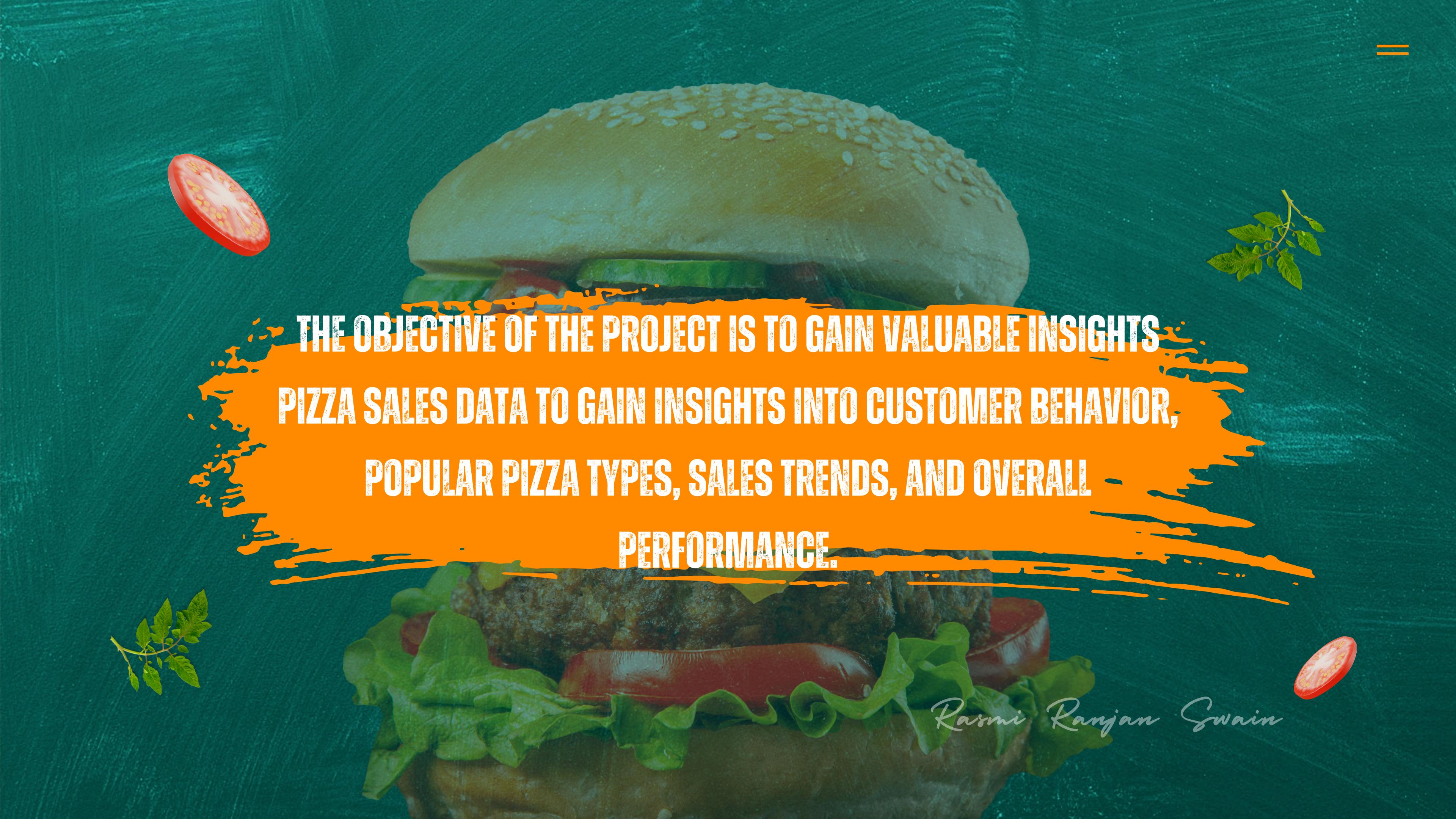




RASMI RANJAN SWAIN

UNVEILING THE SECRETS BEHIND PIZZA SALES!

