**Coding Challenges - PetPals, The Pet Adoption Platform**

**J520-Srimathi** **R**

**Problem Statement:**

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

**TASK:**

1. **Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals”.**
2. **Create tables for pets, shelters, donations, adoption events, and participants.**
3. **Define appropriate primary keys, foreign keys, and constraints.**
4. **Ensure the script handles potential errors, such as if the database or tables already exist.**

**Creating database:**

create database petpals;

use petpals;

**Creating tables:**

**Shelters table**

create table shelters (

shelterid int primary key auto\_increment,

name varchar(150) not null,

location varchar(255) not null

);

**Pets table**

create table pets (

petid int primary key auto\_increment,

name varchar(100) not null,

age int not null,

breed varchar(100) not null,

type varchar(50) not null,

availableforadoption bit not null,

shelterid int,

ownerid int default null,

foreign key (shelterid) references shelters(shelterid) on delete set null

);

**Donations table**

create table donations (

donationid int primary key auto\_increment,

donorname varchar(150) not null,

donationtype enum('cash', 'item') not null,

donationamount decimal(10,2) default null,

donationitem varchar(255) default null,

donationdate datetime not null,

shelterid int,

foreign key (shelterid) references shelters(shelterid) on delete set null

);

**Adoption Events table**

create table adoptionevents (

eventid int primary key auto\_increment,

eventname varchar(200) not null,

eventdate datetime not null,

location varchar(255) not null

);

**Participants table**

create table participants (

participantid int primary key auto\_increment,

participantname varchar(150) not null,

participanttype enum('shelter', 'adopter') not null,

eventid int,

foreign key (eventid) references adoptionevents(eventid) on delete set null

);

**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 pets where availableforadoption = 1;

**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.**

select participantname, participanttype from participants where eventid = 1;

**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.**

delimiter //

create procedure update\_shelter\_info(

in shelter\_id int,

in new\_name varchar(150),

in new\_location varchar(255)

)

begin

if (select count(\*) from shelters where shelterid = shelter\_id) = 0 then

signal sqlstate '45000' set message\_text = 'Invalid shelter ID';

end if;

update shelters set name = new\_name, location = new\_location where shelterid = shelter\_id;

end //

delimiter ;

**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, coalesce(sum(d.donationamount), 0) as total\_donations from shelters s left join donations d on s.shelterid = d.shelterid group by s.shelterid;

**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, type from pets where ownerid is null;

**10. 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 date\_format(donationdate, '%M %Y') as month\_year, sum(donationamount) as total\_donations from donations group by month\_year;

**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 between 1 and 3) or (age > 5);

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

select p.name as pet\_name, s.name as shelter\_name from pets p join shelters s on p.shelterid = s.shelterid where p.availableforadoption = 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 adoptionevents e on p.eventid = e.eventid where e.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 selecting their information from the 'Pet' table.**

select name, age, breed, type from pets where ownerid is null;

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

select p.name as pet\_name, pr.participantname as adopter\_name from pets p join participants pr on p.ownerid = pr.participantid;

**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.petid) as available\_pets from shelters s left join pets p on s.shelterid = p.shelterid and p.availableforadoption = 1 group by s.shelterid;

**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.shelterid = p2.shelterid and p1.breed = p2.breed and p1.petid < p2.petid

join shelters s on p1.shelterid = s.shelterid;

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

select s.name as shelter\_name, e.eventname as event\_name from shelters s cross join adoptionevents e;

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

select s.name as shelter\_name, count(p.petid) as adopted\_pets from shelters s join pets p on s.shelterid = p.shelterid where p.ownerid is not null group by s.shelterid order by adopted\_pets desc limit 1;