PIZZA SALES ANALYSIS

PIZZA HUT

PIZZA HUT

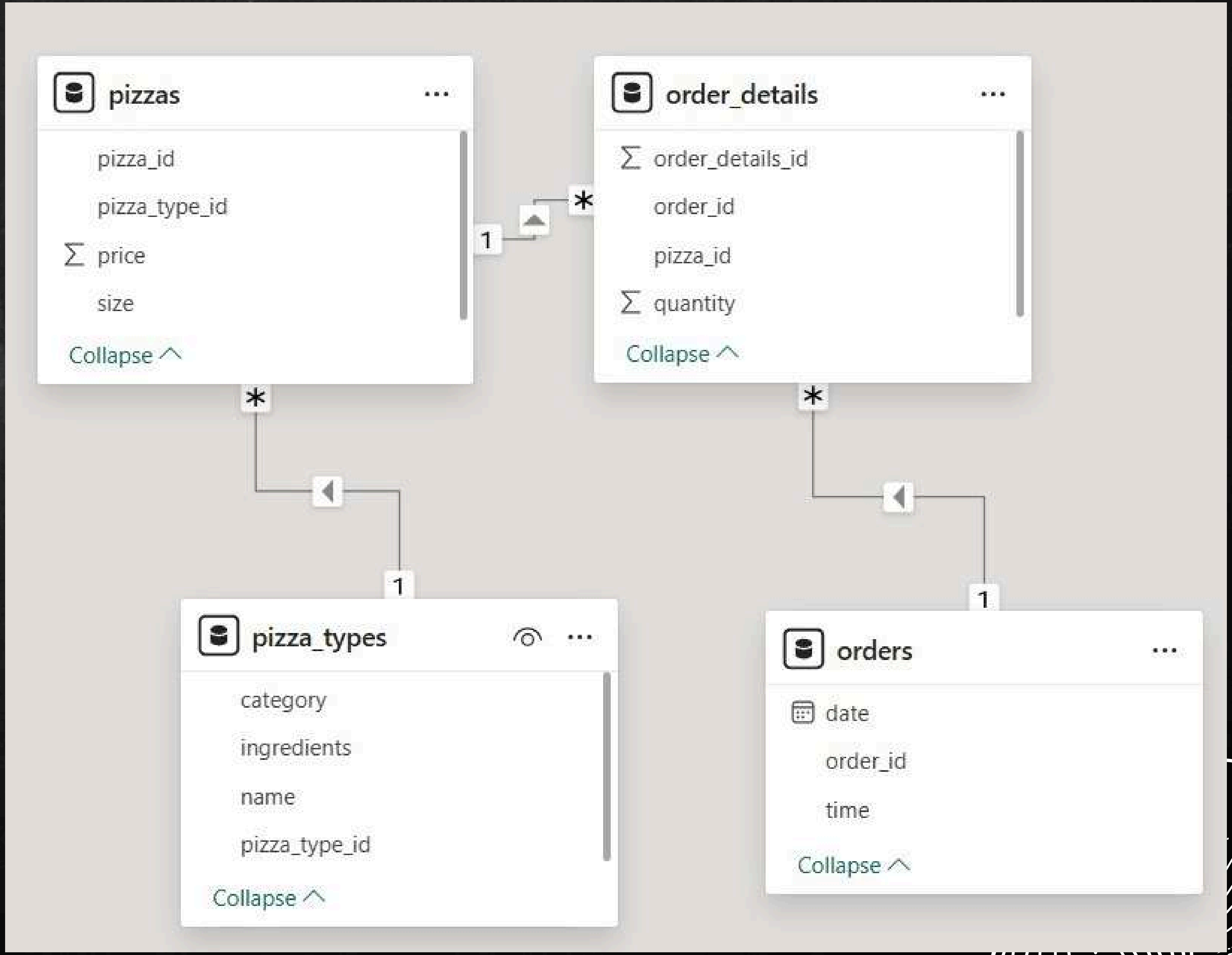
INTRODUCTION

HELLO!

MY NAME IS SHITAL,
IN THIS PROJECT I HAVE UTILIZED SQL QUERIES AND POWER BI TO SOLVE QUESTIONS THAT WERE RELATED TO PIZZA SALES.

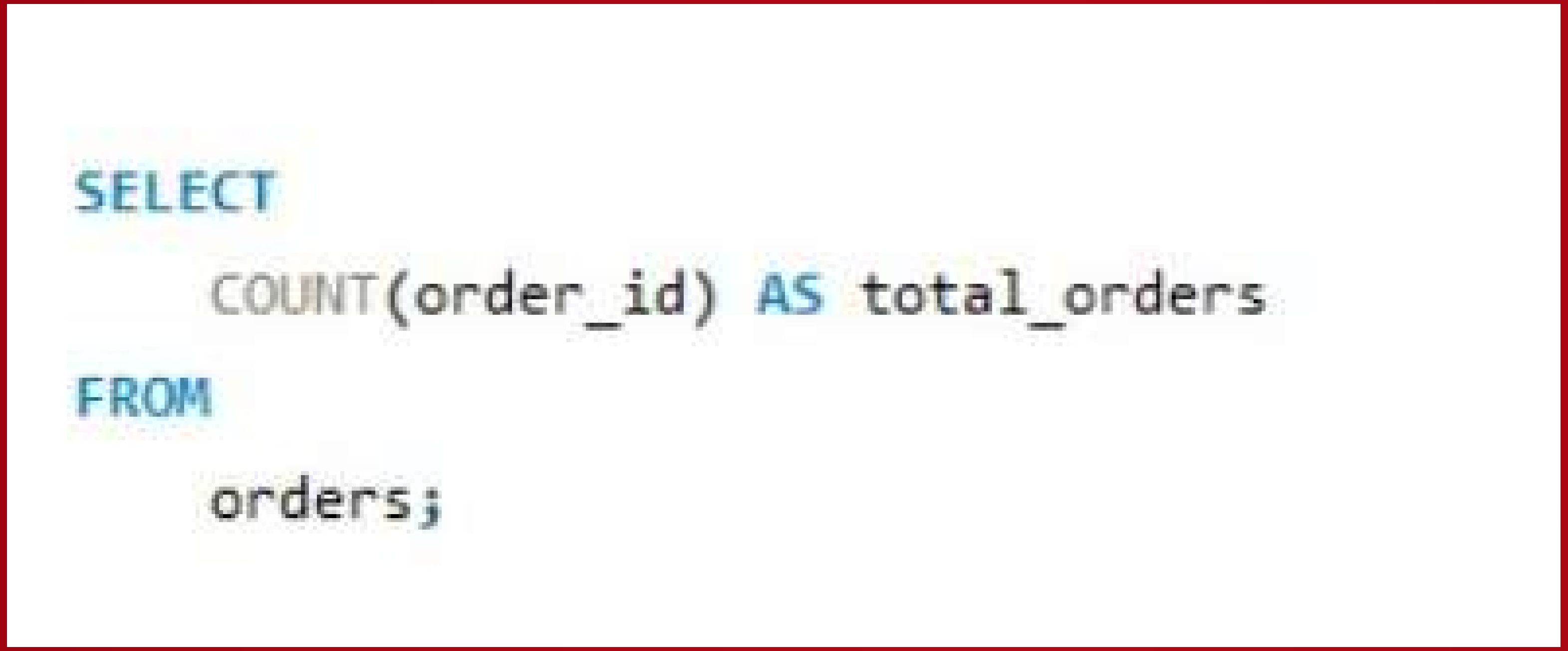
PIZZA HUT

DATA MODLING

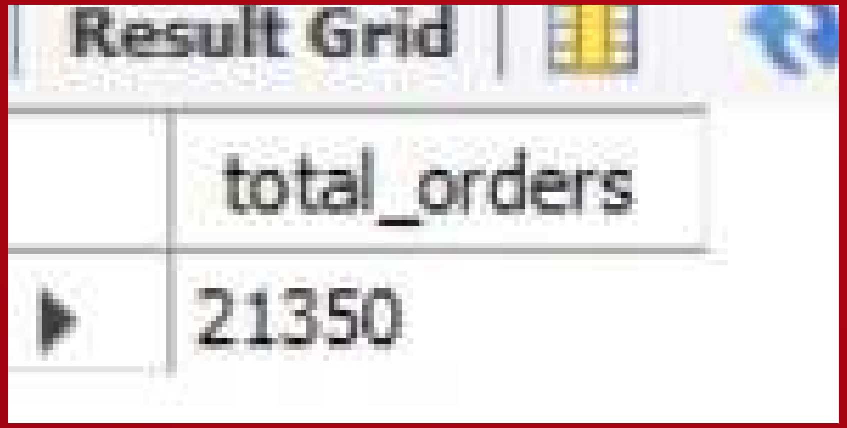


PIZZA HUT

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



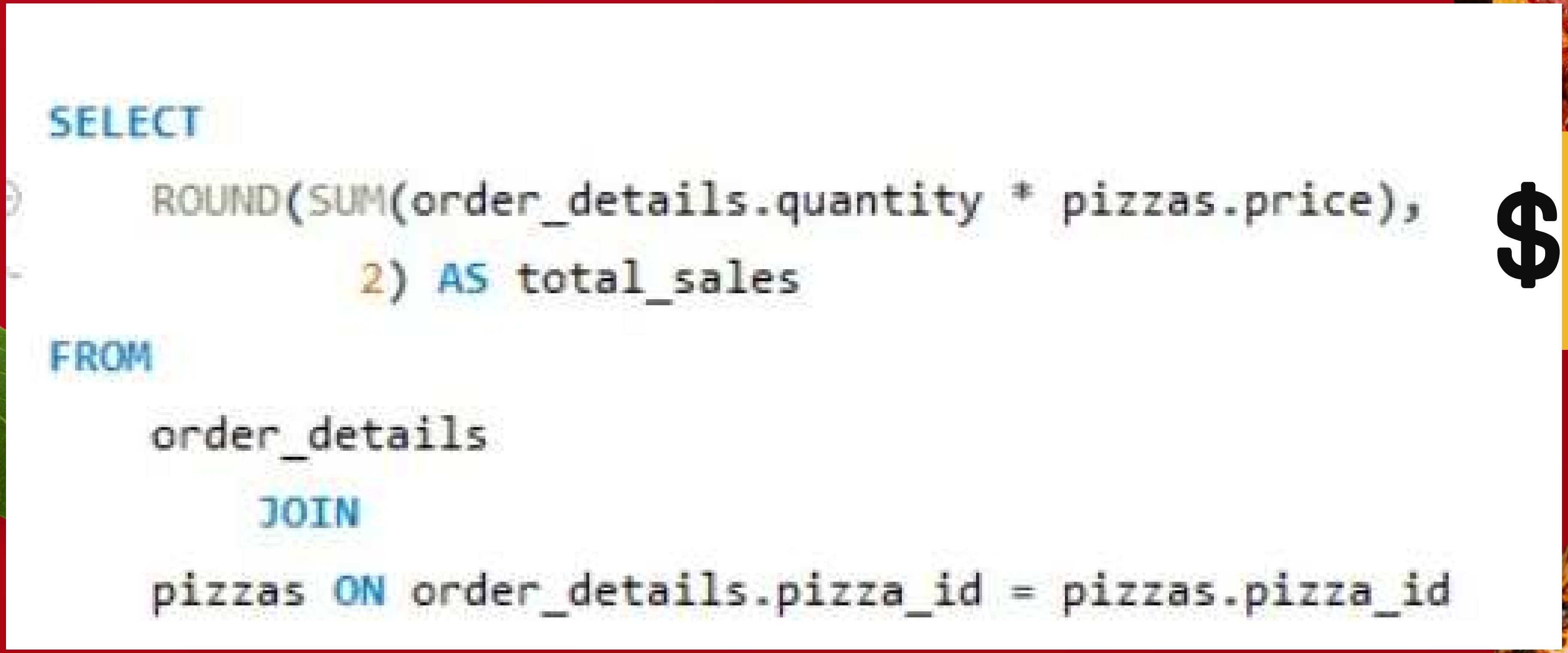
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



