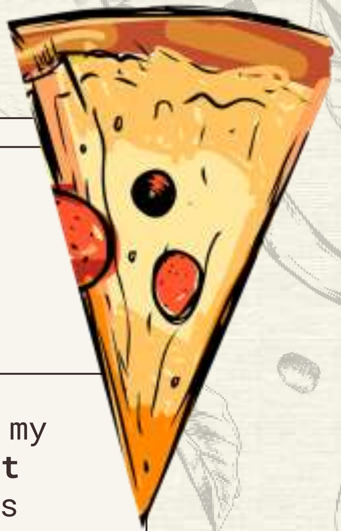
SQL PROJECT ON PIZZA SALES

Powered By-

Vikas G R

INTRODUCTION



Hello! My name is Vikas G R, and I am excited to share my project on creating a comprehensive **Pizza Sales Report** using SQL. This project involves analyzing pizza sales data through queries that provide insights into various aspects of the sales process. The report highlights key metrics, such as the most popular pizza types and sales trends over time. By leveraging SQL's capabilities, I uncover valuable information to inform business decisions and optimize sales strategies.

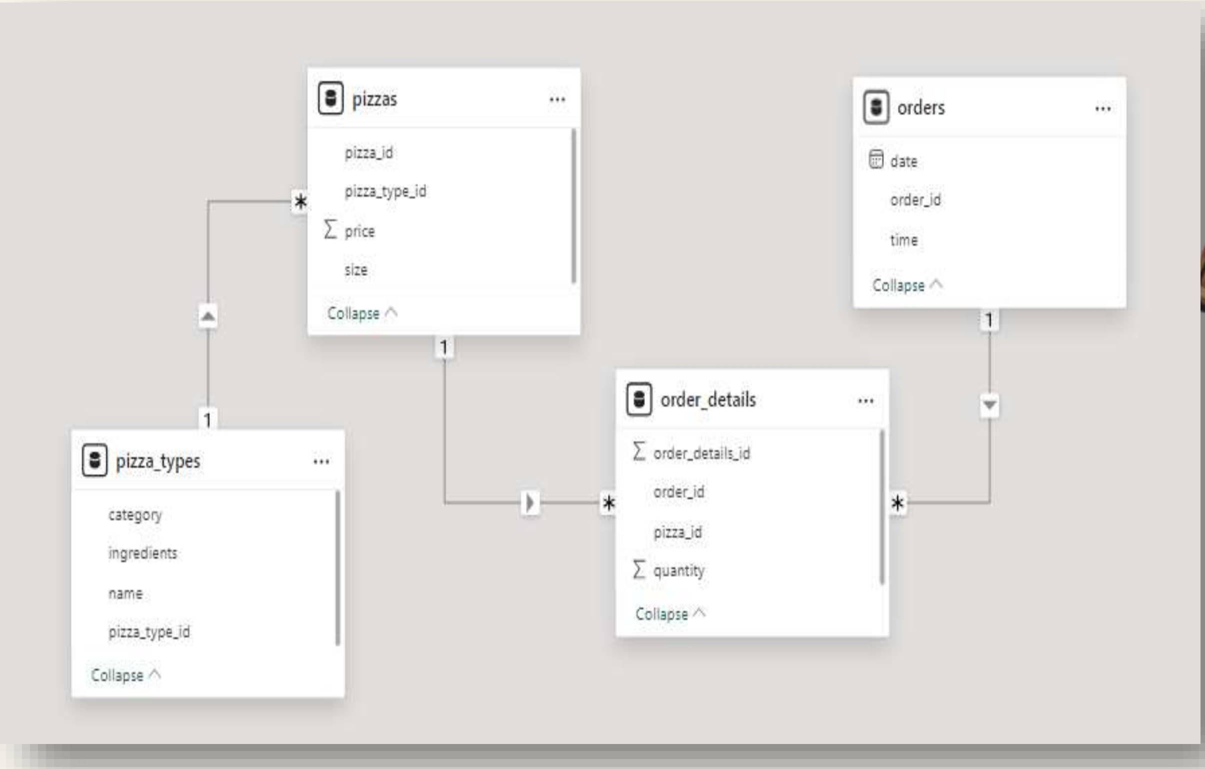
https://github.com/Vikasgr36/Pizza_Sales---SQL

