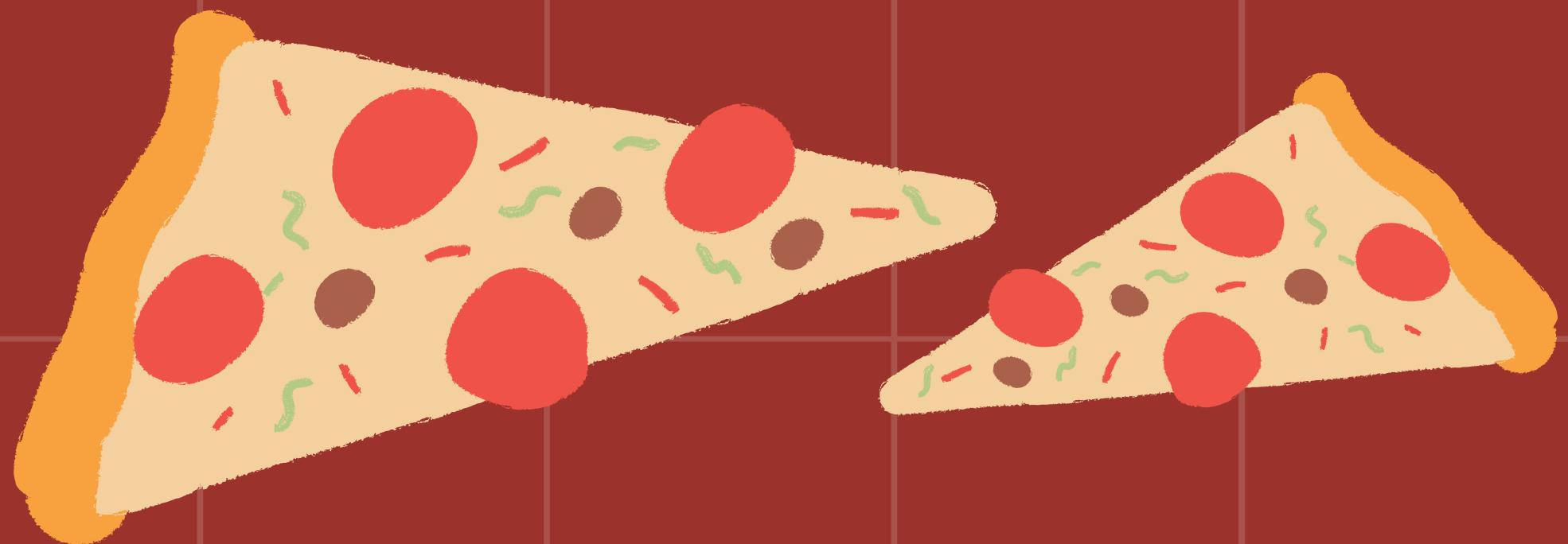


PIZZA HUT SALES ANALYSIS



INTRODUCTION

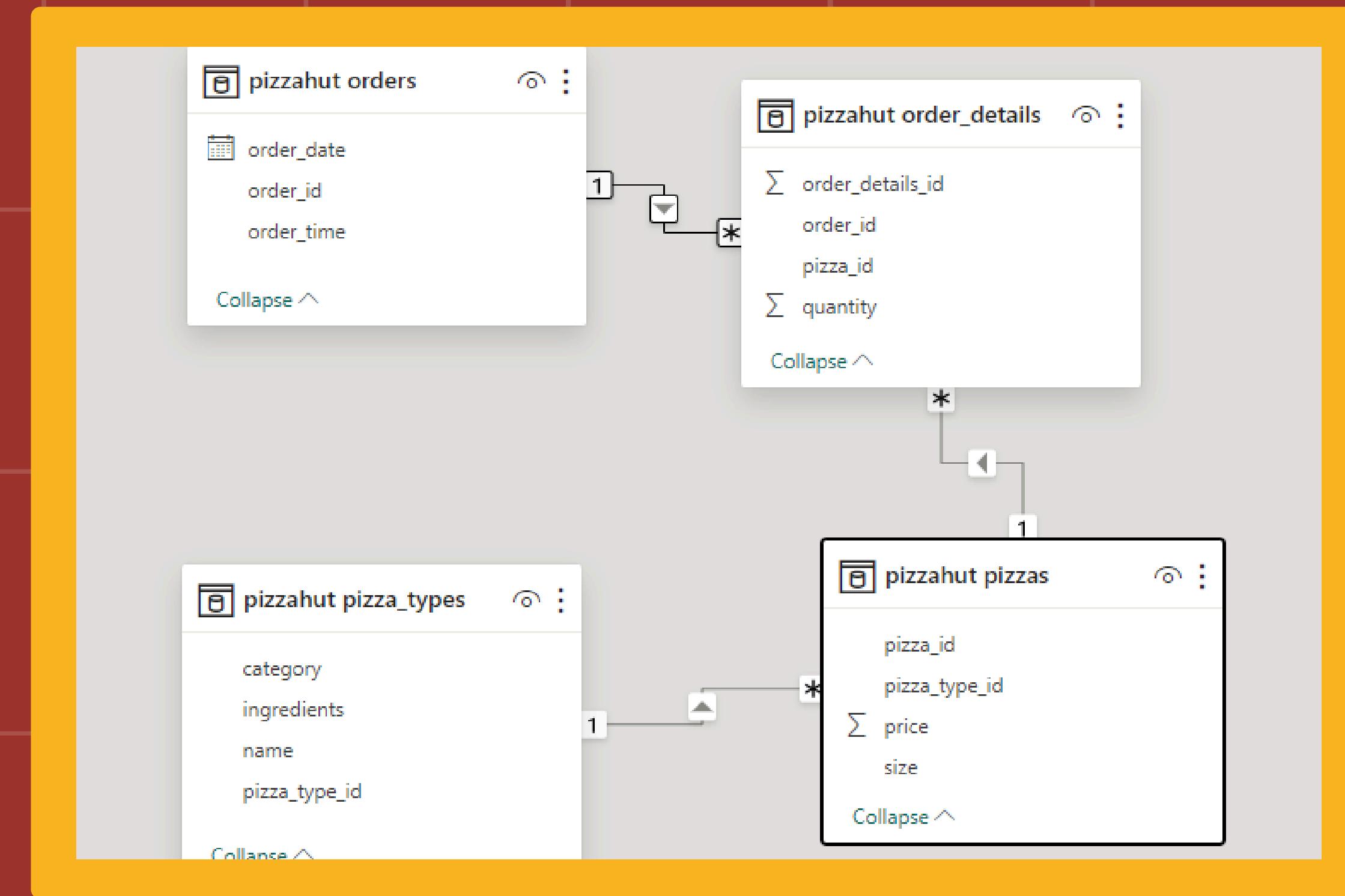
Hi, my name is **Sumit**, and I am a professional **Data Analyst**. In this project, I conducted a comprehensive analysis of Pizza Hut sales over a specified period. Utilizing **SQL queries**, I addressed key business questions and provided actionable insights to support decision-making.

