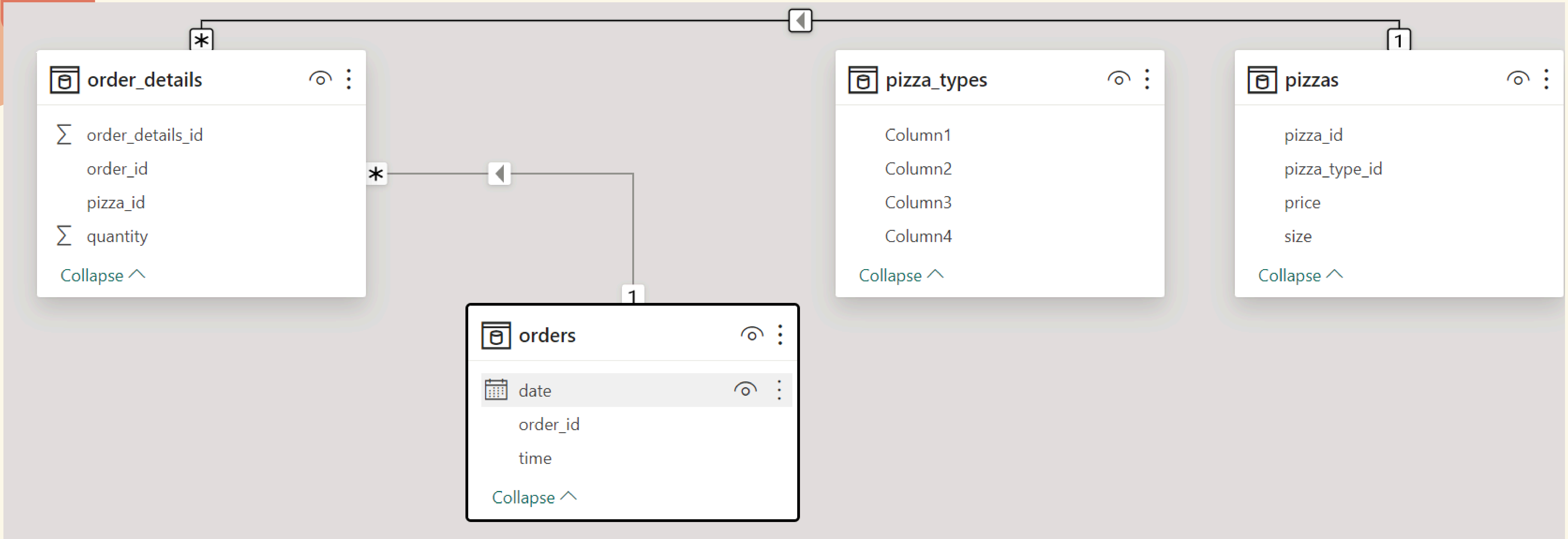
PIZZA SALES ANALYSIS

By: Ayush Kumar

BACKGROUND

This project showcases my expertise in SQL by addressing various problem statements using a pizza store dataset. By analyzing and querying the data, I derived meaningful insights that highlight customer preferences, sales trends, and operational efficiencies. The project demonstrates the practical application of SQL techniques such as data aggregation, filtering, joins, and subqueries to solve real-world business challenges for a pizza store.

TABLE VIEW



order_details
- ∑ order_details_id
- order_id
- pizza_id
- ∑ quantity

Collapse

pizza_types
- Column1
- Column2
- Column3
- Column4

Collapse

pizzas
- pizza_id
- pizza_type_id
- price
- size

Collapse

orders
- date
- order_id
- time

Collapse

PROJECT GOALS

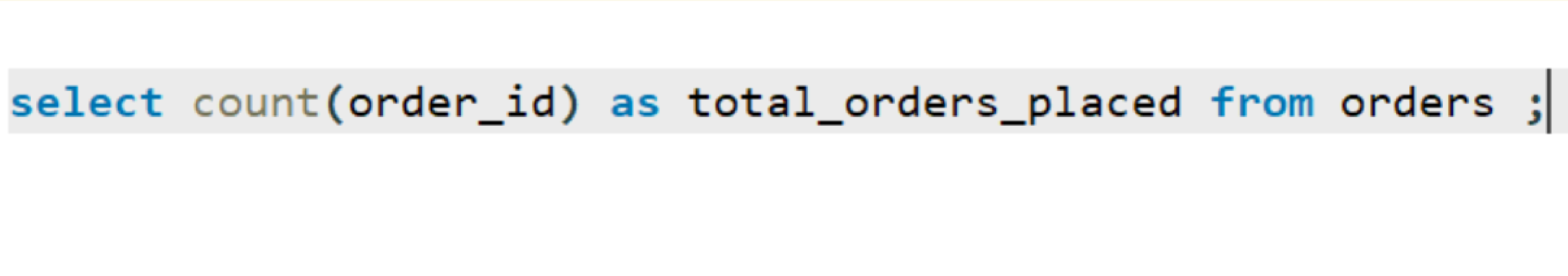
- QI > RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- Q2 > CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- Q3 > IDENTIFY THE HIGHEST-PRICED PIZZA.
- Q4 > IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- Q5 > LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

- Q6> JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- Q7> DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- Q8> JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- Q9> GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
- QIO> DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

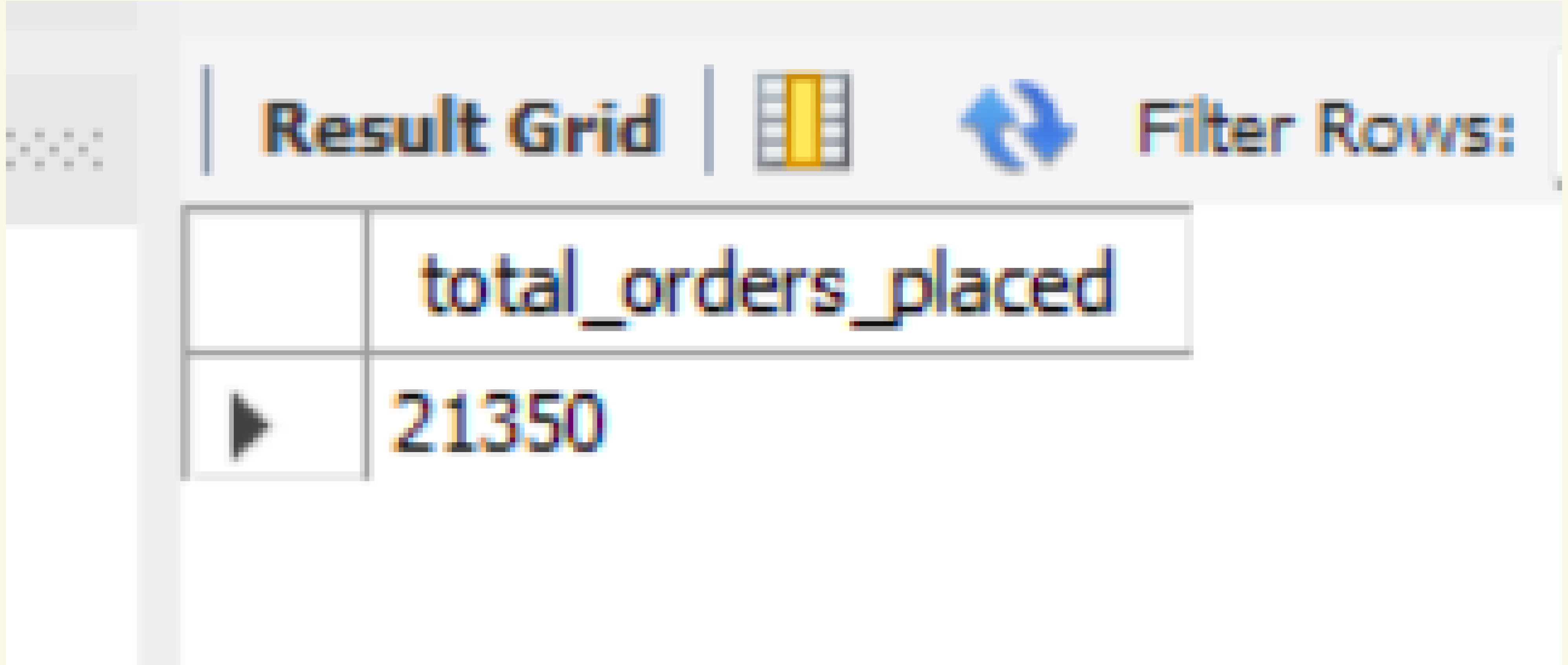
PROJECT GOALS

- Q11> CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
- Q12> ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- Q13> DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

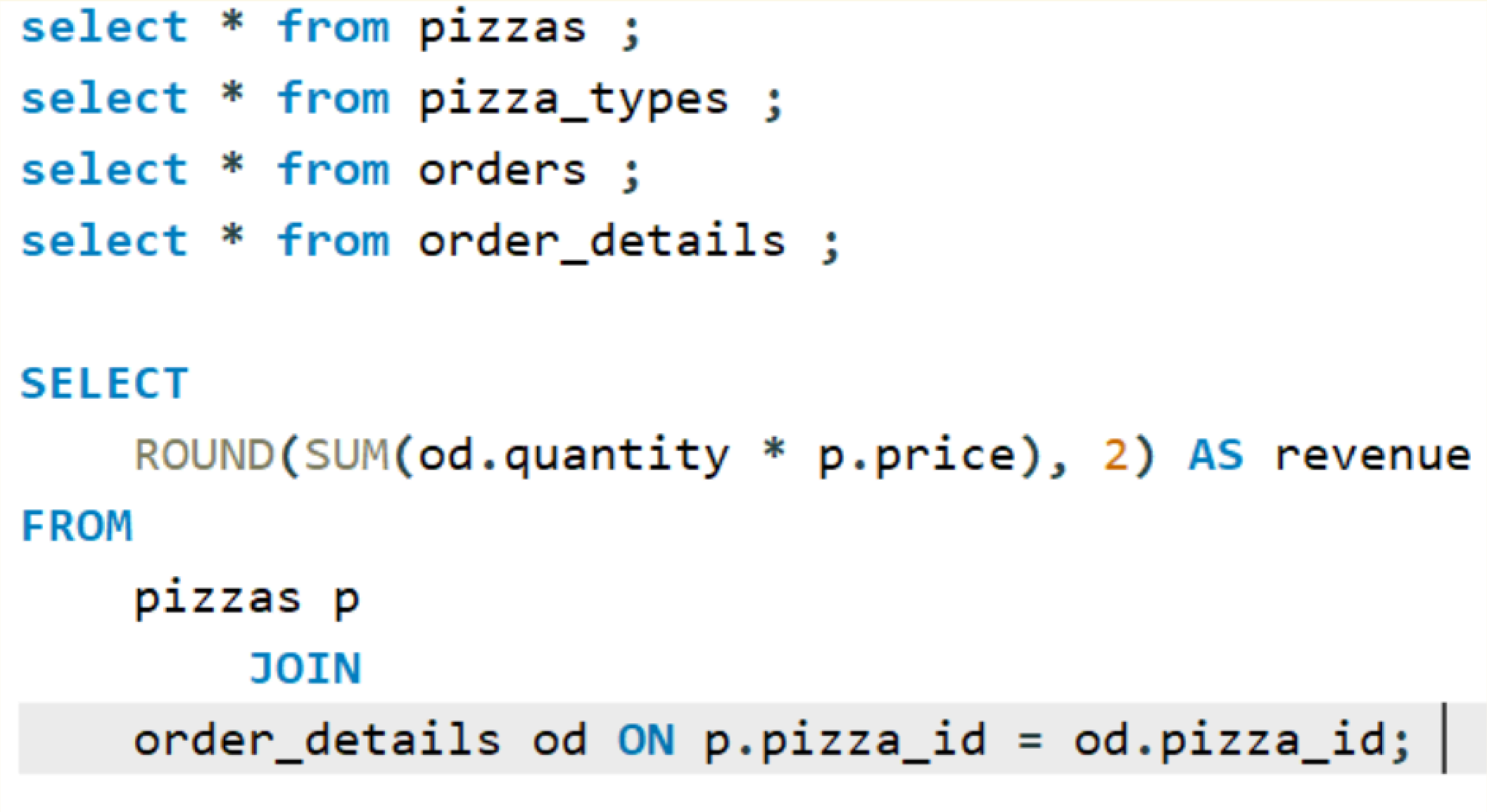


```
select count(order_id) as total_orders_placed from orders ;
```



