PIZZA SALES ANALYSIS USING SQL

-Nidhi Sharma
16,Sep,2024

Project Overview:

In this project, I conducted an analysis of pizza sales data from the PizzaHut database using SQL. The analysis provides insights into key business metrics such as revenue, popular pizza types, and order patterns. By examining both basic and advanced queries, this project offers a comprehensive view of sales performance, customer preferences, and operational trends.

Objectives:

- To retrieve essential business metrics (e.g., total orders, revenue).
- To analyze pizza preferences based on size, type, and category.
- To investigate order patterns by time and day.
- To explore pizza sales trends and determine key revenue drivers.

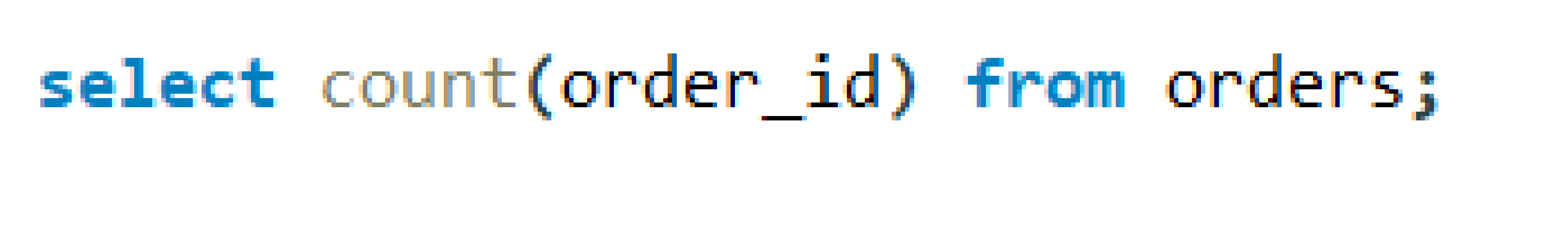
Dataset Description

PizzaHut Database Overview

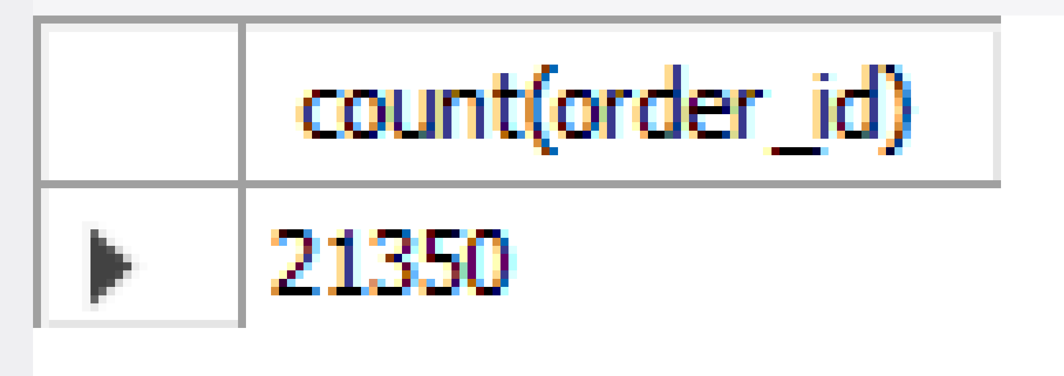
The PizzaHut database consists of 4 key tables:

1. order_details: Contains details of each pizza ordered.
   - ○ Columns: order_id, pizza_id, quantity
2. orders: Stores information about when each order was placed.
   - ○ Columns: order_date, order_time
3. pizzas: Contains information about each pizza.
   - ○ Columns: pizza_id, pizza_type_id, size, price
4. pizza_types: Stores details about the pizza type and its ingredients.
   - ○ Columns: pizza_type_id, name, category, ingredients

Retrieve the total number of orders placed

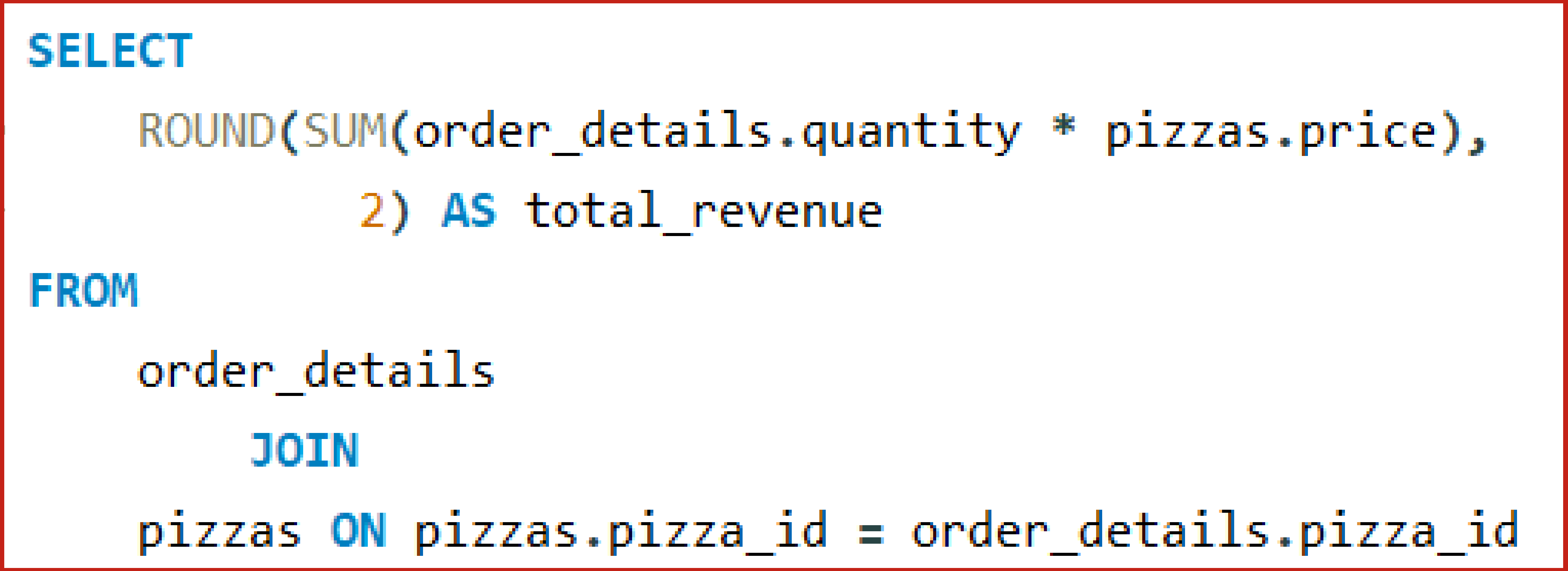


```sql
select count(order_id) from orders;
```

