PIZZA SALES ANALYSIS REPORT

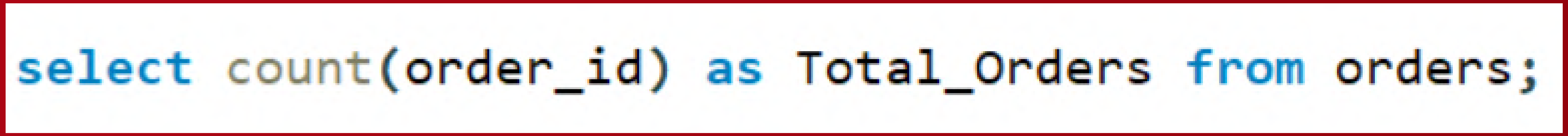
INTRODUCTION

I AM VIKAS SINGH, A DATA ANALYST WITH HANDS-ON EXPERIENCE IN SQL-BASED DATA ANALYSIS. I RECENTLY WORKED ON A PIZZA SALES ANALYSIS PROJECT TO DERIVE BUSINESS INSIGHTS USING SQL QUERIES, JOINS, AGGREGATIONS, AND WINDOWS FUNC.

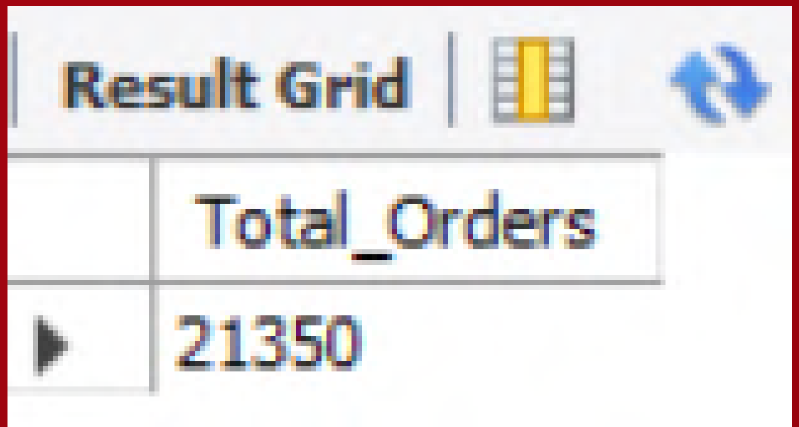
BUSINESS QUESTIONS SOLVED USING SQL

1.Retrieve the total number of orders placed.
2.Calculate the total revenue generated from pizza sales.
3.Identify the highest-priced pizza.
4.Identify the most common pizza size ordered.
5.List the top 5 most ordered pizza types along with their quantities.
6.Join the necessary tables to find the total quantity of each pizza category ordered.
7.Determine the distribution of orders by hour of the day.
8.Join relevant tables to find the category-wise distribution of pizzas.
9.Group the orders by date and calculate the average number of pizzas ordered per day.
10.Determine the top 3 most ordered pizza types based on revenue.
11.Calculate the percentage contribution of each pizza type to total revenue.
12.Analyze the cumulative revenue generated over time.
13.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

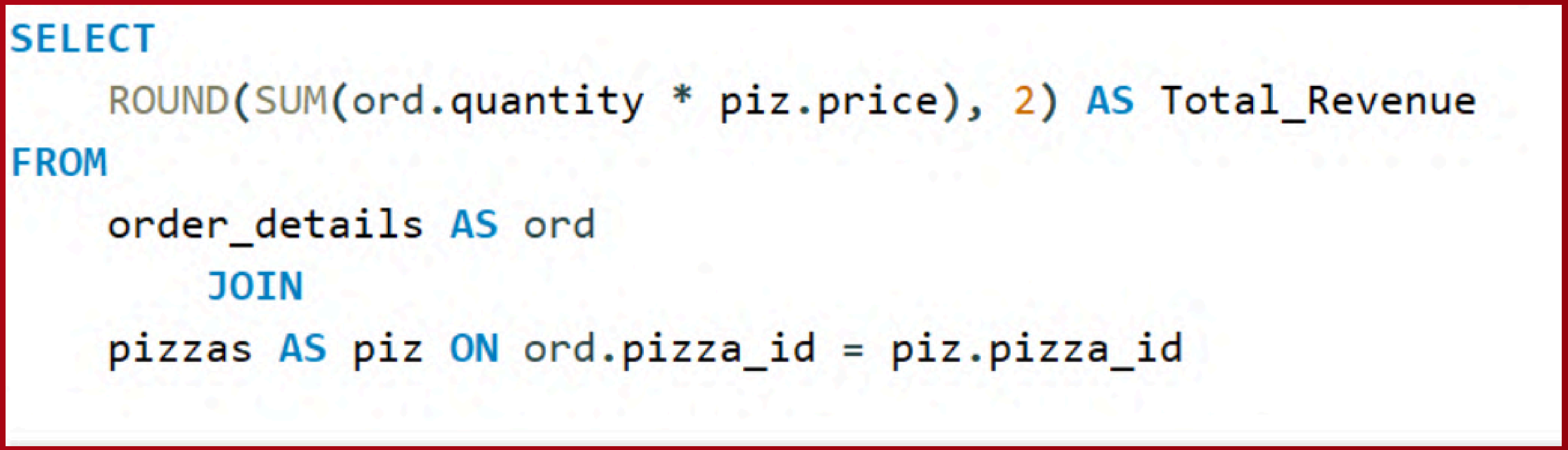


```
select count(order_id) as Total_Orders from orders;
```

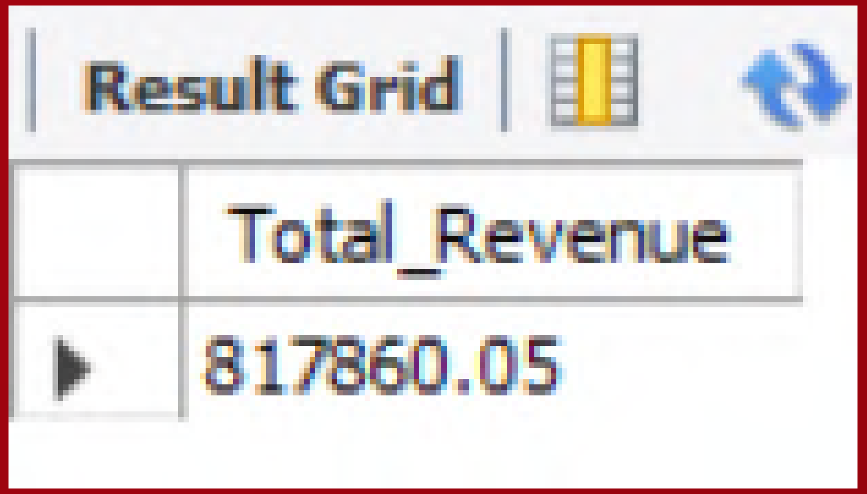


```
  Total_Orders
▶ 21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALE

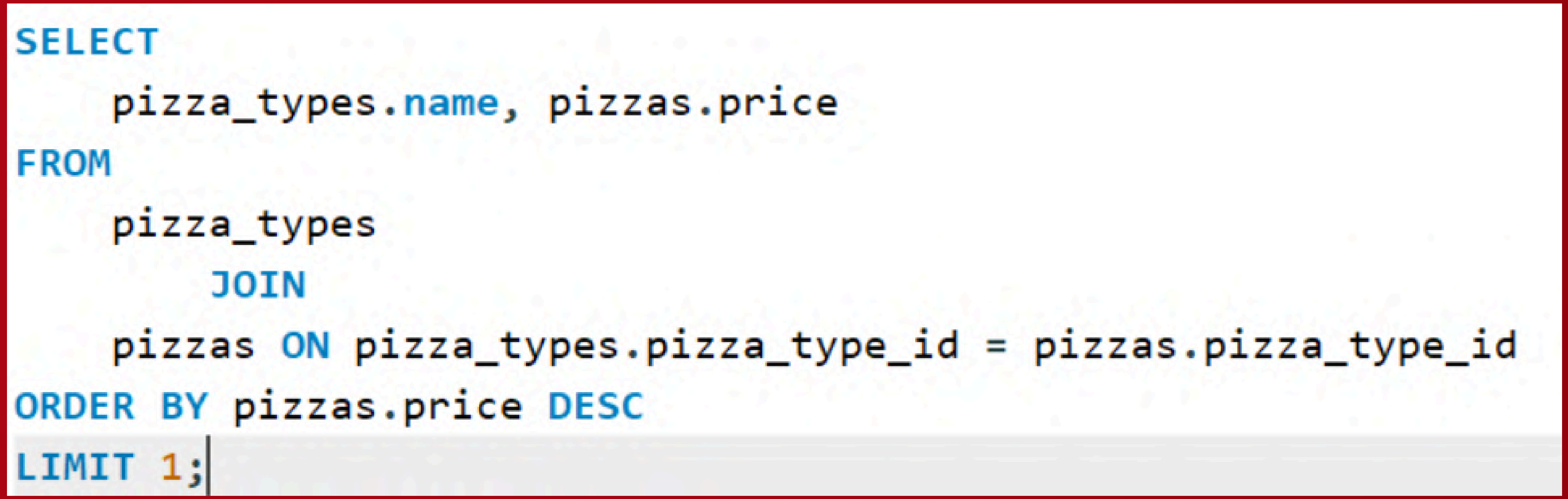


