HOT & SPICY

PIZZA

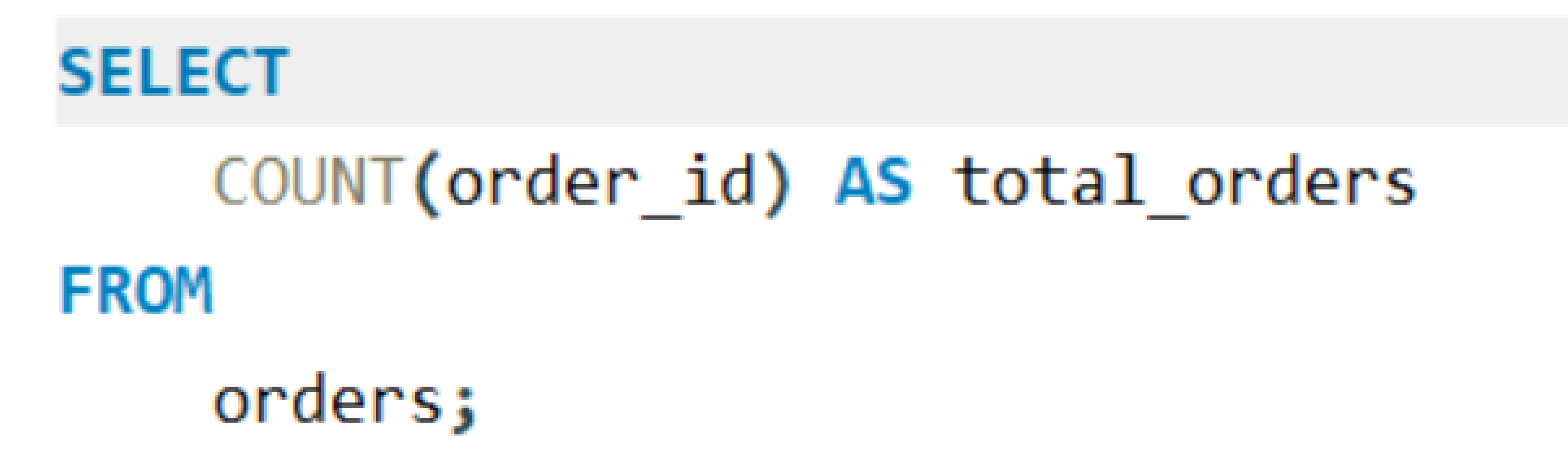
50% OFF

ORDER NOW!

GANPATI MISHRA

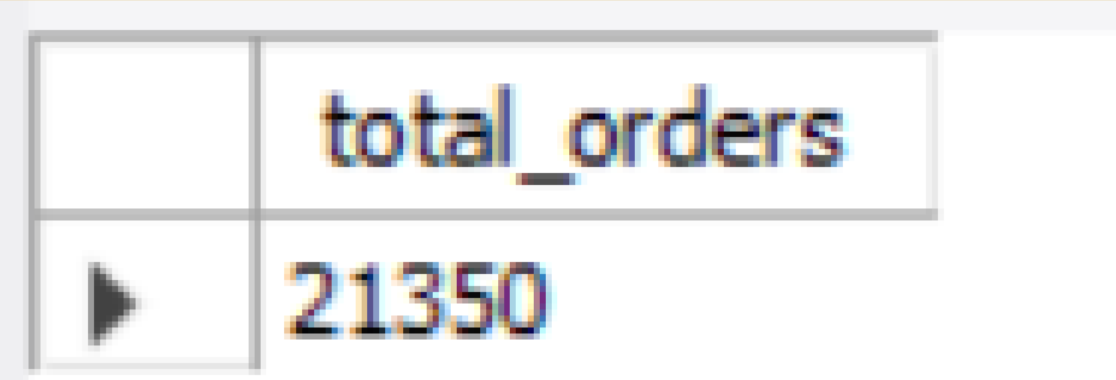
This SQL project analyzes pizza sales data to uncover trends, improve inventory management, and enhance customer satisfaction. Key metrics include sales volume, peak times, and popular toppings.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



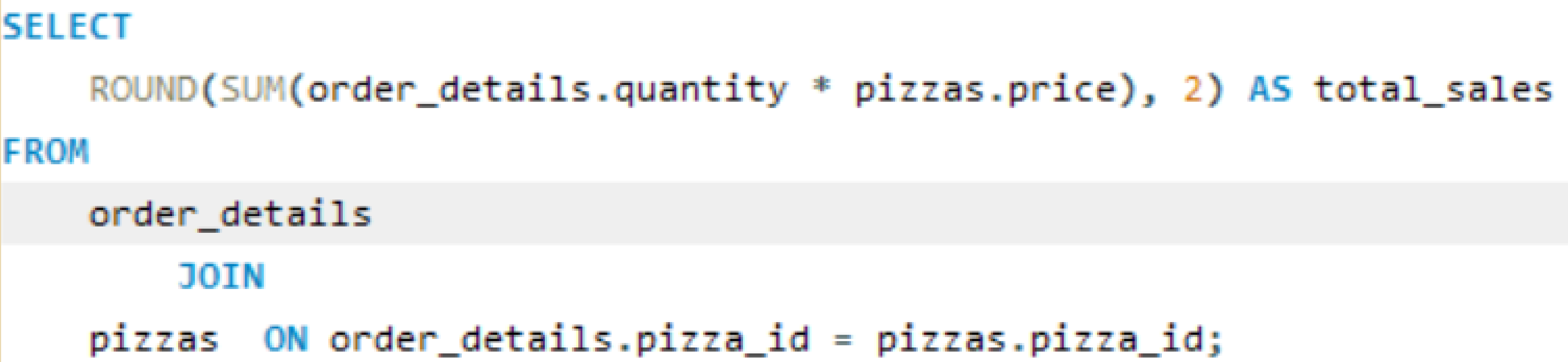
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

OUTPUT:



