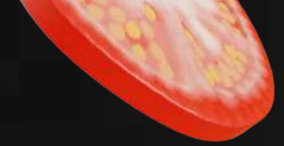
SQL Project on Pizza Sales

HELLO!

I'm Keerthi Kelam.

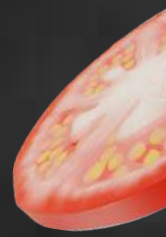
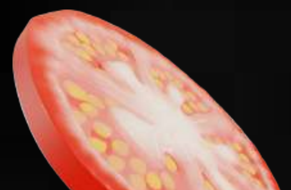
In this project, I have used **SQL** to analyze and solve various queries based on a Pizza Sales dataset. The goal was to extract meaningful insights from the data using SQL operations such as filtering, aggregation, and grouping.

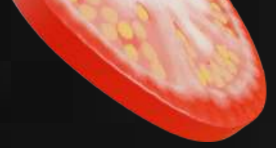


SQL QUERIES SOLVED

Basic Level:

- Retrieved the total number of orders placed
- Calculated the total revenue generated from pizza sales
- Identified the highest-priced pizza
- Found the most common pizza size ordered
- Listed the top 5 most ordered pizza types with their quantities

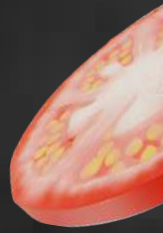
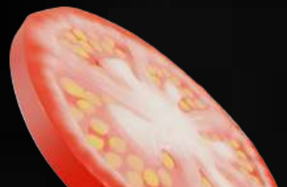



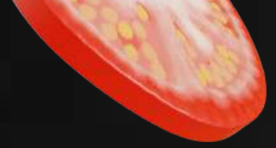


SQL QUERIES SOLVED

Intermediate Level:

- Joined necessary tables to find total quantity of each pizza category
- Analyzed distribution of orders by hour of the day
- Found category-wise distribution of pizzas
- Grouped orders by date to calculate average pizzas ordered per day
- Identified top 3 most ordered pizza types based on revenue

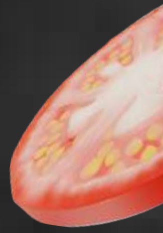
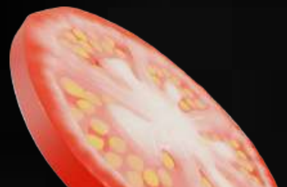





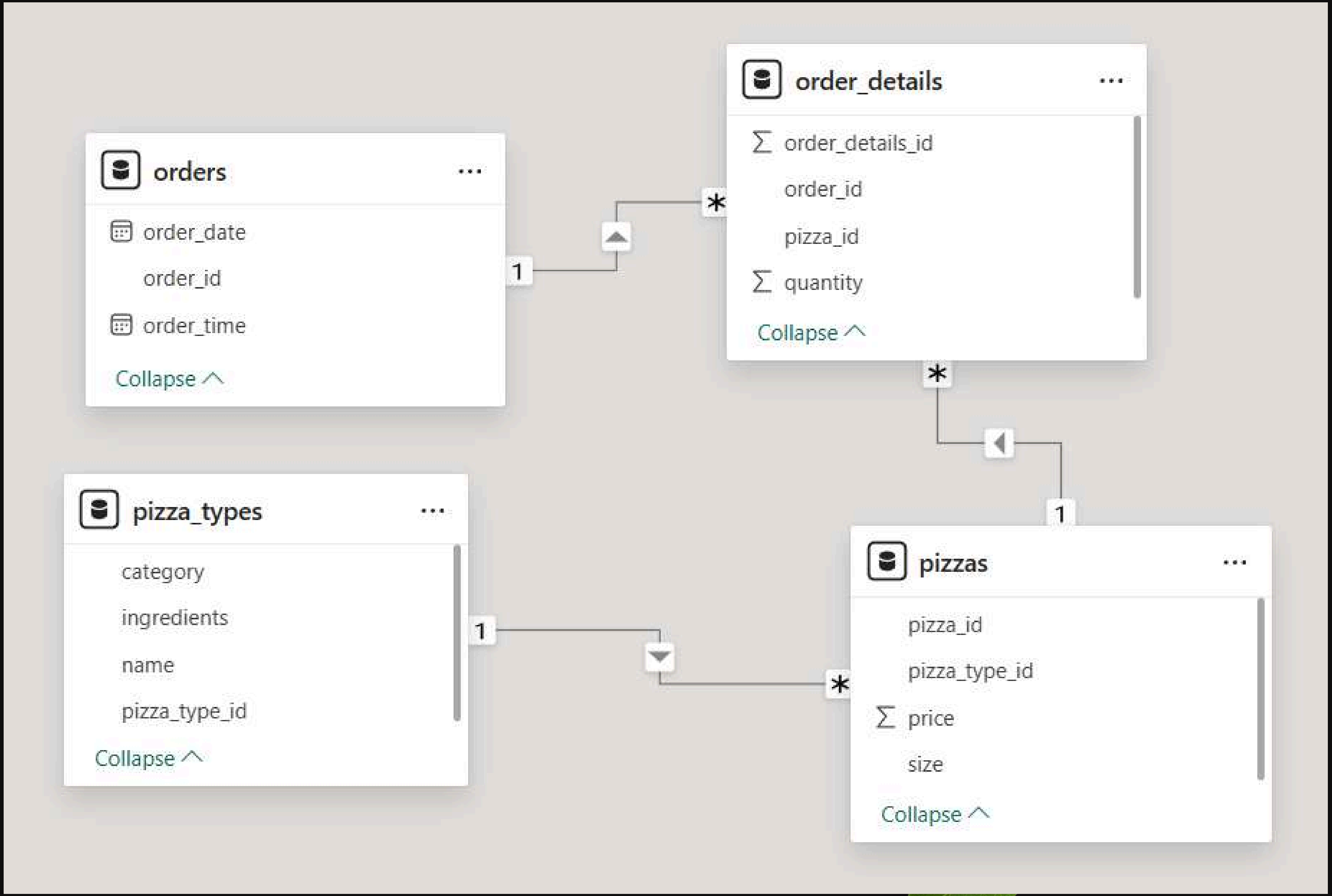
SQL QUERIES SOLVED

Advanced Level:

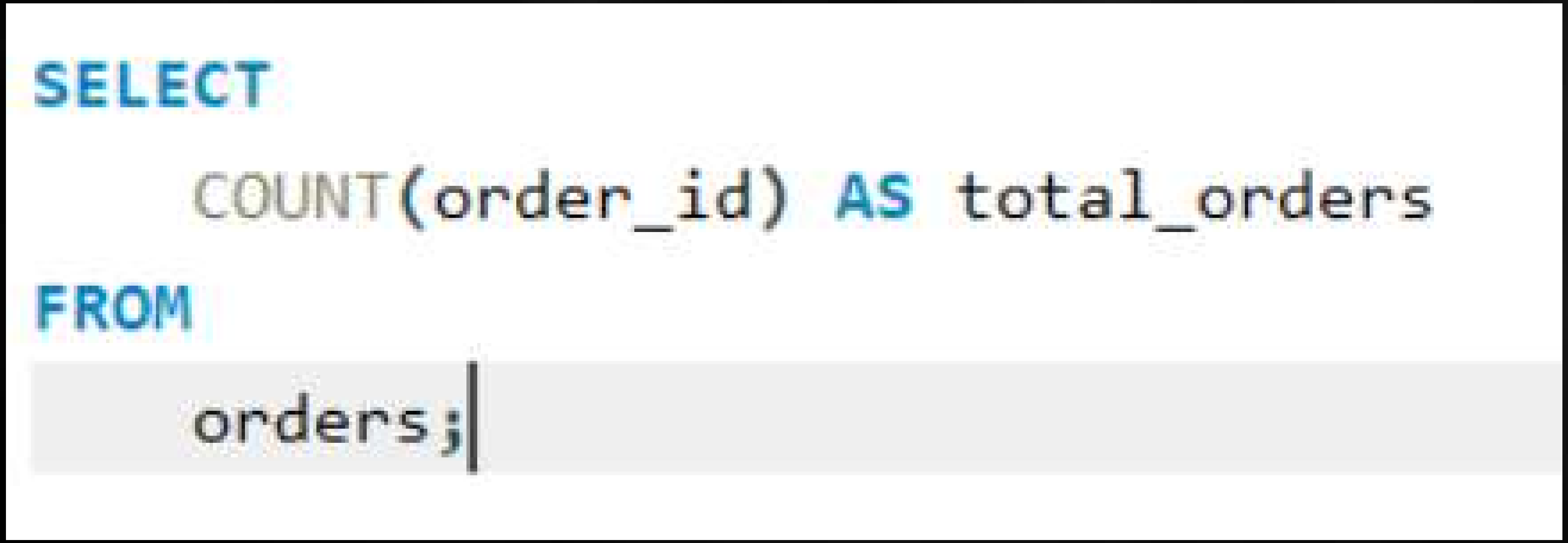
- Calculated each pizza type's percentage contribution to total revenue
- Analyzed cumulative revenue over time
- Found top 3 most ordered pizza types by revenue for each pizza category

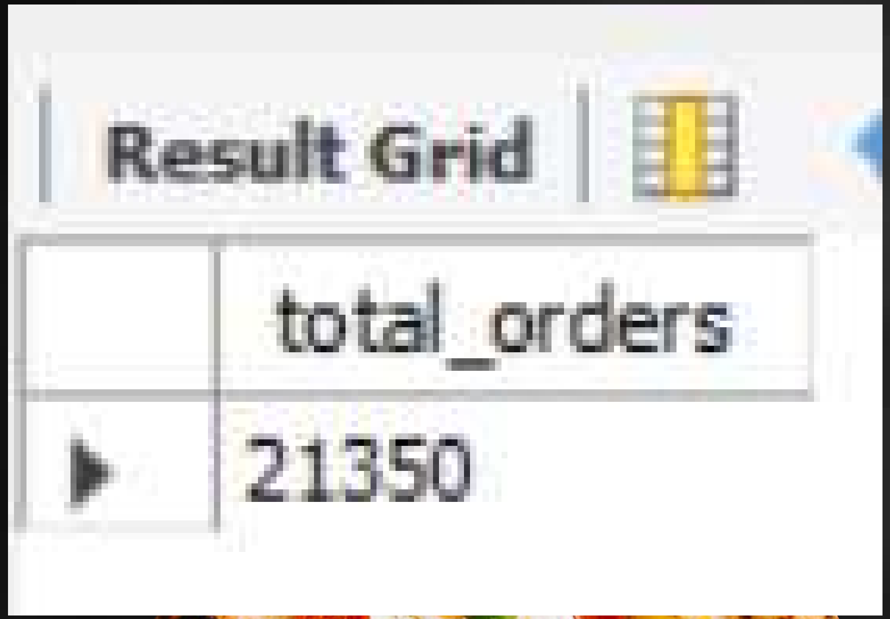
DATABASE SCHEMA



Retrieve the total number of orders placed.