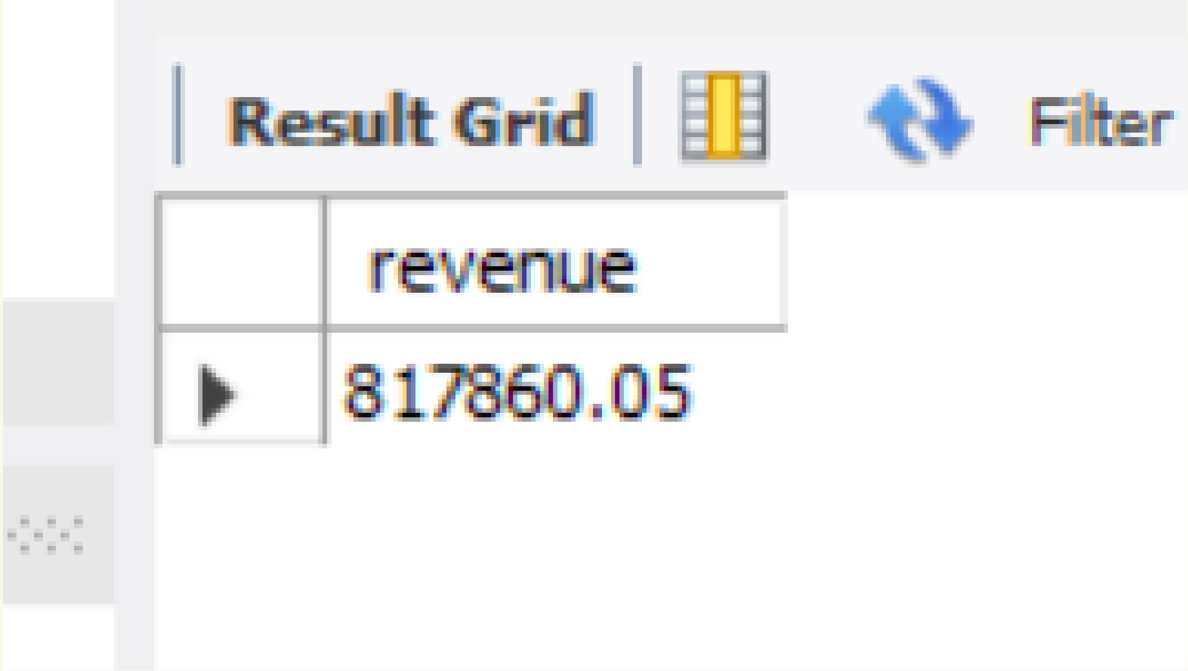
	total_orders_placed
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



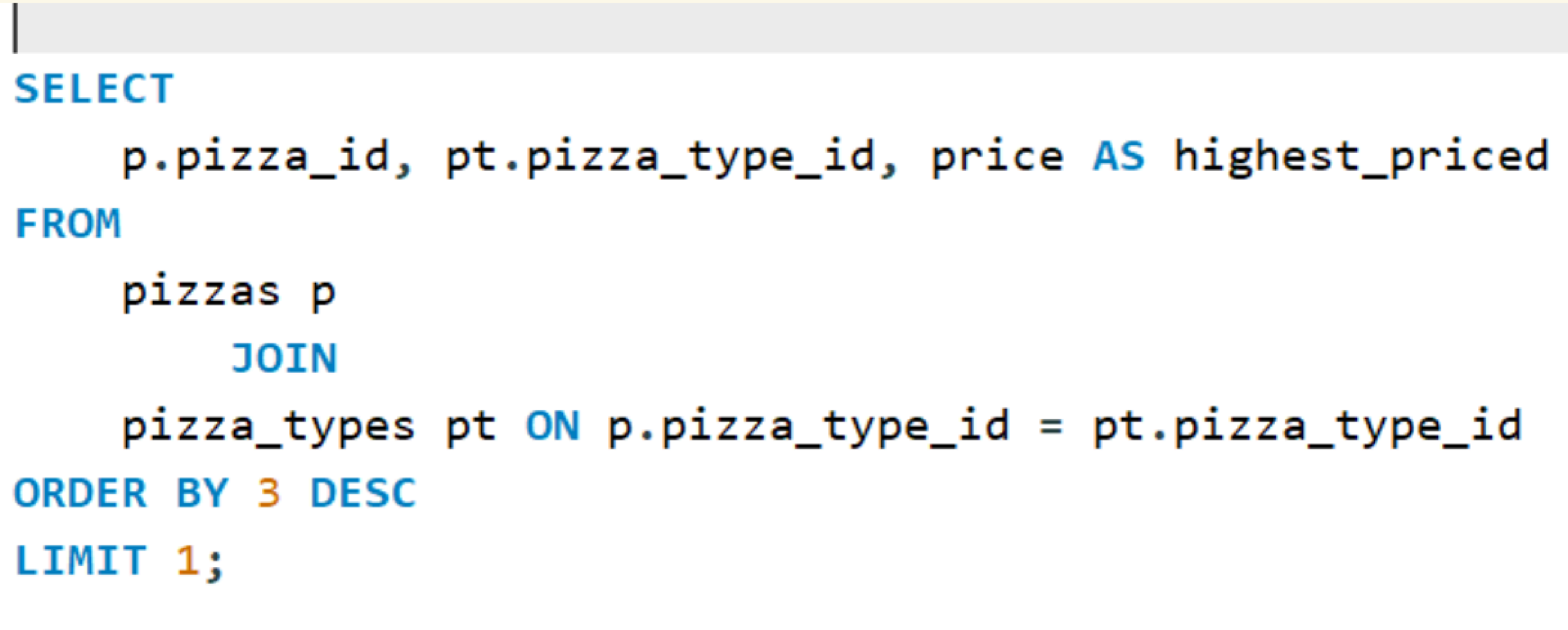
```
select * from pizzas ;
select * from pizza_types ;
select * from orders ;
select * from order_details ;

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

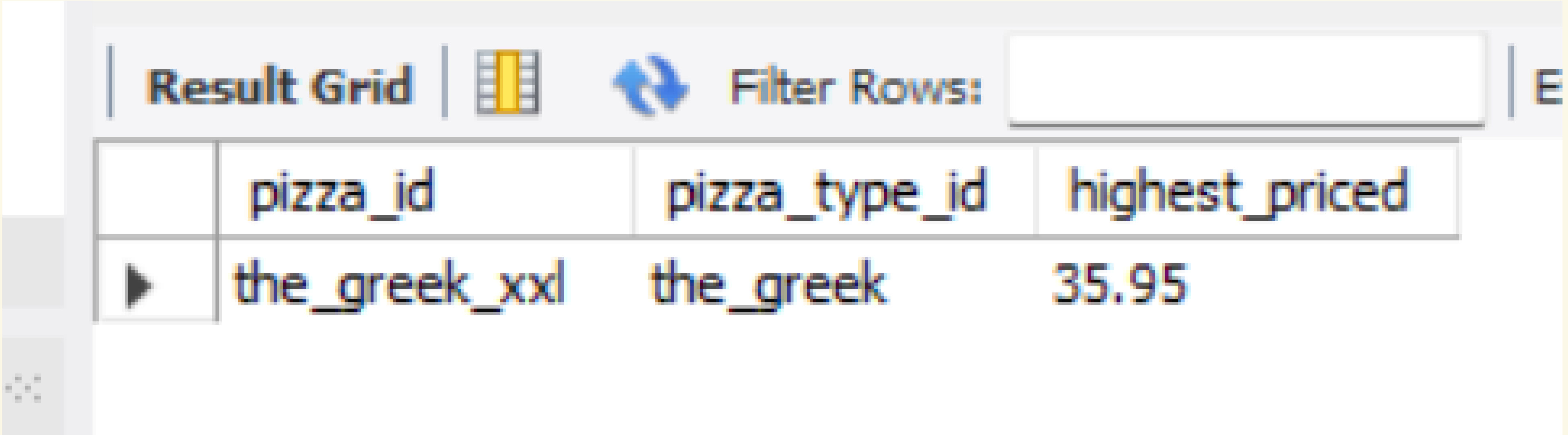


revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

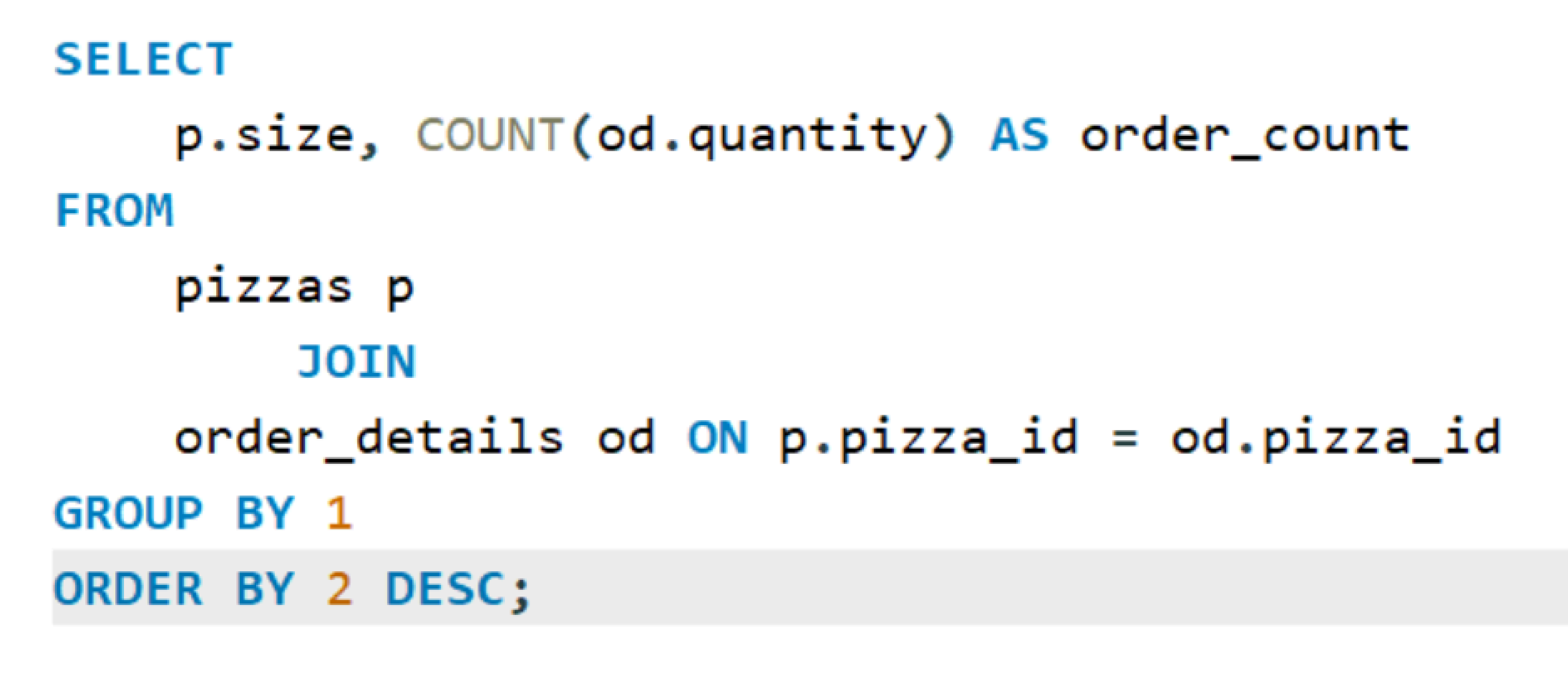


