SHODWE
Pizza Resto

Pizza Sales Data Analysis using SQL

Transforming raw sales data into actionable insights

This project analyzes pizza sales data using SQL to uncover actionable business insights.

The analysis focuses on understanding sales performance, identifying top-selling and high-revenue pizzas, analyzing order trends over time, and evaluating category-wise revenue contribution to support data-driven business decisions.

Tools & Skills:

SQL | Joins | Aggregations | Window Functions | Business Analysis

SHODWE
Pizza Resto

Dataset Description

The project utilizes a relational database consisting of **four primary tables:**

- **orders**
  - order_id
  - order_date
  - order_time
- **order_details**
  - order_details_id
  - order_id
  - pizza_id
  - quantity
- **pizzas**
  - pizza_id
  - pizza_type_id
  - size
  - price
- **pizza_types**
  - pizza_type_id
  - name
  - category
  - ingredients

Tools & Skills: SQL | Joins | Aggregations | Window Functions | Business Analysis

SHODWE
Pizza Resto

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11*

Limit to 1000 rows

```sql
-- Calculate the total revenue generated from pizza sales.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_sales
27664.7

SHODWE
Pizza Resto

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

SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL

Limit to 1000 rows

```sql
-- 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	quantity
The Pepperoni Pizza	99
The Classic Deluxe Pizza	80
The Barbecue Chicken Pizza	80
The Thai Chicken Pizza	78
The Italian Supreme Pizza	76

SHODWE
Pizza Resto

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Classic	496
Supreme	416
Veggie	402
Chicken	359

SHODWE
Pizza Resto

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

SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11*

Limit to 1000 rows

```sql
-- Group the orders by date and calculate the average
-- number of pizzas ordered per day.

SELECT ROUND(AVG(quantity), 0) as avg_pizza_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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_pizza_day
129

SHODWE
Pizza Resto

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

SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12*

Limit to 1000 rows

```sql
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	revenue
The Thai Chicken Pizza	1446.5
The Barbecue Chicken Pizza	1416
The Italian Supreme Pizza	1365.75

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE OF TOTAL REVENUE

SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11*

Limit to 1000 rows

```sql
SELECT
    pt.category,
    ROUND(
        SUM(od.quantity * p.price) /
        (SELECT SUM(od2.quantity * p2.price)
         FROM order_details od2
         JOIN pizzas p2 ON p2.pizza_id = od2.pizza_id
        ) * 100,
    2) AS revenue_percentage
FROM pizza_types pt
JOIN pizzas p
    ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
    ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue_percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue_percentage
Classic	26.28
Supreme	26.2
Veggie	24.47
Chicken	23.04

SHODWE
Pizza Resto

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

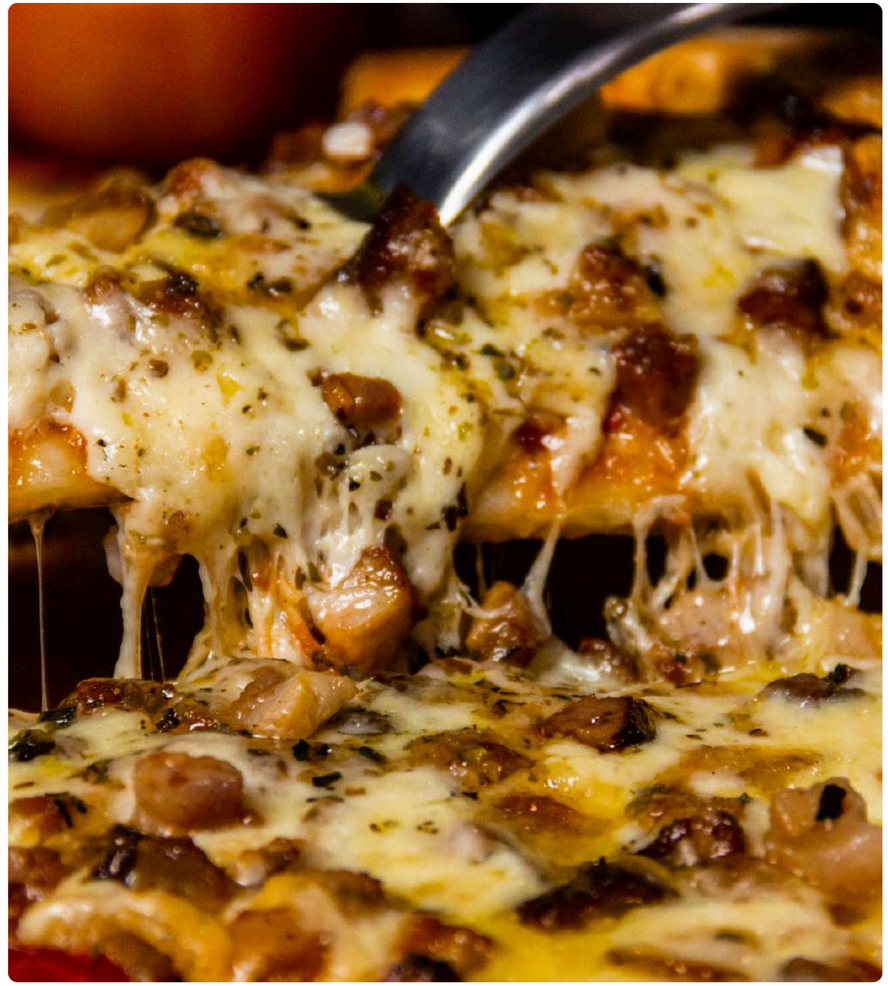
SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11*

Limit to 1000 rows

