PIZZA SALES

NEXT >

ABOUT ME

Hi, I'm Vaishnavi Gawande, a passionate Data Analyst with hands-on experience in SQL and Excel. I enjoy transforming raw data into meaningful insights that drive decision-making. This pizza sales analysis project showcases my ability to solve real-world business problems using only SQL queries without any external tools.

< BACK

NEXT >

PIZZA SALES ANALYSIS

This project focuses on analyzing pizza sales data using pure SQL queries. My goal was to identify sales performance, customer preferences, and revenue trends to help a pizza business make data-driven decisions.

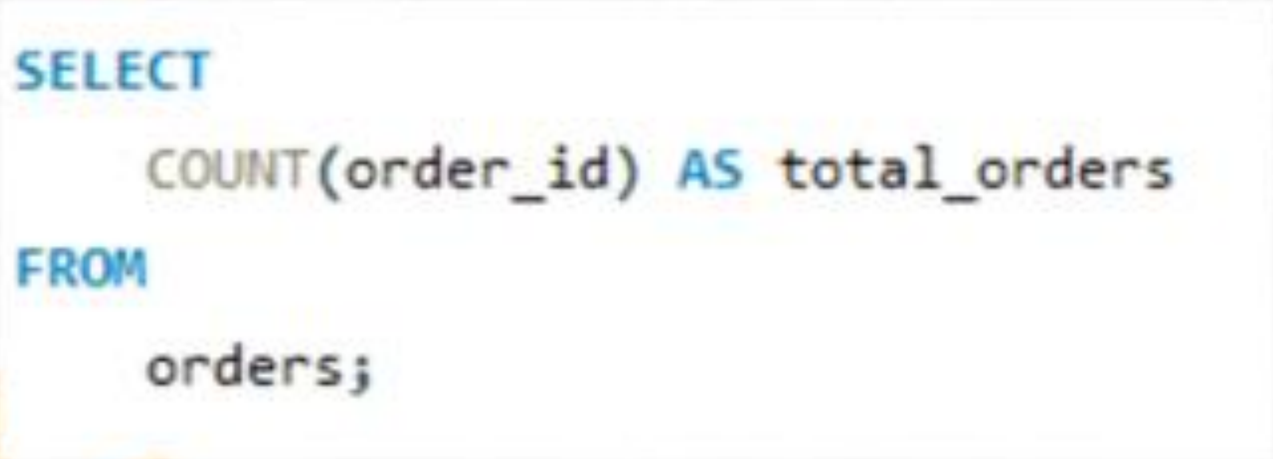
This structure allows us to:

- Track sales volume and revenue per pizza type, size, or category.
- Identify top-performing products based on order frequency.
- Analyze time-based trends like peak sales days or hours.
- Understand customer demand patterns for better inventory & marketing planning

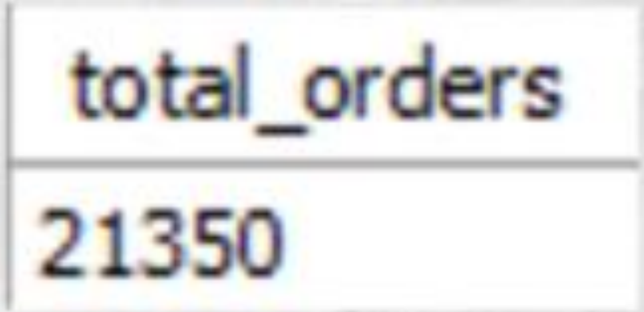
< BACK

NEXT >

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



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

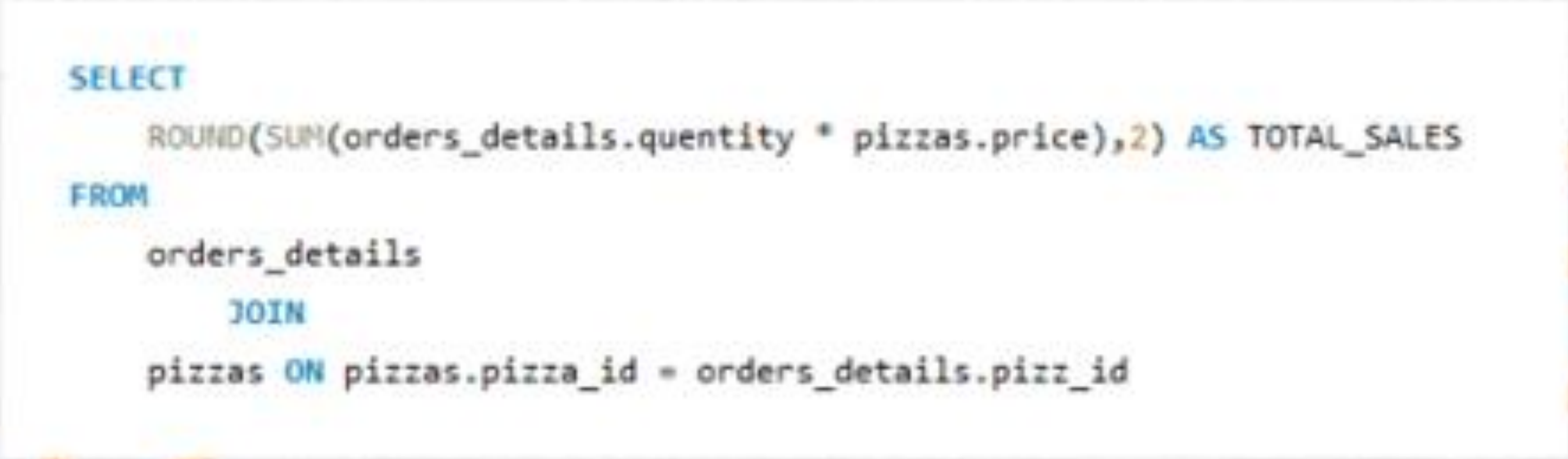


total_orders
21350

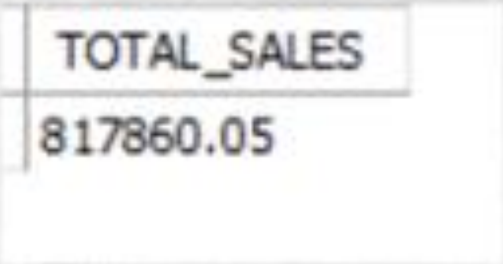
< BACK

NEXT >

CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES.



```
SELECT
    ROUND(SUM(orders_details.quentity * pizzas.price),2) AS TOTAL_SALES
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizz_id
```

