****

**Bahir Dar Institute of Technology, Bahir Dar University**

**Faculty of Computing**

**Departments of Computer Science**

**Project title: - Blood Donation Data Management system**

**Name of group members ID**

**ABEL BERHANU------------------------------------------------------------------1309026**

**ANMUT ALEMNEH---------------------------------------------------------------1306126**

**YISHAK ABREHAM--------------------------------------------------------------1308824**

**NEJAT LEJEBA--------------------------------------------------------------------1308181**

**NIGST W/MICHAEL-----------------------------------------------------------1308041**

**Submitted to: - BELETE**

**BAHIR DAR, ETHIOPIA**

**20/08/****2014 E.C**

Table of Contents

[Abstract 4](#_Toc2153338)

[Chapter one 5](#_Toc2153339)

[1.Introduction 5](#_Toc2153340)

[1.1. Background of the system 5](#_Toc2153341)

[1.2. Problems of current system 6](#_Toc2153342)

[1.3. Objectives 7](#_Toc2153343)

[1.3.1. General objectives 7](#_Toc2153344)

[1.3.2. Specific objective 7](#_Toc2153345)

[1.4. Scope of the project 7](#_Toc2153346)

[1.5. Limitations 7](#_Toc2153347)

[1.6. Significance of the System 8](#_Toc2153348)

[1.7. Methodology 9](#_Toc2153349)

[1.7.1 interview 9](#_Toc2153350)

[1.7.2 observation 9](#_Toc2153351)

[1.7.3 Document analysis 9](#_Toc2153352)

[Chapter two 10](#_Toc2153354)

[Database design 10](#_Toc2153355)

[2.1. Conceptual database design 10](#_Toc2153356)

[2.1.1. Identify entity& attribute 10](#_Toc2153357)

[2.1.2. Construct schema 13](#_Toc2153358)

[2.1.3. Role of attribute in relationship 15](#_Toc2153359)

[2.1.4. Entity, relationship, cardinality &participation 15](#_Toc2153360)

[2.1.5. ER\_digram 18](#_Toc2153362)

[2.2. Logical data base design 19](#_Toc2153363)

[2.2.1. Converting ER-diagram into Relation 20](#_Toc2153364)

[2.2.2 Normalization 21](#_Toc2153366)

[2.3. Physical database design 23](#_Toc2153367)

[2.3.1. Physical implementation on SQL server 23](#_Toc2153368)

[2.3.2. Database diagrams of Blood Donation Data Management System 31](#_Toc2153369)

Chapter Three 32

3.1 Conclusion--------------------------------------------------------------------------------------------32

Appends

Sample inserted data

**Abstract**

The BLOOD DONATION DATA MANAGEMENT SYSTEM is great project. this project is designed for successful completion of project on blood donation data management system. the basic building aim is to provide blood donation service. Blood Donation Data Management System (BDDMS) is a DBMS based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

**Chapter one**

**1.Introduction**

The ‘Blood Donation Data Management System’ allows us to keep records of donor blood and also available blood when requested by the acceptor. By using this system searching the available blood becomes easy and saves lot of time than the manual system. It will store, retrieve, recover and analyze information concerned with all the information of the blood within a blood bank. This system is developed in a manner that it is manageable, time effective, cost effective, flexible and much man power is not required. For this project we have tried to show how we can improve the existing system of blood donation management using the tools and knowledge we got from our SQL database management class.

* 1. **Background of the system**

There are lots of communication gap among patients (accepting blood), donors (who donate blood), blood banks and hospitals in our country. Medical facility is not available equally in every part of the country. If someone needs blood, first of all he searches it within his family members, then nearest hospitals and blood banks. If they cannot manage blood in these ways, it is really hard for them to contact other people to collect blood in a short time, hence there is a need for a blood donor information system (Akkas, 2015).

The blood donor information management system offers functionalities to quick access to register the donor, and collect donor details from various parts of the Provinces. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked. In this system we are providing an efficient search for any patient who needs the blood in a particular city, name, and blood groups as fast as possible. Blood Bank or the Hospital accepts the donated blood, only if donor satisfies some standard conditions.

In Western countries, blood is usually collected from donors, i.e., unpaid individuals who give blood voluntarily. Hence the mechanism of matching a patient who needs blood, to a particular donor whom blood matches with that of the patient, is of paramount important (Giuliana, 2010).

Donors, Blood Collection and Screening process starts with the arrival of the donor at the blood center. Donors can be divided in returning donors, who donate on an almost regular basis, and walk-in donors, who are entering the system occasionally or for the ﬁrst time. In any case, donations can be made after a deﬁned rest period from the previous one, which is deﬁned by law. As donors have a crucial importance in the system, their availability, frequency and motivation have been studied from both a statistical and a social perspective. Social Aspects The main reasons for blood donation and their relative importance have been studied by Bani and (Giussani, 2011).

Blood is classiﬁed into groups (A, B, O, and AB) and based on the Rhesus factor (Rh+ or Rh-), and each donor should be correctly matched with the patient who receives his/her blood. Moreover, as it may transmit diseases, blood must be screened before utilization. Generally, there are two types of donations: whole-blood donation, in which the whole blood is directly collected in a plastic bag, and apheresis, i.e., the donation of speciﬁc components in which a mechanical gathering unit decays the required blood parts. Blood requires particular precautions for collection and storage, and its shelf life from donation to utilization is limited, thus requiring a continuous feeding of the system (*Greening et al, 2010*).

**1.2. Problems of current system**

Due to the fact that the present system of processing and accessing the blood donor records is done manually and information is not dissimulated on time to the public. The following are the problem found on the system: Missing of Recordin the current blood donor information system which makes use of the manual system to record information. With the manual system, there are problems in managing the donors' records. The records of the donor might not be kept safe and there might be missing of donor's information or record due to human error or disasters. Besides that, errors might occur when the staff keeps more than one record for the same donor and there is no Centralized Database to keep the donors' records

Lack of immediate retrievalof data due to the fact that there is no computerized database of volunteer donors. So, it becomes really tedious for a person to search blood in case of emergency. The only option is to manually search and match donors and then make phone calls to every donor, Time Consumingn Retrieving of a volunteer donor records due to the fact that to find out about the donor history, the user has to go through various registers.

**1.3. Objectives**

**1.3.1. General objectives**

The aim of this study is to develop system that will automate the process of searching for blood donors in case of emergencies.

