"SQL Bites: A Deep Dive into Pizza Sales Performance"

PIZZA SALES

ABOUT ME

Hello, my name is Naval Saxena, currently pursuing a PGDM in Data Science and Operations from Bangalore. I am deeply passionate about deriving insights from data and using them to solve real-world business problems. With a strong foundation in SQL, Power BI, and Python, I aim to become a skilled Data Analyst. In this project, I have analyzed a pizza sales dataset using SQL to uncover valuable insights that can help optimize sales and improve overall business performance.

PROJECT OVERVIEW

The objective of this project is to perform a detailed analysis of a pizza restaurant’s sales data using SQL. By leveraging structured queries, I aimed to identify trends, patterns, and insights that can drive better business decisions. The tools used in this project include MySQL for data analysis and GitHub for version control and showcasing the project. The dataset consists of several tables covering orders, pizzas, pizza types, and order details, enabling a comprehensive exploration of sales performance

PROBLEM STATEMENT

- Identify the most and least popular pizzas.
- Understand peak order times and days.
- Analyze revenue by pizza type and size.
- Suggest ways to improve business performance.

DATA UNDERSTANDING

orders

order_details

pizzas

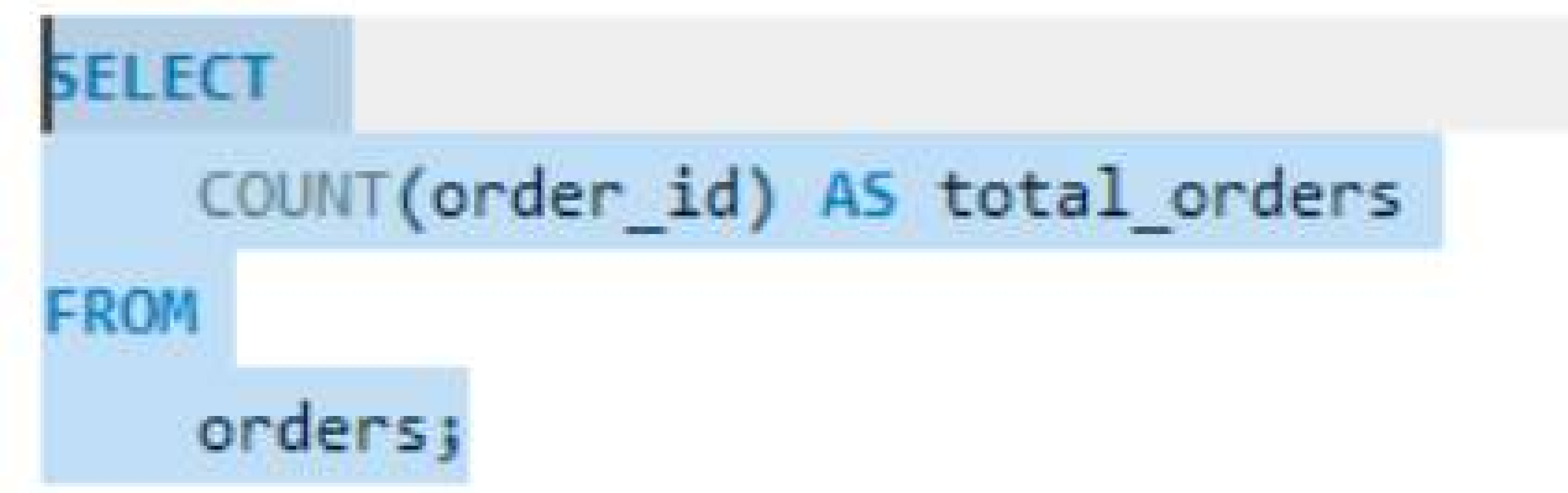
pizzas_types

DATA UNDERSTANDING

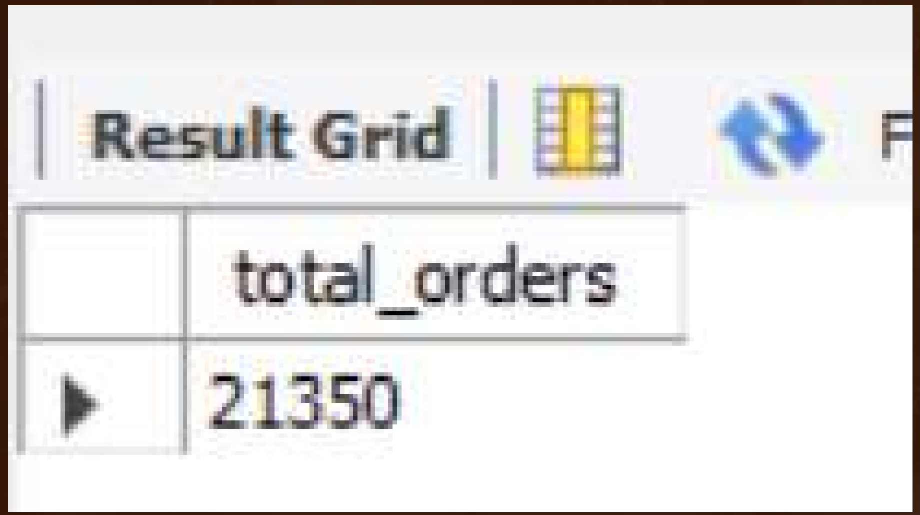
The dataset used in this project is composed of four main tables:

- The orders table contains order IDs along with date and time.
- The order_details table links each order to specific pizzas and quantities.
- The pizzas table lists all available pizzas along with their size and price.
- The pizza_types table classifies pizzas by category and ingredients.
- The data covers a full calendar year and provides enough depth to analyze customer behavior, pizza preferences, and revenue trends.

THE TOTAL NUMBER OF ORDER PLACED

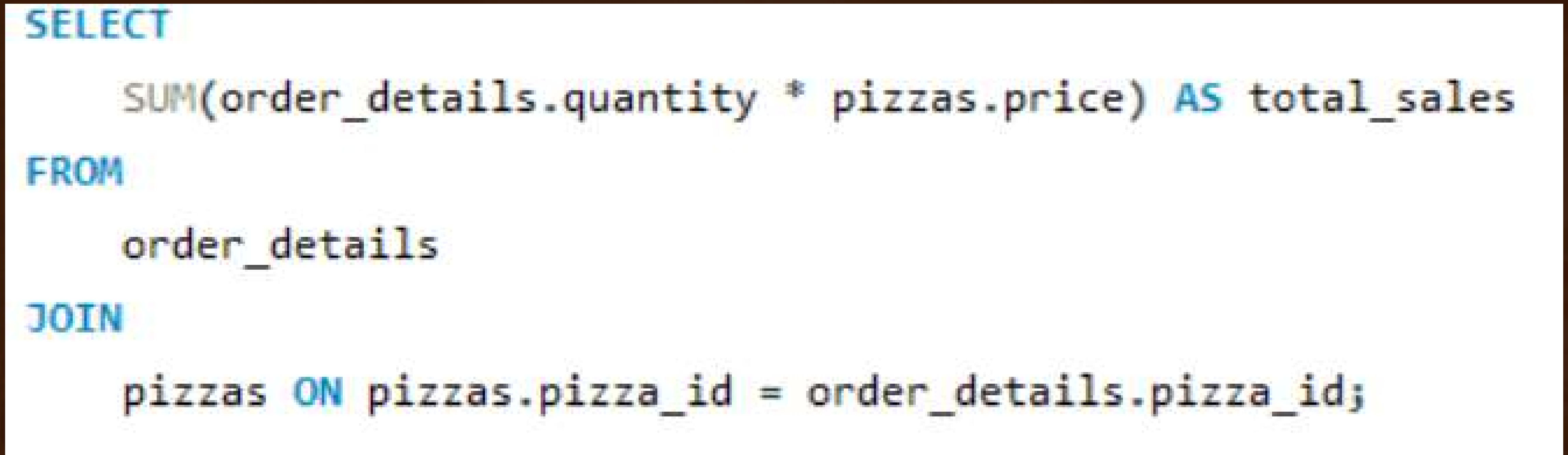


```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

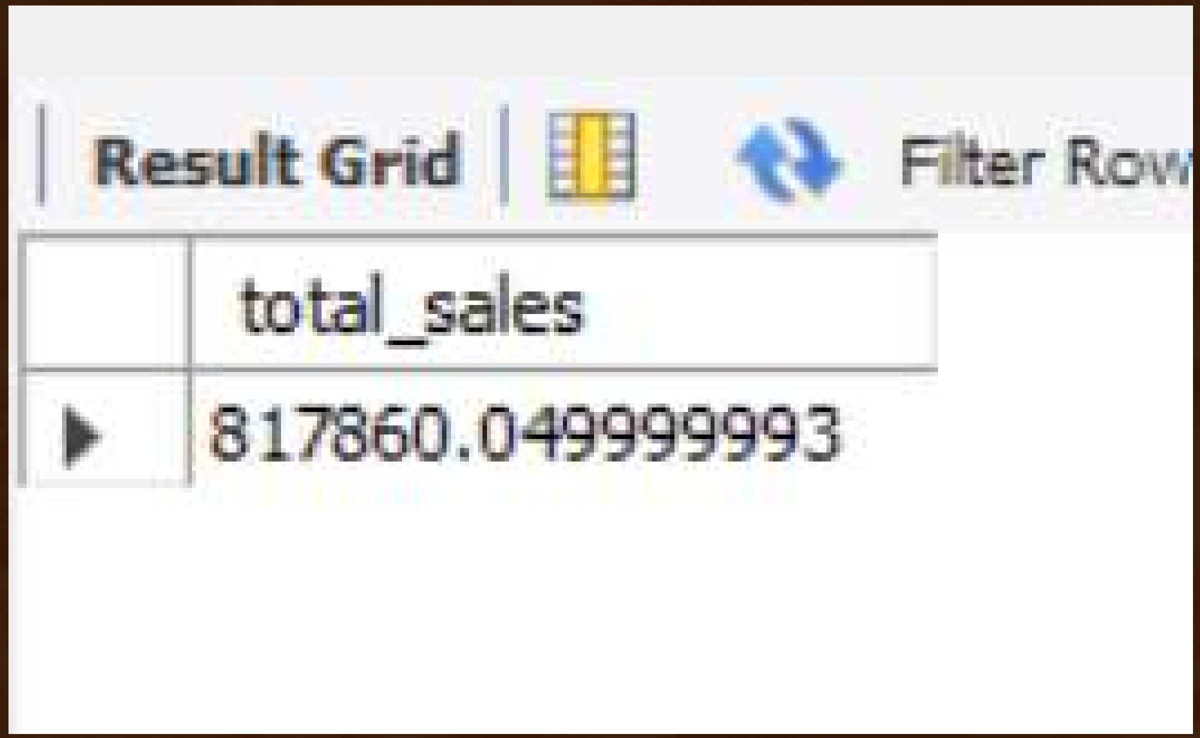