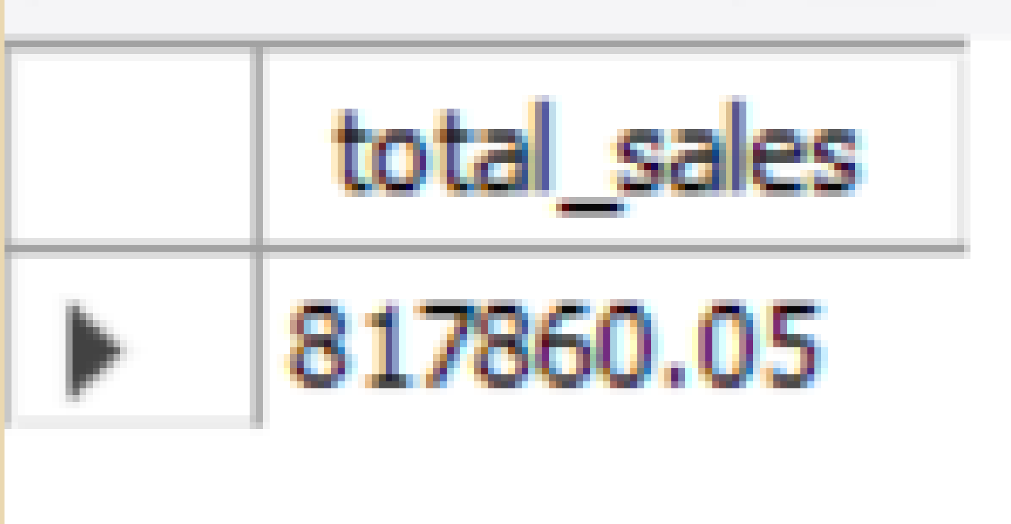
	total_orders
▶	21350

2 .CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



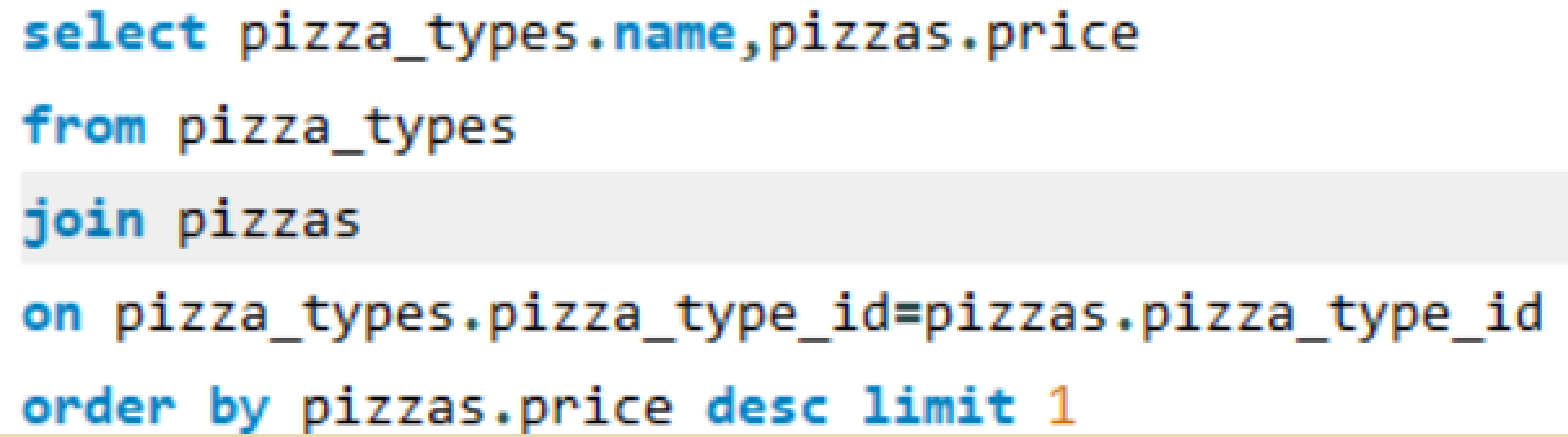
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
FROM
    order_details
        JOIN
    pizzas  ON order_details.pizza_id = pizzas.pizza_id;
```

OUTPUT:



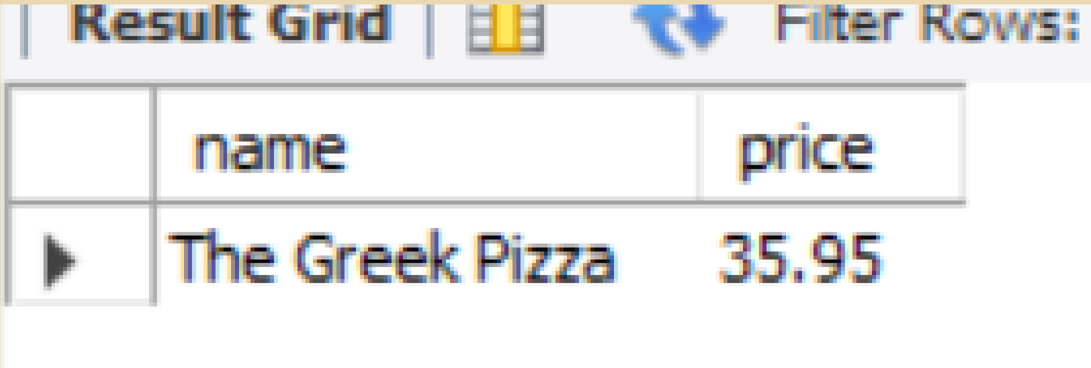
	total_sales
▶	817860.05

3.IDENTIFY THE HIGHEST-PRICED PIZZA.



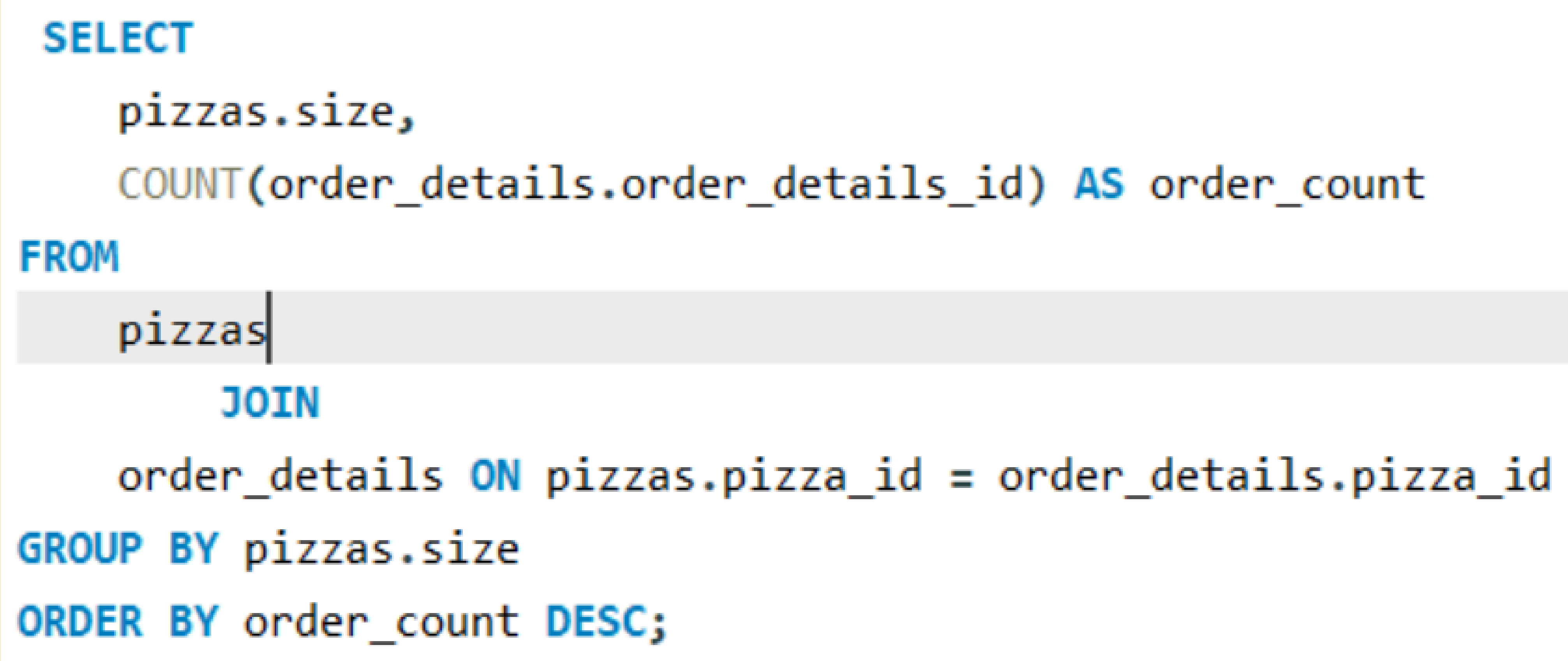
```
select pizza_types.name,pizzas.price
from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
order by pizzas.price desc limit 1
```

OUTPUT:



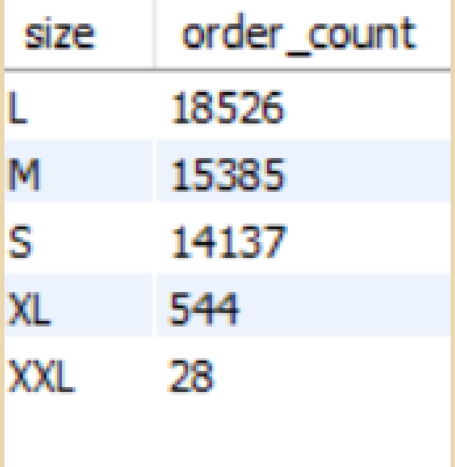
```
  name             price
▶ The Greek Pizza  35.95
```

4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



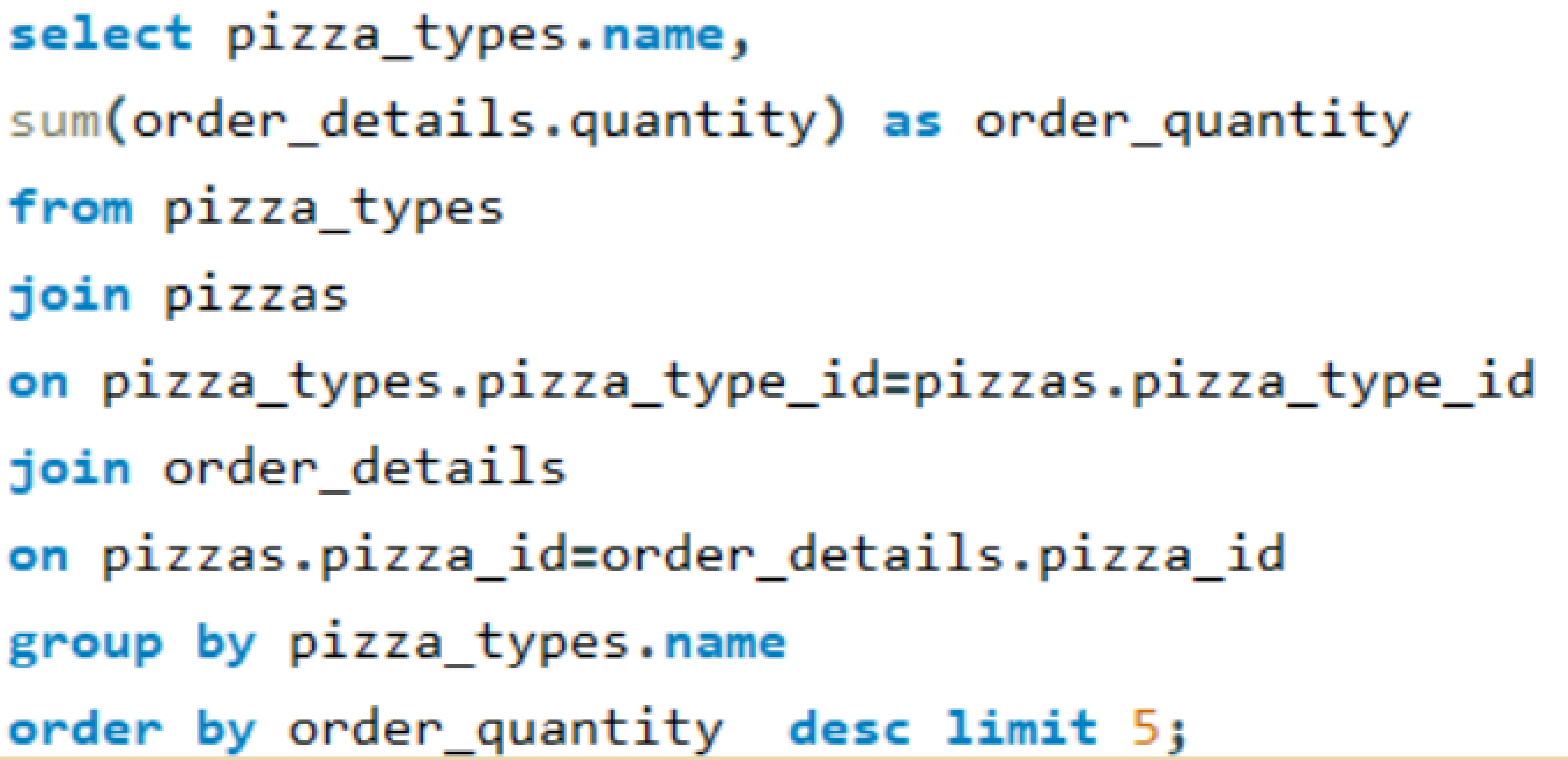
```
 SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

