PIZZA SALES

DATA ANALYSIS USING SQL

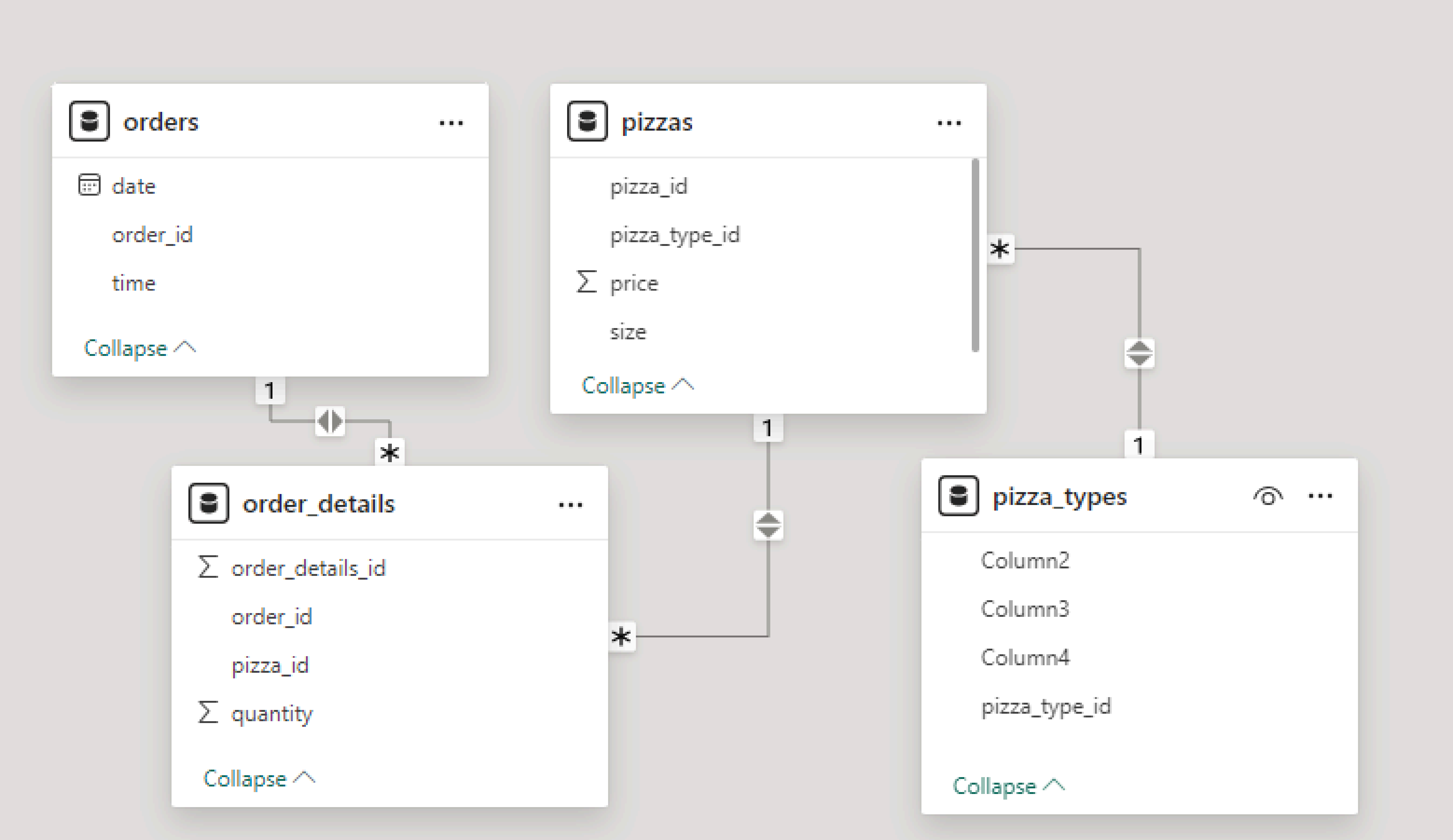
Presented by
Pranav

INTRODUCTION

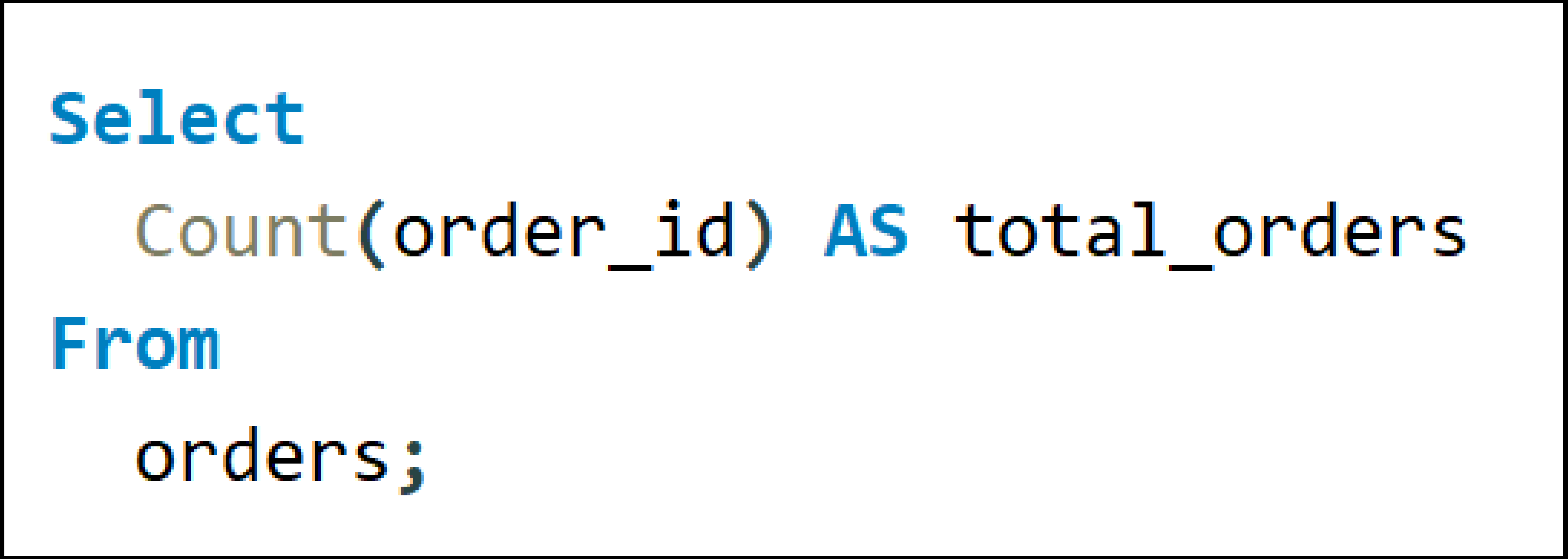
I’m S. Pranav Kumar. I recently completed a data analysis project focused on pizza sales. In this project, I utilized SQL to delve into the dataset, extract meaningful insights, and address specific queries related to the pizza sales data. I solved some questions on the basis of pizza sales.Through this project, I gained valuable experience in SQL query writing, data manipulation, and data analysis. I'm excited to share my findings and insights with you.

SCHEMA

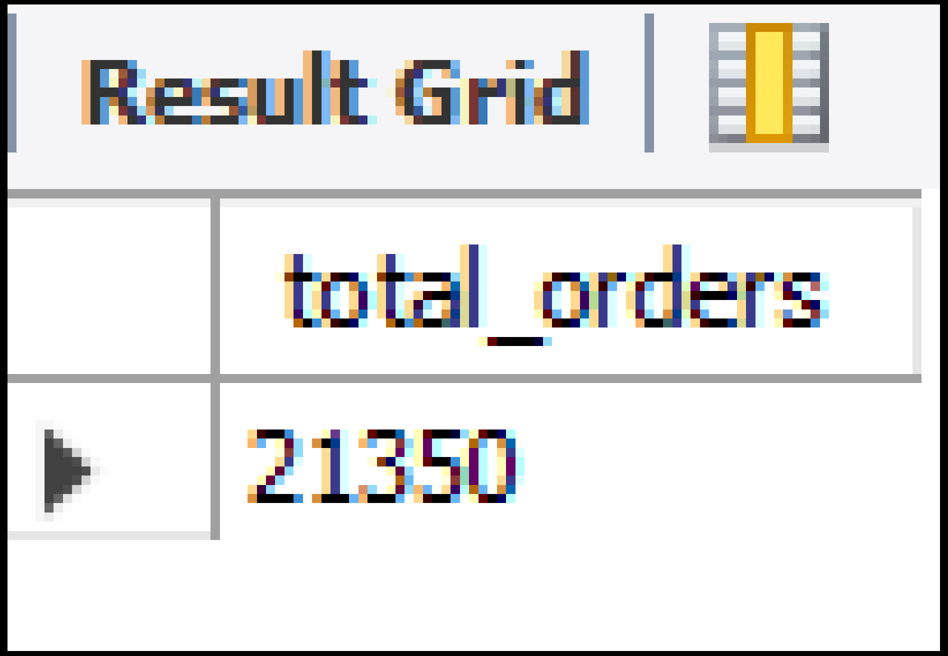
1.ORDERS
2.ORDER_DETAILS
3.PIZZA
4.PIZZA_TYPES



