PIZZA SALES
SQL PROJECT

INTRODUCTION TO THE PROJECT

WELCOME

I AM ABHIK DAS

In this project, I have utilized SQL queries to solve real-life problems. This project focuses on analyzing and deriving insights from a pizza sales dataset using SQL. The dataset contains information about customer orders, pizzas sold, their types, prices, and quantities. The goal of this project is to answer key business questions that can help a pizza store optimize its operations, understand customer preferences, and improve overall sales performance.

OBJECTIVES:

1. ANALYZE SALES PERFORMANCE:

- Calculate total sales and revenue trends over time.
- Identify the best-selling pizza types and categories.

2. CUSTOMER INSIGHTS:

- Determine customer preferences based on pizza types and sizes.
- Analyze the frequency of repeat orders.

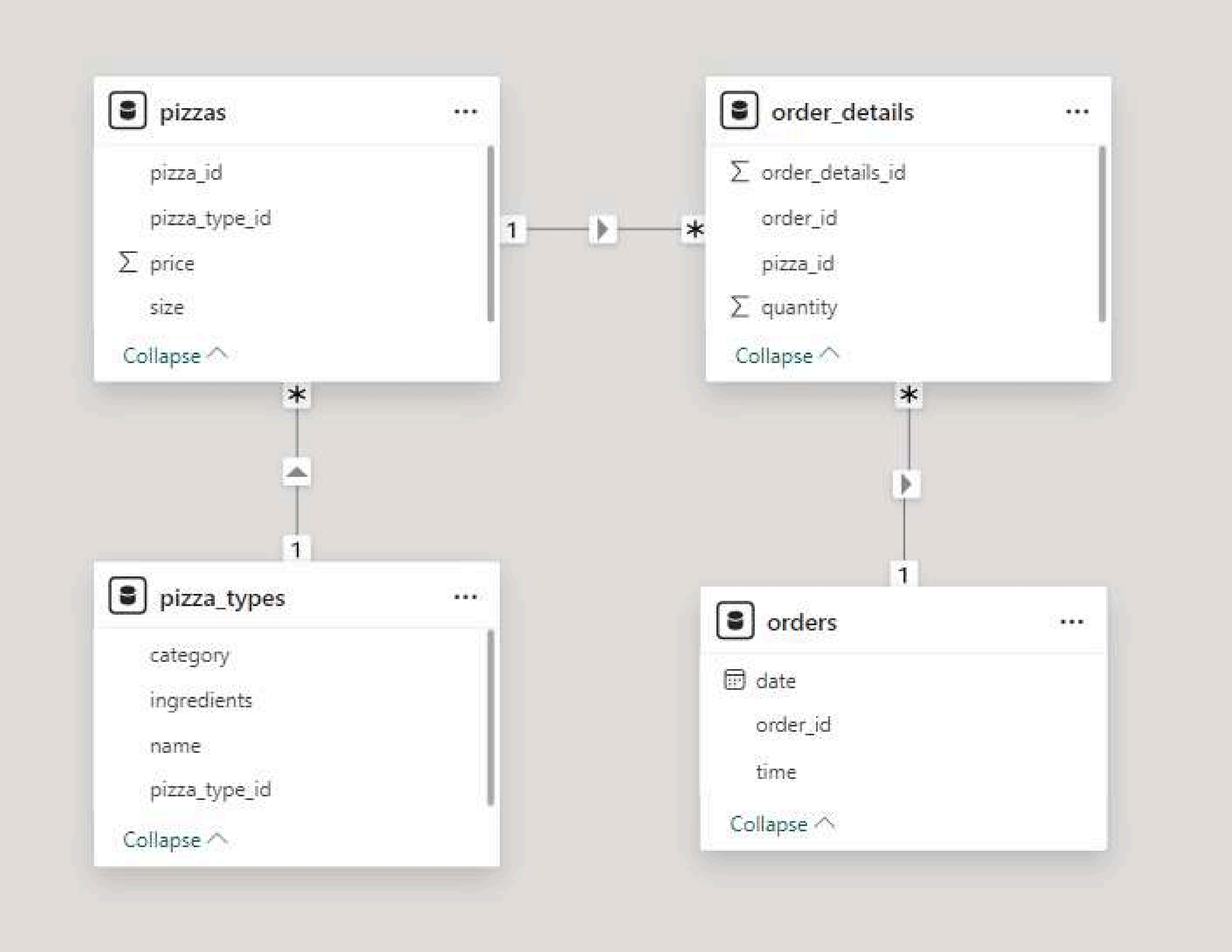
3. REVENUE OPTIMIZATION:

- Calculate the percentage revenue contribution of each pizza type/category.
- Identify low-performing pizzas to focus marketing or revise the menu.

4. OPERATIONAL EFFICIENCY:

- Understand peak order times and days to optimize staffing and inventory.
- Evaluate the average order size and value to improve upselling strategies.

DATABASE STRUCTURE AND TABLE RELATIONSHIPS



pizzas ···

pizza_id
pizza_type_id
∑ price
size
Collapse ^

order_details ···

∑ order_details_id
order_id
pizza_id
∑ quantity
Collapse ^

pizza_types ···

category
ingredients
name
pizza_type_id
Collapse ^

orders ···

date
order_id
time
Collapse ^

SQL PROJECT

THERE ARE 13 QUESTIONS THAT HAS BEEN SOLVED THROUGH THIS PROJECT

Basic:

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.

Intermediate:

1.Join the necessary tables to find the total quantity of each pizza category ordered.
2.Determine the distribution of orders by hour of the day.
3.Join relevant tables to find the category-wise distribution of pizzas.
4.Group the orders by date and calculate the average number of pizzas ordered per day.
5.Determine the top 3 most ordered pizza types based on revenue.

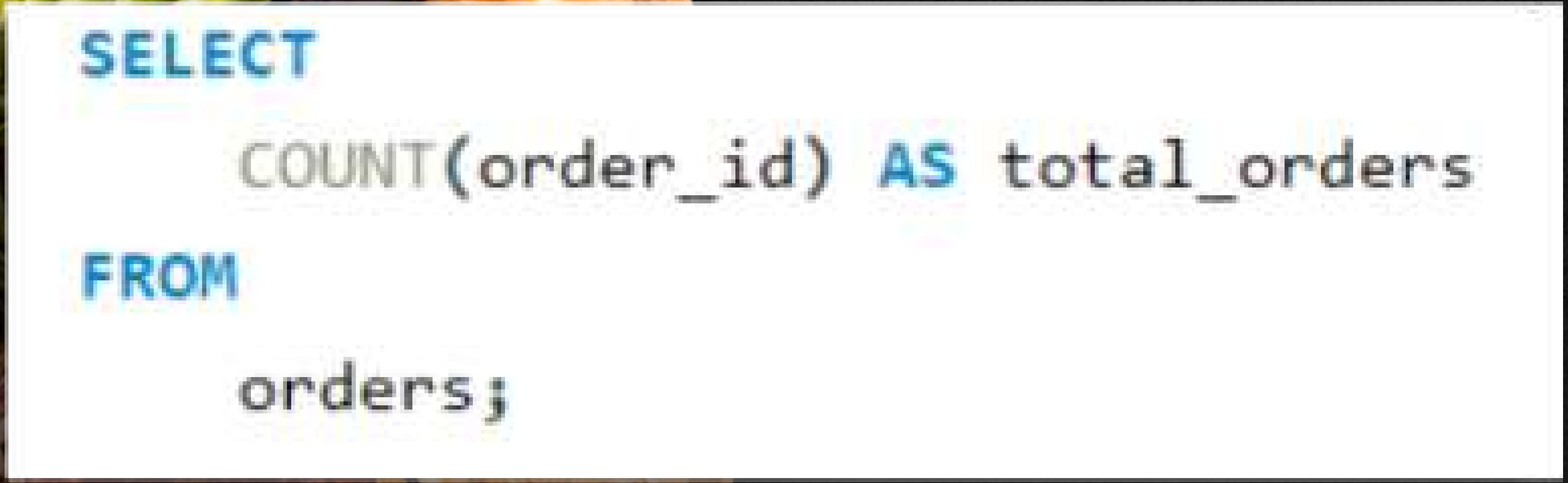
Advanced:

1.Calculate the percentage contribution of each pizza type to total revenue.
2.Analyze the cumulative revenue generated over time.
3.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

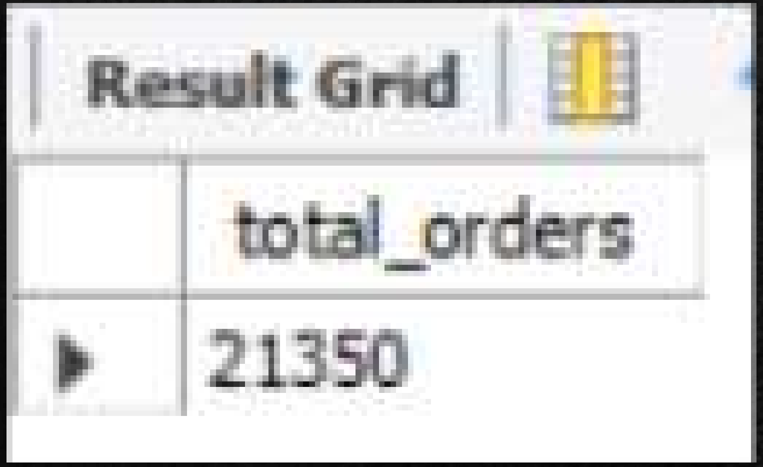
BASIC LEVEL QUESTIONS

★★☆☆☆

Retrieve the total number of orders placed.

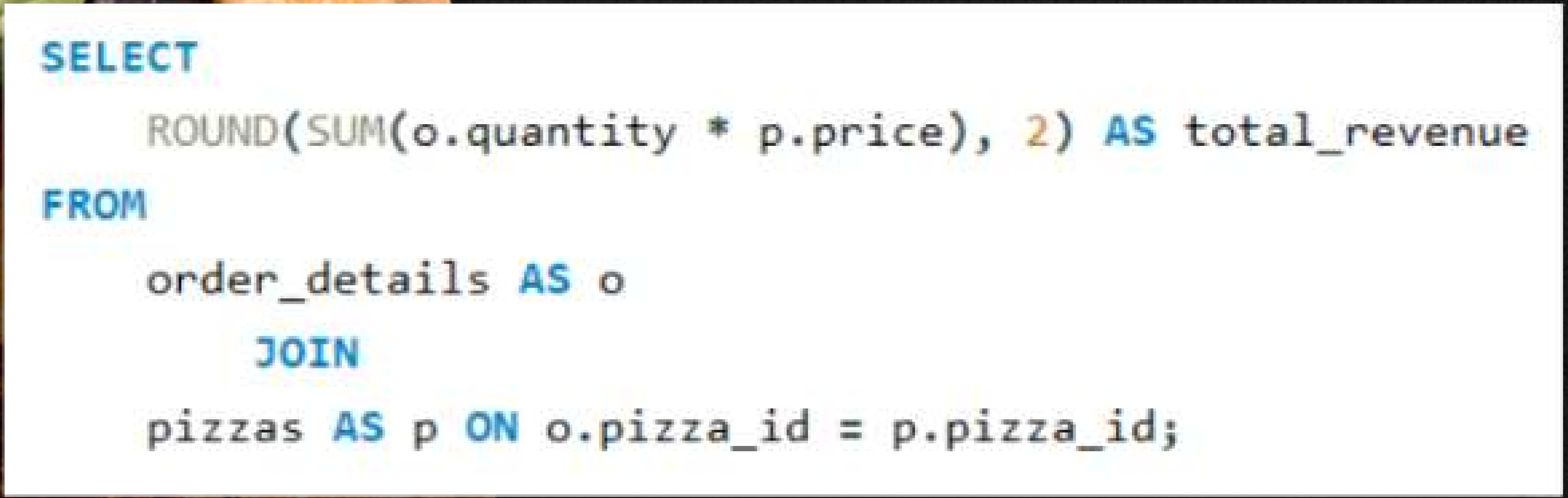


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

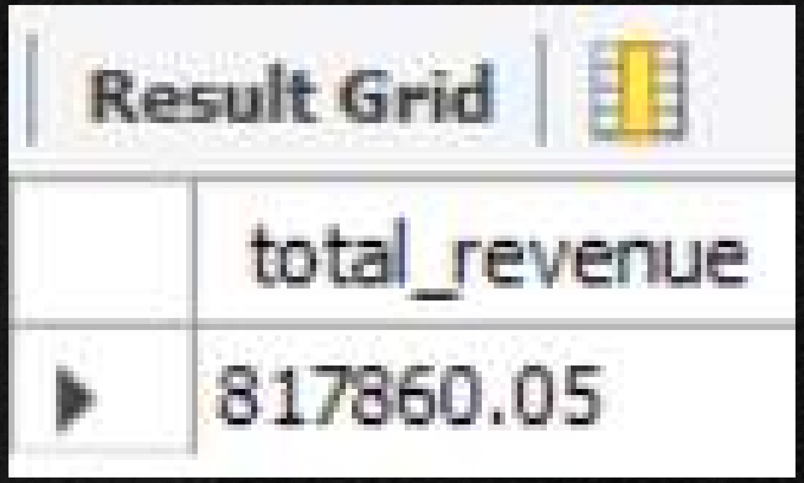


```
total_orders
21350
```

Calculate the total revenue generated from pizza sales.

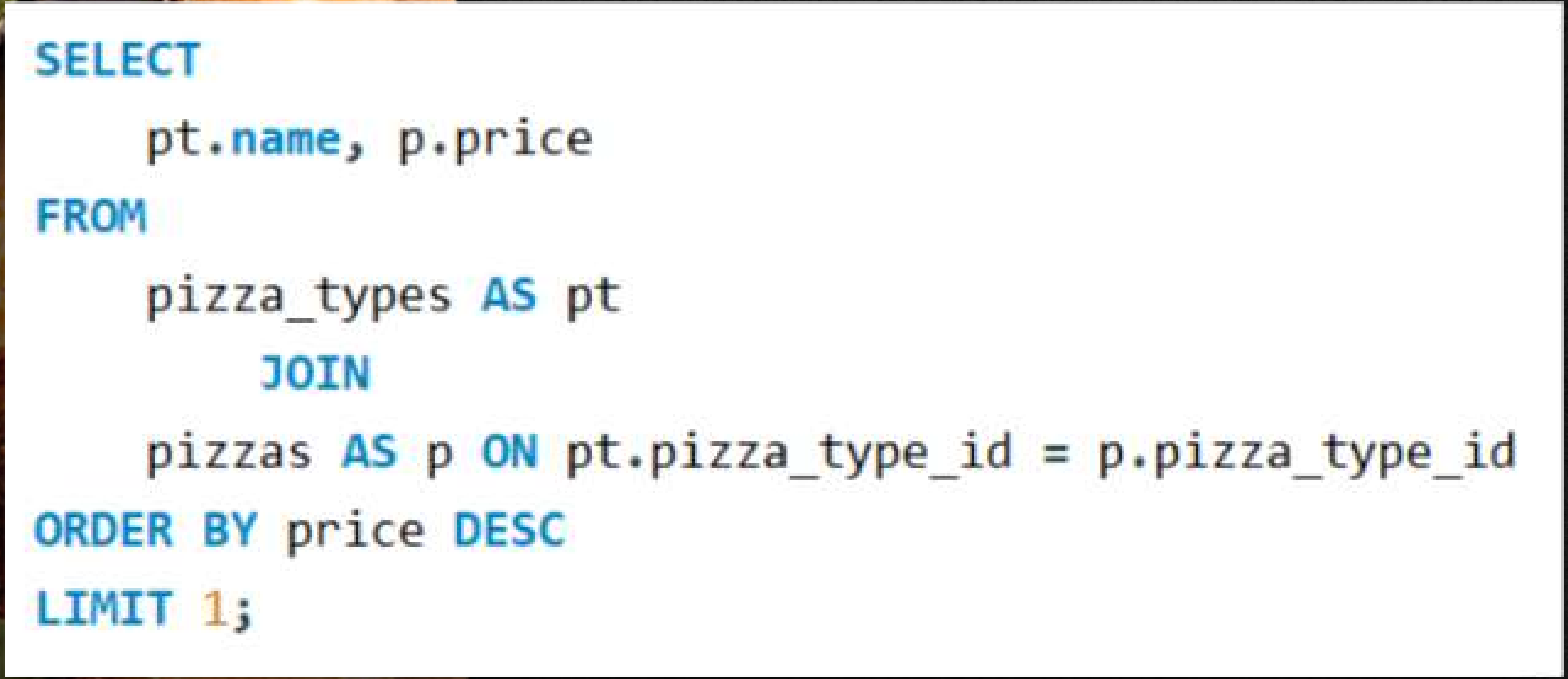


```
SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS total_revenue
FROM
    order_details AS o
        JOIN
    pizzas AS p ON o.pizza_id = p.pizza_id;
```

