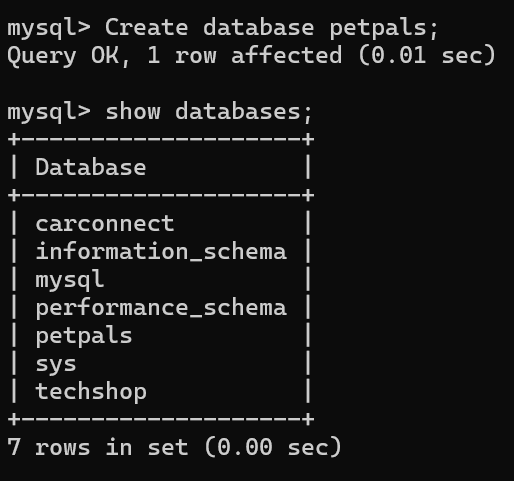
1. **Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals**”.

Create database petpals;



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

Create table Pets(

PetID INT PRIMARY KEY,

Name VARCHAR(200),

Age INT,

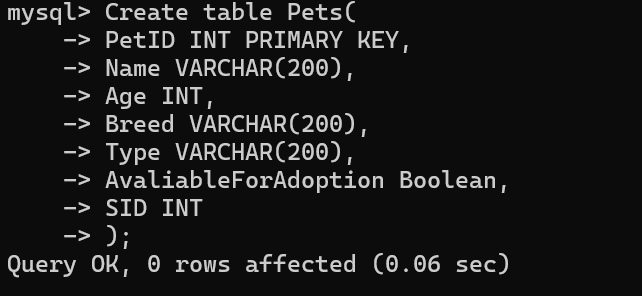
Breed VARCHAR(200),

Type VARCHAR(200),

AvaliableForAdoption Boolean

SID INT

);



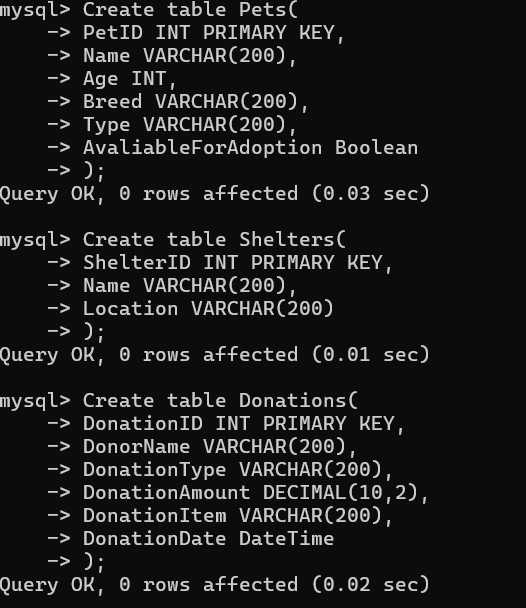
Create table Shelters(

ShelterID INT PRIMARY KEY,

Name VARCHAR(200),

Location VARCHAR(200)

);



Create table Donations(

DonationID INT PRIMARY KEY,

DonorName VARCHAR(200),

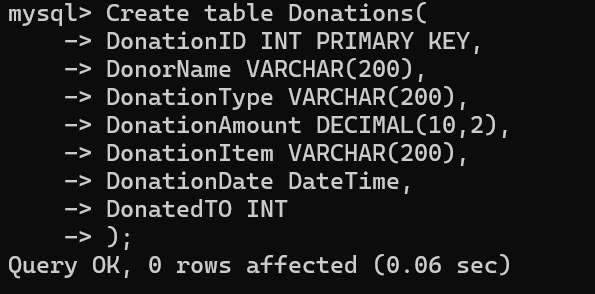
DonationType VARCHAR(200),

DonationAmount DECIMAL(10,2),

DonationItem VARCHAR(200),

DonationDate DateTime

);



Create Table AdoptionEvents(

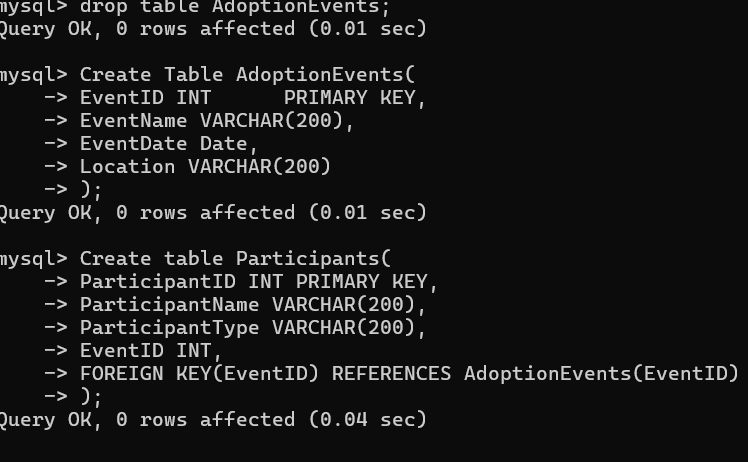
EventID INT PRIMARY KEY,

EventName VARCHAR(200),

EventDate Date,

Location VARCHAR(200)

