

# **DATA-DRIVEN INSIGHTS: ANALYZING PIZZA SALES WITH SQL**

**Name : Bhanu**

**Registration No. : 24PIM4171**

**Course : M.S.(Pharmacoinformatics)**

**College : NIPER S.A.S Nagar**

The background features a large, stylized house-like shape composed of overlapping geometric polygons in shades of blue, grey, and red. Inside this shape, there are several icons representing technology and data: a server rack on the left, a smartphone with a green status bar at the top, and various bar charts and data visualizations. The word 'SQL' is prominently displayed in the center of the house shape.

# SQL

# HELLO

My name is Bhanu

In this project, I utilized the Pizza Sales dataset to solve a variety of SQL queries, focusing on sales trends, customer preferences, and operational insights.

The dataset includes multiple relational tables covering orders, pizzas, order details, and ingredients, allowing for in-depth analysis.

# BUSINESS QUESTIONS FROM PIZZA SALES

## Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

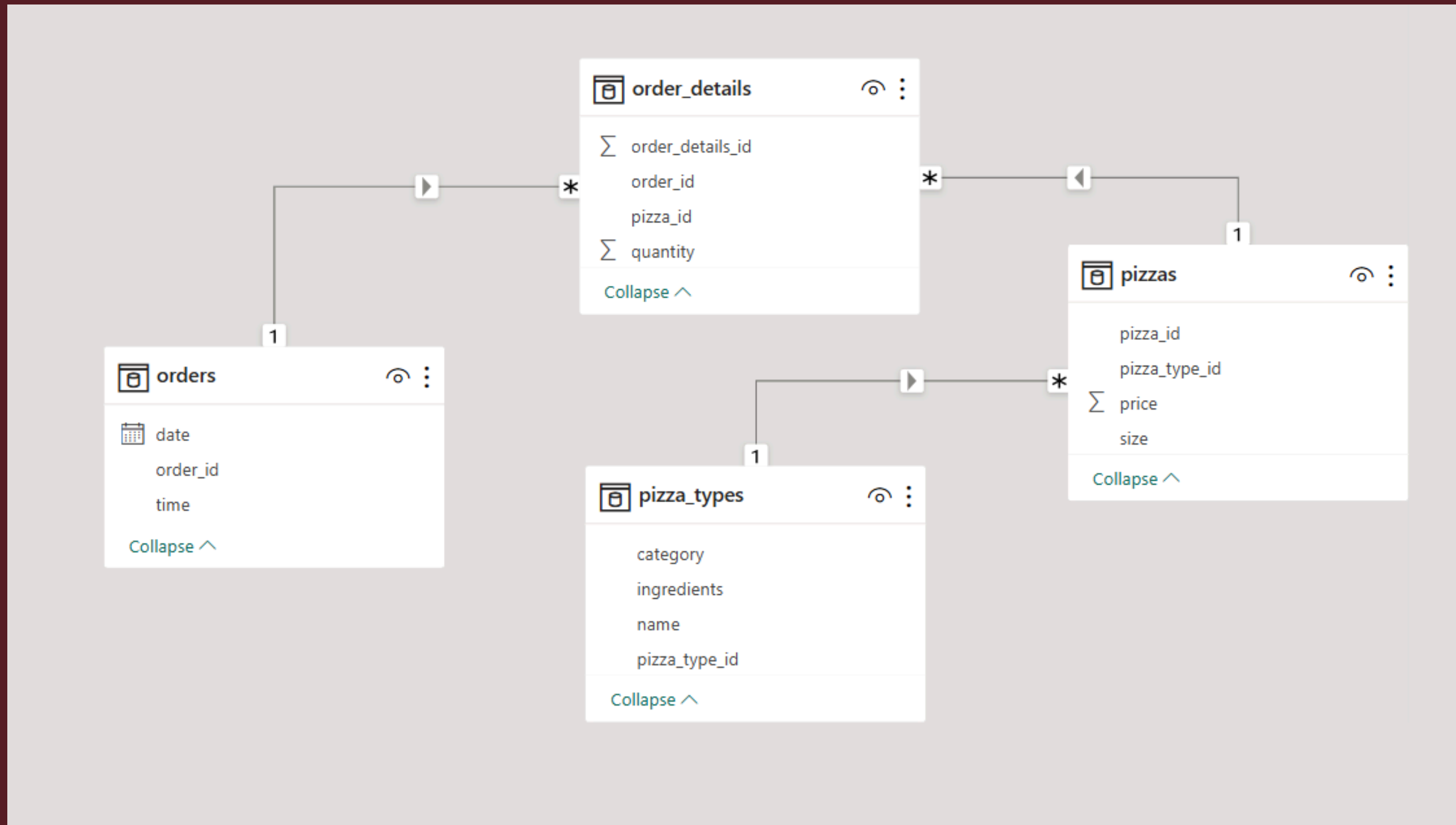
## Intermediate:

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

## Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# RELATIONAL DATA SCHEMA FOR PIZZA SALES ANALYSIS



# Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

**SELECT**

COUNT(order\_id) AS total\_orders

**FROM**

orders;

Result Grid	
	total_orders
▶	21350

# Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid		Filter
	total_sales	
▶	817860.05	

# Q3. IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid			Filter Rows
	price	name	
▶	35.95	The Greek Pizza	

# Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS total_order
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_order DESC;
```

Result Grid			Filter Rows:
	size	total_order	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	



# Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_ordered
FROM
    pizzas
    JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_ordered DESC
LIMIT 5;
```

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

# Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_count
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_count DESC;
```

Result Grid			Filter Rows:
	category	total_count	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

# Q7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    HOUR(orders.order_time) AS time,
    COUNT(order_details.order_id) AS Count
FROM
    orders
    JOIN
    order_details ON order_details.order_id = orders.order_id
GROUP BY time;
```

Result Grid		
	time	Count
	11	2672
	12	6543
	13	6203
	14	3521
	15	3170
	16	4185
	17	5143
	18	5359
	19	4350
	20	3487
	21	2528
	22	1370
	23	68
	10	17
▶	9	4

# Q8. JOIN RELEVANT TABLES TO 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	

# Q9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT
    AVG(quantity) as Average_order
FROM
    (SELECT
        DATE(orders.order_date) AS date_orders,
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY date_orders) AS total_orders;
```

Result Grid		Filter Rows:
	Average_order	
▶	138.4749	

# Q10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
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		

# Q11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 AS Percentage_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 Percentage_revenue;
```

Result Grid			Filter Rows:
	category	Percentage_revenue	
▶	Veggie	23.682590927384577	
	Chicken	23.955137556847287	
	Supreme	25.45631126009862	
	Classic	26.90596025566967	

# Q12. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES 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((order_details.quantity)*pizzas.price) AS revenue
FROM pizza_types JOIN pizzas
ON pizzas.pizza_type_id=pizza_types.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 rn <=3;
```

Result Grid			Filter Rows:	Ex
	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		
	The Pepperoni Pizza	30161.75		
	The Spicy Italian Pizza	34831.25		
	The Italian Supreme Pizza	33476.75		
	The Sicilian Pizza	30940.5		
	The Four Cheese Pizza	32265.70000000065		
	The Mexicana Pizza	26780.75		
	The Five Cheese Pizza	26066.5		



The image features a dark maroon background with several overlapping, semi-transparent hexagonal shapes of varying sizes and orientations. These shapes create a layered, geometric effect. Scattered across the background are several small, solid maroon hexagons. In the center, the words "THANK YOU" are written in a bold, white, sans-serif font.

**THANK YOU**