LAZIZ PIZZA

PIZZA SALES DATA ANALYSIS

BY SHUBHAM DABI

INTRODUCTION

Laziz Pizza is a renowned pizzeria known for its delicious variety of pizzas.

This sales data analysis uncovers key insights like total orders, revenue, and popular pizza types, helping to understand customer preferences and business performance. By analyzing trends and contributions, it supports strategic decisions for growth.

BY SHUBHAM DABI

OBJECTIVES

- **Identify Popular Pizza:** Highlight the most ordered pizza types and sizes to understand customer preferences.
- **Order Trends:** Examine the distribution of orders by time of day and across dates.
- **Optimize Inventory and Strategy:** Use insights to optimize inventory management and marketing strategies.
- **Daily Sales Insights:** Calculate the average number of pizzas sold per day to evaluate daily performance.
- **Peak Ordering Times:** Identify peak hours for orders to allocate resources effectively during high-demand periods.
- **Revenue Growth Analysis:** Track cumulative revenue trends over time for better forecasting.

BY SHUBHAM DABI

ANALYSIS TECHNIQUES

- **Aggregation Functions:** Used COUNT(), SUM(), and ROUND() for calculating totals, revenue, and averages.
- **Sorting and Ranking:** Implemented ORDER BY, LIMIT, and RANK() to sort data and identify top-performing pizzas.
- **Joins:** Applied INNER JOIN to link related tables like orders, order_details, pizzas, and pizza_types.
- **Window Functions**: Utilized SUM() OVER() and RANK() OVER() for cumulative revenue and ranking within categories.
- **Grouping and Subtotals:** Used GROUP BY and WITH ROLLUP for summarizing data by category, size, or date.
- **Conditional Logic:** Leveraged CASE statements for categorizing shifts and other dynamic grouping.

BY SHUBHAM DABI

BUSINESS PROBLEMS QUESTION

1. Find out how many orders were placed overall.
2. Calculate the total income earned by pizza sales.
3. Determine which Top 5 pizzas has the highest price.
4. Determine the most common pizza size ordered.
5. List the Top 10 most popular pizza types, along with their quantities.
6. Find the total amount of each pizza ordered by joining the relevant tables.
7. Determine the order distribution by hour of the day.

BY SHUBHAM DABI

BUSINESS PROBLEMS QUESTION

8. Join relevant tables to determine the distribution of pizzas by category.

9. Sort the orders by date and get the average number of pizzas ordered per day.

10. Based on sales, identify the three most popular pizza varieties.

11. Determine the percentage contribution of each pizza type to overall revenue.

12. Analyze the overall revenue generated over time.

13. Determine the top three pizza kinds ordered by revenue in each pizza category.

BY SHUBHAM DABI

BUSINESS PROBLEMS QUESTION

8. Join relevant tables to determine the distribution of pizzas by category.

9. Sort the orders by date and get the average number of pizzas ordered per day.

10. Based on sales, identify the three most popular pizza varieties.

11. Determine the percentage contribution of each pizza type to overall revenue.

12. Analyze the overall revenue generated over time.

13. Determine the top three pizza kinds ordered by revenue in each pizza category.

BY SHUBHAM DABI

BUSINESS PROBLEM SOLUTIONS

```sql
-- Total Number of Order Places

SELECT
    COUNT(Order_id) AS 'TOTAL ORDERS'
FROM
    orders
```

	TOTAL ORDERS
▶	21350

```sql
-- Total Revenue From Pizza Sales

SELECT
    ROUND(SUM((quantity * price)), 2) AS 'Total_Revenue'
FROM
    order_details
        INNER JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

	Total_Revenue
▶	817860.05

BUSINESS PROBLEM SOLUTIONS

```sql
-- TOP 5 Higgest Priced Pizza

SELECT
    name, price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizzas.pizza_type_id
ORDER BY (pizzas.price) DESC
LIMIT 5
```

name	price
The Greek Pizza	35.95
The Italian Capocollo Pizza	35.95
The Green Garden Pizza	35.95
The Prosciutto and Arugula Pizza	35.95
The Calabrese Pizza	35.95

