FRESH PIZZAHUB

FRESH PIZZAHUB

FRESH PIZZAHUB

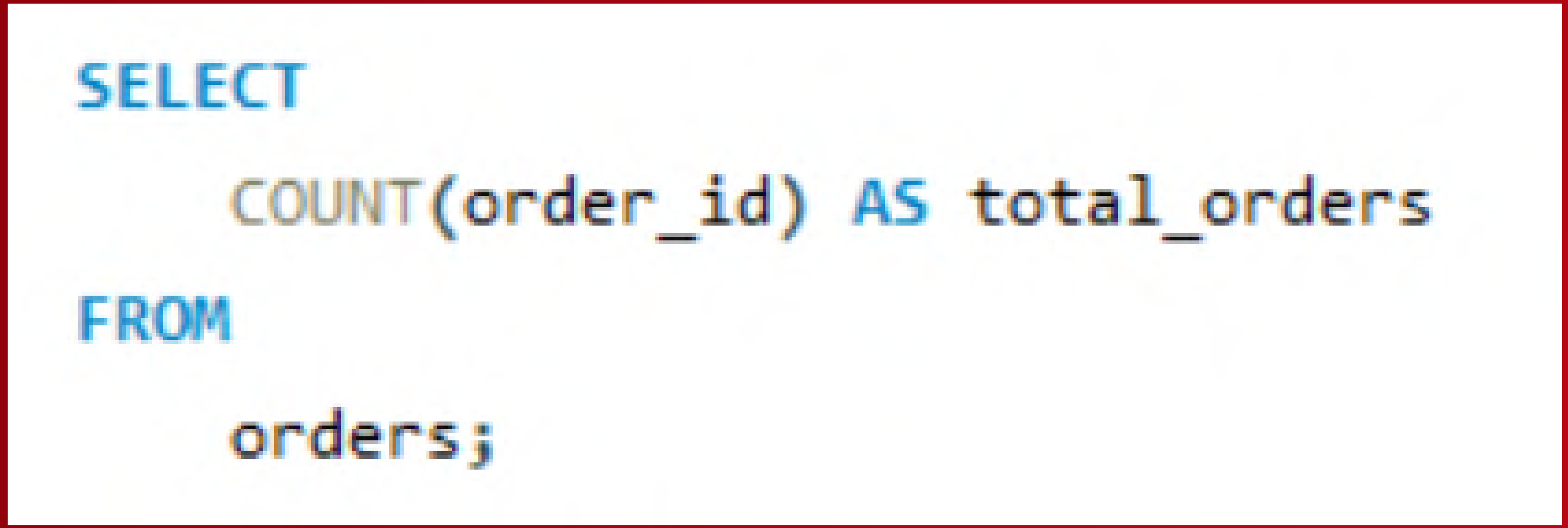
INTRODUCTION

"In a world where flavors delight and data empowers, FRESH PIZZAHUB combines culinary excellence with analytical precision! This project dives deep into the heart of our pizza operations, where MySQL isn't just a database – it's the backbone of smarter decisions and exceptional customer experiences.

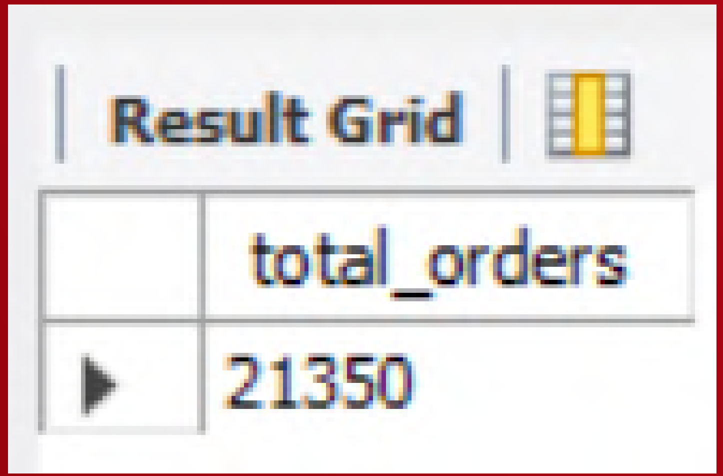
From analyzing top-selling pizzas to understanding customer preferences, we've cooked up valuable insights to optimize every slice served. With data as our secret recipe, we uncover trends, boost efficiency, and ensure every decision adds a sprinkle of success.

Get ready to explore how queries fuel innovation and every result drives us closer to perfection. Welcome to Fresh PizzaHub, where data meets deliciousness!"

