**MySQL Coding Challenge(19.06.2025)**

**Problem Statement:**

Create SQL Schema from the pet and user class, use the class attributes for table column names.

SQL Schema:

**Table: Pets Attributes:**

• PetID (Primary Key, int): Unique identifier for each pet.

• Name (string): The name of the pet.

• Age (int): The age of the pet.

• Breed (string): The breed of the pet.

• Type (string): Type of pet (e.g., "Dog," "Cat").

• AvailableForAdoption (bit): Indicates whether the pet is available for adoption (0 for not available, 1 for available).

**Table: Shelters Attributes:**

• ShelterID (Primary Key, int): Unique identifier for each shelter.

• Name (string): The name of the shelter.

• Location (string): The location or address of the shelter.

**Table: Donations Attributes:**

• DonationID (Primary Key, int): Unique identifier for each donation.

• DonorName (string): The name of the donor.

• DonationType (string): Type of donation (e.g., "Cash," "Item").

• DonationAmount (decimal): The amount donated (for cash donations).

• DonationItem (string): The type of item donated (for item donations).

• DonationDate (datetime): Date and time of the donation.

**Table: AdoptionEvents Attributes:**

• EventID (Primary Key, int): Unique identifier for each adoption event.

• EventName (string): The name or title of the event.

• EventDate (datetime): Date and time of the event.

• Location (string): The location or venue of the event.

**Table: Participants Attributes:**

• ParticipantID (Primary Key, int): Unique identifier for each participant.

• ParticipantName (string): The name of the participant (shelter or adopter).

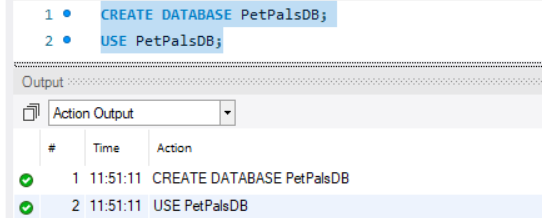
• ParticipantType (string): Type of participant (e.g., "Shelter," "Adopter").

• EventID (Foreign Key, int): References the EventID of the associated adoption event (if applicable).

**Q1**. Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals”.

CREATE DATABASE PetPalsDB;

USE PetPalsDB;



**Q2**. Create tables for pets, shelters, donations, adoption events, and participants.

-- Shelters Table

CREATE TABLE Shelters (

ShelterID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Location VARCHAR(255) NOT NULL

);

-- Users Table (for Adopters)

CREATE TABLE Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL

);

-- Pets Table

CREATE TABLE Pets (

PetID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Age INT,

Breed VARCHAR(100),

Type VARCHAR(50),

AvailableForAdoption BIT,

ShelterID INT,

OwnerID INT,

FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID),

FOREIGN KEY (OwnerID) REFERENCES Users(UserID)

);

-- Donations Table

CREATE TABLE Donations (

DonationID INT PRIMARY KEY AUTO\_INCREMENT,

DonorName VARCHAR(100),

DonationType VARCHAR(50),

DonationAmount DECIMAL(10,2),

DonationItem VARCHAR(100),

DonationDate DATETIME,

ShelterID INT,

FOREIGN KEY (ShelterID) REFERENCES Shelters(ShelterID)

);

-- Adoption Events Table

CREATE TABLE AdoptionEvents (

EventID INT PRIMARY KEY AUTO\_INCREMENT,

EventName VARCHAR(100),

EventDate DATETIME,

Location VARCHAR(255)

);

-- Participants Table

CREATE TABLE Participants (

ParticipantID INT PRIMARY KEY AUTO\_INCREMENT,

ParticipantName VARCHAR(100),

ParticipantType VARCHAR(50),

EventID INT,

FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)

);

-- Adoption Table

CREATE TABLE Adoption (

AdoptionID INT PRIMARY KEY AUTO\_INCREMENT,

PetID INT,

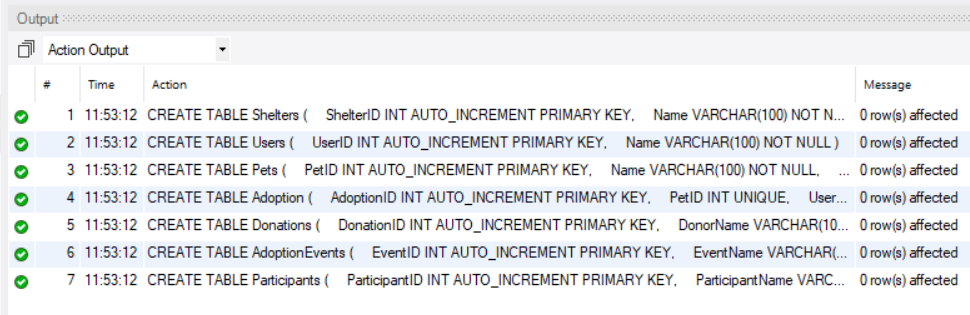
AdopterID INT,

AdoptionDate DATETIME,

FOREIGN KEY (PetID) REFERENCES Pets(PetID),

FOREIGN KEY (AdopterID) REFERENCES Users(UserID)

);



**Q3**. Define appropriate primary keys, foreign keys, and constraints.

SELECT

TABLE\_NAME,

COLUMN\_NAME,

CONSTRAINT\_NAME,

REFERENCED\_TABLE\_NAME,

REFERENCED\_COLUMN\_NAME

FROM

INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE

WHERE

TABLE\_SCHEMA = 'PetPalsDB' AND

REFERENCED\_TABLE\_NAME IS NOT NULL;

INSERT INTO Shelters (Name, Location) VALUES

('Chennai Pets', 'Chennai'),

('Bangalore Animals', 'Bangalore'),

('Mumbai Strays', 'Mumbai'),

('Delhi Care', 'Delhi'),

('Hyderabad Friends', 'Hyderabad');

INSERT INTO Users (Name) VALUES

('Ramesh Kumar'),

('Priya Patel'),

('Amit Shah'),

('Deepika Singh'),

('Vijay Reddy');

INSERT INTO Pets (Name, Age, Breed, Type, AvailableForAdoption, ShelterID) VALUES

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

('Misty', 4, 'Siamese', 'Cat', TRUE, 1),

('Rocky', 5, 'Beagle', 'Dog', FALSE, 2),

('Whiskers', 1, 'Persian', 'Cat', TRUE, 3),

('Bruno', 3, 'Boxer', 'Dog', FALSE, 4);

INSERT INTO Adoption (PetID, UserID, AdoptionDate) VALUES

(3, 1, '2025-01-15'),

(5, 2, '2025-02-20');

INSERT INTO Donations (DonorName, DonationType, DonationAmount, DonationItem, ShelterID) VALUES

('Arun Jain', 'Cash', 1000.00, NULL, 1),

('Neha Gupta', 'Item', NULL, 'Pet Food', 1),

('Rajesh Mehta', 'Cash', 500.00, NULL, 2),

('Sonia Kapoor', 'Item', NULL, 'Toys', 3),

('Vikram Singh', 'Cash', 750.00, NULL, 4);

INSERT INTO AdoptionEvents (EventName, EventDate, Location) VALUES

('Pet Fest', '2025-06-10 10:00:00', 'Chennai'),

('Adoptathon', '2025-07-15 11:00:00', 'Bangalore'),

('Furry Friends Day', '2025-08-20 09:00:00', 'Mumbai'),

('Pet Carnival', '2025-09-05 14:00:00', 'Delhi'),

('Animal Love', '2025-10-12 10:00:00', 'Hyderabad');

INSERT INTO Participants (ParticipantName, ParticipantType, EventID) VALUES

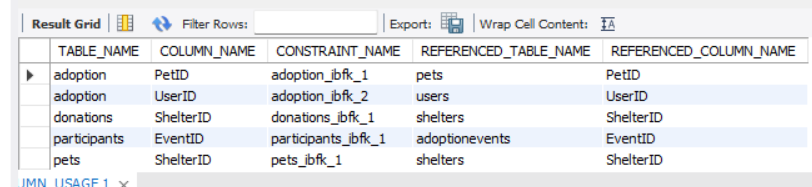
('Chennai Pets', 'Shelter', 1),

('Ramesh Kumar', 'Adopter', 1),

('Bangalore Animals', 'Shelter', 2),

('Priya Patel', 'Adopter', 2),

('Mumbai Strays', 'Shelter', 3);



**Q4.** Ensure the script handles potential errors, such as if the database or tables already exist.

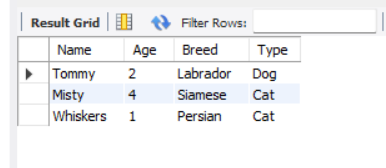
select table\_name, column\_name, referenced\_table\_name

from information\_schema.key\_column\_usage

where table\_schema = 'petpalsdb' and referenced\_table\_name is not null;

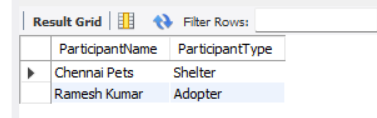
**Q5**. Write an SQL query that retrieves a list of available pets (those marked as available for adoption) from the "Pets" table. Include the pet's name, age, breed, and type in the result set. Ensure that the query filters out pets that are not available for adoption.

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



**Q6.** 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;



**Q7.** 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.

DELIMITER //

CREATE PROCEDURE UpdateShelterInfo(IN p\_ShelterID INT, IN p\_NewName VARCHAR(100), IN p\_NewLocation VARCHAR(255))

BEGIN

IF EXISTS (SELECT 1 FROM Shelters WHERE ShelterID = p\_ShelterID) THEN

UPDATE Shelters SET Name = p\_NewName, Location = p\_NewLocation WHERE ShelterID = p\_ShelterID;

ELSE

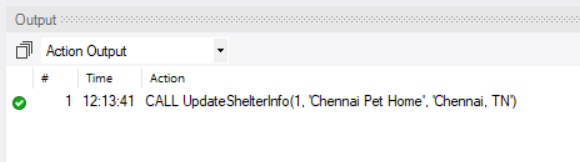
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Shelter not found';

END IF;

END //

DELIMITER ;

CALL UpdateShelterInfo(1, 'Chennai Pet Home', 'Chennai, TN');

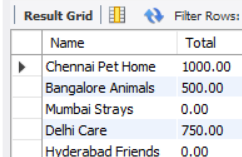


**Q8.** Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.

SELECT s.Name, COALESCE(SUM(d.DonationAmount), 0) AS Total

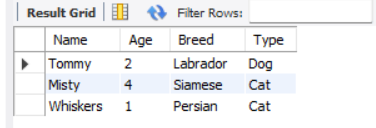
FROM Shelters s LEFT JOIN Donations d ON s.ShelterID = d.ShelterID

GROUP BY s.ShelterID;



**Q9.** 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, Type FROM Pets WHERE PetID NOT IN (SELECT PetID FROM Adoption);



**Q10**. 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

MonthYear,

SUM(DonationAmount) AS Total

FROM (

SELECT

DATE\_FORMAT(DonationDate, '%M %Y') AS MonthYear,

DATE\_FORMAT(DonationDate, '%Y-%m') AS YearMonth,

DonationAmount

FROM Donations

) AS formatted\_donations

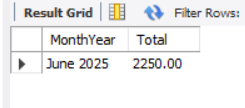
GROUP BY

YearMonth,

MonthYear

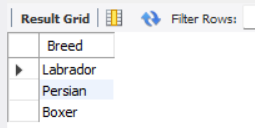
ORDER BY

YearMonth



**Q11**. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

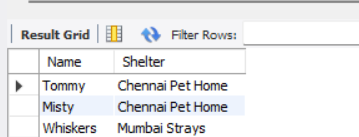
SELECT DISTINCT Breed FROM Pets WHERE Age BETWEEN 1 AND 3 OR Age > 5;



**Q12.** Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

SELECT p.Name, s.Name AS Shelter FROM Pets p JOIN Shelters s ON p.ShelterID = s.ShelterID

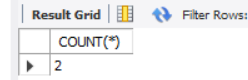
WHERE p.AvailableForAdoption = TRUE;



**Q13**. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

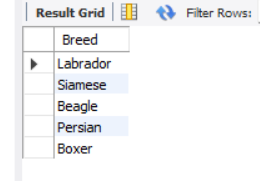
SELECT COUNT(\*) FROM Participants p JOIN AdoptionEvents e ON p.EventID = e.EventID

WHERE e.Location LIKE '%Chennai%';



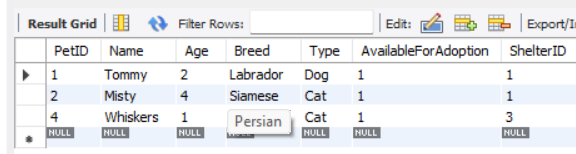
**Q14**. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

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



**Q15.** Find the pets that have not been adopted by selecting their information from the 'Pet' table.

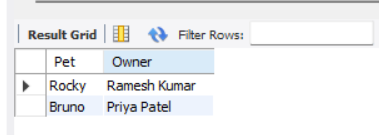
SELECT \* FROM Pets WHERE PetID NOT IN (SELECT PetID FROM Adoption);



**Q16.** Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'User' tables.

SELECT p.Name AS Pet, u.Name AS Owner FROM Adoption a

JOIN Pets p ON a.PetID = p.PetID JOIN Users u ON a.UserID = u.UserID;

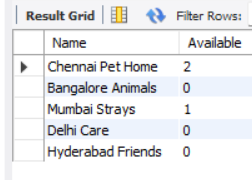


**Q17**. Retrieve a list of all shelters along with the count of pets currently available for adoption in each shelter.

SELECT s.Name, COUNT(p.PetID) AS Available FROM Shelters s

LEFT JOIN Pets p ON s.ShelterID = p.ShelterID AND p.AvailableForAdoption = TRUE

GROUP BY s.ShelterID;

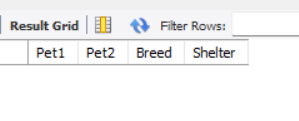


**Q18**. Find pairs of pets from the same shelter that have the same breed.

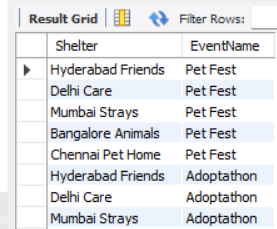
SELECT a.Name AS Pet1, b.Name AS Pet2, a.Breed, s.Name AS Shelter

FROM Pets a JOIN Pets b ON a.Breed = b.Breed AND a.PetID < b.PetID AND a.ShelterID = b.ShelterID

JOIN Shelters s ON a.ShelterID = s.ShelterID;



**Q19**. List all possible combinations of shelters and adoption events.  
SELECT s.Name AS Shelter, e.EventName FROM Shelters s CROSS JOIN AdoptionEvents e;



**Q20**. Determine the shelter that has the highest number of adopted pets.

SELECT s.Name, COUNT(\*) AS Adoptions FROM Adoption a

JOIN Pets p ON a.PetID = p.PetID JOIN Shelters s ON p.ShelterID = s.ShelterID

GROUP BY s.ShelterID ORDER BY Adoptions DESC LIMIT 1;