);



Create table Participants(

ParticipantID INT PRIMARY KEY,

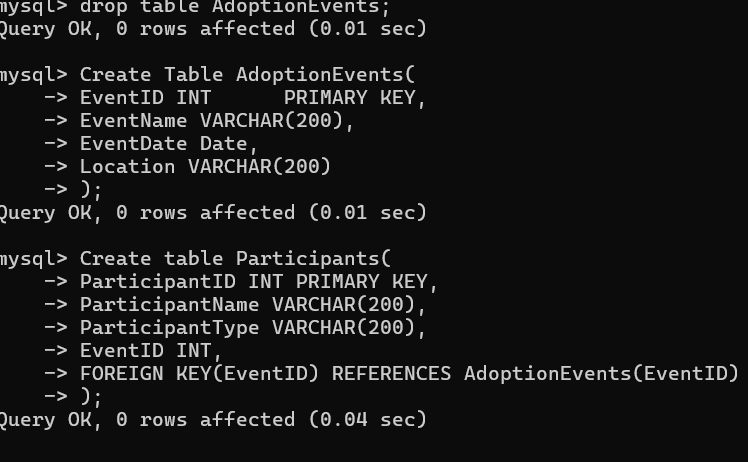
ParticipantName VARCHAR(200),

ParticipantType VARCHAR(200),

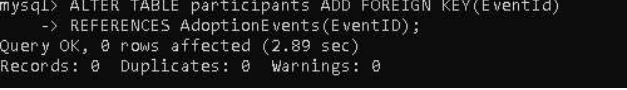
EventID INT,

FOREIGN KEY(EventID) REFERENCES AdoptionEvents(EventID)

);



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



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

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.**

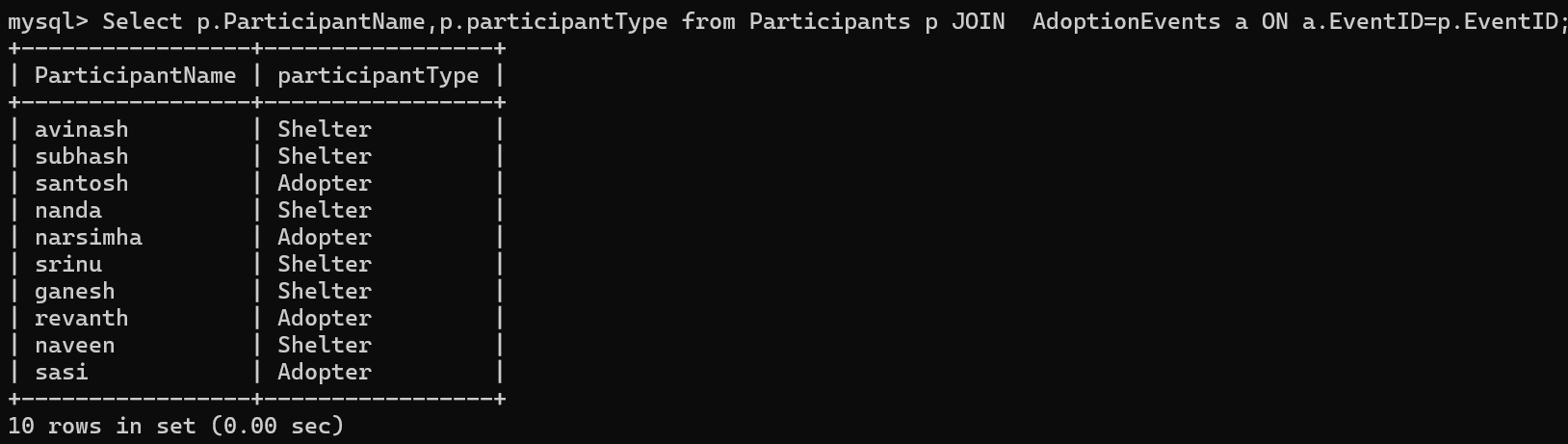
Select PetID, Name, Age, Breed, Type from Pets where AvaliableForAdoption=1;



**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**.

Set @eventID=5;