**1.3.2. Specific objective**

The specific objectives are listed below:

1. To computerize all details regarding blood donor details and matches them to a patient.
2. To automate the process of contacting donors and maintain their records effectively.
3. To manage current blood group of the donors and maintaining new events.
4. To provide Security of data this is very important due to the fact that hackers and intruders are very disruptive to any kind of system, such as Standalone or Web based Systems. So, this system will be very secures rather than other systems which uses the manual recording system.

**1.4. Scope of the project**

The scope of this study is to develop a system that will identify a blood donor, send SMS to that donor and establish a quick communication between the donor and the patient in critical situation. The system will be a web-based responsive database application system that supports three categories of users Admin, Patients and donors and each user activities will run independently without affecting the operation of another. The system will also integrate the online reservation module that will enable patients request for blood online

**1.5. Limitations**

The major concern each blood bank has is to monitor the quality of the blood and monitor the people who donates the blood, that is ‘donors. But this a tough job. The existing system will not satisfy the need of maintaining quality blood and keep track of donors. To overcome all these limitations, we introduced a new system called ‘Blood Donation Management System’.

The **‘Blood Donation Data Management System’** allows us to keep track of quality of blood and also keeps track of available blood when requested by the acceptor. The existing systems are Manual systems which are time consuming and not so effective. ‘Blood Bank Management system’ automates the distribution of blood. This database consists of thousands of records of each blood bank.

By using this system searching the available blood becomes easy and saves lot of time than the manual system. It will hoard, operate, recover and analyze information concerned with the administrative and inventory management within a blood bank. This system is developed in a manner that it is manageable, time effective, cost effective, flexible and much man power is not required.

**1.6. Significance of the System**

The need for an Automated Blood Donor’s Information Management System can be summarized as follows: -

1. *Immediate retrieval of information****:* -** The main objective of the proposed system is to provide a quick and efficient retrieval of information. Any type of information would be available whenever users require them.
2. *Immediate storage of information****:* -** In manual system, lots of problems are encountered in trying to store large amount of information.
3. *Create Awareness****:* -** The Proposed automated system will be used to create an effective channel of addressing the public about the blood donor programs and it importance to the society.
4. *Accuracy****:* -** The level of accuracy in the proposed automated system will be higher. All operations would be done correctly and accurately. In practice, errors are not completely eliminated, they are reduced.
5. *Reliability and Security:* - The reliability of the proposed system will be high as information is stored properly and securely.
6. *No Redundancy****:* -** In the proposed system paramount care would be taken to ensure that no information is repeated anywhere, in storage. This would assure economic use of storage space and consistency in the data stored.
7. *Easy to Operate****:* -** The system should be easy to operate and should be such that it can be developed within a short period of time and fit the limited budget of the user.

**1.7. Methodology**

The development methodologies are the various processes or methodologies that are being selected for the development of the project depending on the project’s aims and goals.

* Here more emphasis is given to the **design phases** of the system development life cycle.

The primary aim of each phase is as follows:

* ***Conceptual database design* -**to build the conceptual representation of the database, which has the identification of the important entities, relationships, and attributes.
* ***Logical database design -*** to convert the conceptual representation to the logical structure of the database, which includes designing the relations.
* ***Physical database design -*** to decide how the logical structure is to be physically implemented (as base relations) in the target Database Management System (DBMS).
* **Implementation**: the testing and deployment of the designed database for use.

In this project, the methods of data collection adopted include the following: Interview, World Wide Web, references to published and unpublished collection.

**1.7.1 interview**

We used this method as a primary source of information; In writing this project some few persons were interview to get some insight about blood donation and to know if there is a general blood donation program.

**1.7.2 observation**

We used this as a primary source of our data for this project we saw firsthand how the data was being handled and managed and we used that for our design.

**1.7.3 Document analysis**

The proposed system will help the people who are in need of a blood by giving them all details of blood group availability or regarding the donors with the same blood group. The system works 24x7 so user can get information of blood donor any time and capable of supporting any number of donor’s data in any geographical area and each donor’s activity will run independently and not affect the operation of another donor.

**Chapter two**

***Database design***

**2.1. Conceptual database design**

is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships. In this data modeling level, there is hardly any detail available on the actual database structure. Business stakeholders and data architects typically create a conceptual data model.

The 3 basic tenants of Conceptual Data Model are

**Entity**: A real-world thing

**Attribute**: Characteristics or properties of an entity

**Relationship**: Dependency or association between two entities

**2.1.1. Identify entity& attribute**

**ENTITIES**

* Entities are real world physical or logical object
* Are persons, places, things etc. which the organization has to deal with.

**ATTRIBUTES**

* Are items of information which characterize and describe entities.
* Attributes will give rise to recorded items of data in the database.
* we need to know such things as:
* Attribute name (be explanatory words or phrases)
* The domain from which attribute values are taken (A DOMAIN is a set of values from which attribute values may be taken.) Each attribute has values taken from a domain.

**The entities are given below with its attributes:**

1. ***Donor***

donor\_id , donor\_first\_name ,donor\_last\_Name, age, gender ,addres ,donor\_blood\_type , medical\_status ,receptionists (foreign key references receptionists(emp\_id))

The donor is the person who donates blood, on donation a donor id (donor \_ID) is generated and used as primary key to identify the donor information. Other than that name, age , sex , blood group, phone number and will be stored in database under Donor entity.

***2. receptionists***

emp\_id, emp\_first\_name ,emp\_last\_name ,position,, salary int, emp\_addres

The receptionists is a person who registers the blood donor and recipients and the receptionists \_ entity has emp\_ID which is primary key along with emp\_first\_ name and receptionists\_phone number will also be stored in the data

base under receptionists entity.

***3.blood***

b\_id, store\_blood\_type, donors int foreign key references donors(donor\_id), recipients int foreign key references recipients(rec\_id ), blood\_banks int foreign key references blood\_banks(bb\_id))

In data base, under Blood entity we will store the information of blood samples which are available in the blood bank.

***4, recipients***

rec\_id, rec\_first\_name ,rec\_last\_name , gender , rec\_blood\_type ,age int, hospitals int foreign key references hospitals (hos\_id)

The Recipient is the person who recivies blood from blood bank, when blood is given

to a recipient a rericipient ID (rec\_ID) is generated and used as primary key for

the recipient entity to indentify blood recipients information. Along with it name ,age, blood group , phone number,under recipient entity.

