Blood Bank Management System

Submitted By:

MANISHA

2019UMC4557

MATHEMATICS AND COMPUTING

OBJECTIVE

The purpose of the blood bank management system is to simplify and automate the process of searching for blood in case of emergency especially COVID-19 and maintain the records of blood donors, recipients, hospitals which requested the blood and blood sample details.

During COVID-19, the requirement of a well organized and efficient database system is indispensable.

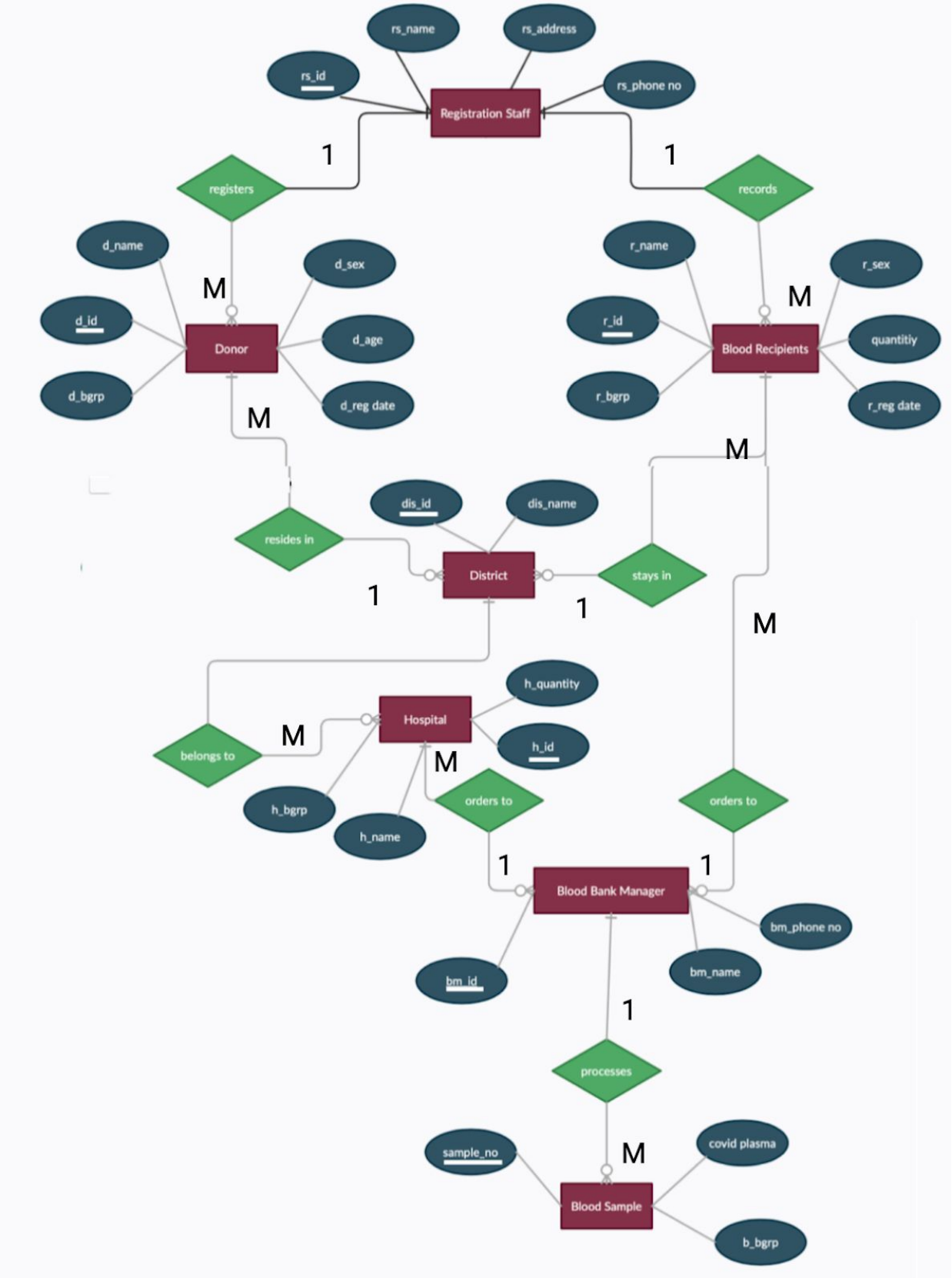
Treating patients suffering from coronavirus infection with transfusion of blood plasma from those who already recovered from the disease have been found to be safe and beneficial, a study suggested.The treatment, also called convalescent plasma, found that people who recover from infection have virus-fighting antibodies floating in their blood, and transfusions of their plasma -- the clear liquid after blood cells are removed -- may give recipients' immune systems an assist in fighting off the virus, The Washington Post reported.

