Food Bank Management System

Semester - IV

Database Management System(CSE-250)

Group - 44

Roll No.	Name	Program
AU1940067	Kathan Joshi	Btech CSE
AU1940190	Devanshu Magiawala	Btech CSE
AU1940205	Henil Shah	Btech CSE

Table of Content:

1. Acknowledgment:

Tenacious motivation and inspiration have consistently assumed a critical part in the accomplishment of any understudy. This task has become a reality with the benevolent help and help of numerous people. We might want to stretch out our sincerest appreciation to every one of them. We might want to offer our most prominent thanks to Professor, Shefali Naik who urged us to the most noteworthy pinnacle, furnished us with a chance to chip away at this task, and guided us towards the finishing of the undertaking. We might likewise want to offer our most profound thanks to saying thanks to Ahmedabad University for giving us the Database Management course to assist us with investigating skylines in this field. We offer our thanks to our companions for their caring participation and support which help us in the finish of this venture.

2. Description of Project:

- ➔ During the past weeks of this pandemic, food banks across the country are doing what they do best - feeding people in need within their communities. In this time of uncertainty, feeding those who are vulnerable to ensure that the pandemic doesn’t provide inequality to receive food due to colour or income. In this project, we are trying to build an application with Mysql and Flask for management of a specific food bank location.
- ➔ The main motive to design such an application/website is to help them maintain the records perfectly and to have all the necessary information regarding food bank, so that they can manage well and effectively.
- ➔ The project contains a total of 10 main tables. Following is the list of tables that are created:
  - ➔ Employee

    This table basically has the data of all the employees that are working in the organization.
  - ➔ Volunteer

This table has the data of all the volunteers that have been working for the organization.

- ➔ Donar

  This contains all the donors who have donated food
- ➔ Donee

  This data table consists of data about donees
- ➔ Donation

  This table contains the food items data which a particular donor has donated
- ➔ Donated

  This represents the data that has been for the food items that are given to a particular donee
- ➔ Food Bank

  This is the whole bank of data about the food item stock that is maintained
- ➔ Grain

  This is the data about the grains that are present in the food bank
- ➔ Pulses

  This is the data about the pulses that are present in the food bank
- ➔ Produce

  This is the data about the produced items that are present in the food bank

- So, firstly to make this system work an employee or a volunteer needs to login with their username and password to access the further functionalities.
- For an employee, there are different sets of functionalities and for volunteers the system is restricted. Employees can manage the list of volunteers, donors, food bank, grain, pulses ,produced items while volunteers can only access the donee and the food bank.
- If some donor donates food, then the food bank will automatically increase the quantity if its an item that already exists and if not then it will add one.
- Similarly, it will automatically decrease the quantity of food items when a food item is given to donee.
- Also, donors can donate any number of items in food while the donee has some restriction for the food items.

3. Entity-Relationship Diagram:

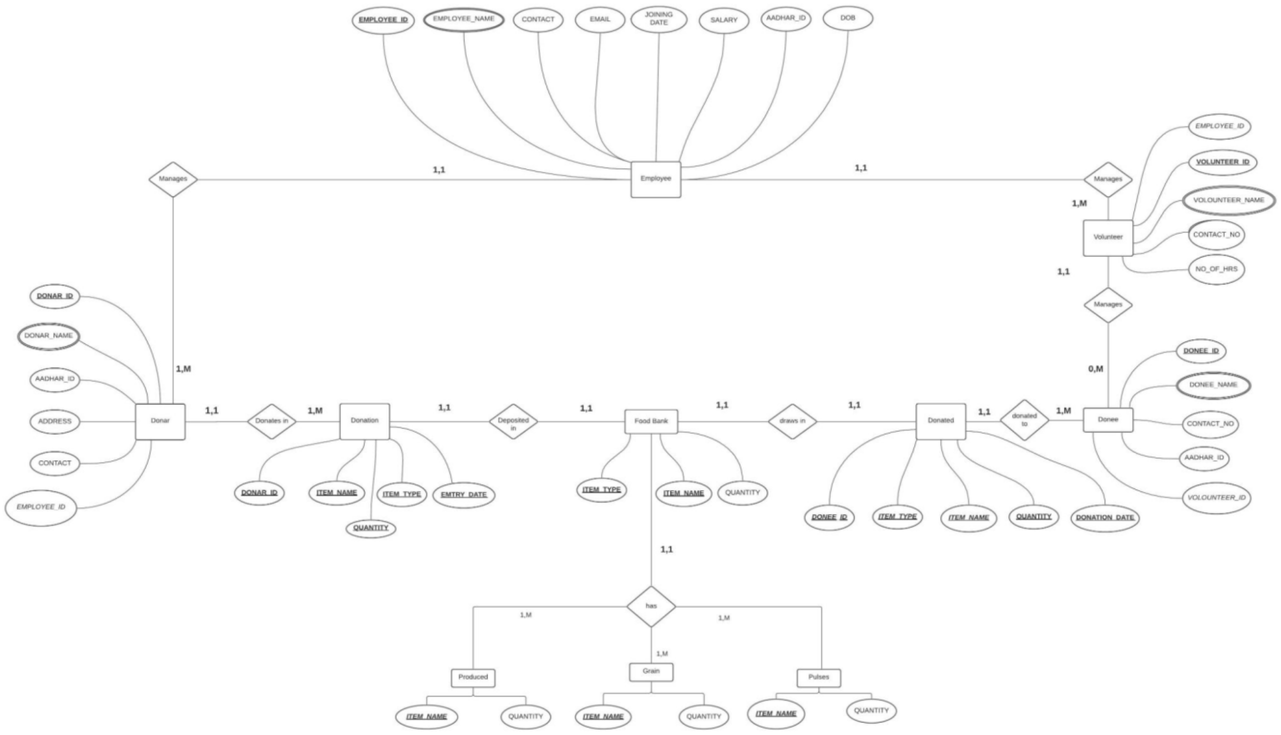
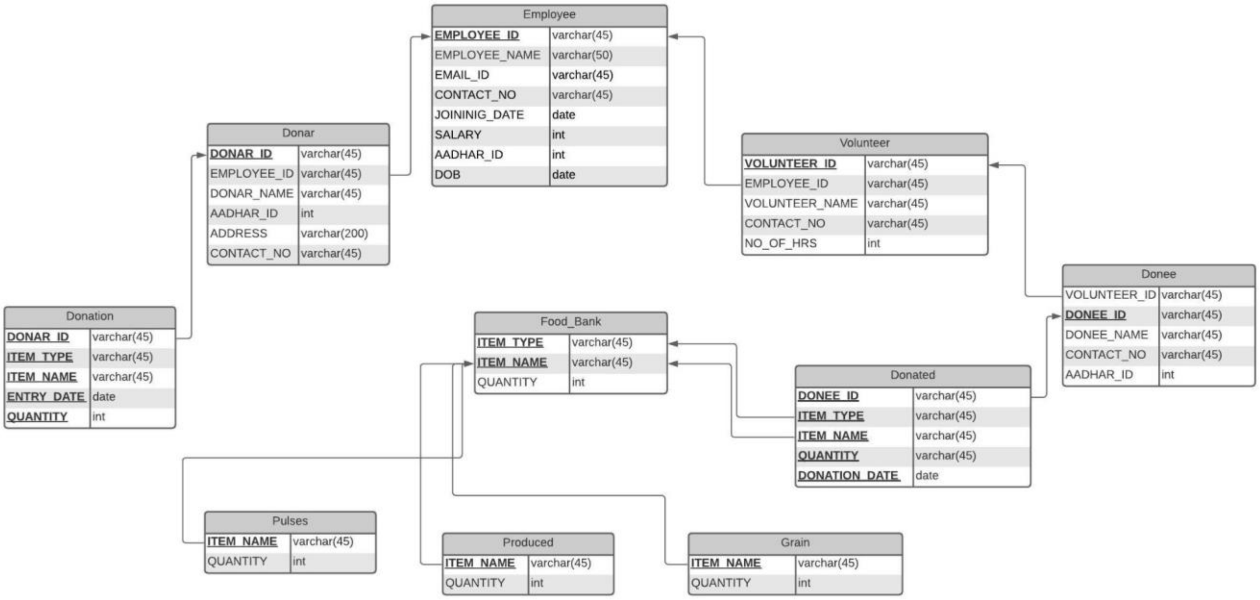


