Where Every Slice is a Taste of Perfection

PIZZA HUT SALES ANALYSIS USING SQL

SLICE & STATS

HELLO!
MY NAME IS
JAKESH NANDURKAR

In this project, I've applied my SQL skills to analyze Pizza Hut sales data. Using structured queries, I explored key sales insights, answered business-related questions, and extracted actionable data-driven conclusions

This project helped me strengthen my expertise in:

- SQL joins and aggregations
- Window functions
- Real-world business data analysis

BASIC INSIGHTS FROM PIZZA SALES DATA

✅ Key Findings:

TOTAL REVENUE: $817860.05

Total Orders Placed: 1,000

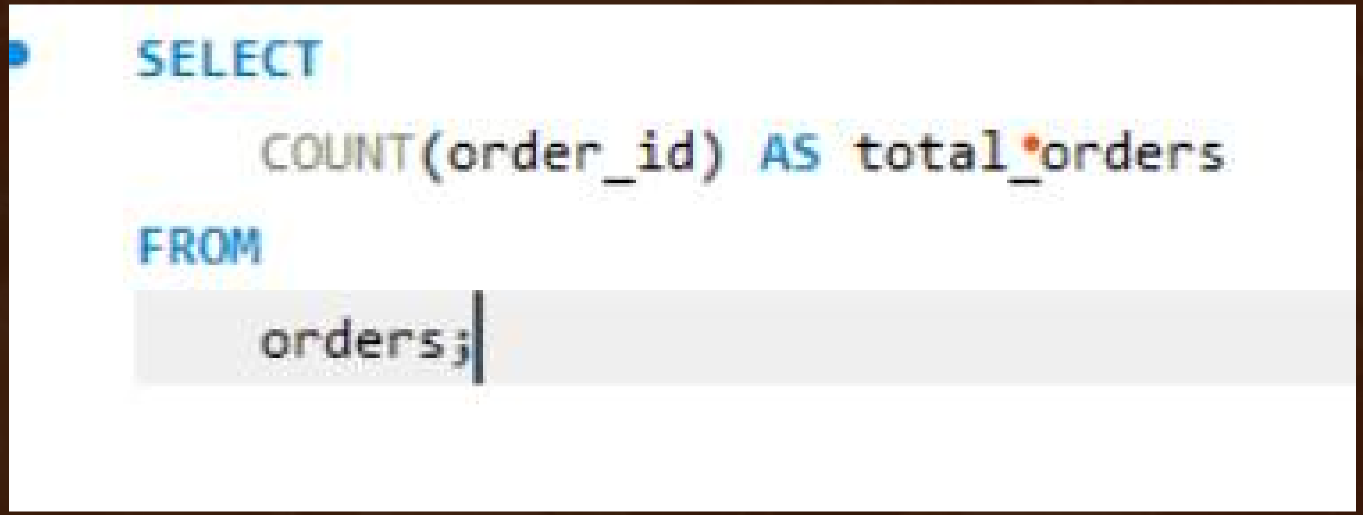
Highest Priced Pizza: The Greek Pizza (XXL) –$35.95

Most Common Size Ordered: Medium (M)

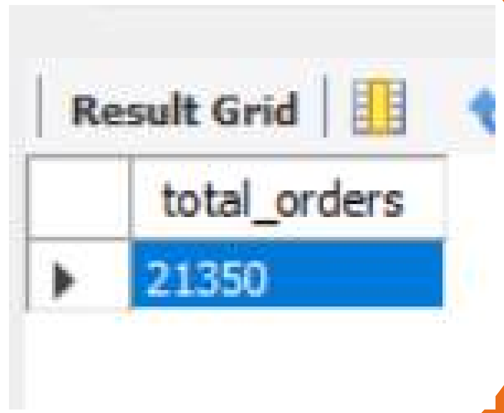
TOP 5 MOST ORDERED PIZZA TYPES:

1. Pepperoni Pizza
2. Classic Deluxe Pizza
3. The Greek Pizza
4. BBQ Chicken Pizza
5. Hawaiian Pizza

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



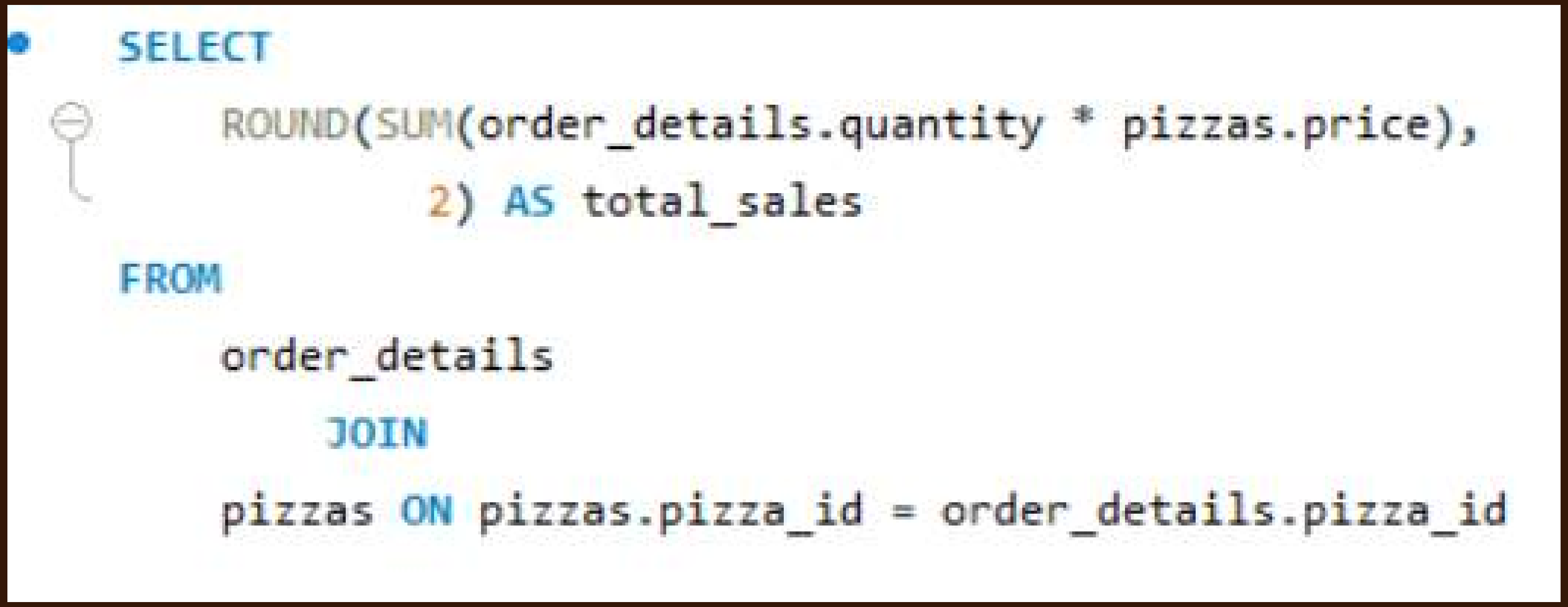
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



