**SQL CODING CHALLENGE**

-- 1) creating database

create database petpals;

use petpals;

drop database petpals;

-- 2) nd 3) creating tables with constraints

create table Pets(

PetID int primary key, Name varchar(100), Age int,

Breed varchar(100), Type varchar(50), AvailableForAdoption bit);

create table Shelters(

ShelterID int primary key, Name varchar(100), Location varchar(100));

-- foreign key shelter id is added since it is needed in 8th ques

create table Donations(

DonationID int primary key, DonorName varchar(100), DonationType varchar (100),

DonationAmount decimal(10,2), DonationItem varchar(100), DonationDate datetime,

ShelterID int, foreign key(ShelterID) references Shelters(ShelterID));

drop table Donations;

create table AdoptionEvents(

EventID int primary key, EventName varchar(255), EventDate datetime,

Location varchar(100));

create table Participants(

ParticipantID int primary key, ParticipantName varchar(100),

ParticipantType varchar(50), EventID int, foreign key(EventID) references AdoptionEvents(EventID));

-- 4) no errors

-- 5) list of available pets for adoption

select Name, Age, Breed, Type from Pets where AvailableForAdoption = 1;

-- 6) participants names nd types for specific event

select p.ParticipantName, p.ParticipantType, e.EventName from Participants p join AdoptionEvents e

on p.EventID = e.EventID where p.EventID = 1;

select ParticipantName, ParticipantType from Participants where EventID = 1;

-- 7) stored procedure to update shelter info

/\* create procedure updation(

in sID int,

in newName varchar(50),

in newLocation varchar(100))

begin

update Shelters

set Name = newName, Location = newLocation

where ShelterID = sID;

end

maybe we can use if else to handle errors \*/

-- 8) total donation amount by shelter

select s.Name as ShelterName, ifnull(sum(d.DonationAmount), 0) as TotalDonation

from Shelters s left join Donations d on d.ShelterID = s.ShelterID group by s.ShelterID, s.Name;

select s.Name as ShelterName, sum(d.DonationAmount) as TotalDonation

from Shelters s left join Donations d on d.DonorName = s.Name group by s.Name;

-- 9) pets without owners

alter table Pets add OwnerID int;

select Name, Age, Breed, Type from Pets where OwnerID is null;

-- 10) monthly donation totals

select date\_format(DonationDate, '%M%Y') as MonthYear,

sum(DonationAmount) as TotalDonation from Donations group by MonthYear order by MonthYear;

-- 11) breeds for pets age 1-3 or >5

select distinct breed from Pets where Age between 1 and 3 or Age > 5;

-- 12) available pets with their shelters

select p.Name as Petname, s.Name as Sheltername from Pets p join Shelters s

on p.PetID = s.ShelterID where p.AvailableForAdoption = 1;

-- 13) Participants in events by city

select count(p.ParticipantID) as TotalParticipants

from Participants p join AdoptionEvents e on p.EventID = e.EventID

WHERE e.Location = 'Chennai';

-- 14) unique breeds of pets - age(1-5)

select distinct Breed from Pets

where Age between 1 and 5;

-- 15) pets not adopted

select\*from Pets where AvailableForAdoption = 1;

-- 16) adopted pets and adopters

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;

-- 17) count of available pets

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;

-- 18) same breed pairs in same shelter

select p1.Name as Pet1, p2.Name as Pet2, p1.Breed

from Pets p1 join Pets p2 on p1.ShelterID = p2.ShelterID

and p1.Breed = p2.Breed and p1.PetID < p2.PetID;

-- 19) all shelter and event combinations

select s.Name as ShelterName, e.EventName

from Shelters s cross join AdoptionEvents e;

-- 20) shelter with highest adopted pets

select s.Name as ShelterName, COUNT(a.PetID) AS AdoptedCount

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 AdoptedCount desc

limit 1;