

PIZZA SALES ANALYSIS USING MY SQL

BY :- PANKAJ VERMA



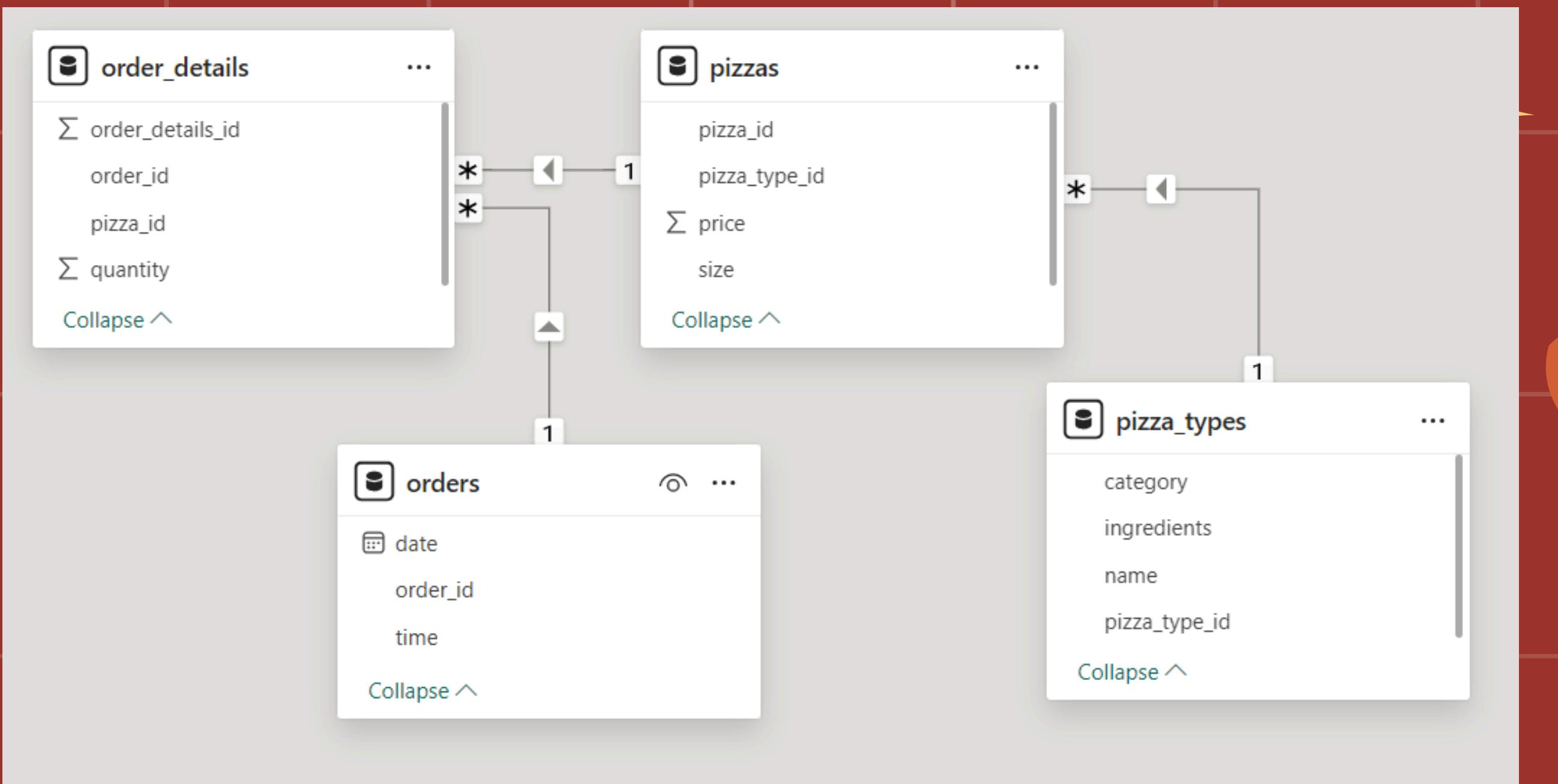
INTRODUCTION

This presentation provides a comprehensive analysis of pizza sales over the past time . The goal is to understand key sales trends, identify top-selling products, and explore customer demographics. By examining these factors, we aim to gain insights that will help in making informed decisions for future marketing strategies, inventory management, and customer engagement.

The analysis covers:

- Overall Sales Performance: A review of sales figures and growth trends.
- Product Insights: Identification of the most popular pizza offerings.
- Sales Trends: Patterns in sales over time, highlighting peak periods and potential areas for improvement.

TABLES SCHEMA



QUERIES TO SOLVE

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.

01

-- Retrieve the total number of orders placed.

```
• select count(order_id) as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350

DE

-- Calculate the total revenue generated from pizza sales.

```
select  
round(sum(order_details.quantity * pizzas.price))as total_Revenue from order_details  
join pizzas on pizzas.pizza_id = order_details.pizza_id
```

Result Grid

	total_Revenue
▶	817860

03

```
1   -- Identify the highest-priced pizza.  
2  
3 • select pizza_types.name , pizzas.price from pizza_types join pizzas  
4   on pizza_types.pizza_type_id = pizzas.pizza_type_id order by price desc limit 1
```

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

04

```
1  -- Identify the most common pizza size ordered.  
2 • select pizzas.size , count(order_details.order_details_id) as order_count  
3   from pizzas  
4   join order_details on order_details.pizza_id = pizzas.pizza_id  
5   group by pizzas.size
```

Result Grid | Filter R

	size	order_count
▶	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 orders  
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 orders desc limit 5;
```

Result Grid | Filter Rows:

	name	orders
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

06

-- Join the necessary tables to find the total quantity of each pizza category ordered.

```
select pizza_types.category,sum(order_details.quantity) as orders  
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 ;
```

Result Grid | F

	category	orders
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

07

-- Determine the distribution of orders by hour of the day.

```
select hour(order_time),count(order_id) from orders  
group by hour(order_time)
```

Result Grid | Filter Rows:

	hour(order_time)	count(order_id)
▶	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

DB

-- Join relevant tables to find the category-wise distribution of pizzas.

```
select category , count(name) from pizza_types  
group by category
```

Result Grid | Filter Row

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

09

```
-- Group the orders by date and calculate the average number of pizzas ordered per day  
select avg(quantity) from  
(select orders.order_data, sum(order_details.quantity) as quantity from orders  
join order_details on order_details.order_id =orders.order_id  
group by orders.order_data) as order_quantity;
```

Result Grid | Filter

	order_data	quantity
▶	2015-01-01	162
	2015-01-02	165
	2015-01-03	158
	2015-01-04	106
	2015-01-05	125
	2015-01-06	147
	2015-01-07	138
	2015-01-08	173
	2015-01-09	127

Result Grid | Filter

	avg(quantity)
▶	138.4749

10

-- 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

```
-- Calculate the percentage contribution of each pizza type to total revenue.  
SELECT  
    pizza_types.category,  
    round(SUM(order_details.quantity * pizzas.price),3 )AS revenue,  
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_Revenue  
    FROM  
        order_details  
        JOIN  
        pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,  
    2) AS Revenue_as_percent  
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 | Filter Rows:

	category	revenue	Revenue_as_percent
▶	Classic	220053.1	26.91
▶	Supreme	208197	25.46
▶	Chicken	195919.5	23.96
▶	Veggie	193690.45	23.68

12

-- Analyze the cumulative revenue generated over time.

```
select order_data , sum(total_Revenue) over(order by order_data) as cum_revenue
from
(select orders.order_data,
round(sum(order_details.quantity * pizzas.price),2)as total_Revenue from order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id join orders
on orders.order_id =order_details.order_id
group by orders.order_data) as sales;
```

Result Grid | Filter Rows:

	order_data	cum_revenue
1	2015-01-01	2713.85
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

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
select category, name ,revenue from  
(select category , name , revenue ,  
rank() over (partition by category order by revenue desc ) as top  
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 top <=3 ;
```

13

Result Grid Filter Rows: Export:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5

THANK
YOU!