Adoption Management System

Name	Id	Contribution
Syeda Sarah Ferdous	20-44176-2	ER Diagram,Normalization,Schema Diagram , Table Creation,SQL query, Relational-Algebra, PLSQL(Procedure,Function)
M.O.B. JIHAD	21-44811-1	User Interface,Table Creation , Database Connection , SQL Query, PLSQL(Record,Package)
MD. OLI ULLAH RAFI	20-42934-1	Introduction, project Proposal, Scenario Description, Conclusion,PLSQL(Cursor,Trigger)

Course : Advance Database Management System

Section : E

Faculty name: Juena Ahmed Noshin

Contents

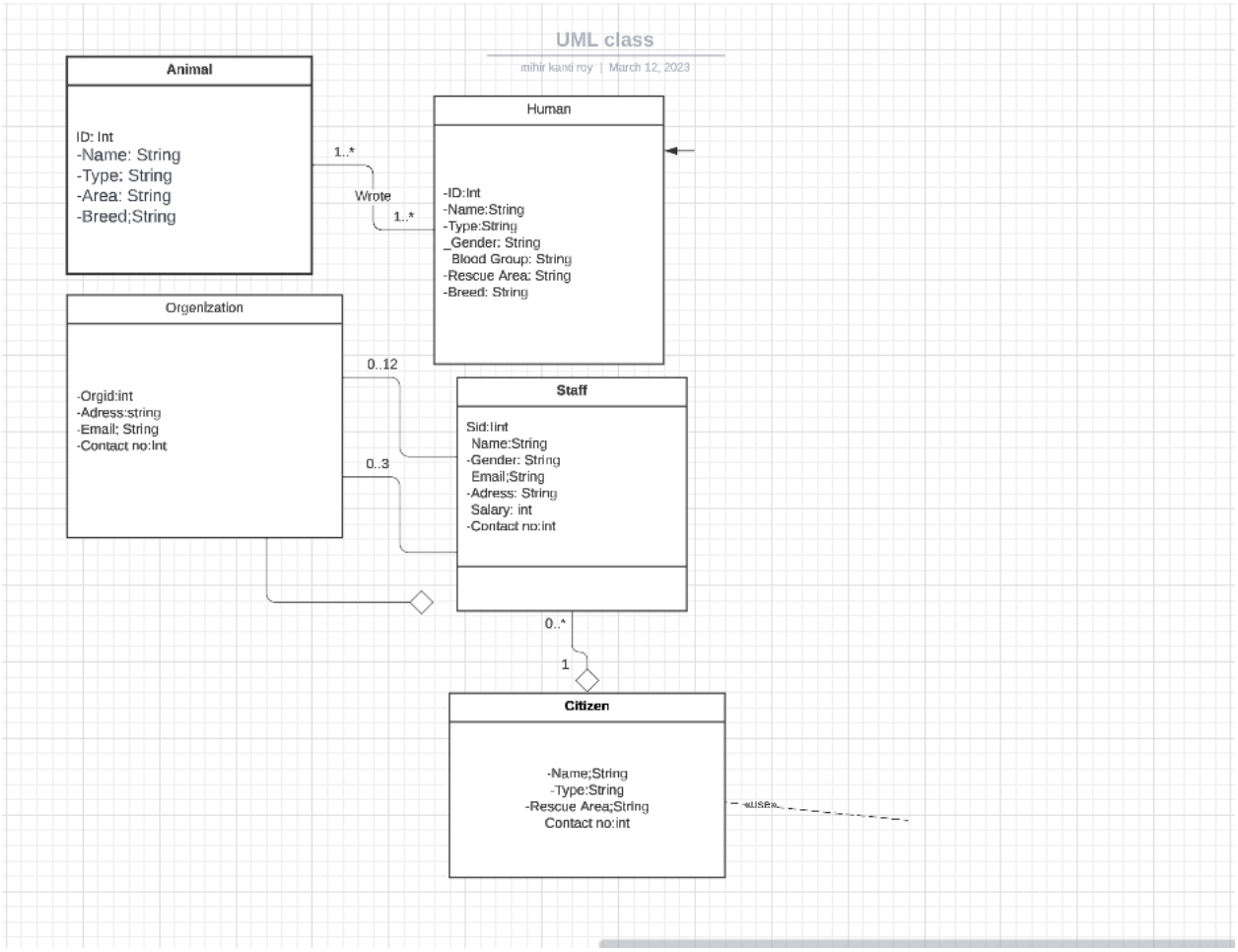
Introduction

Shantir Nir is a non-profit organization dedicated to rescuing and providing a home for dogs, cats, children, and senior citizens who are in need. For anyone who are interested in providing these people and animals with a loving home, it offers adoption services. The organization is open to volunteers and money to further its purpose. Anyone who encounters a homeless child or senior can get in touch with Shantir Nir for help rescuing and caring for them. In the same way, the group is committed to saving and finding homes for stray dogs or cats that are found. For all individuals under their care, Shantir Nir is dedicated to offering the best care and assistance.

Project proposal

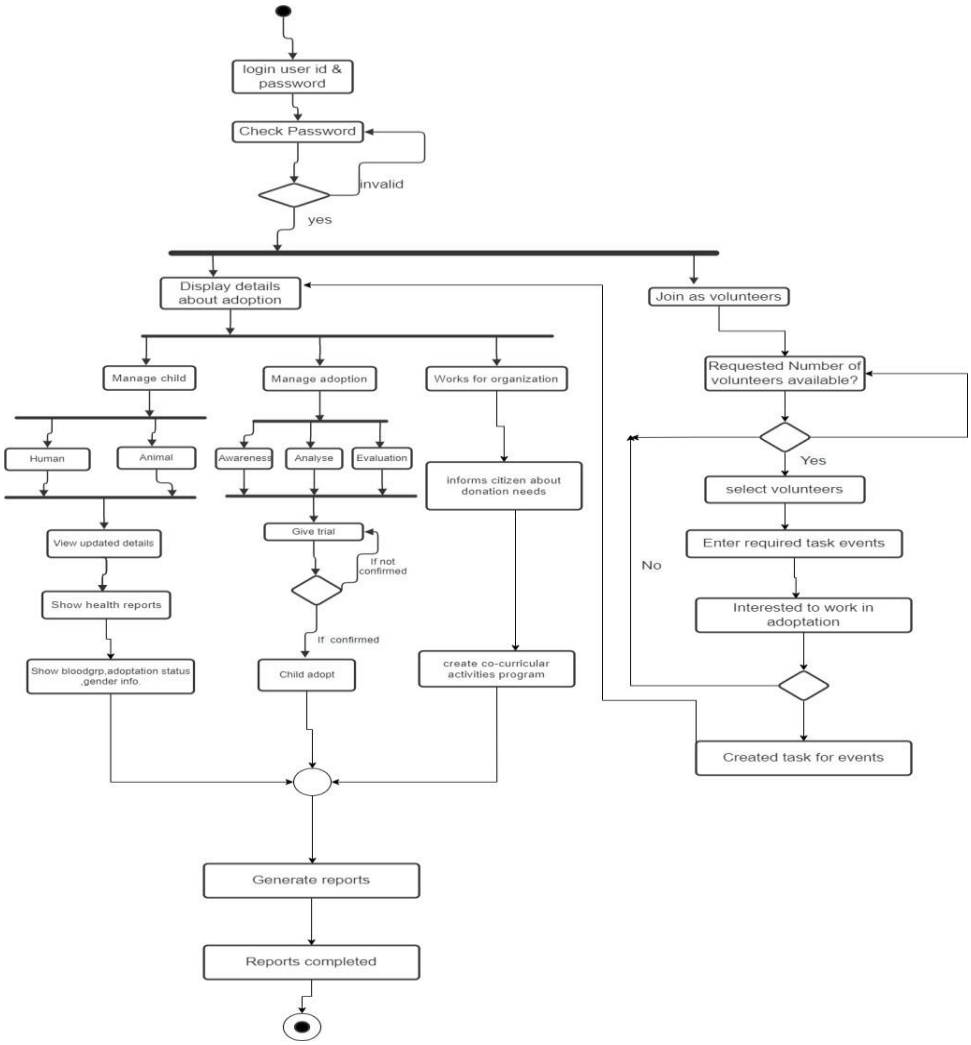
This organization's mission is to rescue and relocate homeless dogs, cats, kids, and senior adults. We want to spread the word about our cause and encourage the adoption of these people and animals. Additionally, we hope to provide volunteers with a chance to support our cause and improve the lives of the people we assist. A group of staff members and volunteers will oversee the operation. The team will be in charge of helping individuals in need, promoting adoption services, and carrying outreach activities to spread the word about our cause. To keep our supporters informed and involved, the team will also oversee donations and volunteer opportunities. The lives of the people that we serve will be significantly impacted by this organization. We will enhance the quality of life and offer a caring atmosphere for dogs, cats, kids, and senior folks by giving them a home. By advertising adoption services, we can spread the word about our cause and inspire others to do the same. By offering volunteer opportunities, we will include the neighborhood and provide people a chance to support our cause. Not just the people we help but also our volunteers and supporters will gain from this endeavor.