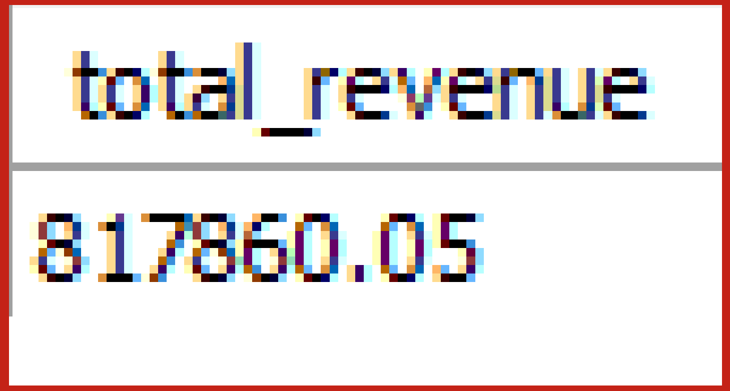


```
   count(order_id)
▶  21350
```

Calculate the total revenue generated from pizza sales

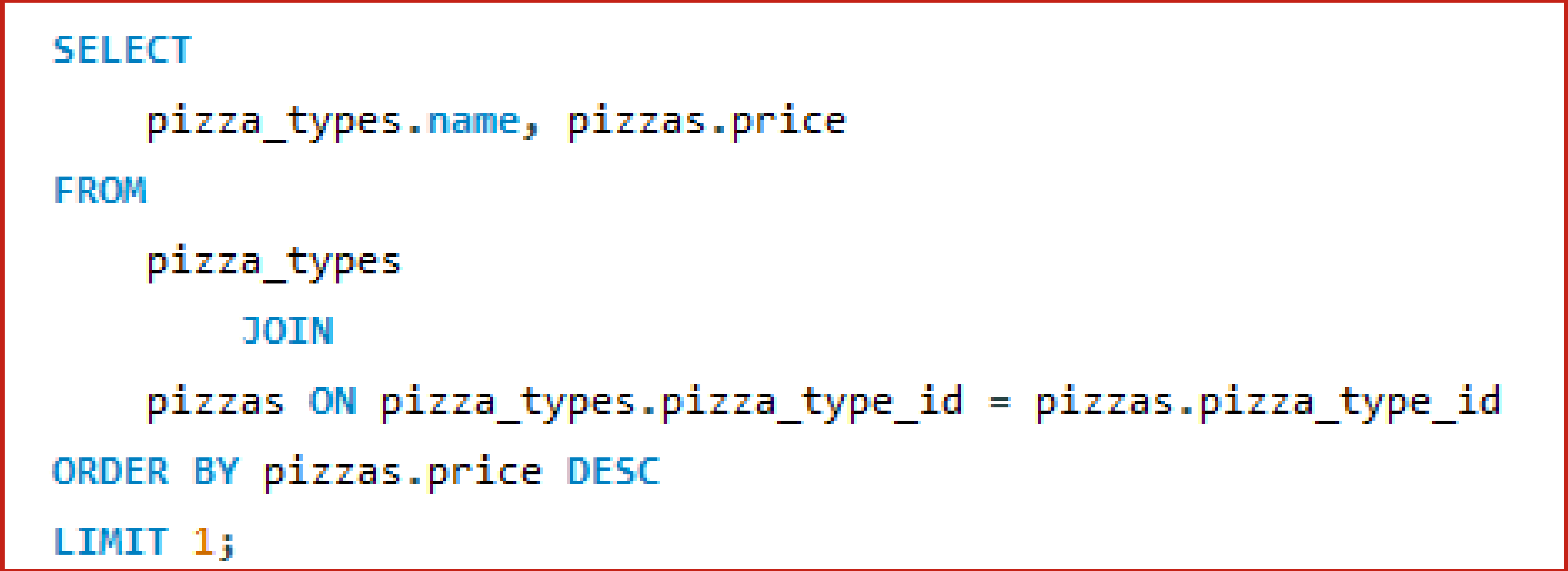


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

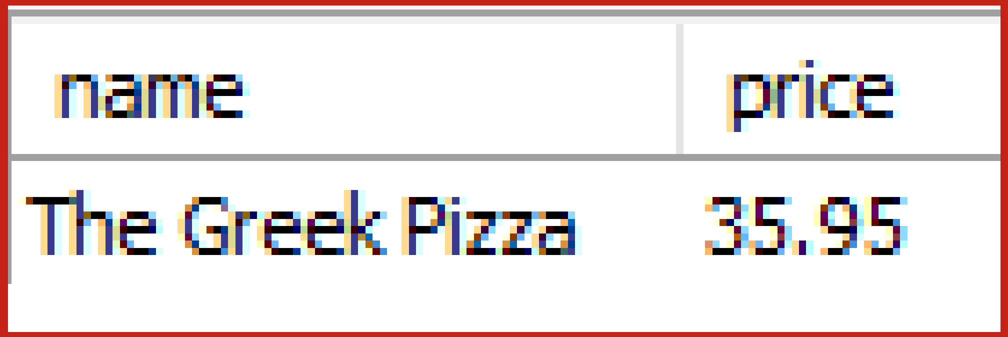


total_revenue
817860.05

Identify the highest-priced pizza

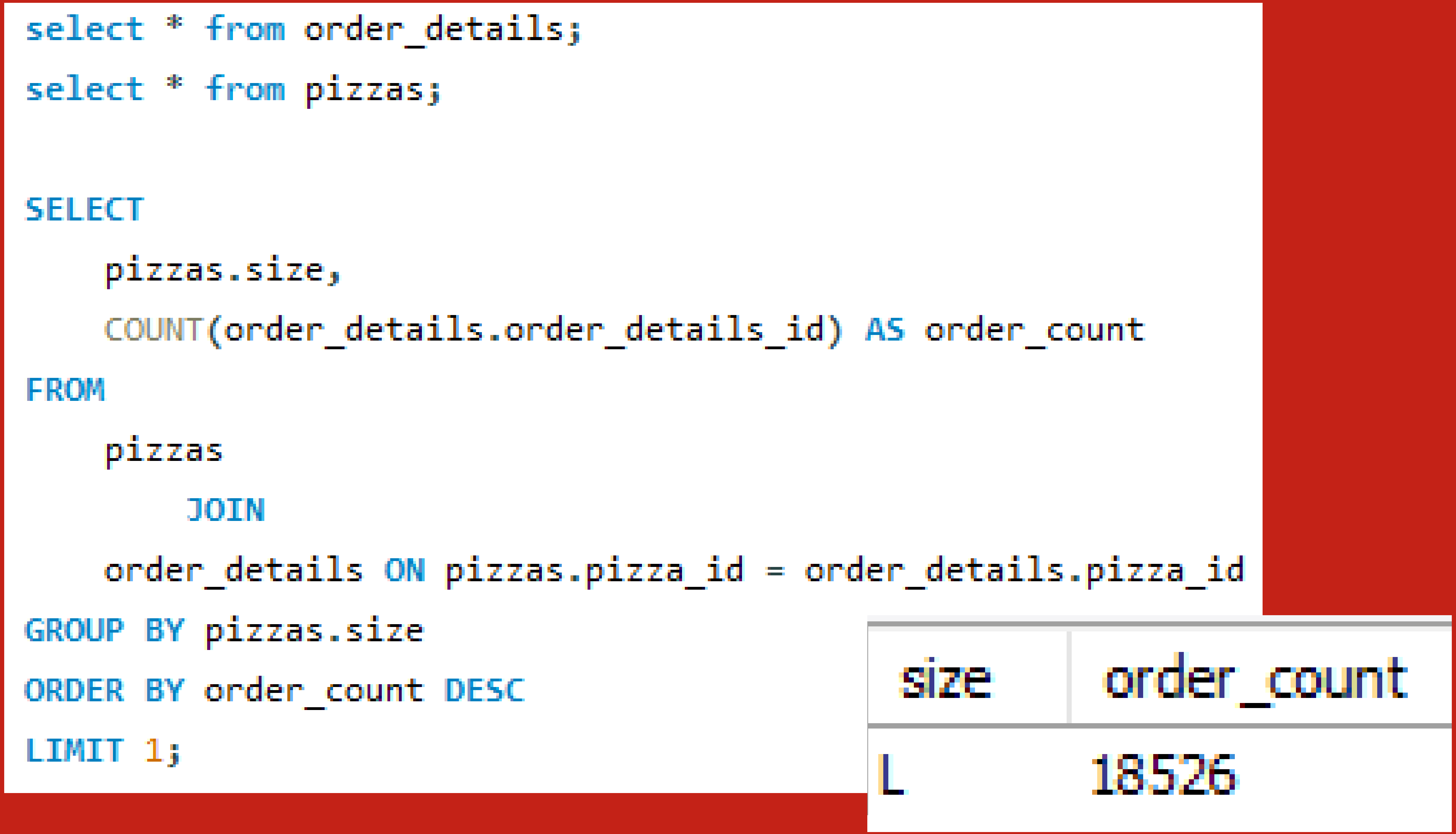


