MENU ABOUT US **HOME**

PIZZA HUT

PIZZA HUT

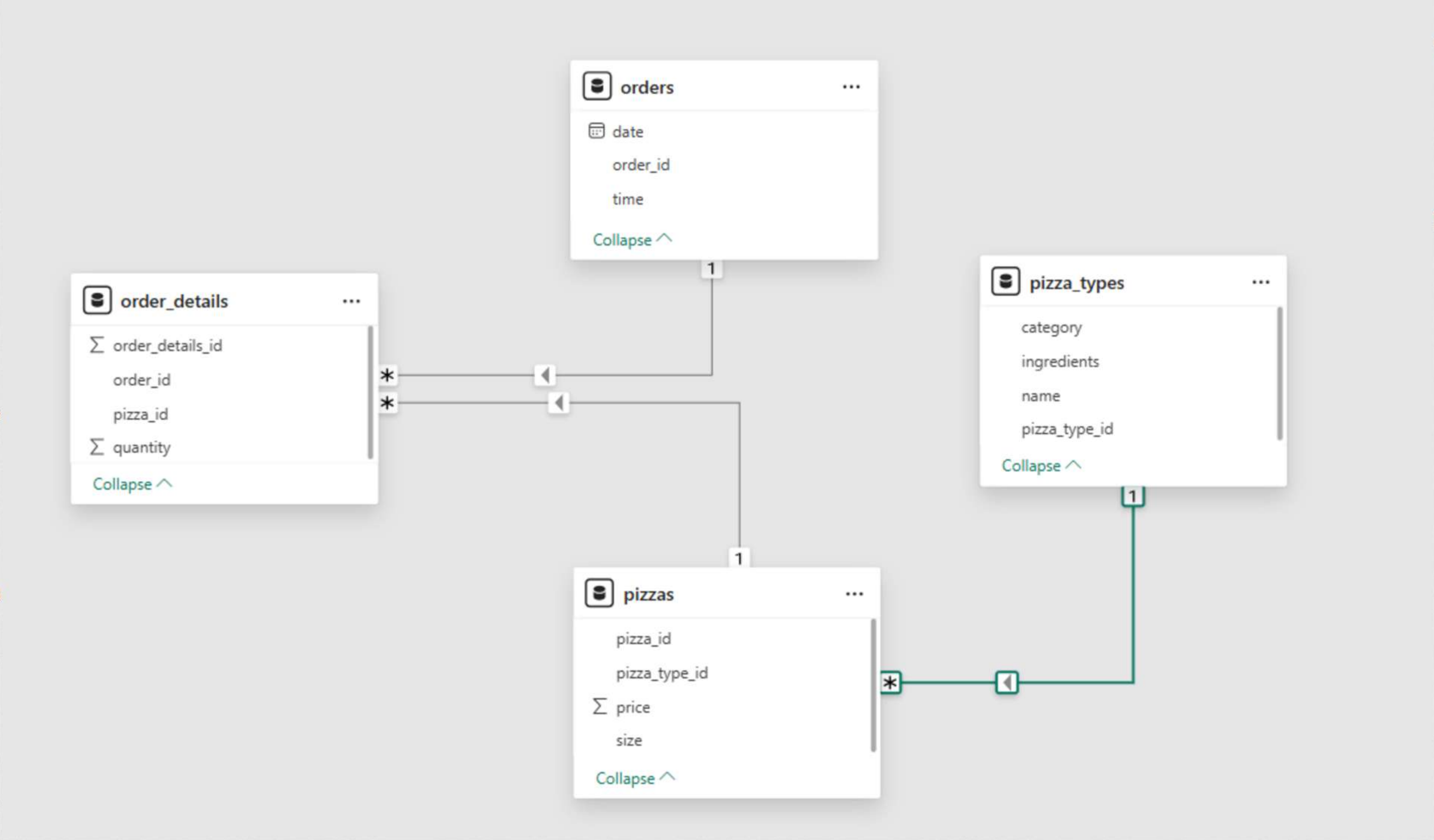
NEXT >

AFROSE BASHA

Hello, my name is Afrose Basha, and I am an aspiring data scientist. In this project, I utilized SQL to address various problem statements related to the data, generating reports that provide valuable insights for stakeholders.

< BACK NEXT >

DATA BASE SCHEMA



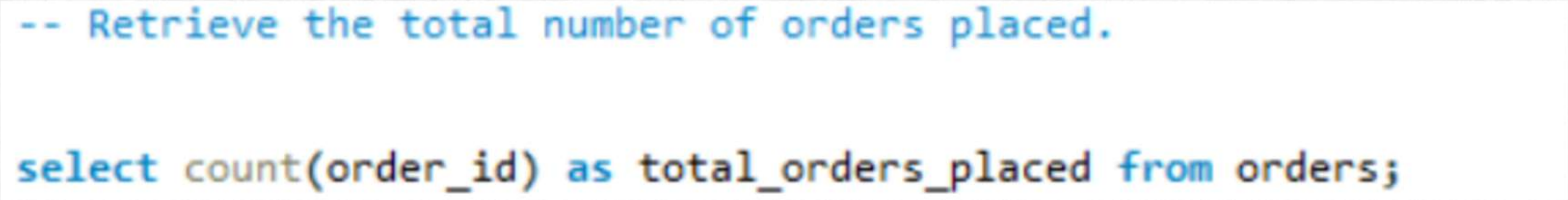
orders
- date
- order_id
- time

order_details
- order_details_id
- order_id
- pizza_id
- quantity

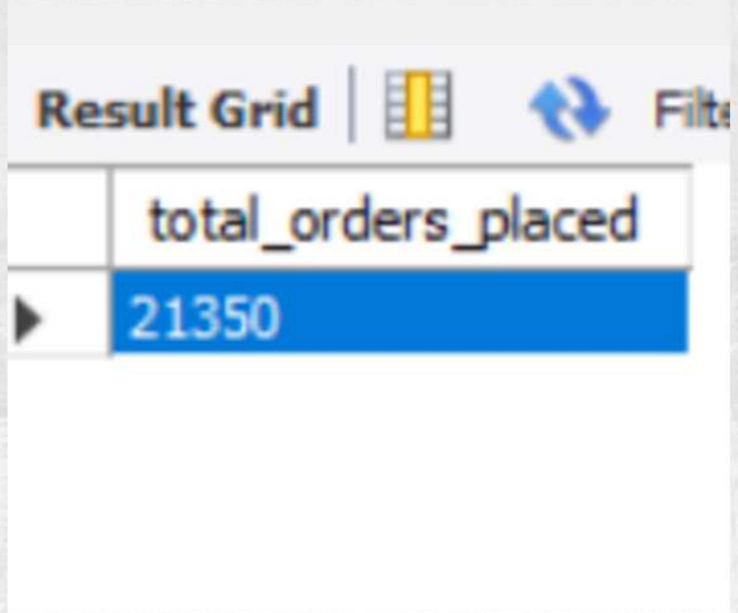
pizza_types
- category
- ingredients
- name
- pizza_type_id

