
SQL PROJECT ON
PIZZA SALES

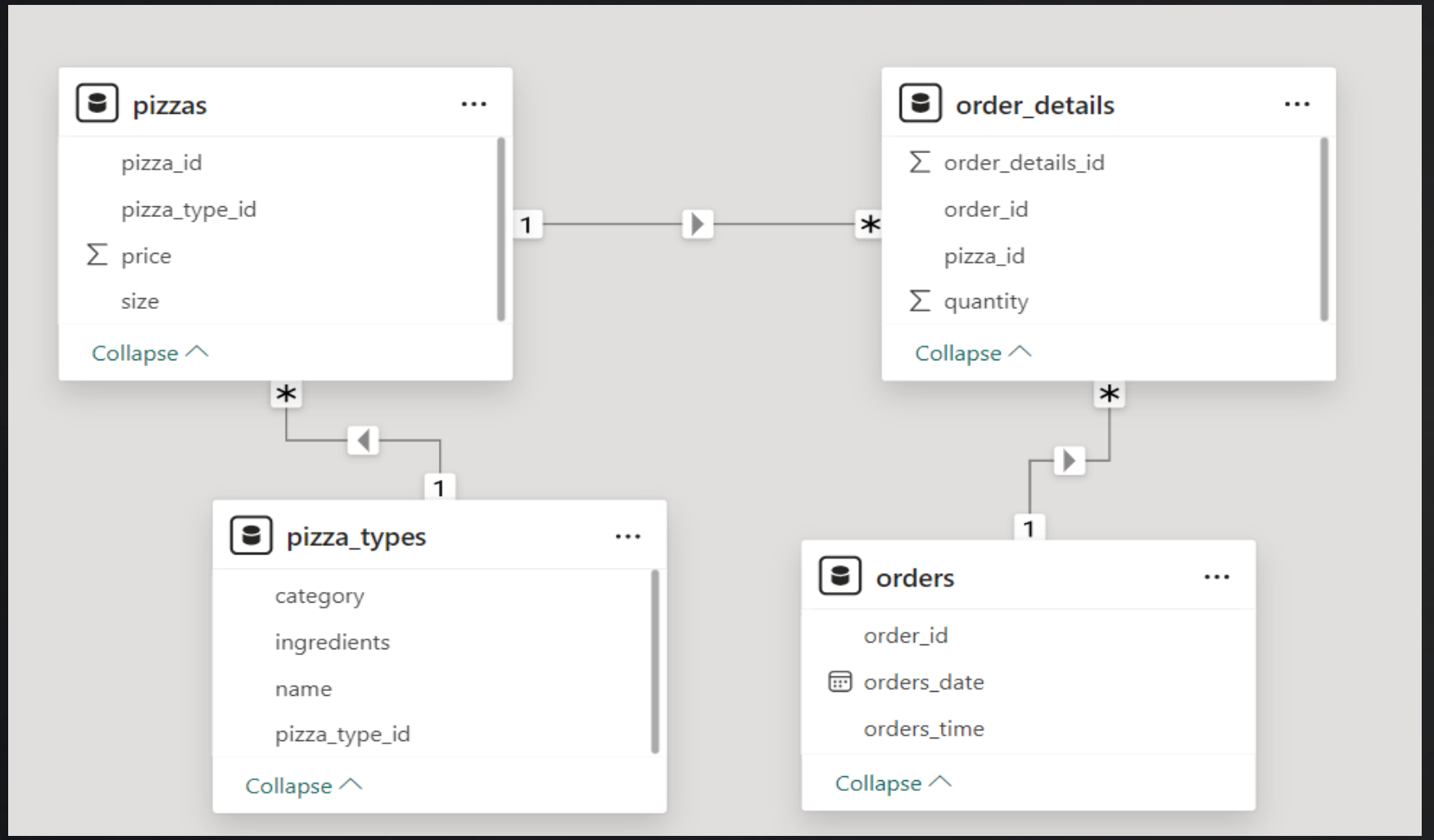
HELLO !

My name is Jayshree Raghuwanshi, In this project, I utilized SQL queries to extract, analyze, and visualize sales data. This included writing complex queries to track sales performance, identify top-selling products, and understand customer preferences, ultimately providing actionable insights for business optimization.

PROJECT OVERVIEW

The primary goal is to leverage advanced data analysis methodologies to extract actionable insights from an extensive database of pizza sales data. This analytical approach equips the business with the knowledge to make informed decisions, enabling strategic growth and a competitive edge in the pizza industry. By understanding customer preferences, sales trends, and operational efficiencies, we aim to drive effective strategies that maximize profitability and market share. This comprehensive analysis serves as a cornerstone for informed decision-making, fostering innovation and adaptability in a dynamic market landscape.

SCHEMA


pizzas
pizza_id
pizza_type_id
price
size
Collapse
order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
orders
order_id
orders_date
orders_time
Collapse
1
*
1
*
1
*
1

Queries

Retierve the total number of orders placed

```
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_Sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	Total_Sales
▶	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 Rows:

name	price
The Greek Pizza	35.95

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 order_details.pizza_id = pizzas.pizza_id
GROUP BY Pizzas.size
ORDER BY Order_Count DESC;
```

Result Grid | Filter

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

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

```
SELECT pizza_types.name, SUM(order_details.quantity) AS Quantities
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 Quantities DESC LIMIT 5;
```

Result Grid | Filter Rows:

name	Quantities
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.

```sql
SELECT
    HOUR(orders_time) AS Hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(orders_time);
```

Result Grid

Hour	order_count
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) AS Pizzas
FROM
    pizza_types
GROUP BY category;
```

Result Grid

	Category	Pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```sql
SELECT
    ROUND(AVG(quantiy), 0) AS Pizzas_order_per_day
FROM
    (SELECT
        orders.orders_date, SUM(order_details.quantity) AS quantiy
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.orders_date) AS Order_Quantity;
```

Result Grid | Filter R

	Pizzas_order_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 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 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 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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Revenue DESC;
```

Result Grid | Filter

category	Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```
SELECT orders_date, SUM(revenue) OVER(Order BY orders_date) AS Cumulative_Revenue
FROM
(SELECT orders.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 orders.order_id = order_details.order_id GROUP BY orders.orders_date) AS Sales;
```

Result Grid | Filter Rows:

orders_date	Cumulative_Revenue
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
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4

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

```sql
SELECT Category, Name, Revenue, Top_3_Rank
FROM
(SELECT Category, Name, Revenue,
rank() over(partition by category order by revenue desc) as Top_3_Rank
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 Top_3_Rank <=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

Category	Name	Revenue	Top_3_Rank
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3


THANK
YOU