—Key Queries Addressed

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.



pizzas

pizza_id

pizza_type_id

∑ price

size

Collapse ^

orders

date

order_id

time

Collapse ^

pizza_types

category

ingredients

name

pizza_type_id

Collapse ^

order_details

∑ order_details_id

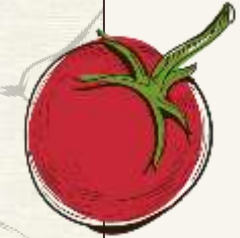
order_id

pizza_id

∑ quantity

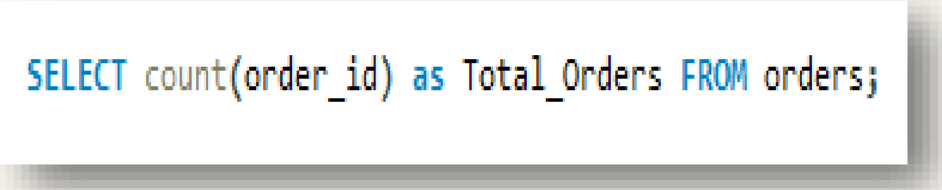
Collapse ^

Question 1



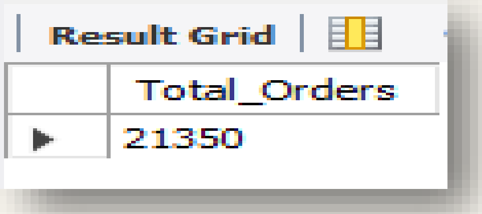
Retrieve the total number of orders placed.

Query :



```sql
SELECT count(order_id) as Total_Orders FROM orders;
```

Result :

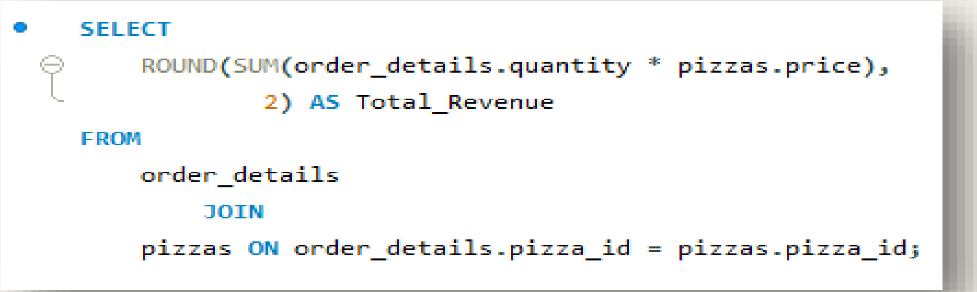


	Total_Orders
▶	21350

Question 2

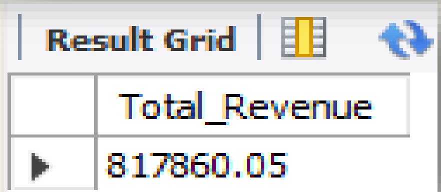
Calculate the total revenue generated from pizza sales.

Query :



```
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;
```

Result :

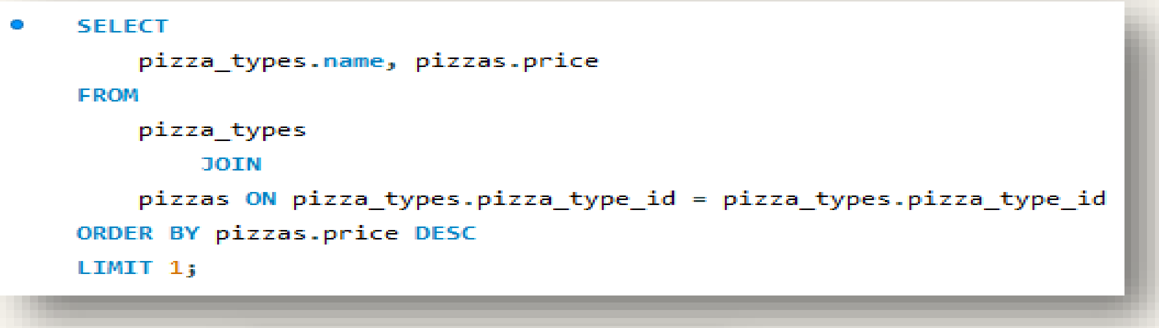


	Total_Revenue
▶	817860.05

Question 3

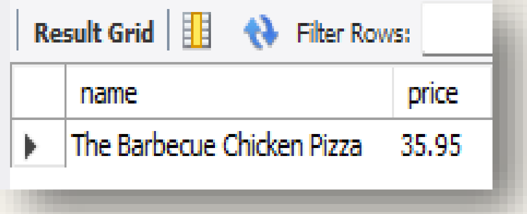
Identify the highest-priced pizza.