Figure 1

Please refer the following link for better readability:

https://lucid.app/lucidchart/invitations/accept/inv_f1d78b89-7364-484d-8b70-e9def7c96ce0?viewport_loc=-630%2C-517%2C4044%2C1851%2C0_0

4. Table Design (Data Dictionary):



Employee	
EMPLOYEE_ID	varchar(45)
EMPLOYEE_NAME	varchar(50)
EMAIL_ID	varchar(45)
CONTACT_NO	varchar(45)
JOININIG_DATE	date
SALARY	int
AADHAR_ID	int
DOB	date

Donar	
DONAR_ID	varchar(45)
EMPLOYEE_ID	varchar(45)
DONAR_NAME	varchar(45)
AADHAR_ID	int
ADDRESS	varchar(200)
CONTACT_NO	varchar(45)

Volunteer	
VOLUNTEER_ID	varchar(45)
EMPLOYEE_ID	varchar(45)
VOLUNTEER_NAME	varchar(45)
CONTACT_NO	varchar(45)
NO_OF_HRS	int

Donee	
VOLUNTEER_ID	varchar(45)
DONEE_ID	varchar(45)
DONEE_NAME	varchar(45)
CONTACT_NO	varchar(45)
AADHAR_ID	int

Donation	
DONAR_ID	varchar(45)
ITEM_TYPE	varchar(45)
ITEM_NAME	varchar(45)
ENTRY_DATE	date
QUANTITY	int

Food_Bank	
ITEM_TYPE	varchar(45)
ITEM_NAME	varchar(45)
QUANTITY	int

Donated	
DONEE_ID	varchar(45)
ITEM_TYPE	varchar(45)
ITEM_NAME	varchar(45)
QUANTITY	varchar(45)
DONATION_DATE	date

Pulses	
ITEM_NAME	varchar(45)
QUANTITY	int

Produced	
ITEM_NAME	varchar(45)
QUANTITY	int

Grain	
ITEM_NAME	varchar(45)
QUANTITY	int

Figure 2

Please refer the following link for better readability:https://lucid.app/lucidchart/invitations/accept/inv_661b24b5-9995-4edc-9b88-a05108c3cdaf?viewport_loc=-251%2C240%2C2447%2C1120%2C0_0

5. Stored Procedure, Functions, and Triggers:

a. Procedures:

i.

```
DELIMITER //
create procedure name_search_donar(name varchar(45))
begin
select * from donar where donar_name like concat('%', name,'%');
end //
```

```
DELIMITER ;
```

Calling Function:

```
cur.execute('call name_search_donar(%s);',donar_name)
```

ii.

```
DELIMITER //
create procedure name_search_donee(name varchar(45))
begin
select * from donee where donee_name like concat('%', name,'%');
end //
DELIMITER ;
```

Calling Function:

```
cur.execute('call name_search_donee(%s);',donee_name)
```

iii.

```
DELIMITER //
create procedure city_search_donar(city varchar(45))
begin
select * from donar where address like concat('%', city,'%');
end //
DELIMITER ;
```

Calling Function:

```
cur.execute('call city_search_donar(%s);',donar_name)
```

iv.

```
DELIMITER //
create procedure disp_volunteers_under_employee(name
varchar(45))
begin
select * from volunteer where employee_id in (select employee_id
from employee where employee_name like concat('%', name ,'%'));
end //
```

```
DELIMITER ;
```

Calling Function:

cur.execute('call disp_volunteers_under
_employee(%s);',donar_name)

v.

```
DELIMITER //
create procedure donation_date_details(dt date)
begin
select * from donation where entry_date = dt;
end ;
//
DELIMITER ;
```

Calling Function:

cur.execute('call donation_date_details(%s);',dt)

vi.

```
DELIMITER //
CREATE PROCEDURE donated_date_details(dt date)
begin
select * from donated where donation_date = dt;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('call donated_date_details(%s);',dt)

vii.

```
DELIMITER //
CREATE PROCEDURE calc_new_salary()
BEGIN
```

```
update employee set SALARY=SALARY+(0.1*SALARY) where
TIMESTAMPDIFF(YEAR, JOINING_DATE, CURDATE())>16;
select * from employee where TIMESTAMPDIFF(YEAR,
JOINING_DATE, CURDATE())>16;
END;
//
DELIMITER ;
```

Calling Function:

cur.execute('call calc_new_salary();')

b. Functions:

i.

