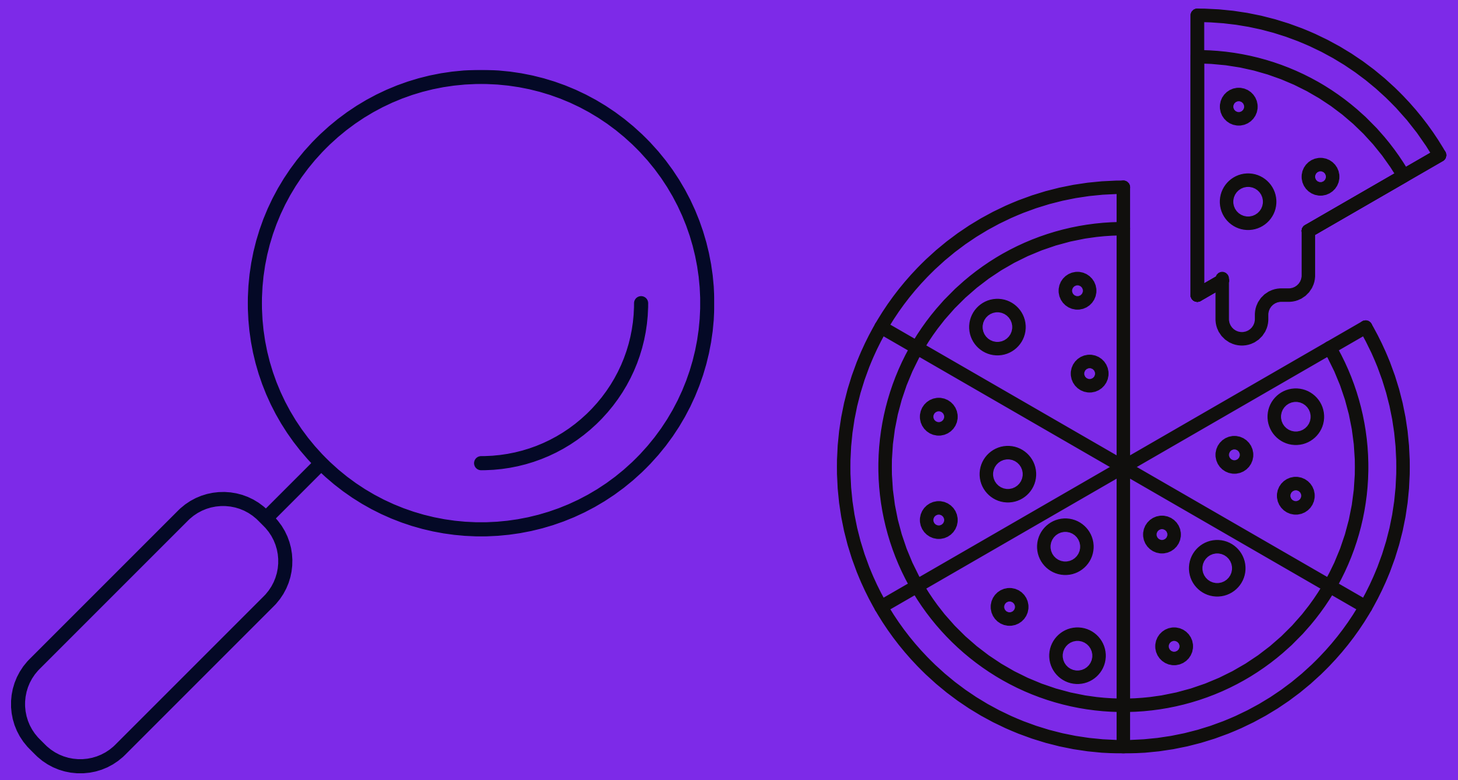
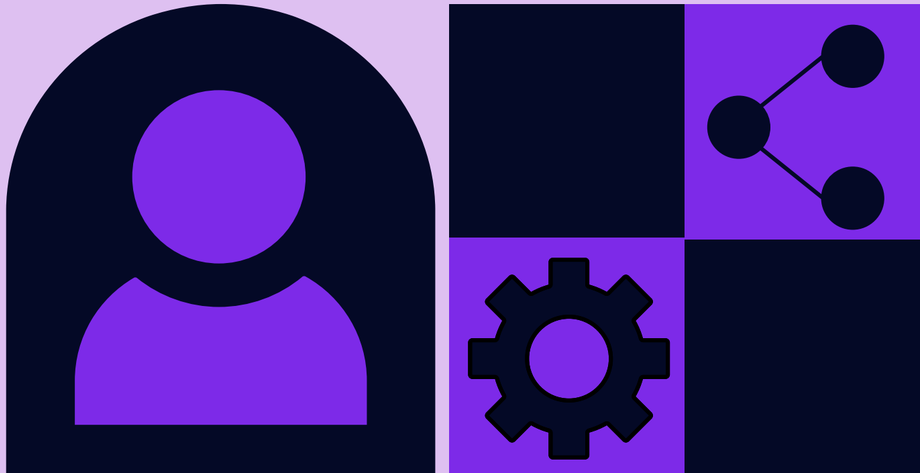


