Coding Challenges - PetPals, The Pet Adoption Platform

Solutions:

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.**

**Pets table:**  
CREATE TABLE Pets (PetID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Breed VARCHAR(50), Type VARCHAR(50), AvailableForAdoption BIT);

**Shelters table:**  
CREATE TABLE Shelters (ShelterID INT PRIMARY KEY, Name VARCHAR(100), Location VARCHAR(255));

**Donations table:**  
CREATE TABLE Donations (DonationID INT PRIMARY KEY, DonorName VARCHAR(100), DonationType VARCHAR(50), DonationAmount DECIMAL(10, 2), DonationItem VARCHAR(100), DonationDate DATETIME);

**AdoptionEvents table:**  
CREATE TABLE AdoptionEvents (EventID INT PRIMARY KEY, EventName VARCHAR(100), EventDate DATETIME, Location VARCHAR(255));

**Participants table:**  
CREATE TABLE Participants (ParticipantID INT PRIMARY KEY, ParticipantName VARCHAR(100), ParticipantType VARCHAR(50), EventID INT, FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID));



**4. Delete pets whose age > 5**

DELETE FROM Pets WHERE Age > 5;

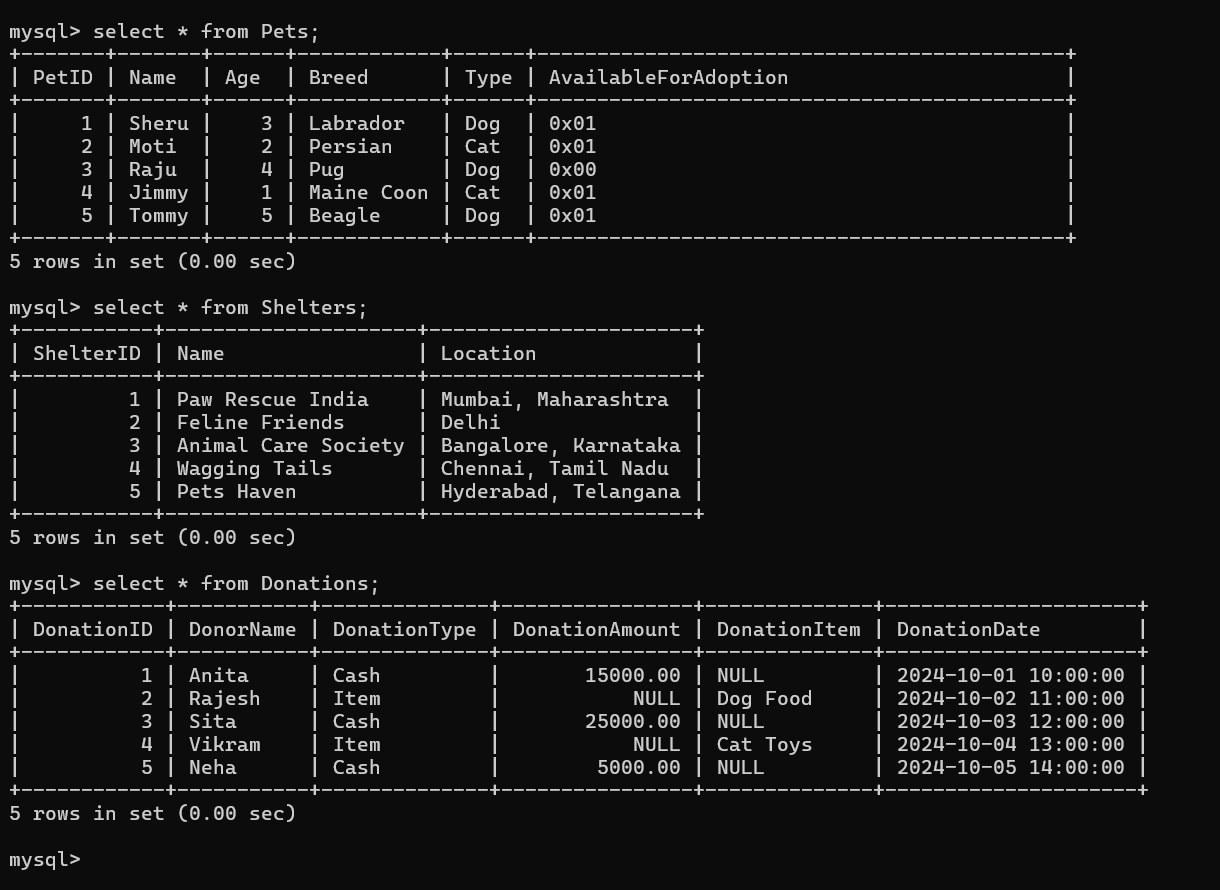
**Inserting data in tables:**

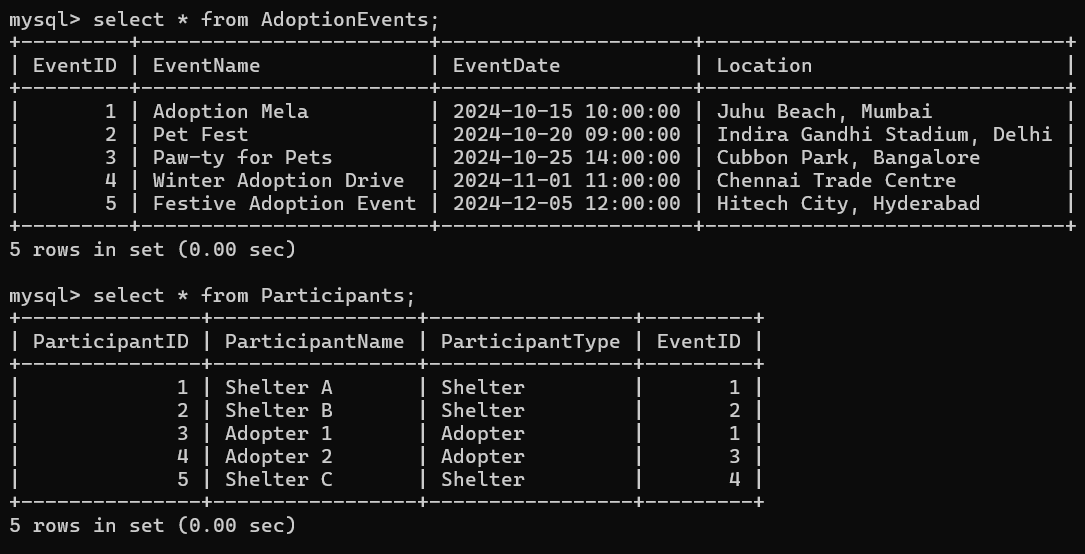
INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption) VALUES (1, 'Sheru', 3, 'Labrador', 'Dog', 1), (2, 'Moti', 2, 'Persian', 'Cat', 1), (3, 'Raju', 4, 'Pug', 'Dog', 0), (4, 'Jimmy', 1, 'Maine Coon', 'Cat', 1), (5, 'Tommy', 5, 'Beagle', 'Dog', 1);

INSERT INTO Shelters (ShelterID, Name, Location) VALUES (1, 'Paw Rescue India', 'Mumbai, Maharashtra'), (2, 'Feline Friends', 'Delhi'), (3, 'Animal Care Society', 'Bangalore, Karnataka'), (4, 'Wagging Tails', 'Chennai, Tamil Nadu'), (5, 'Pets Haven', 'Hyderabad, Telangana');

