SQL PROJECT ON

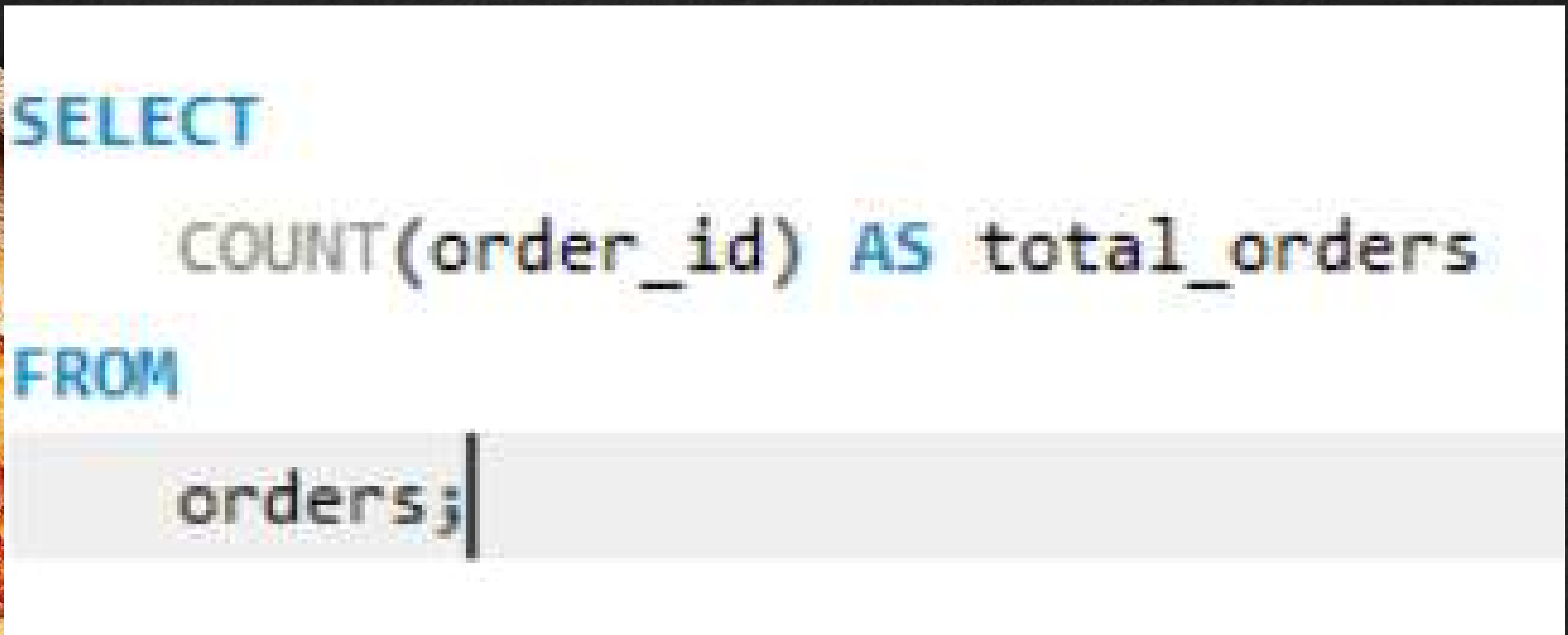
PIZZA SALES ANALYSIS

INTRODUCTION

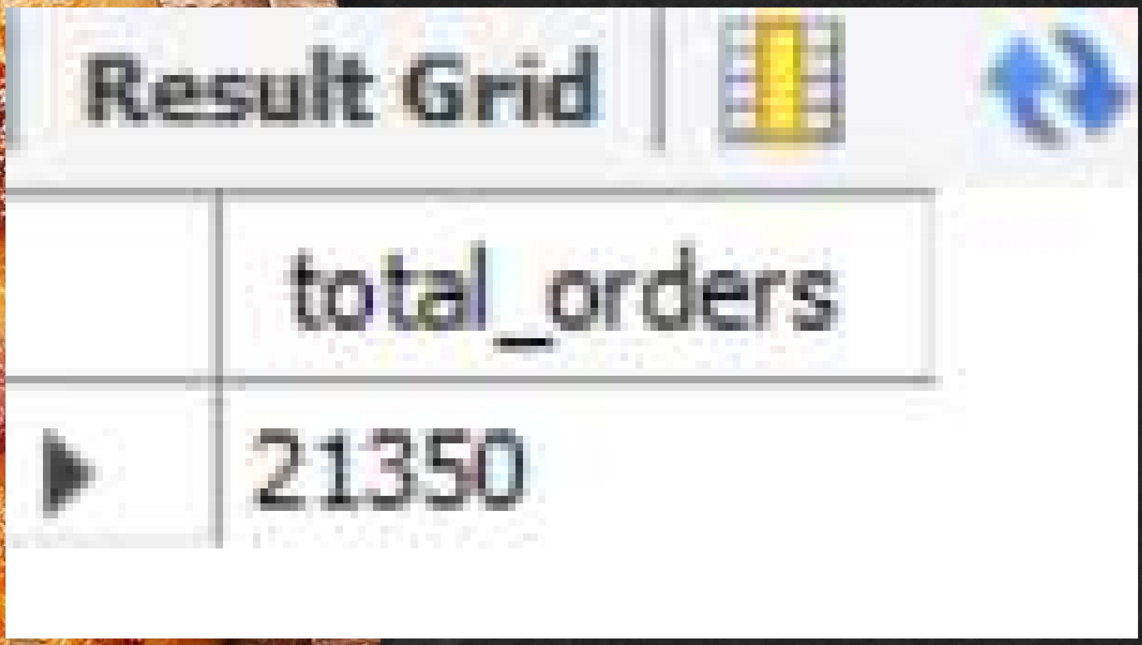
Used some data set on pizza sales in the form of csv files.

Executed various easy,medium and hard level queries for the data analysis.

Retrieve the total number of orders placed.

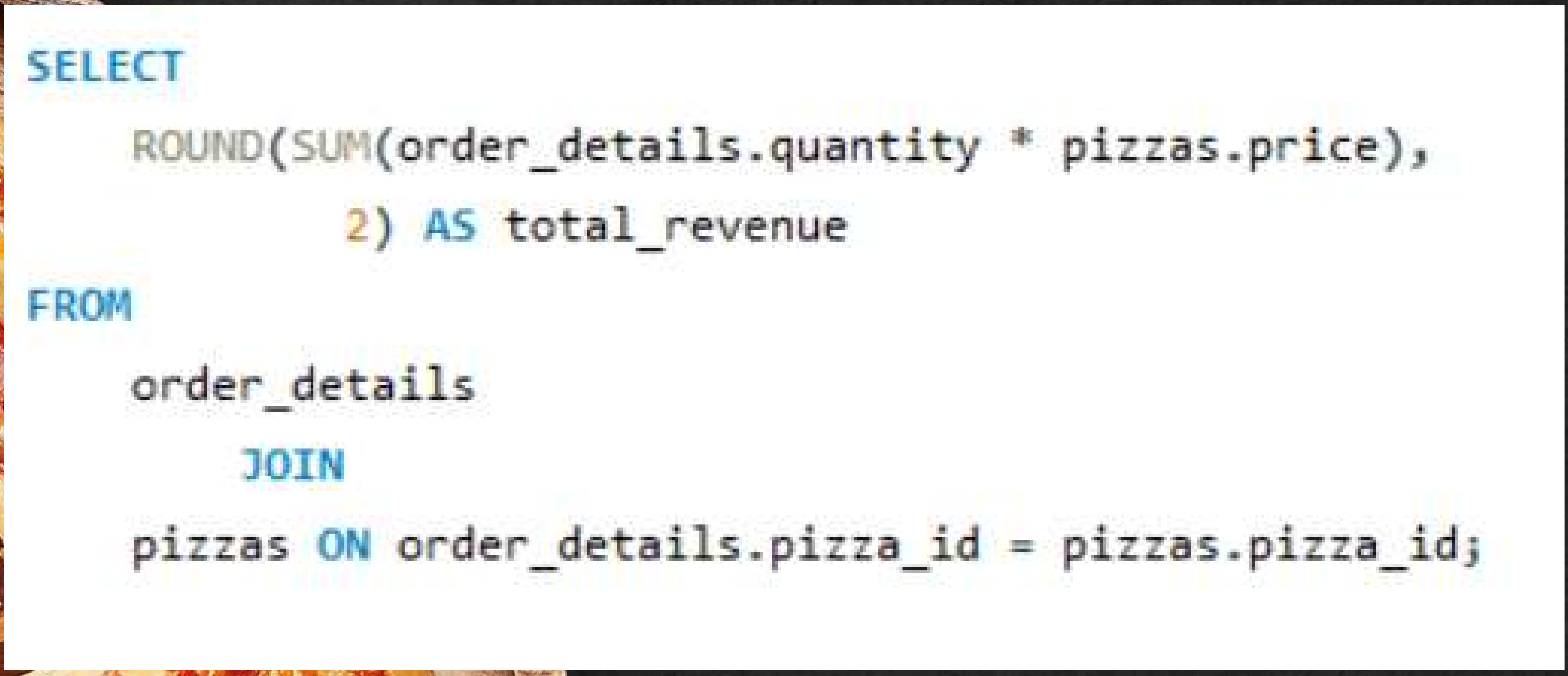


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

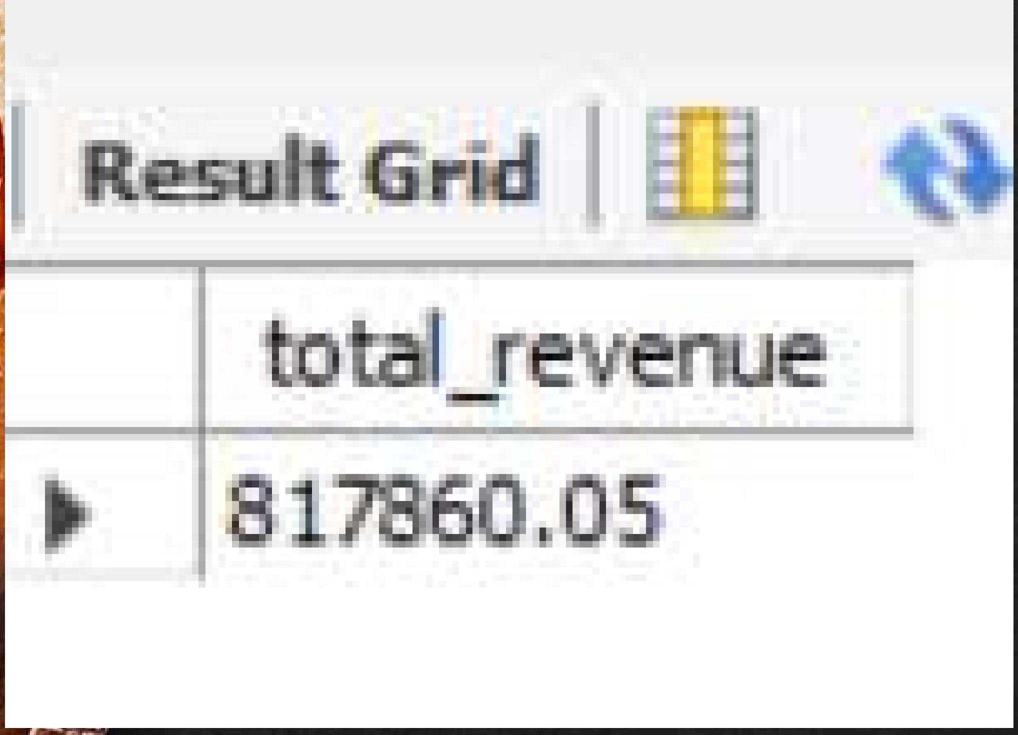


total_orders
21350

Calculate the total revenue generated from pizza sales.

