SQL Data Analysis Project: Pizza Sales Insights

Presented by: Sahil Alam

Mail :- official.sahilalam@gmail.com

Project Overview

📂 Dataset Description:

The dataset contains detailed records of customer orders, pizza types, prices, and sales across multiple days. It includes tables such as:

- orders
- order_details
- pizzas
- pizza_types

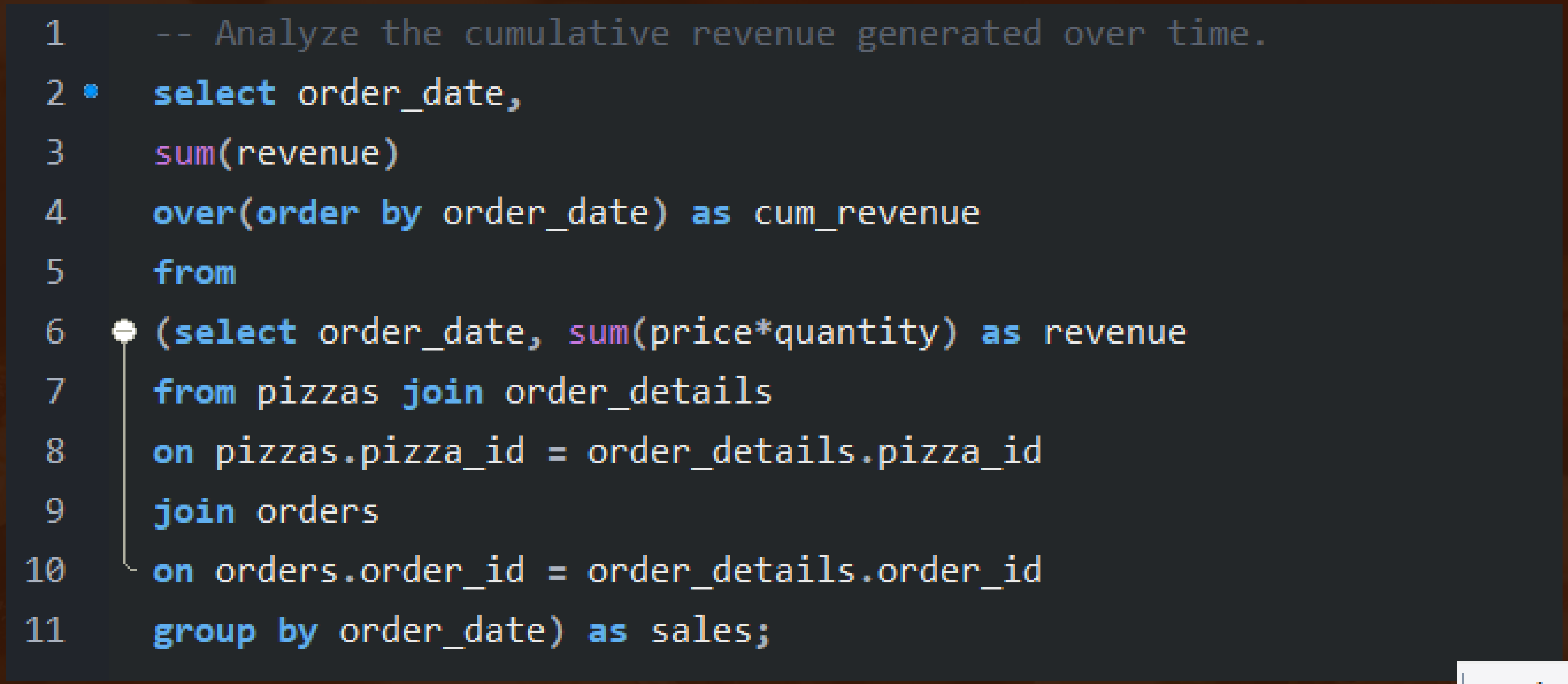
🎯 Objective:

To extract valuable business insights from raw sales data using SQL queries. The analysis helps identify trends, performance, and improvement areas for the business.