pizzas
- pizza_id
- pizza_type_id
- price
- size

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



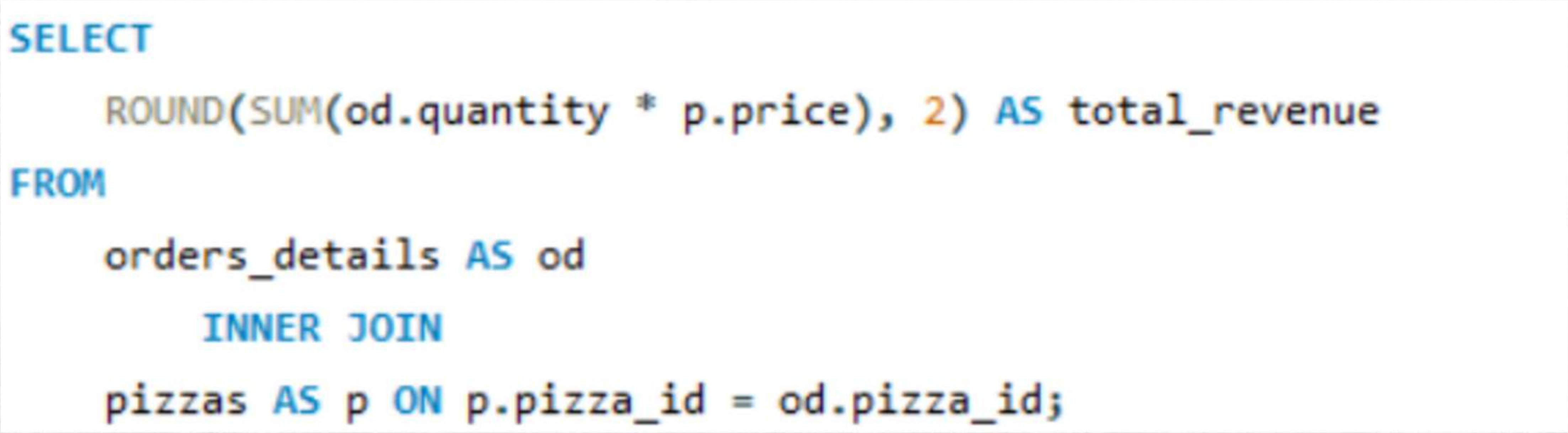
```
-- Retrieve the total number of orders placed.


select count(order_id) as total_orders_placed from orders;
```



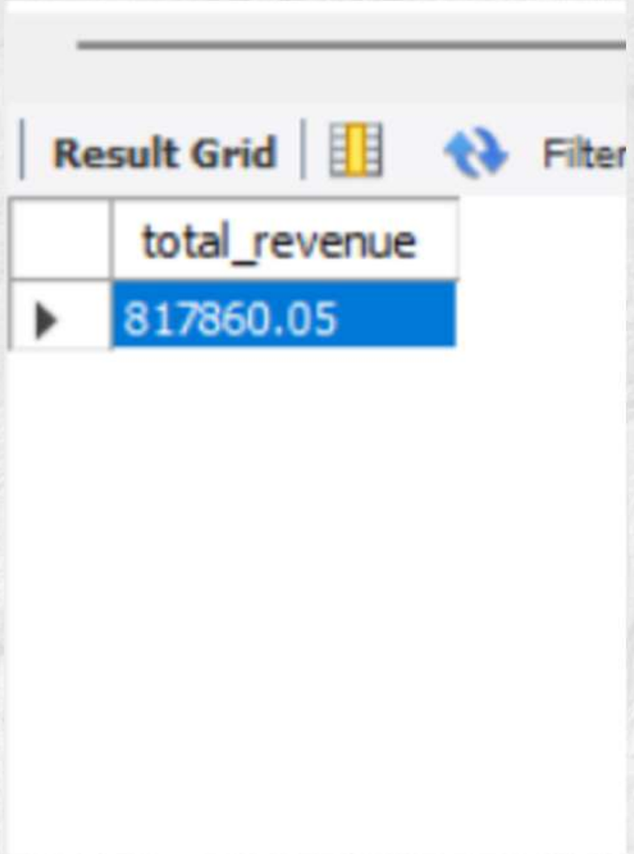
```
total_orders_placed
21350
```

NEXT >

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



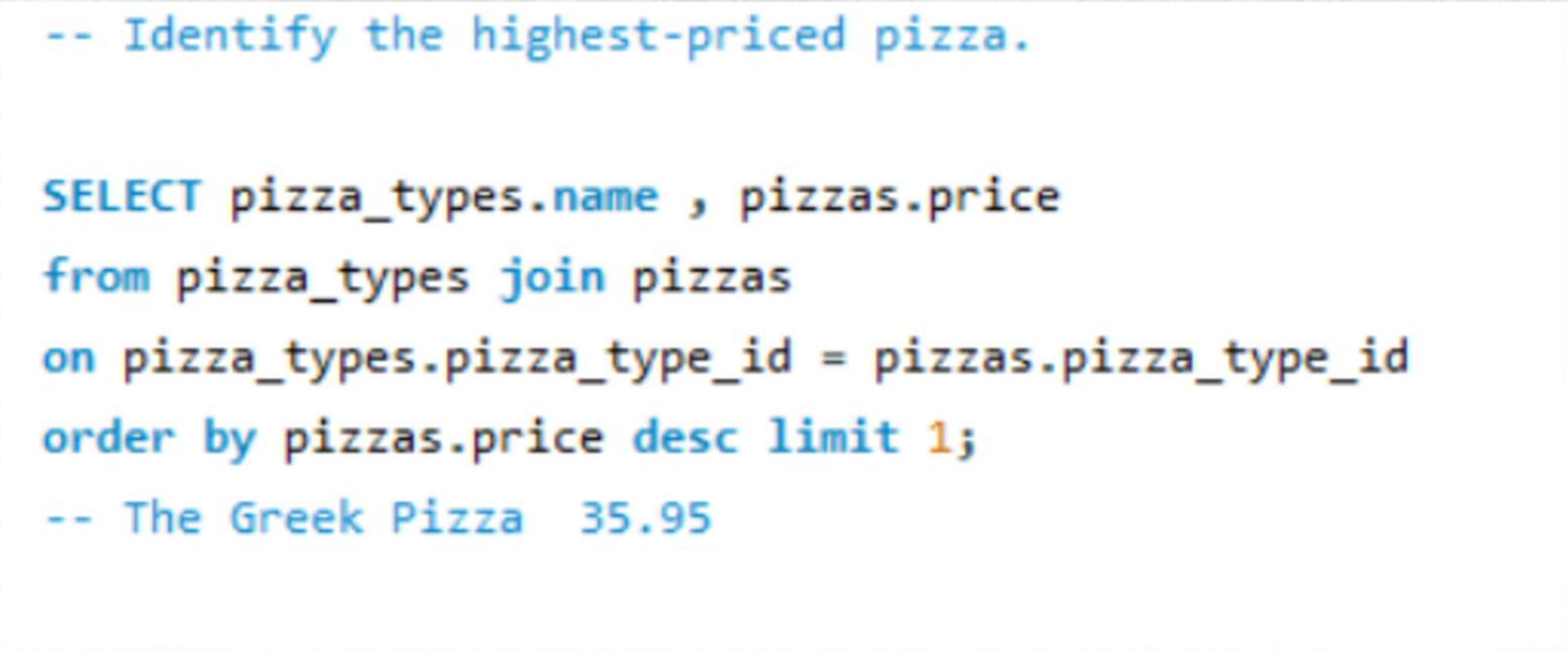
```
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
    orders_details AS od
        INNER JOIN
    pizzas AS p ON p.pizza_id = od.pizza_id;
```



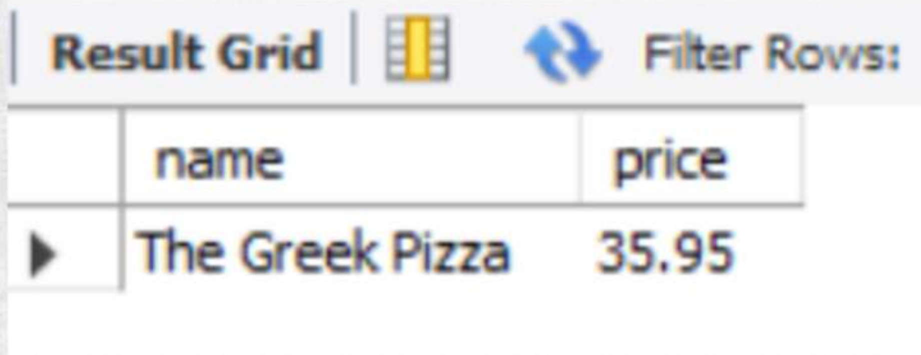
```
total_revenue
817860.05
```

NEXT >

IDENTIFY THE HIGHEST-PRICED PIZZA.



