PIZZA SALES ANALYSIS USING SQL

● By Arun R

Hello, my name is Arun R.
For this project, I completed a Pizza Sales Analysis using SQL. I utilized SQL queries to explore the dataset and extract key insights that could help enhance sales and improve operational efficiency.
This project highlights my SQL skills and my ability to work with data to support informed business decisions. I analyzed and answered 13 insightful, business-related questions based on pizza sales data.
Through this project, I gained a deeper understanding of database querying, business intelligence, and customer behavior using real-time data.

OBJECTIVES:

1.Sales and Revenue Analysis:

Evaluate total orders and revenue to identify best-selling and highest-priced pizzas.

2.Customer Preference Insights:

Discover customer preferences by analyzing the most common pizza types and sizes ordered.

3.Category-Level Trends:

Examine order quantities and revenues across pizza categories to reveal performance patterns.

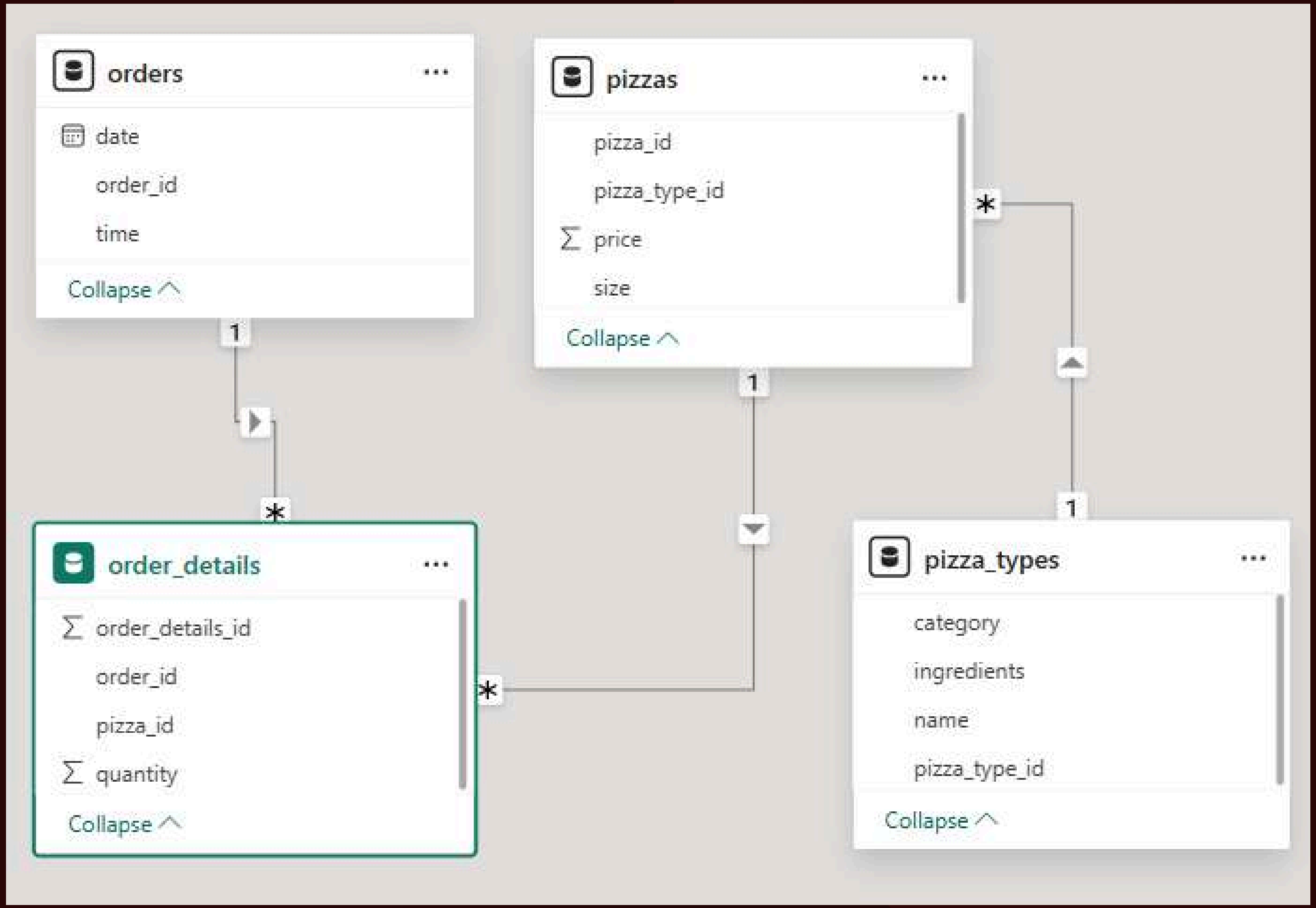
4.Time-Based Sales Patterns:

Analyze hourly and daily order trends to identify peak sales periods.

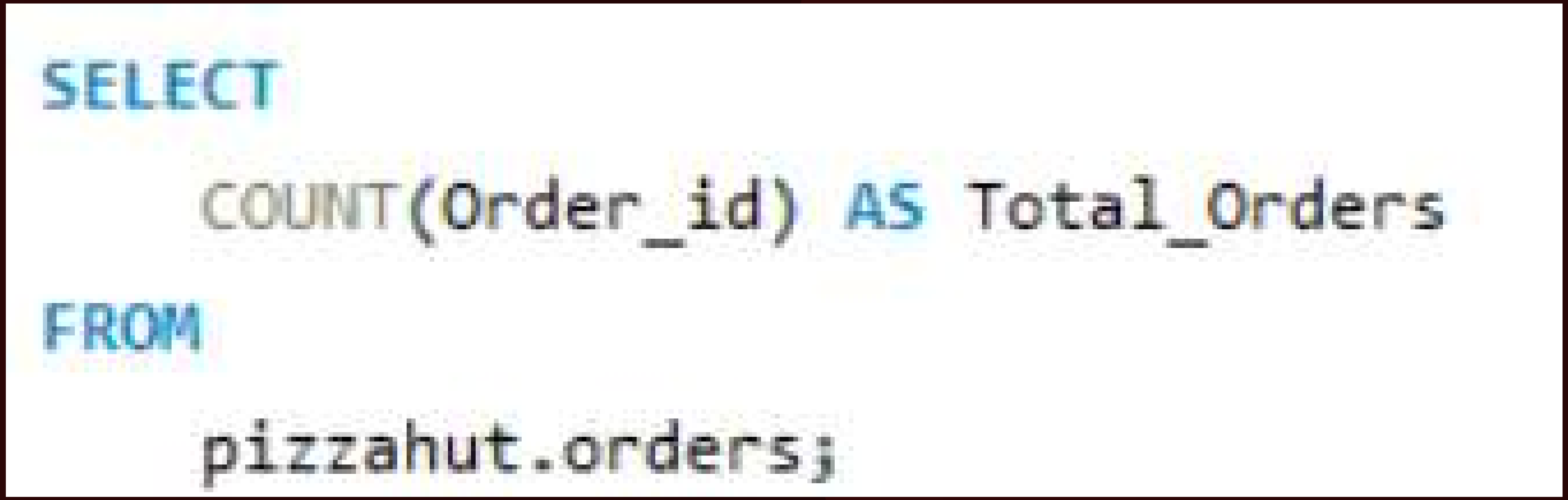
5.Revenue Contribution and Growth Analysis:

Assess each pizza type's contribution to total revenue and track cumulative revenue over time.

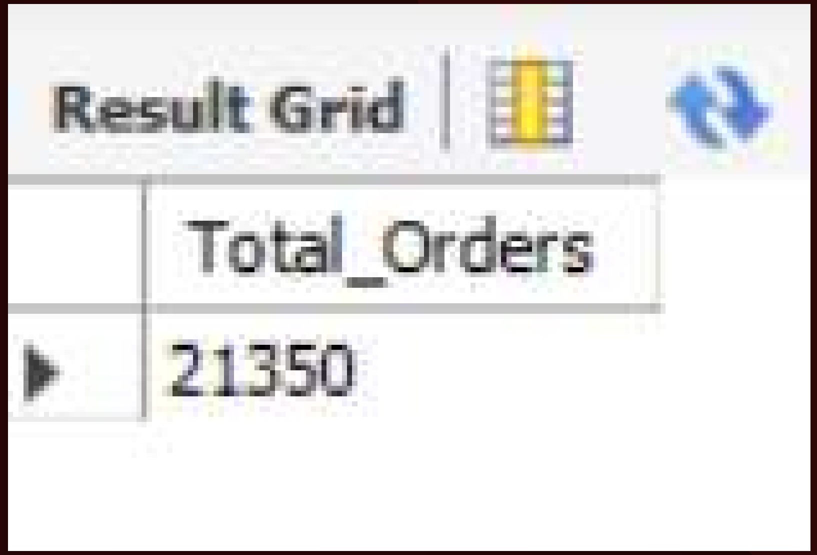
🛠️ Schema Blueprint: Tables & Relationships



Retrieve the total number of orders placed.

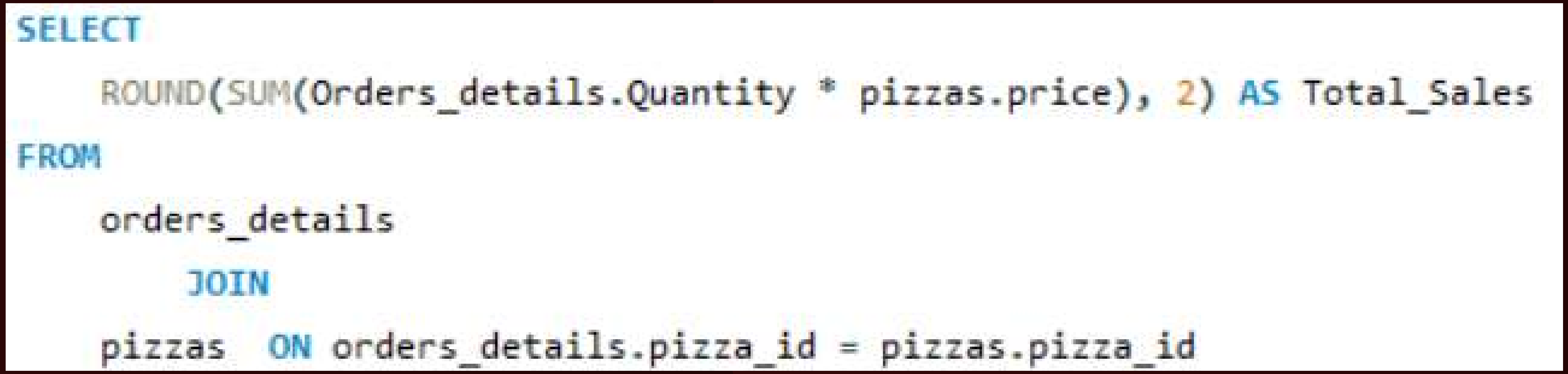


```
SELECT
    COUNT(Order_id) AS Total_Orders
FROM
    pizzahut.orders;
```

