Introducing

The Pizza Sales Analysis Report using SQL

Utkarsh Srivastava

Hello !

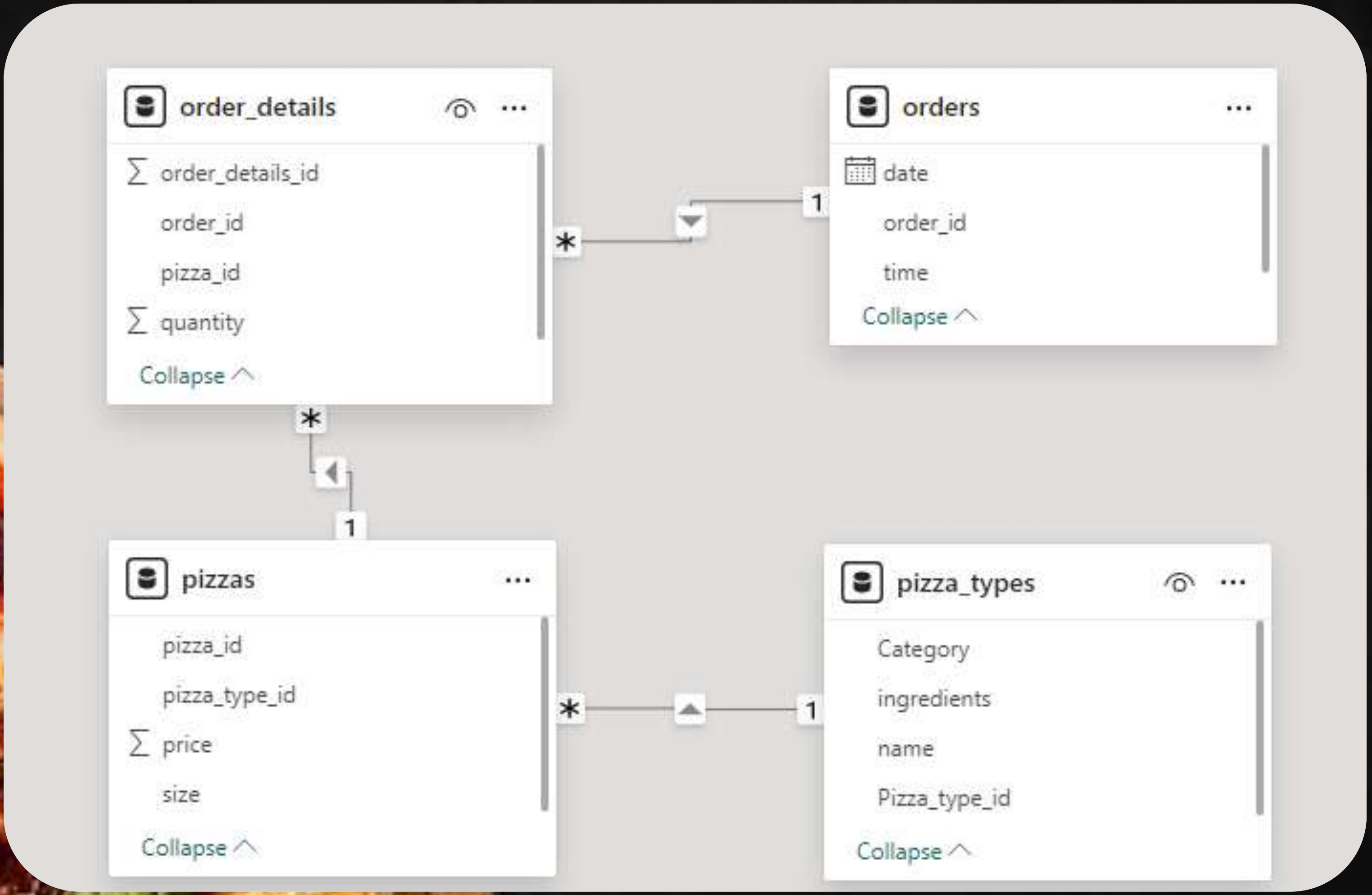
In this Project I am utilizing the SQL to undertake an comprehensive analysis of pizza sales with the aim of deriving valuable insights and informed strategic decision making process.

Project Overview

The objective is to utilize advanced data analysis methodologies to extract actionable insights from a comprehensive database, this initiative empowers the business to make well informed decision, thereby facilitating strategic growth within the competitive pizza industry.

Pizza Sales Insights through Data Modeling

Pizza sales data to understand relationship between orders, pizzas, order_details and Pizza_types



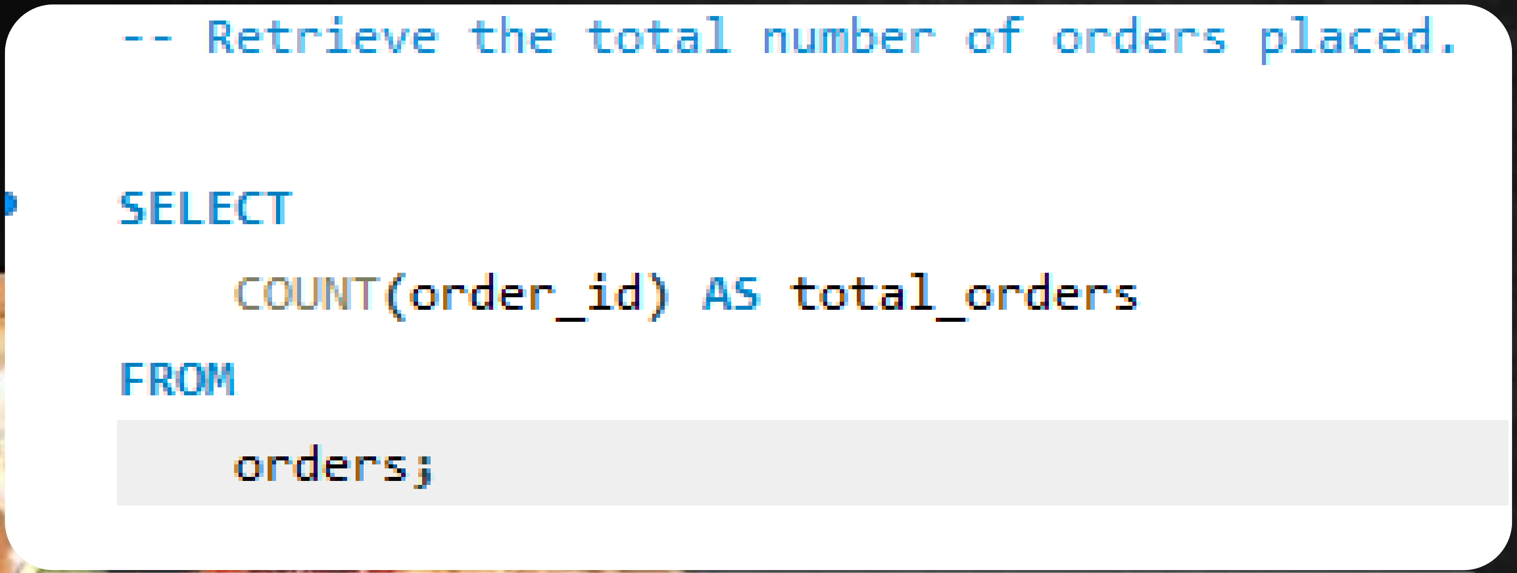
order_details
- order_details_id
- order_id
- pizza_id
- quantity

orders
- date
- order_id
- time

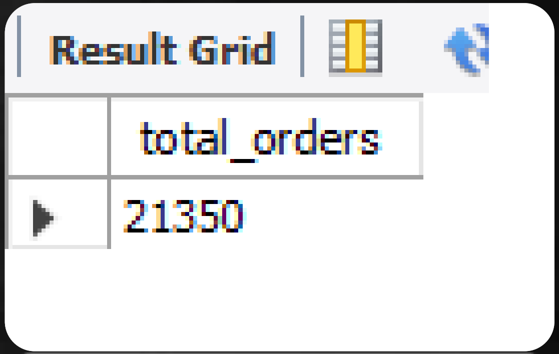
pizzas
- pizza_id
- pizza_type_id
- price
- size

