

# BURGERHUT'S SALES ANALYSIS

Optimizing Business Strategies through  
Data Insights

Prepared by: Maheen Khan

<https://www.linkedin.com/in/maheen-khan-134a86209/>



# ABOUT THE PROJECT

## Description

This project explores BurgerHut's order data using SQL queries to derive insights into sales performance, customer preferences, and revenue generation, including identifying top burger categories, peak ordering times, and cumulative revenue trends over time.



# DATA OVERVIEW

The dataset used in this project provides a comprehensive overview of a restaurant's sales activities. It includes key attributes such as **Order ID, burger types, sizes, prices, dates, quantities, and categories**. Each entry represents a unique order, capturing critical details that enable the analysis of sales patterns and customer preferences. This dataset serves as the backbone of the project, allowing for insights into revenue generation, popular burger choices, and order trends, all of which are crucial for optimizing business strategies.

## The total number of orders placed

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

	total_orders
▶	21350

## The total revenue generated from burger sales

```
SELECT  
    ROUND(SUM(order_details.quantity * burgers.price), 2) AS total_sales  
FROM order_details  
JOIN burgers ON burgers.burger_id = order_details.burger_id;
```

	total_sales
▶	\$17860.05

## The total revenue generated from burger sales

- ```
SELECT  
    burger_types.name, burgers.price  
FROM  
    burger_types  
    JOIN  
    burgers ON burger_types.burger_type_id = burgers.burger_type_id  
ORDER BY burgers.price DESC  
LIMIT 1;
```

|   | name              | price |
|---|-------------------|-------|
| ▶ | The Mighty Burger | 35.95 |



## Identify the most common burger size ordered

```

SELECT
    burgers.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    burgers
        JOIN
            order_details ON burgers.burger_id = order_details.burger_id
GROUP BY burgers.size
ORDER BY order_count DESC;
-- Identify the most common pizza size ordered.
SELECT

```

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

## The total revenue generated from burger sales

```

SELECT
    burger_types.name, SUM(order_details.quantity) AS quantity
FROM
    burger_types
        JOIN
            burgers ON burger_types.burger_type_id = burgers.burger_type_id
                JOIN
                    order_details ON order_details.burger_id = burgers.burger_id
GROUP BY burger_types.name
ORDER BY quantity DESC
LIMIT 5;

```

|   | Burger Name                 | Quantity |
|---|-----------------------------|----------|
| ▶ | The Classic Zinger Burger   | 2,453    |
|   | The Barbecue Chicken Burger | 2,432    |
|   | The Hawaiian Burger         | 2,422    |
|   | The Spicy Beef Burger       | 2,418    |
|   | The Thai Chicken Burger     | 2,371    |



# Total quantity of each hamburger category ordered

```
SELECT
    hamburger_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    hamburger_types
    JOIN
    hamburgers ON hamburger_types.hamburger_type_id = hamburgers.hamburger_type_id
    JOIN
    order_details ON order_details.hamburger_id = hamburgers.hamburger_id
GROUP BY hamburger_types.category
ORDER BY quantity DESC;
```



|   | Category       | Quantity |
|---|----------------|----------|
| ▶ | Classic Burger | 14,888   |
|   | Supreme Burger | 11,987   |
|   | Veggie Burger  | 11,649   |
|   | Chicken Burger | 11,050   |

# 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(order_time);
```

|   | 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           |

## Category-wise distribution of hamburgers

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

|   | category | count(name) |
|---|----------|-------------|
| ▶ | Chicken  | 6           |
|   | Classic  | 8           |
|   | Supreme  | 9           |
|   | Veggie   | 9           |

```
SELECT
    ROUND(AVG(quantity), 0)
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;
```

|   | ROUND(AVG(quantity), 0) |
|---|-------------------------|
| ▶ | 138                     |

## Top 3 most ordered hamburger types based on revenue

```
SELECT
    hamburger_types.name,
    SUM(order_details.quantity * hamburgers.price) AS revenue
FROM
    hamburger_types
    JOIN
    hamburgers ON hamburgers.hamburger_type_id = hamburger_types.hamburger_type_id
    JOIN
    order_details ON order_details.hamburger_id = hamburgers.hamburger_id
GROUP BY hamburger_types.name
ORDER BY revenue DESC
LIMIT 3;
```

|   | name                          | revenue |
|---|-------------------------------|---------|
| ▶ | The Thai Chicken Burger       | 43,434  |
|   | The Barbecue Chicken Burger   | 42,768  |
|   | The California Chicken Burger | 41,410  |



## Percentage contribution of each hamburger category to total revenue

```

SELECT
    hamburger_types.category,
    (SUM(order_details.quantity * hamburgers.price) / (SELECT
        ROUND(SUM(order_details.quantity * hamburgers.price),
        2) AS total_sales
    )
    FROM
        order_details
        JOIN
            hamburgers ON hamburgers.hamburger_id = order_details.hamburger_id)) * 100 AS revenue
    FROM
        hamburger_types
        JOIN
            hamburgers ON hamburger_types.hamburger_type_id = hamburgers.hamburger_type_id
        JOIN
            order_details ON order_details.hamburger_id = hamburgers.hamburger_id
    GROUP BY hamburger_types.category
    ORDER BY revenue DESC;

```

|   | category        | revenue |
|---|-----------------|---------|
| ▶ | Classic Burgers | 26.91%  |
|   | Supreme Burgers | 25.46%  |
|   | Chicken Burgers | 23.96%  |
|   | Veggie Burgers  | 23.68%  |

## Cumulative revenue generated over time

```

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

```

| 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 | 23990.35000000002  |
| 2015-01-11 | 25862.65           |
| 2015-01-12 | 27781.7            |
| 2015-01-13 | 29831.30000000003  |
| 2015-01-14 | 32358.70000000004  |
| 2015-01-15 | 34343.5000000001   |
| 2015-01-16 | 36937.6500000001   |
| 2015-01-17 | 39001.7500000001   |



# RECOMMENDATIONS AND KEY TAKEAWAYS



01

Based on the analysis, BurgerHut should focus on promoting its top-selling items like the Classic Zinger Burger and Barbecue Chicken Burger, which have consistently high order volumes. These popular burgers can be featured in marketing campaigns and combo deals to increase their visibility and encourage repeat orders. Additionally, the higher-priced Mighty Burger offers an opportunity to boost revenue, and promoting it with special offers or exclusive deals could attract customers looking for premium options.

02

The data shows that Classic Burgers and Supreme Burgers contribute the most to overall sales, while Chicken and Veggie Burgers slightly lag behind. By developing targeted promotions or adding new variations in these categories, BurgerHut can increase sales and balance revenue distribution across its product lines. Focusing on upselling larger sizes like 'L', the most commonly ordered size, can also help improve sales.

03

Lastly, BurgerHut should utilize the cumulative revenue trends to identify periods of slower sales and introduce time-specific promotions or discounts to boost performance during these times. Additionally, reassessing low-performing items, like the XXL size, could streamline operations and allow for more focus on high-demand products, improving overall efficiency and customer satisfaction.

# THANK YOU