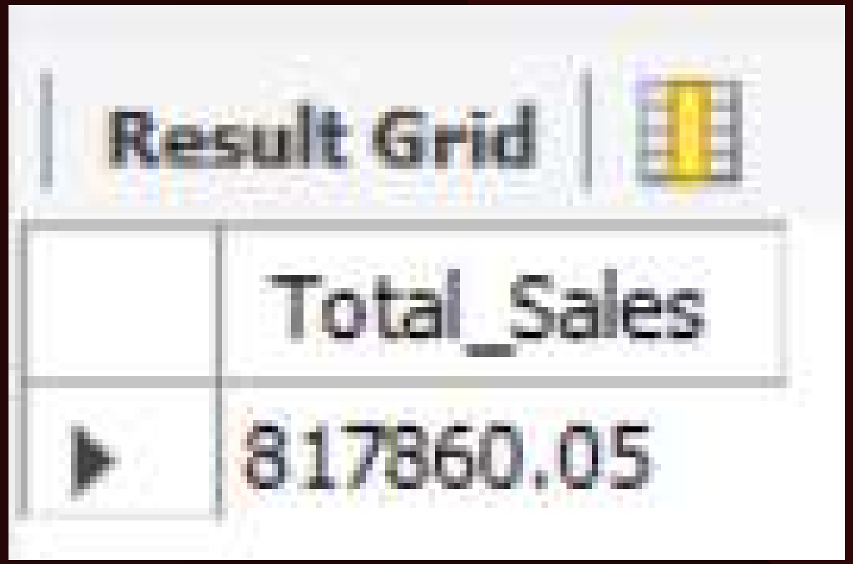


```
Total_Orders
21350
```

Calculate the total revenue generated from pizza sales.



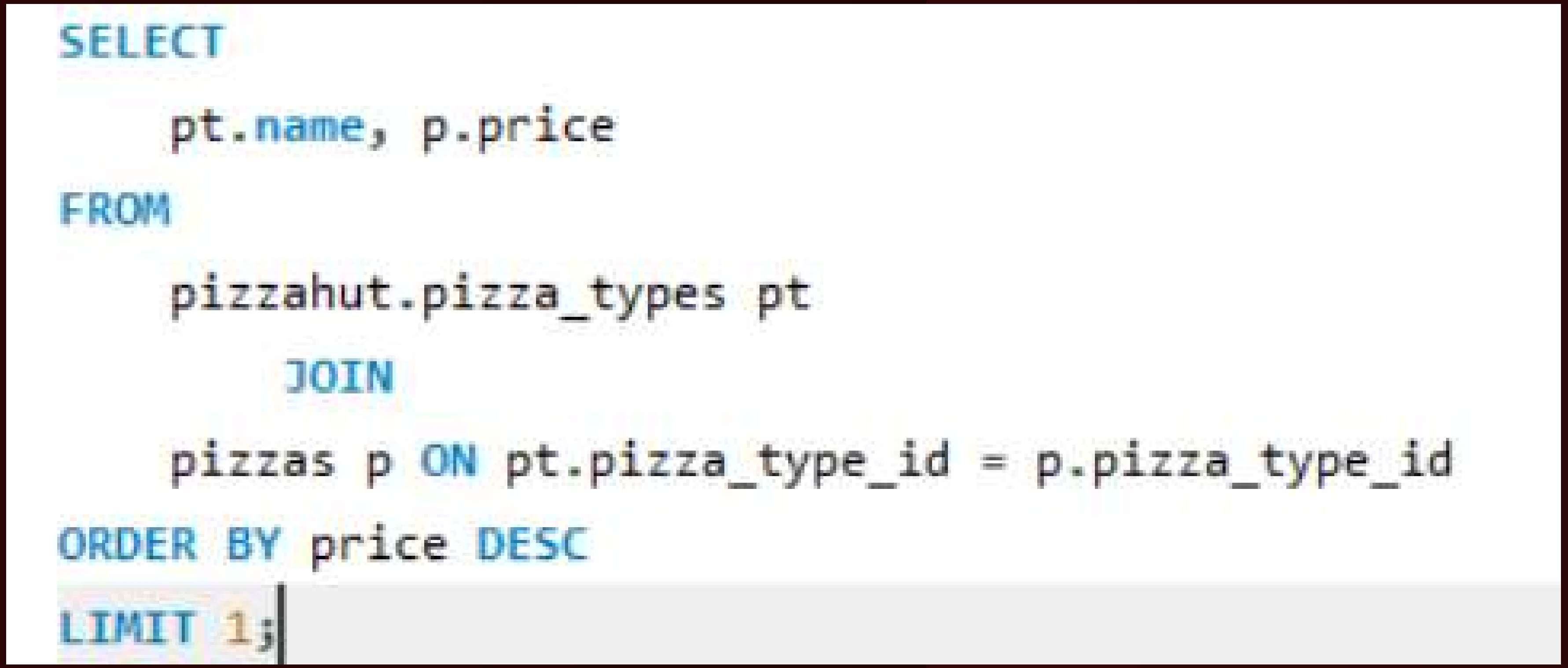
```
SELECT
    ROUND(SUM(Orders_details.Quantity * pizzas.price), 2) AS Total_Sales
FROM
    orders_details
        JOIN
    pizzas  ON orders_details.pizza_id = pizzas.pizza_id
```



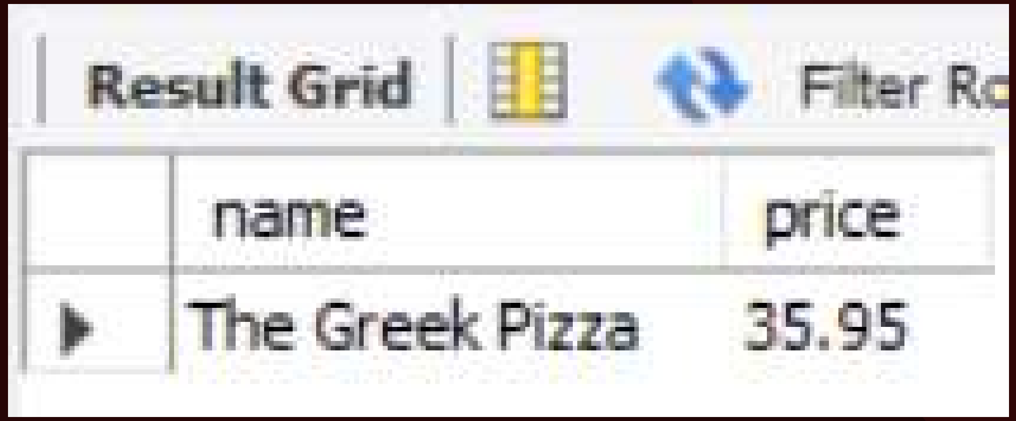
Result Grid

Total_Sales
817860.05

Identify the highest-priced pizza.

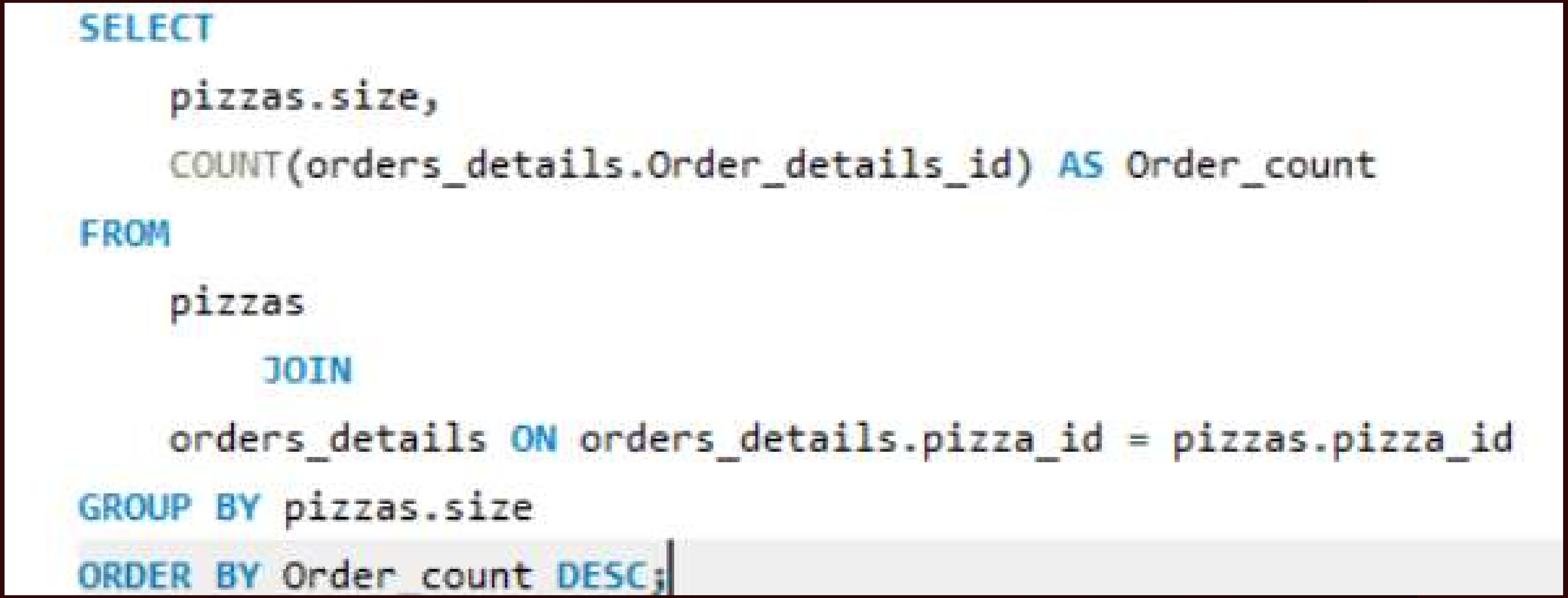


```
SELECT
    pt.name, p.price
FROM
    pizzahut.pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

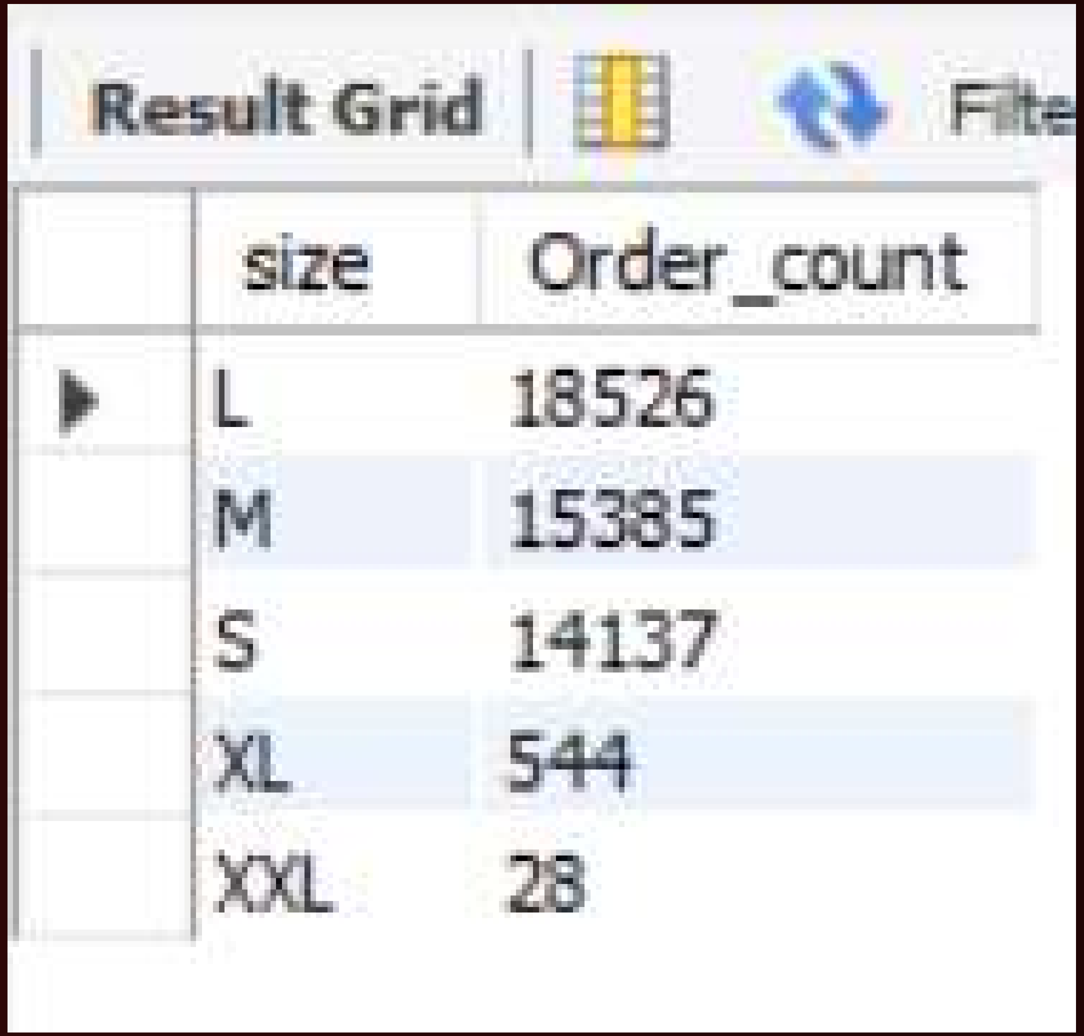