Query :



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

Result :

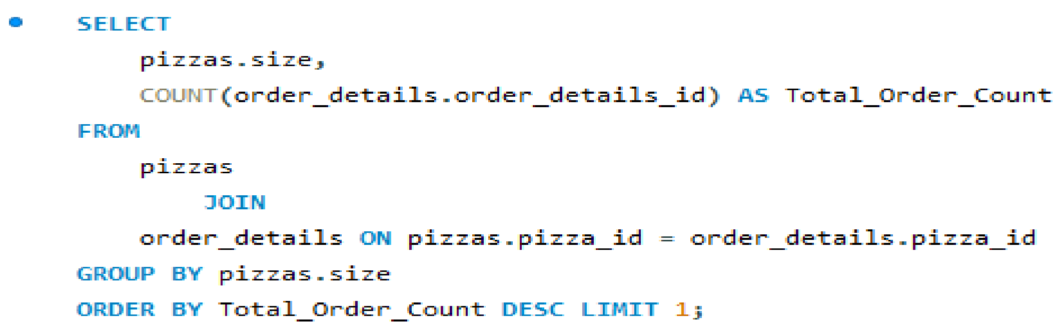


	name	price
▶	The Barbecue Chicken Pizza	35.95

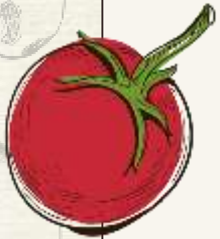
Question 4

Identify the most common pizza size ordered.

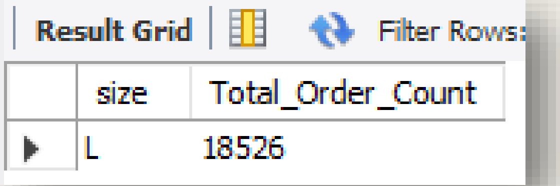
Query :



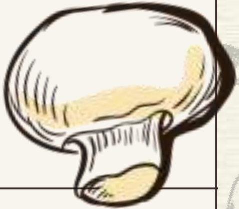
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Total_Order_Count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_Order_Count DESC LIMIT 1;
```



Result :



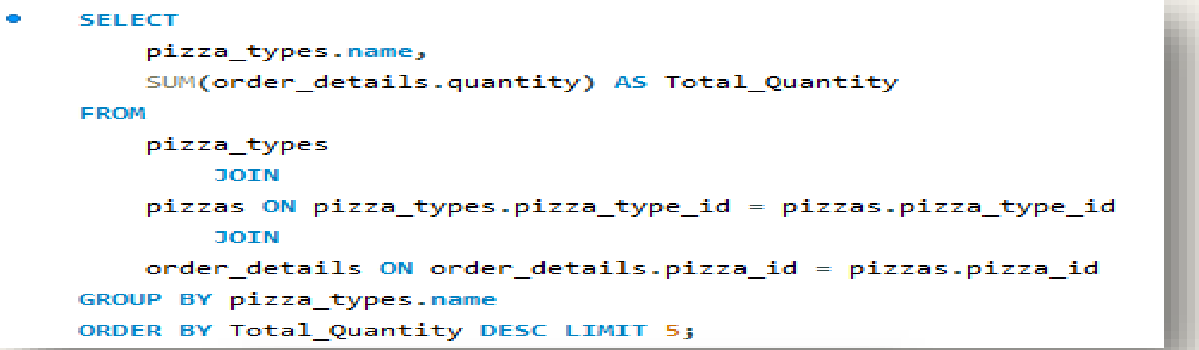
size	Total_Order_Count
L	18526



Question 5

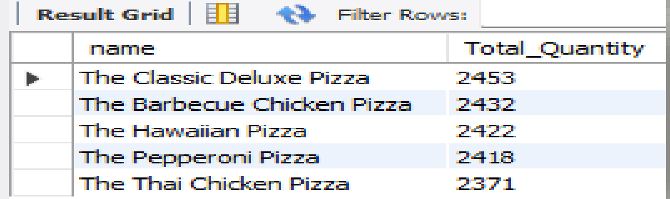
List the top 5 most ordered pizza types along with their quantities.

Query :



