Final Project Report

Modern Database Systems and Applications

Name: **Md Ashraful Islam**

Student ID.: **149103**

Name of Project: **Blood Donation System Database Project**.

Introduction

Database management is at the core of any organization. Blood Donation system are one of the organizations that need to keep track of their data. And to keep track of the data, I have created a small database that allows the user of the database to keep track of many things revolving around the blood donation system.

This is a system where the particulars of the patient, blood bank, blood bags, data of the donor, nurse, blood bank manager will be saved and will be interrelated with each other.

In this project, I will be providing a description of the database. It will help us understand that requirement from which we can start to design our database. In this description, all the requirements about each of the entities will be recorded. The description needs to provide us with enough information that it can be used to design a database.

Afterwards a conceptual E/R diagram was created from the description. It showed all the entities, attributes, and the relation between them. Then relational design was created which was used to implement the database in MySQL.

There are 6 tables created in MySQL. After the E/R diagram drawn, the relational Figure 2 and the implementation of the table can be seen from

Figure 3 to Figure 16. In the end, I have created some queries that help get a better understanding of how queries work and how to implement them.

The database should be designed with this description:

- Data of Patient: Name of the patient, ID, Contact number.
- Data of Donor: Name and ID of Donor, DOB, Contact number.
- Data of Nurse: Name and ID of Nurse, Contact Number.
- Blood Bank Manager: Name, Employee ID., Contact Number
- Data of Blood: Blood No., Quantity, Donation type, Blood type.
- Data of Blood Bank: Blood Bank ID., Name of the blood bank, Contact number, Address.

Conceptual Design:

All the requirement above for the description was used to design the conceptual design and only the notation given during the lecture are used to design this ER Diagram. Figure 1 shows the E/R Diagram of the description above.

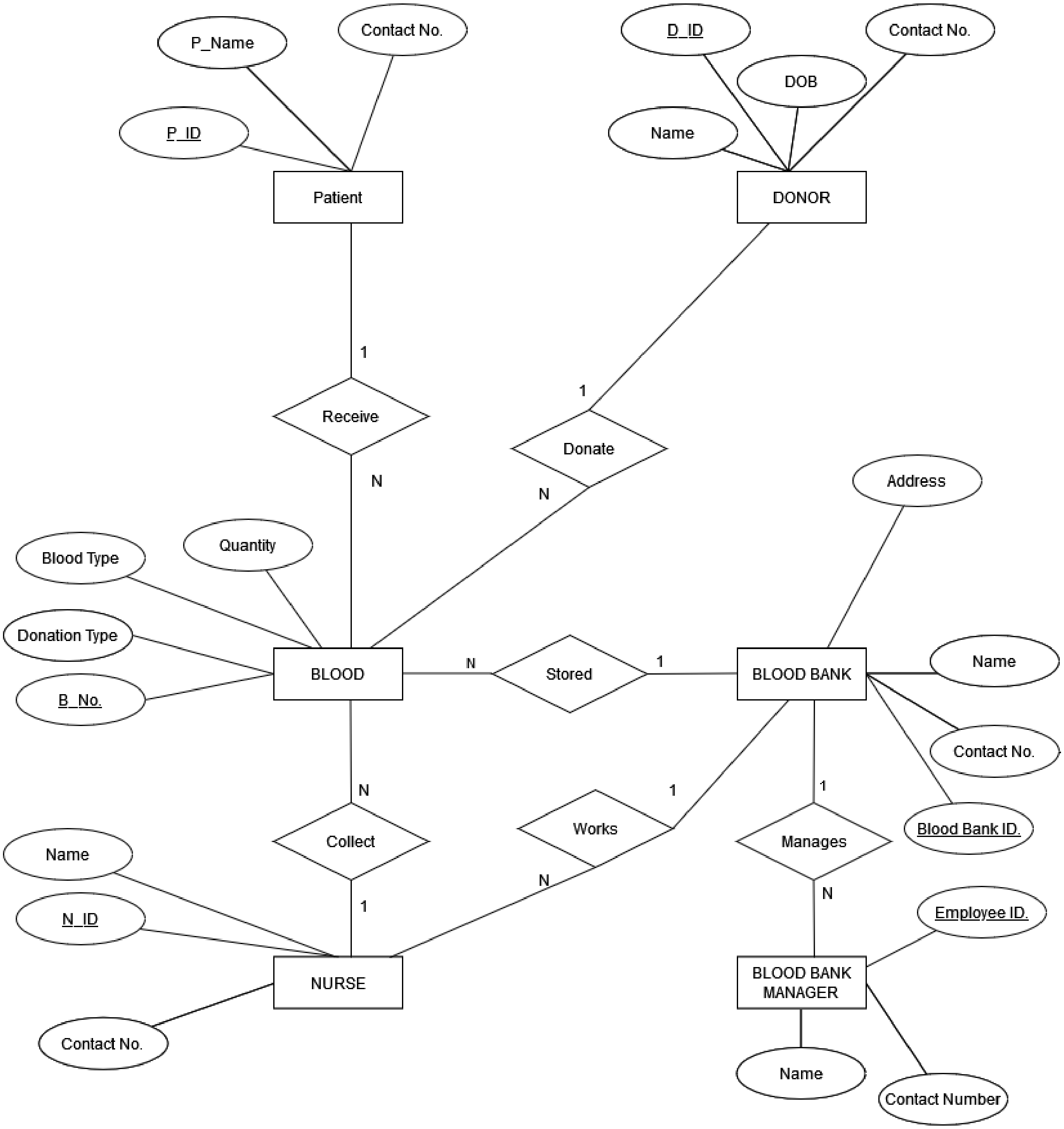


Figure 1: Conceptual design using ER diagram

Relational Design:

After the conceptual design was drawn using all the notations required, I have created the relation design as shown in figure 2.