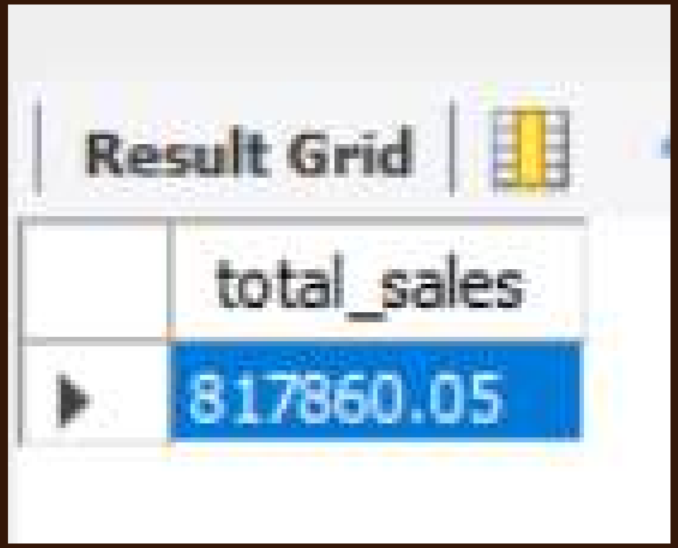
Result Grid

```
  total_orders
▶ 21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

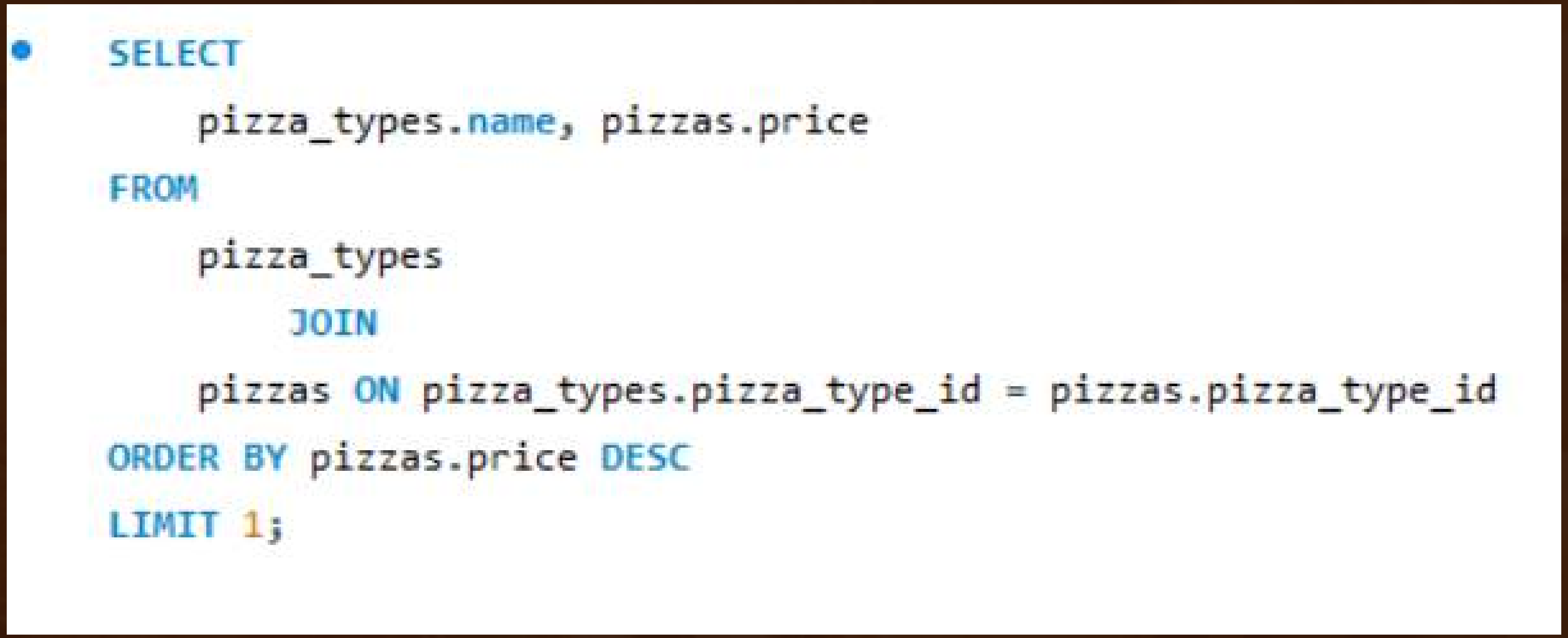


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

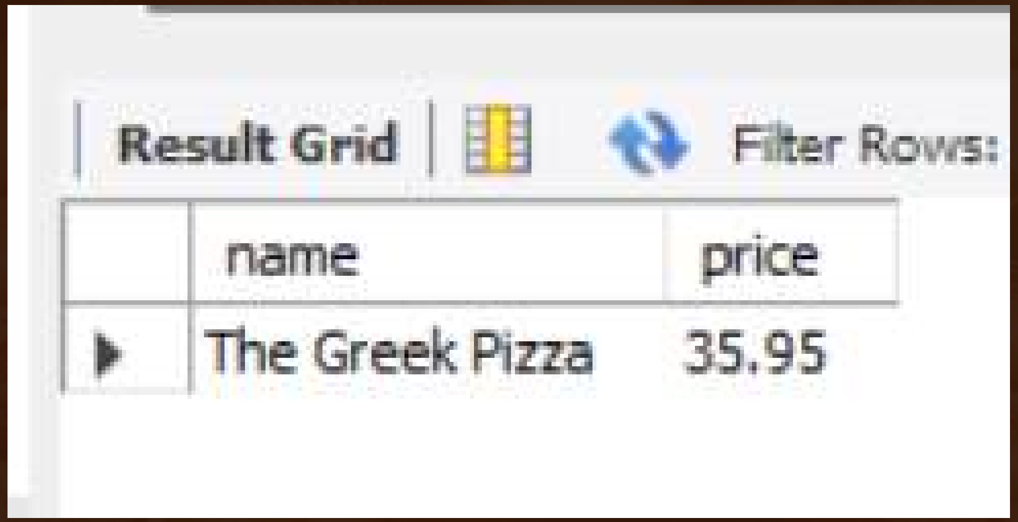


total_sales
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

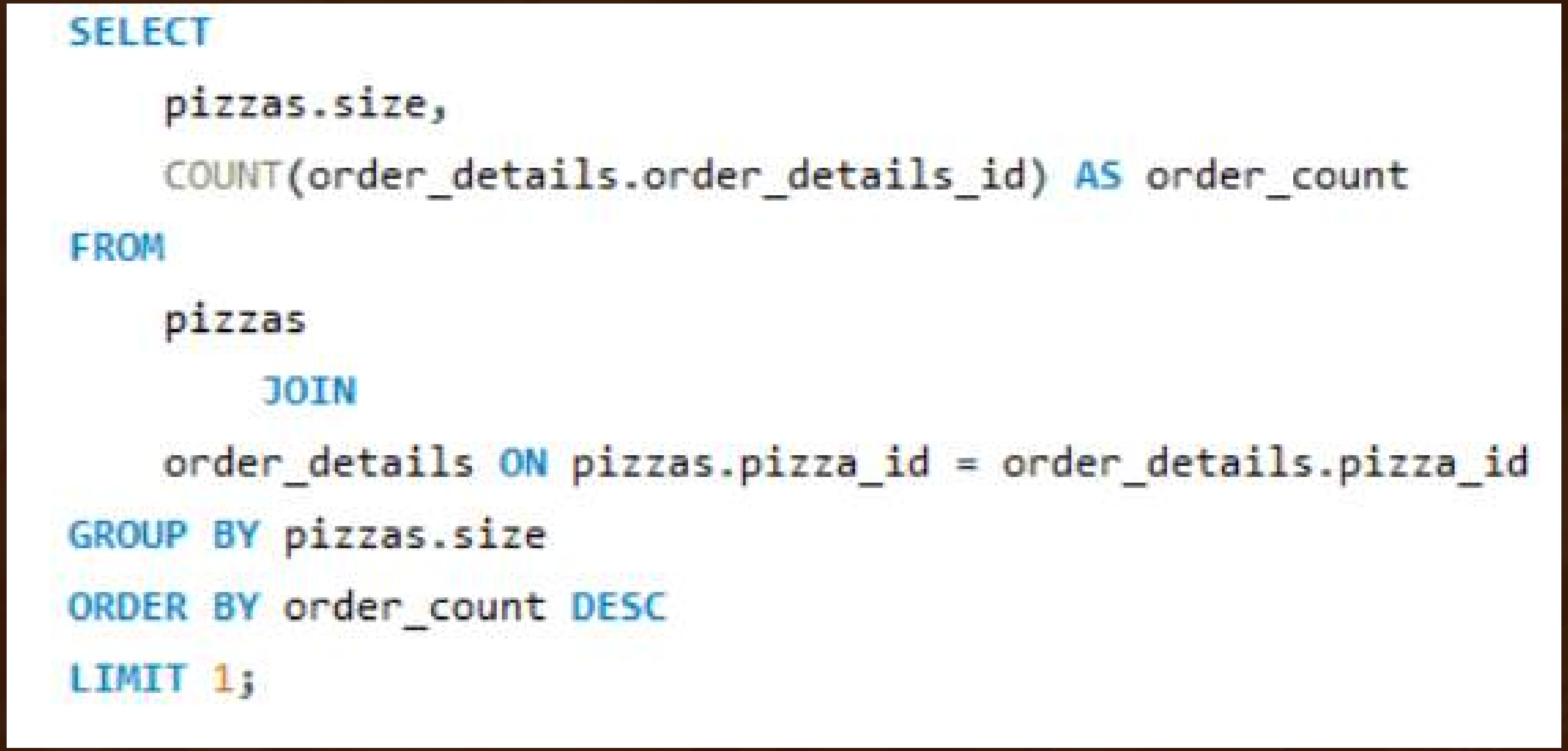


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

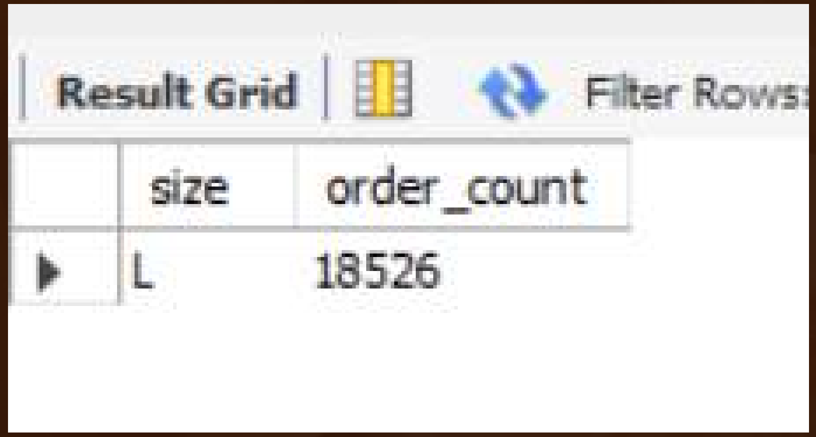


	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

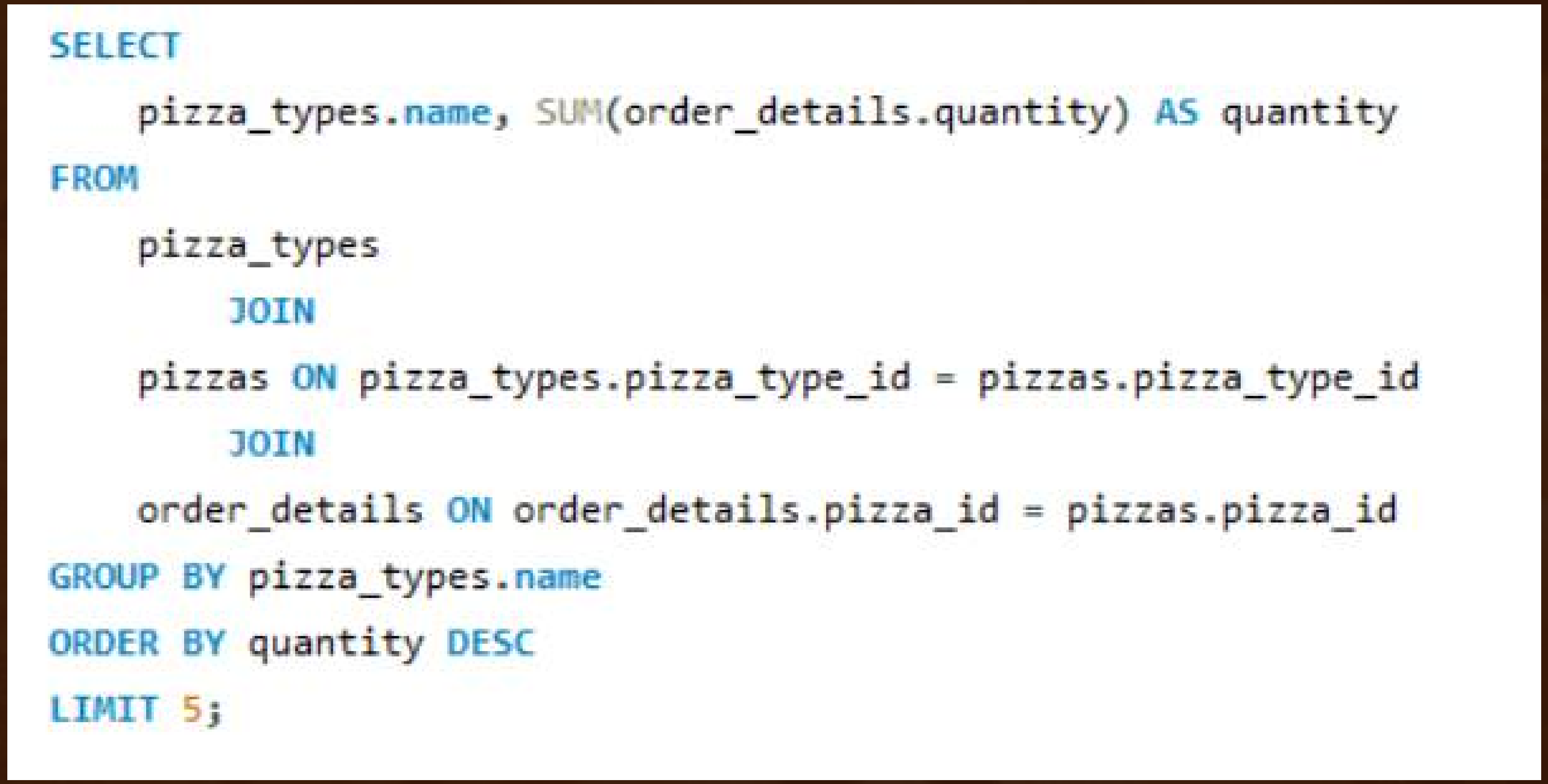


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

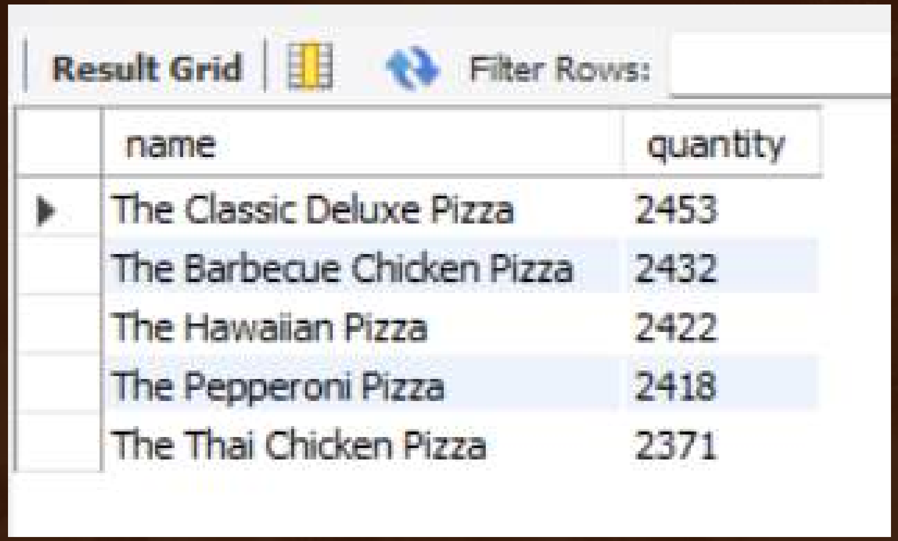


