SQL PROJECT

INTRODUCTION

Hi , My Name is Jaipaul A Koola
In this project I make use of mysql queries to solve some problems related to pizza sales.

Objective

The objective of this project is to analyze pizza sales data to identify trends and provide actionable insights that can help to increase sales and aim to key metrics and patterns within the sales data by leveraging SQL queries in MySQL.

Tools Used

- Database Management System : MySQL
- Query Tool : MySQL Workbench

DATABASE SCHEMA

1. Order_details

- order_details_id
- order_id
- pizza_id
- quantity

2. orders

- order_date
- order_id
- order_time

3. pizzas

- pizza_id
- pizza_type_id
- size
- price

4. pizza_types

- pizza_type_id
- name
- category
- ingredients

Overview

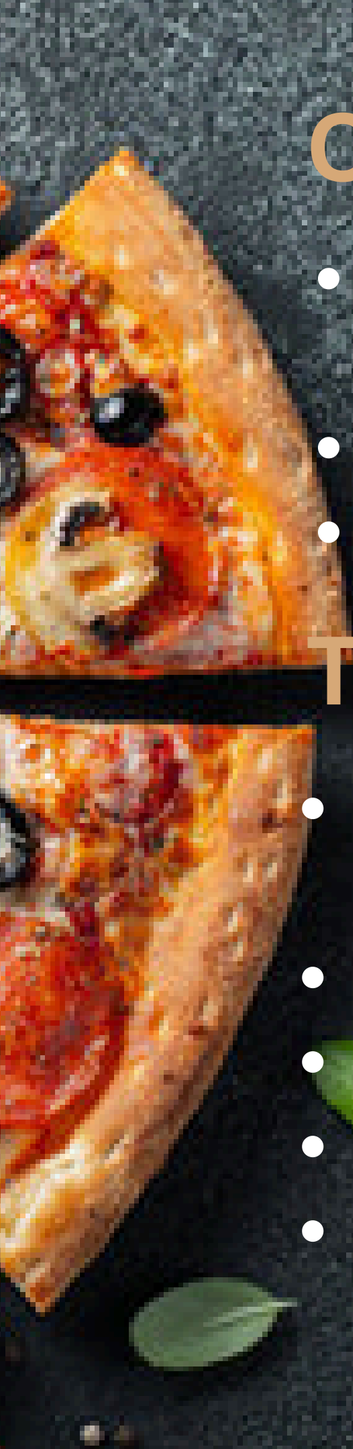
In this project, addressed several critical questions related to pizza sales. These questions are organized under different categories to provide a structured analysis

Sales Volume and Revenue

- Retrieve the total number of orders placed
- Calculate the total revenue generated from pizza sales

Product Analysis

- Identify the highest-priced pizza
- Identify the most common pizza size ordered
- List the 5 most ordered pizza types along with their quantities



Category and Time Analysis

- Join the necessary tables and find the total quantity of each pizza category ordered
- Determine the distribution of orders by hour of the day
- Join the relevant tables and find the category-wise distribution of pizzas

Trend and Performance Analysis

- Group the orders by date and calculate the average number of pizzas ordered per day
- Determine the top 3 most ordered pizza types based on revenue
- Calculate the percentage contribution of each pizza type to total revenue
- Analyze the cumulative revenue generated over time
- Determine the top 3 most ordered pizza types based on revenue for each pizza category

Retrieve the total number of orders placed

```sql
select count(order_id) as total_orders from orders;
```

Result Grid

total_orders
21350

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

	total_sales
▶	817860.05

Identify the highest-priced pizza

```sql
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;
```

Result Grid | Filter Ro

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered

```sql
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;
```

Result Grid | Filter

size	order_count
L	18526

List the top 5 most ordered pizza types along with their quantities

```sql
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:

name	quantity
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.

```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 Rov

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

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

Result Grid | Filter

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Join relevent tables to find the category-wise distribution of pizzas

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

Result Grid | Filter Row

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_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;
```

Result Grid | Filter Rows:

avg_pizzas_ordered_per_day
138

Determine the top 3 most ordered pizza types on revenue

```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:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (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)) * 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 pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time

```sql
select  order_date,
sum(revenue) over(order by order_date) as cum_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;
```

