MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Skill Based Mini Project Report

on

BLOOD BANK MANAGEMENT SYSTEM

Submitted By:

PRATIBHA DHURVE

0901CS201088.

Faculty Mentor:
Ms. Jaimala Jha
Assistant Professor
Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JAN-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Pratibha Dhurve** (0901CS201088) has submitted the project report titled **blood bank Management System** under the mentorship of Ms. **Jaimala Jha** , in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Ms. Jaimala jha
Faculty Mentor
Assistant professor
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment

of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship , Ms. **Jaimala jha ,Assistant professor, Computer Science and Enginnering.**

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Pratibha Dhurve
0901cs201088
2nd year,4 sem
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the FlexibleCurriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of , **Ms. Jaimala Jha , Assistant professor, Computer Science and Engineering ,**for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Pratibha Dhurve
0901cs201088
2nd year,4 sem
Computer Science and Engineering

ABSTRACT

This project aims to develop a Blood Bank Management System. A Blood Bank Management System can be used in any clinic, hospital, labs or any emergency situation which requires blood units for survival. Our system can be used to find required type of blood in emergency situations from either blood bank or even blood donors.

Current system uses a grapevine communication for finding blood in cases of emergency, may It be by a donor or blood bank. The intentions of proposing such a system is to abolish the panic caused during an emergency due to unavailability of blood

TABLE OF CONTENTS

LIST OF FIGURES

Figure number	Figure caption	Page No.

LIST OF TABLES

Table Number **Table Title**
PageNo.

INTRODUCTION

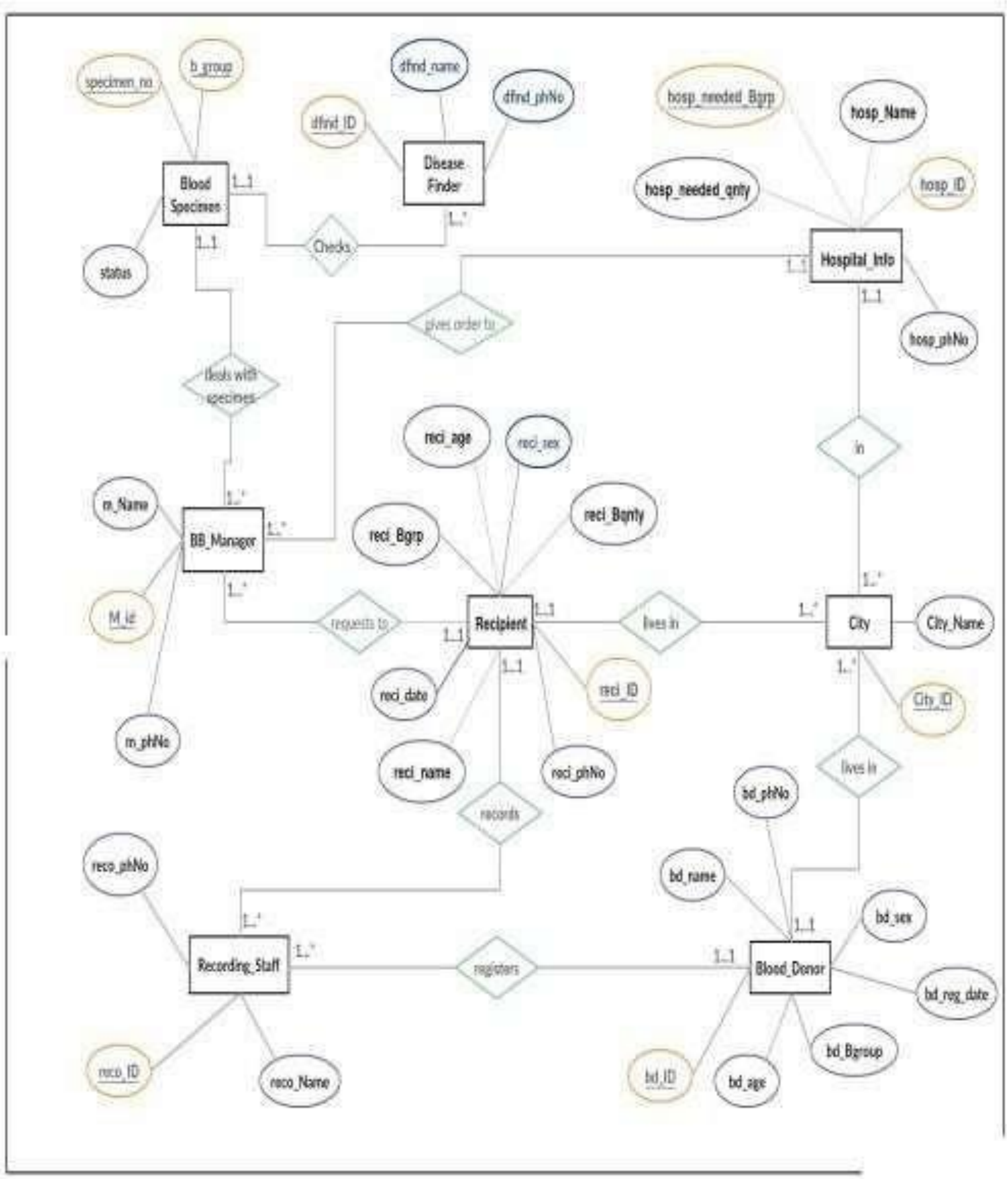
Blood banks collect, store and provide collected blood to the patients who are in need of blood. The people who donate blood are called ‘donors’.
The banks then group the blood which they receive according to the blood groups. They also make sure that the blood is not contaminated. The main mission of 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 healthcare 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 a dministrative 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 USING CREATLY AND RELATION BETWEEN THE ENTITIES



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 donatesblood, 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 keyfor 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 bankmanager is the person who takes care of the avaible blood samples in theblood 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 numberof 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. **Blood Specimen :** (Attributes – specimen_number, b_group , status) In data base, under BloodSpecimen entity we will store the information ofblood samples which are available in the blood bank. In this entity specimen_number and b_group together will be primary key along withstatus 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 thedoctor who checks the blood for any kind of contaminations. To store thatinformation we have unique identification number (dfind_ID) as primary
7. **Hospital_Info :** (Attributes – hosp_ID, hosp_name, hosp_needed_Bgrp, h osp_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 hospitalname 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. Aunique indentification number (City_ID) will be used as primary key to indefiy the information about the city. Along with ID city names will alsobe stored under this entity.

RELETIONSHIP 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 inone city.

2. **City and Blood_Donor:** Relationship = “lives in” Type of relation = to many Explanation = In a city, many donor can live. One donor will belong toone 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 = 1to many Explanation = One recording staff can register many donors. One donor will register with one recording officer.

5. **Recording_Staff and Recipient:** Relationship = “records” Typeof relation =1 to many Explanation = One recording staff can record many recipients. One recipient will be recorded by one recording officer.