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



name	price
The Greek Pizza	35.95

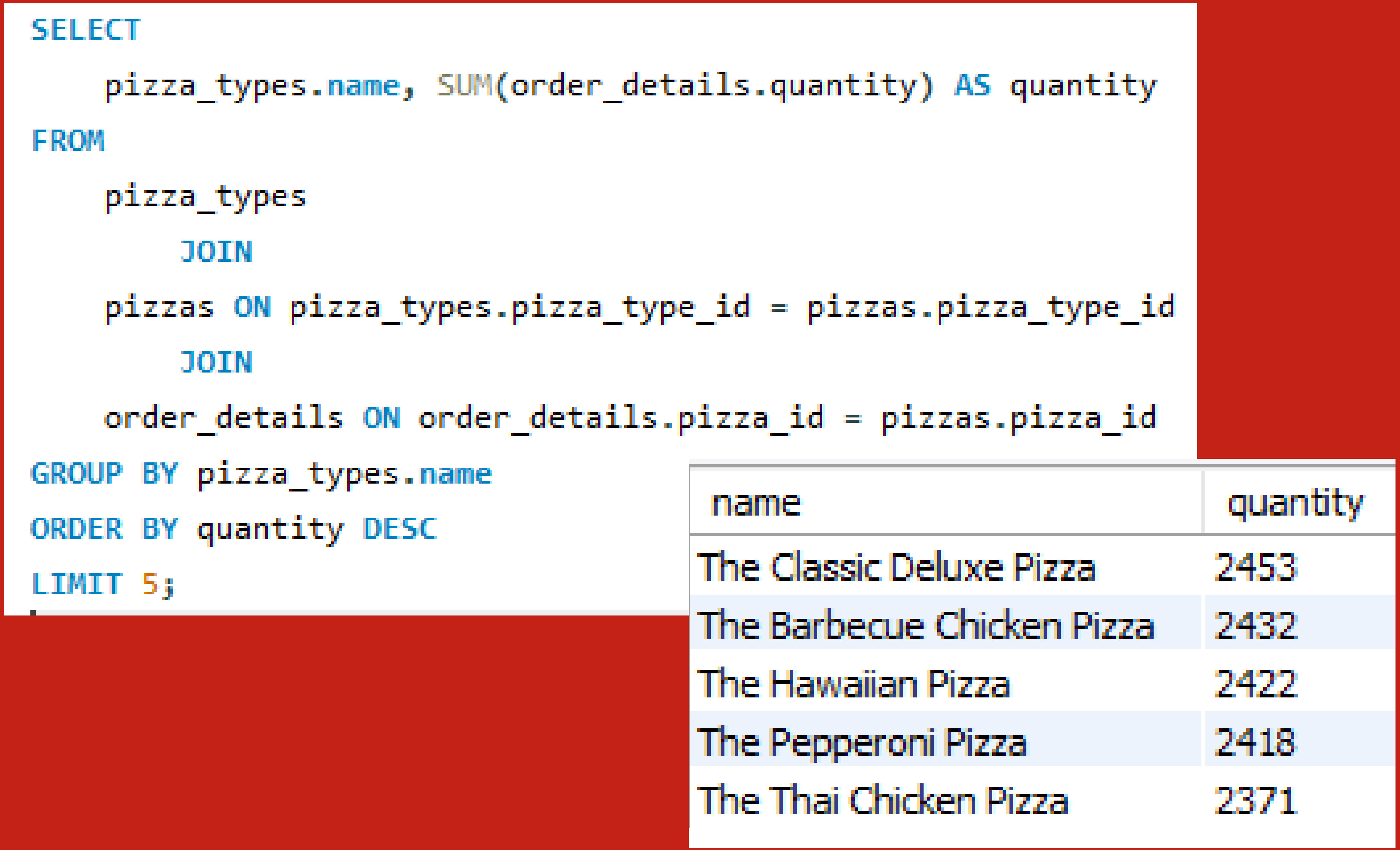
Identify the most common pizza size ordered



```
select * from order_details;
select * from pizzas;


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

size	order_count
L	18526

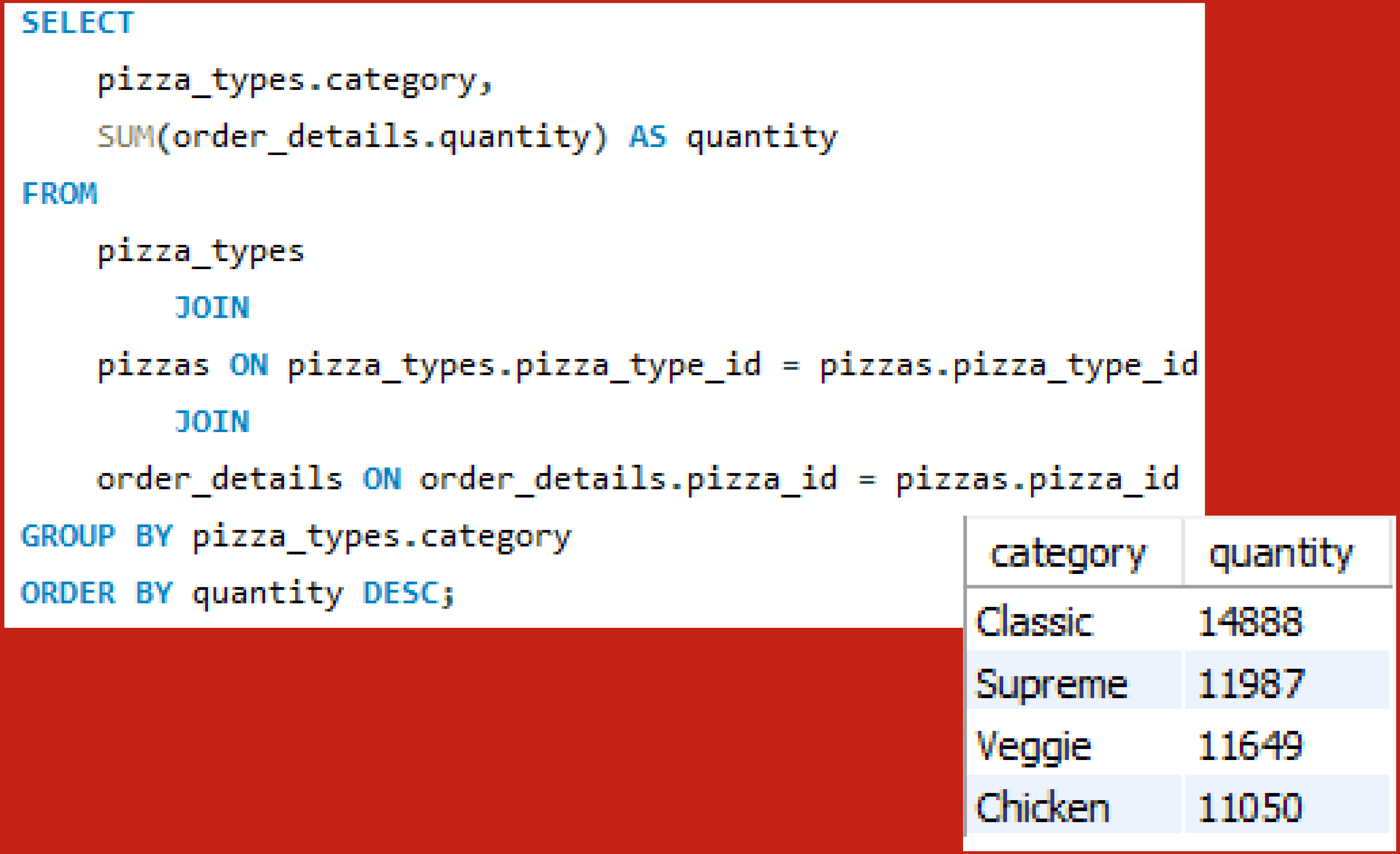
List the top 5 most ordered pizza types along with their quantities



