PIZZA SALES ANALYSIS USING SQL

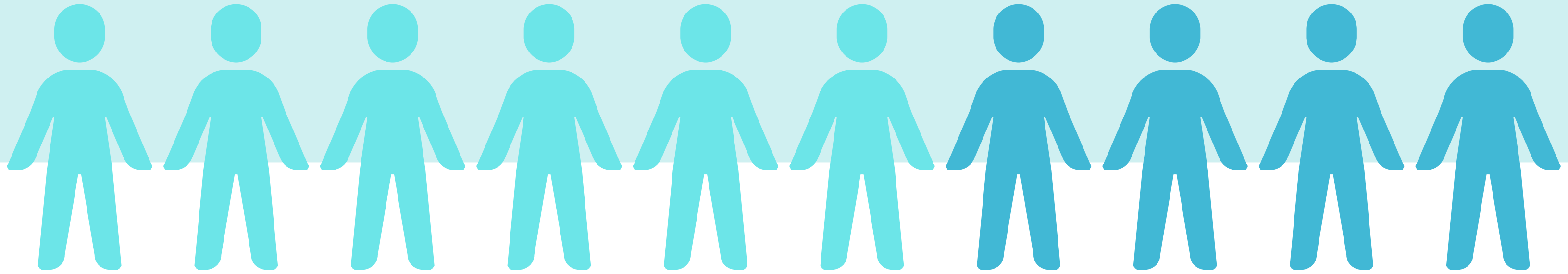
By : Sohail Alam

OBJECTIVE

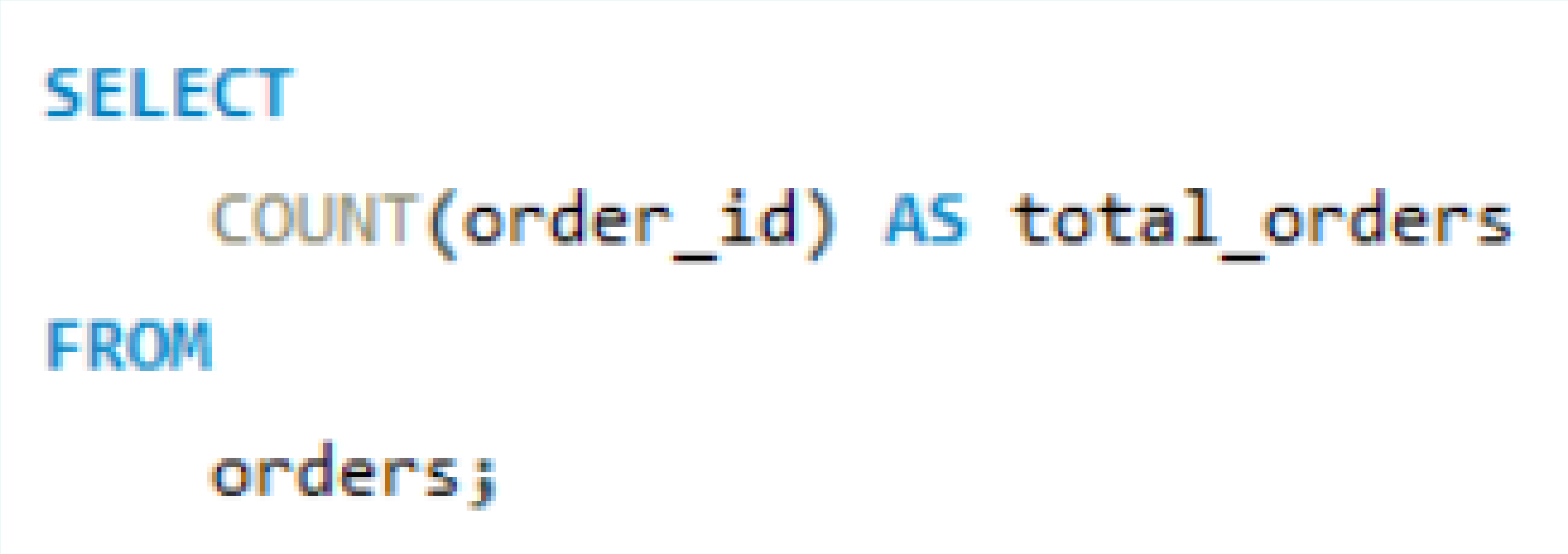
Objective : To analyze pizza sales data using SQL queries.

Dataset : Pizza sales database containing tables like orders, order_details, pizzas, pizza_types.