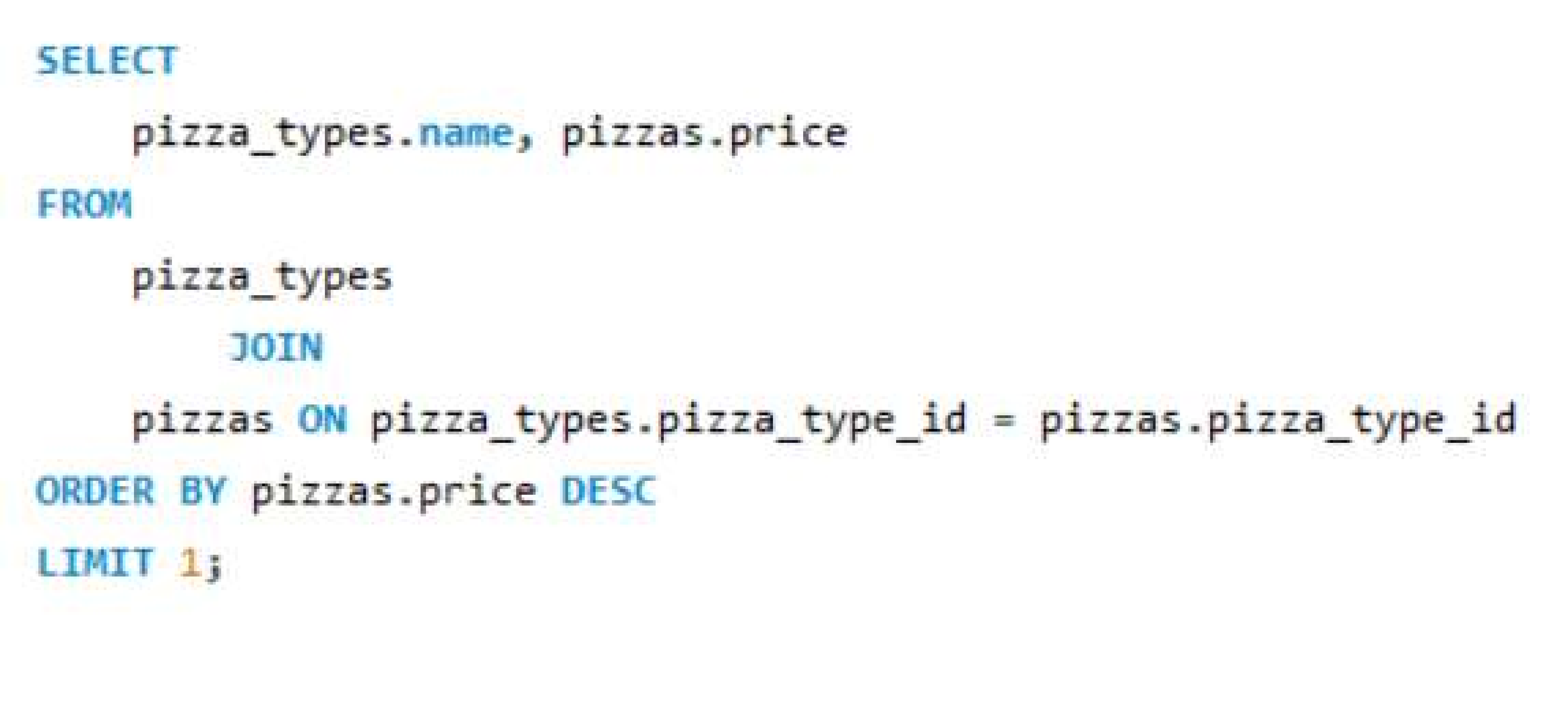


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

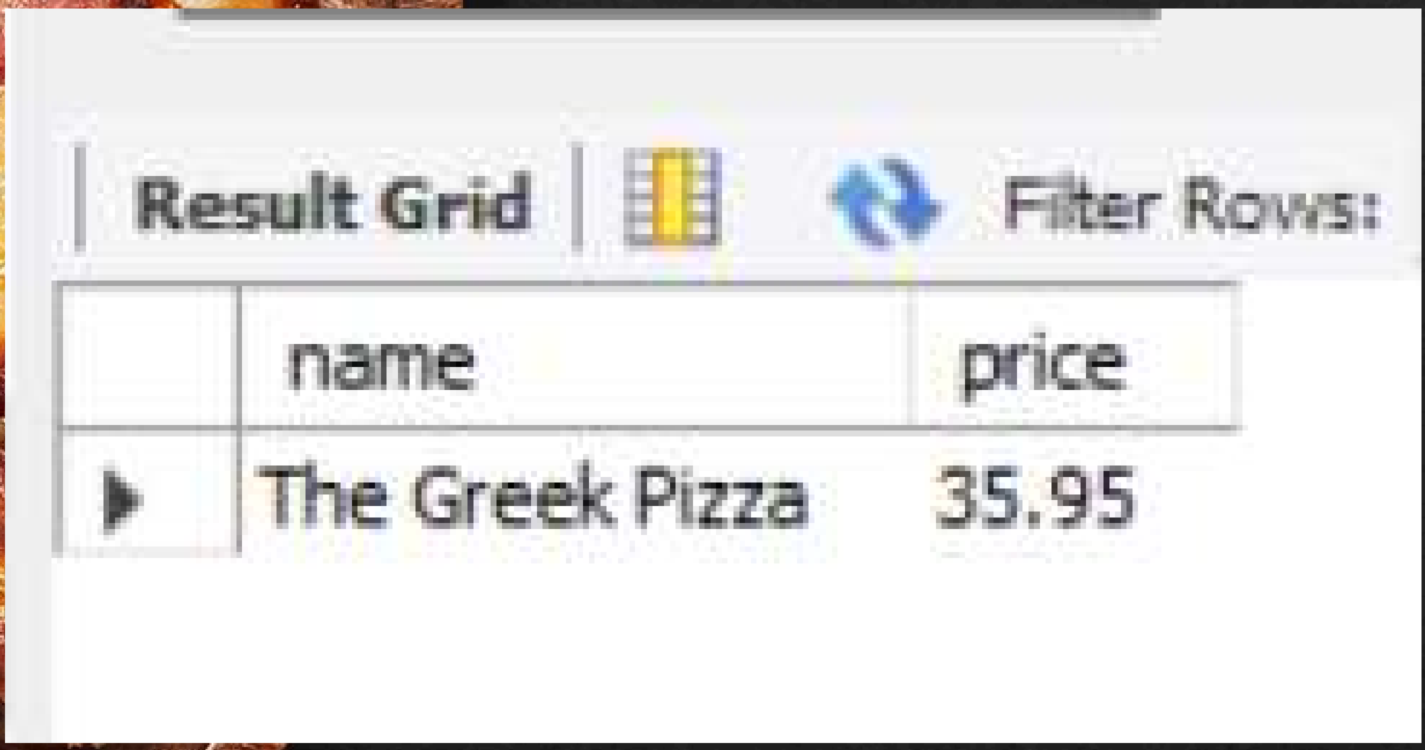


```
total_revenue
817860.05
```

Identify the highest-priced pizza.