Mentorness Internship Program

Pizza Sales Analysis using SQL



Project overview

The main aim of this project is to analyze pizza sales to gain insights into customer behaviour, popular pizza types, sales trends and overall performance.

The analysis will be performed using four tables: order_details, orders, pizza_types and pizzas.

Dataset:

order_details

- order_details_id: Unique identifier for the order detail.
- order_id: Identifier linking to the orders table.
- pizza_id: Identifier linking to the pizza table.
- quantity: Number of pizzas ordered.

orders

- order_id: Unique identifier for the order.
- date: Date the order was placed.
- time: Time the order was placed.

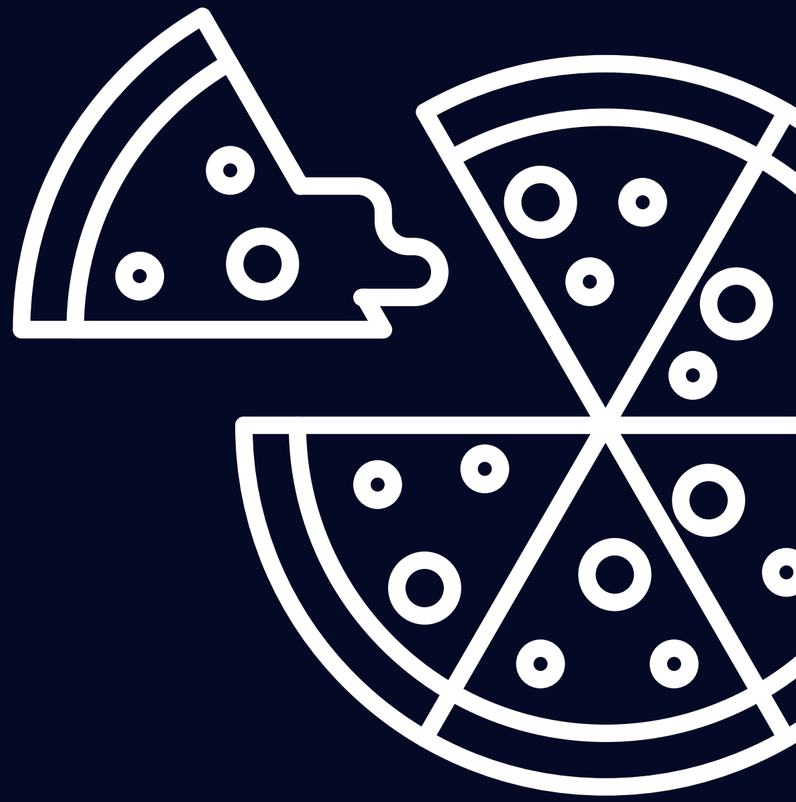
Dataset:

pizza_type

- pizza_type_id: Unique identifier for the pizza type.
- name: Name of the pizza.
- category: Category of the pizza(vegetarian, meat, etc.)
- ingredients: List of ingredients used in the pizza.

pizzas

- pizza_id: Unique identifier for the pizza.
- pizza_type_id: Identifier linking to the pizza_type table.
- size: Size of the pizza (e.g., small, medium, large).
- price: Price of the pizza.



QUERIES

Question 1
The total number of orders placed.

Query:

```
create database pizza;
use pizza;
#Q1
select COUNT(*) as Total_orders from orders;
```

Output:

	Total_orders
▶	21350

Question 2
The total revenue generated from pizza sales

Query:

```
#Q2
SELECT ROUND(SUM(O.QUANTITY*P.PRICE),1) AS Total_revenue FROM
ORDER_DETAILS O JOIN PIZZAS P ON O.PIZZA_ID=P.PIZZA_ID;
```

Output:

	Total_revenue
▶	817860

Question 3
The highest priced pizza.

