
PIZZA SALES
PROJECT

HELLO!

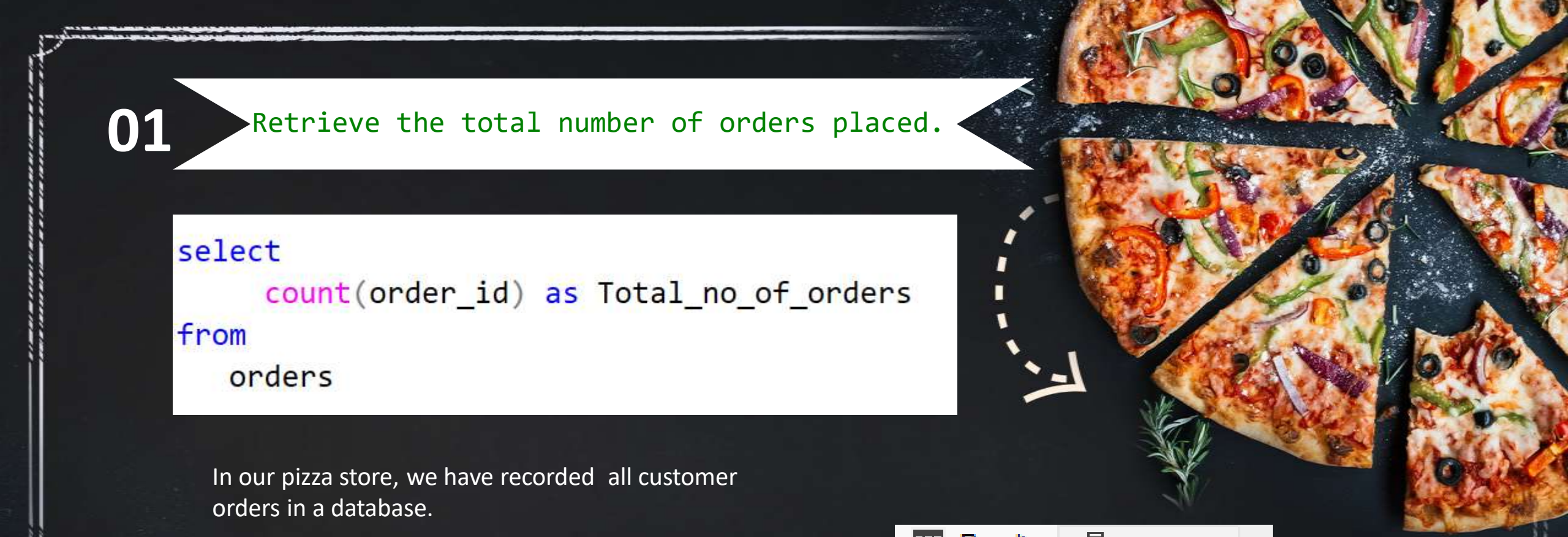
My name is Vikas Koli, and I am excited to present my project on analyzing pizza sales using SQL queries.

In today's data-driven world, businesses need to leverage data to make informed decisions and optimize their operations. This project focuses on utilizing SQL queries to analyze sales data from a pizza store, with the aim of answering key business questions related to sales performance, customer preferences, and operational efficiency.

The objective of this project is to use SQL to extract valuable insights from the pizza sales data, which can help the store understand its sales patterns, identify popular items, and improve its overall business strategy.

01 Retrieve the total number of orders placed.

```
select
      count(order_id) as Total_no_of_orders
from
   orders
```



In our pizza store, we have recorded all customer orders in a database.

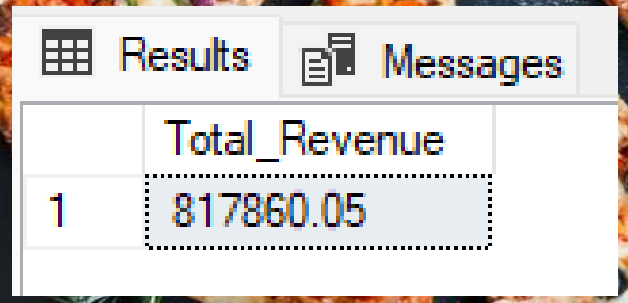
This query counts all the rows in the "orders" table, where each row represents a unique order. The result of this query shows that there were 21,350 orders placed over the course of a year.