This database would contain entities and relationships between the entities which would make it efficient,easy to access,modify,retrievable and consistent. It would further improve data integrity and facilitate addition of new data.

Many donors donate blood,each of different blood groups/type. A donor may donate blood more than once and he is identified by a donor id(d_id),name, sex, age, blood type and registration date. Before each donor donates his blood , he is required to register himself as a donor with the volunteer/registration staff who works at the Blood Bank.

The registration staff is identified by employee id, name , address and phone number. The Blood Banks receive orders for blood from many hospitals for emergency purposes and other surgical requirements and each blood bank issues the same blood type.The Blood Bank is managed by the blood bank manager who is identified by employee id , name and phone number. He is responsible for the proper management of the blood bank.

ER Diagram


rs_name
rs_address
rs_id
rs_phone no
Registration Staff
1
1
registers
records
d_name
d_sex
M
M
d_id
Donor
d_age
r_name
r_sex
r_id
Blood Recipients
quantitiy
d_bgrp
d_reg date
r_bgrp
r_reg date
M
M
dis_id
dis_name
resides in
District
stays in
1
1
M
h_quantity
Hospital
belongs to
M
h_id
M
h_bgrp
orders to
orders to
h_name
1
1
Blood Bank Manager
bm_phone no
bm_id
bm_name
1
processes
sample_no
M
covid plasma
Blood Sample
b_bgrp

The above ER Diagram represents 7 entity sets.

They are—

a) Donor: (Attributes- d_Name, d_Id, d_sex, age, d_reg date, d_bgrp).

b) District: (Attributes- dis_id, dis_name).

c) Registration_Staff: (Attributes- rs_id, rs_name, rs_address,rs_phone no).

d) Blood_Recipient: (Attributes- r_Id, r_sex, r_reg date, r_name, quantity, r_bgrp).

e) Blood_Sample: (Attributes- b_bgrp, sample_no, covid plasma).

f) Blood Bank Manager: (Attributes- bm_id, bm_name, bm_phone no).

g) Hospital: (Attributes- h_Id, h_name, h_bgrp, h_quantity).

Entities

Donor- Who donates blood. When a donor will donate, an id(a serial number will be given for a specific identification (primary key)); age, sex, name, registration date (d_regdate) and blood group will be stored in the database under entity Donor.

District- Every district has a unique Id (primary key).

Registration Staff- Registration staff will register the information of donors and the recipients. Each member will have a unique id(Primary key). The name, address and phone no of the member will also be stored in the database.

Blood Bank Manager- They will take orders from the hospitals and fulfill their needed requirements of blood samples. The id(primary key), name and phone no will be stored under this entity.

Blood Sample- The quantities of blood that the Blood_bank has. Their group, sample_no(primary key), covid plasma(Yes/No) will be stored.

Hospital- Hospitals of each district, where blood samples are needed, also included in the database.

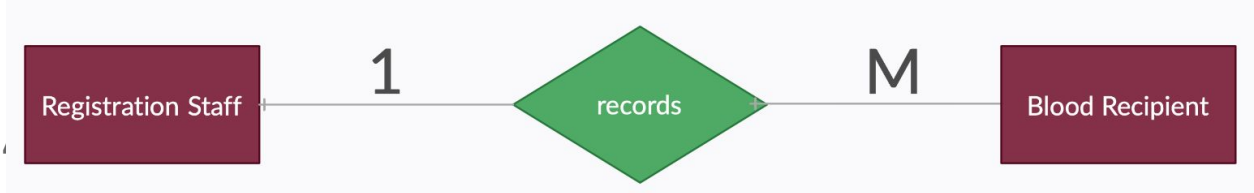
Blood Recipient- Who needs blood. A recipient's id (primary key), name, sex, blood group and quantity of blood required will be stored in a database.

