Pizza Sales Analysis Project

Presented by: Nitesh Verma

Date: 09/05/2024

Introduction to Pizza Sales Analysis

- ❑ The Pizza Sales Analysis project aims to analyse sales data from a pizza restaurant chain to gain insights into customer preferences, popular pizza types, and revenue trends.
- ❑ By leveraging SQL (Structured Query Language), we will explore the database of orders, order details, pizza types, and pricing information to uncover valuable insights that can inform business decisions and improve performance.
- ❑ This presentation will showcase our approach to data exploration, SQL queries used for analysis, key findings, and recommendations for optimizing sales and customer satisfaction.
- ❑ Through this analysis, we aim to provide actionable insights to the pizza restaurant chain's management team, enabling them to make informed decision to enhance profitability and customer experience.

Problem Statement

- ❑ The pizza restaurant chain is seeking to optimize sales performance and customer satisfaction by understanding key factors influencing pizza sales.
- ❑ Lack of insights into customer preferences, popular pizza type s, and revenue trends hinders the restaurant chain's ability to make informed business decisions.
- ❑ The management team requires a comprehensive analysis of sales data to identify opportunities for menu optimization, marketing strategies, and operational improvements.
- ❑ The objective of this project is to leverage SQL analysis techniques to   analyse pizza sales data and provide actionable insights to drive business growth and enhance customer experience.

Database Schema

- The database schema consists of four tables that capture essential information about pizza sales:

1. Orders:
   order_id (Primary Key)
   order_date
   order_time
2. Order_details:
   order_detail_id (Primary Key)
   order_id (Foreign Key referencing Orders)
   pizza_id (Foreign Key referencing Pizzas)
   quantity

Database Schema

3. Pizza_types:
    - pizza_type_id (Primary Key)
    - name
    - category
    - Ingredients
4. Pizzas:
    - pizza_id (Primary Key)
    - pizza_type_id (Foreign Key referencing Pizza_types)
    - size
    - price

- These tables are interconnected through foreign key relationships, allowing for efficient retrieval and analysis of pizza sales data.

Question to Solve

Basic:

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

Intermediate:

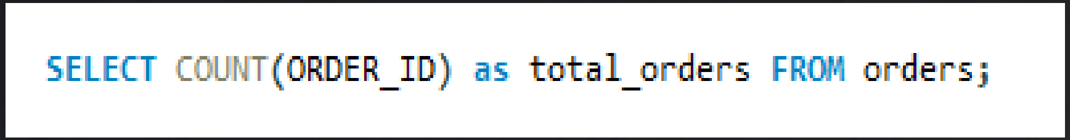
1. *Join the necessary tables to find the total quantity of each pizza category ordered.*
2. *Determine the distribution of orders by hour of the day.*
3. *Join relevant tables to find the category-wise distribution of pizzas.*
4. *Group the orders by date and calculate the average number of pizzas ordered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

Advanced:

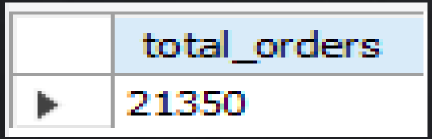
1. *Calculate the percentage contribution of each pizza type to total revenue.*
2. *Analyse the cumulative revenue generated over time.*
3. *Determine the top 3 most ordered pizza types based on revenue for each pizza category.*

BASIC QUESTIONS

1. Retrieve the total number of orders placed.

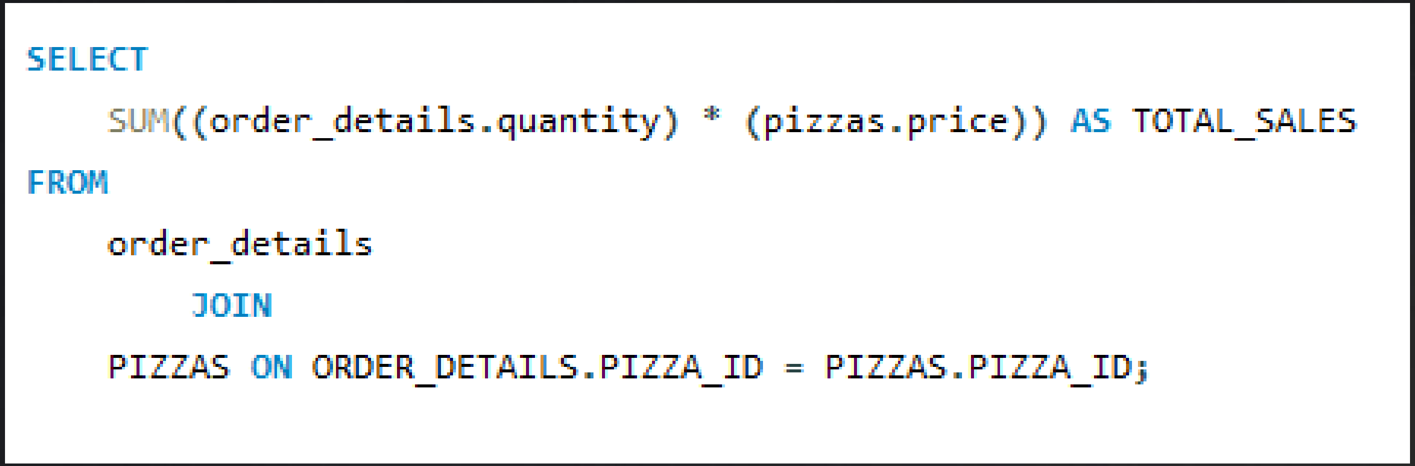


```sql
SELECT COUNT(ORDER_ID) as total_orders FROM orders;
```

