PIZZA SALES ANALYSIS

BY

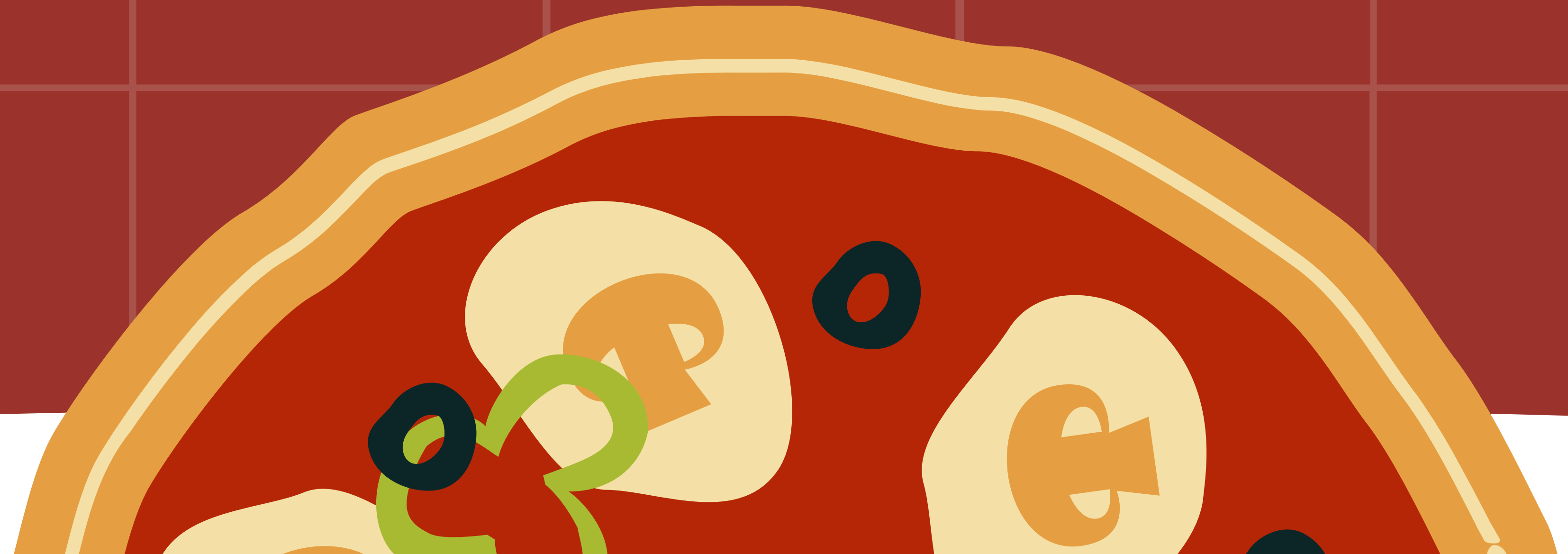
VISMAYA K K

HELLO!

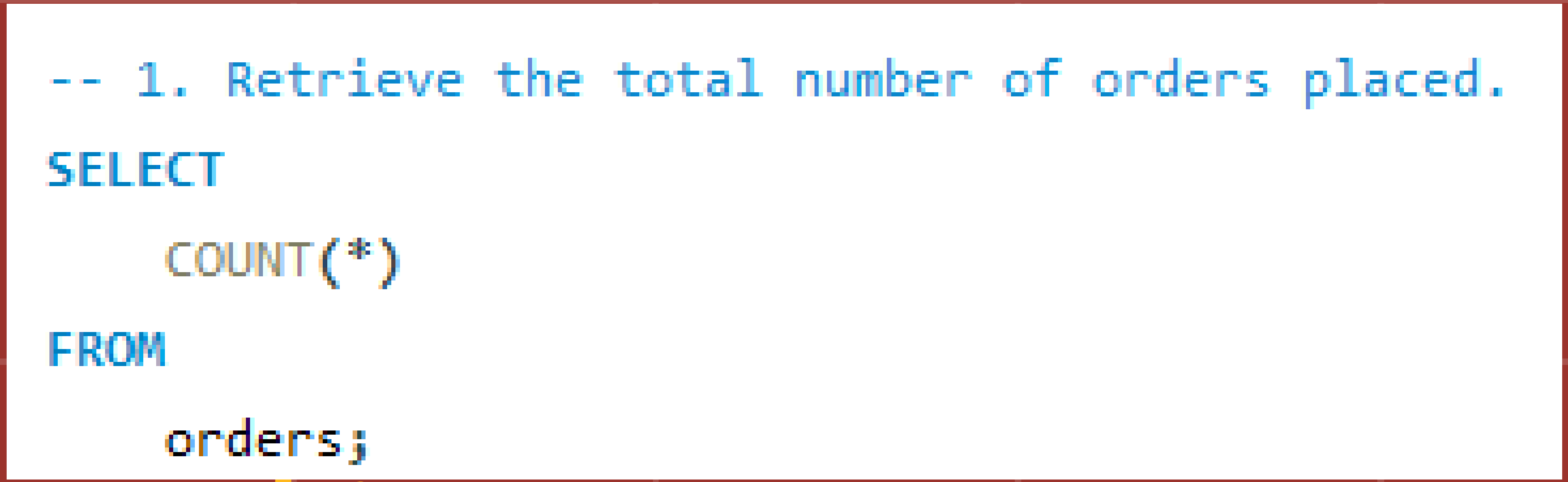
In this project i have utilized SQL to solve all questions that related to Pizza Sales

OBJECTIVES

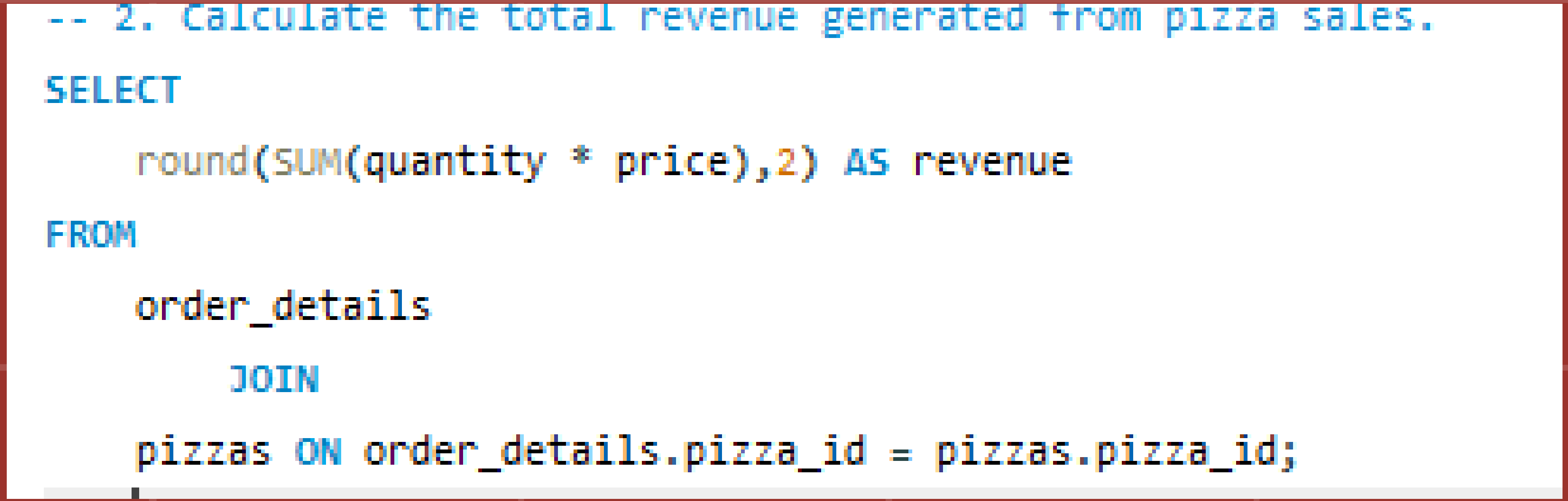
- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total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