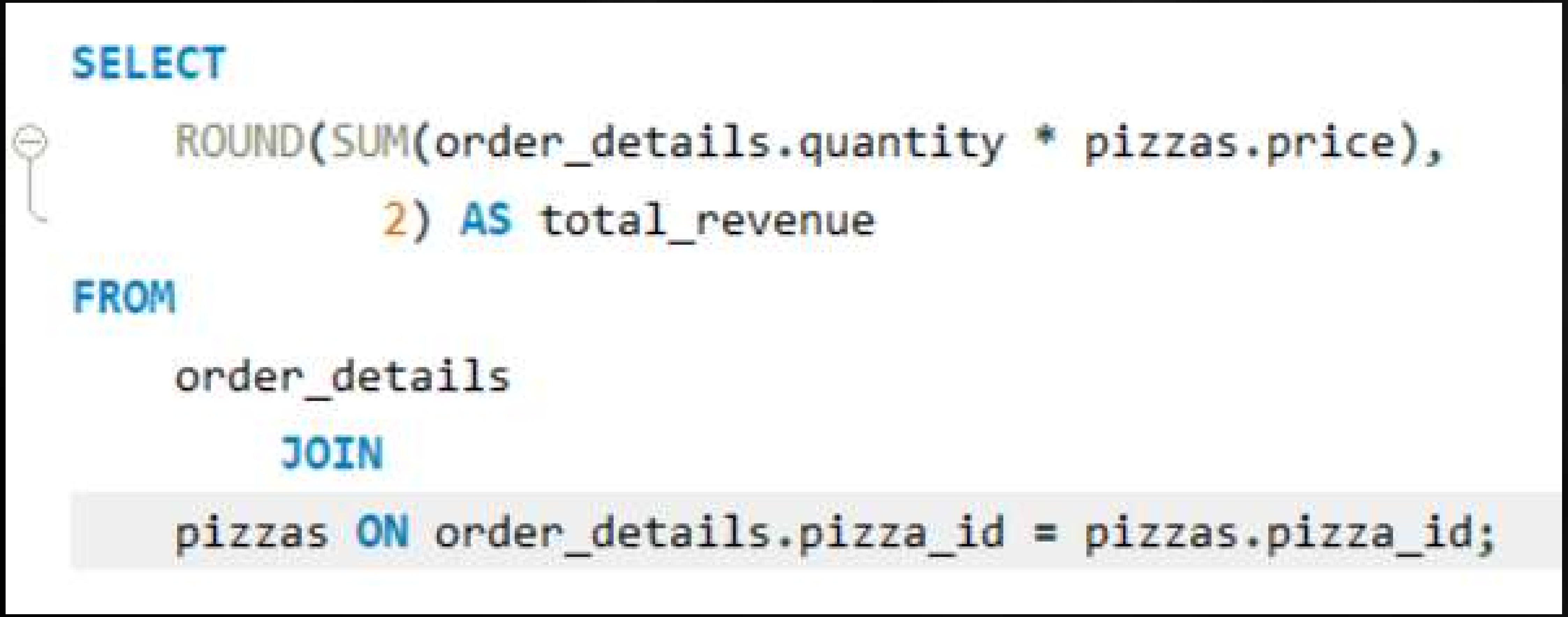


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

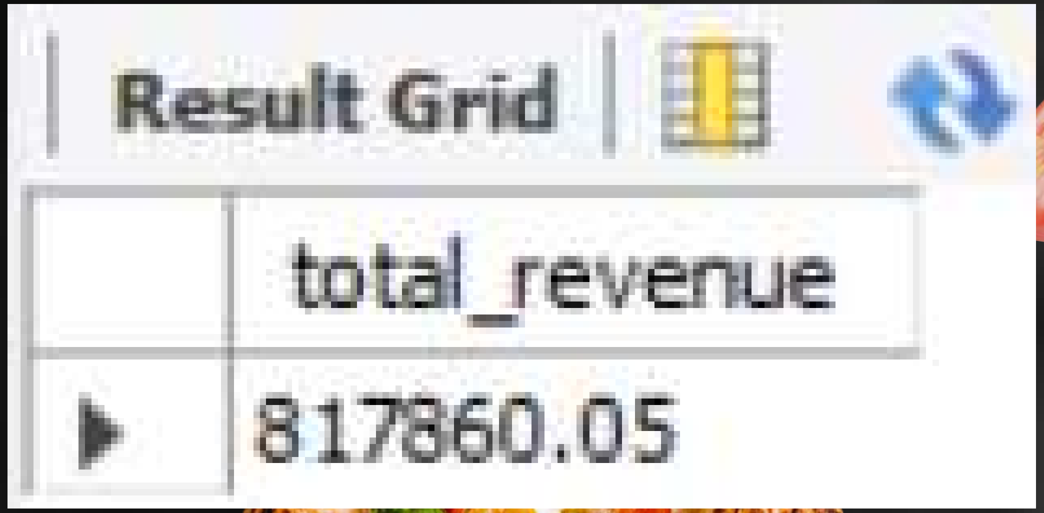


```
total_orders
21350
```

Calculate the total revenue generated from pizza sales.



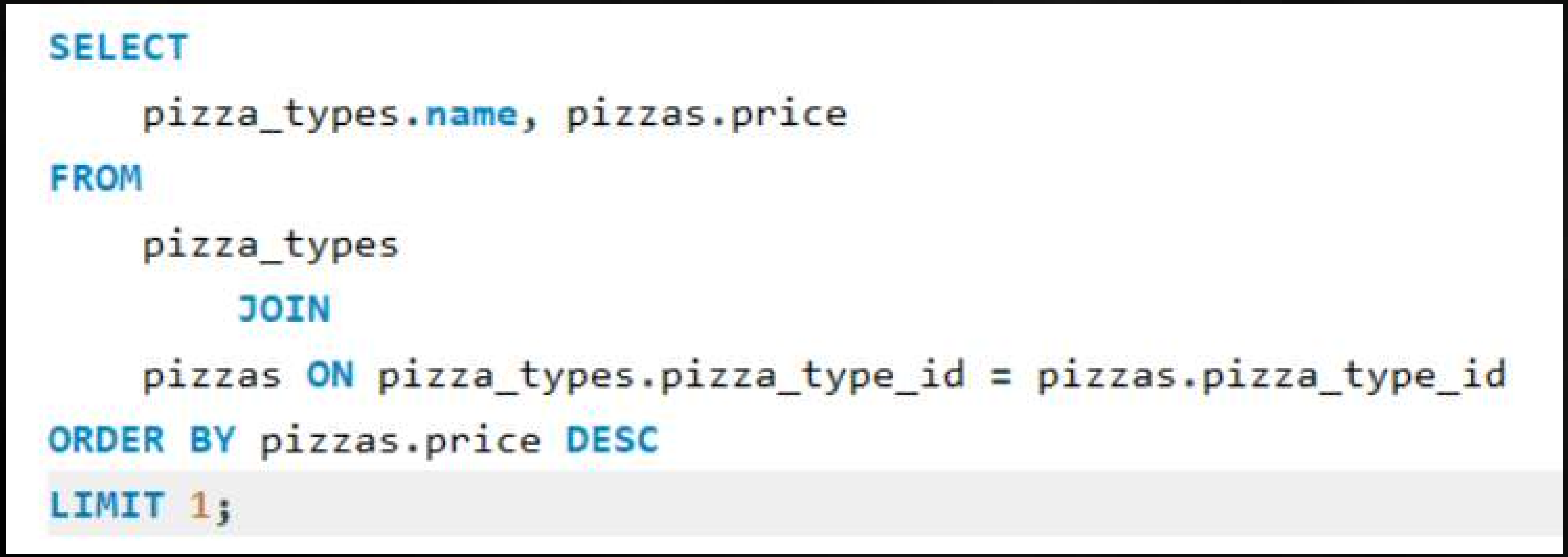
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



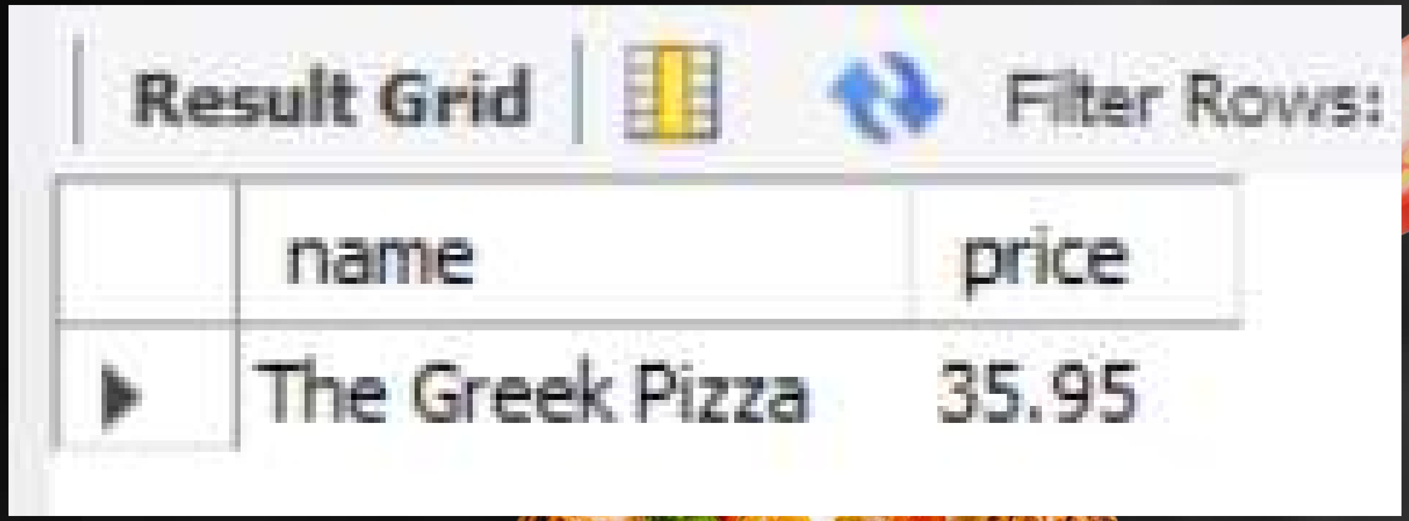
```
Result Grid

  total_revenue
▶ 817860.05
```

Identify the highest-priced pizza.