OUTPUT:



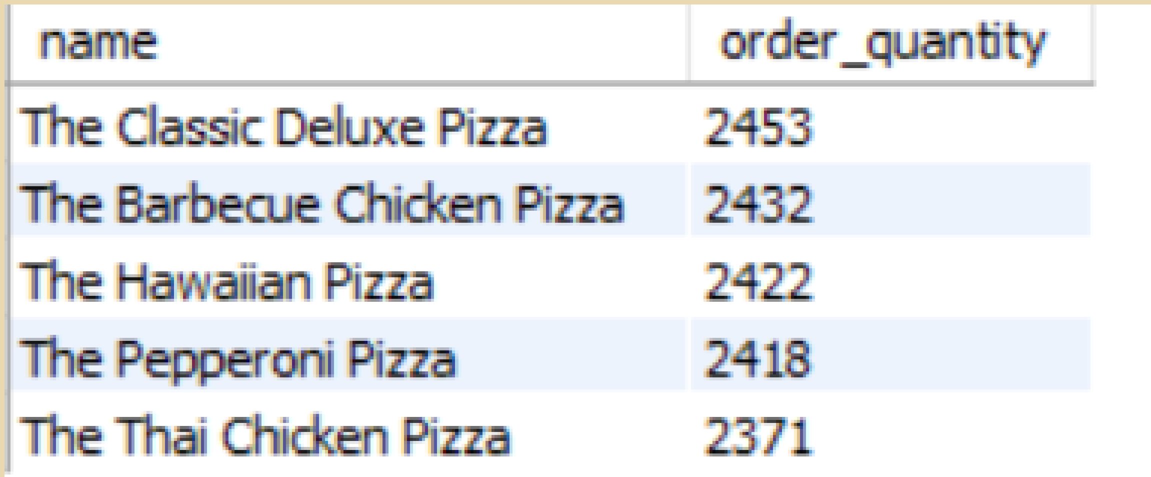
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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