```
SELECT
    p.pizza_id, pt.pizza_type_id, price AS highest_priced
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY 3 DESC
LIMIT 1;
```

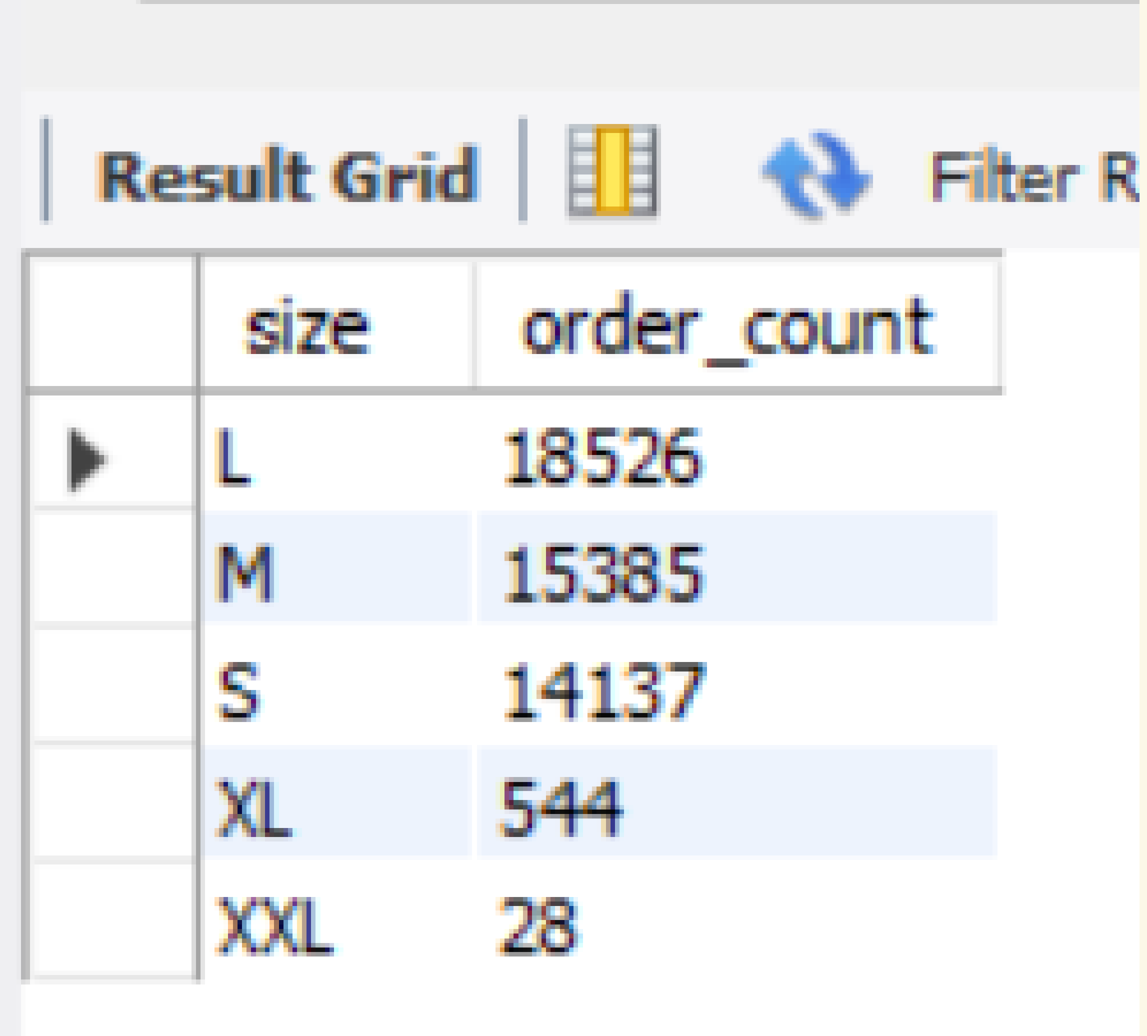


pizza_id	pizza_type_id	highest_priced
the_greek_xxl	the_greek	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

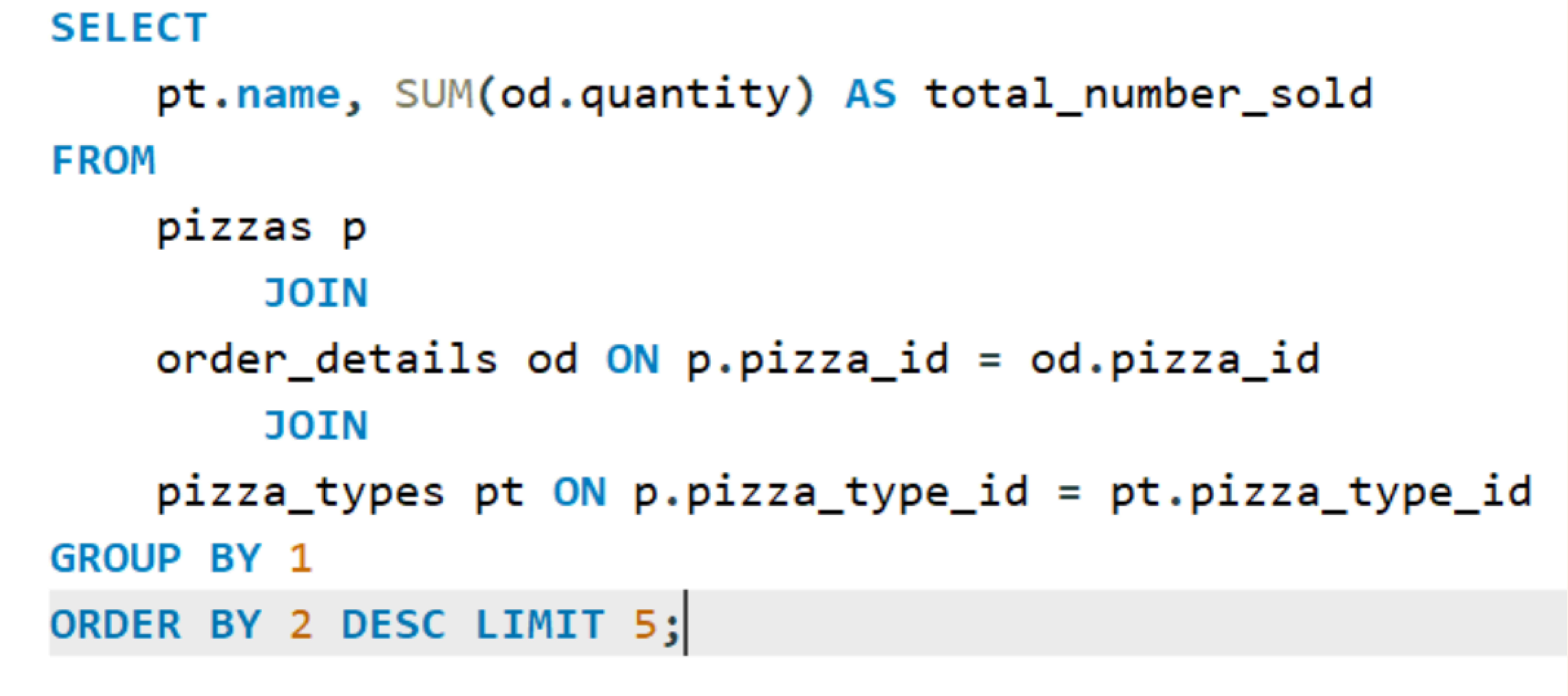