```
total_orders
21350
```

-- CALCULATE THE TOTAL REVEN8UE GENERATED FROM PIZZA SALES

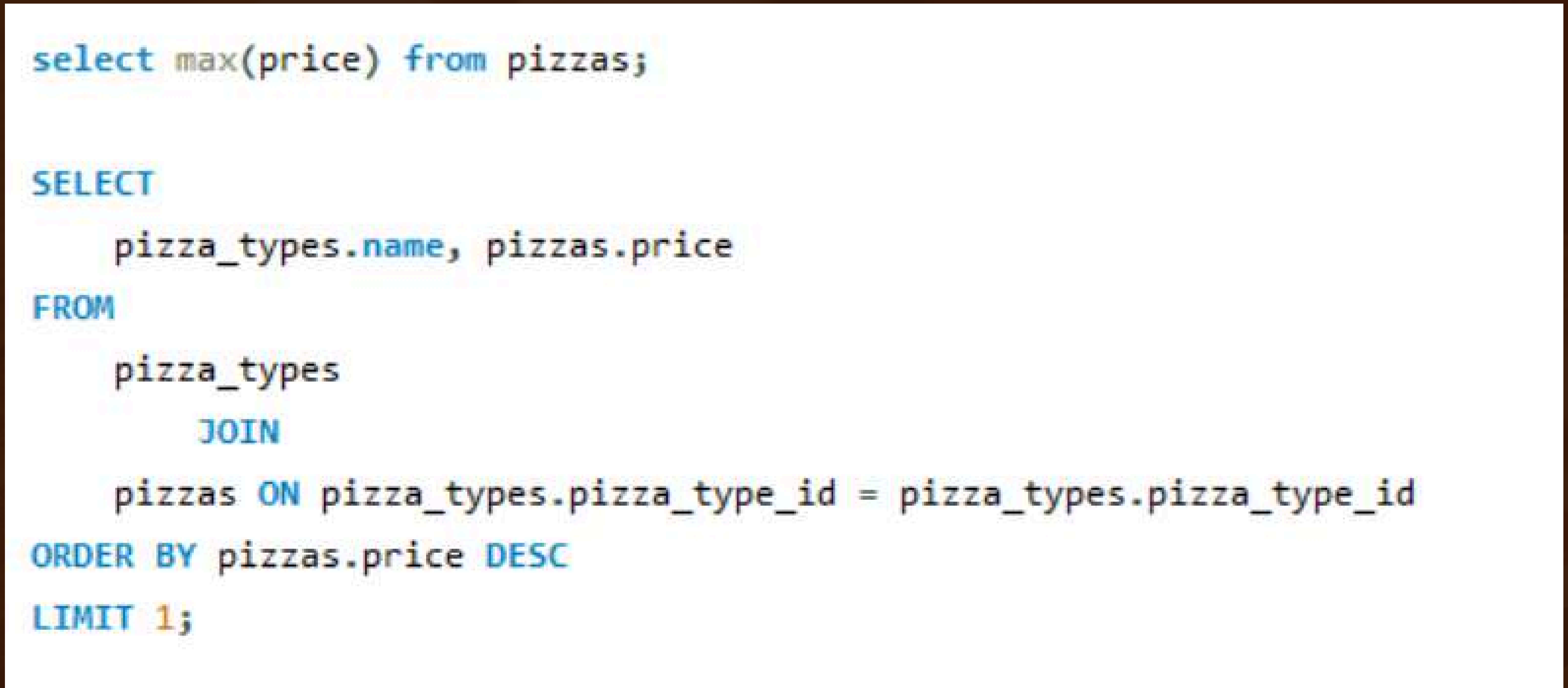


```sql
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_sales
FROM
    order_details
JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

