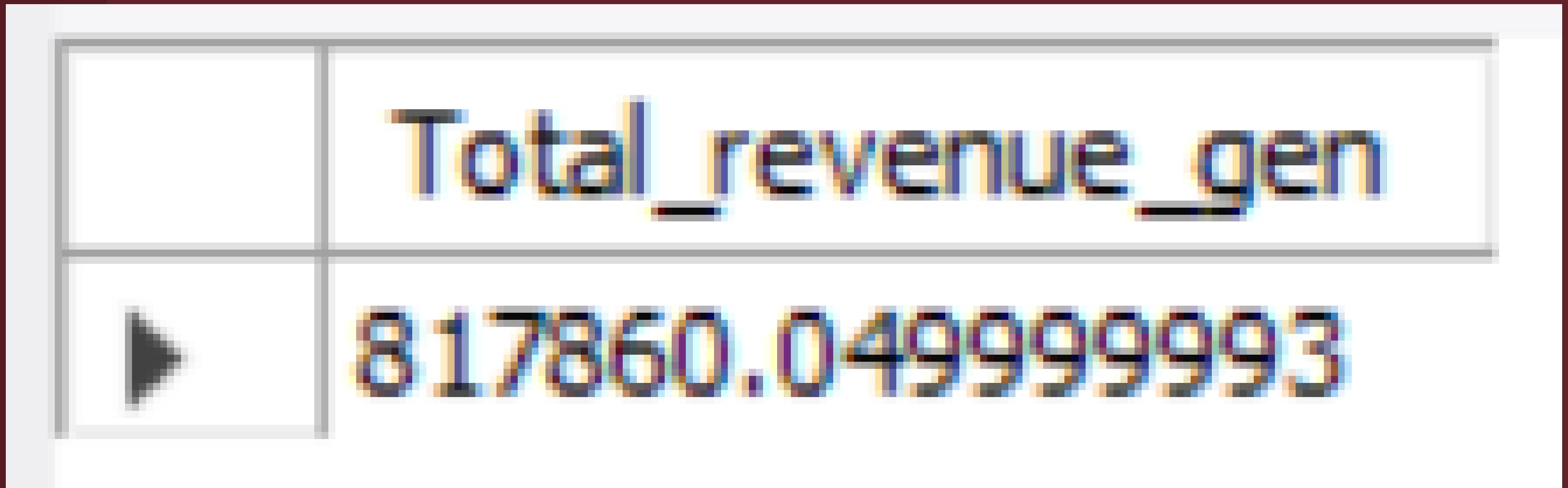


```
  total_orders
▶ 21350
```

2) CALCULATE THE TOTAL REVENUE GENERATED FROM THE PIZA SALES

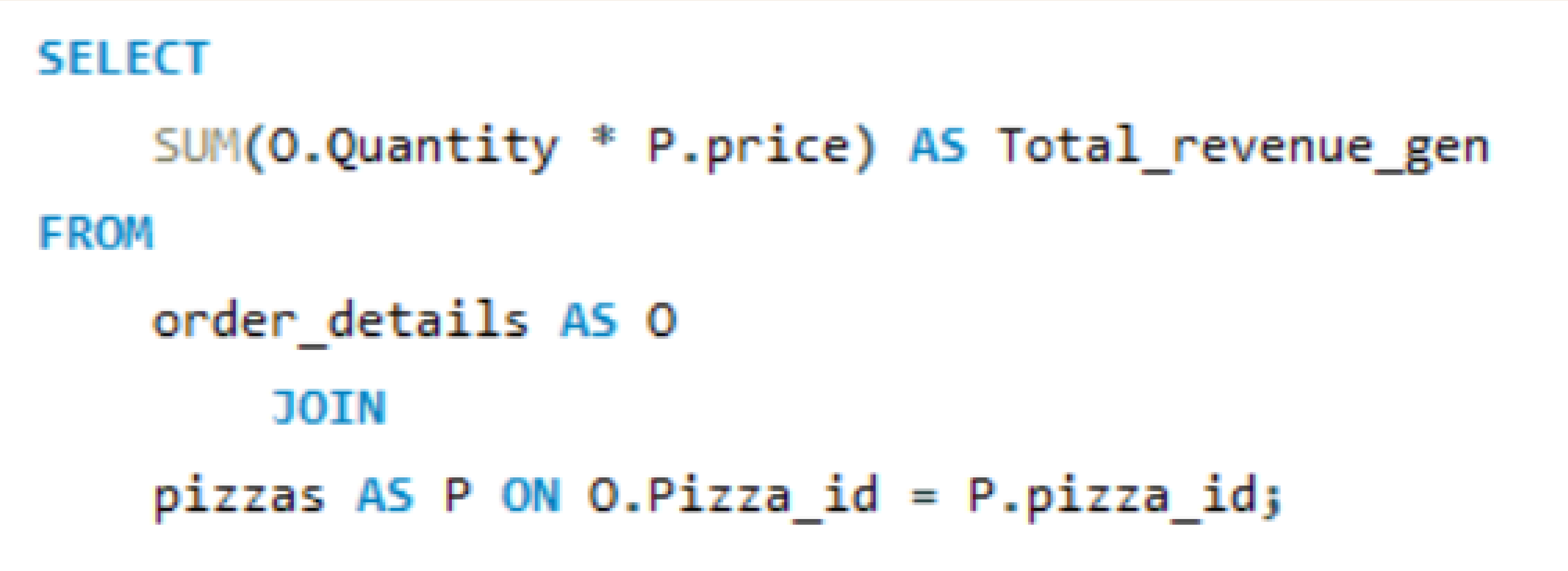


```
SELECT
    SUM(O.Quantity * P.price) AS Total_revenue_gen
FROM
    order_details AS O
        JOIN
    pizzas AS P ON O.Pizza_id = P.pizza_id;
```

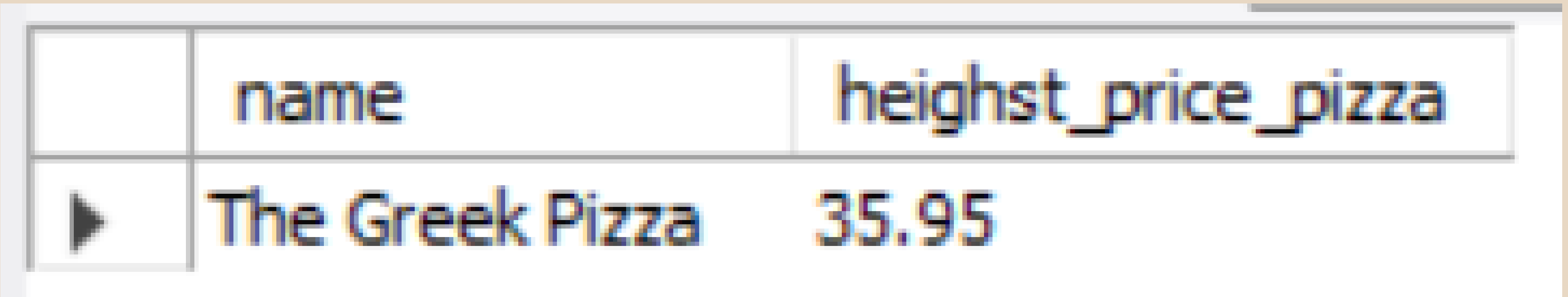


	Total_revenue_gen
▶	817860.049999993

3) Identified the highest price pizza

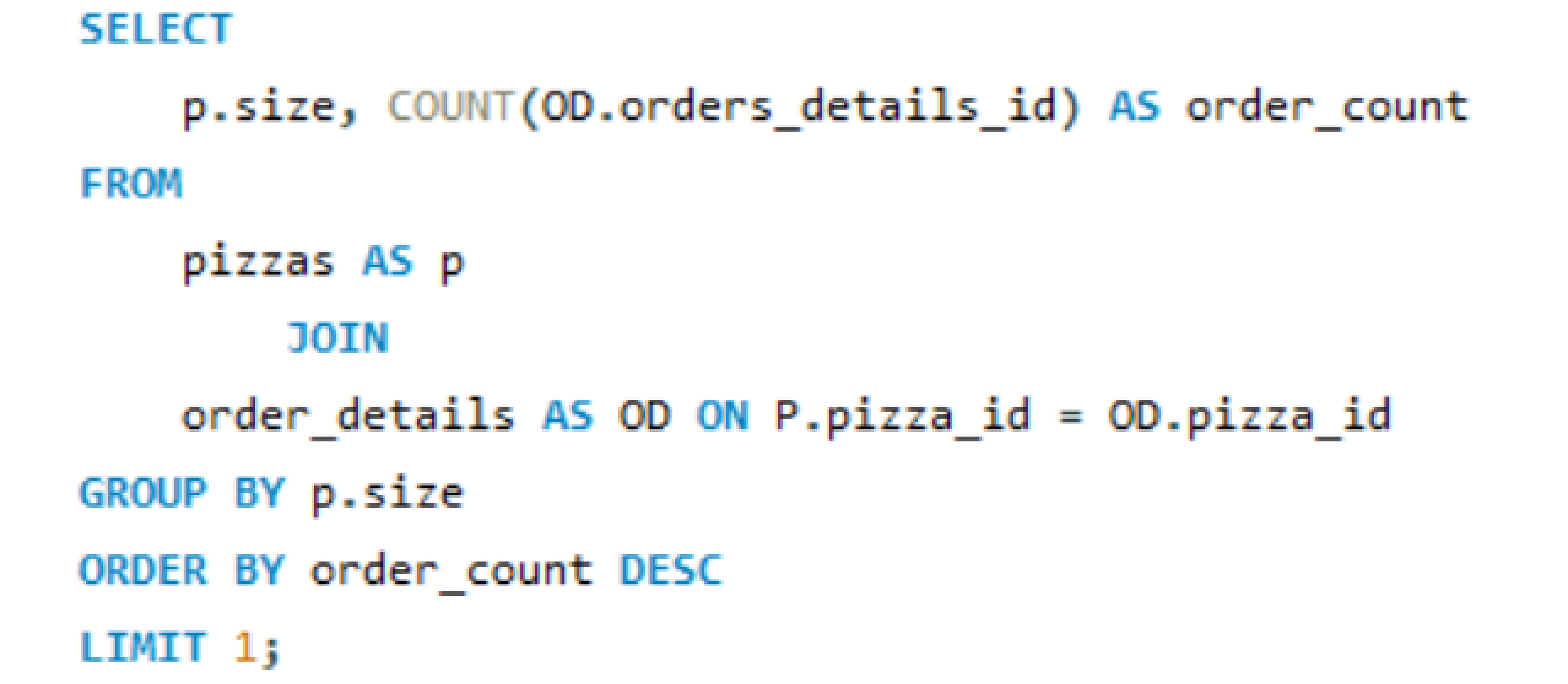


```
SELECT
    SUM(O.Quantity * P.price) AS Total_revenue_gen
FROM
    order_details AS O
        JOIN
    pizzas AS P ON O.Pizza_id = P.pizza_id;
```

