DBMS Project
Coding Platform Database

What is Blood Bank?

- A blood bank is a cache or bank of blood or blood components, gathered as a result of blood donation or collection,stored and preserved for later use in blood transfusion.

- The term "blood bank" typically refers to a division of a hospital where the storage of blood products occurs and where proper testing is performed.

Need of Blood Bank Management System

Blood Bank donation systems can collect blood from many donors in short from various sources and distribute that blood to needy people who require blood.

Problem Statement

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received has to be managed thoroughly so that there will be no negative effect to the blood receiver once they receive blood. From the observations and interviews conducted that have been made during the user requirements phase.

The blood donation event schedule is normally advertised to the public so that they are aware of the blood donation campaign period. At the blood house unit, the staff and nurses only are informed about the blood donation schedule for each month on the whiteboard at the blood house. So they are using a manual way of informing the schedule. The problem arises when the space provided is not enough. The medium used to inform the staff about the schedule of the month is using a whiteboard and it is written by using a whiteboard marker. Therefore, the writing tends to become unclear. The public did not have knowledge about blood donation.

Blood Bank Information System Database Design

The project entitled “Blood Bank Database” is a database system that records and manages the transactions of blood donation and blood issuance. The main purpose of this system is to keep an organized records management of blood inventory. It would be a great help in the proper monitoring of blood available in the blood bank and for easy processing of blood requests.

1. Assumption :

Our E-R diagram represents the Blood-Bank Management system. It has eight entity sets. They are–

Donor,DIstrict,Registration_staff,Disease_Recognizer,Blood_Processing_Manager, Blood_Sample,Hospital,Blood_Recipient.

1.1 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, donor registration date (dreg_date) and blood group will be stored in the database under entity Donor.

a) **Donar Attributes- (don_Name,don_Id,sex,age,don_reg_date,don_b_grp)**

1.2 District

Every district’s/location’s id is different (primary key).

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

1.3 Registration_Staff

Registration staff will register the information of donors and the recipients.

c) Registration_Staff: Attributes-

(rs_id, rs_name, sex)

1.4 Disease_Recognizer

Disease recognizer will test blood samples whether the samples are contaminated or okay.

d) Disease_Recognizer: Attributes-

(drecog_id, drecog_name, sex).

1.5 Blood_Processing_Manager

They will take orders from the hospitals and fulfill their needed requirements of blood samples.

e) Blood_Processing_Manager: (Attributes-bm_id, bm_name, sex).

1.6 Blood_Sample

The quantities of blood that the Blood_bank has. Their group, sample_no, status will be stored.

f) Blood_Sample: Attributes-

(b_group, sample_no,status)

1.7 Hospital

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

g) Hospital: Attributes-

(hId, hName, hb_grp, hb_qnty).

