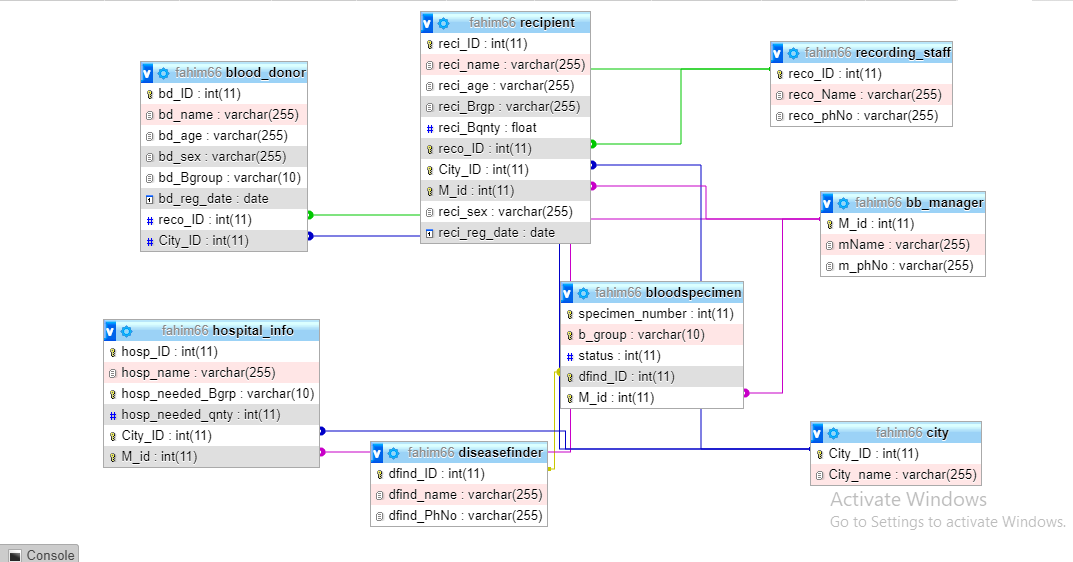
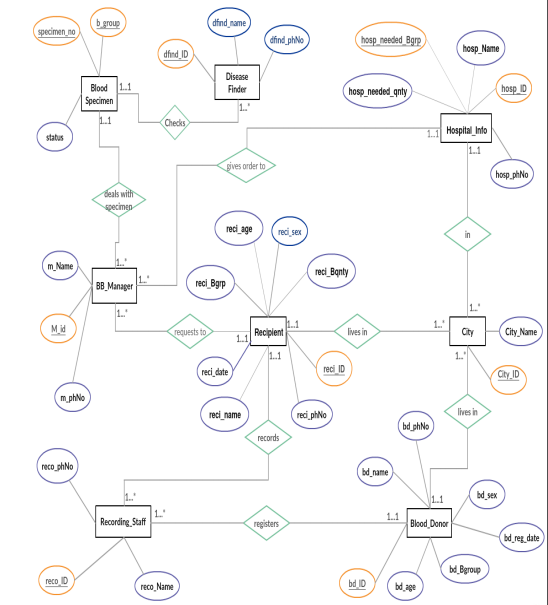
Proposal:

Blood banks collect, store and provide collected blood to the patients who are in need of blood. The people who donate blood are called ‘donors’. The banks then group the blood which they receive according to the blood groups. They also make sure that the blood is not contaminated. The main mission of the blood bank is to provide the blood to the hospitals and health care systems which saves the patient’s life. No hospital can maintain the health care system without pure and adequate blood. 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 Bank 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.

ER Diagram with Digital View tables



HAND ART ER-DIAGRAM:



INFORMATION OF ENTITIES :

In total we have eight entities and information of each entity is mentioned below:-

1. Blood\_Donor: (Attributes – bd\_ID, bd\_name, bd\_sex, bd\_age, bd\_Bgroup, bd\_reg\_date,

bd\_phNo)

The donor is the person who donates blood, on donation a donor id (bd\_ID) is generated and

used as primary key to identify the donor information. Other than that name, age , sex , blood

group, phone number and registration dates will be stored in database under Blood\_Donor

entity.

