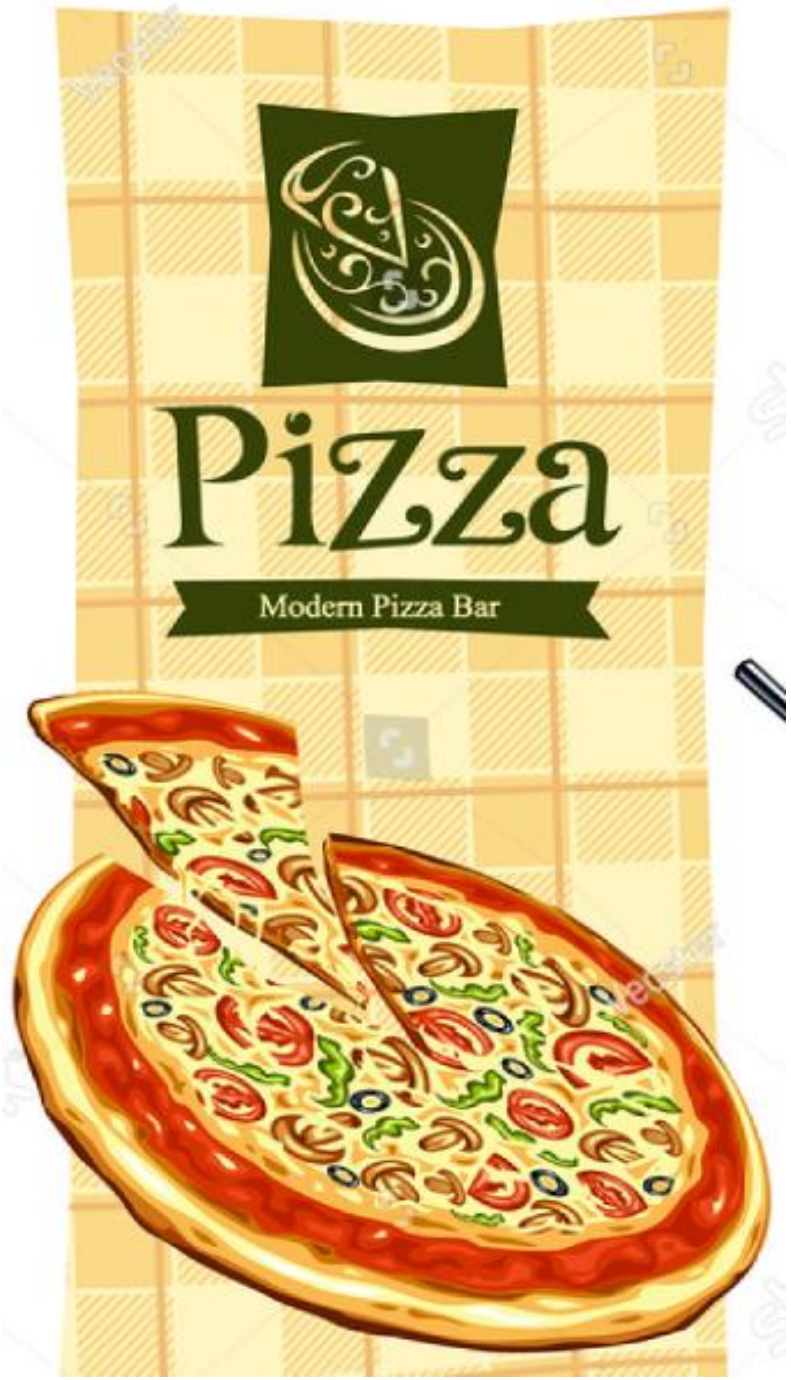

PizzaHub

PIZZA SALES ANALYSIS USING SQL

About

This project involves analyzing pizza sales data to derive actionable insights using SQL.

The dataset includes details on orders, customers, pizzas, and sales transactions.

The analysis covers various aspects such as identifying the most popular pizza types and sizes, peak sales times, revenue trends, and customer preferences.

