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

create database pp;

use pp;

**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.**

**Answer:**

create table pets(

petid int auto\_increment primary key,

name varchar(50) not null,

age int not null,

breed varchar(50) not null,

type varchar(50),

shelterid int,

foreign key(shelterif) references shelters(shelterid) on delete cascade,

availableforadoption bit

);

create table shelters(

shelterid int auto\_increment primary key,

name varchar(50) not null,

location varchar(100) not null

);

create table donations(

donationid int auto\_increment primary key,

donorname varchar(50) not null,

donationtype varchar(50) not null,

donationamount float,

donationitem varchar(100),

donationdate datetime not null,

shelterid int,

foreign key(shelterid) references shelters(shelterid)

);

create table adoptionevents(

eventid int auto\_increment primary key,

eventname varchar(50) not null,

eventdate datetime not null,

location varchar(100) not null

);

create table participants(

participantid int auto\_increment primary key,

participantname varchar(50) not null,

participanttype varchar(50) not null,

eventid int ,

foreign key(eventid) references adoptionevents(eventid) on delete cascade

);

create table adoption(adoptionid int auto\_increment primary key,

petid int,

participantid int,

foreign key(petid) references pets(petid) on delete cascade,

foreign key(participantid) references participants(participantid) on delete cascade

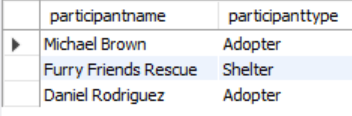
);

**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=3; 

**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 modifyshelter(in id int,in newname varchar(50),in newlocation varchar(50))

begin

if id in (select shelterid from shelters) then

update shelters set name=newname , location=newlocation where shelterid=id;

else

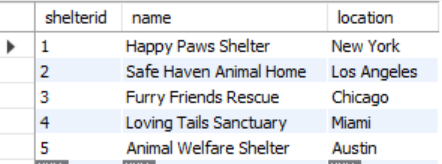
select "Data not found";

end if;

end //

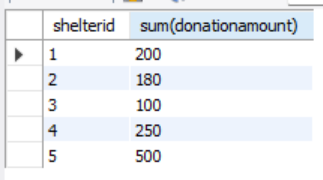
delimiter ;

call modifyshelter(5,"Animal Welfare Shelter","Austin");



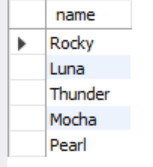
**8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.**

select shelterid,sum(donationamount) from donations group by shelterid;



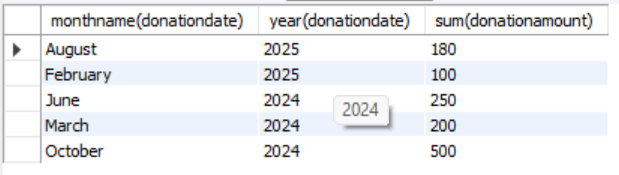
**9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.**

select p.name from pets p left join adoption a on a.petid=p.petid where a.adoptionid 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 monthname(donationdate),year(donationdate),sum(donationamount) from donations group by year(donationdate),monthname(donationdate) having sum(donationamount) is not null order by year(donationdate) desc, monthname(donationdate)asc;



**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;



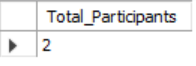
**12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.**

select name,shelterid from pets where 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 inner join adoptionevents a on p.eventid=a.eventid where location="Miami Beach Park";



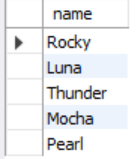
**14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.**

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 p.name from adoption a right join pets p on a.petid=p.petid where a.petid is null;



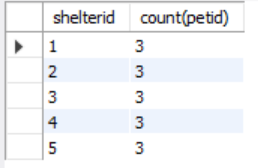
**16. Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'User' tables.**

select p.name as Pet\_Name,pa.participantname as Owner\_Name from pets p inner join adoption a on p.petid=a.petid inner join participants pa on pa.participantid=a.participantid ;



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

select shelterid,count(petid) from pets group by shelterid;



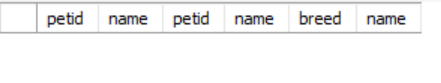
**18. Find pairs of pets from the same shelter that have the same breed.**

select p1.petid , p1.name, p2.petid, p2.name, p1.breed, s.name

from pets p1

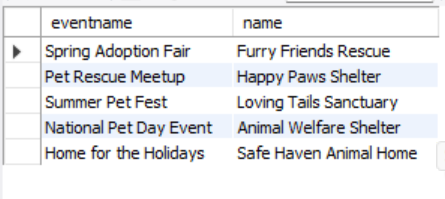
join pets p2 on p1.breed = p2.breed and p1.petid < p2.petid

join shelters s on p1.shelterid = s.shelterid and p2.shelterid = s.shelterid;



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

select a.eventname,s.name from adoptionevents a inner join shelters s on s.shelterid=a.shelterid;



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

select name from shelters where shelterid in( select shelterid from pets group by shelterid having count(\*)=(select count(\*) from pets group by shelterid order by count(\*)desc limit 1));

