SQL PROJECT

PIZZA SALES

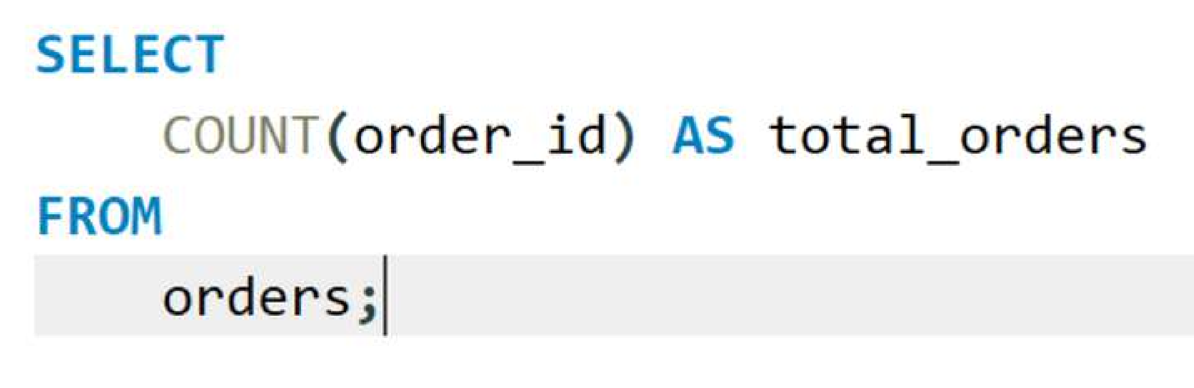
BY - ANSH RATHOD

ABOUT

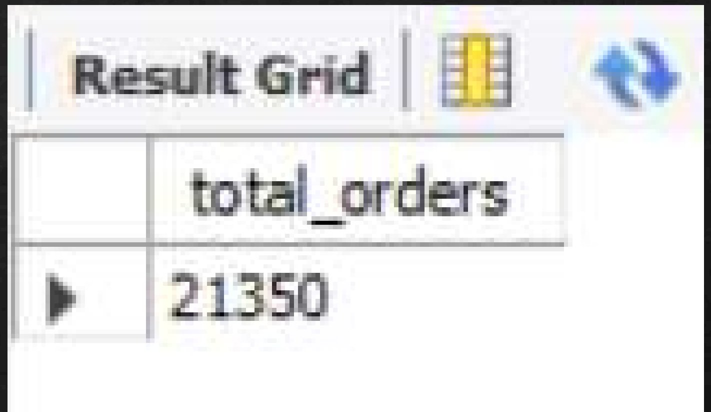
PIZZA SALES

THIS PROJECT ANALYZES PIZZA SALES DATA USING SQL TO EXTRACT ACTIONABLE BUSINESS INSIGHTS. IT COVERS TOTAL ORDERS, REVENUE, AND TOP-SELLING ITEMS, WHILE ALSO EXAMINING TRENDS BY TIME, CATEGORY, AND PERFORMANCE. THE ANALYSIS HELPS TO IDENTIFY SALES DRIVERS, CUSTOMER PREFERENCES, AND REVENUE PATTERNS, SUPPORTING SMARTER DECISIONS IN OPERATIONS AND MENU PLANNING.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

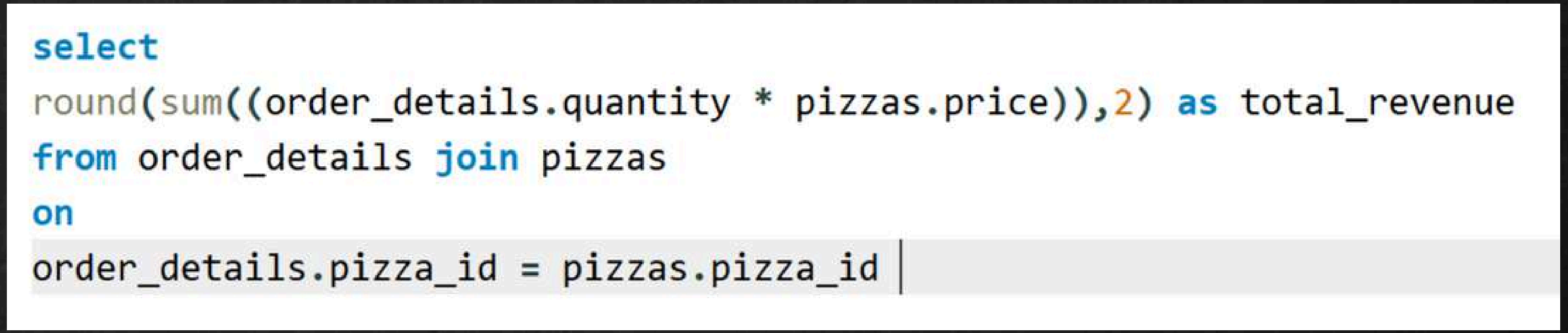


```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

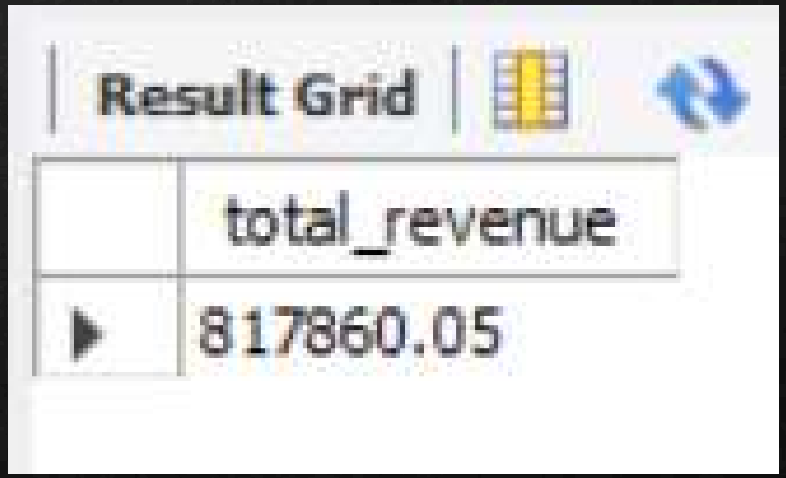


