PIZZA SALES REPORT

2024

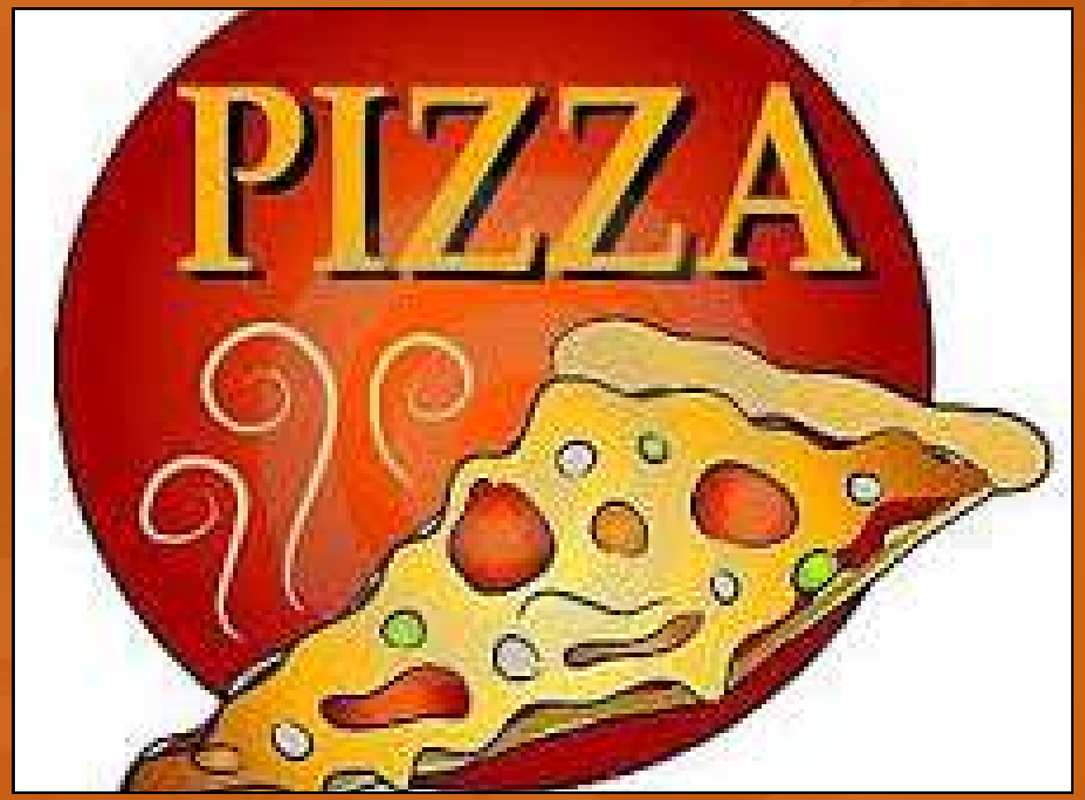
PROJECT OVERVIEW

This SQL project delves into the world of pizza sales data, utilizing SQL queries to extract valuable insights. By employing various SQL techniques, including basic to advanced queries, the project aims to uncover trends, patterns, and relationships within the dataset. The project is organized into 13 distinct queries, each designed to answer a specific question. The queries progressively increase in complexity, starting with basic data retrieval and moving towards more advanced analytical techniques.

KEY OBJECTIVES

2024

- **Data Exploration:** Thoroughly explored the pizza sales dataset to understand its structure, content, and potential insights.
- **Query Development:** Craft a series of SQL queries to address specific questions related to sales, products, customers, and other relevant aspects.
- **Data Analysis:** Analyze the query results to identify trends, patterns, and correlations within the data.
- **Insight Generation:** Derive meaningful conclusions and insights based on the analysis, providing valuable information for business decision-making.



2024

KEY INSIGHTS

The key insights from the analysis are that the total revenue is $ 817860.05, 'The Greek Pizza' is the highest priced, large size pizzas are the most ordered, classic pizzas have the highest sales, and the peak sales hour is between 11 AM and 4 PM.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

20
24

SQL File 4* | SQL File 8* | SQL File 12* | SQL File 13* | SQL File 14* | SQ

Limit to 50000 rows

```
-- Retrieve the total number of orders placed.
use pizzasales;
select count(order_id) as total_orders from orders;
```

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

total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

2024

```sql
-- Calculate the total revenue generated from pizza sales.
use pizzasales;
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

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

revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

2024

```sql
-- Identify the highest-priced pizza

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

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

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

20
24

```sql
-- Identify the most common pizza size ordered.
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS total_orders
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_orders DESC
LIMIT 3;
```

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

size	total_orders
L	18526
M	15385
S	14137

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

2024

```sql
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS most_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered DESC
LIMIT 5;
```

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

name	most_ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

20
24

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_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	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

20
24

```sql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS orders
FROM
    orders
GROUP BY hours;
```

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

hours	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

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

2024

```
-- Join relevant tables to find the category-wise distribution of pizzas.

select category, count(name) as pizza_name from pizza_types
group by category;
```

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

category	pizza_name
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

2024

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) AS average_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS a;
```

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

average_per_day
138

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

20
24

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
            1) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
```

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

name	revenue
The Thai Chicken Pizza	43434.2
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

20
24

```sql
SELECT
    pizza_types.category,
  round(  (SUM(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),0)
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id) )*100,0) as revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

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

category	revenue
Classic	27
Supreme	25
Veggie	24
Chicken	24

THANK YOU!