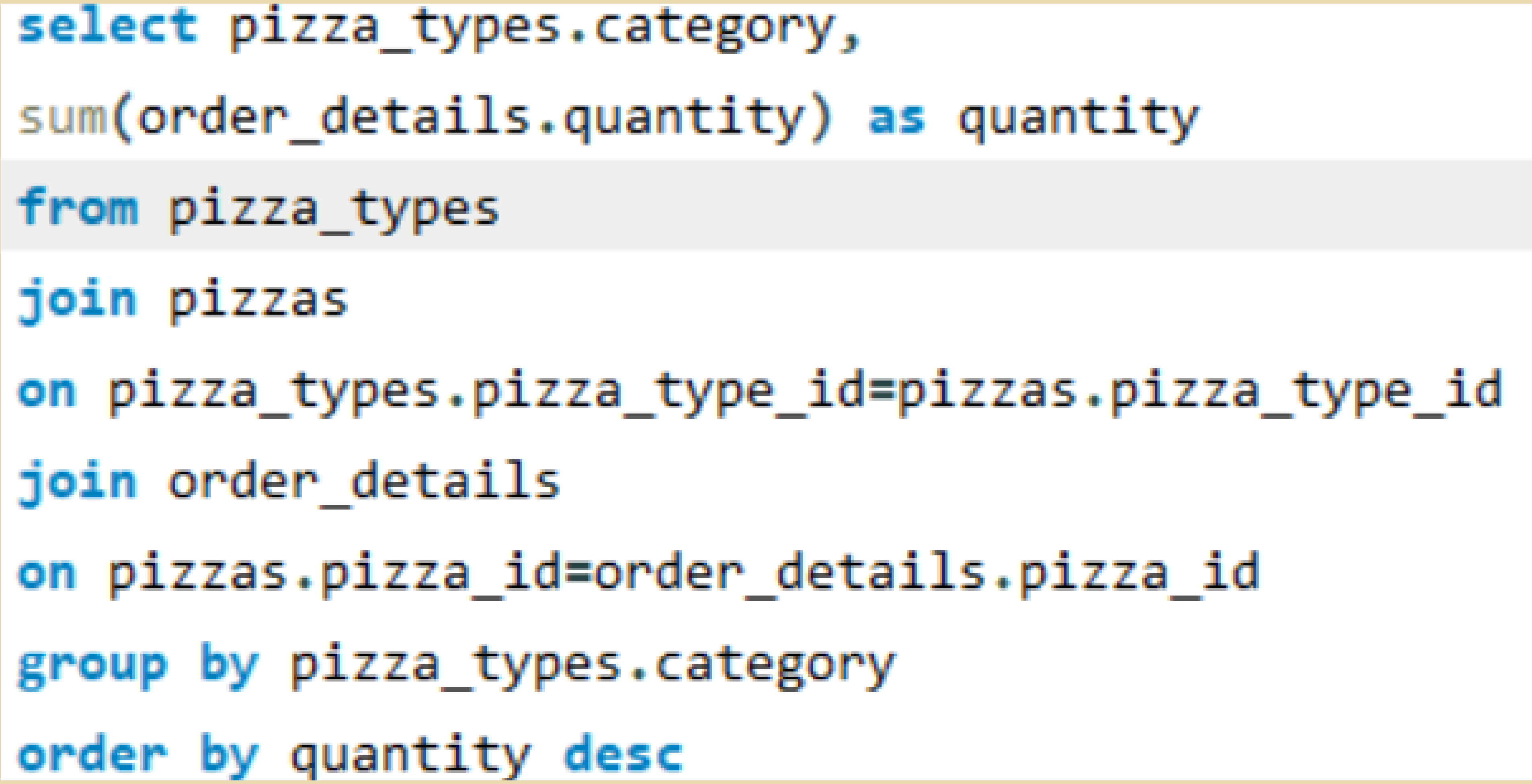
```
select pizza_types.name,
sum(order_details.quantity) as order_quantity
from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name
order by order_quantity  desc limit 5;
```

OUTPUT:



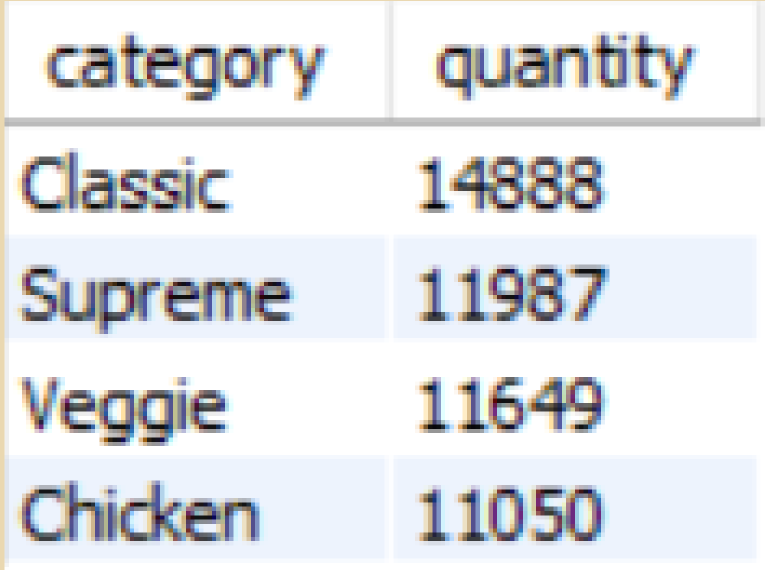
name	order_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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



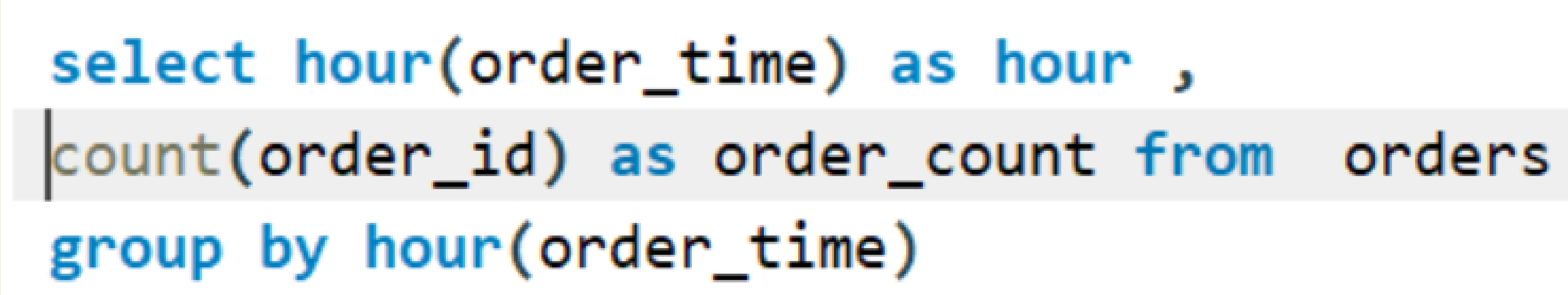
```sql
select pizza_types.category,
sum(order_details.quantity) as quantity
from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.category
order by quantity desc
```

OUTPUT:



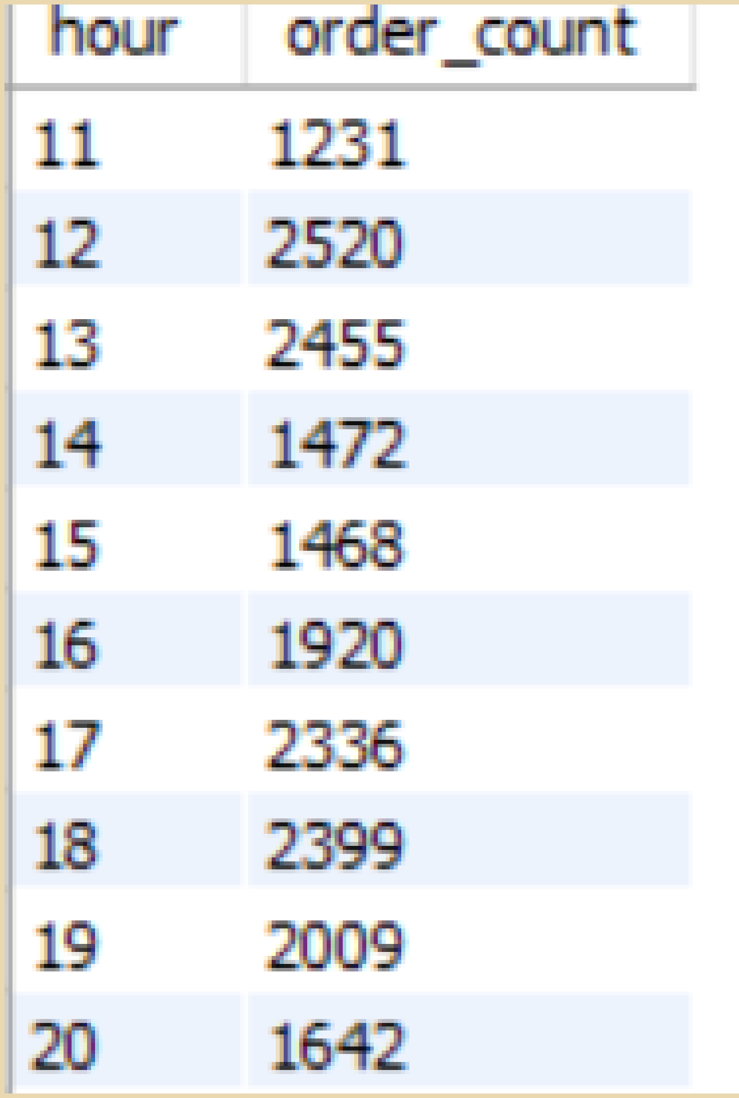
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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



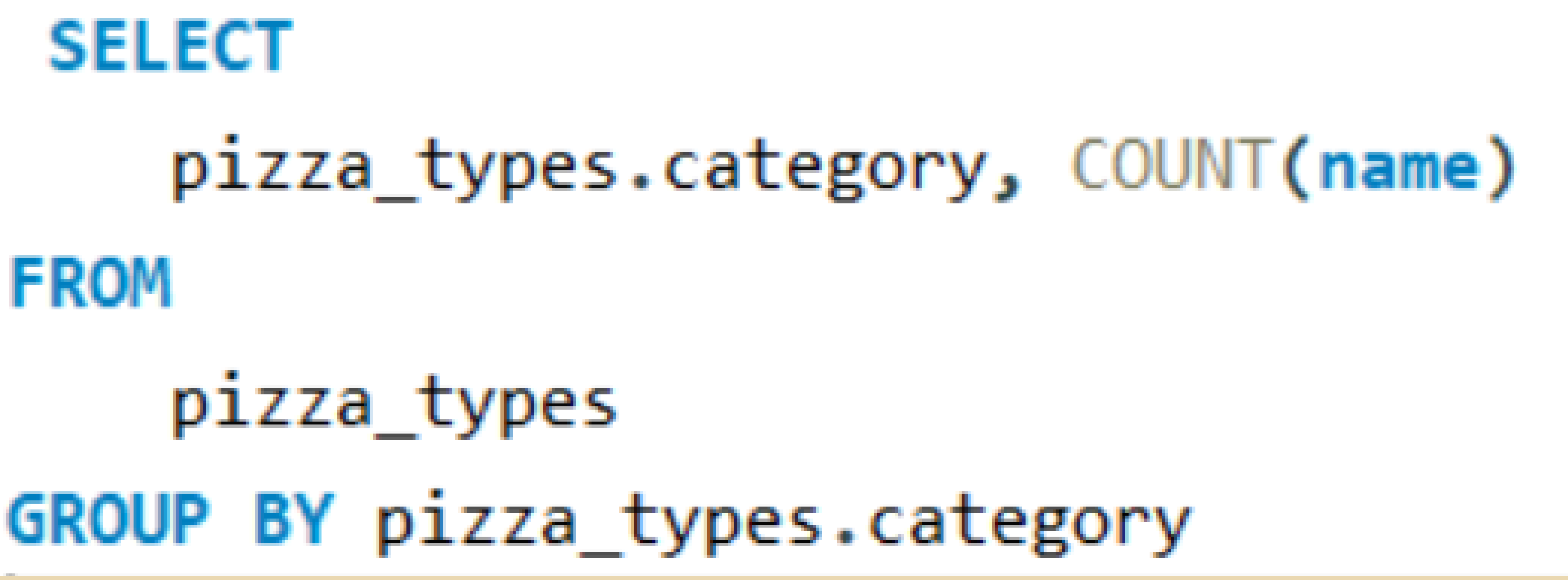
```
select hour(order_time) as hour ,
count(order_id) as order_count from  orders
group by hour(order_time)
```

OUTPUT:



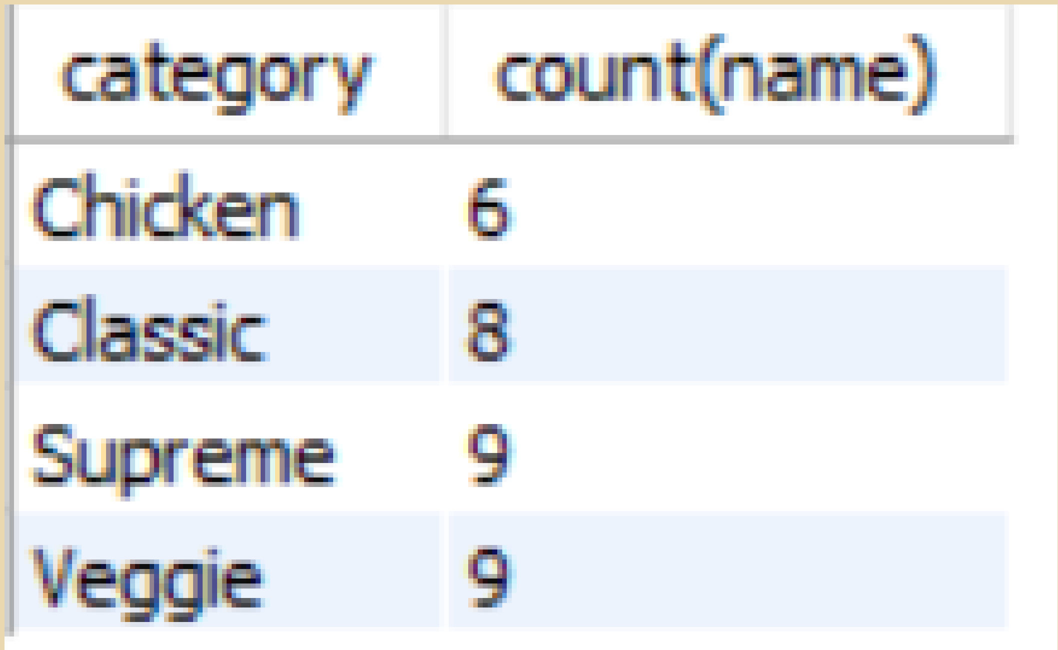
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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