```
  size   order_count
▶ L      18526
```

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



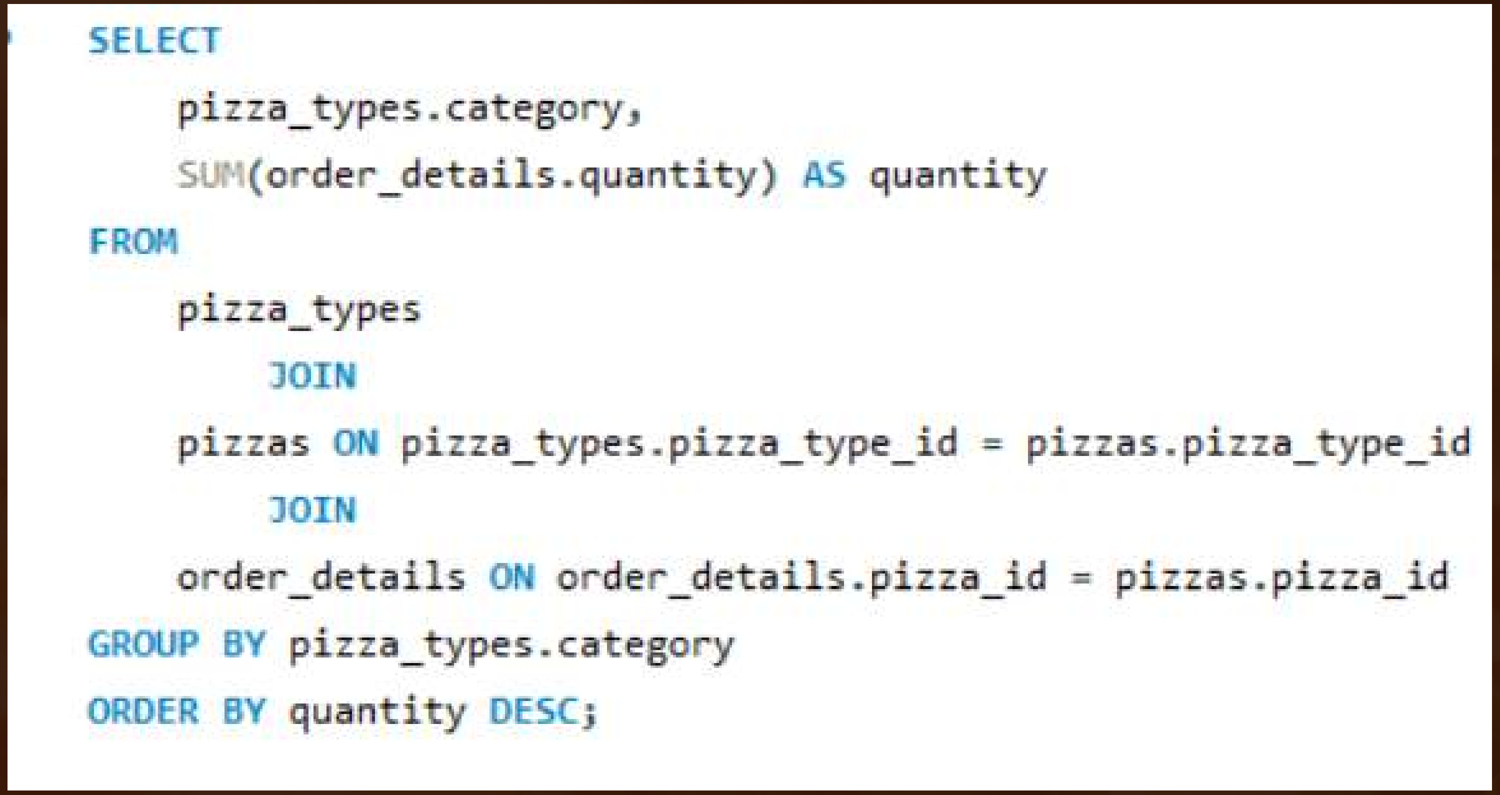
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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



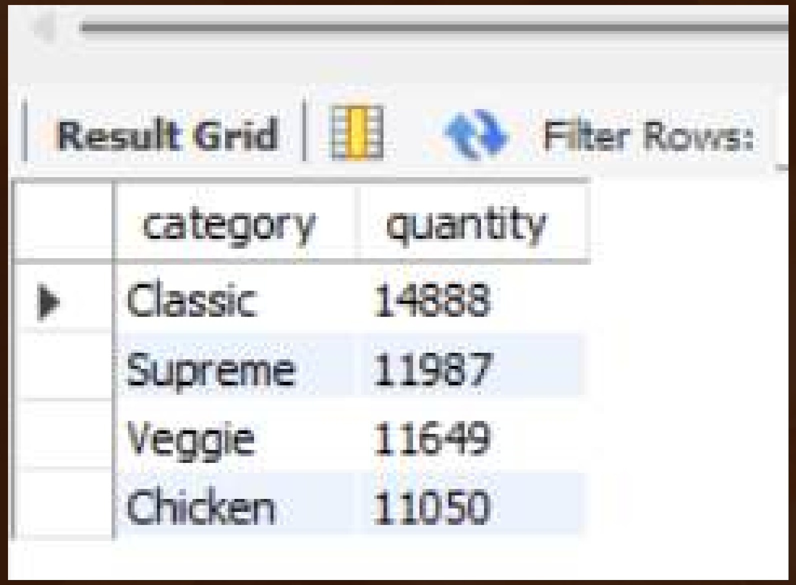
Result Grid | Filter Rows:

```
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

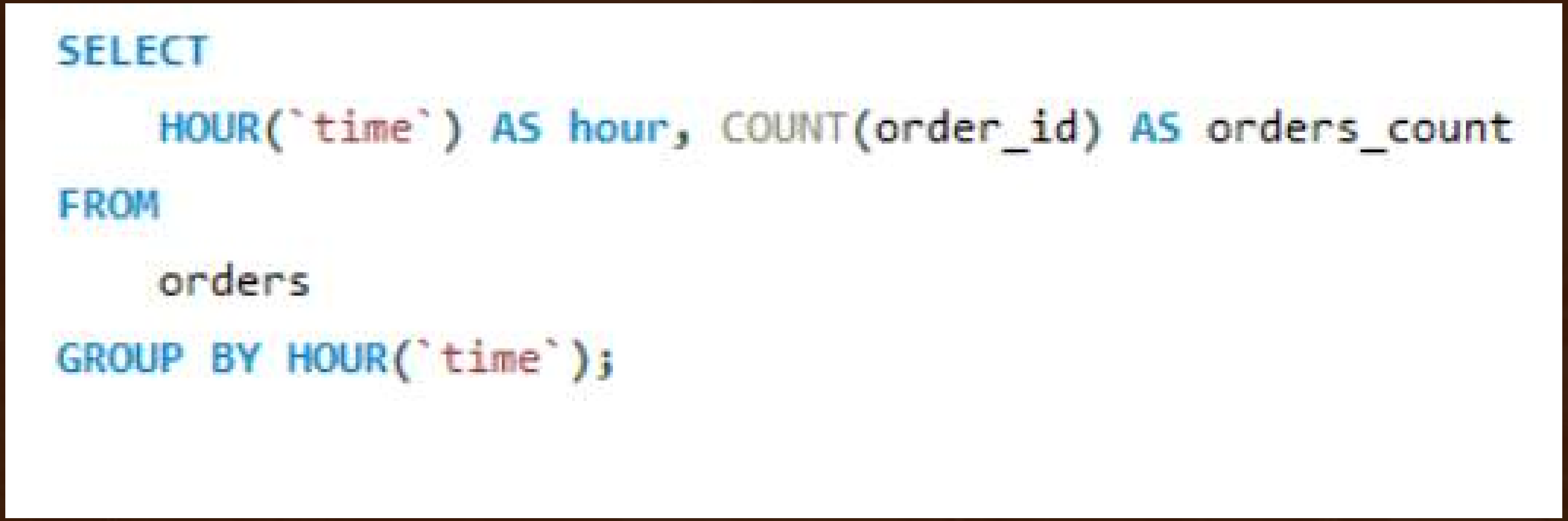


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

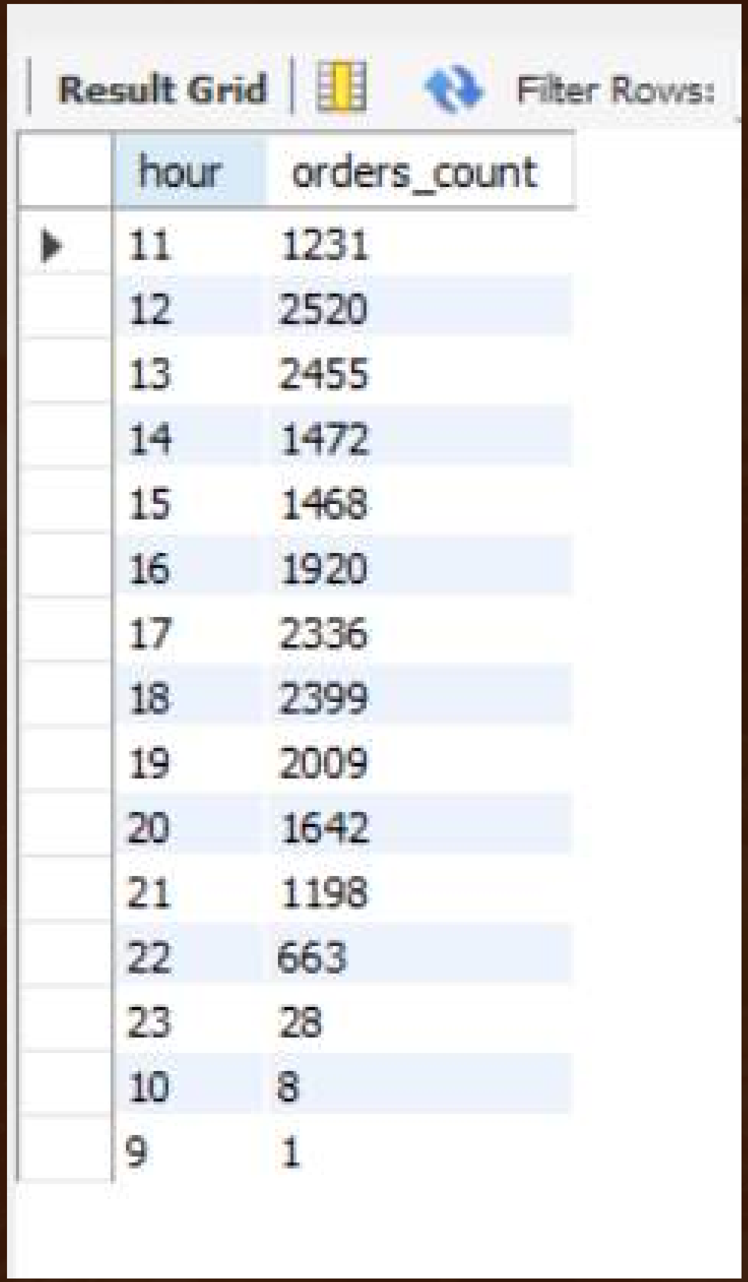


```
category   quantity
Classic    14888
Supreme    11987
Veggie     11649
Chicken    11050
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

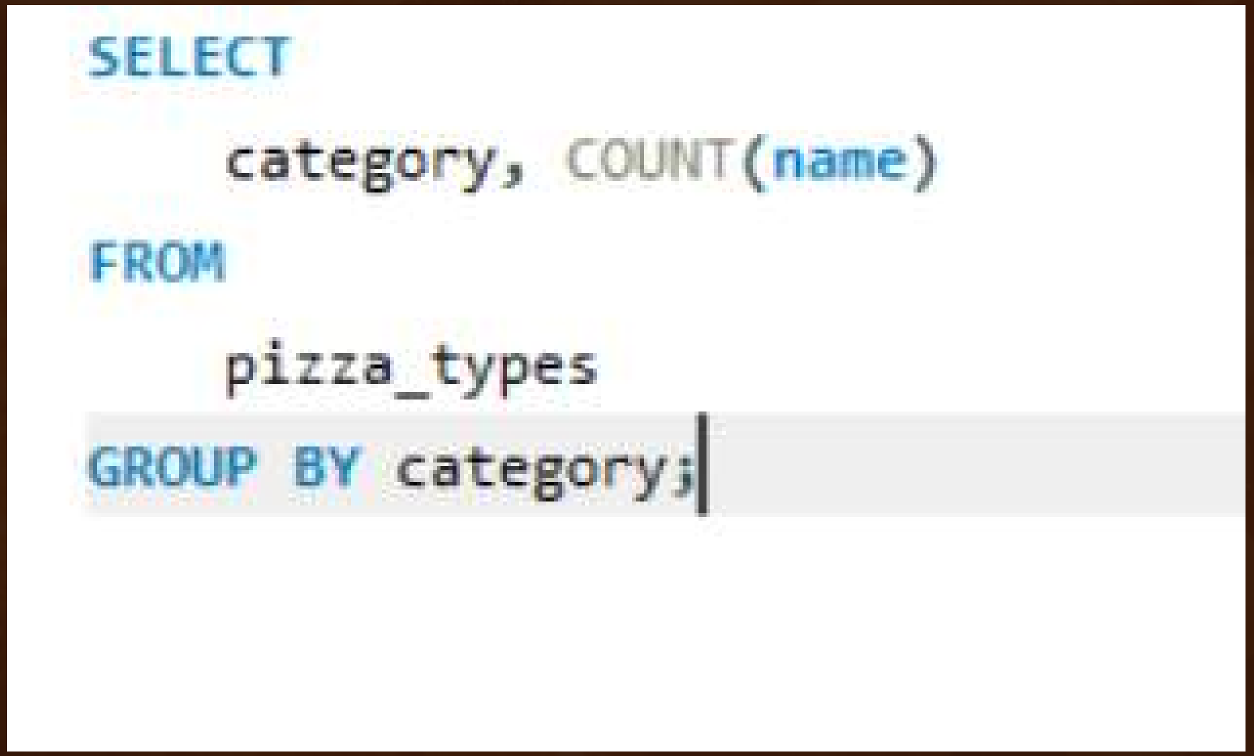


