**PROJECT**

**ON**

**BLOOD BANK MANAGEMENT SYSTEM**



SUBMITTED TO: SUBMITTED BY:

Dr. Anika Chirag Setia

GROUP: 101603079

COE6 (G2) Chirag Madan

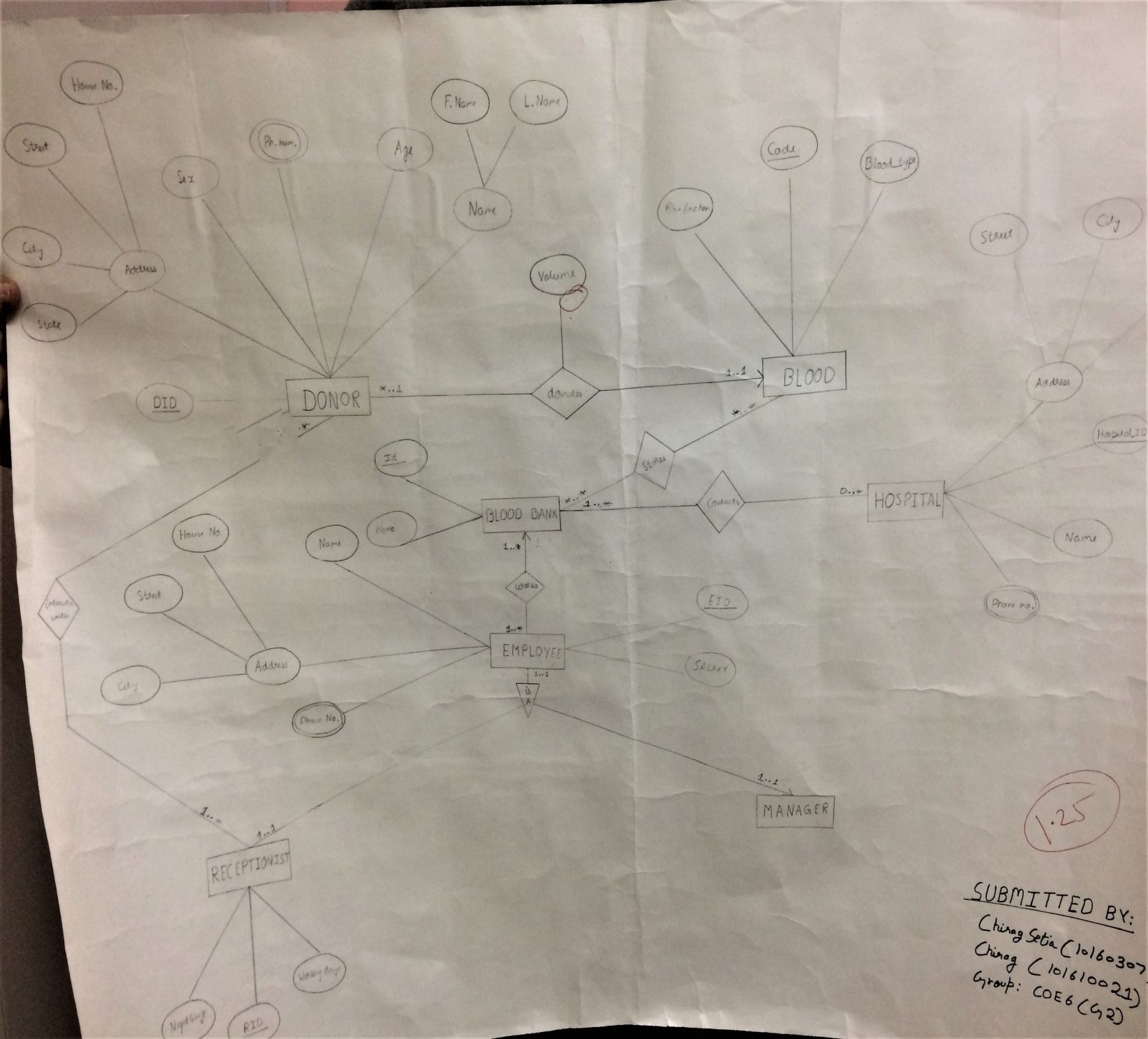
101610021

**ER DIAGRAM - CASE STUDY**

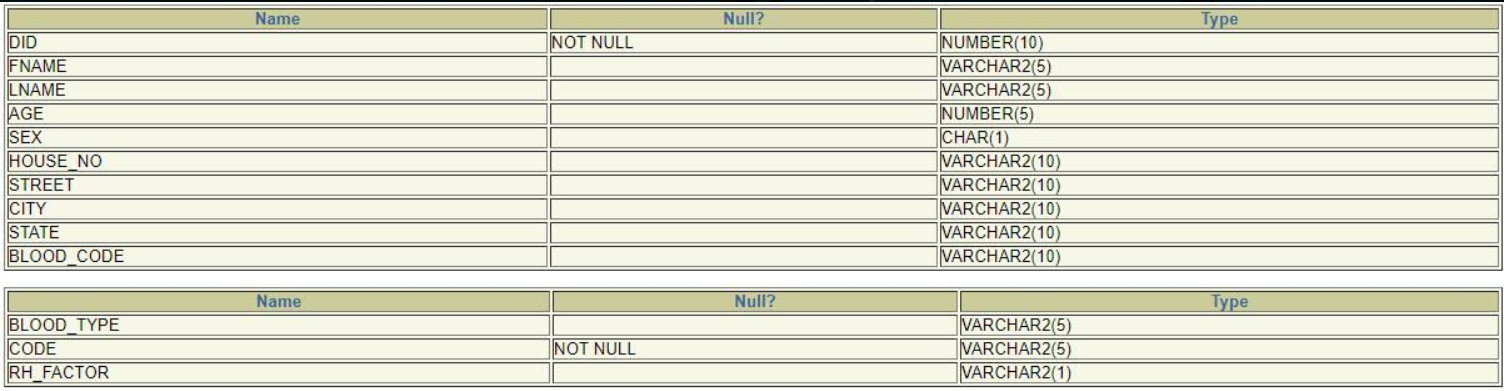
Requirement analysis

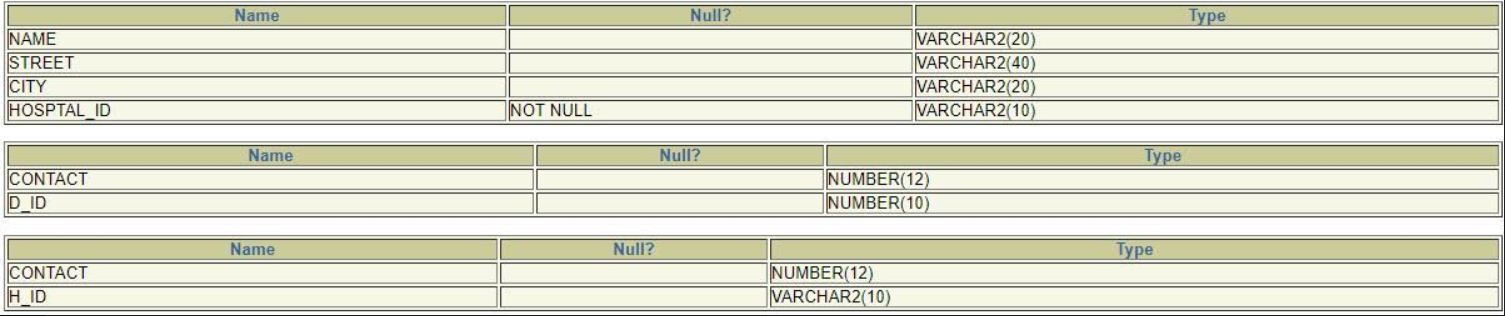
Consider a blood bank management system. Each blood bank contains many employees under it. The employee can be a manager or a receptionist. Each blood bank has only one manager. Each manager is associated with only one blood bank. Every blood bank can have any number of receptionist, but a receptionist can be associated with only one blood bank. A receptionist can have day or night shift or both. A receptionist can interact with multiple donors and vice versa. A donor can donate blood any number of time. A blood bank stores multiple types of blood and multiple type of blood are present in different blood banks. Multiple hospitals contact multiple blood banks. Every blood bank must have blood of every type.