***5, blood\_banks***

bb\_id ,bb\_location ,hospitals ,BB\_ BB\_managers , receptionists

***6. hospitals***

hos\_id , hos\_name, hos\_location

In the data base, under hospitals entity we will store the information of hospitals. In this hos\_ID and name , location . We will store hospital name and the hos\_location at the hospital.

***7. BB\_managers***

m\_id , first\_name

The blood bank manager is the person who takes care of the avaible blood samples in the blood bank, he is also resposible for handaling blood requests from recipients and hospitals. Blood manager has a unique indentfication number (m\_ID) used as primary key of blood bank manager will be stored in data base under BB\_Manager entity.

2.1.2. Construct schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Schema

* The overall design of the database is called schema or description of database.
* Schema is a physical representation of data which is present in the database management system.
* In simple words we can call a schema the structure of any database.
* It defines how the data was stored in a database and also shows the relationship among those data, but it does not show the data present in those tables.
* The database schema includes the definition of the database, entities and the components.
* Schema of a database can only modify the DDL statement but does not change by performing certain operations like insertion, updating, and deletion.
* Database schema explains the integrity constraints of the database, domains of all attributes, foreign, and primary key of all the relations.

The goal of the three-schema architecture, illustrated in Figure 2.1 is to separate the user applications from the physical database. In this architecture, schemas can be defined at the following three levels:

**1 *External Level***: Users' view of the database. It describes that part of database that is relevant to a particular user. Different users have their own customized view of the database independent of other users.

**2 *Conceptual Level:*** Community view of the database. Describes what data is stored in database and relationships among the data along with the business constraints.

**3** ***Internal Level:*** Physical representation of the database on the computer. Describes how the data is stored in the database.

Blood Donation Management System

Donor

Add Donor

**1**

Update Donor Profile

Blood

donation

Database

Blood group Search

Next Donations Period

Donor

Search

**3**

Area of the blood group

Add Precipitant

Precipitant

Blood request details

Request Blood

**2**

Figure 2.1 the three schema Architecture of the Blood donation management system

**2.1.3. Role of attribute in relationship**

The particular properties of entities are called attributes. Attributes represent what we want to know

about entities. In this blood donation management system the Donor entities mat be described by Donor\_id( primary ), Name( Composite ), Age, gender, Address ,medical conditions. These attributes hold values that describe each occurrence, and represent the main source of data stored in the database.

Simple and composite attributes

Simple attributes cannot be further subdivided. Examples of simple attributes include the ,gender ,blood\_type and donor\_id attributes for a donor. Simple attributes are sometimes called atomic attributes.

Composite attributes can be further divided to yield smaller components with an independent existence. For example, the name attribute of the donor and recipient entity with the name sub divided into fname,and lname

The majority of attributes are single-valued for this entities

**2.1.4. Entity, relationship, cardinality &participation**

**Relationship**

* Donor -----------------> [Donate] --------> Blood
* A Receptionists -----> [Register]-------> Donor
* Blood Bank-----------> [Store] ---------> Blood
* BB.managers ---------> [Manages] ------>Blood Bank
* Hospital --------------> [Orders] --------> Blood Bank
* Recipient-------------> [Received]------->Blood
* Recipient-------------> [Asks]------------>Hospital

**Structural Constraints on Relationship**

One-to-one relationship

Donate

Blood

Donor

1.1 1.1

* A Donor is associated with at most one Blood via the relationship Donate.
* A Blood is associated with at most one Donor via Donate.

Many to Many relationship

Register

Receptionist

Donor

M.N M.N

* A Receptionist is Registers with several Donor via Register
* A Donor is associated with several Receptionist via Register

One to Many relationship

Blood Bank

Store

Blood

1.1 1.1

* A Blood is associated with at most one Blood Bank via Store
* A Blood Bank is associated with several Blood via Store

One-to-one relationship

Blood Bank

Manages

BB\_managers

1.1 1.1

* A BB\_managers is associated with at most one Blood Bank via Manages.
* A Blood Bank is associated with at most one BB\_managers via Manages.

Many to Many relationship

Orders

Blood

Hospital

M.N M.N

* A Hospital is associated with several Blood Bank via Orders
* A Blood Bank is associated with several Hospital via Orders

Many to Many relationship

Received

Blood

Recipient

M.N M.N

* A Recipient is associated with at most one Blood via Received.
* A Blood is associated with at most one Recipient via Received.