RASMI RANJAN SWAIN



**THE OBJECTIVE OF THE PROJECT IS TO GAIN VALUABLE INSIGHTS
PIZZA SALES DATA TO GAIN INSIGHTS INTO CUSTOMER BEHAVIOR,
POPULAR PIZZA TYPES, SALES TRENDS, AND OVERALL
PERFORMANCE.**

Rasmi Ranjan Swain

CONTENT

- OVERVIEW
- DATA SOURCE & METHODOLOGY
- UPLOADING CSVS INTO MYSQL
- DATA MODELLING (ER DIAGRAM)
- BUSINESS QUESTIONS
- CONCLUSION



Rasmi Ranjan Swain

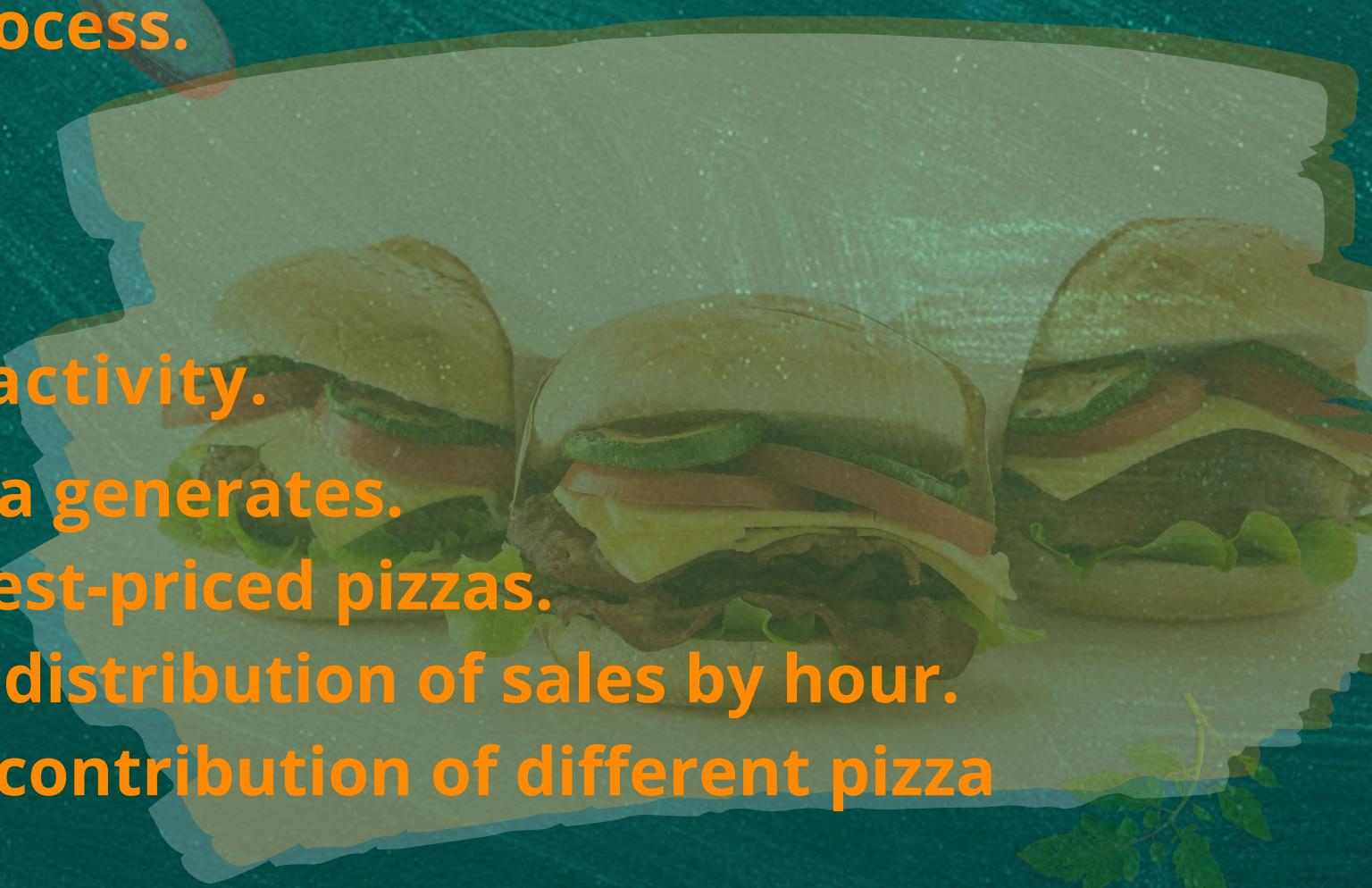
OVERVIEW

01

This project aims to conduct a comprehensive analysis of pizza sales data in order to gain insight into customer preferences, revenue generation, and order patterns. As a result of analyzing various metrics, we aim to find trends and opportunities for improving the pizza sales process.

WHAT WE'RE EXPLORING:

-  **Total Orders Placed:** Understanding the volume of activity.
-  **Total Revenue:** Analyzing how much revenue each pizza generates.
-  **Top Pizza Picks:** Identifying the most ordered and highest-priced pizzas.
-  **Sales Trends:** Discovering peak ordering times and the distribution of sales by hour.
-  **Pizza Categories:** Examining the quantity and revenue contribution of different pizza types.
-  **Revenue Over Time:** Tracking cumulative revenue trends.



Rashi Ranjan Swain

DATA SOURCE & METHODOLOGY

01

For our pizza sales analysis, we used data provided by @mentorness.

In order to ensure a reliable and efficient storage solution for Pizza sales data, the datasets were imported into the MySQL database management system. To ensure that the data was clean, relevant, and easy to work with, certain columns were restructured or modified to facilitate easier analysis and querying.

Rashi Ranjan Swain

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.

Create a database in MySQL where you want to import the CSV data.



Database Creation:

```
CREATE DATABASE  
Pizza_Sales_Analysis
```

Use MySQL's LOAD DATA INFILE statement to import the data from CSV files into your MySQL tables

