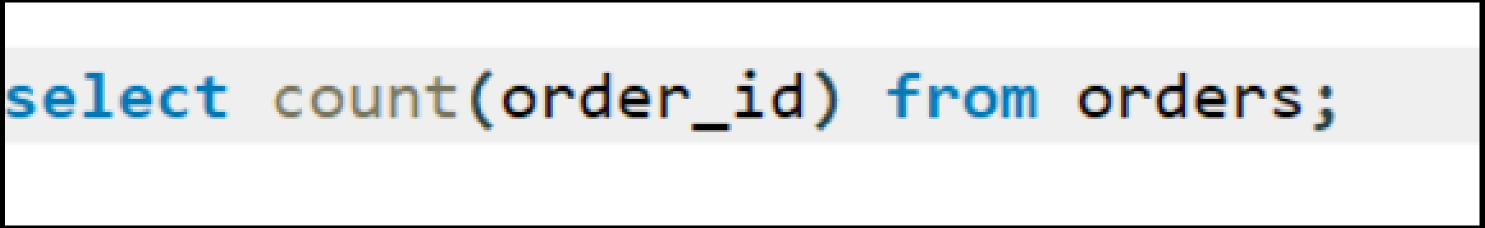
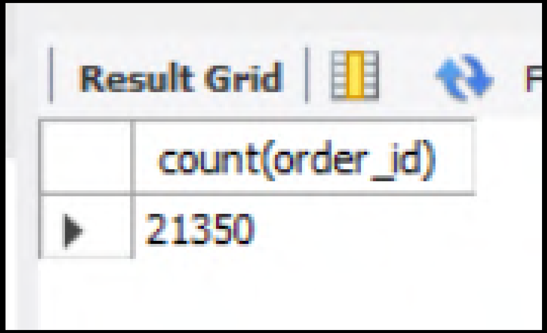
PIZZA SALES REPORT ANALYSIS

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

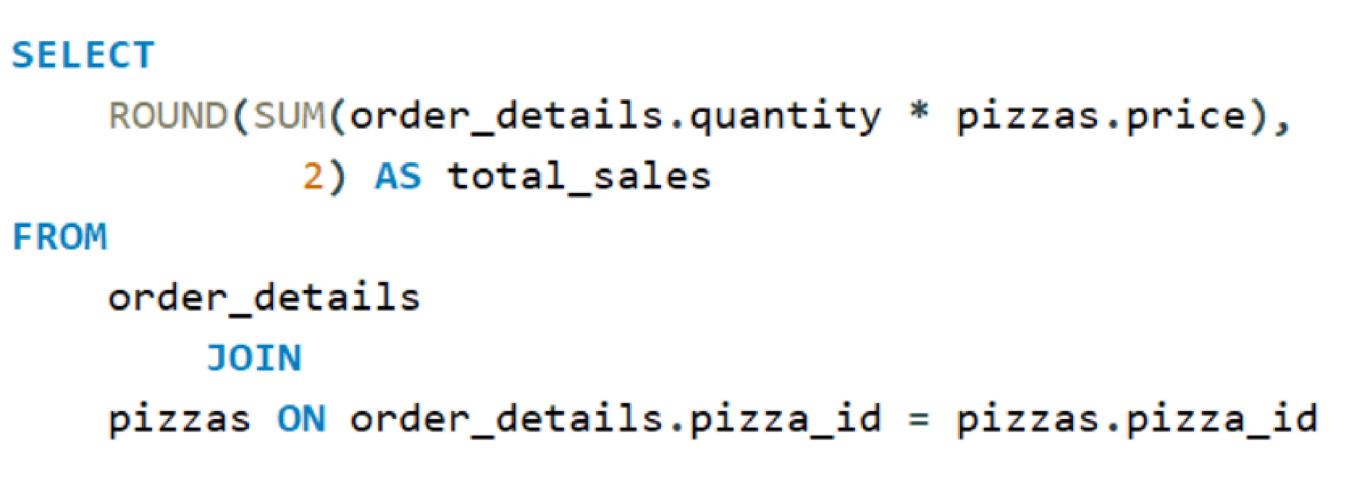


```sql
select count(order_id) from orders;
```