One-to-one relationship

Asks

Hospital

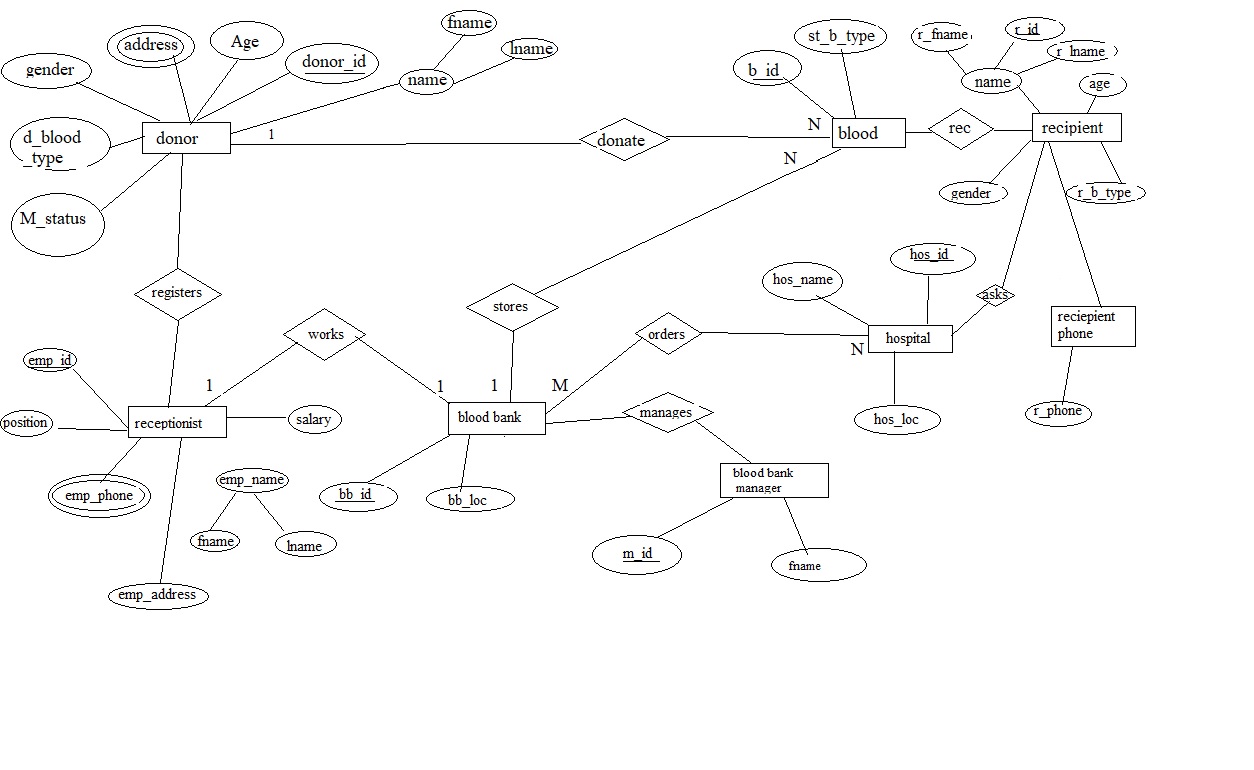
Recipient

1.1 1.1

* A Recipient is associated with at most one via Asks
* A Hospital is associated with several Recipient via Asks

**2.1.5. ER\_digram**

**BLOOD DONATION E - R diagram:**



**2.2. Logical data base design**

**Converting ER Diagram in to Relational Database**

Three basic rules to convert ER into tables or relations:

Rule 1: Entity Names will automatically be table names

Rule 2: Mapping of attributes: attributes will be columns of the respective tables.

* Atomic or single-valued or derived or stored attributes will be columns
* Composite attributes: the parent attribute will be ignored and the decomposed attributes (child attributes) will be columns of the table.

Rule 3: Relationships: relationship will be mapped by using a foreign key attribute. Foreign key is a primary or candidate key of one relation used to create association between tables.

**Converting ER Diagram in to Relational Database**

Three basic rules to convert ER into tables or relations:

Rule 1: Entity Names will automatically be table names

Rule 2: Mapping of attributes: attributes will be columns of the respective tables.

* Atomic or single-valued or derived or stored attributes will be columns
* Composite attributes: the parent attribute will be ignored and the decomposed attributes (child attributes) will be columns of the table.

Rule 3: Relationships: relationship will be mapped by using a foreign key attribute. Foreign key is a primary or candidate key of one relation used to create association between tables.

**2.2.1. Converting ER-diagram into Relation**

**Relational Table**

Donor-Donate-Blood

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **donor\_id\*** | **Donor\_first\_name** | **Donor\_last\_name** | **age** | **gender** | **gender** | **Donor\_blood\_type** | **b\_id\*** |

Receptionist-resistor-Donor

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **emp\_id\*** | **emp\_first\_name** | **emp\_last\_name** | **age** | **gender** | **gender** | **Donor\_blood\_type** | **donor\_id\*** |

Recipient-receive-Blood

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **rec\_id\*** | **rec\_first\_name** | **donor\_last\_name** | **gender** | **gender** | **rec\_blood\_type** | **b\_id\*** |

Blood\_bank –store-blood

|  |  |  |  |
| --- | --- | --- | --- |
| bb\_id\* | b\_id | blood\_type | bb.addrese |

Recipient-Asks-Hospital

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hos\_id\* | rec\_id | Fname | Lname | age | gender | Blood\_type |

Hospital-request-Blood\_bank

|  |  |  |
| --- | --- | --- |
| hos\_id\* | bb\_id | Location |

BB.Managers-manages-Blood\_Bank

|  |  |  |
| --- | --- | --- |
| m\_id\* | bb\_id | bb.Location |

**2.2.2 Normalization**

Normalization

Database normalization is a series of steps followed to obtain a database design that allows for consistent storage and efficient access of data in a relational database. These steps reduce data redundancy and the risk of data becoming inconsistent.

* **1NF (First Normalization Form)**

We have two ways of achieving this:

1. Putting each repeating group into a separate table and connecting them with a primary key-foreign key relationship

2. Moving these repeating groups to a new row by repeating the nonrepeating attributes known as “flattening” the table. If so then Find the key with which you can find all data

***When the Unnormalized form is converted to 1NF we make the following changes to our database***

**Recipient**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **rec\_Id\*** | **Rec\_first\_Name** | **Rec\_last\_Name** | **Age** | **Gender** | **Blood\_Type** |

**Recipient -Phone**

|  |  |
| --- | --- |
| **rec\_Id \*** | **Rec\_Phone** |

**Receptionists**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **emp\_first\_Name** | **emp\_first\_Name** | **Emp\_Id \*** | **position** | **salary** | **emp\_address** |

**Receptionists-Phone**

|  |  |
| --- | --- |
| **Emp\_Id \*** | **Emp\_Phone\_No** |

**Hospital**

|  |  |  |
| --- | --- | --- |
| **Hos\_Id** | **Hos\_name** | **hos\_location** |

**Hospital -Phone**

|  |  |
| --- | --- |
| **Hos\_Id \*** | **Phone\_No** |

* **2NF(Second Normalization Form)**

No partial dependency of a non- key attribute on part of the primary key. This will result in a set of relations with a level of Second Normal Form. Any table that is in 1NF and has a single-attribute (i.e., a non-composite) key is automatically also in 2NF.

