

# SQL-ICIOUS PIZZA SALES



CHEESE AND QUERIES



# INTRODUCTION

HELLO! I'M VARTIKA PARASHAR, AN ASPIRING DATA ANALYST SHOWCASING MY SQL SKILLS THROUGH THIS PROJECT.

OBJECTIVE-TO ANALYZE PIZZA SALES USING MYSQL AND DERIVE ACTIONABLE INSIGHTS FOR UNDERSTANDING CUSTOMER PREFERENCES, IDENTIFYING SALES PATTERNS, AND OPTIMIZING INVENTORY MANAGEMENT.

## KEY ACTIVITIES

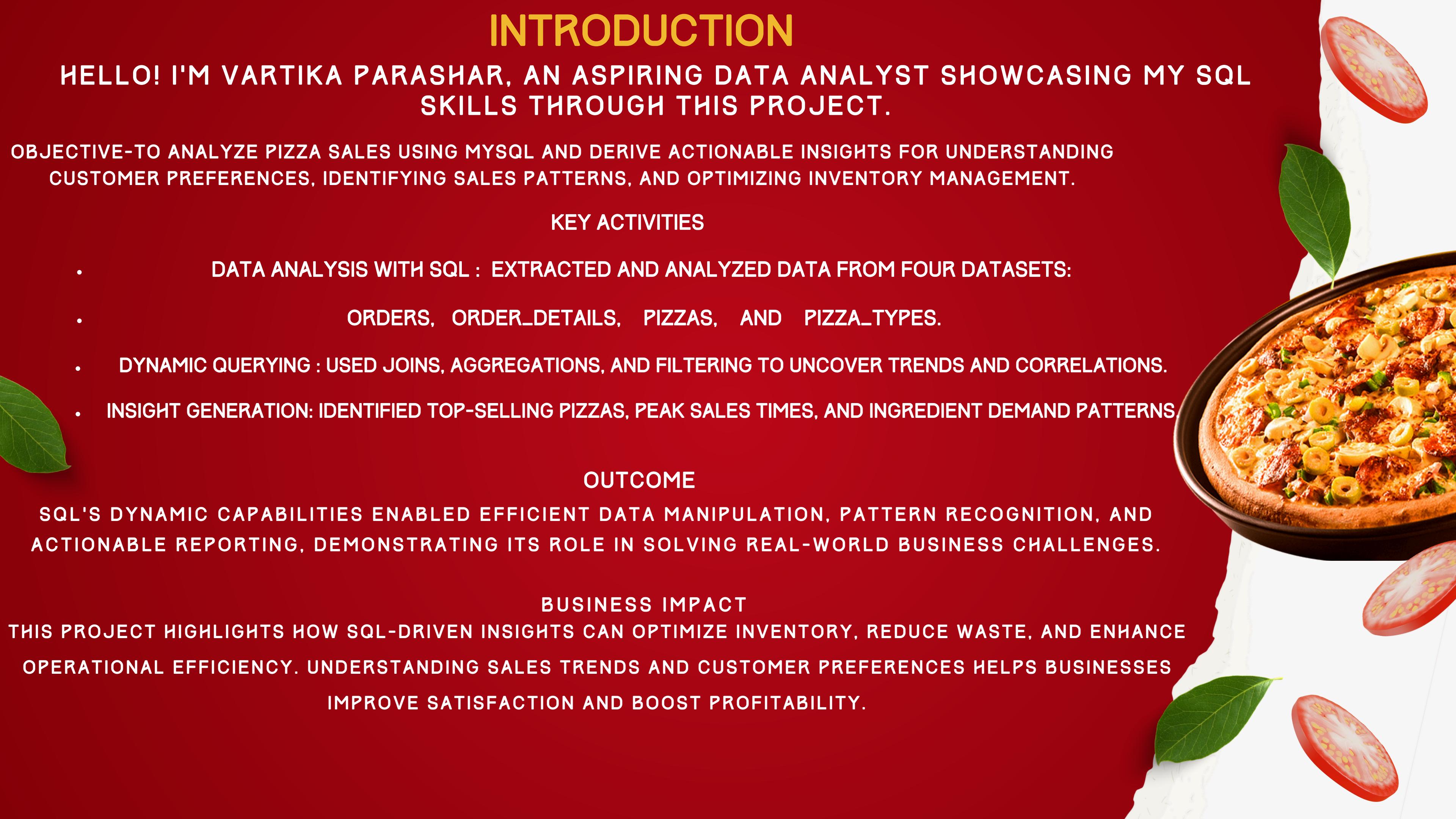
- DATA ANALYSIS WITH SQL : EXTRACTED AND ANALYZED DATA FROM FOUR DATASETS: ORDERS, ORDER\_DETAILS, PIZZAS, AND PIZZA\_TYPES.
- DYNAMIC QUERYING : USED JOINS, AGGREGATIONS, AND FILTERING TO UNCOVER TRENDS AND CORRELATIONS.
- INSIGHT GENERATION: IDENTIFIED TOP-SELLING PIZZAS, PEAK SALES TIMES, AND INGREDIENT DEMAND PATTERNS

