**SQL Coding Challege**

**Topic- PetPals**

**Tasks:**

1.Write update query to change the donation amount Increase 50% if it is less than 1000.

UPDATE Donations

SET DonationAmount = DonationAmount \* 1.5

WHERE DonationAmount < 5000;

2. Create tables for pets, shelters, donations, adoption events, and participants.

3. Define appropriate primary keys, foreign keys, and constraints.

DROP DATABASE IF EXISTS PetPals;

CREATE DATABASE PetPals;

USE PetPals;

CREATE TABLE Pets (

PetID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

Age INT NOT NULL,

Breed VARCHAR(50),

Type VARCHAR(50) NOT NULL,

AvailableForAdoption BIT NOT NULL

);

CREATE TABLE Shelters (

ShelterID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

Location VARCHAR(200) NOT NULL

);

CREATE TABLE Donations (

DonationID INT PRIMARY KEY,

DonorName VARCHAR(50) NOT NULL,

DonationType VARCHAR(50) NOT NULL,

DonationAmount DECIMAL(10, 2),

DonationItem VARCHAR(50),

DonationDate DATETIME NOT NULL

);

CREATE TABLE AdoptionEvents (

EventID INT PRIMARY KEY,

EventName VARCHAR(100) NOT NULL,

EventDate DATETIME NOT NULL,

Location VARCHAR(200) NOT NULL

);

CREATE TABLE Participants (

ParticipantID INT PRIMARY KEY,

ParticipantName VARCHAR(50) NOT NULL,

ParticipantType VARCHAR(50) NOT NULL,

EventID INT,

CONSTRAINT fk\_event FOREIGN KEY (EventID)

REFERENCES AdoptionEvents(EventID)

);

INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption)

VALUES

(1, 'Moti', 2, 'Labrador', 'Dog', 1),

(2, 'Sheru', 3, 'German Shepherd', 'Dog', 0),

(3, 'Whiskers', 1, 'Persian', 'Cat', 1),

(4, 'Tommy', 4, 'Pug', 'Dog', 1),

(5, 'Billi', 5, 'Siamese', 'Cat', 0);

INSERT INTO Shelters (ShelterID, Name, Location)

VALUES

(101, 'Paws of India', 'Delhi, India'),

(102, 'Animal Care Trust', 'Mumbai, Maharashtra'),

(103, 'Karuna Animal Shelter', 'Bangalore, Karnataka'),

(104, 'Love and Care Shelter', 'Chennai, Tamil Nadu'),

(105, 'Helping Paws', 'Kolkata, West Bengal');

INSERT INTO Donations (DonationID, DonorName, DonationType, DonationAmount, DonationItem, DonationDate)

VALUES

(1, 'Rajesh Kumar', 'Cash', 5000.00, NULL, '2024-09-01 10:30:00'),

(2, 'Sunita Sharma', 'Item', NULL, 'Dog Food', '2024-09-03 12:45:00'),

(3, 'Anil Verma', 'Cash', 3000.00, NULL, '2024-09-05 09:15:00'),

(4, 'Pooja Jain', 'Item', NULL, 'Cat Toys', '2024-09-07 11:20:00'),

(5, 'Amit Singh', 'Cash', 2000.00, NULL, '2024-09-10 14:00:00');

INSERT INTO AdoptionEvents (EventID, EventName, EventDate, Location)

VALUES

(1, 'Paws Adoption Drive', '2024-10-01 09:00:00', 'Delhi, India'),

(2, 'Love and Care Pet Fair', '2024-10-10 11:00:00', 'Chennai, Tamil Nadu'),

(3, 'Karuna Adoption Camp', '2024-11-05 10:00:00', 'Bangalore, Karnataka'),

(4, 'Animal Care Trust Pet Meet', '2024-10-15 12:00:00', 'Mumbai, Maharashtra'),

(5, 'Helping Paws Day Out', '2024-12-01 09:30:00', 'Kolkata, West Bengal');

INSERT INTO Participants (ParticipantID, ParticipantName, ParticipantType, EventID)

VALUES

(1, 'Paws of India', 'Shelter', 1),

(2, 'Love and Care Shelter', 'Shelter', 2),

(3, 'Karuna Animal Shelter', 'Shelter', 3),

(4, 'Rajesh Kumar', 'Adopter', 1),

(5, 'Sunita Sharma', 'Adopter', 2);

4. Delete the pet whose age is greater than equal to 5.

DELETE FROM Pets

WHERE Age >= 5;

5. Write an SQL query that retrieves a list of available pets (those marked as available for adoption) from the "Pets" table. Include the pet's name, age, breed, and type in the result set. Ensure that the query filters out pets that are not available for adoption.