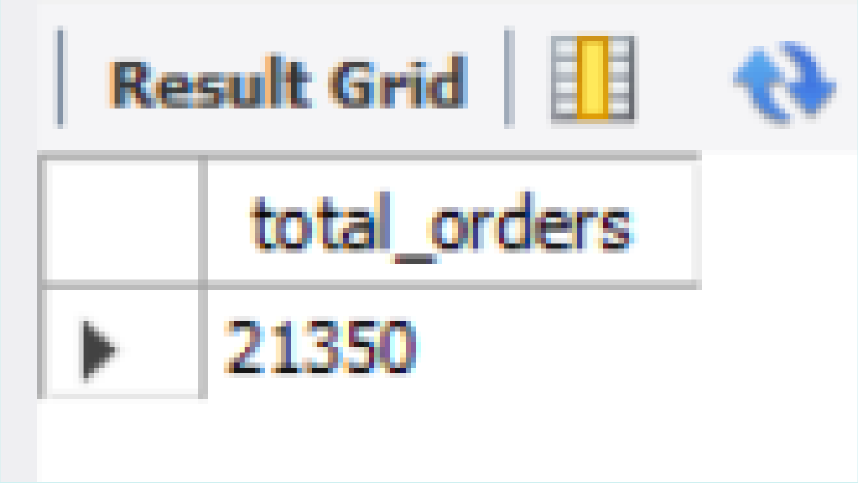
Tools Used : SQL



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

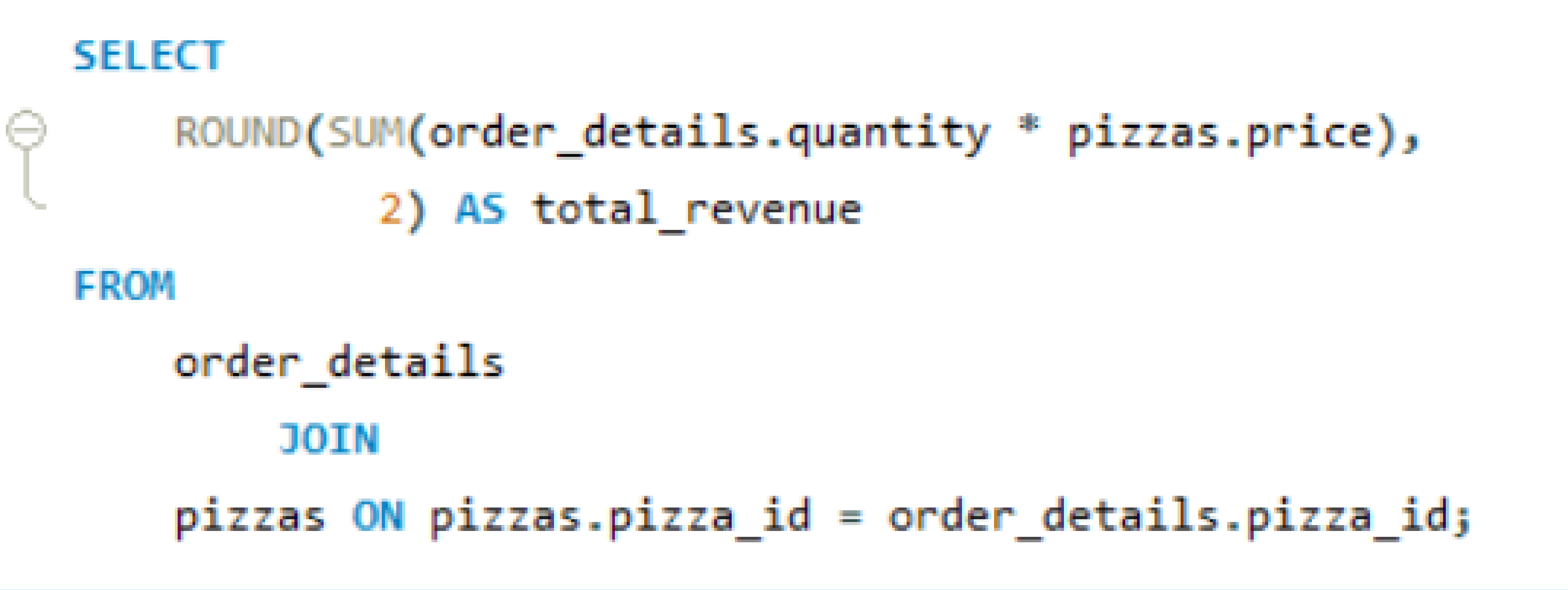


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

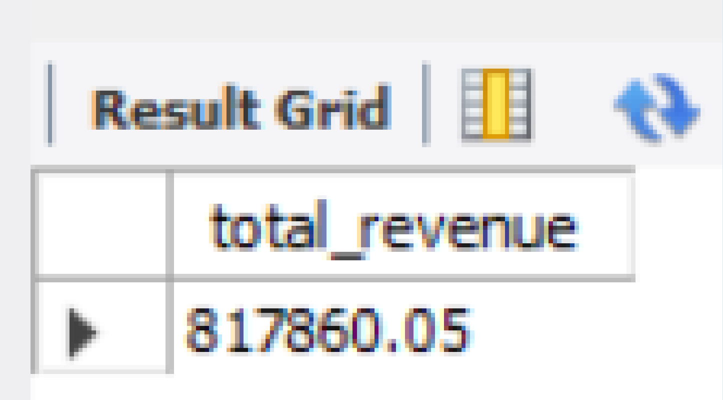


```
total_orders
21350
```

CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES

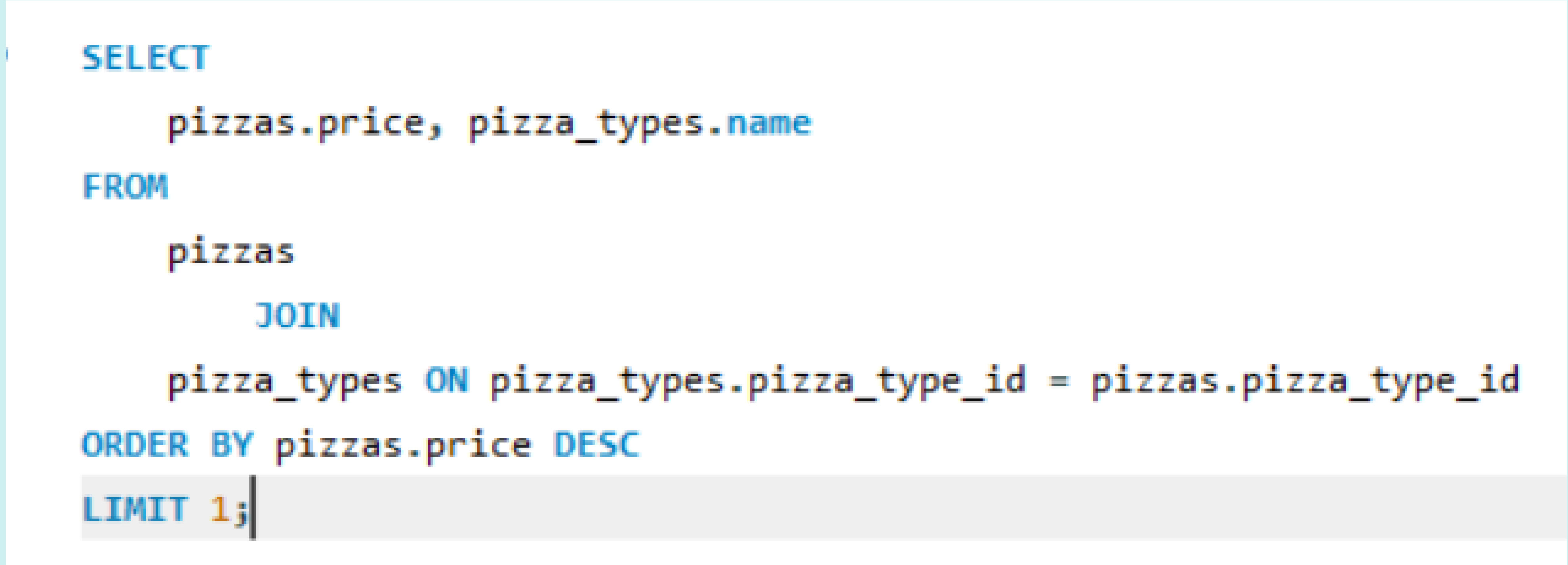


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

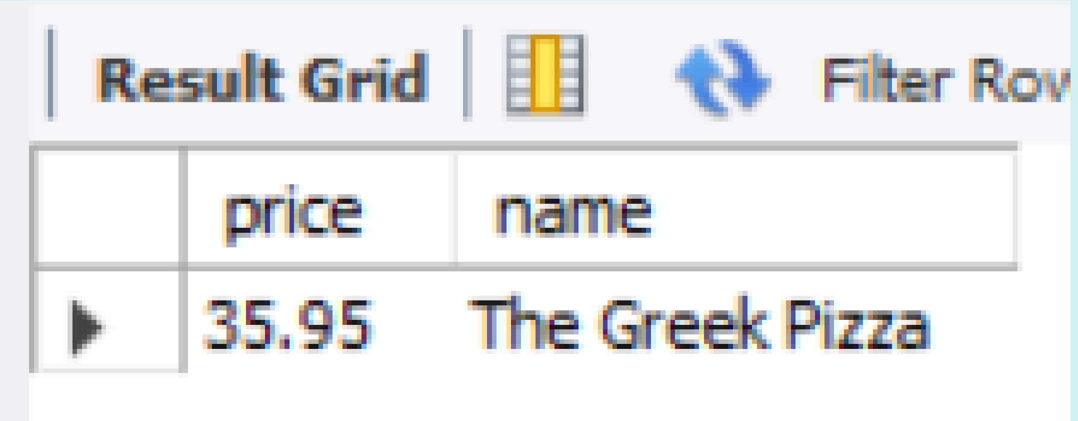


```
total_revenue
817860.05
