SQL CODE CHALLENGE - **PetPals , The Pet Adoption Platform**

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

   --->CREATE DATABASE PetPals;

   USE PetPals;

2. Create tables for pets, shelters, donations, adoption events, and participants.
3. Define appropriate primary keys, foreign keys, and constraints.

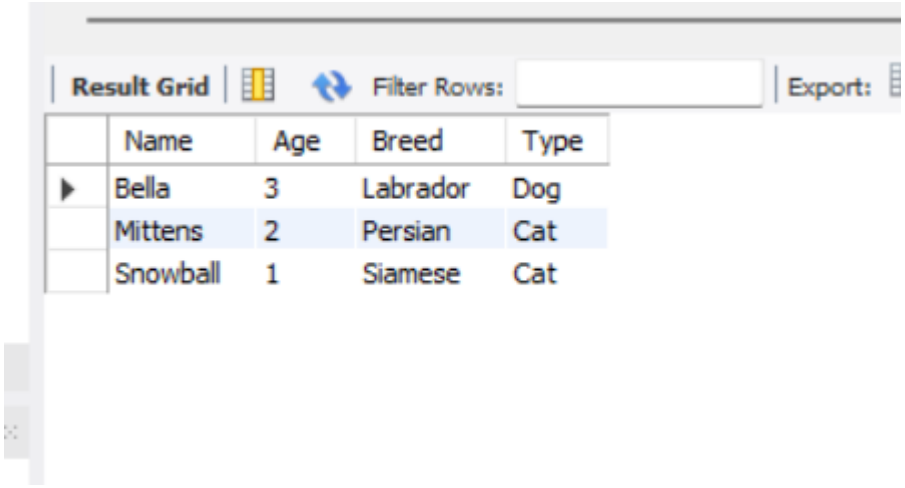
```
CREATE TABLE Pets (
PetID INT PRIMARY KEY,
Name VARCHAR(100),
Age INT,
Breed VARCHAR(100),
Type VARCHAR(50),
AvailableForAdoption BIT,
ShelterID INT
);

CREATE TABLE Shelters (
ShelterID INT PRIMARY KEY,
Name VARCHAR(100),
Location VARCHAR(200)
);

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

CREATE TABLE AdoptionEvents (
EventID INT PRIMARY KEY,
EventName VARCHAR(100),
EventDate DATETIME,
Location VARCHAR(200)
);
```

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


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

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



INSERT INTO Shelters (ShelterID, Name, Location)
VALUES
(1, 'Sunny Days Shelter', '123 Sunshine Ave, Cityville'),
(2, 'Happy Paws Shelter', '456 Happy St, Townsville');


INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption, ShelterID)
VALUES
(1, 'Bella', 3, 'Labrador', 'Dog', 1, 1),
(2, 'Mittens', 2, 'Persian', 'Cat', 1, 2),
(3, 'Charlie', 5, 'Beagle', 'Dog', 0, 1),
(4, 'Snowball', 1, 'Siamese', 'Cat', 1, 2);


INSERT INTO Donations (DonationID, DonorName, DonationType, DonationAmount,
DonationItem, DonationDate, ShelterID)
VALUES
(1, 'John Doe', 'Cash', 500.00, NULL, '2025-04-15 12:00:00', 1),
(2, 'Jane Smith', 'Item', NULL, 'Cat Food', '2025-04-14 14:00:00', 2),
(3, 'Mark Lee', 'Cash', 250.00, NULL, '2025-04-13 16:00:00', 1);


INSERT INTO AdoptionEvents (EventID, EventName, EventDate, Location)
VALUES
(1, 'Spring Adoption Event', '2025-05-01 10:00:00', 'Sunny Days Shelter'),
(2, 'Fall Adoption Event', '2025-09-01 12:00:00', 'Happy Paws Shelter');



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

VALUES
(1, 'Sunny Days Shelter', 'Shelter', 1),
(2, 'Happy Paws Shelter', 'Shelter', 2),
(3, 'Alice Green', 'Adopter', 1),
(4, 'Bob Brown', 'Adopter', 2);

4. 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.

