**NORMALIZATION:**

**ORGANIZATION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| O\_ID | O\_NAME | O\_EMAIL | O\_PHONE | O\_ZIP |

1NF: YES. O\_ID is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on O\_ID.

3NF: YES. O\_ID directly depends every other non-PK attribute.

**CUSTOMER:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C\_ID | C\_FNAME | C\_LNAME | C\_EMAIL | C\_PHONE | C\_ZIP |

1NF: YES. C\_ID is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on C\_ID.

3NF: YES. C\_ID directly depends every other non-PK attribute.

**ADDRESS**

|  |  |  |
| --- | --- | --- |
| ZIP | CITY | STATE |

1NF: YES. ZIP is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on ZIP.

3NF: YES. ZIP directly depends every other non-PK attribute.

**ANIMAL:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A\_ID | A\_AGE | A\_COLOR | A\_GENDER | O\_ID | C\_ID | Breed\_ID |

1NF: YES. A\_ID is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on A\_ID.

3NF: YES. A\_ID directly depends every other non-PK attribute.

**BREED:**

|  |  |  |
| --- | --- | --- |
| B\_ID | B\_ANIMAL | B\_BREED |

1NF: YES. B\_ID is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on B\_ID.

3NF: YES. B\_ID directly depends every other non-PK attribute.

**REQUEST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUEST\_ID | REQUESTER\_ID | BREED\_ID | ANIMAL\_ID | APPROVAL\_STATUS |

1NF: YES. REQUEST\_ID is the PK and determines every other non-PK attribute.

2NF: YES. No composite PK, every non-PK attribute fully FDs on REQUEST\_ID.

3NF: YES. REQUEST\_ID directly depends every other non-PK attribute.

**MEDICAL HISTORY: (Our original idea was to show whether the animals are vaccinated or not,thus if the animal is vaccinated again with the same vaccine, then we overwrite the values)**

|  |  |  |  |
| --- | --- | --- | --- |
| DISEASE | VACCINATION\_DATE | VACCINATION\_TYPE | ANIMAL\_ID |

1NF: Yes, primary key is present and all other attributes depends on pk.

2NF: No, Reason: further normalization would result into too many small tables.

**\*ADOPTION HISTORY:** (one of ORG\_ID and CUST\_ID is null)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORG\_ID | CUST\_ID | ADOPTION\_DATE | ADOPTER\_ID | PET\_ID |

FK-ORG\_ID,CUST\_ID,ADOPTER\_ID,PET\_ID

No Primary Key present so Not present in 1NF,2NF or 3NF .

Mapping

**ADDRESS**

|  |  |  |
| --- | --- | --- |
| ZIP | CITY | STATE |

**ORGANIZATION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| O\_ID | O\_NAME | O\_EMAIL | O\_PHONE | O\_ZIP |

**CUSTOMER:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C\_ID | C\_FNAME | C\_LNAME | C\_EMAIL | C\_PHONE | C\_ZIP |

**ANIMAL:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A\_ID | A\_AGE | A\_COLOR | A\_GENDER | O\_ID | C\_ID | BREED\_ID |

**BREED:**

|  |  |  |
| --- | --- | --- |
| B\_ID | B\_ANIMAL | B\_BREED |

**REQUEST:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUEST\_ID | REQUESTER\_ID | BREED\_ID | ANIMAL\_ID | APPROVAL\_STATUS |

**MEDICAL HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| ANIMAL\_ID | VACCINATION\_TYPE | VACCINATION\_DATE | DISEASE |

**ADOPTION HISTORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORG\_ID | CUST\_ID | ADOPTION\_DATE | ADOPTER\_ID | PET\_ID |

Demo Queries:

1. See the request made by user whose request\_id is 1001.

Select \*

from REQUEST

where REQUESTER\_ID ='1001';

1. See the request made by user whose first name is “Charlie”.

Select \*

from REQUEST

where REQUESTER\_ID in (select C\_ID from CUSTOMER where C\_FNAME= 'Charlie');

