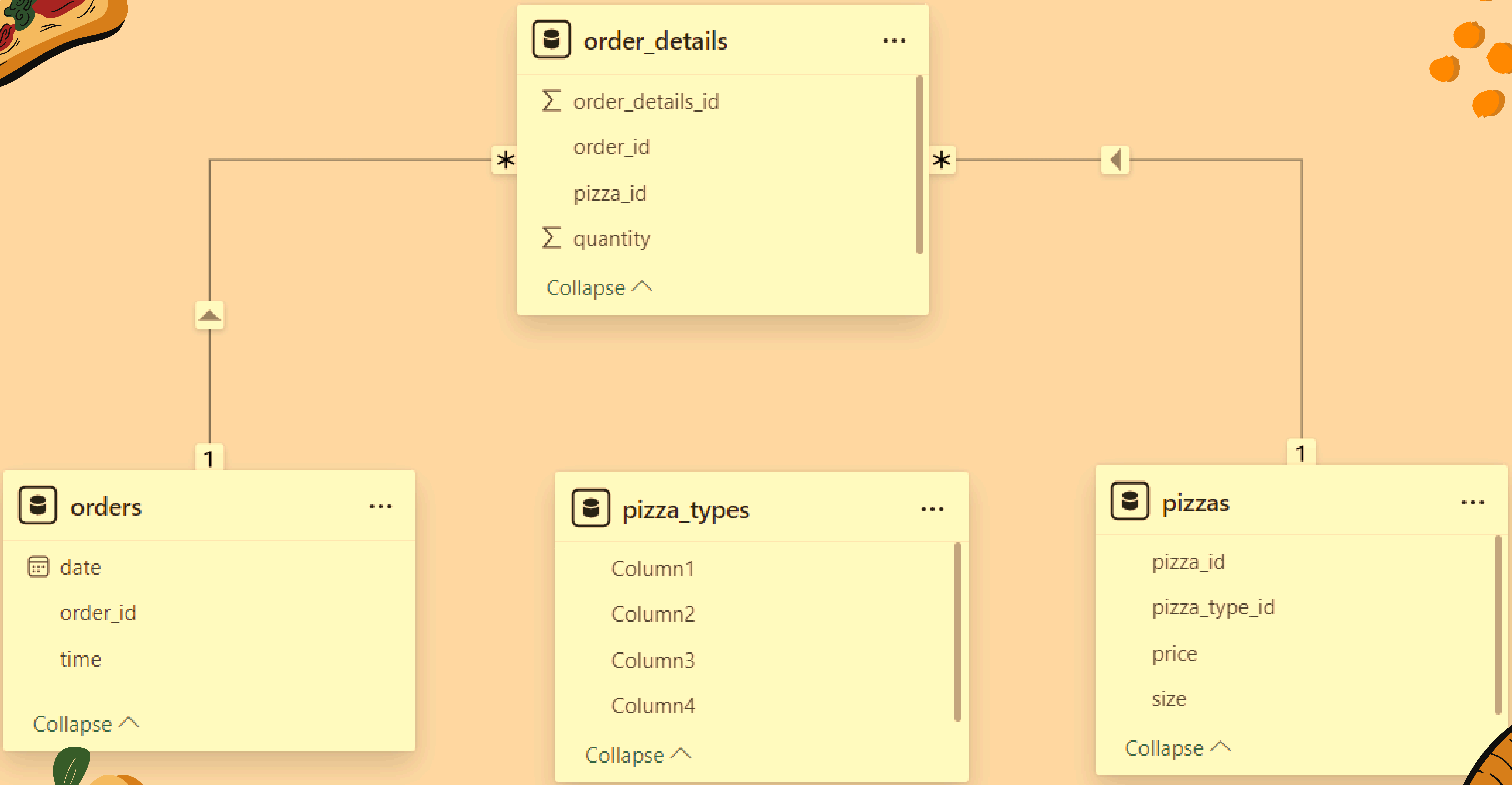
PIZZA SALES SQL PROJECT :

ANALYZING SALES DATA WITH SQL QUERIES

Hello!

My name is Maulik kakkar. This project focuses on analyzing pizza sales data using SQL. The goal is to extract valuable insights from a hypothetical pizza restaurant's database by writing and executing SQL queries. Through this project, I aim to showcase how SQL can be used to manage, retrieve, and analyze data to provide a deeper understanding of business performance.



order_details

- order_details_id
- order_id
- pizza_id
- quantity

orders

- date
- order_id
- time

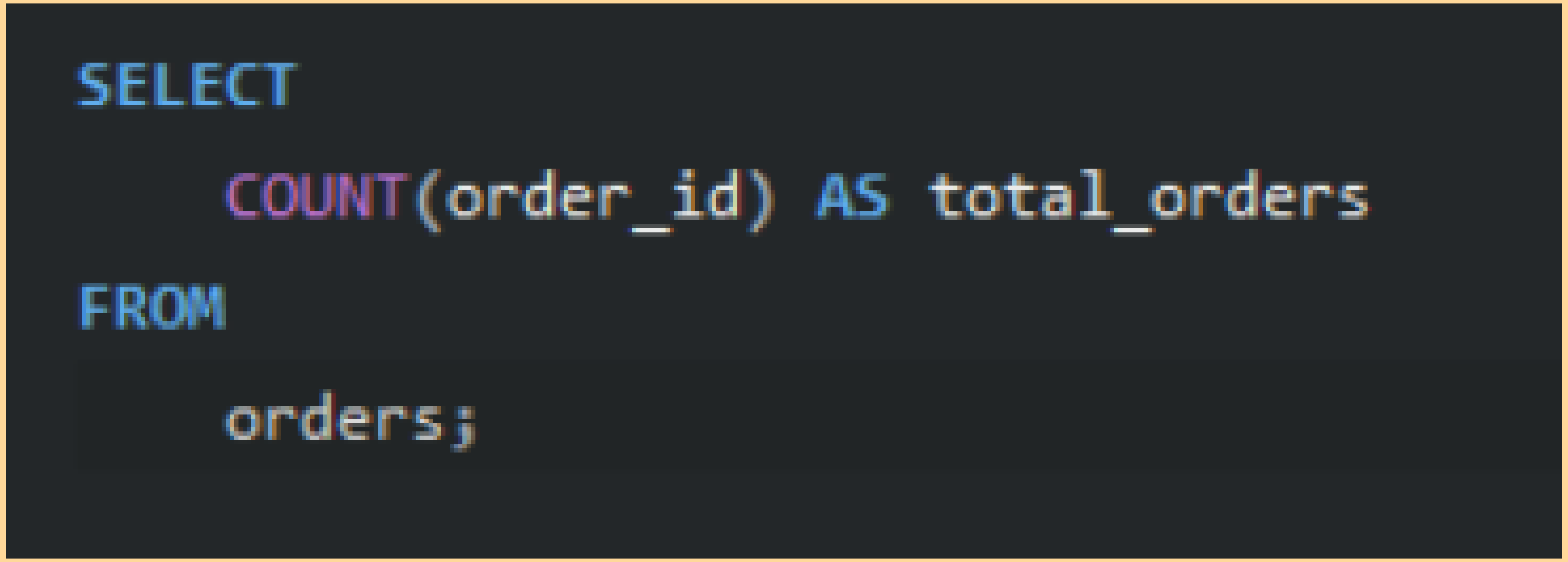
pizza_types

- Column1
- Column2
- Column3
- Column4

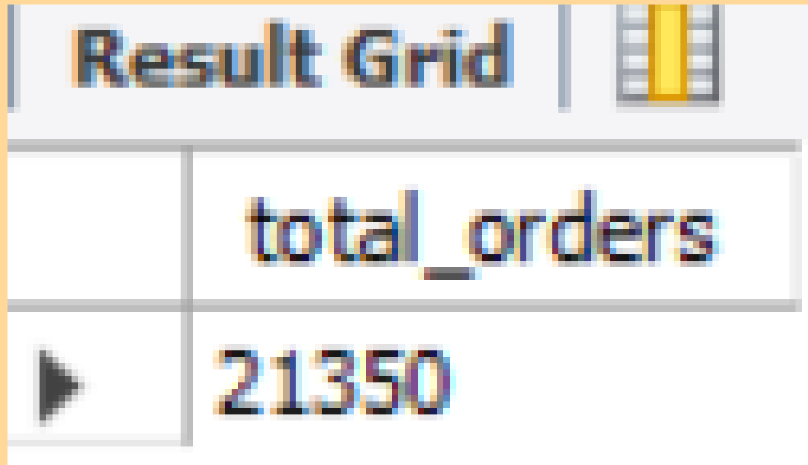
pizzas

- pizza_id
- pizza_type_id
- price
- size

Retrieve the total number of orders placed.