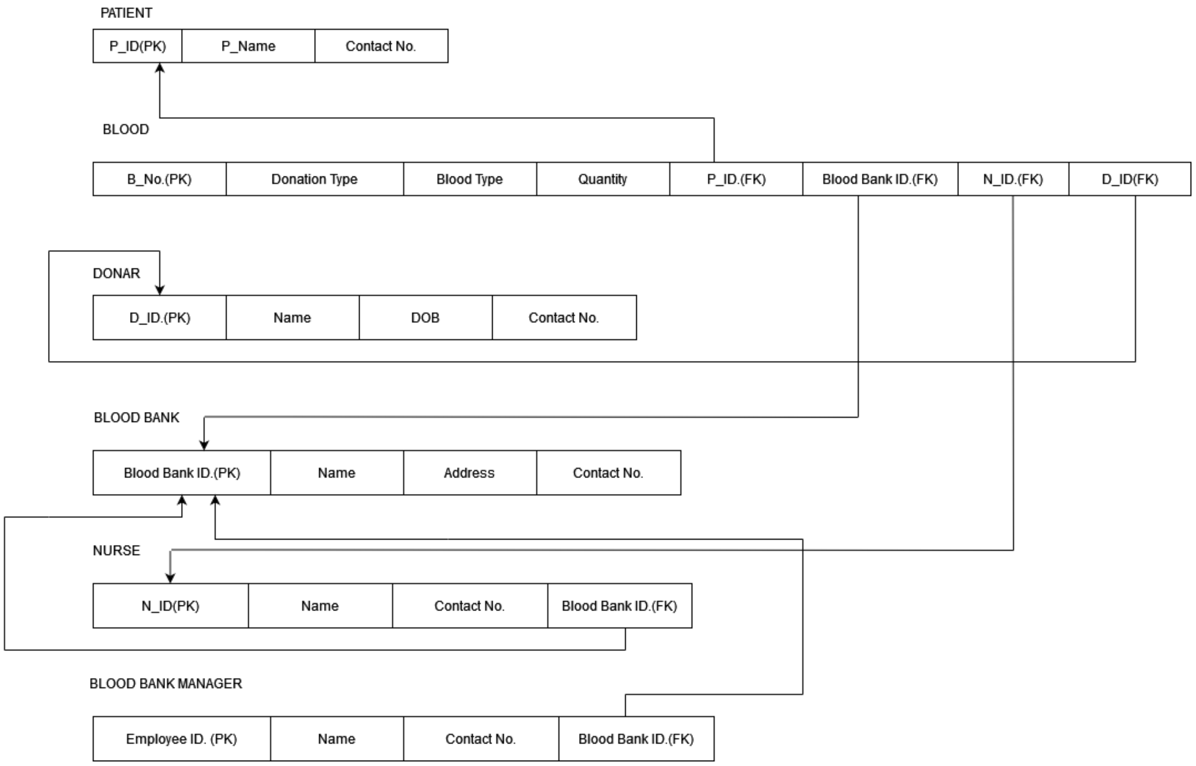
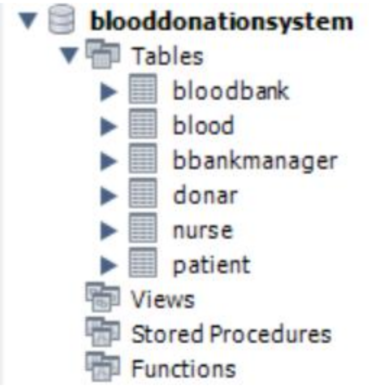


Figure 2: Relational diagram

Database Implementation:

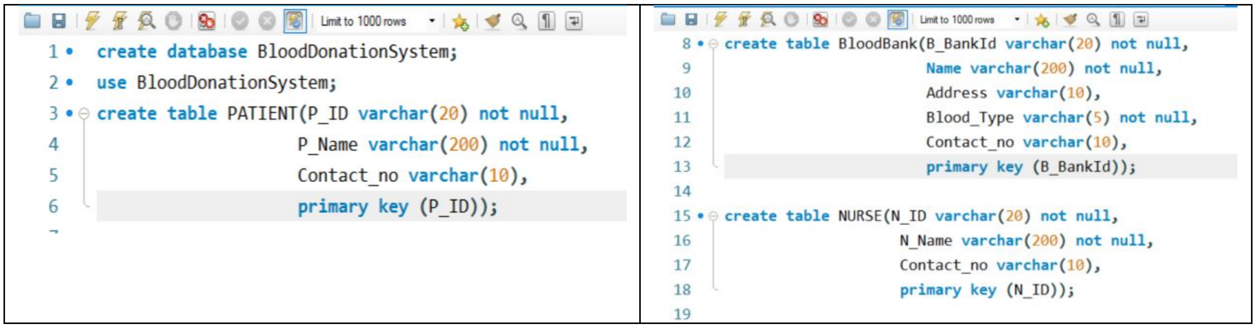
In this section, I implemented the project in MySQL. I have created six tables and each table populated with some sample data. Figure 3 shows the created database and tables in MySQL.



```
▼ blooddonationsystem
  ▼ Tables
    ▶ bloodbank
    ▶ blood
    ▶ bbankmanager
    ▶ donar
    ▶ nurse
    ▶ patient
    Views
    Stored Procedures
    Functions
```

Figure 3: Blood Donation System database and related table in MySQL

In next few figures, I added the MYSQL code that are used to create the database as well as all tables. Figure 4 shows the code for ‘Blood Donation System’ database and the code for the tables (PATIENT, BLOODBANK, NURSE). All these cases, I added necessary constrain and primary key during table creation.