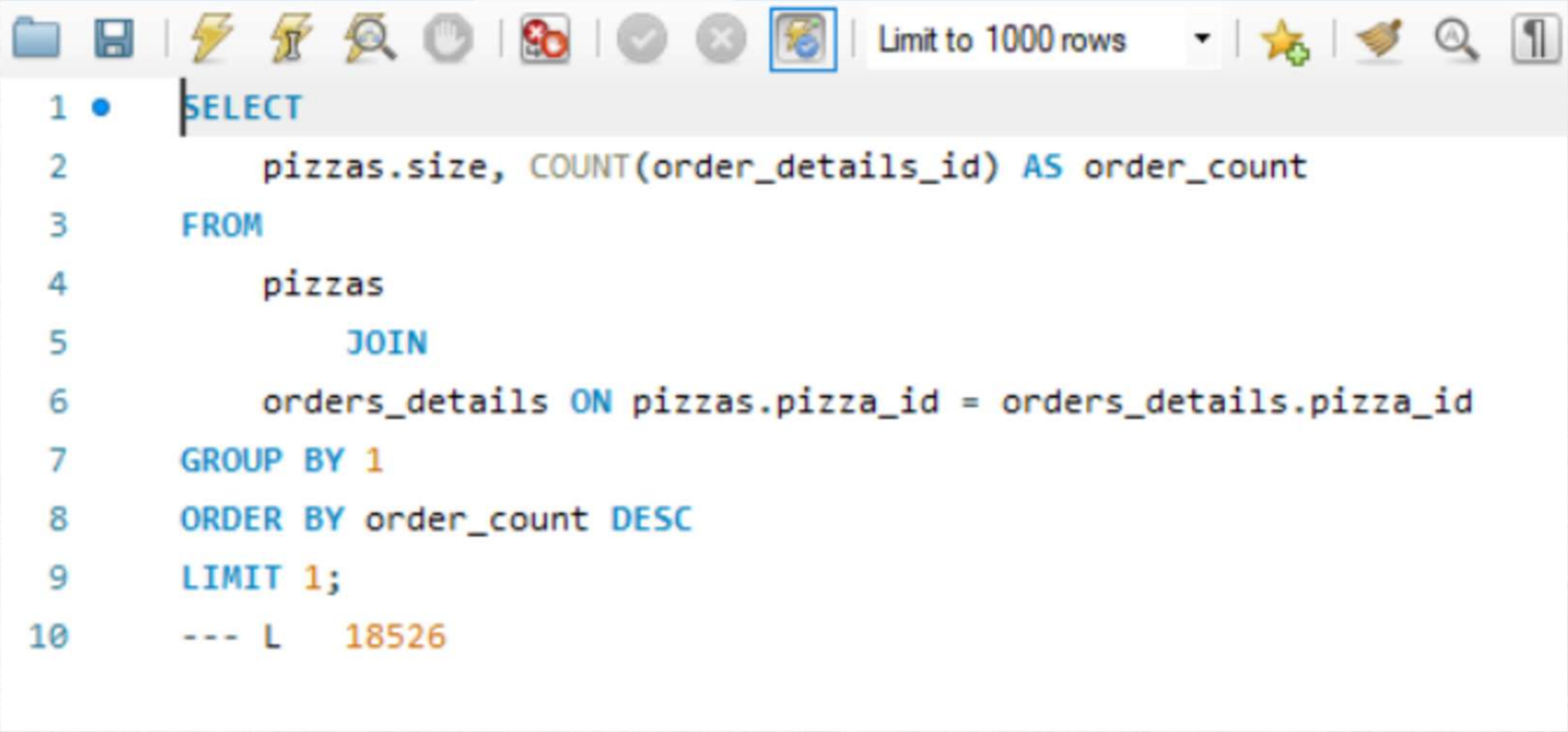
```
-- Identify the highest-priced pizza.

SELECT pizza_types.name , pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc limit 1;
-- The Greek Pizza  35.95
```