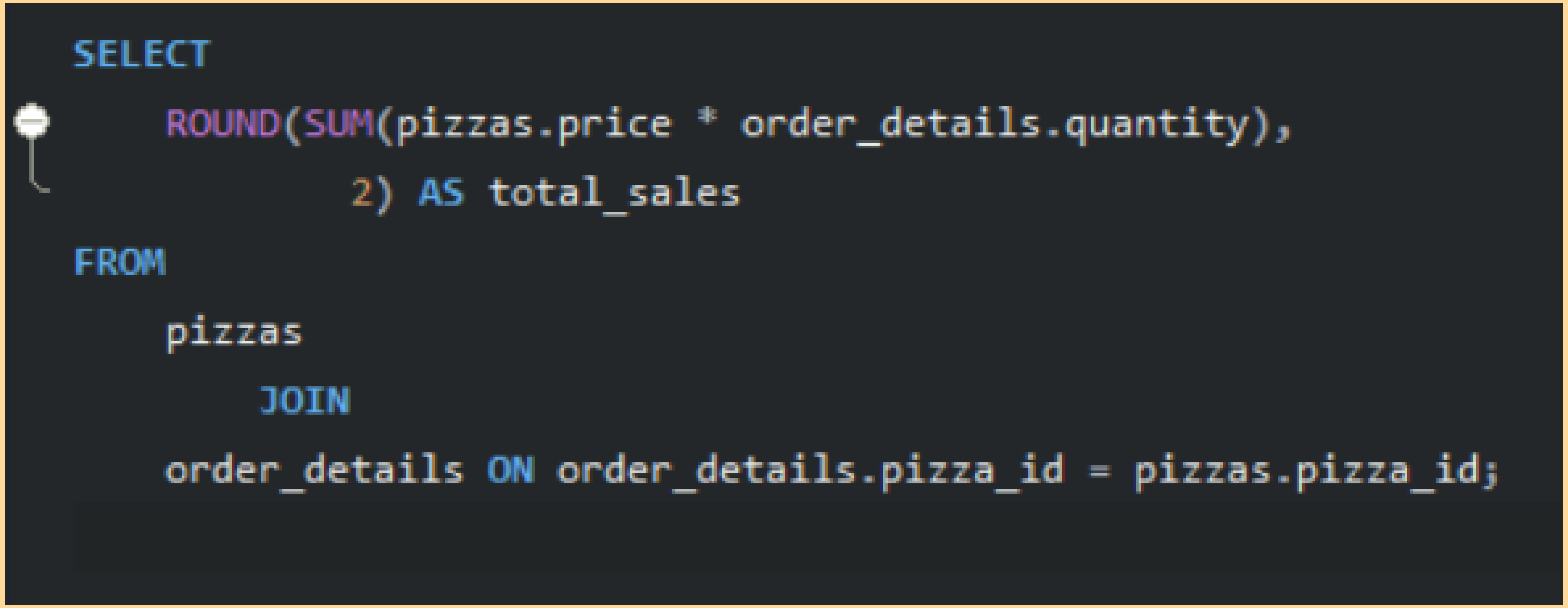


```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

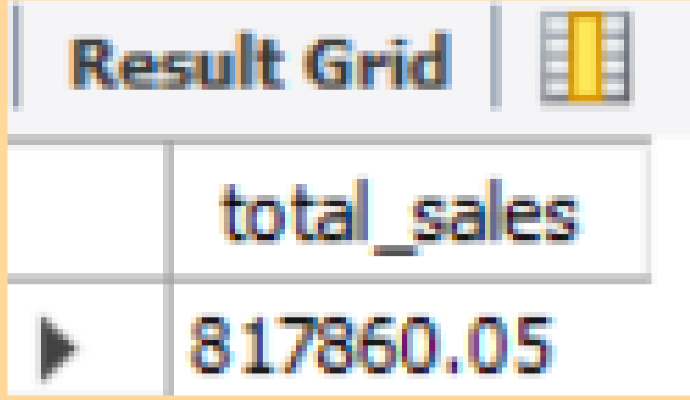


total_orders
21350

Calculate the total revenue generated from pizza sales.

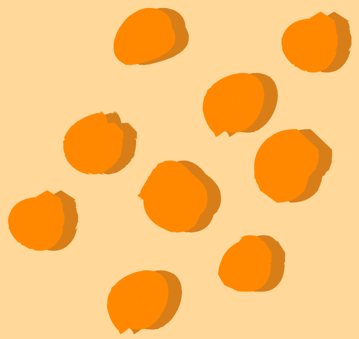


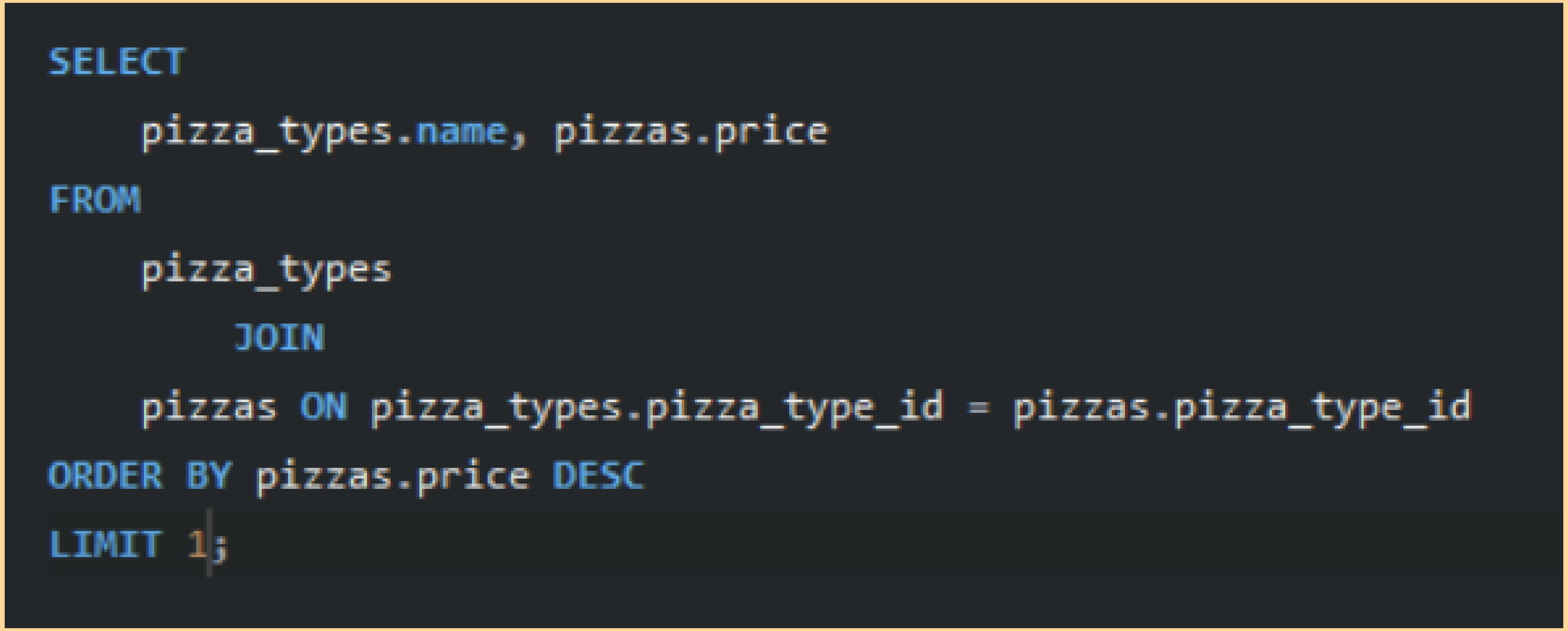
```
SELECT
    ROUND(SUM(pizzas.price * order_details.quantity),
            2) AS total_sales
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id;
```



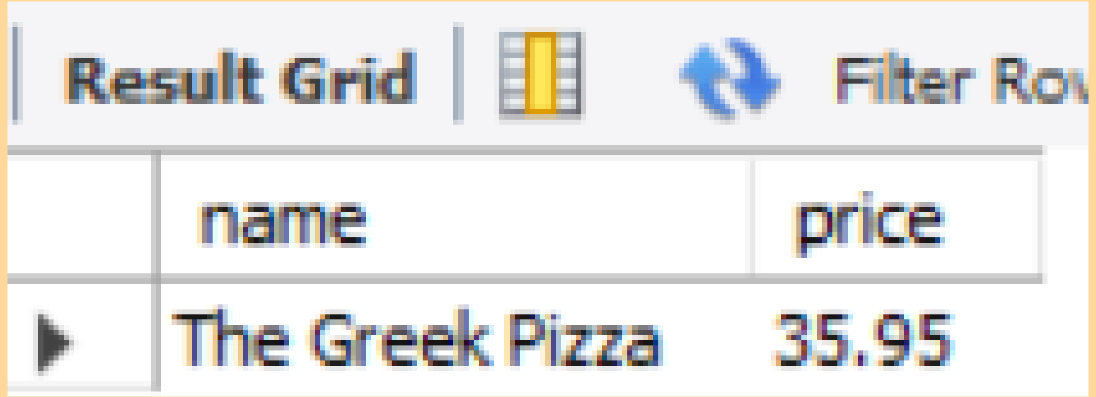
```
total_sales