1.8 Blood_Recipient

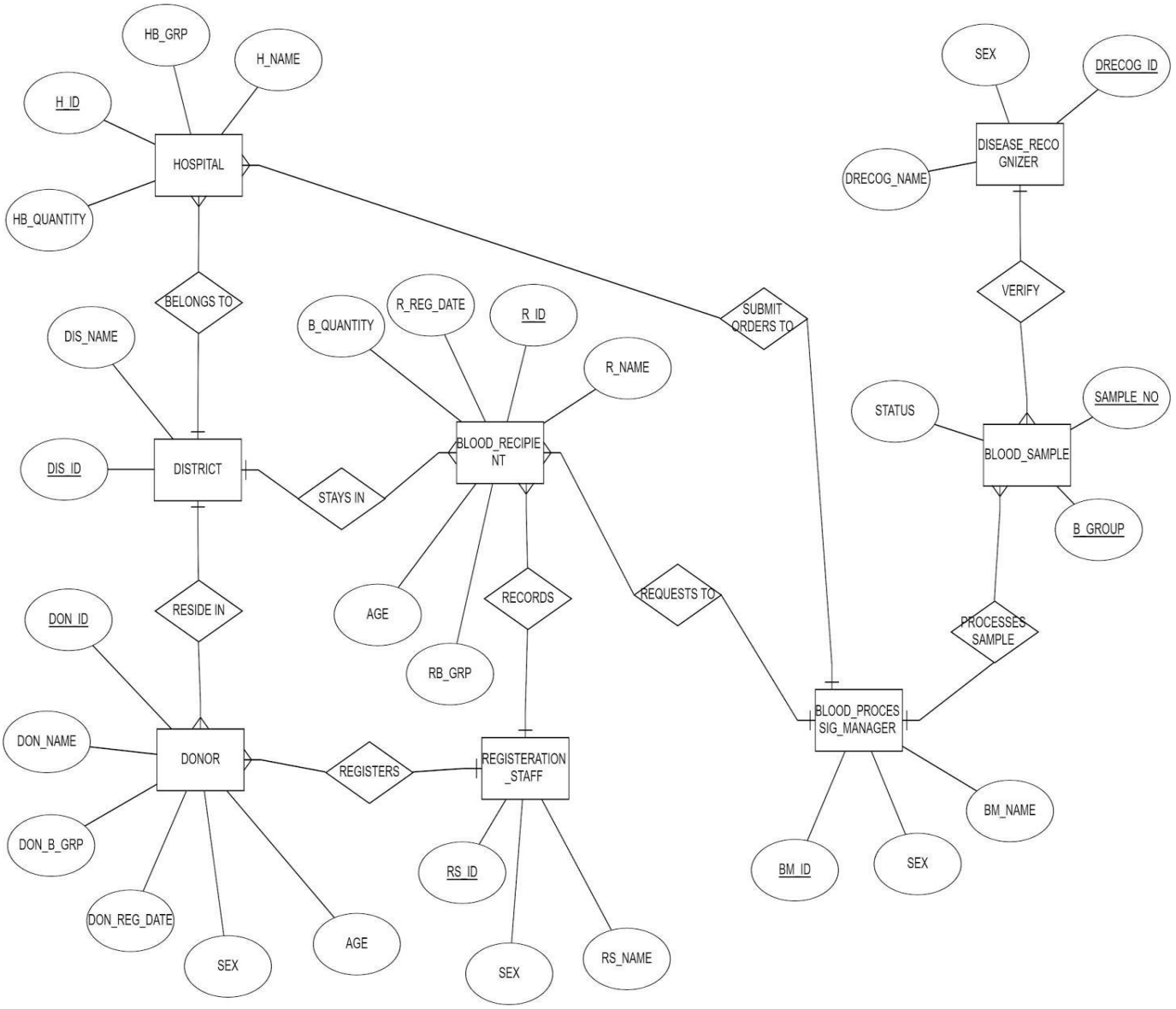
Who needs blood. A recipient's id, name, age, sex, the blood sample's group information will be stored in database.

h) Blood_Recipient: Attributes-

(rId, sex, age, r_regdate, rName, b_qnty, rb_grp).

2. Entity Relationship Diagram :

The below figure shows the Entity Relation diagram of Blood Bank Management System.



Cardinality:

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

Registration_Staff & Donor- (Relationship-(registers), 1 to many). 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-(resides_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_Processing_Manager & Hospital- (Relationship-(submit_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_Processing_Manager & Blood_Sample-(Relationship-(processes_sample), 1 to many). A manager can process many samples of blood. One blood sample can be processed by one blood processing manager.

Disease_Recognizer & Blood_sample- (Relationship-(verify), 1 to many). A disease recognizer can verify many blood samples. One blood sample is verified by one disease recognizer.

Blood_Processing_Manager & Blood_Recipient-(Relationship-(request_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.

2.2 Relational Schemas

Donor

Attribute Name	Description	Type
don_Name	Name of the donor	varchar
Don_id	Id of the donor	int
sex	Sex of donor	char
Age	Age of donor	int
don_reg_date	Registration date of the donor	date
rs_id(fk)	Id of the registration staff	int
dis_id(fk)	District id	int

don_b_grp	Donor's blood group	varchar

The relationship with Registration_staff and Donor is 1 to many. That's why primary key of Registration_staff is used as a foreign key in Donor.
The relationship with District and Donor is 1 to many. That's why primary key of District is used as a foreign key in Donor.

DistrictTable

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
Sex	Sex of the registration staff	char

Blood_Recipient

Attribute Name	Description	Type
Rid	Id of the recipient	int
Sex	Sex of the recipient	char
Age	Age of the recipient	int
r_regdate	Registration date of the recipient	date
Rname	Name of the recipient	varchar
b_qnty	Needed quantity of blood	int
rb_grp	Recipient's blood group	varchar
rs_id (fk)	Id of the registration staff	int
dis_id (fk)	District id	int
bm_id (fk)	Blood processing manager's id	int

The relationship with Registration_staff and Blood_Recipient is 1 to many. That's why primary key of Registration_staff is used as a foreign key inBlood_Recipient.