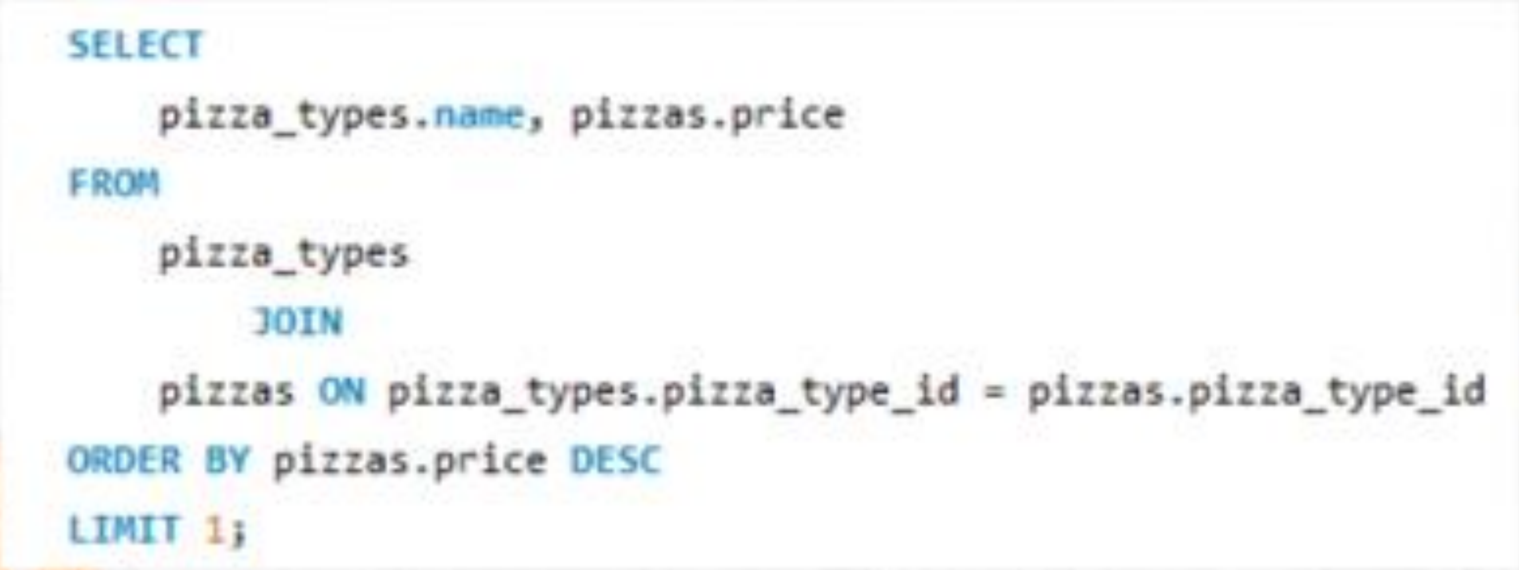


```
TOTAL_SALES
817860.05
```

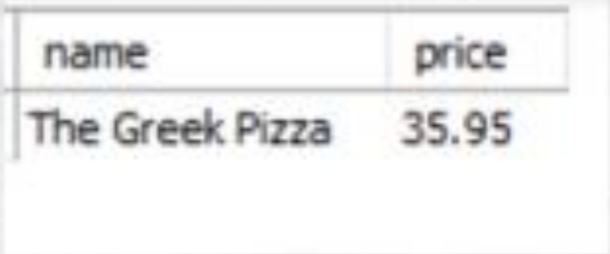
< BACK

NEXT >

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;
```

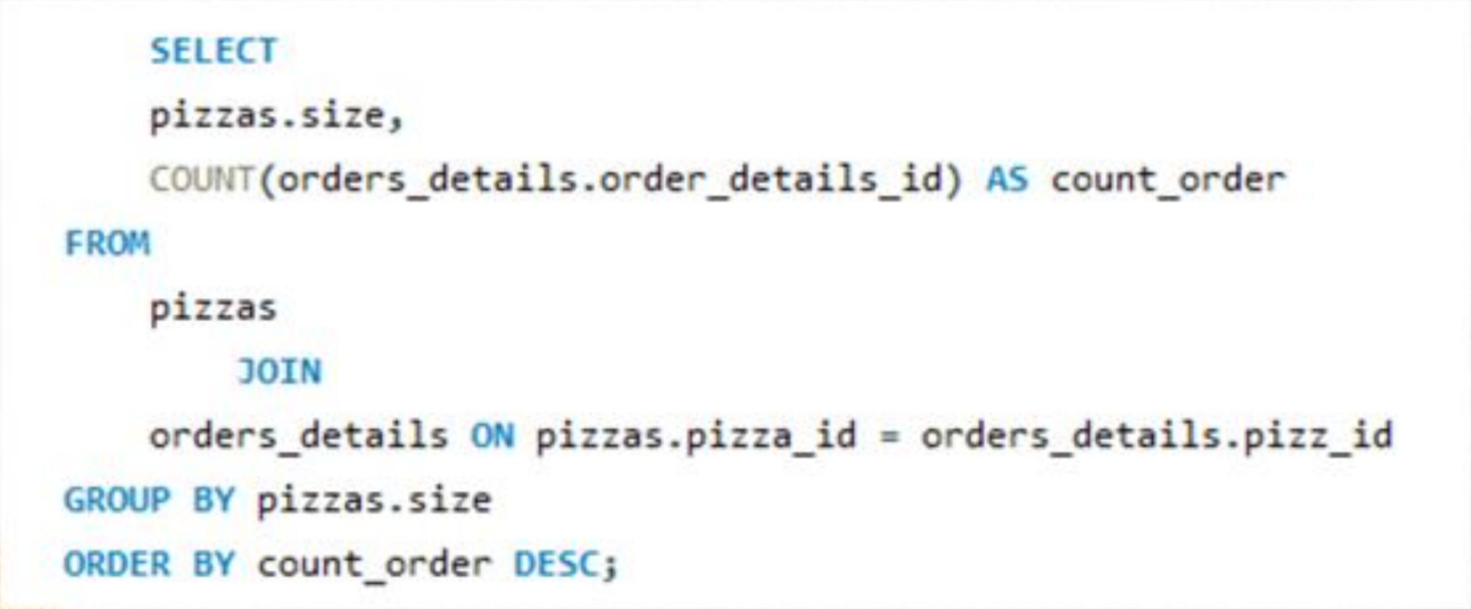


name	price
The Greek Pizza	35.95

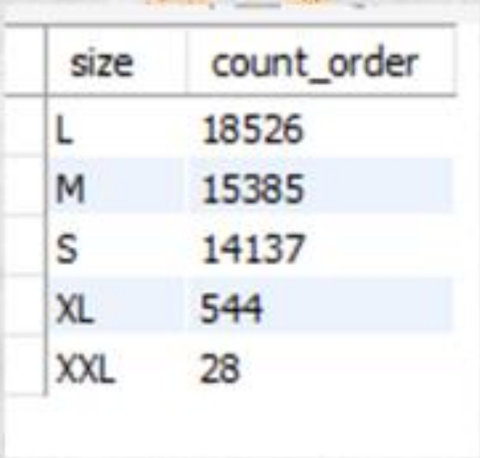
< BACK

NEXT >

IDENTIDY THE MOST COMMON SIZE ORDERED.



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

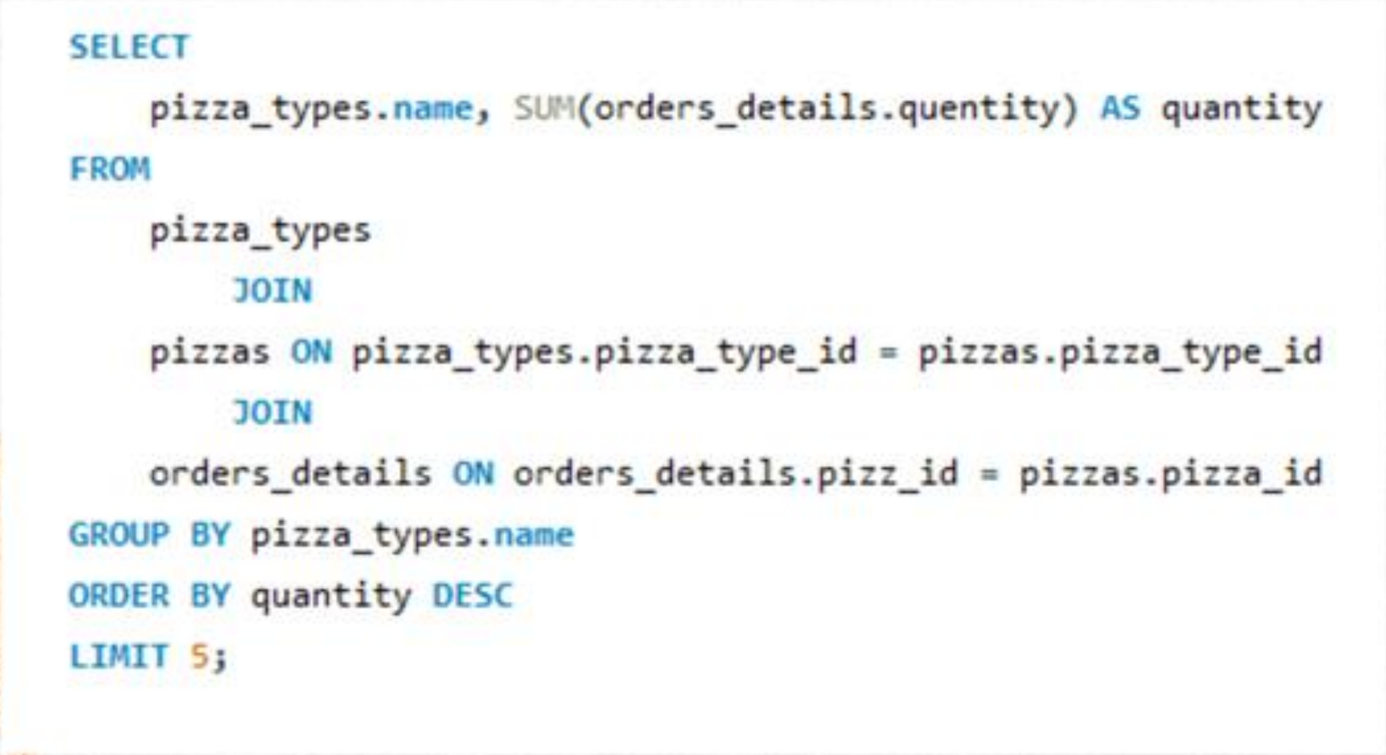


