PIZZA SALES DATA ANALYSIS

USING MYSQL

PIZZA SALES DATA ANALYSIS

ABOUT ME

Hi, I'm Mohammad Junaid, a recent graduate with a strong interest in data analysis, business intelligence, and SQL development. I'm passionate about uncovering insights from data and continuously improving my analytical skills through hands-on projects.

ABOUT THE PROJECT

This project focuses on analyzing a Pizza Sales Dataset (Dummy) using MySQL. It includes a wide range of queries — from basic aggregations to advanced window functions — aimed at answering real-world business questions such as:

- Top-selling pizza types and sizes
- Hourly and daily order patterns
- Revenue breakdowns by category
- Cumulative sales trends and ranking-based insights

The goal was to deepen my SQL knowledge and showcase practical problem-solving using structured data.

SCHEMA

```sql
CREATE DATABASE pizzahut;
USE pizzahut;

CREATE TABLE orders(
order_id INT PRIMARY KEY,
order_date DATE NOT NULL,
order_time TIME NOT NULL
);

CREATE TABLE orders_details(
order_details_id INT PRIMARY KEY,
order_id INT NOT NULL,
pizza_id TEXT NOT NULL,
quantity INT NOT NULL
);
```

PIZZA SALES DATA ANALYSIS

QUERY: 1

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid

	total_orders
▶	21350

PIZZA SALES DATA ANALYSIS

QUERY: 2

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid

	total_sales
▶	817860.05

PIZZA SALES DATA ANALYSIS

QUERY: 3

IDENTIFY THE HIGHEST-PRICED PIZZA

```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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

PIZZA SALES DATA ANALYSIS

QUERY: 4

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    pizzas.size,
    SUM(orders_details.quantity) AS total_quantity_ordered
FROM
    pizzas
JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY
    pizzas.size
ORDER BY
    total_quantity_ordered DESC
LIMIT 1;
```

Result Grid | Filter Rows:

size	total_quantity_ordered
L	18956

PIZZA SALES DATA ANALYSIS

QUERY: 5

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

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS Total_Quantity
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.name
ORDER BY Total_Quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	Total_Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

PIZZA SALES DATA ANALYSIS

QUERY: 6

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

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS Total_Quantity
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
ORDER BY Total_Quantity DESC;
```

Result Grid | Filter Rows:

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

PIZZA SALES DATA ANALYSIS

QUERY: 7

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    HOUR(order_time) AS Hours, COUNT(order_id) AS Orders
FROM
    orders
GROUP BY Hours
ORDER BY Hours;
```

Result Grid

Hours	Orders
9	1
10	8
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

PIZZA SALES DATA ANALYSIS

QUERY: 8

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

```
SELECT
    category,
    COUNT(name),
    GROUP_CONCAT(name
        SEPARATOR ', ') AS pizza_names
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

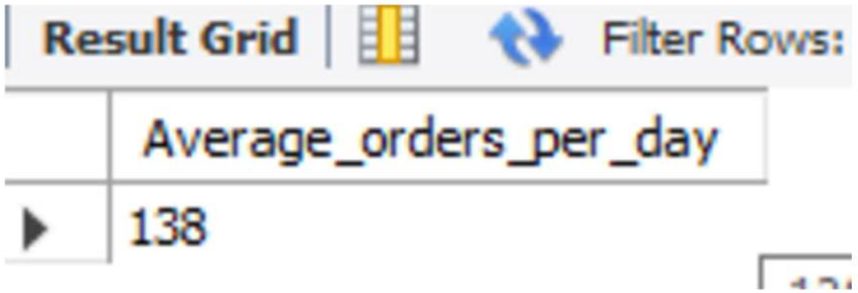
category	COUNT(name)	pizza_names
Chicken	6	The Barbecue Chicken Pizza, The California Chic...
Classic	8	The Big Meat Pizza, The Classic Deluxe Pizza, T...
Supreme	9	The Brie Carre Pizza, The Calabrese Pizza, The ...
Veggie	9	The Five Cheese Pizza, The Four Cheese Pizza, ...

PIZZA SALES DATA ANALYSIS

QUERY: 9

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

```
SELECT
    ROUND(AVG(Total_orders), 0) AS Average_orders_per_day
FROM
    (SELECT
        orders.order_date AS Order_date,
            SUM(orders_details.quantity) AS Total_orders
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY Order_date) AS Total_orders_per_day
```



PIZZA SALES DATA ANALYSIS

QUERY: 10

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

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS Revenue
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.name
ORDER BY Revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

PIZZA SALES DATA ANALYSIS

QUERY: 11

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

```
SELECT
    pizza_types.category,
      CONCAT(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 orders_details.pizza_id = pizzas.pizza_id) * 100,
            2),'%') AS percentage_distribution
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;
```

Result Grid | Filter Rows:

category	percentage_distribution
Classic	26.91%
Veggie	23.68%
Supreme	25.46%
Chicken	23.96%

PIZZA SALES DATA ANALYSIS

QUERY: 12

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT order_date, Day_wise_Revenue,
SUM(Day_wise_Revenue) OVER(ORDER BY order_date) as Cum_Revenue
FROM
(SELECT orders.order_date,
ROUND(SUM(orders_details.quantity * pizzas.price),2) AS Day_wise_Revenue
FROM orders
JOIN orders_details
ON orders_details.order_id = orders.order_id
JOIN pizzas
ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY orders.order_date) as sales
```