1. List all the organizations located in california where zip code is 90007.

Select \*

from ORGANIZATION

where O\_ZIP = 90007;

1. List all the available “persian cats”

Select \*

from ANIMAL inner join BREED on (ANIMAL.BREED\_ID = BREED.B\_ID

and BREED.B\_ANIMAL='CAT' and BREED.B\_BREED='PERSIAN');

OR

Select \*

From ANIMAL

Where BREED\_ID IN (

Select B\_ID

From BREED

Where B\_ANIMAL = 'Cat' and

B\_BREED = 'Persian'

);

1. List the A\_ID, O\_ID, and C\_ID of all the animals who has been vaccinated with “Rabbies”

Select A\_ID, O\_ID, C\_ID

from ANIMAL inner join MEDICAL\_HISTORY on A\_ID = ANIMAL\_ID

where MEDICAL\_HISTORY.VACCINATION\_TYPE='RABBIES';

1. Register new user.

insert into CUSTOMER values(1011, Sid,Gore’, 'bjones@gmail.com',6699894832,95054);

1. List the animal id and the breed id of all the “dogs” available in the organization from the same zip area as of customer ‘1001’ .

Select A\_ID, B\_ID

From ANIMAL inner join BREED on BREED\_ID = B\_ID

Where B\_ANIMAL = ‘Dog’

and

O\_ID = (

Select O\_ID

From ORGANIZATION

Where O\_ZIP = (

Select C\_ZIP

From CUSTOMER

Where C\_ZIP = ‘1001’

)

);

1. List the wish list.(ask abhi)
2. List the A\_ID, O\_ID, C\_ID of the dogs who are of age <= 1 year.

Select A\_ID, O\_ID, C\_ID

From ANIMAL

Where A\_AGE <= 1

and

BREED\_ID IN (

Select B\_ID

From BREED

Where B\_ANIMAL = ‘Dog’

);

1. Count all the requests made by the customer for “Marshall”

1. List all the pending requests and then approve the request made for animal\_id =”

List all the available animals.

**OPERATIONS:**

1. **A** person should be able to register himself (update his details if already exist).**Sign up of a new customer, who wish to give/adopt a pet via this agency. 4.) Modify the personal information**

**insert into ORGANIZATION values(01, 'MARSHALL',** [**'marshall@gmail.com**](mailto:%27marshall@gmail.com)**',1129894832,95054);**

**insert into CUSTOMER values(1001, 'Bob','Jones',** [**'bjones@gmail.com**](mailto:%27bjones@gmail.com)**',6699894832,95054);**

1. **The** db user(organization/customer) should be able to see his details. OR Login Functionality for the customers and organization.
2. **A** donor(organization/ person) should be able to see the request, whose approval are pending on him.
3. **A** donor should be able to approve the request (this approval of request should update the columns in the table: request, animal and adoption history).Approve and reject feature let organizations approve or deny the requests.
4. **An** adopter, while looking for an animal should be able to -
5. Look at the breeds offered.
6. And then using the breed id know the pets available.

5. **A** requester should be able to know the adoption history of an animal from the animal id.

6. **A** requester should be able to browse the medical history of an animal from the animal id.

7. **A** requester should be able to see the status of the request he made.

8. **A** donor should be able to put down the information of a new pet he wants to donate.

**Update from the organization to agency about the changes in the availabilty of the pet.**

9. A donor should be able to put on re-adoption an animal which he took from the agency. (WHAT>>)

10. **S**how all the requests made by a person till date.