Select p.ParticipantName,p.participantType from Participants p JOIN AdoptionEvents a ON a.EventID=p.EventID;



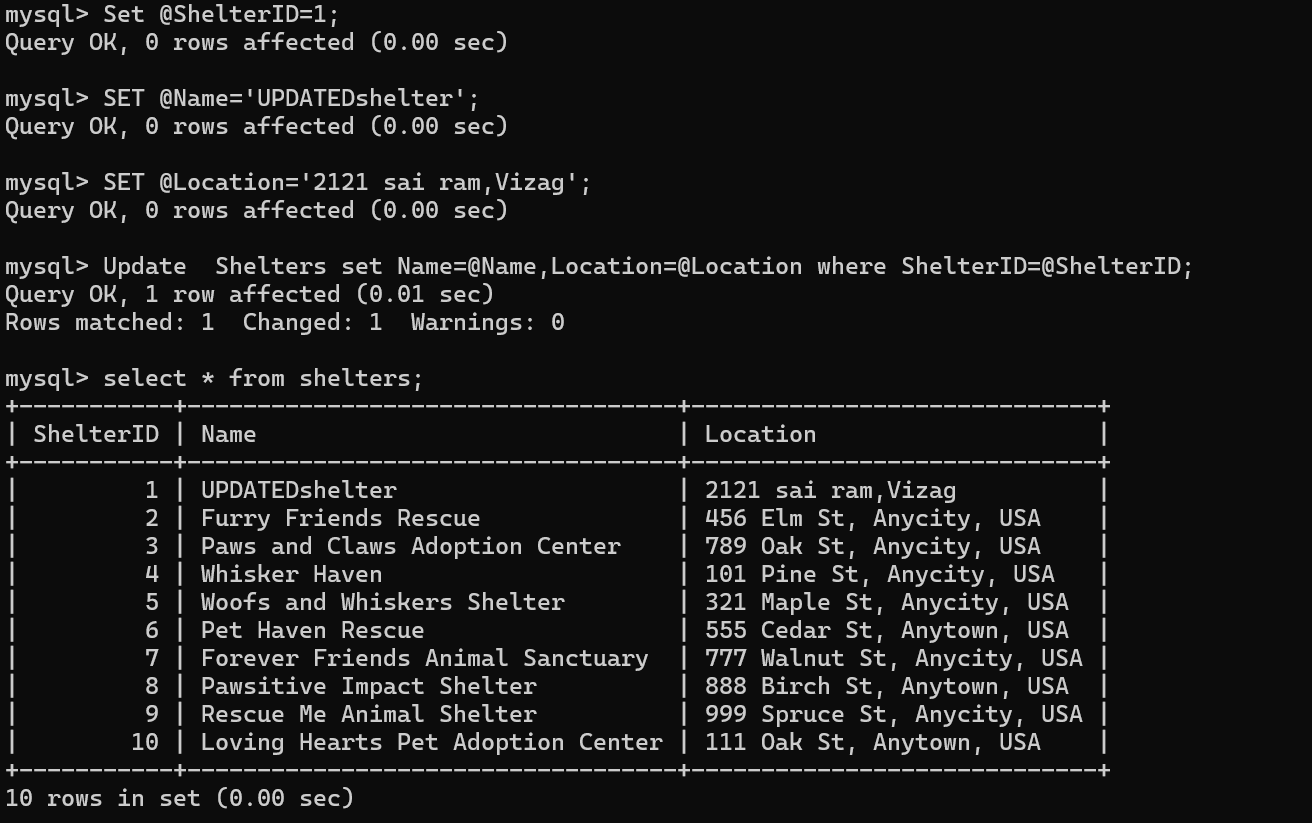
7. **Create a stored procedure in SQL that allows a shelter to update its information (name and location) in the "Shelters" table. Use parameters to pass the shelter ID and the new information. Ensure that the procedure performs the update and handles potential errors, such as an invalid shelter ID.**

Set @ShelterID=1;

SET @Name='UPDATEDshelter';