```
SELECT
    p.size, COUNT(od.quantity) AS order_count
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 1
ORDER BY 2 DESC;
```



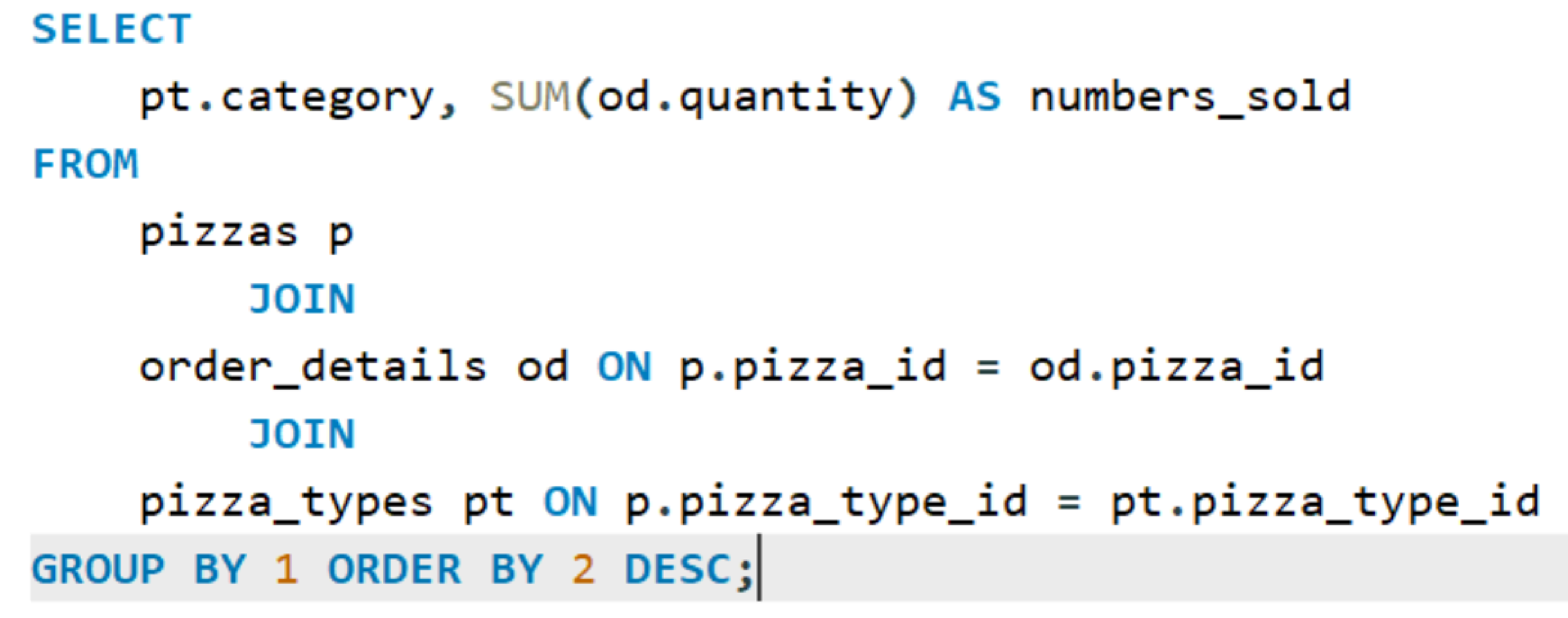
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

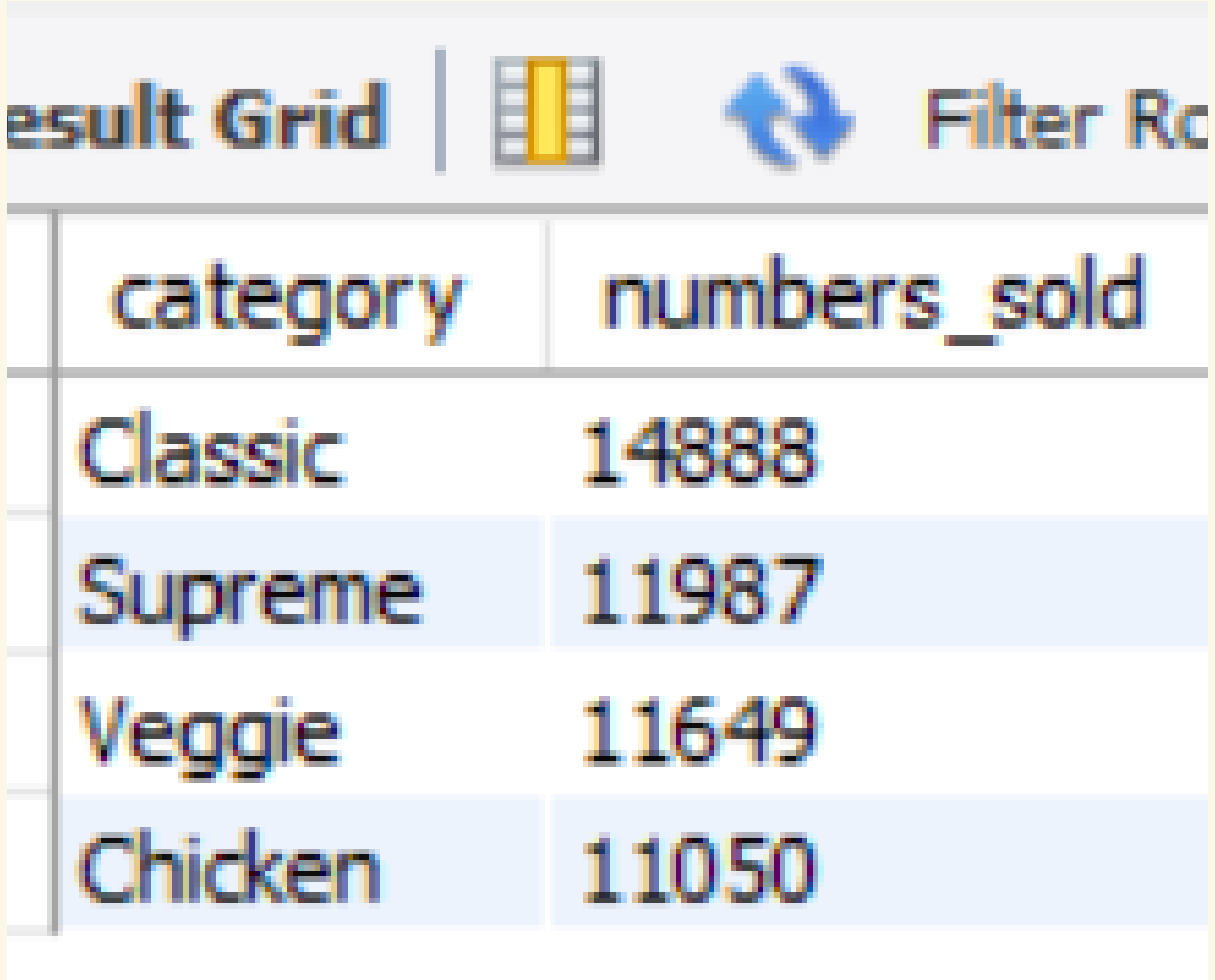


```
SELECT
    pt.name, SUM(od.quantity) AS total_number_sold
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY 1
ORDER BY 2 DESC LIMIT 5;
```

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

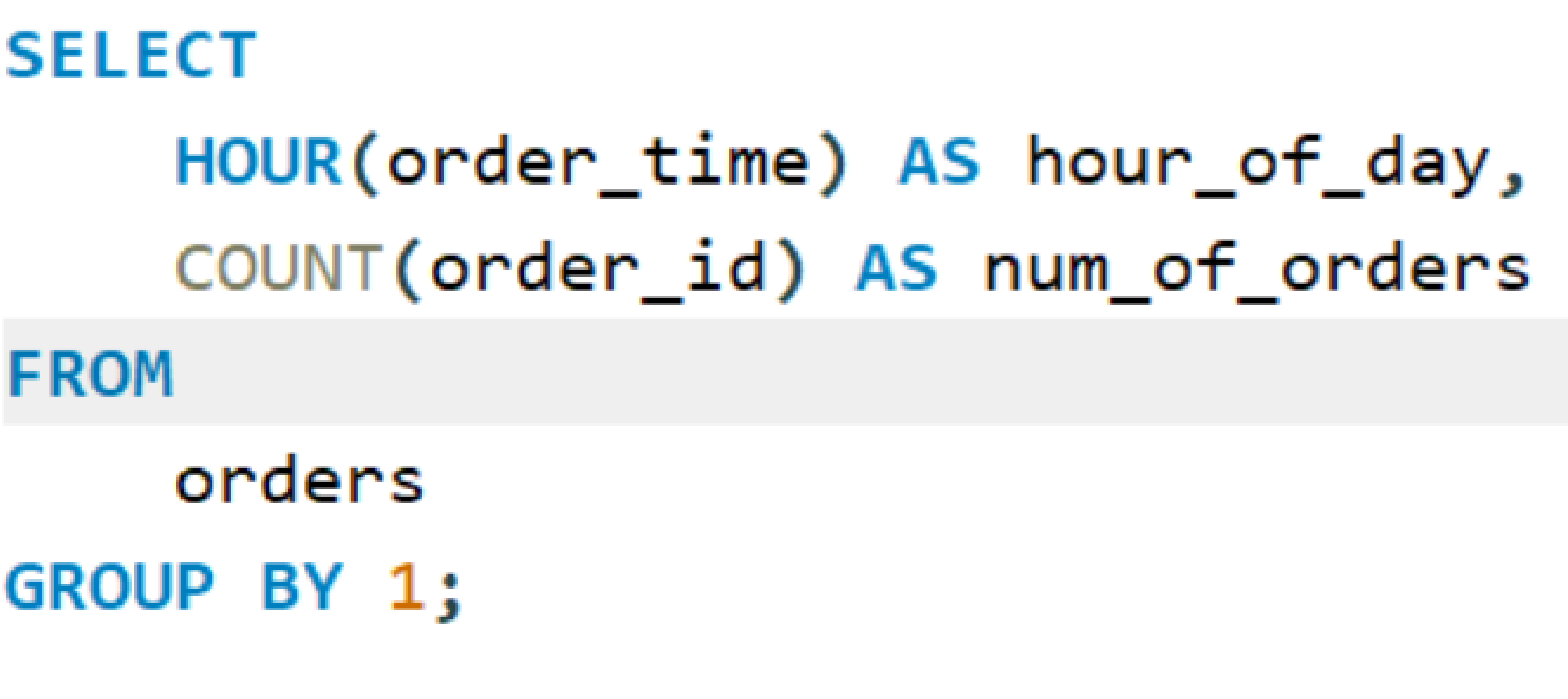


