**Internship Task - RDBMS and SQL Task #5**

**SQL case-based assignment with a scenario and 10 queries for a food booking application like Zomato. Scenario :You are managing a database for a food delivery application like Zomato. The database includes the following tables:**

create DATABASE zomato;

CREATE TABLE Restaurants (

RestaurantID INT PRIMARY KEY ,

Name VARCHAR(100) ,

City VARCHAR(50),

CuisineType VARCHAR(50) ,

Rating DECIMAL(2, 1) CHECK (Rating BETWEEN 0 AND 5),

AverageCostForTwo DECIMAL(10, 2)

);

INSERT INTO Restaurants VALUES

(1, 'Taj Hotel', 'Mumbai', 'Indian', 4.5, 2000.00),

(2, 'Punjab Grill', 'Delhi', 'Punjabi', 4.2, 1500.00),

(3, 'Paradise Biryani', 'Hyderabad', 'Biryani', 4.8, 1000.00),

(4, 'Barbeque Nation', 'Bangalore', 'Barbecue', 4.3, 1800.00),

(5, 'Anjappar Chettinad', 'Chennai', 'Chettinad', 4.1, 1200.00),

(6, 'Oh! Calcutta', 'Kolkata', 'Bengali', 4.6, 1400.00),

(7, 'Banana Leaf', 'Mumbai', 'South Indian', 4.0, 800.00),

(8, 'Bikanervala', 'Delhi', 'Rajasthani', 4.3, 900.00),

(9, 'Dakshin', 'Chennai', 'South Indian', 4.4, 1500.00),

(10, 'Indian Accent', 'Delhi', 'Fusion', 4.9, 2500.00),

(11, 'Saravana Bhavan', 'Chennai', 'South Indian', 4.0, 500.00),

(12, 'Cafe Madras', 'Mumbai', 'South Indian', 4.7, 700.00),

(13, 'Peshawri', 'Mumbai', 'North Indian', 4.5, 2200.00),

(14, 'Trishna', 'Mumbai', 'Seafood', 4.6, 2700.00),

(15, 'Karim’s', 'Delhi', 'Mughlai', 4.3, 1200.00);

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY ,

FirstName VARCHAR(50) ,

LastName VARCHAR(50) ,

Phone VARCHAR(15) ,

City VARCHAR(50) ,

JoinDate DATE

);

INSERT INTO Customers VALUES

(1, 'Amit', 'Sharma', '9876543210', 'Mumbai', '2023-01-15'),

(2, 'Priya', 'Singh', '8765432109', 'Delhi', '2023-02-20'),

(3, 'Rahul', 'Verma', '7654321098', 'Hyderabad', '2023-03-10'),

(4, 'Sneha', 'Kumar', '6543210987', 'Bangalore', '2023-04-25'),

(5, 'Ravi', 'Reddy', '5432109876', 'Chennai', '2023-05-18'),

(6, 'Sunita', 'Mishra', '4321098765', 'Kolkata', '2023-06-12'),

(7, 'Vikram', 'Das', '3210987654', 'Mumbai', '2023-07-08'),

(8, 'Pooja', 'Iyer', '2109876543', 'Delhi', '2023-08-05'),

(9, 'Kiran', 'Rao', '1098765432', 'Chennai', '2023-09-01'),

(10, 'Anjali', 'Mehta', '1987654321', 'Hyderabad', '2023-09-15'),

(11, 'Rohit', 'Kapoor', '9871234560', 'Mumbai', '2023-01-05'),

(12, 'Suman', 'Joshi', '8762345671', 'Delhi', '2023-02-18'),

(13, 'Manoj', 'Naidu', '7653456782', 'Bangalore', '2023-03-21'),

(14, 'Deepa', 'Pillai', '6544567893', 'Kolkata', '2023-04-11'),

(15, 'Raj', 'Chauhan', '5435678904', 'Chennai', '2023-05-24');

CREATE TABLE Orders (

OrderID INT PRIMARY KEY ,

CustomerID INT ,

RestaurantID INT ,

OrderDate DATETIME ,

OrderAmount DECIMAL(10, 2) ,

OrderStatus varchar(100),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)

);

INSERT INTO Orders VALUES

(1, 1, 1, '2023-11-01 12:30:00', 1500.00, 'Completed'),

(2, 2, 2, '2023-11-05 14:00:00', 1800.00, 'Completed'),

(3, 3, 3, '2023-11-10 19:00:00', 1200.00, 'Cancelled'),

(4, 4, 4, '2023-11-15 20:00:00', 2500.00, 'Completed'),

(5, 5, 5, '2023-11-20 13:00:00', 900.00, 'Pending'),

(6, 6, 6, '2023-11-25 18:00:00', 1700.00, 'Completed'),

(7, 7, 7, '2023-11-28 15:00:00', 800.00, 'Completed'),

(8, 8, 8, '2023-12-01 21:00:00', 1400.00, 'Completed'),

(9, 9, 9, '2023-12-05 12:30:00', 1200.00, 'Completed'),

(10, 10, 10, '2023-12-07 20:00:00', 3000.00, 'Completed'),

(11, 11, 11, '2023-12-10 11:00:00', 500.00, 'Completed'),

(12, 12, 12, '2023-12-12 19:30:00', 700.00, 'Pending'),

(13, 13, 13, '2023-12-15 18:45:00', 2200.00, 'Completed'),

(14, 14, 14, '2023-12-18 13:15:00', 2700.00, 'Completed'),

(15, 15, 15, '2023-12-20 20:00:00', 1200.00, 'Cancelled');