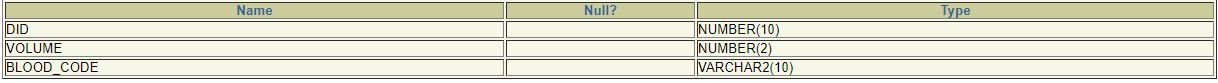
**Er Diagram**

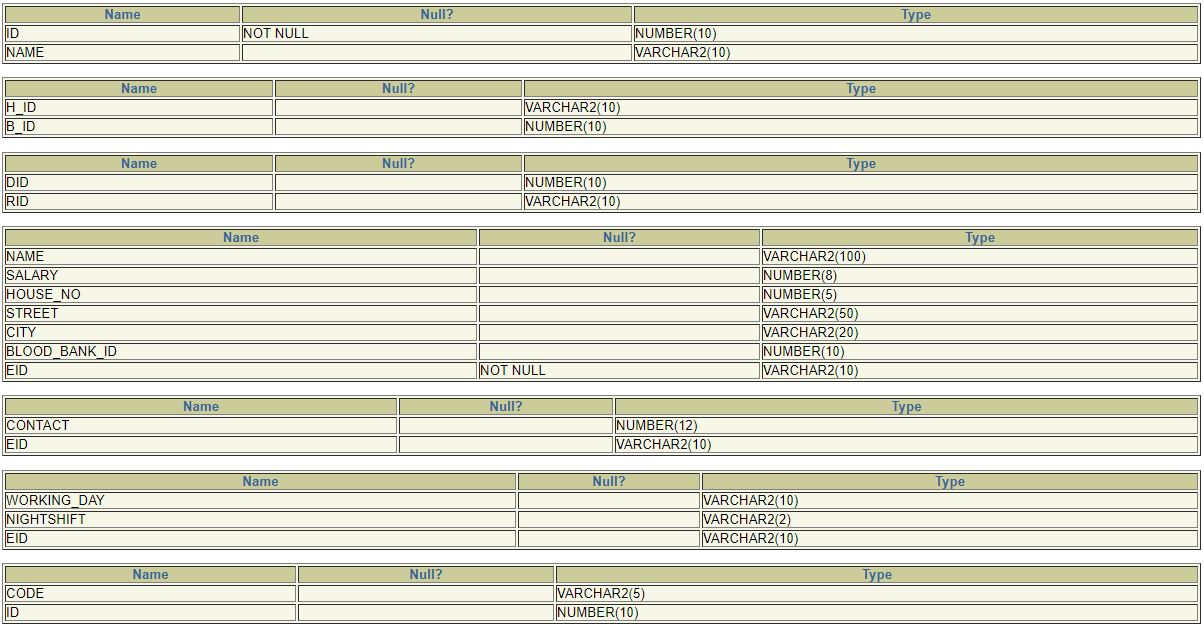
****

ER TO NORMALIZED TABLES



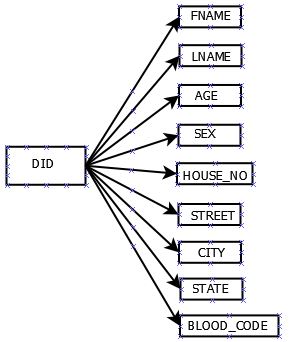
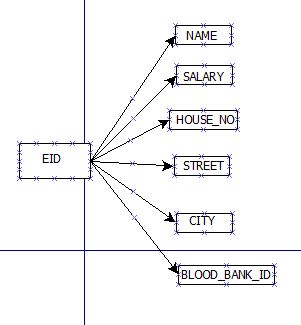




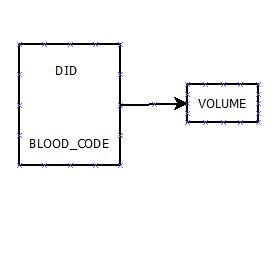
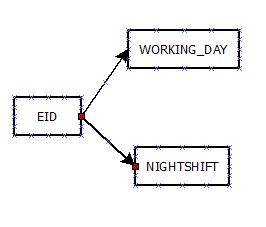


**FUNCTIONAL DEPENDENCE DIAGRAM OF NORMALIZED TABLES**

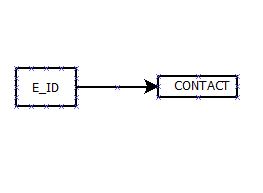
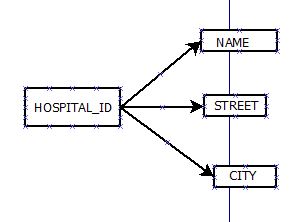
EMPLOYEE TABLE DONOR TABLE



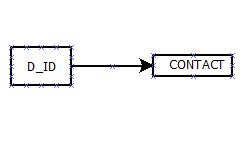
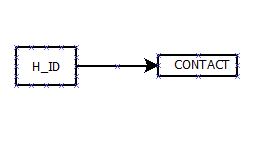
RECEPTIONIST TABLE VOLUME TABLE(DONOR\_BLOOD)



