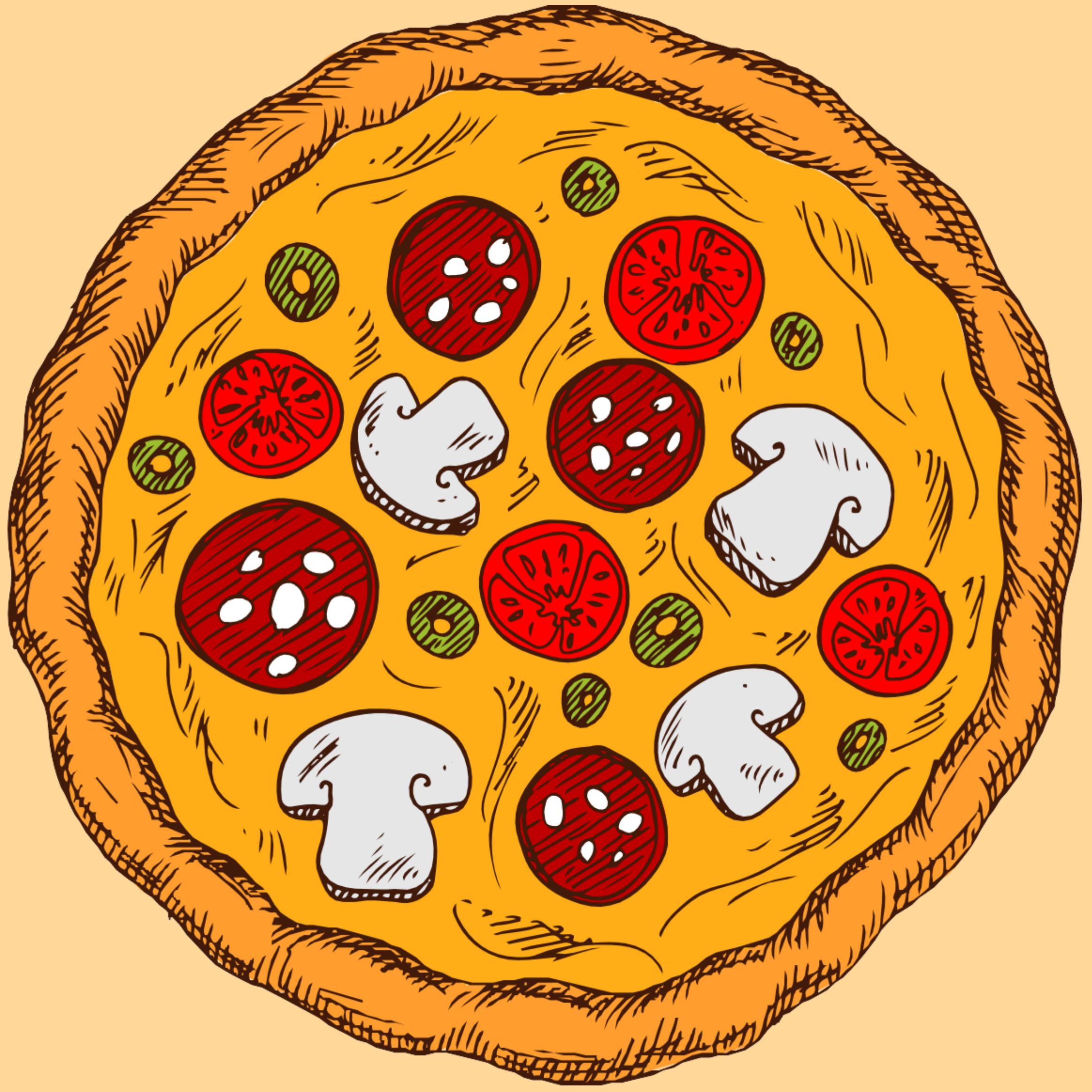
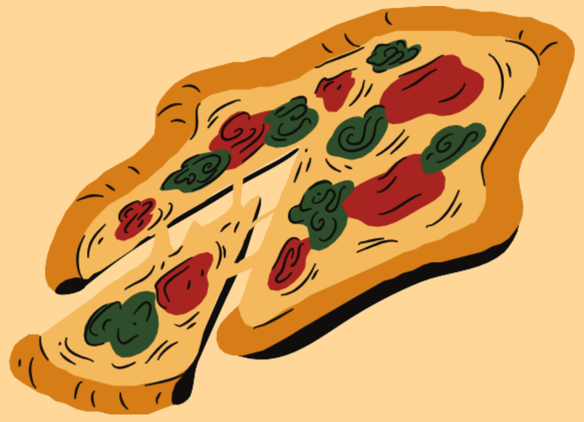


PIZZA SALES ANALYSIS USING MYSQL

A Data-driven Approach to Understand Pizza Sales Patterns

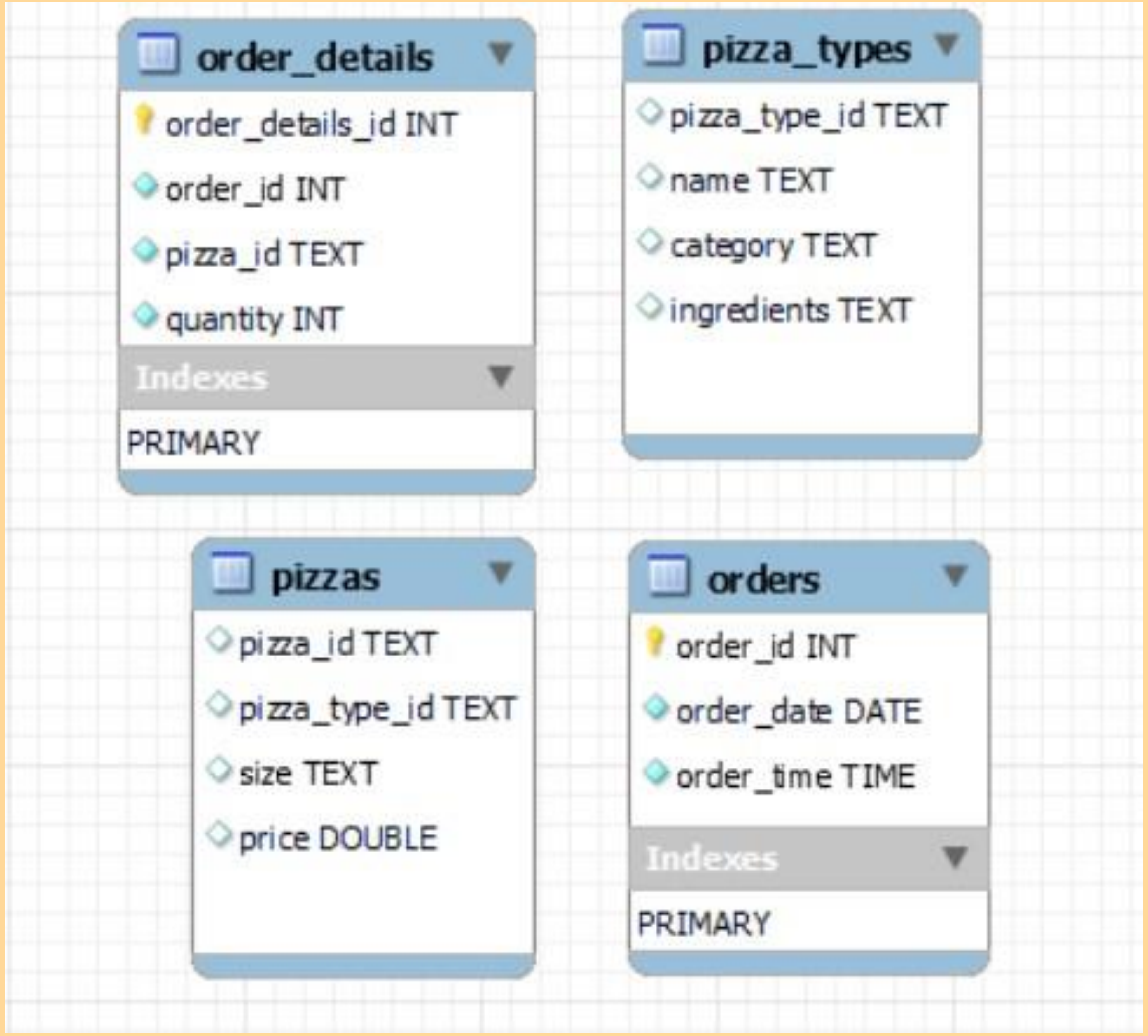
Presented by: Adithya Nagaraj
E-mail Id : Adithya.nagaraj938@gmail.com



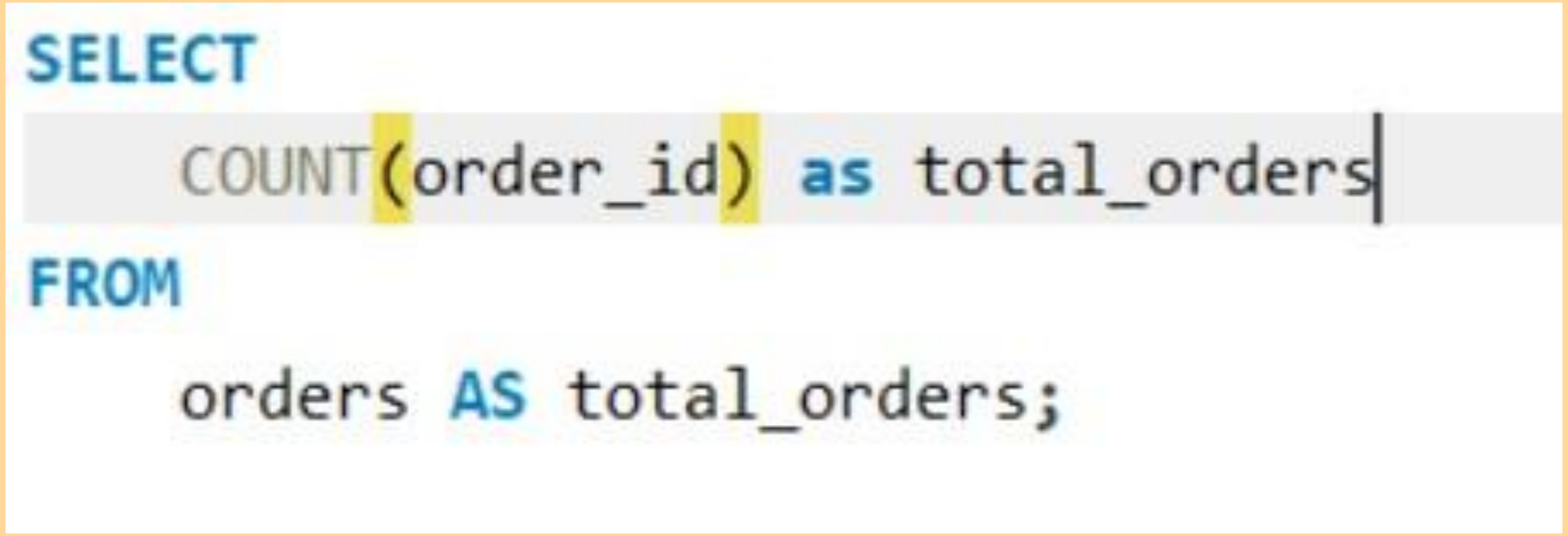
HELLO LINKEDIN COMMUNITY!!!

I am excited to share My new project on Pizza Sales And Order Analysis using MySQL !!!

