Project

PIZZAHUT SALES ANALYSIS USING SQL

Summary

Hello,my name is Deepak Kushwaha. In this project ,I have utilised SQL query to solve various problems and questions that were related to Pizza Sales.

Questions

Basic:

Q.1 Retrieve the total number of orders placed.

Q.2 Calculate the total revenue generated from pizza sales.

Q.3 Identify the highest-priced pizza.

Q.4 Identify the most common pizza size ordered.

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

Intermediate:

Q.1 Join the necessary tables to find the total quantity of each pizza category ordered.

Q.2 Determine the distribution of orders by hour of the day.

Q.3 Join relevant tables to find the category-wise distribution of pizzas.

Q.4 Group the orders by date and calculate the average number of pizzas ordered per day.

Q.5 Determine the top 3 most ordered pizza types based on revenue.

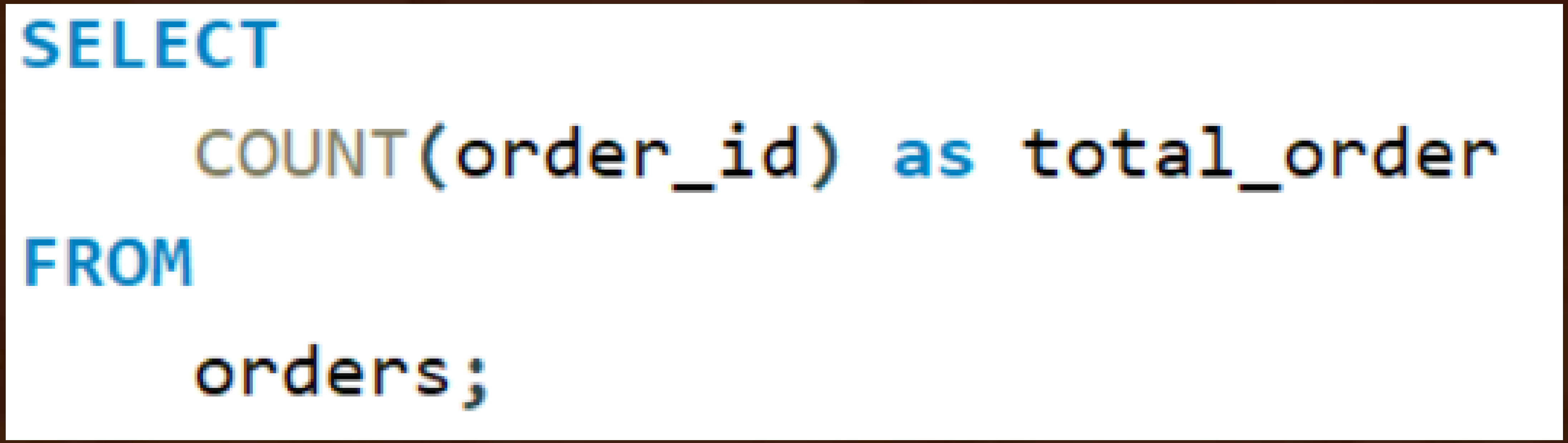
Advanced:

Q.1 Calculate the percentage contribution of each pizza type to total revenue.

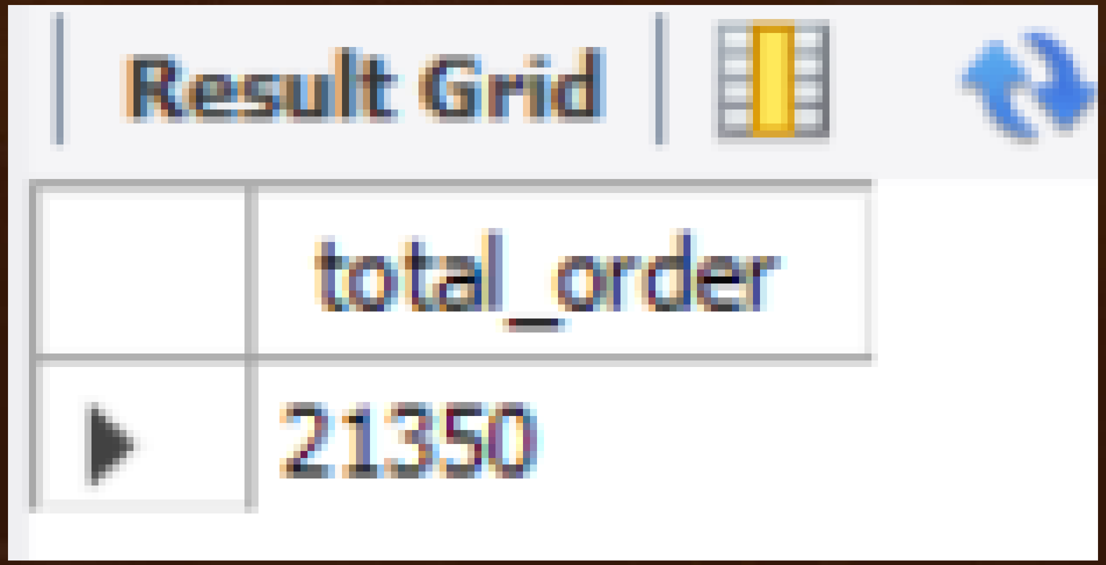
Q.2 Analyze the cumulative revenue generated over time.

Q.3 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Q-1. Retrieve the total number of orders placed.

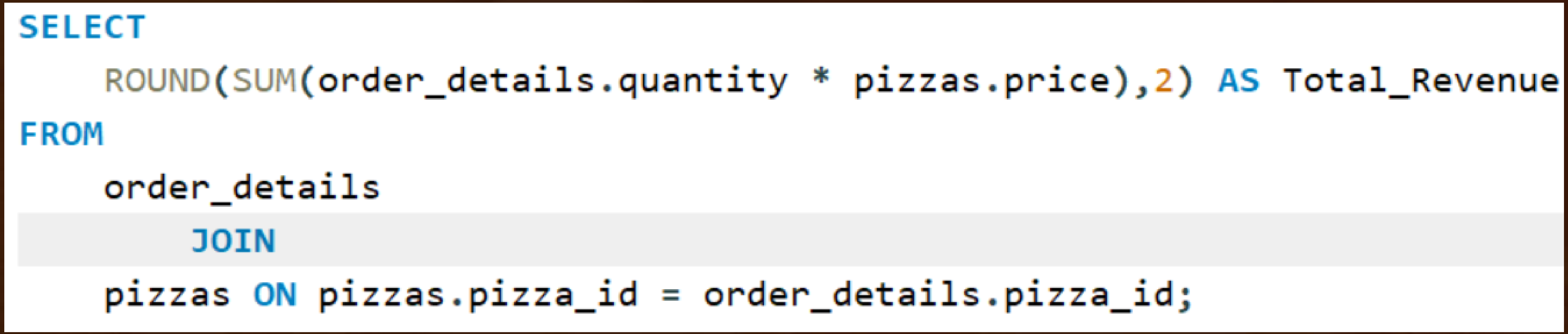


```
SELECT
    COUNT(order_id) as total_order
FROM
    orders;
```

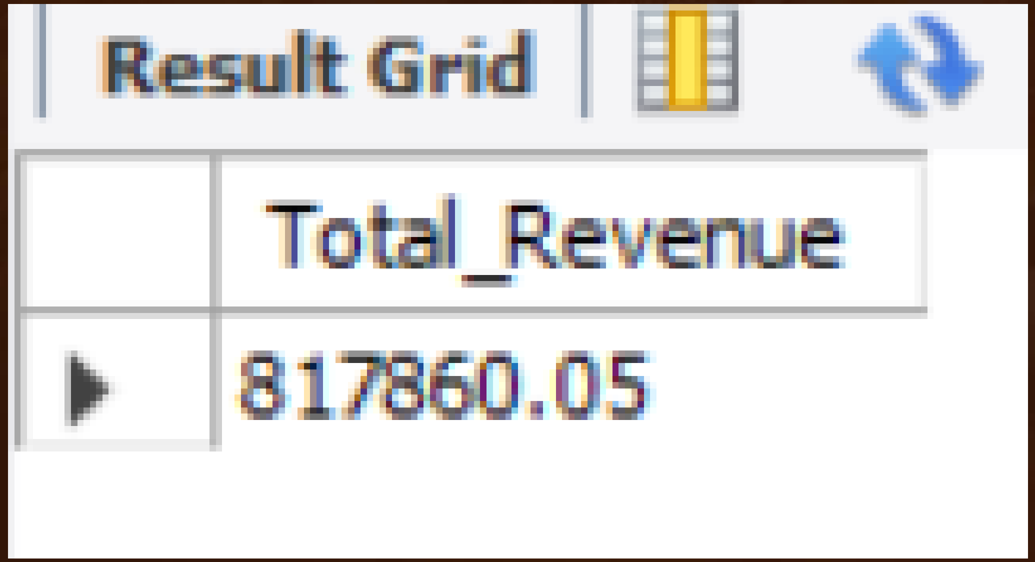


```
total_order
21350
```

Q-2. Calculate the total revenue generated from pizza sales.

