Pizza Sales Report

A complete SQL based project

Introduction

Welcome to the Pizza Sales Project presentation. This analysis focuses on understanding and optimizing pizza sales performance through data-driven insights. Our dataset includes detailed records of transactions, customer preferences, and product performance.

Project Objective

Analyze Sales Trends:

Identify peak sales periods and best-selling pizza varieties.

Customer Preferences:

Understand customer demographics and preferences.

Revenue Insights:

Evaluate revenue patterns and highlight high-performing products.

Operational Efficiency:

Optimize inventory and supply chain based on sales data.

Basic Level Queries

1.Retrieve the total number of orders placed.

```
SELECT COUNT(order_id) AS Total_Orders
FROM   orders;
```

Results Messages

	Total_Orders
1	21350

2.Calculate the total revenue generated from pizza sales.

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

	Total_Revenue
1	817860.05

3.Identify the highest-priced pizza.

```
SELECT TOP (1) pt.name, ROUND(MAX(p.price), 2) AS Highest_price_pizza
FROM   pizza_types AS pt INNER JOIN
           pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY Highest_price_pizza DESC
```

	name	Highest_price_pizza
1	The Greek Pizza	35.95

4.Identify the most common pizza size ordered.

```
SELECT p.size, COUNT(pd.quantity) AS total_quant
FROM   pizzas AS p INNER JOIN
            order_details AS pd ON p.pizza_id = pd.pizza_id
GROUP BY p.size
ORDER BY total_quant DESC
```

Results | Messages

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

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

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

	name	total_quant
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

Intermediate Level Queries

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

```
SELECT pt.category, ROUND(SUM(od.quantity), 2) AS Total_quantity
FROM   pizza_types AS pt INNER JOIN
           pizzas AS p ON pt.pizza_type_id = p.pizza_type_id INNER JOIN
           order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY Total_quantity DESC
```

Results Messages

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

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

```
SELECT DATEPART(hour, time) AS hrs,
COUNT(order_id) AS Total_orders
FROM   orders
GROUP BY DATEPART(hour, time)
ORDER BY Total_orders DESC
```

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

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

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

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

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

```
SELECT AVG(Avg_count_pizzas) AS pizzas_per_day
FROM   (SELECT o.date AS Order_date, SUM(od.quantity) AS Avg_count_pizzas
        FROM   orders AS o INNER JOIN
               order_details AS od ON o.order_id = od.order_id
        GROUP BY o.date) AS Order_quantity;
```

	pizzas_per_day
1	138

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

```sql
SELECT TOP (3) pt.name, SUM(od.quantity * p.price) AS Revenue
FROM   pizza_types AS pt INNER JOIN
            pizzas AS p ON p.pizza_type_id = pt.pizza_type_id INNER JOIN
            order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY Revenue DESC
```

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

Advanced Level Queries

1.Calculate the percentage contribution of each pizza type to total revenue.(pizza name wise %)

```
SELECT pt.name, ROUND(SUM(od.quantity * p.price) /
            (SELECT ROUND(SUM(od.quantity * p.price), 2) AS Expr1
            FROM    order_details AS od INNER JOIN
                    pizzas AS p ON p.pizza_id = od.pizza_id) * 100, 2)
                    AS Revenue_Percentage
FROM   pizza_types AS pt INNER JOIN
        pizzas AS p ON p.pizza_type_id = pt.pizza_type_id INNER JOIN
        order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY Revenue_Percentage DESC
```

