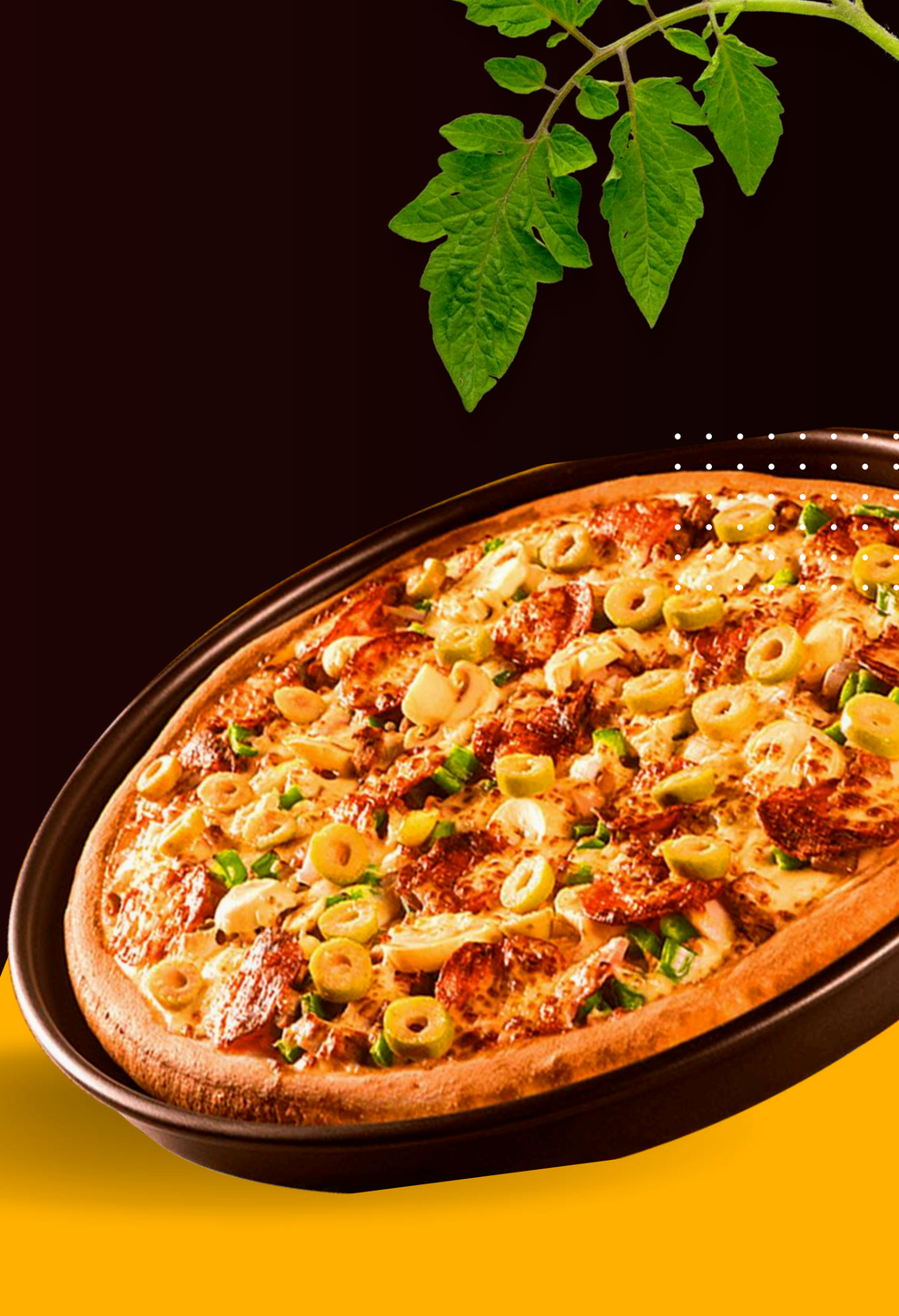
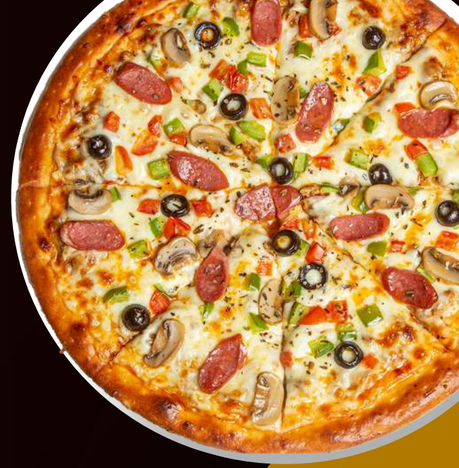
PIZZA SALES ANALYSIS



ABOUT ME



KOMAN MORUNGMAN MARING

I am currently pursuing 4th Year in B Tech in Chemical Science and Technology in Indian Institute of Technology, Guwahati, Assam, India. I am from Kangpokpi District, Manipur, India. I am deeply committed to data analytics, consistently leveraging data-driven insights to inform strategic decision-making and enhance organizational performance.

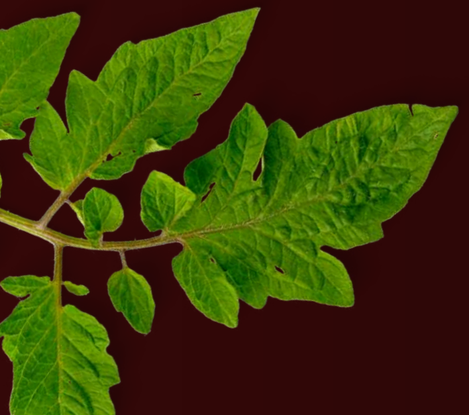
CONTACTS

Email: kmmorungman@gmail.com

LinkedIn: linkedin.com/in/man273

ABOUT THE PROJECT

This project is a comprehensive study of pizza sales data collected from a pizzeria. I utilize MySQL to manage and analyze the dataset, uncovering key insights into sales trends, customer preferences, and overall operational efficiency. The findings aim to support informed decision-making and drive meaningful improvements in business performance.