11:Sign up for a new animal-rescue organization who wish to be part of Pet Word

**12.A wish list feature in which a customer can reserve a choice of pet when it is not available**

**13. Updates from the agency to the organization if a new request for an animal arrives.**

14. Raise a request for a pet.

**STORED PROCEDURES-**

create or replace procedure

approve\_pet\_request(req\_id in integer)

is

anim\_id integer(5);

request\_by\_id integer(5);

owner\_id integer(5);

begin

select animal\_id, requester\_id

into anim\_id, request\_by\_id

from request

where request\_id = req\_id;

select nvl(o\_id,c\_id)

into owner\_id

from animal

where a\_id = anim\_id;

update request

set approval\_status='Y'

where request\_id = req\_id;

if (sql%found) then

update animal

set o\_id = null, c\_id = request\_by\_id

where a\_id = anim\_id;

if(sql%found) then

update adoption\_history

set adopter\_id = request\_by\_id

where pet\_id = anim\_id

and (org\_id = owner\_id or cust\_id = owner\_id);

if(sql%found) then

insert into adoption\_history values(null,request\_by\_id,to\_date(sysdate,'dd-mon-yyyy'),null,anim\_id);

commit;

end if;

end if;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('No such request available');

end;

-------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

enter\_pet\_medical\_history(anim\_id integer, vacc\_type varchar, vacc\_date varchar, disease varchar)

is

animal\_id integer(5);

begin

select count(a\_id)

into animal\_id

from animal

where a\_id = anim\_id;

if(animal\_id>0) then

insert into medical\_history values(vacc\_type,to\_date(vacc\_date,'dd-mon-yyyy'),disease,anim\_id);

else

dbms\_output.put\_line('Cannot find a pet with the mentioned id.');

end if;

commit;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

----------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

register\_new\_customer(firstname varchar,

lastname varchar,

email varchar,

phone number,

cus\_zip number,

cus\_city varchar,

cus\_state varchar)

is

z integer;

begin

begin

select zip into z

from address

where zip = cus\_zip;

exception

when no\_data\_found then

insert into address values(cus\_zip,cus\_city,cus\_state);

when too\_many\_rows then

dbms\_output.put\_line('Inconsistancy in table address');

end;

select max(c\_id) into z

from customer;

insert into customer values(z+1,firstname,lastname,email,phone,cus\_zip);

commit;

dbms\_output.put\_line('New Customer registered. Registration ID is : '||(z+1));

exception

when others then

dbms\_output.put\_line('Error while executing procedure: register\_new\_customer');

dbms\_output.put\_line(SQLERRM);

end;

-------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

register\_new\_pet(your\_id integer,

type\_of\_cust varchar, --organization , --customer

age integer,

color varchar,

gender varchar,

breed number)

is

no\_such\_customer exception;

no\_such\_breed exception;

max\_id integer(5);

begin

if (upper(substr(type\_of\_cust,0,1)) = 'O') then

select o\_id

into max\_id

from organization

where o\_id = your\_id;

elsif (upper(substr(type\_of\_cust,0,1)) = 'C') then

select c\_id

into max\_id

from customer

where c\_id = your\_id;

else

raise no\_such\_customer;

end if;

select count(b\_id)

into max\_id

from breed

where b\_id = breed;

if(max\_id = 0) then

raise no\_such\_breed;

end if;

select max(a\_id)

into max\_id

from animal;

if (upper(substr(type\_of\_cust,0,1)) = 'O') then

insert into animal values(max\_id+1, age, color, gender, your\_id, null , breed);

insert into adoption\_history values (your\_id,null,to\_date(sysdate,'dd-MON-yyyy'),null,max\_id+1);

else

insert into animal values(max\_id+1, age, color, gender, null, your\_id, breed);

insert into adoption\_history values (null, your\_id,to\_date(sysdate,'dd-MON-yyyy'),null,max\_id+1);

end if;

dbms\_output.put\_line(' You pet has been registered. Pet id : '||(max\_id+1));

commit;

exception

when no\_such\_customer then

dbms\_output.put\_line('No such customer type exist.');

when no\_such\_breed then

dbms\_output.put\_line('We dont support any such kind of Breed.');

dbms\_output.put\_line('Please revisit the breeds we list');

when no\_data\_found then

dbms\_output.put\_line('Customer record not found.');

dbms\_output.put\_line('Please register your self and then put your animal for donation');

when others then

rollback;

dbms\_output.put\_line(SQLERRM);

end;