SELECT Name, Age, Breed, Type

FROM Pets

WHERE AvailableForAdoption = 1;

6. Write an SQL query that retrieves the names of participants (shelters and adopters) registered for a specific adoption event. Use a parameter to specify the event ID. Ensure that the query joins the necessary tables to retrieve the participant names and types.

SELECT p.ParticipantName, p.ParticipantType

FROM Participants p

JOIN AdoptionEvents e ON p.EventID = e.EventID

WHERE e.EventID = 1;

7. Show details of pets along with AdoptionEvents-all columns and Participants-all columns. Display pet\_id, Pet\_name, breed and type.

SELECT

p.PetID AS PetID,

p.Name AS PetName,

p.Breed,

p.Type,

a.EventID AS EventID,

a.EventName AS EventName,

a.EventDate AS EventDate,

a.Location AS EventLocation,

par.ParticipantID AS ParticipantID,

par.ParticipantName AS ParticipantName,

par.ParticipantType AS ParticipantType

FROM

Pets p

LEFT JOIN

Participants par ON par.EventID IN (

SELECT EventID FROM AdoptionEvents WHERE EventID = par.EventID

)

LEFT JOIN

AdoptionEvents a ON a.EventID = par.EventID;

8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.

SELECT

s.Name AS ShelterName,

COALESCE(SUM(d.DonationAmount), 0) AS TotalDonationAmount

FROM Shelters s

LEFT JOIN Donations d ON s.Name = d.DonorName

GROUP BY s.Name;

9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "Participant\_id" is null). Include the pet's name, age, breed, and type in the result set.

SELECT

p.Name AS PetName,

p.Age,

p.Breed,

p.Type

FROM Pets p

LEFT JOIN Participants par ON par.ParticipantID = p.PetID

WHERE par.ParticipantID IS NULL;

10. Write an SQL query that retrieves the total donation amount for each month and year (e.g., January 2023) from the "Donations" table. The result should include the month-year and the corresponding total donation amount. Ensure that the query handles cases where no donations were made in a specific month-year.

SELECT

DATE\_FORMAT(DonationDate, '%M %Y') AS MonthYear,

COALESCE(SUM(DonationAmount), 0) AS TotalDonationAmount

FROM Donations

GROUP BY MonthYear

ORDER BY MIN(DonationDate);

11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

SELECT DISTINCT Breed

FROM Pets

WHERE Age BETWEEN 1 AND 3 OR Age > 5;

12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

SELECT

p.Name AS PetName,

s.Name AS ShelterName

FROM Pets p

JOIN Participants par ON par.EventID IN (SELECT EventID FROM AdoptionEvents)

JOIN Shelters s ON s.Name = par.ParticipantName

WHERE p.AvailableForAdoption = 1;

13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

SELECT COUNT(p.ParticipantID) AS TotalParticipants

FROM Participants p

JOIN AdoptionEvents e ON p.EventID = e.EventID

JOIN Shelters s ON s.Name = p.ParticipantName

WHERE s.Location LIKE '%Chennai%';

14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

SELECT DISTINCT Breed

FROM Pets

WHERE Age BETWEEN 1 AND 5;

15. Find the pets that have not been adopted by selecting their information from the 'Pet' table.

SELECT \* FROM Pets

WHERE AvailableForAdoption = 1;

16. Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'User' tables.

SELECT

p.Name AS AdoptedPetName,

par.ParticipantName AS AdopterName

FROM Pets p

JOIN Participants par ON p.Name = par.ParticipantName

WHERE par.ParticipantType = 'Adopter';

17. Retrieve a list of all shelters along with the count of pets currently available for adoption in each shelter.

SELECT

s.Name AS ShelterName,

COUNT(p.PetID) AS AvailablePetsCount

FROM Shelters s

LEFT JOIN Pets p ON p.AvailableForAdoption = 1

GROUP BY s.Name;

18. Find pairs of pets from the same shelter that have the same breed.

SELECT

p1.Name AS Pet1,

p2.Name AS Pet2,

p1.Breed

FROM Pets p1

JOIN Pets p2 ON p1.Breed = p2.Breed AND p1.PetID <> p2.PetID;

19. List all possible combinations of shelters and adoption events.

SELECT

s.Name AS ShelterName,

e.EventName

FROM Shelters s

CROSS JOIN AdoptionEvents e;

20. Determine the shelter that has the highest number of adopted pets.

SELECT

s.Name AS ShelterName,

COUNT(p.PetID) AS AdoptedPetsCount

FROM Shelters s

JOIN Pets p ON p.AvailableForAdoption = 0 -- Assuming not available means adopted

GROUP BY s.Name

ORDER BY AdoptedPetsCount DESC

LIMIT 1;