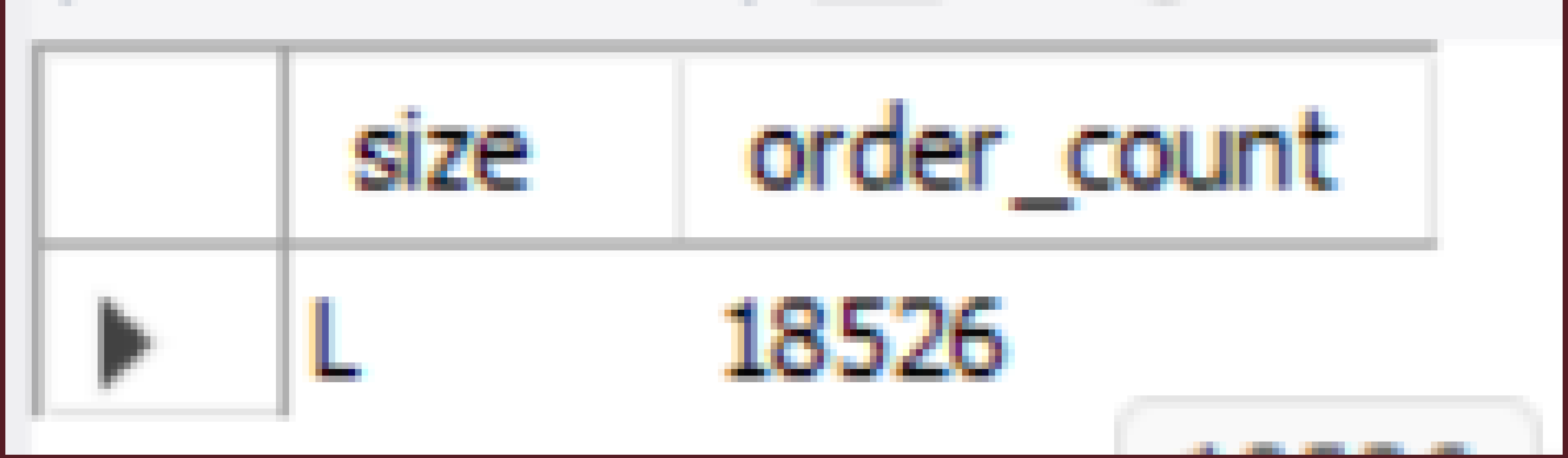


	name	heighst_price_pizza
▶	The Greek Pizza	35.95

4) IDENTIFIED THE MOST COMMON PIZZA SIZE ORDERED

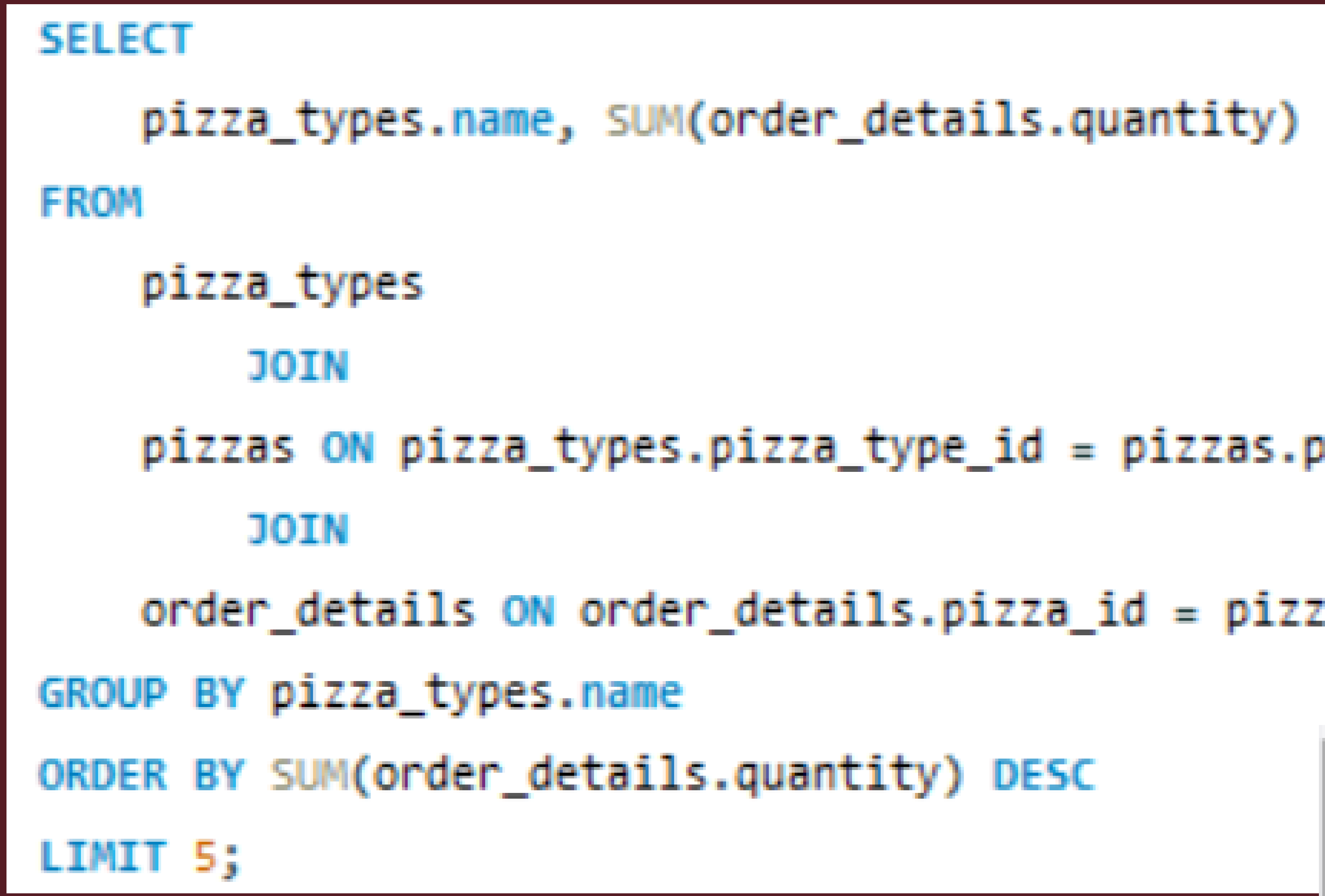


```
SELECT
    p.size, COUNT(OD.orders_details_id) AS order_count
FROM
    pizzas AS p
        JOIN
    order_details AS OD ON P.pizza_id = OD.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1;
```