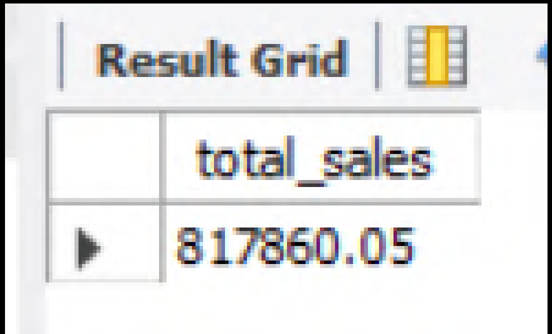


```
count(order_id)
21350
```

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

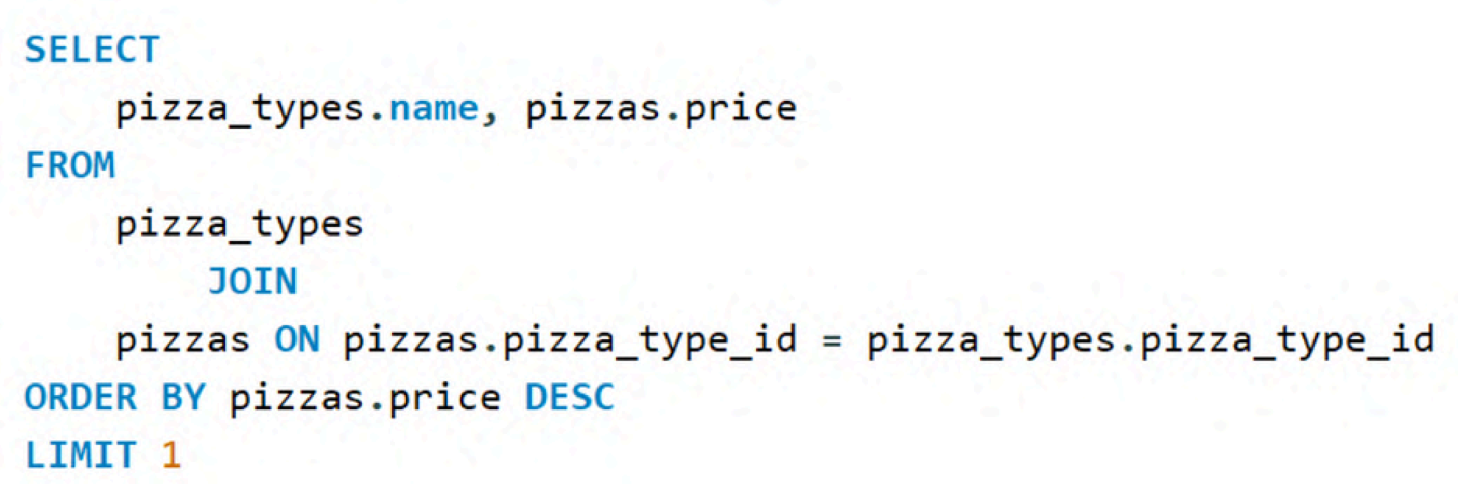


```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

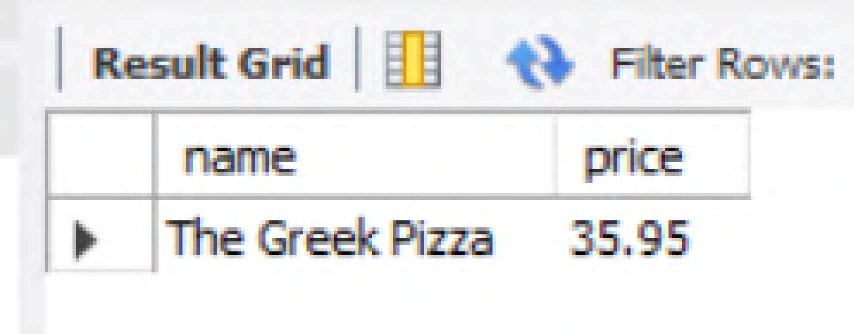


```
total_sales
817860.05
```

Q3 .IDENTIFY THE HIGHEST-PRICED PIZZA.

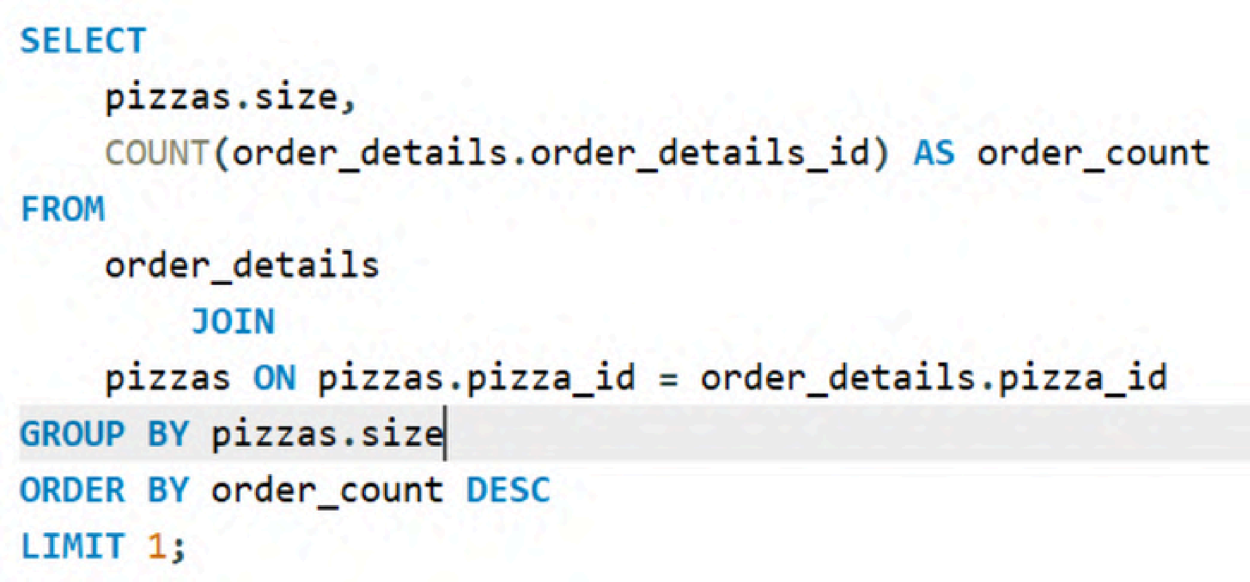


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1
```

