SQL

Pizza sale project

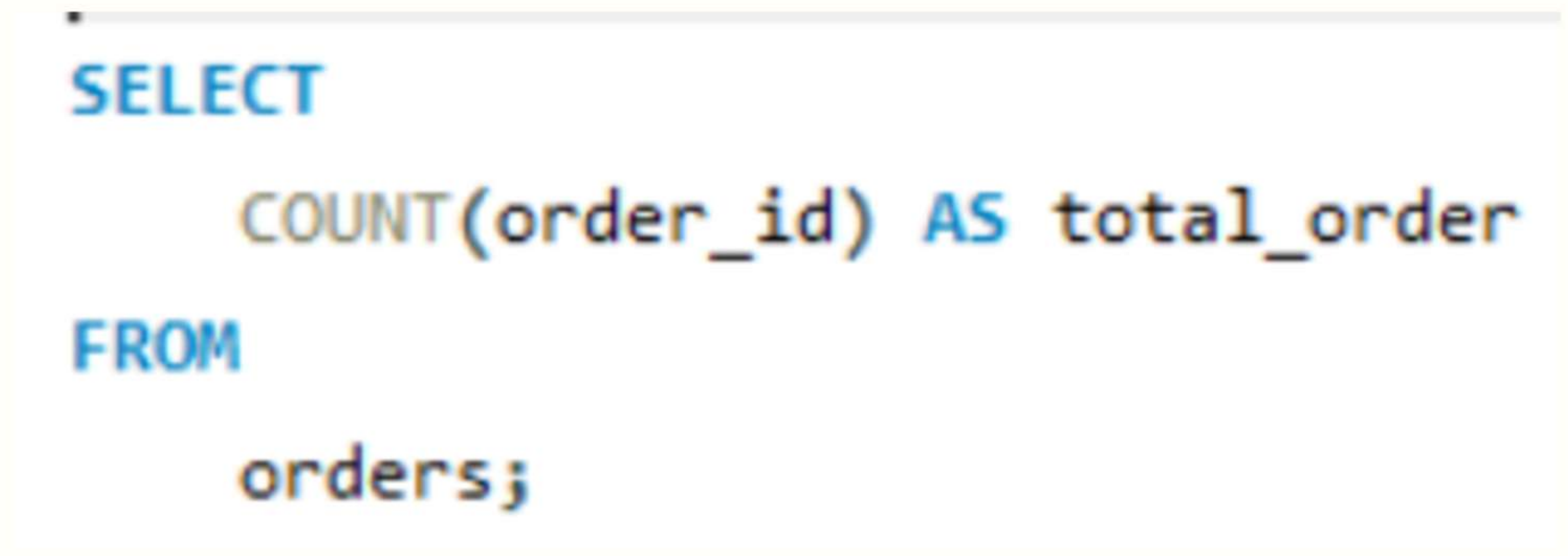
Welcome

Hello, my name is Shivaji Paudel. In this project, I have utilized SQL queries to analyze and solve business questions related to pizza sales data.

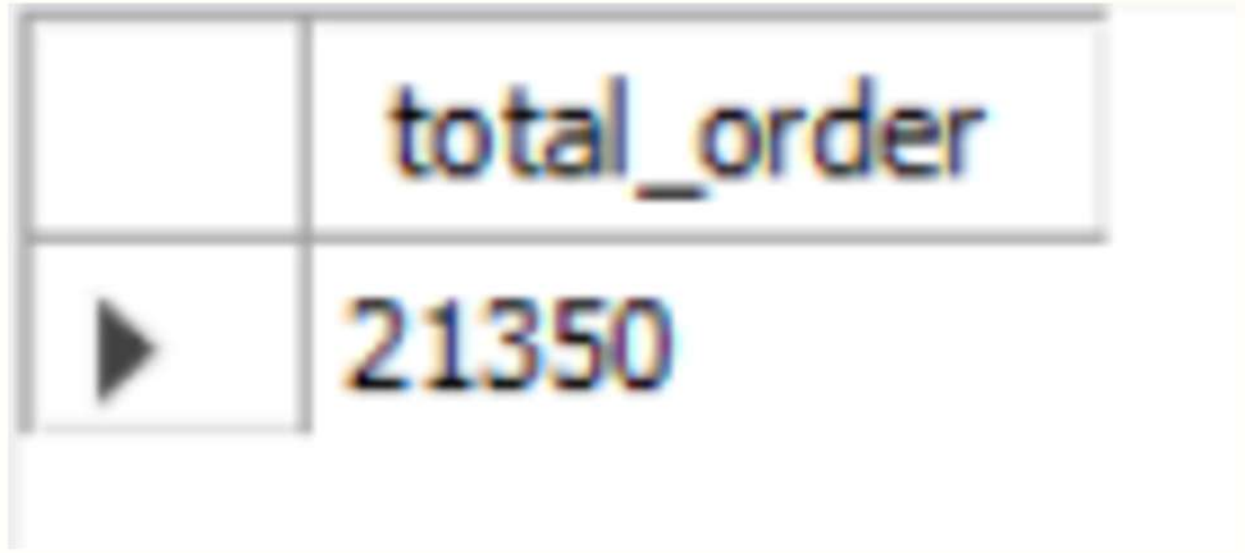
Key Objective

- 📦 Total Orders & Revenue Insights
- 💰 Identify Highest-Priced & Top-Selling Pizzas
- 📏 Analyze Most Popular Pizza Sizes & Categories
- ⏰ Understand Order Patterns by Time of Day
- 📊 Explore Sales Distribution by Pizza Type & Category
- 🔢 Track Daily Averages & Cumulative Revenue Trends
- 🥇 Rank Pizza Types by Revenue, Quantity, and Category
- 📈 Calculate Each Pizza Type's Revenue Contribution

Retrieve the total number of orders placed.