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

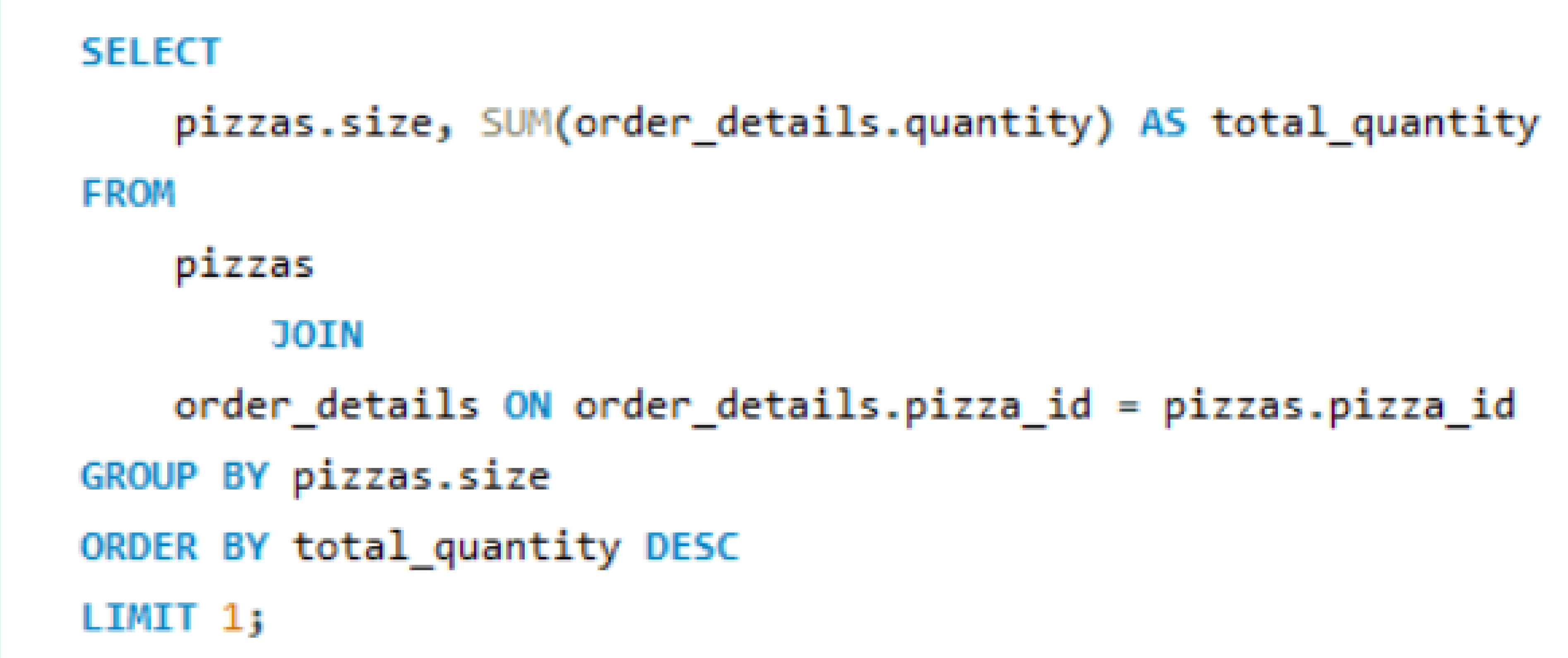


```
SELECT
    pizzas.price, pizza_types.name
FROM
    pizzas
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

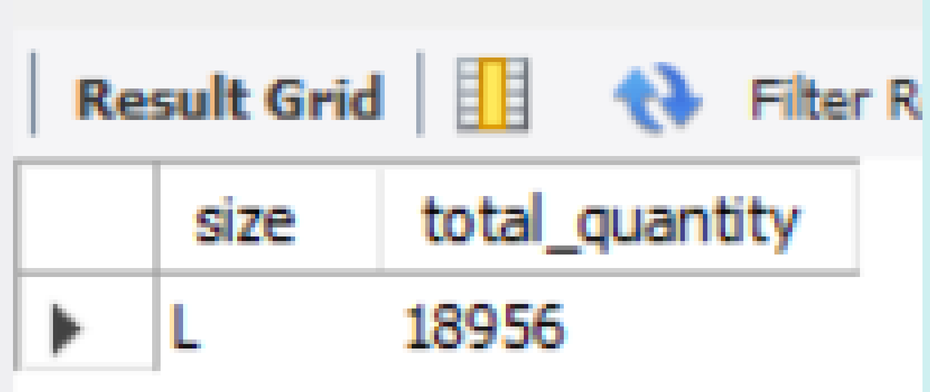


	price	name
▶	35.95	The Greek Pizza

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

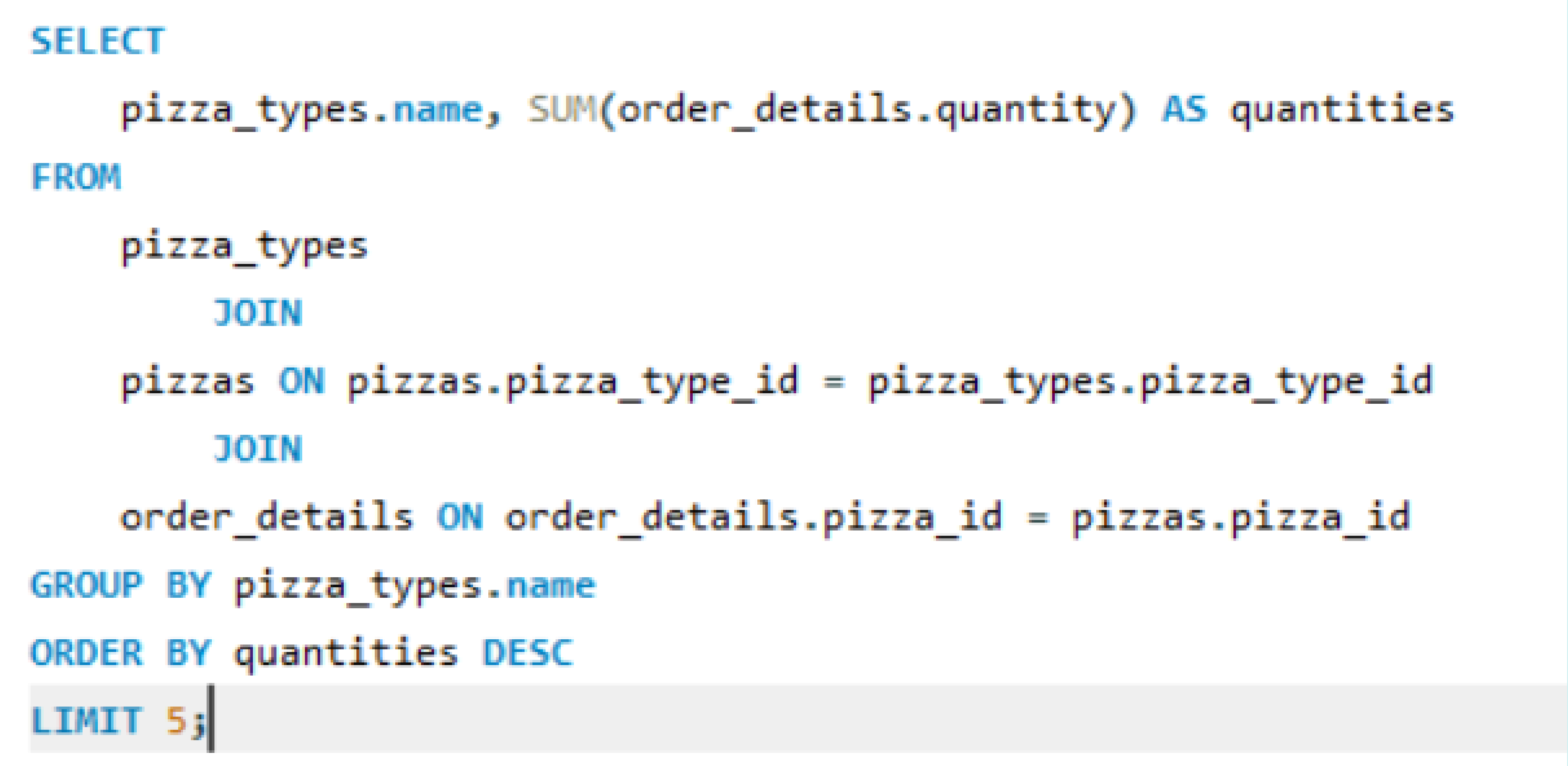


```
SELECT
    pizzas.size, SUM(order_details.quantity) AS total_quantity
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY total_quantity DESC
LIMIT 1;
```

