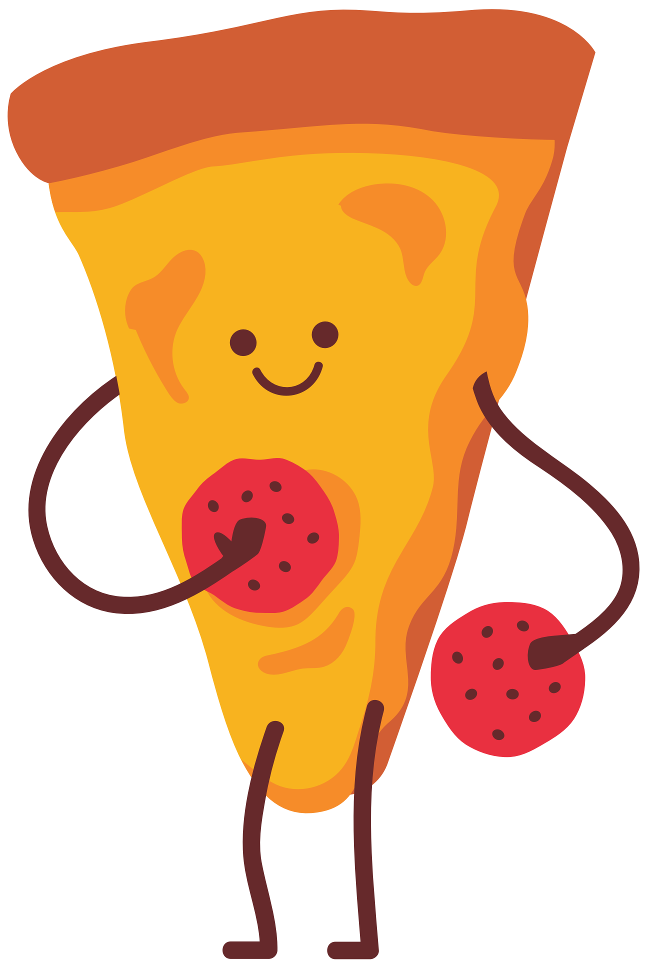
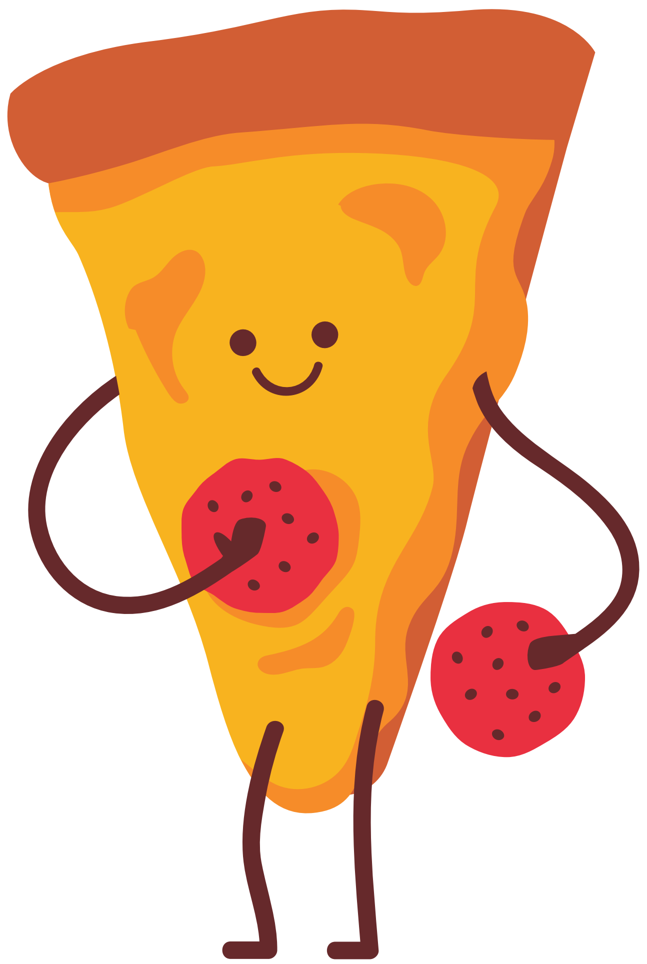
PIZZA