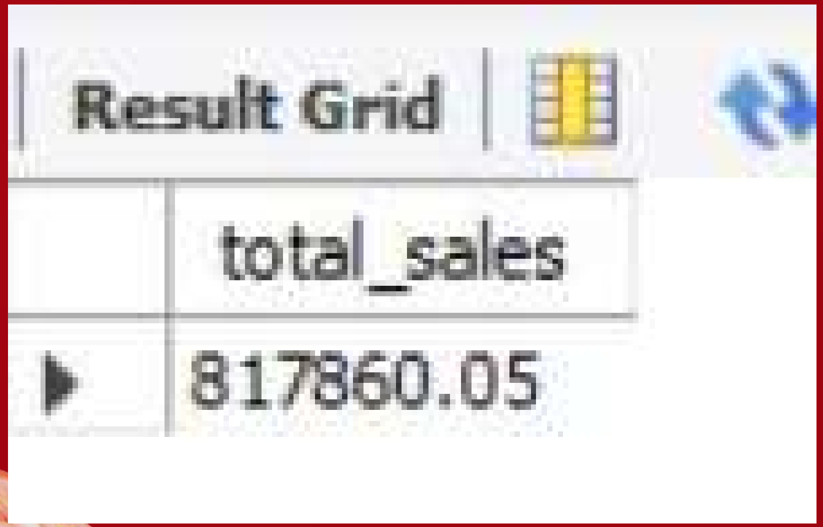
```
total_orders
21350
```

PIZZA HUT

2. 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 order_details.pizza_id = pizzas.pizza_id
```

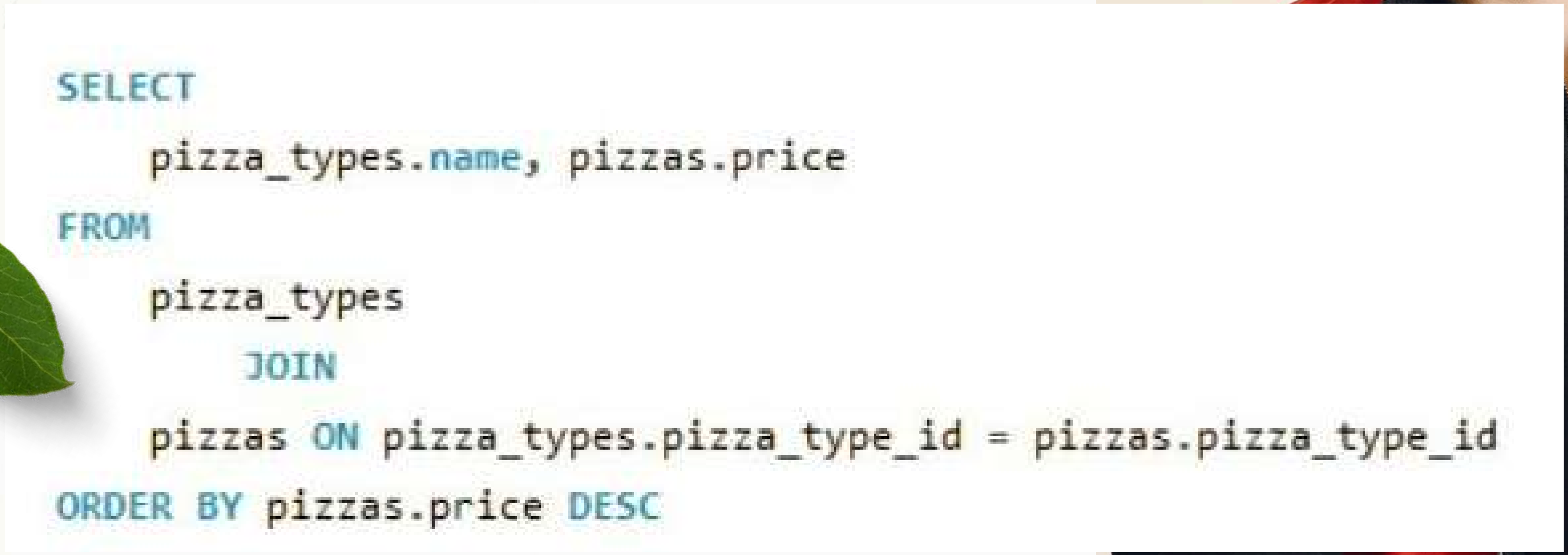


Result Grid

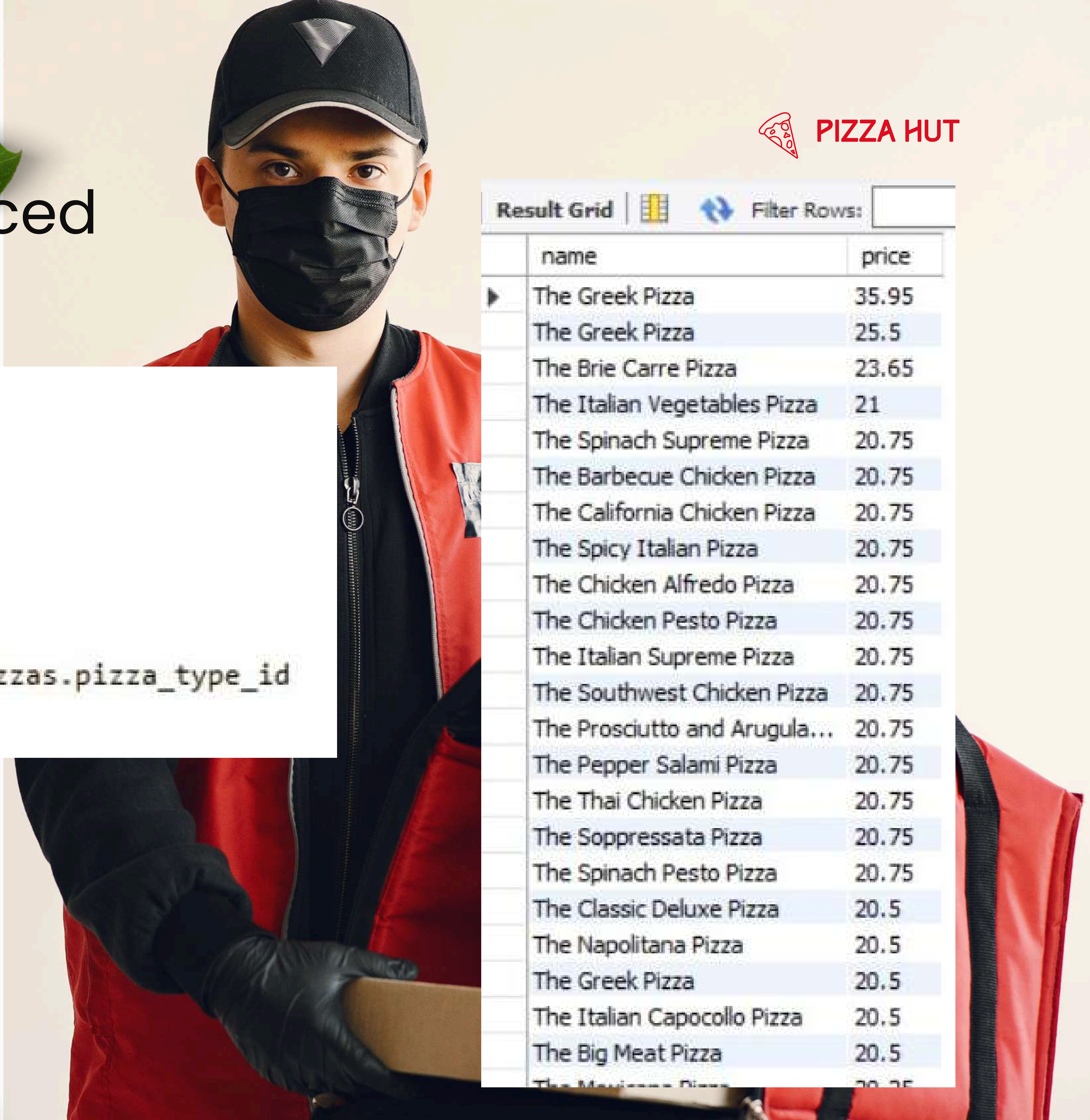
```
total_sales
817860.05
```

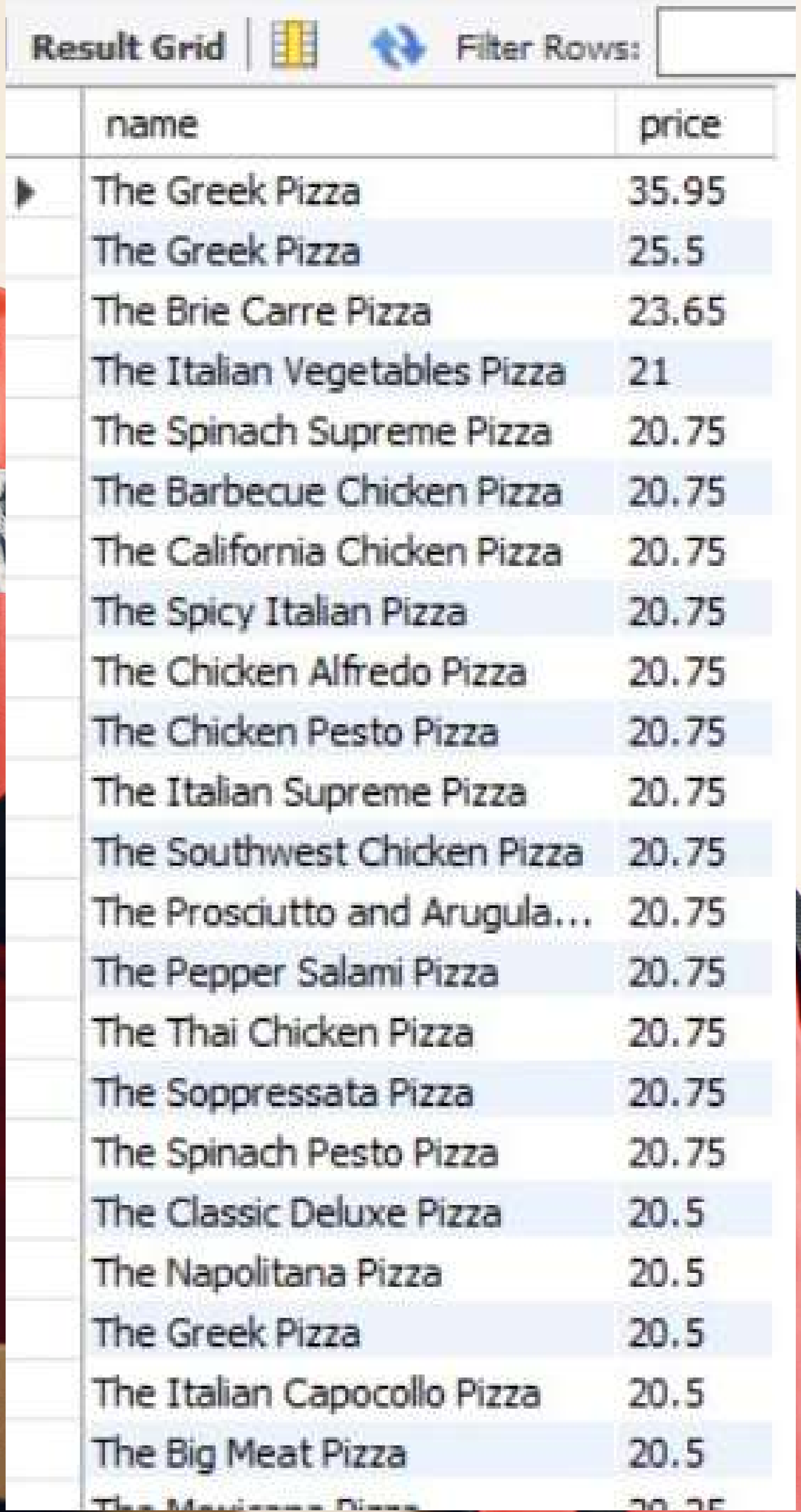
PIZZA HUT

3. 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
```