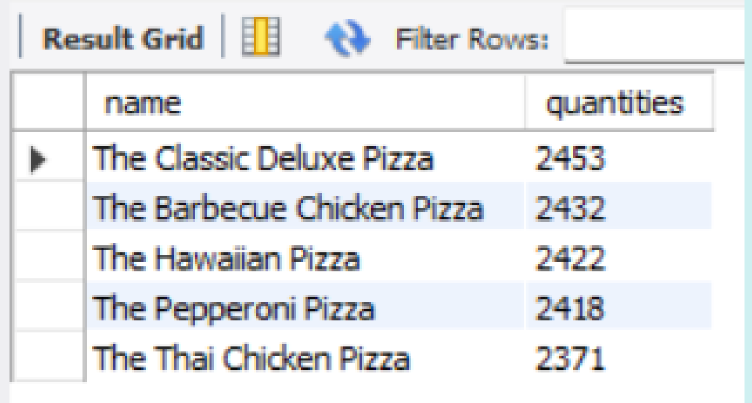


```
   size   total_quantity
▶  L      18956
```

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

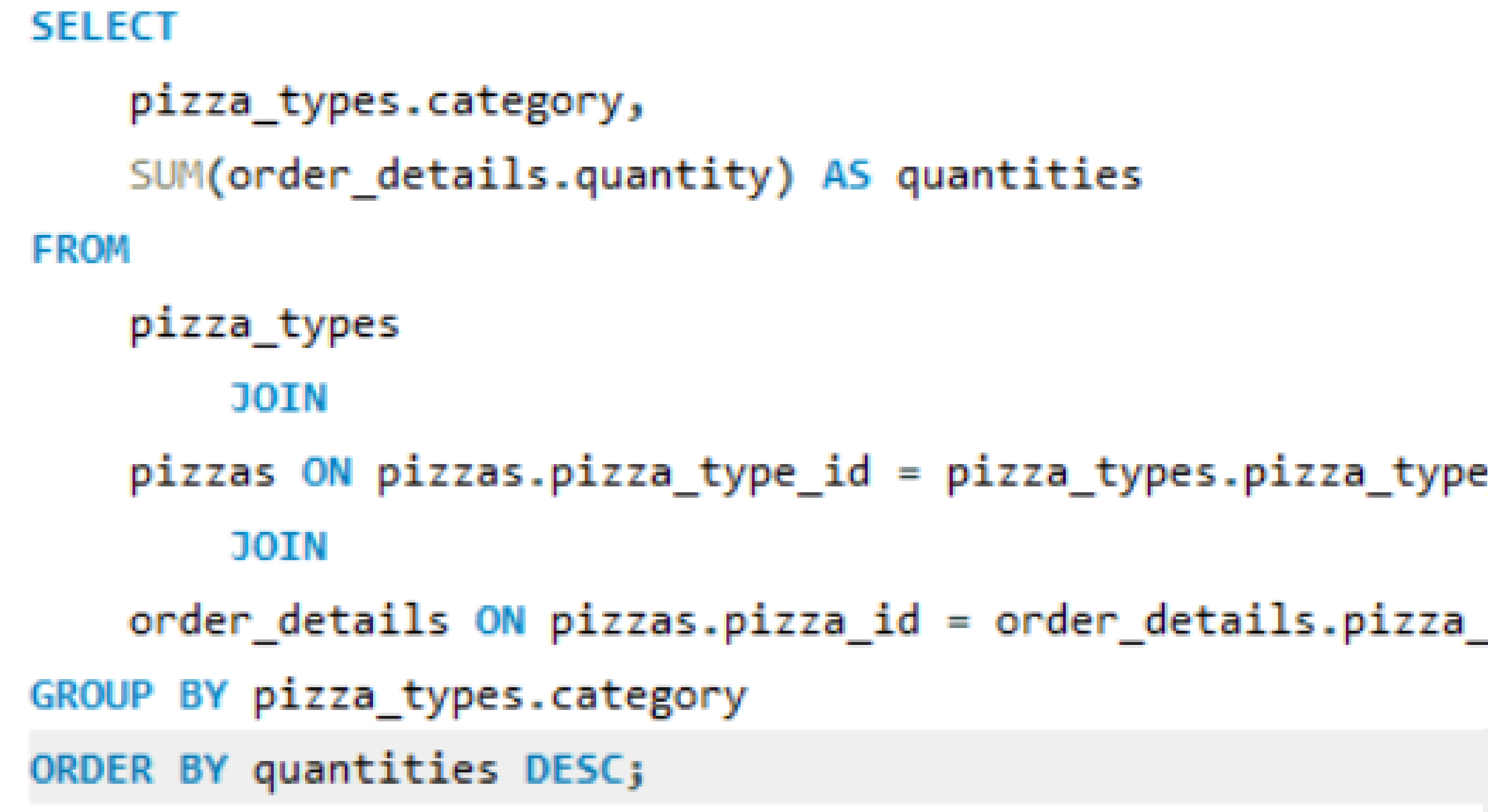


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantities DESC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