pizza_types
- Category
- ingredients
- name
- Pizza_type_id

Retrieve the total number of orders placed

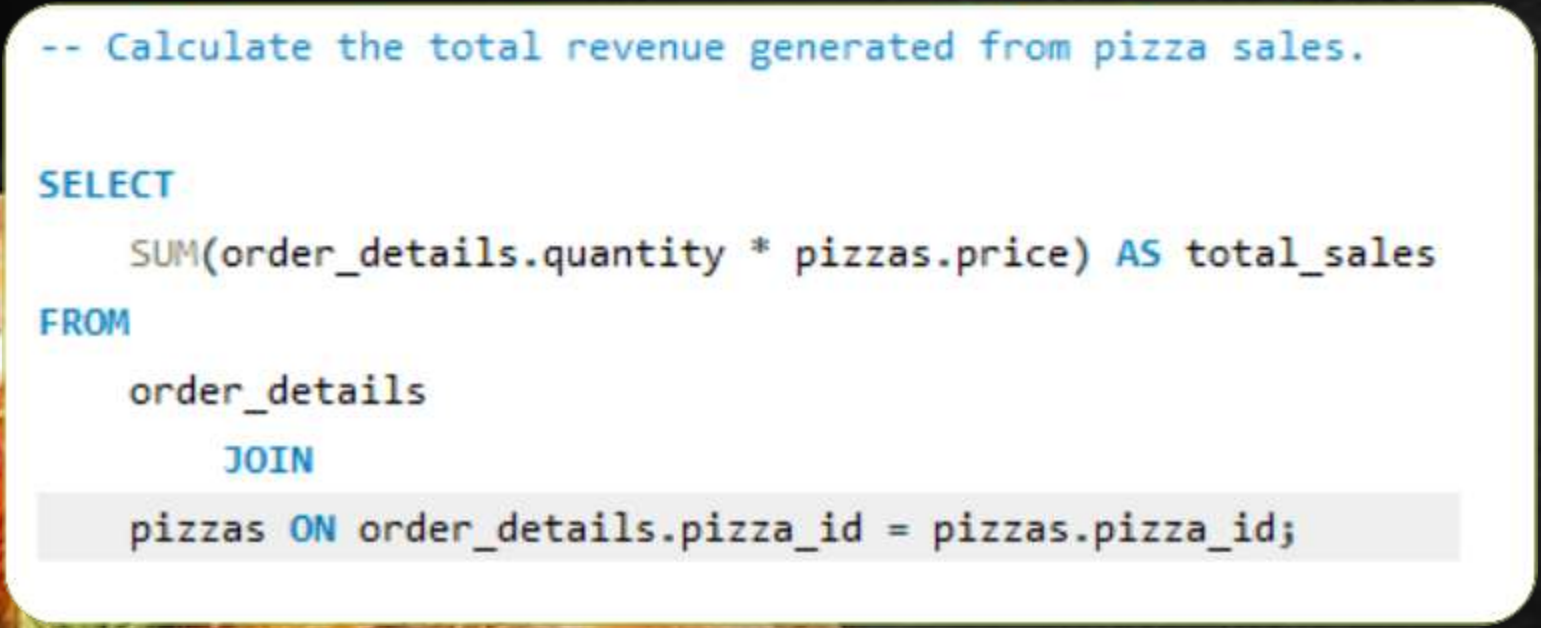


```sql
-- Retrieve the total number of orders placed.

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

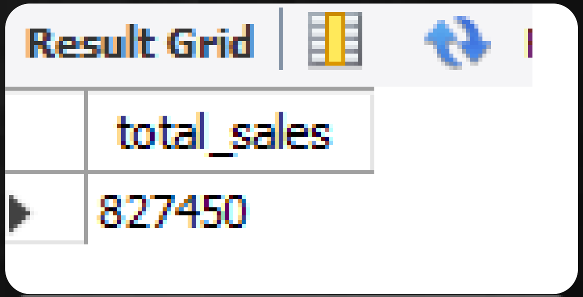


```
total_orders
21350
```

Calculate the total revenue generated from pizza sales

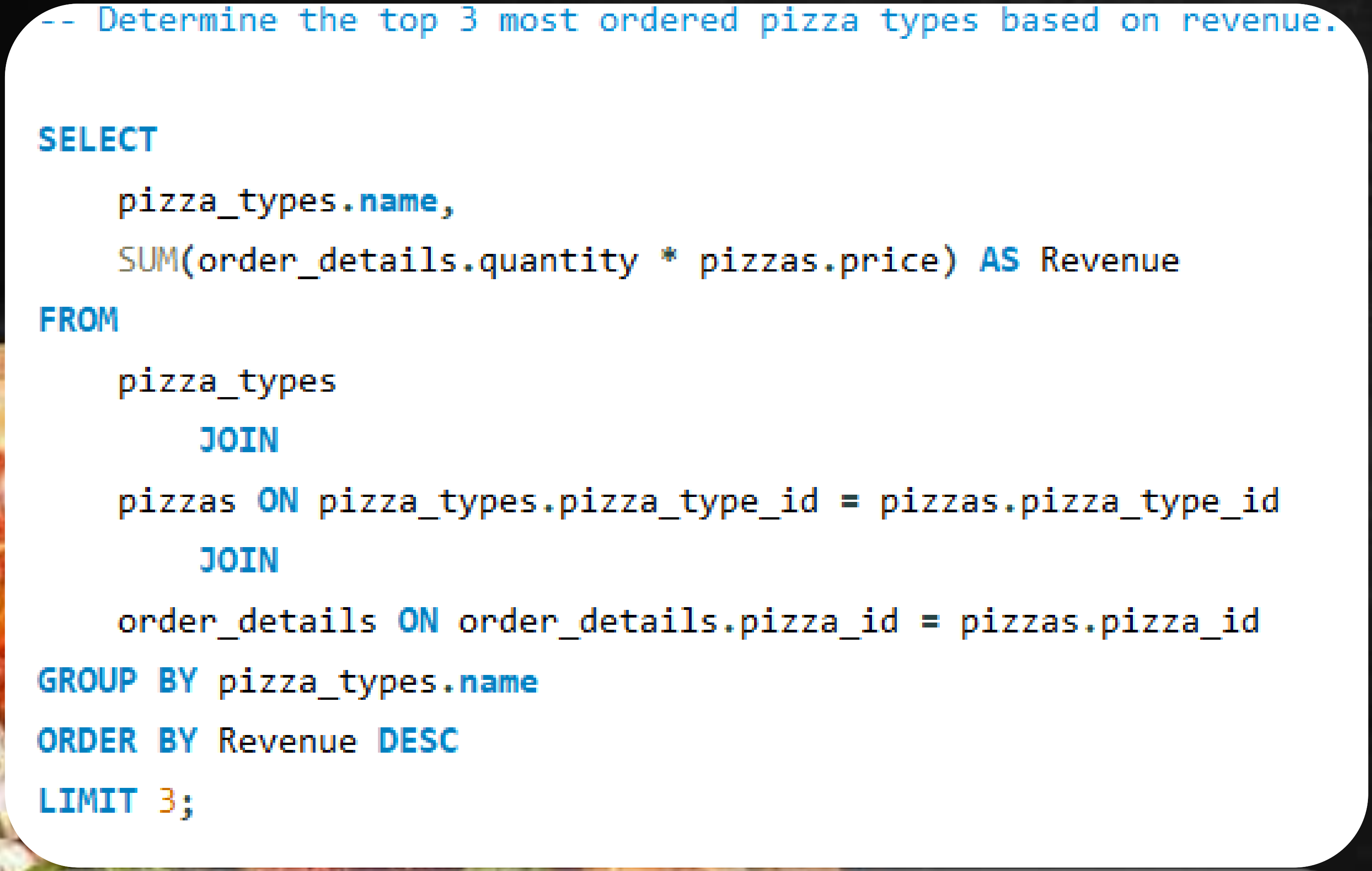


```
-- Calculate the total revenue generated from pizza sales.