Class Diagram

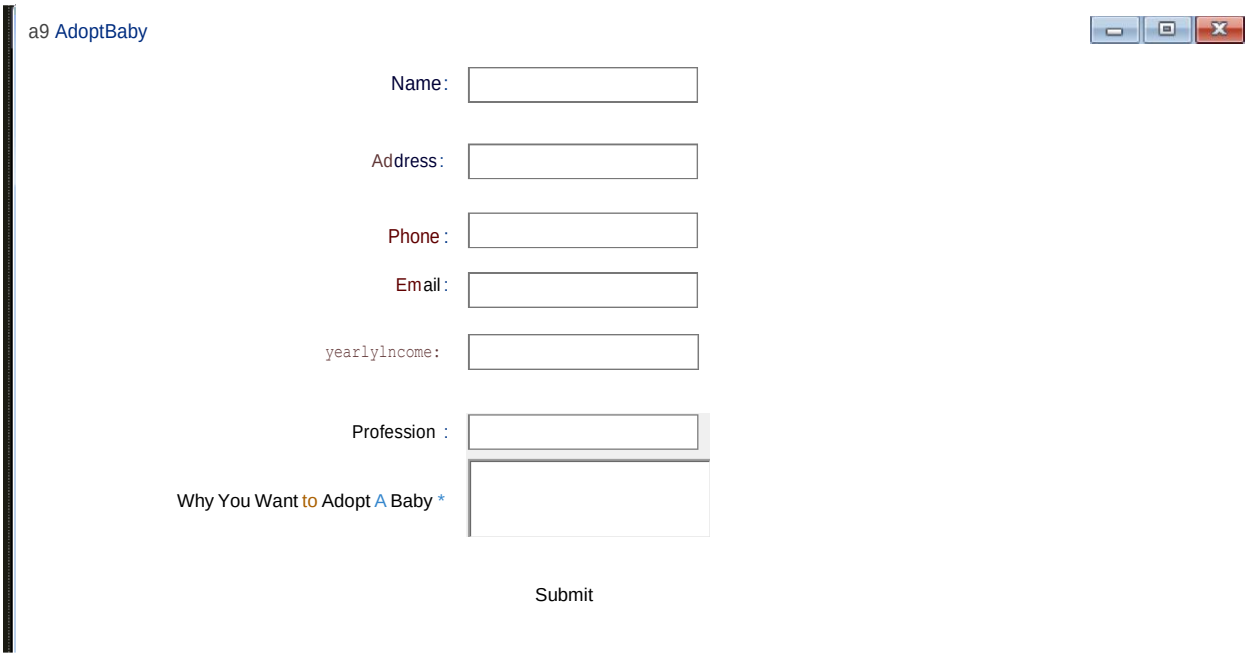


Use case diagram

Activity Diagram



User Interface



Animal Type :

bread :

Age:

Health Condition:

Cell Number :

last Checkup Date: Sunday , March 12, 2023

Associate Staff :

nD Abmn

Card Holder Name :

Card Number :

Expiry Date:

Amount !

a9 HumanEntry

Name :

Gender

Age

Type :

Rescued Area :

Assigned House :

Blood group :

Adoption State '

Insert

Animal Type :

Contact person's Name !

Contact person's Phone :

Address :

Submit

person Type :

Contact person's Name :

Contact person's phone :

Address :

SubmC

Nam e !

Gender :

Father's Name :

Nid/passport/Birth Certificate:

Phone:

Age:

Email :

Staff Category :

Salary Grade :

Address :

z0 VolunteerApp

Name :

Gender :

Father's Name :

Nid/passport/Birth Certificate :

IN StltUtl ON :

Phone

Age :

Email :

Any p revious Work Experience ?

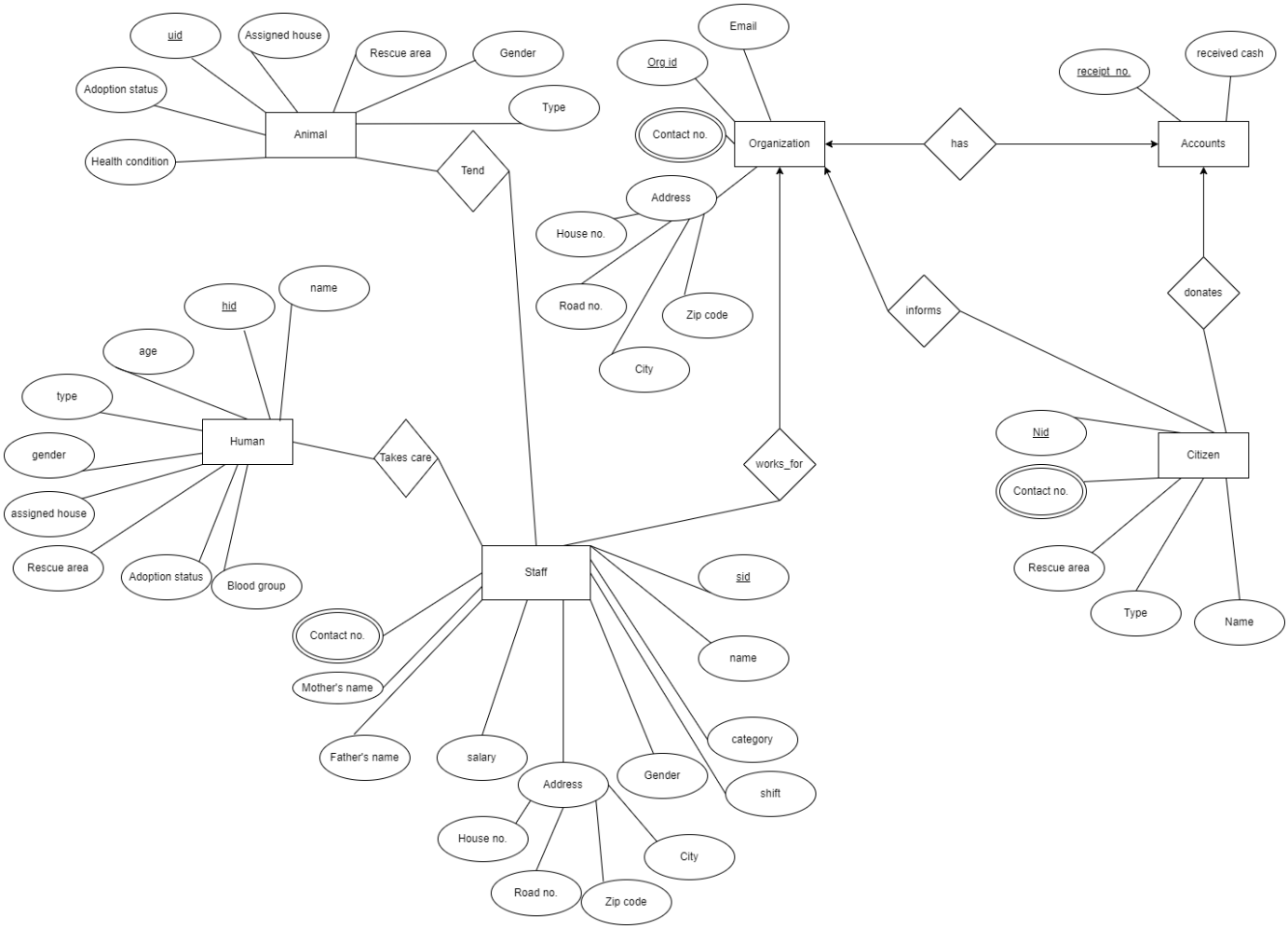
Why you want to join US ?

Apply Now

Scenario Description

In an adoption management system, an animal can be identified by it's unique id, type, gender, rescue area, health condition and assigned house. Multiples animals will be taken care by multiple stuffs. Each stuff will be provided a unique id. Staff data such as name, mother's name, father's name, gender, category, shift, salary, and address are recorded in the system. The address is retrieved by zip code, road no., house no. and city. Each member of staff may have multiple contact numbers. Staff take care of humans too. There is a type of human which implies whether it is a child, infant or senior citizen. There can be multiple humans in the organization, all of them are assigned a unique id. Each human's data such as name, age, gender, type, assignedhouse, rescue_area, adoption status is stored in the system. There can be multiple staff, but the staff only work for one organization. The organization will have a unique organization id. Also, it will have an email address, address, and contact number. An organization may have multiple contact numbers. The address is retrieved by zip code, road no., house no. and city. An organization will have only one accounts section which will take care of all the transactions and donations. Accounts will have data such as receipt no. and total received cash stored in the system. Citizens can directly contact organizations to inform them about a child/animal rescue. Each citizen will be identified by their NID no. and other data such as name, contact no., rescue area information (about the animal/human), type(child/human). Citizens can also donate directly to accounts. Citizens may have multiple contact numbers.

ER-Diagram:



Google drive link (For better quality):

https://drive.google.com/file/d/1DQVGOthSDHq92kcAtWZcLO0gPjiDnptX/view?usp=sharing

Normalization:

Tend (uid, type, breed, gender, color, assigned house, rescue area, health condition, adoption status, sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category)

1NF-> Contact no. is a multivalued attribute.

2NF -> uid, type, breed, gender, color, assigned house, rescue area, health condition, adoption status

sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category

3NF-> uid, type, breed, gender, color, assigned house, rescue area, health condition, adoption status

sid, name, father's name, mother's name, contact no., email, road no., house no., age, gender, salary, shift, category

zip code, city

Tables from **Tend**:

1. uid, type, breed, gender, color, assigned house, rescue area, health condition, adoption status
2. sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift, category, zid
3. zid, zip code, city
4. sid, contact no.
5. uid, sid, a_id

Takes care (hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status, sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category)

1NF-> Contact no. is a multivalued attribute.

2NF-> hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status

sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category

3NF-> hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status

sid, name, father's name, mother's name, contact no., email, road no., house no., age, gender, salary, shift, category

zip code, city

Tables from **Takes care**:

1. hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status
2. sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift, category, zid
3. zid, zip code, city
4. sid, contact no.
5. hid, sid, b_id

Works (sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category, org id, contact no., email, zip code, road no., house no., city)

1NF-> Contact no. is a multivalued attribute.

2NF-> sid, name, father's name, mother's name, contact no., email, zip code, road no., house no., city, age, gender, salary, shift, category

org id, contact no., email, zip code, road no., house no., city

3NF-> sid, name, father's name, mother's name, contact no., email, road no., house no., age, gender, salary, shift, category

zip code, city

org id, contact no., email, road no., house no.

zip code, city

Tables from **Works**:

1. sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift, category ,zid, org id
2. zid, zip code, city
3. sid, contact no.
4. org id, email, road no., house no, zid
5. zid, zip code, city
6. org id ,contact no.

Inform (org id, contact no., email, zip code, road no., house no., city, nid, name, type, contact no. rescue area)

1NF-> Contact no. is a multivalued attribute.