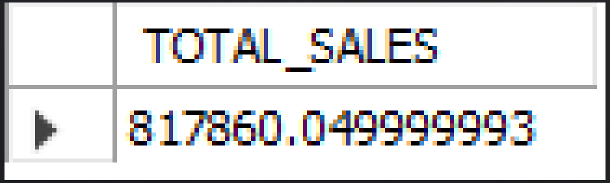


total_orders
21350

2. Calculate the total revenue generated from pizza sales.

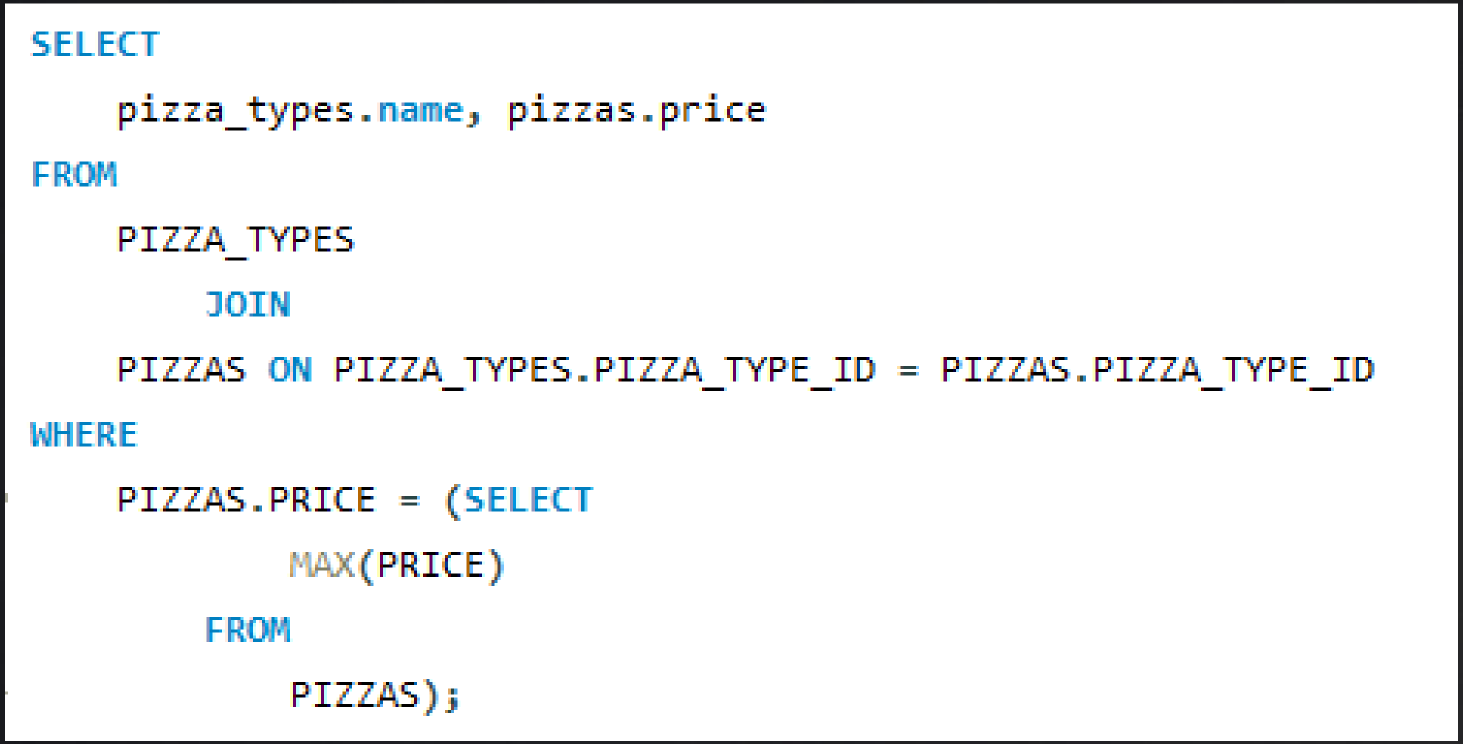


```
SELECT
    SUM((order_details.quantity) * (pizzas.price)) AS TOTAL_SALES
FROM
    order_details
        JOIN
    PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID;
```

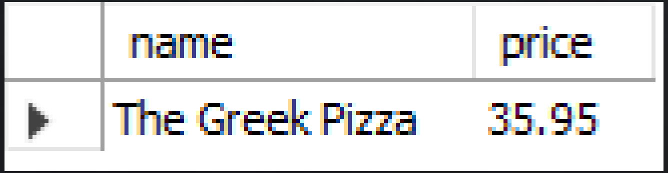


```
TOTAL_SALES
817860.049999993
```

3. Identify the highest-priced pizza.

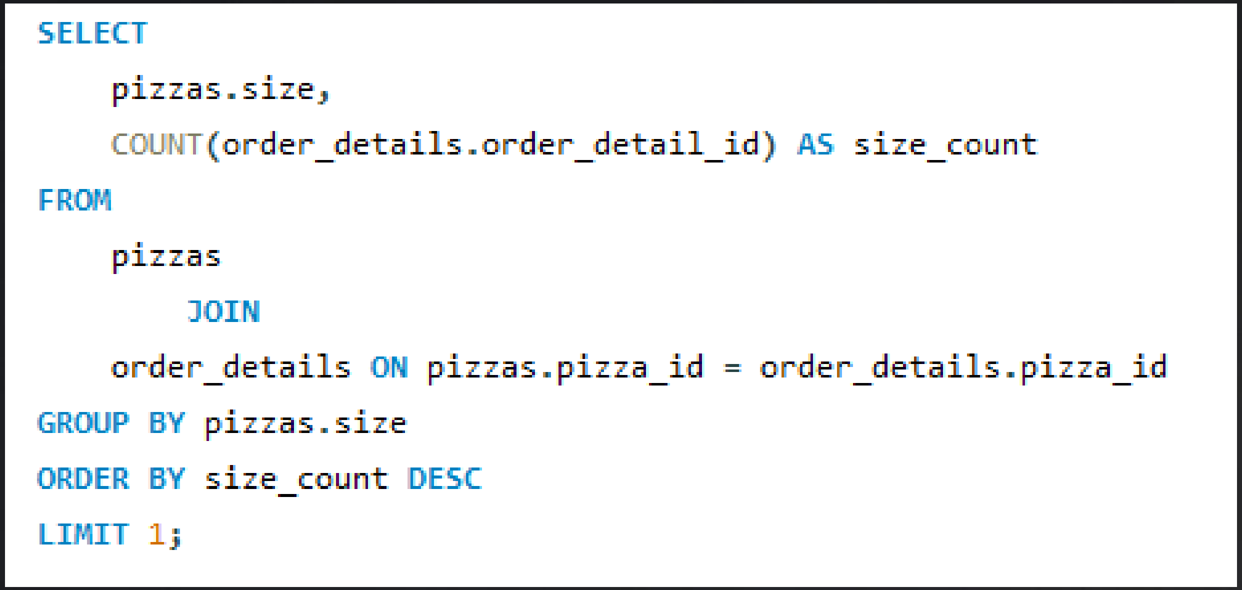


```
SELECT
    pizza_types.name, pizzas.price
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
WHERE
    PIZZAS.PRICE = (SELECT
            MAX(PRICE)
        FROM
            PIZZAS);
```