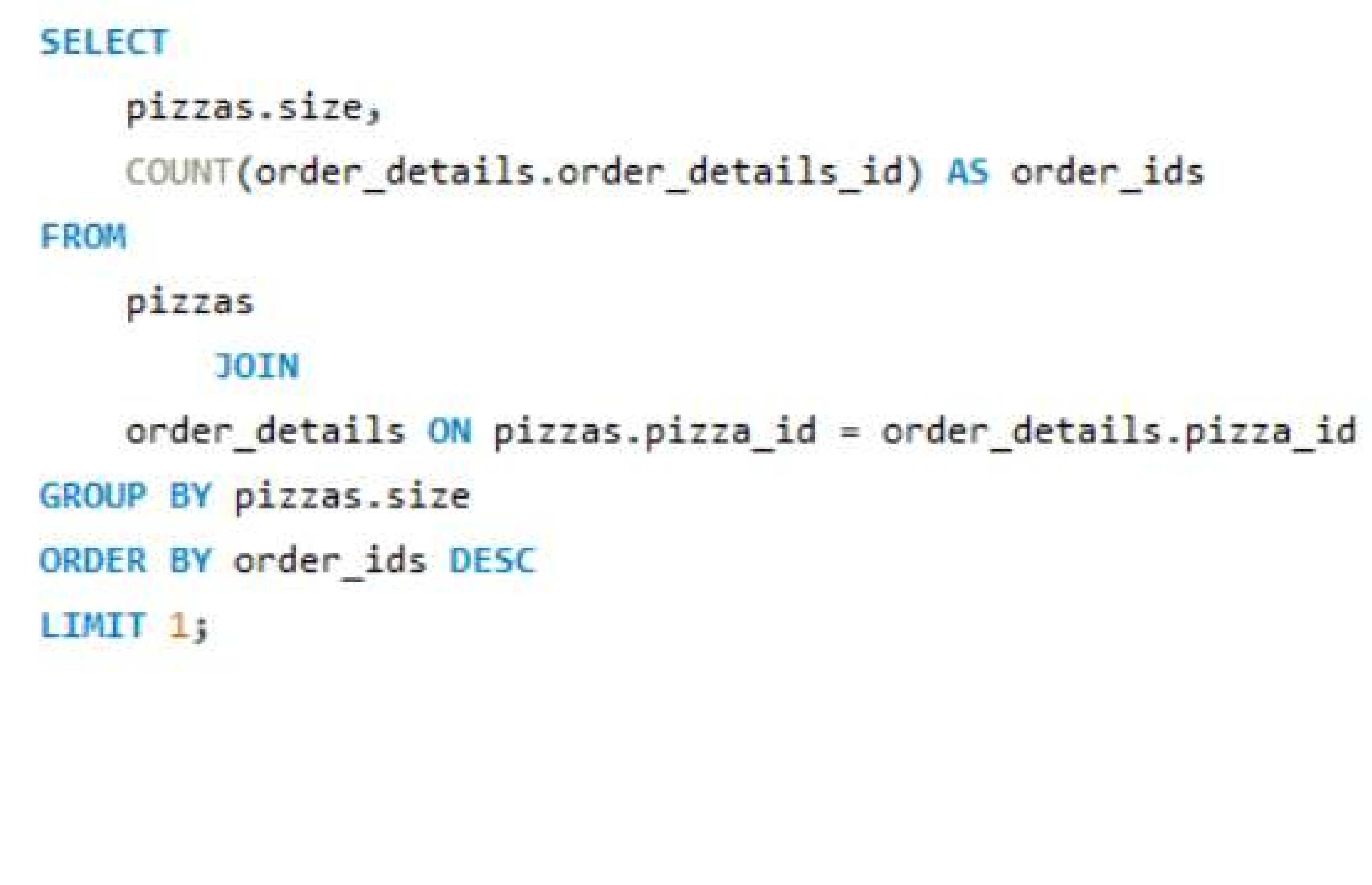


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

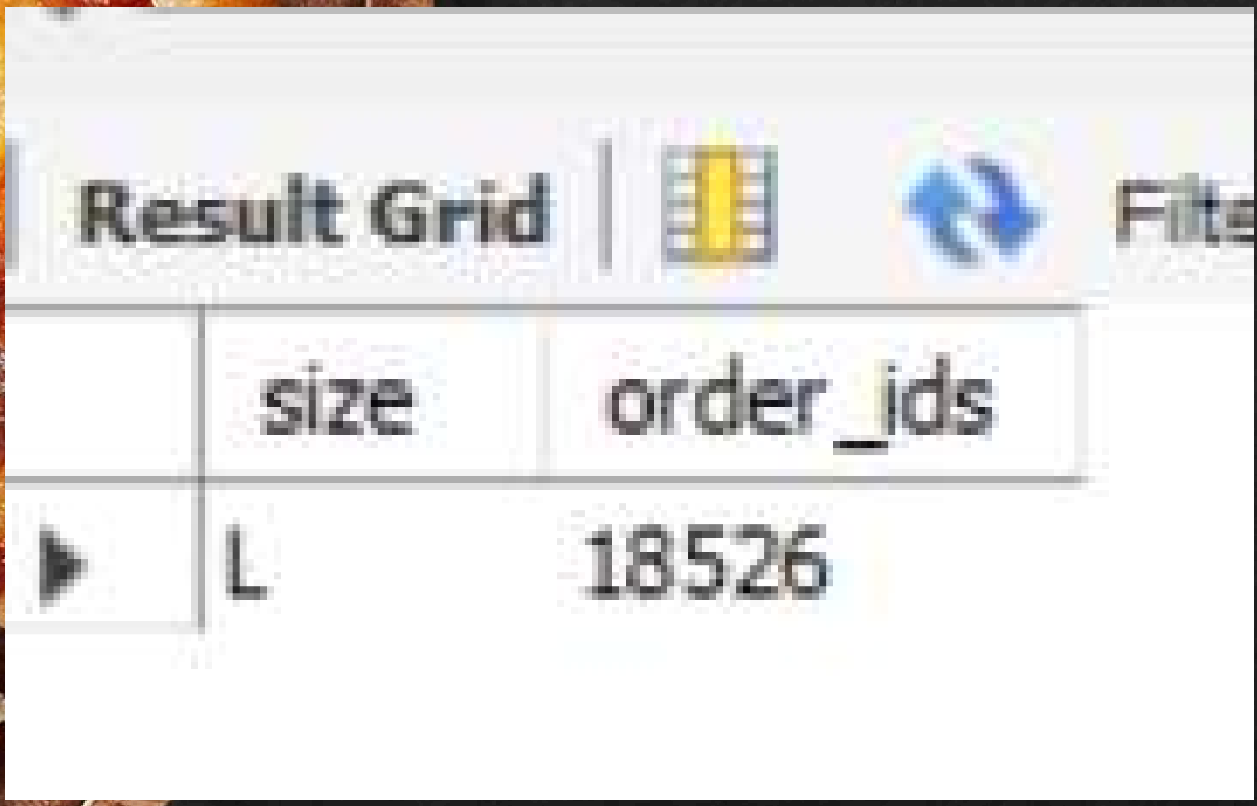


	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

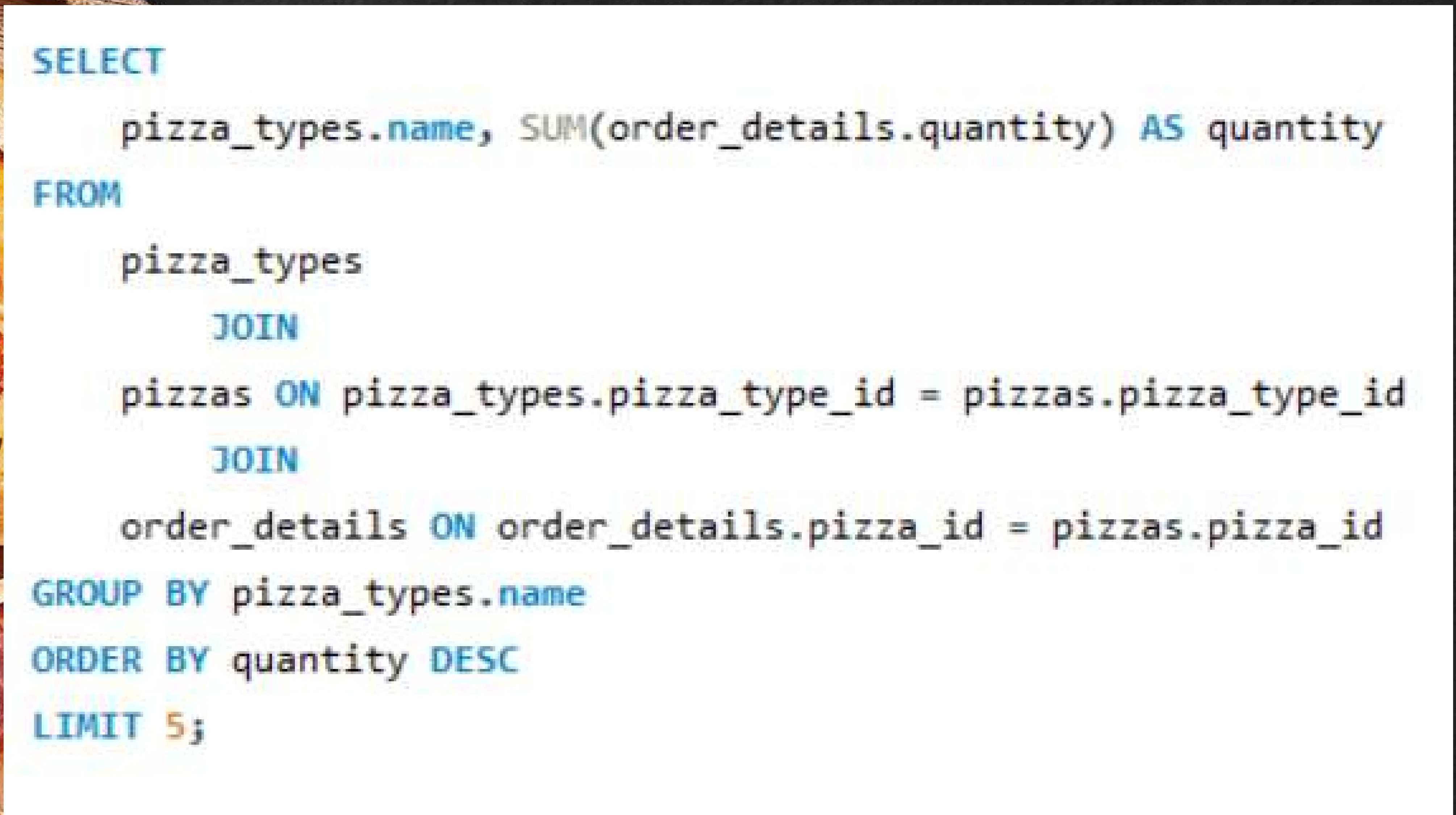


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_ids
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_ids DESC
LIMIT 1;
```

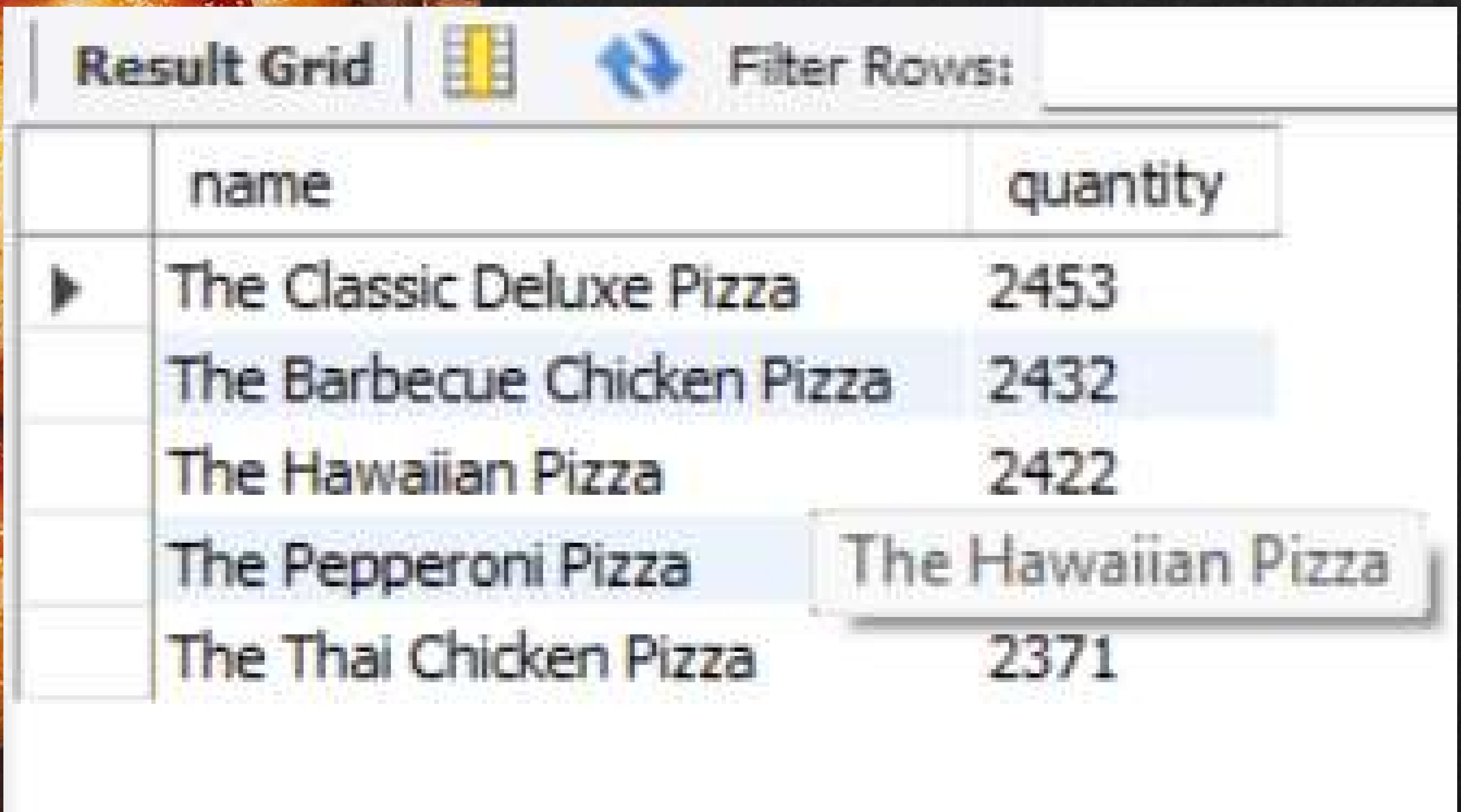


```
  size   order_ids
▶ L      18526
```

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

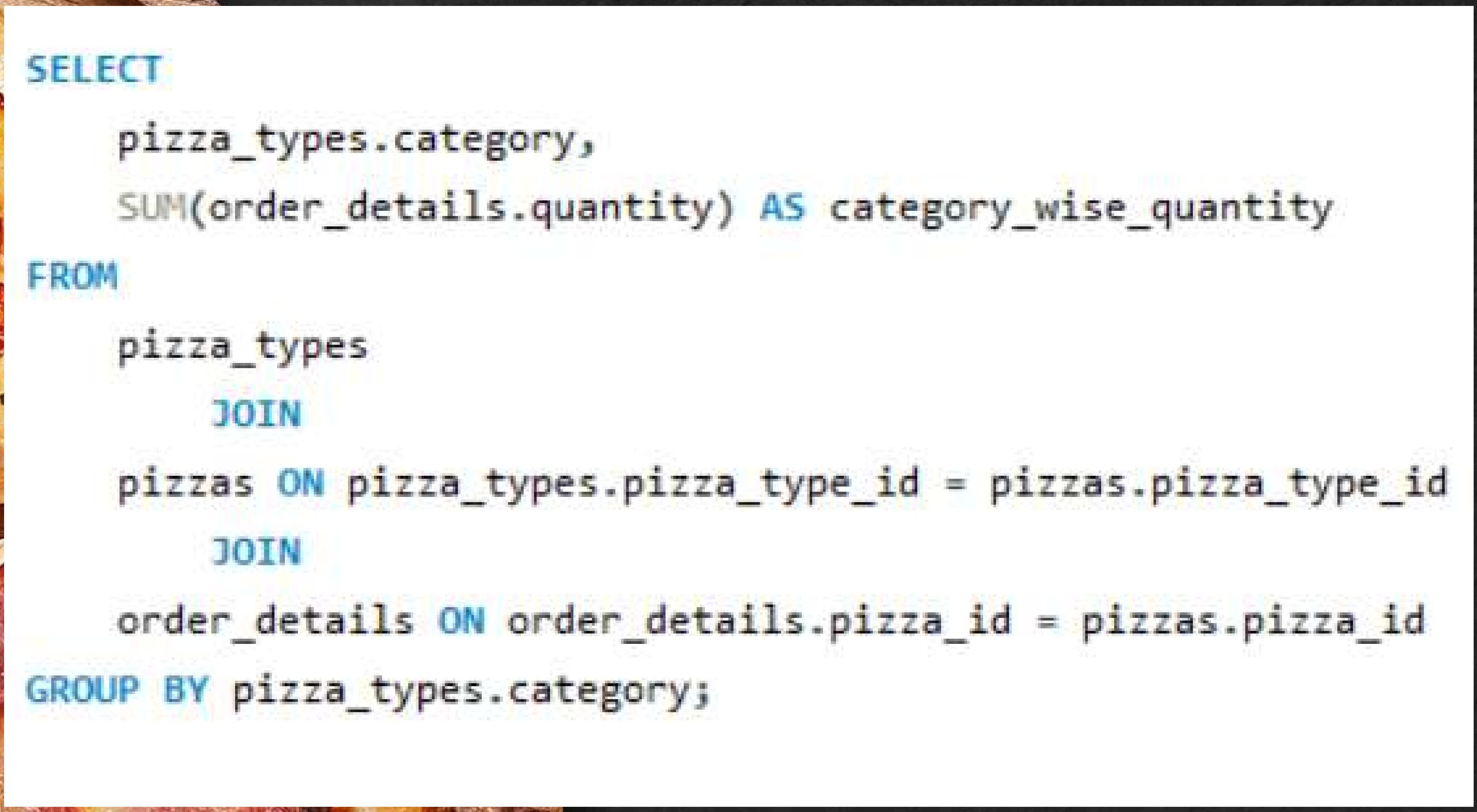