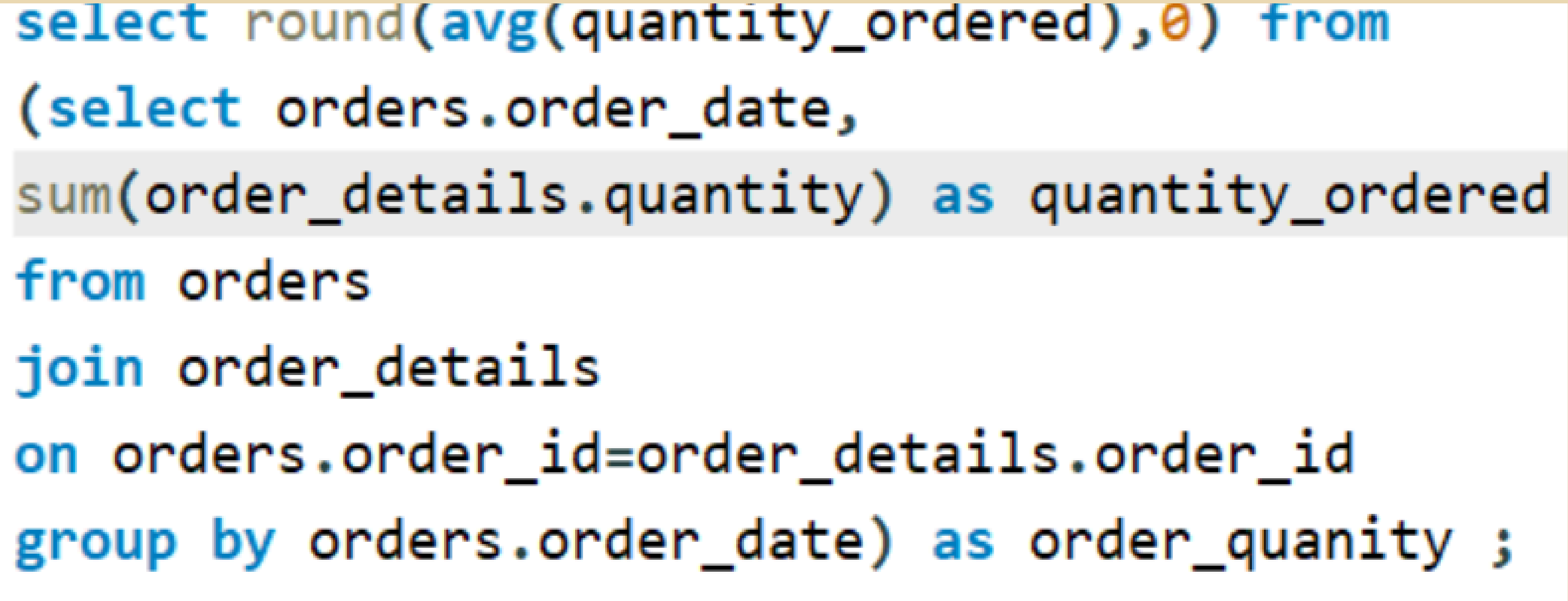
```
 SELECT
    pizza_types.category, COUNT(name)
FROM
    pizza_types
GROUP BY pizza_types.category
```

OUTPUT:



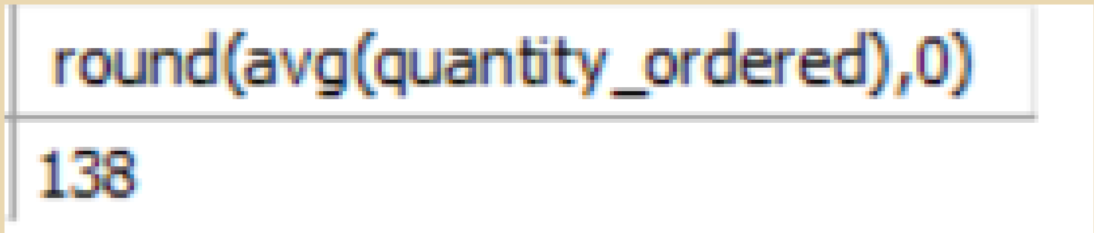
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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



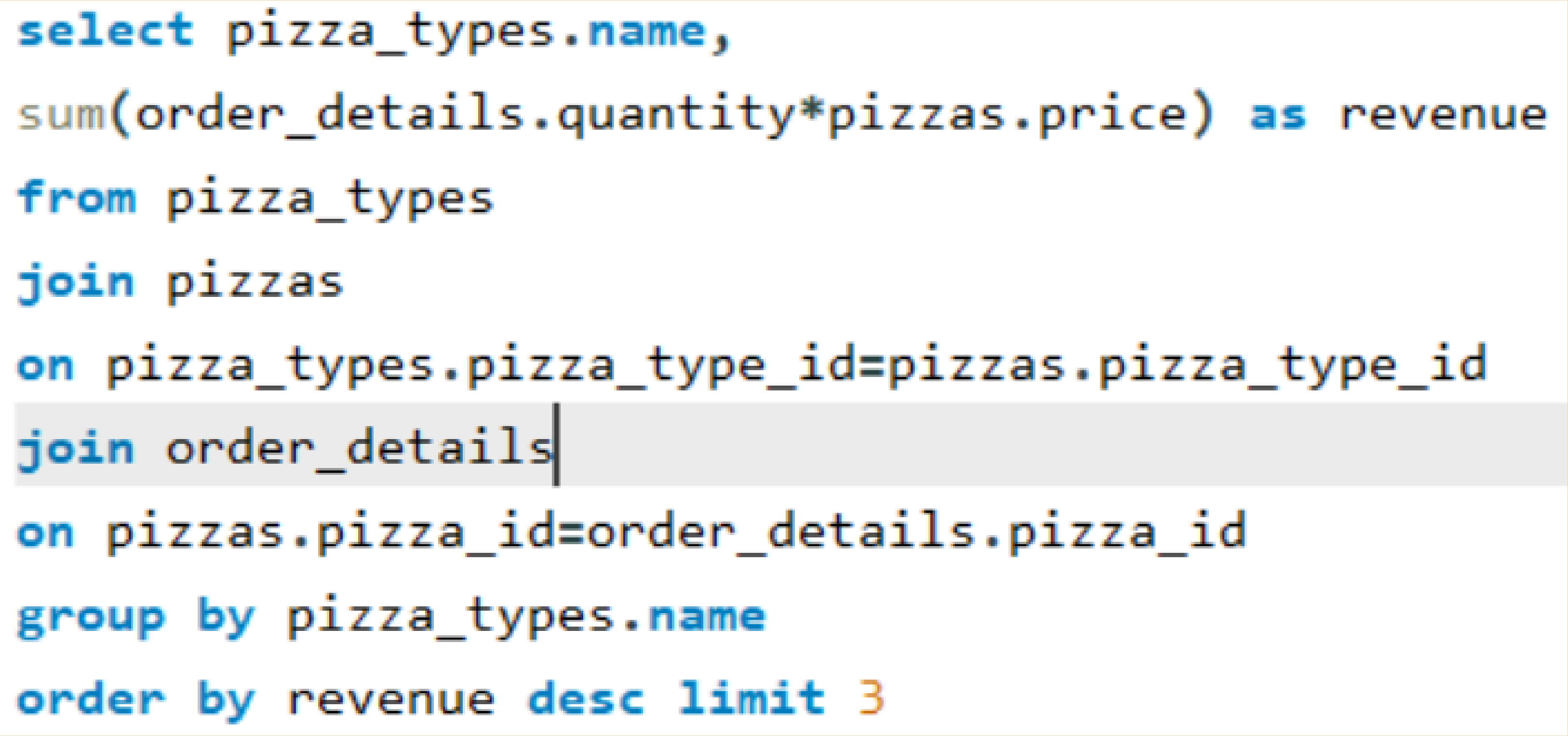
```
select round(avg(quantity_ordered),0) from
(select orders.order_date,
sum(order_details.quantity) as quantity_ordered
from orders
join order_details
on orders.order_id=order_details.order_id
group by orders.order_date) as order_quanity ;
```