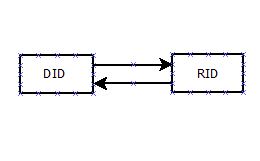
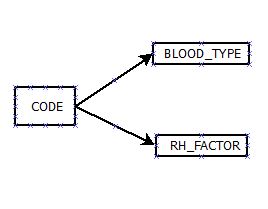
HOSPITAL TABLE EMPLOYEE CONTACT TABLE(E\_P)



DONOR CONTACT TABLE(D\_P) HOSPITAL CONTACT TABLE(H\_P)

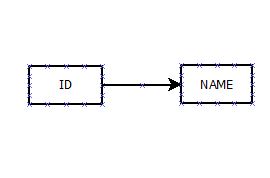
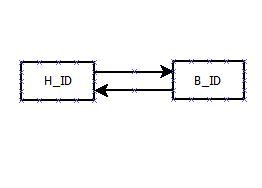
 

BLOOD TABLE DONAR & RECEPTIONIST TABLE(R\_D)



HOSPITAL & BLOOD BANK TABLE BLOOD BANK TABLE(BLOOD\_BANK)

(BANK\_HOSPITAL)



**PL-SQL CODES**

---------------------START--------------------------

-------------------- entry in blood\_bank--------------

set verify off;

declare

name varchar2(10) := '&name';

i number(10) := &id;

name\_is\_must exception;

begin

if name is null

then

raise name\_is\_must;

end if;

insert into blood\_bank values(i,name);

bbp(i);

exception

when name\_is\_must

then

raise\_application\_error (-20005,' blood name is not null');

end;

------------------------- entry in bank code-------------

create or replace procedure bbp(i in number)

as

a blood.code%type;

cursor cdd is select code from blood;

begin

open cdd;

loop

fetch cdd into a;

insert into b\_bb values(a,i);

exit when cdd%notfound;

end loop;

end;

/\* for input the value in b\_bb

\*/

-------------------------------hospital--------------------------------

declare

name varchar2(20) := '&name';

street varchar2(40) := '&street';

city varchar2(20) := '&city';

id varchar2(20) := '&id';

a number(15) := &number1;

b number(15) := &number2;

begin

hp(name,street,city,id,a,b);

end;

/\*

input in hospital

\*/

create or replace procedure hp(x in varchar2,y in varchar2,z in varchar2, i in varchar2, n in number, n1 in number)

as

begin

insert into hospital values(x,y,z,i);

insert into h\_p values(n,i);

insert into h\_p values(n1,i);

end;

---------------------------------employee-----------------------------

declare

name varchar2(20) := '&name';

hno number(5):= &house\_no ;

s number(5) := &salary ;

street varchar2(40) := '&street';

city varchar2(20) := '&city';

id varchar2(10) := '&id';

bb number(10):='&blood\_bank\_id';

a number(15) := &number1;

b number(15) := &number2;

e exception;

begin

if bb is null

then

raise e;

end if;

insert into employee values(name,s,hno,street,city,bb,id);

insert into e\_p values(a,id);

insert into e\_p values(b,id);

exception

when e

then

dbms\_output.put\_line('every employee must be associated with one blood bank.');

end;

/\* employee entry

\*/

---------------------------------reception---------------------------------

declare

name varchar2(20) := '&working\_days';

n varchar2(3) := '&Nightshift\_with\_caps\_YES\_or\_NO';

s varchar2(10) := '&employee\_id';

begin

insert into receptionist values(name,n,s);

end;

/\* RECETION INPUT\*/

----------------------------Relation b/w donor and reception----------------

declare

name number(10) := '&donar\_id';

n varchar2(10) := '&receptionin\_id';

begin

insert into r\_d values(name,n);

end;

/\*input\*/

----------------------------relation b/w hospital and bank---------------------

declare

name varchar2(10) := '&hospital\_id';

n number(10) := '&blood\_bank\_id';

begin

insert into bank\_hospital values(name,n);

end;

/\*input\*/

---------------------Starting sql block(donor comes)----------------------------

DECLARE

fname varchar2(10);

lname varchar2(10);

phone number(12);

age number;

sex varchar2(5);

house\_no varchar2(10);

stt varchar2(10);

city varchar2(10);

statei varchar2(10);

blood\_typei varchar2(10);

rh\_factori varchar2(10);

blood\_code varchar2(10);

vol number;

id number;

vid number;

too\_much\_blood EXCEPTION;