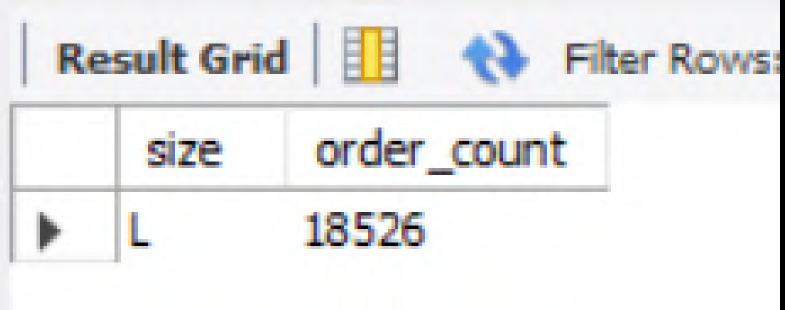


```
   name             price
▶  The Greek Pizza  35.95
```

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

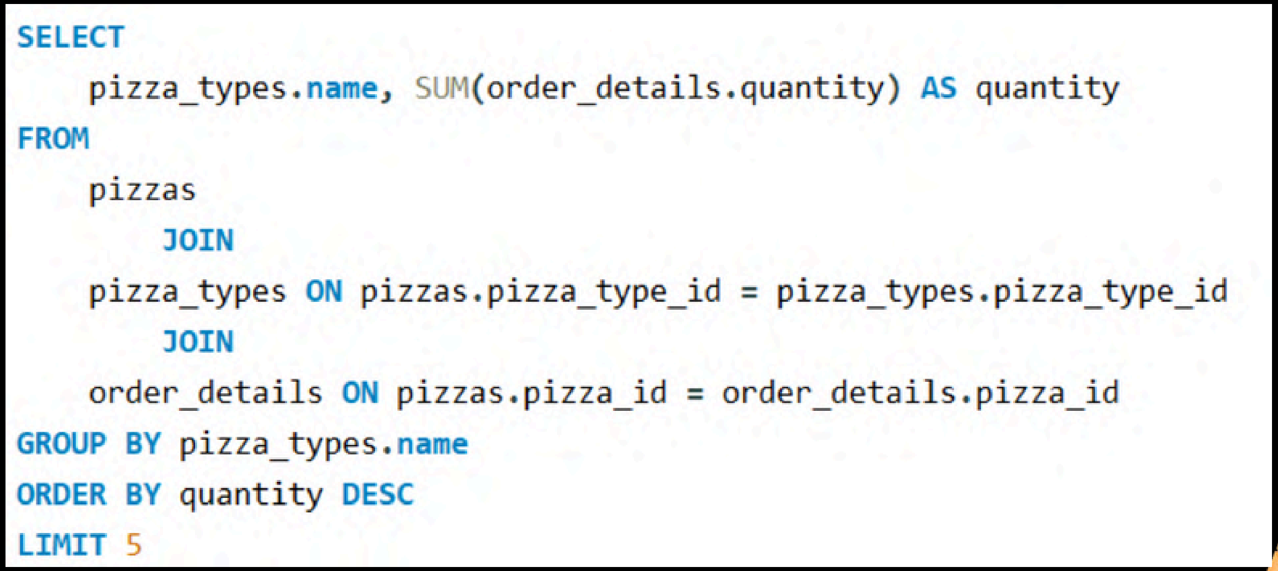


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

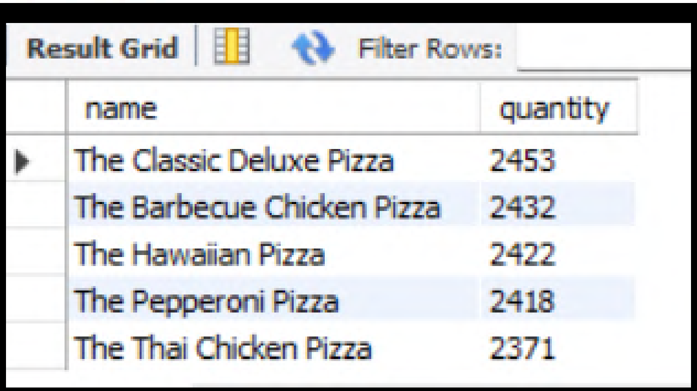


```
   size   order_count
▶  L      18526
```