2NF-> org id, contact no., email, zip code, road no., house no., city

nid, name, type, contact no. rescue area

3NF-> org id, contact no., email, road no., house no.

zip code, city

nid, name, type, contact no. rescue area

Tables from **Inform:**

1. org id, email, road no., house no., zid
2. org id, contact no.
3. zid, zip code, city
4. nid, name, type, rescue area, org id
5. nid, contact no.

Has (org id, contact no., email, zip code, road no., house no., city, receipt no., received_cash)

1NF-> Contact no. is a multivalued attribute.

2NF-> org id, contact no., email, zip code, road no., house no., city

receipt no., received_cash

3NF-> org id, contact no., email, road no., house no.

zip code, city

receipt no., received_cash

Tables from **Has:**

1. org id, email, road no., house no., zid , receipt no
2. zid, zip code, city
3. org id, contact no.
4. receipt no., received_cash

Donates (nid, name, type, contact no. rescue area, receipt no., received_cash)

1NF-> Contact no. is a multivalued attribute.

2NF-> nid, name, type, contact no. rescue area

receipt no., received_cash

3NF-> nid, name, type, contact no. rescue area

receipt no., received_cash

Tables from **Informs**:

1. nid, name, type. rescue area, receipt no.
2. nid, contact no.
3. receipt no., received_cash

Tables after Normalization:

1. uid, type, breed, gender, color, assigned house, rescue area, health condition, adoption status- - > **Animal**
2. ~~sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift,~~ category, ~~zid~~
3. zid, zip code, city-- -> **Address**
4. sid, contact no ------> **Staff_contact**
5. uid, sid, a_id---------> **Info1**
6. hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status------> **Human**
7. ~~sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift~~, ~~category, zid~~
8. ~~zid, zip code, city~~
9. ~~sid, contact no.~~
10. hid, sid, b_id----- > **Info2**
11. sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift, category ,zid, org id -----> **Staff**
12. zid, zip code, city
13. ~~sid, contact no.~~
14. ~~org id, email, road no., house no, zid~~
15. ~~zid, zip code, city~~
16. org id ,contact no --------> **Organization_Contact**
17. ~~org id, email, road no., house no, zid~~
18. ~~org id, contact no.~~
19. zid, zip code, city
20. nid, name, type, rescue area, org id------ > **Rescue**
21. nid, contact no ------ > **citizen_contact**
22. org id, email, road no., house no., zid, receipt no ------> **Organization**
23. ~~zid, zip code, city~~
24. ~~org id, contact no.~~
25. receipt no., received_cash------- > **Accounts**
26. nid, name, type, rescue area, receipt no.---------> **Donation**
27. ~~nid, contact no.~~
28. ~~receipt no., received_cash~~

Final Tables:

1. uidofhuman, type, breed, gender, color, assigned_house, rescue _area, health_condition, adoption_st ------ **Animal**
2. zid, zip code, city--- **Address**
3. sid, contact no ------**Staff_contact**
4. uid, sid, a_id ---------**Info1**
5. hid, name, age, gender, type, blood group, assigned house, rescue area, adoption status---- **Human**
6. hid, sid, b_id **------Info2**
7. sid, name, father's name, mother's name, email, road no., house no., age, gender, salary, shift, category ,zid, org id ---- **Staff**
8. org id ,contact no ---------- **Organization_Contact**
9. nid, name, type, rescue area, org id--- **Rescue**
10. nid, contact no ---------- **citizen_contact**
11. org id, email, road no., house no., zid, receipt no. –**Organization**
12. receipt no., received_cash ----------**Accounts**
13. nid, name, type, rescue area, receipt no. ----------**Donation**

Schema diagram

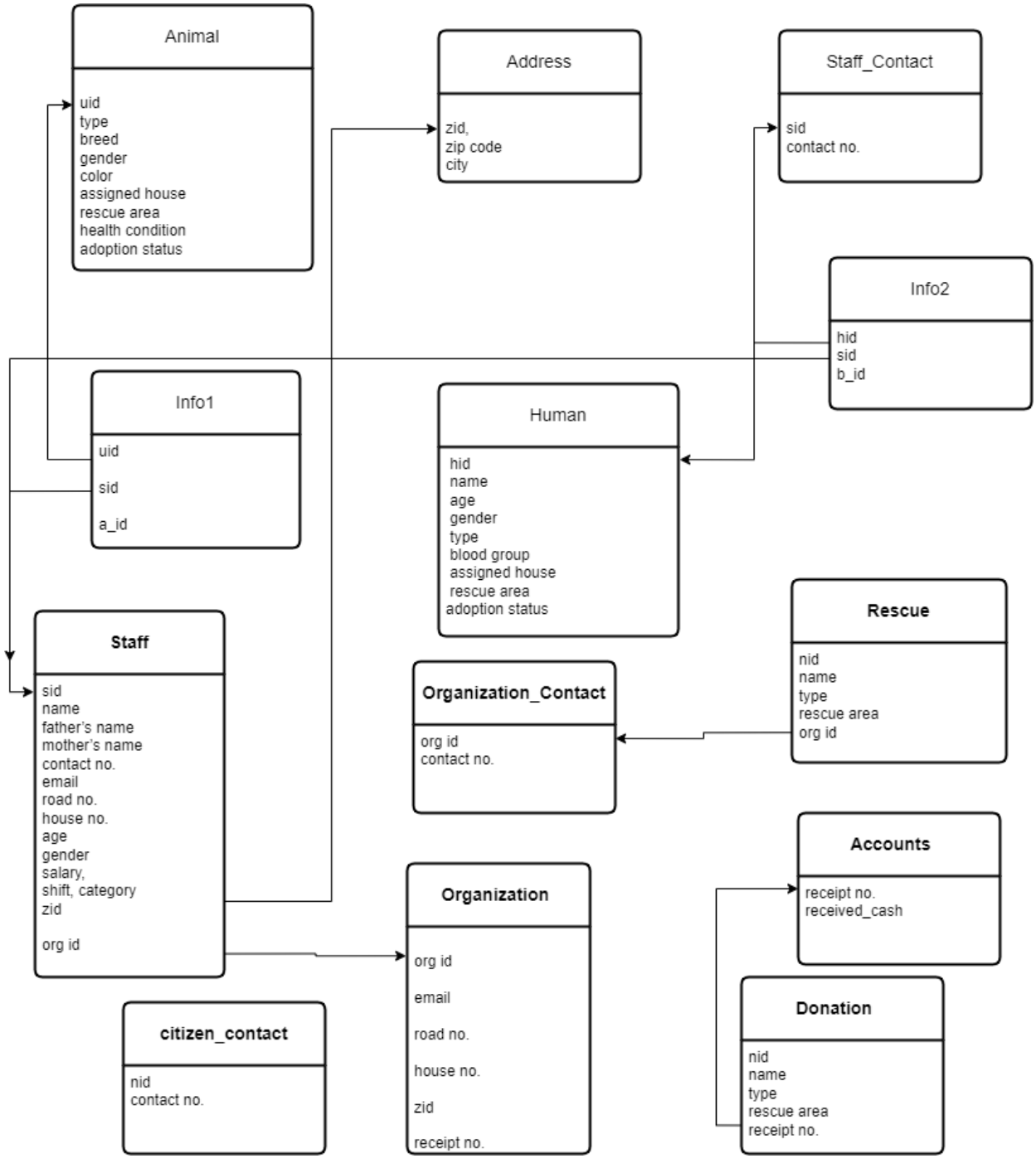


Table creation:

Animal

create table animal(

uidofAnimal Number NOT NULL PRIMARY KEY,

type VARCHAR2(20),

Gender VARCHAR2(20),

AssignedHouse VARCHAR2(20),

rescue_Area VARCHAR2(20),

Health_Condition VARCHAR2(20),

Adoption_St VARCHAR2(20))

Results Explain Describe Saved SQL History

Object Type TABLE Object ANIMAL

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Defaul
ANIMAL	UIDOFANIMAL	Number	-	-	-	1	-	-
	TYPE	Varchar2	20	-	-	-	✓	-
	GENDER	Varchar2	20	-	-	-	✓	-
	ASSIGNEDHOUSE	Varchar2	20	-	-	-	✓	-
	RESCUE_AREA	Varchar2	20	-	-	-	✓	-
	HEALTH_CONDITION	Varchar2	20	-	-	-	✓	-
	ADOPTION_ST	Varchar2	20	-	-	-	✓	-

Address

Create Table Address(

zid Number NOT NULL PRIMARY KEY,

zip_code Number,

city Varchar2(30))

Results Explain Describe Saved SQL History

Object Type TABLE Object ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Co
ADDRESS	ZID	Number	-	-	-	1	-	-	-
	ZIP_CODE	Number	-	-	-	-	✓	-	-
	CITY	Varchar2	30	-	-	-	✓	-	-

1 - 3

Staff_Contact

Create Table Staff_Contact(

sid number,

contact_no varchar2(20))

alter table Staff_Contact add constraint s_pk primary key(sid, contact_no)

Object Type TABLE Object STAFF_CONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Def
STAFF_CONTACT	SID	Number	-	-	-	1	-	-
	CONTACT_NO	Varchar2	20	-	-	2	-	-

