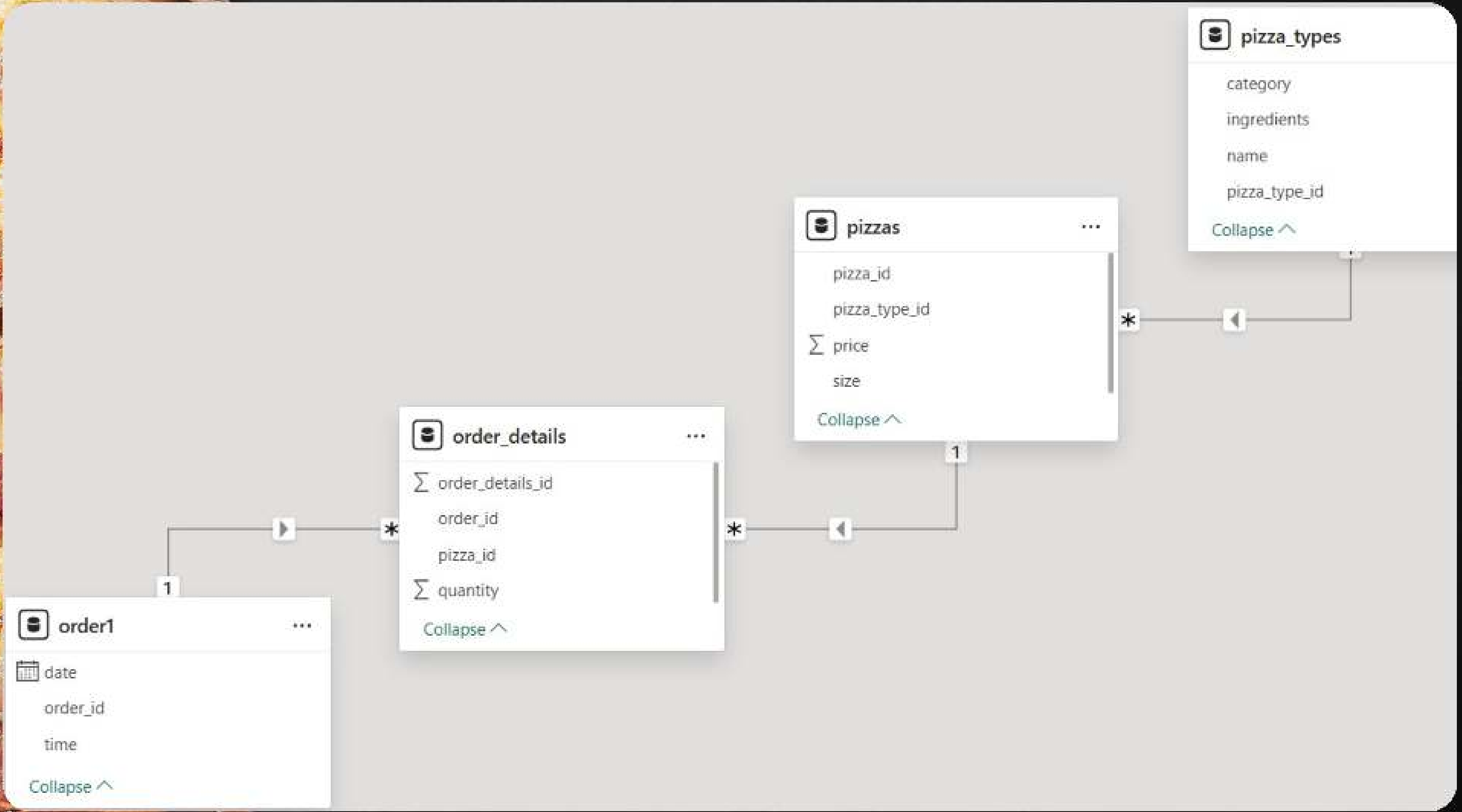
SQL PROJECT ON PIZZA SALES

Ciao! I'm Ushoshi Bose, an MBA student specializing in Business Analytics. This SQL project analyzes pizza sales data to uncover key insights on revenue, order trends, and customer preferences.

It involves data extraction and querying to generate actionable business insights.

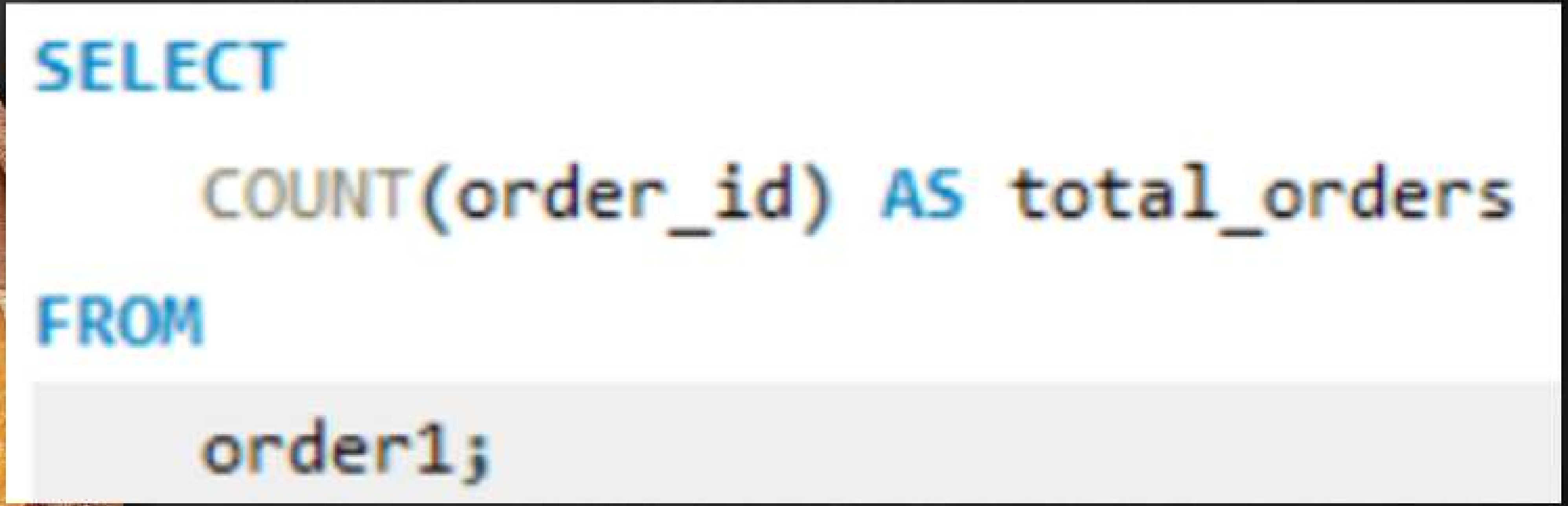
DATA MODEL VIEW



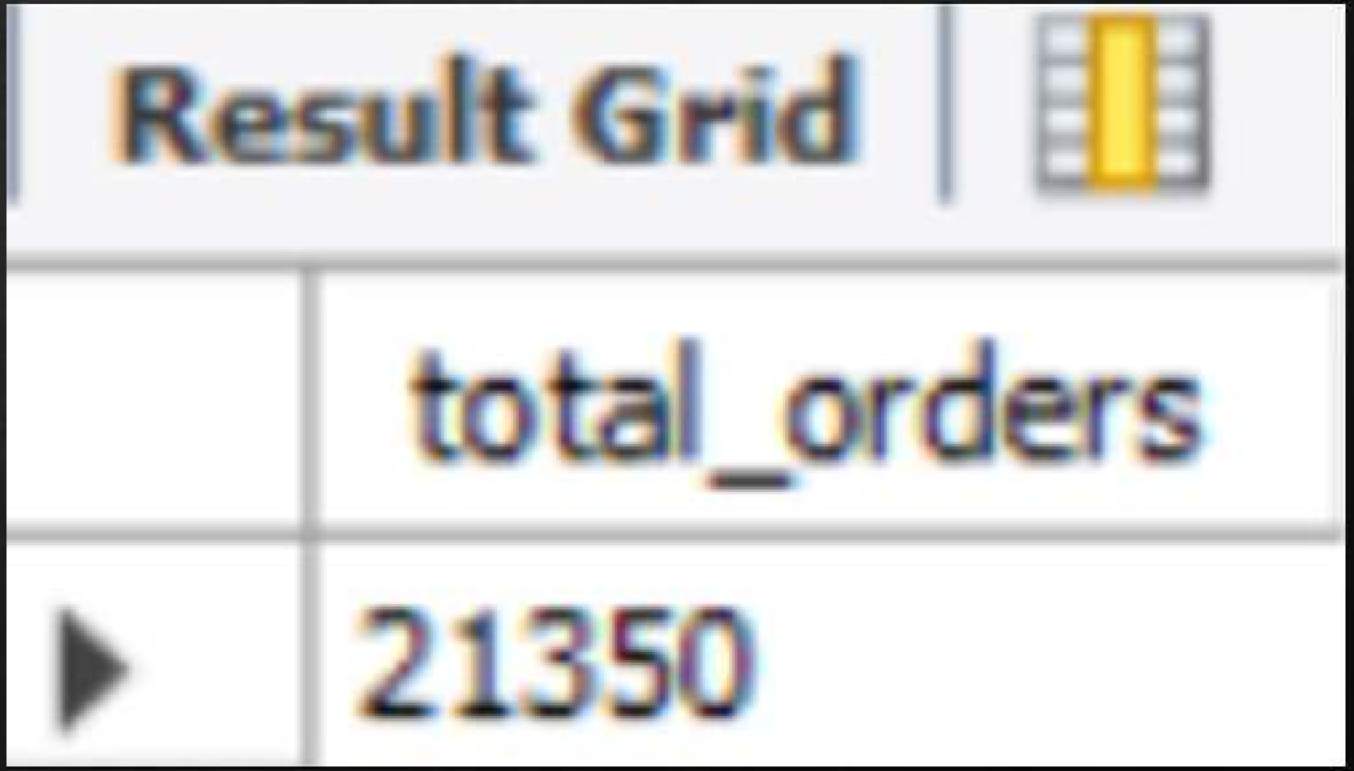
The image shows an Entity Relationship Diagram (ERD) of a pizza sales database in Power BI or a similar tool, depicting relationships between four tables: order1, order_details, pizzas, and pizza_types. The order_details table acts as a bridge between order1 and pizzas, while pizzas is linked to pizza_types via pizza_type_id.