BEGIN

vid:=101;

id:='&Your\_donor\_id';

fname :='&first\_name';

lname:='&last\_name';

phone :='&contact';

age:='&age';

sex :='&gender';

house\_no :='&house\_number';

stt:='&street\_name';

city :='&city';

statei :='&state';

blood\_typei :='&blood\_grp';

rh\_factori :='&rh\_factor';

vol:='&volume';

if vol>5 then

raise too\_much\_blood;

end if;

if id is NULL then

id:=vid+1;

vid:=vid+1;

new\_reg(id,fname,lname,age,sex,house\_no,stt,city,statei,blood\_typei,rh\_factori,phone,vol);--procedure call for new donor--

dbms\_output.put\_line('Your id is '||id);

dbms\_output.put\_line('Current status of blood in blood bank');

display\_total\_blood;

else

for\_already\_reg(id,vol);--procedure call for already registered member--

dbms\_output.put\_line('Current status of blood in blood bank');

display\_total\_blood;--procedure call for current status of blood--

end if;

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

EXCEPTION

when too\_much\_blood then

raise\_application\_error(-20001,'You cannot donate too much blood at once');

END;

-----------------------procedure for already registered member----------------

create or replace procedure for\_already\_reg(id in number,vol in number)

is

rowcoun number;

oldvol number;

newvol number;

BEGIN

select count(\*) into rowcoun from donor where did=id;

if rowcoun=0 then

dbms\_output.put\_line('your donor id is not registered with us..kindly go for registration process');

raise\_application\_error(-20001,'User not validated');

else

dbms\_output.put\_line('Donor id validated');

select volume into oldvol from donor\_blood where did=id;

newvol:=oldvol+vol;

update donor\_blood set volume=newvol where did=id;

dbms\_output.put\_line('Thank you for donating blood');

end if;

end;

-------------------procedure for new donor---------------------------------------

create or replace procedure new\_reg(

id number,

fname varchar2,

lname varchar2,

age number,

sex varchar2,

house\_no varchar2,

stt varchar2,

city varchar2,

statei varchar2,

blood\_typei varchar2,

rh\_factori varchar2,

phone number,

vol number

)

is

blood\_code varchar2(10);

rowcount number;

oldvol number;

newvol number;

BEGIN

Select code into blood\_code from blood where blood\_type=blood\_typei and rh\_factor=rh\_factori;

insert into donor values(id,fname,lname,age,sex,house\_no,stt,city,statei,blood\_code);

insert into D\_P values(phone,id);

select count(\*) into rowcount from donor\_blood where did=id;

if rowcount=0 then

insert into donor\_blood values(id,vol,blood\_code);

else

select volume into oldvol from donor\_blood where did=id;

newvol:=oldvol+vol;

update donor\_blood set volume=newvol where did=id;

end if;

dbms\_output.put\_line('THANK YOU FOR DONATING BLOOD');

end new\_reg;

----------------Trigger for age check of donor (age of donor should be greater than 18)---------------------

CREATE OR REPLACE TRIGGER check\_age

BEFORE INSERT ON donor

FOR EACH ROW

BEGIN

if :new.age<18 THEN

dbms\_output.put\_line('age is less than 18');

raise\_application\_error(-20001,'You are not eligible to donate blood');

end if;

end;

------------------procedure to calculate total blood for each blood group present in blood bank currently--------------

create or replace procedure display\_total\_blood

as

total number;

bcode varchar2(10);

btype varchar2(5);

rh varchar2(5);

ctotal number:=0;

cursor c1 is select sum(volume),blood\_code from donor\_blood group by blood\_code;

BEGIN

OPEN c1;

dbms\_output.put\_line('BLOOD\_GRP\_\_\_\_\_\_\_\_\_\_\_VOLUME AT PRESENT');

LOOP

fetch c1 into total, bcode;

select blood\_type,rh\_factor into btype,rh from blood where code=bcode;

exit when c1%notfound;

dbms\_output.put\_line('\_\_\_\_\_\_'||btype||rh||'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'||total||'\_\_\_\_\_\_\_');

ctotal:=ctotal+total;

end loop;

close c1;

dbms\_output.put\_line('Total volume of blood irrespective of blood group = '||ctotal);

end;