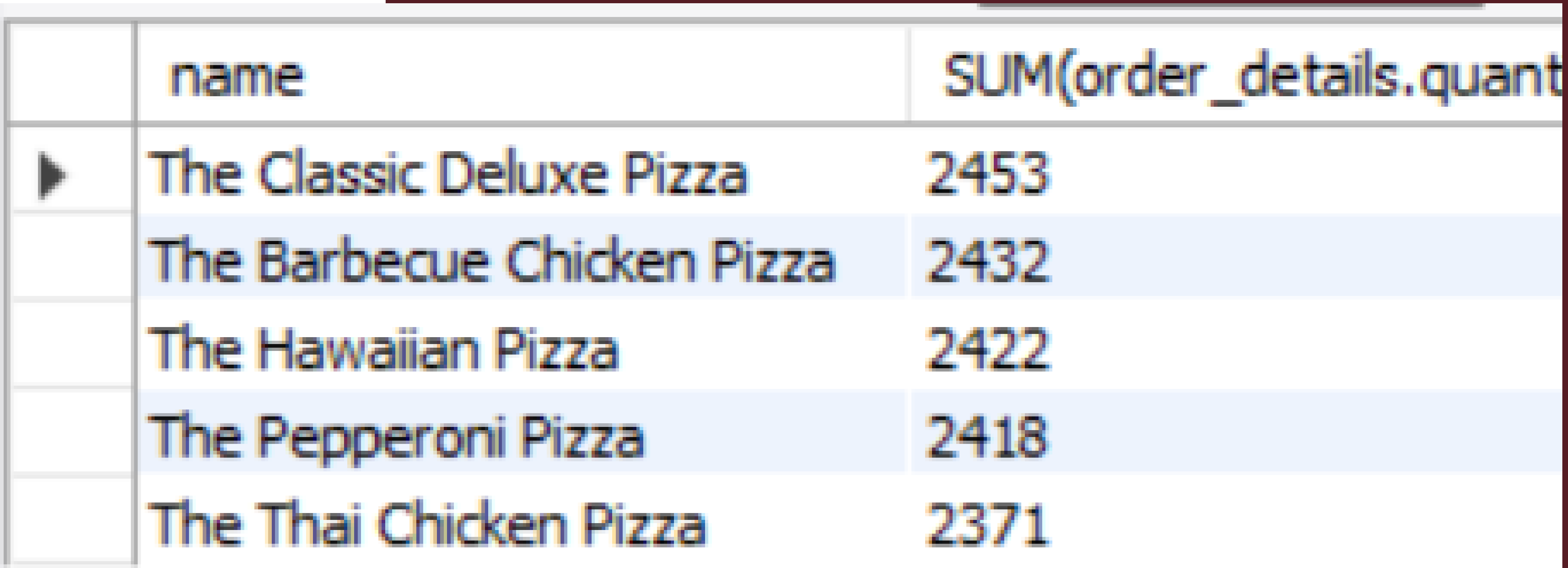


	size	order_count
▶	L	18526

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

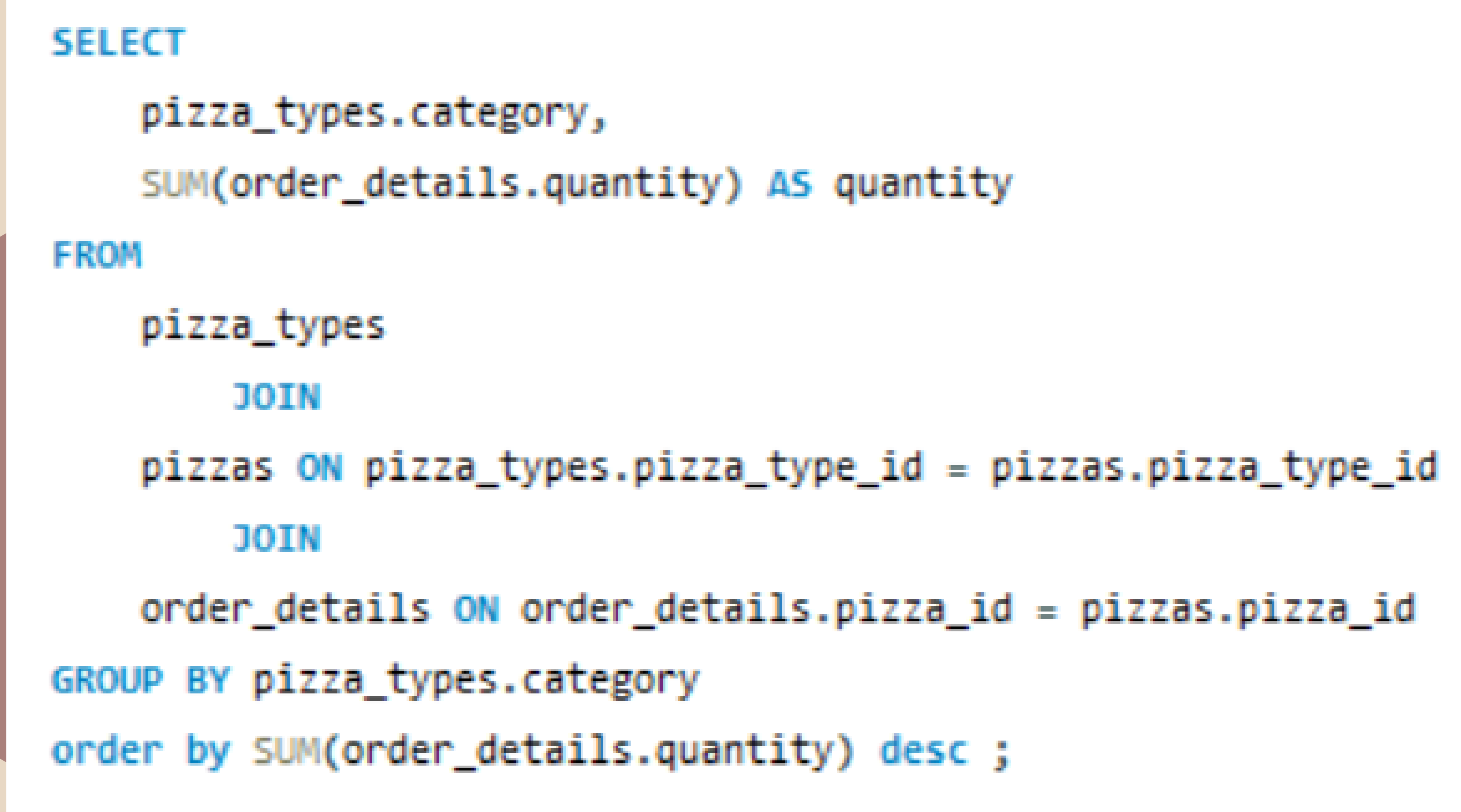


```
SELECT
    pizza_types.name, SUM(order_details.quantity)
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY SUM(order_details.quantity) DESC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

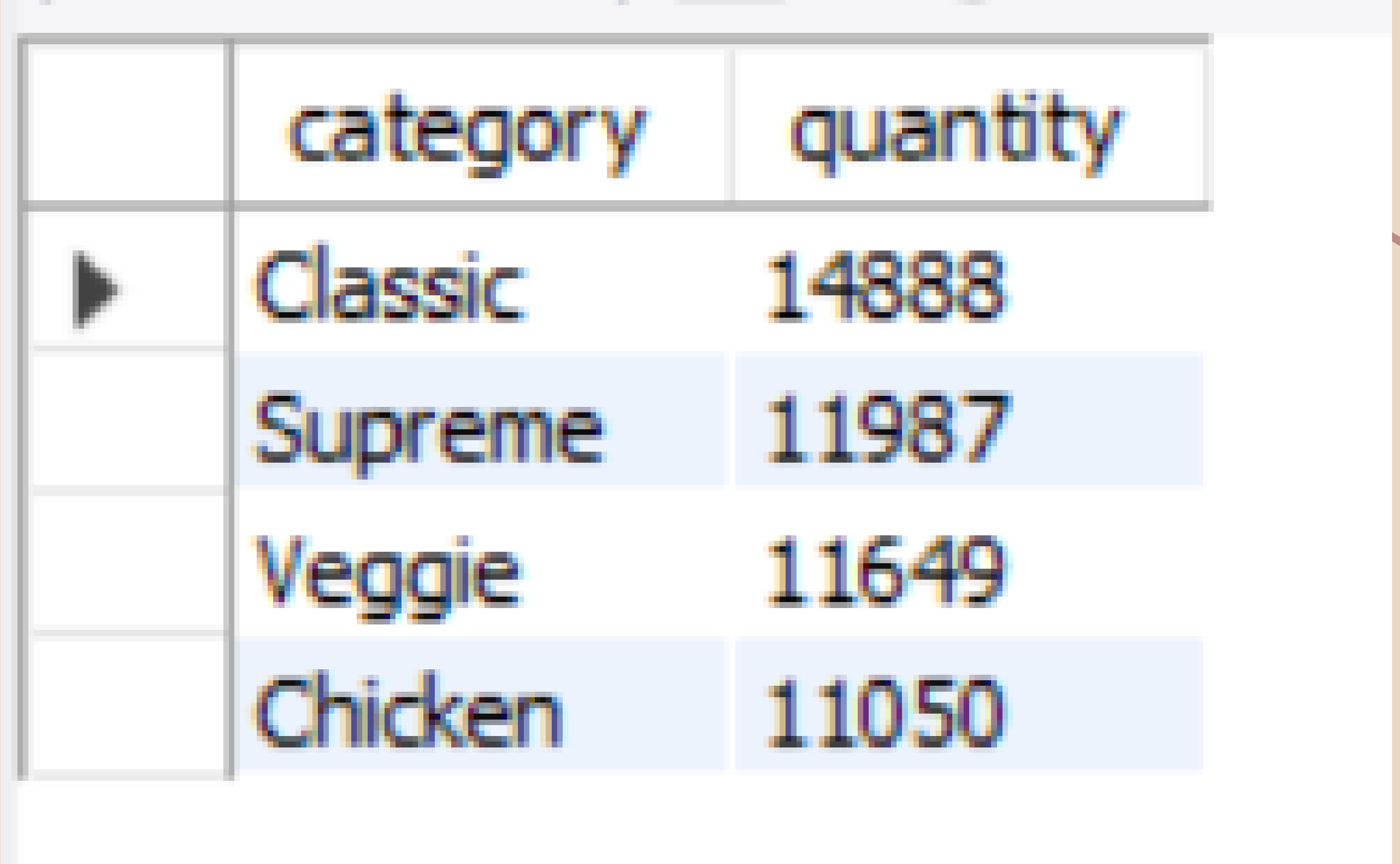


name	SUM(order_details.quant
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