BASIC QUERIES

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



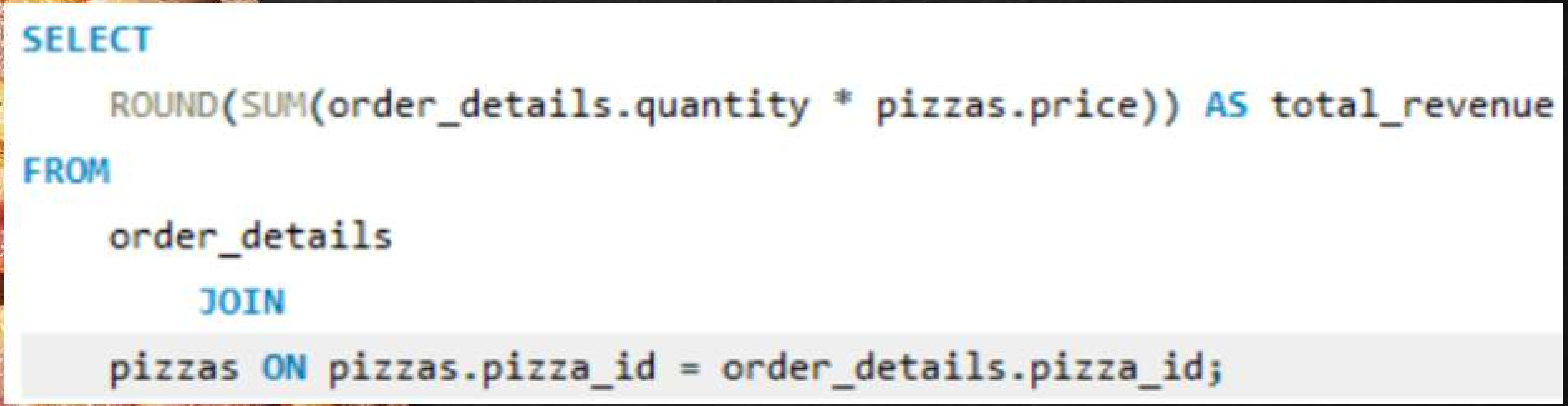
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    order1;
```



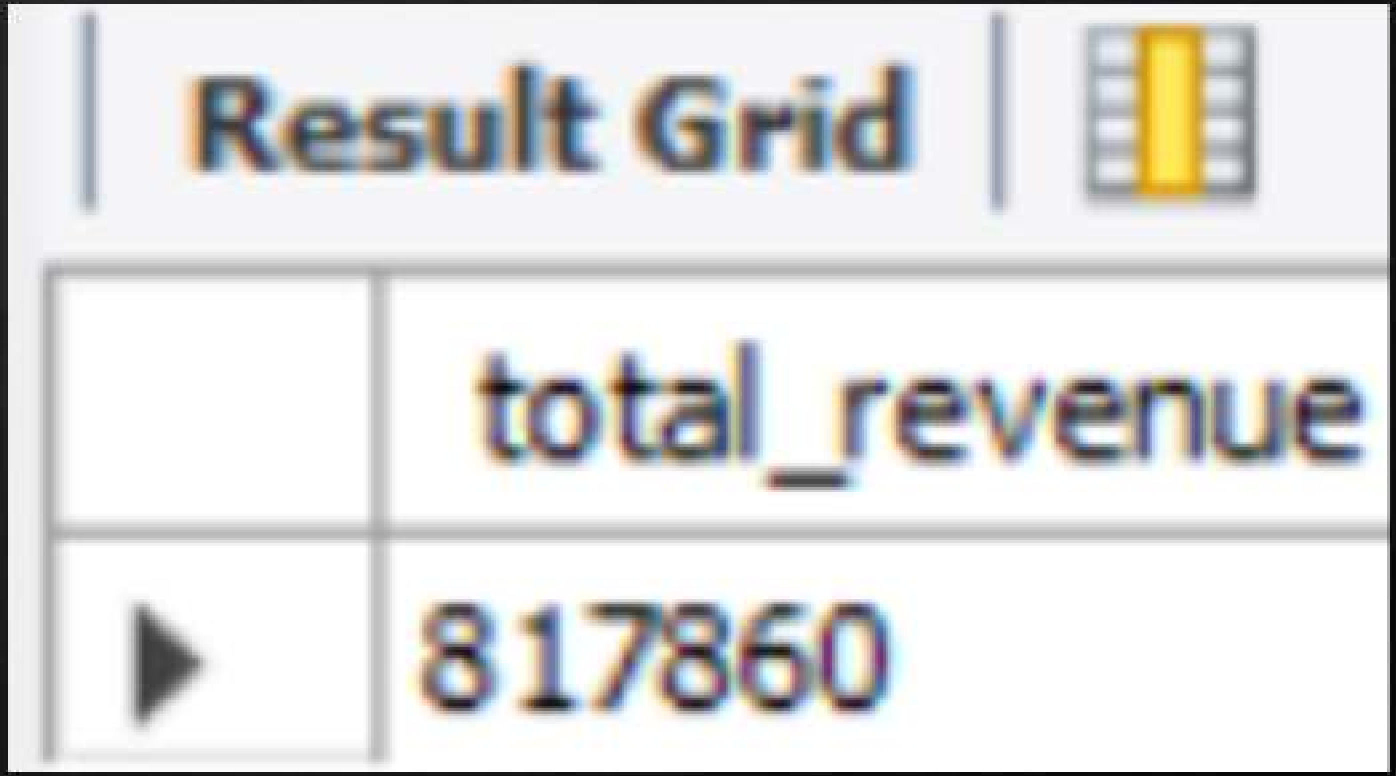
```
total_orders
21350
```

BASIC QUERIES

2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



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

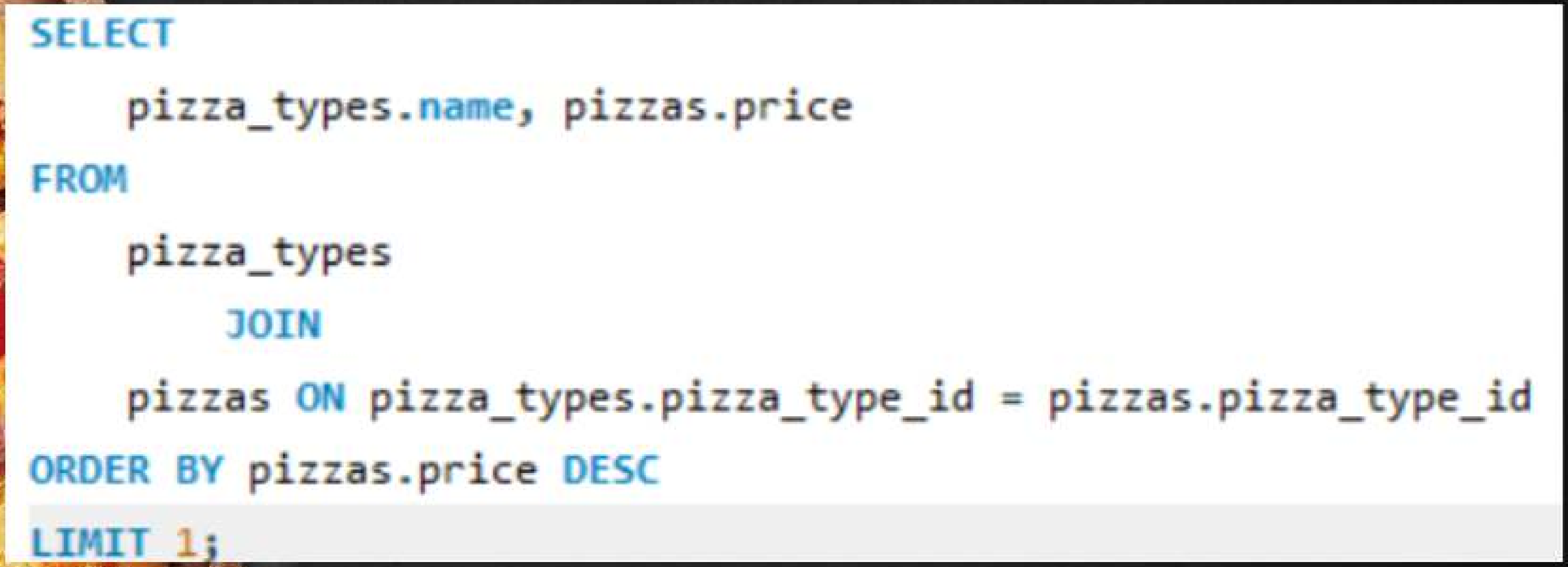


Result Grid

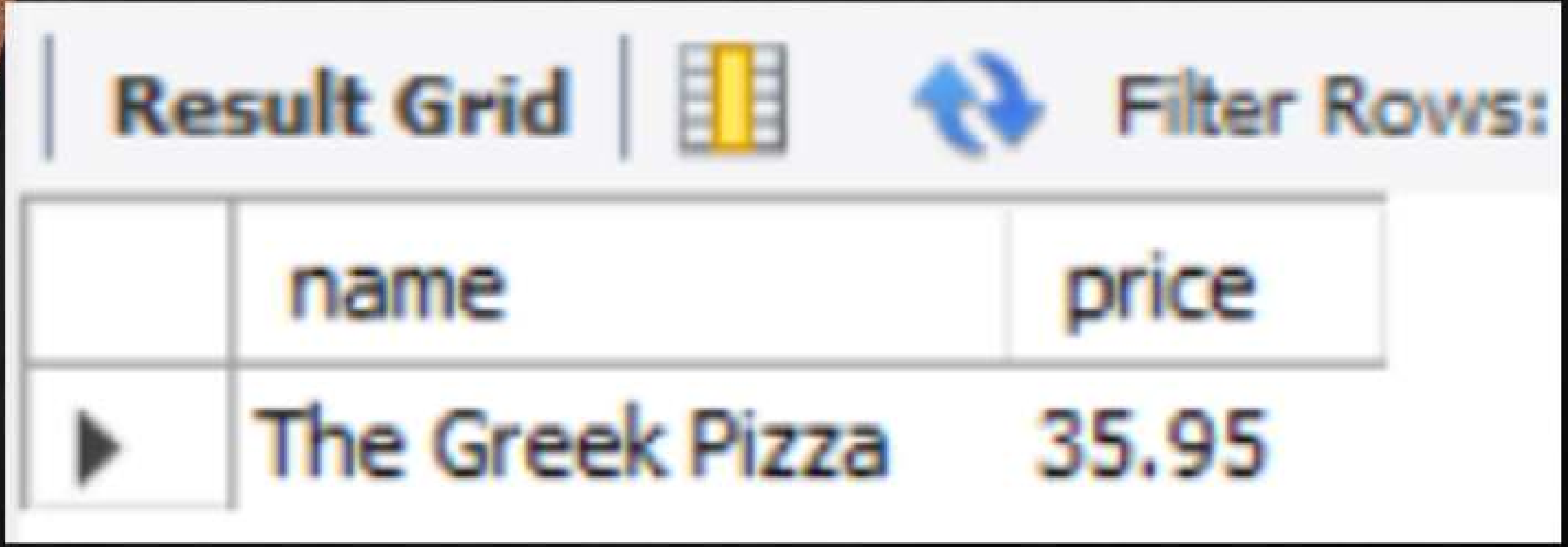
	total_revenue
▶	817860

BASIC QUERIES

3.IDENTIFY THE HIGHEST-PRICED PIZZA.



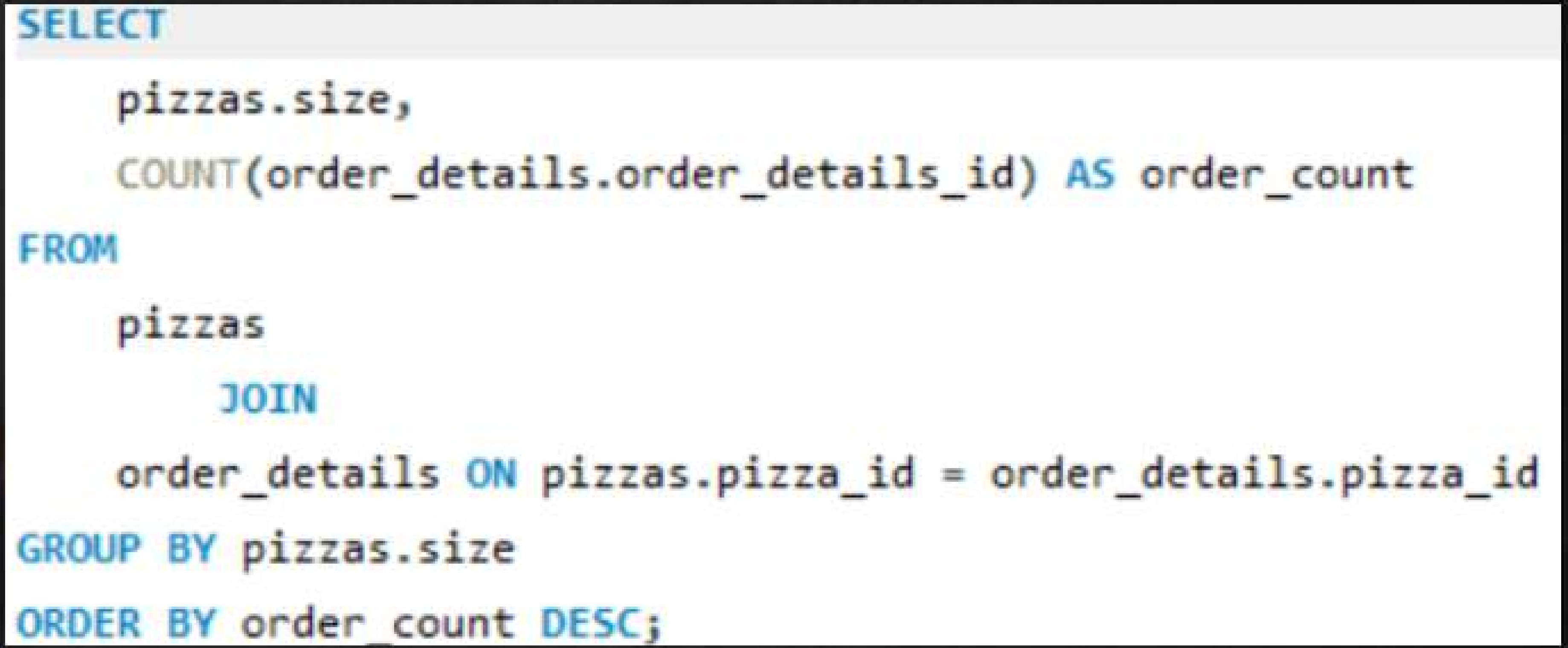
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