817860.05
```

Identify the highest-priced pizza.



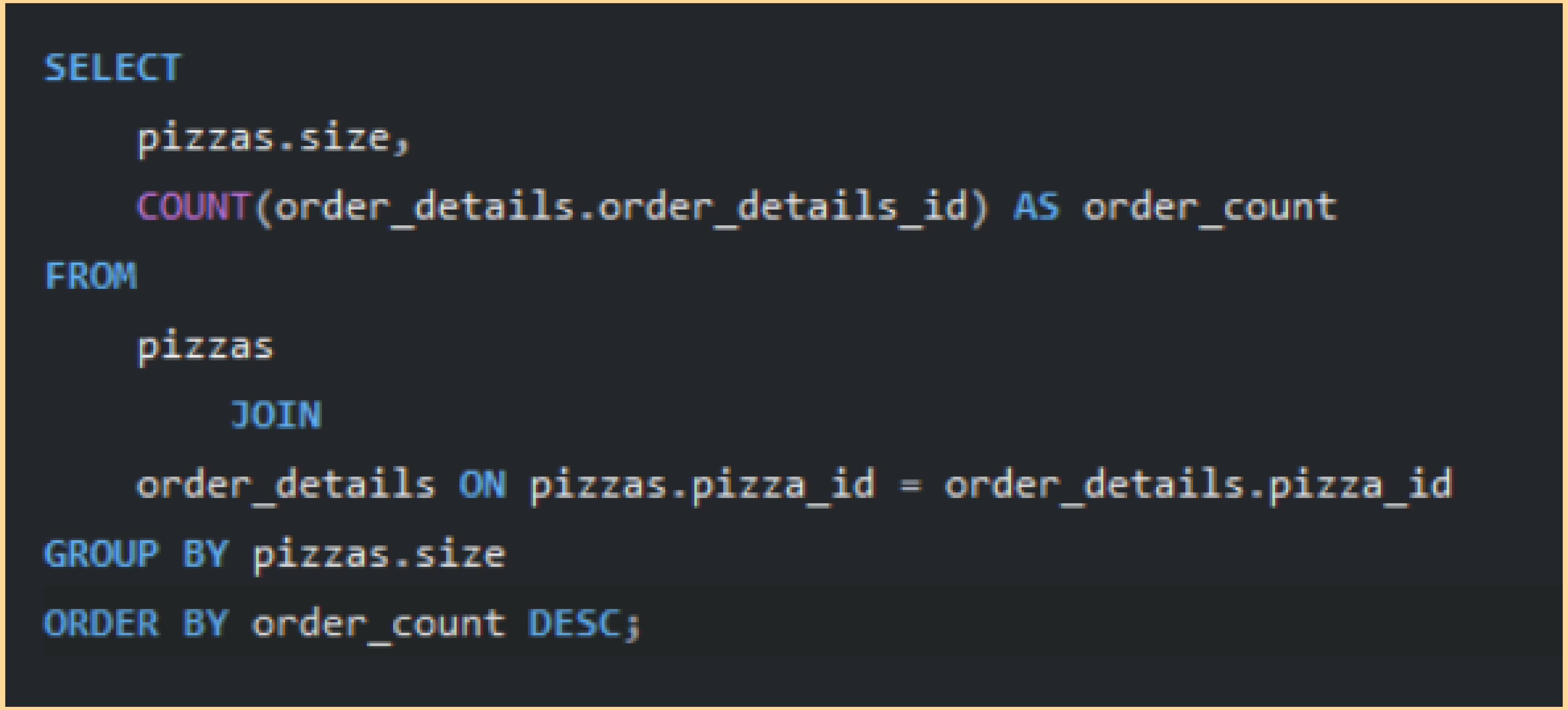


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

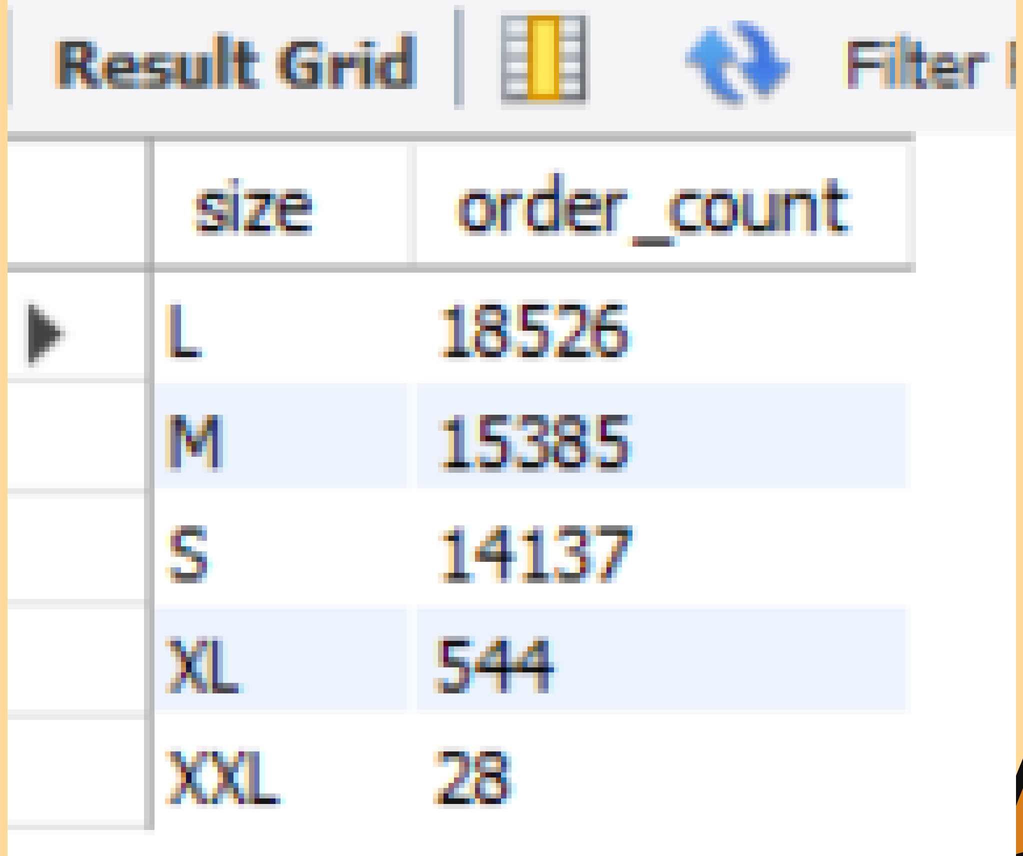


name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

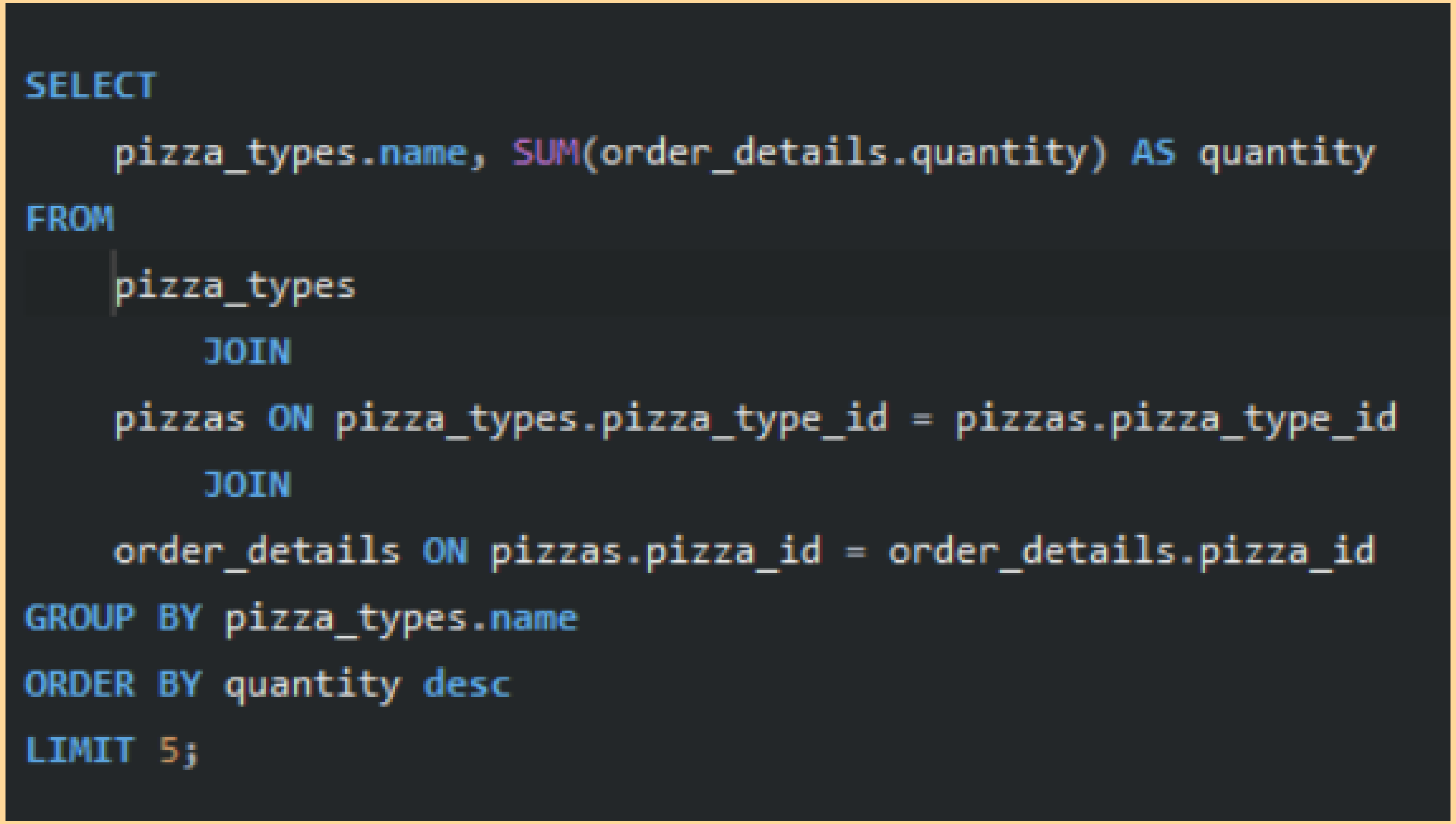


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

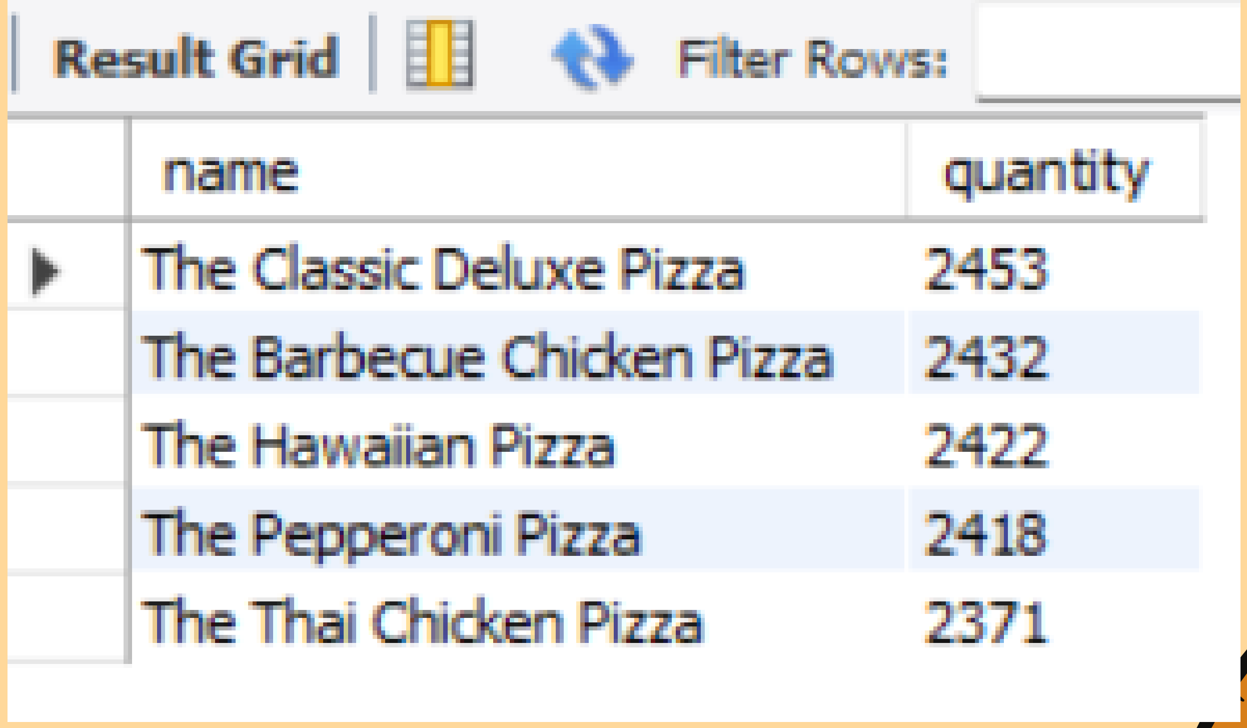


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

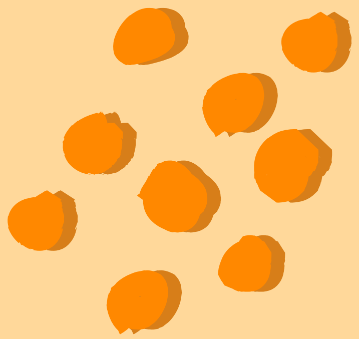


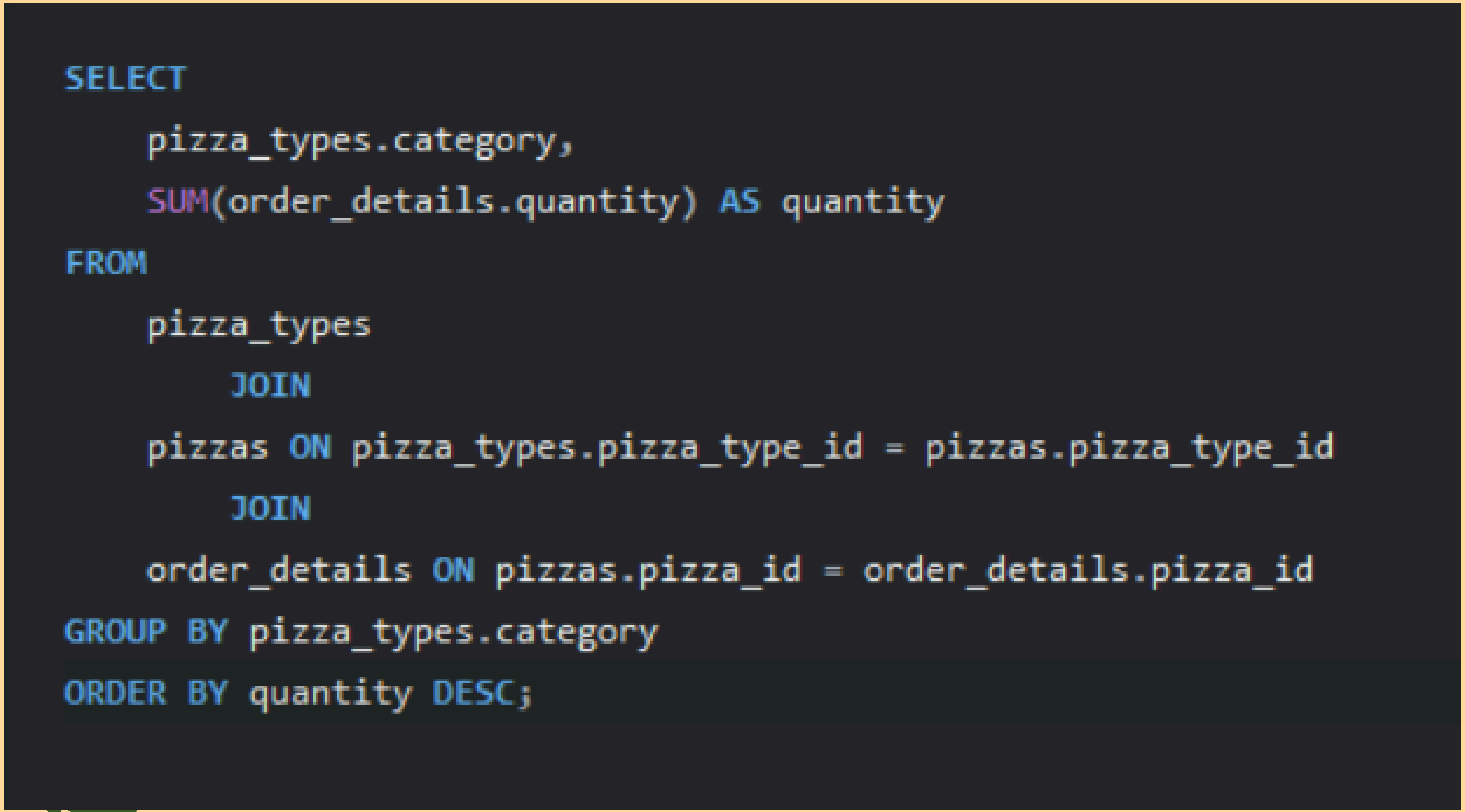