INSERT INTO Donations (DonationID, DonorName, DonationType, DonationAmount, DonationItem, DonationDate) VALUES (1, 'Anita', 'Cash', 15000.00, NULL, '2024-10-01 10:00:00'), (2, 'Rajesh', 'Item', NULL, 'Dog Food', '2024-10-02 11:00:00'), (3, 'Sita', 'Cash', 25000.00, NULL, '2024-10-03 12:00:00'), (4, 'Vikram', 'Item', NULL, 'Cat Toys', '2024-10-04 13:00:00'), (5, 'Neha', 'Cash', 5000.00, NULL, '2024-10-05 14:00:00');  
INSERT INTO AdoptionEvents (EventID, EventName, EventDate, Location) VALUES (1, 'Adoption Mela', '2024-10-15 10:00:00', 'Juhu Beach, Mumbai'), (2, 'Pet Fest', '2024-10-20 09:00:00', 'Indira Gandhi Stadium, Delhi'), (3, 'Paw-ty for Pets', '2024-10-25 14:00:00', 'Cubbon Park, Bangalore'), (4, 'Winter Adoption Drive', '2024-11-01 11:00:00', 'Chennai Trade Centre'), (5, 'Festive Adoption Event', '2024-12-05 12:00:00', 'Hitech City, Hyderabad');

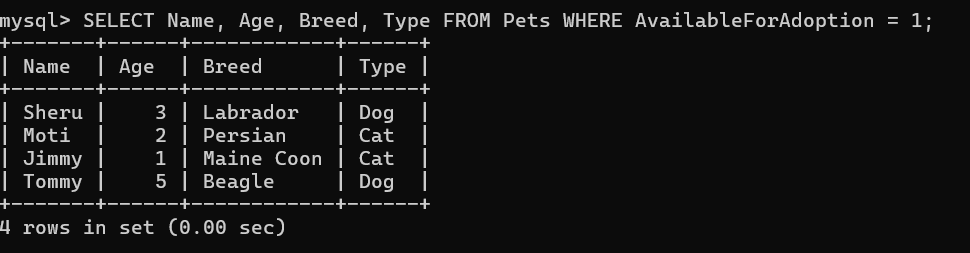
INSERT INTO Participants (ParticipantID, ParticipantName, ParticipantType, EventID) VALUES (1, 'Shelter A', 'Shelter', 1), (2, 'Shelter B', 'Shelter', 2), (3, 'Adopter 1', 'Adopter', 1), (4, 'Adopter 2', 'Adopter', 3), (5, 'Shelter C', 'Shelter', 4);





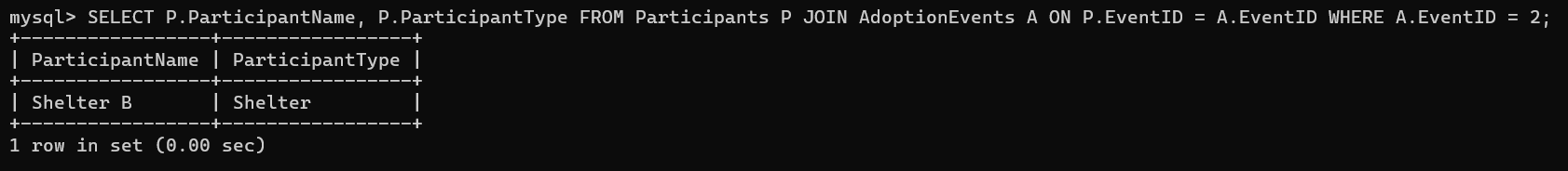
**5. Write an SQL query that retrieves a list of available pets (those marked as available for adoption) from the "Pets" table. Include the pet's name, age, breed, and type in the result set. Ensure that the query filters out pets that are not available for adoption.**

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 A ON P.EventID = A.EventID WHERE A.EventID = 2;



**7. Show the details of pet id, pet name, breed and pet type from pets along with all columns adoptionevents and all columns participant.**