Q1.RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

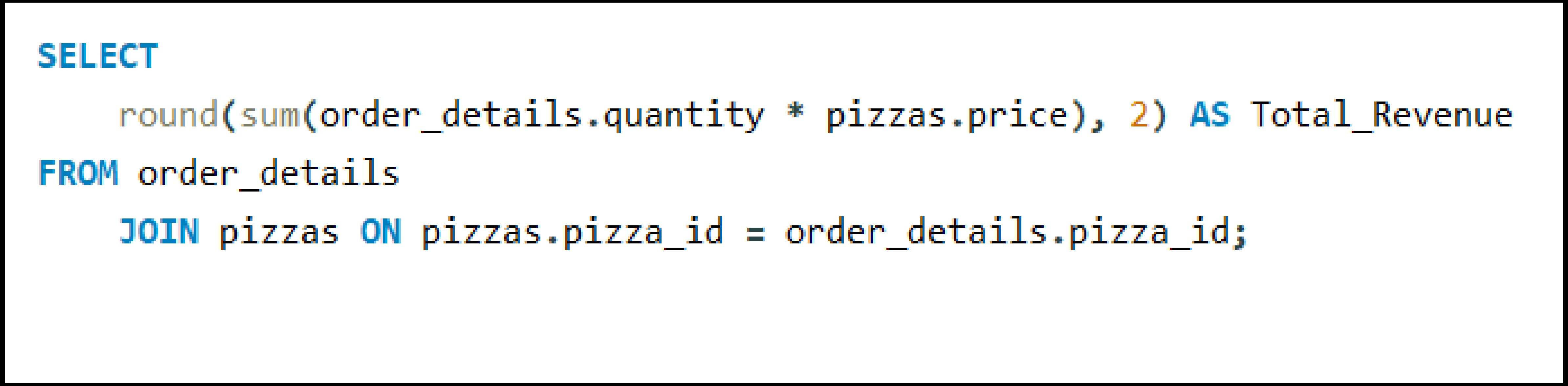


```
Select
   Count(order_id) AS total_orders
From
   orders;
```

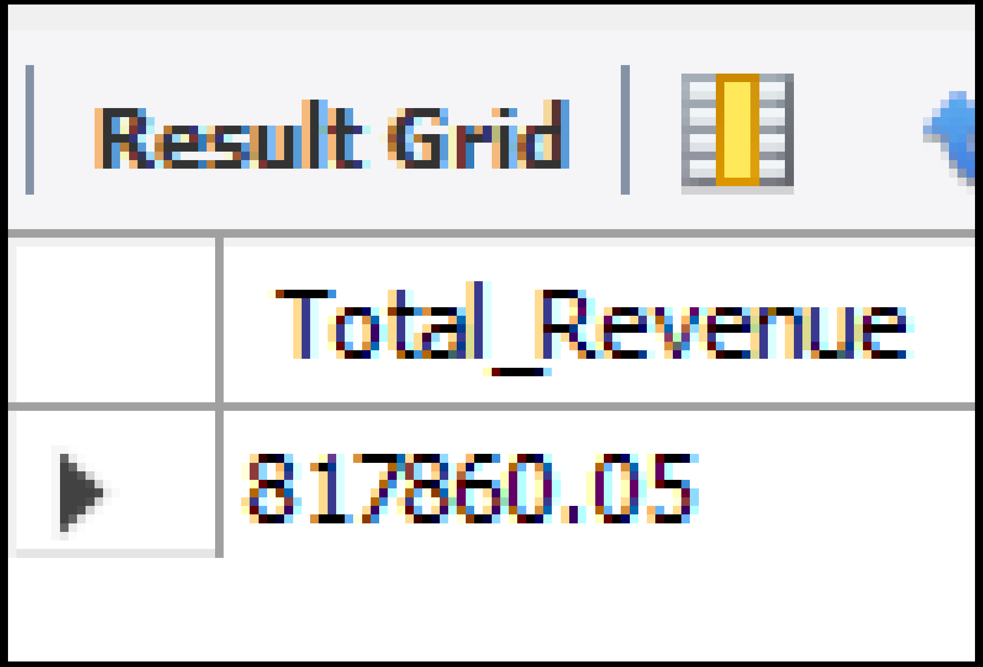


```
total_orders
21350
```

Q2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

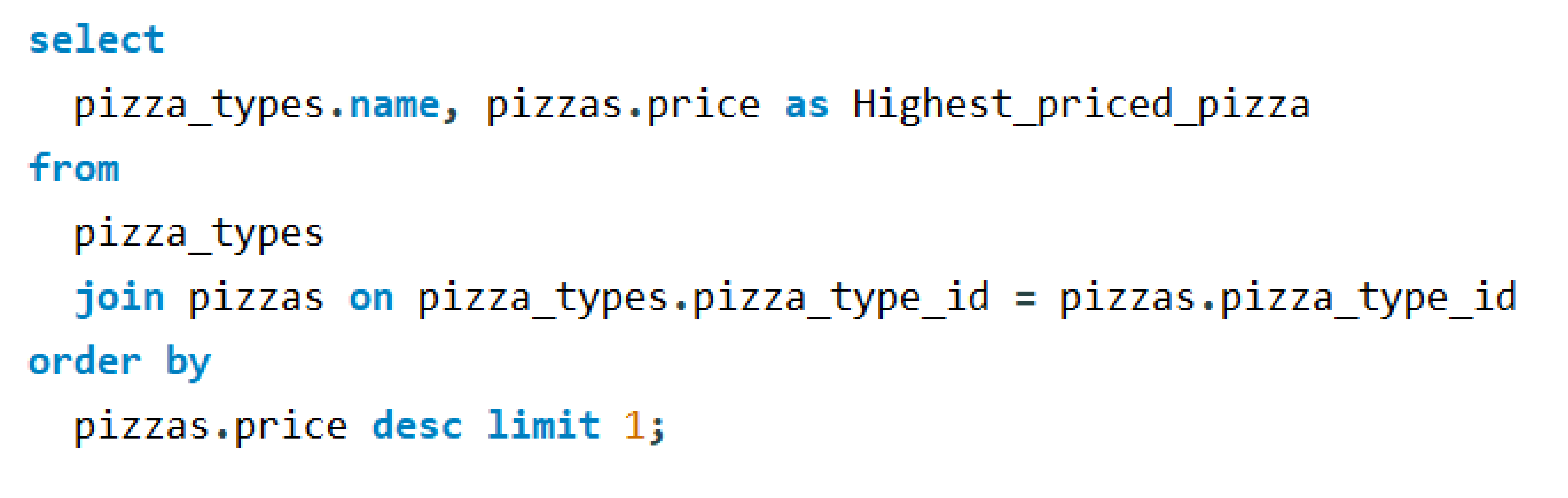


```sql
SELECT
    round(sum(order_details.quantity * pizzas.price), 2) AS Total_Revenue
FROM order_details
    JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

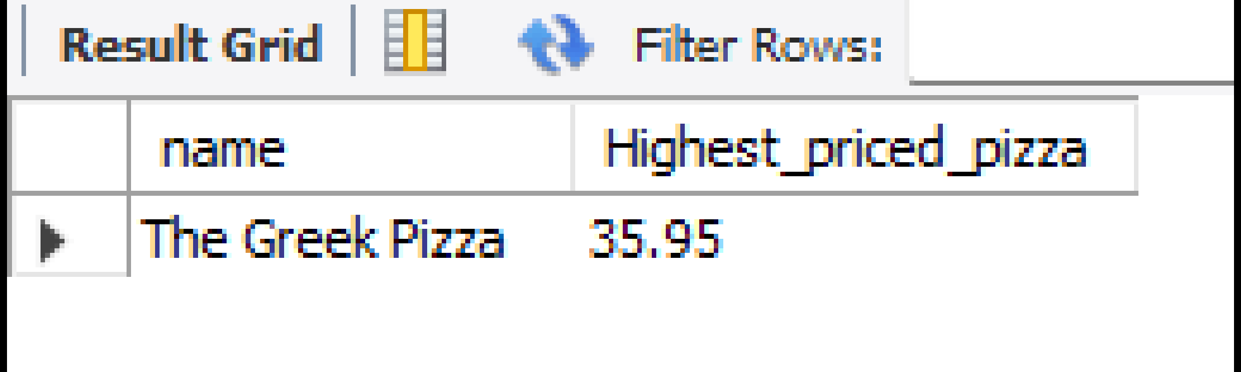


Total_Revenue
817860.05

Q3.IDENTIFY THE HIGHEST-PRICED PIZZA.

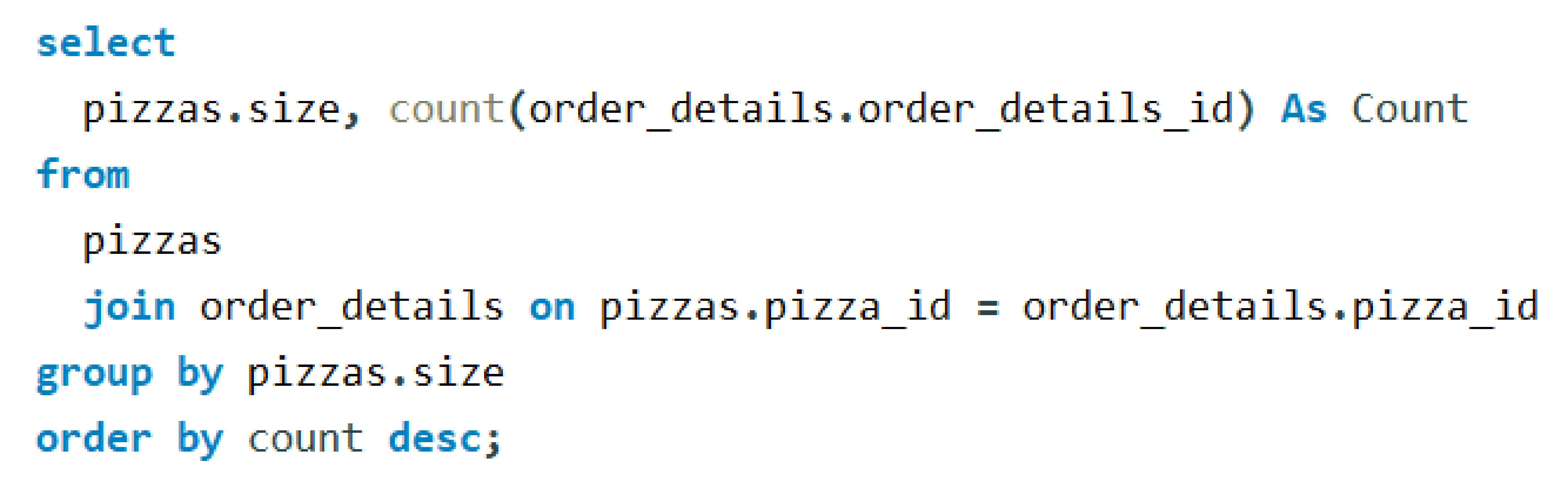