```
SELECT
    ROUND(SUM(ord.quantity * piz.price), 2) AS Total_Revenue
FROM
    order_details AS ord
        JOIN
    pizzas AS piz ON ord.pizza_id = piz.pizza_id
```

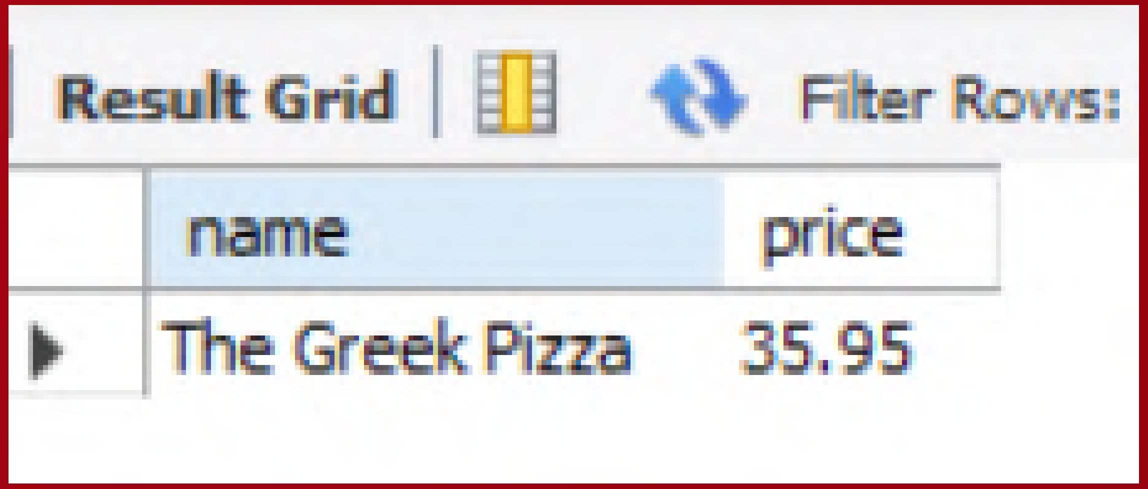


Total_Revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

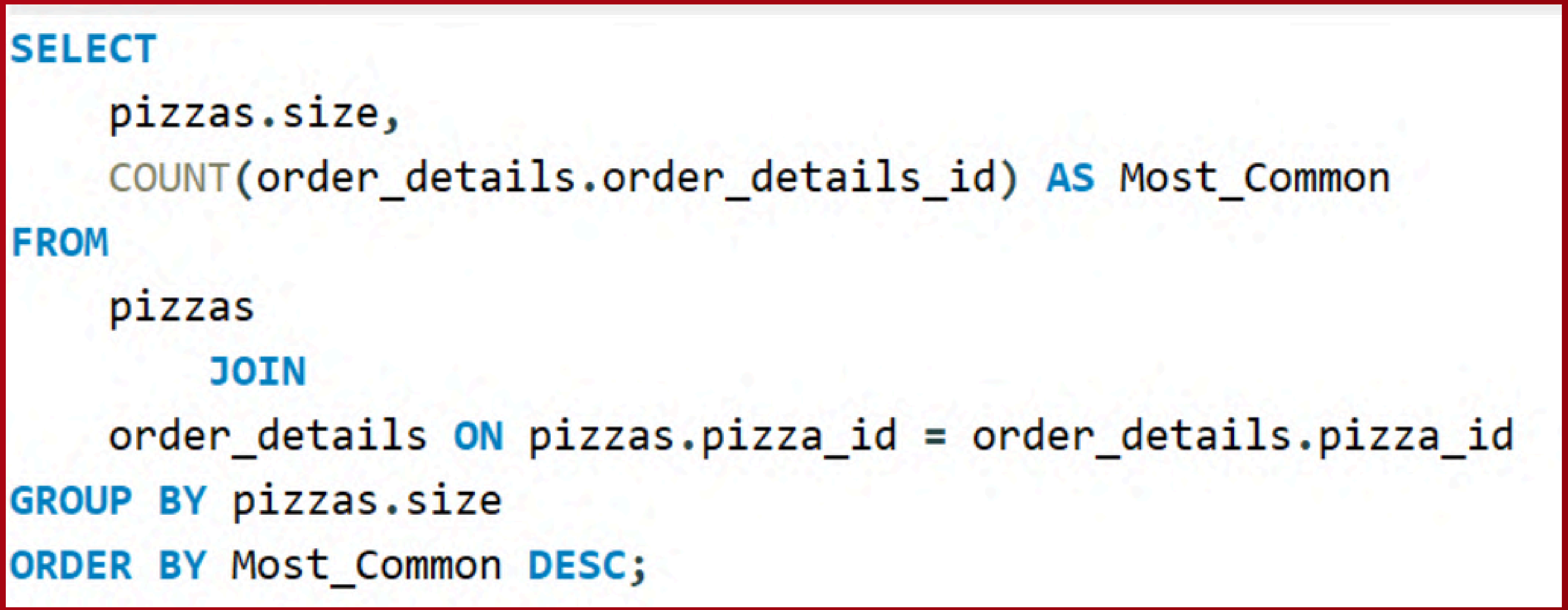