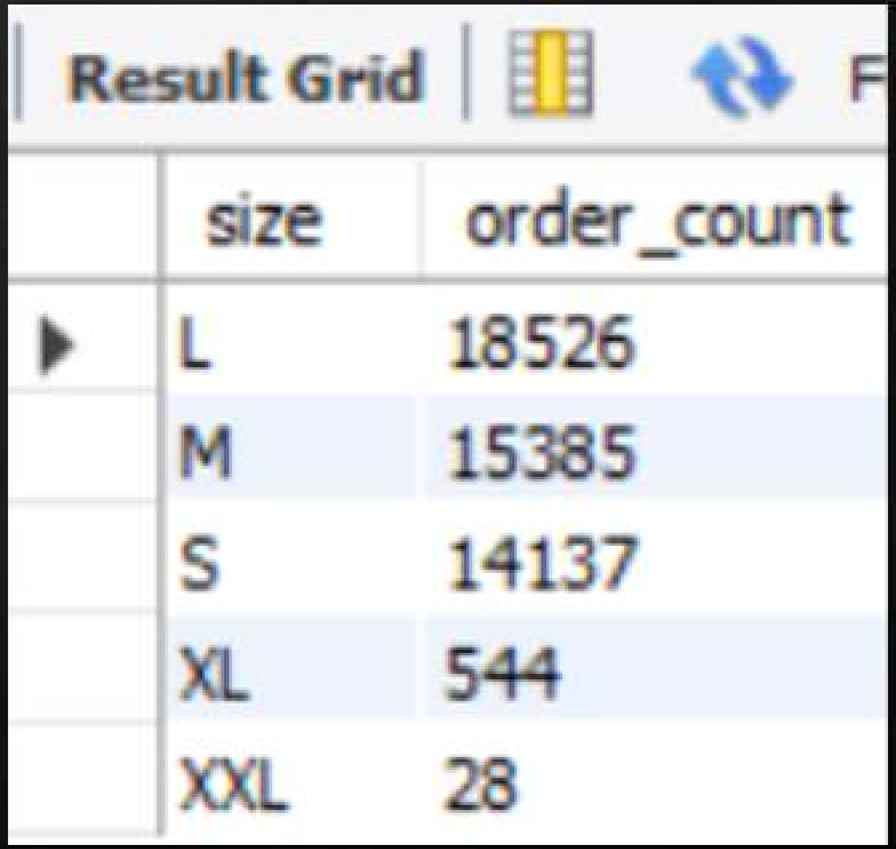
```
name              price
The Greek Pizza   35.95
```

BASIC QUERIES

4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



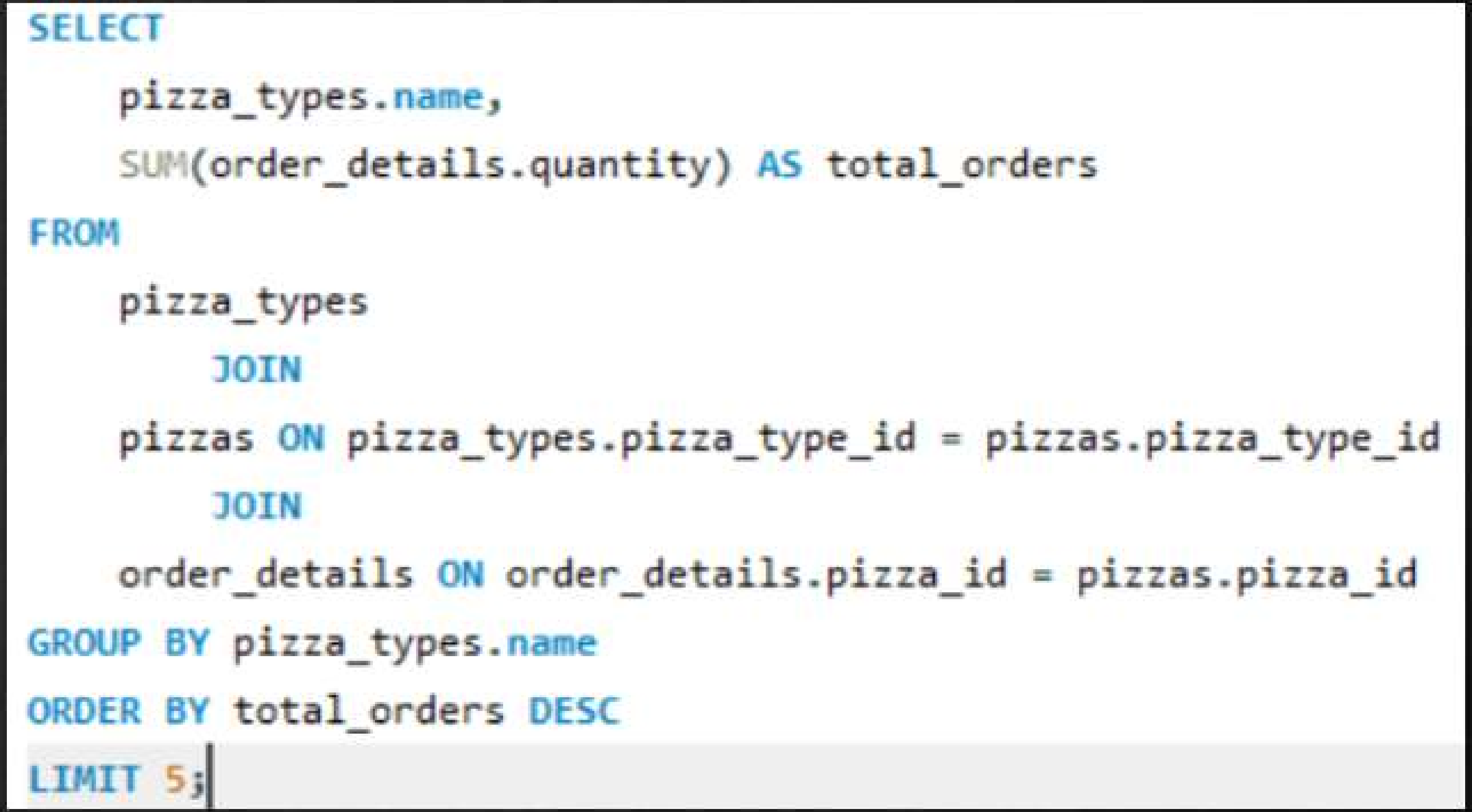
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



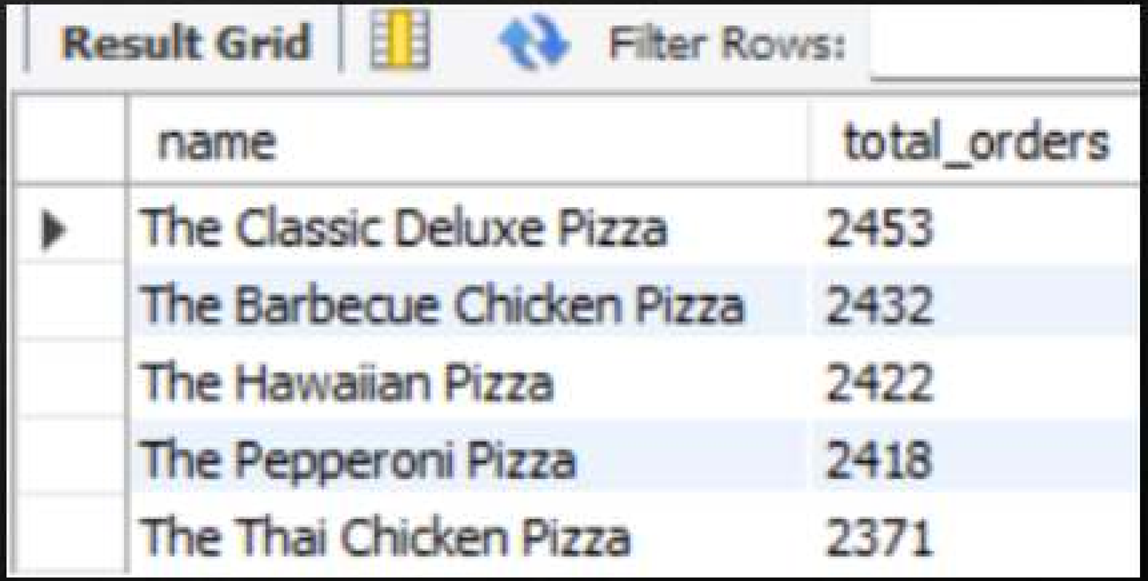
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

BASIC QUERIES

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



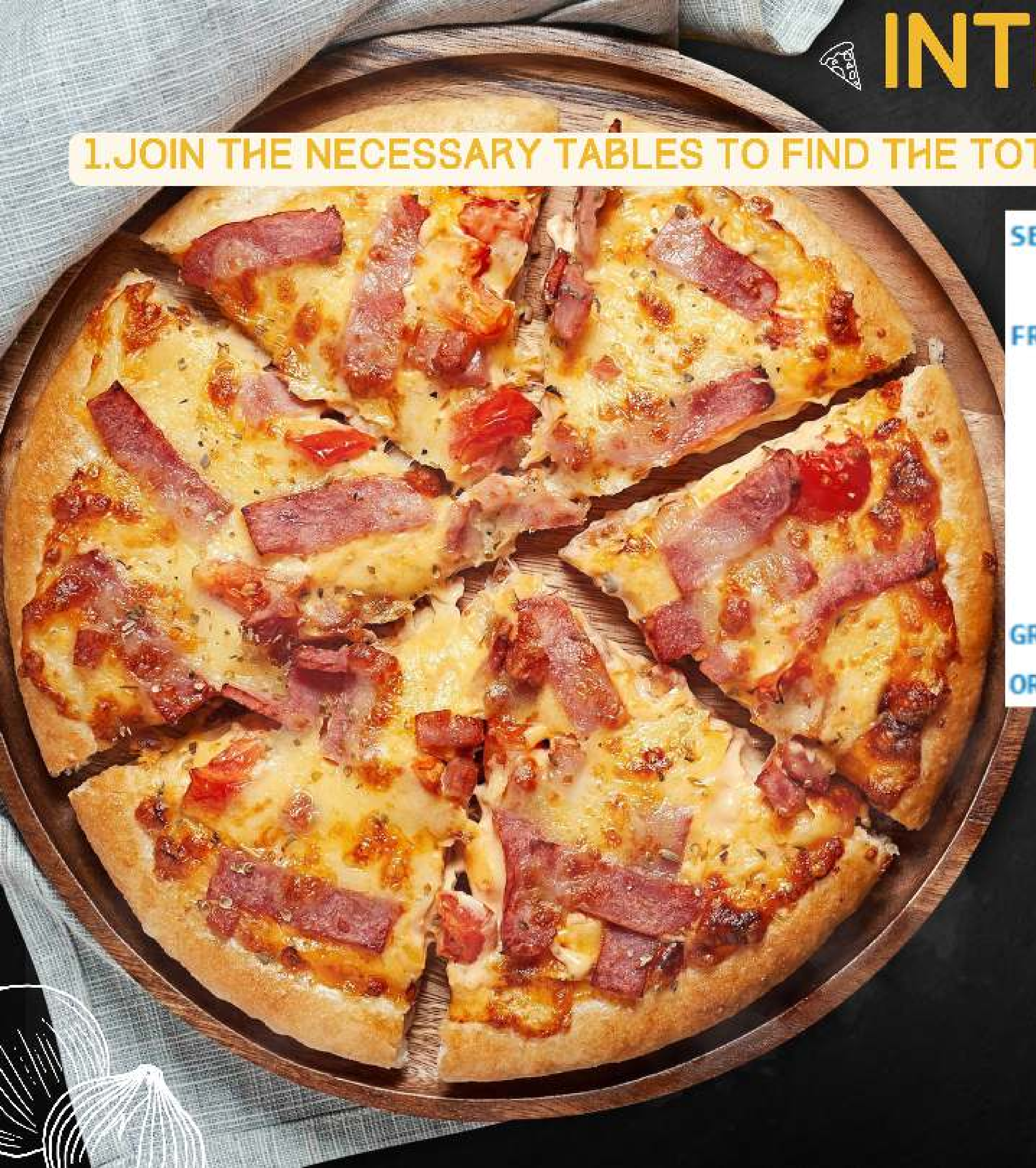
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_orders
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_orders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

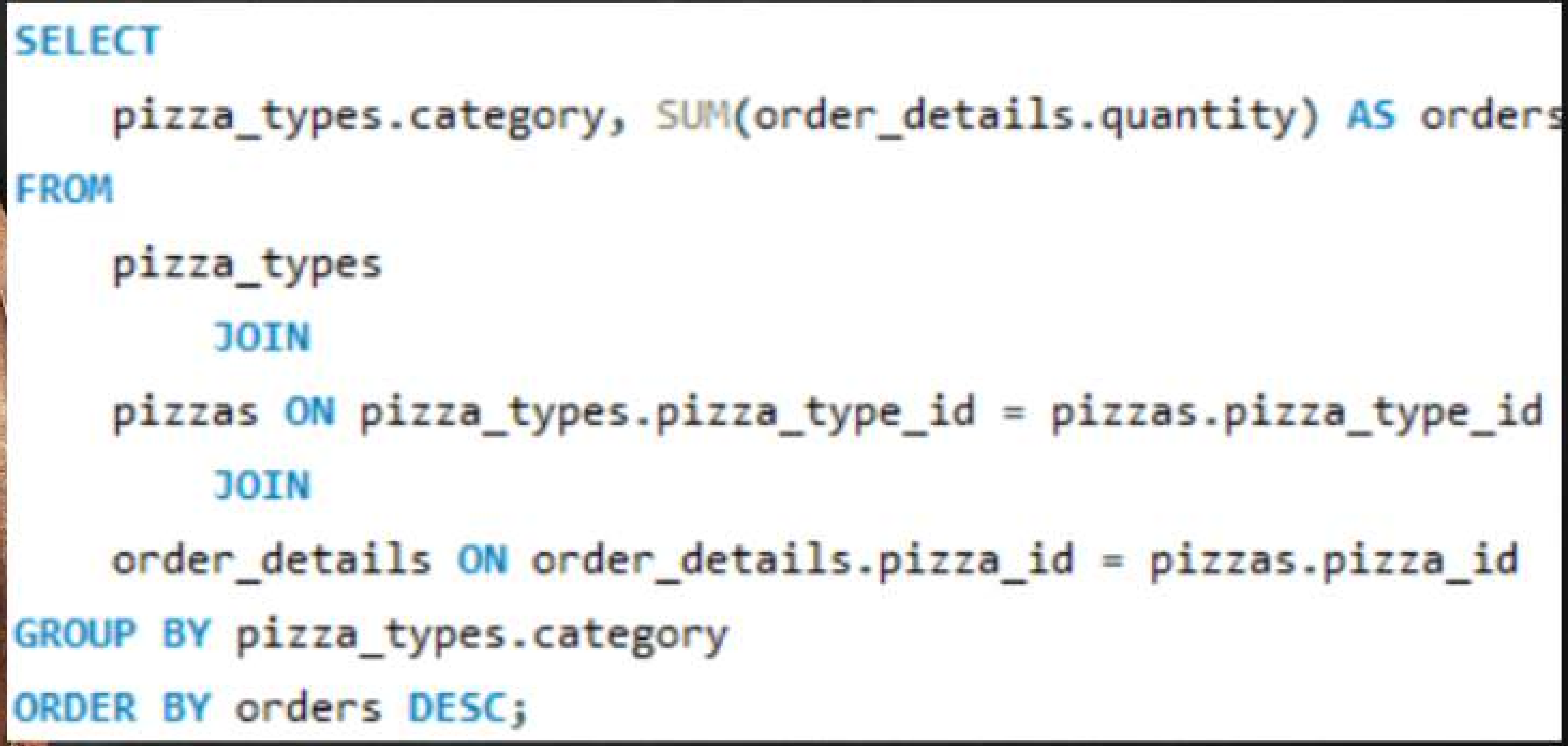