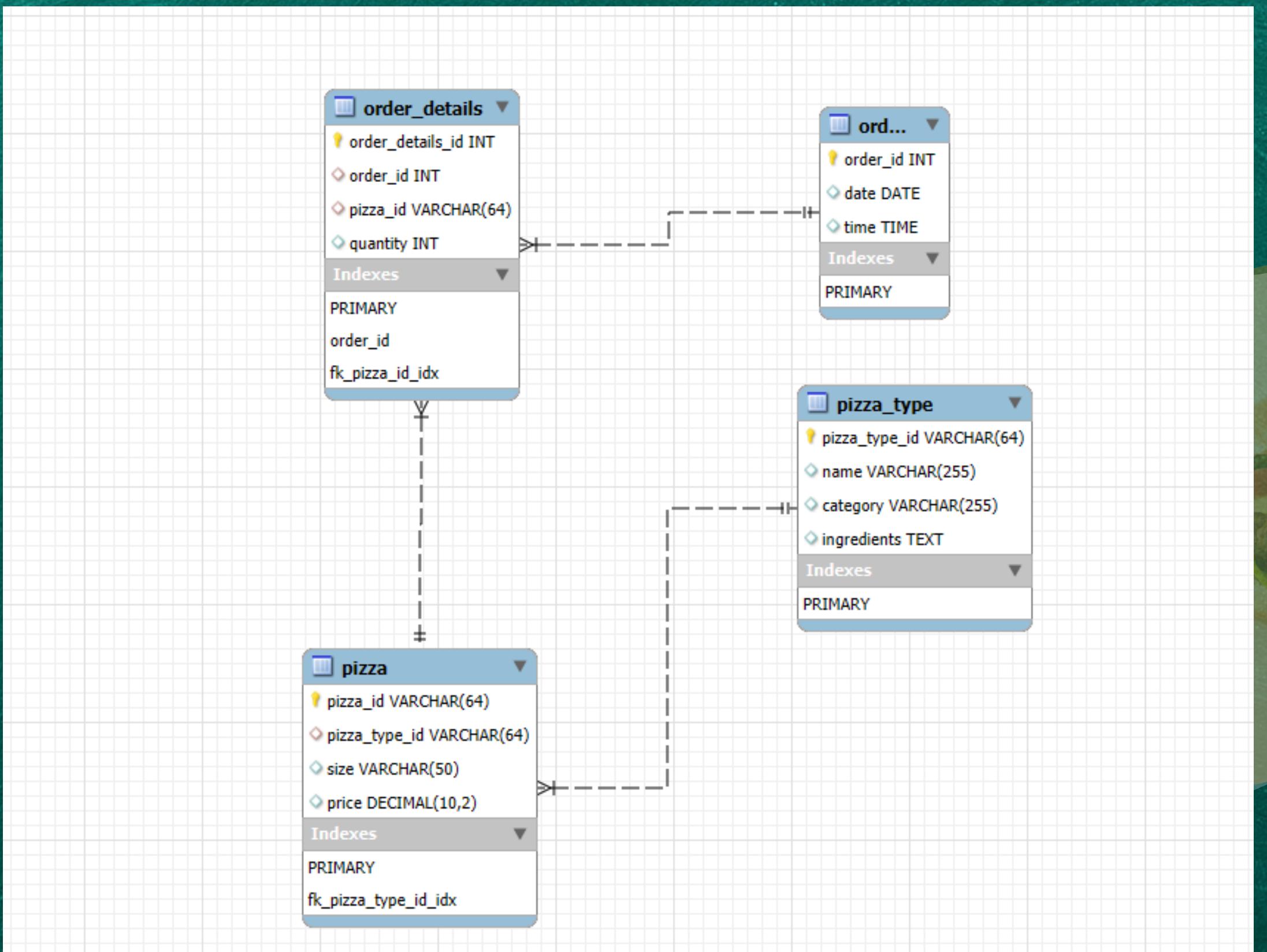


Data Import



Rashi Ranjan Swain

DATA MODELLING (ER DIAGRAM)



Rashi Ranjan Swain

BUSINESS QUESTIONS

SALES ANALYSIS



Rashi Ranjan Swain

BUSINESS QUESTIONS

SALES ANALYSIS

Q1: The total number of order place

QUERY

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

RESULT

Result Grid	
	Total_Orders
▶	21350



BUSINESS QUESTIONS

SALES ANALYSIS

Q2: The total revenue generated from pizza sales

QUERY

```
SELECT  
SUM(order_details.quantity * pizza.price) AS 'Total Revenue'  
FROM  
order_details  
JOIN  
pizza ON order_details.pizza_id = pizza.pizza_id;
```

RESULT

Result Grid	
	Total Revenue
▶	42663.55



BUSINESS QUESTIONS

SALES ANALYSIS

Q3: The highest priced pizza.

QUERY

```
SELECT pizza_type.name, pizza.pric  
FROM  
pizza_type  
JOIN  
pizza  
WHERE  
pizza_type.pizza_type_id = pizza.pizza_type_id  
ORDER BY price DESC limit 1;
```

RESULT

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



BUSINESS QUESTIONS

PRODUCT PERFORMANCE



Rashi Ranjan Swain

Q4: The most common pizza size ordered.

QUERY

```
SELECT
    pizza.size, SUM(order_details.quantity) AS total_quantity
FROM
    order_details
    JOIN
    pizza ON order_details.pizza_id = pizza.pizza_id
GROUP BY pizza.size limit 1;
```

RESULT

Result Grid | Filter R

	size	total_quantity
▼	L	1018



BUSINESS QUESTIONS

PRODUCT PERFORMANCE

Q5: The top 5 most ordered pizza types along their quantities.