The relationship with District and Blood_Recipient is 1 to many. That's why 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. That's why primary key of Blood_Sample is used as a foreign key in Blood_Recipient.

Blood_Sample

Attribute Name	Description	Type
b_group	Blood group of the sample	varchar
sample_no	Sample identification number	int
Status	Status of the blood sample	varchar
drecog_id (fk)	Disease Recognizer's id	int
bm_id (fk)	Blood processing manager's id	int

The relationship with Disease_Recognizer and Blood_Sample is 1 to many. That's why primary key of Disease_Recognizer is used as a foreign key in Blood_Sample.
The relationship with Blood processing manager and Blood_Sample is 1 to many. That's why primary key of Blood processing manager is used as a foreign key in Blood_Sample.

Disease_Recognizer

Attribute Name	Description	Type
drecog_id	Disease Recognizer's id	Int
drecog_name	Disease Recognizer's name	varchar
Sex	Disease Recognizer's sex	char

Hospital

Attribute Name	Description	Type
Hid	Hospital's id	int
hb_qnty	Needed quantity of blood in a hospital	int
hb_grp	Needed blood group	varchar
HName	Hospital's Name	varchar
dis_id(fk)	District's id	int
bm_id(fk)	Blood processing manager's id	int

Blood_Processing_Manage

Attribute Name	Description	Type
bm_id	Blood processing manager's id	int
bm_name	Blood processing manager's name	varchar
Sex	Blood processing manager's sex	char

The relationship with District and Hospital is 1 to many. That's why primary key of District is used as a foreign key in Hospital.
The relationship with Blood processing manager and Hospital is 1 to many. That's why primary key of Blood processing manager is used as a foreign key in Hospital.

2.3 Normalization

it is the sequence of steps by which a relational database model is both created and improved upon. The sequence of steps involved in the normalization process is called normal forms. Essentially, normal forms applied during a process of normalization allow creation of a relational database model as a step-by-step progression.

Advantages of normalization:

i. Many unnecessary redundancies are avoided.
ii. Anomalies with input, deletion and updates can be avoided.
iii. Fully normalized, relations tend to need less space than if not normalized.

Disadvantages of normalization:

i. Normalization splits entities and relationships into many relations, thus making them harder to understand.
ii. Queries become more complex because they have to involve more relations.
iii. Response times are longer because of a higher number of joins in the queries.

Normalization of Blood Bank database:

1. **Donor** (don_Id, don_Name, sex, age, don_reg_date, rs_id, dis_id, don_b_grp)
{don_Id} = > {dName} (functional dependency exists, because two different don_Names do not correspond to the same dId).
{don_Id} = > {sex} (functional dependency exists).
{don_Id} = > {age} (functional dependency exists).
{don_Id} = > {don_reg}_date (functional dependency exists).
{don_Id} = > {rs_id} (functional dependency exists).
{don_Id} = > {dis_id} (functional dependency exists).
{don_Id} = > {don_b_grp} (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 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 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

3. **Registration_staff** (rs_id, rs_name, sex)
{rs_id} = > {rs_name} (functional dependency exists).
{rs_id} = > {sex} (functional dependency exists).
The relation is in 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

4. **Blood_recipient** (rld, sex, age, r_regdate, rName, b_qnty, rb_grp, rs_id, dis_id, bm_id)
{rld} = > {sex} (functional dependency exists).
{rld} = > {age} (functional dependency exists).
{rld} = > {r_regdate} (functional dependency exists).
{rld} = > {rName} (functional dependency exists).
{rld} = > {b_qnty} (functional dependency exists).
rld} = > {rb_grp} (functional dependency exists).
{rld} = > {rs_id} (functional dependency exists).
{rld} = > {dis_id} (functional dependency exists).
{rld} = > {bm_id} (functional dependency exists).

The relation is in 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

5. **Blood_Sample** (b_group, sample_no, status, drecog_id, bm_id)
{b_group,sample_no} = > {status} (functional dependency exists).
{b_group,sample_no} = > {drecog_id} (functional dependency exists).
{b_group,sample_no} = > {bm_id} (functional dependency exists).

The relation is in 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

6. **Disease_recognizer (** drecog_id, drecog_name, sex)
{drecog_id} = > {drecog_name}.
{drecog_id} = > {sex} (functional dependency exists).