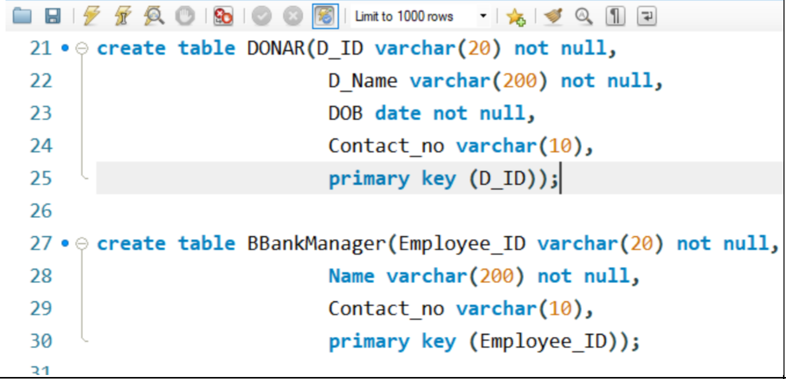
```
create database BloodDonationSystem;
use BloodDonationSystem;
create table PATIENT(P_ID varchar(20) not null,
                     P_Name varchar(200) not null,
                     Contact_no varchar(10),
                     primary key (P_ID));
```

```
create table BloodBank(B_BankId varchar(20) not null,
                       Name varchar(200) not null,
                       Address varchar(10),
                       Blood_Type varchar(5) not null,
                       Contact_no varchar(10),
                       primary key (B_BankId));

create table NURSE(N_ID varchar(20) not null,
                   N_Name varchar(200) not null,
                   Contact_no varchar(10),
                   primary key (N_ID));
```

Figure 4: Create database and tables (PATIENT, BLOODBANK, NURSE)

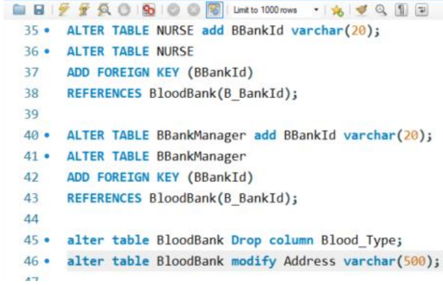
Figure 5 shows the MySQL code for the table DONAR and BBankManager. In Figure 6, I alter these two tables: added foreign key, drop a column, and change the type of the column.



```
create table DONAR(D_ID varchar(20) not null,
                         D_Name varchar(200) not null,
                         DOB date not null,
                         Contact_no varchar(10),
                         primary key (D_ID));

create table BBankManager(Employee_ID varchar(20) not null,
                         Name varchar(200) not null,
                         Contact_no varchar(10),
                         primary key (Employee_ID));
```

Figure 5: Create DONAR and BBankManager table



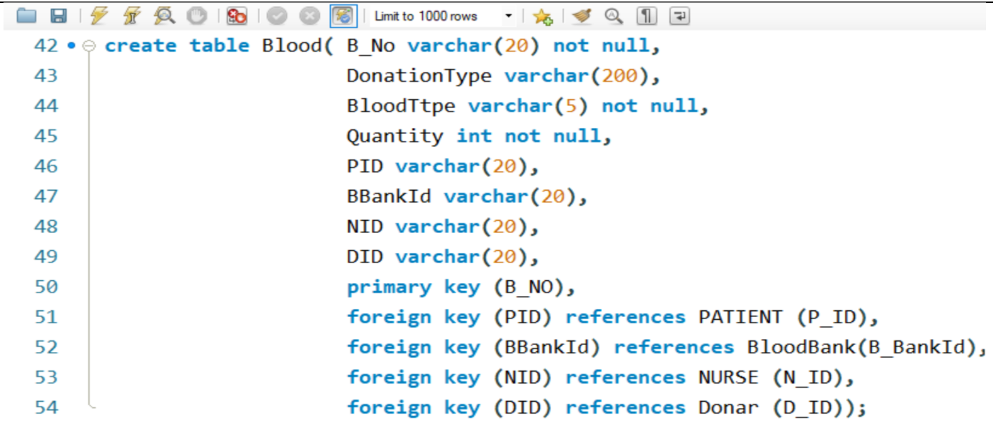
```
ALTER TABLE NURSE add BBankId varchar(20);
ALTER TABLE NURSE
ADD FOREIGN KEY (BBankId)
REFERENCES BloodBank(B_BankId);

ALTER TABLE BBankManager add BBankId varchar(20);
ALTER TABLE BBankManager
ADD FOREIGN KEY (BBankId)
REFERENCES BloodBank(B_BankId);

alter table BloodBank Drop column Blood_Type;
alter table BloodBank modify Address varchar(500);
```

