## Coding Challenges - PetPals, The Pet Adoption Platform

#### Student name: Abhishek Arvind

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

```
CREATE DATABASE PetPals;
```

- 2. Create tables for pets, shelters, donations, adoption events, and participants.
- 3. Define appropriate primary keys, foreign keys, and constraints.
- 4. Ensure the script handles potential errors, such as if the database or tables already exist.

```
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'PetPals')
BEGIN
    CREATE DATABASE PetPals;
ELSE
BEGIN
    PRINT 'Database already exists.';
use PetPals
Messages
   Database already exists.
   Completion time: 2024-03-04T14:00:38.3992228+05:30
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Shelters')
BEGIN
   CREATE TABLE Shelters (
        ShelterID INT PRIMARY KEY,
        Name VARCHAR(255),
        Location VARCHAR(255),
        City VARCHAR(255)
    );
   PRINT 'Table "Shelters" created successfully.';
END
FLSE
BEGIN
    PRINT 'Table "Shelters" already exists.';
END
```

```
    Messages

  Table "Shelters" created successfully.
  Completion time: 2024-03-04T14:03:10.0070557+05:30
Messages
  Table "Shelters" already exists.
   Completion time: 2024-03-04T14:03:33.9721521+05:30
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Donations')
BEGIN
   CREATE TABLE Donations (
        DonationID INT PRIMARY KEY,
        DonorName VARCHAR(255),
        DonationType VARCHAR(255),
        DonationAmount DECIMAL,
        DonationItem VARCHAR(255),
        DonationDate DATETIME,
        ShelterID INT,
        FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID)
    );
    PRINT 'Table "Donations" created successfully.';
END
ELSE
BEGIN
    PRINT 'Table "Donations" already exists.';
END

    Messages

   Table "Donations" created successfully.
   Completion time: 2024-03-04T14:16:17.8334847+05:30
Messages
  Table "Donations" already exists.
  Completion time: 2024-03-04T14:16:42.2113451+05:30
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'AdoptionEvents')
BEGIN
   CREATE TABLE AdoptionEvents (
        EventID INT PRIMARY KEY,
        EventName VARCHAR(255),
        EventDate DATETIME,
        Location VARCHAR(255),
        City VARCHAR(255),
        OrganizerID INT,
        FOREIGN KEY (OrganizerID) REFERENCES Shelters(ShelterID)
    );
    PRINT 'Table "AdoptionEvents" created successfully.';
END
ELSE
BEGIN
```

```
PRINT 'Table "AdoptionEvents" already exists.';
END
Messages
   Table "AdoptionEvents" created successfully.
   Completion time: 2024-03-04T14:04:49.3976770+05:30
Messages
  Table "AdoptionEvents" already exists.
  Completion time: 2024-03-04T14:05:10.2468676+05:30
IF NOT EXISTS (SELECT * FROM INFORMATION SCHEMA.TABLES WHERE TABLE NAME = 'Participants')
BEGIN
    CREATE TABLE Participants (
        ParticipantID INT PRIMARY KEY,
        ParticipantName VARCHAR(255),
        ParticipantType VARCHAR(255),
        EventID INT,
        FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID),
        City VARCHAR(255)
    );
    PRINT 'Table "Participants" created successfully.';
END
ELSE
BEGIN
    PRINT 'Table "Participants" already exists.';
FND
Messages
   Table "Participants" created successfully.
   Completion time: 2024-03-04T14:06:56.5215715+05:30

    Messages

   Table "Participants" already exists.
   Completion time: 2024-03-04T14:15:08.2391910+05:30
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Pets')
BEGIN
    CREATE TABLE Pets (
        PetID INT PRIMARY KEY,
        Name VARCHAR(255),
        Age INT,
        Breed VARCHAR(255),
        Type VARCHAR(255),
        AvailableForAdoption BIT,
        ShelterName VARCHAR(255),
        OwnerID INT,
        ShelterID INT
        FOREIGN KEY (OwnerID) REFERENCES Participants(ParticipantID),
        FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID)
    );
    PRINT 'Table "Pets" created successfully.';
END
ELSE
BEGIN
    PRINT 'Table "Pets" already exists.';
END
```

```
Table "Pets" already exists.

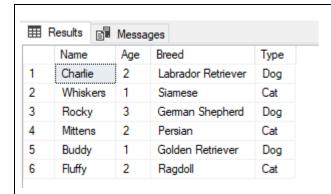
Completion time: 2024-03-04T14:00:01.7085966+05:30

Messages
Table "Pets" created successfully.

Completion time: 2024-03-04T13:59:14.8455525+05:30
```