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

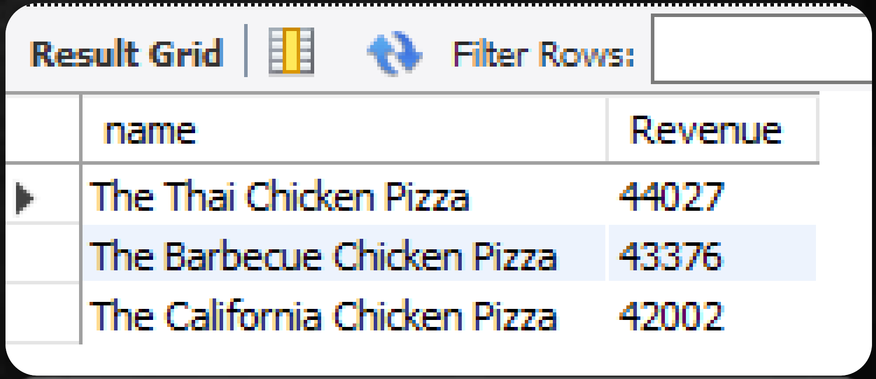


```
total_sales
827450
```

Determine the top 3 most ordered pizza types based on revenue

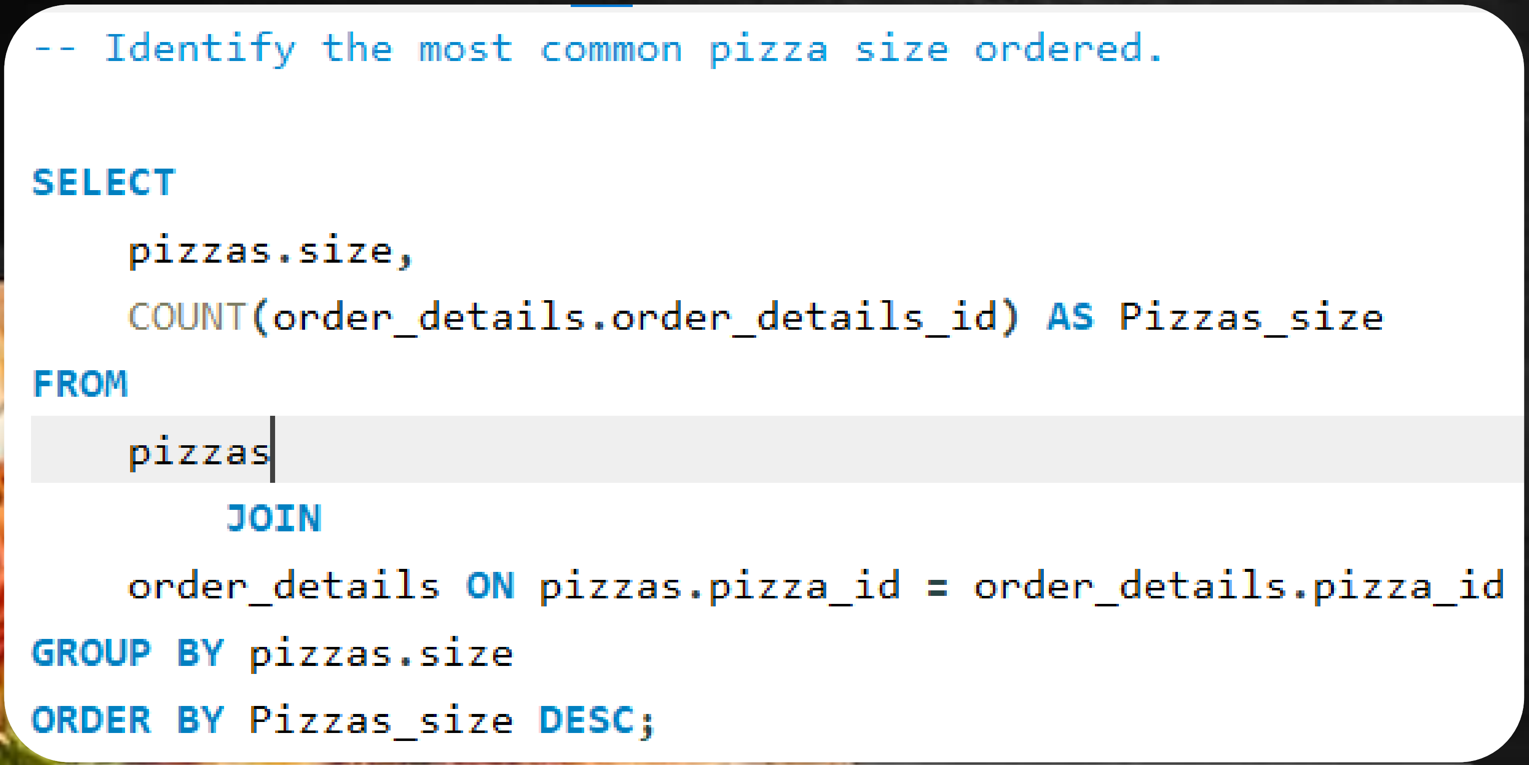


```sql
-- Determine the top 3 most ordered pizza types based on revenue.

SELECT
    pizza_types.name,
    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;
```

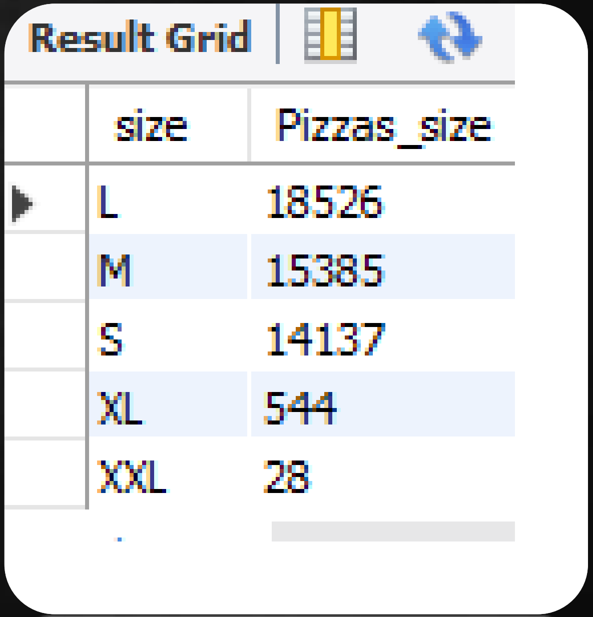


name	Revenue
The Thai Chicken Pizza	44027
The Barbecue Chicken Pizza	43376
The California Chicken Pizza	42002