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

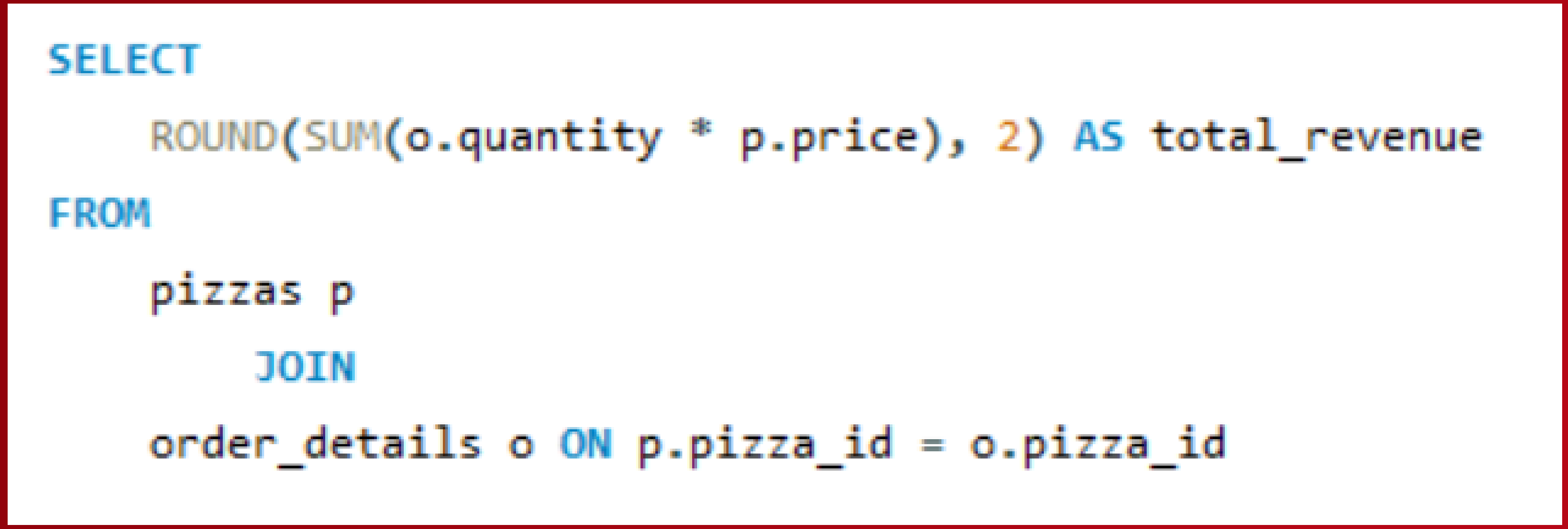


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

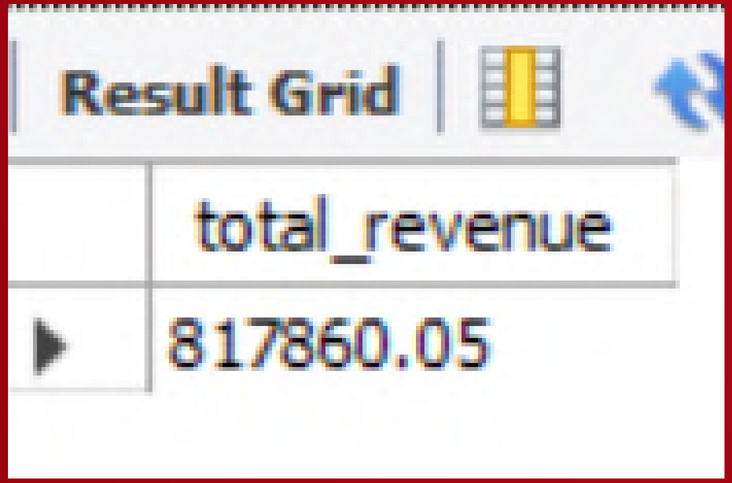


total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

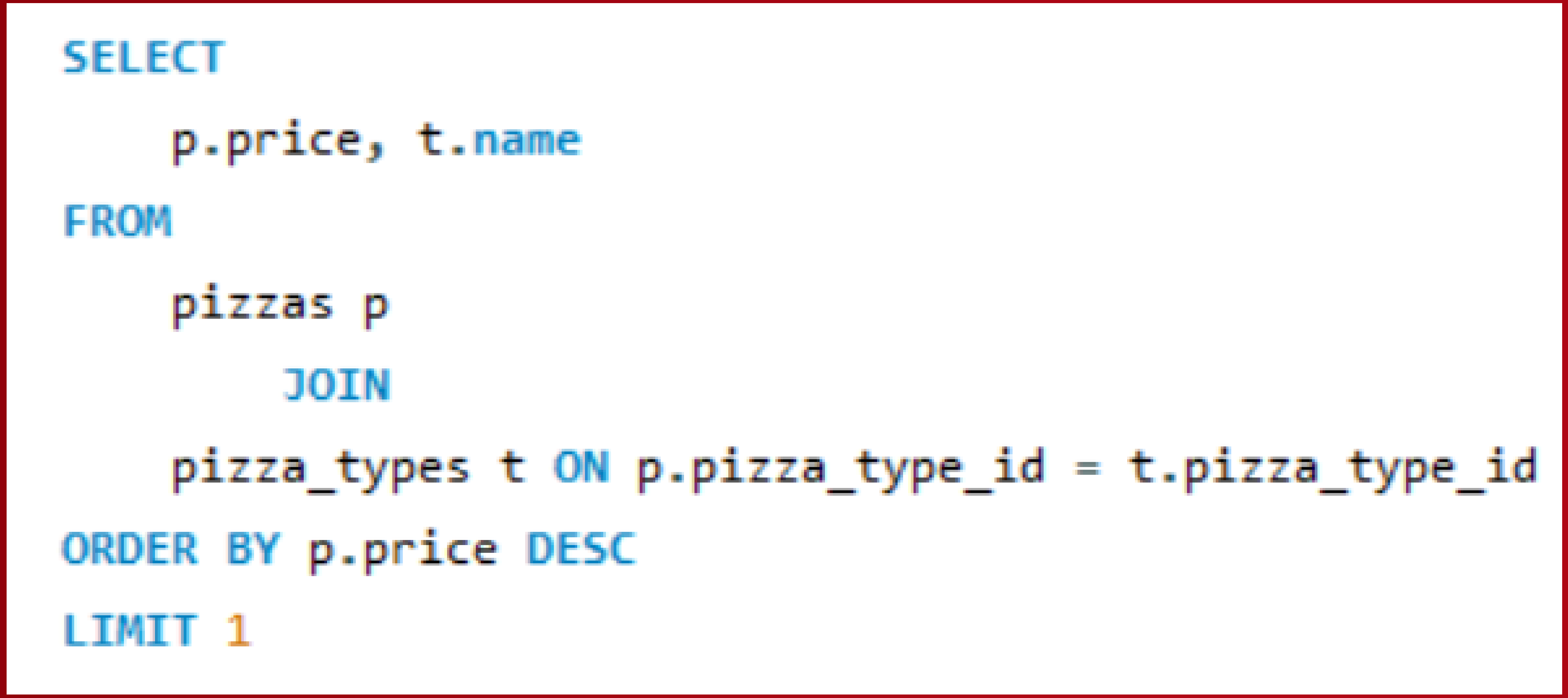


```
SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS total_revenue
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
```

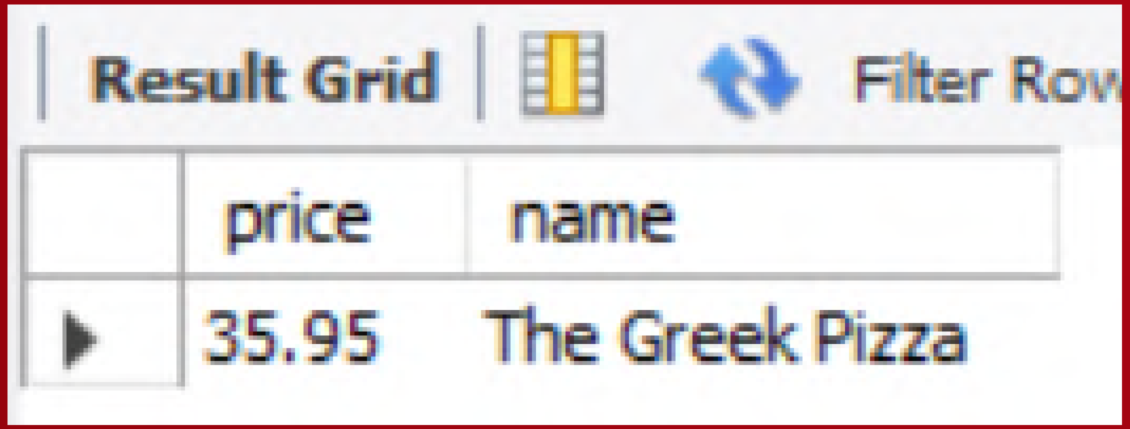


```
total_revenue
▶ 817860.05
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

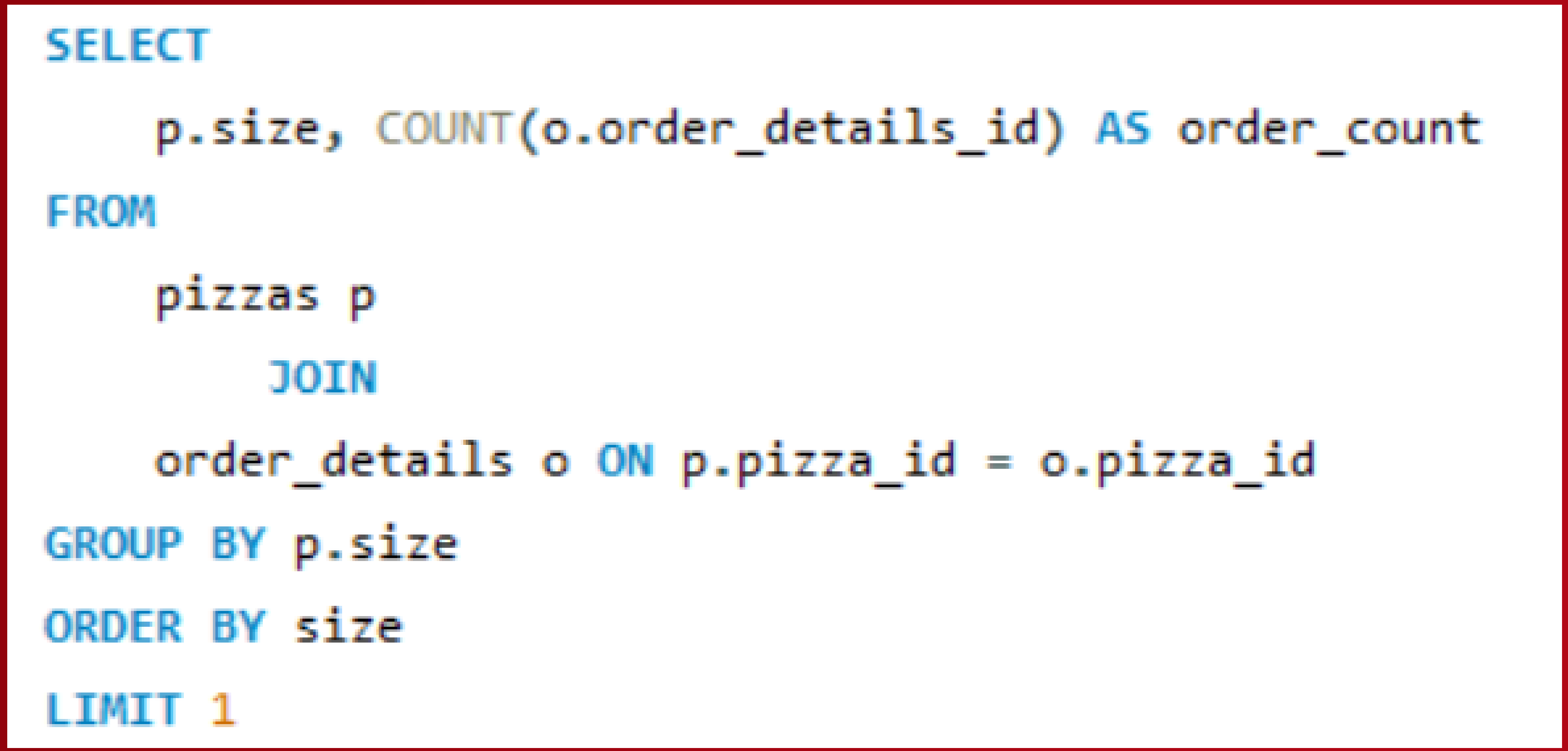