-----------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_animal\_adoption\_history( anim\_id in integer )

is

cursor adoption\_data(a\_id integer) is

select nvl(A.ORG\_ID,A.CUST\_ID) donor\_id,

a.adoption\_date adoption\_date,

C1.C\_FNAME||' '||C1.C\_LNAME adopter\_name

from adoption\_history a, customer c1

where A.ADOPTER\_ID = C1.C\_ID

and A.PET\_ID=a\_id

order by adoption\_date;

adoption\_rec adoption\_data%rowtype;

donor\_fname varchar(30);

donor\_lname varchar(30);

animal\_age integer(3);

animal\_color varchar(20);

animal\_gender varchar(8);

begin

open adoption\_data(anim\_id);

fetch adoption\_data into adoption\_rec;

if(adoption\_data%notfound) then

dbms\_output.put\_line('Adoption history for the pet does not exist.');

else

select a\_age, a\_color, decode(upper(a\_gender),'M','Male','F','Female','na')

into animal\_age, animal\_color, animal\_gender

from animal

where a\_id = anim\_id;

dbms\_output.put\_line('Age :'||animal\_age||' yrs');

dbms\_output.put\_line('Color :'||animal\_color);

dbms\_output.put\_line('Gender :'||animal\_gender);

dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------');

dbms\_output.put\_line(rpad(' Owner name', 50)||rpad('Handover date', 25)||rpad('Currently with', 50));

dbms\_output.put\_line('----------------------------------------------------------------------------------------------------------------');

loop

donor\_fname := null;

donor\_lname := null;

begin

select o\_name into donor\_fname

from organization

where o\_id = adoption\_rec.donor\_id;

exception

when no\_data\_found then

begin

select c\_fname, c\_lname

into donor\_fname, donor\_lname

from customer

where c\_id = adoption\_rec.donor\_id;

exception

when others then

dbms\_output.put\_line('Error fetching data from customer');

dbms\_output.put\_line(SQLERRM);

end;

when others then

dbms\_output.put\_line('Error fetching data from organization');

dbms\_output.put\_line(SQLERRM);

end;

dbms\_output.put\_line(rpad(donor\_fname||' '||donor\_lname, 50)||

rpad(to\_char(adoption\_rec.adoption\_date,'MON DD YYYY'), 25)||

rpad(adoption\_rec.adopter\_name, 50));

fetch adoption\_data into adoption\_rec;

exit when (adoption\_data%notfound);

end loop;

end if;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

-------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_animal\_medical\_history( anim\_id in integer)

is

vacc\_rec medical\_history%rowtype;

cursor vacc\_data is

select \*

from medical\_history

where animal\_id = anim\_id

order by VACCINATION\_DATE;

begin

open vacc\_data;

fetch vacc\_data into vacc\_rec;

if(vacc\_data%notfound) then

dbms\_output.put\_line('Pets Medical record does not exist.');

else

dbms\_output.put\_line('-------------------------------------------------------------------');

dbms\_output.put\_line(rpad(' Vaccination type', 25)||rpad('Vaccination data', 25)||rpad('Disease', 20));

dbms\_output.put\_line('-------------------------------------------------------------------');

loop

dbms\_output.put\_line(rpad(vacc\_rec.vaccination\_type, 25)||

rpad(to\_char(vacc\_rec.vaccination\_date,'MON DD YYYY'), 25)||

rpad(vacc\_rec.disease, 20));

fetch vacc\_data into vacc\_rec;

exit when (vacc\_data%notfound);

end loop;

end if;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

--------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_animals\_for\_a\_breed (breed integer)

is

animal\_rec animal%rowtype;