```
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

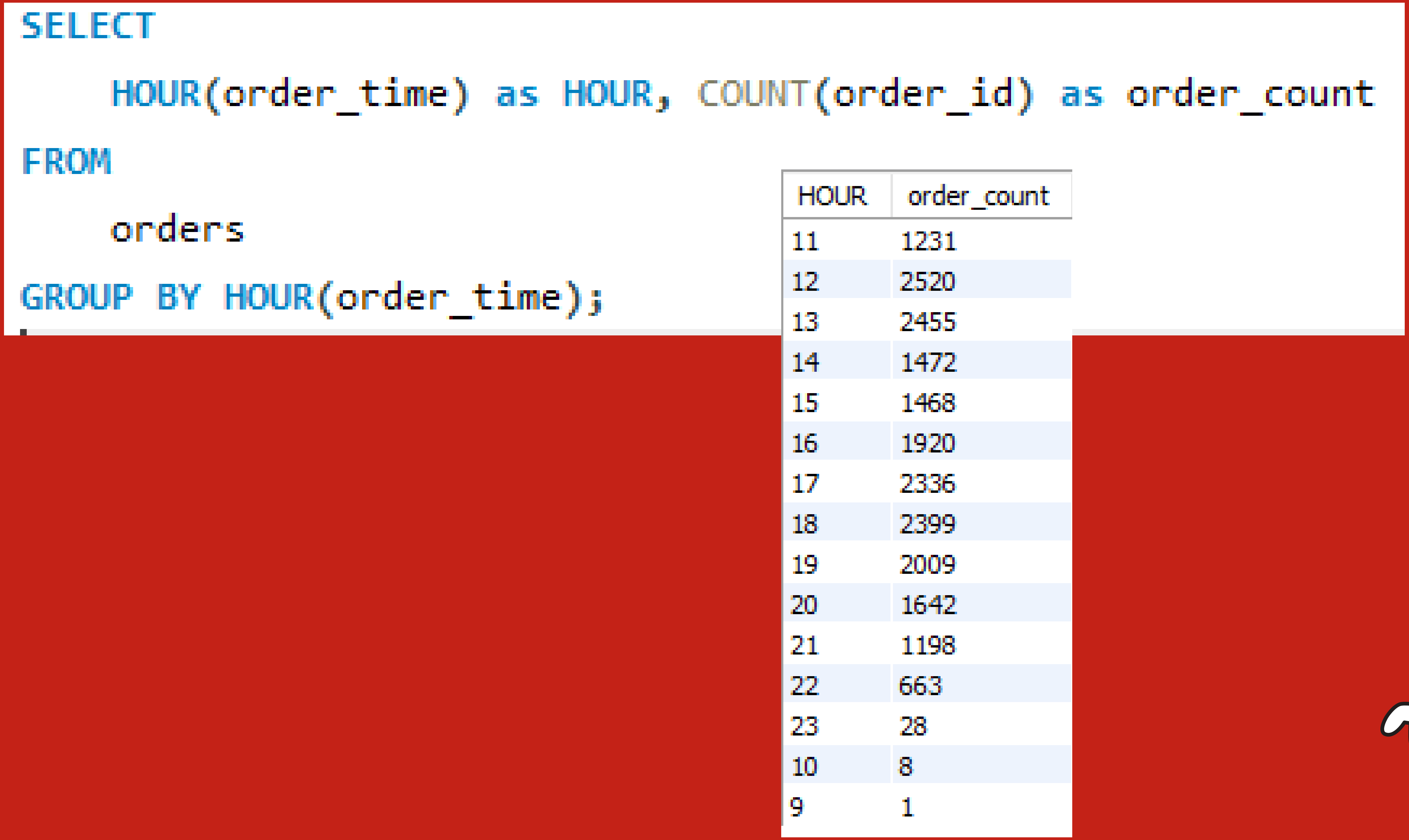
Join the necessary tables to find the total quantity of each pizza category ordered



```
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 quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

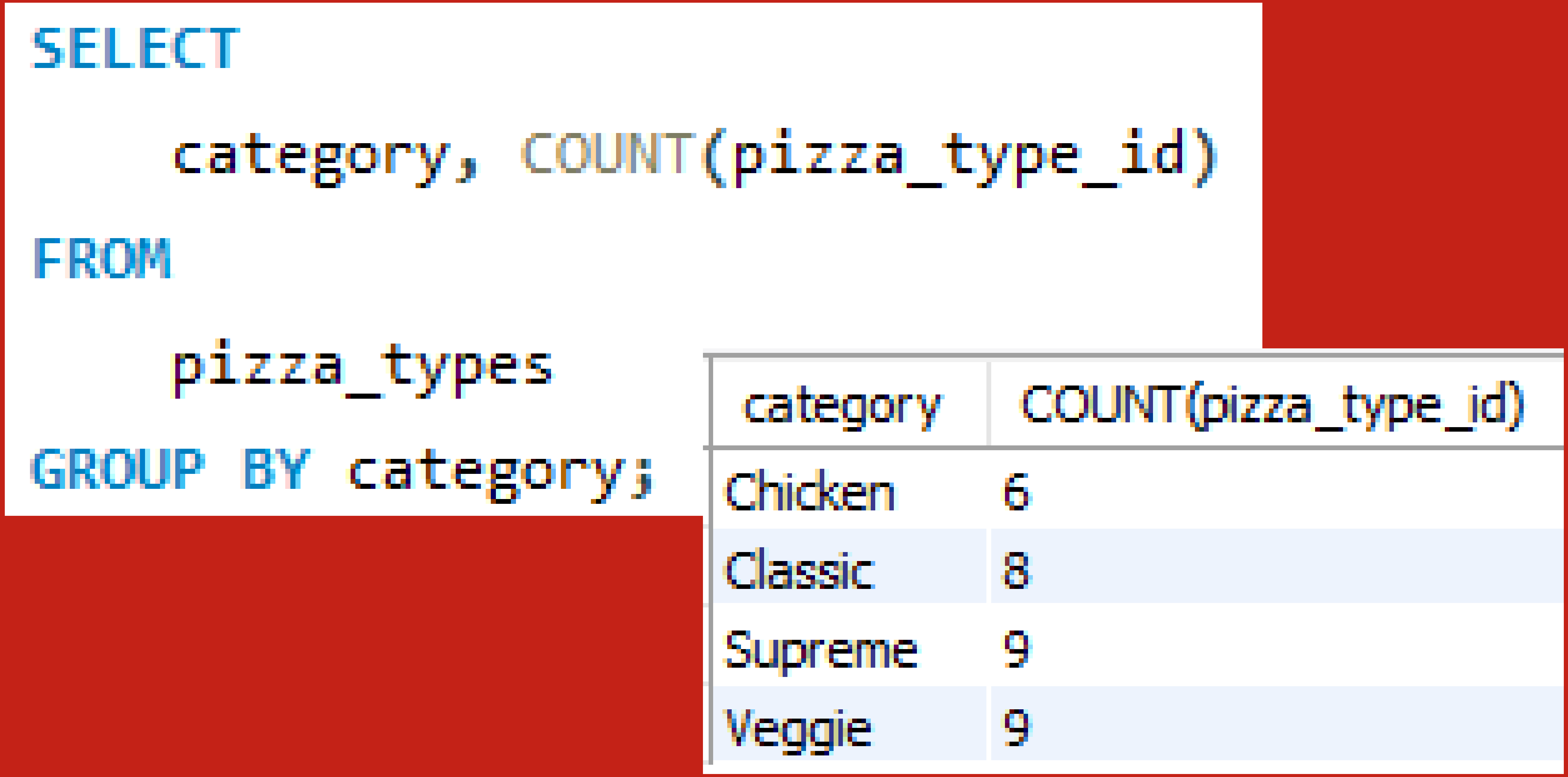
Determine the distribution of orders by hour of the day