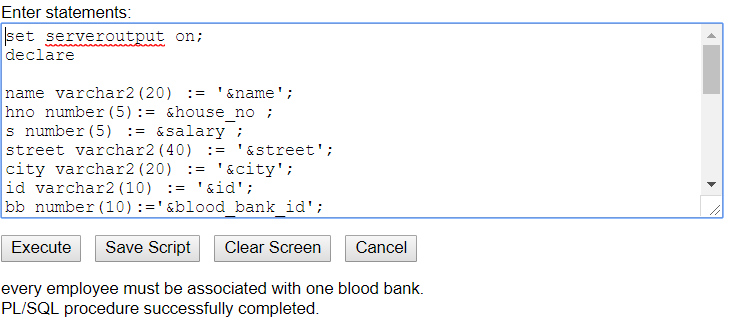
-------------------END---------------------

**EXCEPTIONS AND PROCEDURE OUTPUT**

* EXCEPTION HANDLING IN EMPLOYEE TABLE

Every employee must be associated with Blood bank

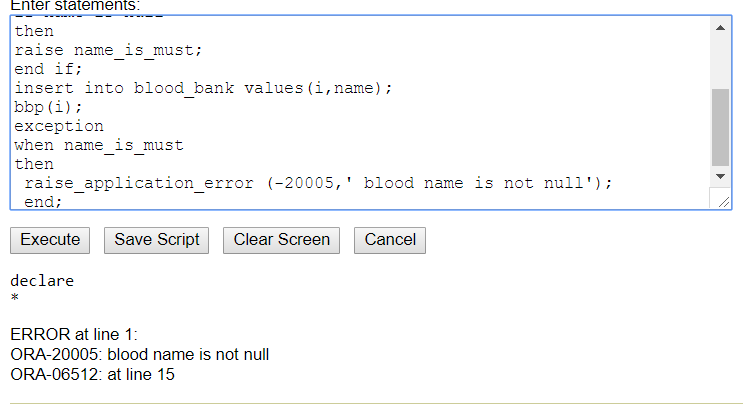
ie blood bank id can’t be null



* EXCEPTION HANDLING IN BLOOD BANK TABLE

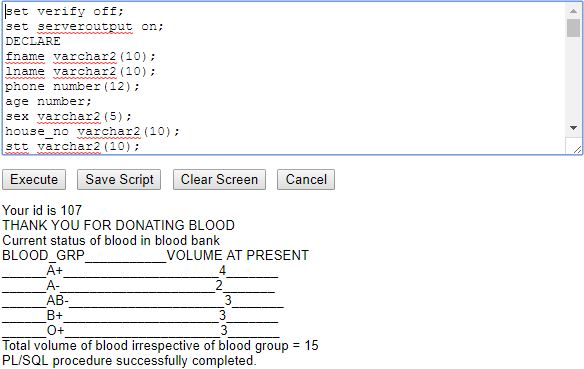
Every blood bank must have some name.

ie name can’t be null



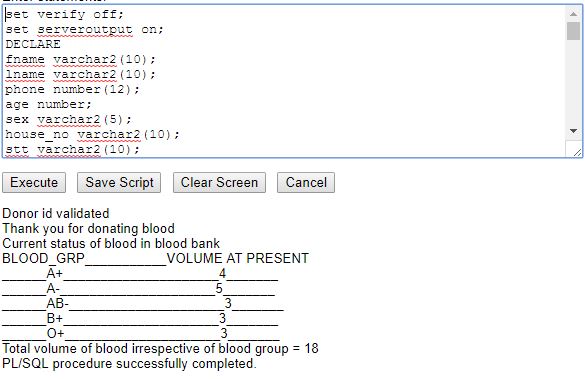
* PROCEDURE FOR NEW DONOR

In this a new donor id is allotted to Donor and simultaneously new row is inserted in donar\_blood table using apt values and total blood is displayed thereafter.

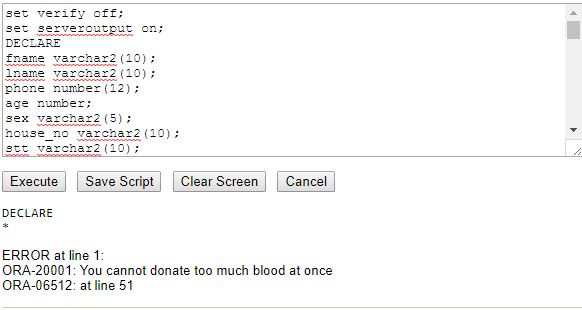


* **Procedure for already registered donor**

In this only donor id and volume of blood is taken as input as is updated in Donor\_blood table. Total blood is displayed after donation.



* **EXCEPTION HANDLING IF INPUT VOLUME IS >5**



* **TRIGGER IF AGE IS LESS THAN 18**

This trigger raises an application error in Donor table when input age is less than 18.