```
DELIMITER //
create function max_donation ()
returns varchar(45)
deterministic
begin
                set @name1 = (select donar_name from donar where
        donar_id = (select DONAR_ID from donation group by
        DONAR_ID order by sum(quantity) desc limit 1));
   return @name1;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select max_donation();')

ii.

```
DELIMITER //
create function sum_quantity ()
returns int
deterministic
begin
set @cnt := (select sum(quantity) from food_bank);
```

```
return @cnt;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select sum_quality();')

iii.

```
DELIMITER //
create function donated_count (dt date)
returns int
deterministic
begin
set @cnt := (select count(*) from donated where
DONATION_DATE=dt);
return @cnt;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select donated_count(%s);',date)

iv.

```
DELIMITER //
create function donation_count (dt date)
returns int
deterministic
begin
set @cnt := (select count(*) from donation where
ENTRY_DATE=dt);
return @cnt;
```

```
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select donation_count(%s);',date)

v.

```
DELIMITER //
create function donor_count (emp_id varchar(20))
returns int
deterministic
begin
set @cnt := (select count(*) from donar where
EMPLOYEE_ID=emp_id);
return @cnt;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select donar_count(%s);',employee_id)

vi.

```
DELIMITER //
create function max_time_donar ()
returns varchar(45)
deterministic
```

```
begin
set @name1 = (select DONEE_NAME from donee where
DONEE_ID = (select DONEE_ID from donated group by
DONEE_ID order by count(DONEE_ID) desc limit 1));
    return @name1;
end;
//
DELIMITER ;
```

Calling Function:

cur.execute('select max_time_donar();')

c. **Triggers:**

i.

```
DELIMITER //
create trigger grain_pulses_produced after insert on food_bank
for each row
begin
          if new.ITEM_TYPE='GRAIN' then
                    insert into grain
values(new.ITEM_NAME,new.QUANTITY);
          end if;
    if new.ITEM_TYPE='PULSES' then
                    insert into pulses
values(new.ITEM_NAME,new.QUANTITY);
          end if;

    if new.ITEM_TYPE='PRODUCED' then
                    insert into produced
values(new.ITEM_NAME,new.QUANTITY);
```

```
        end if;
end;
//
DELIMITER ;
```

ii.

```
DELIMITER //
create trigger food_bank_update_insert after insert on donation
for each row
begin
        declare done int;
    declare temp int;
    declare c_food_bank cursor for select QUANTITY from
food_bank where ITEM_TYPE=new.ITEM_TYPE and
ITEM_NAME=new.ITEM_NAME;
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET
done = 1;
    open c_food_bank;
    fetch c_food_bank into temp;
    if done then
                insert into food_bank
values(new.ITEM_TYPE,new.ITEM_NAME,new.QUANTITY);
    else
                update food_bank set
QUANTITY=new.QUANTITY+temp where
ITEM_TYPE=new.ITEM_TYPE and
ITEM_NAME=new.ITEM_NAME;
                if(new.Item_TYPE='GRAIN') then
                        update grain set
QUANTITY=new.QUANTITY+temp where
ITEM_NAME=new.ITEM_NAME;
                end if;
        if(new.Item_TYPE='PULSES') then
                        update pulses set
QUANTITY=new.QUANTITY+temp where
ITEM_NAME=new.ITEM_NAME;
                end if;
        if(new.Item_TYPE='PRODUCED') then
                        update produced set
QUANTITY=new.QUANTITY+temp where
ITEM_NAME=new.ITEM_NAME;
        end if;
    end if;
    close c_food_bank;
end;
//
DELIMITER ;
```

iii.

```
DELIMITER //
create trigger food_bank_update_delete after insert on donated
for each row
begin
          declare done int default false;
    declare temp int;
    declare c_food_bank cursor for select QUANTITY from
food_bank where ITEM_TYPE=new.ITEM_TYPE and
ITEM_NAME=new.ITEM_NAME;
          DECLARE CONTINUE HANDLER FOR NOT FOUND SET
done = true;
    open c_food_bank;
    fetch c_food_bank into temp;
    if not done then
                update food_bank set QUANTITY=temp-
new.QUANTITY where ITEM_TYPE=new.ITEM_TYPE and
ITEM_NAME=new.ITEM_NAME;
                if(new.Item_TYPE='GRAIN') then
```

```
            update grain set QUANTITY=temp-
new.QUANTITY where ITEM_NAME=new.ITEM_NAME;
        end if;
   if(new.Item_TYPE='PULSES') then
            update pulses set QUANTITY=temp-
new.QUANTITY where ITEM_NAME=new.ITEM_NAME;
        end if;
   if(new.Item_TYPE='PRODUCED') then
            update produced set QUANTITY=temp-
new.QUANTITY where ITEM_NAME=new.ITEM_NAME;
   end if;
  end if;
  close c_food_bank;
end;
//
DELIMITER ;
```

iv.

```
DELIMITER //
create trigger validate_volunteer before insert on volunteer
for each row
begin
    if new.CONTACT_NO < 0 then
        SIGNAL SQLSTATE
        '45020' SET MESSAGE_TEXT = 'Contact Number
can''t be negative';
    end if;

  if new.NO_OF_HRS < 0 then
        SIGNAL SQLSTATE
        '45050' SET MESSAGE_TEXT = 'Number of Hours
worked can''t be negative';
    end if;
end;
//
DELIMITER ;
```

v.

```
DELIMITER //
create trigger validate_employee before insert on employee
```

```
for each row
begin
    if new.CONTACT_NO < 0 then
        SIGNAL SQLSTATE
        '45020' SET MESSAGE_TEXT = 'Contact Number
can''t be negative';
    end if;

  if new.CONTACT_NO>0 then
        if length(new.CONTACT_NO) <> 10 then
            SIGNAL SQLSTATE
            '45025' SET MESSAGE_TEXT = 'Contact
Number can''t be less than 10 digits';
        end if;
    end if;

  if new.SALARY < 0 then
        SIGNAL SQLSTATE
        '45030' SET MESSAGE_TEXT = 'Salary can''t be
negative';
    end if;

  if new.AADHAR_ID < 0 then
        SIGNAL SQLSTATE
        '45035' SET MESSAGE_TEXT = 'Aadhar ID can''t be
negative';
    end if;

  if TIMESTAMPDIFF(YEAR, new.DOB, new.JOINING_DATE)<18
then
        SIGNAL SQLSTATE
            '45040' SET MESSAGE_TEXT = 'Employee is
under aged to work';
    end if;
end;
//
DELIMITER ;
```