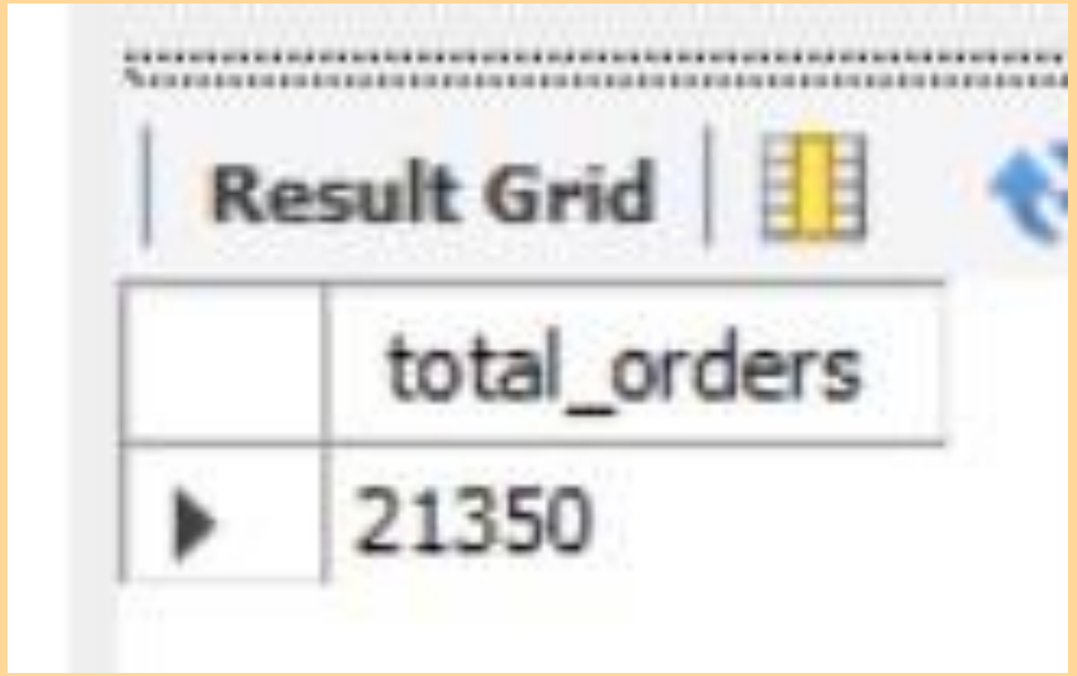
ER DIAGRAM


order_details
order_details_id INT
order_id INT
pizza_id TEXT
quantity INT
Indexes
PRIMARY
pizza_types
pizza_type_id TEXT
name TEXT
category TEXT
ingredients TEXT
pizzas
pizza_id TEXT
pizza_type_id TEXT
size TEXT
price DOUBLE
orders
order_id INT
order_date DATE
order_time TIME
Indexes
PRIMARY

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

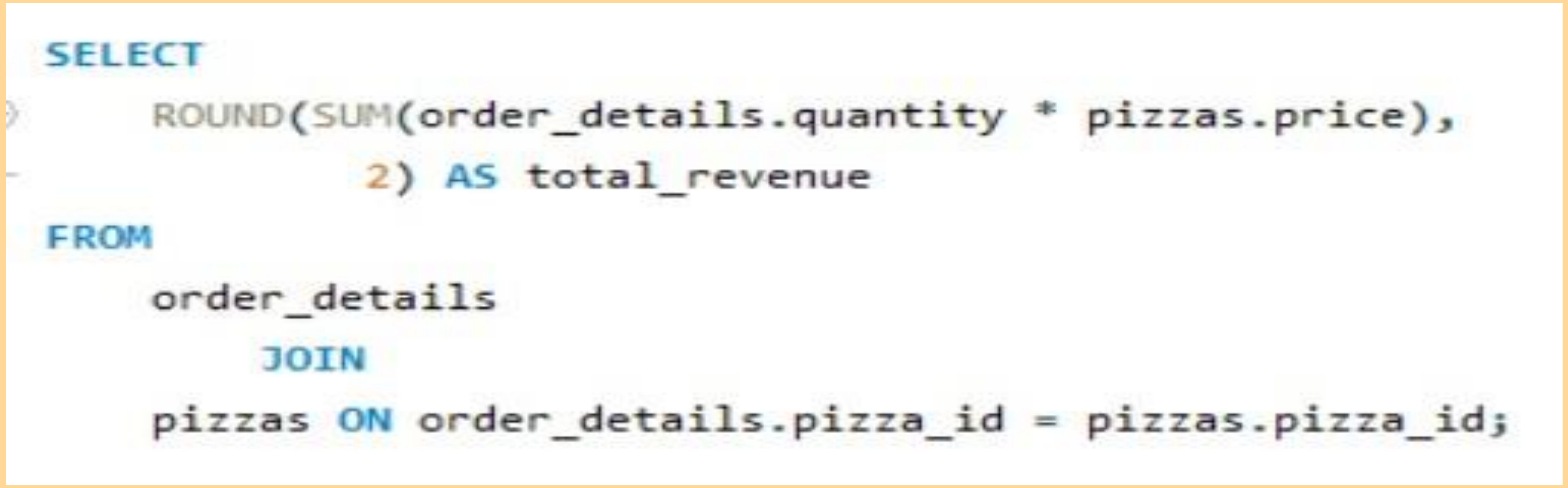


```
SELECT
    COUNT(order_id) as total_orders
FROM
    orders AS total_orders;
```