```
SELECT
    pizza_types.name,
    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.name
ORDER BY Total_Quantity DESC LIMIT 5;
```

Result :

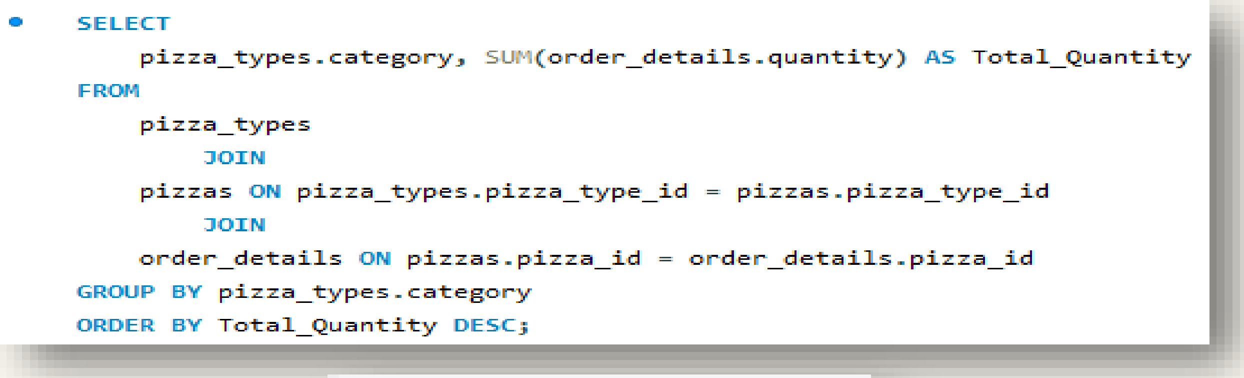


name	Total_Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Question 6

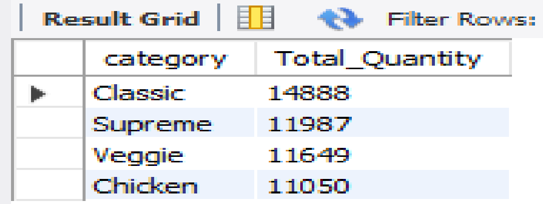
Join the necessary tables to find the total quantity of each pizza category ordered.

Query :



```
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_Quantity DESC;
```

Result :

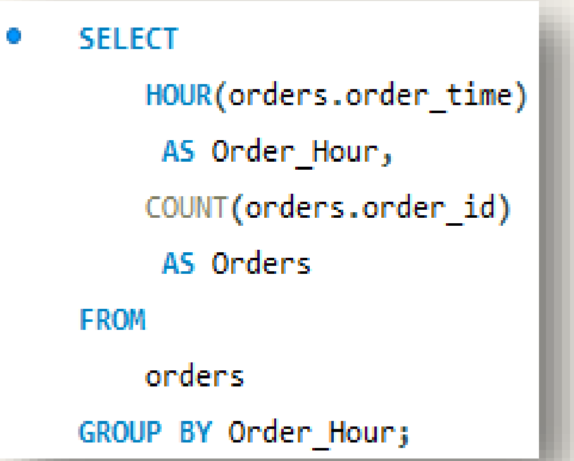


category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Question 7

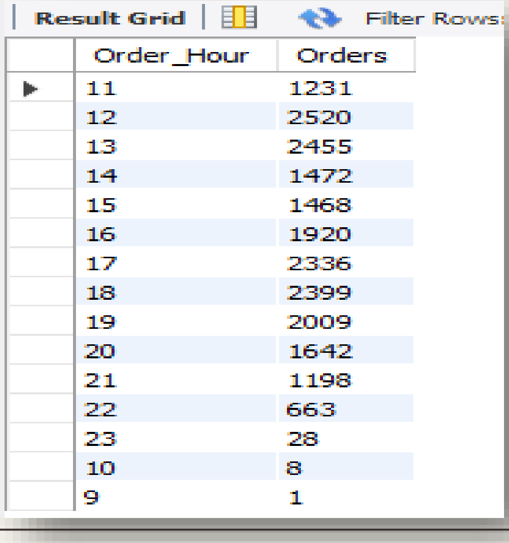
Determine the distribution of orders by hour of the day.

Query :