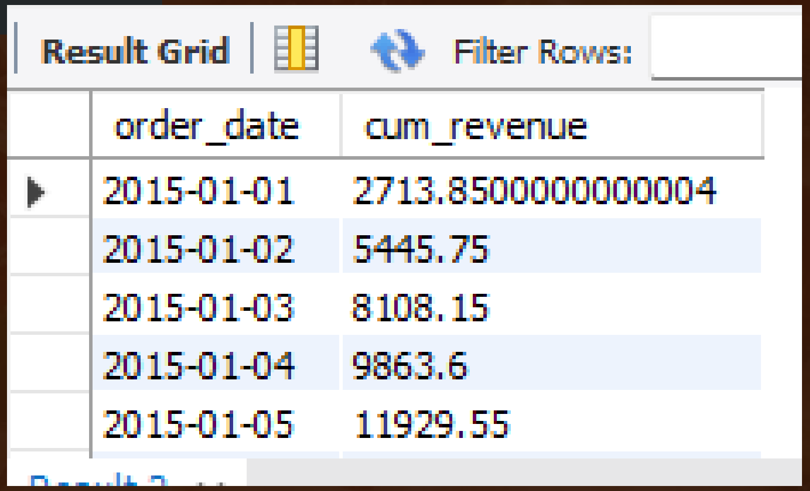
Problem Statements

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

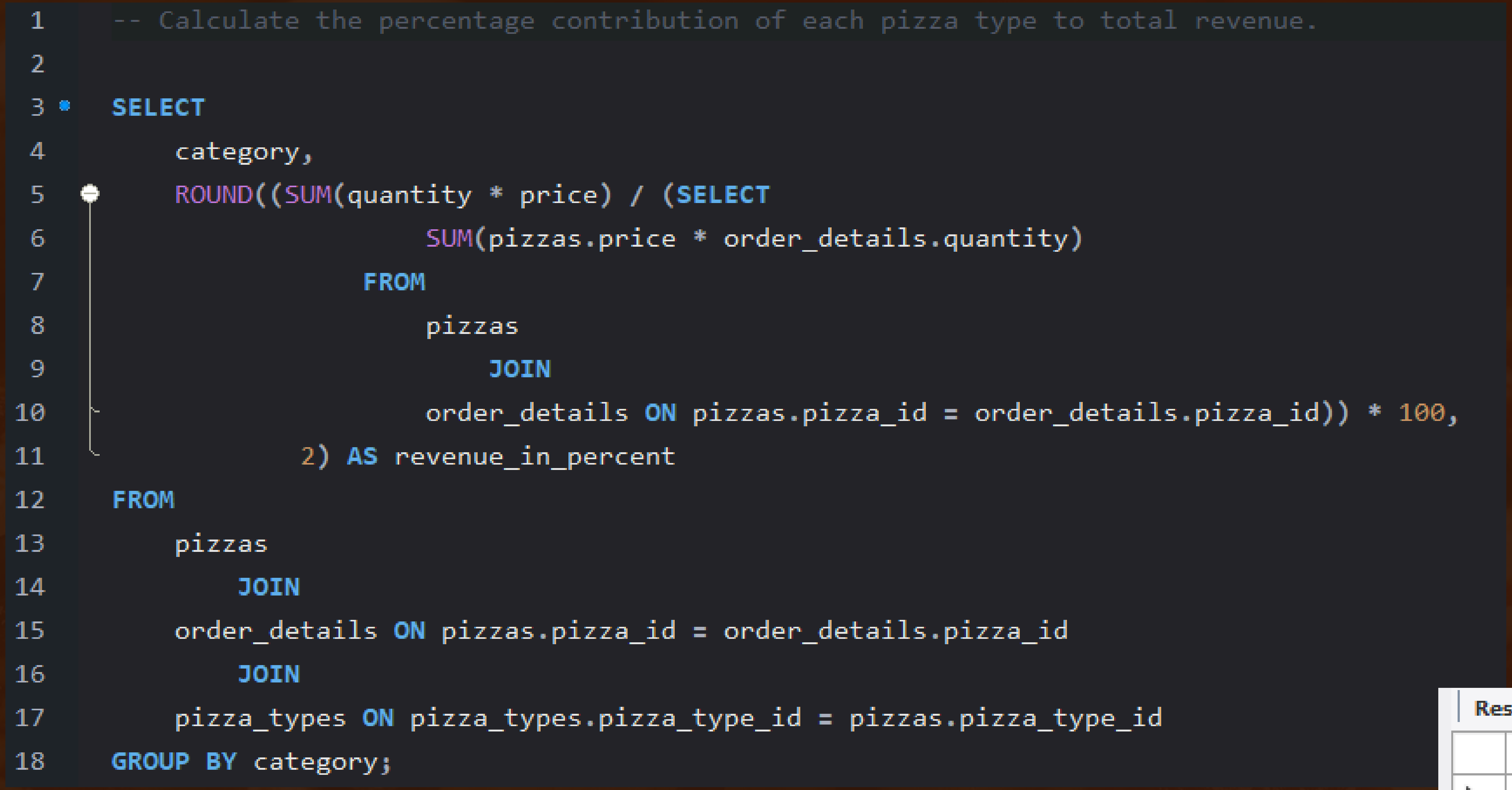


```sql
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue)
over(order by order_date) as cum_revenue
from
(select order_date, sum(price*quantity) as revenue
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_id = order_details.order_id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



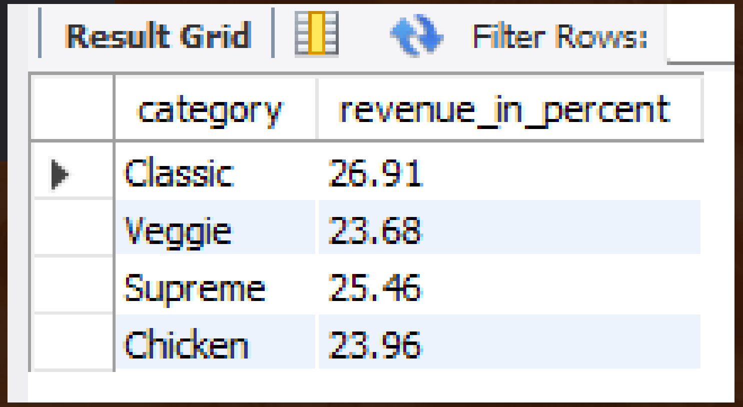
```
order_date    cum_revenue
2015-01-01    2713.8500000000004
2015-01-02    5445.75
2015-01-03    8108.15
2015-01-04    9863.6
2015-01-05    11929.55
```

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



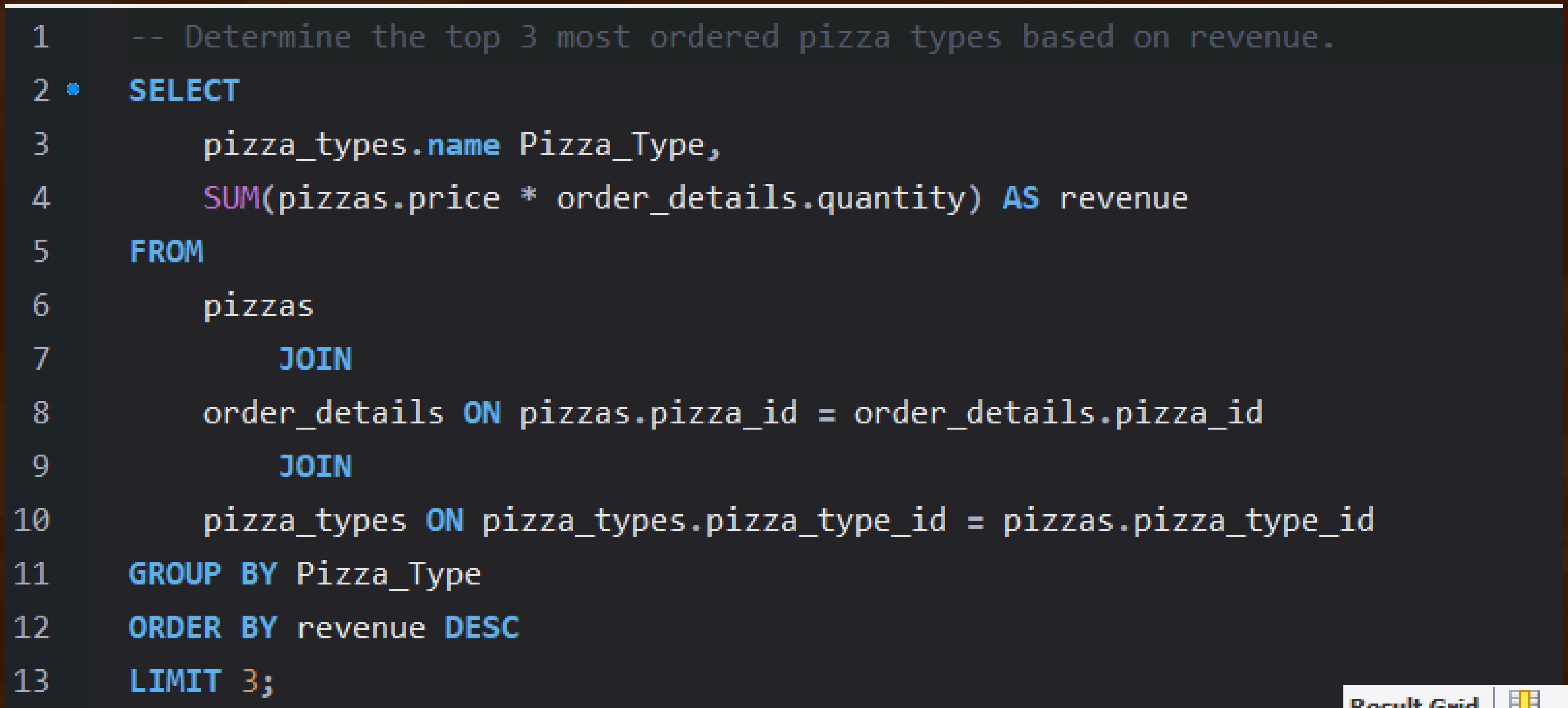
```sql
-- Calculate the percentage contribution of each pizza type to total revenue.