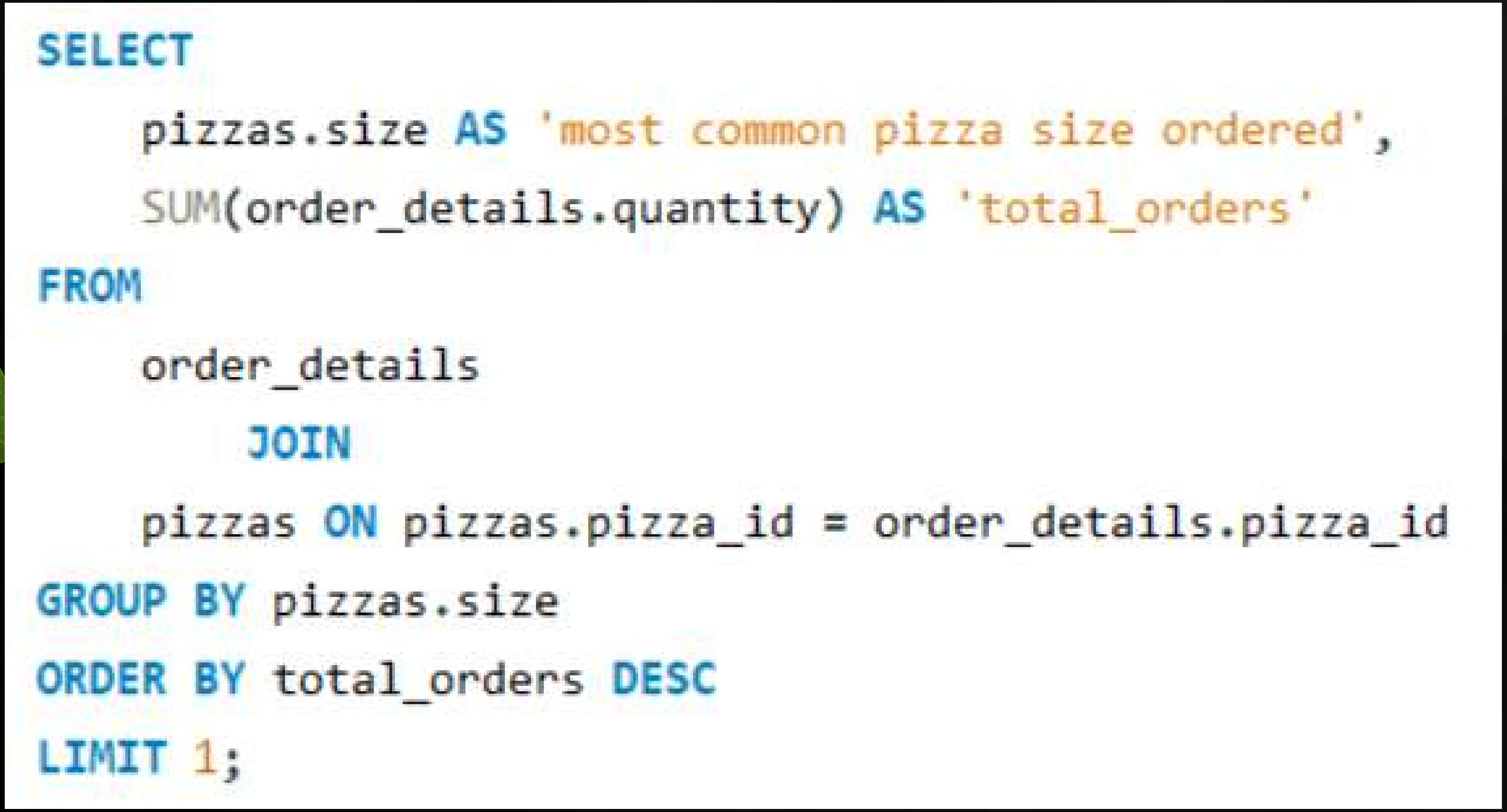


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

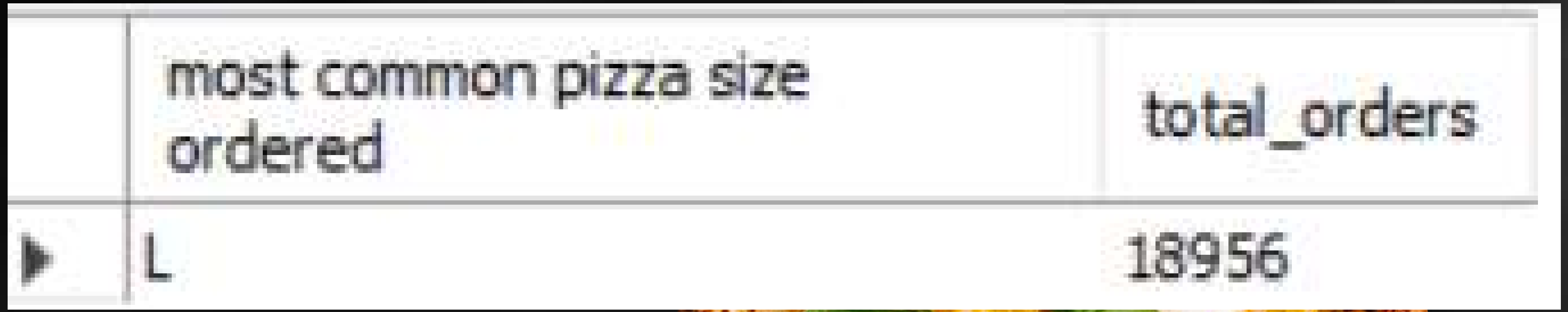


```
  name              price
▶ The Greek Pizza   35.95
```

Identify the most common pizza size ordered.

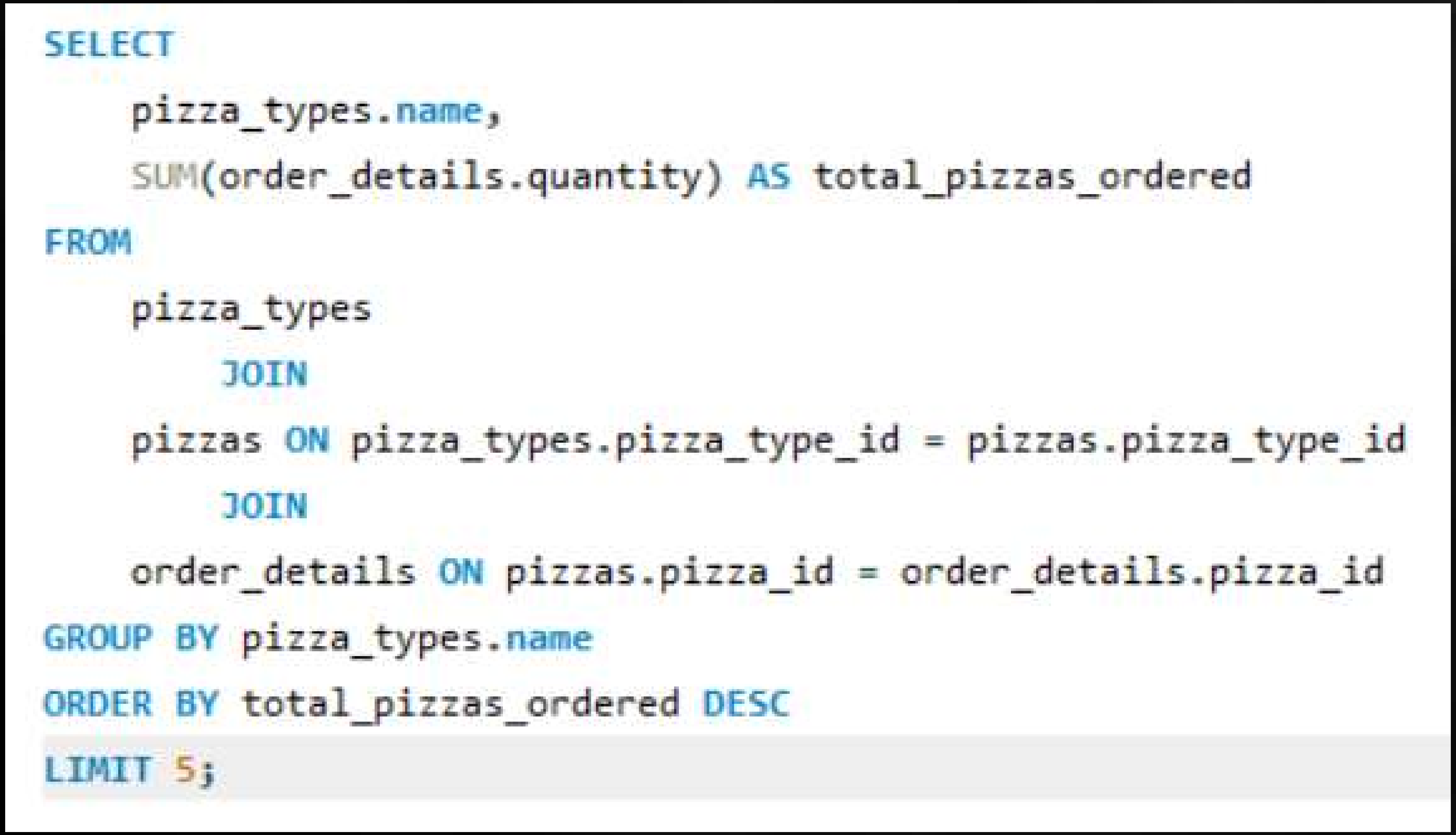


```
SELECT
    pizzas.size AS 'most common pizza size ordered',
    SUM(order_details.quantity) AS 'total_orders'
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_orders DESC
LIMIT 1;
```