Figure 6: Alter few tables and did necessary change

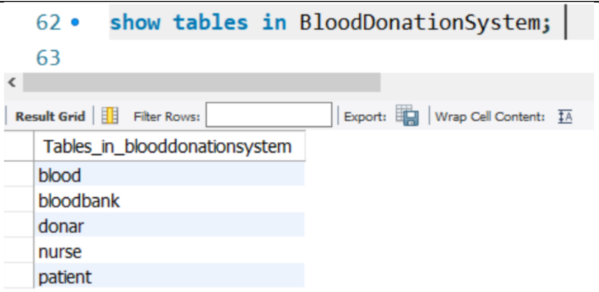
Figure 7 shows the MySQL code of Blood table. At this stage I added all necessary foreign keys during table creation to maintain the connection with other tables.



```
create table Blood( B_No varchar(20) not null,
                    DonationType varchar(200),
                    BloodTtpe varchar(5) not null,
                    Quantity int not null,
                    PID varchar(20),
                    BBankId varchar(20),
                    NID varchar(20),
                    DID varchar(20),
                    primary key (B_NO),
                    foreign key (PID) references PATIENT (P_ID),
                    foreign key (BBankId) references BloodBank(B_BankId),
                    foreign key (NID) references NURSE (N_ID),
                    foreign key (DID) references Donar (D_ID));
```

Figure 7: Create BLOOD table along with all foreign key

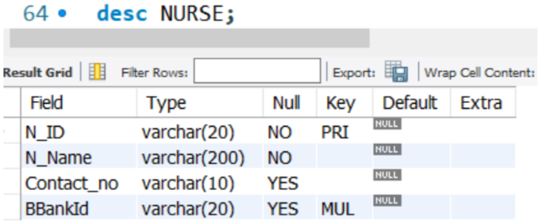
Figure 8 shows the list of tables in the ‘Blood Donation System’ database and Figure shows the structure of NURSE table. We can use the same code to see the structure of other tables.



```
show tables in BloodDonationSystem;
```

Tables_in_blooddonationsystem
blood
bloodbank
donar
nurse
patient

Figure 8: See all table in the database

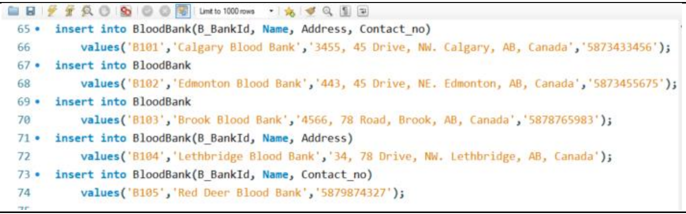


```
desc NURSE;
```

Field	Type	Null	Key	Default	Extra
N_ID	varchar(20)	NO	PRI	NULL	
N_Name	varchar(200)	NO		NULL	
Contact_no	varchar(10)	YES		NULL	
BBankId	varchar(20)	YES	MUL	NULL	

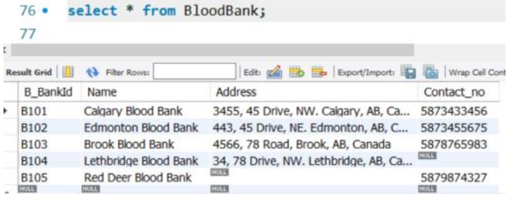
Figure 9: See the structure of NURSE tables

Figure 10 has a list of MySQL code to insert the data in Blood Bank table and Figure 11 shows the inserted data in the Blood Bank table. I followed two different types of syntax to insert the data in the table.



```
insert into BloodBank(B_BankId, Name, Address, Contact_no)
    values('B101','Calgary Blood Bank','3455, 45 Drive, NW. Calgary, AB, Canada','5873433456');
insert into BloodBank
    values('B102','Edmonton Blood Bank','443, 45 Drive, NE. Edmonton, AB, Canada','5873455675');
insert into BloodBank
    values('B103','Brook Blood Bank','4566, 78 Road, Brook, AB, Canada','5878765983');
insert into BloodBank(B_BankId, Name, Address)
    values('B104','Lethbridge Blood Bank','34, 78 Drive, NW. Lethbridge, AB, Canada');
insert into BloodBank(B_BankId, Name, Contact_no)
    values('B105','Red Deer Blood Bank','5879874327');
```