Identify the most common pizza size ordered

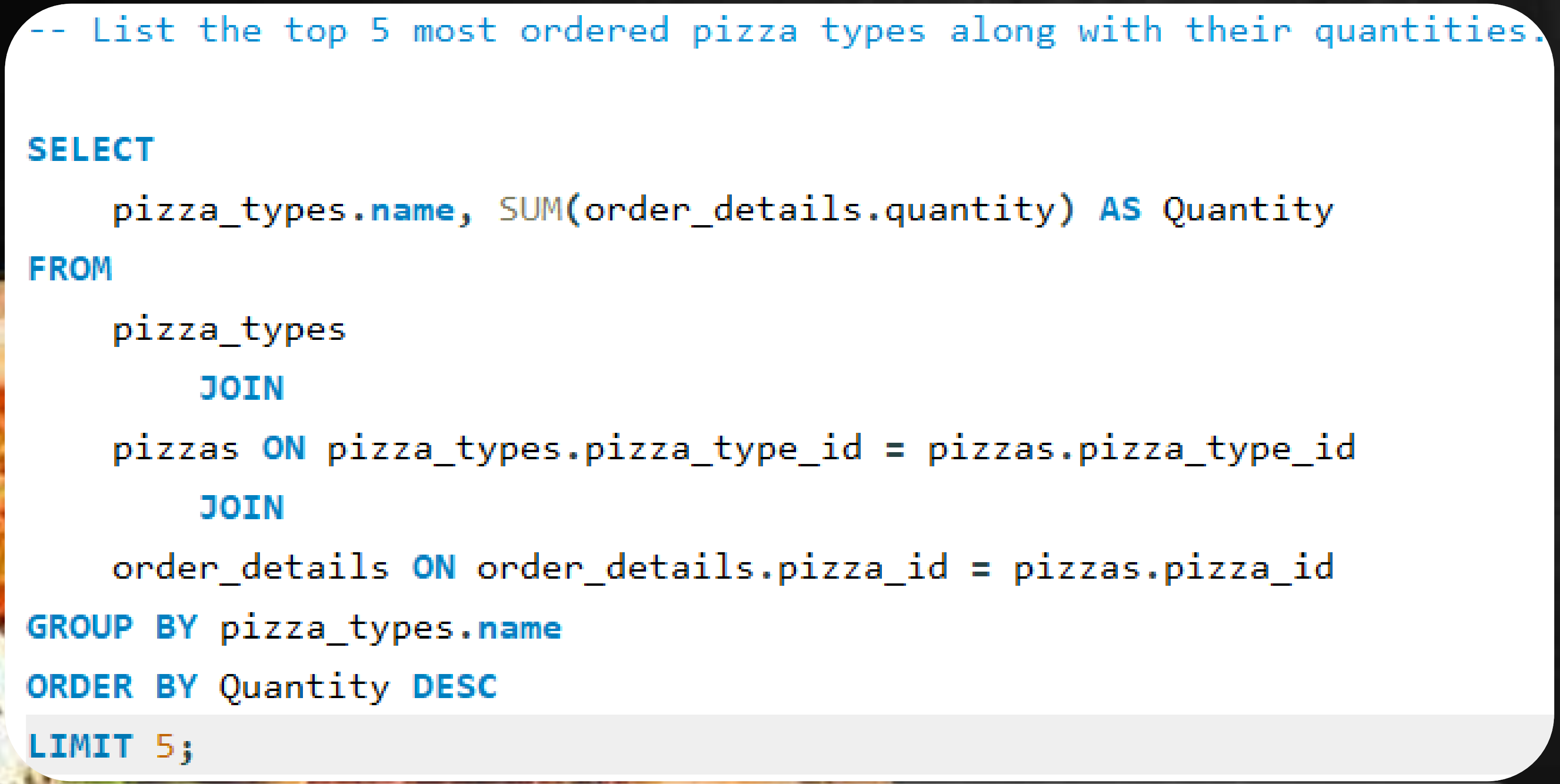


```sql
-- Identify the most common pizza size ordered.

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

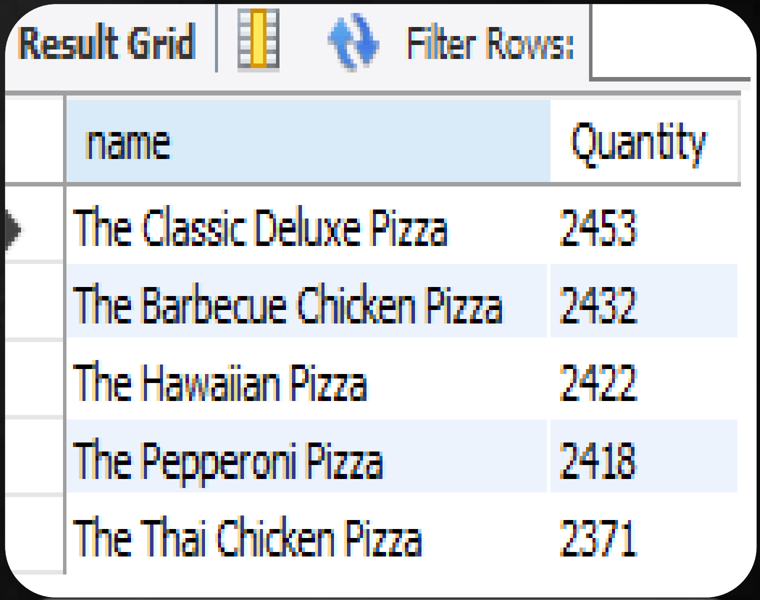


size	Pizzas_size
L	18526
M	15385
S	14137
XL	544
XXL	28

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

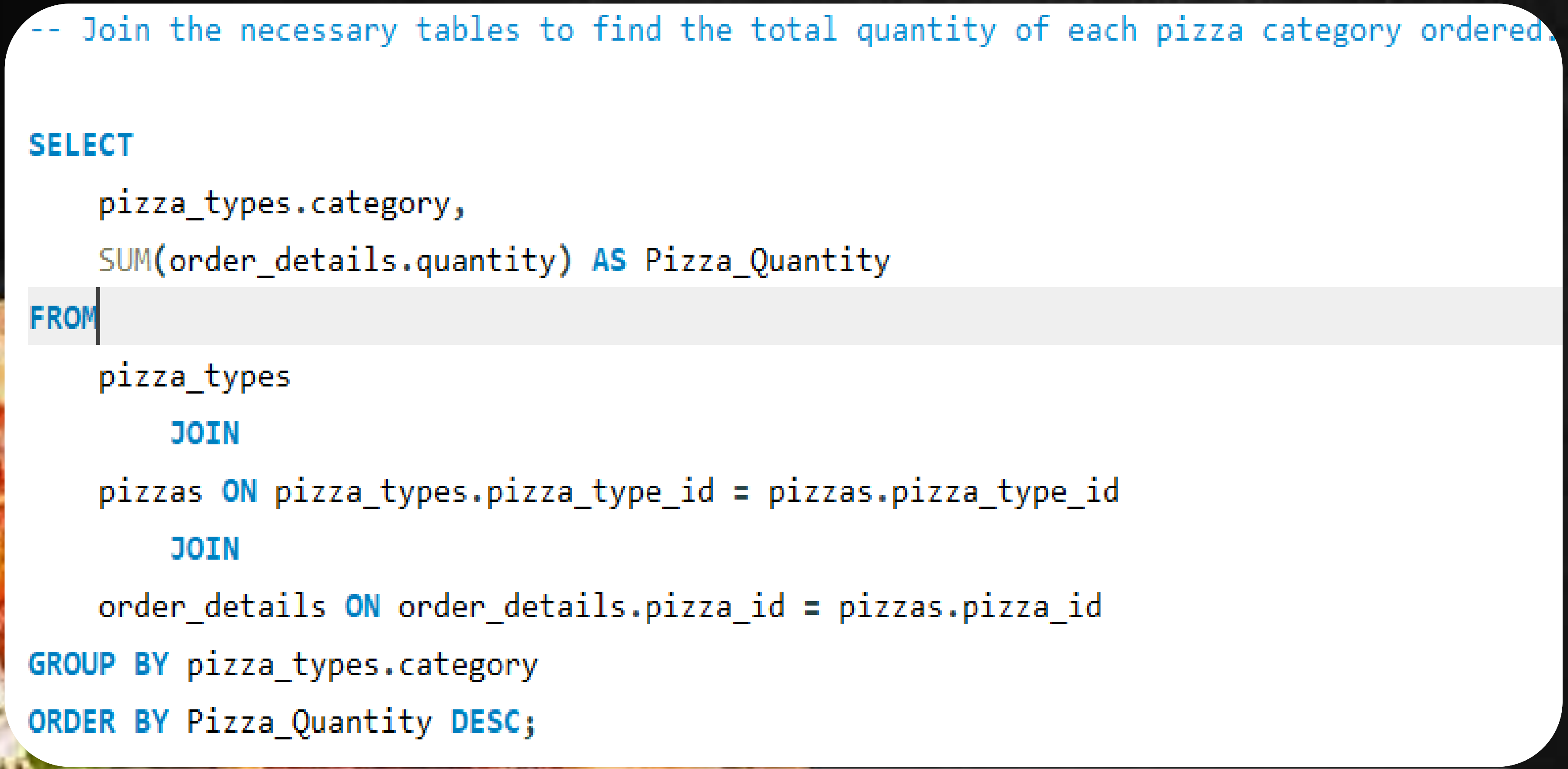


```
-- List the top 5 most ordered pizza types along with their quantities.