Cardinality:

Donor & District- (Relationship- (resides in), many to 1). One donor stays in one district. In one district, many donors can stay.

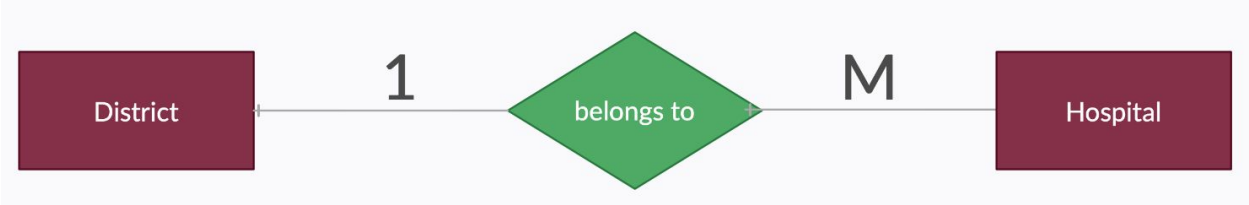
Donor & Registration Staff - (Relationship-(registers), many to 1). A staff can ensure many donors' registration. One donor can get registered by one staff.

Registration Staff & Blood Recipient- (Relationship-(records), 1 to many). A staff can ensure many blood_recipients' registration. One blood_recipient can get registered by one staff member.



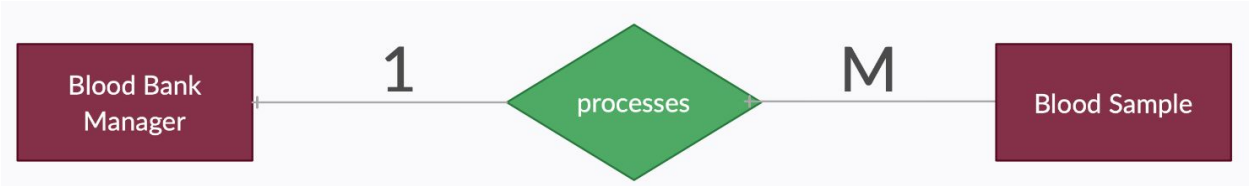
District & Blood Recipient - (Relationship-(stays in), 1 to many). One recipient stays in one district. In one district, many recipients can stay.

District & Hospital- (Relationship-(belongs to), 1 to many).In a district, there are many hospitals. One hospital belongs to one district.



Blood Bank Manager & Hospital- (Relationship-(orders to), 1 to many). A blood processing manager can get orders from many hospitals. One hospital submits an order to a blood processing manager.

Blood Bank Manager & Blood Sample-(Relationship-(processes), 1 to many). A manager can process many samples of blood. One blood sample can be processed by one blood processing manager.



Blood Bank Manager & Blood Recipient-(Relationship-(orders to), 1 to many). The samples of blood are given according to the necessity of the recipients, processed by the manager. A manager can process many samples of blood that are requested by the recipients. But one recipient can request only one blood processing manager.

Relational Schemas

Donor

Attribute Name	Description	Type
d_name	Name of the donor	varchar
d_id	Id of the donor	int
d_sex	Sex of the donor	char
d_age	Age of the donor	int
d_reg date	Registration date of the donor	date
d_bgrp	Donor's blood group	varchar
rs_id (fk)	Id of the registration staff	int
dis_id(fk)	District id	int

The relationship with Registration_staff and Donor is 1 to many. Hence, the primary key of Registration_staff is used as a foreign key in Donor.

The relationship with District and Donor is 1 to many. Hence, the primary key of District is used as a foreign key in Donor.

District

Attribute Name	Description	Type
dis_id	District id	int
dis_name	Name of the district	varchar

Registration Staff

Attribute Name	Description	Type
rs_id	Id of the registration staff	int
rs_name	Name of the registration staff	varchar
rs_address	Address of the registration staff	varchar
rs_phone no	Phone no. of the registration staff	varchar

Blood Recipient

Attribute Name	Description	Type
r_name	Name of the recipient	varchar
r_id	Id of the recipient	int
r_sex	Sex of the recipient	char
quantity	Needed quantity of blood	int
r_reg date	Registration date of the recipient	date
r_bgrp	Recipient's blood group	varchar
rs_id (fk)	Id of the registration staff	int
dis_id(fk)	District id	int
bm_id(fk)	Blood bank manager's id	int

The relationship with Registration_staff and Blood_Recipient is 1 to many. Hence, the primary key of Registration_staff is used as a foreign key in Blood_Recipient.