Result Grid | Filter Rows: | Export:

order_date	Day_wise_Revenue	Cum_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	21526.399999999998

PIZZA SALES DATA ANALYSIS

QUERY: 13

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

```
SELECT name, category, revenue
FROM
(SELECT name, category, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS Ranks
FROM
(SELECT pizza_types.name,
pizza_types.category, SUM(orders_details.quantity * pizzas.price) 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.name, pizza_types.category) AS subquery1) AS subquery2
WHERE Ranks <=3
```

Result Grid | Filter Rows: | Export:

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

PIZZA SALES DATA ANALYSIS

QUERY: 14

FIND THE TOP 3 DATES WITH THE HIGHEST NUMBER OF PIZZAS SOLD.

```
SELECT
    orders.order_date,
    SUM(orders_details.quantity) AS Total_pizzas_sold
FROM
    orders
        JOIN
    orders_details ON orders.order_id = orders_details.order_id
GROUP BY orders.order_date
ORDER BY Total_pizzas_sold DESC
LIMIT 3;
```

Result Grid | Filter Rows:

order_date	Total_pizzas_sold
2015-11-26	266
2015-11-27	264
2015-10-15	262

PIZZA SALES DATA ANALYSIS

QUERY: 15

IDENTIFY THE PIZZA TYPE WITH THE HIGHEST AVERAGE REVENUE PER ORDER (ONLY INCLUDE PIZZA TYPES THAT WERE ORDERED AT LEAST 5 TIMES).

```
SELECT pt.name AS Pizza_Type,
pt.category AS Category,
ROUND(SUM(orders_details.quantity * pizzas.price),2) AS Total_Revenue,
COUNT(DISTINCT orders_details.order_id) AS Total_Orders,
ROUND(SUM(orders_details.quantity * pizzas.price)/COUNT(DISTINCT orders_details.order_id),2)
AS Avg_Revenue_Per_Order
FROM pizza_types pt
JOIN pizzas
ON pizzas.pizza_type_id = pt.pizza_type_id
JOIN orders_details
ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pt.name, pt.category
HAVING COUNT(DISTINCT orders_details.order_id) >=500
ORDER BY Avg_Revenue_Per_Order DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Pizza_Type	Category	Total_Revenue	Total_Orders	Avg_Revenue_Per_Order
The Greek Pizza	Classic	28454.1	1361	20.91

PIZZA SALES DATA ANALYSIS

QUERY: 16

FIND THE TOP 3 HOURS OF THE DAY WHEN THE MOST PIZZAS WERE SOLD.

```
SELECT HOUR(order_time) AS Top_Selling_Hours,
SUM(orders_details.quantity) AS Quantity_Sold
FROM orders
JOIN orders_details
ON orders_details.order_id = orders.order_id
GROUP BY Top_Selling_Hours
ORDER BY Quantity_Sold DESC
LIMIT 3;
```

Result Grid | Filter Rows:

Top_Selling_Hours	Quantity_Sold
12	6776
13	6413
18	5417

PIZZA SALES DATA ANALYSIS

QUERY: 17

FOR EACH PIZZA CATEGORY, CALCULATE THE AVERAGE NUMBER OF PIZZAS SOLD PER ORDER.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity) / COUNT(DISTINCT orders_details.order_id),
            0) AS Avg_Sales_Per_Order
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
ORDER BY Avg_Sales_Per_Order DESC;
```

Result Grid | Filter Rows:

category	Avg_Sales_Per_Order
Chicken	1
Classic	1
Supreme	1
Veggie	1

PIZZA SALES DATA ANALYSIS

QUERY: 18

FIND THE MOST POPULAR PIZZA SIZE (S/M/L) FOR EACH PIZZA CATEGORY.

```
SELECT category, Revenue, size
FROM
(SELECT pizza_types.category,
ROUND(SUM(orders_details.quantity * pizzas.price),2)
AS Revenue, pizzas.size,
RANK() OVER(PARTITION BY pizza_types.category
ORDER BY SUM(orders_details.quantity * pizzas.price) DESC )
AS Ranks
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, pizzas.size) AS ranked_sizes
WHERE Ranks = 1;
```

Result Grid | Filter Rows:

category	Revenue	size
Chicken	102339	L
Classic	74518.5	L
Supreme	94258.5	L
Veggie	104202.7	L

I'M ALWAYS EAGER TO LEARN, GROW, AND IMPROVE. THIS SQL PROJECT HAS BEEN A REWARDING STEP IN MY JOURNEY TOWARD BECOMING A SKILLED DATA ANALYST.

I WELCOME ANY FEEDBACK, SUGGESTIONS, OR QUESTIONS THAT CAN HELP ME ENHANCE MY APPROACH OR PERSPECTIVE.

"LEARNING NEVER STOPS — AND EVERY QUERY BRINGS ME ONE STEP CLOSER."

THANK YOU!

I appreciate you taking the time to view my SQL project.
If you're interested in:

- Exploring the dummy dataset I used
- Reviewing this entire presentation deck
- Running the queries on your own

You can find everything on my GitHub:
🔗 github.com/MohdJunaid23/Pizza-Sales-Data-Analysis---MySQL