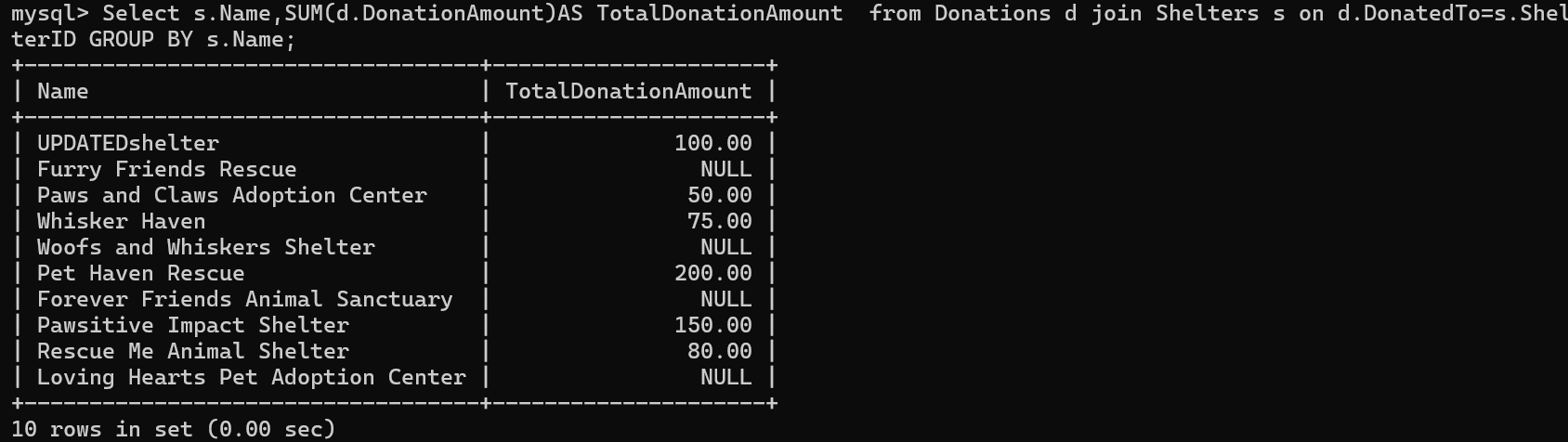
SET @Location='2121 sai ram,Vizag';

Update Shelter set Name=@Name,Location=@Location where ShelterID=@ ShelterID;



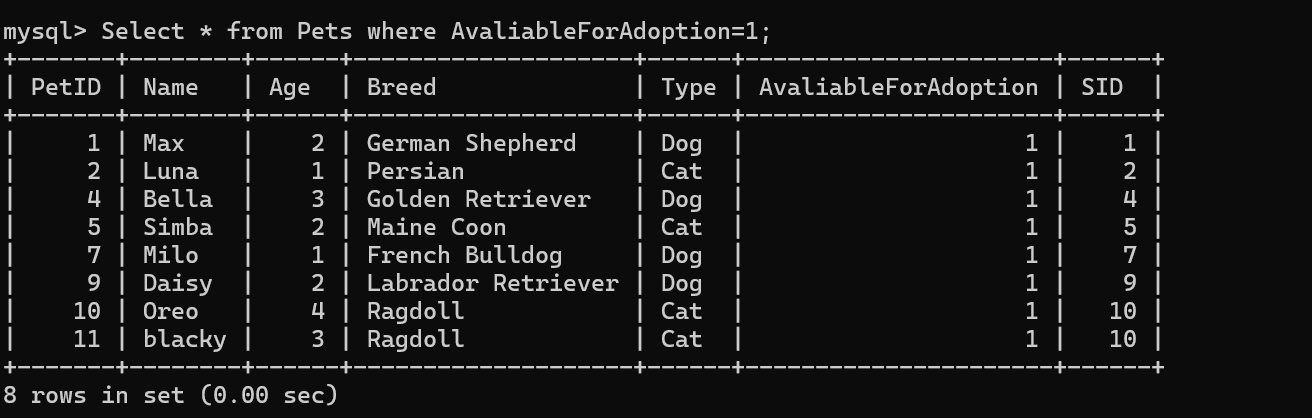
**8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.**

Select s.Name,SUM(d.DonationAmount)AS TotalDonationAmount from Donations d join Shelters s on d.DonatedTo=s.ShelterID 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 "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.**

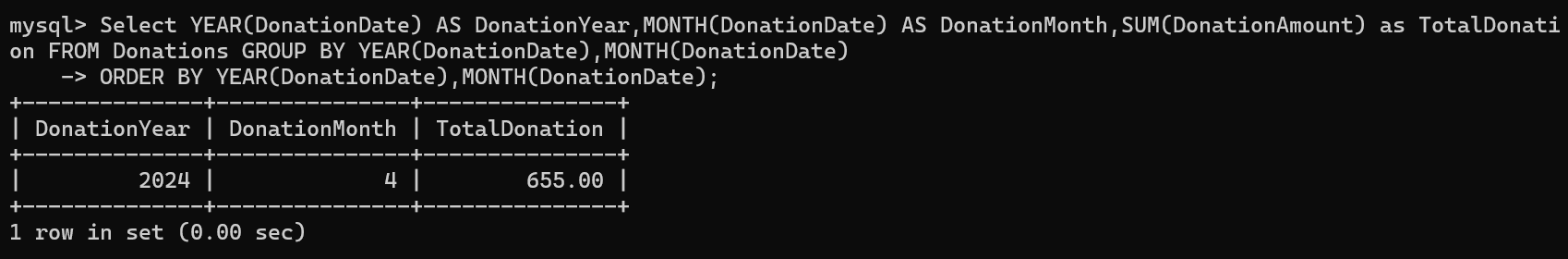
Select \* from Pets whereAvaliableForAdoption=1;