The relation is in 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

7. **Blood_processing_manager** (bm_id, bm_name, sex)
{bm_id} = >{bm_name}
{bm_id} = > {sex} (functional dependency exists)

The relation is in 1NF.
The relation is in second normal form.
The relation is in third normal form.
The relation is in BCNF.

8. **Hospital** (hId, hb_qnty, hb_grp ,dis_id, bm_id, hName)
{hId}= > {hName, dis_id, bm_id}
{hId, hb_grp} = > hb_qnty (functional dependency exists)

The relation is in 1NF.
The relation is not in second normal form, as it is in 1NF but not every non-primary key attribute is fully functionally dependent on the primary key of the relation.
So, we have to split the relation.
Hospital_1(hId, hName,dis_id,bm_id).
Hospital_2(hId, hb_grp, hb_qnty)
Now it is in 2NF.
The relation is in third normal form.
The relation is in BCNF

2.4 Tables with sample values after Normalization

Blood_Processing_Manager

```
CREATE TABLE Blood_Processing_Manager(bm_id NUMBER,
  bm_name varchar(10),
  SEX VARCHAR(5)    );
 alter  table blood_processing_manager add primary key(BM_id);
  insert INTO blood_processing_manager values(6,'DEEPA','F');
  insert INTO blood_processing_manager values(36,'MEHRAB','M');
  insert INTO blood_processing_manager values(47,'URMI','F');
  insert INTO blood_processing_manager values(74,'DINAR','M');
  SELECT
      *
  FROM BLOOD_PROCESSING_MANAGER;
```

```
Table BLOOD_PROCESSING_MANAGER created.


Table BLOOD_PROCESSING_MANAGER altered.
```

```
     BM_ID BM_NAME    SEX
---------- ---------- -----
         6 DEEPA      F
        36 MEHRAB     M
        47 URMI       F
        74 DINAR      M
```

DISTRICT

```
CREATE TABLE District (dis_id int primary key , dis_name VARCHAR(15)) ;

 insert INTO District values(10,'DHAKA');
  insert INTO District values(20,'KHULNA');
   insert INTO District values(30,'RAJASHI');
    insert INTO District values(40,'CHITTAGONG');
     insert INTO District values(50,'BARISHAL');
      insert INTO District values(60,'SYLHET');
       insert INTO District values(70,'RANGPUR');
 SELECT
     *
 FROM District;
```

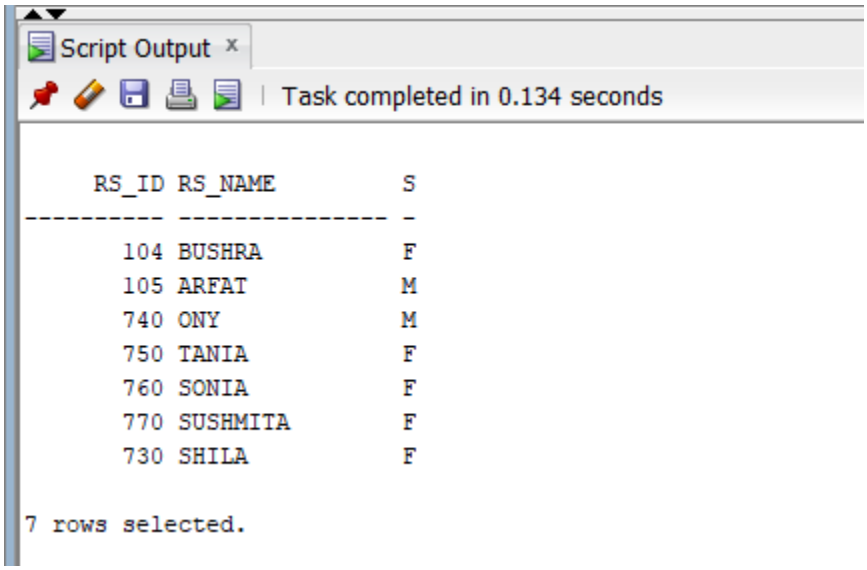
```
Table DISTRICT created.
```

```
    DIS_ID DIS_NAME
---------- ---------------
        10 DHAKA
        20 KHULNA
        30 RAJASHI
        40 CHITTAGONG
        50 BARISHAL
        60 SYLHET
        70 RANGPUR

7 rows selected.
```