size	count_order
L	18526
M	15385
S	14137
XL	544
XXL	28

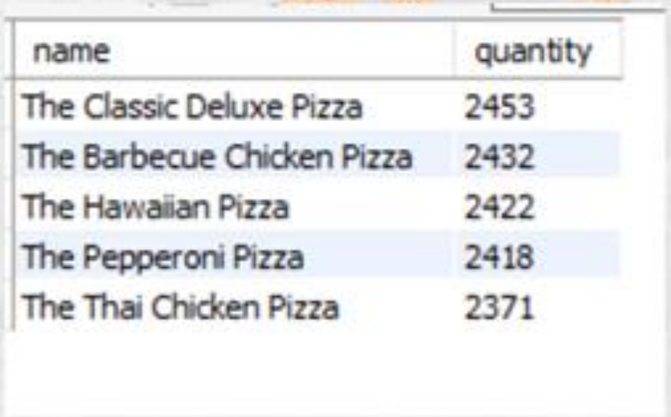
< BACK

NEXT >

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



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

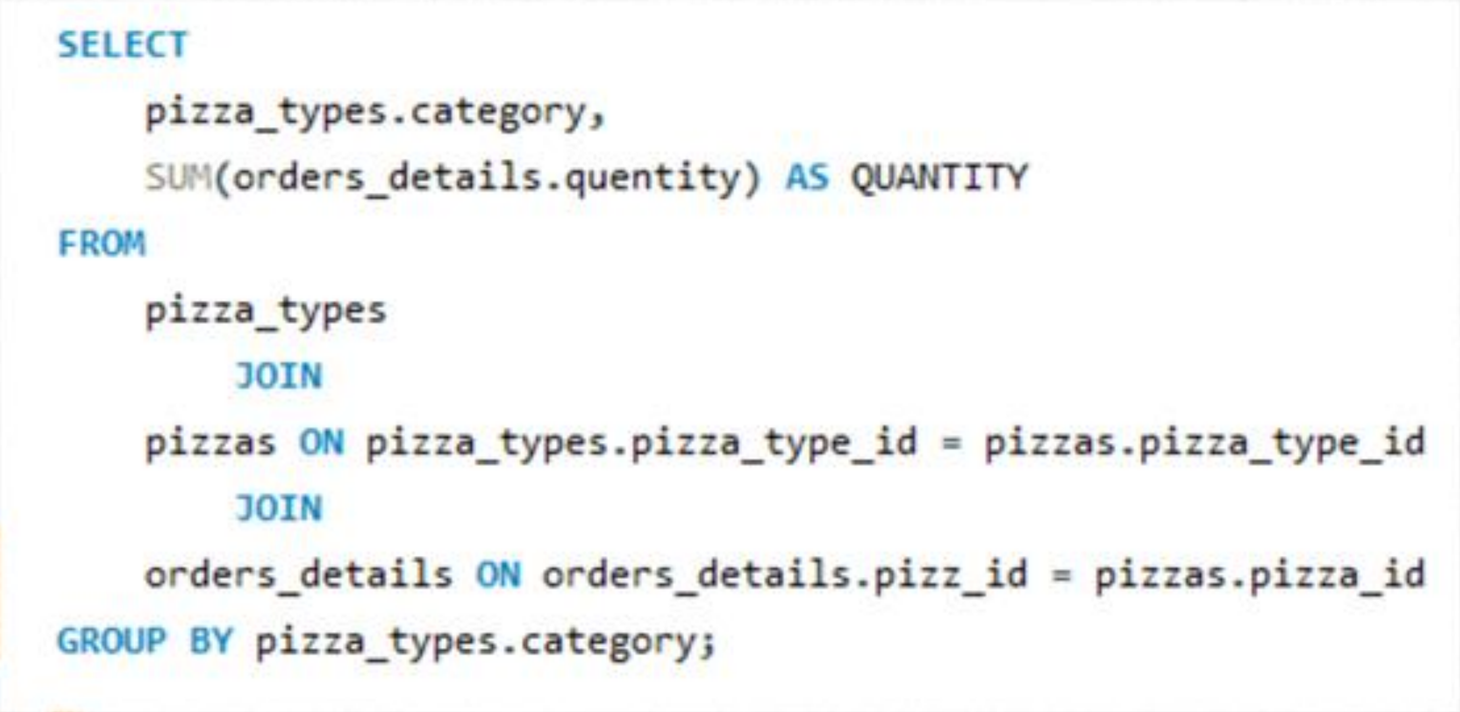


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

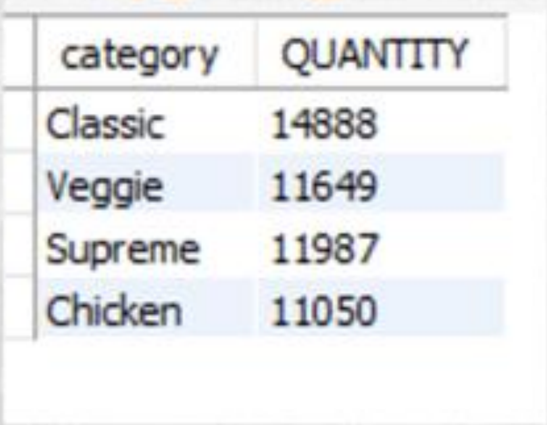
< BACK

NEXT >

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