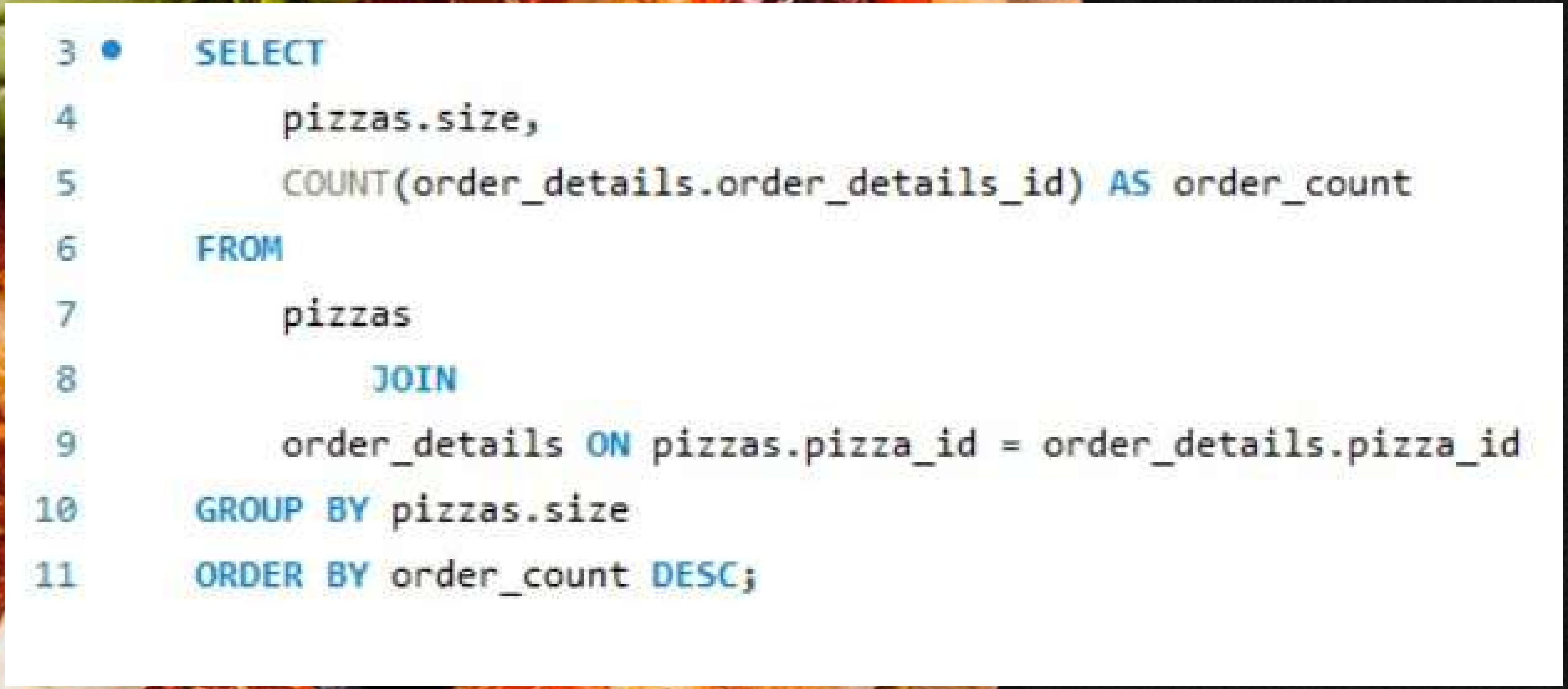


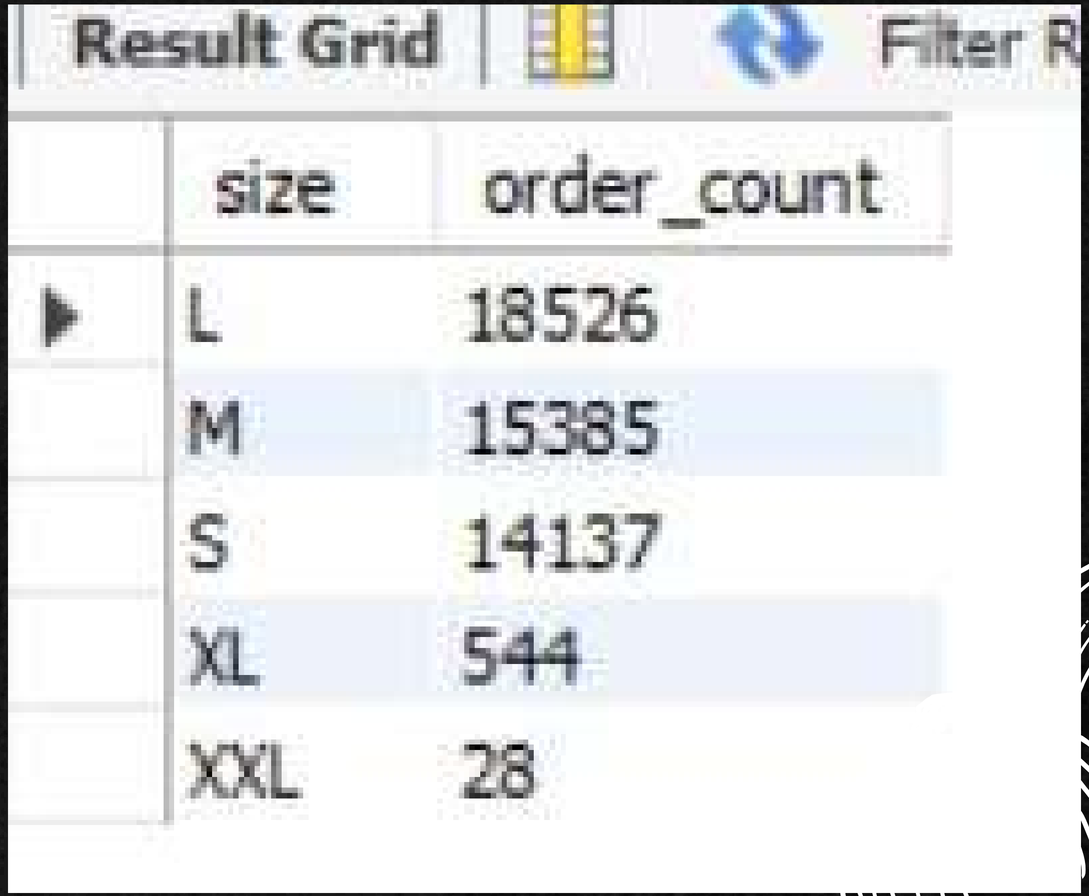
Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95
The Greek Pizza	25.5
The Brie Carre Pizza	23.65
The Italian Vegetables Pizza	21
The Spinach Supreme Pizza	20.75
The Barbecue Chicken Pizza	20.75
The California Chicken Pizza	20.75
The Spicy Italian Pizza	20.75
The Chicken Alfredo Pizza	20.75
The Chicken Pesto Pizza	20.75
The Italian Supreme Pizza	20.75
The Southwest Chicken Pizza	20.75
The Prosciutto and Arugula...	20.75
The Pepper Salami Pizza	20.75
The Thai Chicken Pizza	20.75
The Soppressata Pizza	20.75
The Spinach Pesto Pizza	20.75
The Classic Deluxe Pizza	20.5
The Napolitana Pizza	20.5
The Greek Pizza	20.5
The Italian Capocollo Pizza	20.5
The Big Meat Pizza	20.5

PIZZA HUT

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

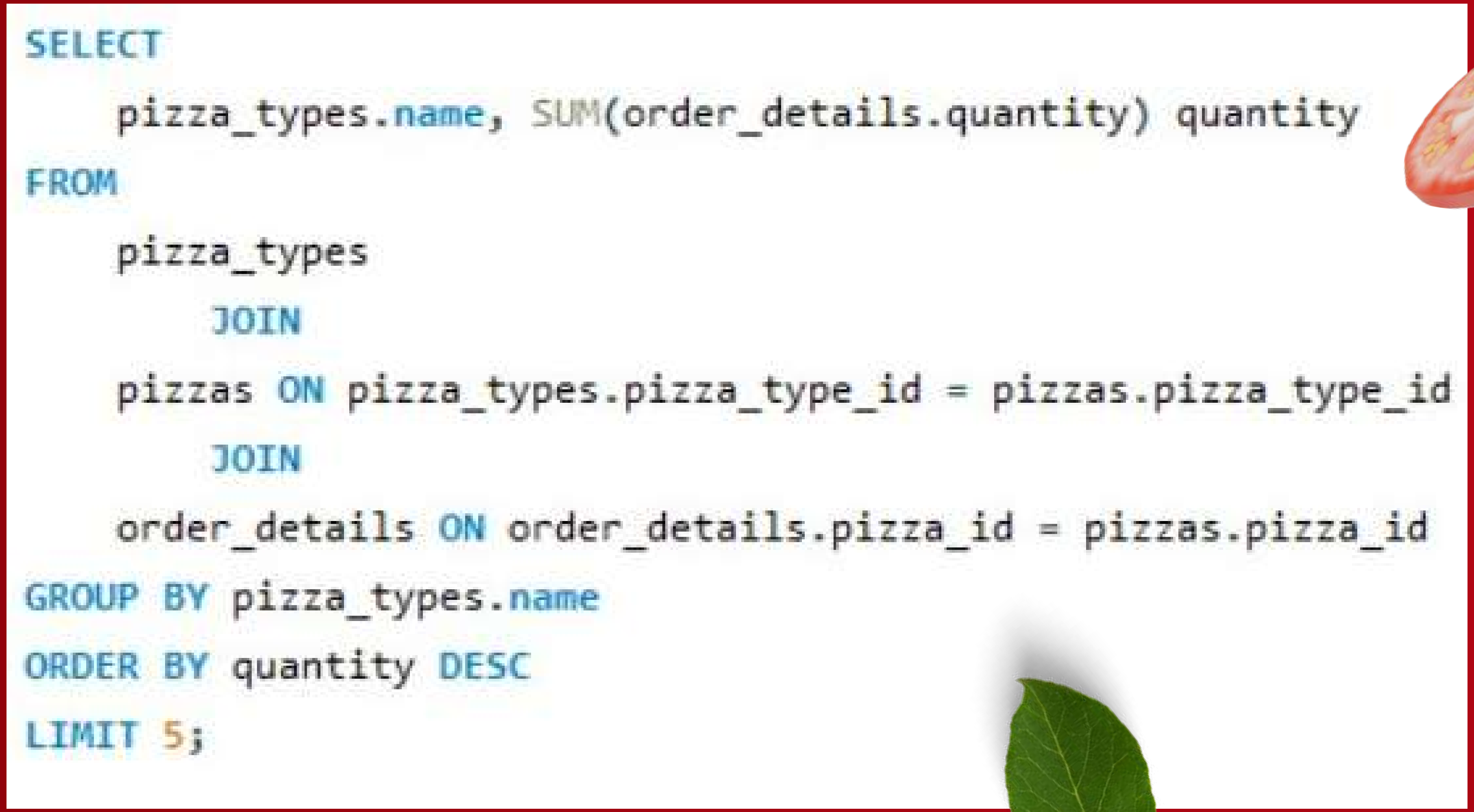

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



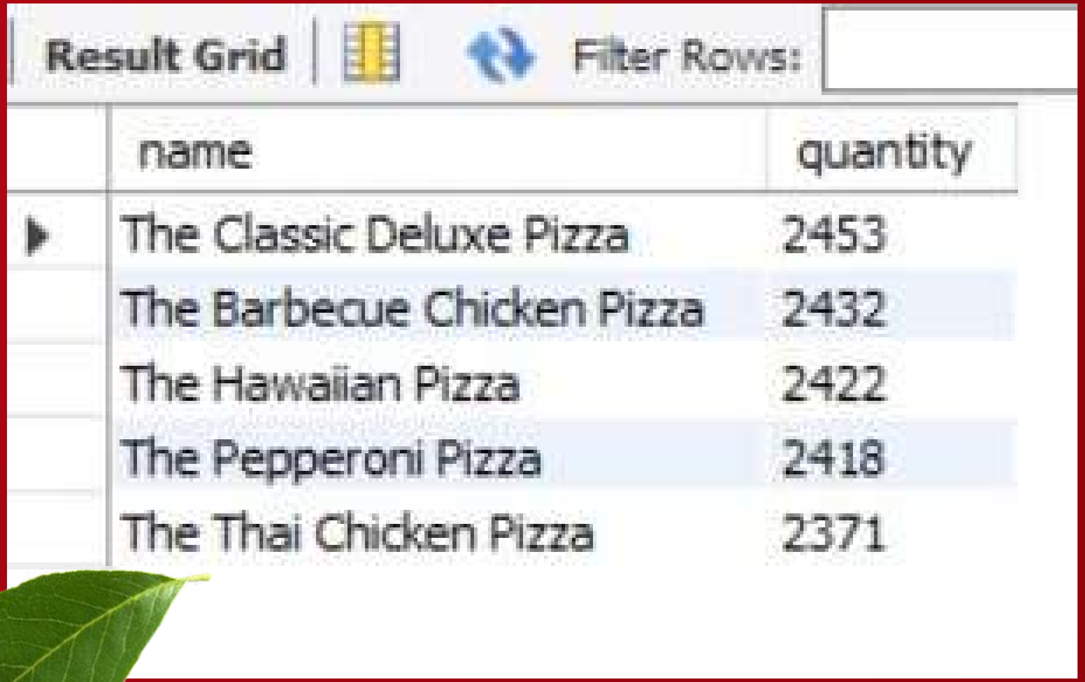
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