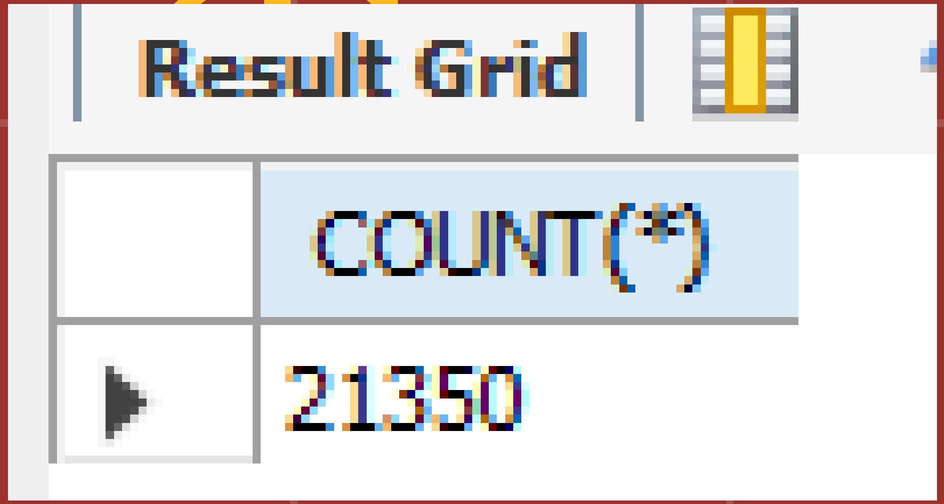
TOTAL NUMBER OF ORDERS PLACED AND TOTAL REVENUE



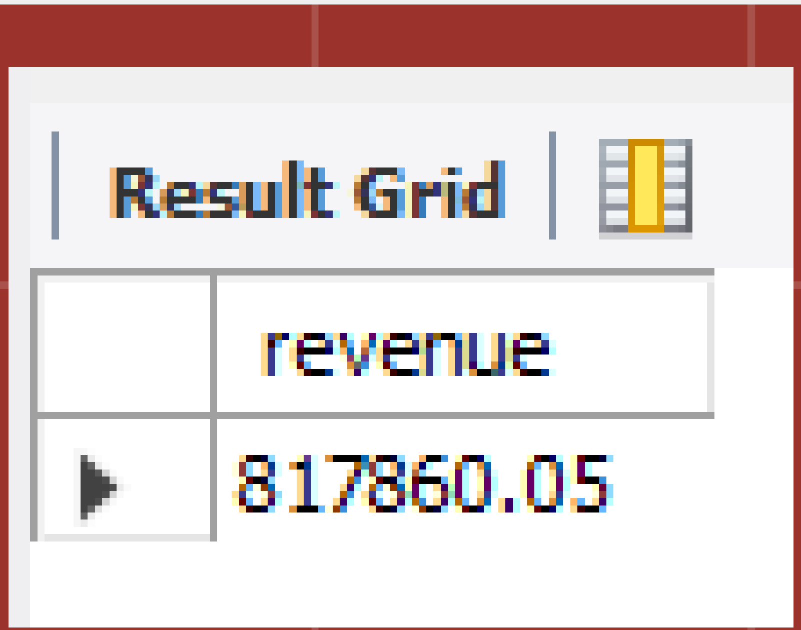
```
-- 1. Retrieve the total number of orders placed.
SELECT
    COUNT(*)
FROM
    orders;
```



```
-- 2. Calculate the total revenue generated from pizza sales.
SELECT
    round(SUM(quantity * price),2) AS revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

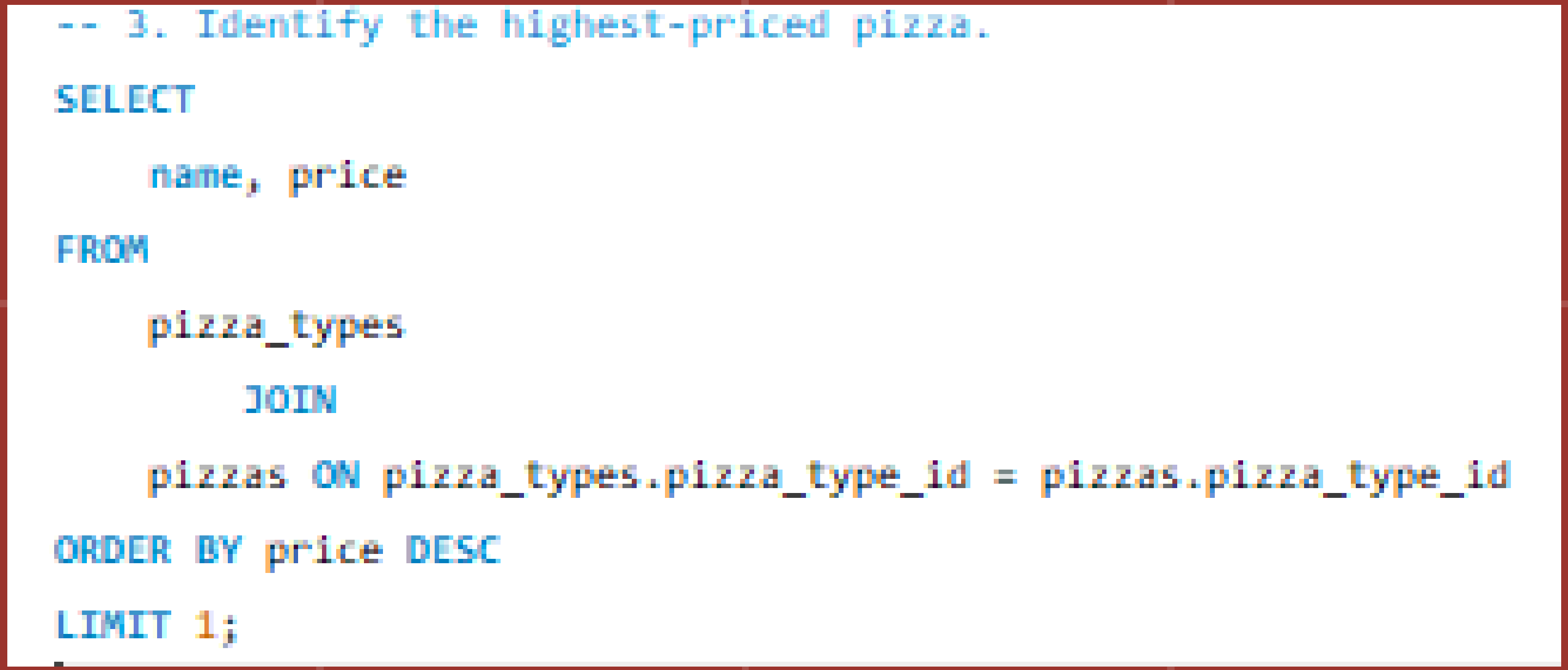


	COUNT(*)
▶	21350

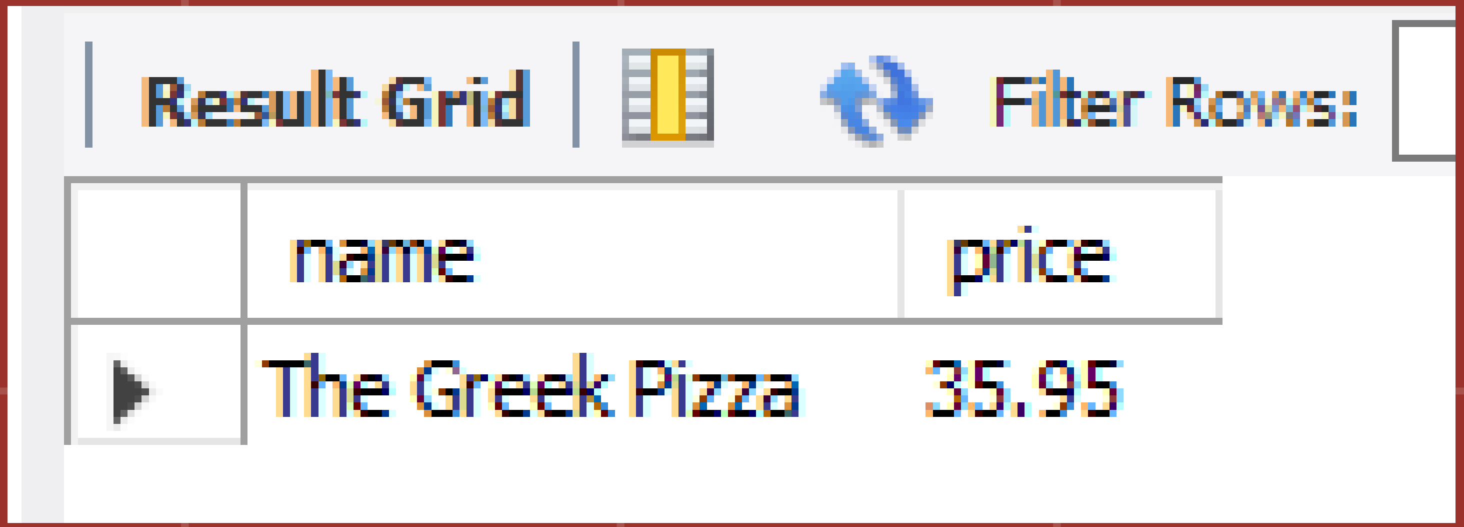


	revenue
▶	817860.05

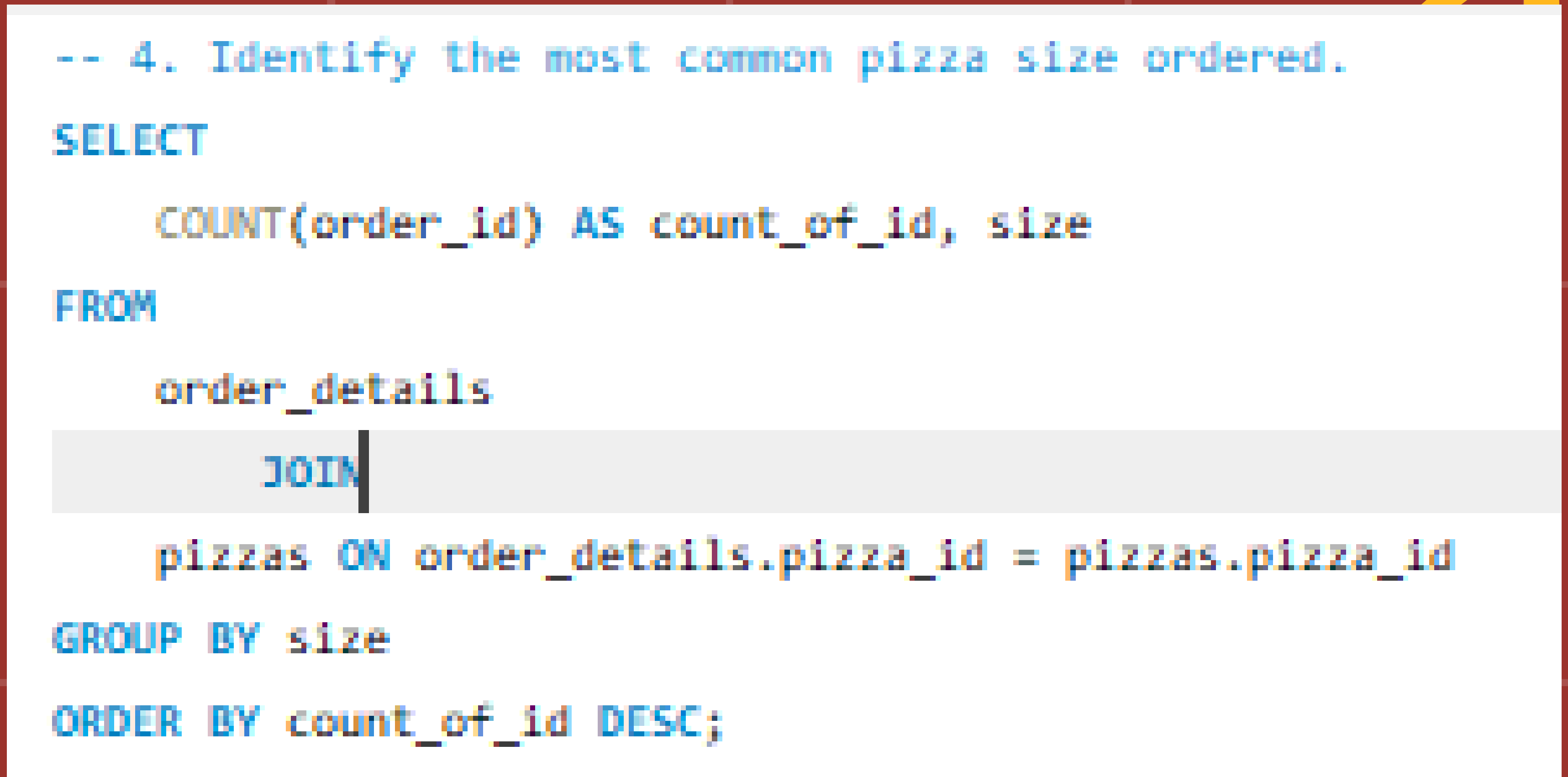
HIGHEST PRICED PIZZA AND MOST COMMON PIZZA SIZE ORDERD



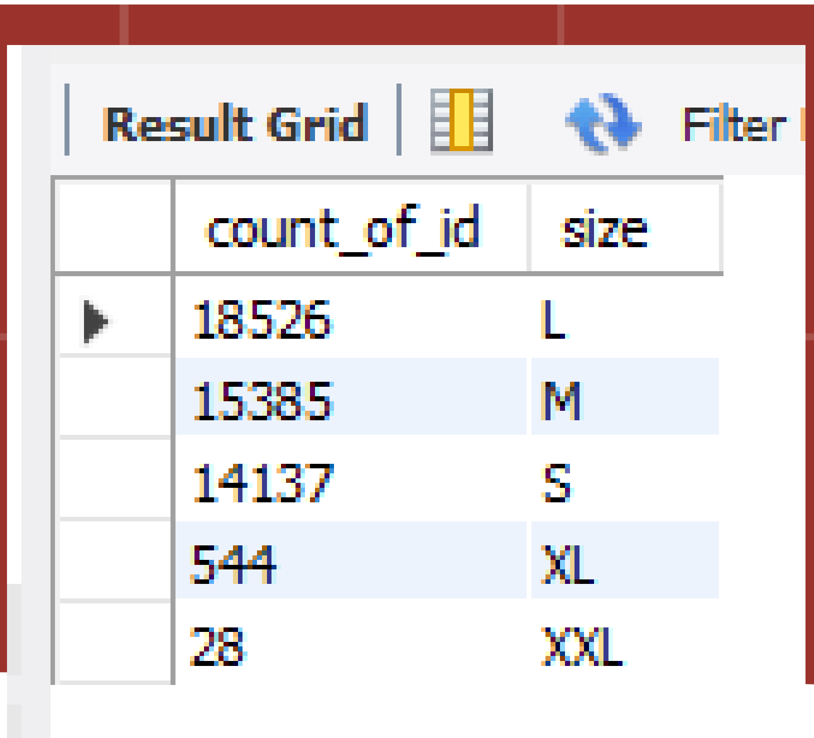
```
-- 3. Identify the highest-priced pizza.
SELECT
    name, price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```



