Blood Donation Management System

GROUP MEMEBERS:

1.Omar Khaled (LEADER)

2.Abdulrahman Alnowaiser

3.Yazan Alansari

4.Loay Aloqbi

Table of tasks

Task	Done by
Stage 1	All members
Stage 2	All members
Stage 3	All members

Project proposal

Introduction: We need a system to manage blood donations. This will help hospitals and blood banks manage workers and donors and track appointments and inventory.

Objective: help blood banks handle donations and appointments and inventory.

Scope: editing of donors; planning of appointments; inventory tracking; security brushing.

Advantages: This system will make us faster and safer. We will be able to manage our resources, get more work done, and save more donors.

Project Analysis:

1.Problems:

- Paperwork: not only is hardly surprising, but also slows everything down and results in occasional mistakes.
- Insufficient information: ignorance of real-time data doesn't allow for proper preparation.
- Lack of rule-following: it is unwise to expect compliance with all the rules without a system.
- Lack of donors' interest: without good accessibility of the system, it is hard to motivate them.

2.Solutions:

- Digitalization: replacing the paper system with a digital one will vastly increase the speed of work and avoid many of the mistakes.
- Coordination: the linking of our system with other systems will make things greatly smoother.
- Apps: creation of an app for donors will create additional incentives and make the process more accessible.
- Analysis: the analysis of the data available will make our planning better and will allow us to make smarter decisions.

3.Benefits:

- Save Money: If we went digital and got better at everything, we'd save money.
- Better Health: If we made sure all blood was safe, patients would do better.
- Build Trust: If we showed that we cared about safety, donors would trust us.

4.Risks:

- Protect Data: We must keep donor data safe at all costs.
- Technical Hurdles: Making everything work well together would be hard.
- Follow Rules: We have to follow all the rules or face serious problems.

5.Implementation:

- Start Small: Begin small and slow to avoid causing huge problems.
- Talk to Everyone: Make sure you understand what everyone wants and needs.
- Support Staff: Train and help everyone to make sure the system works.

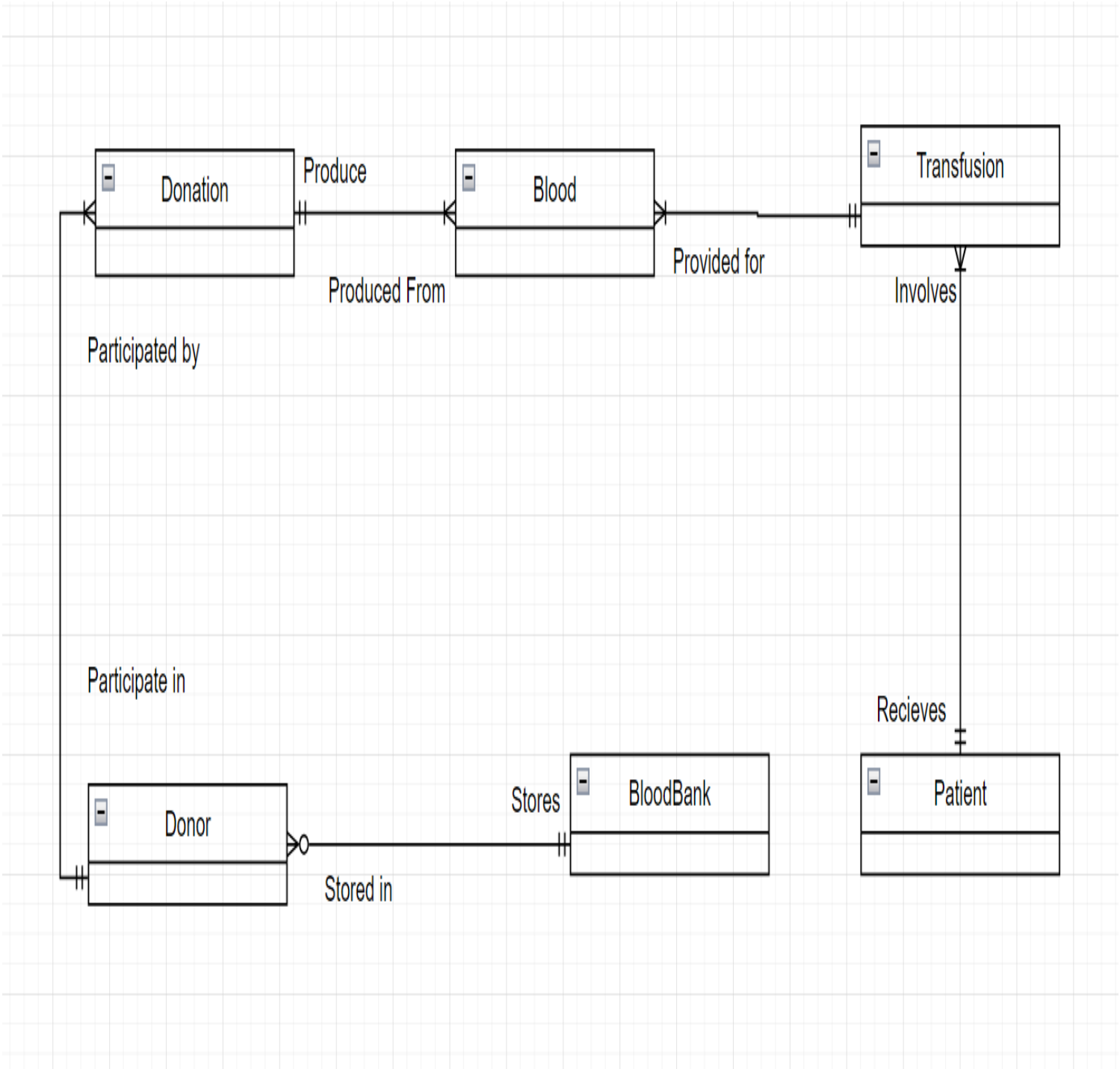
6.Future Opportunities:

- Expand Services: We can use the system for more than just blood donations.
- Collaborate: Working with others can help us learn and improve.
- Engage the Community: Making it easy for the community to donate will save more lives.

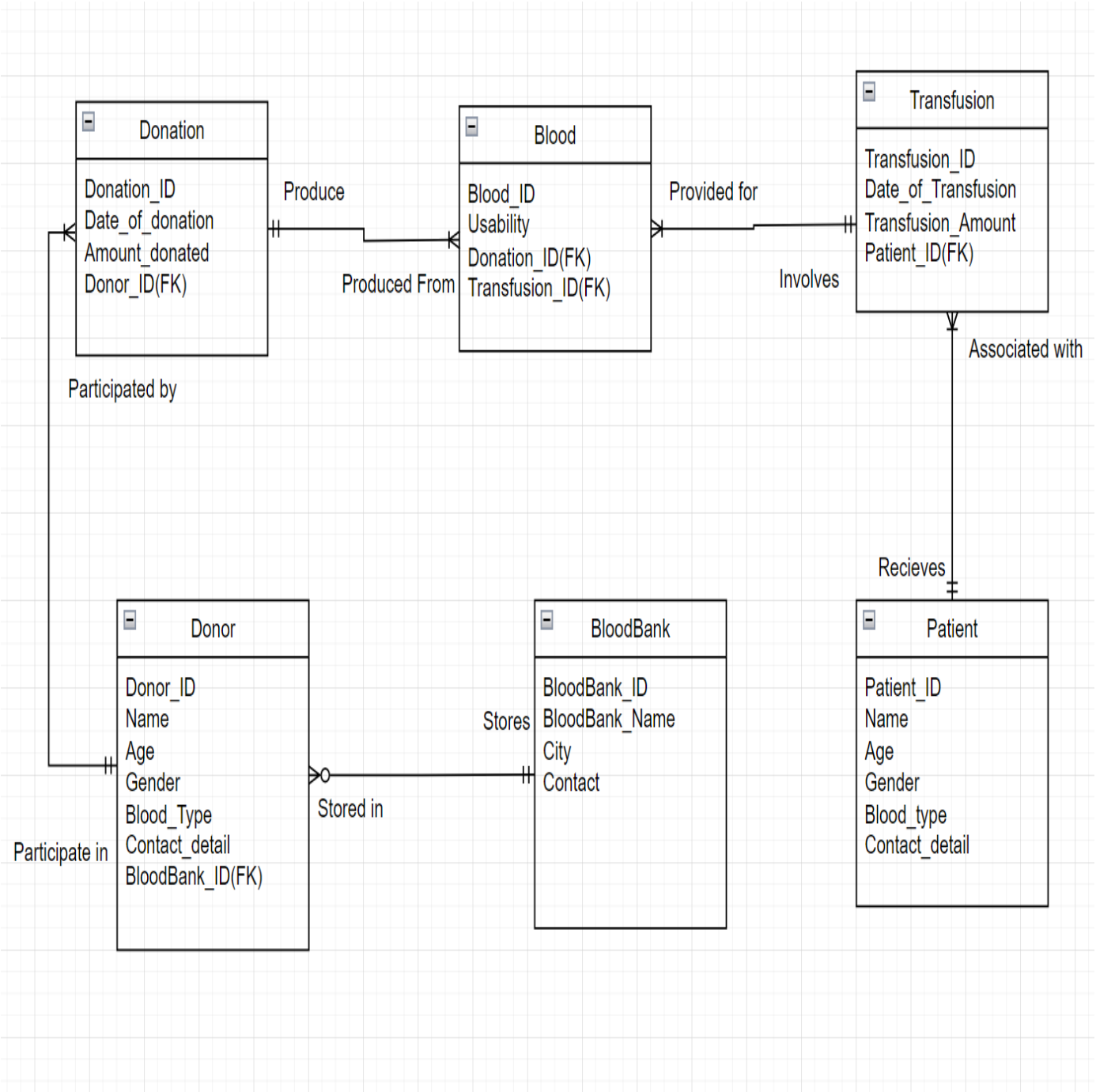
Entites	Attributes of entities	Data Type	Data Descreption
Donation	Donation_ID	Single&SimplePK	Identifier number for the donation
	Date_Of_Donation	Simple	The day of the donation
	Amount_Donated	Simple	The amount that donated
	Donor_ID	Single&SimpleFK	Identifier number for the donor
Blood	Blood_ID	Single&SimplePK	Identifier number for the Blood
	Usability	Simple	The status of the blood
	Donation_ID	Single&SimpleFK	Identifier number for the donation
	Transfusion_ID	Single&SimpleFK	Identifier number for the Transfusion
Transfusion	Transfusion_ID	Single&SimplePK	Identifier number for the Transfusion
	Date_of_ Transfusion	Simple	The day of the Transfusion
	Transfusion_Amount	Simple	The amount of Transfusion
	Patient_ID	Single&SimpleFK	Identifier number for the patient

Patient	Patient_ID Name Age Gender Blood_Type Contace_Detail	Single&SimplePK Simple Simple Simple Simple Simple	Identifier number for the Patient Patient's name Patient's age Patient's gender Patient's blood type Patient's details of contact
BloodBank	BloodBank_ID BloodBank_Name City Contact	Single&SimplePK Simple Simple Simple	Identifier number for the Bloodbank Bank's Name The city of the bank Bank's contacts
Donor	Donor_ID Name Age Gender Blood_Type Contact_detail BloodBank_ID	Single&SimplePK Simple Simple Simple Simple Simple Single&SimpleFK	Identifier number for the Donor Donor's Name Donor's Age Donor's Gender Donor's Blood type Donor's details of contact Identifier number for the Bloodbank