PIZZA HUT

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



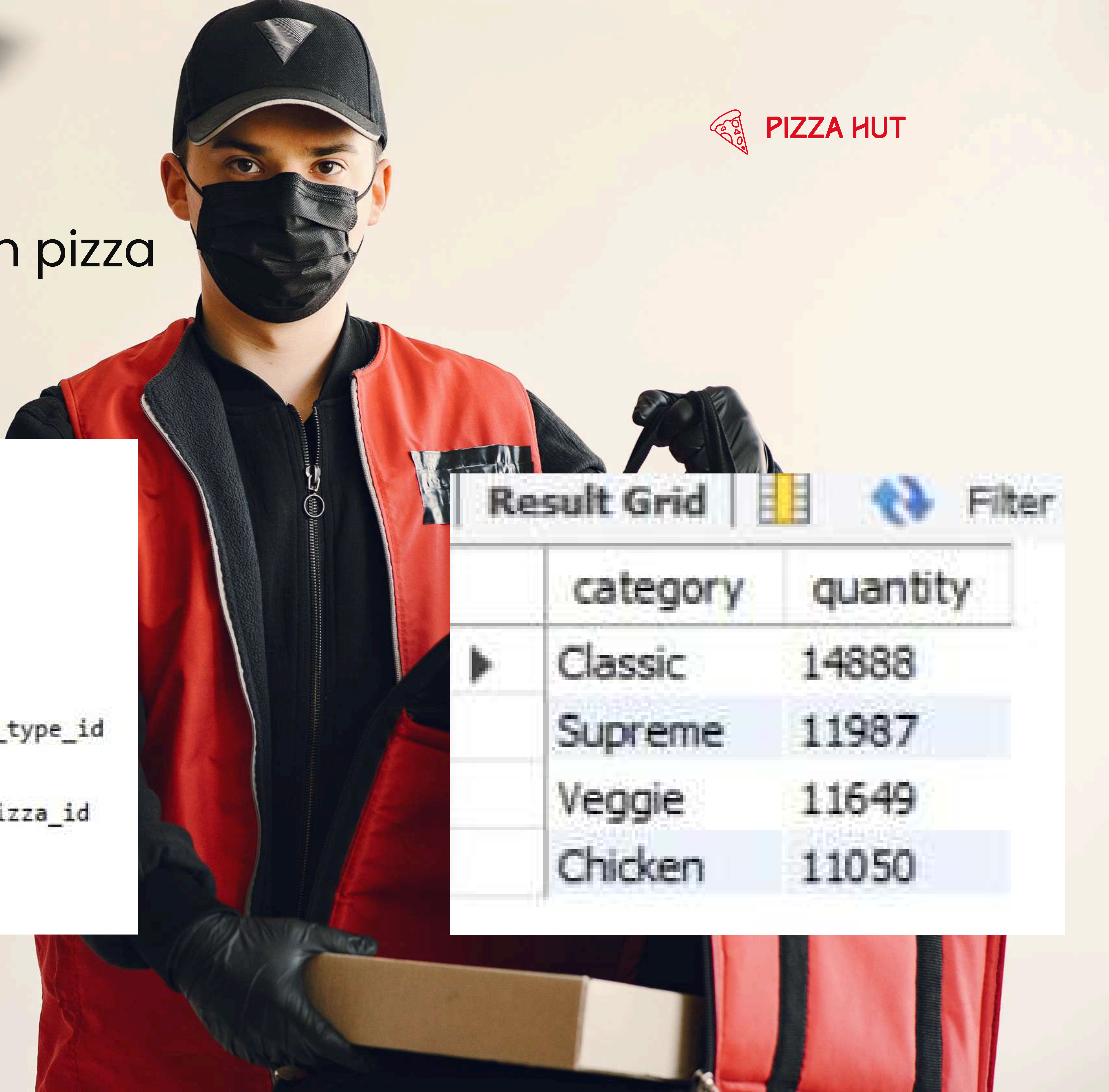
```
SELECT
    pizza_types.name, SUM(order_details.quantity) quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

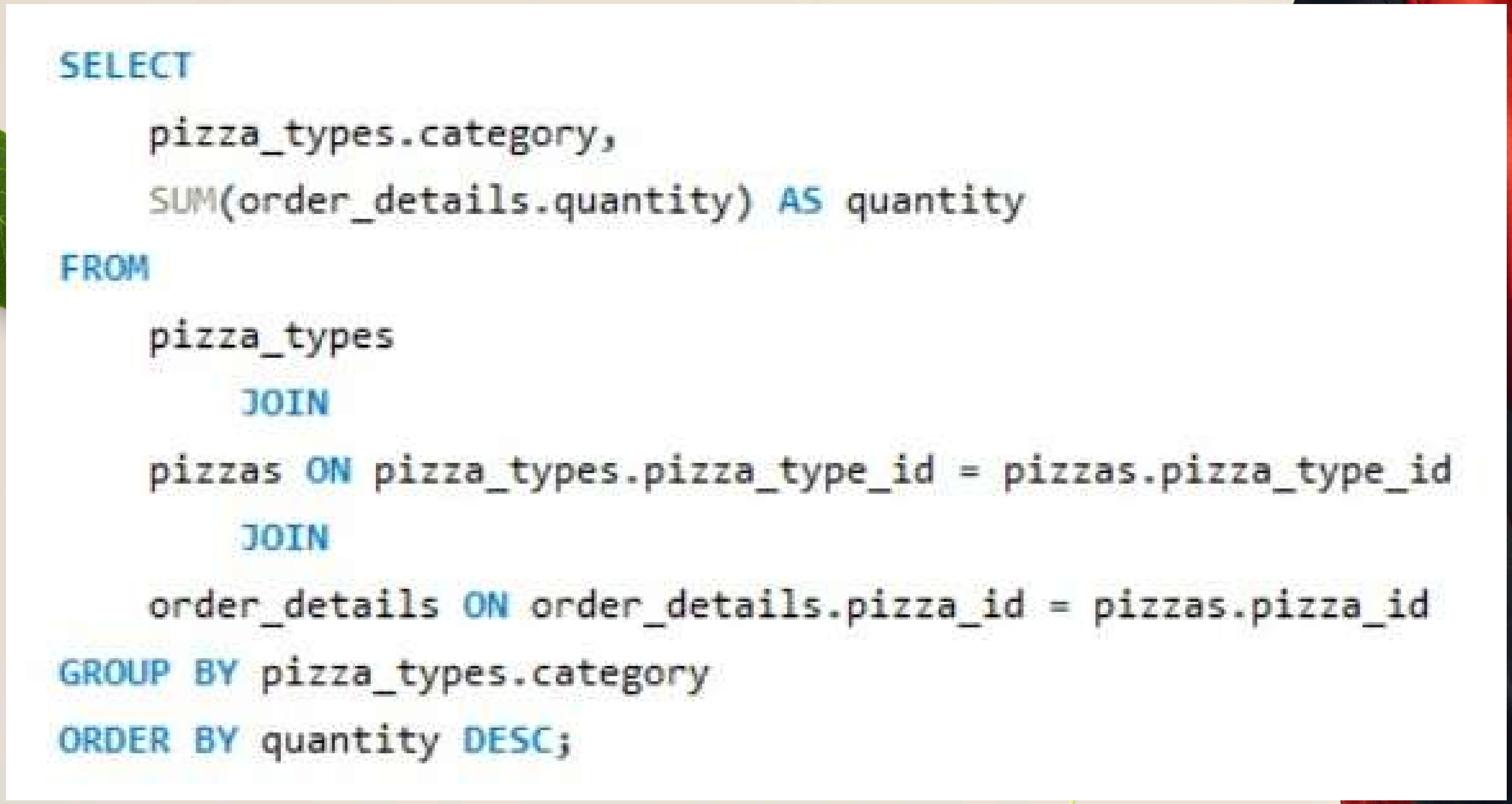


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

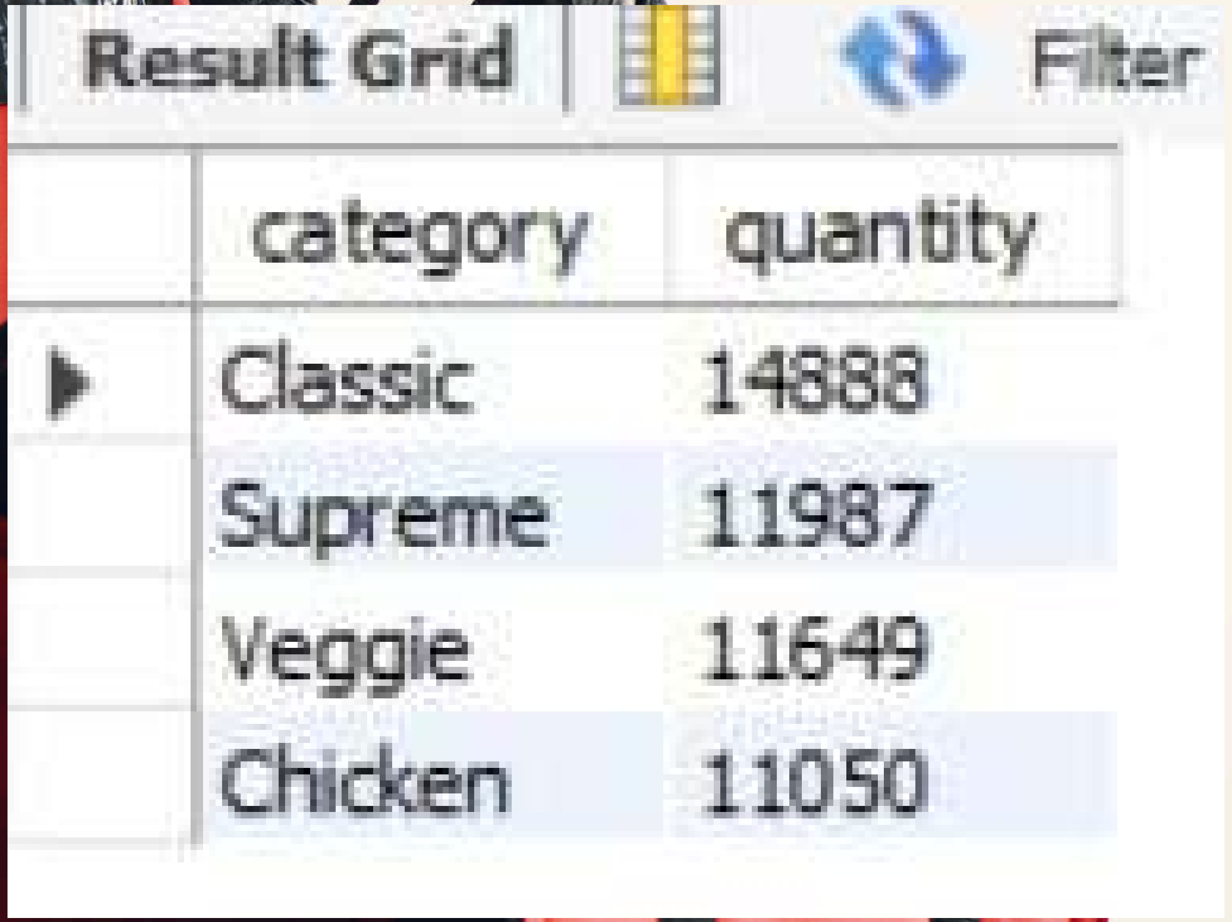
PIZZA HUT

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



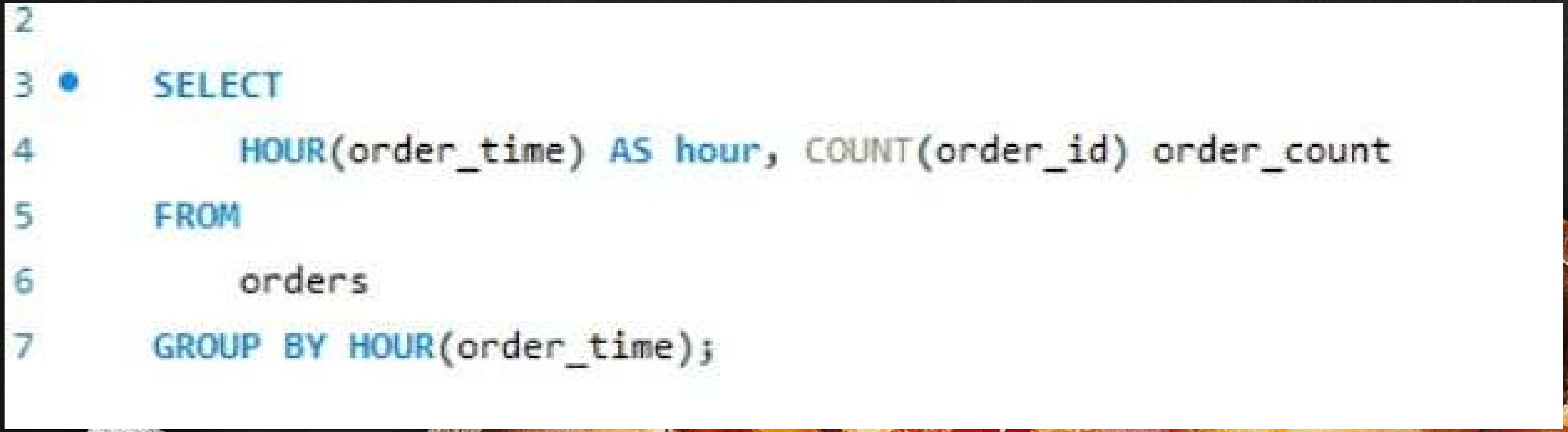


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

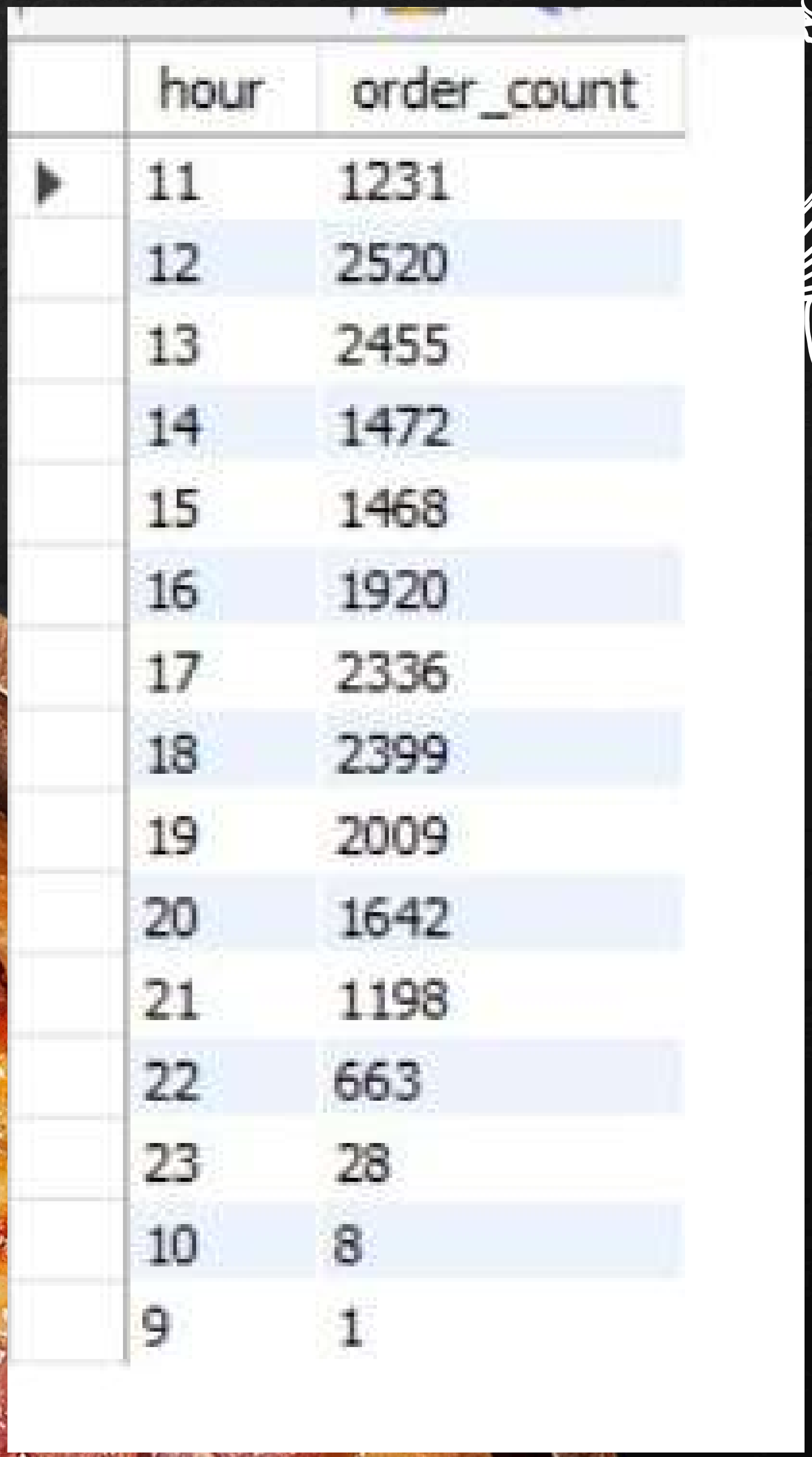