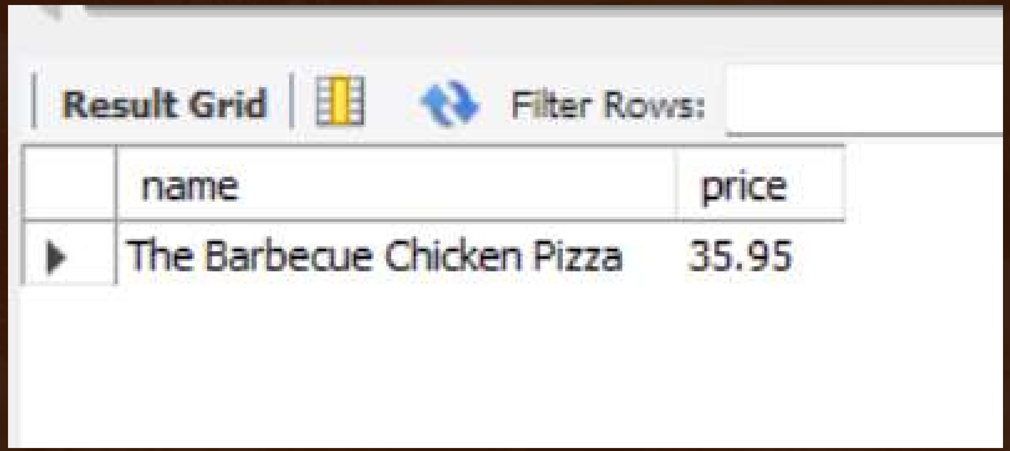


```
total_sales
817860.049999993
```

HIGHEST PIZZA PRICE

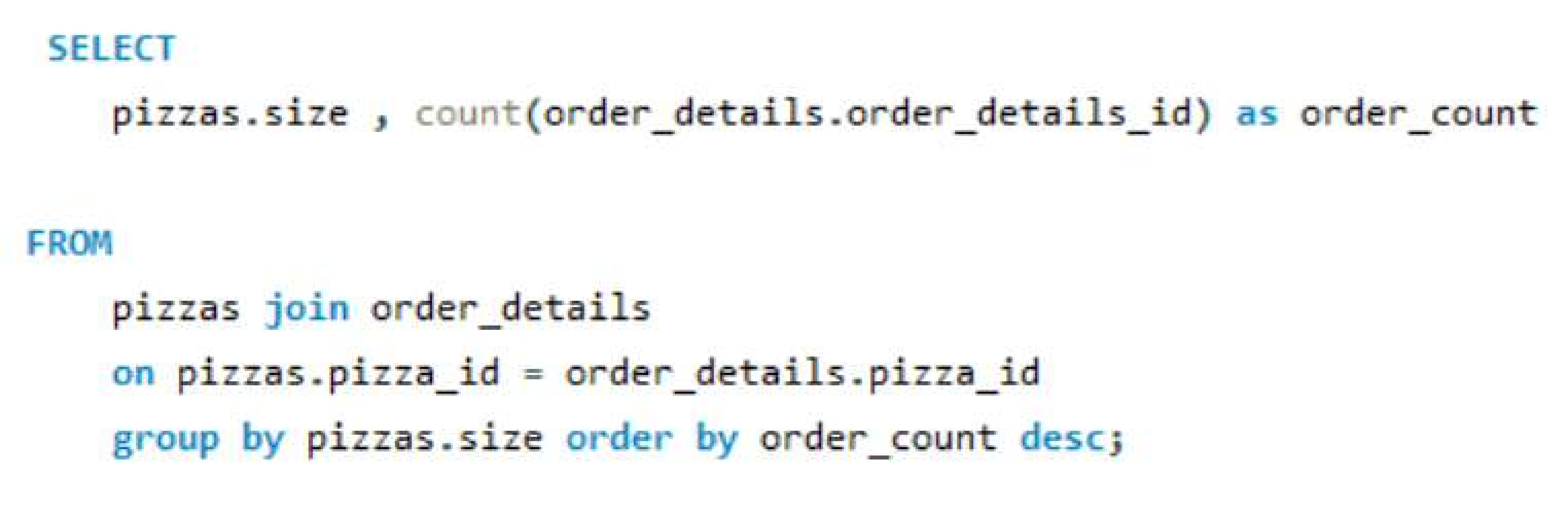


```
select max(price) from pizzas;


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

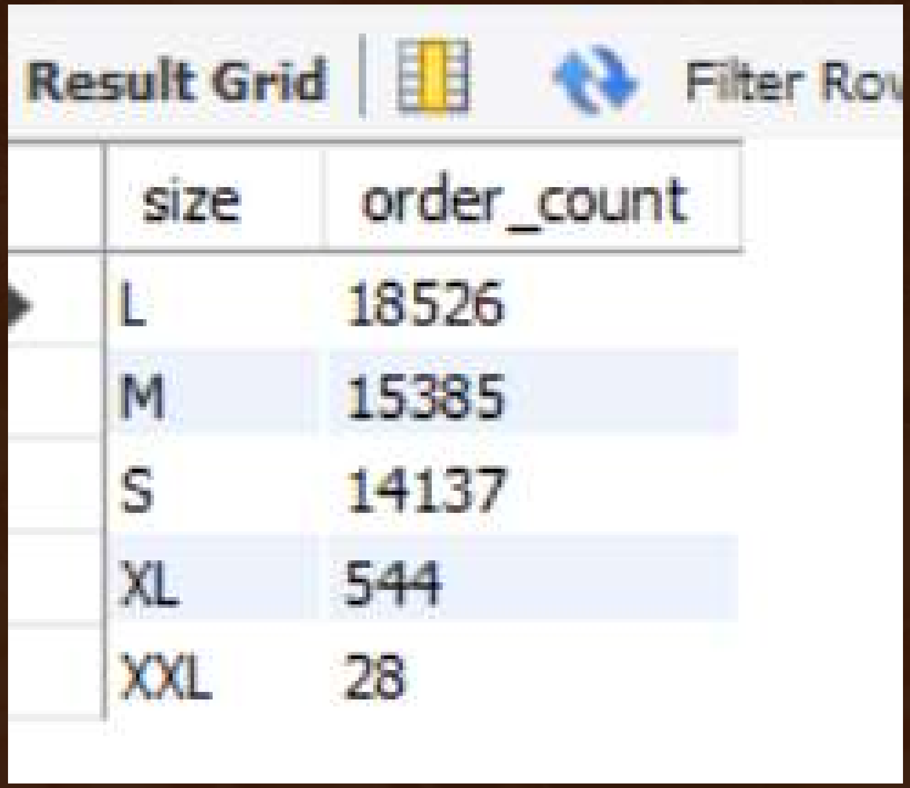


name	price
The Barbecue Chicken Pizza	35.95

MOST ORDER SIZE COUNT

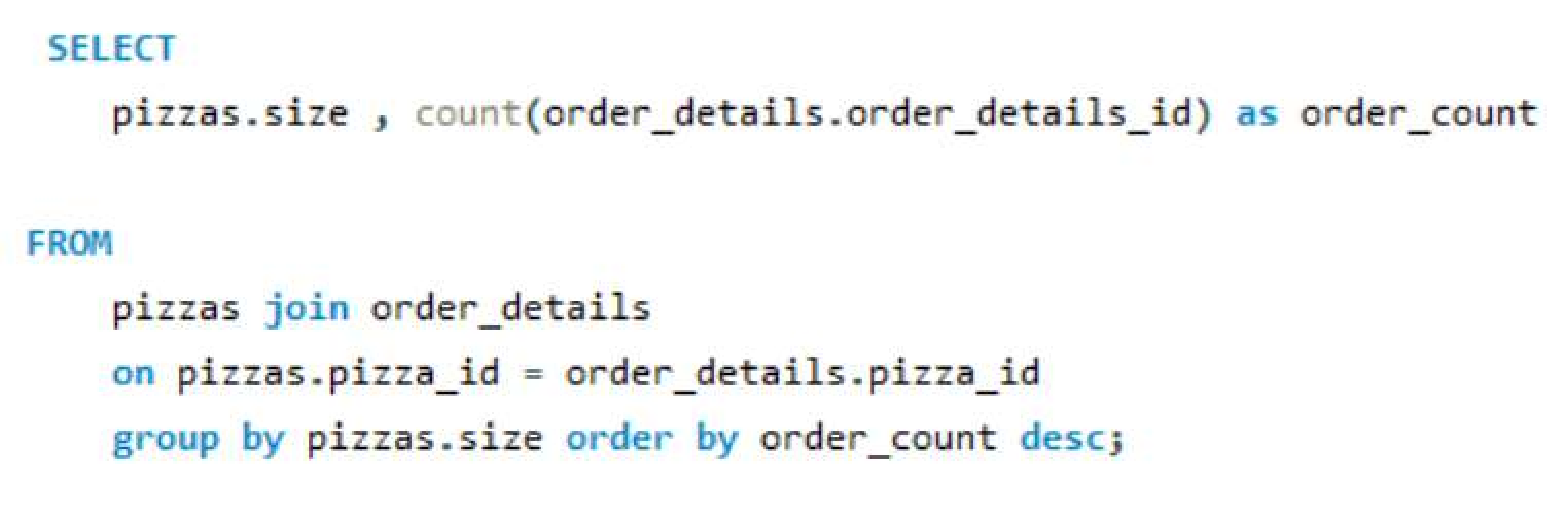


```
 SELECT
    pizzas.size , count(order_details.order_details_id) as order_count

FROM
    pizzas join order_details
    on pizzas.pizza_id = order_details.pizza_id
    group by pizzas.size order by order_count desc;
```

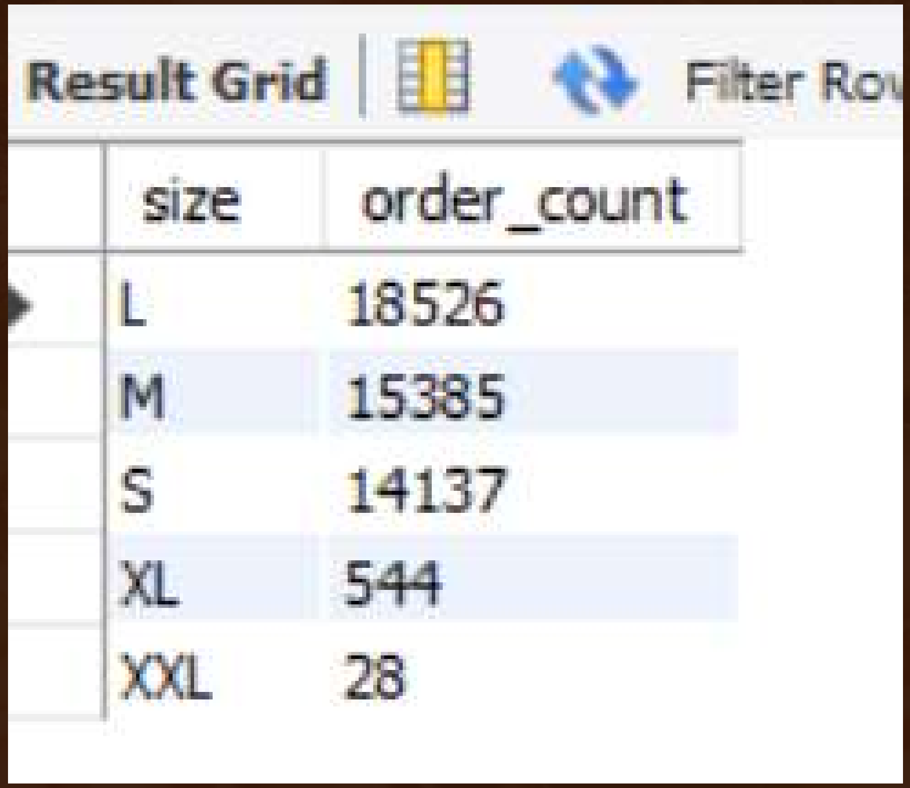


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

HIGHEST PIZZA PRICE

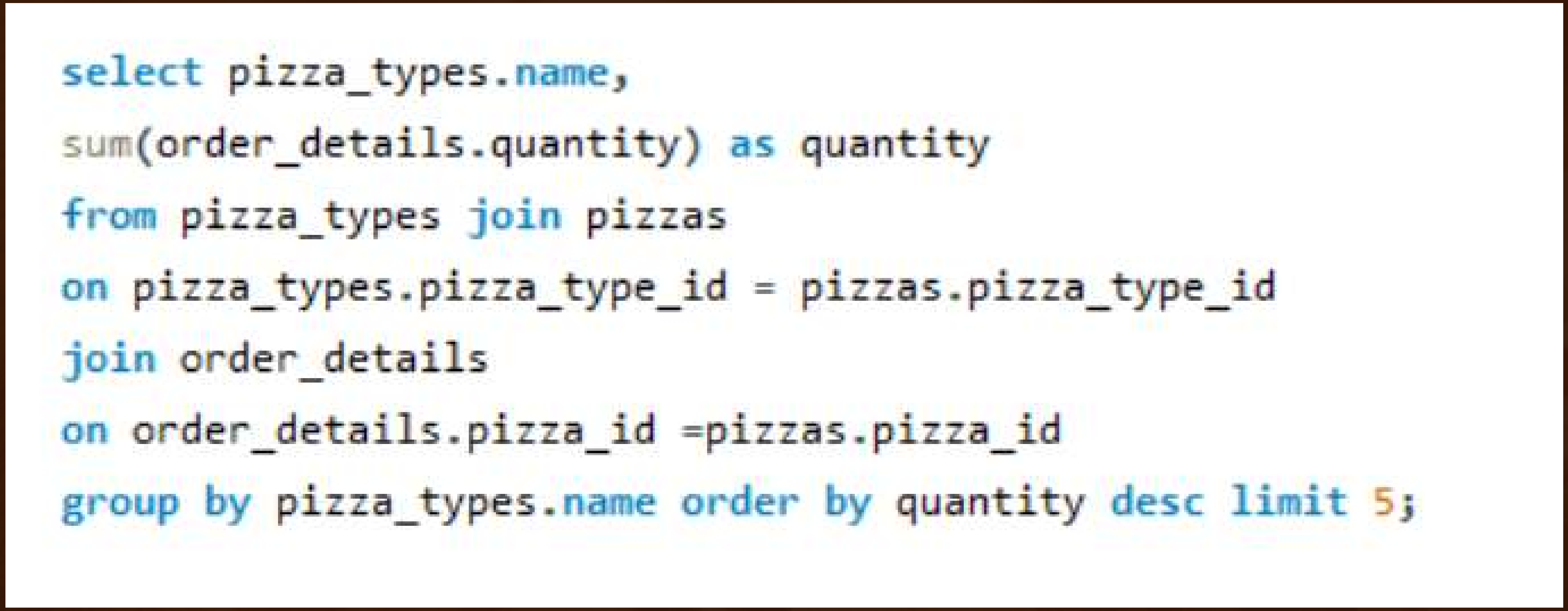


```
 SELECT
    pizzas.size , count(order_details.order_details_id) as order_count