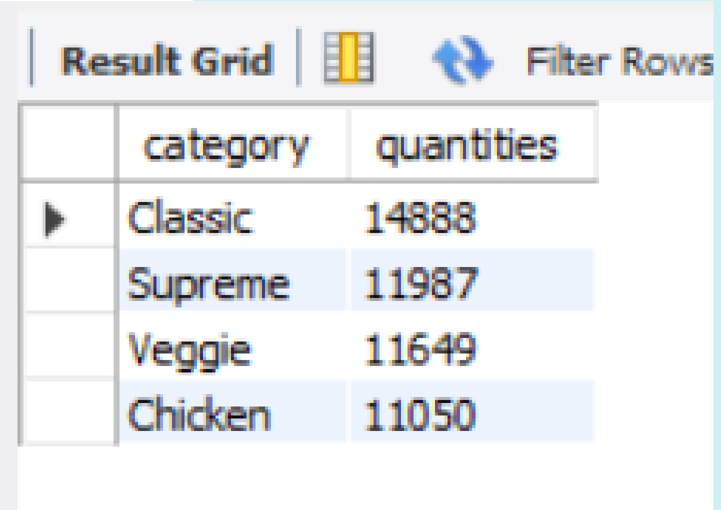


name	quantities
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

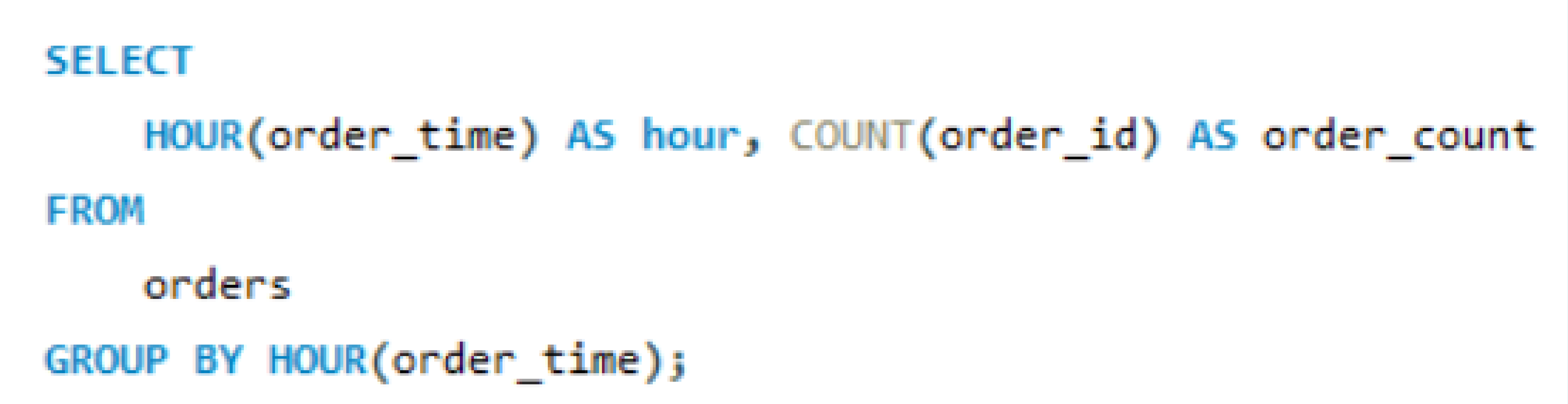


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantities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