```
SELECT
    p.price, t.name
FROM
    pizzas p
        JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
ORDER BY p.price DESC
LIMIT 1
```

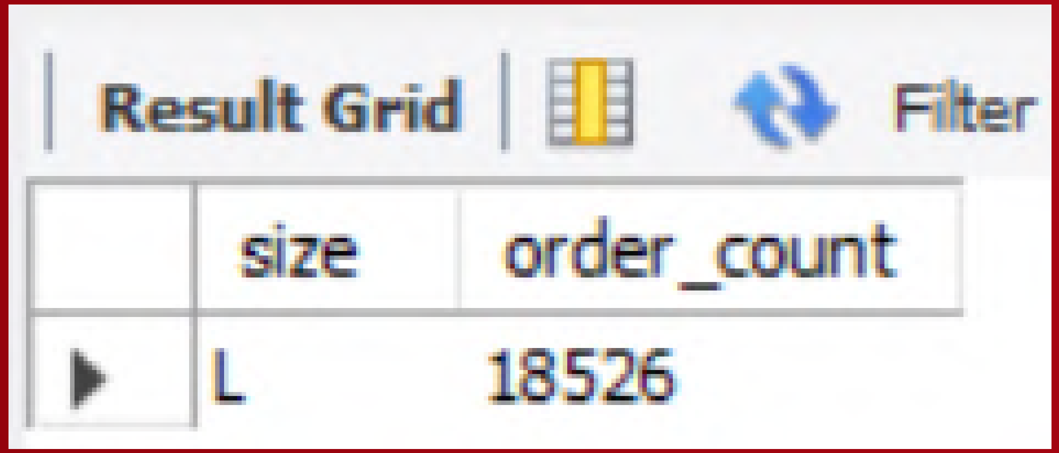


```
  price   name
▶ 35.95   The Greek Pizza
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

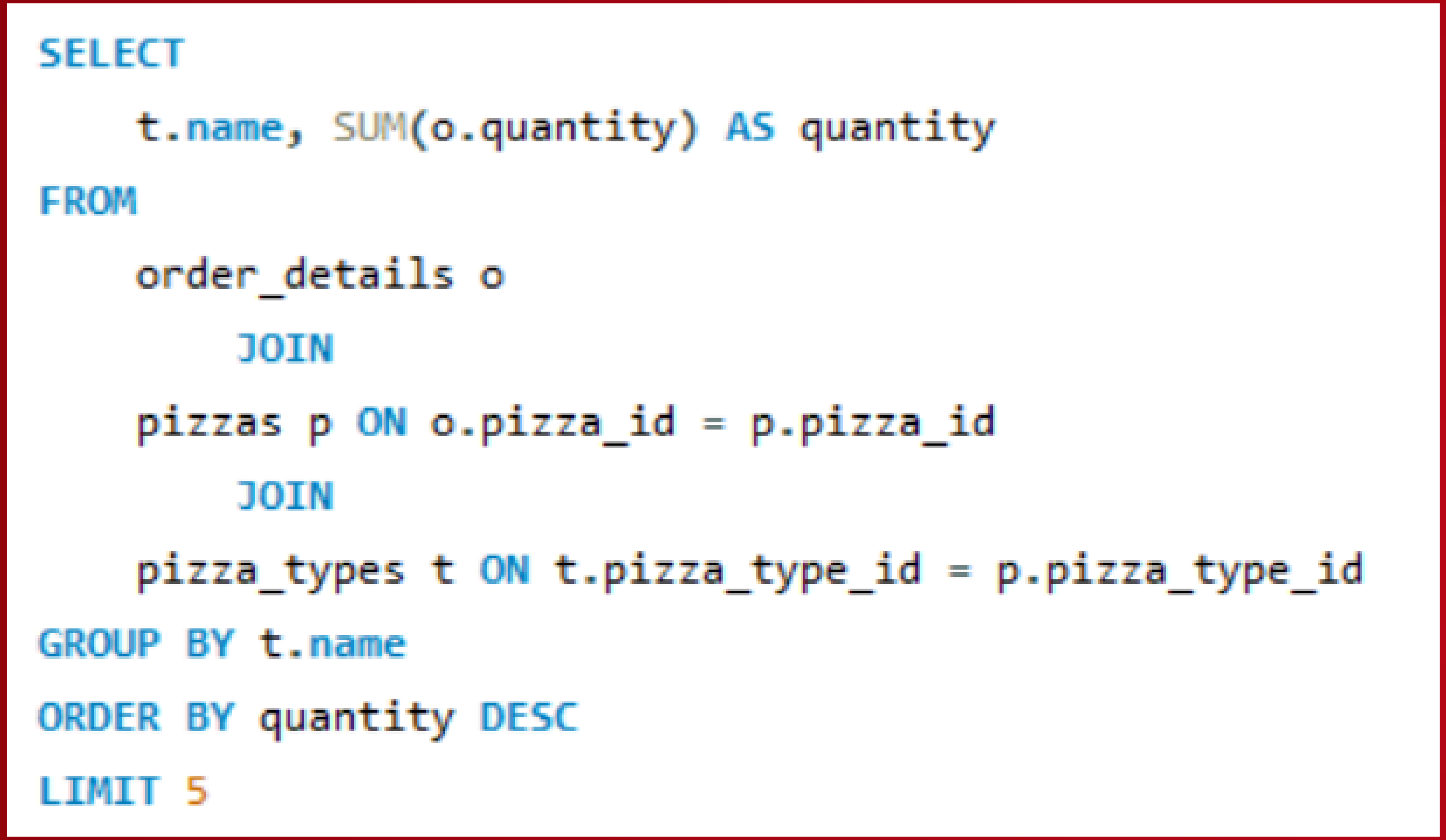


```
SELECT
    p.size, COUNT(o.order_details_id) AS order_count
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY p.size
ORDER BY size
LIMIT 1
```