```sql
SELECT
    HOUR(order_time) as HOUR, COUNT(order_id) as order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

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

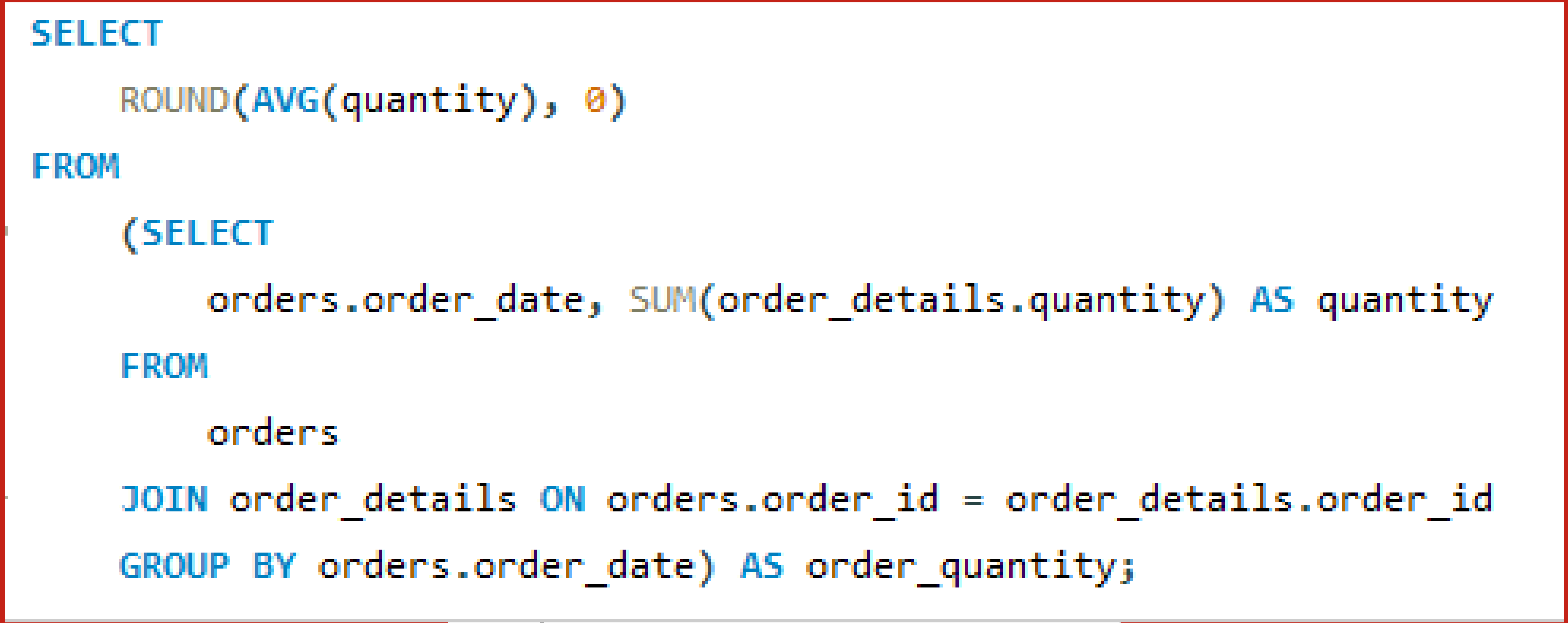
Join relevant tables to find the category-wise distribution of pizzas



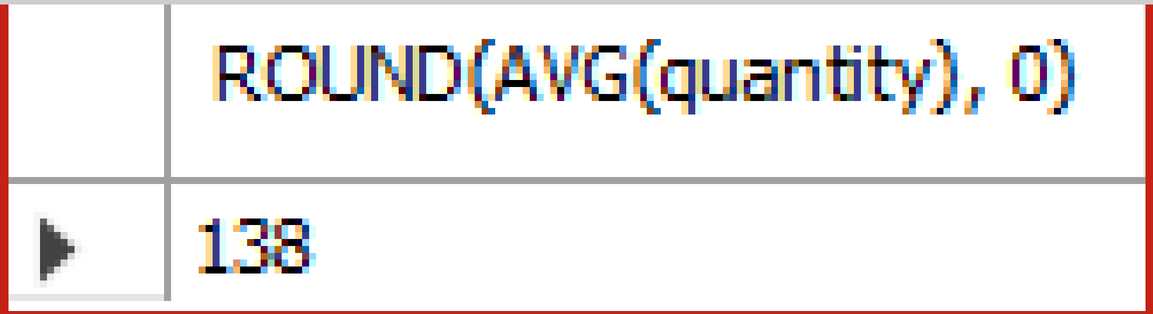
```
SELECT

    category, COUNT(pizza_type_id)
FROM

    pizza_types
GROUP BY category;
```

