PIZZA HUT SALES - SQL ANALYSIS

ABOUT THE PROJECT

For this Pizza Sales Analysis project, I used SQL to dive into sales data and uncover key insights about customer preferences, top-selling pizzas, and revenue trends. The goal was to analyze sales performance, identify peak category selling, and suggest data-driven improvements for better business decisions. I worked with a structured pizza sales dataset, which included tables for orders, customers, pizzas, and types of pizza.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid

	Total_Orders
▶	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
    price AS Highest_Price
FROM
    pizzas
ORDER BY price DESC
LIMIT 1;
```

Result Grid

	Highest_Price
▶	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid | Filter Rov

size	Number_of_Orders
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 AS Pizza_Type,
    SUM(order_details.quantity) AS Quantity_Ordered
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.name
ORDER BY COUNT(order_details_id) DESC
LIMIT 5;
```

Result Grid | Filter Rows:

Pizza_Type	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

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

```
SELECT
    pizza_types.category AS Pizza_Category,
    SUM(order_details.quantity) AS Quantity_Ordered
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;
```

Result Grid | Filter Rows:

Pizza_Category	Quantity_Ordered
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY HOUR(order_time) ASC;
```

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
9	1
10	8
11	1231
12	2520
13	2455

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

```sql
SELECT
    pizza_types.category AS Pizza_Category,
    COUNT(order_details.order_id) AS Number_Of_Orders
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 COUNT(order_details.order_id) DESC;
```

Result Grid | Filter Rows:

Pizza_Category	Number_Of_Orders
Classic	14579
Supreme	11777
Veggie	11449
Chicken	10815

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

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

Result Grid

Avg_Pizza_Per_Day
138

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

```sql
SELECT
    pizza_types.name AS Pizza_Type,
    SUM(order_details.quantity * pizzas.price) AS Total_Revenue
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.name
ORDER BY SUM(order_details.quantity * pizzas.price) DESC
LIMIT 3;
```

Result Grid | Filter Rows:

Pizza_Type	Total_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) / (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) * 100),
            2) AS revenue_in_percent
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 revenue_in_percent DESC;
```

Result Grid | Filter Rows:

category	revenue_in_percent
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

Result Grid | Filter Rows:

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

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

```sql
SELECT name, category, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS ranking
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category, pizza_types.name)
AS a) AS b
WHERE ranking <=3;
```

Result Grid | Filter Rows:

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25

DATA SET AND SQL FILE

GITHUB LINK:

HTTPS://GITHUB.COM/PRIYAMSRI/PIZZA_HUT_SALES_SQL_ANALYSIS.GIT

THANK YOU!

COMPILED BY -: PRIYAM SRIVASTAVA