	name	Revenue_Percentage
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
10	The Sicilian Pizza	3.78
11	The Pepperoni Pizza	3.69
12	The Greek Pizza	3.48
13	The Mexicana Pizza	3.27
14	The Five Cheese Pizza	3.19
15	The Pepper Salami Pizza	3.12
16	The Italian Capocollo Pizza	3.07
17	The Vegetables + Vegetabl...	2.98
18	The Prosciutto and Arugula ...	2.96
19	The Napolitana Pizza	2.95
20	The Spinach and Feta Pizza	2.85
21	The Big Meat Pizza	2.81
22	The Pepperoni, Mushroom, ...	2.3
23	The Chicken Alfredo Pizza	2.07
24	The Chicken Pesto Pizza	2.04
25	The Soppressata Pizza	2.01
26	The Italian Vegetables Pizza	1.96
27	The Calabrese Pizza	1.95
28	The Spinach Pesto Pizza	1.91
29	The Mediterranean Pizza	1.88
30	The Spinach Supreme Pizza	1.87
31	The Green Garden Pizza	1.71
32	The Brie Carre Pizza	1.42

2.Calculate the percentage contribution of each pizza type to total revenue.(pizza category wise %)

```
SELECT pt.category, ROUND(SUM(od.quantity * p.price) /
                (SELECT ROUND(SUM(od.quantity * p.price), 2) AS Expr1
                FROM    order_details AS od INNER JOIN
                                pizzas AS p ON p.pizza_id = od.pizza_id) * 100, 2)
                                AS Revenue_Percentage
FROM   pizza_types AS pt INNER JOIN
            pizzas AS p ON p.pizza_type_id = pt.pizza_type_id INNER JOIN
            order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY Revenue_Percentage DESC
```

	category	Revenue_Percentage
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

3.Analyze the cumulative revenue generated over time.

```sql
select  date,sum(revenue) over (order by date) as cum_revenue
from
(select o.date,round(sum(od.quantity*p.price),2) as revenue
from order_details od  inner join pizzas p
 on od.pizza_id=p.pizza_id
 inner join
 orders o
 on o.order_id=od.order_id
 group by o.date) as Sales
```

	date	cum_revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.3
14	2015-01-14	32358.7
15	2015-01-15	34343.5
16	2015-01-16	36937.65
17	2015-01-17	39001.75
18	2015-01-18	40978.6
19	2015-01-19	43365.75
20	2015-01-20	45763.65
21	2015-01-21	47804.2
22	2015-01-22	50300.9
23	2015-01-23	52724.6
24	2015-01-24	55013.85
25	2015-01-25	56631.4
26	2015-01-26	58515.8
27	2015-01-27	61043.85
28	2015-01-28	63059.85
29	2015-01-29	65105.15
30	2015-01-30	67375.45
31	2015-01-31	69793.3
32	2015-02-01	72982.5
33	2015-02-02	75311.1
34	2015-02-03	77925.9
35	2015-02-04	80159.8
36	2015-02-05	82375.6
37	2015-02-06	84885.55
38	2015-02-07	87123.2
39	2015-02-08	89158.2
40	2015-02-09	91353.55
41	2015-02-10	93410.05
42	2015-02-11	95870.05
43	2015-02-12	98028.85
44	2015-02-13	100783.35
45	2015-02-14	103102.5
46	2015-02-15	105243.75
47	2015-02-16	107212.55
48	2015-02-17	109334.45
49	2015-02-18	111977.3
50	2015-02-19	114007.55

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

```
select category,name,Revenue from
(select category,name,Revenue,
rank() over(partition by category order by Revenue desc) as rn from
(select pt.category,pt.name,ROUND(sum(od.quantity*p.price),2) as Revenue
from pizza_types pt
inner join pizzas p on p.pizza_type_id=pt.pizza_type_id
inner join order_details od on od.pizza_id=p.pizza_id
group by pt.category,pt.name) as a) as b where rn<=3;
```

Results | Messages

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

Key Insights

Sales Peaks:
Identified weekends and holidays as peak sales periods.

Popular Pizzas:
Margherita and Pepperoni are the top-selling pizzas.

Customer Segmentation:
Regular customers contribute significantly to sales.

Revenue Boosters:
Promotions and combo offers drive higher sales.

Tools & Techniques

Data Extraction and Cleaning:
Ensuring data accuracy and consistency.

Data Analysis:
Leveraging SQL for querying sales data and generating insights.

Thank you