```
  total_orders
▶ 21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

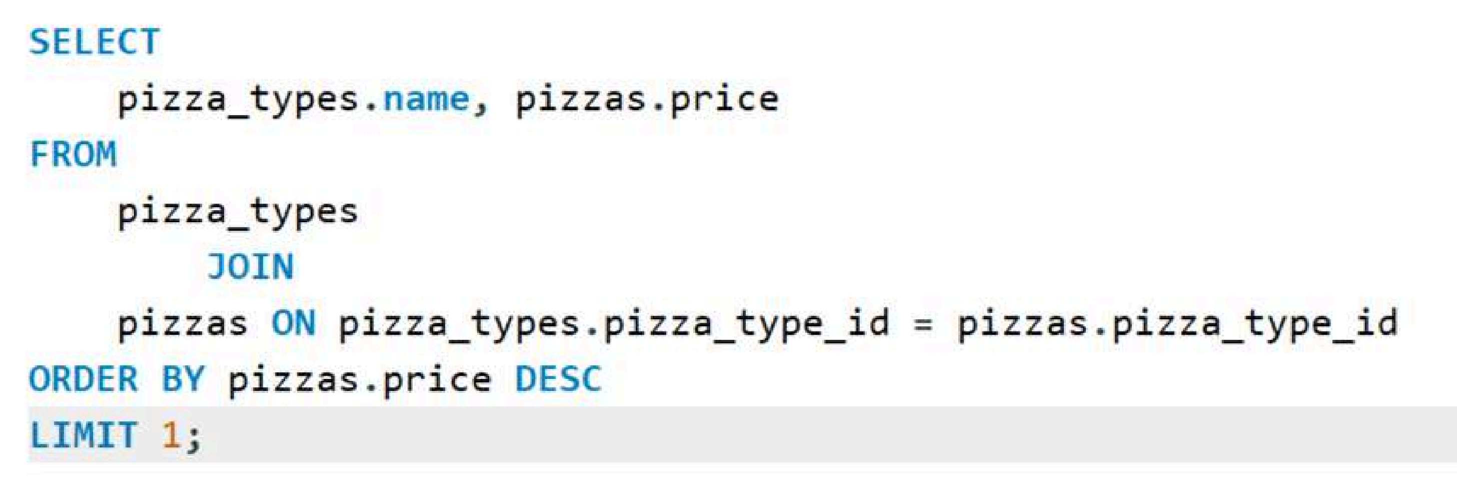


```
select
round(sum((order_details.quantity * pizzas.price)),2) as total_revenue
from order_details join pizzas
on
order_details.pizza_id = pizzas.pizza_id
```