RAW DATA

The data consist of four tables as listed below

PIzza types	**Pizzas**	Orders	Order Details
• It consists of the types of the pizzas, their names and their ingredients • Link to table: https://drive.google.com/file/d/1wDSjojQ1tntfkcxnFUroPtO3pTQgfHDZ/view?usp=sharing	• It consists of the prices and sizes of each type of the pizzas • Link to table: https://drive.google.com/file/d/1VLZWi41aawnBnZYBCvNNeZG4RepRc4ZI/view?usp=sharing	• It consists of the time and dates of orders and their corresponding IDs • Link to table: https://drive.google.com/file/d/1Cx-FHD1Rj88Ix11uizAT1WSTmN73sAsN/view?usp=drive_link	• It consists of the types of pizza ordered and their quantities for each order ID. • Link to table: https://drive.google.com/file/d/1z6pqZ23xxM_-3O5iFNwNj1C8S-6AbnMn/view?usp=drive_link

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

MySQL Query

```
SELECT
    COUNT(order_id)
FROM
    orders;
```

Output

	COUNT(order_id)
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

MySQL Query

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2)
FROM
    order_details
        INNER JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Output

	Total_Revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

MySQL Query

```
SELECT
    pizza_types.name
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
WHERE
    pizzas.price = (SELECT
            MAX(pizzas.price)
        FROM
            pizzas);
```

Output

	name
▶	The Greek Pizza

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

MySQL Query

```
SELECT
    pizzas.size, COUNT(pizzas.size)
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(pizzas.size) DESC
LIMIT 1;
```

Output

	size	COUNT(pizzas.size)
▶	L	18526

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

MySQL Query

```
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 SUM(order_details.quantity) DESC
LIMIT 5;
```

Output

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

FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

MySQL Query

```
SELECT
    pizza_types.category,
    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.category;
```

Output

	category	Quantity
	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

MySQL Query

```
SELECT
    HOUR(time) AS Hour, COUNT(order_id) AS Quantity
FROM
    orders
GROUP BY HOUR(time)
ORDER BY COUNT(order_id) DESC;
```

Output

Hour	Quantity
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
10	8
9	1

CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

MySQL Query

```
SELECT
    ROUND(AVG(quantity), 0) AS Average_Order
FROM
    (SELECT
        b.date, SUM(a.quantity) AS quantity
    FROM
        order_details AS a
    JOIN orders AS b ON a.order_id = b.order_id
    GROUP BY b.date) AS per_day_quantity;
```

Output

Average_Order
138

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

MySQL Query

```
SELECT
    SUM(a.quantity * b.price) AS Sale, c.name
FROM
    order_details AS a
        JOIN
    pizzas AS b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types AS c ON b.pizza_type_id = c.pizza_type_id
GROUP BY (c.name)
ORDER BY Sale DESC
LIMIT 3;
```

Output

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

MySQL Query

```
Select date, round(sum(revenue) over (order by date),2) as cum_revenue from
(Select orders.date, sum(order_details.quantity*pizzas.price) as revenue
from pizzas join order_details on pizzas.pizza_id=order_details.pizza_id
join orders
on orders.order_id=order_details.order_id
group by orders.date) as revenue table;
```

Output

	A	B
1	date	cum_revenue
2	2015-01-01	2713.85
3	2015-01-02	5445.75
4	2015-01-03	8108.15
5	2015-01-04	9863.6
6	2015-01-05	11929.55
7	2015-01-06	14358.5
8	2015-01-07	16560.7
9	2015-01-08	19399.05
10	2015-01-09	21526.4
11	2015-01-10	23990.35
12	2015-01-11	25862.65
13	2015-01-12	27781.7
14	2015-01-13	29831.3
15	2015-01-14	32358.7
16	2015-01-15	34343.5
17	2015-01-16	36937.65
18	2015-01-17	39001.75
19	2015-01-18	40978.6
20	2015-01-19	43365.75
21	2015-01-20	45763.65
22	2015-01-21	47804.2
23	2015-01-22	50300.9
24	2015-01-23	52724.6
25	2015-01-24	55013.85
26	2015-01-25	56631.4
27	2015-01-26	58515.8
28	2015-01-27	61043.85
29	2015-01-28	63059.85
30	2015-01-29	65105.15
31	2015-01-30	67375.45
32	2015-01-31	69793.3
33	2015-02-01	72982.5
34	2015-02-02	75311.1
35	2015-02-03	77925.9
36	2015-02-04	80159.8

https://drive.google.com/file/d/1p7h5dp-QzwtQ9AV50L2drRkVXNeN7YuM/view?usp=drive_link

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

MySQL Query

```
Select category,name,revenue from
(Select category,name,revenue, rank() over (partition by category order by revenue desc) as Rnk from
(Select pizza_types.name,pizza_types.category,round(sum(order_details.quantity*pizzas.price),2) as revenue from
order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join
pizza_types
on pizzas.pizza_type_id=pizza_types.pizza_type_id
group by pizza_types.name,pizza_types.category) as a) as b
where Rnk<=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	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Output


THANK YOU