```
name             price
The Greek Pizza  35.95
```

Identify the most common pizza size ordered.

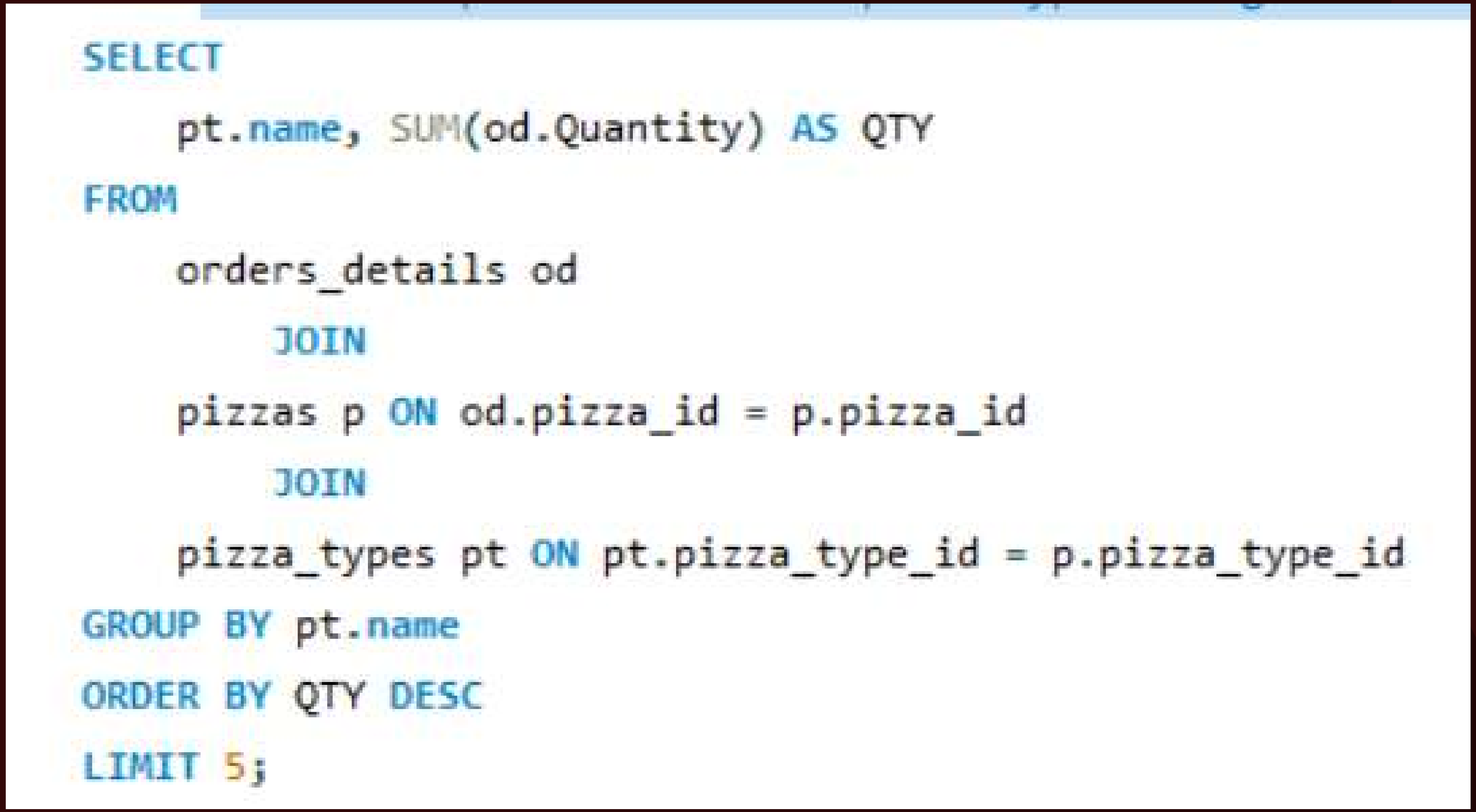


```
SELECT
    pizzas.size,
    COUNT(orders_details.Order_details_id) AS Order_count
FROM
    pizzas
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY Order_count DESC;
```

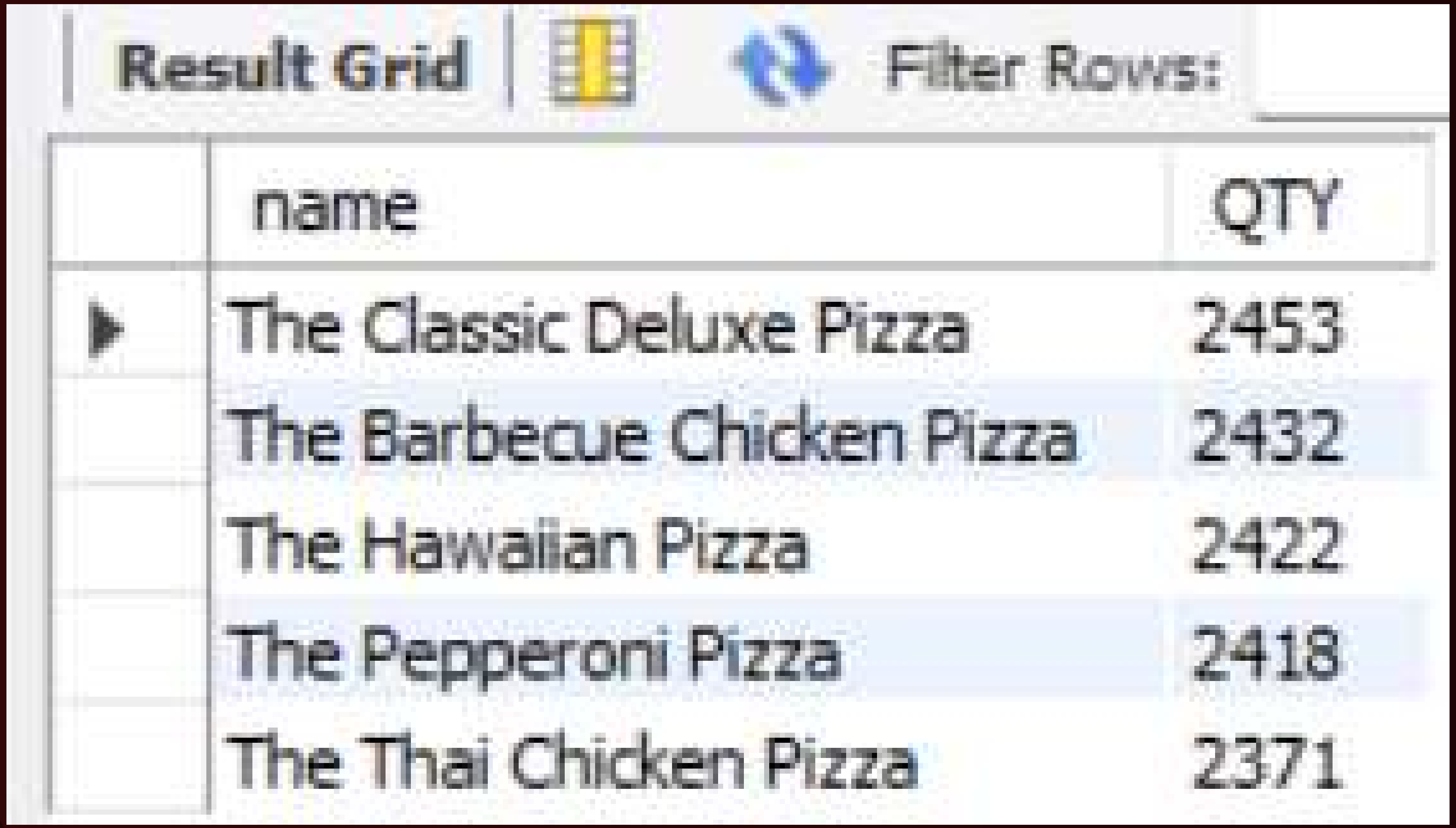


size	Order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

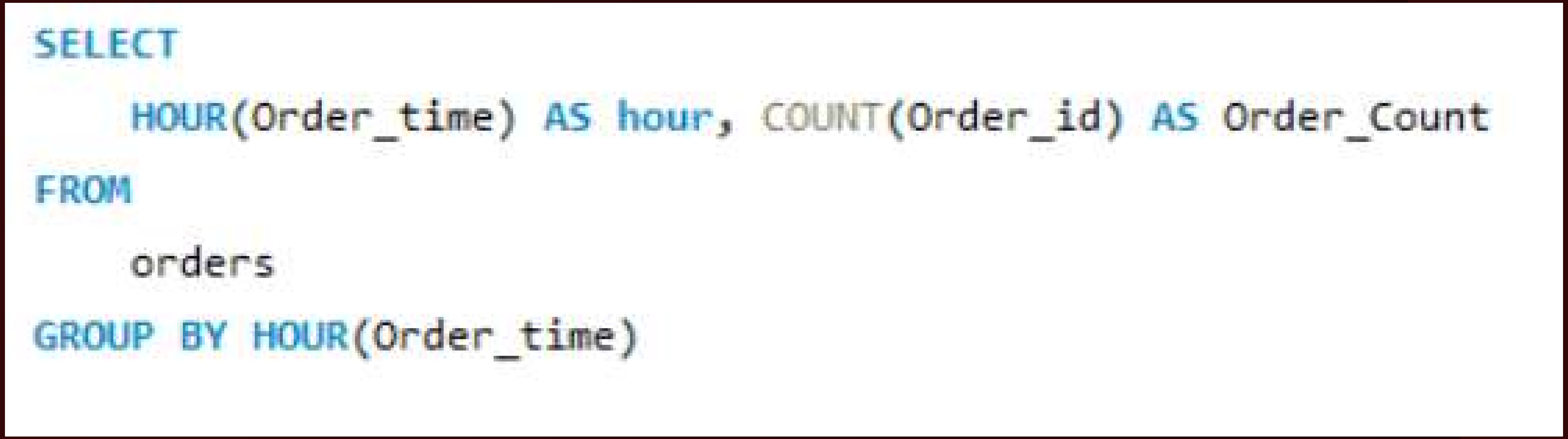