```
SELECT
    pt.category, SUM(od.quantity) AS numbers_sold
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY 1 ORDER BY 2 DESC;
```

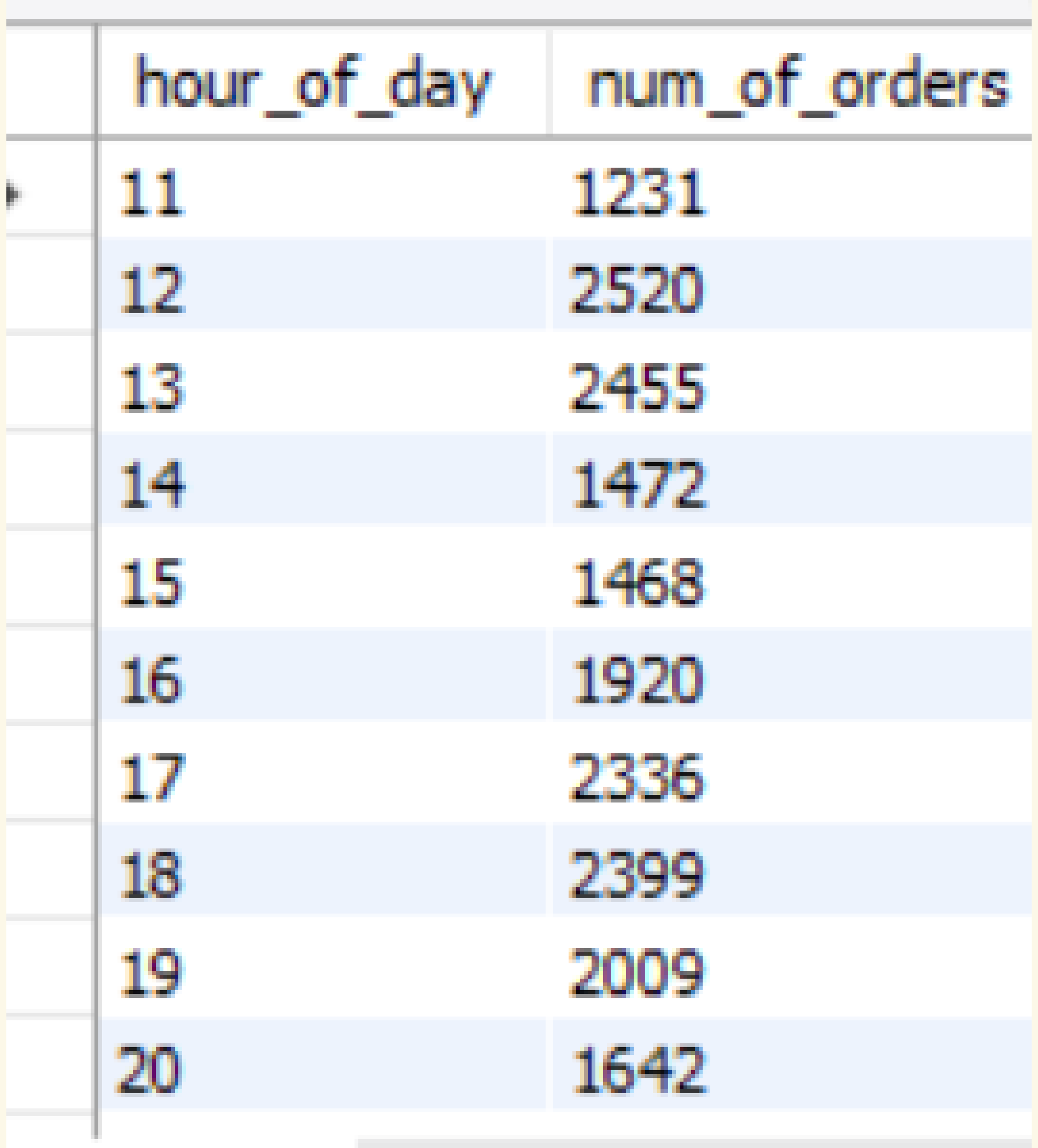


category	numbers_sold
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

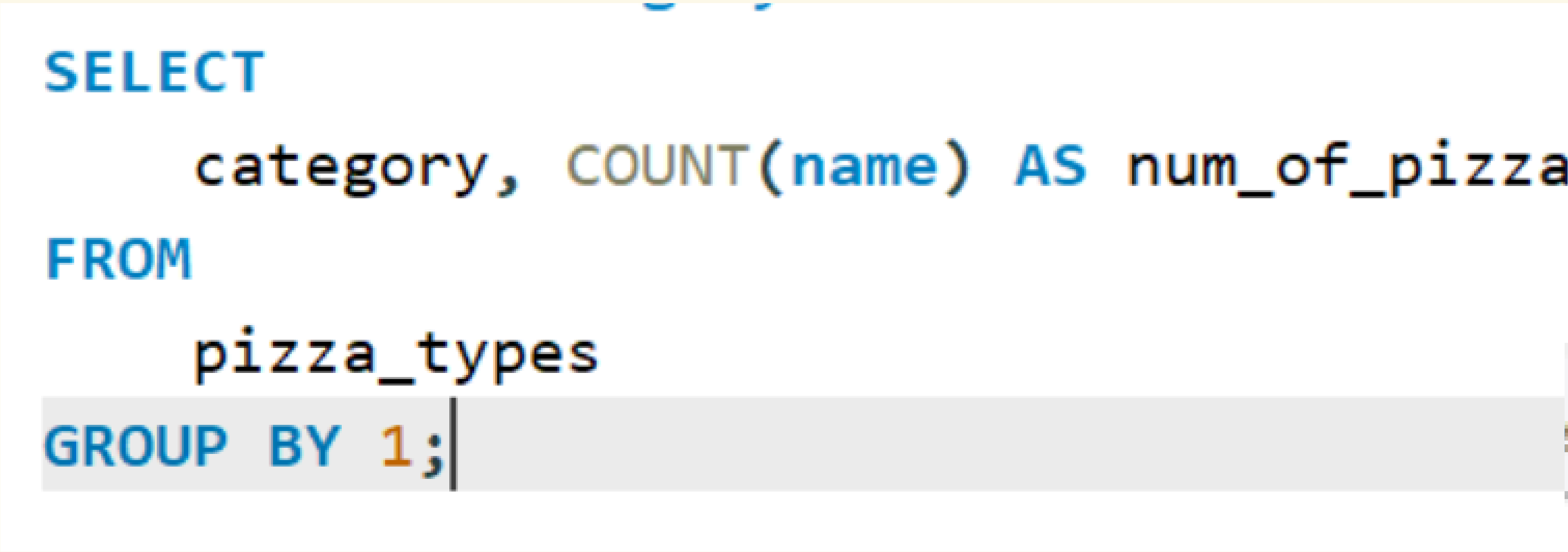


```
SELECT
    HOUR(order_time) AS hour_of_day,
    COUNT(order_id) AS num_of_orders
FROM
    orders
GROUP BY 1;
```

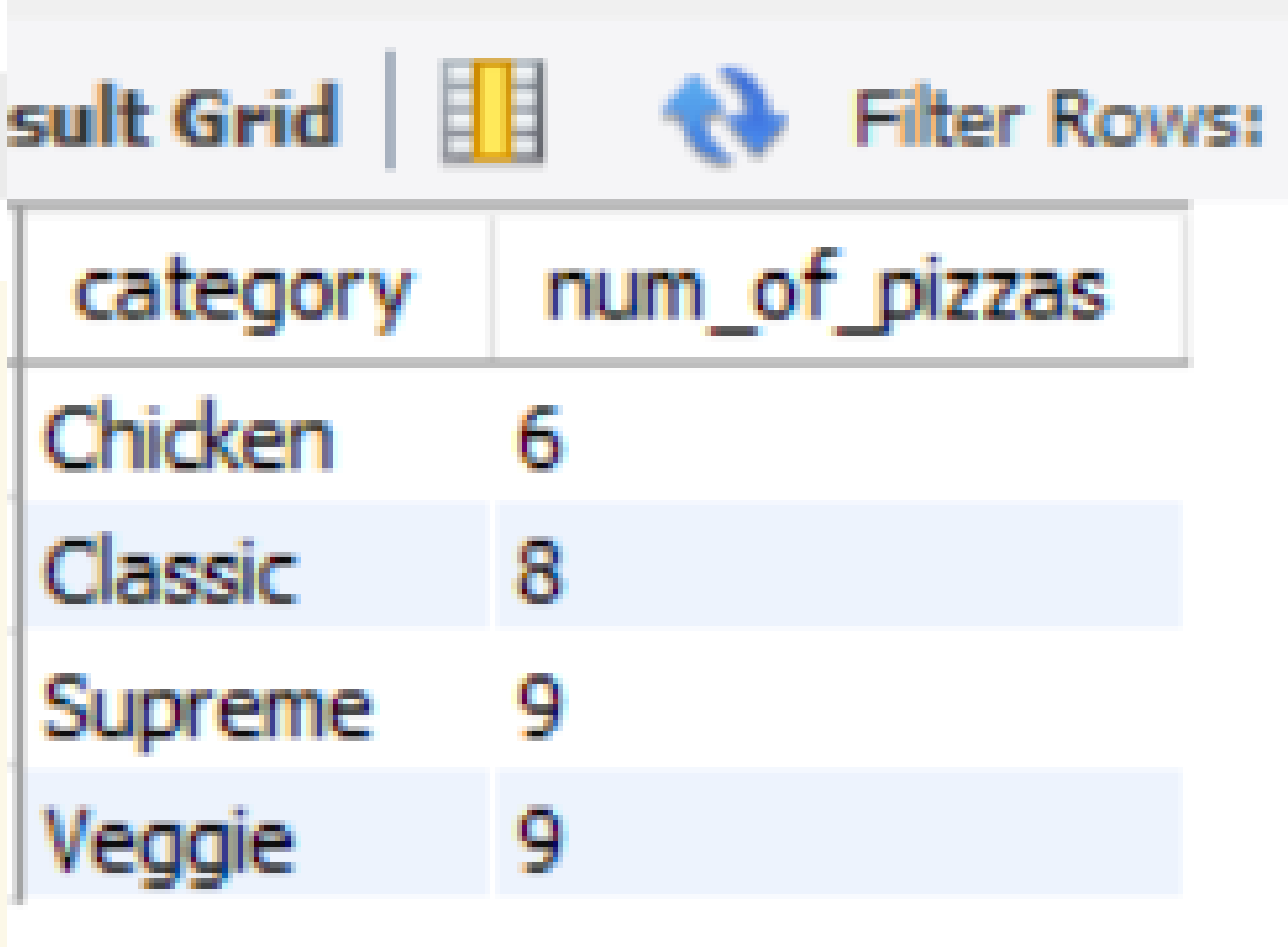


hour_of_day	num_of_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

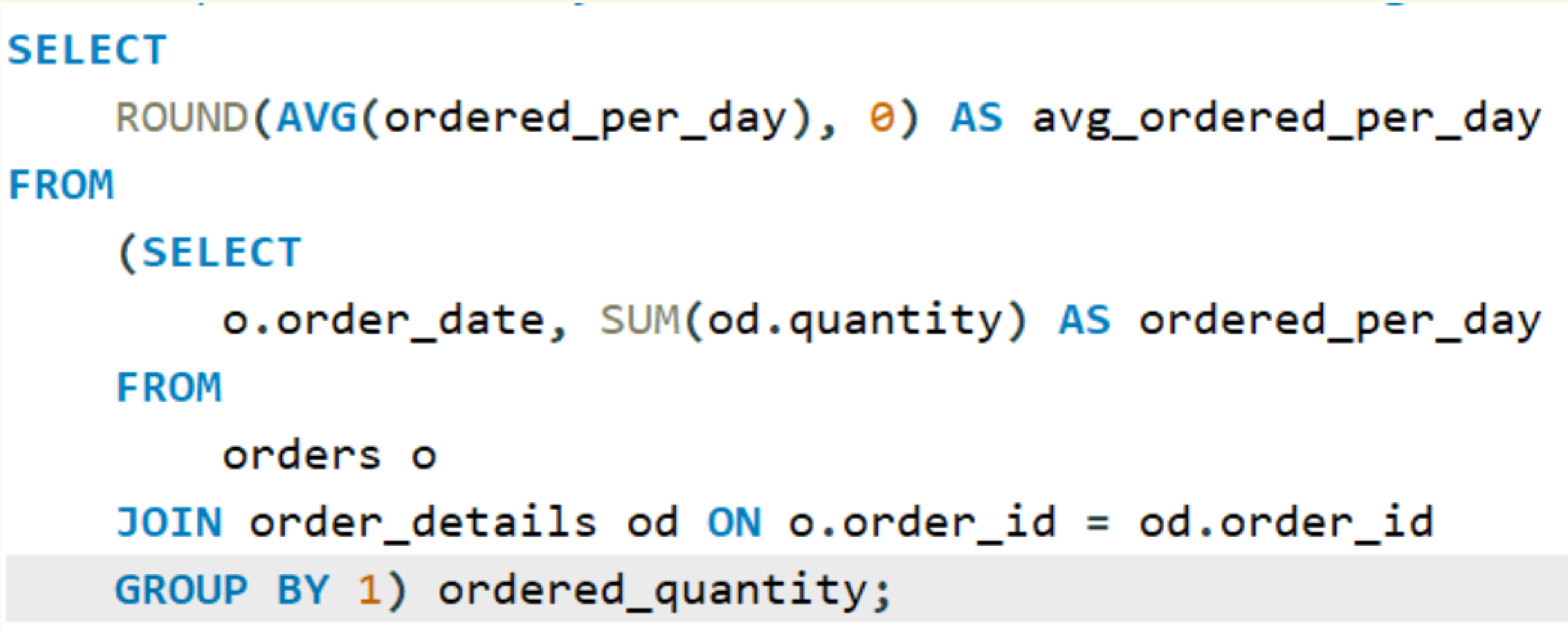