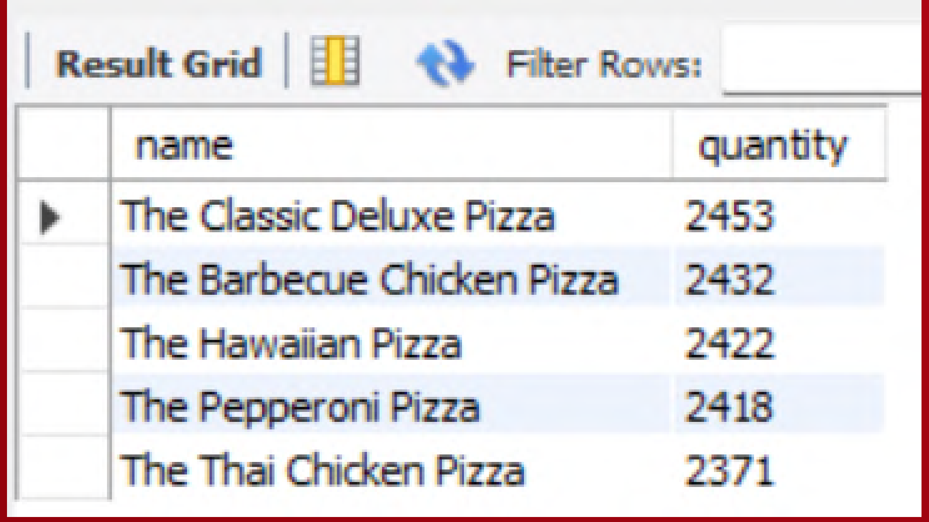


	size	order_count
▶	L	18526

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

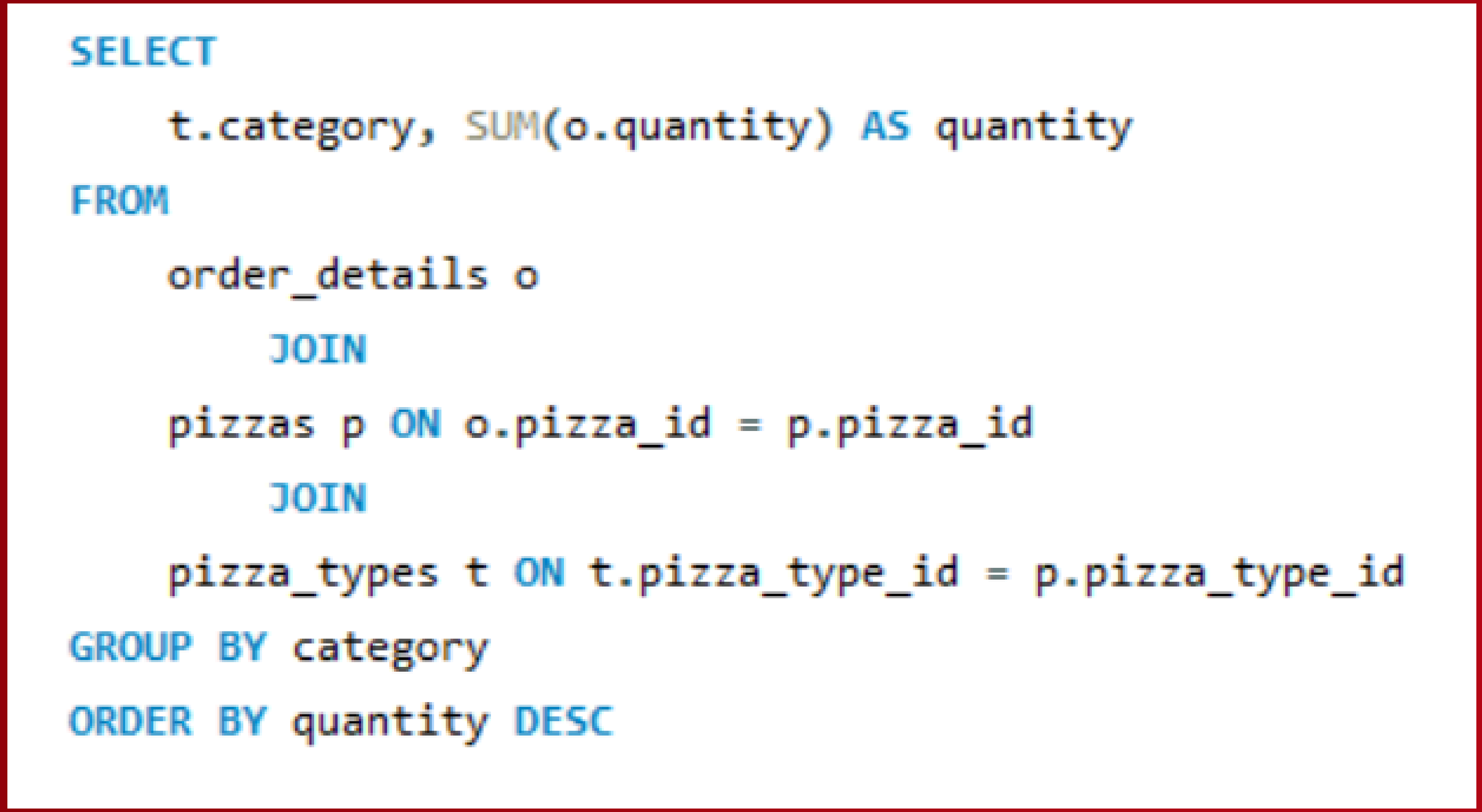


```
SELECT
    t.name, SUM(o.quantity) AS quantity
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY t.name
ORDER BY quantity DESC
LIMIT 5
```

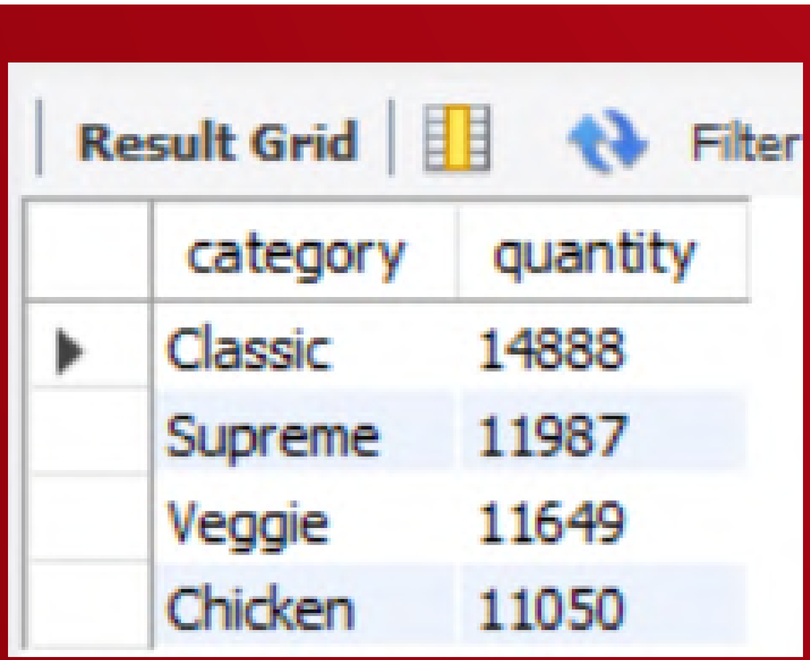


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

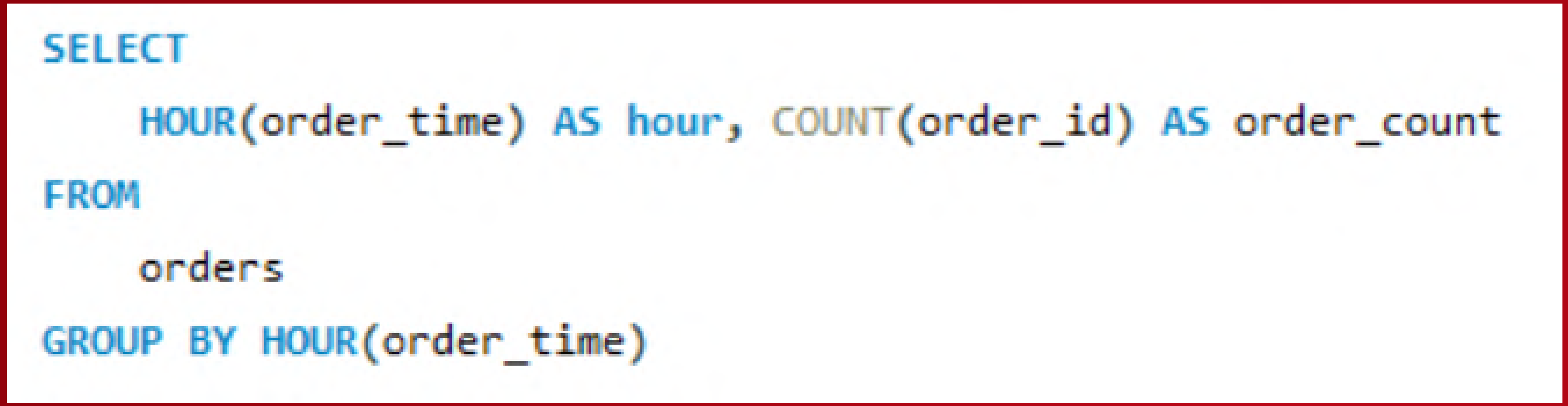


```
SELECT
    t.category, SUM(o.quantity) AS quantity
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY category
ORDER BY quantity DESC
```

