SQL PIZZA SALES ANALYSIS PROJECT

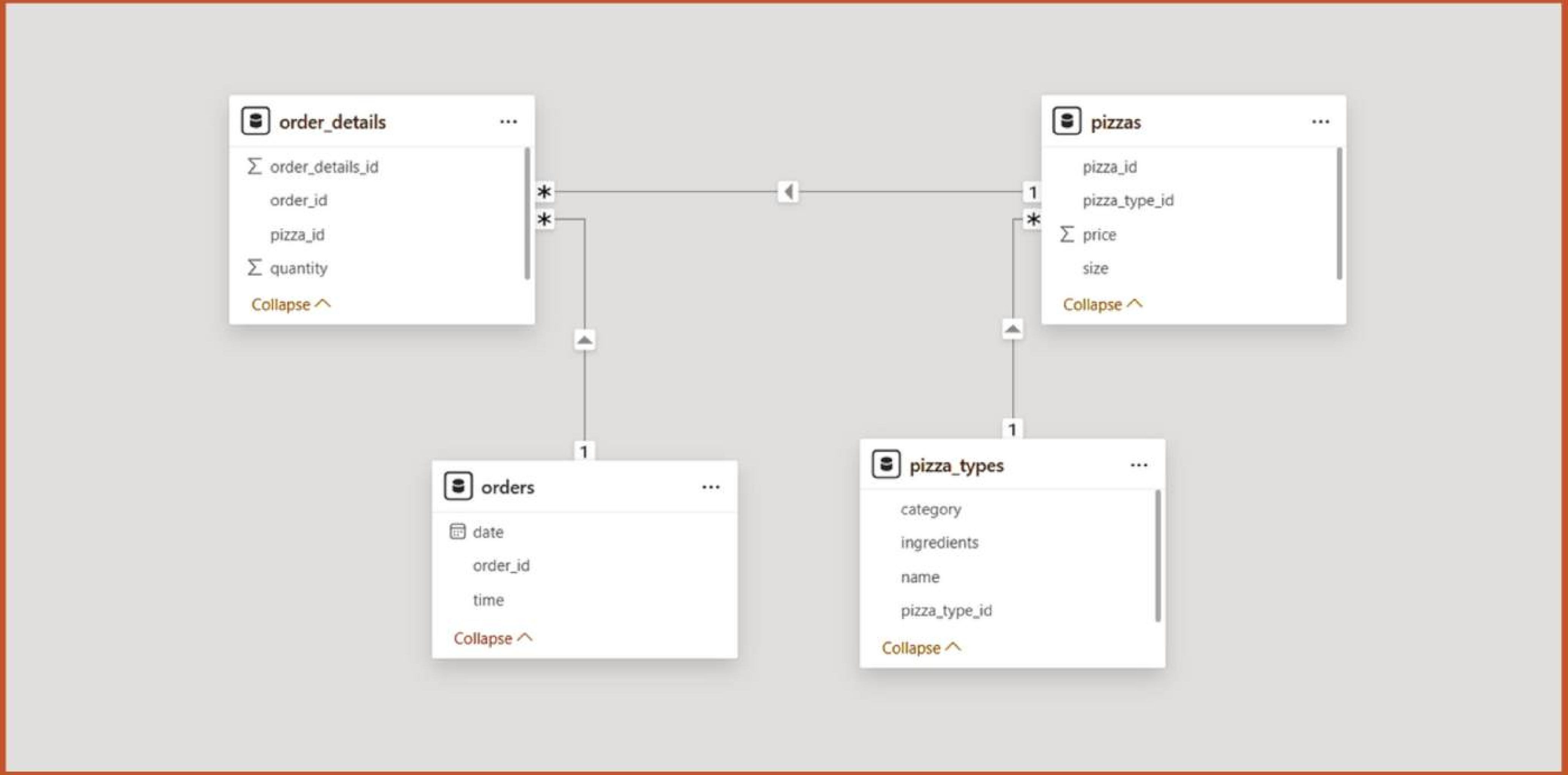
Data analysis project

Tools used: MySQL | Joins | Aggregations | Window Functions

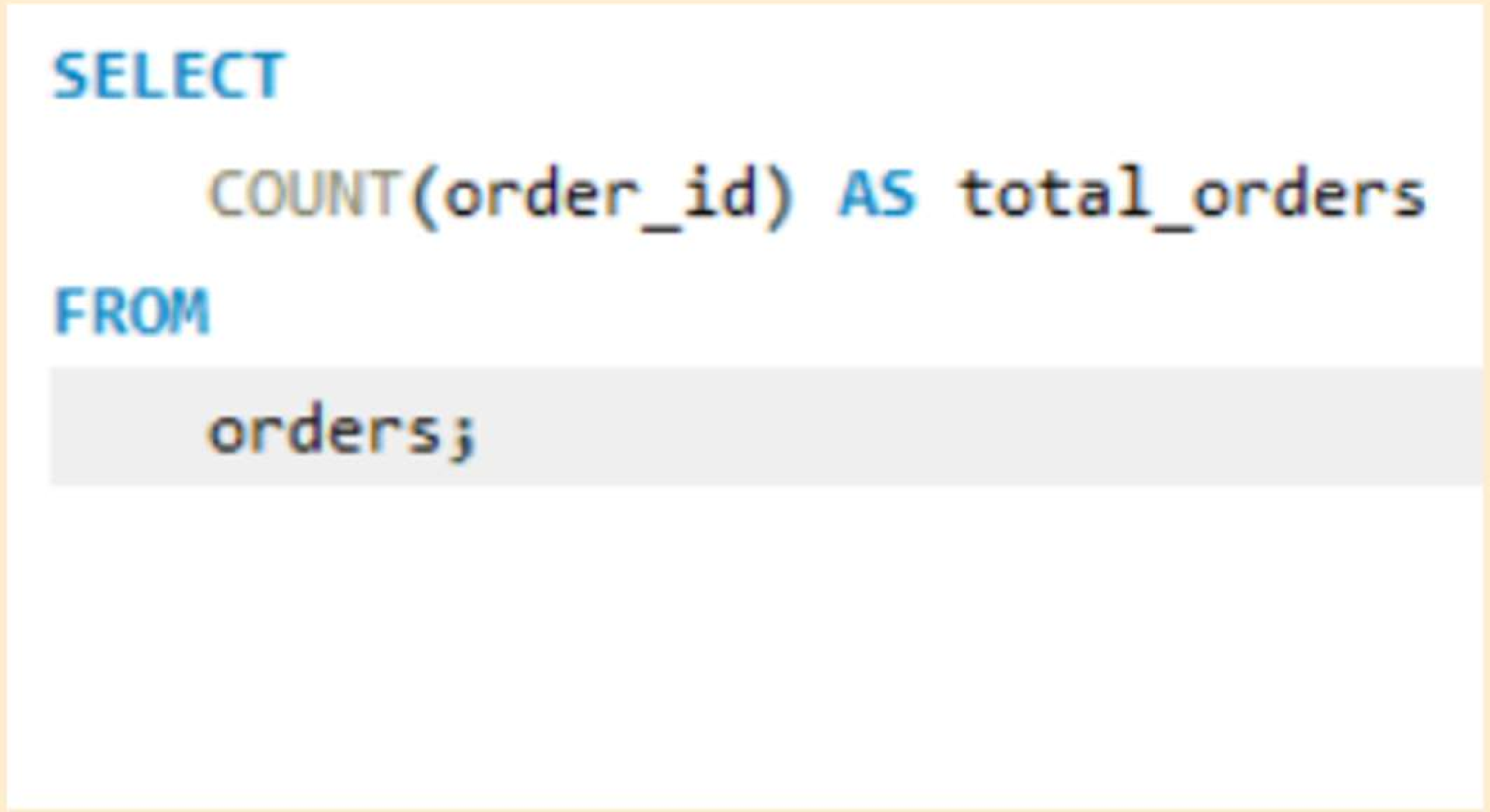
INTRODUCTION

Hello, my name is Chanjal. This project showcases how I used SQL queries to analyze pizza sales data and answer key business questions related to orders, revenue, and product performance.

