# SQL PROJECT ON PIZZA SALES

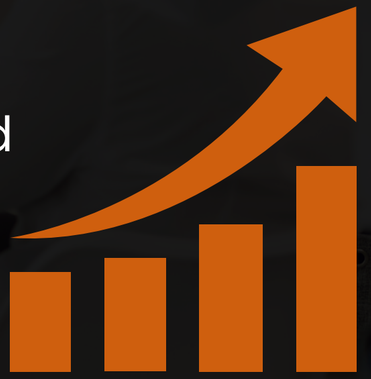
presented by
Mahesh Roge

# INTRODUCTION TO PIZZA SALES DATA ANALYSIS

In this project, we analyzed a comprehensive dataset from a fictional pizza restaurant to uncover valuable business insights. Using MySQL, we explored and processed data from four interconnected tables: order_details, orders, pizzas, and pizza_types. The goal was to extract meaningful patterns and performance metrics that could assist in decision-making for operational efficiency, menu optimization, and revenue growth.

Key focus areas included:

1. Sales Performance: Quantifying total orders, revenue generation, and identifying high-performing pizza types and sizes.
2. Consumer Preferences: Determining the most common pizza sizes and identifying popular categories based on order trends.
3. Revenue Analysis: Evaluating top-performing pizzas by revenue contribution and analyzing cumulative earnings over time.
4. Operational Insights: Examining order distribution by time of day and grouping orders by date for performance benchmarking.



# QUESTIONS

- 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.
- Find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of daily pizzas.
- 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.

# 1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
-- Retrieve the total number of orders placed.


SELECT
    COUNT(order_id)
FROM
    orders;
```

**OUTPUT**

| | COUNT(order_id) |
|---|---|
| ▶ | 21350 |

# 2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

## OUTPUT

| total_revenue |
| --- |
| 817860.05 |

# 3. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
3
4 •     SELECT
5           pizza_types.name, pizzas.price
6       FROM
7           pizza_types JOIN pizzas
8           ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9       ORDER BY pizzas.price DESC
10      LIMIT 1;
```

**OUTPUT**

| name | price |
|---|---|
| The Greek Pizza | 35.95 |

# 4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

## OUTPUT

| size | order_count |
|---|---|
| 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(order_details.quantity) AS sum_order_quantity
FROM
    pizza_types JOIN pizzas
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details
    ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY sum_order_quantity DESC
LIMIT 5;
```

## OUTPUT

| name | sum_order_q |
|---|---|
| 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
    pizza_types.category,
    SUM(order_details.quantity) AS Total_category
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 Total_category DESC;
```

## OUTPUT

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

# 7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

OUTPUT

| | hours | order_count |
|---|---|---|
| | 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 |

# 8. FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
3
4 •     SELECT
5           pizza_types.category, COUNT(name)
6       FROM
7           pizza_types
8       GROUP BY category;
```

## OUTPUT

| category | count(name) |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

# 9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF DAILY PIZZAS.

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

**OUTPUT**

| avg_pizza_ordered |
| --- |
| 138 |

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

```sql
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;
```

## OUTPUT

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

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

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT SUM(order_details.quantity * pizzas.price)
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;
```

## OUTPUT

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

# 12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
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 orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS sales;
```

**OUTPUT**

| order_date | cumulative_revenue |
|---|---|
| 2015-01-01 | 2713.850000000000 |
| 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 |

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

```sql
SELECT
    name, revenue, category
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 ;
```

## OUTPUT

| name | revenue | category |
|---|---|---|
| The Thai Chicken Pizza | 43434.25 | Chicken |
| The Barbecue Chicken Pizza | 42768 | Chicken |
| The California Chicken Pizza | 41409.5 | Chicken |
| The Classic Deluxe Pizza | 38180.5 | Classic |
| The Hawaiian Pizza | 32273.25 | Classic |
| The Pepperoni Pizza | 30161.75 | Classic |
| The Spicy Italian Pizza | 34831.25 | Supreme |

# THANK YOU