QUERY

```
SELECT
    pt.pizza_type_id AS pizza_type,
    pt.name,
    SUM(quantity) AS total_order_quantities
FROM
    order_details ord
        JOIN
    pizza pz ON ord.pizza_id = pz.pizza_id
        JOIN
    pizza_type pt ON pz.pizza_type_id = pt.pizza_type_id
GROUP BY pt.pizza_type_id
ORDER BY total_order_quantities DESC
LIMIT 5;
```

RESULT

	pizza_type	name	total_order_quantities
▶	pepperoni	The Pepperoni Pizza	157
	bbq_dkn	The Barbecue Chicken Pizza	129
	thai_dkn	The Thai Chicken Pizza	123
	cali_dkn	The California Chicken Pizza	122
	sicilian	The Sicilian Pizza	117



Q6: The quantity of each pizza categories ordered.

QUERY

```
select pt.category,sum(quantity)
as
total_order_quantity
from order_details ord
join
pizza pz on ord.pizza_id=pz.pizza_id
join
pizza_type pt on pt.pizza_type_id=pz.pizza_type_id
group by pt.category;
```

RESULT

Result Grid | Filter Rows:

	category	total_order_quantity
▶	Chicken	570
	Classic	762
	Supreme	636
	Veggie	611



BUSINESS QUESTIONS

CUSTOMER BEHAVIOR



Rashi Ranjan Swain

Q7: The distribution of orders by hours of the day.

QUERY

```
SELECT  
    HOUR(time) AS distribution_hour, COUNT(*) AS  
Total_order  
FROM  
    orders  
GROUP BY distribution_hour  
ORDER BY distribution_hour;
```

RESULT

	distribution_hour	Total_order
▶	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

Q8: The category-wise distribution of pizzas.

QUERY

```
SELECT pt.category, COUNT(p.pizza_id) AS total_pizzas  
FROM pizza p  
JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY pt.category  
ORDER BY total_pizzas DESC;
```

RESULT

	category	total_pizzas
▶	Veggie	27
	Classic	26
	Supreme	25
	Chicken	18



BUSINESS QUESTIONS

PRICING STRATEGY



Rashi Ranjan Swain

BUSINESS QUESTIONS

PRICING STRATEGY

Q9: The average number of pizzas ordered per day.

QUERY

```
SELECT
    AVG(total_quantity)
FROM
    (SELECT
        DATE(ord.date), SUM(quantity) AS total_quantity
    FROM
        order_details od
    JOIN orders ord ON od.order_id = od.order_id
    GROUP BY DATE(ord.date)) t;
```

RESULT

Result Grid	
	AVG(total_quantity)
▶	153803.4916



Q10: Top 3 most ordered pizza type base on revenue.

QUERY

```
SELECT
    pt.name AS pizza_type,
    SUM(od.quantity * p.price) AS total_revenue
FROM
    order_details od
        JOIN
    pizza p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3;
```

RESULT

	pizza_type	total_revenue
▶	The Barbecue Chicken Pizza	2300.75
	The Thai Chicken Pizza	2280.25
	The California Chicken Pizza	2099.50



Q11: The percentage contribution of each pizza type to revenue.

QUERY

```

SELECT pt.name AS pizza_type,
       SUM(od.quantity * p.price) AS total_revenue,
       (SUM(od.quantity * p.price) / (SELECT SUM(od2.quantity * p2.price)
                                         FROM order_details od2
                                         JOIN pizza p2 ON od2.pizza_id = p2.pizza_id)) * 100 AS
       percentage_contribution
  FROM order_details od
  JOIN pizza p ON od.pizza_id = p.pizza_id
  JOIN pizza_type pt ON p.pizza_type_id = pt.pizza_type_id
 GROUP BY pt.name
 ORDER BY percentage_contribution DESC;
    
```