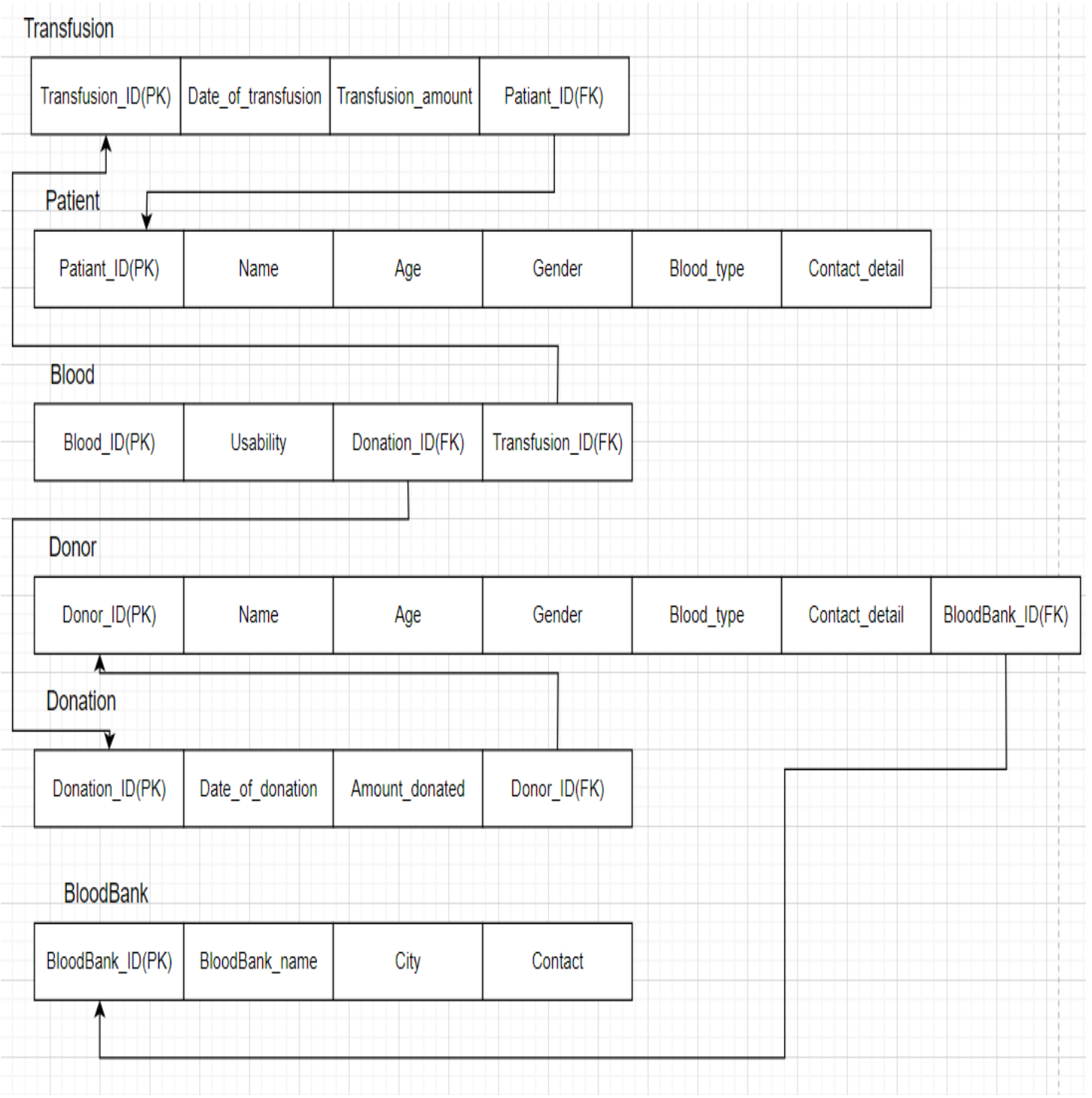
Conceptual ER Model



ERD Model


Donation
Donation_ID
Date_of_donation
Amount_donated
Donor_ID(FK)
Produce
Produced From
Blood
Blood_ID
Usability
Donation_ID(FK)
Transfusion_ID(FK)
Provided for
Involves
Transfusion
Transfusion_ID
Date_of_Transfusion
Transfusion_Amount
Patient_ID(FK)
Associated with
Recieves
Participated by
Participate in
Donor
Donor_ID
Name
Age
Gender
Blood_Type
Contact_detail
BloodBank_ID(FK)
Stored in
Stores
BloodBank
BloodBank_ID
BloodBank_Name
City
Contact
Patient
Patient_ID
Name
Age
Gender
Blood_type
Contact_detail