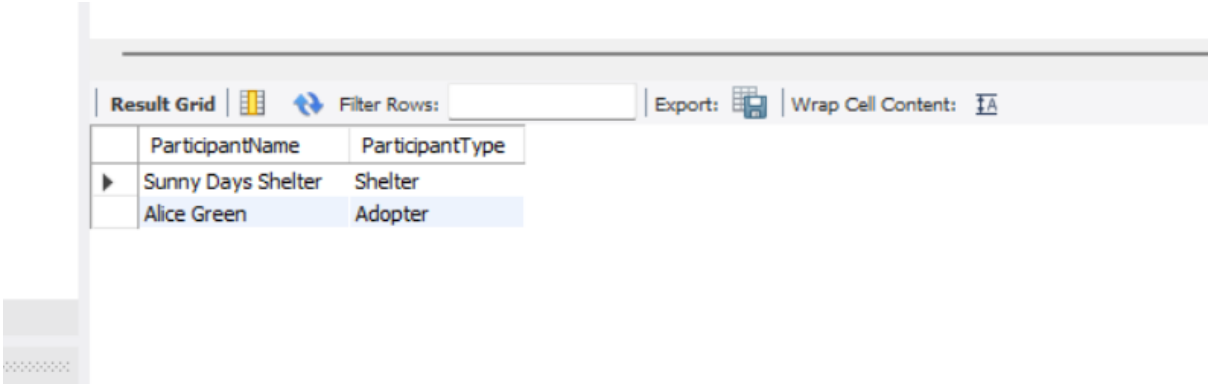
→ SELECT Name, Age, Breed, Type
FROM Pets
WHERE AvailableForAdoption = 1;



Name	Age	Breed	Type
Bella	3	Labrador	Dog
Mittens	2	Persian	Cat
Snowball	1	Siamese	Cat

5. 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 ParticipantName, ParticipantType
FROM Participants
WHERE EventID = 1;

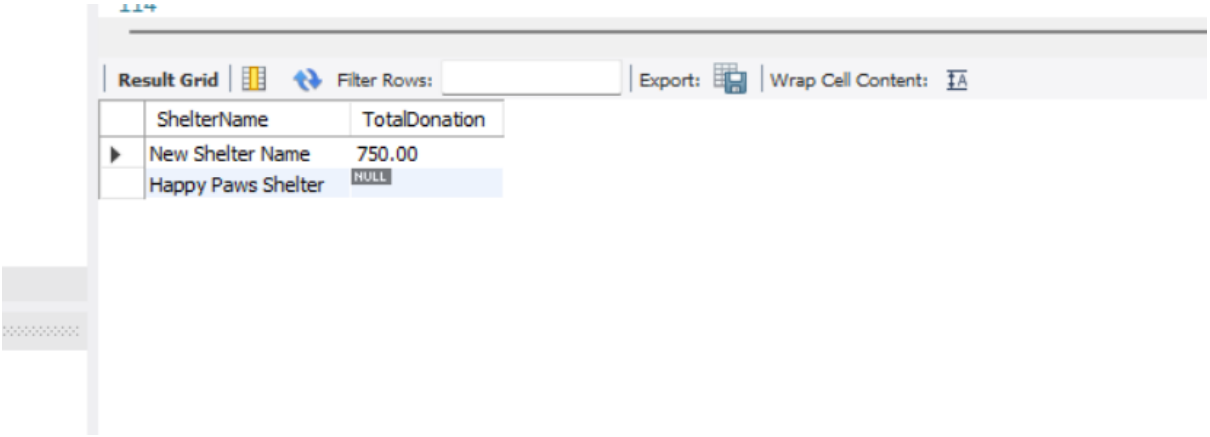


6. Create a stored procedure in SQL that allows a shelter to update its information (name and location) in the "Shelters" table. Use parameters to pass the shelter ID and the new information. Ensure that the procedure performs the update and handles potential errors, such as an invalid shelter ID.

→ UPDATE Shelters
SET Name = 'New Shelter Name', Location = 'New Shelter Location'
WHERE ShelterID = 1;

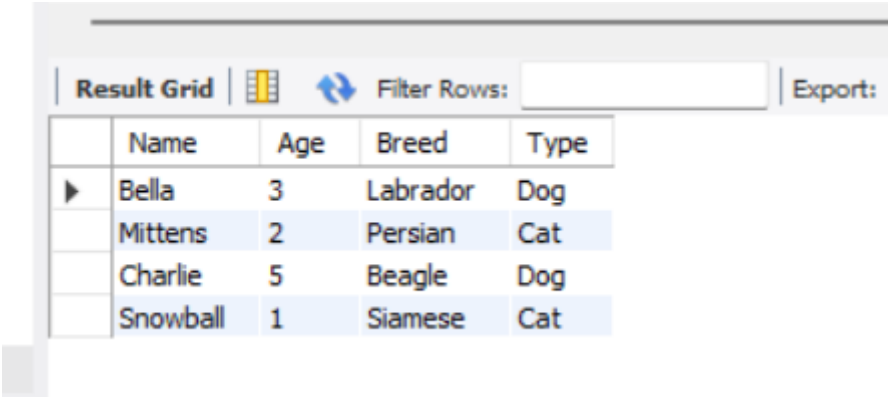
7. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.