Query:

```
#Q3
SELECT PT.NAME, P.PRICE FROM
PIZZA_TYPES PT JOIN PIZZAS P ON
PT.PIZZA_TYPE_ID=P.PIZZA_TYPE_ID
WHERE P.PRICE=(SELECT MAX(PRICE) FROM PIZZAS);
```

Output:

	NAME	PRICE
▶	The Greek Pizza	35.95

Question 4
The most common pizza size ordered.

Query:

```
#Q4
SELECT P.SIZE, COUNT(O.QUANTITY) AS 'NO. OF ORDERS' FROM
PIZZA_DETAILS P JOIN ORDER_DETAILS O ON
P.PIZZA_ID = O.PIZZA_ID
GROUP BY P.SIZE ORDER BY MAX(O.QUANTITY) DESC;
```

Output:

	SIZE	NO. OF ORDERS
▶	L	18526
	S	14137
	M	15385
	XL	544
	XXL	28

Question 5
The top 5 most ordered pizza types along their quantities.

Query:

```
#Q5
SELECT PT.NAME,SUM(O.QUANTITY) AS 'TOTAL QUANTITY'
FROM ORDER_DETAILS O JOIN PIZZAS P ON O.PIZZA_ID=P.PIZZA_ID
JOIN PIZZA_TYPES PT ON P.PIZZA_TYPE_ID=PT.PIZZA_TYPE_ID
GROUP BY PT.NAME ORDER BY 'TOTAL QUANTITY' LIMIT 5;
```

Output:

NAME	TOTAL QUANTITY
The Hawaiian Pizza	2422
The Classic Deluxe Pizza	2453
The Five Cheese Pizza	1409
The Italian Supreme Pizza	1884
The Mexicana Pizza	1484

Question 6
The quantity of each pizza categories ordered.

Query:

```
#6
SELECT PT.CATEGORY,SUM(O.QUANTITY) AS 'TOTAL QUANTITY'
FROM ORDER_DETAILS O JOIN PIZZAS P ON O.PIZZA_ID=P.PIZZA_ID
JOIN PIZZA_TYPES PT ON P.PIZZA_TYPE_ID=PT.PIZZA_TYPE_ID
GROUP BY PT.CATEGORY ORDER BY 'TOTAL QUANTITY';
```

Output:

CATEGORY	TOTAL QUANTITY
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Question 7
The distribution of orders by hours of the day.

Query:

```
#7
SELECT HOUR(TIME) AS 'HOUR OF THE DAY',
COUNT(DISTINCT ORDER_ID) AS 'NO. OF ORDERS' FROM ORDERS
GROUP BY HOUR(TIME);
```

Output:

HOUR OF THE DAY	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

Question 8
The category-wise distribution of pizzas.

Query:

```
#8
SELECT CATEGORY,COUNT(CATEGORY) AS 'NO. OF PIZZA' FROM PIZZA_TYPES
GROUP BY CATEGORY;
```

Output:

CATEGORY	NO. OF PIZZA
Chicken	6
Classic	8
Supreme	9
Veggie	9

Question 9
The average number of pizzas ordered per day.

Query:

```
#Q9
SELECT ROUND(AVG(PERDAY_TOTAL.TOTAL_QUANTITY)) AS 'AVG ORDERS PER DAY'
FROM (SELECT O.DATE, SUM(OD.QUANTITY) AS 'TOTAL_QUANTITY' FROM
ORDER_DETAILS OD JOIN ORDERS O ON O.ORDER_ID = OD.ORDER_ID
GROUP BY O.DATE) AS PERDAY_TOTAL;
```

Output:

	AVG ORDERS PER DAY
▶	138

Question 10
Top 3 most ordered pizza type based on revenue.

Query:

```
#Q10
SELECT PT.NAME, SUM(O.QUANTITY*P.PRICE) AS TOTAL_REVENUE FROM ORDER_DETAILS O
PIZZAS P ON O.PIZZA_ID=P.PIZZA_ID JOIN PIZZA_TYPES PT
ON P.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID GROUP BY PT.NAME
ORDER BY TOTAL_REVENUE DESC LIMIT 3;
```

Output:

NAME	TOTAL_REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Question 11
The percentage contribution of each pizza type to revenue.