```
SELECT
    HOUR(`time`) AS hour, COUNT(order_id) AS orders_count
FROM
    orders
GROUP BY HOUR(`time`);
```

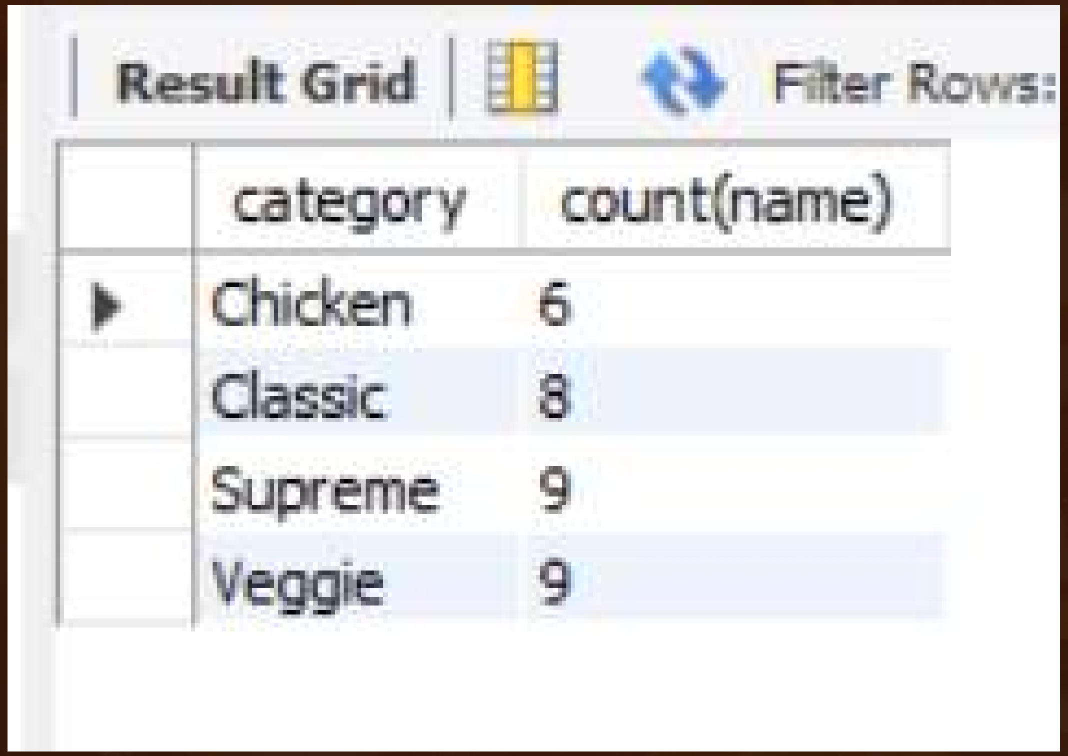


```
hour  orders_count
11    1231
12    2520
13    2455
14    1472
15    1468
16    1920
17    2336
18    2399
19    2009
20    1642
21    1198
22    663
23    28
10    8
9     1
```

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

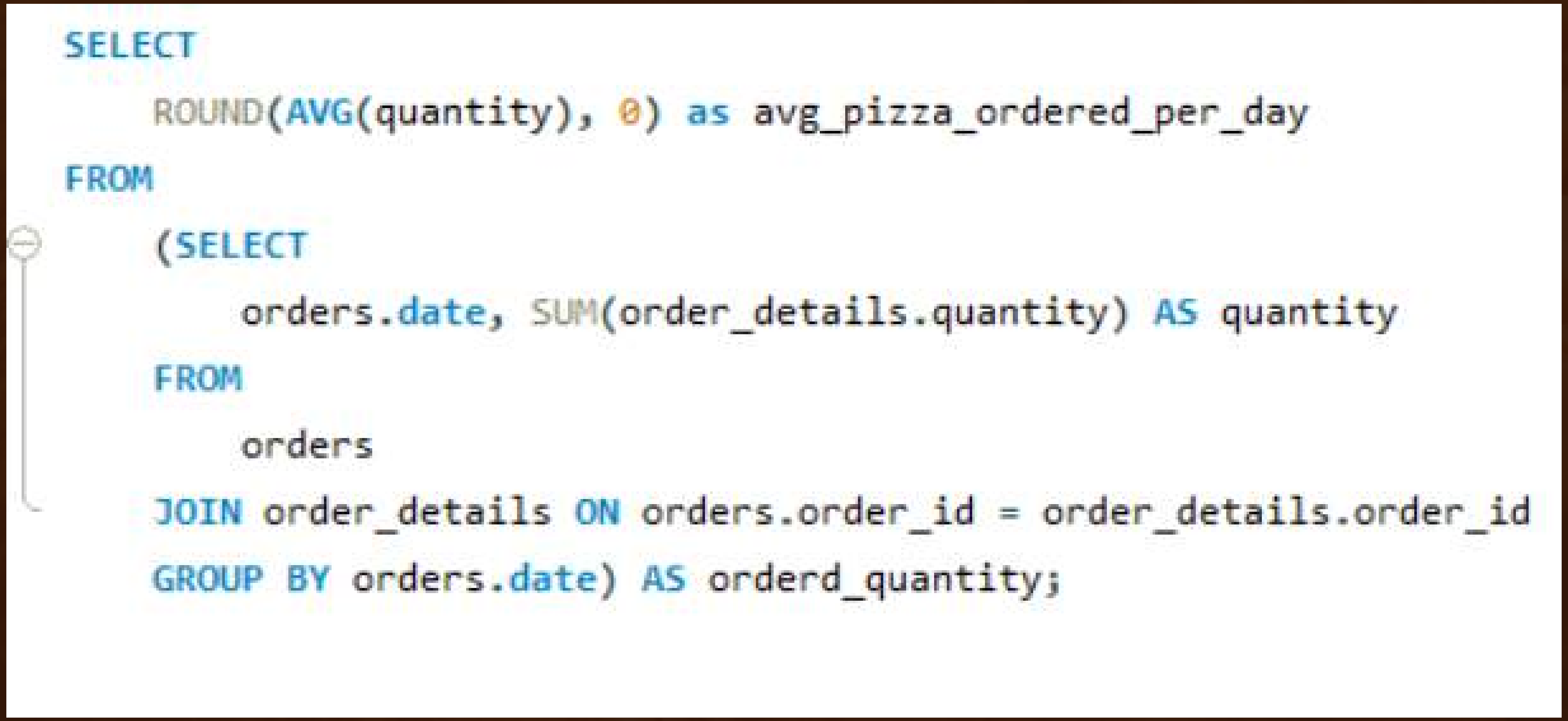


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

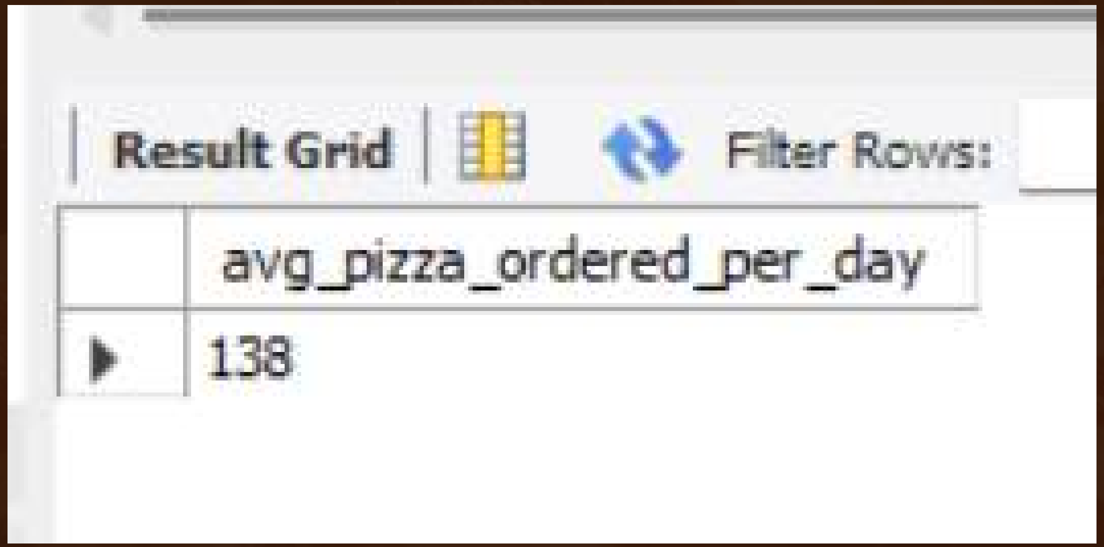