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

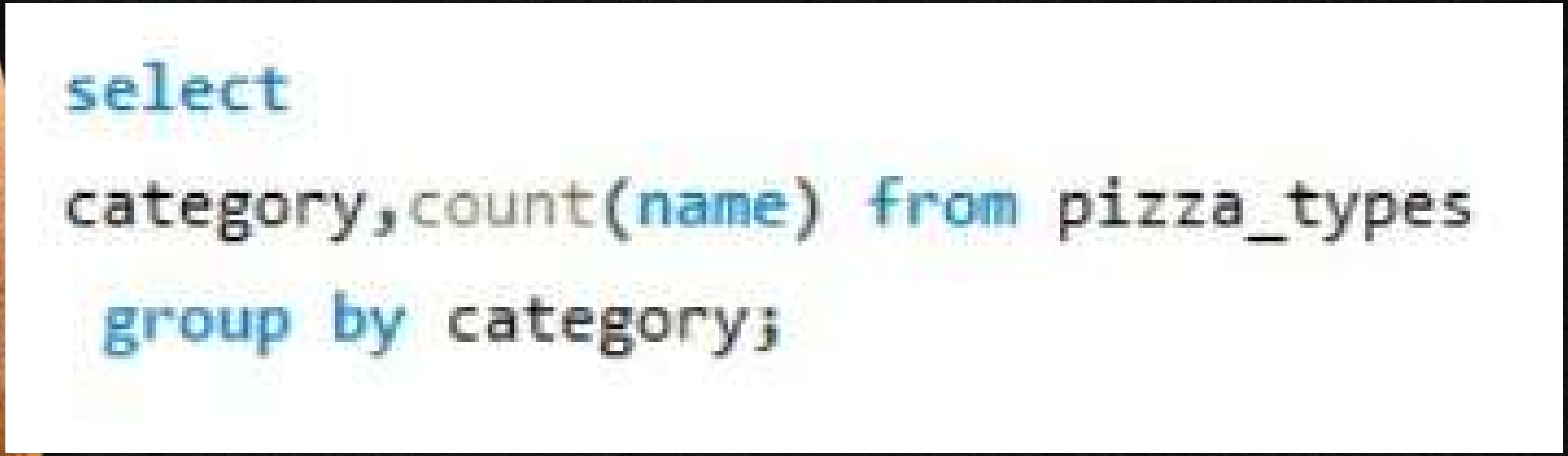


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

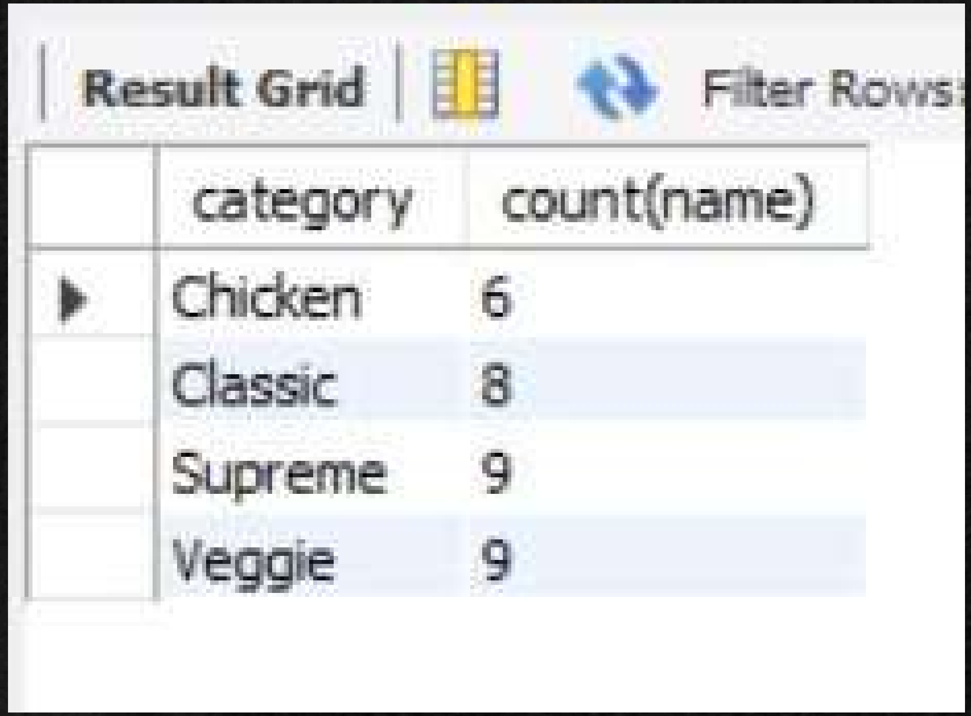


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

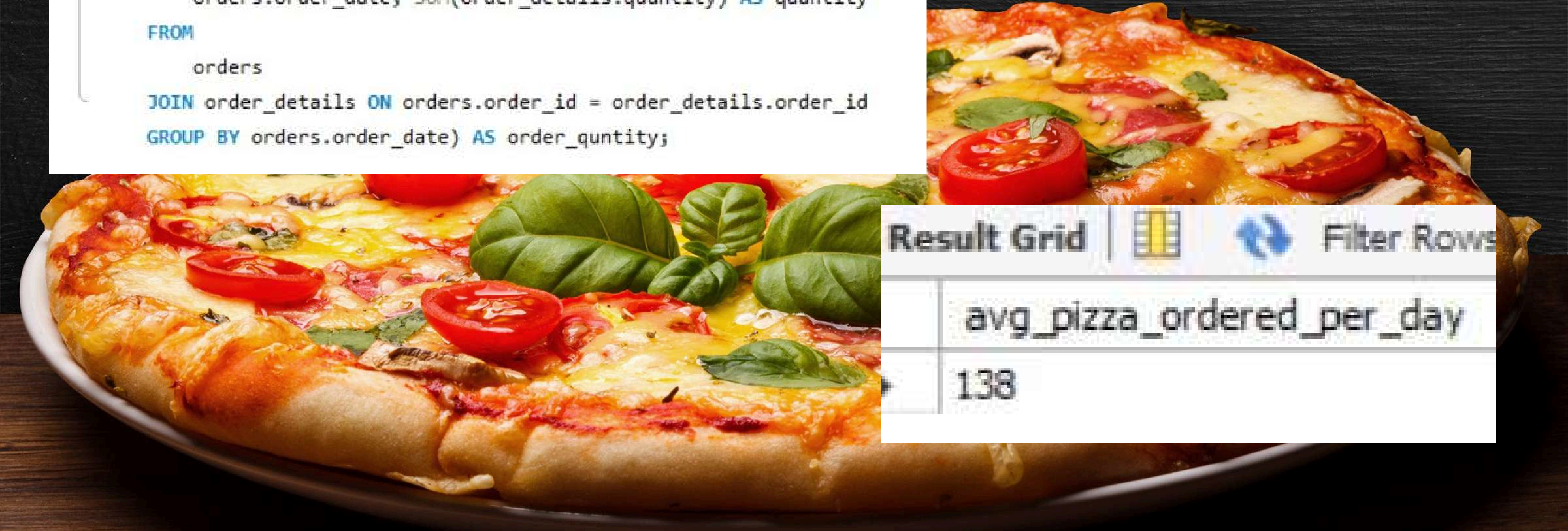


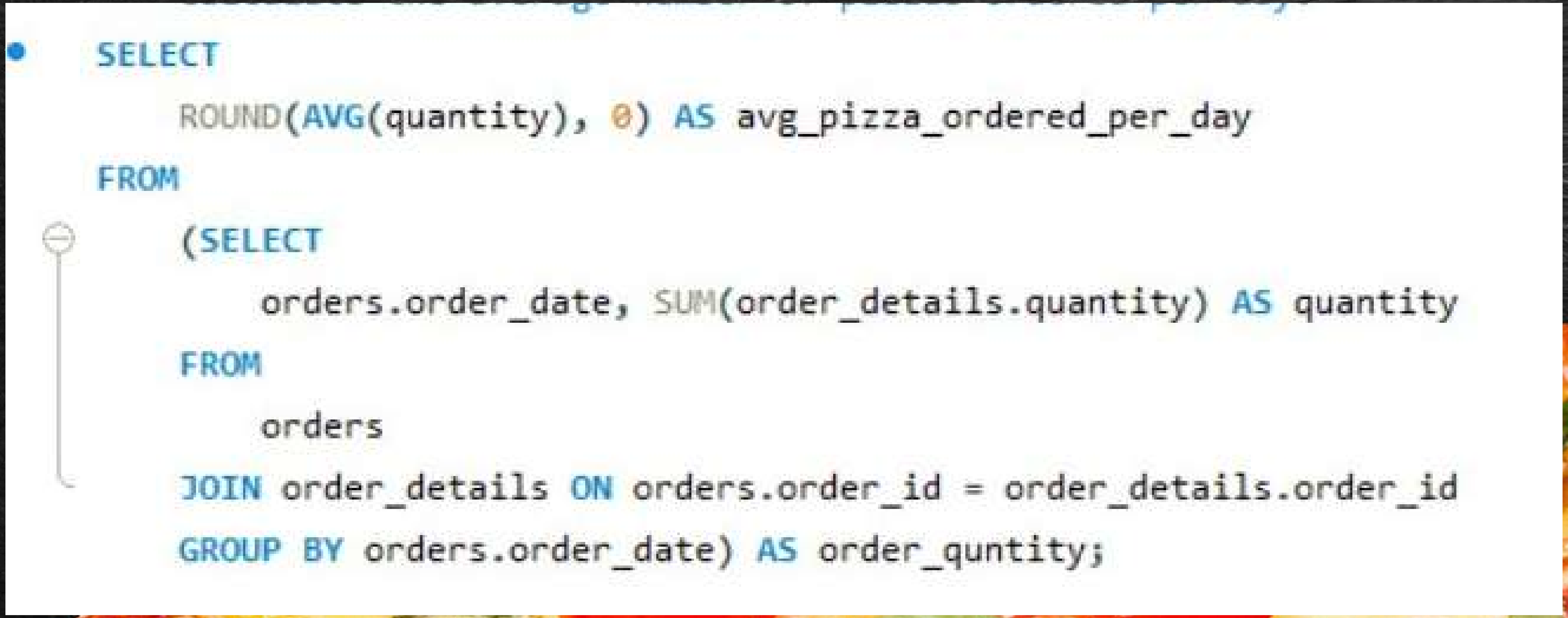
```
select
category,count(name) from pizza_types
 group by category;
```



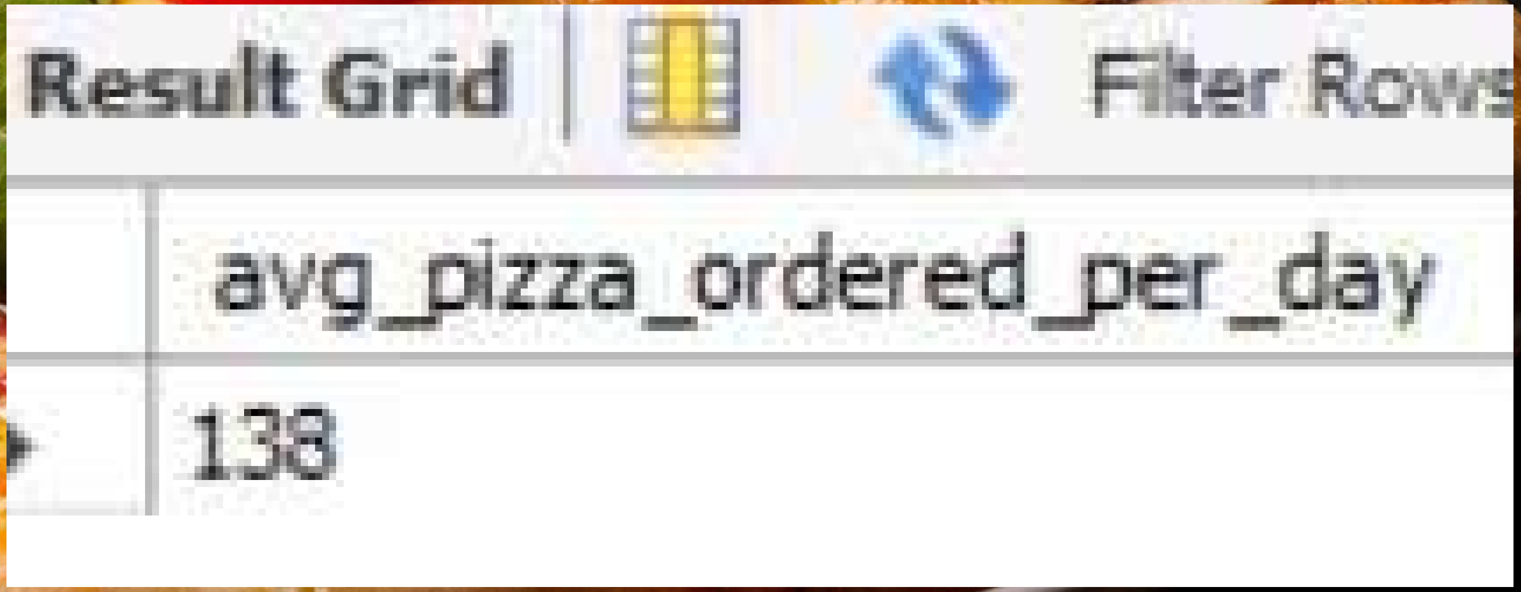
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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.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_quntity;
```