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity desc
LIMIT 5;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



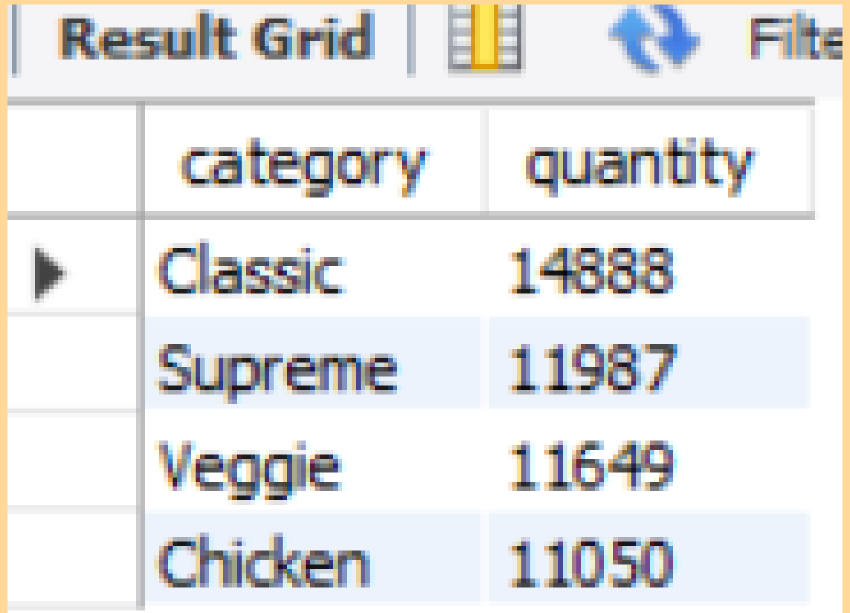
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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



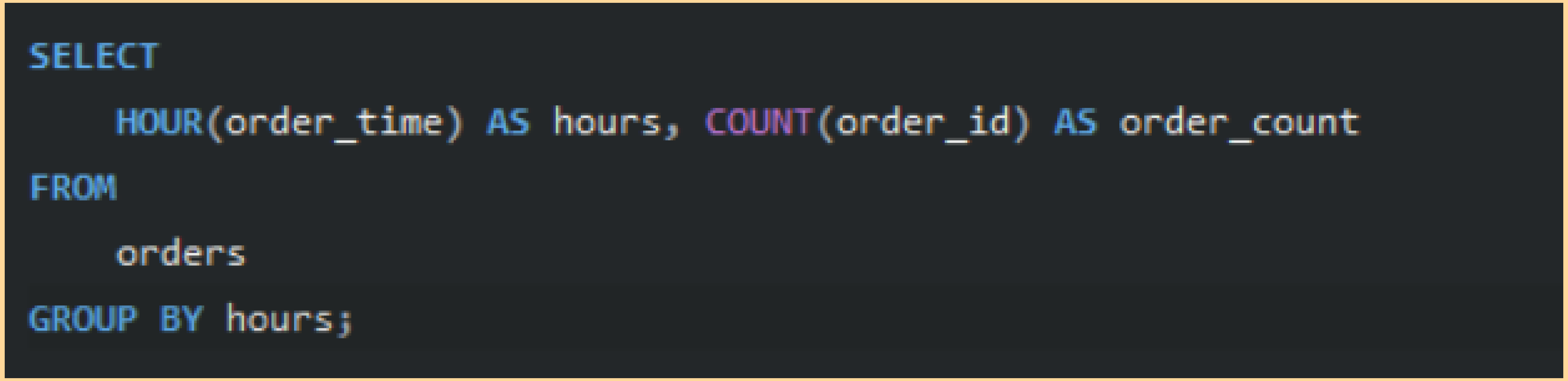


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

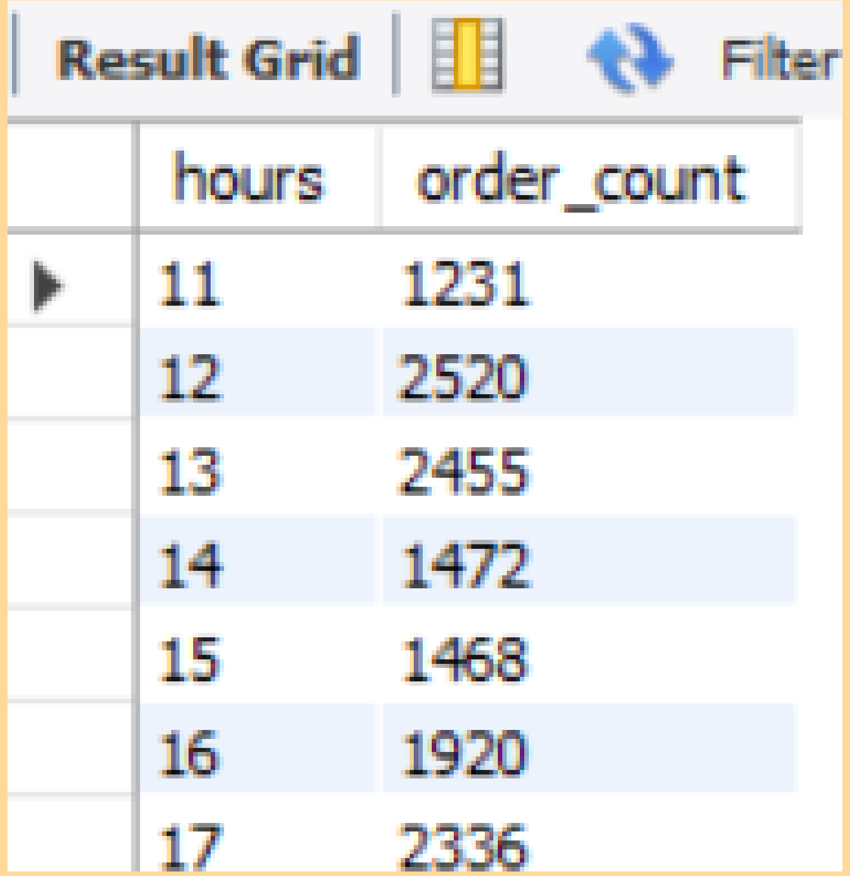


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

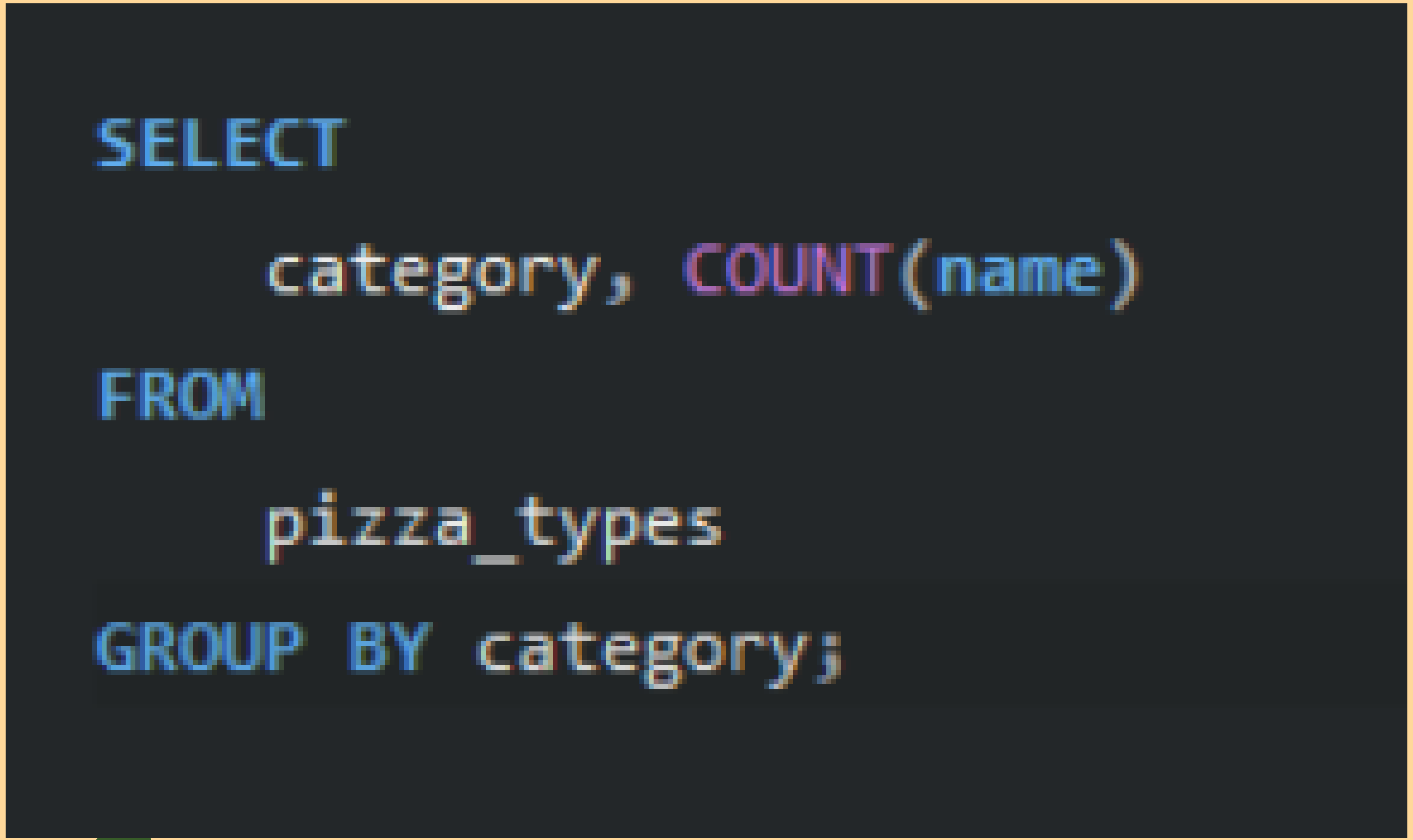


