Hot

PIZZA HUT

WELCOME MESSAGE

This project aimed to leverage SQL to analyze sales data for the PizzaHut Brand and uncover valuable insights to improve business understanding, optimize marketing strategies, and predict future sales trends.

Customer Preferences:

- What are the most popular pizzas (toppings, sizes, crusts) overall and by revenue?
- Are there any hourly trends in pizza sales?

Sales Performance:

- What are the top-performing days, hours, and pizza types for sales?
- Which marketing campaigns have been most effective in driving sales?

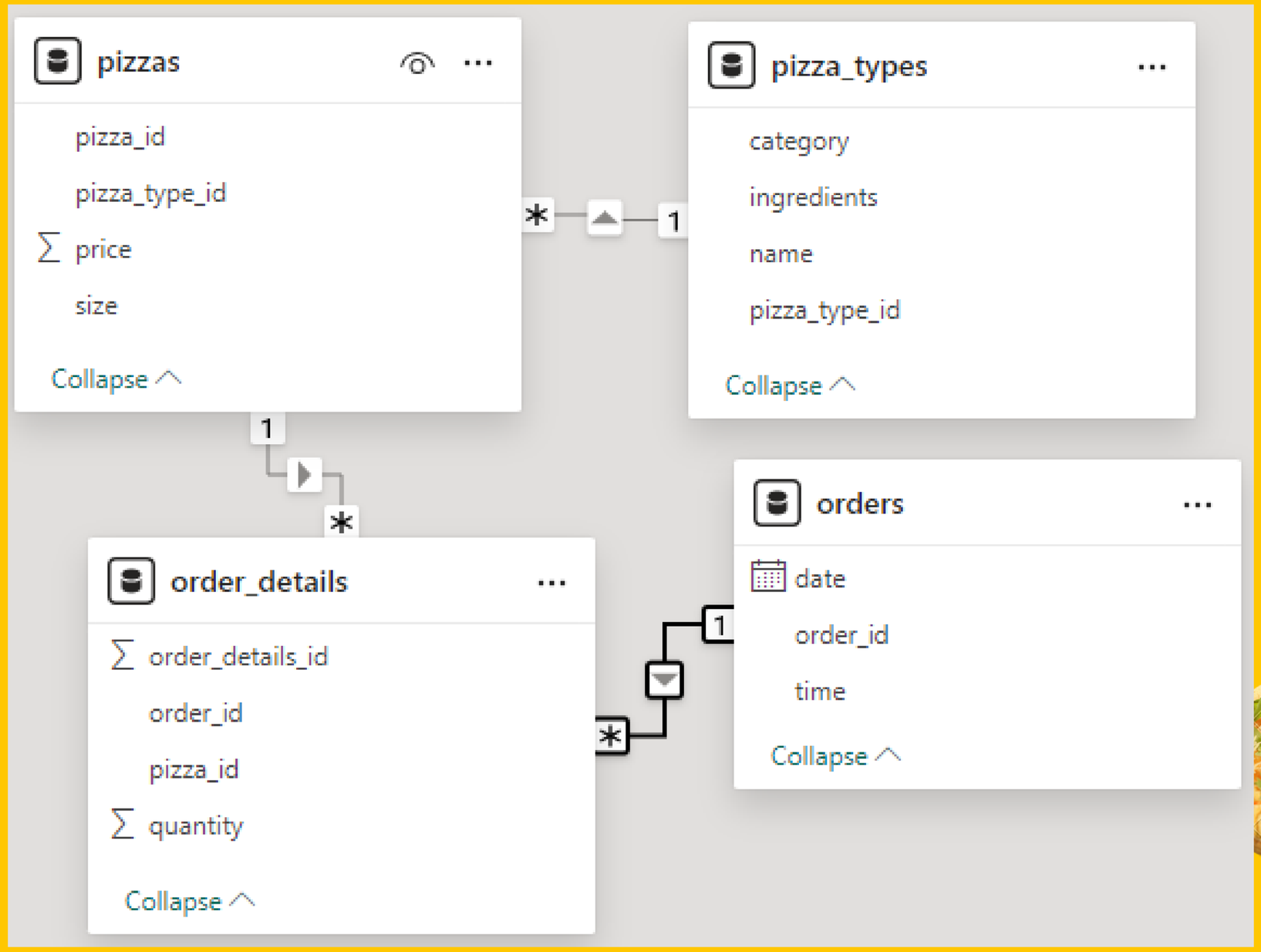
PROBLEM STATEMENTS

1. Retrieve total number of orders placed.
2. Calculate total revenue generated from pizza sales.
3. List the top 5 most ordered pizza types along with their quntities.
4. Indentify highest priced Pizza
5. Identify the most common pizza size ordered

1. Find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Find the category wise distribution of pizzas
4. Determine the top 3 most ordered pizza types based on revenue
5. Group the orders by date and calculate the average number of pizzas ordered per day.
6. Determine the top 3 most ordered pizza types based on revenue

1. Rank pizzas based on revenue for each pizza category.
2. Calculate the percentage contribution of each piza type to total revenue.
3. Analyze the cumulative revenue generated over time.
4. Rank pizzas based on revenue for each pizza category.

DATABASE SCHEMA



BASIC ANALYSIS

Retrieve total number of orders placed

```sql
SELECT
    COUNT(*) AS Total_Orders
FROM
    orders;
```

Result Grid

	Total_Orders
▶	21350

Calculate total revenue generated from pizza sales

```sql
SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS Revenue
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id;
```

Result Grid

	Revenue
▶	817860.05

Indentify highest priced Pizza

```sql
SELECT
    t.name AS Highest_Priced_Piza, p.price AS Price
FROM
    pizzas p
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
WHERE
    price = (SELECT
            MAX(price)
        FROM
            pizzas);
```

Result Grid | Filter Rows:

	Highest_Priced_Piza	Price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered

```sql
SELECT