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

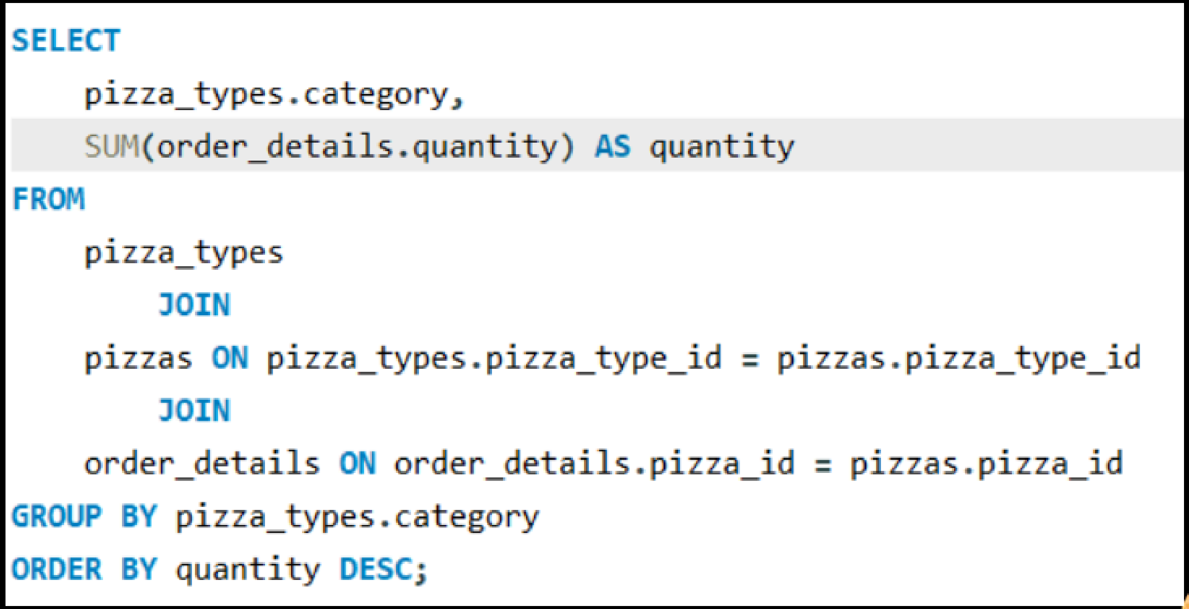


```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

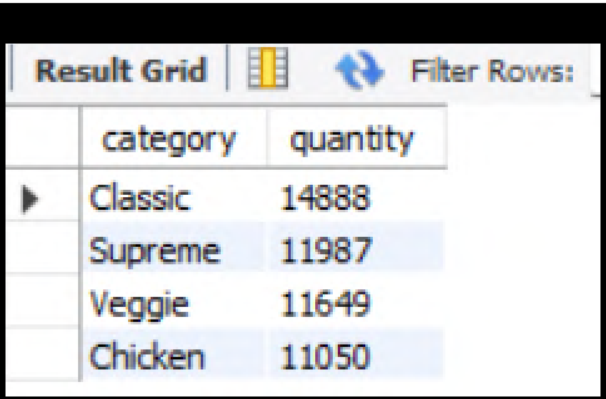


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

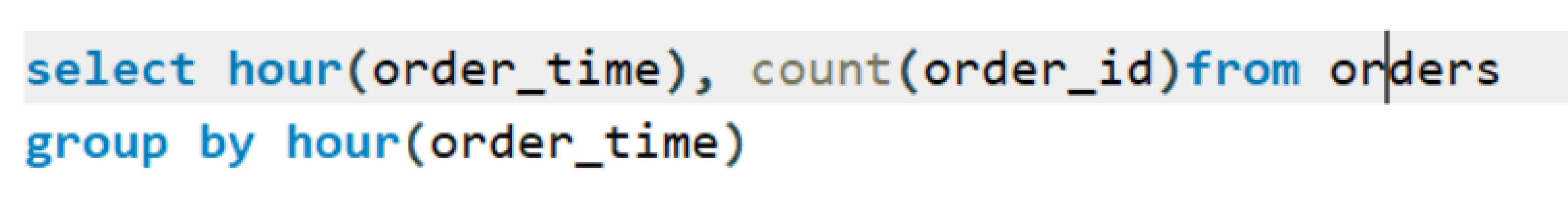


