Coding Challenges - PetPals, The Pet Adoption Platform

## Task

### 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 shelters (

shelter\_id INT PRIMARY KEY IDENTITY(1,1),

name VARCHAR(100) NOT NULL,

location VARCHAR(255) NOT NULL

);

CREATE TABLE pets (

pet\_id INT PRIMARY KEY IDENTITY(1,1),

name VARCHAR(100) NOT NULL,

age INT,

breed VARCHAR(100),

type VARCHAR(50),

available\_for\_adoption BIT DEFAULT 1,

owner\_id INT NULL

);

CREATE TABLE donations (

donation\_id INT PRIMARY KEY IDENTITY(1,1),

donor\_name VARCHAR(100) NOT NULL,

donation\_type VARCHAR(50),

donation\_amount DECIMAL(10, 2),

donation\_item VARCHAR(100),

donation\_date DATETIME,

shelter\_id INT,

FOREIGN KEY (shelter\_id) REFERENCES shelters(shelter\_id)

);

CREATE TABLE adoption\_events (

event\_id INT PRIMARY KEY IDENTITY(1,1),

event\_name VARCHAR(100) NOT NULL,

event\_date DATETIME NOT NULL,

location VARCHAR(255) NOT NULL

);

CREATE TABLE participants (

participant\_id INT PRIMARY KEY IDENTITY(1,1),

participant\_name VARCHAR(100),

participant\_type VARCHAR(50),

event\_id INT,

FOREIGN KEY (event\_id) REFERENCES adoption\_events(event\_id)

);

INSERT INTO shelters (name, location) VALUES

('PawCare', 'Chennai'),

('SafeHome', 'Mumbai'),

('PetMitra', 'Delhi'),

('PawPoint', 'Bangalore'),

('PetNest', 'Hyderabad'),

('StrayHelp', 'Kolkata'),

('LoveTails', 'Pune'),

('Home4Pets', 'Jaipur'),

('ResQZone', 'Lucknow'),

('PawShelter', 'Coimbatore');

INSERT INTO pets (name, age, breed, type, available\_for\_adoption, owner\_id) VALUES

('Tommy', 2, 'Labrador', 'Dog', 1, NULL),

('Simba', 1, 'Persian', 'Cat', 1, NULL),

('Rani', 3, 'Beagle', 'Dog', 0, 1),

('Moti', 4, 'Siamese', 'Cat', 1, NULL),

('Sheru', 5, 'Pug', 'Dog', 1, NULL),

('Milo', 2, 'Pomeranian', 'Dog', 1, NULL),

('Lalu', 1, 'Indie', 'Dog', 0, 2),

('Tara', 3, 'Tabby', 'Cat', 1, NULL),

('Chiku', 2, 'Indie', 'Cat', 1, NULL),

('Bunny', 4, 'Golden Retriever', 'Dog', 1, NULL);

INSERT INTO adoption\_events (event\_name, event\_date, location) VALUES

('Chennai Pets Day', '2025-07-01 10:00:00', 'Chennai'),

('Mumbai Pet Mela', '2025-07-05 11:00:00', 'Mumbai'),

('Delhi AdoptFest', '2025-07-10 09:00:00', 'Delhi'),

('Bangalore Pet Drive', '2025-07-15 10:30:00', 'Bangalore'),

('Hyd Pet Meet', '2025-07-20 14:00:00', 'Hyderabad'),

('Kolkata PawDay', '2025-07-22 10:00:00', 'Kolkata'),

('Pune FurFriends', '2025-07-25 13:00:00', 'Pune'),

('Jaipur Pet Fest', '2025-07-28 15:00:00', 'Jaipur'),

('Lucknow Pet Fair', '2025-08-01 10:00:00', 'Lucknow'),

('Coimbatore PetPal', '2025-08-05 12:00:00', 'Coimbatore');

INSERT INTO donations (donor\_name, donation\_type, donation\_amount, donation\_item, donation\_date, shelter\_id) VALUES

('Amit', 'Cash', 500.00, NULL, '2025-06-01 10:00:00', 1),

('Sita', 'Item', NULL, 'Dog Food', '2025-06-02 11:00:00', 2),

('Raj', 'Cash', 750.00, NULL, '2025-06-03 09:30:00', 3),

('Neha', 'Item', NULL, 'Toys', '2025-06-04 12:00:00', 4),

('Ravi', 'Cash', 1000.00, NULL, '2025-06-05 14:00:00', 5),

('Anu', 'Item', NULL, 'Blanket', '2025-06-06 13:00:00', 6),

('Karan', 'Cash', 1200.00, NULL, '2025-06-07 16:00:00', 7),

('Pooja', 'Item', NULL, 'Collars', '2025-06-08 10:30:00', 8),

('Meena', 'Cash', 300.00, NULL, '2025-06-09 15:00:00', 9),

('Vijay', 'Cash', 600.00, NULL, '2025-06-10 11:00:00', 10);

INSERT INTO participants (participant\_name, participant\_type, event\_id) VALUES

('PawCare', 'Shelter', 1),

('SafeHome', 'Shelter', 2),

('PetMitra', 'Shelter', 3),

('Asha', 'Adopter', 1),

('Raju', 'Adopter', 2),

('Geeta', 'Adopter', 3),

('StrayHelp', 'Shelter', 4),

('Mohan', 'Adopter', 4),

('LoveTails', 'Shelter', 5),

('Kiran', 'Adopter', 5);

### 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 petsWHERE available\_for\_adoption = 1;

### 6.Write an SQL query that retrieves the names of participants (shelters and adopters) registered or 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.participant\_name, p.participant\_type,a.event\_name FROM participants

INNER JOIN adoption\_events ON p.event\_id =a.event\_id

WHERE p.event\_id = 101;

### 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.

Includes procedure topic so skipped

### 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 shelter\_name,ISNULL(SUM(d.donation\_amount), 0) AS total\_donation

FROM shelters s LEFT JOIN donations d ON s.shelter\_id = d.shelter\_id

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 name, age, breed, typeFROM pets

WHERE owner\_id IS NULL;

### 10Write 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(donation\_date) AS donation\_year,

MONTH(donation\_date) AS donation\_month,

SUM(donation\_amount) AS total\_donation

FROM donations GROUP BY YEAR(donation\_date), MONTH(donation\_date)

ORDER BY YEAR(donation\_date), MONTH(donation\_date);

### 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 >= 1 AND age <= 3) OR (age > 5);

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

ALTER TABLE pets

ADD shelter\_id INT;

ALTER TABLE pets

ADD CONSTRAINT FK\_Pets\_Shelters

FOREIGN KEY (shelter\_id) REFERENCES shelters(shelter\_id);

select \* from shelters

select \* from pets

UPDATE pets SET shelter\_id = 1 WHERE pet\_id IN (1, 2, 3);

UPDATE pets SET shelter\_id = 2 WHERE pet\_id IN (4, 5, 6);

UPDATE pets SET shelter\_id = 3 WHERE pet\_id IN (7, 8, 9, 10);

SELECT p.name AS pet\_name,p.breed, p.type, s.name AS shelter\_name, s.location

FROM pets p JOIN shelters s ON p.shelter\_id = s.shelter\_id

WHERE p.available\_for\_adoption = 1;

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

SELECT COUNT(\*) AS total\_participants

FROM participants p

JOIN adoption\_events a ON p.event\_id = a.event\_id

JOIN shelters s ON a.location = s.location

WHERE s.location = '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 electing their information from the 'Pet' table.

CREATE TABLE adoption (

adoption\_id INT PRIMARY KEY IDENTITY(1,1),

pet\_id INT,

adopter\_name VARCHAR(100),

adoption\_date DATE,

FOREIGN KEY (pet\_id) REFERENCES pets(pet\_id)

);

INSERT INTO adoption (pet\_id, adopter\_name, adoption\_date)

VALUES

(2, 'Ravi Kumar', '2024-08-01'),

(4, 'Priya Sharma', '2024-09-10'),

(6, 'Anil Verma', '2025-01-05');

SELECT \*FROM pets WHERE pet\_id NOT IN (SELECT pet\_id FROM adoption);

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

CREATE TABLE users (

user\_id INT PRIMARY KEY IDENTITY(1,1),

name VARCHAR(100) NOT NULL,

email VARCHAR(100),

phone VARCHAR(20)

);

INSERT INTO users (name, email, phone) VALUES

('Aarav Sharma', 'aarav.sharma@example.com', '9876543210'),

('Anaya Iyer', 'anaya.iyer@example.com', '9898989898'),

('Vihaan Mehta', 'vihaan.mehta@example.com', '9812345678'),

('Isha Reddy', 'isha.reddy@example.com', '9123456780'),

('Aditya Verma', 'aditya.verma@example.com', '9001234567'),

('Kavya Nair', 'kavya.nair@example.com', '9786543210'),

('Arjun Das', 'arjun.das@example.com', '9654321870'),

('Saanvi Rao', 'saanvi.rao@example.com', '9876501234'),

('Dev Patel', 'dev.patel@example.com', '9765432180'),

('Meera Singh', 'meera.singh@example.com', '9543218765');

select \* from users

select \* from adoption

ALTER TABLE adoption ADD user\_id INT;

SELECT p.name AS pet\_name,a.adopter\_name

FROM adoption a JOIN pets p ON a.pet\_id = p.pet\_id;

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

SELECT s.name AS shelter\_name, COUNT(p.pet\_id) AS available\_pet\_count

FROM shelters s LEFT JOIN pets p ON s.shelter\_id = p.shelter\_id AND p.available\_for\_adoption = 1 GROUP BY s.name;

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

SELECT p1.name AS pet1,p2.name AS pet2, p1.breed,s.name AS shelter\_name

FROM pets p1 JOIN pets p2 ON p1.shelter\_id = p2.shelter\_id AND

p1.breed = p2.breed AND p1.pet\_id < p2.pet\_id

JOIN shelters s ON p1.shelter\_id = s.shelter\_id;

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

SELECT s.name AS shelter\_name,e.event\_name

FROM shelters s CROSS JOIN adoption\_events e;

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

SELECT TOP 1 s.name AS shelter\_name, COUNT(a.pet\_id) AS adopted\_pet\_count

FROM adoption a JOIN pets p ON a.pet\_id = p.pet\_id

JOIN shelters s ON p.shelter\_id = s.shelter\_id

GROUP BY s.name ORDER BY adopted\_pet\_count DESC;

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

IF EXISTS (SELECT name FROM sys.databases WHERE name = 'petpals')

DROP DATABASE petpals;

CREATE DATABASE petpals;

USE petpals;