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.

```
INSERT INTO Shelters (ShelterID, Name, Location, City)
VALUES(1, 'Chennai Pet Shelter', 'Anna Nagar', 'Chennai'),
(2, 'Coimbatore Animal Care', 'Gandhipuram', 'Coimbatore'),
(3, 'Madurai Paws Haven', 'Kochadai', 'Madurai'),
(4, 'Trichy Furry Friends', 'Thillai Nagar', 'Trichy'),
(5, 'Salem Animal Sanctuary', 'Shevapet', 'Salem'),
(6, 'Vellore Pet Haven', 'Gandhi Road', 'Vellore');
INSERT INTO Donations (DonationID, DonorName, DonationType, DonationAmount, DonationItem,
DonationDate, ShelterID)
VALUES(1, 'Rajesh Kumar', 'Cash', 500.00, NULL, '2024-03-04 10:30:00', 1),
(2, 'Deepa Sharma', 'Food', NULL, 'Dog Food', '2024-03-05 15:45:00', 2), (3, 'Suresh Menon', 'Cash', 1000.00, NULL, '2024-03-06 12:15:00', 3), (4, 'Asha Patel', 'Medicine', NULL, 'Flea Treatment', '2024-03-07 09:00:00', 4),
(5, 'Arjun Rajan', 'Cash', 750.00, NULL, '2024-03-08 14:20:00', 5), (6, 'Ananya Gupta', 'Toys', NULL, 'Cat Toys', '2024-03-09 17:30:00', 6);
INSERT INTO AdoptionEvents (EventID, EventName, EventDate, Location, City, OrganizerID)
VALUES(1, 'Pet Adoption Day', '2024-03-15 14:00:00', 'VGP Golden Beach', 'Chennai', 1), (2, 'Furry Friends Fiesta', '2024-03-20 11:30:00', 'Race Course', 'Coimbatore', 2),
(3, 'Paws Parade', '2024-03-25 13:45:00', 'Goripalayam Ground', 'Madurai', 3),
(4, 'Trichy Pet Carnival', '2024-04-02 10:00:00', 'Maris Theater Ground', 'Trichy', 4),
(5, 'Salem Pet Fest', '2024-04-10 15:15:00', 'Anna Park', 'Salem', 5),
(6, 'Vellore Adoption Drive', '2024-04-18 12:30:00', 'VIT University Ground', 'Vellore', 6);
INSERT INTO Participants (ParticipantID, ParticipantName, ParticipantType, EventID, City)
VALUES(1, 'Aruna Nair', 'Volunteer', 1, 'Chennai'),
(2, 'Karthik Raj', 'Adopter', 2, 'Coimbatore'),
(3, 'Meera Devi', 'Volunteer', 3, 'Madurai'),
(4, 'Vijay Kumar', 'Adopter', 4, 'Trichy'),
(5, 'Priya Reddy', 'Volunteer', 5, 'Salem'),
(6, 'Gopal Krishnan', 'Adopter', 6, 'Vellore');
INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption, ShelterName, OwnerID,
ShelterID)
VALUES(1, 'Charlie', 2, 'Labrador Retriever', 'Dog', 1, 'Chennai Pet Shelter', NULL, 1),
(2, 'Whiskers', 1, 'Siamese', 'Cat', 1, 'Coimbatore Animal Care', NULL, 2),
(3, 'Rocky', 3, 'German Shepherd', 'Dog', 1, 'Madurai Paws Haven', NULL, 3),
(4, 'Mittens', 2, 'Persian', 'Cat', 1, 'Trichy Furry Friends', NULL, 4),
(5, 'Buddy', 1, 'Golden Retriever', 'Dog', 1, 'Salem Animal Sanctuary', NULL, 5),
(6, 'Fluffy', 2, 'Ragdoll', 'Cat', 1, 'Vellore Pet Haven', NULL, 6);
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 Participants.ParticipantName, Participants.ParticipantType
FROM Participants
JOIN AdoptionEvents ON Participants.EventID = AdoptionEvents.EventID
WHERE AdoptionEvents.EventName = 'Pet Adoption Day';

Results Messages

ParticipantName ParticipantType

1 Aruna Nair Volunteer
```