vi. DELIMITER //

```
create trigger validate_donar before insert on donar
for each row
begin
    if new.CONTACT_NO < 0 then
        SIGNAL SQLSTATE
        '45020' SET MESSAGE_TEXT = 'Contact Number
can''t be negative';
    end if;
  if new.AADHAR_ID < 0 then
        SIGNAL SQLSTATE
        '45035' SET MESSAGE_TEXT = 'Aadhar ID can''t be
negative';
    end if;
end;
//
DELIMITER ;
```

vii.

```
DELIMITER //
create trigger validate_donee before insert on donee
for each row
begin
    if new.CONTACT_NO < 0 then
        SIGNAL SQLSTATE
        '45020' SET MESSAGE_TEXT = 'Contact Number
can''t be negative';
    end if;

  if new.CONTACT_NO>0 then
        if length(new.CONTACT_NO) <> 10 then
            SIGNAL SQLSTATE
            '45025' SET MESSAGE_TEXT = 'Contact
Number can''t be less than 10 digits';
        end if;
    end if;

  if new.AADHAR_ID < 0 then
        SIGNAL SQLSTATE
        '45035' SET MESSAGE_TEXT = 'Aadhar ID can''t be
negative';
```

```
    end if;
end;
//

DELIMITER ;
```

viii.

```
DELIMITER//
create trigger qty_check before insert on donated
for each row
begin
    set @qty := (select QUANTITY from food_bank where
ITEM_NAME=new.ITEM_NAME and
ITEM_TYPE=new.ITEM_TYPE);
    if new.QUANTITY>0.1*@qty then
        SIGNAL SQLSTATE
        '45000' SET MESSAGE_TEXT = 'You cannot take out
more than 10% from the food bank';
    end if;
  if new.QUANTITY<0 then
        SIGNAL SQLSTATE
        '45010' SET MESSAGE_TEXT = 'Quantity cannot be
negative';
    end if;
end;
//
DELIMITER ;
```

ix.

```
DELIMITER //
create trigger qty_chk before insert on donation
for each row
```

```
begin
   if new.QUANTITY < 0 then
x              SIGNAL SQLSTATE
               '45010' SET MESSAGE_TEXT = 'Quantity cannot be
negative';
        end if;
end;
//
DELIMITER ;
```

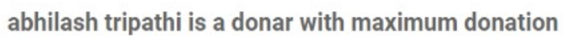
x.

```
DELIMITER //
create trigger date_donee_contraint before insert on donated
for each row
begin
        set @var_date:=(select DONEE_ID from donated where
DONEE_ID = new.DONEE_ID and
DONATION_DATE=new.DONATION_DATE);
   if new.DONEE_ID = @var_date then
               SIGNAL SQLSTATE
               '45020' SET MESSAGE_TEXT = 'Same Donee
cannot take donation more than once';
        end if;
end;
//
DELIMITER ;
```

6. **Screenshots of results generated after procedure and function are called on the front-end:**

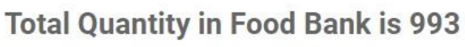
Functions:

I. **max_donation()**



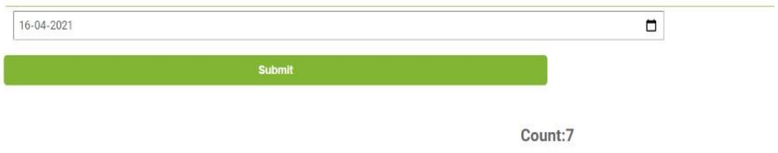
abhilash tripathi is a donar with maximum donation

II. **sum_quantity()**



Total Quantity in Food Bank is 993

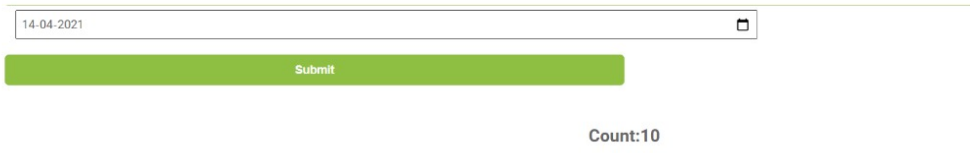
III. **donated_count(dt date)**



16-04-2021

Count:7

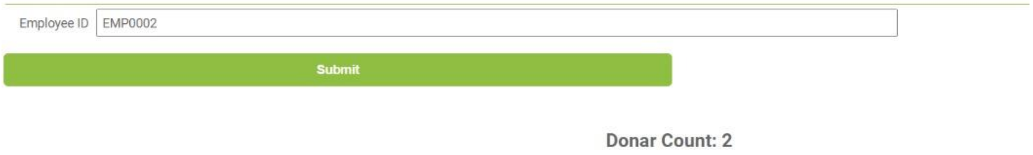
IV. **donation_count(dt date)**



14-04-2021

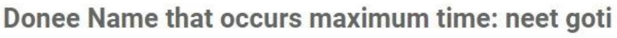
Count:10

V. **donar_count(employee_id)**



Donar Count: 2

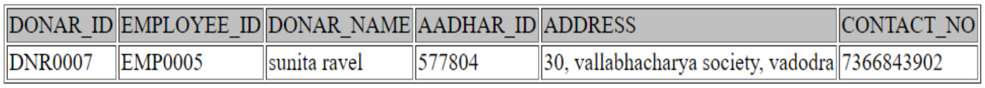
VI. max_time_donor()



Donee Name that occurs maximum time: neet goti

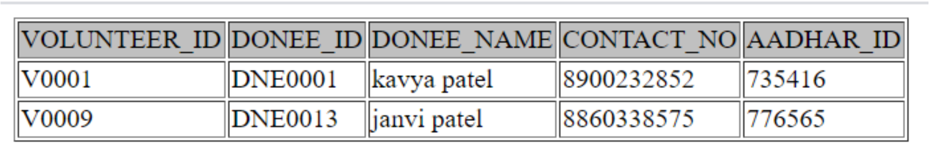
Procedure:

I. name_search_donar(name varchar(45))



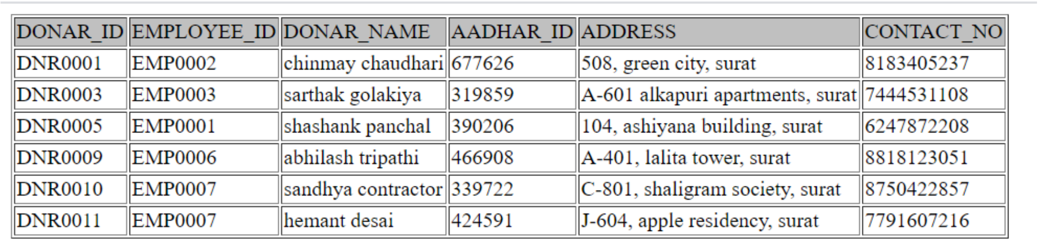
DONAR_ID	EMPLOYEE_ID	DONAR_NAME	AADHAR_ID	ADDRESS	CONTACT_NO
DNR0007	EMP0005	sunita ravel	577804	30, vallabhacharya society, vadodra	7366843902

II. name_search_donee(name varchar(45))



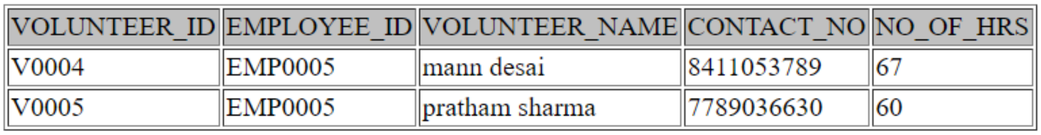
VOLUNTEER_ID	DONEE_ID	DONEE_NAME	CONTACT_NO	AADHAR_ID
V0001	DNE0001	kavya patel	8900232852	735416
V0009	DNE0013	janvi patel	8860338575	776565

III. city_search_donar(city varchar(45



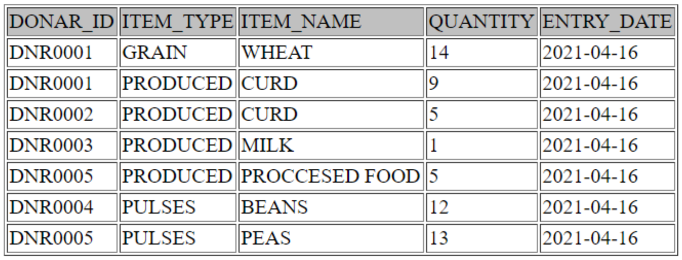
DONAR_ID	EMPLOYEE_ID	DONAR_NAME	AADHAR_ID	ADDRESS	CONTACT_NO
DNR0001	EMP0002	chinmay chaudhari	677626	508, green city, surat	8183405237
DNR0003	EMP0003	sarthak golakiya	319859	A-601 alkapuri apartments, surat	7444531108
DNR0005	EMP0001	shashank panchal	390206	104, ashiyana building, surat	6247872208
DNR0009	EMP0006	abhilash tripathi	466908	A-401, lalita tower, surat	8818123051
DNR0010	EMP0007	sandhya contractor	339722	C-801, shaligram society, surat	8750422857
DNR0011	EMP0007	hemant desai	424591	J-604, apple residency, surat	7791607216

IV. disp_volunteers_under_employee(name varchar(45))



VOLUNTEER_ID	EMPLOYEE_ID	VOLUNTEER_NAME	CONTACT_NO	NO_OF_HRS
V0004	EMP0005	mann desai	8411053789	67
V0005	EMP0005	pratham sharma	7789036630	60

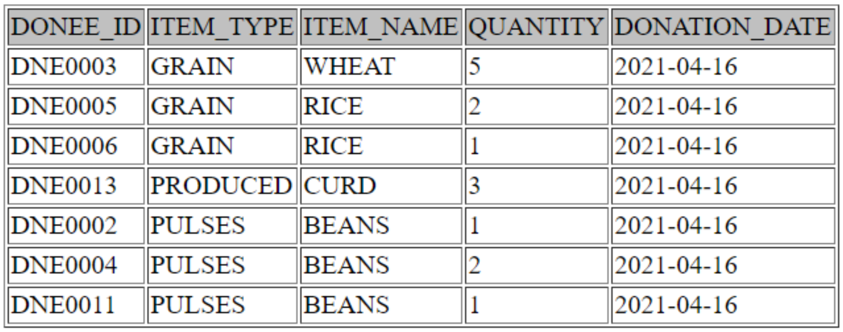
V. donation_date_details(dt dat



DONAR_ID	ITEM_TYPE	ITEM_NAME	QUANTITY	ENTRY_DATE
DNR0001	GRAIN	WHEAT	14	2021-04-16
DNR0001	PRODUCED	CURD	9	2021-04-16
DNR0002	PRODUCED	CURD	5	2021-04-16
DNR0003	PRODUCED	MILK	1	2021-04-16
DNR0005	PRODUCED	PROCCESED FOOD	5	2021-04-16
DNR0004	PULSES	BEANS	12	2021-04-16
DNR0005	PULSES	PEAS	13	2021-04-16

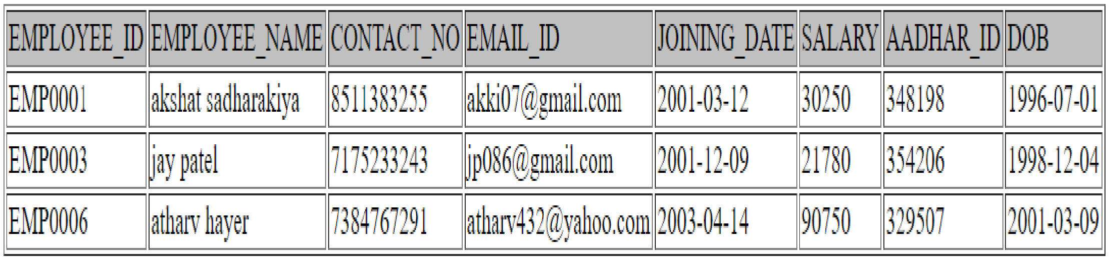
e)

VI. donated_date_details(dt date)



DONEE_ID	ITEM_TYPE	ITEM_NAME	QUANTITY	DONATION_DATE
DNE0003	GRAIN	WHEAT	5	2021-04-16
DNE0005	GRAIN	RICE	2	2021-04-16
DNE0006	GRAIN	RICE	1	2021-04-16
DNE0013	PRODUCED	CURD	3	2021-04-16
DNE0002	PULSES	BEANS	1	2021-04-16
DNE0004	PULSES	BEANS	2	2021-04-16
DNE0011	PULSES	BEANS	1	2021-04-16