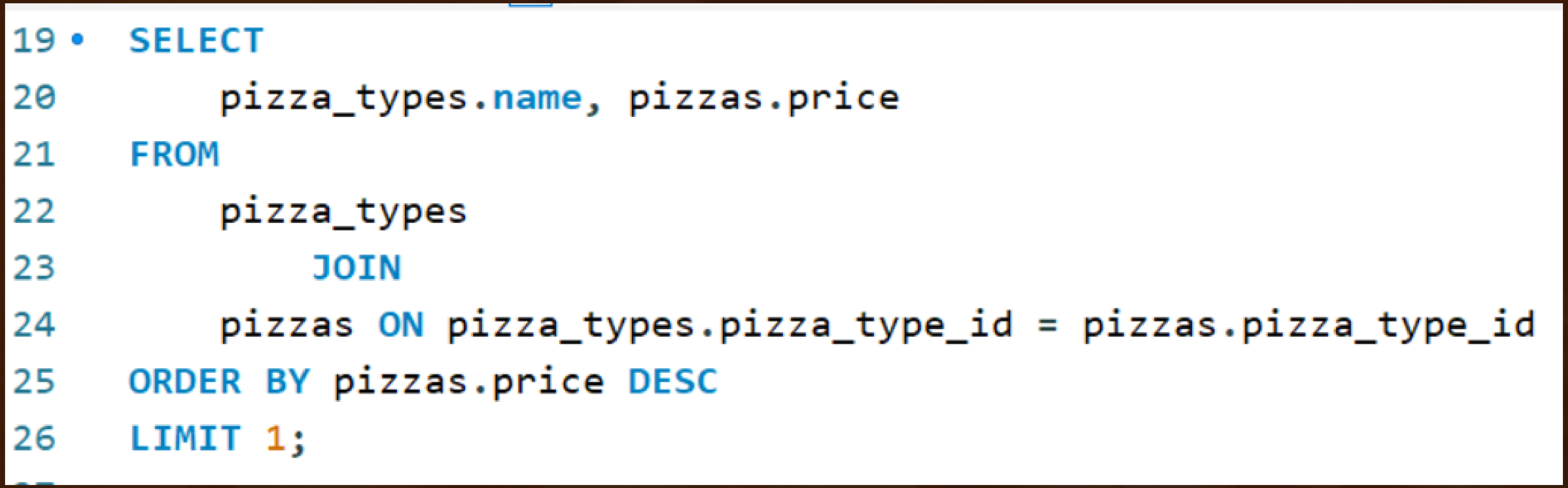


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_Revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

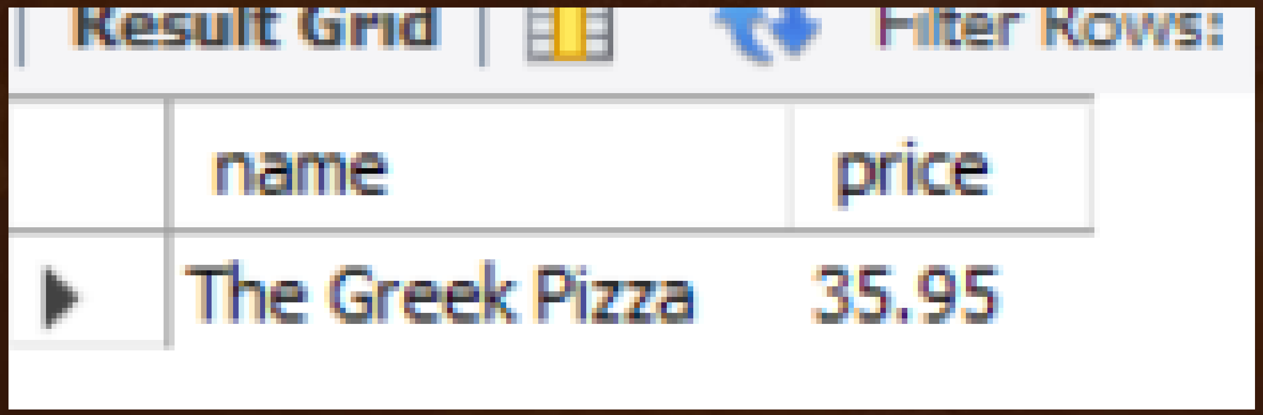


	Total_Revenue
▶	817860.05

Q-3. Identify the highest-priced pizza.