Registration_Staff

```
CREATE TABLE Registration_staff (rs_id INT PRIMARY KEY, rs_name
VARCHAR(15), sex CHAR)    ;

  insert INTO Registration_staff values(104,'BUSHRA','F');
   insert INTO Registration_staff values(105,'ARFAT','M');
    insert INTO Registration_staffvalues(730,'SHILA','F');
     insert INTO Registration_staff values(740,'ONY','M');
      insert INTO Registration_staff values(750,'TANIA','F');
     insert INTO Registration_staff values(760,'SONIA','F');
        insert INTO Registration_staff values(770,'SUSHMITA','F');
  SELECT * FROM Registration_staff;
```

```
Table REGISTRATION_STAFF created.
```




```
    RS_ID RS_NAME         S
--------- --------------- -
       104 BUSHRA          F
       105 ARFAT           M
       740 ONY             M
       750 TANIA           F
       760 SONIA           F
       770 SUSHMITA        F
       730 SHILA           F

7 rows selected.
```

Disease_Recognizer

```
CREATE TABLE Disease_recognizer ( drecog_id int PRIMARY KEY , drecog_name
VARCHAR(15), sex char )  ;
    insert INTO Disease_recognizer values(401,'JAMIL','M');
    insert INTO Disease_recognizer values(501,'MILA','F');
   insert INTO Disease_recognizer values(601,'HELAL','M');
```

```
 insert INTO Disease_recognizer values(810,'SHILA','F');
SELECT  * FROM Disease_recognizer;
```

```
Table DISEASE_RECOGNIZER created.
```

Script Output | Task completed in 0.145 seconds

```
 DRECOG_ID DRECOG_NAME     S
---------- --------------- -
       401 JAMIL           M
       501 MILA            F
       601 HELAL           M
       810 SHILA           F
```

Blood_Sample

```
CREATE TABLE  Blood_Sample ( b_group VARCHAR(5) , sample_no INT , status
VARCHAR(6), drecog_id INT, bm_id  INT
, FOREIGN KEY(drecog_id) references disease_Recognizer(drecog_id),
FOREIGN KEY(bm_id) references Blood_Processing_Manager(bm_id)) ;
    insert INTO Blood_Sample values('A+',305,'NO',601,36);
    insert INTO Blood_Sample values('A+',401,'YES',810,6);
     insert INTO Blood_Sample values('B+',405,'YES',810,47);
      insert INTO Blood_Sample values('O+',202,'YES',501,74);
       insert INTO Blood_Sample values('O+',410,'NO',810,74);
  SELECT  * FROM Blood_Sample;
```

Script Output | Task completed in 0.256 seconds

```
Table BLOOD_SAMPLE created.
```

B_GRO	SAMPLE_NO	STATUS	DRECOG_ID	BM_ID
A+	305	NO	601	36
A+	401	YES	810	6
B+	405	YES	810	47
O+	202	YES	501	74
O+	410	NO	810	74

Blood_Recipient

```
CREATE TABLE Blood_recipient (
                          rId int PRIMARY KEY , sex char , age int ,
                        r_regdate date, rName varchar(15),  b_qnty int,
                    rb_grp VARCHAR(10), rs_id int , dis_id int , bm_id int
,
       FOREIGN KEY(dis_id) references District(dIS_ID)
FOREIGN KEY(rs_id) references Registration_Staff(rs_id),
    FOREIGN KEY(bm_id) references Blood_Processing_Manager(bm_id));
```

```
insert INTO Blood_recipient
values(44,'M',23,'8-JAN-2020','TOMMY',1,'A+',730,10,6);
insert INTO Blood_recipient
values(87,'F',22,'2-JAN-2020','SOPNIL',2,'B+',770,10,36);
insert INTO Blood_recipient
values(88,'M',23,'1-JAN-2020','SHOHAG',1,'A+',760,10,36);
insert INTO Blood_recipient
values(90,'F',24,'5-FEB-2020','FARZANA',1,'O+',730,20,74);
 SELECT
                                    *
                                FROM Blood_recipient;
```

```
       RID S        AGE R_REGDATE   RNAME              B_QNTY RB_GRP         RS_ID     DIS_ID      BM_ID
---------- - ---------- ----------- --------------- ---------- ---------- ---------- ---------- ----------
        88 M         23 01-JAN-2020 SHOHAG                  1 A+                760         10         36
        44 M         23 08-JAN-2020 TOMMY                   1 A+                730         10          6
        87 F         22 02-JAN-2020 SOPNIL                  2 B+                770         10         36
        90 F         24 05-FEB-2020 FARZANA                 1 O+                730         20         74
```