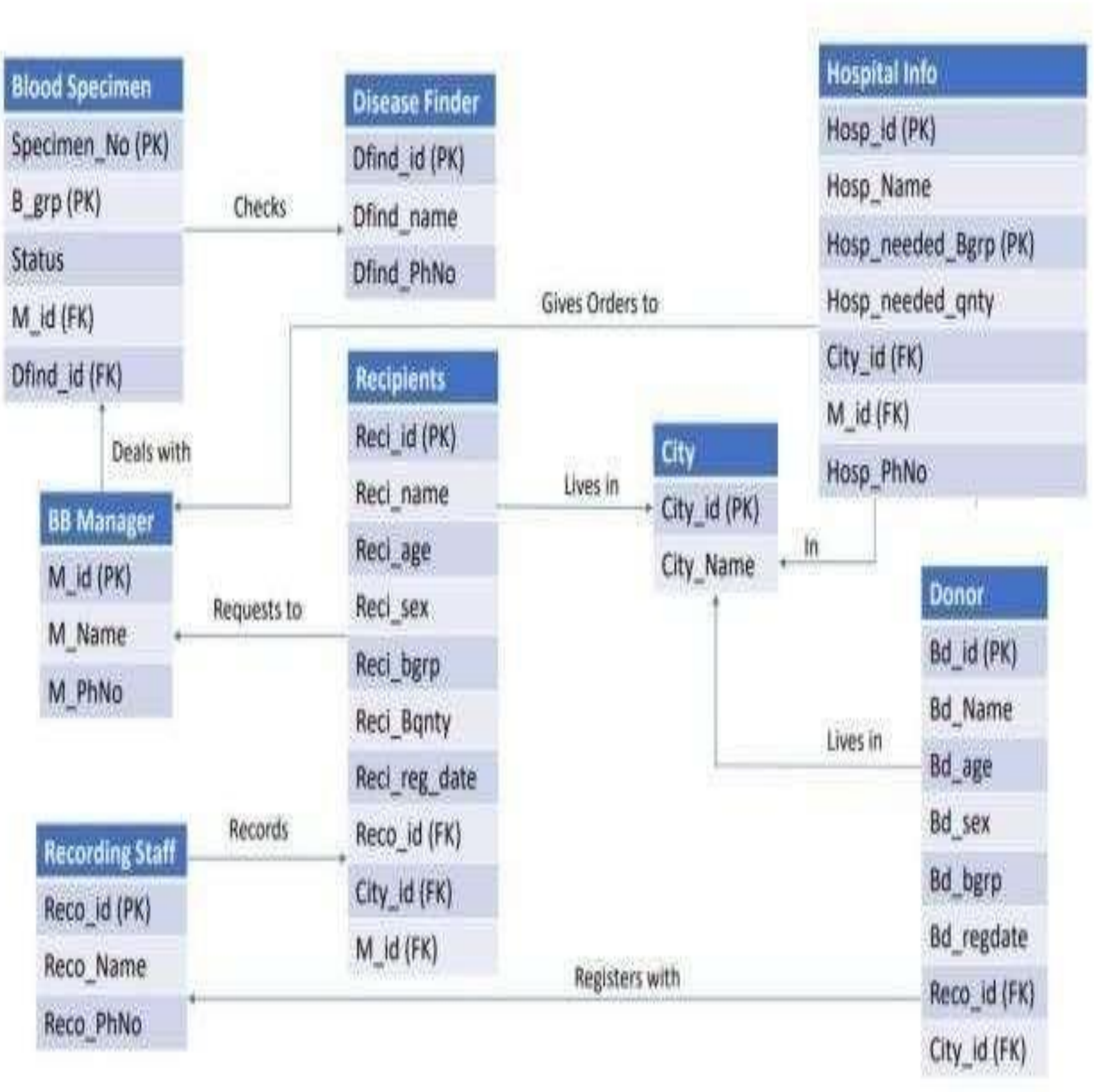
6. **Hospital_Info and BB_Manager:** Relationship = “gives order to” Type of relation = 1 to many P a g e | 8 Explanation = One Blood bank manager can handle and process requests from many hospitals. One hospital will place request to on blood bank manager.

7. **BB_Manager and Blood Specimen**: Relationship = “deales with specimen”Type of relation = 1 to many Explanation = One Blood bank manager can

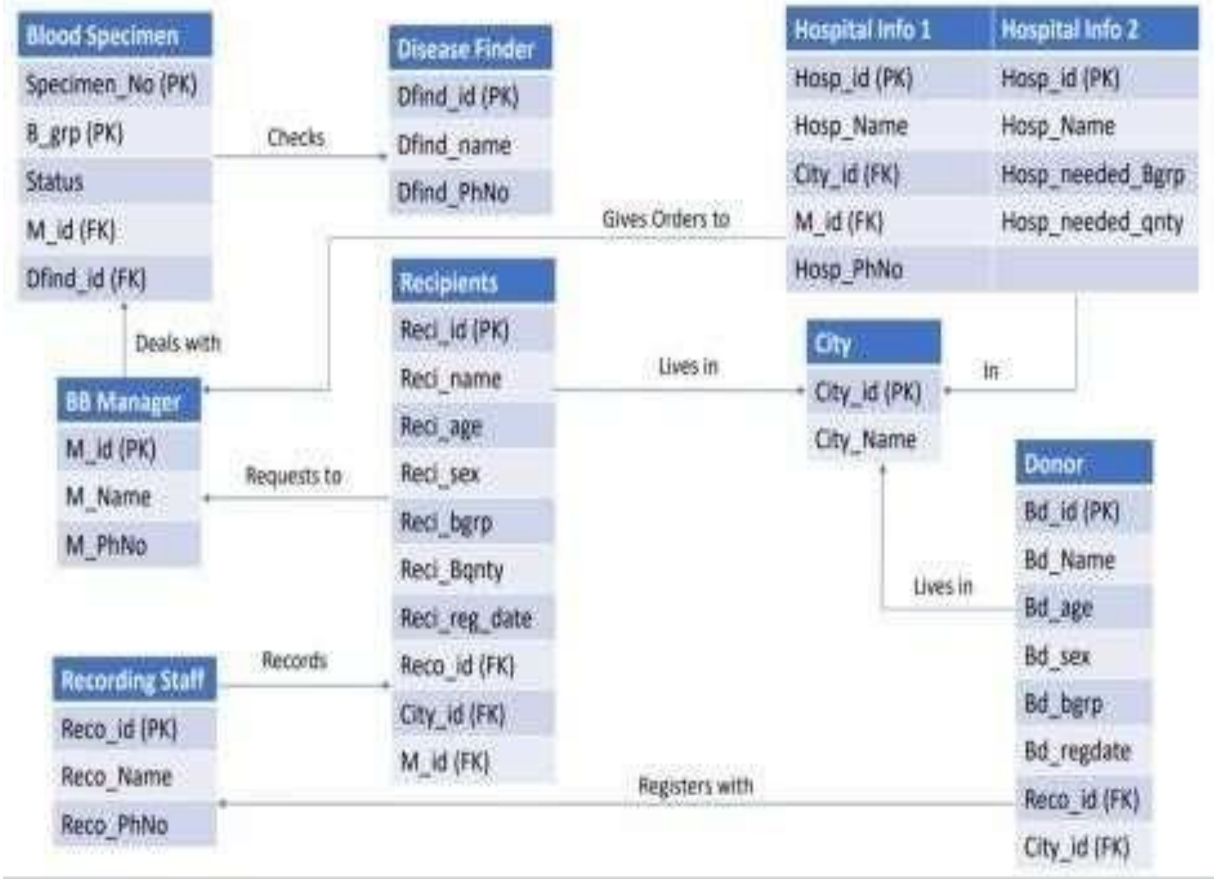
8. Recipient and BB_Manager: Relationship = "requests to" Type of relation = 1 to many Explanation = One recipient can request blood to one manager andone manager can handle requests from many recipients.

