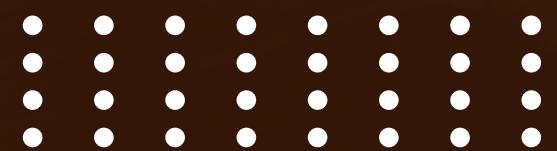


Pizza Sales Analysis

A Data-Driven Pizza Sales Project Using MYSQL

-Rakhi kumari



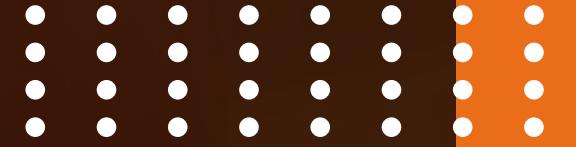


INTRODUCTION

Hi, I'm Rakhi Kumari. I completed a data-driven MySQL project on pizza sales, where I analyzed trends, cumulative sales, and customer behavior, revealing multiple insights that can guide strategic decisions and improve performance. The project helped me understand peak sales periods, popular pizza categories, and patterns in customer orders, while strengthening my skills in data analysis and MySQL.

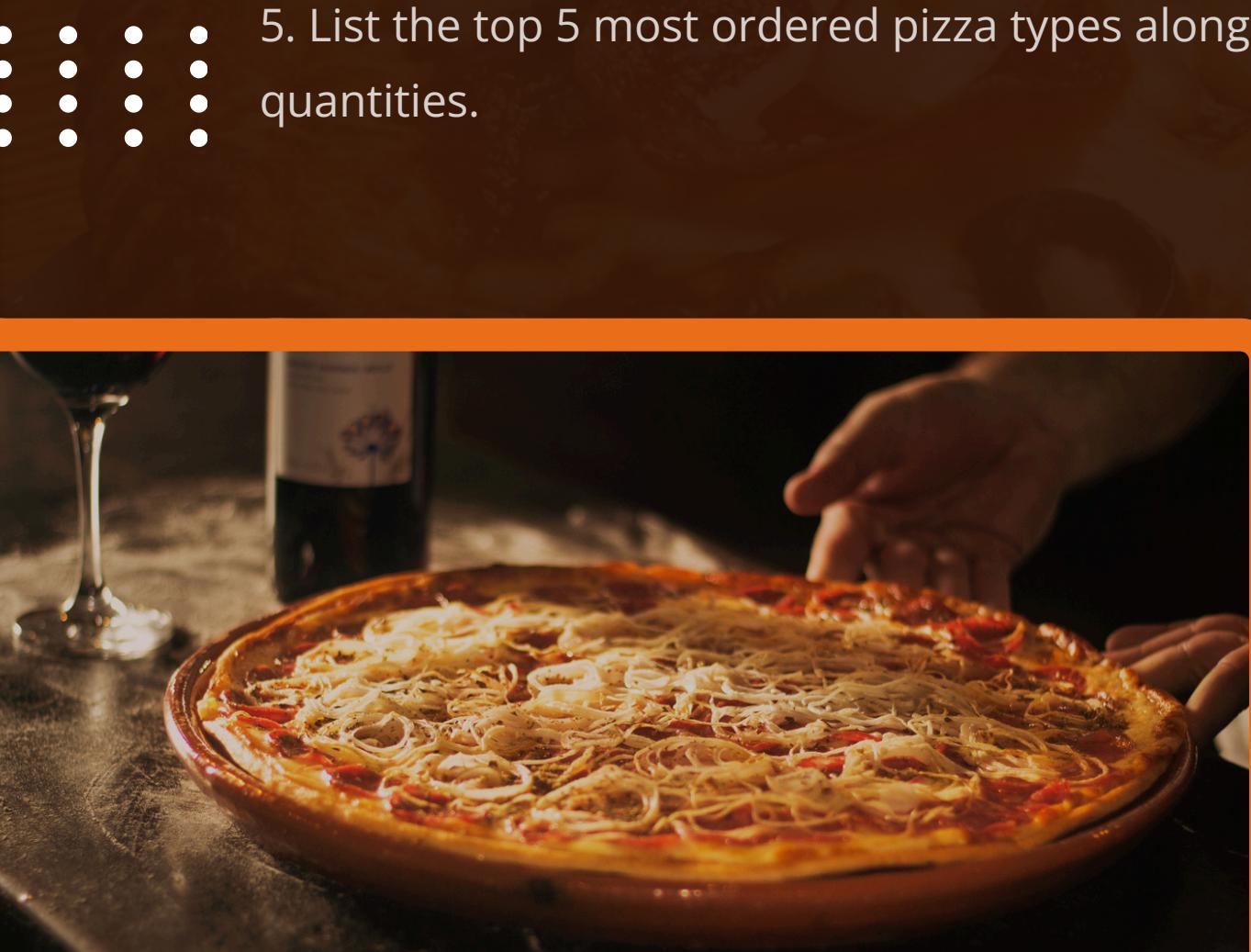


OBJECTIVES



BASIC

- 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.



