



MySQL Project on Pizza sales

By Mohammad Asad

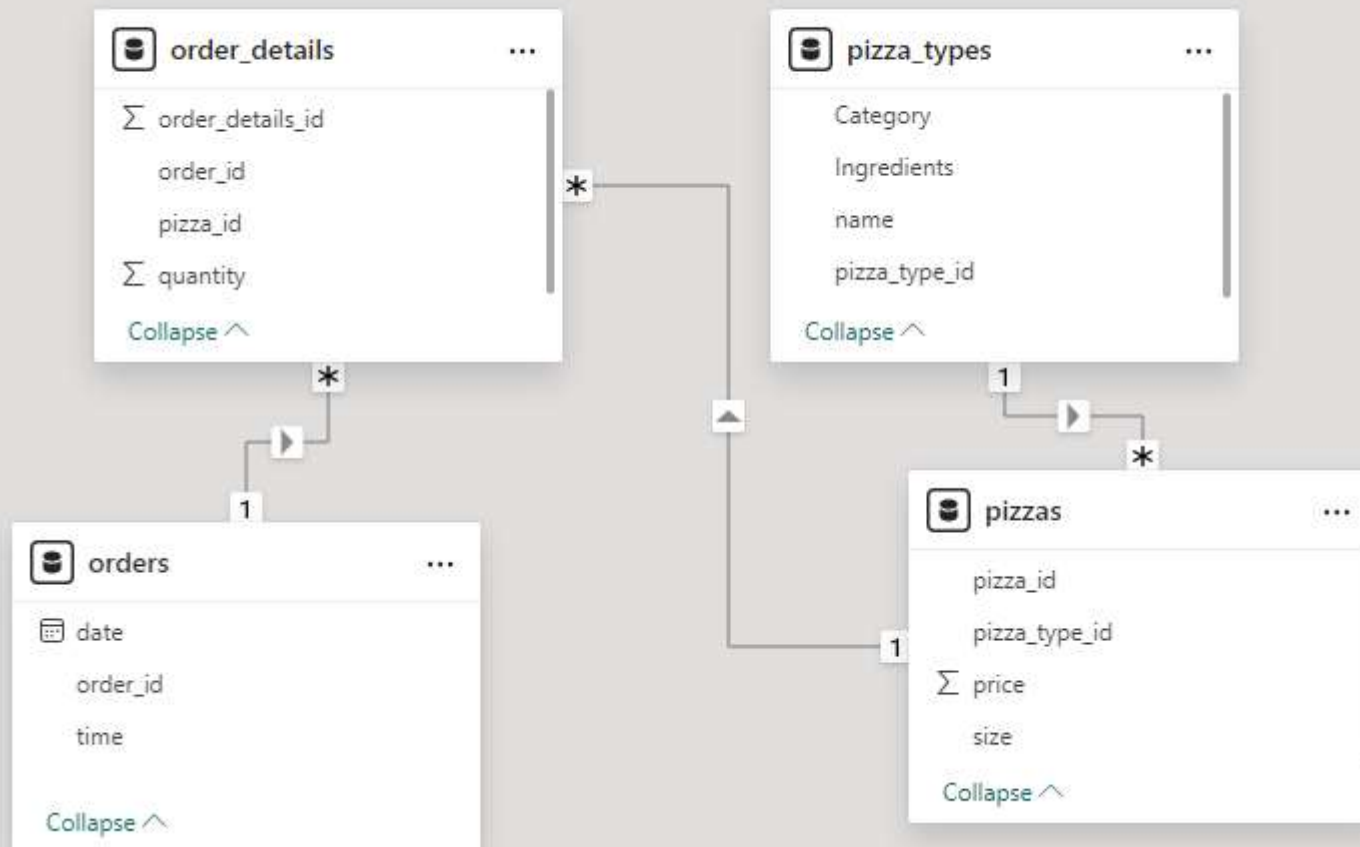
GitHub link - https://github.com/Asad19971/SQL_Pizza_Sales_Project

A decorative border surrounds the slide content. In the top-left corner, there is a whole red tomato, a yellow bell pepper, and a head of garlic. In the bottom-left corner, several slices of pizza with various toppings like olives, onions, and peppers are visible. In the bottom-right corner, there are mushrooms and a yellow bell pepper. The background is a dark, textured surface.

Overview

In this project I have utilized SQL queries to solve the questions related to pizza sales. This project typically involves analyzing data related to pizza orders, customers choice, Revenue and possibly preferred order time within a pizza restaurant or chain. The goal is to extract insights from the data using SQL queries to understand trends, customer preferences, popular items, sales performance. This project could encompass tasks such as querying sales data to identify top-selling pizzas, calculating revenue by time period, and generating reports for management or marketing purposes. Ultimately, the project aims to leverage SQL skills to drive business decisions and improve operational efficiency within the pizza sales domain.

