"PIZZA SALES ANALYSIS USING SQL"

Presented By : Saurabh Ronghe

INTRODUCTION :

"TO ANALYZE PIZZA SALES TRENDS, IDENTIFY TOP-SELLING ITEMS, AND RECOMMEND STRATEGIES FOR BUSINESS GROWTH."

DESCRIPTION :

THIS PRESENTATION FOCUSES ON ANALYZING PIZZA SALES DATA USING SQL TO IDENTIFY PATTERNS, TRENDS, AND INSIGHTS THAT CAN DRIVE BETTER DECISION-MAKING. BY LEVERAGING SQL QUERIES, THE ANALYSIS DELVES INTO KEY PERFORMANCE METRICS SUCH AS TOTAL SALES, POPULAR PIZZA CATEGORIES, PEAK SALES PERIODS, CUSTOMER PREFERENCES, AND GEOGRAPHIC TRENDS. THE GOAL IS TO PROVIDE A DATA-DRIVEN UNDERSTANDING OF BUSINESS PERFORMANCE AND OFFER STRATEGIC RECOMMENDATIONS FOR IMPROVING SALES AND OPERATIONAL EFFICIENCY.

PURPOSE :

TO ANALYZE PIZZA SALES DATA USING SQL TO UNCOVER KEY INSIGHTS INTO SALES TRENDS, CUSTOMER PREFERENCES, AND PERFORMANCE METRICS. THE PRESENTATION AIMS TO PROVIDE ACTIONABLE RECOMMENDATIONS TO OPTIMIZE INVENTORY, IMPROVE MARKETING STRATEGIES, AND ENHANCE OVERALL BUSINESS EFFICIENCY.

DATASET OVERVIEW

ORDERS TABLE
COLUMNS : ORDER_ID, ORDER_DATE, ORDER_TIME.
INSIGHTS : HELPS ANALYZE SALES TRENDS OVER TIME AND TRACK CUSTOMER BEHAVIOR.

ORDER_DETAILS TABLE
COLUMNS : ORDER_DETAIL_ID, ORDER_ID, PIZZA_ID, QUANTITY.
INSIGHTS : LINKS ORDERS TO SPECIFIC PIZZA ITEMS, ENABLING DETAILED SALES ANALYSIS.

PIZZA_TYPES TABLE
COLUMNS : PIZZA_TYPE_ID, NAME, INGREDIENTS, CATEGORY.
INSIGHTS : USEFUL FOR ANALYZING THE POPULARITY OF DIFFERENT PIZZA TYPES AND CATEGORIES.

PIZZAS TABLE
COLUMNS : PIZZA_ID, PIZZA_TYPE_ID, SIZE.
INSIGHTS: HELPS DETERMINE SIZE PREFERENCES AND VARIANT-LEVEL PERFORMANCE.

1. Retrieve the total number of orders placed.

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

Total_Orders
21350

2. Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS Total_Revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

| Result Grid | | |

Total_Revenue
▶ 817860.05

3. 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
LIMIT 1;
```

name	price
The Greek Pizza	35.95

4.Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS Order_Count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY Order_Count DESC;
```

Result Grid	Filter Row

size	Order_Count
L	18526
M	15385
S	14137
XL	544
XXL	28

Result 2 ✕

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

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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

Result 1 ×

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

```sql
select pizza_types.category,
sum(orders_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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

Result 1 ✕

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

```sql
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

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

- **SELECT**

    category, COUNT(name)

  FROM

    pizza_types

  GROUP BY category;

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

Result Grid | Filter Rows:

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as average_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	Filter Rows:
	average_pizza_ordered_per_day
▶	138

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

```sql
select pizza_types.name,
sum(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join orders_details
on orders_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

BEST SELLER

THE THIE CHICKEN PIZZA

Conclusion :

- "Sales increased steadily over weekends, with dinner hours being the peak period."
- "Combo meals are the most popular, contributing 45% of total revenue."
- "Pepperoni Pizza is the highest-selling item, accounting for 30% of sales."
- "Vegetarian Pizza has the lowest demand at 5%."
- "Peak sales occur on weekends between 6-9 PM."
- "Slow sales during weekday afternoons."
- "Families order combos more frequently on weekends."

Future Scope :

- "Launch weekend promotions and combo meal discounts."
- "Target weekday afternoon customers with 'Happy Hour' offers."
- "Introduce new vegetarian options to boost underperforming categories."
- "Offer seasonal pizzas based on customer preferences."
- "Hire additional staff during peak weekend hours."
- "Optimize ingredient inventory for high-demand items to reduce wastage."