**Task 5 – Food Booking Application**

CREATE DATABASE Zomato;

USE Zomato;

**Create Tables –**

**Restaurants Table –**

CREATE TABLE Restaurants ( RestaurantID INT PRIMARY KEY, Name VARCHAR(100), City VARCHAR(50), CuisineType VARCHAR(50), Rating DECIMAL(2, 1), AverageCostForTwo INT );

**Customers Table –**

CREATE TABLE Customers ( CustomerID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Phone VARCHAR(15), City VARCHAR(50), JoinDate DATE );

**Orders Table –**

CREATE TABLE Orders ( OrderID INT PRIMARY KEY, CustomerID INT, RestaurantID INT, OrderDate DATE, OrderAmount DECIMAL(10, 2), OrderStatus VARCHAR(20), FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID), FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID) );

**Reviews Table –**

CREATE TABLE Reviews ( ReviewID INT PRIMARY KEY, CustomerID INT, RestaurantID INT, Rating DECIMAL(2, 1), Comment TEXT, ReviewDate DATE, FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID), FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID) );

**Payments Table –**

CREATE TABLE Payments ( PaymentID INT PRIMARY KEY, OrderID INT, PaymentMethod VARCHAR(20), Amount DECIMAL(10, 2), PaymentDate DATE, FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) );

**Insert data in the Table –**

**Restaurants Table –**

INSERT INTO Restaurants (RestaurantID, Name, City, CuisineType, Rating, AverageCostForTwo) VALUES (1, 'Saravana Bhavan', 'Chennai', 'South Indian', 4.8, 500), (2, 'Biryani Blues', 'Delhi', 'Indian', 4.5, 600), (3, 'Pizza Hut', 'Mumbai', 'Italian', 4.2, 700), (4, 'Mainland China', 'Kolkata', 'Chinese', 4.3, 800), (5, 'Punjab Grill', 'Delhi', 'North Indian', 4.6, 1000), (6, 'Barbeque Nation', 'Bangalore', 'Indian', 4.4, 1500), (7, 'Haldiram\'s', 'Noida', 'Indian', 4.1, 400), (8, 'Cafe Coffee Day', 'Hyderabad', 'Cafe', 4.0, 300), (9, 'KFC', 'Mumbai', 'Fast Food', 4.3, 500), (10, 'Paradise Biryani', 'Hyderabad', 'Indian', 4.7, 700);

**Customers Table –**

INSERT INTO Customers (CustomerID, FirstName, LastName, Phone, City, JoinDate) VALUES (1, 'Rahul', 'Sharma', '9876543210', 'Mumbai', '2023-01-01'), (2, 'Priya', 'Singh', '9876543211', 'Delhi', '2023-01-15'), (3, 'Amit', 'Kumar', '9876543212', 'Chennai', '2023-02-10'), (4, 'Sneha', 'Roy', '9876543213', 'Kolkata', '2023-03-05'), (5, 'Ravi', 'Verma', '9876543214', 'Bangalore', '2023-04-01'), (6, 'Pooja', 'Iyer', '9876543215', 'Hyderabad', '2023-04-20'), (7, 'Vikram', 'Naik', '9876543216', 'Pune', '2023-05-15'), (8, 'Simran', 'Gupta', '9876543217', 'Noida', '2023-06-01'), (9, 'Ankit', 'Mehta', '9876543218', 'Mumbai', '2023-07-10'), (10, 'Neha', 'Joshi', '9876543219', 'Chennai', '2023-08-25');

**Orders Table –**

INSERT INTO Orders (OrderID, CustomerID, RestaurantID, OrderDate, OrderAmount, OrderStatus) VALUES (1, 1, 3, '2023-09-01', 750.00, 'Delivered'), (2, 2, 2, '2023-09-05', 650.00, 'Delivered'), (3, 3, 1, '2023-09-10', 550.00, 'Cancelled'), (4, 4, 4, '2023-09-15', 850.00, 'Delivered'), (5, 5, 5, '2023-09-20', 1000.00, 'Delivered'), (6, 6, 6, '2023-09-25', 1450.00, 'Pending'), (7, 7, 7, '2023-10-01', 400.00, 'Delivered'), (8, 8, 8, '2023-10-05', 350.00, 'Delivered'), (9, 9, 9, '2023-10-10', 550.00, 'Cancelled'), (10, 10, 10, '2023-10-15', 750.00, 'Delivered');