Figure 10: Insert data in Blood Bank Table

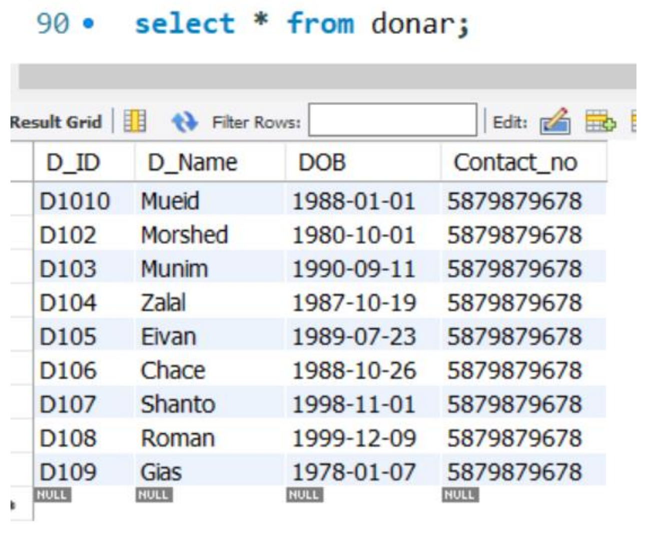


```
select * from BloodBank;
```

B_BankId	Name	Address	Contact_no
B101	Calgary Blood Bank	3455, 45 Drive, NW. Calgary, AB, Ca...	5873433456
B102	Edmonton Blood Bank	443, 45 Drive, NE. Edmonton, AB, C...	5873455675
B103	Brook Blood Bank	4566, 78 Road, Brook, AB, Canada	5878765983
B104	Lethbridge Blood Bank	34, 78 Drive, NW. Lethbridge, AB, Ca...	NULL
B105	Red Deer Blood Bank	NULL	5879874327
NULL	NULL	NULL	NULL

Figure 11: Show data in Blood Bank Table

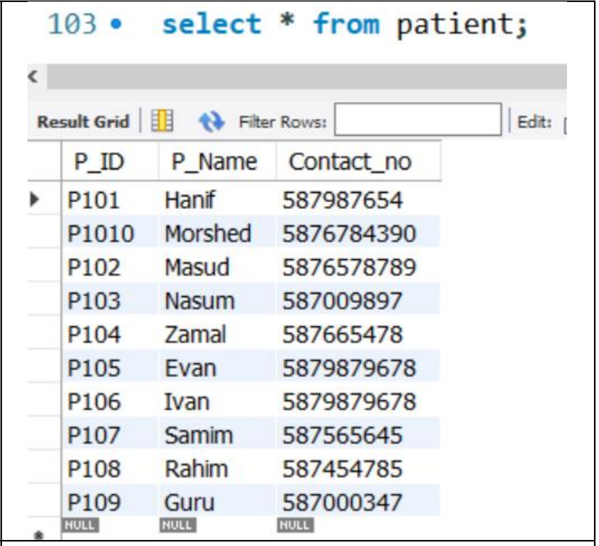
Figure 12 -16 shows the data in other tables. During insert the data in table we satisfy attribute constrain as well as foreign key constrains.



```
90 •    select * from donar;
```

D_ID	D_Name	DOB	Contact_no
D1010	Mueid	1988-01-01	5879879678
D102	Morshed	1980-10-01	5879879678
D103	Munim	1990-09-11	5879879678
D104	Zalal	1987-10-19	5879879678
D105	Eivan	1989-07-23	5879879678
D106	Chace	1988-10-26	5879879678
D107	Shanto	1998-11-01	5879879678
D108	Roman	1999-12-09	5879879678
D109	Gias	1978-01-07	5879879678
NULL	NULL	NULL	NULL

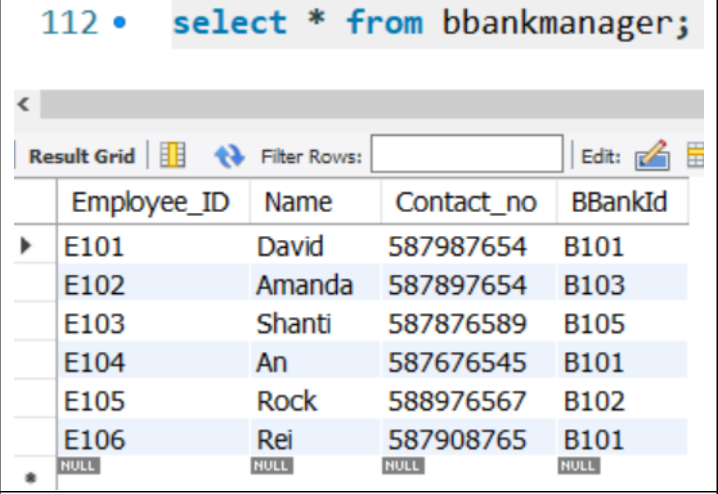
Figure 12: Data in Donor table



```
103 •    select * from patient;
```

P_ID	P_Name	Contact_no
P101	Hanif	587987654
P1010	Morshed	5876784390
P102	Masud	5876578789
P103	Nasum	587009897
P104	Zamal	587665478
P105	Evan	5879879678
P106	Ivan	5879879678
P107	Samim	587565645
P108	Rahim	587454785
P109	Guru	587000347
NULL	NULL	NULL

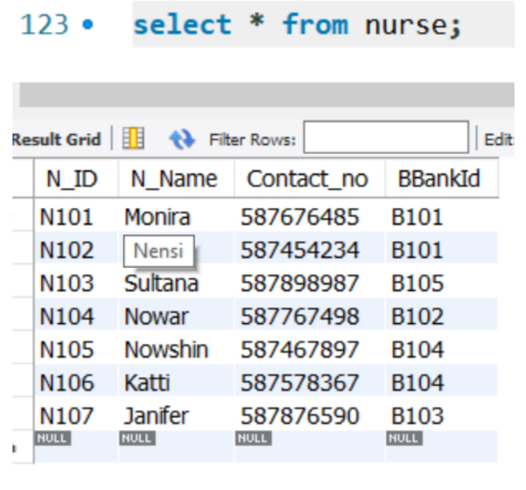
Figure 13: Data in Patient table



```
112 •    select * from bbankmanager;
```