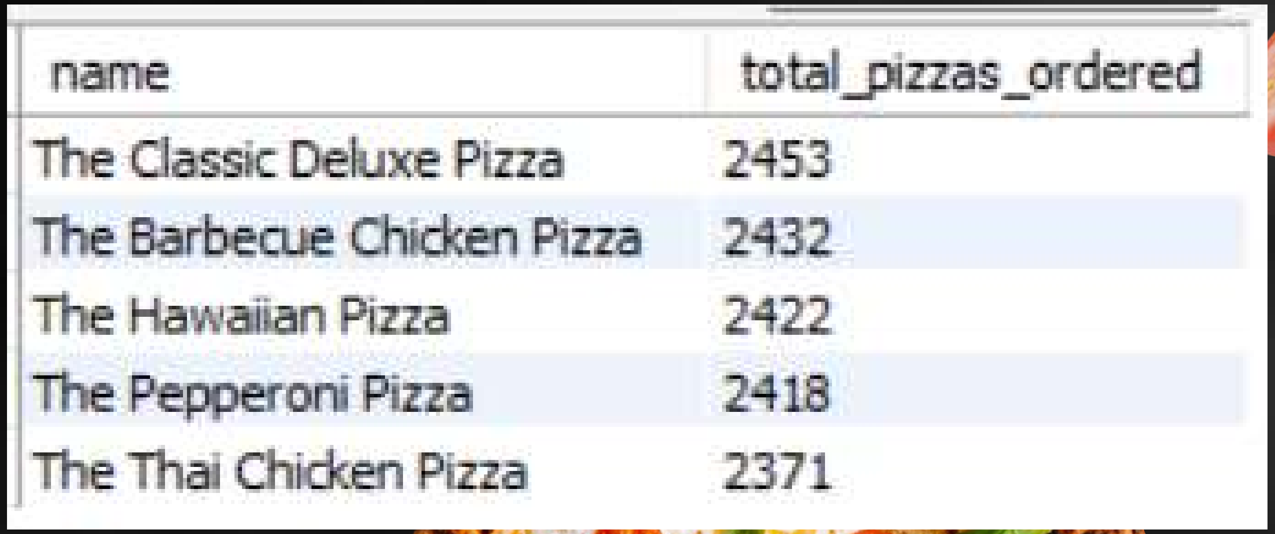


	most common pizza size ordered	total_orders
▶	L	18956

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

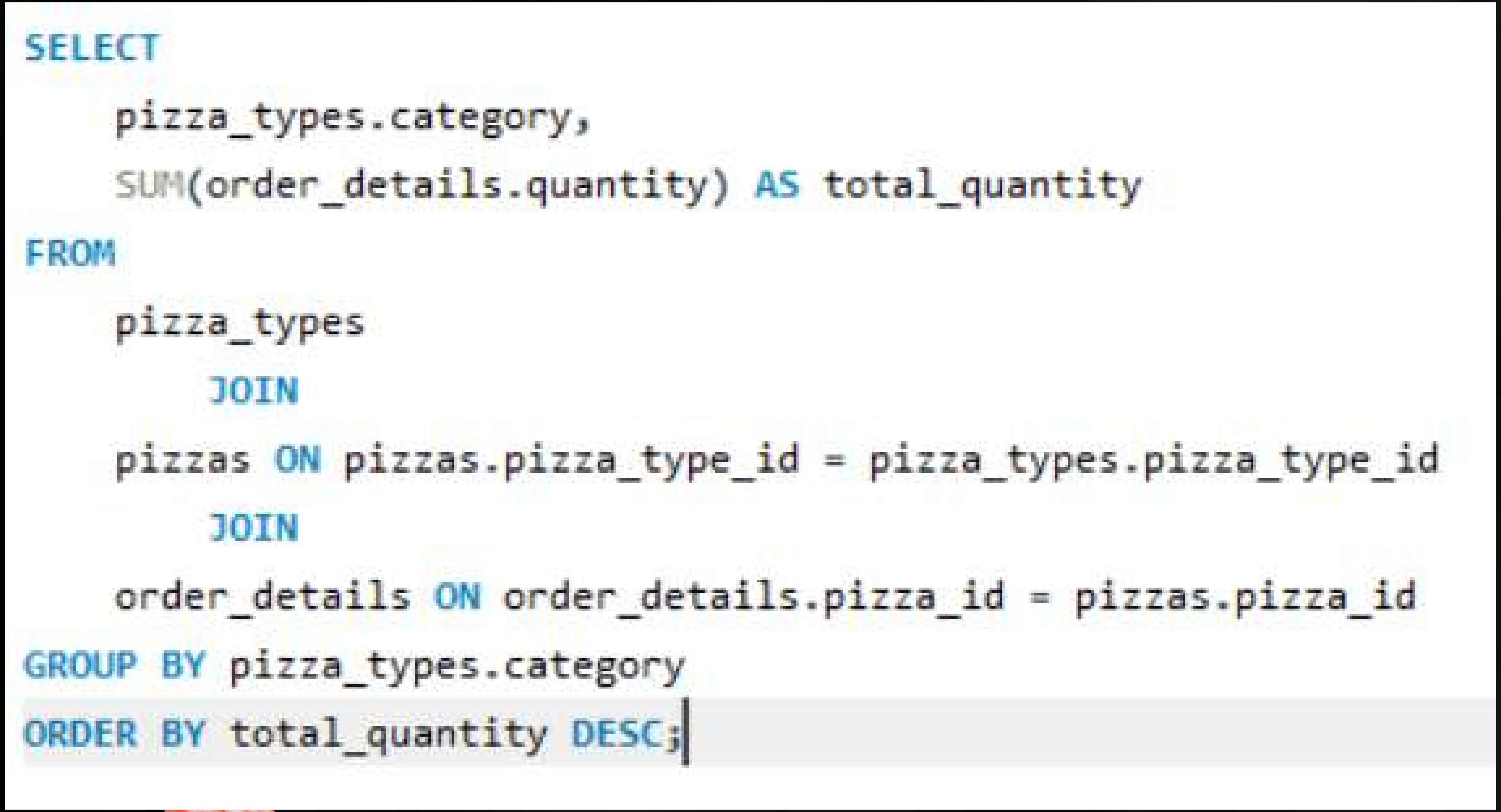


```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_pizzas_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_pizzas_ordered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

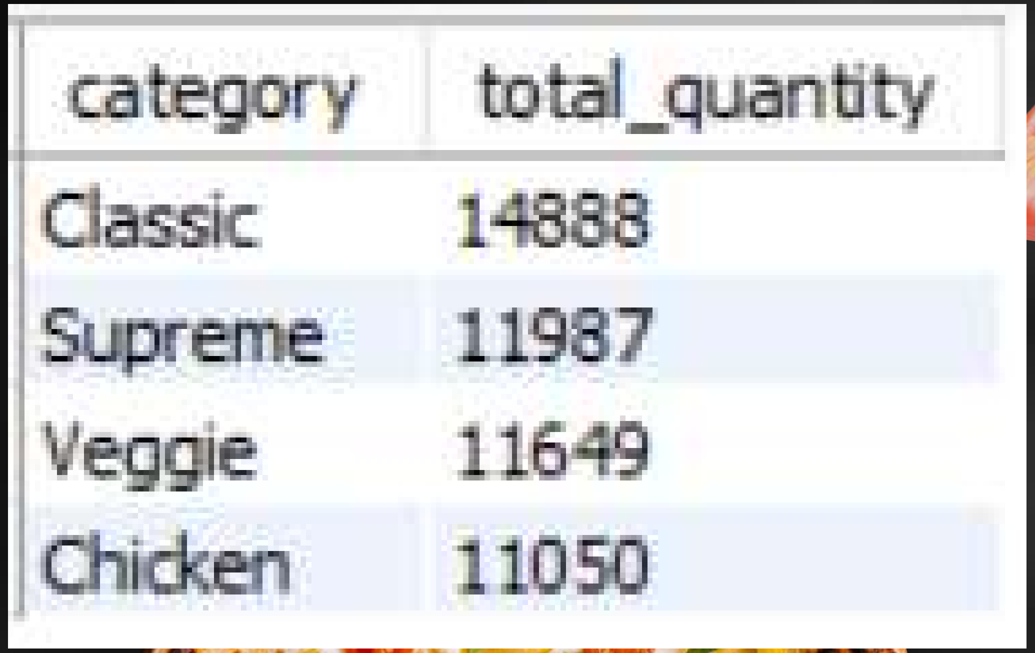