```sql
select
  pizza_types.name, pizzas.price as Highest_priced_pizza
from
  pizza_types
  join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by
  pizzas.price desc limit 1;
```

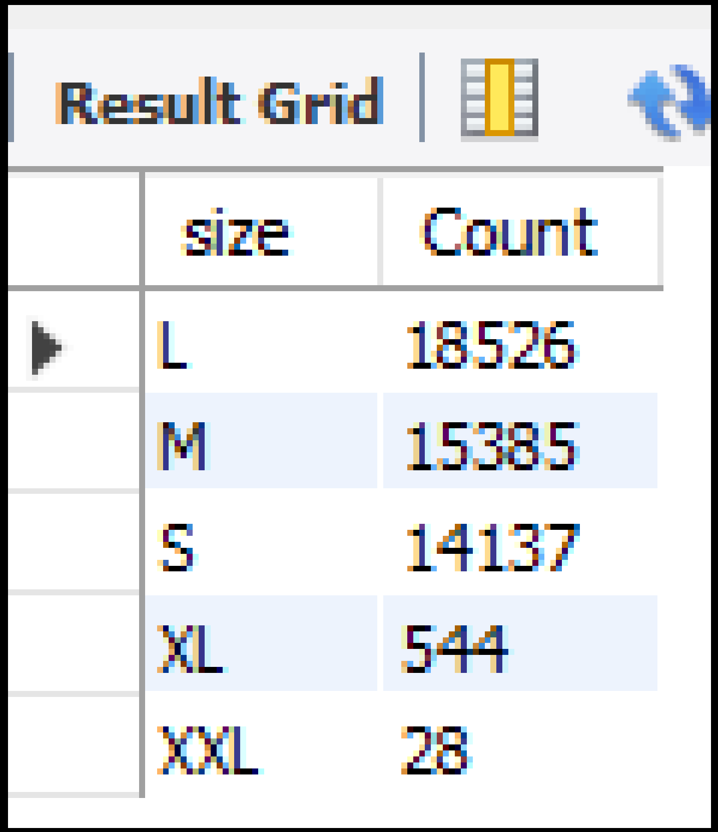


name	Highest_priced_pizza
The Greek Pizza	35.95

Q4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

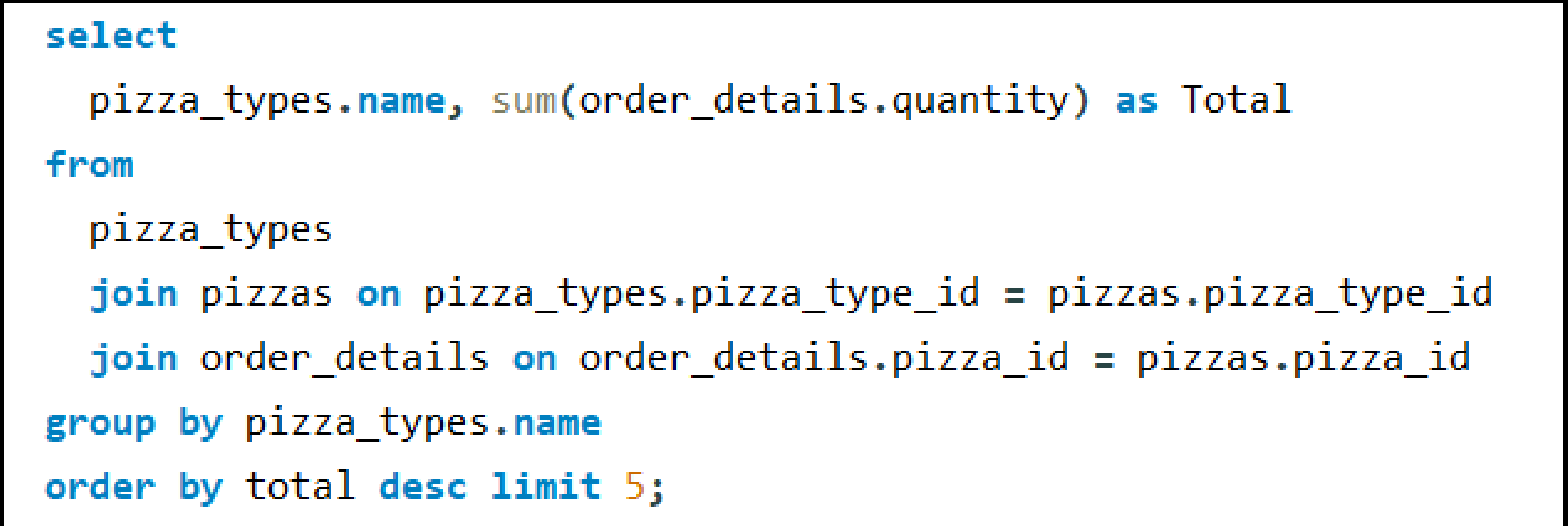


```
select
  pizzas.size, count(order_details.order_details_id) As Count
from
  pizzas
  join order_details on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by count desc;
```

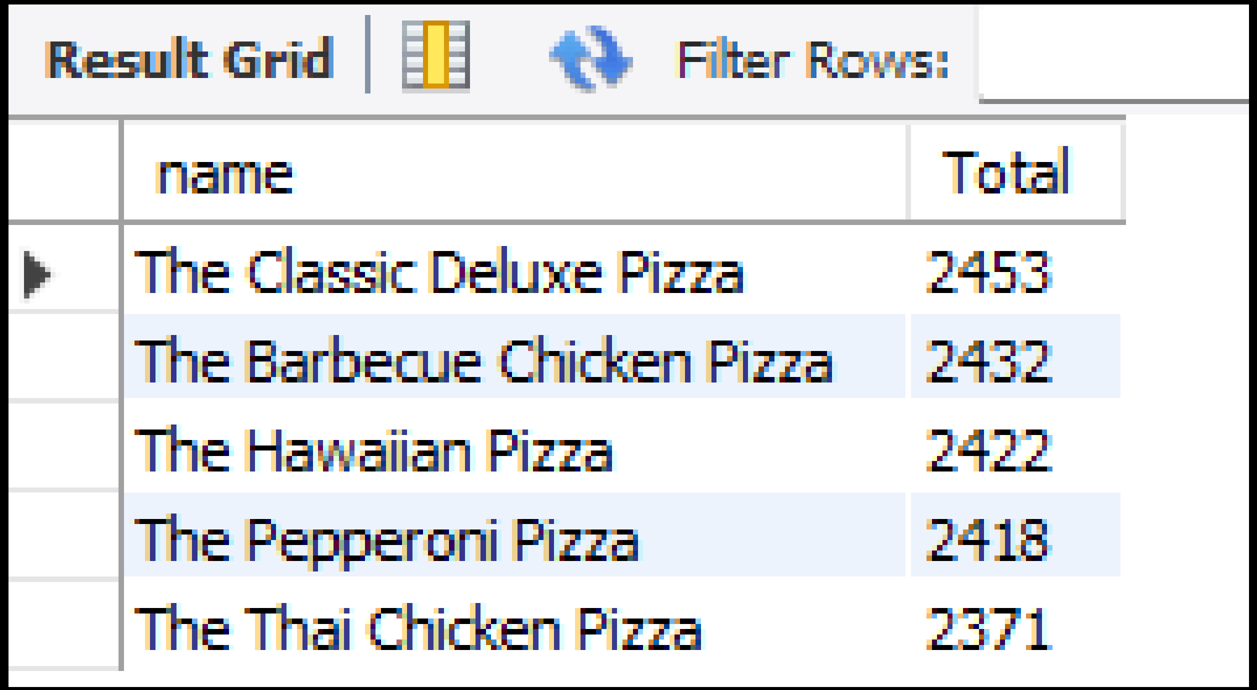


size	Count
L	18526
M	15385
S	14137
XL	544
XXL	28

Q5.LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

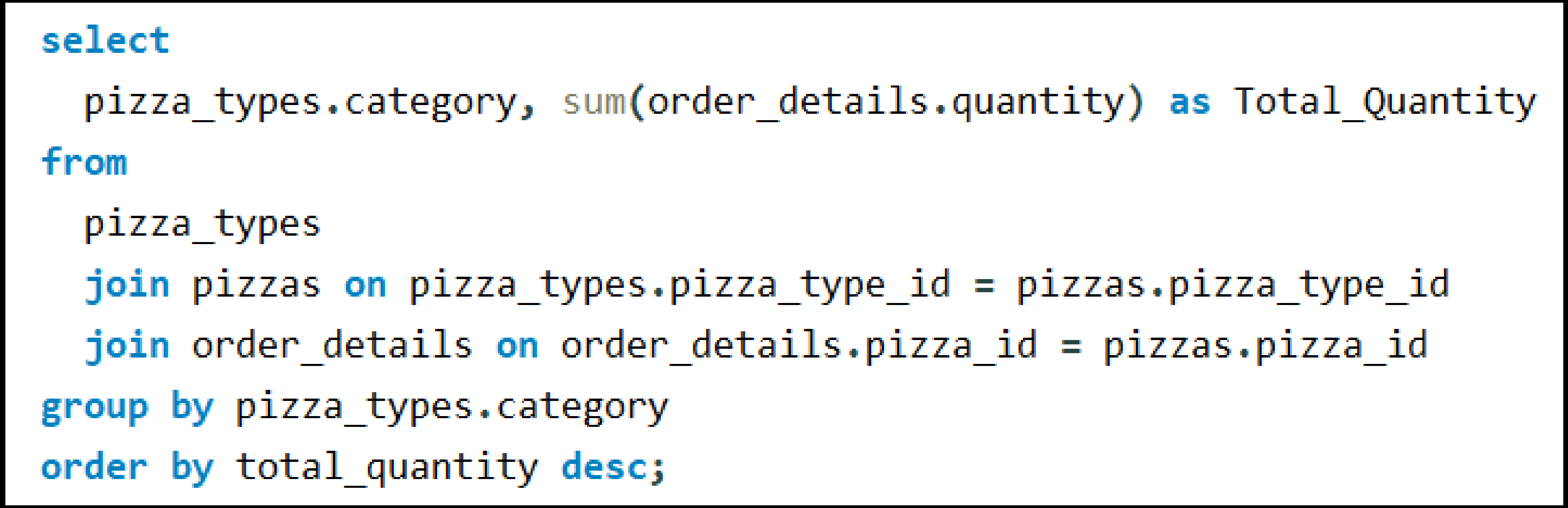