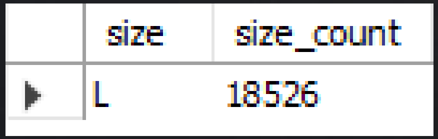


	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

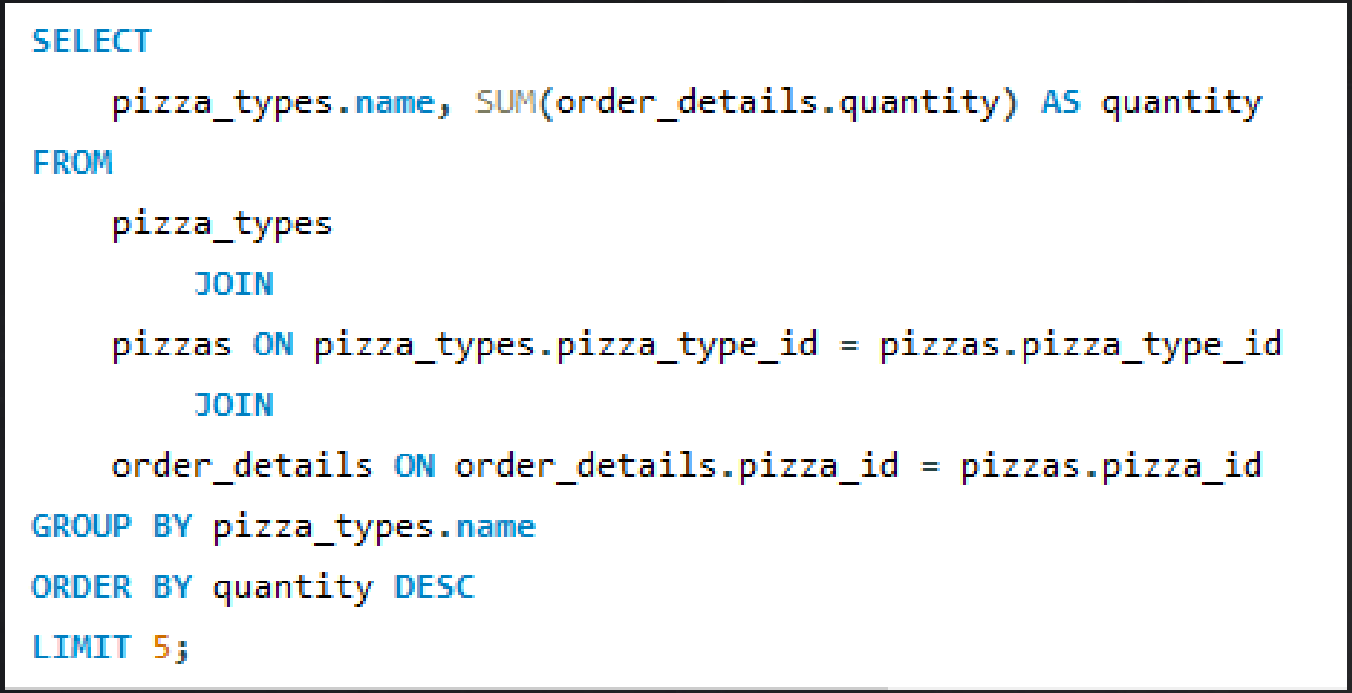


```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_detail_id) AS size_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY size_count DESC
LIMIT 1;
```

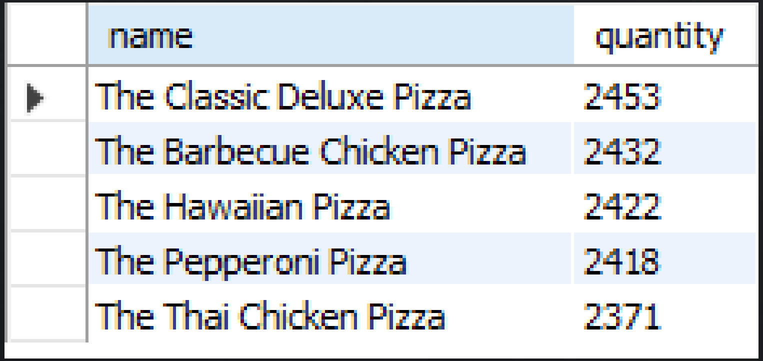


```
   size   size_count
▶  L      18526
```