```
SELECT
    pt.name, SUM(od.Quantity) AS QTY
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY QTY DESC
LIMIT 5;
```

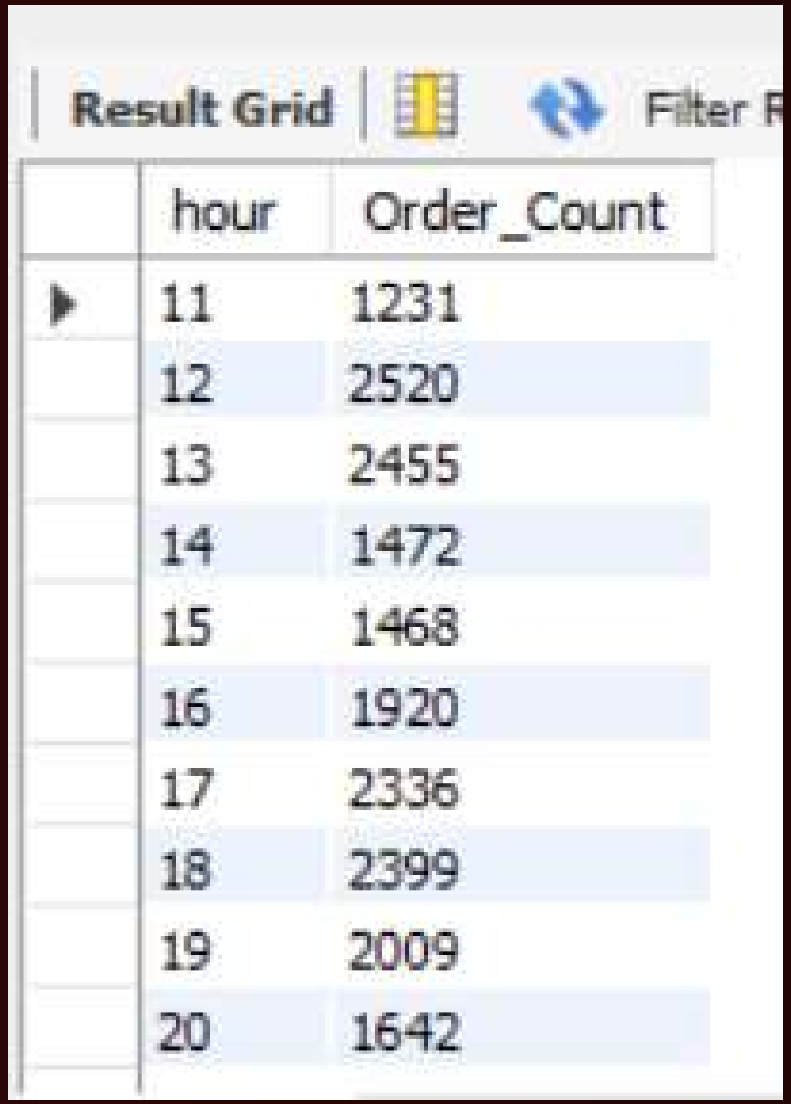


name	QTY
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Determine the distribution of orders by hour of the day.

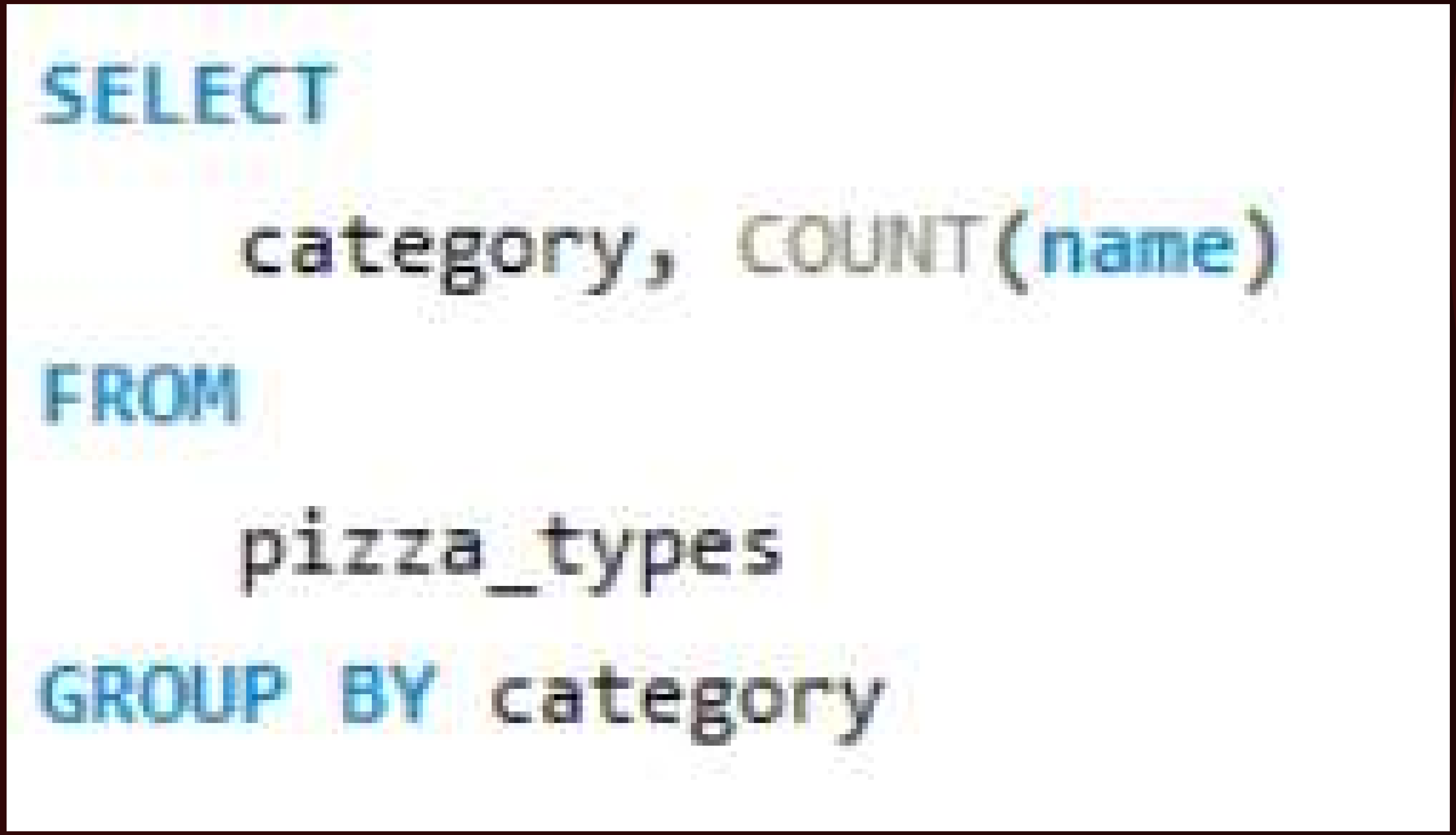


