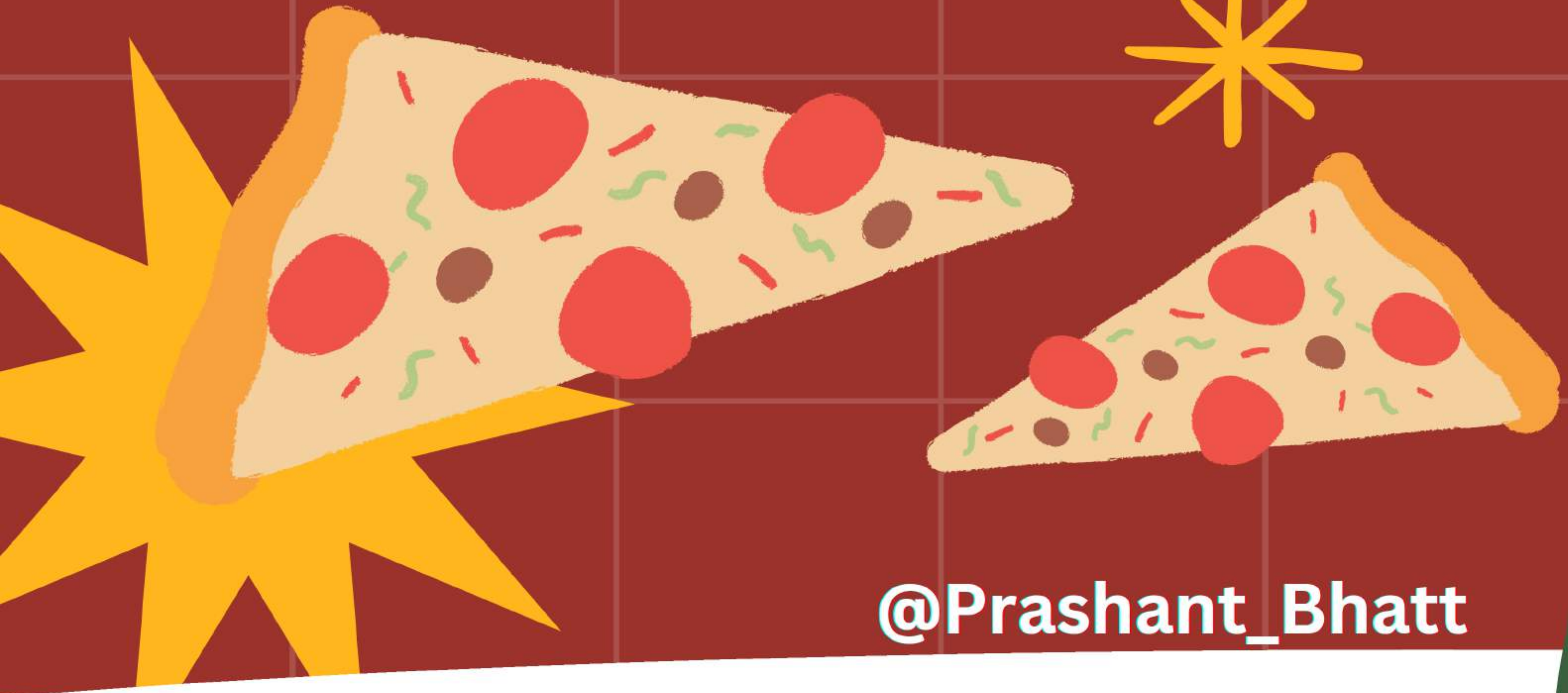
PIZZA SALES ANALYSIS



@Prashant_Bhatt

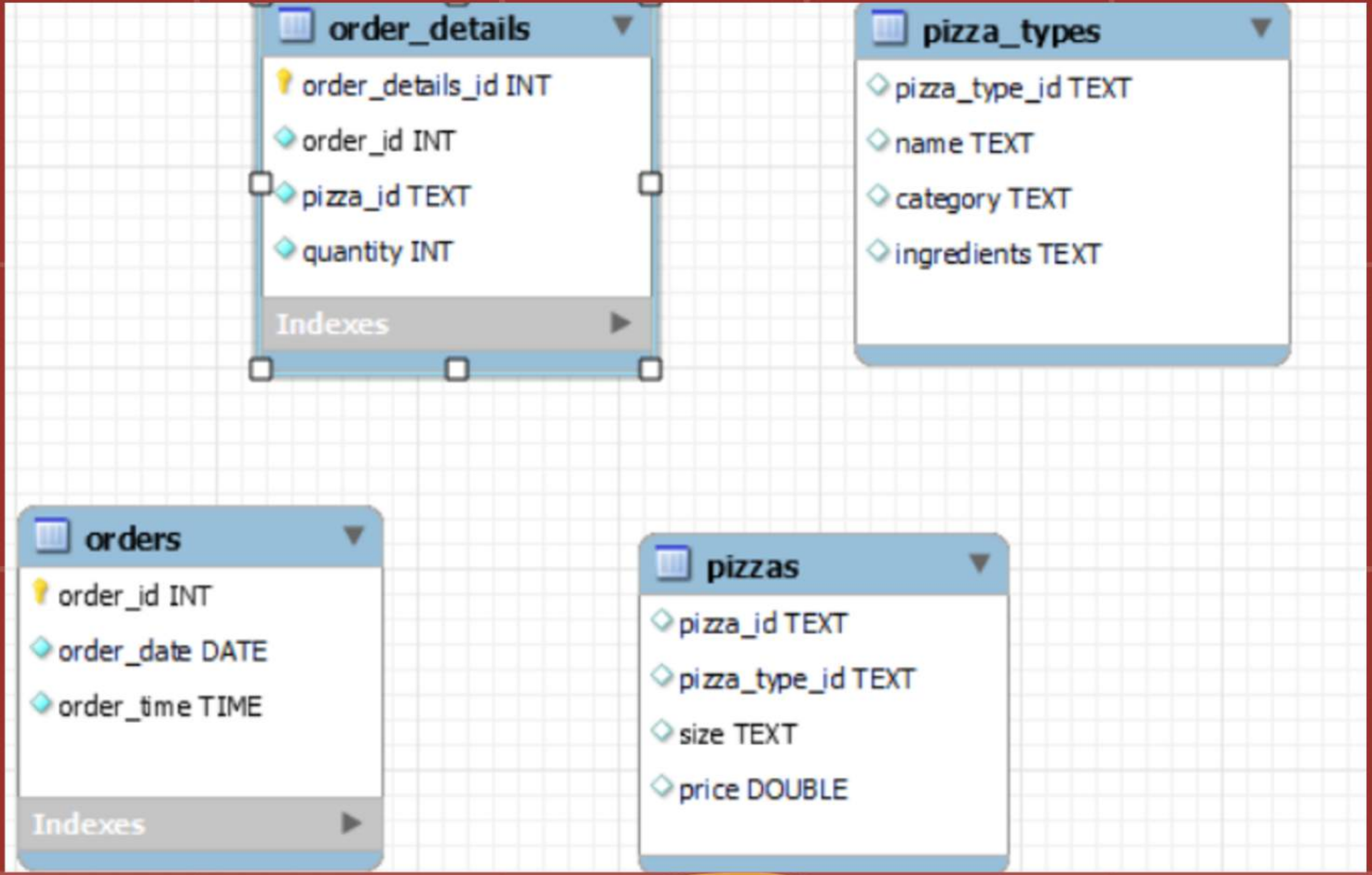
Project Overview

The Pizza Sales Analysis project leverages SQL to explore sales patterns,customer,preferences,and business performance of a pizza resturant.Basic analysis includes calculating total order,revenue,identifing the highest-priced pizza. The most common pizza size.,and the top 5 most orderedpizza_types. **Intermediate analysis** invoilves joiniong tables to find total quantities of each pizza category. distribution of orders by hour,category-wise pizza distribution,daily average orders,and top 3 pizza type of revenue.

Advanced Analysis calculate each pizza type’s revenue contribution,cumulative revenue over time,and top3 pizza types by revenue within each category.This insights guide strategic decisions to optimize operations and boost profitablity.