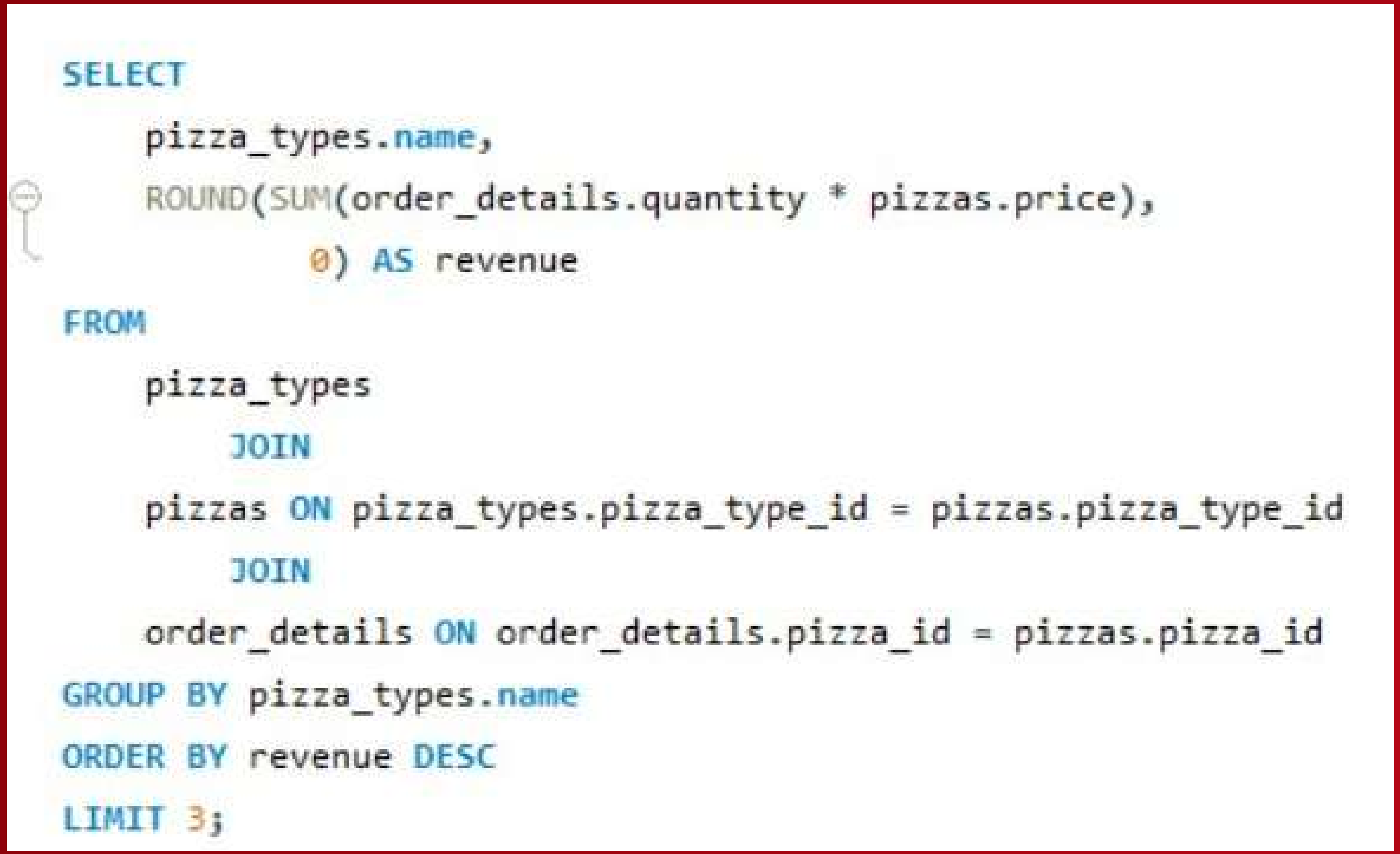


Result Grid | Filter Rows

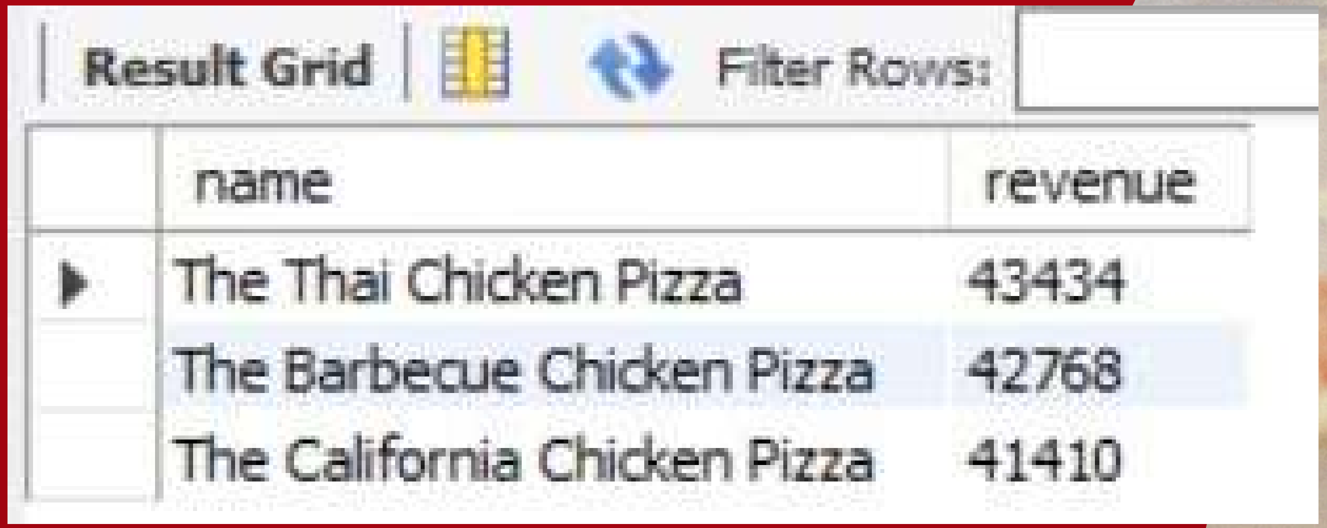
	avg_pizza_ordered_per_day
▸	138

PIZZA HUT

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



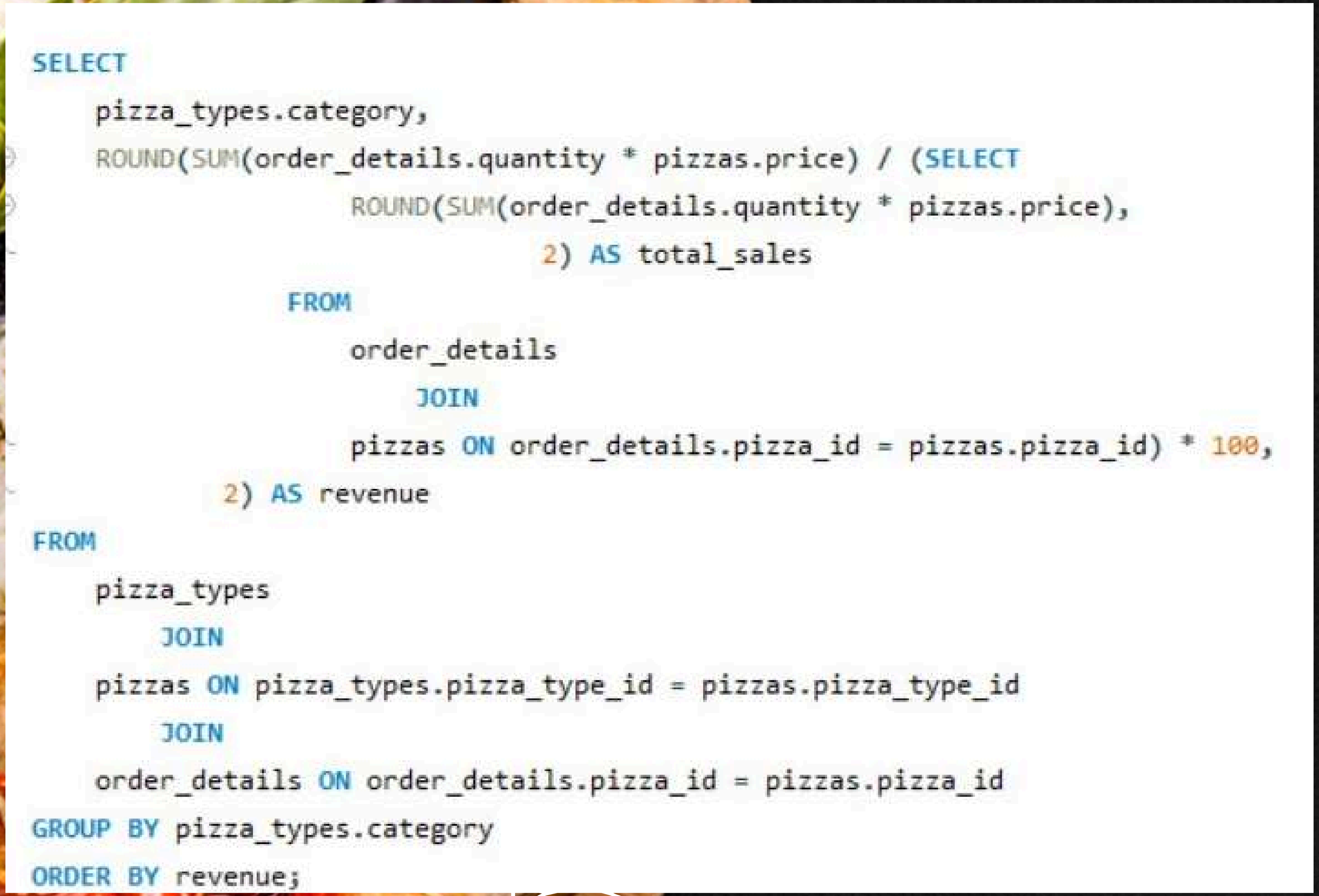
```
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name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



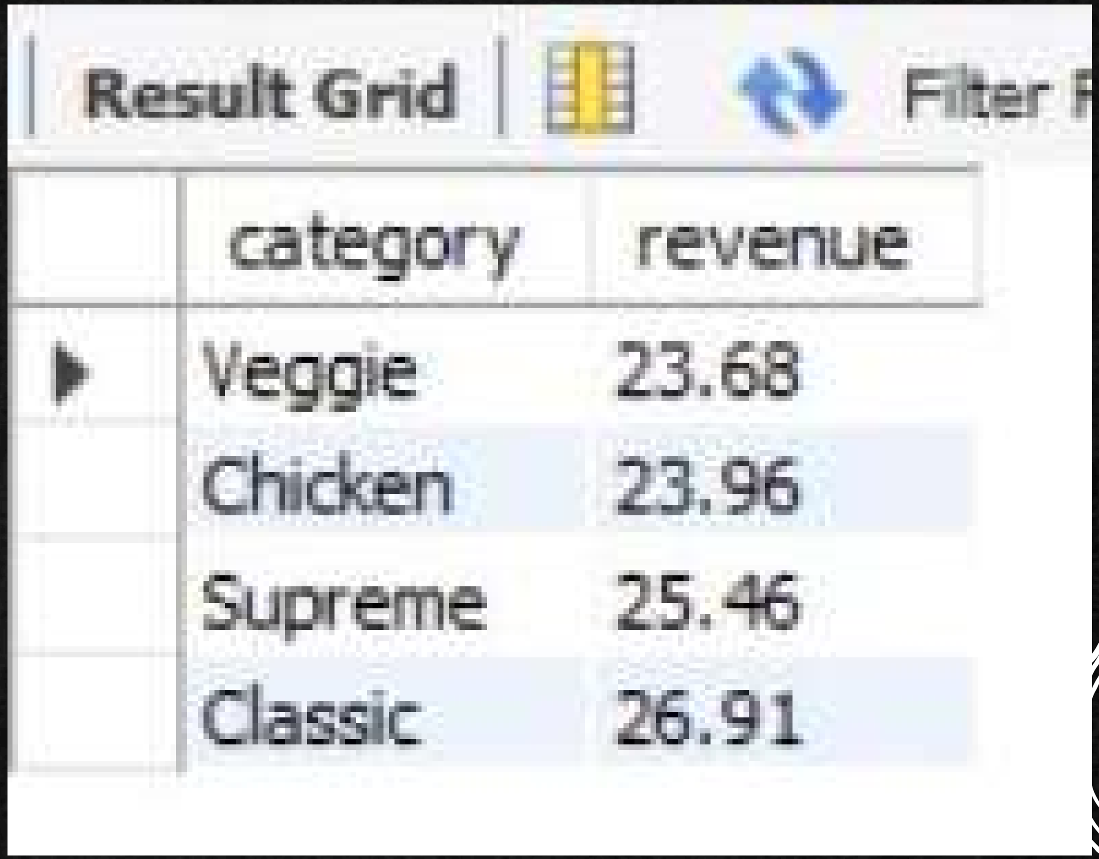
Result Grid | Filter Rows:

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

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



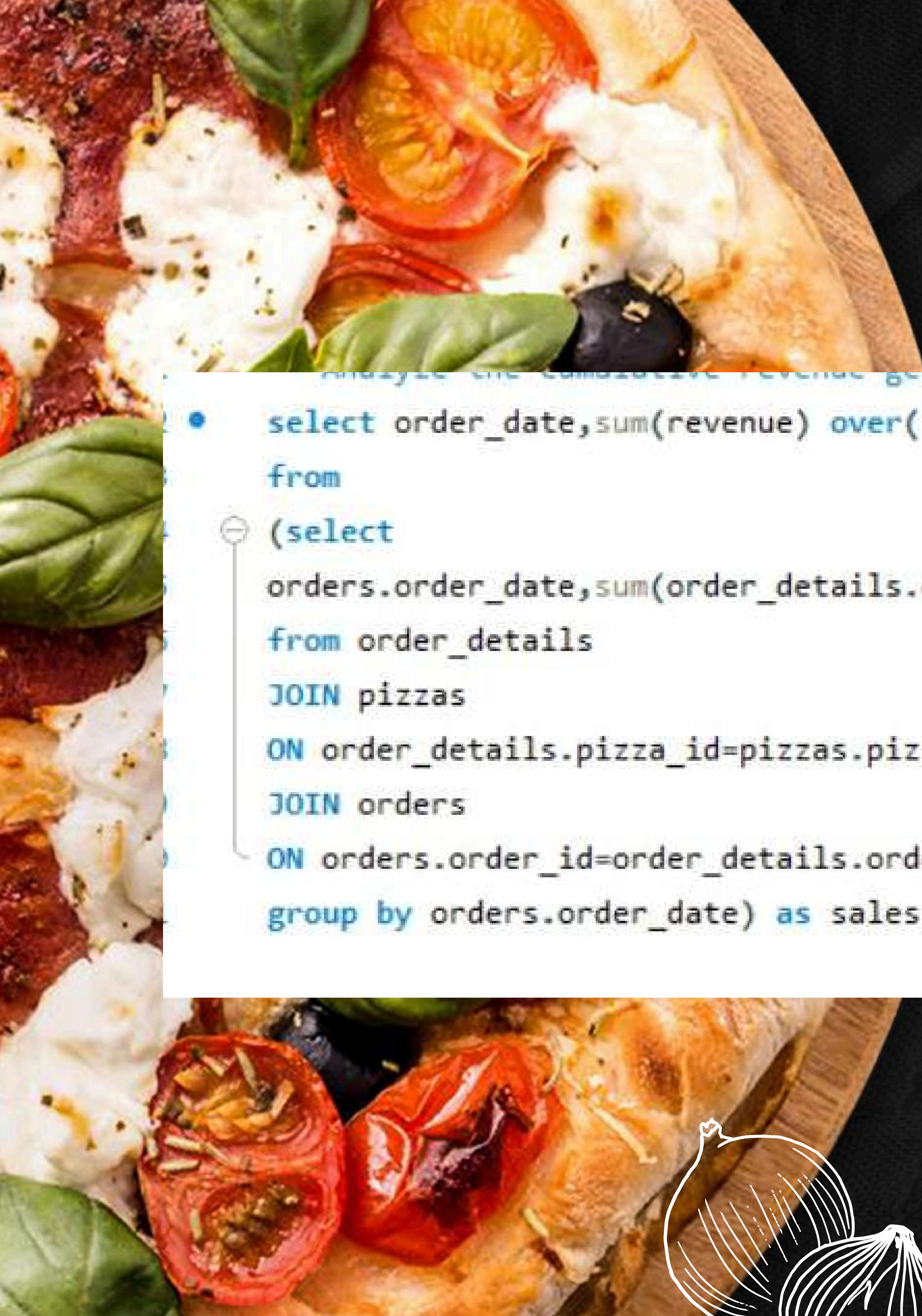
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                            2) AS total_sales