```
SELECT
    pizza_types.category,
    SUM(orders_details.quentity) AS QUANTITY
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizz_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

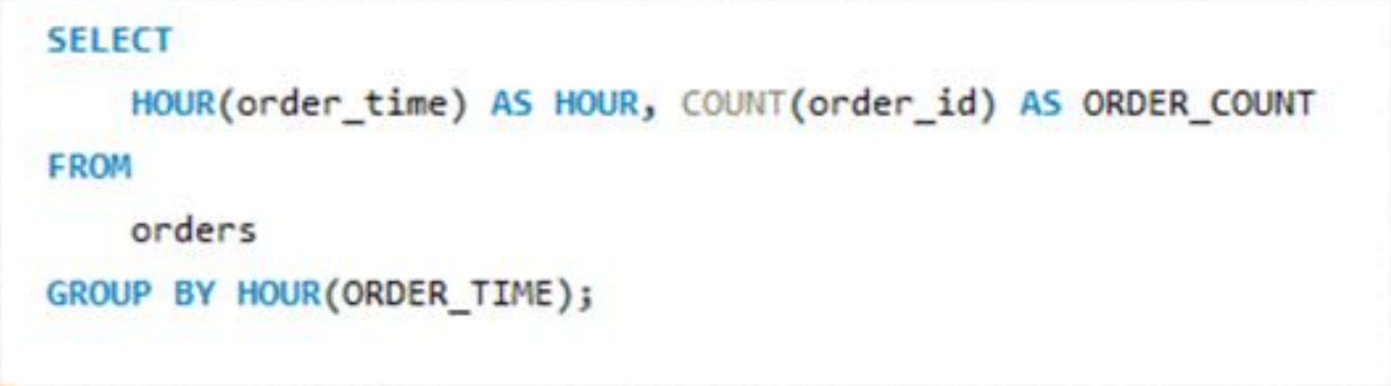


category	QUANTITY
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

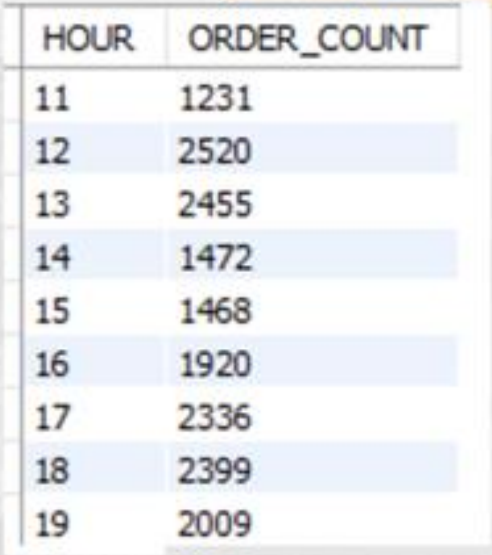
< BACK

NEXT >

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(ORDER_TIME);
```

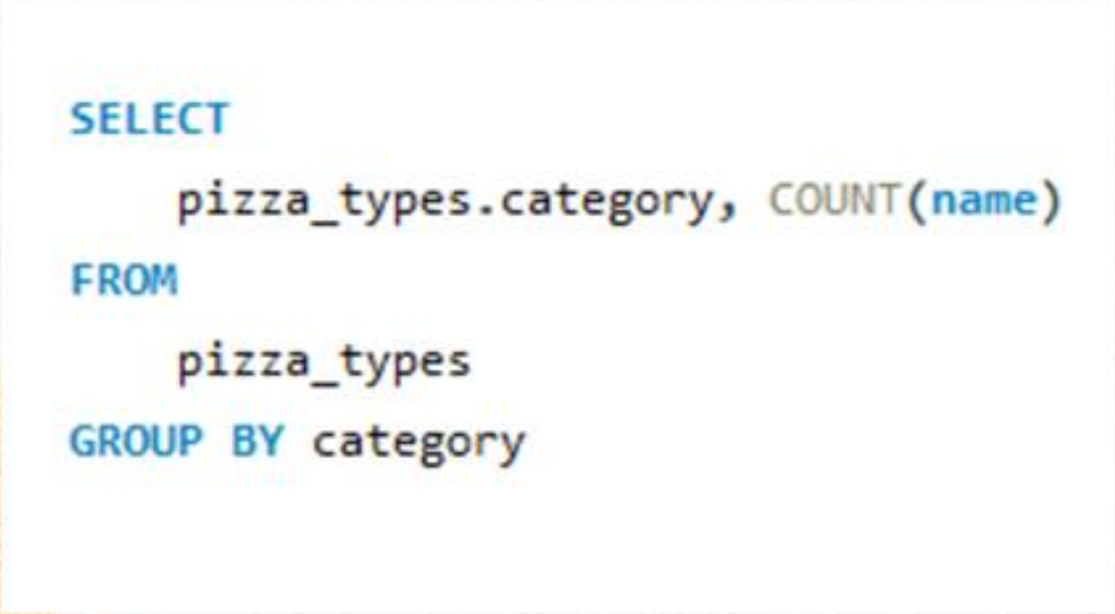


HOUR	ORDER_COUNT
