



# PIZZA SALES

Analyzing sales performance and customer preferences using SQL

**BY - NISHANTH NALANAGULA**

# PROJECT OVERVIEW

This project focuses on analyzing pizza sales data using MySQL to understand overall business performance.

The analysis helps identify revenue trends, popular pizza types, customer ordering patterns, and peak sales periods.

By querying transactional data, meaningful insights were derived to support data-driven business decisions.



# DATASET DESCRIPTION

## DATASET SOURCE

Pizza Sales Dataset ( Kaggle )

## TABLES USED

- orders
- order\_details
- pizzas
- pizza\_types

## DATASET DETAILS

- Total records: 48,000+
- Time period: One year of sales data
- Data type: Transactional sales data



# BUSINESS OBJECTIVES

## Basic:

- 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.

## Intermediate:

- 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.

## Advanced:

- 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

```
SELECT  
    COUNT(*) AS total_orders  
FROM  
    orders;
```

Result Grid	
	total_orders
▶	21350

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

```
SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS total_sales  
FROM  
    orders_details od  
        JOIN  
    pizzas p ON od.pizza_id = p.pizza_id;
```

Result Grid	
	total_sales
▶	817860.05

### 3. IDENTIFY THE HIGHEST-PRICED PIZZA

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

Result Grid | Filter

	name	price
▶	The Greek Pizza	35.95

## 4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT
    p.size, COUNT(od.order_details_id) AS freq
FROM
    orders_details od
    JOIN
    pizzas p ON p.pizza_id = od.pizza_id
GROUP BY size
ORDER BY freq DESC;
```

Result Grid

	size	freq
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

# 5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT
    pt.name, SUM(od.quantity) AS total_quantity
FROM
    orders_details od
    JOIN
    pizzas p ON p.pizza_id = od.pizza_id
    JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

	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

```
SELECT pt.category, SUM(od.quantity) AS quantity
FROM orders_details od
    JOIN pizzas p ON p.pizza_id = od.pizza_id
    JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
ORDER BY od.quantity DESC;
```

Result Grid		
	category	quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

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

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

Result Grid | Filter

	hour	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. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
SELECT  
    category, COUNT(name) AS pizza_count  
FROM  
    pizza_types  
GROUP BY category;
```

	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# 9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```
WITH order_quantity AS (
  SELECT
    o.order_date, SUM(quantity) AS pizza_count
  FROM
    orders o
    JOIN
    orders_details od ON od.order_id = o.order_id
  GROUP BY order_date
)
SELECT
  ROUND(AVG(pizza_count), 0) AS avg_pizzas_ordered_per_day
FROM
  order_quantity;
```

Result Grid		Filter Rows:
	avg_pizzas_ordered_per_day	
▶	138	

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

```
SELECT
    pt.name, SUM(price * quantity) AS revenue
FROM
    orders_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 revenue DESC
LIMIT 3;
```

	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

```
SELECT
    pt.category,
    ROUND((SUM(price * quantity) / (SELECT
        ROUND(SUM(od.quantity * p.price), 2) AS total_sales
    FROM
        orders_details od
        JOIN
            pizzas p ON od.pizza_id = p.pizza_id)) * 100, 2) AS revenue_pct
FROM
    orders_details od
    JOIN
        pizzas p ON od.pizza_id = p.pizza_id
    JOIN
        pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY revenue_pct DESC;
```

Result Grid		
	category	revenue_pct
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# 12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
WITH sales AS (
  SELECT
    o.order_date, SUM(p.price * od.quantity) AS revenue
  FROM
    orders o
    JOIN
    orders_details od ON od.order_id = o.order_id
    JOIN
    pizzas p ON p.pizza_id = od.pizza_id
  GROUP BY o.order_date
)
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
FROM sales;
```