```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

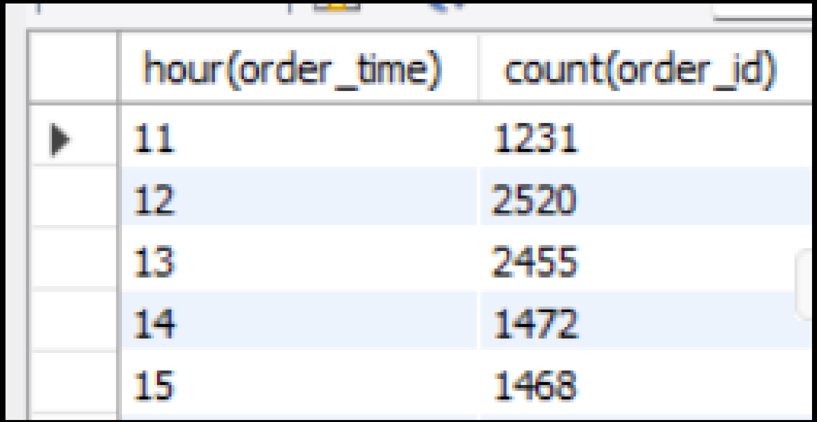


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

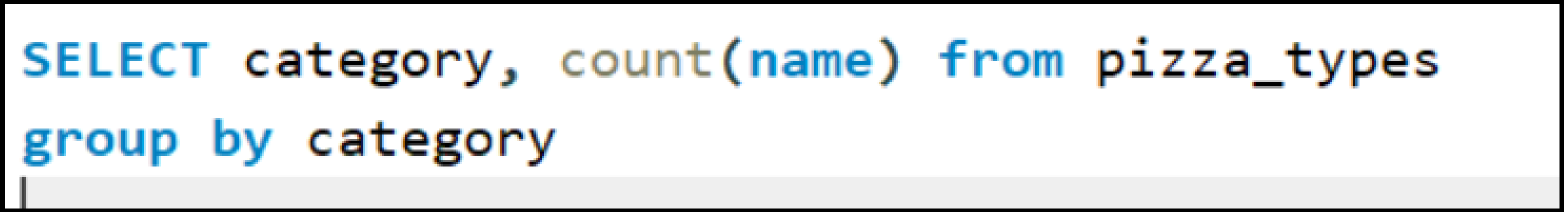


```
select hour(order_time), count(order_id)from orders
group by hour(order_time)
```

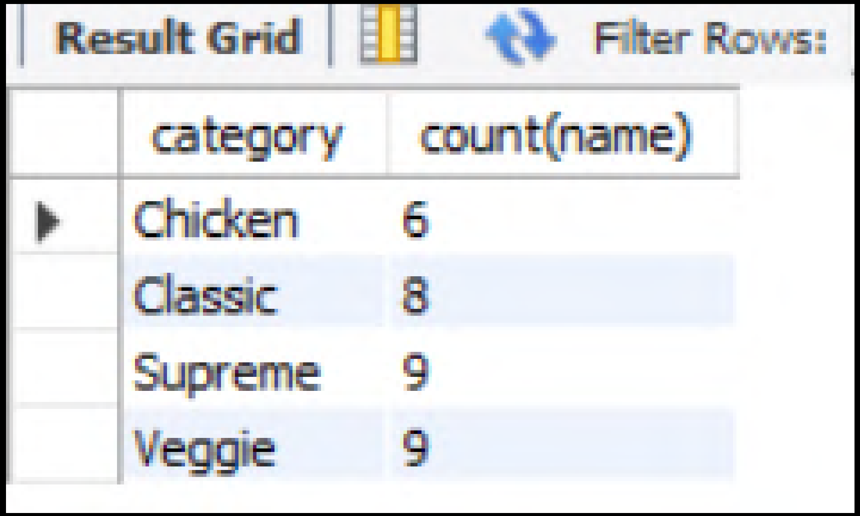


hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468

Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

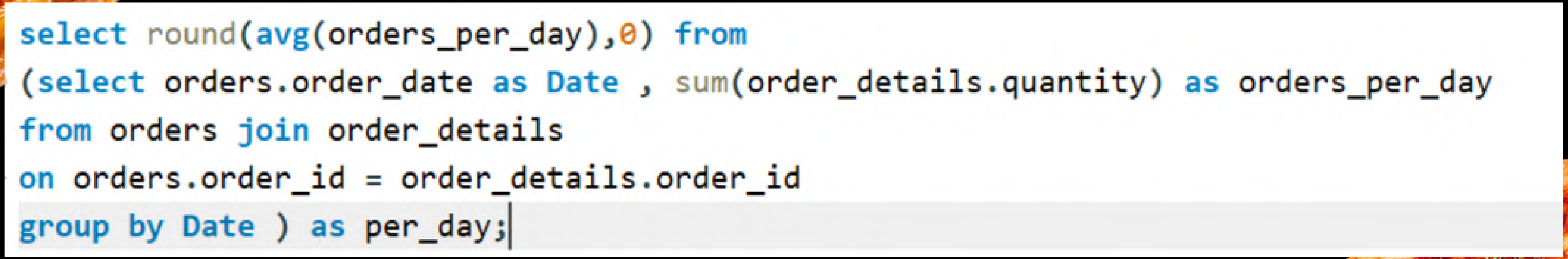


```
SELECT category, count(name) from pizza_types
group by category
```