SCHEMA

Entity Relationship Diagram



order_details
- order_details_id INT
- order_id INT
- pizza_id TEXT
- quantity INT

Indexes

pizza_types
- pizza_type_id TEXT
- name TEXT
- category TEXT
- ingredients TEXT

orders
- order_id INT
- order_date DATE
- order_time TIME

Indexes

pizzas
- pizza_id TEXT
- pizza_type_id TEXT
- size TEXT
- price DOUBLE

ANALYSIS QUESTIONS

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.

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

7.Determine the distribution of orders by hour of the day.

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

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

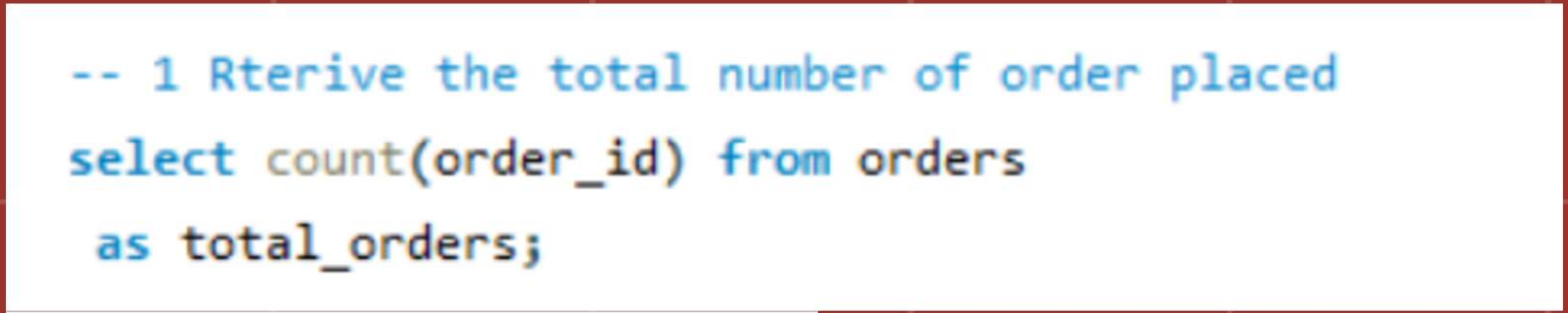
10.Determine the top 3 most ordered pizza types based on revenue.

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

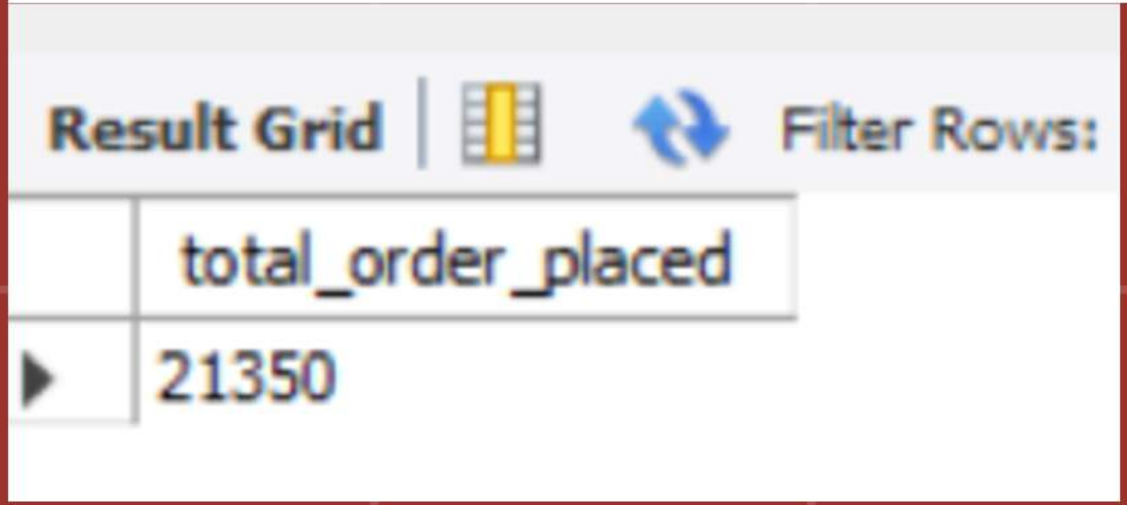
12.Analyze the cumulative revenue generated over time.

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

1.Retrive the total number of order Placed

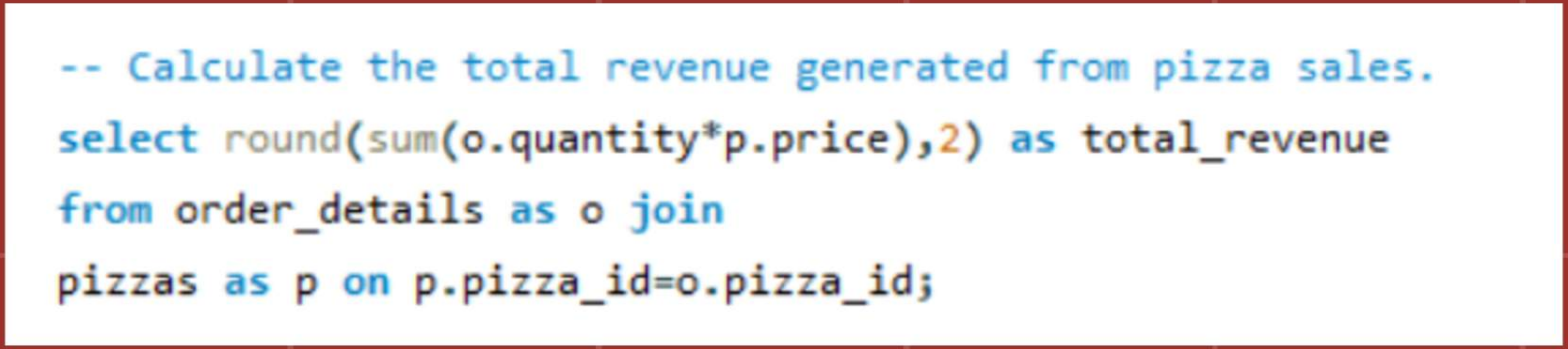


```
-- 1 Rterive the total number of order placed
select count(order_id) from orders
 as total_orders;
```

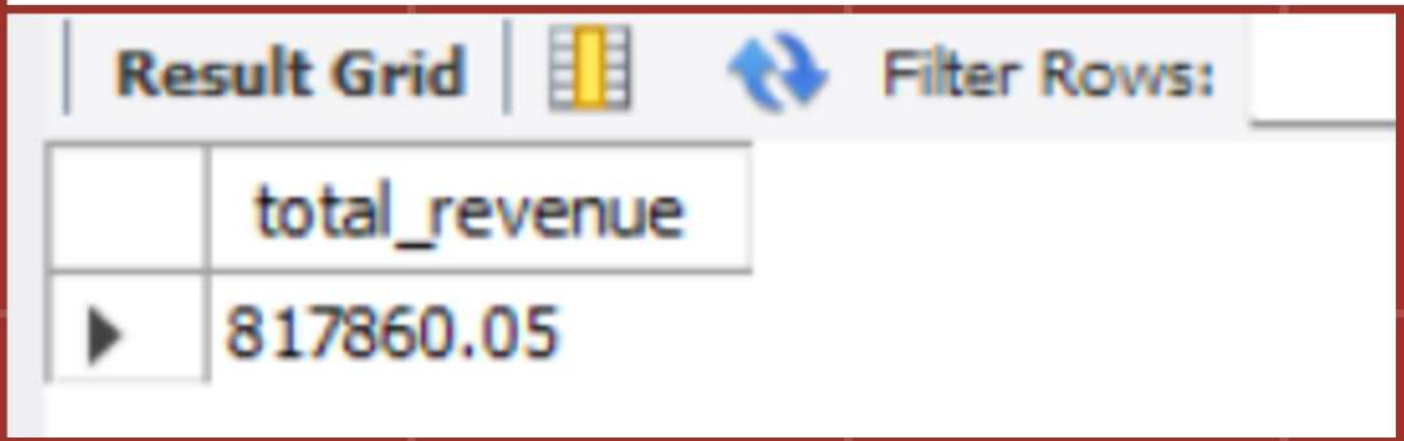


```
total_order_placed
21350
```

2.Calculate the total revenue generated from pizza sales.