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;
```

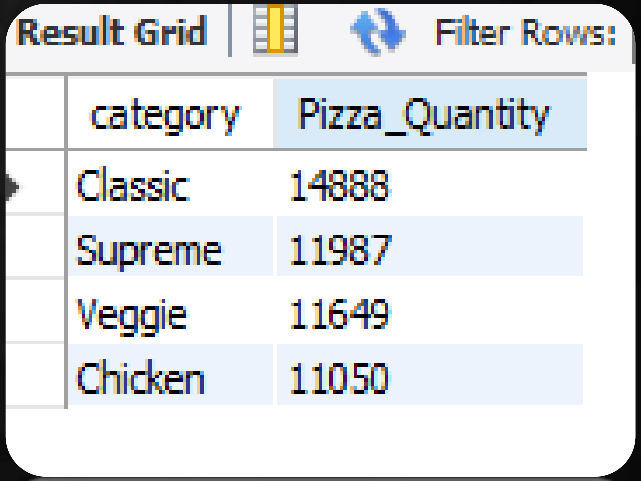


name	Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

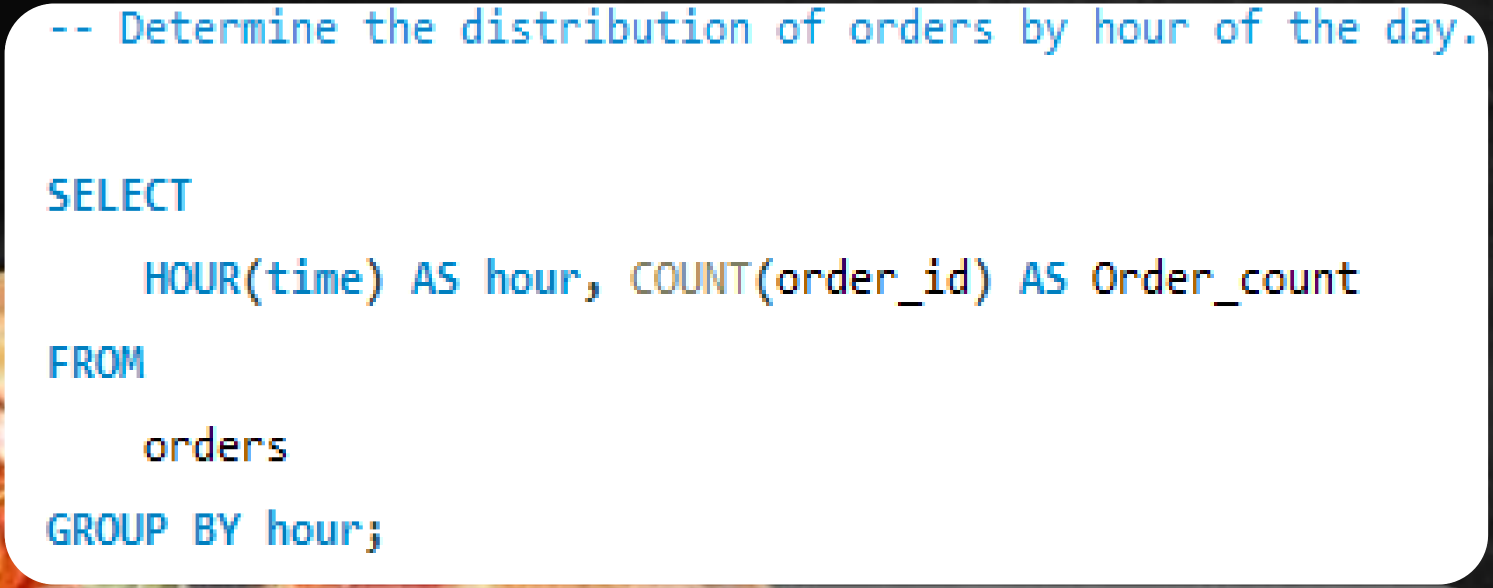


```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.


SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS Pizza_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 Pizza_Quantity DESC;
```

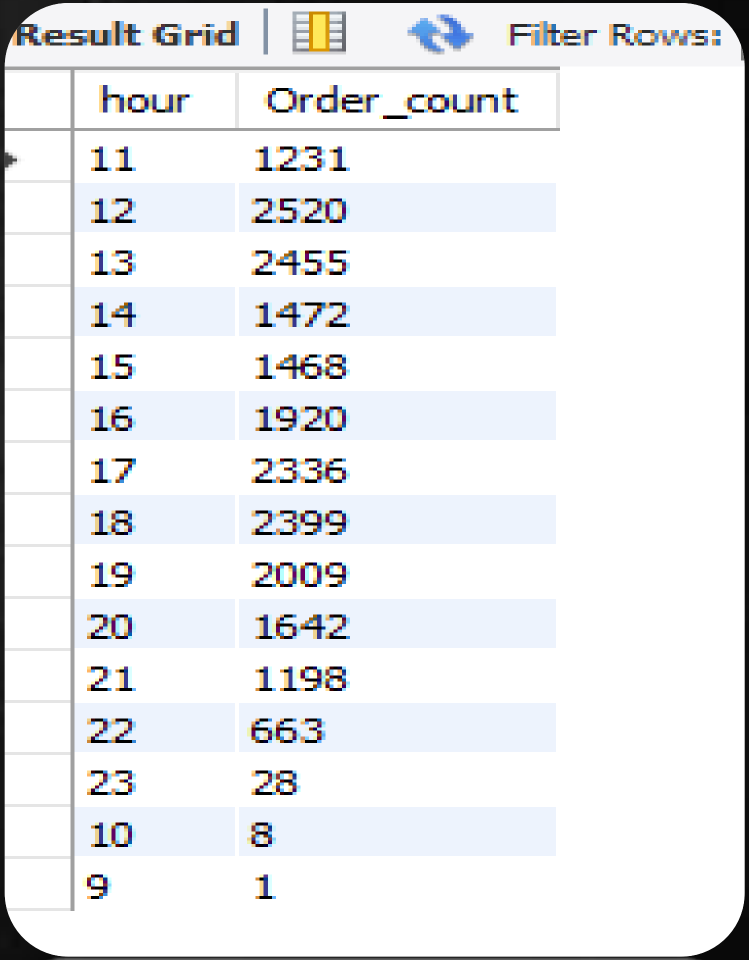


category	Pizza_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

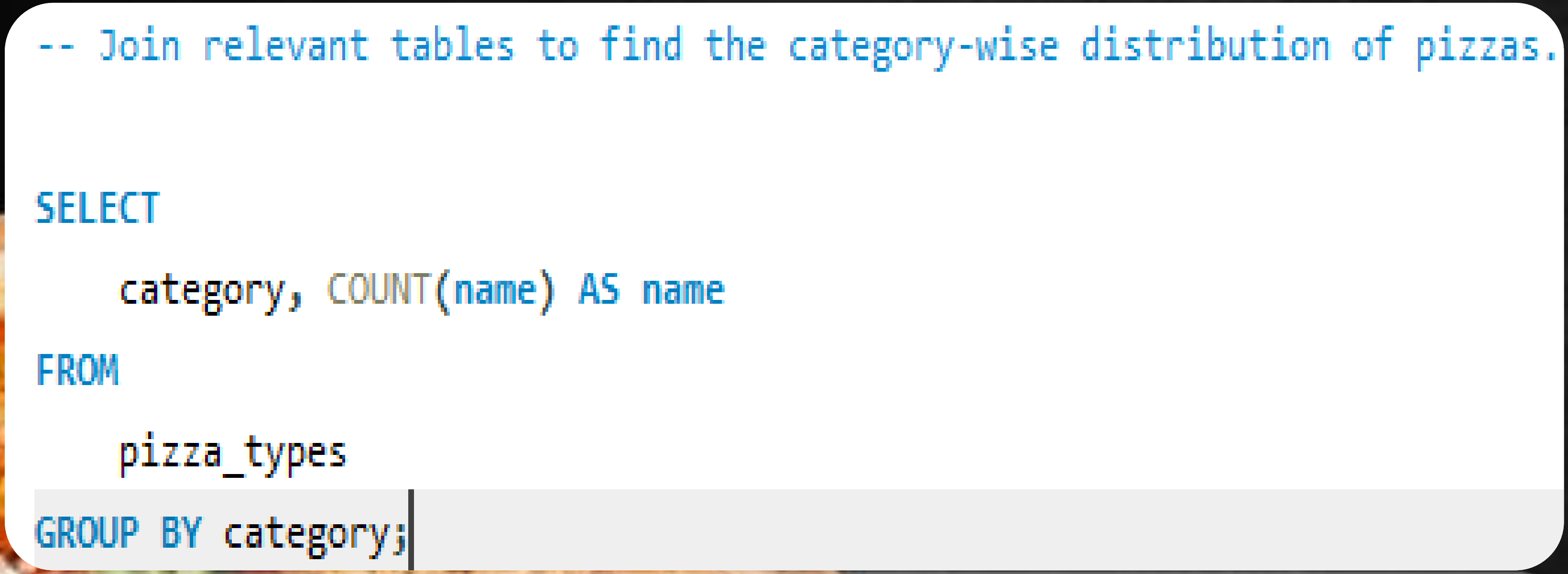


```
-- Determine the distribution of orders by hour of the day.