```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

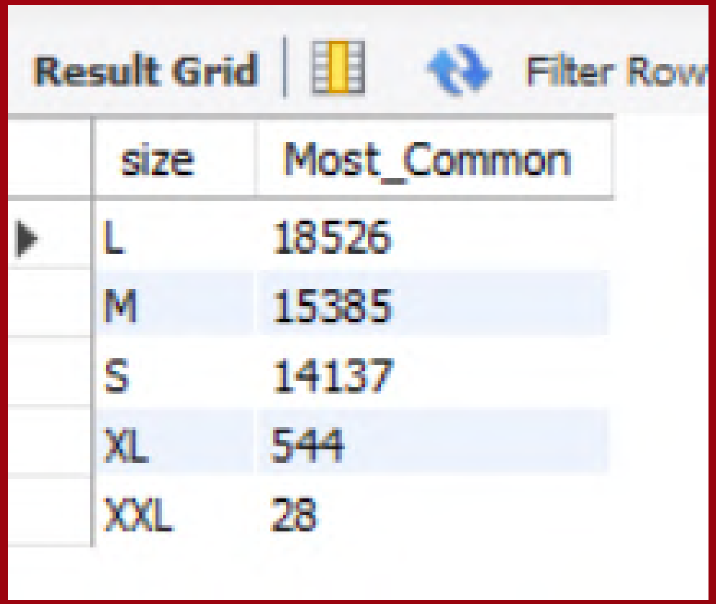


name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

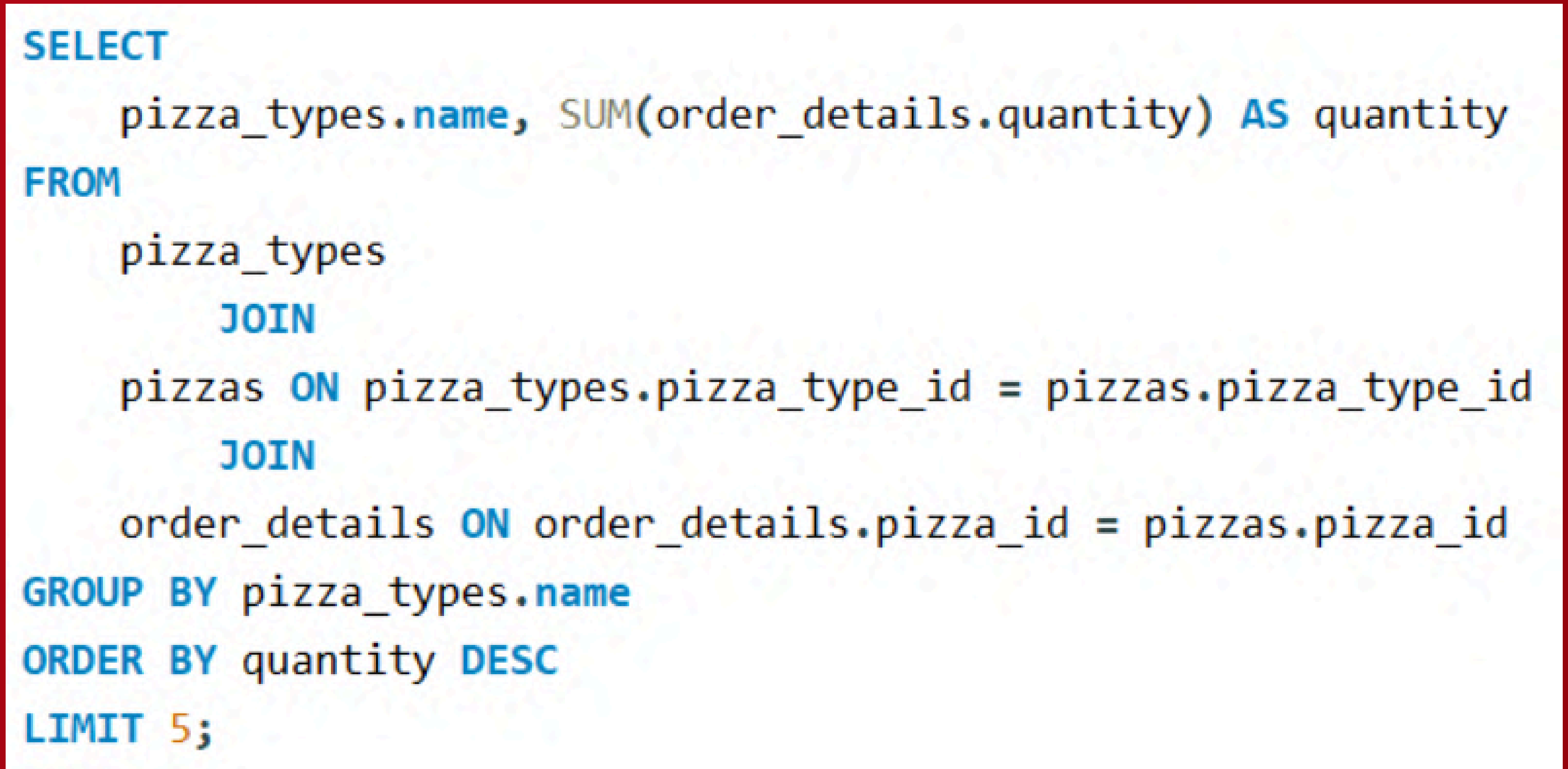


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Most_Common
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Most_Common DESC;
```

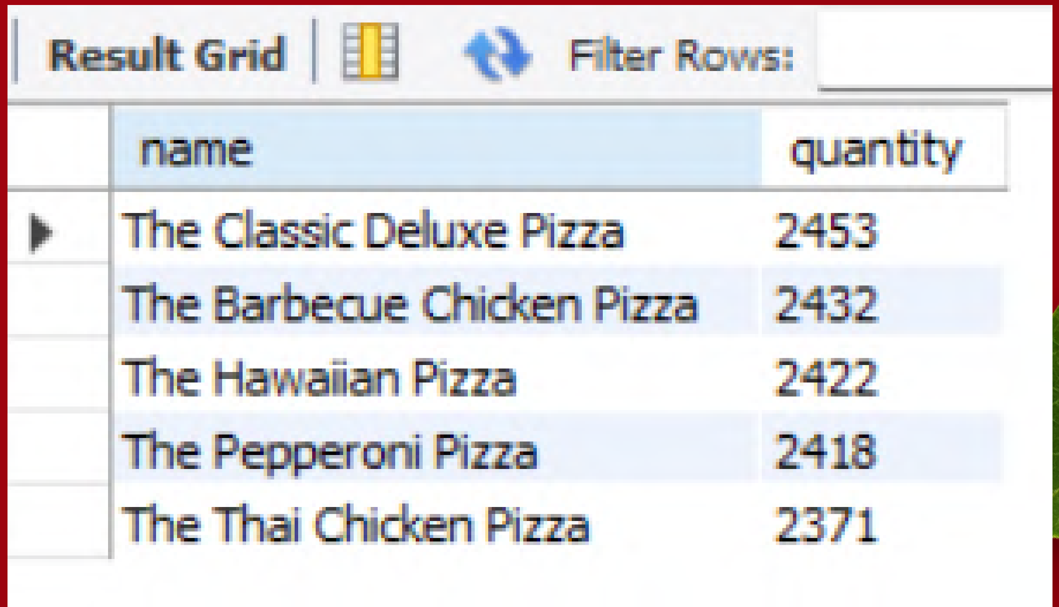


size	Most_Common
L	18526
M	15385
S	14137
XL	544
XXL	28

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

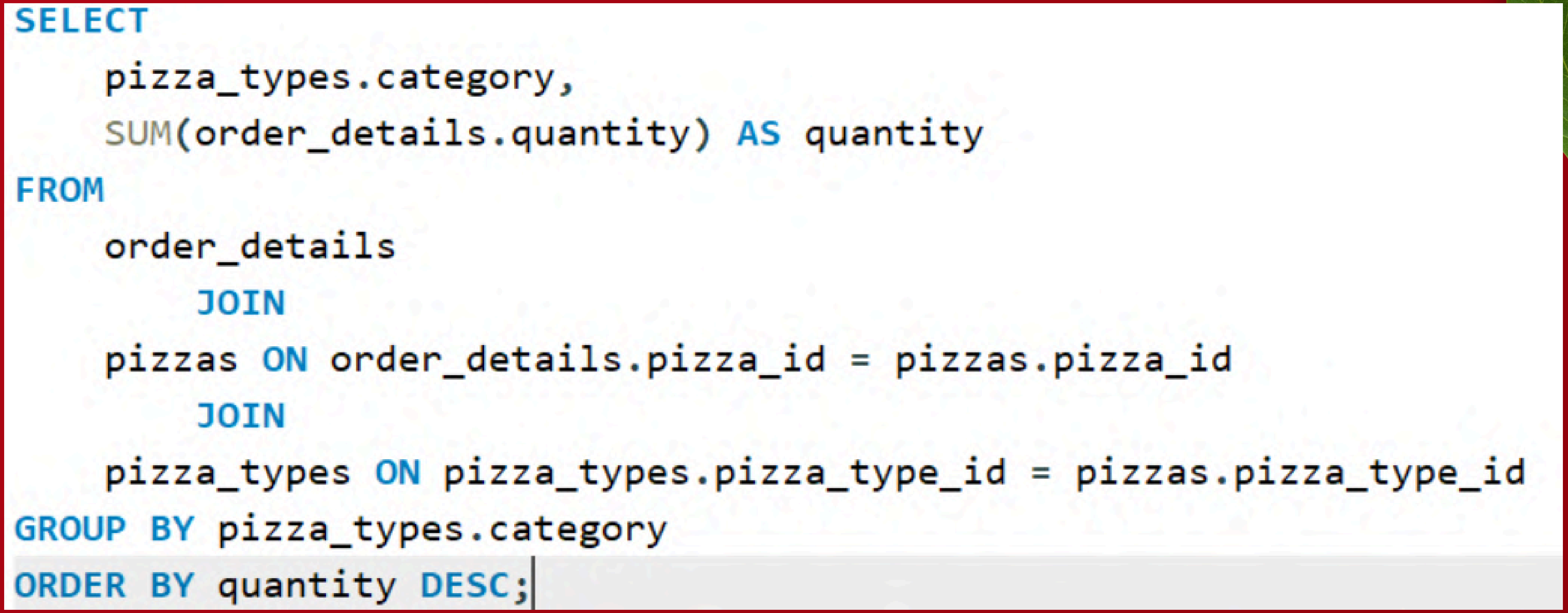


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