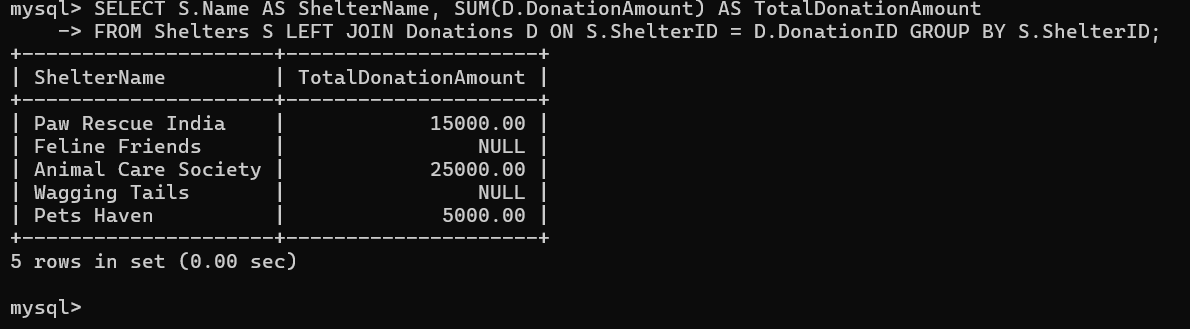
SELECT p.PetID, p.Name AS PetName, p.Breed, p.Type AS PetType, ae.EventID, ae.EventName, ae.EventDate, ae.Location, par.ParticipantID, par.ParticipantName, par.ParticipantType FROM Pets p LEFT JOIN Participants par ON par.EventID IN (SELECT EventID FROM AdoptionEvents) LEFT JOIN AdoptionEvents ae ON par.EventID = ae.EventID;



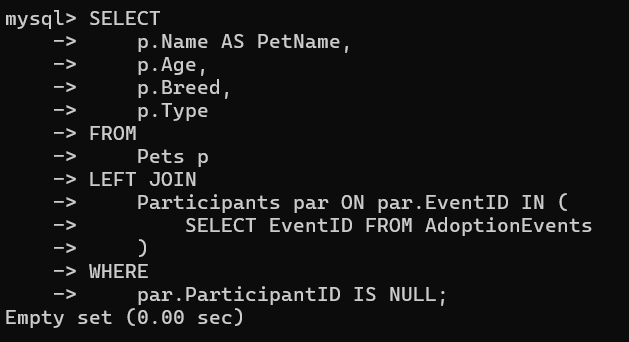
**8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.**

SELECT S.Name AS ShelterName, SUM(D.DonationAmount) AS TotalDonationAmount

FROM Shelters S LEFT JOIN Donations D ON S.ShelterID = D.DonationID GROUP BY S.ShelterID;

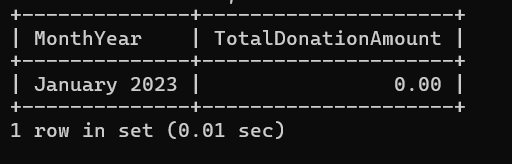


**9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.**



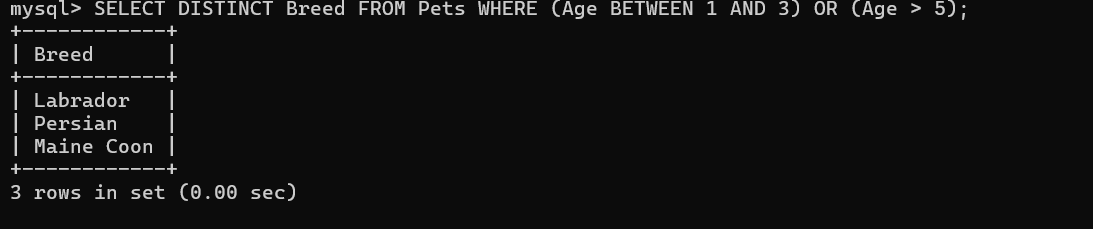
**10.** **Write an SQL query that retrieves the total donation amount for each month and year (e.g.,January 2023) from the "Donations" table. The result should include the month-year and the corresponding total donation amount. Ensure that the query handles cases where no donations were made in a specific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

SELECT DATE\_FORMAT(m.Month, '%M %Y') AS MonthYear, COALESCE(SUM(d.DonationAmount), 0) AS TotalDonationAmount FROM (SELECT DATE\_FORMAT('2023-01-01', '%Y-%m-01') AS Month) AS m LEFT JOIN Donations d ON DATE\_FORMAT(d.DonationDate, '%Y-%m') = DATE\_FORMAT(m.Month, '%Y-%m') GROUP BY m.Month ORDER BY m.Month;



