task-5

1.Restaurants-table

create database task-5

CREATE TABLE Restaurants (

RestaurantID INT PRIMARY KEY,

Name VARCHAR(100),

City VARCHAR(50),

CuisineType VARCHAR(50),

Rating DECIMAL(2,1),

AverageCostForTwo DECIMAL(10,2)

);

INSERT INTO Restaurants (RestaurantID, Name, City, CuisineType, Rating, AverageCostForTwo) VALUES

(1, 'Biryani Bhavan', 'Mumbai', 'Indian', 4.5, 600),

(2, 'Tandoori Flames', 'Delhi', 'Indian', 4.2, 800),

(3, 'Royal Sweets', 'Bengaluru', 'Indian', 4.7, 500),

(4, 'Masala Magic', 'Chennai', 'Indian', 3.8, 350),

(5, 'Spice Junction', 'Kolkata', 'Indian', 4.3, 750),

(6, 'Dhaba Delights', 'Amritsar', 'Indian', 4.6, 650),

(7, 'Gulab Jamun House', 'Hyderabad', 'Indian', 4.4, 550),

(8, 'South Spice', 'Chennai', 'Indian', 4.0, 700),

(9, 'Paneer Paradise', 'Lucknow', 'Indian', 3.9, 400),

(10, 'Kebab King', 'Ahmedabad', 'Indian', 4.8, 900);

==================================================================================

2.Customers-table

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Phone VARCHAR(15),

City VARCHAR(50),

JoinDate DATE

);

INSERT INTO Customers (CustomerID, FirstName, LastName, Phone, City, JoinDate) VALUES

(1, 'Amit', 'Sharma', '9876543210', 'Mumbai', '2023-05-10'),

(2, 'Priya', 'Verma', '8765432109', 'Delhi', '2024-01-01'),

(3, 'Rajesh', 'Kumar', '9654321098', 'Bengaluru', '2022-08-19'),

(4, 'Neha', 'Patel', '8543210987', 'Chennai', '2023-11-25'),

(5, 'Arjun', 'Singh', '7432109876', 'Kolkata', '2024-02-15'),

(6, 'Sunita', 'Iyer', '6321098765', 'Amritsar', '2021-12-20'),

(7, 'Vikram', 'Reddy', '5210987654', 'Hyderabad', '2022-06-10'),

(8, 'Ananya', 'Nair', '4109876543', 'Chennai', '2023-07-05'),

(9, 'Deepak', 'Yadav', '3098765432', 'Lucknow', '2023-09-30'),

(10, 'Kavya', 'Rao', '2087654321', 'Ahmedabad', '2024-03-18');

================================================================================================

3.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)

);

INSERT INTO Orders (OrderID, CustomerID, RestaurantID, OrderDate, OrderAmount, OrderStatus) VALUES

(1, 1, 1, '2024-01-02', 600, 'Delivered'),

(2, 2, 2, '2024-01-05', 800, 'Pending'),

(3, 3, 3, '2023-12-25', 500, 'Cancelled'),

(4, 4, 4, '2023-11-10', 350, 'Delivered'),

(5, 5, 5, '2024-01-08', 750, 'Delivered'),

(6, 6, 6, '2024-01-04', 650, 'Pending'),

(7, 7, 7, '2023-12-30', 550, 'Delivered'),

(8, 8, 8, '2023-12-15', 700, 'Cancelled'),

(9, 9, 9, '2024-01-03', 400, 'Delivered'),

(10, 10, 10, '2024-01-06', 900, 'Pending');

===========================================================================================

4.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)

);

INSERT INTO Reviews (ReviewID, CustomerID, RestaurantID, Rating, Comment, ReviewDate) VALUES

(1, 1, 1, 4.5, 'Delicious biryani with great taste!', '2024-01-03'),

(2, 2, 2, 4.2, 'Good tandoori dishes, but a bit spicy for my taste.', '2024-01-07'),