cursor animal\_data is

select \*

from animal

where breed\_id = breed

and a\_id not in (

select animal\_id

from request

where approval\_status <> 'Y')

order by a\_id;

begin

open animal\_data;

fetch animal\_data into animal\_rec;

if(animal\_data%notfound) then

dbms\_output.put\_line(' No records available for this breed. Try a different breed. ');

else

dbms\_output.put\_line('-------------------------------------------------------------------');

dbms\_output.put\_line(rpad(' Animal Id', 15)||rpad('Age', 15)||rpad('Color', 20)||rpad('Gender', 10));

dbms\_output.put\_line('-------------------------------------------------------------------');

loop

exit when (animal\_data%notfound);

dbms\_output.put\_line(rpad(to\_char(animal\_rec.a\_id), 15)||rpad(to\_char(animal\_rec.a\_age), 15)||

rpad(animal\_rec.a\_color, 20)||rpad(animal\_rec.a\_gender, 10));

fetch animal\_data into animal\_rec;

end loop;

close animal\_data;

end if;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

-----------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_approval\_pending\_on\_me(cust\_id integer)

is

cursor my\_request(id integer) is

select r.request\_id,

b.b\_animal,

a.a\_age,

a\_color,

a.a\_gender,

decode(r.approval\_status,'P','Pending','-') approval\_status

from request r, animal a, breed b

where a.a\_id = r.animal\_id

and r.approval\_status = 'P'

and a.breed\_id = b.b\_id and a.o\_id = id or a.c\_id = id;

request\_rec my\_request%rowtype;

begin

open my\_request(cust\_id);

fetch my\_request into request\_rec;

if (my\_request%found) then

dbms\_output.put\_line('--------------------------------------------------------------------------------------------');

dbms\_output.put\_line(rpad(' Request Id', 15)||rpad('Animal', 15)||rpad('Animal Age', 15)||rpad('Color', 25)||rpad('Gender', 10)||rpad('Approval Status', 20));

dbms\_output.put\_line('--------------------------------------------------------------------------------------------');

loop

dbms\_output.put\_line(' '||rpad(to\_char(request\_rec.request\_id),15)||

rpad(request\_rec.b\_animal,15)||

rpad(to\_char(request\_rec.a\_age),15)||

rpad(request\_rec.a\_color,25)||

rpad(request\_rec.a\_gender,10)||

rpad(request\_rec.approval\_status,15));

fetch my\_request into request\_rec;

exit when (my\_request%notfound);

end loop;

else

dbms\_output.put\_line('No pending approval request exists for you');

end if;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

------------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_available\_breeds

is

breed\_rec breed%rowtype;

cursor all\_breeds is

select \*

from breed

order by b\_id;

begin

open all\_breeds;

dbms\_output.put\_line('-----------------------------------------------------');

dbms\_output.put\_line(rpad(' Breed Id', 15)||rpad('Animal', 15)||rpad('Breed', 20));

dbms\_output.put\_line('-----------------------------------------------------');

loop

fetch all\_breeds into breed\_rec;

exit when (all\_breeds%notfound);

dbms\_output.put\_line(rpad(to\_char(breed\_rec.b\_id), 15)||rpad(to\_char(breed\_rec.b\_animal), 15)||rpad(breed\_rec.b\_breed, 20));

end loop;

close all\_breeds;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

-----------------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_details\_of\_customer(cust\_id in integer)

is

fname varchar(30) :=null;

lname varchar(30) := null;

email varchar(30);

phone integer(10);

zipcode integer(6);

city varchar(30);

state varchar(30);

begin

begin

select c\_fname, c\_lname, c\_email, c\_phone, c\_zip

into fname, lname, email, phone, zipcode

from customer

where c\_id = cust\_id;

exception

when no\_data\_found then

begin

select o\_name, o\_email, o\_phone, o\_zip

into fname, email, phone, zipcode

from organization

where o\_id = cust\_id;

end;