Info1

Create Table info1(

uidofanimal number,

sid number,

a_id number not null primary key,

Foreign Key(uidofanimal) references Animal(uidofanimal),

Foreign Key(sid) references staff_Contact(sid))

```
desc info1
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **INFO1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
INFO1	UIDOFANIMAL	Number	-	-	-	-	✓	-	-
	SID	Number	-	-	-	-	✓	-	-
	A_ID	Number	-	-	-	1	-	-	-

1 - 3

Language: en-us

Human

Create table Human(

uidofHuman number(10) not null primary key,

Name varchar2 (40),

Age number(10),

Gender varchar2(20),

type VARCHAR2(20),

AssignedHouse VARCHAR2(20),

rescue_Area VARCHAR2(20),

Adoption_St VARCHAR2(20));

Object Type **TABLE** Object **HUMAN**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default
HUMAN	UIDOFHUMAN	Number	-	10	0	1	-	-
	NAME	Varchar2	40	-	-	-	✓	-
	AGE	Number	-	10	0	-	✓	-
	GENDER	Varchar2	20	-	-	-	✓	-
	TYPE	Varchar2	20	-	-	-	✓	-
	ASSIGNEDHOUSE	Varchar2	20	-	-	-	✓	-
	RESCUE_AREA	Varchar2	20	-	-	-	✓	-
	ADOPTION_ST	Varchar2	20	-	-	-	✓	-

1 -

Application Express 2.1.0.00.39

Info2

Create Table info2(

b_id number not null primary key,

uidofhuman number,

sid number,

Foreign Key(uidofhuman) references Human(uidofhuman),

Foreign Key(sid) references staff_Contact(sid)

)

Accounts

Create table Accounts(

Receipt_no varchar2(20) not null primary key,

Received_cash int);

Object Type TABLE Object ACCOUNTS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Defa
ACCOUNTS	RECEIPT_NO	Varchar2	20	-	-	1	-	-
	RECEIVED_CASH	Number	-	-	0	-	✓	-

Donation

Create table Donation(

Receipt_no varchar2(20),

donation_id varchar2(30) not null primary key,

Name varchar2(40),

Type varchar2(20),

rescue_Area varchar2(20),

FOREIGN KEY (Receipt_no) REFERENCES Accounts(Receipt_no));

Object Type TABLE Object DONATION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default
DONATION	RECEIPT_NO	Varchar2	20	-	-	-	✓	-
	DONATION_ID	Varchar2	30	-	-	1	-	-
	NAME	Varchar2	40	-	-	-	✓	-
	TYPE	Varchar2	20	-	-	-	✓	-
	RESCUE_AREA	Varchar2	20	-	-	-	✓	-

Organization:

Create table Organization(

zid number,

Receipt_no varchar2(20),

Org_id varchar2(30) not null primary key,

Email varchar2(20),

Road_no varchar2(20),

House_no varchar2(20),

FOREIGN KEY (zid) REFERENCES Address(zid),

FOREIGN KEY (receipt_no) REFERENCES Accounts(receipt_no));

Object Type TABLE Object ORGANIZATION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Defau
ORGANIZATION	ZID	Number	-	-	-	-	✓	-
	RECEIPT_NO	Varchar2	20	-	-	-	✓	-
	ORG_ID	Varchar2	30	-	-	1	-	-
	EMAIL	Varchar2	20	-	-	-	✓	-
	ROAD_NO	Varchar2	20	-	-	-	✓	-
	HOUSE_NO	Varchar2	20	-	-	-	✓	-

Staff

Create table Staff(

zid number,

org_id varchar2(30),

sid number (10),

Name varchar2 (40),

Fathers_name varchar2(40),

Mothers_name varchar2(40),

Email varchar2(20),

Road_no varchar2(20),

House_no varchar2(20),

Age number(10),

Gender varchar2(20),

Salary int,

Shift varchar2(20),

FOREIGN KEY (zid) REFERENCES Address(zid),

FOREIGN KEY (org_id) REFERENCES Organization(org_id));

Object Type TABLE Object STAFF

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default
STAFF	ZID	Number	-	-	-	-	✓	-
	ORG_ID	Varchar2	30	-	-	-	✓	-
	SID	Number	-	10	0	-	✓	-
	NAME	Varchar2	40	-	-	-	✓	-
	FATHERS_NAME	Varchar2	40	-	-	-	✓	-
	MOTHERS_NAME	Varchar2	40	-	-	-	✓	-
	EMAIL	Varchar2	20	-	-	-	✓	-

Organization_Contact:

Create table Organization_Contact(
org_id varchar2(30),
contact_no varchar2(20));
alter table Organization_Contact add constraint oc_pk primary key(org_id, contact_no)

Object Type TABLE Object ORGANIZATION_CONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullabl
ORGANIZATION_CONTACT	ORG_ID	Varchar2	30	-	-	1	-
	CONTACT_NO	Varchar2	20	-	-	2	-

Rescue

Create table Rescue(
org_id varchar2(30),
Nid varchar2(30) not null primary key,
Name varchar2(40),
Type varchar2(20),
Rescue_area varchar2(20),
FOREIGN KEY (org_id) REFERENCES Organization(org_id));

Object Type TABLE Object RESCUE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default
RESCUE	ORG_ID	Varchar2	30	-	-	-	✓	-
	NID	Varchar2	30	-	-	1	-	-
	NAME	Varchar2	40	-	-	-	✓	-
	TYPE	Varchar2	20	-	-	-	✓	-
	RESCUE_AREA	Varchar2	20	-	-	-	✓	-

1

Citizen_Contact

Create table Citizen_Contact(
Nid varchar2(30),
contact_no varchar2(20));
alter table Citizen_Contact add constraint c_pk primary key(nid,contact_no);

Object Type TABLE Object CITIZEN_CONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	D
CITIZEN_CONTACT	NID	Varchar2	30	-	-	1	-	-
	CONTACT_NO	Varchar2	20	-	-	2	-	-

Sequences

Animal

create sequence Animal_UID

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Address

create sequence Address_zID

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Staff_Contact

create Sequence StaffC_SID

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Info1

create Sequence Info1_aid

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Human

create Sequence human_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Info2

create Sequence Info2_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Staff

create Sequence staff_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Organization_contact

create Sequence org_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Rescue

create Sequence res_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Citizen_contact

create Sequence citcon_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Organization

create Sequence o_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Accounts

create Sequence a_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

Donation

create Sequence do_seq

INCREMENT BY 1

START WITH 1

MAXVALUE 10000000

HUMAN_SEQ	1	999999999999999999999999999	1	N
INFO2_SEQ	1	10000000	1	N
STAFF_SEQ	1	10000000	1	N
ORG_SEQ	1	10000000	1	N
RES_SEQ	1	10000000	1	N
CITCON_SEQ	1	10000000	1	N
O_SEQ	1	10000000	1	N
A_SEQ	1	10000000	1	N
DO_SEQ	1	10000000	1	N
ANIMAL_UID	1	10000000	1	N
ADDRESS_ZID	1	10000000	1	N
STAFFC_SID	1	10000000	1	N
INFO1_AID	1	10000000	1	N

33 rows returned in 0.00 seconds CSV Export

Index

Animal

CREATE INDEX AnimalShortSummary

ON animal(uidofAnimal,AssignedHouse,Adoption_St);

Address

CREATE INDEX address_zAndCity

ON address(zip_code,city);

Staff_contact

CREATE INDEX staffC

ON staff_contact(sid,contact_no);

Info1

CREATE INDEX info1data

ON info1(uidofanimal,sid,a_id);

Human

CREATE INDEX humandata

ON Human(uidofhuman, type, AssignedHouse);

Info2

CREATE INDEX info2data

ON info2(uidofhuman, b_id);

Staff

CREATE INDEX staffdata

ON staff(sid , category);

Organization_Contact

CREATE INDEX oc_index

ON organization_contact(contact_no);

Rescue

CREATE INDEX rescue_alert

ON rescue(rescue_area,

type); **Citizen_contact**

CREATE INDEX ct_index

ON Citizen_contact (contact_no);

Organization

CREATE INDEX ot_index

ON Organization(email, receipt_no);

Accounts

CREATE INDEX a_index

ON accounts(received_cash);

Donation

CREATE INDEX don_index

ON donation(receipt_no, type);

Data Insertion

#Animal

```
Insert INTO Animal(uidofanimal,
type,gender,assignedhouse,rescue_area,health_condition,adoption_st) values
(Animal_UID.NEXTVAL,'Cat','Male','C3','Dhaka','Healthy','Available');
Insert INTO Animal(uidofanimal,
type,gender,assignedhouse,rescue_area,health_condition,adoption_st) values
(Animal_UID.NEXTVAL,'Dog','Male','A2','Dhaka','Good','Available');
Insert INTO Animal(uidofanimal,
type,gender,assignedhouse,rescue_area,health_condition,adoption_st) values
(Animal_UID.NEXTVAL,'Cat','Female','A3','Dhaka','Average','Available');
Insert INTO Animal(uidofanimal,
type,gender,assignedhouse,rescue_area,health_condition,adoption_st) values
(Animal_UID.NEXTVAL,'Cat','Female','A4','Dhaka','Healthy','Available');
;
Insert INTO Animal(uidofanimal,
type,gender,assignedhouse,rescue_area,health_condition,adoption_st) values
(Animal_UID.NEXTVAL,'Dog','Female','A5','Dhaka','Healthy','Adopted');
```

#Address