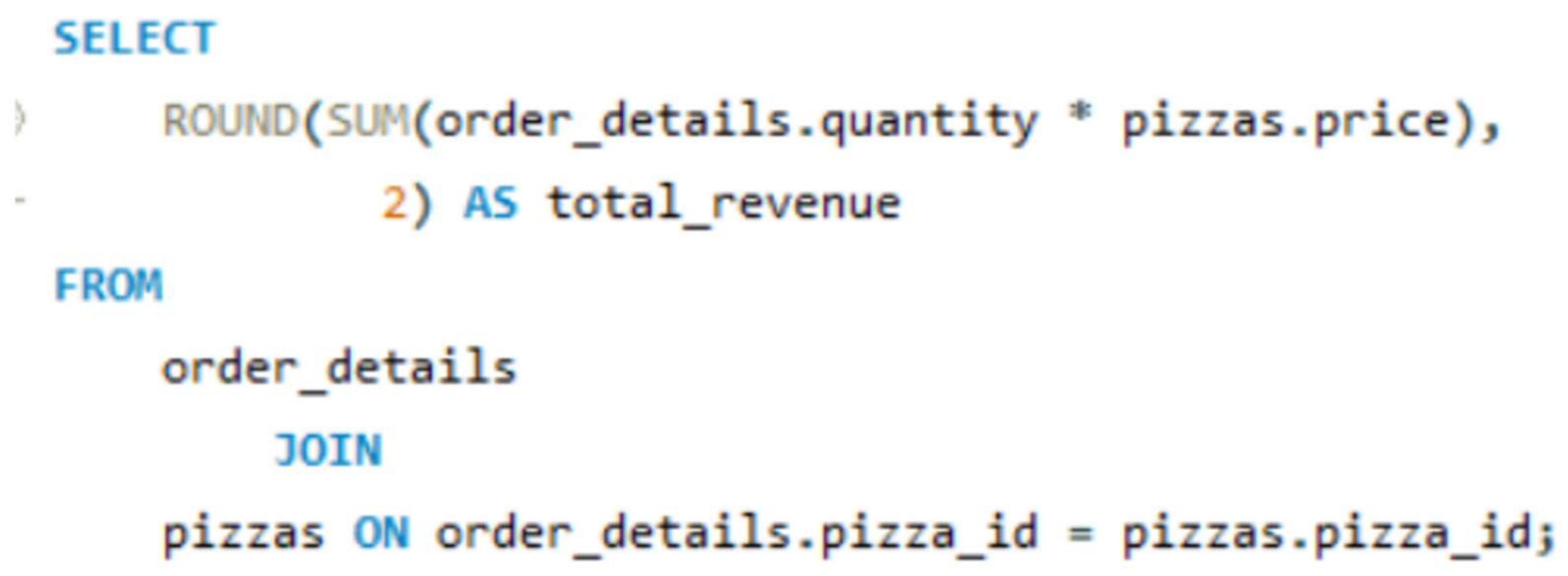


```
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```

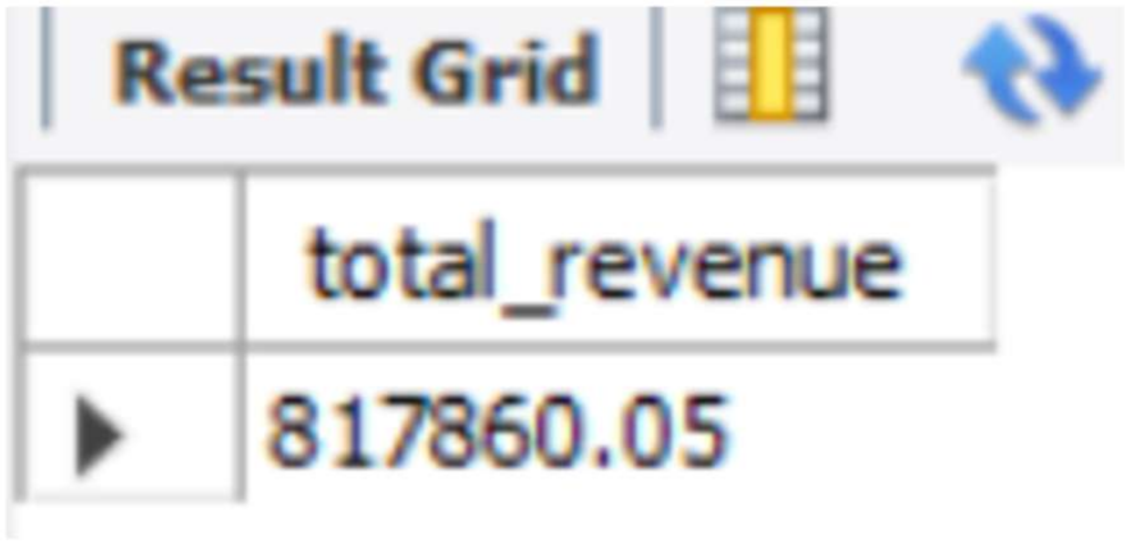


```
  | total_order
▶ | 21350
```

Calculate the total revenue generated from pizza sales.



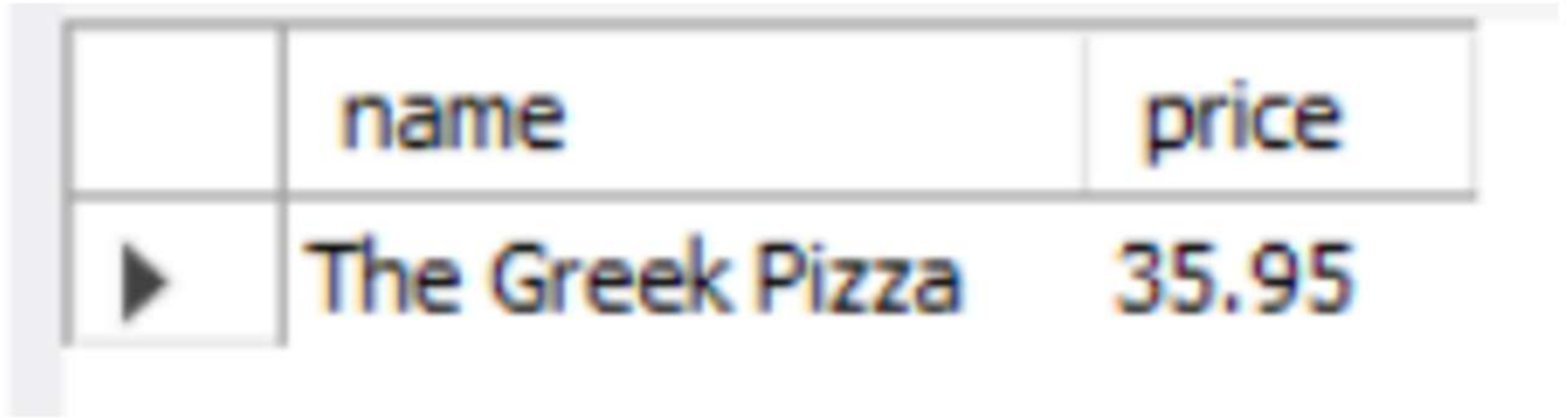
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