name	price
The Greek Pizza	35.95

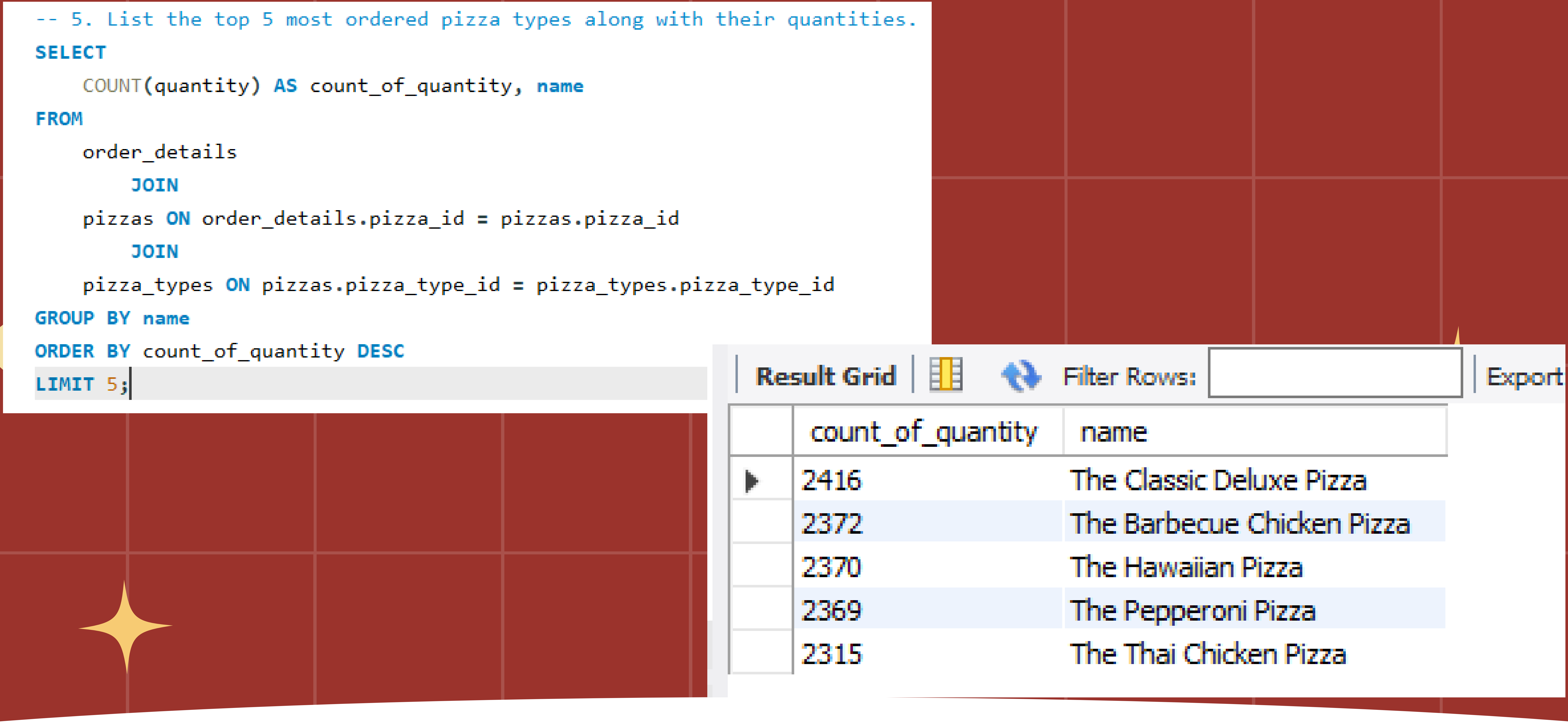


```
-- 4. Identify the most common pizza size ordered.
SELECT
    COUNT(order_id) AS count_of_id, size
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY size
ORDER BY count_of_id DESC;
```