```
SELECT
    HOUR(Order_time) AS hour, COUNT(Order_id) AS Order_Count
FROM
    orders
GROUP BY HOUR(Order_time)
```

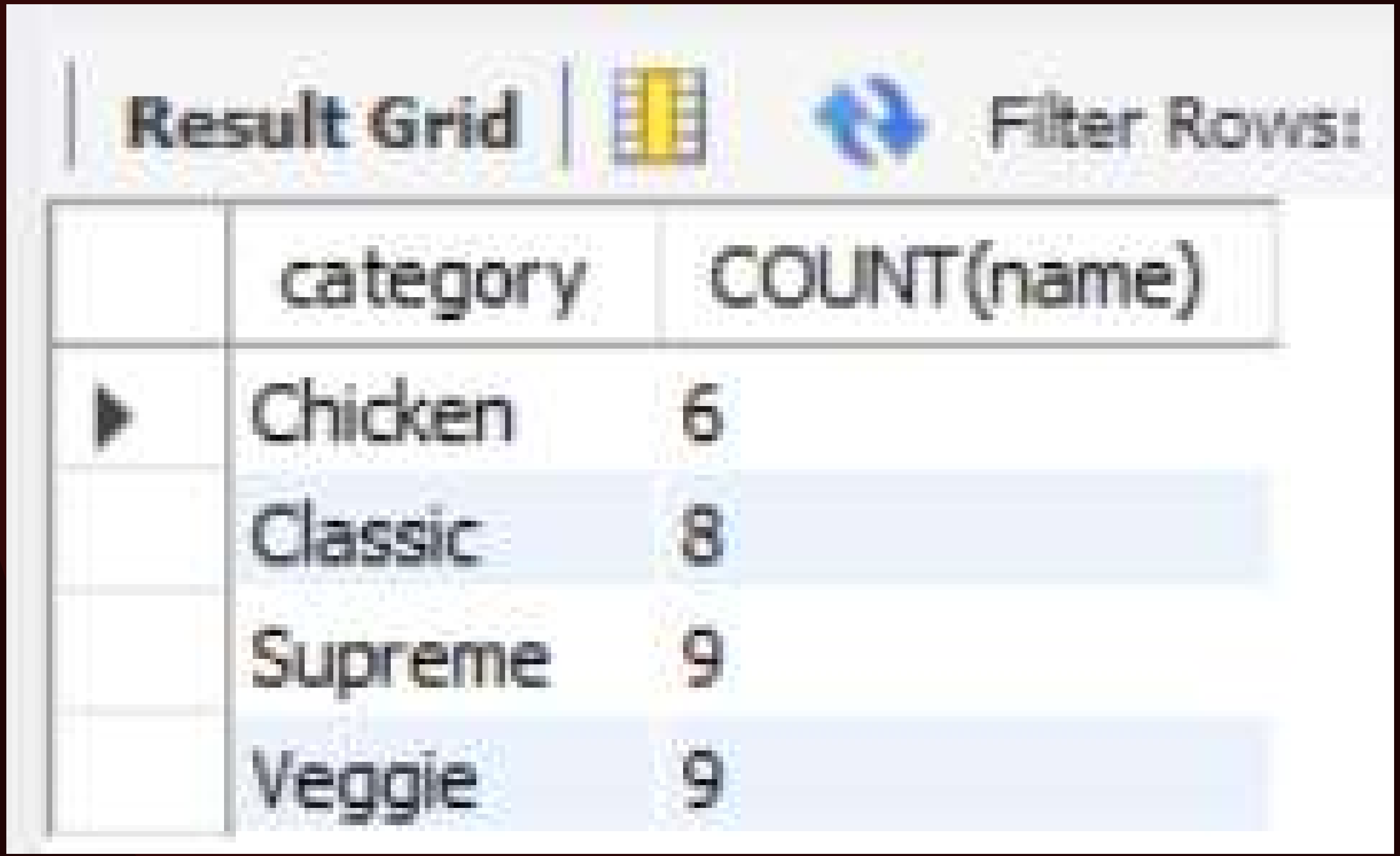


hour	Order_Count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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

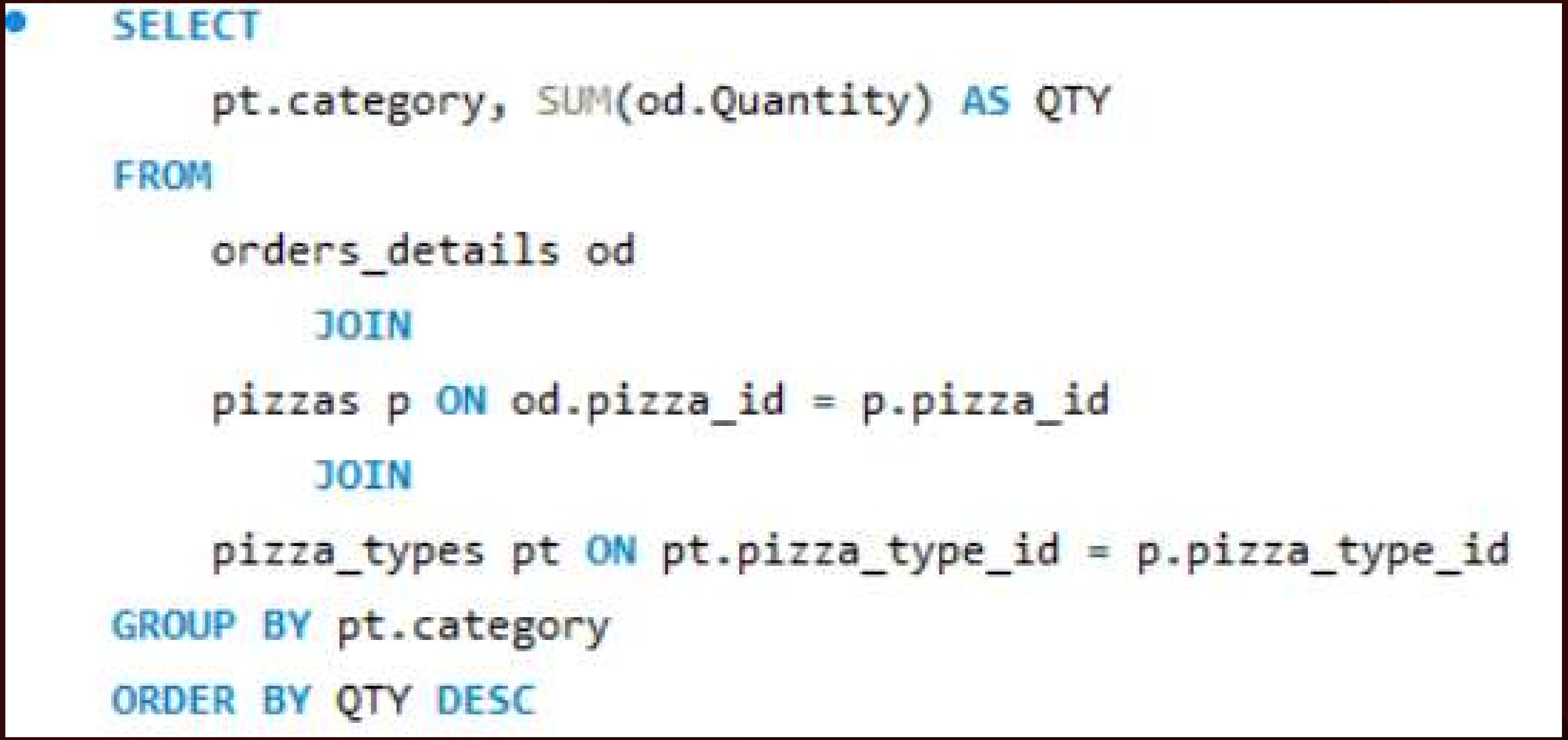


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

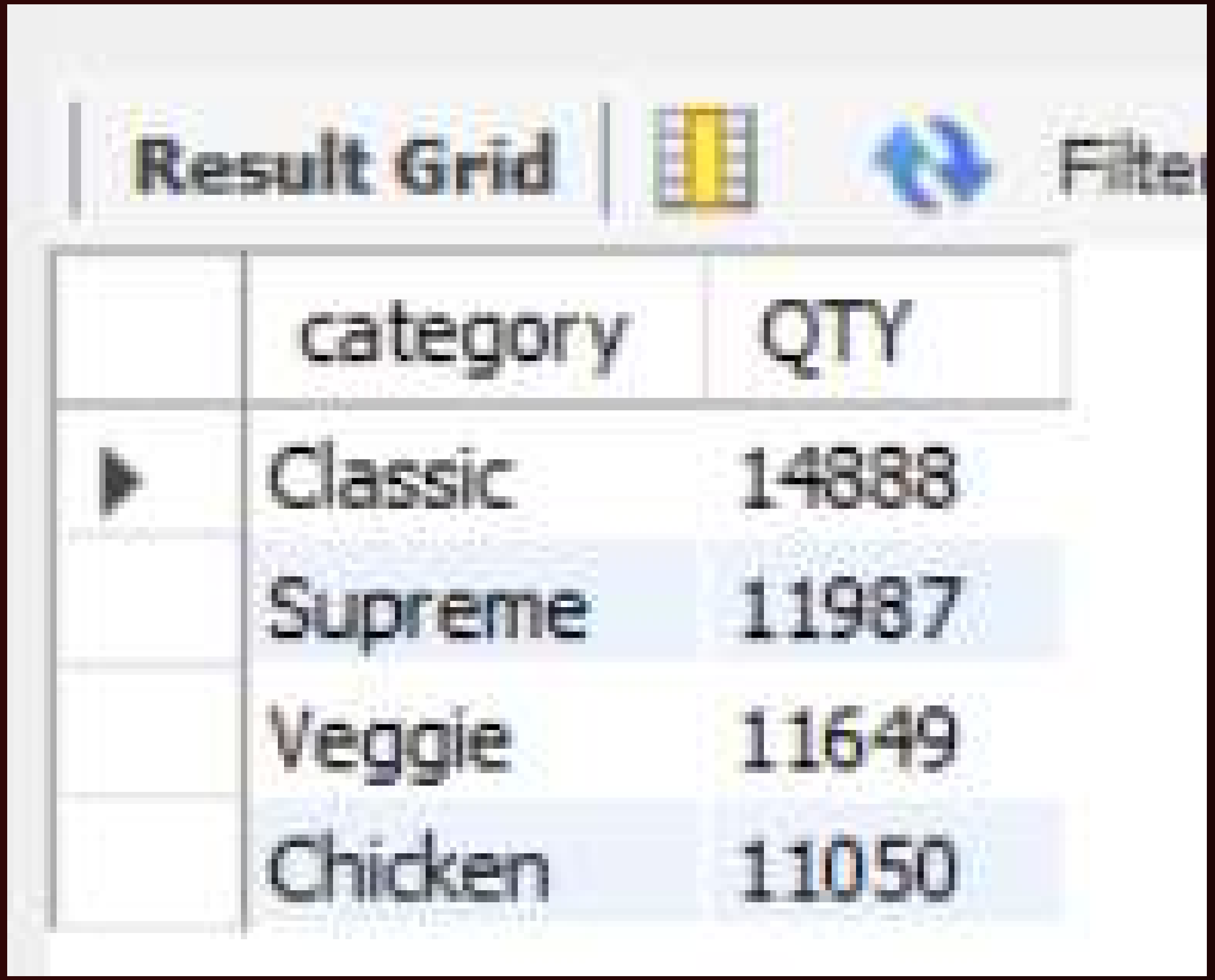


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

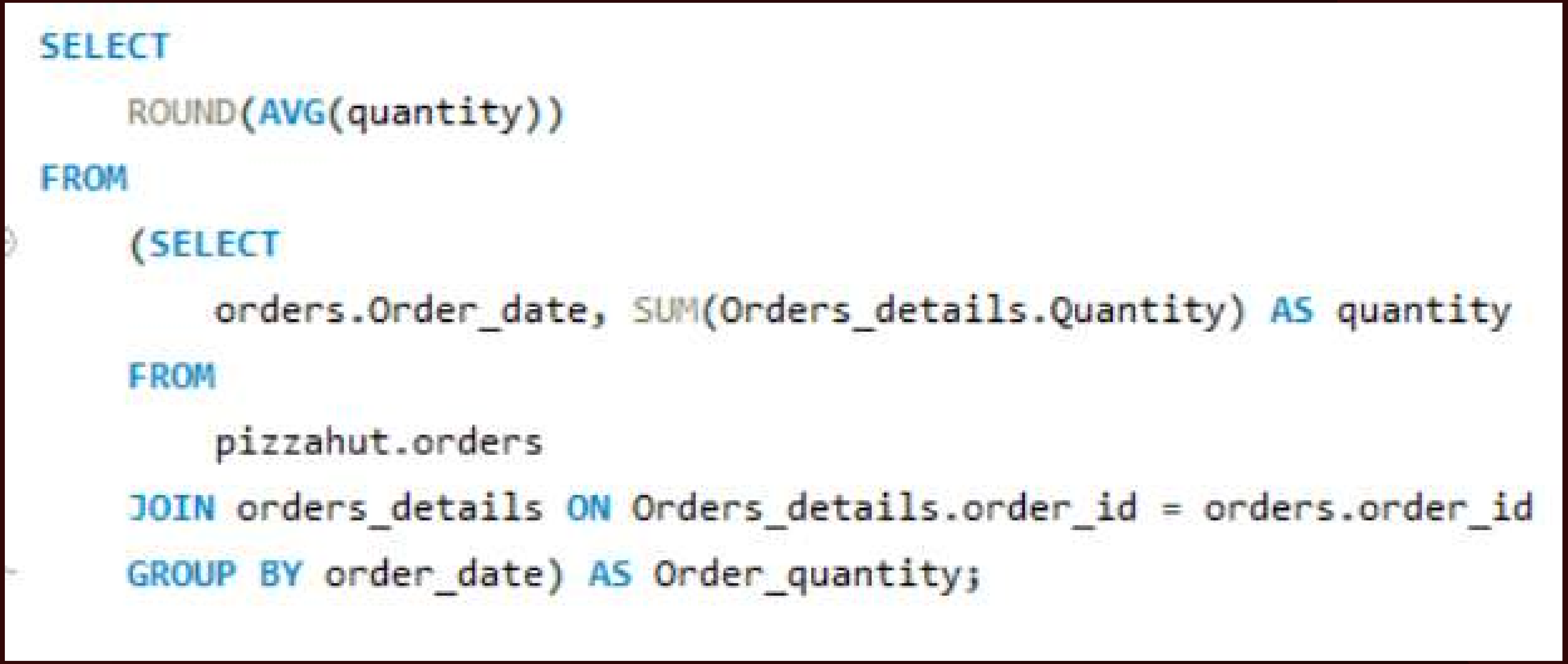