```sql
SELECT
    category, COUNT(name) AS num_of_pizzas
FROM
    pizza_types
GROUP BY 1;
```

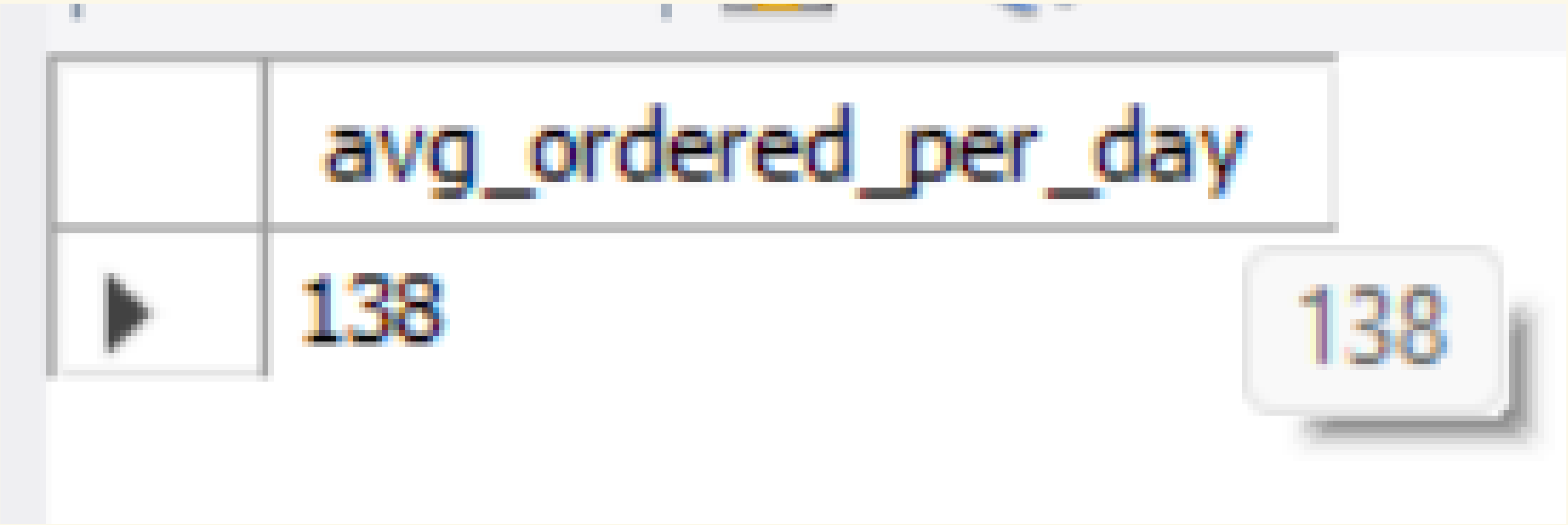


category	num_of_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

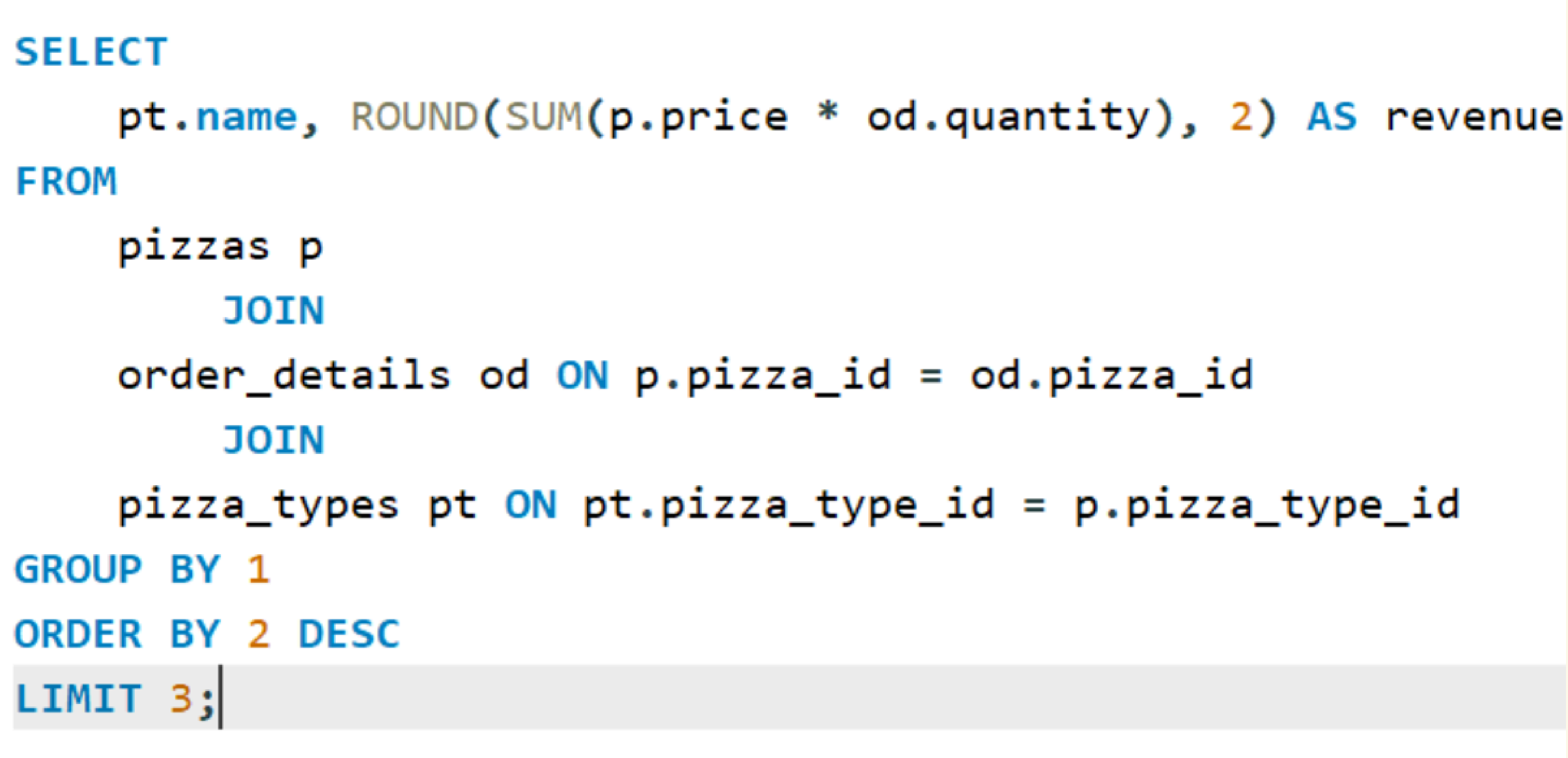


