A Practical activity Report submitted

for Database Management System (UCS310)

by

Neelakshi Gupta 102103766

Kanishka 102103772

2COE27

Submitted to:

Ms. Archana Singh

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

INDIA

Jan-May 2023

TABLE OF CONTENT

TOPIC	**PAGE NO.**
INTRODUCTION	3
INFO ABOUT ENTITIES	4
ER DIAGRAM	5
RELATION BW ENTITIES	6
ER TO TABLES BEFORE NORMALIZATION	7
ER TO TABLES AFTER NORMALIZATION	9
RELATIONAL SCHEMAS AFTER NORMALIZATION	11
QUERIES	19
CONCLUSION	27
REFERENCES	27

INTRODUCTION

OBJECTIVE: BLOOD BANK MANAGEMENT SYSTEM

ABSTRACT

Blood donation safety is a relevant and important public health issue. This database project allows hospitals to make inventories of their blood online, subsequently allowing each hospital to check the availability of blood anytime possible. With this project threats on improper blood donor documentation or misplaced records will be totally removed. Our system aims in providing desired blood in emergency from either blood bank or direct donors.

PROJECT OVERVIEW

Blood Bank Management System is a database system to link between the donors and blood banks and act as an interface for the patient to find his/her desired blood in a fast and efficient way. Transfusion of blood is a complex organization requiring careful management and design. Essential requirements can be stated as Donor requirement, blood collection, testing of donor blood, component preparation and supply.

Blood Banks collect, store and provide collected blood to the patients from donors. The banks then group the blood which they receive according to their blood groups. Along with grouping, they need to check contamination of blood.

This database automates the distribution of blood. The entire project has been developed keeping in view of the distributed client server architecture with centralized storage of database in mind.

Using the constructs of MySQL Server database has been designed, normalized for better results.

BACKGROUND OF STUDY

For hospitals, a blood bank known as blood collection centre, also is an area in which collected blood bags are stored and preserved for future use in blood transfusion services. Most blood banks are still running manual system in its processes. As such, there is a lack of efficiency because it is still paper-based in collecting information about donors, inventories of blood bags, and blood transfusion services. The lack of proper documentation may endanger patients due to possibility of having contaminate blood bags.

Hence, a web-based blood bank management system is in high need.

INFORMATION OF ENTITIES

PROJECT ENTITIES:

1. **Donor**:
   (Attributes: bd_ID,bd_name,bd_age,bd_sex,bd_Bgroup,bd_reg_date,bd_phNo)
   Donor is the person who donates blood, on donation a donor id(bd_ID) is generated and used as primary key to identify the donor information. Name, age, sex, phone number and registration dates will be stored in database under Donor entity.

2. **Recipient**:

   (Attributes-reci_ID,reci_name,reci_age,reci_Bgrp,reci_Bqnty,reci_sex,reci_reg_date,reci_phNo)
   Recipient is the person who receives blood from blood bank, when blood ig given recipient ID(reci_ID) is generated which is used as primary key for the recipient entity to identify blood recipients information.

3. **Manager**:
   (Attributes: M_id,mName,m_phNo)
   Manager takes care of the available blood samples in the blood bank, also responsible for handling blood requests from recipients and hospitals. Blood Manager has a unique identification number (M_id) used as a primary key along with name and number of manager stored under 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 entity has reco_ID which is primary key along with recorder's name along with phone number stored under this entity.

5. **BloodSpecimen**:
   (Attributes- specimen_number, b_group, status)
   Under this entity, we will store the information of blood samples which are available in the blood bank. Here, specimen_number and b_group together will be primary key along with status attribute which shows whether blood is contaminated or not.

6. **DiseaseFinder**:
   (Attributes- dfind_ID, dfind_name,dfind_PhNo)
   Under this entity, we will store the information of the doctor who checks the blood foor any kind of contaminations. We have a unique identification number(dfind_ID) as primary key along with other attributes.

