it's Pizza Time

Pizza Sales Performance Analysis with Sql Server

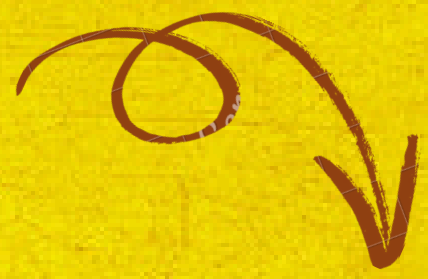
Presented By:
Siblan Dutami Horo
Data Analytics Project

introduction

in this project, we aim to analyze pizza sales data using SQL to extract valuable insights. By examining sales patterns, identifying customer preferences, and understanding revenue trends, we can provide actionable recommendations to improve business performance.

SQL will help us effciently query large datasets, perform cleaning, and generate reports for better decision-making. This analysis can assist in optimizing inventory inventory, tailoring market strategies and customer satisfaction.

Overview of the Project



This project focuses on analysing pizza sales data using SQL Server. By writing and executing SQL queries, we im to gain insights into custo- mer preferences, identify best selling pizzas, and evaluate overall sales performance. The project also explore how SQL can be used to clean, manipulate, and analyze large datasets effectively.

Objectives:

*Analyze pizza sales data to extract insights.

*identify best selling pizzas, peak sales times, and customer prefernces.

*Provide data driven recommendations to optimise sales and inventory.

Scope:

*customer purchase pattern.

*Revenue generation by pizza types and sizes.

*Store performance comparison

Database Schema

includes key tables like:

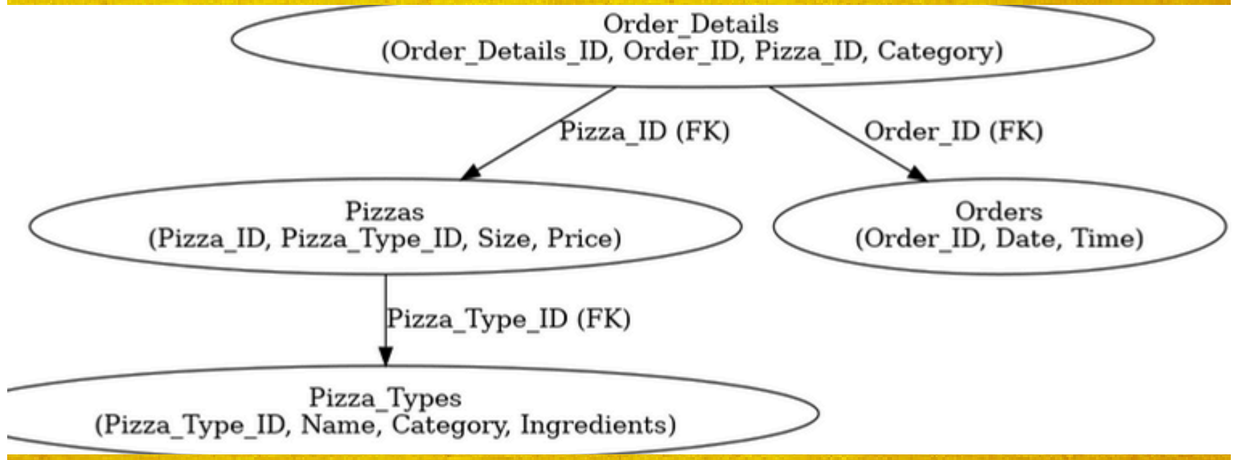
*Pizzas ⟨Pizza__id, Pizza__type__id, Size, Price⟩

*Pizza__Types⟨pizza__type__id, Name, Category, ingredients⟩

*orders⟨order__id, Date, Time⟩

*order__Details⟨order__details__id, order__id, pizza__id, Category⟩

The database schema digram



Data Exploration & Cleaning

- Record Count: Determine the total number of records in the dataset.
- Customer Count: Find out how many unique customers are in the dataset.
- Category Count: identify all unique product categories in the dataset.
- Null Value Check: Check for any null values in the dataset and delete records with missing data.