The relationship with District and Blood_Recipient is 1 to many. Hence, the primary key of District is used as a foreign key in Blood_Recipient.

The relationship with Blood_Processing_Manager and Blood_Recipient is 1 to many. Hence, the primary key of Blood_Sample is used as a foreign key in Blood_Recipient.

Blood_Sample

Attribute Name	Description	Type
b_bgrp	Blood group of the sample	varchar
sample_no	Sample identification number	int
Covid plasma	Specifies whether it contains COVID antibodies	enum(yes/no)
bm_id(fk)	Blood Bank manager's id	int

The relationship with Blood bank manager and Blood_Sample is 1 to many. Hence, the primary key of the Blood processing manager is used as a foreign key in Blood_Sample.

Blood Bank Manager

Attribute Name	Description	Type
bm_id	Blood bank manager's id	int
bm_name	Blood bank manager's name	varchar
bm_phone no	Blood bank manager's phone no.	varchar

Hospital

Attribute Name	Description	Type
h_id	Hospital's id	int
h_name	Hospital's name	varchar
h_bgrp	Needed blood group	varchar
h_quantity	Needed quantity of blood in a hospital	int
dis_id(fk)	District's id	int
bm_id(fk)	Blood bank manager's id	int

The relationship with the District and Hospital is 1 to many. Hence, the primary key of the District is used as a foreign key in the Hospital.

The relationship with the Blood processing manager and Hospital is 1 to many. Hence, the primary key of Blood processing manager is used as a foreign key in Hospital.

Normalization

1. **Donor (d_id, d_name, d_sex, d_age, d_regdate, rs_id, dis_id, d_bgrp)**

{d_id} = > {d_name} (functional dependency exists, because two different names do not correspond to the same d_id).

{d_id} = > {d_sex} (functional dependency exists).

{d_id} = > {d_age} (functional dependency exists).

{d_id} = > {d_regdate (functional dependency exists).

{d_id} = > {rs_id} (functional dependency exists).

{d_id} = > {dis_id} (functional dependency exists).

{d_id} = > {d_bgrp} (functional dependency exists).

The relation is in 1NF because its attributes do not have sub attributes.

The relation is in second normal form, as it is in 1NF and every non-primary key attribute is fully functionally dependent on the primary key of the relation.

The relation is in third normal form, as it is in 2NF and no non-primary key attribute is transitively dependent on the primary key.

No part of the primary key is fully functionally dependent on non-primary key.
So, the relation is in BCNF

2. **District (dis_id , dis_name)**

{dis_id}= > {dis_name}

The relation is in BCNF.

3. **Registration_staff (rs_id, rs_name, rs_address,rs_phoneno)**

{rs_id} = > {rs_name} (functional dependency exists).

{rs_id} = > {rs_phoneno} (functional dependency exists).

{rs_id} = > {rs_address} (functional dependency exists).

The relation is in BCNF.

4. Blood_recipient (r_id, r_sex, r_regdate, r_name,quantity, r_bgrp, rs_id, dis_id, bm_id)

{r_id} = > {r_sex} (functional dependency exists).

{r_id} = > {r_bgrp} (functional dependency exists).

{r_id} = > {r_regdate} (functional dependency exists).

{r_id} = > {r_name} (functional dependency exists).

{r_id} = > {quantity} (functional dependency exists).

{r_id} = > {rs_id} (functional dependency exists).

{r_id} = > {dis_id} (functional dependency exists).

{r_id} = > {bm_id} (functional dependency exists).

The relation is in BCNF.

5. Blood_Sample (b_bgrp, sample_no, covid_plasma, bm_id)

{b_bgrp,sample_no} = > {covid_plasma} (functional dependency exists).
{b_bgrp,sample_no} = > {bm_id} (functional dependency exists).

The relation is in BCNF.

6. Blood_Bank_manager (bm_id, bm_name, bm_phoneno)

{bm_id} = >{bm_name}

{bm_id} = > {bm_phoneno} (functional dependency exists)

The relation is in BCNF.