Results | Messages

	Total_no_of_orders
1	21350

02 Calculate the total revenue generated from pizza sales.

```
select
	round(sum(order_details.quantity * pizzas.price),2) as Total_Revenue
from
	order_details
		join
	pizzas on order_details.pizza_id = pizzas.pizza_id
```



This part of the query multiplies the quantity of each pizza ordered by its price & This join operation links the order_details table, which contains the quantities of pizzas ordered, with the pizzas table, which contains the prices of the pizzas. The join condition ensures that we correctly match each order with the corresponding pizza price. This query effectively calculates the total revenue from all pizza sales by combining order quantities with their respective prices, giving us a comprehensive view of our sales performance

Our pizza shop has generated a total revenue of $817,860 in one year. This figure is derived from aggregating the sales of all individual orders placed by customers throughout the year. Each order contributes to this total, and the revenue from each order is recorded in our sales database. The total annual revenue of $817,860 is a key indicator of our shop's financial health and operational success.

03 Identify the highest-priced pizza.

```sql
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
	OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY;
```

Results | Messages

	name	price
1	The Greek Pizza	35.95

This query fetches the name and price of the pizza type that has the highest price among all available pizza types. It does this by joining two tables: pizza_types and pizzas. The join condition ensures that each pizza type is matched with its corresponding price based on their common attribute, which is the pizza_type_id.

The Greek pizza, priced at $35.95, is the most expensive pizza sold in pizza shop in a single year due to a combination of premium ingredients, complex preparation, strategic market positioning, and balancing demand with supply. This pricing strategy not only covers the costs and ensures profitability but also caters to a segment of customers seeking high-quality, gourmet options, thereby contributing to the shop's overall success.

04 Identify the most common pizza size ordered.

```
select
     count(order_details.quantity)as total_count , pizzas.size
from
     order_details
       join
     pizzas on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by total_count desc
```

Results | Messages

	total_count	size
1	18526	L
2	15385	M
3	14137	S
4	544	XL
5	28	XXL

This query calculates the total number of pizzas ordered for each size available in the database. It achieves this by joining two tables: order _details and pizzas. The join condition ensures that each order detail is matched with its corresponding pizza size based on their common attribute, which is the pizza_id. this query provides a breakdown of the total number of pizzas ordered for each available size, helping to analyze customer preferences based on pizza size.

The L size pizza is the most popular order, with a total of 18,526 orders placed over the course of a year. This indicates that customers have a strong preference for the L size pizza compared to other sizes available. The L size pizza might be the most popular because of Value for Money, Variety of Toppings, Promotional Offers, Customer Preferences etc. Overall, the high number of orders for the L size pizza suggests that it meets the needs and preferences of a significant portion of the customer base, making it the most popular choice among available options.

05 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
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

Results | Messages

	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

The top 5 most popular pizza varieties are retrieved by this query based on the total amount ordered. It uses the SUM(order_details.quantity) function to determine the total quantity ordered for each variety of pizza. Three tables are joined by the query: order_details, pizzas, and pizza_types. By adding up the quantities ordered for each variety of pizza and sorting the results in descending order of total quantity. This makes it easier to determine which varieties of pizza are most popular among customers.

The popularity of these pizzas indicates strong customer preferences and can provide valuable insights for menu planning, inventory management, and targeted marketing campaigns. This data highlights which pizza types are most favored by our customers, helping to ensure we focus on maintaining the quality and availability of these popular options.

06

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

```
select
      pizza_types.category, sum(order_details.quantity) as total_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 total_quantity desc;
```

Results | Messages

	category	total_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

This query aims to determine the total quantity of pizzas ordered for each category of pizza types. The SUM(order_details.quantity) function calculates the total quantity ordered for each pizza category. The query joins the pizza_types table with the pizzas table on the pizza_type_id column to link each pizza with its corresponding type and sorting the results in descending order of total quantity.