****

**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.**

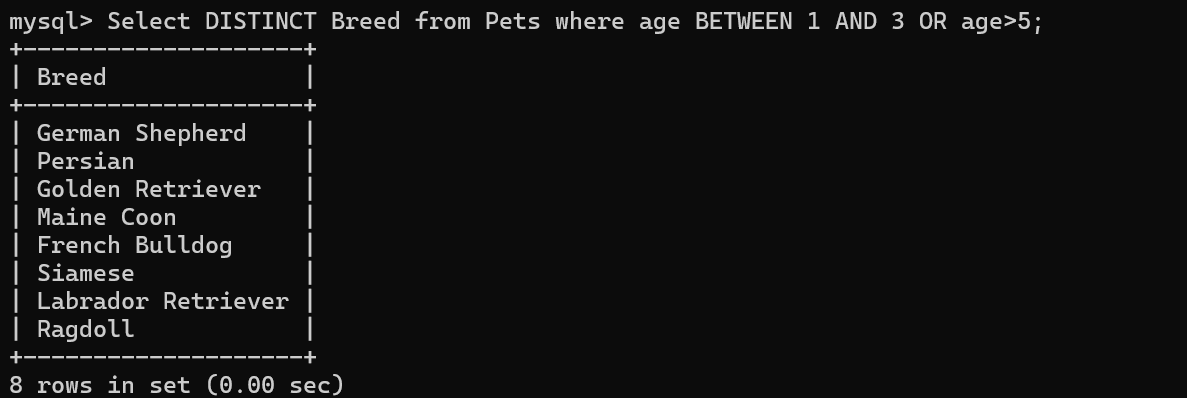
Select YEAR(DonationDate) AS DonationYear,MONTH(DonationDate) AS DonationMonth,SUM(DonationAmount) as TotalDonation FROM Donations GROUP BY YEAR(DonationDate),MONTH(DonationDate)

ORDER BY YEAR(DonationDate),MONTH(DonationDate);

****

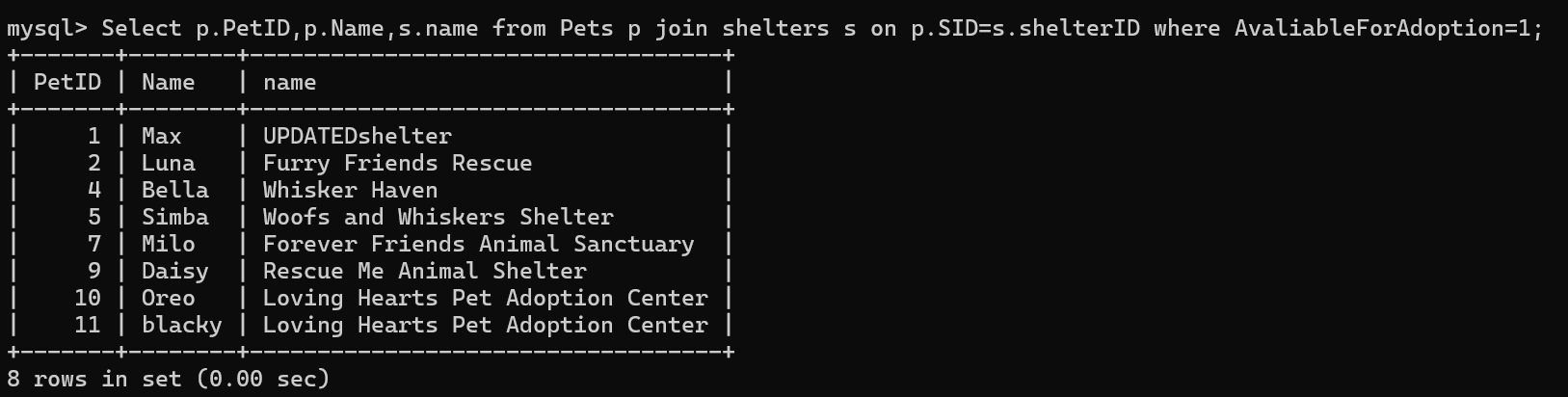
**11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.**

Select DISTINCT Breed from Pets where age BETWEEN 1 AND 3 OR age>5;