7. Hospital (h_id,h_quantity, h_bgrp ,dis_id, bm_id, h_name)

{h_id}= > {h_name, dis_id, bm_id,h_bgrp}

{h_id} = >h_quantity (functional dependency exists)

The relation is in BCNF.

Queries

1. **Show the donors having the blood groups that are required by recipients living in the same district.**

```
mysql> SELECT d_id, d_name, r_id, r_name FROM Donor, Blood_Recipients
    -> WHERE d_bgrp=r_bgrp AND Donor.dis_id=Blood_Recipient.dis_id;
+------------+-----------+-----------+------------+
|   d_id     |  d_name   |   r_id    |  r_name    |
+------------+-----------+-----------+------------+
|   8923     |  Megha    |   9983    |  Ananya    |
|   8933     |  Tanvi    |   9281    |  Divyanshu |
|   8943     |  Rohan    |   9567    |  Nikhil    |
+------------+-----------+-----------+------------+
```

2. **Show the donor and recipients details having the same blood group registered by staff Tanya on the same date.**

```
mysql> SELECT d_id, d_name, r_id, r_name FROM Donor, Blood_Recipients,
    -> Registration_staff WHERE d_bgrp=r_bgrp AND d_regdate=r_regdate
    -> Donor.rs_id=Blood_Recipient.rs_id AND rs_name='Tanya';
+------------+-----------+-----------+------------+
|   d_id     |  d_name   |   r_id    |  r_name    |
+------------+-----------+-----------+------------+
|   8963     |  Prachi   |   9913    |  Akash     |
```

3. **Show detailed information of the recipients and hospitals in a district with code 08 who need A+ blood group.**

```
mysql> SELECT r_id, r_name, h_id, h_name FROM Hospital, Blood_Recipients
    -> WHERE r_bgrp='A+' AND dis_id=08 AND h_bgrp=r_bgrp;
+------------+-----------+-----------+------------+
|   r_id     |  r_name   |   h_id    |  h_name    |
+------------+-----------+-----------+------------+
|   9913     |  Akash    |   21      |  XYZ       |
+------------+-----------+-----------+------------+
```

4. **List the name, age and id of donor who is registered by registration staff ‘Bushra’ or who have B+ blood group**

```
mysql> SELECT d_id, d_name, d_age FROM Donor, Registration_Staff
    -> WHERE Donor.rs_id=Registration_Staff.rs_id
    -> UNION
    -> select d_id, d_name, d_age FROM Donor where d_bgrp='B+';
+------------+-----------+-----------+
|   d_id     |  d_name   |  d_age    |
+------------+-----------+-----------+
|   8923     |  Megha    |  25       |
|   8943     |  Rohan    |  32       |
+------------+-----------+-----------+
```

5. **Find out the donor name, id who is registered by registration staff whose name starts from a.Also show the name of the person.**

```
mysql> SELECT d_id, d_name, rs_name FROM Donor, Registration_staff
    ->  WHERE d_bgrp=r_bgrp AND Donor.rs_id=Registration_staff.rs_id
    ->  INTERSECTION
    ->  SELECT rs_name FROM Registration_staff where rs_name like "A%
+------------+-----------+-----------+
|   d_id     |  d_name   |  rs_name  |
+------------+-----------+-----------+
|   8963     |  Prachi   |  Aman     |
+------------+-----------+-----------+
```

6. **Show the COVID-19 plasma containing blood samples verified by Dr. Rama.**

```
mysql> SELECT sample_no, b_bgrp FROM Blood_sample, Blood_bank_Manager
    -> WHERE Blood_sample.bm_id=Blood_bank_Manager.bm_id
    -> AND
    -> bm_name='Rama' AND covid_plasma='Yes';
+------------+-----------+
| sample_no  |  b_bgrp   |
+------------+-----------+
|   452      |  A+       |
|   198      |  B+       |
+------------+-----------+
```

7. **Count the number of covid plasma blood sample having blood group=’B-’**

```
mysql> SELECT COUNT(*) AS Number FROM Blood_sample where b_bgrp='B-'
    -> AND covid_plasma="Yes";
+------------+
|   Number   |
+------------+
|   1        |
+------------+
```