```
Insert INTO Address(zid,zip code,city) values (Address_zID.NEXTVAL,'301','Dhaka'); Insert
INTO Address(zid,zip code,city) values (Address_zID.NEXTVAL,'302','Chattogram'); Insert
INTO Address(zid,zip code,city) values (Address_zID.NEXTVAL,'303','Khulna'); Insert INTO
Address(zid,zip code,city) values (Address_zID.NEXTVAL,'304','Sylhet'); Insert INTO
Address(zid,zip code,city) values (Address_zID.NEXTVAL,'305','Khulna');
```

#StaffContact

Insert INTO StaffContact(sid, contact_no) values (StaffC_SID.NEXTVAL,'01452665746');
Insert INTO StaffContact(sid, contact_no) values (StaffC_SID.NEXTVAL,'01452665756');
Insert INTO StaffContact(sid, contact_no) values (StaffC_SID.NEXTVAL,'01452665774');
Insert INTO StaffContact(sid, contact_no) values (StaffC_SID.NEXTVAL,'01452663786');
Insert INTO StaffContact(sid, contact_no) values (StaffC_SID.NEXTVAL,'01452666666');

#Info1

Insert INTO Info1(uidofanimal, sid, a_id) values ('101','20010',Info1_aid.NEXTVAL);
Insert INTO Info1(uidofanimal, sid, a_id) values ('102','20011',Info1_aid.NEXTVAL);
Insert INTO Info1(uidofanimal, sid, a_id) values ('103','20012',Info1_aid.NEXTVAL);
Insert INTO Info1(uidofanimal, sid, a_id) values ('104','20013',Info1_aid.NEXTVAL);
Insert INTO Info1(uidofanimal, sid, a_id) values ('105','20014',Info1_aid.NEXTVAL);

#Info2

Insert INTO Info1(uidofhuman, sid, b_id) values ('3011','20010',Info2_seq.NEXTVAL);
Insert INTO Info1(uidofhuman, sid, b_id) values ('3012','20011',Info2_seq.NEXTVAL);
Insert INTO Info1(uidofhuman, sid, b_id) values ('3013','20012',Info2_seq.NEXTVAL);
Insert INTO Info1(uidofhuman, sid, b_id) values ('3014','20013',Info2_seq.NEXTVAL);
Insert INTO Info1(uidofhuman, sid, b_id) values ('3015','20014',Info2_seq.NEXTVAL);

#Staff

Insert INTO Staff(sid, name, fathers_name, mothers_name, email, road_no, house_no, age, gender, salary, shift, category ,zid, org_id) values (staff_seq.NEXTVAL,'Rahim','Mohammad Ali','Ammena','Rahim@gmail.com','113','52','32','Male','6800','Morning','Full time','1','2');
Insert INTO Staff(sid, name, fathers_name, mothers_name, email, road_no, house_no, age, gender, salary, shift, category ,zid, org_id) values (staff_seq.NEXTVAL,'Jerry','Mr.Johnson','Miss katty','Jerry@gmail.com','112','51','35','Male','7000','Night','Part time','2','24');

Insert INTO Staff(sid, name, fathers_name, mothers_name, email, road_no, house_no, age, gender, salary, shift, category ,zid, org_id) values (staff_seq.NEXTVAL,'Karim','Md.Abdul','Rahima','Abdul@gmail.com','114','53', '27','Male','6500','Night','Full time','3','28');

Insert INTO Staff(sid, name, fathers_name, mothers_name, email, road_no, house_no, age, gender, salary, shift, category ,zid, org_id) values (staff_seq.NEXTVAL,'Jonny','Mr.Akbar','Shahnara Begum','Jonny@gmail.com','115','54','36','Male','7800','Morning','Part time','4','29');
Insert INTO Staff(sid, name, fathers_name, mothers_name, email, road_no, house_no, age, gender, salary, shift, category ,zid, org_id) values (staff_seq.NEXTVAL,'Harry','Mr.Atkinso','Mrs. Helen','Harry@gmail.com','111','50','30','Male','6000','Morning','Part time','5','30');

#Organization_Contact

Insert INTO OrganizationContact(org_id,contact_no)
values (org_seq.NEXTVAL,'01882665706');
Insert INTO OrganizationContact(org_id,contact_no
)values (org_seq.NEXTVAL,'01752665786');
Insert INTO OrganizationContact(org_id,contact_no)
values (org_seq.NEXTVAL,'01982665796');
Insert INTO OrganizationContact(org_id,contact_no)
values (org_seq.NEXTVAL,'01472665746');
Insert INTO OrganizationContact(org_id,contact_no)
values (org_seq.NEXTVAL,'01999665789');

#Rescue

Insert INTO Rescue(nid,name,type,rescue_area,org_id)
values (res_seq.NEXTVAL,'Parry','Senior','Halishahr','2');
Insert INTO Rescue(nid,name,type,rescue_area,org_id) values
(res_seq.NEXTVAL,'Kaira','Senior','khilkhet','24');
Insert INTO Rescue(nid,name,type,rescue_area,org_id) values
(res_seq.NEXTVAL,'Mary','Junior','Mohammadpur','28');
Insert INTO Rescue(nid,name,type,rescue_area,org_id) values
(res_seq.NEXTVAL,'Helen','Senior','Sylhet','29');
Insert INTO Rescue(nid,name,type,rescue_area,org_id) values
(res_seq.NEXTVAL,'kilen','Junior','Tejgaon','30')

#Citizen_contact

Insert INTO citizen_contact(nid, contact_no) values (citcon_seq.NEXTVAL,'01882982346');
Insert INTO citizen_contact(nid, contact_no) values (citcon_seq.NEXTVAL,'01882980076');
Insert INTO citizen_contact(nid, contact_no) values (citcon_seq.NEXTVAL,'01856876346');
Insert INTO citizen_contact(nid, contact_no) values (citcon_seq.NEXTVAL,'01882710946');
Insert INTO citizen_contact(nid, contact_no) values (citcon_seq.NEXTVAL,'01884321346');

#Organization

```
Insert INTO Organization(org_id,email,road_no,house_no,zid,receipt_no)
values
(o_seq.NEXTVAL,'Robin@gmail.com','901','61',1,'1');
Insert INTO Organization(org_id,email,road_no,house_no,zid,receipt_no)
values
(o_seq.NEXTVAL,'hobin@gmail.com','902','62',2,'2');
Insert INTO Organization(org_id,email,road_no,house_no,zid,receipt_no)
values
(o_seq.NEXTVAL,'sarah@gmail.com','903','63',3,'21');
Insert INTO Organization(org_id,email,road_no,house_no,zid,receipt_no)
values
(o_seq.NEXTVAL,'rah@gmail.com','904','64',4,'22');
Insert INTO Organization(org_id,email,road_no,house_no,zid,receipt_no) values
(o_seq.NEXTVAL,'tayef@gmail.com','905','65',5,'23');
```

#Accounts

```
Insert INTO Accounts(receipt_no,received_cash)) values (a_seq.NEXTVAL,'10000');
Insert INTO Accounts(receipt_no,received_cash)) values (a_seq.NEXTVAL,'15000');
Insert INTO Accounts(receipt_no,received_cash) values (a_seq.NEXTVAL,'20000');
Insert INTO Accounts(receipt_no,received_cash) values (a_seq.NEXTVAL,'43000');
Insert INTO Accounts(receipt_no,received_cash) values (a_seq.NEXTVAL,'40000');
```

#Donation

```
Insert INTO Donation(receipt_no,donation_id, name,type,rescue_area) values ('1',do_seq.NEXTVAL, 'Sarah','','');
Insert INTO Donation(receipt_no,donation_id, name,type,rescue_area) values ('2',do_seq.NEXTVAL, 'Tayef', '','');
Insert INTO Donation(receipt_no,donation_id, name,type,rescue_area values ('21',do_seq.NEXTVAL, 'Jubayer', '','');
Insert INTO Donation(receipt_no,donation_id, name,type,rescue_area) values ('22',do_seq.NEXTVAL, 'Nishi',
'Child','');
Insert INTO Donation(receipt_no,donation_id, name,type,rescue_area) values ('23',do_seq.NEXTVAL, 'Anonymous',
'','');
```

#Human

```
Insert INTO Human(uidofhuman, name, age, gender, type, assignedhouse, rescue_area,
adoption_st) values (human_seq.NEXTVAL,'rahi,'06','Male',’Child’,'B1','Dhaka','Available');
Insert INTO Human(uidofhuman, name, age, gender, type, assignedhouse, rescue_area,
adoption_st) values (human_seq.NEXTVAL,'ruhi,'07','Female',’Child’,'B2','Dhaka','Available');
Insert INTO Human(uidofhuman, name, age, gender, type, assignedhouse, rescue_area,
adoption_st) values (human_seq.NEXTVAL,'riti’,'08','Female',’Child’,'B3','Dhaka','Available');
Insert INTO Human(uidofhuman, name, age, gender, type, assignedhouse, rescue_area,
adoption_st) values (human_seq.NEXTVAL,'rakib’,'06’,'Male',’Child’,'B5','Dhaka','Available');
```

SQL Query Writing

Single Row:

1. Write query to show the jobs of employee who work in Department 20 and sort the result by their names.

```
SELECT job

FROM emp

WHERE deptno = 20

    ORDER BY ename DESC;
```

2. Use substr function to show only the first three characters of empno=7566 Job title.

```
SELECT SUBSTR(job, 1, 3) AS job_prefix
FROM emp
    WHERE empno = 7566;
```

3. Use UPPER() and LOWER() function to convert the name of a specific employee into upper and lowercase letters.

```
SELECT UPPER(ename) AS uppercase_name, LOWER(ename) AS lowercase_name
FROM emp
    WHERE empno = 7788;
```

Group Function:

1. show how Number of animals are reday to get adopted select COUNT(*)

from animal where adoption_Status="Available"

2. show the oldest age the home

select Max(age) from Human

3. show how Number of babies are reday to get adopted select COUNT(*)

from Human where adoption_Status="Available"

Subquery:

1. show the details of the oldest citizen

select * from Human where age=(select Max(age) from Human)

2. show the staffs whose salary is greater than 4500

```
SELECT *
  FROM Staff
  WHERE sid IN (SELECT sID
      FROM staff
      WHERE SALARY > 4500)
```

3. deletes the records from the staff table for all the staffs whose AGE is greater than or equal to 27.

```
 DELETE FROM Staff
  WHERE AGE IN (SELECT AGE FROM Staff WHERE
    AGE >= 27 )
```

View:

1. create a view containing all details of staff CREATE

VIEW StaffsInfo AS

SELECT * from Staff

2. create a view containing all details animals available to adopt

CREATE VIEW AnimalsAvailableToAdopt AS

SELECT * from animal where Adoption_Status="Available"

3. create a view containing all details of organization

CREATE VIEW orgInfo AS SELECT *

from organization

Join:

select sid,name, road_no from staff, organization where staff.orgid=organization.orgid

select name,type,received cash from donation,accounts where accounts.receiptNo=Donation.receiptNo

Synonym:

CREATE PUBLIC SYNONYM SDetails

FOR staff;

CREATE PUBLIC SYNONYM ADetails

FOR Animals;

CREATE PUBLIC SYNONYM donationinfo FOR

Donations;

PL/SQL Query

Procedure:

1. Create a procedure to update the salary of the staffs from Staff table.

Solution:

```
CREATE OR REPLACE PROCEDURE adjust_salary(
   in_employee_id IN STAFF.SID%TYPE,
   in_percent IN NUMBER
) IS
BEGIN
  -- update employee's salary
  UPDATE Staff
  SET salary = salary + salary * in_percent / 100
  WHERE sid = in_employee_id;
END;
```

Execution:

```
BEGIN
  adjust_salary(in_employee_id => 2, in_percent => 10);
END;
```

```
CREATE OR REPLACE PROCEDURE adjust_salary(
    in_employee_id IN STAFF.SID%TYPE,
    in_percent IN NUMBER
) IS
BEGIN
   -- update employee's salary
   UPDATE Staff
   SET salary = salary + salary * in_percent / 100
   WHERE sid = in_employee_id;
END;
Execution:
BEGIN
   adjust_salary(in_employee_id => 2, in_percent => 10);
END;
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

2. Create a procedure to get the information about an individual rescue (caller id, type, and rescue area) from the rescue table.