```
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hours;
```

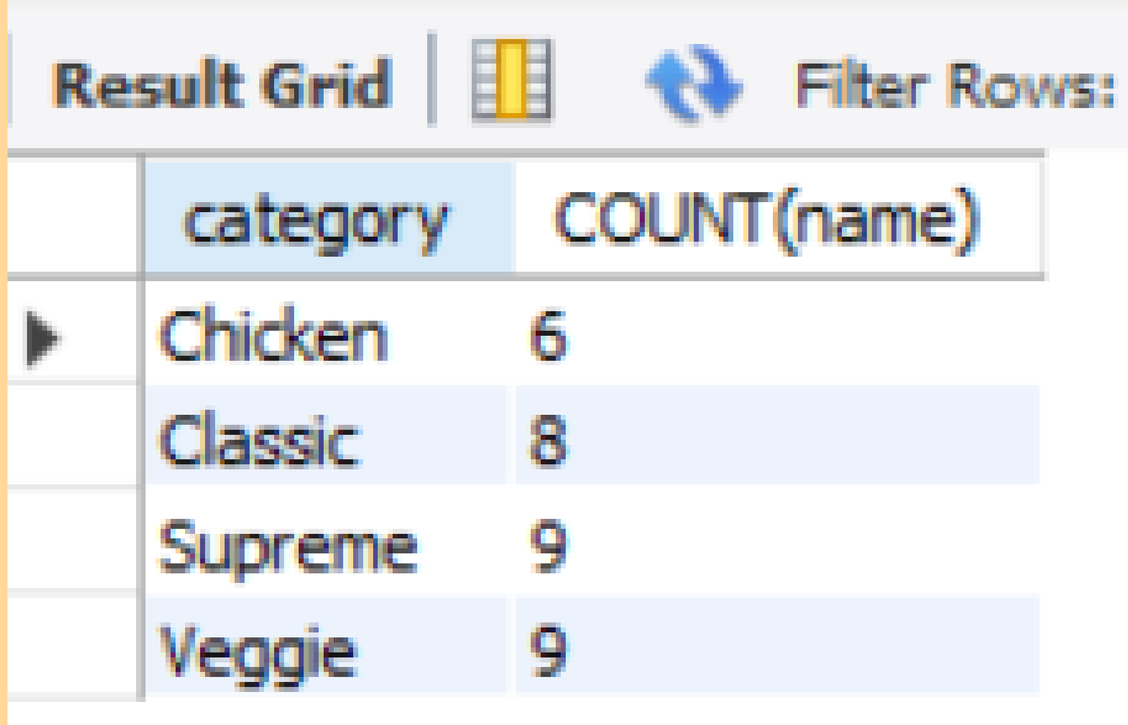


hours	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

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

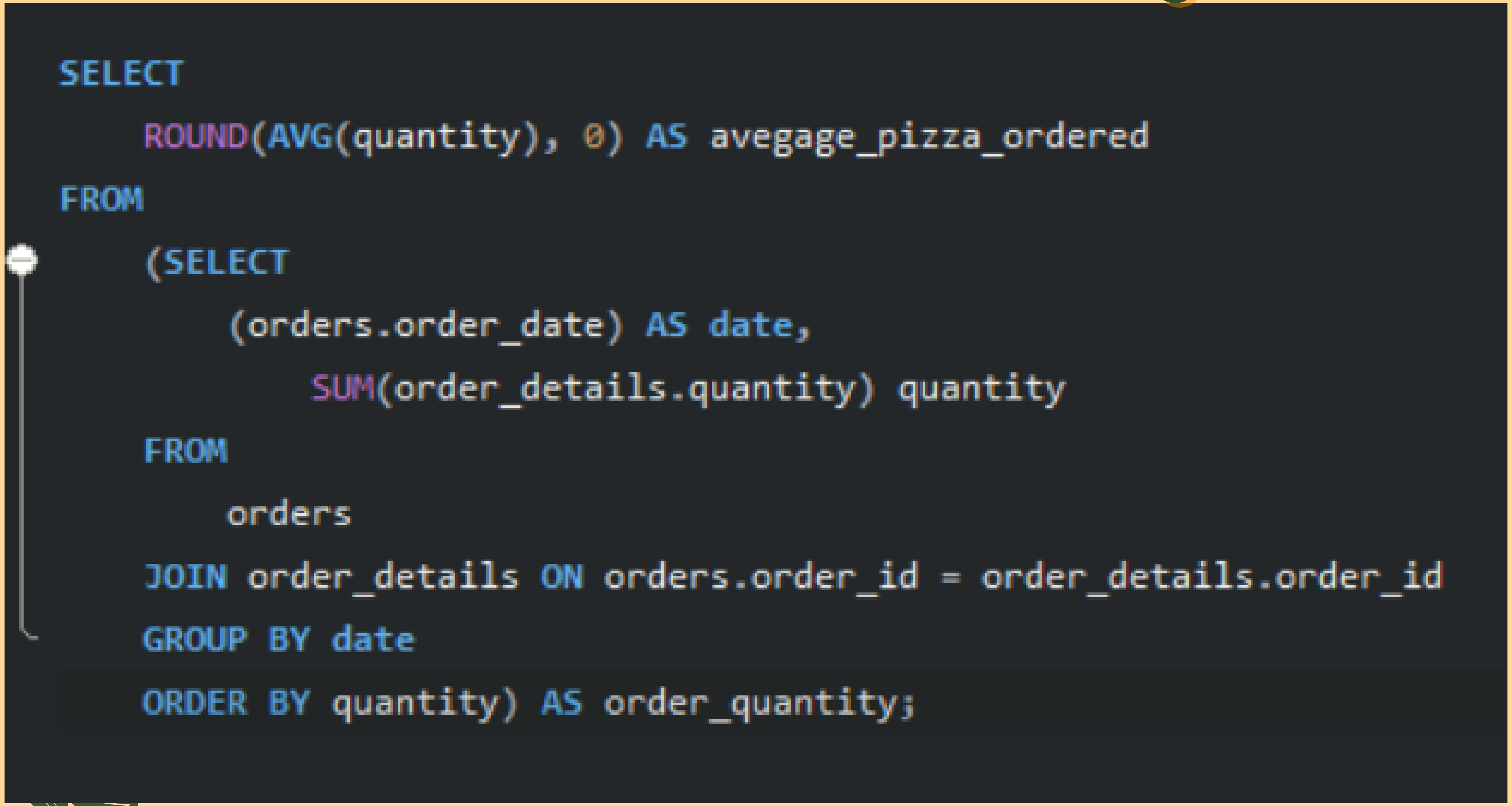