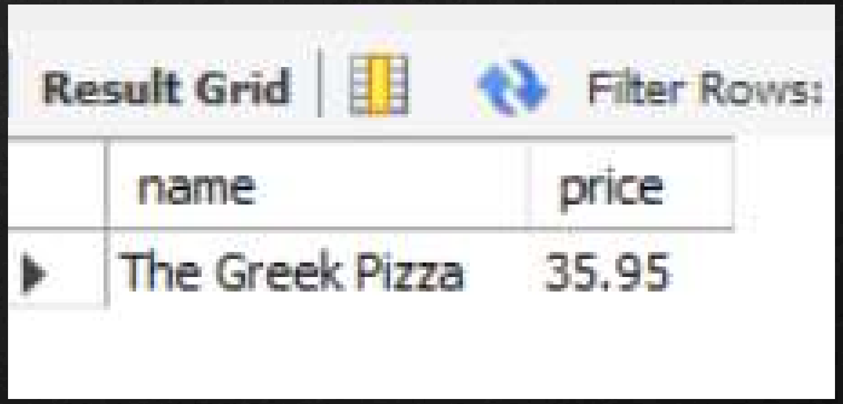


total_revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

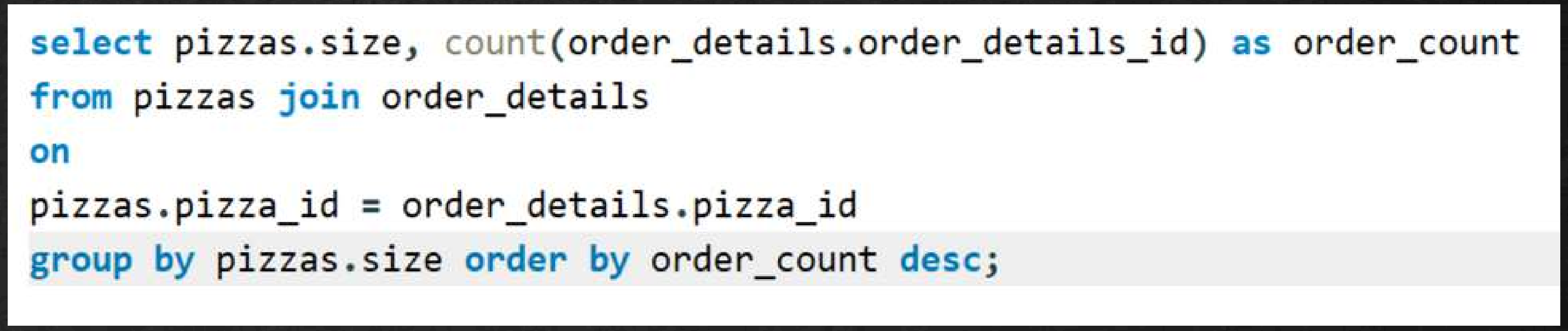


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

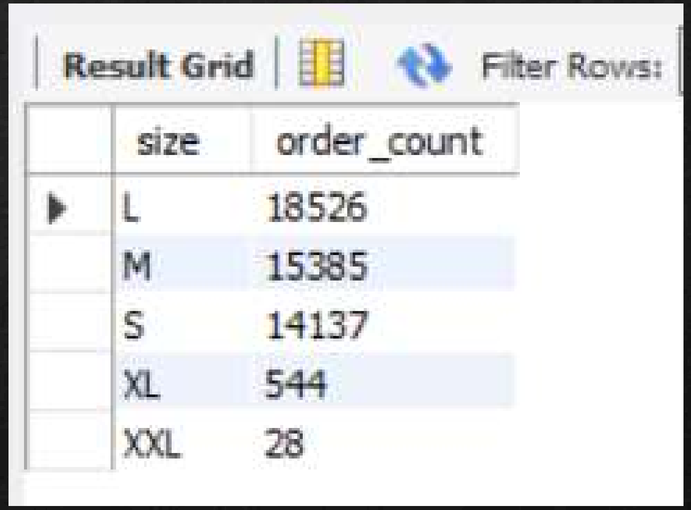


name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

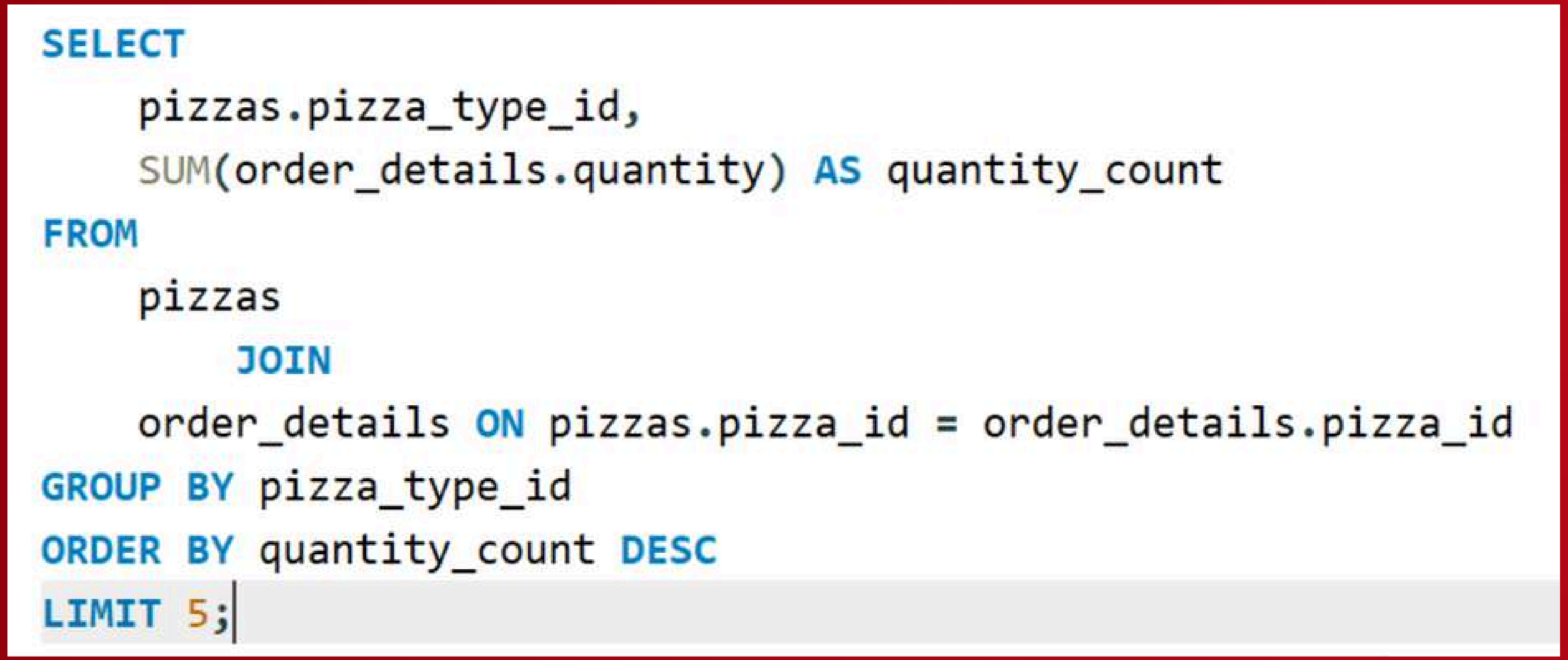


```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on
pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

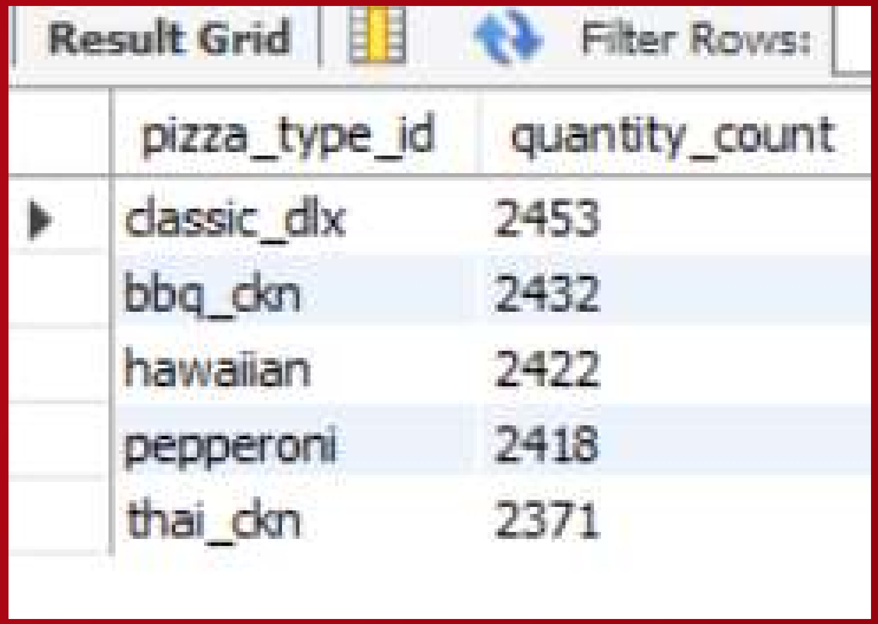