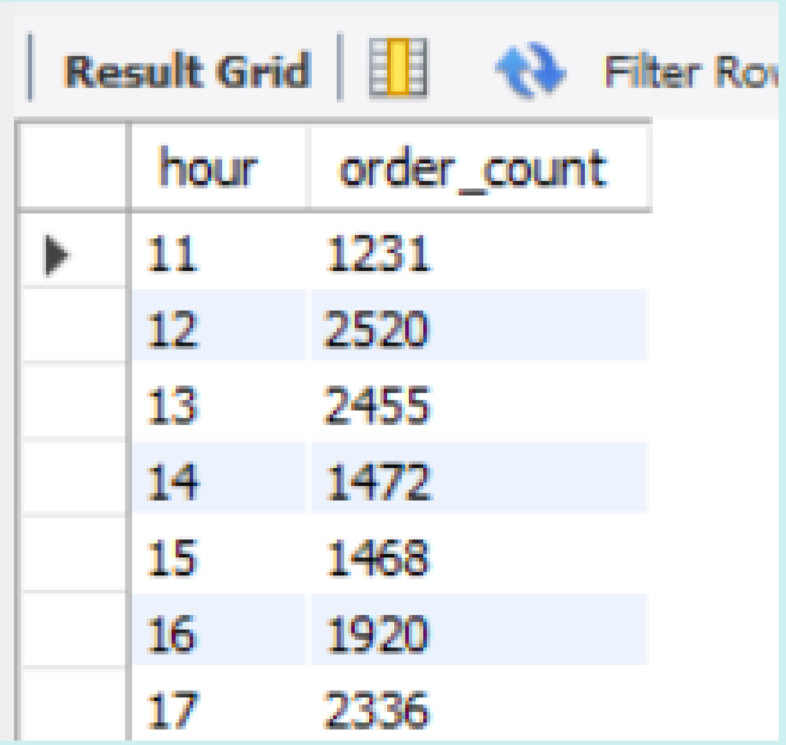


category	quantities
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

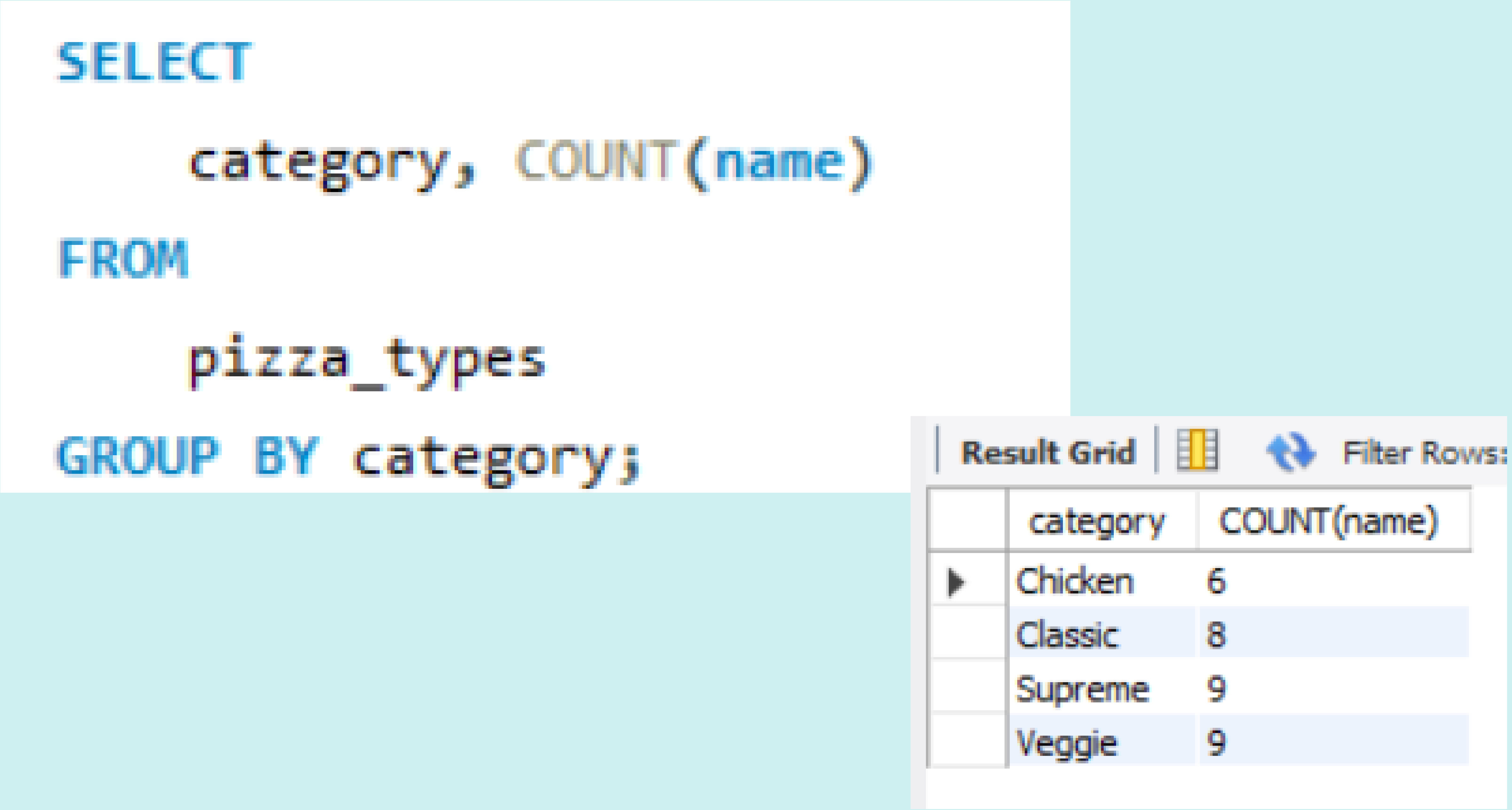


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```



hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

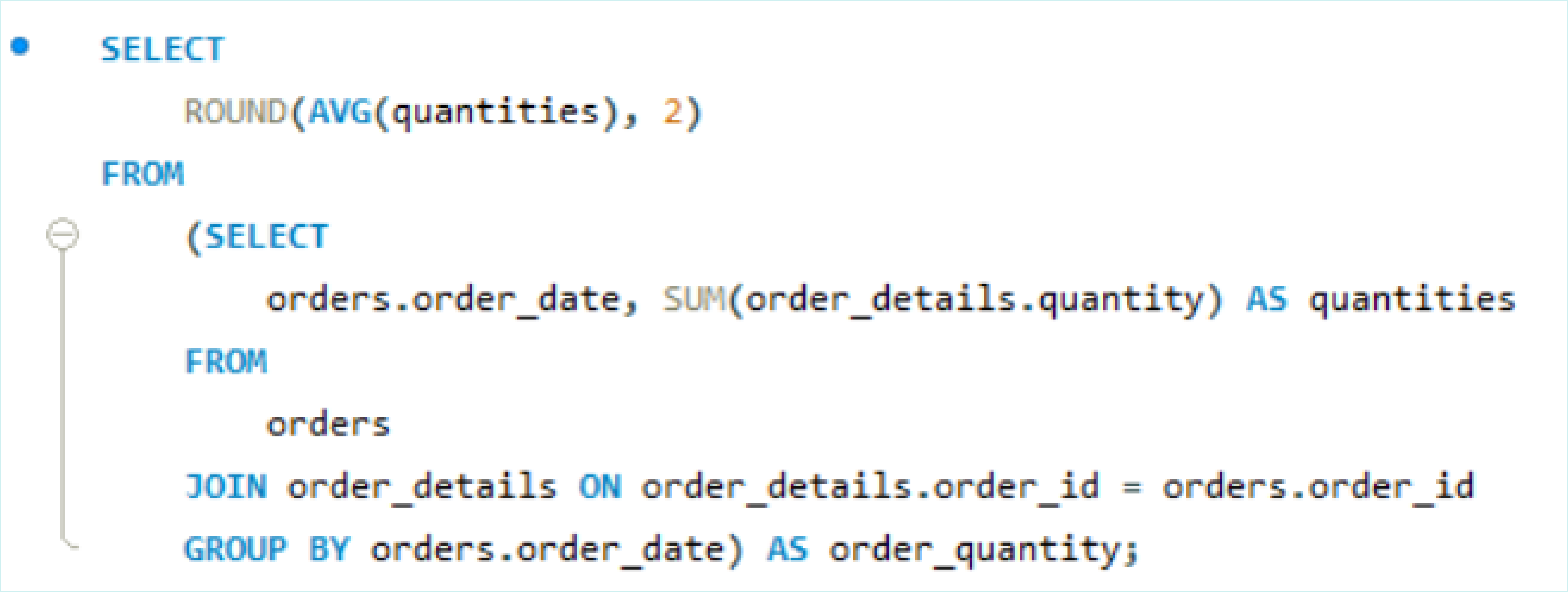
JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



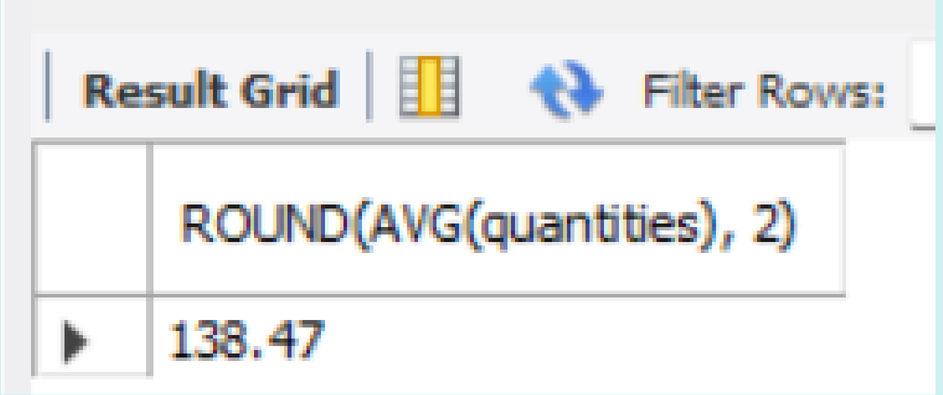
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

```
  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.

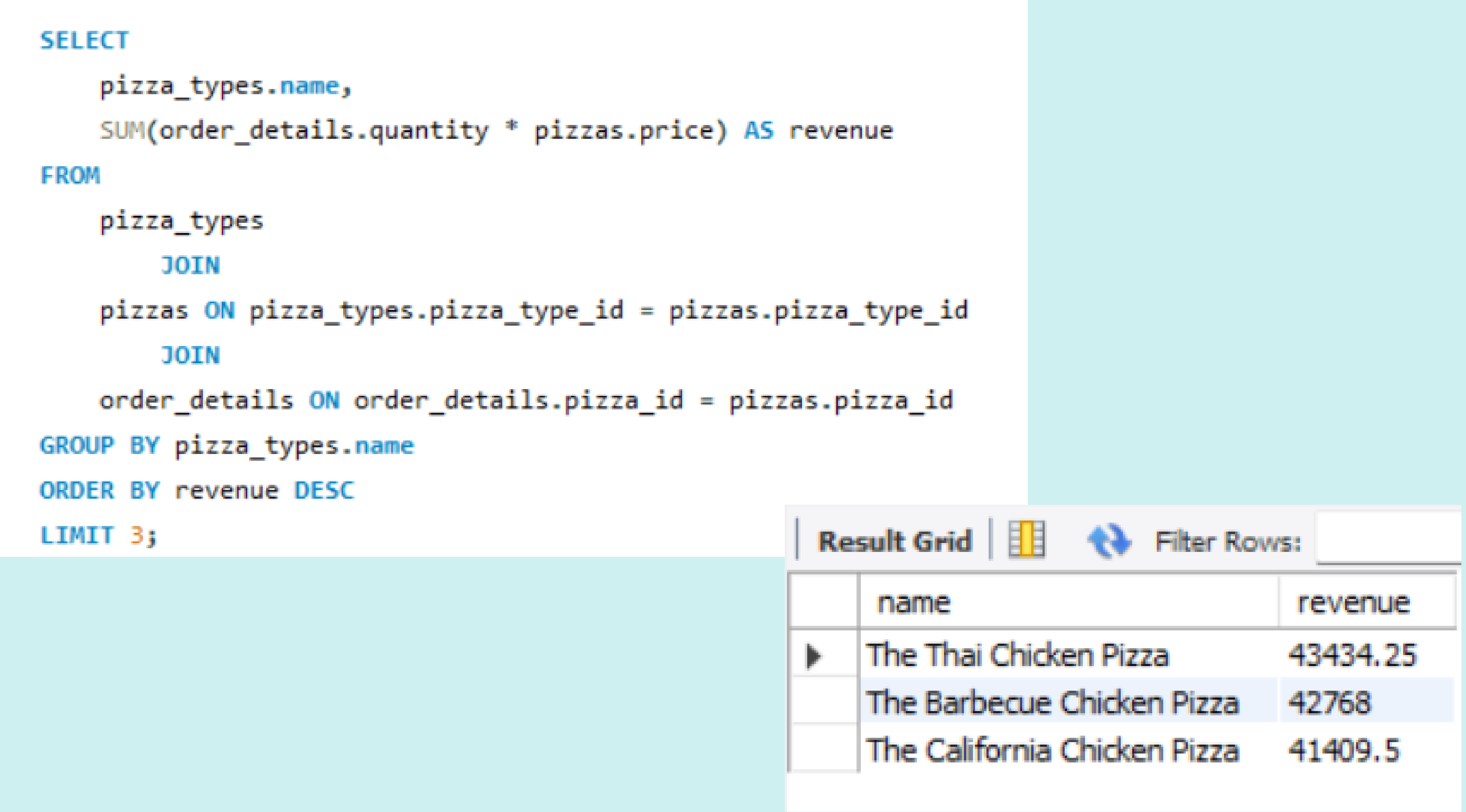


