Blood Bank Management System (BBMS) Database

Final Project Report

CSE 3110: Database Systems Laboratory

June, 2018

Sakib Reza (1507007)

Project Overview

Blood Bank Management System (BBMS) is database system to link between the donors and blood banks and act as an interface for the patient to find his/her desired blood in a fast and efficient way. It will make the blood transfusion service and its management more reliable and efficient than the conventional system.

Transfusion of blood and blood components is an established standard way of treating patients who are deficient in one or more blood constituents and is therefore an essential part of health care. A blood transfusion service is a complex organization requiring careful design and management. Essential functions of a blood transfusion service are donor recruitment, blood collection, testing of donor blood, component preparation and supply of these components to the patients.

The organization of a blood transfusion service should receive utmost attention and care for smooth functioning of various components of the service. The goal of blood transfusion service is to provide effective blood and blood components which are as safe as possible and adequate to meet the patients' needs.

The basic functions of a blood transfusion center may be listed as follows - Recruitment and retention of voluntary and replacement blood donors, collection, processing, storage and transportation of blood and its components, laboratory procedures , participation in the clinical use of blood and blood components, teaching and training of personnel, research and development need for Blood Bank.

As a blood transfusion service deals with different functions related to donors and patients, it is imperative to keep in mind the safety of both donors and recipients. The blood transfusion service has to be planned and organized in such a way that it fulfils its ideal aims and objectives i.e. Recruitment of blood donors - voluntary & replacement, care of donor, donated unit and the recipient, maintain adequate blood stock, provide clinically effective blood components, optimal use of available blood.

In Bangladesh, Blood transfusion services are mostly hospital-based. In a hospital based blood transfusion service, each hospital runs its own blood collection programme with or without central regulation. This system utilizes existing institutions and does not require creation of separate blood transfusion centers. As the organization of voluntary blood donation in hospital-based system is usually unsatisfactory, replacement donors form the main source of blood supply in the hospital. Replacement donors are usually friends and family members of the patient and are under pressure to donate blood. A voluntary donor system is far more satisfactory as there is no compulsion and the donor is motivated to donate blood.

Objectives of the project is to Improve operational & Streamlining operations, maintain all the project, employee, project database, maintain global standards of the project development, reduce Manpower and manual paper works for maintaining the records offline, main the accuracy, integrity and consistency of the data, providing such a mechanism to make the man power fast to maintain all the information about the project, improved management and control of the inventory, sales, stock.

Database Structure

The database consists of four tables. The main tables are the patient, donor and bloodbank tables. The remaining table ‘donate’ is the relational table which links ‘donor’ and bloodbank tables with its foreign keys.

DATABASE TABLE	DESCRIPTION
Patient	Contains all of the patient’s information.
Donor	Contains all of the donor’s information.
Bloodbank	Contains all of the blood bank’s information.
Donate	Serves as the main linking table for the ‘donor’ and ‘bloodbank’ table by containing the foreign keys of them.

Functionality

The database and web application allow technicians to complete the following functions:

- Insert a new patient’s information.
- Insert a new donor’s information.
- Insert a new blood’s bank information.
- Update/modify any of these records.
- Find the desired blood from the database in different constraints.
- Find the quantity of available desired blood in different constraints.
- Find the desired blood bank in different constraints.
- Find the desired donor in different constraints.
- Check any updated information about any patient, donor or blood bank

Customers/Audience

The main customers/audience for the system are the patients, donors and blood banks. A patient who need blood for him/herself can use to find his desired blood. A donor who is willing to donate blood can donate to any of the blood banks. A blood bank can use to collect blood from the donors and deliver it to the needy patients

Database Design Process

To build the database system I have used the Oracle DBMS, Toad, basic SQL (structured query language) and PL/SQL (procedural language/ structured query language). Use of the PL/SQL alongside with basic SQL made the system more powerful and efficient. PL SQL consists of blocks of code, which can be nested within each other. Each block forms a unit of a task or a logical module. PL/SQL Blocks can be stored in the database and reused. PL SQL consists of procedural language constructs such as conditional statements (if else statements) and loops like (FOR loops). PL SQL engine processes multiple SQL statements simultaneously as a single block, thereby reducing network traffic. PL/SQL handles errors or exceptions effectively during the execution of a PL/SQL program. Once an exception is caught, specific actions can be taken depending upon the type of the exception or it can be displayed to the user with a message.

First, I have designed my tables and relationship between them. We have make a relation between ‘bloodbank’ and ‘donor’ table using the ‘donate’ table. Than I have implemented some queries and views to show the desired results.