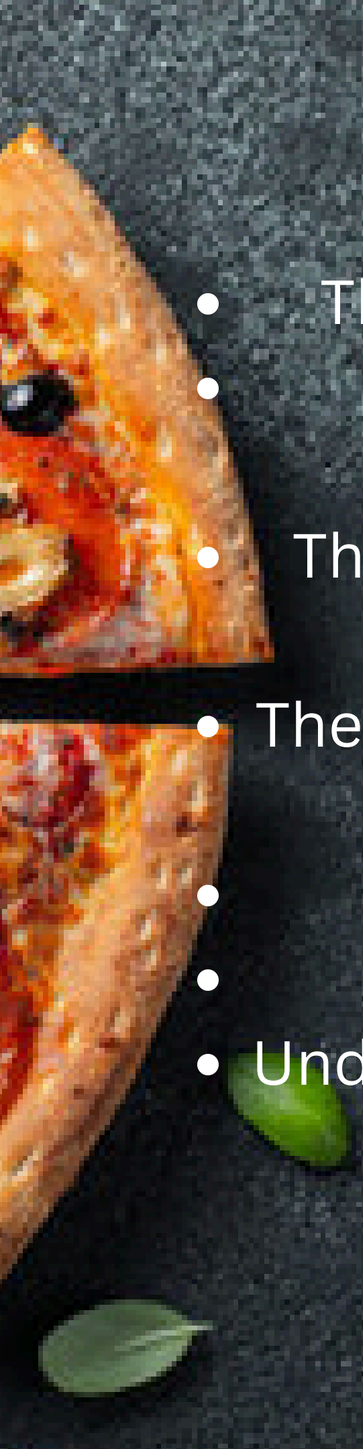
Determine the top 3 most ordered pizza types based on revenue for each pizza category

```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, 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;
```

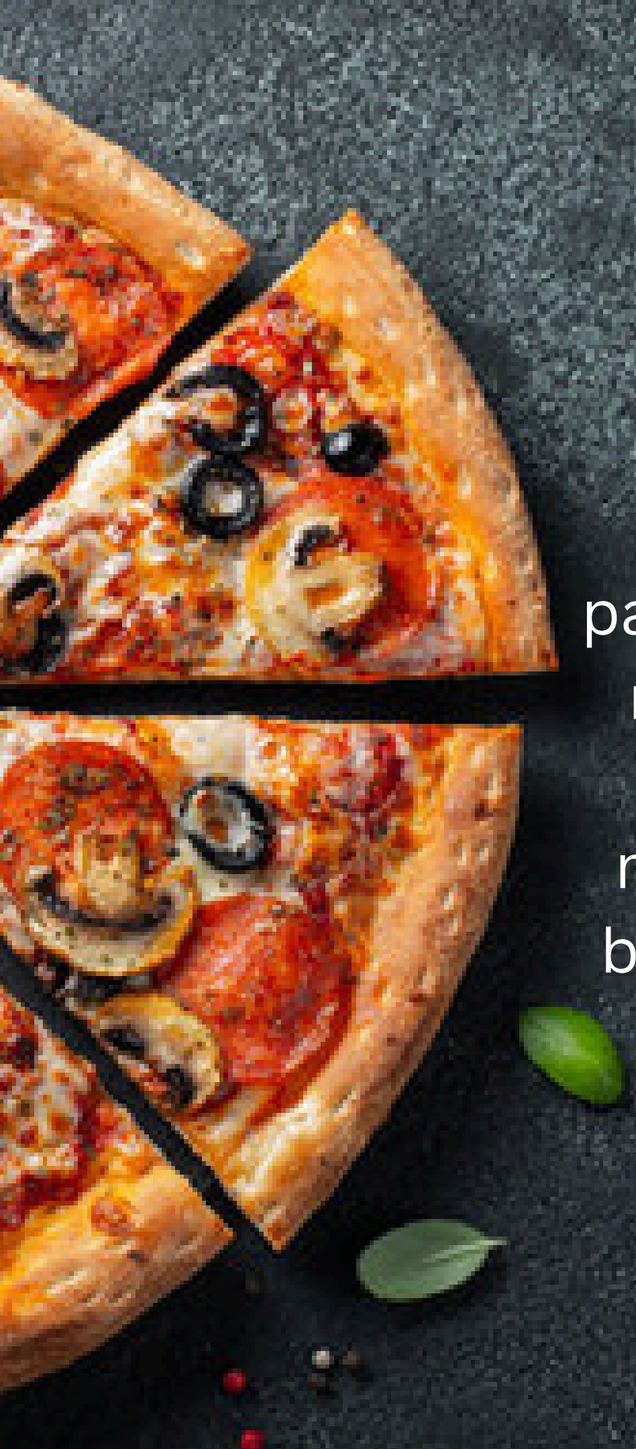
Result Grid | Filter Rows:

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
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Insights



- The analysis revealed that a Large(L) size pizza is most commonly ordered.
- The Thai Chicken Pizza($43434.25), Barbecue Chiken Pizza($42768) and California Chiken pizza($41409.5) generate the highest revenue.
- The most ordered pizza types based on quantities are the classic delux pizza (2453). the barbecue chicken pizza(2432) and the Hawaiian pizza (2422)
- The highest-priced pizza is the Greek Pizza ($35.95) is contributing significantly to the revenue.
- The average number of pizzas ordered per day is 138.
- Cumulative revenue trends provide a long-term view of performance.
- Understanding the percentage contribution of each pizza type to total revenue helps in identifying customer preferences.



Conclusion

The analysis of pizza sales data reveals valuable insights for optimizing business operations and driving sales growth. By leveraging MySQL queries we've identified total orders and revenue, as well as customer preferences and temporal patterns in ordering behavior. These findings inform actionable recommendations for menu optimization, pricing strategies, and resource allocation. Moving forward, continuous monitoring and adaptation based on data-driven insights will be crucial for sustaining competitive advantage and achieving long-term success.

THANK YOU