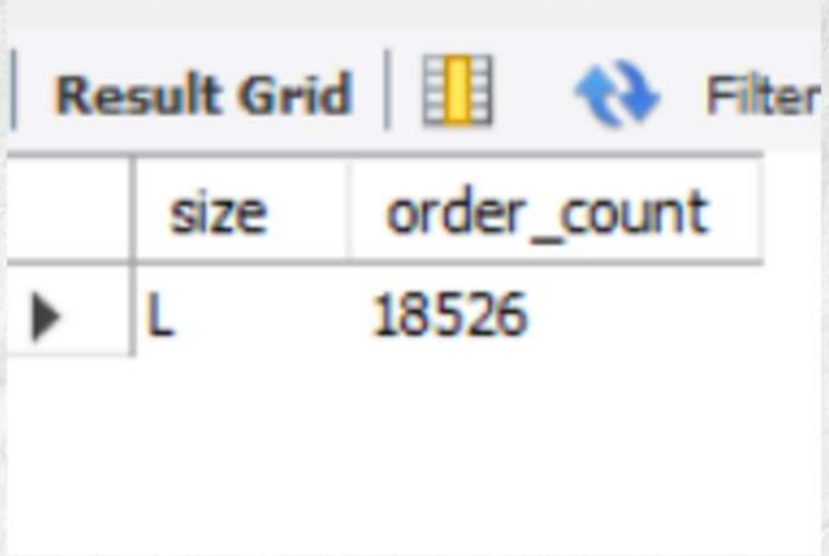
name	price
The Greek Pizza	35.95

NEXT >

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



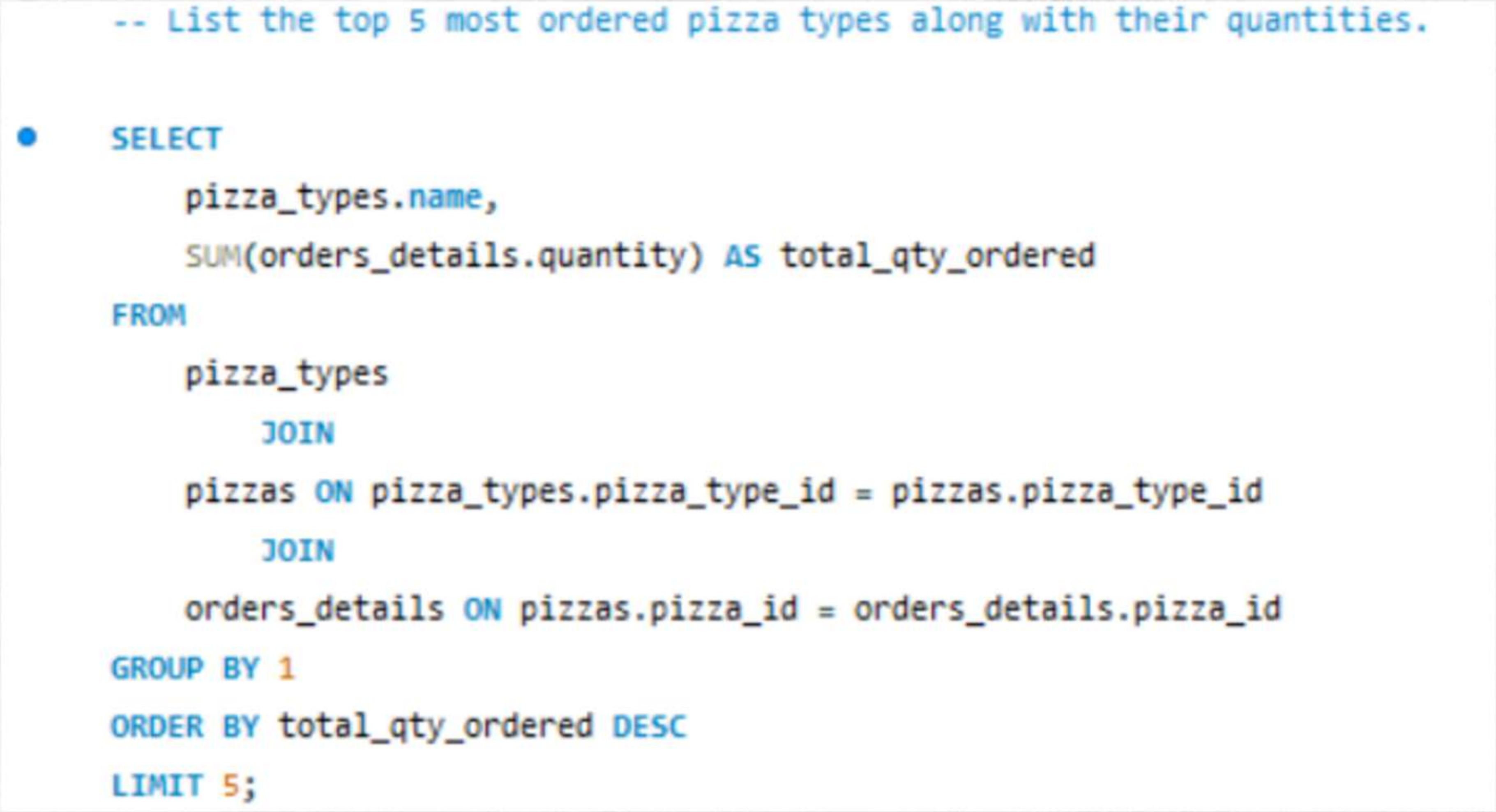
```
SELECT
    pizzas.size, COUNT(order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY 1
ORDER BY order_count DESC
LIMIT 1;
--- L   18526
```



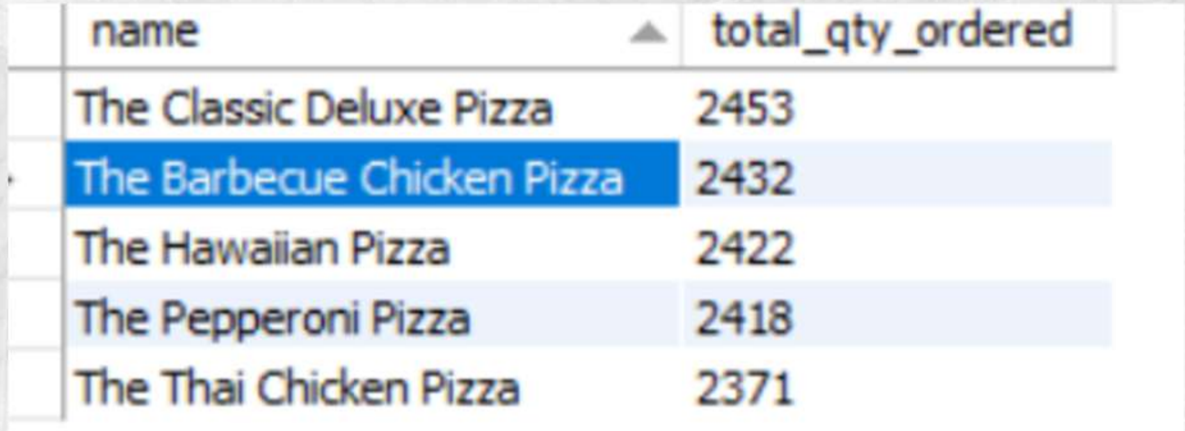
size	order_count
L	18526

NEXT >

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



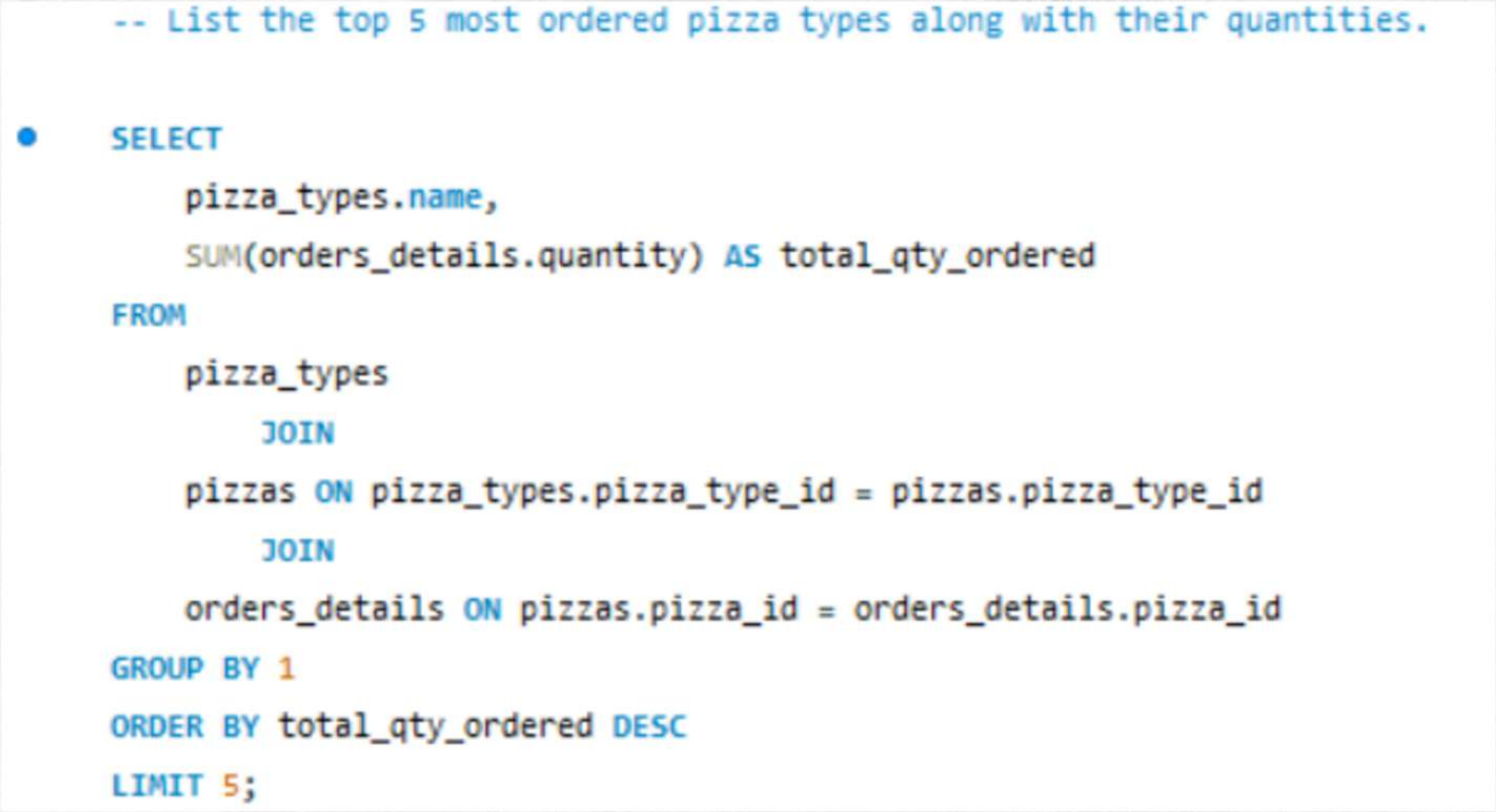
```
-- List the top 5 most ordered pizza types along with their quantities.

SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS total_qty_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY 1
ORDER BY total_qty_ordered DESC
LIMIT 5;
```