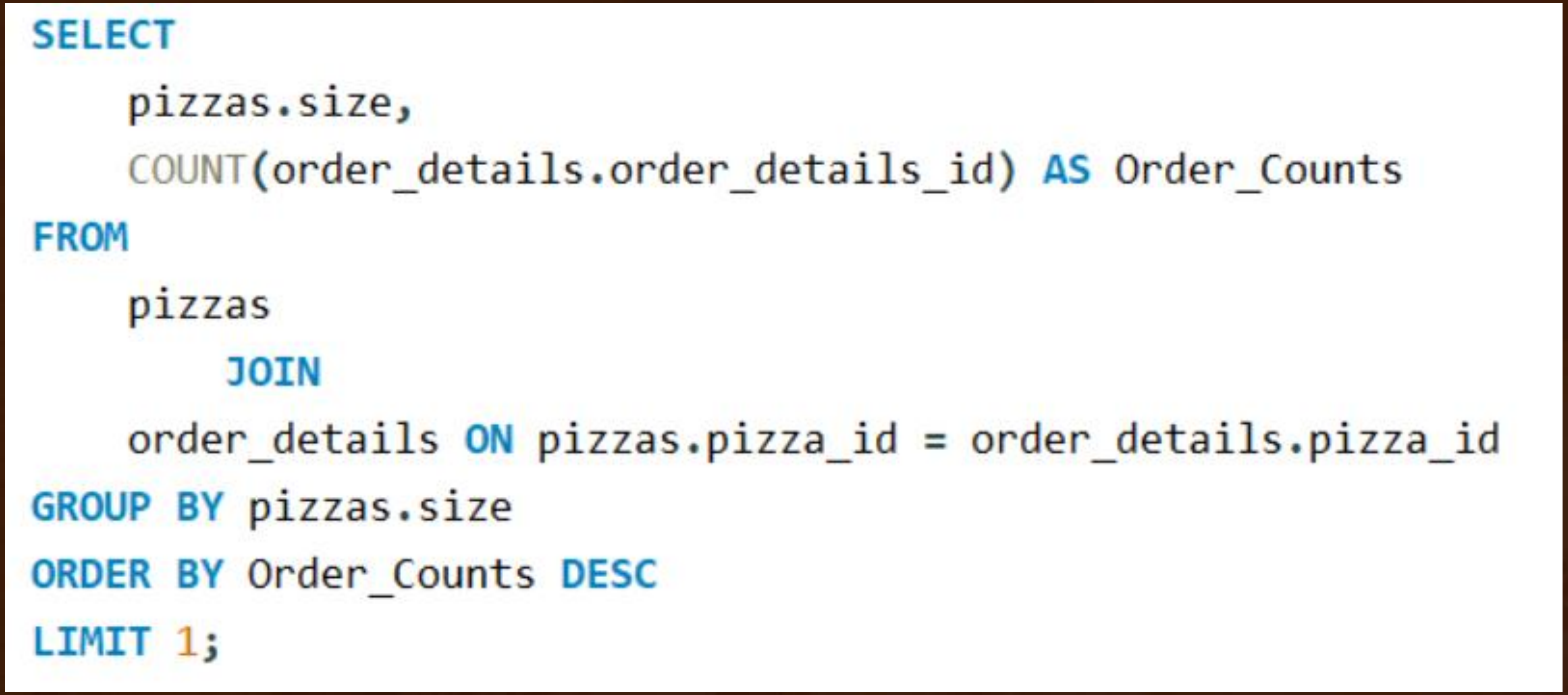


```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

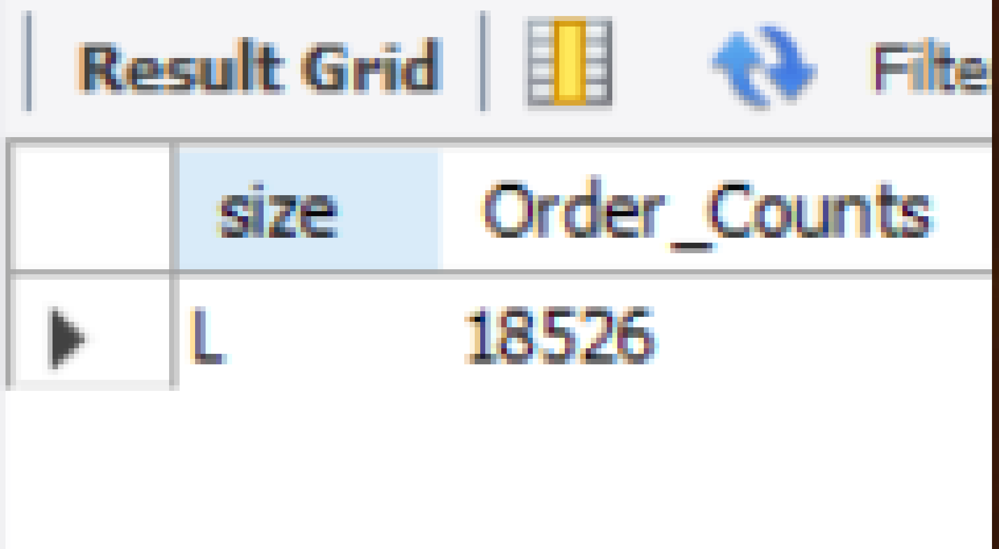


name	price
The Greek Pizza	35.95

Q-4. Identify the most common pizza size ordered.

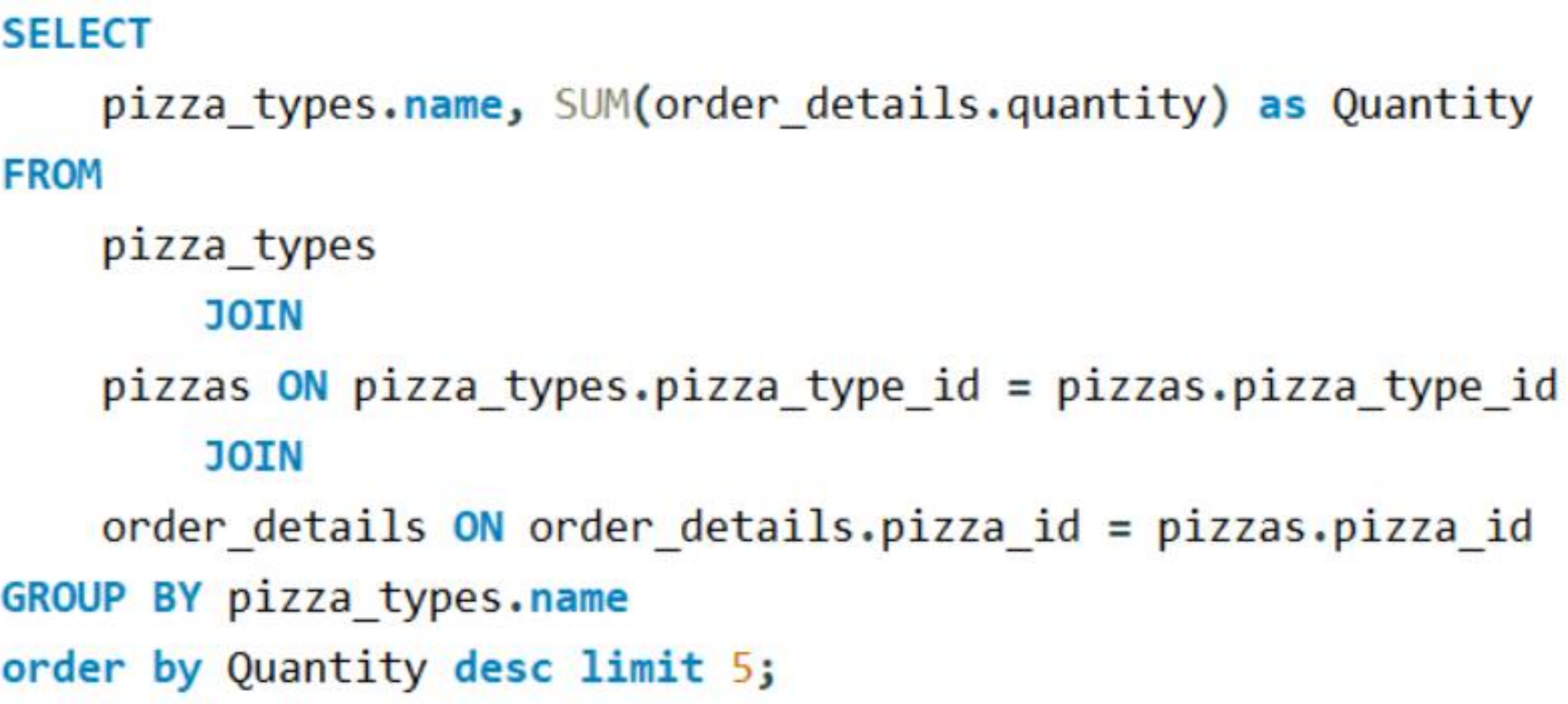


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Order_Counts
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Order_Counts DESC
LIMIT 1;
```

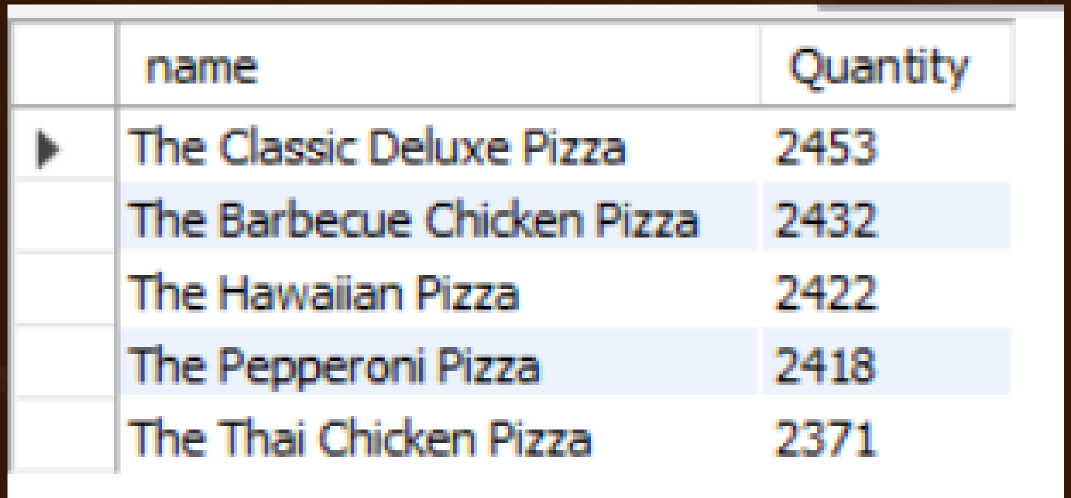


```
  size  Order_Counts
▶ L     18526
```

