



# PIZZA SALES DATA ANALYSIS.

TASTE  
THE  
BEST  
FROM  
THE  
OVEN



NAME :- VARAD NAIK

MODULE :- SQL



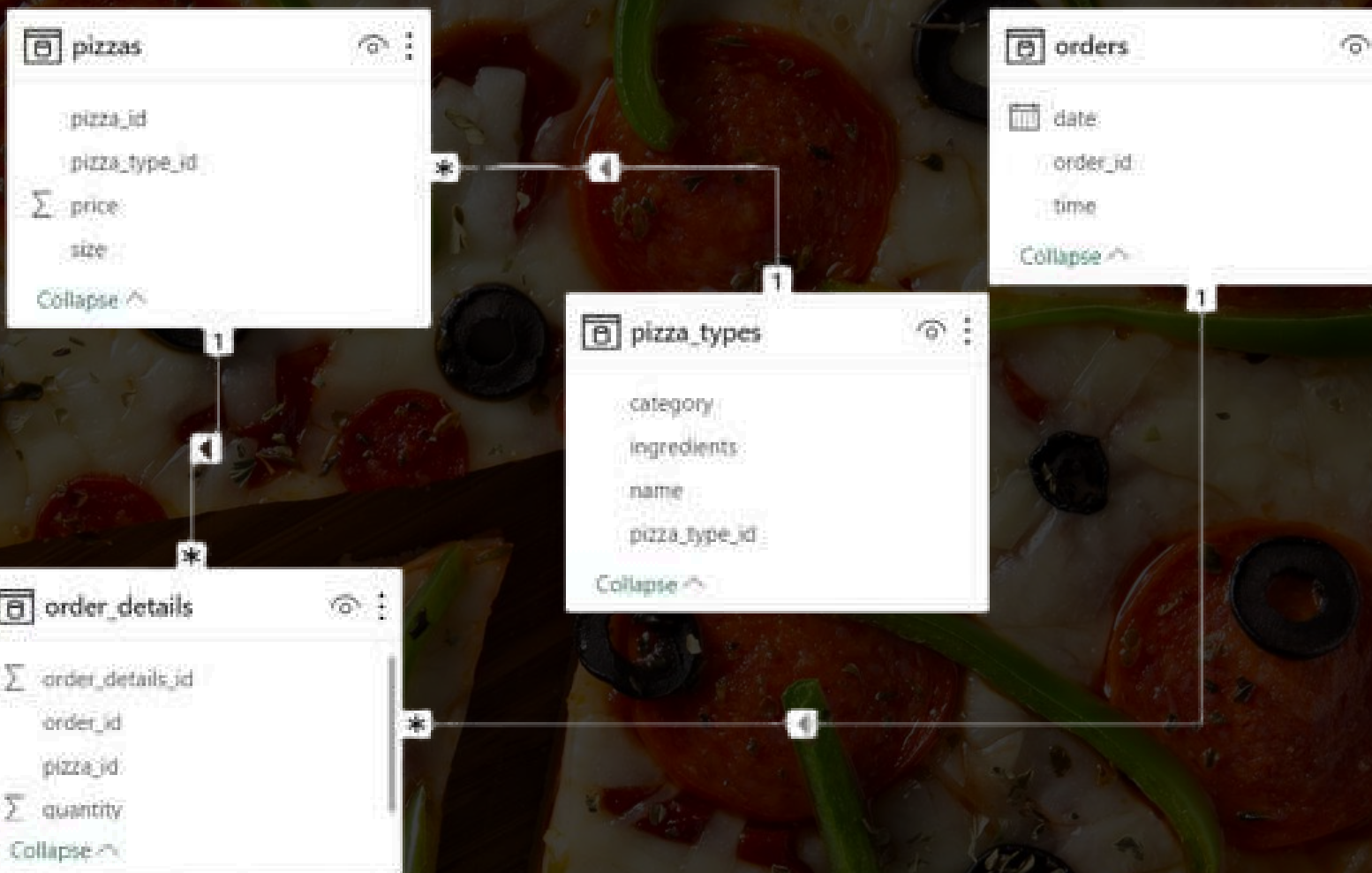


# **TOOLS USED**

- 1. MySQL Workbench 8.0 - For database management and SQL queries.**
- 2. Power BI - For creating the Entity Relationship Diagram (ERD) and data visualization.**
- 3. Canva Pro - For designing the presentation slides.**
- 4. GitHub - For project repository and version control.**
- 5. Kaggle - Source of the Pizza Sales Data dataset.**