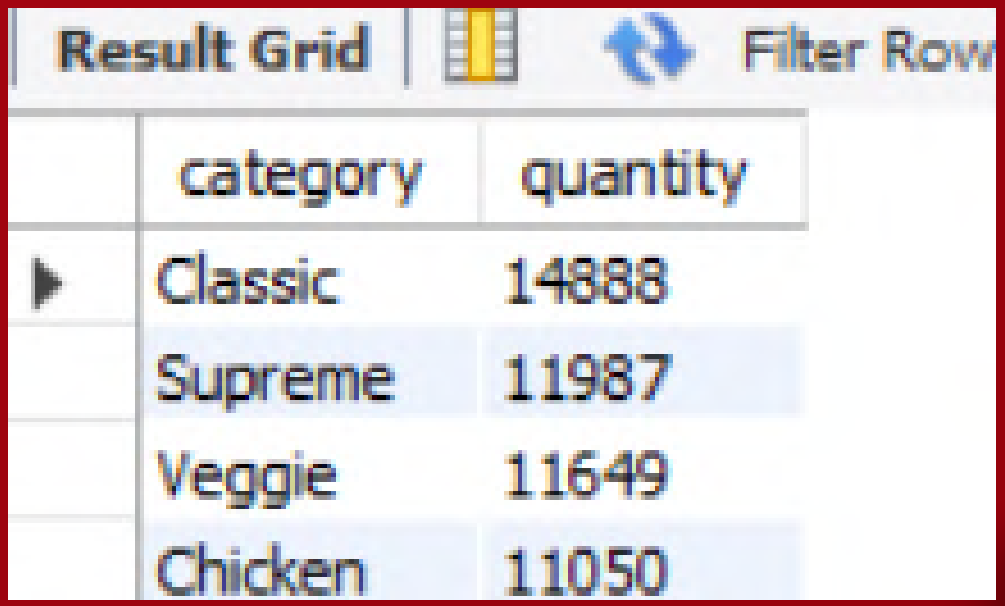


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

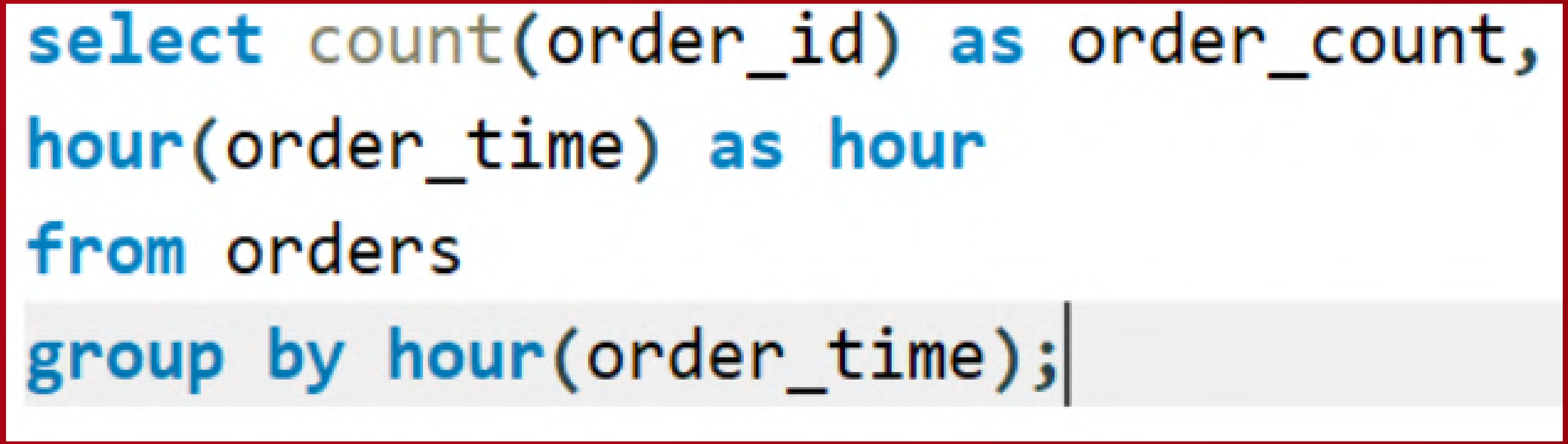


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY quantity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

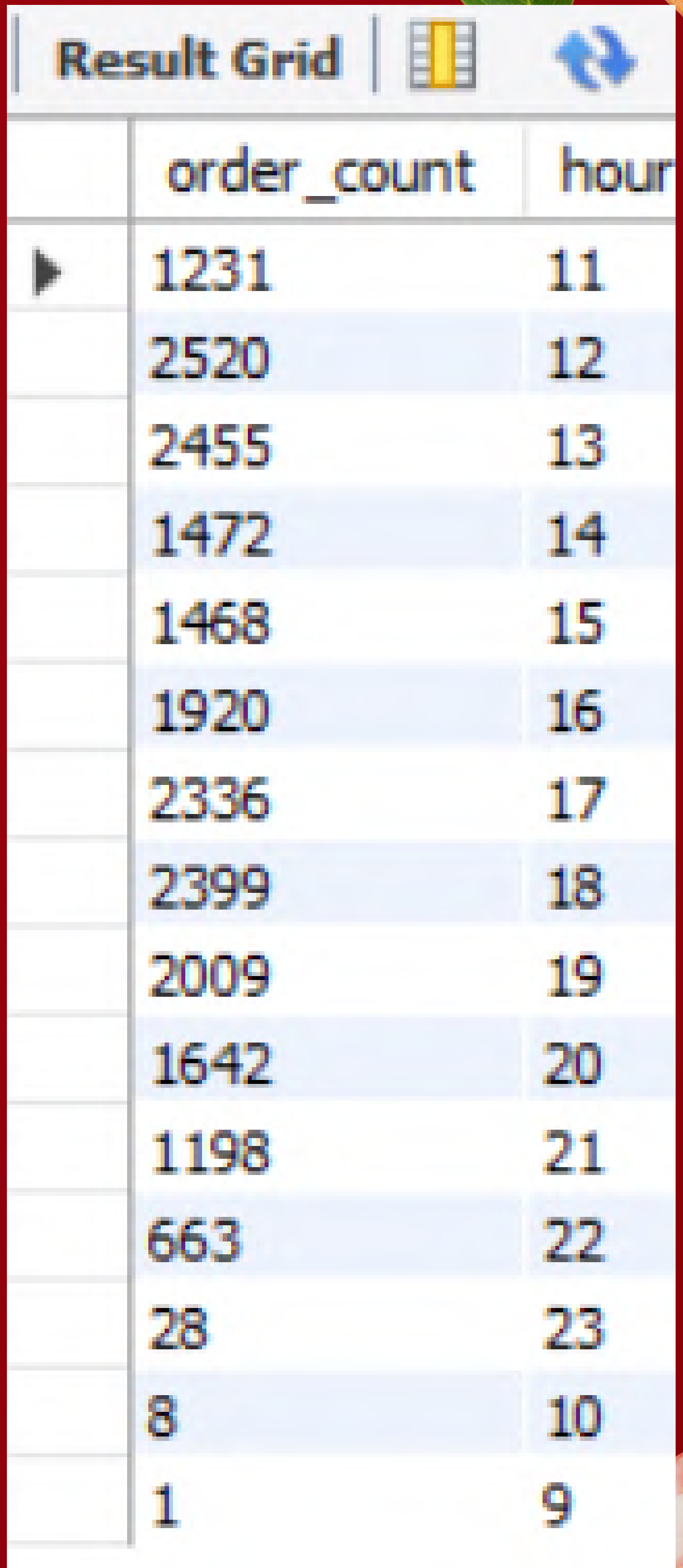


```
  category   quantity
▶ Classic    14888
  Supreme    11987
  Veggie     11649
  Chicken    11050
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

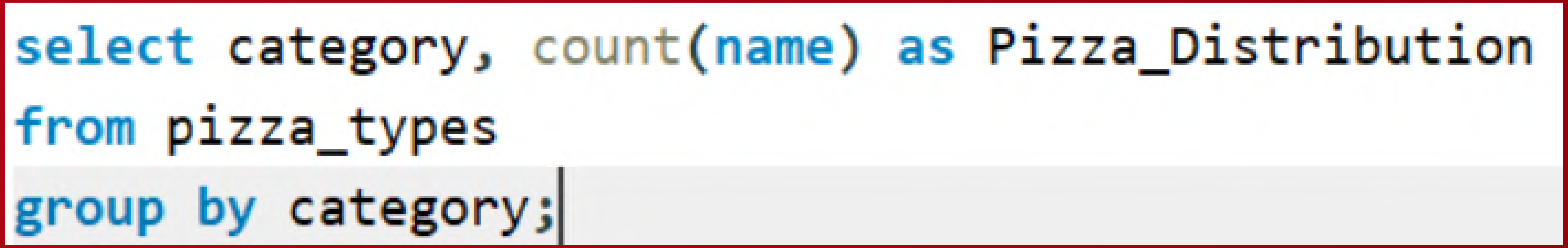


```
select count(order_id) as order_count,
hour(order_time) as hour
from orders
group by hour(order_time);
```