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

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

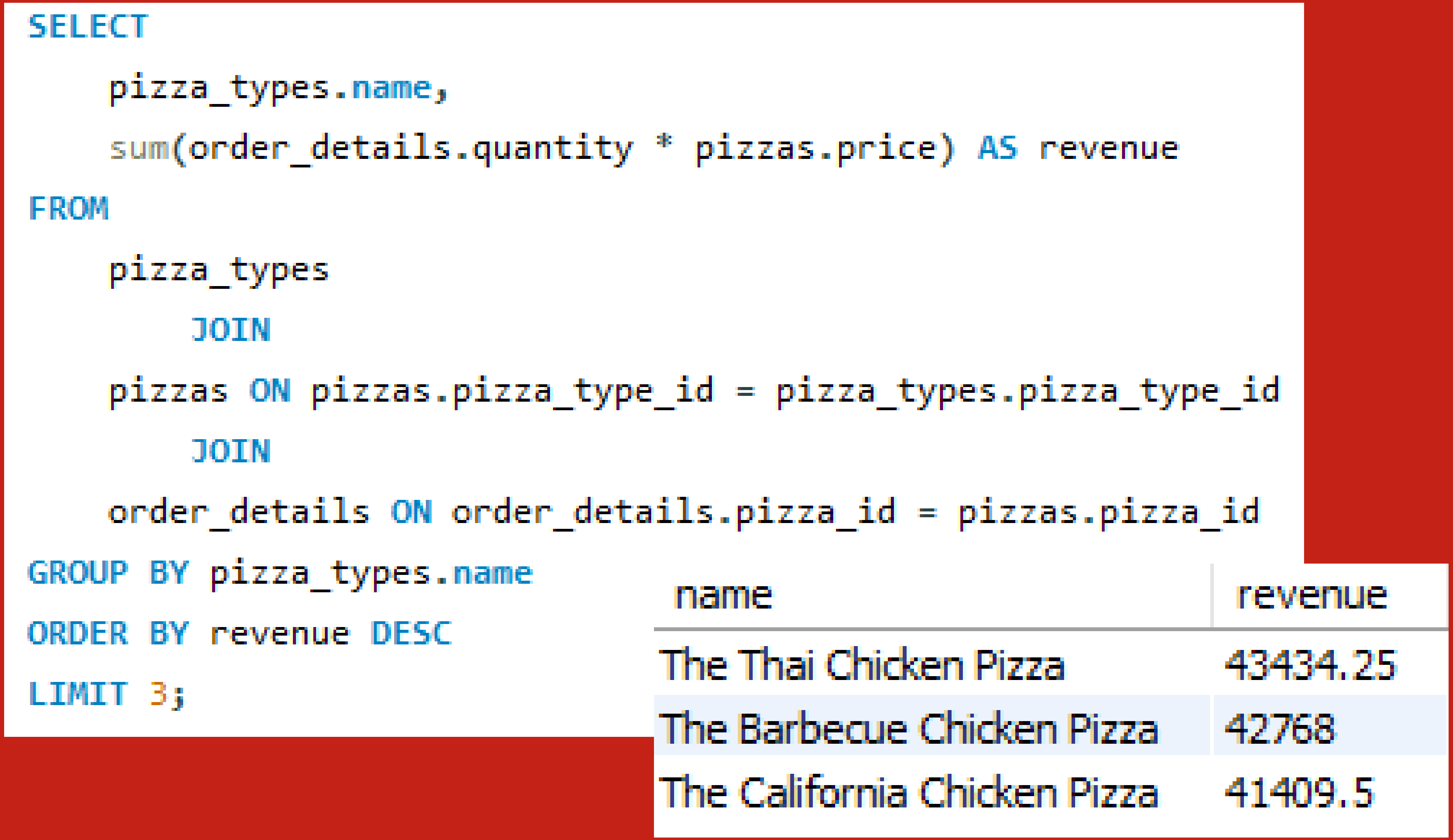


```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



	ROUND(AVG(quantity), 0)
▶	138

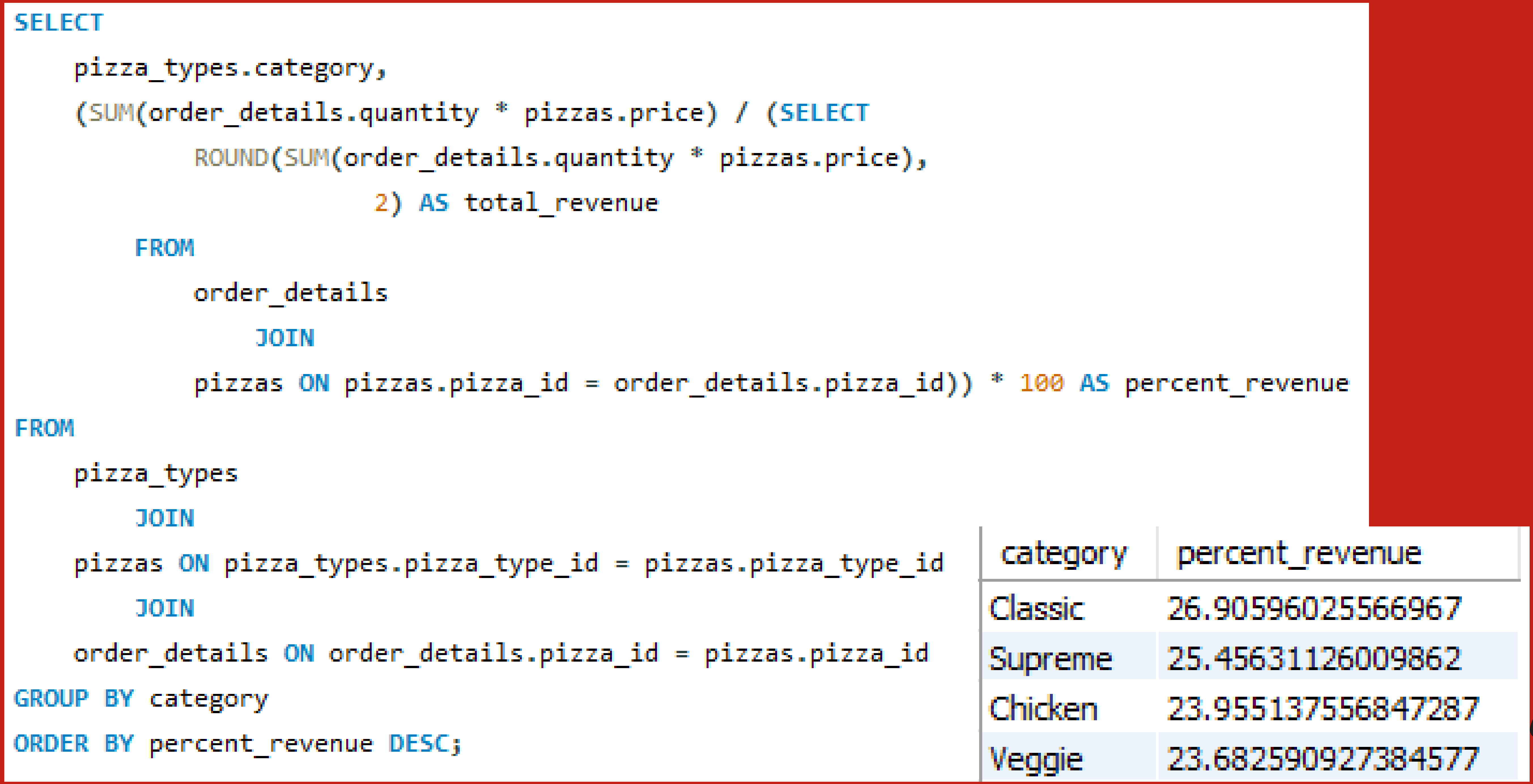
Determine the top 3 most ordered pizza types based on revenue