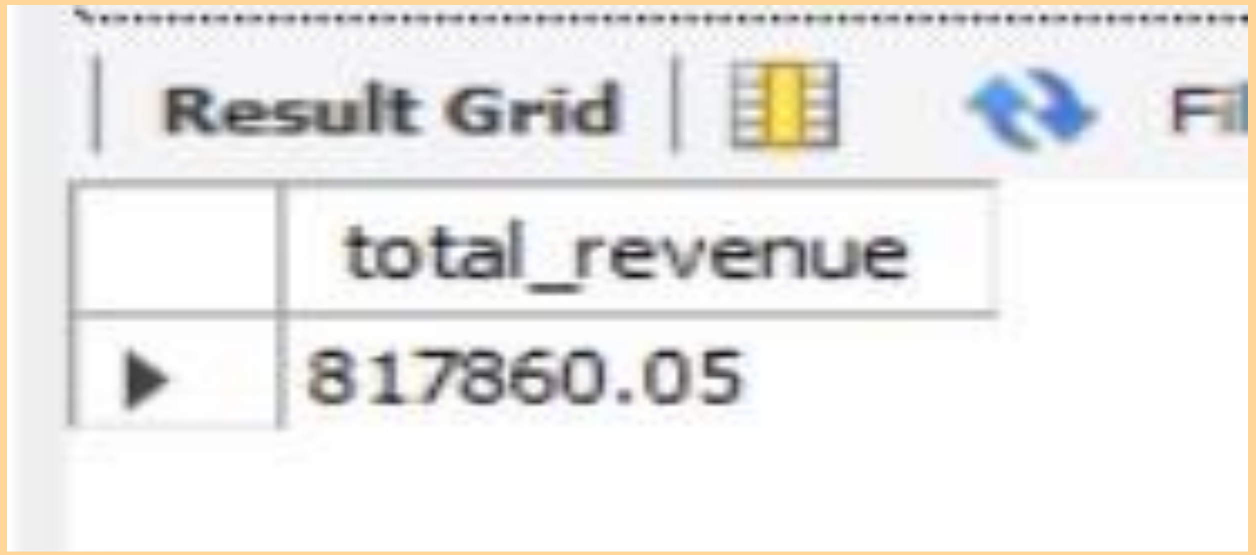


```
total_orders
21350
```

2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

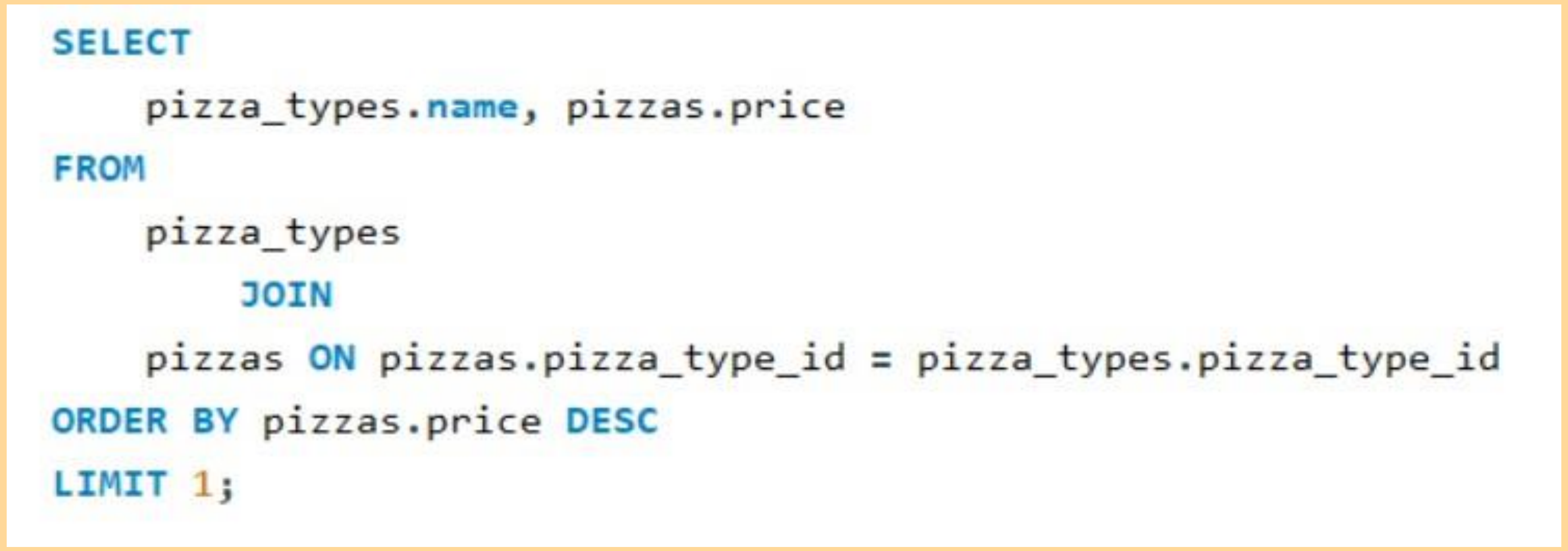


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

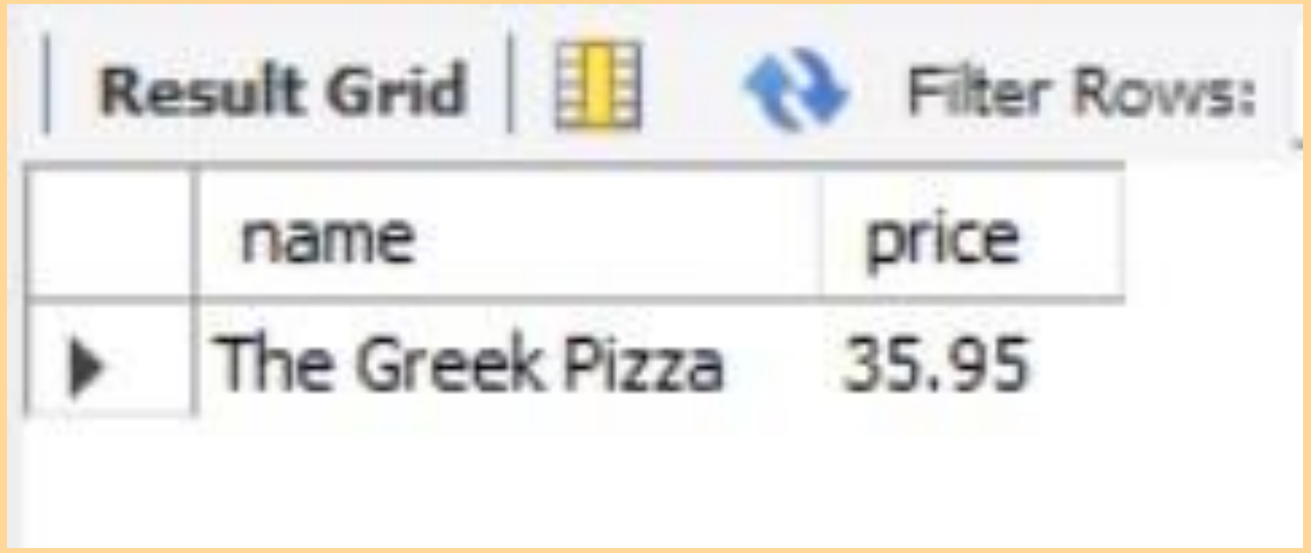


```
total_revenue
817860.05
```

3.IDENTIFY THE HIGHEST-PRICED PIZZA

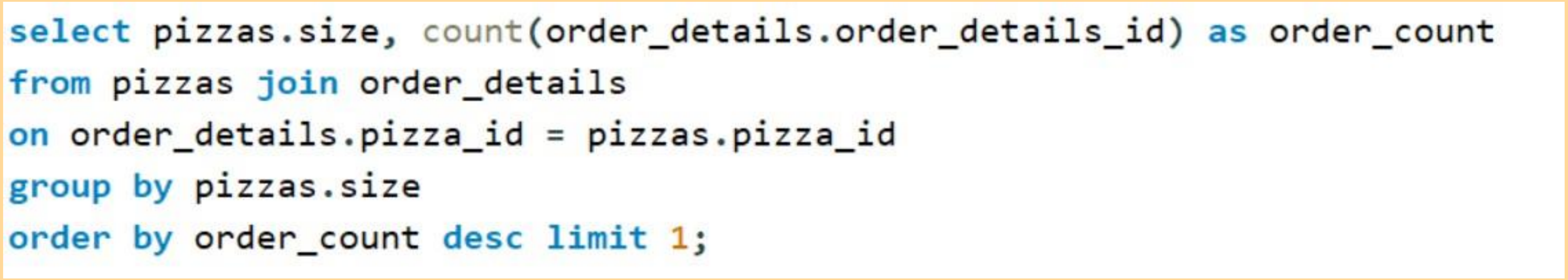


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

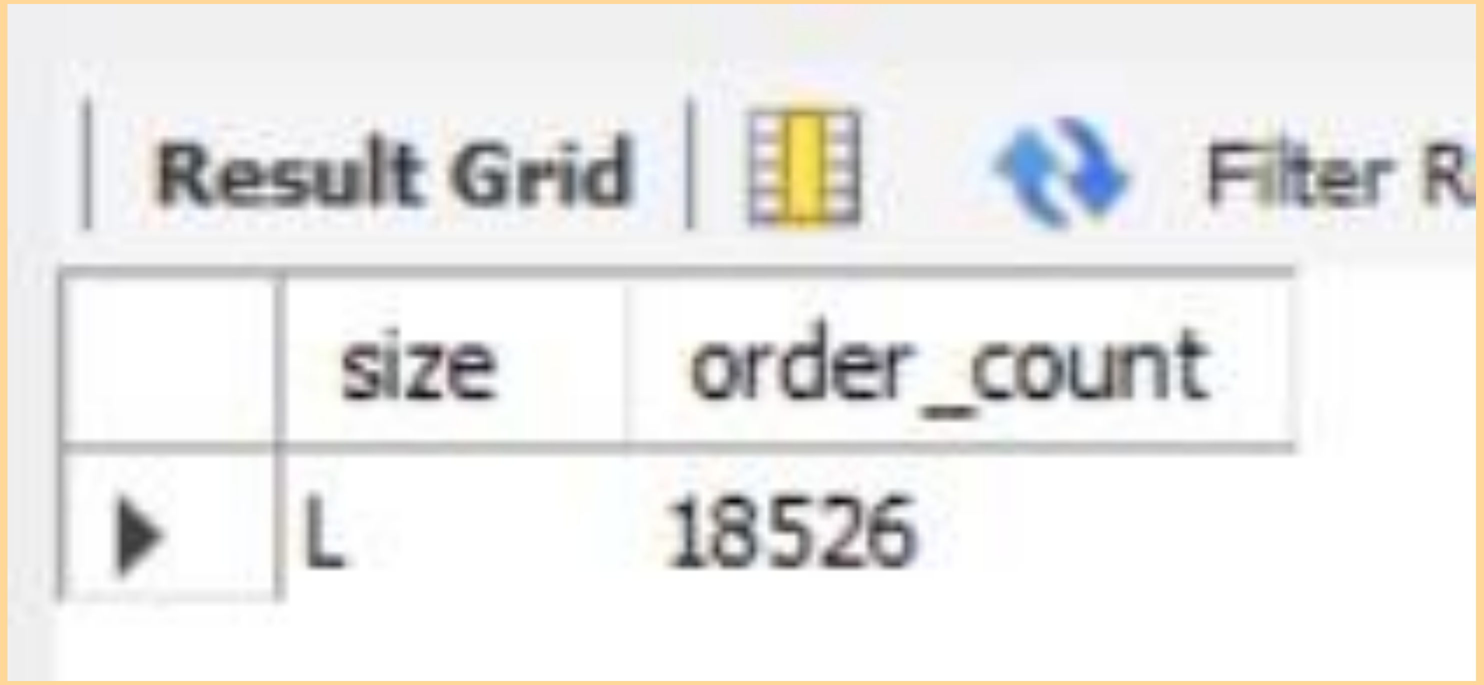


name	price
The Greek Pizza	35.95

4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

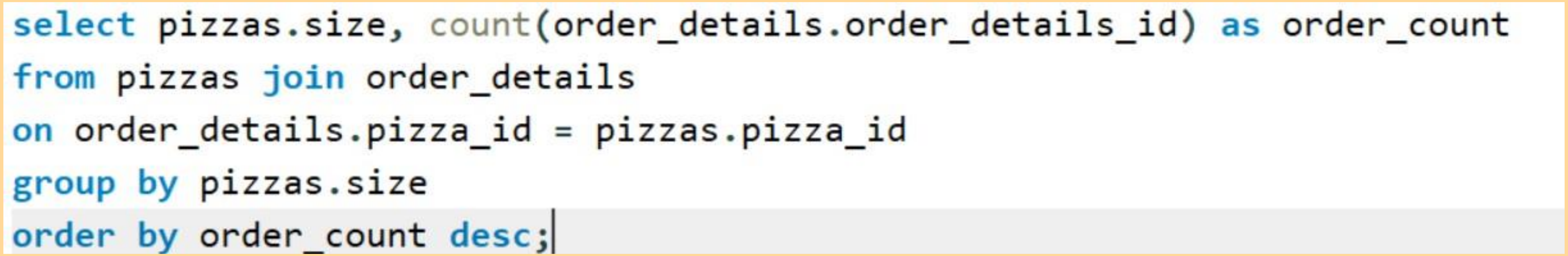


```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by order_count desc limit 1;
```

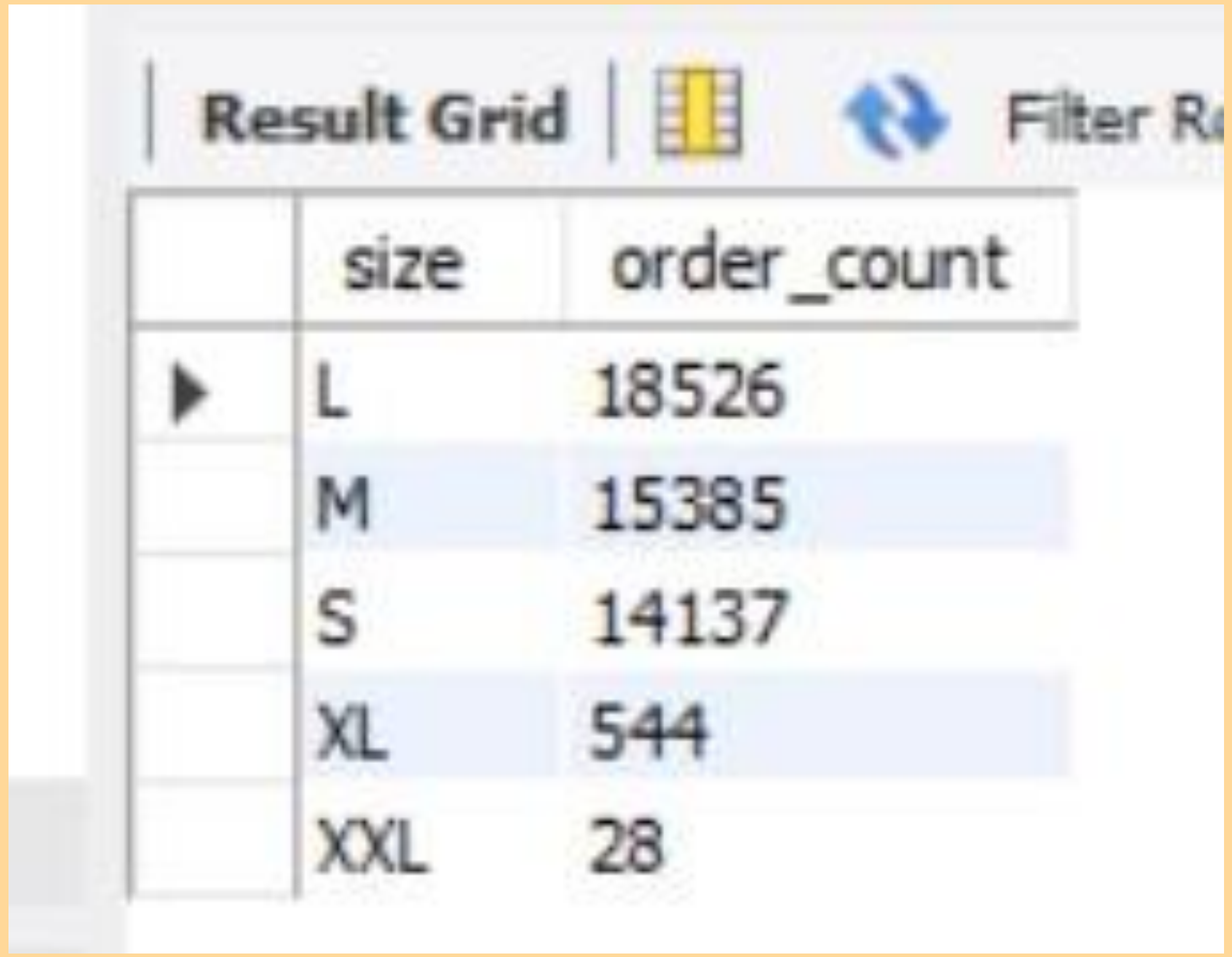


```
size  order_count
L     18526
```

5.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

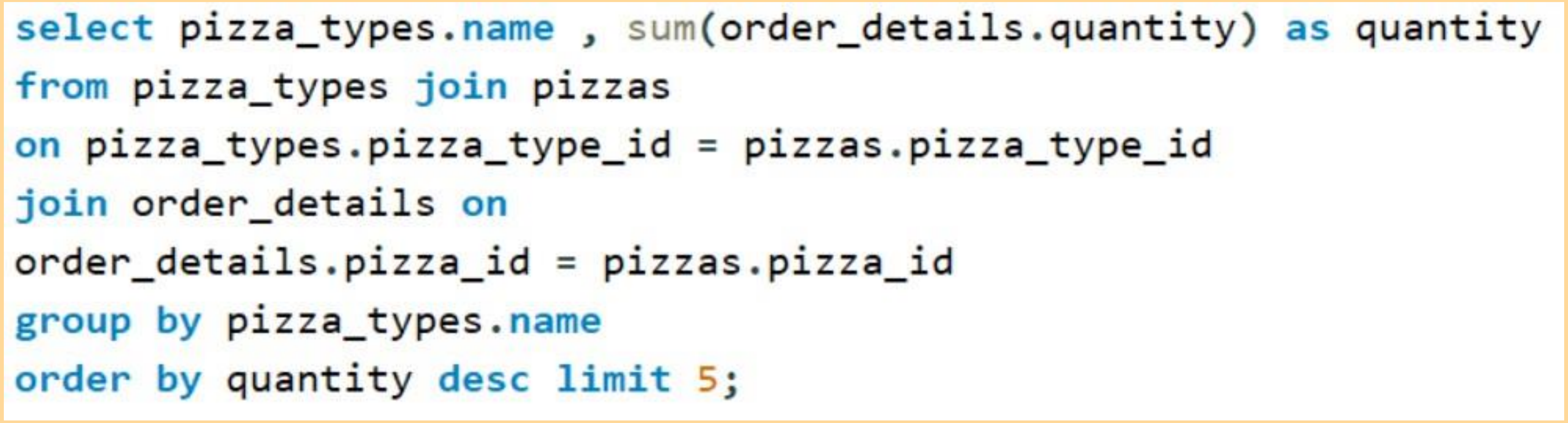


```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by order_count desc;
```