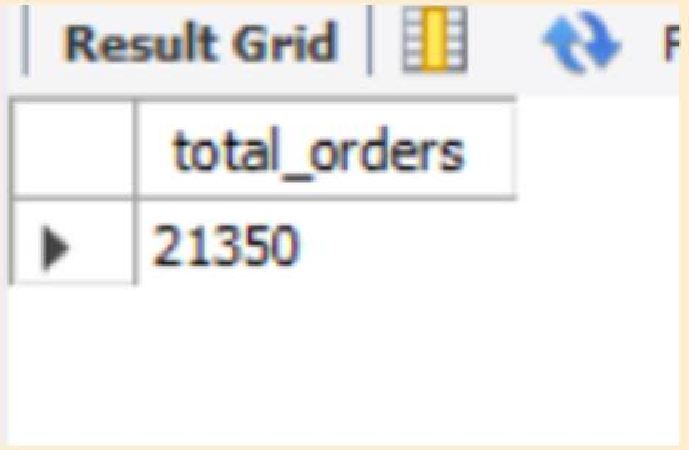
DATA MODEL_PIZZA



RETRIEVE TOTAL NUMBER OF ORDERS

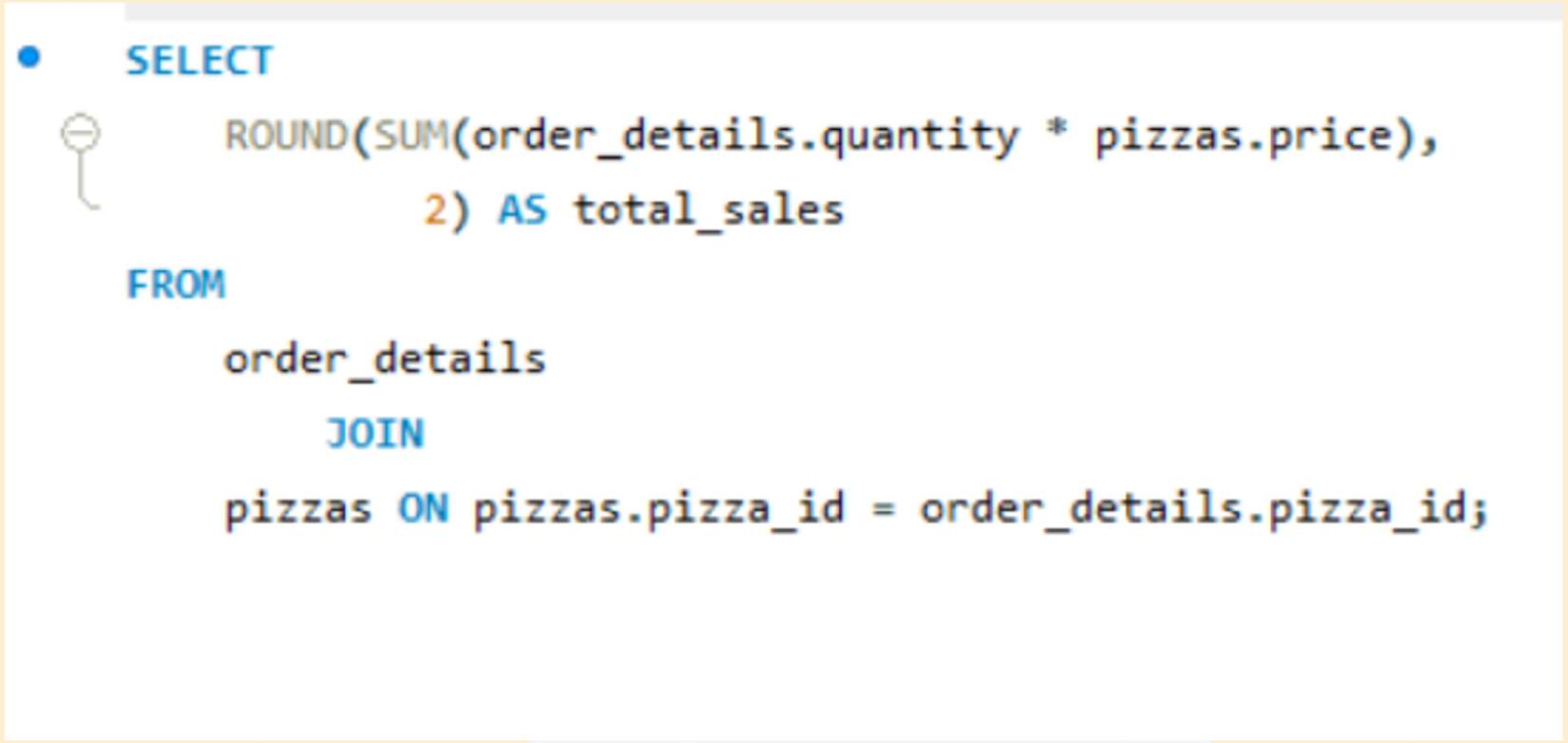


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

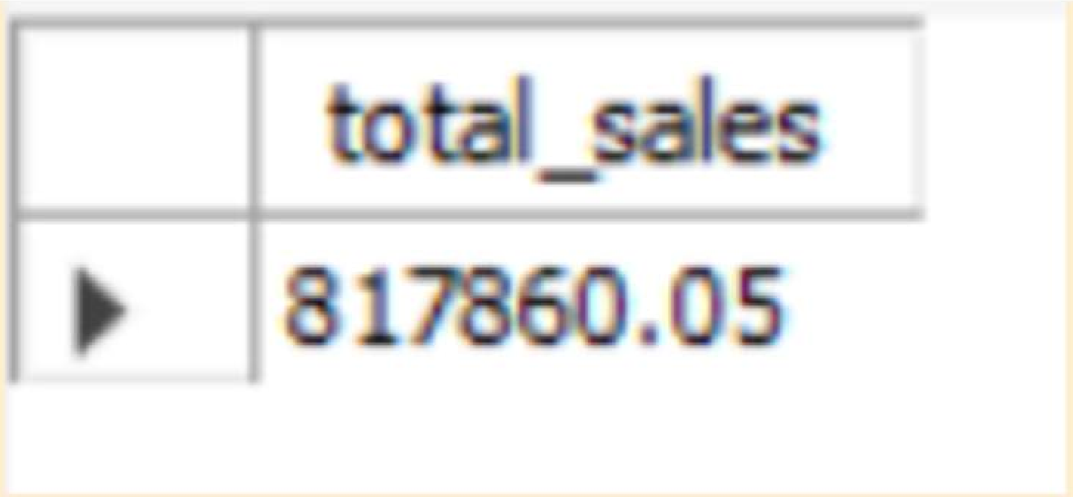


```
total_orders
21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

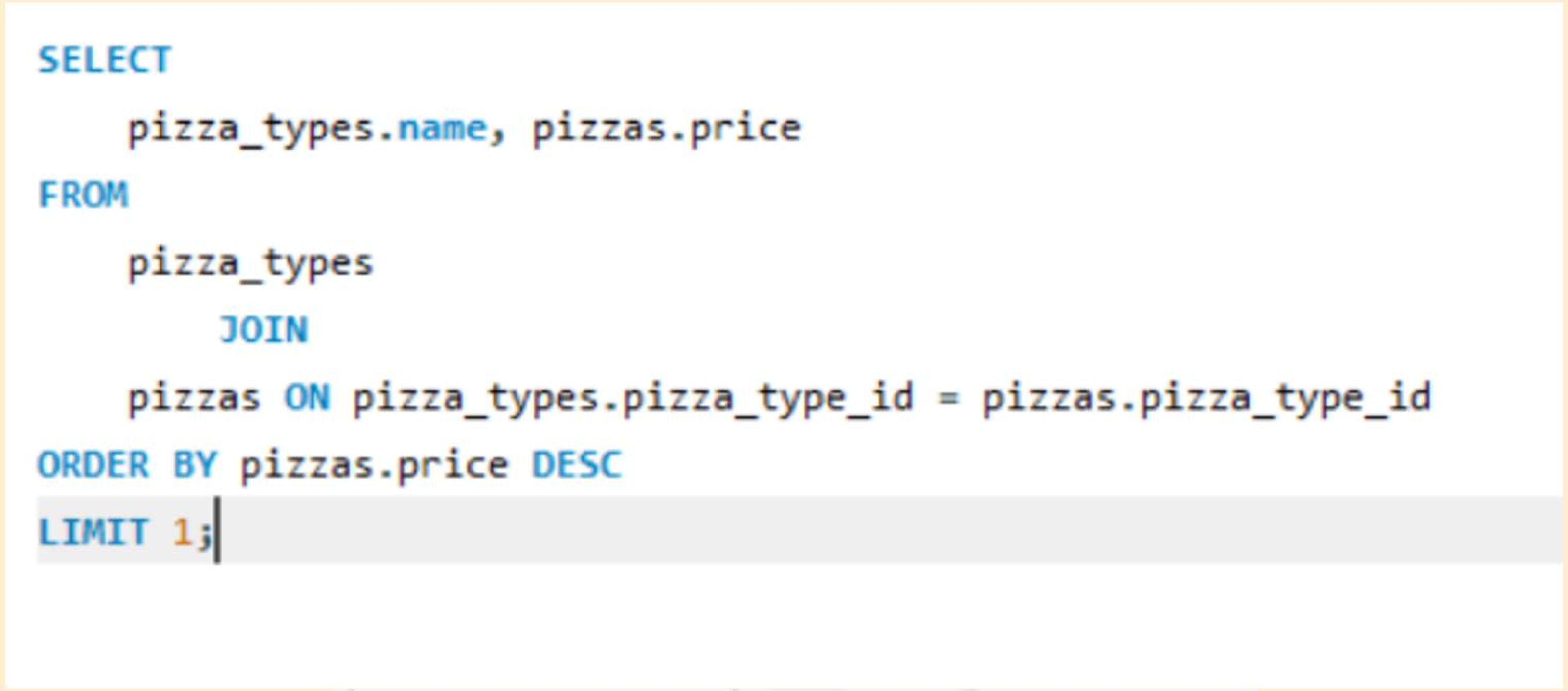