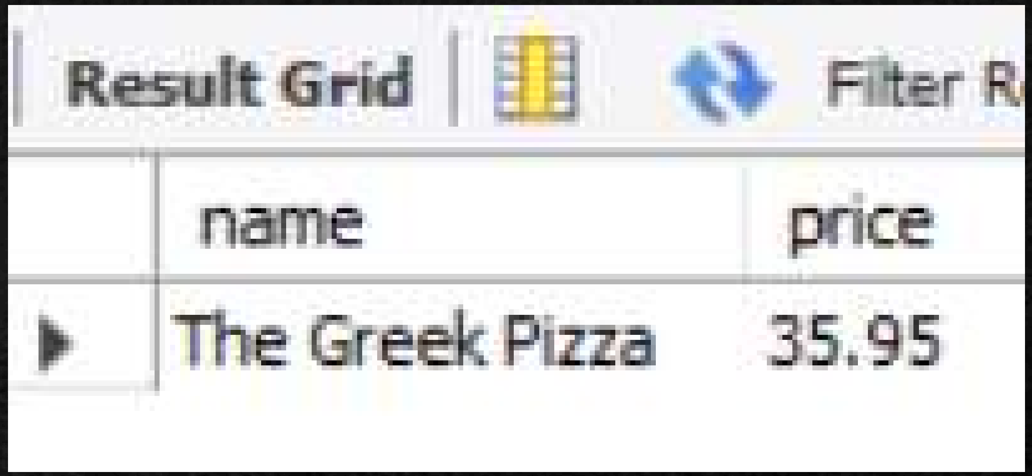


```
total_revenue
817860.05
```

Identify the highest-priced pizza.



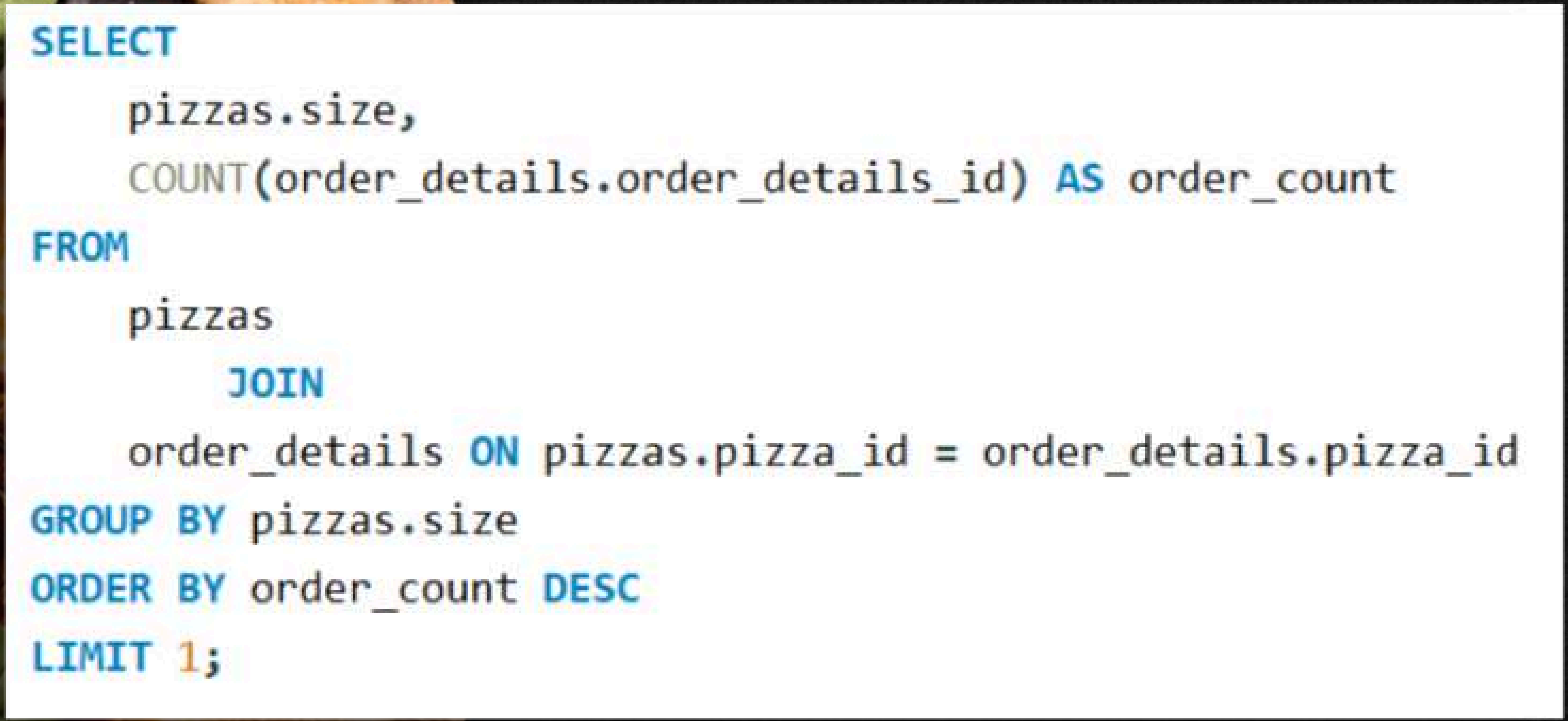
```
SELECT
    pt.name, p.price
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```



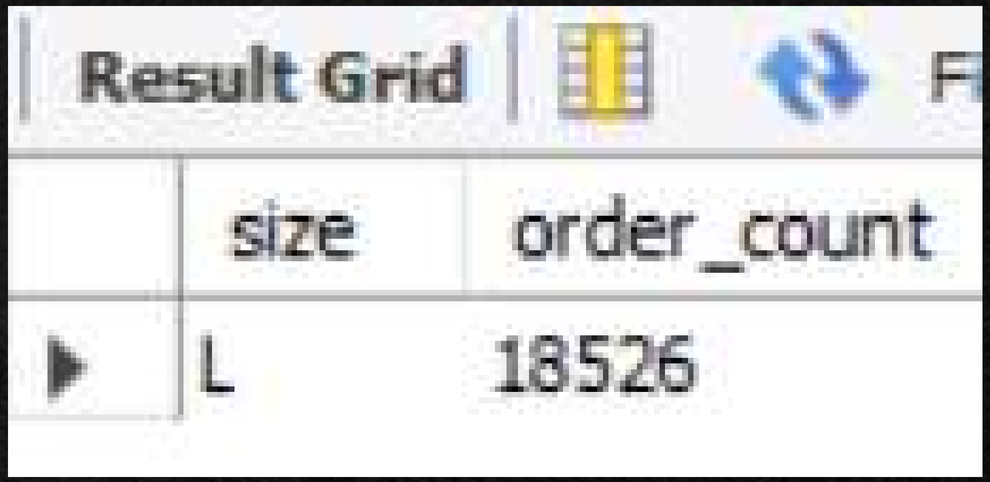
```
  name              price
▶ The Greek Pizza   35.95
```

SQL PROJECT

Identify the most common pizza size ordered.



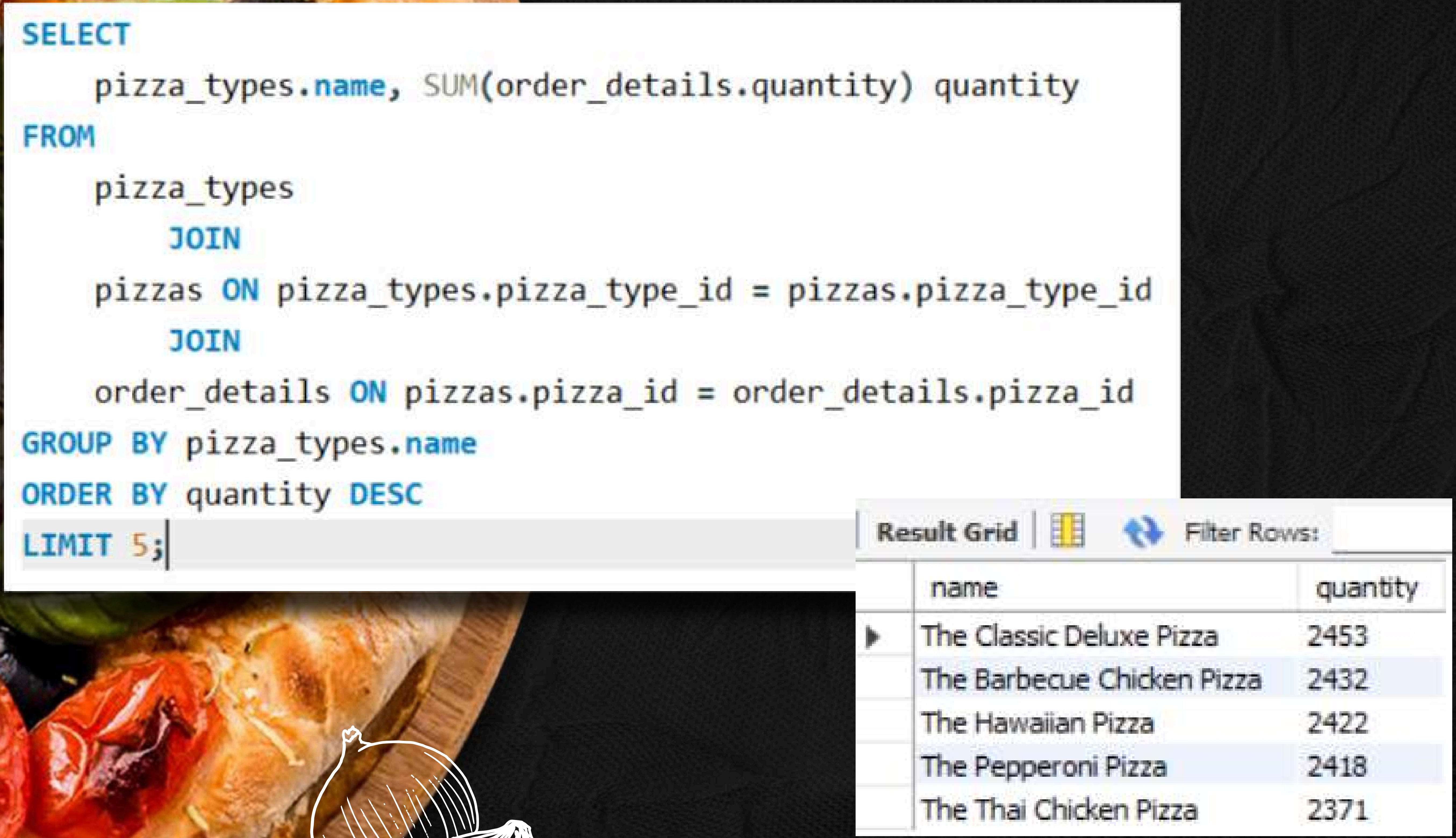
```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```