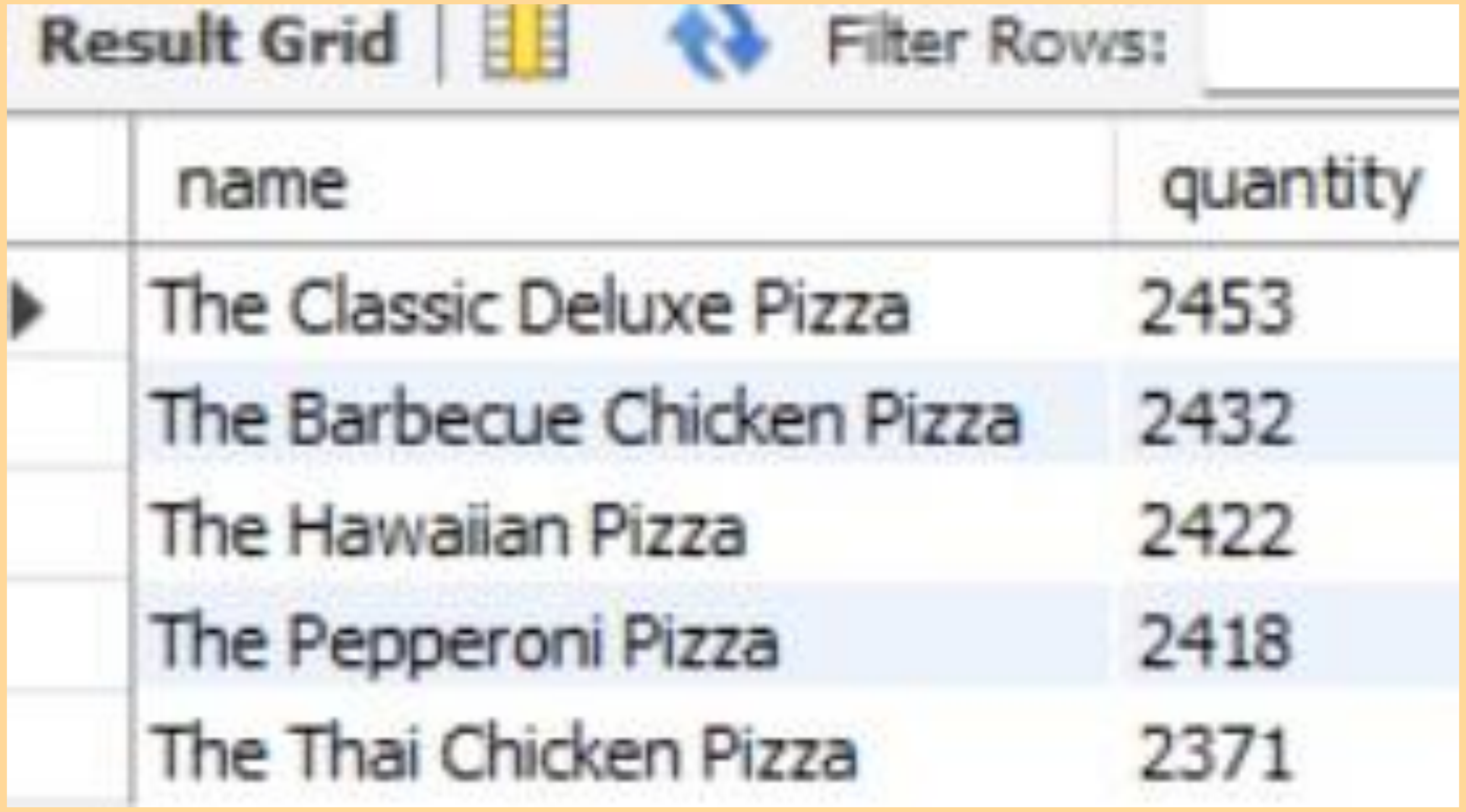


size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

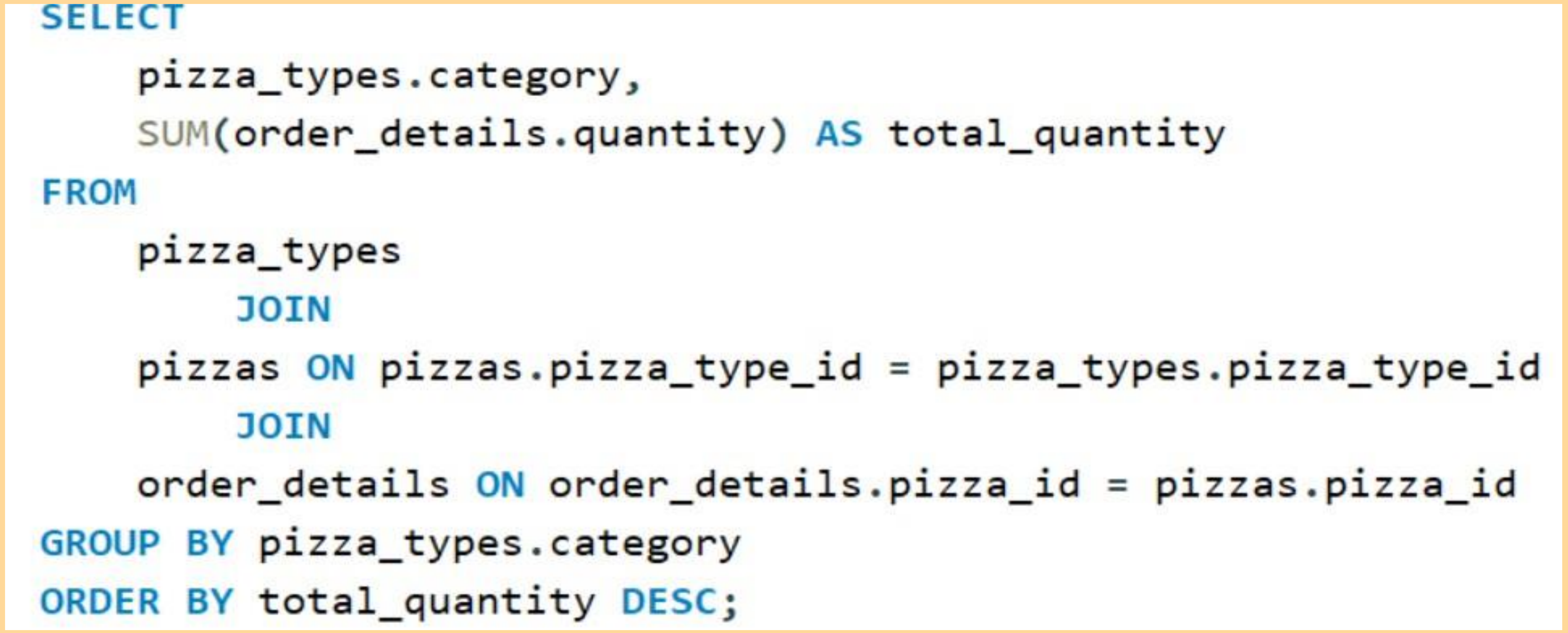


```sql
select pizza_types.name , sum(order_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on
order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by quantity desc limit 5;
```