Solution:

```
CREATE OR REPLACE PROCEDURE get_rescue_details(p_res_id IN Rescue.NID%TYPE)
IS
  v_caller_id Rescue.NID%TYPE;
  v_type Rescue.Type%TYPE;
  v_area Rescue.Rescue_area%TYPE;
BEGIN

  SELECT nid, type, rescue_area
  INTO v_caller_id,  v_type, v_area
```

```
  FROM Rescue
  WHERE NID = p_res_id;

  DBMS_OUTPUT.PUT_LINE('Caller id: ' || v_caller_id);
  DBMS_OUTPUT.PUT_LINE('Type: ' || v_type);
  DBMS_OUTPUT.PUT_LINE('Area: ' || v_area);
END;
```

Execution:

```
BEGIN
  get_rescue_details(p_res_id => 2);
END;
```

```
  SELECT nid, type, rescue_area
  INTO v_caller_id,  v_type, v_area
  FROM Rescue
  WHERE NID = p_res_id;

  DBMS_OUTPUT.PUT_LINE('Caller id: ' || v_caller_id);
  DBMS_OUTPUT.PUT_LINE('Type: ' || v_type);
  DBMS_OUTPUT.PUT_LINE('Area: ' || v_area);
END;
Execution:
BEGIN
  get_rescue_details(p_res_id => 2);
END;
```

Results Explain Describe Saved SQL History

```
Caller id: 2
Type: Senior
Area: khilkhet

Statement processed.
```

3. Create a procedure that retrieves the animal details based on the unique id of an animal(primary key) from the animal table.

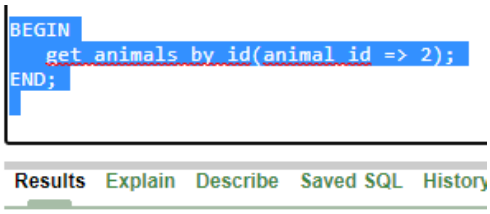
Solution:

```
CREATE OR REPLACE PROCEDURE get_animals_by_id(
   animal_id IN Animal.uidofanimal%TYPE
) IS
BEGIN
  FOR an_rec IN (
     SELECT uidofanimal, type,adoption_st, assignedhouse
     FROM animal
     WHERE uidofanimal = animal_id
   ) LOOP
     DBMS_OUTPUT.PUT_LINE('Animal uid ' || an_rec.uidofanimal);
     DBMS_OUTPUT.PUT_LINE('Type: ' || an_rec.type );
     DBMS_OUTPUT.PUT_LINE('House: ' || an_rec.assignedhouse);
     DBMS_OUTPUT.PUT_LINE('Status: ' || an_rec.adoption_st);

   END LOOP;
END;
```

Execution:

```
BEGIN
  get_animals_by_id(animal_id => 2);
END;
```




```
Animal uid 2
Type: Cat
House: A3
Status: Availabl
e

Statement processed.
```

Function:

1. Create a function to get the total count of staff working as a “Full time” employee from staff table.

Solution:

```
CREATE OR REPLACE FUNCTION get_staff_count(category IN varchar2) RETURN NUMBER
IS
  s_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO s_count
  FROM staff
  WHERE category = get_staff_count.category ;
  RETURN s_count;
END;
/
DECLARE
  staff_count NUMBER;
BEGIN
  staff_count := get_staff_count('Full time');
  DBMS_OUTPUT.PUT_LINE('Number of staff working in Full time: ' || staff_count);
END;
```

```
BEGIN
  SELECT COUNT(*) INTO s_count
  FROM staff
  WHERE category = get_staff_count.category ;
  RETURN s_count;
END;


DECLARE
  staff_count NUMBER;
BEGIN
```

Results Explain Describe Saved SQL History

```
Number of staff working in Full time: 2

Statement processed.
```

0.02 seconds

2. Create a function that to get the average salary of all staffs.

Solution:

```
CREATE OR REPLACE FUNCTION get_avg_salary_staff RETURN NUMBER
IS
 avg_salary NUMBER;
BEGIN
 SELECT AVG(salary) INTO avg_salary
 FROM staff;
 RETURN avg_salary;
END;
/
DECLARE
 avg_salary NUMBER;
BEGIN
 avg_salary := get_avg_salary_staff();
 DBMS_OUTPUT.PUT_LINE('The average salary of all staff members is: ' || avg_salary);
END;
```

```
  avg_salary NUMBER;
BEGIN
  avg_salary := get_avg_salary_staff();
  DBMS_OUTPUT.PUT_LINE('The average salary of all
```

Results Explain Describe Saved SQL History

```
The average salary of all staff members is: 6820

Statement processed.
```

3. Create a function to display the salary of an individual staff member.

Solution:

```
CREATE OR REPLACE FUNCTION salary_total(staff_id IN NUMBER) RETURN NUMBER
IS
 total_salary NUMBER;
BEGIN
```

```
  SELECT SUM(salary) INTO total_salary
  FROM staff
  WHERE sid = staff_id;

  RETURN total_salary;
END;
/
DECLARE
  staff_id NUMBER := 22;
  total_salary NUMBER;
BEGIN
  total_salary := salary_total(staff_id);
  DBMS_OUTPUT.PUT_LINE('Total salary for staff ' || staff_id || ': ' || total_salary);
END;
```

```
DECLARE
  staff_id NUMBER := 22;
  total_salary NUMBER;
BEGIN
```

Results Explain Describe Saved SQL

```
Total salary for staff 22: 7000

Statement processed.
```

0.04 seconds

Cursor

Question 01: How can I retrieve and display the UIDOFANIMAL, HEALTH_CONDITION, and ADOPTION_ST for animals in the 'Animal' table with a RESCUE_AREA of 'Dhaka' using PL/SQL? Additionally, how can I differentiate between animals that are adopted and those that are available for adoption?

Answer:
```
DECLARE
  CURSOR c_animals IS
    SELECT UIDOFANIMAL, HEALTH_CONDITION, ADOPTION_ST
    FROM Animal
    WHERE RESCUE_AREA = 'Dhaka';
BEGIN
  FOR animal_rec IN c_animals LOOP
    DBMS_OUTPUT.PUT_LINE('Animal ID: ' || animal_rec.UIDOFANIMAL);
    DBMS_OUTPUT.PUT_LINE('Health Condition: ' || animal_rec.HEALTH_CONDITION);
    IF animal_rec.ADOPTION_ST = 'adopted' THEN
      DBMS_OUTPUT.PUT_LINE('Adoption Status: Adopted');
    ELSE
      DBMS_OUTPUT.PUT_LINE('Adoption Status: Available');
    END IF;
    DBMS_OUTPUT.PUT_LINE('--------------------------');
  END LOOP;
END;
/
```

```
Animal ID: 1
Health Condition: Healthy
Adoption Status: Available
--------------------------
Animal ID: 2
Health Condition: Good
Adoption Status: Available
--------------------------
Animal ID: 3
Health Condition: Average
Adoption Status: Available
--------------------------
Animal ID: 4
Health Condition: Healthy
Adoption Status: Available
--------------------------
Animal ID: 5
Health Condition: Healthy
Adoption Status: Available
--------------------------

Statement processed.

0.03 seconds
```

Question02: Write a PL/SQL command with PL/SQL Cursor that retrieves the addresses from the 'Address' table where the 'ZIP_CODE' is either 301, 302, 303, or 304 and the 'CITY' is either 'Dhaka', 'Khulna', or 'Sylhet'. Display the address ID, ZIP code, and city for each address.

Answer:

```
DECLARE
  CURSOR c_address IS
    SELECT ZID, ZIP_CODE, CITY
    FROM Address
    WHERE ZIP_CODE IN (301, 302, 303, 304)
    AND CITY IN ('Dhaka', 'Khulna', 'Sylhet');
BEGIN
  FOR address_rec IN c_address LOOP
    DBMS_OUTPUT.PUT_LINE('Address ID: ' || address_rec.ZID);
    DBMS_OUTPUT.PUT_LINE('ZIP Code: ' || address_rec.ZIP_CODE);
    DBMS_OUTPUT.PUT_LINE('City: ' || address_rec.CITY);
    DBMS_OUTPUT.PUT_LINE('---------------------------------------');
  END LOOP;
END;
/
```

```
Address ID: 1
ZIP Code: 301
City: Dhaka
---------------------------------------
Address ID: 2
ZIP Code: 303
City: Khulna
---------------------------------------
Address ID: 3
ZIP Code: 304
City: Sylhet
---------------------------------------

Statement processed.

0.00 seconds
```