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.manage many blood specimen and one specimen will be managed by onemanager.

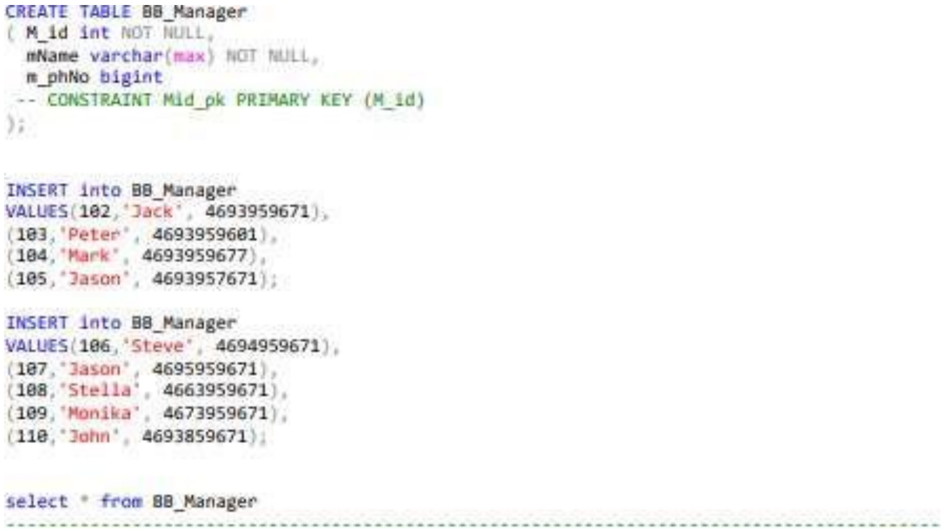
ER DIAGRAM WITH TABLES



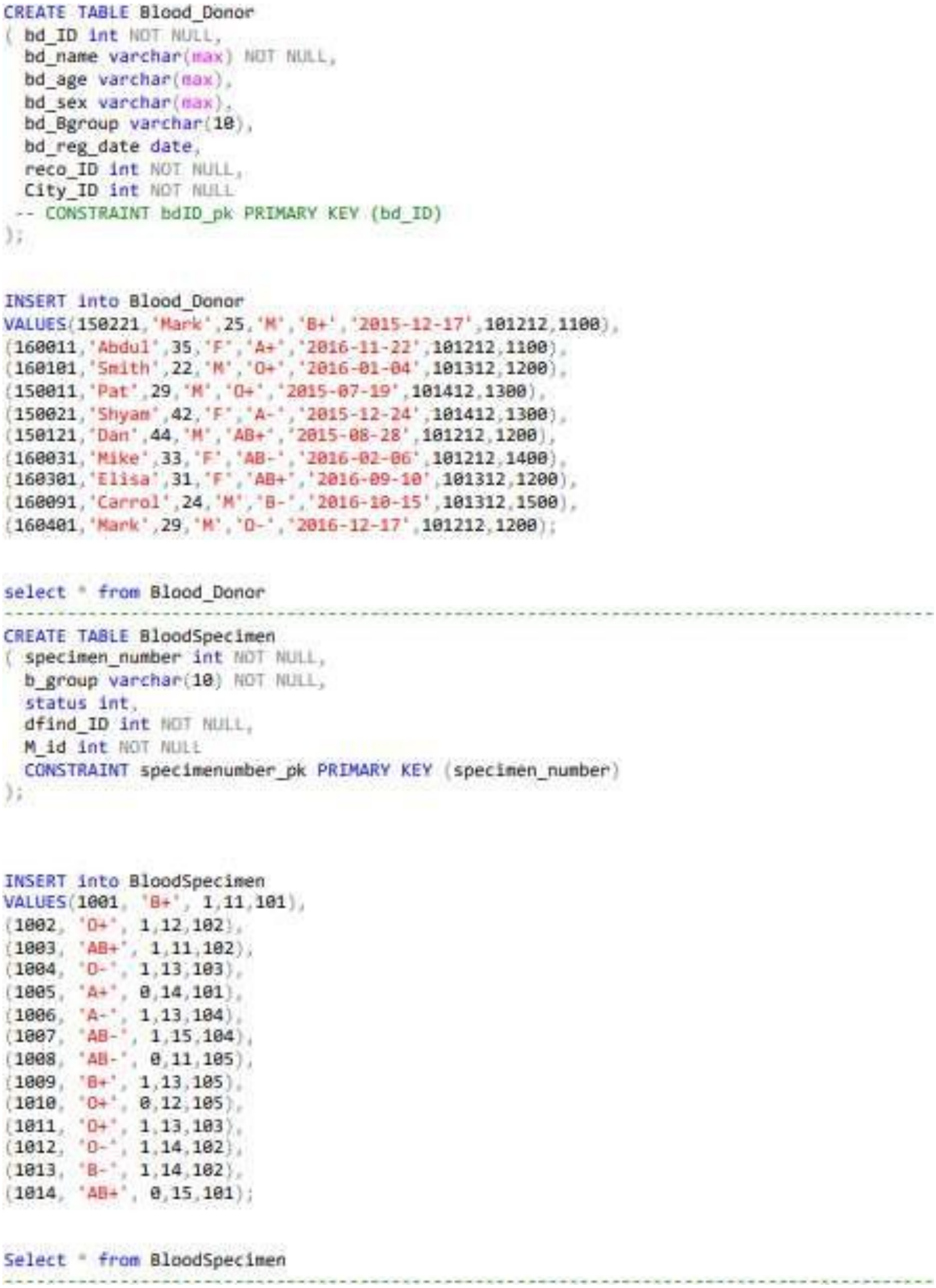
ER DIAGRAM AFTER NORMALIZATION



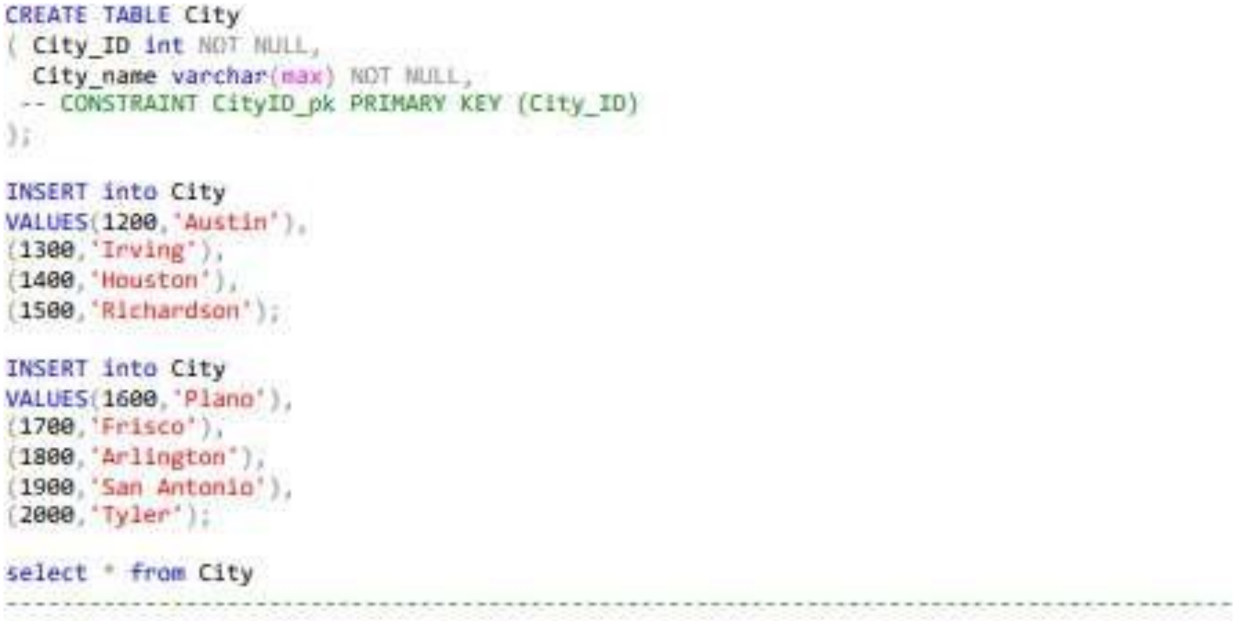
SQL IMPLEMENTATION



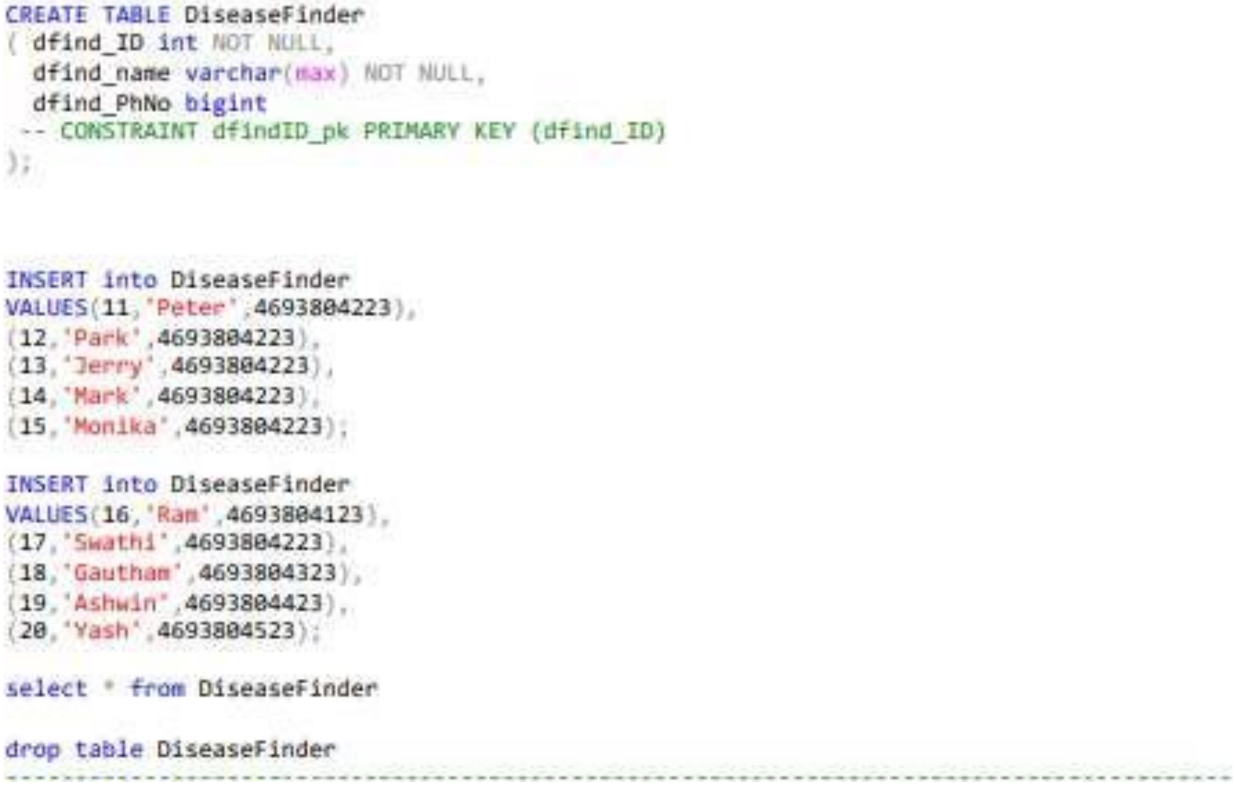
```
CREATE TABLE BB_Manager
( M_id int NOT NULL,
  mName varchar(max) NOT NULL,
  m_phNo bigint
 -- CONSTRAINT Mid_pk PRIMARY KEY (M_id)
);


INSERT into BB_Manager
VALUES(102,'Jack', 4693959671),
(103,'Peter', 4693959601),
(104,'Mark', 4693959677),
(105,'Jason', 4693957671);

INSERT into BB_Manager
VALUES(106,'Steve', 4694959671),
(107,'Jason', 4695959671),
(108,'Stella', 4663959671),
(109,'Monika', 4673959671),
(110,'John', 4693859671);


select * from BB_Manager
------------------------------------------------------------------------------------------
```



