

PIZZA SALES ANALYSIS

USING SQL

PRESENTATION BY

MOSTAQUIL ARIF



CONTENTS OF THIS PRESENTATION

- **OVERVIEW**
- **DATA SOURCE AND METHODOLOGY**
- **UPLOADING CSVS INTO MYSQL**
- **DATA MODELLING (ER DIAGRAM)**
- **INSIGHT SLIDES (SQL QUERIES & THEIR SIGNIFICANCE)**
- **CONCLUSION**



OVERVIEW

- Diving into the world of pizza sales to analyze customer behavior.
- Studying sales data to identify key metrics and patterns.
- Understanding how menu items and promotions influence customer choices.
- Providing insights to help Pizza Hut enhance its sales strategies.
- Embarking on an exciting journey of decoding pizza sale behavior together!



DATA SOURCE & METHODOLOGY

THE DATA FOR THIS PROJECT WAS PROVIDED BY **PIZZA HUT**, SERVING AS THE PRIMARY SOURCE FOR OUR PIZZA SALES ANALYSIS. THE DATASETS WERE IMPORTED INTO A **MYSQL** DATABASE MANAGEMENT SYSTEM, ENSURING A RELIABLE AND EFFICIENT STORAGE SOLUTION FOR Pizza Hut's sales data. AS PART OF THE DATA PREPARATION PROCESS, CERTAIN COLUMNS WERE RESTRUCTURED OR MODIFIED TO FACILITATE EASIER ANALYSIS AND QUERYING, ENSURING THAT THE **DATA WAS CLEAN, RELEVANT, AND EASY TO WORK WITH.**

THE METHODOLOGY INVOLVED **SOLVING 15 PROBLEM STATEMENTS RELATED TO PIZZA SALES USING SQL QUERIES**. THESE PROBLEM STATEMENTS, PROVIDED BY **Pizza Hut**, GUIDED THE ANALYSIS AND HELPED UNCOVER **KEY INSIGHTS** INTO PIZZA SALES BEHAVIOR. LEVERAGING SQL AND THE **MYSQL DATABASE SYSTEM** ALLOWED FOR **ROBUST DATA MANIPULATION** AND **QUERYING CAPABILITIES**, ENABLING A THOROUGH EXPLORATION OF THE DATASET.

THIS APPROACH ENSURED A SYSTEMATIC AND RIGOROUS ANALYSIS OF PIZZA HUT'S SALES DATA, LEADING TO VALUABLE INSIGHTS AND CONCLUSIONS THAT CAN INFORM SALES STRATEGIES AND ENHANCE THE PIZZA ORDERING EXPERIENCE.

UPLOADING CSVs INTO MYSQL

To upload CSV files into MySQL, you can follow these steps:

- Preparation: Ensure that you have the CSV files ready and accessible.
- Database Creation: Create a database in MySQL where you want to import the CSV data.

```
CREATE DATABASE Pizza_Sales_Analysis;
```

- Data Import: Use MySQL's LOAD DATA INFILE statement to import the data from CSV files into your MySQL tables.