7. Create a stored procedure in SQL that allows a shelter to update its information (name and location) in the "Shelters" table. Use parameters to pass the shelter ID and the new information. Ensure that the procedure performs the update and handles potential errors, such as an invalid shelter ID.

```
CREATE PROCEDURE UpdateShelterInfo
   @ShelterID INT,
    @NewName VARCHAR(255),
    @NewLocation VARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON;
    IF NOT EXISTS (SELECT 1 FROM Shelters WHERE ShelterID = @ShelterID)
        PRINT 'Error: ShelterID does not exist.';
        RETURN;
   FND
   UPDATE Shelters
    SET Name = @NewName,
        Location = @NewLocation
   WHERE ShelterID = @ShelterID;
   PRINT 'Shelter information updated successfully.';
END;
EXEC UpdateShelterInfo @ShelterID = 1, @NewName = 'Fullfy Friends', @NewLocation = 'Perungalathur';
```

```
Messages
Shelter information updated successfully.

Completion time: 2024-03-04T14:40:21.3523909+05:30
```

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.ShelterID = D.ShelterID GROUP BY S.Name;

<b>   </b>	Results	Messages	
	Shelter	Name	Total Donation Amount
1	Coimb	atore Animal Care	0
2	Fullfy F	riends Festival	500
3	Madur	ai Paws Haven	1000
4	Salem	Animal Sanctuary	750
5	Trichy	Furry Friends	0
6	Vellore	Pet Haven	0

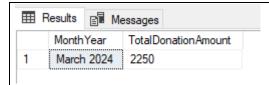
9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.

SELECT Name, Age, Breed, Type FROM Pets WHERE OwnerID 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 FORMAT(DonationDate, 'MMMM yyyy') AS MonthYear, COALESCE(SUM(DonationAmount), 0) AS TotalDonationAmount FROM Donations GROUP BY FORMAT(DonationDate, 'MMMM yyyy');
```

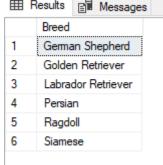


11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

SELECT DISTINCT Breed FROM Pets WHERE (Age BETWEEN 1 AND 3) OR (Age > 5);

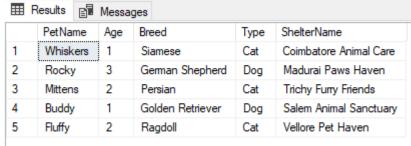
Results Messages

Breed



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

SELECT P.Name AS PetName, P.Age, P.Breed, P.Type, S.Name AS ShelterName
FROM Pets P JOIN Shelters S ON P.ShelterName = S.Name WHERE P.AvailableForAdoption = 1;



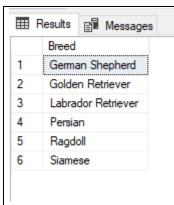
13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

SELECT S.Name AS ShelterName, COUNT(P.ParticipantID) AS TotalParticipants
FROM Participants P JOIN AdoptionEvents AE ON P.EventID = AE.EventID
JOIN Shelters S ON AE.OrganizerID = S.ShelterID WHERE S.City = 'Chennai' GROUP BY S.Name;



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 OwnerID IS NULL;

	PetID	Name	Age	Breed	Type	AvailableForAdoption	ShelterName	OwnerID	ShelterID
1	1	Charlie	2	Labrador Retriever	Dog	1	Chennai Pet Shelter	NULL	1
2	2	Whiskers	1	Siamese	Cat	1	Coimbatore Animal Care	NULL	2
3	3	Rocky	3	German Shepherd	Dog	1	Madurai Paws Haven	NULL	3
4	4	Mittens	2	Persian	Cat	1	Trichy Furry Friends	NULL	4
5	5	Buddy	1	Golden Retriever	Dog	1	Salem Animal Sanctuary	NULL	5
6	6	Fluffy	2	Ragdoll	Cat	1	Vellore Pet Haven	NULL	6

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