We learned several important lessons through the design process. These include:

1) Designing your tables is the most important step and must be done early in the project.

2) Initial stage of building a database system is very important as modify any part of the design after completing the system is pretty much time consuming and difficult.

3) Being able to design a database well for a targeted audience requires the practical understanding of the environment.

Future of the Database

The database is currently just a prototype. I have not yet built any front end GUI for this database system. To make this database system user-friendly for general people I need to build a GUI so that they can use this system easily.

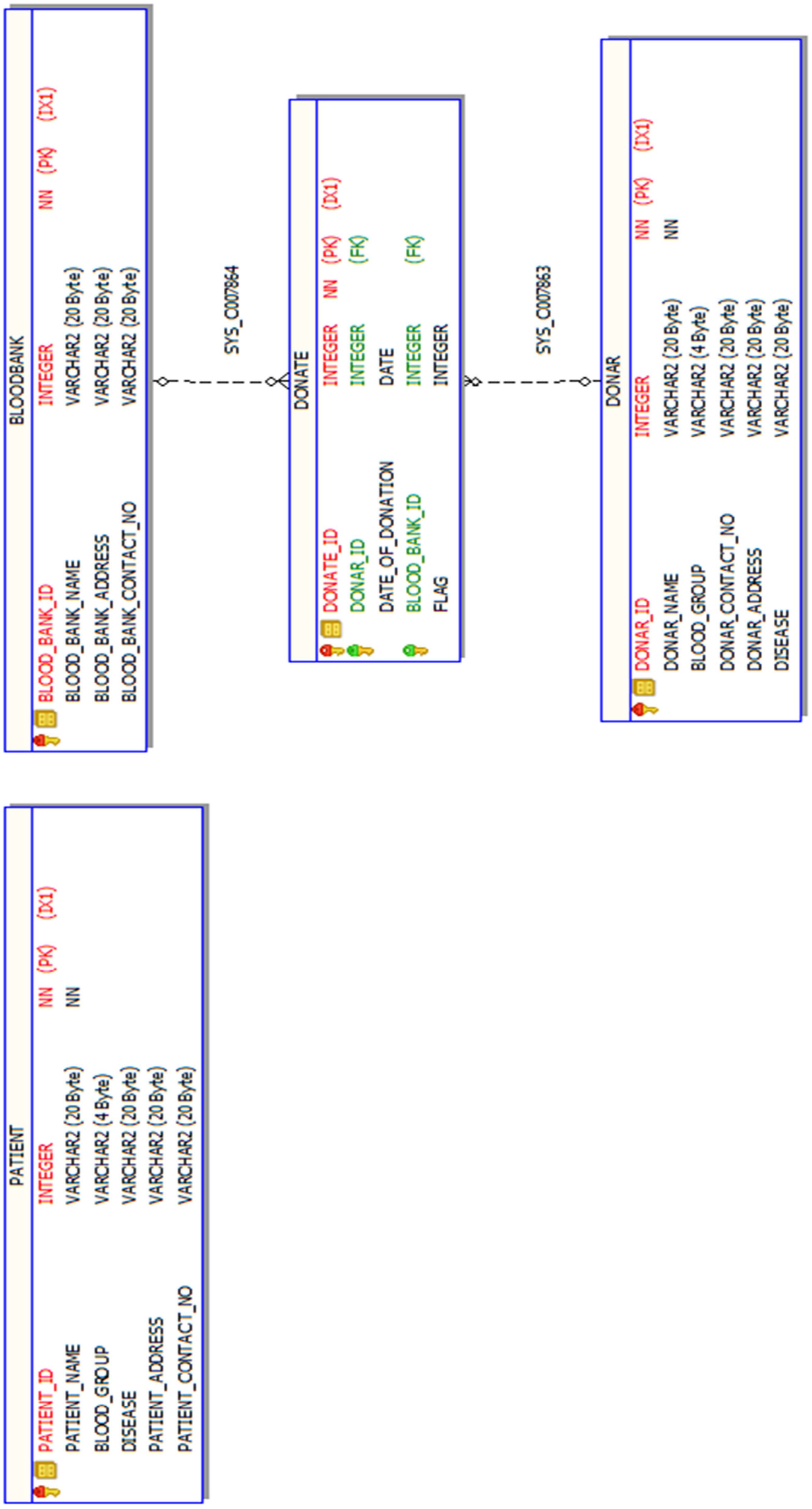
Web application is nowadays the most popular form to build a database system. So, in future I want to make a web application for this database system and host it on a online server so that people from anywhere around the world can have access to this system using internet.

