

Where Every Slice is a Taste of Perfection

# WELCOME TO PIZZA RESTO SALES

Start Your Slide

START  
NOW





## HELLO!!

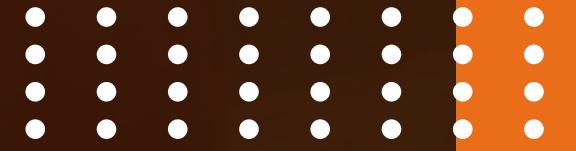
Hi, I am Vanshika, currently in my 3rd year pursuing B.Sc. Statistics (Hons.) from Ramjas College, University of Delhi. I am a passionate data enthusiast with a keen interest in exploring how data can be transformed into meaningful insights.

For this project, I worked on a Pizza Sales Dataset to perform data analysis using SQL. I wrote a variety of SQL queries, including both basic and complex queries, to extract valuable information from the dataset. These queries helped me analyze sales trends, customer preferences, and performance metrics.

Through this project, I aimed to strengthen my skills in SQL for data analysis while applying statistical and analytical thinking to real-world data.



# BRIEF ABOUT DATASET



## DATASETS COVERAGE

- Covers 21,350 orders with 48,620 individual items, meaning on average ~2.3 pizzas per order.
- Each pizza type comes in multiple sizes (S, M, L, XL, XXL), making 96 unique pizza & size combos from 32 core recipes.

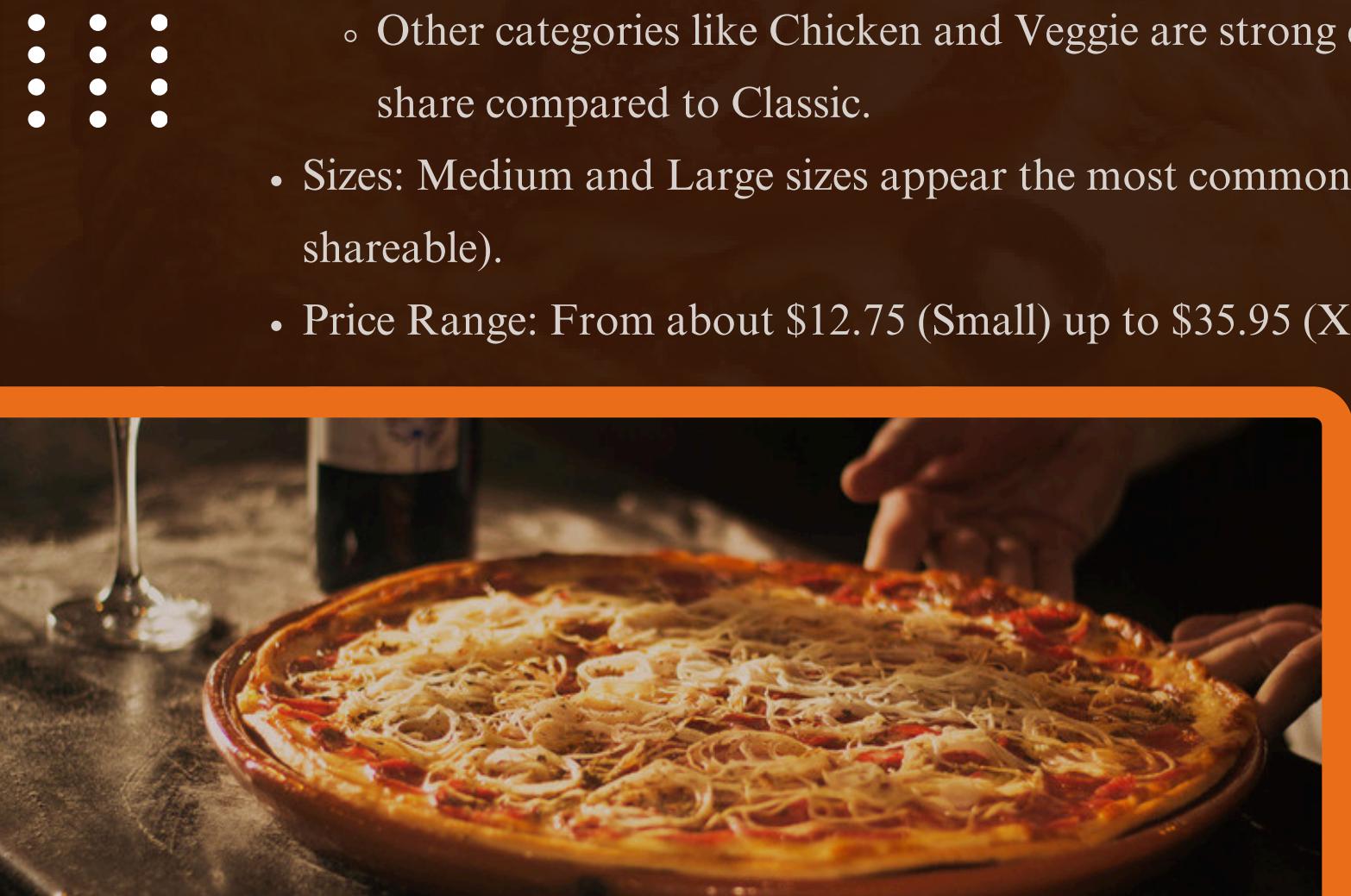
## 🍕 PRODUCT INSIGHTS

- Pizza Categories (32 types): Chicken, Classic, Supreme, Veggie...
  - Classic pizzas dominate sales volume.
  - Other categories like Chicken and Veggie are strong contributors but smaller share compared to Classic.
- Sizes: Medium and Large sizes appear the most common (higher affordability + shareable).
- Price Range: From about \$12.75 (Small) up to \$35.95 (XXL).



## 💰 SALES PERFORMANCE

- Revenue: ~\$818K total.
- Volume: ~49.6K pizzas sold.
- Average Order Value (AOV): about \$38.30 per order.
- Revenue Drivers:
  - Large & XL pizzas generate most of the revenue.
  - Classics bring in both high volume and strong revenue share.





# Q1) RETRIEVE THE TOTAL NUMBER OF ORDER PLACES.



```
SELECT
```

```
    COUNT(Order_id) AS Total_orders  
FROM orders;
```

Result Grid	
	Total_orders
▶	21350



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

```
SELECT
```

```
    SUM(order_details.quantity * pizzas.price) as total_revenue  
FROM order_details  
JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.049999993



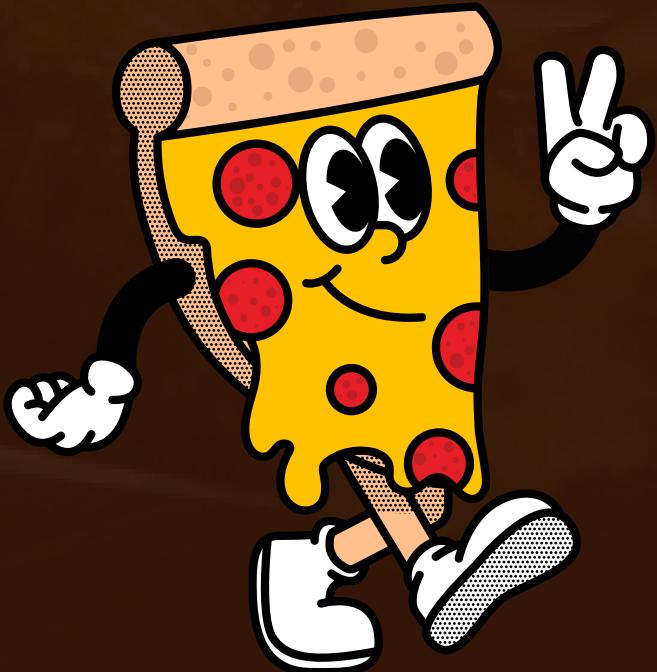
## Q4) IDENTIFY THE HIGHEST-PRICED PIZZA.



```
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 Rows
	name	price
▶	The Greek Pizza	35.95

## Q5) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



```
SELECT pizzas.size , count(order_details.order_details_id) as most_common_pizza  
From pizzas  
Join order_details  
    On pizzas.pizza_id = order_details.pizza_id  
Group by pizzas.size  
order by count(order_details.order_details_id) desc  
limit 1;
```

Result Grid		Filter Rows
	size	most_common_pizza
▶	L	18526

## Q5) 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		
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



Margherita Pizza

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

SELECT

```
    SUM(order_details.quantity) as total_quantity , pizza_types.category
```

```
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 total_quantity;
```

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



Veggie Delight

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

```
SELECT  
    HOUR(order_time), count(order_id)  
FROM orders  
GROUP BY  
    HOUR (order_time)  
ORDER BY  
    COUNT(order_id) desc;
```

	HOUR(order_time)	count(order_id)
▶	12	2520
	13	2455
	18	2399
	17	2336 2399
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1



Margherita Pizza

## Q8) 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



Veggie Delight

## Q8) 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
 LIMIT 3;
```

Result Grid		
	category	revenue
	Classic	26.91
	Supreme	25.46
	Chicken	23.96

## Q9) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
SELECT order_date,
       SUM(total_revenue) over(order by order_date )as cum_rev
  FROM
  (SELECT orders.order_date,
         SUM(order_details.quantity * pizzas.price) as total_revenue
    FROM orders
    JOIN order_details
    ON orders.order_id = order_details.order_id
    JOIN pizzas
    ON pizzas.pizza_id = order_details.pizza_id
   Group by orders.order_date) as sales;
```

Result Grid	
order_date	cum_rev
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.60000000006
2015-01-19	43365.75000000001

Q11) Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT * FROM orders;
SELECT AVG(Quantity)
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	
	AVG(Quantity)
▶	138.4749



Q12) Determine the top 3 most ordered pizza types based on revenue.

```
SELECT pizza_types.name ,
       SUM(order_details.quantity * pizzas.price) as total_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 total_revenue desc
LIMIT 3;
```

Result Grid	
name	total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



# CONCLUSION



## Sales Analysis - Key Takeaways :

- 21,350 Orders | 48,620 Pizzas | \$817K Revenue
- Classic pizzas are the top-selling category
- Medium & Large sizes most ordered; Large & XL sizes drive revenue
  - Customers spend ~\$38 per order (2–3 pizzas on average)
  - Peak sales: Lunch, Dinner & Weekends
  - Growth Opportunities:
    - Upsell to larger sizes
    - Bundle offers for 2–3 pizzas
    - Targeted weekend & peak-hour promotions

Focusing on best-selling classics and upselling larger sizes during peak hours can maximize revenue and customer satisfaction.

Pizza Resto Presentation

**THANK YOU  
FOR ATTENTION**

See You Next

vanshikakr2004@gmail.com