SELECT
    category,
    ROUND((SUM(quantity * price) / (SELECT
                    SUM(pizzas.price * order_details.quantity)
                FROM
                    pizzas
                        JOIN
                    order_details ON pizzas.pizza_id = order_details.pizza_id)) * 100,
            2) AS revenue_in_percent
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY category;
```

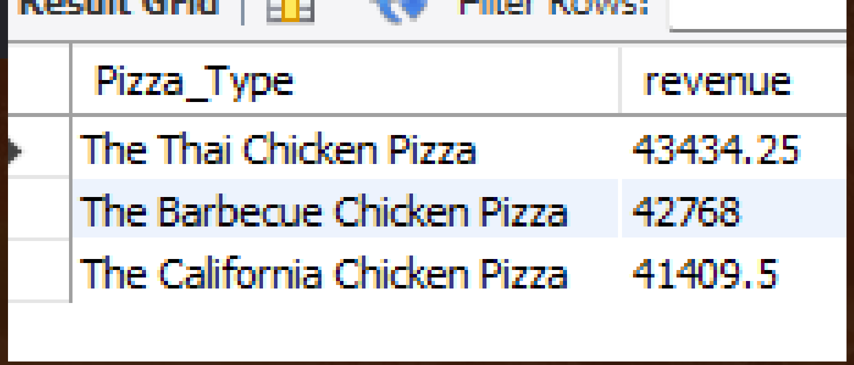


category	revenue_in_percent
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

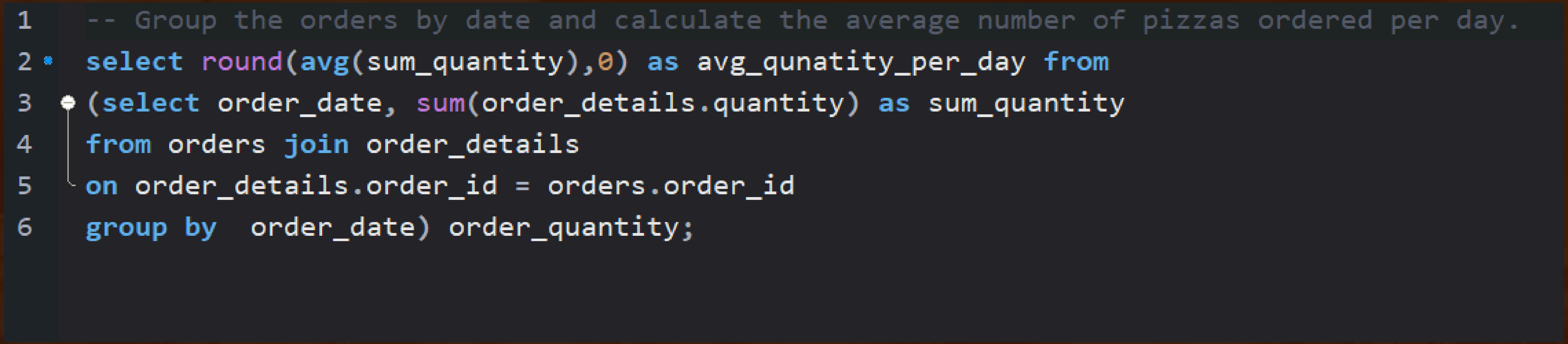