2. Recipient: (Attributes – reci\_ID, reci\_name, reci\_age, reci\_Bgrp, reci\_Bqnty , reci\_sex,

reci\_reg\_date, reci\_phNo)

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

recipient a rericipient ID (reci\_ID) is generated and used as primary key for the recipient entity

to indentify blood recipients information. Along with it name ,age, sex, blood group (needed),

blood quantity(needed) , phone number, and registration dates are also stored in the data base

under recipient entity.

3. BB\_Manager: (Attributes – m\_ID, m\_Name, m\_phNo)

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 along with name and

phone number of blood bank manager will be stored in data base under BB\_Manager entity.

4. Recording\_Staff : (Attributes – reco\_ID, reco\_Name, reco\_phNo)

The recording staff is a person who registers the blood donor and recipients and the

Recording\_Staff enitity has reco\_ID which is primary key along with recoder’s name and

recodrer’s phone number will also be stored in the data base under Recording\_Staff entity.

5. BloodSpecimen : (Attributes – specimen\_number, b\_group , status)

In data base, under BloodSpecimen entity we will store the information of blood samples which

are available in the blood bank. In this entity specimen\_number and b\_group together will be

primary key along with status attribute which will show if the blood is contaminated on not.

6. DiseaseFinder : (Attributes - dfind\_ID, dfind\_name, dfind\_PhNo)

In data base , under DiseaseFinder entity we will store the information of the doctor who checks

the blood for any kind of contaminations. To store that information we have unique

identification number (dfind\_ID) as primary key. Along with name and phone number of the

doctor will also be stored under same entitity.

7. Hospital\_Info : (Attributes – hosp\_ID, hosp\_name, hosp\_needed\_Bgrp, hosp\_needed\_Bqnty)P a g e | 7

In the data base, under Hospital\_Info entity we will store the information of hospitals. In this

hosp\_ID and hosp\_needed\_Bgrp toether makes the primary key. We will store hospital name

and the blood quantity reqiured at the hospital.

8. city: (Attributes- city\_ID, city\_name)

This entity will store the information of cities where donors, recipients and hospitals are

present. A unique indentification number (City\_ID) will be used as primary key to indefiy the

information about the city. Along with ID city names will also be stored under this entity.

RELATIONSHIP BETWEEN ENTITIES :

1. City and Hospital\_Info:

Relationship = “in”

Type of relation = 1 to many

Explanation = A city can have many hospital in it. One hospital will belong in one city.

2. City and Blood\_Donor:

Relationship = “lives in”

Type of relation = 1 to many

Explanation = In a city, many donor can live. One donor will belong to one city.

3. City and Recipient:

Relationship = “lives in”

Type of relation = 1 to many

Explanation = In a city, many recipient can live. One recipient will belong to one city.

4. Recording\_Staff and Donor:

Relationship = “registers”

Type of relation = 1 to many

Explanation = One recording staff can register many donors. One donor will register with

one recording officer.

5. Recording\_Staff and Recipient:

Relationship = “records”

Type of relation = 1 to many

Explanation = One recording staff can record many recipients. One recipient will be

recorded by one recording officer.

6. Hospital\_Info and BB\_Manager:

Relationship = “gives order to”

Type of relation = 1 to many P a g e | 8

Explanation = One Blood bank manager can handle and process requests from many

hospitals. One hospital will place request to on blood bank manager.

7. BB\_Manager and Blood Specimen:

Relationship = “deales with specimen”

Type of relation = 1 to many

Explanation = One Blood bank manager can manage many blood specimen and one

specimen will be managed by one manager.

8. Recipient and BB\_Manager:

Relationship = “requests to”

Type of relation = 1 to many

Explanation = One recipient can request blood to one manager and one manager can

handle requests from many recipients.