```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

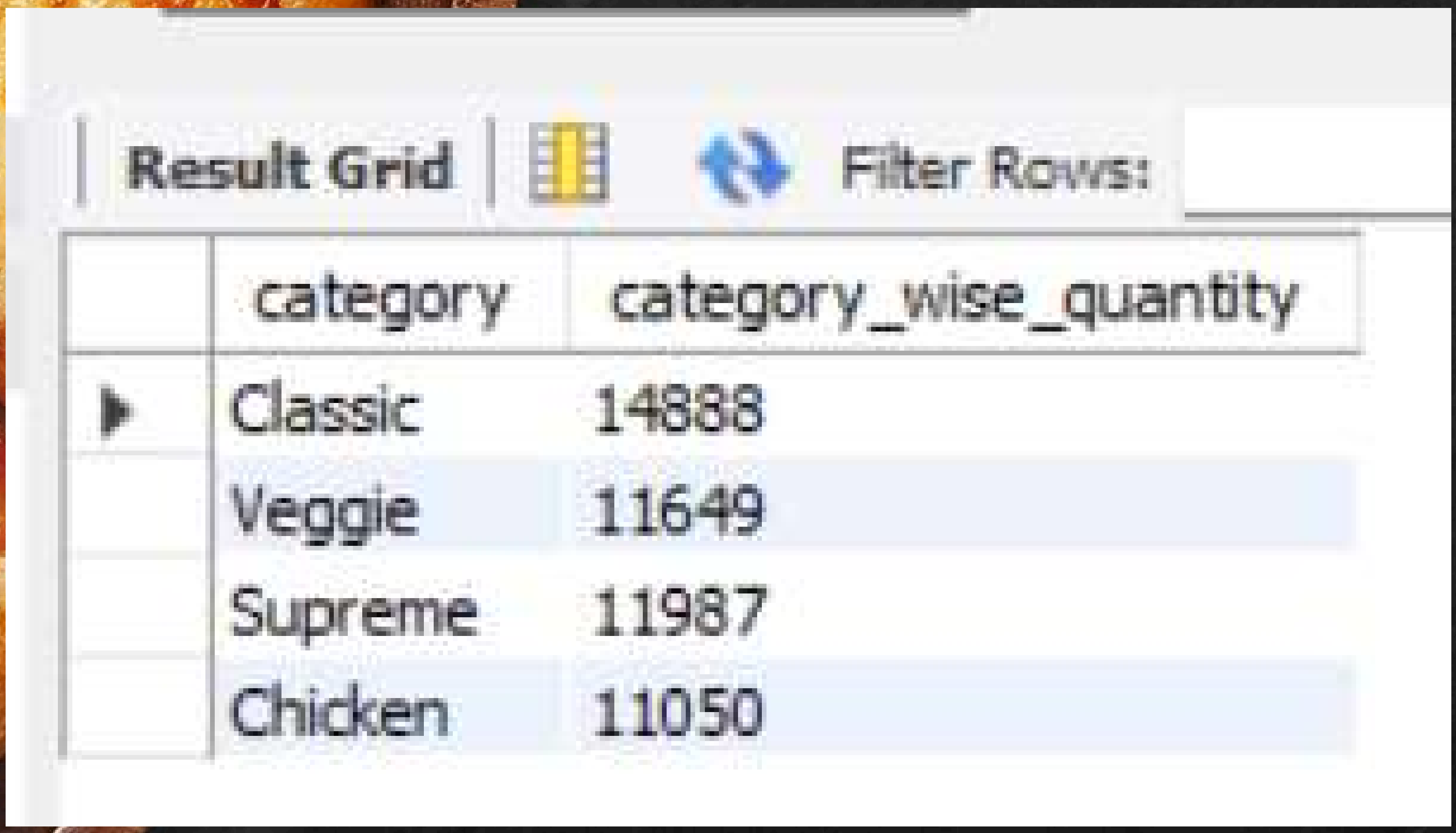


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	[illegible]
The Thai Chicken Pizza	2371

Join the necessary tables to find the total quantity of each pizza category ordered.

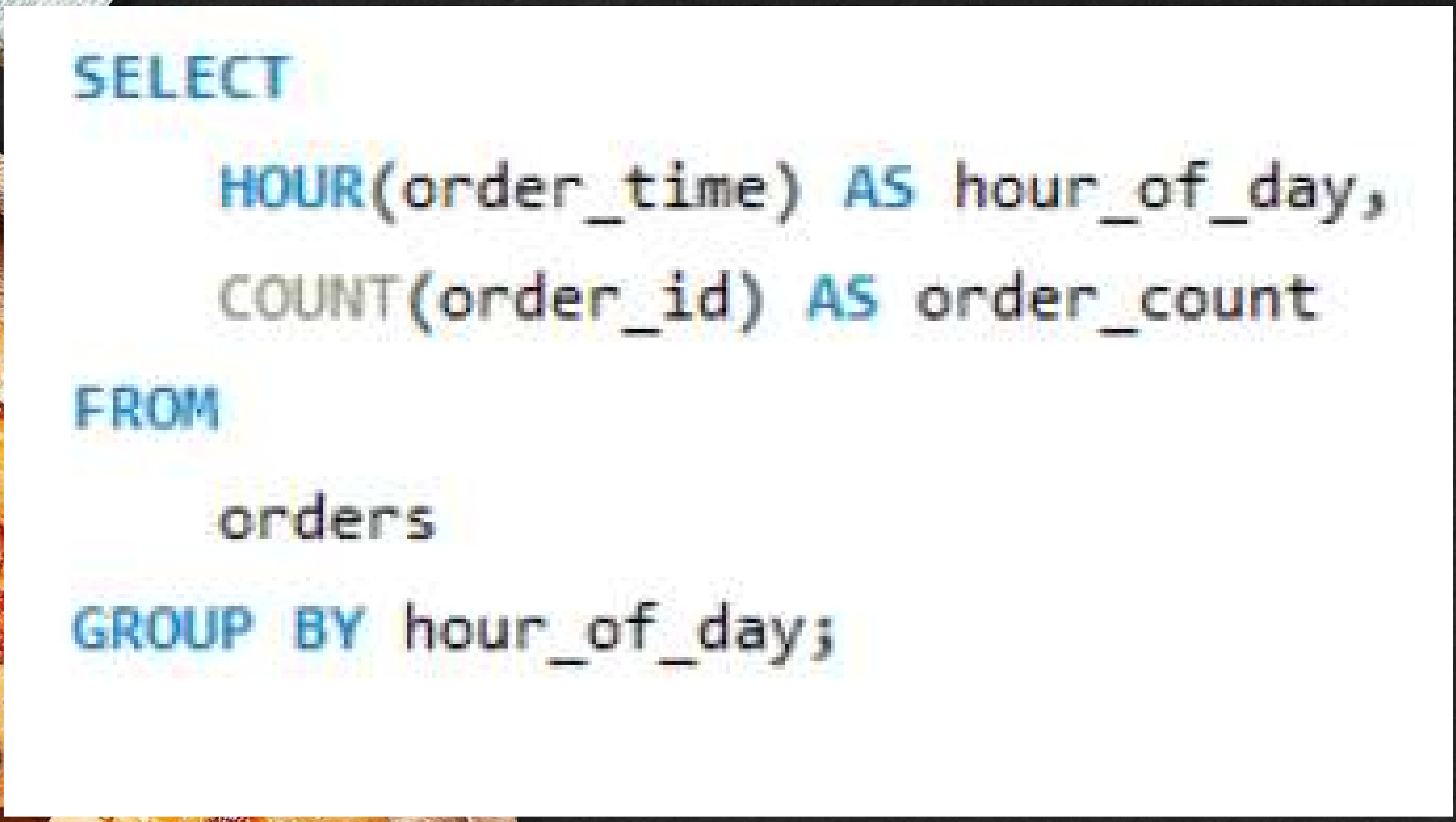


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS category_wise_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

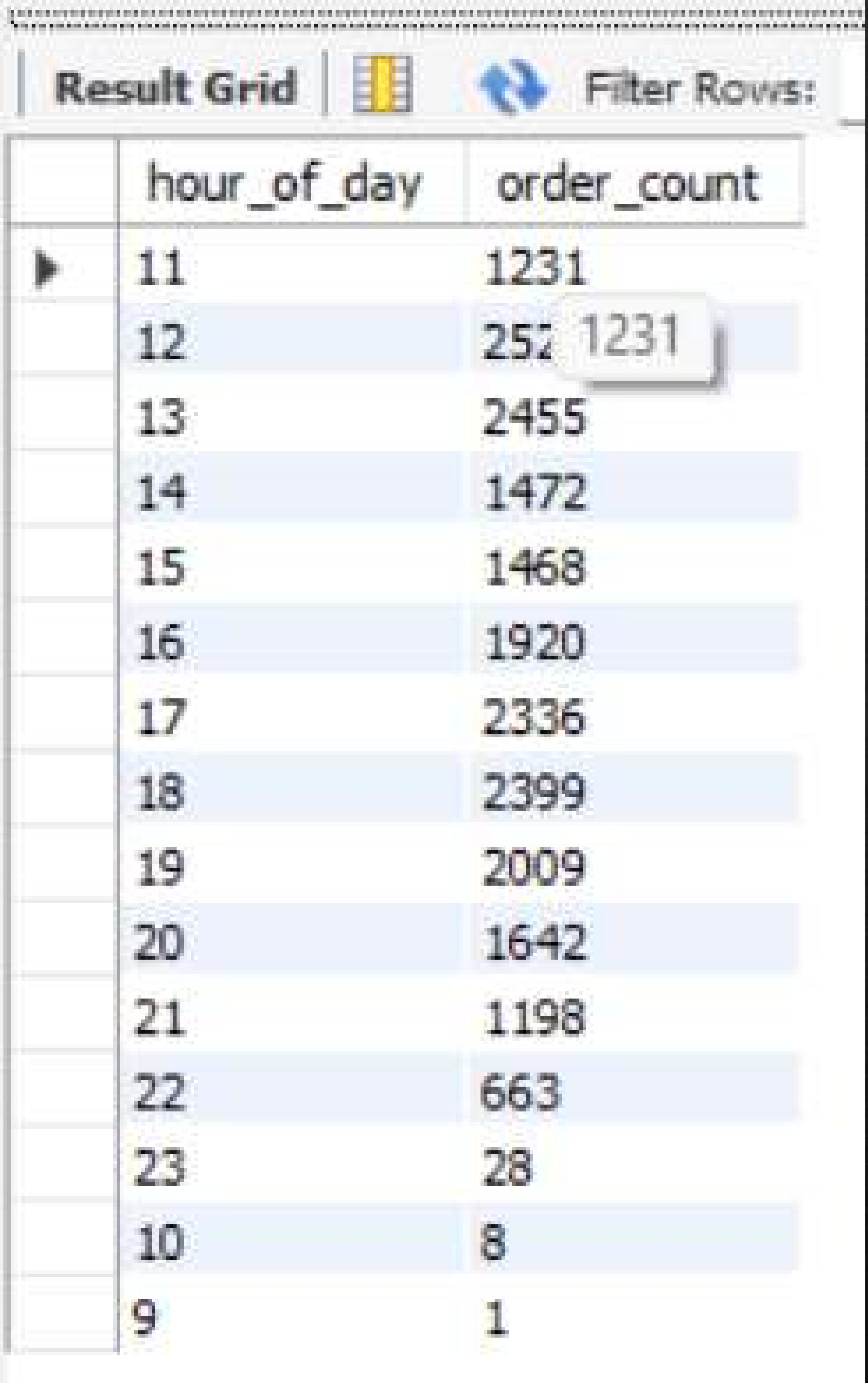


category	category_wise_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day.