```
SELECT
    pizza_types.name,
    sum(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON 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 DESC
LIMIT 3;
```

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

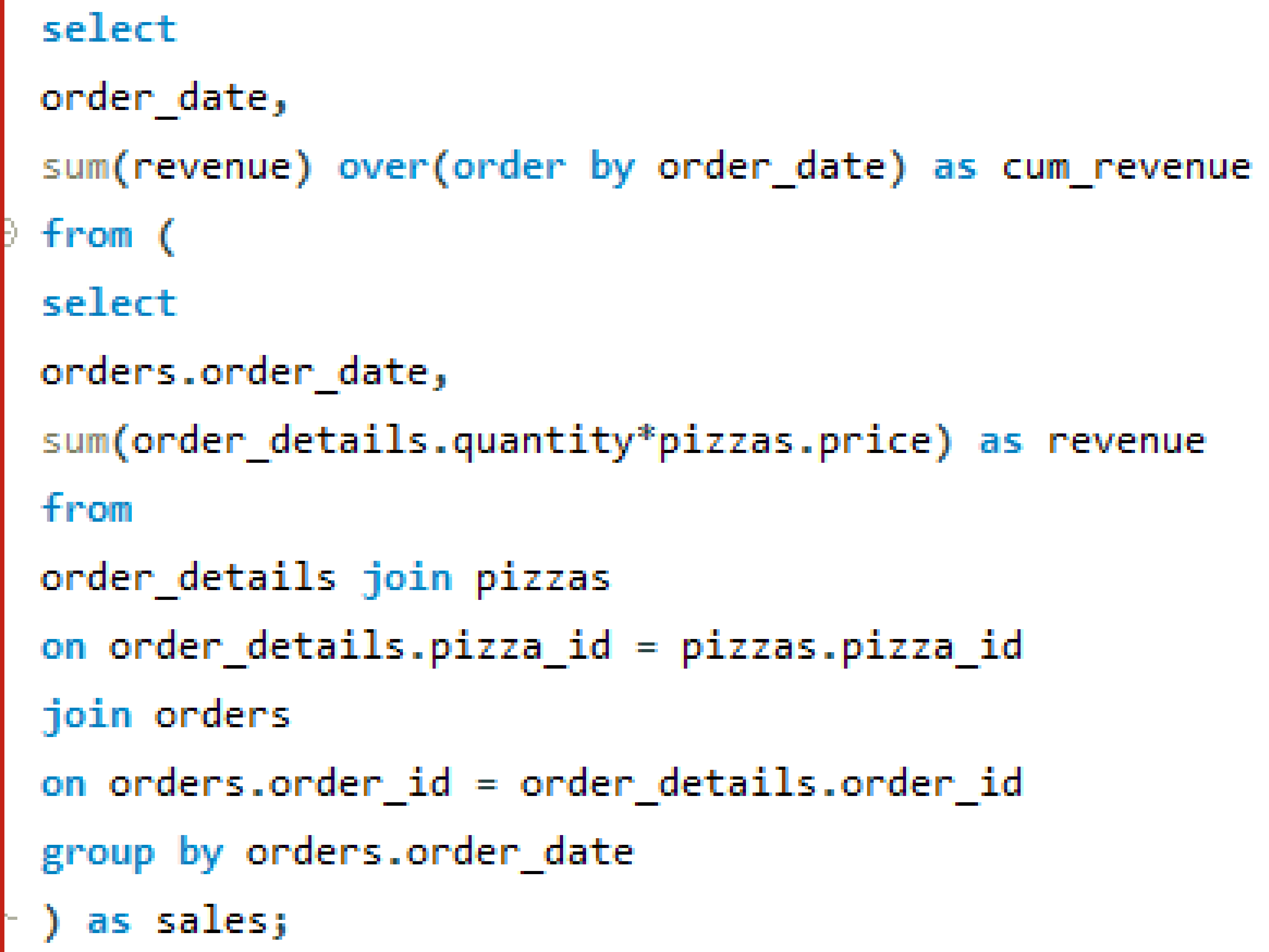
Calculate the percentage contribution of each pizza type to total revenue



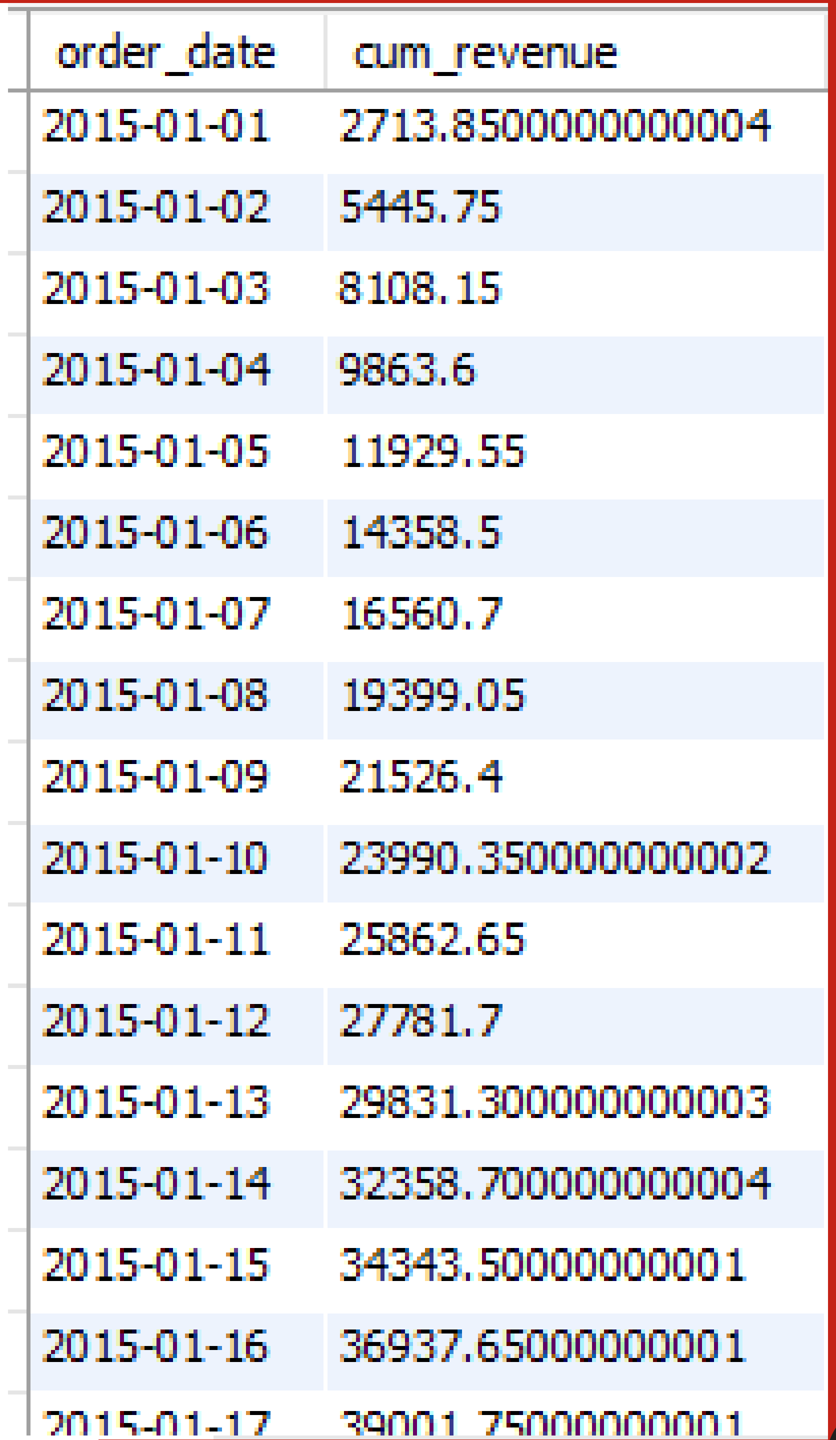
```sql
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_revenue
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 AS percent_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 category
ORDER BY percent_revenue DESC;
```

category	percent_revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