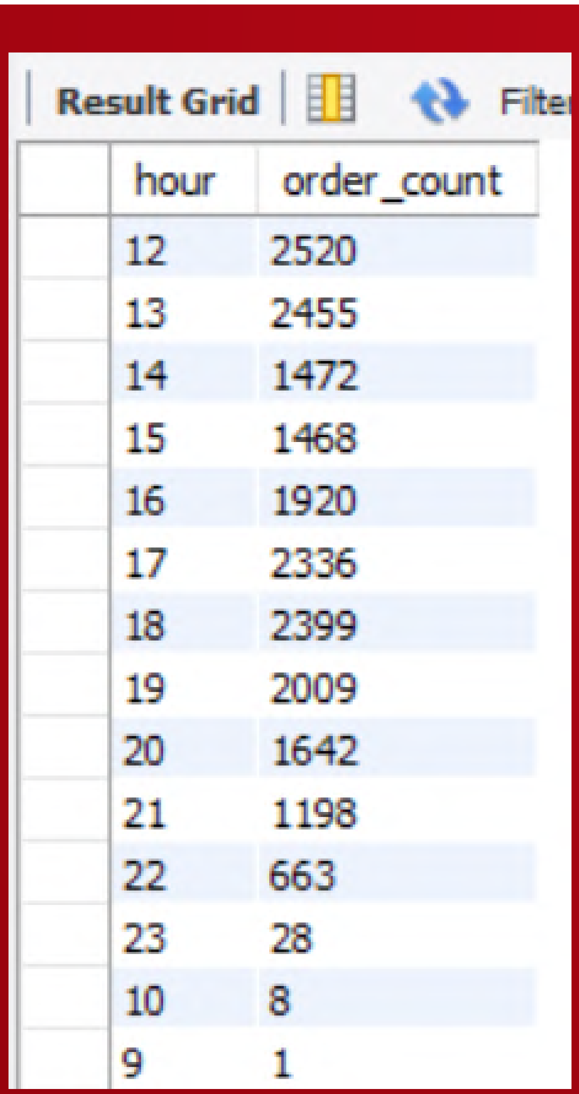


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

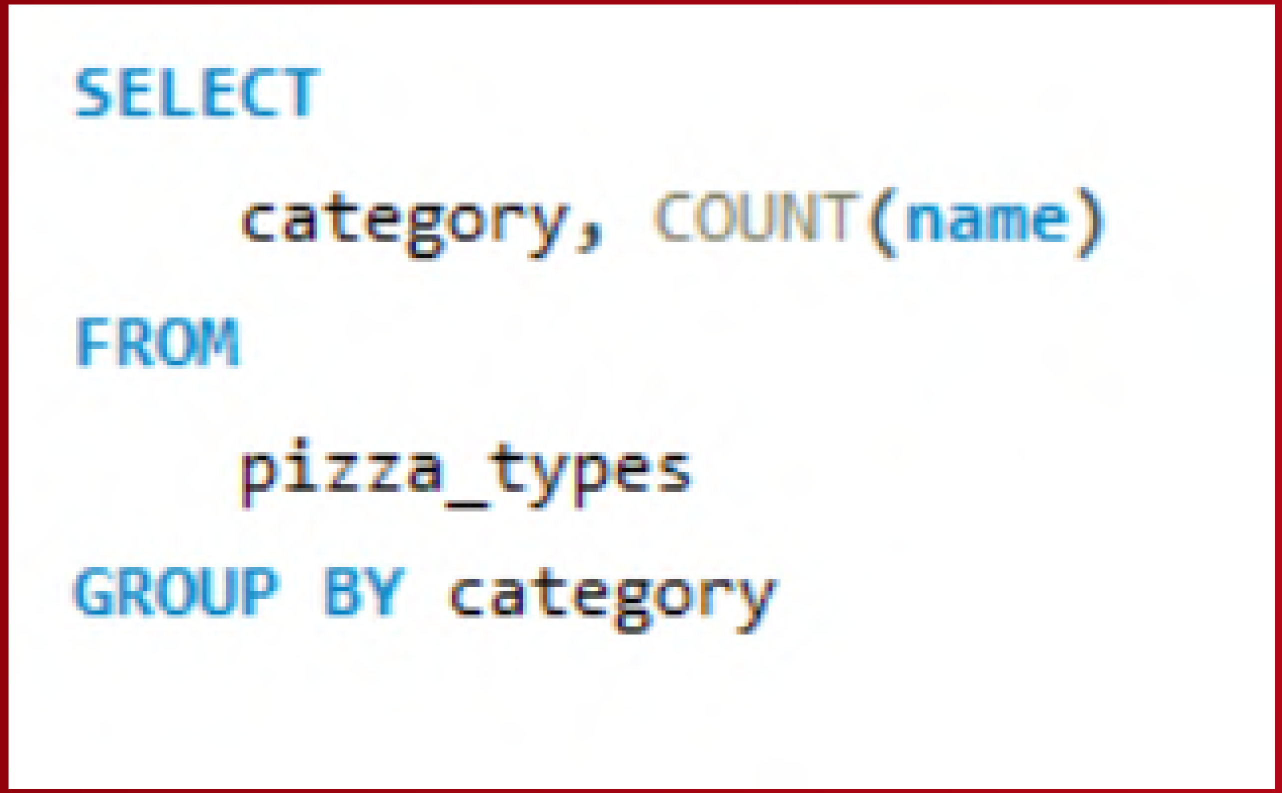


```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
```

