Internship Task - RDBMS and SQL Task #5

(1) Create Database

Query=>

CREATE DATABASE Zomato;

(2)Use Database

Query=>

use Zomato;

(3)create table Restaurants

Query=>

CREATE TABLE Restaurants(

RestaurantID INT Primary Key,

Name VARCHAR(255) NOT NULL,

City VARCHAR(100) NOT NULL,

CuisineType VARCHAR(100) NOT NULL,

Rating DECIMAL(2,1) CHECK (RATING >= 0 AND RATING <= 5),

AverageCostForTwo DECIMAL(10,2)

);

(4)create table Customers

Query=>

CREATE TABLE Customers

(CustomerID INT Primary Key,

FirstName VARCHAR(255) NOT NULL,

LastName VARCHAR(255) NOT NULL,

Phone INT(10) NOT NULL,

City VARCHAR(100) NOT NULL,

JoinDate DATE

);

(5)create table Orders

Query=>

CREATE TABLE Orders

(OrderID INT Primary Key,

CustomerID INT,

Foreign Key (CustomerID)REFERENCES Customers(CustomerID),

RestaurantID INT,

Foreign Key(RestaurantID) REFERENCES Restaurants(RestaurantID),

OrderDate DATE,

OrderAmount DECIMAL(10,2),

OrderStatus VARCHAR(50)

);

(6)create table Reviews

Query=>

CREATE TABLE Reviews(

ReviewID INT Primary Key,

CustomerID INT,

Foreign Key(CustomerID) REFERENCES Customers(CustomerID),

RestaurantID INT,

Foreign Key (RestaurantID) REFERENCES Restaurants(RestaurantID),

Rating INT CHECK (RATING >= 1 AND RATING <= 5),

Comment TEXT,

ReviewDate DATE

);

(7)create table Payments

Query=>

CREATE TABLE Payments(

PaymentID INT Primary Key,

OrderID INT,

Foreign Key(OrderID) REFERENCES Orders(OrderID),

PaymentMethod VARCHAR(50),

Amount DECIMAL( 10,2),

PaymentDate DATE

);

(8)Insert data in table Restaurants

Query=>

INSERT INTO Restaurants

VALUES

(1, 'The Gourmet Kitchen', 'bhopal', 'Italian', 4.5, 45.00),

(2, 'Sushi Paradise', 'indore', 'Japanese', 4.8, 60.00),

(3, 'Spice & Sizzle', 'ujjain', 'Indian', 4.2, 30.00),

(4, 'punjabi daba', 'dewas', 'Indian', 4.6, 50.00),

(5, 'Burger Town', 'indore','Mexican', 3.9, 25.00),

(6, 'Veggie Delight', 'pune', 'Vegetarian', 4.7, 35.00),

(7, 'Taco Haven', 'bombay', 'Mexican', 4.3, 20.00),

(8, 'Curry Corner', 'indore', 'Indian', 4.0, 40.00),

(9, 'Pasta Bella', 'Bhopal', 'Italian', 4.4, 55.00),

(10, 'Seafood Shack', 'indore', 'Seafood', 4.9, 70.00);

(9)Insert data in table Customers

Query=>

INSERT INTO Customers

VALUES

(101, 'Jiya', 'Das', '8978674567', 'bhopal', '2022-01-10'),

(102, 'swati', 'Sen', '8976354567', 'ujjain', '2021-08-25'),

(103, 'neha', 'solanki','9087567848', 'indore', '2020-11-05'),

(104, 'aman', 'sen', '7890567890', 'bhopal', '2022-02-18'),

(105, 'adarsh', 'sharma', '9856784567', 'pune', '2021-12-30'),

(106, 'Seema', 'verma', '7654567898', 'indore', '2023-03-15'),

(107, 'yogendra', 'roy', '9767898678', 'Dewas', '2020-07-22'),

(108, 'Ojas', 'kulkarni', '9078456787', 'ujjain', '2022-09-10'),

(109, 'Danish', 'khan', '7689567856', 'bhopal', '2021-10-11'),

(110, 'Asha', 'rajput','7575653456', 'agra', '2023-04-25');

(10)Insert data in table Orders

Query=>

INSERT INTO Orders

VALUES

(201, 101,1, '2023-01-15', 45.50, 'Delivered'),

(202, 102,2, '2023-02-10', 60.00, 'Pending'),

(203, 103,3, '2023-03-05', 35.00, 'Shipped'),

(204, 104,4, '2023-04-20', 25.00, 'Delivered'),

(205, 105,5, '2023-05-25', 40.00, 'Shipped'),

(206, 106,6, '2023-06-10', 55.00, 'Delivered'),

(207, 107,7, '2023-07-01', 70.00, 'Cancelled'),

(208, 108,8, '2023-08-12', 50.00, 'Pending'),

(209, 109,9, '2023-09-17', 30.00, 'Shipped'),

(210, 110,10, '2023-10-01', 60.00, 'Delivered');