Donor

```
create table  Donor (don_Id INT PRIMARY KEY , don_Name VARCHAR(15),
                    sex CHAR, age INT , don_reg_date DATE, rs_id INT ,
                    dis_id INT , don_b_grp VARCHAR(5),
                    FOREIGN KEY(dis_id) references District(dis_id),
                FOREIGN KEY(rs_id) references Registration_Staff(rs_id));
```

```
Table DONOR created.
```

```
insert INTO Donor values(3,'NASIF','M',23,'2-JAN-2020',105,10,'A+');
insert INTO Donor values(7,'NIMI','F',22,'2-JAN-2020',104,10,'B+');
insert INTO Donor values(10,'JENIFER','F',22,'1-JAN-2020',730,10,'O+');
insert INTO Donor values(14,'TANZIMA','F',22,'1-JAN-2020',760,30,'A+');
insert INTO Donor values(16,'KANIZ','F',22,'3-JAN-2020',740,30,'B+');
  SELECT
      *
  FROM dONOR;
```

```
    DON_ID DON_NAME        S        AGE DON_REG_DAT      RS_ID     DIS_ID DON_B
---------- --------------- - ---------- ----------- ---------- ---------- -----
         3 NASIF           M         23 02-JAN-2020        105         10 A+
         7 NIMI            F         22 02-JAN-2020        104         10 B+
        10 JENIFER         F         22 01-JAN-2020        730         10 O+
        14 TANZIMA         F         22 01-JAN-2020        760         30 A+
        16 KANIZ           F         22 03-JAN-2020        740         30 B+
```

Hospital_1

```
 CREATE TABLE  Hospital_1(hId INT PRIMARY KEY , hName VARCHAR(20),dis_id
INT ,bm_id INT,
  FOREIGN KEY(dis_id) references District(dis_id),
 FOREIGN KEY(bm_id) references Blood_Processing_Manager(bm_id)) ;
```

```
Table HOSPITAL_1 created.
```

```
insert INTO hospital_1 values(910,'DHAKA MEDICAL',10,6);
insert INTO hospital_1 values(920,'KHULNA MEDICAL',20,36);
 insert INTO hospital_1 values(930,'RAJASHAHI MEDICAL',30,74);
  insert INTO hospital_1 values(940,'CHITTAGONG MEDICAL',40,47);

   SELECT  * FROM hospital_1;
```

HID	HNAME	DIS_ID	BM_ID
910	DHAKA MEDICAL	10	6
920	KHULNA MEDICAL	20	36
930	RAJASHAHI MEDICAL	30	74
940	CHITTAGONG MEDICAL	40	47