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



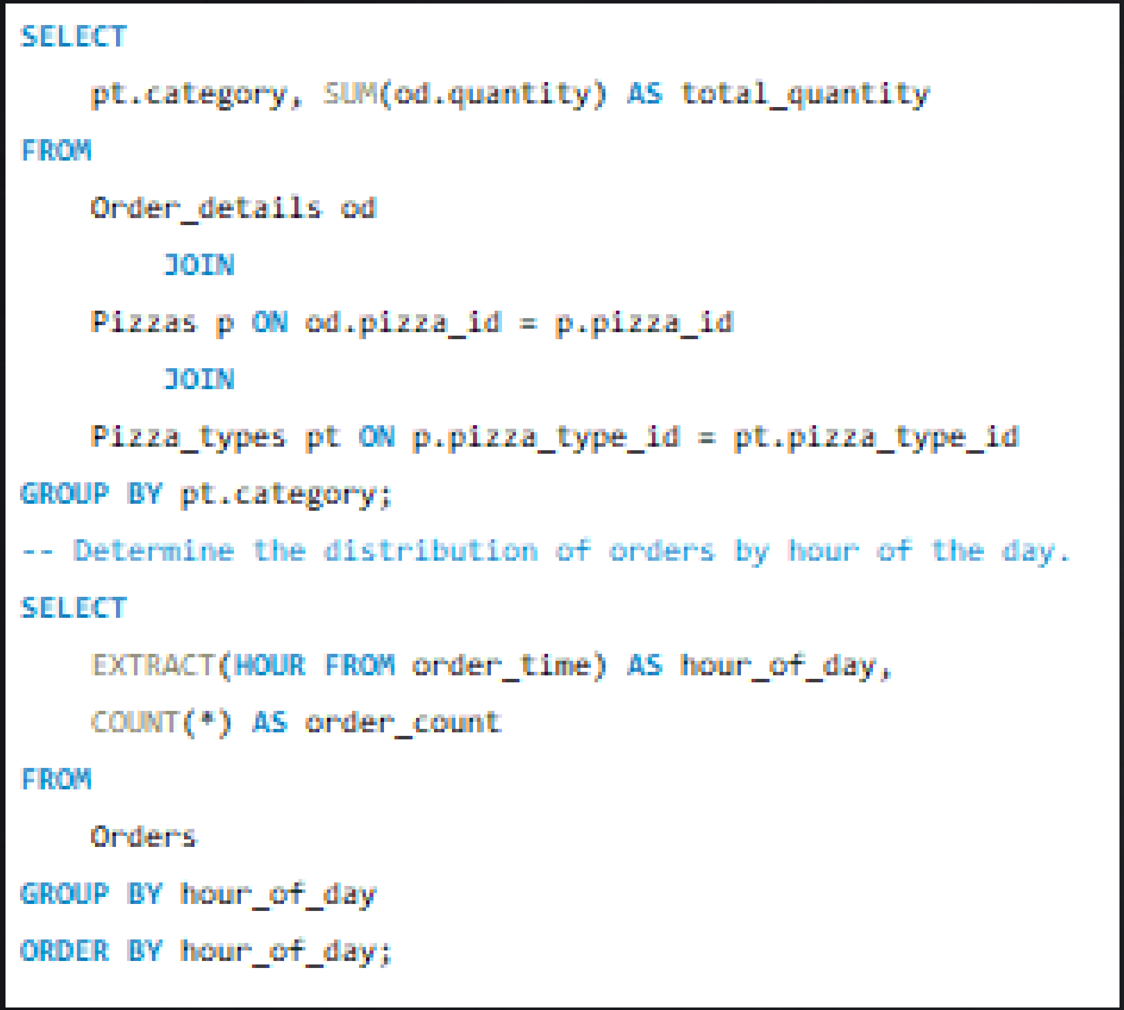
```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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



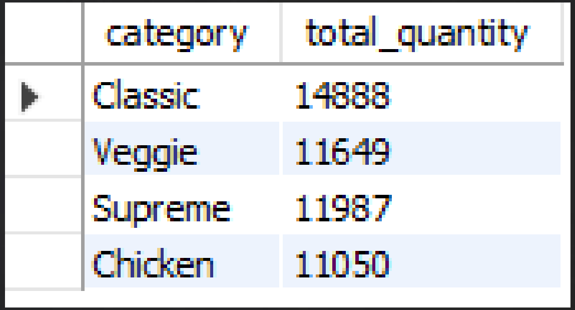
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

INTERMEDIATE QUESTIONS

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

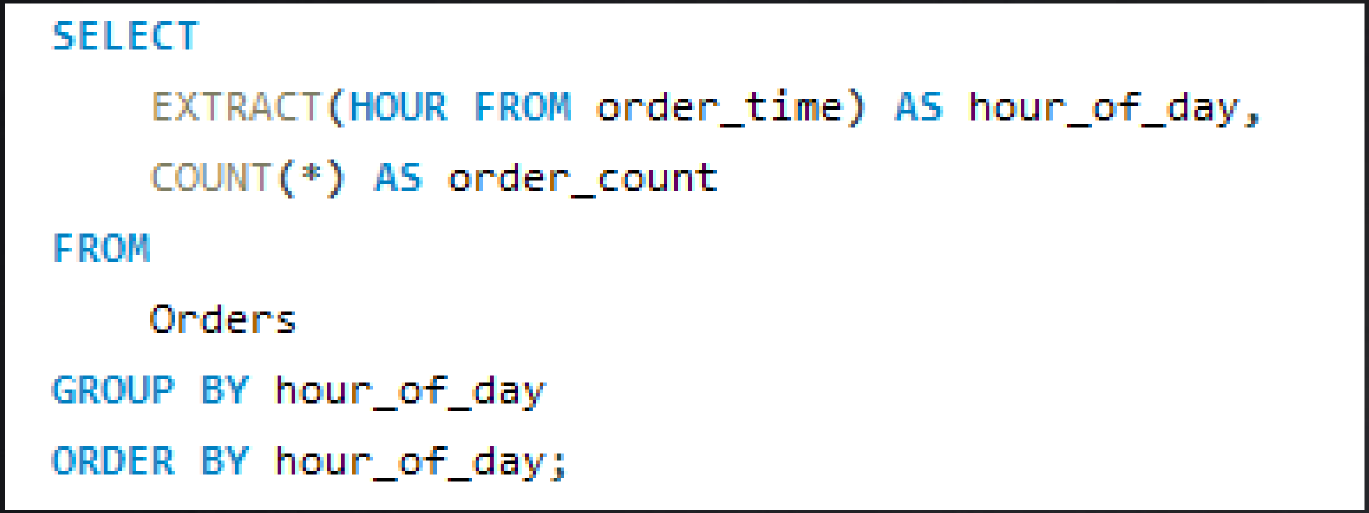