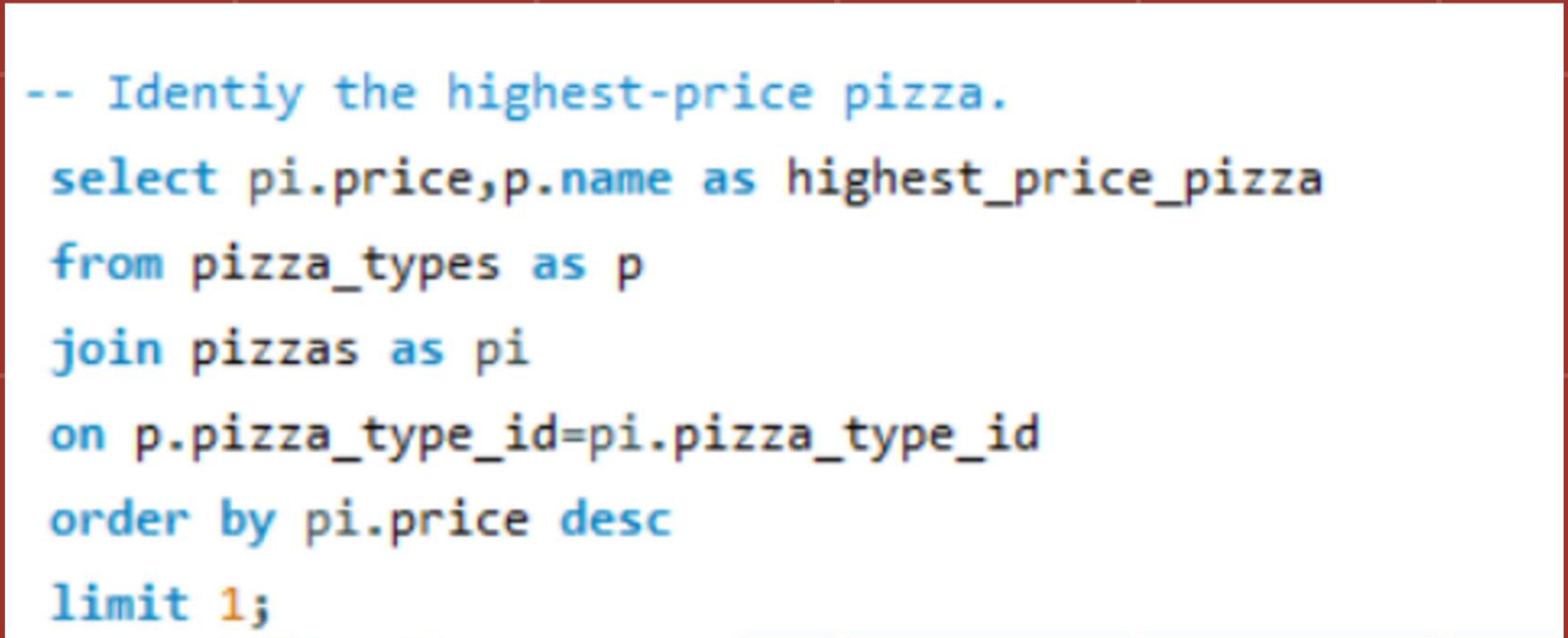


```sql
-- Calculate the total revenue generated from pizza sales.
select round(sum(o.quantity*p.price),2) as total_revenue
from order_details as o join
pizzas as p on p.pizza_id=o.pizza_id;
```

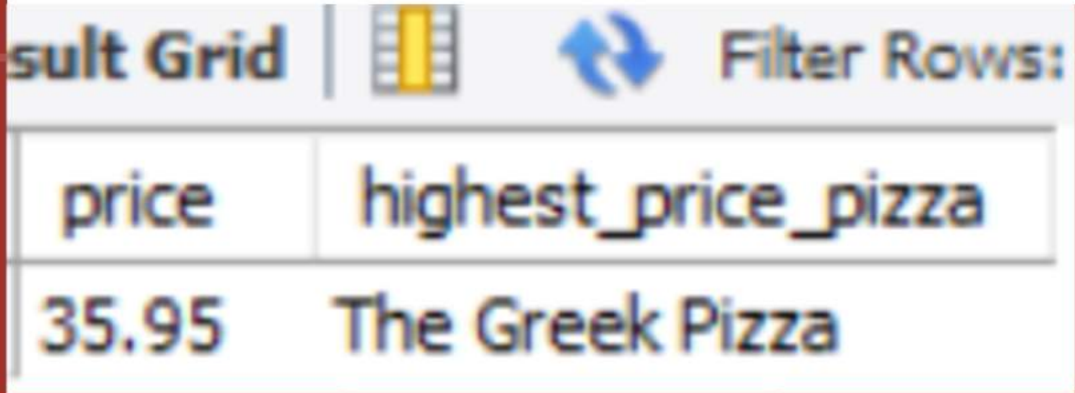


	total_revenue
▶	817860.05

3.Identify the highest-priced pizza.

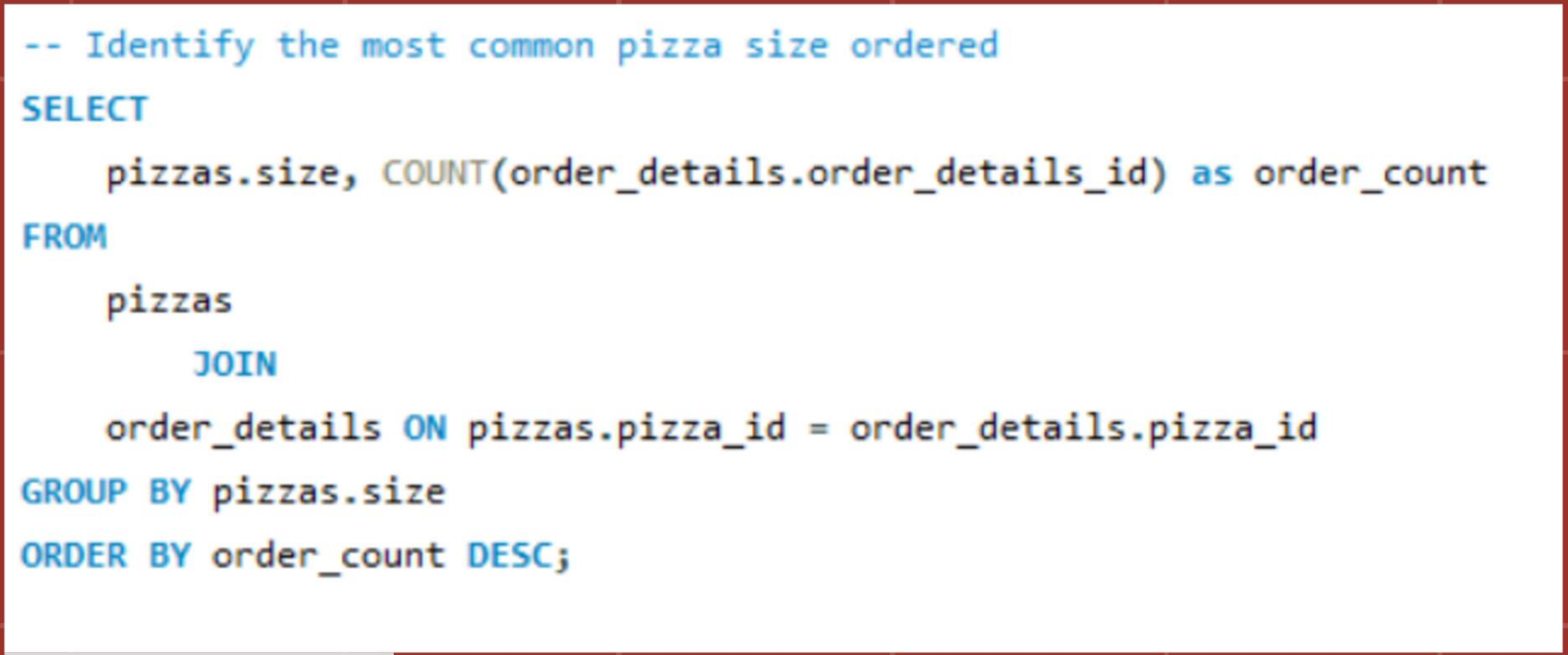


```sql
-- Identiy the highest-price pizza.
 select pi.price,p.name as highest_price_pizza
 from pizza_types as p
 join pizzas as pi
 on p.pizza_type_id=pi.pizza_type_id
 order by pi.price desc
 limit 1;
```