```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name Pizza_Type,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY Pizza_Type
ORDER BY revenue DESC
LIMIT 3;
```

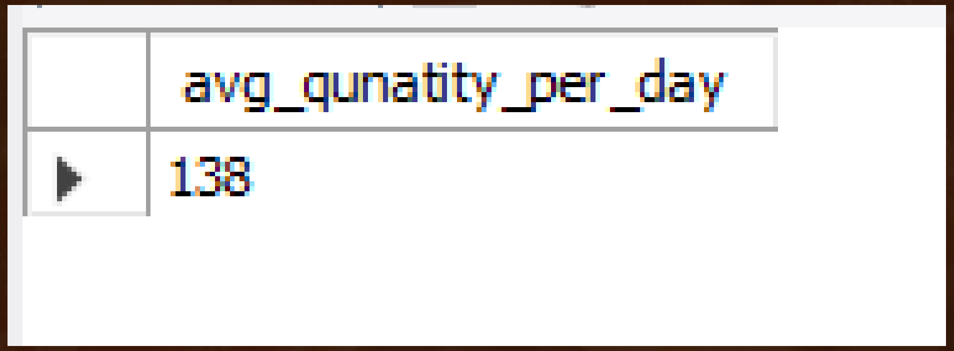


Pizza_Type	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

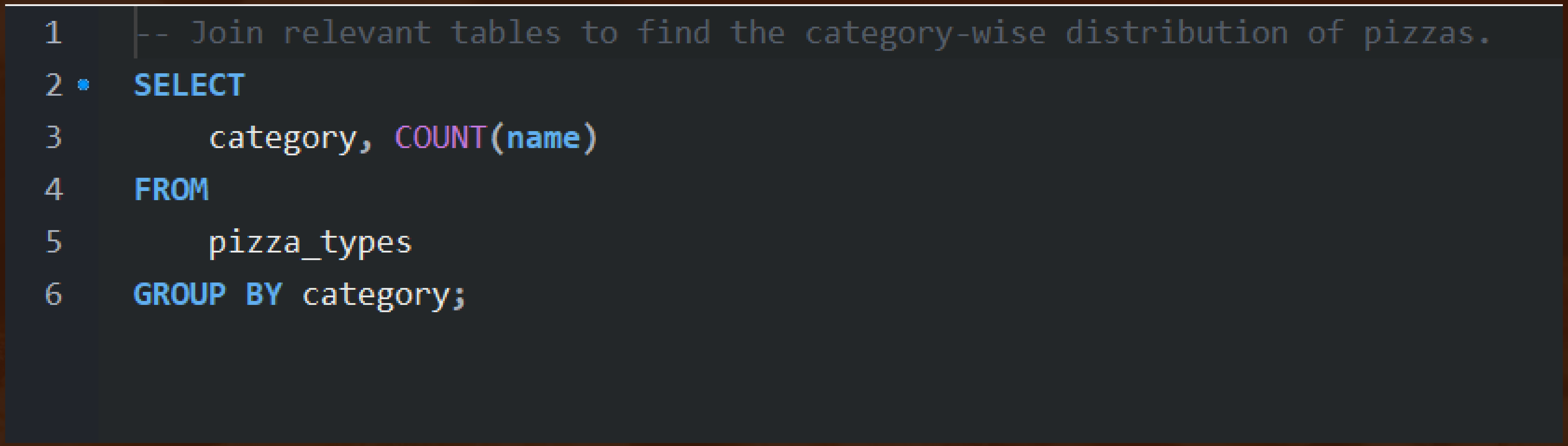


```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
select round(avg(sum_quantity),0) as avg_qunatity_per_day from
(select order_date, sum(order_details.quantity) as sum_quantity
from orders join order_details
on order_details.order_id = orders.order_id
group by  order_date) order_quantity;
```

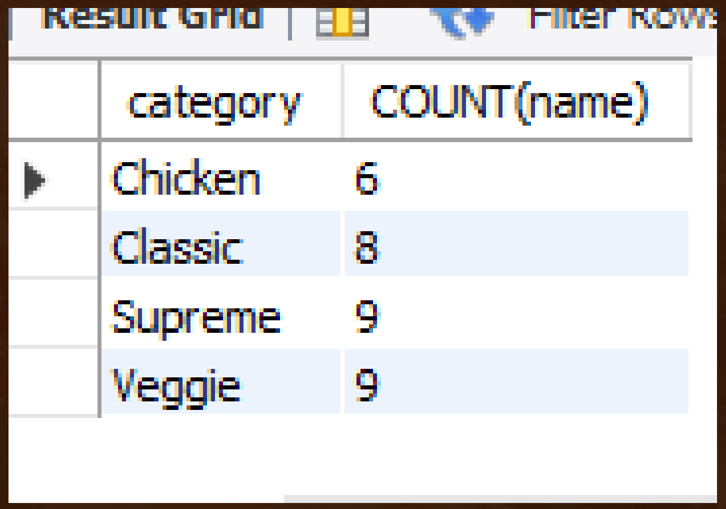


avg_qunatity_per_day
138

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

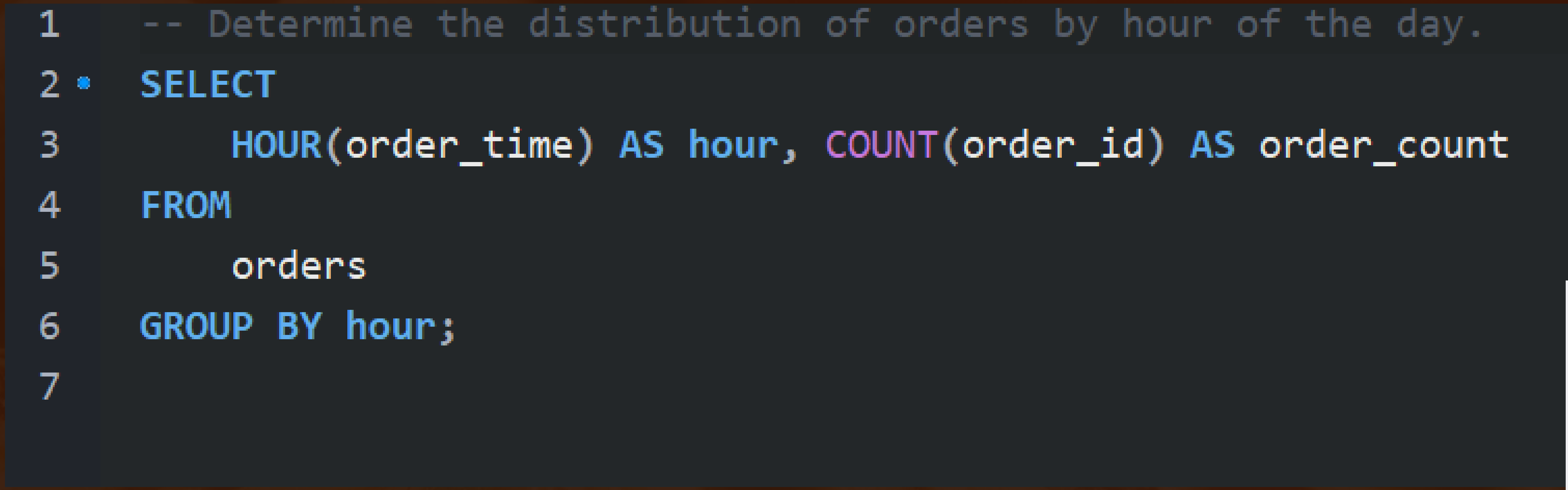