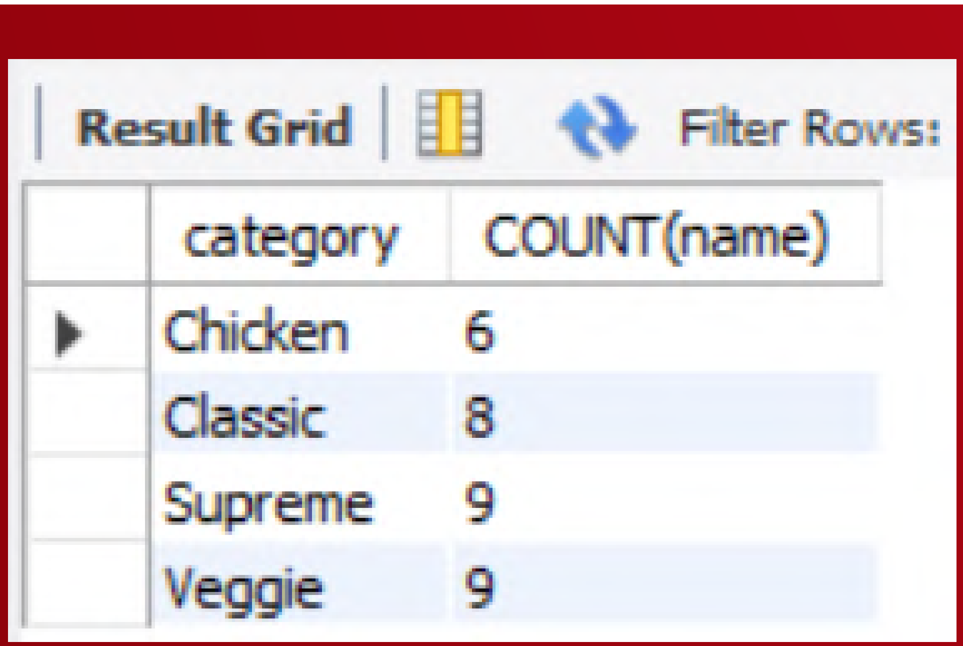


hour	order_count
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

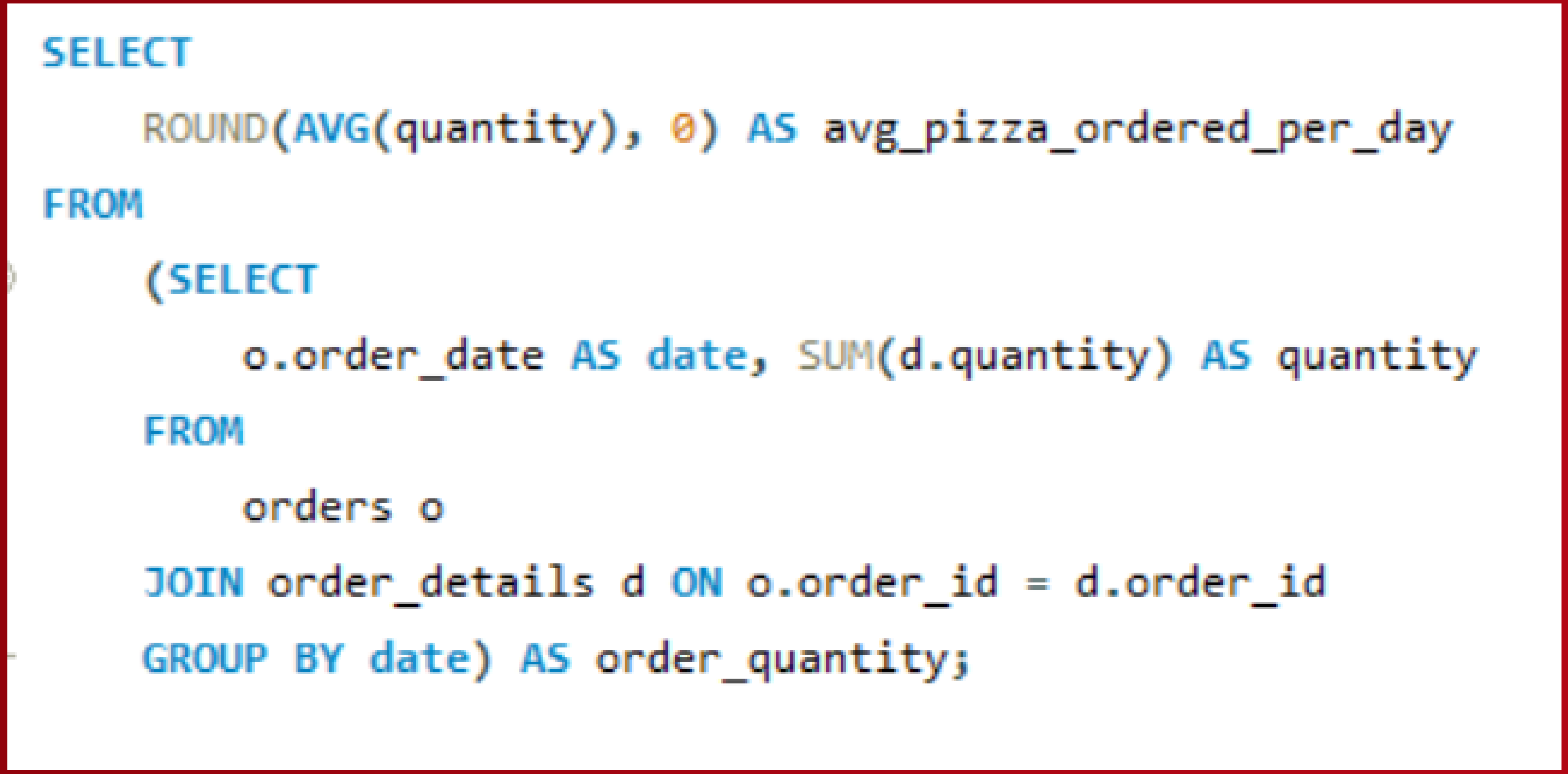


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

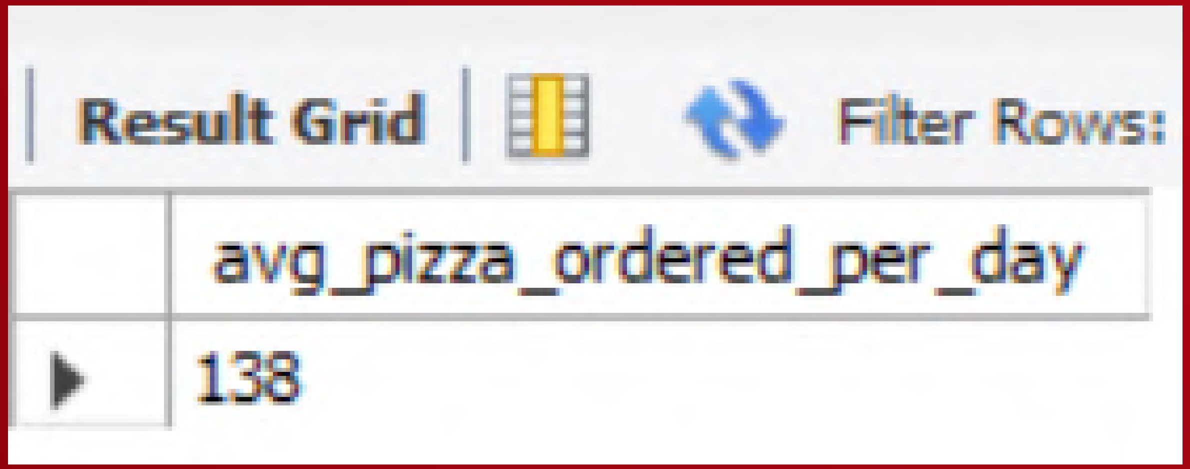


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

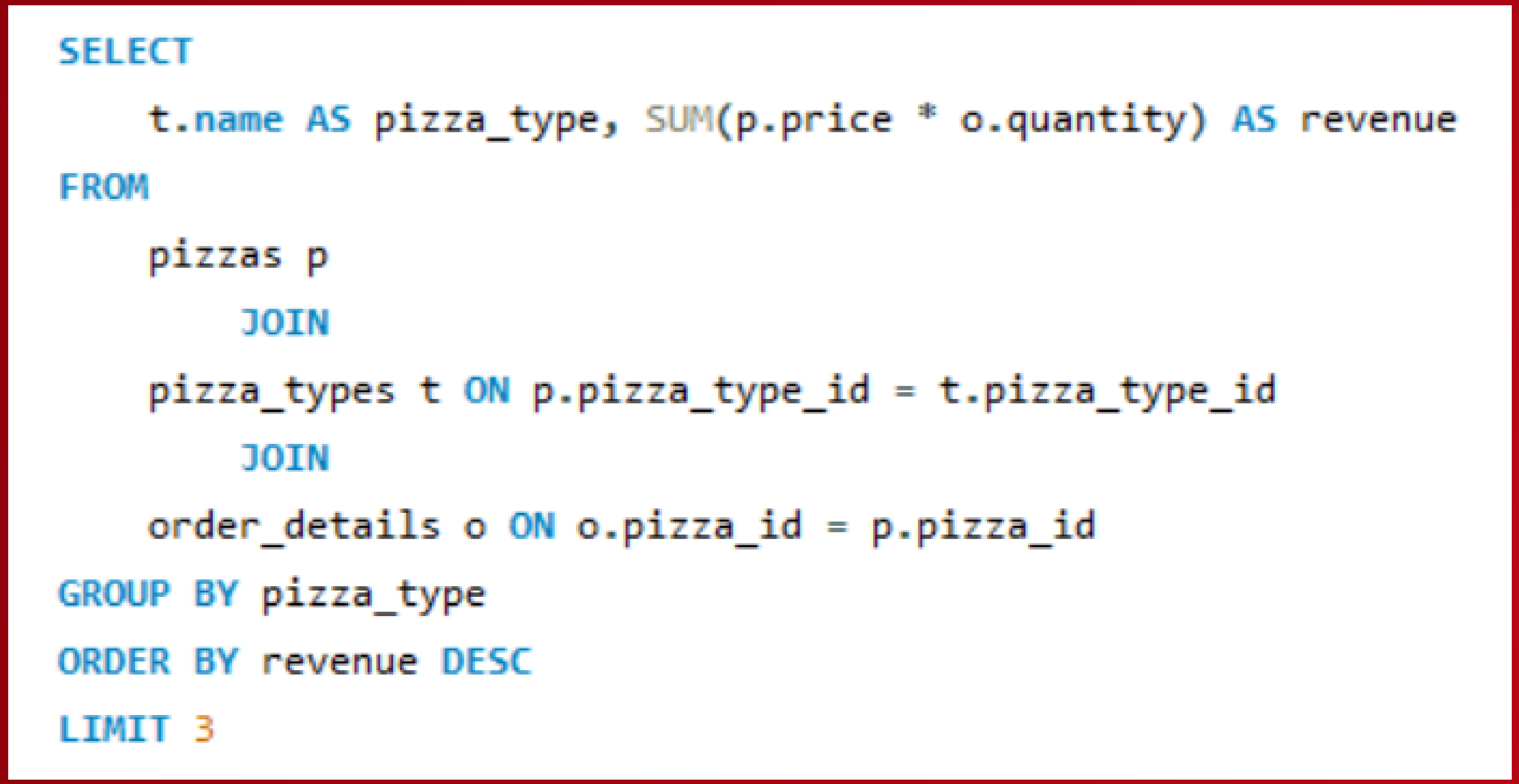


```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        o.order_date AS date, SUM(d.quantity) AS quantity
    FROM
        orders o
    JOIN order_details d ON o.order_id = d.order_id
    GROUP BY date) AS order_quantity;
```