9. Disease\_finder and Blood Specimen:

Relationship = “checks”,

Type of relation = 1 to many

Explanation = A disease finder can check many blood samples. One blood sample is

checked by one disease finder.

Query of My Database:

-- phpMyAdmin SQL Dump  
-- version 5.0.2  
-- https://www.phpmyadmin.net/  
--  
-- Host: 127.0.0.1  
-- Generation Time: Apr 27, 2021 at 08:30 PM  
-- Server version: 10.4.14-MariaDB  
-- PHP Version: 7.4.10  
  
SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";  
START TRANSACTION;  
SET time\_zone = "+00:00";  
  
  
/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  
/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  
/\*!40101 SET NAMES utf8mb4 \*/;  
  
--  
-- Database: `fahim66`  
--  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `bb\_manager`  
--  
  
CREATE TABLE `bb\_manager` (  
 `M\_id` int(11) NOT NULL,  
 `mName` varchar(255) NOT NULL,  
 `m\_phNo` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `bb\_manager`  
--  
  
INSERT INTO `bb\_manager` (`M\_id`, `mName`, `m\_phNo`) VALUES  
(102, 'Jack', '4693959671'),  
(103, 'Peter', '4693959601'),  
(104, 'Mark', '4693959677'),  
(105, 'Jason', '4693957671'),  
(106, 'Steve', '4694959671'),  
(107, 'Jason', '4695959671'),  
(108, 'Stella', '4663959671'),  
(109, 'Monika', '4673959671'),  
(110, 'John', '4693859671'),  
(111, 'Ash', '4693859684'),  
(112, 'FAM', '4673959671'),  
(121, 'HAM', '4673959671'),  
(131, 'SAMON', '4673959671'),  
(141, 'MASB', '4673959671'),  
(151, 'DADB', '4673959671'),  
(161, 'DADBHA', '4673959671'),  
(171, 'DABD', '4673959671'),  
(181, 'DADB', '4673959671'),  
(191, 'DADN', '4673959671'),  
(192, 'RANIKA', '4673959671');  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `bloodspecimen`  
--  
  
CREATE TABLE `bloodspecimen` (  
 `specimen\_number` int(11) NOT NULL,  
 `b\_group` varchar(10) NOT NULL,  
 `status` int(11) DEFAULT NULL,  
 `dfind\_ID` int(11) NOT NULL,  
 `M\_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `bloodspecimen`  
--  
  
INSERT INTO `bloodspecimen` (`specimen\_number`, `b\_group`, `status`, `dfind\_ID`, `M\_id`) VALUES  
(1001, 'B+', 1, 11, 102),  
(1002, 'O+', 1, 12, 102),  
(1003, 'AB+', 1, 11, 102),  
(1004, 'O-', 1, 13, 103),  
(1005, 'A+', 0, 14, 112),  
(1006, 'A-', 1, 13, 104),  
(1007, 'AB-', 1, 15, 104),  
(1008, 'AB-', 0, 11, 105),  
(1009, 'B+', 1, 13, 105),  
(1010, 'O+', 0, 12, 105),  
(1011, 'O+', 1, 13, 103),  
(1012, 'O-', 1, 14, 102),  
(1013, 'B-', 1, 14, 102),  
(1014, 'AB+', 0, 15, 181),  
(1015, 'AB+', 0, 15, 121),  
(1016, 'B+', 0, 15, 131),  
(1017, 'AB+', 0, 15, 141),  
(1018, 'AB+', 0, 15, 151),  
(1019, 'AB+', 0, 15, 161),  
(1020, 'AB+', 0, 15, 104),  
(1021, 'AB+', 0, 15, 104);  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `blood\_donor`  
--  
  