Relational Model



Normalization

1-unNormalized

BloodBankDataBase(

Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,Date_of_Donation,Amount_Donated

(Blood(Blood_ID,Usablity,Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details,Date_of_Transfusion

(BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)))

2-First Normal Form

To get our table to the 1NF there must be no repeating Groups and multivalued attributes

Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,Blood,Date_of_Donation,Amount_Donated,Bank_ID(FK))

Blood (Blood_ID, usability, Patient_ID,
Name,Age,Gender,Blood_Type,Contact_Details,
Date_of_Transfusion)

BloodBank(BloodBank_ID,Bank_Name,City,Contact)

We split the groups into 3 groups instead of 1 and we connect them by using primary/foreign keys

3-Second Normal Form

To get our table in the 2NF the table must be in the 1NF and all non-key attributes is functional and dependent on the primary key

For example, Blood group the only thing that is depended on the primary key is usability and the name age gender … all depended on patient so we split them into 2 groups

Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID(FK))

Blood(Blood_ID,Usability,Patient_ID(FK),Donor_ID(FK))

Transfusion(Transfuion_ID,Date_of_Transfusion,Transfusion_Amount,

Patient_ID(FK),Blood_ID(FK))

Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)

Donation(Donation_ID,Date_of_Donation,Amount_Donated,Donor_ID(FK),

Blood_ID(FK))

BloodBank(BloodBank_ID,Bank_Name,City)

In 1NF we had attributes that were depended one group and others were depended on another while having attributes that were depended on both so we split them into 3 tables

4-Third Normal Form

To get our table in 3NF it must be first in 1NF and 2NF while having no Transitive dependencies between Attributes

Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID(FK))

Blood(Blood_ID,Usability,Donation_ID(FK),Transfusion_ID(FK))

Transfusion(Transfuion_ID,Date_of_Transfusion,Transfusion_Amount,Patient_ID(FK))

Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)

Donation(Donation_ID,Date_of_Donation,Amount_Donated,Donor_ID(FK))

BloodBank(BloodBank_ID,Bank_Name,City)