```
category   count(name)
Chicken    6
Classic    8
Supreme    9
Veggie     9
```

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

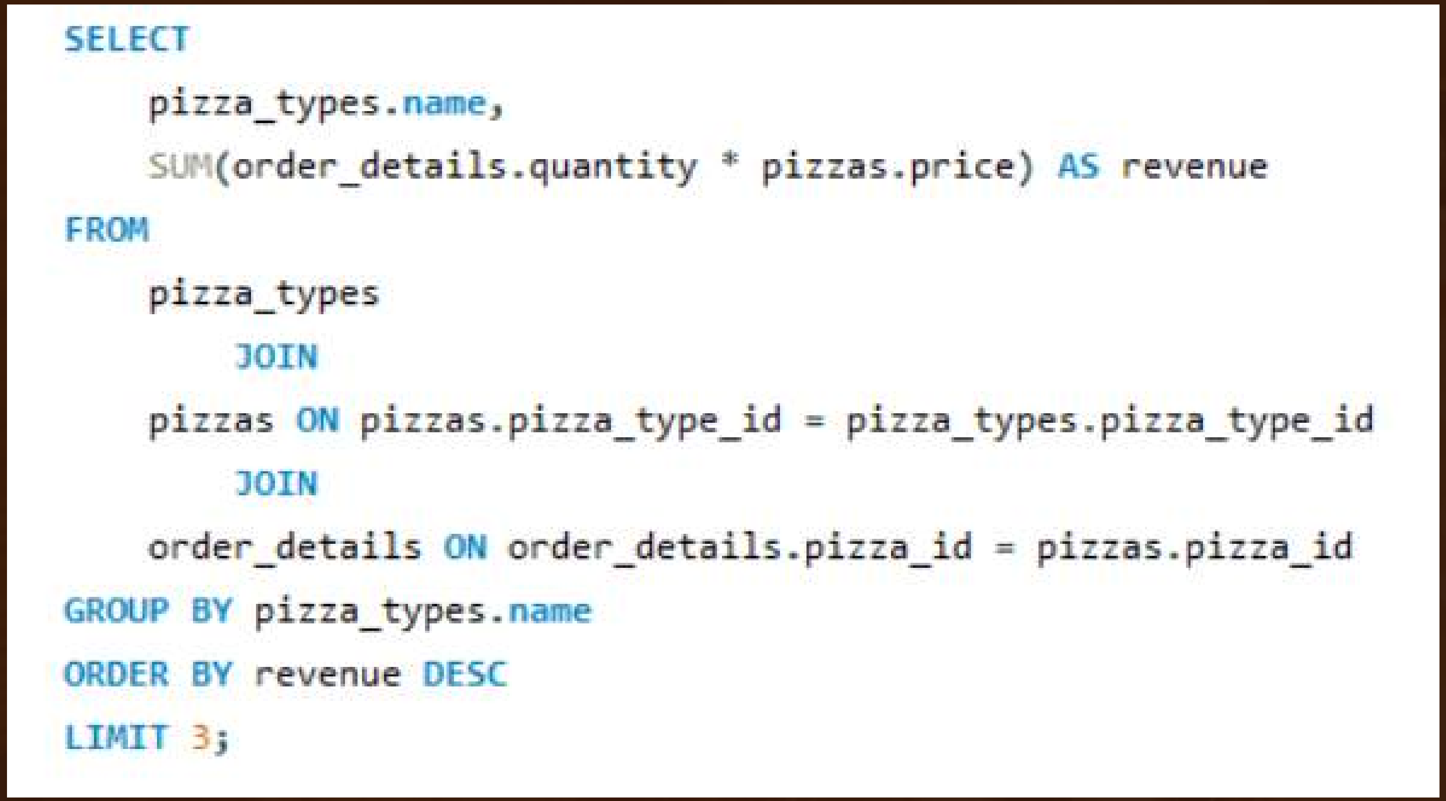


```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS orderd_quantity;
```

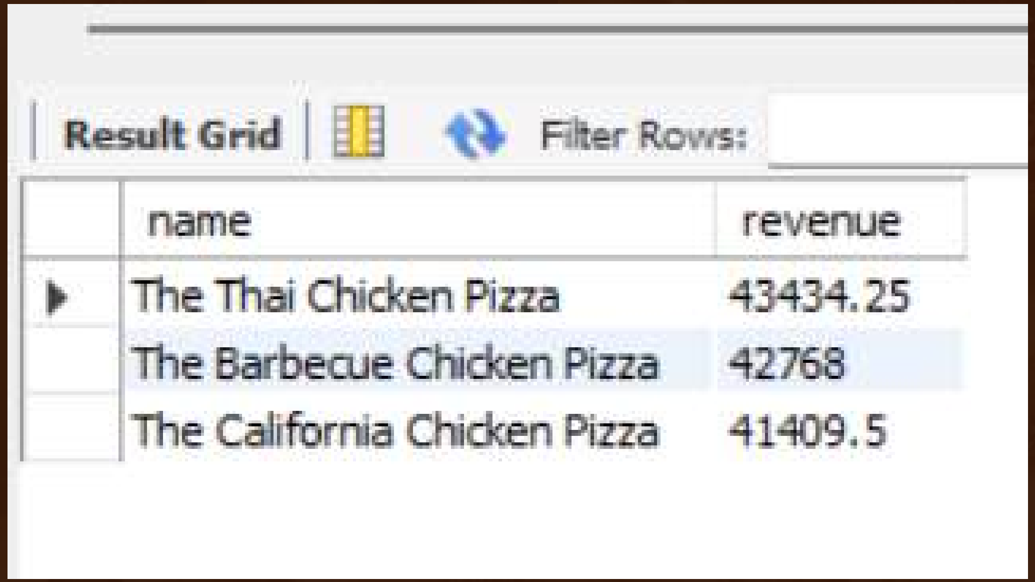


```
avg_pizza_ordered_per_day
138
```

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 pizzas.pizza_type_id = pizza_types.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;
```



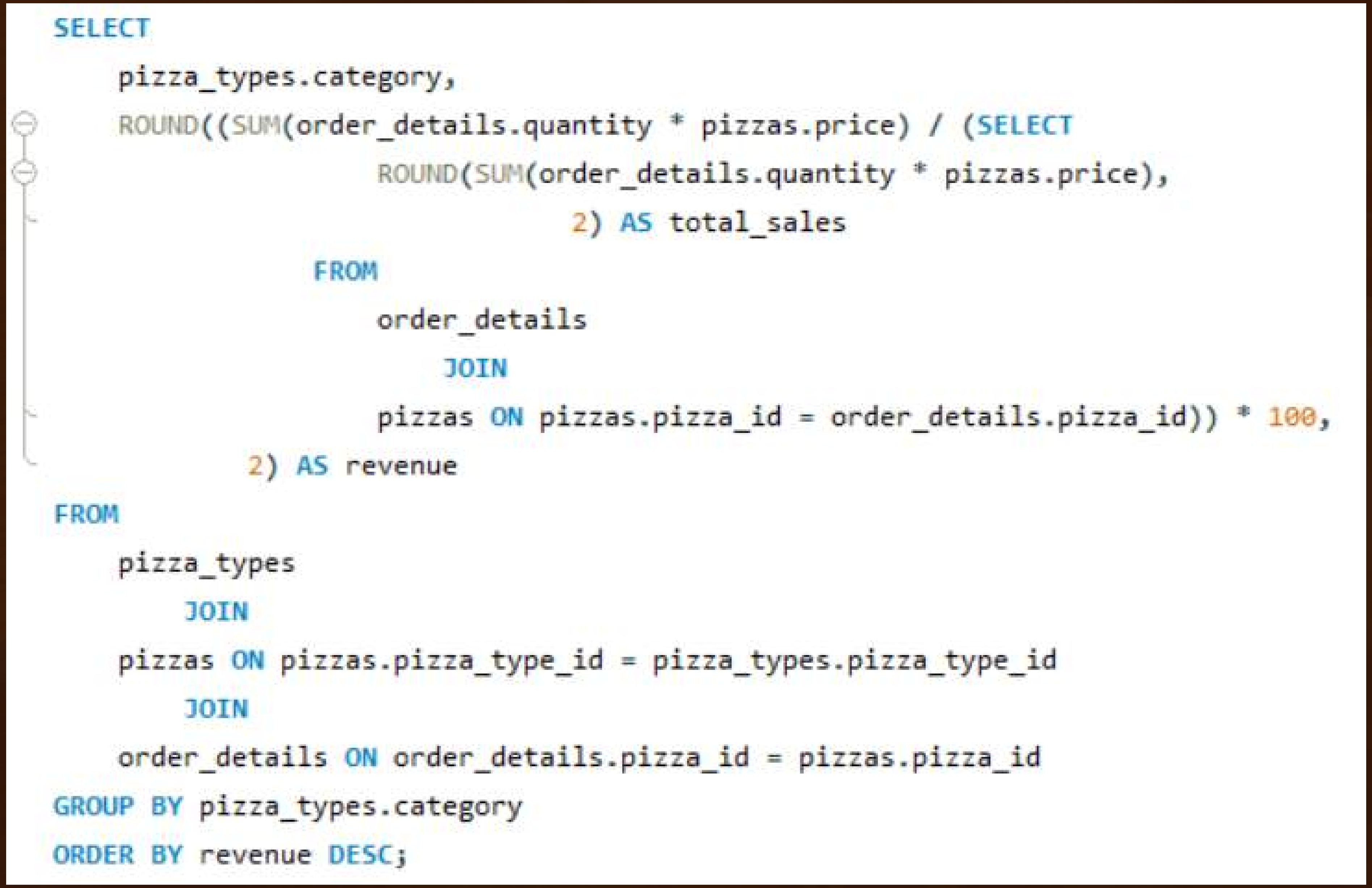
```
name                              revenue
The Thai Chicken Pizza            43434.25
The Barbecue Chicken Pizza        42768
The California Chicken Pizza      41409.5
```