Key SQL techniques such as data cleaning, aggregations, joins, and window functions were employed to generate insights.

The project aims to provide a clear understanding of sales performance, enhance decision-making for inventory management, and improve marketing strategies for the pizza business.

Preview Datasets

PIZZAS

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75
ckn_alfredo_m	ckn_alfredo	M	16.75
ckn_alfredo_l	ckn_alfredo	L	20.75
ckn_pesto_s	ckn_pesto	S	12.75

PIZZA_TYPES

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe...
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpclo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Chee...

ORDERS

Order_id	Order_data	Order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37
8	2015-01-01	12:51:37
9	2015-01-01	12:52:01
10	2015-01-01	13:00:15

ORDER_DETAILS

Order_details_id	Order_id	Pizza_id	Quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1

Questions

Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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

	Total_Orders
▶	21350

WE PLACED 21,350 NO OF ORDERS

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

	Total_Revenue
▶	817860.05

TOTAL REVENUE GENERATED IS 8,17,860/- RS

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1
```

	name	price
▶	The Greek Pizza	35.95

OUR HIGHEST PRICED PIZZA IS ‘THE GREEK PIZZA’

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

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

THE MOST ORDERED PIZZA IS OF ‘LARGE’ SIZE

```sql
SELECT
pizza_types.name, SUM(order_details.Quantity) AS 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 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

HERE IS THE TOP 5 MOST ORDERED PIZZAS

```sql
SELECT
    pizza_types.category,
    SUM(order_details.Quantity) AS 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 pizza_types.category
ORDER BY quantity DESC
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

HERE IS THE TOTAL QUANTITY OF EACH PIZZA CATEGORY

SO, WE CAN SAY MOST ORDERED PIZZAS ARE OF CLASSIC TYPE

```sql
SELECT
    HOUR(order_time) AS Time, COUNT(order_id) AS No_of_Orders
FROM
    orders
GROUP BY HOUR(order_time)
```

	Time	No_of_Orders
▶	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

FROM 12PM TO 1PM WE GET MAXIMUM NO OF ORDERS

```sql
Select category, count(name) from pizza_types
group by category
```

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

HERE IS THE CATEGORY WISE DISTRIBUTION OF PIZZAS

```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.Order_data, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.Order_id = order_details.Order_id
    GROUP BY orders.Order_data) AS order_quantity;
```

	round(avg(quantity),0)
▶	138

138 IS THE AVERAGE NO OF PIZZAS ORDERED PER DAY

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.Quantity) AS revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.Pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

THESE 3 PIZZAS GENERATE HIGHEST REVENUE

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.Quantity) / (SELECT
                    ROUND(SUM(order_details.Quantity * pizzas.price),
                                2) AS Total_Revenue
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS Revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.Pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
```

	category	Revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

CLASSIC CATEGORY PIZZA GENERATE REVENUE OF 26.91%

VEGGIE CATEGORY PIZZA GENERATE REVENUE OF 23.68%

SUPREME CATEGORY PIZZA GENERATE REVENUE OF 25.46%

CHICKEN CATEGORY PIZZA GENERATE REVENUE OF 23.96%

```sql
select order_data,
round(sum(revenue) over(order by order_data),2) as cummulative_revenue
from
(select orders.order_data,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_data) as sales;
```

order_data	cummulative_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35

HERE IS THE CUMULATIVE REVENUE GENERATED OVER A DAY

```sql
select
category, name, revenue
from
(select category, name, revenue, rank() over(partition by category order by revenue desc) as ord_revenue
from
(select pizza_types.category, pizza_types.name, sum((order_details.quantity) * pizzas.price) as revenue
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 pizza_types.category, pizza_types.name) as a) as b
where ord_revenue <=3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

HERE WE HAVE TOP 3 BEST SELLING PIZZAS OF EACH CATEGORY WHICH GENERATE THE HIGHEST REVENUE

Conclusion

By leveraging the findings from the analysis, the pizza store can enhance its sales and customer satisfaction, ultimately leading to increased revenue.