# Pizza Sales Analysis using SQL

A Comprehensive Data Analysis Project

Made by
Ritesh Tatikonda

# Introduction

This project analyzes pizza sales data to uncover trends, optimize product offerings, and improve sales performance. Using SQL, we can explore customer preferences, sales patterns, and revenue generation to provide actionable insights for business growth.

## Objectives

- Retrieve key metrics like total orders and revenue.
- Identify top-performing pizzas and categories.
- Analyze order trends by time and day.
- Calculate revenue distribution across different pizza types.

## Tools used

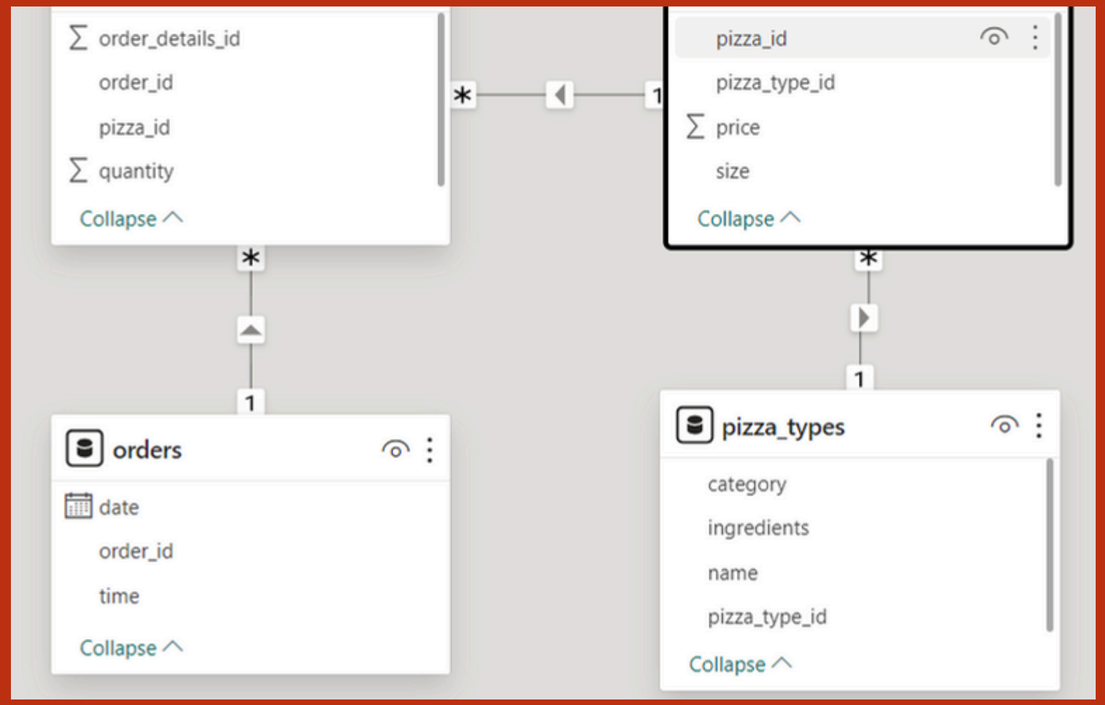
**SQL: For querying and analyzing data**

**Database: MySql**

# Data model overview

The pizza sales data is structured across four main tables: Orders, Order_Details, Pizzas, and Pizza_Types.

These tables are linked through key relationships, enabling a comprehensive analysis of pizza sales and customer orders.




# 1)Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(orderid) AS total_orders
FROM
    orders;
```

Result Grid

| | total_orders |
|---|---|
| ▶ | 21350 |

# 2)Calculate the total revenue  generated from pizza sales

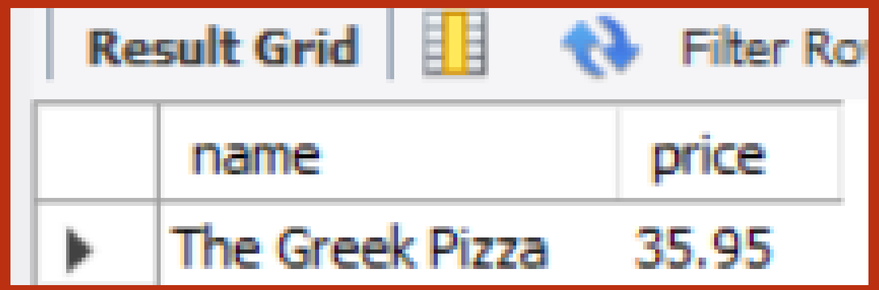
```sql
SELECT
    ROUND(SUM((quantity * price)), 2) AS Total_revenue
FROM
    orderdetails
        JOIN
    pizzas ON orderdetails.pizzaid = pizzas.pizza_id;
```

Result Grid

| | Total_revenue |
|---|---|
| ▶ | 817860.05 |

# 3)Identify the highest-priced pizza

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



# 4)Identify the most common pizza size ordered