```
  size  order_count
▶ L     18526
  M     15385
  S     14137
  XL    544
  XXL   28
```

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

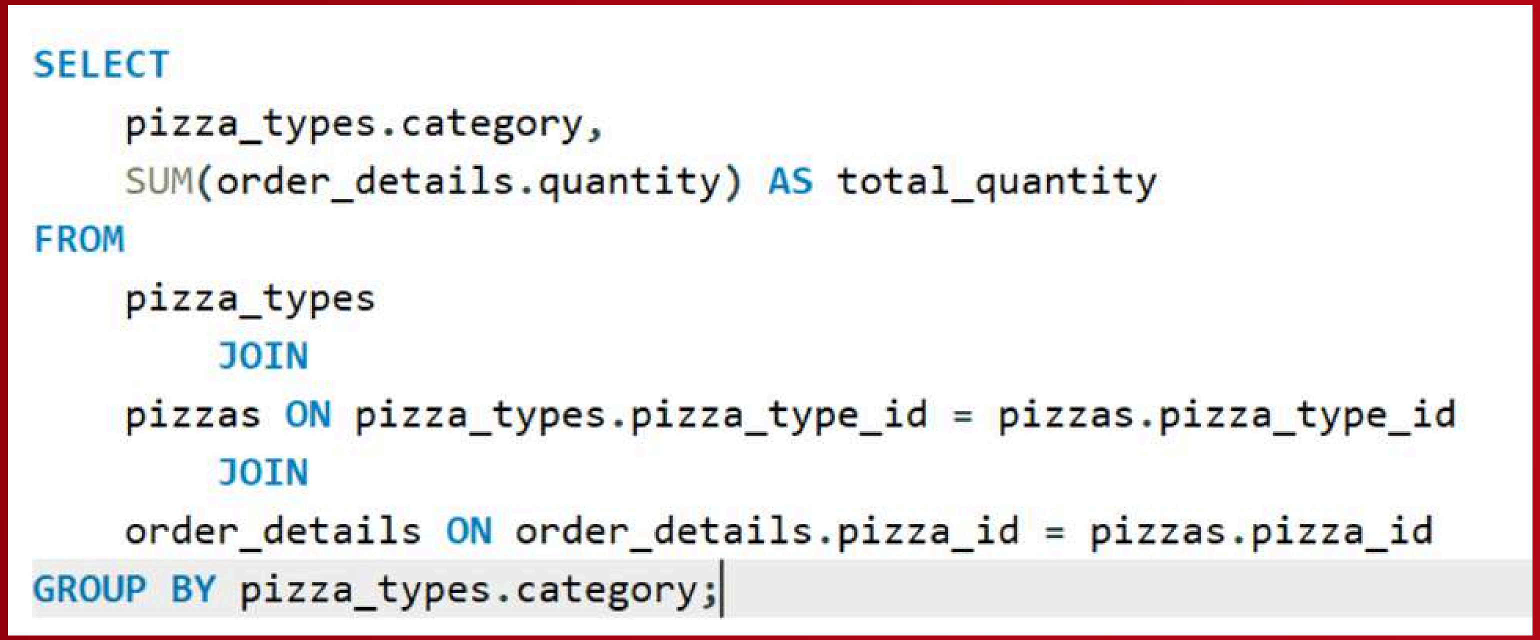


```sql
SELECT
    pizzas.pizza_type_id,
    SUM(order_details.quantity) AS quantity_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_type_id
ORDER BY quantity_count DESC
LIMIT 5;
```

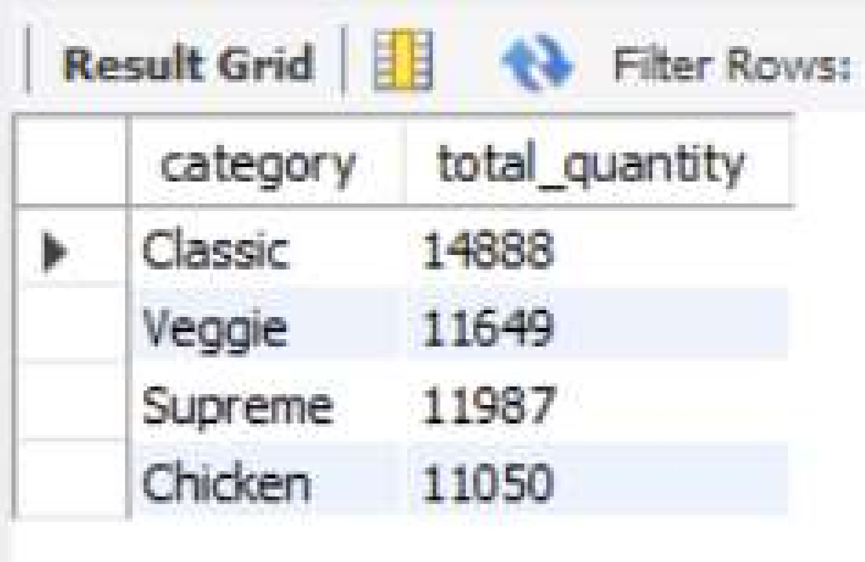


pizza_type_id	quantity_count
classic_dlx	2453
bbq_ckn	2432
hawaiian	2422
pepperoni	2418
thai_ckn	2371

FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

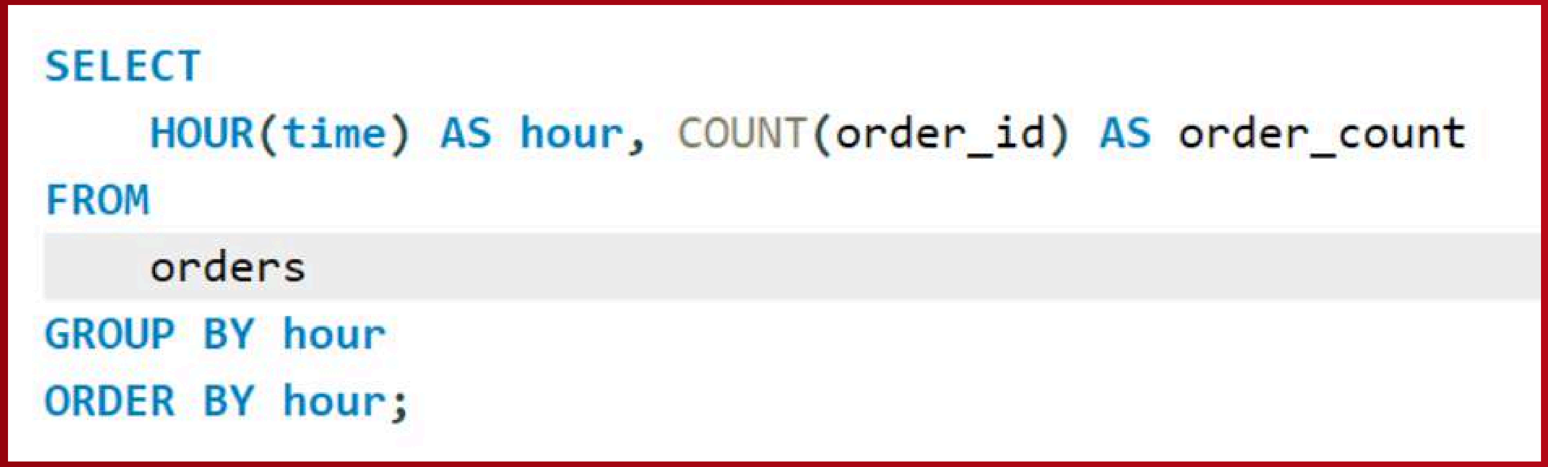