6) join the necessary tables to find the total quantity of each pizza category

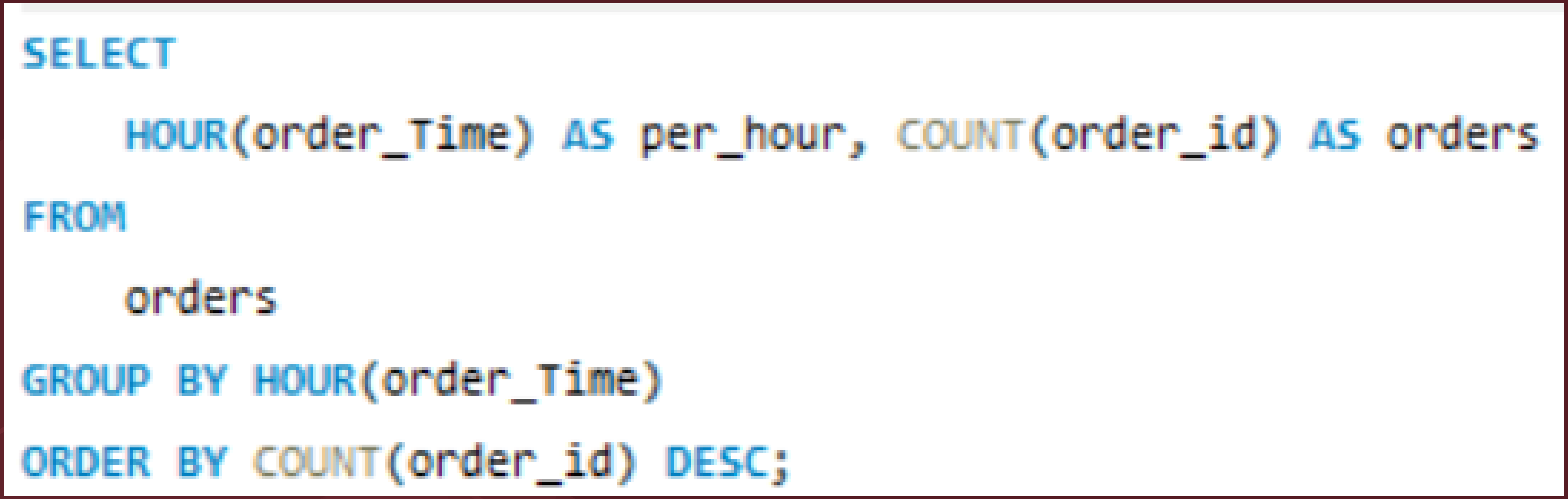


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
order by SUM(order_details.quantity)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

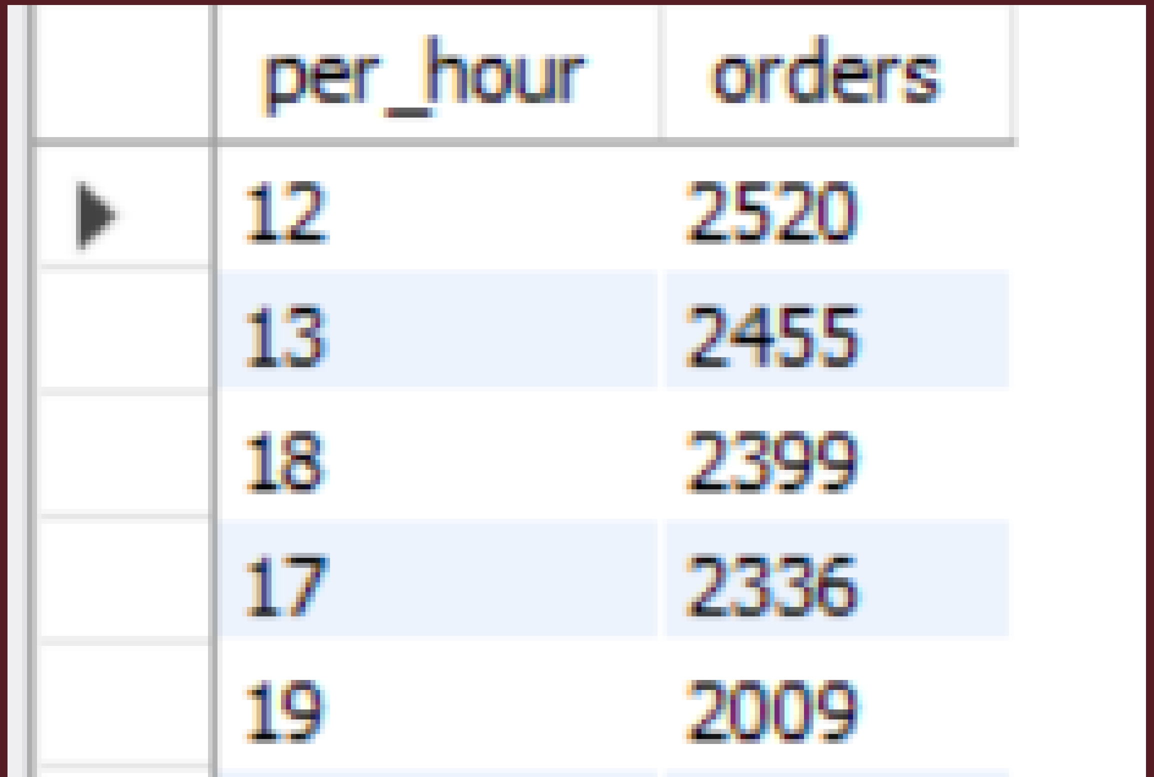


	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

7)determine the distribution of orders by an hour of the day

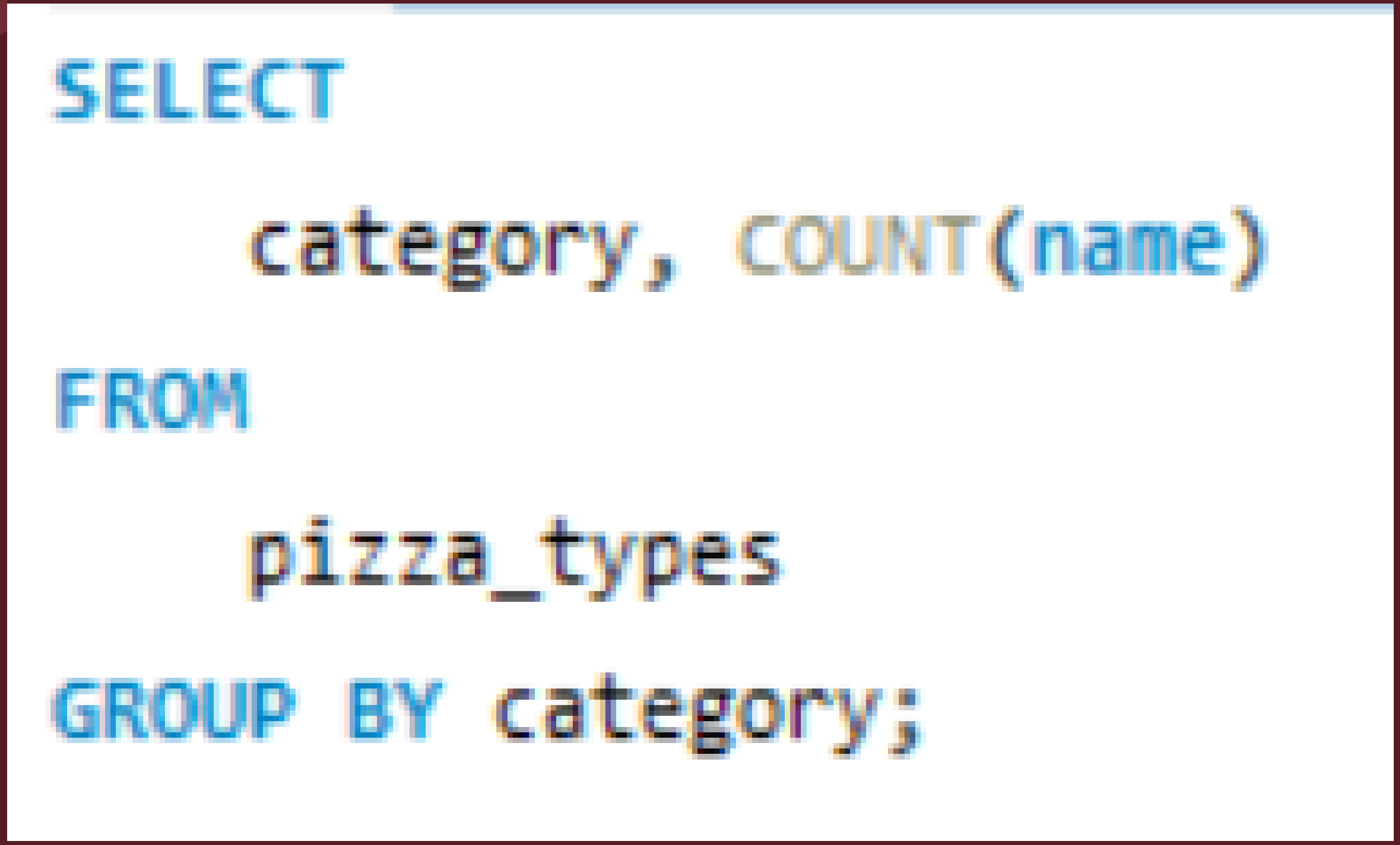