INTERMEDIATE

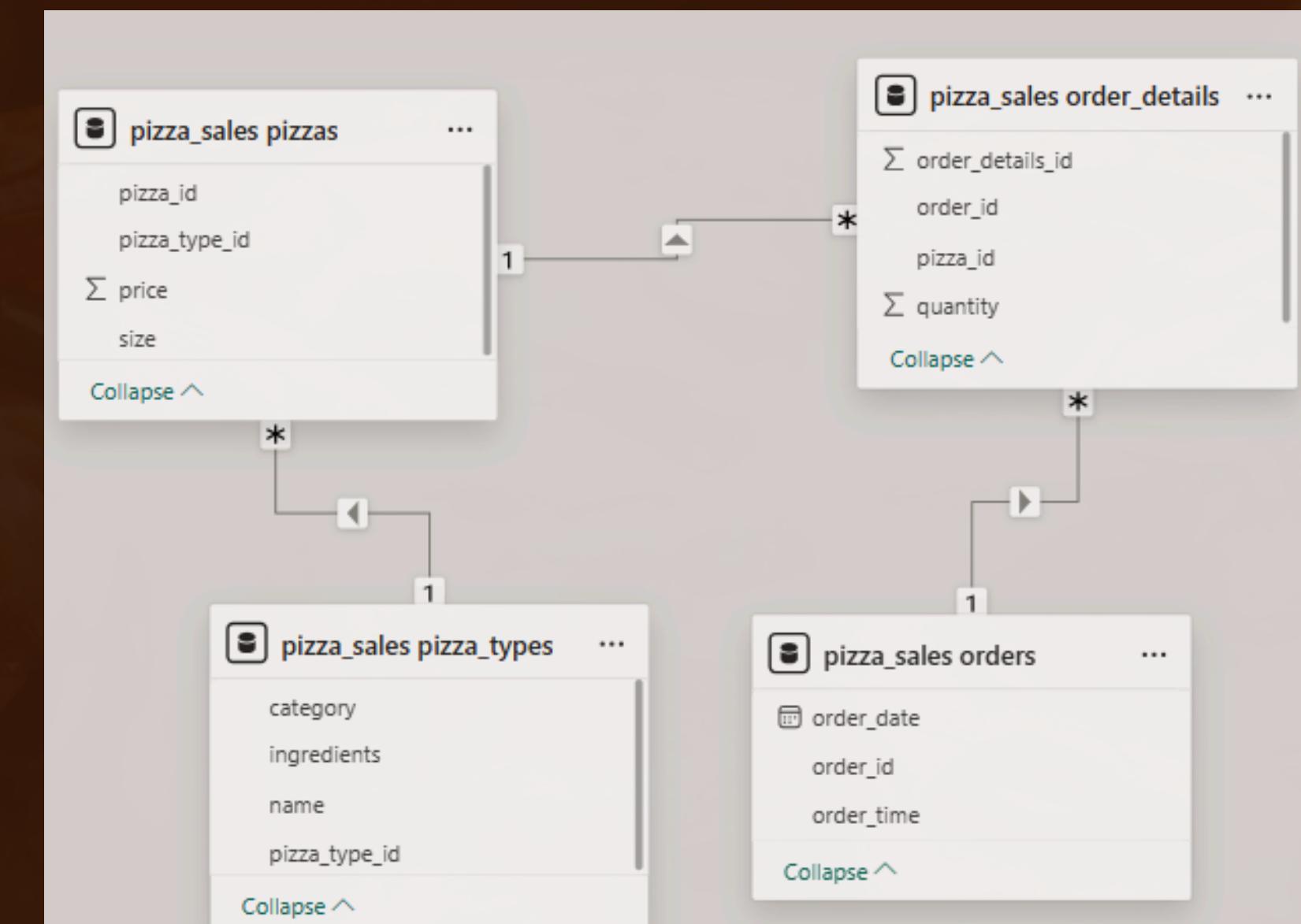
- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
- 7. Determine the distribution of orders by hour of the day.
- 8. Join relevant tables to find the category-wise distribution of pizzas.
- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
- 10. Determine the top 3 most ordered pizza types based on revenue.



ADVANCED

- 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.