```
total_revenue
817860.05
```

Identify the highest-priced pizza.

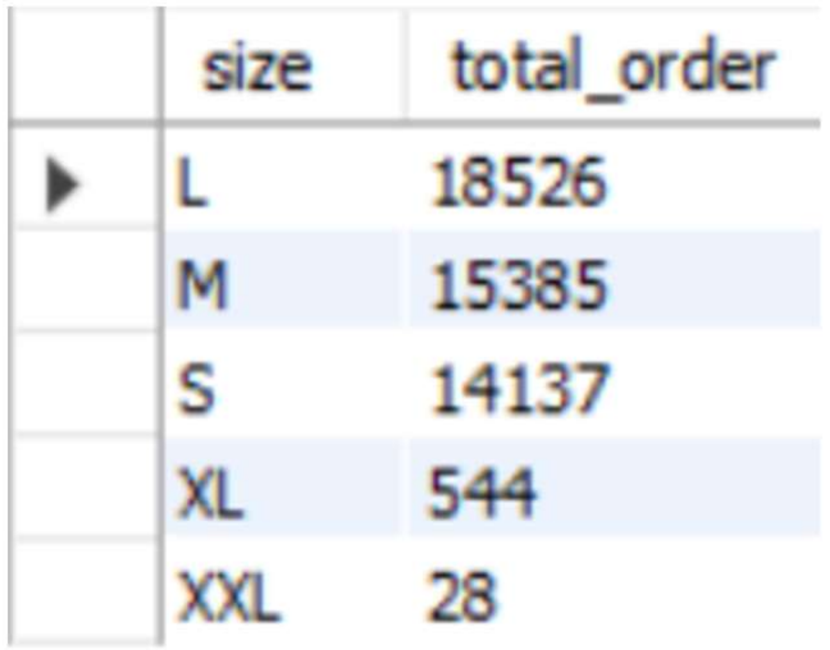
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

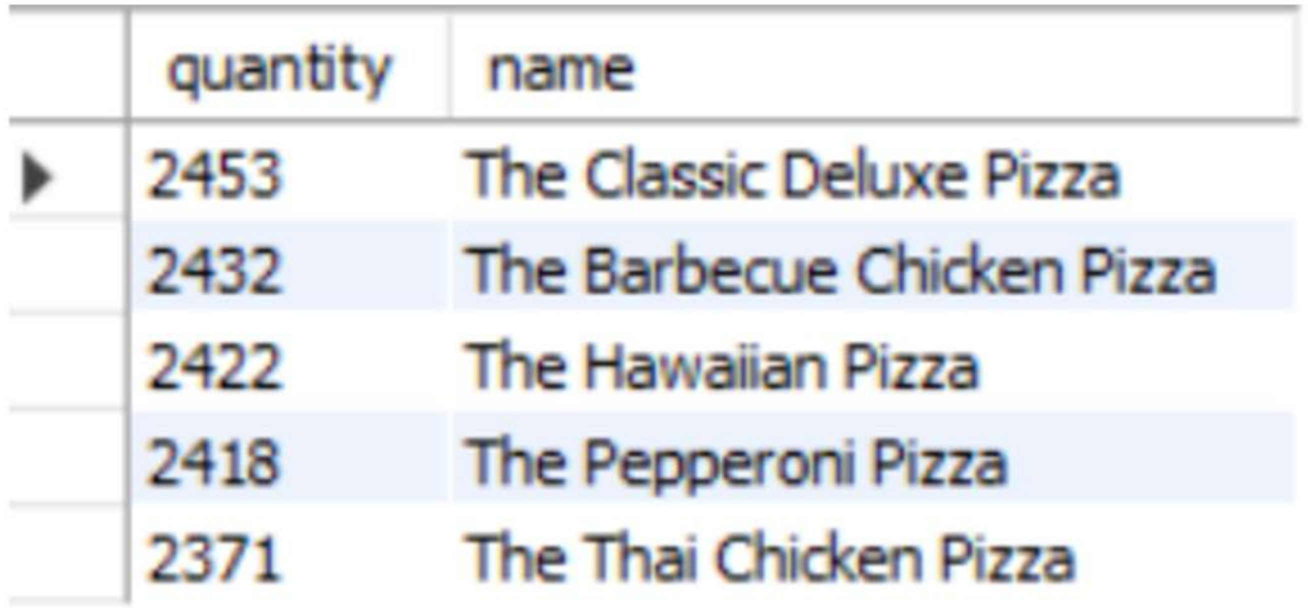
```
SELECT
    size, COUNT(order_details_id) AS total_order
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY total_order DESC;
```



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

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

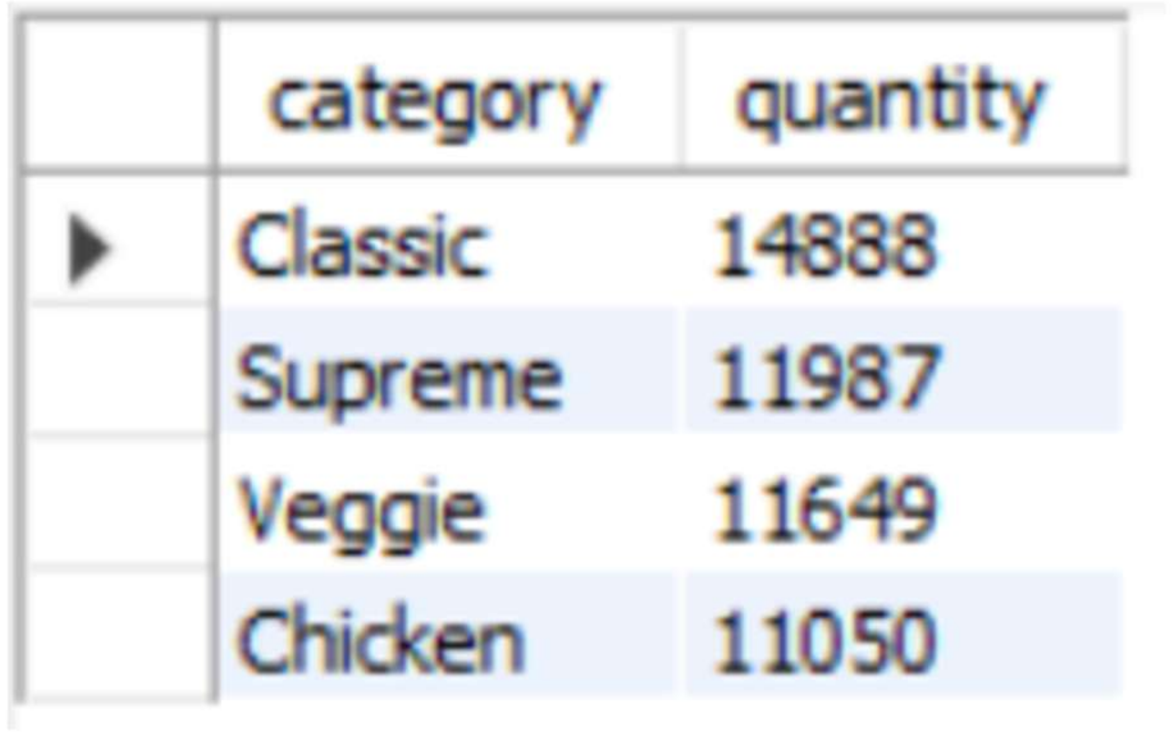
```sql
SELECT
    SUM(order_details.quantity) AS quantity, pizza_types.name