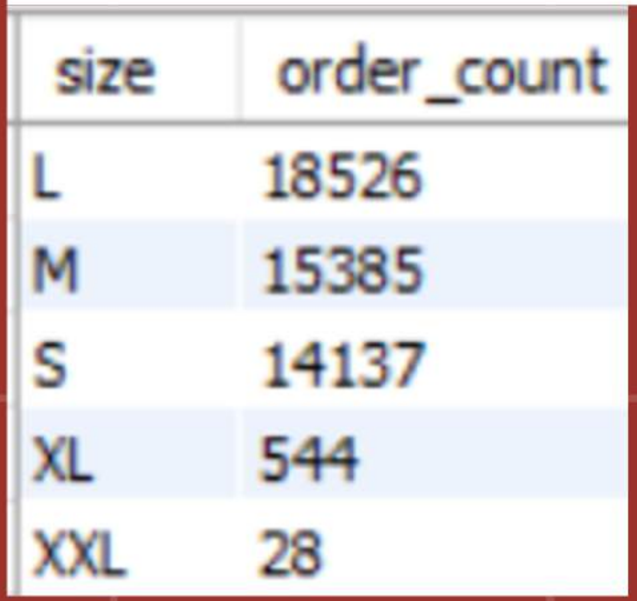


price	highest_price_pizza
35.95	The Greek Pizza

4.Identify the most common pizza size ordered.

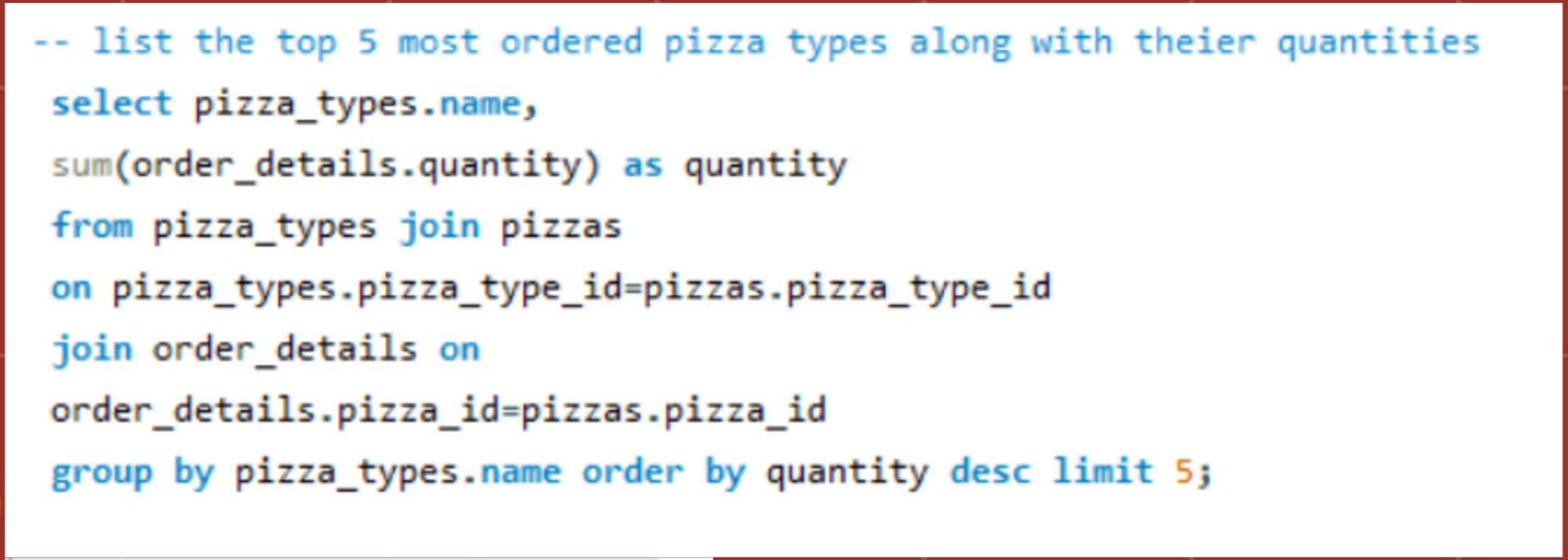


```
-- Identify the most common pizza size ordered
SELECT
    pizzas.size, COUNT(order_details.order_details_id) as order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

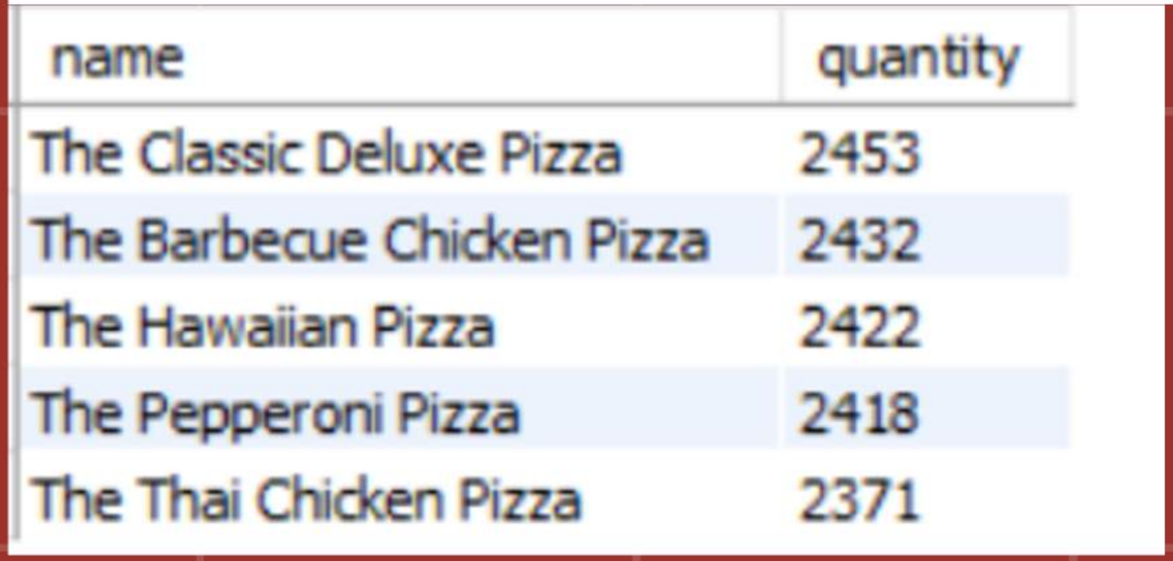


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

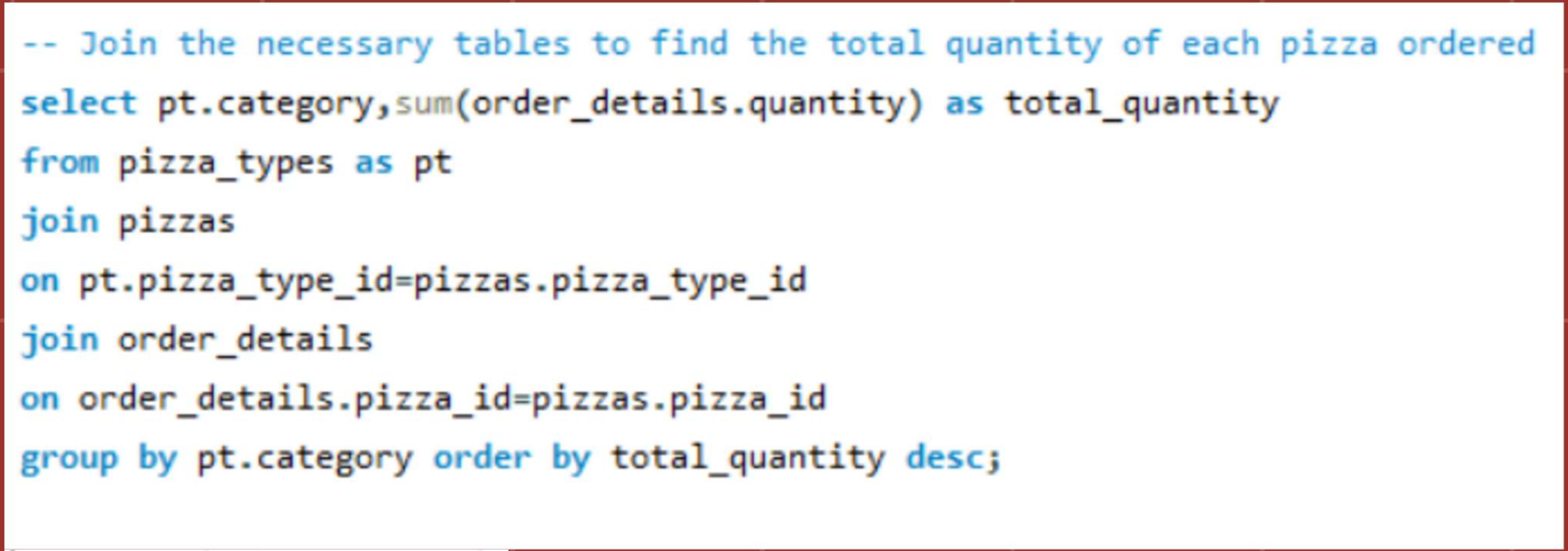


