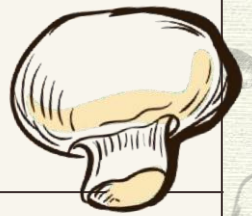
PIZZA SALES Analysis Using SQL

INTRODUCTION

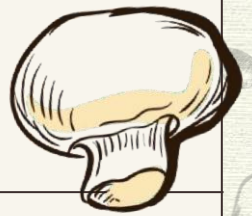
"Hello, everyone. My name is Yash Dwivedi, I am excited to present my SQL project focused on analyzing pizza sales. In this project, I have utilized SQL to delve deep into our sales data, uncovering valuable insights that can help us better understand our market and improve our operations. We will explore various aspects such as sales trends over different periods, the popularity of different pizza types, customer demographics, and the impact of promotions and discounts on sales."



OBJECTIVE

We are diving into an insightful analysis of pizza sales, where we explore various facts of our business performance and customer preferences. As the pizza industry continues to grow and evolve, understanding sales trends and customer behavior is crucial for making informed decisions that drive growth and customer satisfaction.

we are cover key aspects such as sales trends over time, the performance of different pizza varieties, customer demographics, and peak sales periods. By leveraging data-driven insights, our aim to identify areas of opportunity and develop strategies to enhance our offerings and boost overall sales.



Description about data set

- pizzas.csv: Contains information about pizza types.
- pizza_types.csv: Provides details about pizza categories and prices.
- orders.csv: Includes order information (order IDs, timestamps, etc.).
- orders_details.csv: Contains transaction details (order items, quantities, and amounts).

Database schema

Questions to be answered

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- 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.

1. Retrieve the total number of orders placed.

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

Result Grid

	Total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales.

```
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 Grid

	Total_revenue
▶	817860.05

3. 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 2 DESC
LIMIT 1;
```

4. Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Count_orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

Result Grid | Filter

	size	Count_orders
▶	L	18526

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

```sql
SELECT
    pt.name,
SUM(od.quantity) AS Quantity
FROM
    pizza_types as pt
        JOIN
    pizzas as p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details as od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 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

6. Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT
    pt.category, SUM(od.quantity) AS Quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rows:

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7.Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS Order_count
FROM
    orders
GROUP BY 1;
```

Result Grid

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

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

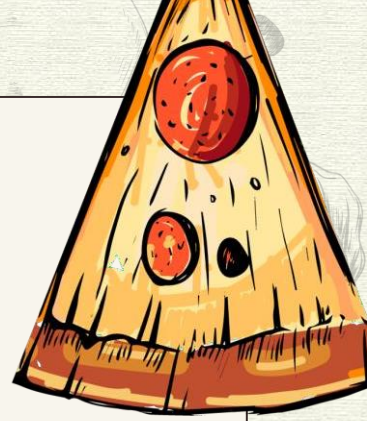
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

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

9. 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_per_day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders AS o
    JOIN order_details AS od ON o.order_id = od.order_id
    GROUP BY 1) AS order_qty;
```



Result Grid | Filter Rows:

	Avg_pizza_ordered_per_day
▶	138

10. Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pt.name, ROUND(SUM(od.quantity * p.price), 2) AS Revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 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

11. Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                    ROUND(SUM(od.quantity * p.price), 2)
                FROM
                    order_details AS od
                        JOIN
                    pizzas AS p ON od.pizza_id = p.pizza_id) * 100,2) AS Revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rov

	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12. Analyze the cumulative revenue generated over time.

```
select order_date,Revenue,
       round(sum(Revenue) over(order by order_date),2) as Cum_Revenue
from
        (select o.order_date,
        round(sum(od.quantity*p.price),2) as Revenue
        from order_details as od
        join pizzas as p
        on od.pizza_id=p.pizza_id
        join orders as o
        on o.order_id=od.order_id
        group by 1) as Sales;
```

Result Grid | Filter Rows:

order_date	Revenue	Cum_Revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5

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

```
select name,Revenue,category from
(select category,name,Revenue,
rank() over(partition by category order by Revenue desc) as rn
from
(select pt.category,pt.name,
sum(od.quantity*p.price) as Revenue
from pizza_types as pt
join pizzas as p
on p.pizza_type_id=pt.pizza_type_id
join order_details as od
on od.pizza_id=p.pizza_id
group by 1,2) as a) as b
where rn<=3
group by 1,2,3;
```

Result Grid | Filter Rows: | Export:

name	Revenue	category
The Thai Chicken Pizza	43434.25	Chicken
The Barbecue Chicken Pizza	42768	Chicken
The California Chicken Pizza	41409.5	Chicken
The Classic Deluxe Pizza	38180.5	Classic
The Hawaiian Pizza	32273.25	Classic
The Pepperoni Pizza	30161.75	Classic
The Spicy Italian Pizza	34831.25	Supreme
The Italian Supreme Pizza	33476.75	Supreme
The Sicilian Pizza	30940.5	Supreme
The Four Cheese Pizza	32265.70000000065	Veggie
The Mexicana Pizza	26780.75	Veggie
The Five Cheese Pizza	26066.5	Veggie

THANKS!