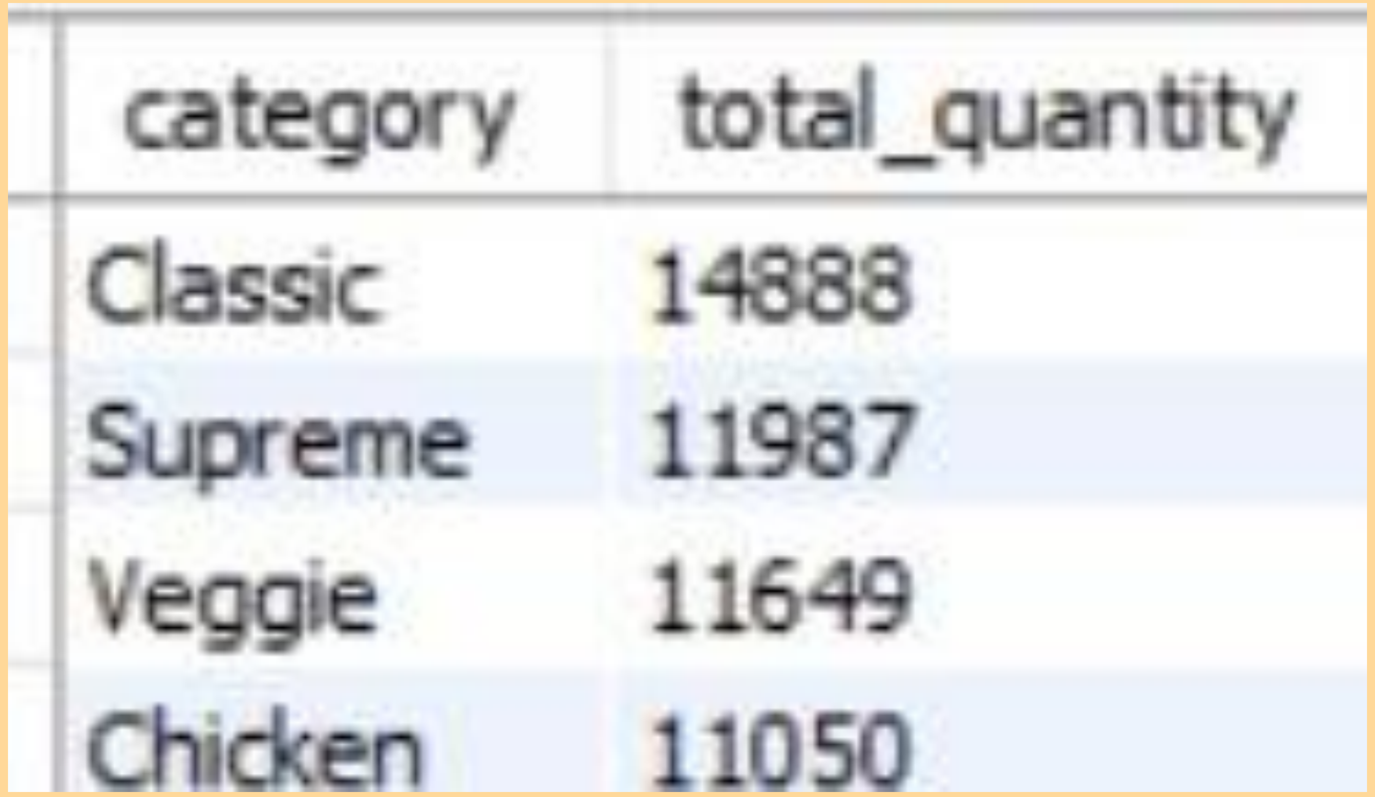


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

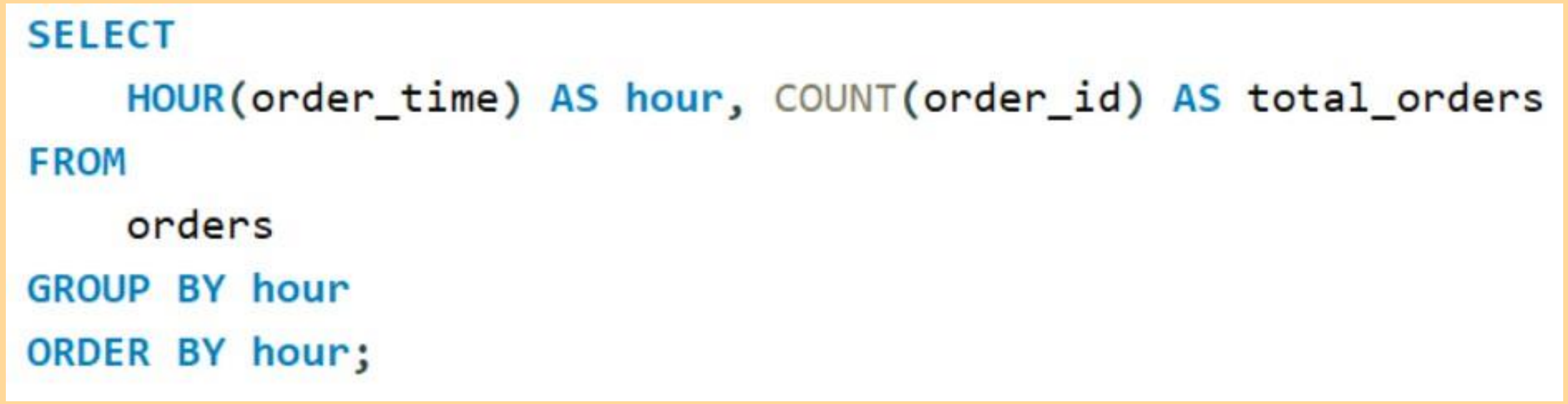


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity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

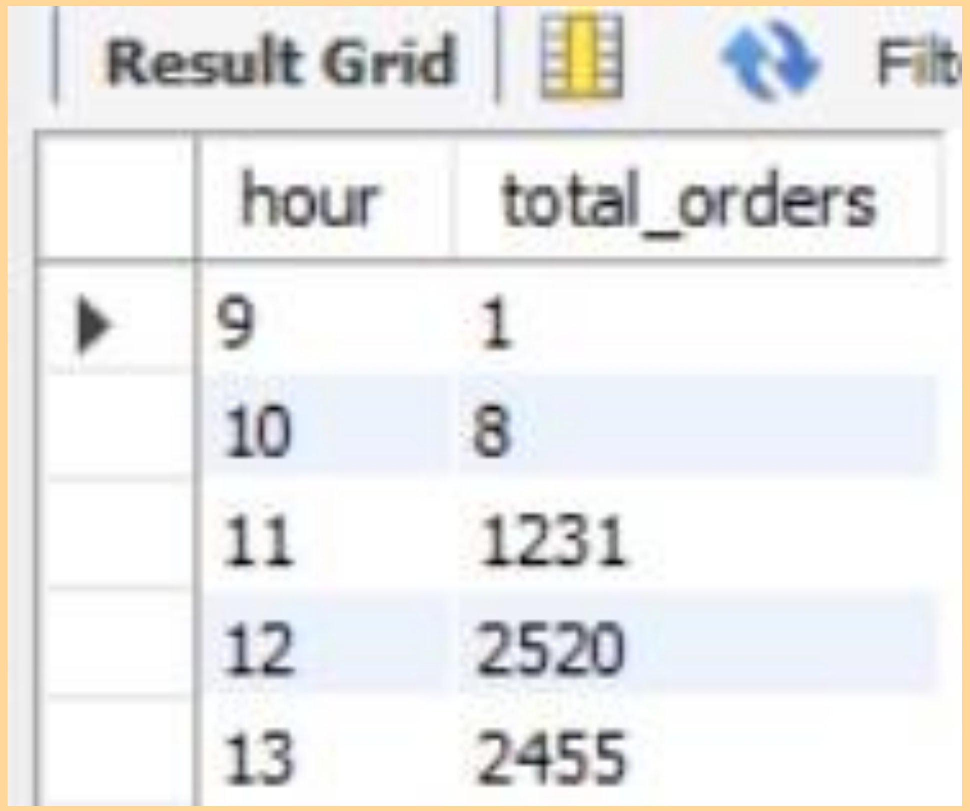


category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

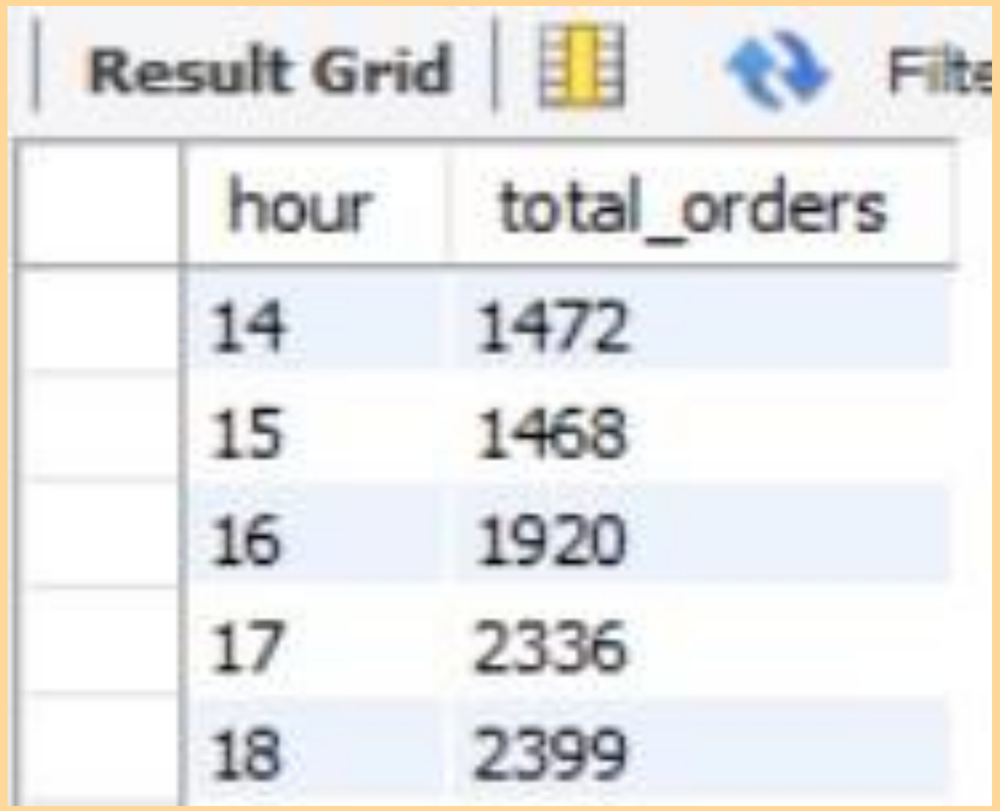
8.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



