Blood Bank System

Members:
Md. Sabbir Hassan (14.02.04.058)
Nowshin Nawar Arony (14.02.04.067)
Sanjana Farial (14.02.04.100)

Introduction

The main aim of developing this system is to provide information of Donors to the people who are in need of blood. Almost everyday people face situations where they require blood of different groups. Using this system an user can search for a blood group and can also get the contact information of the donor who has the same blood group needed. The prime benefit of this system is that it can provide information of available Donors. When blood is needed in an operation, people often don't have much time to search for blood. So using a system like this can ease the searching hassles.

Overview of the system

We have developed our system based on Oracle PL/SQL procedure language. All the codes run in *sqlplus command prompt. As our system is based on distributed database concept here we have used 1 Server site and 1 host site.
We have 4 tables in total for storing detailed data of donors who have donated blood, recipients who have received blood, information of the blood and blood donation event in details. DONOR table holds all the required information of a donor who has donated blood to a recipient and in RECIPIENT table the information of the recipients' are stored. BLOOD_INVENTORY table saves the value of the bag numbers of the blood donated by a donor, hemoglobin and platelets number of that corresponding blood bag. Lastly, in the DONATION_DETAILS table, details of any blood donation event like the hospital at the event occurred, the amount of blood that was received and the date when the blood was given.
The functionalities of this project are described below:

- Insert information of donor into DONOR table.
- Delete information of donor from DONOR table.
- Update information of donor into DONOR table.
- Search donor from DONOR table by blood group.
- Search donor from DONOR table by donor id.
- Search donor from DONOR table by area.

- Search donor from DONOR table by eligibility of donor.
- Search recipient from RECIPIENT table by recipient id.
- Count total number of bags of a specific blood group.
- Calculate eligibility of donor.

These functionalities can be used in the host PC. In case of the site PC, only select, insert, update and delete can be used to manipulate the tables.

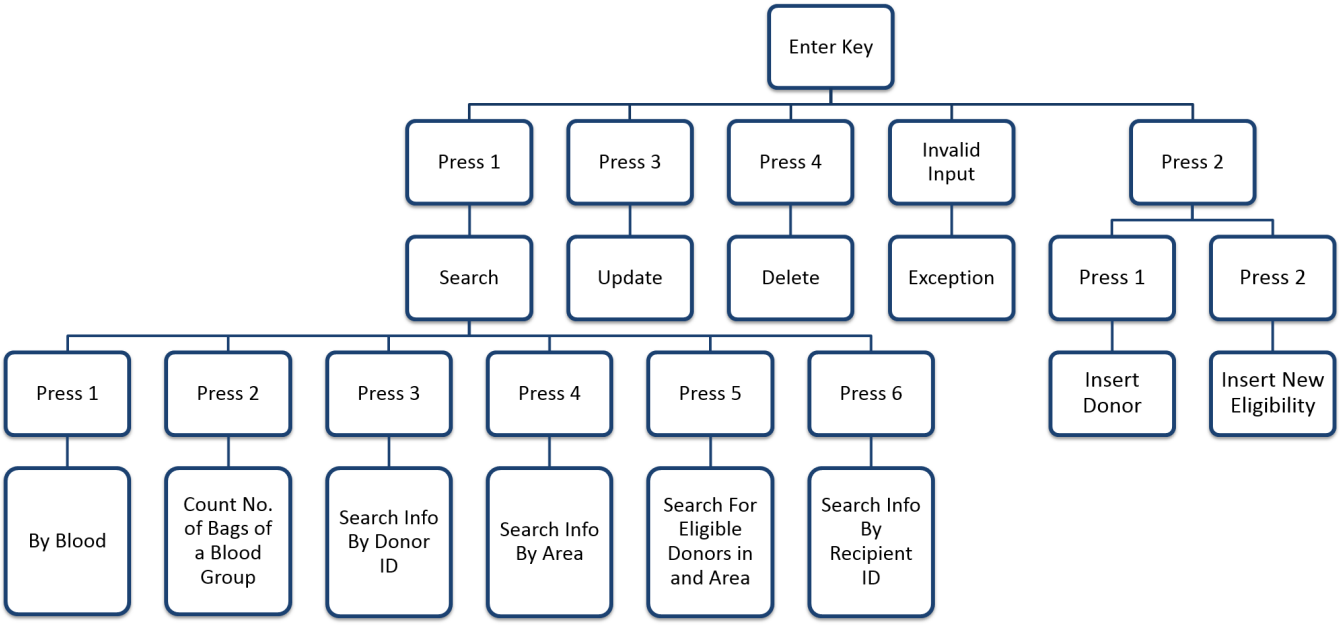

Figure 1: Flow Diagram of the System

Sites and tables

In this project we are using 1 site and 1 host. The site is installed in VMware. The tables of the project are run in the site. Then accessed via a site_link connection. And the main PC is the host from where the tables of the site are accessed.
There are 4 relational schema in this system. The Relational schema are shown below:
DONOR (DID, Dname, Dage, Dgender, Dbloodgroup, Darea, Daddress, Dphonenum, Deligibility)
RECIPIENT (RID, Rname, Rage, Rgender, Rbloodgroup, Raddress, Rphonenum, DID)
BLOOD_INVENTORY (DID, bagnumber, heamoglobin, platelets)
DONATION_DETAILS (DID, donationnumber, hospital, amount, givenat)

Database links

We are using one site and one host. The site link is created using this following code:

```
drop database link site_link;

create database link site_link
connect to system identified by "123"
using '(DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)
 (HOST = 192.168.128.101)
 (PORT = 1521))
       )
       (CONNECT_DATA =
         (SID = XE)
       )
     )'
;
```

The tables that we have used are first generated in the site. Then those can be accessed by the host PC. Four types of commands are allowed from the host to the site.
We will be using the blood_inventory table as an example to show the 4 types of commands which were used to access the tables in the site PC.