```
•   SELECT
        HOUR(orders.order_time)
         AS Order_Hour,
        COUNT(orders.order_id)
         AS Orders
    FROM
        orders
    GROUP BY Order_Hour;
```

Result :

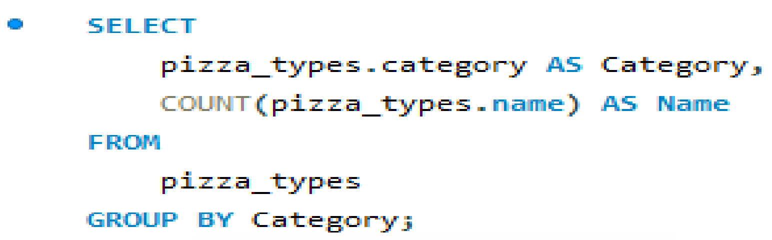


Order_Hour	Orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Question 8

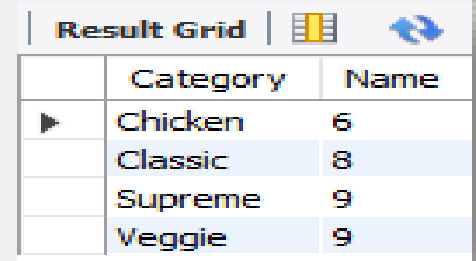
Join relevant tables to find the category-wise distribution of pizzas.

Query :

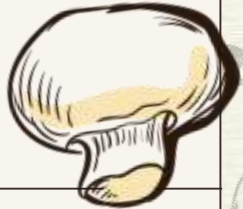


```
SELECT
    pizza_types.category AS Category,
    COUNT(pizza_types.name) AS Name
FROM
    pizza_types
GROUP BY Category;
```

Result :



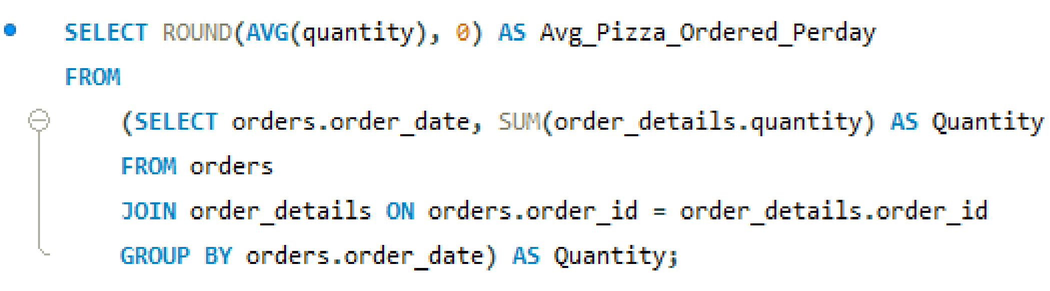
Category	Name
Chicken	6
Classic	8
Supreme	9
Veggie	9



Question 9

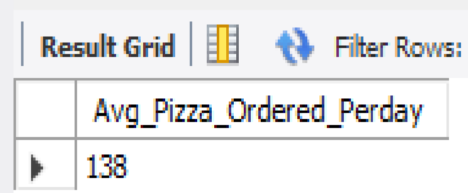
Group the orders by date and calculate the average number of pizzas ordered per day.

Query :