```sql
SELECT
    size, COUNT(orderdetailsid) AS Total_quantity
FROM
    pizzas
        JOIN
    orderdetails ON pizzas.pizza_id = orderdetails.pizzaid
GROUP BY size
ORDER BY total_quantity DESC;
```

Result Grid | Filter

| size | Total_quantity |
|---|---|
| L | 18526 |
| M | 15385 |
| S | 14137 |
| XL | 544 |
| XXL | 28 |

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

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

Result Grid | Filter Rows:

| 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 |

# 6) Join the necessary tables to find the total quantity of each pizza category ordered

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

Result Grid | Filter

| category | Quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# 7)Determine the distribution of orders by hour of the day

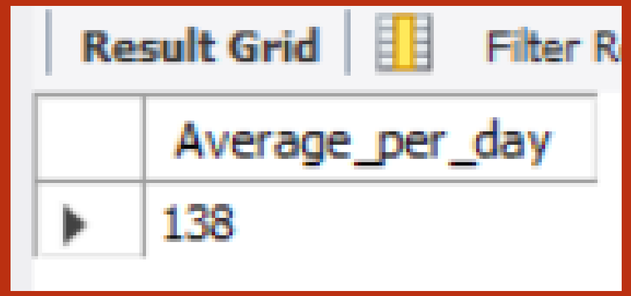
```sql
SELECT
    HOUR(ordertime) AS Hour, COUNT(orderid) AS Order_count
FROM
    orders
GROUP BY HOUR(ordertime)
ORDER BY HOUR(ordertime);
```

Result Grid

| 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 |

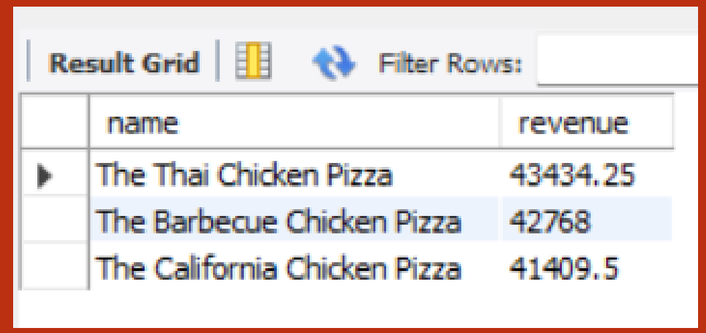
# 8)Group the orders by date and calculate the average number of pizzas ordered per day

```
with x as
    (select orderdate,sum(quantity) as Count
from orders join orderdetails on
    orders.orderid=orderdetails.orderid
group by orderdate)
    select round(avg(count),0) as Average_per_day from x;
```



# 9)Determine the top 3 most ordered pizza types based on revenue

```sql
SELECT
    name, SUM(orderdetails.quantity*pizzas.price) AS revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orderdetails ON pizzas.pizza_id = orderdetails.pizzaid
GROUP BY name
ORDER BY revenue DESC limit 3;
```

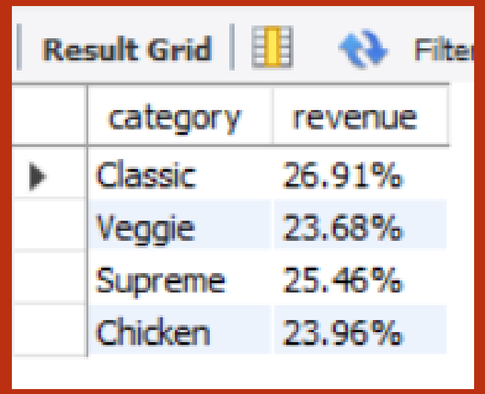


Result Grid | Filter Rows:

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

# 10)Calculate the percentage contribution of each pizza type to total revenue