name	total_pizzas_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

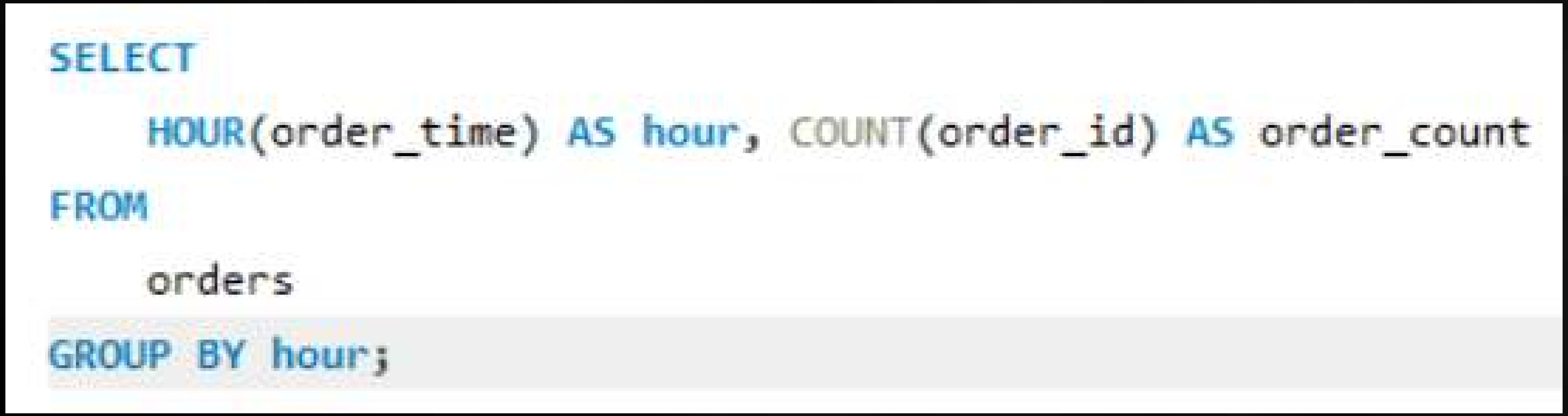


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity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

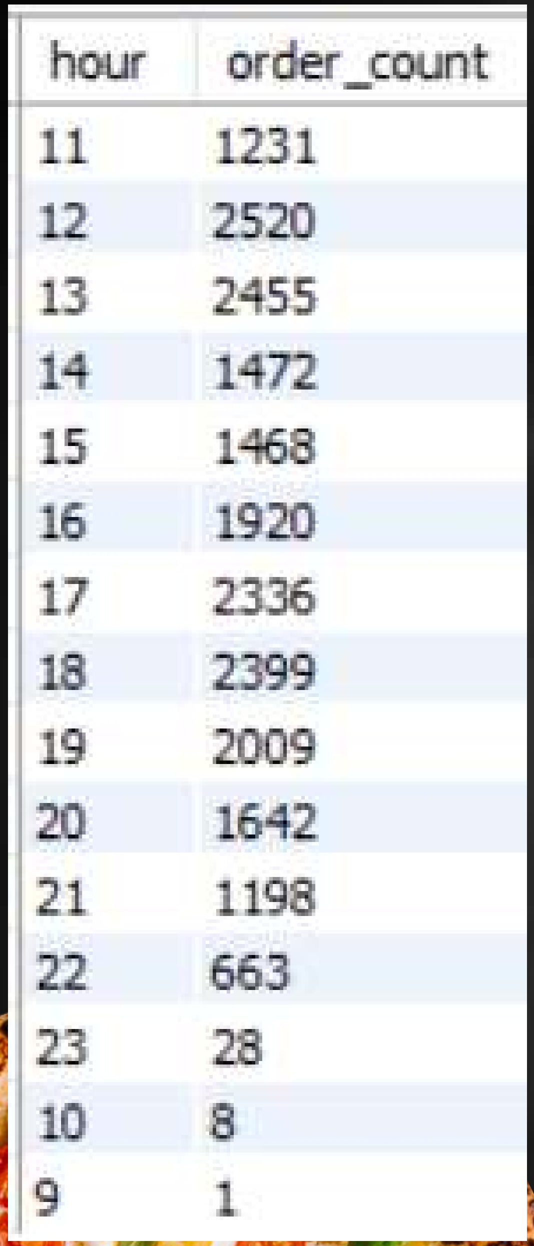


category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

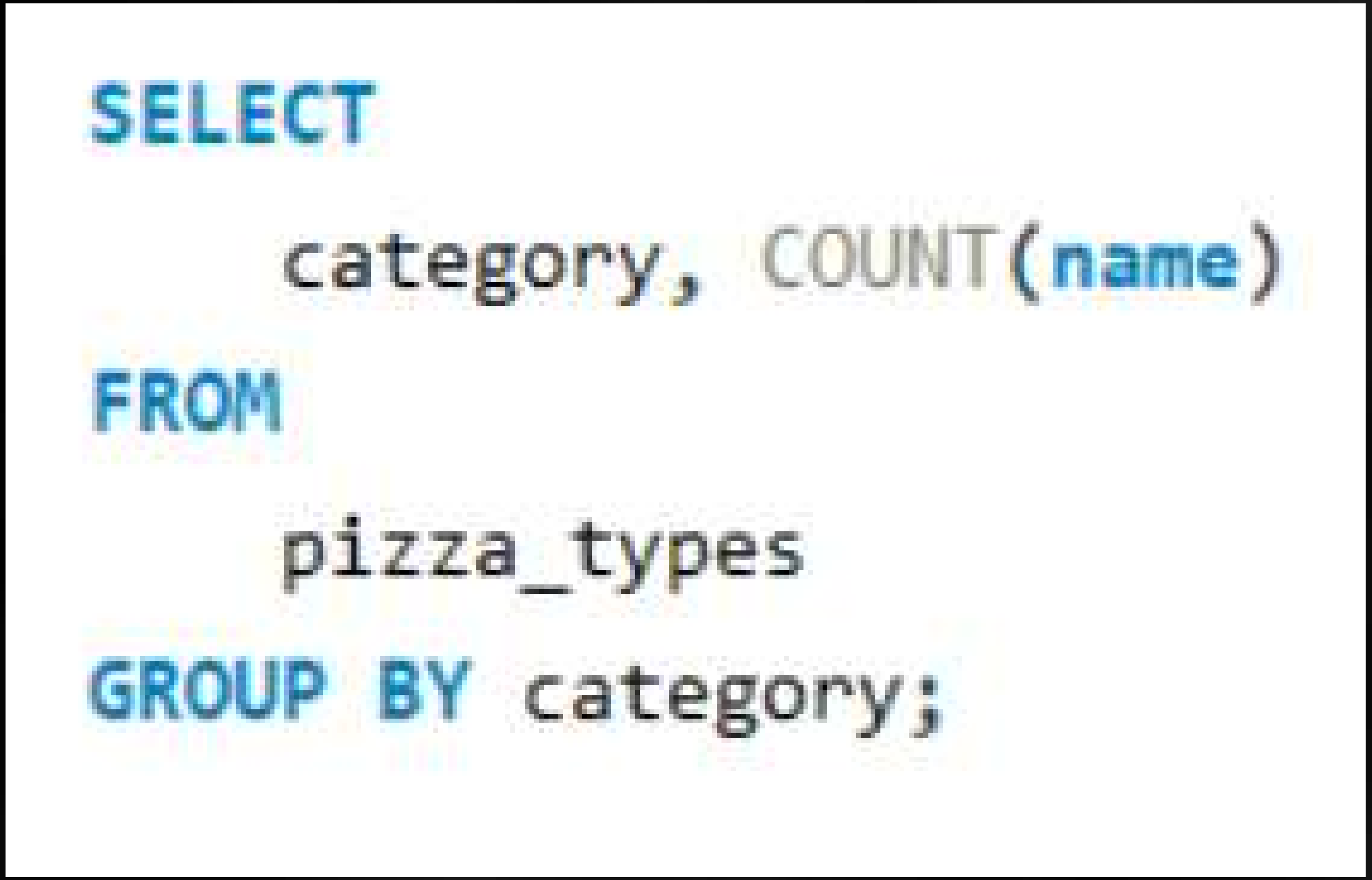


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour;
```