SALES REPORT



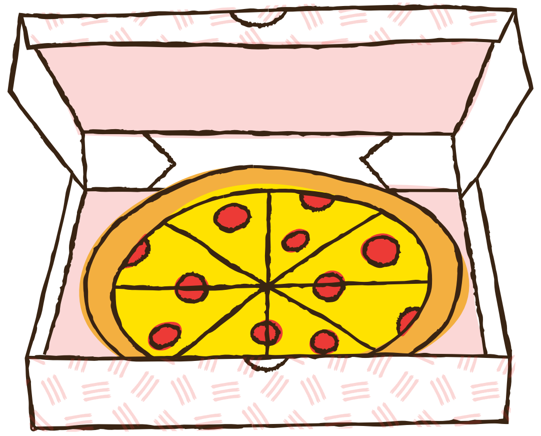


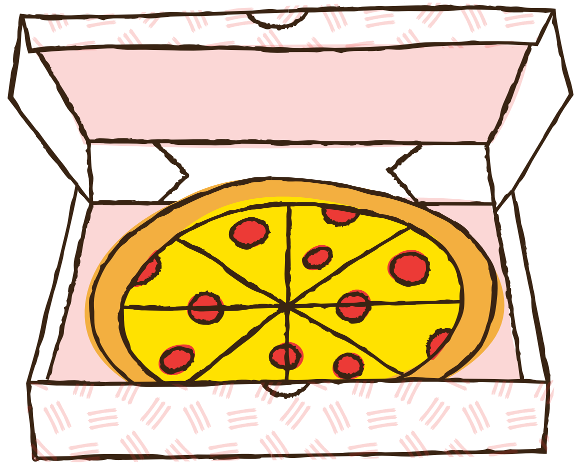
DELICIOUS PIZZA FOR EVERY ONE !

Hello ! ,My name is Priyanshu Bhardwaj , and in this project i have utilized SQL queries to solve the questions , which are related to analyzing pizza sales , so that data can provide valuable insights into customer preferences, popular toppings, sales trends, and more.

Exploring Pizza Sales Data: Insights from Kaggle

Analyzing pizza sales data retrieved from Kaggle, this dataset offers insights into customer preferences, popular toppings, sales trends, and other key metrics, enabling comprehensive understanding and informed decision-making within the pizza industry





PizzaHut Database Overview:

The PizzaHut Schema comprises four primary tables:

1. order_details: This table contains detailed information about each pizza order, including the quantity, size, toppings, and total price.
2. orders: The orders table stores general information about each order, such as the order ID, customer ID, order date, and delivery status.
3. pizza_types: This table lists the various types of pizzas offered by PizzaHut, along with their unique identifiers and prices.
4. pizza: The pizza table provides additional details about each pizza, such as the crust type, sauce, cheese, and any additional toppings.

Retrive the total number of orders placed ?

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

Result Grid

	total
▶	21350

Calculate the total revenue generated from pizza sales.?

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

Result Grid

	total_revenue
▶	817860.05

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

Identify the most common pizza size ordered. ?

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

Result Grid | Filter

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

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

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

Result Grid | Filter Rows:

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

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

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

Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS hour_of_day,
    COUNT(*) AS order_count
FROM
    orders
GROUP BY
    HOUR(order_time);
```

Result Grid | Filter Rows:

hour_of_day	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

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

```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid Filter Rows:

avg_pizza_ordered_per_day
138

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

```
select pizza_types.name,
 SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.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

Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.category,
  round(  SUM(order_details.quantity * pizzas.price),0) AS revenue,
   round((SUM(order_details.quantity * pizzas.price) / total_revenue.total),3) * 100 AS revenue_percentage
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    (SELECT
        SUM(order_details.quantity * pizzas.price) AS total
    FROM
        order_details
    JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id) AS total_revenue
GROUP BY pizza_types.category , total_revenue.total
ORDER BY revenue DESC;
```

Result Grid | Filter Rows:

category	revenue	revenue_percentage
Veggie	193690	23.7
Supreme	208197	25.5
Classic	220053	26.900000000000002
Chicken	195920	24

Analyze the cumulative revenue generated over time.

```
SELECT
    sales.order_date,
    SUM(sales.revenue) OVER (ORDER BY sales.order_date) AS cum_revenue
FROM
    (
    SELECT
        orders.order_date,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM
        order_details
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
    JOIN
        orders ON orders.order_id = order_details.order_id
    GROUP BY
        orders.order_date
    ) AS sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001

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

```sql
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((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category , pizza_types.name) as a) as b;
where rn <= 3
```

Result Grid | Filter Rows: | Export:

name	revenue
The Prosciutto and Arugula ...	24193.25
The Soppressata Pizza	16425.75
The Calabrese Pizza	15934.25
The Spinach Supreme Pizza	15277.75
The Brie Carre Pizza	11588.4999999999
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5
The Vegetables + Vegetable...	24374.75
The Spinach and Feta Pizza	23271.25
The Italian Vegetables Pizza	16019.25
The Spinach Pesto Pizza	15596
The Mediterranean Pizza	15360.5
The Green Garden Pizza	13955.75

PIZZA PARTY!