size	order_count
L	18526

SQL PROJECT

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



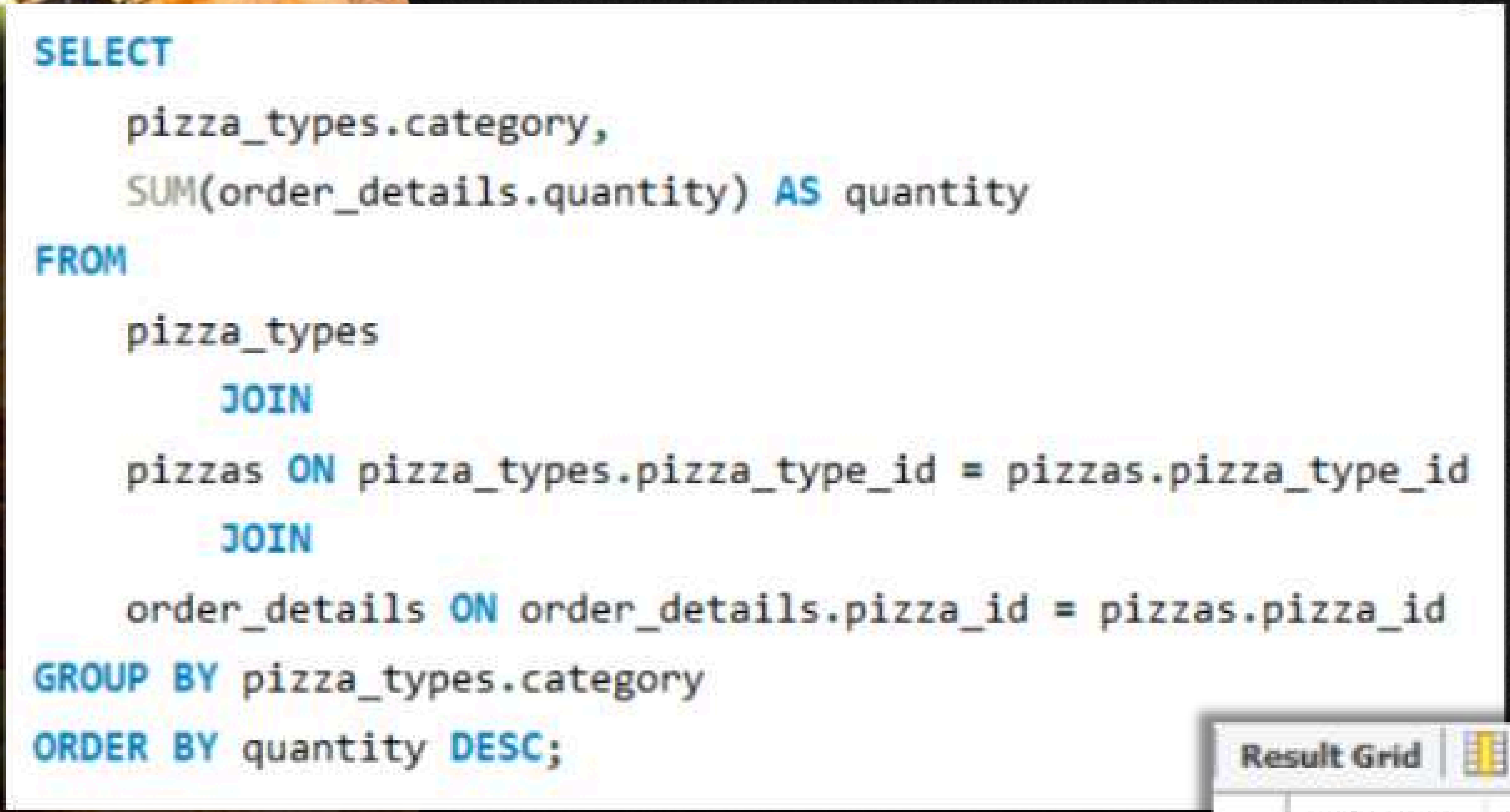
```
SELECT
    pizza_types.name, SUM(order_details.quantity) quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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

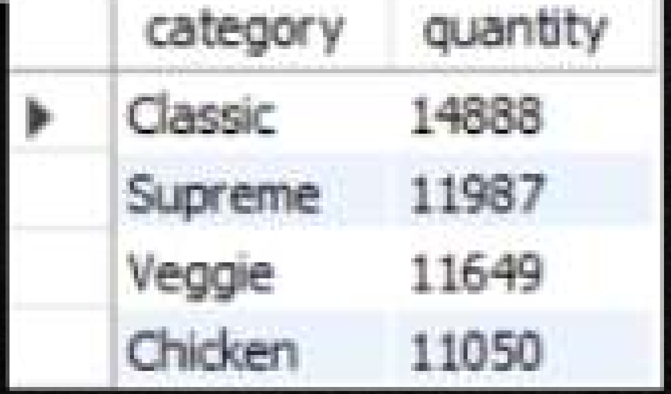
INTERMEDIATE LEVEL QUESTIONS

★★★☆☆