Data Model



-- Retrieve 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.



```
SELECT  
    ROUND(SUM(p.price * od.quantity), 2) AS total_revenue  
FROM  
    pizzas p  
    JOIN  
    order_details od ON od.pizza_id = p.pizza_id
```

	Result Grid	grid icon
	total_revenue	
▶	817860.05	

-- Identify the highest-priced pizza.



```
SELECT  
    p.price, pt.name  
FROM  
    pizzas p  
        JOIN  
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id  
ORDER BY p.price DESC  
LIMIT 1;
```

Result Grid | Filter Row

	price	name
▶	35.95	The Greek Pizza

-- Identify the most common pizza size ordered. :::::



```
SELECT  
    size, COUNT(size) AS size_count  
FROM  
    pizzas  
GROUP BY size  
ORDER BY size_count DESC  
LIMIT 1;
```

Result Grid

size	size_count
S	32

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

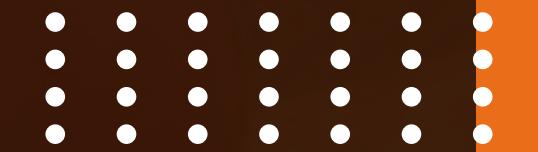
SELECT

```
pt.name, SUM(od.quantity) AS Quantity  
FROM  
pizzas p  
JOIN  
pizza_types pt ON pt.pizza_type_id = p.pizza_type_id  
JOIN  
order_details od ON od.pizza_id = p.pizza_id  
GROUP BY pt.name  
ORDER BY Quantity DESC  
LIMIT 5
```

Result Grid | Filter Rows:

	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

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



```
SELECT pt.category, SUM(od.quantity) AS total_quantity
FROM pizzas p
JOIN pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
```

Result Grid | Filter Rows:

	category	total_quantity
▶	Classic	14888
▶	Veggie	11649
▶	Supreme	11987
▶	Chicken	11050

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

SELECT

HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time)

hour	order_count
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) AS count  
FROM  
    pizza_types  
GROUP BY category
```

Result Grid |

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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



```
with as_order_quantity AS
(SELECT o.order_date, SUM(od.quantity) AS total_orders
 FROM orders o
 JOIN order_details od
 ON od.order_id = o.order_id
 GROUP BY o.order_date)

SELECT round(avg(total_orders),2) AS avg_pizza_ordered_per_day
FROM as_order_quantity
```

Result Grid		Filter Rows:
		avg_pizza_ordered_per_day
▶		138.47

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

```
SELECT pt.name, ROUND(SUM(quantity * price), 2) AS revenue
FROM order_details od
JOIN pizzas p ON p.pizza_id = od.pizza_id
JOIN pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

	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.



```
SELECT
    pt.category,
    concat(round(sum(quantity*price) / (SELECT
        ROUND(SUM(quantity * price), 2) AS revenue
    FROM
        order_details od
        JOIN
        pizzas p ON p.pizza_id = od.pizza_id) *100,2), "%") AS revenue
FROM
    order_details od
    JOIN
    pizzas p ON p.pizza_id = od.pizza_id
    JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY revenue DESC
```

	category	revenue
▶	Classic	26.91%
	Supreme	25.46%
	Chicken	23.96%
	Veggie	23.68%

-- Analyze the cumulative revenue generated over time.



```
WITH total_revenue AS
(SELECT
year(o.order_date) AS year, month(o.order_date) AS month,
sum(price*quantity) AS total_revenue
FROM order_details od
JOIN pizzas p
ON p.pizza_id = od.pizza_id
JOIN orders o
ON o.order_id = od.order_id
GROUP BY year(o.order_date),month(o.order_date))

SELECT concat(month,"/",year) as order_date,
round(sum(total_revenue) OVER(order by month),2) AS cumulative_revenue
FROM total_revenue
```

	order_date	cumulative_revenue
▶	1/2015	69793.3
	2/2015	134952.9
	3/2015	205350
	4/2015	274086.8
	5/2015	345489.55
	6/2015	413719.75
	7/2015	486277.65
	8/2015	554555.9
	9/2015	618735.95
	10/2015	682763.55
	11/2015	753158.9
	12/2015	817860.05

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



```
• SELECT
    pt.category,
    concat(round(sum(quantity*price) / (SELECT
        ROUND(SUM(quantity * price), 2) AS revenue
    FROM
        order_details od
        JOIN
            pizzas p ON p.pizza_id = od.pizza_id) *100,2),"%) AS revenue
FROM
    order_details od
    JOIN
        pizzas p ON p.pizza_id = od.pizza_id
        JOIN
            pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY revenue DESC
```

category	revenue
Classic	26.91%
Supreme	25.46%
Chicken	23.96%
Veggie	23.68%



Pizza Sales Presentation

**THANK YOU
FOR ATTENTION**

See You Next