



# Pizza Sales Analysis

@ Rishabh Bisht (SQL Project)

# Introduction

- Objective: Analyze pizza sales data to identify trends, popular items, and potential areas for growth.
- Data Sources: Description of the datasets used, such as sales transactions, customer details, and pizza inventory.

# SQL Query Techniques Used:

- Data Cleaning and Preparation: Methods used to handle missing values, normalize data, and join multiple tables.
- Aggregate Functions: Use of SUM(), AVG(), COUNT() and GROUP BY to summarize the sales data.
- Analytical Functions: Examples of using 'WINDOW' function like 'ROW\_NUMBER()', 'RANK()' and 'PARTITION BY' for advance analysis.

# Key Insights and Finding

- Top-Selling Pizzas : Identification of the most popular pizzas based on sales and revenue.
- Sales Trends : Analysis of sales pattern over time(e.g., peak sales period, seasonal trends).
- Customer Insights : Segmentation of customers based on purchasing behavior and preference.

# BASICS



# 1. Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.  
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    pizzahut.orders;
```

	total_orders
▶	21350

## 2. Calculate the total revenue generated from pizza sales.

```
-- Calculate the total revenue generated from pizza sales.  
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
          3) AS total_sales  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

	total_sales
▶	851518.7

### 3. Identify the highest-priced pizza.

```
-- 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;
```

	name	price
▶	The Greek Pizza	35.95



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

```
-- Identify the most common pizza size ordered.  
SELECT  
    pizzas.size,  
    COUNT(orders_details.order_details_id) AS order_count  
FROM  
    pizzas  
    JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY order_count DESC;
```

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

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

```
-- List the top 5 most ordered pizza types along with their quantities.  
SELECT  
    pizza_types.name,  
    SUM(orders_details.quantity) AS total_quantity  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY total_quantity DESC  
LIMIT 5;
```

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

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

```
-- List the top 5 most ordered pizza types along with their quantities.  
SELECT  
    pizza_types.name,  
    SUM(orders_details.quantity) AS total_quantity  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY total_quantity DESC  
LIMIT 5;
```

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

# INTERMEDIATE

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

```
-- Join the necessary tables to find the total quantity of each pizza category ordered.  
SELECT  
    pizza_types.category,  
    SUM(orders_details.quantity) AS quantity  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



## 2. Determine the distribution of orders by hour of the day.

```
-- 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);
```

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

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

```
-- Join relevant tables to find the category-wise distribution of pizzas.  
select category, count(name) from pizza_types group by category;
```

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

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

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.  
SELECT  
    ROUND(AVG(quantity), 0) 'average pizza ordered per day'  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) as quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

	average pizza ordered per day
▶	138

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

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

```
SELECT
```

```
    pizza_types.name,
```

```
    SUM(orders_details.quantity * pizzas.price) AS revenue
```

```
FROM
```

```
    pizza_types
```

```
    JOIN
```

```
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

```
    JOIN
```

```
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
```

```
GROUP BY pizza_types.name
```

```
ORDER BY revenue DESC
```

```
LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

**ADVANCE**



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

```
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    CONCAT(ROUND((SUM(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
    3) AS total_sales
    FROM
        orders_details
        JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100,
    2),
    '%') AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	25.84%
Supreme	24.45%
Chicken	23.01%
Veggie	22.75%

## 2. Analyze the cumulative revenue generated over time.

```
-- 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(orders_details.quantity*pizzas.price) as revenue
```

```
from orders_details join pizzas
```

```
on orders_details.pizza_id=pizzas.pizza_id
```

```
join orders
```

```
on orders.order_id=orders_details.order_id
```

```
group by 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	23990.350000000002
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

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

```
-- 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) as ranks from
(select pizza_types.category, pizza_types.name,
sum((orders_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as A)as B
where ranks<=3;
```

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75
The Calabrese Pizza	15934.25
The Green Garden Pizza	13955.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596