



PIZZA SALES DATA ANALYSIS USING SQL



PROJECT CREATED BY -
KRUSHNA PAWAR

Tools Used: MySQL, Excel, DB
Browser

Date: July 2025



LARANA PIZZA

PROJECT OBJECTIVES

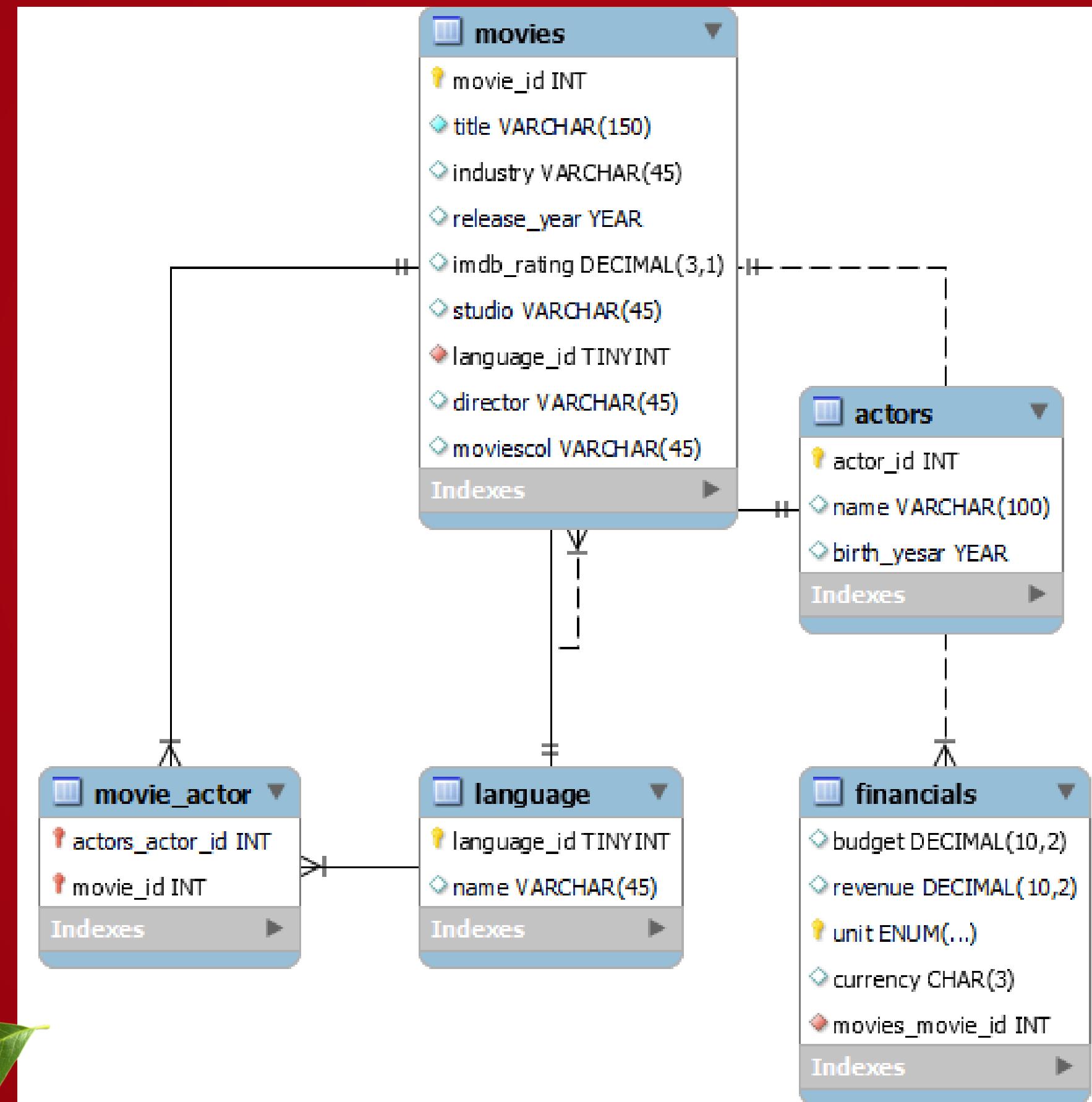
- Analyze pizza sales data using SQL.
- Solve real-world business questions using queries.
- Apply Joins, Aggregations, Window Functions, and Date Functions.
- Identify sales trends, top-performing pizzas, and order patterns.
- Generate insights to support data-driven business decisions.