VII. calc_new_salary()



EMPLOYEE_ID	EMPLOYEE_NAME	CONTACT_NO	EMAIL_ID	JOINING_DATE	SALARY	AADHAR_ID	DOB
EMP0001	akshat sadharakiya	8511383255	akki07@gmail.com	2001-03-12	30250	348198	1996-07-01
EMP0003	jay patel	7175233243	jp086@gmail.com	2001-12-09	21780	354206	1998-12-04
EMP0006	atharv hayer	7384767291	atharv432@yahoo.com	2003-04-14	90750	329507	2001-03-09

7. **Screenshots of errors generated on the front-end when the trigger is fired**

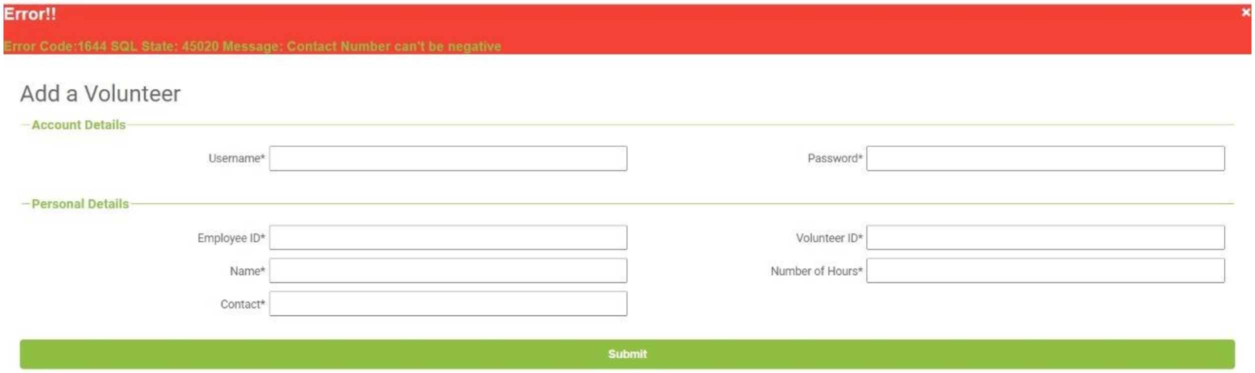
I. **grain_pulses_produced after insert on food_bank**

II. **food_bank_update_insert after insert on donation**

III. **food_bank_update_delete after insert on donated**

IV. **validate_volunteer before insert on volunteer:**

The trigger activates when: Contact number negative and number of hours negative



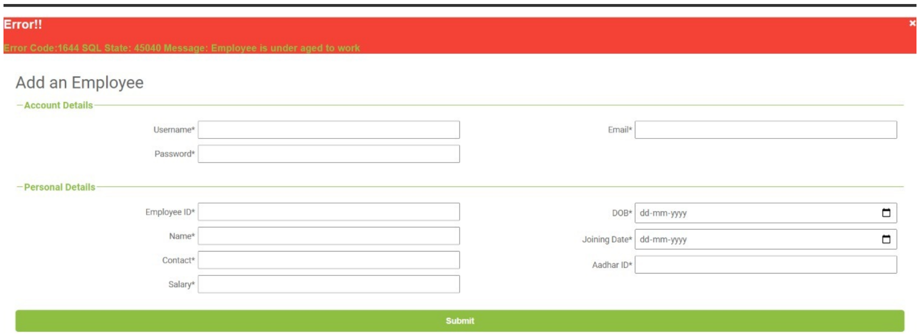
Error!!

Error Code:1644 SQL State: 45020 Message: Contact Number can't be negative

Add a Volunteer

V. **validate_employee before insert on employee:**

The trigger activates when: Contact number negative and less than 10 digits.

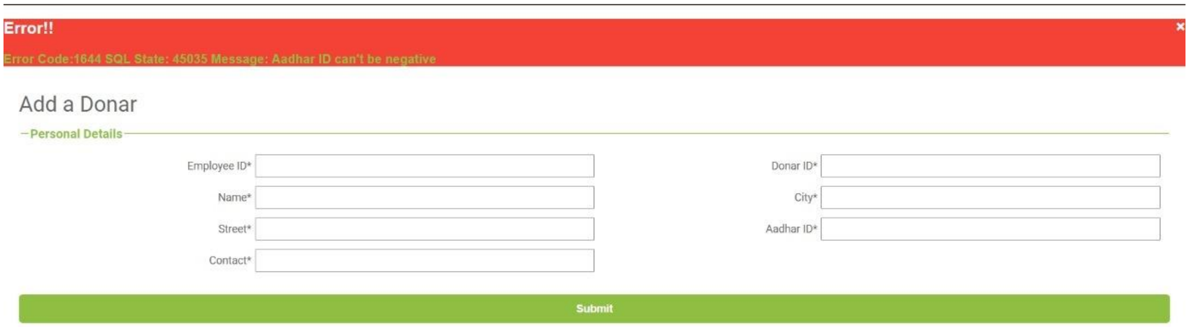


Error!!

Error Code:1644 SQL State: 45040 Message: Employee is under aged to work

VI. **validate_donar before insert on donar**

The trigger activates when: Contact number negative and Aadhar ID can’t be negative

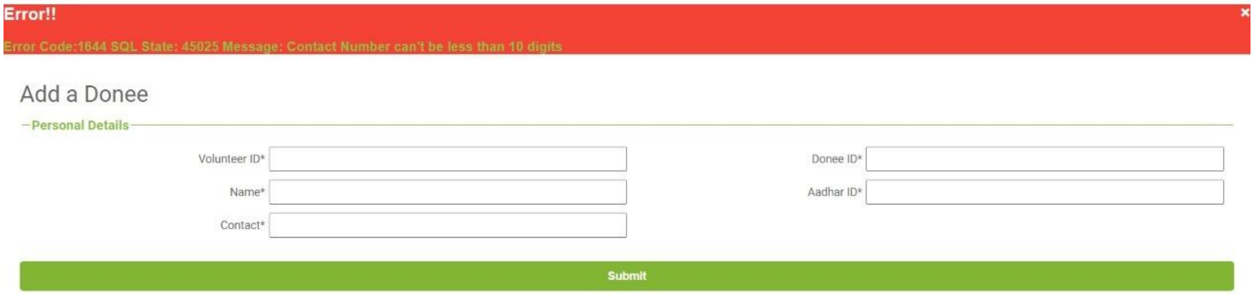


Error!!