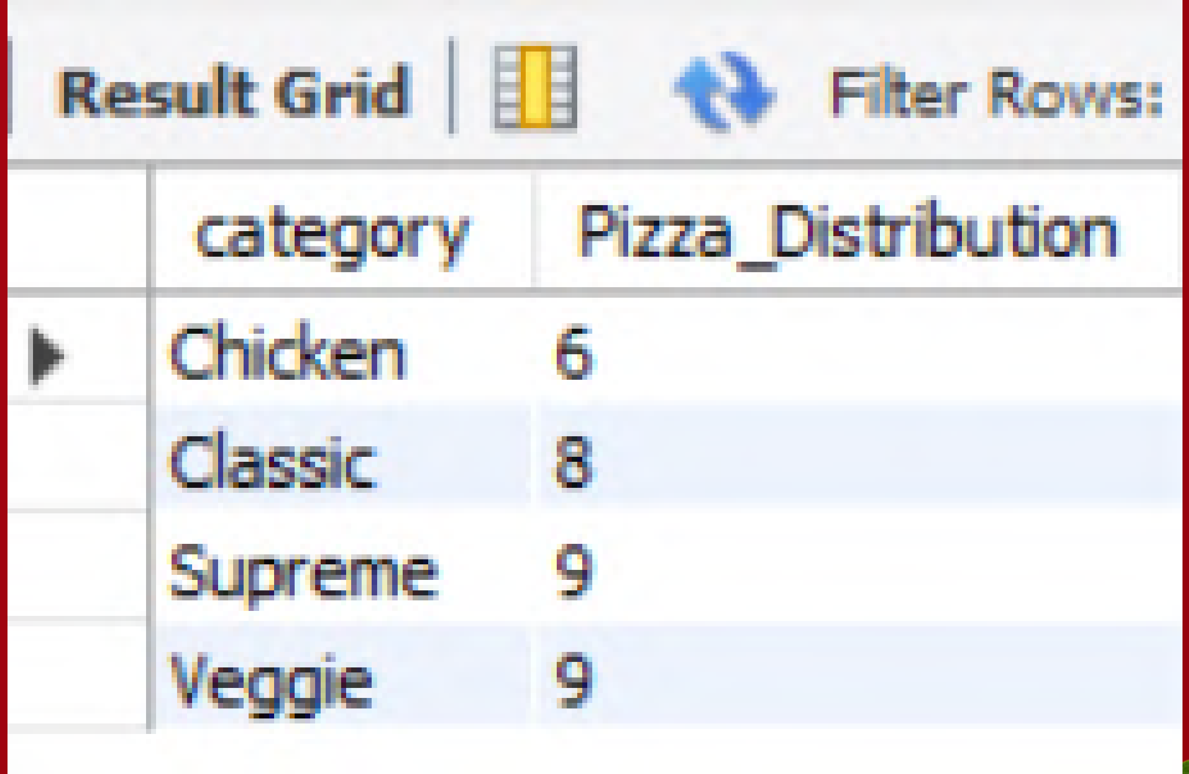


order_count	hour
1231	11
2520	12
2455	13
1472	14
1468	15
1920	16
2336	17
2399	18
2009	19
1642	20
1198	21
663	22
28	23
8	10
1	9

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

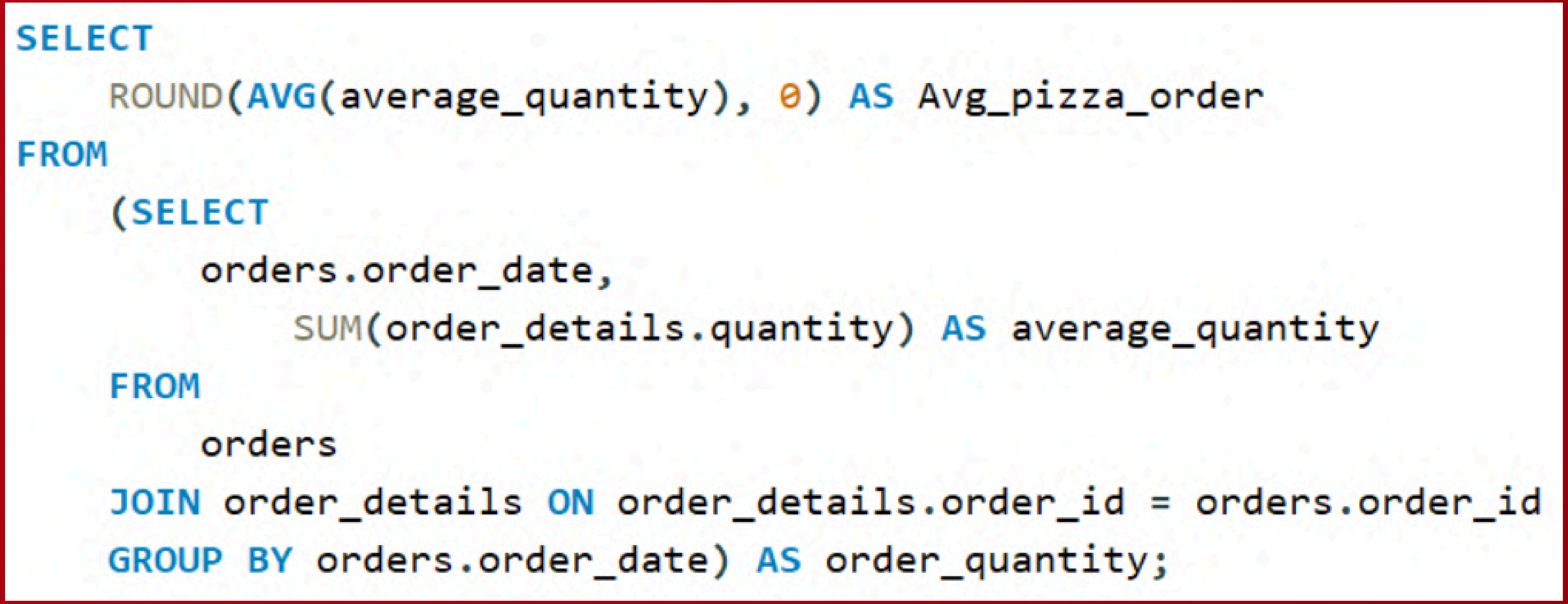


```
select category, count(name) as Pizza_Distribution
from pizza_types
group by category;
```