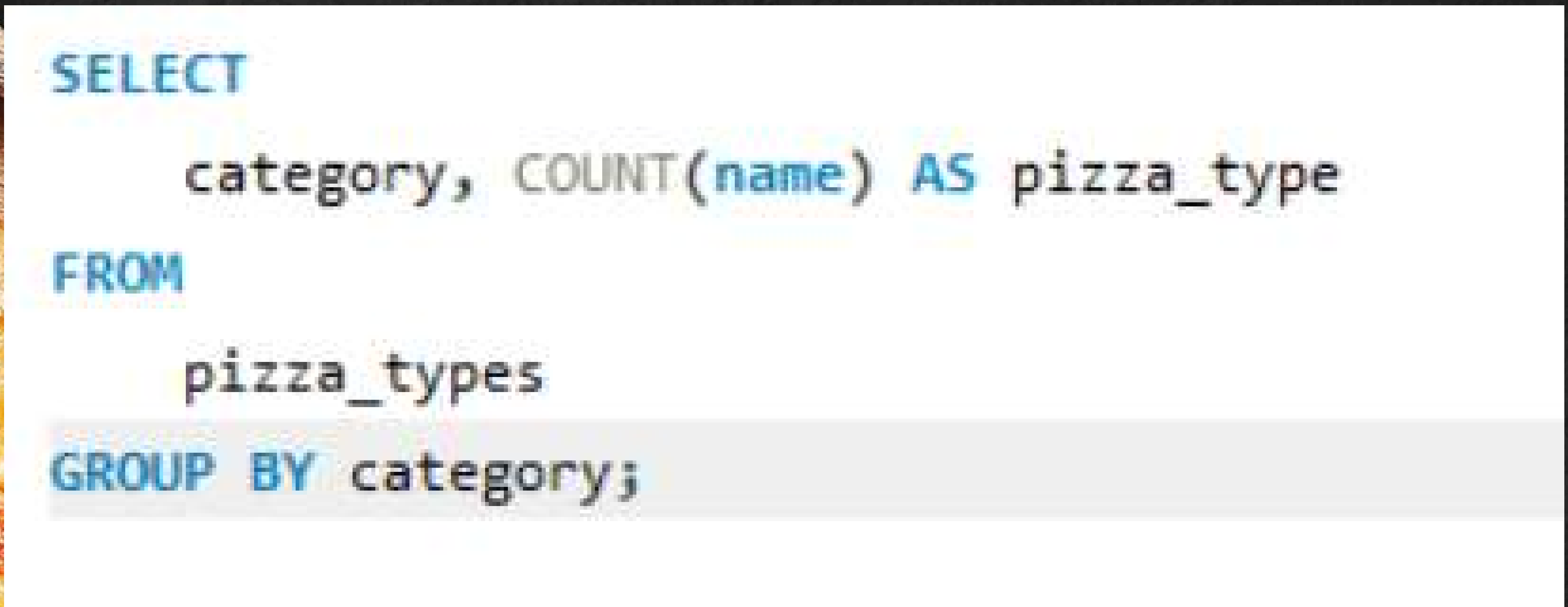


```sql
SELECT
    HOUR(order_time) AS hour_of_day,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour_of_day;
```

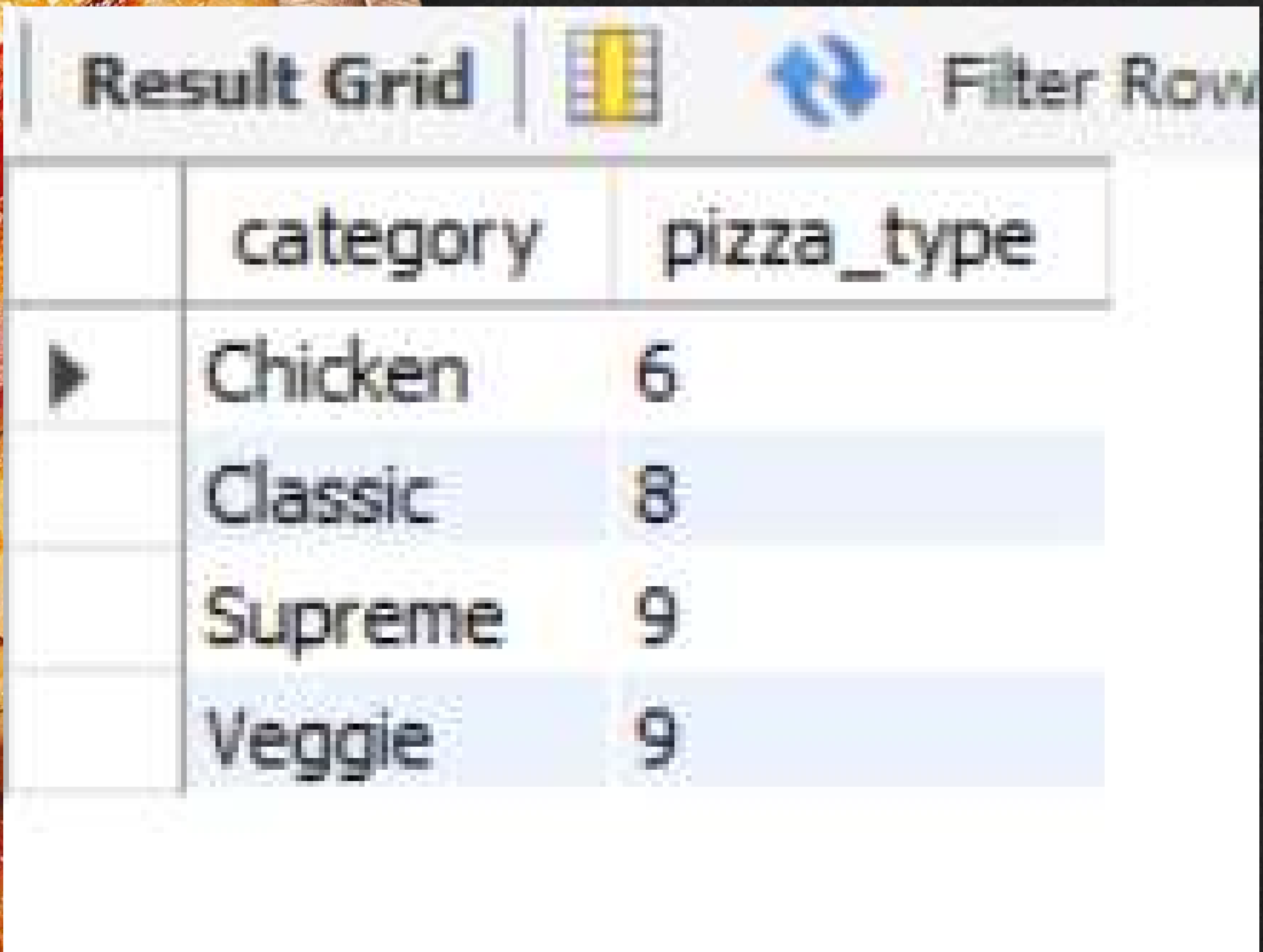


hour_of_day	order_count
11	1231
12	252 1231
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Join relevant tables to find the category-wise distribution of pizzas.

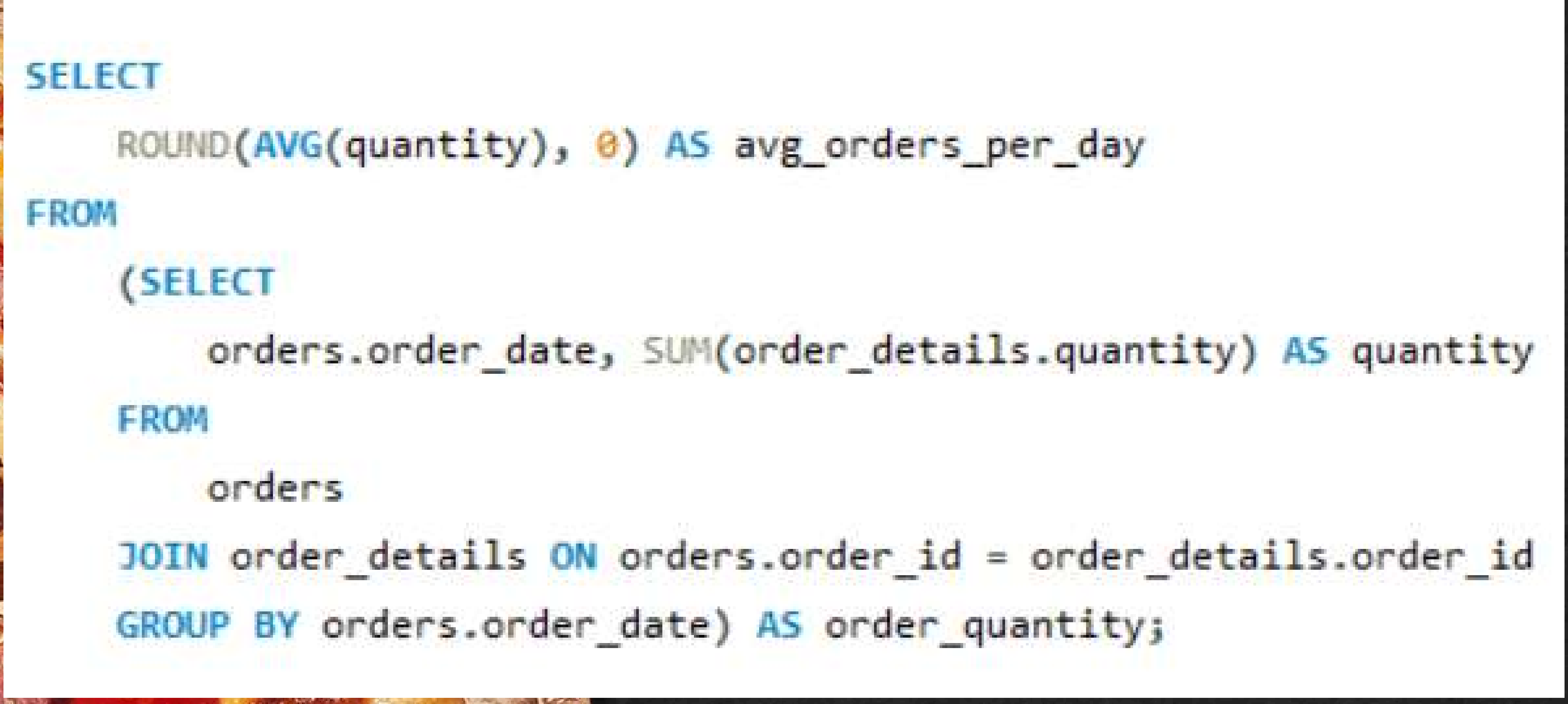


```
SELECT
    category, COUNT(name) AS pizza_type
FROM
    pizza_types
GROUP BY category;
```