```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



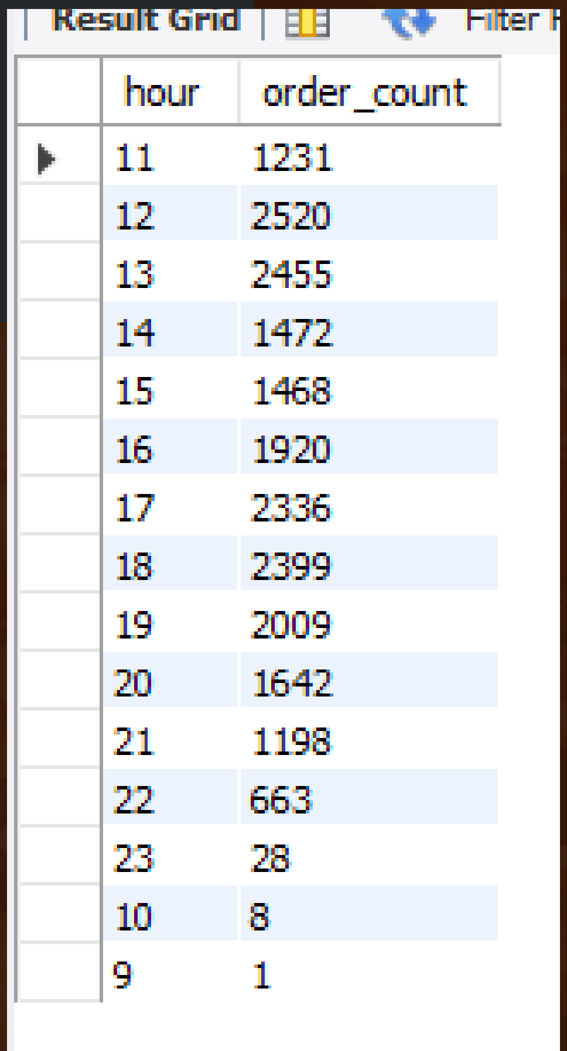
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



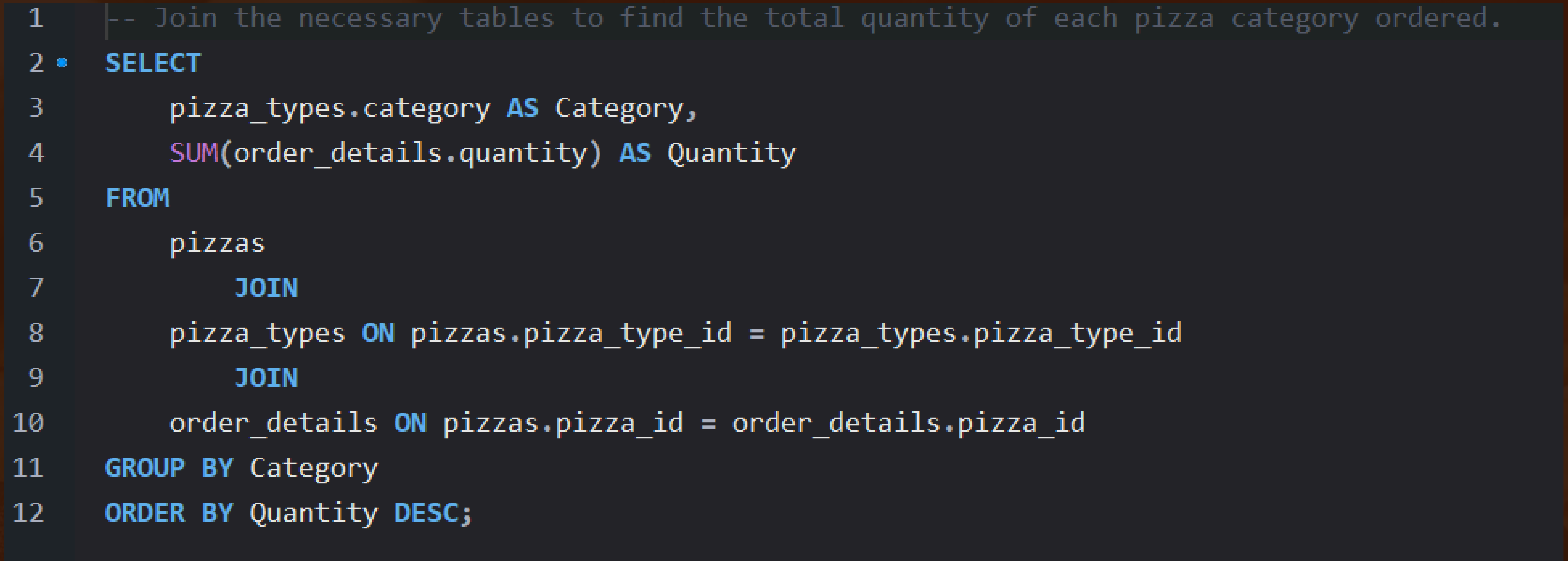
```sql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour;