**Donor**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **donor\_id\*** | **Donor\_first\_name** | **Donor\_last\_name** | **age** | **gender** | **gender** | **Donor\_blood\_type** | **Md\_status** |

**Blood**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Blood\_Id** | **Blood\_Type** | **donor\_ID\*** | **BB\_ID\*** | **Rec\_ID\*** |

**Receptionists**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **emp\_first\_Name** | **emp\_first\_Name** | **Emp\_Id \*** | **position** | **salary** | **emp\_address** |

**Receptionists-Phone**

|  |  |
| --- | --- |
| **Emp\_Id \*** | **Emp\_Phone\_No** |

**Recipient**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **rec\_Id\*** | **Rec\_first\_Name** | **Rec\_last\_Name** | **Age** | **Gender** | **Blood\_Type** |

**Recipient -Phone**

|  |  |
| --- | --- |
| **rec\_Id \*** | **Rec\_Phone** |

**Blood\_Bank**

|  |  |  |
| --- | --- | --- |
| **BB\_Id** | **Location** | **hos\_ID\*** |

**Hospital**

|  |  |  |
| --- | --- | --- |
| **Hos\_Id** | **Hos\_name** | **hos\_location** |

**Hospital -Phone**

|  |  |
| --- | --- |
| **Hos\_Id \*** | **Phone\_No** |

**3NF(Third Normalization Form)**

A Table (Relation) is in 3NF If

* It is in 2NF and
* There are no transitive dependencies between a primary key and non-primary key attributes

***Since there are no transitive dependencies between primary and non-primary attributes doing this is stage useless, thus we use our 2NF as final normalized form.***

**2.3. Physical database design**

**2.3.1. Physical implementation on SQL server**

* **IMPLEMENTATION**

### Create a Database and implement tables, relationships and constraints.

create database blood\_donation\_data\_mana\_sys --drop database blood\_donation\_data\_mana\_sys

use blood\_donation\_data\_mana\_sys

create table receptionists( -- drop table receptionists

emp\_id int primary key,

emp\_first\_name varchar(25),

emp\_last\_name varchar(25),

position varchar(20),

salary int,

emp\_addres varchar(30)

)

create table donors( -- drop table donors

donor\_id int primary key,

donor\_first\_name varchar(100),

donor\_last\_Name varchar(100),

age int,

gender varchar(20),

addres varchar(50),

donor\_blood\_type varchar(10),

medical\_status varchar(25),

receptionists int foreign key references receptionists(emp\_id)

)

create table hospitals( -- drop table hospitals

hos\_id int primary key,

hos\_name varchar(50),

hos\_location varchar(20)

)

create table recipients( -- drop table recipients

rec\_id int primary key,

rec\_first\_name varchar(100),

rec\_last\_name varchar(100),

gender varchar(10),

rec\_blood\_type varchar(10),

age int,

hospitals int foreign key references hospitals (hos\_id)

)

create table BB\_managers(

m\_id int primary key,

first\_name varchar(50),

)

create table blood\_banks (

bb\_id int primary key,

bb\_location varchar(25),

hospitals int foreign key references hospitals (hos\_id),

BB\_managers int foreign key references BB\_managers (m\_id),

receptionists int foreign key references receptionists (emp\_id)

)

create table blood(

b\_id int primary key,

store\_blood\_type varchar(20),

donors int foreign key references donors(donor\_id),

recipients int foreign key references recipients(rec\_id ),

blood\_banks int foreign key references blood\_banks(bb\_id)

)

create table receptionist\_phones( --drop table receptionist\_phones

emp\_id int foreign key references receptionists(emp\_id) ,

phone\_emp int

)

create table hospital\_phones(

hos\_id int foreign key references hospitals(hos\_id) , --drop table hospital\_phones

hos\_phone int ,

)

create table recipient\_phones( -- drop table

rec\_id int foreign key references recipients(rec\_id ),

rec\_phone int

)

insert into receptionists values(501,'SALADIN','YESUF','DOCTOR',9500,'AA')

insert into receptionists values(502,'KALKIDAN','TESFAYE','NURSE',3500,'ADAMA')

insert into receptionists values(503,'AZEB','KETEM','DOCTOR',9500,'BDR')

insert into receptionists values(504,'EYOB','YESHITLA','LABRATORIST',5000,'AWASH')

insert into receptionists values(505,'HANA','TEKA','LABRATORIST',4500,'HAWASA')

insert into receptionists values(506,'AHMED','YESUF','DOCTOR',9300,'MEKELE')

insert into receptionists values(507,'HAYMANOT','YESUF','NURSE',3500,'AA')

insert into receptionists values(508,'MEKDES','GETU','LABRATORIST',5500,'MOJO')

insert into receptionists values(509,'ABREHAM','GETNET','NURSE',5500,'AA')

insert into receptionists values(510,'AYELE','ALEM','DOCTOR',9500,'BDR')

insert into donors values(101,'ABEL', 'DINKU',25,'M','BDR','A+','Pass',504)

insert into donors values(102,'SARA', 'ASMAMAW',19,'F','AA','A-','PASS',507)

insert into donors values(103,'NARDOS', 'GEBRE',23,'F','HAWASA','B+','PASS',503)

insert into donors values(104,'YORDANOS', 'Desta',22,'F','MOJO','B-','FALL',508)

insert into donors values(105,'ASCHALEW','DEBEBE',28,'M','DEBREMARKOS','AB+','PASS',509)

insert into donors values(106,'NAHOM', 'TESFAYE',38,'M','AWASH','AB','PASS',502)

insert into donors values(107,'YONAS', 'ADENAGR',28,'M','ADAMA','O+','PASS',510)

insert into donors values(108,'FREZER', 'ZEWDU',32,'M','MEKELE','O-','PASS',506)

insert into donors values(109,'GENET', 'TESFAYE',26,'F','BDR','O+','FAILL',502)

insert into donors values(110,'HABTAMU', 'ESUBALEW',42,'M','AA','A+','PASS',501)

insert into hospitals values(201,'FELEGEHIWET\_HOSPITAL','BDR')

insert into hospitals values(202,'GANDI\_HOSPITAL','BDR')

insert into hospitals values(203,'PAULOS\_HOSPITAL','AA')

insert into hospitals values(204,'YEKATIT12\_HOSPITAL','AA')

insert into hospitals values(205,'EDEN\_HOSPITAL','MEKELE')

insert into hospitals values(206,'MESERET\_HOSPITAL','ADAMA')

insert into hospitals values(207,'KAL\_HOSPITAL','MOJO')

insert into hospitals values(208,'TESFAYE\_HOSPITAL','HAWASA')

insert into hospitals values(209,'ADDISKETEMA\_HOSPITAL','BDR')

insert into hospitals values(210,'ARADA\_HOSPITAL','AA')

insert into recipients values(301,'HABTAMU','YIHUN','M','O-',45,203)

insert into recipients values(302,'HENOK','WOREDE','M','O+',28,201)

insert into recipients values(303,'KALEB','TESHALE','M','AB-',25,210)

insert into recipients values(304,'YONAS','MEHARI','M','AB+',35,206)

insert into recipients values(305,'MAHLET','MOLLA','F','B-',32,204)