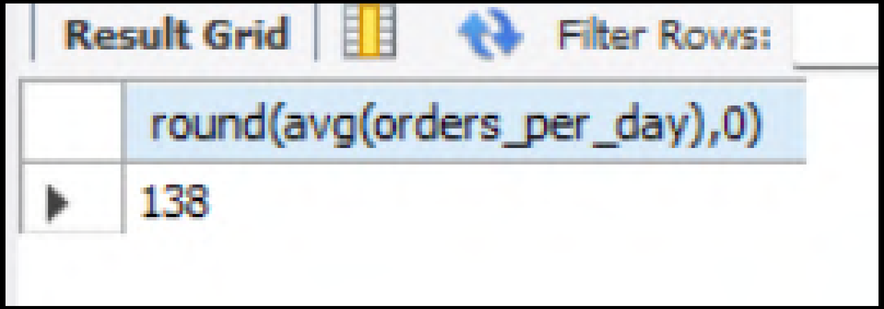


category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

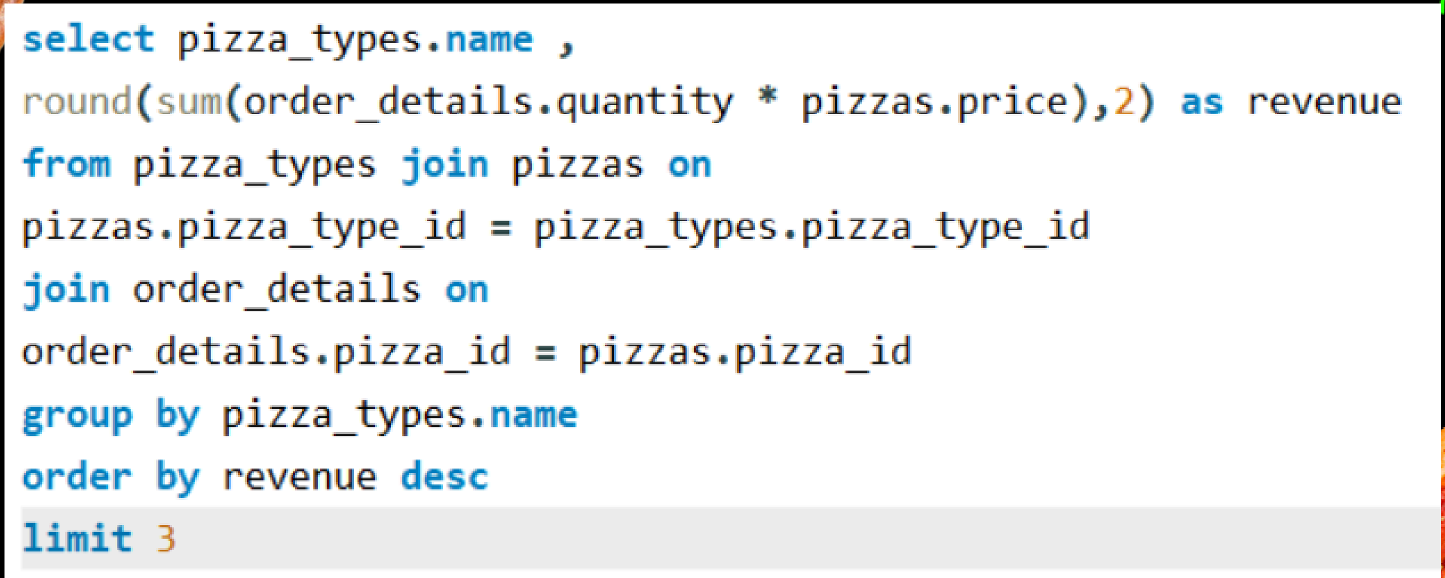


```
select round(avg(orders_per_day),0) from
(select orders.order_date as Date , sum(order_details.quantity) as orders_per_day
from orders join order_details
on orders.order_id = order_details.order_id
group by Date ) as per_day;
```

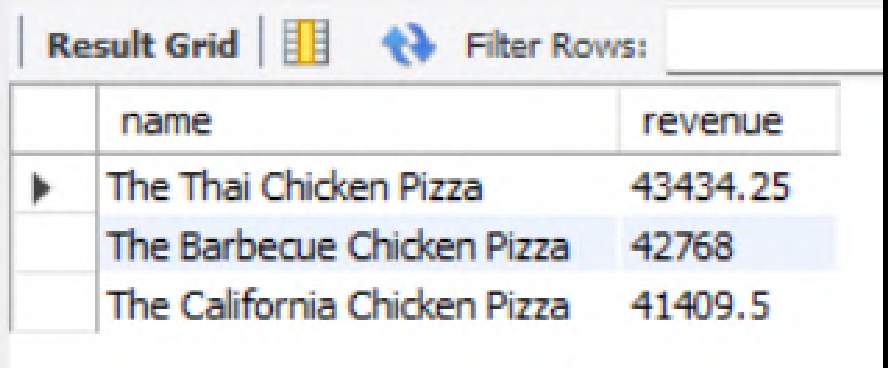


```
round(avg(orders_per_day),0)
138
```