1. Retrieve the total number of orders placed.

```sql
--Retrieve the total number of orders placed.

SELECT
    COUNT(*) AS Total_Orders
FROM
    orders$
```

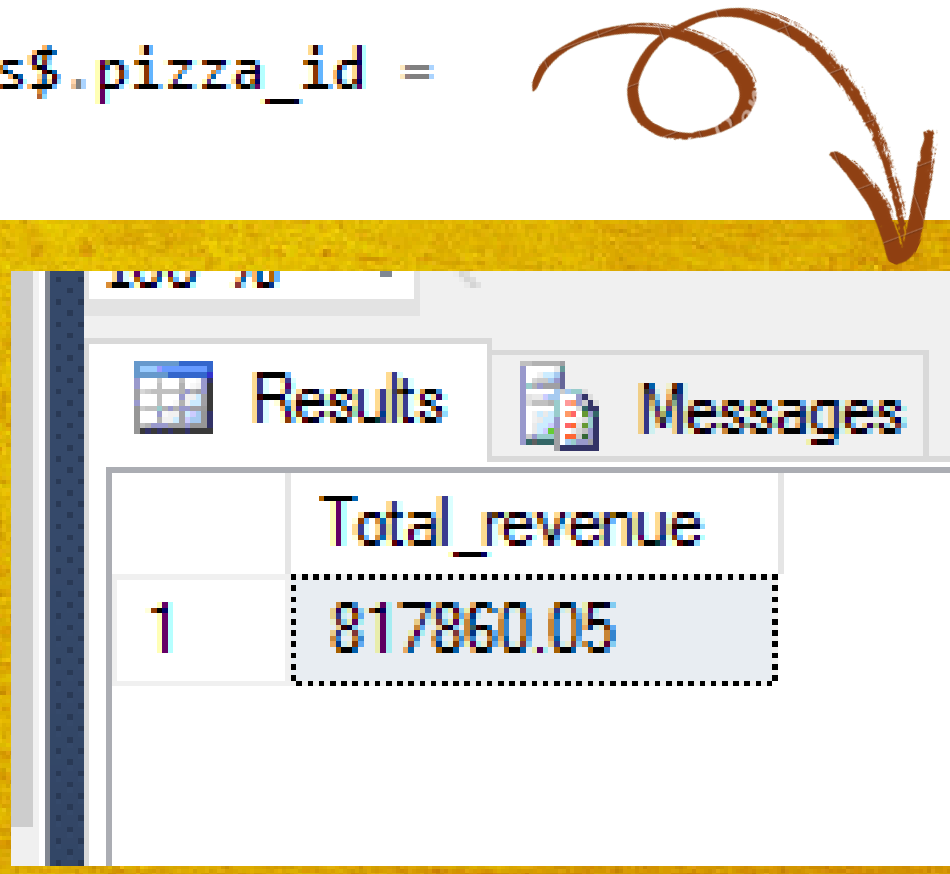
Results | Messages

	Total_Orders
1	21350

2.calculate the total revenue generated from pizza sales

```sql
---Calculate the total revenue generated from pizza sales.

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



3. identify the highest-priced pizza

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

SELECT TOP 1
    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;
```

Results | Messages

	Name	Price
1	The Greek Pizza	35.95

4. identify the most common pizza size ordered

```sql
---Identify the most common pizza size ordered.

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;
```

Results | Messages

	size	Order_Count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

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

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

SELECT TOP 5
    pizza_types$.name,
    SUM(order_details$.quantity) AS Quantity
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
    Quantity DESC;
```

Results | Messages

	name	Quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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

```sql
---Join the necessary tables to find the total quantity of each pizza category ordered.

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

Results | Messages

	category	Total_Quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

```sql
----Determine the distribution of orders by hour of the day.

SELECT
     DATEPART(HOUR, time) AS Time_in_Hours,
     COUNT(order_id) AS Orders_Count
FROM
     orders$
GROUP BY
     DATEPART(HOUR, time)
ORDER BY
     Time_in_Hours;
```

Results | Messages

	Hours_Order_Time	Count_Orders
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

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

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

SELECT
    Category,
    COUNT(Name) AS Pizza_Types
FROM
    pizza_types$
GROUP BY
    Category;
```

Results | Messages

	Category	Count_Pizza_Type
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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

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

SELECT
    ROUND(AVG(Quantity), 0) AS
Average_Orders_PerDay
FROM (
    SELECT
        orders$.Date,
        SUM(order_details$.Quantity) AS
Quantity
    FROM
        orders$
    JOIN
        order_details$
        ON orders$.order_id =
order_details$.order_id
    GROUP BY
        orders$.Date
) AS Total_Quantity;
```