Employee_ID	Name	Contact_no	BBankId
E101	David	587987654	B101
E102	Amanda	587897654	B103
E103	Shanti	587876589	B105
E104	An	587676545	B101
E105	Rock	588976567	B102
E106	Rei	587908765	B101
NULL	NULL	NULL	NULL

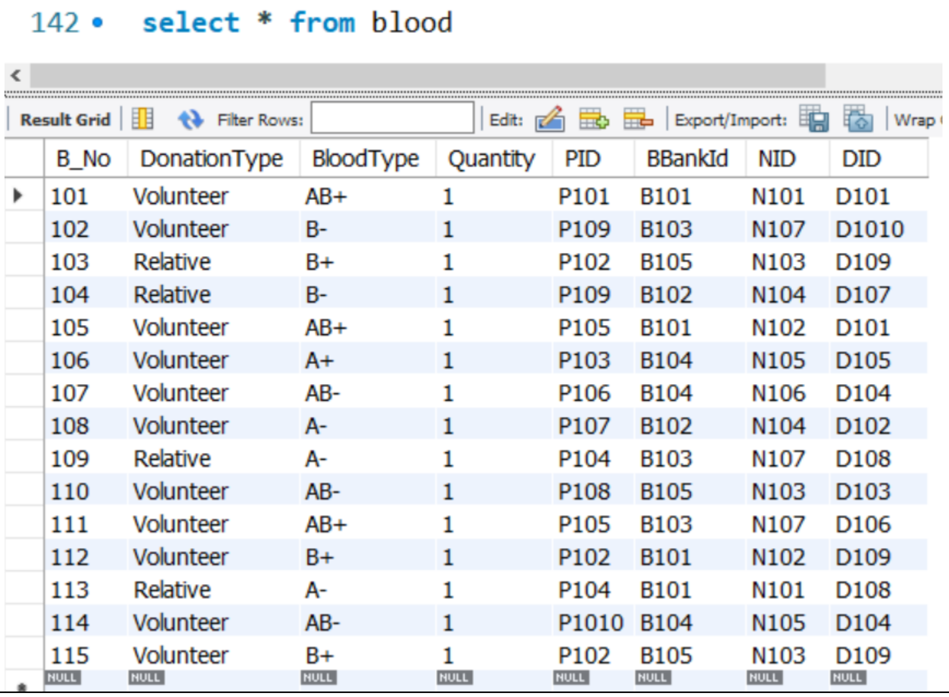
Figure 14: Data in BBankManager table



```
123 •    select * from nurse;
```

N_ID	N_Name	Contact_no	BBankId
N101	Monira	587676485	B101
N102	Nensi	587454234	B101
N103	Sultana	587898987	B105
N104	Nowar	587767498	B102
N105	Nowshin	587467897	B104
N106	Katti	587578367	B104
N107	Janifer	587876590	B103
NULL	NULL	NULL	NULL

Figure 15: Data in Nurse table



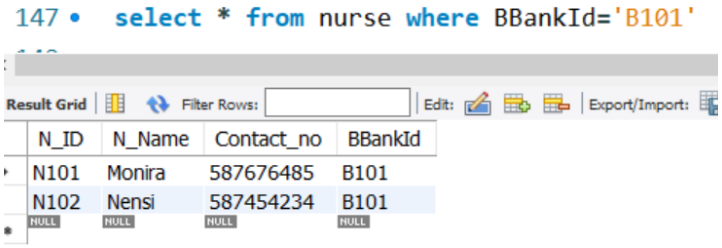
```
select * from blood
```

B_No	DonationType	BloodType	Quantity	PID	BBankId	NID	DID
101	Volunteer	AB+	1	P101	B101	N101	D101
102	Volunteer	B-	1	P109	B103	N107	D1010
103	Relative	B+	1	P102	B105	N103	D109
104	Relative	B-	1	P109	B102	N104	D107
105	Volunteer	AB+	1	P105	B101	N102	D101
106	Volunteer	A+	1	P103	B104	N105	D105
107	Volunteer	AB-	1	P106	B104	N106	D104
108	Volunteer	A-	1	P107	B102	N104	D102
109	Relative	A-	1	P104	B103	N107	D108
110	Volunteer	AB-	1	P108	B105	N103	D103
111	Volunteer	AB+	1	P105	B103	N107	D106
112	Volunteer	B+	1	P102	B101	N102	D109
113	Relative	A-	1	P104	B101	N101	D108
114	Volunteer	AB-	1	P1010	B104	N105	D104
115	Volunteer	B+	1	P102	B105	N103	D109
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 16: Data in Blood table

In this table, I added a list of queries based on the data of the given tables as well as their relation.