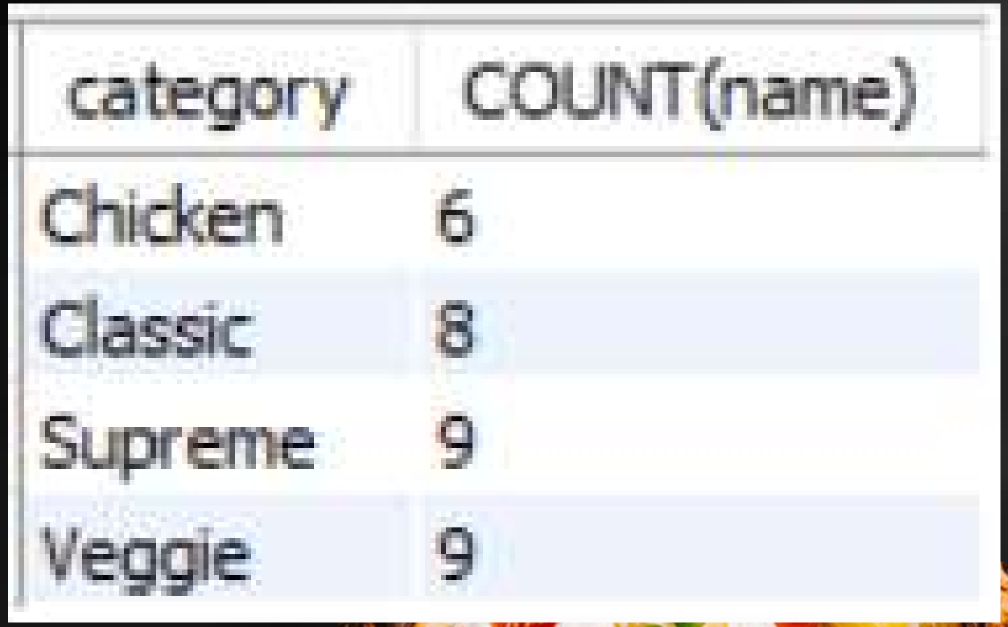


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

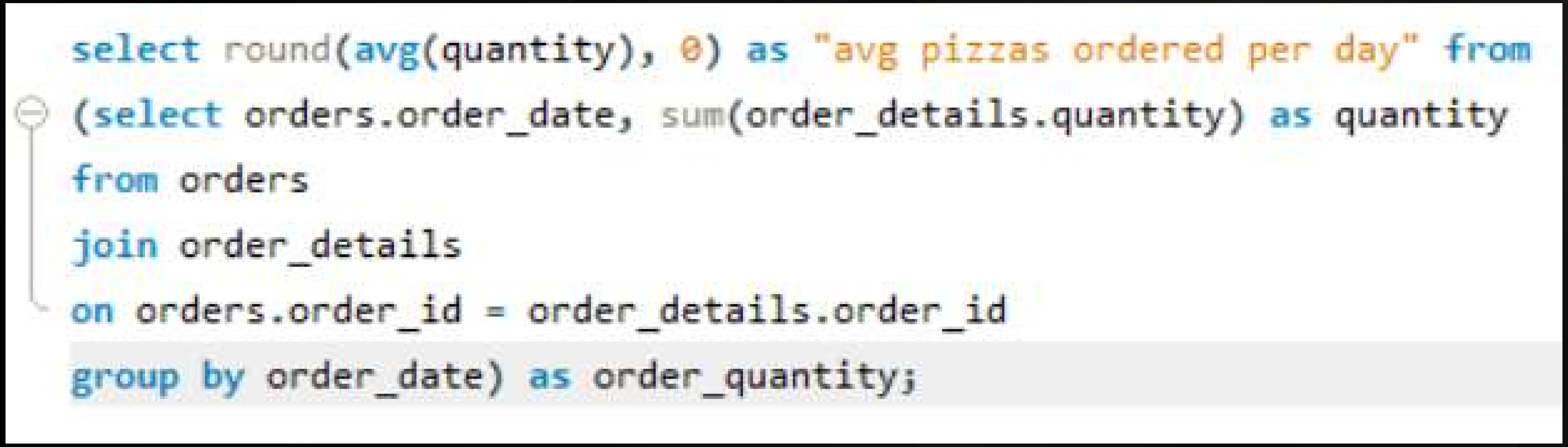


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

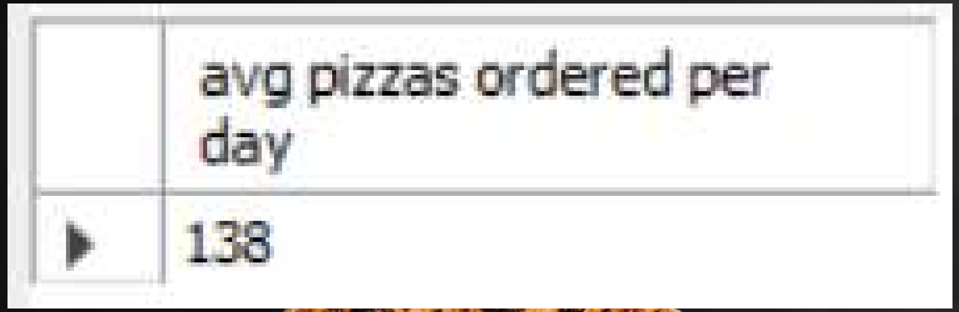


```
category   COUNT(name)
Chicken    6
Classic    8
Supreme    9
Veggie     9
```

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

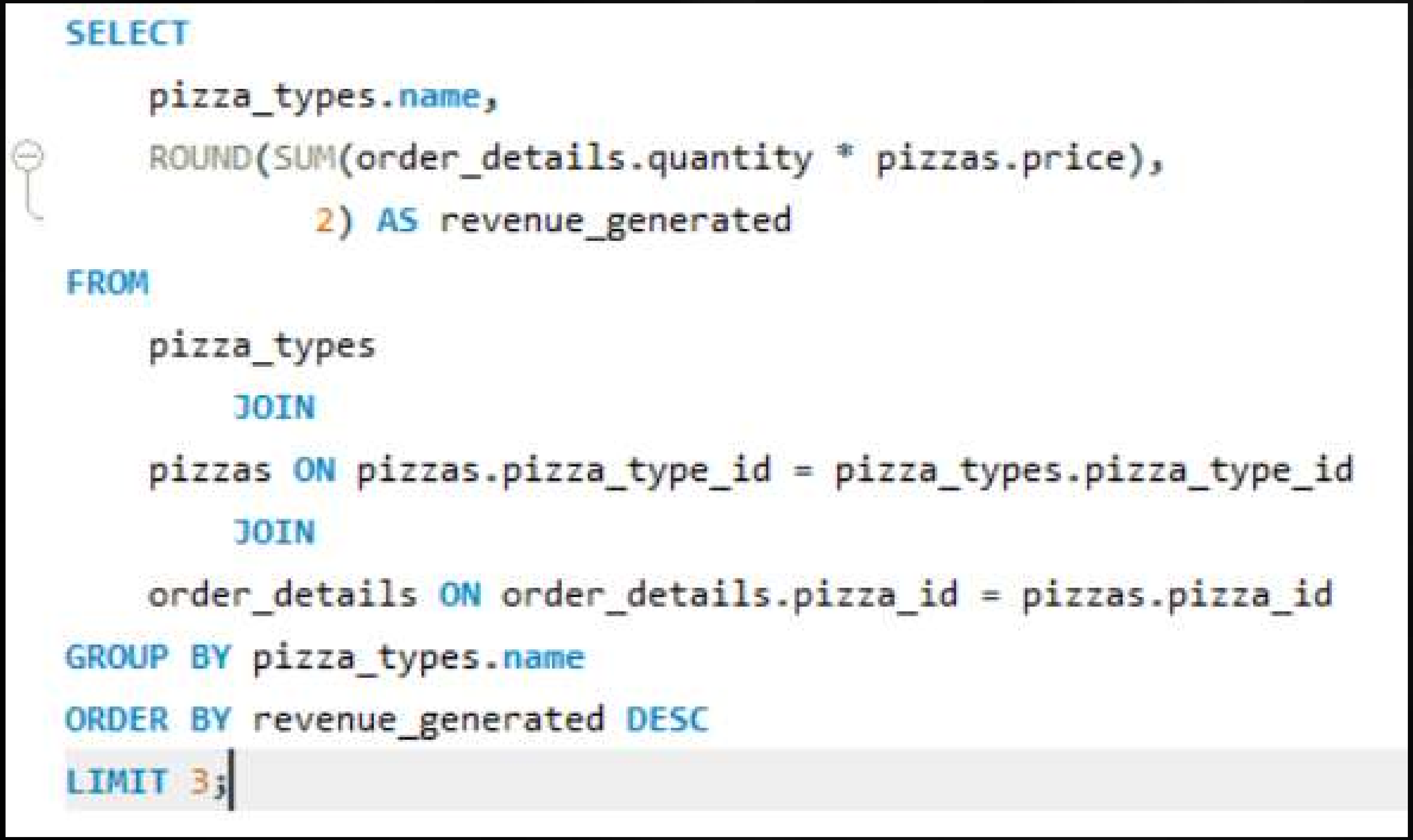


```sql
select round(avg(quantity), 0) as "avg pizzas ordered per day" from
(select orders.order_date, sum(order_details.quantity) as quantity
from orders
join order_details
on orders.order_id = order_details.order_id
group by order_date) as order_quantity;
```