Hospital_2

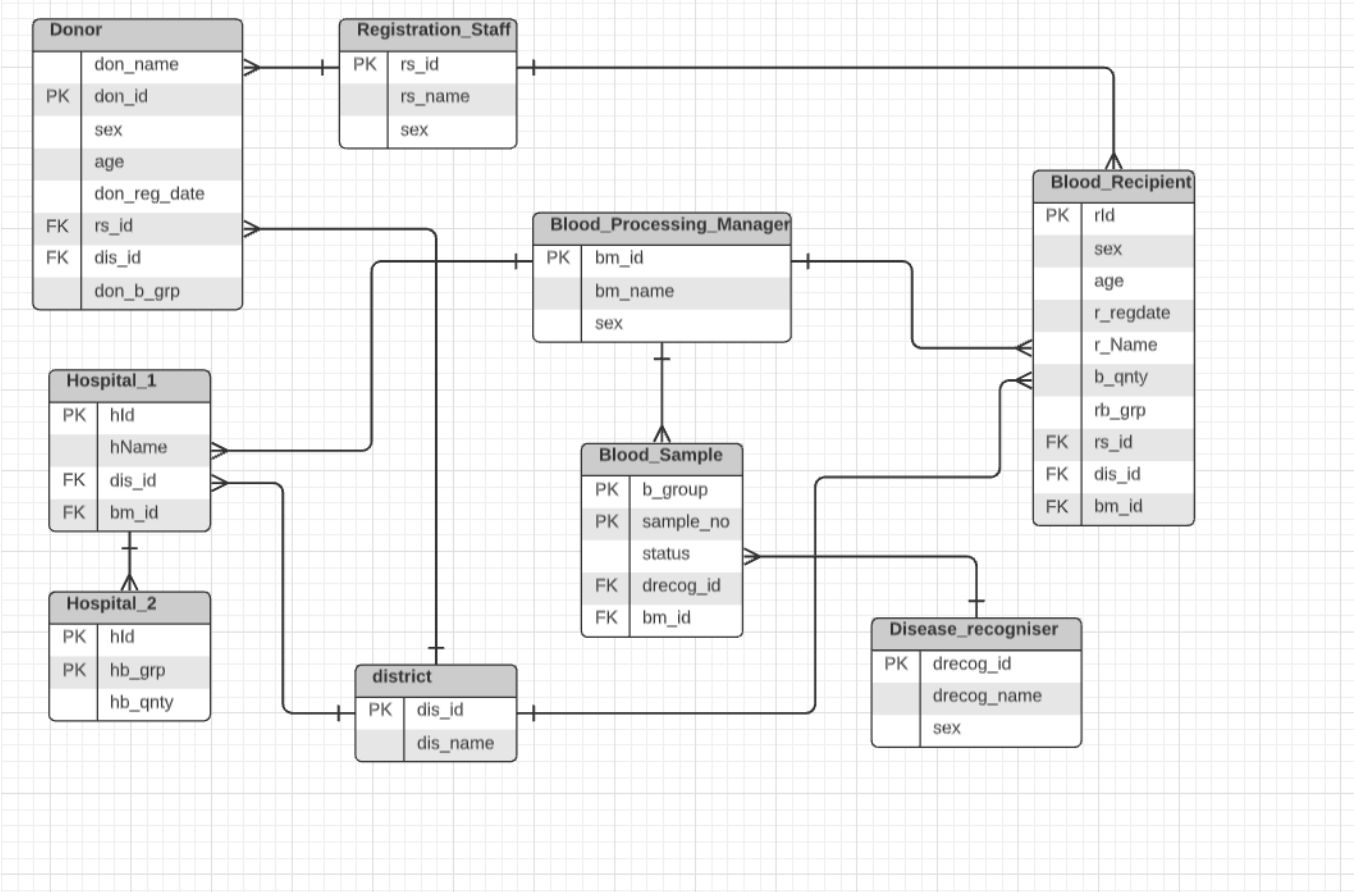
```
 CREATE TABLE  Hospital_2( hId INT  ,
                         hb_grp  VARCHAR(5), hb_qnty INT ,
,FOREIGN KEY(HID) REFERENCES HOSPITAL_1(HID)) ;
```

```
Table HOSPITAL_2 created.
```

```
insert INTO hospital_2 values(910,'O+',30);
  insert INTO hospital_2 values(920,'A+',40);
   insert INTO hospital_2 values(910,'B+',50);
    insert INTO hospital_2 values(910,'A+',10);
     insert INTO hospital_2 values(910,'O+',20);
     SELECT  * FROM hospital_2;
```

HID	HB_GR	HB_QNTY
910	O+	30
920	A+	40
910	B+	50
910	A+	10
910	O+	20

Implementation in SQL Server-



Blood-Bank Management Database System

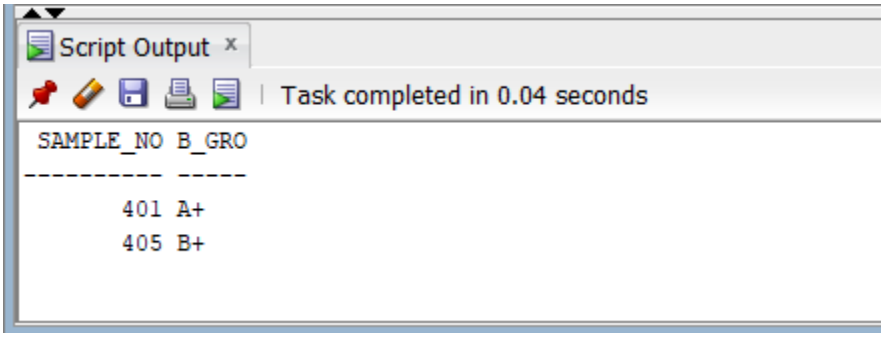
2.5 QUERIES :

1. Show the uncontaminated blood samples verified by Dr. Shila.

QUERY

```
SELECT SAMPLE_NO,B_GROUP
FROM blood_sample BS, disease_recognizer DR
WHERE  bs.drecog_id=dr.drecog_id
            AND
       DRECOG_NAME='SHILA'
            AND
    STATUS='YES';
```