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



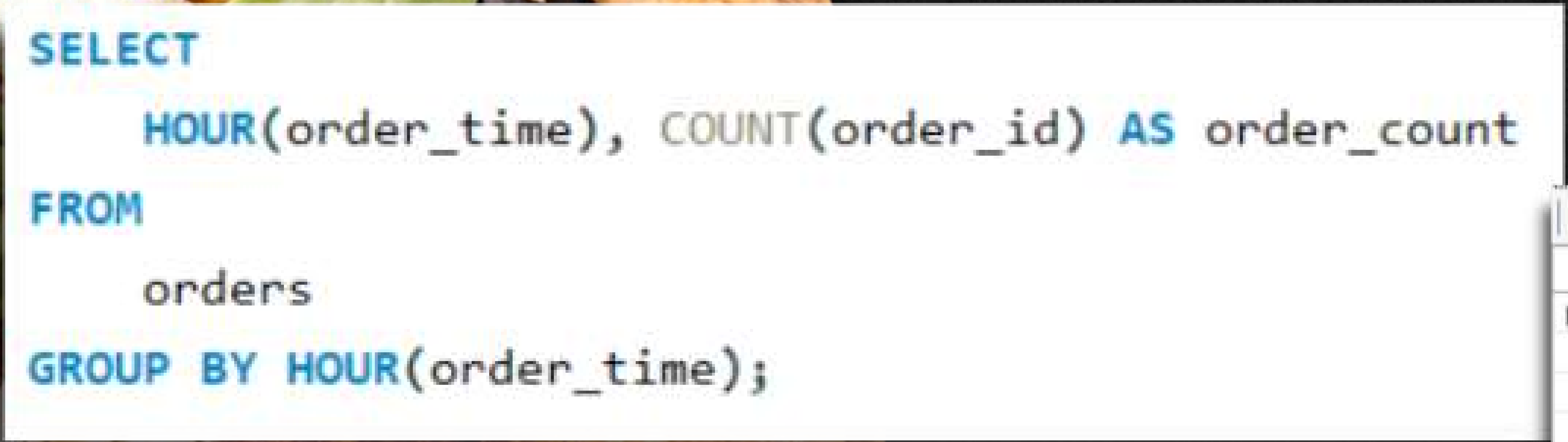
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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



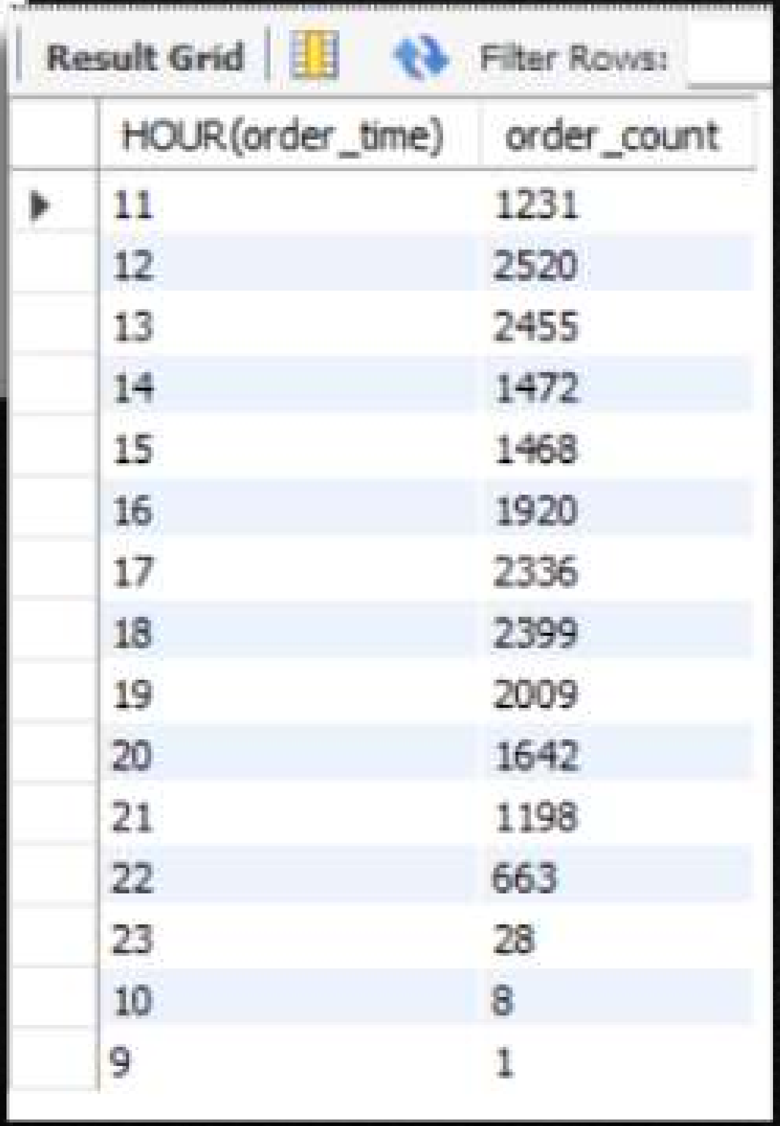
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

SQL PROJECT

Determine the distribution of orders by hour of the day.

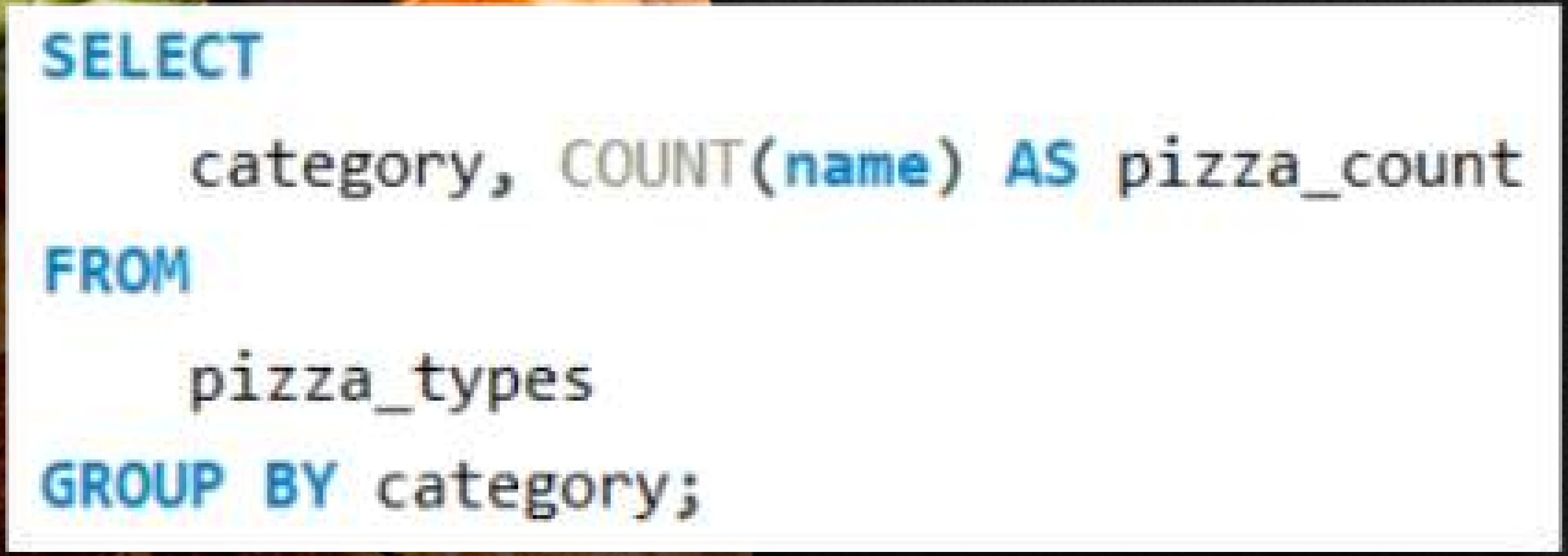