```
SELECT
    ROUND(AVG(quantities), 2)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantities
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



```
ROUND(AVG(quantities), 2)
138.47
```

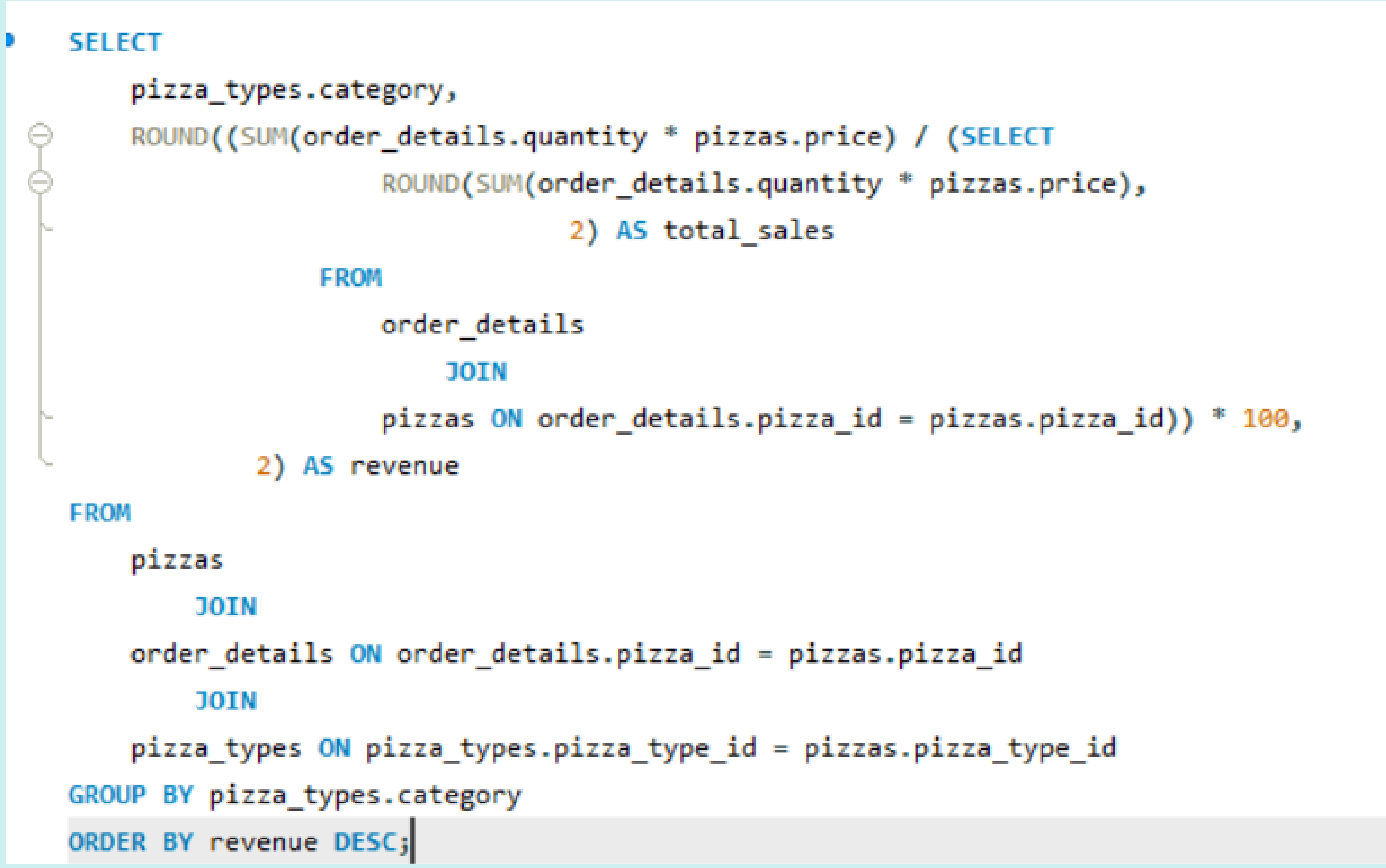
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



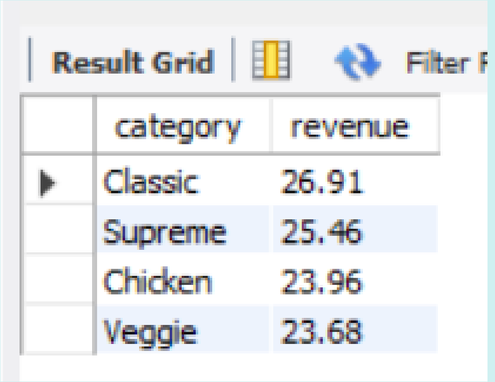
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

