

S2_T3

Animal Shelter Management

Group Members:

Tirth Patel – 201801126

Parth Katrodiya – 201801150

Mihir Dangar – 201801220

Mahiraj Rathod – 201801456

Mentor : Aashish Paliwal

Guided by : Prof. Minal Bhinse

Section1: Final version of SRS

Database Requirements Specification

for

Animal Shelter Management System

Version 2.0 approved

Prepared by Group S2_3

**Dhirubhai Ambani Institute of Information and Communication
Technology**

1 October 2020

Table of Contents

Purpose.....	3
Intended audience and reading suggestions.....	3
Project Scope.....	3
Description.....	4
Background Readings.....	4
Interview.....	6
Questionnaire.....	8
Observation.....	11
Fact Finding Chart.....	12
List of requirements.....	13
User Classes and Characteristics.....	15
Operating System.....	16
Product Functions.....	16
Priveleges.....	18
Assumptions.....	19
Business constraints.....	20

Purpose

The animal shelter management has become a necessary due to rapid decline in animals shelter . The shelter assures that necessary information is available for proper care of animals. To maximize the efficiency , information must be provided accurately. Its also helpful for creating and maintaining budgets. This ensures that animal have everything they need. Its very helpful in coordinating pet adoptions and to encouraging people to be responsible about animals . Thousands of animals may be roaming in city and it would have impact on public safety and health . To improve animal health and educate public is the main motto.

Intended audience and Reading suggestions

This database is focused on animal shelter home where the lost animals or the animals not having homes are taken care of. This can be useful for people who are searching an animal to adopt, the administration can check the number of animals present ,For those people who are willing to donate their pet, It can also be useful for people who are willing to donate money for the welfare of animals, different shelters can also check each other if there is shortage of space in the shelter, the administration can also check the time for routine check-up of different animals, this database can also give us the contact of doctors of different animals, Government can also check if there is space for the animals found on roads.

Project Scope

This database will be used to create a software that will be used to simplify the process of keeping a track of the animals adopted by the shelter that rescues the animals in need. The software will help in making a number of processes smooth and easy with accommodation of all data together. Animal care and staff related problems such as the type of animal, breed, age, weight, ID, relative health, special care, instructions specific to helpers/workers etc. Apart from that another set of data essential to this software is data specific to the adoption-arrival which includes data such as date of arrival and adoption, reason of shelter, information specific to donor, adapter etc.

We need to meaningfully manage and utilize all of these to the benefit of the Animal Shelter for the welfare of animals, the shelter and ultimately others too. Thus, the ultimate goal of this software will be the scaling of this adoption model with minimal efforts as well as animal welfare with effective interpretation of all the collected data.

Description

We made this database for solving the problem of managing the animal shelter. This database will allow us to check which kind of animals are present, how many potential buyers are present, the characteristics like age, gender, sex, etc of each animal, the need for the routine medical check-up of the animals, etc. This database will also tell us the business related aspect of the animal shelter farm.

This database will operate on PostgreSQL 10 software and it will require an active internet connection. This database will be completely accessible to the users and the administration. They can work with the database using MySQL language.

In this database we have assumed that the animals which are present in the database will be handed over to the perfect adopter. Only the administration will be allowed to make changes in the database and the users can only read it. There will be different department for the animal adopters and the animal donators.

Background Readings

The background readings referred are given below:

1)‘Shelterbuddy.com’ is an Australia’s leading Non-Profit Animal Welfare Organization which is very similar to our desired final software. So, it was a really helpful resource for us to gain insights of all the essential features to be included, tools to be used, services incorporated into the software etc.

To include and collect information about the animal which the shelter might need to take proper care such as type of animal, breed, ID, age, special health remarks, information about donor (in case), date of arrival, medications and its cost, size/occupancy etc. As well as data from the potential adapter such as Name, email, contact no., address, date of adoption and finally create a community or similar thing to stay connected with them.

REFERENCE = <https://www.shelterbuddy.com/>

2) Most important thing here is to address the lack of standards for shelters. Shelters must be good enough for animal keeping. Staff and public safety must also be taken care . Regular updates and review to stay updated about policies and protocols from government side. Every shelter has a particular capacity thus management is key part.

(Dr. Annette Litster, Director of the Maddie's Shelter Medicine Program at the Purdue University School of Veterinary Medicine presents a review of the Association of Shelter Veterinarian's newly published Shelter Standards Document.)

REFERENCE = <https://youtu.be/xL-NoDAisDY>

3) Basics things you must take care while deciding to build shelter in area are

- 1) The existing shelters and capacity and how they serve.
- 2) The need of shelters compare to number of animals in that area.
- 3) Financial situation in the area whether public donations will be available .
- 4) The local attitude of dogs and cats.
- 5) Shelter should take care of insurance, food & water facility, equipments for the animal management.

REFERENCE = <https://caninerabiesblueprint.org/Guidelines-for-the-design-and,178>

4) Sudha Shelter home is an Indian non profit Organisation which works completely online. In this organisation they rescue cats from different places and allow people to adopt them all over India. They complete the check up of the rescued cats and then people can adopt them.

The website of this NGO contained the frequently asked questions related to the adoption process, how the cats are transferred and many other things. So we can use this information in our database to improve it.

REFERENCE = <http://sudhashelterhome.in/>

5) Sanjay Gandhi Animal Care Centre(SGACC) is Delhi's Largest all animal care centre and it is one of the oldest animal care founded in 1980. This centre provides shelter to all animals. They are also having animal ambulance service to reach the needy animals in much less time. It is a home to about more than 3000 animals which can be adopted by a right person.

From this website we got the information regarding the services provided by different animal shelter homes. The different services provided here are :

- 1) In-Patient department(open for 24 hours)
- 2) Out-Patient department
- 3) Intensive care unit
- 4) Operation theatre for animals
- 5) Free treatment given to animals
- 6) Laboratory
- 7) Dental check up
- 8) Animal birth control programmes for dogs
- 9) Animal cemetery

REFERENCE = <http://www.sanjaygandhianimalcarecentre.org/index.html>

Interview

Interview plan :

Animal Shelter Management System : (Role Play) Interview

Plan System: Animal Shelter Management System Database

Interviewee:

1) Akash Katrodiya (Role Play)

Contact Details: 9909513787

Organization Details: gondal

Interviewer:

1) Mihir Dangar Designation: Business Development Executive

2) Parth Katrodiya Designation: Developer

Date: 28/9/2020

Time: 14:30

Duration: 45 minutes

Place: Akash's Office

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding software usability and maintenance

Agenda:

Problems with software usability and maintenance

Current accounting procedures

Expectations from the developed software

Documents to be brought to the interview:

Current book of accounts

Interview summary :

Summary System: Shelter Management(Role Play)

Interviewee:

1) Akash Katrodiya(Role Play)

Designation: CEO at Season's Flavour

Interviewer:

1) Mihir Dangar Designation: Business Development Executive

2) Parth Katrodiya Designation: Developer

Date: 29/9/2020

Time: 14:30

Duration: 45 minutes

Place: Akash's Office

Purpose of Interview:

Preliminary meeting to identify problems and requirements regarding software usability and maintenance

Not familiar with using new technology to keep records of animals.

Concerned if he could operate the software by himself effectively or not. If not how can he find a reliable person who would fit into the place.

How will the software generate meaningful and useful insights?

Need to optimize the number of staff organising the schedules of all animals (like feeding, cleaning etc)

Current system to keep track of the animals is not able to make the ends meet for the work amounting at the shelter

Being a non-profit organisation, they are financially heavily dependent on the monetary funds of people. The software needs to keep a track of it. Cash or supplies.

Would like to add a few new categories of animals.

Maintenance regarding all the new technologies implemented.

Expects optimized scheduling and reminder features.

Wants to better connect with the past as well as potential customers (adopters and donors).

Visual (graphical) interpretation of the analysis

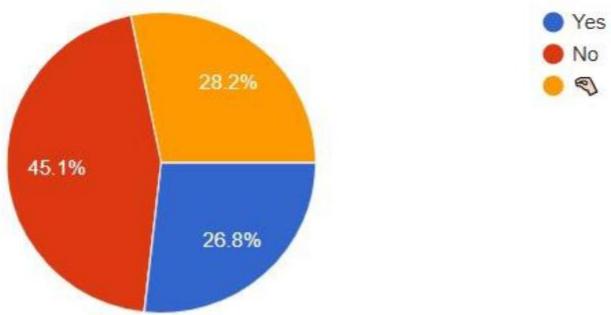
Personalised profiles for customers to keep separate data of adoption/donation and feedback etc

Questionnaire

We conducted a survey regarding Animal shelter home and the thoughts of people on shelter home using Google forms and the results obtained are shown below :

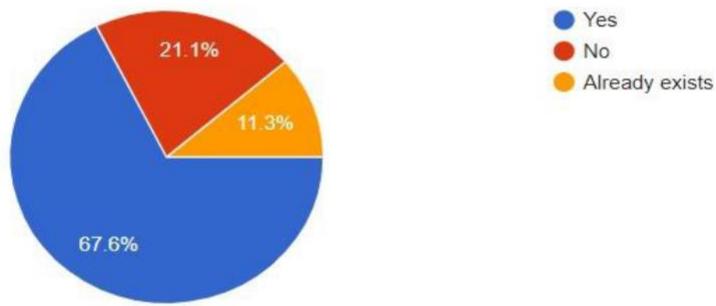
Is there a nuisance of stray animals in your area?

71 responses



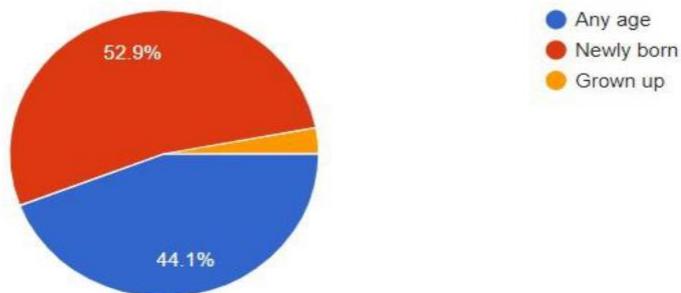
Do you wish that an animal shelter operates in your area?

71 responses



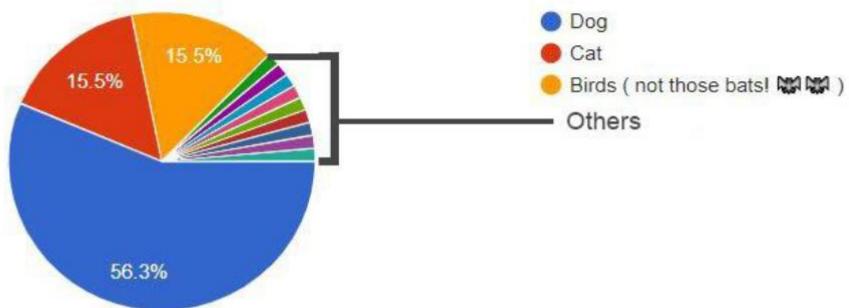
Which age group of pet do you prefer to adopt?

68 responses



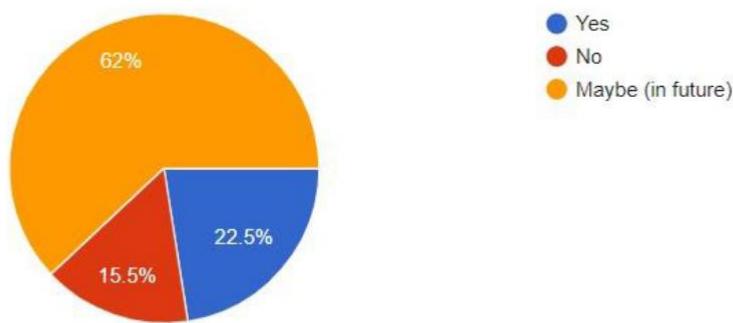
Which animal do you prefer as a pet?

71 responses



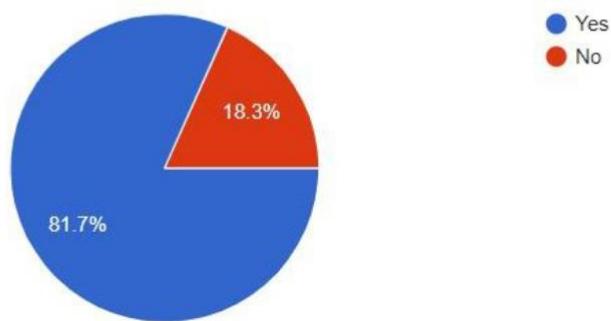
Would you like to sponsor an animal from the shelter?

71 responses



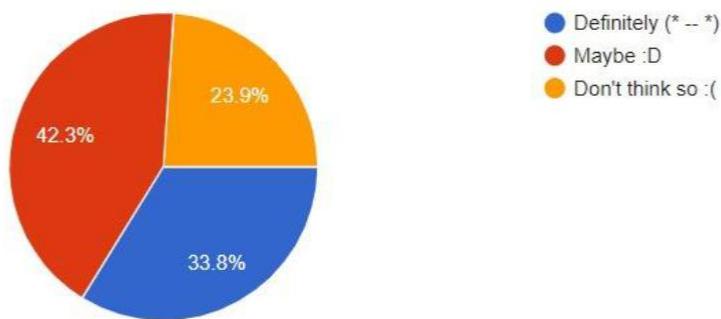
Would you prefer a doctor recommended by the shelter or otherwise?

71 responses



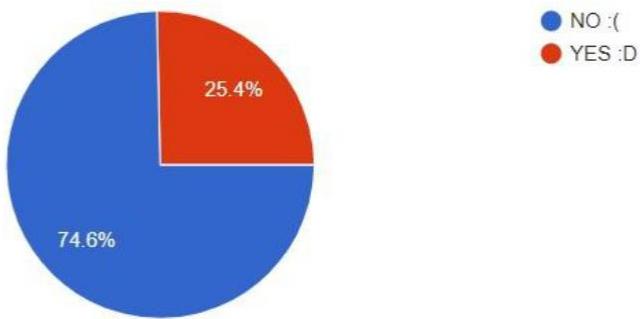
Would you volunteer to work for an animal shelter?

71 responses



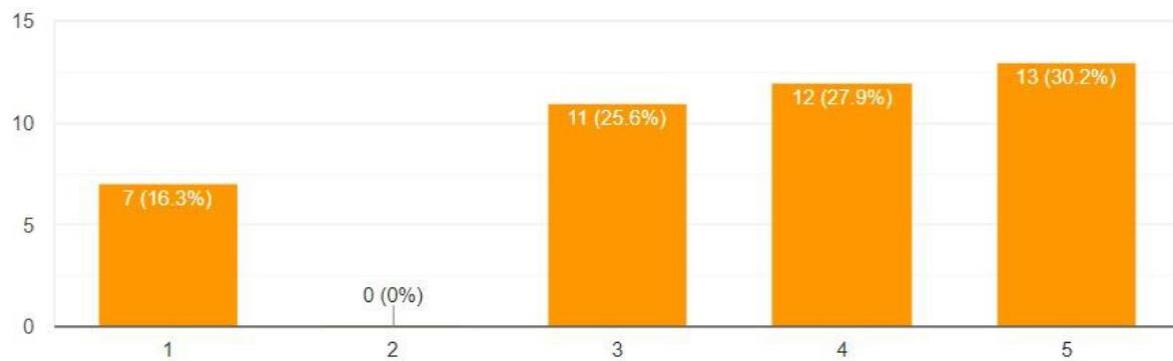
Have you ever had a pet before? □

71 responses



How was/is your experience with your pet?

43 responses



Observation

We had made a vast survey and found many things about animal shelter management. With the increasing population of domestic animals such as dogs and cats and on the same side care taking facilities are not increasing at that rate. Such inadequate facilities can cause starvation, animal abuse and such various problems . Thus there is a great need of animal shelter in places where domestic animals are found most. There is also need of such staff which are well trained and qualified to handle and care for animals. Many things which must be taken care in shelter is nutrition, disease and disease control , welfare of animals . Animal abuse and cruelty can be reduced significantly. There is a great need to bring awareness among people. Many awareness programs can be done by shelter management.

Monitoring progress of each animal regularly is the main task. Regular observations should be taken of all animals such has their eating habits , in case of health issues we can consider vomiting, urine period , digestion of food. Diet of all animals should be regularly given and diet habits should be taken care of. Each animal should be observed properly whether it like live alone or to be in groups. It helps a lot in its social development. The housing and construction of place where the animals live should be well ventilated. Holding capacity must be enough to hold animals in any emergency need Doctors should visit the shelter every week for checkup and health related issues. Proper Trainers should be called to train animals.

The shelter faces many difficulties sometimes due to lack of donations received.The shelter must follow an optimized schedule for proper training and nursing of the animals so that the amount of money required is relatively low and the shelter can work properly from the amount of donations received.

Animal shelters sometime struggle to function correctly in terms of management, taking care of animals. In such cases feedback is a key point. Management team should regularly get feedback from adopters, clients, doctors, trainers, etc. A survey should be conducted.

Fact Finding Chart

Objective	Technique Used	Subject	Time commitment
To gather information regarding the adoption process in shelter homes	Background reading	Websites of different shelter homes	1 hr for each work
To gather information regarding how different animals are taken care of	Background reading	YouTube video and Website	1 hr for each work
What type of health care services are provided from shelter for animals	Background reading	Websites of shelter home	1 hr
How the management and the donations of the shelter home works	Interview	Role play of manager	2 hr
Normal person's thoughts on animal shelters	Questionnaire through Google form	A normal person	2 hr
Information regarding different types of doctors needed	Background reading	Website of shelter home	1 hr
The special characteristics that a type of animal example a dog can have	Background reading	Our basic knowledge of animals	1 hr
To gather information regarding stray animals	Observation	A team member	0.5 hr
Holding capacity of cages in shelter must be optimized according to the animals	Observation	Our team member	1 hr

Combined list of requirements

1) For any animal we can include very specific information such as

- Id
- Age
- Sex
- Name

2) Then for a particular animal we have to check accordingly

- Type of animal
- Breed
- Weight
- Height

3) It also contains adoptions information such as

- Date
- Time
- Day
- Animals information
- Adopter

4) Medical record of the animal

- Date on which medication was done
- Cost
- Vaccine details
- Doctor name
- Trainer name
- Food habits

5) For the residency of animal the details of the cage includes

- Cage number
- Length of cage
- Width and height of cage

6) Details of adopter include

- First name
- Last name
- Permanent address
- City

- State
- Mobile number
- Email address

7) Details of all the staff and volunteers working in shelter

- First name
- Last name
- Permanent address
- City
- State
- Mobile number
- Email address
- Position
- Start day of job
- Salary

8) Details of veterinarian visiting the shelter

- License number
- First name
- Last name
- Email address
- Speciality

9) Details of all the donations and expenses made by the shelter
For donations

- Details of the person who donated for the shelter

Expenses

- Details of all the expenses made by the shelter on a particular thing

User Classes and Characteristics

The different kinds of users who will use the database are :

- 1) The shelter management :They are the main people who would be using the database. All the administrative power like making changes, insertion, deletion and other administrative functions. They will also be able to see the number of animals present. The number of more animals that can be accommodated in the shelter. They can also check the time needed to call the doctor for routine check up. They can also calculate the amount of money from the number and type of animals required for proper functioning of the shelter home. The total amount of the donations can also be checked.
- 2) Animal adopter : They can go through the database and select the best for them.
- 3) Animal sponsor : They can sponsor an animal they like.
- 4) Animal donator : They can check if there is a proper space present for their animal.
- 5) Government : To check if the work is functioning properly or not.
- 6) Doctors : To know the time for the routine check-up
- 7) Donators : They can check the functioning of the shelter and can donate the amount they can.
- 8) Volunteers : To help the shelter in taking care of the animals.
- 9) Pet owners : They can check for doctors for the check up of their pet
- 10) Animal Trainers : They can check which animal need training.

Operating system

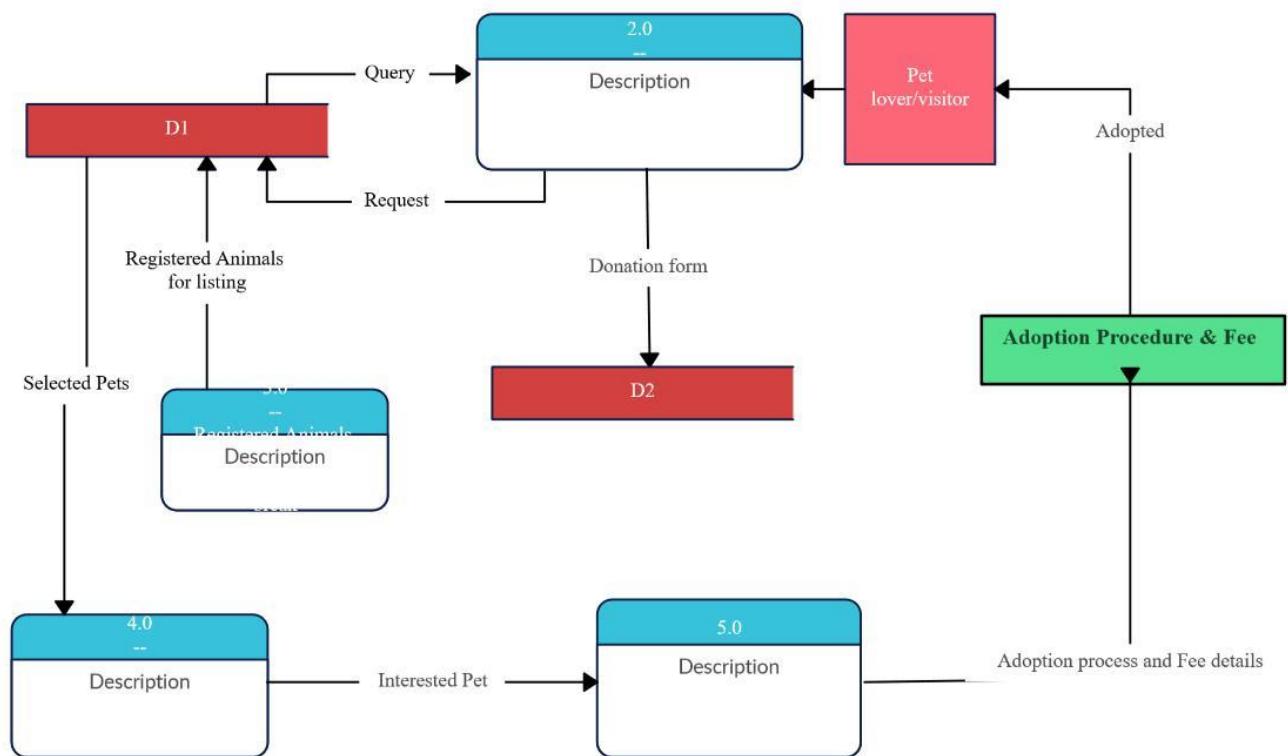
We will use Postgres SQL 10.

It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.

It needs a web browser and PgAdmin must be installed on the device.

Product Functions

Here a data diagram of animal shelter management has been depicted .It gives an overview of the animal management system briefly.



Functionalities of the above diagram:

1. The system shall display all details of animals.
2. The adopter must be able to make its profile on software for adoption of animals.
3. The system should display detailed information of animals present in shelter and also which are adopted earlier .
4. The system should differentiate among different animals based on various properties and habits such as eating habits, conditions suitable for that particular animal and many more.
5. The system should allow user to search and display relevant information related to search.
6. The system shall have all the navigation buttons .
7. The system must have a section to fill up for details if you wish to pick up stray animals in your area.
8. The system should provide online help, FAQs customer support , toll free number service.
9. All the details of the animal shelter management must be displayed on the home page .
10. The system should have option for staff recruitment.
11. The system should display the details of the doctors and trainers visiting the shelter.
12. It should have a complete profile of animal staying in shelter .
13. A email/message based conversation should be developed in software between the staff and the users.
14. Regular updates should be send to adopters for medical health checkup and eating habits.
15. For any payment requirement, the payment should be very securely done and it should not display credit card details or any such confidential details on the back-end side.
16. The software must not leave any cookies on the user computer.
17. The system must logout all customers after a certain period of time.
18. The system must be able to handle many users at a time.
19. The user interface for the software shall be compatible to any browser such as Mozilla, Internet explorer , Google Chrome, etc
20. All the back end databases must be encrypted .

Privileges given to different class :

All detailed information of the animals in the shelter :

- 1) The Shelter Management
- 2) Animal adopter
- 3) Government
- 4) Doctors
- 5) Volunteers
- 6) Trainers

Generating useful insights (charts/graphs/informatics) :

- 1) The Shelter Management
- 2) Volunteers

Optimized scheduling and reminder features :

- 1) The Shelter Management
- 2) Animal trainers
- 3) Doctors
- 4) Volunteers

Portal for donations :

- 1) The Shelter Management
- 2) Donors

Updating the database (Entry/exit of animals)

- 1) The Shelter Management
- 2) Volunteers

Adopter Profile

- 1) Adopter
- 2) The Shelter Management

Assumptions

- Availability of affordable skillful labour to run the software and keep it updated, in case the owner or organisation is not able to operate it by itself
- The business is scaling up and open to multiple locations instead of being restricted to its current singular location and enough staff is available for the same purpose.
- Availability of doctors, adopters and trainers for all kinds of animals that the shelter is willing to accommodate.
- Once handed over the owner/organisation will make the most efficient use of the software or at least not misuse it and keep the database of software updated all the time.
- Customers are willing and trust on an online platform to communicate or discover pets for adoption or donation.
- The organization will maintain transparency about its records regarding the statistics and data generated by the software with its customers.

Business Constraints

- In the village area, running an online database is quite challenging because people in village areas usually don't have internet facilities. Most of them are unaware that how to use the software.
- Due to all online process, possibly there can be delay in pet adoption because of checking profile of Adopter very sharply. Which is more efficient and transparent in offline process. Thus there could be time constraint in online process because there can be a case of fake documents in online process.
- As shelter grow and change, their staffing and management needs change, as well. This can constrain the shelter growth and productivity when employees cannot adapt to new demands or when additional employees are needed but the capital to pay them is not yet available.
- Sometimes, Government regulations can be a constraint to shelter growth.
- During the starting days, The shelter would not have enough donations for its proper functioning.
- The employed must be highly trained who would use this database.
- The amount of donations must be sufficient or else a time will come when the shelter needs to shut down and the animals must be shifted to other place which is very difficult to find.

Section2: Final Noun Analysis

Description

The Animal Shelter Management Database System will help all the entities involved in the process of pet donation by the donor, management people involved in the shelter to the customers who adopt the pets from the shelter.

First and foremost it helps the Shelter Management people to carry out the operations of shelter very effectively. Increasing cases of animal nuisance as well as animal abuse being reported in the areas of operation of the shelter as well as beyond its scope shows that the shelter has chances and areas for itself to expand to and improvise as per the data collection and interviews, experiences etc. Moreover, the traditional system of operation and function is not able to meet the requirements of the shelter, as the methods have gone outdated and expired there is a need to adapt with the change in technology to scale up the operations as it is in the betterment of both the animals and the management team as well as the society as a whole.

Running for a really long time now, the staff of experience at the shelter is well aware of the importance of data and the potential of it. The new system implementation only helps improve the classification and maintenance of all the current and past data. The new system can be modified - improved a lot, to be specific and much variable kind of data can be collected and managed. This can be useful in a bunch of ways such as optimising the schedules of the staff to nurture the animals, study the health of animals, generate useful insights and probable trends, making best use of available resources, automation and much more.

For attracting more customers and keeping the past and usual customers engaged with the shelter as well as create a community of pet loving people, the shelter management can also plan an online platform

by designing and implementing a social media feed that is specific to the media and content related to their healthy relation with the pets as well as anything and everything a person would like to associate with a pet. The new system database can get extremely useful information from

such activities as well as it helps in maintaining a very vibrant culture amongst a strong community. This really is a modern and creative way of developing a bond and interacting with the community.

The Shelter management can do something really outside the box with the database it has. Using its strong network with the community of animal lovers and its digital increased influence, it can also plan to conduct animal awareness campaigns and programmes to incorporate empathy towards animals. It will be twice as effective, it can be very effective to create a positive impact on the citizens about the sympathy towards animals which in turn results in less abuse towards animals and the shelter gets a chance to widen its community with such activities.

We made this database for solving the problem of managing the animal shelter. This database will allow us to check which kind of animals are present, how many potential buyers are present, the characteristics like age, gender, sex, etc of each animal, the need for the routine medical check-up of the animals, etc. This database will also tell us the business related aspect of the animal shelter farm.

For SRS, We ask people for their opinions and thoughts on animal shelters. In which we ask them about
their favourite pet/pets,
their thoughts on sponsoring a animal in shelter,
Is there any animal shelter present near their home ,
They had a pet in past or they want one in future ?,
If yes then from which age group? New born, adult or old

This SRS includes all the observations and study which reflects the business requirements gained through multiple interviews .Animal shelter is a very basic need in all areas. The Shelter will assure that necessary information will be available to provide care to their animals. It may be a better option for shelter if it opens one branch instead of multiple branches.

There are three types of animal shelters. First include municipal animal control agencies which are run by city or county governments; second which include non-profit agencies run by a board of directors; and third as private, non-profit agencies with a government contract to provide animal control services.

To assure that high quality care for shelter animals is provided .

Management of relationships with all people who interact with the shelter.

Here we have tried to make a database which is very user friendly. Due to inadequate facilities of animal shelter management in many areas there is a strong need for animal shelter management and its database should be developed such that it contributes greatly to our shelter management. Well trained staff is also needed . Awareness among people is also needed. For monitoring all activities we need a good database and a software which can handle all things at a time. Database keeps the track of the progress of all animals present in the shelter and also which has been adopted earlier.

This database will operate on PostgreSQL 10 software and it will require an active internet connection. This database will be completely accessible to the users and the administration. They can work with the database using MySQL language.

In this database we have assumed that the animals which are present in the database will be handed over to the perfect adopter. Only the administration will be allowed to make changes in the database and the users can only read it. There will be different departments for the animal adopters and the animal donators.

To make a proper database we need to search over some topics such as different habits of animals, how the management team and workers will work, how to maintain funds received and how to spend ,which doctor and trainer is treating and maintaining the animal.

For the database it will require following information which may include animal's sex, age, name, breed, height, weight . It also includes

adoptions details such as who adopts, date and time of adoptions .It also includes medical information of each animal. Details of all staff and volunteers working at a shelter.

In the village area, running business is quite challenging because people in village areas usually don't have internet facilities. Most of them are unaware that how to use the software.

Due to all online process, possibly there can be delay in pet adoption because of checking profile of Adopter very sharply. Which is more better and transparent in offline process. Thus there could be time constraint in online process because there can be a case of fake documents in online process.

As businesses grow and change, their staffing and management needs change, as well. This can constrain business growth and productivity when employees cannot adapt to new demands or when additional employees are needed but the capital to pay them is not yet available.

Sometimes, Government regulations can be a constraint to business growth. Because it regulates import/export.

As our platform is an online platform, there could be a tough competition among all businessmen. Then probably less profit to shelter and if a store does not have the money to buy more inventory, its ability to sell is constrained.

Hierarchy:

Our ERD does not contain any entity in hierarchical structure. The staff, volunteers, doctors, trainers etc are framed as separate entities in itself.

Aggregation:

In the process of construction of our ERD, we removed all the redundancy (i.e in case of entities: *Doctor*, *Medical record* and *Animal Profile*) so we need not to aggregate at the end. Hence, no aggregation.

Recursive:

We have no relation for any entity over itself. Hence, non-recursive.

Association Relations:

Here we have made a database in which it has weak entity sets , one to one and many to one relationships with their respective cardinality.

One to many relationships include :

Donor to staff -- A donor communicates with various staff members

Donor to Animal Profile -- A Donor can donates more than one animals in the shelter

Volunteer to Animal Profile -- A volunteer can nurtures more than one animal

Trainer to Animal Profile -- A trainer trains more than one animals

Doctor to Animal profile -- A doctor treats various animals

Adopter to Animal Profile -- A adopter can adopt various animals from shelter

Adopter to Staff -- A adopter communicates with various staff members in the shelter.

Doctor to Medical Record -- A doctor can treat more than one animal

Cage to Animal Profile -- A cage can accomodates many animals

Many to Many relationship include :

Sponsor to Animal Profile -- Many sponsor can sponsors various animals

Staff to Animal Profile -- Many staff members can nurtures many animals

One to One relationships include :

Animal Profile to medical record -- A animal can have a medical record

Weak Entity set :

Medical Record --- Here medical record is a weak entity set. So it depends on the animal profile and doctor for its primary key. Thus it can be said as weak entity set.

Nouns	Verbs
Donor	Donates
Staff	nurtures
Customers	adopts
Management people	manages
animals	operations
data	scale
shelter	schedule
insights	automation
businessmen	
Residential people	
community	
doctors	medication
trainers	training
Animal profile	Maintaining
Volunteers	nurtures
Characteristics	
Funds	donate
Sponsors	sponsors
Cage	Captivity

Table - 1: Nouns and Verbs

Candidate entity Set	Candidate Attribute Set	Candidate relationship set
Animal Profile	ID Name Age Sex Breed Cage no.	cumulates
Staff	First name Last name Address Age Mobile number Email Position Start date of job Salary	nurtures
Donor	First name Last name Address Mobile number Email Date of donation Donation details	donates
Adopter	First name Last name Address Mobile number Email Date of adoption Amount paid	adopts
Sponsor	First name Last name Address Mobile number Email Starting date of sponsoring Period of sponsorship Animal IDs	sponsors
Volunteer	First name Last name Address Mobile number Email Starting date of working End date of working Weekly working hours	nurtures
Doctor	First name	medicates

	Last name Address Mobile number Email License no. Speciality Weekly working hours	
Trainer	First name Last name Address Mobile number Email Animals IDs Weekly working hours	traines
Cage	Cage no. Length Height Weight Animal ID	captivates
Medical Records	Treatment Date Cost	Records Required Doctor

Table - 2: Candidate Entities

Noun	Reject Reason
Management people	Generic word for staff, doctor, trainers etc
Animals	The word is too general
Data	It means the same as Animal Profile entity
Characteristics	Duplicate of Animal profile
Shelter	Irrelevant
Insights	Irrelevant
Businessmen	Irrelevant
Residents	Duplicate for donors, customers
Community	Vague
Funds	Generic word for Sponsors and Donors

Table - 3: Rejected Nouns

Section 3: Final ER-Diagrams all versions

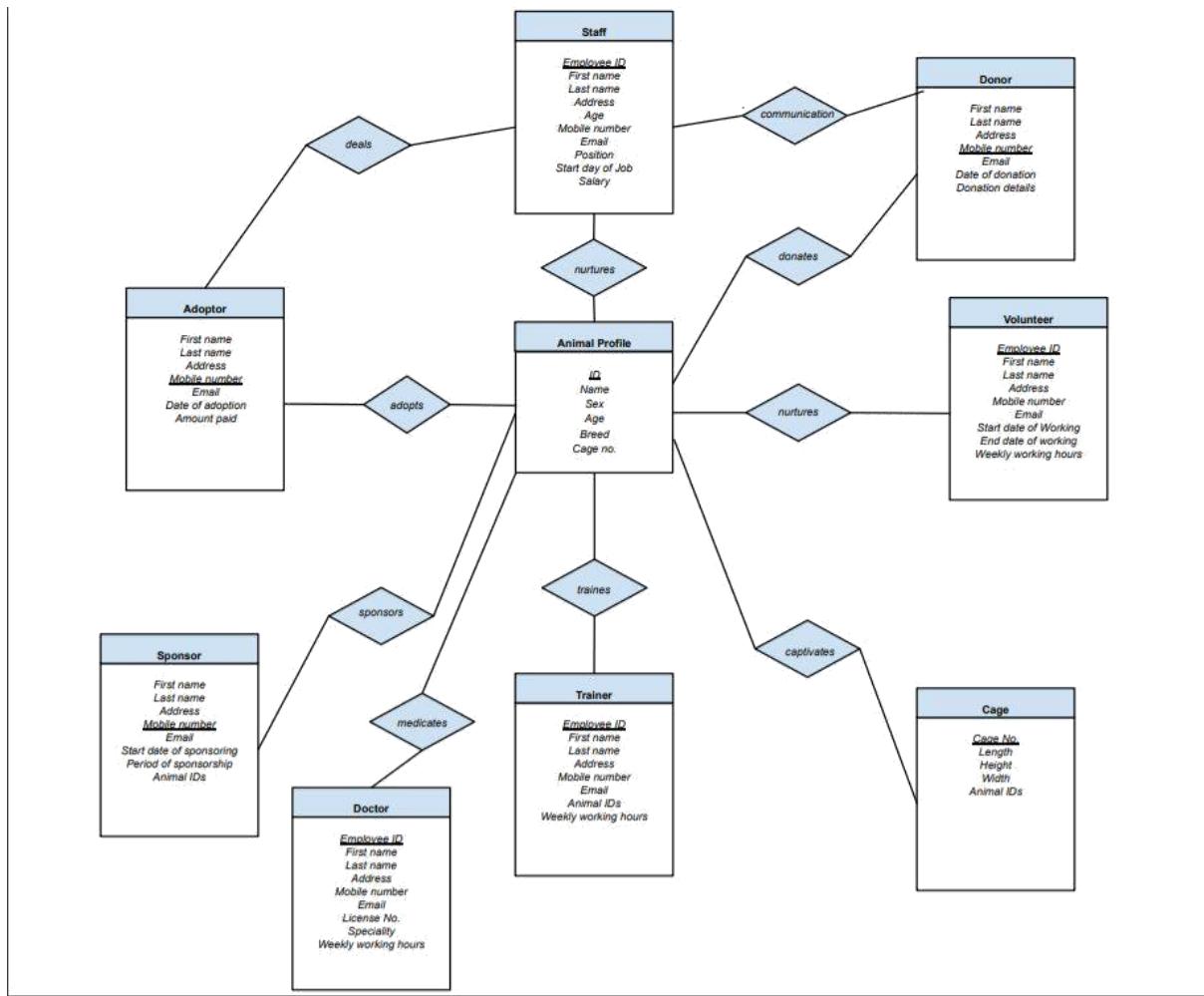


Figure: ER Diagram V1

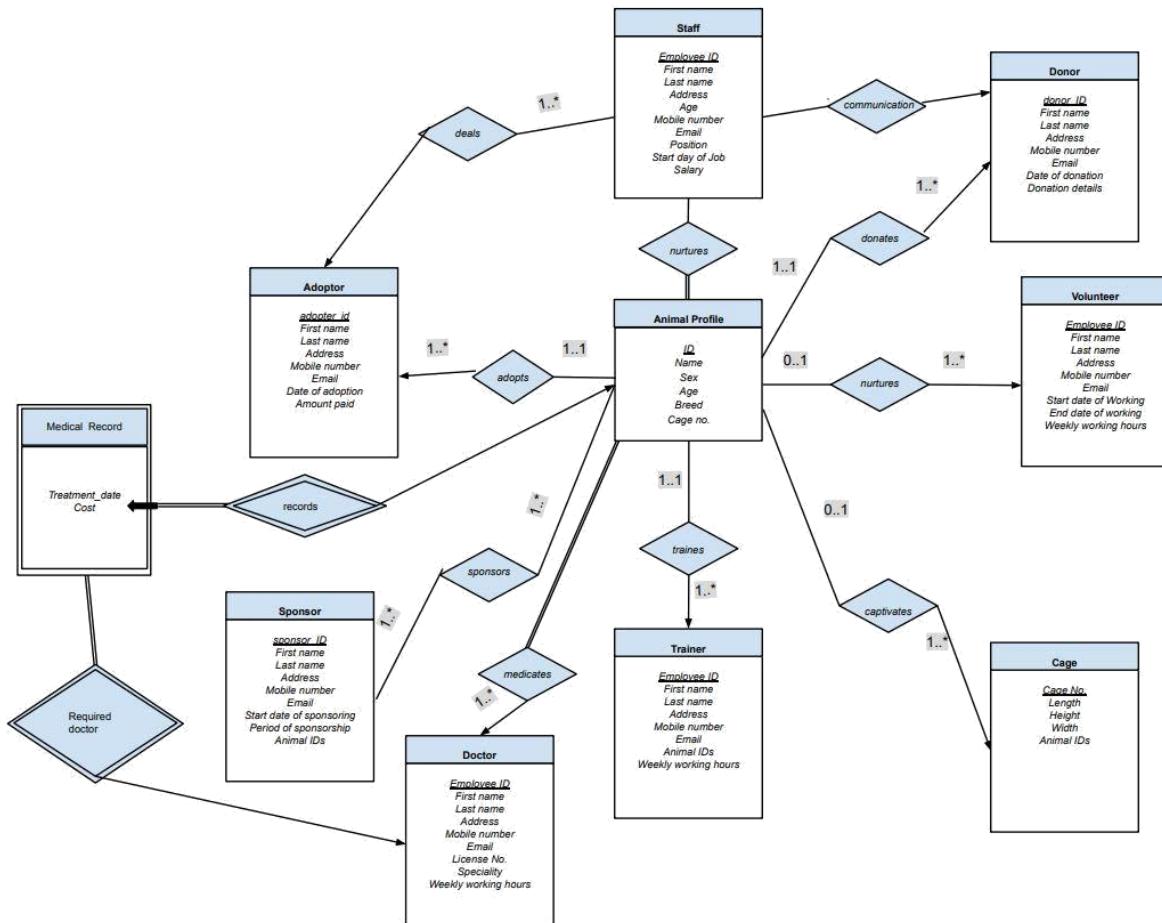
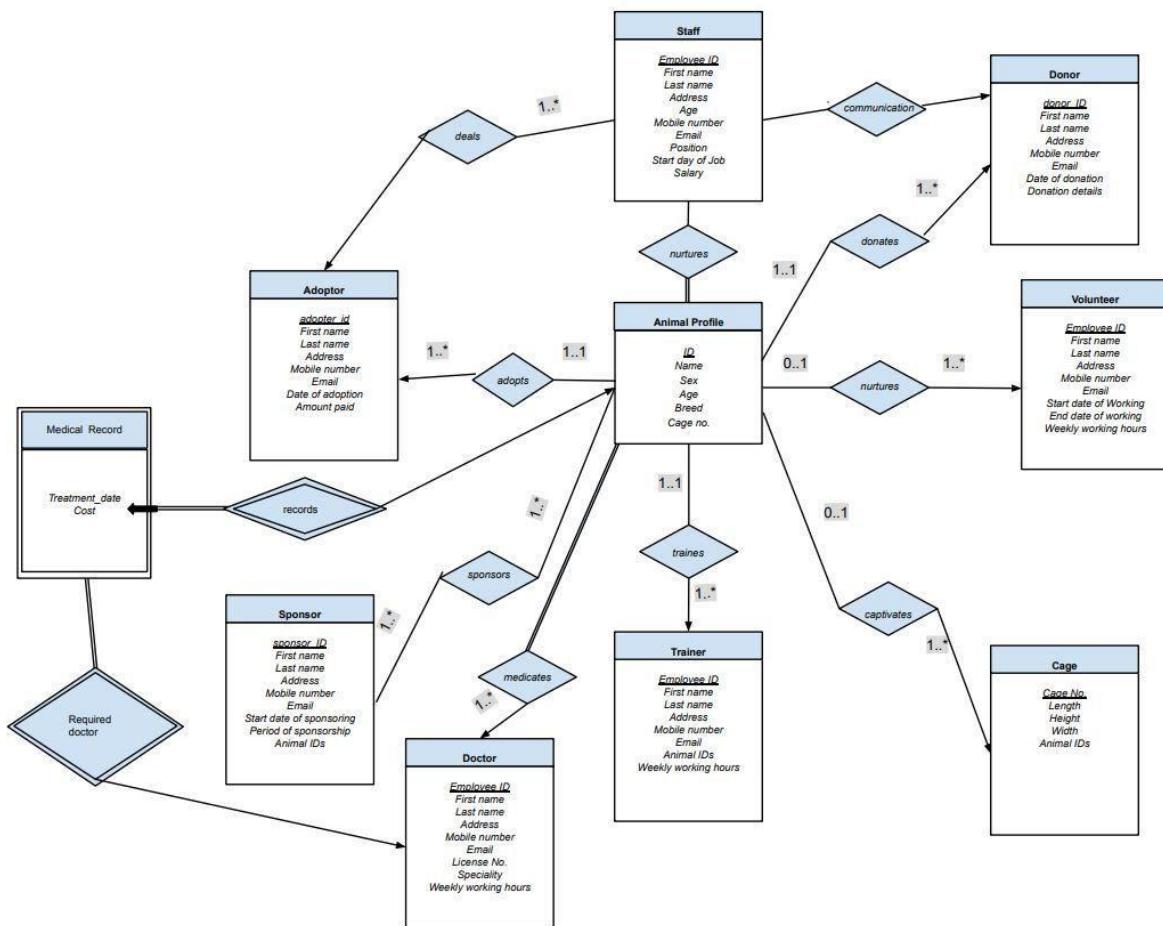


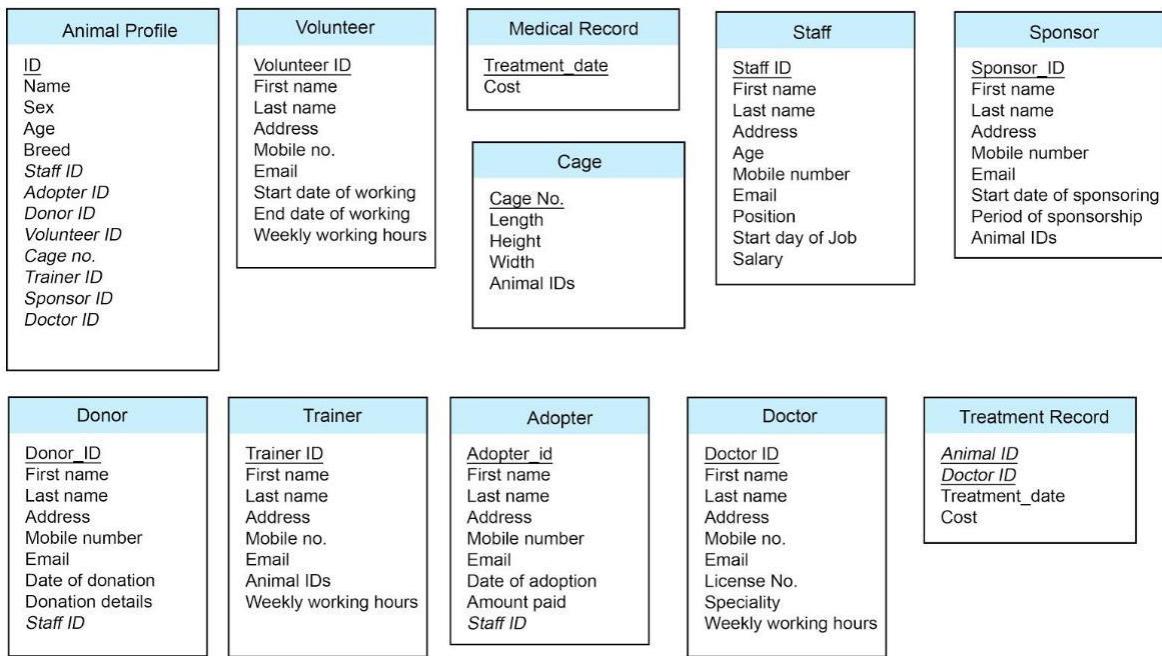
Figure: ER Diagram V2

ER Diagram V3



Section4: Conversion of Final ER-Diagram to Relational Model.

Relationship Model



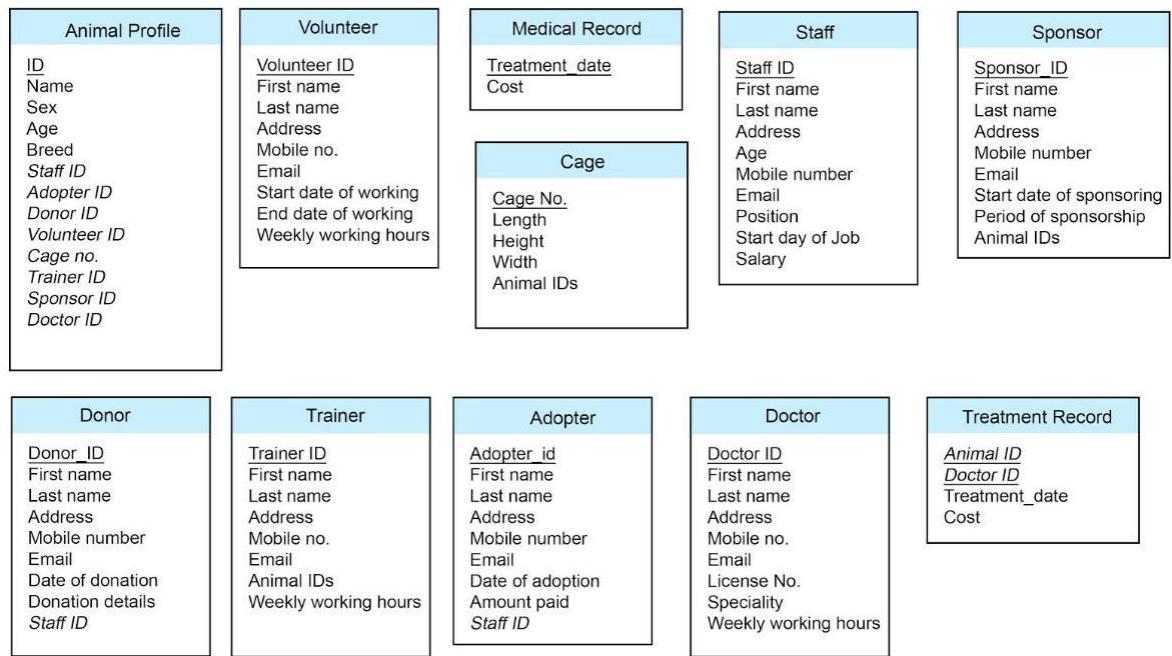
Note: Attributes in *italics* are foreign keys.

Section5: Normalization and Schema Refinement

1. Normalization & Schema Refinement

1.

Relationship Model



Note: Attributes in *italics* are foreign keys.

2. Redundancies

In Sponsor:

Animal_ID

In Trainer:

Animal ID

In cage:

Animal ID

We can eliminate *Animal_ID*s from all the above relations and access it from *Animal Profile* relation as all these relations are otherwise connected with *Animal Profile*

We did not find any other redundancy in our relational model.

3. Final relations of the schema

- Animal Profile (ID, Name, Sex, Age, Breed, Staff ID, Adopter ID, Donor ID, Volunteer ID, Cage no., Trainer ID, Sponsor ID, Doctor ID)
- Volunteer (Volunteer ID, First name, Last name, Address, Mobile no., Email, Start date of working, End date of working, Weekly working hours)
- Medical Record (Treatment_date, Cost)
- Cage (Cage No., Length, Height, Width)
- Staff (Staff ID, First name, Last name, Address, Age, Mobile number, Email, Position, Start day of Job, Salary)
- Sponsors (Sponsor ID, First name, Last name, Address, Mobile number, Email, Start date of sponsoring, Period of sponsorship)
- Donor (Donor ID, First name, Last name, Address, Mobile number, Email, Date of donation, Donation details, Staff ID)
- Trainer (Trainer ID, First name, Last name, Address, Mobile no., Email, Weekly working hours)
- Adopter (Adopter id, First name, Last name, Address, Mobile number, Email, Date of adoption, Amount paid, Staff ID)
- Doctor (Doctor ID, First name, Last name, Address, Mobile no., Email, License No., Speciality, Weekly working hours)
- Treatment Record (Animal ID, Doctor ID, Treatment_date, Cost)

Section6: SQL: Final DDL Scripts, Insert statements, 40 SQL Queries, Snapshots of output of each query

DDL Scripts

```
CREATE TABLE Animal_Profile (
ID int PRIMARY KEY,
name varchar(50),
sex varchar(20),
```

```
age int,  
breed varchar(50)  
FOREIGN KEY (staff_id) REFERENCE staff (employee_ID),  
FOREIGN KEY (adopter_id) REFERENCE adopter (adopter_ID),  
FOREIGN KEY (donor_id) REFERENCE donor (donor_ID), FOREIGN  
KEY (volunteer_id) REFERENCE volunteer (employee_ID), FOREIGN  
KEY (cage_no) REFERENCE cage (cage_no), FOREIGN KEY  
(trainer_id) REFERENCE trainer (employee_ID), FOREIGN KEY  
(sponsor_id) REFERENCE sponsor (sponsor_ID), FOREIGN KEY  
(doctor_id) REFERENCE doctor(employee_ID),  
)
```

```
CREATE TABLE donor (  
donor_ID int PRIMARY KEY,  
first_name varchar(20),  
last_name varchar(20),  
address varchar(200),  
mobile_number bigint NOT NULL,  
email varchar(50),  
date_of_donation date,  
donation_detail varchar(100),  
FOREIGN KEY (staff_ID) REFERENCE staff (employee_ID),  
)
```

```
CREATE TABLE volunteer (  
volunteer_ID int PRIMARY KEY,  
first_name varchar(20),  
last_name varchar(20),  
address varchar(200),  
mobile_number bigint NOT NULL,  
email varchar(50),  
start_date date NOT NULL,  
end_date date NOT NULL,  
working_hours int NOT NULL,  
)
```

```
CREATE Table Trainer(  
trainer_id int PRIMARY KEY,  
first_name varchar NOT NULL,  
last_name varchar,
```

```
        address varchar(30),  
        mobile_no bigint(10) NOT NULL,  
        email varchar,  
        animal_ids int NOT NULL,  
        weekly_working_hours int  
)
```

```
CREATE Table Adopter(  
    adopter_id int PRIMARY KEY,  
    first_name varchar NOT NULL,  
    last_name varchar,  
    address varchar(30),  
    mobile_no bigint(10) NOT NULL,  
    email varchar,  
    date_of_adoption date NOT NULL  
    amount_paid int NOT NULL,  
    FOREIGN KEY (staff_ID) REFERENCES Staff,  
        ON DELETE CASCADE  
)
```

```
CREATE Table Doctor(  
    doctor_id int PRIMARY KEY,  
    first_name varchar NOT NULL,  
    last_name varchar,  
    address varchar(30),  
    mobile_no bigint(10) NOT NULL,  
    email varchar,  
    license_no bigint NOT NULL  
    speciality varchar NOT NULL,  
    weekly_working_hours int  
)
```

```
CREATE Table Treatment Record(  
    PRIMARY KEY (animal_ID,doctor_ID),  
    FOREIGN KEY (animal_ID) REFERENCES Animal Profile,  
        ON DELETE CASCADE  
    FOREIGN KEY (doctor_ID) REFERENCES Doctor,  
        ON DELETE CASCADE  
    treatment_date date NOT NULL,  
    cost int NOT NULL  
)
```

```
CREATE TABLE Medical Record
```

```
(  
    Treatment_date varchar(10)  
    Cost INTEGER NOT NULL
```

)

```
CREATE TABLE staff
(
    Staff_ID int PRIMARY KEY
    First_name varchar(20)
    Last_name varchar(20)
    Address varchar(50)
    Age INTEGER
    Mobile_number bigint NOT NULL
    Email varchar(15)
    Position varchar(10)
    Start_day_of_job varchar(10)
    Salary INTEGER NOT NULL
)
```

CREATE TABLE sponsor

```
(

    Sponsor_ID INTEGER PRIMARY KEY
    First_name varchar(20)
    Last_name varchar(20)
    Address varchar(50)
    Mobile_number bigint NOT NULL
    Email varchar(20)
    Position varchar(10)
    Start_day_of_sponsoring varchar(10)
    Period_of_sponsoring varchar(10) NOT NULL
    Animal_IDs bigint

)
```

CREATE TABLE cage

```
(

    Cage_no INTEGER PRIMARY KEY
    Length INTEGER NOT NULL
)
```

```
Height INTEGER NOT NULL  
Width INTEGER NOT NULL  
Animal IDs bigint  
)  
;
```

Insert Statements :

- 1) COPY animal_profile
(id,name,sex,age,breed,staff_id,adopter_id,donor_id,volunteer_id,cage_no,tra
iner_id,sponsor_id,doctor_id)
FROM 'C:\Users\Public\DB data\Animal_Profile.csv'
DELIMITER ','
CSV HEADER;
- 2) COPY cage (cage_no,length,height,width)
FROM 'C:\Users\Public\DB data\Cage.csv'
DELIMITER ','
CSV HEADER;
- 3) COPY doctor
(doctor_id,first_name,last_name,address,mobile_no,email,license_no,specialit
y,weekly_working_hours)
FROM 'C:\Users\Public\DB data\Doctor.csv'
DELIMITER ','
CSV HEADER;
- 4) COPY medical_record (treatment_date,cost)
FROM 'C:\Users\Public\DB data\Treatment_Record.csv'
DELIMITER ','
CSV HEADER;
- 5) COPY staff
(staff_id,first_name,last_name,address,age,mobile_no,email,position,start_da
y_of_job,salary)

```
FROM 'C:\Users\Public\DB data\Staff.csv'  
DELIMITER ','  
CSV HEADER;
```

6) COPY donor
(donor_id,first_name,last_name,address,mobile_number,email,date_of_donation,donation_detail,staff_id)
FROM 'C:\Users\Public\DB data\Donor.csv'

```
DELIMITER ','
```

```
CSV HEADER;
```

7) COPY trainer

```
(trainer_id,first_name,last_name,address,mobile_no,email,weekly_working_hours)
```

```
FROM 'C:\Users\Public\DB data\Trainer.csv'
```

```
DELIMITER ','
```

```
CSV HEADER;
```

8)

```
COPY volunteer
```

```
(volunteer_id,first_name,last_name,address,mobile_number,email,start_date,end_date,working_hours)
```

```
FROM 'C:\Users\Public\DB data\Volunteer.csv'
```

```
DELIMITER ','
```

```
CSV HEADER;
```

9) COPY adopter

```
(adopter_id,first_name,last_name,address,mobile_no,email,amount_paid,staff_id,date_of_adoption)
```

```
FROM 'C:\Users\Public\DB data\Adopter.csv'
```

```
DELIMITER ','
```

```
CSV HEADER;
```

10) COPY treatment_record (animal_id,doctor_id,cost,treatment_date)

```
FROM 'C:\Users\Public\DB data\Treatment_Record.csv'
```

```
DELIMITER ','
```

```
CSV HEADER;
```

```
11) COPY sponsor
(sponsor_id,first_name,last_name,address,mobile_number,email,position,start
 _day_of_sponsoring,period_of_sponsoring)
FROM 'C:\Users\Public\DB data\Sponsor.xlsx'
DELIMITER ','
CSV HEADER;
```

1. Find animals of Pug breed

→ select * from animal_profile where breed='Pug';

The screenshot shows the pgAdmin 4 interface with a database browser on the left and a query editor on the right. The query editor contains the SQL command: `select * from animal_profile where breed='Pug';`. The results are displayed in a table titled 'Data Output' with the following columns: id, name, sex, age, breed, staff_id, adopter_id, donor_id, volunteer_id, and page_no. The data shows 11 rows of animal profiles, all of which are Pugs (breed='Pug').

id	name	sex	age	breed	staff_id	adopter_id	donor_id	volunteer_id	page_no
17	1015 Brandon	1	18	Pug	5640	3075	10075	571	
18	1026 Paul	0	6	Pug	5647	3076	10076	573	
19	1032 Felix	0	17	Pug	5649	3082	10082	573	
20	1038 Ginger	0	18	Pug	5644	3088	10088	571	
21	1092 Acme	0	5	Pug	5656	3092	10092	995	
22	1094 Britlee	0	6	Pug	5622	3094	10094	500	
23	1101 Olida	1	4	Pug	5642	3096	10099	575	

2. Count the number of animals that are male (sex=1)

→ select count(*) from animal_profile where sex='1';

The screenshot shows the pgAdmin 4 interface with a database browser on the left and a query editor on the right. The query editor contains the SQL command: `select count(*) from animal_profile where sex='1';`. The results are displayed in a table titled 'Data Output' with one row showing a count of 51.

count
51

3. Animals with age <10 years

→ select * from animal_profile where age<10

The screenshot shows the PgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the left sidebar, the 'Tables (11)' section is expanded, and 'animal_profile' is selected. Under 'Columns (13)', all columns are listed. A query is run in the 'Query Editor' pane:

```
1 select * from animal_profile where age<10
```

The results are displayed in a table titled 'Data Output' with 13 columns corresponding to the table's schema. The columns are: id, name, sex, age, breed, staff_id, adopter_id, donor_id, volunteer_id, cager_id, sponsor_id, doctor_id, and cage_no. The data shows several rows of animal profiles, with ages ranging from 0 to 10.

id	name	sex	age	breed	staff_id	adopter_id	donor_id	volunteer_id	cager_id	sponsor_id	doctor_id	cage_no
42	1087	Gay	1	4	5664	3087	10067	527				
43	1089	Ursula	0	1	5662	3089	10069	571				
44	1091	Branden	0	6	5612	3091	10091	585				
45	1092	Aceme	0	5	5656	3092	10062	595				
46	1093	Anella	0	8	5644	3093	10093	570				
47	1094	Brielle	0	6	5622	3094	10094	503				
48	1101	Olivia	1	4	5692	3096	10069	575				

4. All the animals nurtured by some specific volunteers

→ select * from animal_profile where volunteer_id>510 and volunteer_id<530

The screenshot shows the PgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the left sidebar, the 'Tables (11)' section is expanded, and 'animal_profile' is selected. Under 'Columns (13)', all columns are listed. A query is run in the 'Query Editor' pane:

```
1 select * from animal_profile where volunteer_id>510 and volunteer_id<530
```

The results are displayed in a table titled 'Data Output' with 13 columns corresponding to the table's schema. The columns are: id, name, sex, age, breed, staff_id, adopter_id, donor_id, volunteer_id, cager_id, sponsor_id, doctor_id, and cage_no. The data shows several rows of animal profiles, with volunteer IDs ranging from 510 to 530.

id	name	sex	age	breed	staff_id	adopter_id	donor_id	volunteer_id	cager_id	sponsor_id	doctor_id	cage_no
22	1077	Mollie	0	15	5682	3077	10077	526				
23	1082	Felix	0	1	5699	3082	10062	513				
24	1087	Gay	1	4	5664	3087	10067	527				
25	1088	Ginger	0	18	5644	3088	10068	511				
26	1090	Bryar	1	15	5606	3090	10060	512				
27	1096	Cody	1	19	5615	3096	10066	526				
28	1103	Mollie	0	15	5682	3077	10022	526				

5. All adopters who paid amount > 10000

→ select * from adopter where amount_paid>10000

The screenshot shows the PgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the 'Query Editor' tab, the following SQL query is run:

```
1 select * from adopter where amount_paid>10000
```

The results are displayed in a table titled 'Data Output' with the following columns:

adopter_id	first_name	last_name	address	mobile_no	email	amount_paid	staff_id	date_of_adoption
84	Haley	Cleveland	2691 Pede Rd.	3690246200	vulpure.maurit sagittis	79017	5642	02/18/2021
85	Basil	Foley	Ap #602-6558 A Rd.	7544024708	fringilla@placeret.net	85897	5557	05/23/2021
86	Delilah	Savage	141-6158 Vulputate St.	8059818211	mijat.ca	97966	5614	03/05/2020
87	Cherie	Roberson	P.O. Box 535, 2373 Ante Av.	3843919555	vel@lectus.nra.edu	81398	5624	10/18/2020
88	Karina	Small	Ap #286-7114 Elit, Ave	9725913786	Suspendisse.turpis,n.	56992	5539	07/28/2020
89	Megree	York	Ap #732-873 Aliquam Ave	2728010041	mi@fringilla.mi@Quesqu...	87615	5684	01/20/2021
90	Clinton	Dalton	733-177 Velti Road	58187845	Lorem ipsum@Nullemw...	92360	5693	05/21/2023

6. Finding maximum height of cage

→ select max(height) from cage

The screenshot shows the PgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the 'Query Editor' tab, the following SQL query is run:

```
1 select max(height) from cage
```

The results are displayed in a table titled 'Data Output' with the following column:

max
20

7. Finding maximum volume of cage

→ select max(height*length*width) from cage

The screenshot shows the PgAdmin 4 interface. On the left, the 'Browser' pane lists database objects: Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Sequences, Tables (11), and cage. Under 'cage', it shows columns: cage_no, length, height, and width. The 'Query Editor' pane contains the SQL query: `select max(height*length*width) from cage;`. The 'Data Output' pane shows the result of the query:

max
integer
5814

. The system tray at the bottom right indicates the date as 28-11-2020 and the time as 09:34.

8. List all the cages with length > 10

→ select * from cage where length>10

The screenshot shows the PgAdmin 4 interface. The 'Browser' pane is identical to the previous screenshot. The 'Query Editor' pane contains the SQL query: `select * from cage where length>10;`. The 'Data Output' pane shows the results of the query:

cage_no	length	height	width
42	5089	19	16
43	5090	20	8
44	5091	15	1
45	5092	18	12
46	5093	12	12
47	5094	12	10
48	5095	12	4
49	5096	12	9

. The system tray at the bottom right indicates the date as 28-11-2020 and the time as 09:35.

9. All the doctors that are available for more than 35 hours per week → select * from doctor where weekly_working_hours>35

```
1 select * from doctor where weekly_working_hours>35
```

doctor_id	first_name	last_name	address	mobile_no	email	license_no	specialty	weekly_working_hours
26	4081	Inez	Leonard	Ap #800-7517 Sed Rd.	8906631224	Nullam@gravideauric...	5	XNR3WULF9IE
27	4083	Abel	Griffin	379-7702 Donec Rd.	1486098532	diam.diculum.sagittis@...	1	EW6SPSB1MK
28	4084	Shee	Marks	840 Dolor Street	3498808376	adipiscing@apurus.net	7	TXG34HQABEC
29	4085	Justina	Mendez	7892 Vell Av.	9859275281	neque@quam.co.uk	1	INR20VGV1GR
30	4087	Leth	Newton	P.O. Box 432, 6133 Non Road	449592220	Fusce dolor rutrum or...	2	PIU2MRC0CJ
31	4095	Veda	Klein	Ap #260-7349 Curae; Rd.	7946112193	Igula.Aliquam.enit@all...	2	LJQ90DPV1VD
32	4098	Constance	Dunn	P.O. Box 112, 9951 Non Street	6472457802	Pellentesque@massen...	6	USP7SAEST1VZ

10. Date of donation of a specific donor

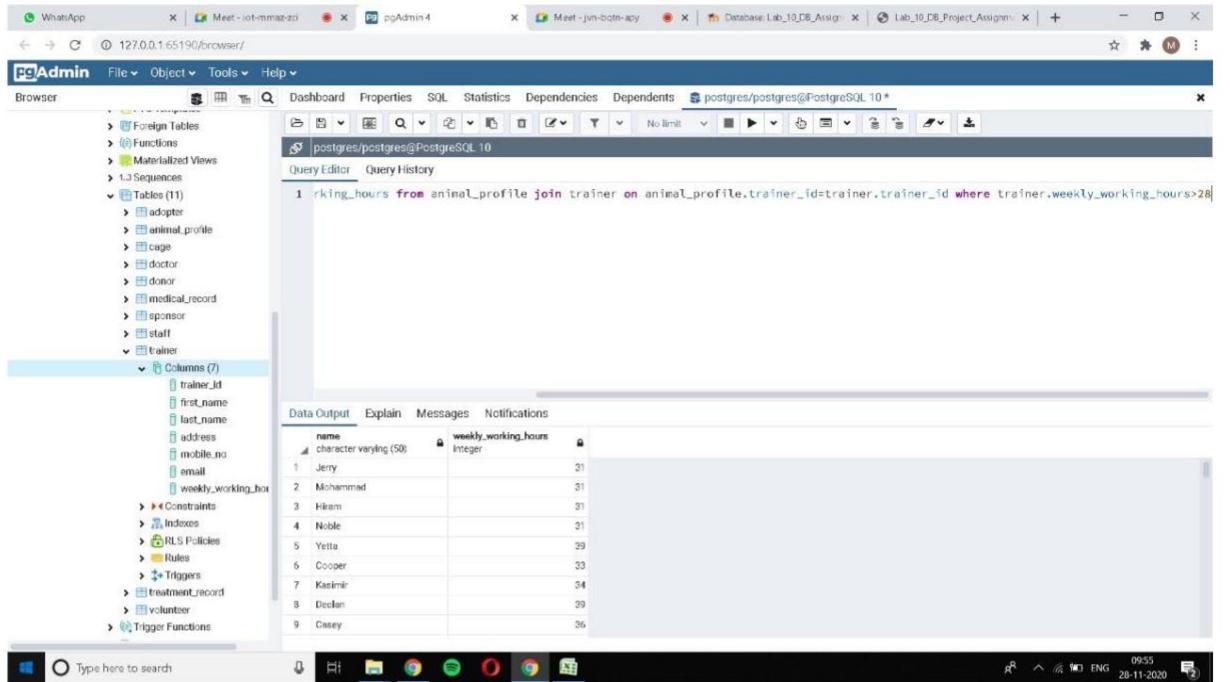
→ select date_of_donation from donor where donor_id=10078

```
1 select date_of_donation from donor where donor_id=10078
```

date_of_donation
1 09-05-2020

11. Find the names of animals whose trainer's training is more than 28 hours in a week.

→ select animal_profile.name, trainer.weekly_working_hours from animal_profile
join trainer on animal_profile.trainer_id=trainer.trainer_id where
trainer.weekly_working_hours>28

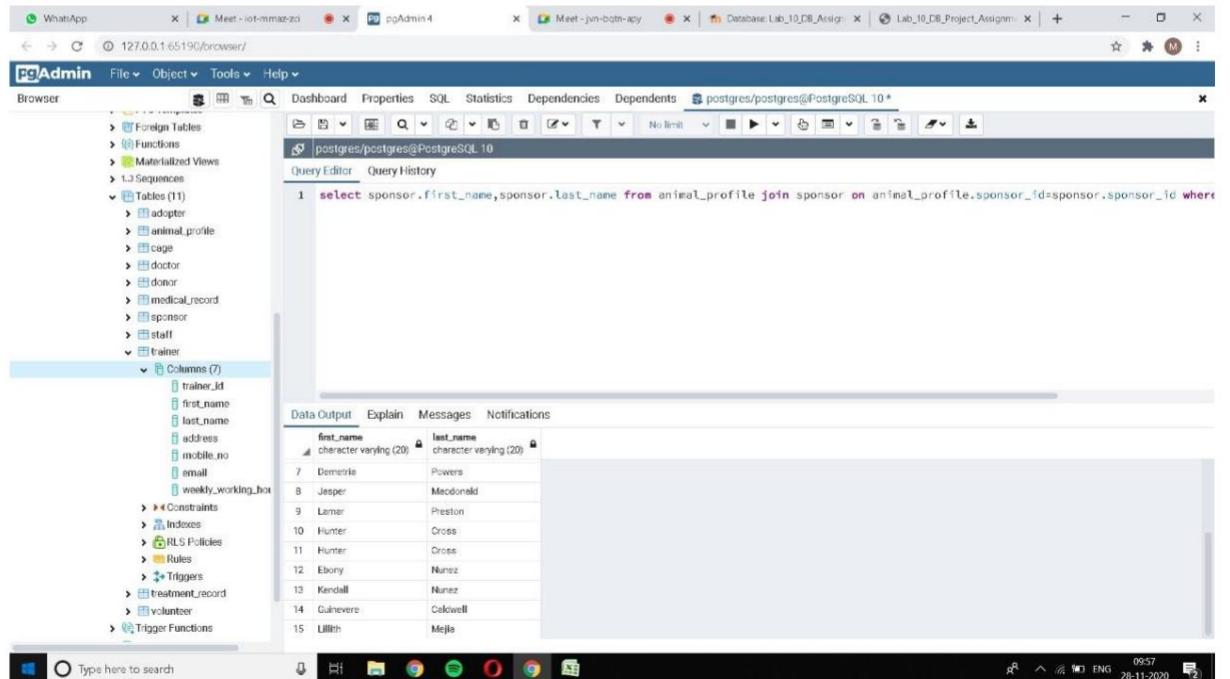


```
1   rking_hours from animal_profile join trainer on animal_profile.trainer_id=trainer.trainer_id where trainer.weekly_working_hours>28
```

name	weekly_working_hours
Jerry	31
Mohammed	31
Hiram	31
Noble	31
Yetta	39
Cooper	33
Kasimir	34
Declan	29
Casey	36

12. Find the first name of sponsors sponsoring the breed 'Desi'

→select sponsor.first_name,sponsor.last_name from animal_profile join sponsor on animal_profile.sponsor_id=sponsor.sponsor_id where animal_profile.breed='Desi'



```
1   select sponsor.first_name,sponsor.last_name from animal_profile join sponsor on animal_profile.sponsor_id=sponsor.sponsor_id where animal_profile.breed='Desi'
```

first_name	last_name
Demetria	Powers
Jesper	Macdonald
Lamar	Preston
Hunter	Cross
Hunter	Grose
Ebony	Nunez
Kendall	Nunez
Guinevere	Cakewell
Ullith	Mejia

13. Find the name of most senior position staff members (3 is highest position)

→ select * from staff where position = '3';

The screenshot shows the pgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the left sidebar, under 'Tables (11)', the 'staff' table is selected. The 'Columns (7)' section is expanded, showing columns: staff_id, first_name, last_name, address, mobile_no, email, and weekly_working_hours. A query is run in the 'Query Editor' tab:

```
select * from staff where position = '13';
```

The results show 28 rows of staff members. The columns are: staff_id, first_name, last_name, address, age, mobile_number, email, and position. The data includes various names like Yoko, Frances, Jacqueline, Tatiana, Baker, Georgi, and Colly, along with their addresses and contact details.

14. Names of all staff dealing with Donors

→ select staff.first_name,staff.last_name from staff join donor
on staff.staff_id=donor.staff_id

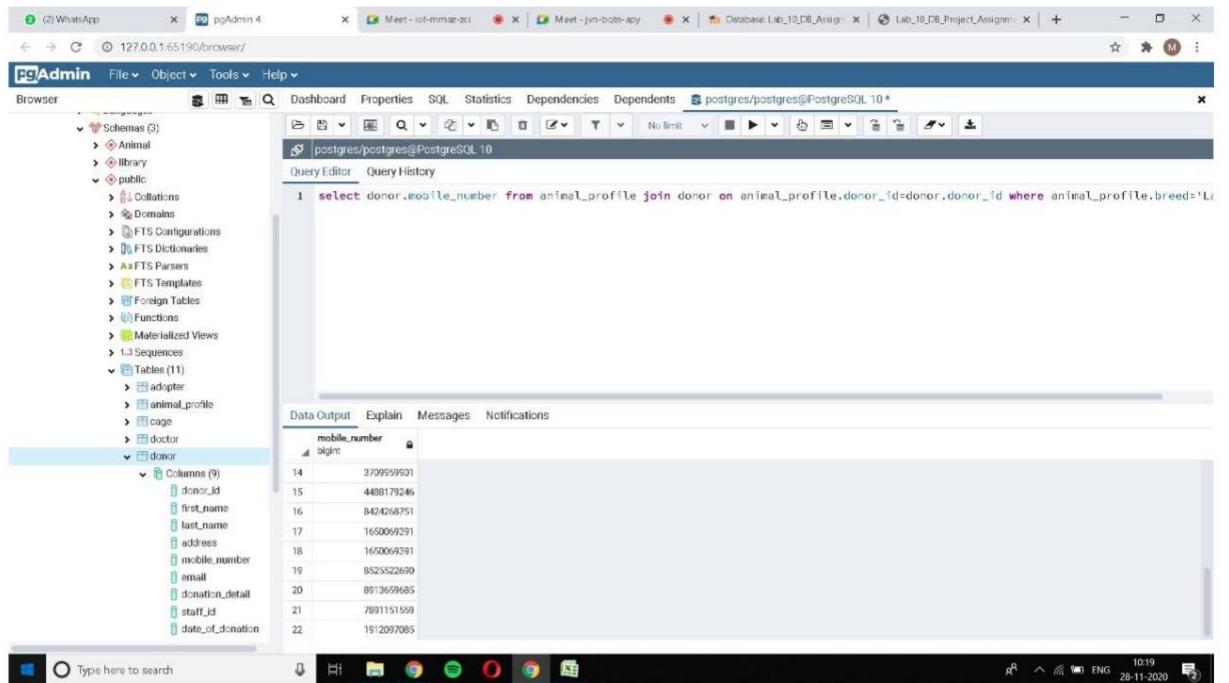
The screenshot shows the pgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the left sidebar, under 'Tables (11)', the 'staff' table is selected. The 'Columns (9)' section is expanded, showing columns: staff_id, first_name, last_name, address, mobile_no, email, amount_paid, staff_id, and date_of_adoption. A query is run in the 'Query Editor' tab:

```
select staff.first_name,staff.last_name from staff join donor on staff.staff_id=donor.staff_id;
```

The results show 100 rows of staff members dealing with donors. The columns are: first_name and last_name. The data includes names like Seda, Alvin, Aline, Forrest, Boris, Fritz, Jelani, Alan, and Lucy, listed sequentially.

15. Mobile number of all donors who donated labrador

→ select donor.mobile_number from animal_profile join donor on animal_profile.donor_id=donor.donor_id where animal_profile.breed='Labrador'

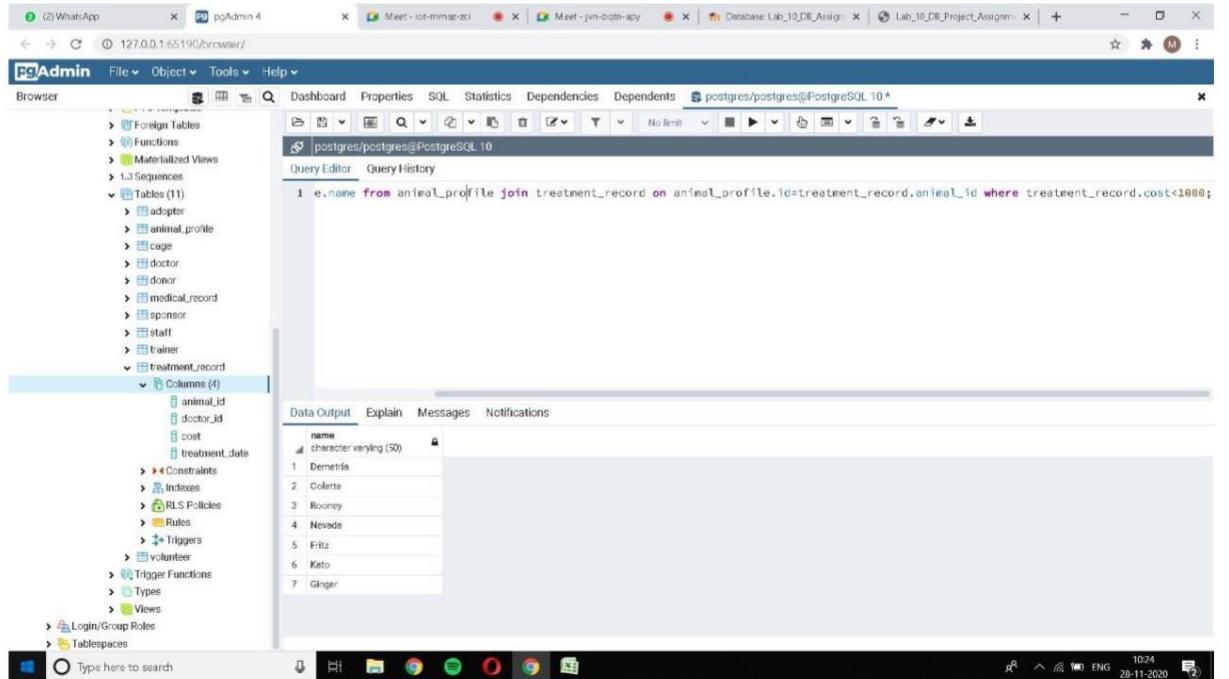


```
1 select donor.mobile_number from animal_profile join donor on animal_profile.donor_id=donor.donor_id where animal_profile.breed='Labrador'
```

mobile_number
3709959501
4480179246
842468751
1650069391
1650069391
852522690
8913659685
7891151559
1912097085

16. Names of animals whose treatment cost is less than 1000

→ select animal_profile.name from animal_profile join treatment_record on animal_profile.id=treatment_record.animal_id where treatment_record.cost<1000;



```
1 e.name from animal_profile join treatment_record on animal_profile.id=treatment_record.animal_id where treatment_record.cost<1000;
```

name
Demetria
Collette
Rooney
Nevade
Fritz
Keto
Ginger

17. Name of animal nurtured by the volunteer who has maximum weekly working hours

→ select animal_profile.name from animal_profile join volunteer on animal_profile.volunteer_id=volunteer.volunteer_id order by volunteer.working_hours desc limit 1;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like adopter, animal_profile, cage, doctor, donor, medical_record, sponsor, staff, trainer, treatment_record, and volunteer. The 'volunteer' table is selected, and its columns (volunteer_id, first_name, last_name, address, mobile_number, email, working_hours, start_date, end_date) are listed. The main window contains a query editor with the following SQL code:

```
1 animal_profile join volunteer on animal_profile.volunteer_id=volunteer.volunteer_id order by volunteer.working_hours desc limit 1;
```

The results pane shows a single row with the name 'Cyrus'.

18. List adopters by the descending number of animals they adopted →

select adopter.first_name,adopter.last_name,count(*) as quantity
from animal_profile join adopter on adopter.adopter_id=animal_profile.adopter_id
group by adopter.first_name,adopter.last_name,adopter.adopter_id order by
quantity desc;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like adopter, animal_profile, cage, doctor, donor, medical_record, sponsor, staff, trainer, treatment_record, and volunteer. The 'adopter' table is selected, and its columns (adopter_id, first_name, last_name, address, mobile_number, email, working_hours, start_date, end_date) are listed. The main window contains a query editor with the following SQL code:

```
1 select adopter.first_name,adopter.last_name,count(*) as quantity
2 from animal_profile
3 join adopter on adopter.adopter_id=animal_profile.adopter_id
4 group by adopter.first_name,adopter.last_name,adopter.adopter_id
5 order by quantity desc;
```

The results pane shows a list of adopters with their names and the count of animals adopted:

first_name	last_name	quantity
Karlie	Small	2
Justina	Resse	2
Keelie	Wolf	2
Hiram	Lopez	2
Lamar	Huffman	2
Julie	Holeen	1
Raymond	Steele	1
Rina	Carroll	1
Kerry	Vaughn	1

19. List of volunteers that have not nurtured any animals

→ Select volunteer.first_name,volunteer.last_name from volunteer where volunteer_id not in (select volunteer_id from animal_profile)

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database schema, including 'Schemas (3)', 'Animal', 'library', and 'public' (which is expanded to show 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Sequences', and 'Tables (11)'). The 'Tables (11)' section contains entries like 'adopter', 'animal_profile', 'cage', 'doctor', 'donor', 'medicalRecord', 'sponsor', 'staff', 'trainer', 'treatment_record', 'volunteer', 'Trigger Functions', 'Types', and 'Views'. The 'animal_profile' table is selected. On the right, the 'Query Editor' pane contains the SQL query:

```
1 Select volunteer.first_name,volunteer.last_name from volunteer where volunteer_id not in (select volunteer_id from animal_profile)
```

The results are displayed in a table titled 'Data Output' with columns 'first_name' and 'last_name'. The data is as follows:

first_name	last_name
Tiger	Hardy
Zephania	Hahn
Teegan	Higgles
Whalemina	Robbins
Quinn	Cerdon
Kyle	Flores
Luke	Pugh
Nichole	Curry
Dacey	Travis

20. List of all animals that have not received any treatment yet

→ Select name from animal_profile where id not in (select animal_id from treatment_record)

The screenshot shows the pgAdmin 4 interface. The 'Browser' pane is identical to the previous screenshot, showing the 'animal_profile' table as selected. The 'Query Editor' pane contains the SQL query:

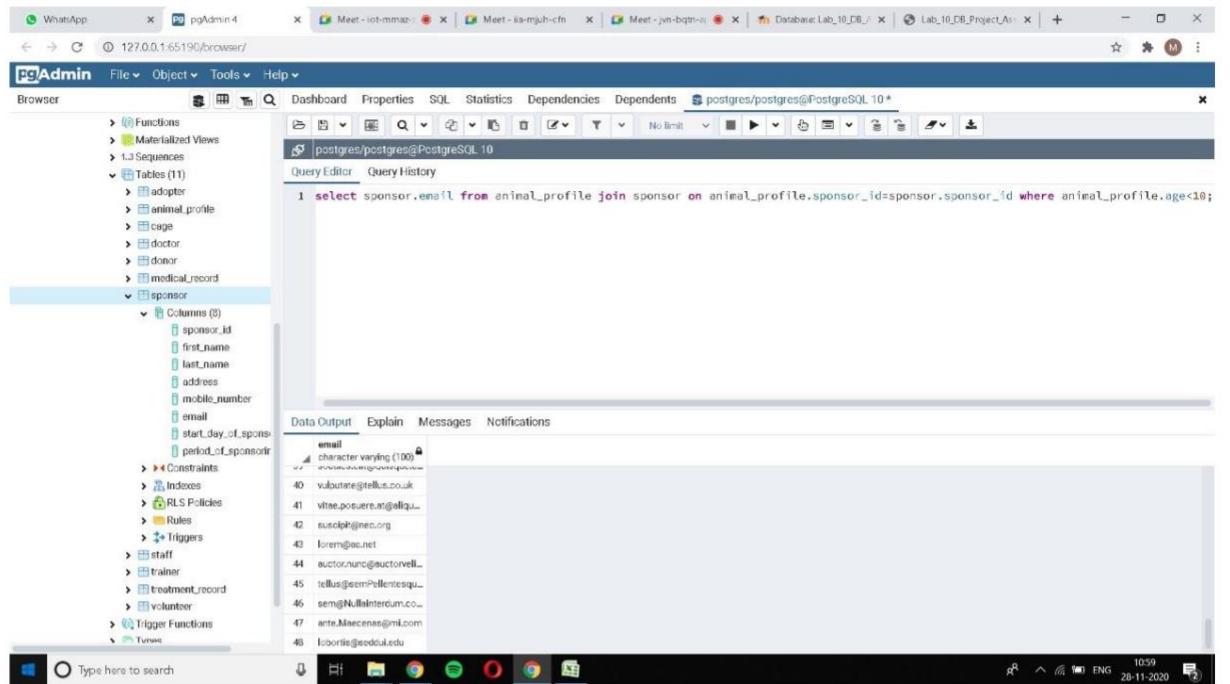
```
1 Select name from animal_profile where id not in (select animal_id from treatment_record)
```

The results are displayed in a table titled 'Data Output' with a single column 'name'. The data is as follows:

name
Felix
Graeden
Hadesrah
Basha
Joseph
Ancilla
Ray
Rana
Sasha

21. Emails of all sponsors that have sponsored all animals of age less than 10 years

→ select sponsor.email from animal_profile join sponsor on animal_profile.sponsor_id=sponsor.sponsor_id where animal_profile.age<10;



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

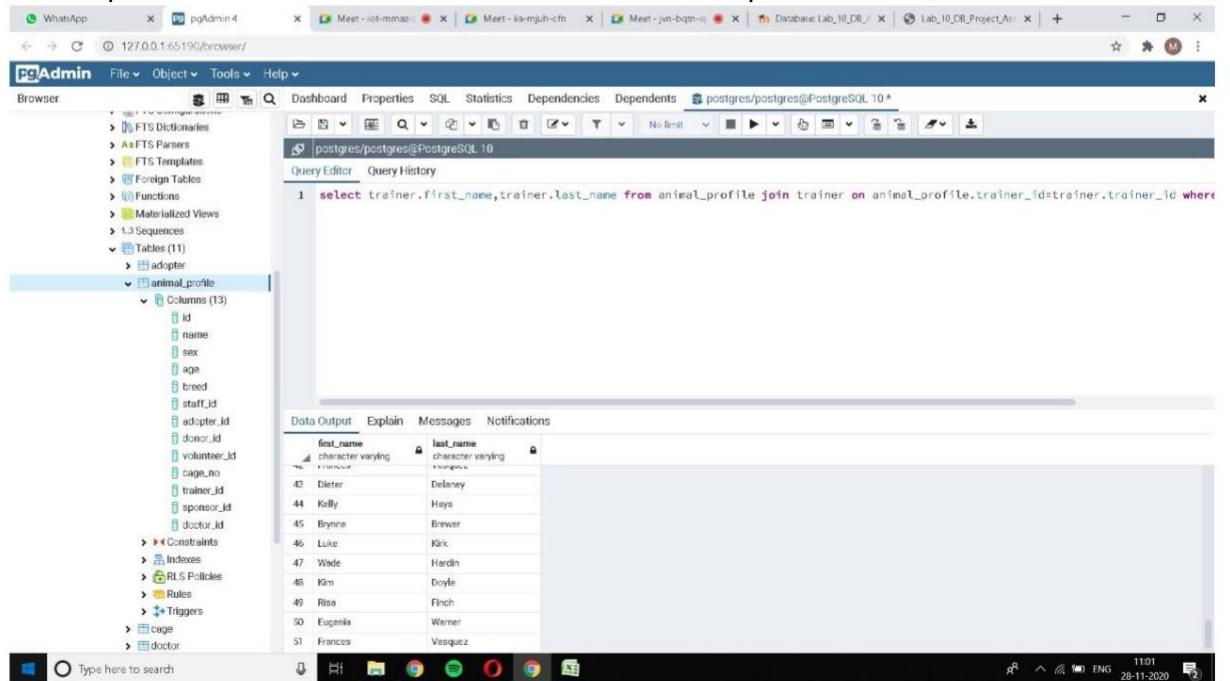
```
1 select sponsor.email from animal_profile join sponsor on animal_profile.sponsor_id=sponsor.sponsor_id where animal_profile.age<10;
```

The results table has columns 'email' and 'character varying (100)'. The data is:

email	character varying (100)
40	vulpinne@telus.co.uk
41	Vitte.pouurrat@liqui...
42	suselph@meo.org
43	foremg@ac.net
44	autor.nunc@autorevelli...
45	telus@comptentesqu...
46	sem@NullInterium.co...
47	ante.Maccus@gmail.com
48	Iborilis@eedcau.edu

22. List of trainers training all male animals

→ select trainer.first_name, trainer.last_name from animal_profile join trainer on animal_profile.trainer_id=trainer.trainer_id where animal_profile.sex='1';



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 select trainer.first_name, trainer.last_name from animal_profile join trainer on animal_profile.trainer_id=trainer.trainer_id where animal_profile.sex='1';
```

The results table has columns 'first_name' and 'last_name'. The data is:

first_name	last_name
43	Briar
44	Kelly
45	Brynnie
46	Luke
47	Wade
48	Kim
49	Rita
50	Eugenia
51	Frances

23. Cage number of animals which has treatment record cost greater than 5000

→ select animal_profile.cage_no from animal_profile join treatment_record on animal_profile.id=treatment_record.animal_id where treatment_record.cost>5000;

The screenshot shows the PgAdmin 4 interface. The left sidebar displays the database schema with the 'animal_profile' table selected. The main area contains a query editor with the following SQL code:

```
1 cage_no from animal_profile join treatment_record on animal_profile.id=treatment_record.animal_id where treatment_record.cost>5000;
```

The results pane shows the following data output:

cage_no
39
5044
40
5039
41
5033
42
5031
43
5090
44
5045
45
5085
46
5058
47
5092

24. All the days when treatment cost is greater than

6000 → select * from medical_record where cost>6000

The screenshot shows the PgAdmin 4 interface. The left sidebar displays the database schema with the 'medical_record' table selected. The main area contains a query editor with the following SQL code:

```
1 select * from medical_record where cost>6000;
```

The results pane shows the following data output:

treatment_date	cost
27	6016
28	9689
29	7875
30	9912
31	8009
32	9549
33	7304
34	7943
35	9876

25. Count of all the adopters whose last name ends with 'n'

→ select count(*) from adopter where last_name like '%n'

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like adopter, animal_profile, cage, doctor, donor, medical_record, and sponsor. The sponsor table is currently selected. The main area shows a query editor with the following SQL command:

```
1 select count(*) from adopter where last_name like '%n'
```

The Data Output tab shows the result of the query:

count	blight
1	24

The status bar at the bottom right indicates the date and time as 28-11-2020 11:07.

26. List of all animals with breed 'labrador' and age less than 10 years →

Select * from animal_profile where breed = 'Labrador' and age<10

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like adopter, animal_profile, cage, doctor, donor, medical_record, and sponsor. The animal_profile table is currently selected. The main area shows a query editor with the following SQL command:

```
1 Select * from animal_profile where breed = 'Labrador' and age<10
```

The Data Output tab shows the results of the query, listing several animals with their details:

id	name	sex	age	breed	staff_id	adopter_id	donor_id	volunteer_id	cage_no
1	1013 Declan	0	6	Labrador	5671	3013	10013	508	51
2	1014 Hella	1	1	Labrador	5669	3014	10014	595	51
3	1015 Aquila	1	9	Labrador	5653	3015	10015	583	51
4	1016 TeShya	0	7	Labrador	5665	3016	10016	599	51
5	1047 Cyrus	0	2	Labrador	5676	3047	10047	519	51
6	1056 Deacon	0	2	Labrador	5671	3096	10096	522	51
7	1067 Gay	1	4	Labrador	5664	3087	10087	527	51

The status bar at the bottom right indicates the date and time as 28-11-2020 11:08.

27. List of sponsors whose period of sponsors is greater than 10 months

→ select * from sponsor where period_of_sponsoring>'10';

The screenshot shows the pgAdmin interface with a query editor window. The query is:

```
1 select * from sponsor where period_of_sponsoring>'10';
```

The results table has the following columns and data:

spnsor_id	first_name	last_name	address	mobile_number	email	start_day_of_sponsoring	period_of_sponsoring	
84	10315	Daria	Cochran	763-9460 Aliquet Rd.	706777908	telus@openPolicies.com	12/17/22	16
85	10316	Felix	Yang	9927 Delibut Road	1319055327	sem@Nullinterdum.co...	01/04/23	21
86	10317	Martin	Blanchard	Ap #687-5078 Maltese Rd...	6112715447	enix.Mecenes@gmail.com	09/11/23	22
87	10318	Lillith	Melje	P.O. Box 310-4113 Neuge...	5396416586	SuspendedSee,adgeuenL...	06/01/23	21
88	10319	Oren	Shelton	Ap #754-2934 Risus Ave	8681511291	dictum.Phasellus.In@mj...	11/01/21	6
89	10320	Amelia	Boyd	2818 Vhemus St.	397120045	scilicet@atque@odil.c...	12/16/19	23
90	10321	Blossom	Valencia	124-1052 Nec Ave	8889861341	nunc@maurisDolorpor...	06/28/21	24
91	10322	Tanner	Alexander	Ap #825-6569 Penitibus Av.	565580941	ligula.consectetur@ho...	05/24/21	2

28. List the staff in order of their salary

→ select first_name,last_name,salary from staff order by salary desc

The screenshot shows the pgAdmin interface with a query editor window. The query is:

```
1 select first_name,last_name,salary from staff order by salary desc
```

The results table has the following columns and data:

first_name	last_name	salary
Kieran	Hull	99810
Myra	Wilder	99783
Katelyn	Bredford	98408
Tatyana	Swanson	97667
Samuel	Mayer	96721
Jean	Hanson	96482
Genevieve	Mayer	96044
Fritz	Shaffer	95379
Rhoda	Wagner	94194
Nancy	Wagner	93494

29. List the animals in descending order of age

→ select name,age from animal_profile order by age desc

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database schema, including 'Schemas (3)', 'Tables (11)', and 'doctor' (which is currently selected). The 'Query Editor' pane contains the SQL query: 'select name,age from animal_profile order by age desc'. The 'Data Output' pane shows the results:

name	age
Gralden	19
Ray	19
Cody	19
Cooper	19
Ainstley	19
Sasha	19
Chester	19
Sasha	19
Ginger	18
CB	10

30. Speciality of all doctors whose weekly working hours are greater than 30

30 → select first_name,last_name,speciality from doctor where weekly_working_hours>30

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database schema, including 'Schemas (3)', 'Tables (11)', and 'doctor' (which is currently selected). The 'Query Editor' pane contains the SQL query: 'select first_name,last_name,speciality from doctor where weekly_working_hours>30'. The 'Data Output' pane shows the results:

first_name	last_name	speciality
Minerva	McDowell	SDW2BX9XMIU
Kai	Bradley	FQ02PCE0LD
Chava	Brown	PXZ3M6C6MR
Althea	Ross	NPX81BY3ZA
Keston	Michael	AUG5BDA23MP
Madaline	Wise	ECC23TRZ29L
Amber	Richmond	NYE08QXF3RY
Emmanuel	Huber	KCY79FBM1QV
Alecia	Freeman	FFZ22OPM2ZH

31. List of all doctors with .com in their emails

→ select * from doctor where email like '%.com'

The screenshot shows the pgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the 'Tables (11)' section, the 'doctor' table is selected. A query is run in the 'Query Editor' tab:

```
1 select * from doctor where email like '%.com'
```

The results are displayed in a table titled 'Data Output' with the following columns: doctor_id, first_name, last_name, address, mobile_no, email, amount_paid, staff_id, date_of_adoption, license_no, speciality, and weekly_working_hours. The results show 12 rows of data.

doctor_id	first_name	last_name	address	mobile_no	email	amount_paid	staff_id	date_of_adoption	license_no	speciality	weekly_working_hours
5	4040	Jones	Lane	4171349228	mauris.Suspendisse.allii...	5	WE754WQE3SE				
6	4042	Bradley	Cole	9254739093	arcu.Uncert...	3	PQV96XJ4BV				
7	4056	Roger	Barrera	8679441	Elt, St.	4699201053	una.Nullam lobortis...		8	EZP73JKJLX	
8	4062	Scott	Muller	Ap#632-4701	Est, Street	4453801731	Infaudous@consequat...		10	WIN32MMNSWD	
9	4067	Maryam	Porte	7284 Lacus Rd.		9071162514	enim@elementum.com		2	KSA30LYB5CIE	
10	4068	Julie	Orr	P.O. Box 119	2384 Praesent...	6270603332	congue.ellis.edeu...		2	SPD54NVV3CG	
11	4090	Chase	Beard	Ap#100-6147	Quisque Street	5962854643	Prolinued.turple@ment...		4	HR028KQ55PY	
12	4091	Laurel	Sellers	458-9584	In Road	1521257762	magnis@metufacili...		7	XH153FNJLJU	

32. Number of all trainers who works for more than 30 weekly hours

→ select count(*) from trainer where weekly_working_hours>30

The screenshot shows the pgAdmin 4 interface with a database connection to 'postgres/postgres@PostgreSQL 10'. In the 'Tables (11)' section, the 'trainer' table is selected. A query is run in the 'Query Editor' tab:

```
1 select count(*) from trainer where weekly_working_hours>30
```

The results are displayed in a table titled 'Data Output' with the following column: count. The result shows 1 row with a value of 62.

count
62

33. Animal's names whose cage length is greater than 10

→ select animal_profile.name from animal_profile join cage on animal_profile.cage_no = cage.cage_no where cage.length>10;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like animal_profile, cage, and trainer. The main area shows a query editor with the following SQL code:

```
1 select animal_profile.name from animal_profile join cage on animal_profile.cage_no = cage.cage_no where cage.length>10;
```

The results pane shows the output of the query:

name
Gay
Ursula
Branden
Adena
Amelia
Brielle
Cody
Sonia
Hadasah

34. List of all animals whose adopter paid amount greater than 90050

→ select animal_profile.name from animal_profile join adopter on animal_profile.adopter_id = adopter.adopter_id where adopter.amount_paid>90050

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like adopter, animal_profile, and donor. The main area shows a query editor with the following SQL code:

```
1 al_profile.name from animal_profile join adopter on animal_profile.adopter_id = adopter.adopter_id where adopter.amount_paid>90050
```

The results pane shows the output of the query:

name
Abraham
Fitzgerald
Sheron
Sasha
Gralden
Gay
Bryar
Brielle
Uta

35. Total amount received from adopters

→ select sum(amount_paid) from adopter

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with the 'adopter' table selected. The main area shows a query editor with the following SQL code:

```
1 select sum(amount_paid) from adopter;
```

The results pane shows the output of the query:

sum	bright
1	554925

The status bar at the bottom right indicates the date and time as 28-11-2020 11:26.

36. Total cost of animal treatment

→ select sum(cost) from treatment_record

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with the 'treatment_record' table selected. The main area shows a query editor with the following SQL code:

```
1 select sum(cost) from treatment_record;
```

The results pane shows the output of the query:

sum	bright
1	518115

The status bar at the bottom right indicates the date and time as 28-11-2020 11:28.

37. Total salaries of all employees combined

→ select sum(salary) from staff

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane lists various database objects under the 'public' schema, including tables like 'staff'. In the center, the 'Query Editor' pane contains the SQL query: 'select sum(salary) from staff;'. The 'Data Output' pane shows the result: a single row with 'sum' and a value of '5786600'. The system tray at the bottom right indicates the date and time as 28-11-2020 11:28.

```
1 select sum(salary) from staff
1   sum
  5786600
```

38. Starting day of working of all staff members with highest job position

→ select start_day_of_job from staff where position = '1';

The screenshot shows the pgAdmin 4 interface. The 'Browser' pane lists database objects, including the 'staff' table. The 'Query Editor' pane contains the SQL query: 'select start_day_of_job from staff where position = '1'';. The 'Data Output' pane displays a list of 35 rows, each containing a staff ID and their start date. The results show various dates ranging from 11/01/20 to 02/08/20. The system tray at the bottom right indicates the date and time as 28-11-2020 11:34.

```
1 select start_day_of_job from staff where position = '1';
1   start_day_of_job
  character varying (10)
27  11/01/20
28  10/19/21
29  02/15/20
30  05/08/21
31  12/11/20
32  07/09/20
33  03/03/20
34  01/10/20
35  02/28/20
```

39. List all animals whose age is less than 10 and breed 'Pug' and sex is male

→ select * from animal_profile where age<10 and breed = 'Pug' and sex = '1';

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with the 'animal_profile' table selected. The 'Query Editor' tab contains the SQL query: `select * from animal_profile where age<10 and breed = 'Pug' and sex = '1';`. The 'Data Output' tab shows the results of the query:

	id	name	sex	age	breed	staff_id	adopter_id	donor_id	volunteer_id	cage_no
1	1001	Madeson	1	2	Pug	5601	3001	10001	572	SI
2	1061	Callista	1	7	Pug	5640	3061	10061	517	SI
3	1063	Fitzgerald	1	3	Pug	5634	3063	10063	517	SI
4	1069	Olive	1	4	Pug	5662	3066	10069	575	SI
5	1101	Olive	1	4	Pug	5662	3096	10009	575	SI

40. Starting day of all staff with salary > 10000

→ select start_day_of_job from staff where salary>10000;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with the 'staff' table selected. The 'Query Editor' tab contains the SQL query: `select start_day_of_job from staff where salary>10000;`. The 'Data Output' tab shows the results of the query:

	start_day_of_job
92	09/13/20
93	09/15/20
94	12/11/20
95	07/31/20
96	07/09/20
97	03/03/20
98	02/01/20
99	01/10/20
100	02/23/20