RESULT

	pizza_type	total_revenue	percentage_contribution
▶	The Barbecue Chicken Pizza	2300.75	5.392777
	The Thai Chicken Pizza	2280.25	5.344726
	The California Chicken Pizza	2099.50	4.921063
	The Pepperoni Pizza	2003.75	4.696632
	The Italian Supreme Pizza	1936.00	4.537831
	The Spicy Italian Pizza	1870.25	4.383719
	The Sicilian Pizza	1813.25	4.250115
	The Classic Deluxe Pizza	1812.50	4.248357
	The Southwest Chicken Pizza	1655.50	3.880362
	The Greek Pizza	1552.50	3.638938
	The Four Cheese Pizza	1534.25	3.596161
	The Five Cheese Pizza	1517.00	3.555728
	The Hawaiian Pizza	1499.50	3.514710
	The Pepper Salami Pizza	1401.25	3.284420
	The Mexicana Pizza	1399.75	3.280904
	The Vegetables + Vegetable...	1326.00	3.108040
	The Napolitana Pizza	1317.00	3.086944
	The Spinach and Feta Pizza	1270.00	2.976780
	The Prosciutto and Arugula ...	1248.00	2.925214
	The Italian Capocollo Pizza	1184.50	2.776375
	The Big Meat Pizza	1092.00	2.559562
	The Chicken Alfredo Pizza	934.00	2.189222

Q12: The cumulative revenue generated over time.

QUERY

```
SELECT o.date,
       SUM(od.quantity * p.price) AS daily_revenue,
       SUM(SUM(od.quantity * p.price)) OVER (ORDER BY o.date) AS
       cumulative_revenue
  FROM orders o
 JOIN order_details od ON o.order_id = od.order_id
 JOIN pizza p ON od.pizza_id = p.pizza_id
 GROUP BY o.date
 ORDER BY o.date;
```

RESULT

	date	daily_revenue	cumulative_revenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.90	5445.75
	2015-01-03	2662.40	8108.15
	2015-01-04	1755.45	9863.60
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.50
	2015-01-07	2202.20	16560.70
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.40
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.30	25862.65
	2015-01-12	1919.05	27781.70
	2015-01-13	2049.60	29831.30
	2015-01-14	2527.40	32358.70
	2015-01-15	1984.80	34343.50
	2015-01-16	2594.15	36937.65
	2015-01-17	2064.10	39001.75
	2015-01-18	1976.85	40978.60
	2015-01-19	1684.95	42663.55

BUSINESS QUESTIONS

PRICING STRATEGY

Q13: The top 3 most ordered pizza type based on revenue for each pizza category.

QUERY

```
SELECT category.name, Revenue
FROM (
    SELECT
        category.name,
        SUM(pizza.price * order_details.quantity) AS Revenue,
        RANK() OVER (PARTITION BY category.name ORDER BY SUM(pizza.price * order_details.quantity) DESC) AS RA
    FROM
        pizza
    JOIN
        order_details ON pizza.pizza_id = order_details.pizza_id
    JOIN
        pizza_type ON pizza_type.pizza_type_id = pizza.pizza_type_id
    GROUP BY category.name
) AS b
WHERE RA <= 3;
```

RESULT

category	name	Revenue
Chicken	The Barbecue Chicken Pizza	2300.75
Chicken	California Chicken Pizza	2099.50
Chicken	The Chicken Alfredo Pizza	934.00
Chicken	The Chicken Pesto Pizza	897.50
Chicken	The Southwest Chicken Pizza	1655.50
Chicken	The Thai Chicken Pizza	2280.25
Classic	The Big Meat Pizza	1092.00
Classic	The Classic Deluxe Pizza	1812.50
Classic	The Greek Pizza	1552.50
Classic	The Hawaiian Pizza	1499.50
Classic	The Italian Capocollo Pizza	1184.50
Classic	The Napolitana Pizza	1317.00
Classic	The Pepperoni Pizza	2003.75
Classic	The Pepperoni, Mushroom, ...	799.00
Supreme	The Brie Carre Pizza	402.05
Supreme	The Calabrese Pizza	653.50

Rasmi Ranjan Swain

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.

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.

Rashi Ranjan Swain

THANK YOU



<https://github.com/sairasmi/>



www.linkedin.com/in/rasmiranjan1



swainrasmiranjan7@gmail.com

RASMI RANJAN SWAIN