

# PIZZA HUT SALES

By- Shubham Kumar



# PIZZA HUT SALES

## Pizza Hut Sales Analysis Using SQL

- Objective: Analyze Pizza Hut's sales data to gain insights into performance trends, and product popularity.
- Dataset: Includes transaction details such as order IDs, product types, quantities, sales values, timestamps.

### Key Metrics:

- Total sales revenue.
- Most popular menu items.
- Sales distribution by region and store.
- Peak sales times and days.
- Customer purchase frequency and order value trends.
- Techniques: Data filtering, aggregation, joins, and trends using SQL.





# SQL TECHNIQUES USED

- Data filtering (WHERE clauses) to isolate specific periods or regions.
- Aggregation functions (SUM, AVG, COUNT) for revenue and volume analysis.
- Joins to combine sales with customer data for deeper insights.
- Subqueries and window functions to identify trends and rankings.

## FINDINGS:

- Highest revenue-generating items .
- Optimal sales periods
- Impact: Improved decision-making for inventory management, promotions, and customer engagement strategies.

## CREATE TABLE

```
CREATE TABLE orders(  
    order_id int not null,  
    order_date date not null,  
    order_time time not null,  
    primary key(order_id));
```

```
CREATE TABLE order_details (  
    order_details_id INT NOT NULL,  
    order_id INT NOT NULL,  
    pizza_id TEXT NOT NULL,  
    quantity INT NOT NULL,  
    PRIMARY KEY (order_details_id)  
);
```

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SELECT COUNT (ORDER\_ID) AS TOTAL\_ORDERS FROM ORDERS;

-- Calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_sales
▶	817860.05

## -- IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95

## -- Identify the most common pizza size ordered.

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

| Result Grid |      | Filter Rows: |
|-------------|------|--------------|
|             | 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.

```
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;
```

|   | name                     | quantity                 |
|---|--------------------------|--------------------------|
| ▶ | The Classic Deluxe Pizza | 2453                     |
|   | The Barbecue Chicken     | The Classic Deluxe Pizza |
|   | 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,
    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;
```

|   | category | quantity |
|---|----------|----------|
| ▶ | Classic  | 14888    |
|   | Supreme  | 11987    |
|   | Veggie   | 11649    |
|   | Chicken  | 11050    |

-- 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(order_time)  
LIMIT 5;
```

| Result Grid |      | Filter Rows |
|-------------|------|-------------|
|             | hour | order_count |
| >           | 11   | 1231        |
|             | 12   | 2520        |
|             | 13   | 2455        |
|             | 14   | 1472        |
|             | 15   | 1468        |

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

```
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.

```
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 |                           |
|-------------|---------------------------|
|             | 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 |                              |          |
|-------------|------------------------------|----------|
|             | 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) / (SELECT
        SUM(order_details.quantity * pizzas.price)
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
    2) AS revenue_percentage
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
ORDER BY revenue_percentage DESC;
```

|   | category | revenue_percentage |
|---|----------|--------------------|
| ▶ | Classic  | 26.91              |
|   | Supreme  | 25.46              |
|   | Chicken  | 23.96              |
|   | Veggie   | 23.68              |

-- ANALYZE THE 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 * 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 limit 3;
```

|   | order_date | cum_revenue        |
|---|------------|--------------------|
| ▶ | 2015-01-01 | 2713.8500000000004 |
|   | 2015-01-02 | 5445.75            |
|   | 2015-01-03 | 8108.15            |

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

```
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 limit 5;
```

| Result Grid |                              |
|-------------|------------------------------|
|             |                              |
|             | name                         |
| ▶           | The Thai Chicken Pizza       |
|             | The Barbecue Chicken Pizza   |
|             | The California Chicken Pizza |
|             | The Classic Deluxe Pizza     |
|             | The Hawaiian Pizza           |

# THANK YOU!

