

TASKS:

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

Query:

```
CREATE DATABASE IF NOT EXISTS petadoptiondatabase;  
usepetadoptiondatabaase
```

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

-- Pets Table

```
CREATE TABLE IF NOT EXISTS Pets (  
    PetID INT PRIMARY KEY,  
    Name VARCHAR(255),  
    Age INT,  
    Breed VARCHAR(255),  
    Type VARCHAR(50),  
    AvailableForAdoption BIT  
);
```

-- Shelters Table

```
CREATE TABLE IF NOT EXISTS Shelters (  
    ShelterID INT PRIMARY KEY,  
    Name VARCHAR(255),  
    Location VARCHAR(255)  
);
```

-- Donations Table

```
CREATE TABLE IF NOT EXISTS Donations (  
    DonationID INT PRIMARY KEY,  
    DonorName VARCHAR(255),  
    DonationType VARCHAR(50),  
    DonationAmount DECIMAL(10, 2),  
    DonationItem VARCHAR(255),  
    DonationDate DATETIME  
);
```

-- AdoptionEvents Table

```
CREATE TABLE IF NOT EXISTS AdoptionEvents (  
    EventID INT PRIMARY KEY,  
    EventName VARCHAR(255),  
    EventDate DATETIME,  
    Location VARCHAR(255)  
);
```

-- Participants Table

```
CREATE TABLE IF NOT EXISTS Participants (  
    ParticipantID INT PRIMARY KEY,  
    ParticipantName VARCHAR(255),  
    ParticipantType VARCHAR(50),  
    EventID INT,  
    FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)  
);
```

3. Define appropriate primary keys, foreign keys, and constraints:

Ans: That was created in the above table

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

Ans: IF NOT EXISTS clause is used in the table and database creation.

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.

query:

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

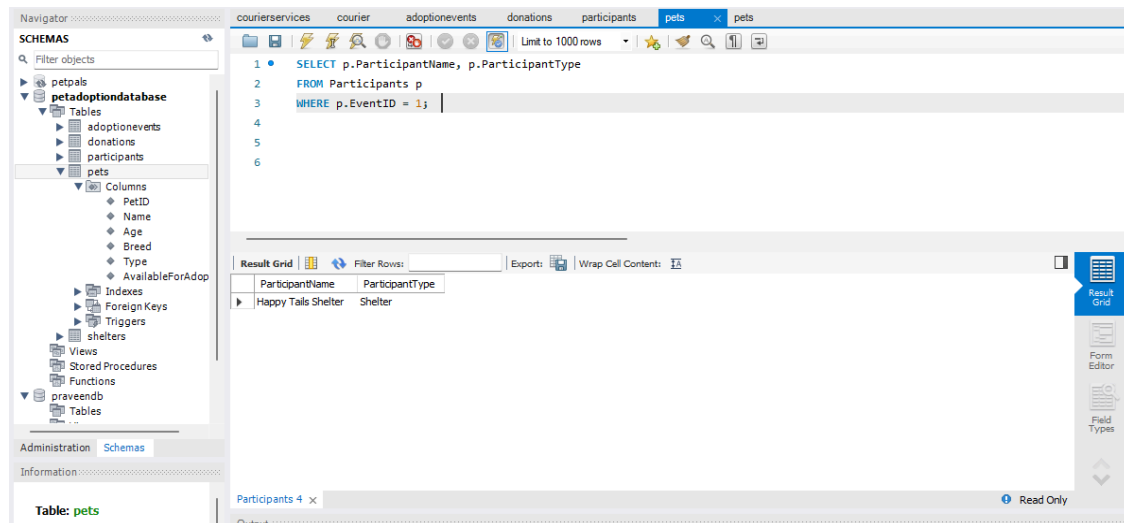
The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'adoptionevents', 'donations', 'participants', and 'pets'. The 'pets' table is selected, showing its columns: PetID, Name, Age, Breed, Type, and AvailableForAdop. The main pane displays the SQL query:
1. SELECT * FROM petadoptiondatabase.pets;
2. SELECT Name, Age, Breed, Type
3. FROM Pets
4. WHERE AvailableForAdoption = 1;
5.
The 'Result Grid' at the bottom shows the results of the query, listing 10 pets with their details.