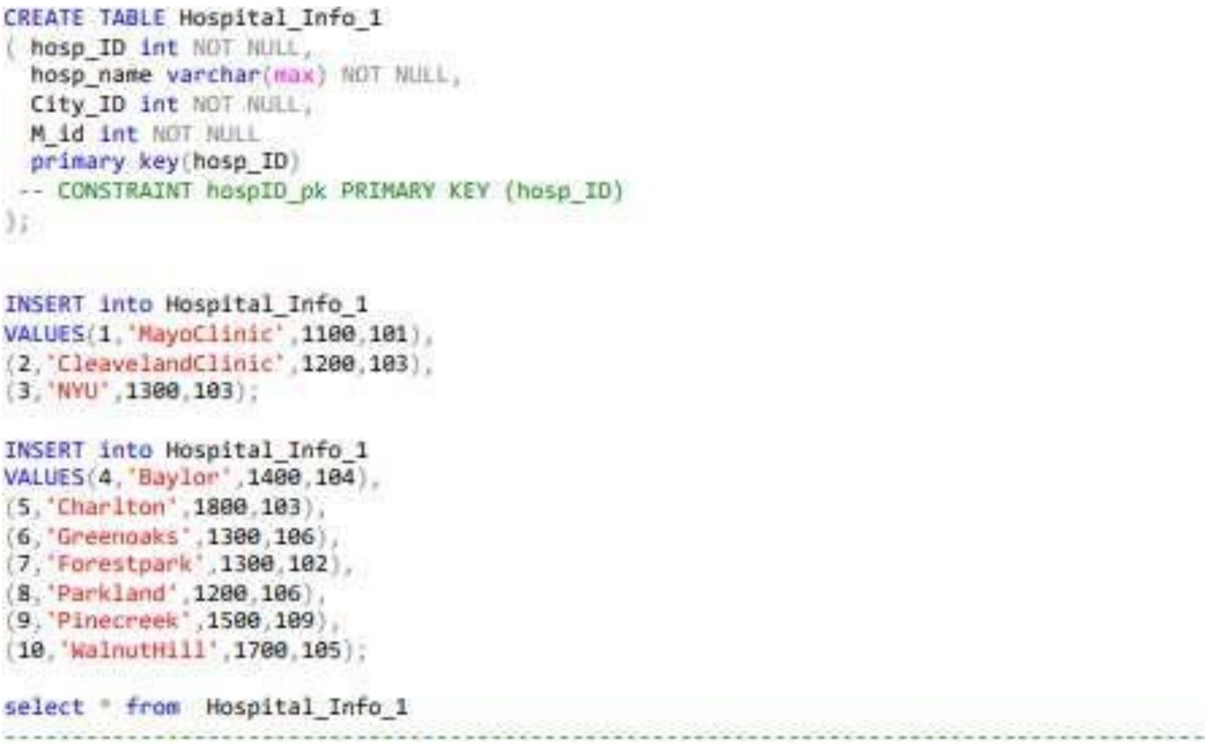
```
CREATE TABLE Blood_Donor
( bd_ID int NOT NULL,
  bd_name varchar(max) NOT NULL,
  bd_age varchar(max),
  bd_sex varchar(max),
  bd_Bgroup varchar(10),
  bd_reg_date date,
  reco_ID int NOT NULL,
  City_ID int NOT NULL
 -- CONSTRAINT bdID_pk PRIMARY KEY (bd_ID)
);


INSERT into Blood_Donor
VALUES(150221,'Mark',25,'M','B+','2015-12-17',101212,1100),
(160011,'Abdul',35,'F','A+','2016-11-22',101212,1100),
(160101,'Smith',22,'M','O+','2016-01-04',101312,1200),
(150011,'Pat',29,'M','O+','2015-07-19',101412,1300),
(150021,'Shyam',42,'F','A-','2015-12-24',101412,1300),
(150121,'Dan',44,'M','AB+','2015-08-28',101212,1200),
(160031,'Mike',33,'F','AB-','2016-02-06',101212,1400),
(160301,'Elisa',31,'F','AB+','2016-09-10',101312,1200),
(160091,'Carrol',24,'M','B-','2016-10-15',101312,1500),
(160401,'Mark',29,'M','O-','2016-12-17',101212,1200);


select * from Blood_Donor
------------------------------------------------------------------------------------------
CREATE TABLE BloodSpecimen
( specimen_number int NOT NULL,
  b_group varchar(10) NOT NULL,
  status int,
  dfind_ID int NOT NULL,
  M_id int NOT NULL
  CONSTRAINT specimenumber_pk PRIMARY KEY (specimen_number)
);



INSERT into BloodSpecimen
VALUES(1001, 'B+', 1,11,101),
(1002, 'O+', 1,12,102),
(1003, 'AB+', 1,11,102),
(1004, 'O-', 1,13,103),
(1005, 'A+', 0,14,101),
(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+', 0,15,101);


Select * from BloodSpecimen
------------------------------------------------------------------------------------------
```



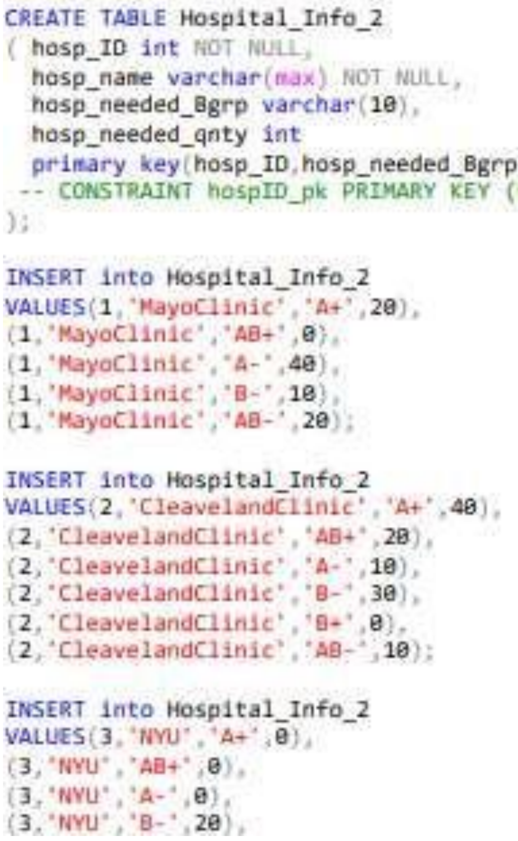
```
CREATE TABLE City
( City_ID int NOT NULL,
  City_name varchar(max) NOT NULL,
 -- CONSTRAINT CityID_pk PRIMARY KEY (City_ID)
);

INSERT into City
VALUES(1200,'Austin'),
(1300,'Irving'),
(1400,'Houston'),
(1500,'Richardson');

INSERT into City
VALUES(1600,'Plano'),
(1700,'Frisco'),
(1800,'Arlington'),
(1900,'San Antonio'),
(2000,'Tyler');

select * from City
------------------------------------------------------------------------------------------
```



