



SHODWE
Pizza Resto

PIZZA RESTO

● WHERE EVERY SLICE TELLS A STORY



Based on a fictional database



SHODWE
Pizza Resto



PIZZA SALES ANALYSIS

AN IN-DEPTH ANALYSIS

- This project analyzes pizza sales data using SQL (MySQL).
- We extract insights related to revenue trends, order frequency, and customer preferences.
- The analysis helps optimize inventory, pricing strategies, and business decisions.

ABOUT THE ANALYSIS

01

BASIC LEVEL: Focuses on key metrics such as total orders, total revenue, top-selling pizzas, and most common sizes.

02

INTERMEDIATE LEVEL: Involves multi-table joins and time-based analysis to explore order patterns, category-wise insights, and daily averages.

03

ADVANCED LEVEL: Dives into revenue contributions, cumulative trends over time, and in-depth category-based performance analysis.



BASIC QUERIES

KEY SALES METRICS AND
TOP-SELLING ITEMS

- Q1. Retrieve the total number of orders placed.
- Q2. Calculate the total revenue generated from pizza sales.
- Q3. Identify the highest-priced pizza.
- Q4. Identify the most common pizza size ordered.
- Q5. List the top 5 most ordered pizza types along with their quantities.



01

Retrieve the total number of orders placed

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

02

Calculate the total revenue generated
from pizza sales

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





03 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  
ORDER BY pizzas.price DESC LIMIT 1;
```

04 Identify the most common pizza size ordered

```
SELECT 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 LIMIT 1;
```





- 05** List the top 5 most ordered pizzas along with their quantities

```
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;
```

INTERMEDIATE QUERIES

TIME-BASED TRENDS AND
CATEGORY INSIGHTS

- Q1. Join the necessary tables to find the total quantity of each pizza category ordered.
- Q2. Determine the distribution of orders by hour of the day.
- Q3. Join relevant tables to find the category-wise distribution of pizzas.
- Q4. Group the orders by date and calculate the average number of pizzas ordered per day.
- Q5. Determine the top 3 most ordered pizza types based on revenue.





- 01** 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;
```

- 02** Determine the distribution of orders by hour

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



03

Category-wise distribution of pizzas

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

04

Average number of pizzas ordered per day

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

05 Top 3 most ordered pizza types based on revenue

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



ADVANCE QUERIES

REVENUE BREAKDOWN AND
IN-DEPTH PERFORMANCE
ANALYSIS

- Q1. Calculate the percentage contribution of each pizza type to total revenue.
- Q2. Analyze the cumulative revenue generated over time.
- Q3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

01

Calculate the percentage contribution of each pizza type to total revenue

```
SELECT pizza_types.category,  
       ROUND(ROUND(SUM(order_details.quantity * pizzas.price),2) /  
             (SELECT ROUND(SUM(order_details.quantity * pizzas.price),2) FROM order_details  
              JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,2) AS revenue_percentage  
  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_percentage DESC;
```



02 Analyze cumulative revenue 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;
```



03

Top 3 most ordered pizza types based on revenue for each category

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





OVERALL LEARNINGS

PEAK ORDER TIMES

Most orders during lunch (12–1 PM) & dinner (6–8 PM)

POPULAR PIZZA SIZE

Large pizzas lead with 18.5K+ orders

TOP-SELLING PIZZAS

Classic Deluxe (2,453)

BBQ Chicken (2,432)

Hawaiian (2,422)

HIGHEST REVENUE PIZZAS

- Thai Chicken – \$43.4K
- BBQ Chicken – \$42.7K
- California Chicken – \$41.4K

CATEGORY-WISE PERFORMANCE

Classic – Most orders & revenue

Veggie & Supreme – Most variety (9 each)

SALES GROWTH

Steady rise to \$817.9K total

INVENTORY INSIGHT

Focus on peak hours & popular items for stock planning



CONCLUSION

ACTIONABLE BUSINESS INSIGHTS

The analysis delivers critical insights into customer preferences, peak sales periods, and top-performing products.

ENHANCED STRATEGY PLANNING

Results support data-driven decisions in inventory management, pricing strategies, and menu optimization.

POWER OF SQL IN ANALYSIS

SQL queries enabled the extraction of meaningful patterns from raw sales data, showing how structured data can unlock business intelligence.

SCALABLE METHODOLOGY

This approach can be replicated across various industries for analyzing sales, customer behavior, and product performance.





SHODWE
Pizza Resto

DIVYANSH KARTIKEY
KAUSHAL

THANK YOU

FOR ATTENTION

- PIZZA SALES ANALYSIS