



PIZZA SALES

PIZZA SALES





PIZZA SALES

WELCOME TO PIZZA SALES

My name is Muhammad Najeed Khan in this project I have utilized SQL queries to solve questions related to pizza sales.





PIZZA SALES

PIZZA SALES SQL PROJECT

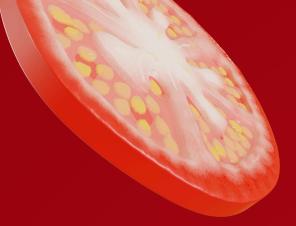
OBJECTIVE

This project analyzes pizza sales data to identify key trends and insights. The goal is to understand customer preferences, sales patterns, and peak times to provide data-driven recommendations for business optimization.

DATA SET

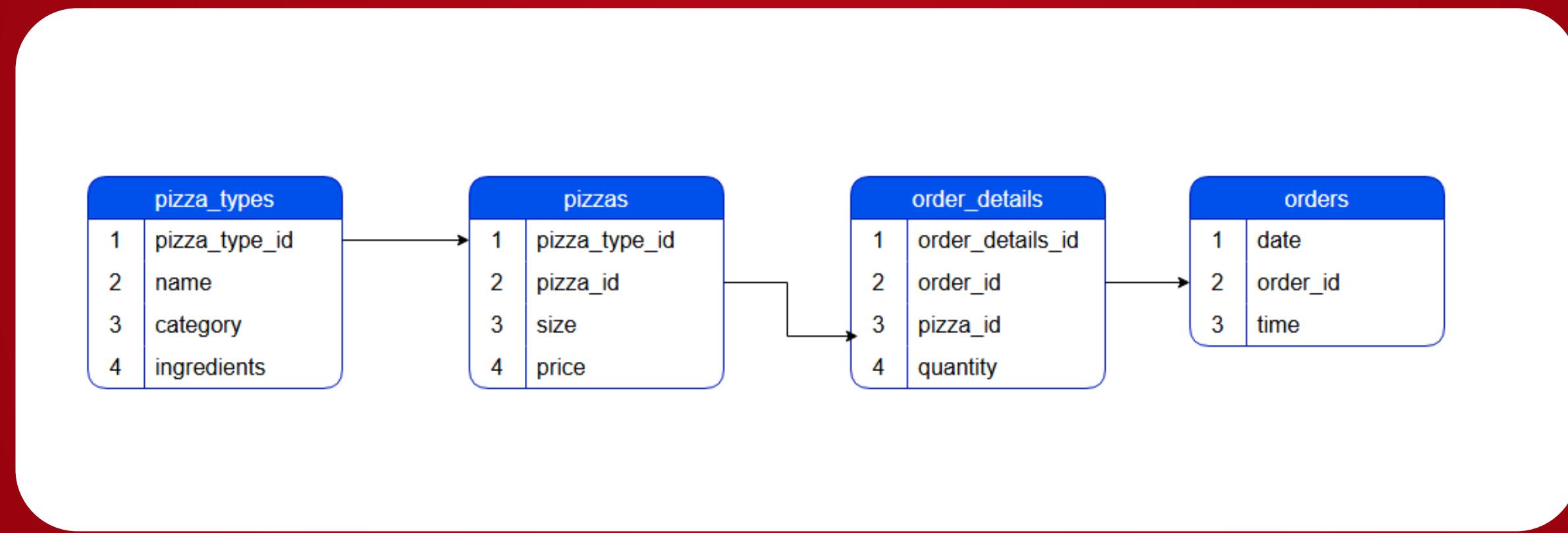
The dataset includes information on various types of pizzas, order details, orders, and pizza types. This structured data enables in-depth exploration of sales metrics across different dimensions.





PIZZA SALES

SCHEMA





RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    SUM (order_details.quantity * pizzas.price) AS revenue  
FROM order_details
```



IDENTIFY THE HIGHEST-PRICED PIZZA.

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



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;
```



LIST THE TOP 5 MOST ORDERED PIZZA TYPES 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;
```



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 pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT EXTRACT(  
    HOUR  
    FROM time  
) AS time,  
    COUNT (order_id) AS order  
FROM orders  
GROUP BY time  
ORDER BY COUNT (order_id) DESC;  
-- Different Way  
SELECT HOUR (time) AS time,  
    COUNT (order_id) AS order_count  
FROM orders  
GROUP BY HOUR (time);
```



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))
FROM (
    SELECT orders.date,
        SUM (order_details.quantity) AS quantity
    FROM orders
        JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.date
) AS order_quantity;
```



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

```
SELECT
    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
LIMIT 3;
```



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

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



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT
    date,
    SUM (revenue) OVER(
        ORDER BY date
    ) as cumulative_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 order_details.order_id = orders.order_id
    GROUP BY orders.date
) AS sales;
```

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

```
SELECT
    category,
    name,
    revenue,
    RANK() OVER (
        partition by category
        ORDER BY revenue DESC
    ) AS rank
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 revenue;
```



PIZZA SALES

THANK YOU!