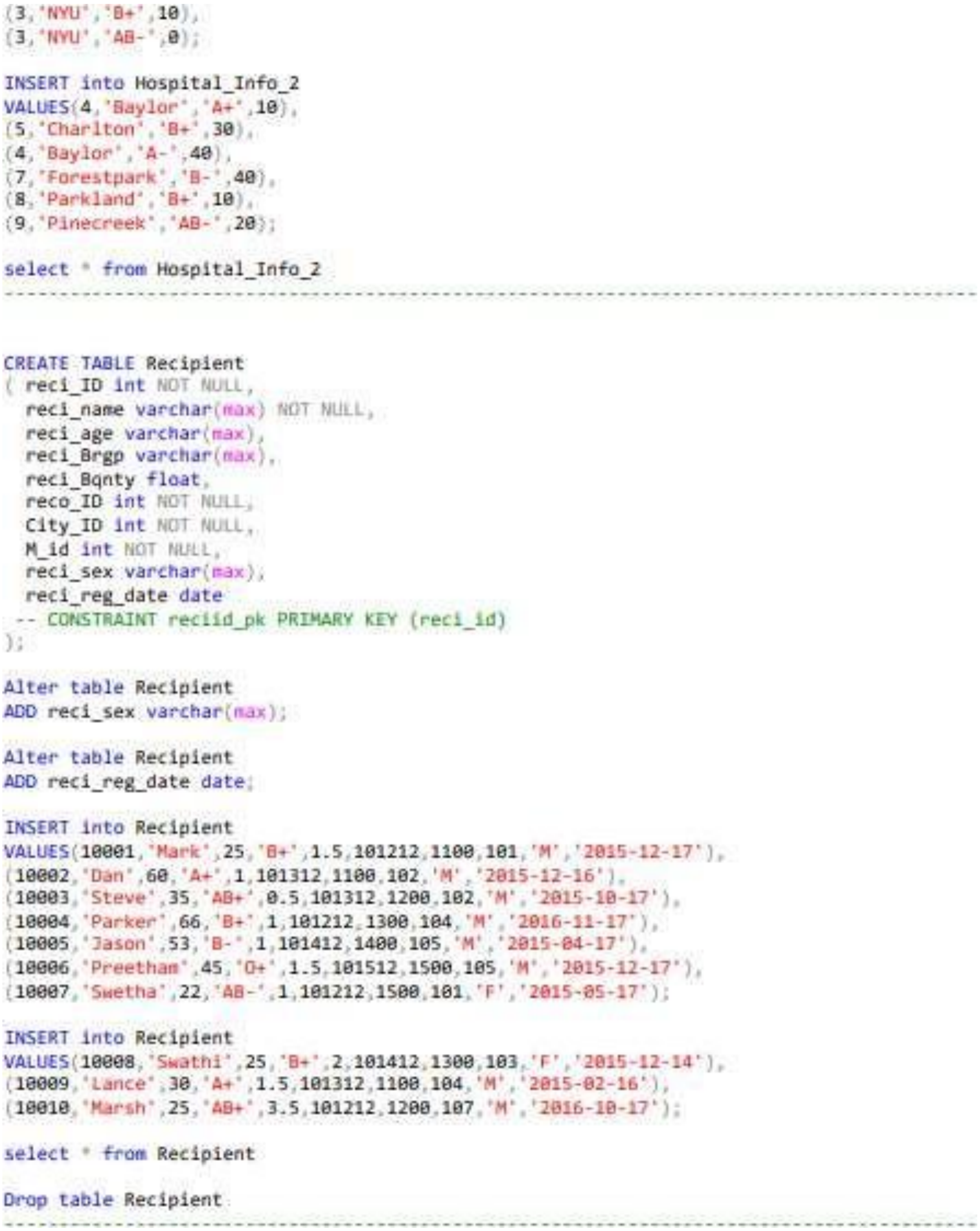
```
CREATE TABLE DiseaseFinder
( dfind_ID int NOT NULL,
  dfind_name varchar(max) NOT NULL,
  dfind_PhNo bigint
 -- CONSTRAINT dfindID_pk PRIMARY KEY (dfind_ID)
);


INSERT into DiseaseFinder
VALUES(11,'Peter',4693804223),
(12,'Park',4693804223),
(13,'Jerry',4693804223),
(14,'Mark',4693804223),
(15,'Monika',4693804223);

INSERT into DiseaseFinder
VALUES(16,'Ram',4693804123),
(17,'Swathi',4693804223),
(18,'Gautham',4693804323),
(19,'Ashwin',4693804423),
(20,'Yash',4693804523);

select * from DiseaseFinder

drop table DiseaseFinder
------------------------------------------------------------------------------------------
```



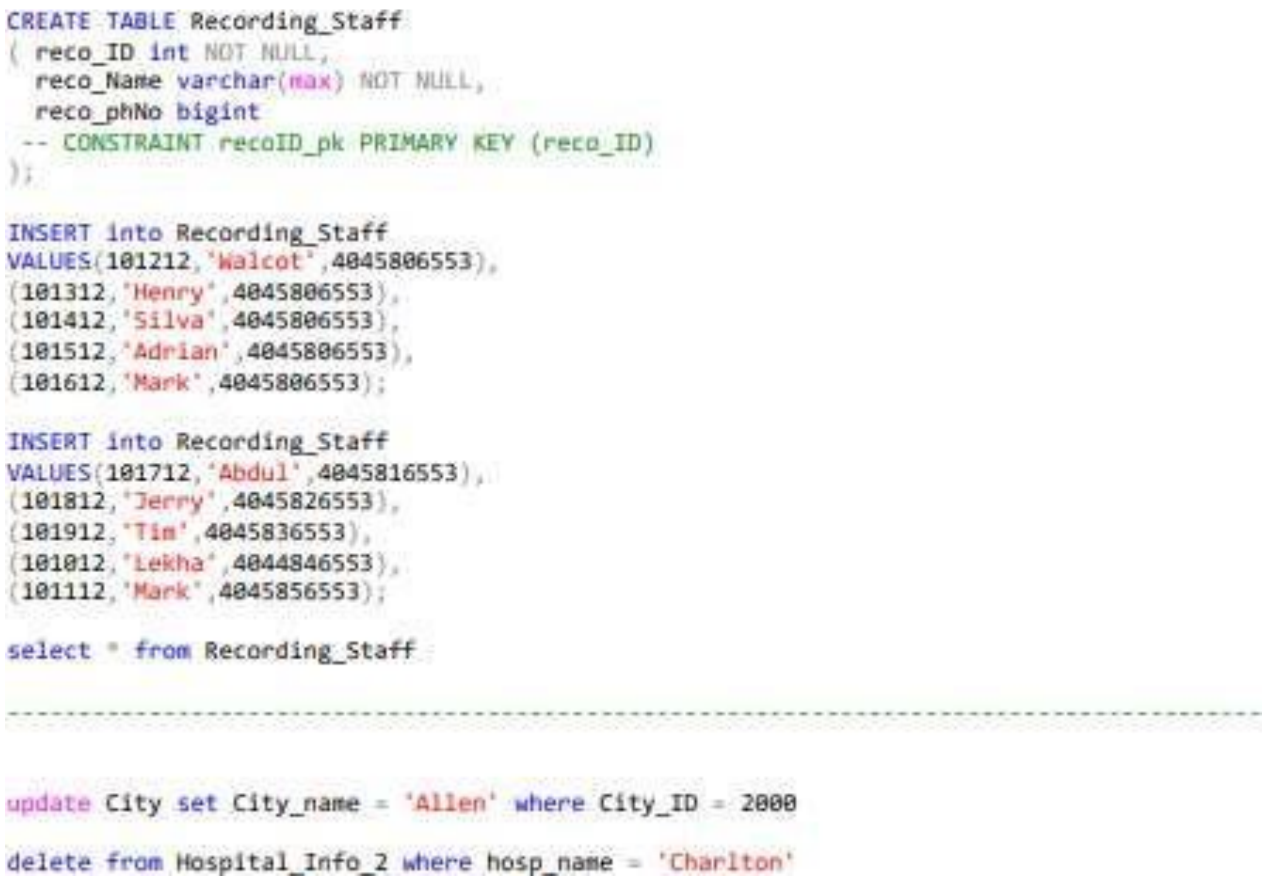
```
CREATE TABLE Hospital_Info_1
( hosp_ID int NOT NULL,
  hosp_name varchar(max) NOT NULL,
  City_ID int NOT NULL,
  M_id int NOT NULL
  primary key(hosp_ID)
 -- CONSTRAINT hospID_pk PRIMARY KEY (hosp_ID)
);


INSERT into Hospital_Info_1
VALUES(1,'MayoClinic',1100,101),
(2,'CleavelandClinic',1200,103),
(3,'NYU',1300,103);

INSERT into Hospital_Info_1
VALUES(4,'Baylor',1400,104),
(5,'Charlton',1800,103),
(6,'Greenoaks',1300,106),
(7,'Forestpark',1300,102),
(8,'Parkland',1200,106),
(9,'Pinecreek',1500,109),
(10,'WalnutHill',1700,105);

select * from  Hospital_Info_1
--------------------------------------------------------------------------------------------
```



```
CREATE TABLE Hospital_Info_2
( hosp_ID int NOT NULL,
  hosp_name varchar(max) NOT NULL,
  hosp_needed_Bgrp varchar(10),
  hosp_needed_qnty int
  primary key(hosp_ID,hosp_needed_Bgrp)
 -- CONSTRAINT hospID_pk PRIMARY KEY (hosp_ID)
);

INSERT into Hospital_Info_2
VALUES(1,'MayoClinic','A+',20),
(1,'MayoClinic','AB+',0),
(1,'MayoClinic','A-',40),
(1,'MayoClinic','B-',10),
(1,'MayoClinic','AB-',20);

INSERT into Hospital_Info_2
VALUES(2,'CleavelandClinic','A+',40),
(2,'CleavelandClinic','AB+',20),
(2,'CleavelandClinic','A-',10),
(2,'CleavelandClinic','B-',30),
(2,'CleavelandClinic','B+',0),
(2,'CleavelandClinic','AB-',10);

INSERT into Hospital_Info_2
VALUES(3,'NYU','A+',0),
(3,'NYU','AB+',0),
(3,'NYU','A-',0),
(3,'NYU','B-',20),
```