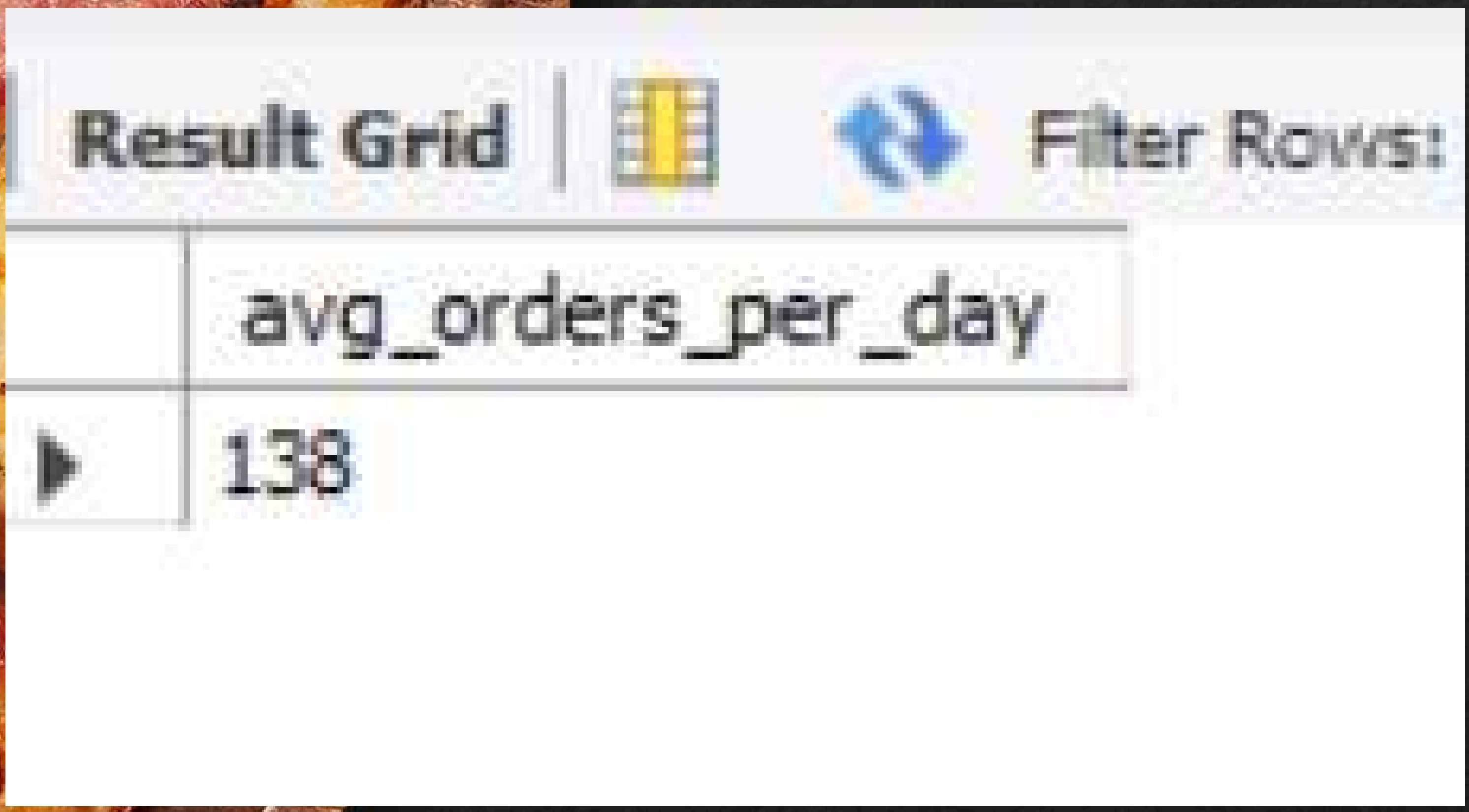


category	pizza_type
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

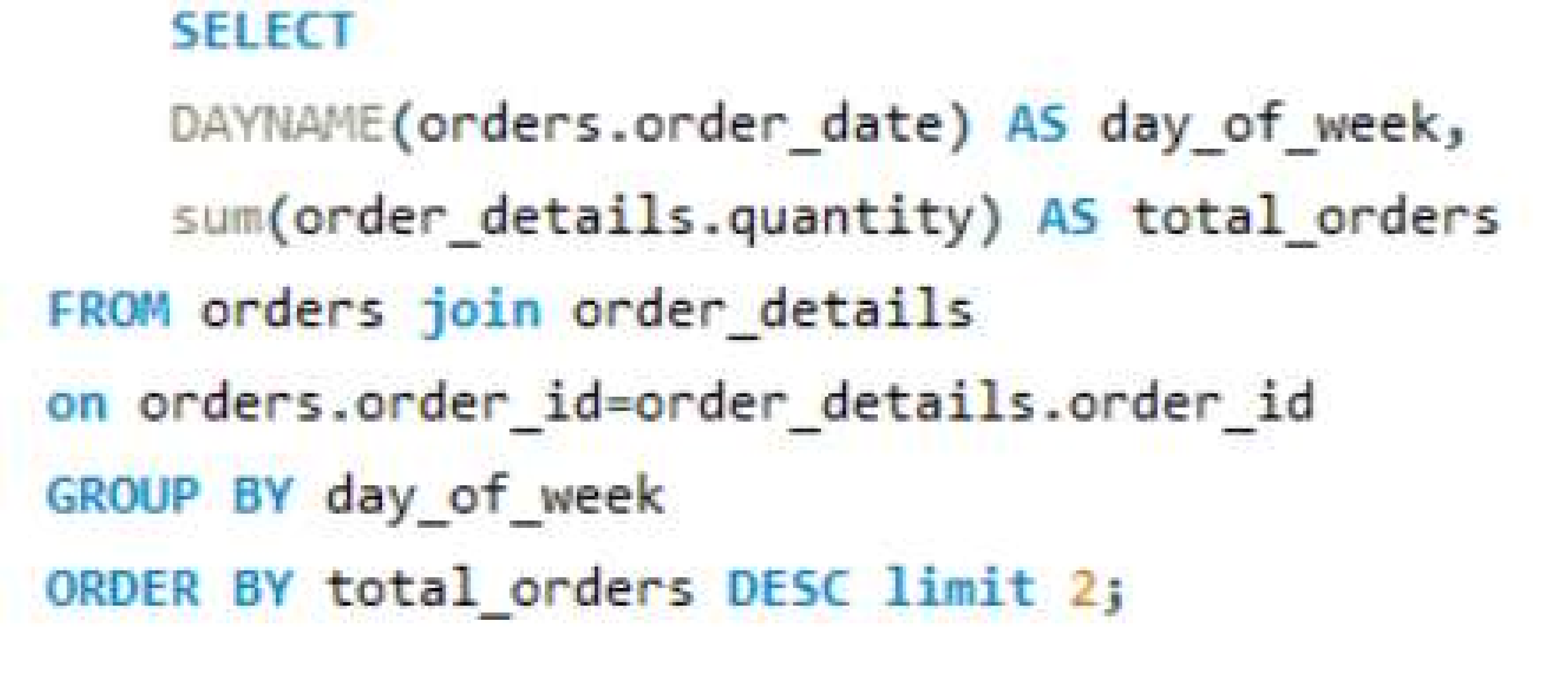


```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_orders_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

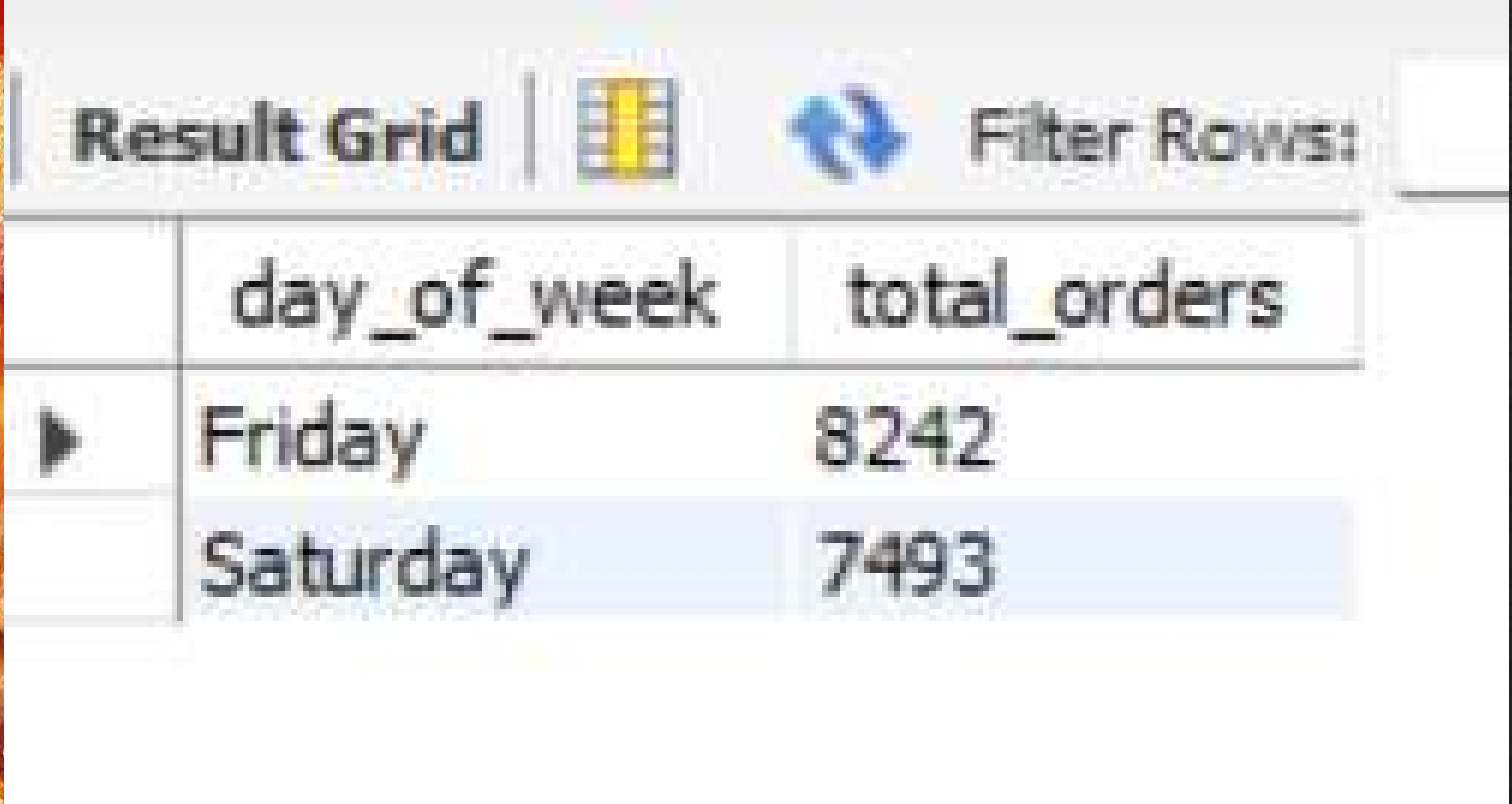


```
avg_orders_per_day
138
```

Identify top 2 ordering days by calculating total orders per day of the week.

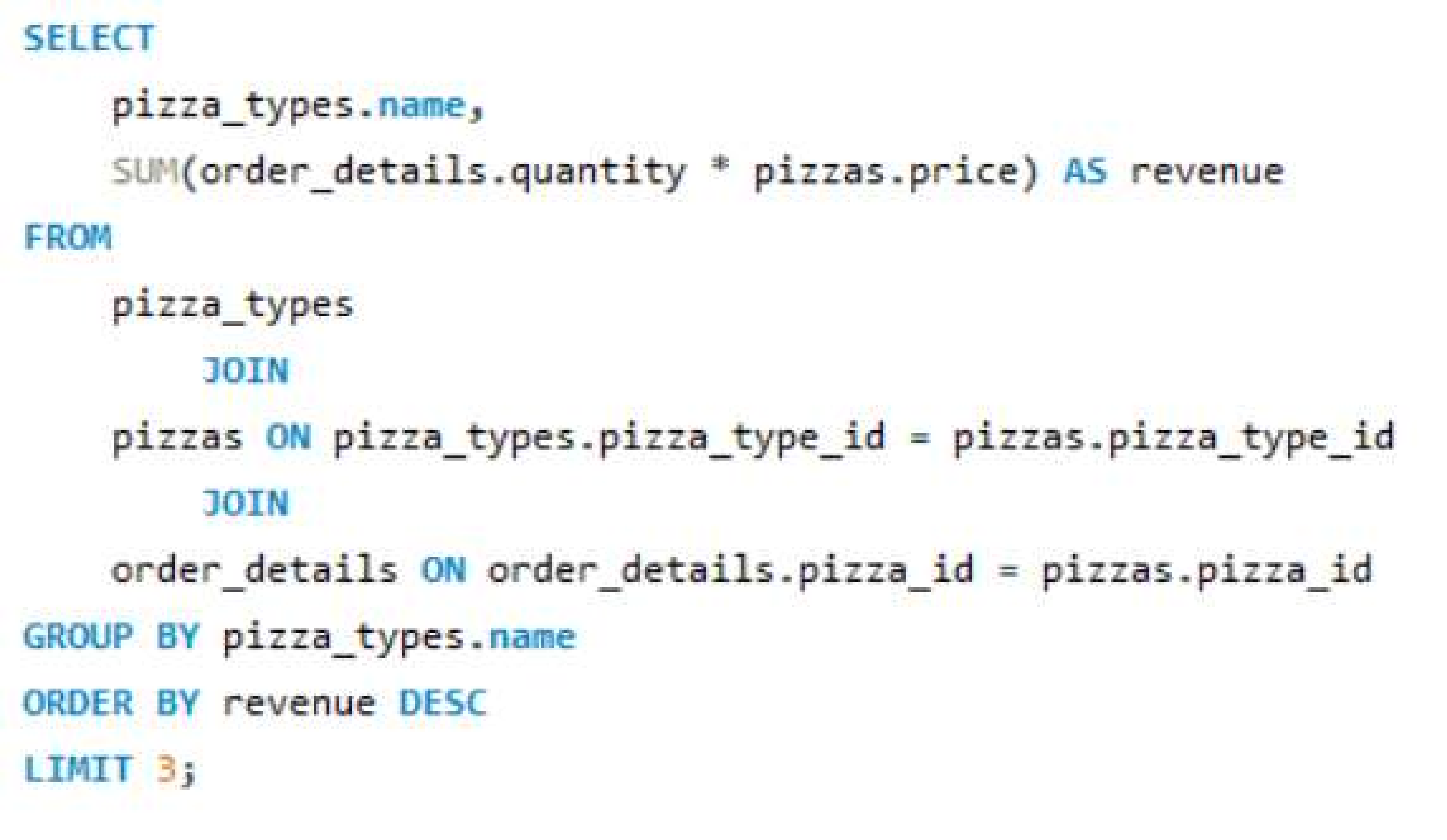