```
-- list the top 5 most ordered pizza types along with theier quantities
 select pizza_types.name,
 sum(order_details.quantity) as quantity
 from pizza_types join pizzas
 on pizza_types.pizza_type_id=pizzas.pizza_type_id
 join order_details on
 order_details.pizza_id=pizzas.pizza_id
 group by pizza_types.name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

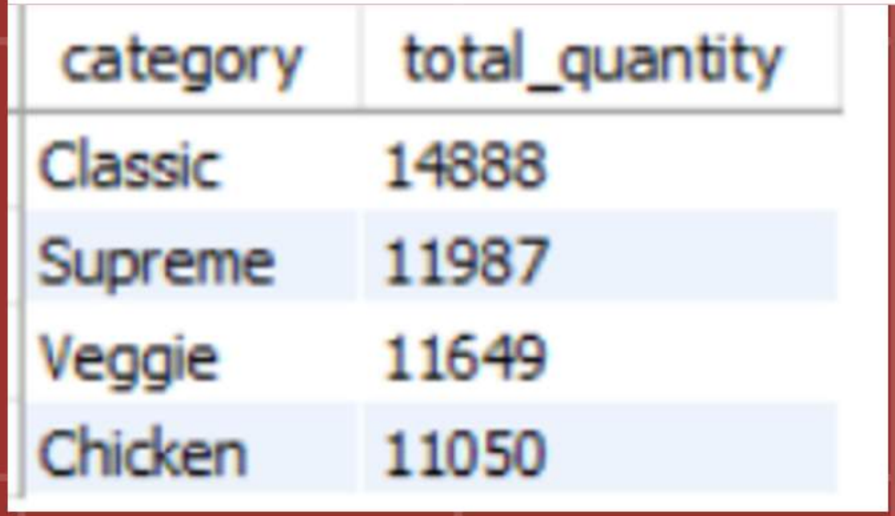


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

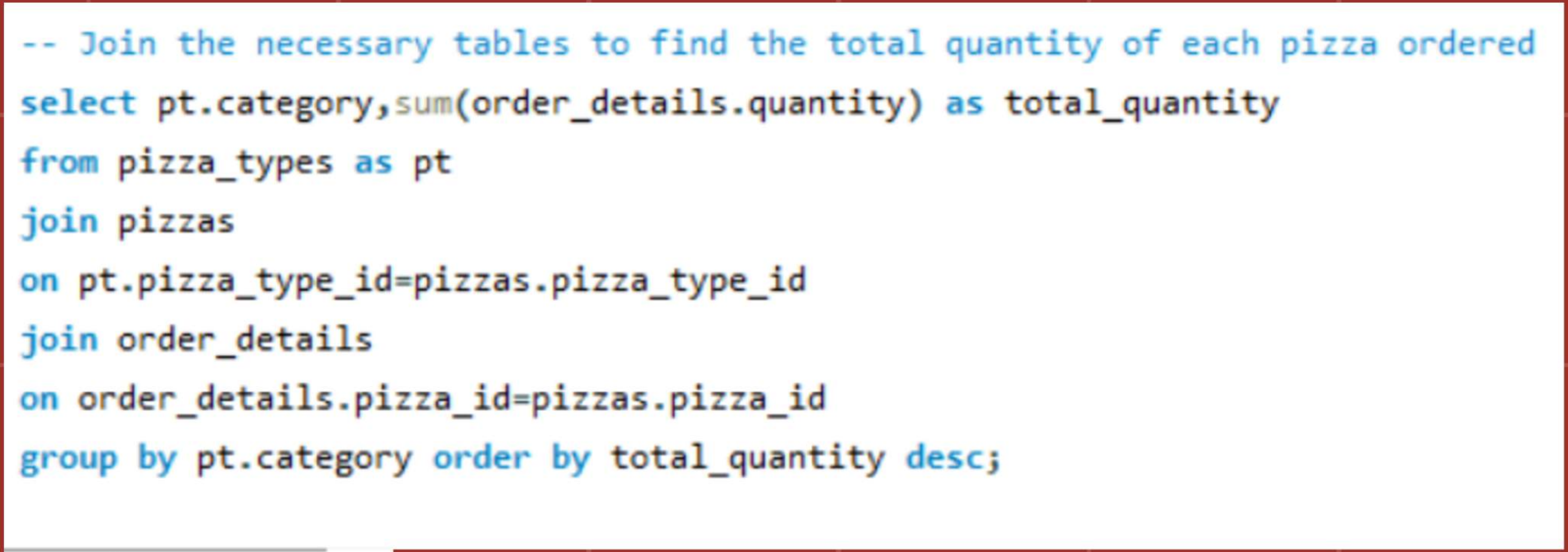


```
-- Join the necessary tables to find the total quantity of each pizza ordered
select pt.category,sum(order_details.quantity) as total_quantity
from pizza_types as pt
join pizzas
on pt.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pt.category order by total_quantity desc;
```

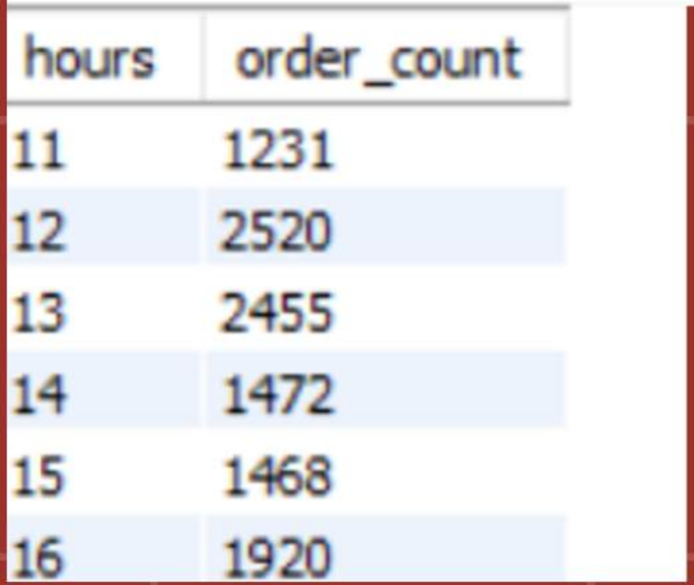


category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7.Determine the distribution of orders by hour of the day.