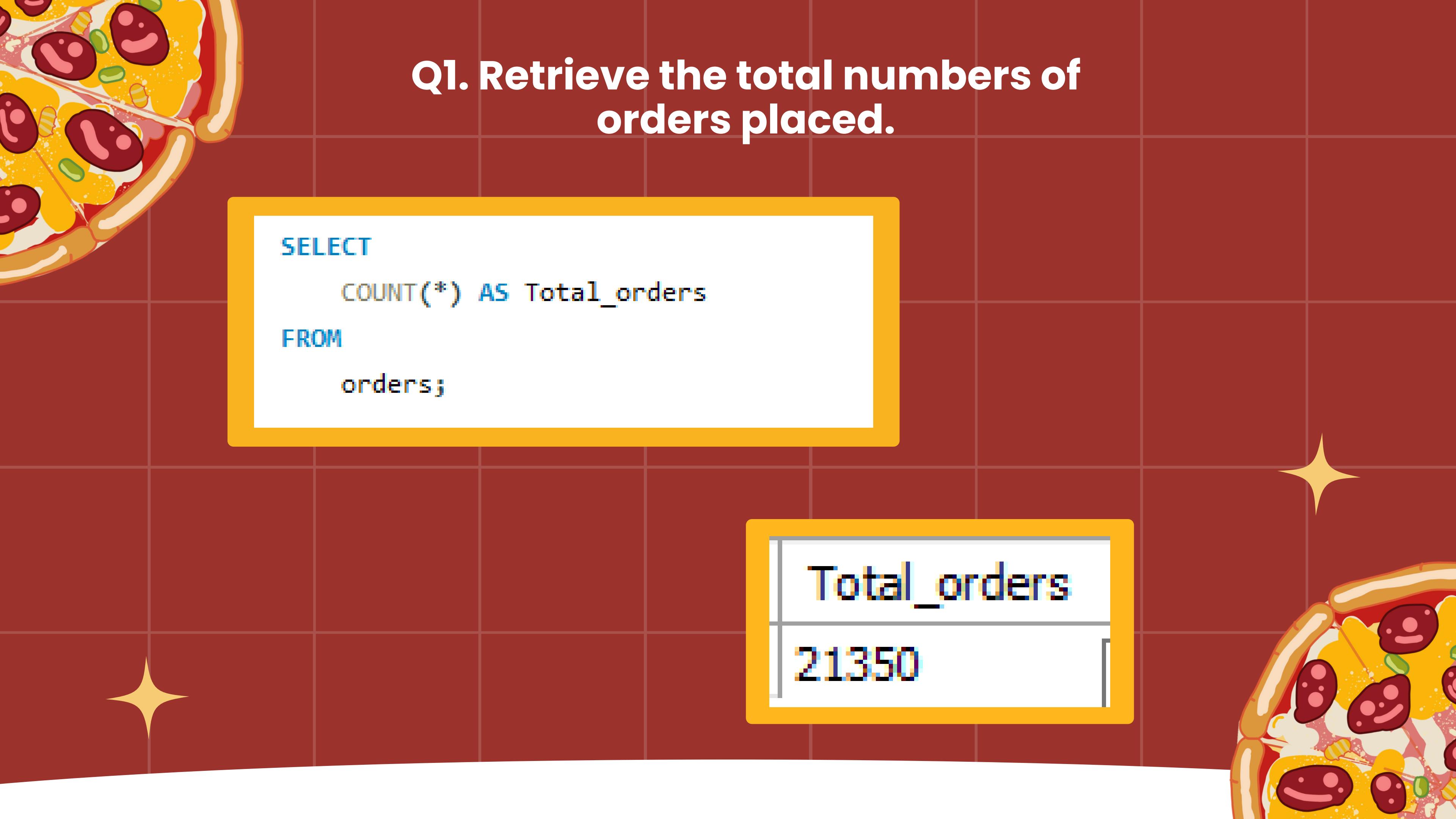
PROBLEM STATEMENT



Pizza Hut aims to optimize sales by analyzing data over a specific period. The goal is to use SQL queries to uncover insights, answer key business questions, and provide actionable recommendations for improved decision-making

DATA MODEL





Q1. Retrieve the total numbers of orders placed.

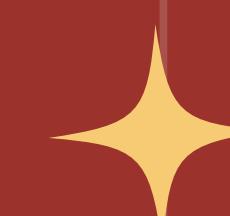
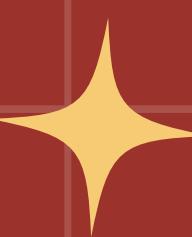
```
SELECT  
    COUNT(*) AS Total_orders  
FROM  
    orders;
```

Total_orders
21350



Q2. Calculate the total revenue generated from pizza sales.

