**SQL Assessment:**

1. **Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals”.**

Create Database PetPals;

Use PetPals;

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

**PETS:**

CREATE TABLE Pets (

PetID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(15) NOT NULL,

Age INT NOT NULL,

Breed VARCHAR(20) NOT NULL,

Type VARCHAR(50) NOT NULL,

AvailableforAdoption BIT NOT NULL

);

**SHELTERS:**

CREATE TABLE Shelters (

ShelterID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Location VARCHAR(255) NOT NULL

FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID);

);

**DONATIONS:**

CREATE TABLE Donations (

DonationID INT PRIMARY KEY AUTO\_INCREMENT,

DonorName VARCHAR(100) NOT NULL,

Donationtype VARCHAR(50) NOT NULL,

DonationAmount DECIMAL(10,2) NULL,

DonationItem VARCHAR(100) NULL,

DonationDate DATETIME NOT NULL

);

**ADOPTIONEVENTS:**

CREATE TABLE AdoptionEvents (

EventID INT PRIMARY KEY AUTO\_INCREMENT,

EventName VARCHAR(100) NOT NULL,

EventDate DATETIME NOT NULL,

Location VARCHAR(100) NOT NULL

);

**PARTICIPANTS:**

CREATE TABLE Participants (

ParticipantID INT PRIMARY KEY AUTO\_INCREMENT,

ParticipantName VARCHAR(100) NOT NULL,

ParticipantType VARCHAR(50) NOT NULL,

EventID INT,

FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)

);

1. **Define appropriate primary keys, foreign keys, and constraints.**

**PRIMARY KEYS:**

PetID, ShelterID , DonationID, EventID, ParticipantID

**FOREIGN KEYS:**

EventID, ShelterID

FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)

ALTER TABLE Pets ADD FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID);

**CONSTRAINTS:**

NOT NULL, AUTO-INCREMENT, BIT

1. Ensure the script handles potential errors, such as if the database or tables already exist

Add if exists or if not exists to the table

1. SELECT Name, Age, Breed, Type FROM Pets WHERE AvailableForAdoption = 1;
2. SELECT ParticipantName, ParticipantType

FROM Participants

WHERE EventID = 1;

1. **------**
2. SELECT s.Name, COALESCE(SUM(d.DonationAmount), 0) AS TotalDonation

FROM Shelters s

LEFT JOIN Donations d ON s.ShelterID = d.ShelterID

GROUP BY s.Name;

1. SELECT Name, Age, Breed, Type FROM Pets WHERE OwnerID IS NULL;
2. SELECT DATE\_FORMAT(DonationDate, '%Y-%m') AS MonthYear,

SUM(DonationAmount) AS TotalDonation

FROM Donations

GROUP BY MonthYear;

1. SELECT DISTINCT Breed FROM Pets WHERE Age BETWEEN 1 AND 3 OR Age > 5;
2. SELECT p.Name, p.Breed, s.Name AS ShelterName

FROM Pets p

JOIN Shelters s ON p.ShelterID = s.ShelterID

WHERE p.AvailableForAdoption = 1;

**13**.SELECT COUNT(\*) AS TotalParticipants

FROM Participants p

JOIN AdoptionEvents ae ON p.EventID = ae.EventID

WHERE ae.Location = 'Chennai';

**14**.SELECT DISTINCT Breed FROM Pets WHERE Age BETWEEN 1 AND 5;

**15**. SELECT \*FROM Pets

WHERE OwnerID IS NULL;

**16.** SELECT p.Name AS PetName, a.AdopterName

FROM Pets p

JOIN Adoption a ON p.PetID = a.PetID;

**17.-------**

**18.** SELECT s.Name, COUNT(p.PetID) AS AvailablePets

FROM Shelters s

JOIN Pets p ON p.AvailableForAdoption = 1

GROUP BY s.Name;

**19.** SELECT s.Name AS ShelterName, ae.EventName

FROM Shelters s

CROSS JOIN AdoptionEvents ae;

**20**. SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AdoptedPets

FROM Shelters s

JOIN Pets p ON p.OwnerID IS NOT NULL AND p.ShelterID = s.ShelterID

GROUP BY s.Name

ORDER BY AdoptedPets DESC LIMIT 1;