

# PIZZA SALES PRESENTATION

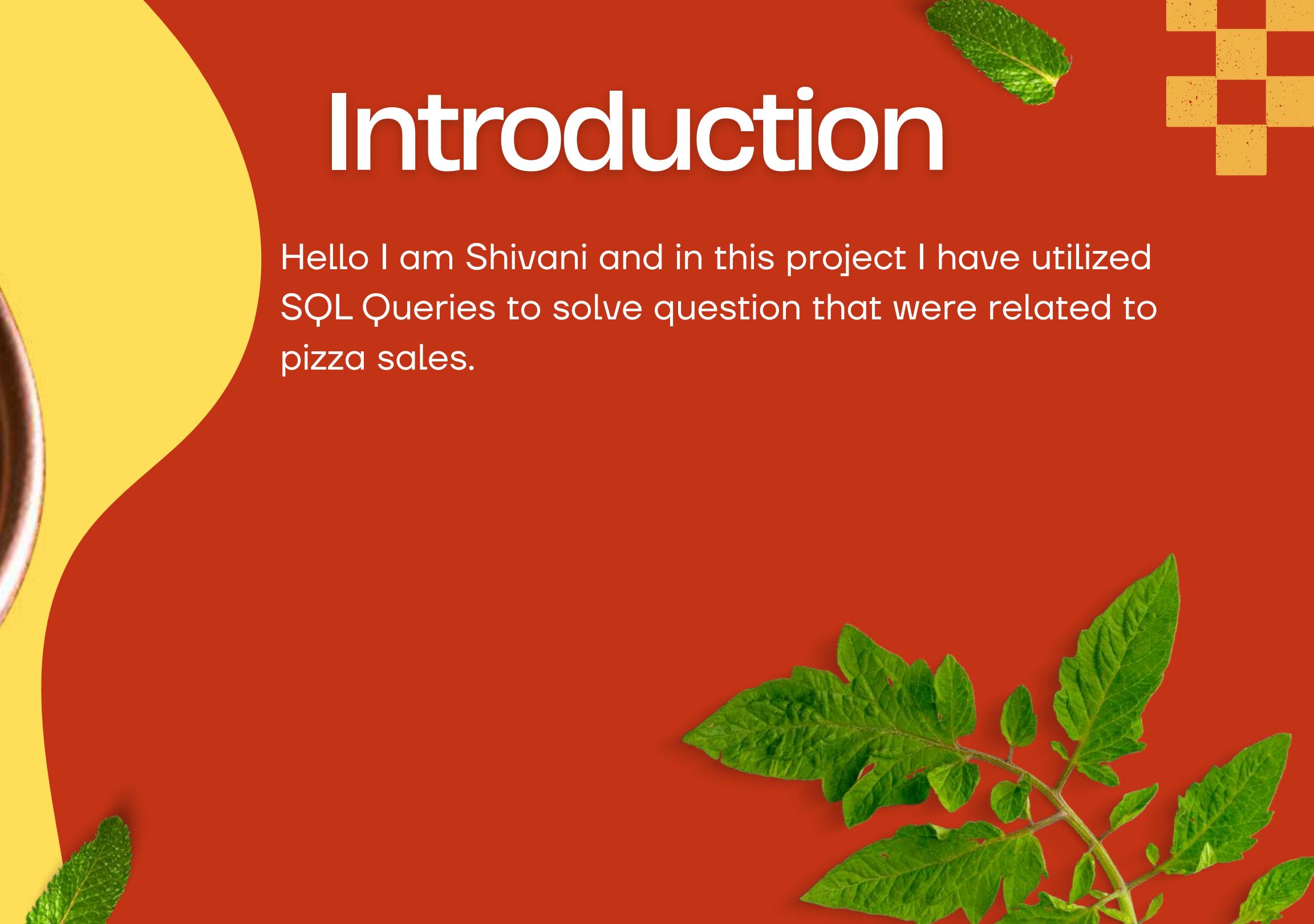
Presented By:

**Shivani Yadav**





# Introduction



Hello I am Shivani and in this project I have utilized SQL Queries to solve question that were related to pizza sales.

# Questions list

- 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.
- 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 pizza ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- 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.



# Retrive the total number of order 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

total_sales
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 Rows:

	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  
LIMIT 1;
```

Result Grid		Filter
	size	order_count
▶	L	18526

List the top 5 most ordered pizza type along with their quantity

```
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
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# Determine the distribution of orders by hour of the day

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

	Hour	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

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)
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



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

```
SELECT  
    ROUND(AVG(quantity), 0) AS avg_pizzaordered  
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   目录	
	avg_pizzaordered
	138

# 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		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	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;
```

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	order_date	cum_revenue	order_date	cum_revenue
2015-01-01	2713.8500000000004	2015-01-18	40978.600000000006	2015-02-04	80159.80000000002
2015-01-02	5445.75	2015-01-19	43365.75000000001	2015-02-05	82375.60000000002
2015-01-03	8108.15	2015-01-20	45763.65000000001	2015-02-06	84885.55000000002
2015-01-04	9863.6	2015-01-21	47804.20000000001	2015-02-07	87123.20000000001
2015-01-05	11929.55	2015-01-22	50300.90000000001	2015-02-08	89158.20000000001
2015-01-06	14358.5	2015-01-23	52724.60000000006	2015-02-09	91353.55000000002
2015-01-07	16560.7	2015-01-24	55013.85000000006	2015-02-10	93410.05000000002
2015-01-08	19399.05	2015-01-25	56631.40000000001	2015-02-11	95870.05000000002
2015-01-09	21526.4	2015-01-26	58515.80000000001	2015-02-12	98028.85000000002
2015-01-10	23990.35000000002	2015-01-27	61043.85000000001	2015-02-13	100783.35000000002
2015-01-11	25862.65	2015-01-28	63059.85000000001	2015-02-14	103102.50000000001
2015-01-12	27781.7	2015-01-29	65105.15000000016	2015-02-15	105243.75000000001
2015-01-13	29831.30000000003	2015-01-30	67375.45000000001	2015-02-16	107212.55000000002
2015-01-14	32358.70000000004	2015-01-31	69793.30000000002	2015-02-17	109334.45000000001
2015-01-15	34343.50000000001	2015-02-01	72982.50000000001	2015-02-18	111977.30000000002
2015-01-16	36937.65000000001	2015-02-02	75311.10000000002	2015-02-19	114007.55000000002
2015-01-17	39001.75000000001	2015-02-03	77925.90000000002	2015-02-20	116898.70000000001

# 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 ct) 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

# THANK YOU