Error Code:1644 SQL State: 45035 Message: Aadhar ID can't be negative

VII. **validate_donee before insert on donee**

The trigger activates when: Contact number negative and Aadhar ID can’t be negative

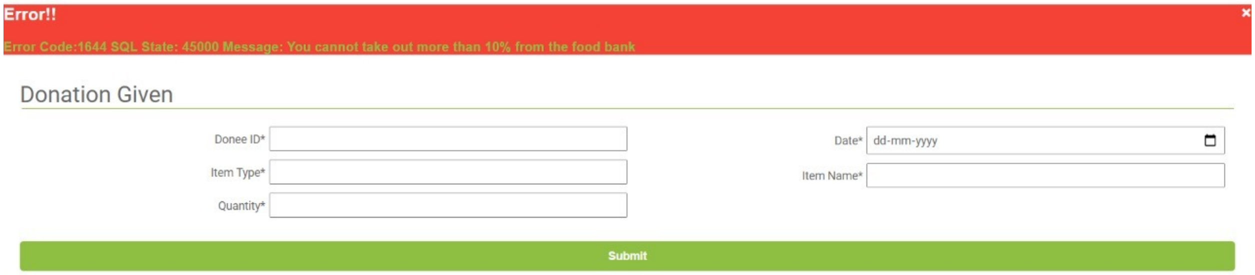


Error!!

Error Code:1644 SQL State: 45025 Message: Contact Number can't be less than 10 digits

VIII. qty_check before insert on donated

The trigger activates when: You cannot take out more than 10% from the food bank and Quantity cannot be negative

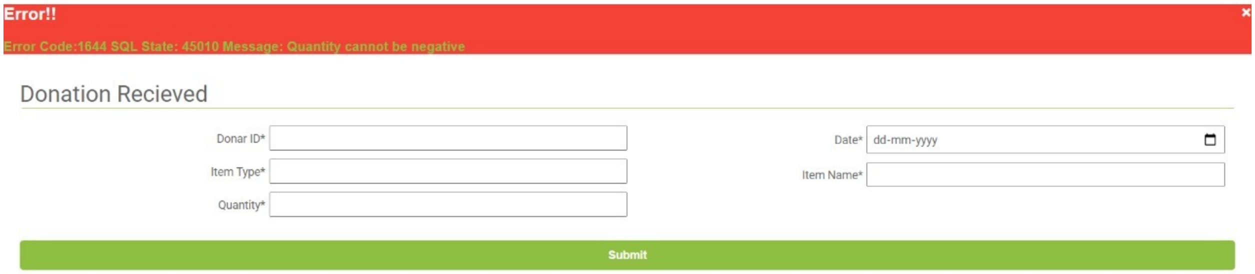


Error!!

Error Code:1644 SQL State: 45000 Message: You cannot take out more than 10% from the food bank

IX. qty_chk before insert on donation

The trigger activates when: Quantity cannot be negative



Error!!

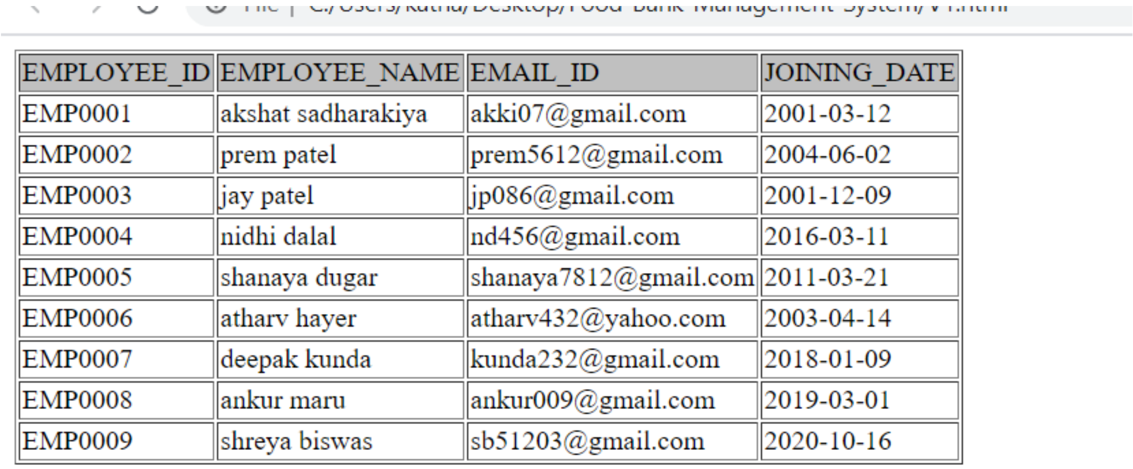
Error Code:1644 SQL State: 45010 Message: Quantity cannot be negative

X. **trigger date_donee_contraint before insert on donated**

The trigger activates when: Same Donee cannot take donation more than once

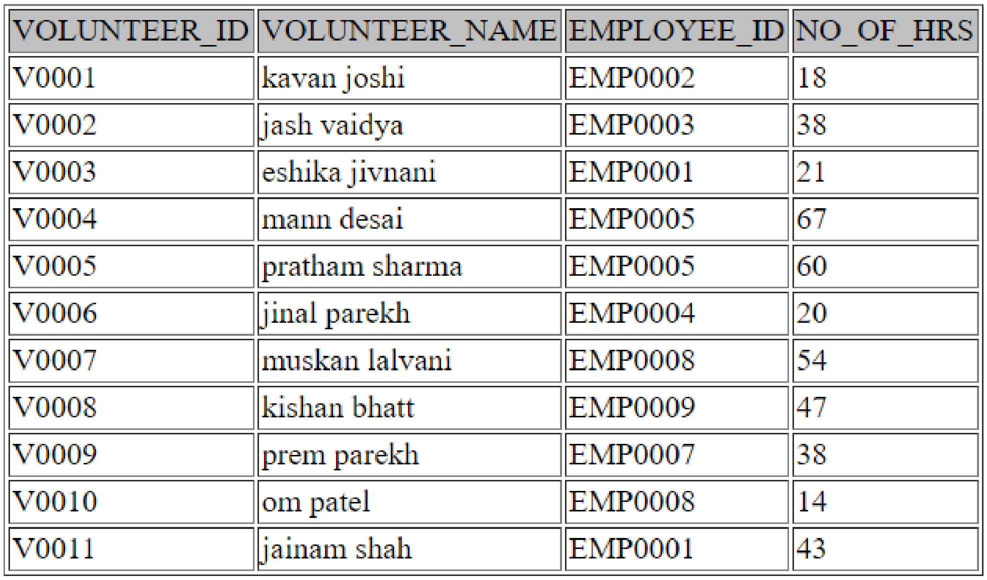
8. Views: are created to restrict a particular user to access confidential information among tables.