```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

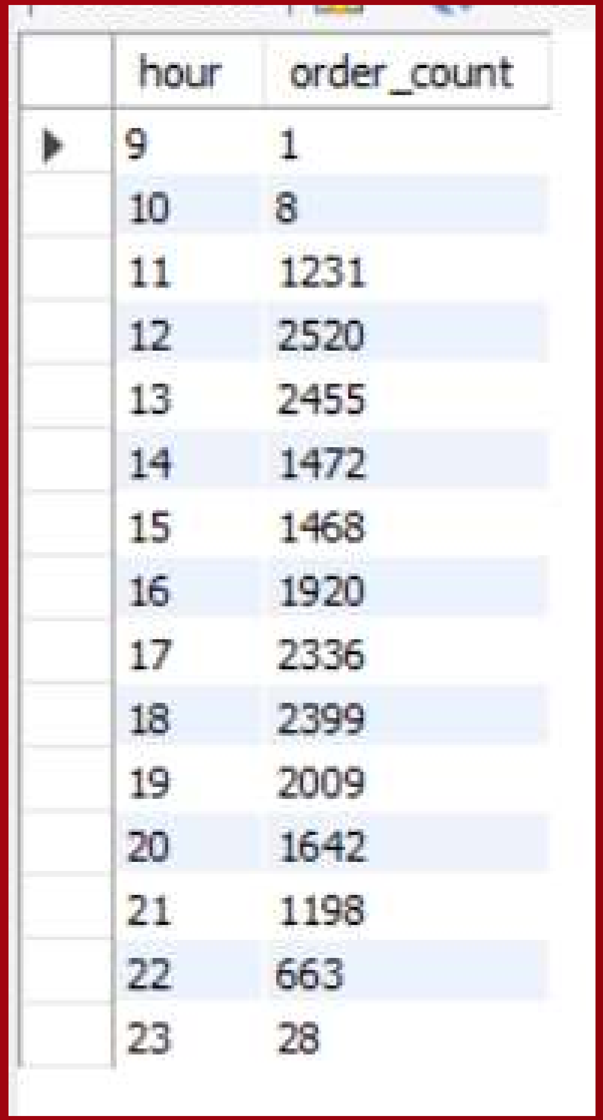


category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

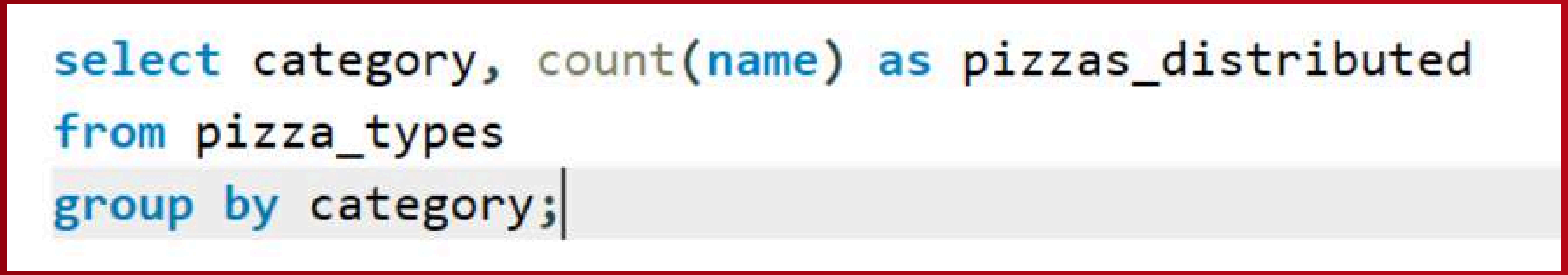


```
SELECT
    HOUR(time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour
ORDER BY hour;
```