Q10.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

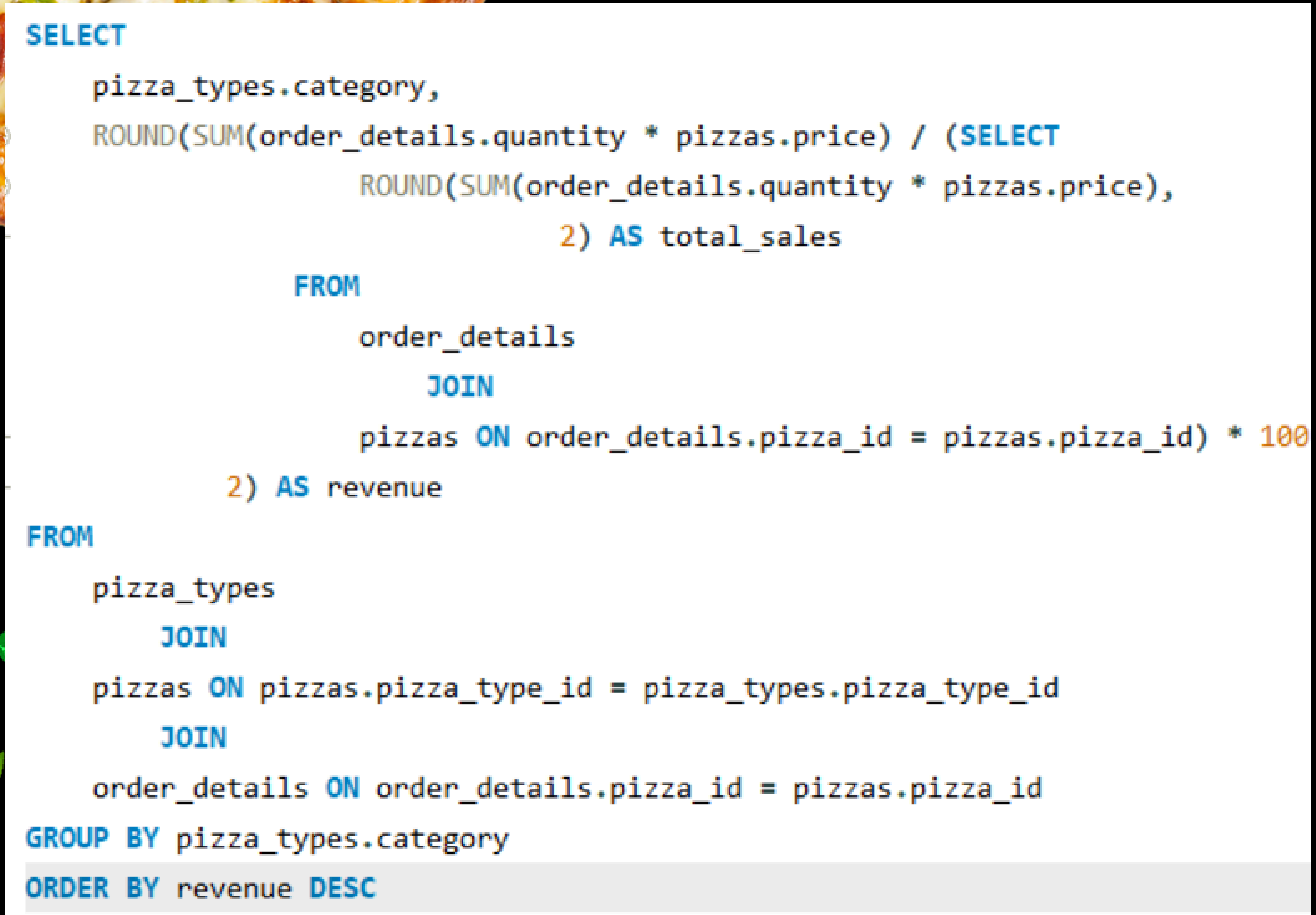


```
select pizza_types.name ,
round(sum(order_details.quantity * pizzas.price),2) as revenue
from pizza_types join pizzas on
pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details on
order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc
limit 3
```

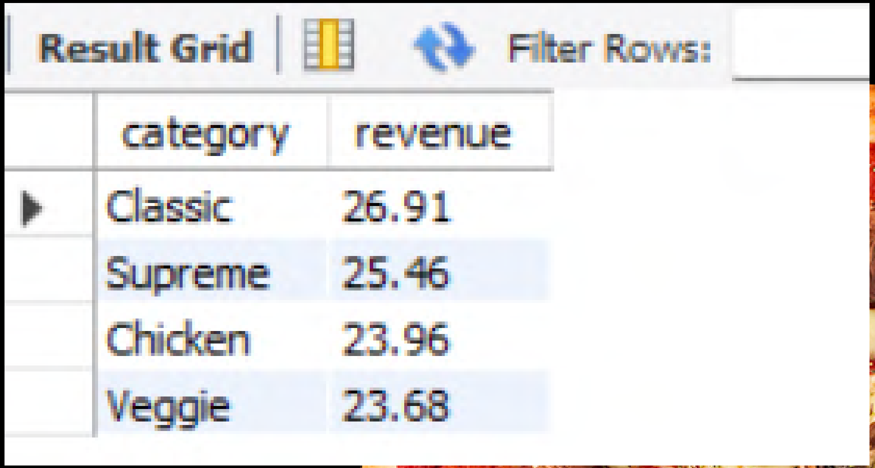


```
   name                          revenue
▶  The Thai Chicken Pizza        43434.25
   The Barbecue Chicken Pizza    42768
   The California Chicken Pizza  41409.5
```