FROM
    pizzas join order_details
    on pizzas.pizza_id = order_details.pizza_id
    group by pizzas.size order by order_count desc;
```

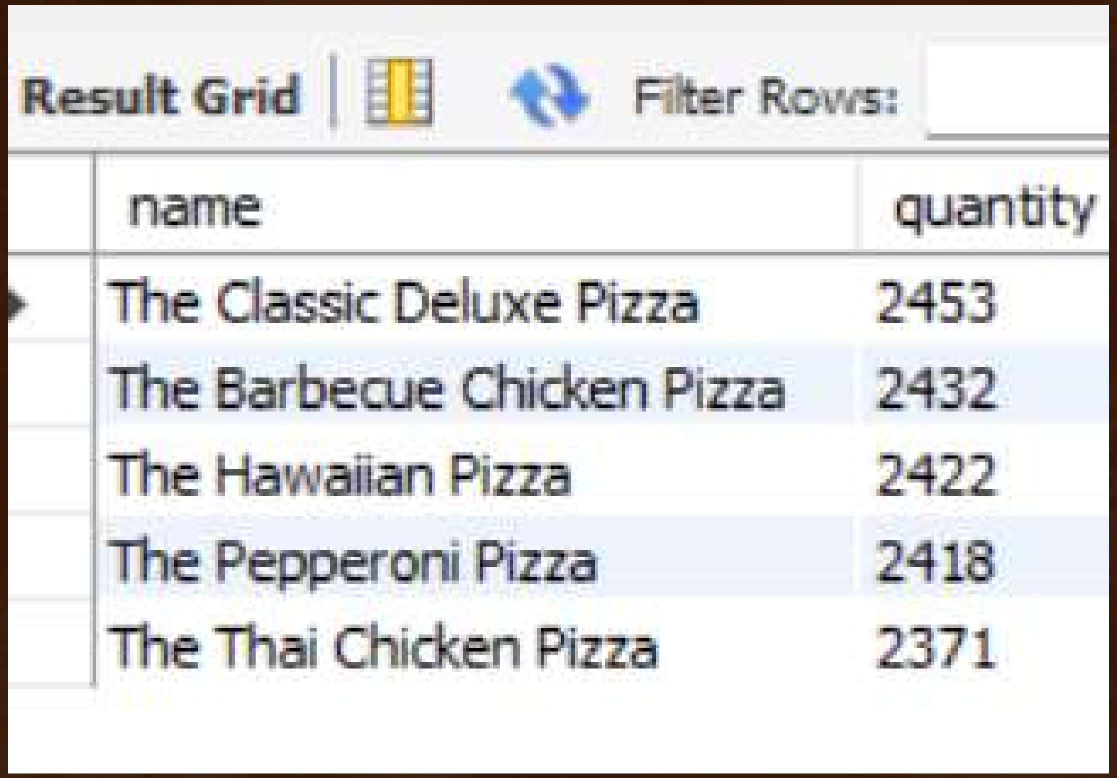


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

HIGHEST PIZZALIST THE TOP 5 ORDERED PIZZA TYPES ALONG WITH THEIR QUNTITIESPRICE

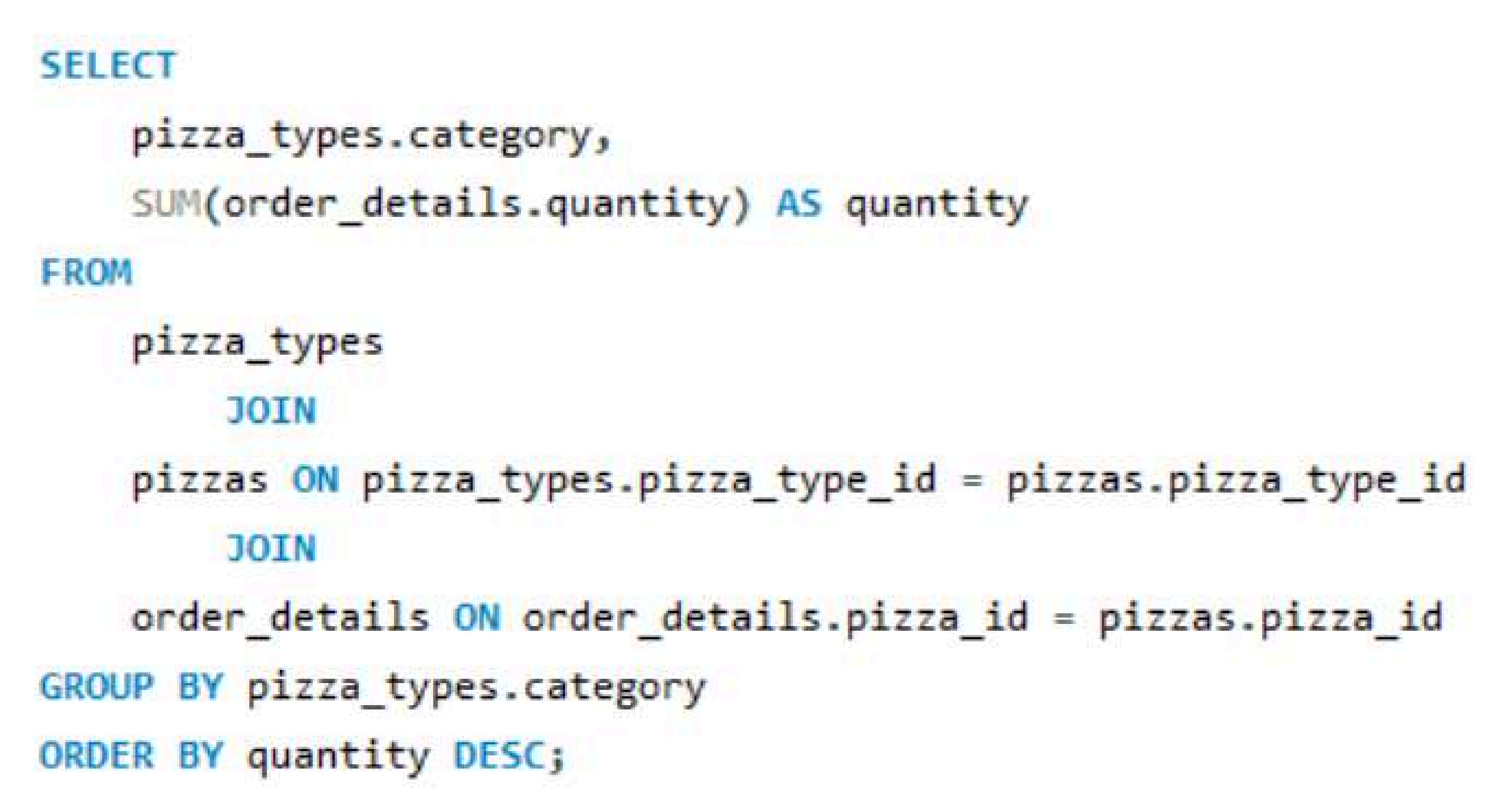