```
select
  pizza_types.name, sum(order_details.quantity) as Total
from
  pizza_types
  join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
  join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by total desc limit 5;
```

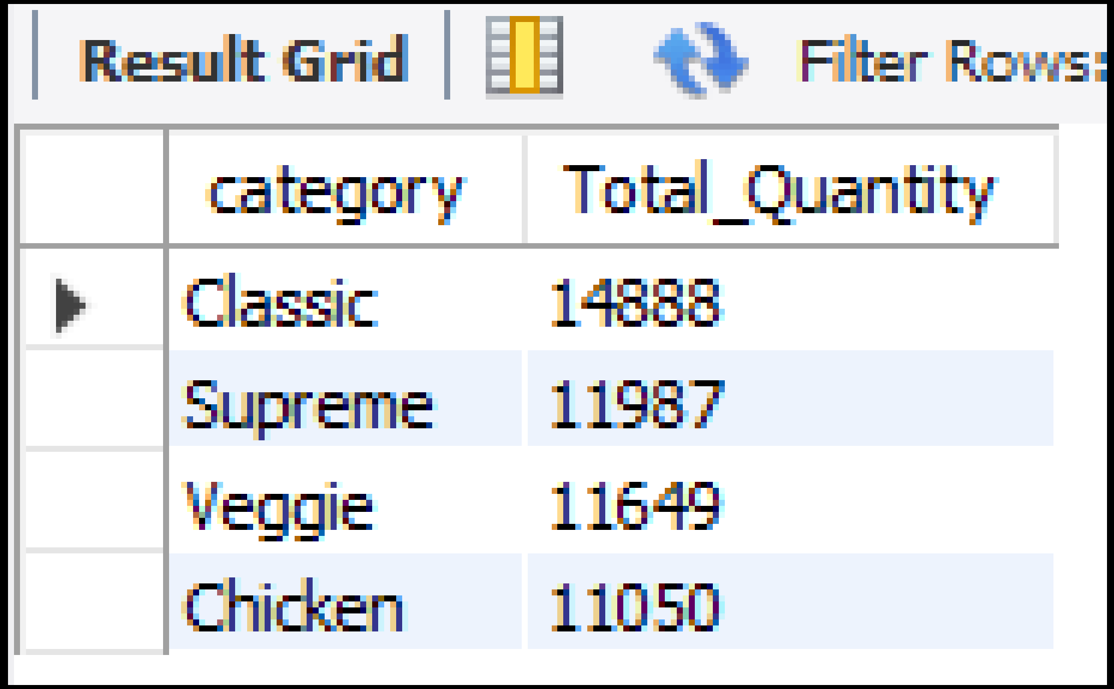


name	Total
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q6.JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

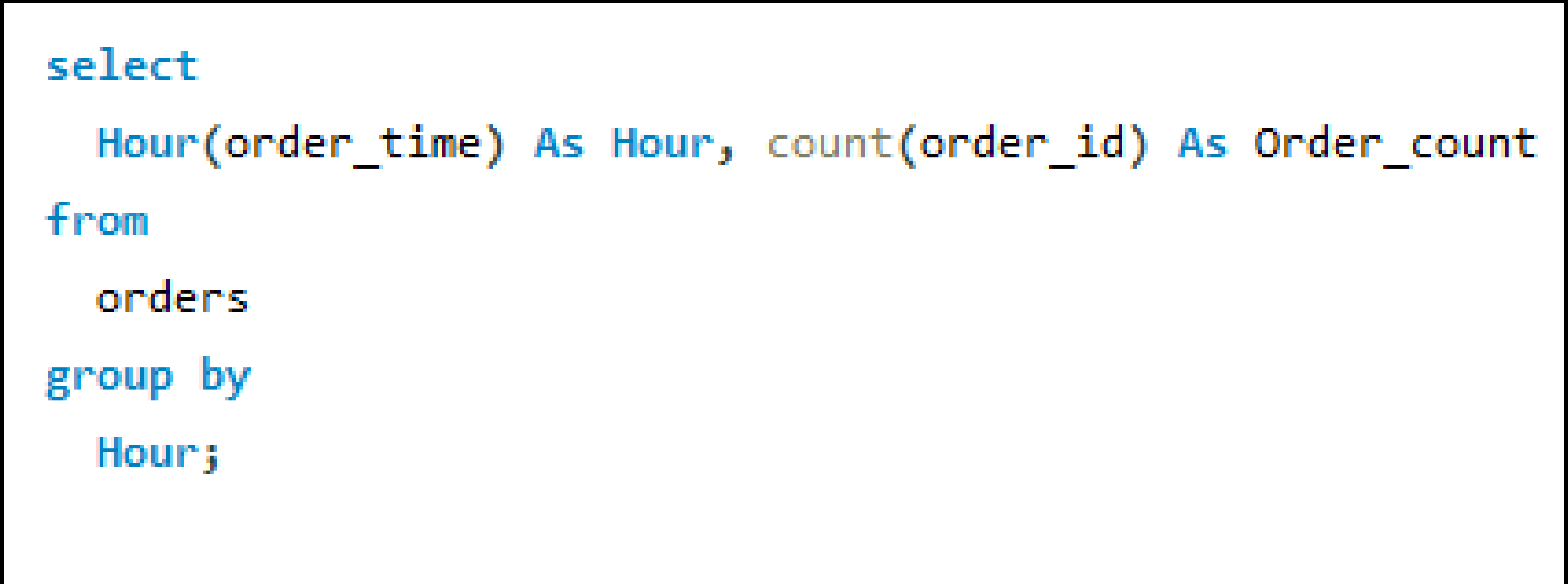