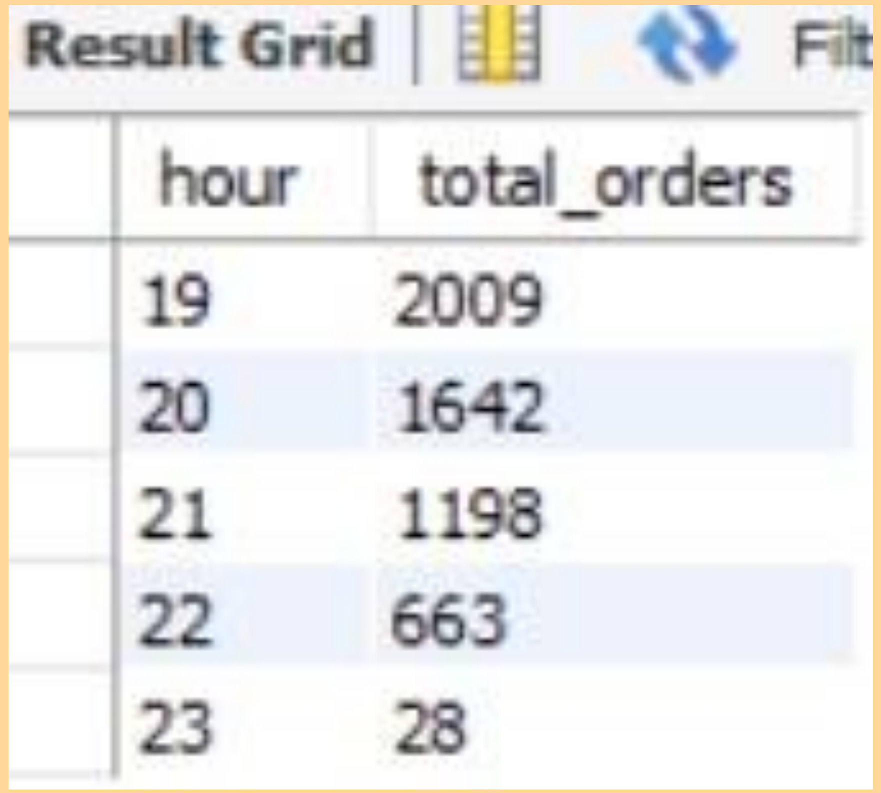
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY hour
ORDER BY hour;
```



hour	total_orders
9	1
10	8
11	1231
12	2520
13	2455

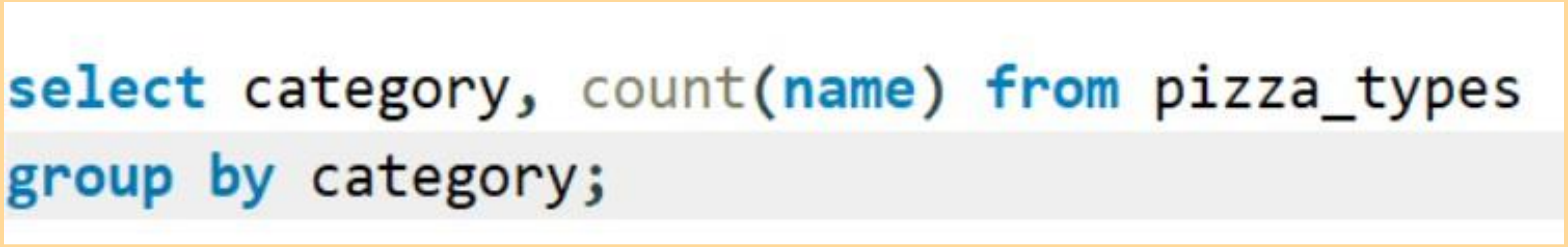


hour	total_orders
14	1472
15	1468
16	1920
17	2336
18	2399

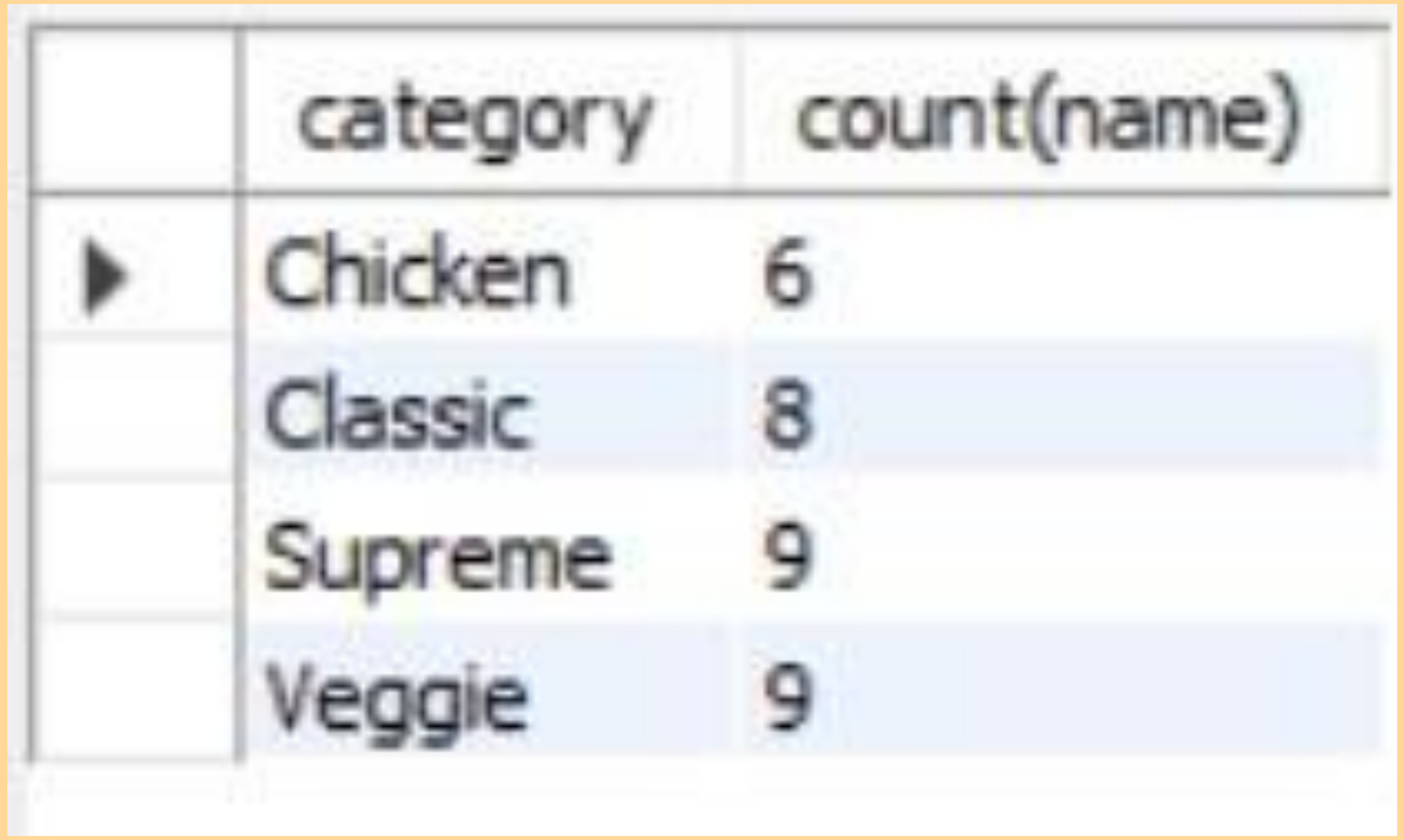


hour	total_orders
19	2009
20	1642
21	1198
22	663
23	28

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

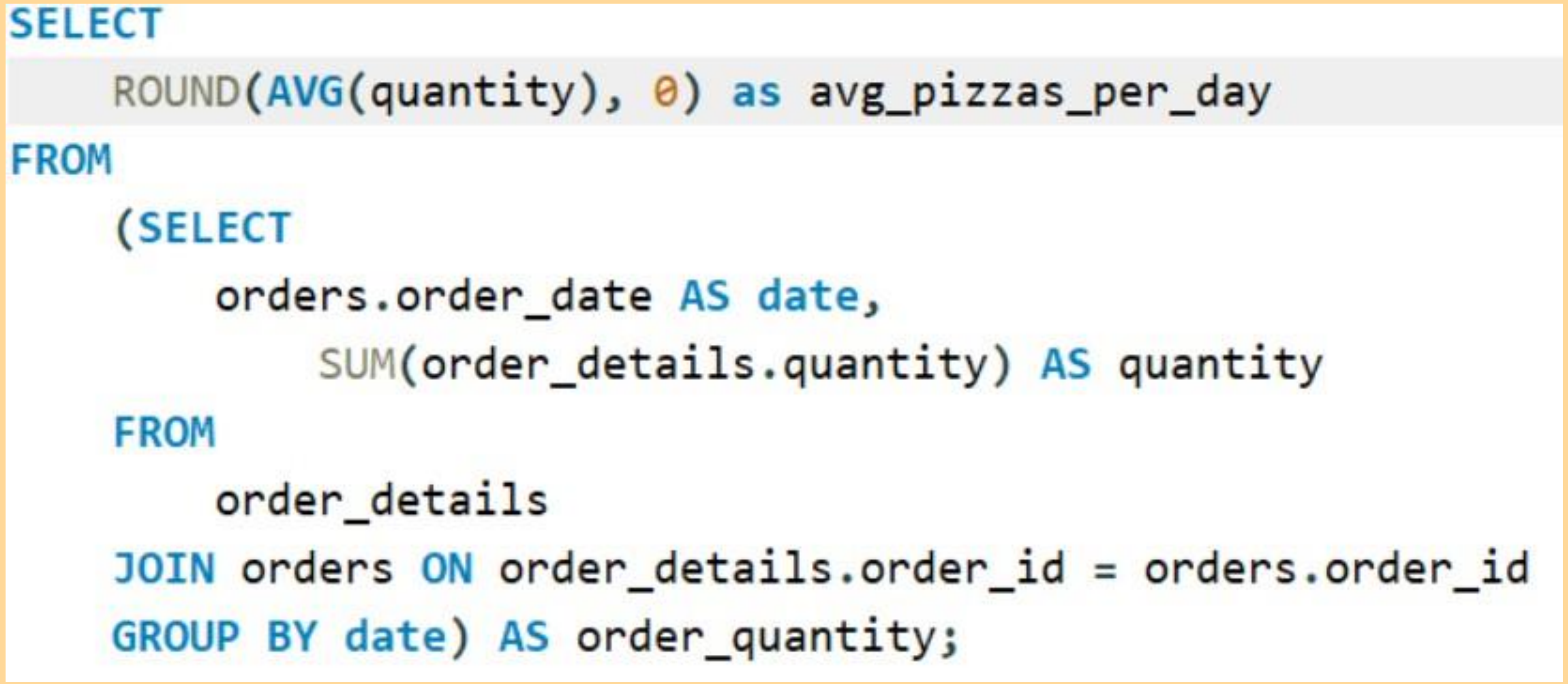


```
select category, count(name) from pizza_types
group by category;
```

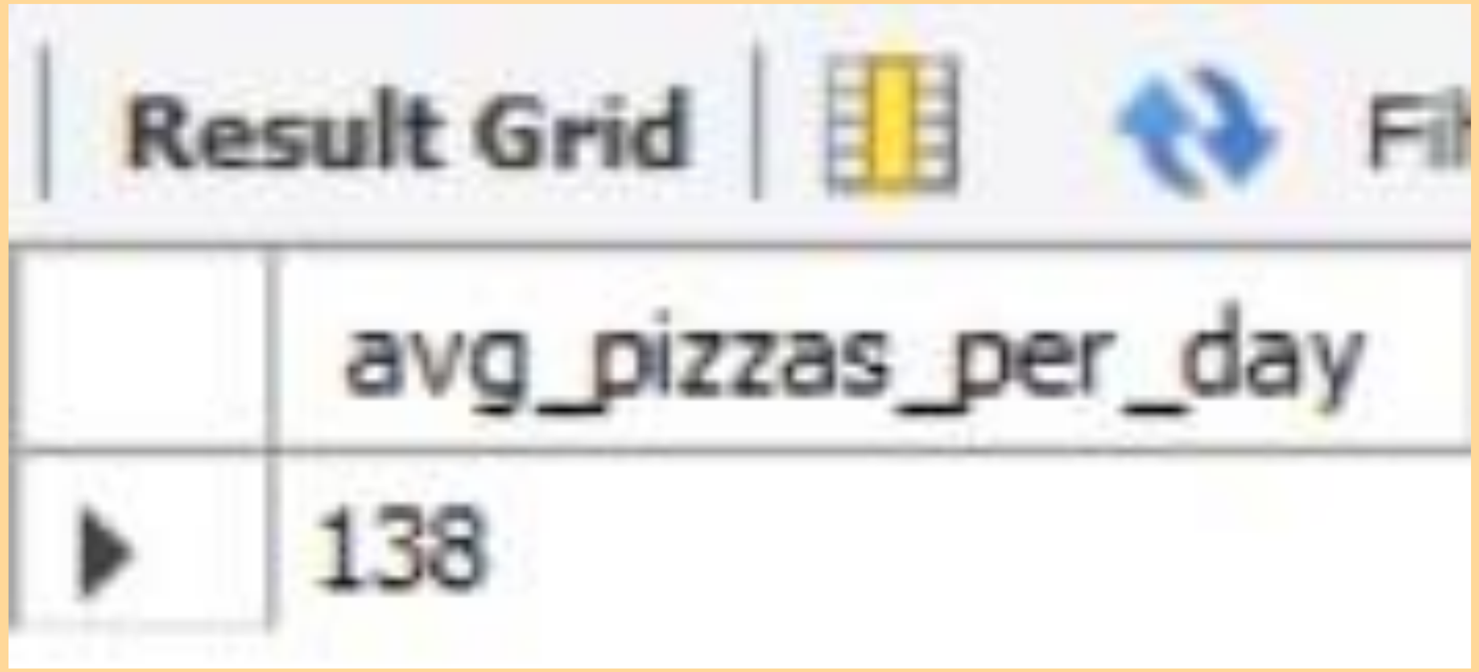


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

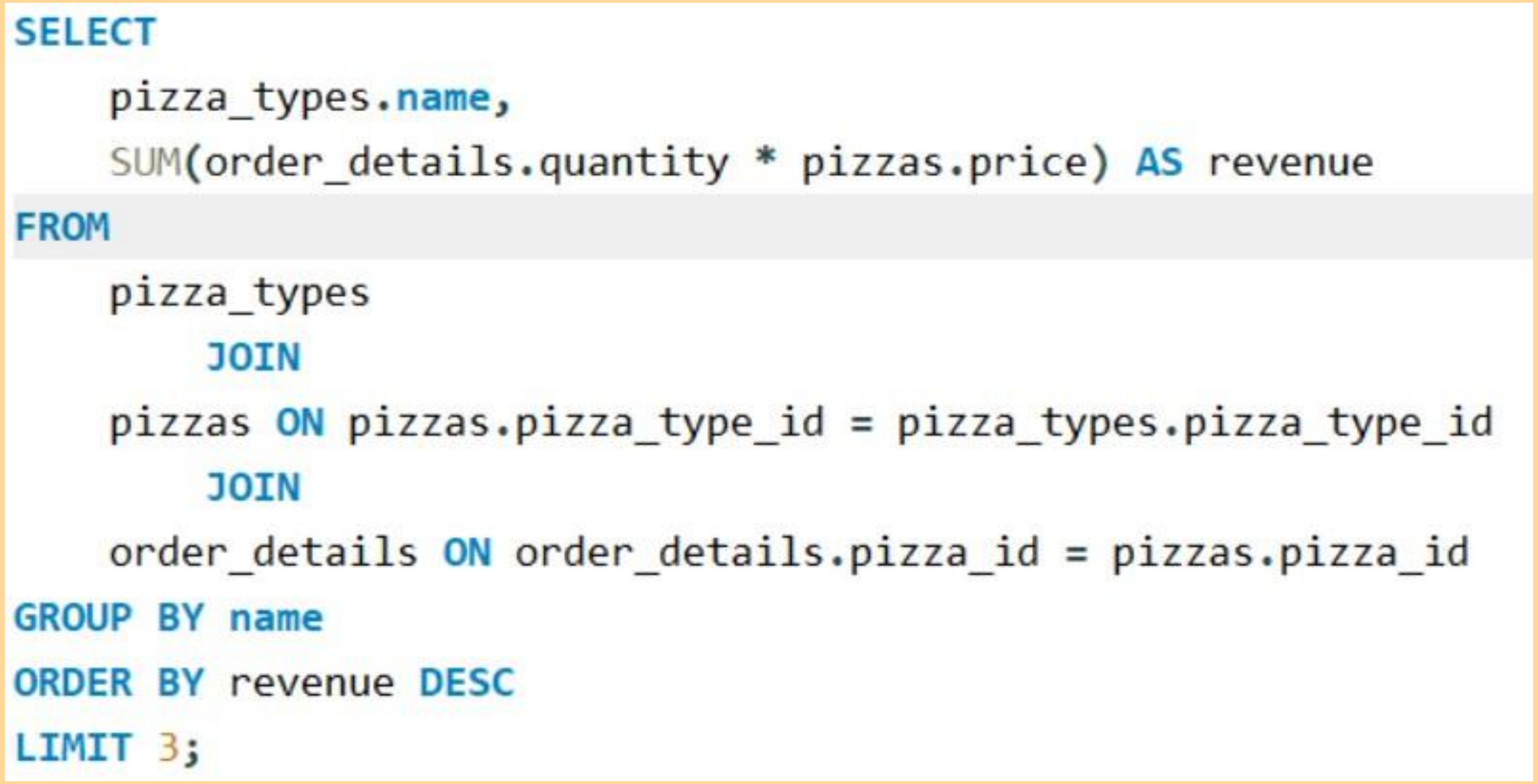