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

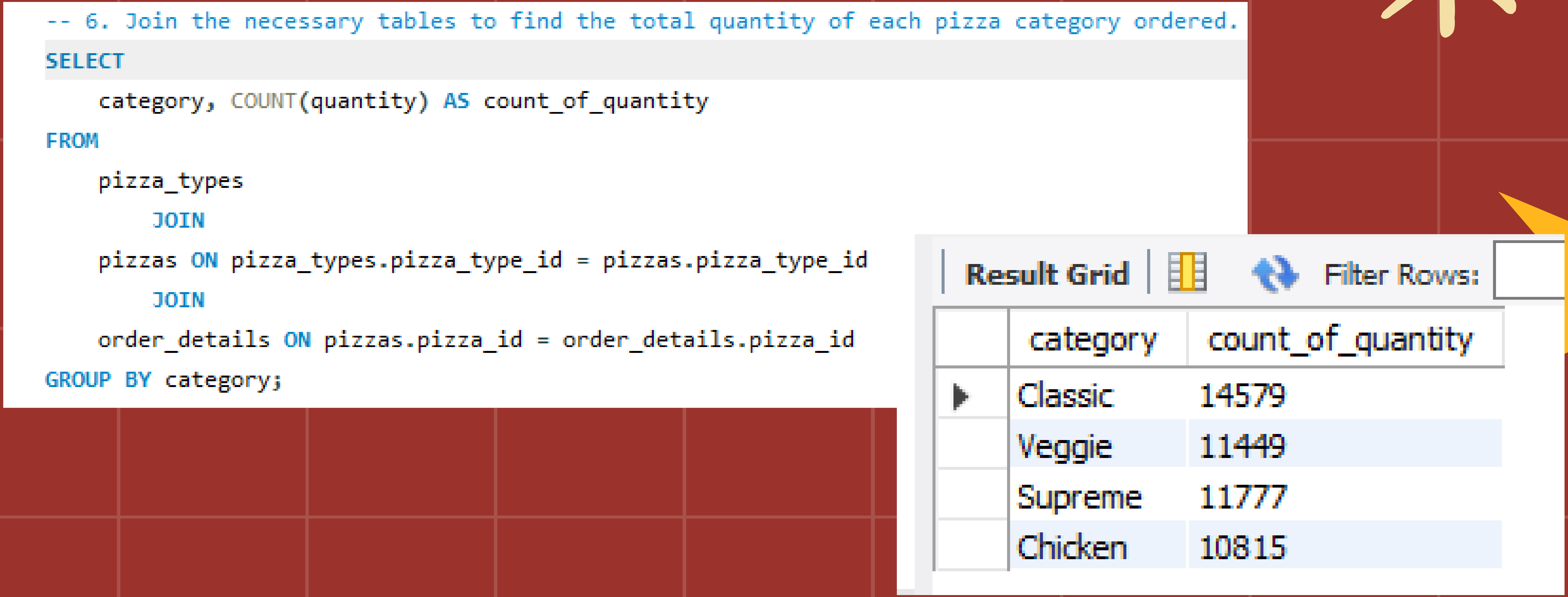
TOP 5 MOST ORDERD PIZZA



```sql
-- 5. List the top 5 most ordered pizza types along with their quantities.
SELECT
    COUNT(quantity) AS count_of_quantity, name
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY name
ORDER BY count_of_quantity DESC
LIMIT 5;
```

count_of_quantity	name
2416	The Classic Deluxe Pizza
2372	The Barbecue Chicken Pizza
2370	The Hawaiian Pizza
2369	The Pepperoni Pizza
2315	The Thai Chicken Pizza

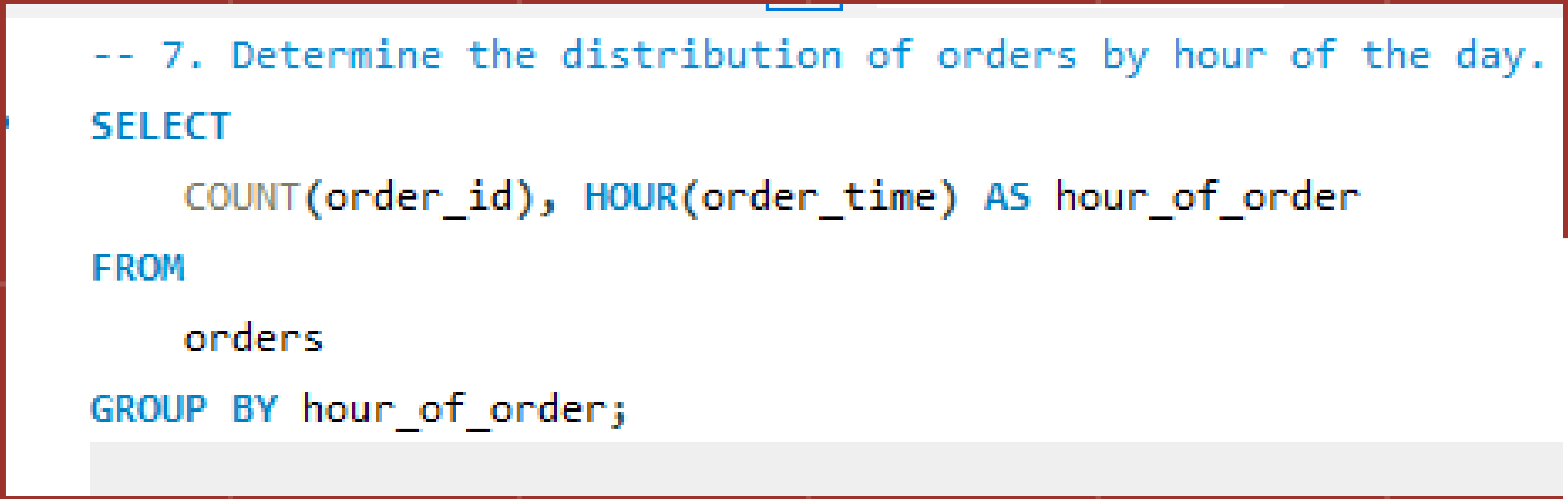
TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED



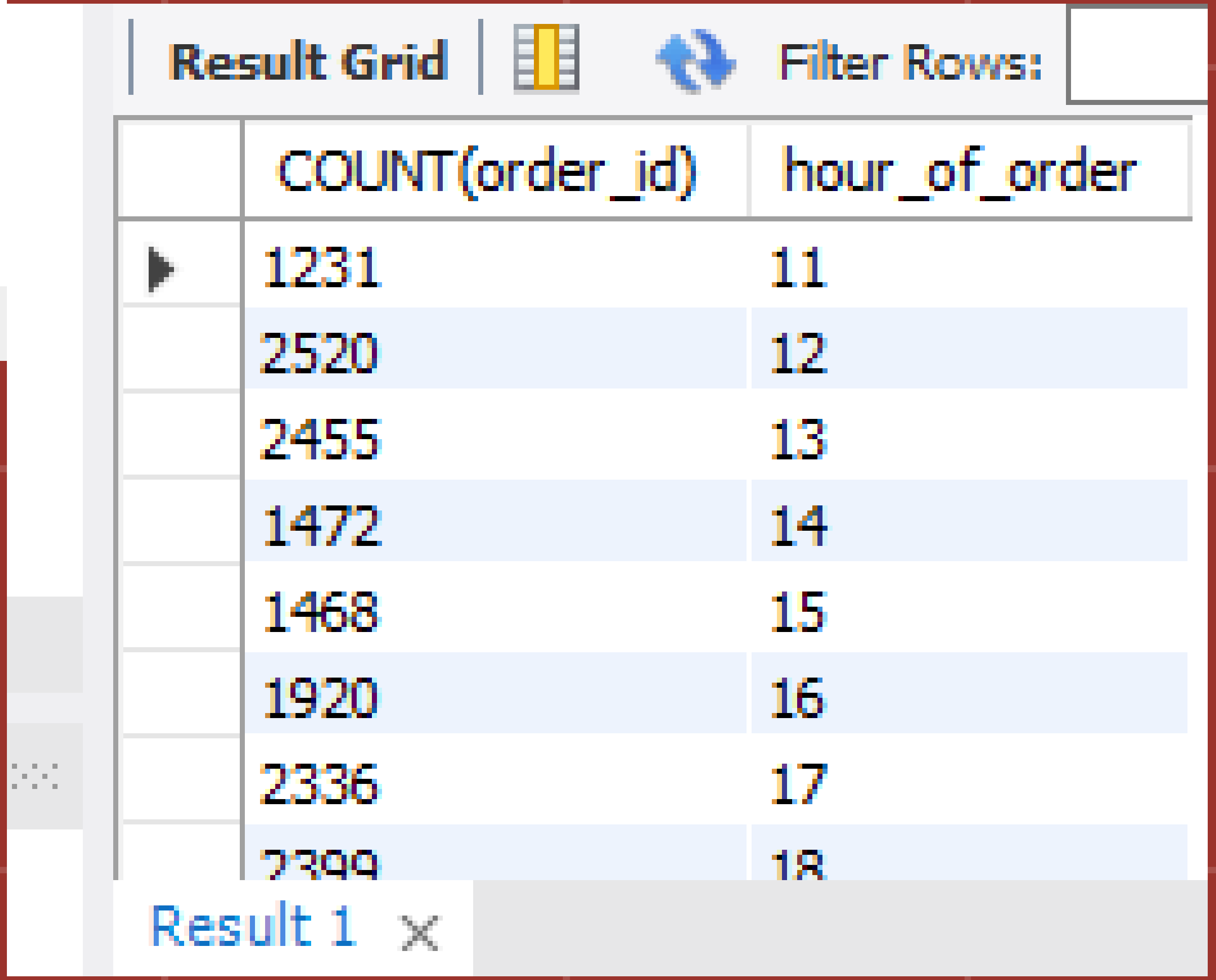
```
-- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    category, COUNT(quantity) AS count_of_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY category;
```

category	count_of_quantity
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

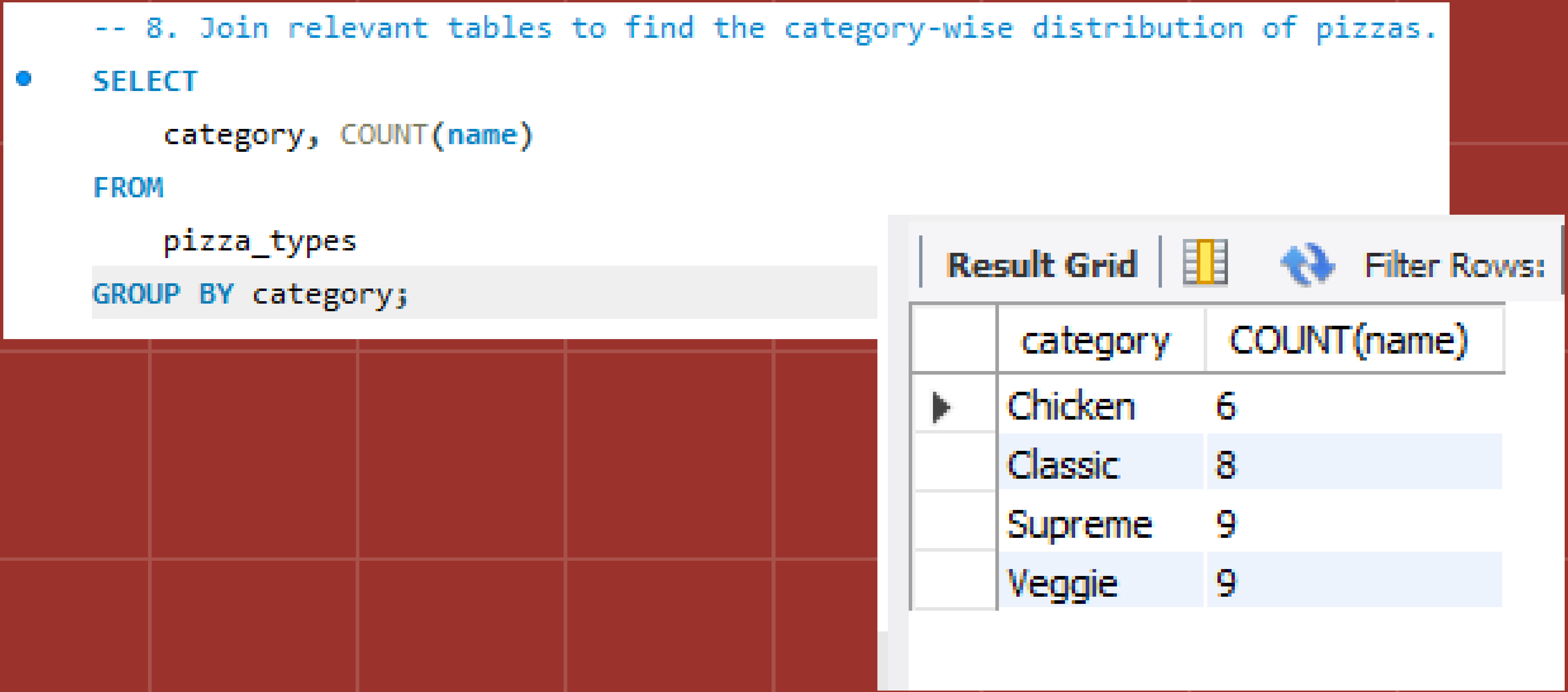


```sql
-- 7. Determine the distribution of orders by hour of the day.
SELECT
    COUNT(order_id), HOUR(order_time) AS hour_of_order
FROM
    orders
GROUP BY hour_of_order;
```