CREATE TABLE `blood\_donor` (  
 `bd\_ID` int(11) NOT NULL,  
 `bd\_name` varchar(255) NOT NULL,  
 `bd\_age` varchar(255) NOT NULL,  
 `bd\_sex` varchar(255) NOT NULL,  
 `bd\_Bgroup` varchar(10) DEFAULT NULL,  
 `bd\_reg\_date` date DEFAULT NULL,  
 `reco\_ID` int(11) NOT NULL,  
 `City\_ID` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
  
  
--  
-- Dumping data for table `blood\_donor`  
--  
  
INSERT INTO `blood\_donor` (`bd\_ID`, `bd\_name`, `bd\_age`, `bd\_sex`, `bd\_Bgroup`, `bd\_reg\_date`, `reco\_ID`, `City\_ID`) VALUES  
(1, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(2, 'Tayan', '20', 'male', 'B+', '2021-04-04', 101112, 1200),  
(3, 'Fahim', '25', 'male', 'A+', '2021-04-29', 101212, 1400),  
(4, 'Fahima', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(5, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(6, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(7, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(8, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(9, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(10, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(12, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(13, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212),  
(14, 'Arnab', '19', 'male', 'A+', '2021-04-29', 101012, 1212);  
  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `city`  
--  
  
CREATE TABLE `city` (  
 `City\_ID` int(11) NOT NULL,  
 `City\_name` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `city`  
--  
  
INSERT INTO `city` (`City\_ID`, `City\_name`) VALUES  
(1200, 'Austin'),  
(1212, 'Badda'),  
(1300, 'Irving'),  
(1400, 'Houston'),  
(1500, 'Richardson'),  
(1600, 'Plano'),  
(1700, 'Frisco'),  
(1800, 'Arlington'),  
(1900, 'San Antonio'),  
(2000, 'Tyler');  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `diseasefinder`  
--  
  
CREATE TABLE `diseasefinder` (  
 `dfind\_ID` int(11) NOT NULL,  
 `dfind\_name` varchar(255) NOT NULL,  
 `dfind\_PhNo` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `diseasefinder`  
--  
  
INSERT INTO `diseasefinder` (`dfind\_ID`, `dfind\_name`, `dfind\_PhNo`) VALUES  
(11, 'Peter', '4693804223'),  
(12, 'Park', '4693804223'),  
(13, 'Jerry', '4693804223'),  
(14, 'Mark', '4693804223'),  
(15, 'Monika', '4693804223'),  
(16, 'Ram', '4693804123'),  
(17, 'Swathi', '4693804223'),  
(18, 'Gautham', '4693804323'),  
(19, 'Ashwin', '4693804423'),  
(20, 'Yash', '4693804523');  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `hospital\_info`  
--  
  
CREATE TABLE `hospital\_info` (  
 `hosp\_ID` int(11) NOT NULL,  
 `hosp\_name` varchar(255) NOT NULL,  
 `hosp\_needed\_Bgrp` varchar(10) NOT NULL,  
 `hosp\_needed\_qnty` int(11) DEFAULT NULL,  
 `City\_ID` int(11) NOT NULL,  
 `M\_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `hospital\_info`  
--  
  
INSERT INTO `hospital\_info` (`hosp\_ID`, `hosp\_name`, `hosp\_needed\_Bgrp`, `hosp\_needed\_qnty`, `City\_ID`, `M\_id`) VALUES  
(51111, 'Square', 'O-', 2, 1200, 191),  
(51112, 'Apollo', 'b-', 1, 1200, 191),  
(51113, 'Medinova', 'a-', 2, 1300, 181),  
(51114, 'Square', 'b+', 2, 1400, 171),  
(51115, 'Square', 'O-', 2, 1500, 161),  
(51116, 'Square', 'O-', 2, 1600, 151),  
(51117, 'Square', 'O-', 2, 1700, 141),  
(51118, 'Square', 'O-', 2, 1800, 191),  
(51119, 'Square', 'O-', 2, 1200, 191),  
(51121, 'Square', 'O-', 2, 1200, 191),  
(51122, 'Square', 'O-', 2, 1200, 191);  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `recipient`  
--  
  