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

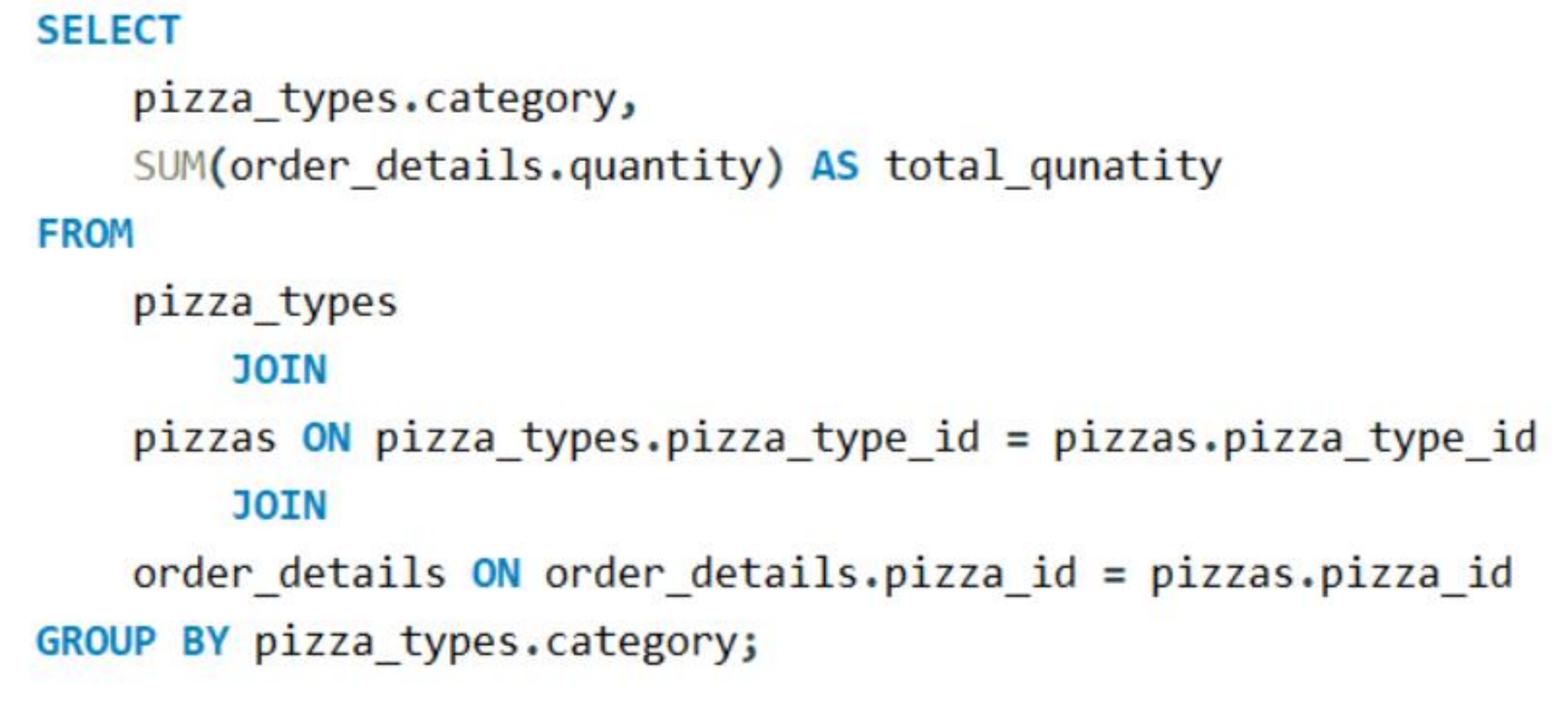


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) as Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
order by Quantity desc limit 5;
```

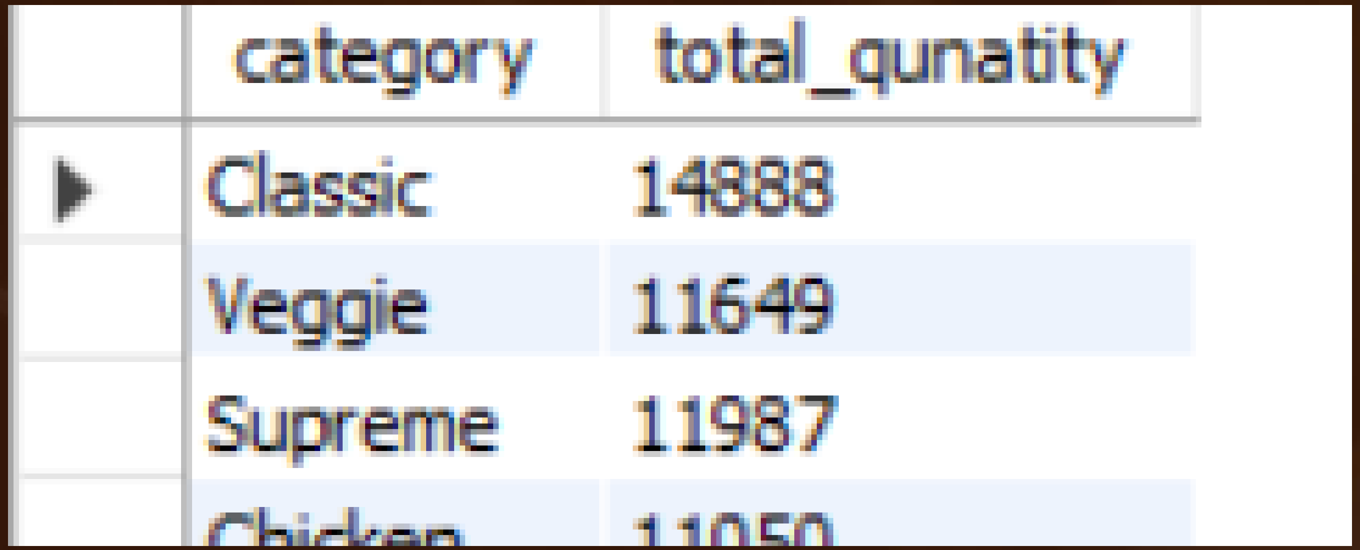


name	Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

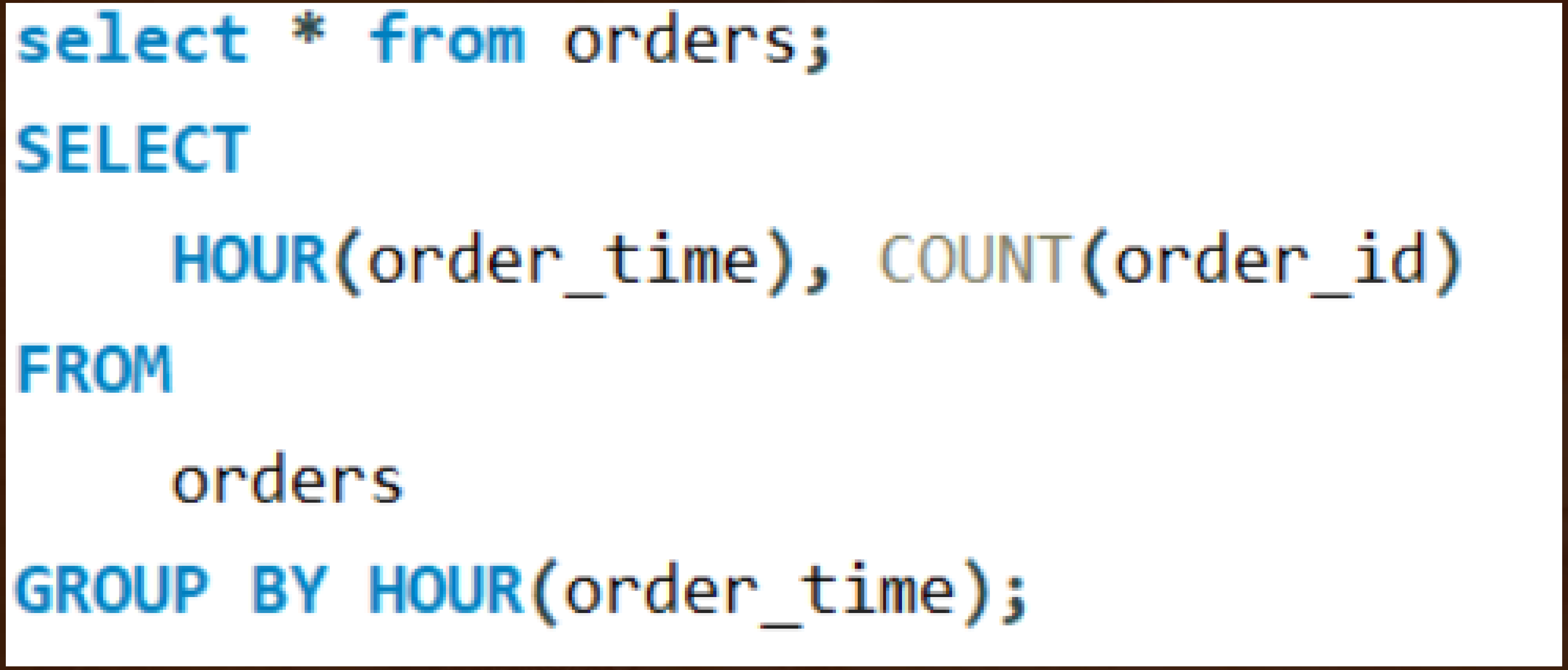


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_qunatity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

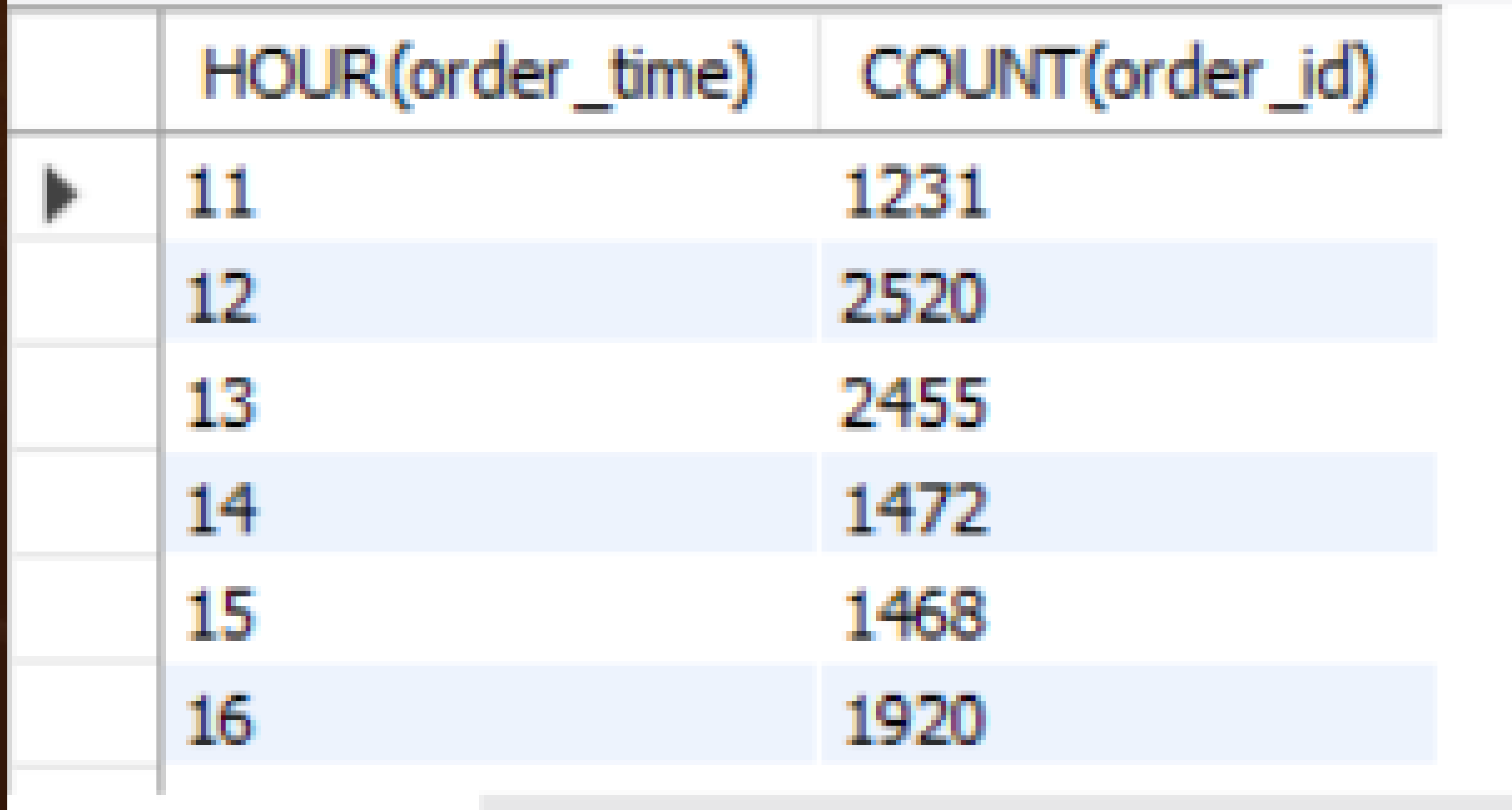


category	total_qunatity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

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

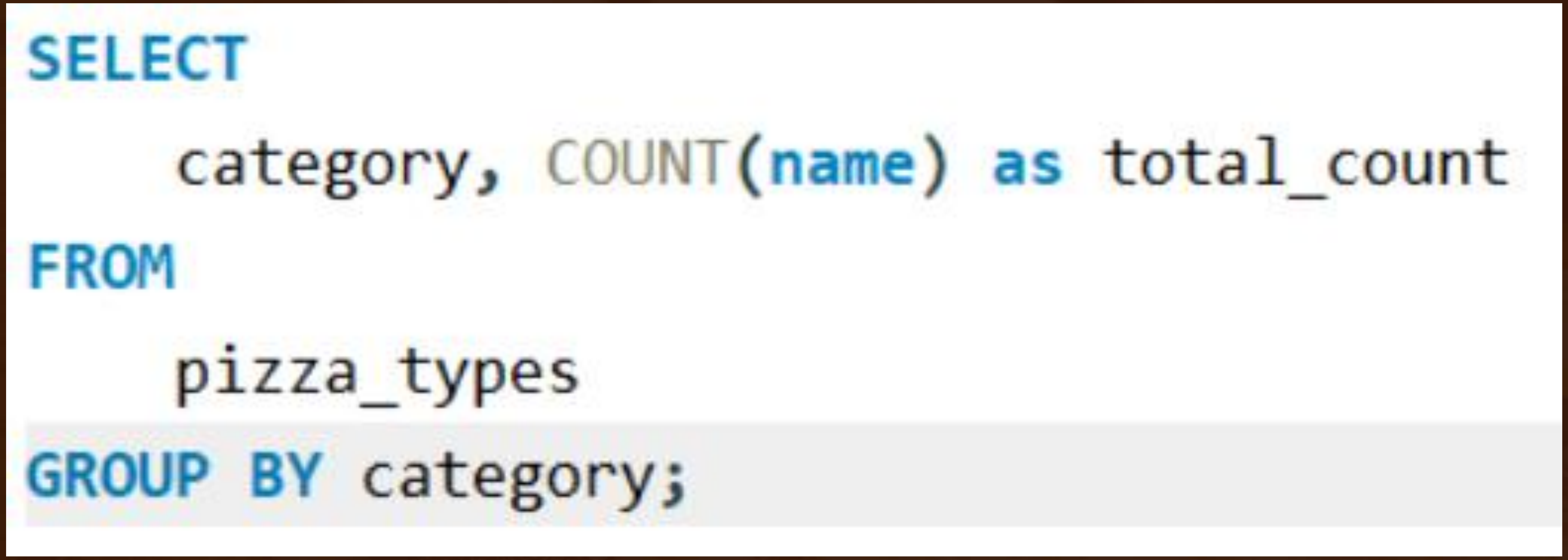