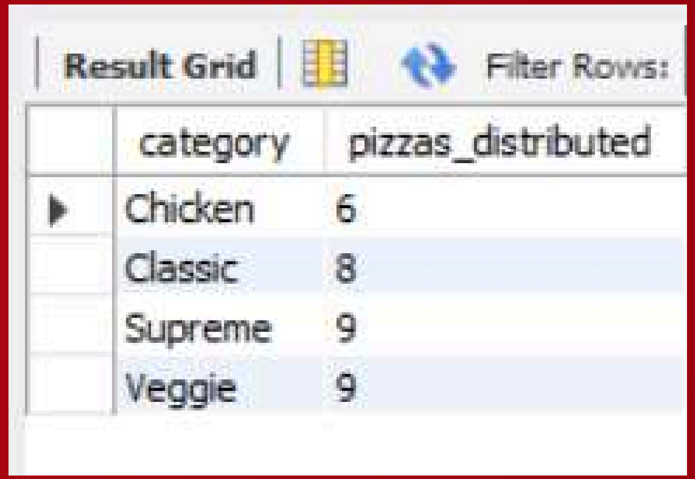


hour	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

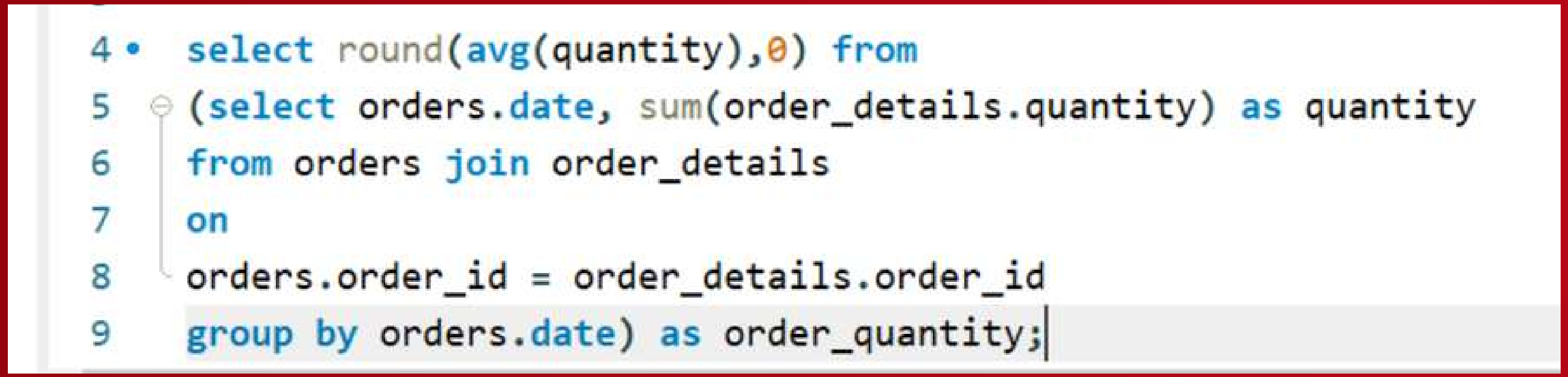


```sql
select category, count(name) as pizzas_distributed
from pizza_types
group by category;
```

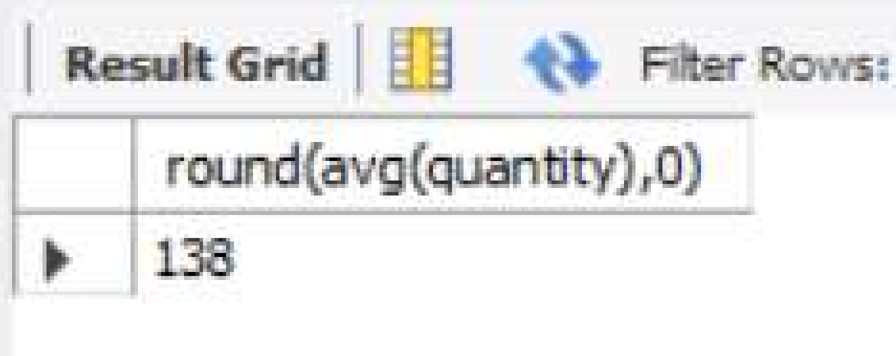


category	pizzas_distributed
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

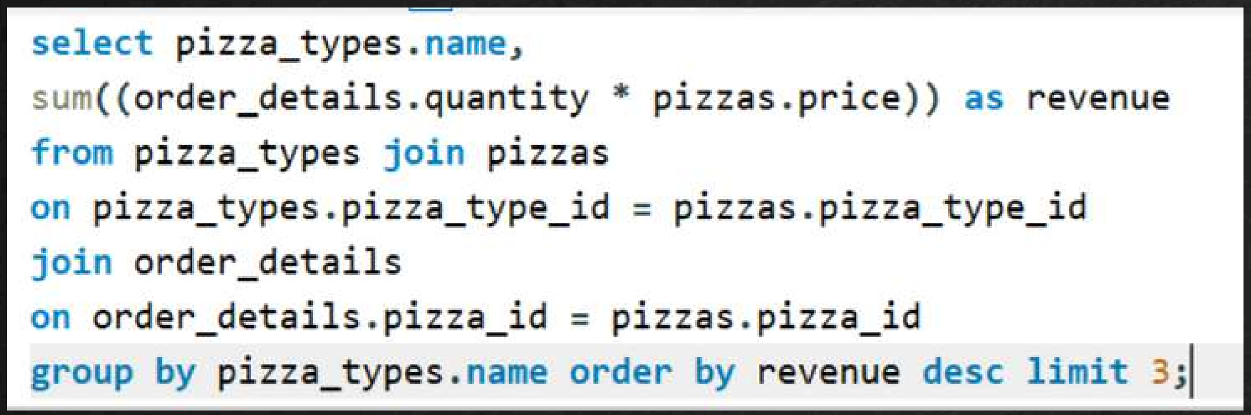


```
select round(avg(quantity),0) from
(select orders.date, sum(order_details.quantity) as quantity
from orders join order_details
on
orders.order_id = order_details.order_id
group by orders.date) as order_quantity;
```