1

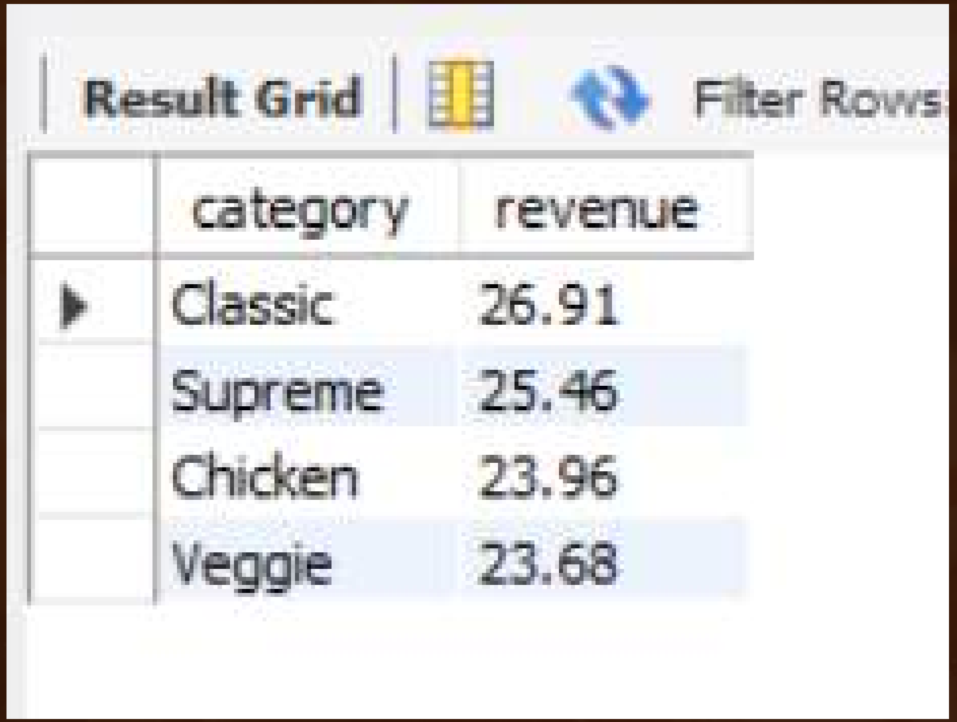
2

3

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

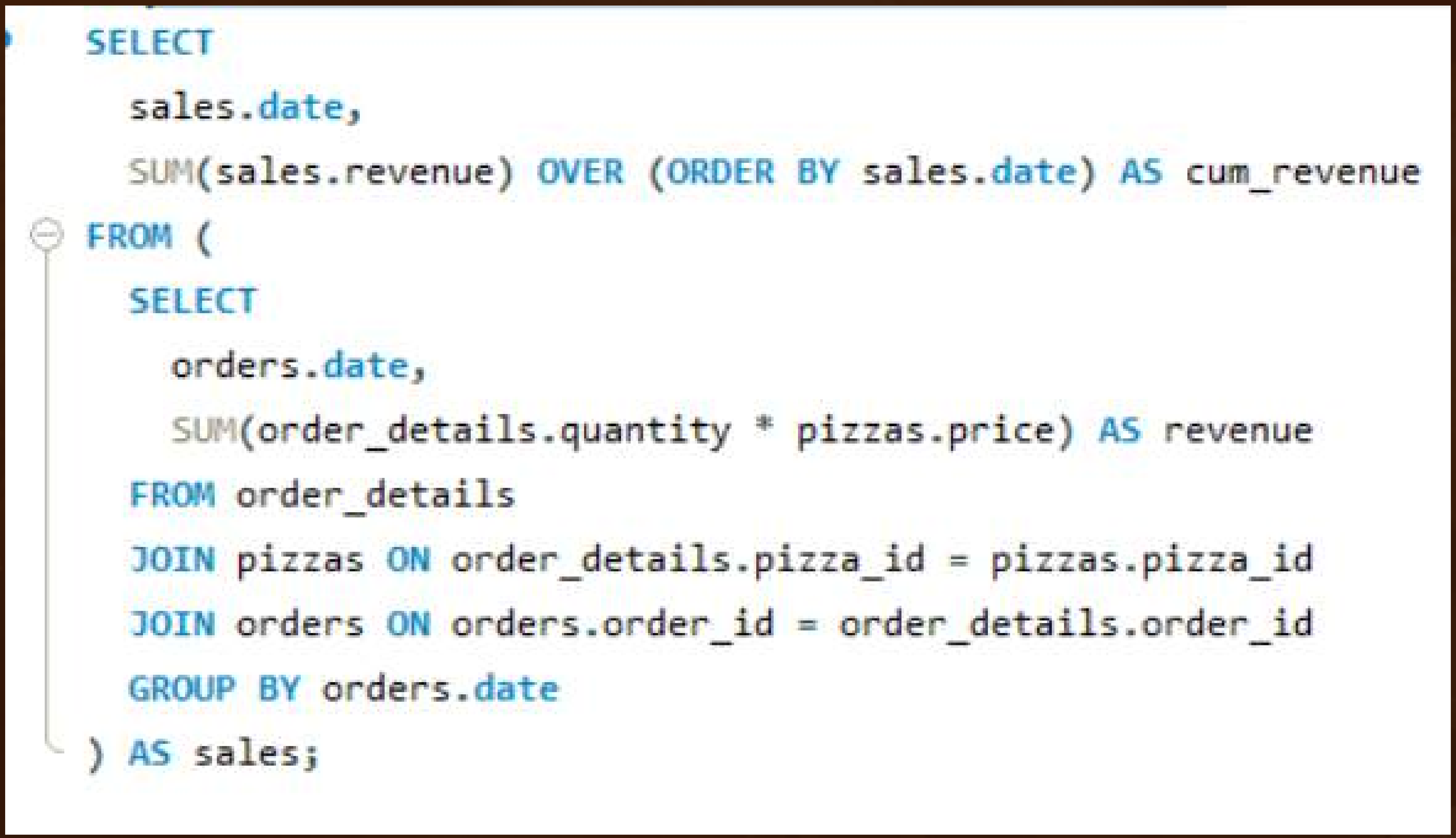


```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
            FROM
                order_details
                    JOIN
                pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

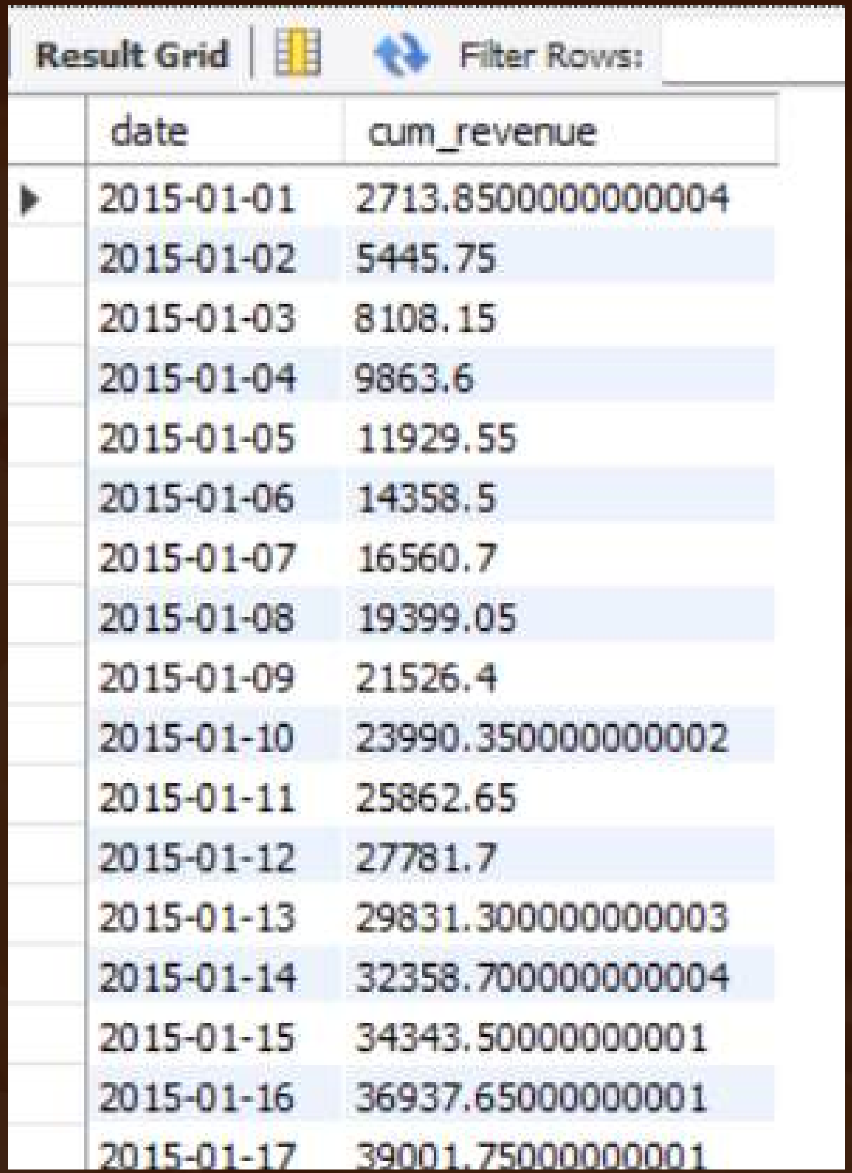


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

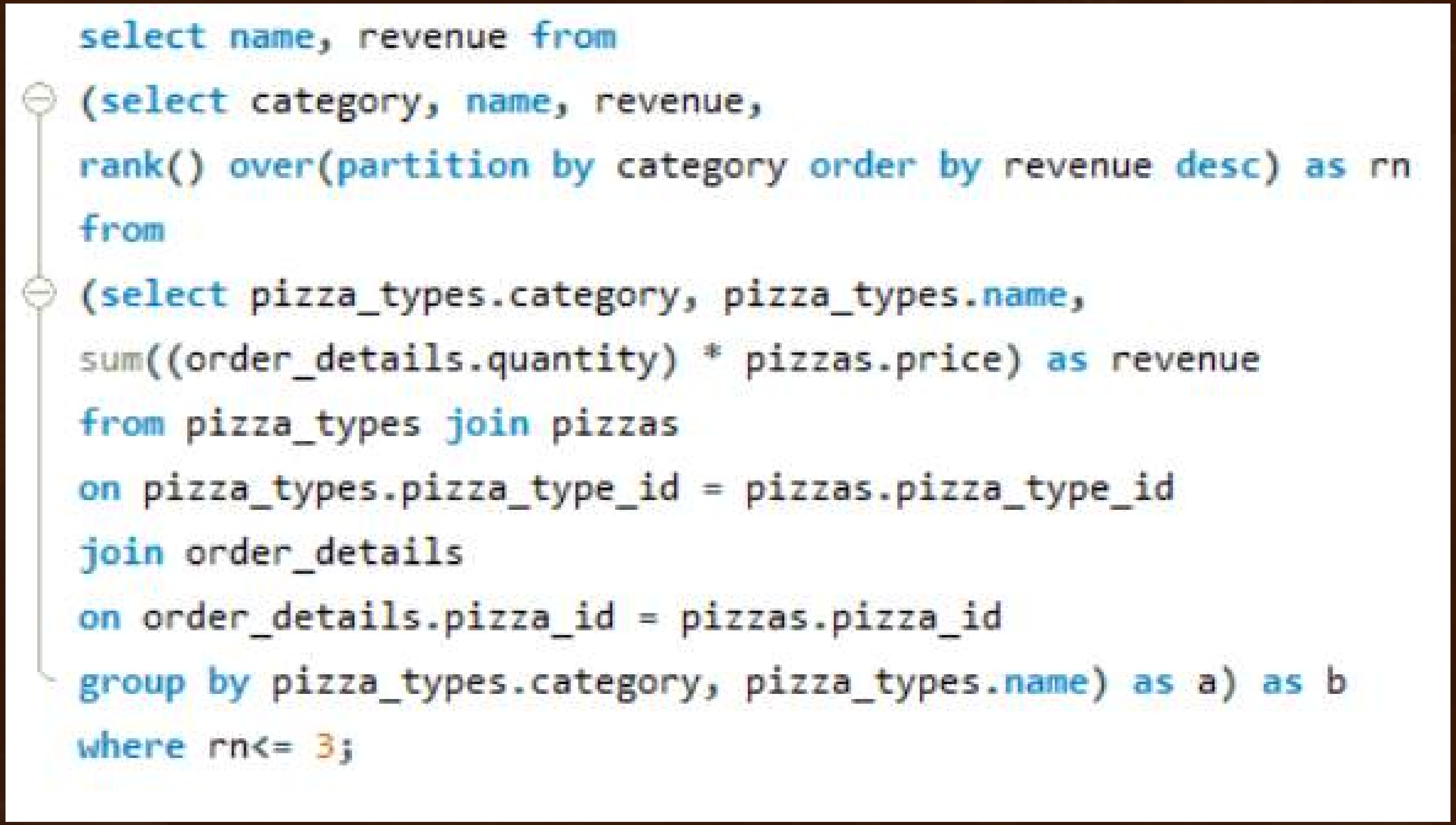