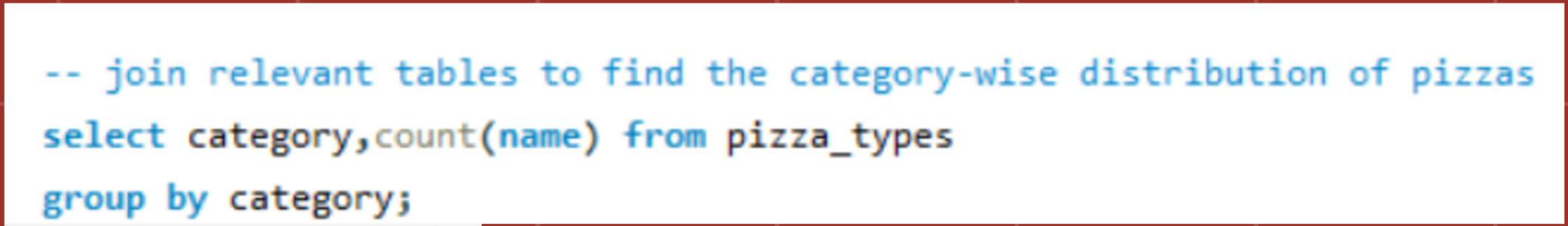


```
-- Join the necessary tables to find the total quantity of each pizza ordered
select pt.category,sum(order_details.quantity) as total_quantity
from pizza_types as pt
join pizzas
on pt.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pt.category order by total_quantity desc;
```

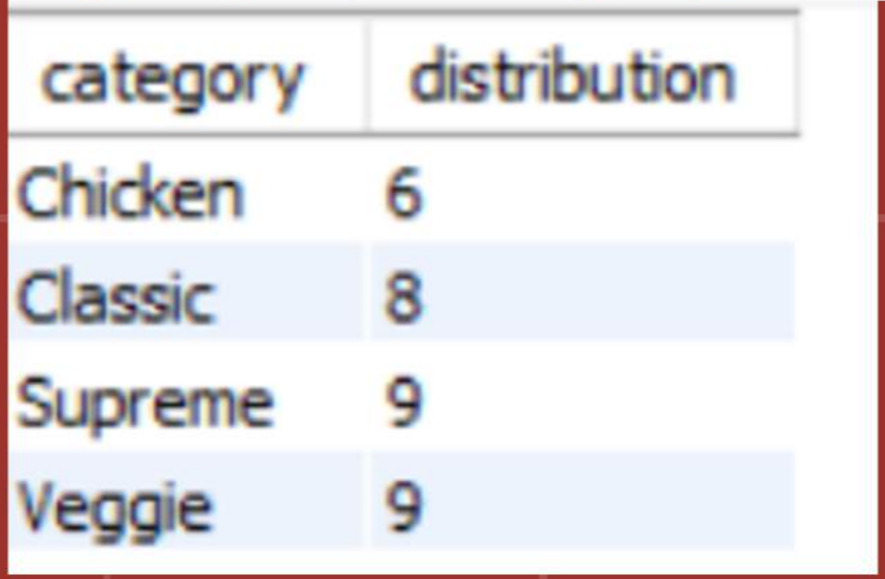


hours	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

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

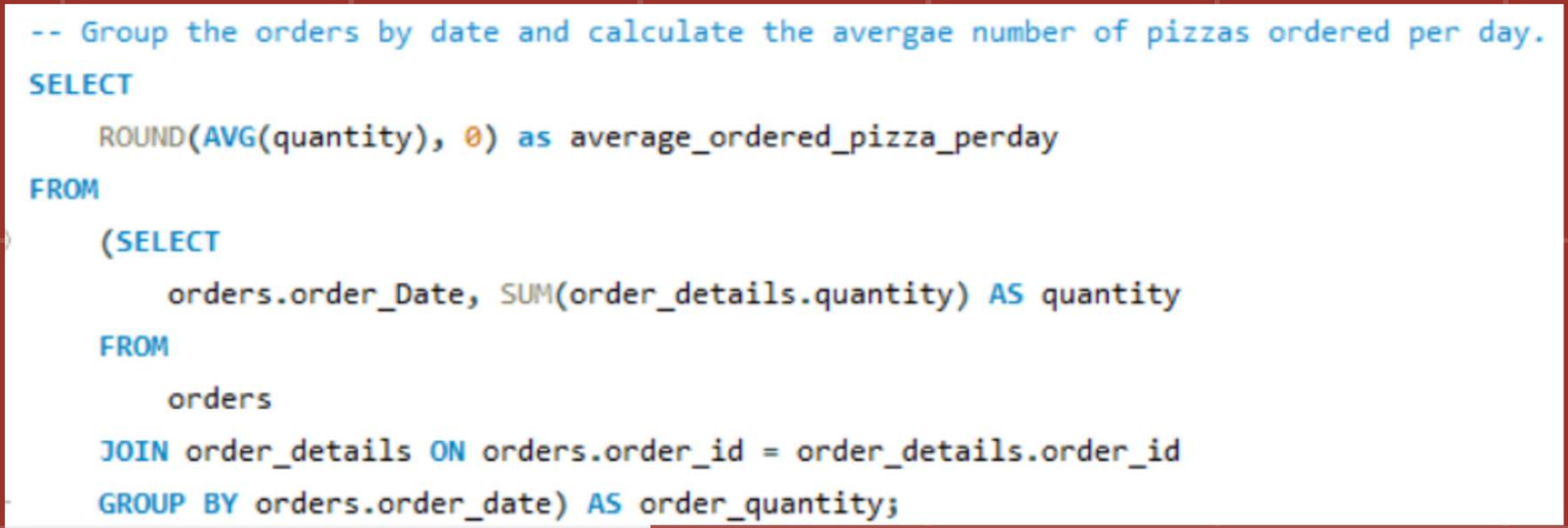


```
-- join relevant tables to find the category-wise distribution of pizzas
select category,count(name) from pizza_types
group by category;
```