```sql
select category,concat(round((SUM(orderdetails.quantity*pizzas.price) / (SELECT
    ROUND(SUM((quantity * price)), 2) AS Total_revenue
FROM
    orderdetails
        JOIN
    pizzas ON orderdetails.pizzaid = pizzas.pizza_id)) *100,2),"%") as revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orderdetails ON pizzas.pizza_id = orderdetails.pizzaid
    group by category
```




| category | revenue |
|---|---|
| Classic | 26.91% |
| Veggie | 23.68% |
| Supreme | 25.46% |
| Chicken | 23.96% |

# 11)Analyze the cumulative revenue generated over time

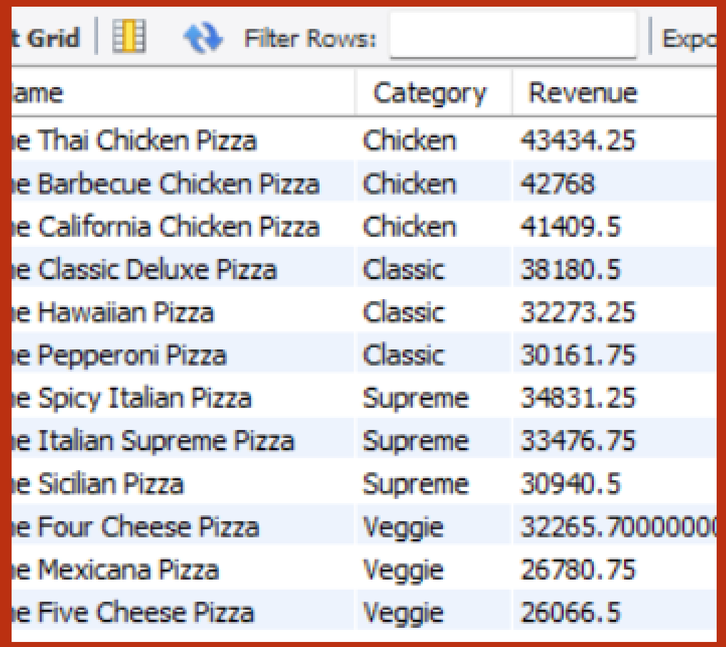
```sql
with cte as (SELECT
    orderdate, SUM(quantity * price) AS Revenue
FROM
    orders
        JOIN
    orderdetails ON orders.orderid = orderdetails.orderid
        JOIN
    pizzas ON orderdetails.pizzaid = pizzas.pizza_id
GROUP BY orderdate)
select orderdate,sum(revenue) over(order by orderdate) as Cumulative_revenue
from cte;
```

Result Grid | Filter Rows:

| orderdate | 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 |

# 12)Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
 select Name,Category,Revenue from
 (select category,name,revenue,
 dense_rank() over(partition by category order by revenue desc) as rnk from
(SELECT
    name,category,sum(quantity*price) as revenue
    FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orderdetails ON pizzas.pizza_id = orderdetails.pizzaid
    group by name,category) as a) as b
    where rnk<=3;
```



t Grid | Filter Rows: | Expo

| ame | Category | Revenue |
|---|---|---|
| e Thai Chicken Pizza | Chicken | 43434.25 |
| e Barbecue Chicken Pizza | Chicken | 42768 |
| e California Chicken Pizza | Chicken | 41409.5 |
| e Classic Deluxe Pizza | Classic | 38180.5 |
| e Hawaiian Pizza | Classic | 32273.25 |
| e Pepperoni Pizza | Classic | 30161.75 |
| e Spicy Italian Pizza | Supreme | 34831.25 |
| e Italian Supreme Pizza | Supreme | 33476.75 |
| e Sicilian Pizza | Supreme | 30940.5 |
| e Four Cheese Pizza | Veggie | 32265.70000000 |
| e Mexicana Pizza | Veggie | 26780.75 |
| e Five Cheese Pizza | Veggie | 26066.5 |

# Conclusion

1)Total Orders & Revenue: We recorded a total of 21350 orders, generating a total revenue of Rs 817860. The most common order times are around noon and early evening, indicating peak meal hours.

2)Top-Selling Pizza Type: The Classic Deluxe pizza is the most ordered, contributing 20.13% to total sales.

3)Highest-Priced Pizza: The Greek pizza is the highest-priced pizza at Rs 39.95 and, although not the most ordered, it still contributes significantly to revenue due to its higher price point.

4)Pizza Sizes: The Large pizza is the most popular size, accounting for 38.1% of orders, suggesting a preference for family-size or group orders.

5)Category-wise Distribution: Classic pizzas dominate the orders, with chicken options being less frequent but still showing potential for targeted marketing.

# Recommendations

1)Promote Popular Pizzas: Since Classic Deluxe and Barbecue chicken pizzas are top sellers, promotional deals like "Buy 1, Get 1" or discounts during peak hours could further increase sales.

2)Target Family/Group Orders: Given the preference for Large pizzas, creating bundles with drinks and sides for families or group gatherings could encourage higher order volumes.

3)Optimize Around Peak Hours: With most orders coming in around noon and early evening, special promotions or happy-hour deals during these times could boost revenue even further

4)Focus on High-Priced Pizzas: The Deluxe Supreme pizza, despite being higher priced, still shows strong performance. Promoting premium offerings could attract customers willing to spend more.

5)Streamline Operations: Analyze order fulfillment times during peak periods to ensure that the kitchen can handle the high volume, avoiding delays and ensuring customer satisfaction.

# Thank you