# ER DIAGRAM





# TABLES


## PIZZAS

Result Grid |  Filter Rows:  | Export: 

	Field	Type	Null	Key	Default	Extra
	pizza_id	text	YES		NULL	
	pizza_type_id	text	YES		NULL	
	size	text	YES		NULL	
	price	double	YES		NULL	

## PIZZA\_TYPE

Result Grid




Filter Rows:


Export

	Field	Type	Null	Key	Default
▶	pizza_type_id	text	YES		NULL
	name	text	YES		NULL
	category	text	YES		NULL
	ingredients	text	YES		NULL

## ORDERS

Result Grid			Filter Rows:		
	Field	Type	Null	Key	Default
▶	order_id	int	NO	PRI	NULL
	order_date	date	NO		NULL
	order_time	time	NO		NULL

## ORDER DETAILS

Result Grid			Filter Rows:		Export
	Field	Type	Null	Key	Default
▶	order_id	int	NO		NULL
	order_details_id	int	NO	PRI	NULL
	pizza_id	text	NO		NULL
	quantity	int	NO		NULL





# SQL QUERIES

## - DDL (Data Definition Language):

- Database creation
- Table creation and constraints (e.g., primary keys, foreign keys)

## - DML (Data Manipulation Language):

### - **\*\*Sample Queries:\*\***

- **\*\*SELECT\*\***: Retrieve pizza sales data, customer details, etc.
- **\*\*INSERT\*\***: Add new pizza sales records.
- **\*\*UPDATE\*\***: Modify existing sales or customer information.

## - Joins and Subqueries:

- Example queries using **\*\*INNER JOIN\*\***, **\*\*LEFT JOIN\*\***, and subqueries for deeper analysis.

## - Other SQL Techniques:



- **LIKE\*\***: Used for pattern matching (e.g., finding pizzas with specific ingredients).
- **GROUP BY\*\***: Grouping data for aggregate functions like total sales by pizza type.
- **ORDER BY\*\***: Sorting the results (e.g., by highest sales).
- **ALTER** : Used to modify table structures (e.g., adding or altering columns).



# CREATE TABLE

```
create table orders(  
order_id int primary key not null,  
order_date date not null,  
order_time time not null);
```

O/P :- TABLE NAME - **ORDERS**

Result Grid    Filter Rows: <input type="text"/>   Export: 						
	Field	Type	Null	Key	Default	Extra
▶	order_id	int	NO	PRI	NULL	
	order_date	date	NO		NULL	
	order_time	time	NO		NULL	
	order_day	date	YES		NULL	



# CREATE TABLE

```
create table orders_details(  
order_id int not null,  
order_details_id int primary key not  
null,  
pizza_id text not null,  
quantity int not null);
```

O/P :- TABLE NAME - **ORDERS\_DETAILS**

Result Grid		Filter Rows:		Export:		
	Field	Type	Null	Key	Default	Extra
▶	order_id	int	NO		NULL	
	order_details_id	int	NO	PRI	NULL	
	pizza_id	text	NO		NULL	
	quantity	int	NO		NULL	



# ALTER COMMANDS

## 1. ADD :-

alter table orders add order\_day int;

O/P :-

	Field	Type	Null	Key	Default	Extra
▶	order_id	int	NO	PRI	NULL	
	order_date	date	NO		NULL	
	order_time	time	NO		NULL	
	order_day	date	YES		NULL	

## 2. MODIFY :-

alter table orders modify order\_day date;

O/P :-

	Field	Type	Null	Key	Default	Extra
▶	order_id	int	NO	PRI	NULL	
	order_date	date	NO		NULL	
	order_time	time	NO		NULL	
	order_day	date	YES		NULL	