```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

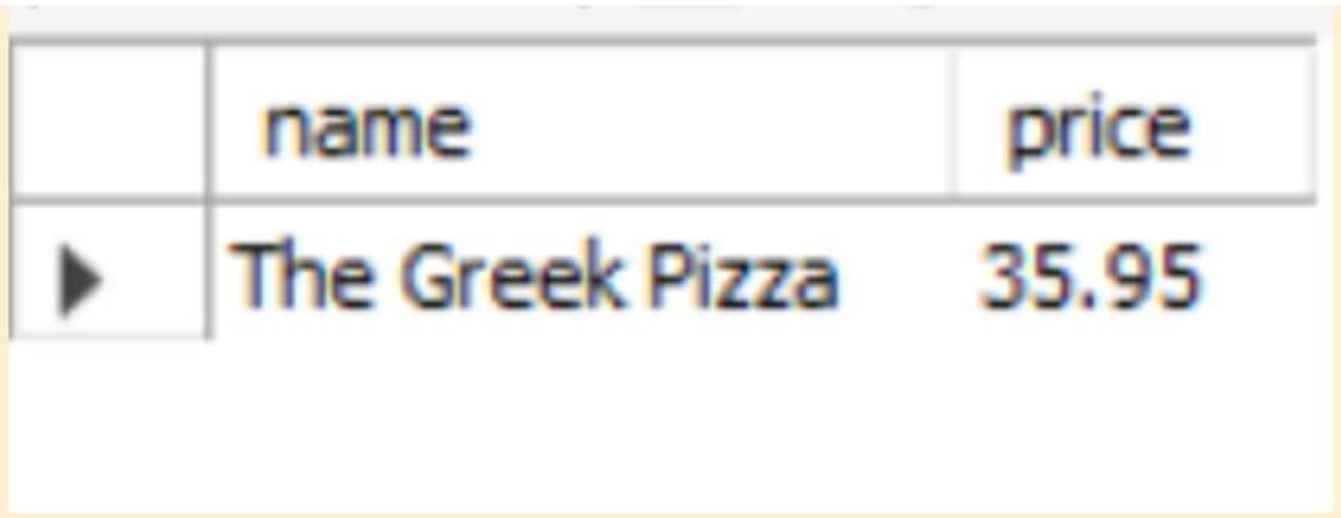


	total_sales
▶	817860.05

Find the highest priced pizza

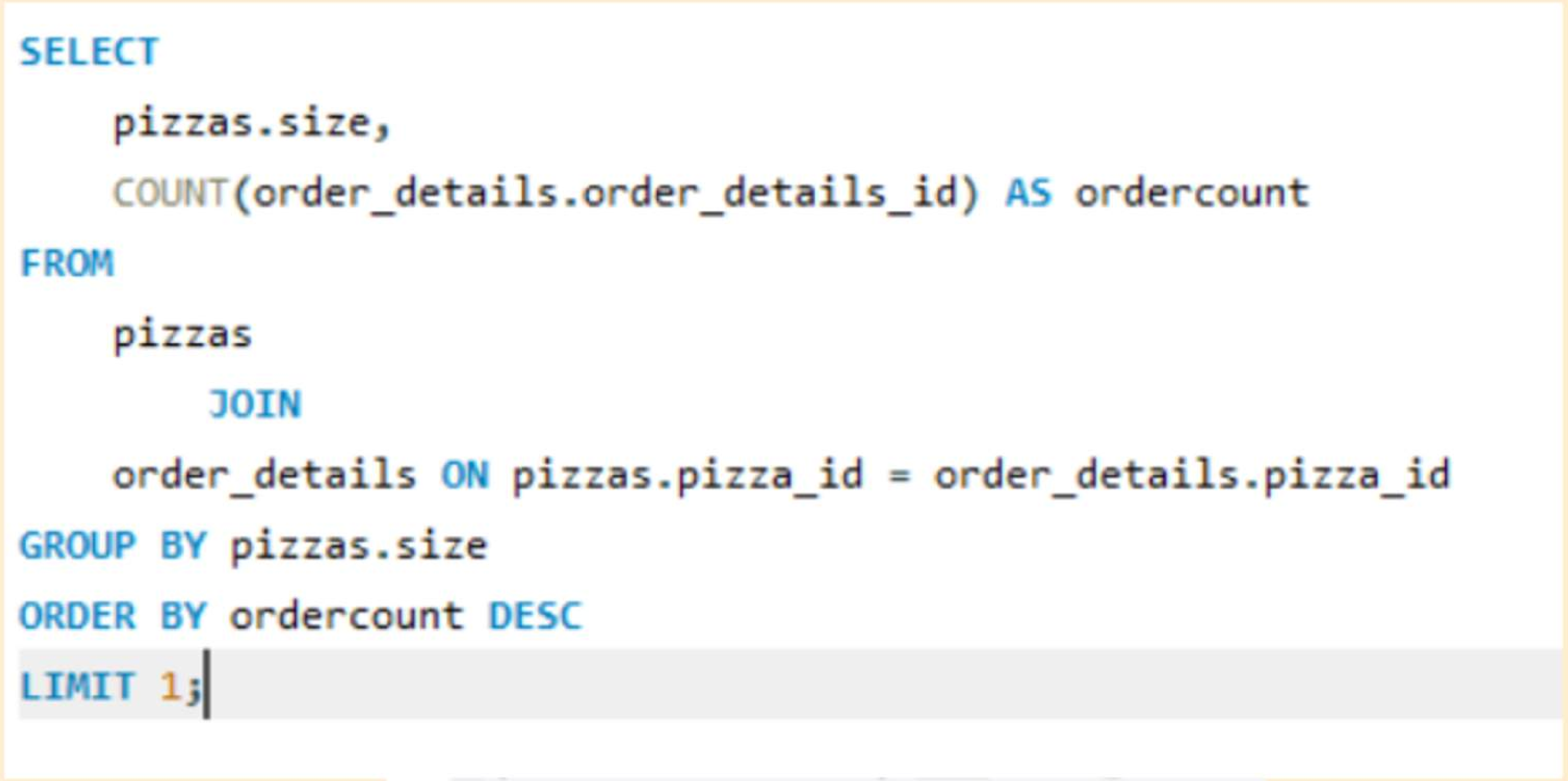


```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

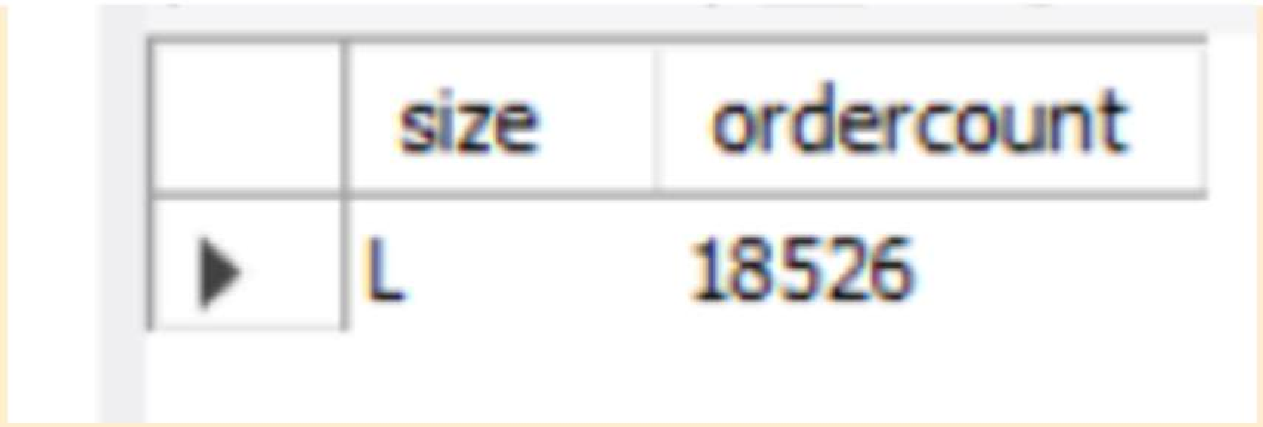


	name	price
▶	The Greek Pizza	35.95

Identify the most ordered pizza size

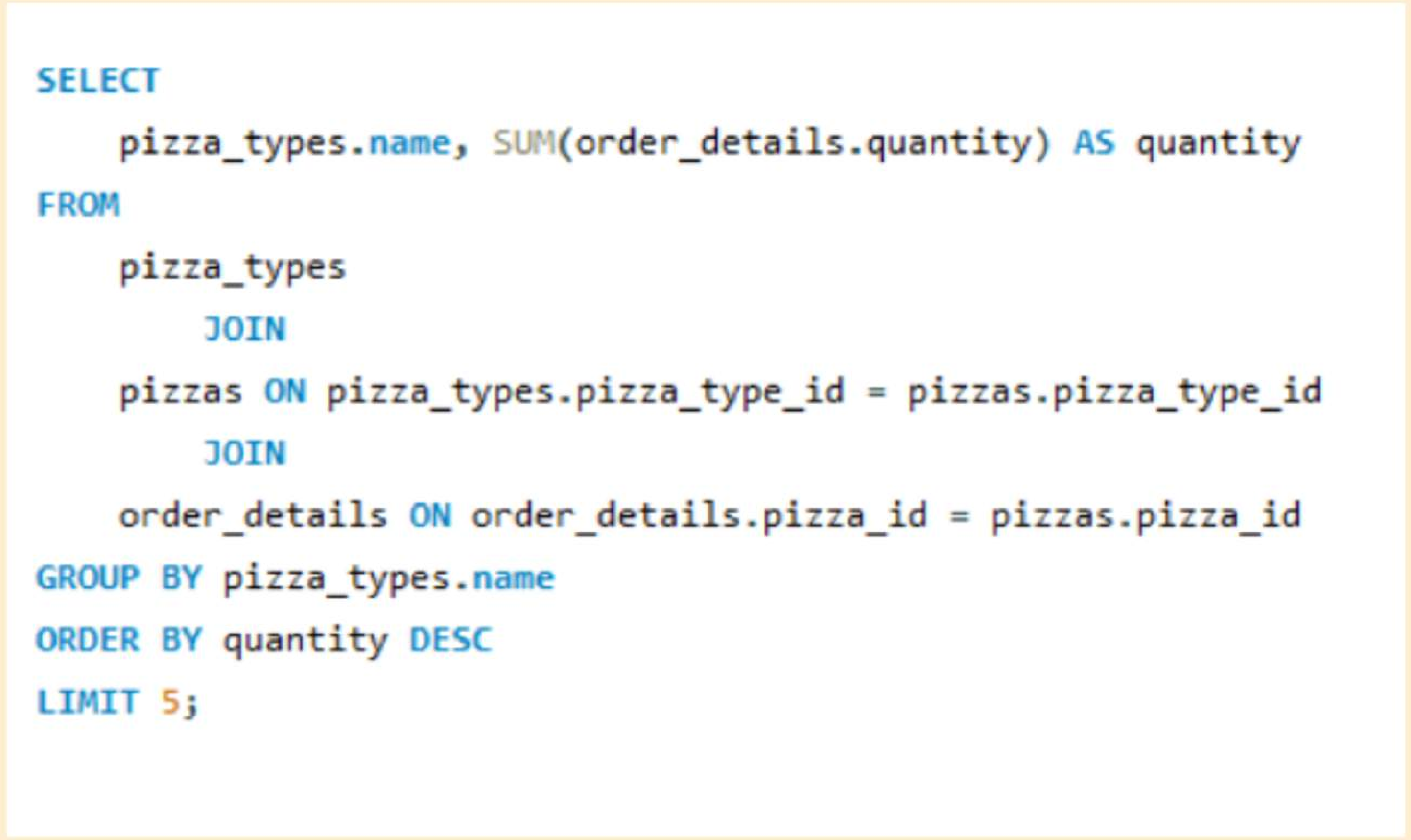


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS ordercount
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY ordercount DESC
LIMIT 1;
```