Before doing so, I need to upgrade this system, make it more robust. Than we need to take survey to be get an idea of which type of services needed in blood transfusion system. After that I will have to study about my targeted user and make a user-friendly interface for them to use my system. Finally, I will host my application built for the system online and make it open for the users.

Summary

The project was a learning experience for me and allowed me to improve upon my SQL, PL/SQL, and Oracle DBMS skills. I developed a usable database for a blood donation organization to keep track of their donors, find available blood bags from blood bank and deliver it to the patients in an efficient way. I look forward to seeing the database being put into actual use in future.

ER Diagram


BLOODBANK
BLOOD_BANK_ID INTEGER NN (PK) (IX1)
BLOOD_BANK_NAME VARCHAR2 (20 Byte)
BLOOD_BANK_ADDRESS VARCHAR2 (20 Byte)
BLOOD_BANK_CONTACT_NO VARCHAR2 (20 Byte)
SYS_C007864
DONATE
DONATE_ID INTEGER NN (PK) (IX1)
DONAR_ID INTEGER (FK)
DATE_OF_DONATION DATE
BLOOD_BANK_ID INTEGER (FK)
FLAG INTEGER
SYS_C007863
DONAR
DONAR_ID INTEGER NN (PK) (IX1)
DONAR_NAME VARCHAR2 (20 Byte) NN
BLOOD_GROUP VARCHAR2 (4 Byte)
DONAR_CONTACT_NO VARCHAR2 (20 Byte)
DONAR_ADDRESS VARCHAR2 (20 Byte)
DISEASE VARCHAR2 (20 Byte)
PATIENT
PATIENT_ID INTEGER NN (PK) (IX1)
PATIENT_NAME VARCHAR2 (20 Byte) NN
BLOOD_GROUP VARCHAR2 (4 Byte)
DISEASE VARCHAR2 (20 Byte)
PATIENT_ADDRESS VARCHAR2 (20 Byte)
PATIENT_CONTACT_NO VARCHAR2 (20 Byte)

Source Code

Procedure, Cursor, SQL Sub-query, Loop:

Problem:

1. Donors available for a particular patient
2. Blood Bank Available for a particular patient
3. Blood Bank Available for a particular patient’s place
4. Blood Bag’s quantity in different Blood Bank of a particular blood group
5. Quantity of Blood Group in a particular Blood Bank
6. Blood Banks with a desired blood group with quantity
7. Quantity of a desired blood group in a desired blood bank
8. Donors available donated before a particular date of a desired blood group
9. Donors with blood banks available donated before a particular date of a desired blood group

Solution:

```
1.
PROCEDURE donars_for_particular_patient (pid IN Patient.patient_id%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of donars for patient id no. '||pid);
    DBMS_OUTPUT.PUT_LINE('----------------------------------------');

    FOR cursor1 IN (select donar_id,donar_name,blood_group,donar_contact_no from
Donar where donar_id in (select donar_id from Donate where donar_id in (select
donar_id from donar where blood_group in (select blood_group from Patient where
patient_id=pid)) ))
          LOOP
            DBMS_OUTPUT.PUT_LINE('donar name: ' || cursor1.donar_name ||
                               ', blood group: ' || cursor1.blood_group||
                               ', contact no: ' || cursor1.donar_contact_no);
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;


2.
PROCEDURE banks_available_for_patient (pid IN Patient.patient_id%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of blood banks for patient id no. '||pid);
    DBMS_OUTPUT.PUT_LINE('----------------------------------------------');
    FOR cursor1 IN (select blood_bank_id,blood_bank_name,blood_bank_contact_no from
BloodBank where blood_bank_id in (select blood_bank_id from Donate where donar_id in
(select donar_id from donar where blood_group in (select blood_group from Patient
where patient_id=1001))))
        LOOP
            DBMS_OUTPUT.PUT_LINE('id: ' || cursor1.blood_bank_id ||
                               ', blood bank name: ' || cursor1.blood_bank_name||
                               ', contact no: ' || cursor1.blood_bank_contact_no);
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;


3.
PROCEDURE bank_in_patients_place (pid IN Patient.patient_id%type) IS
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('List of blood bank for patient id no.'||pid|| ' in his/her
place');
    DBMS_OUTPUT.PUT_LINE('----------------------------------------');

    FOR cursor1 IN (select
blood_bank_id,blood_bank_name,blood_bank_contact_no,blood_bank_address from BloodBank
where blood_bank_id in (select blood_bank_id from Donate where donar_id in (select
donar_id from donar where blood_group = (select blood_group from Patient where
patient_id=1001)) ) and blood_bank_address in (select patient_address from Patient
where patient_id=1001))
          LOOP
            DBMS_OUTPUT.PUT_LINE('id: ' || cursor1.blood_bank_id ||
                               ', blood bank name: ' || cursor1.blood_bank_name||
                               ', contact no: ' || cursor1.blood_bank_contact_no);
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;
```