```
select * from orders;
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

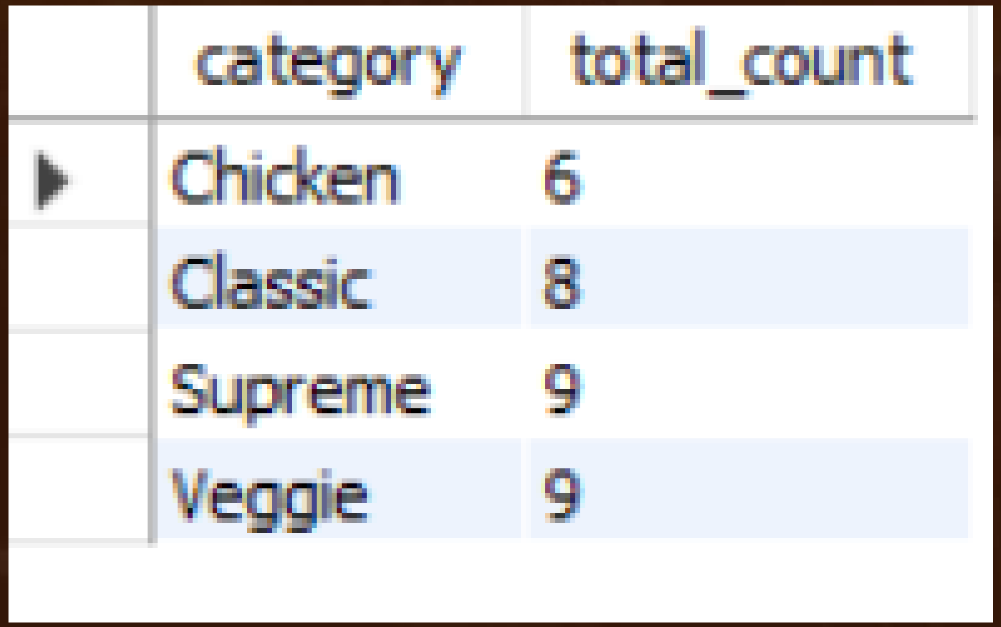


HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

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

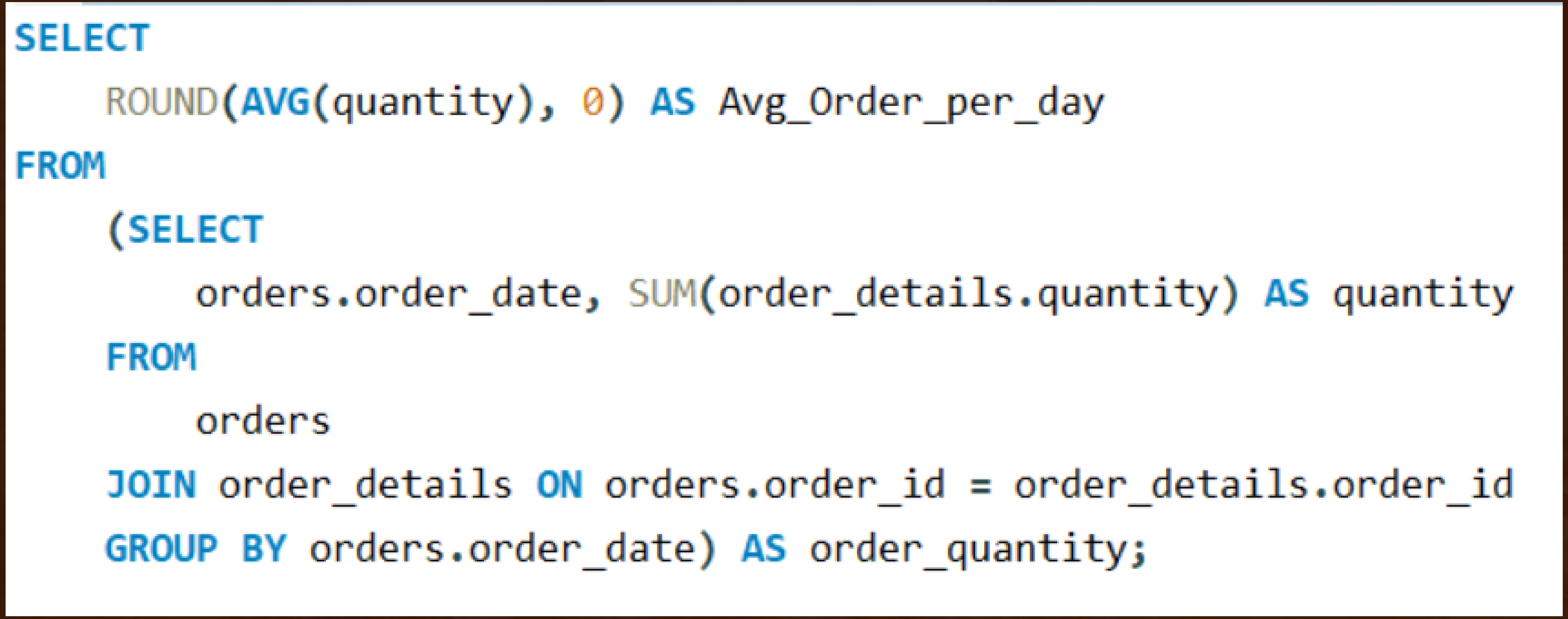


```
SELECT
    category, COUNT(name) as total_count
FROM
    pizza_types
GROUP BY category;
```