```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

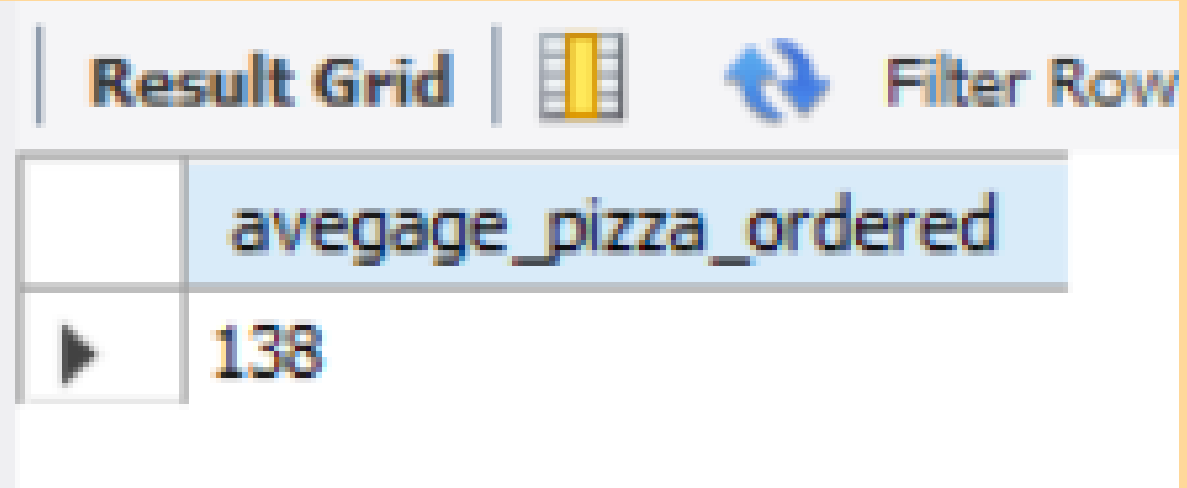


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

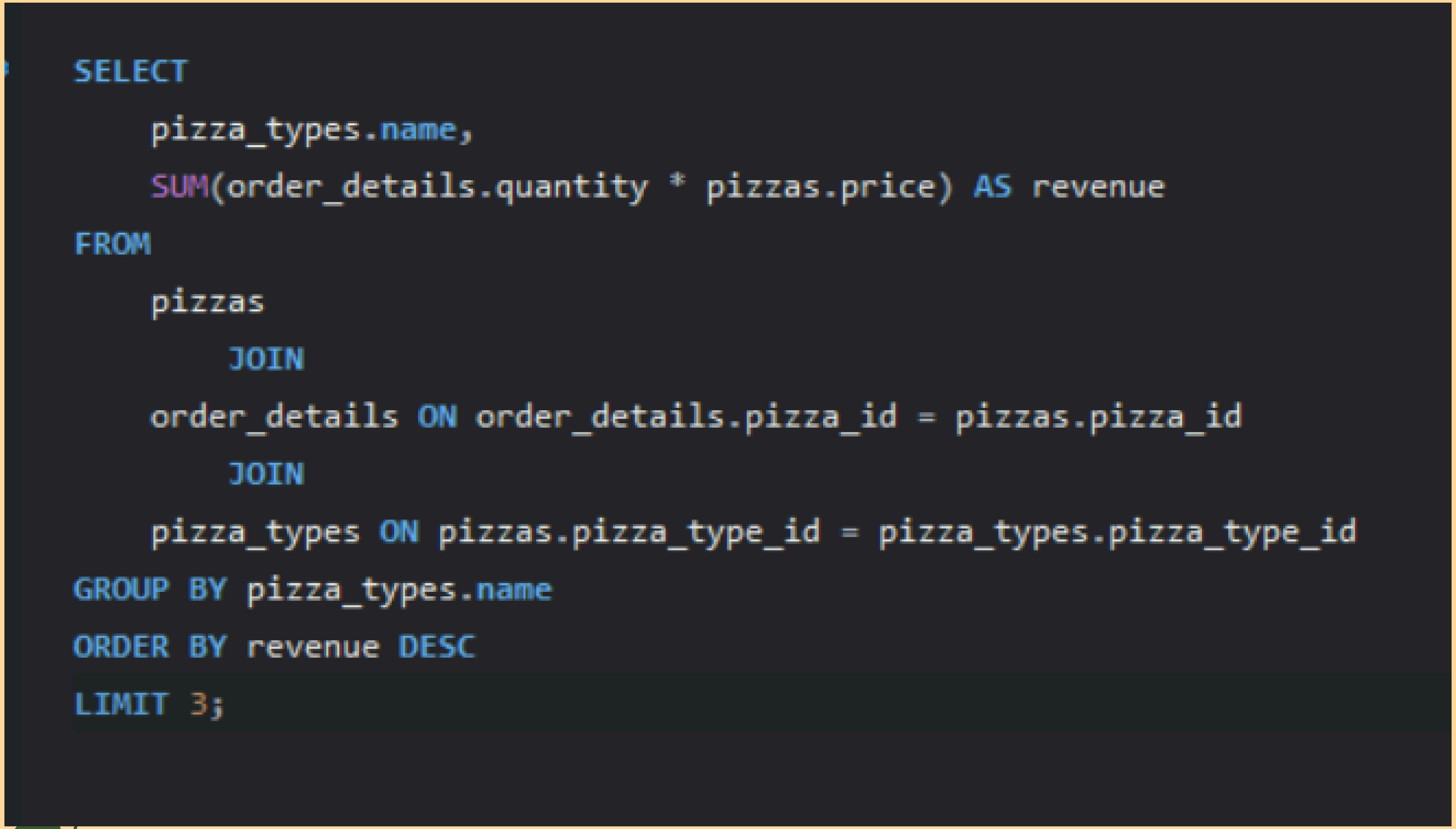


```
SELECT
    ROUND(AVG(quantity), 0) AS avegage_pizza_ordered
FROM
    (SELECT
        (orders.order_date) AS date,
            SUM(order_details.quantity) quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date
    ORDER BY quantity) AS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

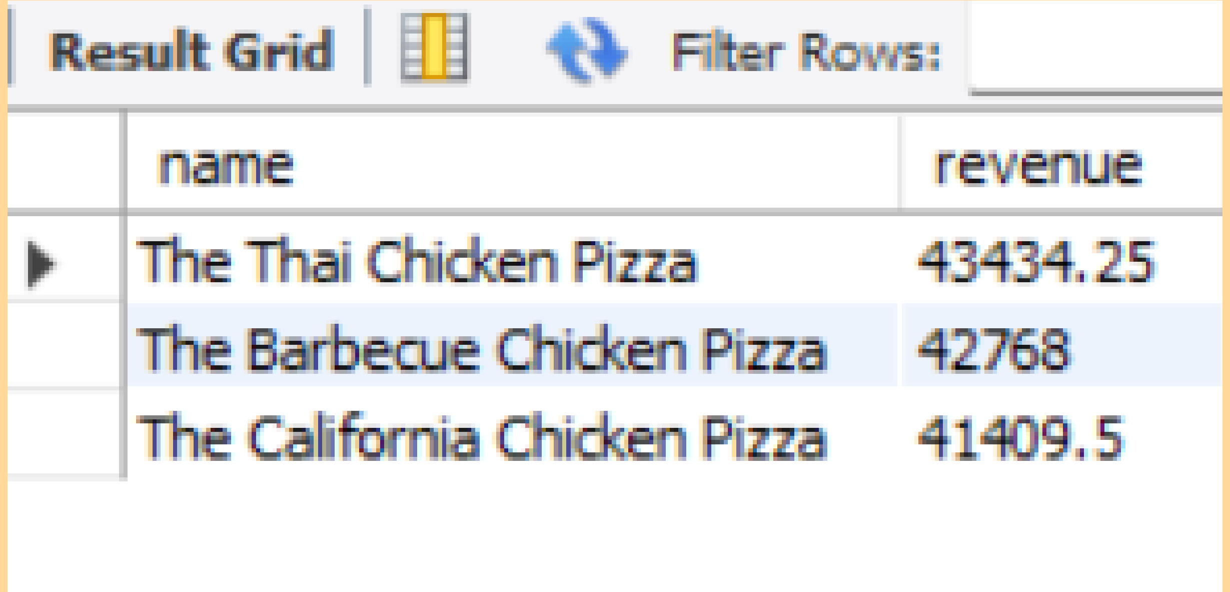


	avegage_pizza_ordered
▶	138

Determine the top 3 most ordered pizza types based on revenue

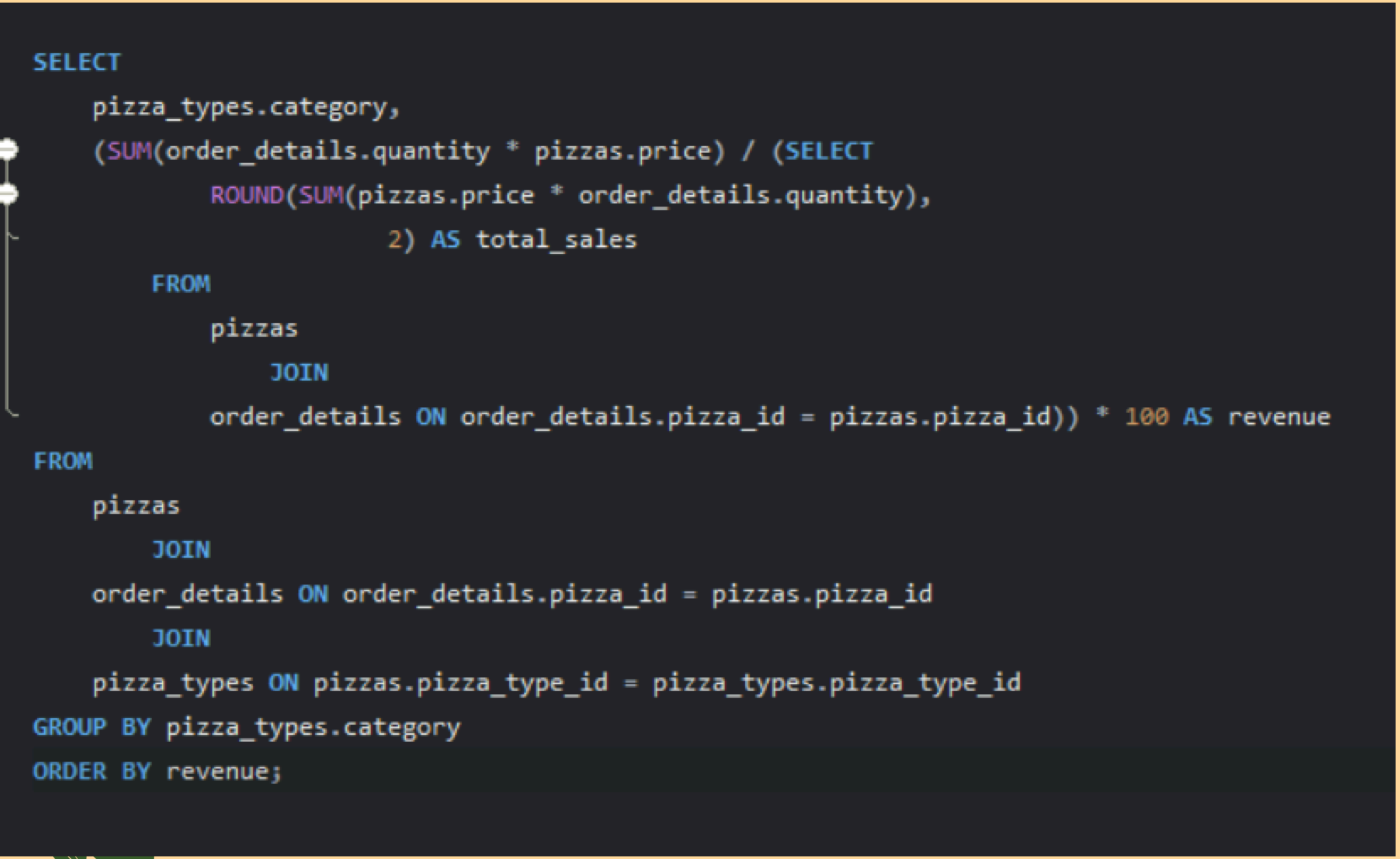


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue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

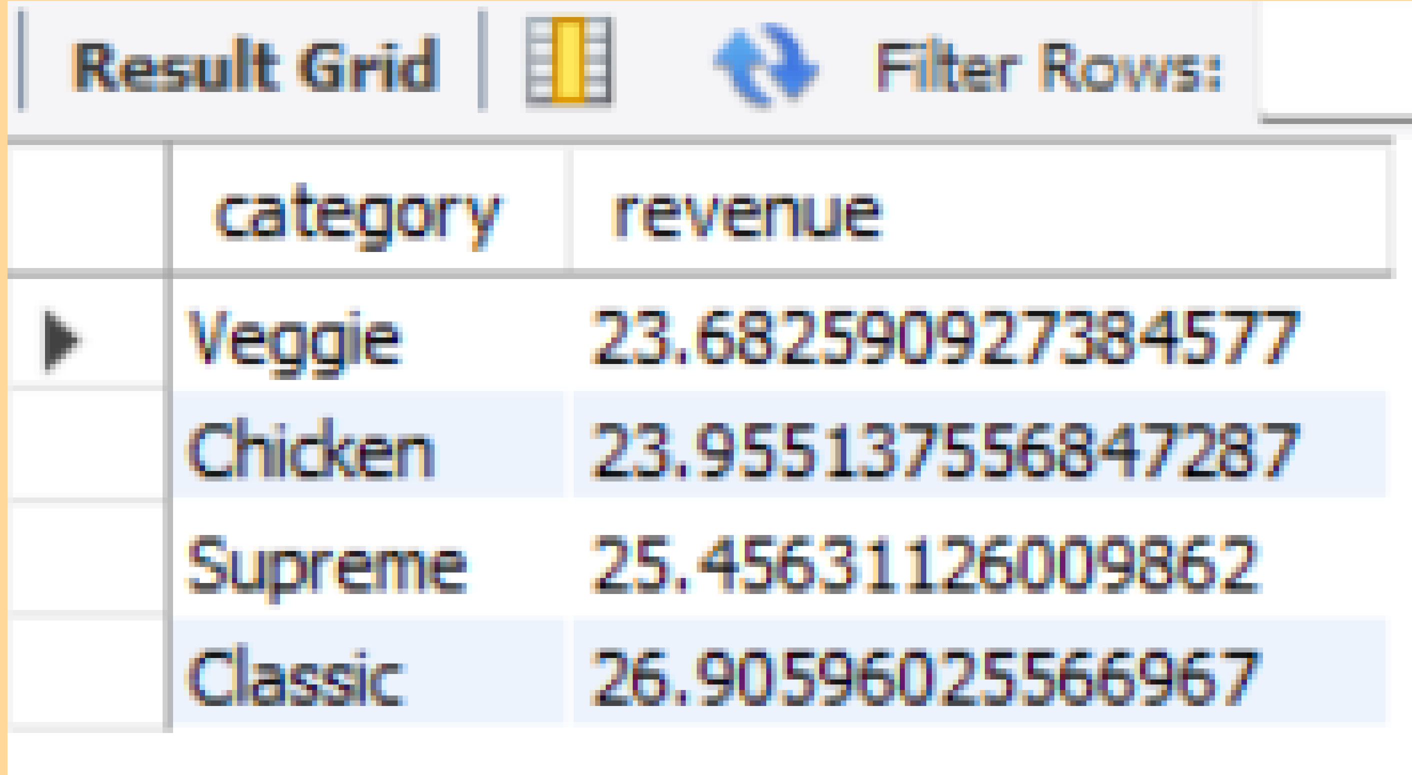


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

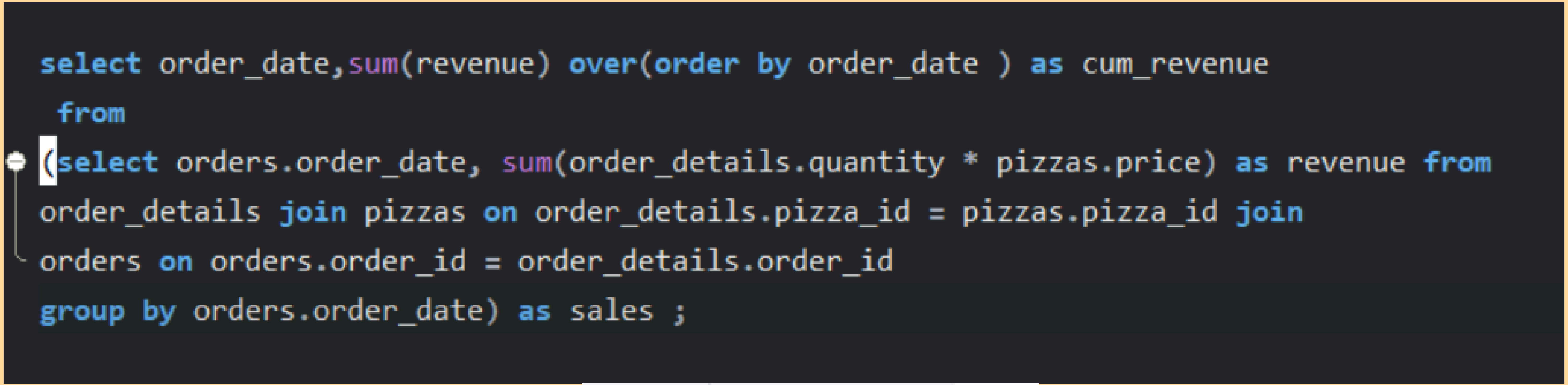