```
select
  pizza_types.category, sum(order_details.quantity) as Total_Quantity
from
  pizza_types
  join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
  join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by total_quantity desc;
```

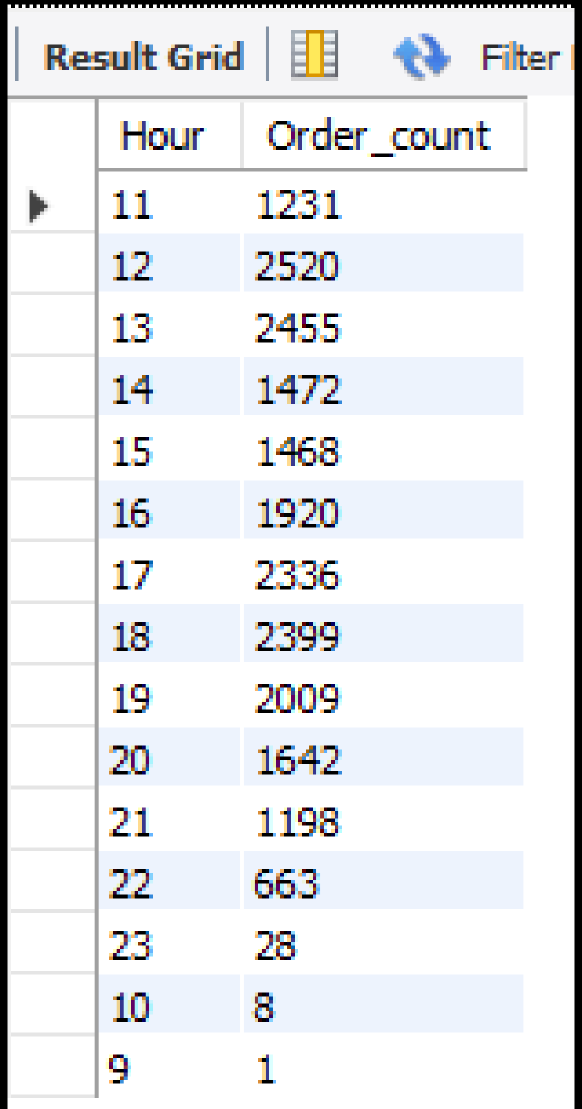


category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

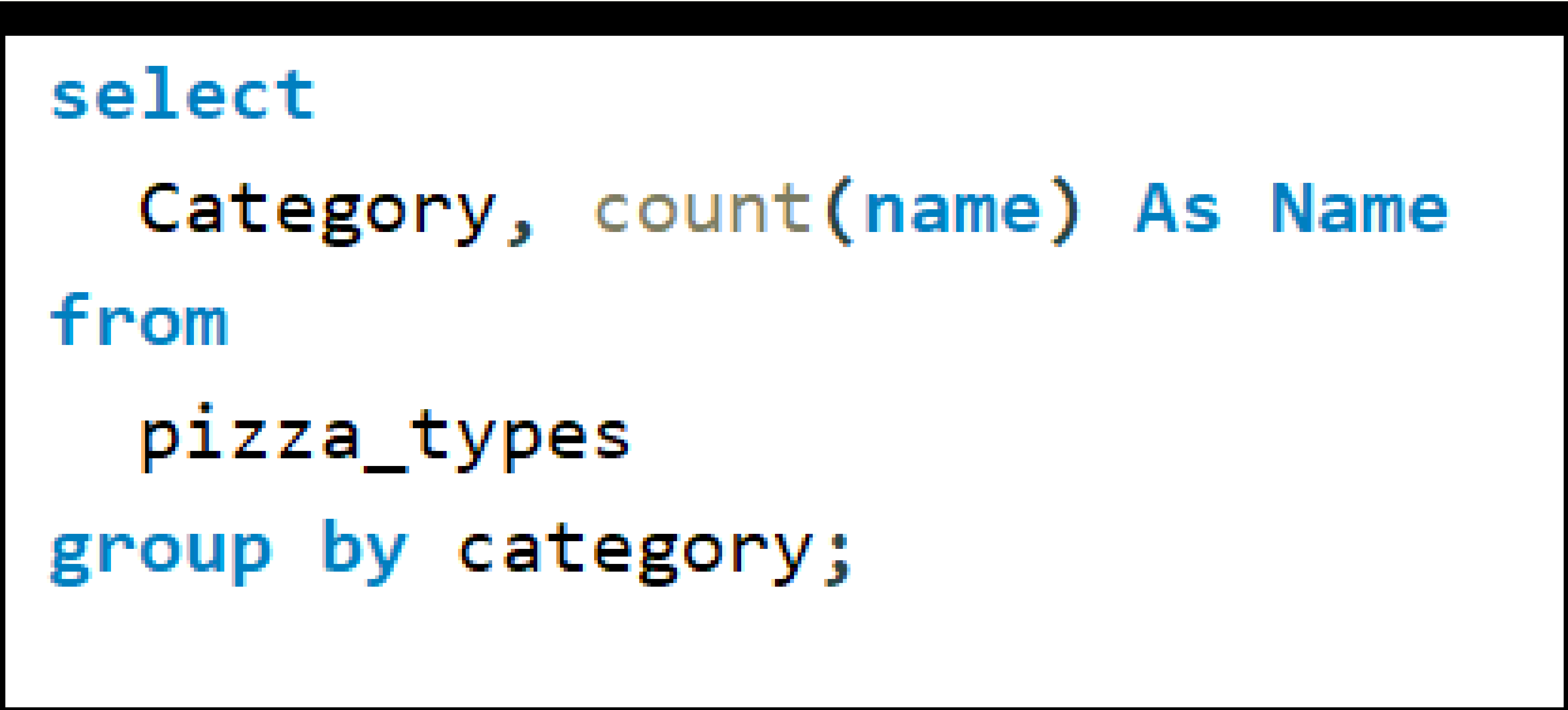


```
select
  Hour(order_time) As Hour, count(order_id) As Order_count
from
  orders
group by
  Hour;
```

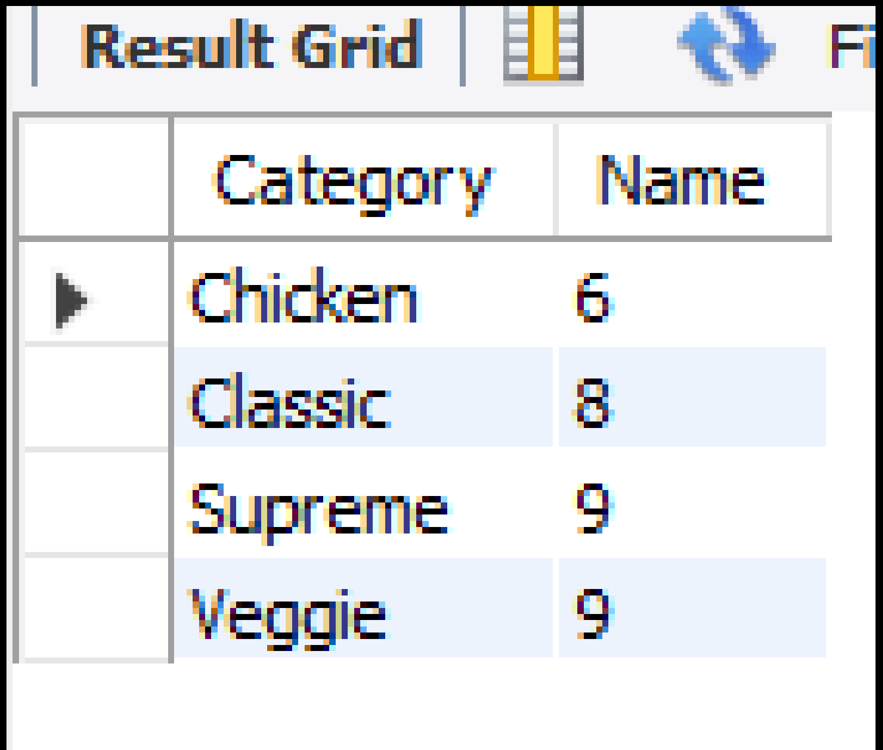


Hour	Order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Q8.JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

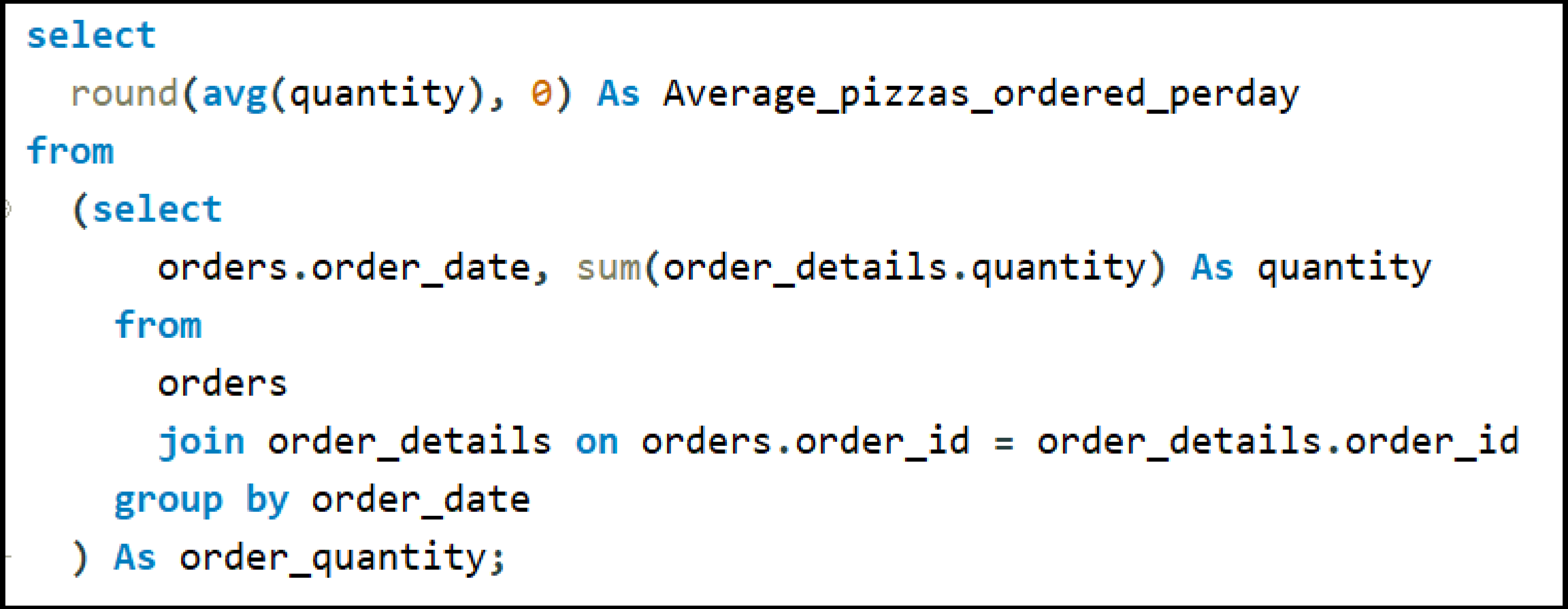