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

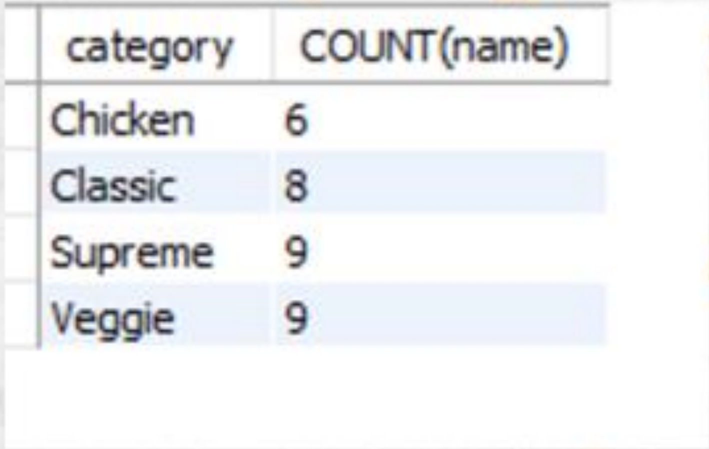
< BACK

NEXT >

JOINT RELEVENT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



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

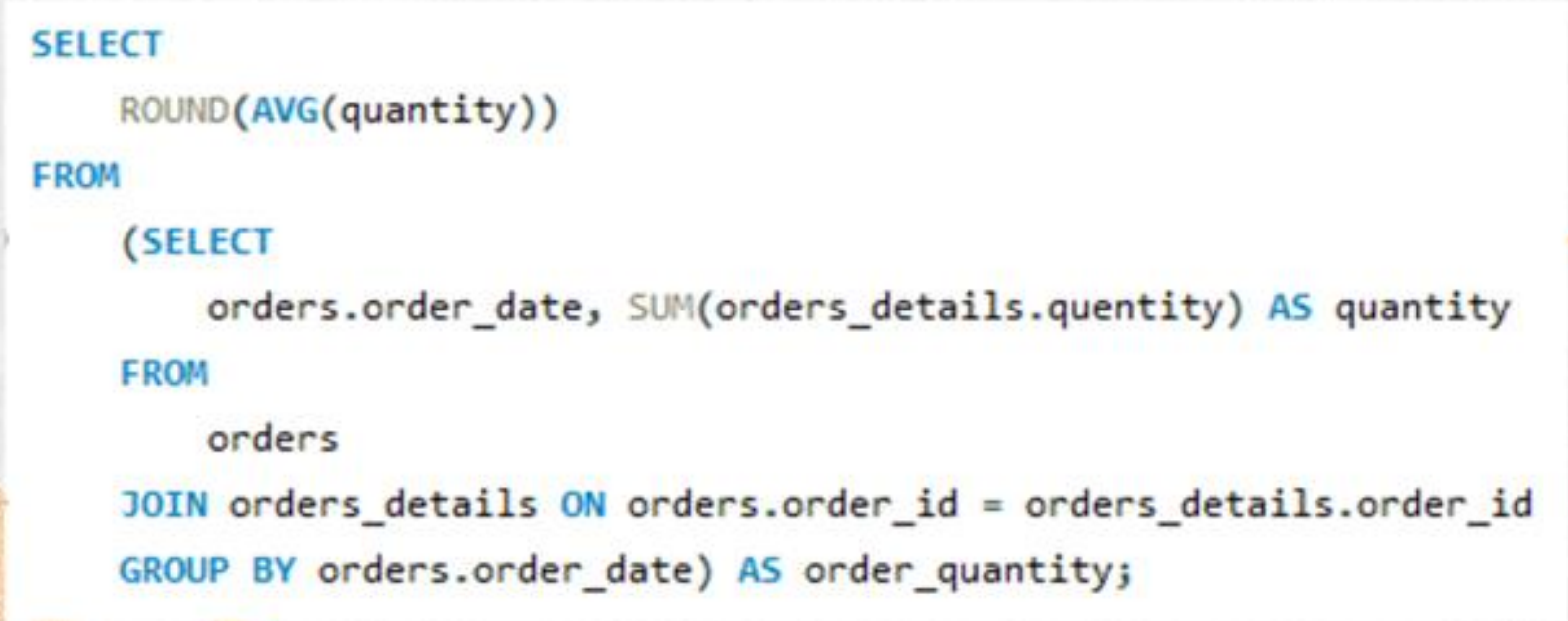


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

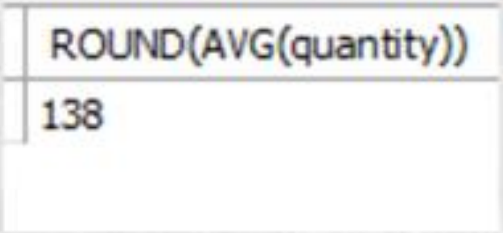
< BACK

NEXT >

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



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

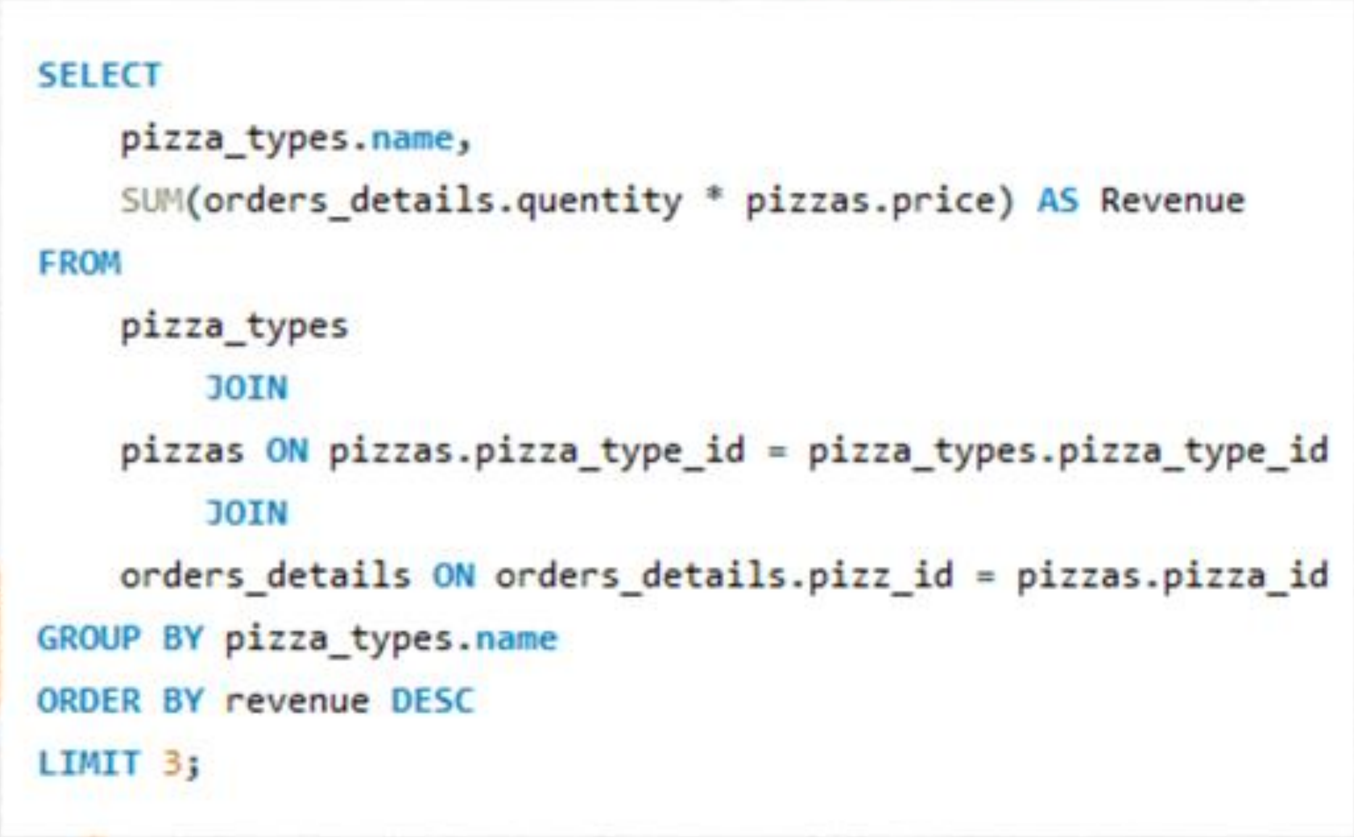