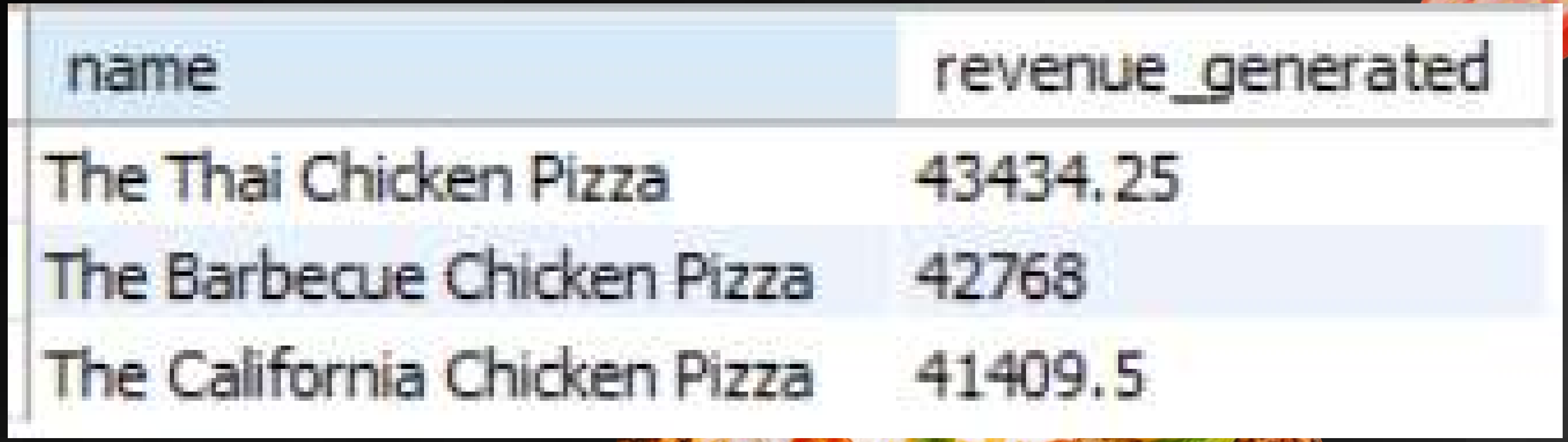


	avg pizzas ordered per day
▶	138

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

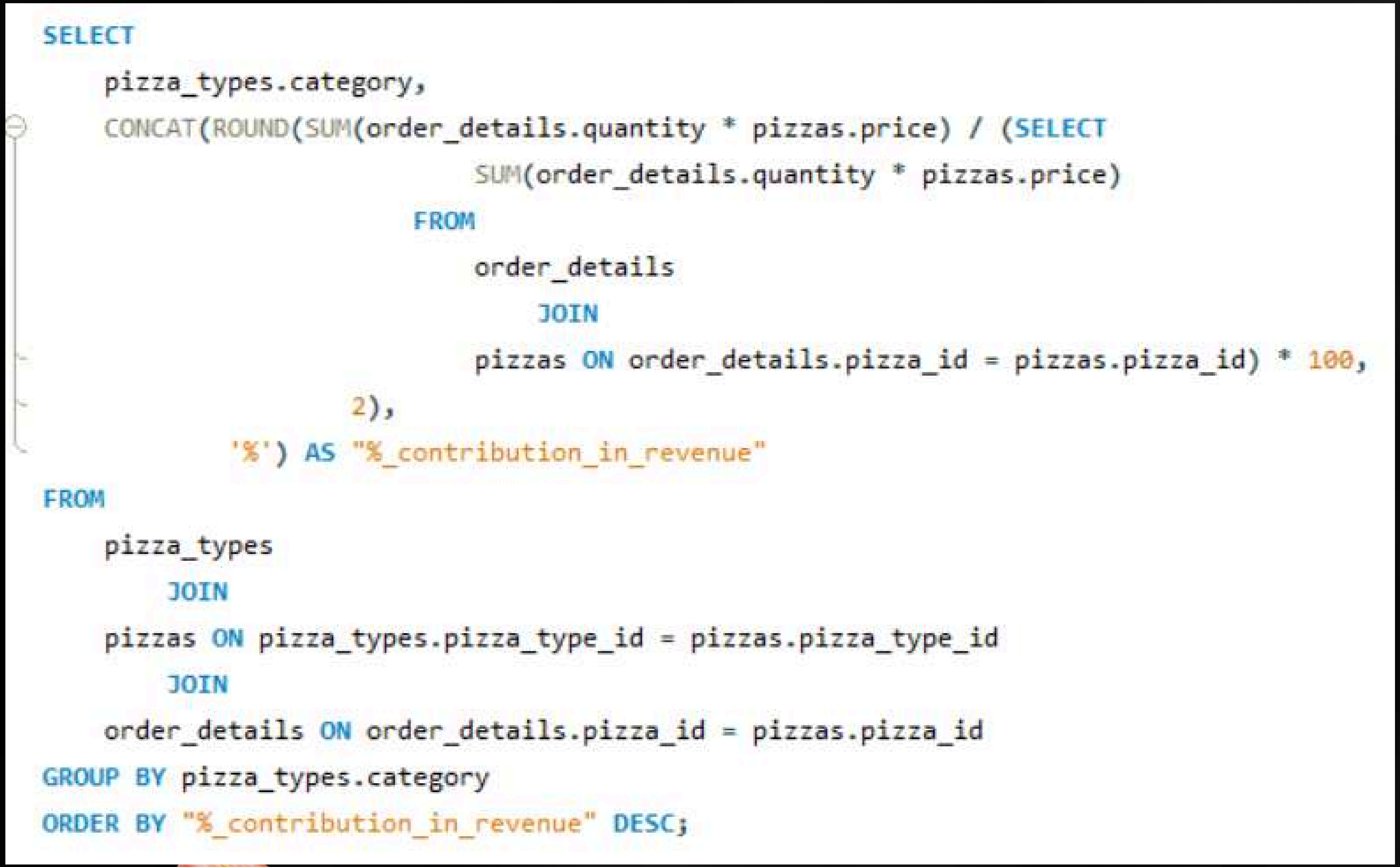


```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS revenue_generated
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP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

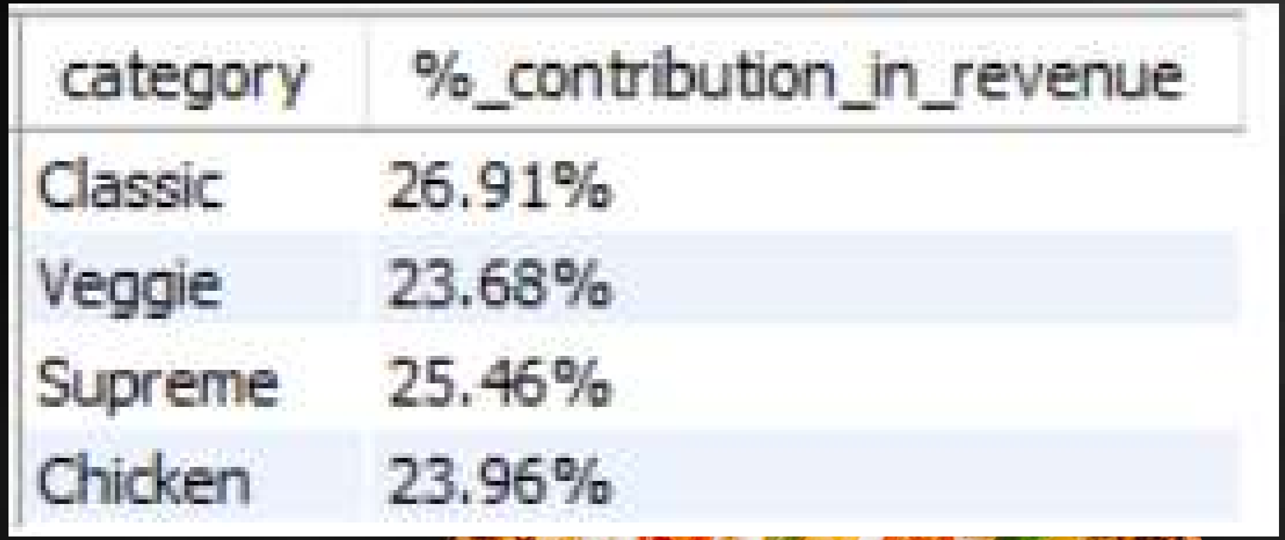


name	revenue_generated
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

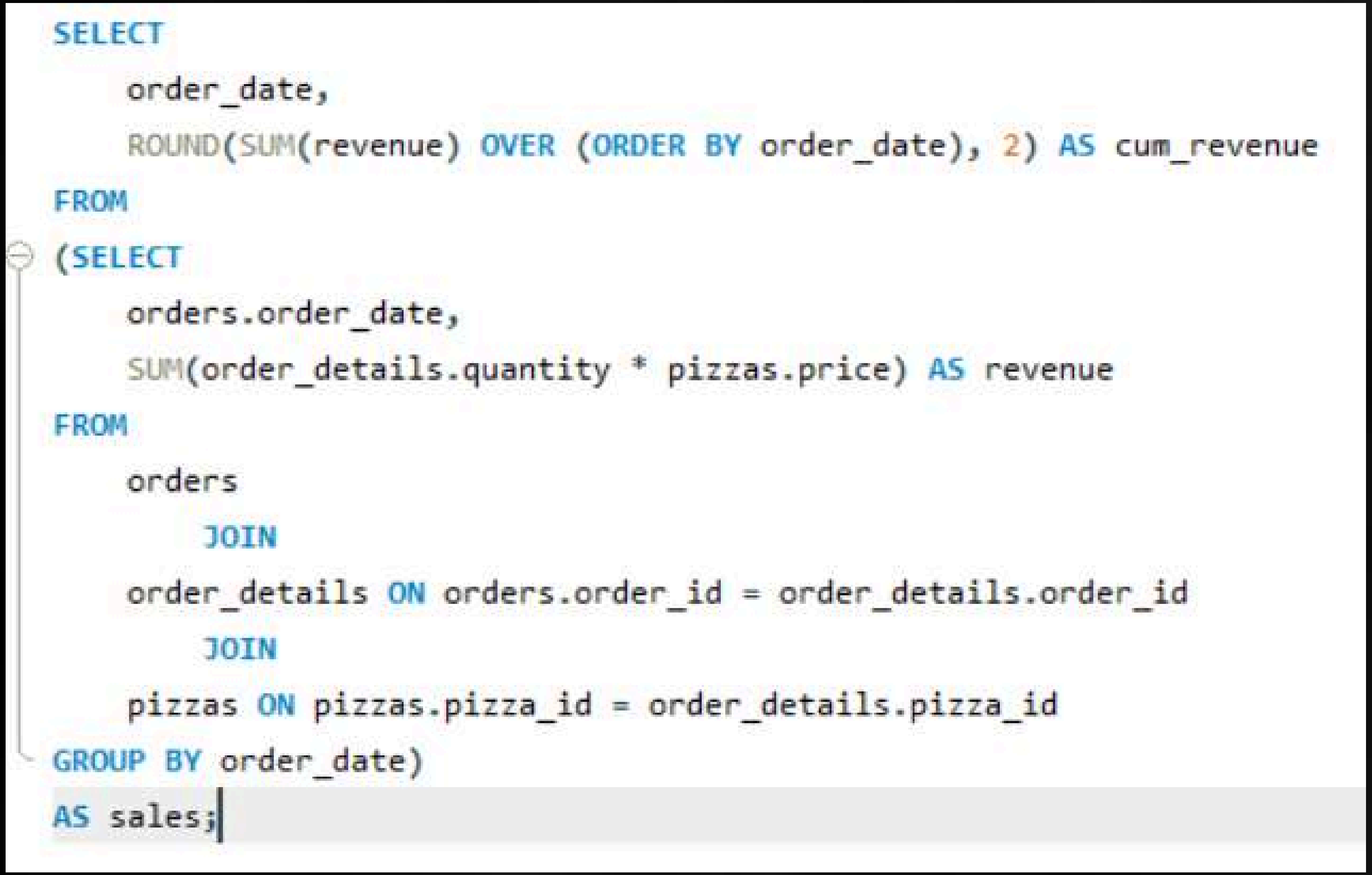