```
select
  Category, count(name) As Name
from
  pizza_types
group by category;
```

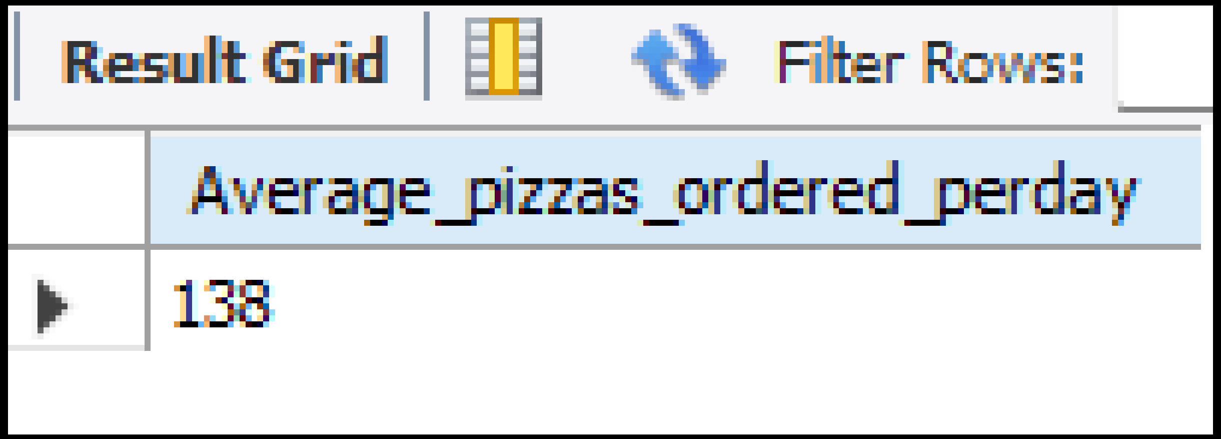


Category	Name
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9.GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

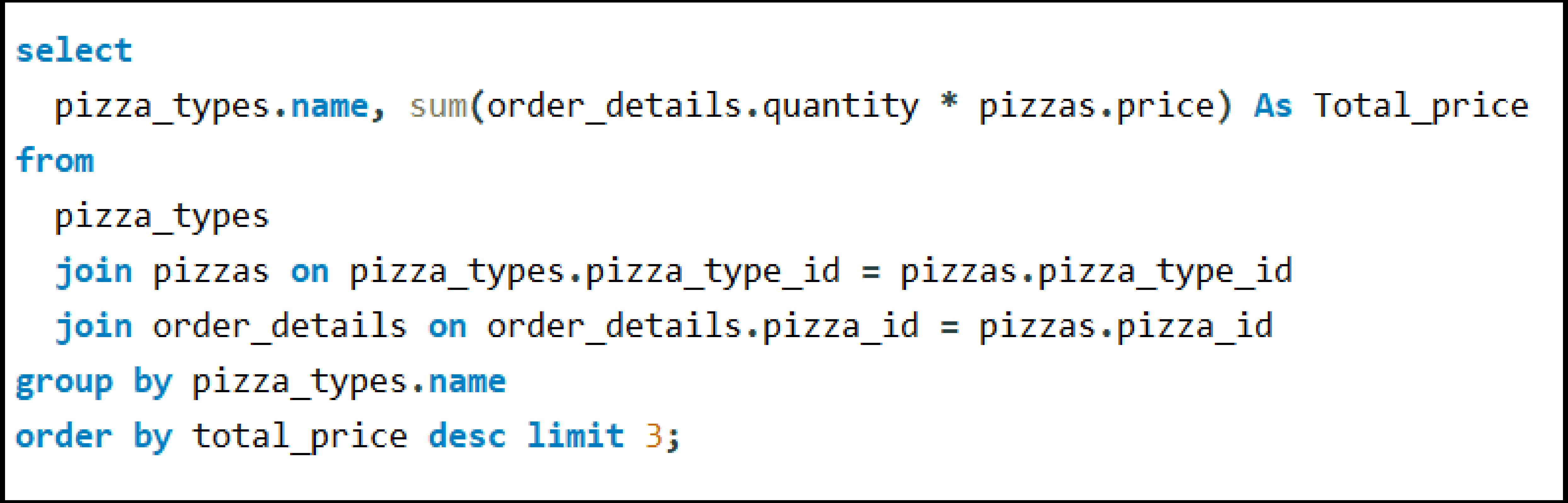


```
select
  round(avg(quantity), 0) As Average_pizzas_ordered_perday
from
  (select
      orders.order_date, sum(order_details.quantity) As quantity
    from
      orders
      join order_details on orders.order_id = order_details.order_id
    group by order_date
  ) As order_quantity;
```

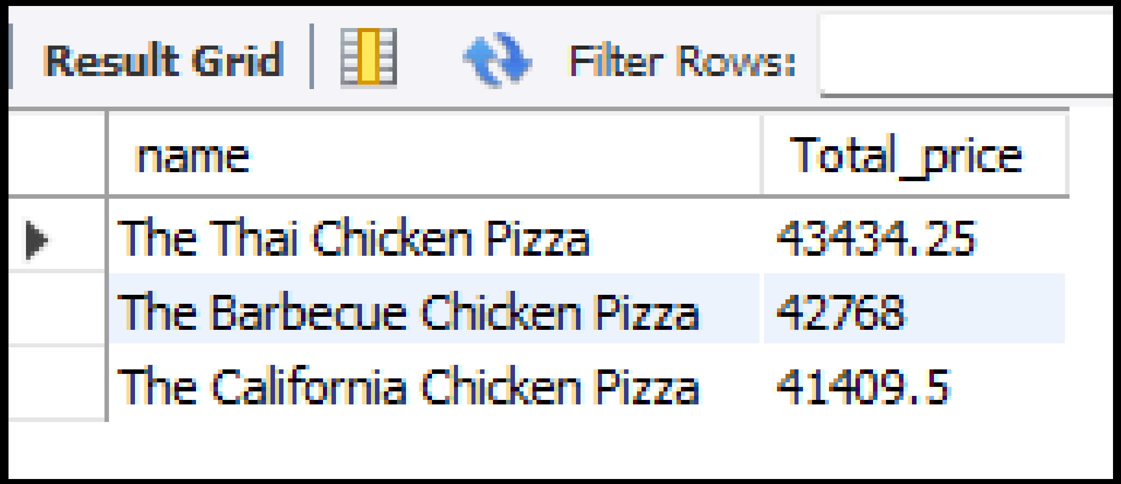


```
Average_pizzas_ordered_perday
138
```

Q10.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

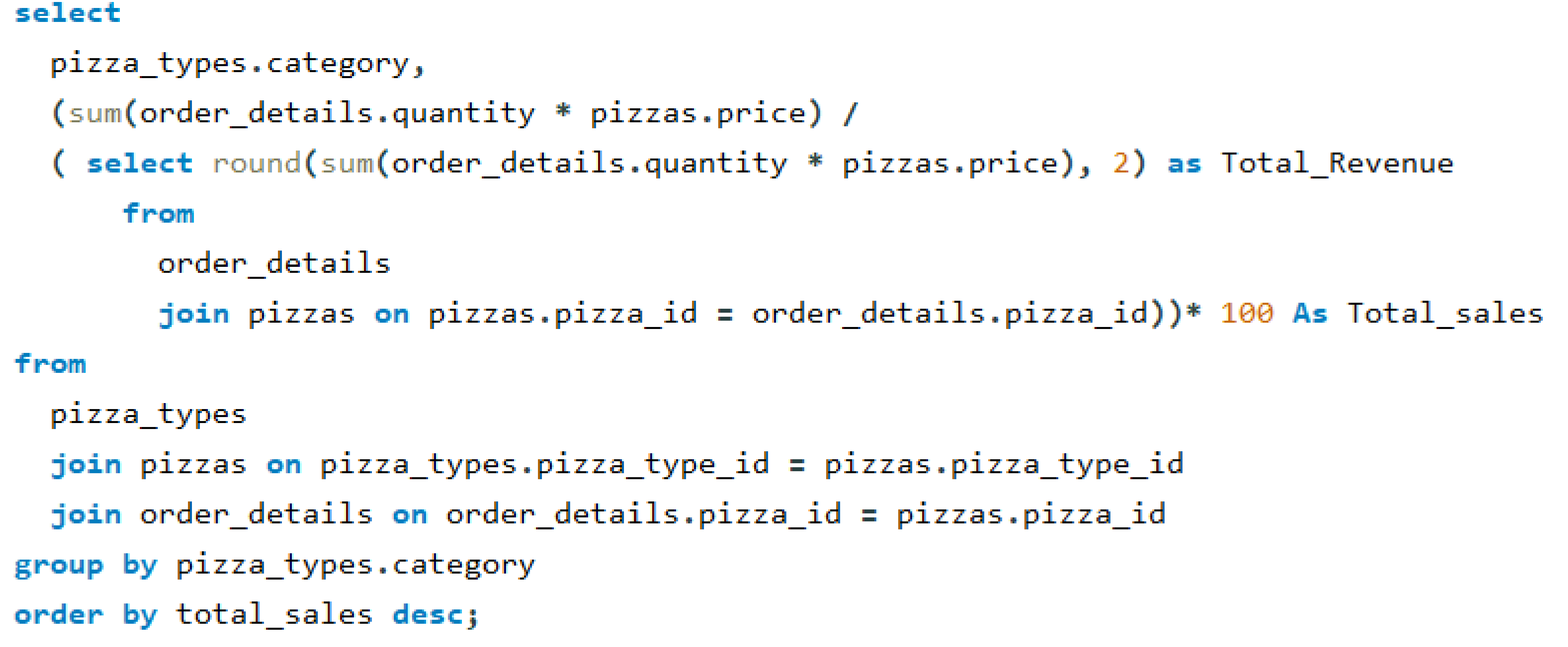