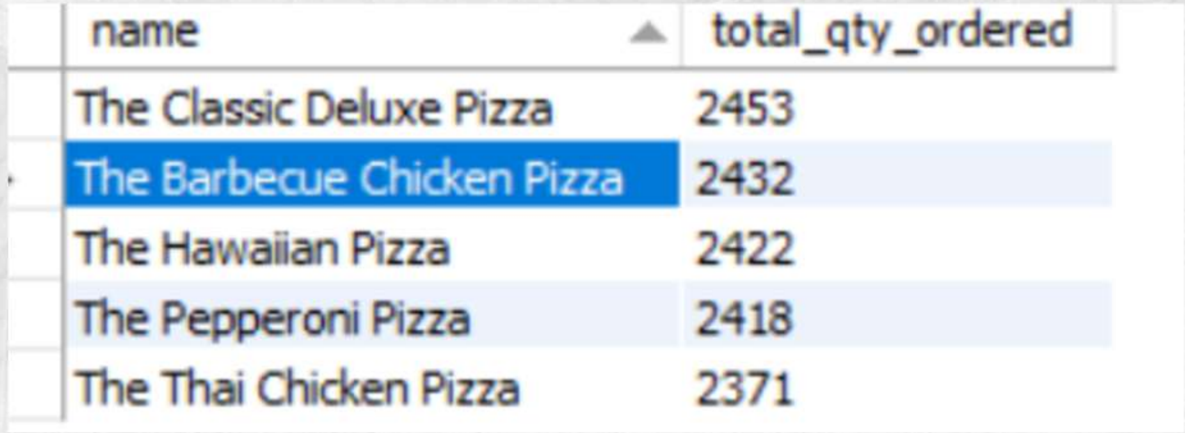
name	total_qty_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

NEXT >

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(orders_details.quantity) AS total_qty_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY 1
ORDER BY total_qty_ordered DESC
LIMIT 5;
```

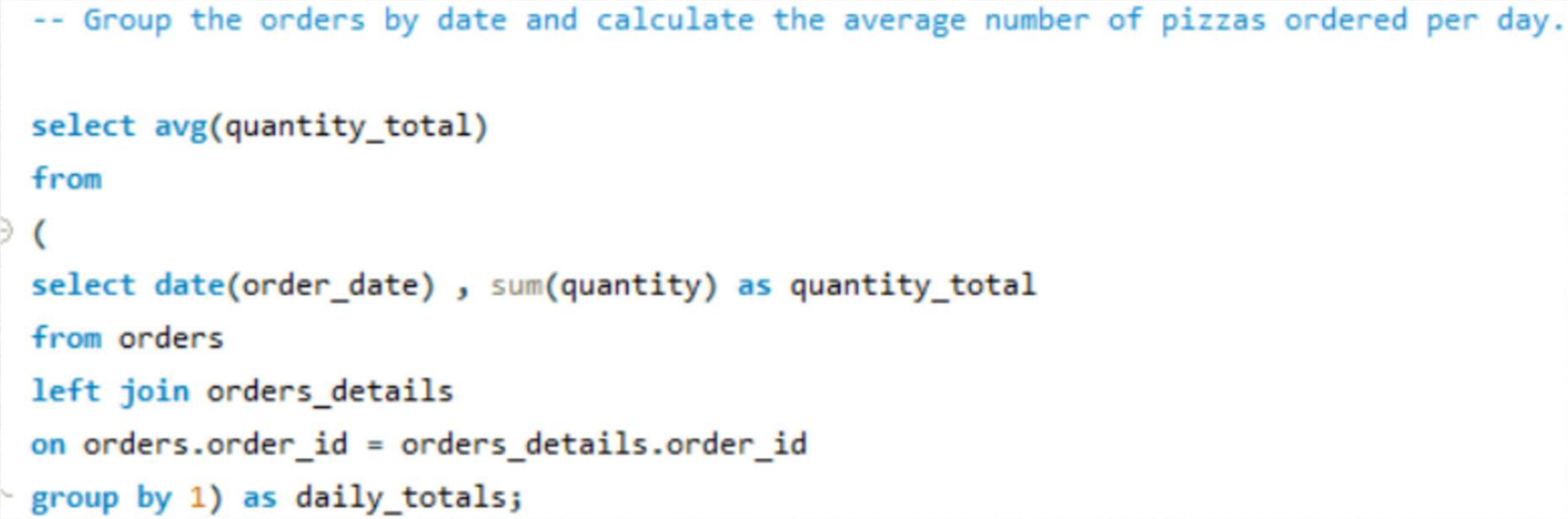


name	total_qty_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

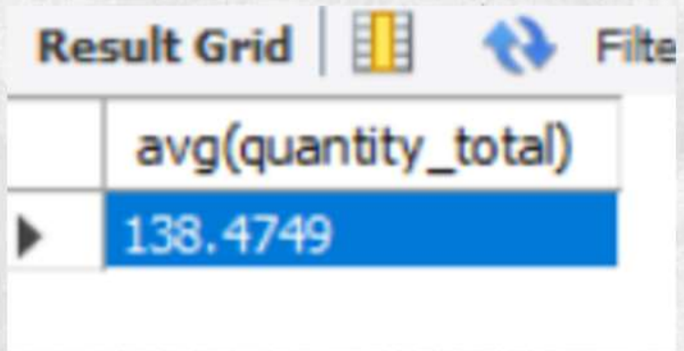
NEXT >

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

PIZZA HUT



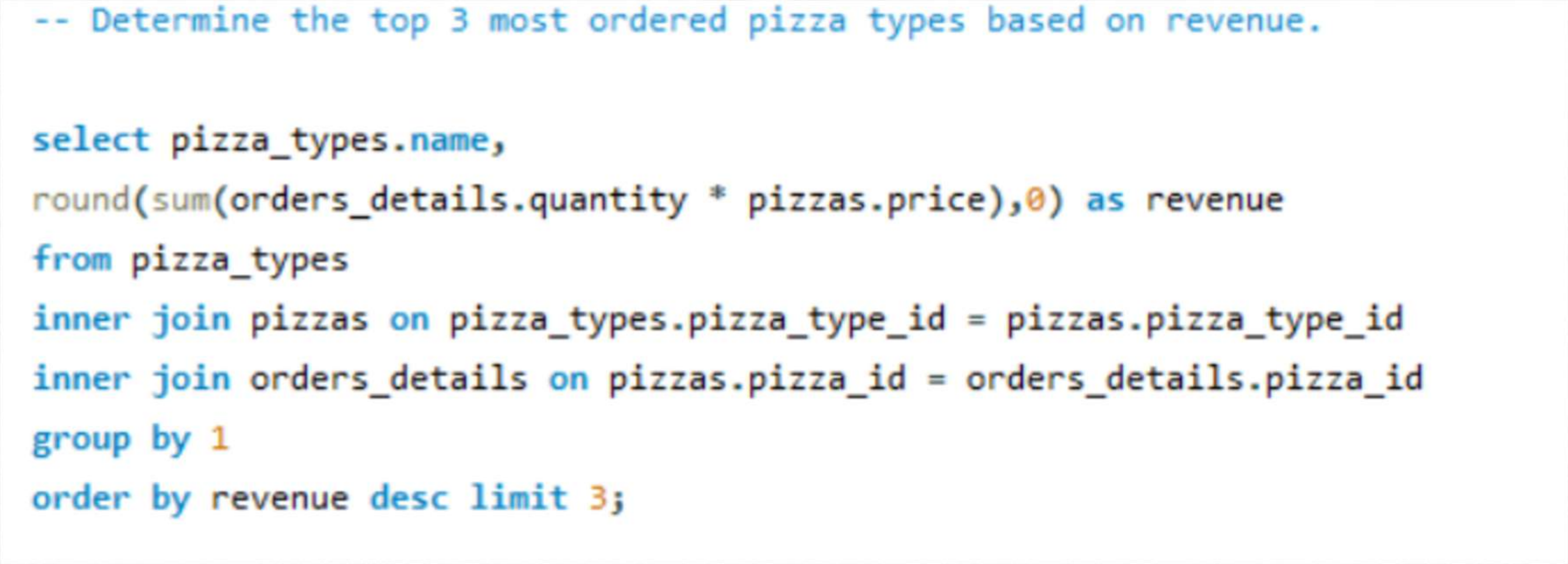
```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