```

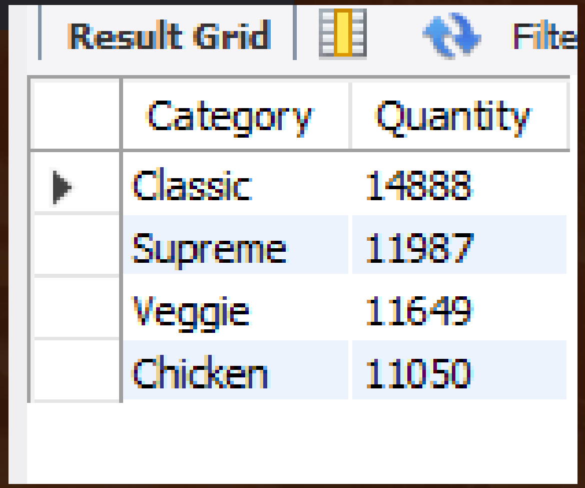


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

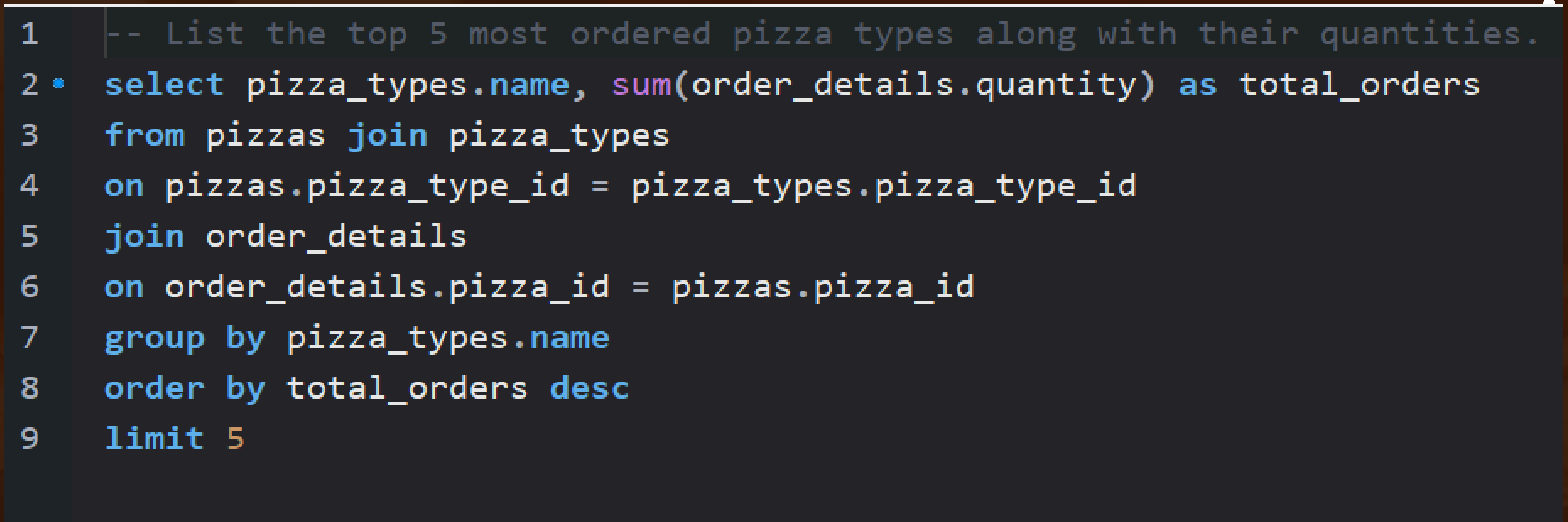


```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category AS Category,
    SUM(order_details.quantity) AS Quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY Category
ORDER BY Quantity DESC;
```

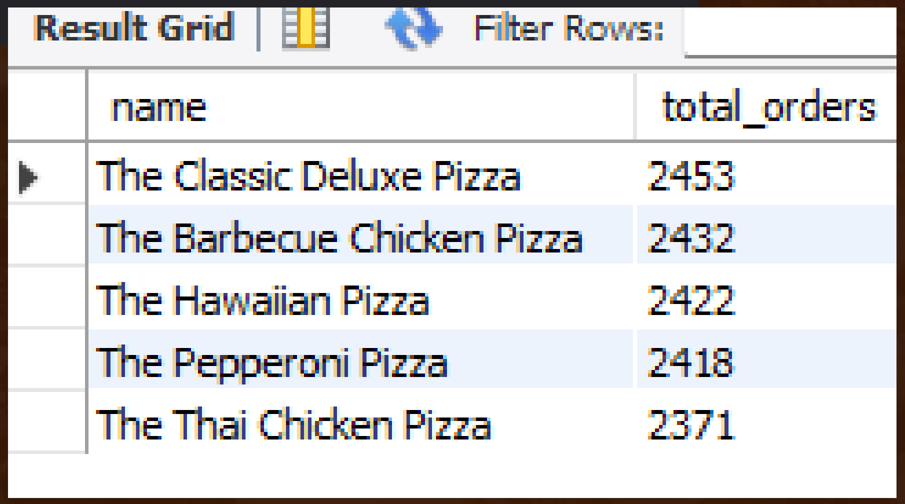


Category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

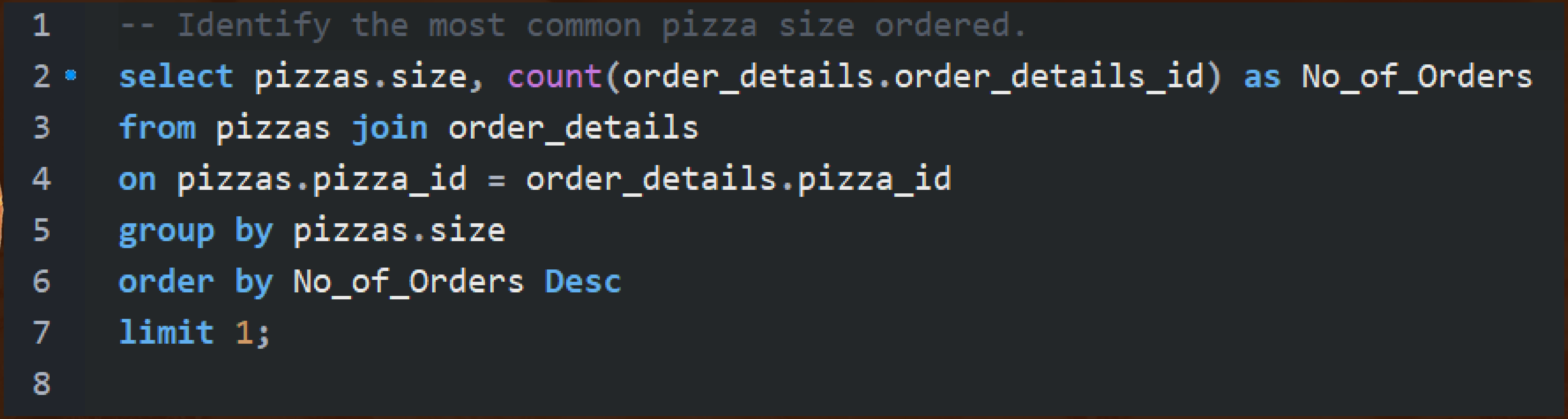


```sql
-- List the top 5 most ordered pizza types along with their quantities.
select pizza_types.name, sum(order_details.quantity) as total_orders
from pizzas join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by total_orders desc
limit 5
```