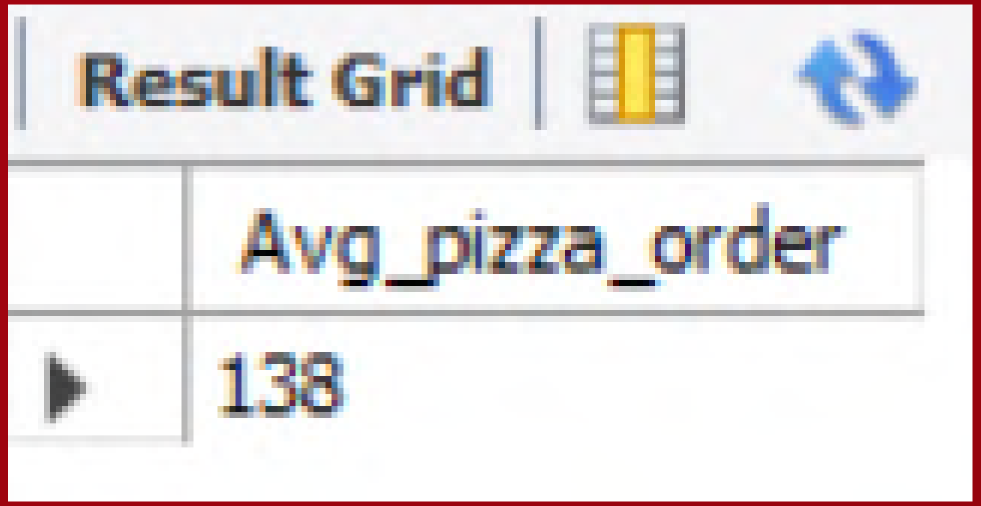


category	Pizza_Distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

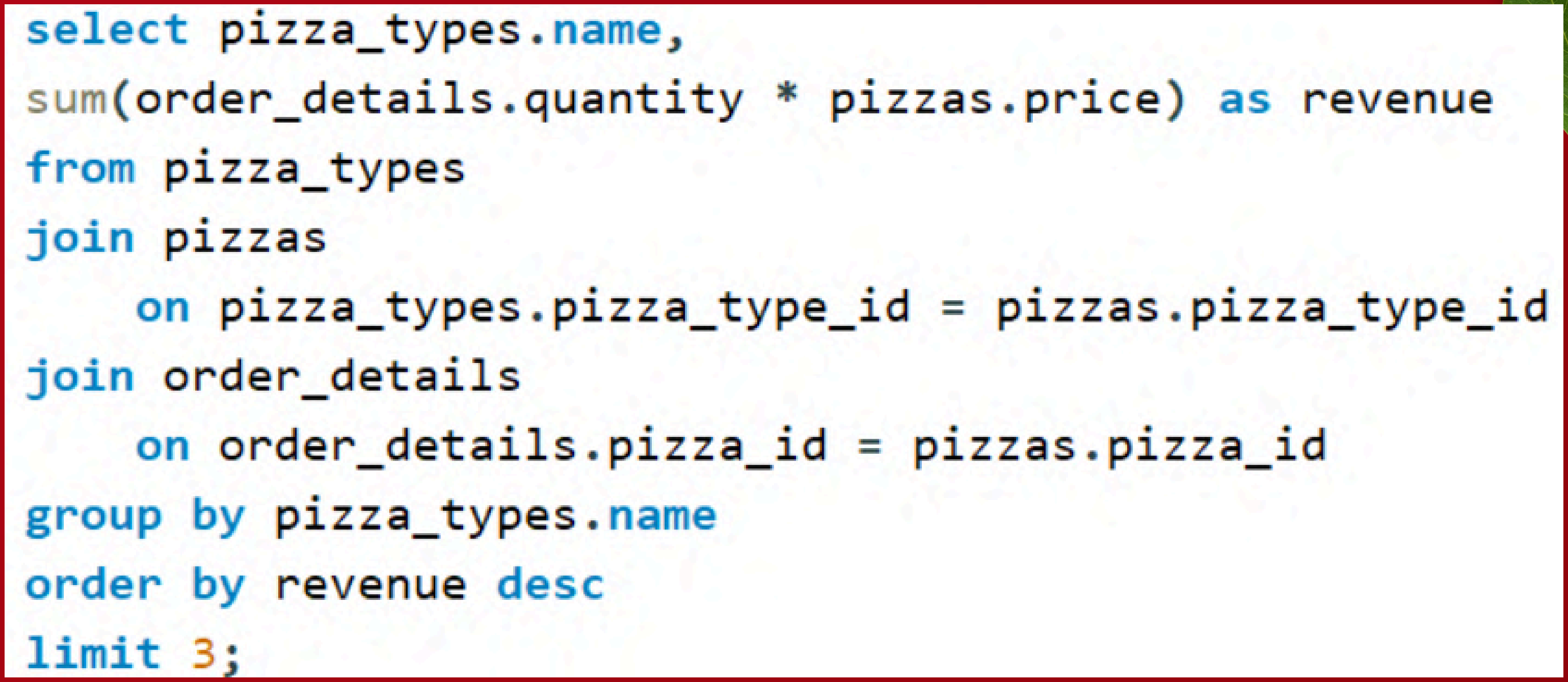


```
SELECT
    ROUND(AVG(average_quantity), 0) AS Avg_pizza_order
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS average_quantity
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS order_quantity;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

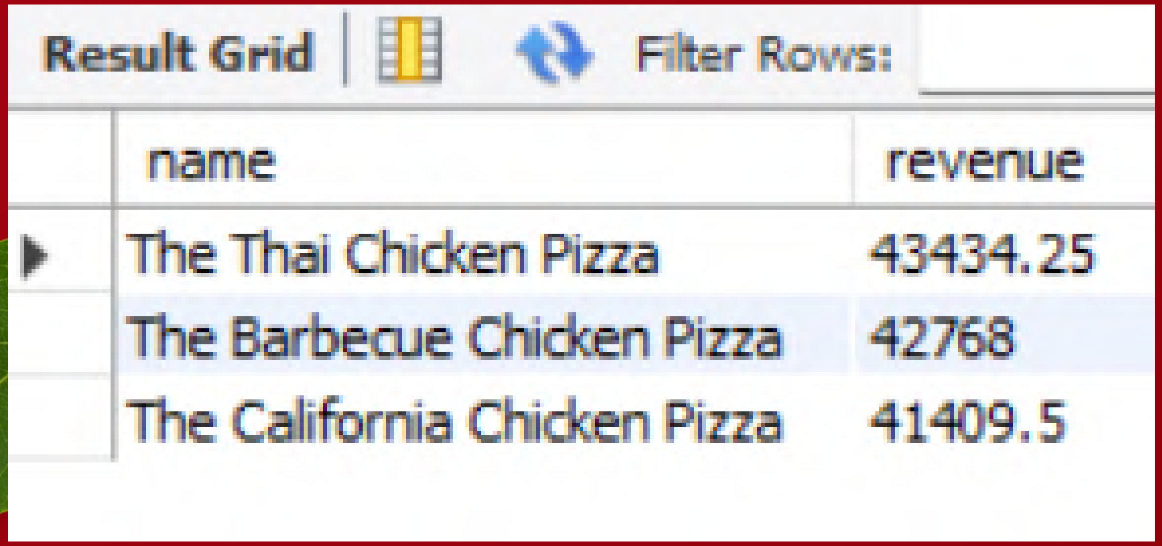


```
Avg_pizza_order
138
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

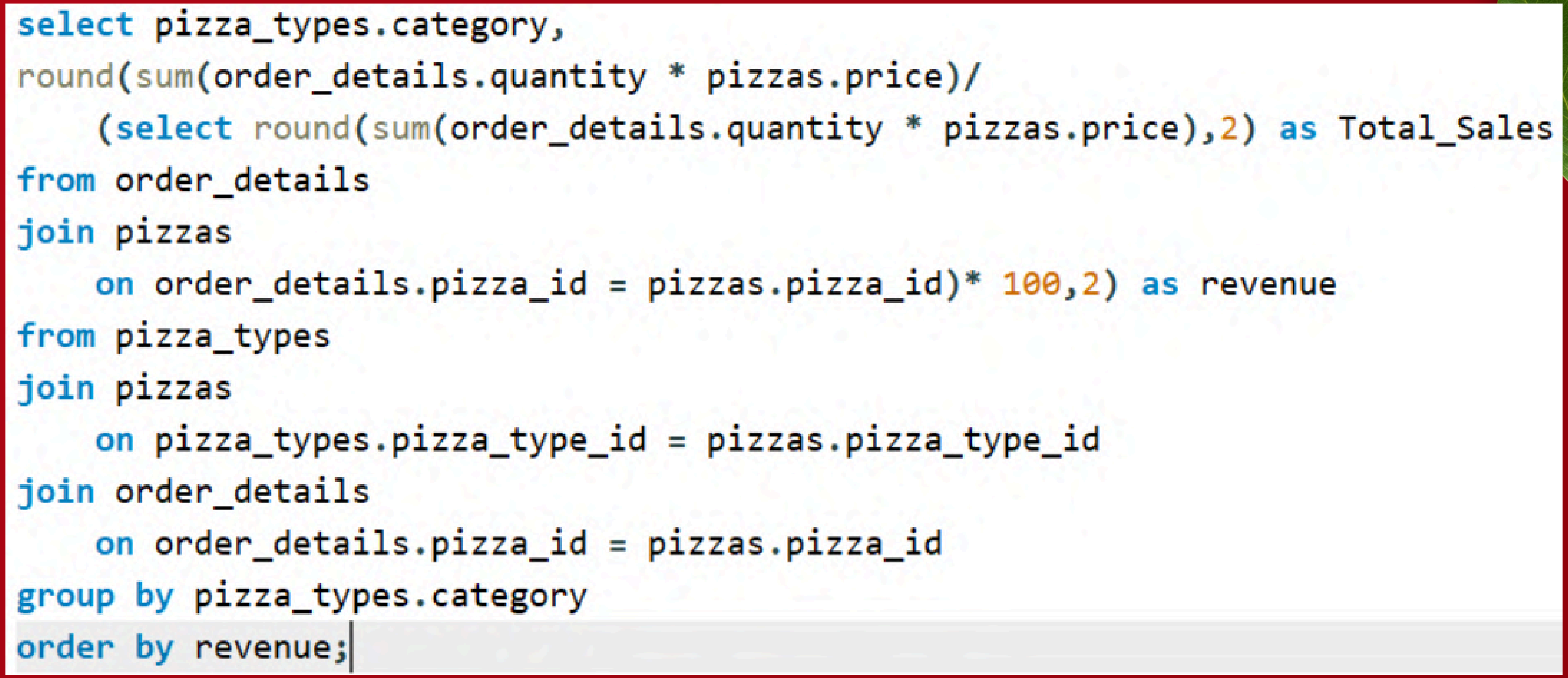


```
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types
join pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
    on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc
limit 3;
```