select avg(quantity_total)
from
(
select date(order_date) , sum(quantity) as quantity_total
from orders
left join orders_details
on orders.order_id = orders_details.order_id
group by 1) as daily_totals;
```



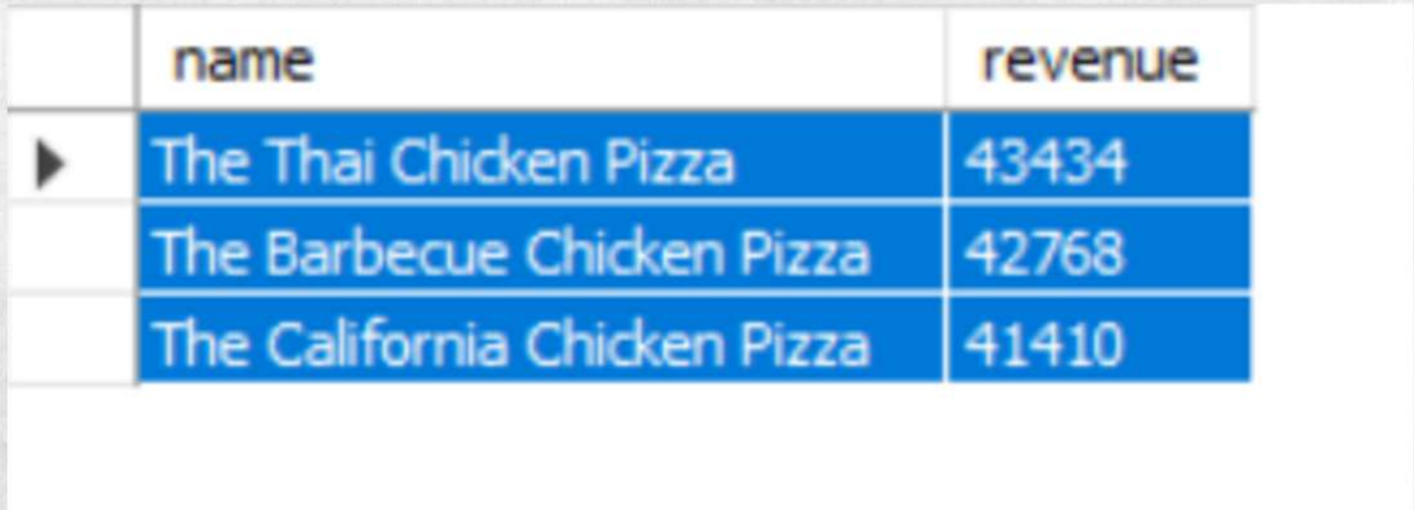
avg(quantity_total)
138.4749

NEXT >

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



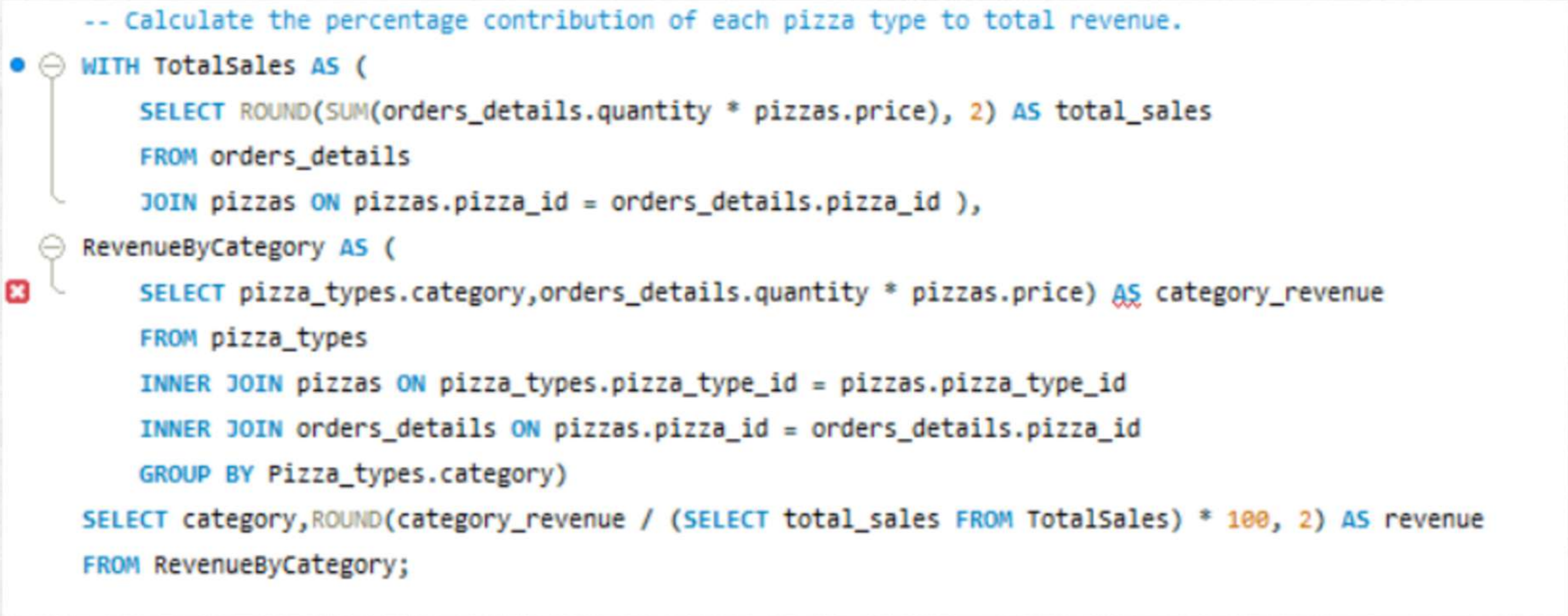
```
-- Determine the top 3 most ordered pizza types based on revenue.