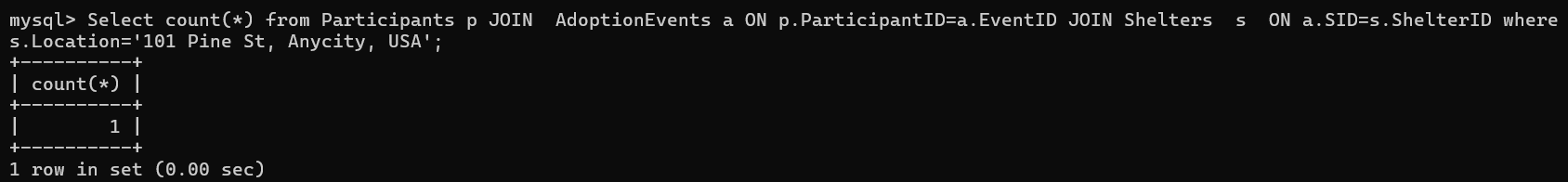


**12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.**

Select p.PetID,p.Name,s.name from Pets p join shelters s on p.SID=s.shelterID where AvaliableForAdoption=1;

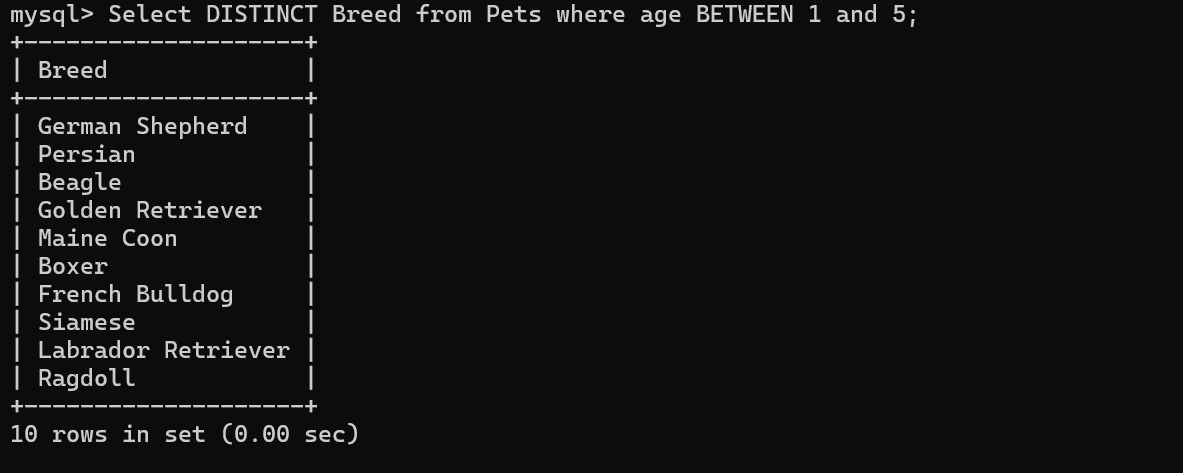


**13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai**

Select count(\*) from Participants p JOIN AdoptionEvents a ON p.ParticipantID=a.EventID JOIN Shelters s ON a.SID=s.ShelterID where s.Location='101 Pine St, Anycity, USA'; 

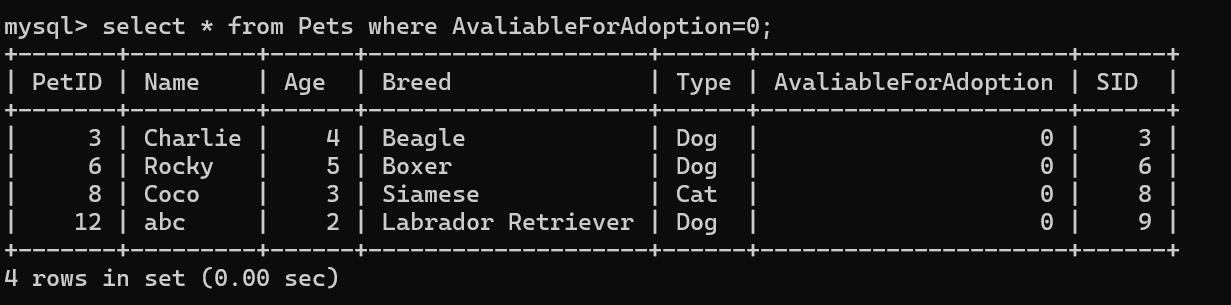
**14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.**

Select DISTINCT Breed from Pets where age BETWEEN 1 and 5;