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

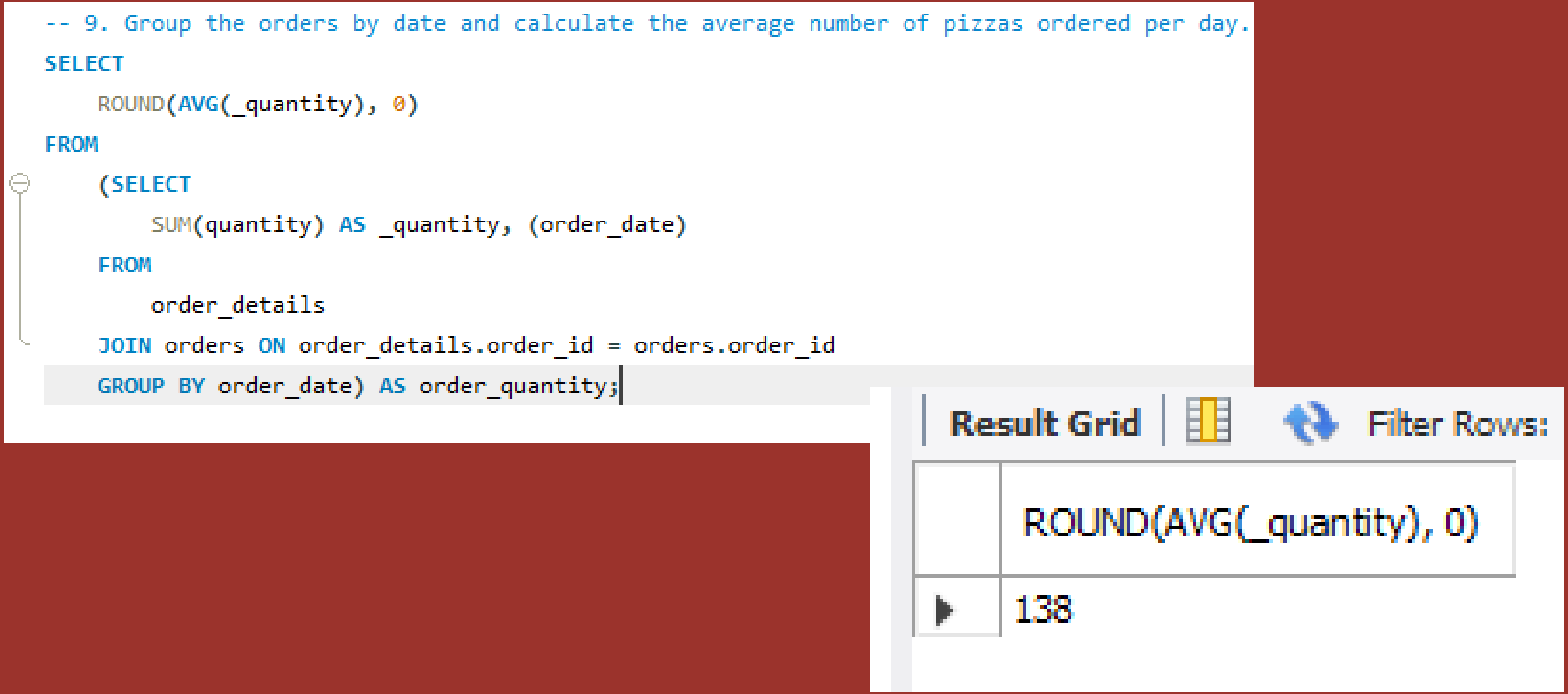
CATEGORY-WISE DISTRIBUTION OF PIZZAS



```
-- 8. Join 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

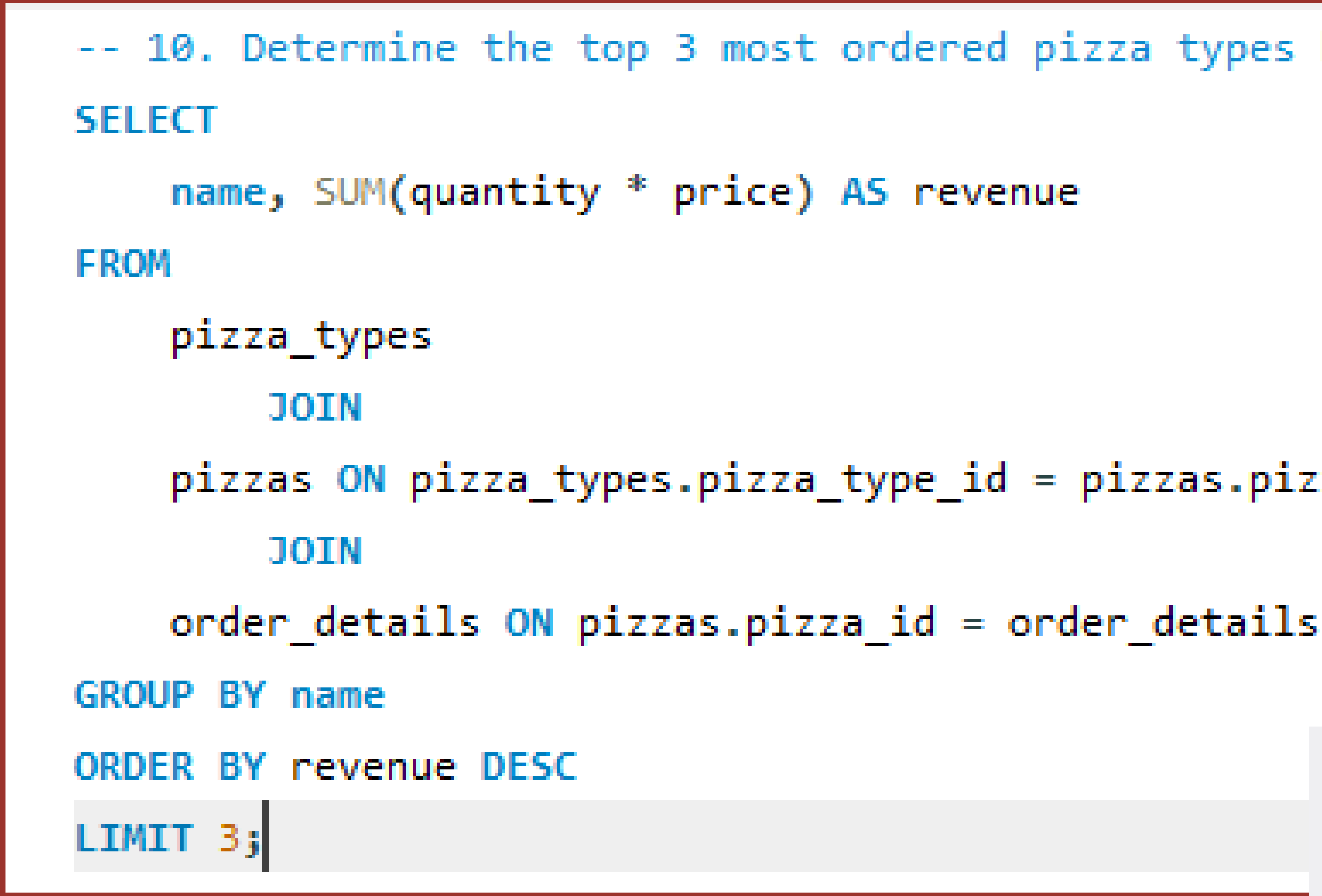
AVERAGE NUMBER OF PIZZAS ORDERED PER DAY



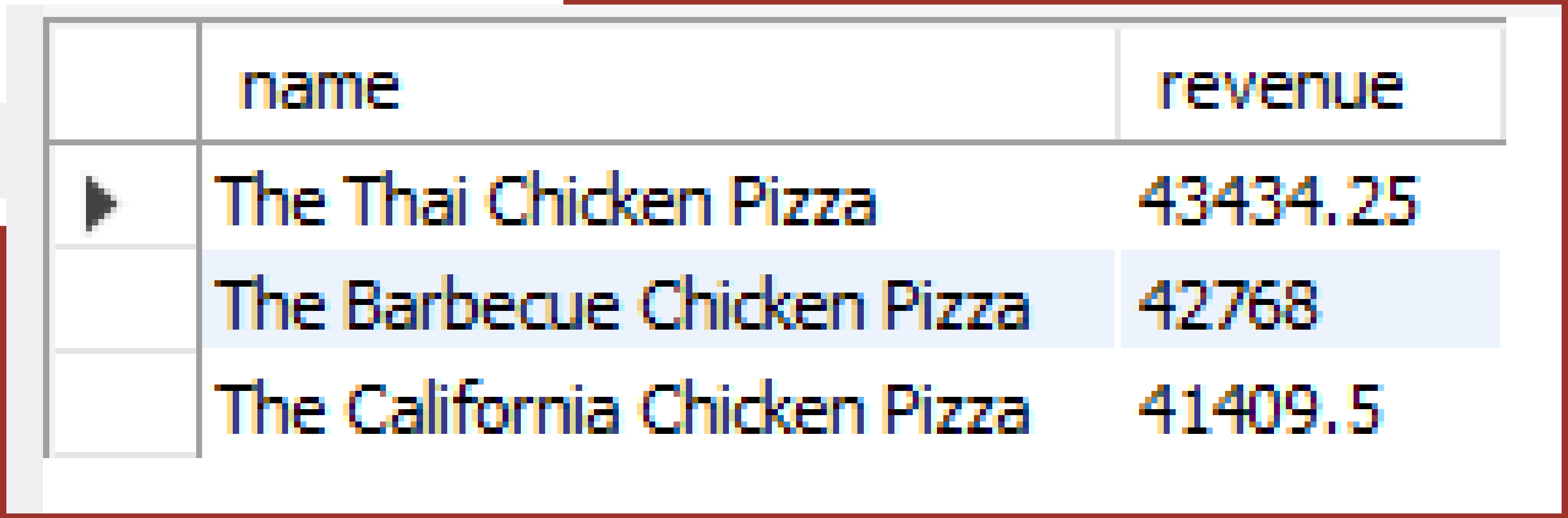
```sql
-- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(_quantity), 0)
FROM
    (SELECT
        SUM(quantity) AS _quantity, (order_date)
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY order_date) AS order_quantity;
```