```
select pizza_types.name,
sum(order_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id =pizzas.pizza_id
group by pizza_types.name order by quantity desc limit 5;
```

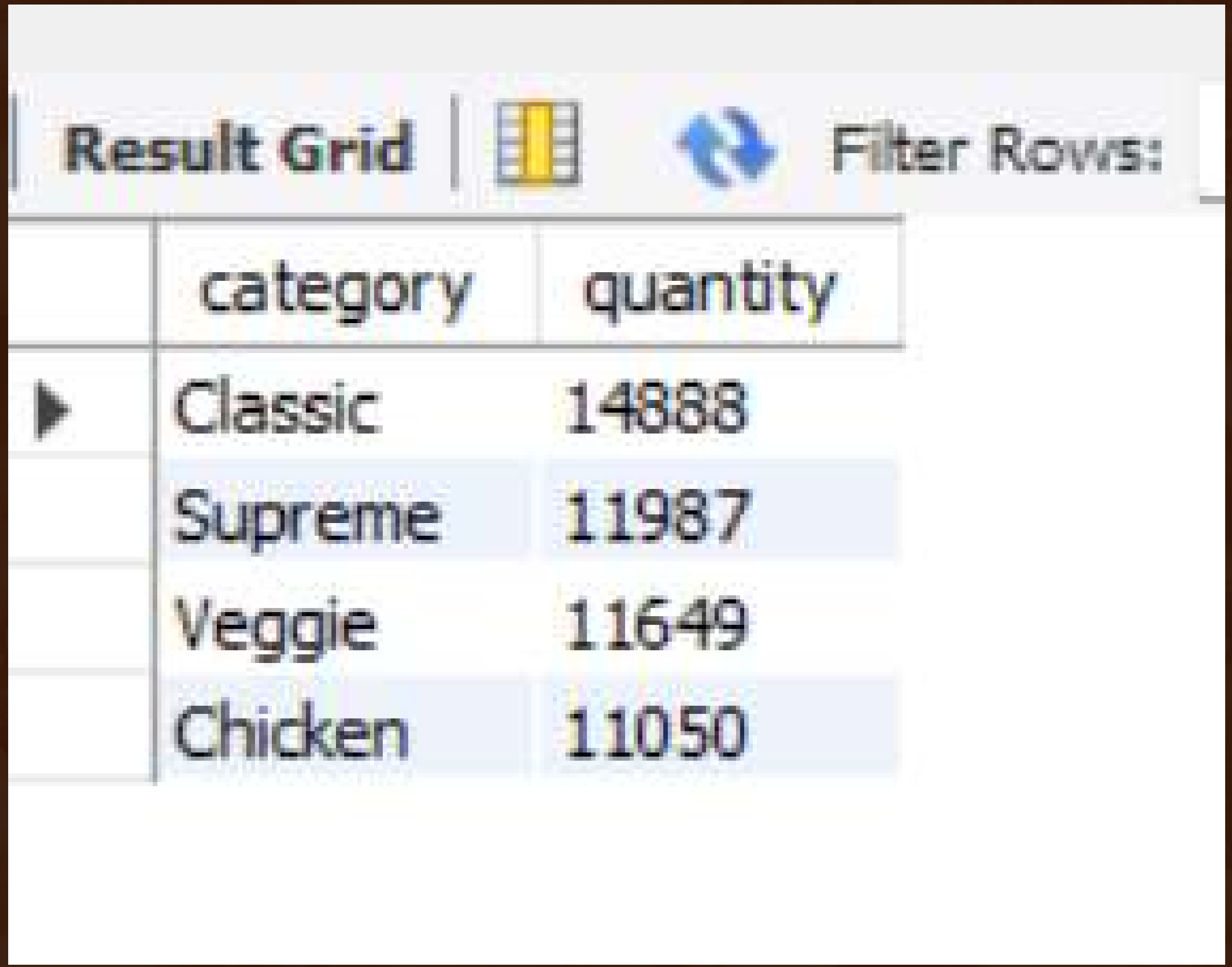


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

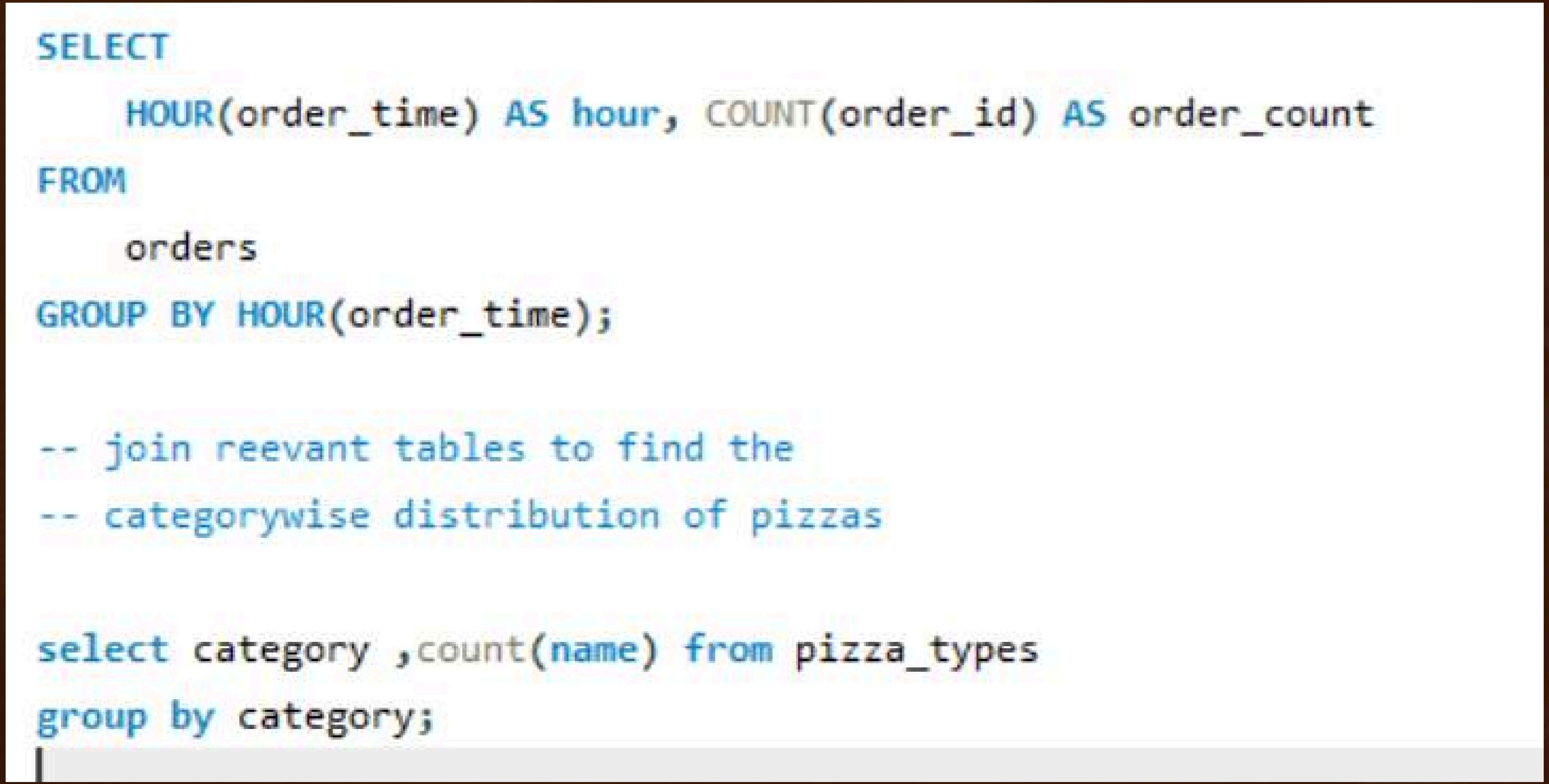


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

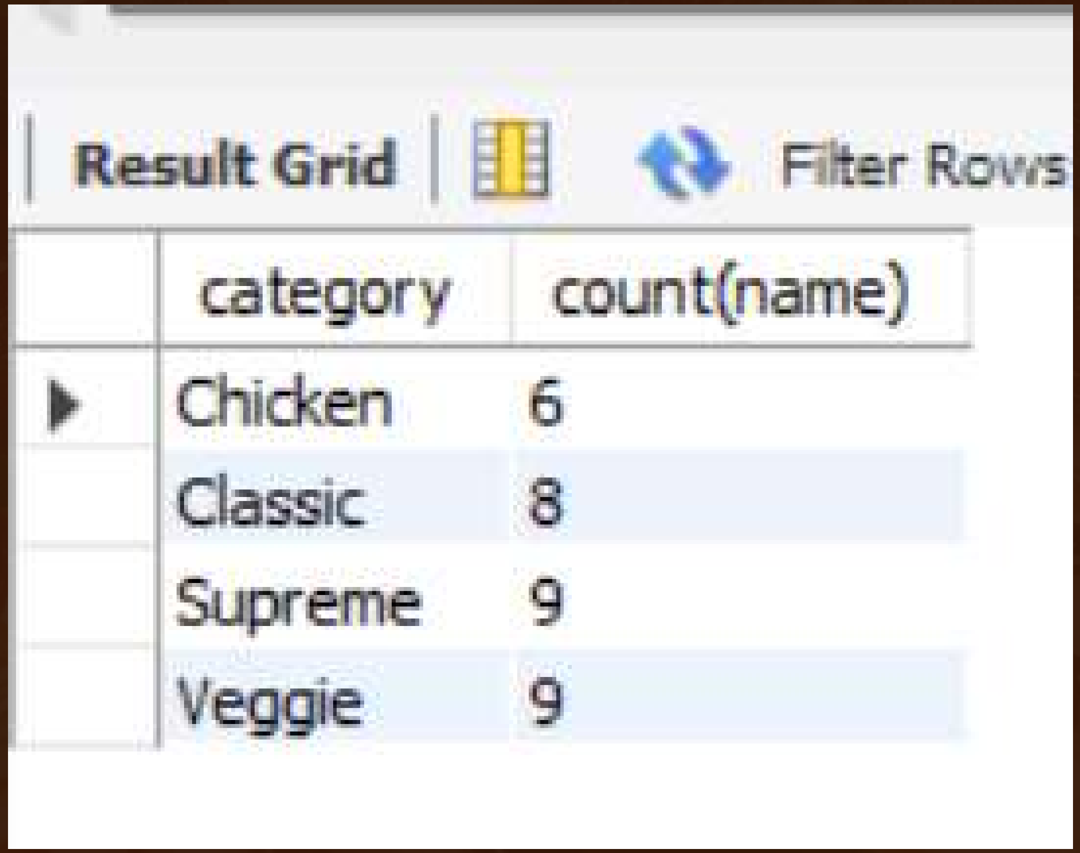


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

-- HOUR BASIS ORDER DISTRIBUTIOND

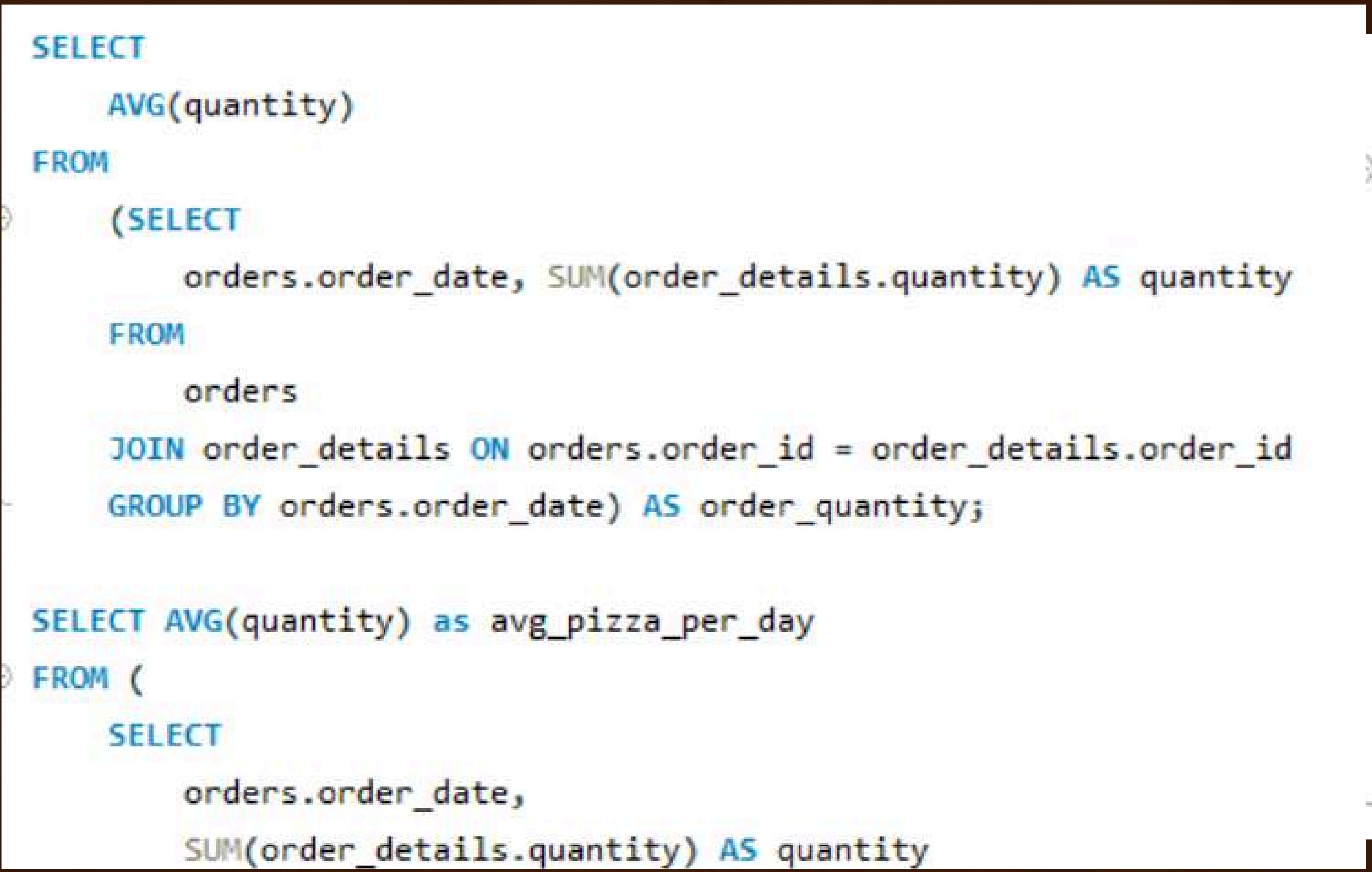


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);

-- join reevant tables to find the
-- categorywise distribution of pizzas

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

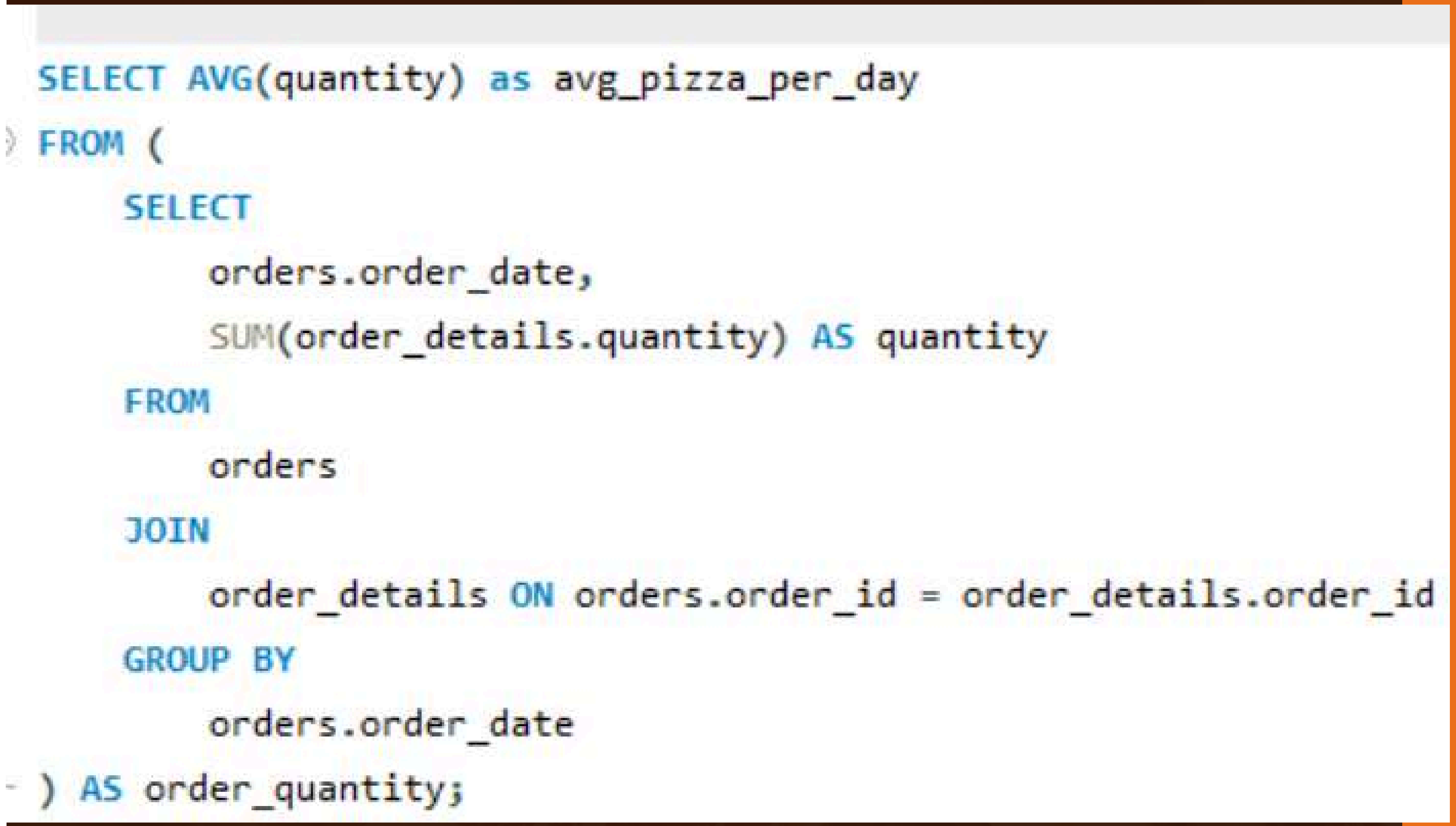


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

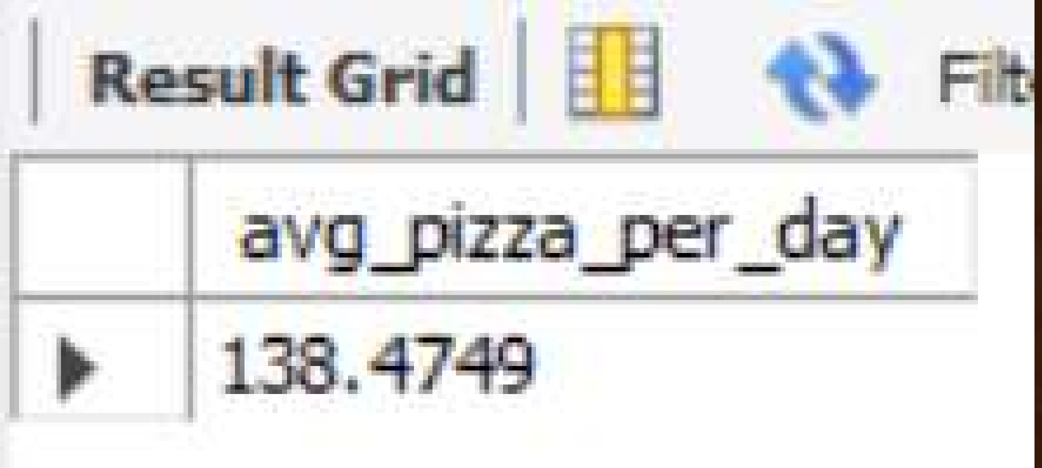
GROUP THE ORDERS BY DATE AND CALCUKATE THE AVERAGE -- NUMBER OF PIZZAS ORDERED PER DAY