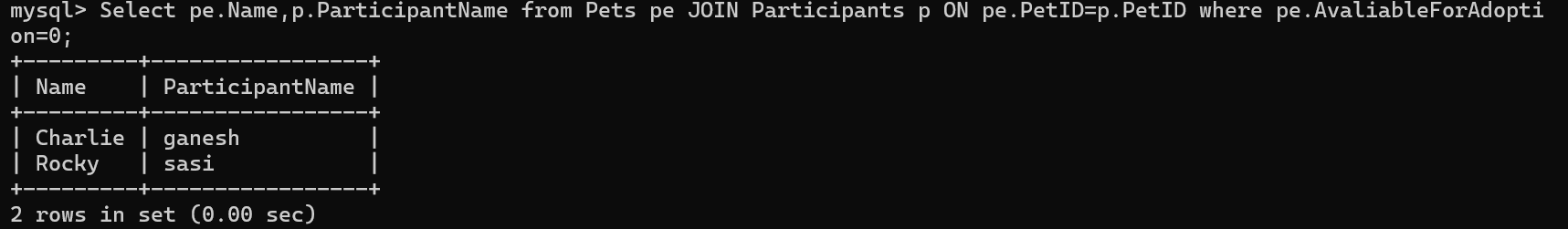
**15. Find the pets that have not been adopted by selecting their information from the 'Pet' table.**

Select \* from Pets where AvailableForAdoption=0;



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

Select pe.Name,p.ParticipantName from Pets pe JOIN Participants p ON pe.PetID=p.PetID where pe.AvaliableForAdoption=0;



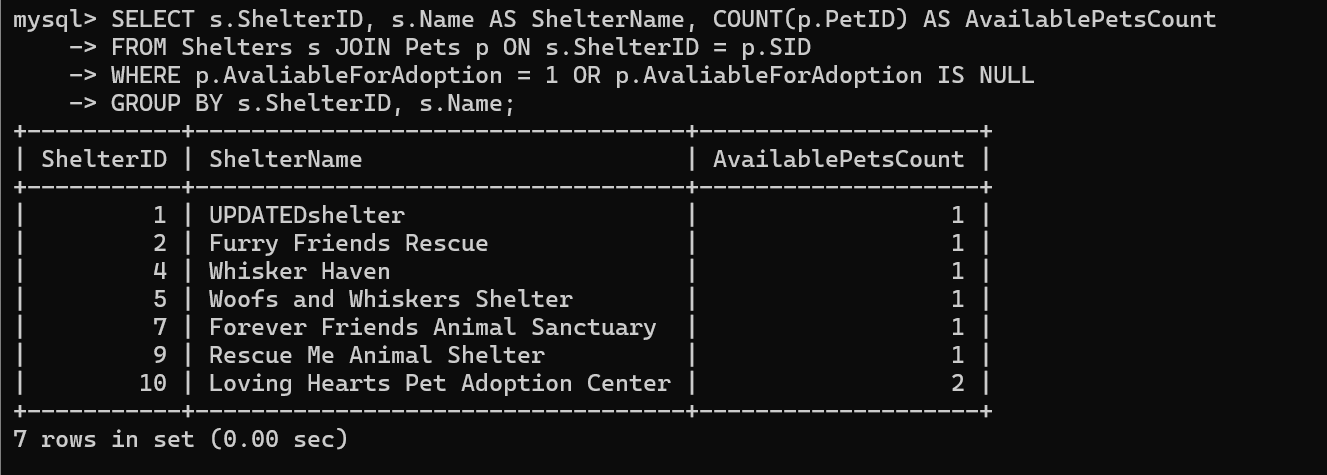
**17. Retrieve a list of all shelters along with the count of pets currently available for adoption in each shelter.**

SELECT s.ShelterID, s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePetsCount

FROM Shelters s JOIN Pets p ON s.ShelterID = p.SID

WHERE p.AvaliableForAdoption = 1 OR p.AvaliableForAdoption IS NULL

GROUP BY s.ShelterID, s.Name;