→ SELECT s.Name AS ShelterName,
SUM(d.DonationAmount) AS TotalDonation
FROM Shelters s
LEFT JOIN Donations d ON s.ShelterID = d.ShelterID
GROUP BY s.Name;



8. 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.

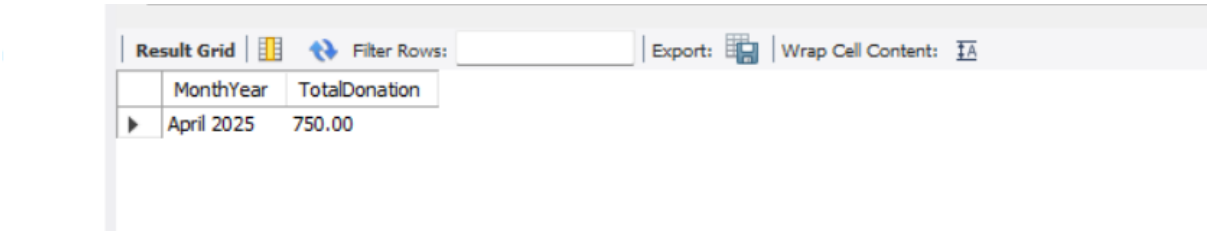
→ ALTER TABLE Pets
ADD COLUMN OwnerID INT;
SELECT Name, Age, Breed, Type
FROM Pets
WHERE OwnerID IS NULL;


Result Grid | Filter Rows: | Export:

Name	Age	Breed	Type
Bella	3	Labrador	Dog
Mittens	2	Persian	Cat
Charlie	5	Beagle	Dog
Snowball	1	Siamese	Cat

9. 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.

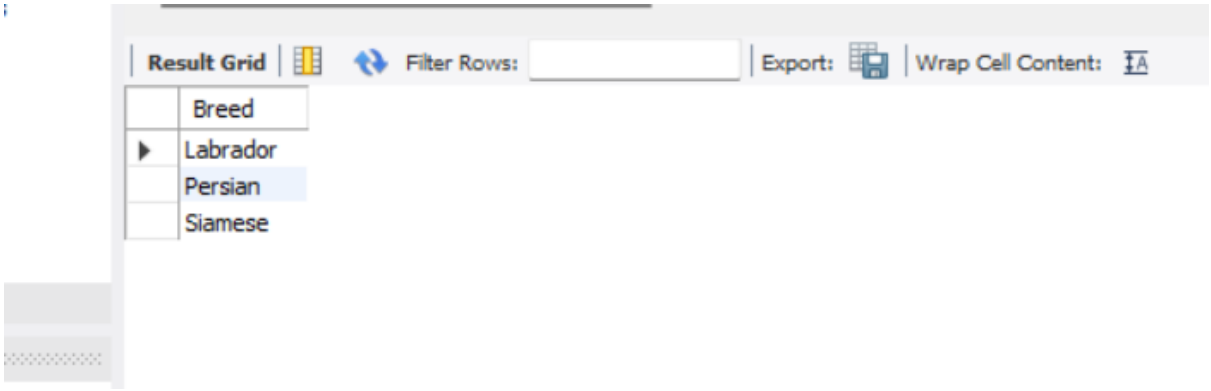
→ SELECT
DATE_FORMAT(DonationDate, '%M %Y') AS MonthYear,
SUM(DonationAmount) AS TotalDonation
FROM Donations
GROUP BY MonthYear;


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MonthYear	TotalDonation
April 2025	750.00