(3, 3, 3, 4.7, 'Excellent sweets, especially the ladoos!', '2023-12-26'),

(4, 4, 4, 3.8, 'The food was okay, but could be improved.', '2023-11-12'),

(5, 5, 5, 4.3, 'Great North Indian flavors, loved the naan.', '2024-01-09'),

(6, 6, 6, 4.6, 'Tasty food and excellent service!', '2024-01-05'),

(7, 7, 7, 4.4, 'Good food, but a little too sweet for my liking.', '2023-12-28'),

(8, 8, 8, 4.0, 'Nice south Indian food, but the portions were small.', '2023-12-20'),

(9, 9, 9, 3.9, 'Not bad, but I expected better quality.', '2024-01-02'),

(10, 10, 10, 4.8, 'Amazing kebabs and perfect spices.', '2024-01-07');

================================================================================

5.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 INTO Payments (PaymentID, OrderID, PaymentMethod, Amount, PaymentDate) VALUES

(1, 1, 'Card', 600, '2024-01-02'),

(2, 2, 'Cash', 800, '2024-01-06'),

(3, 3, 'Wallet', 500, '2023-12-25'),

(4, 4, 'Card', 350, '2023-11-10'),

(5, 5, 'Cash', 750, '2024-01-08'),

(6, 6, 'Card', 650, '2024-01-04'),

(7, 7, 'Wallet', 550, '2023-12-30'),

(8, 8, 'Cash', 700, '2023-12-15'),

(9, 9, 'Card', 400, '2024-01-03'),

(10, 10, 'Wallet', 900, '2024-01-06');

==============================================================================

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

SELECT Name, City, Rating

FROM Restaurants

WHERE Rating >= 4.5;

============================================================================================

2. Find the total number of orders placed by each customer.

SELECT CustomerID, COUNT(OrderID) AS TotalOrders

FROM Orders

GROUP BY CustomerID;

=====================================================================================

3. List all restaurants offering "Italian" cuisine in "Mumbai".

SELECT Name, City, CuisineType

FROM Restaurants

WHERE CuisineType = 'Italian' AND City = 'Mumbai';

==================================================================================================

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

SELECT o.RestaurantID, SUM(o.OrderAmount) AS TotalRevenue

FROM Orders o

JOIN Payments p ON o.OrderID = p.OrderID

WHERE o.OrderStatus = 'Delivered'

GROUP BY o.RestaurantID;

================================================================================================

5. Retrieve the most recent order placed by each customer.

SELECT o.CustomerID, o.OrderID, o.OrderDate, o.OrderAmount

FROM Orders o

JOIN (

SELECT CustomerID, MAX(OrderDate) AS MaxOrderDate

FROM Orders

GROUP BY CustomerID

) AS latest\_order ON o.CustomerID = latest\_order.CustomerID

AND o.OrderDate = latest\_order.MaxOrderDate;

=============================================================================================

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

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

FROM Customers c

LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderID IS NULL;

===============================================================================================

7. Identify the most reviewed restaurants.

SELECT r.RestaurantID, COUNT(r.ReviewID) AS ReviewCount

FROM Reviews r

GROUP BY r.RestaurantID

ORDER BY ReviewCount DESC;

=============================================================================================

8. Find the most preferred payment method.

SELECT p.PaymentMethod, COUNT(p.PaymentMethod) AS MethodCount

FROM Payments p

GROUP BY p.PaymentMethod

ORDER BY MethodCount DESC

LIMIT 1;

===========================================================================================

9. List the top 5 restaurants by total revenue.

SELECT o.RestaurantID, SUM(o.OrderAmount) AS TotalRevenue

FROM Orders o

JOIN Payments p ON o.OrderID = p.OrderID

GROUP BY o.RestaurantID

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, o.OrderStatus,

c.FirstName AS CustomerFirstName, c.LastName AS CustomerLastName, c.Phone AS CustomerPhone, c.City AS CustomerCity,

r.Name AS RestaurantName, r.City AS RestaurantCity

FROM Orders o

JOIN Customers c ON o.CustomerID = c.CustomerID

JOIN Restaurants r ON o.RestaurantID = r.RestaurantID

WHERE o.OrderStatus = 'Cancelled';