```
SELECT
    pt.category, SUM(od.Quantity) AS QTY
FROM
    orders_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY QTY DESC
```

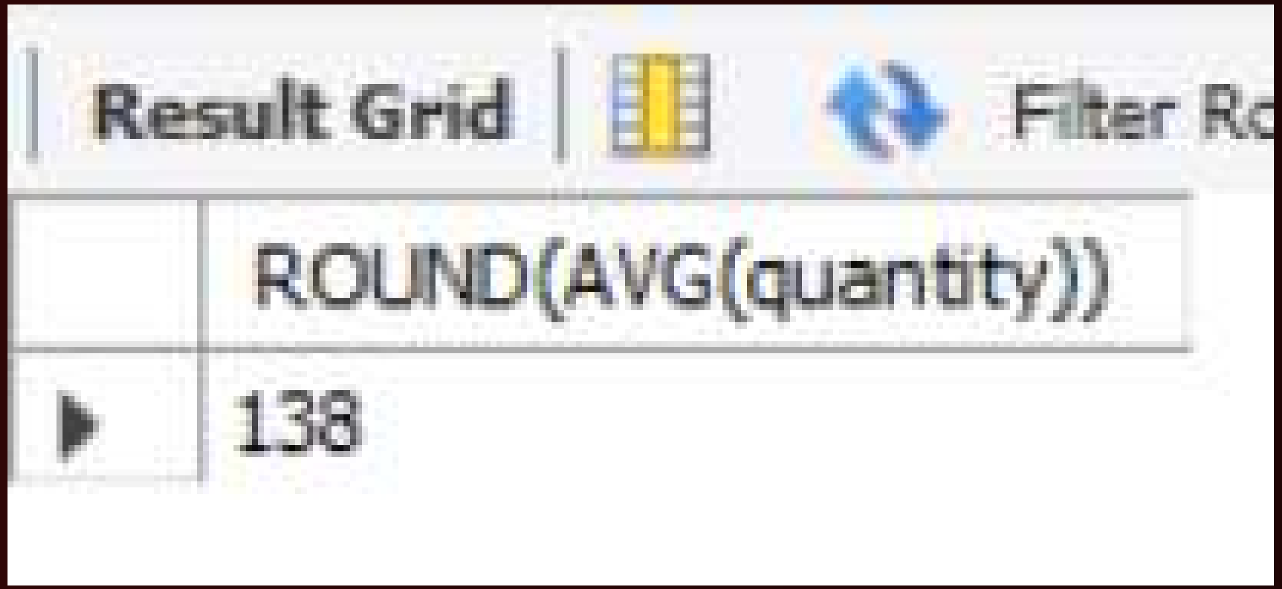


category	QTY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

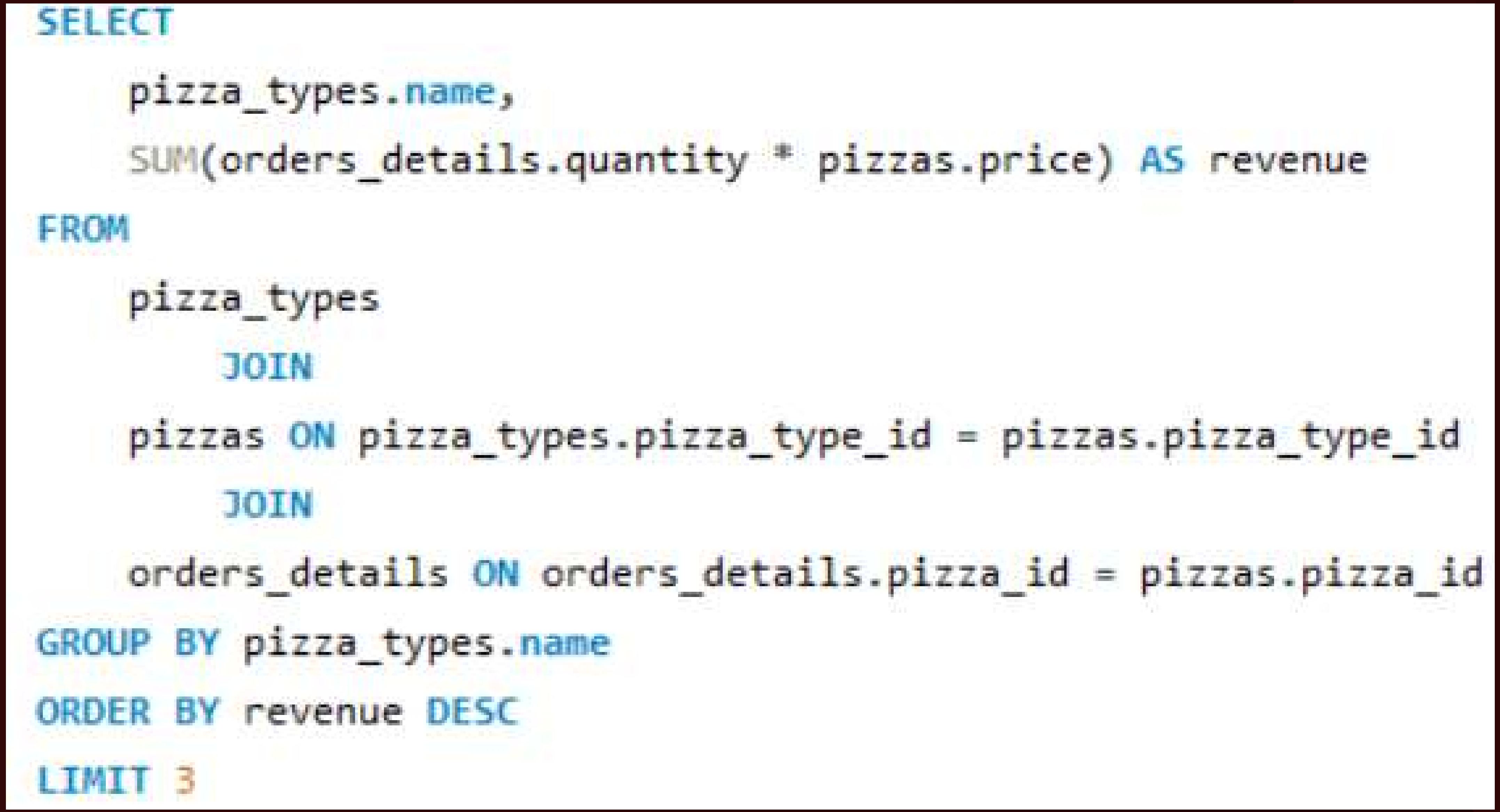


```
SELECT
    ROUND(AVG(quantity))
FROM
    (SELECT
        orders.Order_date, SUM(Orders_details.Quantity) AS quantity
    FROM
        pizzahut.orders
    JOIN orders_details ON Orders_details.order_id = orders.order_id
    GROUP BY order_date) AS Order_quantity;
```

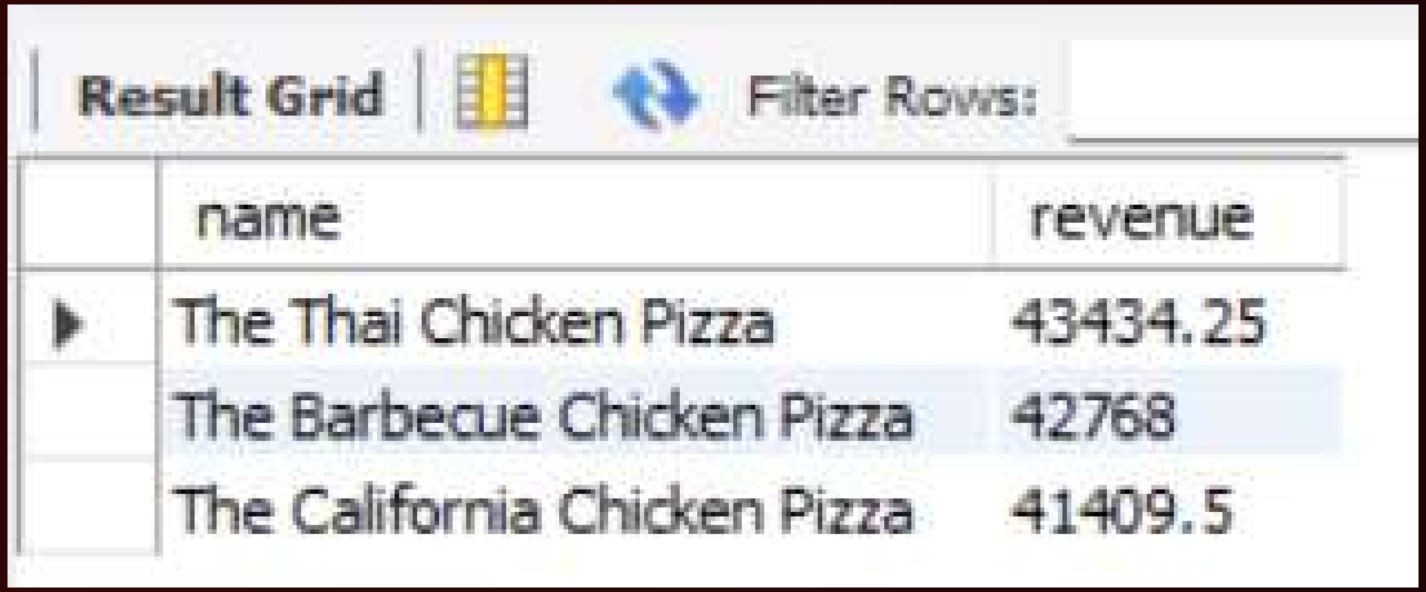