end;

select address.city, address.state

into city, state

from address

where zip = zipcode;

dbms\_output.put\_line('---------------------------------------------');

dbms\_output.put\_line(' Details ');

dbms\_output.put\_line('---------------------------------------------');

dbms\_output.put\_line(' Id :'||cust\_id);

if (lname is null) then

dbms\_output.put\_line(' Type :Organization');

dbms\_output.put\_line(' Name :'||fname);

else

dbms\_output.put\_line(' Type :Customer');

dbms\_output.put\_line(' First Name :'||fname);

dbms\_output.put\_line(' Last Name :'||lname);

end if;

dbms\_output.put\_line(' Email :'||email);

dbms\_output.put\_line(' Phone :'||phone);

dbms\_output.put\_line(' Zip :'||zipcode);

dbms\_output.put\_line(' City :'||city);

dbms\_output.put\_line(' State :'||state);

exception

when no\_data\_found then

dbms\_output.put\_line('No record exists for the customer with id :'||cust\_id);

when others then

dbms\_output.put\_line(SQLERRM);

end;

----------------------------------------------------------------------------------------------------------------------------

create or replace procedure

show\_my\_request\_info(cust\_id integer)

is

cursor my\_request(id integer) is

select r.request\_id,

C.C\_FNAME,

C.C\_LNAME,

b.b\_animal,

a.a\_age,

a\_color,

decode(a.a\_gender,'F','Female','M','Male','-') a\_gender,

decode(r.approval\_status,'P','Pending','Y','Approved','N','Rejected','-') approval\_status

from request r, animal a, breed b, customer c

where a.a\_id = r.animal\_id

and a.breed\_id = b.b\_id

and C.C\_ID = R.REQUESTER\_ID

and C.C\_ID = id;

request\_rec my\_request%rowtype;

begin

open my\_request(cust\_id);

fetch my\_request into request\_rec;

if (my\_request%found) then

dbms\_output.put\_line('---------------------------------------------------------------------------------------------------------------------------------------');

dbms\_output.put\_line(rpad(' Request Id', 15)||rpad('First Name', 25)||rpad('Last Name', 25)||rpad('Animal', 15)||rpad('Animal Age', 15)||rpad('Color', 15)||rpad('Gender', 10)||rpad('Approval Status', 20));

dbms\_output.put\_line('---------------------------------------------------------------------------------------------------------------------------------------');

loop

dbms\_output.put\_line(' '||rpad(to\_char(request\_rec.request\_id),15)||

rpad(request\_rec.c\_fname,25)||

rpad(request\_rec.c\_lname,25)||

rpad(request\_rec.b\_animal,15)||

rpad(to\_char(request\_rec.a\_age),15)||

rpad(request\_rec.a\_color,15)||

rpad(request\_rec.a\_gender,10)||

rpad(request\_rec.approval\_status,15));

fetch my\_request into request\_rec;

exit when (my\_request%notfound);

end loop;

else

dbms\_output.put\_line('There is no request mada by you till present.');

end if;

exception

when others then

dbms\_output.put\_line(SQLERRM);

end;

create or replace procedure

register\_new\_organization(Org\_name varchar,

email varchar,

phone number,

Org\_zip number,

Org\_city varchar,

Org\_state varchar)

is

z integer;

begin

begin

select zip into z

from address

where zip = Org\_zip;

exception

when no\_data\_found then

insert into address values(Org\_zip,Org\_city,Org\_state);

when too\_many\_rows then

dbms\_output.put\_line('Inconsistancy in table address');

end;

select max(o\_id) into z

from organization;

insert into organization values(z+1,Org\_name,email,phone,Org\_zip);

commit;

dbms\_output.put\_line('New Organization registered. Registration ID is : '||(z+1));

exception

when others then

dbms\_output.put\_line('Error while executing procedure: register\_new\_organization');

dbms\_output.put\_line(SQLERRM);

rollback;

end;