```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_per_day
FROM
    (SELECT
        orders.order_date AS date,
            SUM(order_details.quantity) AS quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY date) AS order_quantity;
```

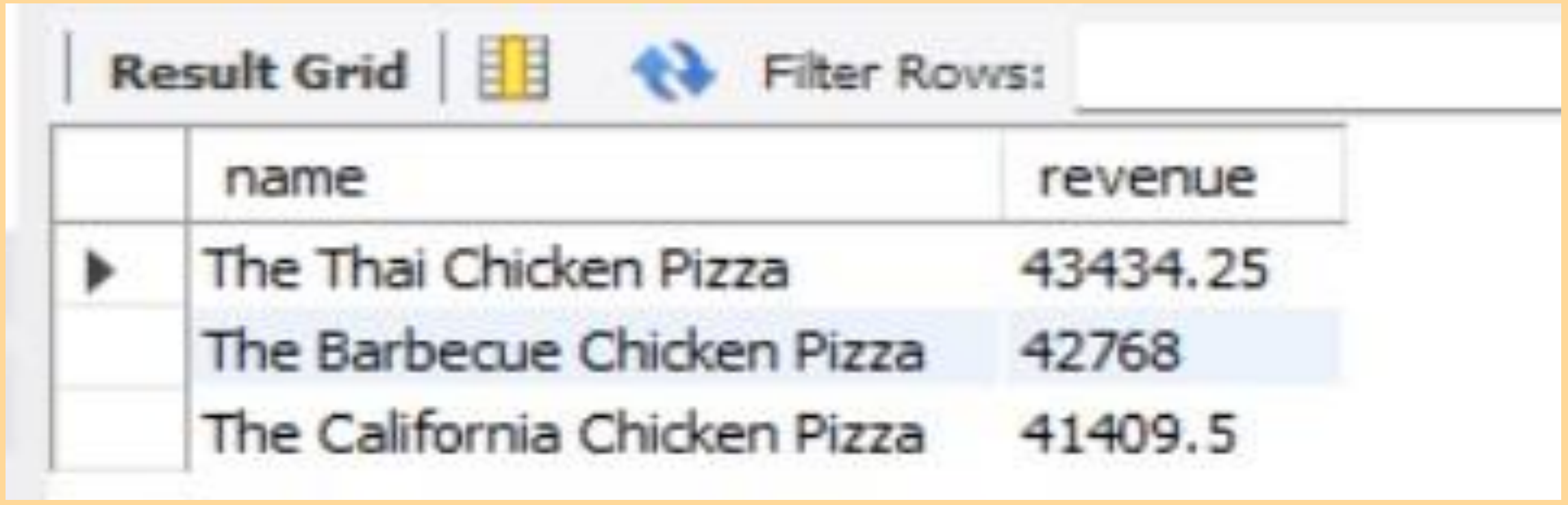