SELECT
    HOUR(time) AS hour, COUNT(order_id) AS Order_count
FROM
    orders
GROUP BY hour;
```

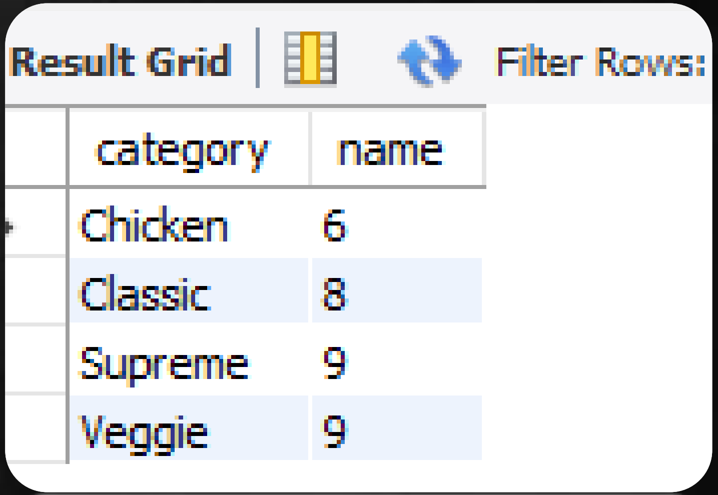


hour	Order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Join relevant tables to find the category wise distribution of pizzas

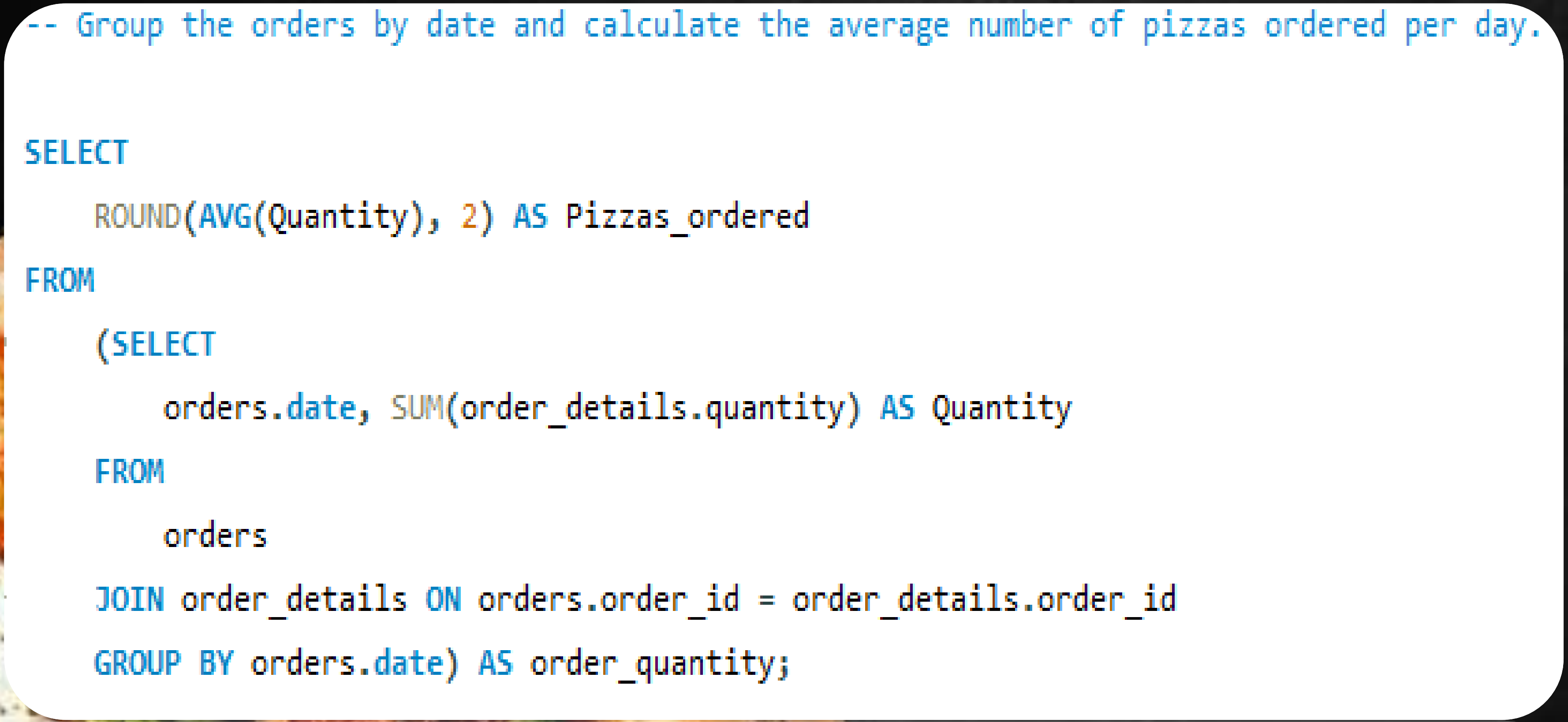


```
-- Join relevant tables to find the category-wise distribution of pizzas.