```
SELECT ROUND(AVG(quantity), 0) AS Avg_Pizza_Ordered_Perday
FROM
    (SELECT orders.order_date, SUM(order_details.quantity) AS Quantity
    FROM orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS Quantity;
```

Result :

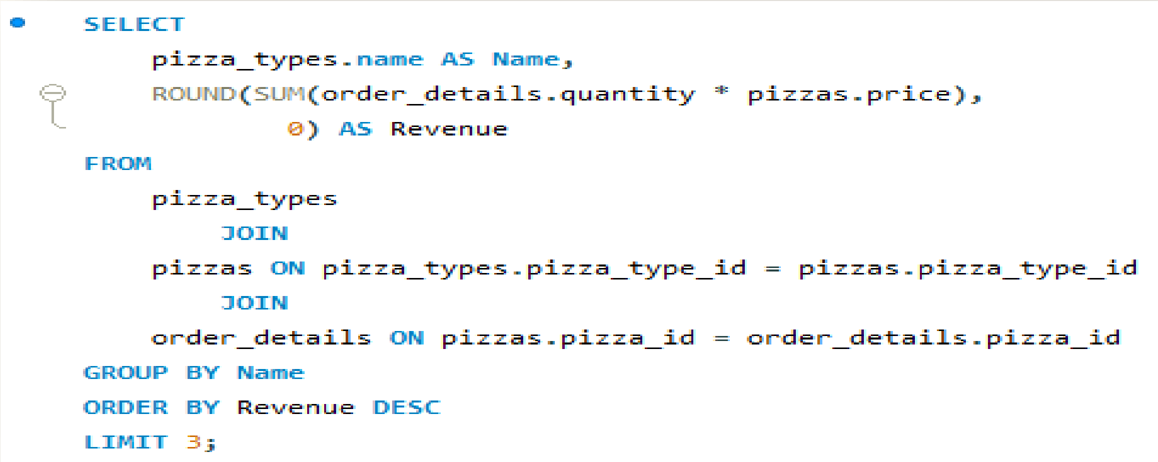


Avg_Pizza_Ordered_Perday
138

Question 10

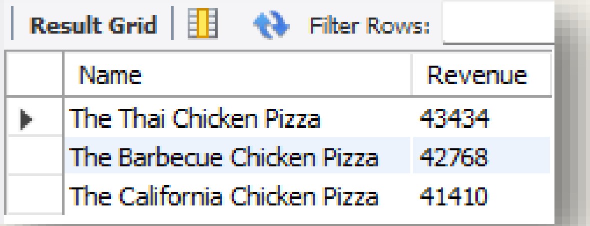
Determine the top 3 most ordered pizza types based on revenue.

Query :

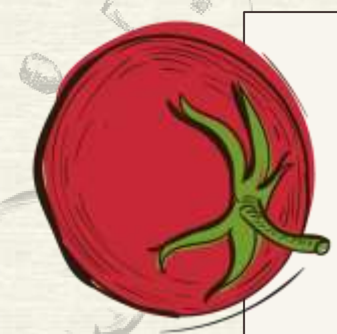