```
    SELECT
    DAYNAME(orders.order_date) AS day_of_week,
    sum(order_details.quantity) AS total_orders
FROM orders join order_details
on orders.order_id=order_details.order_id
GROUP BY day_of_week
ORDER BY total_orders DESC limit 2;
```

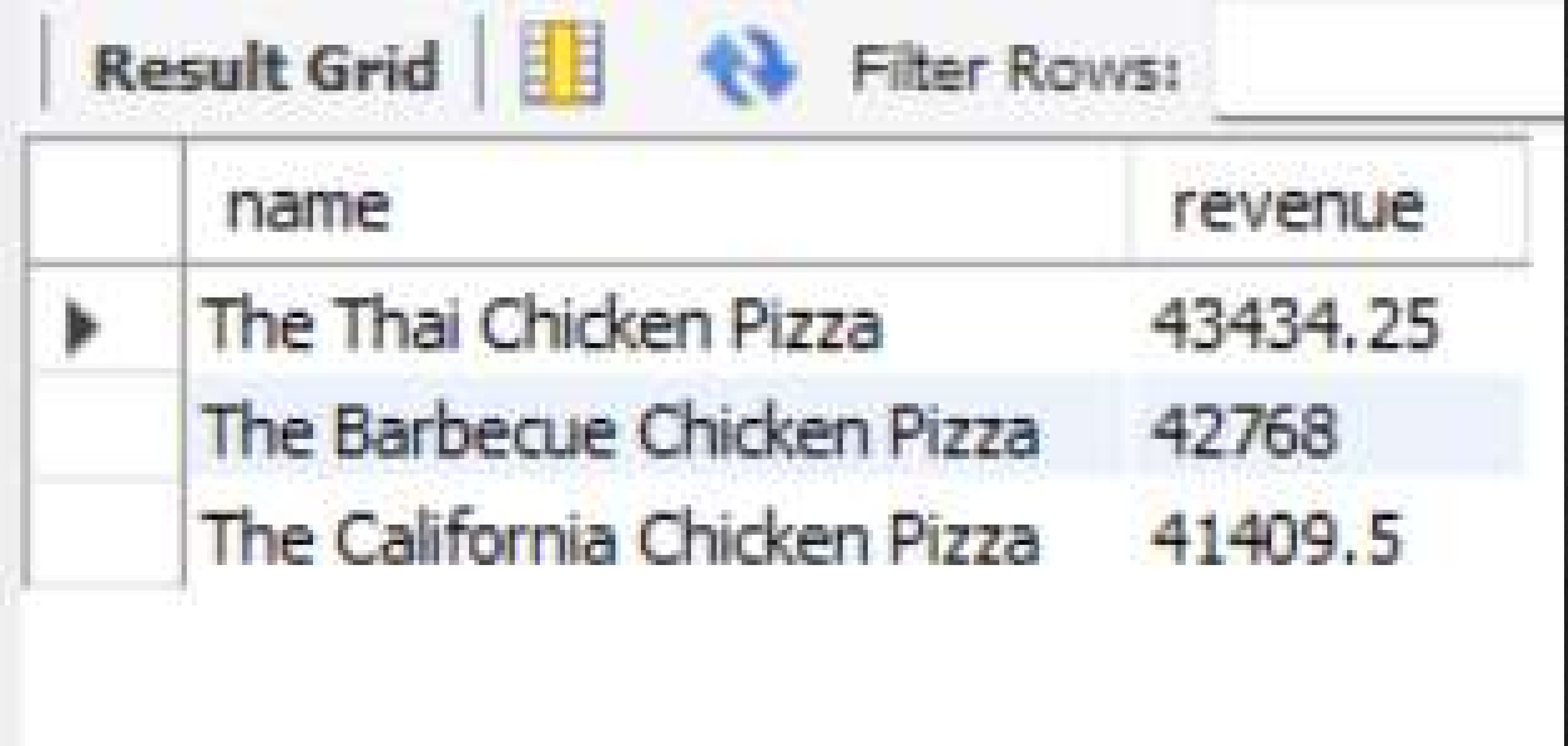


day_of_week	total_orders
Friday	8242
Saturday	7493

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

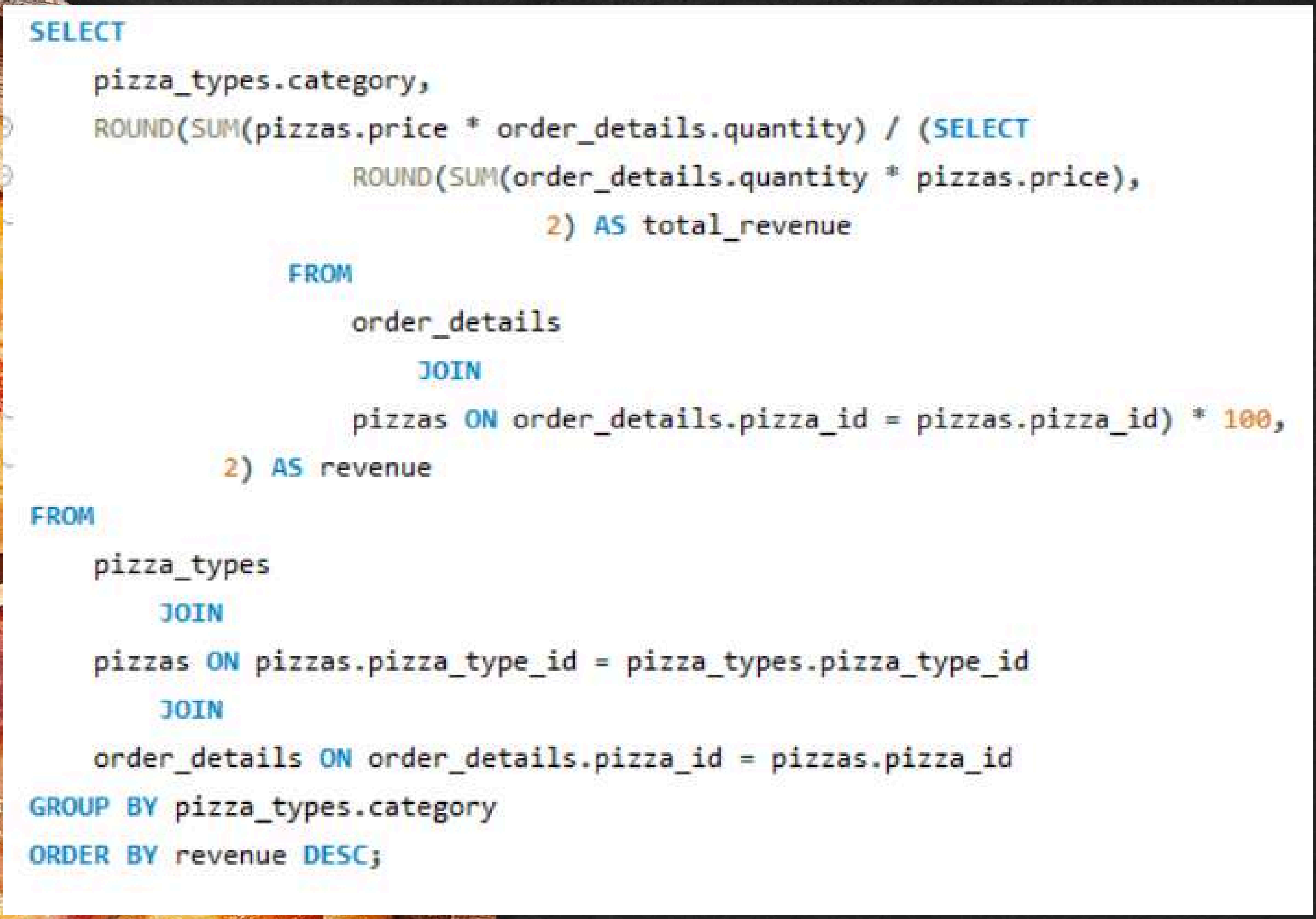


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

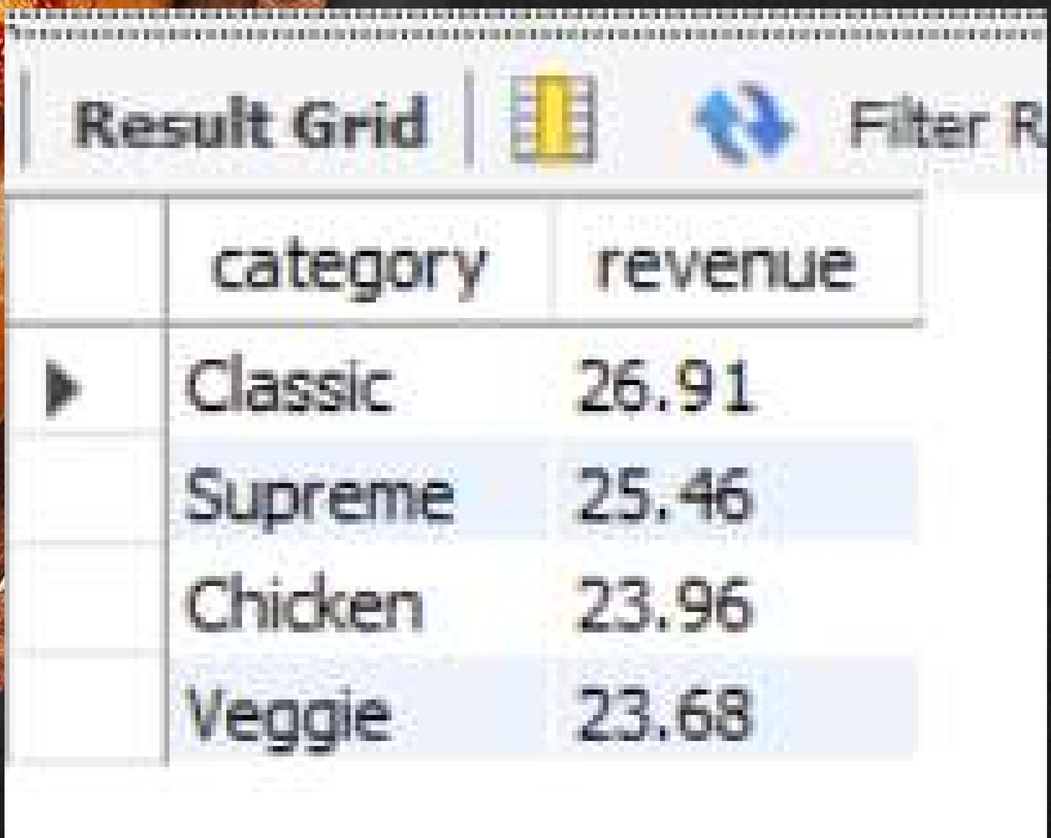


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue.

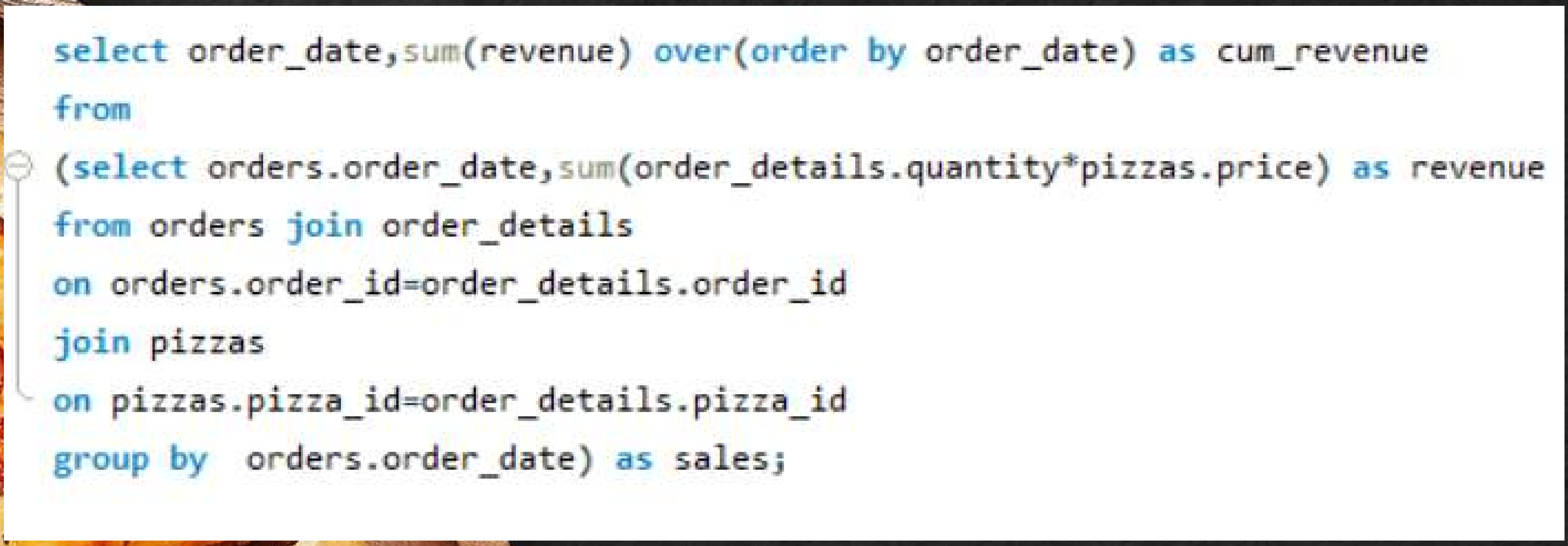