insert into recipients values(306,'ZELALEM','YITAYEW','M','B+',45,205)

insert into recipients values(307,'MELAK','GEDEFAYE','M','A-',18,205)

insert into recipients values(308,'MELAT','TESHOME','F','A+',42,202)

insert into recipients values(321,'kali','Abet','M','O+',45,203)

insert into recipients values(311,'HABTAM','Degefa','M','AB+',45,203)

insert into BB\_managers values(801,'TESHOME')

insert into BB\_managers values(802,'KALEB')

insert into BB\_managers values(803,'YIHUN')

insert into BB\_managers values(804,'SALADIN')

insert into BB\_managers values(805,'AHMED')

insert into BB\_managers values(806,'YORDANOS')

insert into BB\_managers values(807,'AZEB')

insert into BB\_managers values(808,'AYELE')

insert into BB\_managers values(809,'DEBEBE')

insert into BB\_managers values(810,'FREZER')

insert into blood\_banks values(401,'AA',203,806,501)

insert into blood\_banks values(402,'BDR',206,805,510)

insert into blood\_banks values(403,'HAWASA',208,808,505)

insert into blood\_banks values(404,'MOJO',207,809,508)

insert into blood\_banks values(405,'AWASH',201,803,504)

insert into blood\_banks values(406,'ADAMA',203,804,502)

insert into blood\_banks values(407,'AA',203,807,507)

insert into blood\_banks values(408,'BDR',206,810,503)

insert into blood\_banks values(409,'MEKELE',208,801,506)

insert into blood\_banks values(410,'AA',205,802,509)

insert into blood values(701,'A-',102,305,402)

insert into blood values(702,'AB+',105,306,405)

insert into blood values(703,'O+',107,307,407)

insert into blood values(704,'B+',103,305,408)

insert into blood values(705,'B-',104,303,403)

insert into blood values(706,'A-',102,306,408)

insert into blood values(707,'O-',108,307,406)

insert into blood values(708,'O+',109,304,409)

insert into blood values(709,'O-',108,305,405)

insert into blood values(710,'AB-',106,301,406)

insert into receptionist\_phones values(501,0985741212)

insert into receptionist\_phones values(501,0988829458)

insert into receptionist\_phones values(502,0915741212)

insert into receptionist\_phones values(502,0969741212)

insert into receptionist\_phones values(503,0915741212)

insert into receptionist\_phones values(504,0910741212)

insert into receptionist\_phones values(507,0978741212)

insert into receptionist\_phones values(508,0965741212)

insert into receptionist\_phones values(509,0986741212)

insert into receptionist\_phones values(508,0936741212)

insert into receptionist\_phones values(505,0974741212)

insert into receptionist\_phones values(507,0974741214)

insert into receptionist\_phones values(509,0914741218)

insert into receptionist\_phones values(502,0952741212)

insert into receptionist\_phones values(504,0996741212)

insert into hospital\_phones values(202,0909779898)

insert into hospital\_phones values(202,0912779898)

insert into hospital\_phones values(203,0913779898)

insert into hospital\_phones values(204,0985779898)

insert into hospital\_phones values(207,0910779898)

insert into hospital\_phones values(208,0936779898)

insert into hospital\_phones values(207,0935779898)

insert into hospital\_phones values(201,0909789898)

insert into hospital\_phones values(209,0945797898)

insert into hospital\_phones values(206,0909775298)

insert into hospital\_phones values(208,0998779898)

insert into hospital\_phones values(205,0945779898)

insert into hospital\_phones values(207,0999779898)

insert into hospital\_phones values(201,0996779898)

insert into recipient\_phones values(301,0932564333)

insert into recipient\_phones values(306,0952416347)

insert into recipient\_phones values(302,0954288998)

insert into recipient\_phones values(303,0942416347)

insert into recipient\_phones values(304,0912416347)

insert into recipient\_phones values(307,0950476747)

insert into recipient\_phones values(311,0952416347)

insert into recipient\_phones values(305,0988416347)

insert into recipient\_phones values(308,0942416347)

insert into recipient\_phones values(321,0936416347)

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

------------------------Displaying Tables-----------------------------------

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

select\*from receptionists

select\*from receptionist\_phones

select\*from donors

select\*from recipients

select\*from recipient\_phones

select\*from hospitals

select\*from hospital\_phones

select\*from BB\_managers

select\*from blood\_banks

select\*from blood

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

--------------------------------------Join-------------------------------------------

select \* from receptionists join receptionist\_phones on receptionists.emp\_id=receptionist\_phones.emp\_id

select donor\_first\_name,donor\_blood\_type,medical\_status,emp\_first\_name,position from donors join receptionists on donors.receptionists=receptionists.emp\_id where donor\_blood\_type='O+' or donor\_blood\_type='O-'

select hos\_name,bb\_location from hospitals join blood\_banks on hospitals.hos\_id=blood\_banks.hospitals

where hos\_location='AA' or bb\_location='BDR'

select first\_name,bb\_location from blood\_banks join BB\_managers on blood\_banks.BB\_managers=BB\_managers.m\_id

select emp\_first\_name,position,emp\_addres,bb\_location from receptionists join blood\_banks on receptionists.emp\_id=blood\_banks.receptionists

select donor\_first\_name,donor\_blood\_type,rec\_first\_name,store\_blood\_type from donors join blood on donors.donor\_id=blood.donors join

recipients on blood.recipients=recipients.rec\_id

select rec\_first\_name,rec\_blood\_type,store\_blood\_type from recipients join blood on recipients.rec\_id=blood.recipients

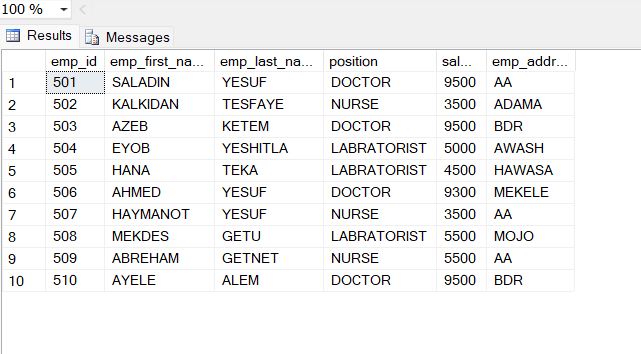
select hos\_name,hos\_phone from hospitals join hospital\_phones on hospitals.hos\_id=hospital\_phones.hos\_id

select rec\_first\_name,rec\_phone from recipients join recipient\_phones on recipient\_phones.rec\_id=recipient\_phones.rec\_id