```
SELECT
    AVG(quantity)
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;

SELECT AVG(quantity) as avg_pizza_per_day
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity) AS quantity
```

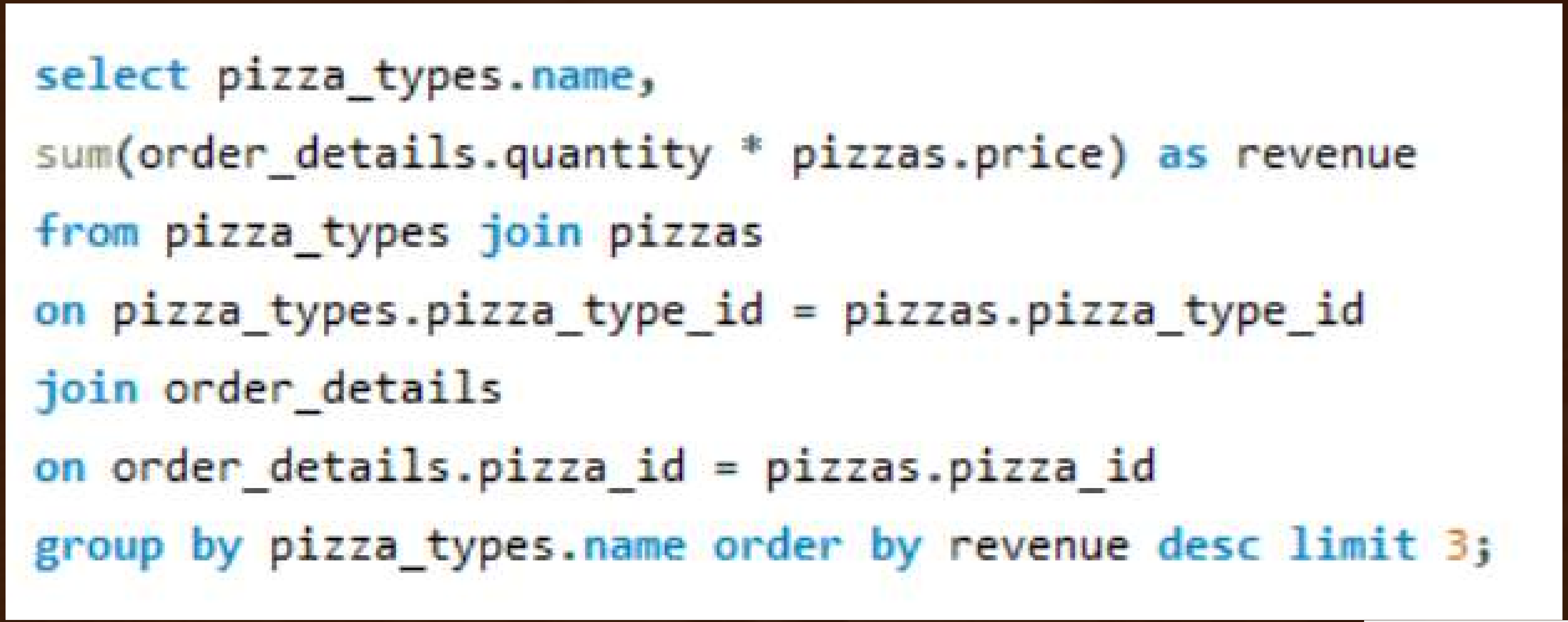