4.

```
PROCEDURE bank_desired_blood_quantity (blood IN Patient.blood_group%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of blood bank for blood group '||blood|| ' with its
quantity');
    DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------
-------');

    FOR cursor1 IN (select blood_bank_name,count(*) as quantity from (select
T.BLOOD_BANK_NAME,T.BLOOD_BANK_CONTACT_NO from (select * from BloodBank natural join
Donate) t natural join (select * from Donar where blood_group=blood)) group by
blood_bank_name)
        LOOP
            DBMS_OUTPUT.PUT_LINE('blood bank name: ' || cursor1.blood_bank_name ||
                               ', quantity: ' || cursor1.quantity
                               );
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;
```

5.

```
PROCEDURE blood_group_quantity_in_bank (bank_no IN BloodBank.blood_bank_id%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of blood group with quantity in blood bank no.
'||bank_no);
    DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------
-------');

    FOR cursor1 IN (select blood_group,count(*) as quantity from (select * from
Donate natural join Donar) where blood_bank_id = bank_no group by blood_group)
        LOOP
            DBMS_OUTPUT.PUT_LINE('blood group: ' || cursor1.blood_group ||
                               ', quantity: ' || cursor1.quantity
                               );
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;
```

6.

```
PROCEDURE bank_with_desired_blood (blood IN donar.blood_group%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of blood banks having '||blood||' blood');
    DBMS_OUTPUT.PUT_LINE('---------------------------------------------');

    FOR cursor1 IN (select blood_bank_id,blood_bank_name,blood_bank_contact_no from
BloodBank where blood_bank_id in (select blood_bank_id from Donate where donar_id in
(select donar_id from donar where blood_group = blood)))
        LOOP
            DBMS_OUTPUT.PUT_LINE('id: ' || cursor1.blood_bank_id ||
                               ', blood bank name: ' || cursor1.blood_bank_name||
                               ', contact no: ' || cursor1.blood_bank_contact_no);
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;
```

```
7.
PROCEDURE quantity_of_group_in_bank (bank IN donate.blood_bank_id%type, blood IN
donar.blood_group%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('quantity of blood in bank no. '||bank||' of group
'||blood);
    DBMS_OUTPUT.PUT_LINE('---------------------------------------------------------
');

    FOR cursor1 IN (select count(donate_id) as quantity from Donate where
blood_bank_id = bank and donar_id in (select donar_id from Donar where blood_group =
blood))
           LOOP
            DBMS_OUTPUT.PUT_LINE('quantity of blood bag : ' || cursor1.quantity);
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;


8.
PROCEDURE donar_of_group_before_date (date_ IN donate.date_of_donation%type, blood IN
donar.blood_group%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of donar of '||blood||' blood donated before
'||date_);
    DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------
----');

    FOR cursor1 IN (select donar_name,blood_group,date_of_donation,donar_contact_no
from (select * from donate natural join donar) where date_of_donation < date_ and
blood_group=blood)
           LOOP
            DBMS_OUTPUT.PUT_LINE('donar name: ' || cursor1.donar_name ||
                                 ', blood group: '|| cursor1.blood_group ||
                                 ', date of donation: '|| cursor1.date_of_donation ||
                                 ', contact no: '|| cursor1.donar_contact_no
                                );
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');
END;


9.
PROCEDURE available_blood_of_group_date (date_ IN donate.date_of_donation%type,blood
IN donar.blood_group%type) IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('List of available '||blood||' blood before '||date_);
    DBMS_OUTPUT.PUT_LINE('--------------------------------------');

    FOR cursor1 IN (select
donate_id,donar_name,blood_group,blood_bank_name,blood_bank_contact_no from (select *
from donate natural join BloodBank) t natural join donar where t.FLAG=1 and
donar.blood_group=blood and t.date_of_donation < date_)           LOOP
            DBMS_OUTPUT.PUT_LINE('donate id: ' || cursor1.donate_id ||
                                 ', donar name: '|| cursor1.donar_name ||
                                 ', blood: '|| blood ||
                                 ', blood bank: '|| cursor1.blood_bank_name||
                                 ', contact no: '||cursor1.blood_bank_contact_no
                                );
          END LOOP;
    DBMS_OUTPUT.PUT_LINE('_________________________________________');

END;
```