```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(pizzas.price * order_details.quantity),
                        2) AS total_sales
        FROM
            pizzas
                JOIN
            order_details ON order_details.pizza_id = pizzas.pizza_id)) * 100 AS revenue
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY revenue;
```

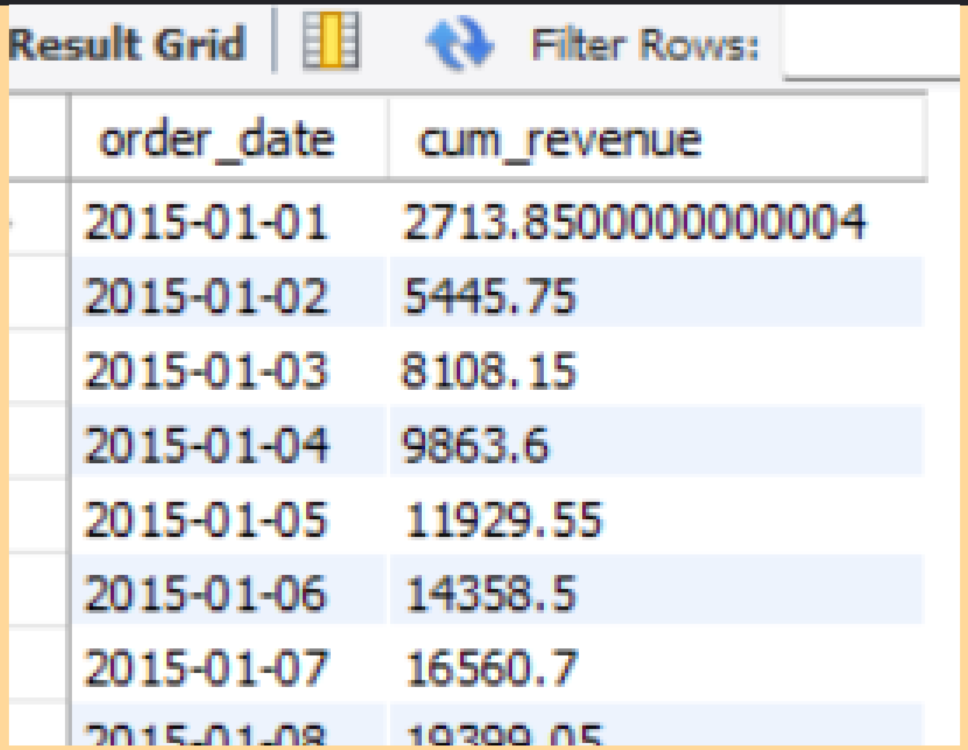


category	revenue
Veggie	23.682590927384577
Chicken	23.955137556847287
Supreme	25.45631126009862
Classic	26.90596025566967

Analyze the cumulative revenue generated over time.

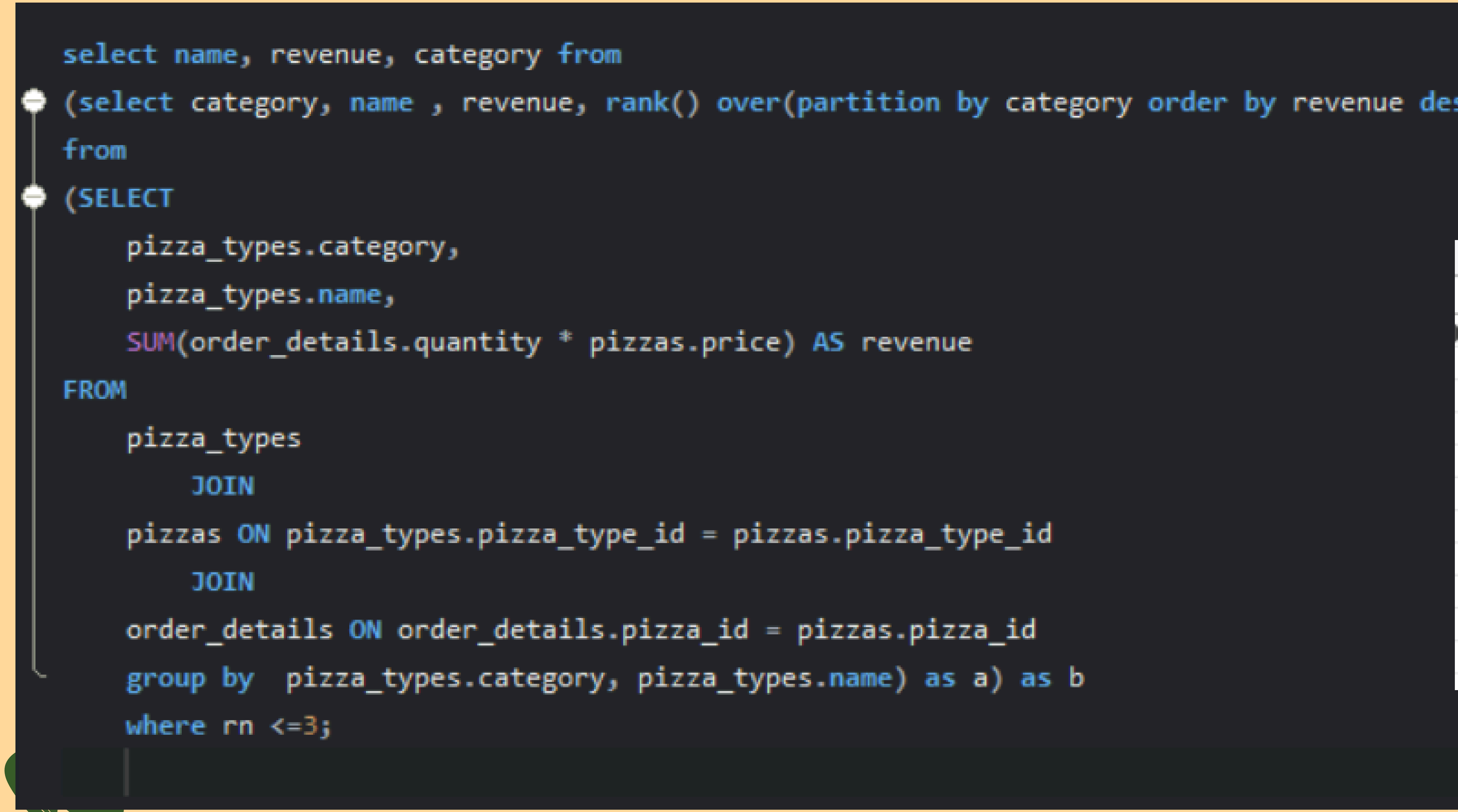


