Data Analysis Of Pizza Sales

By Ananya Kumar

Introduction

The Pizza Sales Project is designed to analyze and optimize the sales performance of a pizza business using SQL and Power BI. The project aims to provide insightful data analysis and visualizations to help understand sales trends, customer preferences, and operational efficiencies.

Agenda

1

- Utilize SQL to extract and transform sales data from various sources.
- Clean and preprocess the data to ensure accuracy and consistency.

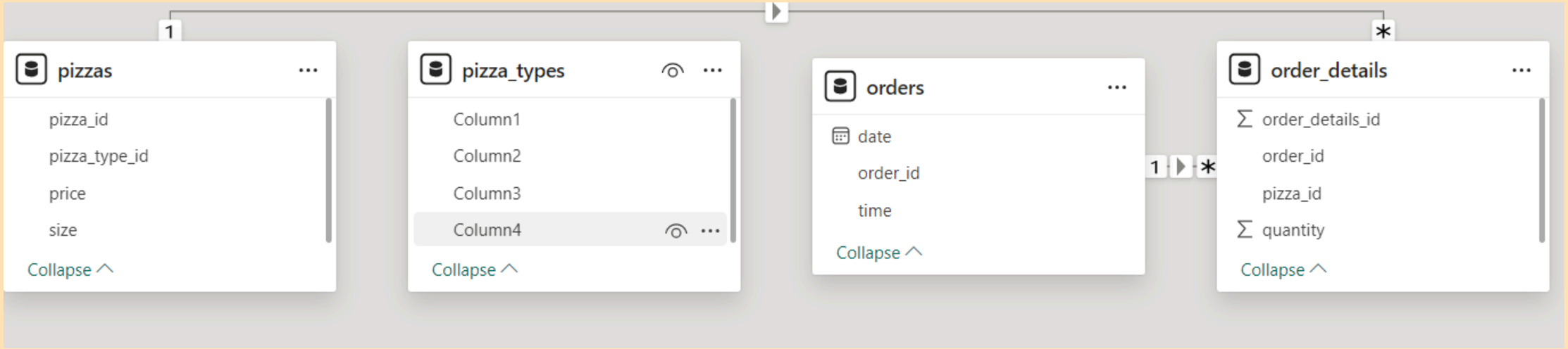
2

- Perform in-depth sales analysis to identify top-selling products, peak sales periods, and customer demographics.
- Analyze sales performance across different regions and outlets.

3

- Use Power BI to create interactive dashboards and reports.
- Visualize key metrics such as total sales, average order value, and sales growth over time.
- Provide visual insights into customer buying behavior and product performance.

Schema


1
*
pizzas
pizza_id
pizza_type_id
price
size
Collapse
pizza_types
Column1
Column2
Column3
Column4
Collapse
orders
date
order_id
time
Collapse
1
*
order_details
order_details_id
order_id
pizza_id
quantity
Collapse

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

```sql
SELECT
    ROUND(SUM(order_details.Quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = Order_details.Pizza_id;
```

Result Grid

	total_revenue
▶	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 most_common_pizza_size
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.Pizza_id
GROUP BY pizzas.size
ORDER BY most_common_pizza_size DESC;
```

Result Grid | Filter Rows:

size	most_common_pizza_size
L	18526
M	15385
S	14137
XL	544
XXL	28

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
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 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

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_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_pizza_ordered_per_day
138

Determine the top 3 most ordered pizza types based 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 Rows:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cummulative revenue generated over time

```sql
SELECT Order_Date,
SUM(revenue) OVER (ORDER BY Order_Date) AS cummulative_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;
```

Result Grid | Filter Rows:

Order_Date	cummulative_revenue
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.8500000001
2015-12-19	797083.05
2015-12-20	799187.9500000001
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

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;
```

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

Conclusion

The Pizza Sales Project has successfully demonstrated the transformative power of data analytics in driving business success. By leveraging SQL for efficient data extraction and transformation, and Power BI for compelling data visualization, the project has provided valuable insights into sales performance, customer preferences, and operational efficiencies. Through detailed analysis, we have identified key trends such as top-selling products, peak sales periods, and regional performance variations. The interactive dashboards created in Power BI have made these insights accessible and actionable, enabling data-driven decision-making at all levels of the organization.


Thank You