```sql
SELECT
    pt.category, SUM(od.quantity) AS total_quantity
FROM
    Order_details od
        JOIN
    Pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    Pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category;
-- Determine the distribution of orders by hour of the day.
SELECT
    EXTRACT(HOUR FROM order_time) AS hour_of_day,
    COUNT(*) AS order_count
FROM
    Orders
GROUP BY hour_of_day
ORDER BY hour_of_day;
```

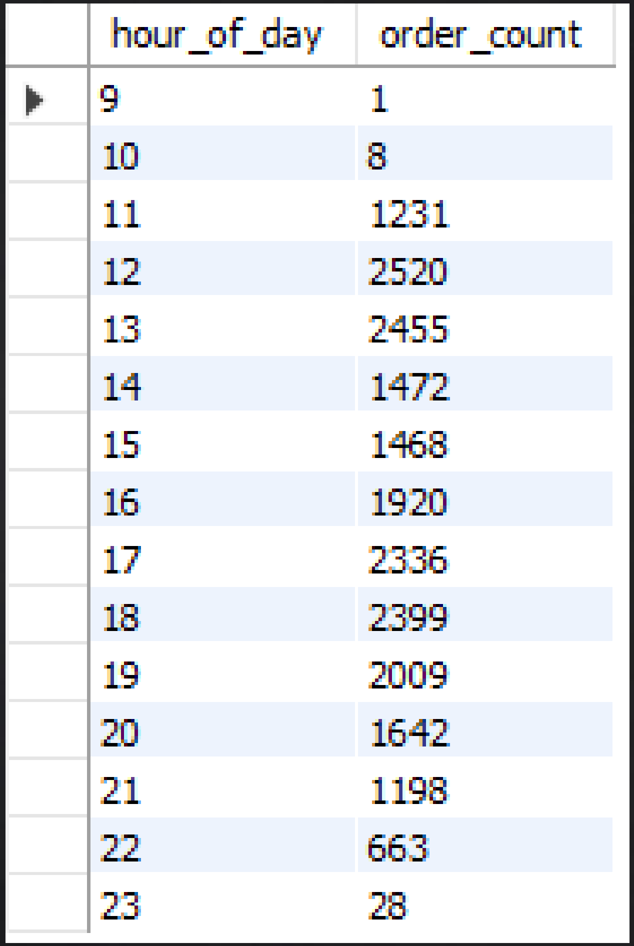


category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

2. Determine the distribution of orders by hour of the day.

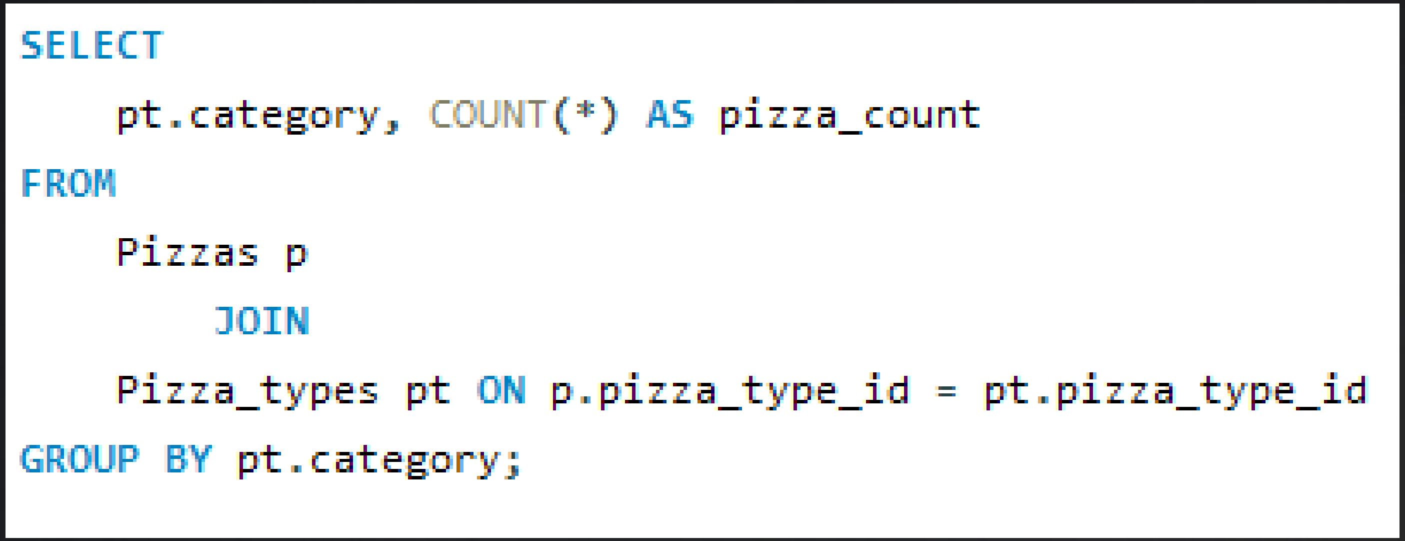


```sql
SELECT
    EXTRACT(HOUR FROM order_time) AS hour_of_day,
    COUNT(*) AS order_count
FROM
    Orders
GROUP BY hour_of_day
ORDER BY hour_of_day;
```

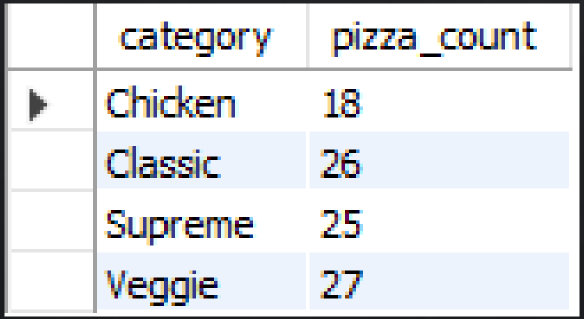


hour_of_day	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

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

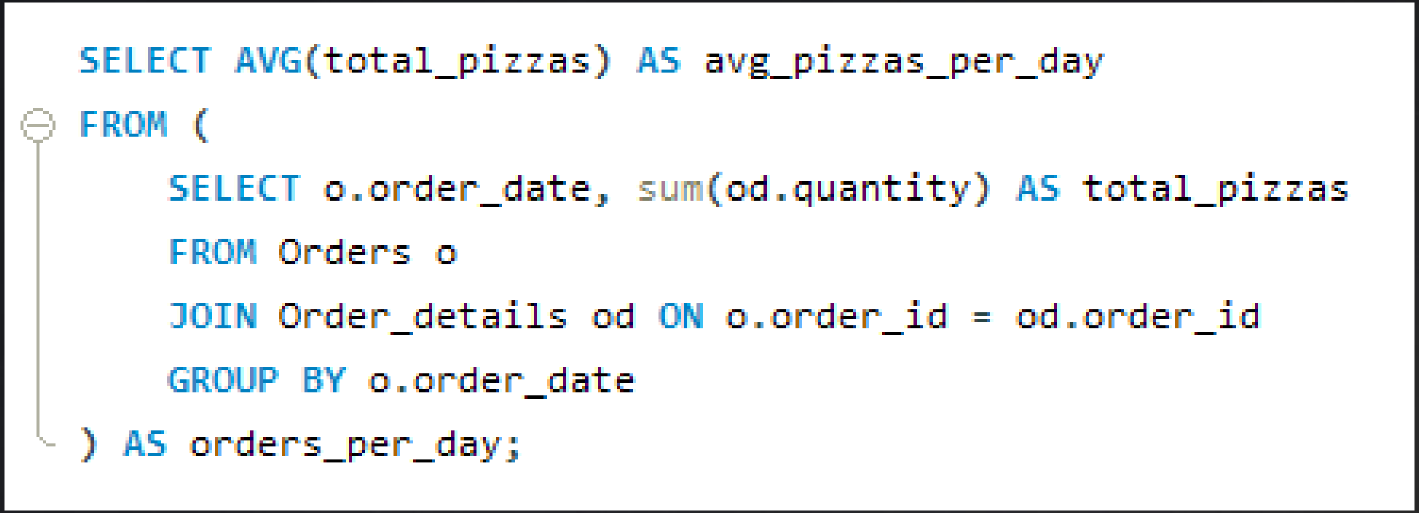