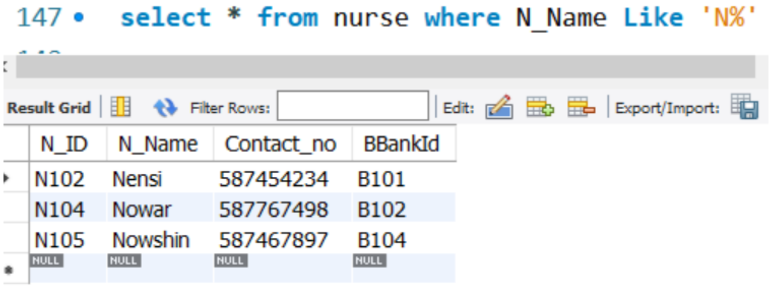
- ❖ Find a list of nurses who work in the Blood Bank with id is ‘B101’



```
select * from nurse where BBankId='B101'
```

N_ID	N_Name	Contact_no	BBankId
N101	Monira	587676485	B101
N102	Nensi	587454234	B101
NULL	NULL	NULL	NULL

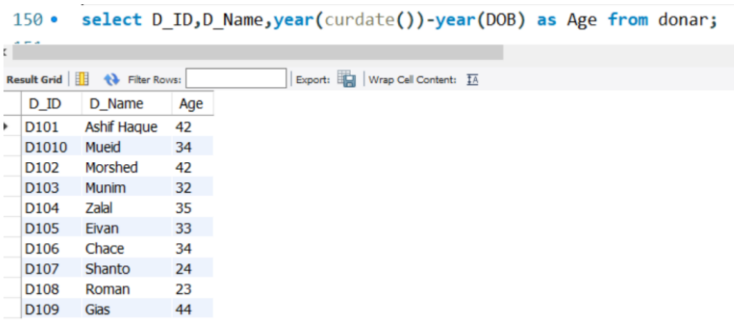
- ❖ Find a list of nurses whose name start with alphabet ‘M’.



```
select * from nurse where N_Name Like 'N%'
```

N_ID	N_Name	Contact_no	BBankId
N102	Nensi	587454234	B101
N104	Nowar	587767498	B102
N105	Nowshin	587467897	B104
NULL	NULL	NULL	NULL

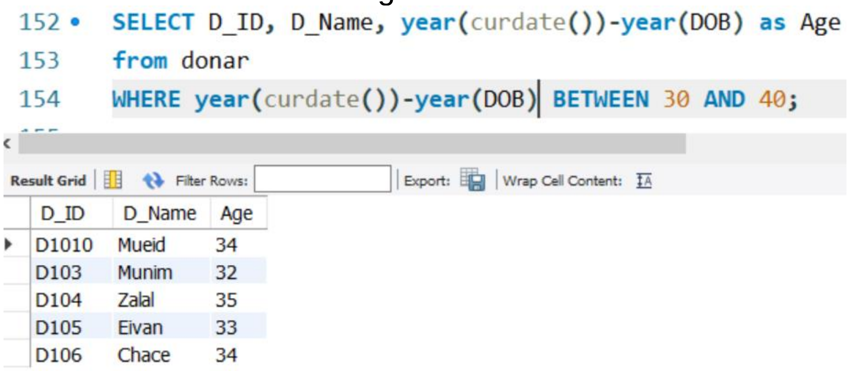
- ❖ Find the age of the donor from their DOB column.



```sql
select D_ID,D_Name,year(curdate())-year(DOB) as Age from donar;
```

D_ID	D_Name	Age
D101	Ashif Haque	42
D1010	Mueid	34
D102	Morshed	42
D103	Munim	32
D104	Zalal	35
D105	Eivan	33
D106	Chace	34
D107	Shanto	24
D108	Roman	23
D109	Gias	44

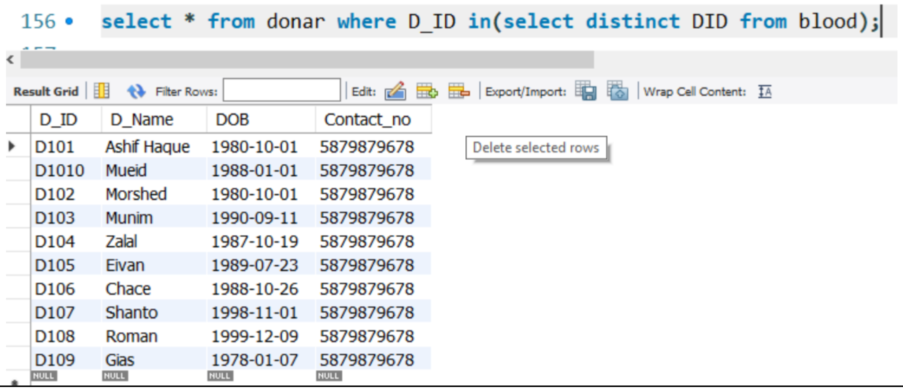
- ❖ Find the name of the donor whose age between 30 to 40.



```sql
SELECT D_ID, D_Name, year(curdate())-year(DOB) as Age
from donar
WHERE year(curdate())-year(DOB) BETWEEN 30 AND 40;
```

D_ID	D_Name	Age
D1010	Mueid	34
D103	Munim	32
D104	Zalal	35
D105	Eivan	33
D106	Chace	34

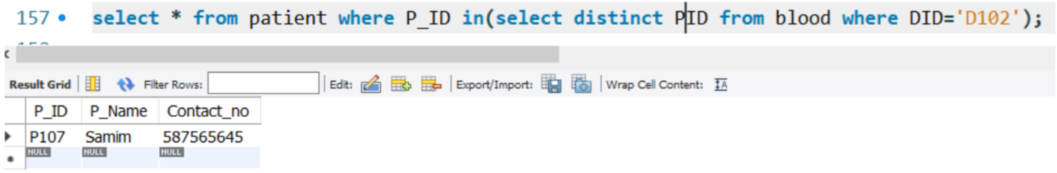
- ❖ Find the name of the donor who donated the blood.



```sql
select * from donar where D_ID in(select distinct DID from blood);
```