Query:

```
#Q11
SELECT PT.CATEGORY,ROUND((SUM(O.QUANTITY*P.PRICE)/(SELECT ROUND
(SUM(O.QUANTITY*P.PRICE),2) AS TOTAL_REV FROM ORDER_DETAILS O
JOIN PIZZAS P ON P.PIZZA_ID=O.PIZZA_ID))*100,2) AS REV FROM
PIZZA_TYPES PT JOIN PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
JOIN ORDER_DETAILS O ON O.PIZZA_ID=P.PIZZA_ID
GROUP BY PT.CATEGORY ORDER BY REV DESC;
```

Output:

	CATEGORY	REV
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Question 12
The cumulative revenue generated over time.

Query:

```
#Q12
SELECT O.DATE,ROUND(SUM(OD.QUANTITY*P.PRICE),2)
AS DAILY_REVENUE,ROUND(SUM(SUM(OD.QUANTITY*P.PRICE))
OVER (ORDER BY O.DATE),2) AS CUMULATIVE_REVENUE
FROM ORDERS O JOIN ORDER_DETAILS OD ON
O.ORDER_ID=OD.ORDER_ID JOIN PIZZAS P ON
P.PIZZA_ID = OD.PIZZA_ID
GROUP BY O.DATE ORDER BY O.DATE;
```

Output:

	DATE	DAILY_REVENUE	CUMULATIVE_REVENUE
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.3	25862.65
	2015-01-12	1919.05	27781.7
	2015-01-13	2049.6	29831.3
	2015-01-14	2527.4	32358.7
	2015-01-15	1984.8	34343.5
	2015-01-16	2594.15	36937.65
	2015-01-17	2064.1	39001.75
	2015-01-18	1976.85	40978.6
	2015-01-19	2387.15	43365.75
	2015-01-20	2397.9	45763.65
	2015-01-21	2040.55	47804.2

Question 13

The top 3 most ordered pizza type based on revenue for each pizza category.

Query:

```
#Q13
SELECT CATEGORY,NAME,REVENUE FROM(
SELECT PT.CATEGORY,PT.NAME,ROUND(SUM(OD.QUANTITY*P.PRICE),2) AS REVENUE,
RANK() OVER (PARTITION BY PT.CATEGORY ORDER BY SUM(OD.QUANTITY*P.PRICE)DESC)
AS ranks FROM ORDER_DETAILS OD JOIN PIZZAS P ON OD.PIZZA_ID=
P.PIZZA_ID JOIN PIZZA_TYPES PT ON P.PIZZA_TYPE_ID=PT.PIZZA_TYPE_ID
GROUP BY PT.CATEGORY,PT.NAME) RANKED_PIZZAS WHERE ranks<=3
ORDER BY CATEGORY,ranks;
```

Output:

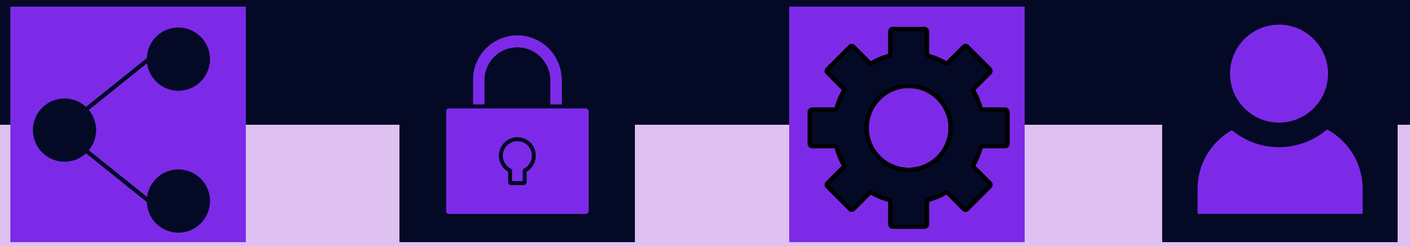
CATEGORY	NAME	REVENUE
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Conclusion

Through the sales analysis we explored various aspects of pizza orders, including customer preferences, overall sales trend, product and cumulative renvenue insights.

Based on these insights the pizza store can make informed decisions to enhance customer experience, drive sales and hence achieve sustained growth.

Thank you!



-Basvi Chunara