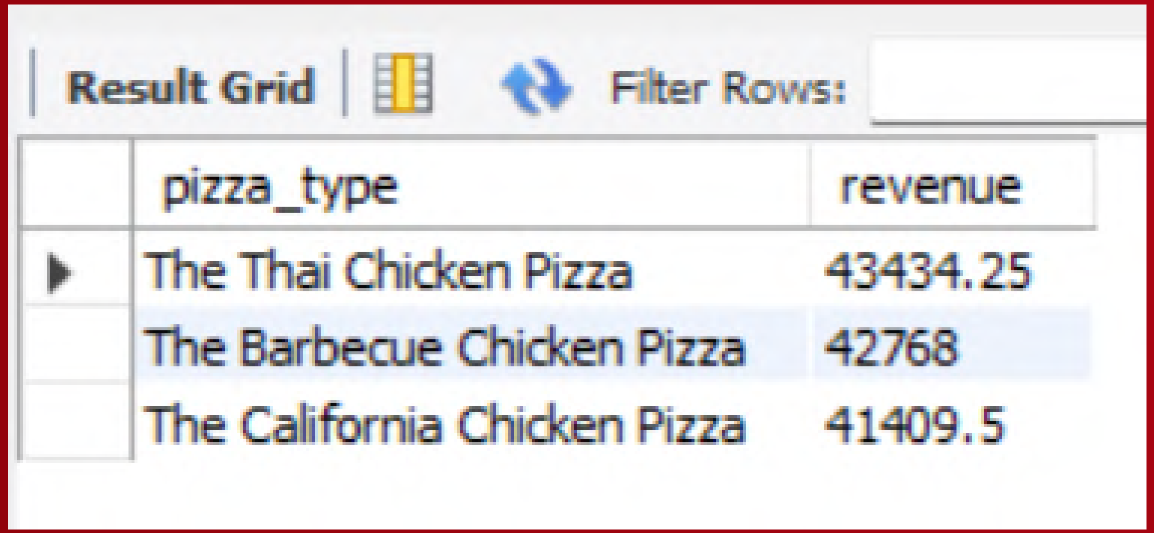


```
avg_pizza_ordered_per_day
138
```

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

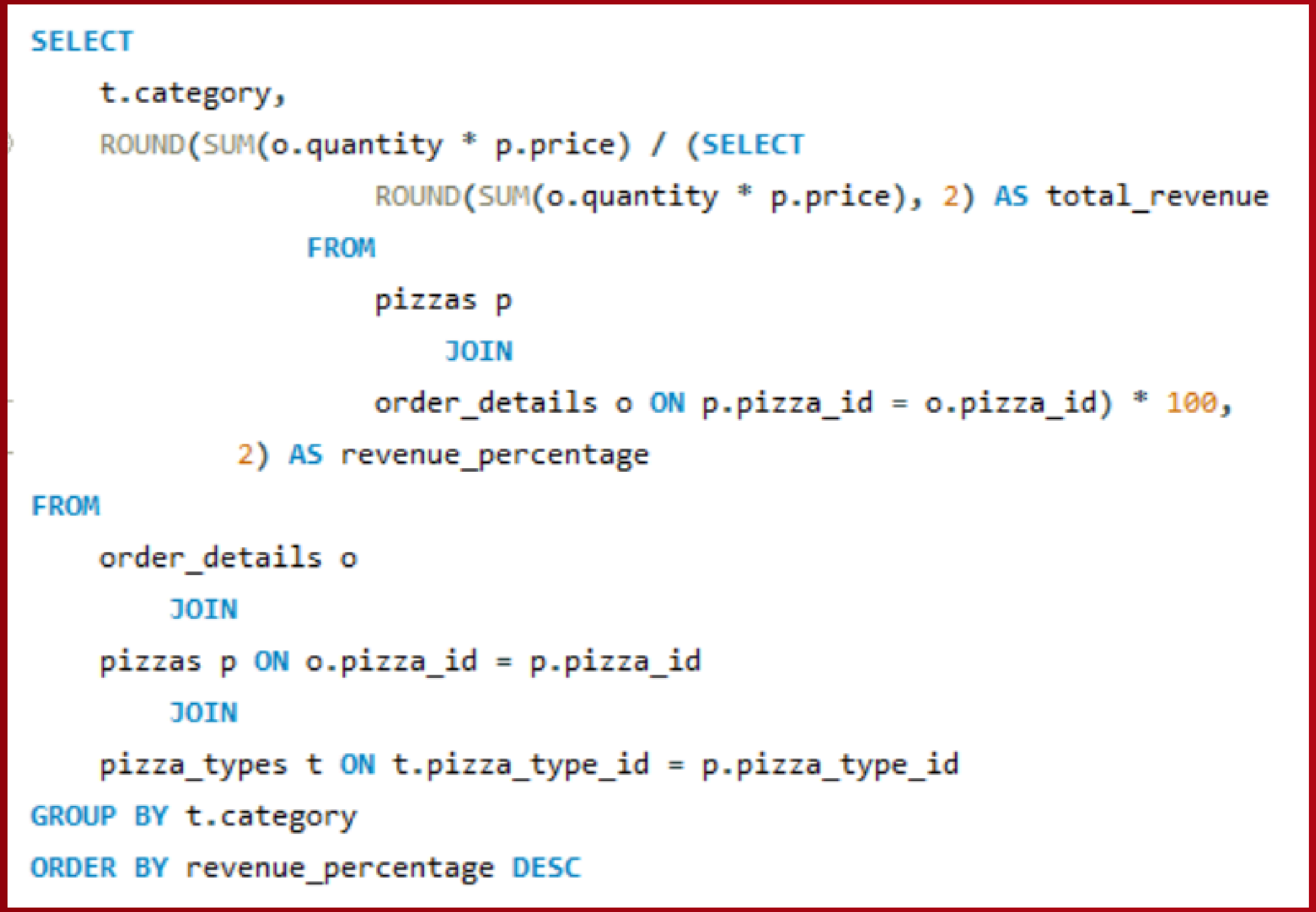


```sql
SELECT
    t.name AS pizza_type, SUM(p.price * o.quantity) AS revenue
FROM
    pizzas p
        JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY pizza_type
ORDER BY revenue DESC
LIMIT 3
```

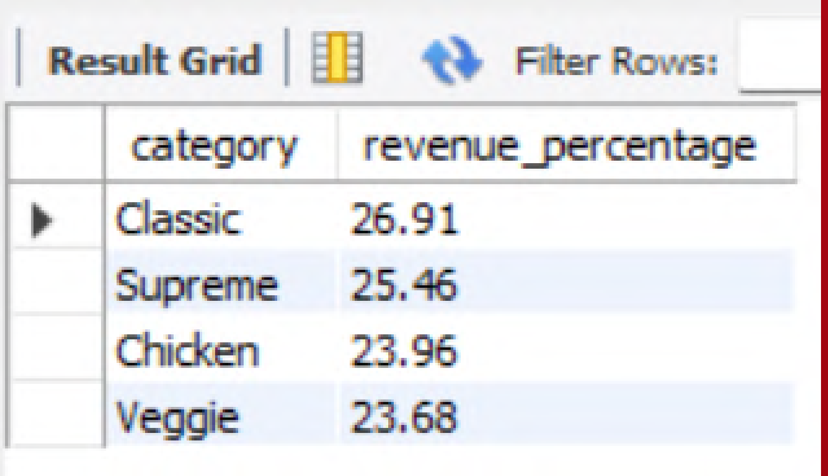


pizza_type	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

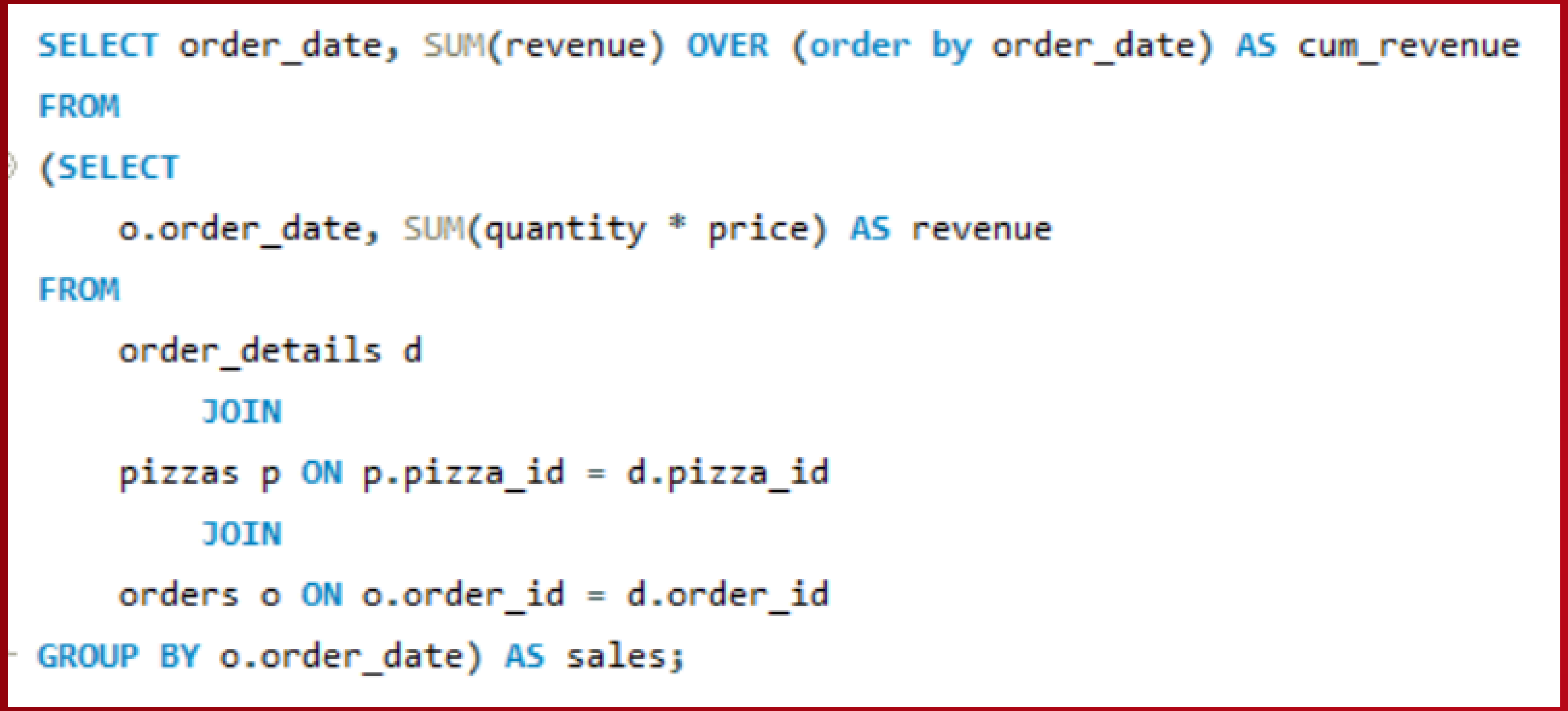


```sql
SELECT
    t.category,
    ROUND(SUM(o.quantity * p.price) / (SELECT
                ROUND(SUM(o.quantity * p.price), 2) AS total_revenue
            FROM
                pizzas p
                    JOIN
                order_details o ON p.pizza_id = o.pizza_id) * 100,
            2) AS revenue_percentage
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY t.category
ORDER BY revenue_percentage DESC
```

