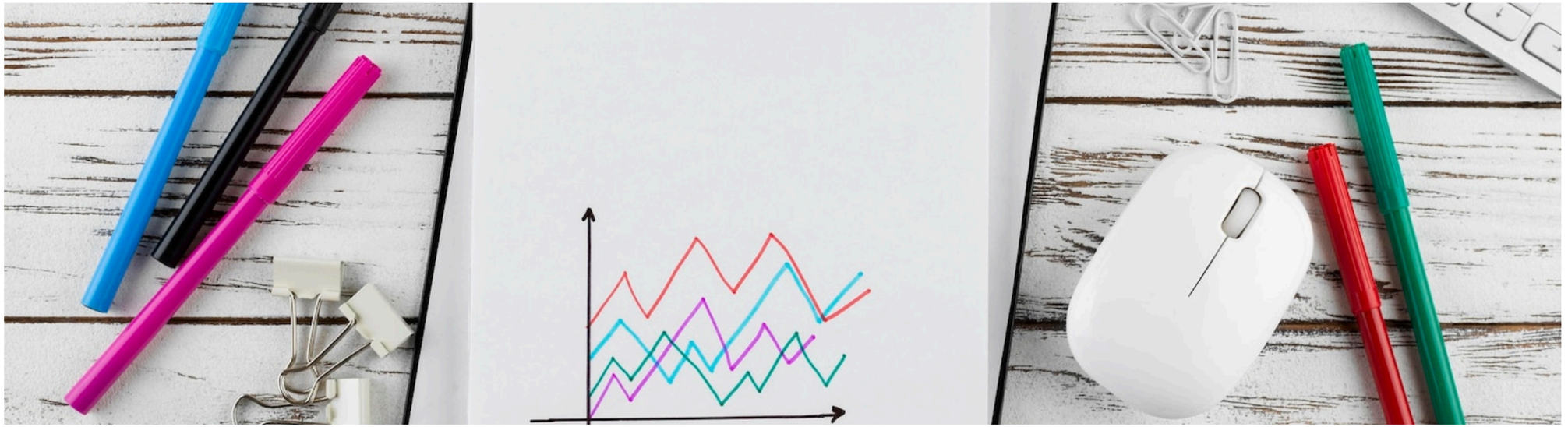

Pizza Sales Analysis project

Insights and Strategies Through Oracle SQL Developer

Introduction to Pizza Sales Analysis

The Pizza Sales Analysis project focuses on analyzing pizza order data to derive insights on sales patterns, popular pizza types, and revenue trends. It uses SQL queries to explore various business metrics such as total orders, revenue, and pizza preferences.

Project Overview

I developed a sales data analysis project using Oracle SQL Developer to derive actionable insights. The database consisted of four relational tables: Pizzas, Pizza Types, Orders, and Order Details. This project focused on optimizing queries and analyzing trends for improved decision-making.

Collecting Pizza Sales Data

This project collects and analyzes data related to pizza sales using four tables: Pizzas, Pizza Types, Orders, and Order Details. It computes various metrics like total orders, revenue, pizza sizes, categories, and distribution. The analysis helps identify trends, popular pizza types, and customer preferences.

Database Schema

Pizzas Table: Pizza ID, Pizza Type ID, Pizza Size, Price.
Pizza Types Table: Pizza Type ID, Name, Category, Ingredients.
Orders Table: Order ID, Order Date, Order Time.
Order Details Table: Order Detail ID, Order ID, Pizza ID, Quantity..

Retrieve the total number of orders placed.

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

	TOTAL_ORDERS
1	21350

Calculate the total revenue generated from pizza sales

```
select
    sum(d.quantity*p.price) as total_sales
from
    order_details d,
    pizzas p
where
    d.pizza_id= p.pizza_id;
```

	TOTAL_SALES
1	817860.05

Identify the most common pizza size ordered

```
select
    p.pizza_size,
    sum(d.quantity) as total_size
from
    pizzas p,
    order_details d
where
    p.pizza_id = d.pizza_id
group by
    p.pizza_size
order by
    total_size desc
fetch first 1 row only;
```

	PIZZA_SIZE	TOTAL_SIZE
1	L	18956

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

```sql
select
    t.name,
    p.pizza_type_id,
    sum(d.quantity) as count
from
    order_details d
join
    pizzas p ON d.pizza_id = p.pizza_id
join
    pizza_types t on t.pizza_type_id = p.pizza_type_id
group by
    t.name,
    p.pizza_type_id
order by
    count desc
fetch first 5 row only;
```

	NAME	PIZZA_TYPE_ID	COUNT
1	The Classic Deluxe Pizza	classic dlx	2453
2	The Barbecue Chicken Pizza	bbq ckn	2432
3	The Hawaiian Pizza	hawaiian	2422
4	The Pepperoni Pizza	pepperoni	2418
5	The Thai Chicken Pizza	thai ckn	2371

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

```sql
select
    t.category,
    sum(d.quantity) as count
from
    order_details d
join
    pizzas p ON d.pizza_id = p.pizza_id
join
    pizza_types t on t.pizza_type_id = p.pizza_type_id
group by
    t.category
order by
    count desc;
```

	CATEGORY	COUNT
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Determine the distribution of orders by hour of the day

```
select * from orders;
SELECT
    TO_CHAR(TO_DATE(order_time, 'HH24:MI:SS'), 'HH24') AS hour,
    count(order_id) as count
FROM
    orders
group by
    TO_CHAR(TO_DATE(order_time, 'HH24:MI:SS'), 'HH24');
```