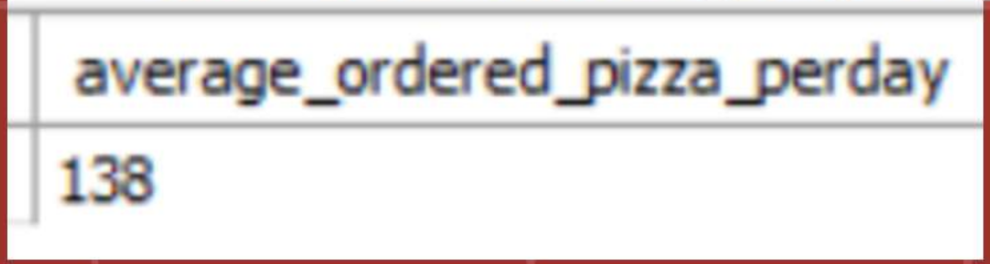


category	distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

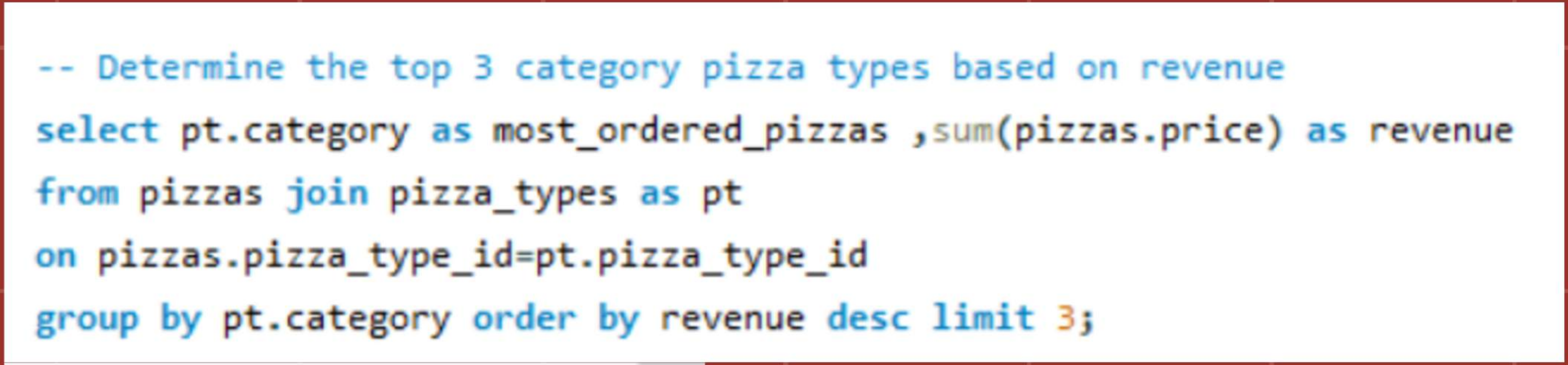


```
-- Group the orders by date and calculate the avergae number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) as average_ordered_pizza_perday
FROM
    (SELECT
        orders.order_Date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

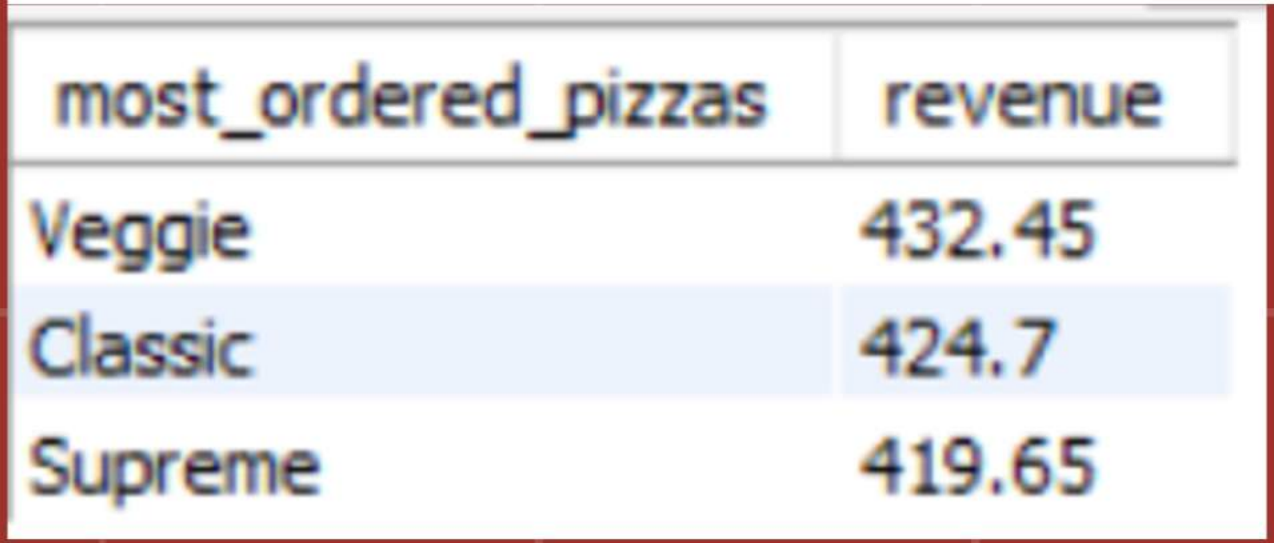


average_ordered_pizza_perday
138

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

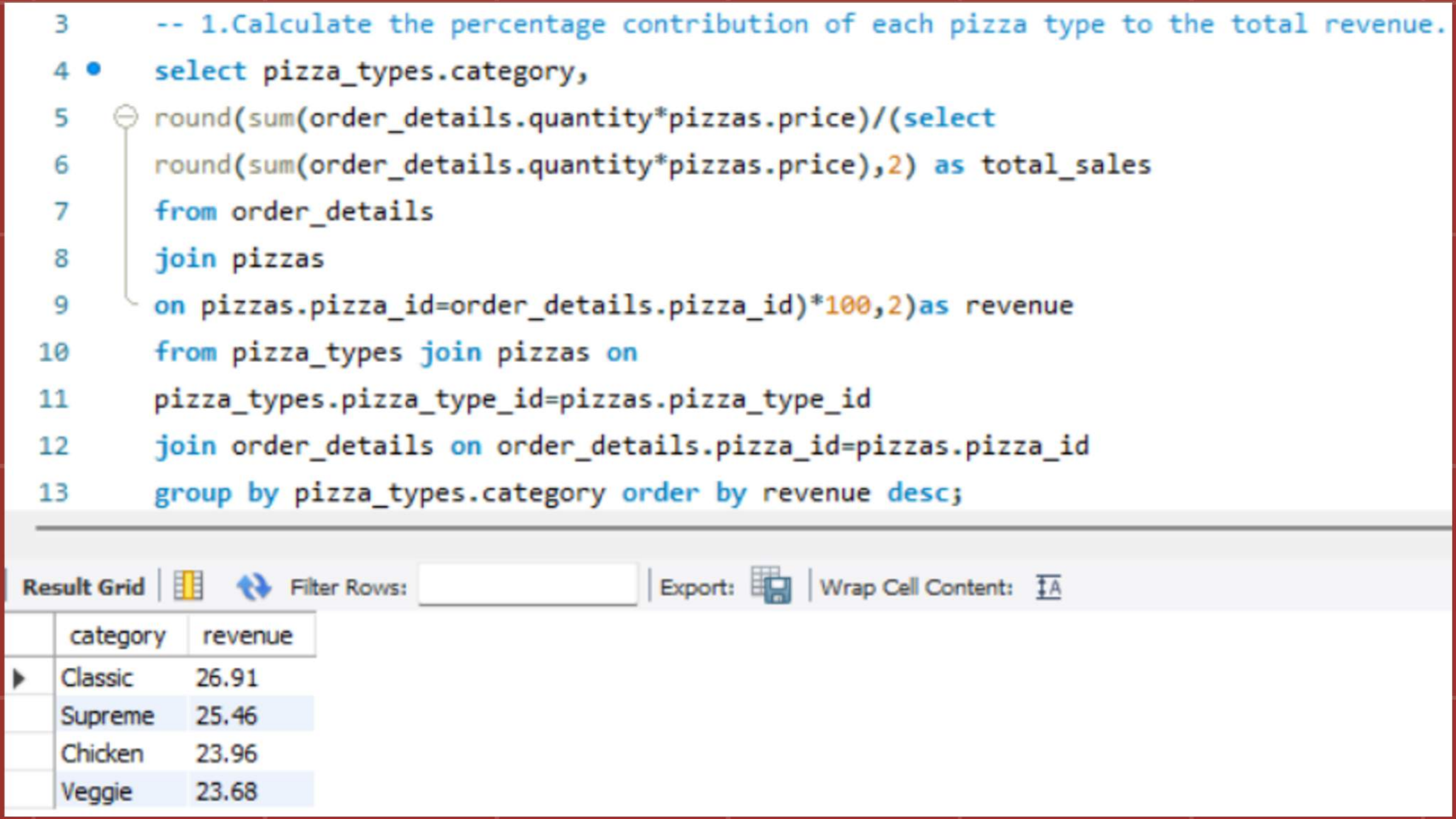