DATASET DESCRIPTION

TABLE NAME	DESCRIPTION
PIZZAS -	PIZZA ID, SIZE, PRICE
PIZZA_TYPES -	PIZZA NAME, CATEGORY, INGREDIENTS
ORDERS -	ORDER ID, DATE, TIME
ORDERS_DETAILS -	PIZZA ID IN EACH ORDER + QUANTITY ORDERED

ER DIAGRAM / SCHEMA



BUSINESS QUESTIONS SOLVED

1. **What is the total number of orders placed?**
2. **Which is the highest-priced pizza?**
3. **What is the most common pizza size ordered?**
4. **What are the top 5 most ordered pizza types (by quantity)?**
5. **What is the percentage revenue contribution of each pizza type?**
6. **What are the top 3 pizzas by revenue in each category?**
- 7.
8. **What is the percentage revenue contribution of each pizza type?**
9. **What are the top 3 pizzas by revenue in each category?**
10. **What are the top 3 pizzas by revenue?**

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES?

SELECT

```
ROUND(SUM(orders_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

```
orders_details
```

JOIN

```
pizzas ON pizzas.pizza_id = orders_details.pizza_id
```



Result Grid

	total_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

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



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT  
    P.size, COUNT(OD.order_details_id)  
FROM  
    pizzas P  
    JOIN  
    orders_details OD ON P.pizza_id = OD.pizza_id  
GROUP BY P.size
```



Result Grid | Filter Rows:

	size	COUNT(OD.order_details_id)
▶	M	15385
L	18526	
S	14137	
XL	544	
XXL	28	

LIST TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT PT.name, OD.quantity AS most_orders  
FROM pizza_types PT  
JOIN pizzas P ON PT.pizza_type_id = P.pizza_type_id  
JOIN orders_details OD ON OD.pizza_id = P.pizza_id  
GROUP BY PT.name  
ORDER BY most_orders DESC  
LIMIT 5
```



Result Grid | Filter Rows:

	name	most_orders
▶	The Mediterranean Pizza	2
	The Hawaiian Pizza	1
	The Classic Deluxe Pizza	1
	The Five Cheese Pizza	1
	The Italian Supreme Pizza	1

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

SELECT

```
PT.name, SUM(OD.quantity * P.price) AS revenue  
FROM  
    pizzas P  
        JOIN  
    pizza_types PT ON P.pizza_type_id = PT.pizza_type_id  
        JOIN  
    orders_details OD ON OD.pizza_id = P.pizza_id  
GROUP BY PT.name  
ORDER BY revenue DESC  
LIMIT 3
```

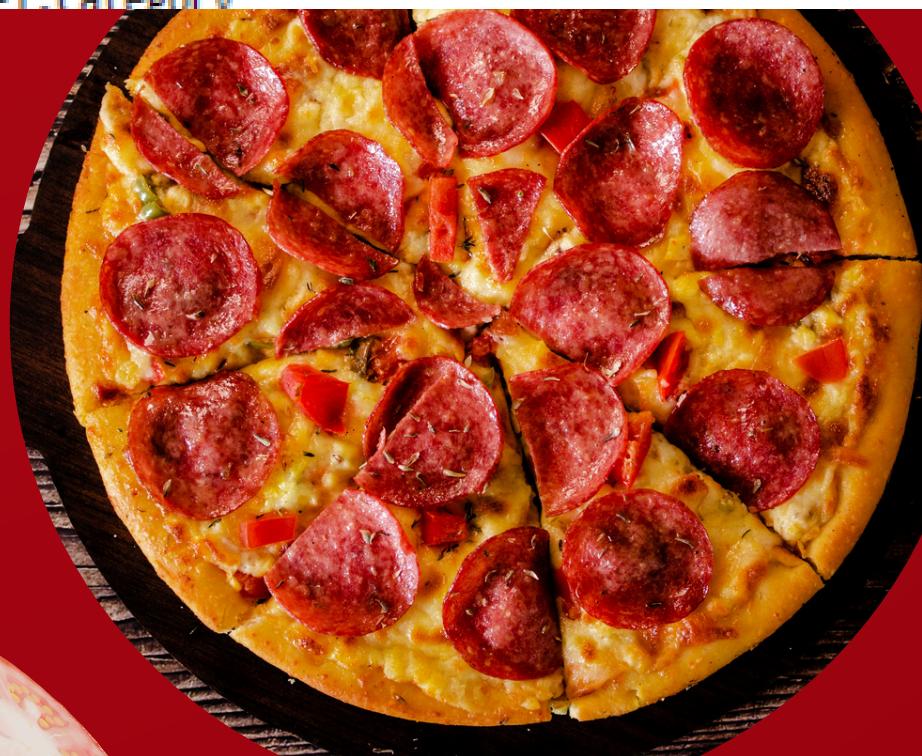


Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Pizza	The Barbecue Chicken Pizza

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
2
3 • SELECT
4     PT.category,
5     ROUND((SUM(OD.quantity * P.price) / (SELECT
6             ROUND(SUM(OD.quantity * P.price), 2) AS total_sales
7         FROM
8             orders_details OD
9             JOIN
10                pizzas P ON P.pizza_id = OD.pizza_id)) * 100,
11        2) AS revenue
12    FROM
13        pizzas P
14        JOIN
15            pizza_types PT ON P.pizza_type_id = PT.pizza_type_id
16        JOIN
17            orders_details OD ON p.pizza_id = OD.pizza_id
18    GROUP BY PT.category
```