Q11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

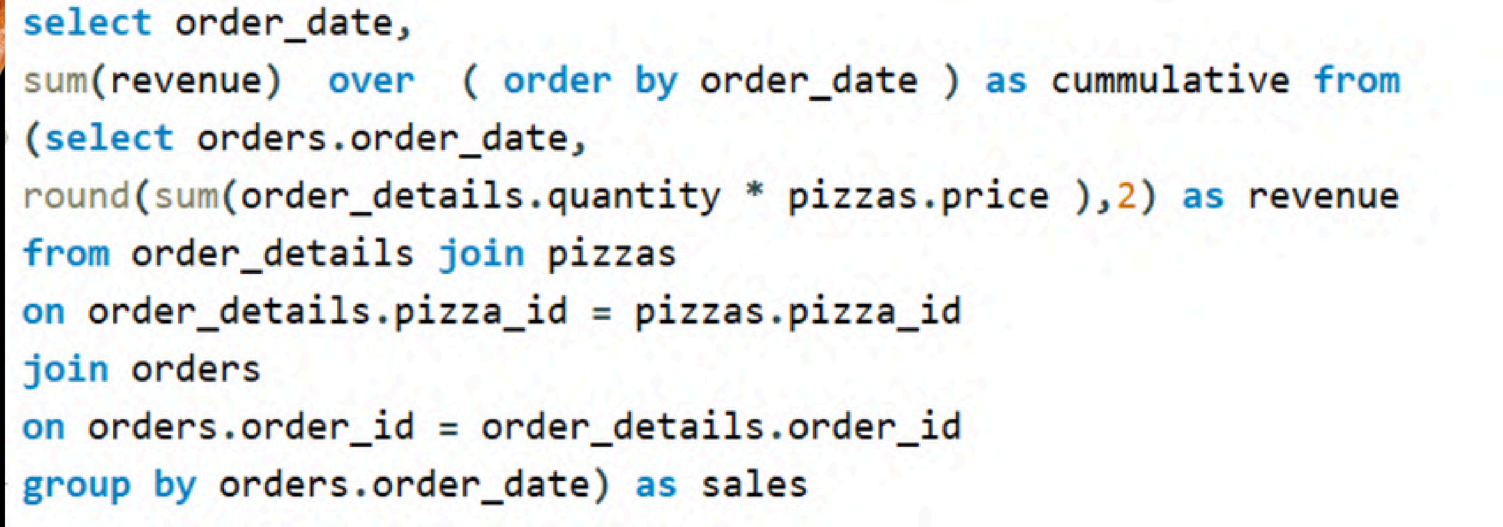


```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
                FROM
                    order_details
                        JOIN
        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

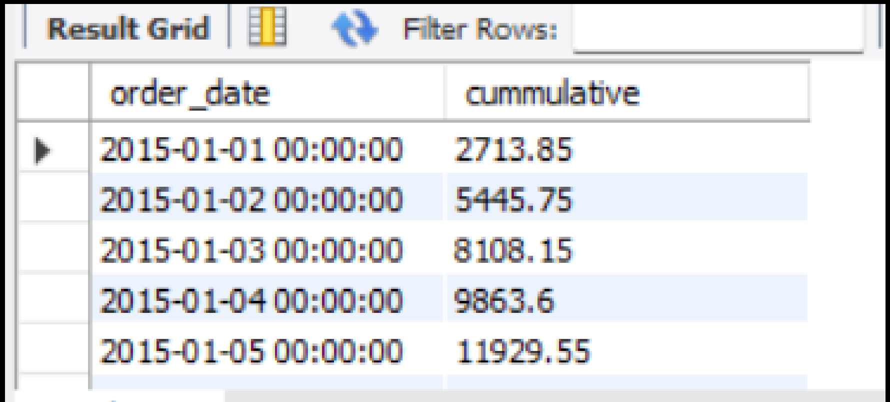


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
select order_date,
sum(revenue)  over  ( order by order_date ) as cummulative from
(select orders.order_date,
round(sum(order_details.quantity * pizzas.price ),2) 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            cummulative
2015-01-01 00:00:00   2713.85
2015-01-02 00:00:00   5445.75
2015-01-03 00:00:00   8108.15
2015-01-04 00:00:00   9863.6
2015-01-05 00:00:00   11929.55
```