OUTPUT

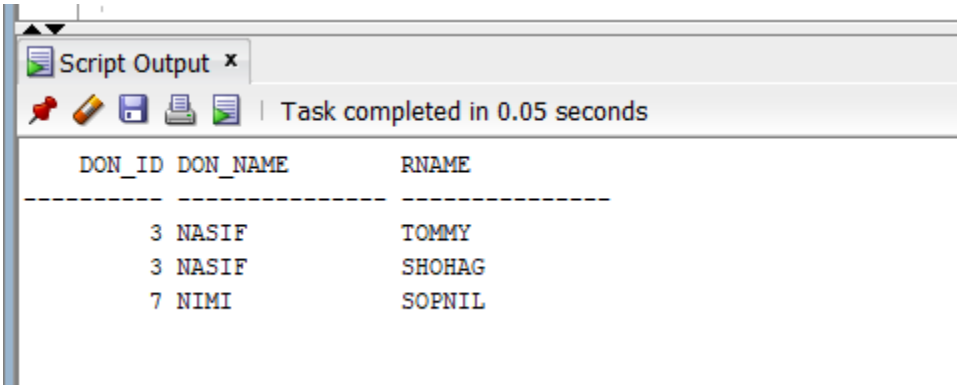


2. Show the donors having the blood groups that are required by recipients living in the same district. Show the recipient details also.

QUERY

```
SELECT donor.don_id,donor.don_name,blood_recipient.rname
FROM DONOR,  blood_recipient
WHERE  DON_B_GRP=RB_GRP
            AND
       donor.dis_id=blood_recipient.dis_id;
```

OUTPUT

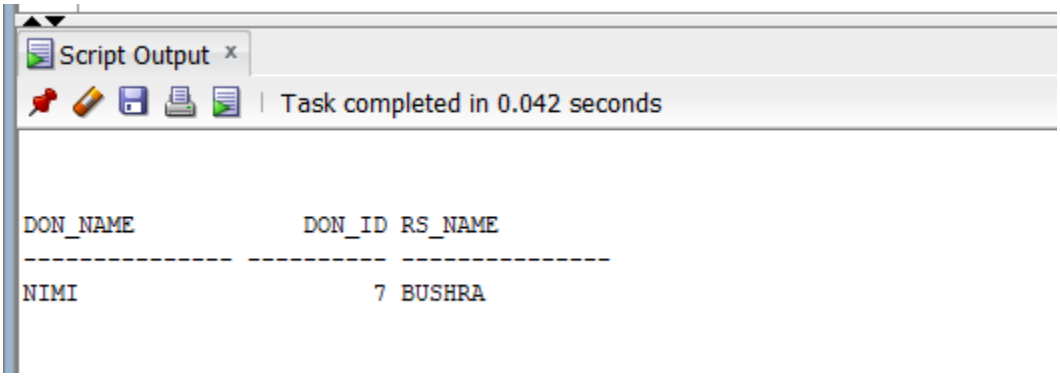


3. Find out donor name, id who is registered by registration staff_id '104' and show the registration staff's name also.

QUERY

```
SELECT donor.don_name,donor.don_id,registration_staff.rs_name
FROM DONOR,registration_staff
WHERE  donor.rs_id=registration_staff.rs_id
                AND
       registration_staff.rs_id=104;
```

OUTPUT



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

QUERY

```
SELECT donor.don_name,donor.don_id,donor.age
FROM DONOR,registration_staff
WHERE  donor.rs_id=registration_staff.rs_id
                UNION
       SELECT DON_NAME,AGE,don_id
       FROM DONOR
       WHERE donor.don_b_grp='B+';
```

OUTPUT

DON_NAME	AGE	DON_ID
NIMI	22	7
KANIZ	22	16

5. Find out all information about hospital_1 which has not been processed by the blood processing manager having id '6'.

QUERY

```
SELECT HNAME,HID,BM_ID,DIS_ID
FROM hospital_1
WHERE BM_ID NOT IN
     (  SELECT BM_ID
       FROM blood_processing_manager
       WHERE BM_ID=6);
```

OUTPUT

HNAME	HID	BM_ID	DIS_ID
RAJASHAHI MEDICAL	930	74	30
KHULNA MEDICAL	920	36	20
CHITTAGONG MEDICAL	940	47	40

THE END