PIZZA SALES ANALYSIS REPORT

PRESENTATION

Introduction of Project

The Pizza Sales Analysis Project is a structured SQL-based data exploration initiative focused on uncovering key insights from a fictional pizzeria's order data. Using powerful SQL queries, the project examines sales trends, revenue drivers, popular items, and time-based ordering patterns to support strategic decision-making for product, pricing, and operations.
It simulates the real-world business needs of a fast-growing pizza chain aiming to understand its best sellers, customer preferences, peak sales hours, and category performance to improve profitability and customer satisfaction.

Dataset Structure

Table Name	Description
orders	Contains order timestamps and unique order IDs.
order_details	Line items of each order, including pizza IDs and quantities.
pizzas	Information on each pizza including size, type ID, and price.
pizza_types	Defines pizza names and their categories (e.g., Classic, Veggie, Chicken).

Source: Kaggle – Pizza Sales Dataset

Key Features Analysis

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
 ROUND(SUM(order_details.quantity * CAST(pizzas.price AS
FLOAT)), 2) AS Total_Revenue_Generated
FROM order_details
JOIN pizzas
 ON CAST(pizzas.pizza_id AS VARCHAR(MAX)) =
CAST(order_details.pizza_id AS VARCHAR(MAX));
```

	Total_Revenue_Generated
1	817860.05

Identify the highest-priced pizza

```
select
 top 1 pizza_types.name Name_of_pizza,
 pizzas.price Price_of_pizza
from pizza_types
join pizzas
 on pizza_types.pizza_type_id = CAST(pizzas.pizza_type_id AS
VARCHAR(MAX))
order by pizzas.price desc;
```

	Name_of_pizza	Price_of_pizza
1	The Greek Pizza	35.95

Identify the most common pizza size ordered

```
SELECT
 pizzas.size Size_of_pizza,
    COUNT(order_details.pizza_id) AS No_of_times_ordered
FROM pizzas
JOIN order_details
    ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
order by No_of_times_ordered desc;
```

	Size_of_pizza	No_of_times_ordered
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

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

```
SELECT
 top 5
 pizza_types.name Name_of_pizza,
    SUM(order_details.quantity) AS Quantity_ordered
FROM pizzas
JOIN order_details
    ON pizzas.pizza_id = order_details.pizza_id
JOIN pizza_types
 ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
order by Quantity_ordered desc;
```

	Name_of_pizza	Quantity_ordered
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

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

```
SELECT
 pizza_types.category Category_of_pizza,
    SUM(order_details.quantity) AS Quantity_ordered
FROM pizzas
JOIN order_details
    ON pizzas.pizza_id = order_details.pizza_id
JOIN pizza_types
 ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
order by Quantity_ordered desc;
```

	Category_of_pizza	Quantity_ordered
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Determine the distribution of orders by hour of the day

```
SELECT
    DATEPART(HOUR, order_time) AS
Hour_during_which_ordered,
    COUNT(order_id) AS Total_Orders
FROM orders
GROUP BY DATEPART(HOUR, order_time)
ORDER BY Total_Orders desc;
```

	Hour_during_which_ordered	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

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

```
select
 pizza_types.category Category_of_pizza,
 count(pizza_types.name) Total_pizzas_ordered
from pizza_types
group by pizza_types.category
order by Total_pizzas_ordered desc;
```

	Category_of_pizza	Total_pizzas_ordered
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6

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

```
select round(avg(quantity),0) Average_Pizzas_Ordered
from
(select
 orders.order_date order_date,
 sum(order_details.quantity) as quantity
from orders
JOIN order_details
on orders.order_id=order_details.order_id
group by orders.order_date) as ordered_quantity;
```

	Average_Pizzas_Ordered
1	138

Determine the top 3 most ordered pizza types based on revenue

```
SELECT
 TOP 3
 pizza_types.name Name_of_pizza,
 ROUND(SUM(order_details.quantity * CAST(pizzas.price AS
FLOAT)), 2) AS Revenue_Generated
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 pizza_types.name
order by Revenue_Generated desc;
```

	Name_of_pizza	Revenue_Generated
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue

```
SELECT
 pizza_types.category Category_of_pizza,
 ROUND(SUM(order_details.quantity * CAST(pizzas.price AS
FLOAT))/
 (SELECT
 ROUND(SUM(order_details.quantity * CAST(pizzas.price AS
FLOAT)), 2) AS Total_Revenue_Generated
  FROM order_details
  JOIN pizzas
   ON CAST(pizzas.pizza_id AS VARCHAR(MAX)) =
CAST(order_details.pizza_id AS VARCHAR(MAX)))*100,2)
 AS Revenue_Generated
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 pizza_types.category
order by Revenue_Generated desc;
```

	Category_of_pizza	Revenue_Generated
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

Analyze the cumulative revenue generated over time