```
select
  pizza_types.name, sum(order_details.quantity * pizzas.price) As Total_price
from
  pizza_types
  join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
  join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by total_price desc limit 3;
```

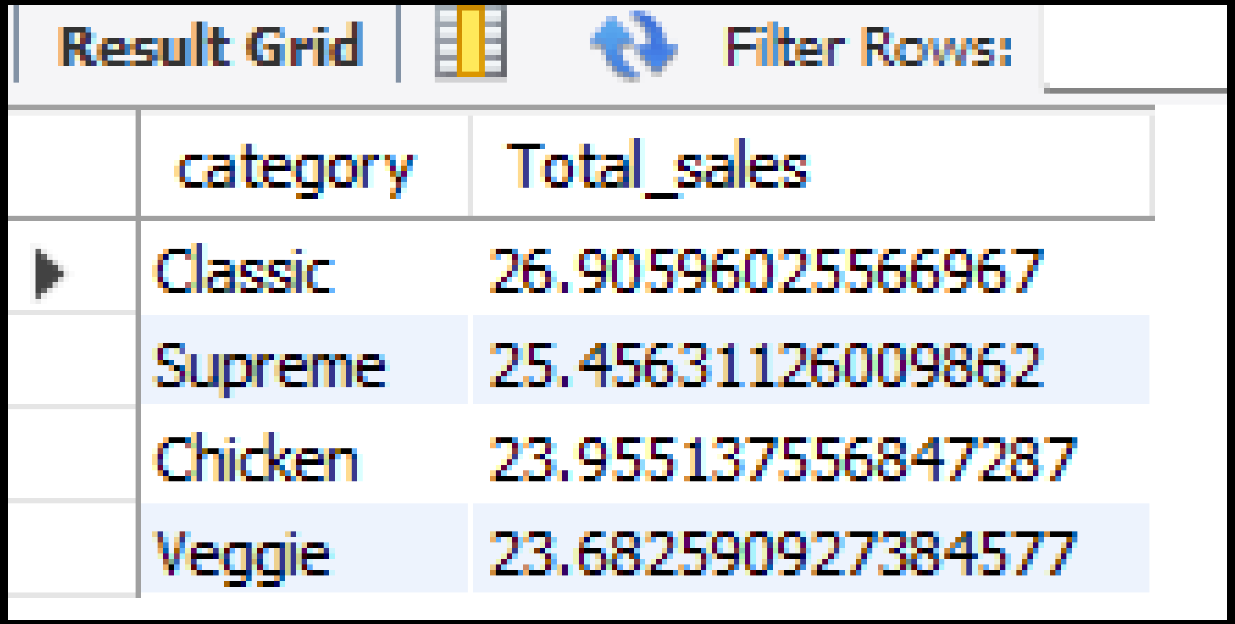


name	Total_price
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Q11.CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

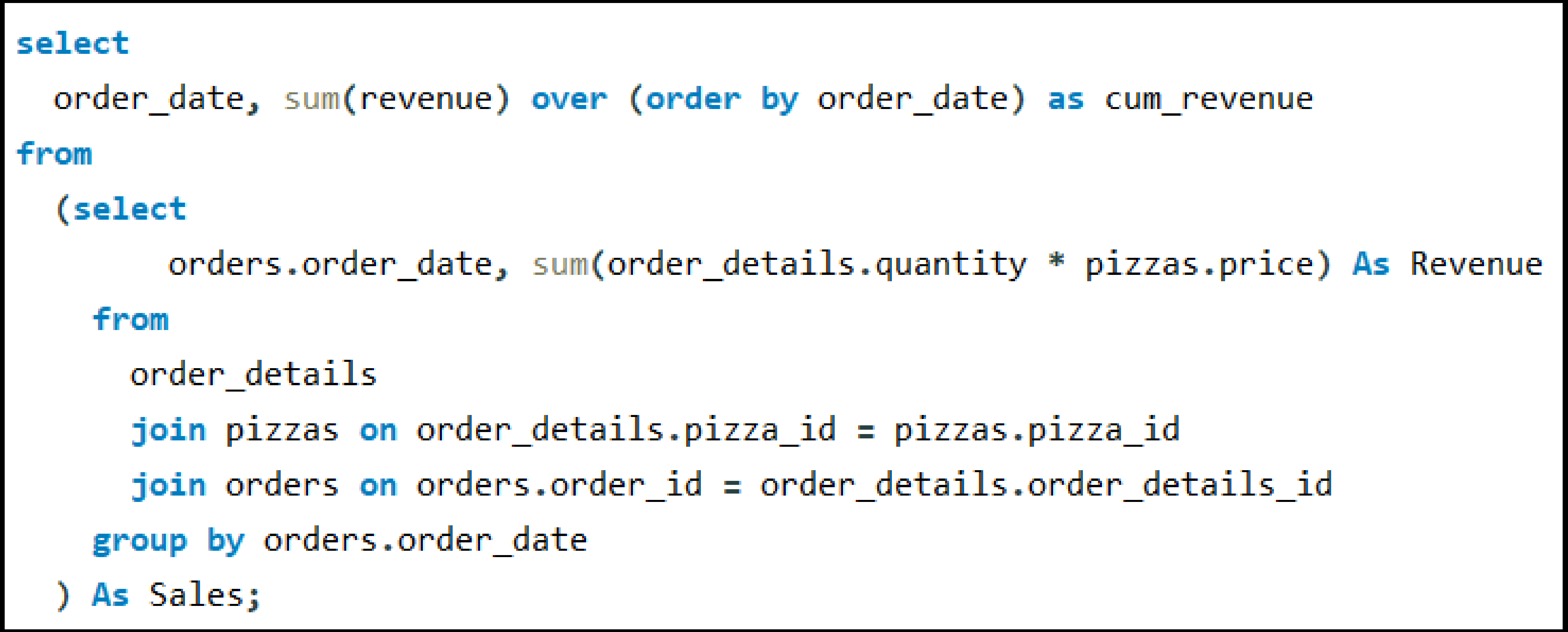