```
SELECT
    HOUR(order_Time) AS per_hour, COUNT(order_id) AS orders
FROM
    orders
GROUP BY HOUR(order_Time)
ORDER BY COUNT(order_id) DESC;
```

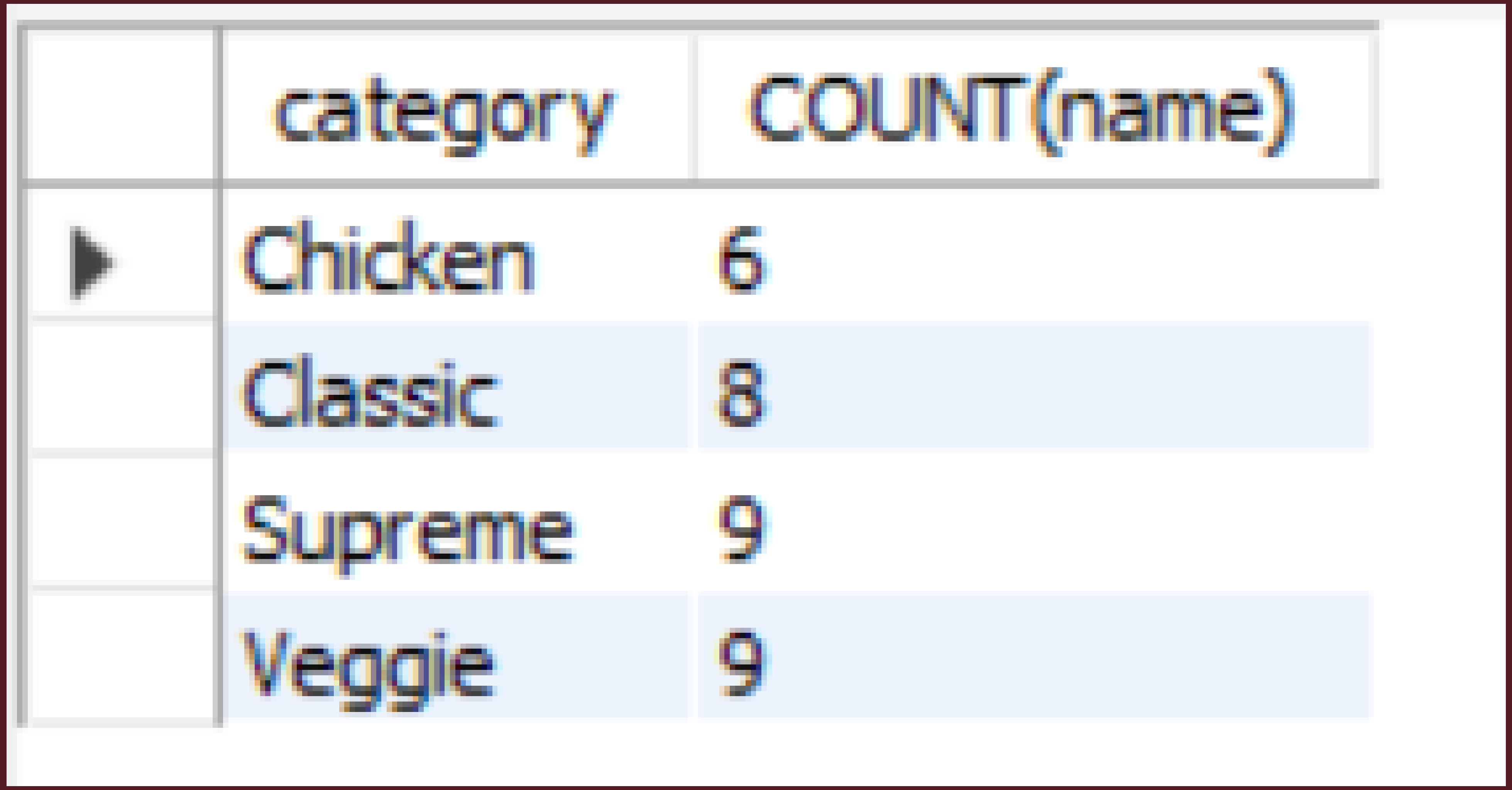


per_hour	orders
12	2520
13	2455
18	2399
17	2336
19	2009

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

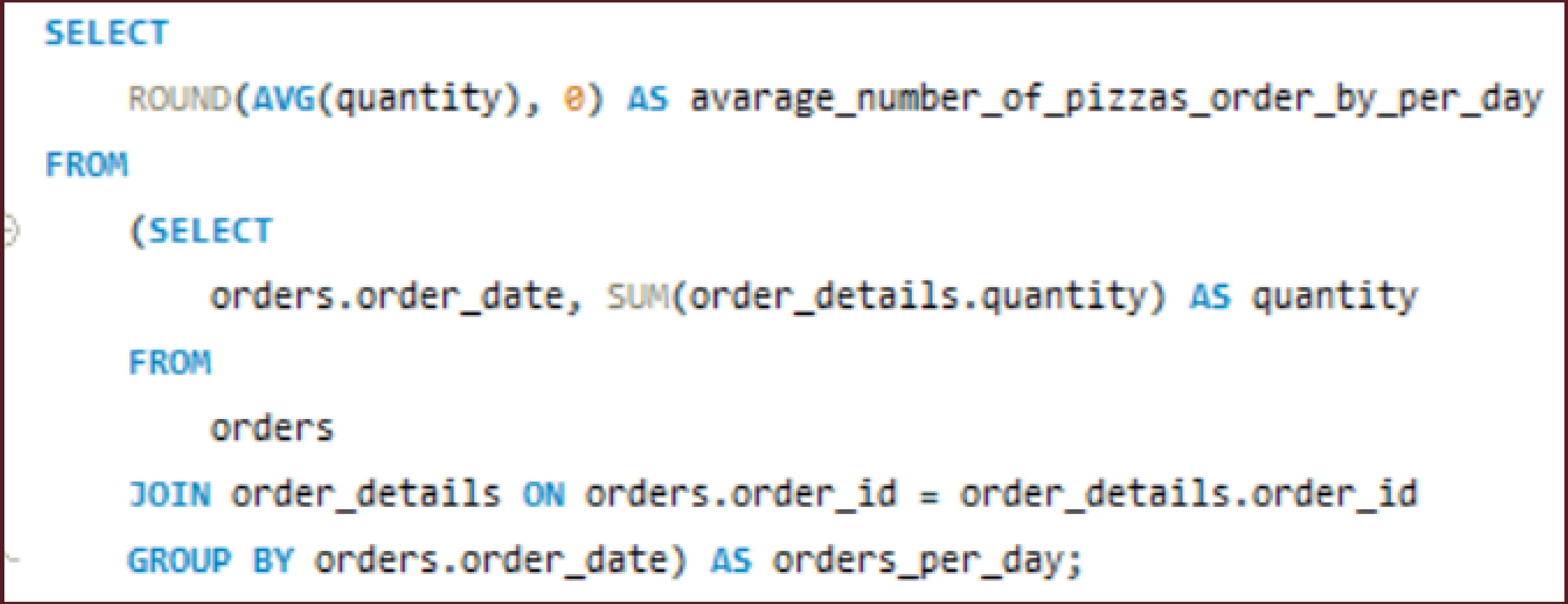


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

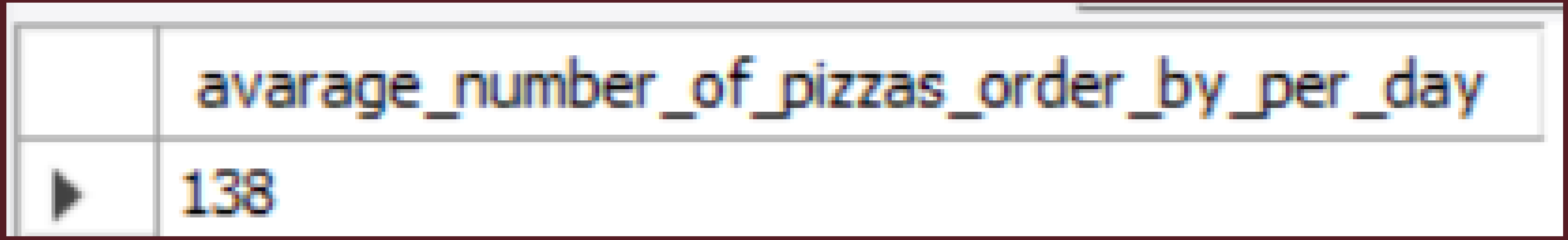


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

9)GROUP BY ORDER BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

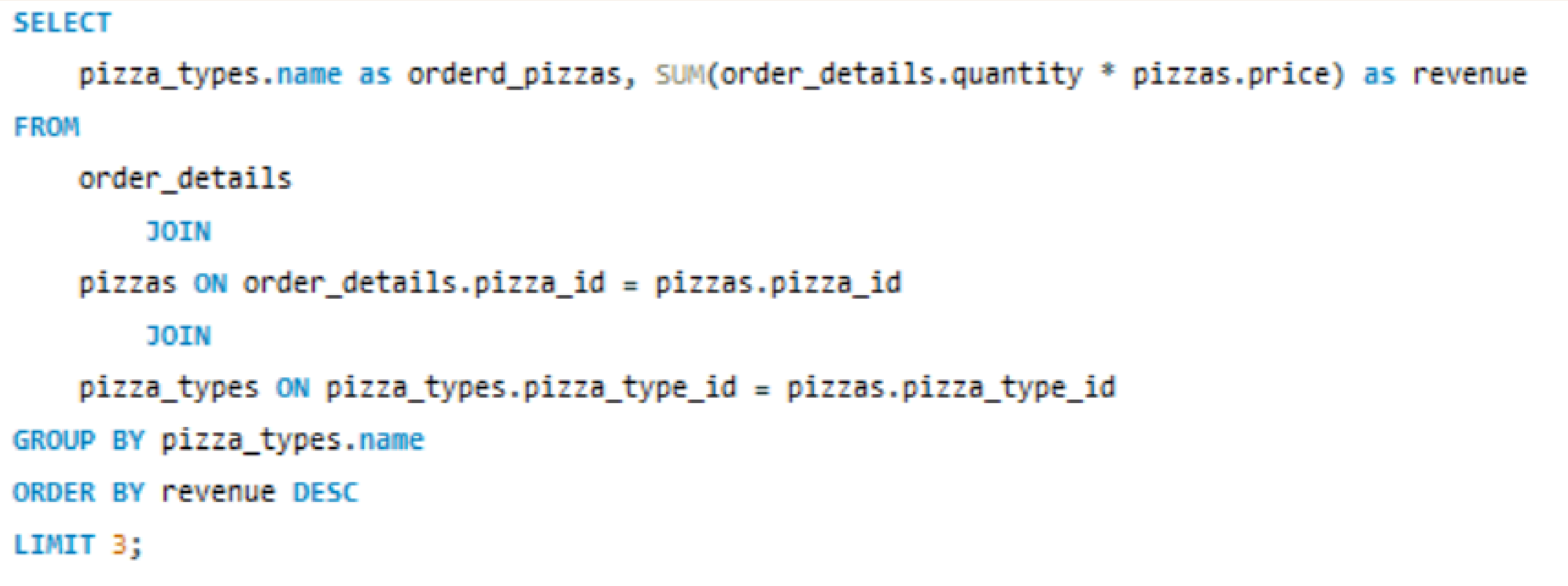