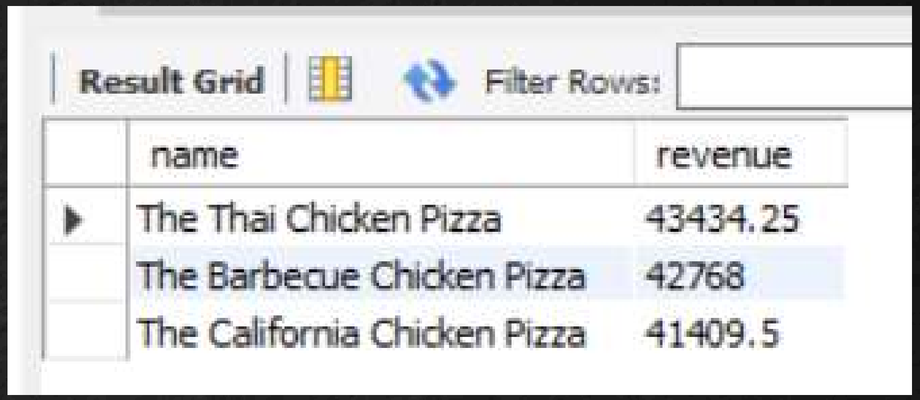


```
round(avg(quantity),0)
138
```

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

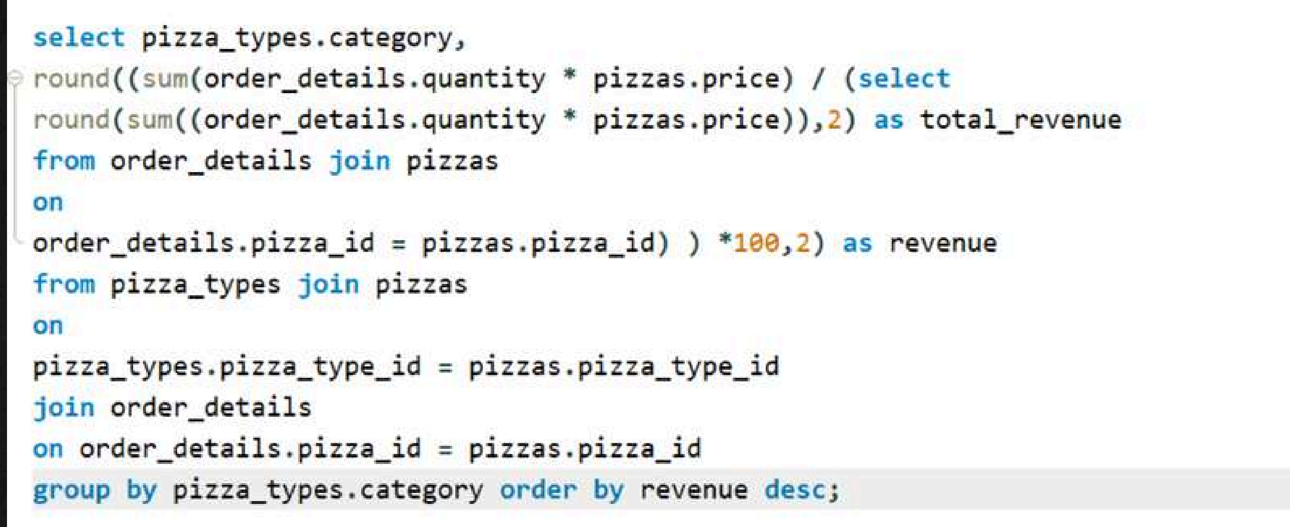


```
select pizza_types.name,
sum((order_details.quantity * pizzas.price)) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

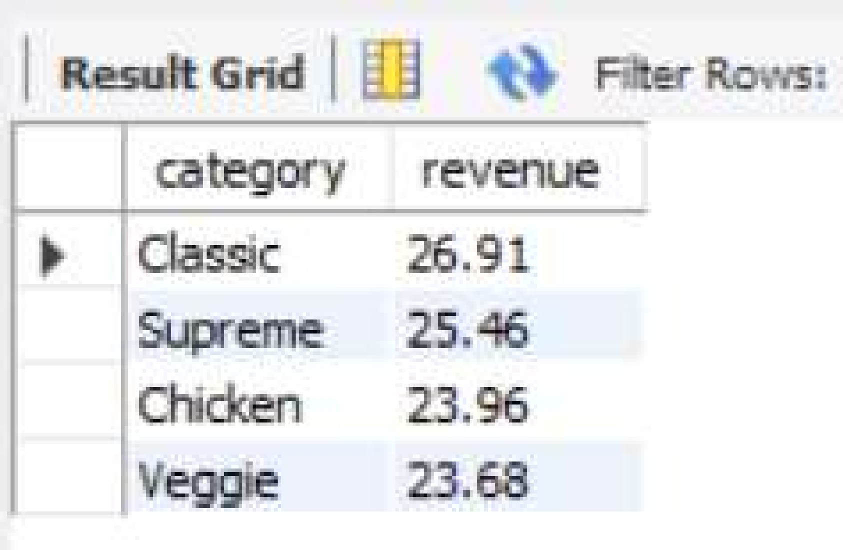


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

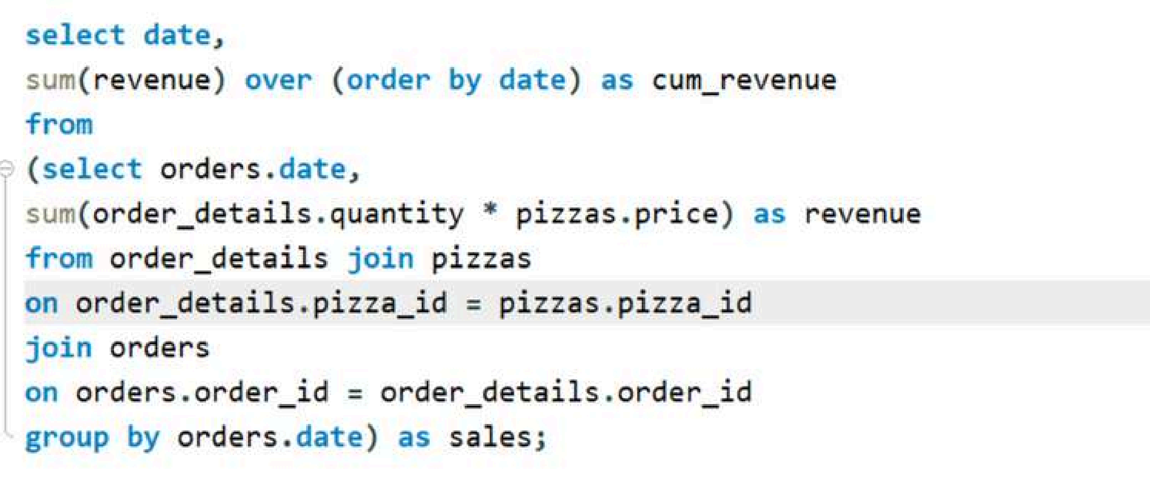