```
name                          total_orders
The Classic Deluxe Pizza      2453
The Barbecue Chicken Pizza    2432
The Hawaiian Pizza            2422
The Pepperoni Pizza           2418
The Thai Chicken Pizza        2371
```

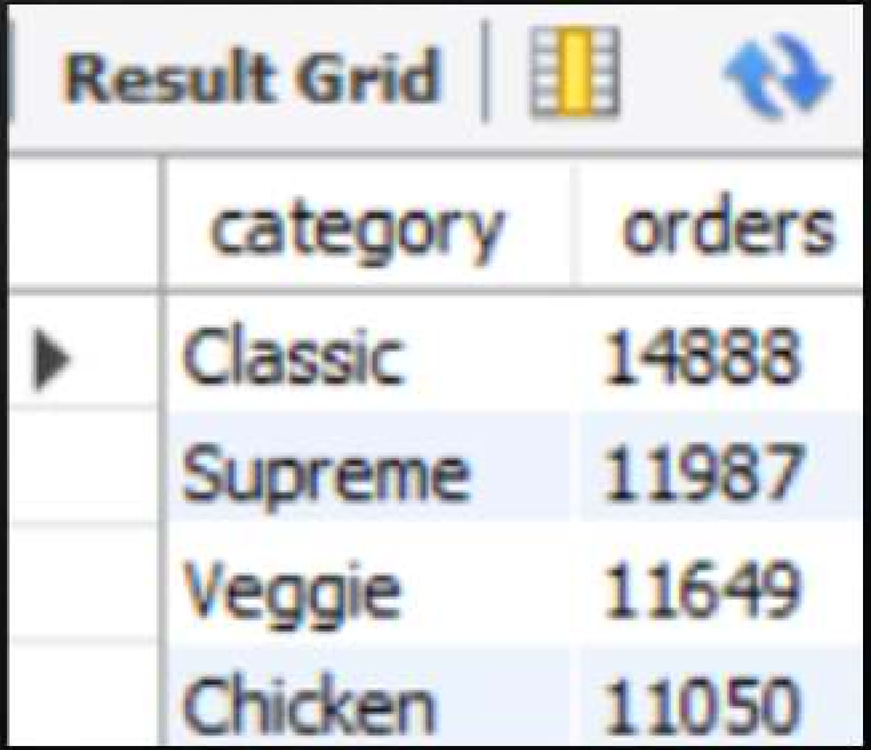
INTERMEDIATE QUERIES

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





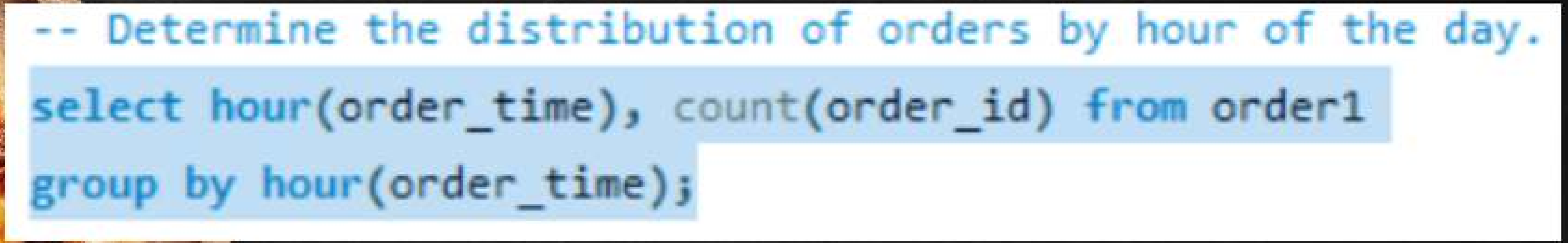
```
SELECT
    pizza_types.category, SUM(order_details.quantity) AS orders
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY orders DESC;
```



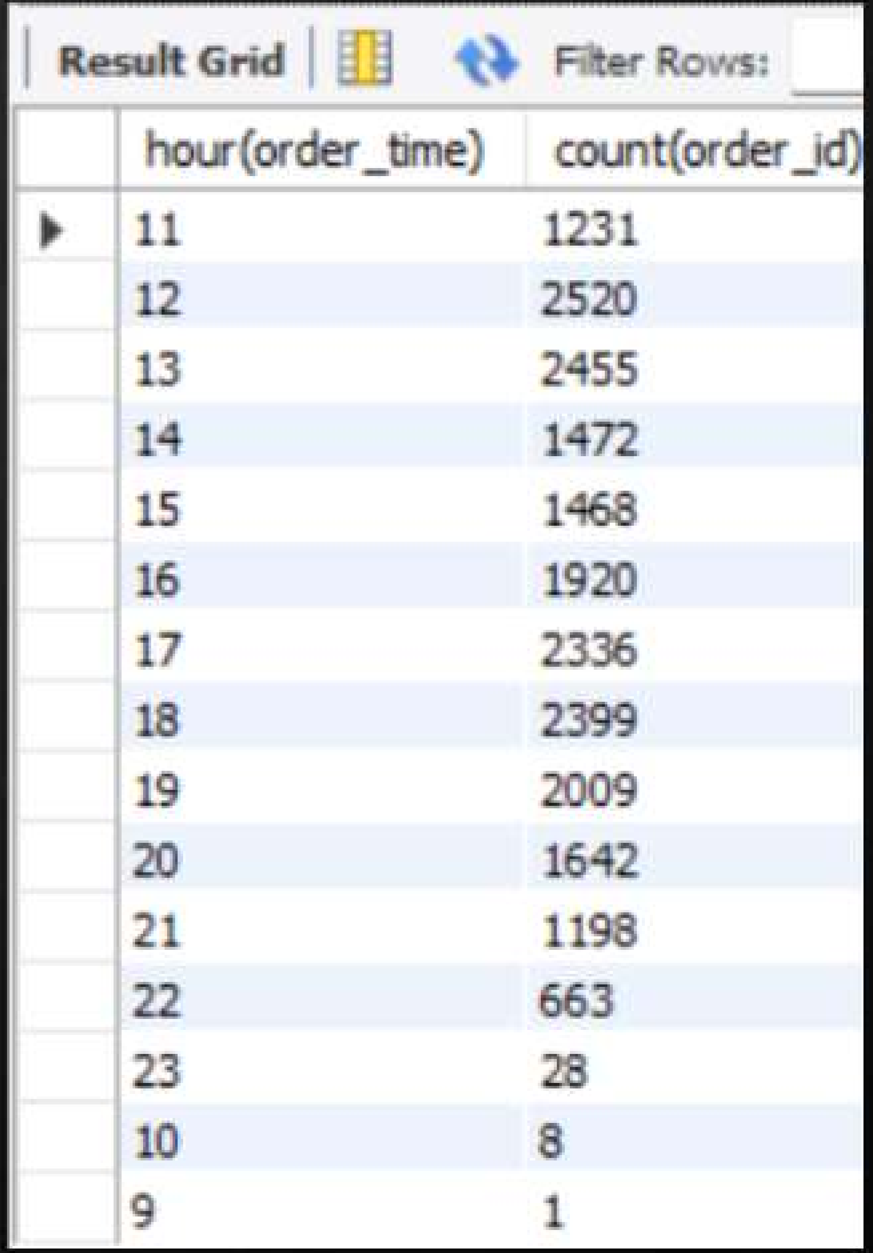
```
category   orders
Classic    14888
Supreme    11987
Veggie     11649
Chicken    11050
```

INTERMEDIATE QUERIES

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