```
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_revenue
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.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;
```

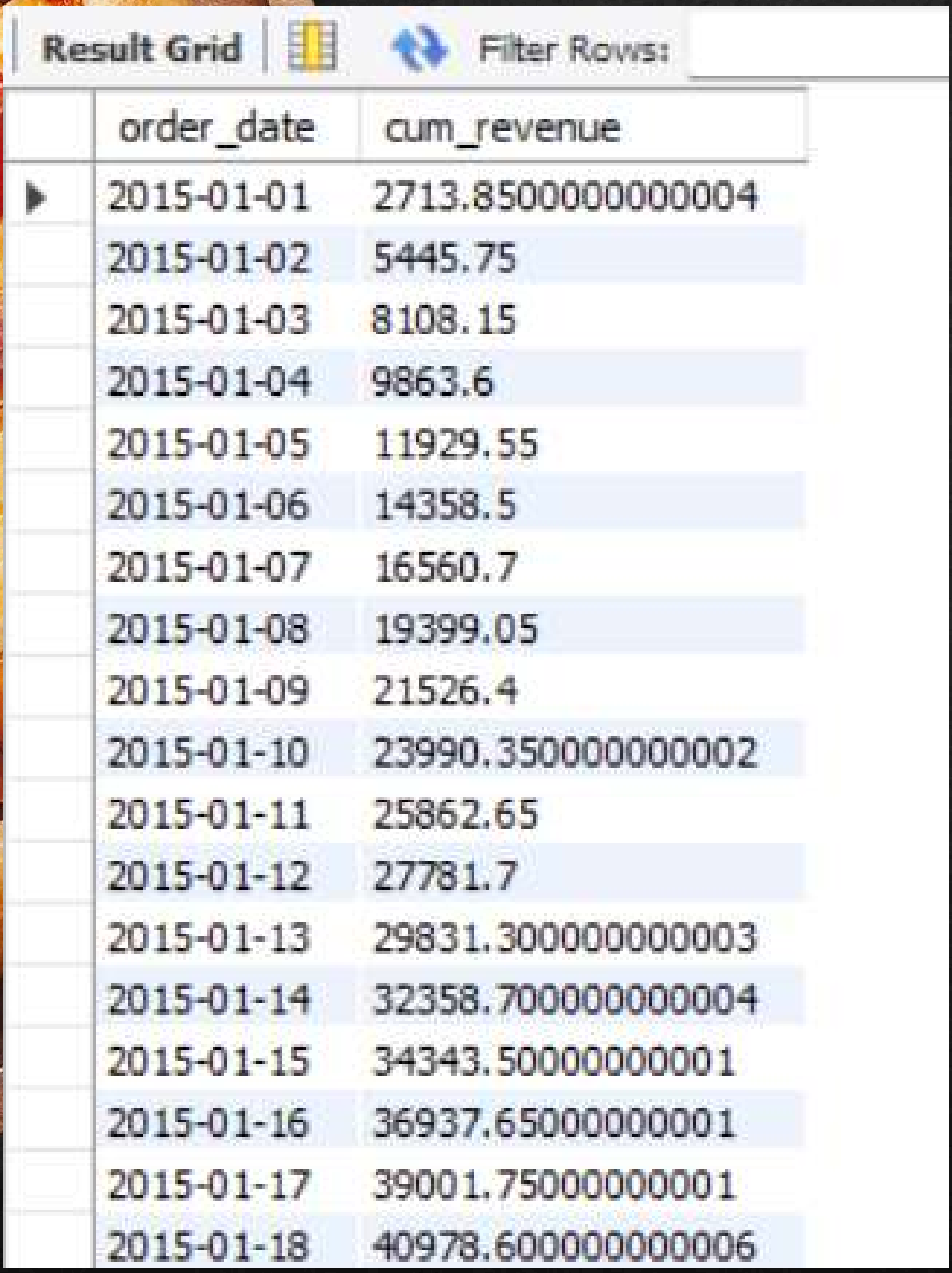


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

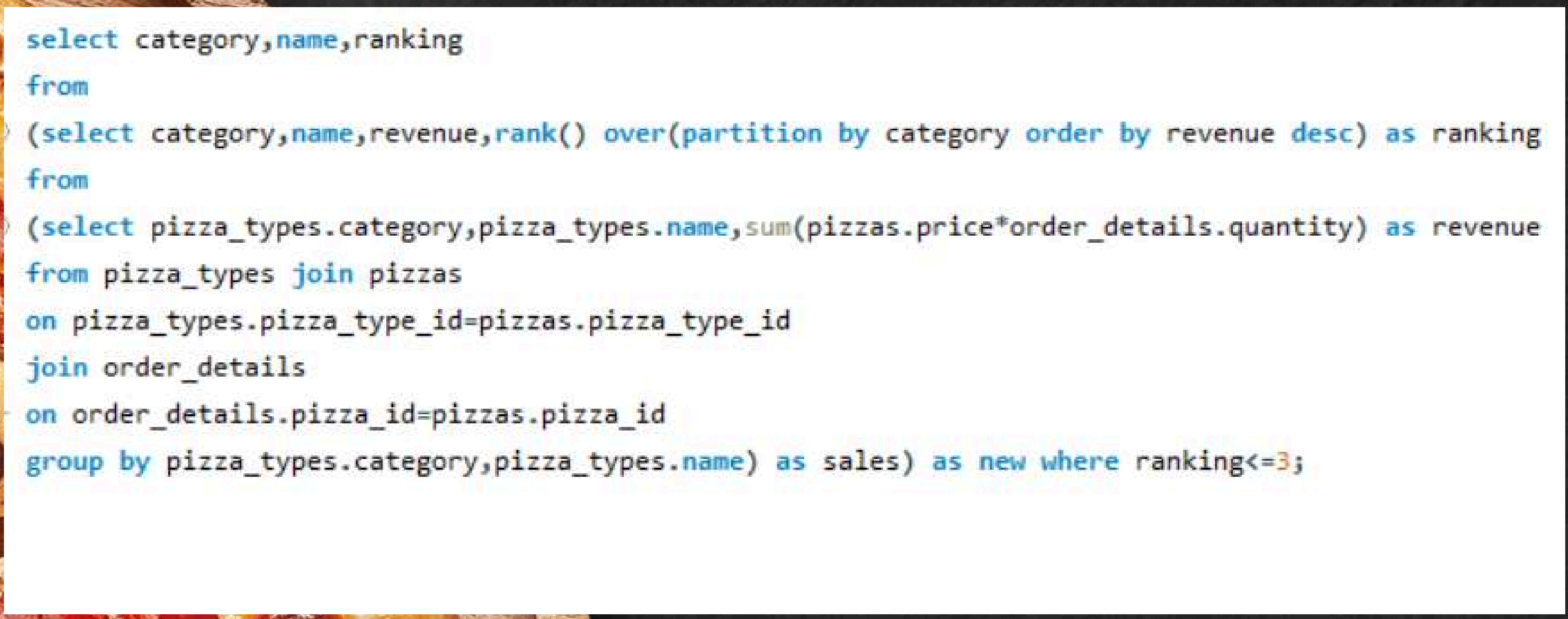


```
select order_date,sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,sum(order_details.quantity*pizzas.price) as revenue
from orders join order_details
on orders.order_id=order_details.order_id
join pizzas
on pizzas.pizza_id=order_details.pizza_id
group by  orders.order_date)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

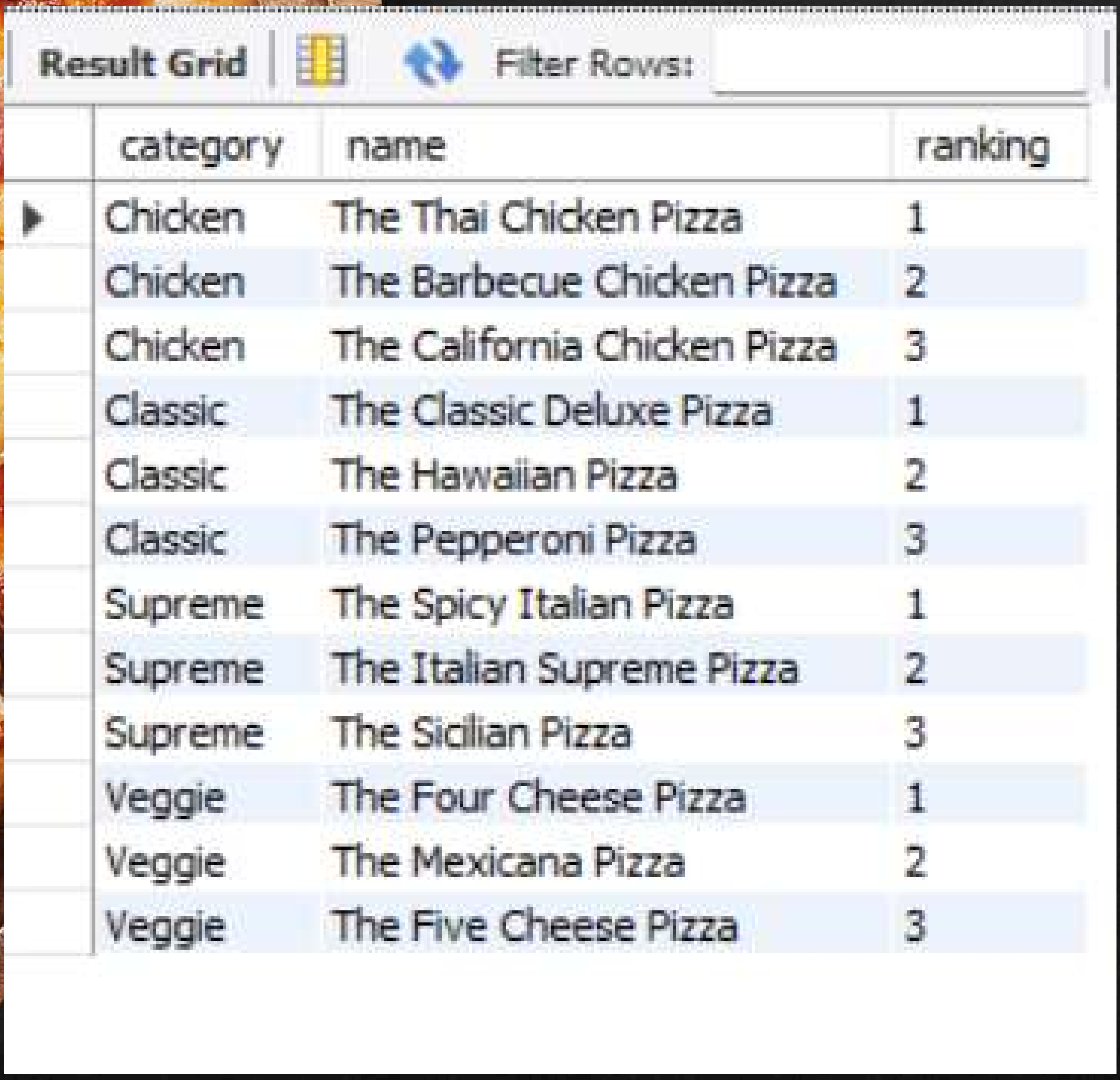


order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006

Determine the top 3 most ordered pizza types based on revenue for each pizza category.



```
select category,name,ranking
from
(select category,name,revenue,rank() over(partition by category order by revenue desc) as ranking
from
(select pizza_types.category,pizza_types.name,sum(pizzas.price*order_details.quantity) 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 sales) as new where ranking<=3;
```



category	name	ranking
Chicken	The Thai Chicken Pizza	1
Chicken	The Barbecue Chicken Pizza	2
Chicken	The California Chicken Pizza	3
Classic	The Classic Deluxe Pizza	1
Classic	The Hawaiian Pizza	2
Classic	The Pepperoni Pizza	3
Supreme	The Spicy Italian Pizza	1
Supreme	The Italian Supreme Pizza	2
Supreme	The Sicilian Pizza	3
Veggie	The Four Cheese Pizza	1
Veggie	The Mexicana Pizza	2
Veggie	The Five Cheese Pizza	3

THANK YOU