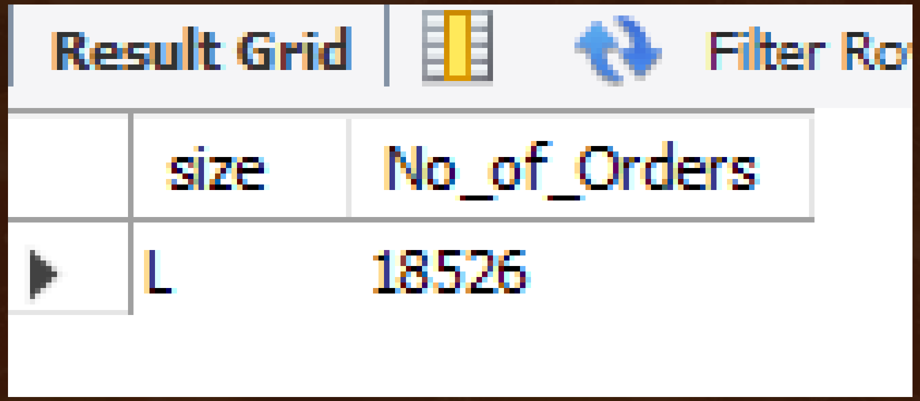


name	total_orders
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

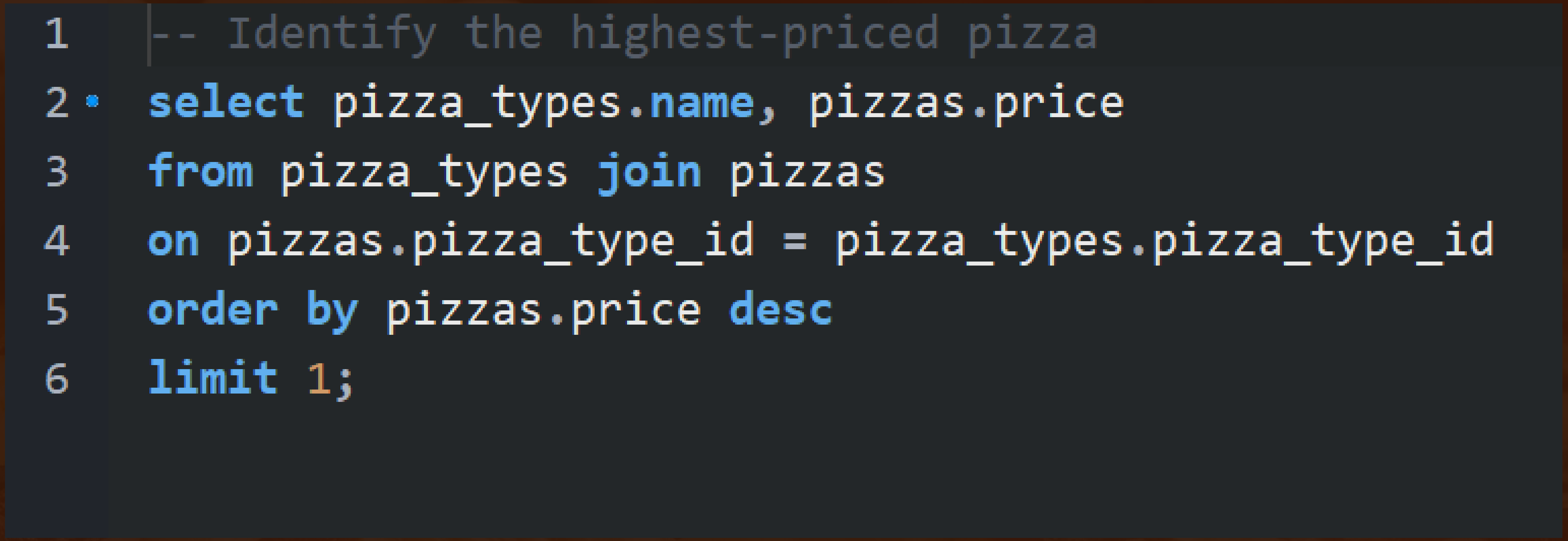


```sql
-- Identify the most common pizza size ordered.
select pizzas.size, count(order_details.order_details_id) as No_of_Orders
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by No_of_Orders Desc
limit 1;
```

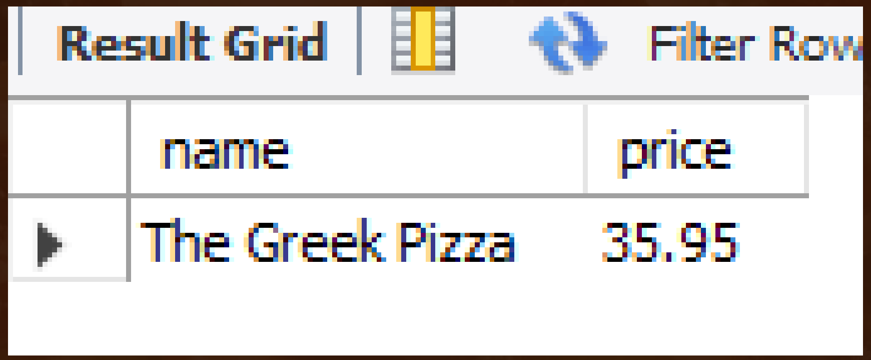


size	No_of_Orders
L	18526

IDENTIFY THE HIGHEST-PRICED PIZZA.