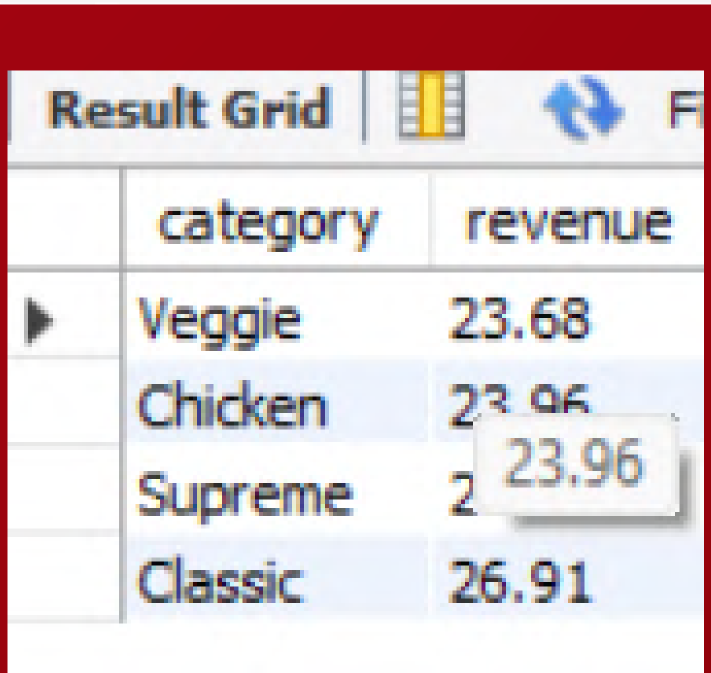


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

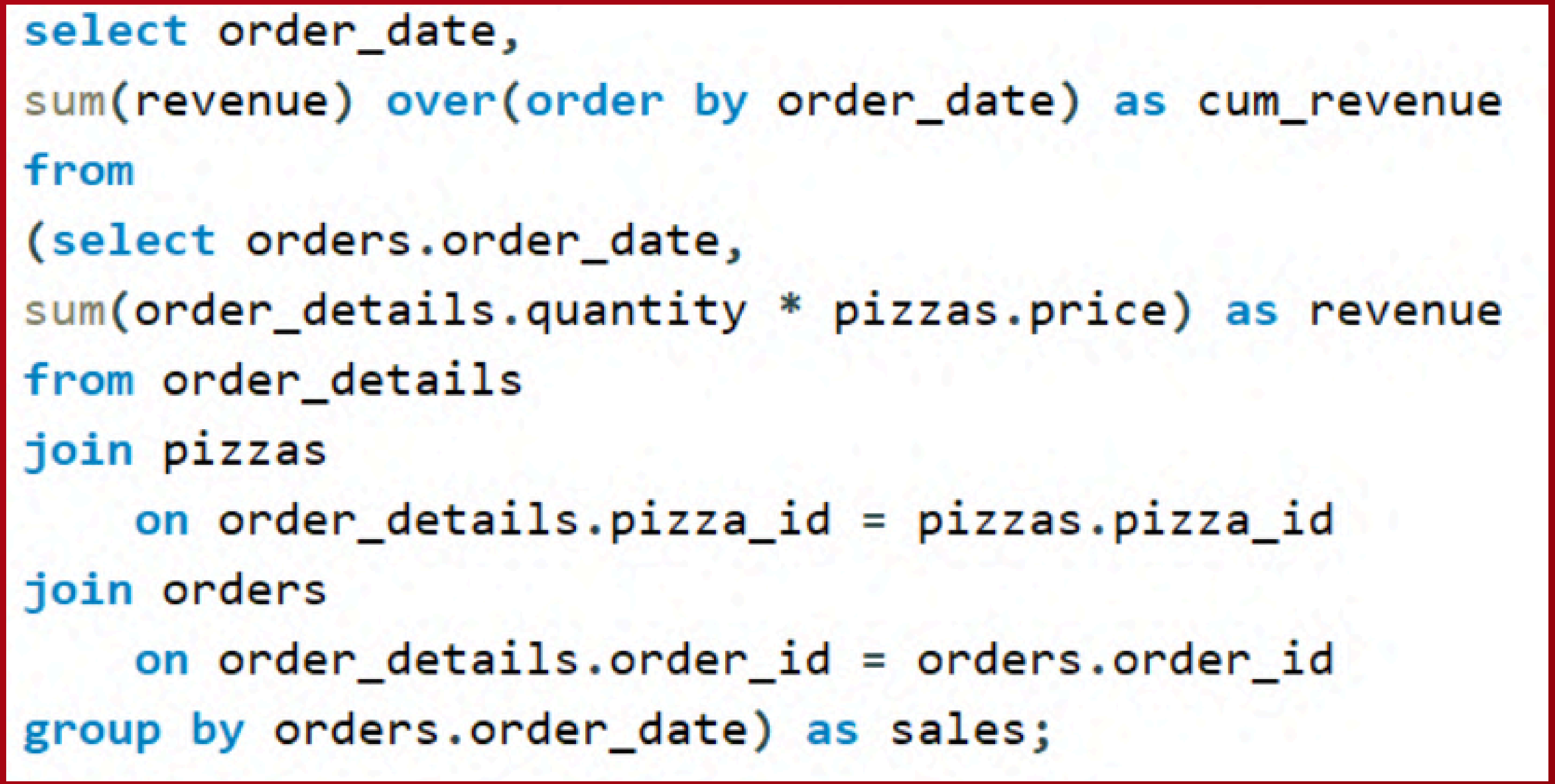


```
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price)/
    (select round(sum(order_details.quantity * pizzas.price),2) as Total_Sales
from order_details
join pizzas
    on order_details.pizza_id = pizzas.pizza_id)* 100,2) as revenue
from pizza_types
join pizzas
    on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
    on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by revenue;
```