D_ID	D_Name	DOB	Contact_no
D101	Ashif Haque	1980-10-01	5879879678
D1010	Mueid	1988-01-01	5879879678
D102	Morshed	1980-10-01	5879879678
D103	Munim	1990-09-11	5879879678
D104	Zalal	1987-10-19	5879879678
D105	Eivan	1989-07-23	5879879678
D106	Chace	1988-10-26	5879879678
D107	Shanto	1998-11-01	5879879678
D108	Roman	1999-12-09	5879879678
D109	Gias	1978-01-07	5879879678
NULL	NULL	NULL	NULL

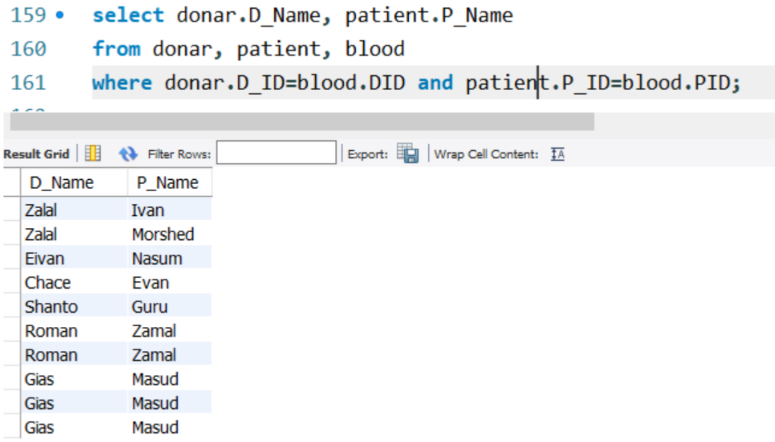
- ❖ Find the name of the patient who received blood from donor ‘D102’



```sql
select * from patient where P_ID in(select distinct PID from blood where DID='D102');
```

P_ID	P_Name	Contact_no
P107	Samim	587565645
NULL	NULL	NULL

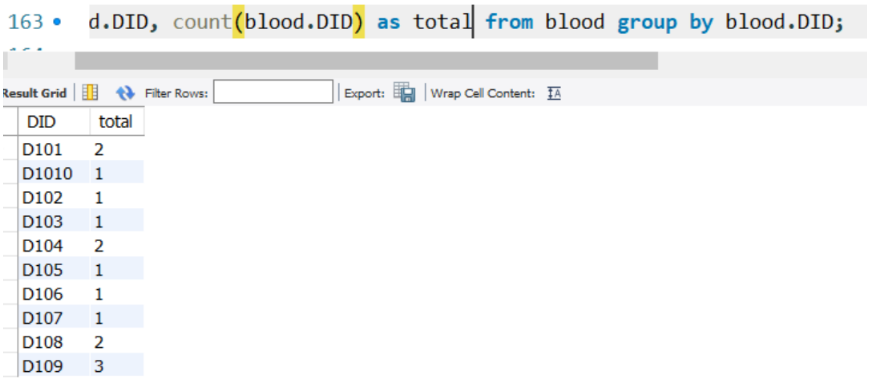
- ❖ Find the donor and their blood receiver name (accept repetition).



```sql
select donar.D_Name, patient.P_Name
from donar, patient, blood
where donar.D_ID=blood.DID and patient.P_ID=blood.PID;
```

D_Name	P_Name
Zalal	Ivan
Zalal	Morshed
Eivan	Nasum
Chace	Evan
Shanto	Guru
Roman	Zamal
Roman	Zamal
Gias	Masud
Gias	Masud
Gias	Masud

❖ Count the number of times a donor donates the blood.



```sql
d.DID, count(blood.DID) as total from blood group by blood.DID;
```

DID	total
D101	2
D1010	1
D102	1
D103	1
D104	2
D105	1
D106	1
D107	1
D108	2
D109	3

Summary

The project was about creating a database for a blood donation system. For this project, I was required to perform five steps. These steps include: 1) having a description of the blood donation system database. 2)Creating a conceptual E/R diagram. 3) Creating a relational diagram 4)Implementing the relational design in MySQL and populating the tables in MySQL. 5)Performing some different queries to get results.

This project gives us a better understanding of how databases work. The different steps required in implementing the database. The description to create the database provides us with the necessary information required in the database. It gives us a rough idea of what needs to be stored and how are entities linked to other entities. The project above it gave us the attributes of each of the entities. It provided us with unique identifiers for each entity. The conceptual entity relation diagram provides us with a visual representation of the description. For the project above, we can see that the conceptual entity relation diagram makes it clear to identify the entities and attributes of each of the entities. It also helps in identifying the primary keys for each entity. It illustrates the relationship between the entities if they are many to many, one to one, or one to many.

The relational design gave us our 6 tables which were later implemented in the MySQL database. Implementation of the design in MySQL was the easiest part since all the information needed to implement it was already there. To implement the design, I created a database, and in which I created the 6 tables. After creating the table, I populated each table with some data.

Finally, I wrote some queries to test. Other queries were also used to get certain information. In the end, the queries and their result were created and run successfully.