```
SELECT
  sales.date,
  SUM(sales.revenue) OVER (ORDER BY sales.date) AS cum_revenue
FROM (
  SELECT
    orders.date,
    SUM(order_details.quantity * pizzas.price) AS revenue
  FROM order_details
  JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
  JOIN orders ON orders.order_id = order_details.order_id
  GROUP BY orders.date
) AS sales;
```

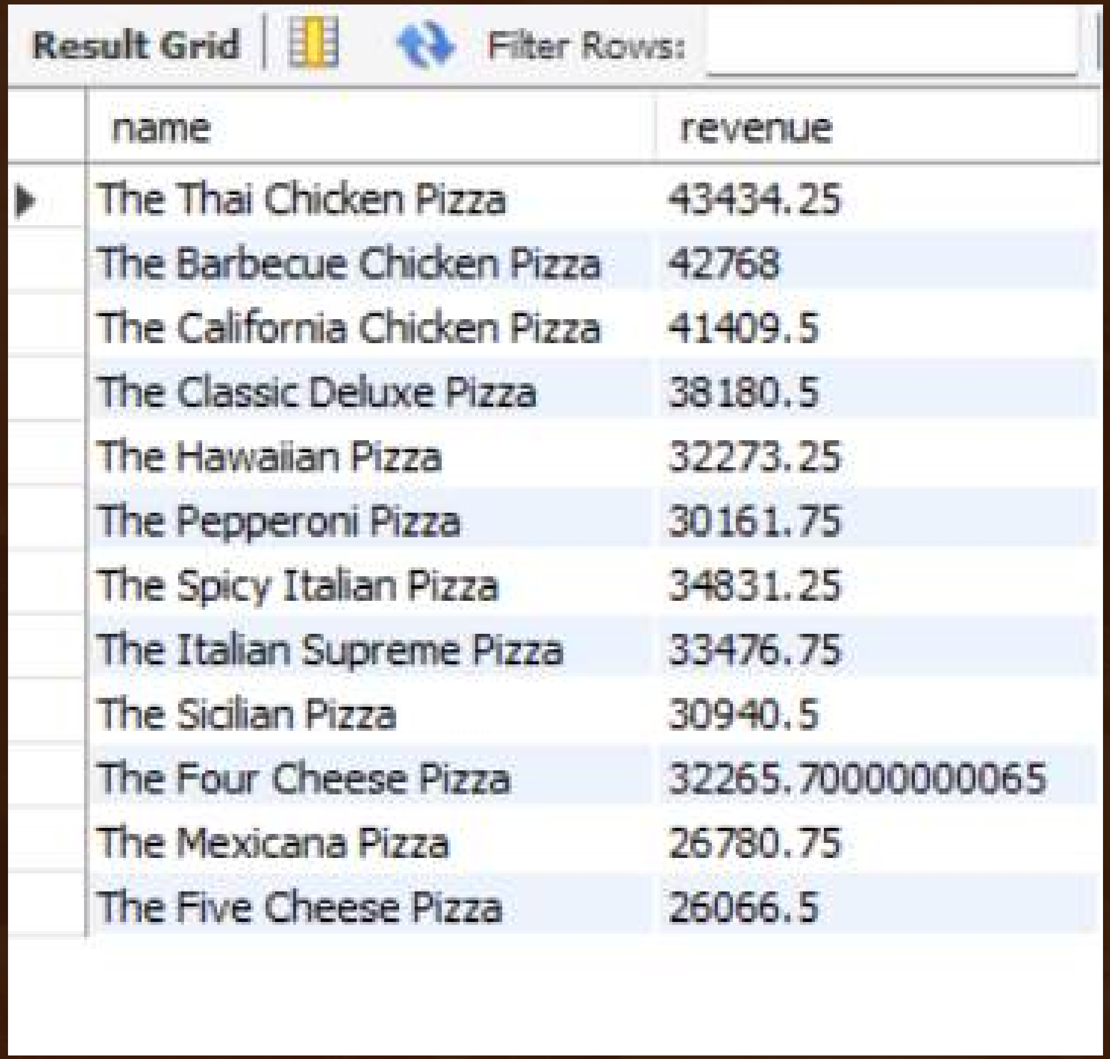


```
date        cum_revenue
2015-01-01  2713.8500000000004
2015-01-02  5445.75
2015-01-03  8108.15
2015-01-04  9863.6
2015-01-05  11929.55
2015-01-06  14358.5
2015-01-07  16560.7
2015-01-08  19399.05
2015-01-09  21526.4
2015-01-10  23990.350000000002
2015-01-11  25862.65
2015-01-12  27781.7
2015-01-13  29831.300000000003
2015-01-14  32358.700000000004
2015-01-15  34343.50000000001
2015-01-16  36937.65000000001
2015-01-17  39001.75000000001
```

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