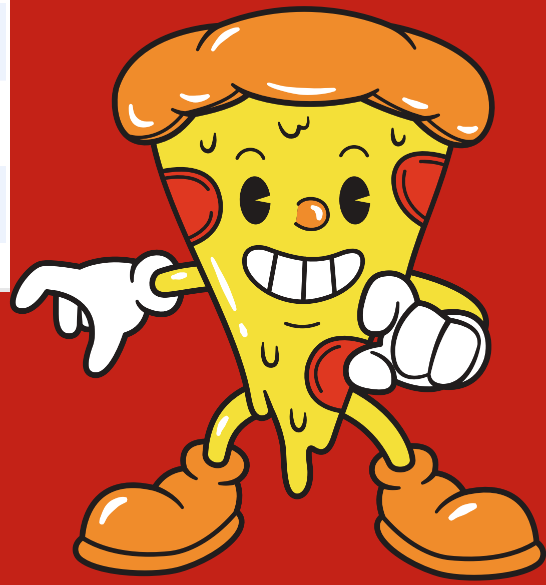
Analyze the cumulative revenue generated over time



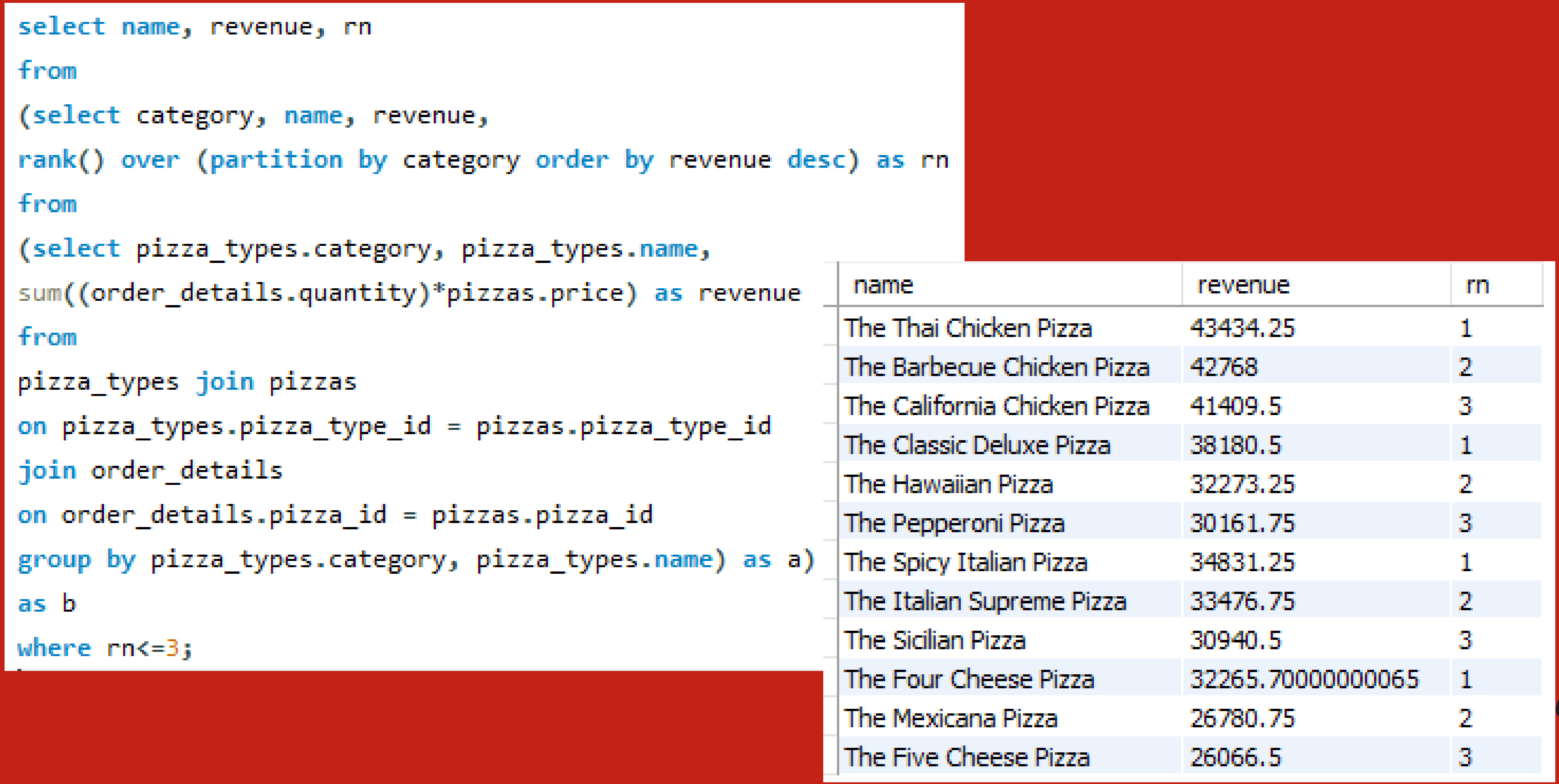
```
select
order_date,
sum(revenue) over(order by order_date) as cum_revenue
from (
select
orders.order_date,
sum(order_details.quantity*pizzas.price) as revenue
from
order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by 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
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001



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



```
select 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;
```

name	revenue	rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3