```
-- Determine the distribution of orders by hour of the day.
select hour(order_time), count(order_id) from order1
group by hour(order_time);
```

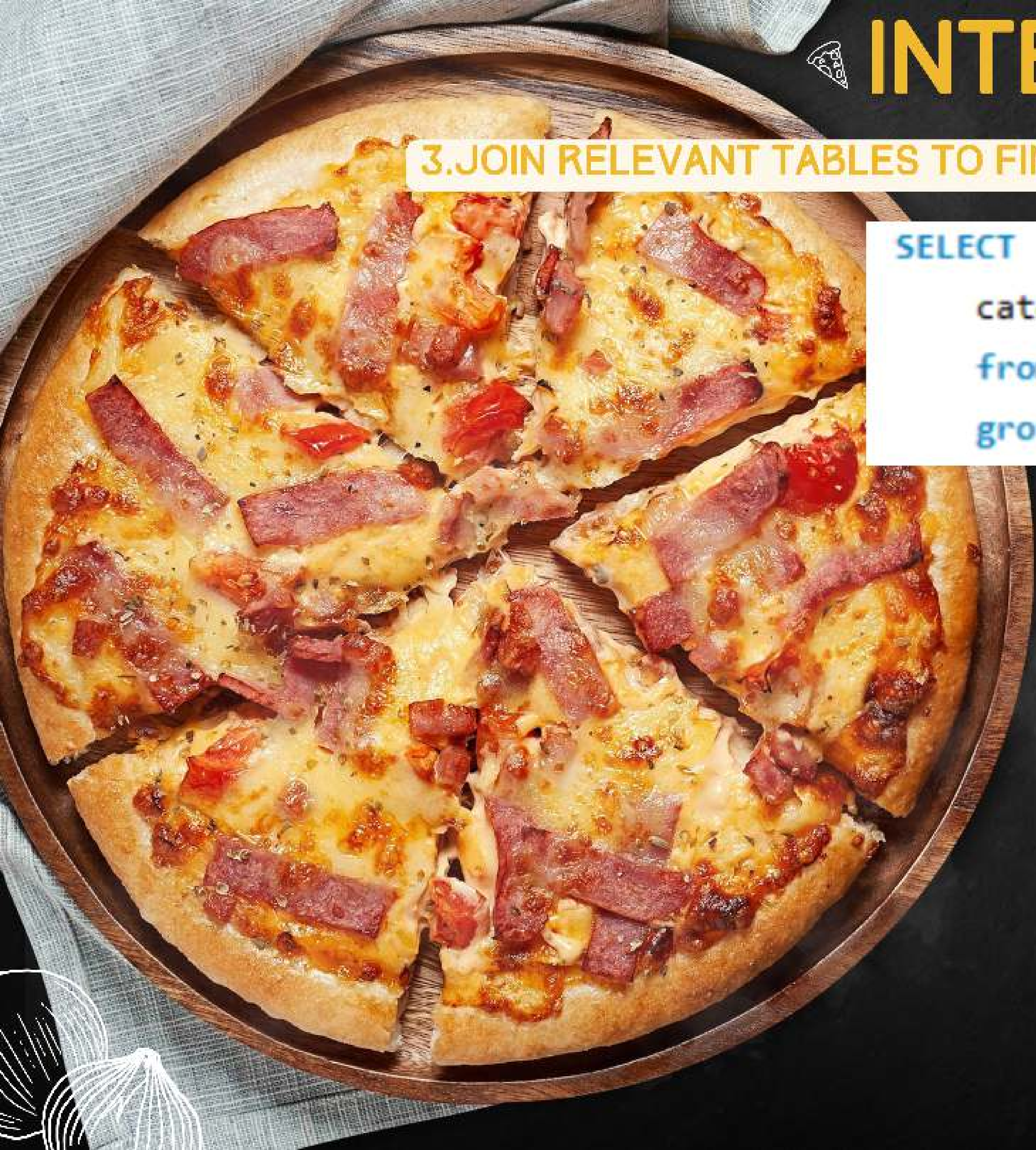


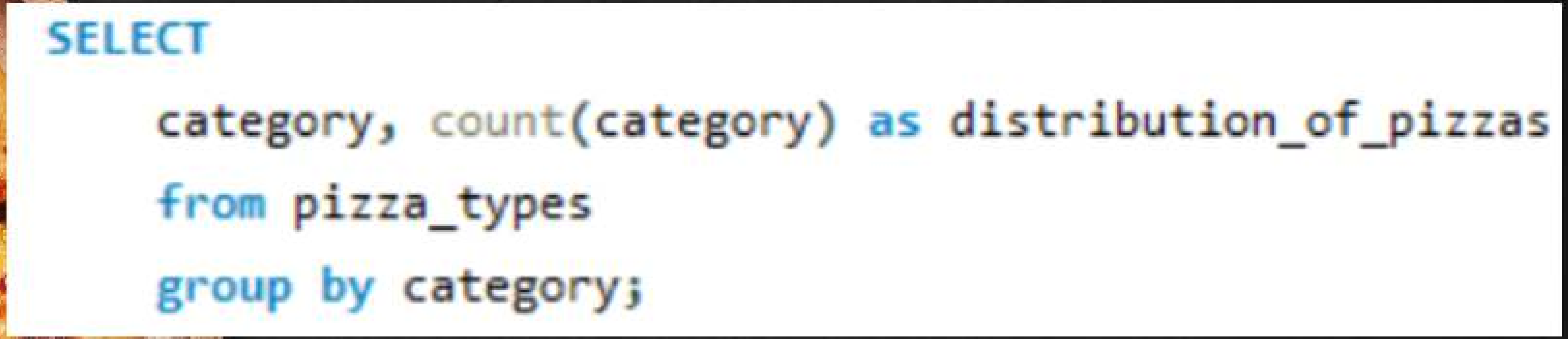
Result Grid
Filter Rows:

hour(order_time)	count(order_id)
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

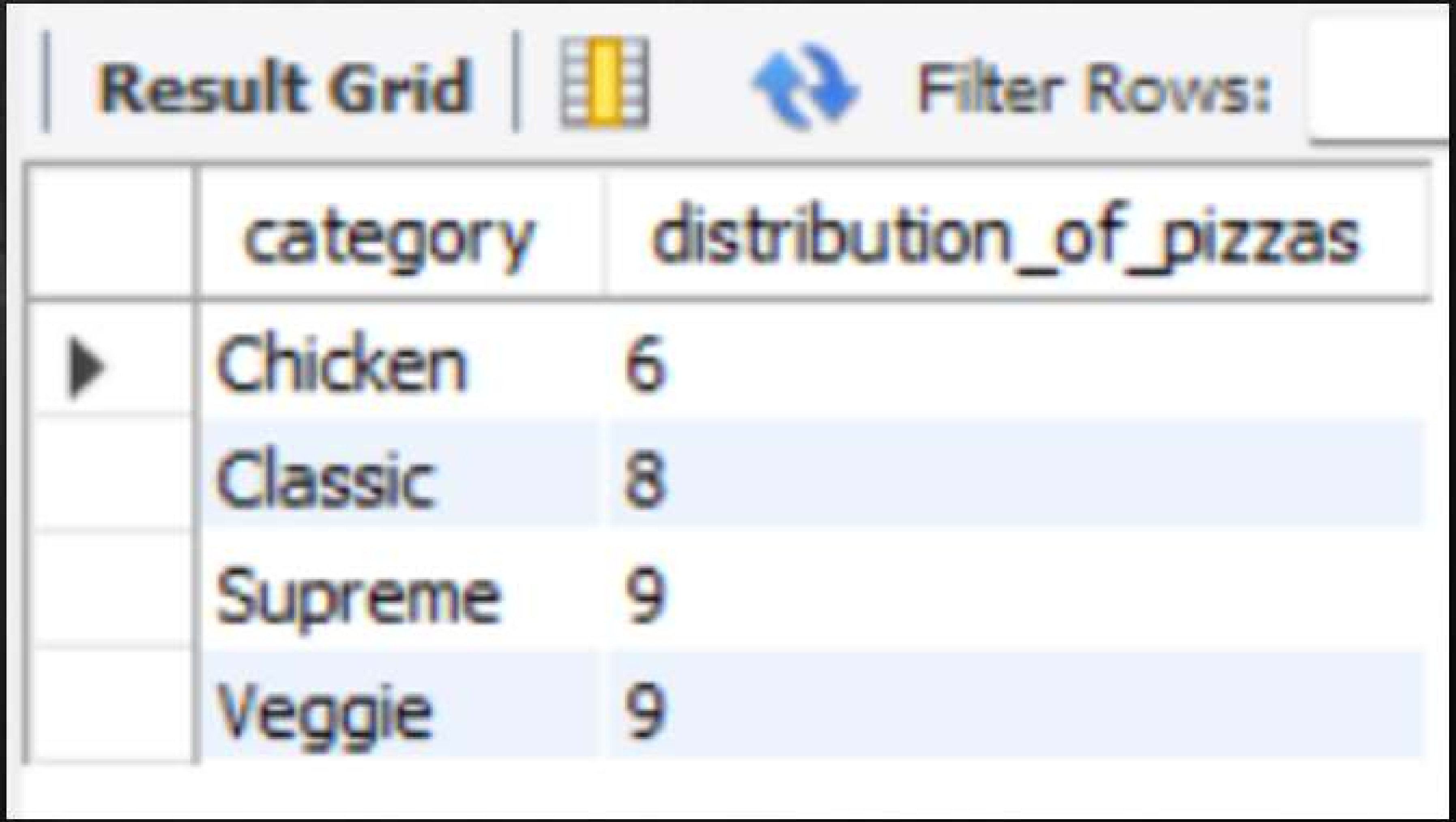
INTERMEDIATE QUERIES

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





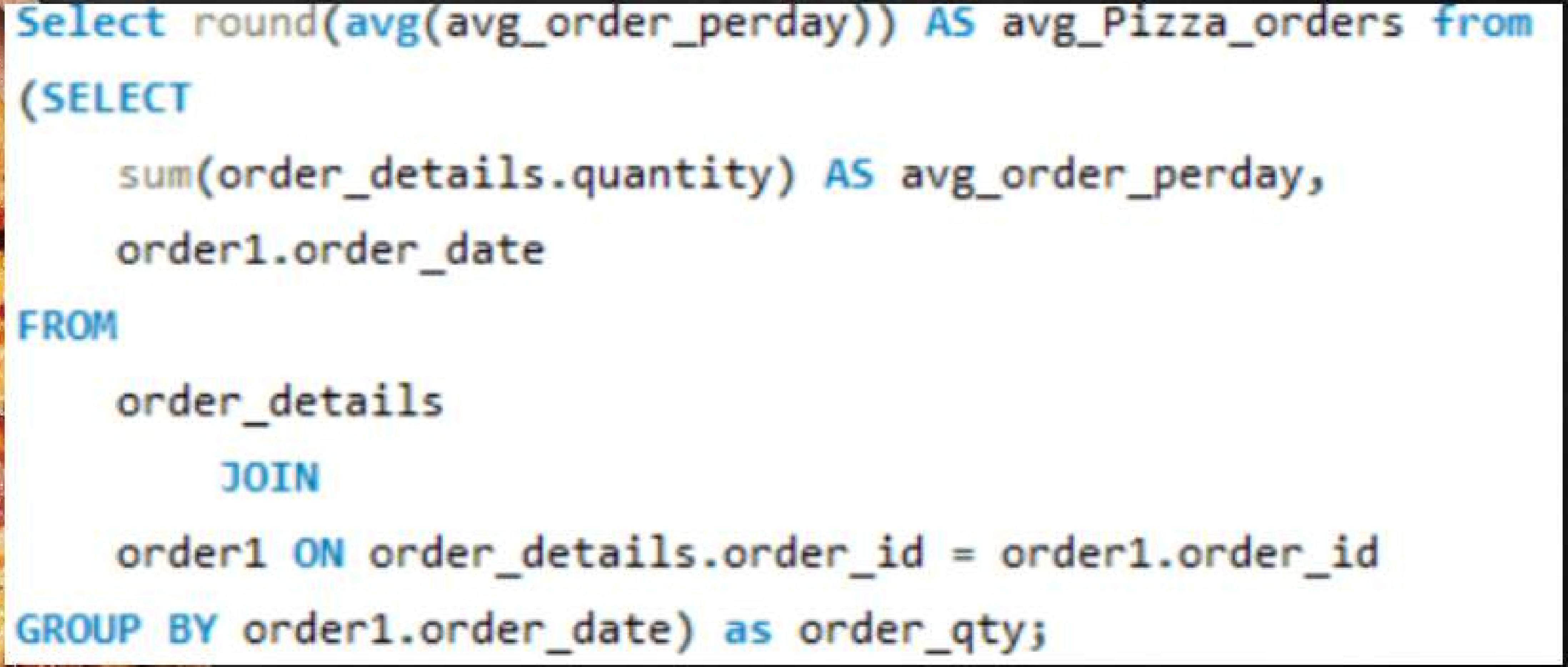
```
SELECT
    category, count(category) as distribution_of_pizzas
    from pizza_types