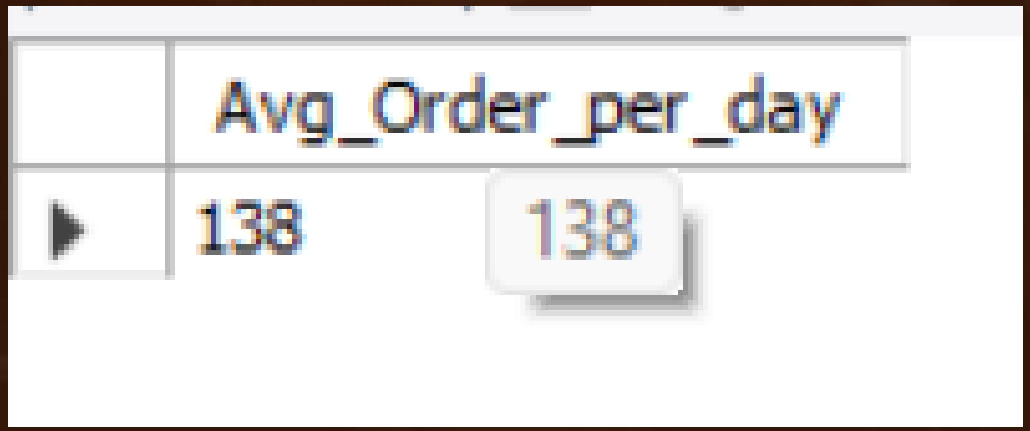


category	total_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

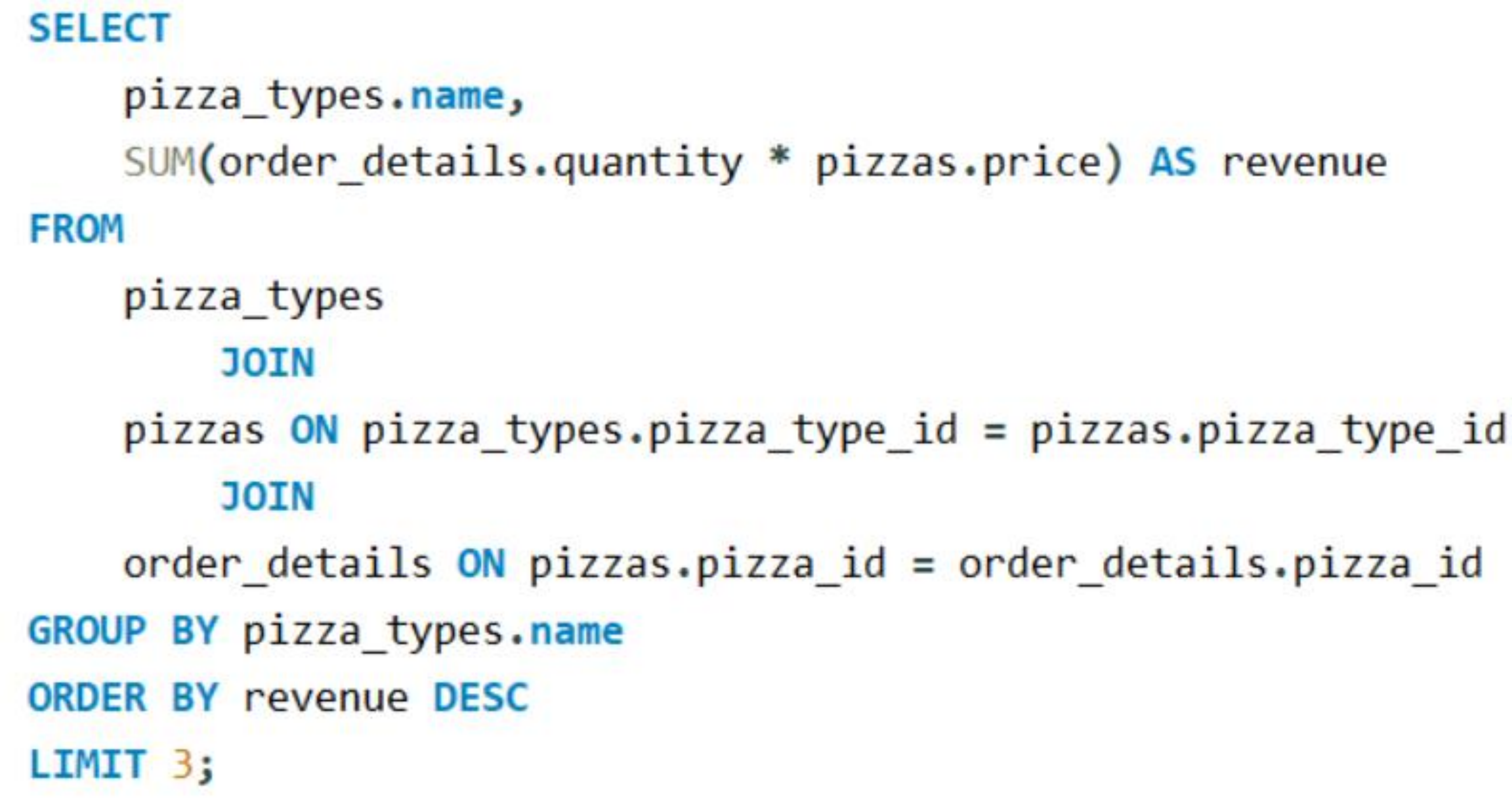


```
SELECT
    ROUND(AVG(quantity), 0) AS Avg_Order_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

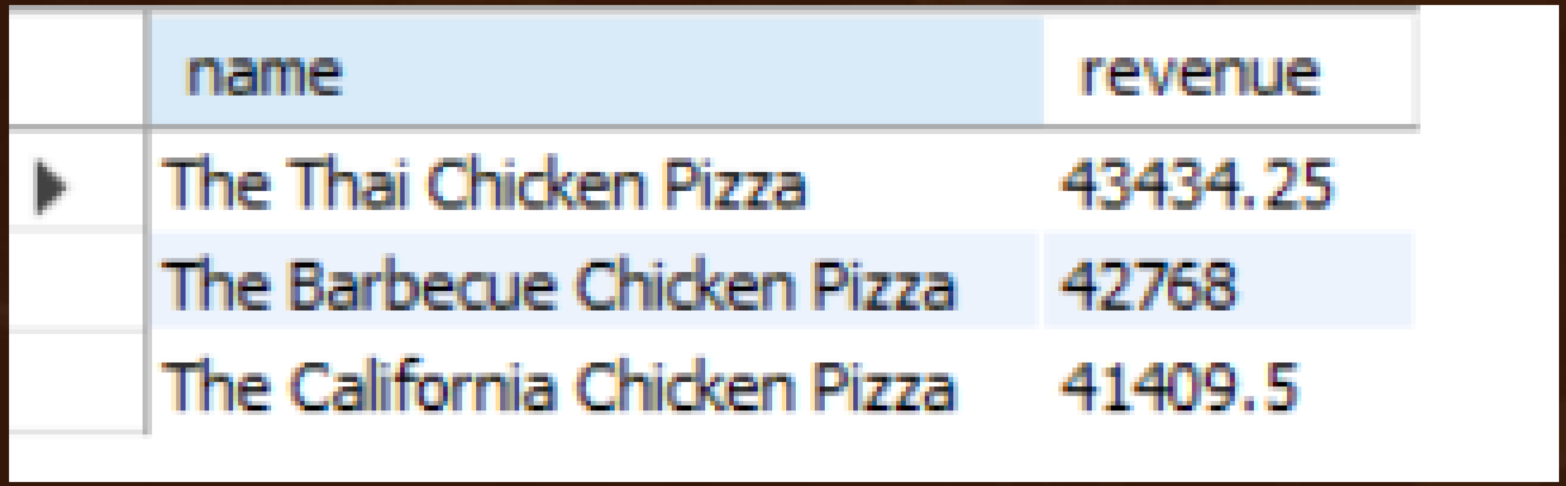


Avg_Order_per_day
138