Question 03: Display the values of 'ROAD_NO', 'HOUSE_NO', 'ORG_ID', 'ZID', 'RECEIPT_NO', and 'EMAIL' for each selected row using PL/SQL cursor.

```
DECLARE
  CURSOR c_organization IS
    SELECT ROAD_NO, HOUSE_NO, ORG_ID, ZID, RECEIPT_NO, EMAIL
    FROM Organization
```

```
   WHERE ROAD_NO IN (901, 902, 903, 903)
   AND HOUSE_NO IN (61, 62, 63, 63)
   AND ORG_ID IN (8, 9, 10, 13)
   AND ZID IN (1, 2, 3, 3)
   AND RECEIPT_NO IN (1, 2, 3, 3)
   AND EMAIL IN ('Robin@gmail.com', 'hobin@gmail.com', 'sarah@gmail.com', 'sarah@gmail.com');
BEGIN
  FOR org_rec IN c_organization LOOP
    DBMS_OUTPUT.PUT_LINE('Road No: ' || org_rec.ROAD_NO);
    DBMS_OUTPUT.PUT_LINE('House No: ' || org_rec.HOUSE_NO);
    DBMS_OUTPUT.PUT_LINE('Organization ID: ' || org_rec.ORG_ID);
    DBMS_OUTPUT.PUT_LINE('ZID: ' || org_rec.ZID);
    DBMS_OUTPUT.PUT_LINE('Receipt No: ' || org_rec.RECEIPT_NO);
    DBMS_OUTPUT.PUT_LINE('Email: ' || org_rec.EMAIL);
    DBMS_OUTPUT.PUT_LINE('---------------------------------------');
  END LOOP;
END;
```

```
Road No: 901
House No: 61
Organization ID: 8
ZID: 1
Receipt No: 1
Email: Robin@gmail.com
---------------------------------------
Road No: 902
House No: 62
Organization ID: 9
ZID: 2
Receipt No: 2
Email: hobin@gmail.com
---------------------------------------
Road No: 903
House No: 63
Organization ID: 10
ZID: 3
Receipt No: 3
Email: sarah@gmail.com
---------------------------------------
Road No: 903
House No: 63
Organization ID: 13
ZID: 3
Receipt No: 3
Email: sarah@gmail.com
---------------------------------------

Statement processed.


0.09 seconds
```

Trigger

Question01: Write a PL/SQL command that creates a trigger for the 'Address' table. The trigger should ensure that only rows with a 'ZIP_CODE' of 301, 302, 303, or 304 and a 'CITY' of 'Dhaka', 'Khulna', or 'Sylhet' can be inserted. If the conditions are not met, an exception should be raised with the message 'Invalid ZIP code or city for Address'.

Solution:

```
CREATE OR REPLACE TRIGGER address_trigger
BEFORE INSERT ON Address
FOR EACH ROW
BEGIN
  IF :NEW.ZIP_CODE IN (301, 302, 303, 304) AND :NEW.CITY IN ('Dhaka', 'Khulna', 'Sylhet') THEN
    -- Perform desired actions or validations here
    -- You can add your logic to handle the specific case
    NULL; -- Placeholder action, replace with your code
  ELSE
    -- Raise an exception or handle the invalid case
```

```
    RAISE_APPLICATION_ERROR(-20001, 'Invalid ZIP code or city for Address');
  END IF;
END;
/
```

Autocommit Display 200

```
CREATE OR REPLACE TRIGGER address_trigger
BEFORE INSERT ON Address
FOR EACH ROW
BEGIN
  IF :NEW.ZIP_CODE IN (301, 302, 303, 304) AND :NEW.CITY IN ('Dhaka', 'Khulna', 'Sylhet') THEN
    -- Perform desired actions or validations here
    -- You can add your logic to handle the specific case
    NULL; -- Placeholder action, replace with your code
  ELSE
    -- Raise an exception or handle the invalid case
    RAISE_APPLICATION_ERROR(-20001, 'Invalid ZIP code or city for Address');
  END IF;
END;
/
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.03 seconds

Question02 : How does the provided PL/SQL trigger and PL/SQL block in Oracle help maintain the adoption status of animals in the 'Animal' table based on their rescue area and health condition?

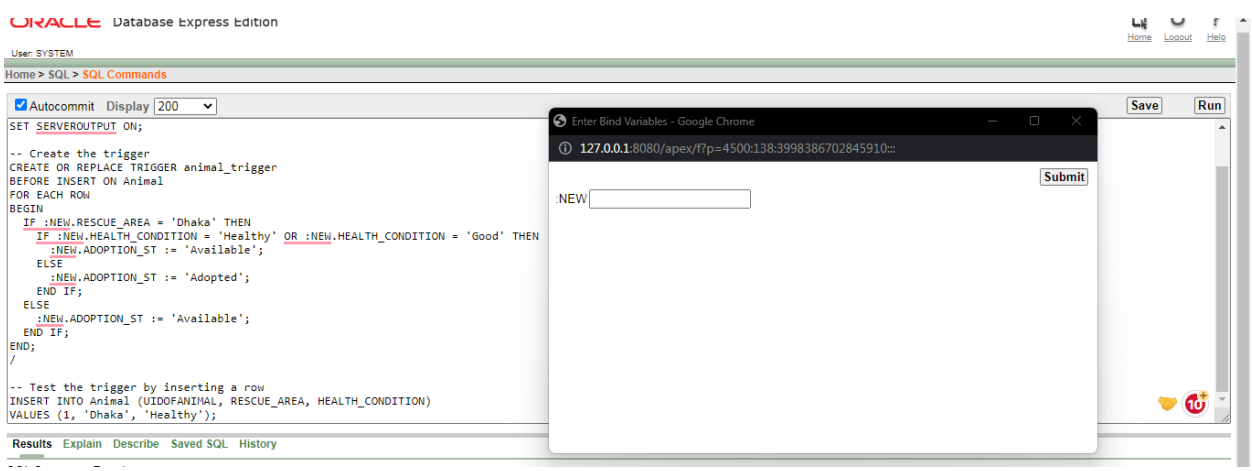
```
CREATE OR REPLACE TRIGGER animal_trigger
BEFORE INSERT ON Animal
FOR EACH ROW
BEGIN
  IF :NEW.RESCUE_AREA = 'Dhaka' THEN
    IF :NEW.HEALTH_CONDITION = 'Healthy' OR :NEW.HEALTH_CONDITION = 'Good' THEN
      :NEW.ADOPTION_ST := 'Available';
    ELSE
      :NEW.ADOPTION_ST := 'Adopted';
    END IF;
  ELSE
    :NEW.ADOPTION_ST := 'Available';
  END IF;
END;
/
```

```
CREATE OR REPLACE TRIGGER animal_trigger
BEFORE INSERT ON Animal
FOR EACH ROW
BEGIN
  IF :NEW.RESCUE_AREA = 'Dhaka' THEN
    IF :NEW.HEALTH_CONDITION = 'Healthy' OR :NEW.HEALTH_CONDITION = 'Good' THEN
      :NEW.ADOPTION_ST := 'Available';
    ELSE
      :NEW.ADOPTION_ST := 'Adopted';
    END IF;
  ELSE
    :NEW.ADOPTION_ST := 'Available';
  END IF;
END;
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

```
CREATE OR REPLACE TRIGGER animal_trigger
BEFORE INSERT ON Animal
FOR EACH ROW
BEGIN
  IF :NEW.RESCUE_AREA = 'Dhaka' THEN
    IF :NEW.HEALTH_CONDITION = 'Healthy' OR :NEW.HEALTH_CONDITION = 'Good' THEN
      :NEW.ADOPTION_ST := 'Available';
    ELSE
      :NEW.ADOPTION_ST := 'Adopted';
    END IF;
  ELSE
    :NEW.ADOPTION_ST := 'Available';
  END IF;
END;
/
-- Testing the trigger by inserting a row
INSERT INTO Animal (UIDOFANIMAL, RESCUE_AREA, HEALTH_CONDITION)
VALUES (1, 'Dhaka', 'Healthy');
```



Question 03: Write a PL/SQL command that creates a trigger for the 'Organization' table. The trigger should ensure that only rows with specific values for the columns 'ROAD_NO', 'HOUSE_NO', 'ORG_ID', 'ZID', 'RECEIPT_NO', and 'EMAIL' can be inserted. If the conditions are not met, an exception should be raised with the message 'Invalid values for Organization'.

Answer:

```
CREATE OR REPLACE TRIGGER organization_trigger
BEFORE INSERT ON Organization
FOR EACH ROW
BEGIN
  IF :NEW.ROAD_NO IN (901, 902, 903, 903)
    AND :NEW.HOUSE_NO IN (61, 62, 63, 63)
    AND :NEW.ORG_ID IN (8, 9, 10, 13)
    AND :NEW.ZID IN (1, 2, 3, 3)
    AND :NEW.RECEIPT_NO IN (1, 2, 3, 3)
    AND :NEW.EMAIL IN ('Robin@gmail.com', 'hobin@gmail.com', 'sarah@gmail.com', 'sarah@gmail.com') THEN
    NULL;
  ELSE
    -- Raising an exception or handle the invalid case
    RAISE_APPLICATION_ERROR(-20001, 'Invalid values for Organization');
  END IF;
END;
/
```

Autocommit Display 200

```
CREATE OR REPLACE TRIGGER organization_trigger
BEFORE INSERT ON Organization
FOR EACH ROW
BEGIN
  IF :NEW.ROAD_NO IN (901, 902, 903, 903)
    AND :NEW.HOUSE_NO IN (61, 62, 63, 63)
    AND :NEW.ORG_ID IN (8, 9, 10, 13)
    AND :NEW.ZID IN (1, 2, 3, 3)
    AND :NEW.RECEIPT_NO IN (1, 2, 3, 3)
    AND :NEW.EMAIL IN ('Robin@gmail.com', 'hobin@gmail.com', 'sarah@gmail.com', 'sarah@gmail.com') THEN
    -- Perform desired actions or validations here
    -- You can add your logic to handle the specific case
    NULL; -- Placeholder action, replace with your code
  ELSE
    -- Raise an exception or handle the invalid case
    RAISE_APPLICATION_ERROR(-20001, 'Invalid values for Organization');
  END IF;
END;
/
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.03 seconds

Record:

1: create a record for animal table. The record should return animal type, assigned house and adoption status of an animal on the basis of animal id.

Ans:

```
declare
animal_rec animal%rowtype;
begin
select * into animal_rec from animal
where uidofanimal=1;
dbms_output.put_line(animal_rec.Type||''||animal_rec.assignedhouse||''||animal_rec.adoption_st);
end
/
```

```
declare
animal_rec animal%rowtype;
begin
select * into animal_rec from animal
where uidofanimal=1;
dbms_output.put_line(animal_rec.Type||' '||animal_rec.assignedhouse||' '||animal_rec.adoption_st);
end
/
```

Results Explain Describe Saved SQL History

```
Cat C3 Available

Statement processed.
```

2: create a record for address table. The record should return uid and city on the basis of zip code.