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY name
ORDER BY quantity DESC
LIMIT 5;
```



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

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

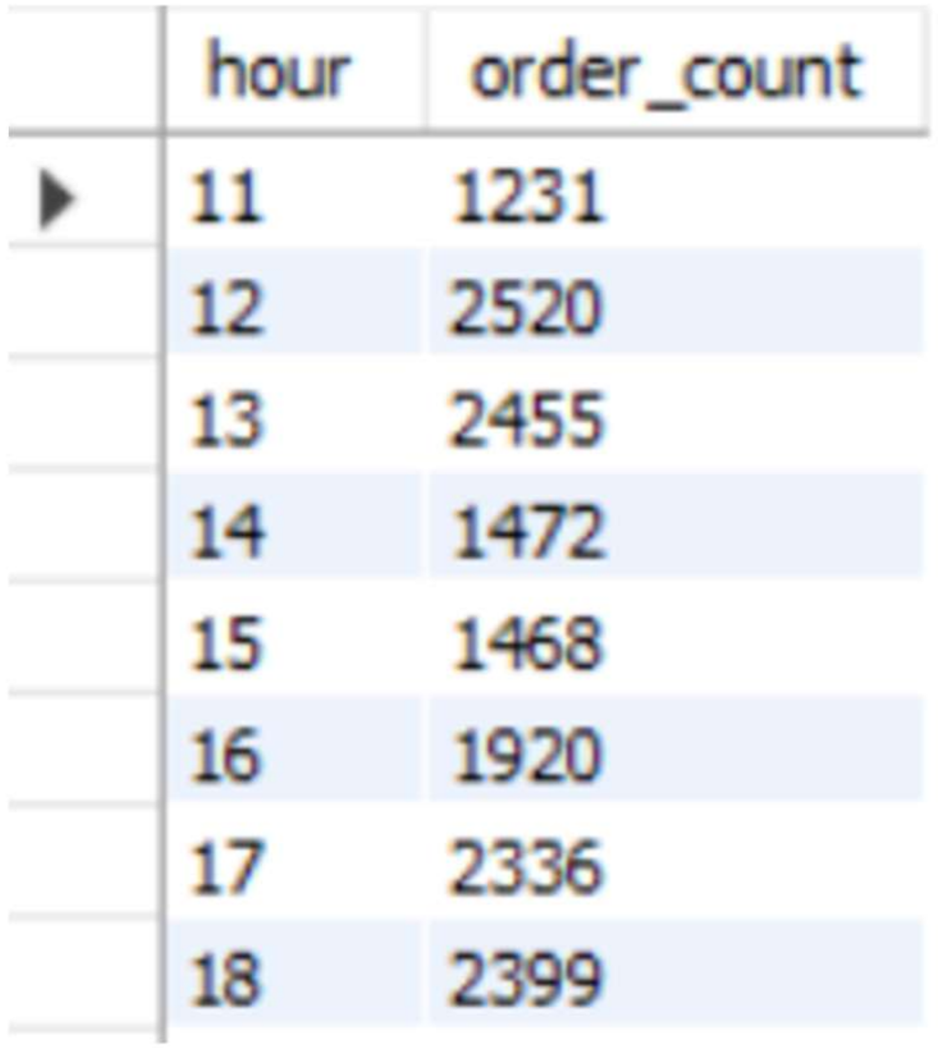
```
SELECT
    category, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY quantity DESC;
```



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

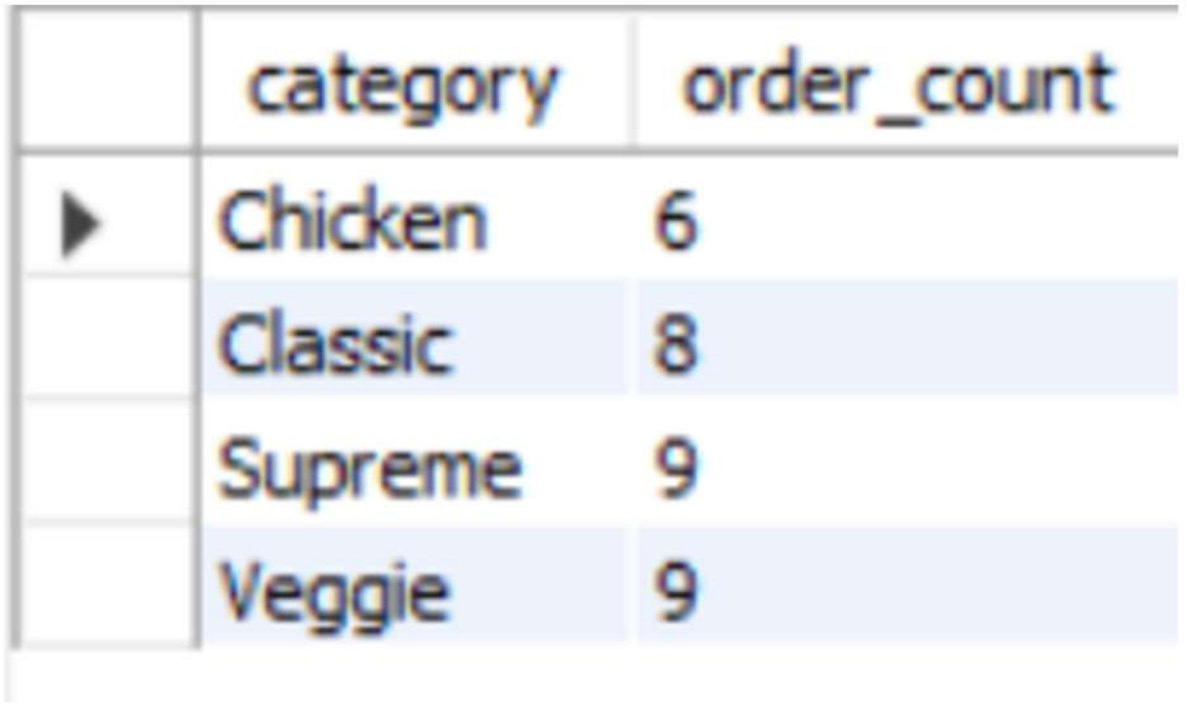
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour;
```



hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

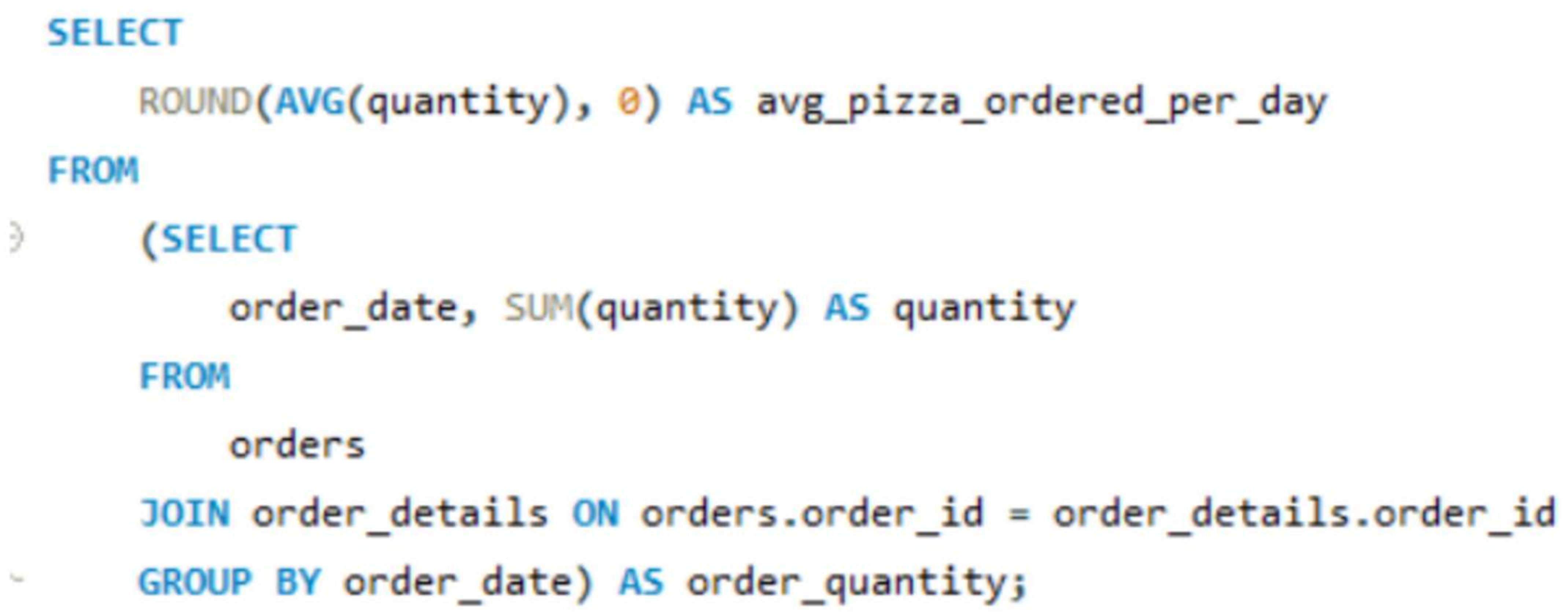
Find the category-wise distribution of pizzas.

```
select category, count(name) as order_count from pizza_types
group by category;
```

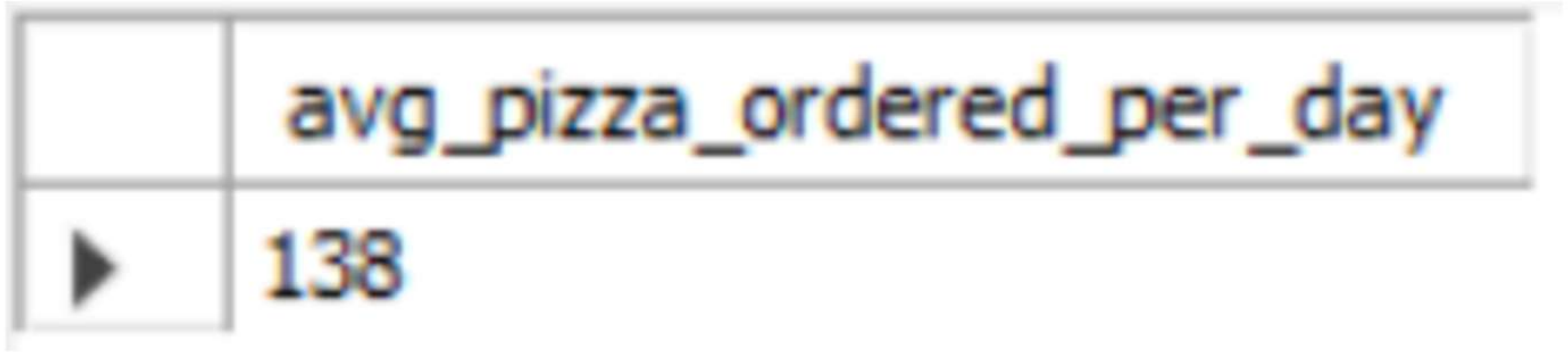


category	order_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

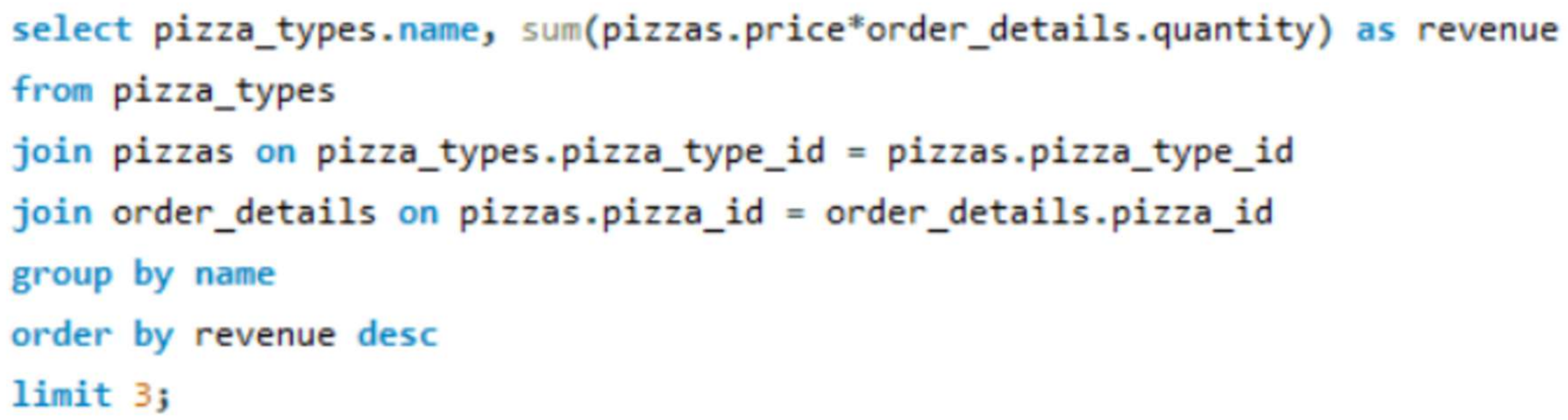


```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        order_date, SUM(quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY order_date) AS order_quantity;
```