CREATE TABLE `recipient` (  
 `reci\_ID` int(11) NOT NULL,  
 `reci\_name` varchar(255) NOT NULL,  
 `reci\_age` varchar(255) NOT NULL,  
 `reci\_Brgp` varchar(255) NOT NULL,  
 `reci\_Bqnty` float DEFAULT NULL,  
 `reco\_ID` int(11) NOT NULL,  
 `City\_ID` int(11) NOT NULL,  
 `M\_id` int(11) NOT NULL,  
 `reci\_sex` varchar(255) NOT NULL,  
 `reci\_reg\_date` date DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `recipient`  
--  
  
INSERT INTO `recipient` (`reci\_ID`, `reci\_name`, `reci\_age`, `reci\_Brgp`, `reci\_Bqnty`, `reco\_ID`, `City\_ID`, `M\_id`, `reci\_sex`, `reci\_reg\_date`) VALUES  
(1000001, 'fahim', '25', 'B+', 2, 101212, 1700, 192, 'Male', '2021-04-12'),  
(1000002, 'fahimma', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000004, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000005, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000006, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000007, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000008, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000009, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000013, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000023, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000033, 'mahin', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13'),  
(1000042, 'fahimma', '24', 'A+', 1, 101212, 1700, 192, 'Male', '2021-04-13');  
  
-- --------------------------------------------------------  
  
--  
-- Table structure for table `recording\_staff`  
--  
  
CREATE TABLE `recording\_staff` (  
 `reco\_ID` int(11) NOT NULL,  
 `reco\_Name` varchar(255) NOT NULL,  
 `reco\_phNo` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `recording\_staff`  
--  
  
INSERT INTO `recording\_staff` (`reco\_ID`, `reco\_Name`, `reco\_phNo`) VALUES  
(101012, 'Lekha', '4044846553'),  
(101112, 'Mark', '4045856553'),  
(101212, 'Walcot', '4045806553'),  
(101312, 'Henry', '4045806553'),  
(101412, 'Silva', '4045806553'),  
(101512, 'Adrian', '4045806553'),  
(101612, 'Mark', '4045806553'),  
(101712, 'Abdul', '4045816553'),  
(101722, 'Jamal', '4045816573'),  
(101732, 'Kashem', '4045816583'),  
(101812, 'Jerry', '4045826553'),  
(101912, 'Tim', '4045836553');  
  
--  
-- Indexes for dumped tables  
--  
  
--  
-- Indexes for table `bb\_manager`  
--  
ALTER TABLE `bb\_manager`  
 ADD PRIMARY KEY (`M\_id`),  
 ADD UNIQUE KEY `M\_id` (`M\_id`);  
  
--  
-- Indexes for table `bloodspecimen`  
--  
ALTER TABLE `bloodspecimen`  
 ADD PRIMARY KEY (`specimen\_number`,`b\_group`),  
 ADD UNIQUE KEY `specimen\_number` (`specimen\_number`,`dfind\_ID`),  
 ADD UNIQUE KEY `specimen\_number\_2` (`specimen\_number`,`dfind\_ID`,`M\_id`),  
 ADD KEY `dfind\_ID` (`dfind\_ID`),  
 ADD KEY `M\_id` (`M\_id`);  
  
--  
-- Indexes for table `blood\_donor`  
--  
ALTER TABLE `blood\_donor`  
 ADD PRIMARY KEY (`bd\_ID`),  
 ADD UNIQUE KEY `bd\_ID` (`bd\_ID`),  
 ADD UNIQUE KEY `bd\_ID\_2` (`bd\_ID`,`reco\_ID`,`City\_ID`),  
 ADD KEY `reco\_ID` (`reco\_ID`),  
 ADD KEY `City\_ID` (`City\_ID`);  
  
--  
-- Indexes for table `city`  
--  
ALTER TABLE `city`  
 ADD PRIMARY KEY (`City\_ID`),  
 ADD UNIQUE KEY `City\_ID` (`City\_ID`);  
  