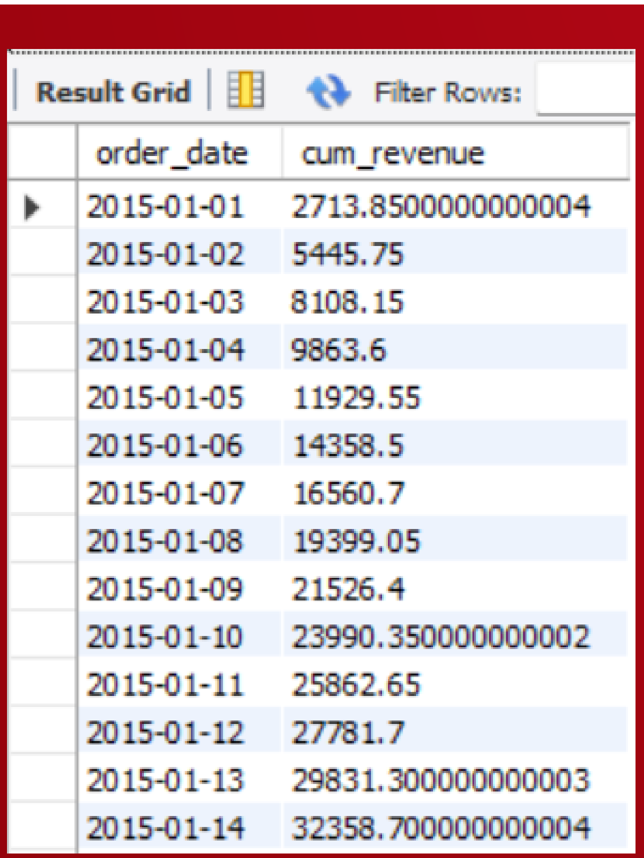


category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

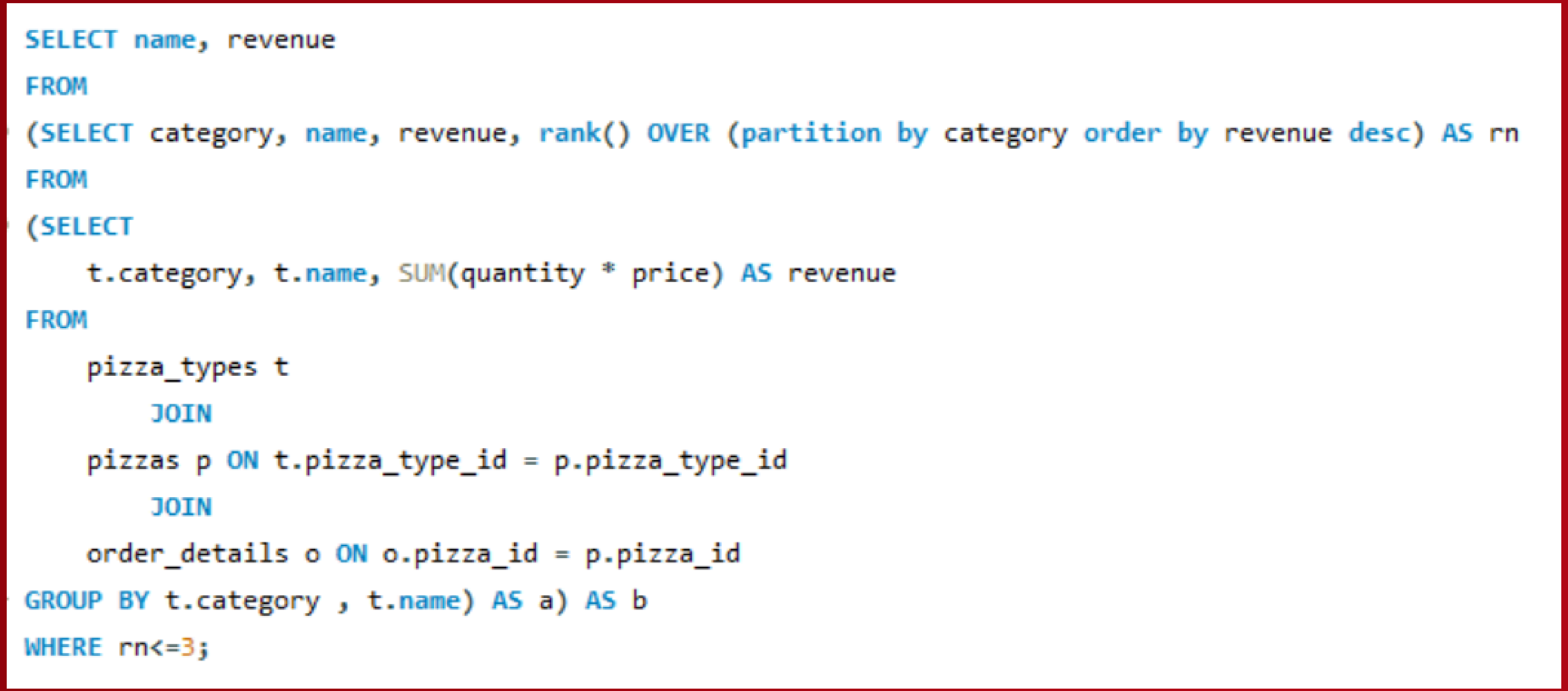


```
SELECT order_date, SUM(revenue) OVER (order by order_date) AS cum_revenue
FROM
(SELECT
    o.order_date, SUM(quantity * price) AS revenue
FROM
    order_details d
        JOIN
    pizzas p ON p.pizza_id = d.pizza_id
        JOIN
    orders o ON o.order_id = d.order_id
GROUP BY o.order_date) AS sales;
```

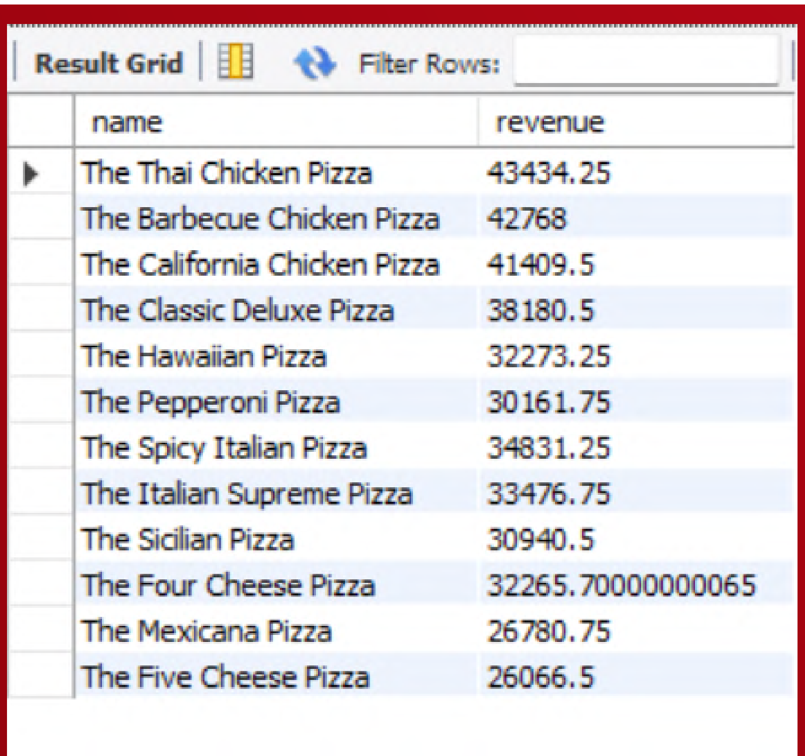


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

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



```sql
SELECT name, revenue
FROM
(SELECT category, name, revenue, rank() OVER (partition by category order by revenue desc) AS rn
FROM
(SELECT
    t.category, t.name, SUM(quantity * price) AS revenue
FROM
    pizza_types t
        JOIN
    pizzas p ON t.pizza_type_id = p.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY t.category , t.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

FRESH PIZZAHUB

THANK YOU!

"Thank you for exploring the data-powered journey of Fresh PizzaHub! With MySQL insights, we're crafting smarter decisions and delivering fresh, flavorful experiences one slice at a time.