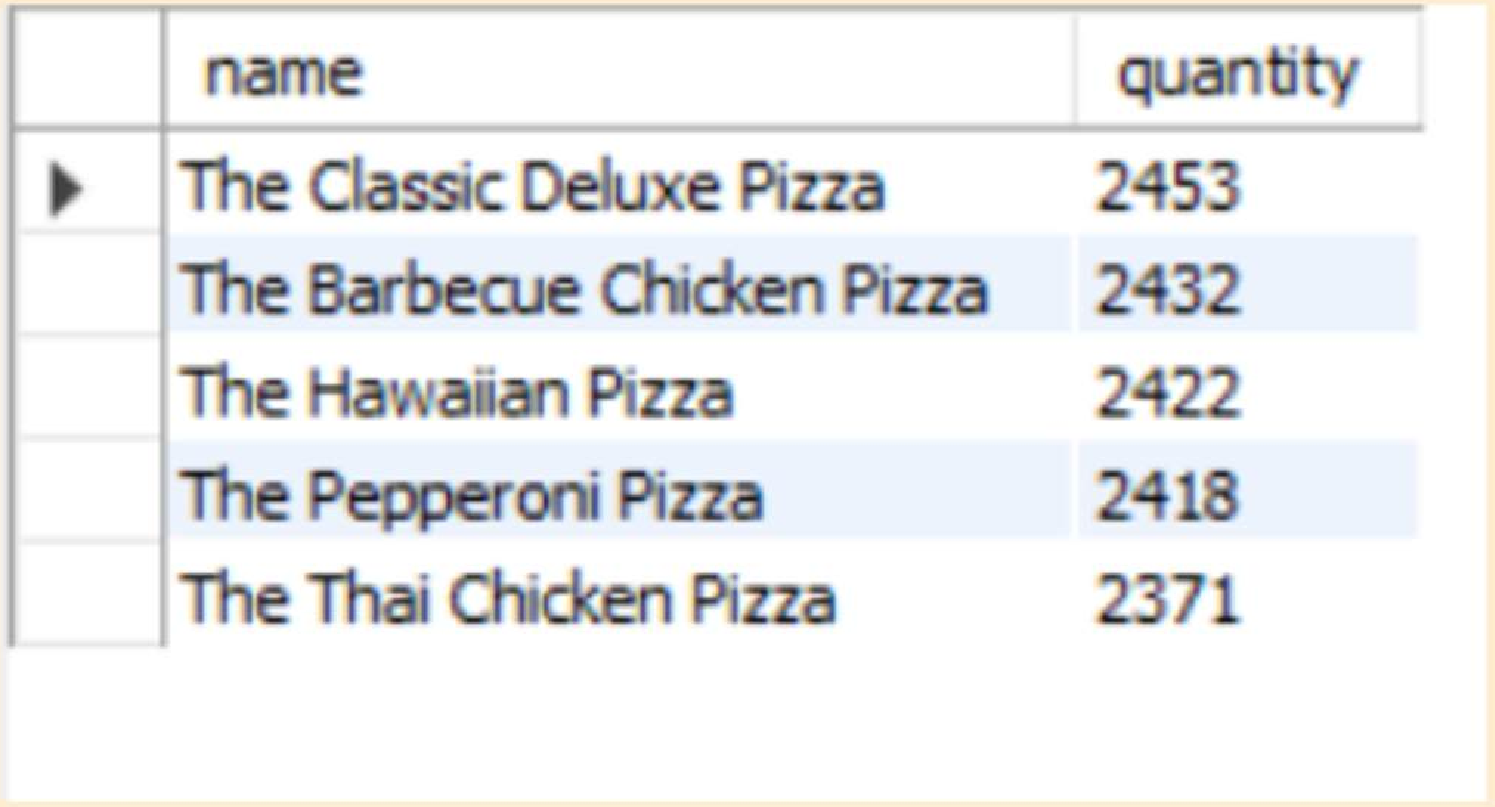


	size	ordercount
▶	L	18526

List top 5 most ordered pizza type along with their quantities

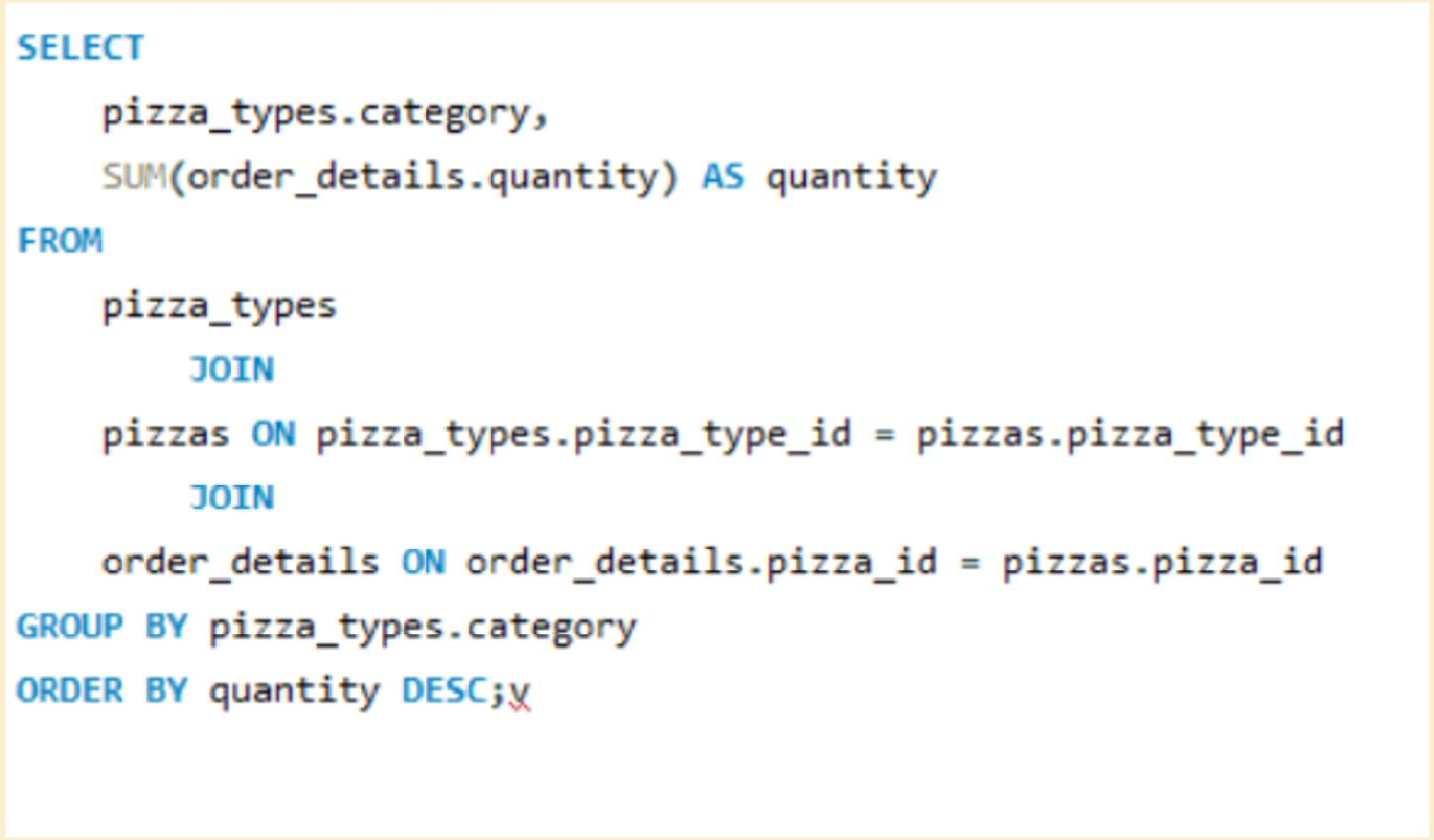


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

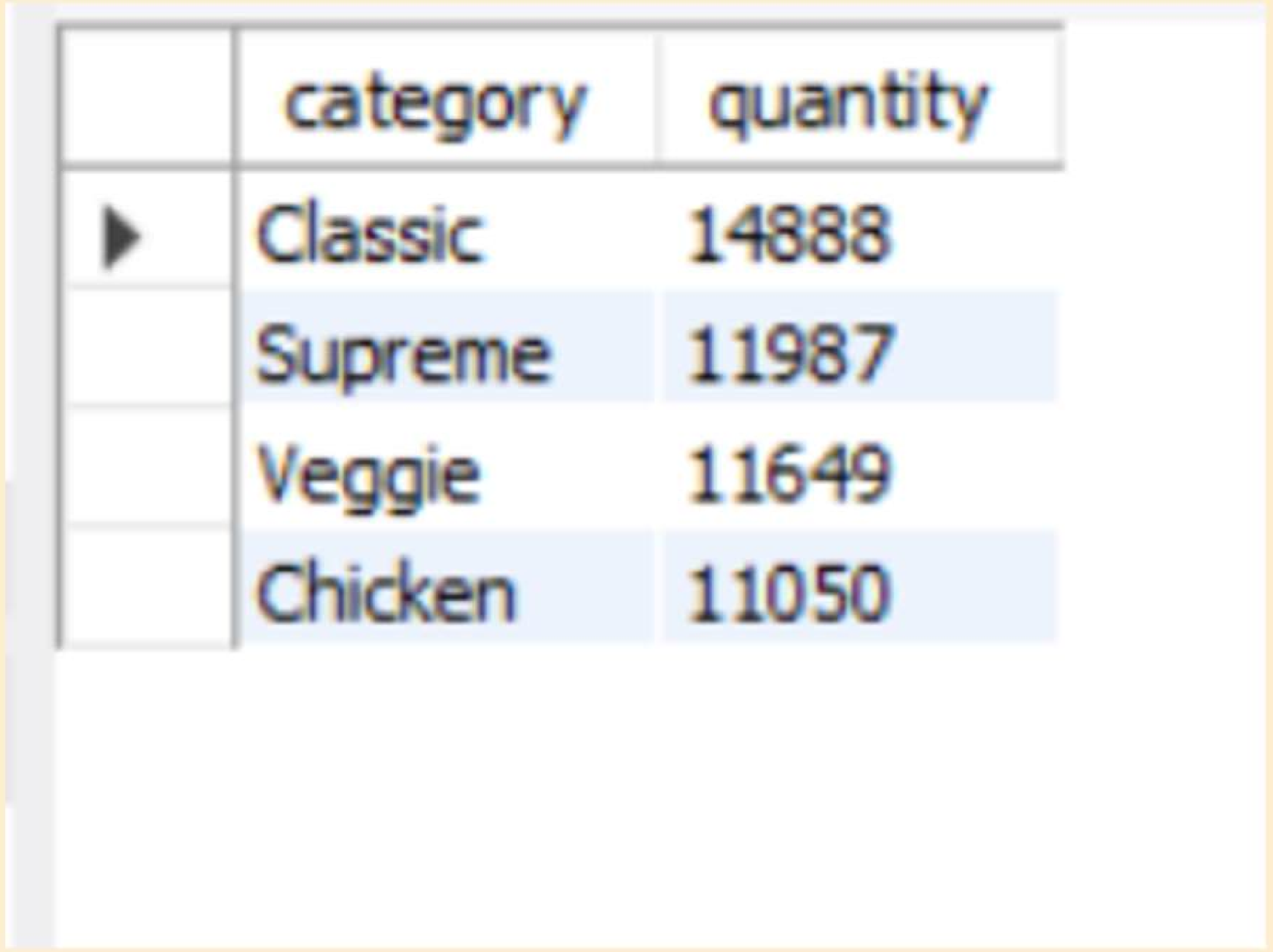


	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