```
SELECT
    pizza_types.name AS Name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY Name
ORDER BY Revenue DESC
LIMIT 3;
```

Result :



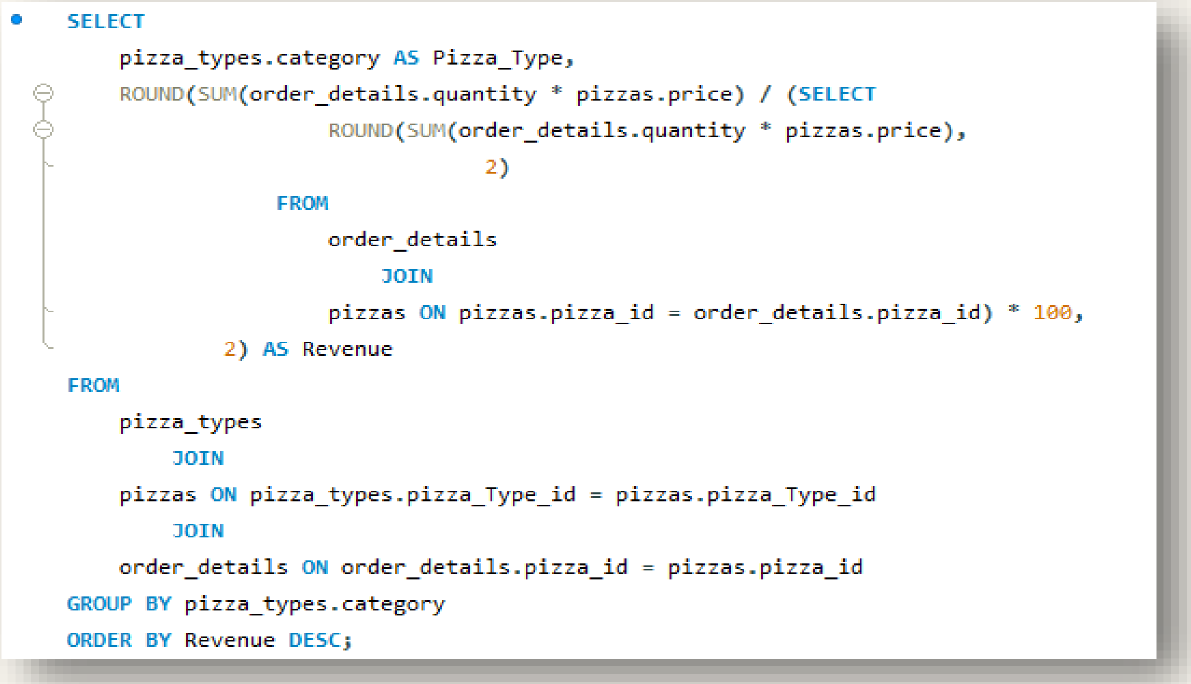
Name	Revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410



Question 11

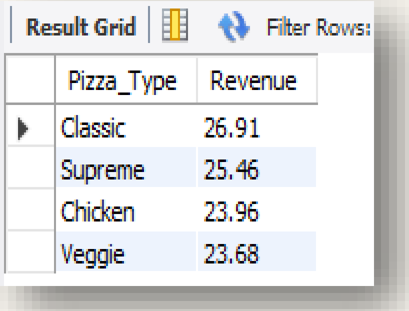
Calculate the percentage contribution of each pizza type to total revenue.

Query :



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

Result :

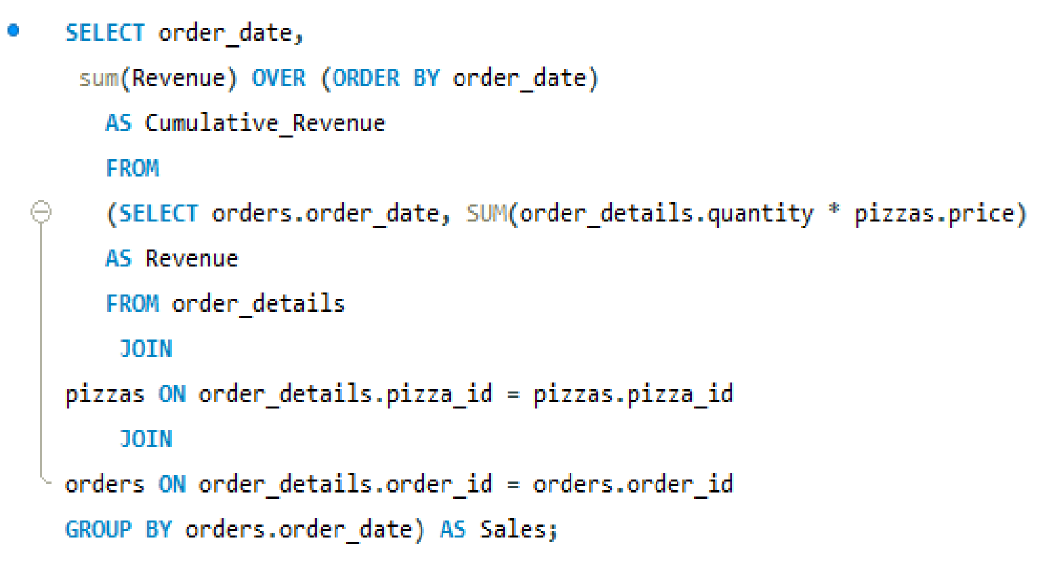


Pizza_Type	Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Question 12

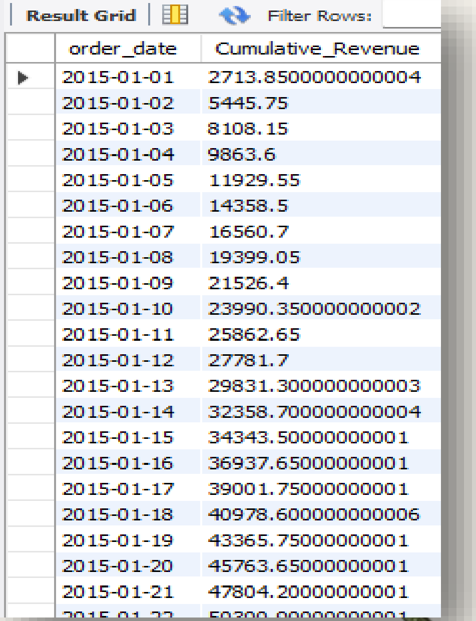
Analyze the cumulative revenue generated over time.

Query :