```sql
-- Most Common Pizza Size Ordered

Select size, count(quantity)as Total_Quantity from order_details
join pizzas on
order_details.pizza_id = pizzas.pizza_id
group by size
```

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

BUSINESS PROBLEM SOLUTIONS

```sql
-- Top 10 Ordered Pizza Type Along With their Quantities

select name, Sum(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 name order by Total_quantity desc
Limit 10
```

name	Total_Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371
The California Chicken Pizza	2370
The Sicilian Pizza	1938
The Spicy Italian Pizza	1924
The Southwest Chicken Pizza	1917
The Big Meat Pizza	1914

```sql
-- total quantity of each pizza ordered

SELECT
    category, SUM(quantity) as Total_Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY category
```

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

BUSINESS PROBLEM SOLUTIONS

```sql
-- Distribution of orders by hour of the day

SELECT
    COUNT(order_id) AS Order_Count,
    CASE
        WHEN HOUR(Time) < 12 THEN CONCAT(MOD(HOUR(Time), 12), ' AM')
        ELSE CONCAT(MOD(HOUR(Time), 12), ' PM')
    END AS hours_12
FROM
    orders
GROUP BY
    hours_12
ORDER BY
    MIN(HOUR(Time))
```

Order_Count	hours_12
1	9 AM
8	10 AM
1231	11 AM
2520	0 PM
2455	1 PM
1472	2 PM
1468	3 PM
1920	4 PM
2336	5 PM
2399	6 PM
2009	7 PM
1642	8 PM
1198	9 PM
663	10 PM
28	11 PM

```sql
-- Catagories wise distribution of Pizza

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

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

BUSINESS PROBLEM SOLUTIONS

```sql
-- Top 5 Most ordered Pizzas Type based on Revenue

SELECT
    name, SUM((price * quantity)) AS Revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY name
ORDER BY (Revenue) DESC
LIMIT 5
```

name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25

```sql
-- Group the Orders by Date and calculate the
-- average number of pizzas order per day

SELECT
    ROUND(AVG(Tot_Quantity), 0) AS AVG_Order_Per_Day
FROM
    (SELECT
        date, SUM(quantity) AS Tot_Quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY date) AS AVG_Quantity
```

AVG_Order_Per_Day
138

BUSINESS PROBLEM SOLUTIONS

```sql
-- Percentages Contribution of each Pizza type to Total Sales

SELECT
    IFNULL(category, 'Total'),
    ROUND(((SUM((price * quantity)) / (SELECT
                    ROUND(SUM((quantity * price)), 2)
                FROM
                    order_details
                        INNER JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100),
            2) AS '% of Category by Sales'
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY category WITH ROLLUP
```

IFNULL(category, 'Total')	% of Category by Sales
Chicken	23.96
Classic	26.91
Supreme	25.46
Veggie	23.68
Total	100

BUSINESS PROBLEM SOLUTIONS

```sql
-- analyze Cumulative revenue generated over Time

select date, Total_Revenue, Round(sum(Total_Revenue) over (order by date),2) as Running_Revenue
From
(select date, round(Sum((quantity*price)),2) as Total_Revenue
from
orders join order_details on order_details.order_id = orders.order_id
join pizzas on order_details.pizza_id = pizzas.pizza_id
group by orders.date) as Total_Sales
```

date	Total_Revenue	Running_Revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5

BUSINESS PROBLEM SOLUTIONS

```sql
-- Top 3 most ordered Pizza type based on Revenue for each Category
select name, category, Total_Revenue From
(Select
name,
category,
sum((quantity * price)) as Total_Revenue,
rank() over(partition by category order by sum((quantity * price))desc) As 'RANKKK'
From
pizza_types join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) As Rankk
where RANKKK <=3
```

name	category	Total_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
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

THANK YOU!

Thank you for taking the time to explore this SQL query analysis!

If you found it informative give it a like.

BY SHUBHAM DABI