select pizza_types.name,
round(sum(orders_details.quantity * pizzas.price),0) as revenue
from pizza_types
inner join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
inner join orders_details on pizzas.pizza_id = orders_details.pizza_id
group by 1
order by revenue desc limit 3;
```



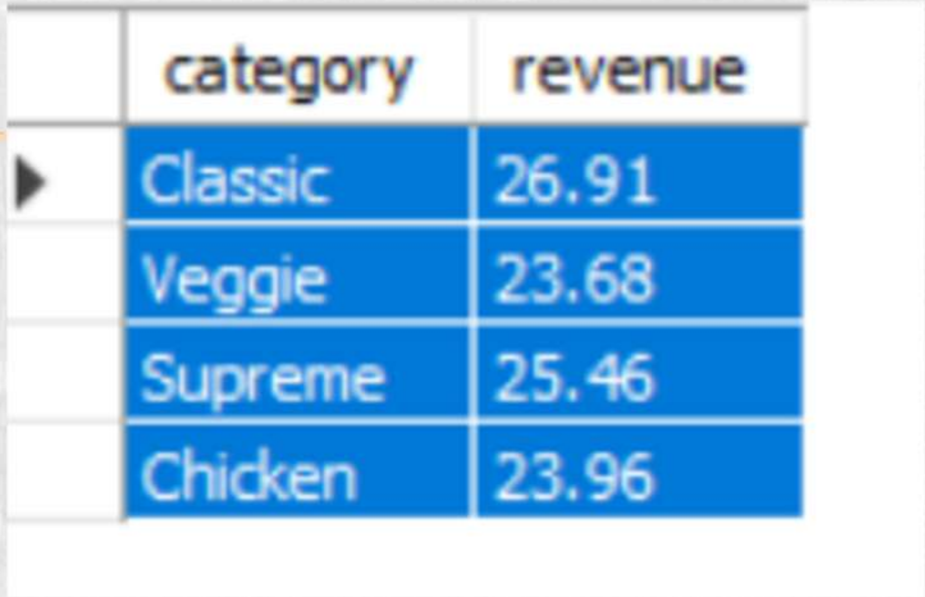
name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

NEXT >

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



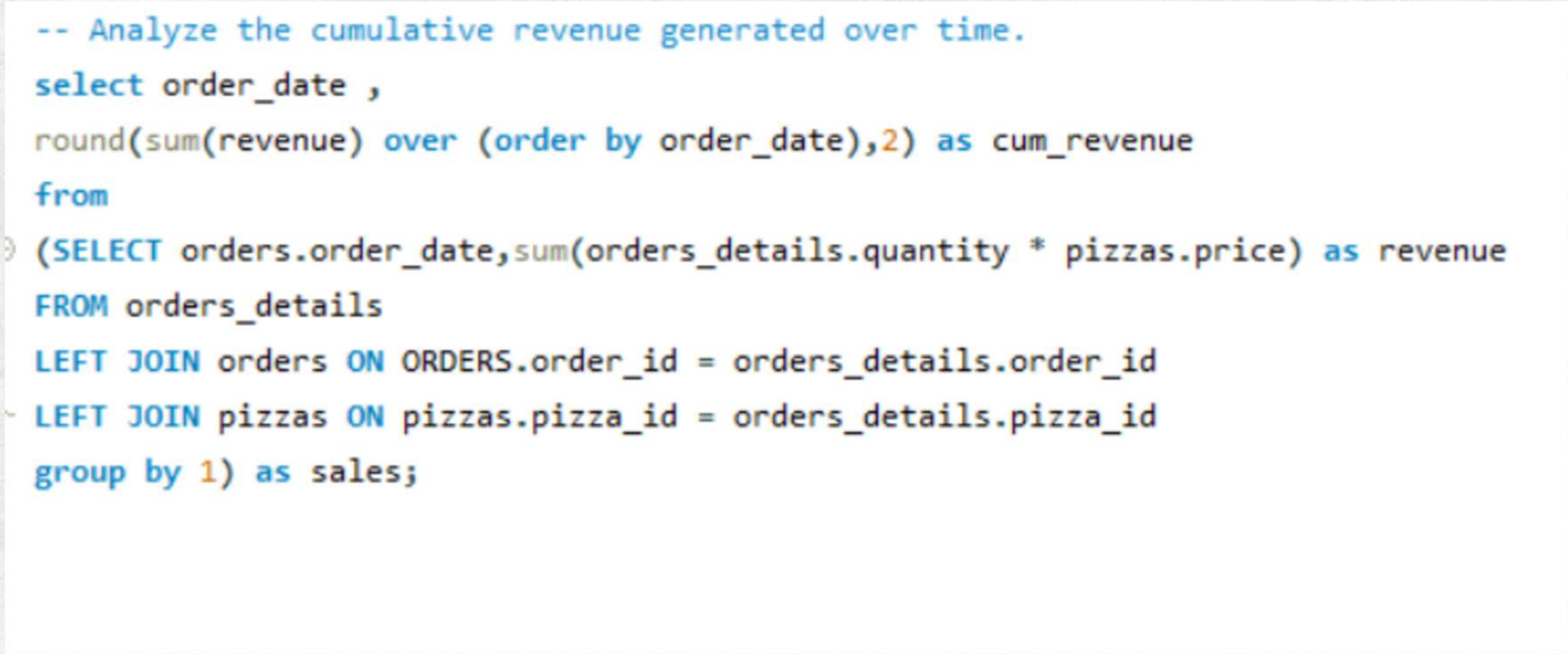
```
-- Calculate the percentage contribution of each pizza type to total revenue.
WITH TotalSales AS (
    SELECT ROUND(SUM(orders_details.quantity * pizzas.price), 2) AS total_sales
    FROM orders_details
    JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id ),
RevenueByCategory AS (
    SELECT pizza_types.category,orders_details.quantity * pizzas.price) AS category_revenue
    FROM pizza_types
    INNER JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    INNER JOIN orders_details ON pizzas.pizza_id = orders_details.pizza_id
    GROUP BY Pizza_types.category)
SELECT category,ROUND(category_revenue / (SELECT total_sales FROM TotalSales) * 100, 2) AS revenue
FROM RevenueByCategory;
```



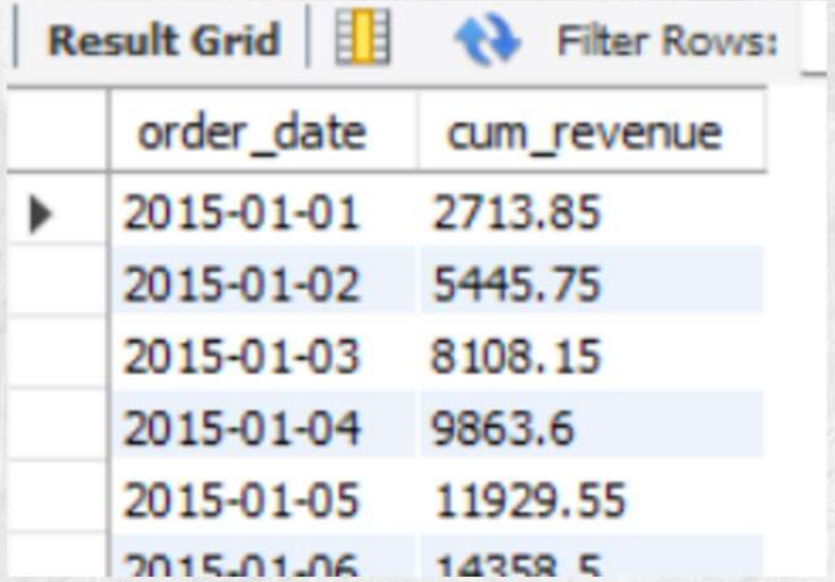
category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

NEXT >

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