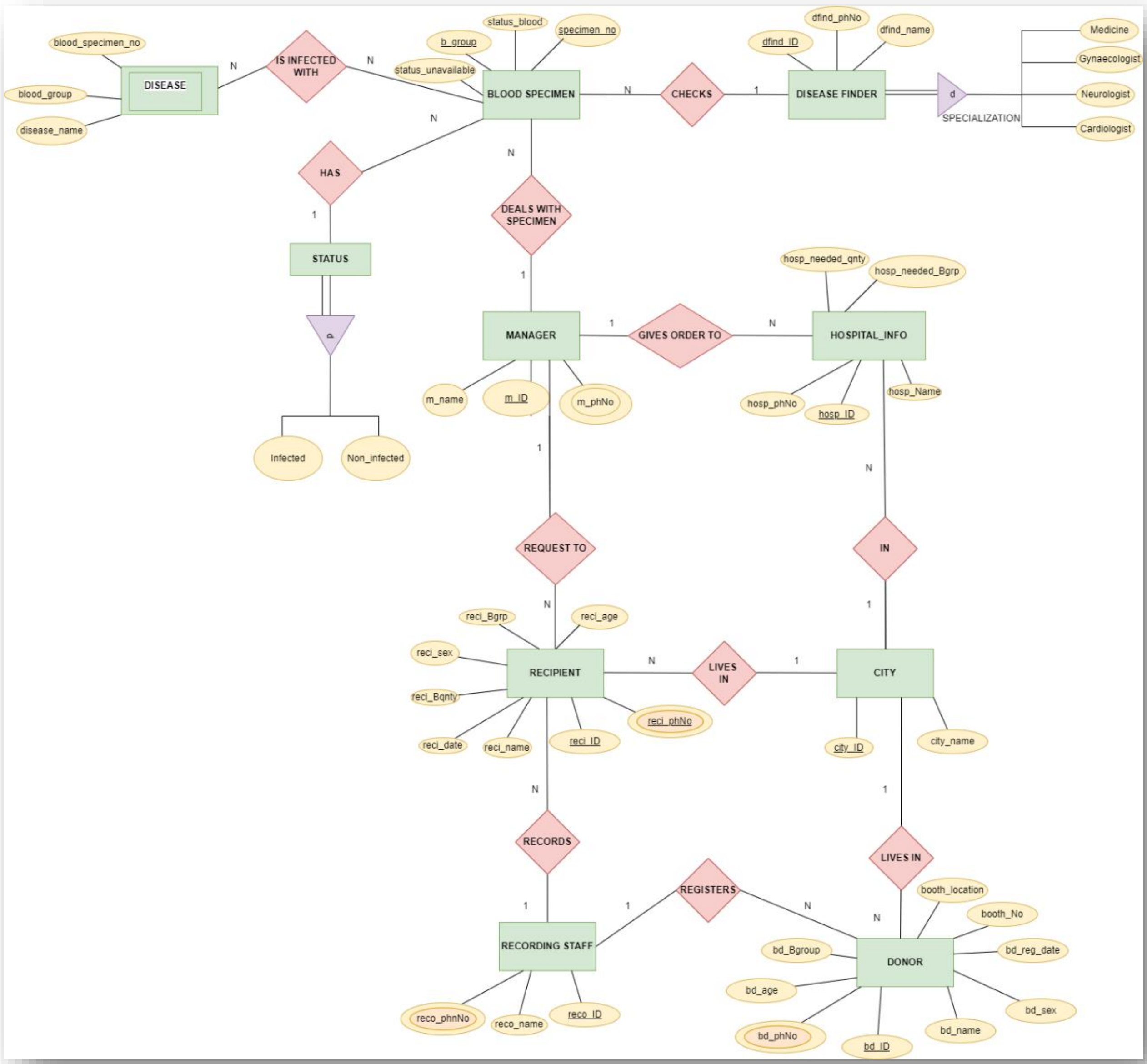
7. **Hospital_Info**:
   (Attributes- hosp_ID,hosp_name,hosp_needed_Bgrp,hosp_needed_Bqnty)

Here under this entity we will store the information of hospitals. In this hosp_ID and hosp_needed_Bgrp together makes the primary key.

8. **City**:
   (Attributes- city_ID, city_name)
   This entity stores the information where donors,recipients and hospitals are present. City_id is used as primary key to identify the information about the city.

9. **Disease**:
   (Attributes: blood_specimen,bgroup,disease_name)
   This entity shows the disease present in a blood specimen.

ER-DIAGRAM



blood_specimen_no
blood_group
disease_name
DISEASE
IS INFECTED WITH
N
N
status_blood
b_group
specimen_no
status_unavailable
BLOOD SPECIMEN
N
CHECKS
1
dfind_phNo
dfind_ID
dfind_name
DISEASE FINDER
d
SPECIALIZATION
Medicine
Gynaecologist
Neurologist
Cardiologist
N
HAS
1
STATUS
d
Infected
Non_infected
N
DEALS WITH SPECIMEN
1
MANAGER
1
GIVES ORDER TO
N
HOSPITAL_INFO
hosp_needed_qnty
hosp_needed_Bgrp
m_name
m_ID
m_phNo
hosp_phNo
hosp_ID
hosp_Name
1
N
REQUEST TO
IN
N
1
reci_Bgrp
reci_age
reci_sex
reci_Bqnty
RECIPIENT
N
LIVES IN
1
CITY
reci_phNo
reci_date
reci_name
reci_ID
city_ID
city_name
N
1
RECORDS
LIVES IN
1
1
REGISTERS
N
N
booth_location
booth_No
RECORDING STAFF
bd_Bgroup
DONOR
bd_reg_date
bd_age
bd_sex
reco_phnNo
reco_name
reco_ID
bd_phNo
bd_ID
bd_name