    group by category;
```



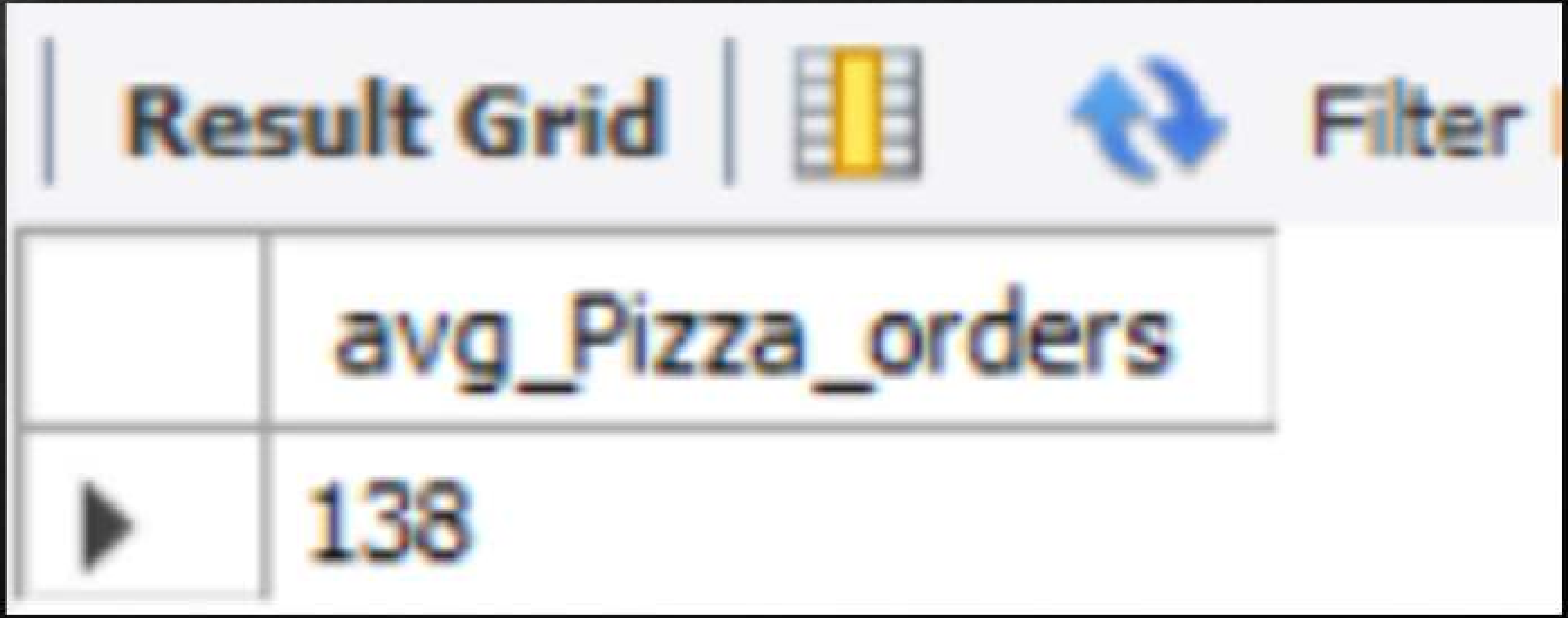
category	distribution_of_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

INTERMEDIATE QUERIES

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



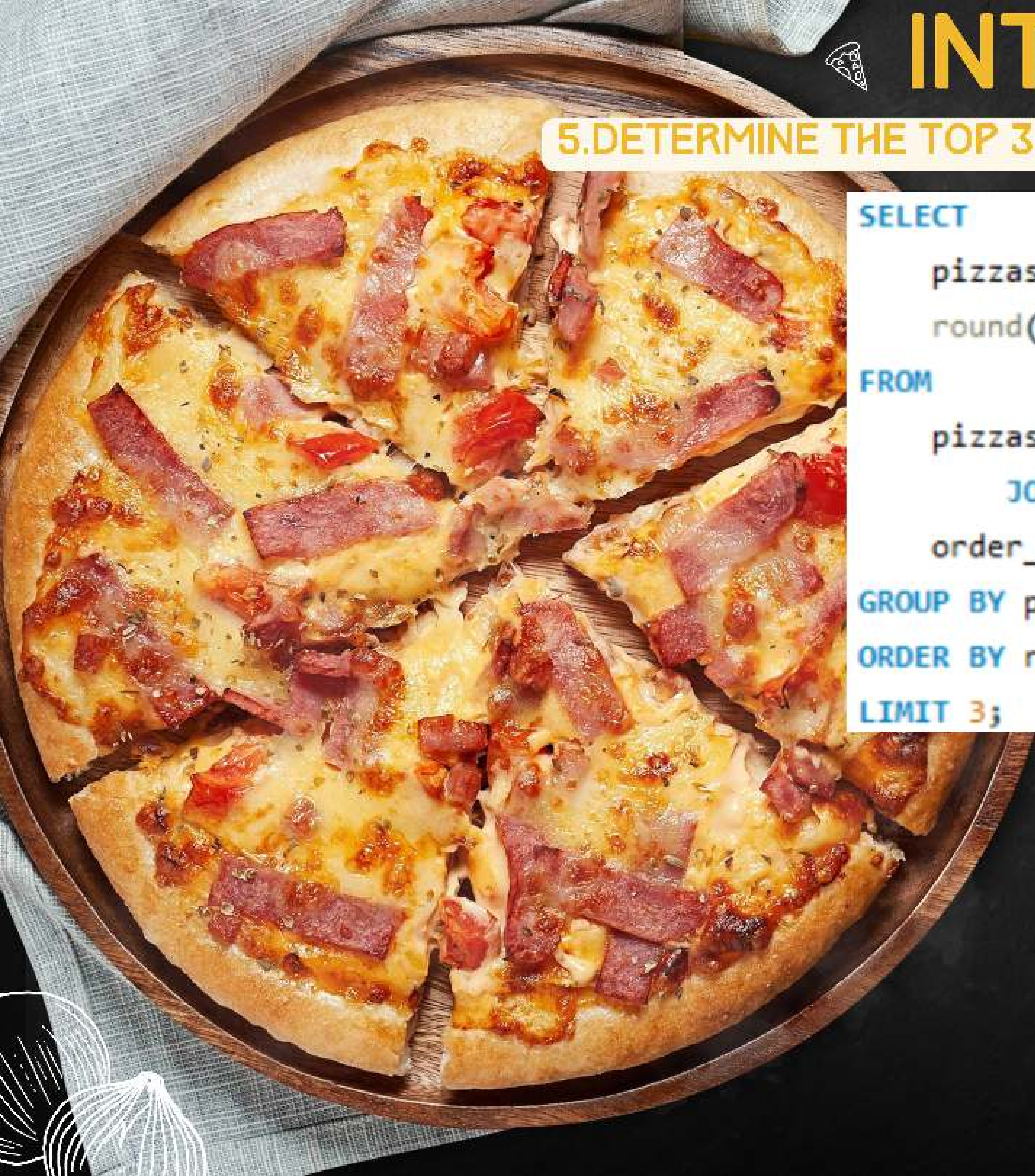
```
Select round(avg(avg_order_perday)) AS avg_Pizza_orders from
(SELECT
    sum(order_details.quantity) AS avg_order_perday,
    order1.order_date
FROM
    order_details
        JOIN
    order1 ON order_details.order_id = order1.order_id
GROUP BY order1.order_date) as order_qty;
```

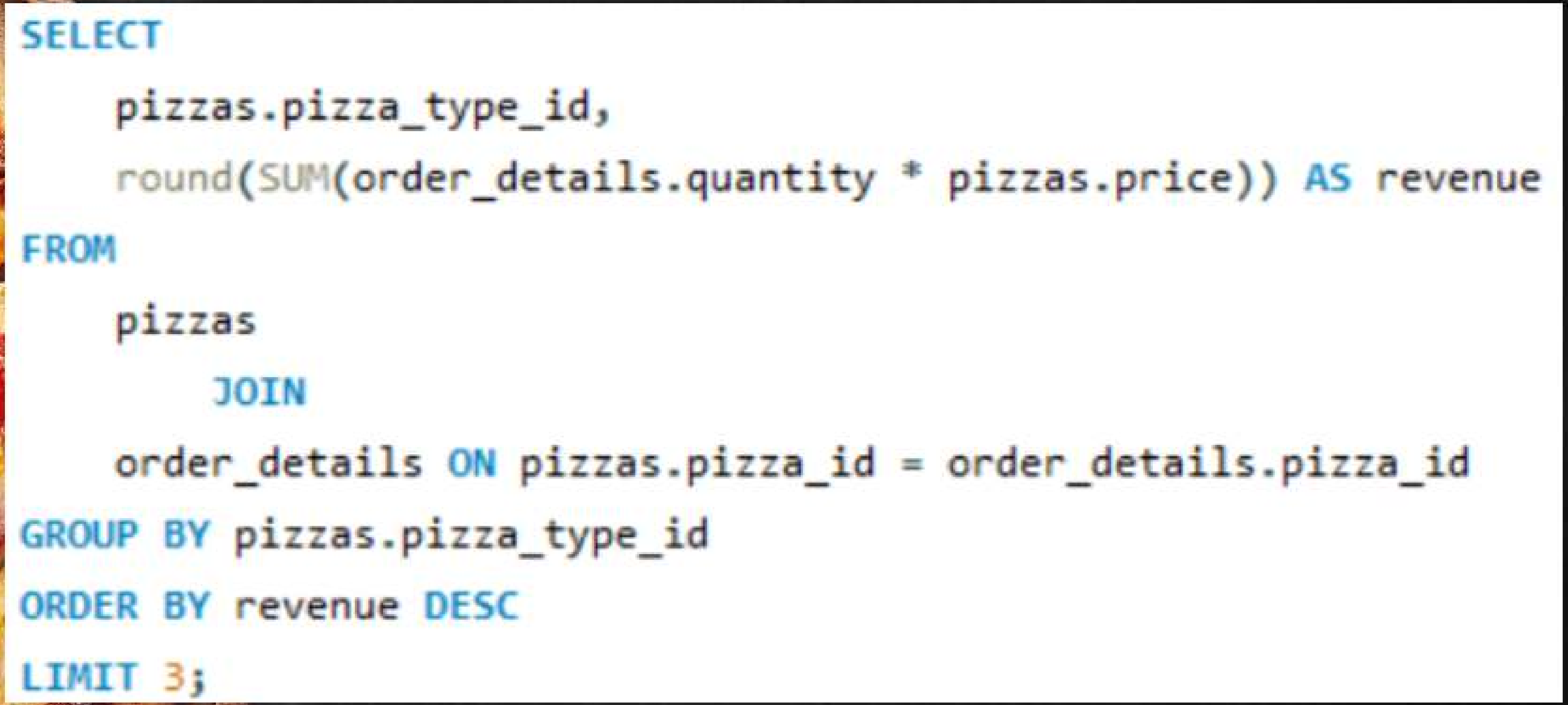