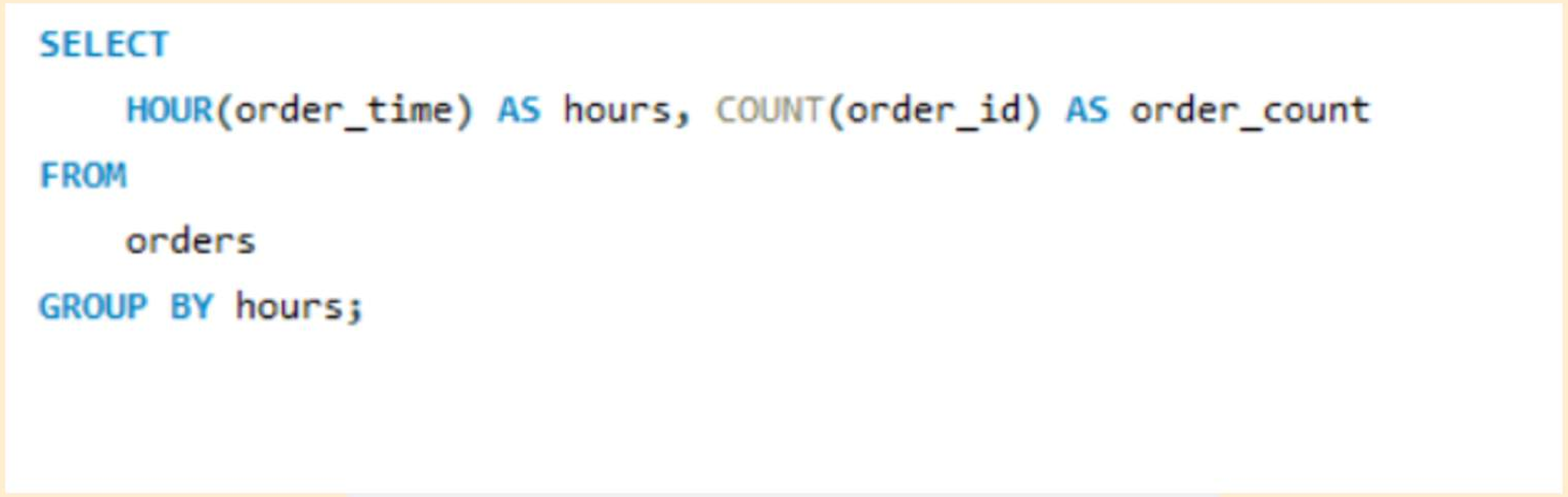


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

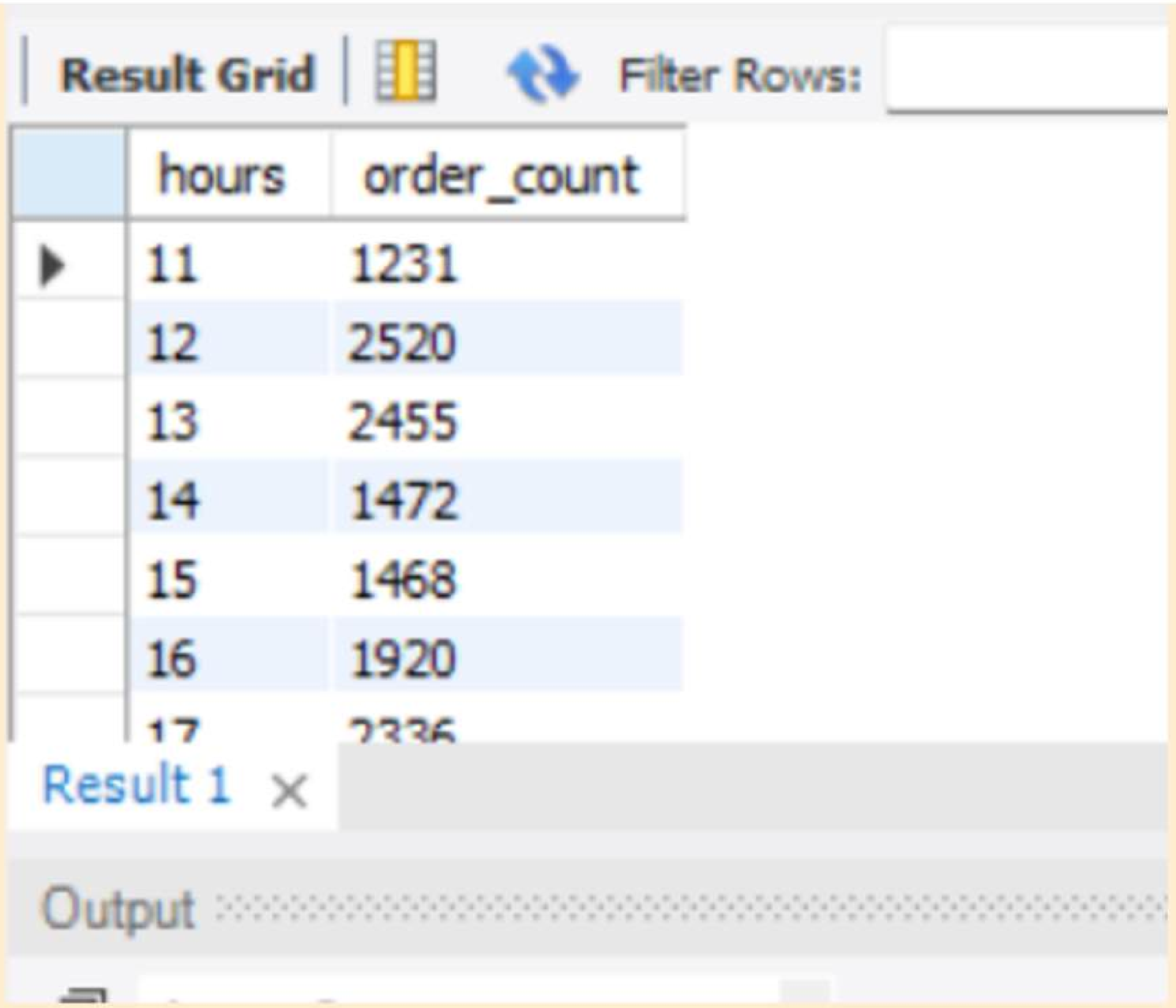


	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hours of the day



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



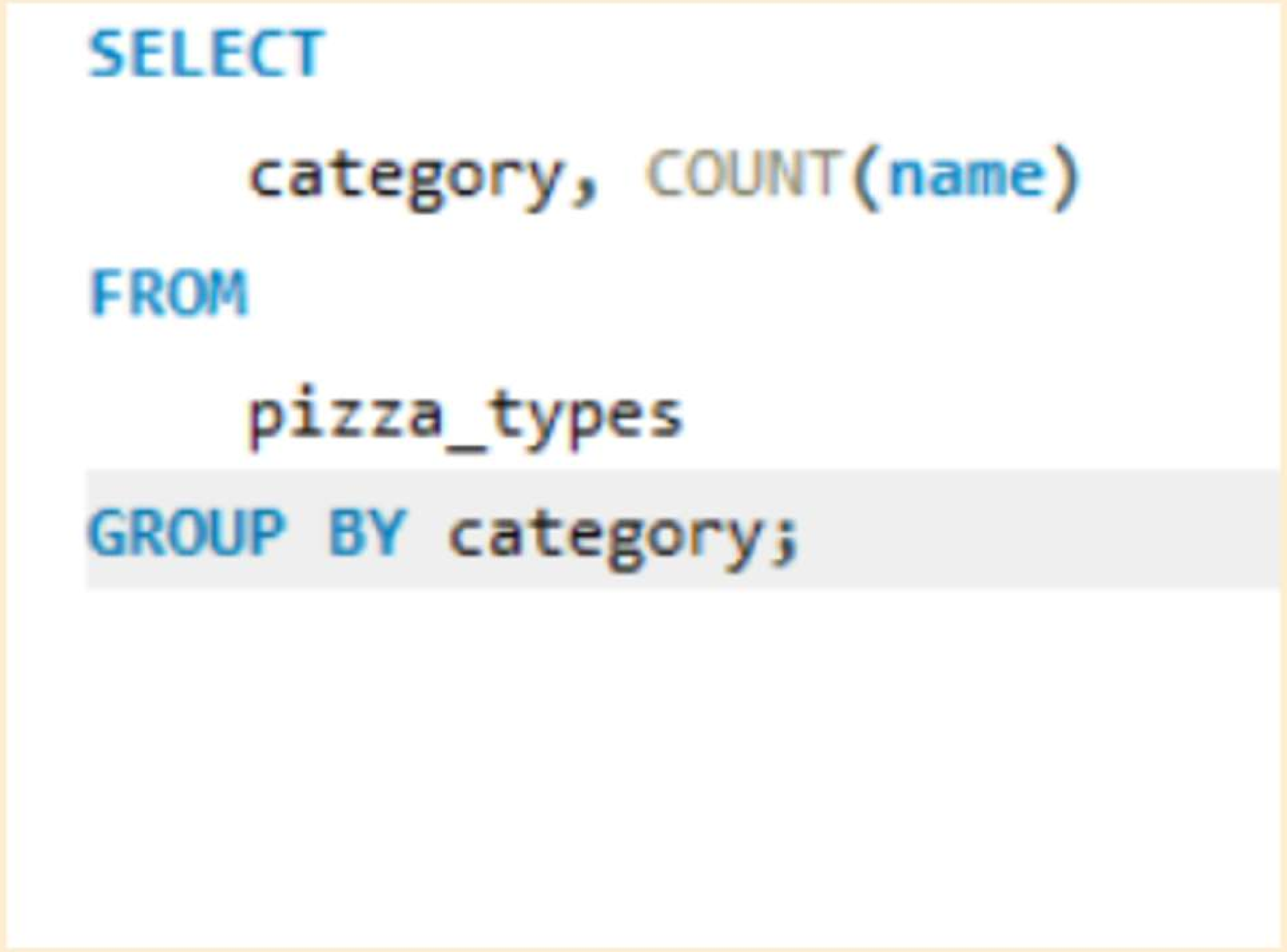
Result Grid | Filter Rows:

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

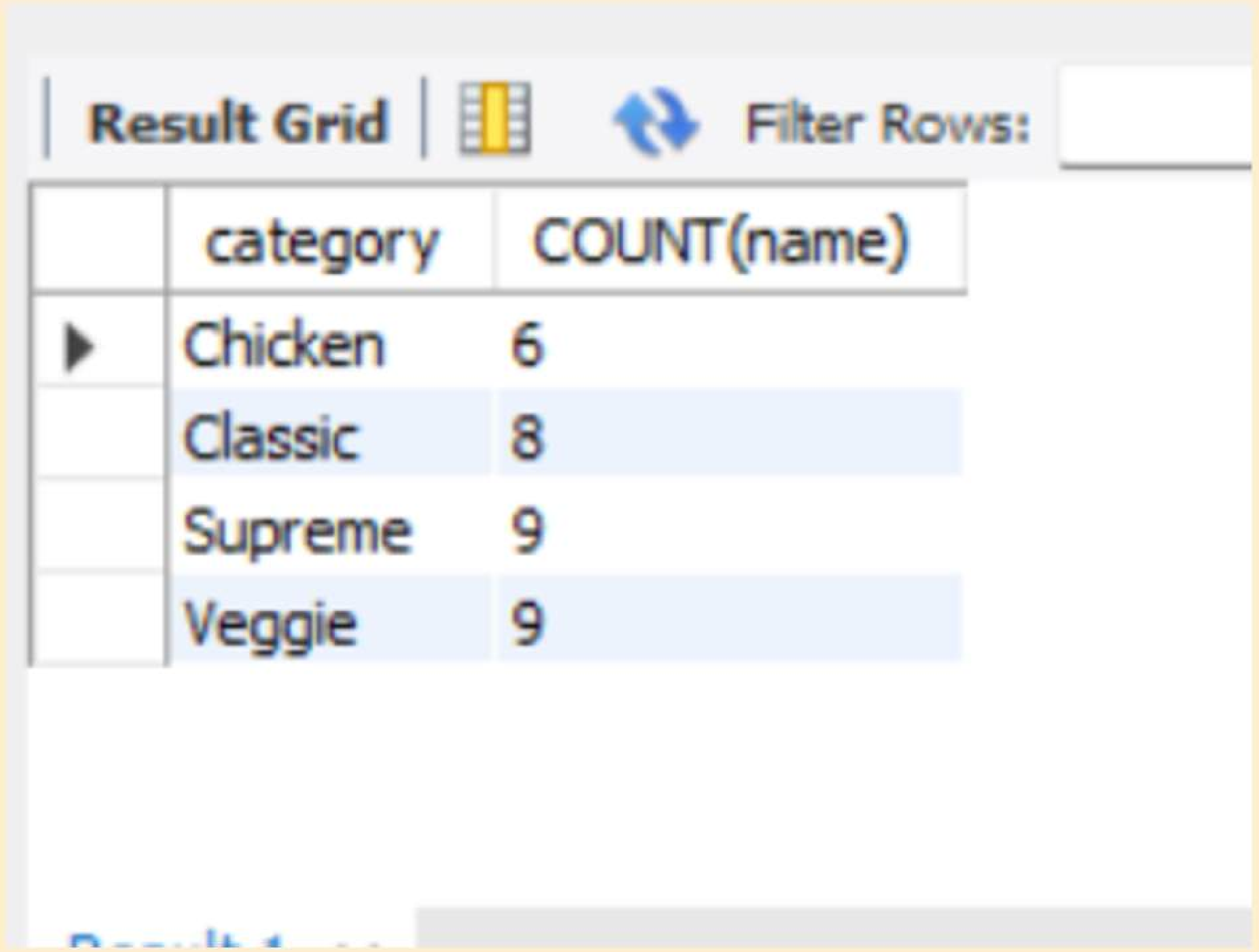
Result 1 ×

Output

join relevant tables to find the category wise distribution of pizzas

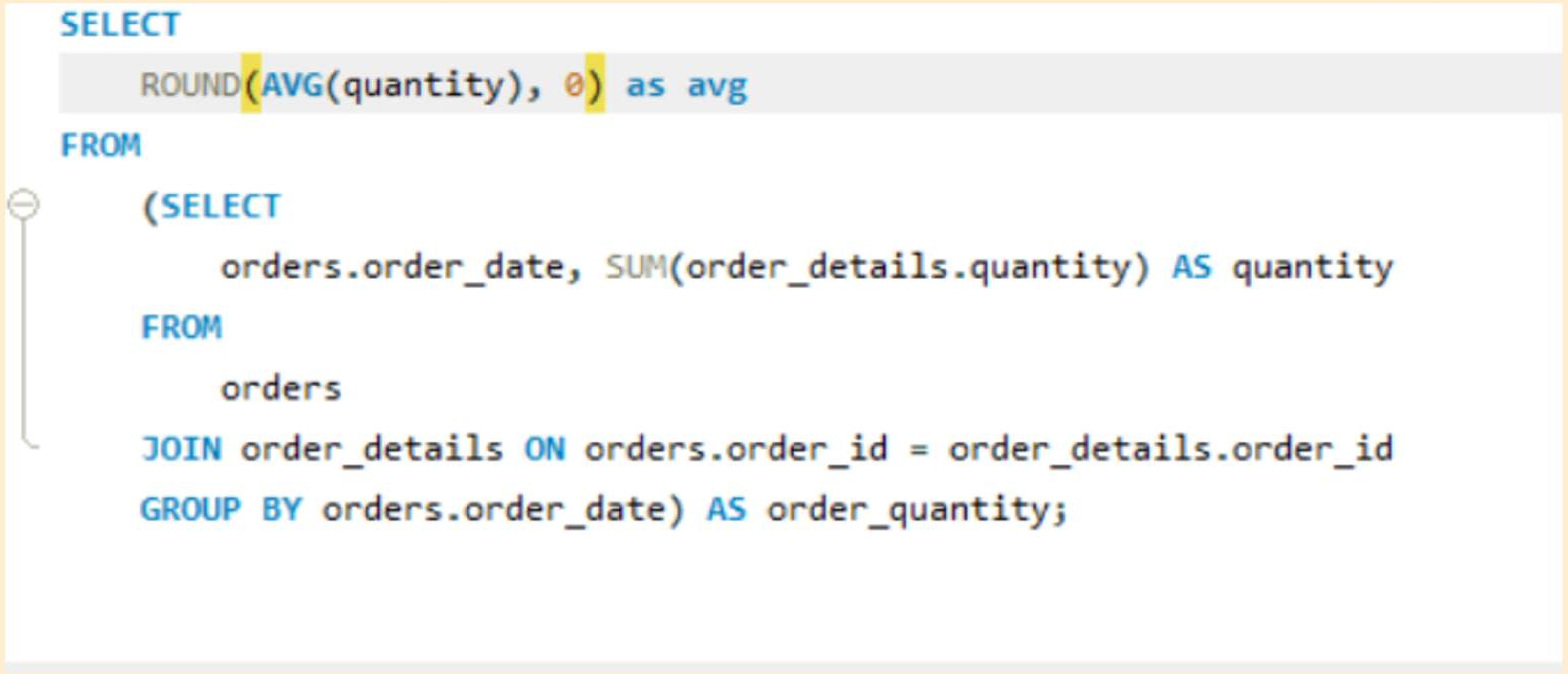


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

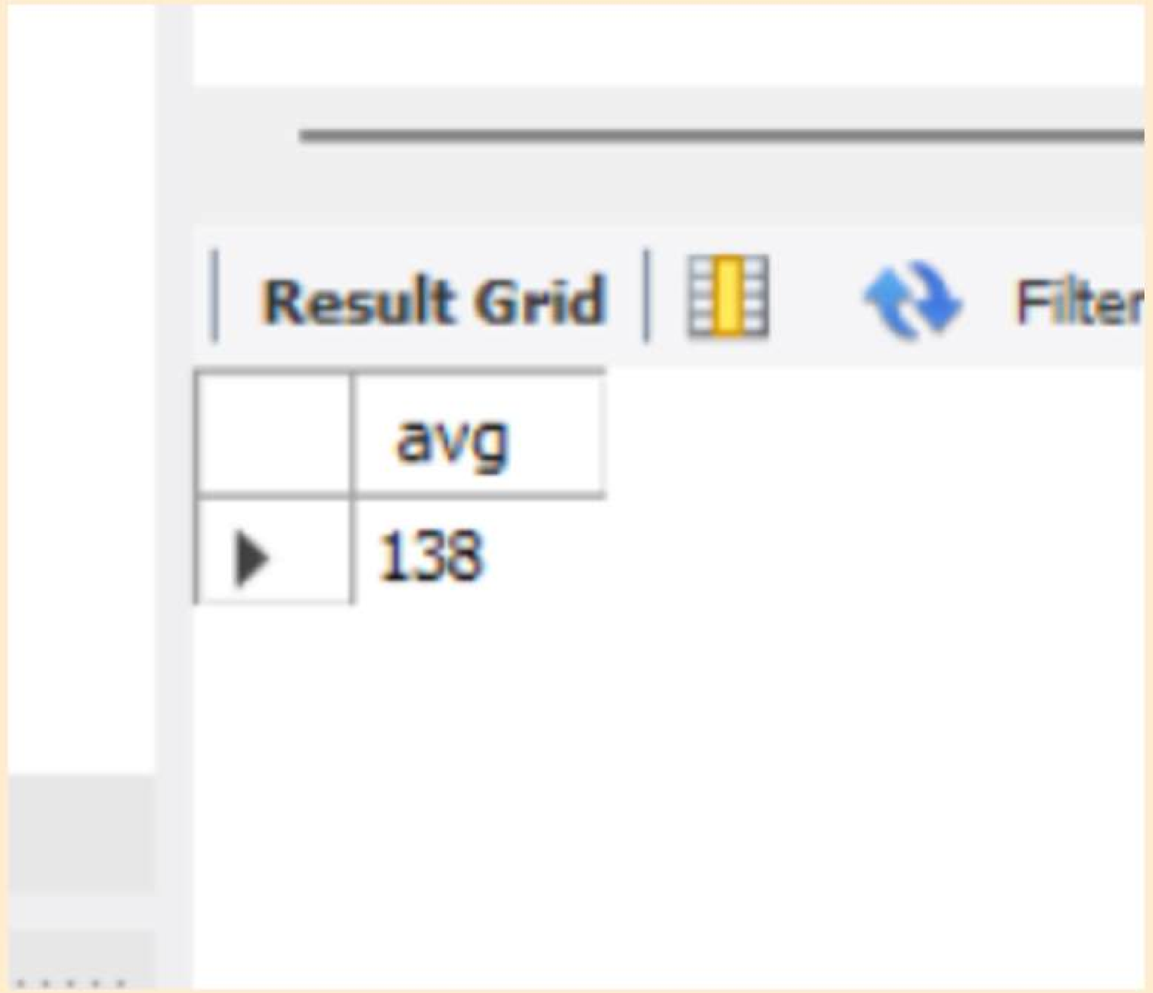


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



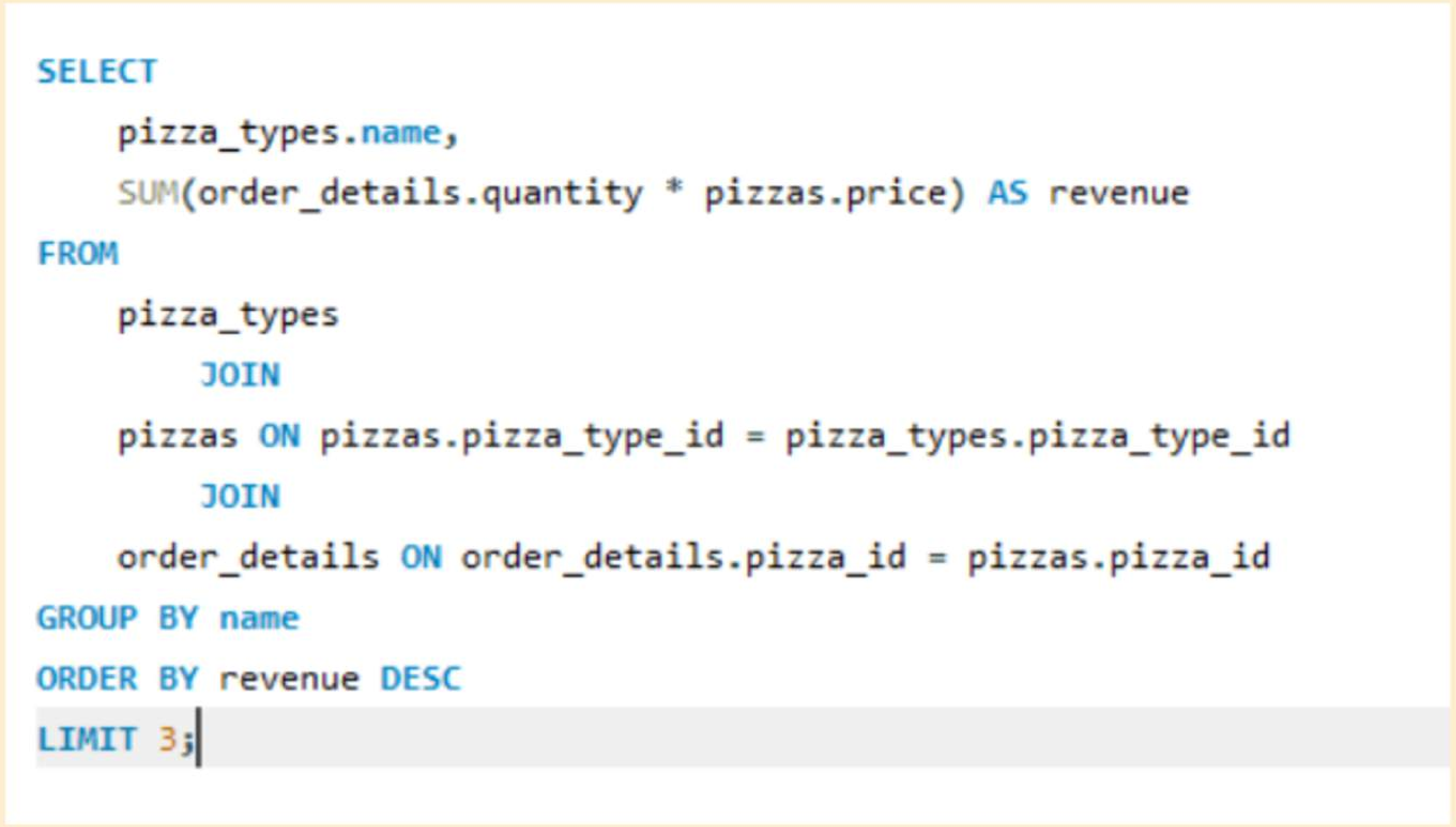
```
SELECT
    ROUND(AVG(quantity), 0) as avg
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



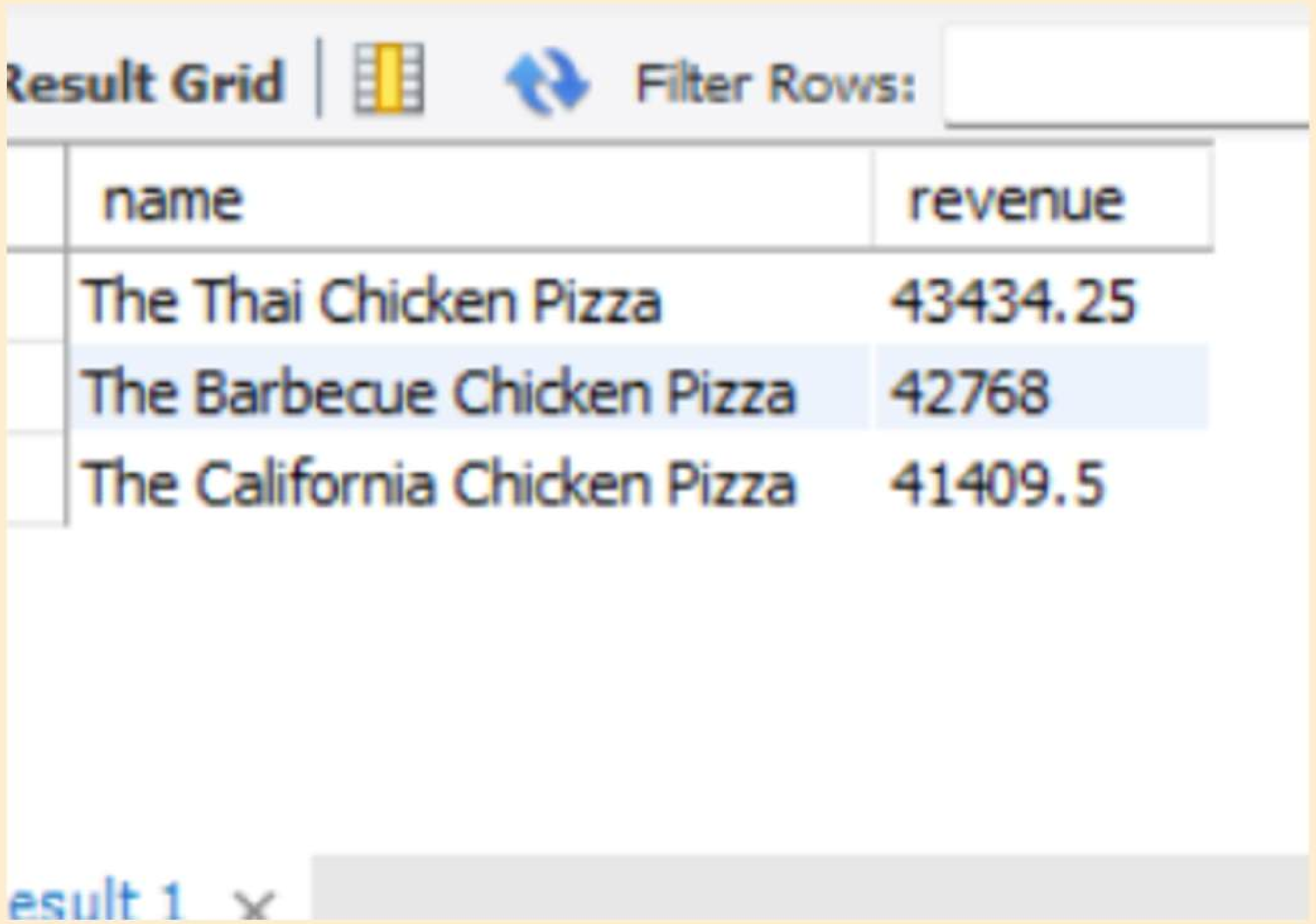
Result Grid | Filter