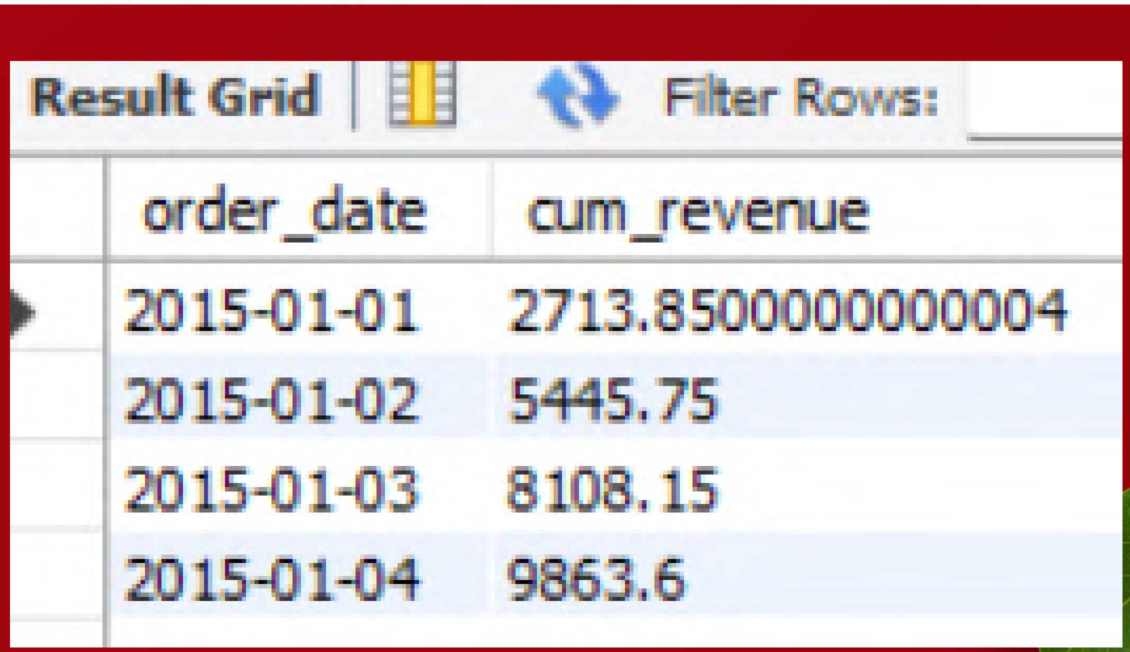


category	revenue
Veggie	23.68
Chicken	23.96
Supreme	2
Classic	26.91

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

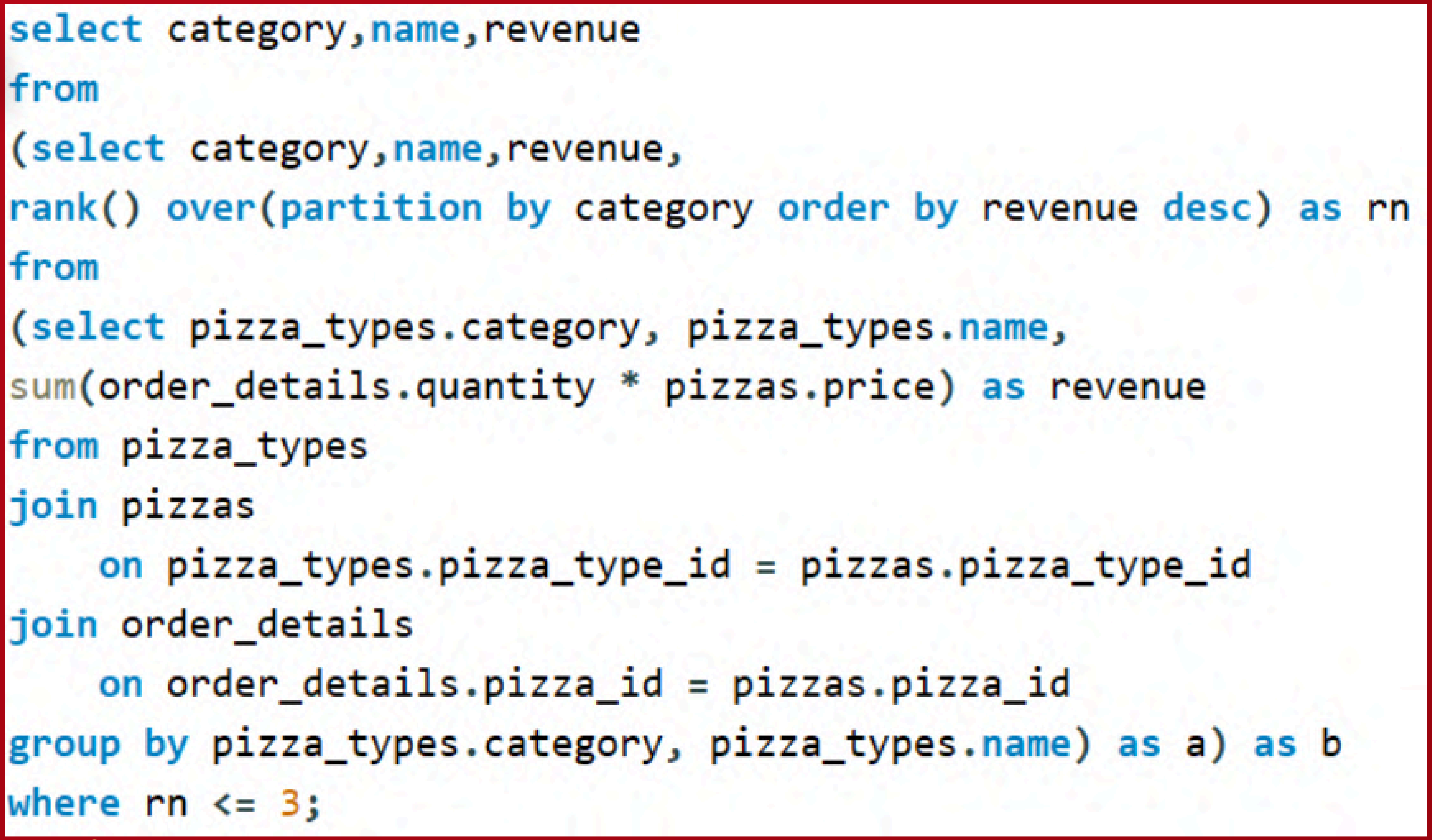


```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details
join pizzas
    on order_details.pizza_id = pizzas.pizza_id
join orders
    on order_details.order_id = orders.order_id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

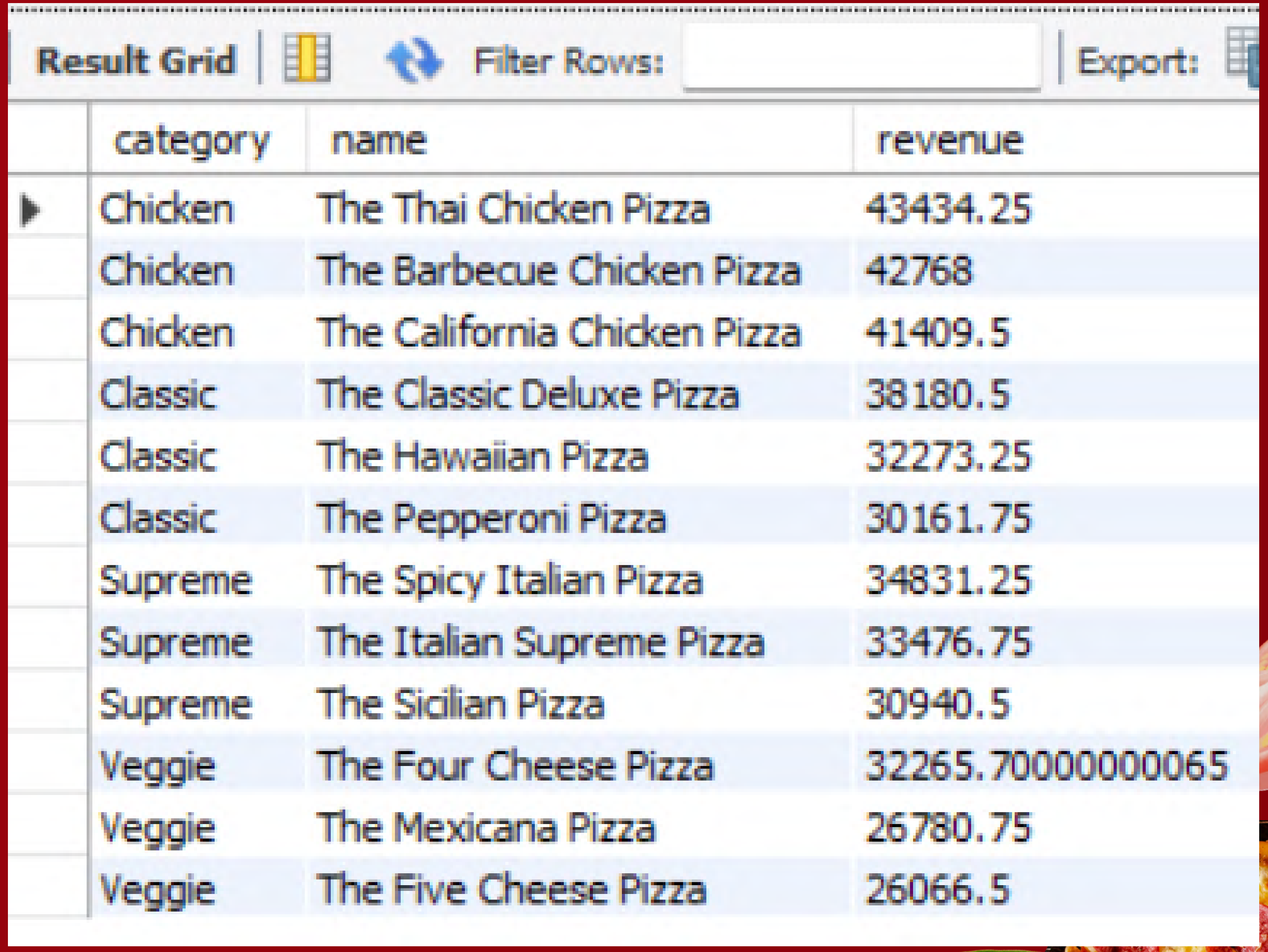


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6

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



```
select category,name,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;
```



category	name	revenue
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	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU!

Thank you for taking the time to review my Pizza Sales Analysis project.
I hope the insights provided highlight the value of data-driven decision-making in business operations.

I’m open to full-time opportunities or internships in Data Analytics roles.
Skilled in SQL, Excel, Power BI, and Python, with hands-on project experience.
Let’s connect!