```
ROUND(AVG(quantity))
138
```

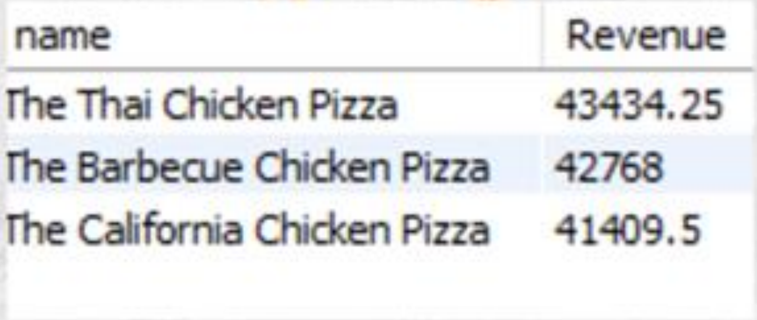
< BACK

NEXT >

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



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

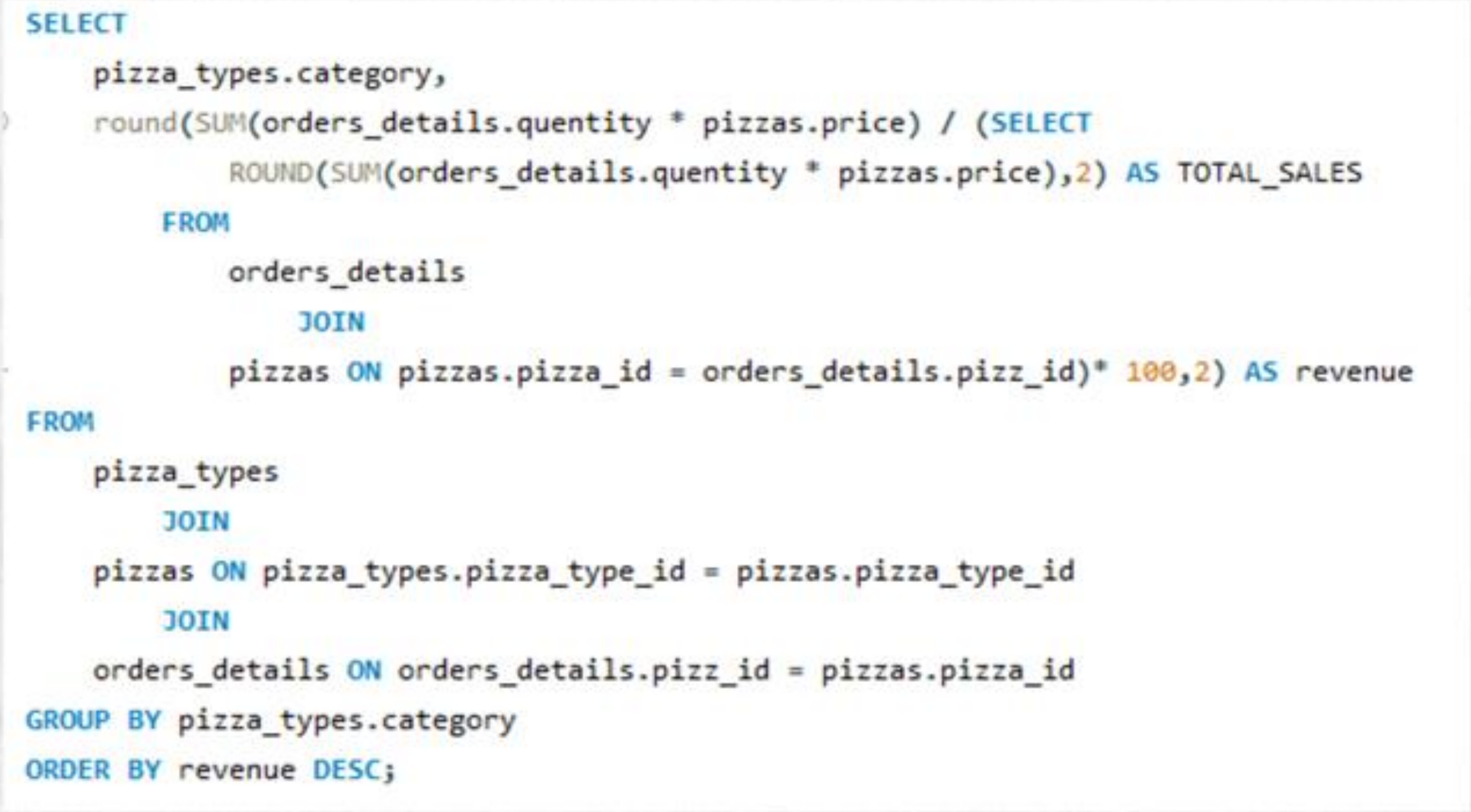


name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

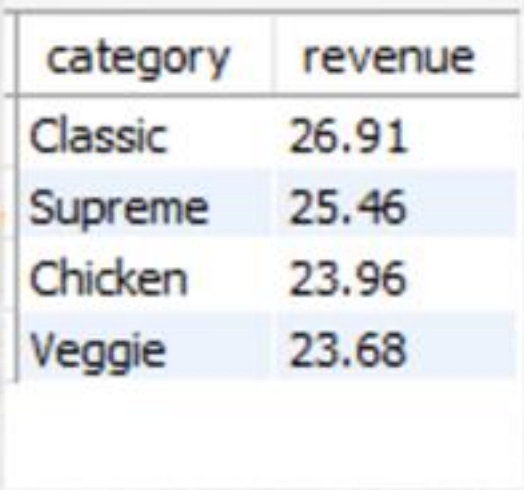
< BACK

NEXT >

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