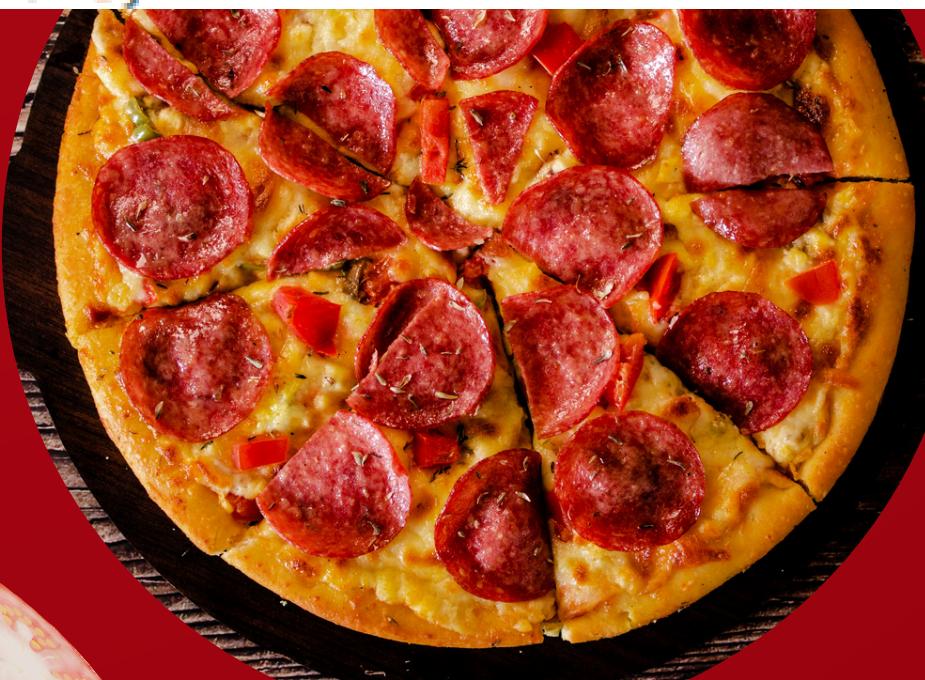


Result Grid | Filter Rows

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

DETERMINE THE TOP 3 ORDERED PIZZA TYPE BASED ON REVENUE FOR EACH PIZZA CATEGORY

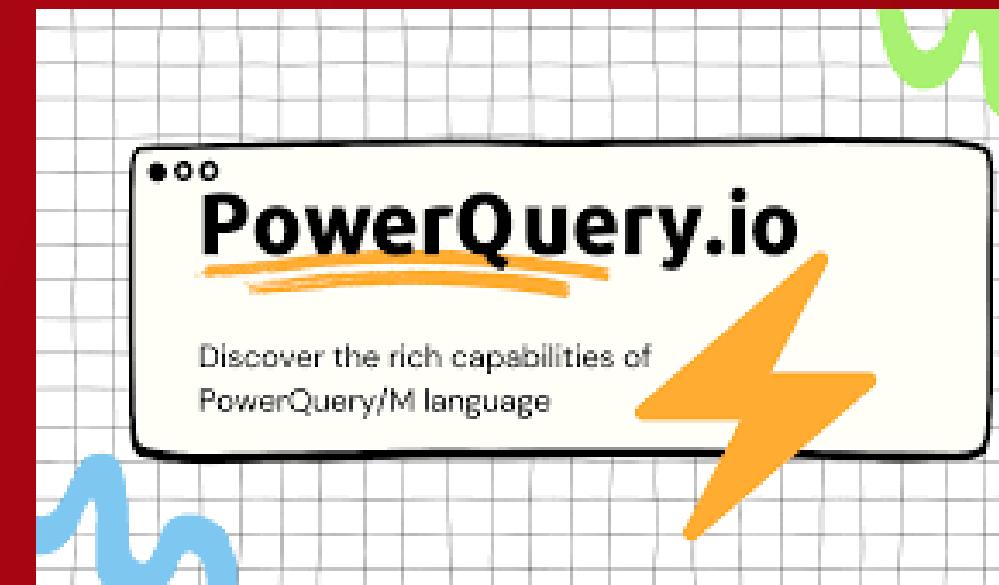
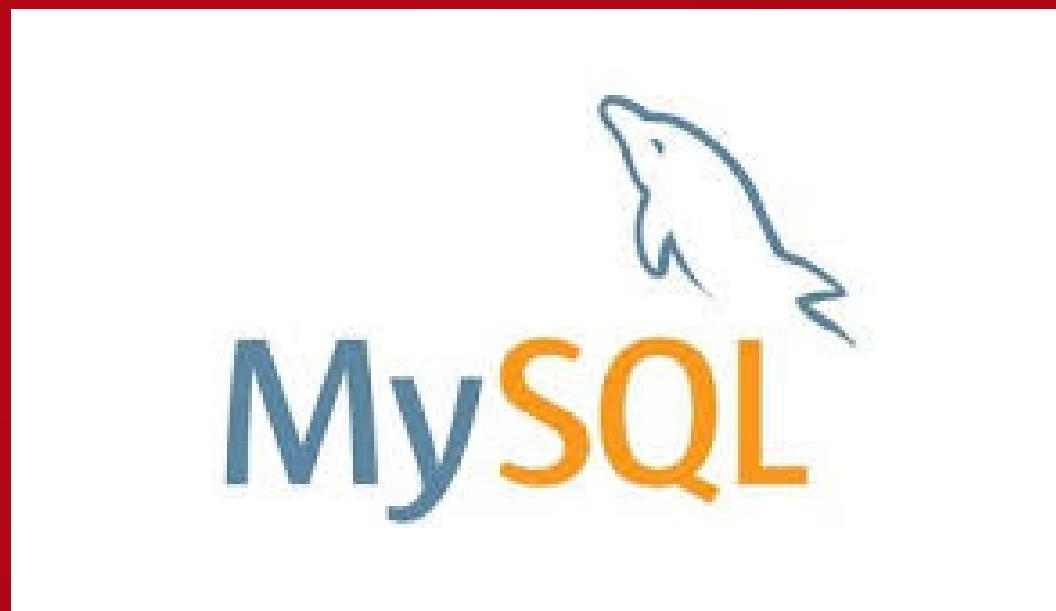
```
• • SELECT name, revenue  
    FROM (  
        SELECT category, name, revenue,  
              RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn  
        FROM (  
            SELECT  
                pizza_types.category,  
                pizza_types.name,  
                SUM(orders_details.quantity * 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 rn <= 3;
```



Result Grid | Filter Rows:

	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

BEST SELLETOOLS & SKILLS USED



- Skills & Concepts:**
- SQL Joins (INNER, LEFT)
 - Aggregations (SUM, COUNT, AVG)
 - GROUP BY and HAVING
 - Subqueries
 - Window Functions (RANK, OVER)
 - Date & Time Functions
 - Data Cleaning & Optimization
 - Business Insight Generation



LARANA PIZZA

THANK YOU!