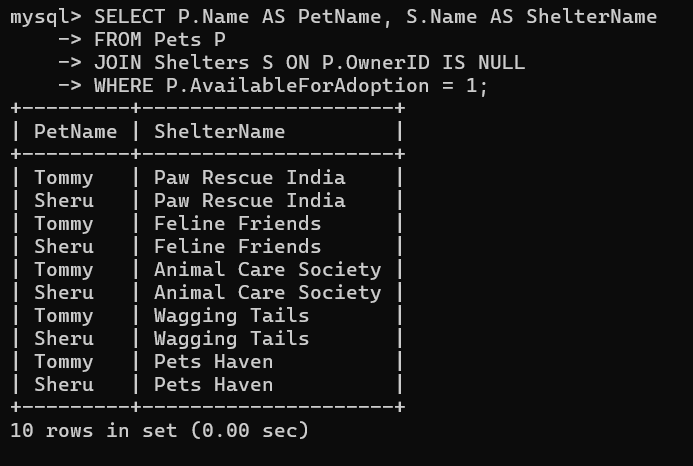
**11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.**

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



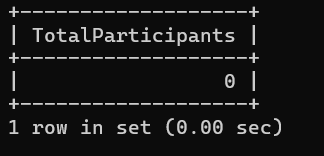
**12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.**

SELECT P.Name AS PetName, S.Name AS ShelterName FROM Pets P JOIN Shelters S ON P.OwnerID IS NULL WHERE P.AvailableForAdoption = 1;



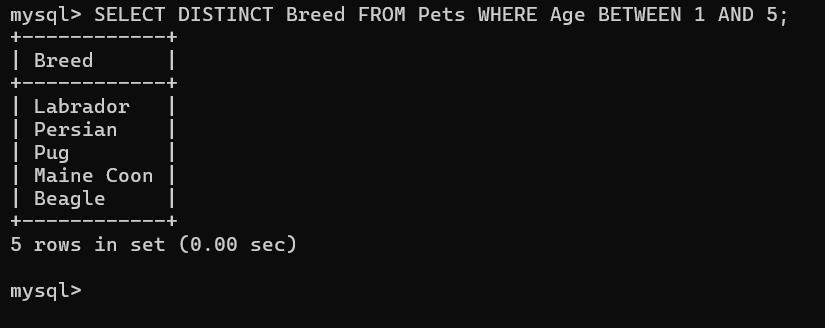
**13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai**

SELECT COUNT(P.ParticipantID) AS TotalParticipants FROM Participants P JOIN AdoptionEvents A ON P.EventID = A.EventID JOIN Shelters S ON A.Location = S.Location WHERE S.Location LIKE 'Chennai%';



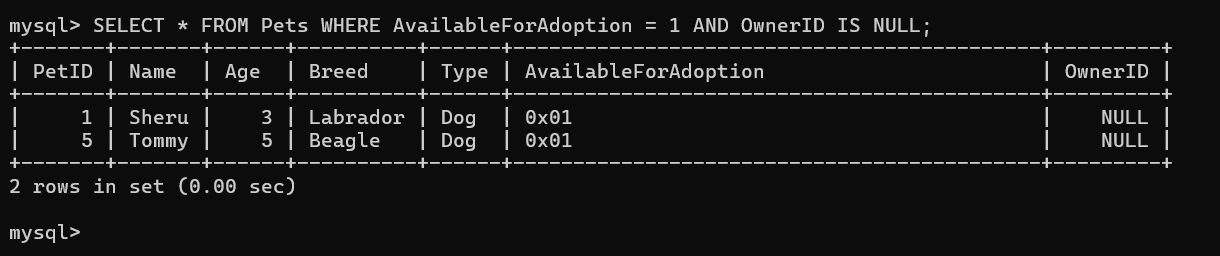
**14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years**