            FROM
                order_details
                    JOIN
                pizzas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

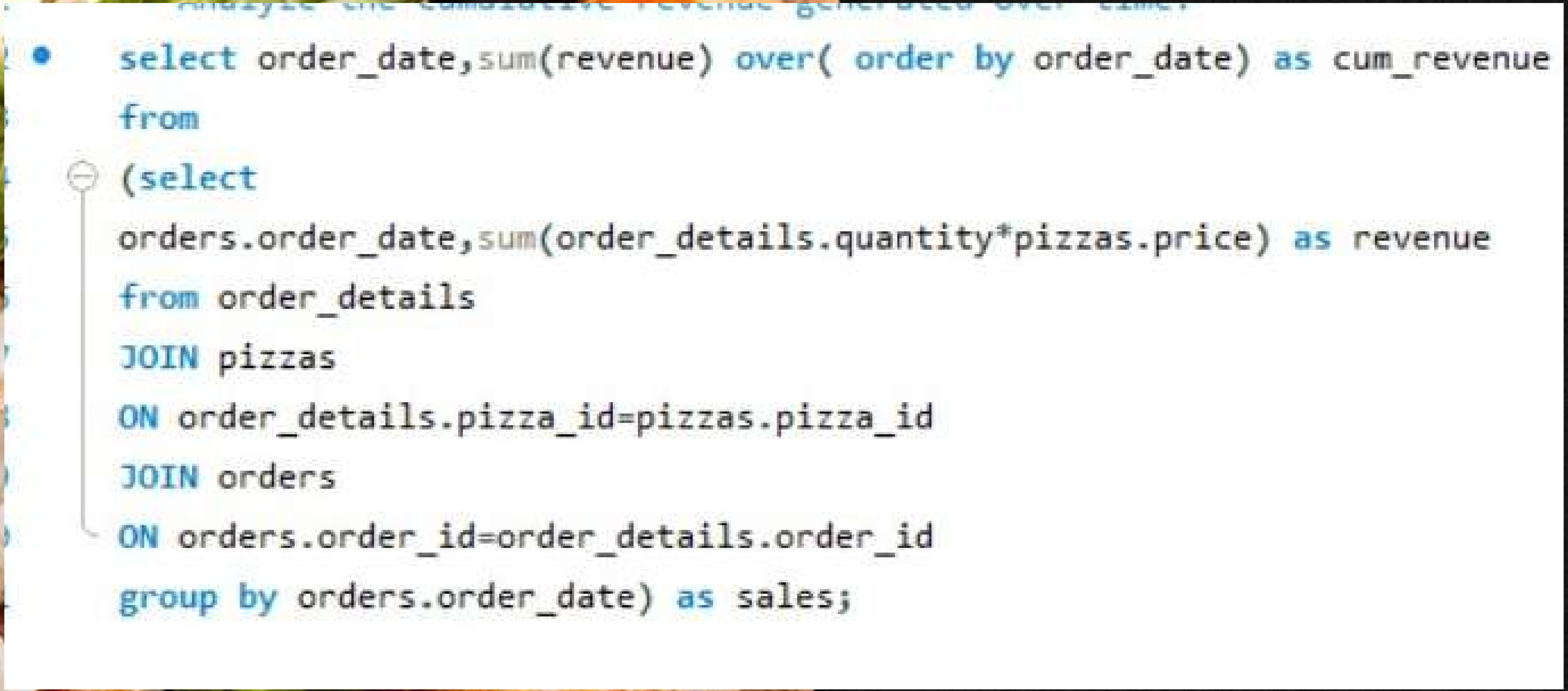


category	revenue
Veggie	23.68
Chicken	23.96
Supreme	25.46
Classic	26.91

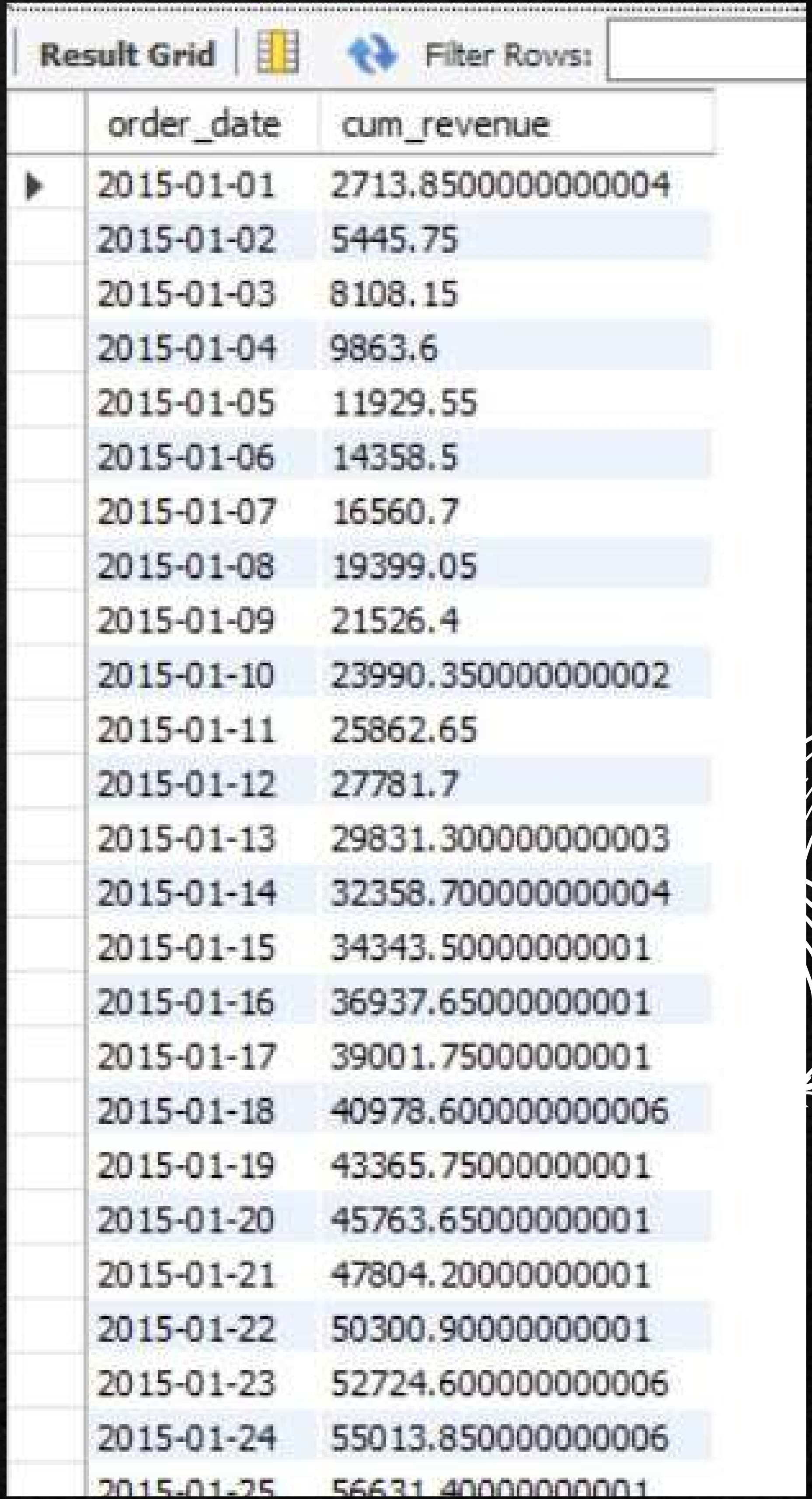
PIZZA HUT

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

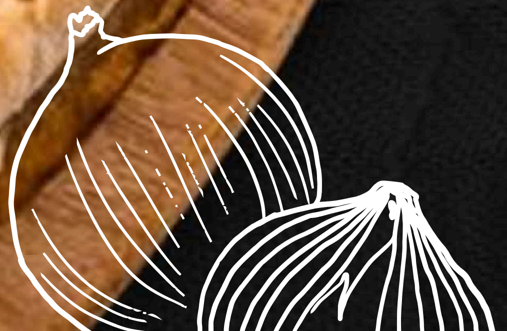




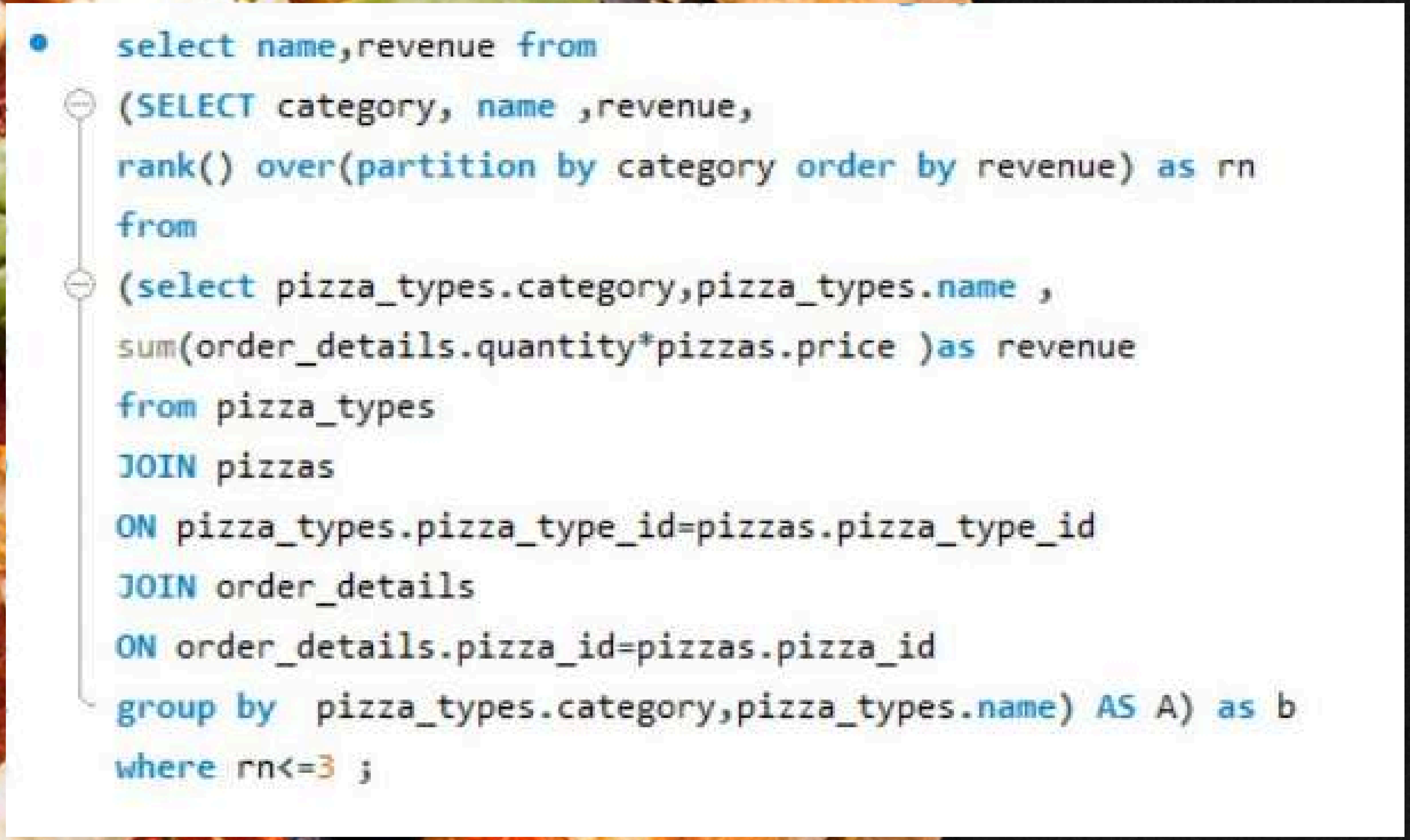
```
Analyze the cumulative revenue generated over time.
select order_date,sum(revenue) over( order by order_date) as cum_revenue
from
(select
orders.order_date,sum(order_details.quantity*pizzas.price) as revenue
from order_details
JOIN pizzas