```
declare
cursor c_address is
select * from address where zip_code=301;
rec_address address%rowtype;
begin
open c_address;
fetch c_address into rec_address;
dbms_output.put_line(rec_address.zid||' '||rec_address.city);
close c_address;
end;
/
```

Home > SQL > SQL Commands

Autocommit Display 10

```
declare
cursor c_address is
select * from address where zip_code=301;
rec_address address%rowtype;
begin
open c_address;
fetch c_address into rec_address;
dbms_output.put_line(rec_address.zid||' '||rec_address.city);
close c_address;
end;
/
```

Results Explain Describe Saved SQL History

```
1 Dhaka

Statement processed.
```

3: Create an user defined record to enter and return animal title and animal id of atleast two animal

```
DECLARE
  type Animal is record
    (title varchar2(50),
     animal_id number);
  animal1 Animal;
  animal2 Animal;

BEGIN
  -- Animal 1 specification
  animal1.title  := 'Cat';
  animal1.animal_id := 6495407;
  -- animal 2 specification
  animal2.title := 'Dog';
  animal2.animal_id := 6495700;

 -- Print animal 1 record
  dbms_output.put_line('animal 1 title : '|| animal1.title);
  dbms_output.put_line('animal 1 animal_id : ' || animal1.animal_id);

  -- Print Animal 2 record
  dbms_output.put_line('animal 2 title : '|| animal2.title);
  dbms_output.put_line('animal 2 animal_id : '|| animal2.animal_id);
END;
/
```

```
DECLARE
   type Animal is record
      (title varchar2(50),
       animal_id number);
   animal1 Animal;
   animal2 Animal;

BEGIN
   -- Animal 1 specification
   animal1.title  := 'Cat';
   animal1.animal_id := 6495407;
   -- animal 2 specification
```

Results Explain Describe Saved SQL History

```
animal 1 title : Cat
animal 1 animal_id : 6495407
animal 2 title : Dog
animal 2 animal_id : 6495700

Statement processed.
```

Package:

1: Create a package on animal table to return Animal type and adoption status by animal id ;

```
CREATE OR REPLACE PACKAGE animal_pack AS

  PROCEDURE display_type(a__id animal.uidofanimal%type);

  PROCEDURE display_adoption_st(a__id animal.uidofanimal%type);

END animal_pack;
```

Autocommit Display 10

```
CREATE OR REPLACE PACKAGE animal_pack AS
   PROCEDURE display_type(a__id animal.uidofanimal%type);
   PROCEDURE display_adoption_st(a__id animal.uidofanimal%type);
END animal_pack;
```

Results Explain Describe Saved SQL History

```
Package created.
```

```
CREATE OR REPLACE PACKAGE BODY  animal_pack AS

  PROCEDURE display_type(a__id animal.uidofanimal%TYPE) IS
  e_nam animal.type%type;
  BEGIN
    SELECT type INTO e_nam
    FROM animal
    WHERE uidofanimal = a__id;
    dbms_output.put_line('Animal Type: '|| e_nam);
  END display_type;

PROCEDURE display_adoption_st(a__id animal.uidofanimal%type) IS
  e_st animal.adoption_st%type;
  BEGIN
    SELECT adoption_st INTO e_st
    FROM animal
    WHERE uidofanimal = a__id;
```

```
    dbms_output.put_line('Adoption Status '|| e_st);

  END display_adoption_st;

END animal_pack;
```

```
CREATE OR REPLACE PACKAGE BODY  animal_pack AS

   PROCEDURE display_type(a__id animal.uidofanimal%TYPE) IS
   e_nam animal.uidofanimal%type;
   BEGIN
      SELECT type INTO e_nam
      FROM animal
      WHERE uidofanimal = a__id;
      dbms_output.put_line('Animal Type: '|| e_nam);
   END display_type;
```

Results Explain Describe Saved SQL History

```
Package Body created.
```

/

```
begin

animal_pack.display_type('1');

animal_pack.display_adoption_st('1');

end
```

```
begin
animal_pack.display_type('1');
animal_pack.display_adoption_st('1');
end
```

Results Explain Describe Saved SQL History

```
Animal Type: Cat
Adoption Status Available

Statement processed.
```

2: Create a package on address table to return zip code and city name by zid ;

CREATE OR REPLACE PACKAGE address_pack AS

PROCEDURE display_Zipcode(a__id address.Zip_code%type);

PROCEDURE display_city(a__id address.Zip_code%type);

END address_pack;

```
CREATE OR REPLACE PACKAGE address_pack AS
   PROCEDURE display_Zipcode(a__id address.Zip_code%type);
   PROCEDURE display_city(a__id address.Zip_code%type);
END address_pack;
```

Results Explain Describe Saved SQL History

```
Package created.
```

CREATE OR REPLACE PACKAGE BODY address_pack AS

```
  PROCEDURE display_zipcode(a__id address.Zip_code%type) IS
  e_nam address.zip_code%type;
  BEGIN
    SELECT Zip_code INTO e_nam
    FROM address
    WHERE zid = a__id;
    dbms_output.put_line('Zip Code : '|| e_nam);
  END display_zipcode;

PROCEDURE display_City(a__id address.Zip_code%type) IS
  e_st address.city%type;
  BEGIN
    SELECT city INTO e_st
    FROM address
    WHERE zid = a__id;
    dbms_output.put_line('City  :'|| e_st);
  END display_City;
END address_pack;
/
```

```
CREATE OR REPLACE PACKAGE BODY  address_pack AS

   PROCEDURE display_zipcode(a__id address.Zip_code%type) IS
   e_nam address.zip_code%type;
   BEGIN
      SELECT Zip_code INTO e_nam
      FROM address
      WHERE zid = a__id;
      dbms_output.put_line('Zip Code : '|| e_nam);
   END display_zipcode;
```

Results Explain Describe Saved SQL History

```
Package Body created.
```

```
begin

address_pack.display_zipcode('1');

address_pack.display_city('1');

end
```

```
begin
address_pack.display_zipcode('1');
address_pack.display_city('1');
end
```

Results Explain Describe Saved SQL History

```
Zip Code : 301
City  :Dhaka

Statement processed.
```

0.02 seconds

3 : Create a package on Staff_contact table to return sid and contact no by SID ;

```
CREATE OR REPLACE PACKAGE staff_contact_pack AS

  PROCEDURE display_sid(a__id staff_contact.sid%type);

  PROCEDURE display_cont(a__id staff_contact.sid%type);

END staff_contact_pack;
```

Autocommit Display 10

```
CREATE OR REPLACE PACKAGE staff_contact_pack AS
   PROCEDURE display_sid(a__id staff_contact.sid%type);
   PROCEDURE display_cont(a__id staff_contact.sid%type);
END staff_contact_pack;
```

Results Explain Describe Saved SQL History

```
Package Body created.
```

0.02 seconds

```
CREATE OR REPLACE PACKAGE BODY  staff_contact_pack AS

  PROCEDURE display_sid(a__id staff_contact.sid%type) IS
  e_nam staff_contact.sid%type;
  BEGIN
    SELECT sid INTO e_nam
    FROM staff_contact
    WHERE sid = a__id;
    dbms_output.put_line(' ID  : '|| e_nam);
  END display_sid;

PROCEDURE display_cont(a__id staff_contact.sid%type) IS
  e_st staff_contact.contact_no%type;
  BEGIN
    SELECT contact_no INTO e_st
```

```
    FROM staff_contact
    WHERE sid = a__id;
    dbms_output.put_line('Contact No :'|| e_st);
  END display_cont;
END staff_contact_pack;
/
```

```
CREATE OR REPLACE PACKAGE BODY  staff_contact_pack AS

   PROCEDURE display_sid(a__id staff_contact.sid%type) IS
   e_nam staff_contact.sid%type;
   BEGIN
      SELECT sid INTO e_nam
      FROM staff_contact
      WHERE sid = a__id;
      dbms_output.put_line(' ID  : '|| e_nam);
   END display_sid;
```

Results Explain Describe Saved SQL History

```
Package Body created.
```

```
begin
staff_contact_pack.display_sid('1');
staff_contact_pack.display_cont('1');
end
```

```
begin
staff_contact_pack.display_sid('1');
staff_contact_pack.display_cont('1');
end
```

Results Explain Describe Saved SQL History

```
 ID  : 1
Contact No :01452665746

Statement processed.
```

Relational Algebra

1. Find the animal who is assigned at house "C3".
   Solution:
   $$\Pi_{uidofanimal}(\sigma_{assignedhouse="C3"}(animal))$$
2. Find all salaries of over $1200
   Solution:
   $$\sigma_{amount > 1200}(salary)$$
3. Delete all rescue record with 'Dhaka'
   Solution:
   $$rescue \leftarrow rescue - \sigma_{rescue\_area = "Dhaka"}(rescue)$$

4. Make salary increase by 5 percent in Staff table.
   Solution:
   $$staff \leftarrow \Pi_{sid, name, salary * 1.05}(staff)$$
5. Find the receipt no. , name from the donation.
   Solutions:
   $$\Pi_{donation\_id}(receipt\_no) \cup \Pi_{donation\_id}(name)$$

Conclusion

The project is currently managed by a team of volunteers and staff members who are responsible for rescuing and providing care for those in need, promoting adoption services, and managing donations and volunteer opportunities. Our project findings indicate that we have made a significant impact on the lives of those we serve by providing a loving environment and care for their well-being. Our outreach efforts have increased public awareness of our cause and encouraged others to make a positive impact. However, we recognize that there is still much work to be done, and we are committed to improving our existing project to make an even greater impact. In the future, we plan to expand our outreach efforts to reach more individuals and raise awareness of our cause. We also plan to increase our efforts to promote adoption services and find loving homes for those in need. Additionally, we aim to provide more volunteer opportunities for individuals to get involved and make a difference in their communities. Our proposed future work will enable us to improve our existing project and make an even greater impact on the lives of those we serve. We welcome donations and volunteers to help support our mission and encourage others to get involved and make a positive difference in the world.

THANK YOU