DATASETS DESCRIPTIONS

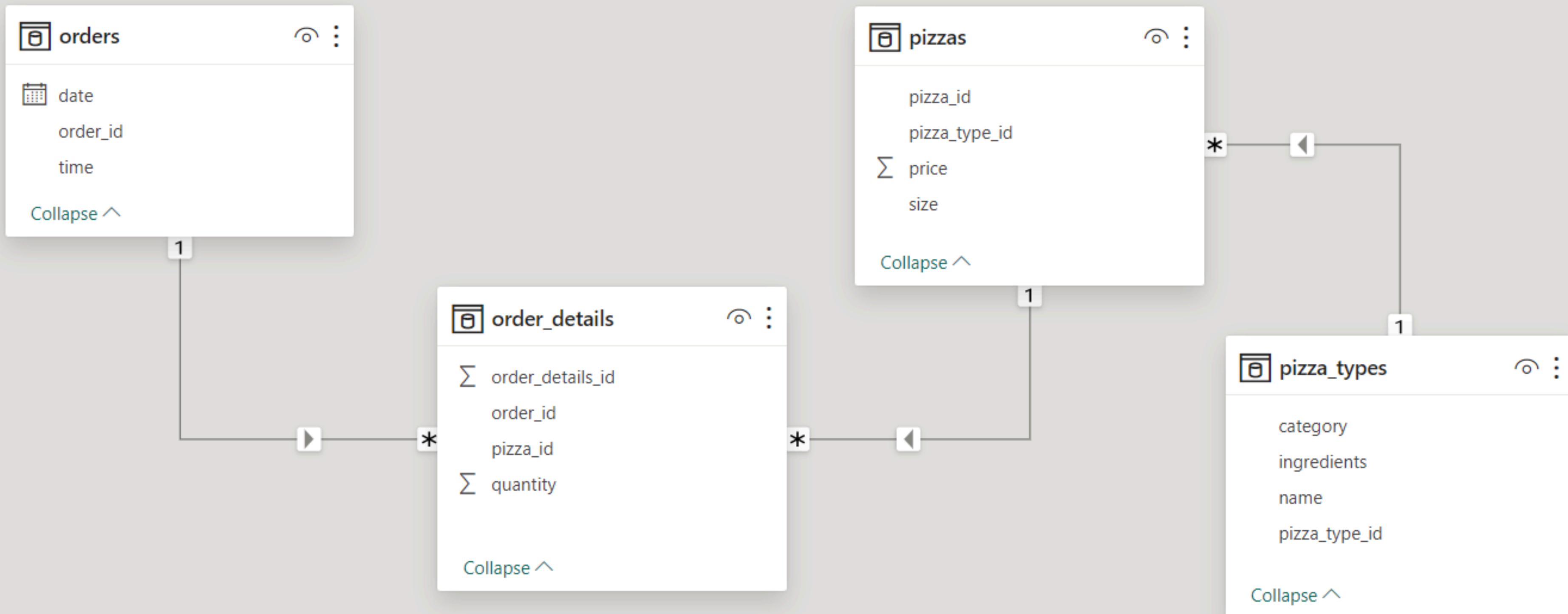
Orders					
Field	Type	Null	Key	Default	Description
order_id	int	NO	PRI	NULL	Unique identifier for each order.
order_date	date	NO		NULL	Date when the order was placed.
order_time	time	NO		NULL	Time at which the order was placed.

Orders_Details					
Field	Type	Null	Key	Default	Description
order_details_id	int	NO	PRI	NULL	Unique identifier for each order detail.
order_id	int	NO		NULL	Unique identifier for each order.
pizza_id	text	NO		NULL	Identifier for the type of pizza ordered.
quantity	int	YES		NULL	Quantity of the specific pizza ordered.

Pizza_Types					
Field	Type	Null	Key	Default	Description
pizza_type_id	text	YES		NULL	Identifier for the type of pizza.
name	text	YES		NULL	Name of the pizza type.
category	text	YES		NULL	Category of the pizza (e.g., vegetarian).
ingredients	text	YES		NULL	Ingredients used in the pizza.

Pizzas					
Field	Type	Null	Key	Default	Description
pizza_id	text	YES		NULL	Unique identifier for each pizza.
pizza_type_id	text	YES		NULL	Identifier for the type of pizza.
size	text	YES		NULL	Size of the pizza (e.g., small, medium).
price	double	YES		NULL	Price of the pizza.

DATA MODELLING (ER DIAGRAM)





Insights, Queries & their Significance

01

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

QUERY

```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    orders
```

OUTPUT

Total_Orders
21350

Significance: Total orders indicate customer demand, guiding Pizza Hut's decisions on staffing, inventory, and marketing.



02

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

OUTPUT

QUERY

SELECT

```
ROUND(SUM(pizzas.price * orders_details.quantity),  
    2) AS Total_Sales
```

FROM

pizzas

JOIN

```
orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

Significance: Total pizza sales revenue measures financial performance, guiding strategic decisions for Pizza Hut.

Total_Sales
817860.05



03

IDENTIFY THE HIGHEST-PRICED PIZZA.

QUERY

SELECT

pizza_types.name, SUM(pizzas.price) AS Total

FROM

pizzas

JOIN

pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id

GROUP BY pizza_types.name

ORDER BY Total DESC

LIMIT 1;

Significance: Identifying the highest-priced pizza informs Pizza Hut's pricing strategies and product offerings.

OUTPUT

Name	Total
The Greek Pizza	109.95



04

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

OUTPUT

QUERY

SELECT

```
pizzas.size, SUM(orders_details.quantity) AS Total
```

FROM

```
pizzas
```

JOIN

```
orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

GROUP BY pizzas.size

ORDER BY Total DESC

Significance: Knowing the most common pizza size streamlines inventory management and ensures customer satisfaction.