(11)Insert data in table Reviews

Query=>

INSERT INTO Reviews

VALUES

(301, 101, 1, 5, 'Great sushi and excellent service!', '2023-01-16'),

(302, 102, 2, 4, 'Good food, but the wait time was a bit long.', '2023-02-12'),

(303, 103, 3, 3, 'The food was decent, but not amazing.', '2023-03-06'),

(304, 104, 4, 4, 'Tasty burgers, friendly staff, would visit again.', '2023-04-22'),

(305, 105, 5, 2, 'Disappointed with the quality of the food.', '2023-05-26'),

(306, 106, 6, 5, 'Amazing seafood, will definitely come back!', '2023-06-12'),

(307, 107, 7, 1, 'Extremely poor service, will never come back.', '2023-07-02'),

(308, 108, 8, 4, 'Pasta was delicious, but the restaurant was a bit noisy.', '2023-08-14'),

(309, 109, 9, 3, 'Average food, nothing special.', '2023-09-19'),

(310, 110, 10, 5, 'Loved the vegetarian options, very tasty!', '2023-10-03');

(12)Insert data in table Payments

Query=>

INSERT INTO Payments

VALUES

(1, 201, 'Credit Card', 45.50, '2023-01-16'),

(2, 202, 'PayPal', 60.00, '2023-02-12'),

(3, 203, 'Debit Card', 35.00, '2023-03-07'),

(4, 204, 'Credit Card', 25.00, '2023-04-23'),

(5, 205, 'Cash', 40.00, '2023-05-26'),

(6, 206, 'PayPal', 55.00, '2023-06-13'),

(7, 207, 'Debit Card', 70.00, '2023-07-03'),

(8, 208, 'Credit Card', 50.00, '2023-08-15'),

(9, 209, 'Cash', 30.00, '2023-09-20'),

(10, 210, 'PayPal', 60.00, '2023-10-04');

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

Assignment Queries

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

Query=>

SELECT Name, City

FROM Restaurants

WHERE Rating >= 4.5;

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

Query=>

SELECT CustomerID, COUNT(OrderID) AS TotalOrders

FROM Orders

GROUP BY CustomerID;

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

Query=>

SELECT Name, City

FROM Restaurants

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

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

Query=>

SELECT r.Name, SUM(o.OrderAmount) AS TotalRevenue

FROM Orders o

JOIN Restaurants r ON o.RestaurantID = r.RestaurantID

WHERE o.OrderStatus = 'Delivered'

GROUP BY r.Name;

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

Query=>

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

FROM Orders o

INNER JOIN (

SELECT CustomerID, MAX(OrderDate) AS LatestOrderDate

FROM Orders

GROUP BY CustomerID

) latest ON o.CustomerID = latest.CustomerID AND o.OrderDate = latest.LatestOrderDate;

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

Query=>

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

FROM Customers c

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

WHERE o.CustomerID IS NULL;

7. Identify the most reviewed restaurants.

Query=>

SELECT r.Name, COUNT(rev.ReviewID) AS ReviewCount

FROM Reviews rev

JOIN Restaurants r ON rev.RestaurantID = r.RestaurantID

GROUP BY r.Name

ORDER BY ReviewCount DESC

LIMIT 1;

8. Find the most preferred payment method.

Query=>

SELECT PaymentMethod, COUNT(PaymentID) AS PaymentCount

FROM Payments

GROUP BY PaymentMethod

ORDER BY PaymentCount DESC

LIMIT 1;

9. List the top 5 restaurants by total revenue.

Query=>

SELECT r.Name, SUM(p.Amount) AS TotalRevenue

FROM Payments p

JOIN Orders o ON p.OrderID = o.OrderID

JOIN Restaurants r ON o.RestaurantID = r.RestaurantID

GROUP BY 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

Query=>

SELECT o.OrderID, o.OrderDate, o.OrderAmount, o.OrderStatus, c.FirstName, c.LastName, r.Name AS RestaurantName

FROM Orders o

JOIN Customers c ON o.CustomerID = c.CustomerID

JOIN Restaurants r ON o.RestaurantID = r.RestaurantID

WHERE o.OrderStatus = 'Cancelled';

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