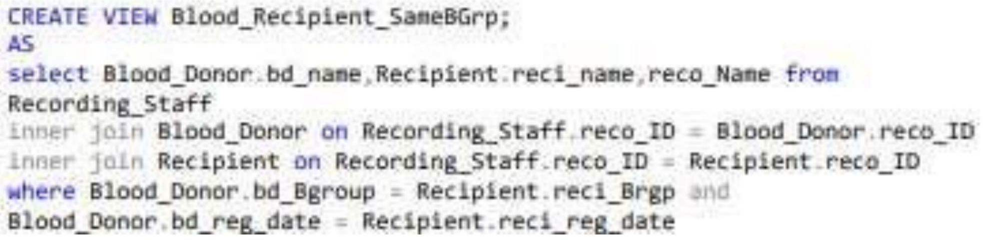
```sql
(3,'NYU','B+',10),
(3,'NYU','AB-',0);

INSERT into Hospital_Info_2
VALUES(4,'Baylor','A+',10),
(5,'Charlton','B+',30),
(4,'Baylor','A-',40),
(7,'Forestpark','B-',40),
(8,'Parkland','B+',10),
(9,'Pinecreek','AB-',20);

select * from Hospital_Info_2
------------------------------------------------------------------------------------------


CREATE TABLE Recipient
( reci_ID int NOT NULL,
  reci_name varchar(max) NOT NULL,
  reci_age varchar(max),
  reci_Brgp varchar(max),
  reci_Bqnty float,
  reco_ID int NOT NULL,
  City_ID int NOT NULL,
  M_id int NOT NULL,
  reci_sex varchar(max),
  reci_reg_date date
 -- CONSTRAINT reciid_pk PRIMARY KEY (reci_id)
);

Alter table Recipient
ADD reci_sex varchar(max);

Alter table Recipient
ADD reci_reg_date date;

INSERT into Recipient
VALUES(10001,'Mark',25,'B+',1.5,101212,1100,101,'M','2015-12-17'),
(10002,'Dan',60,'A+',1,101312,1100,102,'M','2015-12-16'),
(10003,'Steve',35,'AB+',0.5,101312,1200,102,'M','2015-10-17'),
(10004,'Parker',66,'B+',1,101212,1300,104,'M','2016-11-17'),
(10005,'Jason',53,'B-',1,101412,1400,105,'M','2015-04-17'),
(10006,'Preetham',45,'O+',1.5,101512,1500,105,'M','2015-12-17'),
(10007,'Swetha',22,'AB-',1,101212,1500,101,'F','2015-05-17');

INSERT into Recipient
VALUES(10008,'Swathi',25,'B+',2,101412,1300,103,'F','2015-12-14'),
(10009,'Lance',30,'A+',1.5,101312,1100,104,'M','2015-02-16'),
(10010,'Marsh',25,'AB+',3.5,101212,1200,107,'M','2016-10-17');

select * from Recipient

Drop table Recipient
------------------------------------------------------------------------------------------
```



```
CREATE TABLE Recording_Staff
( reco_ID int NOT NULL,
  reco_Name varchar(max) NOT NULL,
  reco_phNo bigint
 -- CONSTRAINT recoID_pk PRIMARY KEY (reco_ID)
);

INSERT into Recording_Staff
VALUES(101212,'Walcot',4045806553),
(101312,'Henry',4045806553),
(101412,'Silva',4045806553),
(101512,'Adrian',4045806553),
(101612,'Mark',4045806553);

INSERT into Recording_Staff
VALUES(101712,'Abdul',4045816553),
(101812,'Jerry',4045826553),
(101912,'Tim',4045836553),
(101012,'Lekha',4044846553),
(101112,'Mark',4045856553);

select * from Recording_Staff

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


update City set City_name = 'Allen' where City_ID = 2000

delete from Hospital_Info_2 where hosp_name = 'Charlton'
```

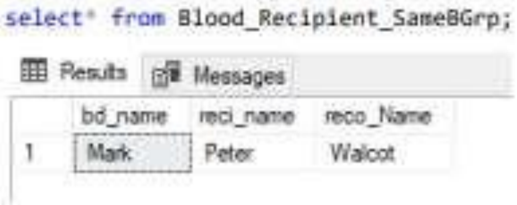
SAMPLE SQL QUERIES

1. **Create a View of recipients and donors names having the same blood group registered on the same date.**



```
CREATE VIEW Blood_Recipient_SameBGrp;
AS
select Blood_Donor.bd_name,Recipient.reci_name,reco_Name from
Recording_Staff
inner join Blood_Donor on Recording_Staff.reco_ID = Blood_Donor.reco_ID
inner join Recipient on Recording_Staff.reco_ID = Recipient.reco_ID
where Blood_Donor.bd_Bgroup = Recipient.reci_Brgp and
Blood_Donor.bd_reg_date = Recipient.reci_reg_date
```