Size	Total
L	18956
M	15635
S	14403
XL	552
XXL	28



05

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

OUTPUT

QUERY

SELECT

```
 pizza_types.name,  
 SUM(orders_details.quantity) AS Total_Quantity  
FROM  
 pizzas  
 JOIN  
 pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
 JOIN  
 orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY Total_Quantity DESC  
LIMIT 5
```

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

Significance: Listing the top 5 most ordered pizza types with quantities informs Pizza Hut's inventory and marketing strategies



06

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

QUERY

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

OUTPUT

Category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



Significance: Joining tables to find total quantity per pizza category guides Pizza Hut's menu planning and marketing strategies.

07

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

QUERY

SELECT

HOUR(order_time) AS Hours, COUNT(order_id) AS Orders

FROM

orders

GROUP BY Hours

ORDER BY orders DESC

OUTPUT

Hours	Orders
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



Significance: Understanding order distribution by hour enables Pizza Hut to optimize staffing and resources for efficient operations.

08

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

QUERY

```
SELECT  
    pizza_types.category,  
    COUNT(orders_details.order_id) AS Orders  
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 Orders DESC
```

OUTPUT

Category	Orders
Classic	14579
Supreme	11777
Veggie	11449
Chicken	10815



Significance: Joining tables for category-wise pizza distribution guides Pizza Hut's menu planning and marketing strategies.

09

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

QUERY

```
SELECT  
    ROUND(AVG(Quantity), 0) AS Avg_Pizza_Ordered_Per_Day  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) AS Quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS data
```

OUTPUT

Avg_Pizza_Ordered_Per_Day
138

Significance: Grouping orders by date and calculating the average pizzas per day guides Pizza Hut's inventory and staffing decisions.



10

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

QUERY

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

OUTPUT

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

Significance: Identifying the top 3 pizza types by revenue guides Pizza Hut's inventory and marketing decisions for maximum profitability.



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

QUERY

```
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * orders_details.quantity),
        2) AS Revenue,
    ROUND(((ROUND(SUM(pizzas.price * orders_details.quantity),
        2)) / (SELECT
        ROUND(SUM(pizzas.price * orders_details.quantity),
        2)) AS Total_Sales
    FROM
        pizzas
        JOIN
            orders_details ON pizzas.pizza_id = orders_details.pizza_id) * 100),
    2) AS Percentage_Revenue
FROM
    pizzas
    JOIN
        pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
        orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Percentage_Revenue DESC
```

OUTPUT

Category	Revenue	Percentage_Revenue
Classic	220053.1	26.91
Supreme	208197	25.46
Chicken	195919.5	23.96
Veggie	193690.45	23.68



Significance: Calculating pizza types percentage contribution to total revenue informs Pizza Hut's strategic decisions on product offerings and marketing.

I2

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY

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

OUTPUT

order_date	Revenue	Cum_Revenue
01-01-2015	2713.85	2713.85
02-01-2015	2731.9	5445.75
03-01-2015	2662.4	8108.15
04-01-2015	1755.45	9863.6



Significance: Analyzing cumulative revenue over time tracks Pizza Hut's performance and informs resource allocation.

13

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

QUERY

```
SELECT 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
      pizzas
    JOIN
      orders_details ON pizzas.pizza_id = orders_details.pizza_id
    JOIN
      pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    GROUP BY
      pizza_types.name, pizza_types.category
  ) AS a
) AS b
WHERE RN <= 3;
```

OUTPUT

Name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5



Significance: Identifying the top 3 revenue-generating pizza types in each category guides Pizza Hut's menu optimization and marketing strategies effectively.

CONCLUSION

Our project utilized Pizza Sales Analysis data, harnessing MySQL for efficient database management. Through meticulous data preparation and SQL analysis, we addressed key inquiries, revealing essential insights into pizza sales behavior.

These insights, ranging from popular pizza types to revenue trends, provide actionable implications for menu optimization and marketing strategies. Our project highlights the versatility of SQL in handling complex datasets, emphasizing the importance of systematic analysis in shaping Pizza Hut's strategies.

The outcomes of this project have the potential to drive decision-making within Pizza Hut, showcasing the value of rigorous data analysis within MySQL environments for the food industry.

**CONTACT
FOR MORE
INFORMATION**



Github



My Portfolio



mostaqul.arif@gmail.com



Linkedin



THANK YOU

