Assignment: SQL Database for Pet Adoption Platform

Instructions:

- Submit your completed assignment via GitHub and share the repository link.
- Ensure that your SQL script handles errors such as duplicate tables or missing references.
- Use appropriate primary keys, foreign keys, and constraints for database integrity.

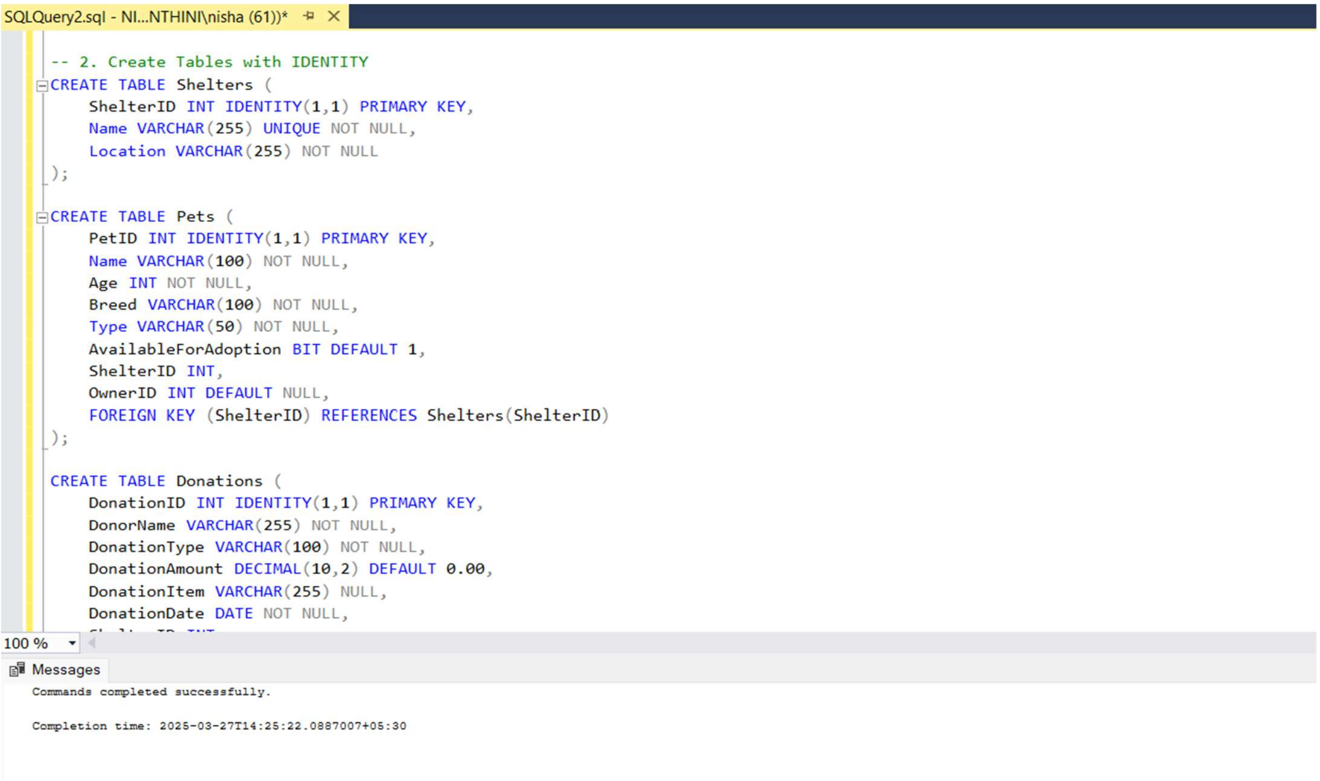
Problem Statement

You are required to design a database schema for a pet adoption platform called **PetCare**. This platform manages pets, shelters, donations, adoption events, and participants.

Tasks

Database Initialization

1. **SQL Schema Creation:**
   - Create a database schema for **PetCare**.
   - Define tables with attributes similar to the ones described below:
     - **Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption)**
     - **Shelters (ShelterID, Name, Location)**
     - **Donations (DonationID, DonorName, DonationType, DonationAmount, DonationItem, DonationDate)**
     - **AdoptionEvents (EventID, EventName, EventDate, Location)**
     - **Participants (ParticipantID, ParticipantName, ParticipantType, EventID as Foreign Key)**



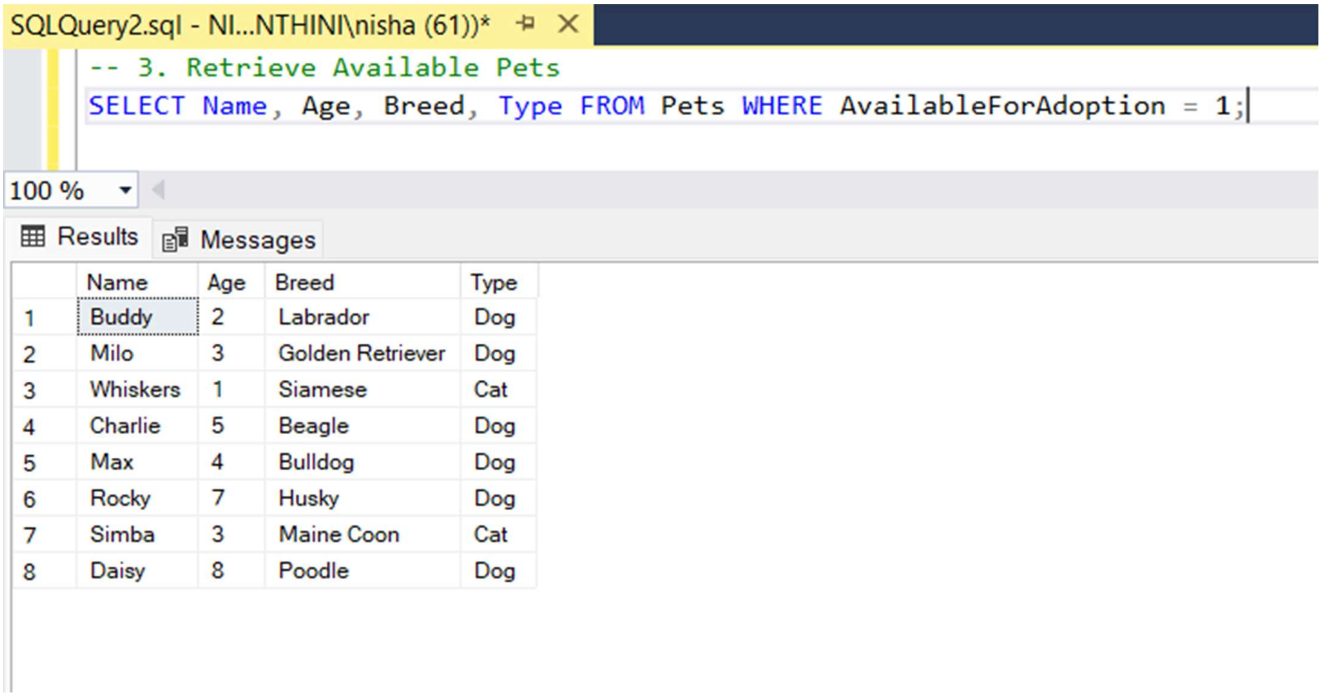
```
-- 2. Create Tables with IDENTITY
CREATE TABLE Shelters (
    ShelterID INT IDENTITY(1,1) PRIMARY KEY,
    Name VARCHAR(255) UNIQUE NOT NULL,
    Location VARCHAR(255) NOT NULL
);

CREATE TABLE Pets (
    PetID INT IDENTITY(1,1) PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Age INT NOT NULL,
    Breed VARCHAR(100) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    AvailableForAdoption BIT DEFAULT 1,
    ShelterID INT,
    OwnerID INT DEFAULT NULL,
    FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID)
);

CREATE TABLE Donations (
    DonationID INT IDENTITY(1,1) PRIMARY KEY,
    DonorName VARCHAR(255) NOT NULL,
    DonationType VARCHAR(100) NOT NULL,
    DonationAmount DECIMAL(10,2) DEFAULT 0.00,
    DonationItem VARCHAR(255) NULL,
    DonationDate DATE NOT NULL,
```

```
Commands completed successfully.

Completion time: 2025-03-27T14:25:22.0887007+05:30
```

2. **Table Constraints:**
    - Define primary and foreign keys.
    - Set constraints for null values and unique attributes where applicable.
    - Ensure the script checks if tables exist before creating them.

SQL Query Challenges

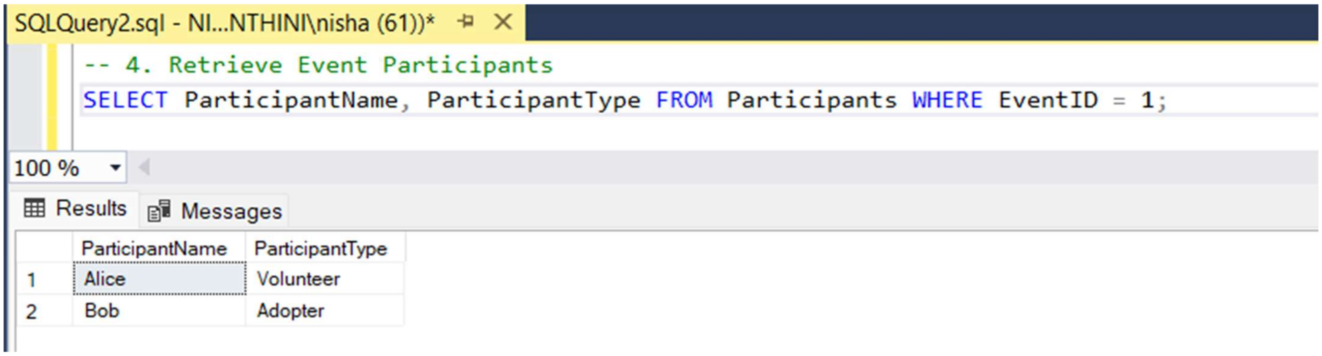
3. **Retrieve Available Pets:**
    - Write an SQL query to list pets available for adoption.
    - Output should include the pet's **Name, Age, Breed, and Type**.



```
-- 3. Retrieve Available Pets
SELECT Name, Age, Breed, Type FROM Pets WHERE AvailableForAdoption = 1;
```

	Name	Age	Breed	Type
1	Buddy	2	Labrador	Dog
2	Milo	3	Golden Retriever	Dog
3	Whiskers	1	Siamese	Cat
4	Charlie	5	Beagle	Dog
5	Max	4	Bulldog	Dog
6	Rocky	7	Husky	Dog
7	Simba	3	Maine Coon	Cat
8	Daisy	8	Poodle	Dog

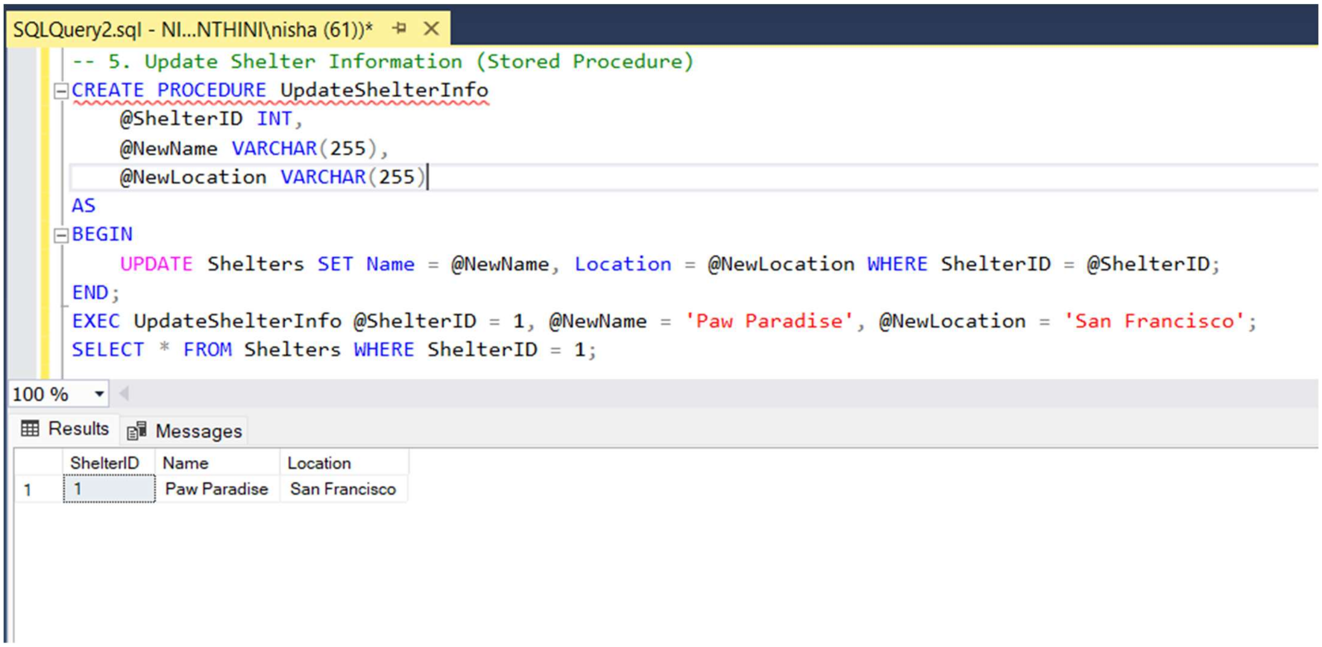
4. **Retrieve Event Participants:**
    - Write an SQL query to list **participant names and types** for a specific event based on EventID.



```
-- 4. Retrieve Event Participants
SELECT ParticipantName, ParticipantType FROM Participants WHERE EventID = 1;
```

	ParticipantName	ParticipantType
1	Alice	Volunteer
2	Bob	Adopter

5. **Update Shelter Information (Stored Procedure):**
    - Create a stored procedure to update a shelter's name and location.
    - The procedure should take ShelterID, NewName, and NewLocation as parameters.

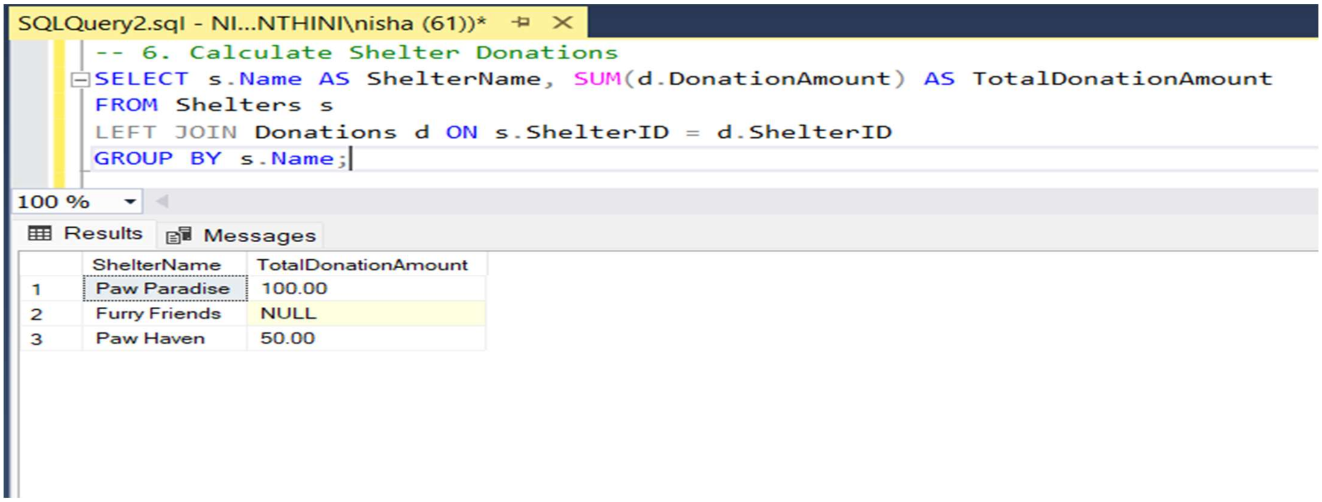


```
-- 5. Update Shelter Information (Stored Procedure)
CREATE PROCEDURE UpdateShelterInfo
    @ShelterID INT,
    @NewName VARCHAR(255),
    @NewLocation VARCHAR(255)
AS
BEGIN
    UPDATE Shelters SET Name = @NewName, Location = @NewLocation WHERE ShelterID = @ShelterID;
END;
EXEC UpdateShelterInfo @ShelterID = 1, @NewName = 'Paw Paradise', @NewLocation = 'San Francisco';
SELECT * FROM Shelters WHERE ShelterID = 1;
```

	ShelterID	Name	Location
1	1	Paw Paradise	San Francisco

6. **Calculate Shelter Donations:**

   - Write an SQL query to calculate the **total donation amount per shelter**.
   - The output should include **Shelter Name and Total Donation Amount**.

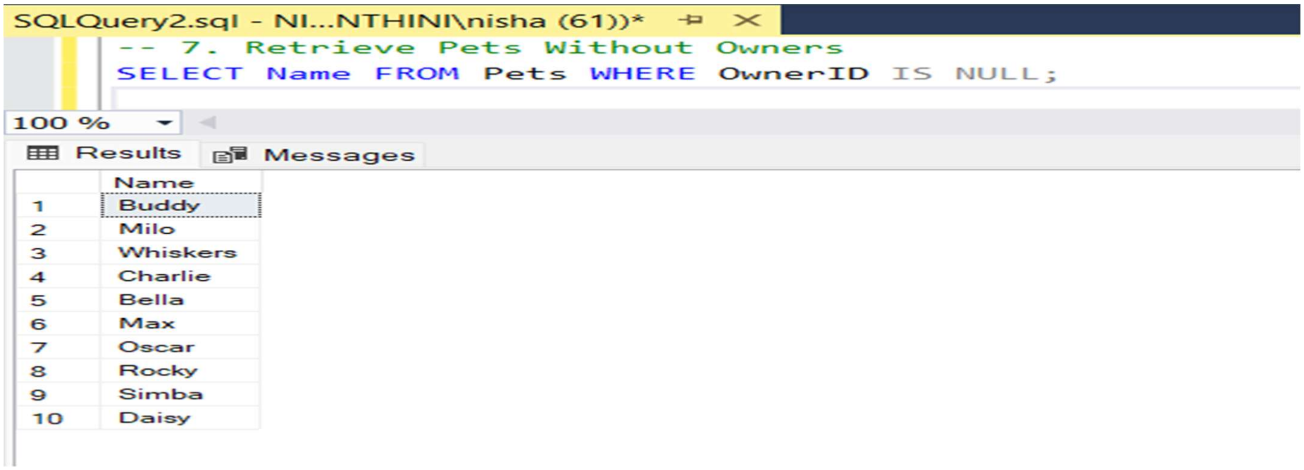


```
-- 6. Calculate Shelter Donations
SELECT s.Name AS ShelterName, SUM(d.DonationAmount) AS TotalDonationAmount
FROM Shelters s
LEFT JOIN Donations d ON s.ShelterID = d.ShelterID
GROUP BY s.Name;
```

	ShelterName	TotalDonationAmount
1	Paw Paradise	100.00
2	Furry Friends	NULL
3	Paw Haven	50.00

7. **Retrieve Pets Without Owners:**

   - Write an SQL query to list all pets that do not have an owner (OwnerID IS NULL).

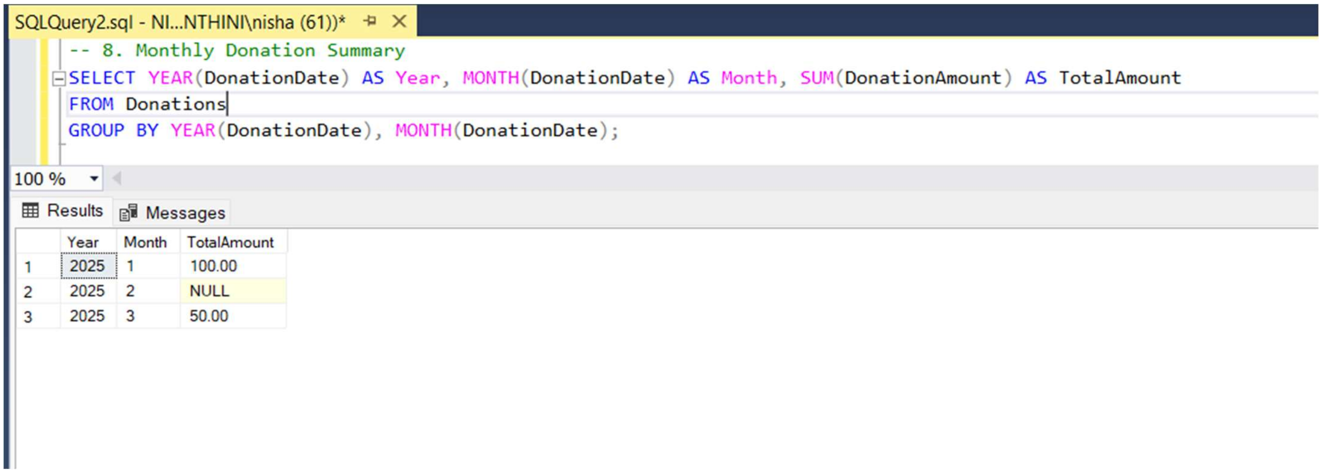


```
-- 7. Retrieve Pets Without Owners
SELECT Name FROM Pets WHERE OwnerID IS NULL;
```

	Name
1	Buddy
2	Milo
3	Whiskers
4	Charlie
5	Bella
6	Max
7	Oscar
8	Rocky
9	Simba
10	Daisy

8. **Monthly Donation Summary:**

   - Write an SQL query to retrieve **total donations per month and year**.

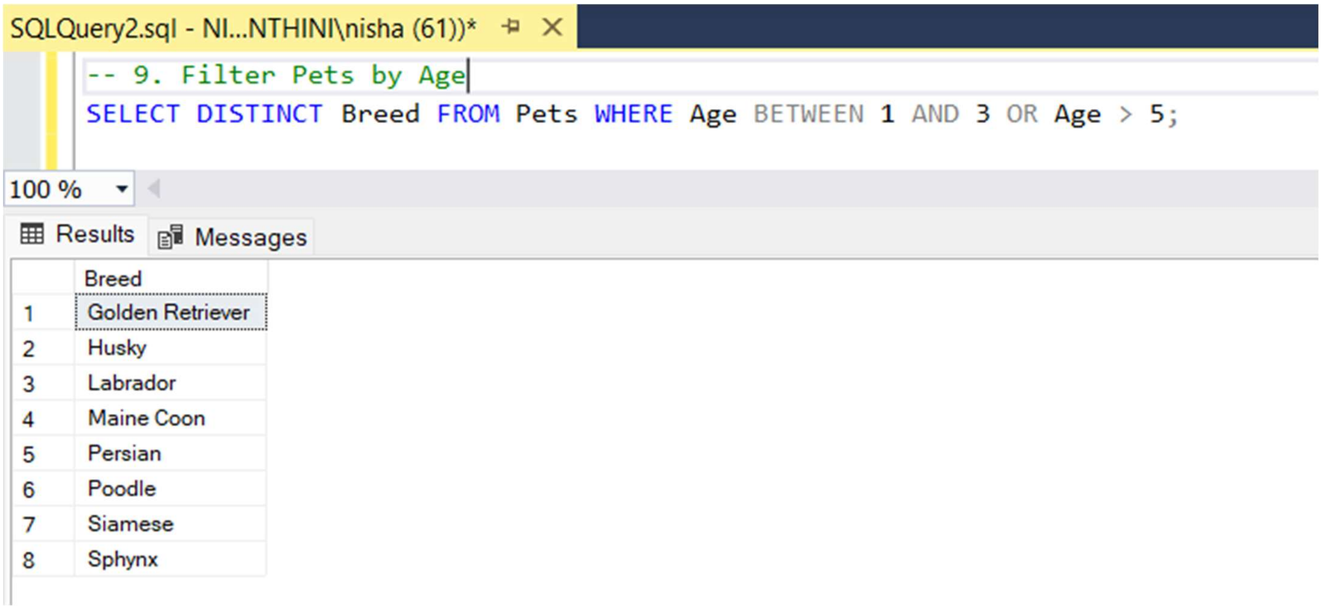


```sql
-- 8. Monthly Donation Summary
SELECT YEAR(DonationDate) AS Year, MONTH(DonationDate) AS Month, SUM(DonationAmount) AS TotalAmount
FROM Donations
GROUP BY YEAR(DonationDate), MONTH(DonationDate);
```

	Year	Month	TotalAmount
1	2025	1	100.00
2	2025	2	NULL
3	2025	3	50.00

9. **Filter Pets by Age:**

   - Retrieve distinct pet breeds where pets are aged **between 1 and 3 years or older than 5 years**.

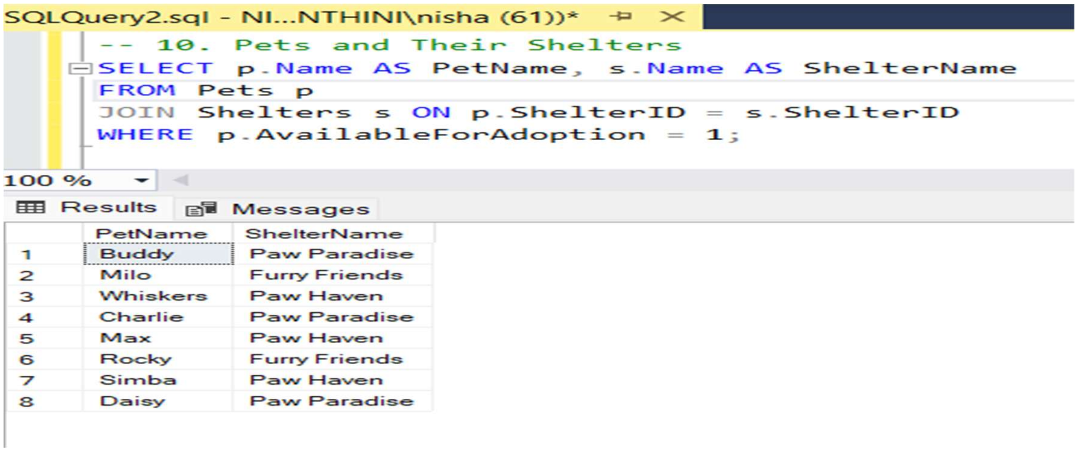


```sql
-- 9. Filter Pets by Age
SELECT DISTINCT Breed FROM Pets WHERE Age BETWEEN 1 AND 3 OR Age > 5;
```

	Breed
1	Golden Retriever
2	Husky
3	Labrador
4	Maine Coon
5	Persian
6	Poodle
7	Siamese
8	Sphynx

10. **Pets and Their Shelters:**

- List all **pets and their respective shelters** where pets are available for adoption.

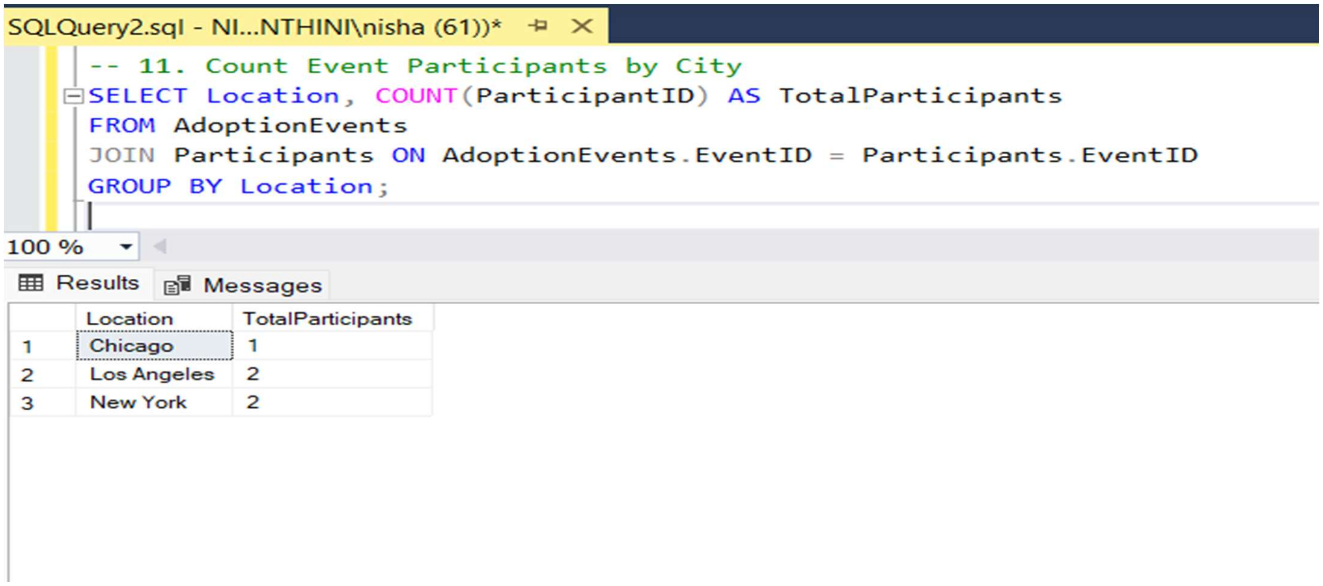


```sql
-- 10. Pets and Their Shelters
SELECT p.Name AS PetName, s.Name AS ShelterName
FROM Pets p
JOIN Shelters s ON p.ShelterID = s.ShelterID
WHERE p.AvailableForAdoption = 1;
```

	PetName	ShelterName
1	Buddy	Paw Paradise
2	Milo	Furry Friends
3	Whiskers	Paw Haven
4	Charlie	Paw Paradise
5	Max	Paw Haven
6	Rocky	Furry Friends
7	Simba	Paw Haven
8	Daisy	Paw Paradise

11. **Count Event Participants by City:**

- Find the total number of **participants in adoption events held in a specific city (e.g., Mumbai)**.

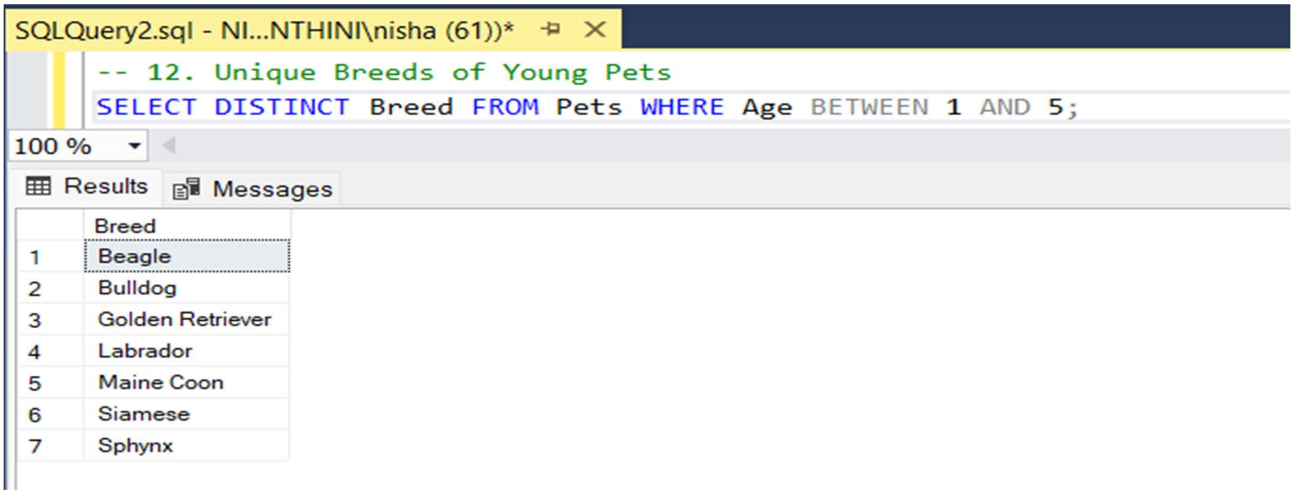


```sql
-- 11. Count Event Participants by City
SELECT Location, COUNT(ParticipantID) AS TotalParticipants
FROM AdoptionEvents
JOIN Participants ON AdoptionEvents.EventID = Participants.EventID
GROUP BY Location;
```

	Location	TotalParticipants
1	Chicago	1
2	Los Angeles	2
3	New York	2

12. **Unique Breeds of Young Pets:**

- Retrieve unique pet breeds for pets aged between **1 and 5 years**.

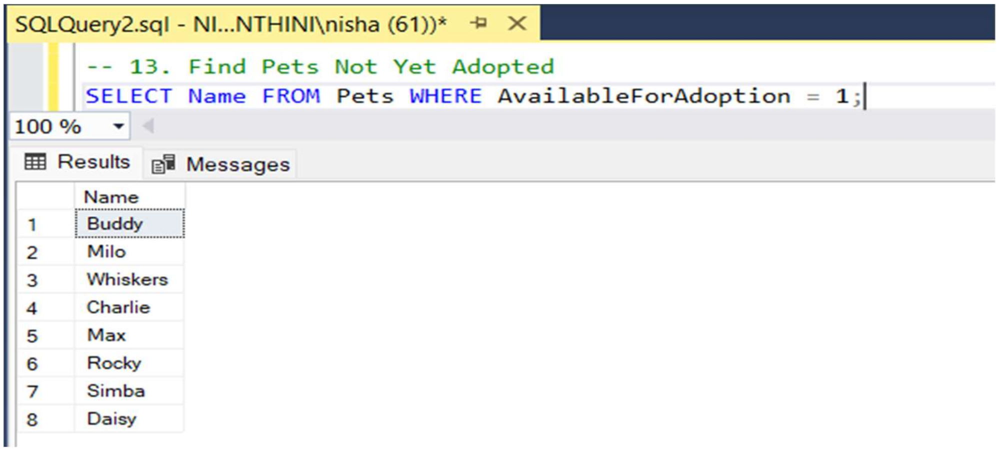


```sql
-- 12. Unique Breeds of Young Pets
SELECT DISTINCT Breed FROM Pets WHERE Age BETWEEN 1 AND 5;
```

	Breed
1	Beagle
2	Bulldog
3	Golden Retriever
4	Labrador
5	Maine Coon
6	Siamese
7	Sphynx

13. **Find Pets Not Yet Adopted:**

- Retrieve a list of pets that **have not been adopted**.

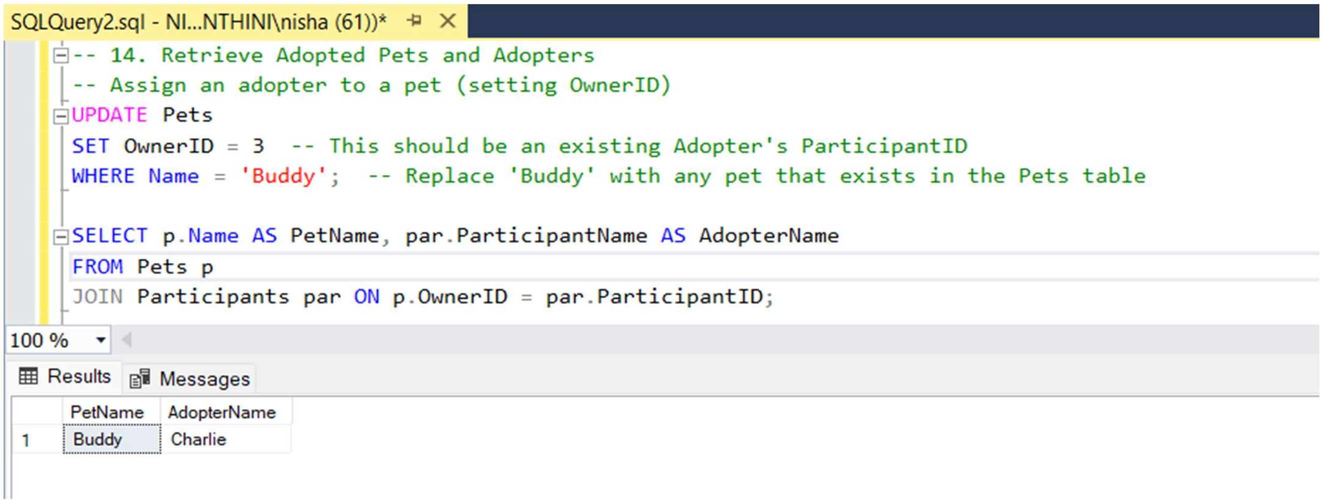


```sql
-- 13. Find Pets Not Yet Adopted
SELECT Name FROM Pets WHERE AvailableForAdoption = 1;
```

	Name
1	Buddy
2	Milo
3	Whiskers
4	Charlie
5	Max
6	Rocky
7	Simba
8	Daisy

14. **Retrieve Adopted Pets and Adopters:**

- Fetch the names of **adopted pets** along with their **adopter's name**.

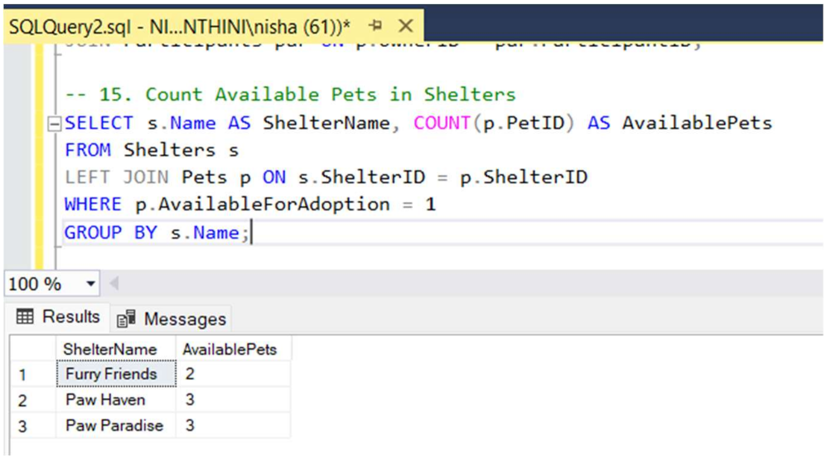


```sql
-- 14. Retrieve Adopted Pets and Adopters
-- Assign an adopter to a pet (setting OwnerID)
UPDATE Pets
SET OwnerID = 3  -- This should be an existing Adopter's ParticipantID
WHERE Name = 'Buddy';  -- Replace 'Buddy' with any pet that exists in the Pets table

SELECT p.Name AS PetName, par.ParticipantName AS AdopterName
FROM Pets p
JOIN Participants par ON p.OwnerID = par.ParticipantID;
```

	PetName	AdopterName
1	Buddy	Charlie

15. **Count Available Pets in Shelters:**

- List **all shelters** along with the **count of pets currently available for adoption** in each shelter.

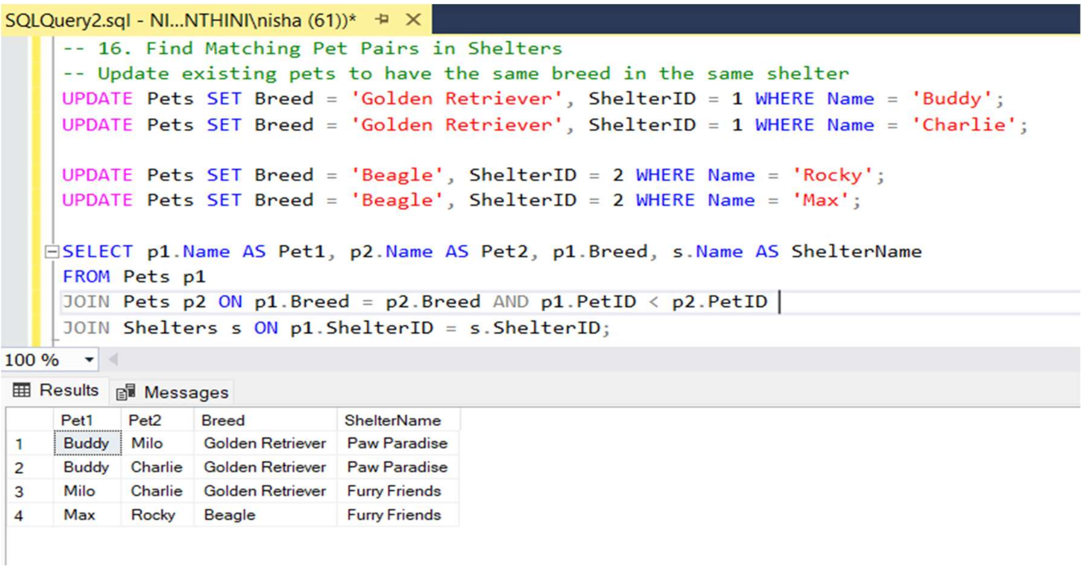


```sql
-- 15. Count Available Pets in Shelters
SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePets
FROM Shelters s
LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
WHERE p.AvailableForAdoption = 1
GROUP BY s.Name;
```

	ShelterName	AvailablePets
1	Furry Friends	2
2	Paw Haven	3
3	Paw Paradise	3

16. **Find Matching Pet Pairs in Shelters:**

- List pairs of pets **from the same shelter** that have the **same breed**.

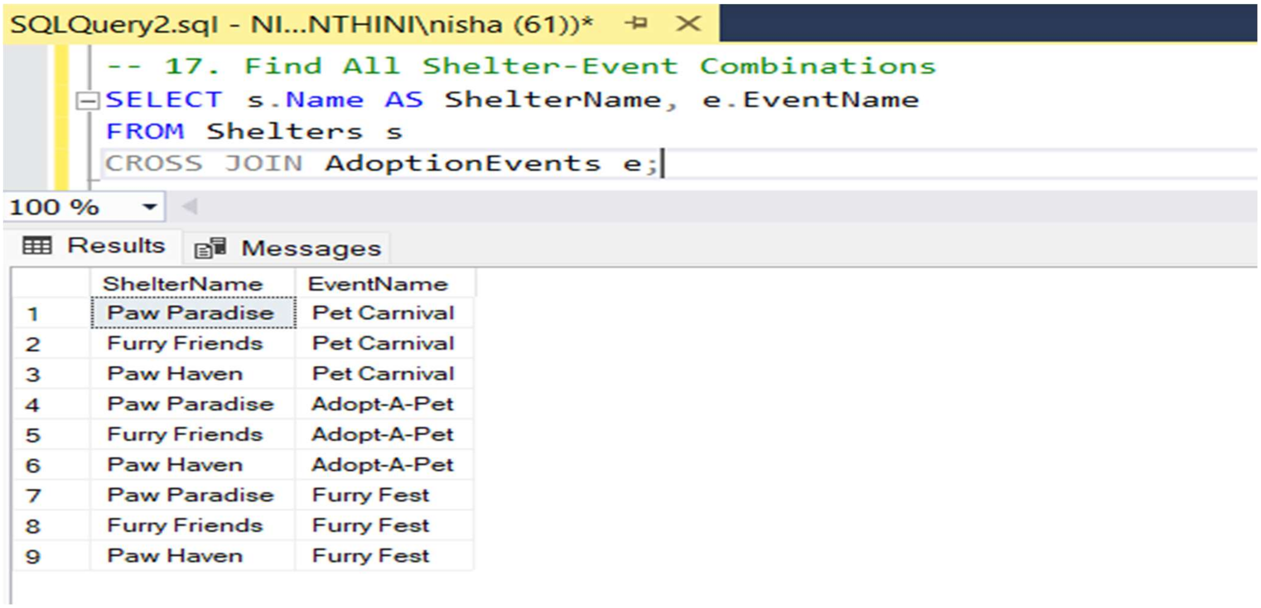


```sql
-- 16. Find Matching Pet Pairs in Shelters
-- Update existing pets to have the same breed in the same shelter
UPDATE Pets SET Breed = 'Golden Retriever', ShelterID = 1 WHERE Name = 'Buddy';
UPDATE Pets SET Breed = 'Golden Retriever', ShelterID = 1 WHERE Name = 'Charlie';

UPDATE Pets SET Breed = 'Beagle', ShelterID = 2 WHERE Name = 'Rocky';
UPDATE Pets SET Breed = 'Beagle', ShelterID = 2 WHERE Name = 'Max';

SELECT p1.Name AS Pet1, p2.Name AS Pet2, 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.ShelterID = s.ShelterID;
```

	Pet1	Pet2	Breed	ShelterName
1	Buddy	Milo	Golden Retriever	Paw Paradise
2	Buddy	Charlie	Golden Retriever	Paw Paradise
3	Milo	Charlie	Golden Retriever	Furry Friends
4	Max	Rocky	Beagle	Furry Friends

17. **Find All Shelter-Event Combinations:**

- Retrieve all possible **combinations of shelters and adoption events**.

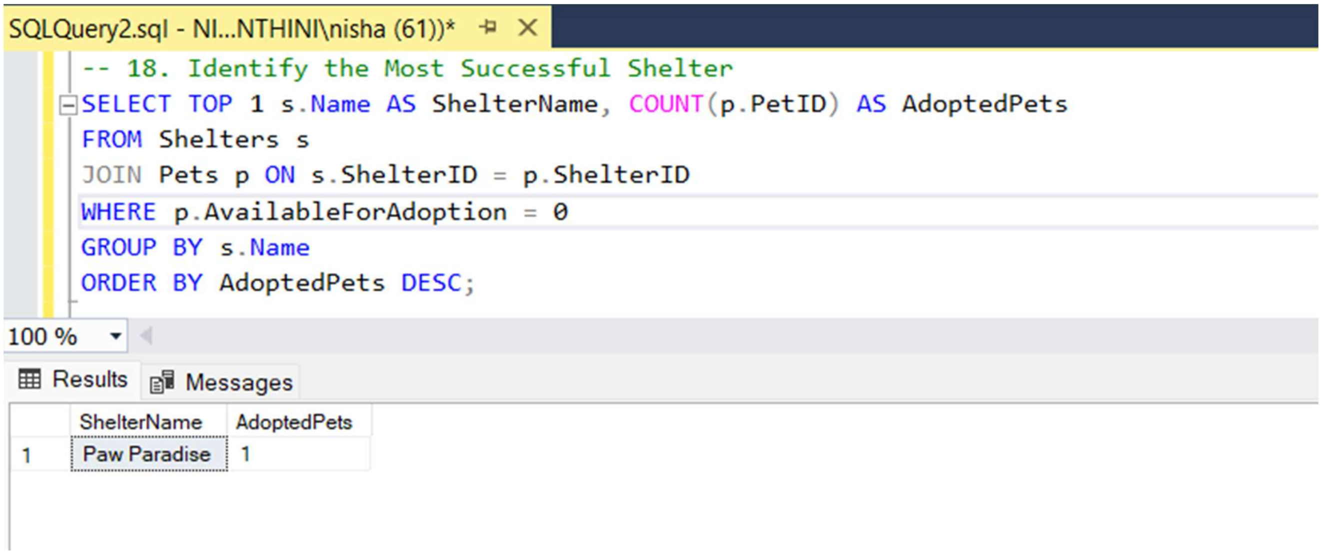


```sql
-- 17. Find All Shelter-Event Combinations
SELECT s.Name AS ShelterName, e.EventName
FROM Shelters s
CROSS JOIN AdoptionEvents e;
```

	ShelterName	EventName
1	Paw Paradise	Pet Carnival
2	Furry Friends	Pet Carnival
3	Paw Haven	Pet Carnival
4	Paw Paradise	Adopt-A-Pet
5	Furry Friends	Adopt-A-Pet
6	Paw Haven	Adopt-A-Pet
7	Paw Paradise	Furry Fest
8	Furry Friends	Furry Fest
9	Paw Haven	Furry Fest

18. **Identify the Most Successful Shelter:**

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



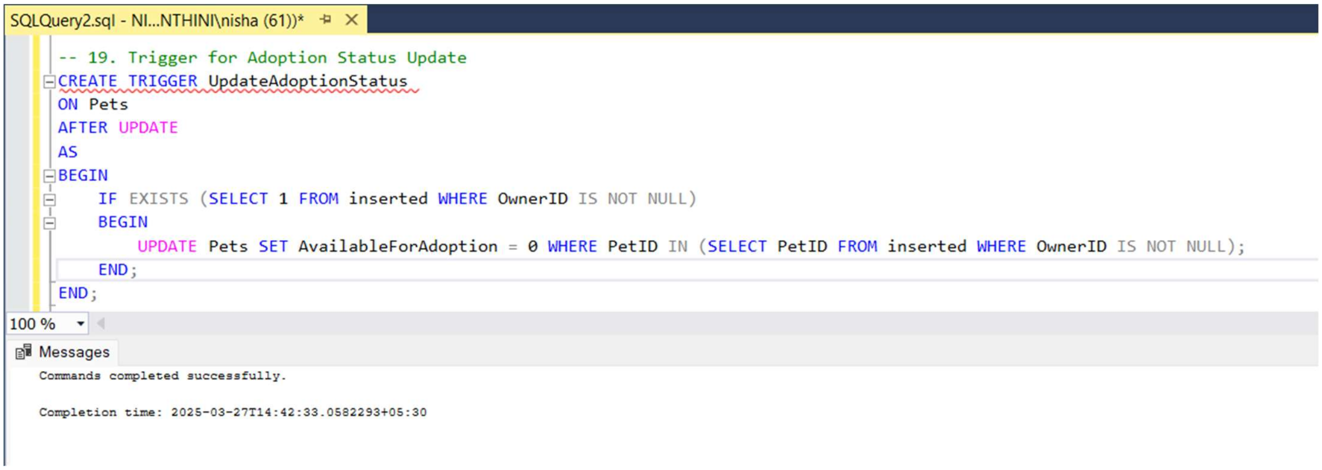
```sql
-- 18. Identify the Most Successful Shelter
SELECT TOP 1 s.Name AS ShelterName, COUNT(p.PetID) AS AdoptedPets
FROM Shelters s
JOIN Pets p ON s.ShelterID = p.ShelterID
WHERE p.AvailableForAdoption = 0
GROUP BY s.Name
ORDER BY AdoptedPets DESC;
```

	ShelterName	AdoptedPets
1	Paw Paradise	1

Bonus Challenges

19. **Trigger for Adoption Status Update:**

- Create a database **trigger** that updates a pet's **AvailableForAdoption** status when it is adopted.

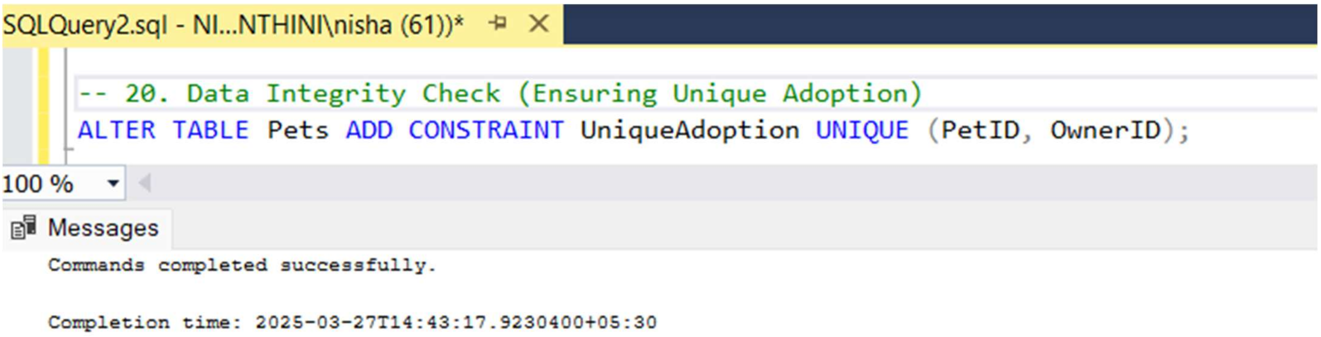


```sql
-- 19. Trigger for Adoption Status Update
CREATE TRIGGER UpdateAdoptionStatus
ON Pets
AFTER UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE OwnerID IS NOT NULL)
    BEGIN
        UPDATE Pets SET AvailableForAdoption = 0 WHERE PetID IN (SELECT PetID FROM inserted WHERE OwnerID IS NOT NULL);
    END;
END;
```

```
Commands completed successfully.

Completion time: 2025-03-27T14:42:33.0582293+05:30
```

20. **Data Integrity Check:**

- Ensure that a pet cannot be adopted twice using a **constraint or validation trigger**.



```sql
-- 20. Data Integrity Check (Ensuring Unique Adoption)
ALTER TABLE Pets ADD CONSTRAINT UniqueAdoption UNIQUE (PetID, OwnerID);
```

```
Commands completed successfully.

Completion time: 2025-03-27T14:43:17.9230400+05:30
```