Where Every Slice is a Taste of Perfection

PIZZA HUT SALES ANALYSIS

- Pranav Pawar

OBJECTIVE

Utilized SQL to analyze one year of pizza sales data, extracting valuable insights for informed decision-making.

The project aimed to identify key trends, customer preferences, and sales patterns, providing a foundation for optimizing inventory, and operational efficiency to drive future growth and profitability

PROBLEM STATEMENT

Basic:

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

Intermediate:

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

Advanced:

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

LET'S BEGIN ANALYSIS USING

MySQL®

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

QUERY

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

OUTPUT

total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

QUERY

```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;
```

OUTPUT

	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

QUERY

```
select pizza_types.name, pizzas.price from pizza_types join pizzas
on pizzas.pizza_type_id=  pizza_types.pizza_type_id order by
pizzas.price desc limit 1;
```

OUTPUT

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

QUERY

```sql
select pizzas.size, count(orders_details.order_details_id) as pizza_count
from pizzas join orders_details on orders_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by count(orders_details.order_details_id) desc
limit 1;
```

OUTPUT

size	pizza_count
L	18526

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

QUERY

```
SELECT PIZZAS.PIZZA_TYPE_ID,SUM(ORDERS_DETAILS.QUANTITY)
FROM ORDERS_DETAILS
JOIN PIZZAS ON ORDERS_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
GROUP BY PIZZAS.PIZZA_TYPE_ID
ORDER BY SUM(ORDERS_DETAILS.QUANTITY) DESC
LIMIT 5;
```

OUTPUT

PIZZA_TYPE_ID	SUM(ORDERS_DETAILS.QUANTITY)
classic_dlx	2453
bbq_ckn	2432
hawaiian	2422
pepperoni	2418
thai_ckn	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

QUERY

```sql
SELECT pizza_types.category,sum(orders_details.quantity) from pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details on pizzas.pizza_id=orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY sum(orders_details.quantity) DESC;
```

OUTPUT

category	sum(orders_details.quantity)
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

11649

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

QUERY

```
SELECT HOUR(ORDER_TIME),COUNT(ORDER_ID)FROM ORDERS
GROUP BY HOUR(order_time) ORDER BY COUNT(ORDER_ID) DESC;
```

OUTPUT

HOUR(ORDER_TIME)	COUNT(ORDER_ID)
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

QUERY

```
SELECT CATEGORY,count(name)FROM pizza_types group by category;
```

OUTPUT

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

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

QUERY

```sql
SELECT ROUND(AVG(QUANTITY),0) FROM
(SELECT ORDERS.ORDER_DATE,
SUM(ORDERS_DETAILS.QUANTITY) AS QUANTITY FROM orders
JOIN orders_details ON orders_details.ORDER_ID = ORDERS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS ORDERS_QUANTITY;
```

OUTPUT

	ROUND(AVG(QUANTITY),0)
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

QUERY

```sql
SELECT SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) AS TOTAL_REVENUE,
PIZZA_TYPE_ID FROM orders_details
JOIN PIZZAS ON orders_details.pizza_id= pizzas.pizza_id
GROUP BY PIZZA_TYPE_ID ORDER BY TOTAL_REVENUE DESC LIMIT 3;
```

OUTPUT

TOTAL_REVENUE	PIZZA_TYPE_ID
43434.25	thai_ckn
42768	bbq_ckn
41409.5	cali_ckn

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

QUERY

```sql
select pizza_types.category,
round(sum(orders_details.quantity * pizzas.price) /
(SELECT ROUND (SUM(orders_details.quantity * pizzas.price),2)
AS total_sales from
orders_details
JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id)*100,2)
from pizza_types join pizzas
on pizza_types.pizza_type_id= pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category;
```

OUTPUT

Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY

```sql
SELECT ORDER_DATE,SUM(REVENUE) OVER (ORDER BY ORDER_DATE) AS CUM_REVENUE
FROM
(SELECT ORDERS.ORDER_DATE,
SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM ORDERS_DETAILS JOIN PIZZAS
ON ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
JOIN ORDERS
ON ORDERS.ORDER_ID = ORDERS_DETAILS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS SALES;
```

OUTPUT

ORDER_DATE	CUM_REVENUE
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

QUERY

```sql
SELECT CATEGORY,NAME,REVENUE,RANKS FROM
(SELECT CATEGORY,name,REVENUE,
RANK() OVER (partition by CATEGORY ORDER BY REVENUE DESC) AS RANKS FROM
(SELECT PIZZA_TYPES.CATEGORY,PIZZA_TYPES.NAME,
SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE FROM
PIZZAS JOIN PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.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 RANKS <= 3 ;
```

OUTPUT

CATEGORY	NAME	REVENUE	RANKS
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

KEY TAKEAWAYS

1) XXL Pizza Size Underperformance:
Out of 21,350 orders, XXL pizzas were ordered only 28 times. It's recommended to discontinue this size and focus on more popular sizes.

2) Classic Pizza Popularity:
Among the four pizza categories, customers prefer the "Classic" category the most, making it a key area to focus on.

3) Order Timing Trend:
There are very few orders between 9 AM and 11 AM, with peak sales starting from 11 PM, likely due to people being busy during the day.

4): Chicken Pizza Category :
The chicken category has only 6 pizza types, leading to fewer sales. Expanding the variety of chicken pizzas, like other categories (supreme, veggies, classic), could boost sales.

5) Average Orders:
With an average of 138 pizzas sold per order, further advertising, discounts, or coupons could help increase this number and attract more customers.

THANK YOU FOR ATTENTION