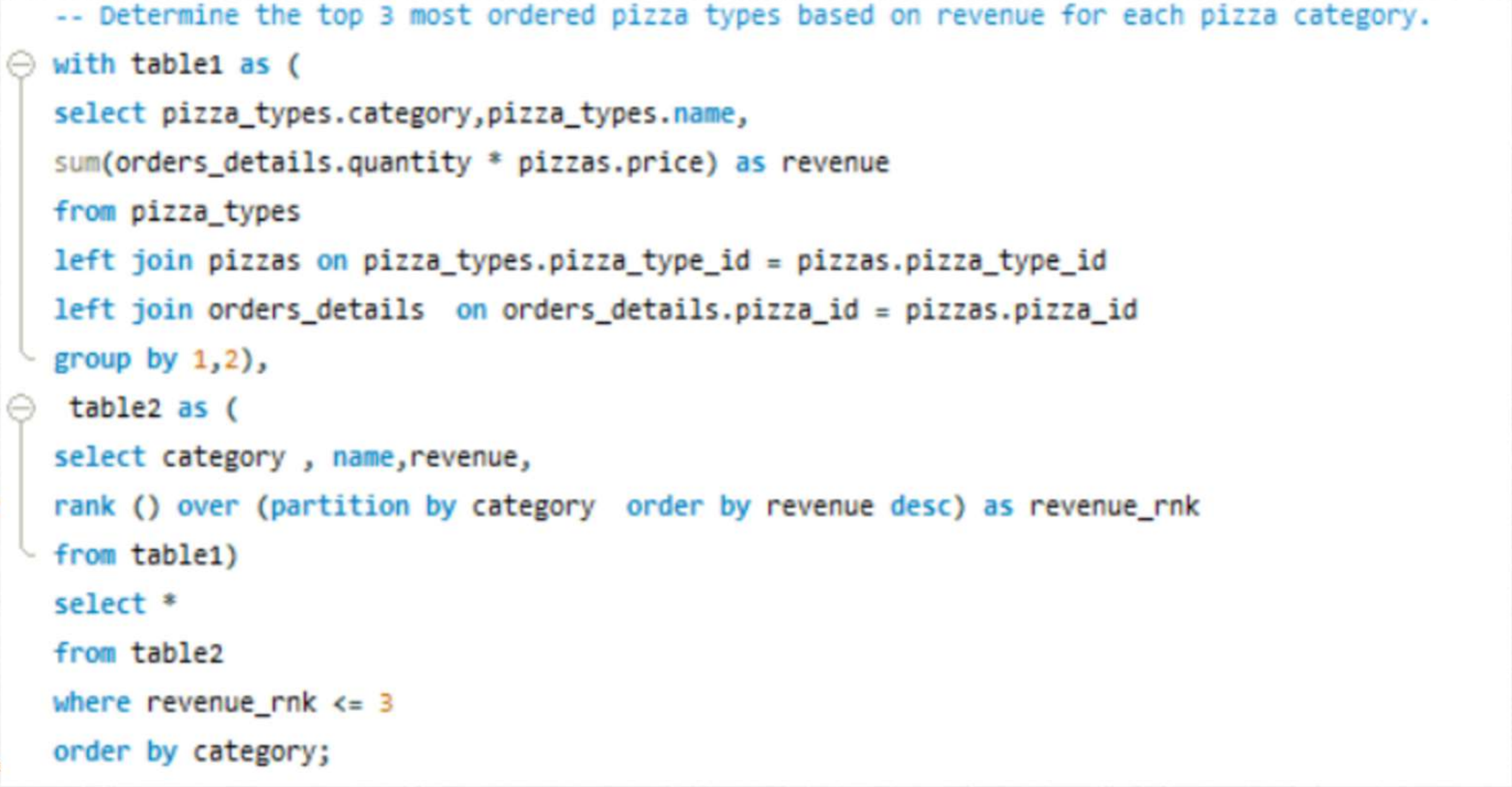
```
-- Analyze the cumulative revenue generated over time.
select order_date ,
round(sum(revenue) over (order by order_date),2) as cum_revenue
from
(SELECT orders.order_date,sum(orders_details.quantity * pizzas.price) as revenue
FROM orders_details
LEFT JOIN orders ON ORDERS.order_id = orders_details.order_id
LEFT JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id
group by 1) as sales;
```



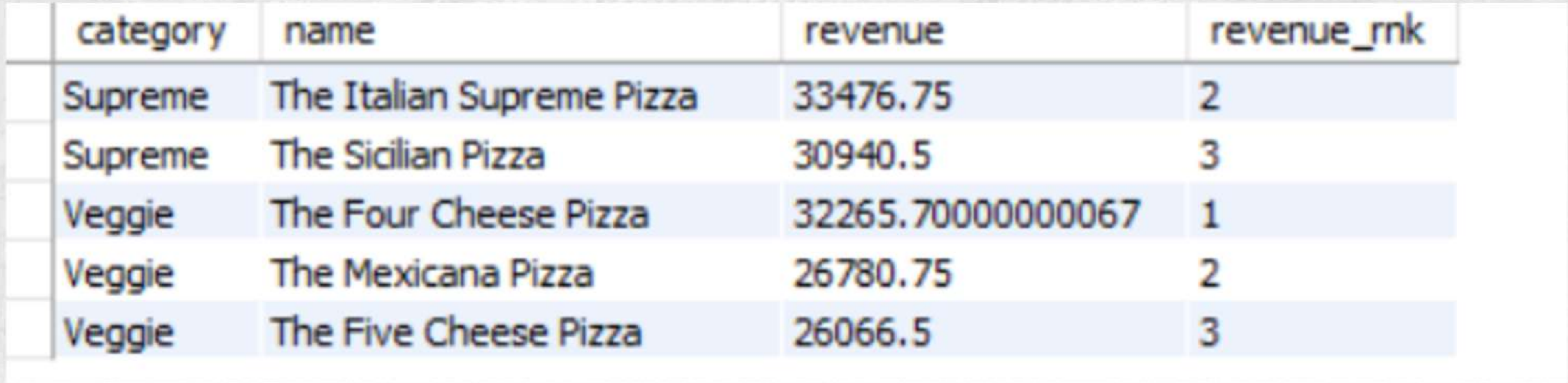
order_date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5

NEXT >

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



```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
with table1 as (
select pizza_types.category,pizza_types.name,
sum(orders_details.quantity * pizzas.price) as revenue
from pizza_types
left join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
left join orders_details  on orders_details.pizza_id = pizzas.pizza_id
group by 1,2),
 table2 as (
select category , name,revenue,
rank () over (partition by category  order by revenue desc) as revenue_rnk
from table1)
select *
from table2
where revenue_rnk <= 3
order by category;
```



category	name	revenue	revenue_rnk
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000067	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

NEXT >

CONTACT ME

LARANA PIZZA

THANK YOU!

AFROSE.BASHA1630@GMAIL.COM

< BACK