```
avg_Pizza_orders
138
```

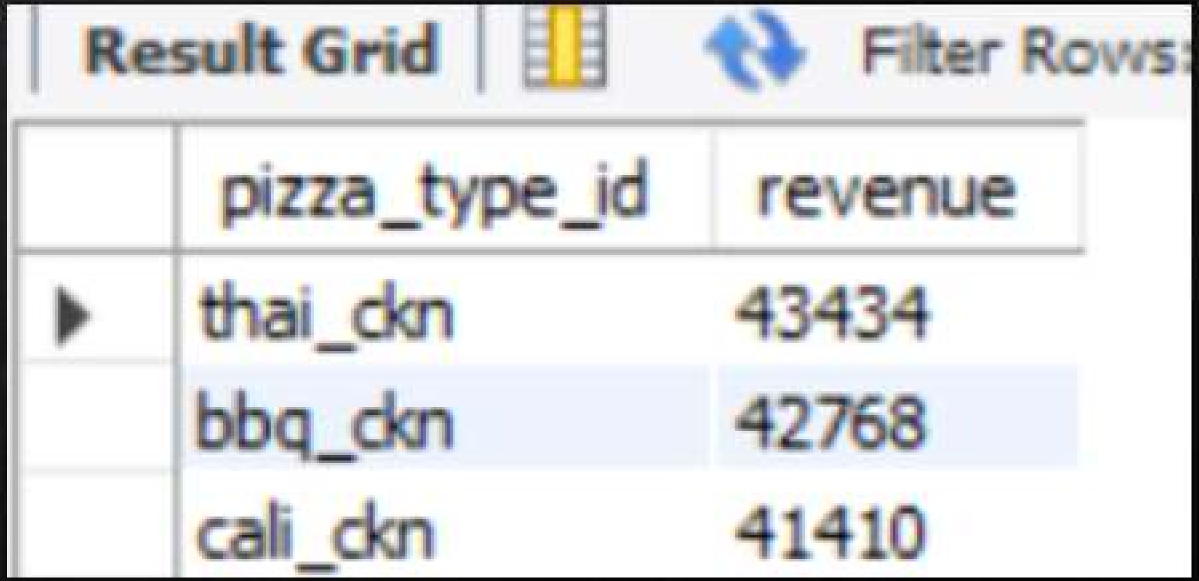
INTERMEDIATE QUERIES

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





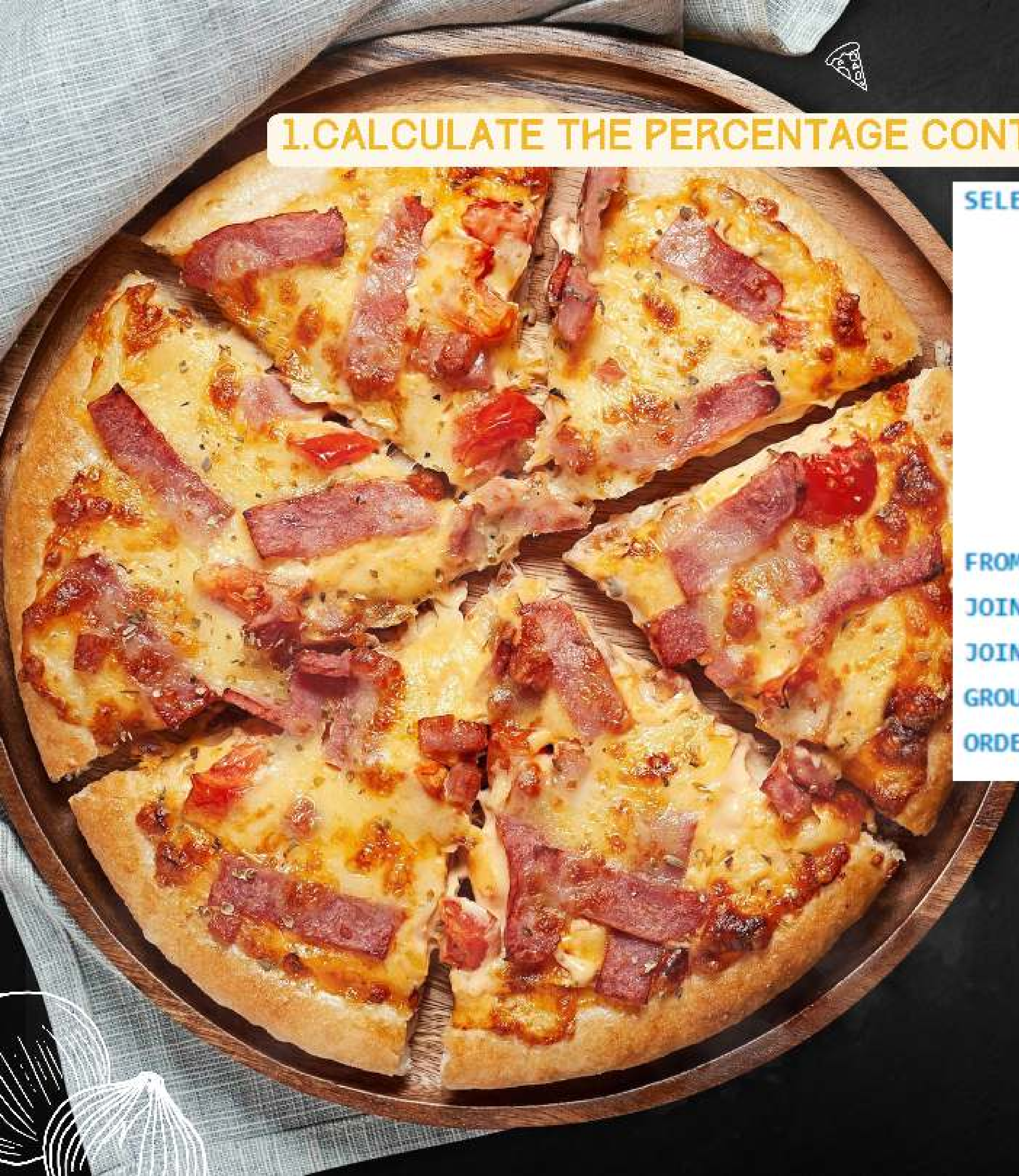
```
SELECT
    pizzas.pizza_type_id,
    round(SUM(order_details.quantity * pizzas.price)) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.pizza_type_id
ORDER BY revenue DESC
LIMIT 3;
```

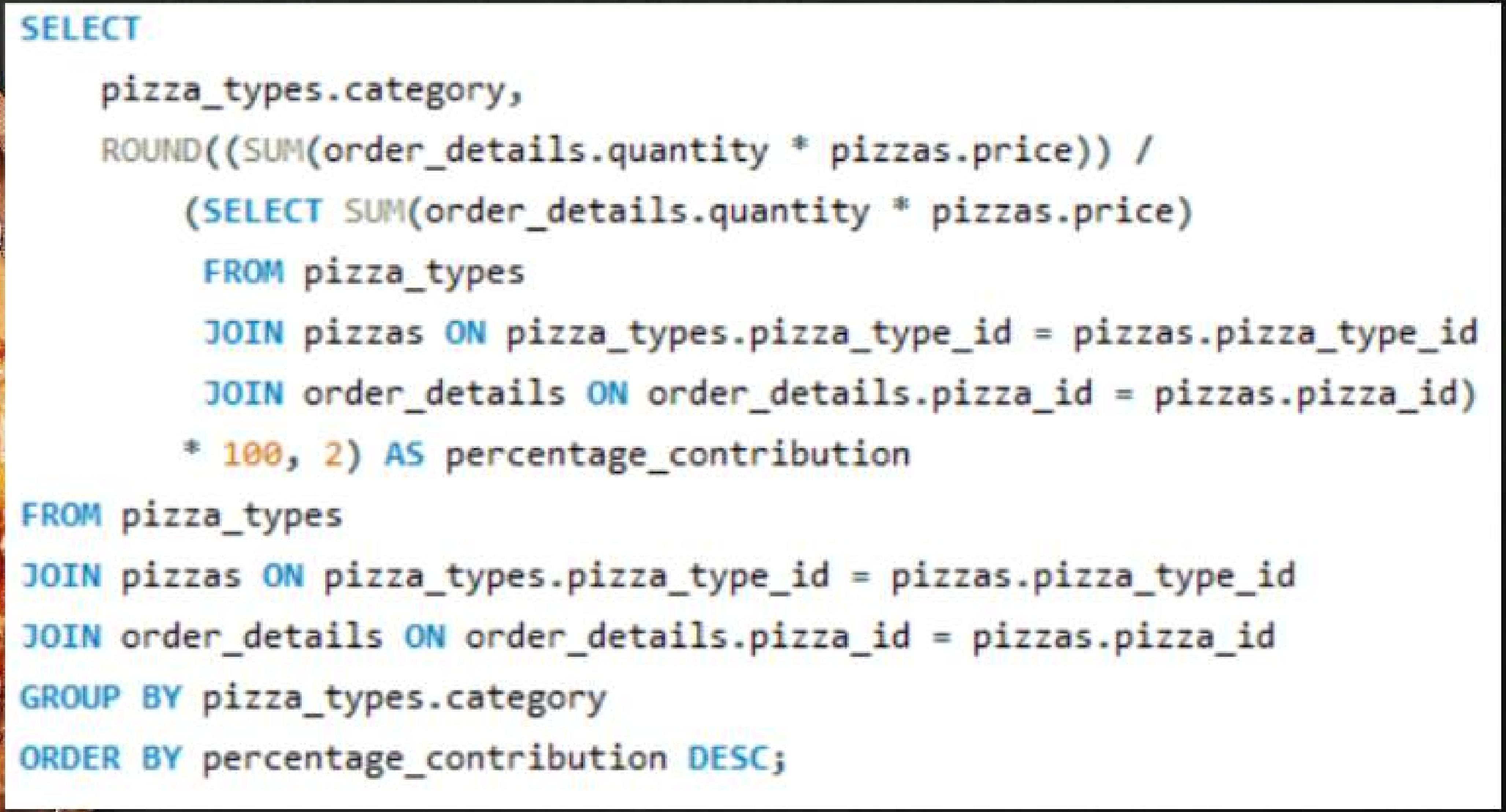


pizza_type_id	revenue
thai_ckn	43434
bbq_ckn	42768
cali_ckn	41410

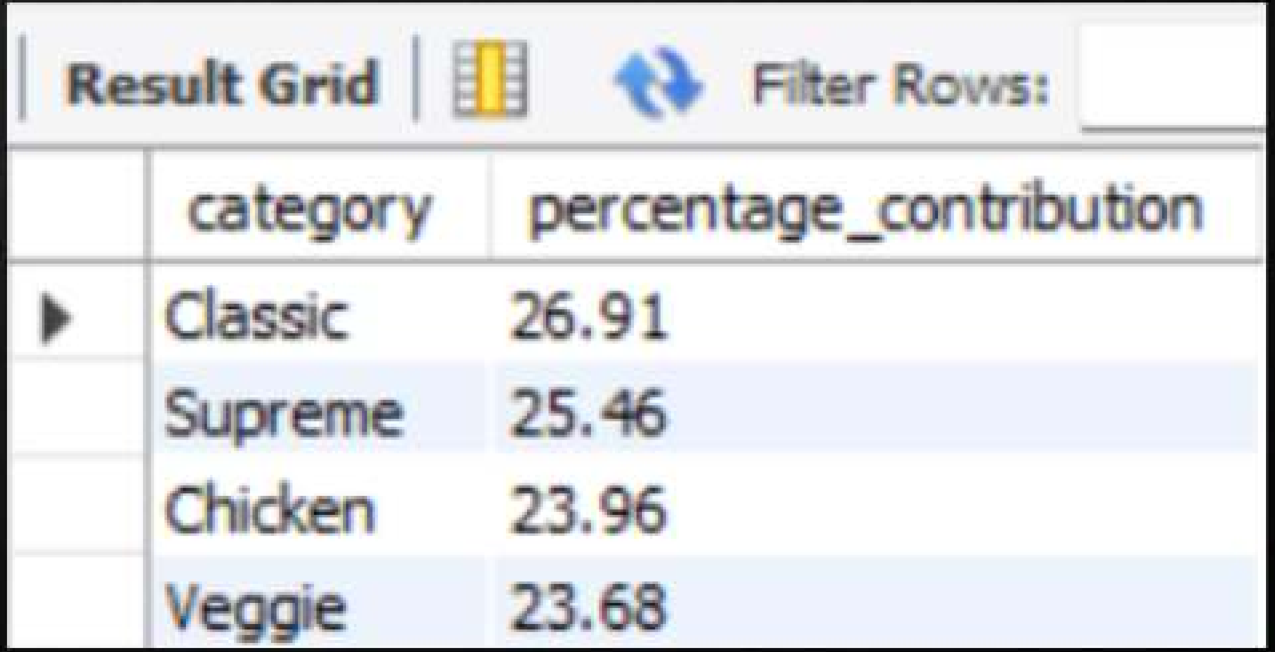
ADVANCED QUERIES

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

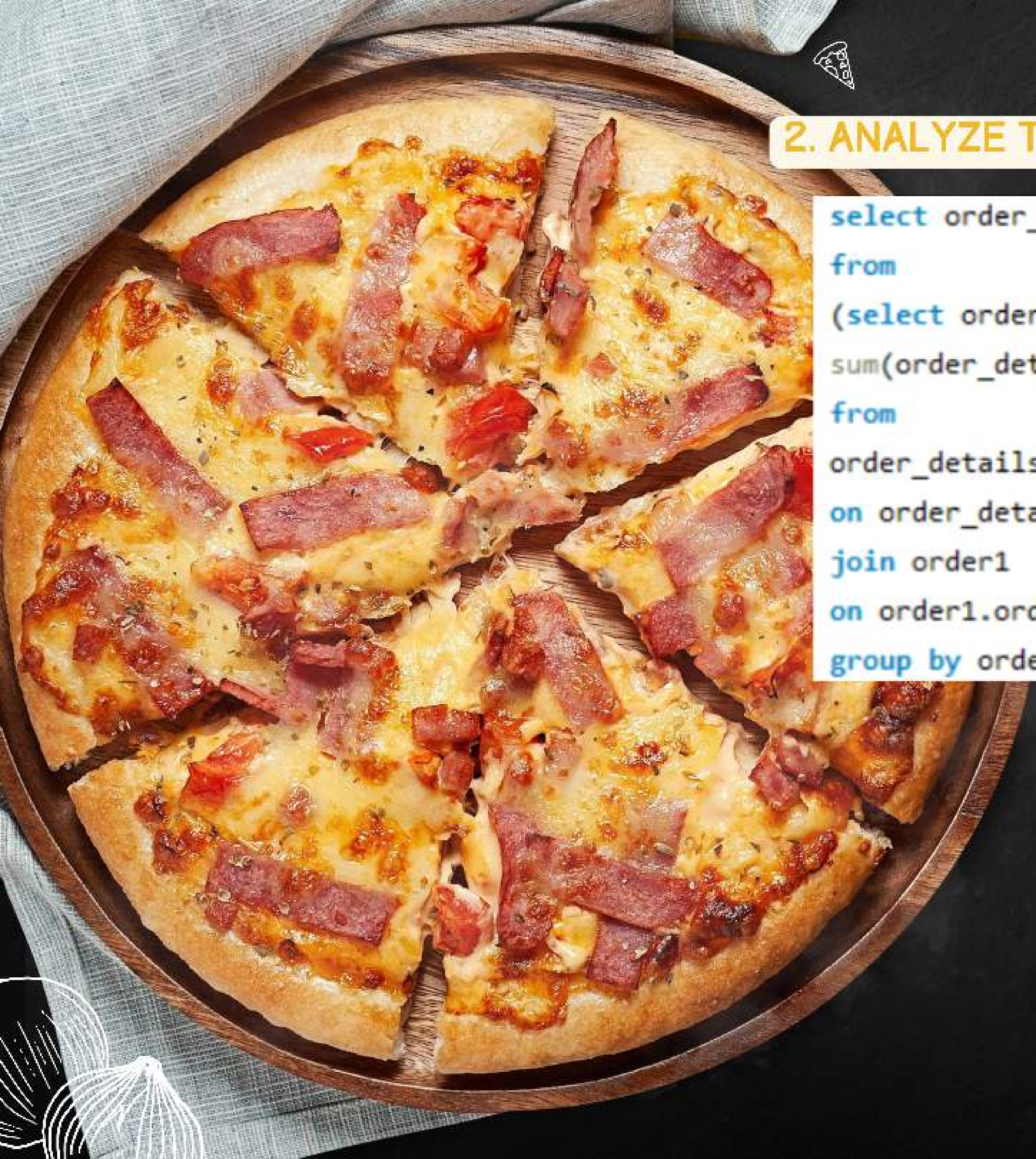