```
SELECT
    pt.category, COUNT(*) AS pizza_count
FROM
    Pizzas p
        JOIN
    Pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category;
```

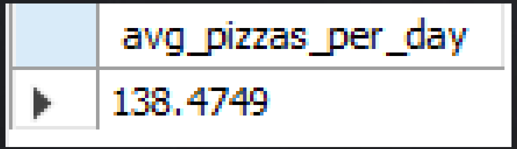


category	pizza_count
Chicken	18
Classic	26
Supreme	25
Veggie	27

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

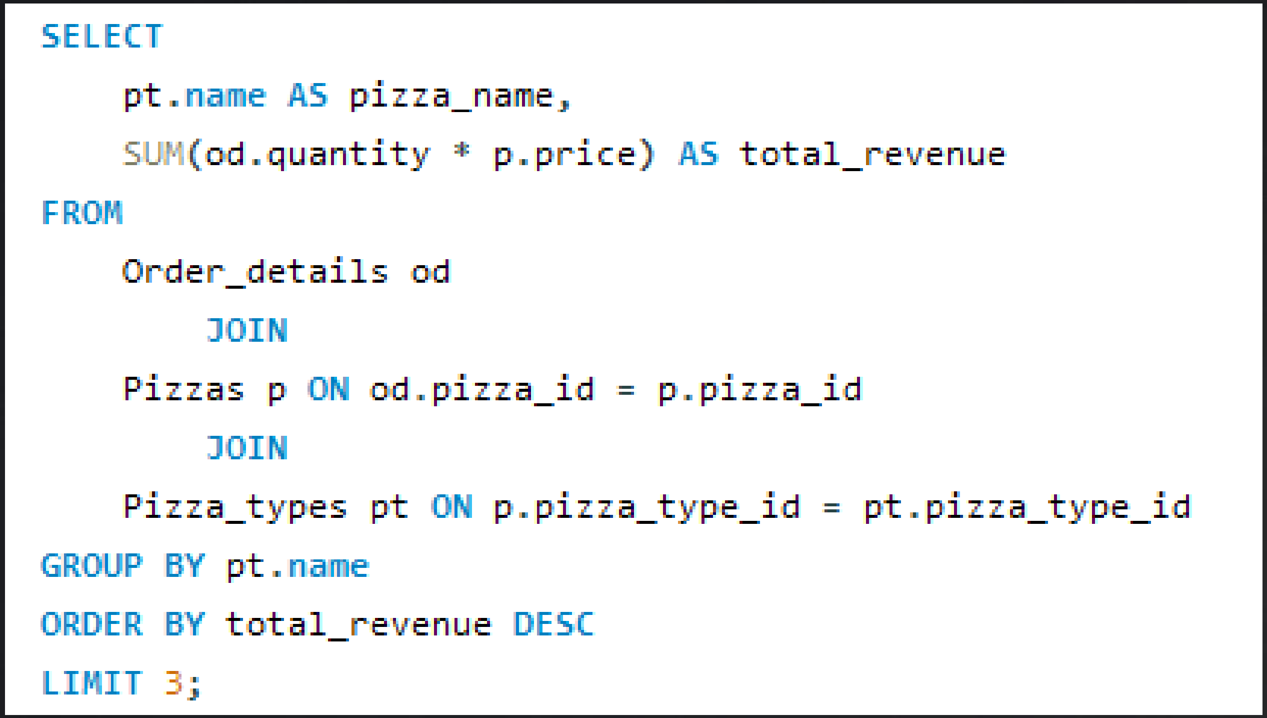