	order_date	cum_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

NOTE : The Output goes on upto 2015-12-31

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

```
WITH pizza_category_revenue AS (

    WITH sales AS (
        SELECT
            pt.category, pt.name, SUM(od.quantity * p.price) AS revenue
        FROM
            orders_details od
                JOIN
            pizzas p ON p.pizza_id = od.pizza_id
                JOIN
            pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
        GROUP BY pt.category , pt.name
    )

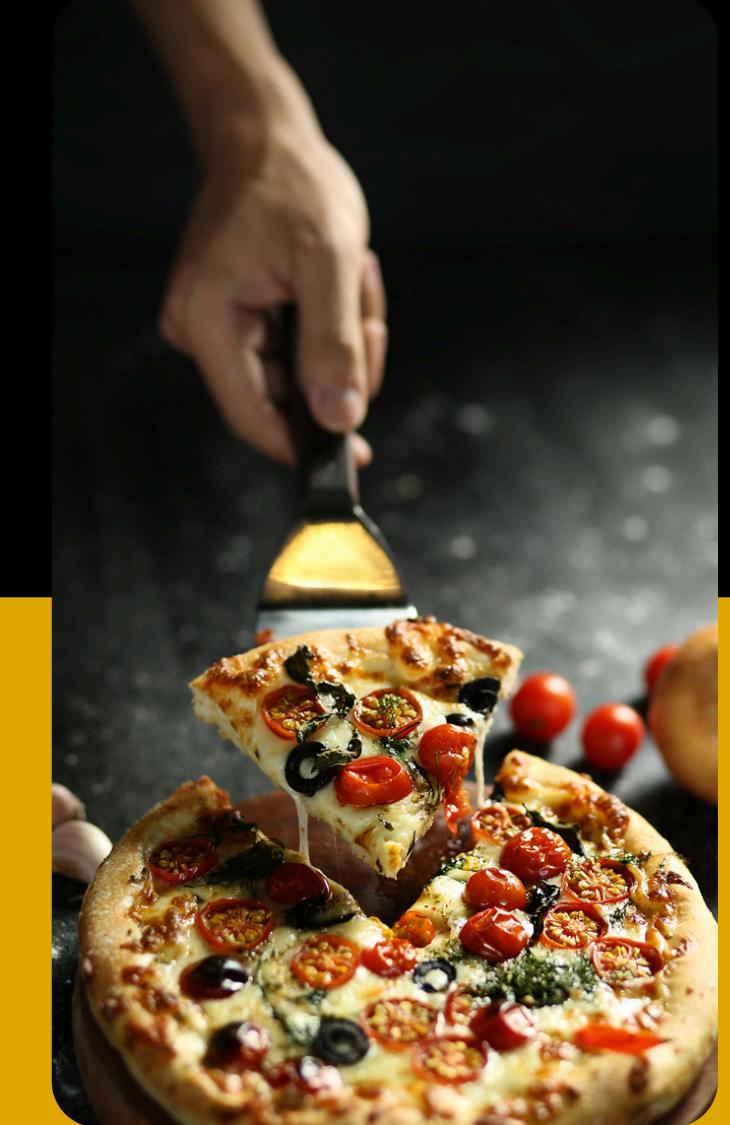
    SELECT *, RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
    FROM sales
)

SELECT
    name, revenue
FROM
    pizza_category_revenue
WHERE
    rn <= 3;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

# KEY INSIGHTS / FINDINGS

- Large-size pizzas are the most frequently ordered.
- Classic category contributes the highest revenue.
- Afternoon hours (12–1 PM) and Evening hours (4–8 PM) see peak order volume.
- A small number of pizza types generate a major share of revenue.
- Revenue grows steadily across the year.



# BUSINESS RECOMMENDATIONS

- Promote best-selling pizzas through combo offers.
- Introduce discounts during low-demand hours.
- Optimize inventory for high-demand pizza categories.
- Upsell premium pizzas to increase revenue.
- Focus marketing efforts on top revenue-generating pizzas.
- Adjust staffing and operations during peak hours.



# THANK YOU

THANK YOU FOR YOUR TIME AND ATTENTION

---

- NISHANTH NALANAGULA



**Email**  
tinkunishanth8@gmail.com



**Linkedin**  
<https://www.linkedin.com/in/nishanth-nalanagula/>



**Github**  
<https://github.com/NishanthNalanagula>