Function Dependencies

Blood_ID : Usability

Patient_ID:Name,Age,Gender,Blood_Type ,Contact_Details

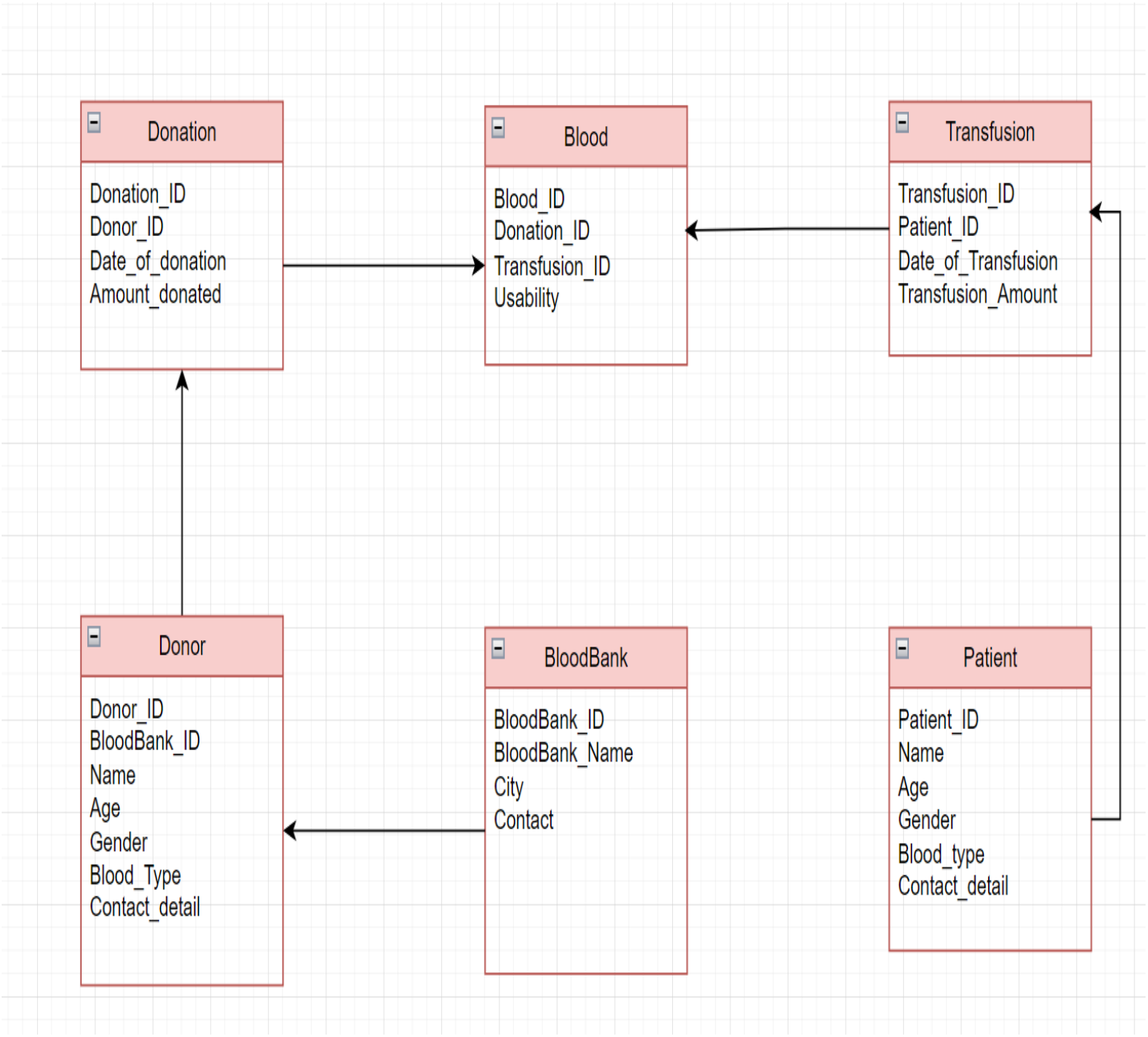
Donor_ID:Name,Age,Gender,Blood_Type,Contact_Detail

BloodBank_ID:BloodBank_Name,city,Contact

Donation_ID:Date_of_Donation,Amount_Donated

Transfusion_ID:Date_Of_Transfusion,Transfusion_Amount

Logic Modeling



Database implantation

1. Tables creation

- Blood bank

```
create table BloodBank(
    BloodBank_ID Number(10),
    BloodBank_Name Varchar2(20),
    City Varchar2(20),
    Contact Number(15),
    constraint BloodBank_ID Primary Key(BloodBank_ID)
    )
```

- Donor

```
create table Donor(
    Donor_ID Number(10) ,
    Name Varchar(20),
    Age Number(3),
    Gender Varchar2(20),
    Blood_Type Varchar2(5),
    Contact_Details Varchar(20),
    BloodBank_ID Number(10),
    constraint Donor_ID Primary Key(Donor_ID),
    Foreign Key(BloodBank_ID) References BloodBank(BloodBank_ID)
)
```

- Donation

```
create table Donation(
    Donation_ID Number(10) ,
    Date_Of_Donation Date,
    Amount_Donated Number(2),
    Donor_ID Number(10),

    constraint Donation_ID Primary Key(Donation_ID),
    Foreign Key(Donor_ID) References Donor(Donor_ID)
)
```

- Patient

```
create table Patient(
    Patient_ID Number(10) ,
    Name Varchar2(20),
    Age Number(3),
    Gender Varchar2(10),
    Blood_Type Varchar2(3),
    Contact_Details Varchar2(20),

    constraint Patient_ID Primary Key(Patient_ID)

)

