

SQL

PIZZA SALES PROJECT

Presented by:

Rashmi Sharma

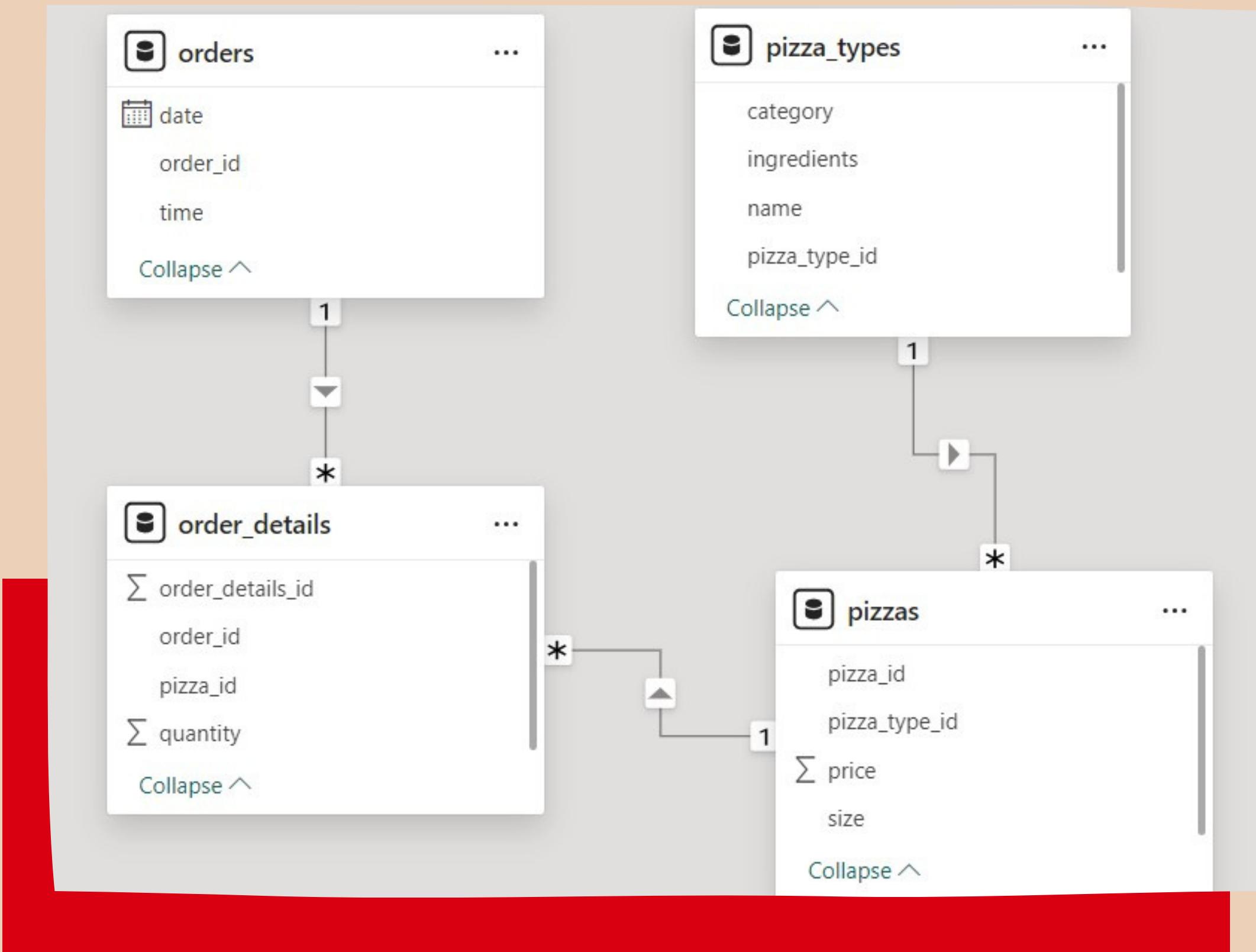


OBJECTIVE

This project aims to explore the metrices like total number of orders, revenue generated, customer preferences and pricing strategies. To gain idea about marketing strategies and inventory management.

We will try to find a clear direction for the analysis and highlights the intended outcomes of the project.

