

# Project

## Pizza Sales Analysis

### -- Creating Database

```
CREATE DATABASE Pizzahut;
```

### -- Use Database

```
USE Pizzahut;
```

### -- Create Tables

```
CREATE TABLE IF NOT EXISTS orders(  
  order_id INT PRIMARY KEY,  
  order_date TEXT NOT NULL,  
  order_time TEXT NOT NULL  
);
```

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

```
CREATE TABLE IF NOT EXISTS pizzas(  
  pizza_id VARCHAR(50),  
  pizza_type_id VARCHAR(100) ,  
  size VARCHAR(100),  
  price DOUBLE
```

```
);
```

```
CREATE TABLE IF NOT EXISTS pizza_types(  
    pizza_type_id VARCHAR(100),  
    name VARCHAR(100),  
    category VARCHAR(50),  
    ingredients VARCHAR(250)  
);
```

```
SELECT * FROM orders  
SELECT * FROM order_details  
SELECT * FROM pizzas  
SELECT * FROM pizza_types
```

**-- 1. Retrieve the total number of total placed.**

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

**-- 2. Calculate the total revenue generated from pizza sales.**

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

**-- 3. Identify the highest-price pizza.**

```
SELECT pt.name, p.price  
FROM pizzas p  
JOIN pizza_types pt
```

```
ON p.pizza_type_id = pt.pizza_type_id
```

```
ORDER BY price DESC
```

```
LIMIT 1;
```

**-- 4. Identify the most common pizza size ordered.**

```
SELECT p.size, COUNT(od.order_detail_id) AS Count
```

```
FROM pizzas p
```

```
JOIN order_details od
```

```
ON p.pizza_id = od.pizza_id
```

```
GROUP BY p.size
```

```
ORDER BY Count DESC;
```

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

```
SELECT pt.name, SUM(od.quantity) AS Quantity
```

```
FROM pizzas p
```

```
JOIN pizza_types pt
```

```
ON p.pizza_type_id = pt.pizza_type_id
```

```
JOIN order_details od
```

```
ON od.pizza_id = p.pizza_id
```

```
GROUP BY pt.name
```

```
ORDER BY Quantity
```

```
LIMIT 5;
```

**-- 6. Join the necessary tables to find the total quantity of each pizza category.**

```
SELECT pt.category, SUM(od.quantity) AS Quantity
```

```
FROM pizza_types pt
```

```
JOIN pizzas p
```

```
ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY Quantity DESC;
```

**-- 7. Determine the distribution of orders by hour of the day.**

```
SELECT HOUR(order_time) AS Hours, COUNT(order_id) AS Order_Count
FROM orders
GROUP BY Hours;
```

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

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

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

```
SELECT ROUND(AVG(Quantity) ,2) as Average_Pizza_Ordered_Per_Day
FROM
(SELECT o.order_date, SUM(od.quantity) AS Quantity
FROM orders o
JOIN order_details od
ON o.order_id = od.order_id
GROUP BY o.order_date) AS Order_Quantity;
```

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

```
SELECT pt.name, ROUND(SUM(od.quantity * p.price),2) AS Revenue
FROM pizza_types pt
```

```

JOIN pizzas p
ON p.pizza_type_id = pt.pizza_type_id
JOIN order_details od
ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY Revenue DESC
LIMIT 3;

```

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

```

SELECT pt.category, ROUND((SUM(od.quantity*p.price) / (SELECT
ROUND(SUM(od.quantity*p.price),2) AS Total_Sales

FROM order_details od

JOIN pizzas p
ON od.pizza_id = p.pizza_id ))*100,2) AS Revenue
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY Revenue DESC;

```

**-- 12. Analyze the cumulative revenue generated over time.**

```

SELECT order_date, SUM(Revenue) OVER(ORDER BY order_date) AS Cum_Revenue
FROM
(SELECT o.order_date, SUM(od.quantity*p.price) AS Revenue
FROM order_details od
JOIN pizzas p

```

```
ON od.pizza_id = p.pizza_id  
JOIN orders o  
ON od.order_id = o.order_id  
GROUP BY o.order_date) AS Sales;
```

**-- 13. Determine the top 3 ordered pizza types based on revenue for each pizza category.**

```
SELECT rn, category, name, revenue  
FROM  
(SELECT category, name, revenue,  
RANK() OVER(  
PARTITION BY category  
ORDER BY revenue DESC) AS rn  
FROM  
(SELECT pt.category, pt.name, SUM((od.quantity)*p.price) AS revenue  
FROM pizza_types pt  
JOIN pizzas p  
ON pt.pizza_type_id = p.pizza_type_id  
JOIN order_details od  
ON od.pizza_id = p.pizza_id  
GROUP BY pt.category, pt.name) AS a) AS b  
WHERE rn <=3;
```