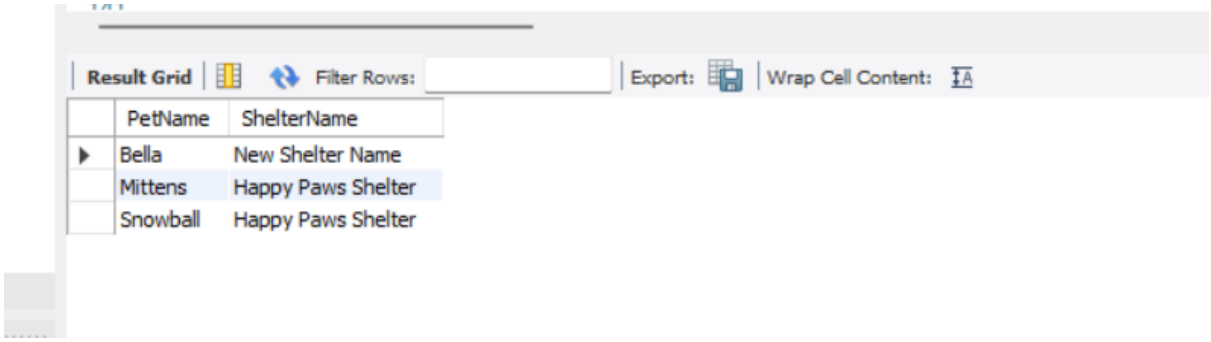
10. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

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



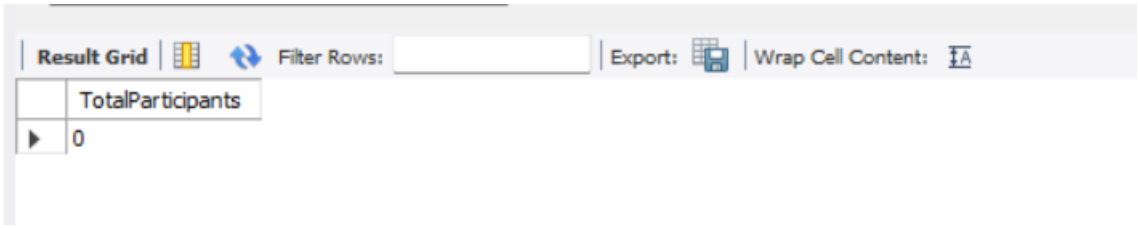
11. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

→ SELECT p.Name AS PetName, s.Name AS ShelterName
FROM Pets p
JOIN Shelters s ON p.ShelterID = s.ShelterID
WHERE p.AvailableForAdoption = 1;



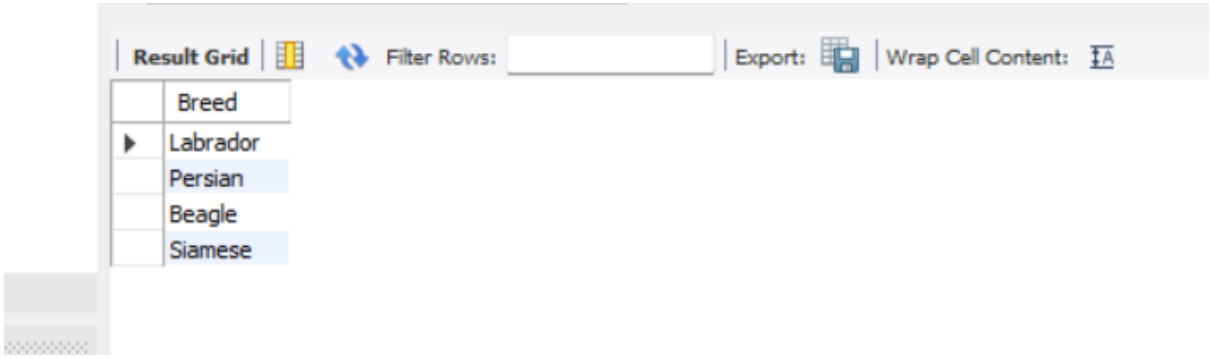
12. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

→ SELECT COUNT(p.ParticipantID) AS TotalParticipants
FROM Participants p
JOIN AdoptionEvents e ON p.EventID = e.EventID
WHERE e.Location = 'Chennai';



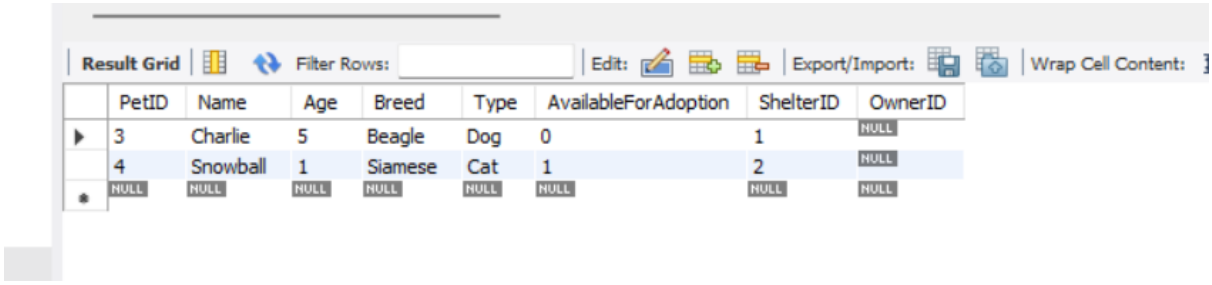
13. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

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



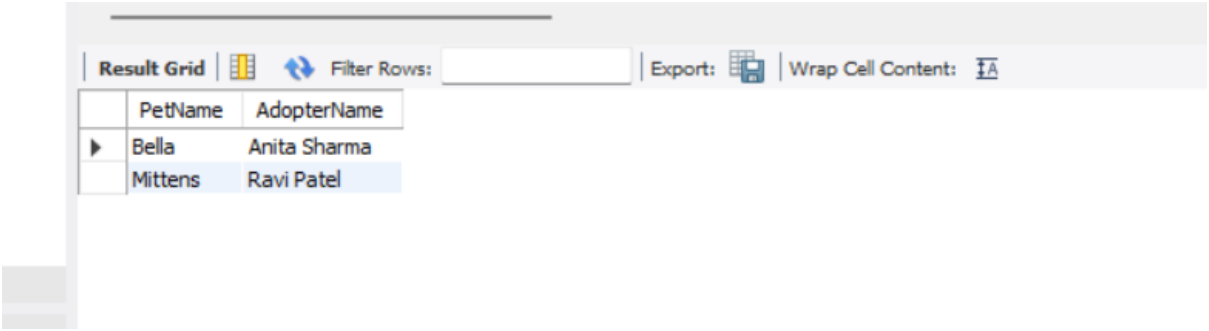
14. Find the pets that have not been adopted by selecting their information from the 'Pet' table.

→ SELECT *
FROM Pets
WHERE PetID NOT IN (SELECT PetID FROM Adoption);



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

→ SELECT p.Name AS PetName, u.Name AS AdopterName
FROM Adoption a
JOIN Pets p ON a.PetID = p.PetID
JOIN Users u ON a.UserID = u.UserID;

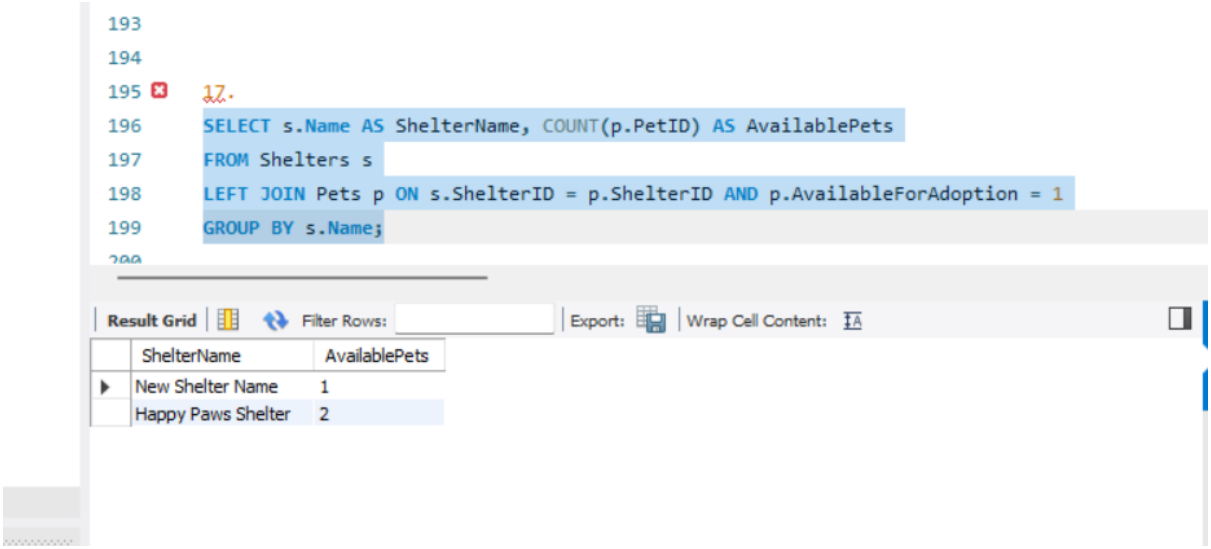


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PetName	AdopterName
Bella	Anita Sharma
Mittens	Ravi Patel

16. 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 AvailablePets
FROM Shelters s
LEFT JOIN Pets p ON s.ShelterID = p.ShelterID AND p.AvailableForAdoption = 1
GROUP BY s.Name;



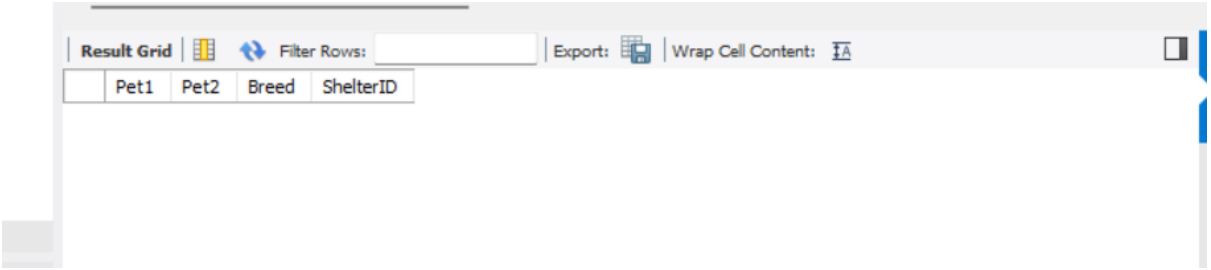
```
193
194
195    17.
196    SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePets
197    FROM Shelters s
198    LEFT JOIN Pets p ON s.ShelterID = p.ShelterID AND p.AvailableForAdoption = 1
199    GROUP BY s.Name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ShelterName	AvailablePets
New Shelter Name	1
Happy Paws Shelter	2

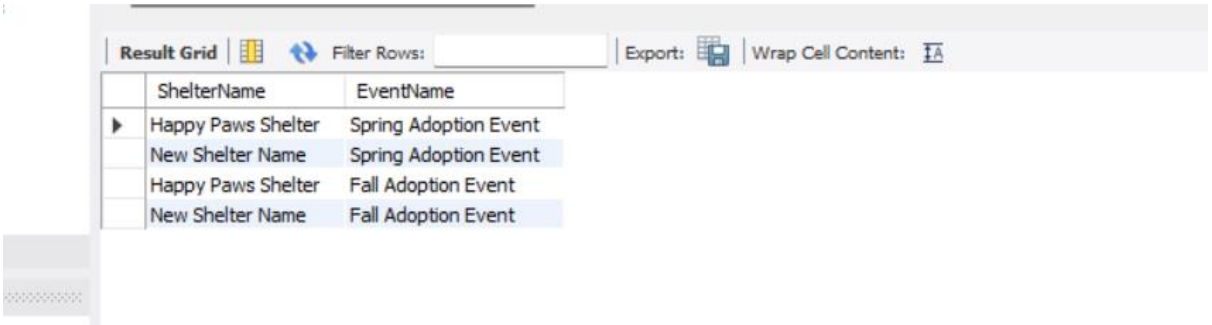
17. Find pairs of pets from the same shelter that have the same breed.

→ SELECT a.Name AS Pet1, b.Name AS Pet2, a.Breed, a.ShelterID
FROM Pets a
JOIN Pets b ON a.ShelterID = b.ShelterID AND a.Breed = b.Breed AND a.PetID < b.PetID;



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

→ SELECT s.Name AS ShelterName, e.EventName
FROM Shelters s
CROSS JOIN AdoptionEvents e;



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

→ SELECT s.Name AS ShelterName, COUNT(a.PetID) AS AdoptedPets
FROM Adoption a
JOIN Pets p ON a.PetID = p.PetID
JOIN Shelters s ON p.ShelterID = s.ShelterID
GROUP BY s.Name
ORDER BY AdoptedPets DESC
LIMIT 1;