```

- Transfusion

```
create table  Transfusion(

    Transfusion_ID Number(10) ,
    Date_Of_Transfusion Date,
    Transfusion_Amount Number(2),
    Patient_ID Number(10),

    constraint Transfusion_ID Primary Key(Transfusion_ID),
    Foreign Key(Patient_ID) References Patient(Patient_ID)
)
```

- Blood

```
create table  Blood(

   Blood_ID Number(10) ,
   Usability Varchar2(10),
   Donation_ID Number(10),
   Transfusion_ID Number(10),


   constraint Blood_ID Primary Key(Blood_ID),
   Foreign Key(Donation_ID) References Donation(Donation_ID),
   Foreign Key(Transfusion_ID) References Transfusion(Transfusion_ID)
)
```

2. Rows insertion

- Blood Bank

```sql
insert into BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)
Values(1337,'Lifeline ','Evergreen City','5874639210');

insert into BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)
Values(1442,'Crimson Flow ','Silver Ridge','5928374610');

insert into BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)
Values(9821,'VitalGift ','Oceanview','5639148275');

insert into BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)
Values(7631,'Hemocare ','Mount Vista','5486972310');

insert into BloodBank(BloodBank_ID,BloodBank_Name,City,Contact)
Values(5432,'PulsePoint','Starlight Town','5036781429');

```

- Donor

```sql
insert into Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID)
Values(301,'Abass almasri',25,'M','O-','5031962847',1337);

insert into Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID)
Values(302,'Ahmad fahad',37,'M','A-','5098124367',1442);

insert into Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID)
Values(303,'Layla hana',28,'F','AB-','5217436980',9821);

insert into Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID)
Values(304,'Fatima Salma',34,'F','B+','5372946180',7631);

insert into Donor(Donor_ID,Name,Age,Gender,Blood_Type,Contact_Details,BloodBank_ID)
Values(305,'Lina Yasmin',25,'F','O+','5291748360',5432);

```

- Donation

```
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1000 , '24-FEBRUARY-29' , 4 , 301);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1001 , '24-MARCH-22' , 2 , 302);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1002 , '24-MARCH-23' , 1 , 303);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1003 , '24-APRIL-3' , 2 , 304);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1004 , '24-MAY-2' , 4 , 301);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1005 , '24-MAY-27' , 2 , 305);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
values(1006 , '24-MAY-27' , 3 , 306);
insert into Donation(Donation_ID , Date_of_donation , Amount_donated , Donor_ID)
```

- Patient

```
insert into Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)
Values(173,'Omar alhalabi',23,'M','O+','583025649');

insert into Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)
Values(174,'Yamen Alom',26,'M','O-','5396024581');

insert into Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)
Values(175,'Aous Yasser',22,'M','A+','5068394251');

insert into Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)
Values(176,'Omar khaled',22,'M','AB+','5089642351');

insert into Patient(Patient_ID,Name,Age,Gender,Blood_Type,Contact_Details)
Values(177,'lara malki',30,'F','B+','5073428569');

```

- Transfusion

```
insert into Transfusion(Transfusion_ID , Date_of_transfusion ,  Transfusion_amount , Patient_ID)
values(100 , '24-MARCH-23' , 4 , 173);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(101 , '24-MARCH-23' , 2 , 174);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(102 , '24-MARCH-25' , 1 , 176);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(103 , '24-APRIL-3' , 2 , 175);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(104 , '24-MAY-22' , 4 , 177);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(105 , '24-MAY-27' , 2 , 178);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
values(106 , '24-MAY-27' , 3 , 173);
insert into Transfusion(Transfusion_ID , Date_of_transfusion , Transfusion_amount , Patient_ID)
```

- Blood

```
insert into Blood(Blood_ID , Usability , Donation_ID , Transfusion_ID)
values(1 , 'N' , 1000 , 105);

insert into Blood(Blood_ID , Usability , Donation_ID , Transfusion_ID)
values(2 , 'N' , 1001 , 105);

insert into Blood(Blood_ID , Usability , Donation_ID , Transfusion_ID)
values(3 , 'Y' , 1002 , NULL);

insert into Blood(Blood_ID , Usability , Donation_ID , Transfusion_ID)
values(4 , 'Y' , 1002 , NULL);

insert into Blood(Blood_ID , Usability , Donation_ID , Transfusion_ID)
values(5 , 'N' , 1007 , 107);