ON order_details.pizza_id=pizzas.pizza_id
JOIN orders
ON orders.order_id=order_details.order_id
group by orders.order_date) as sales;
```



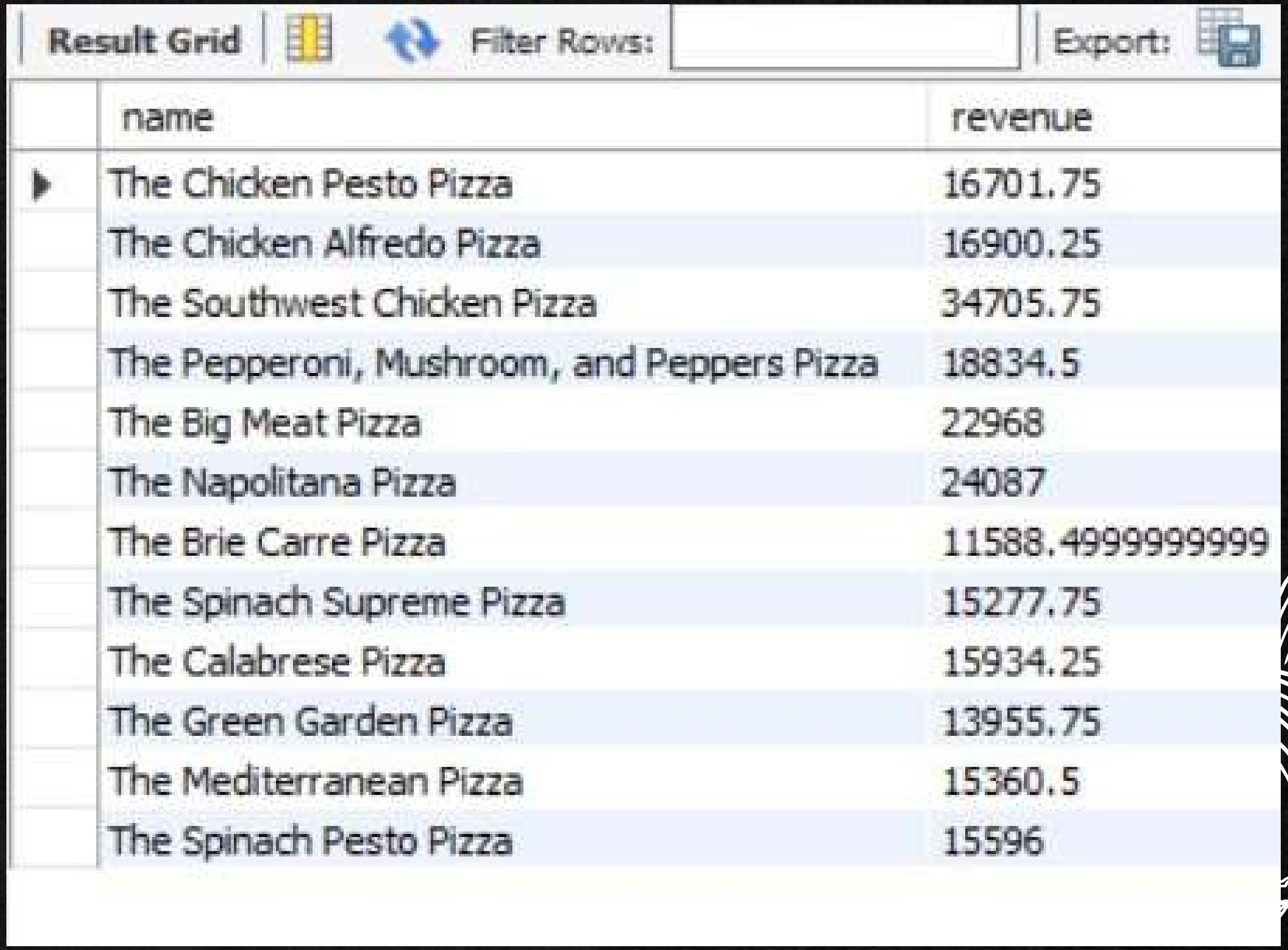
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
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
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001



13. 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) 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 ;
```



Result Grid | Filter Rows: | Export:

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75
The Calabrese Pizza	15934.25
The Green Garden Pizza	13955.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

PIZZA HUT

THANK YOU!