```
ROUND(AVG(_quantity), 0)
138
```

3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

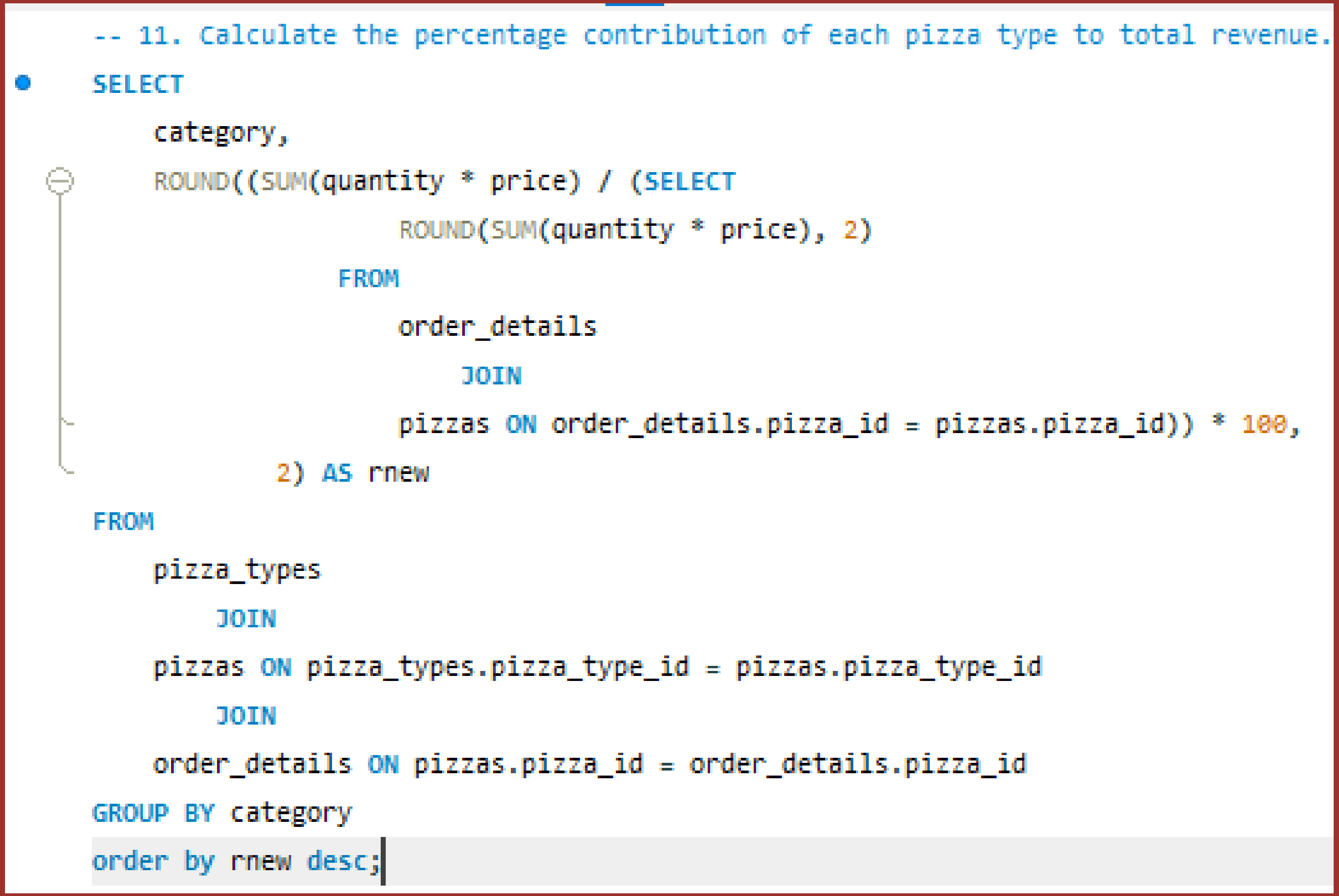


```sql
-- 10. Determine the top 3 most ordered pizza types based on revenue.
SELECT
    name, SUM(quantity * price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY name
ORDER BY revenue DESC
LIMIT 3;
```

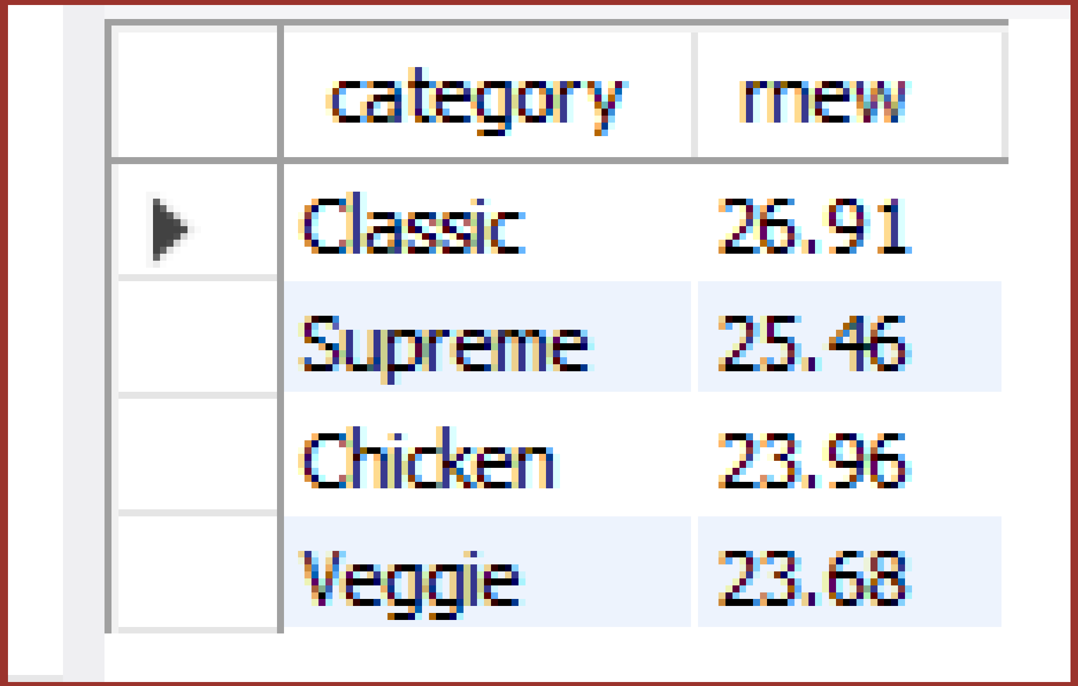


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

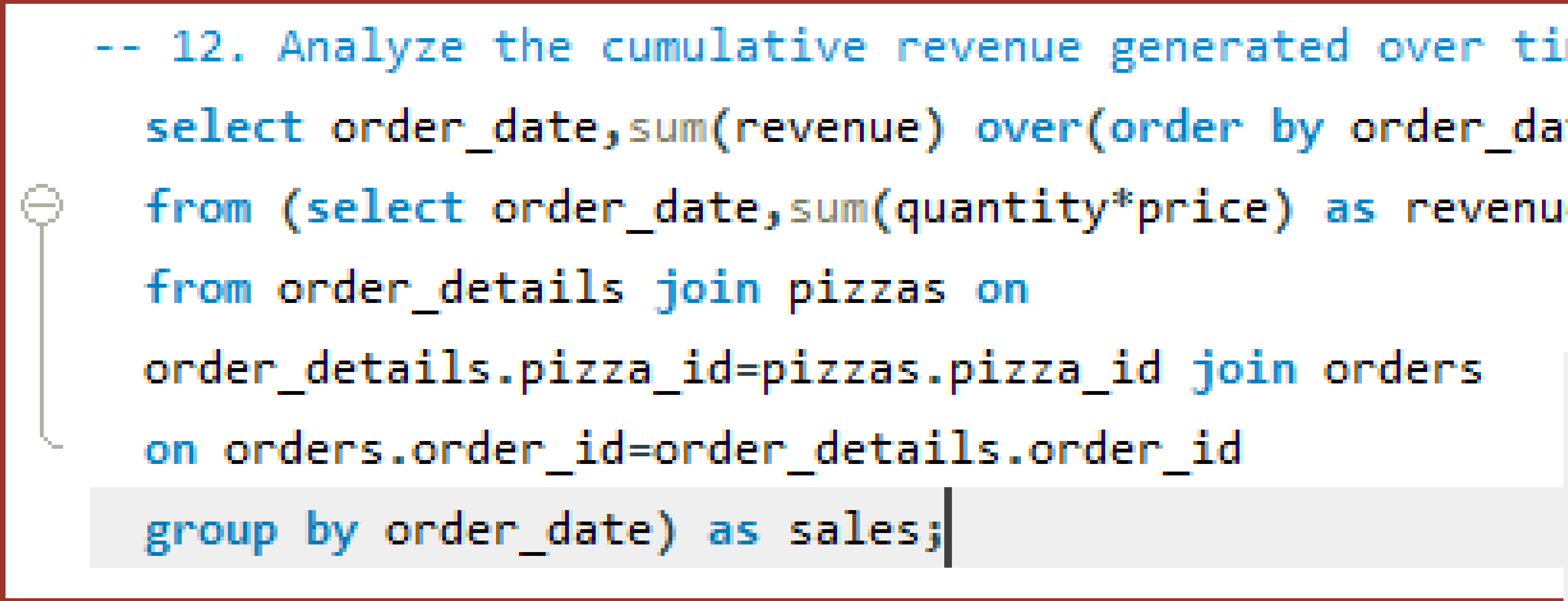