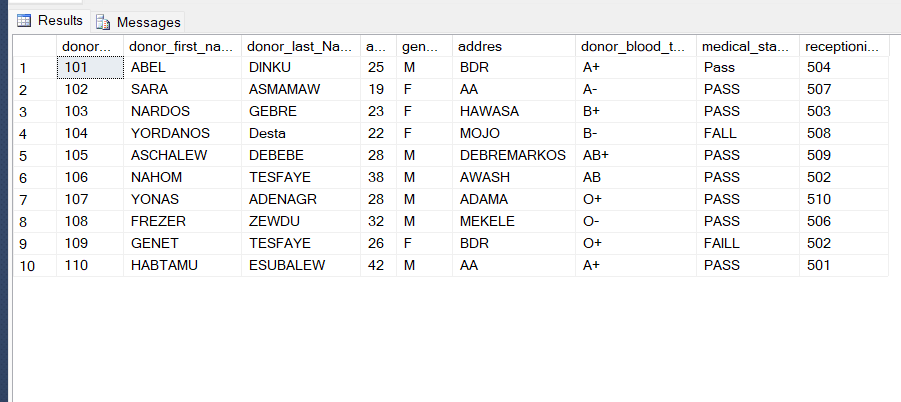
------------------------*Displaying Tables*-----------------------------------

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

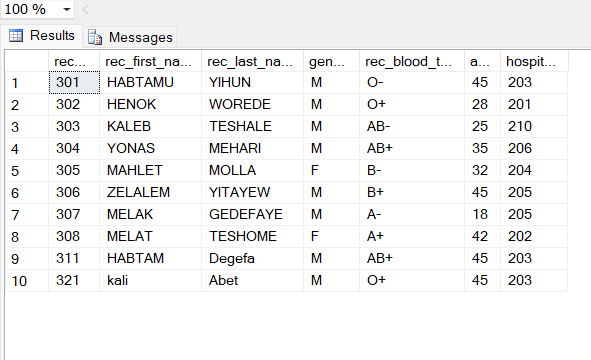
**select\*from receptionists**



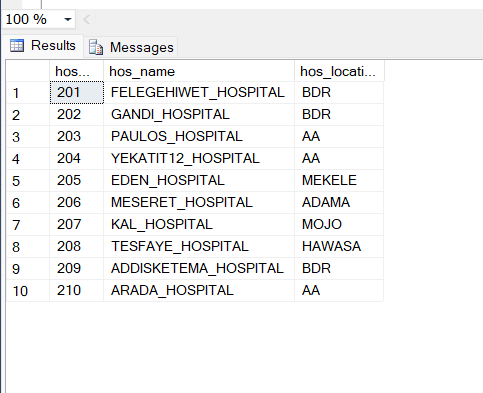
**select\*from donors**



**select\*from recipients**

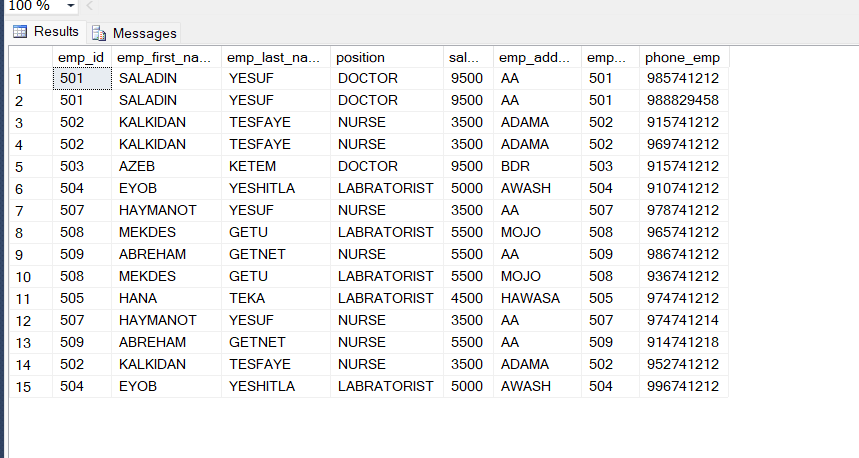


**select\*from hospitals**

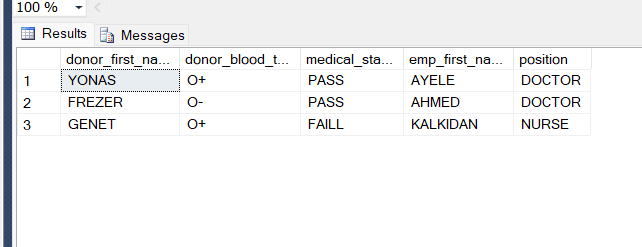


--------------------------------------Join-------------------------------------------

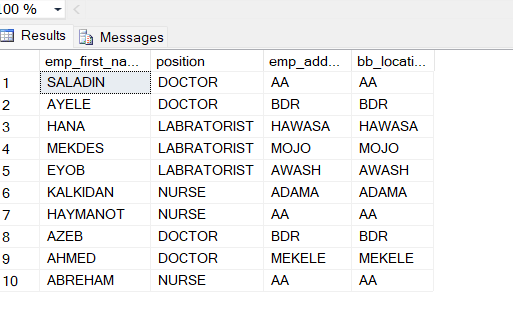
**select \* from receptionists join receptionist\_phones on receptionists.emp\_id=receptionist\_phones.emp\_id**



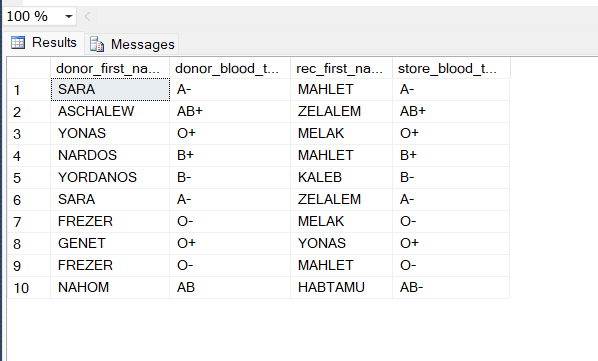
**select donor\_first\_name,donor\_blood\_type,medical\_status,emp\_first\_name,position from donors join receptionists on donors.receptionists=receptionists.emp\_id where donor\_blood\_type='O+' or donor\_blood\_type='O-'**