```
avg
138
```

Determine the top 3 most ordered pizza type based on the revenue

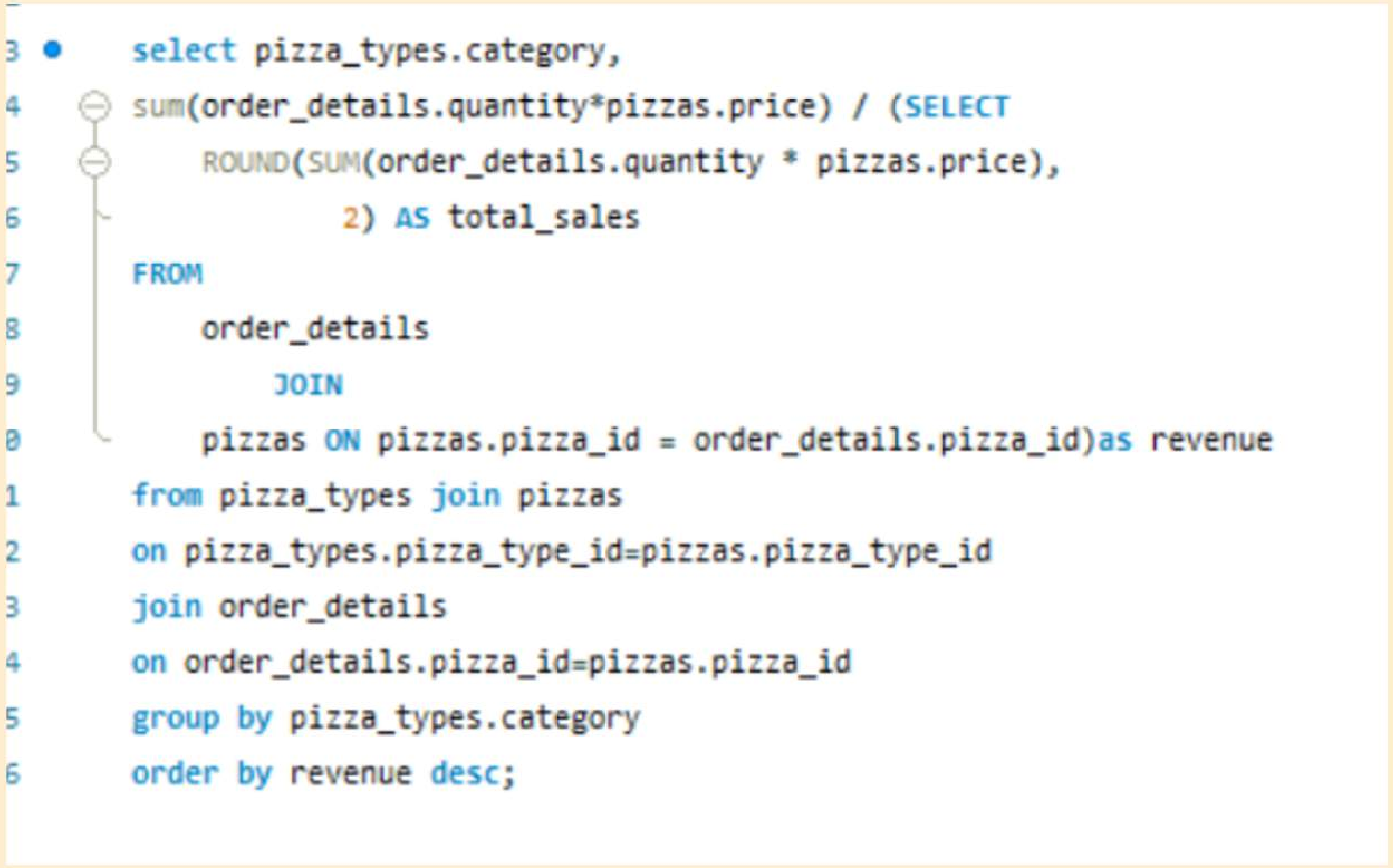


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY name
ORDER BY revenue DESC
LIMIT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

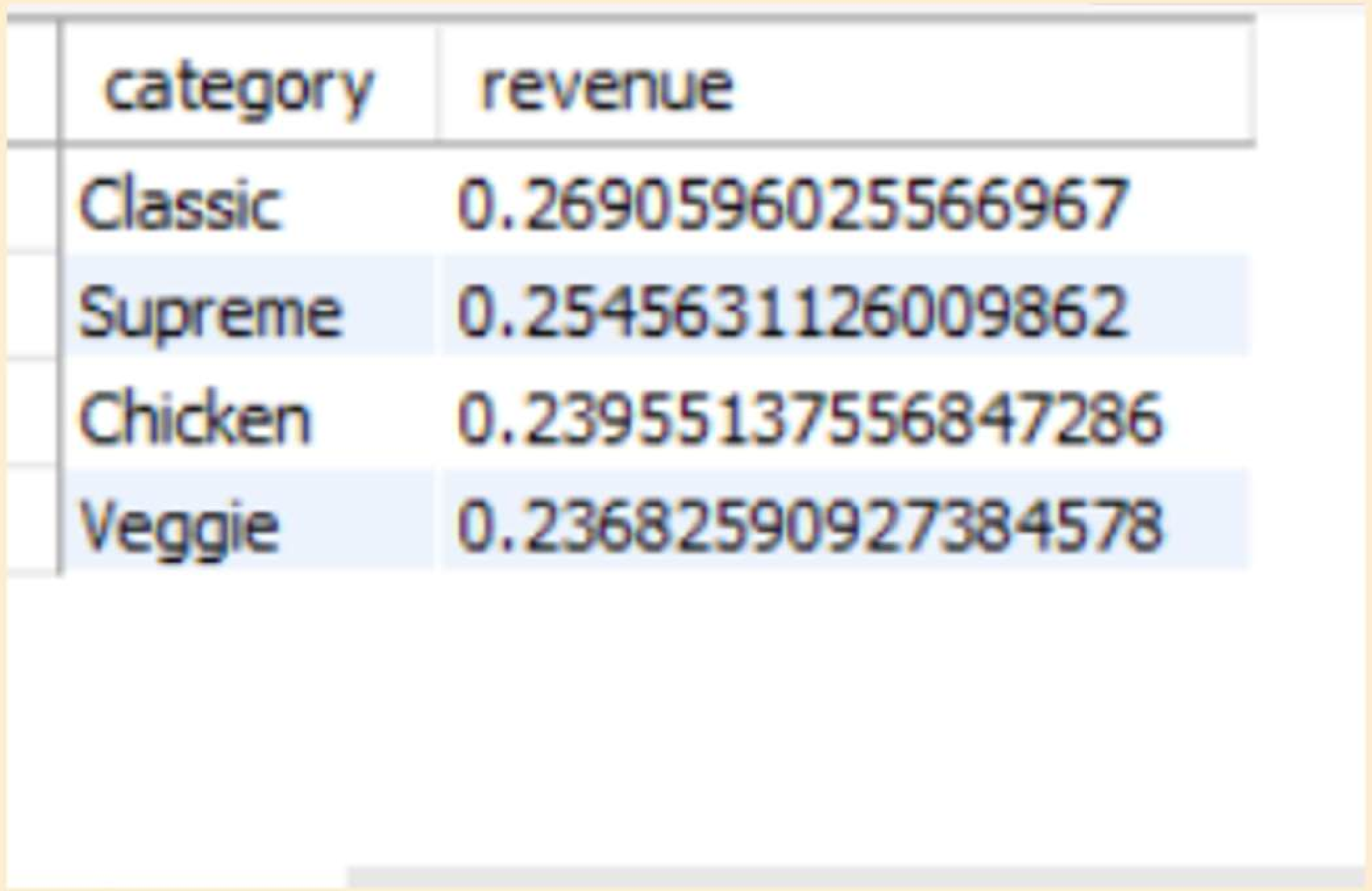


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

calculate the percentage contribution of each pizza type to total revenue

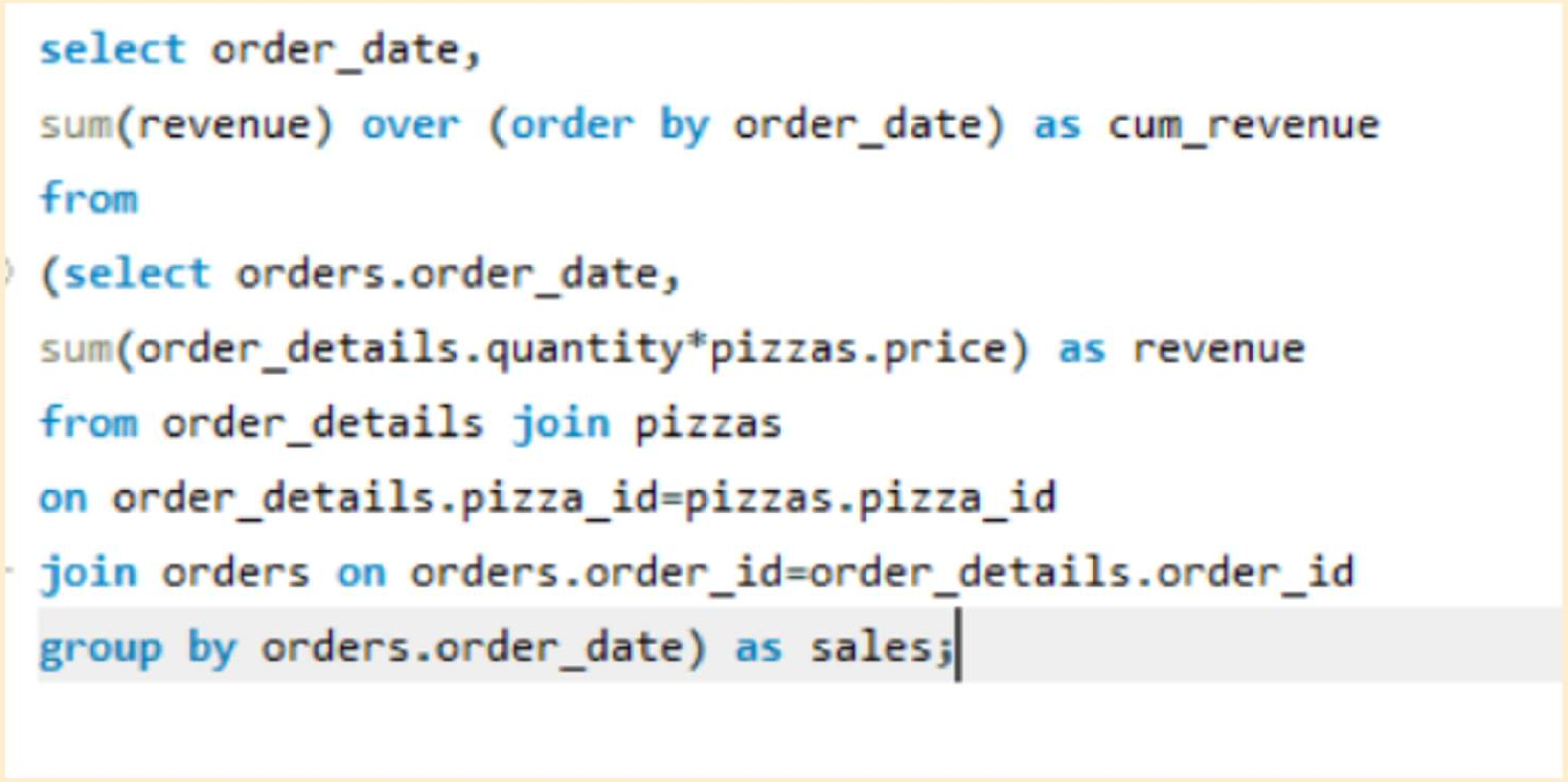