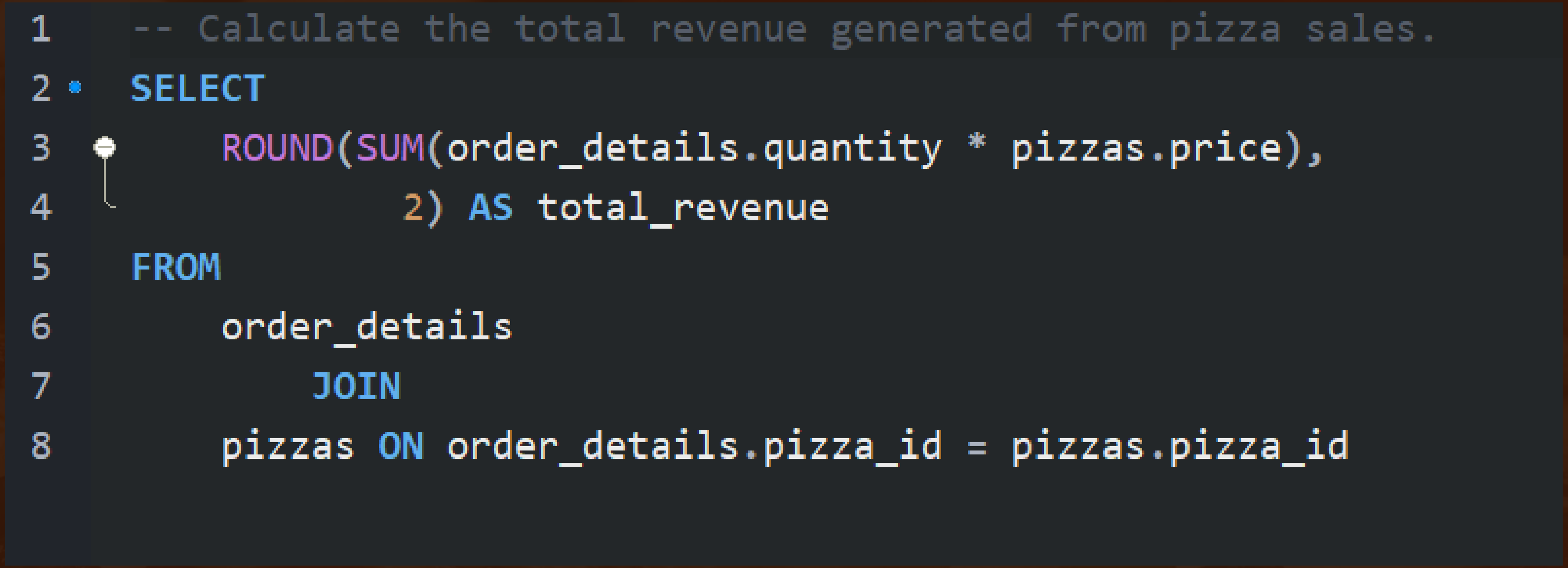


```sql
-- Identify the highest-priced pizza
select pizza_types.name, pizzas.price
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc
limit 1;
```

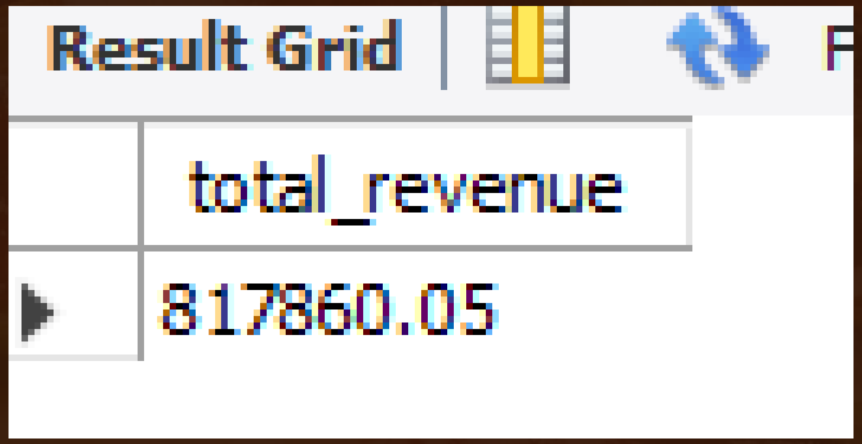


name	price
The Greek Pizza	35.95

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

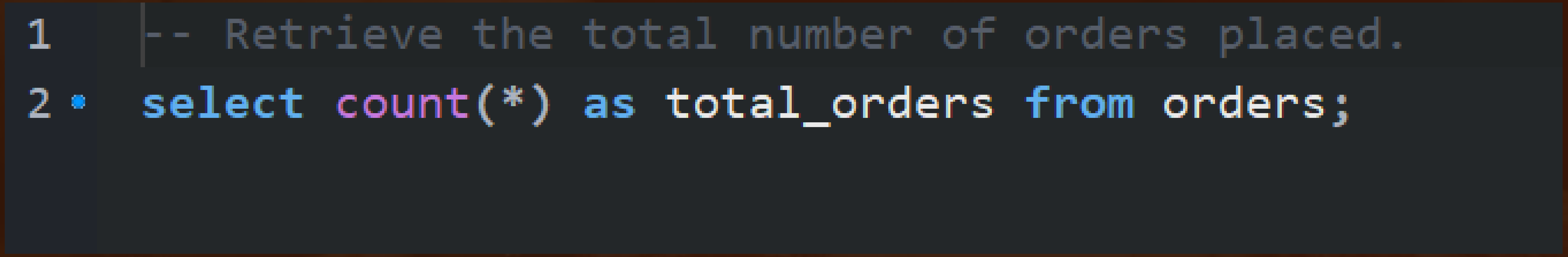


```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

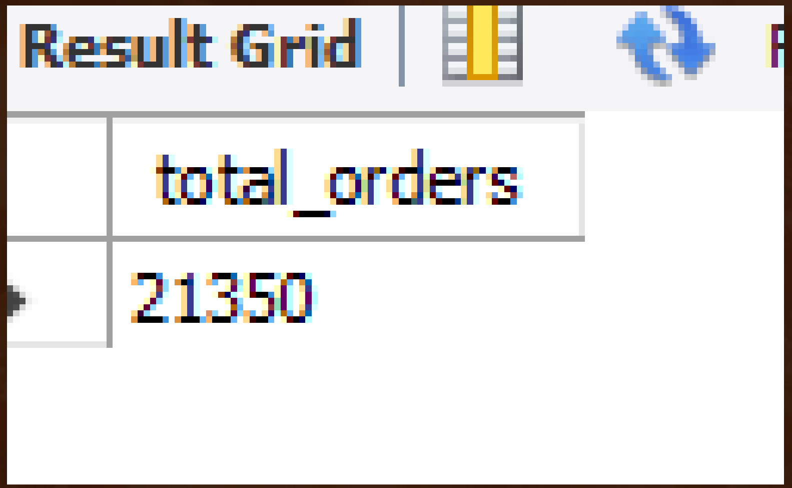


total_revenue
817860.05

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```sql
-- Retrieve the total number of orders placed.
select count(*) as total_orders from orders;
```



total_orders
21350

THANK YOU

Project files :-