```
ROUND(AVG(quantity))
138
```

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

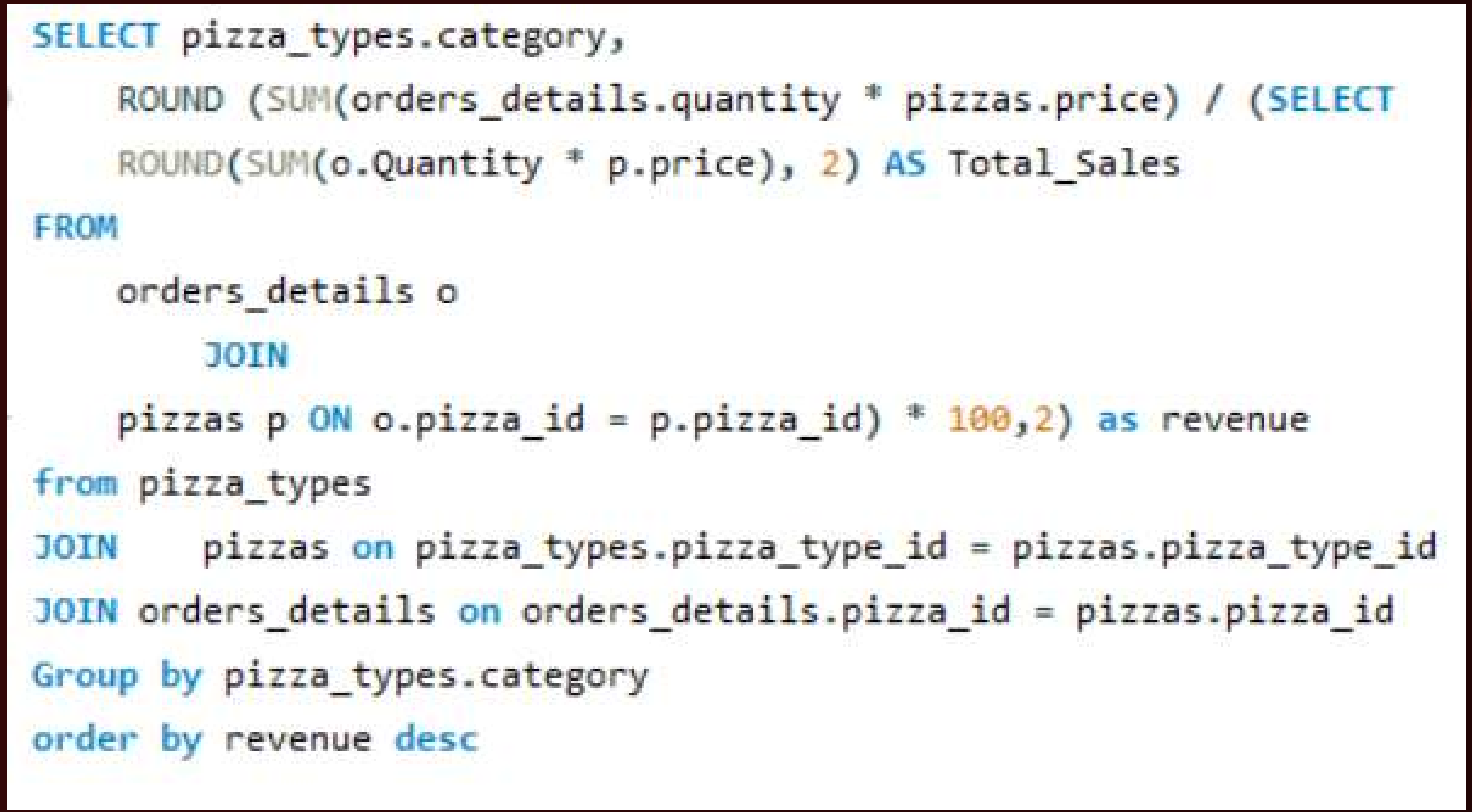


```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY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

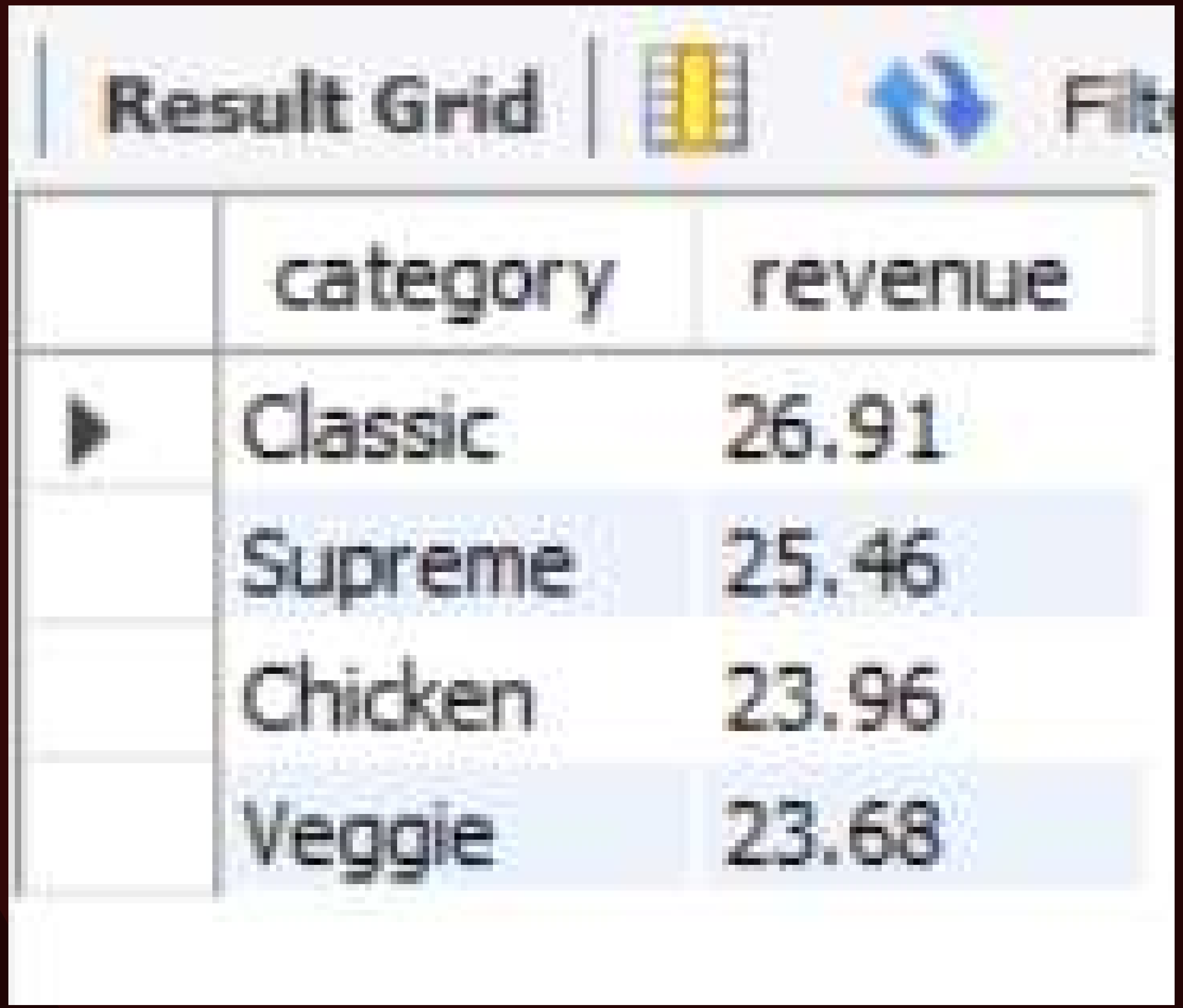


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

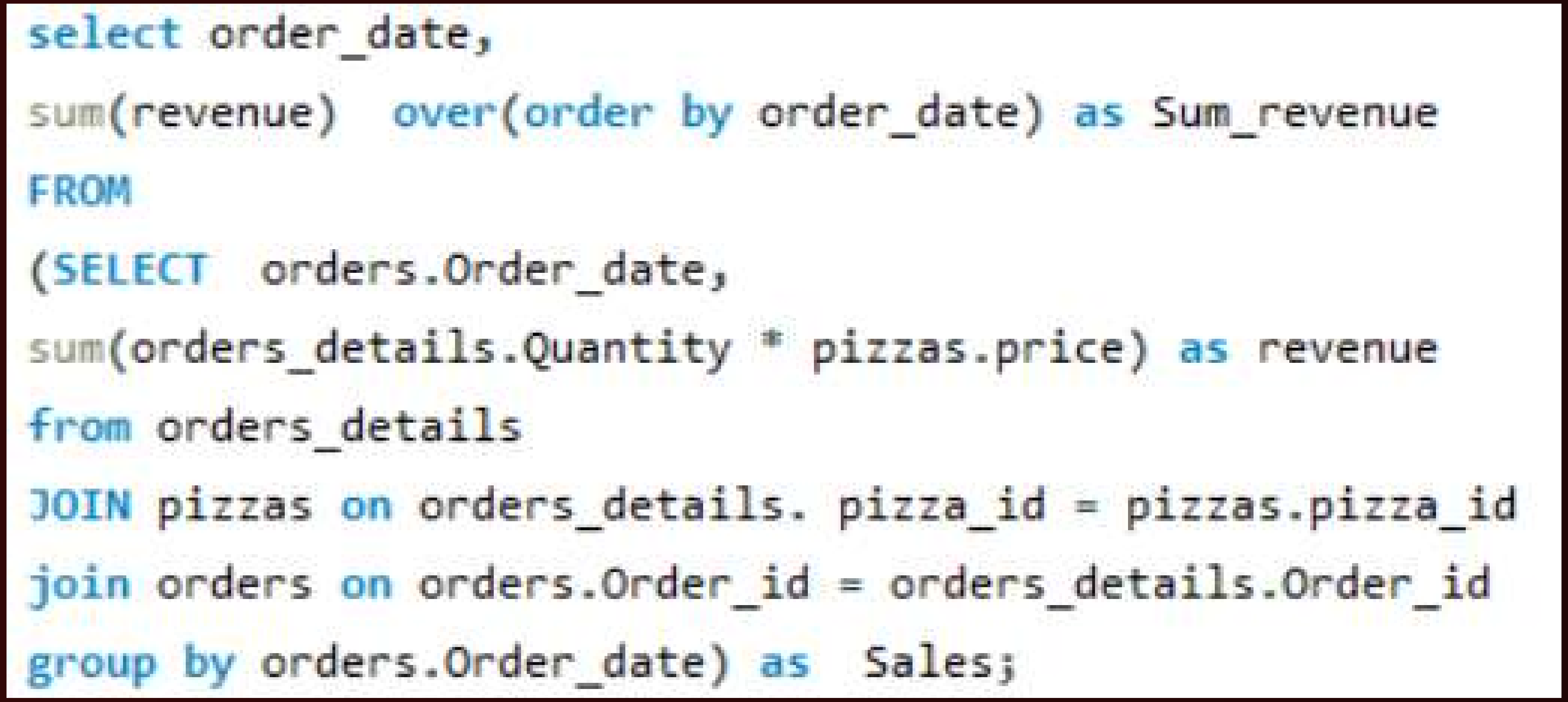


```
SELECT pizza_types.category,
    ROUND (SUM(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(o.Quantity * p.price), 2) AS Total_Sales
FROM
    orders_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id) * 100,2) as revenue
from pizza_types
JOIN    pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details on orders_details.pizza_id = pizzas.pizza_id
Group by pizza_types.category
order by revenue desc
```