Trigger:

Problem:

1. In inserting any donor information if he/she has any fatal disease, it will not be inserted
2. If there is any update of donor's location it will be displayed
3. If any blood bank is given to patient it will be updated & displayed

Solution:

```
1.
create or replace trigger desease_trg
before insert on Donar
for each row
when(NEW.donar_id>0)
begin
    if :NEW.disease = 'HepB' then
        dbms_output.put_line('donar with fatal disease!');
        delete from donar where donar_id = :NEW.donar_id;
    end if;
end;

2.
create or replace trigger loc_change
before update on Donar
for each row
when(NEW.donar_address!=OLD.donar_address)
declare

begin
    dbms_output.put_line('Old location : '|| :OLD.donar_address);
    dbms_output.put_line('New location : '|| :NEW.donar_address);
end;

3.
create or replace trigger blood_received
before update on Donate
for each row
when(NEW.flag!=OLD.flag)
declare

begin
    dbms_output.put_line('Blood bag given to the patient');

end;
```

Function:

Problem:

1. Display the total number of donors

Solution:

```
CREATE OR REPLACE FUNCTION totalDonar
RETURN number IS
   total integer := 0;
BEGIN
   SELECT count(*) into total
   FROM Donar;

   RETURN total;

END;
```

File, Array, Cursor, Loop:

Problem:

1. Read donors information from a file
2. Write donors information in different files group by blood group
3. Write donors information in different files group by location
4. Write blood bank information in different files group by location

Solution:

```
1.
begin
f:= utl_file.fopen('BBMS','new_entry.csv','r');
if utl_file.is_open(f) then
utl_file.get_line(f,line,1000);
loop begin
utl_file.get_line(f,line,1000);
if line is null then exit;
end if;
id:= regexp_substr(line,'[^,]+',1,1);
name:= regexp_substr(line,'[^,]+',1,2);
bgroup:= regexp_substr(line,'[^,]+',1,3);
contact:= regexp_substr(line,'[^,]+',1,4);
place:= regexp_substr(line,'[^,]+',1,5);
insert into donar (donar_id,donar_name,blood_group,donar_contact_no,donar_address)
values(id,name,bgroup,contact,place);
commit;
exception when no_data_found then exit;
end;
end loop;
end if;
--utl_file_fclose(f);

end;



2.
begin
   num := 1;
   cnt := 1;
   dbl_list.extend(11);

   FOR cursor1 IN (select distinct(blood_group) as blood from donar)
          LOOP
            dbl_list(num) := cursor1.blood;
            num:=num+1;
          END LOOP;

   while cnt<=num
   loop
        f:= utl_file.fopen('BBMS',concat(to_char(dbl_list(cnt)),'_donar.csv'),'w');

utl_file.put(f,'id'||','||'name'||','||'blood'||','||'contact'||','||'place');
        utl_file.new_line(f);
        for c_record in (select
donar_id,donar_name,blood_group,donar_contact_no,donar_address from donar where
blood_group = dbl_list(cnt))
        loop

utl_file.put(f,c_record.donar_id||','||c_record.donar_name||','||c_record.blood_group
||','||c_record.donar_contact_no||','||c_record.donar_address);
            utl_file.new_line(f);
        end loop;
        utl_file.fclose(f);
        --dbms_output.put_line(dbl_list(cnt));
        cnt:=cnt+1;
   end loop;

end;
```

3.

Same as (2)

4.

```
begin
   num := 1;
   cnt := 1;
   dbl_list.extend(11);

   FOR cursor1 IN (select distinct(blood_bank_address) as address from bloodbank)
          LOOP
            dbl_list(num) := cursor1.address;
            num:=num+1;
          END LOOP;

   while cnt<=num
   loop
        f:=
utl_file.fopen('BBMS',concat(to_char(dbl_list(cnt)),'_blood_bank.csv'),'w');
        utl_file.put(f,'id'||','||'name'||','||'place'||','||'contact');
        utl_file.new_line(f);
        for c_record in (select * from bloodbank where blood_bank_address =
dbl_list(cnt))
        loop

utl_file.put(f,c_record.blood_bank_id||','||c_record.blood_bank_name||','||c_record.b
lood_bank_address||','||c_record.blood_bank_contact_no);
            utl_file.new_line(f);
        end loop;
        utl_file.fclose(f);
        --dbms_output.put_line(dbl_list(cnt));
        cnt:=cnt+1;
   end loop;

end;
```