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

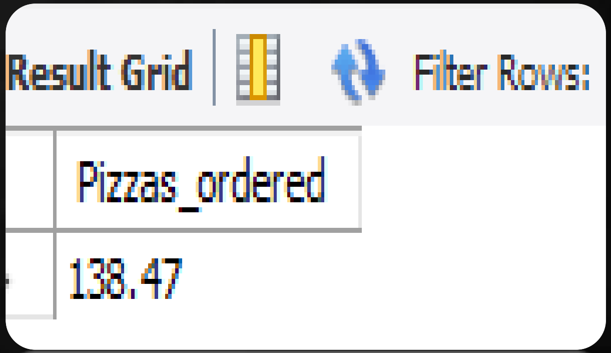


category	name
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

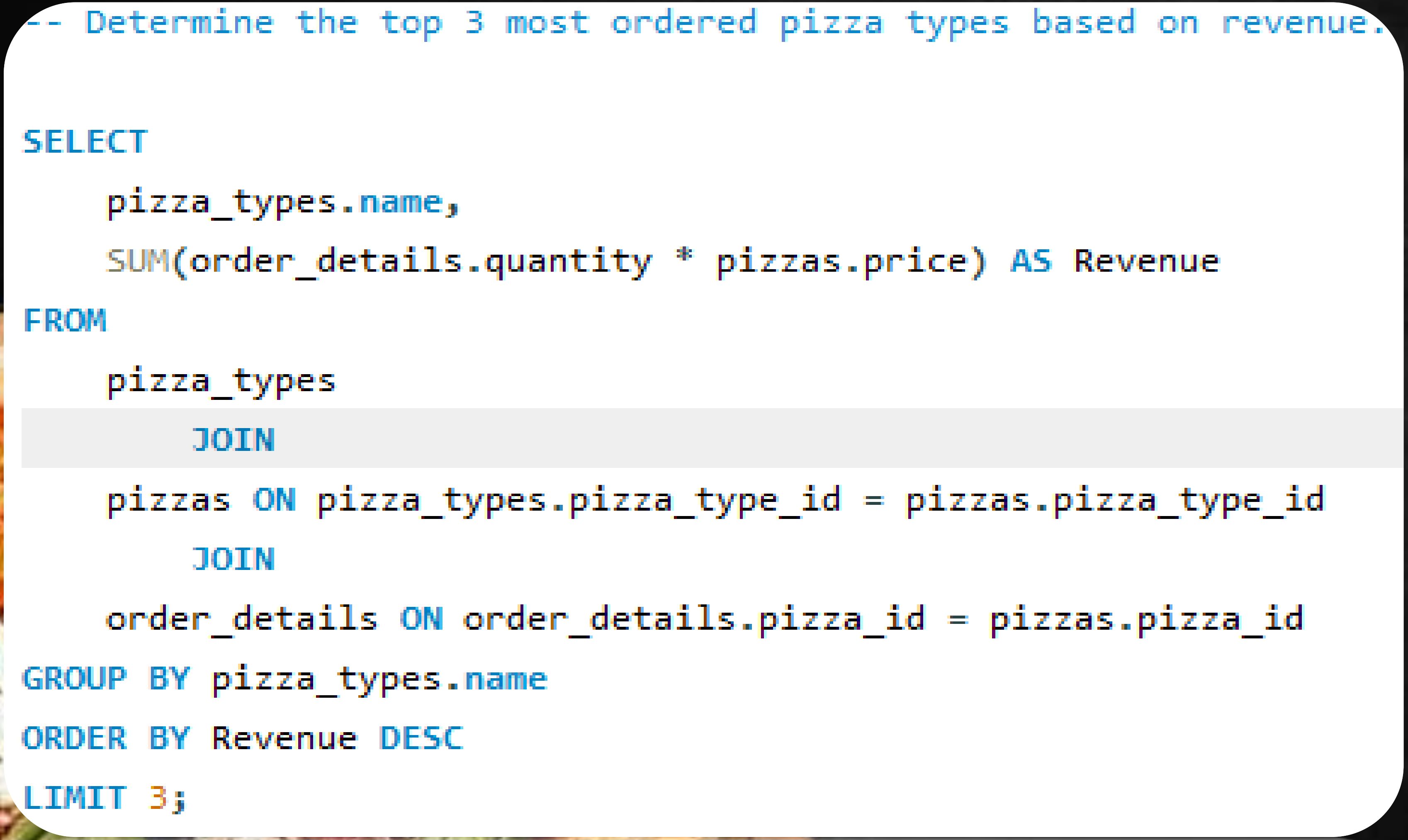


```
-- Group the orders by date and calculate the average number of pizzas ordered per day.


SELECT

    ROUND(AVG(Quantity), 2) AS Pizzas_ordered

FROM

    (SELECT

        orders.date, SUM(order_details.quantity) AS Quantity

    FROM

        orders

    JOIN order_details ON orders.order_id = order_details.order_id

    GROUP BY orders.date) AS order_quantity;
```

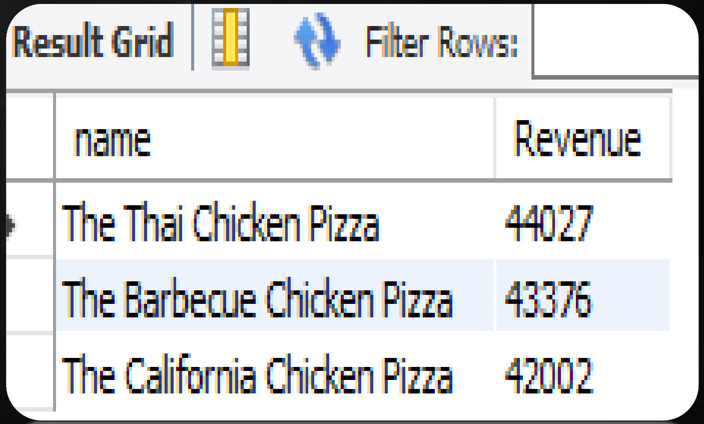


```
Pizzas_ordered
138.47
```

Determine the top 3 most ordered pizza types based on revenue

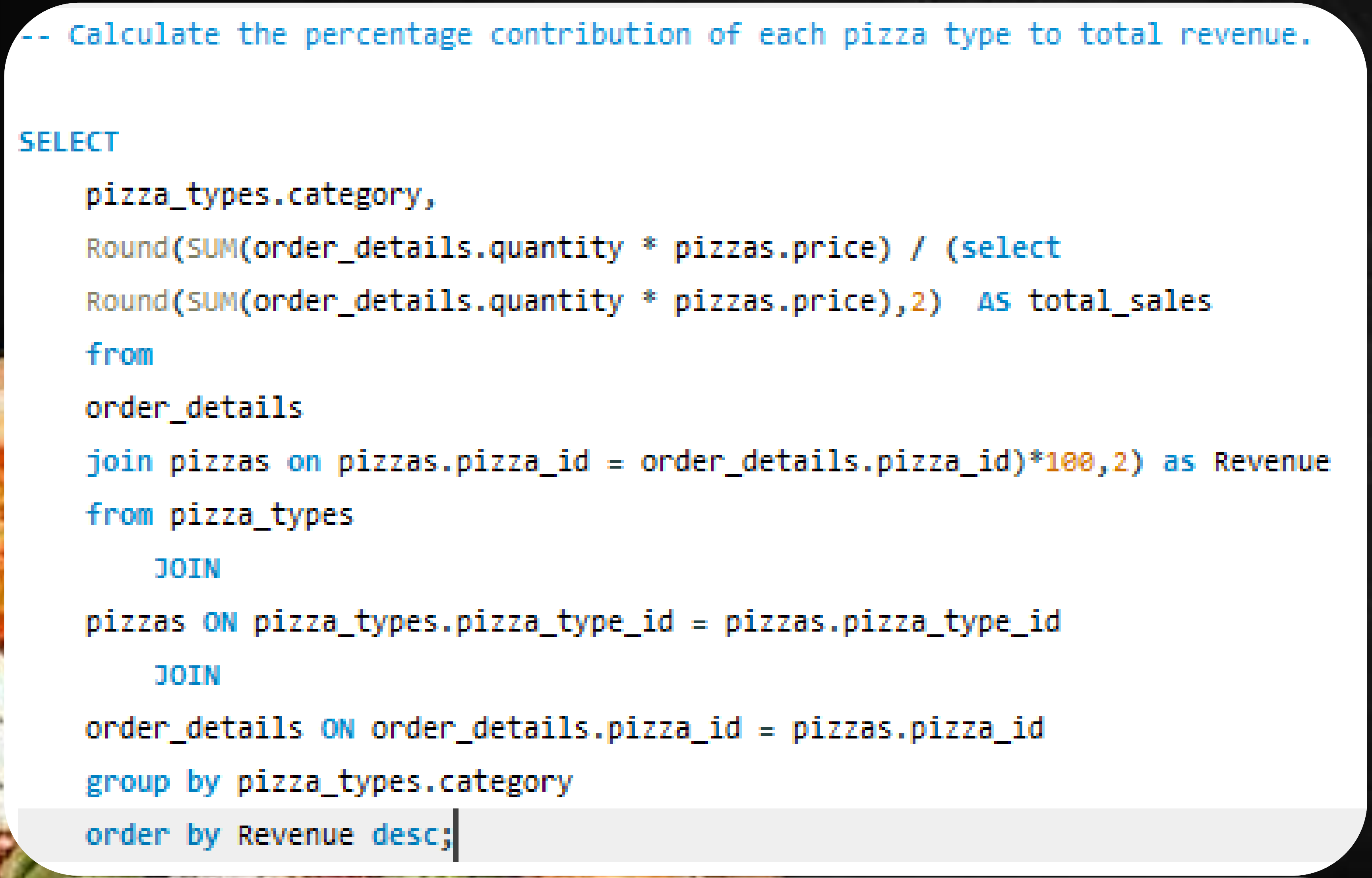


```
-- Determine the top 3 most ordered pizza types based on revenue.

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;
```