```sql
-- Determine the top 3 category pizza types based on revenue
select pt.category as most_ordered_pizzas ,sum(pizzas.price) as revenue
from pizzas join pizza_types as pt
on pizzas.pizza_type_id=pt.pizza_type_id
group by pt.category order by revenue desc limit 3;
```



most_ordered_pizzas	revenue
Veggie	432.45
Classic	424.7
Supreme	419.65

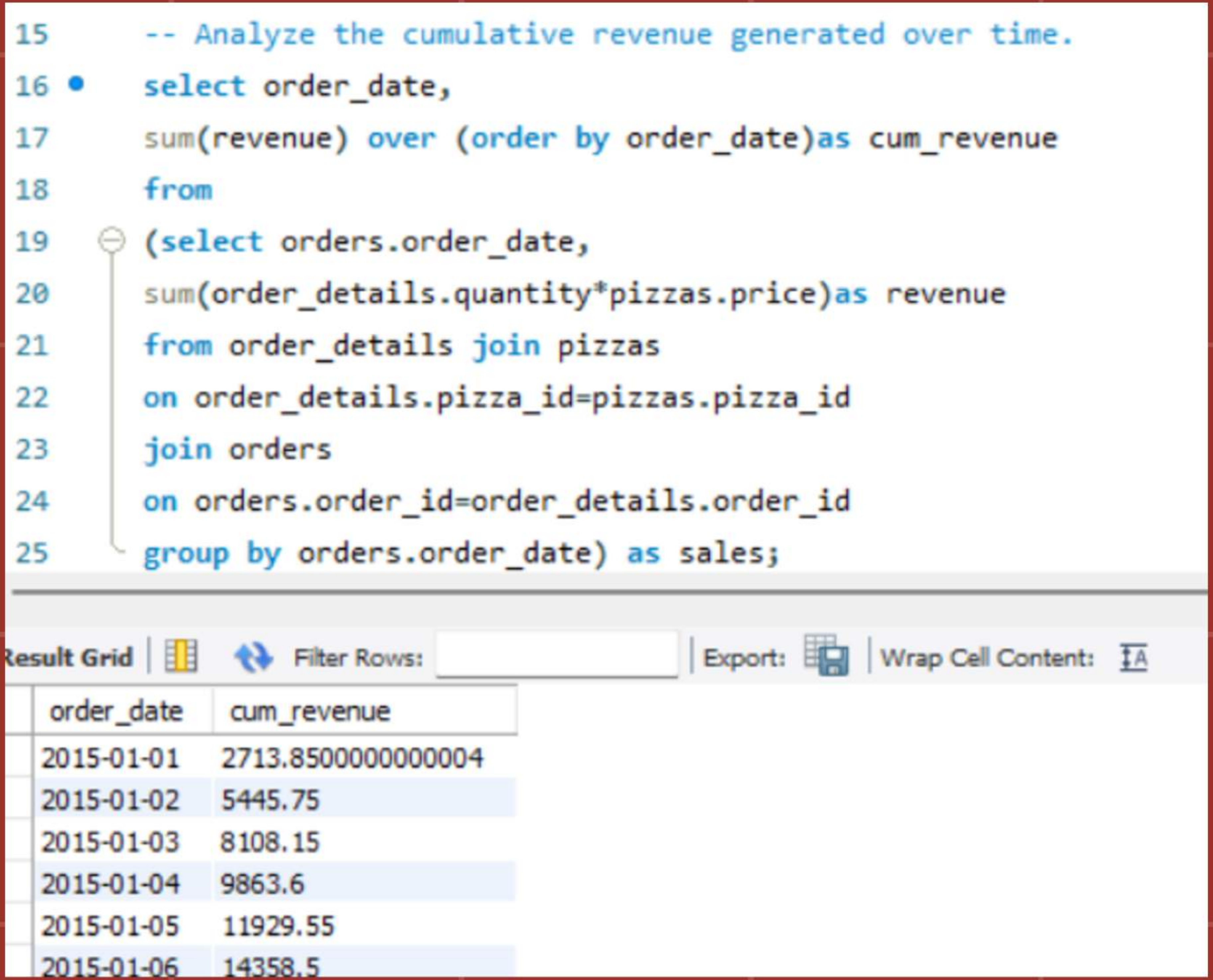
11.Calculate the percentage contribution of each pizza type to total revenue.



```sql
-- 1.Calculate the percentage contribution of each pizza type to the 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	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

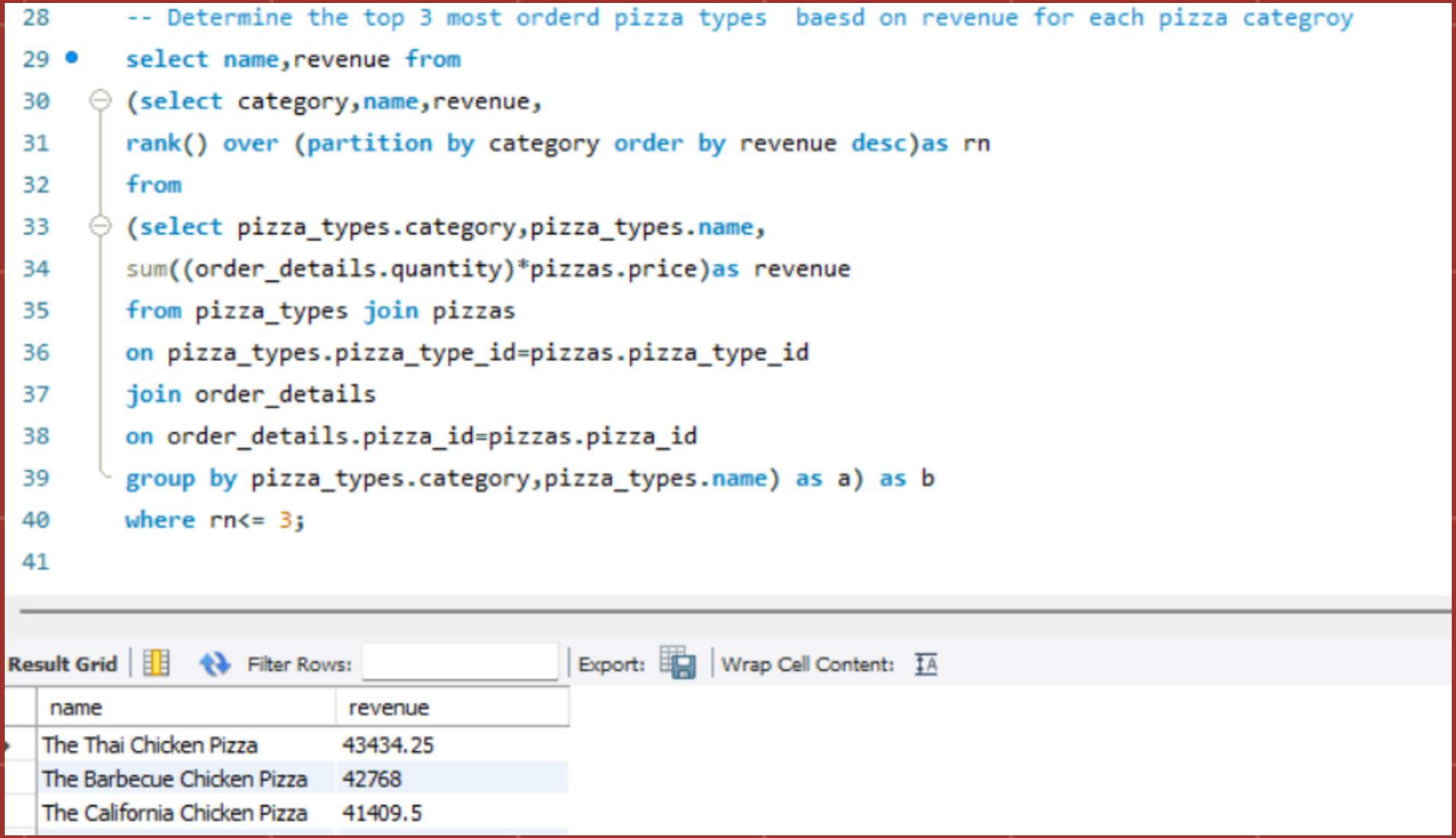
12.Analyze the cumulative revenue generated over time.



```
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue) over (order by order_date)as cum_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 orders.order_id=order_details.order_id
group by orders.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
2015-01-06	14358.5

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



```
-- Determine the top 3 most orderd pizza types  baesd on revenue for each pizza categroy
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((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;
```

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

Conclusion

In the Pizza Sales Analysis project, SQL queries were used to uncover key insights into sales patterns, customer preferences, and business performance at a pizza restaurant. The analysis covered everything from basic metrics like total orders and top-selling pizzas to more advanced evaluations of revenue contributions, category-specific trends, and order distributions. These insights are valuable for guiding strategic decisions aimed at optimizing operations and boosting profitability.

THANK YOU

www.linkedin.com/in/prashant-bhatt-a19754217