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

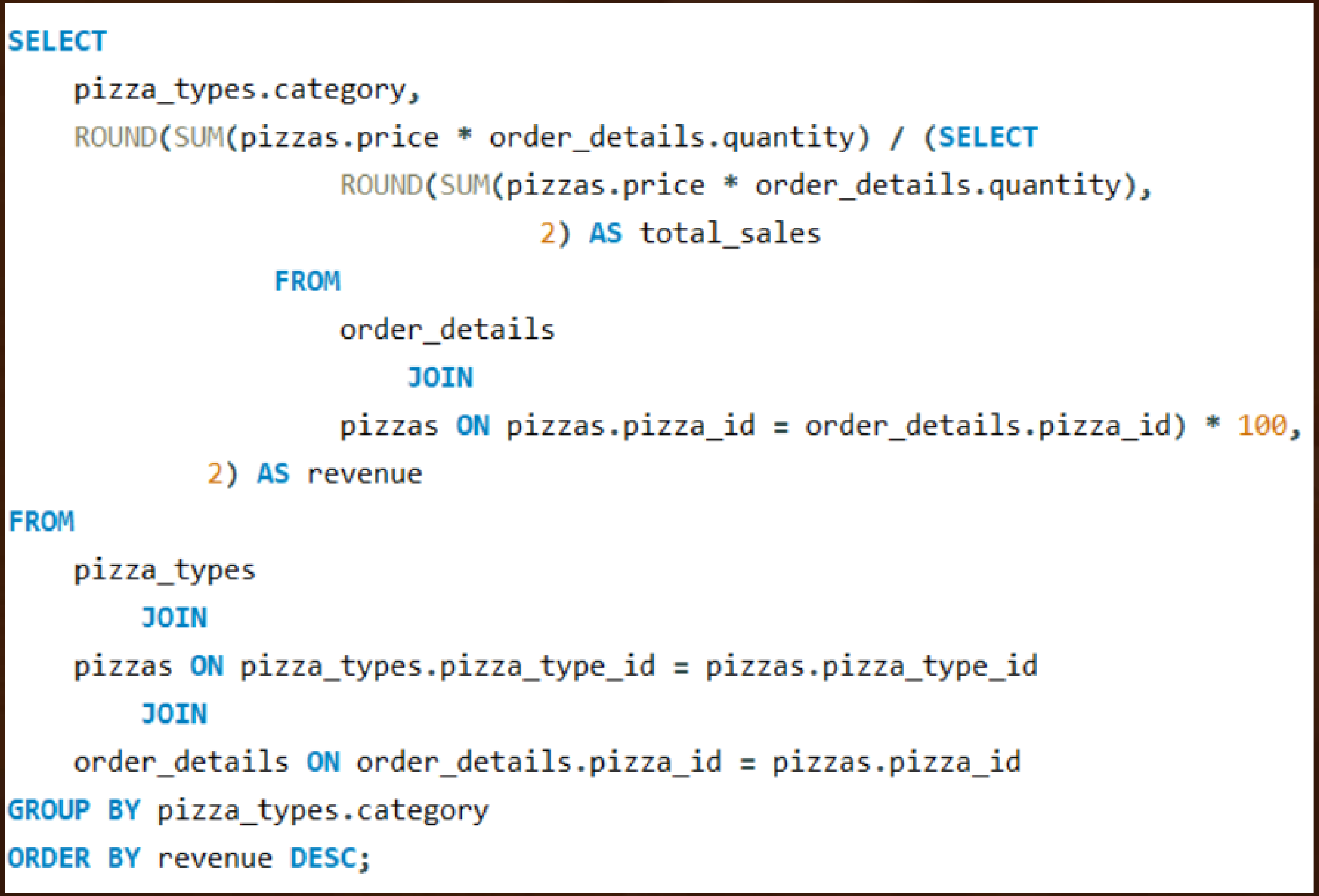


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

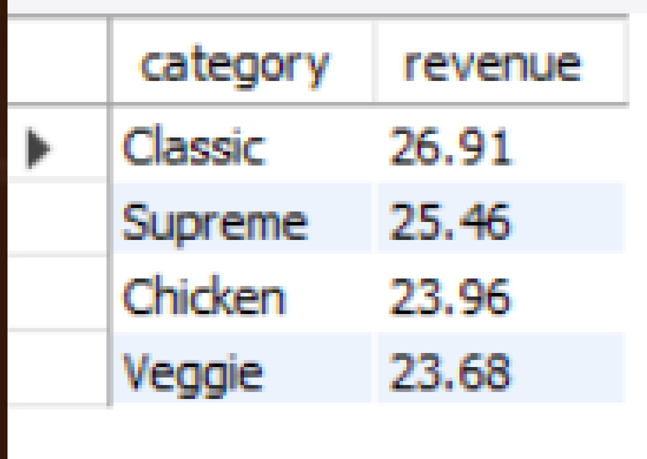


	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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

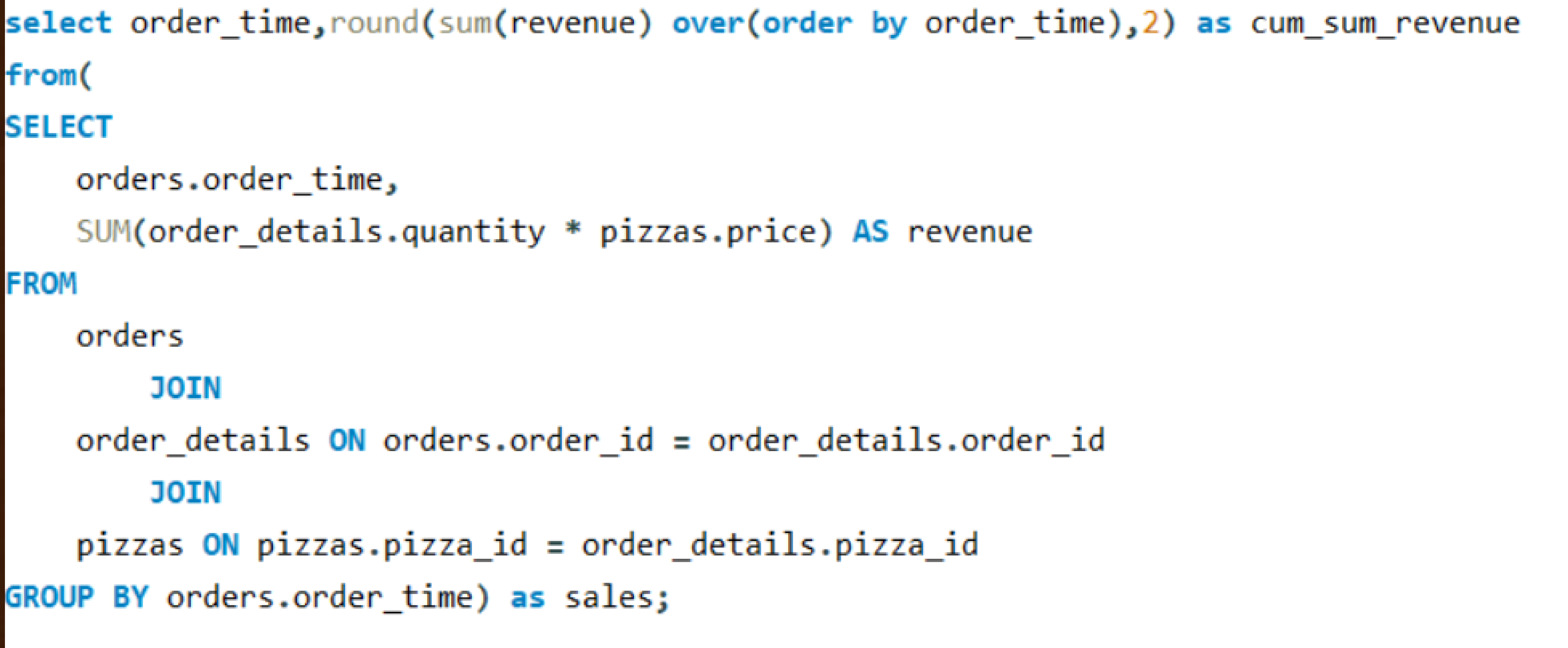