```
select pizza_types.category,
sum(order_details.quantity*pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id)as revenue
from pizza_types join pizzas
on 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
order by revenue desc;
```

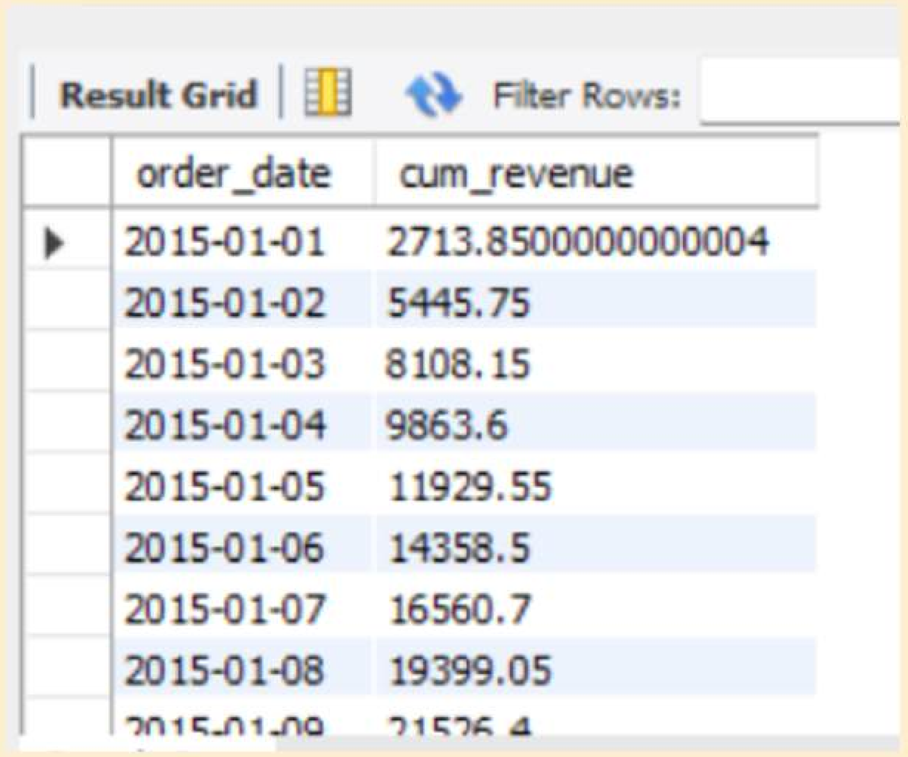


```
category   revenue
Classic    0.2690596025566967
Supreme    0.2545631126009862
Chicken    0.23955137556847286
Veggie     0.23682590927384578
```

Select the cumulative revenue generated over time

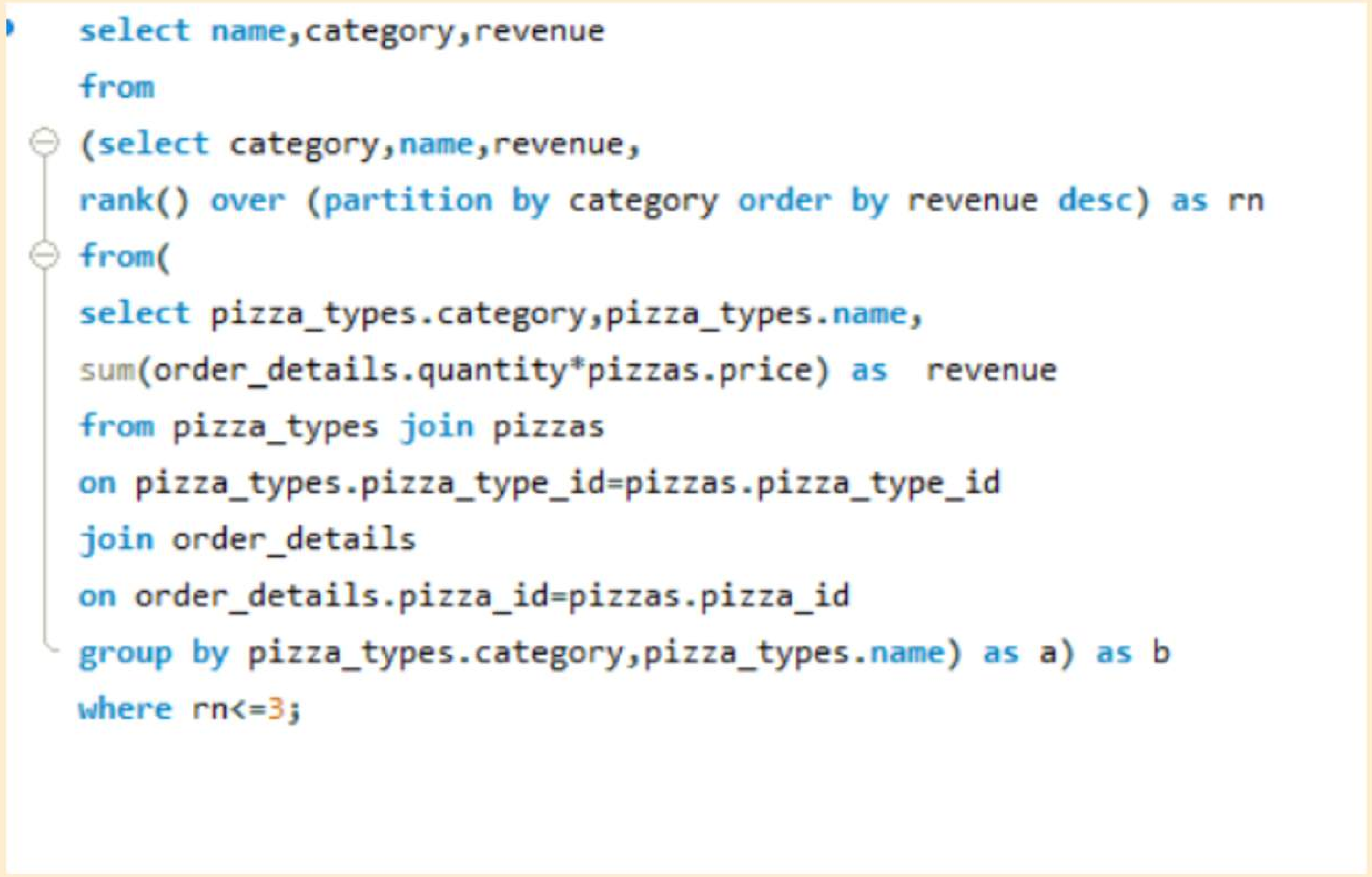


```sql
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join orders on orders.order_id=order_details.order_id
group by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

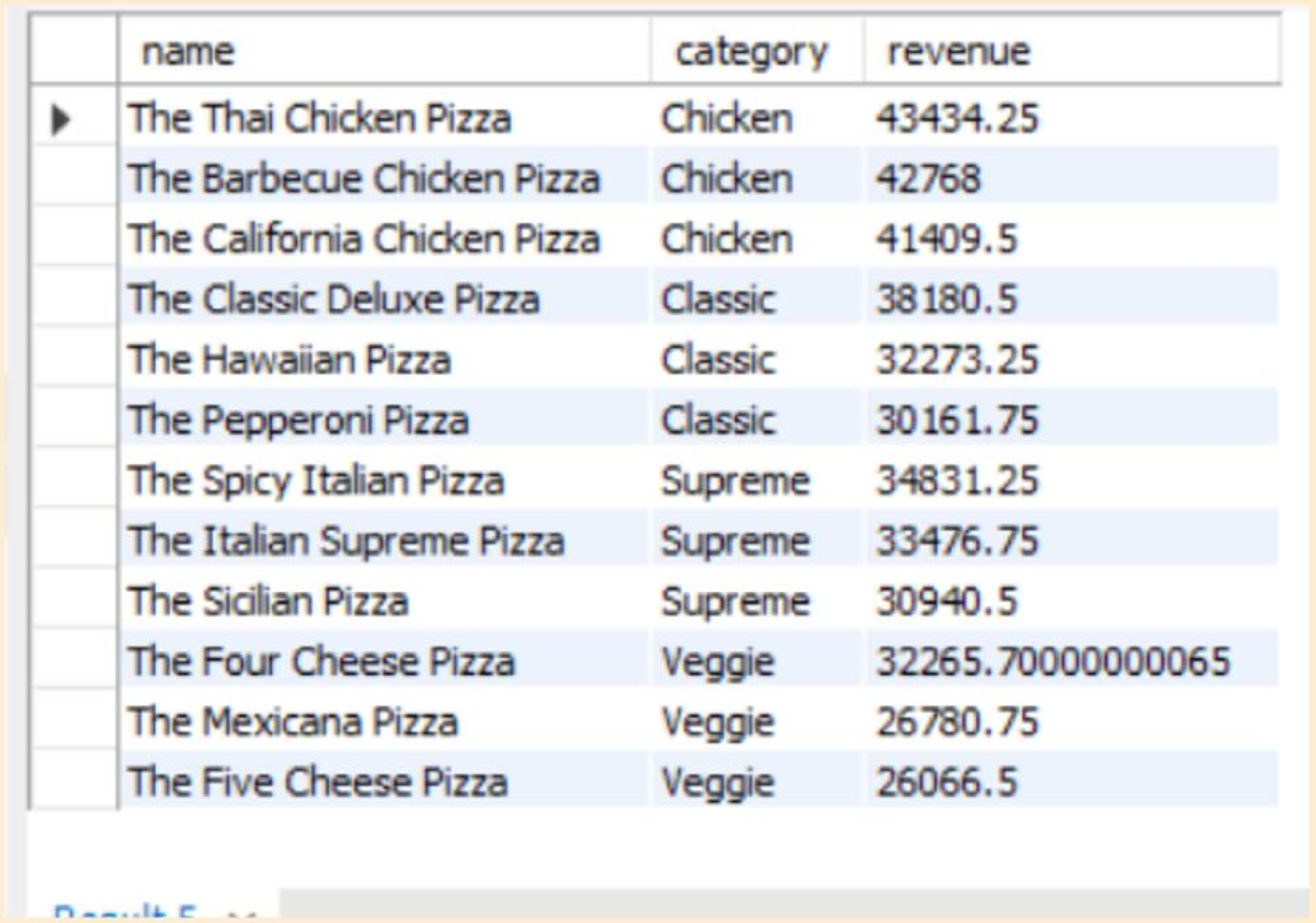


```
order_date    cum_revenue
2015-01-01    2713.8500000000004
2015-01-02    5445.75
2015-01-03    8108.15
2015-01-04    9863.6
2015-01-05    11929.55
2015-01-06    14358.5
2015-01-07    16560.7
2015-01-08    19399.05
2015-01-09    21526.4
```

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



```sql
select name,category,revenue
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	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.7000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

CONCLUSION

This SQL pizza sales analysis project demonstrates how data can be transformed into valuable insights using structured queries. The project focuses on understanding sales trends, popular products, and revenue performance, highlighting the importance of SQL in supporting business decision-making.

- Media : @reallygreatsite

THANK YOU