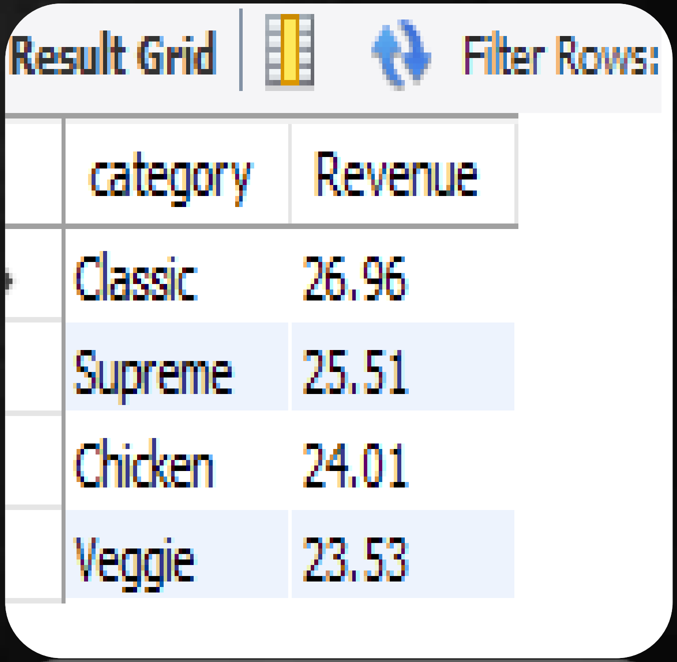


name	Revenue
The Thai Chicken Pizza	44027
The Barbecue Chicken Pizza	43376
The California Chicken Pizza	42002

Calculate the percentage contribution of each pizza type to total revenue

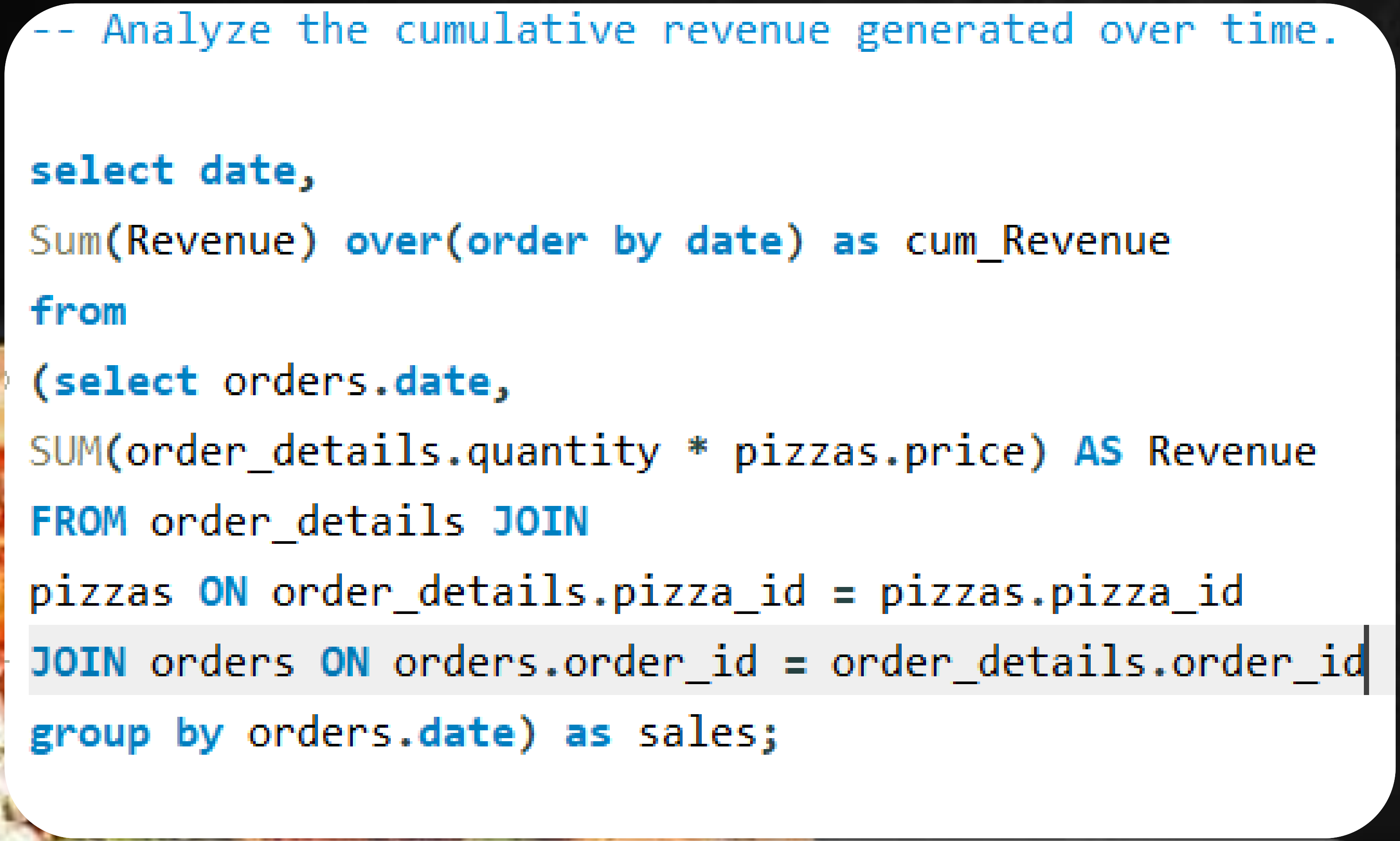


```
-- Calculate the percentage contribution of each pizza type to total revenue.


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 pizzas.pizza_id = order_details.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 desc;
```

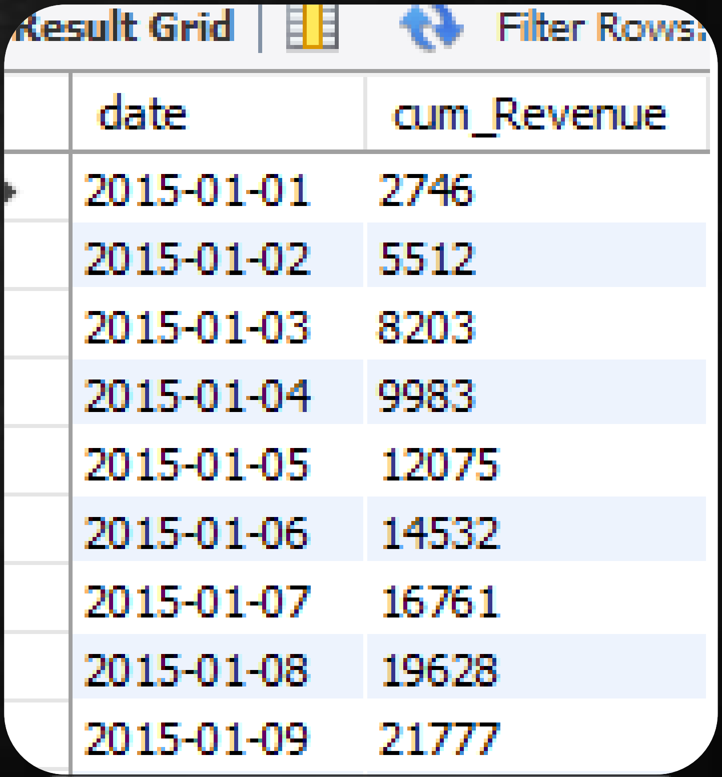


category	Revenue
Classic	26.96
Supreme	25.51
Chicken	24.01
Veggie	23.53

Analyse the cumulative revenue generated over time



```
-- Analyze the cumulative revenue generated over time.


select date,
Sum(Revenue) over(order by date) as cum_Revenue
from
(select orders.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 orders.order_id = order_details.order_id
group by orders.date) as sales;
```



date	cum_Revenue
2015-01-01	2746
2015-01-02	5512
2015-01-03	8203
2015-01-04	9983
2015-01-05	12075
2015-01-06	14532
2015-01-07	16761
2015-01-08	19628
2015-01-09	21777

Thank You !

For much such content you can follow me on
 LinkedIn