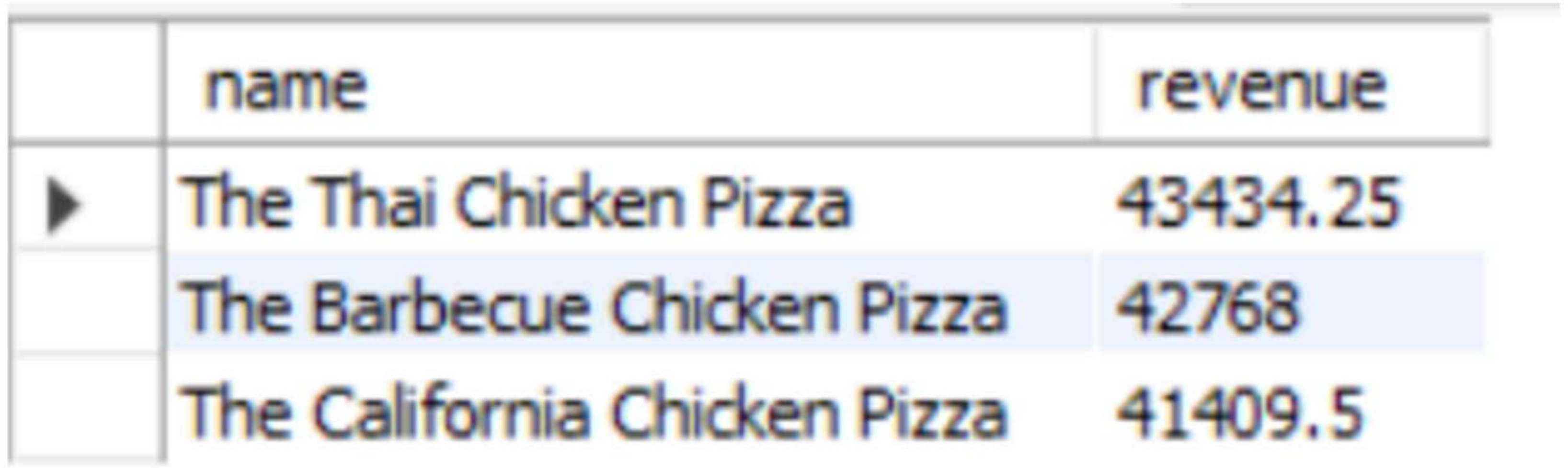


	avg_pizza_ordered_per_day
▶	138

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

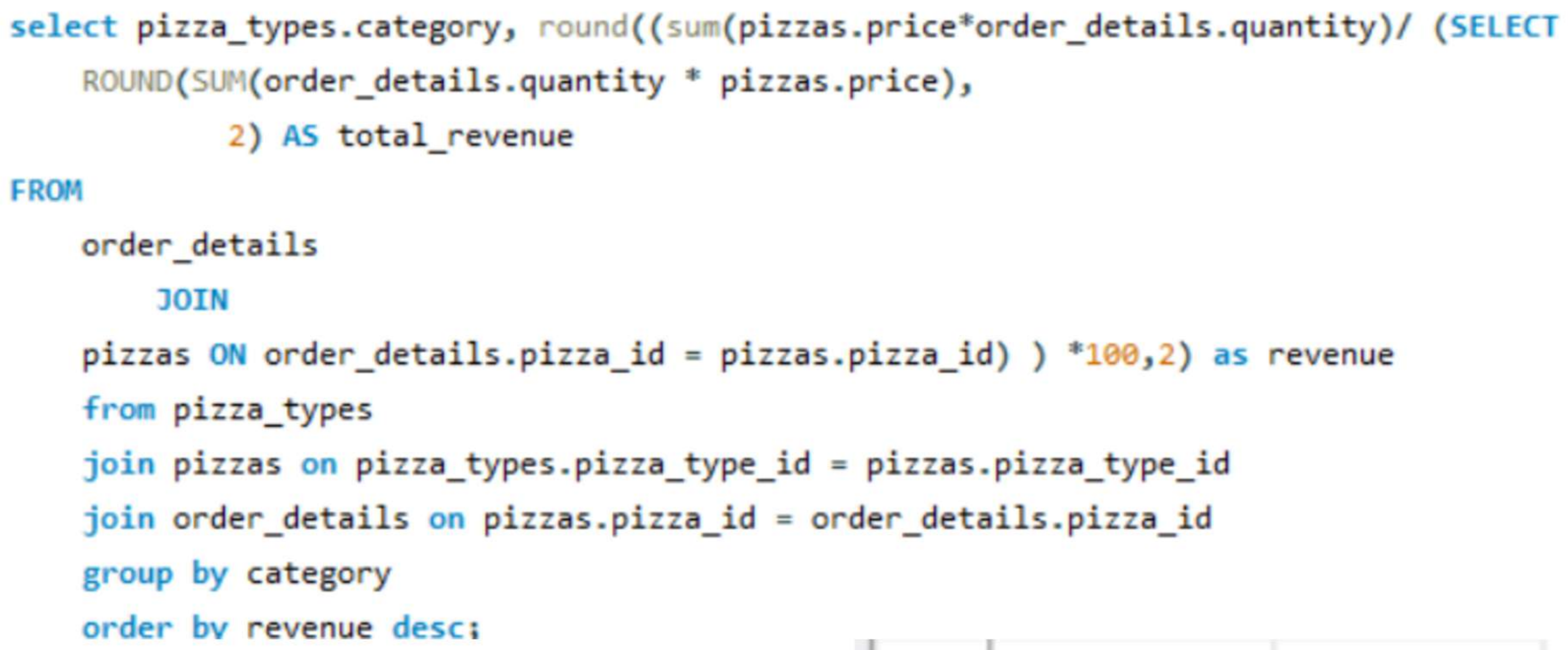


```
select pizza_types.name, sum(pizzas.price*order_details.quantity) as revenue
from pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on pizzas.pizza_id = order_details.pizza_id
group by name
order by revenue desc
limit 3;
```

