SQL PROJECT

PIZZA SALES ANALYSIS USING SQL

By:

Name: Sagar Gupta

E-mail: krishanasagar406@gmail.com

Liked In: https://www.linkedin.com/in/sagar-gupta087/

Qualification: BCA Undergraduate (2021 -2024)

GitHub Link: https://github.com/Sagar-Gupta008/Pizza_Sales_Analysis---SQL-

AIM:

The objective of a project on "Pizza Sales Analysis Using SQL" is to leverage SQL (Structured Query Language) to analyze sales data for a pizza business. This involves extracting, processing, and interpreting data from a database to gain insights into various aspects of the business. Here are some specific objectives:

1. Data Collection and Organization:
- Gather all relevant sales data, including order details, customer information, and product details.
- Organize the data into structured tables within a relational database.

2. Sales Performance Analysis:
- Analyze overall sales performance over different time periods (daily, weekly, monthly).
- Identify peak sales periods and trends.

3. Product Analysis:
- Determine the best-selling and least-selling pizzas.
- Analyze the performance of different pizza categories (e.g., vegetarian, meat lovers).

4. Revenue Analysis:
- Calculate total revenue, average order value, and revenue per product.

5. Order Patterns:
- Identify order patterns, such as popular days of the week and times of day for orders.

WHAT IS SQL?

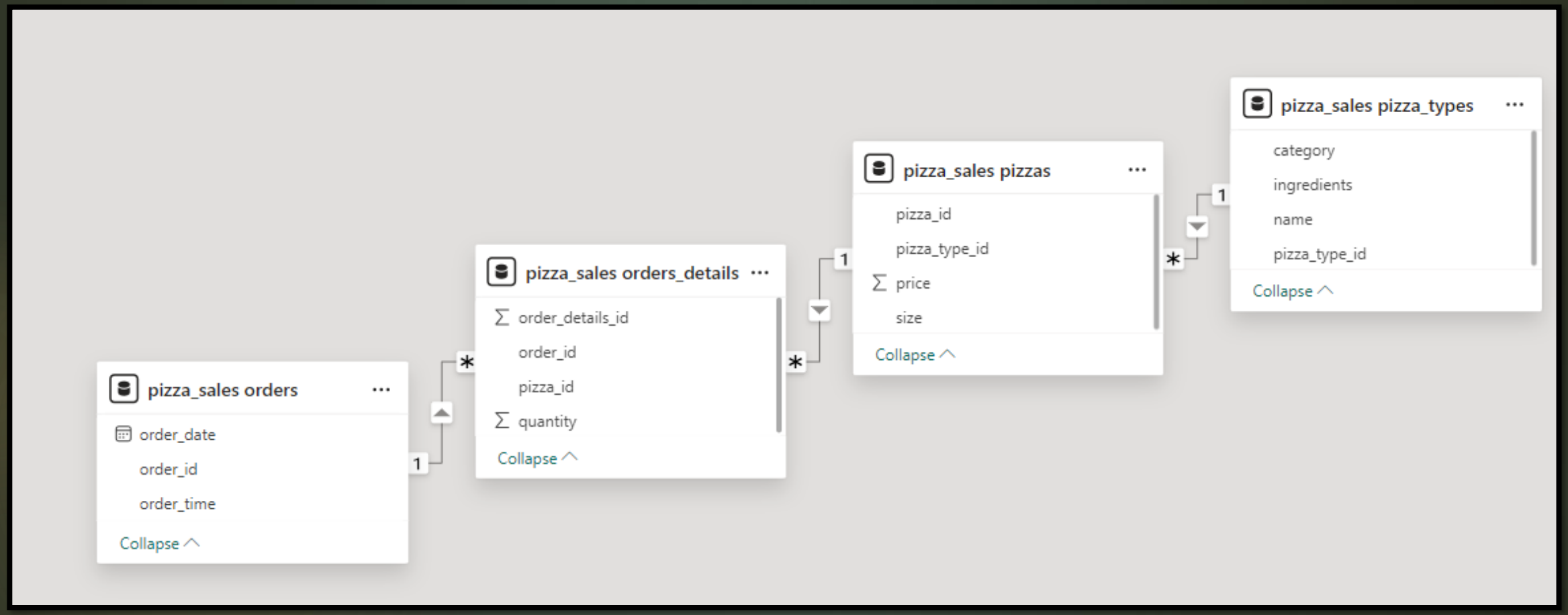
- SQL (Structured Query Language) is a standardized programming language used for managing and manipulating relational databases.
- It is designed for querying, updating, and managing data stored in relational database management systems (RDBMS).