    p.size Most_Common_Size, COUNT(1) AS Total_Orders
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY p.size
ORDER BY COUNT(p.size) DESC
LIMIT 1;
```

Result Grid | Filter Rows:

Most_Common_Size	Total_Orders
L	18526

List the top 5 most ordered pizza types along with their quntities

```sql
SELECT
    name AS Pizza_Name, SUM(o.quantity) AS Quantity_Ordered
FROM
    pizza_types t
        JOIN
    pizzas p ON p.pizza_type_id = t.pizza_type_id
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY t.name
ORDER BY Quantity_Ordered DESC
LIMIT 5;
```

Result Grid | Filter Rows:

Pizza_Name	Quantity_Ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

INTERMEDIATE ANALYSIS

Find the total quantity of each pizza category ordered

```sql
SELECT
    t.category AS Pizza_Category,
    SUM(o.quantity) AS Total_Orders
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY t.category
ORDER BY Total_Orders DESC;
```

Pizza_Category	Total_Orders
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```sql
SELECT
    HOUR(order_time) Hour_of_the_Day,
    COUNT(order_id)  Orders_in_the_Hour
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY Hour_of_the_Day;
```

Result Grid | Filter Rows:

Hour_of_the_Day	Orders_in_the_Hour
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

Find the category wise distribution of pizzas

```sql
SELECT
    category AS Category, COUNT(pizza_type_id) AS Pizza_Count
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rc

Category	Pizza_Count
Chicken	6
Classic	8
Supreme	9
Veggie	9

Determine the top 3 most ordered pizza types based on revenue

```sql
SELECT
    t.category AS Pizza_Type,
    ROUND(SUM(o.quantity * p.price), 2) AS Revenue
FROM
    pizza_types t
        JOIN
    pizzas p ON p.pizza_type_id = t.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY t.category
ORDER BY revenue desc LIMIT 3;
```

Pizza_Type	Revenue
Classic	220053.1
Supreme	208197
Chicken	195919.5

Group the orders by date & Calculate the average no. of pizzas ordered per day

```sql
SELECT
    ROUND(AVG(qty), 0) as Avg_No_Pizza_per_Day
FROM
    (SELECT
        (order_date), SUM(quantity) AS qty
    FROM
        orders o
    JOIN order_details d ON o.order_id = d.order_id
    GROUP BY (order_date)) AS date;
```

