HOT & SPICY

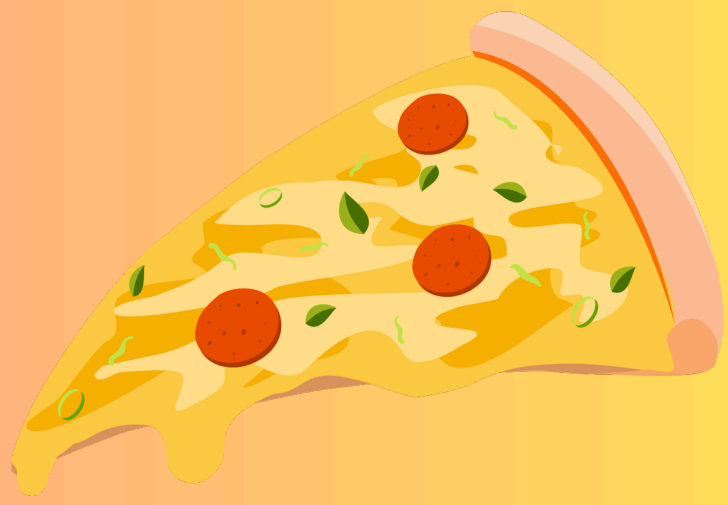
PIZZA

by Rohit

SQL Project

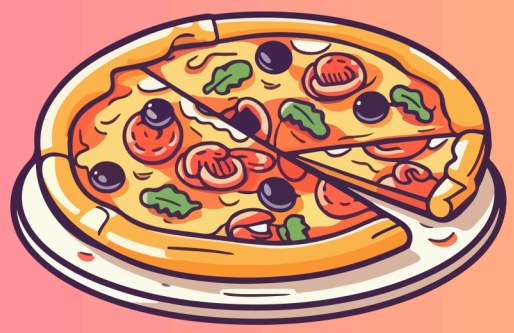
Pizza Sales

By Rohit



Background

The pizza sales analysis project aims to leverage data-driven insights to understand sales patterns and customer preferences. By analyzing sales data, the business can make informed decisions to enhance customer satisfaction and drive revenue growth.



Dataset Description

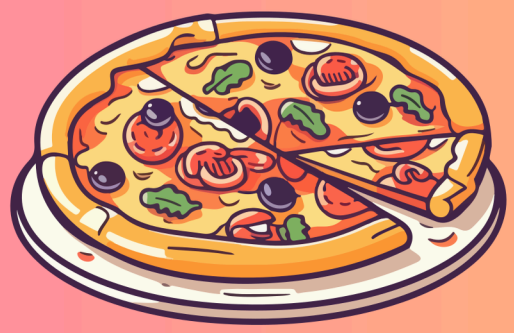
The dataset used for this analysis was sourced from Kaggle, a popular platform for datasets and data science competitions. It includes comprehensive data on orders, pizza types, and detailed order information.

Key Tables in the Dataset:

- **order_details:** Contains details of each order, including pizza ID and quantity.
- **orders:** Includes information about each order, such as order ID, date, and time.
- **pizza_types:** Provides information about the types of pizzas, including names and categories.
- **pizzas:** Contains details about individual pizzas, including their prices and sizes.

Analytical Approach

The analysis employs SQL queries to extract, transform, and analyze the data. Key metrics and insights are derived using various SQL techniques, including aggregations, joins, and filtering. This approach enables a comprehensive understanding of the sales dynamics and customer behavior.



1. *Retrieve the total number of orders placed*

input

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

output

	total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales

input

```
SELECT
  round(SUM(quantity * price),2)  AS Revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

output

	Revenue
▶	817860.05

3. Identify the highest-priced pizza.

input

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

output

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

input

```sql
SELECT
    size, COUNT(order_details_id) AS order_count
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY order_count DESC
limit 5;
```

output

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

input

```
SELECT
    name, SUM(quantity) AS Quantity
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY name
ORDER BY Quantity DESC
LIMIT 5;
```

output

	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.

input

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

output

	category	Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

input

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

output

HOUR	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

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

input

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

output

	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

input

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_no
FROM
    (SELECT
        order_date, SUM(quantity) AS Quantity
    FROM
        order_details
    JOIN orders ON orders.order_id = order_details.order_id
    GROUP BY order_date) AS order_quantity;
```

output

	avg_no
▶	138

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

input

```
SELECT
 name , round(sum(price*quantity ),2) as Revenue
FROM
    pizzas
    JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id
join  pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_id
group by name
order by Revenue desc
limit 3;
```

output

	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.

input

```
SELECT
round(sum(price*quantity)/(SELECT
  round(SUM(quantity * price),2)  AS Revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id)*100,2) as Revenue ,category
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
join pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_id
group by category
order by Revenue desc;
```

output

	Revenue	category
▶	26.91	Classic
	25.46	Supreme
	23.96	Chicken
	23.68	Veggie

12. Analyze the cumulative revenue generated over time.

input

```
select order_date,
sum(revenue) over(order by order_date) as cum_revnue
from
(SELECT
    order_date, SUM(price * quantity) as revenue
FROM
    order_details
        JOIN
    orders ON order_details.order_id = orders.order_id
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY order_date) as sales;
```

output

order_date	cum_revnue
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

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

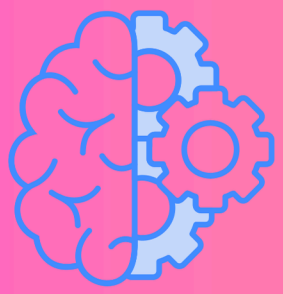
input

```sql
select name,revenue from
(SELECT
    category,  name ,  revenue,
    rank() over(partition by category order by revenue desc) as rn
    from

(SELECT
    category,  name , SUM(price * quantity) 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;
```

output

	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



Summary of Key Findings

- **Total Orders and Revenue:** The analysis revealed that the total number of orders placed was **21,350**, generating a total revenue of **$817,860.05**.

- **Popular Pizzas:** The highest-priced pizza was identified as **The Greek Pizza**, and the most commonly ordered size was **Large(L)**.

- **Top Orders:** The top 5 most ordered pizzas were **The Classic Deluxe Pizza, The barbecue Chicken Pizza, The Hawaiian Pizza, The Pepperoni Pizza, and The Thai Chicken Pizza**, contributing significantly to the sales.

- **Order Distribution:** The majority of orders were placed during **noon**, indicating peak sales periods.

Final Thoughts

This analysis provides valuable insights into pizza sales patterns and customer preferences. By leveraging these findings, the business can make informed decisions to optimize operations and boost sales.

Thank you for your attention, and I welcome any questions or discussions.

Thank You!