```sql
SELECT
    ROUND(AVG(ordered_per_day), 0) AS avg_ordered_per_day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS ordered_per_day
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY 1) ordered_quantity;
```

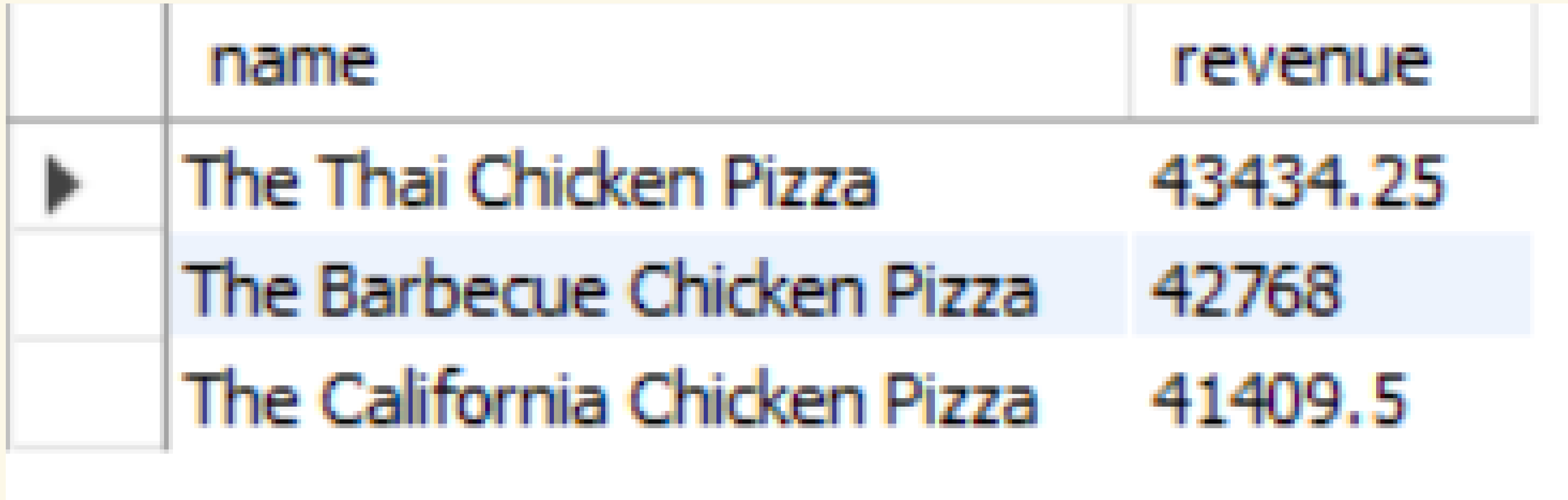


```
   avg_ordered_per_day
▶  138
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

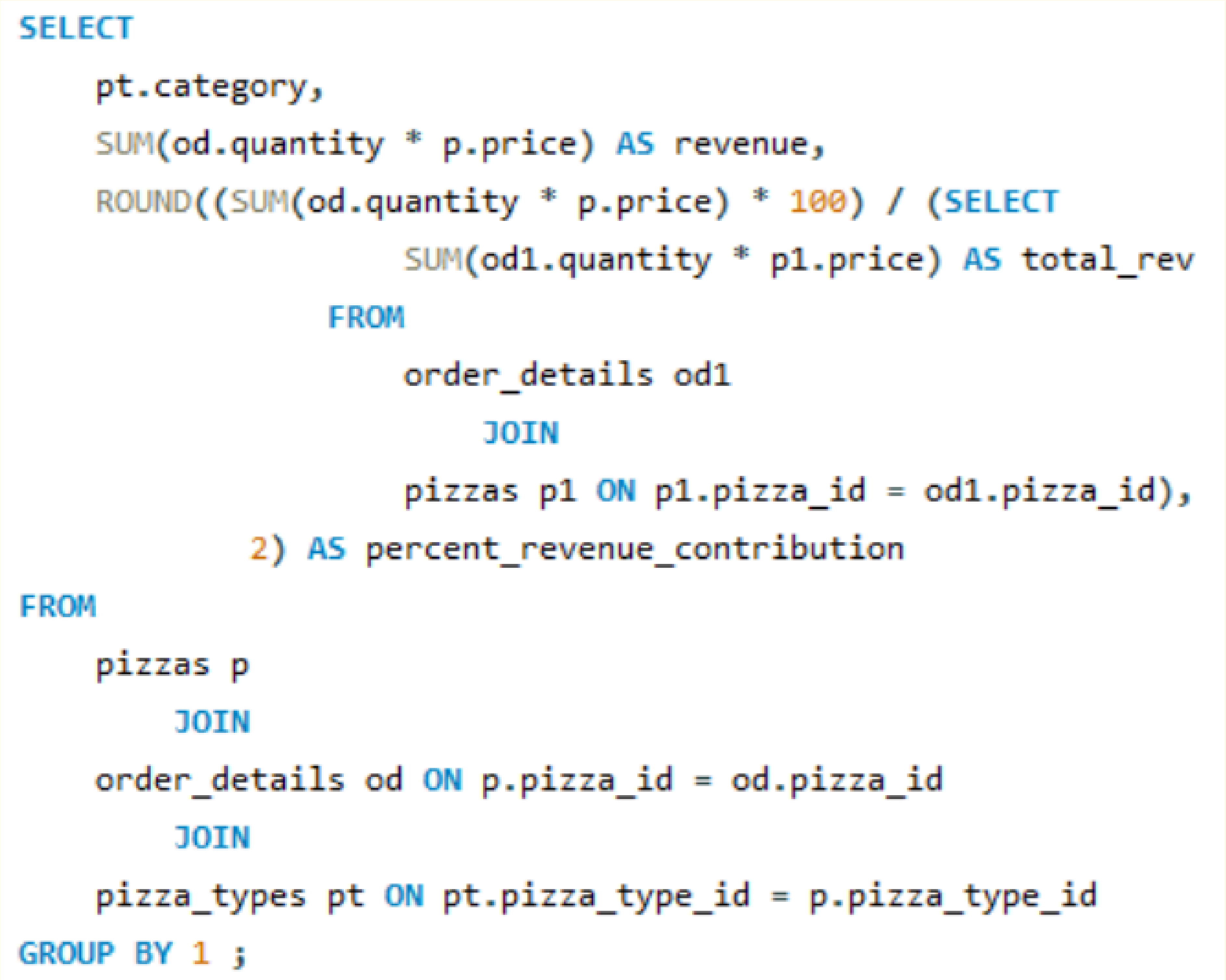


```sql
SELECT
    pt.name, ROUND(SUM(p.price * od.quantity), 2) AS revenue
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3;
```

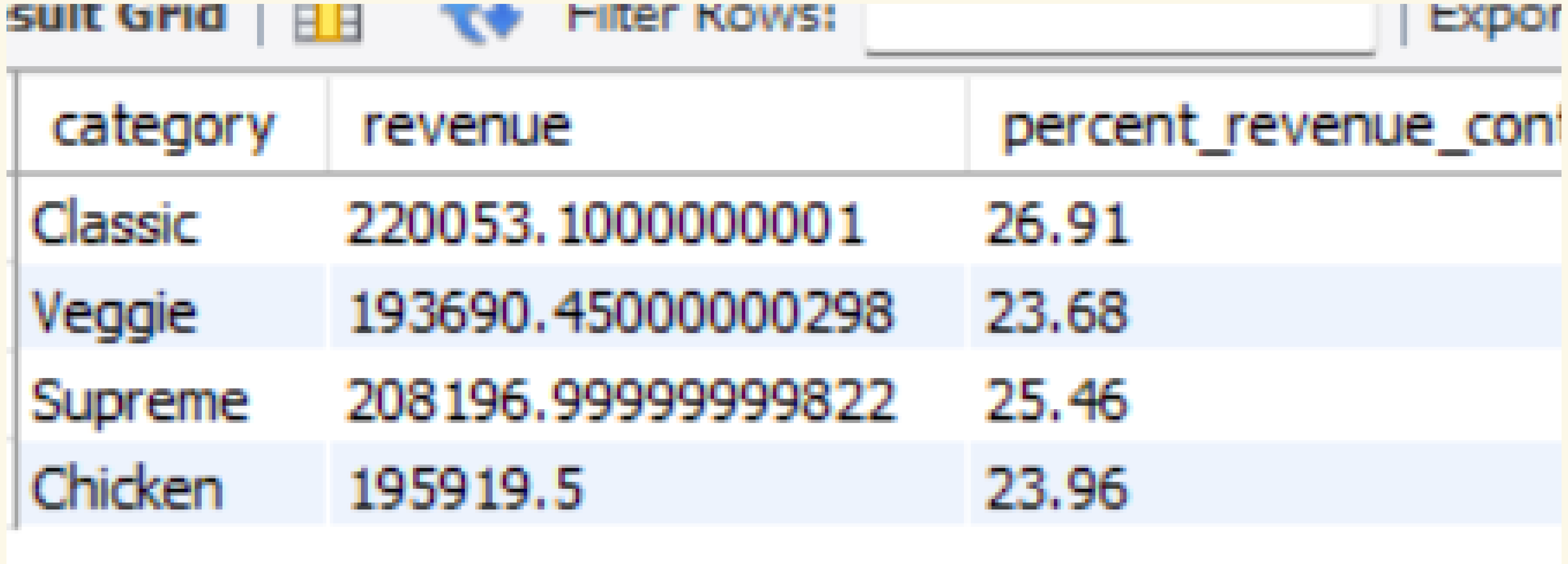


	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

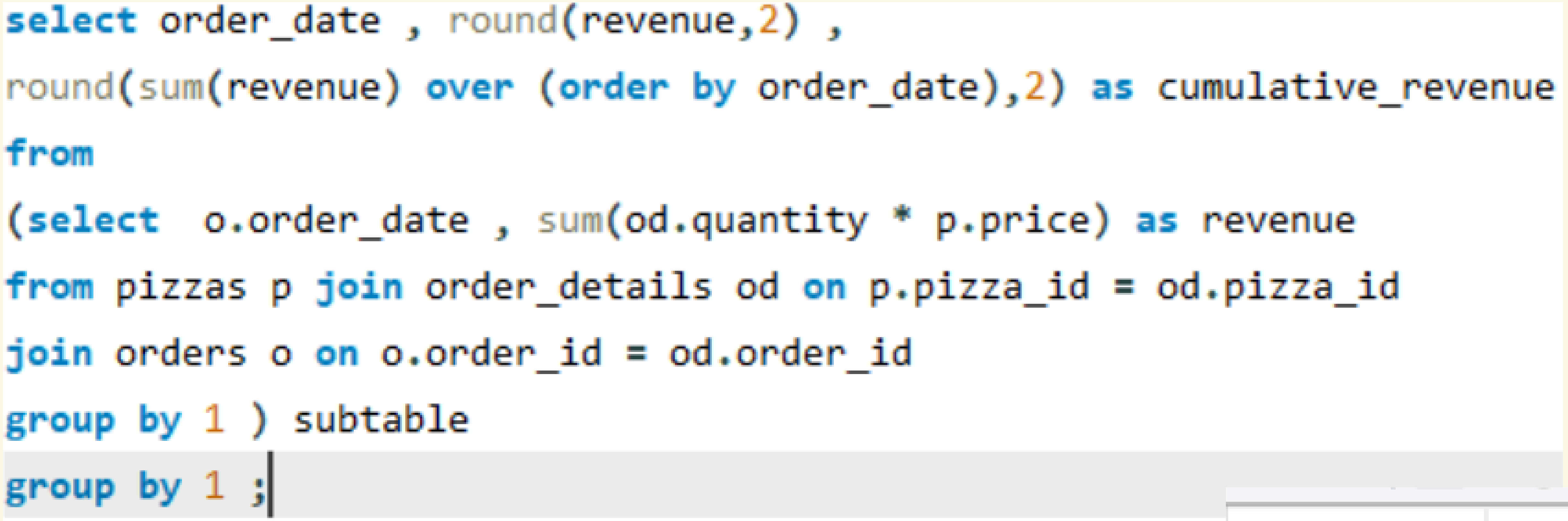


```
SELECT
    pt.category,
    SUM(od.quantity * p.price) AS revenue,
    ROUND((SUM(od.quantity * p.price) * 100) / (SELECT
                    SUM(od1.quantity * p1.price) AS total_rev
                FROM
                    order_details od1
                        JOIN
                    pizzas p1 ON p1.pizza_id = od1.pizza_id),
            2) AS percent_revenue_contribution
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY 1 ;
```