Understanding these preferences is crucial for optimizing our menu, managing inventory effectively, and tailoring marketing strategies to customer tastes. By focusing on the Classic category, we can ensure that our most popular pizzas continue to satisfy our customers' preferences.

07

Determine the distribution of orders by hour of the day.

```
SELECT
     DATEPART(HOUR, time) AS OrderHour, COUNT(*) AS OrderCount
FROM
     Orders
GROUP BY DATEPART(HOUR, time)
ORDER BY OrderHour;
```

Results | Messages

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

This query provides a breakdown of the number of orders placed during each hour of the day. It extracts the hour from the order times, groups the orders by each hour, counts the total orders per hour, and sorts the results chronologically. This information helps to identify peak ordering times

Our analysis of order data reveals that the busiest hours for pizza orders are between 11 AM to 9 PM. During this time frame, we experience the highest volume of orders, indicating strong customer demand. By Optimizing Staffing, Managing Inventory & Planning Promotions we can focuson these peak hours and we can improve our service quality, reduce wait times, and enhance customer satisfaction.

08

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

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

Results | Messages

	category	Total_category
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

This query aims to count the number of different pizza types within each category. It groups the pizza types by their category and then counts how many types fall into each group. This helps in understanding the variety and distribution of pizza offerings across different categories, which can help with marketing strategy and menu planning.

Our pizza menu is divided into four distinct categories. This distribution highlights our focus on offering a wide range of veggie and supreme pizzas, catering to diverse customer preferences. Understanding the variety within each category helps in managing our inventory, planning promotions, and ensuring we meet customer demand effectively.

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

```
select
      round(avg(quantity),0) as Average_no_of_pizza_order_per_day
from
   (select orders.date , sum(order_details.quantity) as quantity
from
     orders
       join
     order_details on orders.order_id = order_details.order_id
group by orders.date) as order_by_date;
```

Results | Messages

	Average_no_of_pizza_order_per_day
1	138

This query determines the average number of pizzas ordered per day by first summing the total quantity of pizzas ordered each day and then calculating the average of these daily totals. The result is rounded to the nearest whole number. This information helps in understanding daily sales performance

Our analysis shows that the average number of pizza orders per day is 138. This figure represents the typical daily demand for our pizzas, calculated by averaging the total number of pizzas ordered each day over a given period. This information is crucial for ensuring, we have adequate inventory and staff to meet daily customer demand. By Understanding the average daily orders helps in planning promotions, managing supply chains, and optimizing our operations to improve efficiency and customer satisfaction.

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

```
select
      pizza_types.name, pizza_types.category,
      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,pizza_types.category
order by revenue desc
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY
```

Results | Messages

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

This query identifies the top 3 pizza types that generate the highest revenue by calculating the total sales revenue for each type and sorting the results. It helps in understanding which pizzas contribute the most to the business's income.

Based on our analysis, the top 3 pizzas generating the highest revenue are Thai Chicken Pizza, Barbecue Chicken Pizza, California Chicken Pizza. All three top-performing pizzas belong to the chicken category, indicating a strong customer preference for chicken-based pizzas. Because these pizzas are so popular, it is a good idea to make sure they are clearly marked on the menu and easily accessible. To take advantage, targeted promotions and marketing efforts can focus on chicken pizzas to capitalize on their high demand. Understanding these trends helps us tailor our operations to meet customer preferences effectively.

Conclusion and Thank You.

"To conclude, this project demonstrated how SQL queries can be effectively used to analyze pizza sales data and derive actionable insights. By understanding sales patterns, identifying popular items, and pinpointing peak sales times, the pizza store can make informed decisions to enhance its business strategies and improve customer satisfaction.

Thank you all for your attention and for giving me the opportunity to present my analysis. I hope you found the insights valuable and can see the potential of using data analysis in driving business growth. I am happy to answer any questions you may have.

Thank you."

THANK YOU