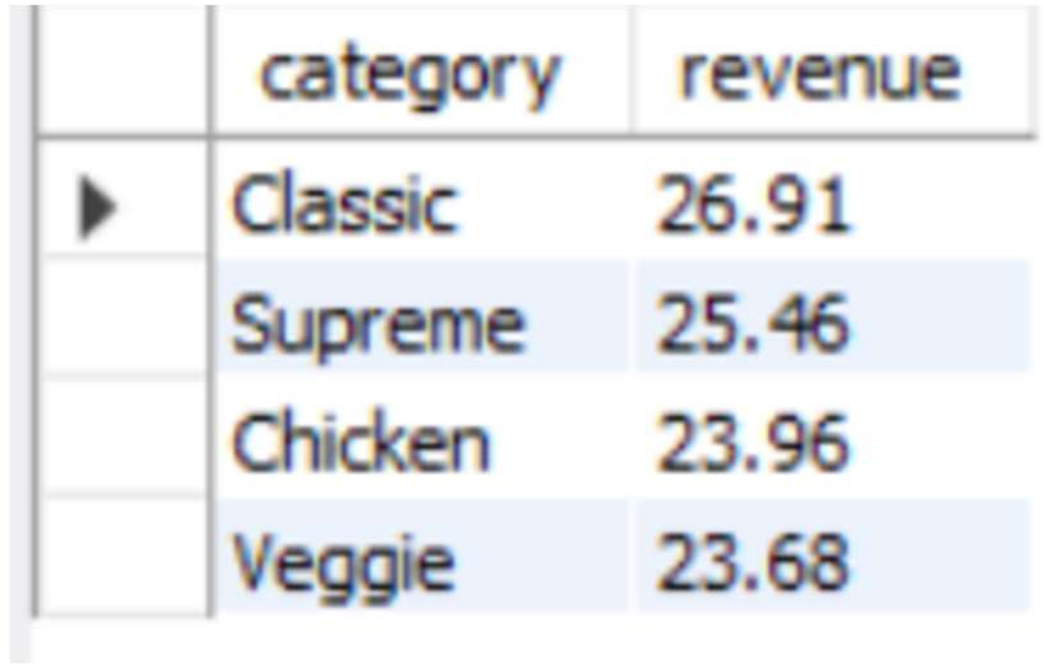


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.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 pizzas.pizza_id = order_details.pizza_id
    group by category
    order by revenue desc;
```

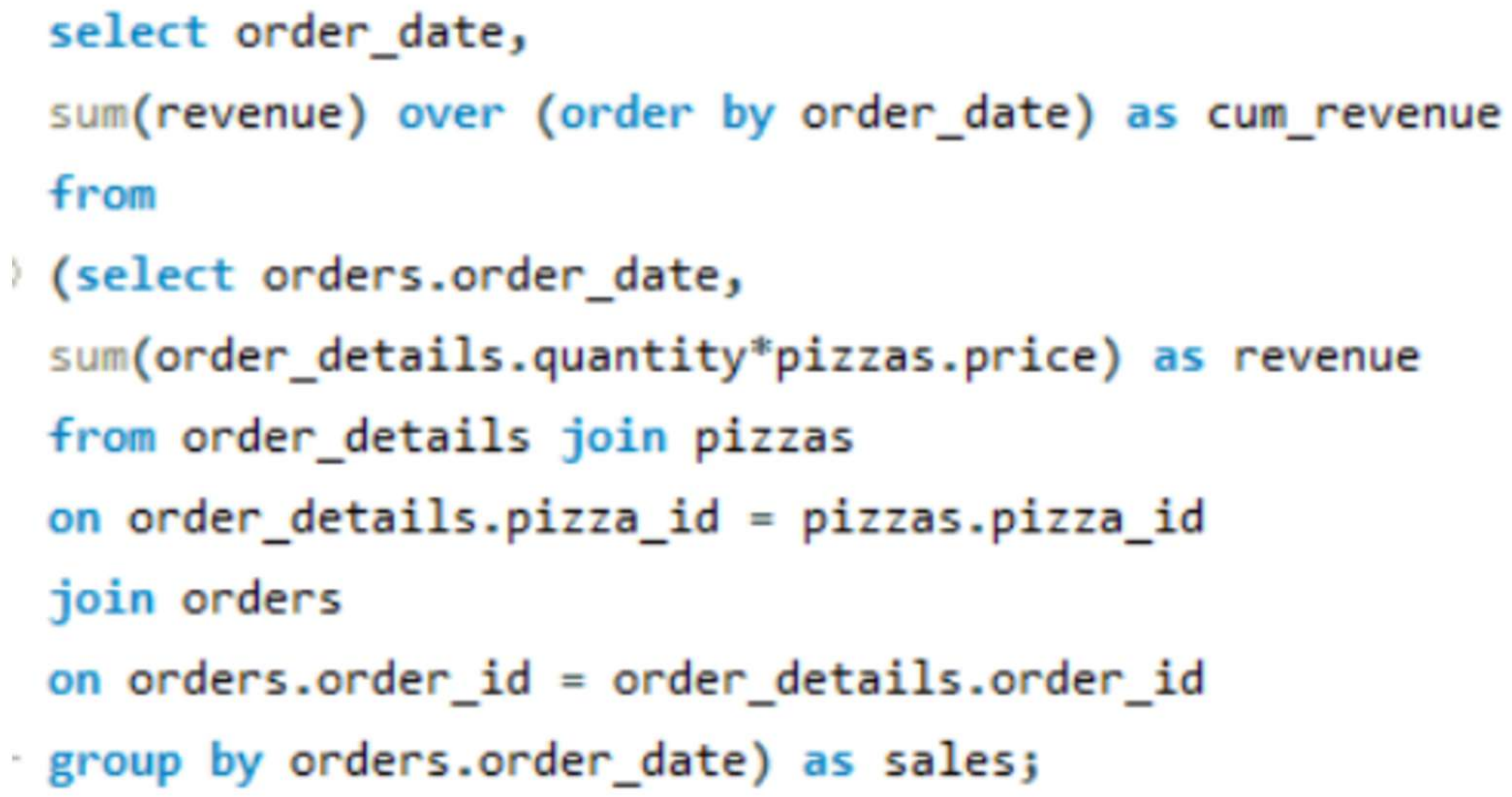


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

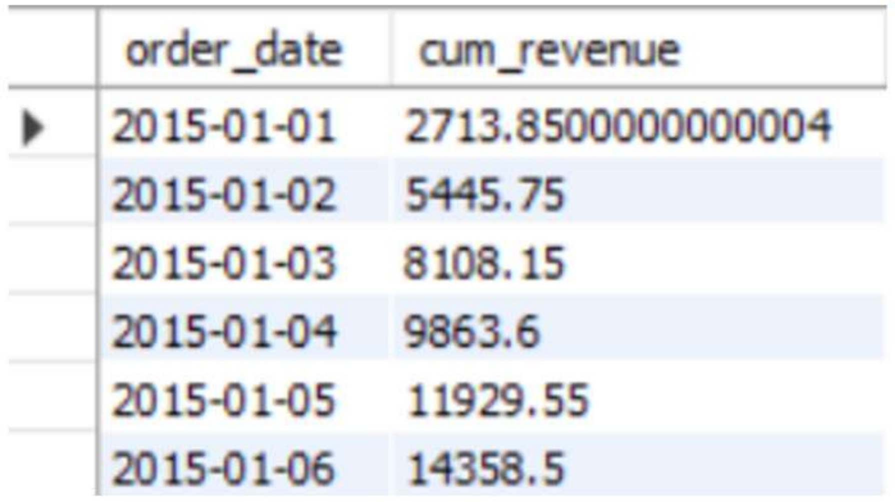
FORMULE

REVENUE % = (REVENUE OF ITEM ÷ TOTAL REVENUE) × 100

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;
```



