# Pizza Sales Analysis Using SQL

### **Objective**

The goal of this project is to analyse pizza sales data using SQL to uncover trends, identify high-performing products, and understand customer purchasing patterns over time.

#### **Dataset Description**

- Source: Kaggle

- Tables:

- order\_details (order\_details\_id, order\_id, pizza\_id, quantity)
- orders (order\_id, date, time)
- pizza\_types (pizza\_type\_id, name, category, ingredients)
- pizzas ((pizza\_id, pizza\_type\_id, size, price)
- Tool Used: Microsoft SQL Server

## **SQL QUERIES**

--Retrieve the total number of orders placed.

select count(order\_id) as total\_orders from orders;



```
--Calculate the total revenue generated from pizza sales.
ISELECT
     ROUND(SUM(order details.quantity * pizzas.price), 2) AS total sales
FROM
    order details
JOIN
     pizzas ON pizzas.pizza id = order details.pizza id;
 total_sales
      817860.05
 1
-- Identify the highest-priced pizza.
SELECT TOP 1
    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;
price
     name
                  35.9500007629395
     The Greek Pizza
-- Identify the most common pizza size ordered.
SELECT TOP 1
    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;

    ⊞ Results

    Messages

      size
            order_count
            18526
 1
```

```
--List the top 5 most ordered pizza types along with their quantities.
    SELECT TOP 5
    pizza_types.name,
    SUM(order_details.quantity) AS quantity
    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;
name
                         quantity
1
   The Classic Deluxe Pizza
                         2453
2
    The Barbecue Chicken Pizza
                         2432
3
    The Hawaiian Pizza
                         2422
```

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

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity

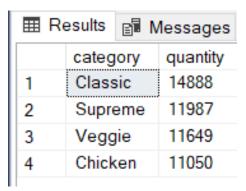
FROM
    pizza_types

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;
```



4

The Pepperoni Pizza

The Thai Chicken Pizza

2418

2371

```
--Determine the distribution of orders by hour of the day.

SELECT

DATEPART(HOUR, time) AS hour,

COUNT(order_id) AS order_count

FROM

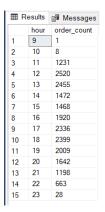
orders

GROUP BY

DATEPART(HOUR, time)

ORDER BY

hour;
```



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

SELECT category , COUNT(name) FROM pizza\_types
GROUP BY category;



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

```
SELECT
ROUND(AVG(quantity), 0) AS average_quantity

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;
```



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

```
SELECT TOP 3
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) 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;
```

Results			
	name	revenue	
1	The Thai Chicken Pizza	43434.25	
2	The Barbecue Chicken Pizza	42768	
3	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) * 100.0 /
         (SELECT SUM(order_details.quantity * pizzas.price)
          FROM order details
          JOIN pizzas ON pizzas.pizza id = order details.pizza id), 2) AS percentage of total sales
 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
     percentage_of_total_sales DESC;
 category
                percentage_of_total_sales
      Classic
                26.91
 2
       Supreme
                25.46
                23.96
 3
       Chicken
       Veggie
                23.68
```

--Analyze the cumulative revenue generated over time.

```
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
ORDER BY
date;
```

```
cum_revenue
    2015-01-01 2713.85000228882
     2015-01-02 5445.7500038147
2
     2015-01-03 8108.15000724792
3
     2015-01-04 9863.60000801086
4
5
     2015-01-05 11929.5500087738
6
     2015-01-06 14358.5000114441
7
     2015-01-07 16560.700012207
8
     2015-01-08 19399.0500183105
9
     2015-01-09 21526.4000225067
     2015-01-10 23990.350025177
10
```

120 70			
⊞ Results			
	name	revenue	
1	The Thai Chicken Pizza	43434.25	
2	The Barbecue Chicken Pizza	42768	
3	The California Chicken Pizza	41409.5	
4	The Classic Deluxe Pizza	38180.5	
5	The Hawaiian Pizza	32273.25	
6	The Pepperoni Pizza	30161.75	
7	The Spicy Italian Pizza	34831.25	
8	The Italian Supreme Pizza	33476.75	
9	The Sicilian Pizza	30940.5	
10	The Four Cheese Pizza	32265.7010040283	
11	The Mexicana Pizza	26780.75	
12	The Five Cheese Pizza	26066.5	

### Insight:

- Total revenue generated is \$817,860.05
- Classic and Chicken categories generate the most revenue.
- Steady revenue growth, with spikes during weekends and holidays.
- The top-selling pizzas are Classic Deluxe, BBQ Chicken, and Five Cheese.

## **Key Learnings:**

Learned to use joins, group by, window functions, and aggregation in SQL.

- Understood how to derive real-world business insights from raw sales data.
- Practiced working with a relational schema and writing optimized queries.

### **Conclusion:**

This analysis helped identify the best-performing pizzas and revenue patterns. The business can focus on high-performing categories and target weekend promotions based on trends.