```
SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

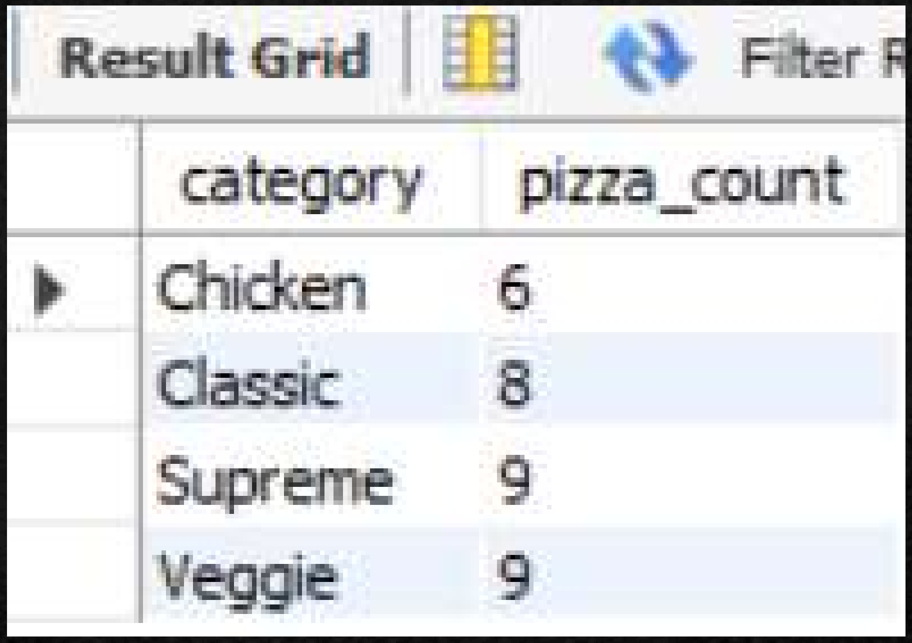


HOUR(order_time)	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Find the category-wise distribution of pizzas.

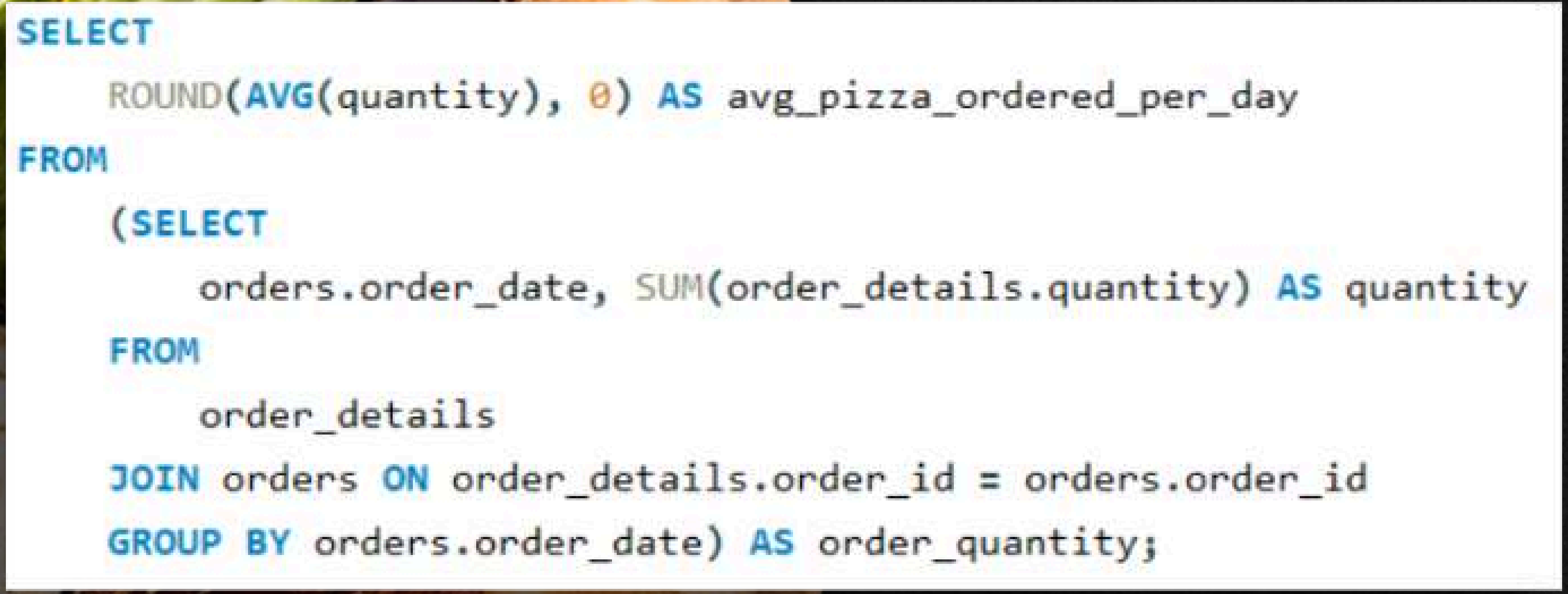


```
SELECT
    category, COUNT(name) AS pizza_count
FROM
    pizza_types
GROUP BY category;
```