Select Command

```
SQL> select * from blood_inventory@site_link;

       DID  BAGNUMBER HEAMOGLOBIN  PLATELETS
---------- ---------- ----------- ----------
         1       2001          14         55
         2       2002          15         52
         3       2003          16         56
         4       2004          18         59
         5       2005          14         45
         6       2006          17         61
         7       2007          15         49
         8       2008          10         65
         9       2009          12         50
        10       2010          14         55

10 rows selected.
```

Update Command

Here we have updated the DID=10 to DID=21.

```
SQL> update blood_inventory@site_link set DID=21 where DID=10;

1 row updated.

SQL> select * from blood_inventory@site_link;

       DID  BAGNUMBER HEAMOGLOBIN  PLATELETS
---------- ---------- ----------- ----------
         1       2001          14         55
         2       2002          15         52
         3       2003          16         56
         4       2004          18         59
         5       2005          14         45
         6       2006          17         61
         7       2007          15         49
         8       2008          10         65
         9       2009          12         50
        21       2010          14         55

10 rows selected.
```

Delete Command

Here we have deleted the row of DID=21.

```
SQL> delete from blood_inventory@site_link where DID=21;

1 row deleted.

SQL> select * from blood_inventory@site_link;

       DID  BAGNUMBER HEAMOGLOBIN  PLATELETS
---------- ---------- ----------- ----------
         1       2001          14         55
         2       2002          15         52
         3       2003          16         56
         4       2004          18         59
         5       2005          14         45
         6       2006          17         61
         7       2007          15         49
         8       2008          10         65
         9       2009          12         50

9 rows selected.
```

Insert Command

Lastly we have inserted a new row using DID = 11, BagNumber = 2011, Haemoglobin = 15 and Paltelets = 56

```
SQL> insert into blood_inventory@site_link values(11, 2011, 15,56);

1 row created.

SQL> select * from blood_inventory@site_link;

       DID  BAGNUMBER HEAMOGLOBIN  PLATELETS
---------- ---------- ----------- ----------
         1       2001          14         55
         2       2002          15         52
         3       2003          16         56
         4       2004          18         59
         5       2005          14         45
         6       2006          17         61
         7       2007          15         49
         8       2008          10         65
         9       2009          12         50
        11       2011          15         56

10 rows selected.
```

Functions and Procedures

Function may have used in this project are described below:
countBagNums()

- Parameter: matchBloodGroup
- Return: totalBags
- Description: Takes blood group as input and return the total number of bags of these type of blood group. It calculates the total number of bags using a subquery. Count the bag number from BLOOD_INVENTORY table by matching the donor id with the donor id of DONOR table where the donor id is taken to the corresponding blood group.

Procedure which has been used in this project is described below:
isEligible()

- Parameter: ID and presentDate
- Return: There is no return type in procedure.
- Description: This procedure check the eligibility for donating blood of a donor. The date of donating blood has been fetched from donation details which takes donor id as input.The output date has been summed with 4 months and checked either it is greater than or less than present date, if resultant date is greater than present then eligibility is canceled else confirmed.

Triggers

We have used 3 triggers in this project. The short description of these are given below:
trigDonorInsert
After inserting any value in the DONOR table a trigger trigDonorInsert is created. Inside the trigger new donor id is stored into a variable named ID. Then the new donor id is inserted into RECIPIENT, BLOOD_INVENTORY and DONATION_DETAILS table.
trigDonorDelete
After deleting any tuple in the DONOR table a trigger trigDonorDelete is created. Inside the trigger old donor id is stored into a variable named ID. Then the tuple of RECIPIENT, BLOOD_INVENTORY and DONATION_DETAILS table are deleted where the donor id of these three tables are equal to the old donor id.
trigDonorUpdate
After updating any value in the DONOR table a trigger trigDonorUpdate is created. Inside the trigger new donor id and old donor id are stored into two variables named ID1 and ID2. Then the new donor id is updated into RECIPIENT, BLOOD_INVENTORY and DONATION_DETAILS table.

Exception

This project asking a number for different option. If user press '1' then the system goes to searchFile.sql file where user can search any information. If user press '2' then the system goes to insertFile.sql file where user can insert new donor information. If press '3' user can update existing donor information. If press '4' then the system approve user for deleting any donor information. If press any other digit without above digit then a user define exception named wrong_value will be raised which will show you a warning message.

Discussion

Finally it can be concluded that, we were able to create a blood bank searching system. This system allows to search, insert and update the information.