Name	Age	Breed	Type
Buddy	3	Labrador Retriever	Dog
Milo	2	Siamese	Cat
Luna	1	German Shepherd	Dog
Max	4	Bulldog	Dog
Charlie	5	Maine Coon	Cat
Bella	2	Golden Retriever	Dog
Lucy	3	Persian	Cat
Daisy	2	Dachshund	Dog
Molly	4	Ragdoll	Cat
Oliver	1	Beagle	Dog

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

query:

```
SELECT p.ParticipantName, p.ParticipantType
FROM Participants p
WHERE p.EventID = 1;
```



7. Update Shelter Information Procedure

Query:

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateShelterInfo(IN ShelterID INT, IN
NewName VARCHAR(255), IN NewLocation VARCHAR(255)) BEGIN
UPDATE Shelters SET Name = NewName, Location = NewLocation
WHERE ShelterID = ShelterID; END//
```

```
DELIMITER ; CALL UpdateShelterInfo(1, 'Paris Street', 'paris');
```

```
SELECT * FROM Shelters WHERE ShelterID = 1; Set
sql_safe_updates=0;
```

Navigator: petpals, petadoptiondatabase, Tables, adoptionevents, donations, participants, pets, Columns (PetID, Name, Age, Breed, Type, AvailableForAdop), Indexes, Foreign Keys, Triggers, shelters, Views, Stored Procedures, Functions, praveendb, Tables.

```

1 DELIMITER //
2 CREATE PROCEDURE UpdateShelterInfo(IN ShelterID INT, IN NewName
3   VARCHAR(255), IN NewLocation VARCHAR(255))
4 BEGIN
5   UPDATE Shelters
6   SET Name = NewName, Location = NewLocation
7   WHERE ShelterID = ShelterID;
8 END//
9 DELIMITER ;

```

Result Grid:

ShelterID	Name	Location
1	Paris Street	paris

8. Total Donation Amount per Shelter

Query:

```

SELECT s.Name, COALESCE(SUM(d.DonationAmount), 0) AS
TotalDonation
FROM Shelters s
LEFT JOIN Donations d ON s.ShelterID = d.ShelterID
GROUP BY s.Name;

```

Navigator: petpals, petadoptiondatabase, Tables, adoptionevents, donations, participants, pets, Columns (PetID, Name, Age, Breed, Type, AvailableForAdop), Indexes, Foreign Keys, Triggers, shelters, Views, Stored Procedures, Functions, praveendb, Tables.

```

1 SELECT s.Name, COALESCE(SUM(d.DonationAmount), 0) AS TotalDonation
2 FROM Shelters s
3 LEFT JOIN Donations d ON s.ShelterID = d.ShelterID
4 GROUP BY s.Name;

```

Result Grid:

Name	TotalDonation
Paris Street	0.00

9. Pets Without Owners

Query:

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

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'pets'. The main query editor contains the following SQL query:

```
1 SELECT Name, Age, Breed, Type
2 FROM Pets
3 WHERE AvailableForAdoption = 1;
4
```

Below the query editor, the 'Result Grid' displays the results of the query:

Name	Age	Breed	Type
Buddy	3	Labrador Retriever	Dog
Milo	2	Siamese	Cat
Luna	1	German Shepherd	Dog
Max	4	Bulldog	Dog
Charlie	5	Maine Coon	Cat
Bella	2	Golden Retriever	Dog
Lucy	3	Persian	Cat
Daisy	2	Dachshund	Dog
Molly	4	Ragdoll	Cat
Oliver	1	Beagle	Dog

10. Total Donation per Month-Year

Query:

```
SELECT DATE_FORMAT(DonationDate, '%Y-%m') AS MonthYear,
SUM(DonationAmount) AS TotalDonationAmount
FROM Donations
GROUP BY DATE_FORMAT(DonationDate, '%Y-%m')
```

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'pets'. The main query editor contains the following SQL query:

```
1 SELECT
2 DATE_FORMAT(DonationDate, '%Y-%m') AS MonthYear,
3 SUM(DonationAmount) AS TotalDonationAmount
4 FROM
5 Donations
6 GROUP BY
7 DATE_FORMAT(DonationDate, '%Y-%m')
```

Below the query editor, the 'Result Grid' displays the results of the query:

MonthYear	TotalDonationAmount
2024-04	655.50

At the bottom left, a status bar indicates 'Table: pets'. At the bottom right, a status bar indicates 'Result 5' and 'Read Only'.

11. Distinct Breeds for Pets Aged 1-3 or >5 Years

Query:

```
SELECT DISTINCT Breed
```

```
FROM Pets
```

```
WHERE (Age BETWEEN 1 AND 3) OR Age > 5;
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'petadoptiondatabase' schema with tables: adoptionevents, donations, participants, and pets. The 'pets' table is selected, showing columns: PetID, Name, Age, Breed, Type, and AvailableForAdop. The central pane shows the following SQL query:

```
1 • SELECT DISTINCT Breed, name
2 FROM Pets
3 WHERE (Age BETWEEN 1 AND 3) OR Age > 5;
4
```

The right pane displays the 'Result Grid' with the following data:

Breed	name
Labrador Retriever	Buddy
Siamese	Milo
German Shepherd	Luna
Golden Retriever	Bella
Persian	Lucy
Dachshund	Daisy
Beagle	Oliver

12. Pets and Their Respective Shelters (Available for Adoption)

Query:

```
SELECT p.Name AS PetName, p.Breed, s.Name AS ShelterName
```

```
FROM Pets p
```

```
JOIN Shelters s ON p.ShelterID = s.ShelterID
```

```
WHERE p.AvailableForAdoption = 1;
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'petadoptiondatabase' schema with tables: adoptionevents, donations, participants, and pets. The 'pets' table is selected, showing columns: PetID, Name, Age, Breed, Type, and AvailableForAdop. The central pane shows the following SQL query:

```
1 • SELECT p.Name AS PetName, p.Breed
2
3 JOIN Shelters s ON p.ShelterID = s.ShelterID
4 WHERE p.AvailableForAdoption = 1;
5
```

The right pane displays the 'Result Grid' with the following data:

PetName	Breed
Buddy	Labrador Retriever
Milo	Siamese
Luna	German Shepherd
Max	Bulldog
Charlie	Maine Coon
Bella	Golden Retriever
Lucy	Persian
Daisy	Dachshund
Molly	Ragdoll
Oliver	Beagle

13. Total Participants in Events by City Query:

```
SELECT COUNT(p.ParticipantID) AS TotalParticipants
FROM Participants p
JOIN AdoptionEvents ae ON p.EventID = ae.EventID
JOIN Shelters s ON s.ShelterID = p.ParticipantID
WHERE s.Location = 'Chennai';
```

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'participants' and 'shelters'. The main query editor displays the following SQL query:

```
1 SELECT COUNT(p.ParticipantID) AS TotalParticipants
2 FROM Participants p
3 JOIN AdoptionEvents ae ON p.EventID = ae.EventID
4 JOIN Shelters s ON s.ShelterID = p.ParticipantID
5 WHERE s.Location = 'Chennai';
6
```

Below the query editor, the 'Result Grid' shows the execution results:

TotalParticipants
0

The interface also includes a sidebar with options like 'Result Grid', 'Form Editor', and 'Field Types'.

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

query:

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

The screenshot shows the same database management tool interface. The main query editor displays the following SQL query:

```
1 SELECT DISTINCT Breed
2 FROM Pets
3 WHERE Age BETWEEN 1 AND 5;
4
```

Below the query editor, the 'Result Grid' shows the execution results:

Breed
Labrador Retriever
Siamese
German Shepherd
Bulldog
Maine Coon
Golden Retriever
Persian
Dachshund
Ragdoll
Beagle

The interface also includes a sidebar with options like 'Result Grid', 'Form Editor', and 'Field Types'.

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

```
SELECT p.Name AS PetName, p.Age, p.Breed, p.Type
FROM Pets p
LEFT JOIN Owners o ON p.PetID = o.PetID
WHERE o.OwnerID IS NULL OR o.OwnerID = '';
```

The screenshot shows a database management tool interface. On the left is a 'SCHEMAS' tree with 'petadoptiondatabase' expanded, showing tables like 'adoptionevents', 'donations', 'participants', 'pets', and 'shelters'. The 'pets' table is selected. The main area displays a SQL query:

```
1 SELECT p.Name AS PetName, p.Age, p.Breed, p.Type
2 FROM Pets p
3 LEFT JOIN Owners o ON p.PetID = o.PetID
4 WHERE o.OwnerID IS NULL OR o.OwnerID = '';
5
```

Below the query is a 'Result Grid' showing the results of the query:

PetName	Age	Breed	Type
Milo	2	Siamese	Cat
Max	4	Bulldog	Dog
Bella	2	Golden Retriever	Dog
Daisy	2	Dachshund	Dog
Oliver	1	Beagle	Dog

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

Query:

```
SELECT p.Name AS PetName, p.Age, p.Breed, p.Type, o.name as
adopter_name
From Pets p
LEFT JOIN Owners o ON p.PetID = o.PetID
```

The screenshot shows the same database management tool interface. The 'owners' table is now selected in the 'SCHEMAS' tree. The SQL query is updated to include the adopter's name:

```
1 SELECT p.Name AS PetName, p.Age, p.Breed, p.Type, o.name as adopter_name
2 FROM Pets p
3 LEFT JOIN Owners o ON p.PetID = o.PetID
4
```

The 'Result Grid' shows the results of this query:

PetName	Age	Breed	Type	adopter_name
Buddy	3	Labrador Retriever	Dog	John Doe
Milo	2	Siamese	Cat	NULL
Luna	1	German Shepherd	Dog	Jane Smith
Max	4	Bulldog	Dog	NULL
Charlie	5	Maine Coon	Cat	Emily Johnson
Bella	2	Golden Retriever	Dog	NULL
Lucy	3	Persian	Cat	Michael Brown
Daisy	2	Dachshund	Dog	NULL
Molly	4	Ragdoll	Cat	Sophia Wilson
Oliver	1	Beagle	Dog	NULL

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

Query:

```
SELECT s.Name, COUNT(p.PetID) AS AvailablePetsCount
FROM Shelters s
LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
WHERE p.AvailableForAdoption = 1 OR p.AvailableForAdoption IS NULL
GROUP BY s.Name;
```

The screenshot shows a database management interface with a 'petadoptiondatabase' schema. The 'shelters' table is selected in the left pane. The main editor displays the following SQL query:

```
1
2 SELECT s.Name, COUNT(p.PetID) AS AvailablePetsCount
3 FROM Shelters s
4 LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
5 WHERE p.AvailableForAdoption = 1 OR p.AvailableForAdoption IS NULL
6 GROUP BY s.Name;
7
8
```

Below the query editor, the 'Result Grid' shows the following data:

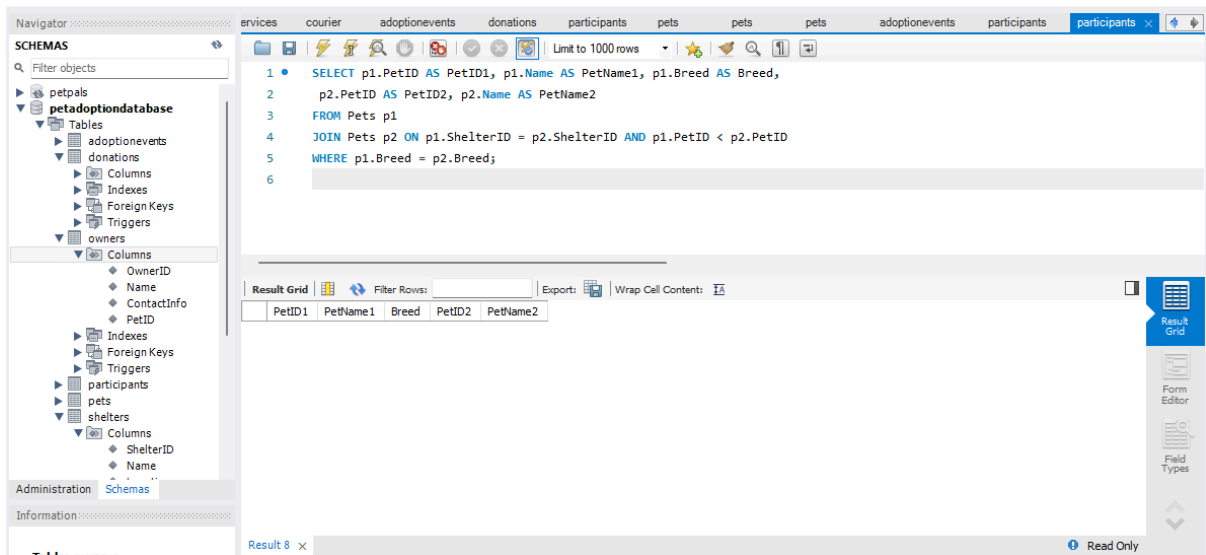
Name	AvailablePetsCount
Paris Street	10

The interface also includes a 'Navigator' pane on the left showing the database schema, a 'Filter Objects' search bar, and a 'Read Only' status indicator at the bottom right.

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

Query:

```
SELECT p1.PetID AS PetID1, p1.Name AS PetName1, p1.Breed AS
Breed, p2.PetID AS PetID2, p2.Name AS PetName2 FROM Pets p1
JOIN Pets p2 ON p1.ShelterID = p2.ShelterID AND p1.PetID < p2.PetID
WHERE p1.Breed = p2.Breed;
```



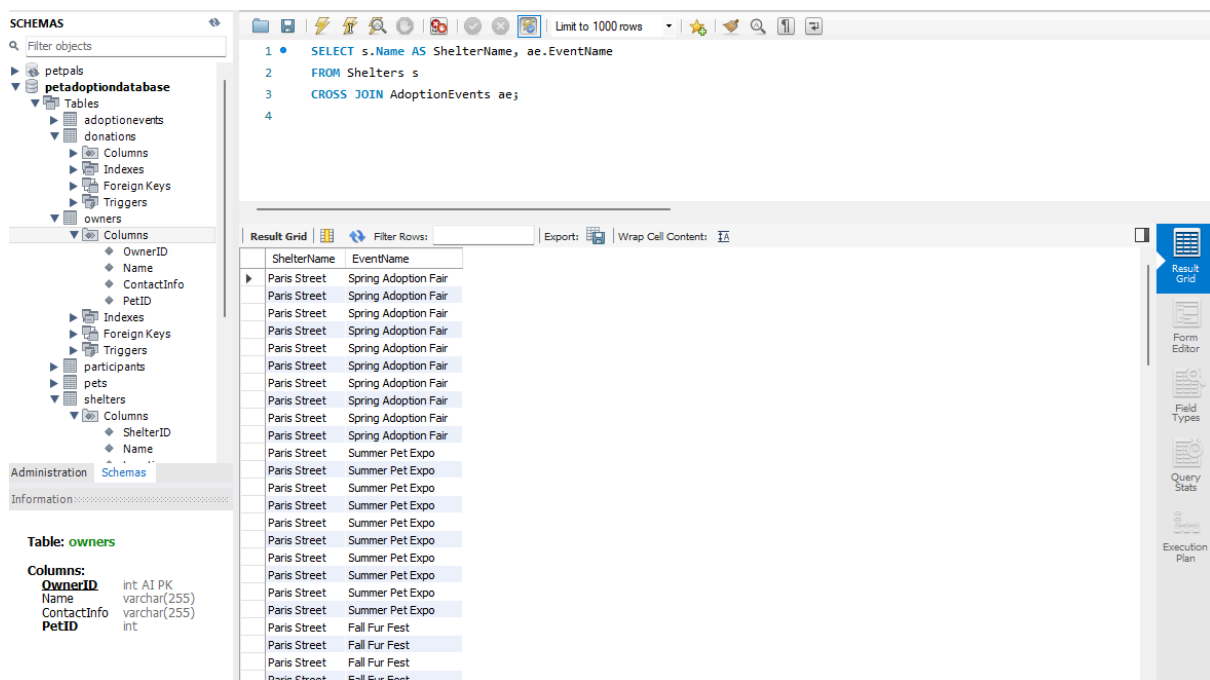
19. List all possible combinations of shelters and adoption events.

Query:

```

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

```



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

Query:

```
SELECT s.Name, COUNT(*) AS Count
FROM Shelters s
JOIN Pets p ON s.ShelterID = p.ShelterID
WHERE p.AvailableForAdoption = 0
GROUP BY s.Name
```

The screenshot shows a database management interface with a sidebar on the left displaying a tree view of database objects. The main area contains a SQL query editor with the following query:

```
1 • SELECT s.Name , COUNT(*) AS Count
2 FROM Shelters s
3 JOIN Pets p ON s.ShelterID = p.ShelterID
4 WHERE p.AvailableForAdoption = 0
5 GROUP BY s.Name
```

Below the query editor is a "Result Grid" tab. The grid has two columns: "Name" and "Count". The grid is currently empty. The interface also includes a top menu bar with various database-related options and a bottom status bar indicating "Result 12" and "Read Only" mode.