```
SELECT AVG(total_pizzas) AS avg_pizzas_per_day
FROM (
    SELECT o.order_date, sum(od.quantity) AS total_pizzas
    FROM Orders o
    JOIN Order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date
) AS orders_per_day;
```

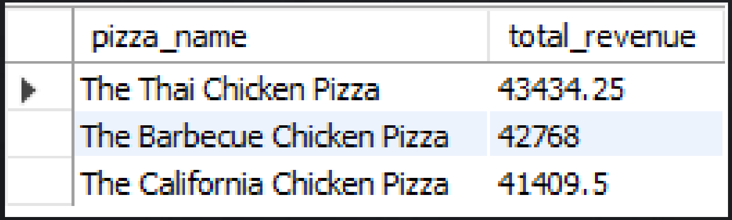


avg_pizzas_per_day
138.4749

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



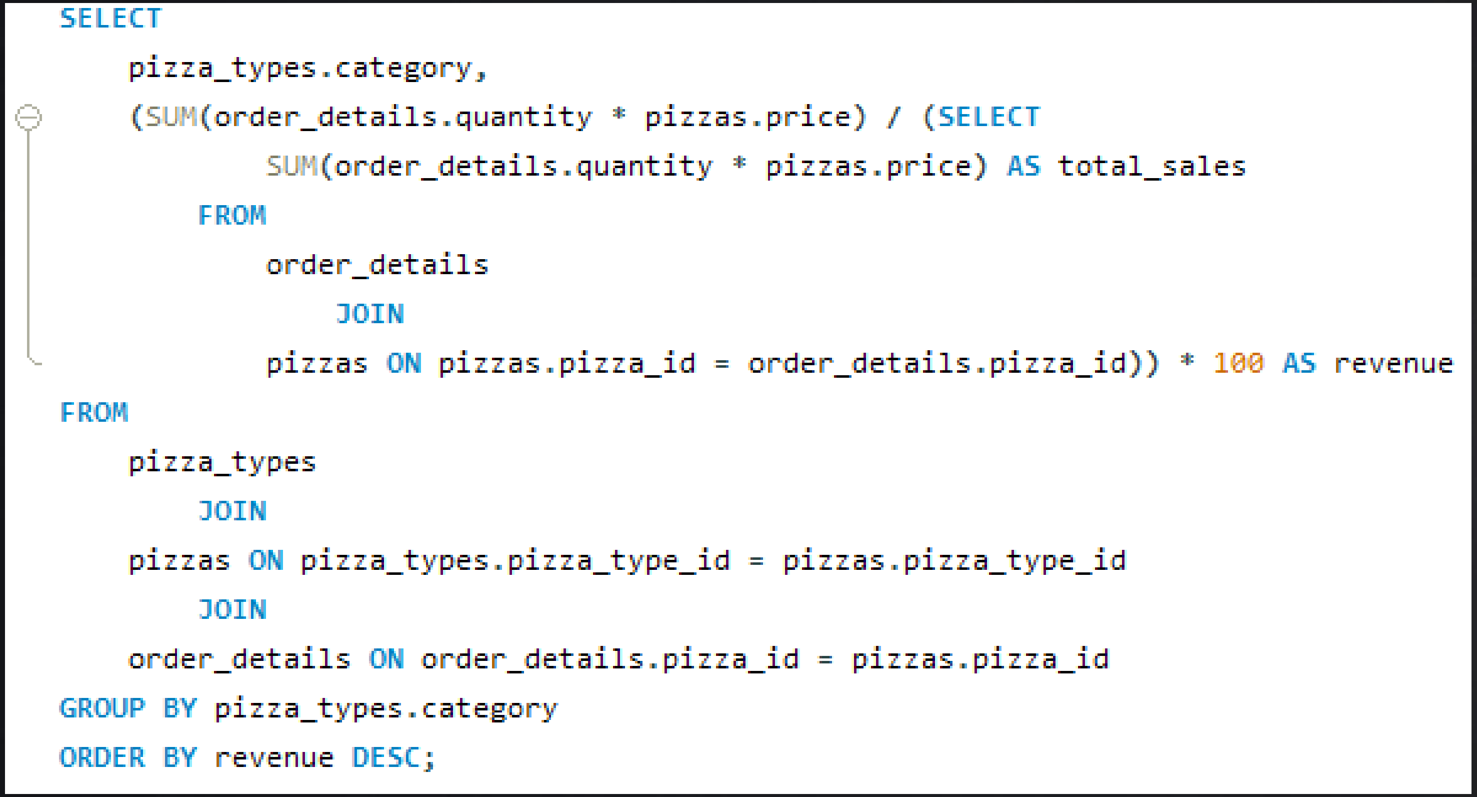
```
SELECT
    pt.name AS pizza_name,
    SUM(od.quantity * p.price) AS total_revenue
FROM
    Order_details od
        JOIN
    Pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    Pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3;
```



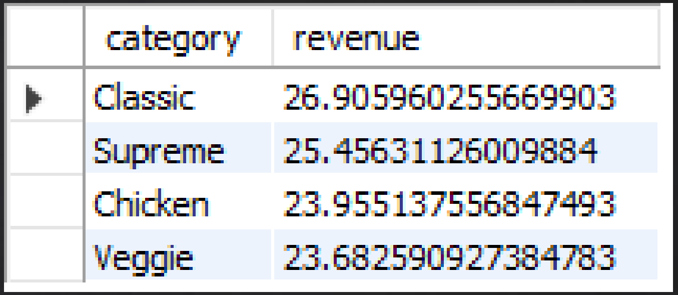
pizza_name	total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

ADVANCED QUESTIONS

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

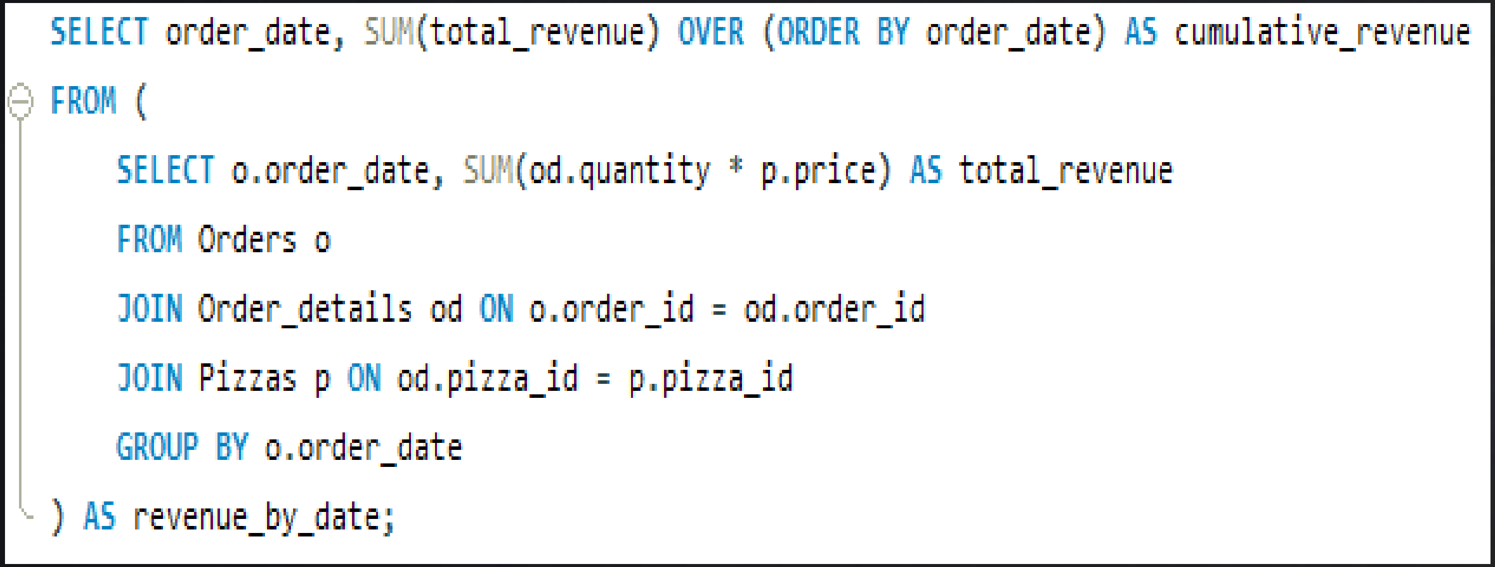


```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
            SUM(order_details.quantity * pizzas.price) AS total_sales
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 AS revenue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

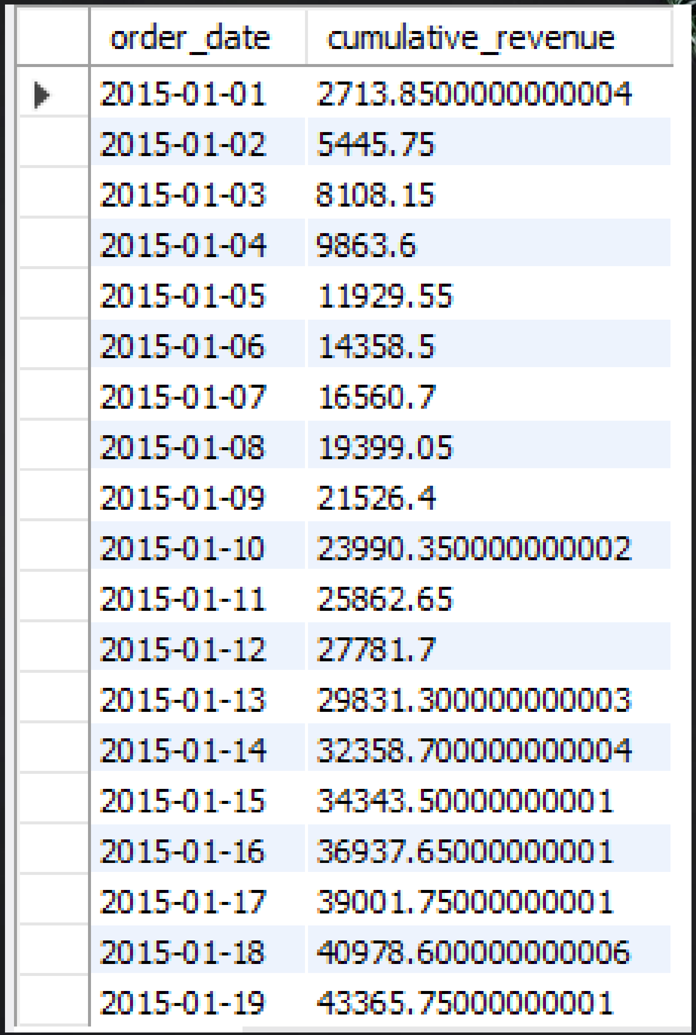


category	revenue
Classic	26.905960255669903
Supreme	25.45631126009884
Chicken	23.955137556847493
Veggie	23.682590927384783