	HOUR	COUNT
1	11	1231
2	12	2520
3	13	2455
4	14	1472
5	15	1468
6	16	1920
7	17	2336
8	18	2399
9	19	2009
10	20	1642
11	21	1198
12	22	663
13	23	28
14	10	8
15	09	1

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

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

	CATEGORY	COUNT
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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

```
select
    round(avg(sum(d.quantity)),2) as average
from orders o,
    order_details d
where
    o.order_id=d.order_id
group by
    o.order_date;
```

	AVERAGE
1	138.47

Determine the top 3 most ordered pizza types based on revenue

```
select
    t.name,
    round(sum(p.price*d.quantity),0) as amount
from pizzas p
join order_details d on p.pizza_id = d.pizza_id
join pizza_types t on p.pizza_type_id = t.pizza_type_id
group by
    p.pizza_type_id,
    t.name
order by
    amount desc
fetch first 3 row only;
```

	NAME	AMOUNT
1	The Thai Chicken Pizza	43434
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41410

Calculate the percentage contribution of each pizza type to total revenue

```sql
select
    t.name,
    round((sum(p.price*d.quantity)/( select sum(d.quantity*p.price) as total_sales
from
    order_details d,
    pizzas p
where d.pizza_id= p.pizza_id))*100,2) as amount
from pizzas p
join order_details d on p.pizza_id=d.pizza_id
join pizza_types t on p.pizza_type_id=t.pizza_type_id
group by
    p.pizza_type_id,
    t.name
order by
    amount desc;
```

	NAME	AMOUNT
1	The Thai Chicken Pizza	5.31
2	The Barbecue Chicken Pizza	5.23
3	The California Chicken Pizza	5.06
4	The Classic Deluxe Pizza	4.67
5	The Spicy Italian Pizza	4.26
6	The Southwest Chicken Pizza	4.24
7	The Italian Supreme Pizza	4.09
8	The Four Cheese Pizza	3.95
9	The Hawaiian Pizza	3.95

Analyze the cumulative revenue generated over time

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

	ORDER_DATE	CUM_REVENUE
1	01-JAN-15	2713.85
2	02-JAN-15	5445.75
3	03-JAN-15	8108.15
4	04-JAN-15	9863.6
5	05-JAN-15	11929.55
6	06-JAN-15	14358.5
7	07-JAN-15	16560.7
8	08-JAN-15	19399.05
9	09-JAN-15	21526.4
10	10-JAN-15	23990.35
11	11-JAN-15	25862.65
12	12-JAN-15	27781.7
13	13-JAN-15	29831.3
14	14-JAN-15	32358.7
15	15-JAN-15	34343.5
16	16-JAN-15	36937.65
17	17-JAN-15	39001.75
18	18-JAN-15	40978.6
19	19-JAN-15	43365.75
20	20-JAN-15	45763.65
21	21-JAN-15	47804.2

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

```sql
with a as
(select
    t.category,
    t.name,
    sum(p.price*d.quantity) as revenue
from
    pizzas p
join
    order_details d on p.pizza_id=d.pizza_id
join
    pizza_types t on p.pizza_type_id=t.pizza_type_id
group by
    t.name,
    t.category
order by
    revenue desc),

b as
(select
    a.category,
    a.name,
    a.revenue,
    rank() over( PARTITION by a.category order by a.revenue desc) as rank
from a)
select
    b.name,
    b.revenue,
    b.rank
from b
where rank <= 3;
```

	NAME	REVENUE	RANK
1	The Thai Chicken Pizza	43434.25	1
2	The Barbecue Chicken Pizza	42768	2
3	The California Chicken Pizza	41409.5	3
4	The Classic Deluxe Pizza	38180.5	1
5	The Hawaiian Pizza	32273.25	2
6	The Pepperoni Pizza	30161.75	3
7	The Spicy Italian Pizza	34831.25	1
8	The Italian Supreme Pizza	33476.75	2
9	The Sicilian Pizza	30940.5	3
10	The Four Cheese Pizza	32265.7	1
11	The Mexicana Pizza	26780.75	2
12	The Five Cheese Pizza	26066.5	3

Data Cleaning and Preparation

Before analysis, data must be cleaned and prepared. This involves removing **duplicates**, correcting errors, and formatting data consistently. Proper preparation is critical as it enhances the reliability of our SQL queries and the insights derived from them.

Key SQL Queries for Analysis

SQL queries for analysis commonly involve aggregation functions like COUNT, SUM, AVG, and ROUND. You can filter data with WHERE, use JOINs to combine tables, and group data with GROUP BY. For advanced analysis, techniques like RANK, PARTITION, and window functions (e.g., OVER) help organize and calculate results.

Developing Strategic Recommendations

Based on our analysis, we can formulate strategic recommendations. These may include adjusting **menu offerings**, targeted marketing campaigns, and optimizing pricing strategies. Data-driven decisions lead to improved customer satisfaction and increased sales.

Challenges in Data Analysis

While analyzing pizza sales data, we may encounter challenges such as **data quality issues**, integration of disparate data sources, and the need for advanced analytical skills. Addressing these challenges is essential for successful data-driven decision-making.

Conclusion and Future Directions

In conclusion, this project utilized Oracle SQL Developer to analyze pizza sales trends effectively. Advanced SQL queries provided actionable insights, demonstrating the power of data-driven strategies to support informed decision-making and future growth.

Thanks!

Do you have any questions?

namajivivek@email.com

+91 8019449739

linkedin.com/in/namaji-vivek