```sql
-- Analyze the cumulative revenue generated over time.
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM (
    SELECT
        o.order_date,
        SUM(od.quantity * p.price) AS revenue
    FROM order_details od
    JOIN pizzas p
        ON od.pizza_id = p.pizza_id
    JOIN orders o
        ON o.order_id = od.order_id
    GROUP BY o.order_date
) AS sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	cum_revenue
2015-01-01	2787.6000000000004
2015-01-02	5519.5
2015-01-03	8181.9



SHODWE
Pizza Resto

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11*

Limit to 1000 rows

```sql
-- Determine the distribution of orders by hour of the day.

SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455

SHODWE
Pizza Resto

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
4 •  SELECT pizzas.size,
5            COUNT(order_details.order_details_id)
6    FROM pizzas
7    JOIN order_details
8    ON pizzas.pizza_id = order_details.pizza_id
9    GROUP BY pizzas.size;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

	size	COUNT(order_details.order_details_id)
▶	M	527

SHODWE
Pizza Resto

IDENTIFY THE HIGHEST-PRICED PIZZA.

SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12*

Limit to 1000 rows

```
1   -- Identify the highest-priced pizza.
2   SELECT pizza_types.name, pizzas.price
3   from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
4   order by pizzas.price desc limit  1;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

SHODWE
Pizza Resto

Conclusion

✓ Key Insights

- Identified **top-selling and highest revenue**-generating pizzas using sales and quantity analysis.
- Analyzed **order trends by time and date** to understand customer ordering behavior.
- Evaluated **category-wise performance** and revenue contribution to total sales.
- Tracked **cumulative revenue growth** to understand overall business performance.

✓ Business Impact

- Helps optimize pricing strategy and **inventory planning**.
- Supports **data-driven decision-making** for sales and marketing.
- Provides clear visibility into **customer preferences** and peak demand periods.

✓ Skills Gained

- Advanced SQL querying using **JOINs**, **Aggregations**, Subqueries, and **Window Functions**.

This project demonstrates the practical use of SQL in real-world data analysis and business decision-making.

SHODWE
Pizza Resto

THANK YOU

FOR ATTENTION

- PIZZA RESTO PRESENTATION