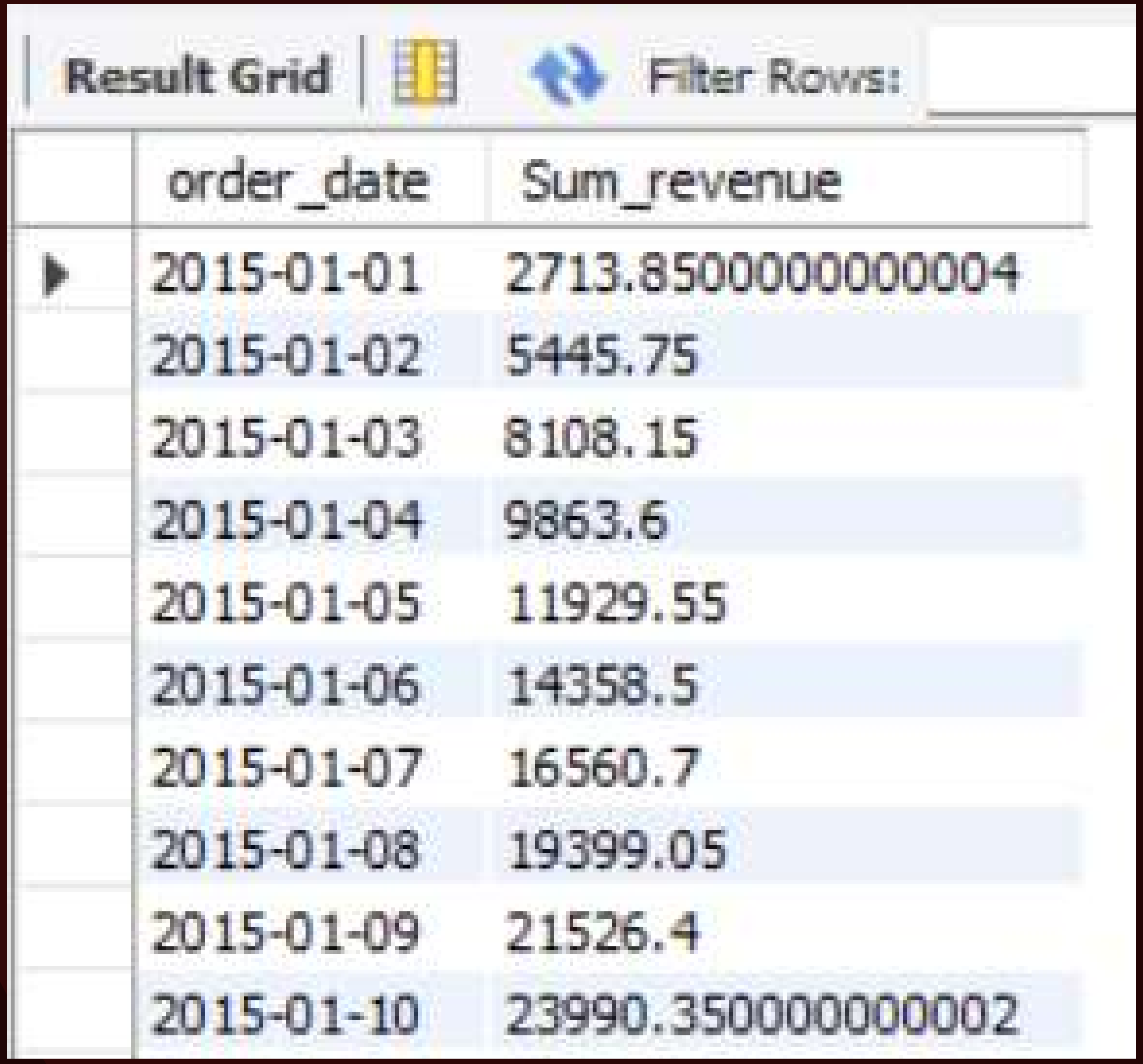


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

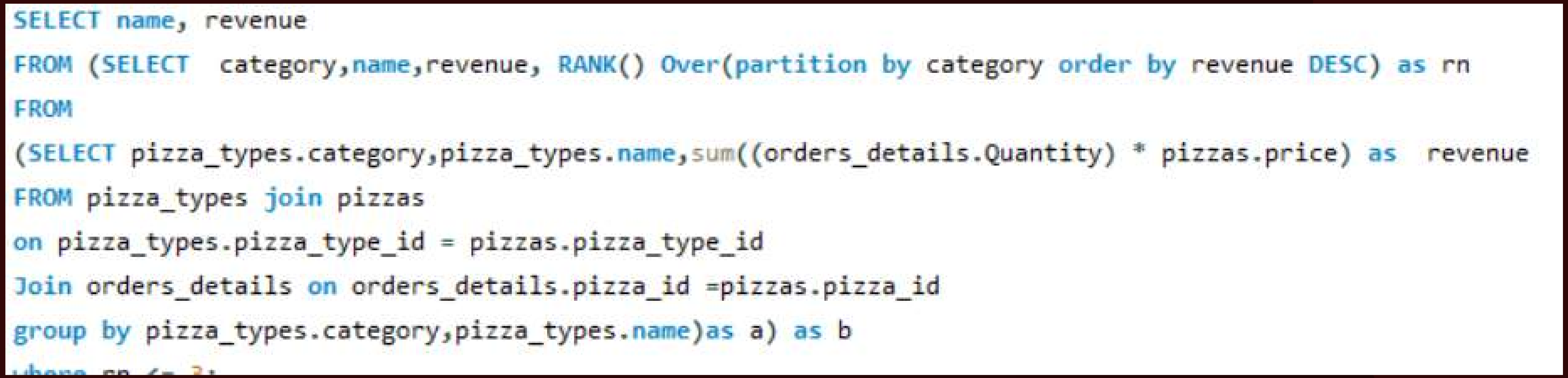


```
select order_date,
sum(revenue)  over(order by order_date) as Sum_revenue
FROM
(SELECT  orders.Order_date,
sum(orders_details.Quantity * pizzas.price) as revenue
from orders_details
JOIN pizzas on orders_details. pizza_id = pizzas.pizza_id
join orders on orders.Order_id = orders_details.Order_id
group by orders.Order_date) as  Sales;
```

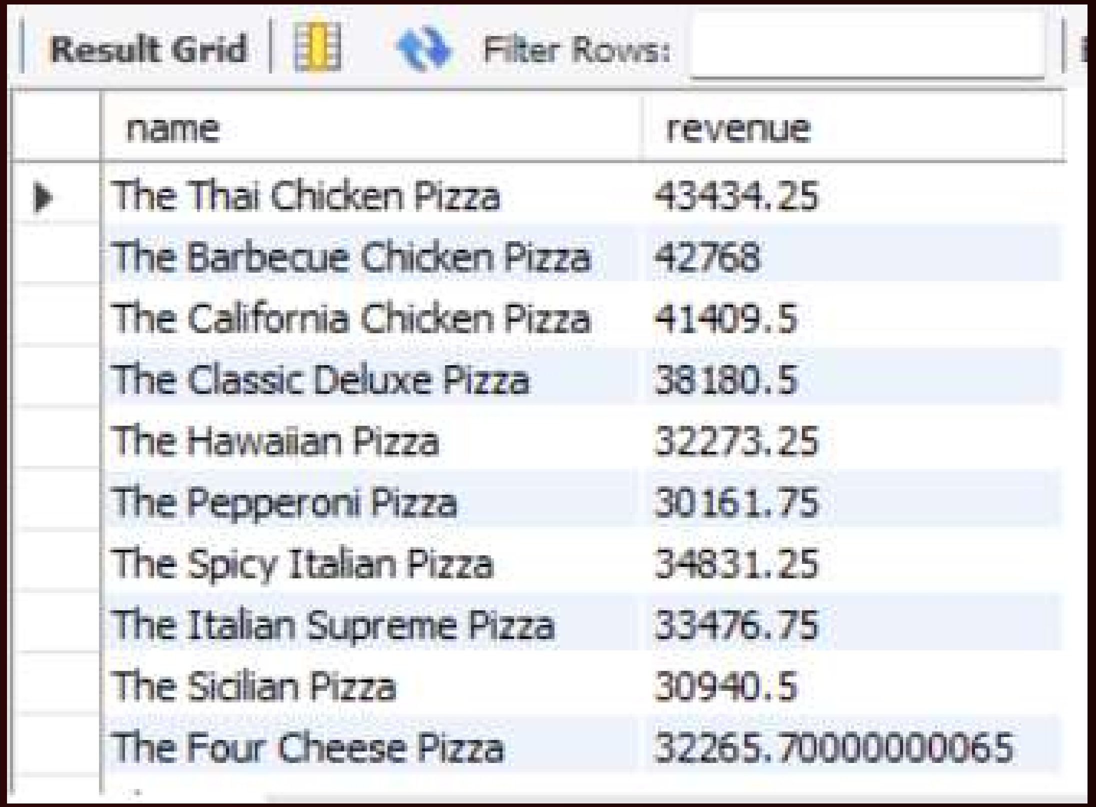


order_date	Sum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002

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



```
SELECT 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((orders_details.Quantity) * pizzas.price) as  revenue
FROM pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
Join orders_details on orders_details.pizza_id =pizzas.pizza_id
group by pizza_types.category,pizza_types.name)as a) as b
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065

THANK YOU

FOR ATTENTION