```
SELECT order_date,
 sum(Revenue) OVER (ORDER BY order_date)
   AS Cumulative_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 order_details.order_id = orders.order_id
GROUP BY orders.order_date) AS Sales;
```

Result :

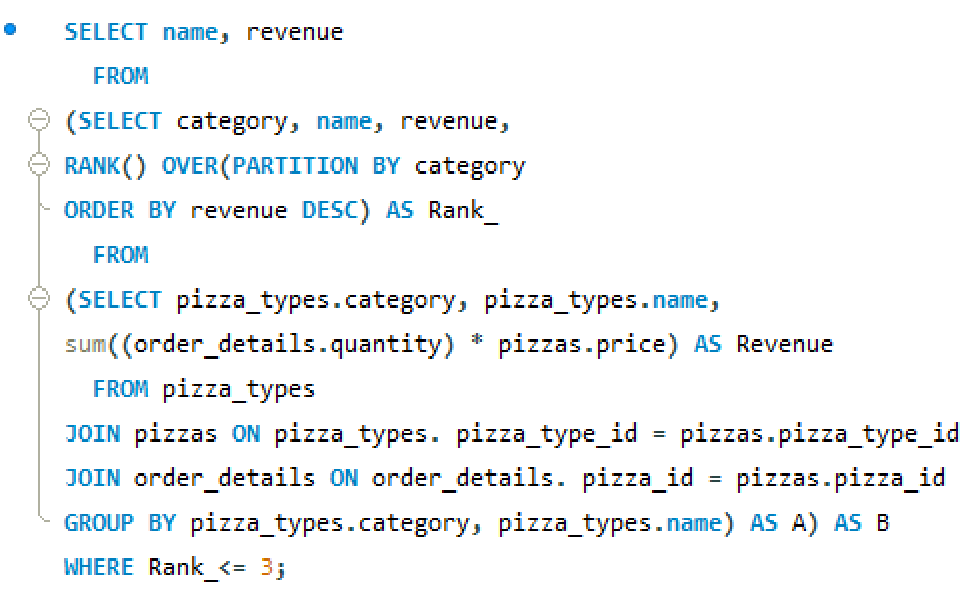


order_date	Cumulative_Revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
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	50300.00000000001

Question 13

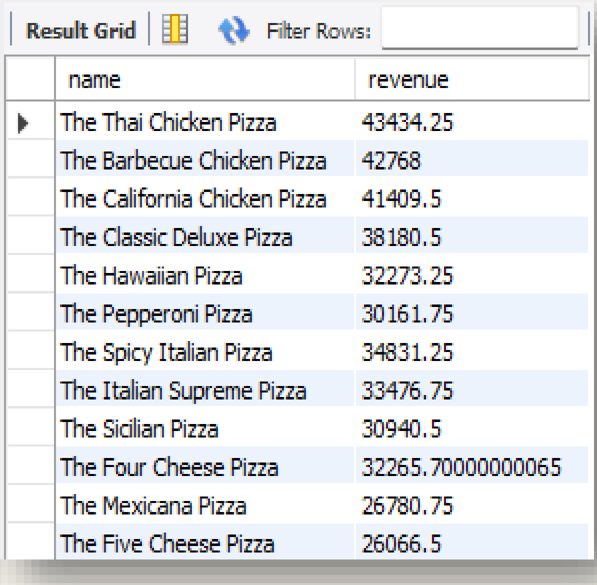
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query :



```
SELECT name, revenue
  FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category
ORDER BY revenue DESC) AS Rank_
  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 Rank_<= 3;
```

Result :



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
The Five Cheese Pizza	26066.5

THANK YOU

Do you have any questions/suggestions?

Vikasgowda3681@gmail.com

https://www.linkedin.com/in/vikas-g-r-6484541b1/