```
SELECT AVG(quantity) as avg_pizza_per_day
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN
        order_details ON orders.order_id = order_details.order_id
    GROUP BY
        orders.order_date
) AS order_quantity;
```

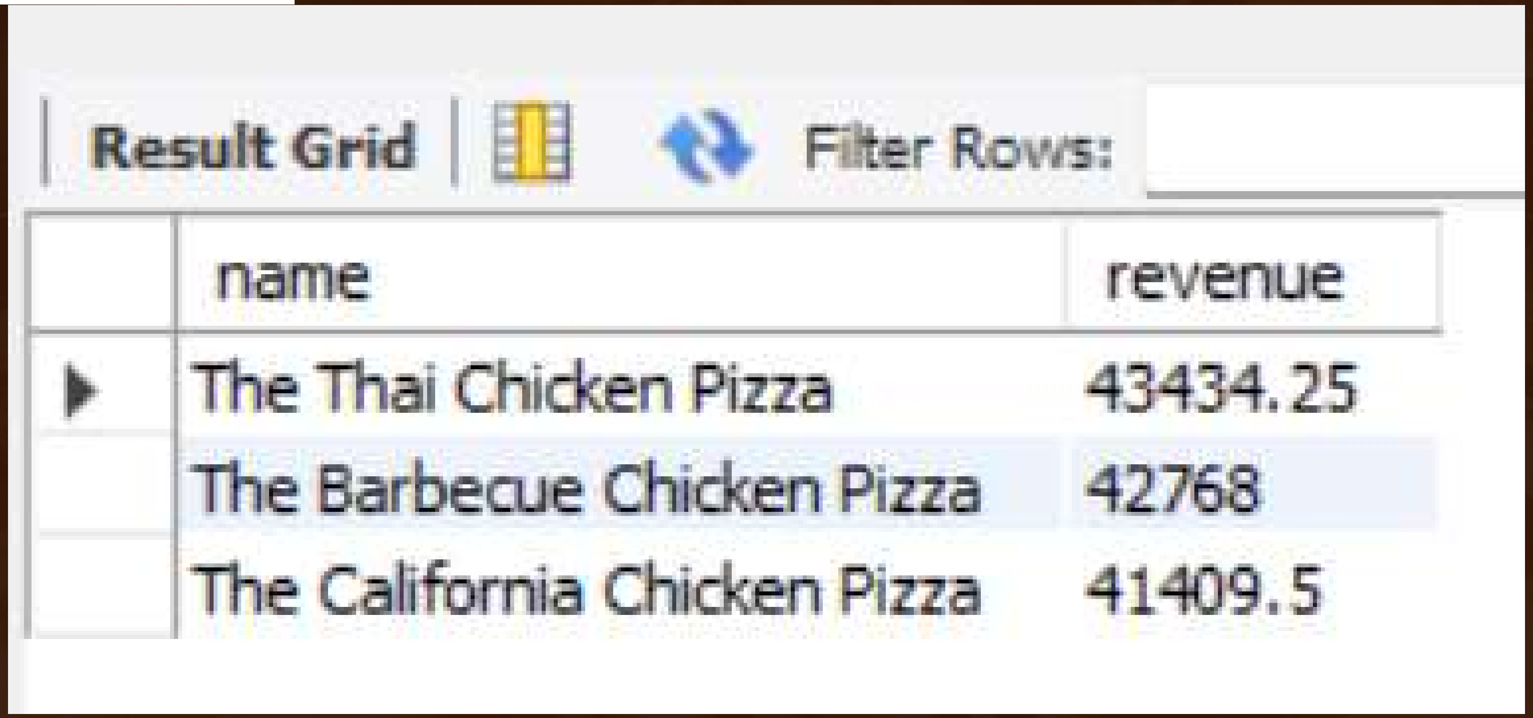


	avg_pizza_per_day
▶	138.4749

- DETERMINE THE TOP THREE 3 PIZZAS ORDERED BASED ON. REVENUE

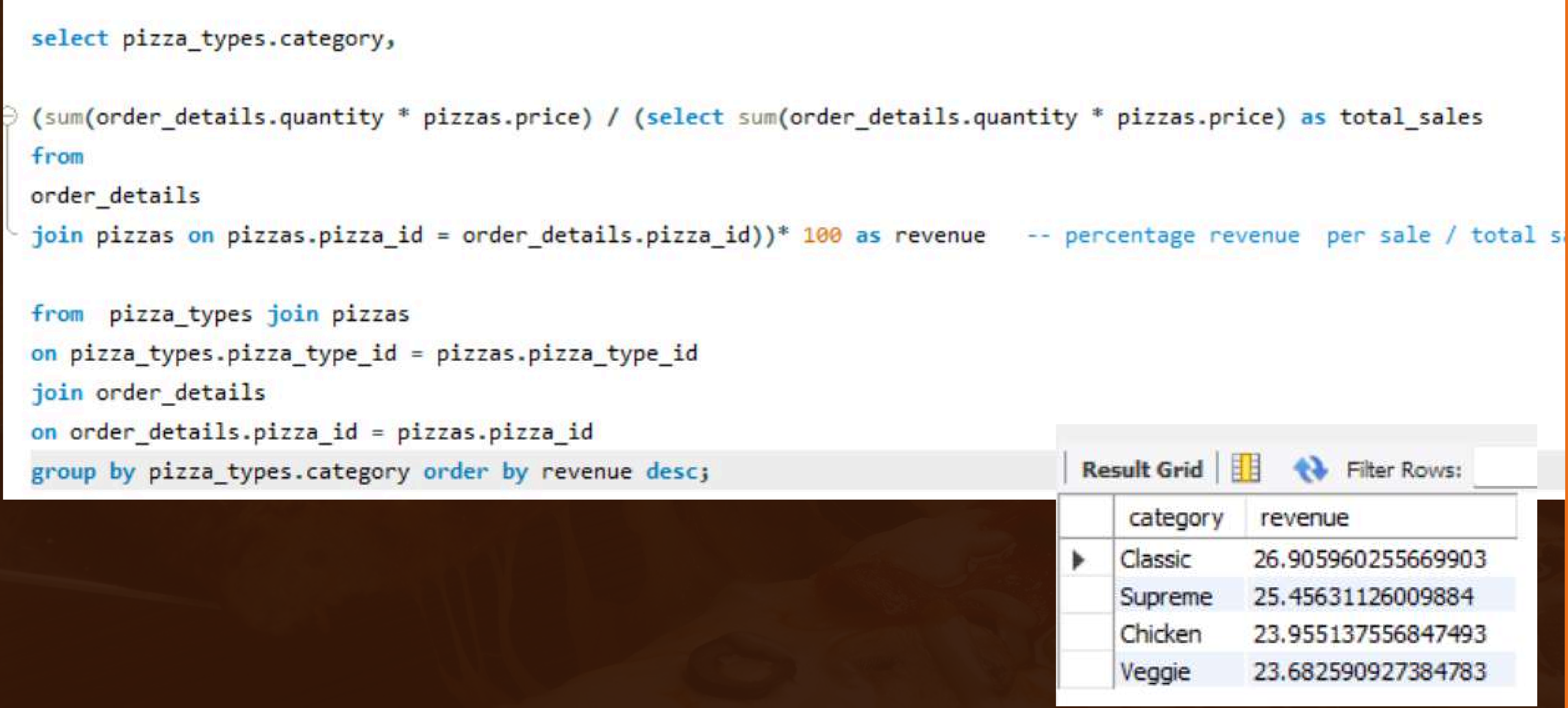


```sql
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```



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

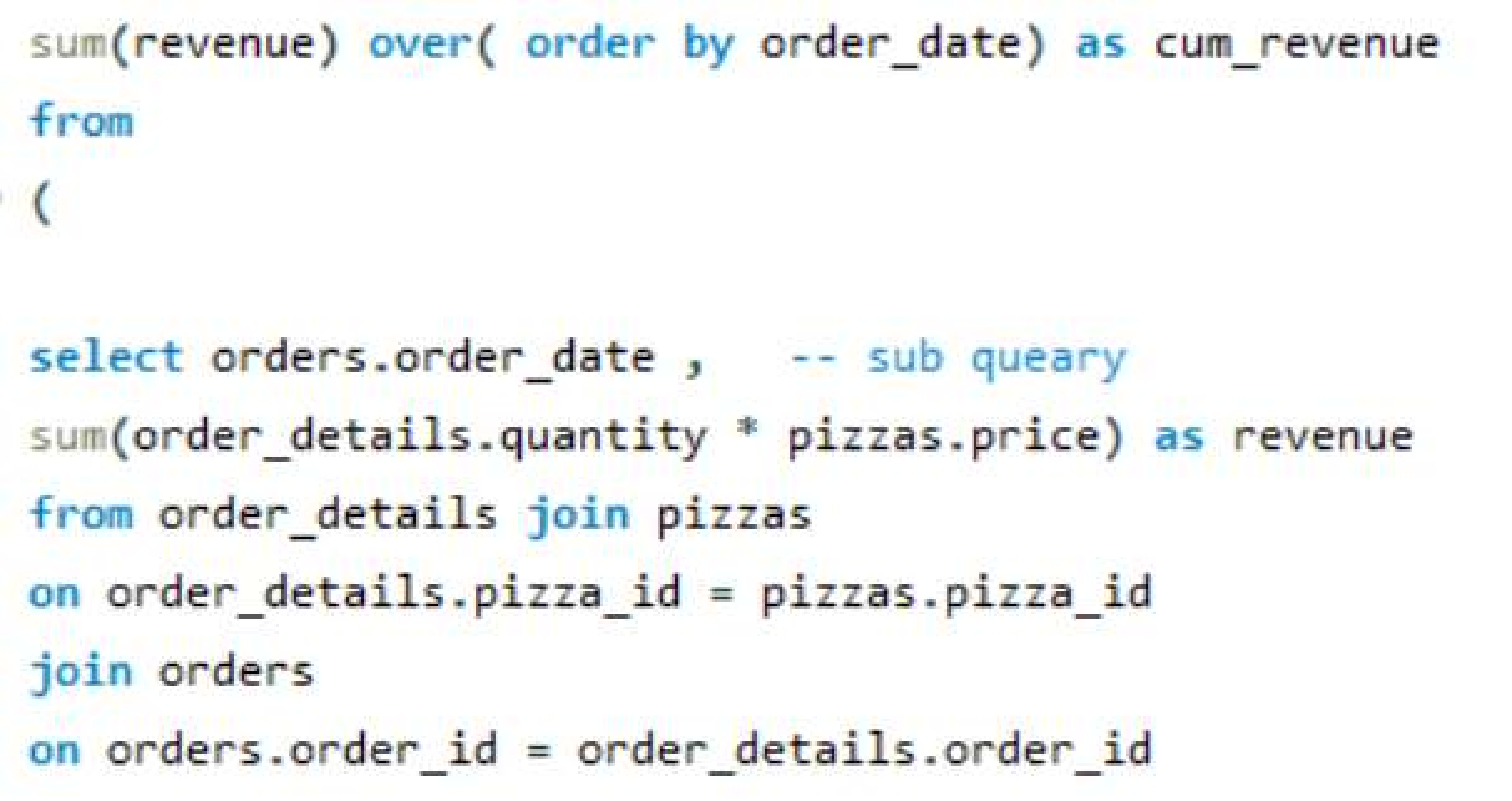
-- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH
-- PIZZA TYPE TO TOTAL REVENUE



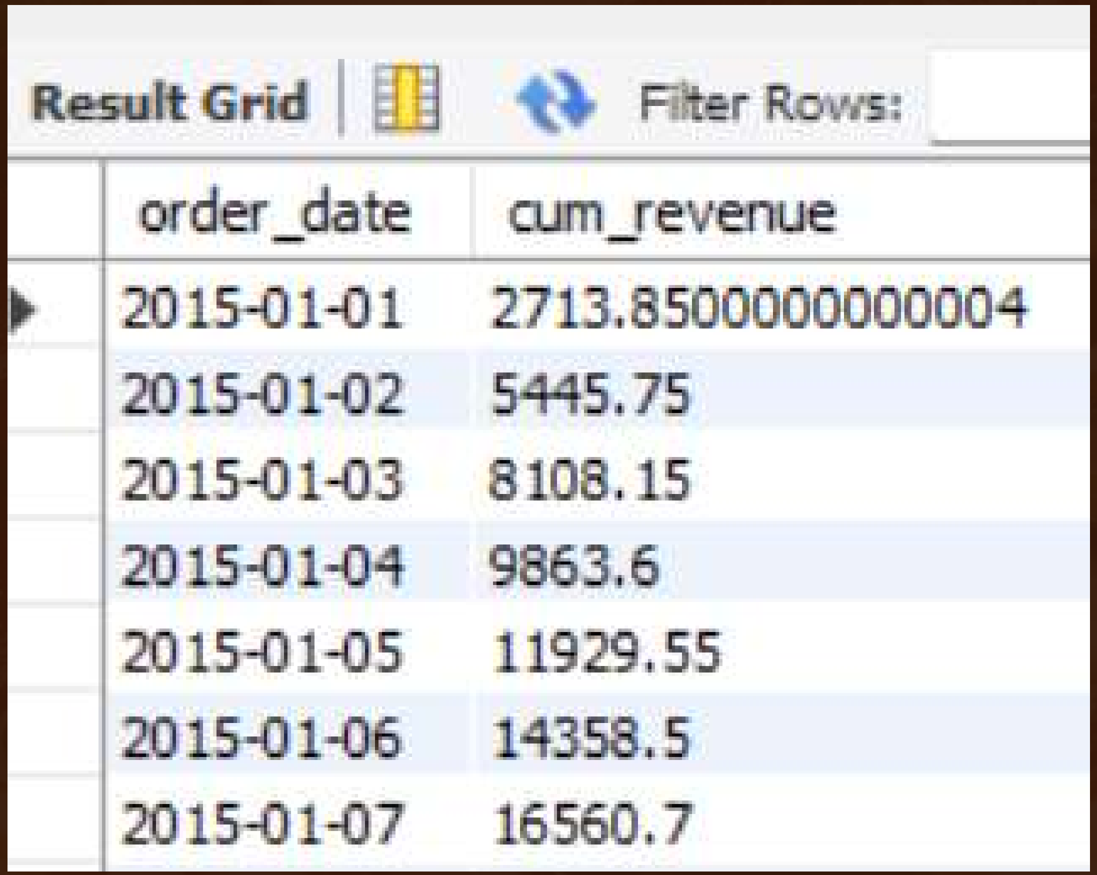
```
select pizza_types.category,

(sum(order_details.quantity * pizzas.price) / (select sum(order_details.quantity * pizzas.price) as total_sales
from
order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id))* 100 as revenue   -- percentage revenue  per sale / total s

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	26.905960255669903
Supreme	25.45631126009884
Chicken	23.955137556847493
Veggie	23.682590927384783

-- ANALYZE THE CUMULATIVE



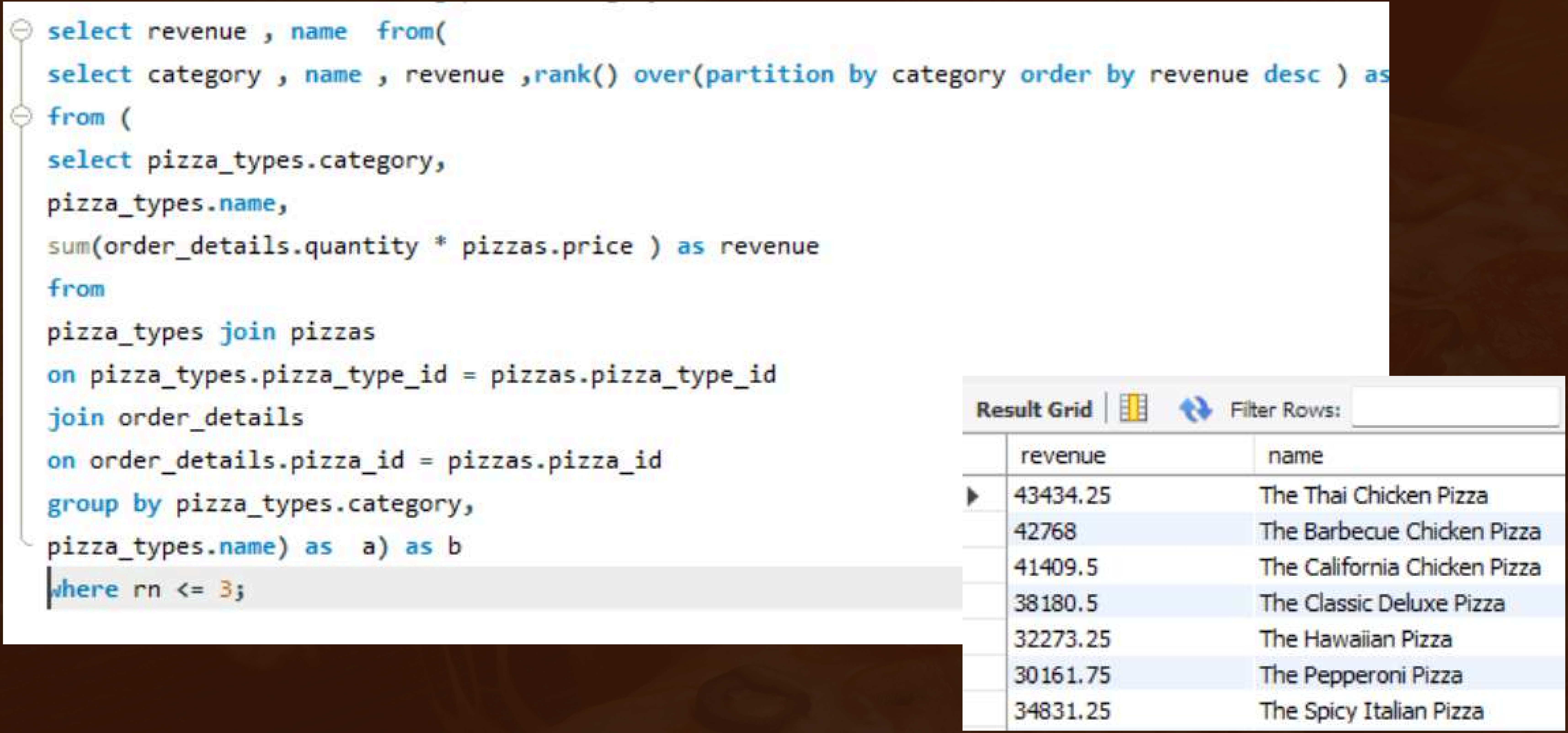
```sql
sum(revenue) over( order by order_date) as cum_revenue
from
(

select orders.order_date ,    -- sub queary
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
```



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

-- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES

-- BASED ON REVENUE FOR EACG PIZZA CATEGORY



```
select revenue , name  from(
select category , name , revenue ,rank() over(partition by category order by revenue desc ) as
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;
```

revenue	name
43434.25	The Thai Chicken Pizza
42768	The Barbecue Chicken Pizza
41409.5	The California Chicken Pizza
38180.5	The Classic Deluxe Pizza
32273.25	The Hawaiian Pizza
30161.75	The Pepperoni Pizza
34831.25	The Spicy Italian Pizza

CONCLUSION

In conclusion, SQL can be an extremely powerful tool for deriving insights from business data. This project demonstrated how structured queries can uncover trends that support better decision-making in areas such as product management, marketing, and operations. Moving forward, this analysis can be expanded by integrating tools like Power BI or Tableau for real-time dashboards, or by diving deeper into customer segmentation for personalized offers.

THANK YOU FOR ATTENTION