# ALTER COMMANDS

## 3. CHANGE :-

alter table orders change order\_day weekdays  
date;

O/P :-

	order_id	order_date	order_time	weekdays
▶	1	2015-01-01	11:38:36	NULL
	2	2015-01-01	11:57:40	NULL
	3	2015-01-01	12:12:28	NULL
	4	2015-01-01	12:16:31	NULL
	5	2015-01-01	12:21:30	NULL
	6	2015-01-01	12:29:36	NULL
	7	2015-01-01	12:50:37	NULL
	8	2015-01-01	12:51:37	NULL

## 4. DROP

alter table orders modify order\_day date;

O/P :-

	order_id	order_date	order_time
▶	1	2015-01-01	2015-01-01
	2	2015-01-01	11:57:40
	3	2015-01-01	12:12:28
	4	2015-01-01	12:16:31
	5	2015-01-01	12:21:30
	6	2015-01-01	12:29:36
	7	2015-01-01	12:50:37
	8	2015-01-01	12:51:37



# ALTER COMMANDS

## 5. UPDATE :-

```
UPDATE pizza_types  
SET  
    name = 'The Medium Meat Pizza'  
WHERE  
    pizza_type_id = 'big_meat';
```

O/P :-

pizza_type_id	name	category	ingredients
thai_chn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	<u>The Medium Meat Pizza</u>	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe...
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese



# LIKE OPERATOR

1. *You need to retrieve all pizza types from the database that contain the word "Chicken" in their name.*

SELECT

\*




FROM

pizza\_types

WHERE

name LIKE '%Chicken%';

O/P :-

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content: 				
	pizza_type_id	name	category	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
	southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
	thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...



# LIKE OPERATOR

2. You need to retrieve all pizza types from the database that contain the word "VEG" in their name.

SELECT

\*





FROM

pizza\_types

WHERE

name LIKE '%Veg%';

O/P :-

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

	pizza_type_id	name	category	ingredients
	ital_veggie	The Italian Vegetables Pizza	Veggie	Eggplant, Artichokes, Tomatoes, Zucchini, Red ...
	veggie_veg	The Vegetables + Vegetables Pizza	Veggie	Mushrooms, Tomatoes, Red Peppers, Green Pe...



# ORDER BY

1. *list all pizza names and their categories from the pizza\_types table, sorted in ascending alphabetical order by name. Write an SQL query to achieve this.*

```
SELECT  
    name, category  
FROM  
    pizza_types  
ORDER BY name ASC;
```

O/P :-

Result Grid	Filter Rows:
name	category
The Barbecue Chicken Pizza	Chicken
The Brie Carre Pizza	Supreme
The Calabrese Pizza	Supreme
The California Chicken Pizza	Chicken
The Chicken Alfredo Pizza	Chicken
The Chicken Pesto Pizza	Chicken
The Classic Deluxe Pizza	Classic
The Five Cheese Pizza	Veggie
The Four Cheese Pizza	Veggie
The Greek Pizza	Classic
The Green Garden Pizza	Veggie
The Hawaiian Pizza	Classic



# ORDER BY

*2. You need to retrieve all pizza names and their categories, but this time sorted in descending alphabetical order by name. Write an SQL query for this.*

```
SELECT  
    name, category  
FROM  
    pizza_types  
ORDER BY name DESC;
```

O/P :-

Result Grid			Filter Rows:
	name	category	
▶	The Vegetables + Vegetables Pizza	Veggie	
	The Thai Chicken Pizza	Chicken	
	The Spinach Supreme Pizza	Supreme	
	The Spinach Pesto Pizza	Veggie	
	The Spinach and Feta Pizza	Veggie	
	The Spicy Italian Pizza	Supreme	
	The Southwest Chicken Pizza	Chicken	
	The Soppressata Pizza	Supreme	
	The Sicilian Pizza	Supreme	
	The Prosciutto and Arugula Pizza	Supreme	
	The Pepperoni, Mushroom, and Pe...	Classic	
	The Pepperoni Pizza	Classic	

pizza\_types 8 x



# GROUP BY

1. You want to find the maximum price of pizzas for each type and display the results for the top 3 pizza types.

```
SELECT  
    pizza_type_id, MAX(price) AS Max_Price  
FROM  
    pizzas  
GROUP BY pizza_type_id  
ORDER BY pizza_type_id  
LIMIT 3;
```

O/P :-

Result Grid			Filter Rows:
	pizza_type_id	Max_Price	
▶	bbq_ckn	20.75	
	big_meat	20.5	
	brie_carre	23.65	



# JOINS

*1. You want to retrieve the names, categories, sizes, and prices of pizzas by combining data from the pizza\_types and pizzas tables, even if some pizza types have no corresponding pizzas..*

```
SELECT  
    PT.name, PT.category, P.size, P.price  
FROM  
    pizza_types PT  
    LEFT JOIN  
    pizzas P ON PT.pizza_type_id =  
    P.pizza_type_id;
```

O/P :-

Result Grid		Filter Rows:		Export:	
	name	category	size	Price	
▶	The Barbecue Chicken Pizza	Chicken	L	20.75	
	The Barbecue Chicken Pizza	Chicken	M	16.75	
	The Barbecue Chicken Pizza	Chicken	S	12.75	
	The California Chicken Pizza	Chicken	L	20.75	
	The California Chicken Pizza	Chicken	M	16.75	
	The California Chicken Pizza	Chicken	S	12.75	
	The Chicken Alfredo Pizza	Chicken	L	20.75	
	The Chicken Alfredo Pizza	Chicken	M	16.75	
	The Chicken Alfredo Pizza	Chicken	S	12.75	
	The Chicken Pesto Pizza	Chicken	L	20.75	
	The Chicken Pesto Pizza	Chicken	M	16.75	
	The Chicken Pesto Pizza	Chicken	S	12.75	

Result 12 x



# JOINS

*2. You want to retrieve the names, categories, ingredients, sizes, and prices of pizzas by combining data from the **pizza\_types** and **pizzas** tables*

SELECT

PT.name, PT.category, PT.ingredients,  
P.size, P.price

FROM

pizza\_types PT

RIGHT JOIN

pizzas P ON PT.pizza\_type\_id =  
P.pizza\_type\_id;

O/P :-

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	name	category	ingredients	size	Price
▶	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...	S	12.75
	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...	M	16.75
	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...	L	20.75
	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	S	12.75
	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	M	16.75
	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	L	20.75
	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	S	12.75
	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	M	16.75
	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	L	20.75
	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...	S	12.75
	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...	M	16.75
	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...	L	20.75

Result 10





# 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.*



# QUESTIONS

1. *Retrieve the total number of orders placed*

```
SELECT  
  COUNT(order_id) AS Total_orders  
FROM  
  orders;
```

O/P :-

Result Grid	
	Total_orders
▶	21350



# QUESTIONS

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

SELECT

ROUND(SUM(O.quantity \* P.price), 2)

AS total

FROM

orders\_details O

JOIN

pizzas P ON P.pizza\_id = O.pizza\_id;

O/P :-

Result Grid	
	total
▶	817860.05



# QUESTIONS

**3. 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;
```

**O/P :-**

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



# QUESTIONS

4. *Identify the most common pizza size ordered.*

```
SELECT
    P.size, COUNT(O.order_details_id) AS
    Total
FROM
    pizzas P
    JOIN
    orders_details O ON P.pizza_id =
    O.pizza_id
GROUP BY size
ORDER BY total DESC;
```

O/P :-

Result Grid		
	size	Total
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



# CONCLUSION:

*The Pizza Sales Data Analysis project successfully utilized SQL to extract, analyze, and interpret valuable information from the pizza sales dataset. By leveraging SQL queries, such as JOINS, GROUP BY, ORDER BY, and aggregation functions, key insights were derived regarding sales trends, popular pizza types, and customer preferences.*

*The project demonstrated how database management and querying techniques can be used to:*

- Identify the best-selling pizzas and pricing strategies.*
- Optimize inventory and sales approaches based on data.*
- Enable businesses to make informed decisions and improve overall efficiency.*