SQL PROJECT ON

ANALYSIS OF PIZZA SALES

UNDERSTANDING TRENDS, REVENUE AND CONSUMER PREFERENCES

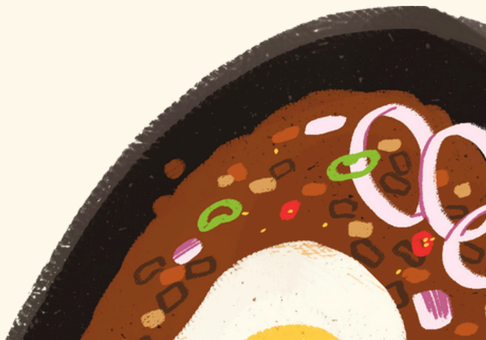
Sudipta Ghosh

INTRODUCTION

This project focuses on analyzing pizza sales data to gain insights into customer preferences, revenue trends, and top-performing categories. By examining detailed sales records and related data, we aim to understand what drives success in the pizza business.

OBJECTIVES:

- Understand sales trends.
- Identify top-performing categories.
- Analyse revenue generation.
- Scope: Time frame, data sources, and target audience.



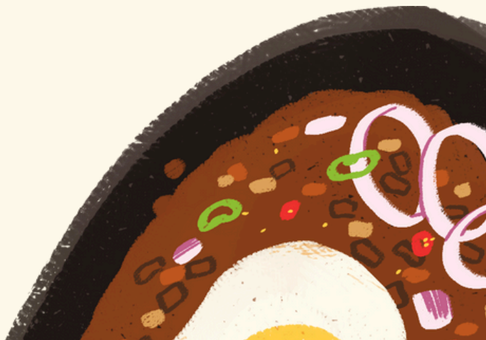
DATA SOURCES

DATA COLLECTION:

TYPES OF DATA:-

- Pizza_types
- Orders
- Orders_details
- Pizzas

TOOLS USED: MYSQL



DATA SETS

pizza

pizza_id	pizza_type_id	size	Price
---	---	---	---

orders

order_id	order_date	order_time
---	---	---

order_details

order_details_id	order_id	pizza_id	quantity
---	---	---	---

pizza_types

pizza_type_id	name	category	ingredients
---	---	---	---

QUE: Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(orders.order_id) AS total_order
FROM
    orders
```

#	total_order
1	21350

QUE: Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(pizzas.price * orders_details.quantity),
            2) AS revenue
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

#	revenue
1	817860.05

QUE: Identify the highest-priced pizza.

```sql
SELECT
    p2.name, p1.price
FROM
    pizzas p1
        JOIN
    pizza_types p2 ON p1.pizza_type_id = p2.pizza_type_id
ORDER BY p1.price DESC
LIMIT 1
```

#	name
1	The Greek Pizza

QUE: Identify the most common pizza size ordered.

```sql
SELECT
    p1.size, count(p2.order_details_id) AS count_order
FROM
    pizzas p1
        JOIN
    orders_details p2 ON p1.pizza_id = p2.pizza_id
GROUP BY p1.size
ORDER BY count_order DESC
```

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

QUE: List the top 5 most ordered pizza types(pizza name) along with their quantities.

```sql
SELECT
    p3.name, SUM(p1.quantity) AS total_quantity
FROM
    orders_details p1
        JOIN
    pizzas p2 ON p1.pizza_id = p2.pizza_id
        JOIN
    pizza_types p3 ON p2.pizza_type_id = p3.pizza_type_id
GROUP BY p3.name
ORDER BY total_quantity DESC
LIMIT 5
```

#	name	total_quantit
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

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

```sql
SELECT
    p1.category, SUM(p3.quantity) AS quantity
FROM
    pizza_types p1
        JOIN
    pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
        JOIN
    orders_details p3 ON p2.pizza_id = p3.pizza_id
GROUP BY p1.category
ORDER BY quantity DESC
```

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

QUE: Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(orders.order_time) AS hours,
    COUNT(orders.order_id) AS total_order
FROM
    orders
GROUP BY hours
```

#	hours	total_order
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	9	1

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

```sql
SELECT
    p1.category, SUM(p3.quantity) AS total_no
FROM
    pizza_types p1
        JOIN
    pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
        JOIN
    orders_details p3 ON p2.pizza_id = p3.pizza_id
GROUP BY p1.category
```

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

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

```sql
SELECT
    ROUND(AVG(total_quantity.total_pizza_ordered),
            2) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        o1.order_date, SUM(o2.quantity) AS total_pizza_ordered
    FROM
        orders o1
    JOIN orders_details o2 ON o1.order_id = o2.order_id
    GROUP BY o1.order_date) AS total_quantity
```

#	avg_pizza_ordered_per_day
1	138.47

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

```sql
SELECT
    p1.name, SUM(p2.price * p3.quantity) AS revenue
FROM
    pizza_types p1
        JOIN
    pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
        JOIN
    orders_details p3 ON p2.pizza_id = p3.pizza_id
GROUP BY p1.name
ORDER BY revenue DESC
LIMIT 3
```

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

QUE: Calculate the percentage contribution of each pizza category to total revenue.

```sql
SELECT
    p3.category,
    (SUM(p1.price * p2.quantity) / (SELECT
            SUM(q1.price * q2.quantity)
        FROM
            pizzas q1
                JOIN
            orders_details q2 ON q1.pizza_id = q2.pizza_id)) * 100 AS percentage_wise_revenue
FROM
    pizzas p1
        JOIN
    orders_details p2 ON p1.pizza_id = p2.pizza_id
        JOIN
    pizza_types p3 ON p1.pizza_type_id = p3.pizza_type_id
GROUP BY p3.category
ORDER BY percentage_wise_revenue DESC
```

#	category	percentage_wise_revenu
1	Classic	26.905960255669903
2	Supreme	25.45631126009884
3	Chicken	23.955137556847493
4	Veggie	23.682590927384783

QUE: Analyse the cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over(order by order_date) as cumulative_revenue
from
(SELECT
    p3.order_date, SUM(p1.quantity * p2.price) AS revenue
FROM
    orders_details p1
        JOIN
    pizzas p2 ON p1.pizza_id = p2.pizza_id
        JOIN
    orders p3 ON p3.order_id = p1.order_id
GROUP BY p3.order_date) as sales
```

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

```sql
select category, name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(SELECT
    pizza_types.category,
    pizza_types.name,
    SUM(pizzas.price * orders_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

#	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	32265.70000000065
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5

RECOMMENDATIONS

- Focus on best-selling pizzas and up sell add-ons.
- Top categories and revenue generators.
- Improve delivery times and packaging for better customer experience.
- Invest in targeted marketing during peak periods.

Thank You