```
select pizza_types.category,
round((sum(order_details.quantity * pizzas.price) / (select
round(sum((order_details.quantity * pizzas.price)),2) as total_revenue
from order_details join pizzas
on
order_details.pizza_id = pizzas.pizza_id) ) *100,2) as revenue
from pizza_types join pizzas
on
pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
```

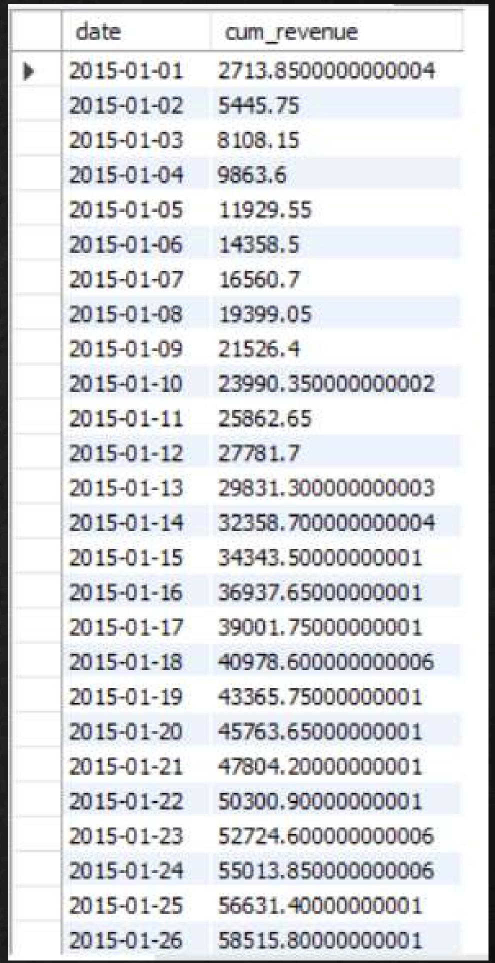


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

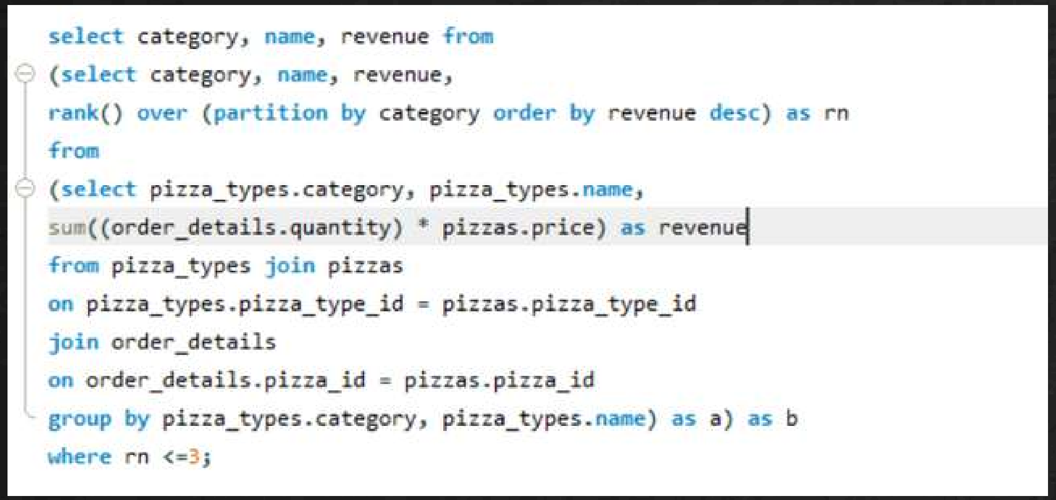


```
select date,
sum(revenue) over (order by date) as cum_revenue
from
(select orders.date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as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

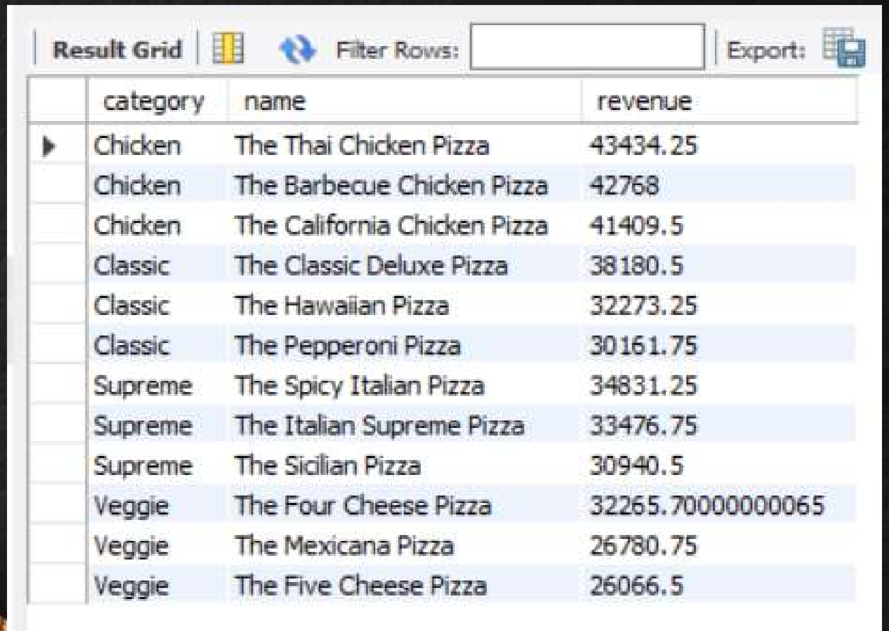


date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001

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



```sql
select category, 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;
```



category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU!