Animal Shelter Management System

Prepared by Section2 Group4

Sudarshan Kundnani – 201801140

Dhruv Chavda – 201801151

Chirag Patel – 201801225

Darshil Rana – 201801462

Mentored By:

Aashish Paliwal

Dhirubhai Ambani Institute of Information and Communication Technology

Table of Contents

Section 1: Software Requirements Specification
INTRODUCTION 5
1.1) Purpose 5
1.2) Intended Audience and Reading Suggestions: 5
1.3) Product Scope: 5
1.4) Description: 6
FACT FINDINGS 11
2.1) Background Reading: 11
2.2) Roleplay Interview: 12
2.3) Questionnaire: 16
2.4) Observations 20
Requirements 22
User Classes and Characteristics: 23
Operating Environment: 24
Product Functions 24
Assumptions 25
Privileges 25
Business Constraints 26
Section 2: Noun Analysis
Table 1: All nouns and verbs from Problem description 28
Table 2: Accepted noun and verb list 32
Table: 3 Reject Nouns 33
Section 3: Final ER Diagrams *(All Versions)*
Final ER Diagram 43
Section 4: Conversion of Final ER Diagram to Relational Model
Relational Tables and their Attributes 45
Section 5: Normalization and Schema Refinement
List of Redundancies 47
List of Anomalies 47
Normalization 47
1NF 47
2NF 47
3NF 47
Section 6: SQL
Final DDL Scripts 49
Animal 49
Shelter 49
Pet_Accessories 49

Staff 50
Donation 50
Adoption 50
Description 51
Breed 51
Animal_Type 51
Customer 51
Medication 52
General Insert statements 52
SQL Queries 54
Easy Queries: 54
Medium Queries: 65
Hard Queries 80
1)Trigger for customer Relation 80
2) Trigger for Adoption Relation 85
3) Trigger for Donation Relation 93
4) animal trigger 102

Section 1

Software Requirements Specification

INTRODUCTION

1.1) Purpose

Our SRS document's primary purpose is to develop an animal shelter management system for animal shelters, rescue groups, and animal control facilities and manage their business operations efficiently and effectively while reducing friction in the pet adoption process.

1.2) Intended Audience and Reading Suggestions:

This database system primarily focuses on those people who are willing to adopt pets. With this database system, they will be able to see description of available pets and so this will help in finding a right pet for themselves. This database will also help owners of such pet shops and adoption centres in maintaining and making changes in such a huge data seamlessly. Also, developers and users who wish to know about our app compatibility could read this document to get a clear idea of our database system (you may skip over to the compatibility part if you only wish to know about the platforms supporting our database).

1.3) Product Scope:

Various businesses can widely use our system. It creates a helpful and easy-to-use environment for people looking for adopting pets and animal shelter groups taking care of animals. Animals that are lost or abandoned are kept in animal shelters while finding potential adopters to input data about themselves and be matched to a pet that fits their lifestyle.

It also maintains a database regarding the animal's essential characteristics like breed, color, age so that the buyer will have more convenience in finding his pet.

1.4) Description:

The product we are building is self-contained, the first one of its type that will store and manipulate the data of different shelter animals. It will be able to keep track of animals that have been rescued and also that the adopted ones.
Our product works on two bases, like providing shelter for animals and looking for adopters who are willing to adopt them.

The shelter wants to ensure that the necessary information is available to provide proper care to their animals.

The Database Management System will include many functionalities for the manager (owner) and customers.

The pets are divided into two categories:

- Ordered pets (Animals from Breeding Mills like puppy mills)
- Donated pets

We will categorize the animals to ease the process of finding the right pet like colour, age, breed and many more.

The following are the functionalities that are divided according to the designation:

Customers:

- View Customer Details.
  - Each customer can see his/her details with previous activities, like which pet they have adopted when they have adopted the pet etc.
- View Pet Details.
  - Customers can see every single detail of a particular animal like what is the age? What is its breed? etc.
- Sort the Pet based on preferences (like breed, gender, etc).
- Compare different Pets.
  - Customers can manually compare different pets for better selection.

- Edit Customer (own) Details.
    - Customers can edit its personal details like his/her postal Address, contact number etc.
    - Customers can edit only their personal data from their side to maintain integrity of data.

Manager:

- Managers are the people who are responsible to sell pets to the customer.
- Managers will require a Login ID and password to access the database.
- Managers can view the details of their customers as well as keep track of the pet he/she has sold the most.
- Managers can also easily see the amount of profit they made to date.
- Developers are people who keep track of the various trends regarding pets, suggest new functionalities, or change existing functionalities for the company's best interest.

Developer:

- A Developer will require a Login ID and a password to access the database.
- A Developer will have access to edit/add/remove functionalities from the current database.

Common Functionalities:

- The feature to sort the pets will be available to every type of customer, agent as well as the Developers.
- The feature to compare one pet with others will also be available to every customer, manager as well as the Developers.

The general idea of the **tables** and their **attributes** is given below:

The customers' table will have the following attributes: -

- Name
- Address
- Contact
- Email
- Liked animals
    - These are the general details that are required to fill by the customers for the identification purposes (if needed).

The Pet's table will have the following attributes: -

- Type
    - This field will show what type of animal it is. An aquatic animal, a canary, or a dog/cat.
- Breed
    - This field will give information about the breed of the animal (if it's a dog, the whether it's a pug or dalmatian)
- height
- weight
- age
- Description
    - This section will include all the meta-information about the pet animal like its food preferences, medication, ambient atmosphere, likes-dislikes, allergies, etc in brief
- Donated/ new
    - This category will justify whether the pet animal is ordered or stray so that customers can choose according to their preference.

General Functionalities:

- The database will be able to map every customer to the pets that they have bought which will help the Developer to identify which policies have higher selling rates as well as which policies need to be changed.

- A customer can view every pet that they have bought along with the details of the pet.

- To provide accurate information to customers regarding the available pets, database will be changed every week. So that customers don't miss out any important information as well as their liked pets.

- Before adopting a pet, customers will be provided important information regarding their liked pets and their cost as well as responsibilities required being a pet owner.

- The above thing will be a very useful tool for animals as well as the customers as performing it will help the owners to understand how it looks like being a pet owner and hence will decrease the number of pets being abandoned or returned.

- Moreover, new customers will also be provided the contact information of other customers to share their experience being a pet owner and the kind of responsibilities required so that they can be well-prepared.

- Also, a feedback type form can be provided to the past customers to share their experience and about the pet being friendly or not. This will further help the customers to decide which pet they should adopt according to their like.

- Customers can also provide donations in the form of animals or pet accessories such as medicines, cage, belts etc

General Process for customers to adopt a pet:

  - Before adopting any pet, customers are required a have an account to access the database and look for their pets.

  - If the customer is new then he would be urged to fill their details that will be stored in the customer table and hence their account will be created in which they could access the database with the login credentials they provided during their account sign-up.

  - This information will further be stored in the customers table to help for future reference.

- Moreover, the customers are now provided the information regarding the pet. He can choose from the list of available pets (based on her liking and preferences).

- Finally, after choosing the right pet he/she is provided enough information about the respective pet i.e it's lifestyle, cage/house required, medication from the database provided in our schema.

General Process for customers to donate a pet/ pet accessories:

- Before donating any pet, customers are required a have an account to access the database and look for their pets.

- If the customer is new then he would be urged to fill their details that will be stored in the customer table and hence their account will be created in which they could access the database with the login credentials they provided during their account sign-up.

- This information will further be stored in the customers table to help for future reference.

- Moreover, the customers are now provided the information regarding the modes of donation. They can provide donations in the form of pet supplies like cage, pet food, cage etc. They can even donate some pet animals to the shelter for other customers to adopt it.

- The donated items will automatically get stored in the database.

FACT FINDINGS

2.1) Background Reading:

So, we went through the internet, finding various articles about animal shelters and pet shops (online modes) and got to know about its applications and the problems faced on which we can work on. We also got to know about the new ideas about how to increase and attract more customers. It summarizes the detailed information of the market segments based on different terms, such as strategy, scope, and manufacturing base. We also got to know that more and more people have started in adopting stray animals and take care of them, so we can attach a separate section for such animals. we also studied some databases which are somewhat similar to our agenda.

References:

- Future Scope of Global Animal Shelter Management Software Market
- Adopting stray animals
- SRS EXAMPLE
- SRS TEMPLATE
- Advancements in Pet Databases

Requirements:

- Currently, very unorganized data in adoption centres.
- There is a severe need for a database that can store and manage the data of all the animals like animal type, id, age, whether spayed or neutered, etc. in the shelter
- Organizers need to compete with others, and so they have to cope-up themselves with the technology
- The shelter wants to maximize their operational efficiency through increased information.
- Cage occupancy and dimensions also need to be managed.

2.2) Roleplay Interview:

- Interview 1:

System: Animal Shelter Management Database

Interviewee: Dhruv Chavda (Role Play) **Designation**: Owner of Happy Paws pet shop and adoption centre.

Interviewer: 1) Darshil Rana **Designation:** Student DAIICT
2) Sudarshan Kundnani **Designation**: Student DAIICT

Date: 29/09/2020 **Time:** 12:30 PM

Duration: 35 minutes **Place:** Google Meet

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding Animal Shelter management System

Agenda:
Views on Current Animal Shelter Database Management System
Problems in the current system
Initial ideas
Suggestions

Documents to be brought to the interview:
Any documents relating to the current database management system.
Documents regarding the current working of the organization

Interview Summary (Interview 1):

System: Animal Shelter Management Database

Interviewee: Dhruv Chavda (Role Play) **Designation**: Owner of Happy Paws pet shop and adoption centre.

Interviewer: 1) Darshil Rana **Designation:** Student DAIICT
2) Sudarshan Kundnani **Designation**: Student DAIICT

Date: 29/09/2020 **Time:** 12:30 PM

Duration: 35 minutes **Place:** Google Meet

Purpose of Interview:

During the whole interview, the main topic of discussion was to make our product completely helpful for the owner and release his burden for maintaining a considerable amount of data. Many suggestions were given from his side which was entirely on point, and we looked after it like data should be easy to manipulate, it should contain a wide range of information about animals, staff, and all.
The following are the main points discussed with the owner and manager of Happy Paws (Pet shop and adoption Center):

- The current data is very unordered. They require a system to store data systematically.
- Since the data is collected in the bookish form, it becomes difficult for them to maintain and change in times of need. The data in the system should be easy to manipulate.
- The system should be able to keep a wide range of information like general information of all the animals like breed, type, gender, age etc.
- There is a need for a separate section, especially for those who are being rescued or donated by the people who no longer need them.
- Customers Should be able to search and sort animals easily according to their preferences.

- Interview 2:

System: Animal Shelter Management Database

Interviewee: Chirag Patel (Role Play) **Designation**: Developer of database (for Happy Paws pet shop and adoption center.

Interviewer: 1) Sudarshan Kundnani **Designation:** Student DAIICT
2) Darshil Rana **Designation**: Student DAIICT

Date: 29/09/2020 **Time:** 12:30 PM

Duration: 35 minutes **Place:** Google Meet

Purpose of Interview:
Meeting, primarily to implement the requirements of the shop owner/ manager and any suggestions related to database.

Agenda:
About Current Animal Shelter Database Management System
flaws in the current data management system
How to create an improved data management system.
Ideas regarding Functionalities of the database
Suggestions

Documents to be brought to the interview:
Any documents relating to the current database management system.
Documents regarding the current working of the organization

Interview Summary (Interview 2):

System: Animal Shelter Management Database

Interviewee: Chirag Patel (Role Play) **Designation**: Developer of database (for Happy Paws pet shop and adoption center.

Interviewer: 1) Sudarshan Kundnani **Designation:** Student DAIICT
2) Darshil Rana **Designation**: Student DAIICT

Date: 29/09/2020 **Time:** 12:30 PM

Duration: 35 minutes **Place:** Google Meet

Purpose of Interview:

The main topic of discussion was how to develop a database system to satisfy the requirements of the owner of the shop. We discussed about the different functionalities that can be added in this database which will ease the whole process and how to include a wide range of information about animals, staff, and all.
The following are the main points discussed with the developer:

- The current data is very unordered. The new system will store data in more ordered fashion.
- Once the database is created, the owner/manager should be able to change/edit the data inside the database.
- Functionalities regarding the sorting of animals based on user preferences should be implemented.
- There is a need for a separate section, especially for those who are being rescued or donated by the people who no longer need them.

2.3) Questionnaire:

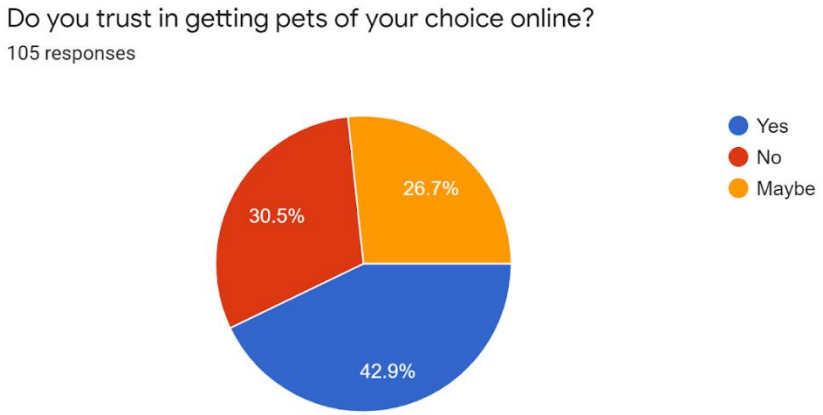
1. Do you trust in getting pets of your choice online?
2. Which type of animal would you prefer to have as a pet?
3. Which age range would you prefer for your pet to be in?
4. Normally where do you prefer to get pets from?
5. Are you willing to adopt handicapped animals?
6. Would you consult anybody if you want to adopt a pet?
7. (follow-up) If yes, who will you consult?
8. Would you like to have more than 1 pet?
9. (follow-up) If yes, would they be of the same type?
10. If you get a chance to create an Animal Shelter management system, what features would you like to see?

We created a google form including the following questions and given below are the responses of it

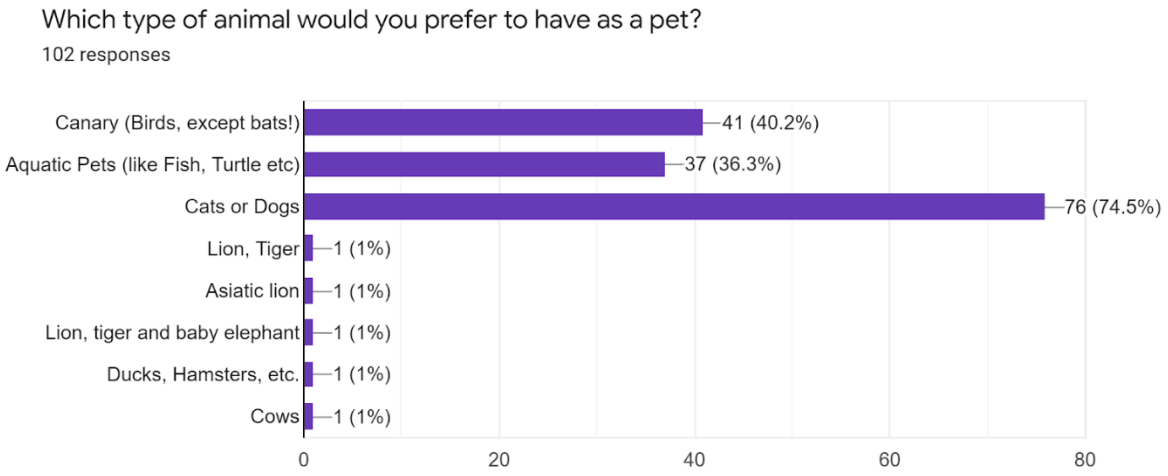
Link of the Form

Following were the responses:

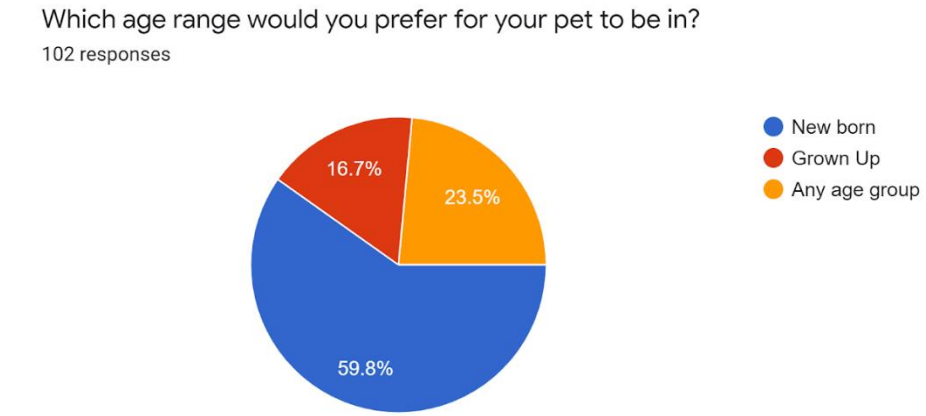
1.



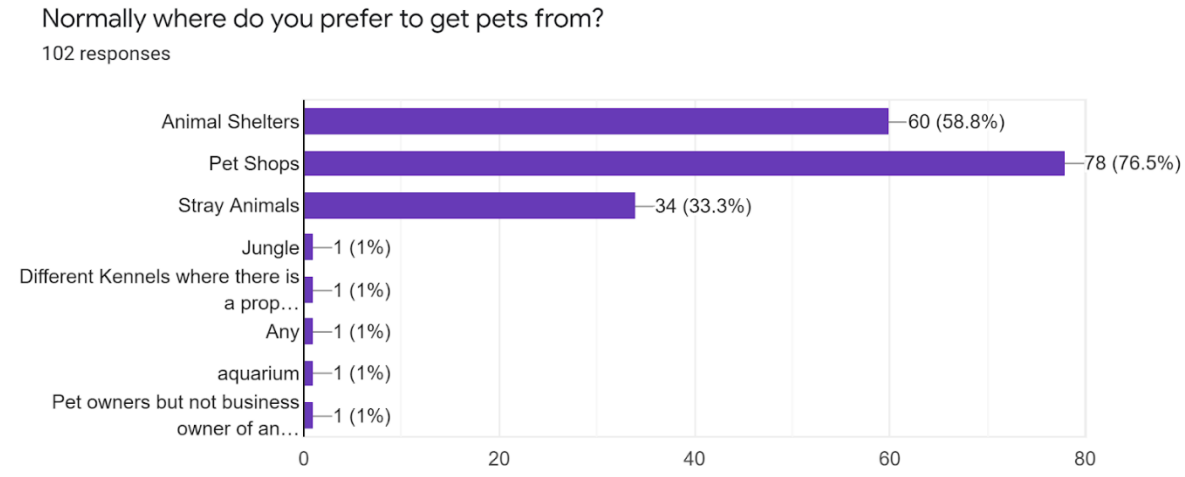
2.


Which type of animal would you prefer to have as a pet?
102 responses
Canary (Birds, except bats!)
41 (40.2%)
Aquatic Pets (like Fish, Turtle etc)
37 (36.3%)
Cats or Dogs
76 (74.5%)
Lion, Tiger
1 (1%)
Asiatic lion
1 (1%)
Lion, tiger and baby elephant
1 (1%)
Ducks, Hamsters, etc.
1 (1%)
Cows
1 (1%)
0
20
40
60
80

3.

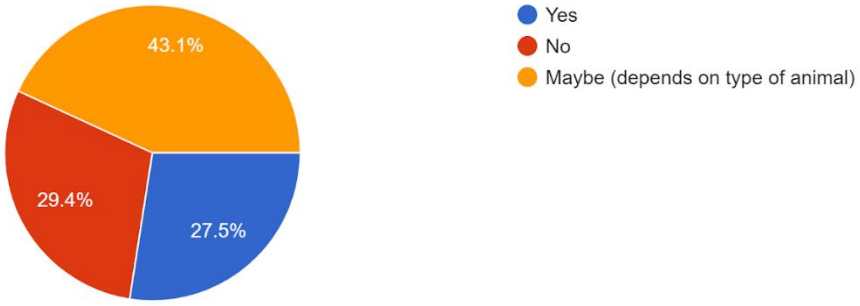

Which age range would you prefer for your pet to be in?
102 responses
16.7%
23.5%
59.8%
New born
Grown Up
Any age group

4.


Normally where do you prefer to get pets from?
102 responses
Animal Shelters
60 (58.8%)
Pet Shops
78 (76.5%)
Stray Animals
34 (33.3%)
Jungle
1 (1%)
Different Kennels where there is a prop…
1 (1%)
Any
1 (1%)
aquarium
1 (1%)
Pet owners but not business owner of an…
1 (1%)
0
20
40
60
80

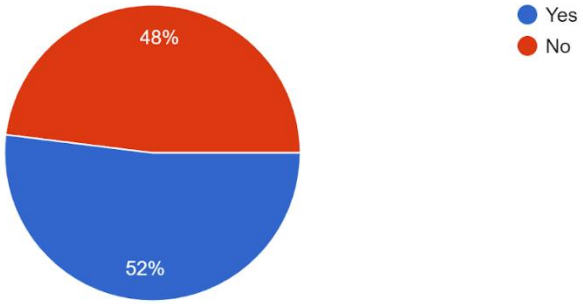
5.

Are you willing to adopt handicapped animals?
102 responses



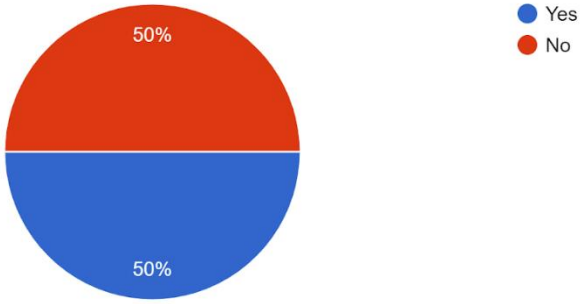
6.

Would you consult anybody if you want to adopt a pet?
102 responses



8.

Would you like to have more than 1 pet?
102 responses



Survey for Animal Shelter Management System

The following are some questions regarding the database for an animal shelter. Please read them carefully and answer them appropriately.

*Required

Do you trust in getting pets of your choice online? *

- Yes
- No
- Maybe

Which type of animal would you prefer to have as a pet? *

- [ ] Canary (Birds, except bats!)
- [ ] Aquatic Pets (like Fish, Turtle etc)
- [ ] Cats or Dogs
- [ ] Other: _____

Which age range would you prefer for your pet to be in? *

- New born
- Grown Up
- Any age group

Normally where do you prefer to get pets from? *

- [ ] Animal Shelters
- [ ] Pet Shops
- [ ] Stray Animals
- [ ] Other: _____

Are you willing to adopt handicapped animals? *

- Yes
- No
- Maybe (depends on type of animal)

Would you consult anybody if you want to adopt a pet? *

- Yes
- No

If yes, who will you consult?

Your answer

Would you like to have more than 1 pet? *

- Yes
- No

If yes, would they be of the same type?

Your answer

If you get a chance to create an Animal Shelter Management system, what features would you like to see?

Your answer

If you get a chance to create an Animal Shelter Management system, what features would you like to see?

Your answer

Thank you for your time, woof! :)

Submit

Never submit passwords through Google Forms.

2.4) Observations

- The majority of the public did not have any problems in online mode.
- Most of the people like Cats and dogs followed by canary and aquatic pets. We concluded that thinking of aquatic animals as pets would be financially high as it requires more care and proper equipment for need.
- Around 60% of the public would like to have a new born pet which was as per expectation as new-borns are the cutest ones.
- Almost ¾ th of the responses were of the view to buying a pet from a pet shop. We found it very surprising that nearly ⅓ rd of the public were not reluctant to have a stray animal as a pet.
- Marjory finds no issues in having a handicapped pet.
- Almost equal people were in favour of consulting someone before buying a pet. They would probably like to consult someone who has the same type of pet they wanted to have.
- Half of the people wanted to have more than one pet and probably of different types.
- Regarding the different features that a user would like to have in their system, we found that people would like to have a proper record of the animals, lots of pictures with descriptions like age, weight, gender, breed etc.

Fact Finding Chart:

Objectives	Techniques	Subjects	Time Commitment
To get an overview of current trends in the Animal Shelter Database Management System	Background Reading	News Articles and Websites	1 Day
To get an idea what features we can include to overcome current liabilities	Interview	Owner of pet shop (Roleplay)	35 Minutes
	Interview	Manager of pet shop (Roleplay)	35 Minutes
To Establish what records and resources are kept	Interview/ Document Sampling	Resource Manager	2 Hours
Data Analysis of customers	Questionnaire (Using Google Forms)	Public Survey	2 Days
Functionalities that customers want in the interface of the system	Questionnaire (Using Google Forms)	Public Survey	2 Days

Requirements

Based on Interviews

- An easy to use Database Management System (Frequency = 2)
- Features to compare and sort based on different required characteristics (Frequency = 2)
- Edit Pet and Customer Details (Frequency = 2)
- Customers can view their History (Frequency = 1)

Based on Google Forms:

- Customers can view the available animals and their characteristics
- There should be an online mode of transaction.
- Pics and a short description of available animals are highly requested.

Almost 75% of the people would be able to take a better decision of having an animal as a pet if they have their pictures.

Schema Overview:

- *covering only main relations.*
  *For more reference visit "**Final DDL Scripts**"*

- Animal
  - id (primary)
  - shelter id (foreign key)
  - type_id (foreign key)
  - breed_id (foreign key)
  - height,
  - weight,
  - skin color,
  - age,
  - rate,
  - cage size
  - description (food preferences and mediations)
- Animal_type
  - name
  - id (primary)
- Breed
  - name
  - id (primary)

- User
  - Name
  - ID (primary)
  - (address
  - contact
  - email)

- Adoption
  - id
  - user (foreign key)
  - animal (foreign key)
  - date of adoption
  - Payment mode

- Donation
  - id
  - user (foreign key)
  - animal (foreign key)
  - date of donation
  - donation info

User Classes and Characteristics:

Customers

- To view various animals that are currently available.
- Customers can watch their past actions/activities.
- View-only mode

Owner / Manager (Admin)

- To view details of his/her customer.
- Animal records
- To change/edit characteristics of animals.
- To delete/edit data of animals that unavailable (or sold).
- Can manipulate the database

Developer

- To edit/change functionalities of the database.
- To add/remove policies and norms.
- To edit/view details of new animals.
- Maintains the database

Operating Environment:

Software Requirements

- Operating System: Windows
- Database System: PostgreSQL

Hardware Requirements

- Computer with working internet connectivity

External Interface Requirement

- Databases of animals present in the shelter

Product Functions

- Add/Remove/Edit data of animals and customers from the database.
- View Customer Details.
- View Pet Details.
- Sort the Pet based on preferences (like breed, gender etc).
- Compare different Pets.
- Edit Customer Details.
- Edit Pet Details.
- Adopt a pet
- Donate a pet
- Buy Pet Accessories

Assumptions

- Customers/ Staff have a basic knowledge of using a computer.
- Database will be operated only by a staff member of the shelter
- Customer Details and Animal details are not Fake.
- We have access to Animal Database of the shop.
- Only admin will be able to change the Functionalities of the database

Privileges

Functions	Customer	Admin	Developer
View Pet Details	Yes	Yes	Yes
View Customer Details	Yes	Yes	Yes
Sort the Pet based on preferences (like breed, gender etc).	Yes	Yes	Yes
Compare different Pets	Yes	Yes	Yes
Edit Pet Details	No	Yes	Yes
Edit Customer Details	No	Yes	Yes
To edit/change functionalities of the database.	No	No	Yes
To add/remove policies and norms	No	No	Yes

Business Constraints

- There will be a time constraint since we have to develop the Database System in a short span.
- Since we have to start the project from the complete base, we have to carefully think of features that are to be included that will facilitate the system management but will eventually lead to higher cost.
- The old data needs to be migrated to the new Database
- The maintenance of the new Database will also cost based on the required additional features or optimization.
- A Developer will require if you want to change functionalities of the database.

Section 2

Noun Analysis

Table 1: All nouns and verbs from Problem description

	Nouns	
Product	preferences	manager
Type	food	people
Shelter	medication	pets
animals	atmosphere	customer
adopters	allergies	manager
information	rates	Login ID
care	developer	password
functionalities	Week	Database
manager	information	manager
customer	responsibilities	details
pets	owner	customers
categories	cost	track
Stray animals	tool	pet
colour	thing	manager
age	experience	amount
breed	Feedback form	profit
Breeding mills	rating	developer
data	General process	people
track	account	trends
product	Login credentials	pets
basis	questionnaire	functionality

shelter	liking	companies
animals	experience	interest
Customer ID	Feedback form	
animals	response	Login id
Database	lifestyle	password
management	feedback	access
system	medication	policies
pets	cage	customers
animals	schema	manager
process	mode	developer
designation	edit	features
customers	address	others
	Contact number	Date

view	idea
pet	table
customer	attributes
activities	name
details	address
customers	contact
age	email
breed	transactions
preferences	identification

gender	purpose
selection	field
edit	dog/cat
address	information
Contact number	pug
email	dalmatian
height	weight
description	age
section	meta-information

	Verbs	
building	store	manipulate
rescued	adopted	providing
looking	include	divided
categorized	finding	following
compare	edit	see
ordered	maintain	sell
require	access	view
keep	sold	made
regarding	suggest	change
sort	add\remove	existing
given	following	liked

show	give	include
choose	ordered	justify
according	map	bought
help	identify	need
provide	regarding	changed
miss	adopting	liked
performing	required	provided
understand	decrease	abandoned
returned	shared	required
decide	adopt	contain
stars	updated	access
stored	fill	urged
created	match	adopt
is	are	will

Table 2: Accepted noun and verb list

Candidate entity set	Candidate attribute set	Candidate relationship set
Customer	Customer Name	Identification
	Customer ID	
	Age	
	Address	
	Contact	
	Email	
Animal	Animal Name	Adopts
	Type	Donates
	Breed	
	Height	
	Weight	
	Age	
	Description	
	Gender	
	Donated/ Ordered	
	Rate	
Staff	Login ID	Designates
	Password	
	Name	
	Age	
	Address	
	Contact	
	Email	
	Designation	
Breed	ID	Classification
	Breed Name	
Adoption	Customer ID	Adopt
	Customer Name	

	Date of Adoption	
	Animal Adopted	
	Date	
Type	Type ID	Classification
	Type Name	
Shelter	ID	Shop
	Name	
	Email	
	Address	
Pet Accessories	ID	Looking for
	Item	
	Quantity	
	price	
Donation	Customer ID	Donate
	Customer Name	
	Date of Donation	
	Item Donated	
	Donation info	
	Date	

Table: 3 Reject Nouns

NOUN	REJECT REASON
Product	general
information	general
care	irrelevant
functionalities	attributes
categories	attributes

Stray animals	attributes
colour	attributes
age	attributes
Breeding mills	general
data	vague
track	irrelevant
product	duplicate
basis	irrelevant
shelter	duplicate
animals	duplicate
preferences	attributes
food	general
medication	associations
atmosphere	irrelevant
Week	irrelevant
information	duplicate
responsibilities	general
cost	duplicate
tool	vague
thing	general
experience	vague
rating	duplicate
General process	general
account	general
Login credentials	associations

questionnaire	general
liking	irrelevant

NOUN	REJECT REASON
Product	general
Type	attributes
Shelter	general
information	general
care	irrelevant
functionalities	attributes
categories	attributes
Stray animals	attributes
colour	attributes
age	attributes
breed	attributes
Breeding mills	general
data	vague
track	irrelevant
product	duplicate
basis	irrelevant
shelter	duplicate

people	general
pets	general
Login ID	associations
password	associations
Database	general
details	general
track	vague
amount	attributes
profit	attributes
people	duplicate
trends	general
pets	duplicate
functionality	duplicate
companies	vague
interest	irrelevant
Database	vague
management	vague
system	vague
pets	duplicate
process	general
designation	irrelevant
response	irrelevant
lifestyle	general
medication	attributes
cage	duplicate

schema	general
mode	vague
access	irrelevant
policies	vague
features	attributes
others	general
view	general
activities	vague
details	general
gender	attributes
selection	vague
edit	irrelevant
address	attributes
Contact number	attributes
email	attributes
height	attributes
description	general
section	general
idea	vague
table	vague
attributes	general
name	attributes
address	attributes
contact	attributes
email	attributes

transactions	irrelevant
identification	irrelevant
purpose	vague
field	vague
dog/cat	general
information	duplicate
pug	general
dalmatian	general
weight	attributes
age	attributes
meta-information	associations

Verbs	Reject reason
building	vague
rescued	irrelevant
looking	vague
categorized	general
compare	genenal
ordered	vague
require	vague
keep	general
regarding	vague
sort	general

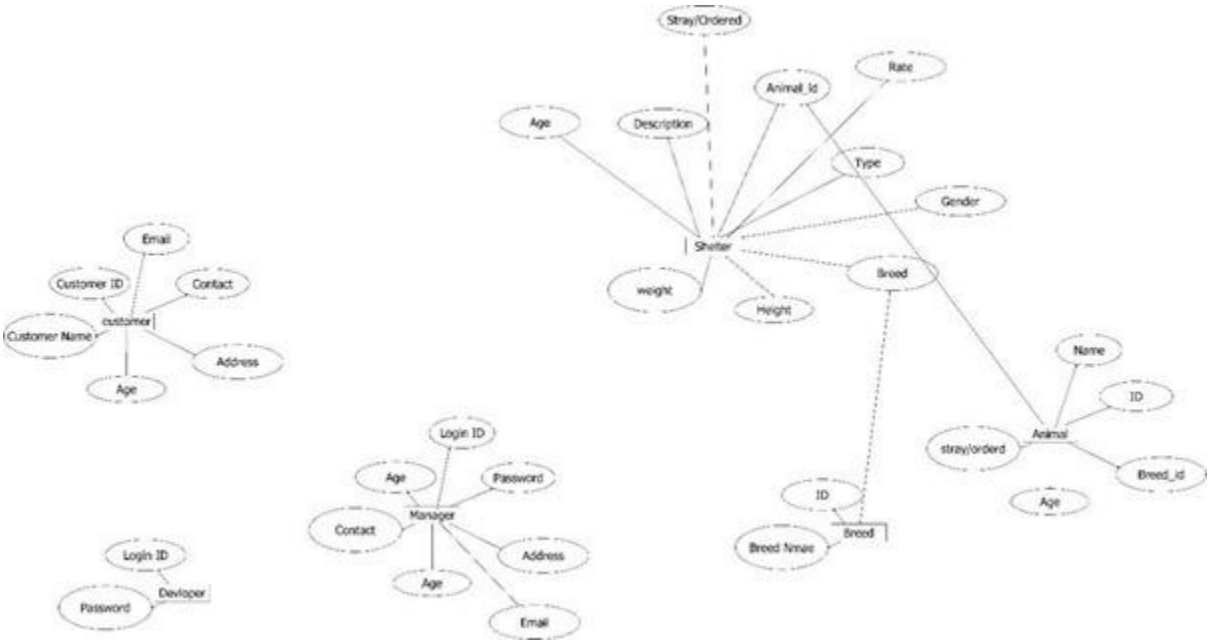
given	vague
show	vague
choose	vague
according	vague
help	vague
provide	vague
miss	general

Section 3

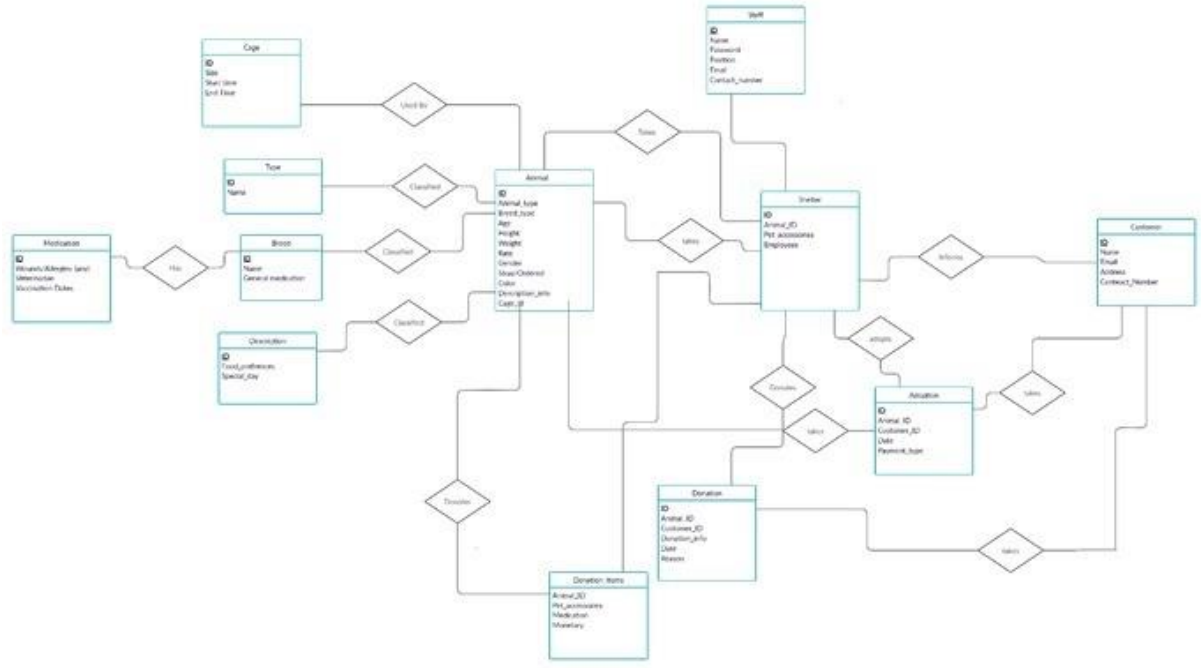
Final ER Diagrams

(All versions)

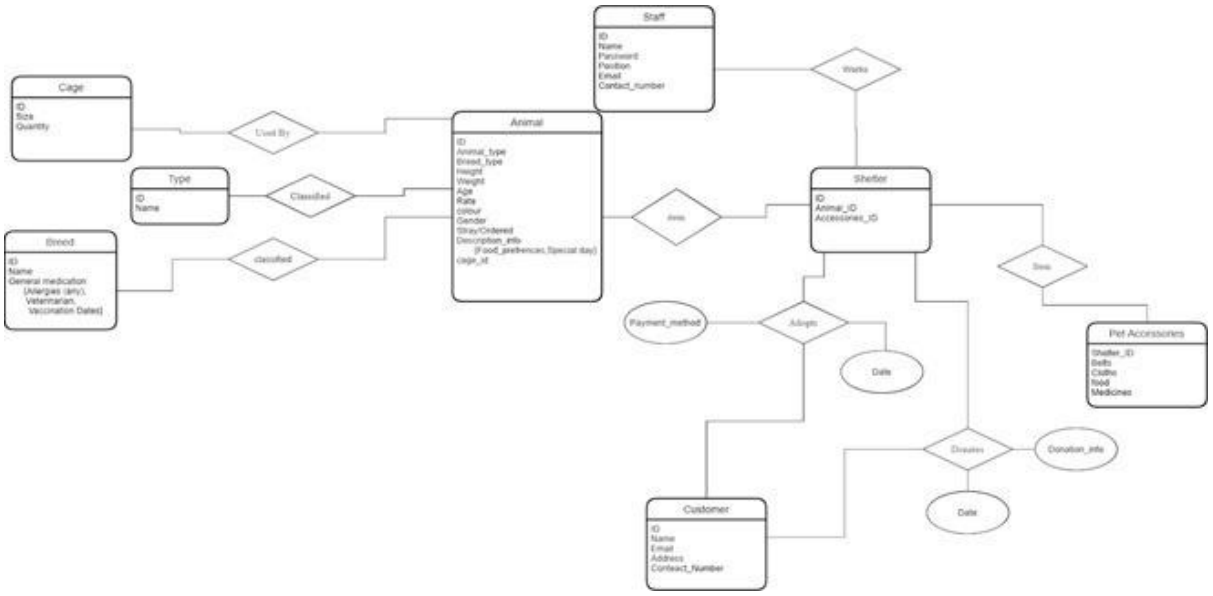
Version 1:



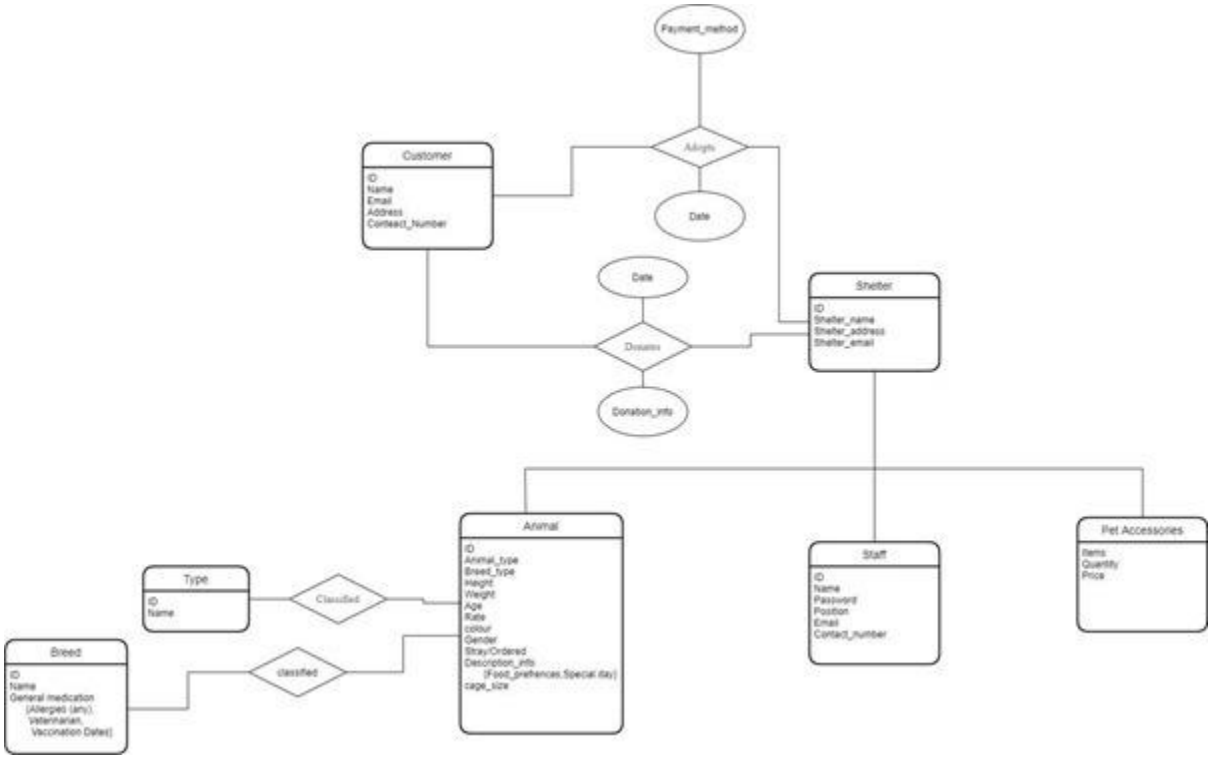
Version 2:



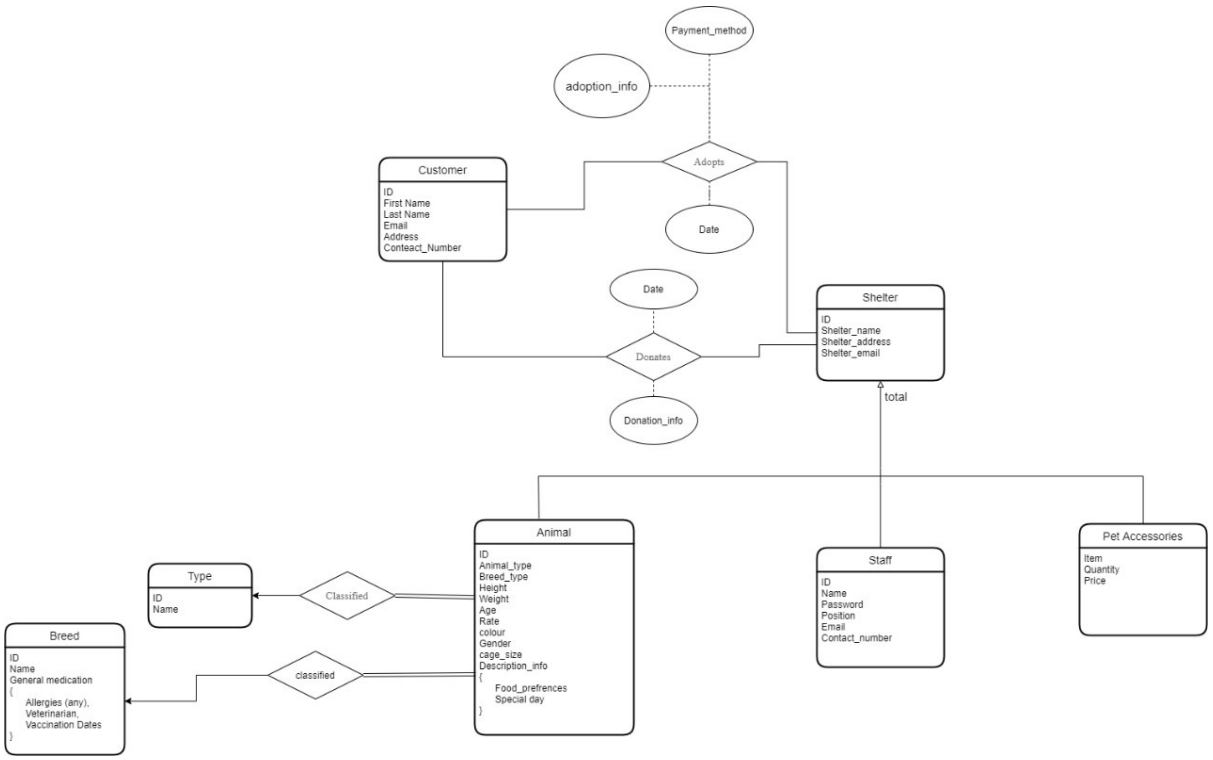
Version 3:



Version 4:



Final ER Diagram



Section 4

Conversion of Final ER-Diagram to Relational Model

Relational Tables and their Attributes

- Shelter(ID, name, address, email)
- Animal (ID,shelter_id, animal_type, breed_type, age, rate, gender, color , cage_size, height, weight)
- Type (ID, name)
- Breed (ID, name)
- Medication (Breed_ID, allergies, veterinarian, vaccination_date)
- Description (Animal_ID, food_preferences, special_day)
- Customer (ID, first_name, last_name, email, address, contact_number)
- Adoption (Animal_ID, Customer_ID, date, payment_type, adoption_info,shelter_id)
- Donation (Animal_ID, Customer_ID, donation_info, date,shelter_id)
- Staff (ID, shelter_ID, name, position, password, email, Contact_number)
- Pet_Accessories (Shelter_ID, item, quantity, price)

Section 5

Normalization and Schema Refinement

List of Redundancies

No redundancies found.

List of Anomalies

No Anomalies found.

Normalization

1NF

Description and medication attributes were multi-valued attributes for relations “animal” and “breed” respectively. These multi-valued attributes violate first normal form. So, the good idea is to create a separate table and push multivalued attribute into it.

2NF

No changes were required since there are no partial dependencies. The tables with composite PKs have no other non-primary attribute dependent on PK.

3NF

There are no Transitive Dependencies and hence the database is in 3NF.

Section 6
SQL

Final DDL Scripts

Animal

```
CREATE TABLE animal
(
        ID integer primary key,
        shelter_id integer references shelter(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        type integer references animal_type(id) ON DELETE CASCADE ON UPDATE
CASCADE,
        breed_type integer references breed(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        age integer,
        rate integer,
        gender varchar(10),
        colour varchar(15),
        cage_size varchar(2),
        height float,
        weight float
);
```

Shelter

```
CREATE TABLE shelter
(
        id int primary key,
        name varchar(255) not null,
        address varchar(255),
        email varchar(150)
);
```

Pet_Accessories

```
CREATE TABLE Pet_Accessories
(
        shelter_id integer references shelter(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        item varchar(100),
        quantity bigint,
        price bigint
```

```
);
```

Staff

```
CREATE TABLE staff
(
        id integer primary key,
        shelter_id integer references shelter(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        name varchar(100) not null,
        position varchar(50),
        password varchar(15),
        email varchar(50),
        contact_number bigint
);
```

Donation

```
CREATE TABLE donation
(
        animal_id integer references animal(id) ON DELETE CASCADE ON UPDATE
CASCADE,
        customer_id integer references customer(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        donation_info varchar(150),
        date varchar(100)
        shelter_id integer references shelter(id) ON DELETE CASCADE ON
UPDATE CASCADE,
);
```

Adoption

```
Create Table adoption
(
        animal_id integer references animal(id) ON DELETE CASCADE ON UPDATE
CASCADE,
        customer_id integer references customer(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        date varchar(50),
        payment_type varchar(10),
        adoption_info varchar(255),
```

```
	Shelter_id integer references animal(id) ON DELETE CASCADE ON
UPDATE CASCADE
);
```

Description

```
CREATE TABLE description
(

	animal_id integer references animal(id) ON DELETE CASCADE ON UPDATE
CASCADE,
	food_preferences varchar(25),
	special_day varchar(100)
);
```

Breed

```
CREATE TABLE breed
(
	id integer primary key,
	name varchar(100) not null
);
```

Animal_Type

```
CREATE TABLE animal_type
(
	id integer primary key,
	name varchar(100) not null
);
```

Customer

```
Create Table customer
(
	ID integer Primary Key,
	first_name varchar(255) NOT NULL,
	last_name varchar(255) NOT NULL,
	email varchar(100),
	Address varchar(255),
	Contact_number bigint
);
```

Medication

```
Create Table medication
(
        breed_id integer references breed(id) ON DELETE CASCADE ON UPDATE
CASCADE,
        allergies varchar(100),
        veterinarian varchar(100),
        vaccination_date varchar(50)
);
```

General Insert statements

animal

```
INSERT INTO "animal shelter".animal(
        id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

shelter

```
INSERT INTO "animal shelter".shelter(
        id, name, address, email)
        VALUES (?, ?, ?, ?);
```

pet_accessories

```
INSERT INTO "animal shelter".pet_accessories(
        shelter_id, item, quantity, price)
        VALUES (?, ?, ?, ?);
```

staff

```
INSERT INTO "animal shelter".staff(
        id, shelter_id, name, "position", password, email, contact_number)
        VALUES (?, ?, ?, ?, ?, ?, ?);
```

donation

```
INSERT INTO "animal shelter".donation(
        animal_id, customer_id, donation_info, date, shelter_id)
        VALUES (?, ?, ?, ?, ?);
```

adoption

```
INSERT INTO "animal shelter".adoption(
        animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
        VALUES (?, ?, ?, ?, ?, ?);
```

description

```
INSERT INTO "animal shelter".description(
        animal_id, food_preferences, special_day)
        VALUES (?, ?, ?);
```

animal_type

```
INSERT INTO "animal shelter".animal_type(
        id, name)
        VALUES (?, ?);
```

breed

```
INSERT INTO "animal shelter".breed(
        id, name)
        VALUES (?, ?);
```

customer

```
INSERT INTO "animal shelter".customer(
        id, first_name, last_name, email, address, contact_number)
        VALUES (?, ?, ?, ?, ?, ?);
```

medication

```
INSERT INTO "animal shelter".medication(
        breed_id, allergies, veterinarian, vaccination_date)
        VALUES (?, ?, ?, ?);
```

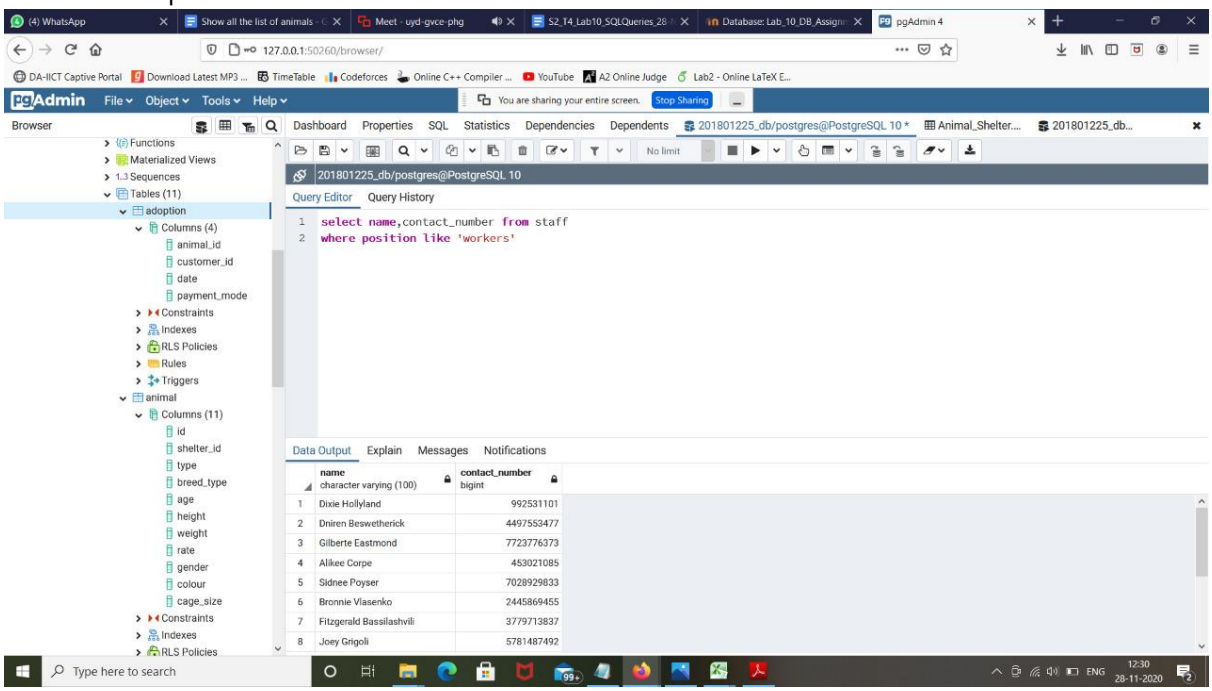
SQL Queries

Easy Queries:

1) List the contact numbers of workers of all shelters

SQL QUERY:

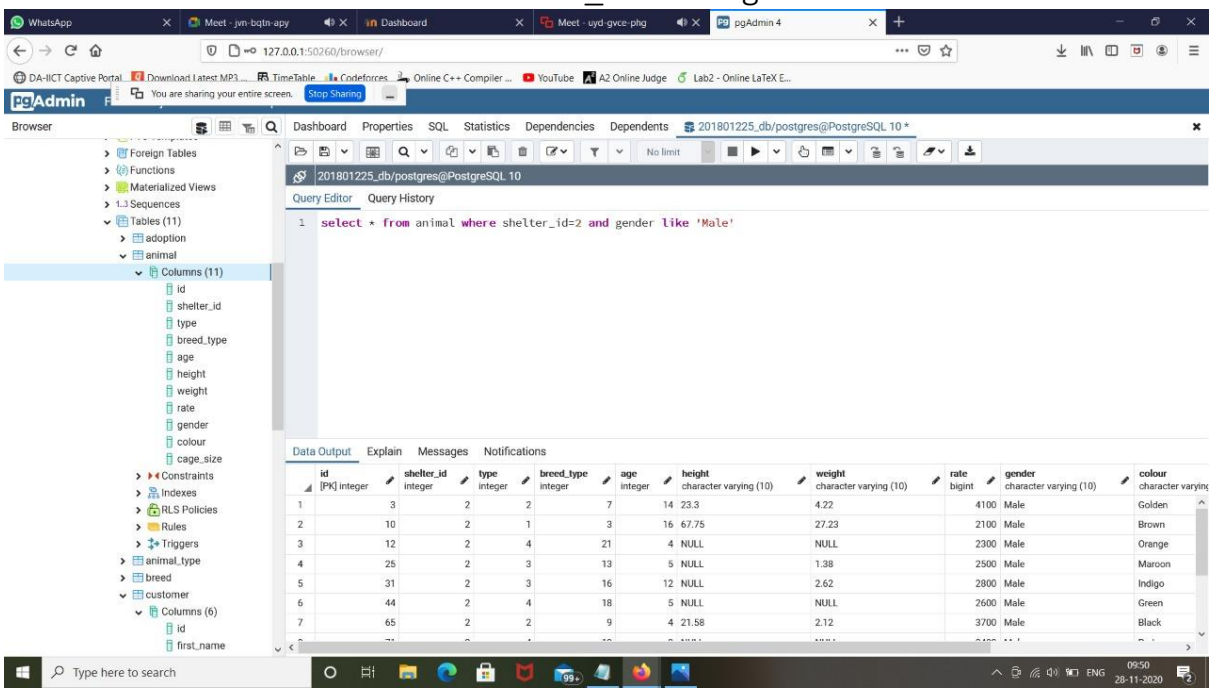
```
select name,contact_number from staff
where position like 'workers'
```



2)List down all the male animal from the shelter 2:

SQL QUERY:

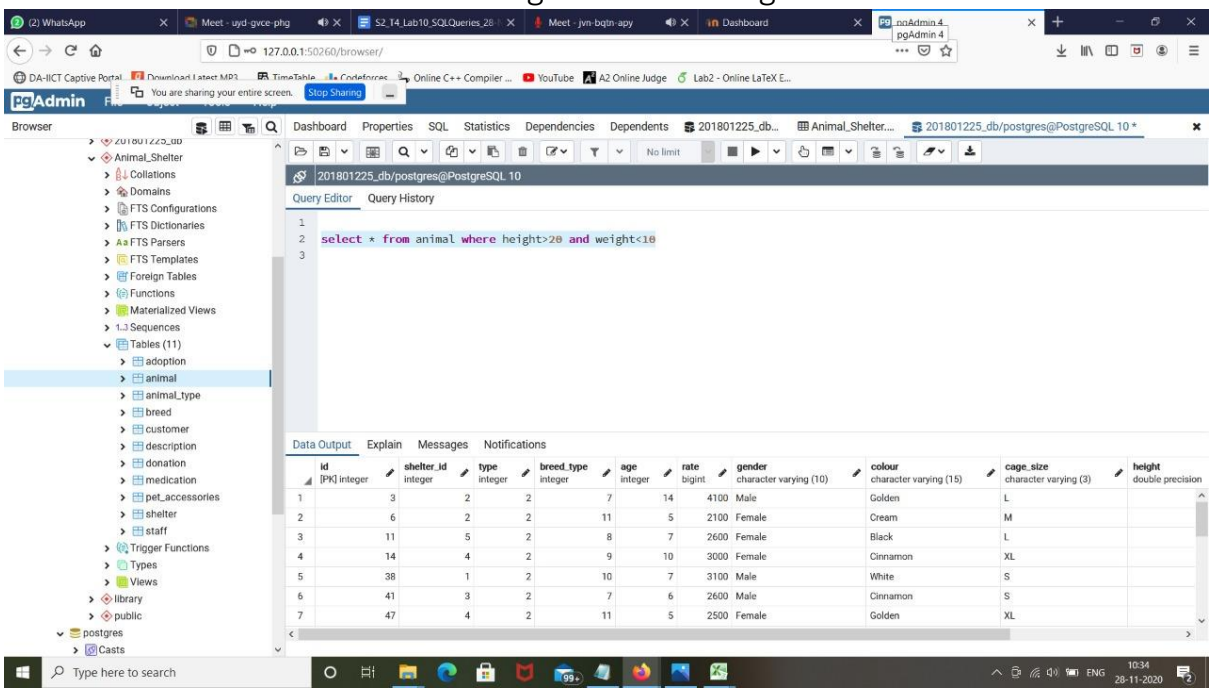
```
select * from animal where shelter_id=2 and gender like 'Male'
```



3) List all animals with height greater than 20cm and weight less than 10kg

SQL QUERY:

```
select * from animal where height>20 and weight<10
```

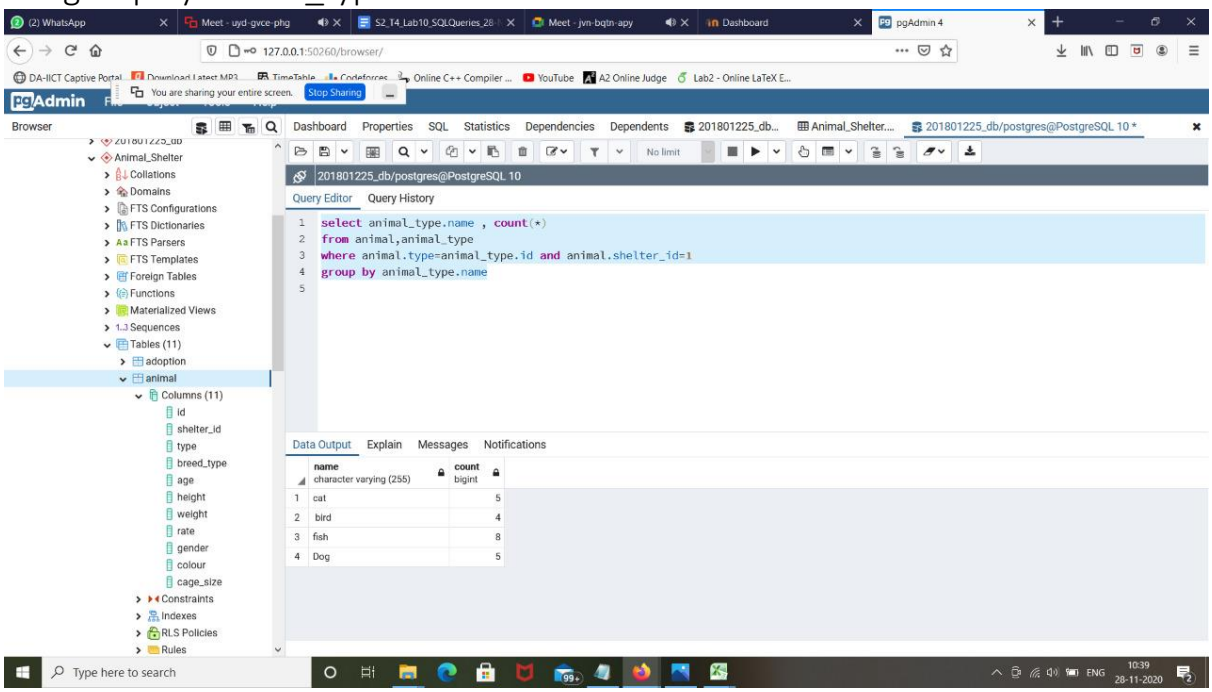


	id [PK] integer	shelter_id integer	type integer	breed_type integer	age integer	rate bigint	gender character varying (10)	colour character varying (15)	cage_size character varying (3)	height double precision
1	3	2	2	7	14	4100	Male	Golden	L	
2	6	2	2	11	5	2100	Female	Cream	M	
3	11	5	2	8	7	2600	Female	Black	L	
4	14	4	2	9	10	3000	Female	Cinnamon	XL	
5	38	1	2	10	7	3100	Male	White	S	
6	41	3	2	7	6	2600	Male	Cinnamon	S	
7	47	4	2	11	5	2500	Female	Golden	XL	

4) Count the number of each type of animals from shelter 1

SQL QUERY:

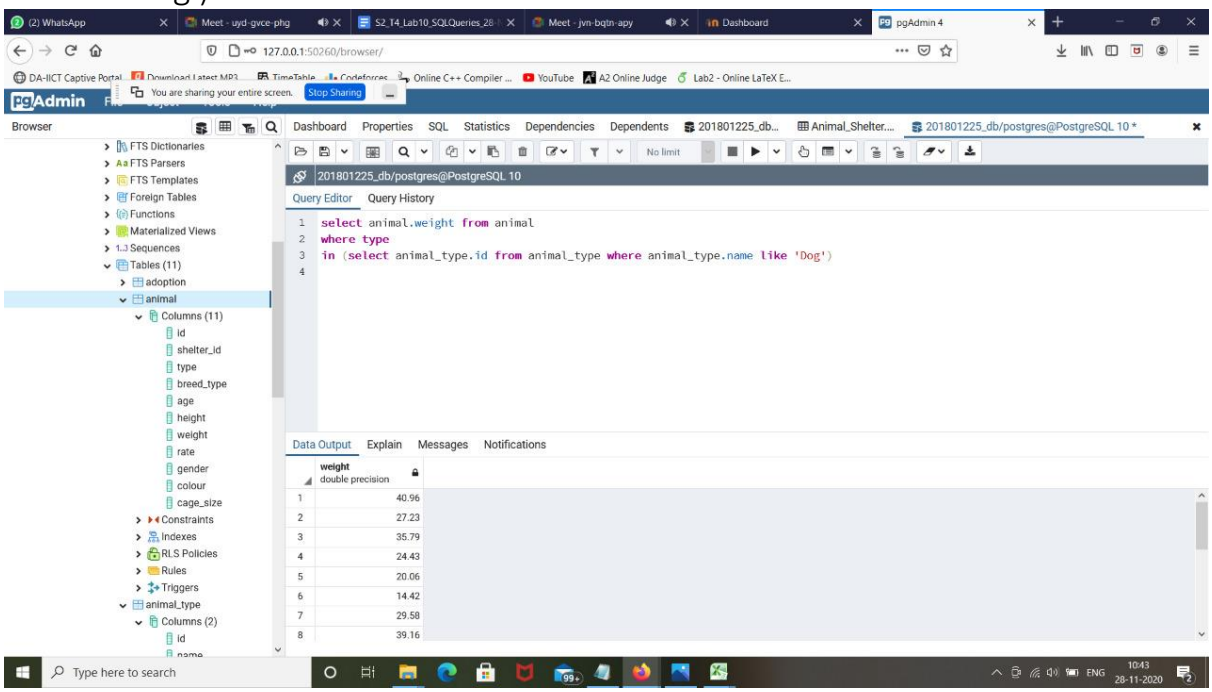
```
select animal_type.name , count(*)
from animal,animal_type
where animal.type=animal_type.id and animal.shelter_id=1
group by animal_type.name
```



5) Display the weights of all dogs

SQL QUERY:

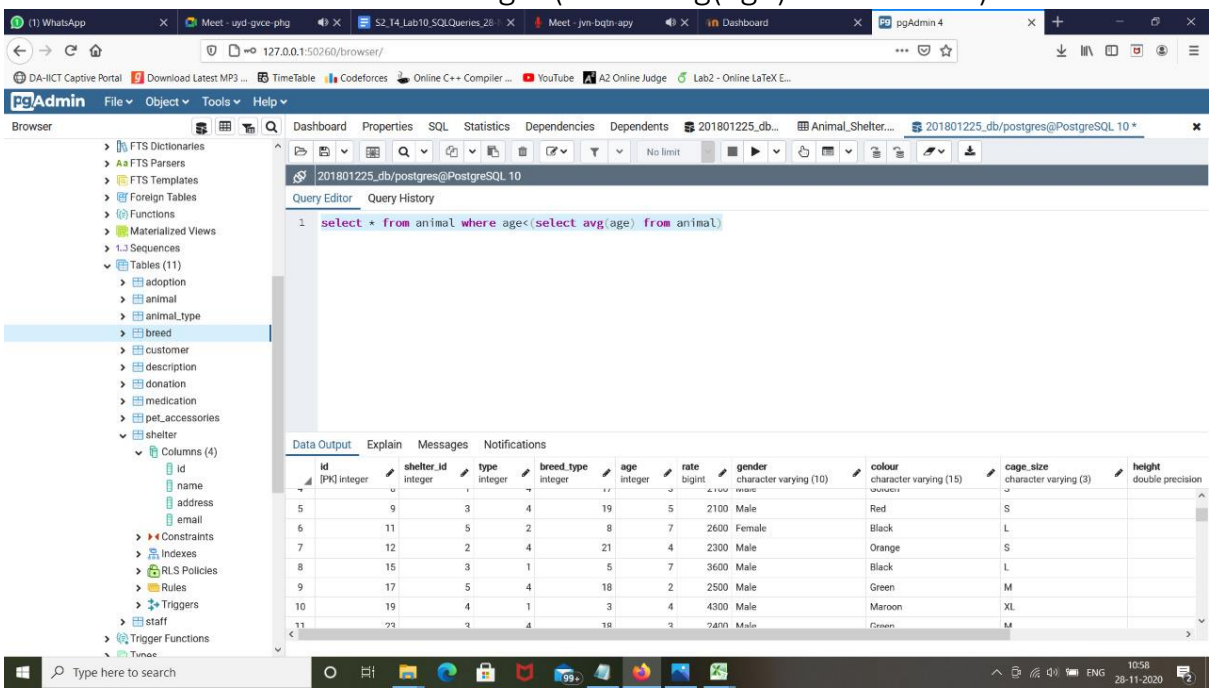
```
select animal.weight from animal
where type in
        (select animal_type.id from animal_type where animal_type.name
like 'Dog')
```



6) List all the animal details whose age is less than the average age

SQL QUERY:

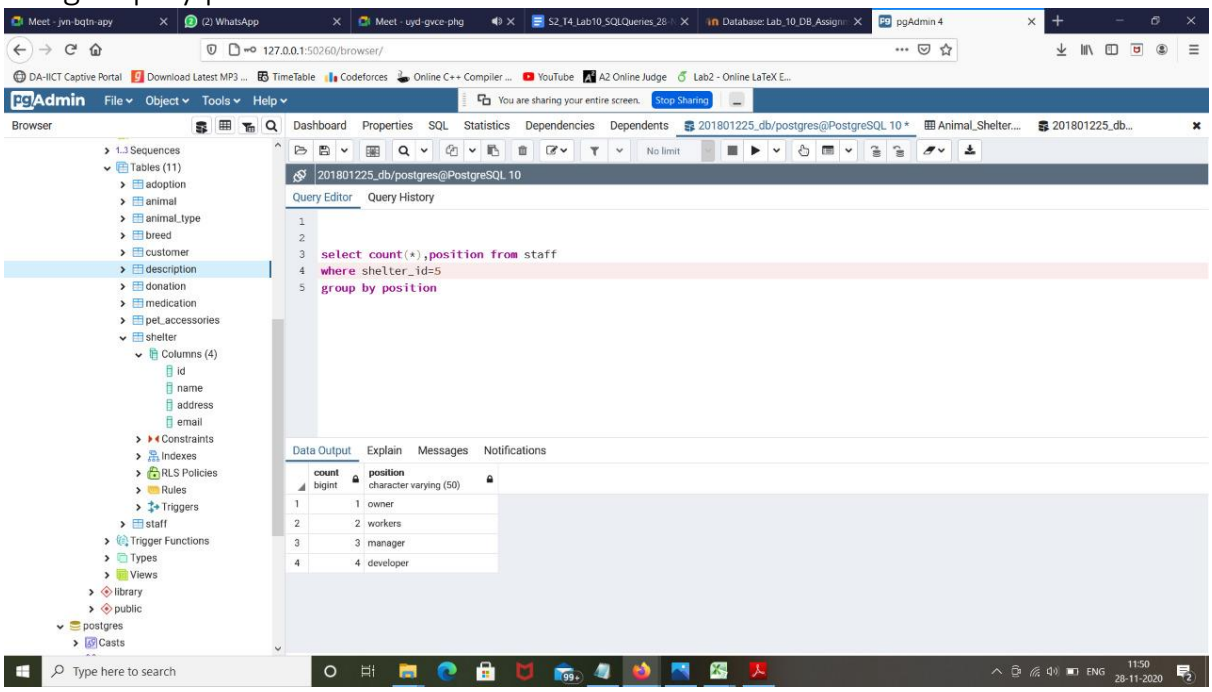
```
select * from animal where age<(select avg(age) from animal)
```



7) Count the number of workers in different position of staff of shelter 5

SQL QUERY:

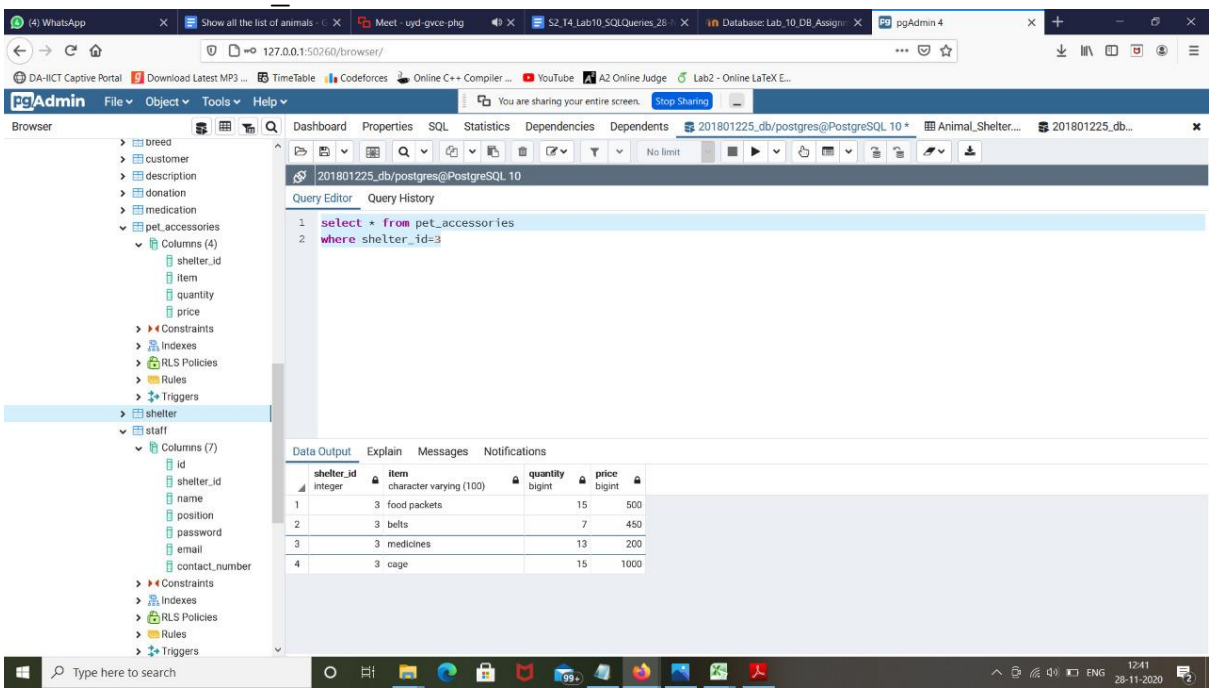
```
select count(*),position from staff
where shelter_id=5
group by position
```



8) List all the details of pet accessories in shelter 3

SQL QUERY:

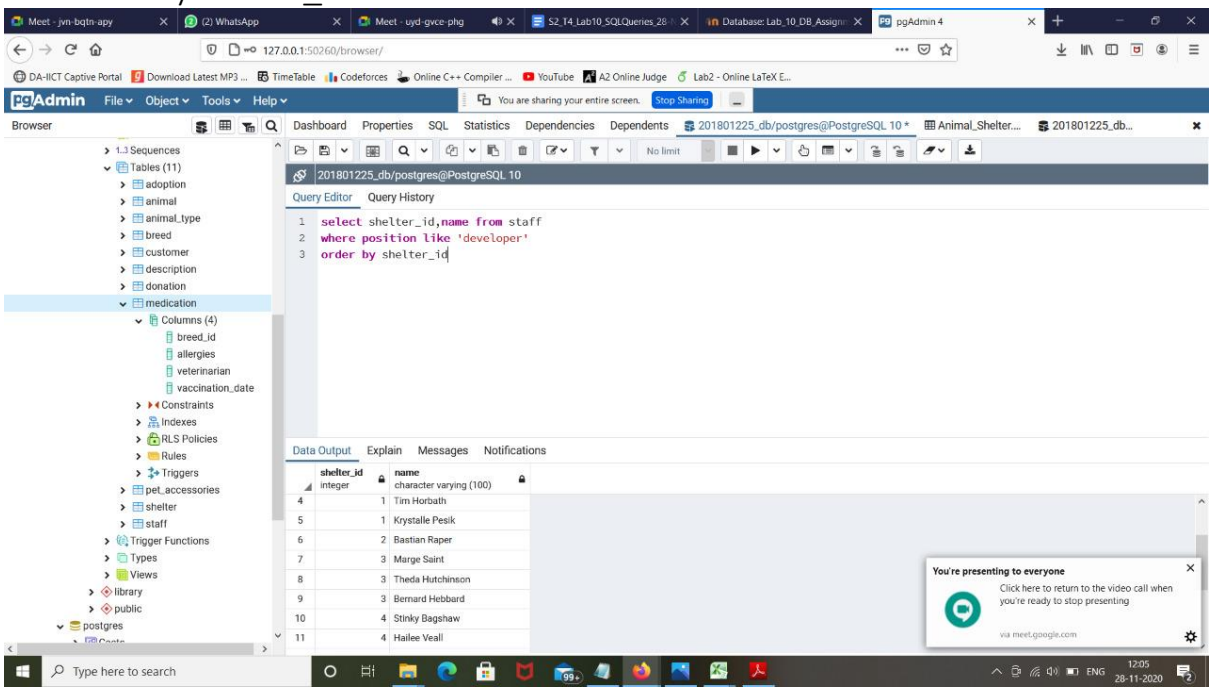
```
select * from pet_accessories
where shelter_id=3
```



9) List all names of developers for all shelters

SQL QUERY:

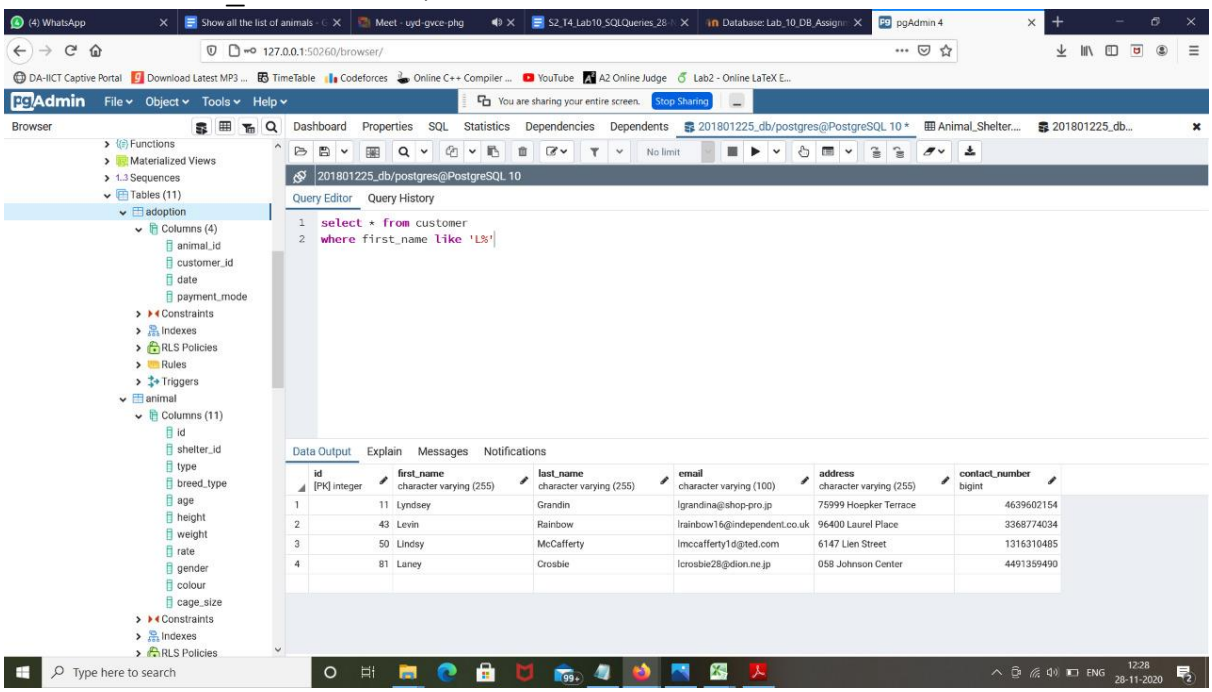
```
select shelter_id,name from staff
where position like 'developer'
order by shelter_id
```



10) List all the customers whose name start with L

SQL QUERY:

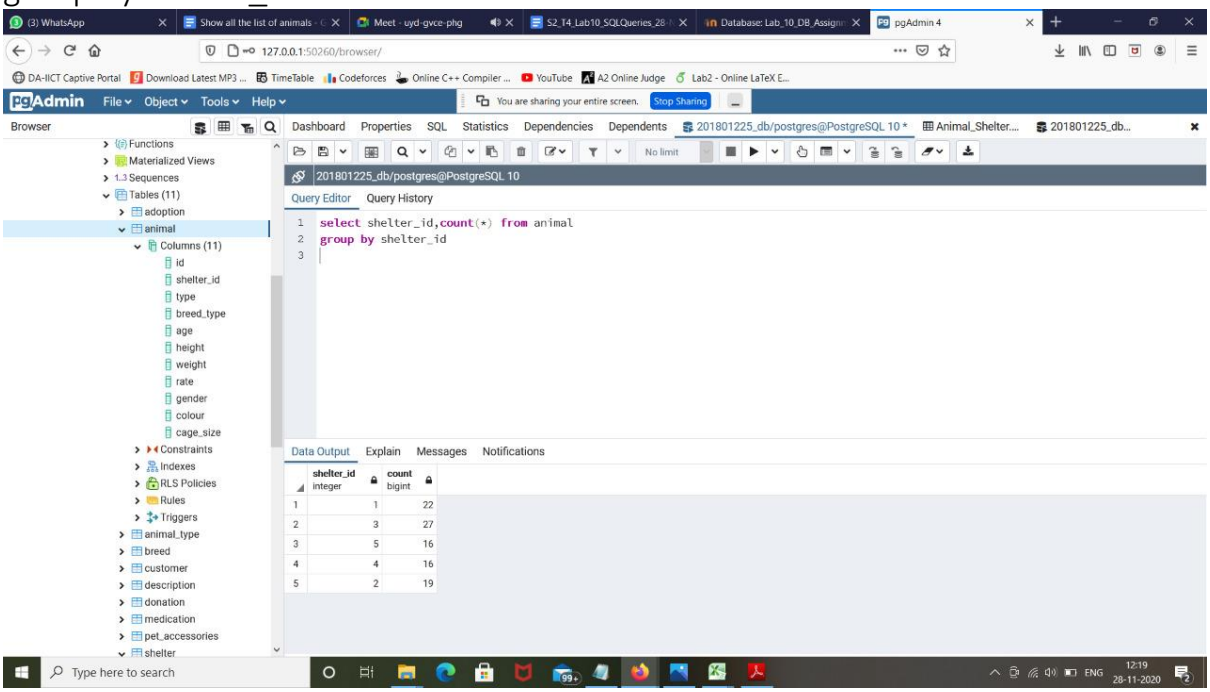
```
select * from customer
where first_name like 'L%'
```



11) List the count of animals in all shelters

SQL QUERY:

```
select shelter_id,count(*) from animal
group by shelter_id
```

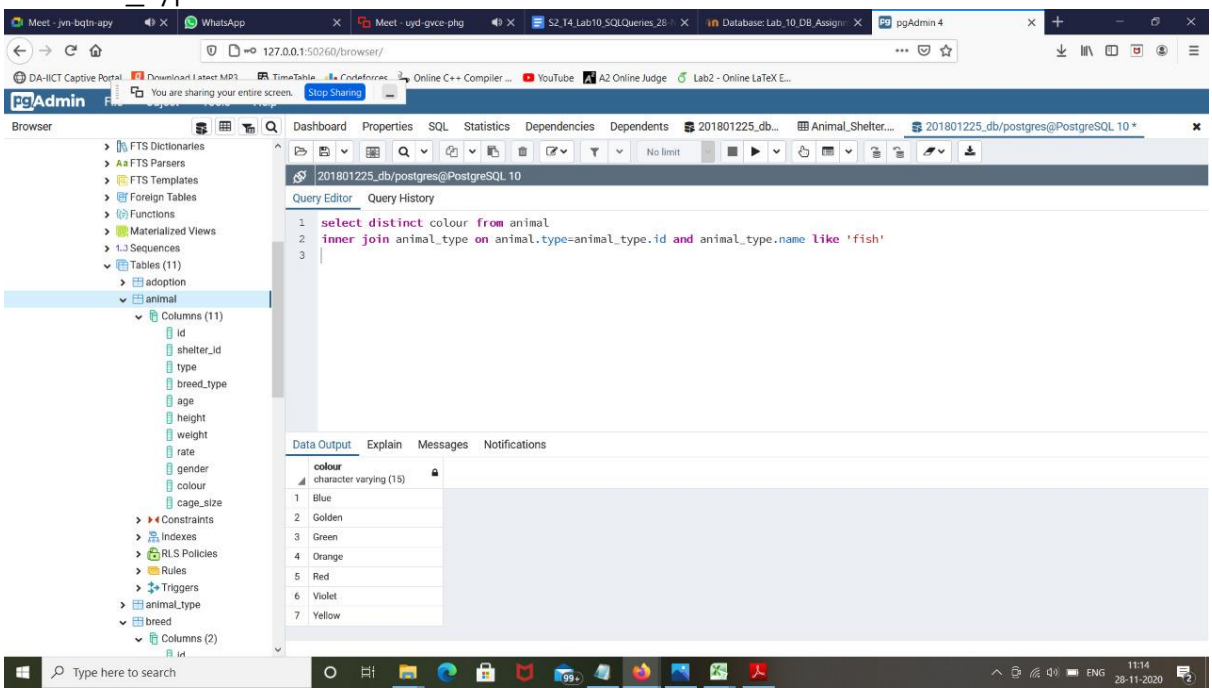


Medium Queries:

12)List the colours of all fishes

SQL QUERY:

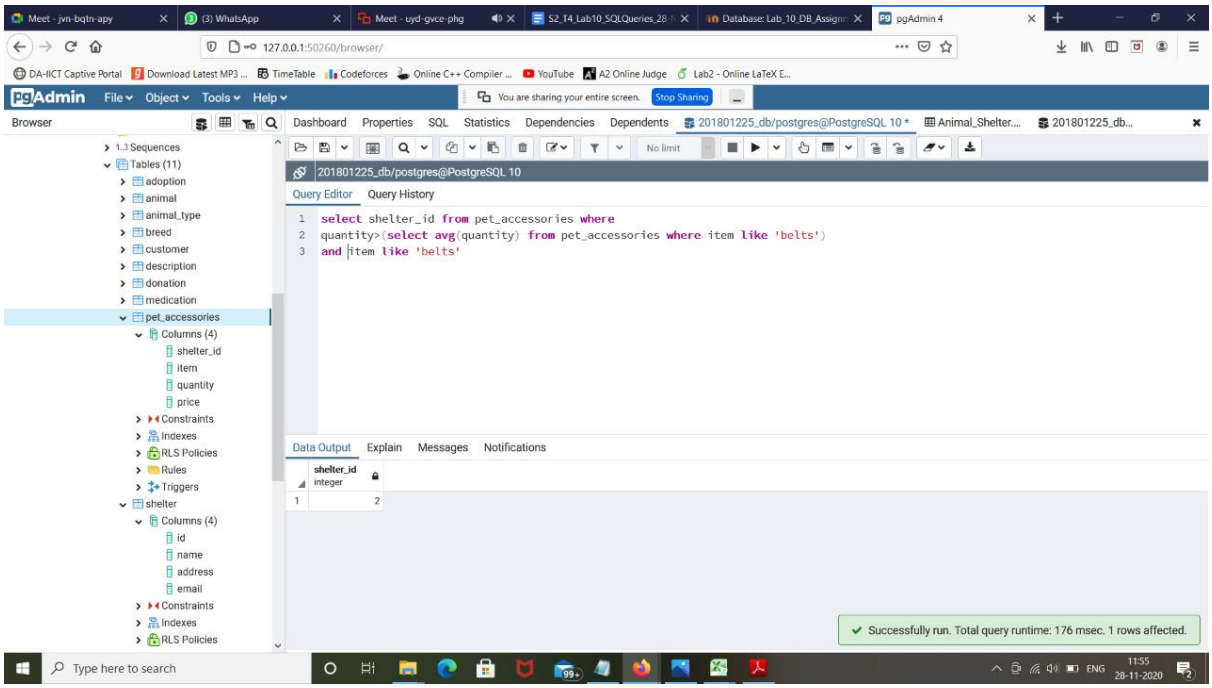
```
select distinct colour from animal
inner join animal_type on animal.type=animal_type.id and
  animal_type.name like 'fish'
```



13) List the shelters whose belts quantities are greater than the average of all quantities of belts.

SQL QUERY:

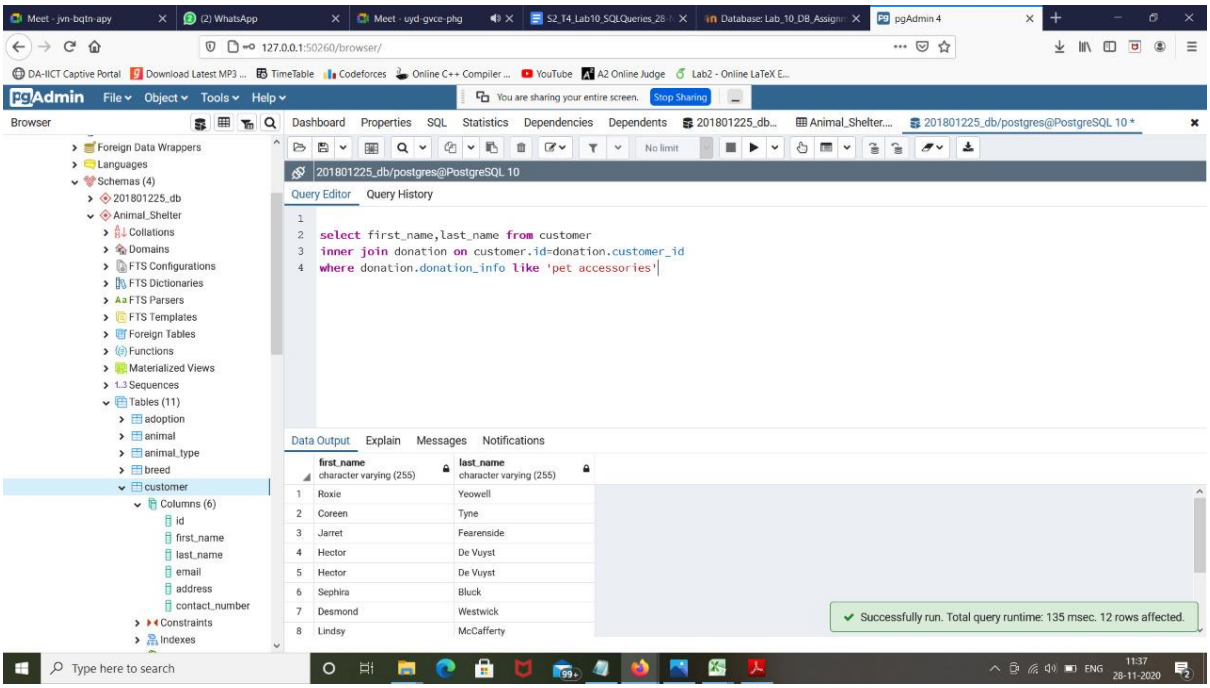
```
select shelter_id from pet_accessories where
                quantity>(select avg(quantity) from pet_accessories where
        item like 'belts')
and item like 'belts'
```



14) List customers name who have donated pet accessories

SQL QUERY:

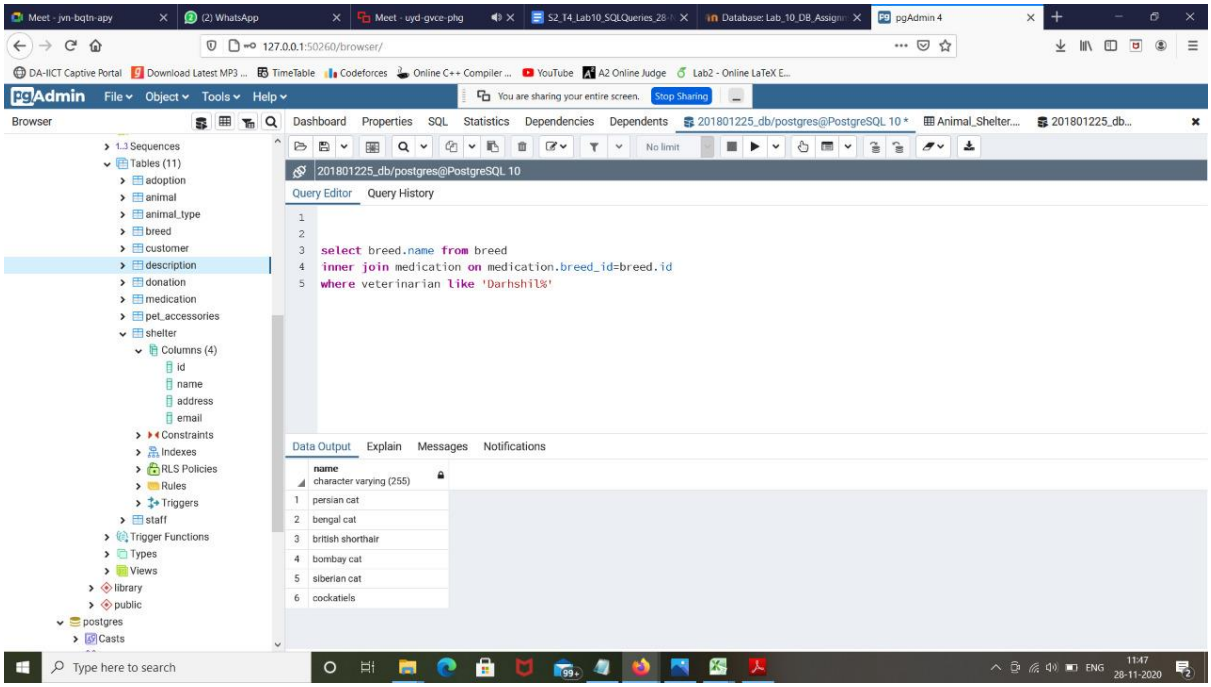
```
select first_name,last_name from customer
inner join donation on customer.id=donation.customer_id
where donation.donation_info like 'pet accessories'
```



15) List the breed names whose veterinarian is Darhshil

SQL QUERY:

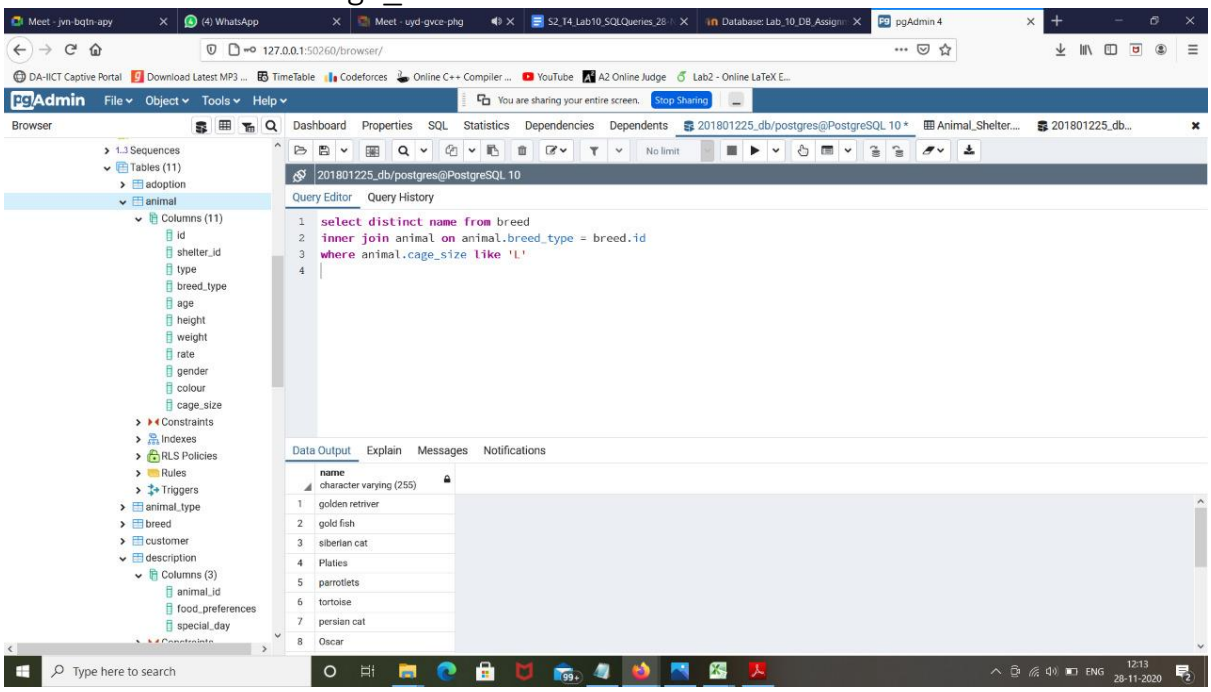
```
select breed.name from breed
inner join medication on medication.breed_id=breed.id
where veterinarian like 'Darhshil%'
```



16) List the breeds of animals with cage size L

SQL QUERY:

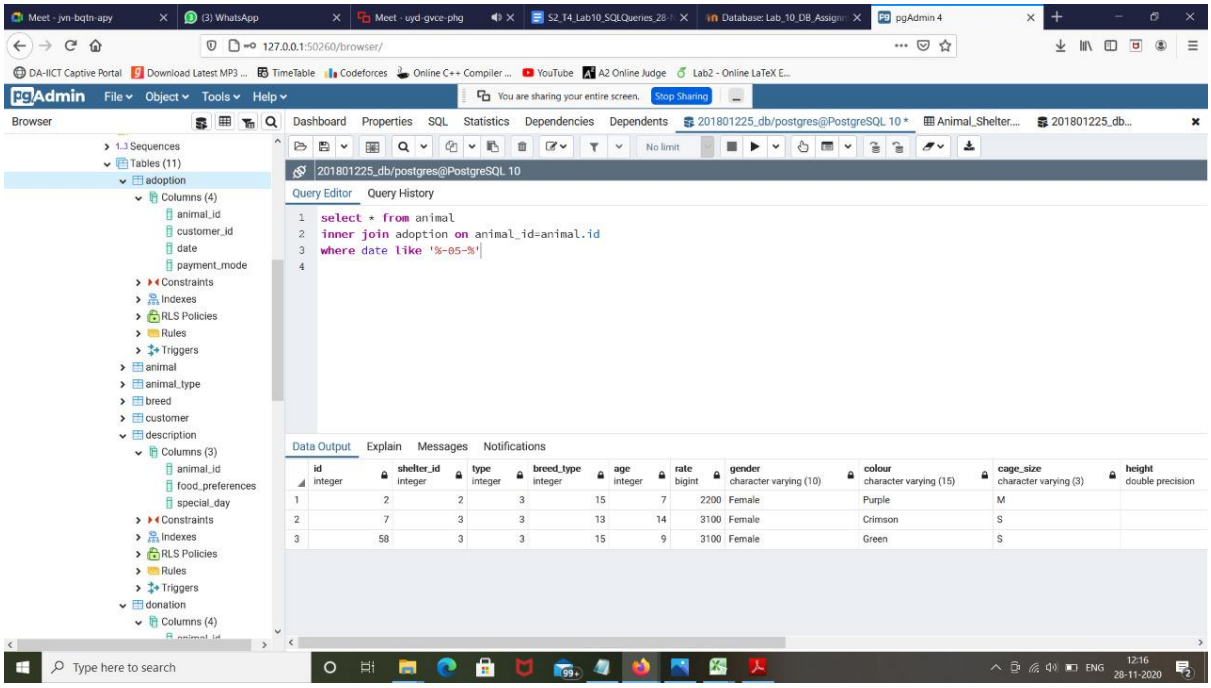
```
select distinct name from breed
inner join animal on animal.breed_type = breed.id
where animal.cage_size like 'L'
```



17) List the details on animals who were adopted in May month in this year

SQL QUERY:

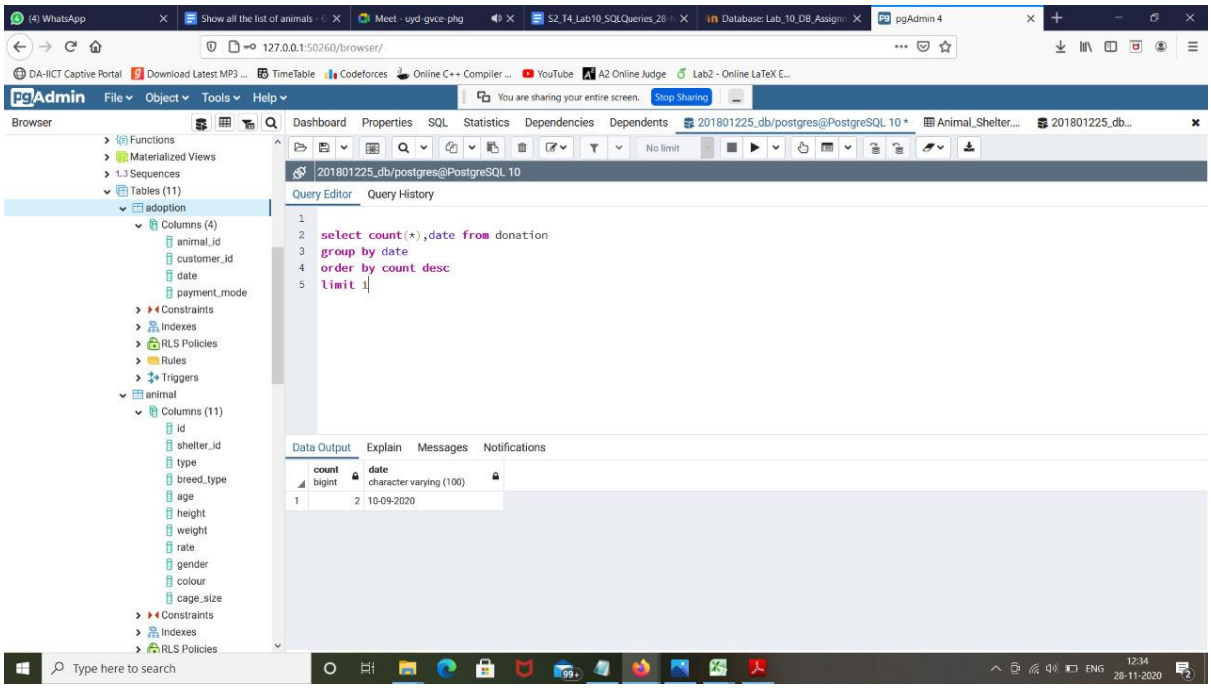
```
select * from animal
inner join adoption on animal_id=animal.id
where date like '%-05-%'
```



18) List the date on which maximum donation was done

SQL QUERY:

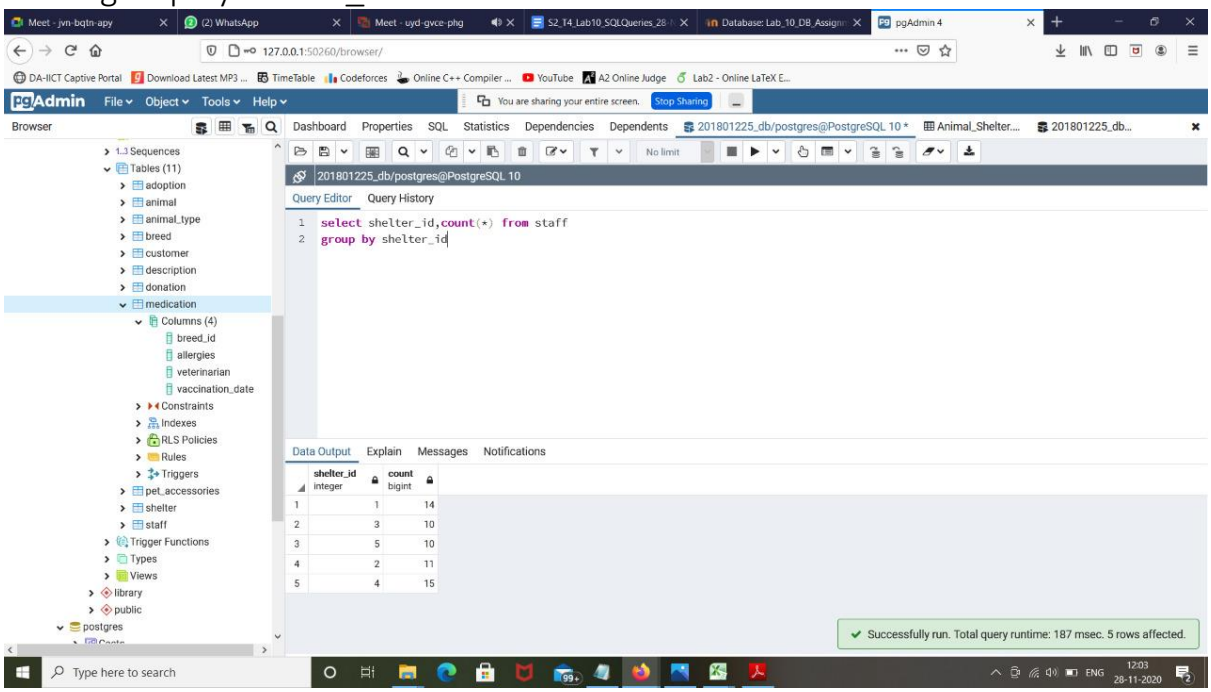
```
select count(*),date from donation
group by date
order by count desc
limit 1
```



19) Count the total number of staff for all shelters

SQL QUERY:

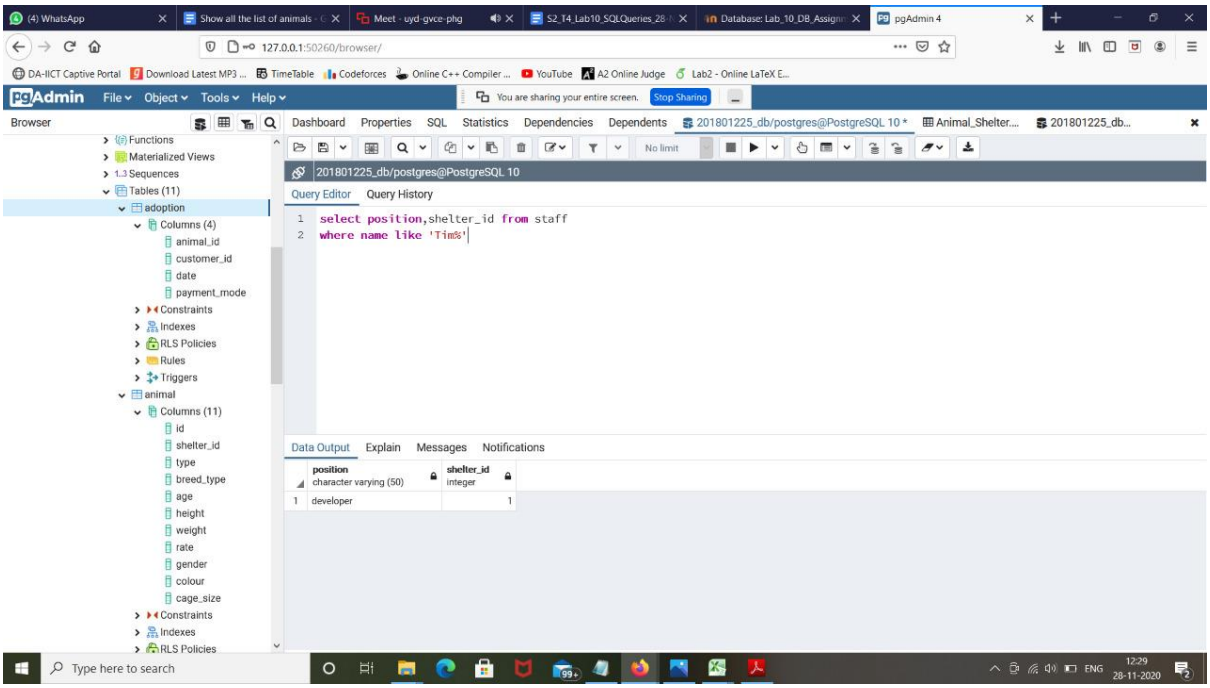
```
select shelter_id,count(*) from staff
group by shelter_id
```



20) Give the position and the shelter for Tim

SQL QUERY:

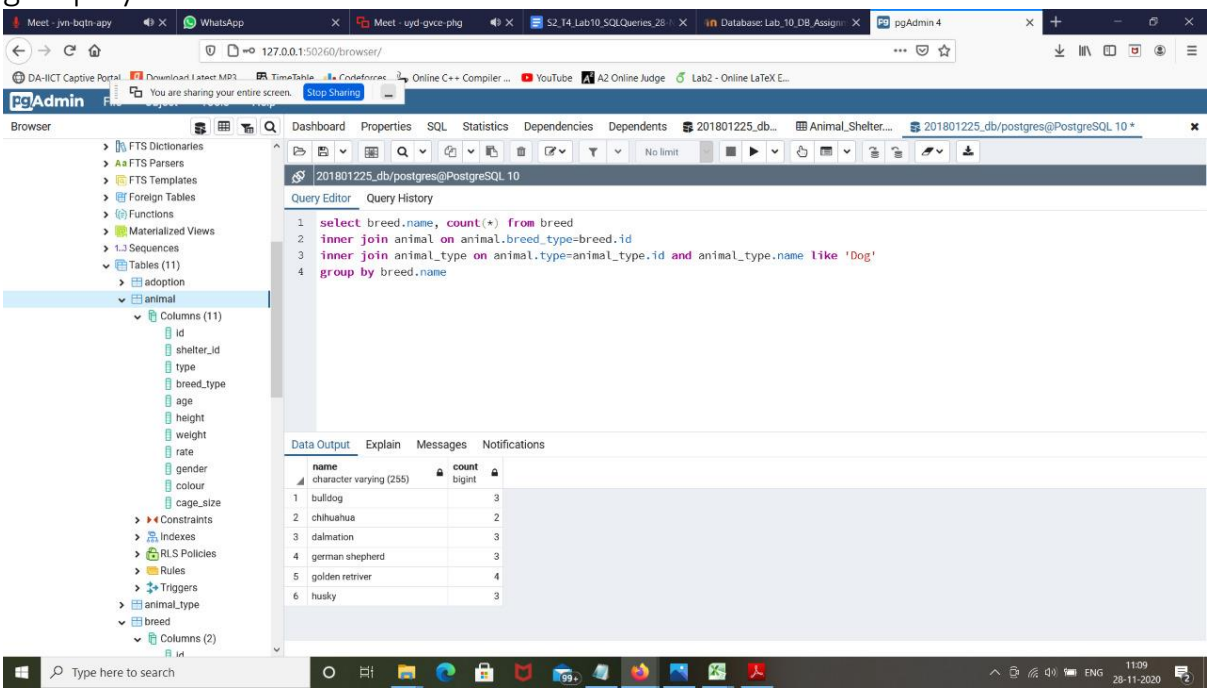
```
select position,shelter_id from staff
where name like 'Tim%'
```



21) List the breeds and it's count of dogs

SQL QUERY:

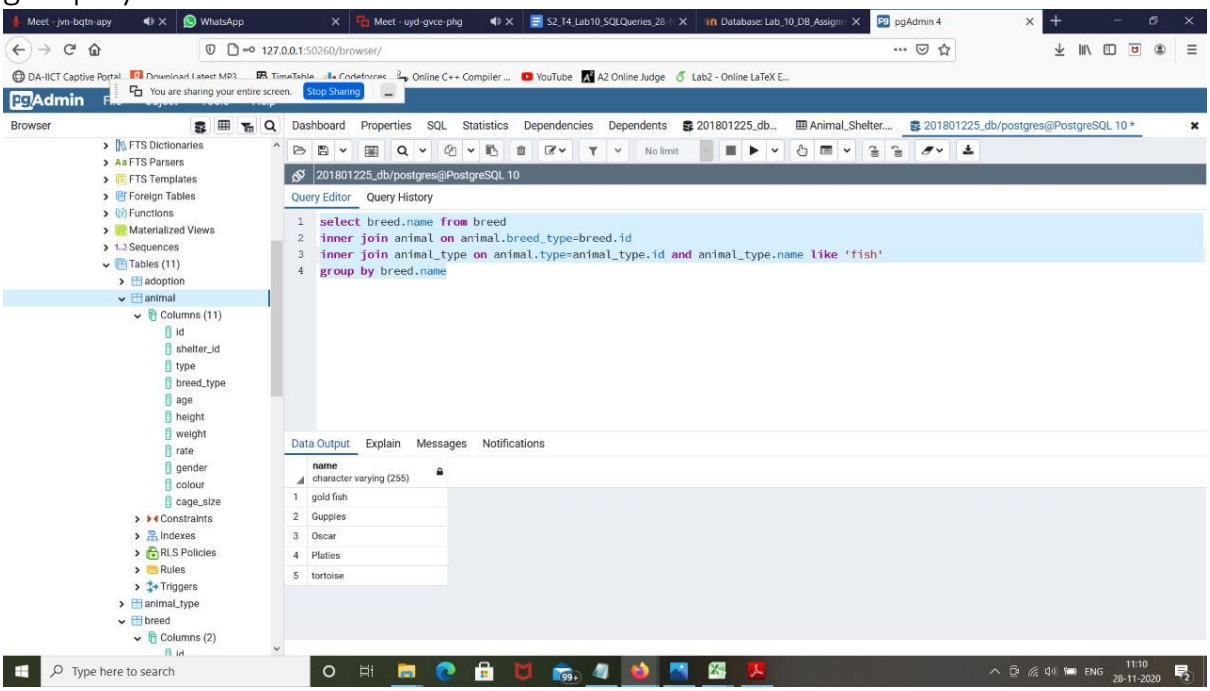
```
select breed.name, count(*) from breed
inner join animal on animal.breed_type=breed.id
inner join animal_type on animal.type=animal_type.id and
animal_type.name like 'Dog'
group by breed.name
```



22) List all the breeds of fishes

SQL QUERY:

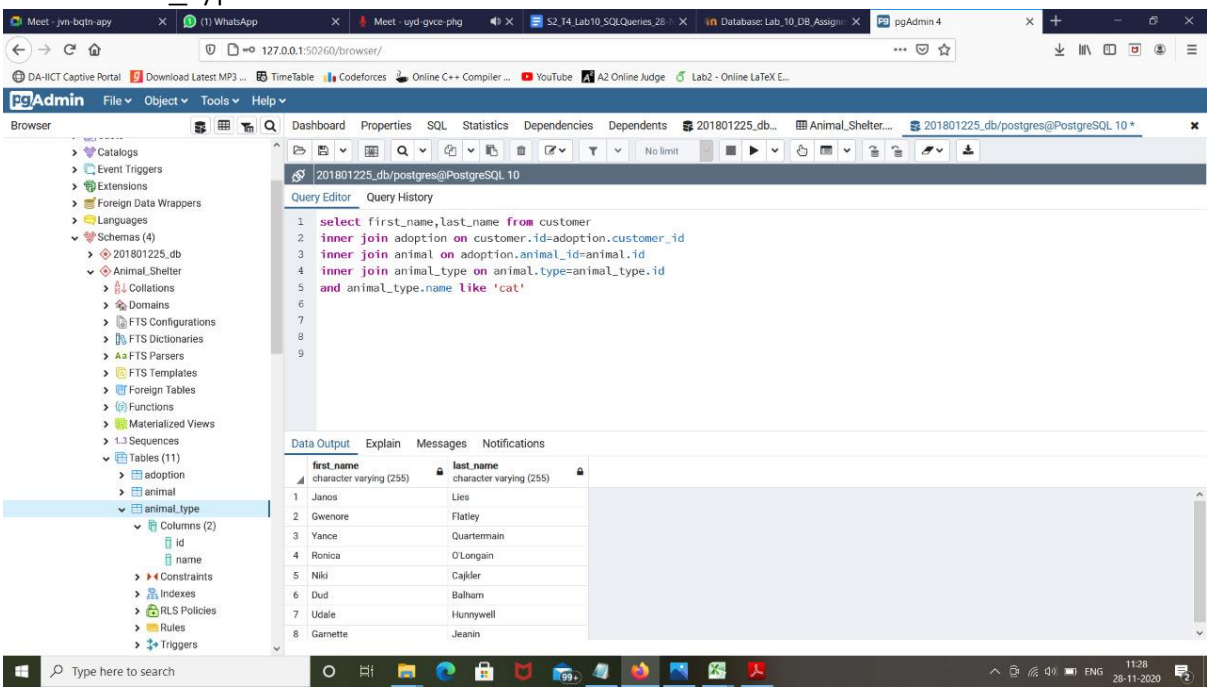
```
select breed.name from breed
inner join animal on animal.breed_type=breed.id
        inner join animal_type on animal.type=animal_type.id and
animal_type.name like 'fish'
group by breed.name
```



23) List all customers name who have adopted cat

SQL QUERY:

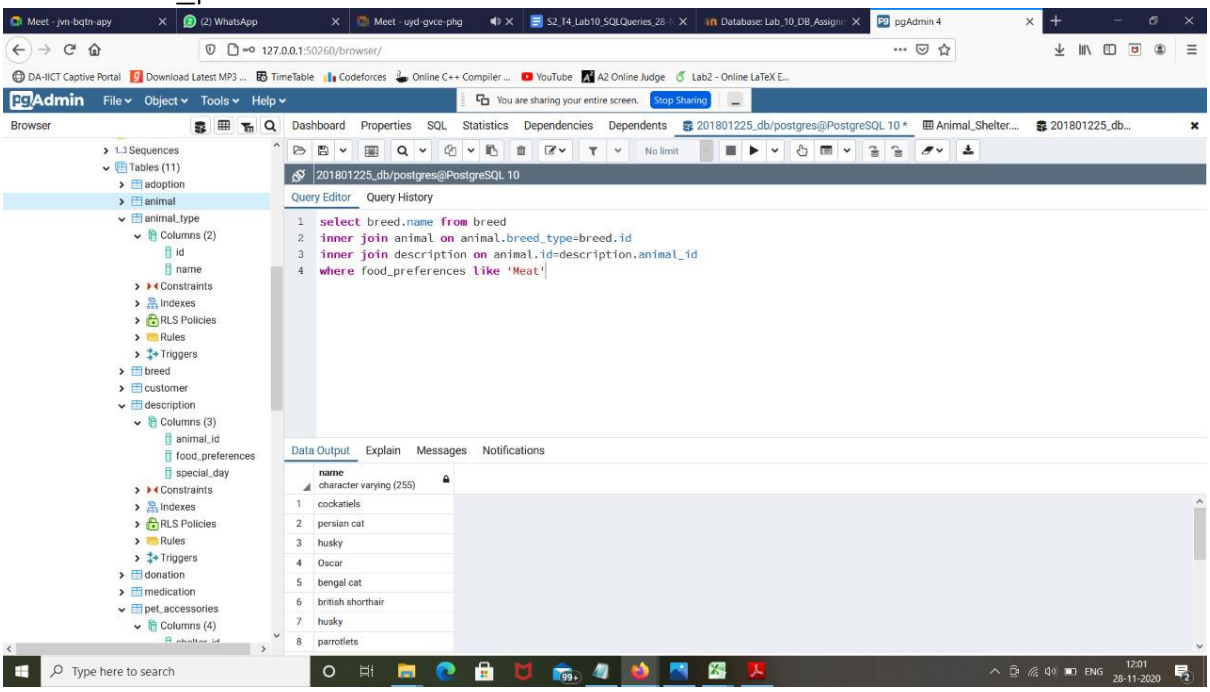
```
select first_name,last_name from customer
inner join adoption on customer.id=adoption.customer_id
inner join animal on adoption.animal_id=animal.id
inner join animal_type on animal.type=animal_type.id
and animal_type.name like 'cat'
```



24) List the breeds of animals who likes to eat Meat

SQL QUERY:

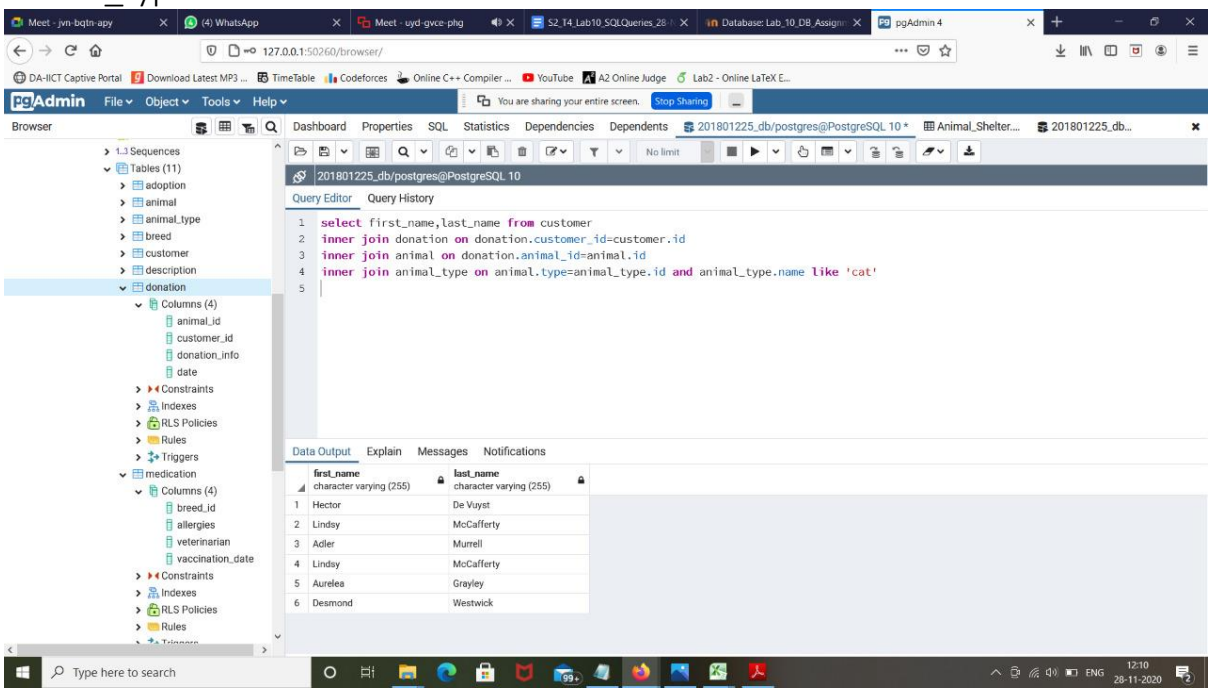
```
select breed.name from breed
inner join animal on animal.breed_type=breed.id
inner join description on animal.id=description.animal_id
where food_preferences like 'Meat'
```



25) List all customers name who have donated cats

SQL QUERY:

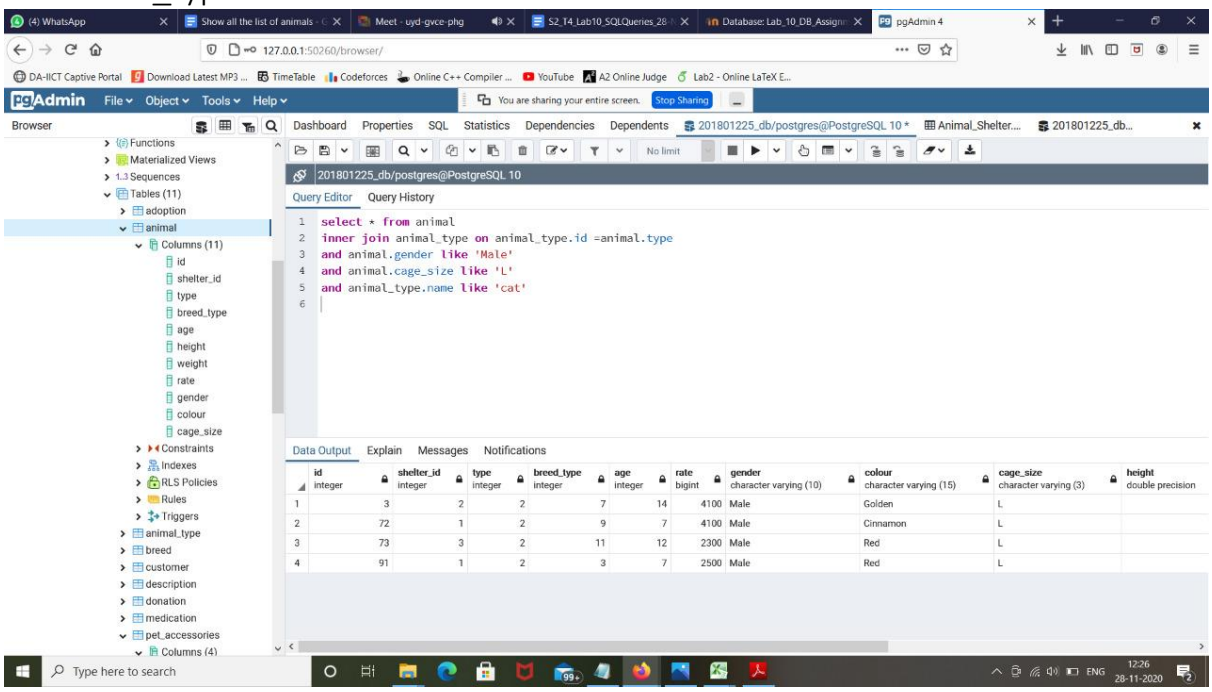
```
select first_name,last_name from customer
inner join donation on donation.customer_id=customer.id
inner join animal on donation.animal_id=animal.id
        inner join animal_type on animal.type=animal_type.id and
animal_type.name like 'cat'
```



26) List all the details of male cat with cage size L

SQL QUERY:

```
select * from animal
inner join animal_type on animal_type.id =animal.type
and animal.gender like 'Male'
and animal.cage_size like 'L'
and animal_type.name like 'cat'
```



Hard Queries

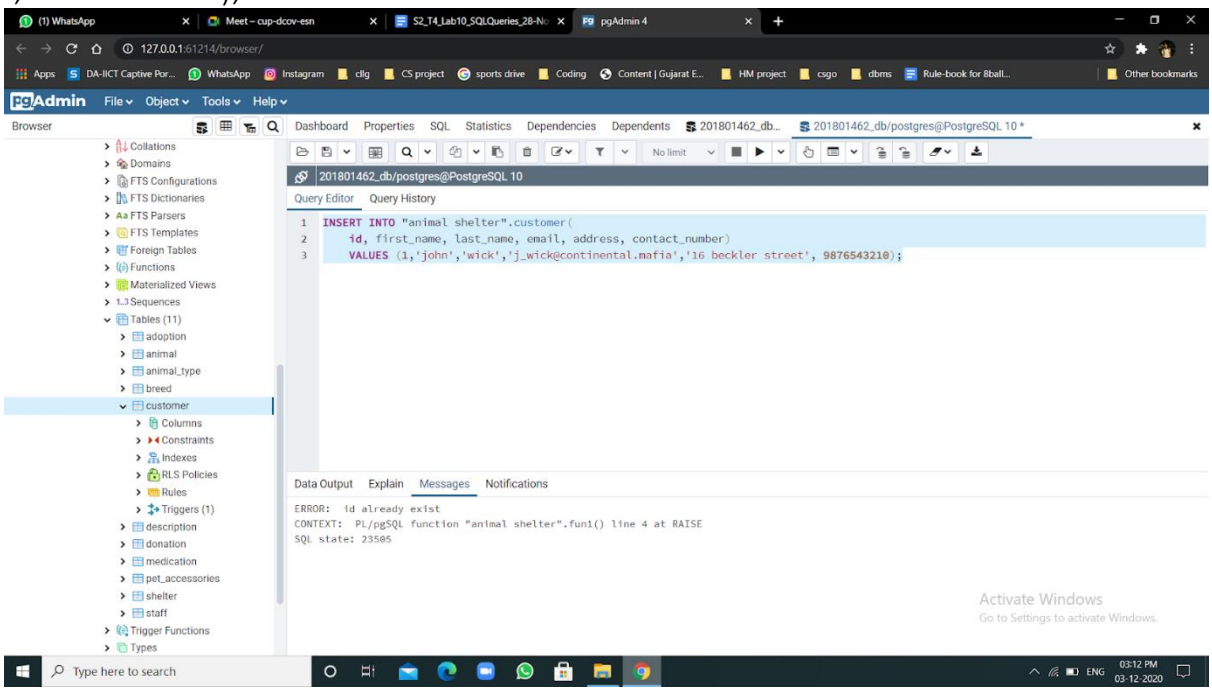
1)Trigger for customer Relation

```
set search_path to "animal_shelter";
create or replace function fun()
returns trigger
language plpgsql
as $body$
begin
	if(new.id in (select id from "animal_shelter".customer)) then
	raise UNIQUE_VIOLATION using message='id already exist';
	end if;

	if(new.email not like '%@_%._%') THEN
	RAISE UNIQUE_VIOLATION USING MESSAGE = 'Email
	Not valid ';
	END IF;

	if(new.contact_number>9999999999 or
new.contact_number<1000000000) then
	RAISE UNIQUE_VIOLATION USING MESSAGE = 'contact number is invalid';
	END IF;

	return new;

end
$body$


create trigger t1
before insert
on animal_shelter.customer
for each row
execute procedure fun();
```

27) Insert in customer table which returns a trigger if the customer id already exist in the table

SQL QUERY:

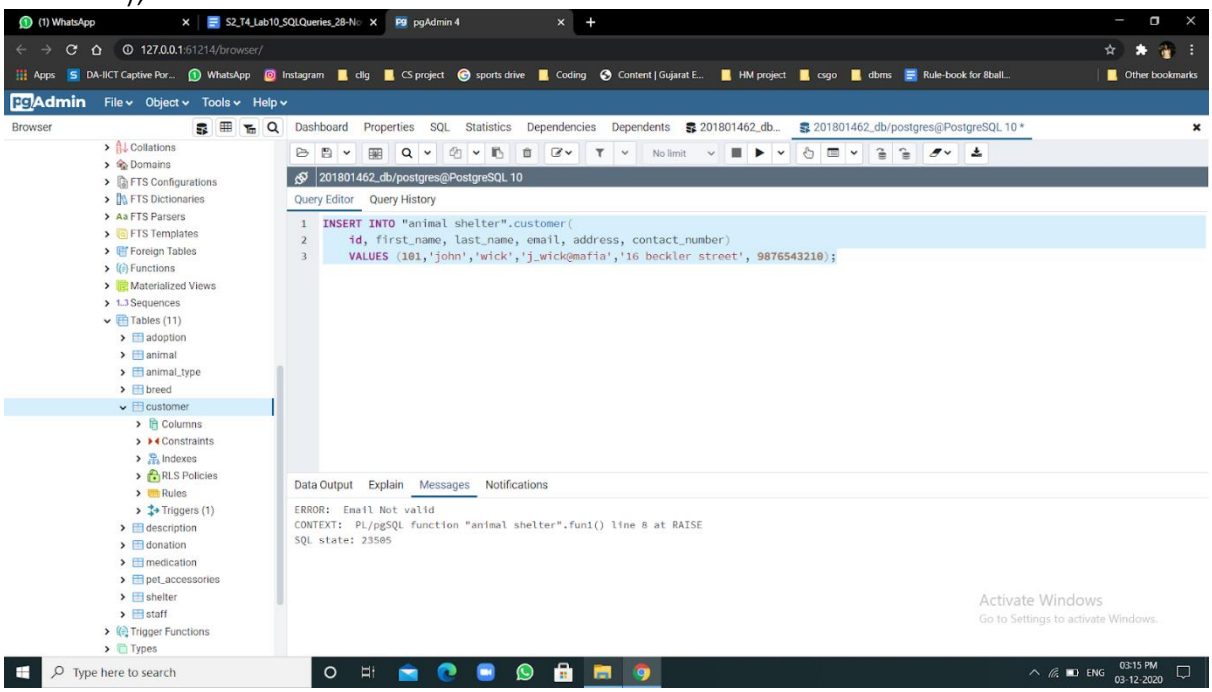
```
INSERT INTO "animal shelter".customer(
id, first_name, last_name, email, address, contact_number)
VALUES (1, 'john' , 'wick' , 'j_wick@continental.mafia' , '16 beckler street
', 9876543210);
```



28) Insert into customer table which returns a trigger if email_id is not valid

SQL QUERY:

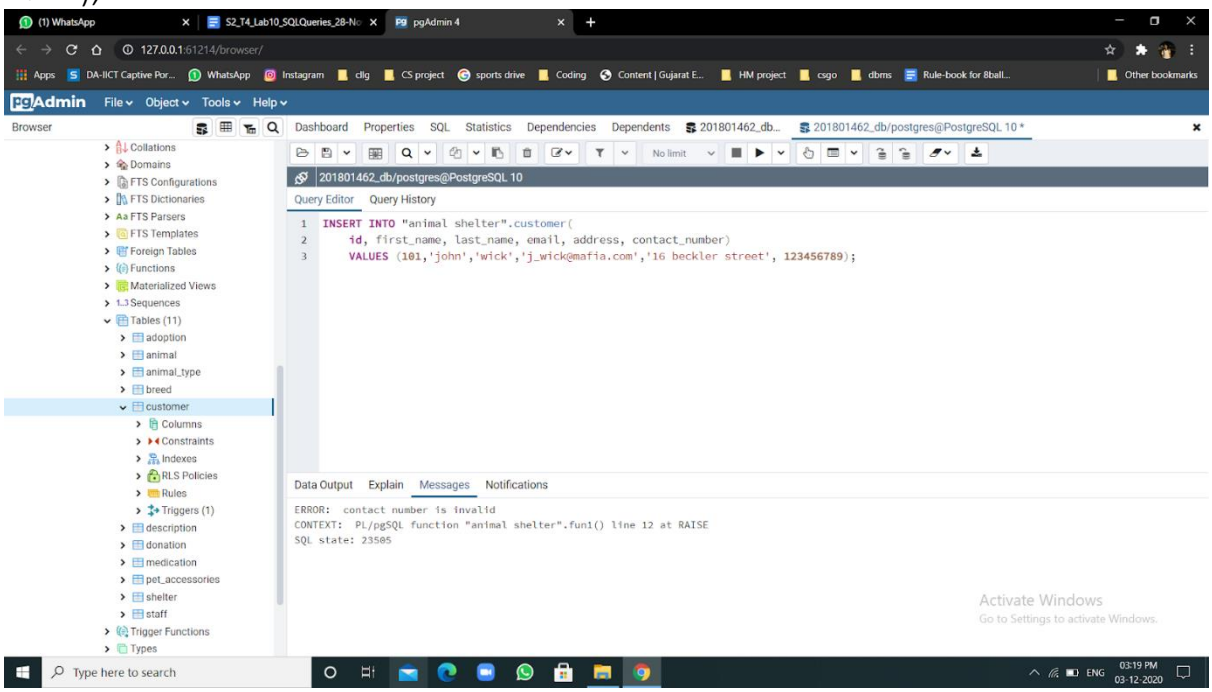
```
INSERT INTO "animal shelter".customer(
id, first_name, last_name, email, address, contact_number)
VALUES (101 , 'john','wick' , 'j_wick@mafia' , '16 beckler street'
,  9876543210);
```



29) Insert into customer table which returns a trigger if contact number of customer is invalid, (not a 10 digit number)

SQL QUERY:

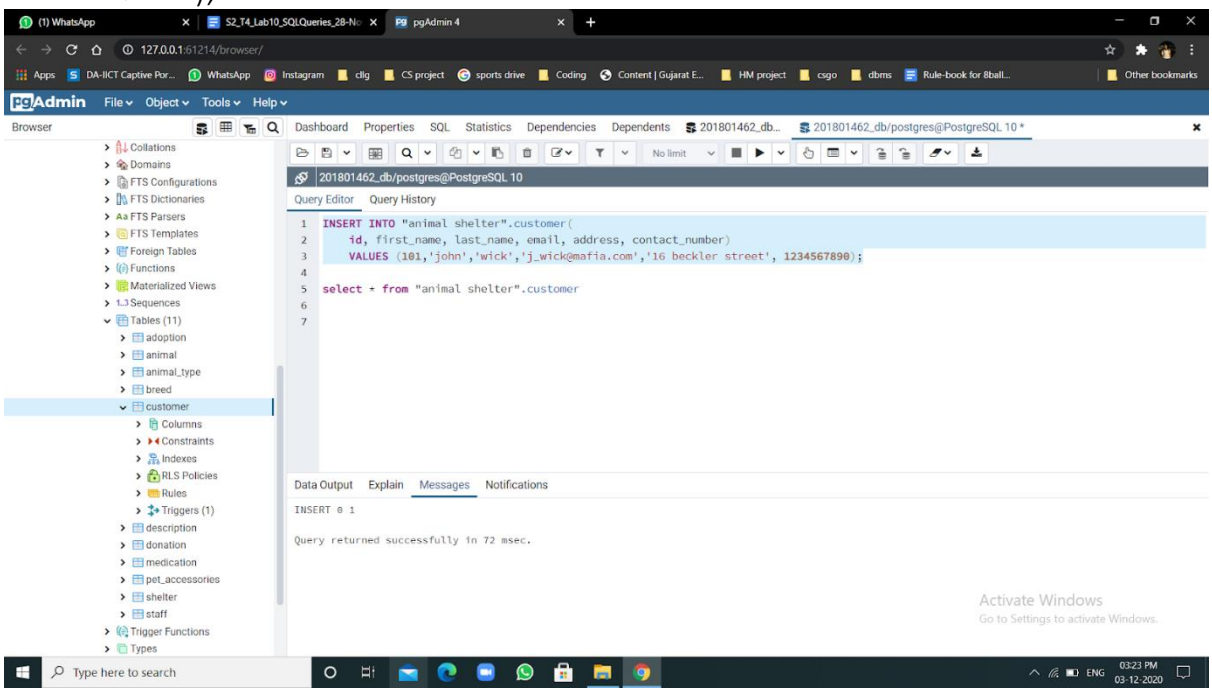
```
INSERT INTO "animal shelter".customer(
id, first_name, last_name, email, address, contact_number)
VALUES (101,'john','wick','j_wick@mafia.com','16 beckler street',
123456789);
```

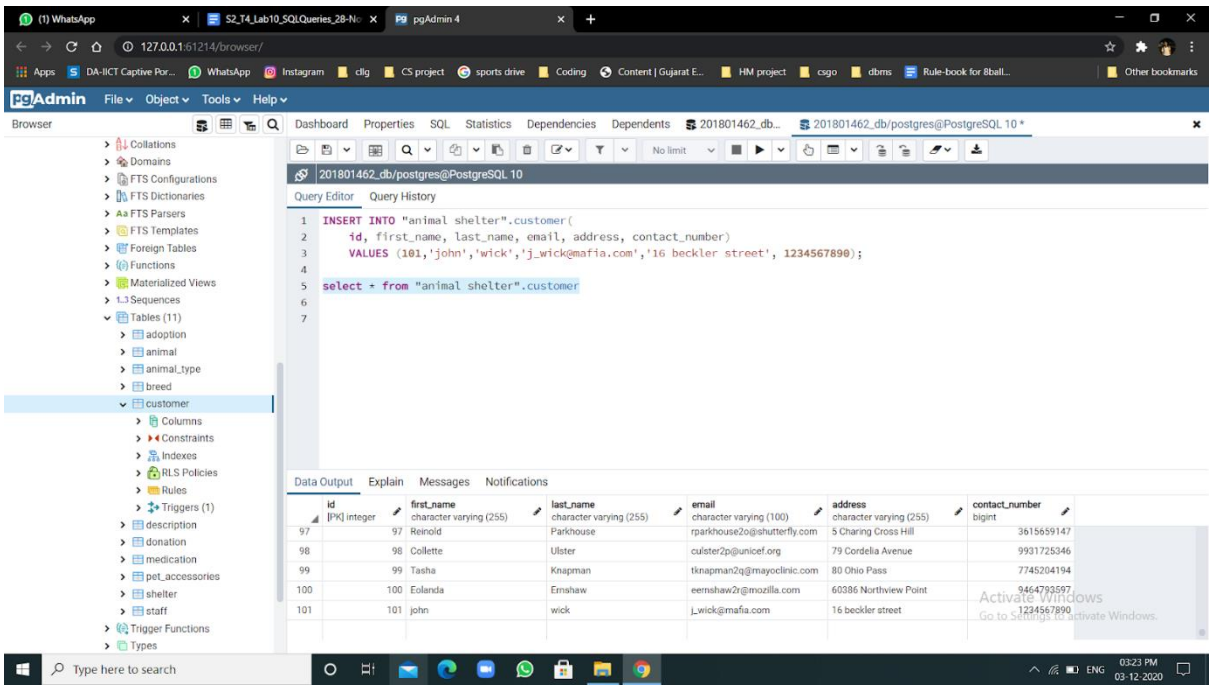


30) Insert into customer table with all valid details which returns the trigger for successful query and add into the customer table

SQL QUERY:

```
INSERT INTO "animal shelter".customer(
        id, first_name, last_name, email, address, contact_number)
        VALUES (101,'john','wick','j_wick@mafia.com','16 beckler street',
1234567890);
```





2) Trigger for Adoption Relation

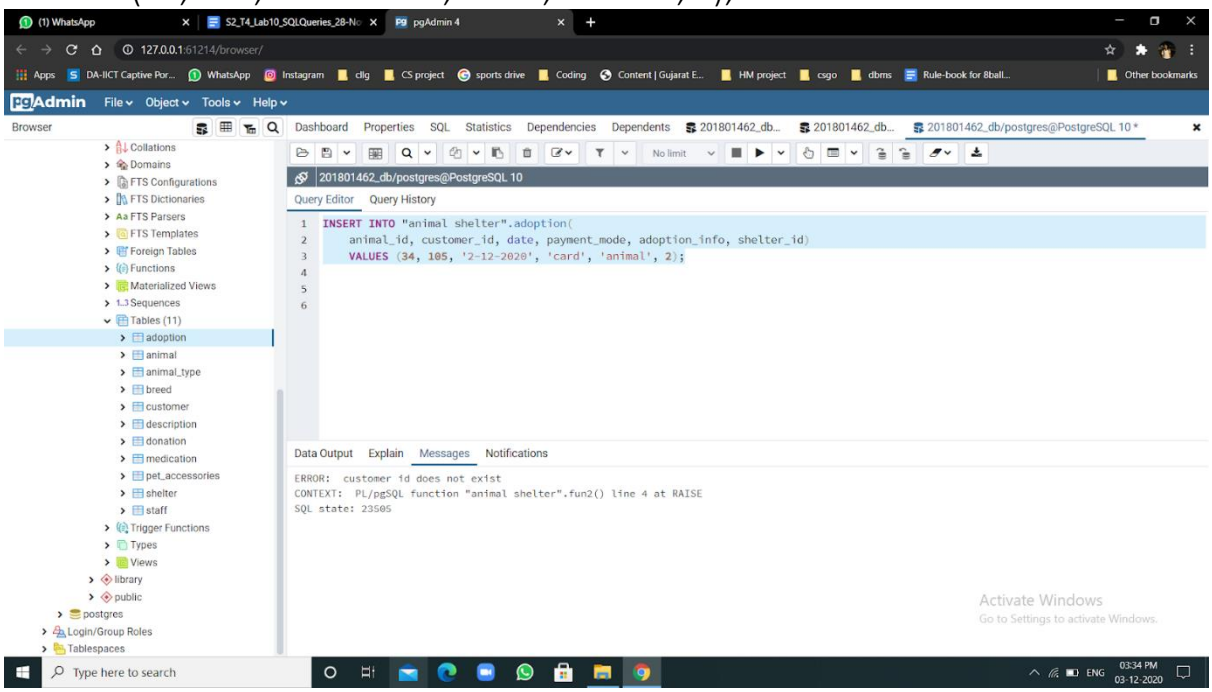
```
set search_path to animal_shelter;
create or replace function fun2()
returns trigger
language plpgsql
as $body$
begin
	if(new.customer_id not in(select id from animal_shelter.customer)) then
	raise UNIQUE_VIOLATION using message='customer id does not exist';
	end if;

	if(new.adoption_info like 'animal') then
		if(new.animal_id in(select adoption.animal_id from
animal_shelter.adoption )) then
		raise UNIQUE_VIOLATION using message='this animal is already
adopted';
		end if;

		if(new.shelter_id != (select shelter_id from animal_shelter.animal
where id =new.animal_id)) then
		raise UNIQUE_VIOLATION using message='this animal does not
exist in this shelter';
		end if;

		return new;

	else
		if((select quantity from animal_shelter.pet_accessories where
		animal_shelter.pet_accessories.shelter_id=new.shelter_id and
		animal_shelter.pet_accessories.item like new.adoption_info) =0)
then
		raise UNIQUE_VIOLATION using message='this item does not exist
in this shelter';
		end if;

		update animal_shelter.pet_accessories set
		quantity=quantity-1
		where animal_shelter.pet_accessories.shelter_id=new.shelter_id
and
```

```
            animal_shelter.pet_accessories.item like new.adoption_info ;
        end if;

        return new;



end
$body$


create trigger t2
before insert
on animal_shelter.adoption
for each row
execute procedure fun2();
```

31) Create a trigger for adoption table which checks if the customer wants to adopt then his customer id exist or not in the database

SQL QUERY:

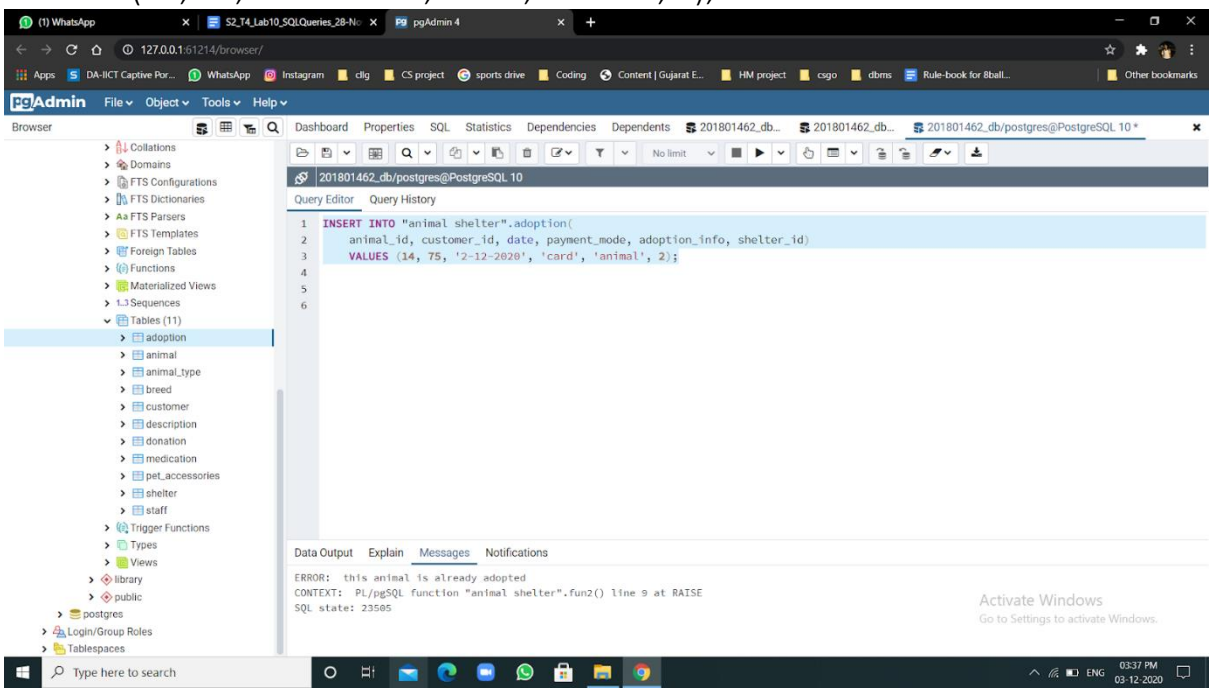
```
        INSERT INTO "animal shelter".adoption(
        animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
        VALUES (34, 105, '2-12-2020', 'card', 'animal', 2);
```



32) A customer wants to adopt a pet which is already adopted, returns a trigger with before inserting into adoption table

SQL QUERY:

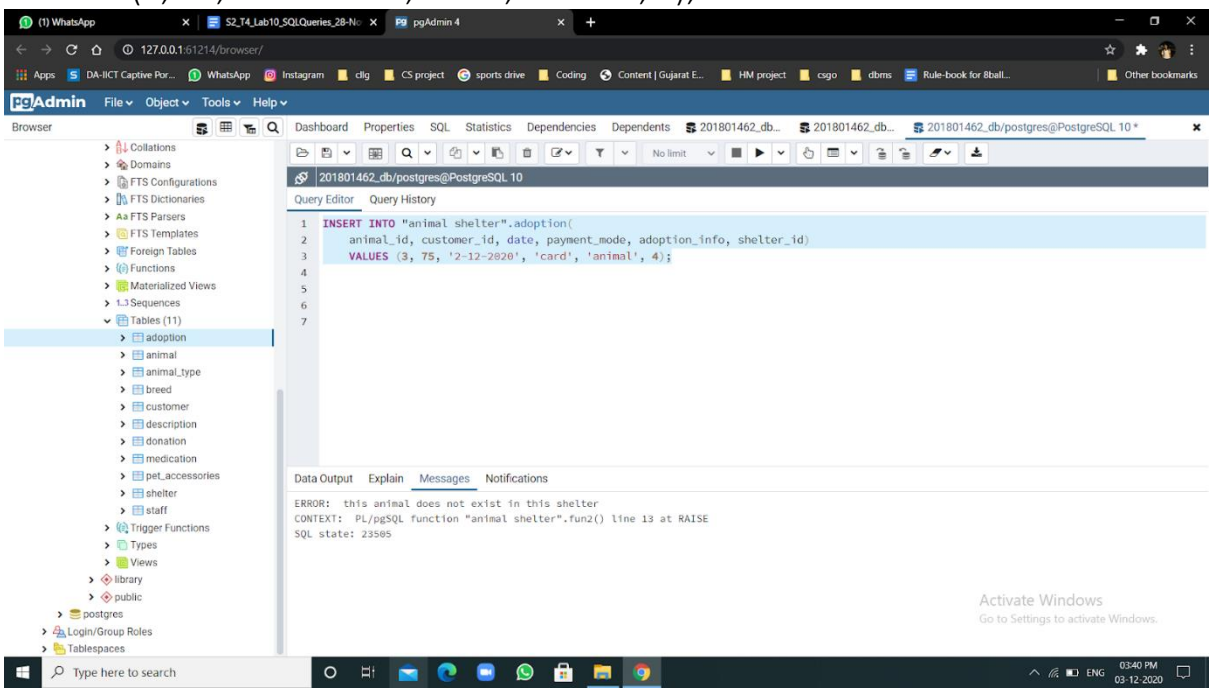
```
INSERT INTO "animal shelter".adoption(
animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
VALUES (14, 75, '2-12-2020', 'card', 'animal', 2);
```



33) A customer wants to adopt a pet which is not present in the given shelter, return a trigger doing the same.

SQL QUERY:

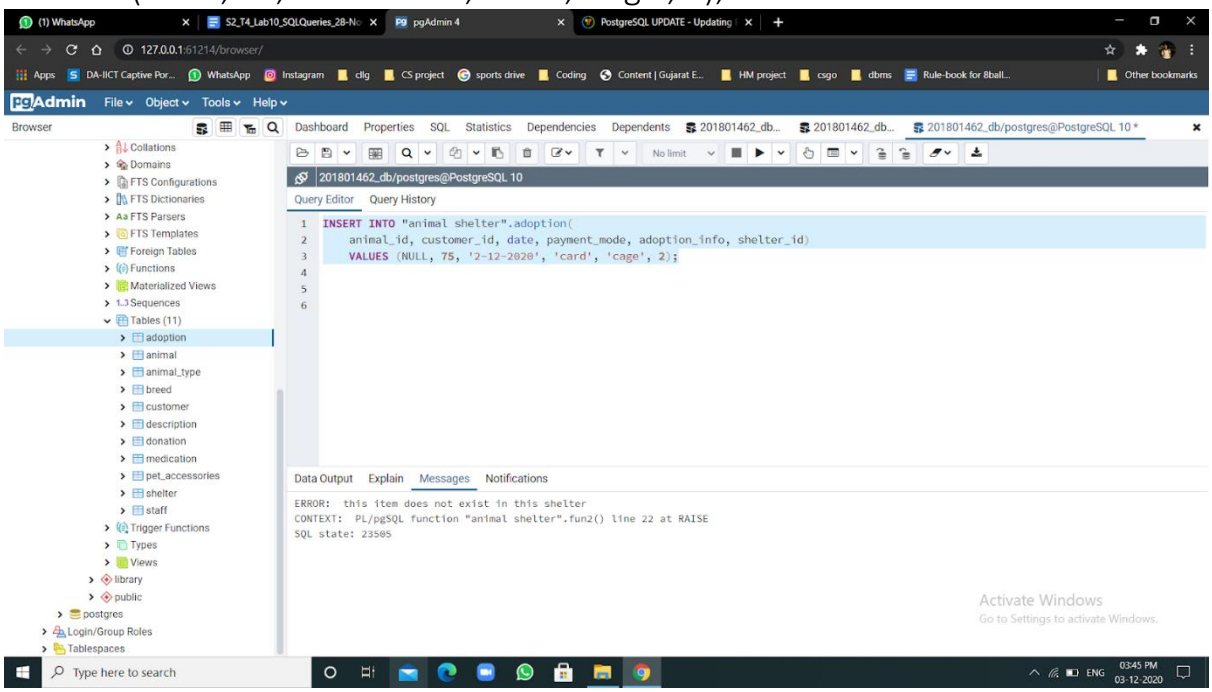
```
INSERT INTO "animal shelter".adoption(
animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
VALUES (3, 75, '2-12-2020', 'card', 'animal', 4);
```



34) A customer wants to buy a pet accessory from a shelter but that item is not present in the shelter, return a trigger for the same.

SQL QUERY:

```
INSERT INTO "animal shelter".adoption(
animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
VALUES (NULL, 75, '2-12-2020', 'card', 'cage', 2);
```

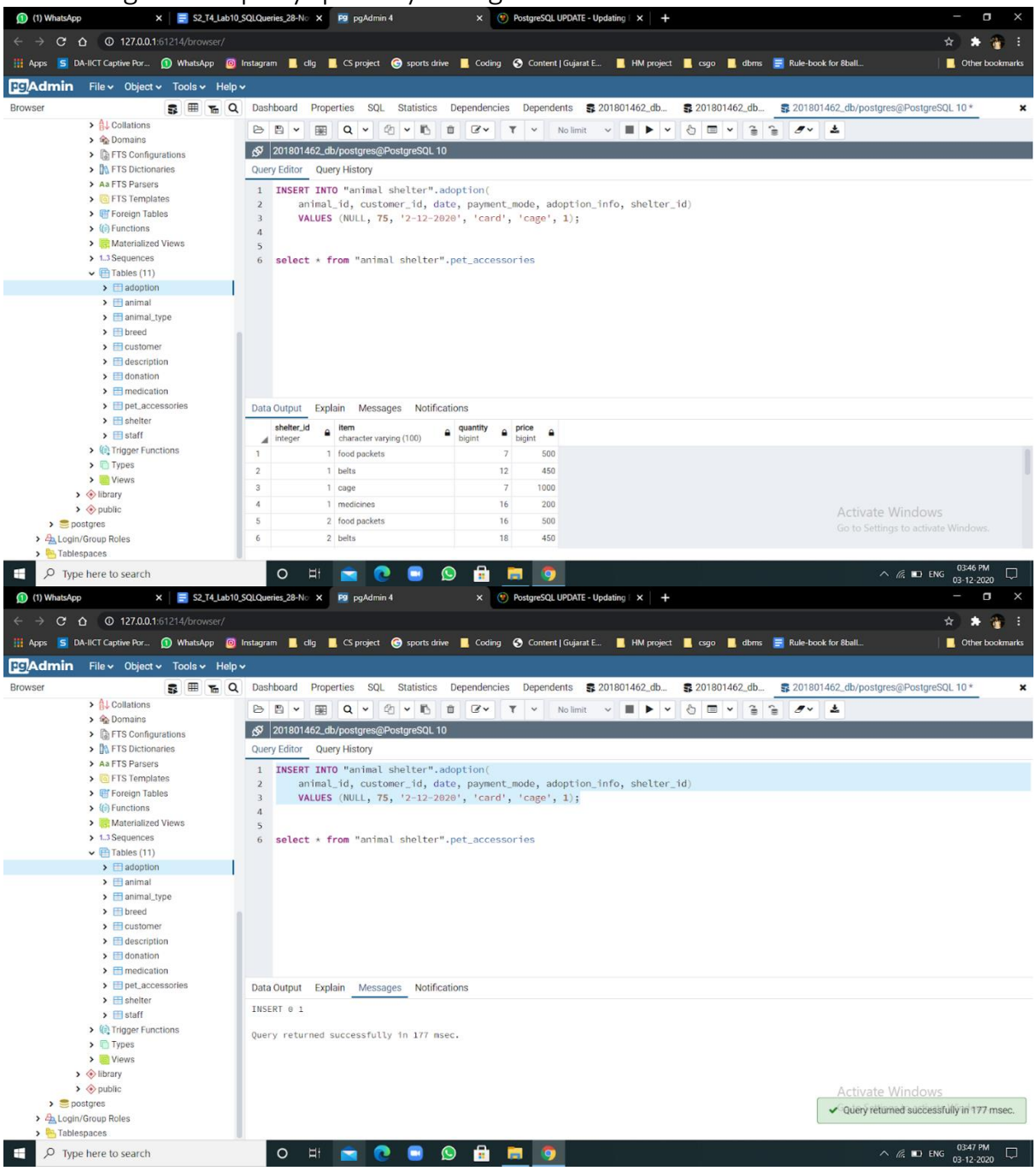


35) Customer wants to buy any pet accessories from the shelter, create a trigger which decreases the quantity after every adoption/buy of pet accessories

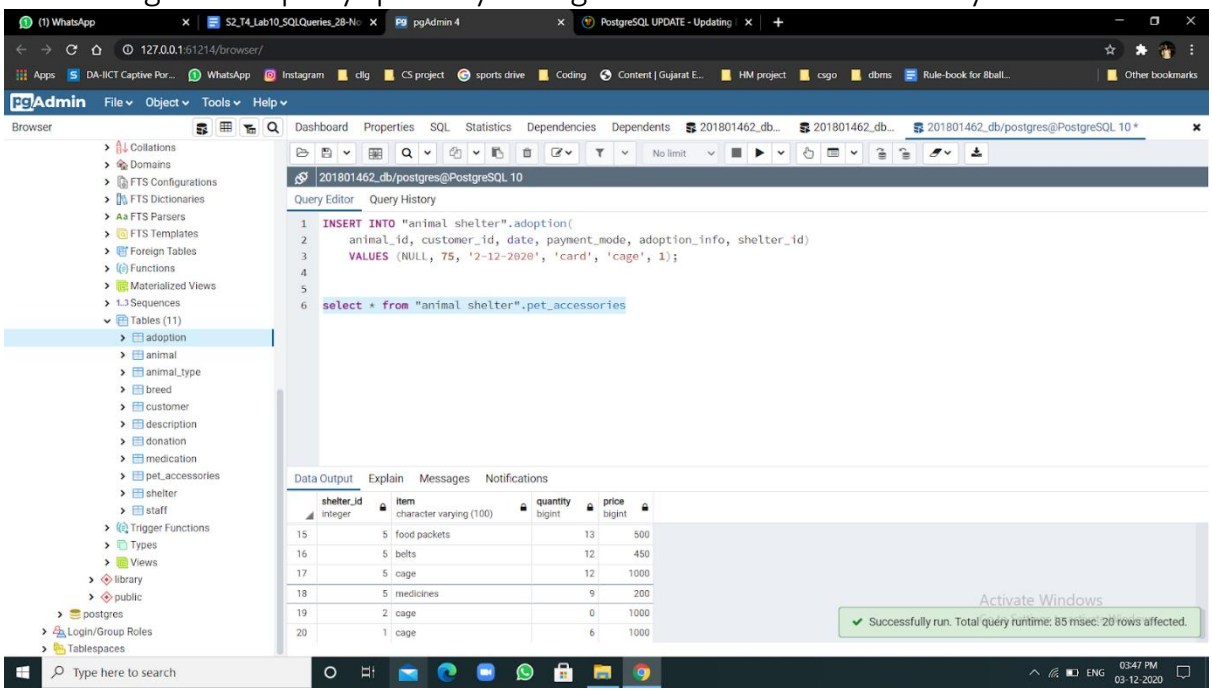
SQL QUERY:

```
INSERT INTO "animal shelter".adoption(
animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
VALUES (NULL, 75, '2-12-2020', 'card', 'cage', 1);
```

//before running above query quantity of cage in shelter 1:



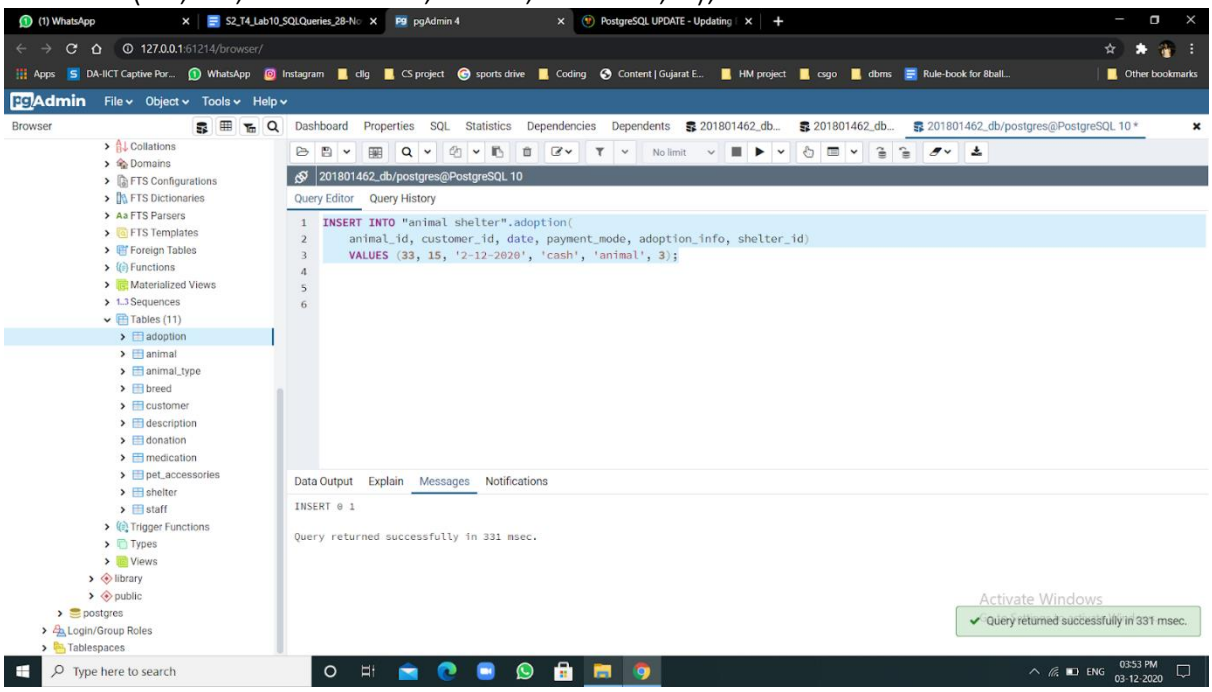
////after running above query quantity of cage in shelter 1 decreases by 1:

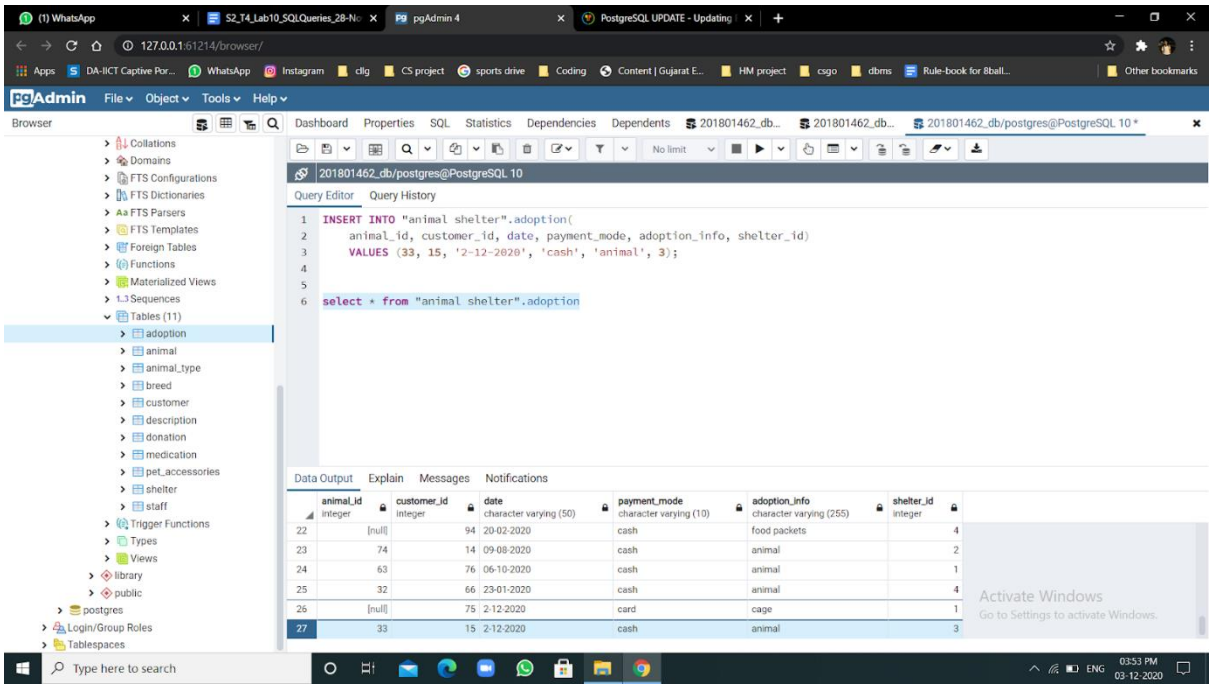


36) Customer adopts a pet with all valid entries, return a trigger for the same

SQL QUERY:

```
INSERT INTO "animal shelter".adoption(
animal_id, customer_id, date, payment_mode, adoption_info, shelter_id)
VALUES (33, 15, '2-12-2020', 'cash', 'animal', 3);
```





3) Trigger for Donation Relation

```
set search_path to animal_shelter;
create or replace function fun3()
returns trigger
language plpgsql
as $body$
begin
	if(new.customer_id not in(select id from animal_shelter.customer)) then
	raise UNIQUE_VIOLATION using message='customer id does not exist';
	end if;

	if(new.donation_info like 'animal') then
		if(new.animal_id not in (select id from animal_shelter.animal))
then
		raise UNIQUE_VIOLATION using message='please enter the animal
details first';
		end if;

		if(new.animal_id in (select animal_id from
animal_shelter.donation)) then
		raise UNIQUE_VIOLATION using message='enter valid animal id';
		end if;

		if(new.animal_id in (select animal_id from
animal_shelter.adoption)) then
		raise UNIQUE_VIOLATION using message='input is invalid: this
animal is adopted';
		end if;

		if(new.shelter_id != (select shelter_id from animal_shelter.animal
where id =new.animal_id)) then
		raise UNIQUE_VIOLATION using message='input is invalid :this
animal does not exist in this shelter';
		end if;


		return new;
	else
		if(new.animal_id is not null) then
```

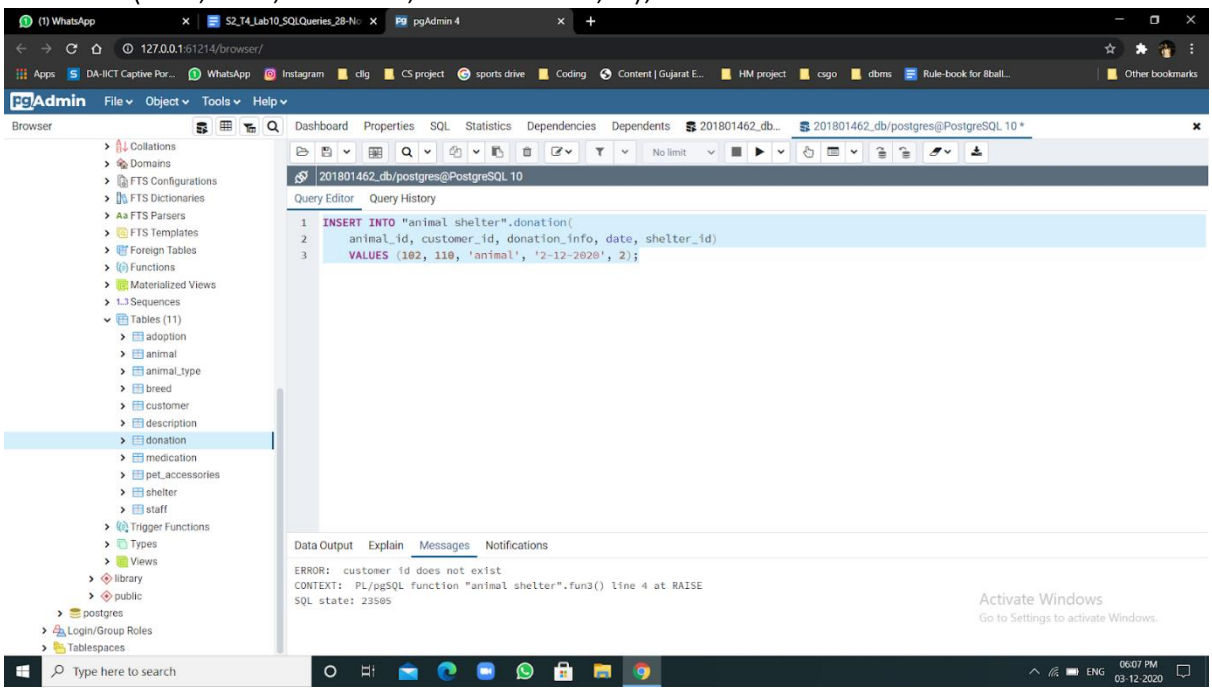
```
                raise UNIQUE_VIOLATION using message='for donation of pet
accessories animal id should be NULL';
                end if;

                update animal_shelter.pet_accessories set
                quantity=quantity+1
                where animal_shelter.pet_accessories.shelter_id=new.shelter_id
and
                animal_shelter.pet_accessories.item like new.donation_info ;

                return new;
        end if;
end
$body$



create trigger t3
before insert
on animal_shelter.donation
for each row
execute procedure fun3();
```

Queries:

37) A customer wants to donate in our shelter, return a trigger if that customer with given id exists or not.

SQL QUERY:

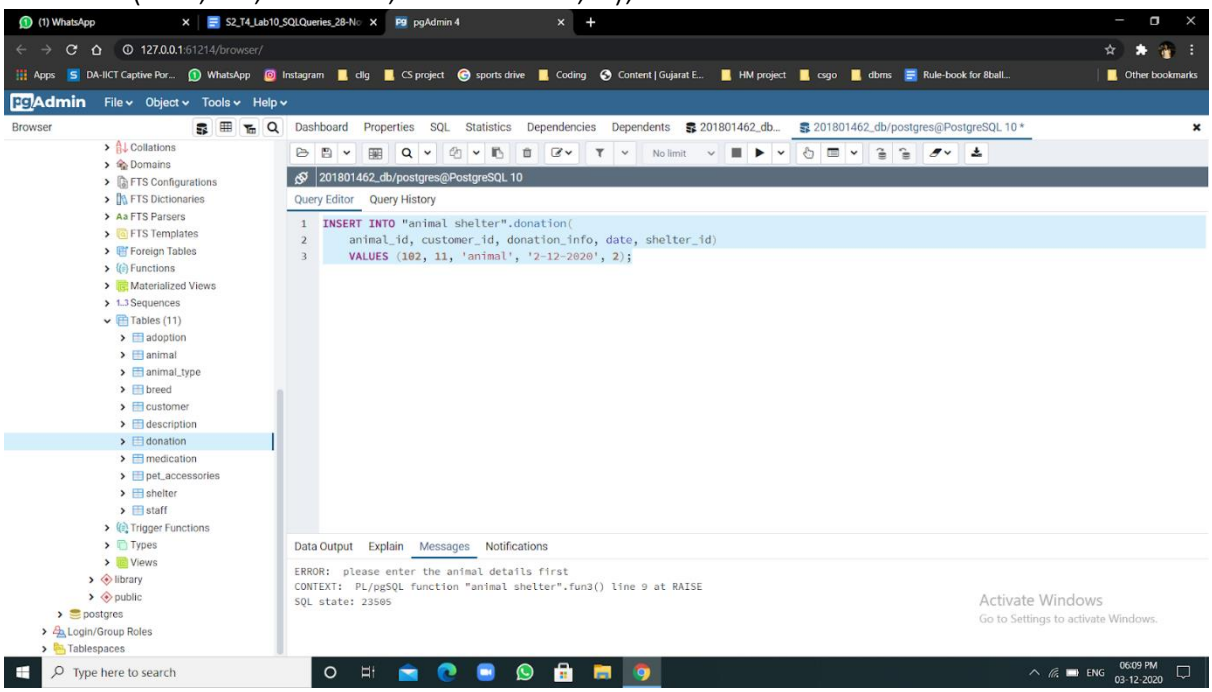
```
INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (102, 110, 'animal', '2-12-2020', 2);
```



38) A customer wants to donate an animal, return a trigger if that animal first exist in our database.

SQL QUERY:

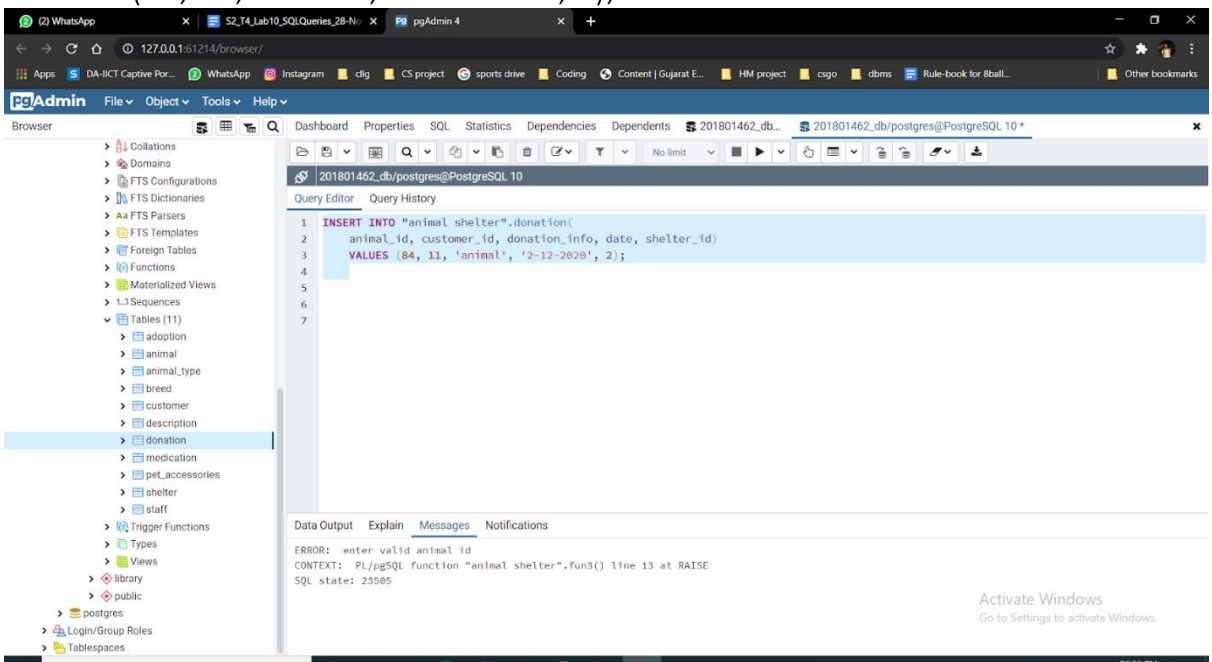
```
INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (102, 11, 'animal', '2-12-2020', 2);
```

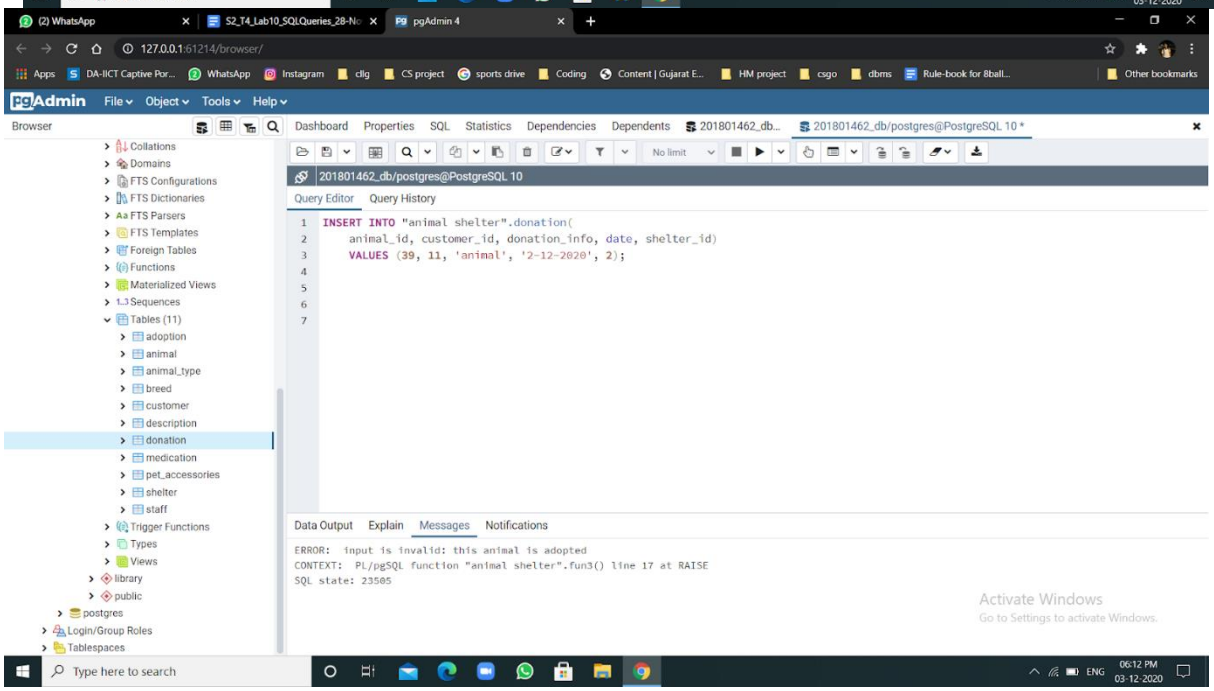


39) A customer wants to donate an animal; return a trigger if the animal id is valid or not.(animal is already in adopted table or donated table)

SQL QUERY:

```
INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (84, 11, 'animal', '2-12-2020', 2);

INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (39, 11, 'animal', '2-12-2020', 2);
```



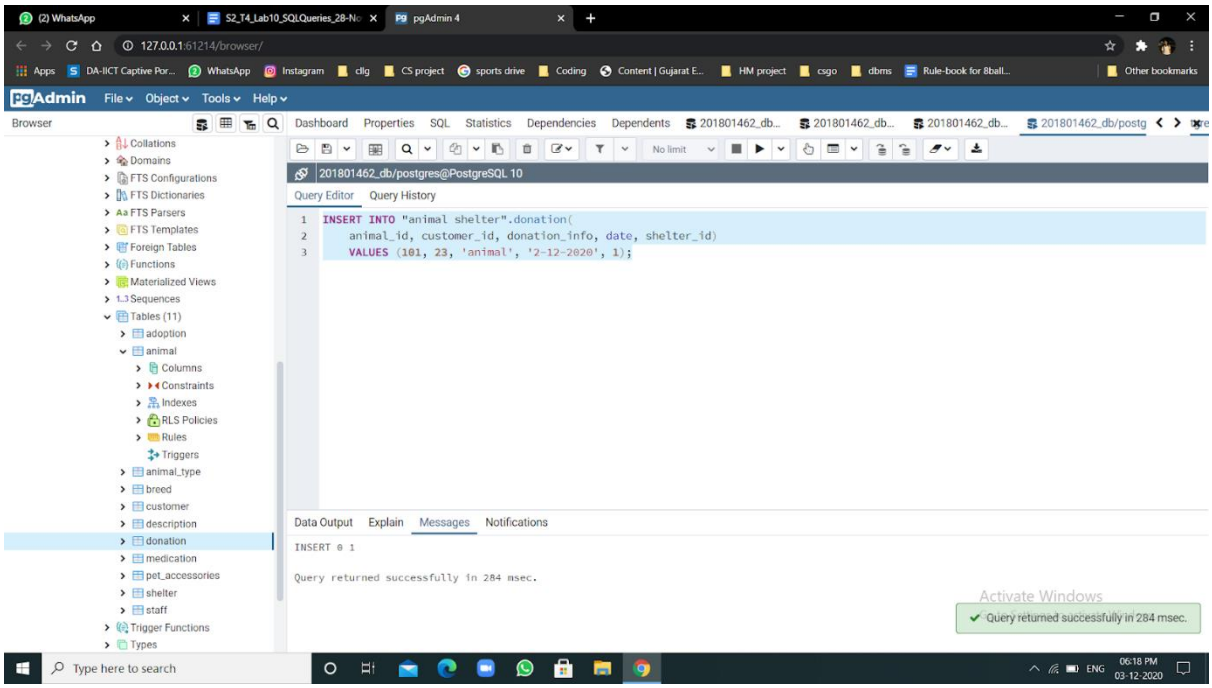


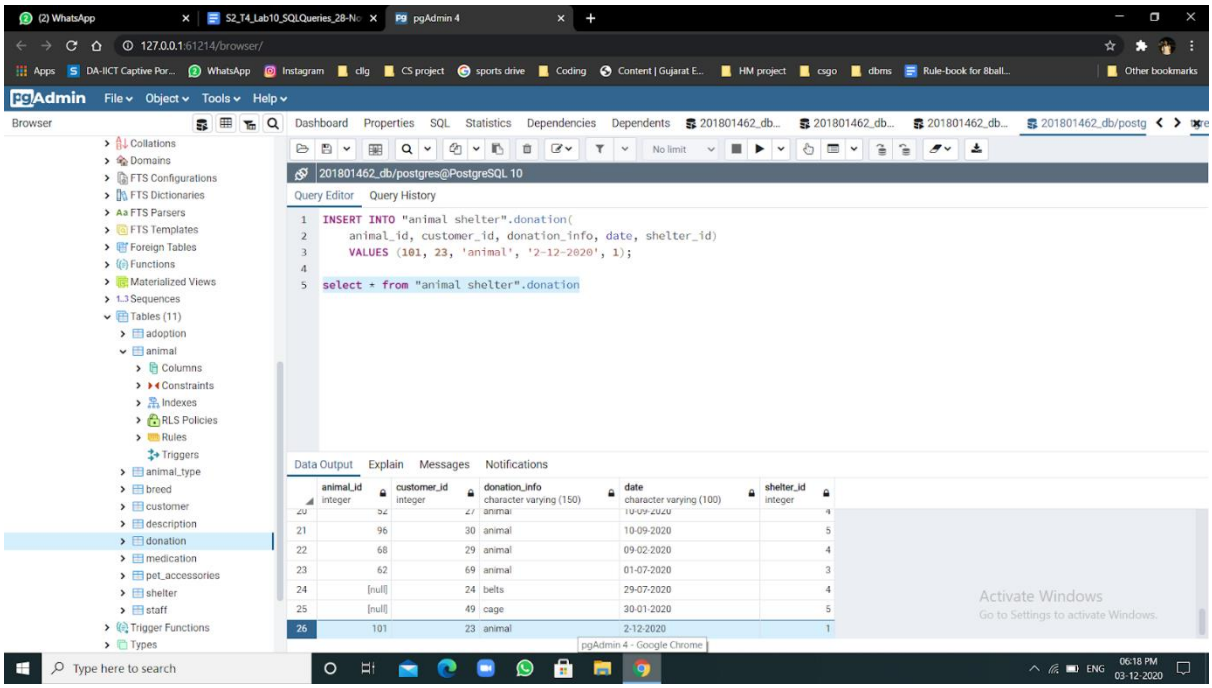
40) A customer wants to donate an animal, insert the tuple if all entries are valid using trigger

//first he/she will insert the animal details then will apply for donation

SQL QUERY:

```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
VALUES (101, 1, 1, 1, 4, 2200, 'Male', 'White', 'M', 12.2, 25);

INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (101, 23, 'animal', '2-12-2020', 1);
```

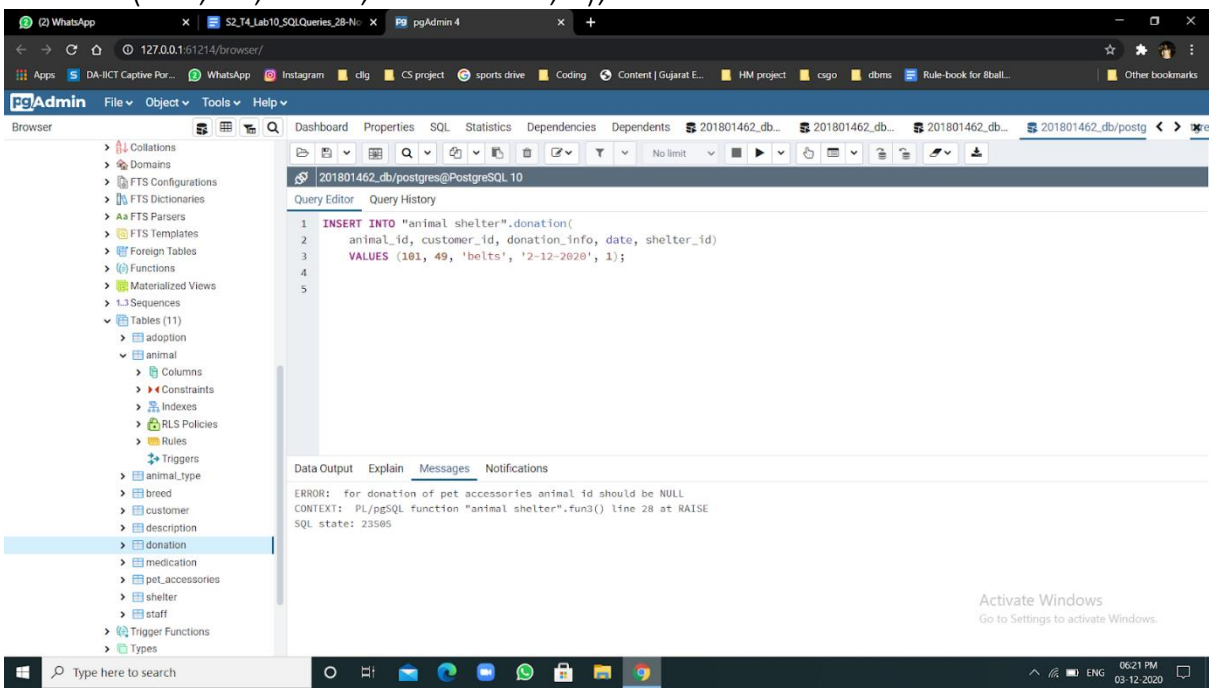




41)A customer wants to donate a pet accessory, he filled the animal id col during application, return a trigger for the same.

SQL:

```
INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (101, 49, 'belts', '2-12-2020', 1);
```

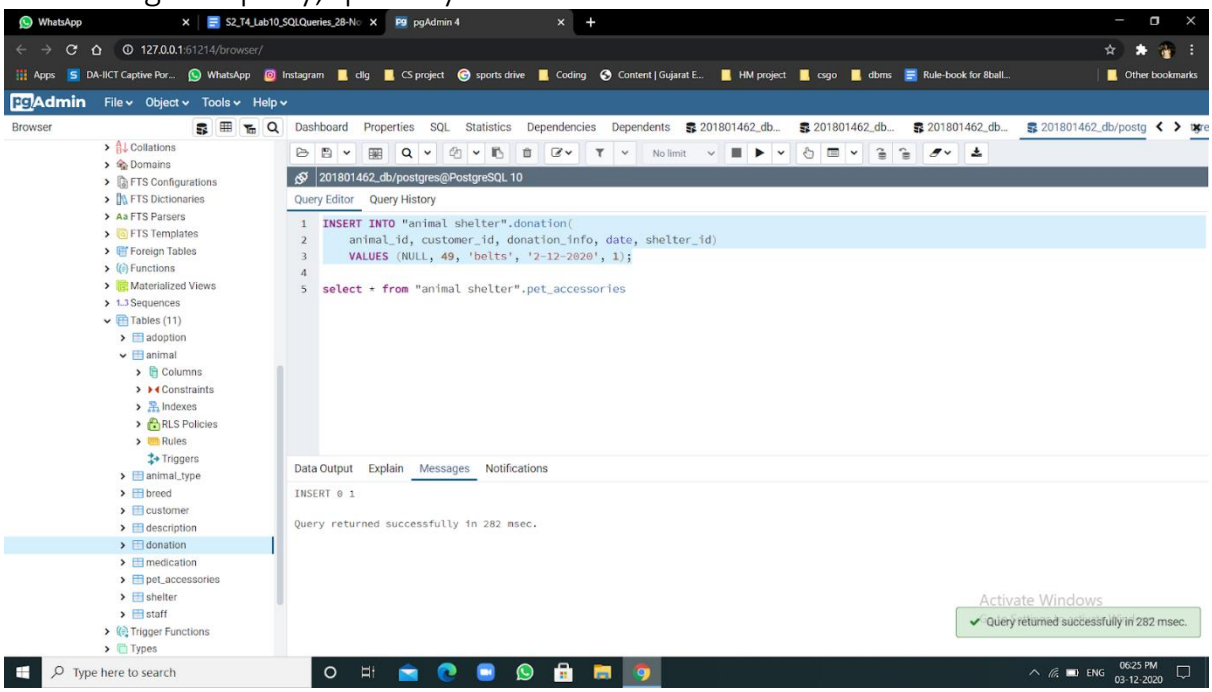


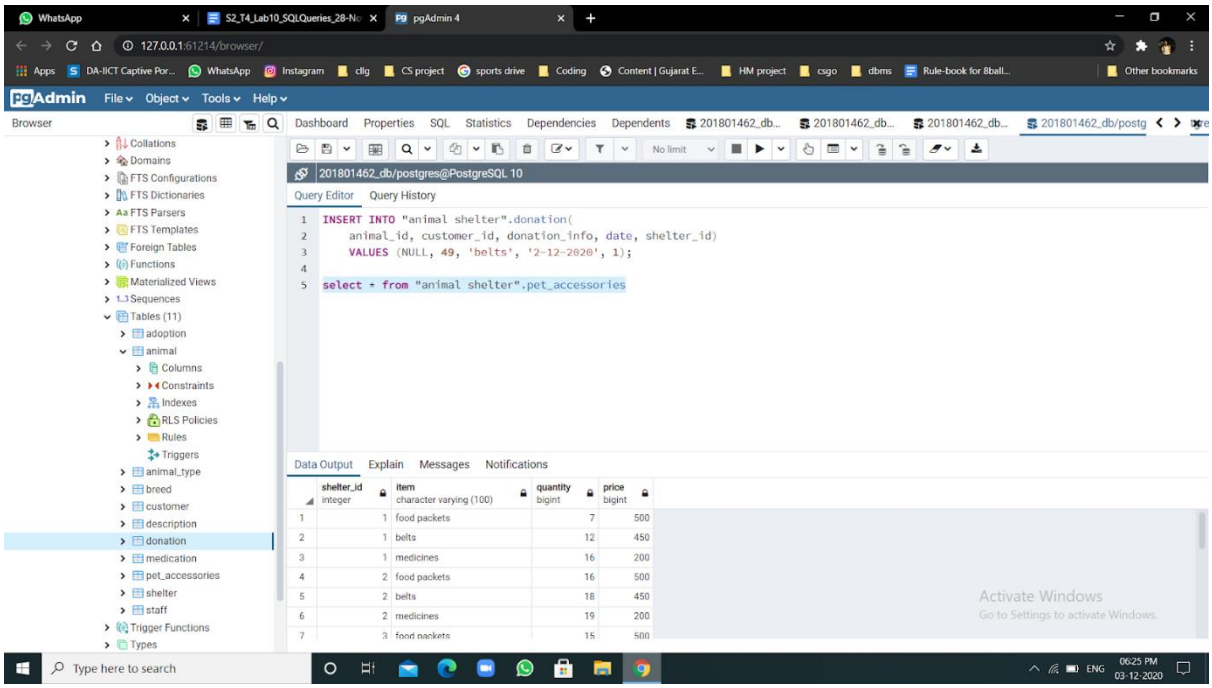
42) A customer wants to donate a pet accessory, write a trigger fun to check for valid details and increment the accessory in the pet accessory table

SQL QUERY:

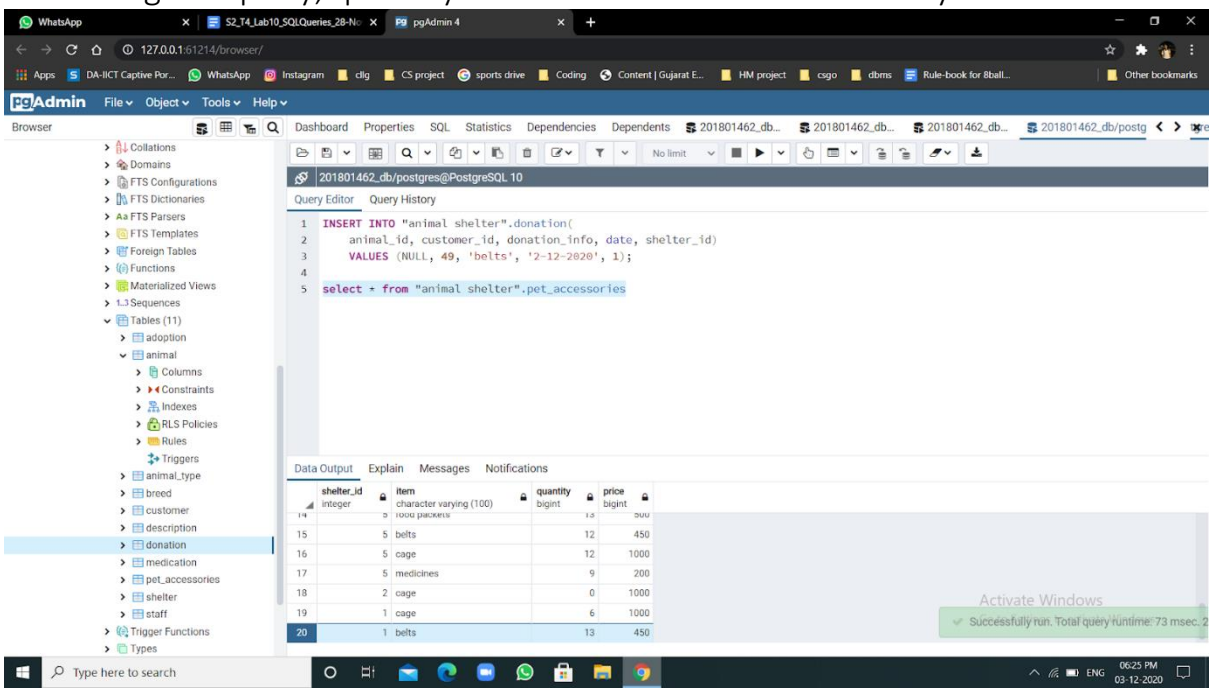
```
INSERT INTO "animal shelter".donation(
animal_id, customer_id, donation_info, date, shelter_id)
VALUES (NULL, 49, 'belts', '2-12-2020', 1);

select * from "animal shelter".pet_accessories
```

//before running the query, quantity of belts in shelter 1





////after running the query, quantity of belts in shelter 1 increased by 1



4) animal trigger

```
set search_path to animal_shelter;
create or replace function fun4()
returns trigger
language plpgsql
as $body$
begin
	if(new.id in (select id from animal_shelter.animal)) then
	raise UNIQUE_VIOLATION using message='input invalid:animal id already
exist';
	end if;

	if(new.type not in (select id from animal_shelter.animal_type)) then
	raise UNIQUE_VIOLATION using message='input invalid:animal type does
not exist';
	end if;

	if(new.shelter_id not in( select id from animal_shelter.shelter)) then
	raise UNIQUE_VIOLATION using message='input invalid:shelter does not
exist';
	end if;

	if(new.type = 1 ) then
		if(new.breed_type <1 or new.breed_type>6 ) then
		raise UNIQUE_VIOLATION using message='input invalid: animal
type and breed are not matching';
		end if;
	end if;

	if(new.type = 2) then
		if(new.breed_type <7 or new.breed_type>11 ) then
		raise UNIQUE_VIOLATION using message='input invalid: animal
type and breed are not matching';
		end if;
	end if;

	if(new.type = 3) then
		if(new.breed_type <12 or new.breed_type>16 ) then
```

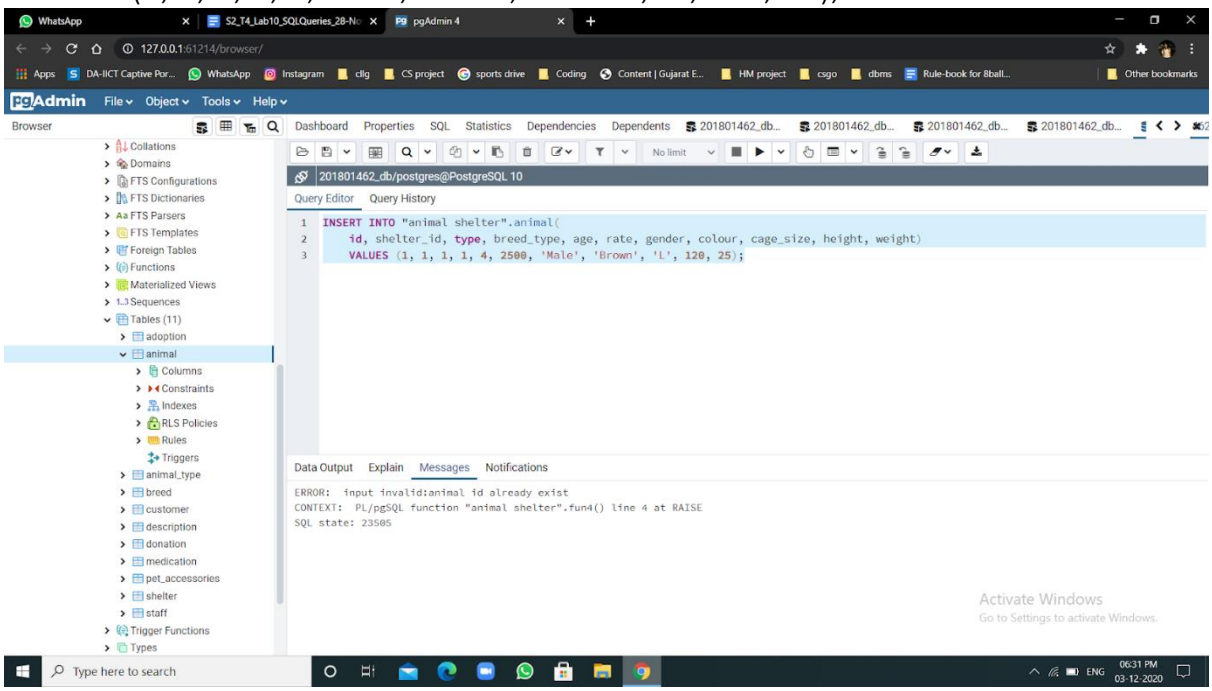
```
                raise UNIQUE_VIOLATION using message='input invalid: animal
type and breed are not matching';
                end if;
        end if;

        if(new.type = 4) then
                if(new.breed_type <17 or new.breed_type>21 ) then
                raise UNIQUE_VIOLATION using message='input invalid: animal
type and breed are not matching';
                end if;
        end if;

        return new;


end
$body$


create trigger t4
before insert
on animal_shelter.animal
for each row
execute procedure fun4();
```

Queries:

43) A new entry of animal is being done, return a trigger if animal id exists in the database

SQL QUERY:

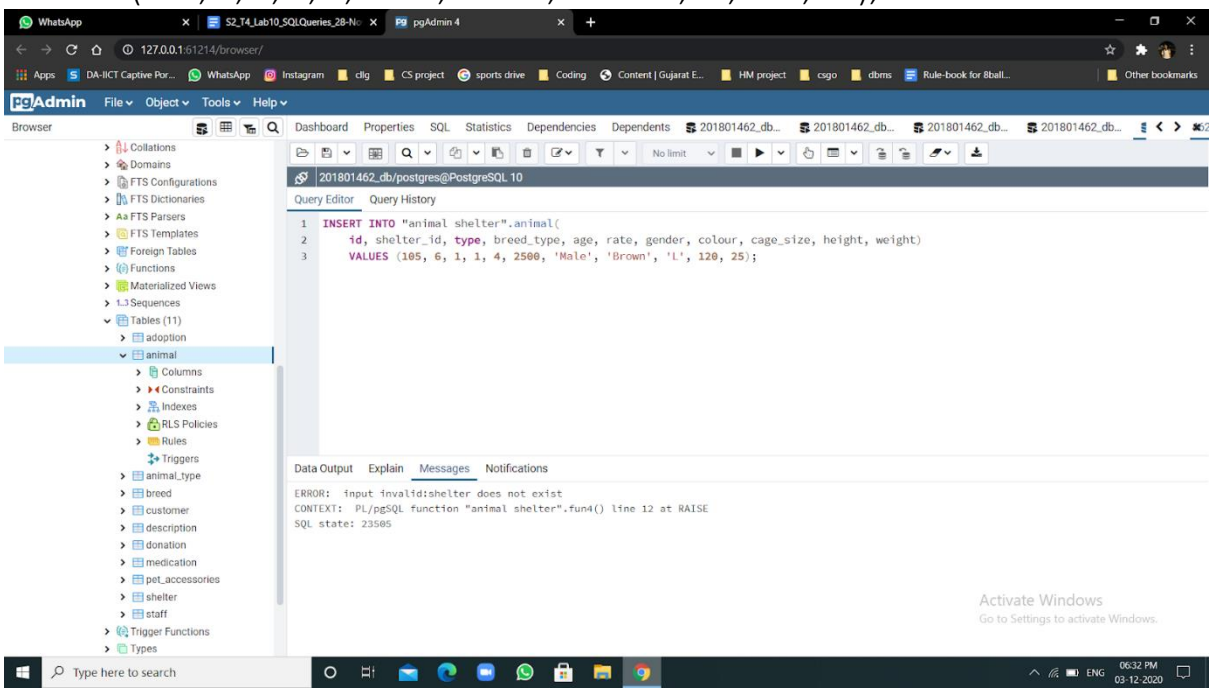
```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
VALUES (1, 1, 1, 1, 4, 2500, 'Male', 'Brown', 'L', 120, 25);
```



44) Insert in animal table, return trigger if input is not valid

SQL QUERY:

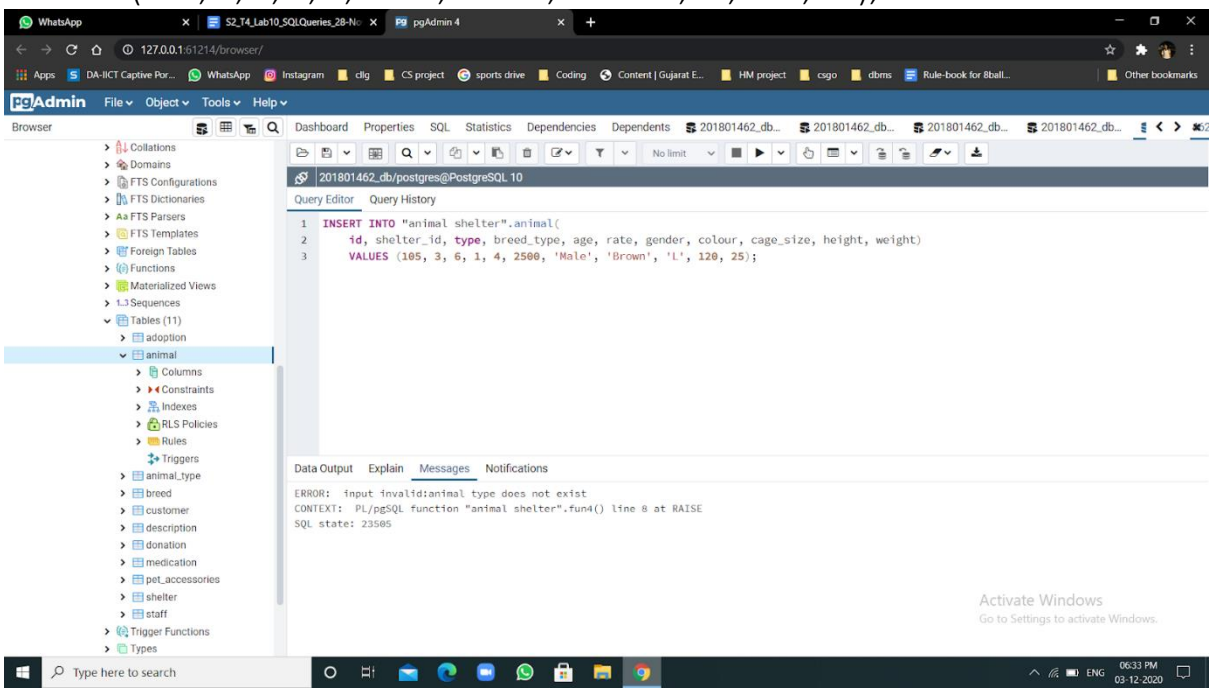
```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
VALUES (105, 6, 1, 1, 4, 2500, 'Male', 'Brown', 'L', 120, 25);
```



45) animal type id out of range

SQL QUERY:

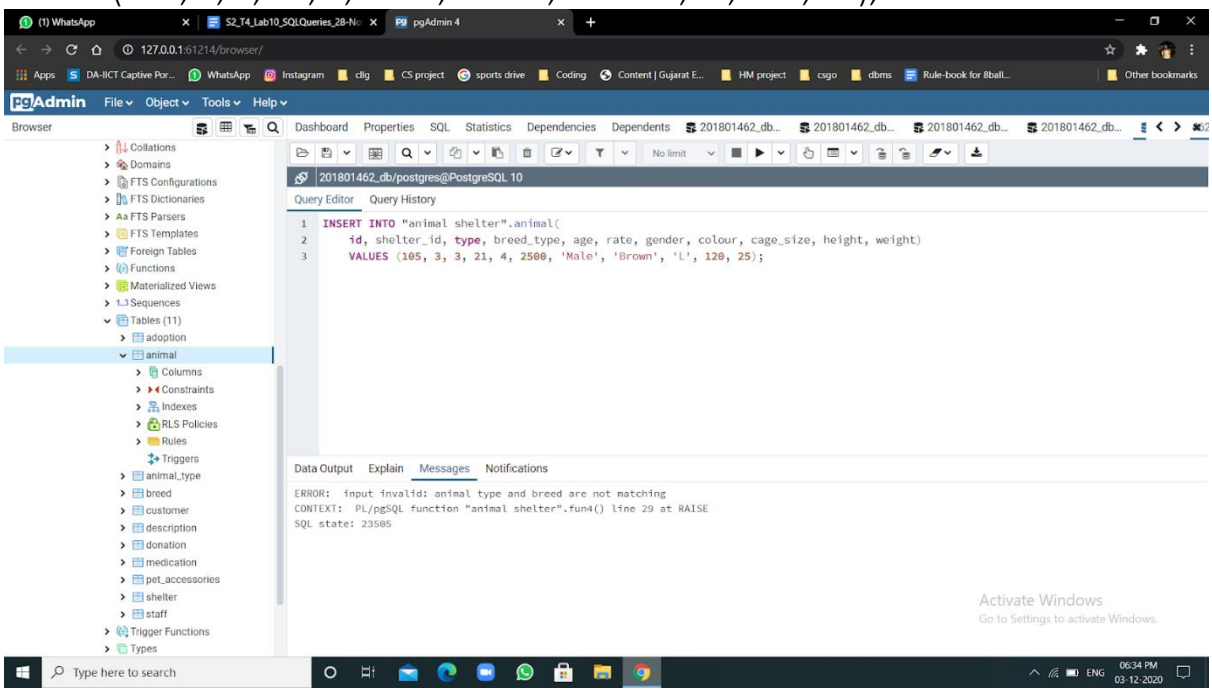
```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size, height, weight)
VALUES (105, 3, 6, 1, 4, 2500, 'Male', 'Brown', 'L', 120, 25);
```



46) animal type id must be in range of breed

SQL QUERY:

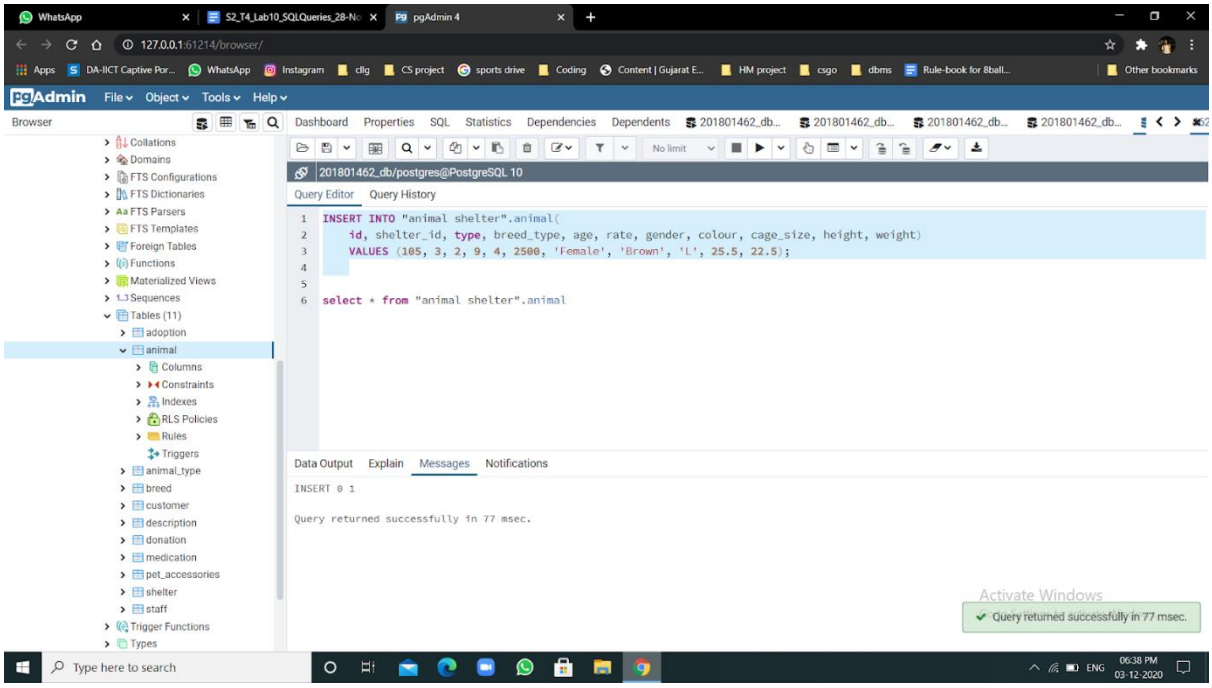
```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
VALUES (105, 3, 3, 21, 4, 2500, 'Male', 'Brown', 'L', 120, 25);
```

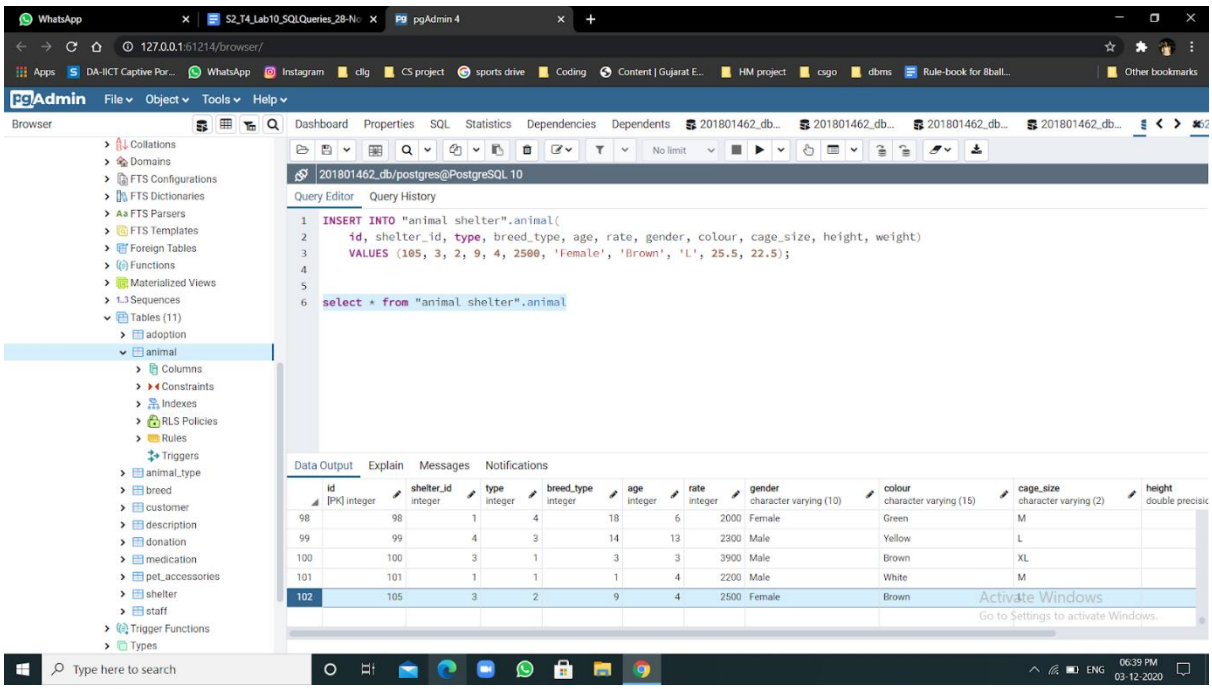


47) Animal is being add, return a trigger and insert in all entries are valid

SQL:

```
INSERT INTO "animal shelter".animal(
id, shelter_id, type, breed_type, age, rate, gender, colour, cage_size,
height, weight)
VALUES (105, 3, 2, 9, 4, 2500, 'Female', 'Brown', 'L', 25.5, 22.5);

select * from "animal shelter".animal
```





48) Print the name of customer with most donations

SQL QUERY:

```
create view  max_donation(id ,count ) as
select customer_id,count(*)
from donation group by customer_id
order by count desc
limit 1

select * from max_donation

select customer.first_name,last_name from customer
where customer.id =
        (select id from max_donation)
```