```
SELECT
    ROUND(AVG(quantity), 0) AS avarage_number_of_pizzas_order_by_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS orders_per_day;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

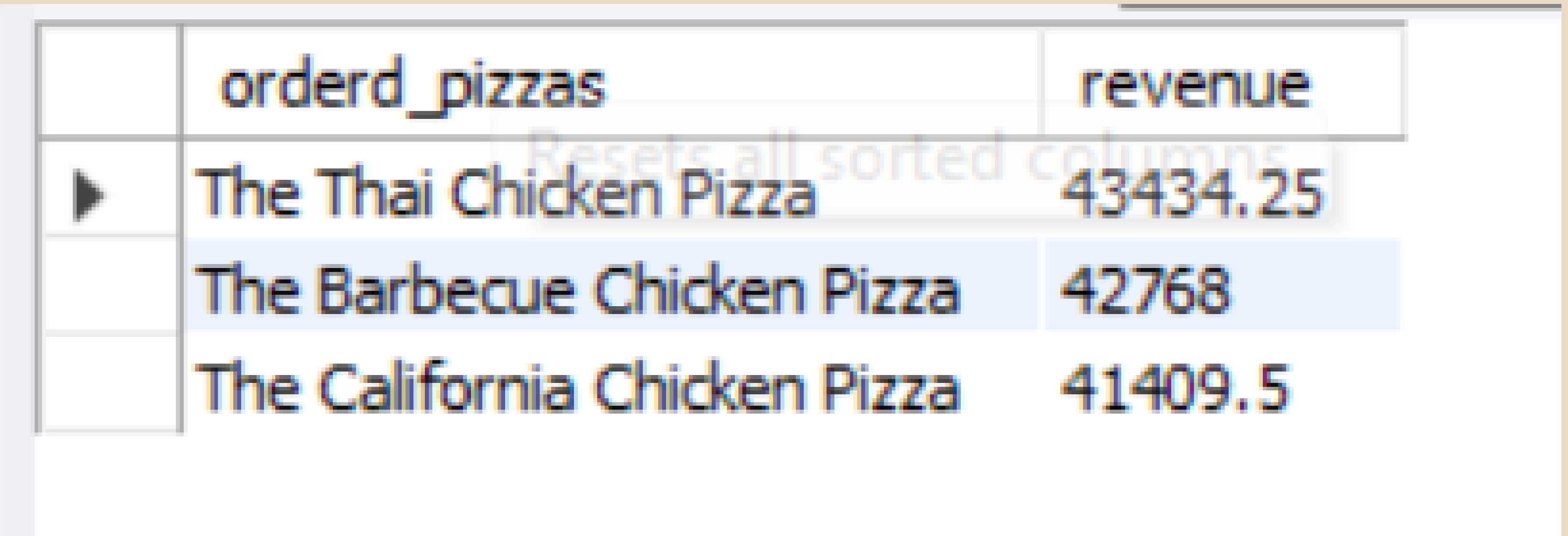


avarage_number_of_pizzas_order_by_per_day
138

10) Determine the top 3 most ordered pizzas type based on revenue

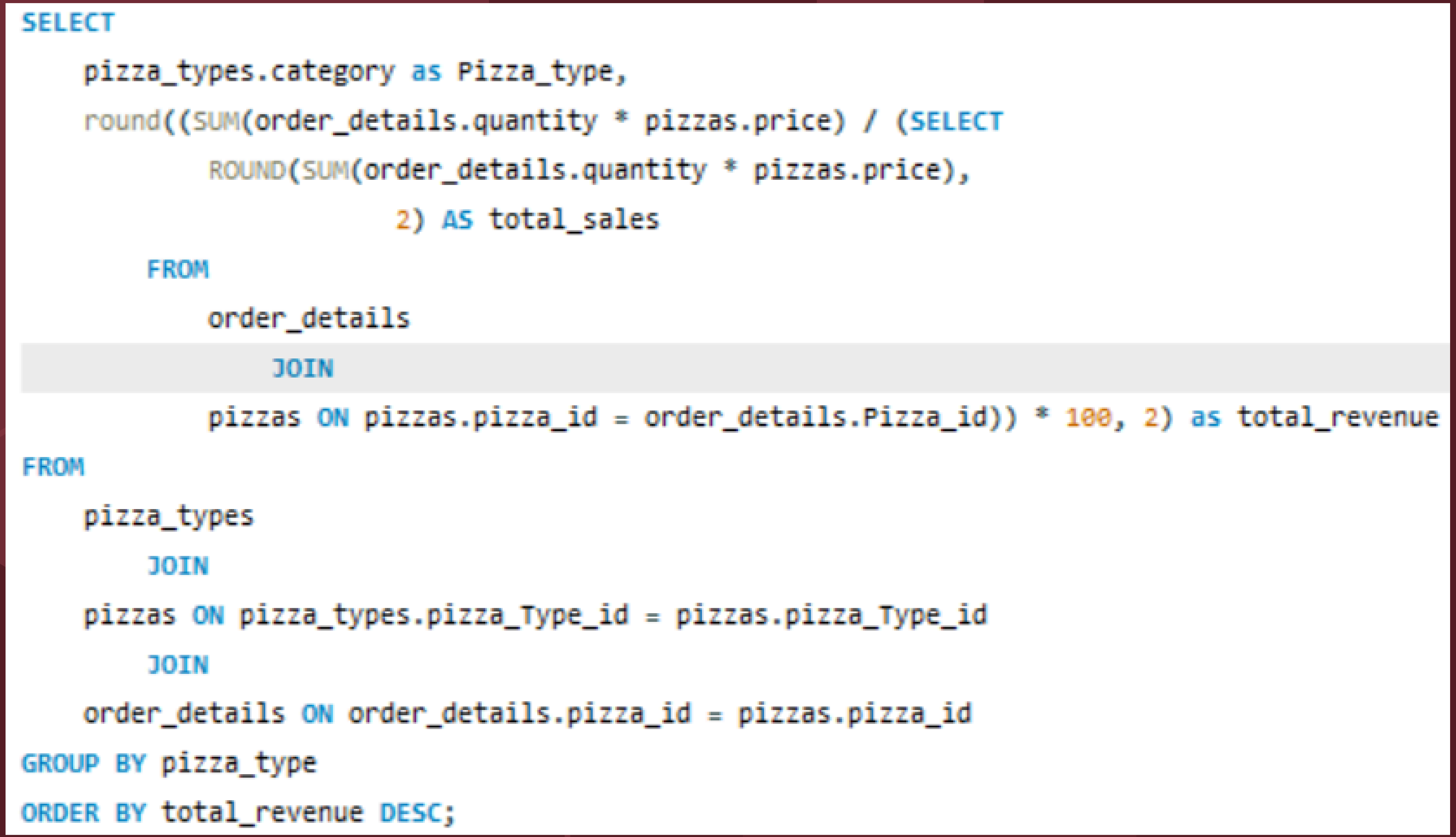


```
SELECT