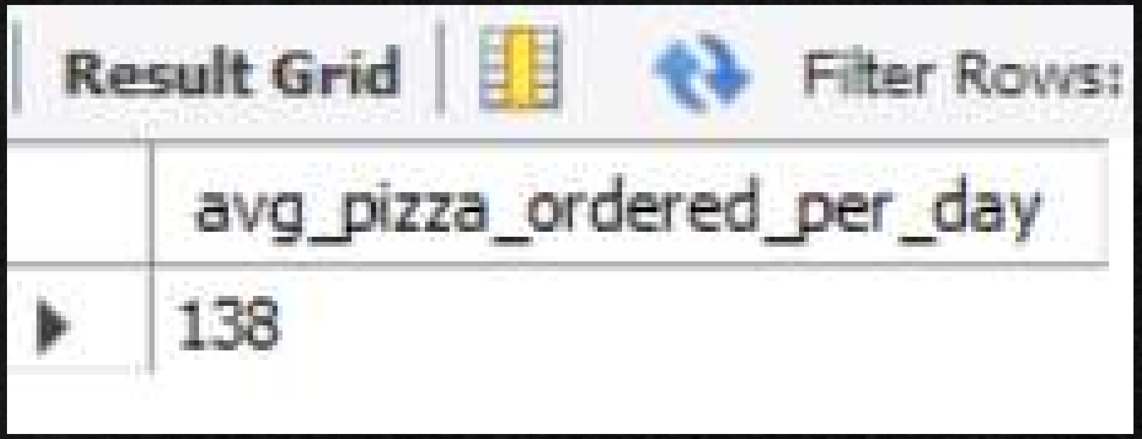


category	pizza_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

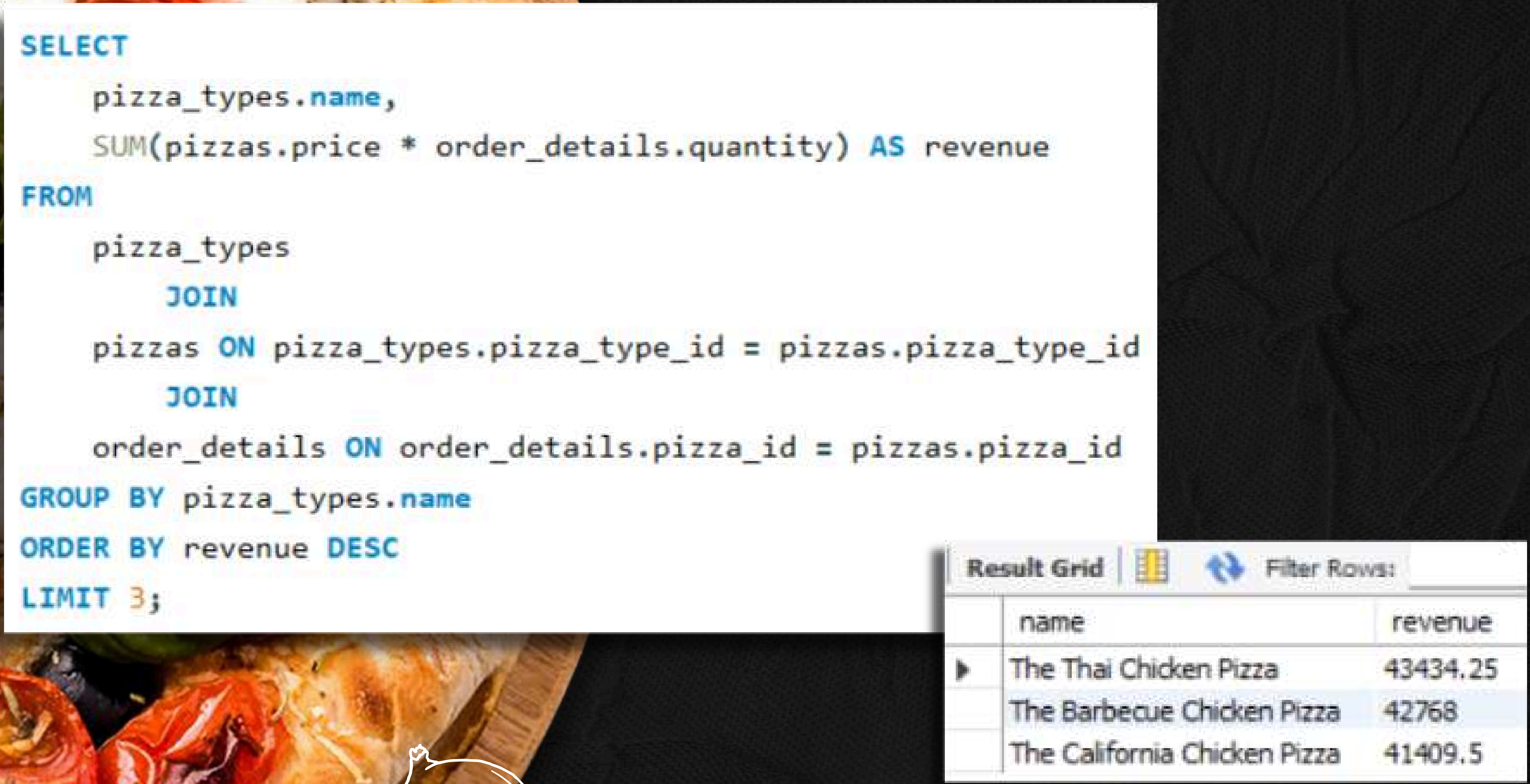


```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



```
avg_pizza_ordered_per_day
138
```

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



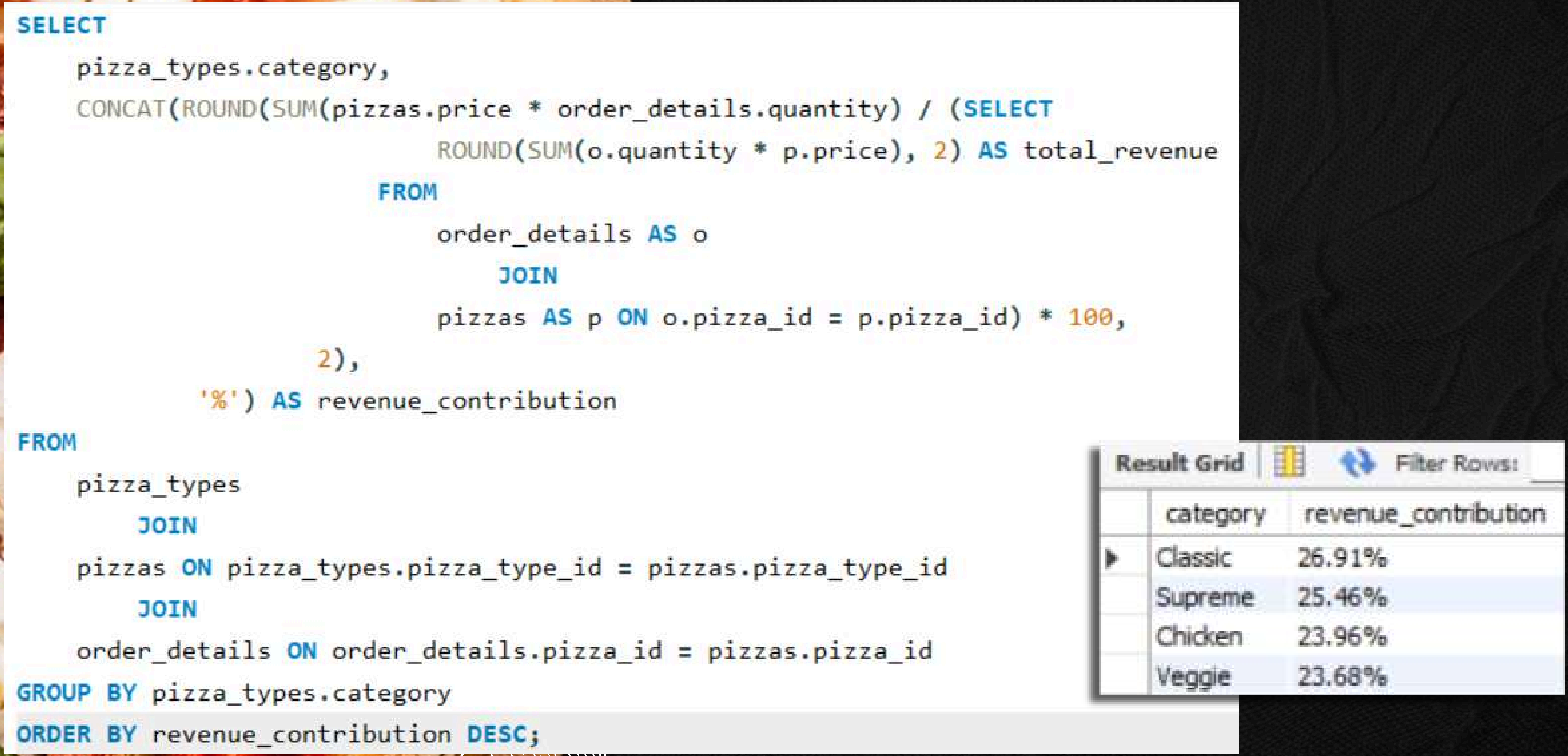
```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