Results | Messages

	Average_Orders_PerDay
1	138

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

```sql
---Determine the top 3 most ordered pizza types based on revenue.

SELECT TOP 3
    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 pizzas$.pizza_id =
order_details$.pizza_id
GROUP BY
    pizza_types$.Name
ORDER BY
    Revenue DESC;
```

Results | Messages

	Name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

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

```sql
---Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types$.Category,
    ROUND(SUM(pizzas$.price * order_details$.Quantity) / (SELECT
    ROUND(SUM(pizzas$.price * order_details$.Quantity),
2) AS Total_Sales
FROM
    order_details$
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 pizzas$.pizza_id =
order_details$.pizza_id
GROUP BY
    pizza_types$.Category
ORDER BY
    Revenue DESC;
```

Results | Messages

	Category	Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

12. Analyze the cumulative revenue generated over time.

```sql
----Analyze the cumulative revenue generated over time.

SELECT Date,
       SUM(Revenue) OVER(ORDER BY Date) AS
Cum_Revenue
FROM (
    SELECT orders$.Date,
           SUM(order_details$.Quantity *
pizzas$.Price) AS Revenue
    FROM order_details$
    JOIN pizzas$ ON order_details$.pizza_id =
pizzas$.pizza_id
    JOIN orders$ ON order_details$.order_id =
orders$.order_id
    GROUP BY orders$.Date
) AS Sales;
```

Results | Messages

	Date	Cum_Revenue
1	2015-01-01 00:00:00.000	2713.85
2	2015-01-02 00:00:00.000	5445.75
3	2015-01-03 00:00:00.000	8108.15
4	2015-01-04 00:00:00.000	9863.6
5	2015-01-05 00:00:00.000	11929.55
6	2015-01-06 00:00:00.000	14358.5
7	2015-01-07 00:00:00.000	16560.7
8	2015-01-08 00:00:00.000	19399.05
9	2015-01-09 00:00:00.000	21526.4
10	2015-01-10 00:00:00.000	23990.35

Results | Messages

	Date	Cum_Revenue
107	2015-04-17 00:00:00.000	245724.8
108	2015-04-18 00:00:00.000	248011
109	2015-04-19 00:00:00.000	249538.95
110	2015-04-20 00:00:00.000	251998.4
111	2015-04-21 00:00:00.000	254211.55
112	2015-04-22 00:00:00.000	256405
113	2015-04-23 00:00:00.000	258831.15
114	2015-04-24 00:00:00.000	261810.35
115	2015-04-25 00:00:00.000	263899.55
116	2015-04-26 00:00:00.000	265666.95

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

```sql
----Determine the top 3 most ordered pizza types based on revenue for each pizza category.

SELECT Name,
       Revenue
FROM (
    SELECT Category,
           Name,
           revenue,
           RANK() OVER(PARTITION BY Category
ORDER BY Revenue DESC) AS RANKS
    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 RANKS <= 3;
```

Results | Messages

	Name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.5
10	The Four Cheese Pizza	32265....
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5

insights and reccomentations

- Sales insights: From previous analysis, it's evident that Large(L) pizzas are most ordered this suggests that customers prefer larger sizes possibly for sharing.

- Order volume by Time of Day: Analyze peak hours of pizza orders. Typically, lunch and dinner hours (1PM-9PM) see most actively.

RECCOMENDATIONS:

- Promote baverages, sides, and desserts with pizza orders to increase average order value.

- if specific regions show high sales, run localized ad campaigns or collaborate with nearby business.

Conclusion

The pizza sales project provided meaningful insights into customer preferences, sales performance, and revenue trends. By leberaging SQL for data extraction and analysis.

the analysis higlighted the dominance of large and medium-sized pizzas in sales volume, confirming their popularity among customers.

Overall, the project successfully demonstrated the value of data-driven decision-making in the food and beverage industry.

This project not only enhances analytical skills but also strengthened the ability to interpret business challanges using data. it serves as a strong foundation for future analytical endeavors in the field of data analytics.

Thank You

Thank you for your time!
i am open to any questions.

Contact info: Siblanhoro4@gmail.com