```
select
 Date_of_order,
 sum(Revenue_Generated) over(order by Date_of_order) Revenue_Generated
from
(SELECT
 orders.order_date Date_of_order,
 SUM(order_details.quantity * CAST(pizzas.price AS FLOAT)) AS
Revenue_Generated
FROM order_details
JOIN pizzas
 ON pizzas.pizza_id=order_details.pizza_id
JOIN orders
 ON order_details.order_id=orders.order_id
group by orders.order_date) as sales;
```

	Date_of_order	Revenue_Generated
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

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

```
select
 Category_of_pizza,
 Name_of_pizza,
 Revenue_Generated
from
(select
 Name_of_pizza,
 Category_of_pizza,
 Revenue_Generated,
 rank() over(partition by Category_of_pizza order by Revenue_Generated desc) Ranks
from
(select
 pizza_types.name Name_of_pizza,
 pizza_types.category Category_of_pizza,
 SUM(order_details.quantity * CAST(pizzas.price AS FLOAT)) AS Revenue_Generated
from pizzas
JOIN pizza_types
on pizzas.pizza_type_id=pizza_types.pizza_type_id
JOIN order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.name,pizza_types.category) as sales) as a
where Ranks<=3;
```

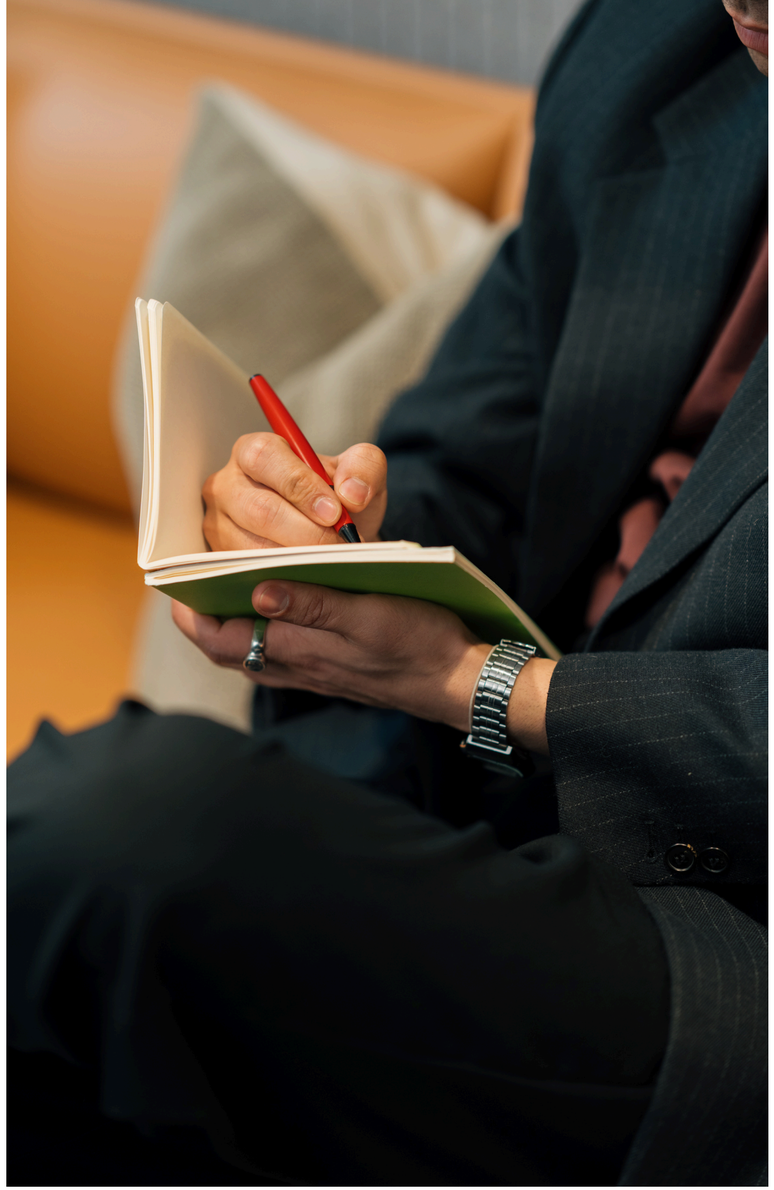
	Category_of_pizza	Name_of_pizza	Revenue_Generated
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
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	32265.7000000006
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5

Insights & Reflection

✅ This project provided a comprehensive understanding of business-oriented data analysis using SQL and Power BI.
🔍 From sales performance to customer behavior and product strategy, we extracted actionable insights that mirror real-world retail decision-making.
🎯 The process enhanced my skills in:

- Writing optimized SQL queries
- Visualizing KPIs in Power BI
- Interpreting trends to support business goals

🚀 This hands-on analysis reflects how data can drive smarter decisions in fast-moving industries like food & beverage.



🔍 Analyzed by Gurleen Saini | ⚙️ SQL • Power BI
🔗 github.com/gurleen-saini/Pizza-Sales-Analysis-SQL