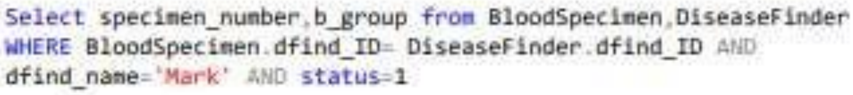
Output:



```
select* from Blood_Recipient_SameBGrp;
```

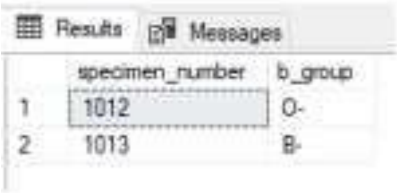
	bd_name	reci_name	reco_Name
1	Mark	Peter	Walcot

2. **Show the blood specimen verified by disease finder Mark which are pure (status=1).**



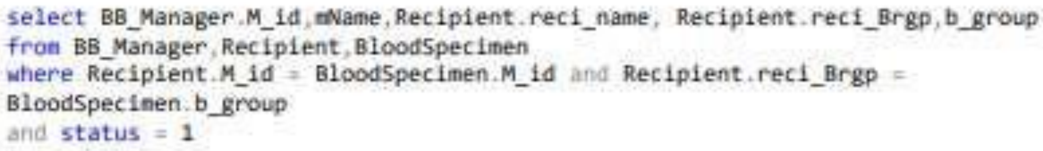
```
Select specimen_number,b_group from BloodSpecimen,DiseaseFinder
WHERE BloodSpecimen.dfind_ID= DiseaseFinder.dfind_ID AND
dfind_name='Mark' AND status=1
```

Output:



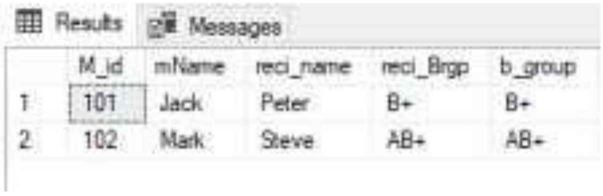
	specimen_number	b_group
1	1012	O-
2	1013	B-

3. **Show the pure blood specimen handled by BB_Manager who also handles a recipient needing the same blood group along with the details of the BB_Manager and Recipient.**



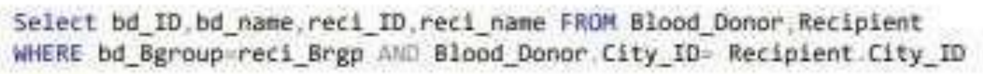
```
select BB_Manager.M_id,mName,Recipient.reci_name, Recipient.reci_Brgp,b_group
from BB_Manager,Recipient,BloodSpecimen
where Recipient.M_id = BloodSpecimen.M_id and Recipient.reci_Brgp =
BloodSpecimen.b_group
and status = 1
```

Output:



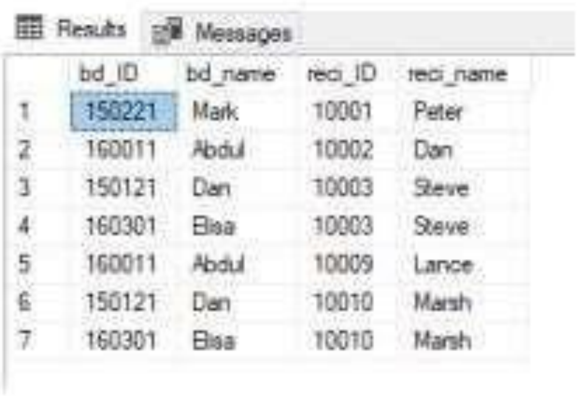
	M_id	mName	reci_name	reci_Brgp	b_group
1	101	Jack	Peter	B+	B+
2	102	Mark	Steve	AB+	AB+

4. **Show the donors having the same blood groups required by the recipient staying in the same city along with recipient details.**



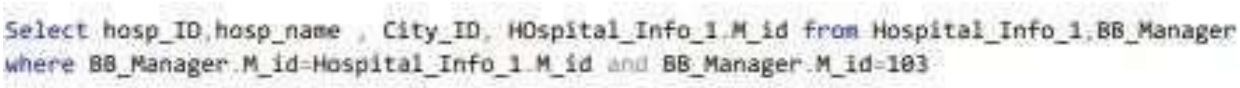
```
Select bd_ID,bd_name,reci_ID,reci_name FROM Blood_Donor,Recipient
WHERE bd_Bgroup=reci_Brgp AND Blood_Donor.City_ID= Recipient.City_ID
```

Output:

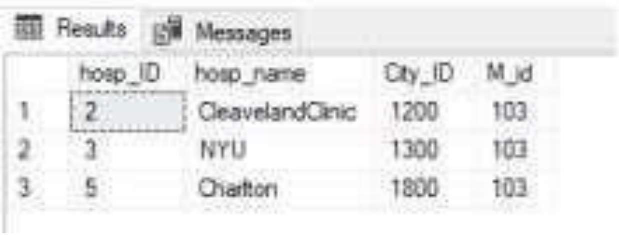


	bd_ID	bd_name	reci_ID	reci_name
1	150221	Mark	10001	Peter
2	160011	Abdul	10002	Dan
3	150121	Dan	10003	Steve
4	160301	Elisa	10003	Steve
5	160011	Abdul	10009	Lance
6	150121	Dan	10010	Marsh
7	160301	Elisa	10010	Marsh

5. **Display the information of Hospital_Info_1 handled by BB_Manager whose ID is 103:**



```
Select hosp_ID,hosp_name , City_ID, HOspital_Info_1.M_id from Hospital_Info_1,BB_Manager
where BB_Manager.M_id=Hospital_Info_1.M_id and BB_Manager.M_id=103
```



	hosp_ID	hosp_name	City_ID	M_id
1	2	CleavelandClinic	1200	103
2	3	NYU	1300	103
3	5	Charlton	1800	103

CONCLUSION

Technology is introducing new innovations day by day, thus reducing the time required to do things. The proposed system can be used to reduce the time required to deliver required blood to the needy in cases of emergency. The Android application can be used by the people interested in donating their blood by locating their nearest blood bank. The web application provides a way of communication and synchronization between the hospitals and the blood banks. It also provides them with the facility of communicating with the nearby donors in emergency. The database is a vital aspect of the system. The database of the hospitals and the blood banks must be checked for consistency on regular basis for smooth working of the system. The proposed system uses Google Maps which provides the user with an efficient way of locating the nearby donors/blood banks. The Android application is developed using Android Studio which is an open source software, while the web application for the hospitals and the blood banks is also developed using open source tools, hence the system developed is quite feasible.