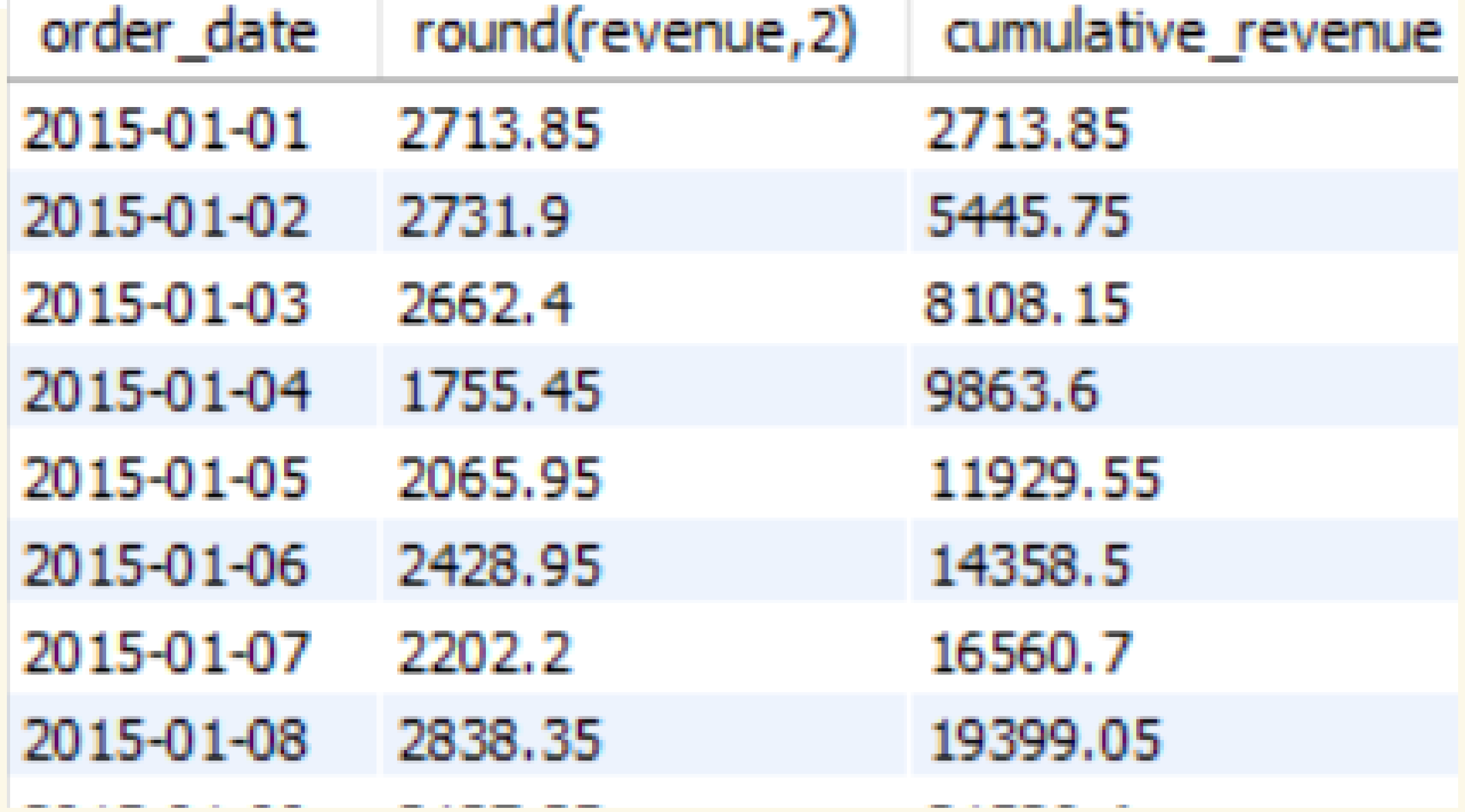


category	revenue	percent_revenue_cont
Classic	220053.100000001	26.91
Veggie	193690.45000000298	23.68
Supreme	208196.99999999822	25.46
Chicken	195919.5	23.96

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

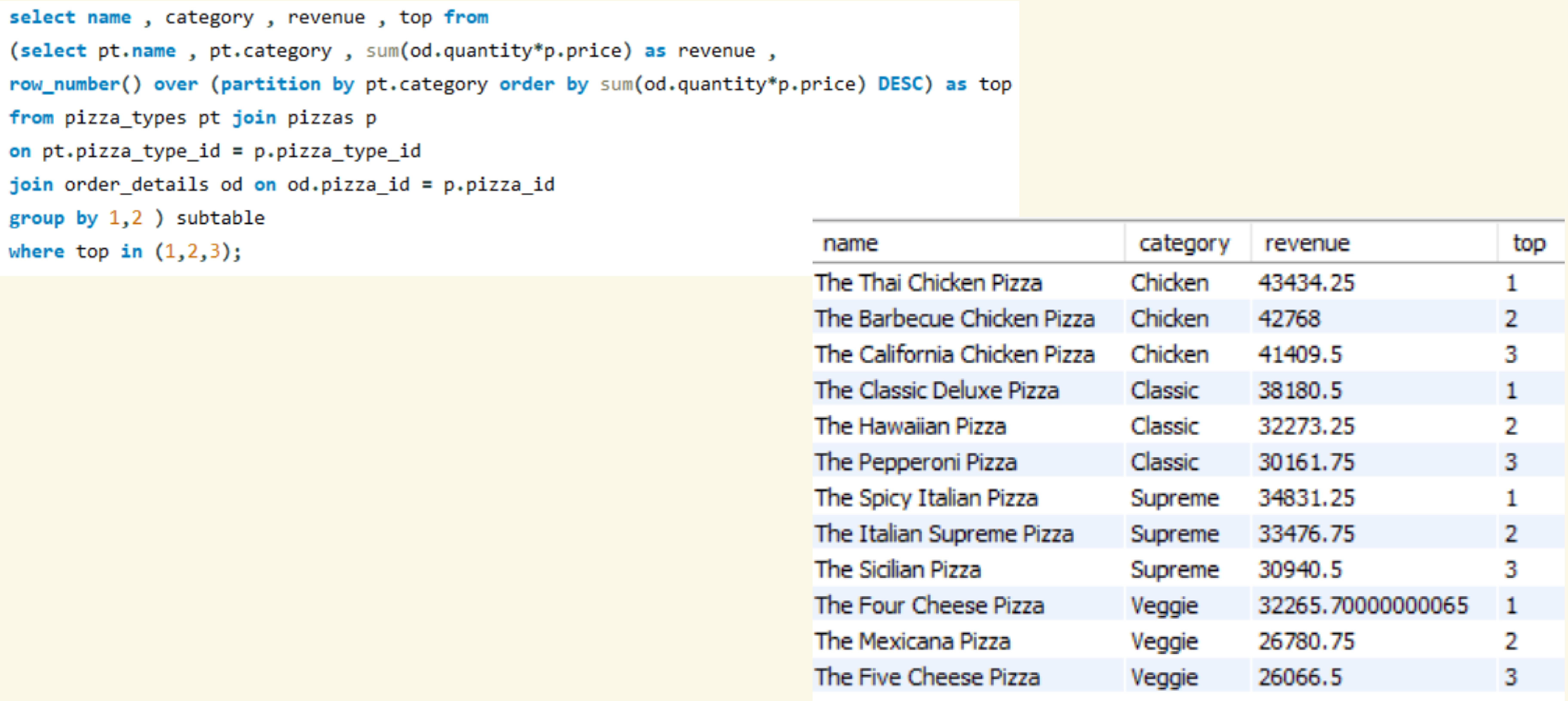


```
select order_date , round(revenue,2) ,
round(sum(revenue) over (order by order_date),2) as cumulative_revenue
from
(select  o.order_date , sum(od.quantity * p.price) as revenue
from pizzas p join order_details od on p.pizza_id = od.pizza_id
join orders o on o.order_id = od.order_id
group by 1 ) subtable
group by 1 ;
```



order_date	round(revenue,2)	cumulative_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY



```
select name , category , revenue , top from
(select pt.name , pt.category , sum(od.quantity*p.price) as revenue ,
row_number() over (partition by pt.category order by sum(od.quantity*p.price) DESC) as top
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od on od.pizza_id = p.pizza_id
group by 1,2 ) subtable
where top in (1,2,3);
```

name	category	revenue	top
The Thai Chicken Pizza	Chicken	43434.25	1
The Barbecue Chicken Pizza	Chicken	42768	2
The California Chicken Pizza	Chicken	41409.5	3
The Classic Deluxe Pizza	Classic	38180.5	1
The Hawaiian Pizza	Classic	32273.25	2
The Pepperoni Pizza	Classic	30161.75	3
The Spicy Italian Pizza	Supreme	34831.25	1
The Italian Supreme Pizza	Supreme	33476.75	2
The Sicilian Pizza	Supreme	30940.5	3
The Four Cheese Pizza	Veggie	32265.7000000065	1
The Mexicana Pizza	Veggie	26780.75	2
The Five Cheese Pizza	Veggie	26066.5	3

THANK YOU