```
SELECT
    pizza_types.category,
    round(SUM(orders_details.quentity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quentity * pizzas.price),2) AS TOTAL_SALES
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizz_id)* 100,2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizz_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

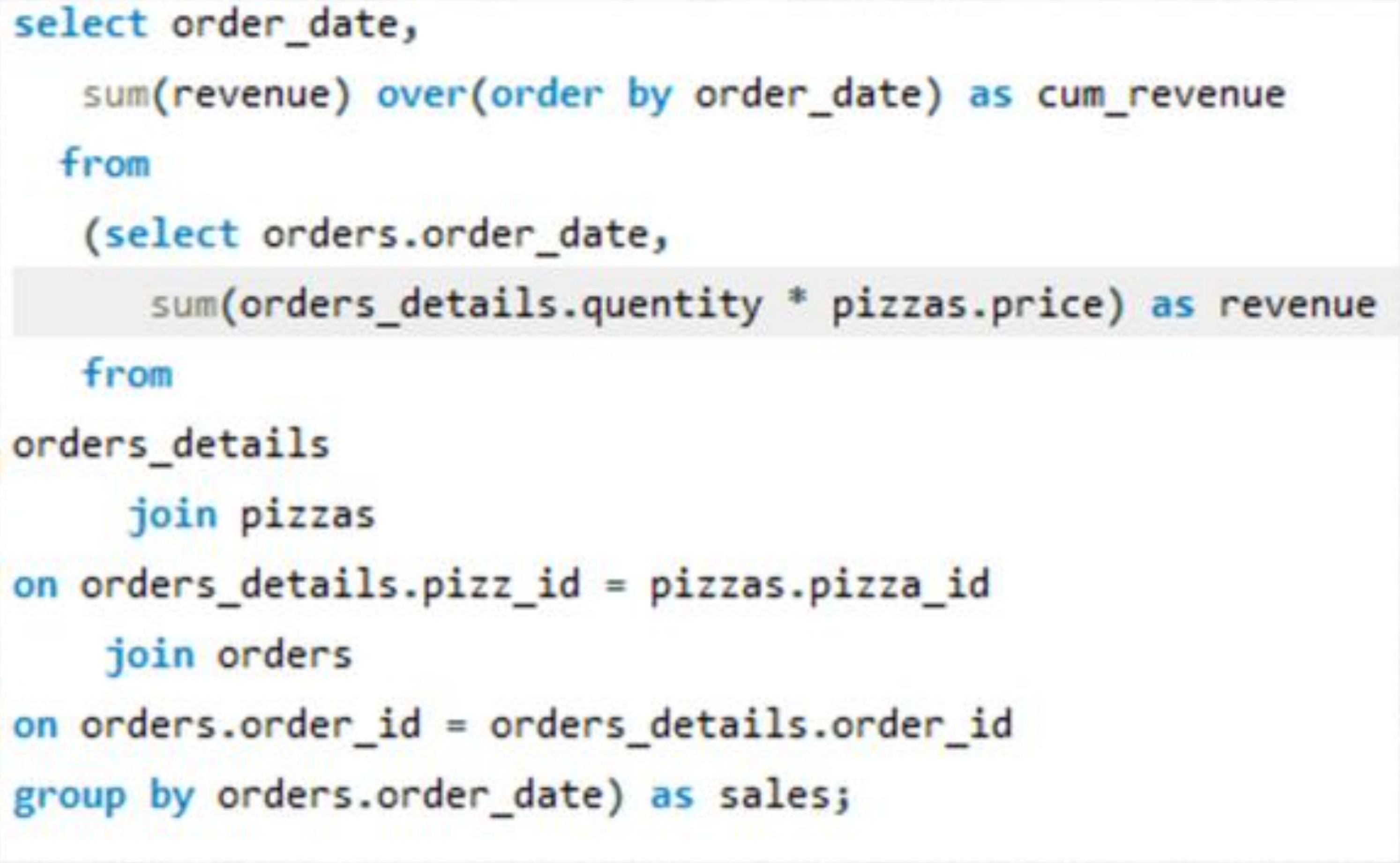


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

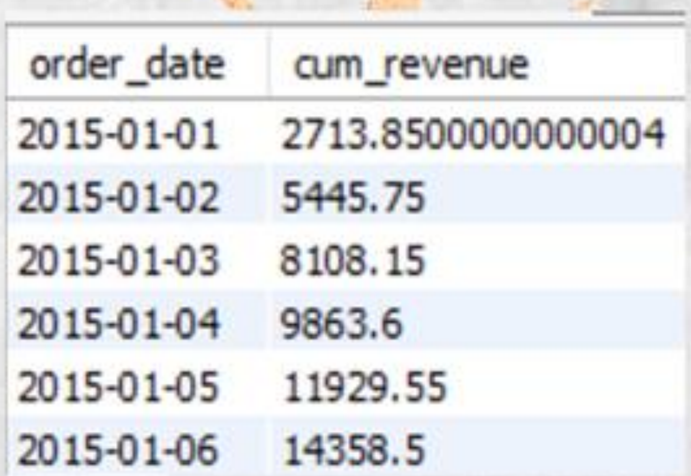
< BACK

NEXT >

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



```
select order_date,
   sum(revenue) over(order by order_date) as cum_revenue