## OUTCOME

SQL'S DYNAMIC CAPABILITIES ENABLED EFFICIENT DATA MANIPULATION, PATTERN RECOGNITION, AND ACTIONABLE REPORTING, DEMONSTRATING ITS ROLE IN SOLVING REAL-WORLD BUSINESS CHALLENGES.

## BUSINESS IMPACT

THIS PROJECT HIGHLIGHTS HOW SQL-DRIVEN INSIGHTS CAN OPTIMIZE INVENTORY, REDUCE WASTE, AND ENHANCE OPERATIONAL EFFICIENCY. UNDERSTANDING SALES TRENDS AND CUSTOMER PREFERENCES HELPS BUSINESSES IMPROVE SATISFACTION AND BOOST PROFITABILITY.



# QUESTIONS

## 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.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# BUSINESS INSIGHTS

- **Top-Selling Pizzas:**

The Thai Chicken Pizza, Barbecue Chicken Pizza, and California Chicken Pizza are the top 3 selling pizzas, generating the highest revenue.

- **Peak Sales Periods:**

Most orders occur during afternoon (**12–1 PM**) and evening (**6–7 PM**), marking the peak sales hours.

- **Total Pizza Category Ordered:**

The **Classic category** has the highest orders, outperforming other categories like Supreme, Veggie, and Chicken.

- **Revenue Contribution by Chicken Pizzas:**

The **Chicken category** generated the highest revenue, with its top 3 pizza types contributing approximately **23.96%**.

- **Total Orders and Revenue:**

A total of **21,350** orders were placed, generating **₹8,17,860** in revenue.

- **Average Pizzas Ordered Per Day:**

An average of **138** pizzas were ordered daily.

- **Most Preferred Pizza Size:**

**Large-sized** pizzas were the most preferred, with **18,526** units sold.

# 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(order_details.Quantity * pizzas.price),  
2) AS total_sales
```

**FROM**

```
order_details
```

**JOIN**

```
pizzas ON pizzas.pizza_id = order_details.Pizza_id;
```

Result Grid	
	<b>total_sales</b>
▶	817860.05

# IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid | Filter R

	name	price
▶	The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    pizzas.size,  
    COUNT(order_details.Order_Details_id) AS order_count  
FROM  
    pizzas  
        JOIN  
    order_details ON pizzas.Pizza_id = order_details.Pizza_id  
GROUP BY pizzas.size  
ORDER BY order_count DESC;
```

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
SELECT pizza_types.name, SUM(order_details.Quantity) AS quantity  
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 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  
    pizza_types.category,  
    SUM(order_details.Quantity) AS quantity  
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 quantity DESC;
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
SELECT  
    HOUR(Order_time), COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	HOUR(Order_time)	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	

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

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.Quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.Order_id = order_details.Order_id  
    GROUP BY orders.Order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizza_ordered_per_day
▶	138

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# 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 pizzas.pizza_type_id = pizza_types.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
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

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

```
select pizza_types.category,  
round(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) * 100,2) 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.category order by revenue desc;
```

Result Grid |

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT order_date,  
       sum(revenue) over (order by order_date) as cum_revenue  
  from  
(select orders.Order_date,  
           sum(order_details.Quantity * pizzas.price) as revenue  
      from order_details join pizzas  
        on order_details.Pizza_id = pizzas.pizza_id  
     join orders  
       on orders.Order_id = order_details.Order_id  
   group by orders.order_date) as sales;
```

	order_date	cum_revenue
1	2015-01-01	2713.8500000000004
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.350000000002
11	2015-01-11	25862.65

# 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 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, pizza_types.name) as a) as b  
where rn <= 3;
```

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



CHEESE AND QUERIES

# THANK YOU!