SCHEMA



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
Select Count(Order_id) as Total_orders from Orders
```

OUTPUT :-

Results		Messages
1	Total_orders	21350

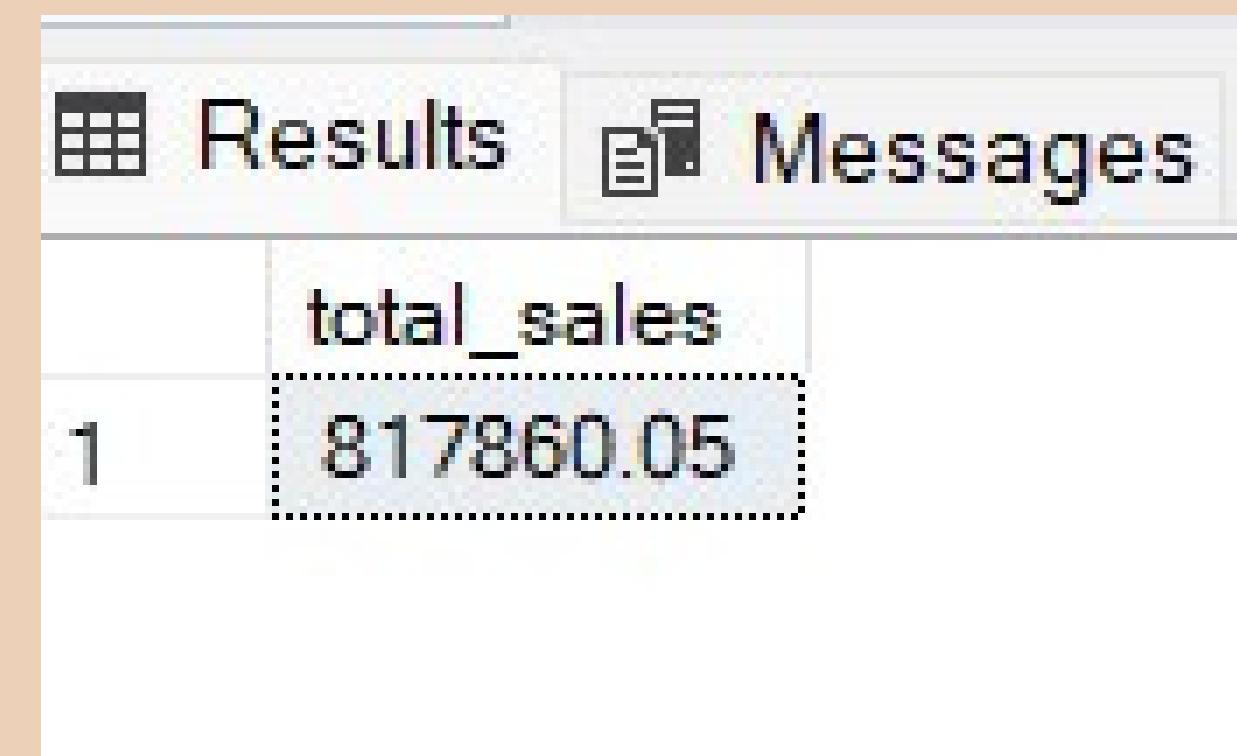


CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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
```



OUTPUT :-



	total_sales
1	817860.05





DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
Select DATEPART(HOUR, [time]) as Hour,  
count(order_id) as Order_count from orders  
group by DATEPART(HOUR, [time])  
order by Hour Desc;
```



OUTPUT :-



	Hour	Order_count
1	23	28
2	22	663
3	21	1198
4	20	1642
5	19	2009
6	18	2399
7	17	2336
8	16	1920
9	15	1468
10	14	1472
11	13	2455
12	12	2520
13	11	1231
14	10	8
15	9	1

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

```
Select Top 3 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 pizzas.pizza_id = order_details.pizza_id  
group by pizza_types.name  
order by Revenue Desc;
```

OUTPUT :-

	name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

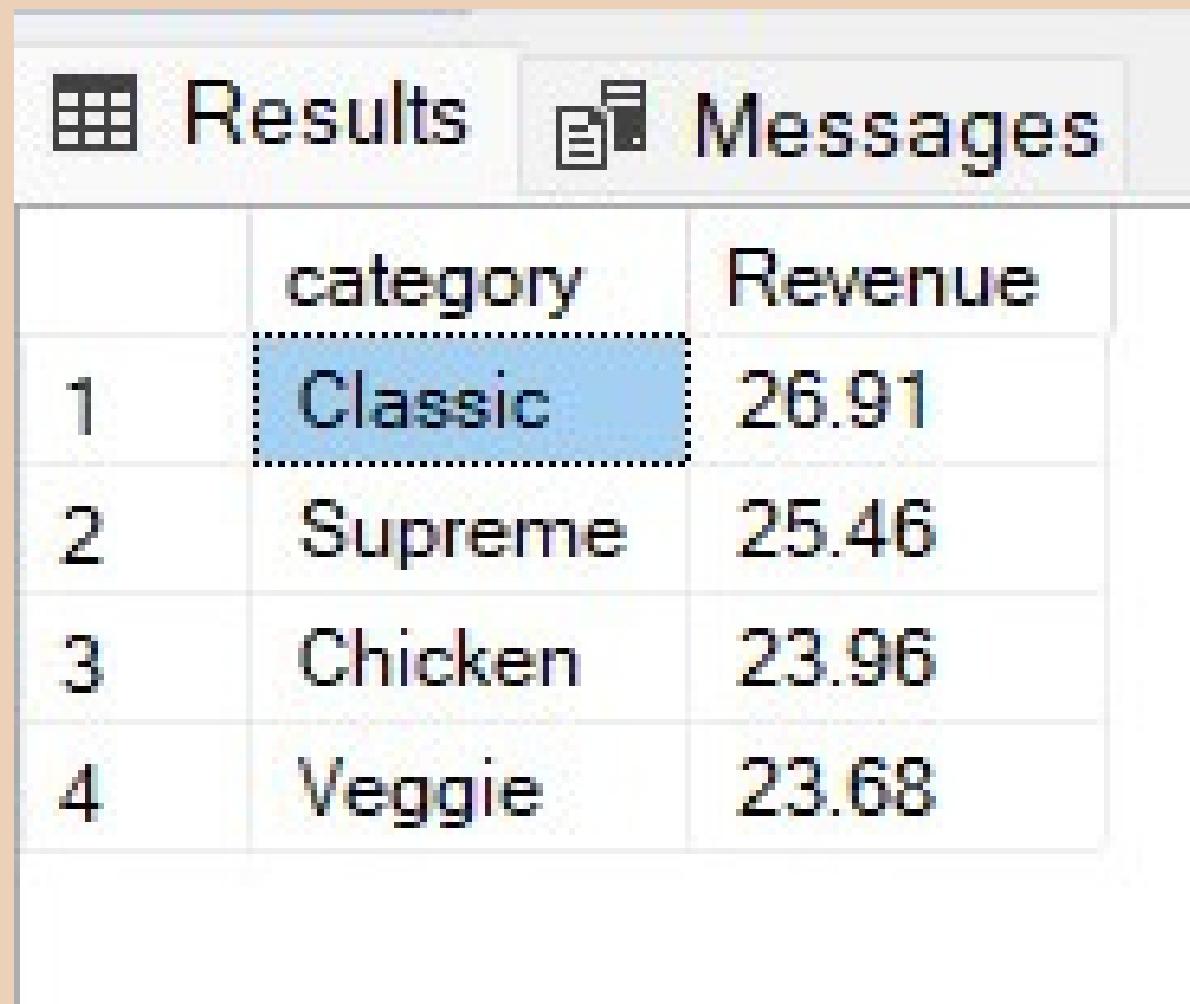


CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
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 pizzas.pizza_id =  
order_details.pizza_id  
group by pizza_types.category  
order by Revenue Desc;
```



OUTPUT :-



	category	Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
WITH PizzaRevenue AS (
    SELECT pizza_types.category,
           pizza_types.name,
           SUM(order_details.quantity * pizzas.price) AS Revenue,
           RANK() OVER (PARTITION BY pizza_types.category
                        ORDER BY SUM(order_details.quantity * pizzas.price) DESC)
           AS RN
    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)
SELECT category, name, Revenue, RN
FROM PizzaRevenue
WHERE RN <= 3;
```

OUTPUT :-

	category	name	Revenue	RN
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken Pizza	42768	2
3	Chicken	The California Chicken Pizza	41409.5	3
4	Classic	The Classic Deluxe Pizza	38180.5	1
5	Classic	The Hawaiian Pizza	32273.25	2
6	Classic	The Pepperoni Pizza	30161.75	3
7	Supreme	The Spicy Italian Pizza	34831.25	1
8	Supreme	The Italian Supreme Pizza	33476.75	2
9	Supreme	The Sicilian Pizza	30940.5	3
10	Veggie	The Four Cheese Pizza	32265.7010040283	1
11	Veggie	The Mexicana Pizza	26780.75	2
12	Veggie	The Five Cheese Pizza	26066.5	3

These are some important queries of this project. By tackling these queries, I gained actionable insights that can inform strategic decision-making, improve operational efficiency, and drive business growth.

THANK YOU