****

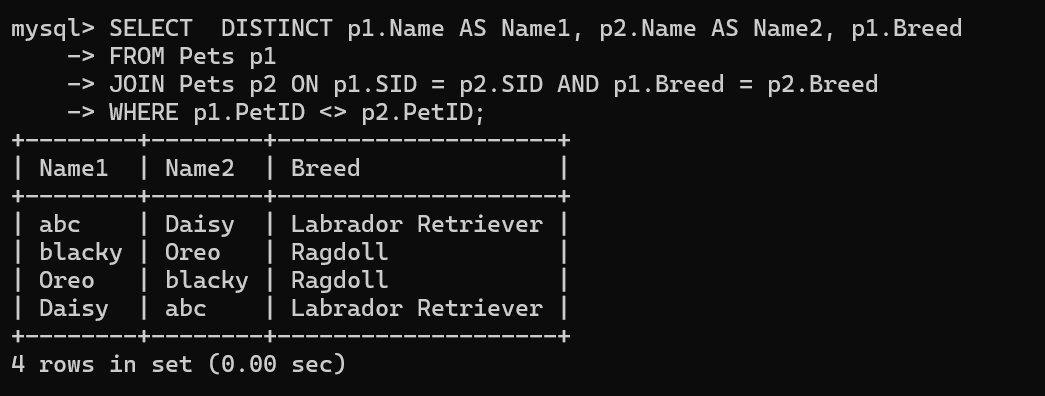
**18. Find pairs of pets from the same shelter that have the same breed.**

SELECT DISTINCT p1.Name AS Name1, p2.Name AS Name2, p1.Breed

FROM Pets p1

JOIN Pets p2 ON p1.SID = p2.SID AND p1.Breed = p2.Breed

WHERE p1.PetID <> p2.PetID;

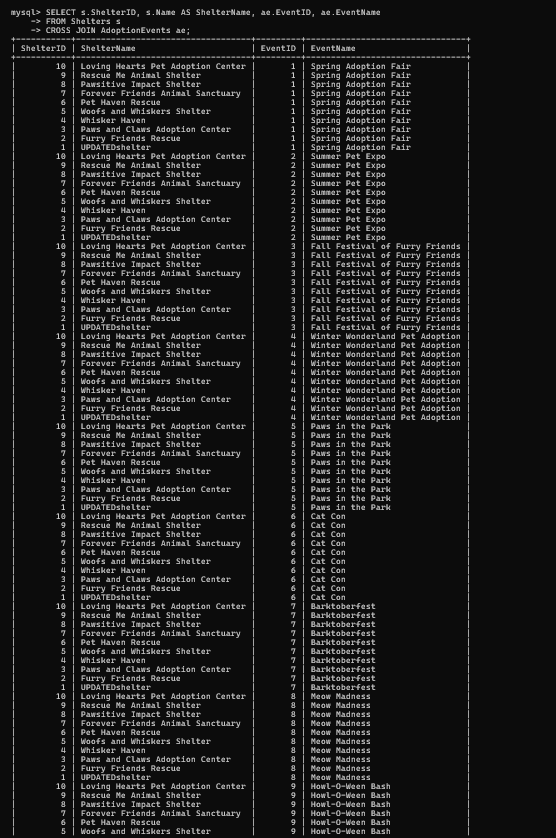
****

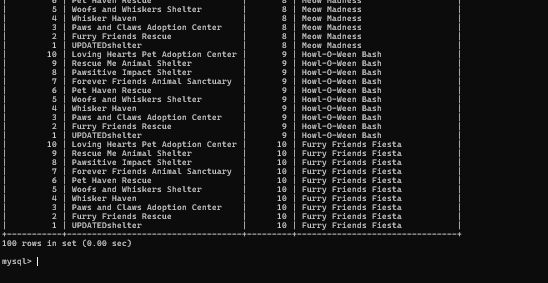
**19. List all possible combinations of shelters and adoption events.**

SELECT s.ShelterID, s.Name AS ShelterName, ae.EventID, ae.EventName

FROM Shelters s

CROSS JOIN AdoptionEvents ae;





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

SELECT s.ShelterID, s.Name AS ShelterName, COUNT(p.PetID) AS AdoptedPetsCount

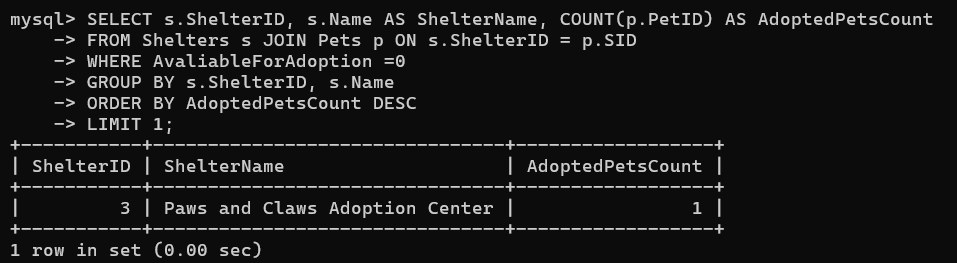
FROM Shelters s JOIN Pets p ON s.ShelterID = p.SID

WHERE AvaliableForAdoption =0

GROUP BY s.ShelterID, s.Name

ORDER BY AdoptedPetsCount DESC

LIMIT 1;

****