```

3. **Queries**

- Aggregate

```
-- Count of donors
SELECT COUNT(Donor_ID) AS Number_of_Donors FROM Donor;

-- Average donation amount
SELECT AVG(Amount_Donated) AS Average_Donation FROM Donation;
```

NUMBER_OF_DONORS
6

Download CSV

AVERAGE_DONATION
2.375

- Order by

```
-- Order patients by age in ascending order
SELECT Name, Age FROM Patient ORDER BY Age ASC;
```

NAME	AGE
Aous Yasser	22
Omar khaled	22
Omar alhalabi	23
Yamen Alom	26
sarah ahmed	28
lara malki	30

Download CSV

6 rows selected.

- Where

```
-- Blood banks in Evergreen City
select BloodBank_ID,BloodBank_Name from BloodBank where city='Evergreen City';
```

BLOODBANK_ID	BLOODBANK_NAME
1337	Lifeline

Download CSV

- Group by

```
-- Count of donors by gender
SELECT COUNT(Donor_ID), Gender FROM Donor GROUP BY Gender;
```

COUNT(DONOR_ID)	GENDER
3	M
3	F

Download CSV

2 rows selected.

- Join query

```
-- Joining tables to get information about blood donation
SELECT Donor.Blood_Type, Donation.Amount_Donated, Blood.Usability
FROM Donation
JOIN Donor ON Donation.Donor_ID = Donor.Donor_ID
JOIN Blood ON Donation.Donation_ID = Blood.Donation_ID
ORDER BY Donor.Donor_ID;
```

BLOOD_TYPE	AMOUNT_DONATED	USABILITY
O-	4	N
A-	2	N
AB-	1	Y
AB-	1	Y
AB-	1	N
B+	2	N

Download CSV

6 rows selected.

- Subquery

```
1  -- Transfusions with amounts greater than average
2  SELECT Transfusion_ID, Date_of_transfusion, Transfusion_amount
3  FROM Transfusion
4  WHERE Transfusion_amount > (SELECT AVG(Transfusion_amount) FROM Transfusion);
5
```

TRANSFUSION_ID	DATE_OF_TRANSFUSION	TRANSFUSION_AMOUNT
100	24-MAR-23	4
104	24-MAY-22	4
106	24-MAY-27	3

Download CSV

3 rows selected.

4. Procedures

- Find blood type donors

```
CREATE OR REPLACE PROCEDURE Find_BloodType_Donors(required_BloodType IN VARCHAR2)
AS
BEGIN
    FOR Donor_temp IN (SELECT Donor.Donor_ID, Donor.Name, Donor.Blood_Type
                       FROM Donor
                       WHERE Donor.Blood_Type = required_BloodType)
    LOOP
        DBMS_OUTPUT.PUT_LINE('Donor_ID: ' || Donor_temp.Donor_ID || '   Name: '
    || Donor_temp.Name || '   BloodType: ' || Donor_temp.Blood_Type);
    END LOOP;
END Find_BloodType_Donors;
/
EXECUTE Find_BloodType_Donors('O+');
```

```
Procedure created.

Statement processed.
Donor_ID: 305   Name: Lina Yasmin   BloodType: O+
```

- Blood Usability

```
CREATE OR REPLACE PROCEDURE BloodUsability(
    p_UpdatedUsability Blood.Usability%TYPE,
    p_Blood_ID         Blood.Blood_ID%TYPE,
    p_Transfusion_ID Transfusion.Transfusion_ID%TYPE)
AS
BEGIN

    UPDATE Blood
    SET Usability = p_UpdatedUsability
    WHERE Blood_ID = p_Blood_ID;


    UPDATE Blood
    SET Transfusion_ID = p_Transfusion_ID
    WHERE Blood_ID = p_Blood_ID;

    COMMIT;
END BloodUsability;
/

execute Bloodusability('Y',3,NULL);
```

```
Procedure created.

Statement processed.
```