WHAT IS MySQL WORKBENCH?

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. It provides a suite of tools to design, develop, and manage MySQL databases. Some key features include:

1. Data Modelling
2. Database Administration
3. SQL Development

DATASET SCHEMA



BASIC QUESTIONS

Q1 Retrieve the total number of orders placed.

```
select count(order_id) as Total_orders from orders;
```

Total_orders
21350

Q2 Calculate the total revenue generated from pizza sales.

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

Total_Revenue
817860.05

Q3 Identify the highest-priced pizza.

```
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;
```

name	price
The Greek Pizza	35.95

Q4 Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(orders_details.order_id)
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(orders_details.order_id);
```

size	COUNT(orders_details.order_id)
XXL	28
XL	544
S	14137
M	15385
L	18526

Q5 List the top 5 most ordered pizza types along with their quantities.


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

Pizza_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

INTERMEDIATE QUESTIONS

Q1 Join the necessary tables to find the total quantity of each pizza category ordered.

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

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q2 Determine the distribution of orders by hour of the day.

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

Hours	Orders
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

Q3 Join relevant tables to find the category-wise distribution of pizzas.

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

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

Q4 Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) AS Avg_Pizza_Ordered_Per_Day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS Order_Quantity;
```

Avg_Pizza_Ordered_Per_Day
138

Q5 Determine the top 3 most ordered pizza types based on revenue.


```sql
SELECT
    pizza_types.name,
    SUM((orders_details.quantity * pizzas.price)) AS Total_Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Revenue DESC
LIMIT 3;
```

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

ADVANCED QUESTIONS

Q1 Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.category,
    ROUND((SUM(orders_details.quantity * pizzas.price)) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS Total_Revenue
                FROM
                    orders_details
                        JOIN
                    pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
            2) AS Percentage_Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Percentage_Revenue DESC;
```

category	Percentage_Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q2 Analyze the cumulative revenue generated over time.


```sql
select order_date,revenue,
sum(Revenue) over(order by order_date) as Cumulative_Revenue
from
(select
orders.order_date,
sum(orders_details.quantity * pizzas.price) as Revenue
from orders join orders_details
on orders.order_id=orders_details.order_id
join pizzas
on pizzas.pizza_id=orders_details.pizza_id
group by orders.order_date) as Sales_Over_Time;
```

order_date	revenue	Cumulative_Revenue
2015-01-01	2713.8500000000004	2713.8500000000004
2015-01-02	2731.8999999999996	5445.75
2015-01-03	2662.3999999999996	8108.15
2015-01-04	1755.4500000000003	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2000000000003	16560.7
2015-01-08	2838.3499999999995	19399.05
2015-01-09	2127.3500000000004	21526.4
2015-01-10	2463.95	23990.350000000002
2015-01-11	1872.3000000000002	25862.65
2015-01-12	1919.0500000000002	27781.7
2015-01-13	2049.6000000000004	29831.300000000003

Q3 Determine the top 3 most ordered pizza types based on revenue for each pizza category.


```
select name, Revenue, rnk from
(select category,name,Revenue,
rank() over(partition by category order by Revenue desc) as rnk
from
(select
pizza_types.category,pizza_types.name,
sum(orders_details.quantity*pizzas.price) as Revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details on
orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where rnk<=3;
```

name	Revenue	rnk
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

FUTURE SCOPE OF PROJECT

The future scope for a project on "Pizza Sales Analysis Using SQL" can be expansive and multifaceted. Here are several potential directions for future development:

1. Advanced Analytics and Reporting:
Predictive Analytics:
- Use historical sales data to forecast future sales trends.

2. Dashboards and Visualization:
- Develop interactive dashboards using tools like Tableau, Power BI, or even SQL-based visualization libraries.

3. KPI Tracking:
- Define and monitor key performance indicators (KPIs) such as average order value, customer lifetime value, and churn rate.

4. Sentiment Analysis:
- Perform sentiment analysis on customer reviews and feedback to gauge customer satisfaction and identify areas for improvement.

5. Supply Chain Optimization:
- Analyze sales data to optimize supply chain and logistics operations, ensuring timely delivery and reducing costs.


Thank you!