```
-- 11. Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    category,
    ROUND((SUM(quantity * price) / (SELECT
                    ROUND(SUM(quantity * price), 2)
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            2) AS rnew
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 category
order by rnew desc;
```

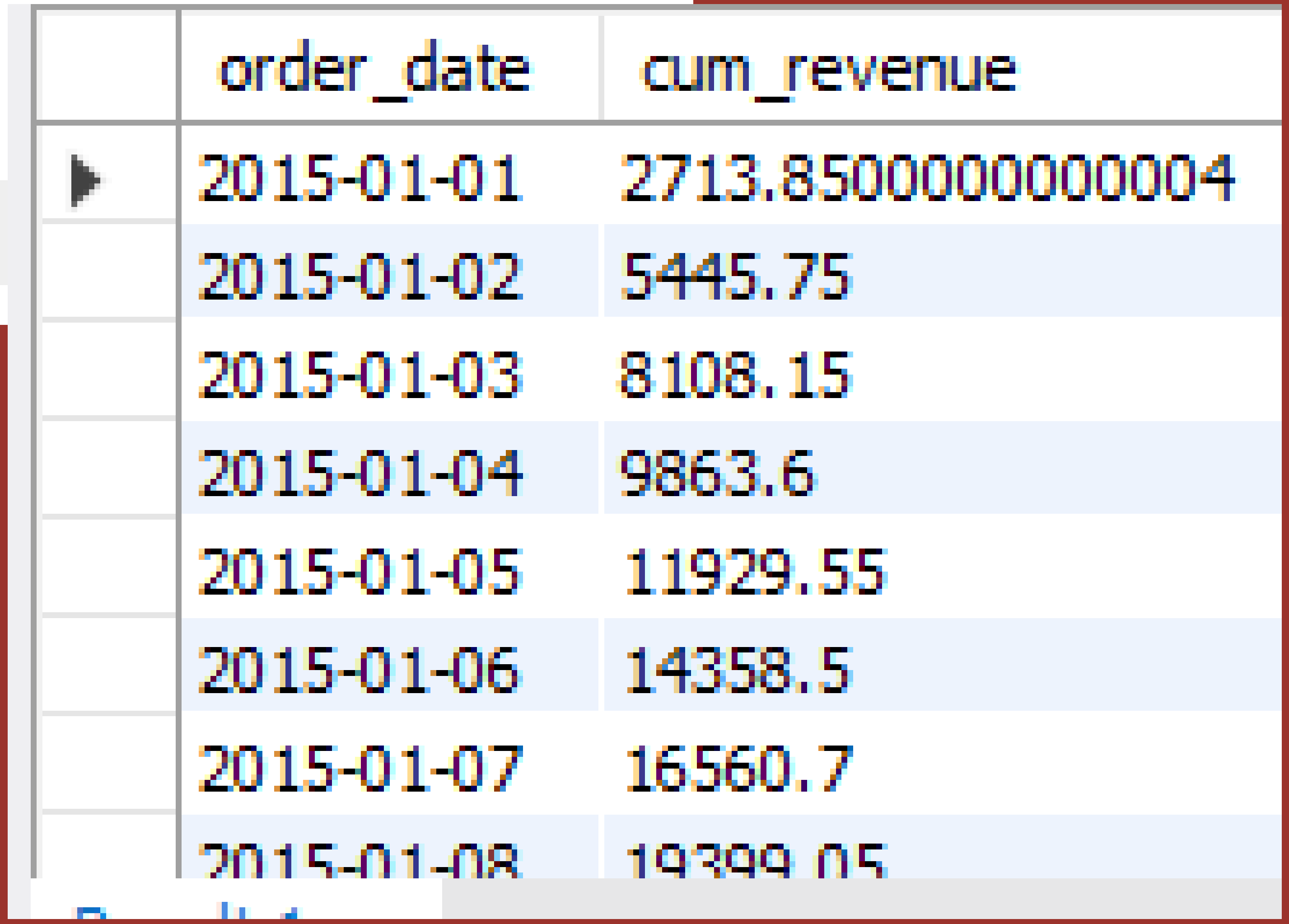


category	rnew
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

CUMULATIVE REVENUE GENERATED OVER TIME

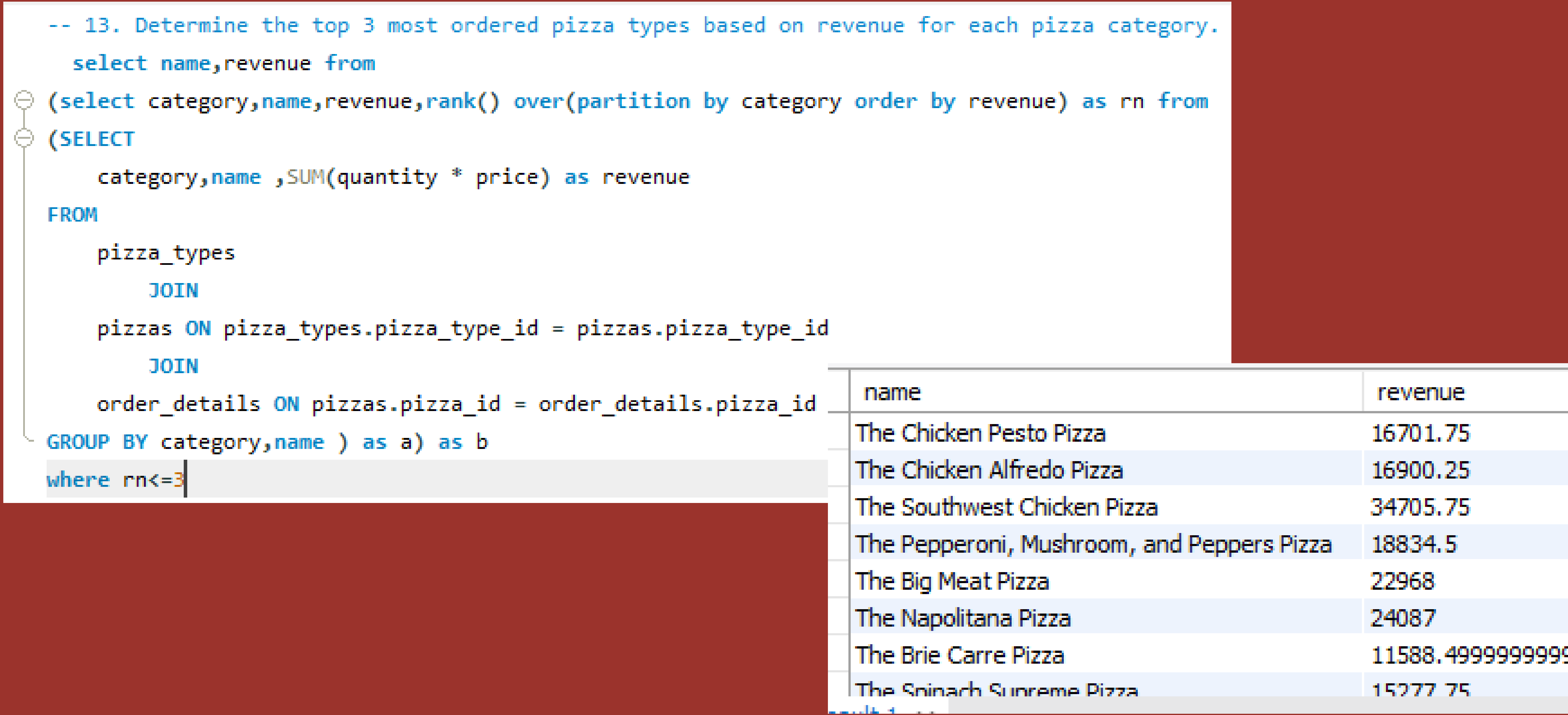


```
-- 12. Analyze the cumulative revenue generated over time.
  select order_date,sum(revenue) over(order by order_date) as cum_revenue
  from (select order_date,sum(quantity*price) as revenue
  from order_details join pizzas on
  order_details.pizza_id=pizzas.pizza_id join orders
  on orders.order_id=order_details.order_id
  group by order_date) as 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

3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY



```
-- 13. Determine the 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) as rn from
(SELECT
    category,name ,SUM(quantity * 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 category,name ) as a) as b
where rn<=3
```

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75

Here I am Developed SQL query to understand order patterns and customer preference on Pizza sales data.

THANK YOU.