2. Analyse the cumulative revenue generated over time.

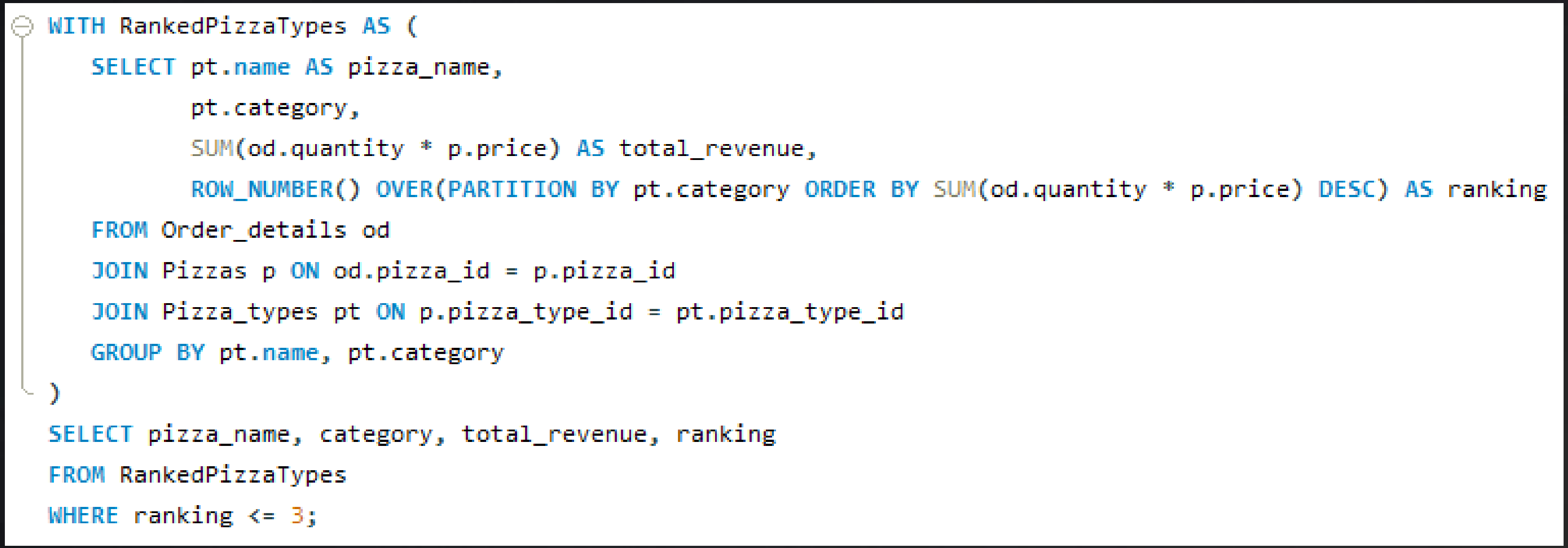


```sql
SELECT order_date, SUM(total_revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM (
    SELECT o.order_date, SUM(od.quantity * p.price) AS total_revenue
    FROM Orders o
    JOIN Order_details od ON o.order_id = od.order_id
    JOIN Pizzas p ON od.pizza_id = p.pizza_id
    GROUP BY o.order_date
) AS revenue_by_date;
```

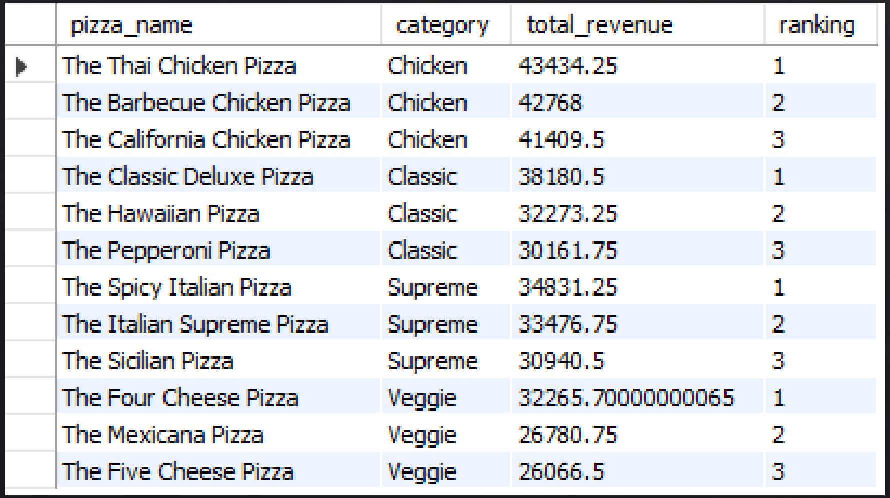


order_date	cumulative_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003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

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



```
WITH RankedPizzaTypes AS (
    SELECT pt.name AS pizza_name,
           pt.category,
           SUM(od.quantity * p.price) AS total_revenue,
           ROW_NUMBER() OVER(PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS ranking
    FROM Order_details od
    JOIN Pizzas p ON od.pizza_id = p.pizza_id
    JOIN Pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
    GROUP BY pt.name, pt.category
)
SELECT pizza_name, category, total_revenue, ranking
FROM RankedPizzaTypes
WHERE ranking <= 3;
```



pizza_name	category	total_revenue	ranking
The Thai Chicken Pizza	Chicken	43434.25	1
The Barbecue Chicken Pizza	Chicken	42768	2
The California Chicken Pizza	Chicken	41409.5	3
The Classic Deluxe Pizza	Classic	38180.5	1
The Hawaiian Pizza	Classic	32273.25	2
The Pepperoni Pizza	Classic	30161.75	3
The Spicy Italian Pizza	Supreme	34831.25	1
The Italian Supreme Pizza	Supreme	33476.75	2
The Sicilian Pizza	Supreme	30940.5	3
The Four Cheese Pizza	Veggie	32265.70000000065	1
The Mexicana Pizza	Veggie	26780.75	2
The Five Cheese Pizza	Veggie	26066.5	3

THANK YOU