    pizza_types.name as orderd_pizzas, SUM(order_details.quantity * pizzas.price) as revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name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

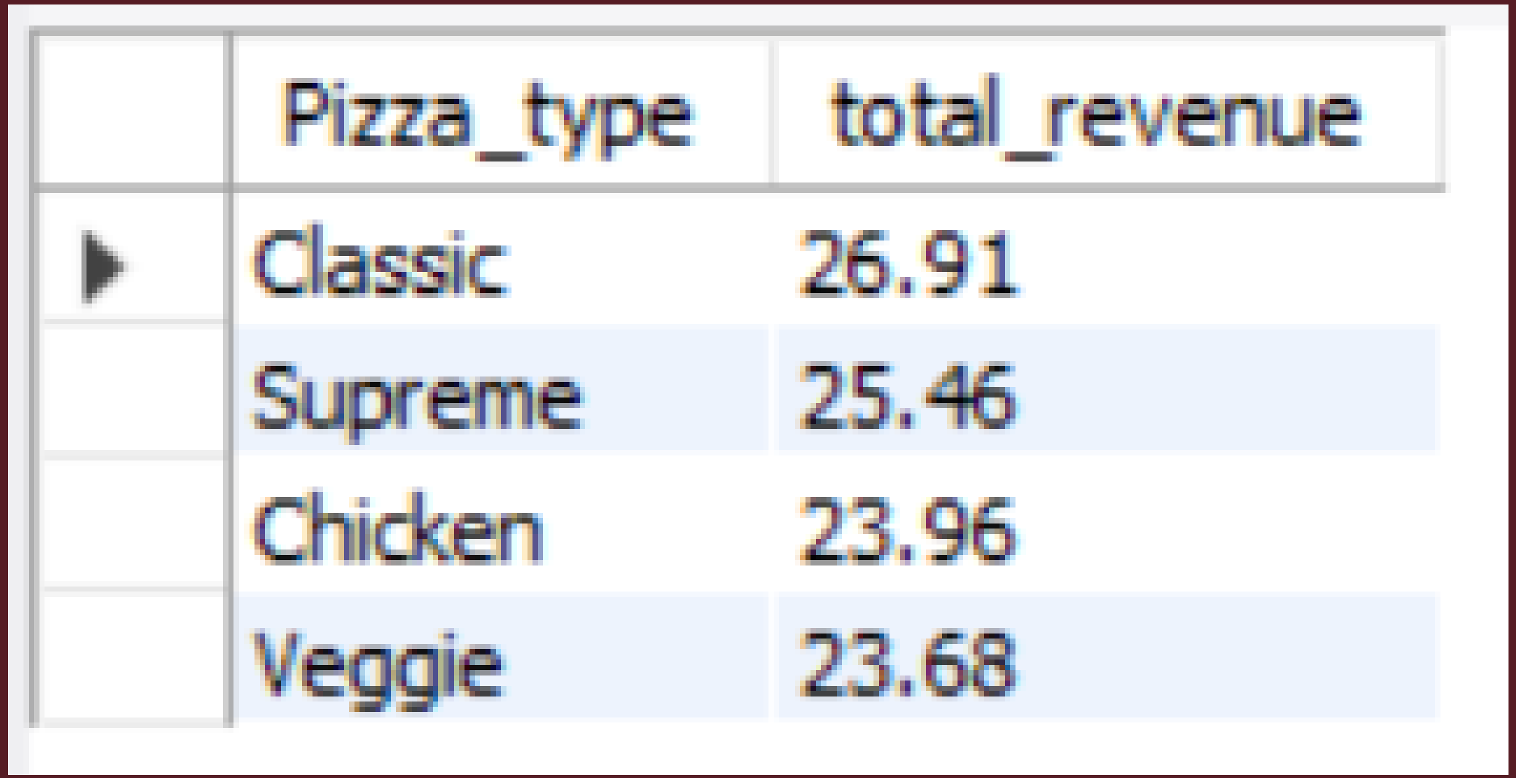


orderd_pizzas	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

11) CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE (NAME OR CATEGORY) TO TOTAL REVENUE.

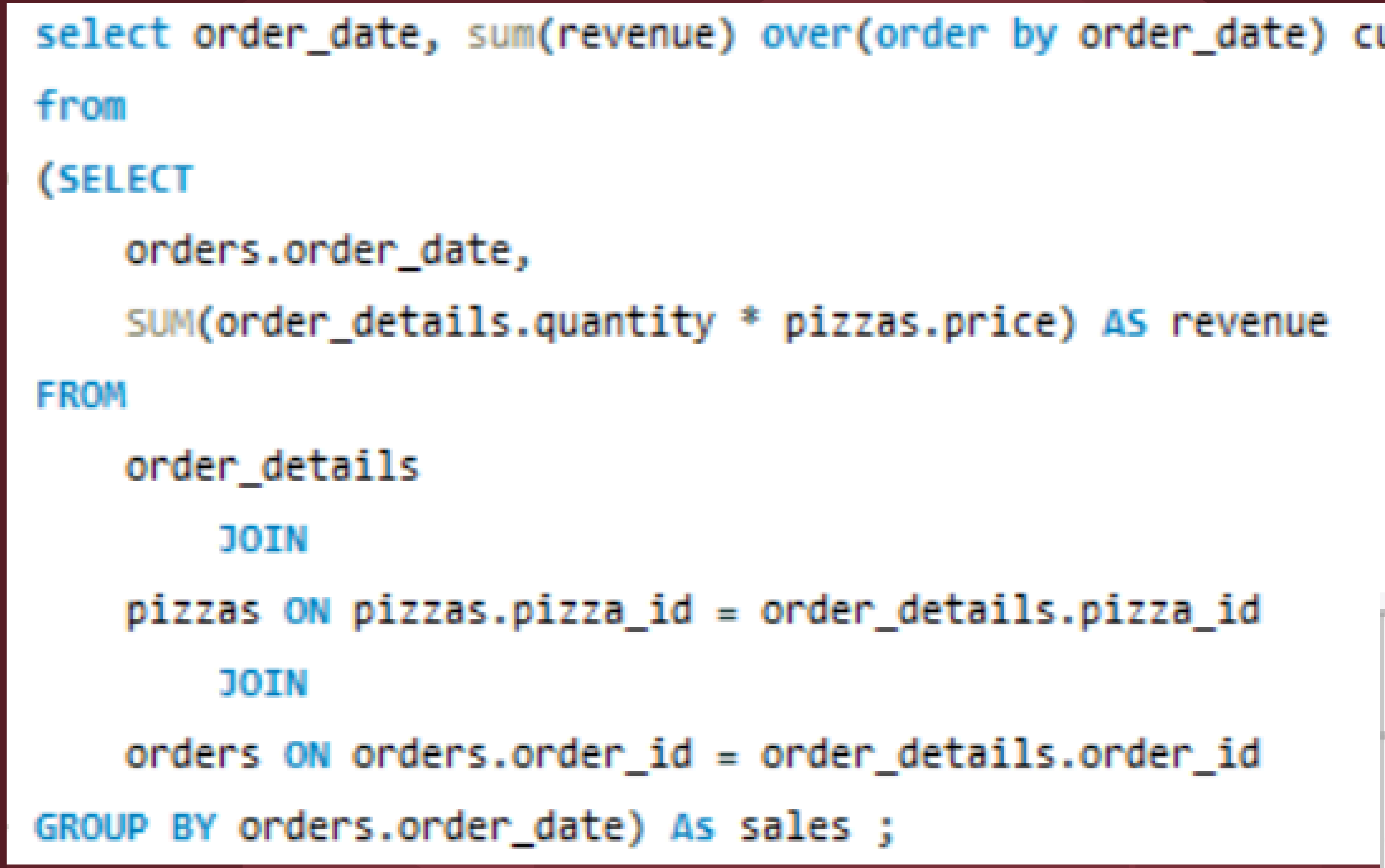