```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price)) /
        (SELECT SUM(order_details.quantity * pizzas.price)
         FROM pizza_types
         JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
         JOIN order_details ON order_details.pizza_id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

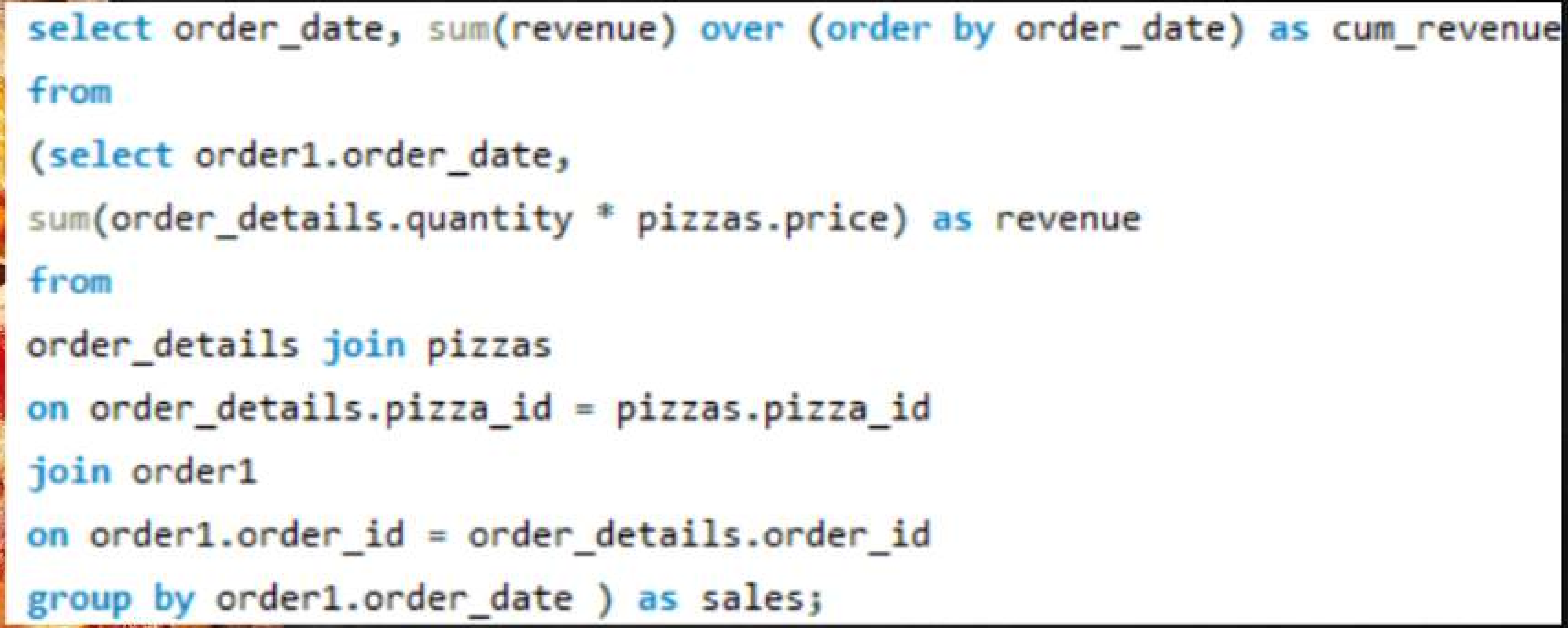


category	percentage_contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

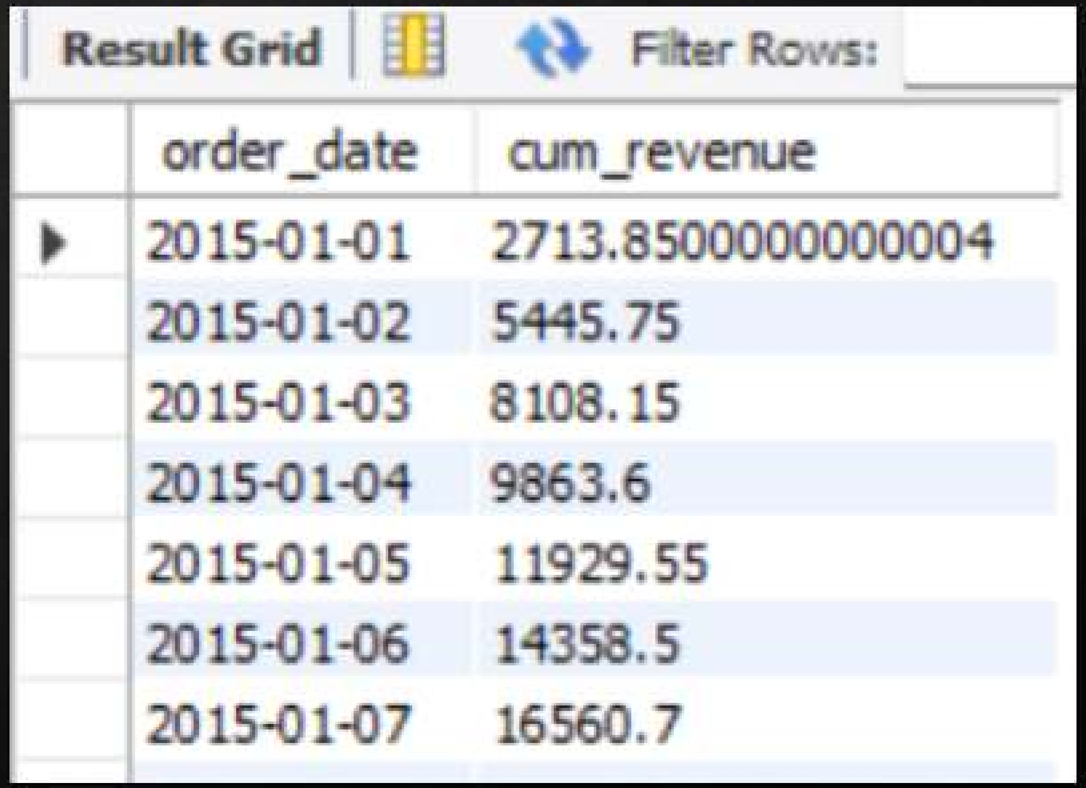


ADVANCED QUERIES

2. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



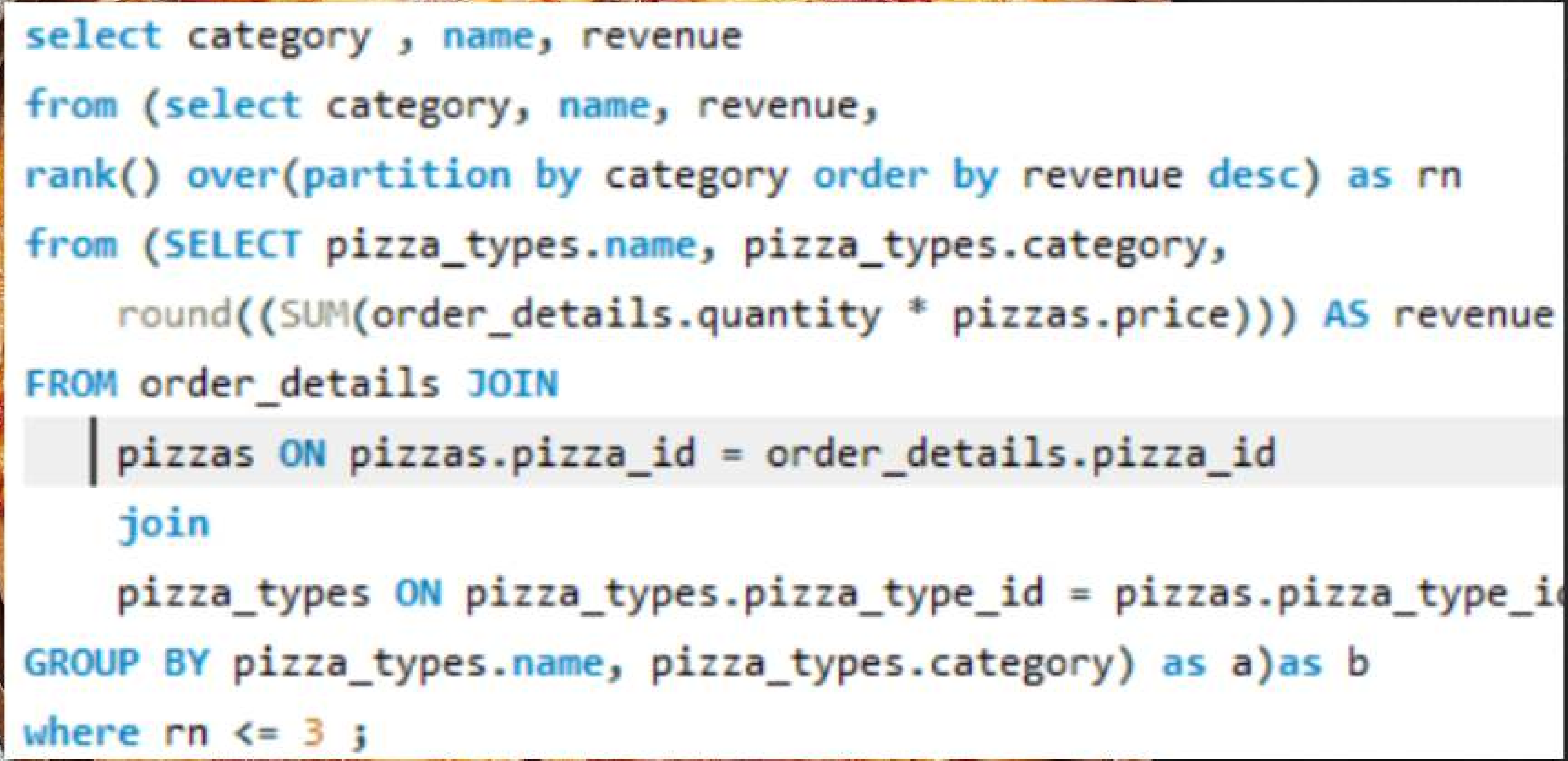
```
select order_date, sum(revenue) over (order by order_date) as cum_revenue
from
(select order1.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from
order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join order1
on order1.order_id = order_details.order_id
group by order1.order_date ) as sales;
```



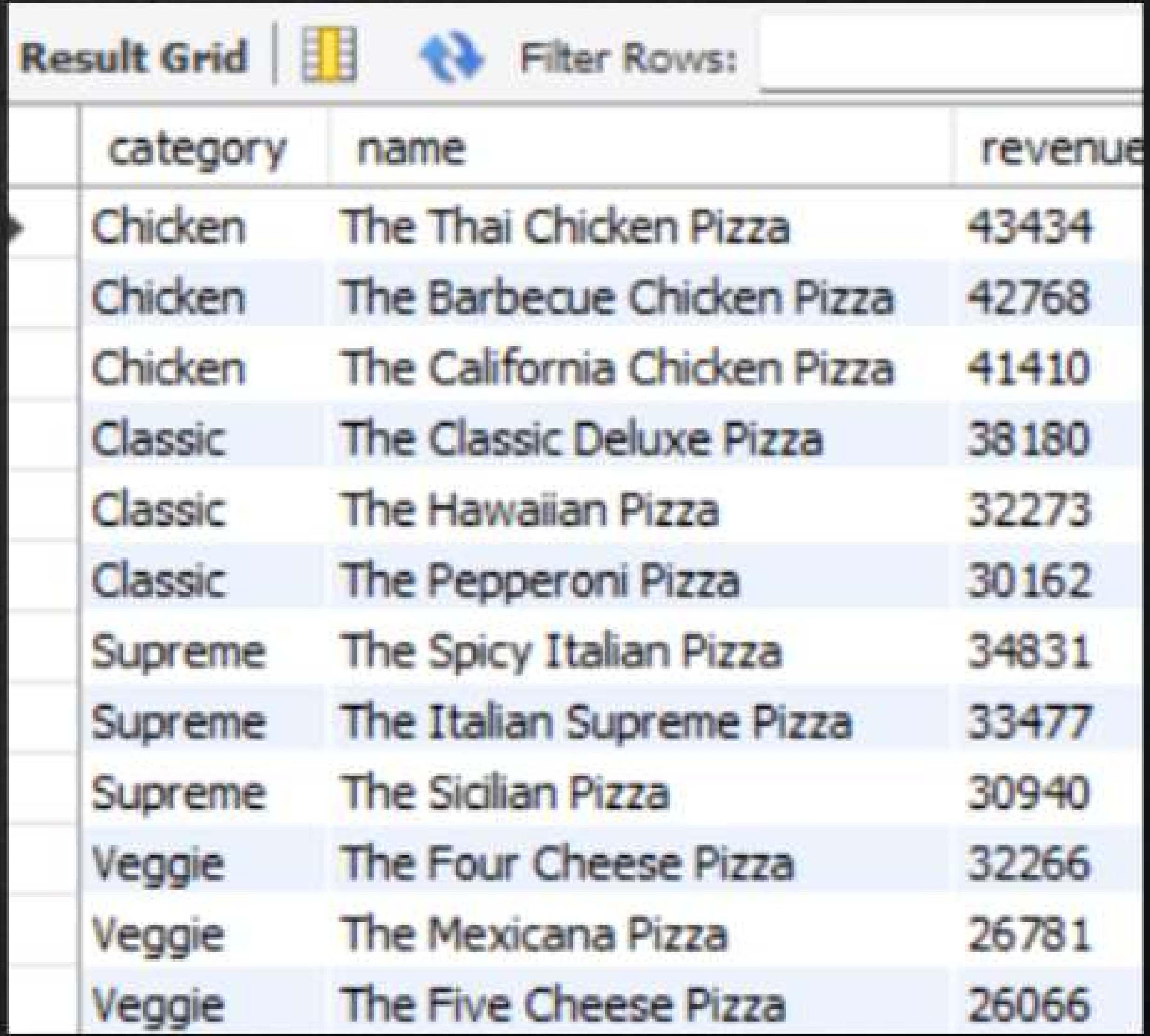
```
order_date   cum_revenue
2015-01-01   2713.8500000000004
2015-01-02   5445.75
2015-01-03   8108.15
2015-01-04   9863.6
2015-01-05   11929.55
2015-01-06   14358.5
2015-01-07   16560.7
```

ADVANCED QUERIES

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



```
select category , name, revenue
from (select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from (SELECT pizza_types.name, pizza_types.category,
    round((SUM(order_details.quantity * pizzas.price))) AS revenue
FROM order_details JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    join
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_i
GROUP BY pizza_types.name, pizza_types.category) as a)as b
where rn <= 3 ;
```



Result Grid | Filter Rows:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41410
Classic	The Classic Deluxe Pizza	38180
Classic	The Hawaiian Pizza	32273
Classic	The Pepperoni Pizza	30162
Supreme	The Spicy Italian Pizza	34831
Supreme	The Italian Supreme Pizza	33477
Supreme	The Sicilian Pizza	30940
Veggie	The Four Cheese Pizza	32266
Veggie	The Mexicana Pizza	26781
Veggie	The Five Cheese Pizza	26066

THANK YOU!