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



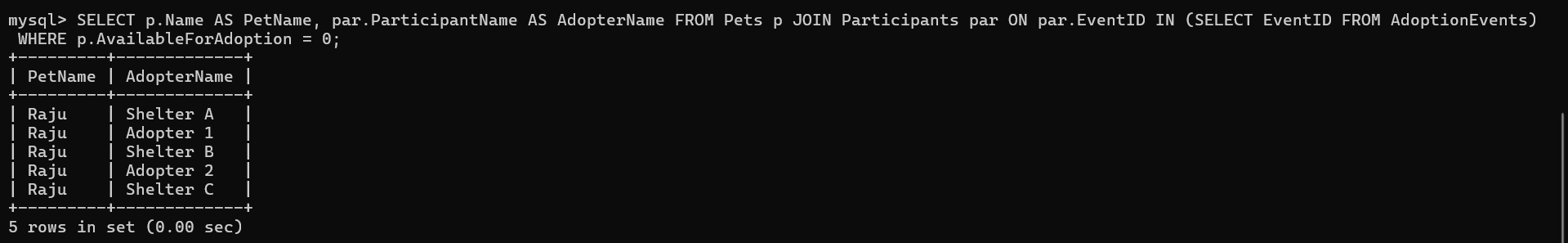
**15.Find the pets that have not been adopted by selecting their information from the 'Pet' table.**

SELECT \* FROM Pets WHERE AvailableForAdoption = 1 AND OwnerID IS NULL;



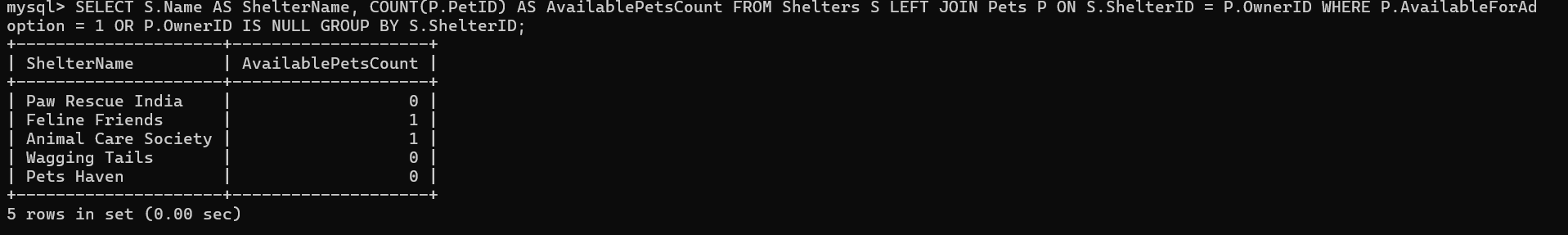
**16. Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'Participants' tables.**

SELECT p.Name AS PetName, par.ParticipantName AS AdopterName FROM Pets p JOIN Participants par ON par.EventID IN (SELECT EventID FROM AdoptionEvents) WHERE p.AvailableForAdoption = 0;



**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 S.ShelterID = P.OwnerID WHERE P.AvailableForAdoption = 1 OR P.OwnerID IS NULL GROUP BY S.ShelterID;



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

SELECT P1.Name AS Pet1Name, P2.Name AS Pet2Name, P1.Breed, S.Name AS ShelterName FROM Pets P1 JOIN Pets P2 ON P1.Breed = P2.Breed AND P1.PetID <> P2.PetID JOIN Shelters S ON P1.OwnerID = S.ShelterID WHERE P1.OwnerID IS NOT NULL AND P2.OwnerID IS NOT NULL;



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

SELECT S.Name AS ShelterName, A.EventName FROM Shelters S, AdoptionEvents A;



**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 S.ShelterID = P.OwnerID WHERE P.OwnerID IS NOT NULL GROUP BY S.ShelterID ORDER BY AdoptedPetsCount DESC LIMIT 1;