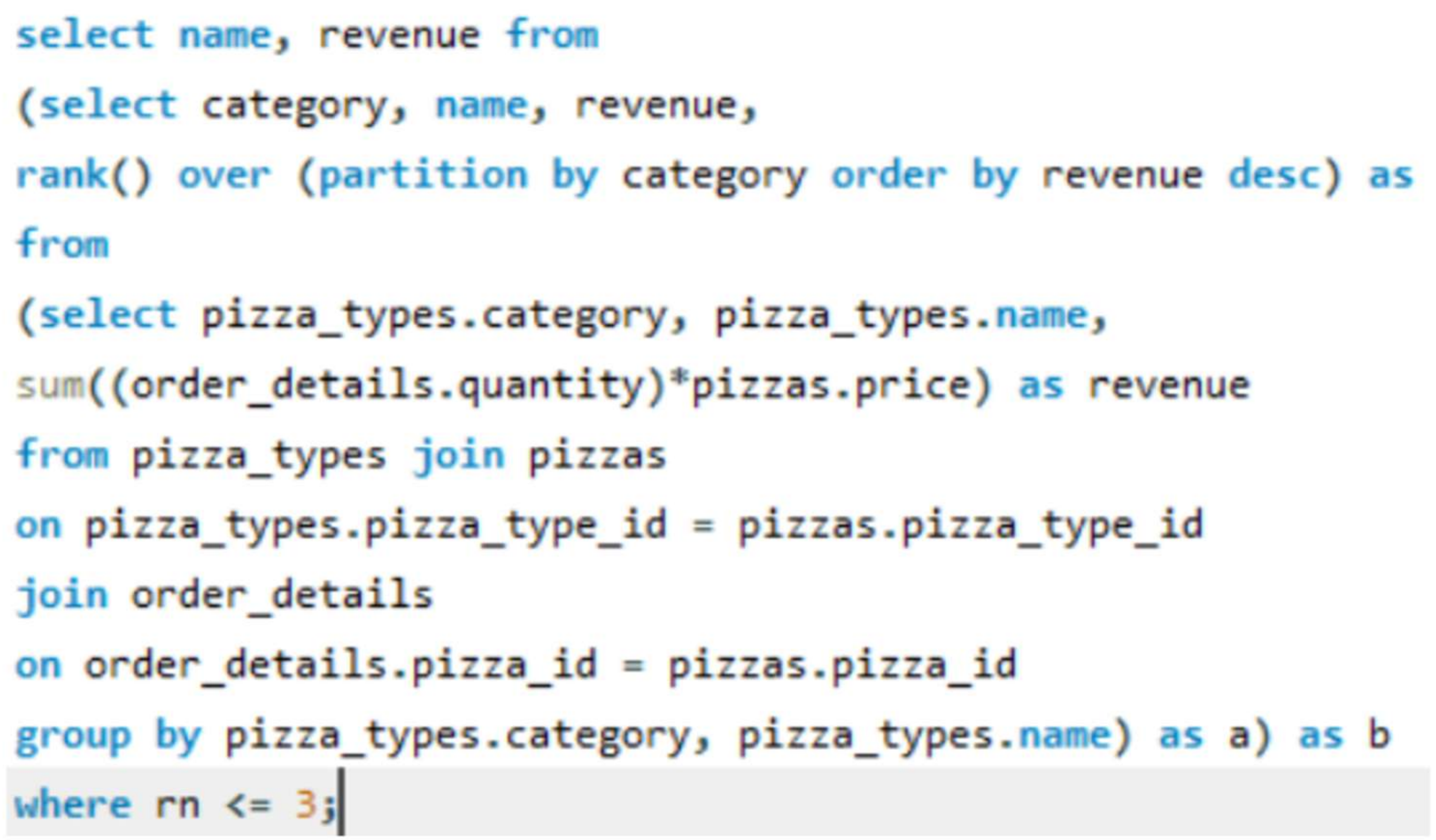
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5

Cumulative revenue means **adding each day's earnings to the previous total.**

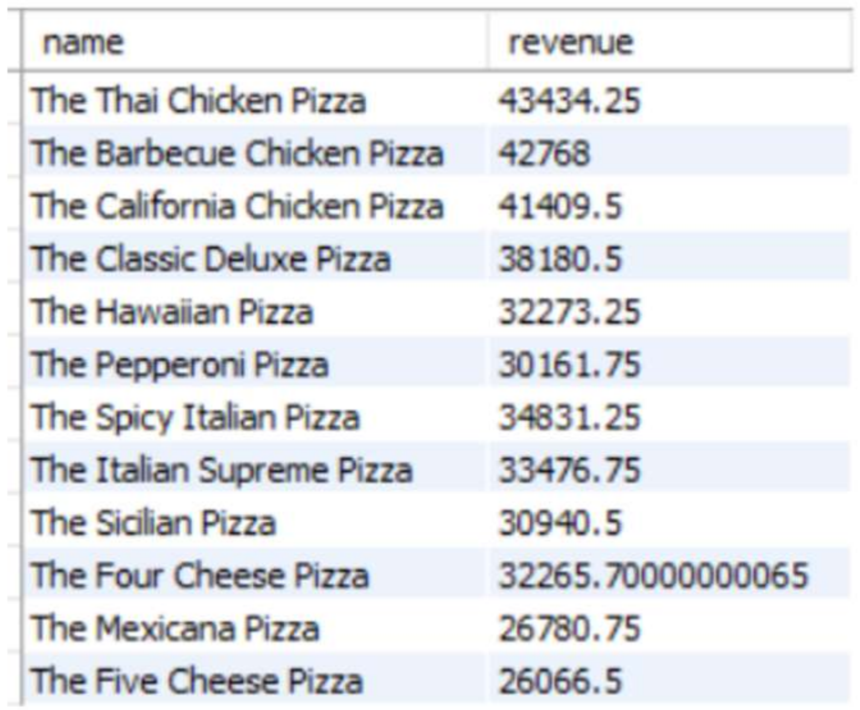
Example:

- 📅 Day 1: Earned ₹200 → **Total = 200**
- 📅 Day 2: Earned ₹100 → 200 + 100 = **300**
- 📅 Day 3: Earned ₹400 → 300 + 400 = **700**

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 desc) as
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 Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Thank You