Result Grid | Filter

	Avg_No_Pizza_per_Day
▶	138

Determine the top 3 most ordered pizza types based on revenue

```sql
SELECT
    t.name AS Pizza_Type,
    ROUND(SUM(o.quantity * p.price), 2) AS Revenue
FROM  pizza_types t
        JOIN
    pizzas p ON p.pizza_type_id = t.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY t.name
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

ADVANCED ANALYSIS

Rank pizzas based on revenue for each pizza category

```
select  t.Category, t.Name,sum(d.quantity * p.price) AS Revenue ,
dense_rank()
over
(partition by  t.category order by sum(d.quantity * p.price) desc)
as Pizza_Rank
from orders o
        JOIN
    order_details d ON o.order_id = d.order_id
        JOIN
    pizzas p ON p.pizza_id = d.pizza_id
        JOIN
    pizza_types t on t.pizza_type_id=p.pizza_type_id
    group by  t.category, t.name;
```

Category	Name	Revenue	Pizza_Rank
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Chicken	The Southwest Chicken Pizza	34705.75	4
Chicken	The Chicken Alfredo Pizza	16900.25	5
Chicken	The Chicken Pesto Pizza	16701.75	6
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Classic	The Greek Pizza	28454.10...	4
Classic	The Italian Capocollo Pizza	25094	5
Classic	The Napolitana Pizza	24087	6
Classic	The Big Meat Pizza	22968	7
Classic	The Pepperoni, Mushroom, ...	18834.5	8
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Supreme	The Pepper Salami Pizza	25529	4
Supreme	The Prosciutto and Arugula ...	24193.25	5
Supreme	The Soppressata Pizza	16425.75	6

Calculate the percentage contribution of each pizza type to total revenue

```sql
SELECT Category, concat (ROUND(((ROUND(SUM(quantity * price), 2) / (SELECT
                    ROUND(SUM(o.quantity * p.price), 2) AS Revenue
                FROM order_details o
                        JOIN
                pizzas p ON o.pizza_id = p.pizza_id)) * 100),
            2) ,'%')AS Category_Rev_Percentage
FROM pizza_types t
        JOIN
    pizzas p ON p.pizza_type_id = t.pizza_type_id
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY category;
```

Category	Category_Rev_Percentage
Classic	26.91%
Veggie	23.68%
Supreme	25.46%
Chicken	23.96%

Analyze the cumulative revenue generated over time

```sql
SELECT order_date,
round(sum(Revenue) over(order by order_date),2) AS Cumulative_Revenue FROM
(SELECT
    Order_date, ROUND(SUM(d.quantity * p.price), 2) AS Revenue
FROM
    orders o
        JOIN
    order_details d ON o.order_id = d.order_id
        JOIN
    pizzas p ON p.pizza_id = d.pizza_id
GROUP BY order_date
ORDER BY o.order_date) as Data;
```

order_date	Cumulative_Revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65

order_date	Cumulative_Revenue
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1
2015-02-03	77925.9
2015-02-04	80159.8
2015-02-05	82375.6
2015-02-06	84885.55
2015-02-07	87123.2
2015-02-08	89158.2
2015-02-09	91353.55

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

```sql
SELECT Category, Name, Revenue FROM
    (SELECT  t.category, t.name,sum(d.quantity * p.price) AS Revenue ,
    dense_rank()
    OVER (partition by  t.category order by sum(d.quantity * p.price) desc) AS rnk
    FROM orders o
        JOIN
    order_details d ON o.order_id = d.order_id
        JOIN
    pizzas p ON p.pizza_id = d.pizza_id
        JOIN
    pizza_types t ON t.pizza_type_id=p.pizza_type_id
    GROUP BY  t.category, t.name) AS Data 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....
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU

This SQL-based market and sales analysis provided valuable insights into PizzaHut's customer base, sales performance, and market trends. These insights can be used to optimize business strategies and predict future sales for improved decision-making and long-term growth.