  from
   (select orders.order_date,
      sum(orders_details.quentity * pizzas.price) as revenue
   from
orders_details
     join pizzas
on orders_details.pizz_id = pizzas.pizza_id
    join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```

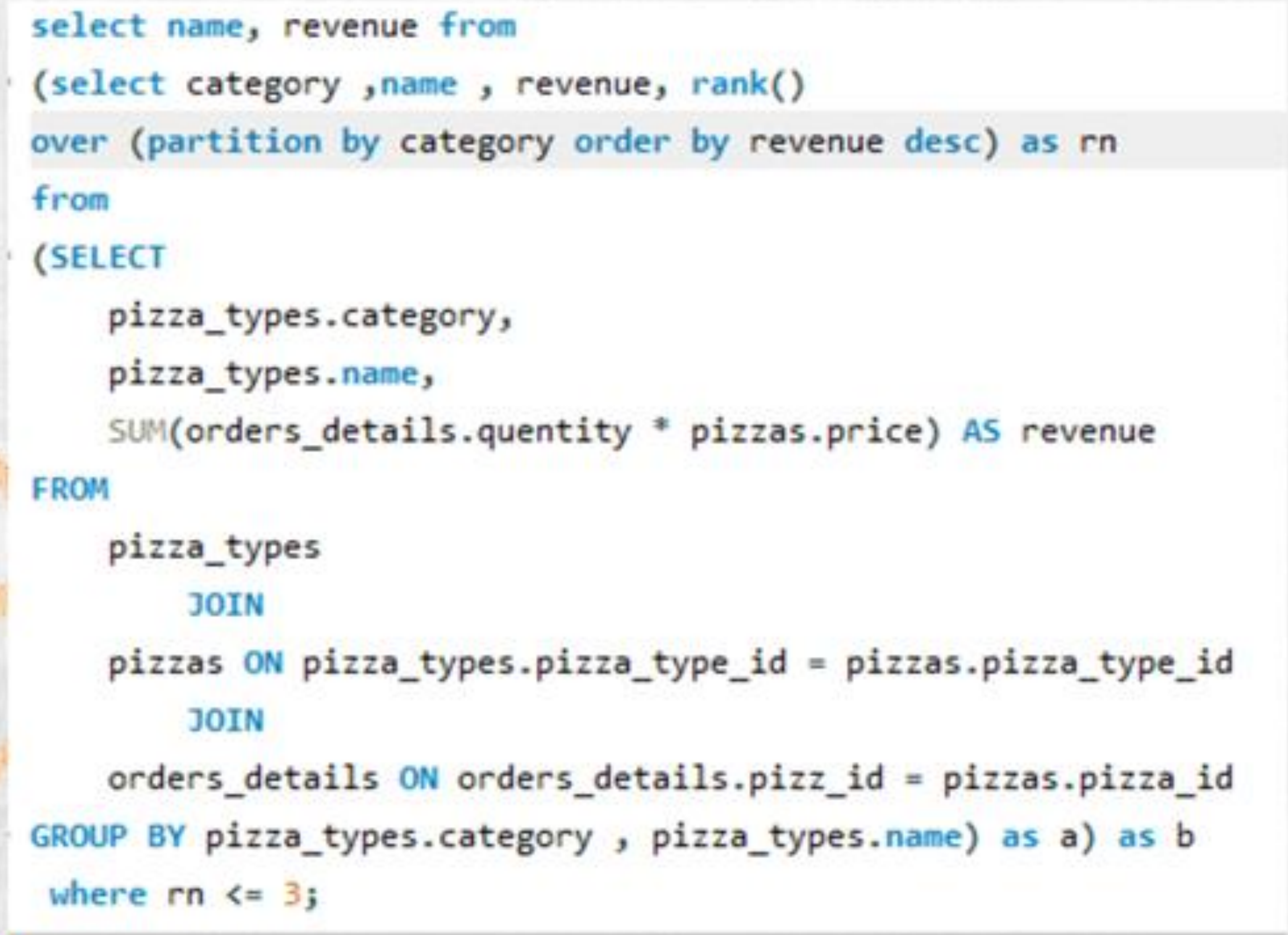


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5

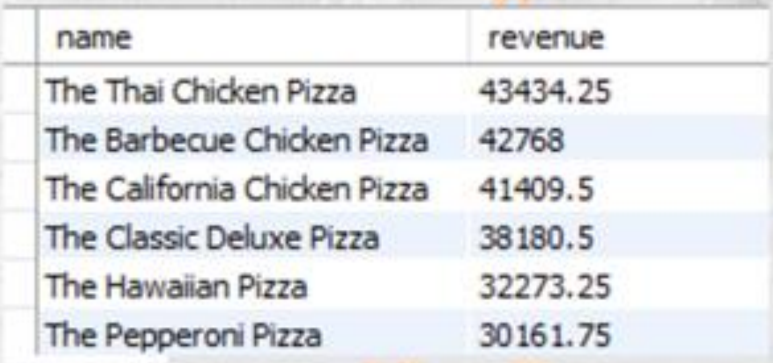
< BACK

NEXT >

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(orders_details.quentity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizz_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

< BACK

NEXT >

THANK YOU!

< BACK