```sql
select
  pizza_types.category,
  (sum(order_details.quantity * pizzas.price) /
  ( select round(sum(order_details.quantity * pizzas.price), 2) as Total_Revenue
      from
        order_details
        join pizzas on pizzas.pizza_id = order_details.pizza_id))* 100 As Total_sales
from
  pizza_types
  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 total_sales desc;
```

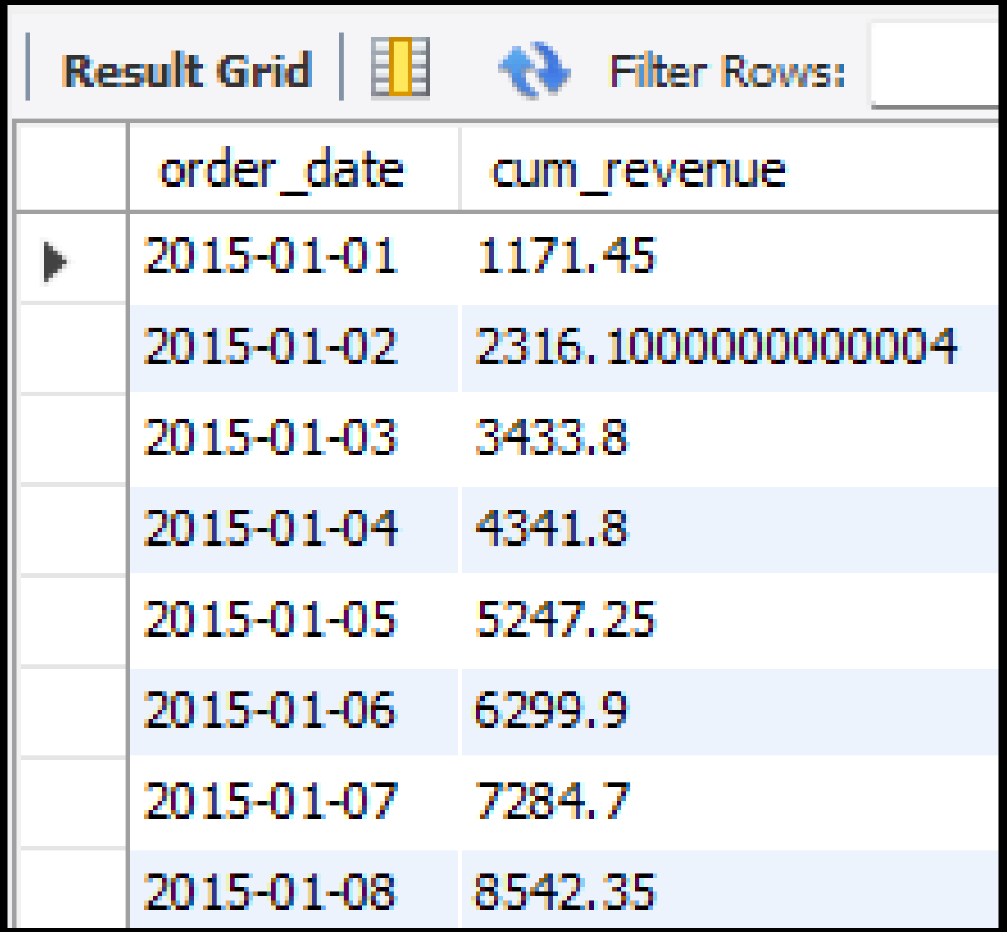


category	Total_sales
Classic	26.9059602556967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

Q12.ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



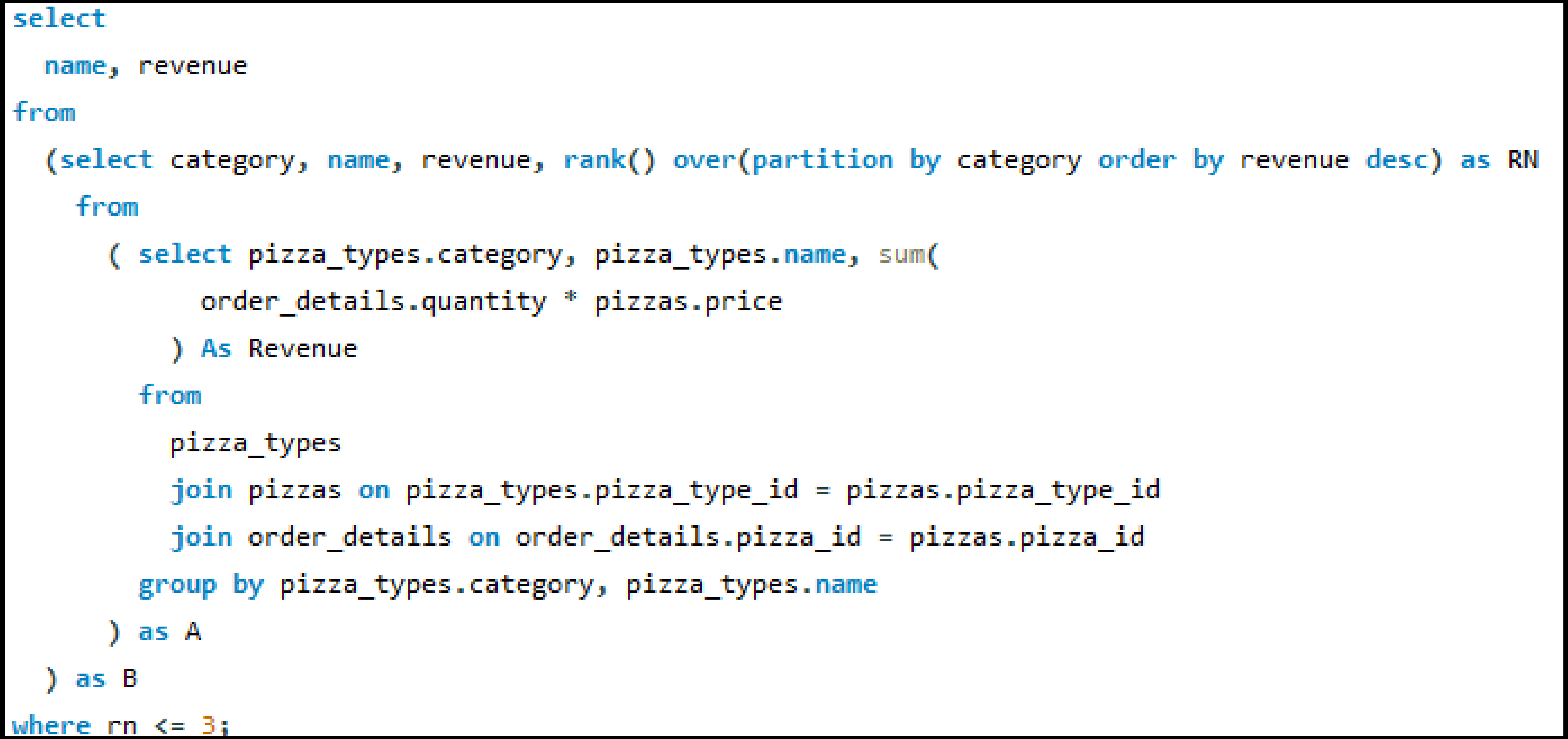
```
select
 order_date, sum(revenue) over (order by order_date) as cum_revenue
from
 (select
       orders.order_date, sum(order_details.quantity * pizzas.price) As Revenue
   from
     order_details
     join pizzas on order_details.pizza_id = pizzas.pizza_id
     join orders on orders.order_id = order_details.order_details_id
   group by orders.order_date
 ) As Sales;
```



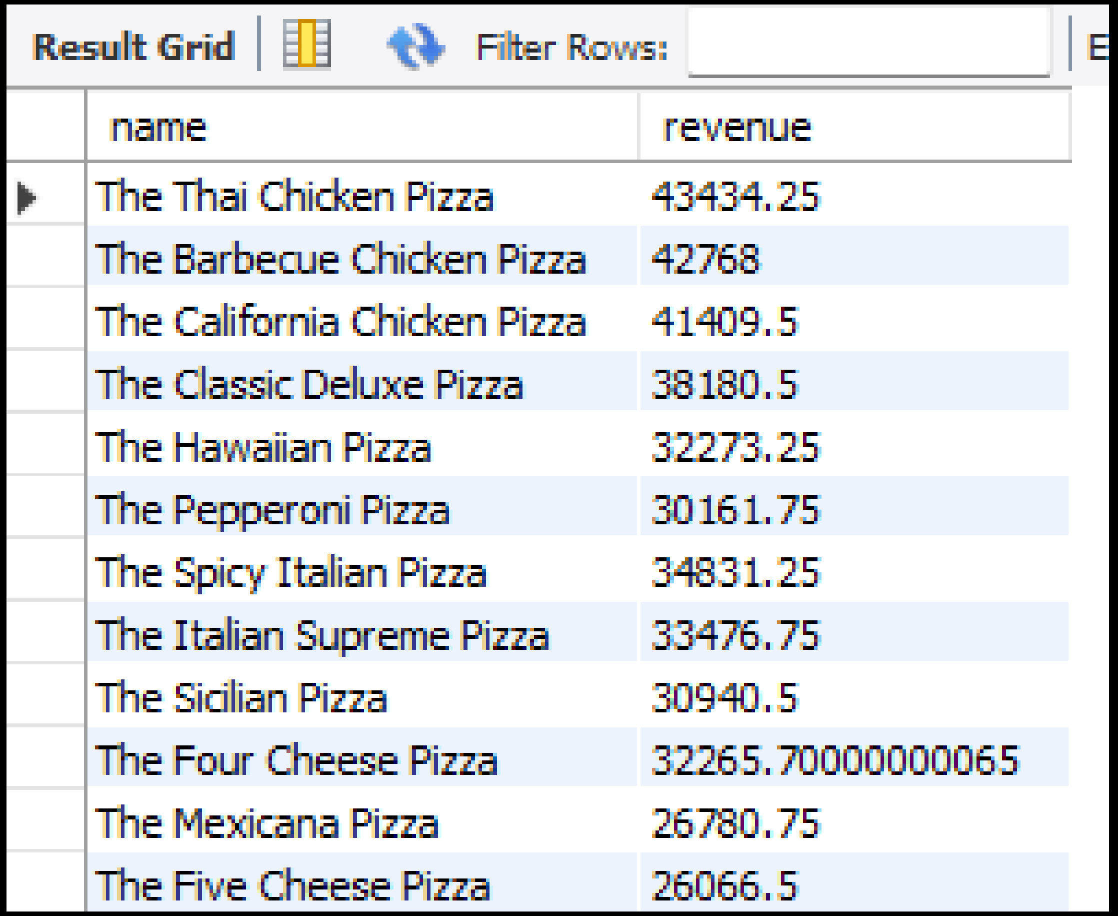
order_date	cum_revenue
2015-01-01	1171.45
2015-01-02	2316.1000000000004
2015-01-03	3433.8
2015-01-04	4341.8
2015-01-05	5247.25
2015-01-06	6299.9
2015-01-07	7284.7
2015-01-08	8542.35

AND MORE

Q13.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

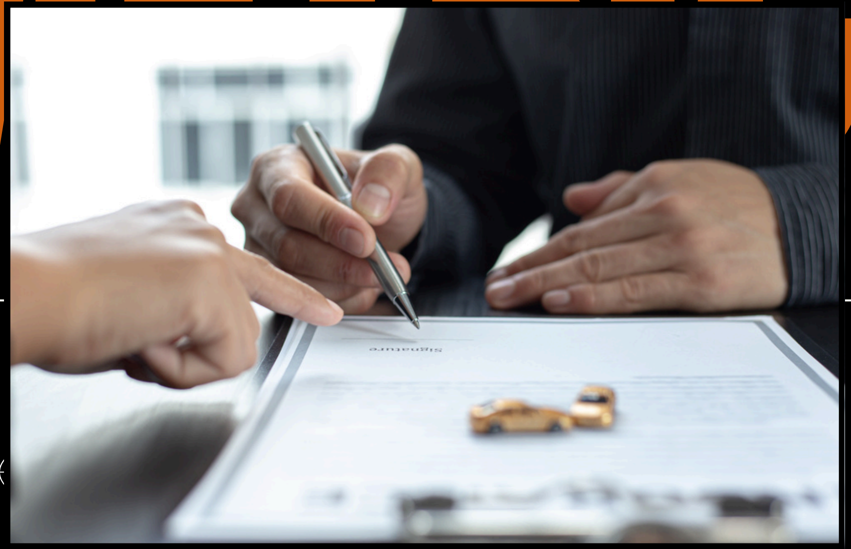


```
select
  name, revenue
from
  (select category, name, revenue, rank() over(partition by category order by revenue desc) as RN
    from
      ( select pizza_types.category, pizza_types.name, sum(
            order_details.quantity * pizzas.price
          ) As Revenue
        from
          pizza_types
          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
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

KEY METRICS



By solving the Queries to the questions I came to know that the points are:-

1.Pizza Sales are pretty high in Classic Category comparatively.
2.People are mostly choosing to order Single Quantity.
3.Mostly customers are ordering at 12:00PM.
4.”L” sized pizza is making best sales.

i suggests the company to maintain the application steady in the rush hours and to increase the offers to the “L” sized pizzas and to make the single quantity ordering customers to order more company to give some extra cashbacks,offers.

+91 7997110065

sappapranav142@gmail.com

THANK YOU!

Thank you for your attention to my sales report presentation.