```
SELECT
    pizza_types.category,
    CONCAT(ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                        SUM(order_details.quantity * pizzas.price)
                    FROM
                        order_details
                            JOIN
                        pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
                2),
            '%') AS "%_contribution_in_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 "%_contribution_in_revenue" DESC;
```



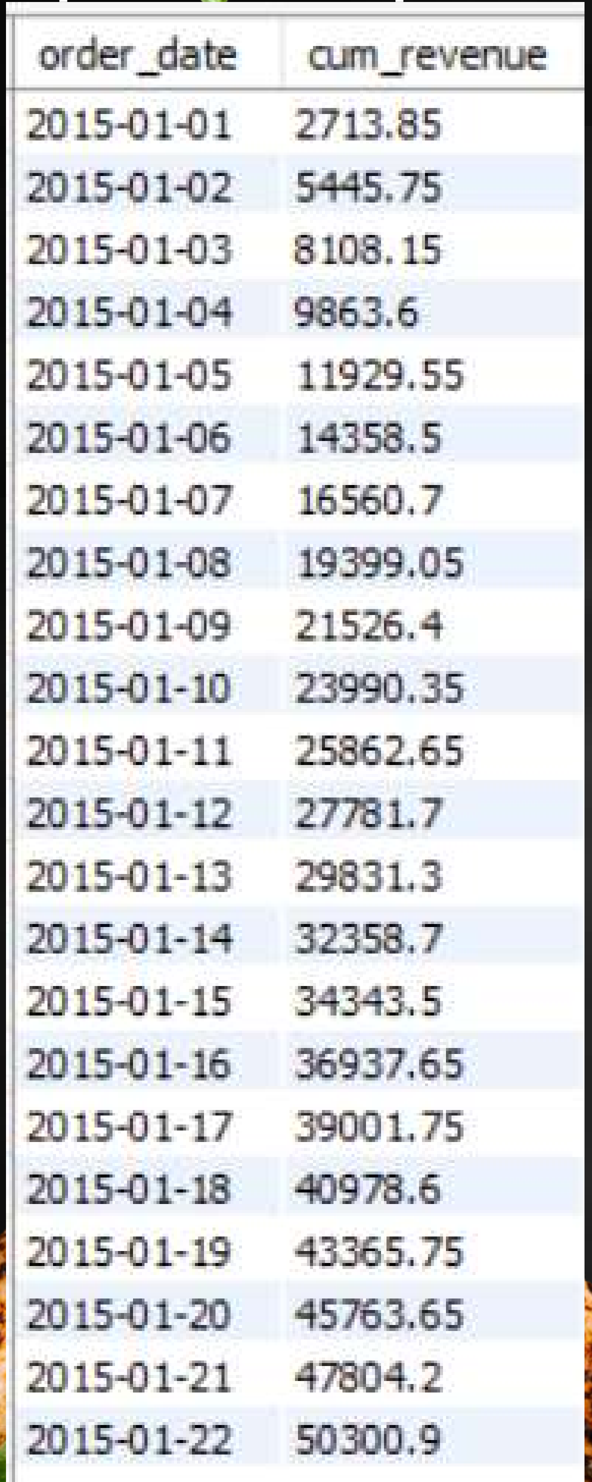
category	%_contribution_in_revenue
Classic	26.91%
Veggie	23.68%
Supreme	25.46%
Chicken	23.96%

Analyze the cumulative revenue generated over time.



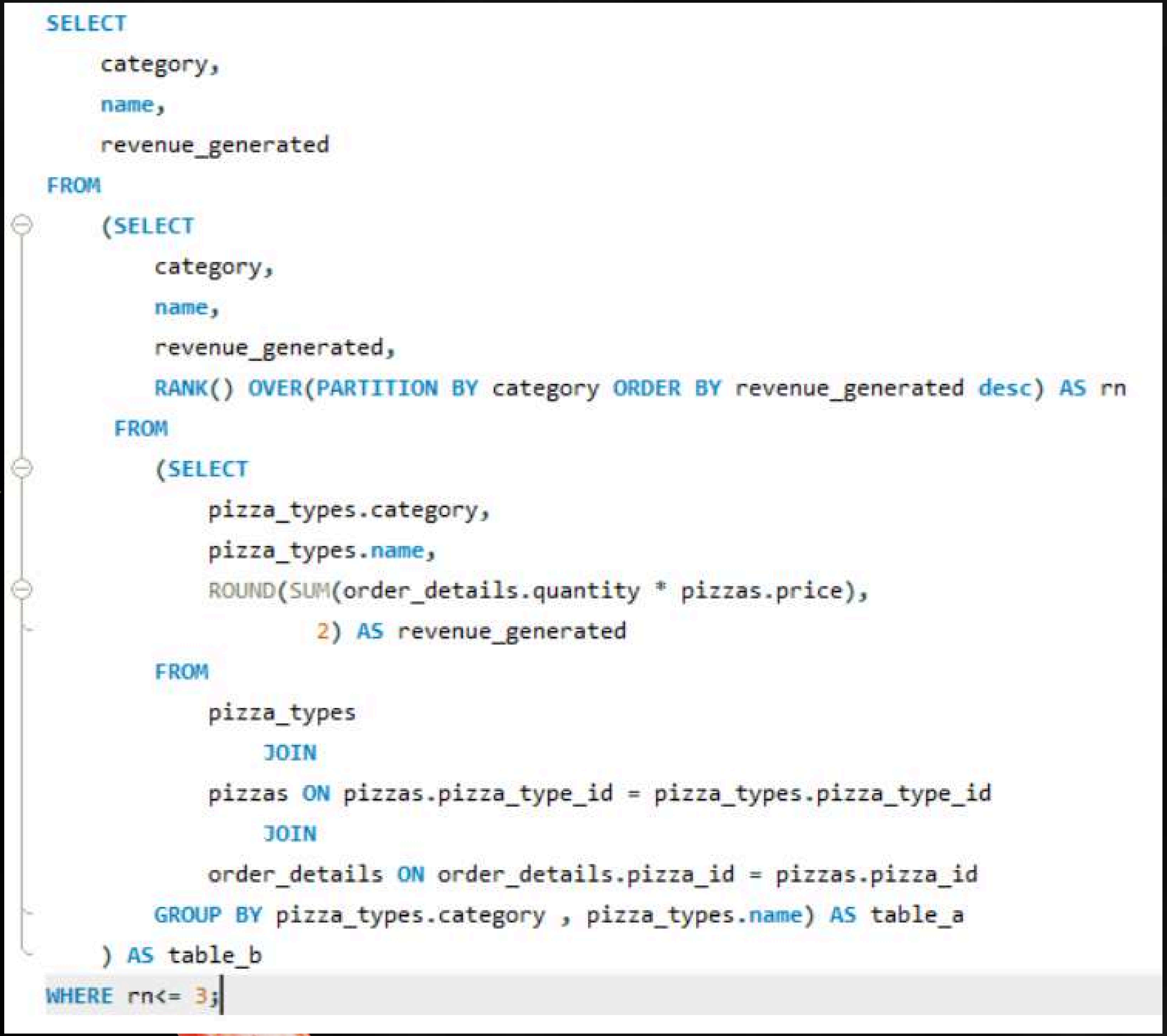
```
SELECT
    order_date,
    ROUND(SUM(revenue) OVER (ORDER BY order_date), 2) AS cum_revenue
FROM
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY order_date)
AS sales;
```

sample output:

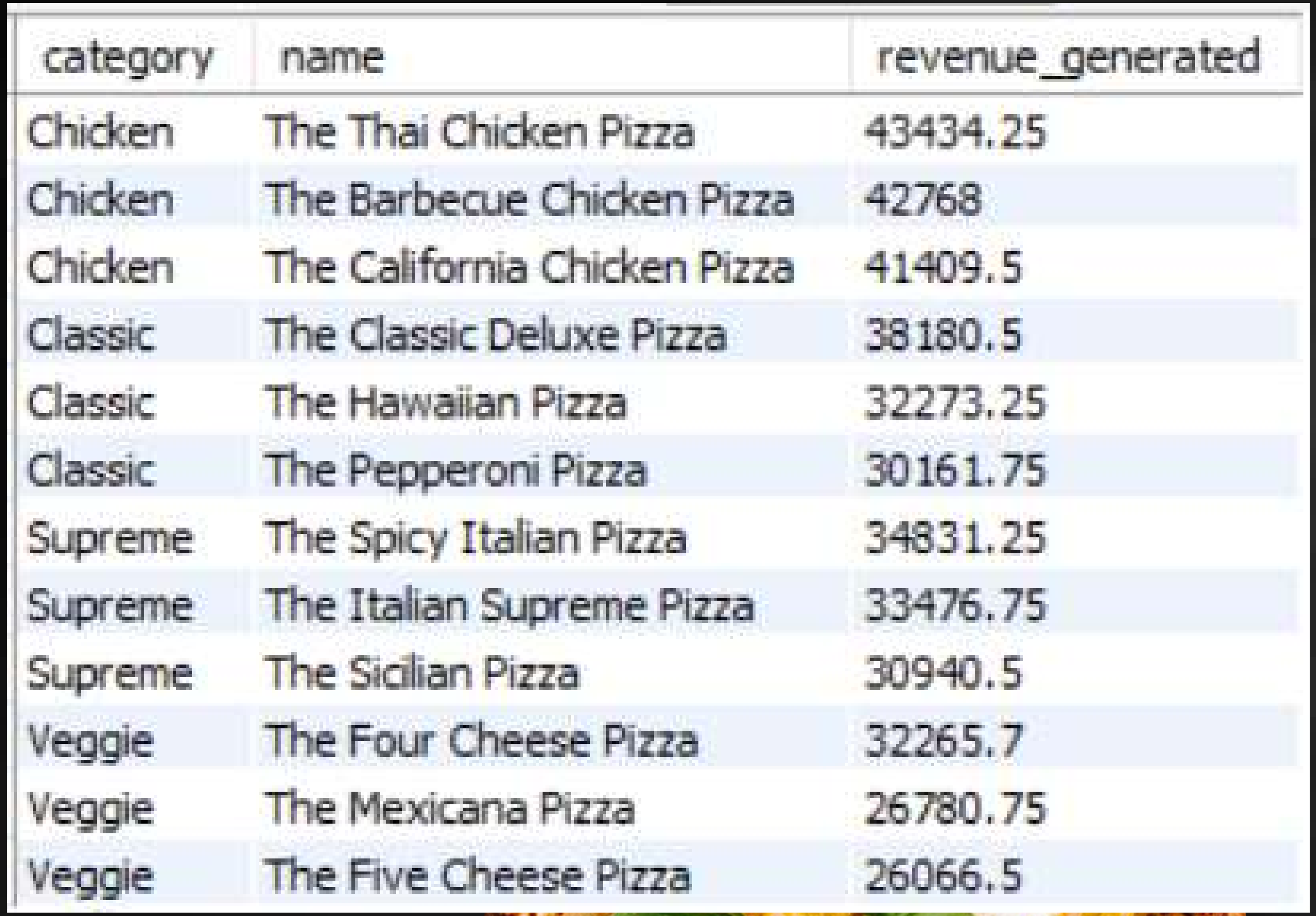


order_date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9

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



```sql
SELECT
    category,
    name,
    revenue_generated
FROM
    (SELECT
        category,
        name,
        revenue_generated,
        RANK() OVER(PARTITION BY category ORDER BY revenue_generated desc) AS rn
     FROM
        (SELECT
            pizza_types.category,
            pizza_types.name,
            ROUND(SUM(order_details.quantity * pizzas.price),
                    2) AS revenue_generated
        FROM
            pizza_types
                JOIN
            pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
                JOIN
            order_details ON order_details.pizza_id = pizzas.pizza_id
        GROUP BY pizza_types.category , pizza_types.name) AS table_a
    ) AS table_b
WHERE rn<= 3;
```



category	name	revenue_generated
Chicken	The Thai Chicken Pizza	43434.25
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
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU!