PIZZA SALES

DATA ANALYSIS PROJECT USING SQL

PRESENTED BY :
SHUBHAM KUMAR

SQL

Welcome !

Today, we will explore our data analysis project focused on pizza sales. In an era where data drives business decisions, understanding sales trends and customer preferences is crucial for optimizing performance.

Objectives:

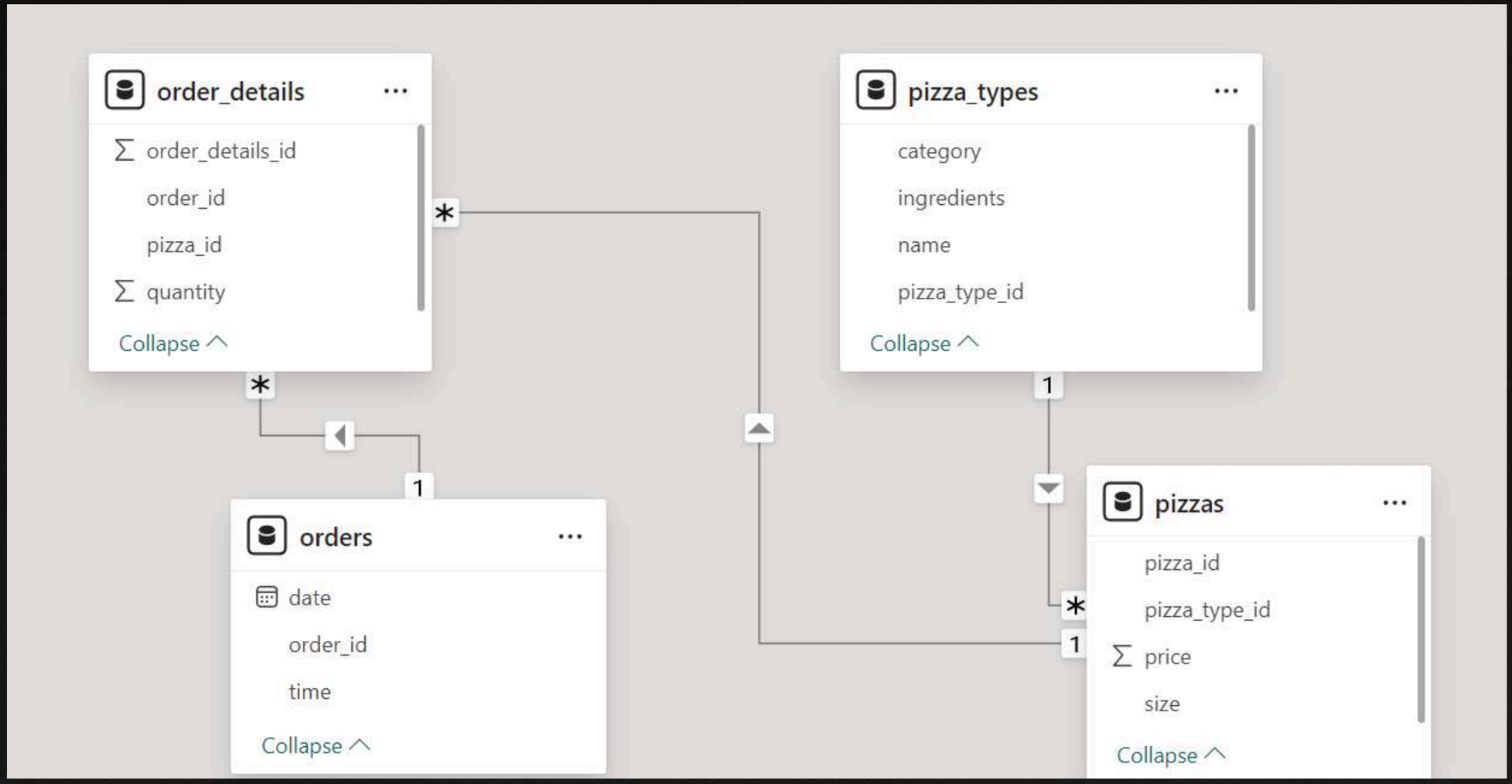
- Highlight the significance of data analysis in the food industry, specifically pizza sales.
- Introduce the SQL techniques to analyze our dataset.
- Share key findings that can drive decision-making for pizza businesses.

Let’s dive into how SQL empowers us to make sense of complex datasets and drives impactful business strategies.

TOOL USED : MYSQL

SQL

Database Schema



SQL

Questions

Basic :

01 Retrieve the total number of orders placed.

02 Calculate the total revenue generated from pizza sales.

03 Identify the highest-priced pizza.

04 Identify the most common pizza size ordered.

05 List the top 5 most ordered pizza types along with their quantities.

SQL

Intermediate:

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

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

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

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

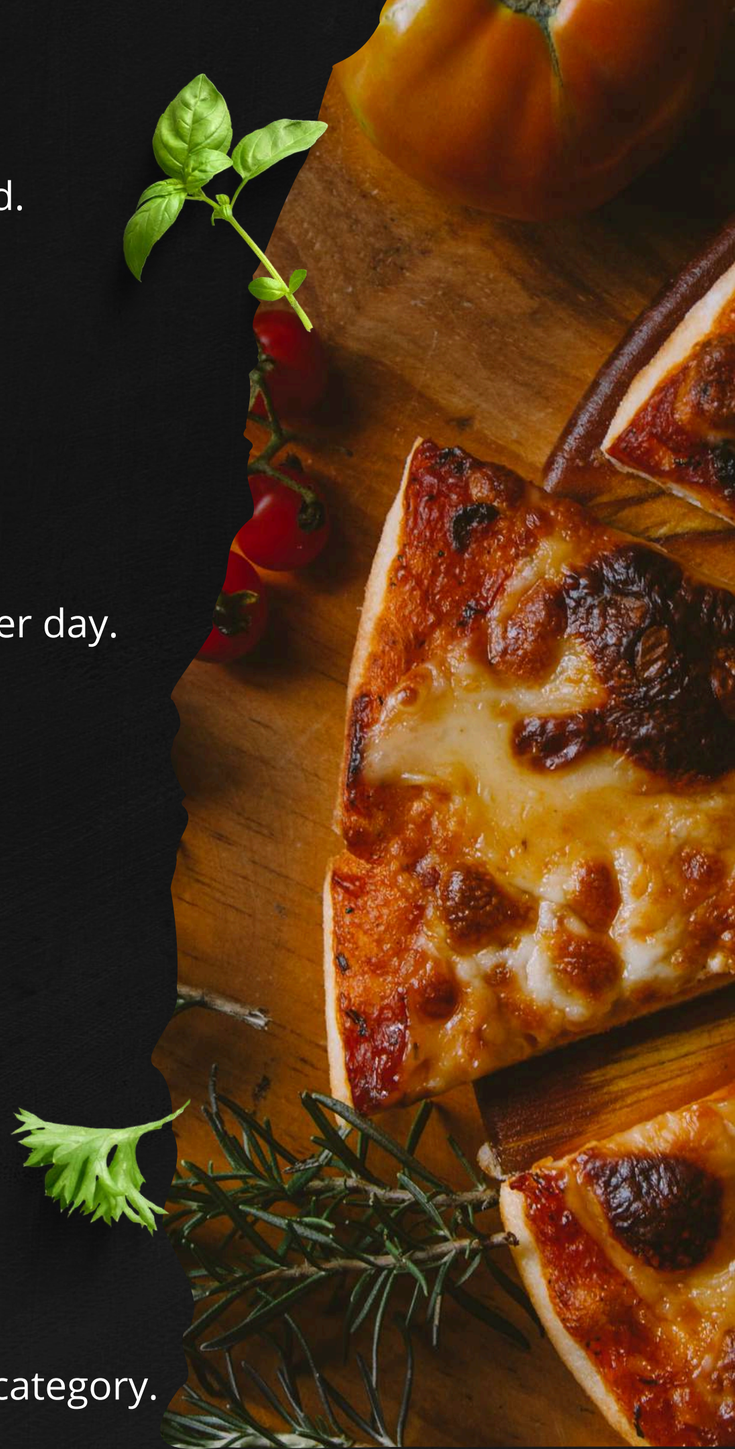
05 Determine the top 3 most ordered pizza types based on revenue.

Advanced:

01 Calculate the percentage contribution of each pizza type to total revenue.

02 Analyze the cumulative revenue generated over time.

03 Determine the top 3 most ordered pizza types based on revenue for each pizza category.



Retrieve the total number of orders placed.

```sql
select count(order_id) as Total_Orders
from orders;
```

Result Grid

	Total_Orders
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(quantity * price), 2) AS Total_Revenue_Generated
FROM
    order_details od
        INNER JOIN
    pizzas p ON od.pizza_id = p.pizza_id;
```

Result Grid | Filter Rows:

	Total_Revenue_Generated
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    name AS Highest_Priced_Pizza, price
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	Highest_Priced_Pizza	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    SUM(quantity) AS Quantity_ordered, size
FROM
    order_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY Quantity_ordered DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	Quantity_ordered	size
▶	18956	L

List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    name, SUM(quantity) AS Quantity_Ordered
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY Quantity_Ordered DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	Quantity_Ordered
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.

```sql
SELECT
    category, SUM(quantity) AS Quantity_Ordered
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY category;
```

Result Grid | Filter Rows:

	category	Quantity_Ordered
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS Hours, COUNT(*) AS Count_Of_Orders
FROM
    orders
GROUP BY Hours;
```

Result Grid | Filter Rows

Hours	Count_Of_Orders
11	1231
12	2520
13	2455

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

```sql
SELECT
    category, COUNT(*) AS No_of_Pizzas
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Ro

	category	No_of_Pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```sql
SELECT
    ROUND(AVG(total_quantity), 0) AS Average_Pizza_ordered_per_day
FROM
    (SELECT
        order_date, SUM(quantity) AS total_quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY order_date) AS quantity_table;
```

Result Grid | Filter Rows:

	Average_Pizza_ordered_per_day
▶	138

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

```sql
SELECT
    name, SUM(quantity * price) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY 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.

```sql
select category,concat(round(sum(quantity * price)/
(select sum(quantity * price)
from order_details od inner join pizzas p
on od.pizza_id = p.pizza_id) * 100,2), ' %') as Percentage_wise_contribution
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od
on p.pizza_id = od.pizza_id
group by category;
```

Result Grid | Filter Rows:

category	Percentage_wise_contribution
Classic	26.91 %
Veggie	23.68 %
Supreme	25.46 %
Chicken	23.96 %

Analyze the cumulative revenue generated over time.

```sql
select order_date,
 sum(revenue) over (order by order_date) as Cumulative_Revenue
from
(select order_date, sum(quantity * price) as revenue
from orders o join order_details od
on o.order_id = od.order_id
join pizzas p
on od.pizza_id = p.pizza_id
group by order_date) as revenue_table;
```

Result Grid | Filter Rows:

order_date	Cumulative_Revenue
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

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select name,category,revenue
from
(select name,category,revenue,
rank() over(partition by category order by revenue desc) as rn
 from
(select name,category, sum(quantity * price) as revenue
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od
on p.pizza_id = od.pizza_id
group by category,name) as a) as b
where rn <=3;
```

	name	category	revenue
▶	The Thai Chicken Pizza	Chicken	43434.25
	The Barbecue Chicken Pizza	Chicken	42768
	The California Chicken Pizza	Chicken	41409.5
	The Classic Deluxe Pizza	Classic	38180.5
	The Hawaiian Pizza	Classic	32273.25

SQL

Contact Info

+91 78071-77384

www.linkedin.com/in/shubham-kumar-499838227

iamshubhamwork2@gmail.com

Looking forward to your feedback!

SQL

Thank You

Thank you for your attention! I appreciate your interest in my data analysis project on pizza sales using SQL.