PIZZA SALES DATA ALNALYSIS PROJECT USING MYSQL

An Analytical Study of
Pizza Sales Trends and Key Metrics

PREPARED BY : SIMRAN KAUR

TABLE OF CONTENT

INTRODUCTION

This project analyzes pizza sales data using MySQL to identify key trends and metrics. By exploring a dataset consisting of order_details, orders, pizza_types, and pizzas, we aim to uncover insights such as total sales, top-selling pizza types, and revenue contributions. The analysis includes basic, intermediate, and advanced SQL queries to provide a comprehensive understanding of customer preferences and sales performance, helping to optimize business decisions and strategies.

DATASET OVERVIEW

The dataset consists of the following tables with corresponding row counts and columns :

1. **order_details :** 48,620 rows
   - **Columns :** order_detail_id, order_id, pizza_id, quantity

2. **orders :** 21,350 rows
   - **Columns :** order_id, date, time

3. **pizza_types :** **32 rows**
   - **Columns :** pizza_type_id, name, category, ingredients

4. **pizzas :** 96 rows
   - **Columns :** pizza_id, pizza_type_id, size, price

BASIC SQL QUERIES

Query 1 : Retrieve Total Orders

```
SELECT
    COUNT(order_id) AS 'Total Orders'
FROM
    orders;
```

Result -

Total Orders
21350

Query 2 : Calculate Total Revenue

```
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;
```

Result -

Total Revenue
817860.05

Query 3: Identify the Most Common Pizza Size

```sql
SELECT
    pizzas.size AS 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;
```

Result -)

Size	Order_Count
L	18526

Query 4 : Identify the Highest Priced Pizza

```sql
SELECT
    pizza_types.name AS 'Name of Pizza',
    pizzas.price AS 'Price'
FROM
    pizza_types
     JOIN
    pizzas
   ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result -

Name of Pizza	Price
The Greek Pizza	35.95

Query 5 : Top 5 Most Ordered Pizza Types with Quantities

```sql
SELECT
    pizza_types.name AS Name,
    SUM(order_details.quantity) AS Total_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 Name
ORDER BY Total_Quantity DESC
LIMIT 5;
```

Result -

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

INTERMEDIATE SQL QUERIES

Query 1 : Determine the Top 3 Most Ordered Pizza Types Based on Revenue

```
SELECT
    pizza_types.name AS Pizza_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_name
ORDER BY Revenue DESC
LIMIT 3;
```

Result -

Pizza_name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Query 2 : Join the Necessary Tables to Find the Total Quantity of Each Pizza Category Ordered

```sql
SELECT
    pizza_types.category AS Pizza_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 pizzas.pizza_id = order_details.pizza_id
GROUP BY Pizza_Category
ORDER BY Quantity DESC;
```

Result -

Pizza_Category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Query 3 : Determine the Distribution of Orders by Hour of the Day

```sql
SELECT
    HOUR(order_time) AS Hour,
    COUNT(order_id) AS Orders
FROM
    orders
GROUP BY Hour;
```

Result -

Hour	Orders
11	1231
12	2520
13	2455

* Note: Only a subset of result is shown

Query 4 : Group the Orders by Date and Calculate the Average Number of Pizzas Ordered per Day

```sql
SELECT
    ROUND(AVG(Quantity), 0)
    as Avg_pizza_ordered_perDay
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 -

Avg_pizza_ordered_perDay
138

Query 5 : Join Relevant Tables to Find the Category-Wise Distribution of Pizzas

```sql
SELECT
    category AS Category,
    COUNT(name) AS Name
FROM
    pizza_types
GROUP BY Category;
```

Result -

Category	Name
Chicken	6
Classic	8
Supreme	9
Veggie	9

ADVANCED SQL QUERIES

Query 1 : Calculate the Percentage Contribution of Each Pizza Type to Total Revenue

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

Result -

Category	Revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Query 2 : Analyze the Cumulative Revenue Generated Over Time

```
Select
   Date,sum(revenue) over(order by Date) as Cum_Revenue
From
(select orders.order_date as Date,
  sum(order_details.quantity * pizzas.price) as Revenue
  from orders
   join
 order_details
   on orders.order_id = order_details.order_id
   join
 pizzas
on order_details.pizza_id = pizzas.pizza_id
group by Date ) as Sales;
```

Result -

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

* Note: Only a subset of result is shown

Query 3 : Determine the Top 3 Most Ordered Pizza Types Based on Revenue for Each Pizza Category

```sql
Select Name,Revenue
 From
(select Category,Name,Revenue ,
 rank() over(partition by Category order by Revenue desc) as rn
  from
(select pizza_types.category as Category,pizza_types.name as 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 Category,Name) as A) as B
  where rn <= 3;
```

Result -

Name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75

*Note : The Top 3 most ordered pizzas of only two categories are displayed here

CONCLUSION

This project analyzed pizza sales data using SQL to uncover key insights. We started with basic queries to calculate total orders, total revenue, and the most popular pizzas, providing a foundational understanding of sales performance. Through intermediate queries, we explored deeper metrics, such as category-wise sales and peak ordering hours. Finally, we used advanced queries, including subqueries and window functions, to rank pizzas by revenue and identify sales trends.