```
SELECT
    pizza_types.category as Pizza_type,
    round((SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.Pizza_id)) * 100, 2) as total_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_type
ORDER BY total_revenue DESC;
```

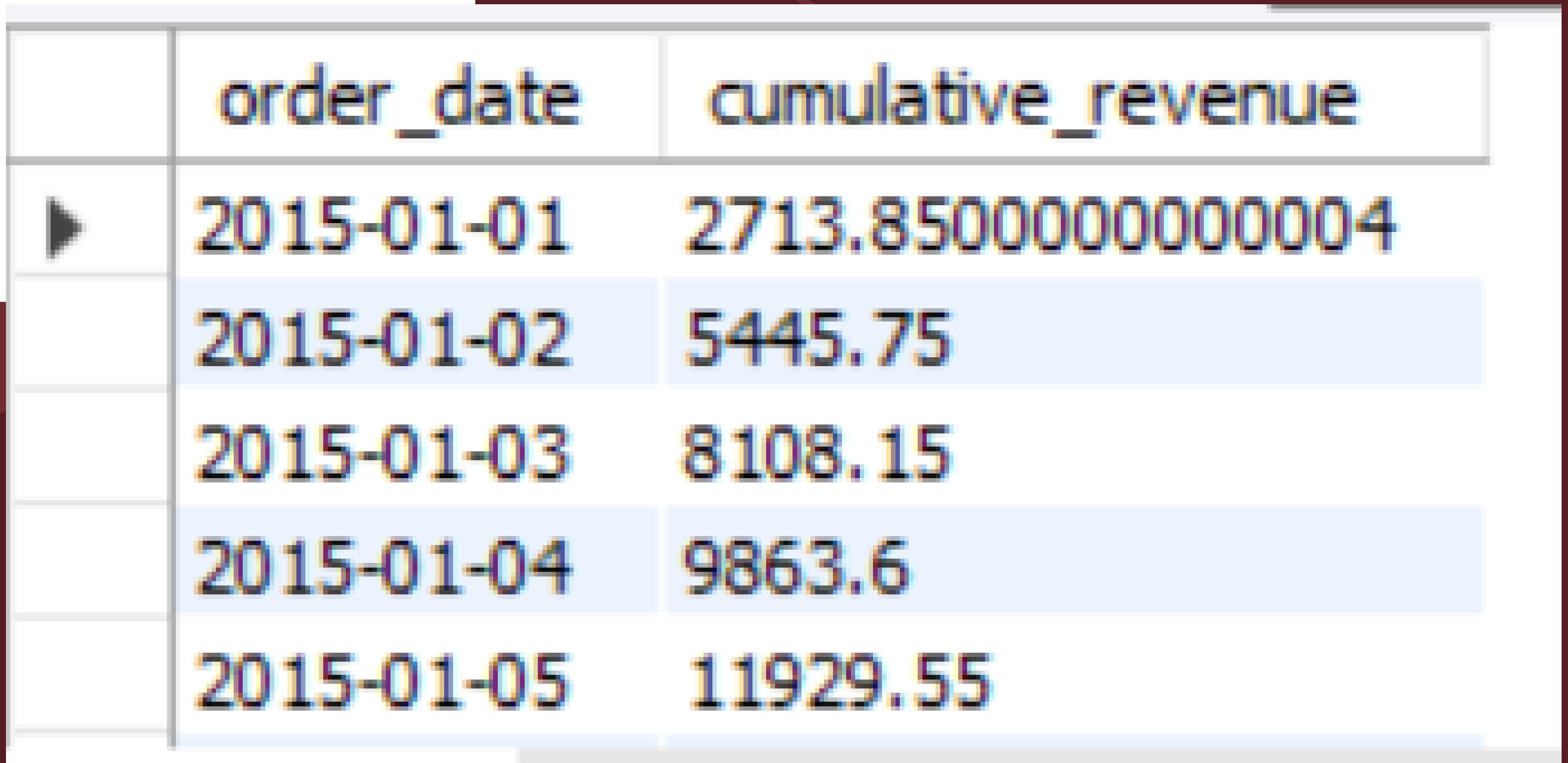


Pizza_type	total_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

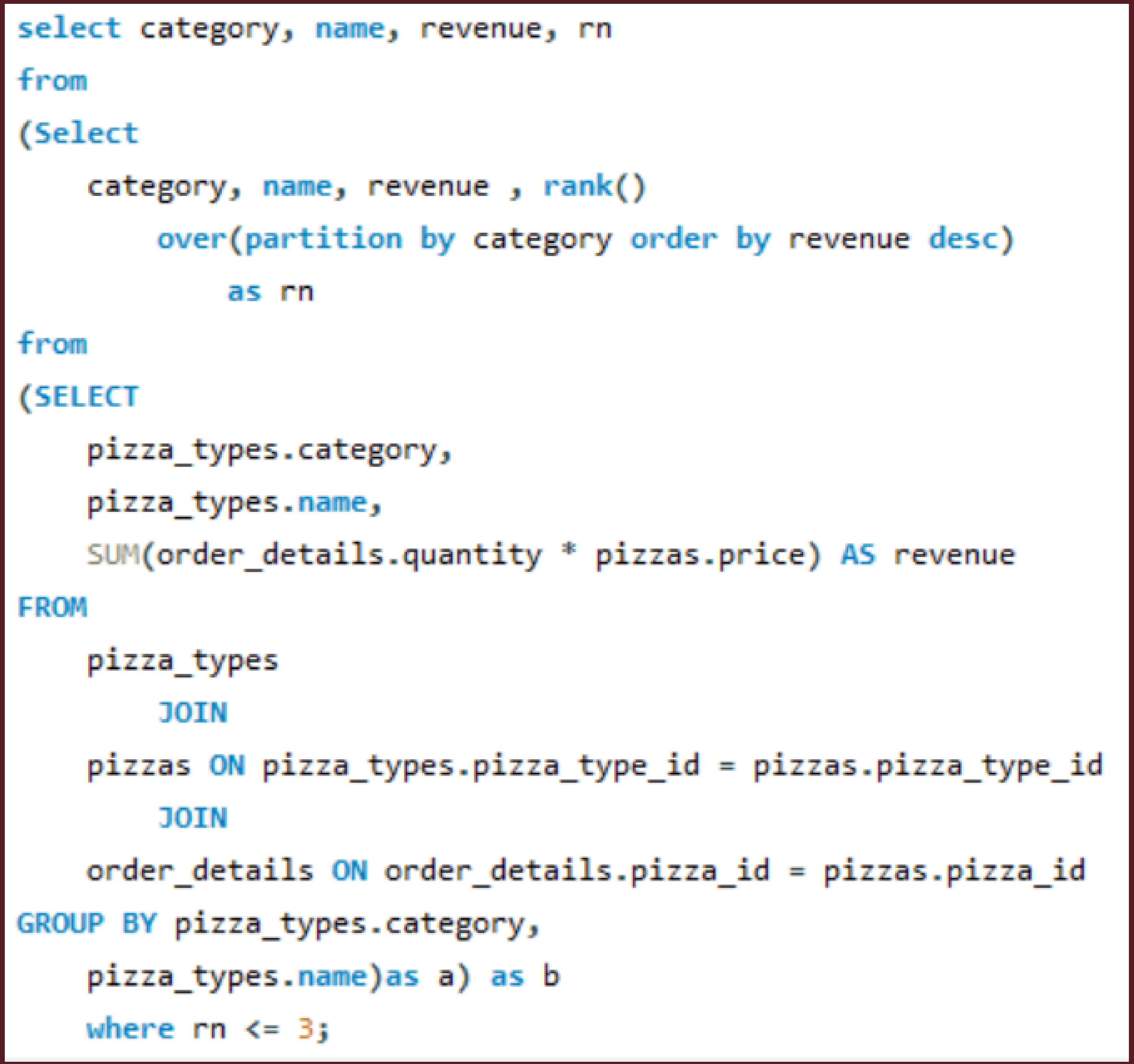


```
select order_date, sum(revenue) over(order by order_date) cumulative_revenue
from
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) As sales ;
```

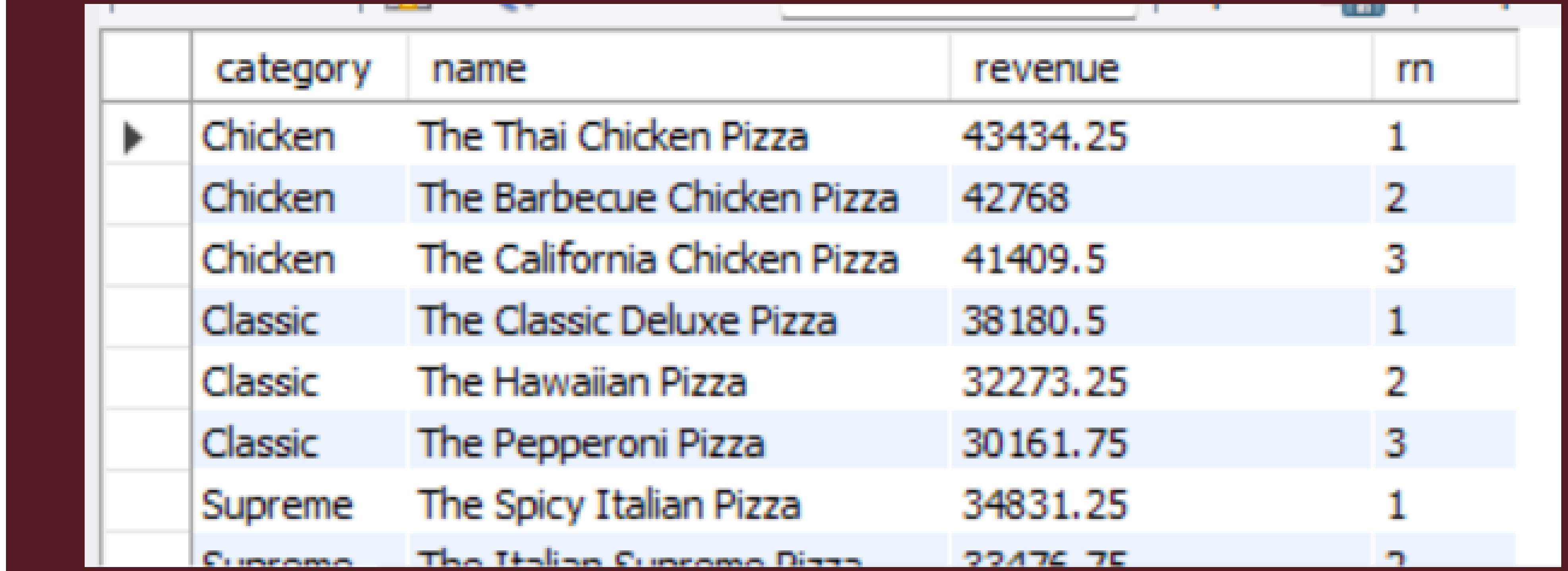


order_date	cumulative_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

13) Determine the top 3 most ordered pizza types based on revenue for each pizza category raph text



```
select category, name, revenue, rn
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;
```



category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2

Thank you so much everyone

These are some questions that i try to solve to improve the sales of pizza

The next time I will try to make a project with SQl And Power Bi