OUTPUT:



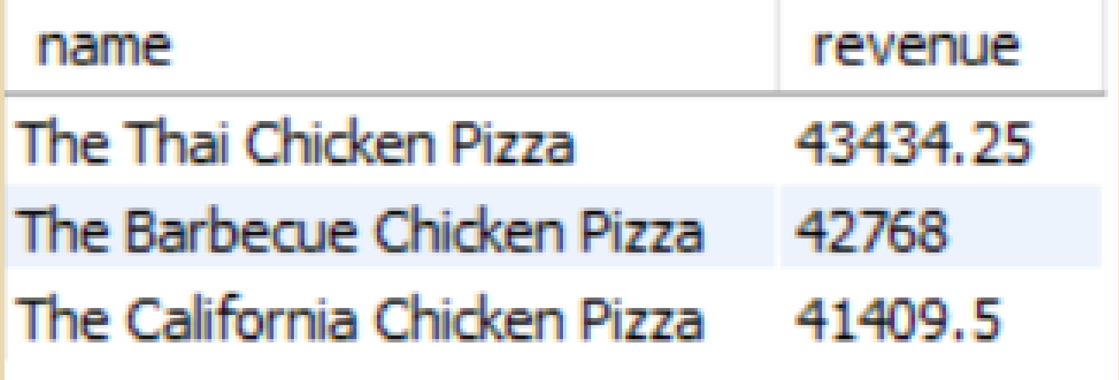
round(avg(quantity_ordered),0)
138

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



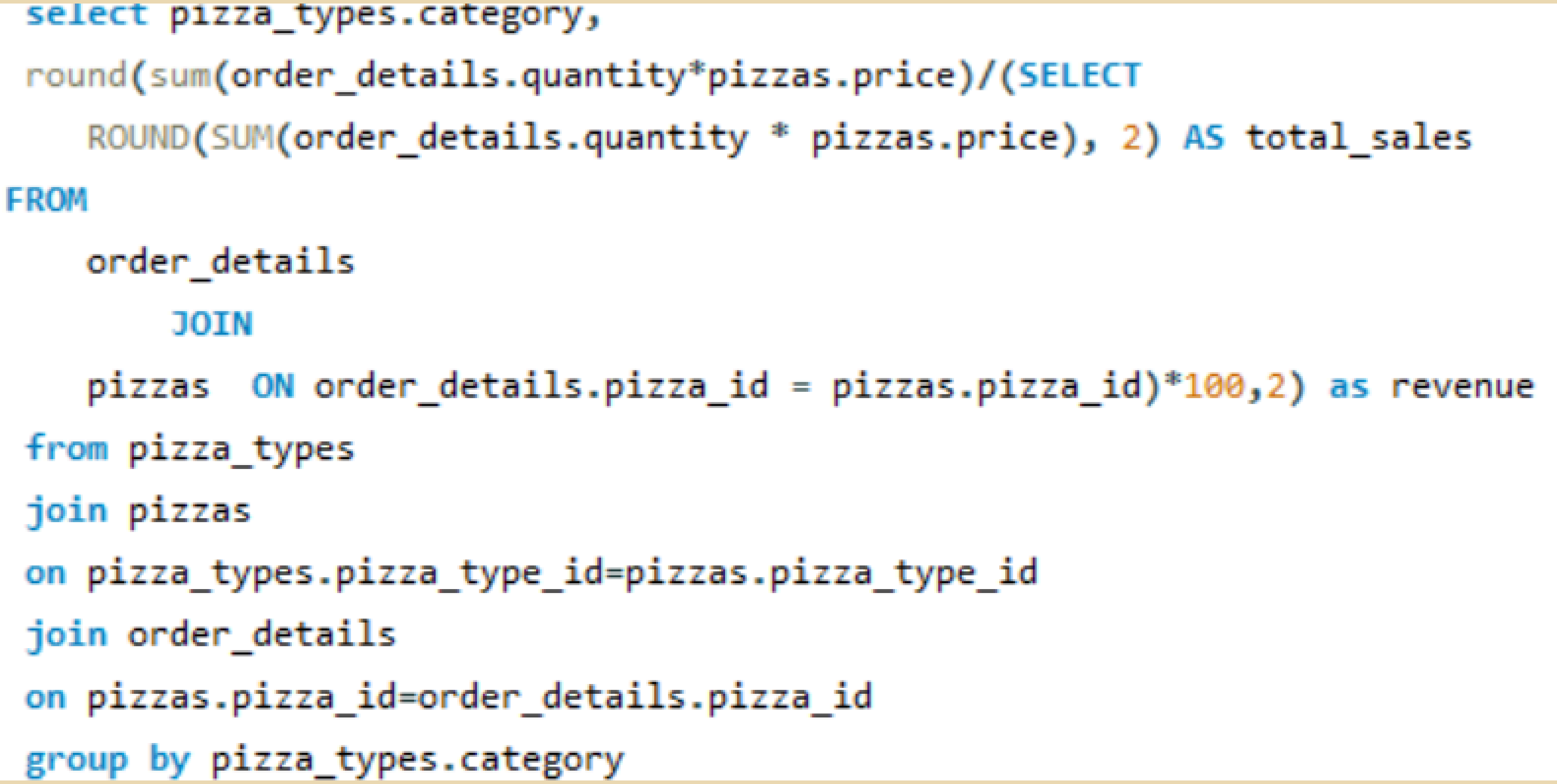
```
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name
order by revenue desc limit 3
```

OUTPUT:



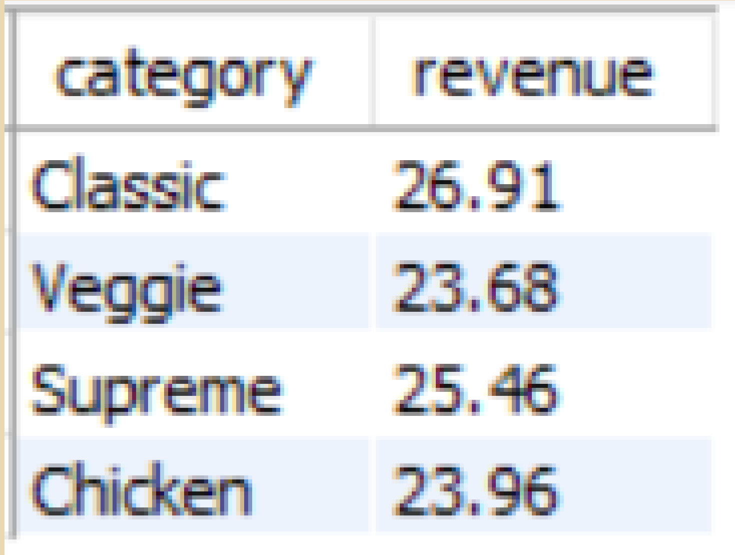
```
name                           revenue
The Thai Chicken Pizza         43434.25
The Barbecue Chicken Pizza     42768
The California Chicken Pizza   41409.5
```

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



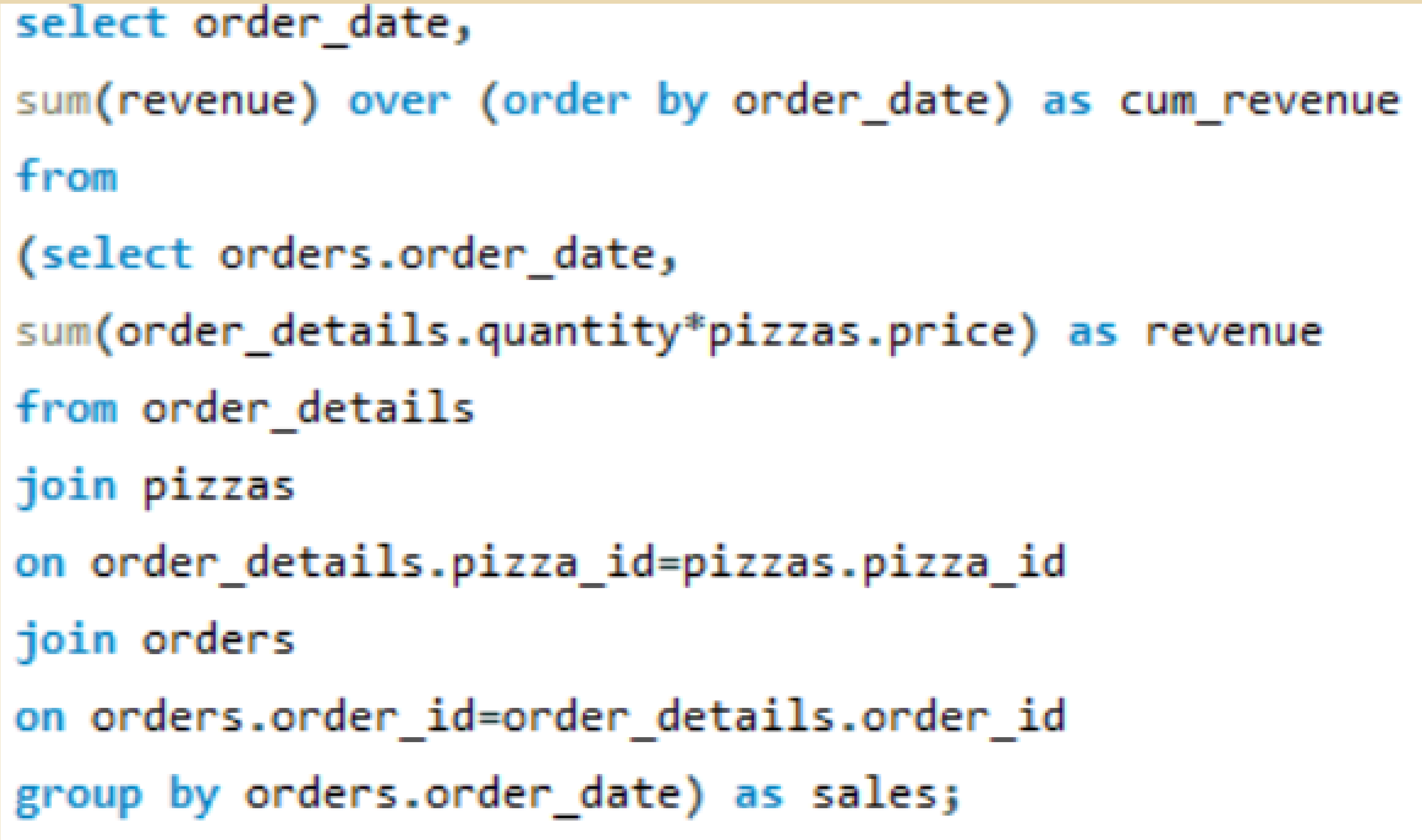
```
 select pizza_types.category,
 round(sum(order_details.quantity*pizzas.price)/(SELECT
    ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
FROM
    order_details
        JOIN
    pizzas  ON order_details.pizza_id = pizzas.pizza_id)*100,2) as revenue
 from pizza_types
 join pizzas
 on pizza_types.pizza_type_id=pizzas.pizza_type_id
 join order_details
 on pizzas.pizza_id=order_details.pizza_id
 group by pizza_types.category
```

OUTPUT:



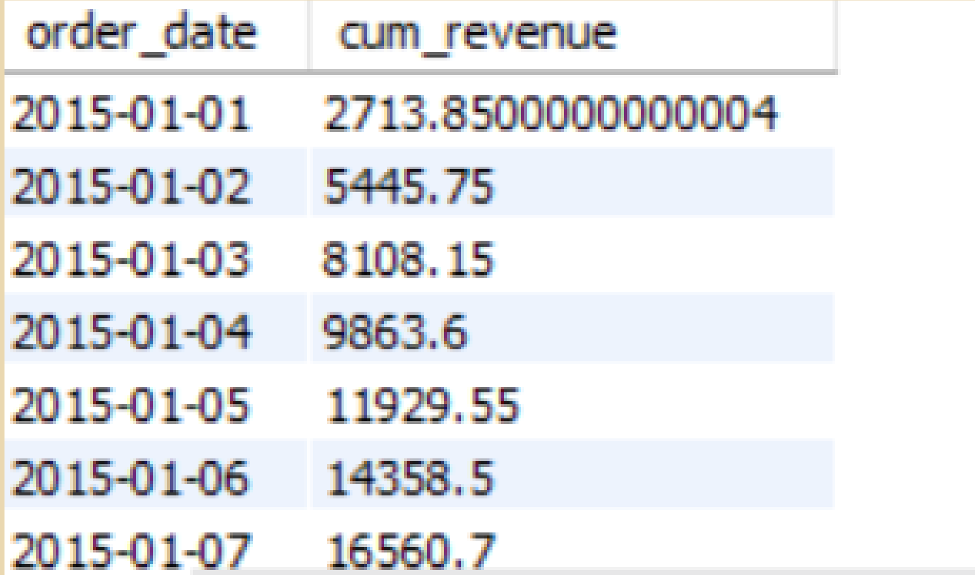
category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

12.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;
```

OUTPUT:



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

THANK YOU !!