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

SQL PROJECT

ADVANCED LEVEL QUESTIONS

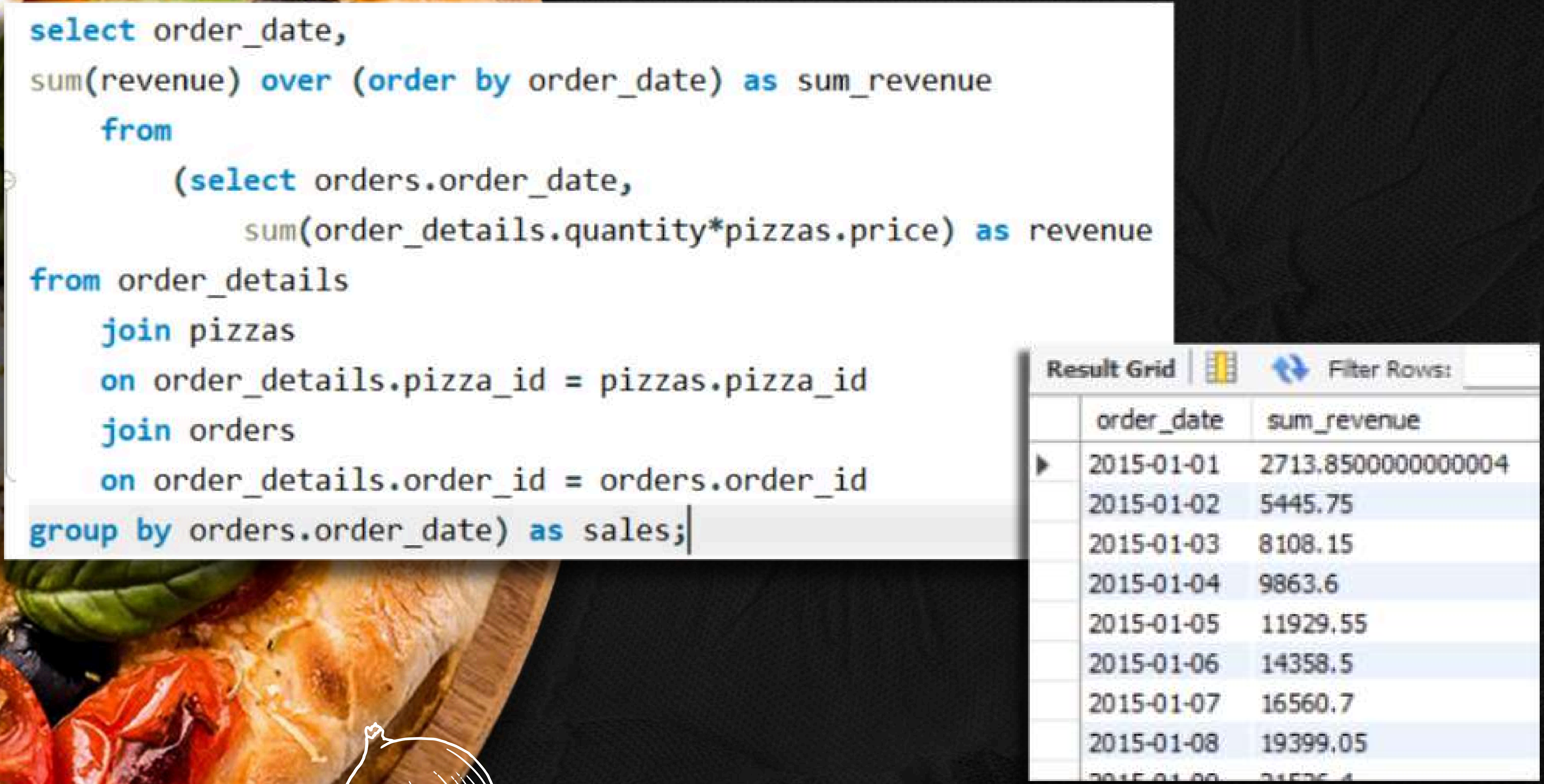
Calculate the percentage contribution of each pizza type to total revenue.



```sql
SELECT
    pizza_types.category,
    CONCAT(ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                            ROUND(SUM(o.quantity * p.price), 2) AS total_revenue
                        FROM
                            order_details AS o
                                JOIN
                            pizzas AS p ON o.pizza_id = p.pizza_id) * 100,
                    2),
            '%') AS revenue_contribution
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_contribution DESC;
```

category	revenue_contribution
Classic	26.91%
Supreme	25.46%
Chicken	23.96%
Veggie	23.68%

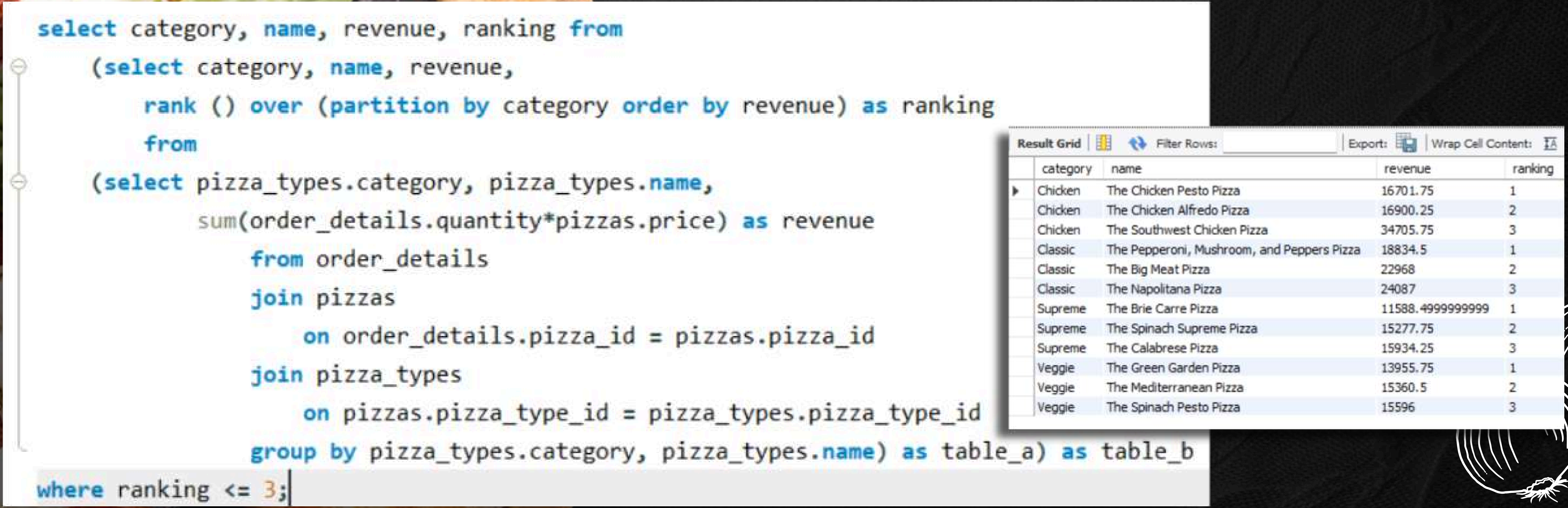
Analyze the cumulative revenue generated over time.



```
select order_date,
sum(revenue) over (order by order_date) as sum_revenue
    from
        (select orders.order_date,
            sum(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	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

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



```sql
select category, name, revenue, ranking from
    (select category, name, revenue,
        rank () over (partition by category order by revenue) as ranking
        from
    (select pizza_types.category, pizza_types.name,
            sum(order_details.quantity*pizzas.price) as revenue
                from order_details
                join pizzas
                    on order_details.pizza_id = pizzas.pizza_id
                join pizza_types
                    on pizzas.pizza_type_id = pizza_types.pizza_type_id
                group by pizza_types.category, pizza_types.name) as table_a) as table_b
where ranking <= 3;
```

```
category  name                                       revenue           ranking
Chicken   The Chicken Pesto Pizza                    16701.75          1
Chicken   The Chicken Alfredo Pizza                  16900.25          2
Chicken   The Southwest Chicken Pizza                34705.75          3
Classic   The Pepperoni, Mushroom, and Peppers Pizza 18834.5           1
Classic   The Big Meat Pizza                         22968             2
Classic   The Napolitana Pizza                       24087             3
Supreme   The Brie Carre Pizza                       11588.4999999999  1
Supreme   The Spinach Supreme Pizza                  15277.75          2
Supreme   The Calabrese Pizza                        15934.25          3
Veggie    The Green Garden Pizza                     13955.75          1
Veggie    The Mediterranean Pizza                    15360.5           2
Veggie    The Spinach Pesto Pizza                    15596             3
```

SQL PROJECT

THANK YOU!

✉ abhikdss98123@gmail.com