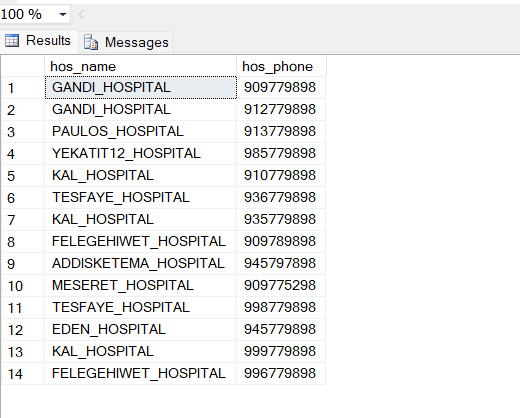
**select emp\_first\_name,position,emp\_addres,bb\_location from receptionists join blood\_banks on receptionists.emp\_id=blood**



**select donor\_first\_name,donor\_blood\_type,rec\_first\_name,store\_blood\_type from donors join blood on donors.donor\_id=blood.donors join recipients on blood.recipients=recipients.rec\_id**

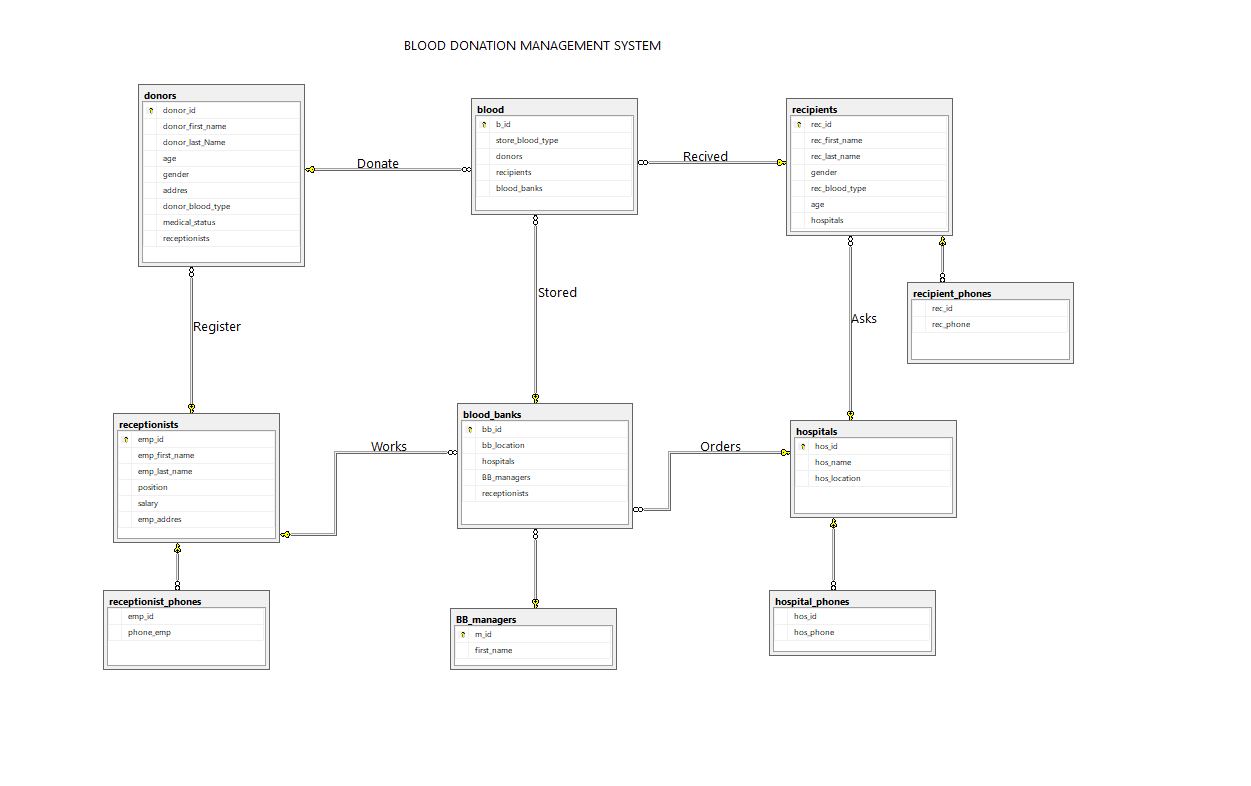


**select hos\_name,hos\_phone from hospitals join hospital\_phones on hospitals.hos\_id=hospital\_phones.hos\_id**



**2.3.2. Database diagrams of Blood Donation management system**

**Database Diagram**



**Chapter Three**

**3.1 Conclusion**

Our project well addressed the limitations of the existing system. We designed well organized database management system which is a challenging job in this era. We have built a database for a Blood Bank using Microsoft SQL Server. Before implementing the database, in the design phase, we have explored various features, operations of a blood donation to figure out required entities, attributes and the relationship among entities to make an efficient Entity Relationship Diagram (ERD). After analyzing all the requirements, we have created our ERD and then converted the ERD to relational model and normalized the tables. Using Microsoft SQL Server, we have created the tables for our database and inserted some sample values in the tables. Finally, we have executed sample queries on our database to check its performance to retrieve useful information accurately and speedily.

## REFERENCES

1. Organ Donation Management System - Chitra , Ramasamy (2007) - Melaka, Malaysia, UTeM.
2. Charity Management System - Sara Aljohani (2015)
3. DATABASE DESIGN, APPLICATION DEVELOPMENT, AND ADMINISTRATION (6thedition) – Michael Mannino
4. Blood Donation Management: Recent trends – Sunil K Joseph, Johnson Michael, Prem Jose Vazhacharickal
5. SQL server tutorial: <https://www.tutorialspoint.com/ms_sql_server/>
6. Notes from Prof. Hakki C Cankaya, University of Texas at Dallas, Dallas, Texas
7. Tool for ER diagram: Creately

[https://creately.com/app/?tempID=hqdgwjki1&login\_type=demo#](https://creately.com/app/?tempID=hqdgwjki1&login_type=demo%23) [8] SQL Implementation Tool: Microsoft SQL Server 2012

Appends

Sample inserted data