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



Total_Revenue
817860.05

Q3. Identify the highest-priced pizza.

```
SELECT  
    name, price AS Highest_priced_pizza  
FROM  
    pizza_types pt  
    INNER JOIN  
    pizzas p ON pt.pizza_type_id = p.pizza_type_id  
GROUP BY name  
ORDER BY Highest_priced_pizza DESC  
LIMIT 1;
```

name	Highest_priced_pizza
The Brie Carre Pizza	23.65

Q4. Identify the most common pizza size ordered.

```
SELECT  
    size, COUNT(order_details_id) AS order_count  
FROM  
    order_details od  
        INNER JOIN  
    pizzas p ON od.pizza_id = p.pizza_id  
GROUP BY size  
ORDER BY order_count DESC;
```

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Q5. List the top 5 most ordered pizza types along with their quantities.

```
SELECT  
    name AS Pizza_type, SUM(quantity) AS Total_quantity  
FROM  
    pizzas p  
        INNER JOIN  
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id  
        INNER JOIN  
    order_details od ON od.pizza_id = p.pizza_id  
GROUP BY Pizza_type  
ORDER BY Total_quantity DESC  
LIMIT 5;
```

Pizza_type	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



Q6. Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT
    category, SUM(quantity) AS Total_quantity
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY Total_quantity DESC;
```

category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS Hour, COUNT(order_id) AS Order_count  
FROM  
    orders  
GROUP BY Hour;
```

Hour	Order_count
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
10	8
9	1



Q8. Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT category, COUNT(*) AS Order_count  
FROM pizza_types  
GROUP BY category;
```

category	Order_count
Chicken	6
Classic	8
Supreme	9
Veggie	9



Q9. Group the orders by date and calculate the average number of pizzas ordered per day

```
SELECT  
    ROUND(AVG(quantity), 0) AS Average_pizza_ordered_per_day  
FROM  
    (SELECT  
        order_date, SUM(quantity) AS quantity  
    FROM  
        orders o  
    INNER JOIN order_details od ON o.order_id = od.order_id  
    GROUP BY order_date) AS temptable;
```

Average_pizza_ordered_per_day

138





Q10. Determine the top 3 most ordered pizza types based on revenue.

```
SELECT name AS Pizza_type, SUM(quantity * price) AS Revenue  
FROM pizza_types pt  
INNER JOIN pizzas p  
ON pt.pizza_type_id = p.pizza_type_id  
INNER JOIN order_details od  
ON od.pizza_id = p.pizza_id  
GROUP BY Pizza_type  
ORDER BY Revenue DESC  
LIMIT 3;
```



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

Q11. Calculate the percentage contribution of each category to total revenue.

```
SELECT
    category,
    ROUND(SUM(quantity * price) / (SELECT
                                         ROUND(SUM(quantity * price), 2) AS Total_Revenue
                                       FROM
                                         pizzas p
                                         INNER JOIN
                                         order_details od ON p.pizza_id = od.pizza_id) * 100,
    2) AS Revenue_contribution_in_percentage
FROM
    pizza_types pt
    INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
    INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY Revenue_contribution_in_percentage DESC;
```

category	Revenue_contribution_in_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. Analyze the cumulative revenue generated over time.

```
SELECT order_date,  
       SUM(Revenue) OVER (ORDER BY order_date) AS Cumulative_revenue FROM  
       (SELECT order_date, SUM(quantity*price) AS Revenue FROM order_details od  
        INNER JOIN pizzas p ON od.pizza_id = p.pizza_id  
        INNER JOIN orders o  
        ON o.order_id = od.order_id  
        GROUP BY order_date  
        ORDER BY Revenue DESC) AS temptable;
```

order_date	Cumulative_revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55
2015-01-06 00:00:00	14358.5
2015-01-07 00:00:00	16560.7
2015-01-08 00:00:00	19399.05
2015-01-09 00:00:00	21526.4
2015-01-10 00:00:00	23990.350000000002
2015-01-11 00:00:00	25862.65
2015-01-12 00:00:00	27781.7



Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
SELECT name, Revenue, Rn FROM
(SELECT category, name, Revenue,
RANK() OVER(PARTITION BY category ORDER BY Revenue DESC) AS Rn FROM
(SELECT category, name, SUM(quantity*price) AS Revenue
FROM pizza_types pt
INNER JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id
INNER JOIN order_details od
ON od.pizza_id = p.pizza_id
GROUP BY category, name
ORDER BY Revenue DESC) AS A) AS B
WHERE Rn <=3;
```

name	Revenue	Rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3





THANK YOU

“Enjoy every bite”

Feel free to ask questions

Like-Share-Comment