```
CREATE TABLE Adoptions (
    AdoptionID INT PRIMARY KEY,
    AdopterName VARCHAR(255),
    PetID INT,
    AdoptionDate DATETIME,
    FOREIGN KEY (PetID) REFERENCES Pets(PetID)
);
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    UserName VARCHAR(255) NOT NULL,
    Email VARCHAR(255) UNIQUE NOT NULL,
    DateOfBirth DATE,
    RegistrationDate DATETIME DEFAULT CURRENT_TIMESTAMP
);
INSERT INTO Users (UserID, UserName, Email, DateOfBirth, RegistrationDate)
VALUES
(1, 'Arvind Kumar', 'arvind.kumar@example.com', '1985-05-10', '2024-03-04 08:30:00'),
(2, 'Priya Devi', 'priya.devi@example.com', '1990-08-15', '2024-03-04 09:45:00'),
(3, 'Karthik Rajan', 'karthik.rajan@example.com', '1988-11-22', '2024-03-04 11:15:00'),
(4, 'Sangeetha Ramesh', 'sangeetha.ramesh@example.com', '1995-03-18', '2024-03-04 13:30:00'),
(5, 'Anand Kumar', 'anand.kumar@example.com', '1980-07-01', '2024-03-04 15:00:00'), (6, 'Deepika Mani', 'deepika.mani@example.com', '1993-12-05', '2024-03-04 16:45:00');
INSERT INTO Adoptions (AdoptionID, AdopterName, PetID, AdoptionDate)
```

# VALUES (1, 'Arvind Kumar', 1, '2024-03-05 10:00:00'), (2, 'Priya Devi', 2, '2024-03-06 11:30:00'), (3, 'Karthik Rajan', 3, '2024-03-07 12:45:00'), (4, 'Sangeetha Ramesh', 4, '2024-03-08 14:15:00'), (5, 'Anand Kumar', 5, '2024-03-09 16:00:00'), (6, 'Deepika Mani', 6, '2024-03-10 17:30:00'); SELECT P.Name AS PetName, U.UserName AS AdopterName FROM Adoptions A JOIN Pets P ON A.PetID = P.PetID JOIN Users U ON A.AdoptionID = U.UserID;



## 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 PetsAvailableForAdoption FROM Shelters S
LEFT JOIN Pets P ON S.ShelterID = P.ShelterID AND P.AvailableForAdoption = 1 GROUP BY S.ShelterID,
S.Name;



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

(8, 'Fluffs', 1, 'Siamese', 'Cat', 1, 'Coimbatore Animal Care', NULL, 2);

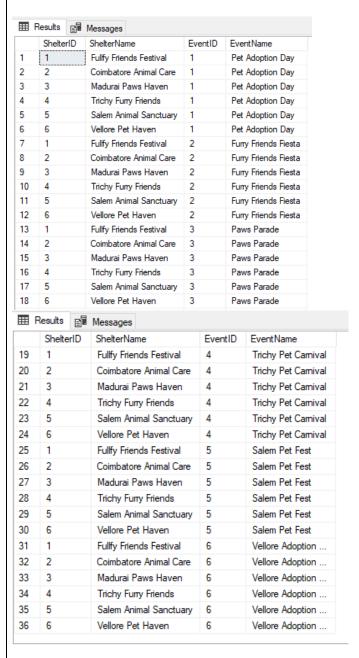
```
INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption, ShelterName, OwnerID,
ShelterID)
VALUES(7, 'Goldy', 2, 'Labrador Retriever', 'Dog', 1, 'Chennai Pet Shelter', NULL, 1),
```

```
SELECT A.PetID AS PetIID, B.PetID AS Pet2ID, A.Breed AS SharedBreed, A.ShelterID AS ShelterID FROM Pets A JOIN Pets B ON A.ShelterID = B.ShelterID AND A.PetID < B.PetID WHERE A.Breed = B.Breed;
```

			SharedBreed	ShelterID
ı	1	7	Labrador Retriever	1
	2	8	Siamese	2

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

SELECT S.ShelterID AS ShelterID, S.Name AS ShelterName, AE.EventID AS EventID, AE.EventName AS EventName FROM Shelters S CROSS JOIN AdoptionEvents AE;



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

SELECT TOP 1 S.ShelterID, S.Name AS ShelterName, COUNT(A.PetID) AS AdoptedPetsCount FROM Shelters S JOIN Pets A ON S.ShelterID = A.ShelterID GROUP BY S.ShelterID, S.Name ORDER BY AdoptedPetsCount DESC;