I. **create view emp as select EMPLOYEE_ID, EMPLOYEE_NAME, EMAIL_ID, JOINING_DATE from employee;**



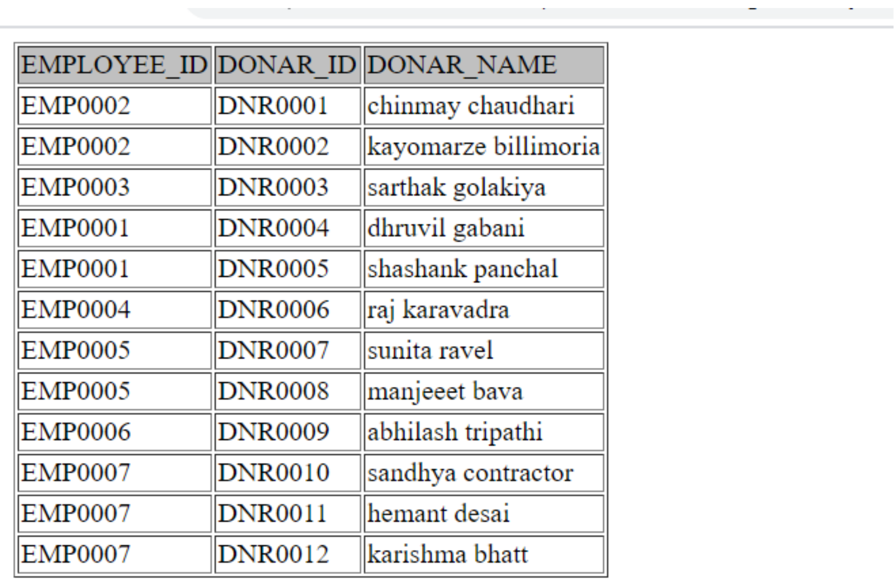
EMPLOYEE_ID	EMPLOYEE_NAME	EMAIL_ID	JOINING_DATE
EMP0001	akshat sadharakiya	akki07@gmail.com	2001-03-12
EMP0002	prem patel	prem5612@gmail.com	2004-06-02
EMP0003	jay patel	jp086@gmail.com	2001-12-09
EMP0004	nidhi dalal	nd456@gmail.com	2016-03-11
EMP0005	shanaya dugar	shanaya7812@gmail.com	2011-03-21
EMP0006	atharv hayer	atharv432@yahoo.com	2003-04-14
EMP0007	deepak kunda	kunda232@gmail.com	2018-01-09
EMP0008	ankur maru	ankur009@gmail.com	2019-03-01
EMP0009	shreya biswas	sb51203@gmail.com	2020-10-16

II. **create view vol as select VOLUNTEER_ID, VOLUNTEER_NAME, EMPLOYEE_ID, NO_OF_HRS from volunteer;**



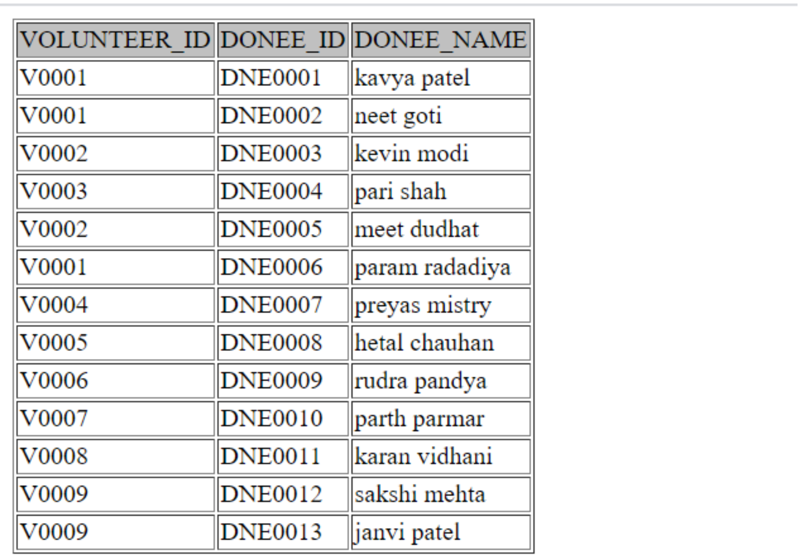
VOLUNTEER_ID	VOLUNTEER_NAME	EMPLOYEE_ID	NO_OF_HRS
V0001	kavan joshi	EMP0002	18
V0002	jash vaidya	EMP0003	38
V0003	eshika jivnani	EMP0001	21
V0004	mann desai	EMP0005	67
V0005	pratham sharma	EMP0005	60
V0006	jinal parekh	EMP0004	20
V0007	muskan lalvani	EMP0008	54
V0008	kishan bhatt	EMP0009	47
V0009	prem parekh	EMP0007	38
V0010	om patel	EMP0008	14
V0011	jainam shah	EMP0001	43

III. **create view dnr as select EMPLOYEE_ID, DONAR_ID, DONAR_NAME from donar;**



EMPLOYEE_ID	DONAR_ID	DONAR_NAME
EMP0002	DNR0001	chinmay chaudhari
EMP0002	DNR0002	kayomarze billimoria
EMP0003	DNR0003	sarthak golakiya
EMP0001	DNR0004	dhruvil gabani
EMP0001	DNR0005	shashank panchal
EMP0004	DNR0006	raj karavadra
EMP0005	DNR0007	sunita ravel
EMP0005	DNR0008	manjeeet bava
EMP0006	DNR0009	abhilash tripathi
EMP0007	DNR0010	sandhya contractor
EMP0007	DNR0011	hemant desai
EMP0007	DNR0012	karishma bhatt

IV. create view dne as select VOLUNTEER_ID, DONEE_ID, DONEE_NAME from donee;



VOLUNTEER_ID	DONEE_ID	DONEE_NAME
V0001	DNE0001	kavya patel
V0001	DNE0002	neet goti
V0002	DNE0003	kevin modi
V0003	DNE0004	pari shah
V0002	DNE0005	meet dudhat
V0001	DNE0006	param radadiya
V0004	DNE0007	preyas mistry
V0005	DNE0008	hetal chauhan
V0006	DNE0009	rudra pandya
V0007	DNE0010	parth parmar
V0008	DNE0011	karan vidhani
V0009	DNE0012	sakshi mehta
V0009	DNE0013	janvi patel