**Reviews Table –**

INSERT INTO Reviews (ReviewID, CustomerID, RestaurantID, Rating, Comment, ReviewDate) VALUES (1, 1, 3, 4.5, 'Great food!', '2023-09-02'), (2, 2, 2, 4.7, 'Amazing biryani!', '2023-09-06'), (3, 3, 1, 4.8, 'Authentic South Indian dishes.', '2023-09-11'), (4, 4, 4, 4.2, 'Good experience.', '2023-09-16'), (5, 5, 5, 4.9, 'Exceptional food quality.', '2023-09-21'), (6, 6, 6, 4.3, 'Nice ambiance.', '2023-09-26'), (7, 7, 7, 4.1, 'Value for money.', '2023-10-02'), (8, 8, 8, 3.9, 'Decent coffee.', '2023-10-06'), (9, 9, 9, 4.4, 'Crispy chicken.', '2023-10-11'), (10, 10, 10, 4.6, 'Best biryani in town!', '2023-10-16');

**Payments Table –**

INSERT INTO Payments (PaymentID, OrderID, PaymentMethod, Amount, PaymentDate) VALUES (1, 1, 'Card', 750.00, '2023-09-01'), (2, 2, 'Cash', 650.00, '2023-09-05'), (3, 4, 'Wallet', 850.00, '2023-09-15'), (4, 5, 'Card', 1000.00, '2023-09-20'), (5, 7, 'Cash', 400.00, '2023-10-01'), (6, 8, 'Wallet', 350.00, '2023-10-05'), (7, 10, 'Card', 750.00, '2023-10-15'), (8, 6, 'Card', 1450.00, '2023-09-25'), (9, 9, 'Wallet', 550.00, '2023-10-10'), (10, 3, 'Cash', 550.00, '2023-09-10');

**Assignment Queries**

1. **Retrieve the names and locations of restaurants with a rating of 4.5 or higher.**

SELECT Name, City FROM Restaurants WHERE Rating >= 4.5;

1. **Find the total number of orders placed by each customer.**

SELECT c.CustomerID,c.FirstName,c.LastName ,

COUNT(o.OrderID) AS TotalOrders

FROM Customers as c

LEFT JOIN Orders as o

ON c.CustomerID = o.CustomerID

GROUP BY c.CustomerID, c.FirstName, c.LastName;

1. **List all restaurants offering "Italian" cuisine in "Mumbai".**

SELECT Name, City FROM Restaurants WHERE CuisineType = 'Italian' AND City = 'Mumbai';

1. **Calculate the total revenue generated by each restaurant from completed orders.**

SELECT R.RestaurantID, R.Name, SUM(O.OrderAmount) AS TotalRevenue

FROM Restaurants as R

JOIN Orders as O

ON R.RestaurantID = O.RestaurantID

WHERE O.OrderStatus = 'Delivered'

GROUP BY R.RestaurantID, R.Name;

1. **Retrieve the most recent order placed by each customer.**

SELECT C.CustomerID,C.FirstName,C.LastName,

MAX(O.OrderDate) AS MostRecentOrderDate

FROM Customers as C

JOIN Orders as O

ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.FirstName, C.LastName;

**6. List customers who have not placed any orders yet.**

SELECT CustomerID,FirstName ,LastName

FROM Customers

WHERE CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);

**7. Identify the most reviewed restaurants.**

SELECT R.RestaurantID, R.Name,

COUNT(Reviews.ReviewID) AS TotalReviews

FROM Restaurants as R

JOIN Reviews ON R.RestaurantID = Reviews.RestaurantID

GROUP BY R.RestaurantID, R.Name

ORDER BY TotalReviews DESC;

**8. Find the most preferred payment method.**

SELECT PaymentMethod, COUNT(PaymentID) AS UsageCount FROM Payments GROUP BY PaymentMethod ORDER BY UsageCount DESC LIMIT 1;

**9. List the top 5 restaurants by total revenue.**

SELECT R.RestaurantID, R.Name,

SUM(O.OrderAmount) AS TotalRevenue

FROM Restaurants as R

JOIN Orders as O

ON R.RestaurantID = O.RestaurantID

WHERE O.OrderStatus = 'Delivered'

GROUP BY R.RestaurantID, R.Name

ORDER BY TotalRevenue DESC

LIMIT 5;

**10. Show the details of all cancelled orders along with the customer's and restaurant's names**

SELECT O.OrderID, O.OrderDate, O.OrderAmount,

C.FirstName, C.LastName,

R.Name AS RestaurantName

FROM Orders as O

JOIN Customers as C

ON O.CustomerID = C.CustomerID

JOIN Restaurants as R

ON O.RestaurantID = R.RestaurantID

WHERE O.OrderStatus = 'Cancelled';