CREATE TABLE Reviews (

ReviewID INT PRIMARY KEY ,

CustomerID INT ,

RestaurantID INT ,

Rating DECIMAL(2, 1) CHECK (Rating BETWEEN 0 AND 5),

Comment varchar(100),

ReviewDate DATETIME ,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID)

);

INSERT INTO Reviews VALUES

(1, 1, 1, 4.5, 'Excellent service', '2023-11-02 13:00:00'),

(2, 2, 2, 4.2, 'Great ambiance', '2023-11-06 15:00:00'),

(3, 3, 3, 4.8, 'Best biryani ever', '2023-11-11 20:00:00'),

(4, 4, 4, 4.3, 'Good barbecue', '2023-11-16 21:00:00'),

(5, 5, 5, 4.1, 'Authentic flavors', '2023-11-21 14:00:00'),

(6, 6, 6, 4.6, 'Amazing experience', '2023-11-26 19:00:00'),

(7, 7, 7, 4.0, 'Affordable and tasty', '2023-11-29 16:00:00'),

(8, 8, 8, 4.3, 'Delicious sweets', '2023-12-02 22:00:00'),

(9, 9, 9, 4.4, 'Authentic dishes', '2023-12-06 13:00:00'),

(10, 10, 10, 4.9, 'Excellent fusion', '2023-12-08 21:00:00'),

(11, 11, 11, 4.0, 'Great value', '2023-12-11 12:00:00'),

(12, 12, 12, 4.7, 'Classic and tasty', '2023-12-13 20:00:00'),

(13, 13, 13, 4.5, 'Excellent quality', '2023-12-16 19:00:00'),

(14, 14, 14, 4.6, 'Amazing seafood', '2023-12-19 14:00:00'),

(15, 15, 15, 4.3, 'Delicious kebabs', '2023-12-21 21:00:00');

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY,

OrderID INT ,

PaymentMethod varchar(100),

Amount DECIMAL(10, 2) ,

PaymentDate DATETIME ,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

);

INSERT INTO Payments VALUES

(1, 1, 'Credit Card', 1500.00, '2023-11-01 12:45:00'),

(2, 2, 'UPI', 1800.00, '2023-11-05 14:15:00'),

(3, 3, 'Cash', 1200.00, '2023-11-10 19:15:00'),

(4, 4, 'Debit Card', 2500.00, '2023-11-15 20:15:00'),

(5, 5, 'UPI', 900.00, '2023-11-20 13:15:00'),

(6, 6, 'Credit Card', 1700.00, '2023-11-25 18:15:00'),

(7, 7, 'Cash', 800.00, '2023-11-28 15:15:00'),

(8, 8, 'UPI', 1400.00, '2023-12-01 21:15:00'),

(9, 9, 'Credit Card', 1200.00, '2023-12-05 12:45:00'),

(10, 10, 'UPI', 3000.00, '2023-12-07 20:15:00'),

(11, 11, 'Cash', 500.00, '2023-12-10 11:15:00'),

(12, 12, 'Debit Card', 700.00, '2023-12-12 19:45:00'),

(13, 13, 'Credit Card', 2200.00, '2023-12-15 19:00:00'),

(14, 14, 'UPI', 2700.00, '2023-12-18 13:45:00'),

(15, 15, 'Cash', 1200.00, '2023-12-20 20:15:00');

-------------------------------------------------------------------

**#Assignment Queries#**

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

**Ans=**

SELECT Name, City from restaurants where Rating>=4.5

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

**Ans=**

SELECT c.CustomerID, c.FirstName, c.LastName, COUNT(o.OrderID)as totalnoOfOrders FROM customers as c left join orders as o on c.CustomerID=o.CustomerID GROUP by c.CustomerID

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

**Ans=**

SELECT RestaurantID FROM restaurants where CuisineType='South Indian' && City='Mumbai' GROUP by RestaurantID

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

**Ans=**

SELECT r.RestaurantID ,r.Name, sum(o.OrderAmount)as totalrevenu from restaurants as r left join orders as o on r.RestaurantID=o.RestaurantID WHERE OrderStatus='Completed' GROUP by r.RestaurantId

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

**Ans=**

SELECT c.CustomerID , c.FirstName, c.LastName, max(o.OrderDate) FROM customers as c left join orders as o on c.CustomerID=o.CustomerID GROUP by c.CustomerID

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

**Ans=**

SELECT c.CustomerID , c.FirstName, c.LastName FROM customers as c left join orders as o on c.CustomerID=o.CustomerID WHERE OrderId is null GROUP by c.CustomerID

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

**Ans=**

SELECT r.Name,COUNT(r1.ReviewID) AS total\_reviews

FROM restaurants r

JOIN reviews r1 ON r.RestaurantID = r1.RestaurantID

GROUP BY r.RestaurantID, r.Name

ORDER BY total\_reviews DESC;

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

**Ans=**

SELECT PaymentMethod, count(PaymentMethod)as most\_prefered\_payment\_method from payments GROUP by PaymentMethod order by most\_prefered\_payment\_method DESC limit 1;

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

**Ans=**

SELECT sum(o.OrderAmount)as total\_revenue ,r.Name from restaurants as r join orders as o on r.RestaurantID=o.RestaurantID GROUP by Name order by total\_revenue DESC limit 5 ;

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

**Ans=**

SELECT o.OrderID , o.OrderDate, o.OrderAmount ,o.OrderStatus ,c.FirstName, c.LastName , r.Name from orders as o join restaurants as r on o.RestaurantID=r.RestaurantID join customers as c on o.CustomerID=c.CustomerID where o.OrderStatus ='Cancelled' GROUP by o.OrderID