--  
-- Indexes for table `diseasefinder`  
--  
ALTER TABLE `diseasefinder`  
 ADD PRIMARY KEY (`dfind\_ID`),  
 ADD UNIQUE KEY `dfind\_ID` (`dfind\_ID`);  
  
--  
-- Indexes for table `hospital\_info`  
--  
ALTER TABLE `hospital\_info`  
 ADD PRIMARY KEY (`hosp\_ID`,`hosp\_needed\_Bgrp`),  
 ADD UNIQUE KEY `hosp\_ID` (`hosp\_ID`,`City\_ID`,`M\_id`),  
 ADD KEY `M\_id` (`M\_id`),  
 ADD KEY `City\_ID` (`City\_ID`);  
  
--  
-- Indexes for table `recipient`  
--  
ALTER TABLE `recipient`  
 ADD PRIMARY KEY (`reci\_ID`),  
 ADD UNIQUE KEY `reci\_ID` (`reci\_ID`,`reco\_ID`),  
 ADD UNIQUE KEY `reci\_ID\_2` (`reci\_ID`,`reco\_ID`,`City\_ID`),  
 ADD UNIQUE KEY `reci\_ID\_3` (`reci\_ID`,`reco\_ID`,`City\_ID`,`M\_id`),  
 ADD KEY `reco\_ID` (`reco\_ID`),  
 ADD KEY `City\_ID` (`City\_ID`),  
 ADD KEY `M\_id` (`M\_id`);  
  
--  
-- Indexes for table `recording\_staff`  
--  
ALTER TABLE `recording\_staff`  
 ADD PRIMARY KEY (`reco\_ID`),  
 ADD UNIQUE KEY `reco\_ID` (`reco\_ID`);  
  
--  
-- Constraints for dumped tables  
--  
  
--  
-- Constraints for table `bloodspecimen`  
--  
ALTER TABLE `bloodspecimen`  
 ADD CONSTRAINT `bloodspecimen\_ibfk\_1` FOREIGN KEY (`dfind\_ID`) REFERENCES `diseasefinder` (`dfind\_ID`),  
 ADD CONSTRAINT `bloodspecimen\_ibfk\_2` FOREIGN KEY (`M\_id`) REFERENCES `bb\_manager` (`M\_id`);  
  
--  
-- Constraints for table `blood\_donor`  
--  
ALTER TABLE `blood\_donor`  
 ADD CONSTRAINT `blood\_donor\_ibfk\_1` FOREIGN KEY (`reco\_ID`) REFERENCES `recording\_staff` (`reco\_ID`),  
 ADD CONSTRAINT `blood\_donor\_ibfk\_2` FOREIGN KEY (`City\_ID`) REFERENCES `city` (`City\_ID`);  
  
--  
-- Constraints for table `hospital\_info`  
--  
ALTER TABLE `hospital\_info`  
 ADD CONSTRAINT `hospital\_info\_ibfk\_1` FOREIGN KEY (`M\_id`) REFERENCES `bb\_manager` (`M\_id`),  
 ADD CONSTRAINT `hospital\_info\_ibfk\_2` FOREIGN KEY (`City\_ID`) REFERENCES `city` (`City\_ID`);  
  
--  
-- Constraints for table `recipient`  
--  
ALTER TABLE `recipient`  
 ADD CONSTRAINT `recipient\_ibfk\_1` FOREIGN KEY (`reco\_ID`) REFERENCES `recording\_staff` (`reco\_ID`),  
 ADD CONSTRAINT `recipient\_ibfk\_2` FOREIGN KEY (`City\_ID`) REFERENCES `city` (`City\_ID`),  
 ADD CONSTRAINT `recipient\_ibfk\_3` FOREIGN KEY (`M\_id`) REFERENCES `bb\_manager` (`M\_id`);  
COMMIT;  
  
/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;  
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;  
/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

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 bank 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.