DATA MODEL





SNAPSHOT OF QUESTIONS

Questions

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

01

1. Retrieve the total number of orders placed.

- ```
select count(order_id)
from orders;
```

| Result Grid |                 |
|-------------|-----------------|
|             | count(order_id) |
| ▶           | 21350           |





# 02

2. Calculate the total revenue generated from pizza sales..

- ```
select round(sum(quantity * price),2)
from orders_details OD inner join Pizzas P
on OD.pizza_id = P.pizza_id;
```

Result Grid



Filter Rows:

round(sum(quantity *
price),2)



817860.05



03

3. Identify the highest-priced pizza.

- ```
select name, price
from pizza_types PT Inner join pizzas P
on PT.pizza_type_id = p.pizza_type_id
order by price desc limit 1;
```

| Result Grid |                 |       | Filter Rows: |
|-------------|-----------------|-------|--------------|
|             | name            | price |              |
| ▶           | The Greek Pizza | 35.95 |              |





# 04

4. Identify the most common pizza size ordered.

- ```
select size, count(order_details_id) as OrderCount
from pizzas p inner join orders_details OD
on P.Pizza_id = OD.Pizza_id
group by size order by OrderCount desc limit 1;
```

Result Grid			Filter Rows:
	size	OrderCount	
▶	L	18526	



05

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

- ```
select name, sum(quantity) SumofQty
from pizzas p inner join orders_details OD
on P.Pizza_id = OD.Pizza_id
Inner Join pizza_types PT
on P.pizza_type_id = PT.pizza_type_id
group by name order by SumofQty desc limit 5;
```


| Result Grid |                            |          | Filter Rows: |
|-------------|----------------------------|----------|--------------|
|             | name                       | SumofQty |              |
| ▶           | The Classic Deluxe Pizza   | 2453     |              |
|             | The Barbecue Chicken Pizza | 2432     |              |
|             | The Hawaiian Pizza         | 2422     |              |
|             | The Pepperoni Pizza        | 2418     |              |
|             | The Thai Chicken Pizza     | 2371     |              |



# 06

6. Find the total quantity of each pizza category ordered.

- ```
select category, sum(quantity) SumofQty
from pizzas p inner join orders_details OD
on P.Pizza_id = OD.Pizza_id
Inner Join pizza_types PT
on P.pizza_type_id = PT.pizza_type_id
group by category;
```



Result Grid


	category	SumofQty
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050



07

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

- `Select Hour(order_time) as Hour, count(order_id) as count
from orders
group by Hour order by Hour;`



	Hour	count
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920



08

8. 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           |             |







# 09

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

- ```
select round(avg(SumofQty),0) AvgPizzaOrderedPerDay
from
  (Select Order_date, sum(quantity) SumofQty
  from orders O inner join orders_details OD
  on O.order_id = OD.Order_id
  Group by order_date) as Temptable;
```



Result Grid |  Filter Rows:


	AvgPizzaOrderedPerDay
▶	138





10

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

- ```
Select name, sum(Quantity * Price) as Revenue
From pizza_types PT Inner join Pizzas P
on PT.pizza_type_id = p.Pizza_type_id
Inner Join Orders_details OD
on OD.pizza_id = P.pizza_id
Group by 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  |






# 11

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

```
• Select Category, (sum(Quantity * Price) / (select round(sum(quantity * price),2)
 from orders_details OD inner join Pizzas P
 on OD.pizza_id = P.pizza_id))*100 as Revenue
 From pizza_types PT Inner join Pizzas P
 on PT.pizza_type_id = p.Pizza_type_id
 Inner Join Orders_details OD
 on OD.pizza_id = P.pizza_id
 Group by Category;
```



| Result Grid |                    | Filter Rows: |
|-------------|--------------------|--------------|
| Category    | Revenue            |              |
| Classic     | 26.90596025566967  |              |
| Veggie      | 23.682590927384577 |              |
| Supreme     | 25.45631126009862  |              |
| Chicken     | 23.955137556847287 |              |



# 12

Q12 Analyze the cumulative revenue generated over time.

```
• Select order_date,
 sum(Revenue) over(order by order_date) as CumulativeRevenue
from
(Select order_date, sum(quantity * price) as Revenue
from Orders_details OD inner join Pizzas P
on OD.pizza_id = p.pizza_id
Inner join Orders O
on O.order_id = OD.Order_id
Group by order_date) as TempTable;
```

| Result Grid |            |                    | Filter Rows: |
|-------------|------------|--------------------|--------------|
|             | order_date | CumulativeRevenue  |              |
| ▶           | 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           |              |

Result 14 x





# 13

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

```
• select * from
 (Select category, name, revenue,
 rank() over(partition by category order by revenue) as RankNo
 from
 (Select Category, name, sum(Quantity * Price) as Revenue
 from pizzas p inner join orders_details OD
 on P.Pizza_id = OD.Pizza_id
 Inner Join pizza_types PT
 on P.pizza_type_id = PT.pizza_type_id
 Group by category, name) as TempTable) as Temptable2
 where RankNo <= 3;
```

| Result Grid | Filter Rows:                               | Export:  | Wrap Cell Content: |
|-------------|--------------------------------------------|----------|--------------------|
| category    | name                                       | revenue  | RankNo             |
| Chicken     | The Chicken Pesto Pizza                    | 16701.75 | 1                  |
| Chicken     | The Chicken Alfredo Pizza                  | 16900.25 | 2                  |
| Chicken     | The Southwest Chicken Pizza                | 34705.75 | 3                  |
| Classic     | The Pepperoni, Mushroom, and Peppers Pizza | 18834.5  | 1                  |
| Classic     | The Big Meat Pizza                         | 22968    | 2                  |





*Based on the queries in previous slides, Management can take wise decisions regarding product Optimization, opening or closing time, How they can increase their revenue.*





THANK  
YOU