```
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                    ROUND(SUM(pizzas.price * order_details.quantity),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    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;
```

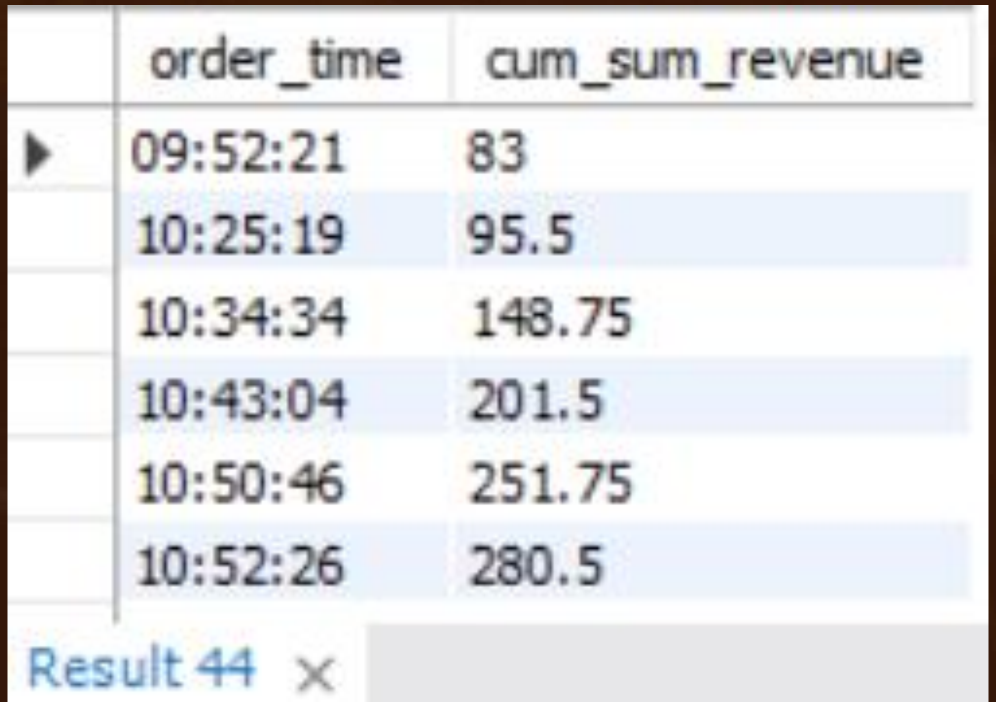


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q-12. Analyze the cumulative revenue generated over time.

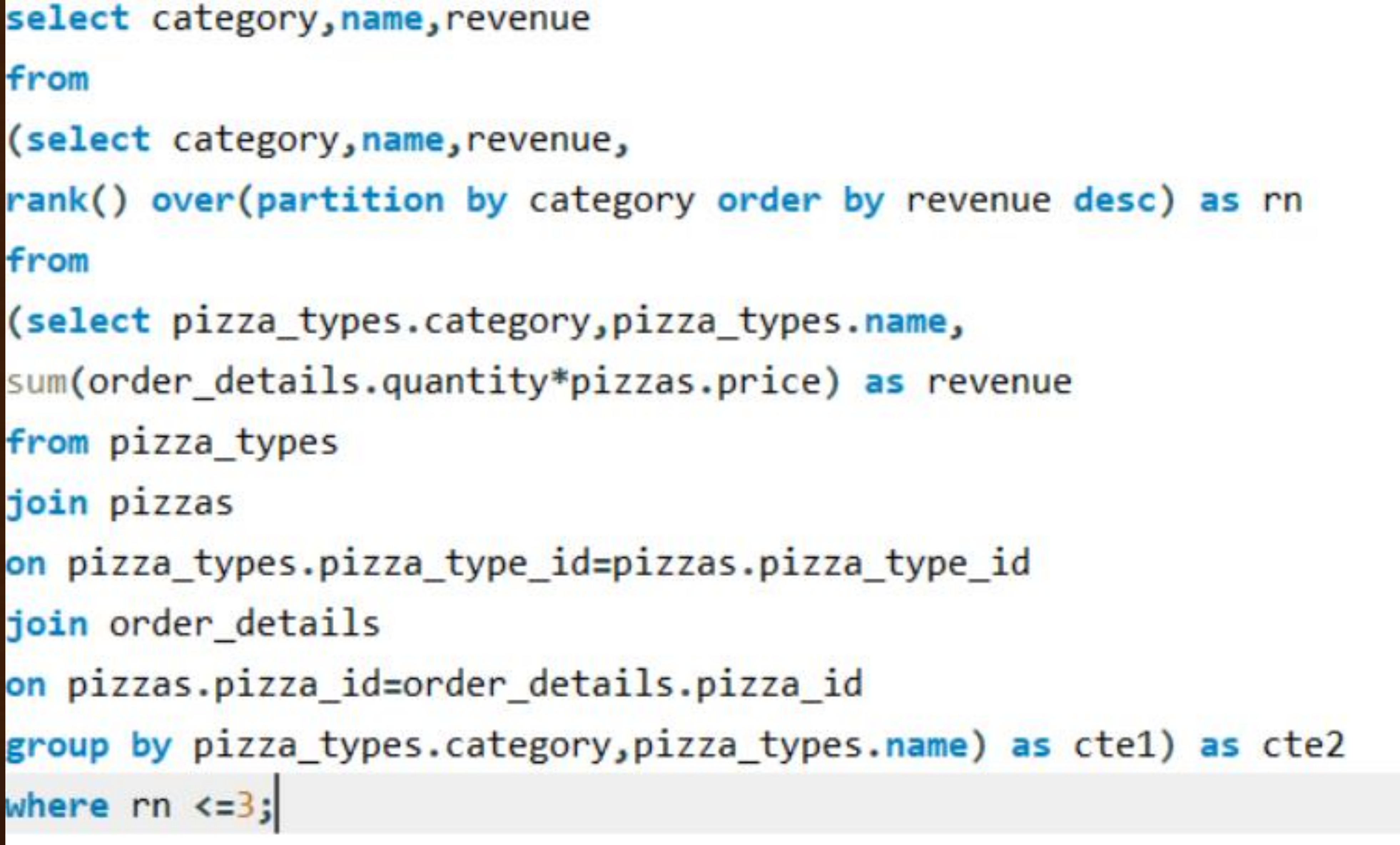


```
select order_time,round(sum(revenue) over(order by order_time),2) as cum_sum_revenue
from(
SELECT
    orders.order_time,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY orders.order_time) as sales;
```

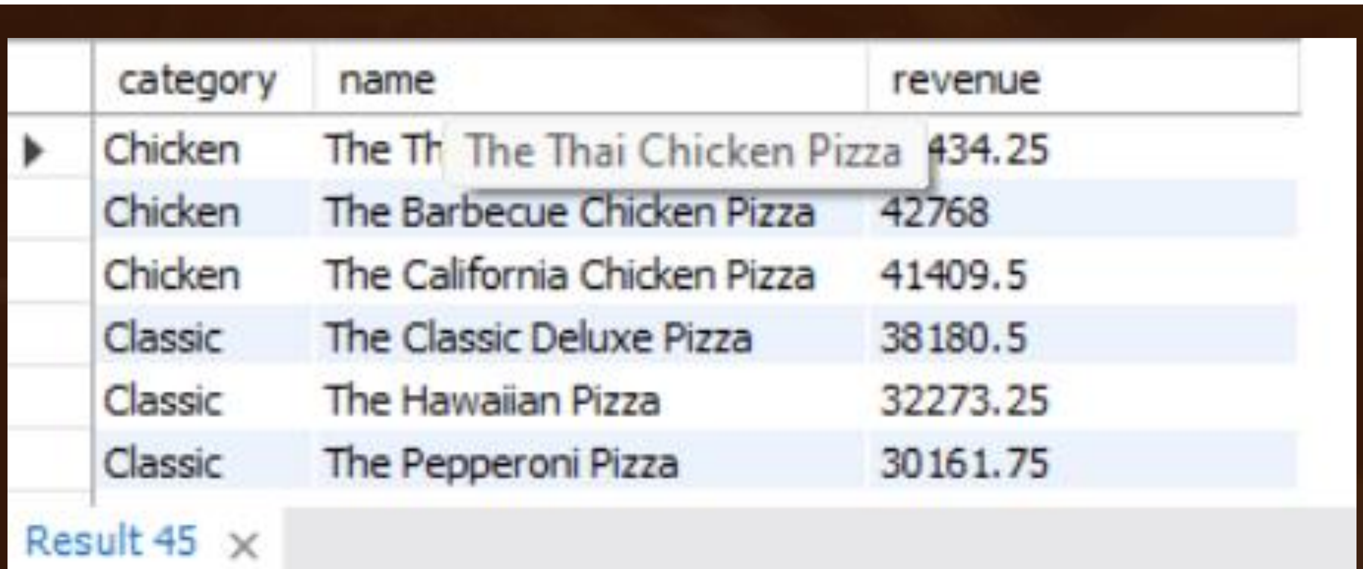


order_time	cum_sum_revenue
09:52:21	83
10:25:19	95.5
10:34:34	148.75
10:43:04	201.5
10:50:46	251.75
10:52:26	280.5

Q-13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```sql
select category,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 pizzas.pizza_id=order_details.pizza_id
group by pizza_types.category,pizza_types.name) as cte1) as cte2
where rn <=3;
```



category	name	revenue
Chicken	The Th The Thai Chicken Pizza	434.25
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

THANK YOU