```
Result Grid
avg_pizzas_per_day
138
```

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



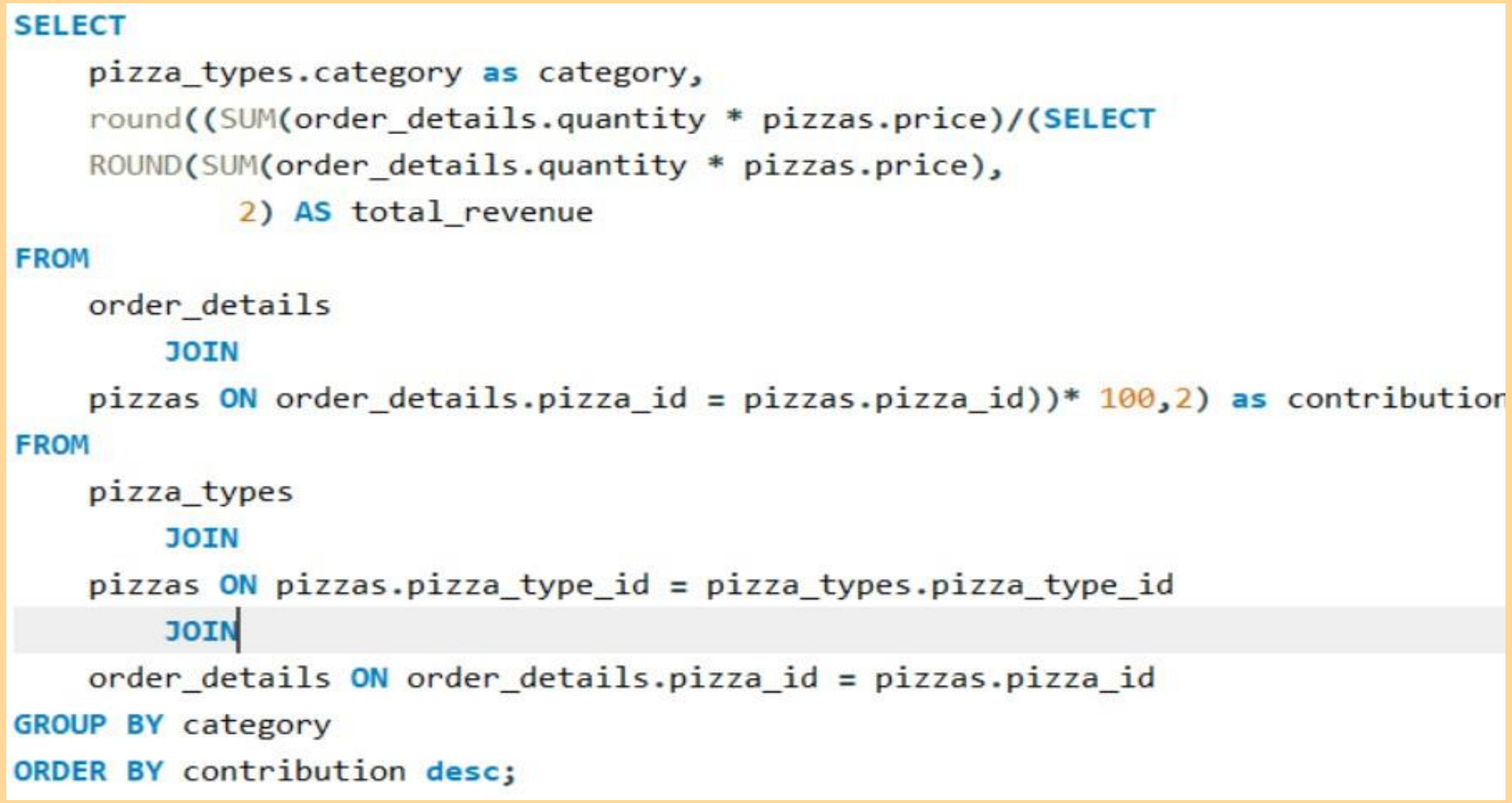
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY name
ORDER BY revenue DESC
LIMIT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



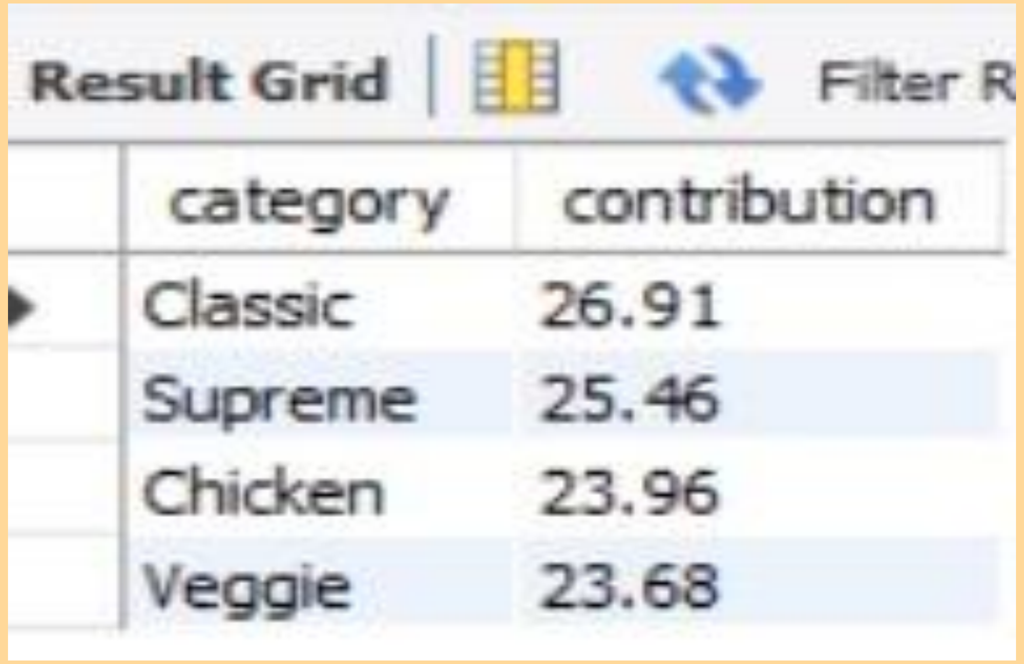
Result Grid | Filter Rows:

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

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



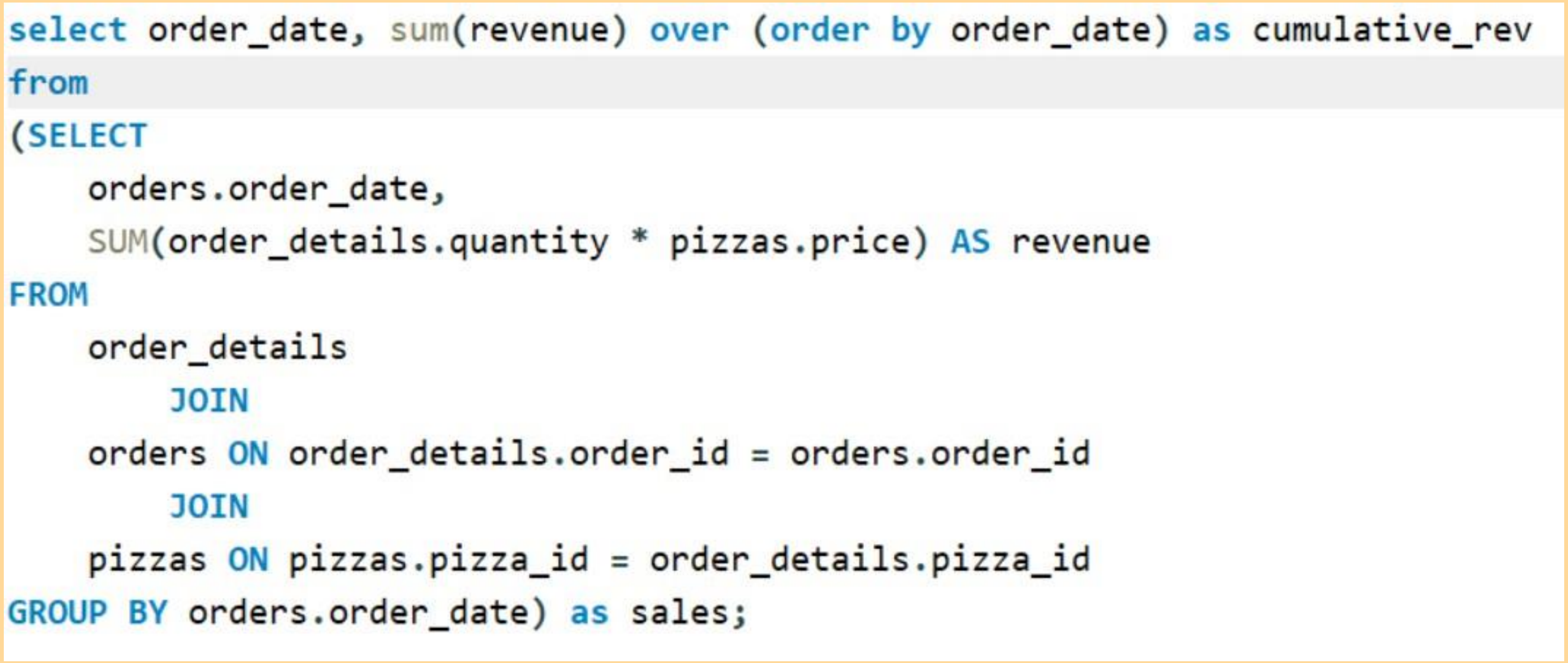
```
SELECT
    pizza_types.category as category,
    round((SUM(order_details.quantity * pizzas.price)/(SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id))* 100,2) as contributior
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 category
ORDER BY contribution desc;
```



Result Grid | Filter R

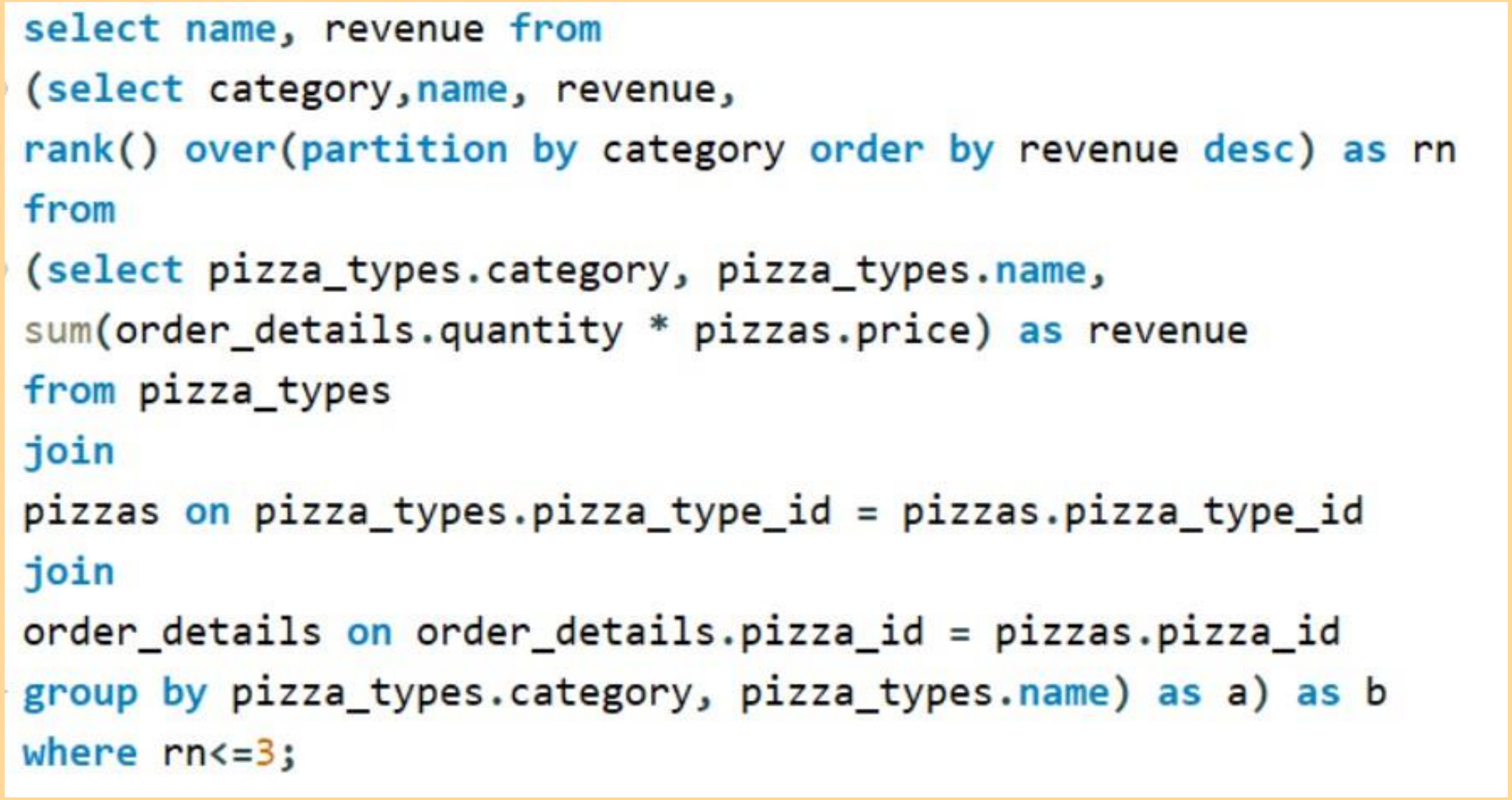
category	contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

13.ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

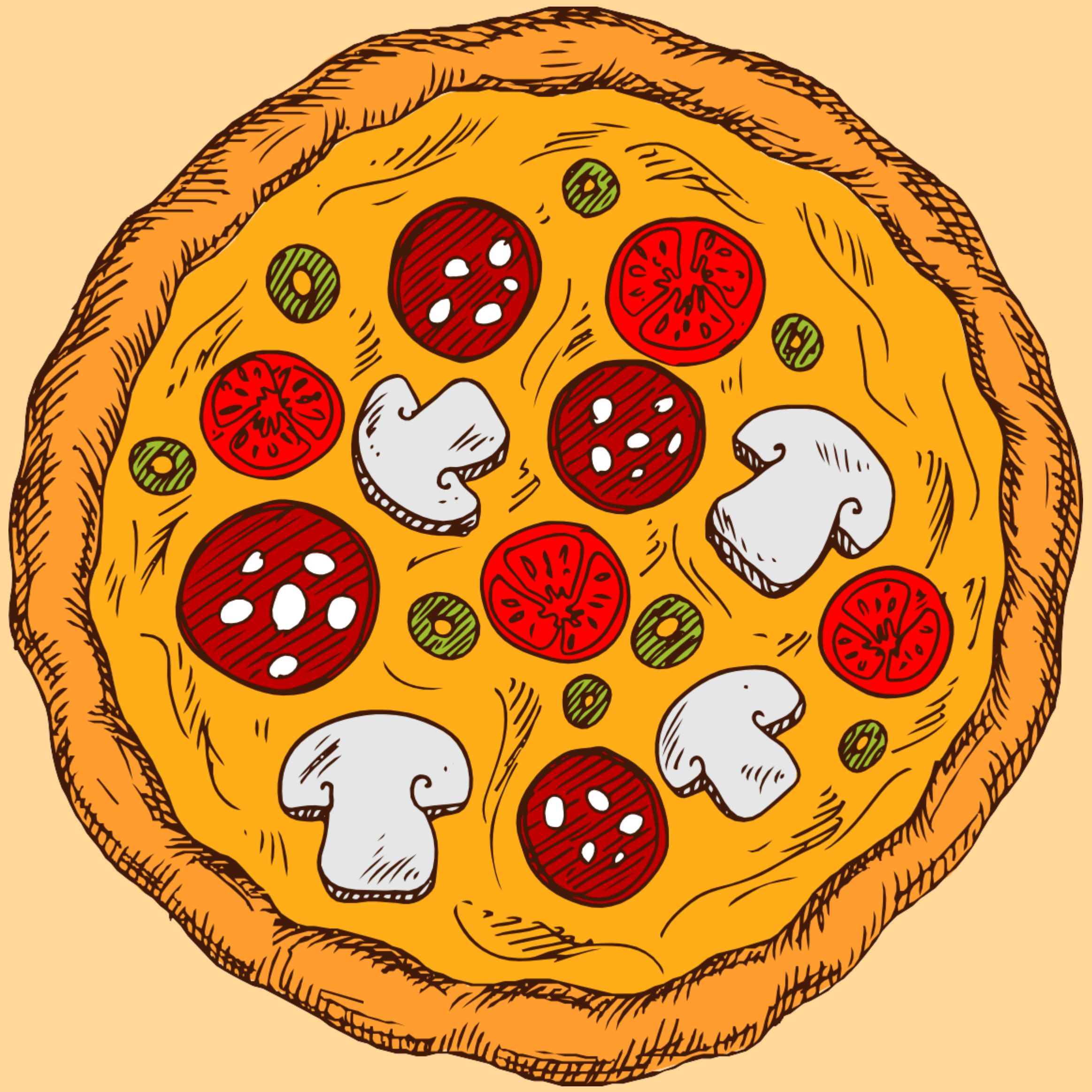


```sql
select order_date, sum(revenue) over (order by order_date) as cumulative_rev
from
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
        JOIN
    orders ON order_details.order_id = orders.order_id
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY orders.order_date) as sales;
```

14.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;
```



Hope you liked it!

THANK YOU