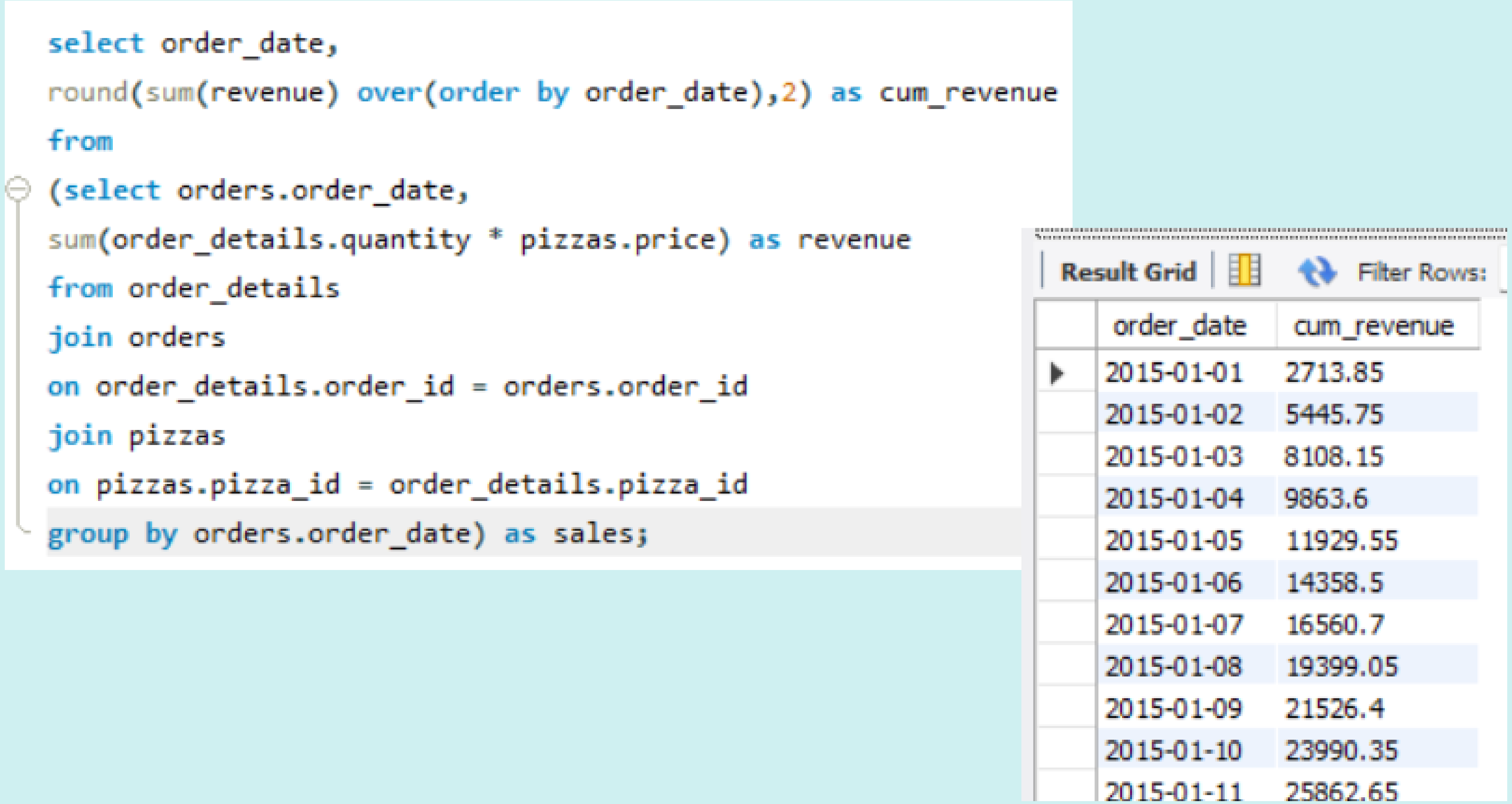


```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN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



```
  category   revenue
▶ Classic    26.91
  Supreme    25.46
  Chicken    23.96
  Veggie     23.68
```

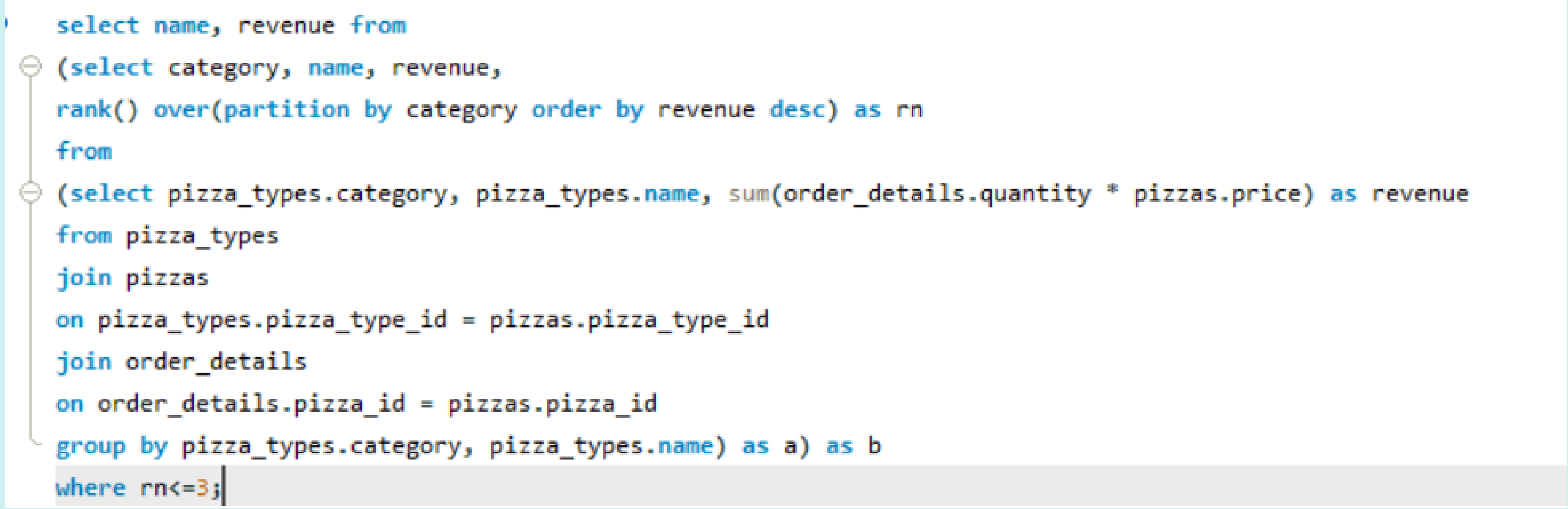
ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



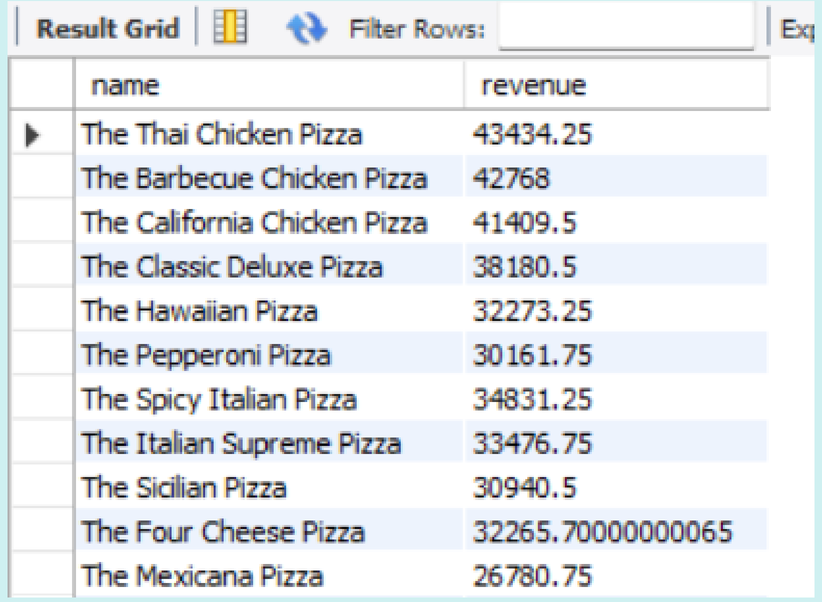
```sql
select order_date,
round(sum(revenue) over(order by order_date),2) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details
join orders
on order_details.order_id = orders.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.order_date) as sales;
```

```
order_date    cum_revenue
2015-01-01    2713.85
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    23990.35
2015-01-11    25862.65
```

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 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;
```



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

CONCLUSION

I started with basic queries to understand orders, revenue, and popular pizzas. Through intermediate queries, we explored order patterns by time, category distributions, and daily averages. Finally, with advanced analysis, we calculated revenue contribution, cumulative sales trends, and top performers within categories.