```
select order_date,sum(revenue) over(order by order_date ) as cum_revenue
 from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue from
order_details join pizzas on order_details.pizza_id = pizzas.pizza_id join
orders on orders.order_id = order_details.order_id
group by orders.order_date) as sales ;
```

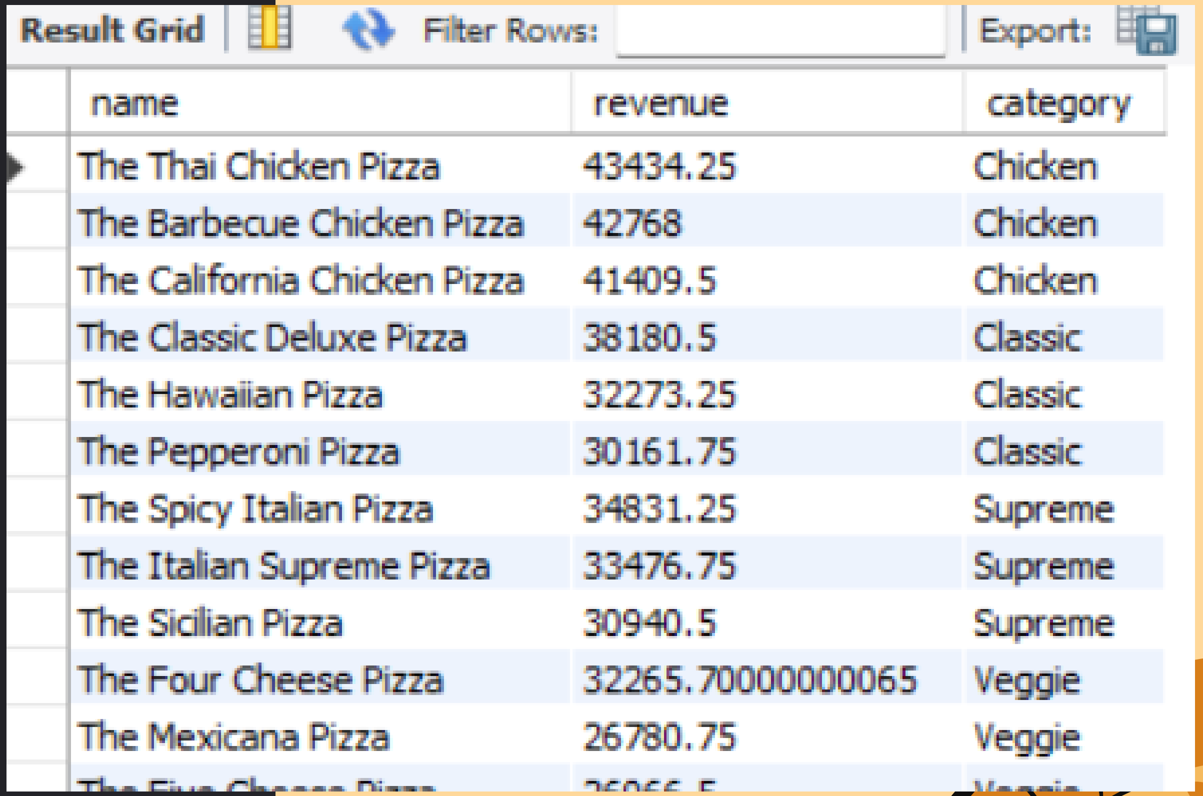


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05

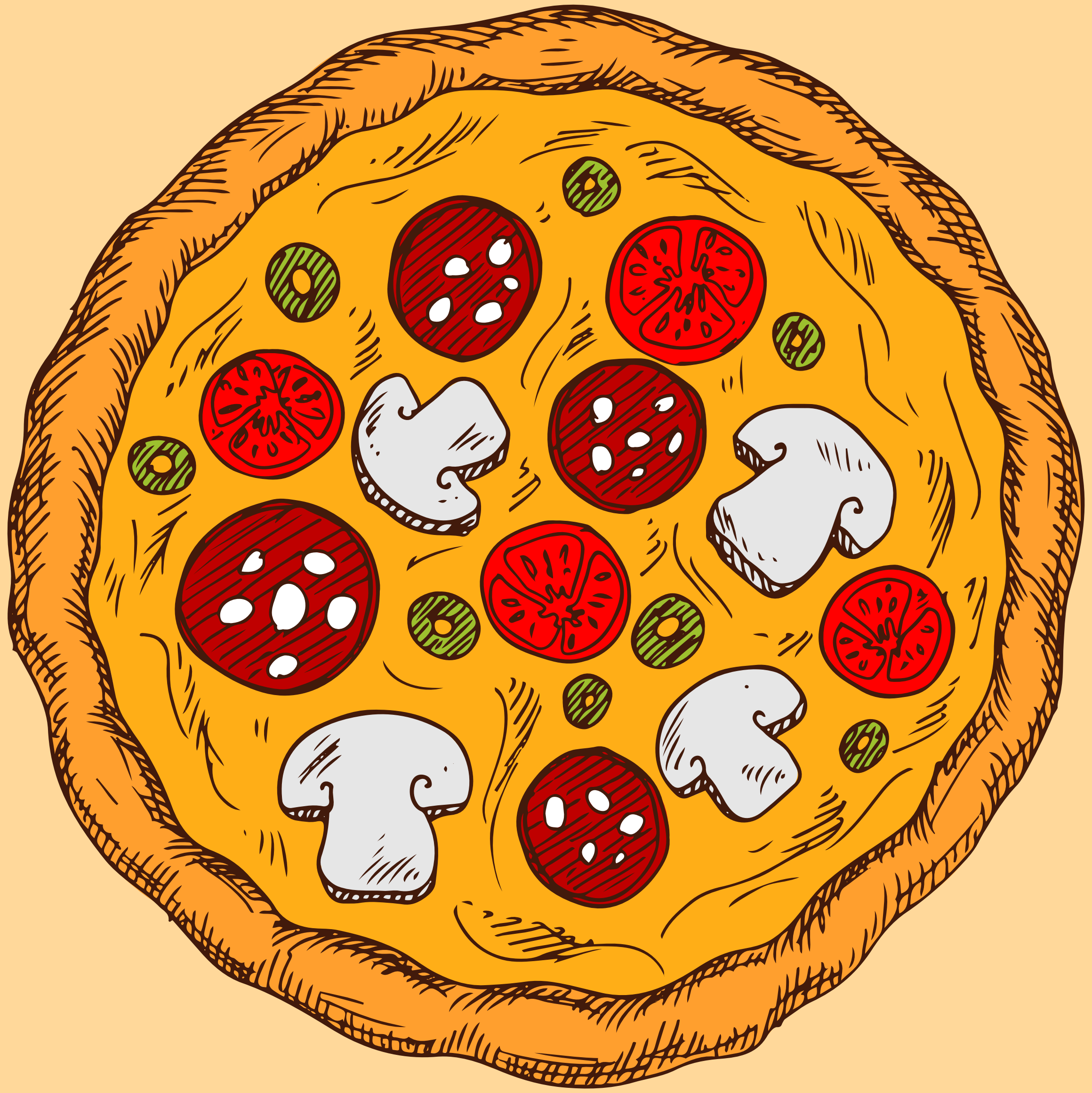
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```
select name, revenue, category 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	category
The Thai Chicken Pizza	43434.25	Chicken
The Barbecue Chicken Pizza	42768	Chicken
The California Chicken Pizza	41409.5	Chicken
The Classic Deluxe Pizza	38180.5	Classic
The Hawaiian Pizza	32273.25	Classic
The Pepperoni Pizza	30161.75	Classic
The Spicy Italian Pizza	34831.25	Supreme
The Italian Supreme Pizza	33476.75	Supreme
The Sicilian Pizza	30940.5	Supreme
The Four Cheese Pizza	32265.70000000065	Veggie
The Mexicana Pizza	26780.75	Veggie
The Five Cheese Pizza	26066.5	Veggie



THANK YOU