RELATIONSHIP BETWEEN ENTITES:

1. City and Hospital_Info:
   Relationship=”in”
   Type: 1 to Many

2. City and Donor:
   Relationship=” lives in”
   Type: 1 to Many

3. City and Recipient
   Relationship=”lives in”
   Type: 1 to Many

4. Recording_Staff and Donor:
   Relationship=”registers”
   Type: 1 to Many

5. Recording_Staff and Recipient:
   Relationship=”records”
   Type: 1 to Many

6. Hospital_Info and Manager
   Relationship=”gives order to”
   Type: 1 to Many

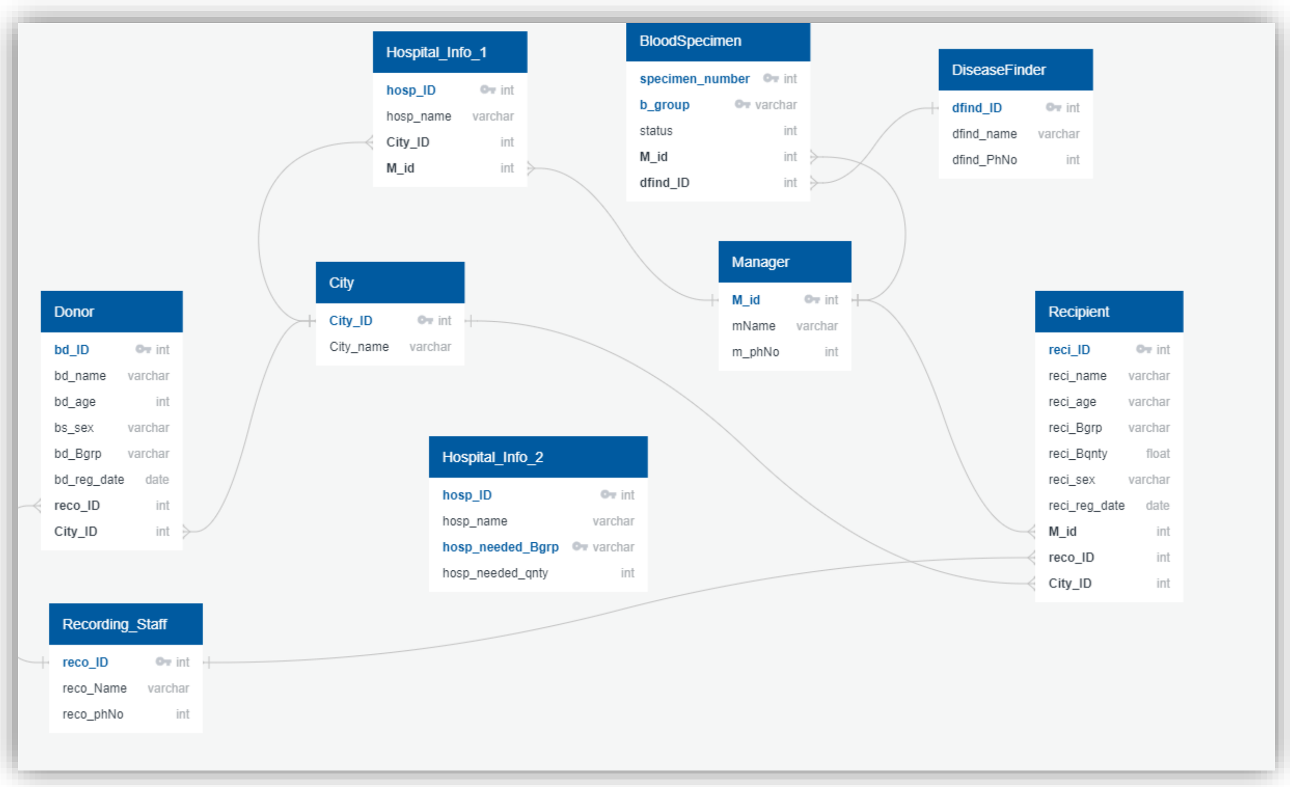
7. Manager and BloodSpecimen
   Relationship=”deals with specimen”
   Type: 1 to Many
8. Recipient and Manager
   Relationship=”requests to”
   Type: 1 to Many

9. DiseaseFinder and BloodSpecimen

Relationship=”checks”

Type: 1 to Many

ER TO TABLES BEFORE NORMALIZATION:



Hospital_Info_1	
hosp_ID	int
hosp_name	varchar
City_ID	int
M_id	int

BloodSpecimen	
specimen_number	int
b_group	varchar
status	int
M_id	int
dfind_ID	int

DiseaseFinder	
dfind_ID	int
dfind_name	varchar
dfind_PhNo	int

City	
City_ID	int
City_name	varchar

Manager	
M_id	int
mName	varchar
m_phNo	int

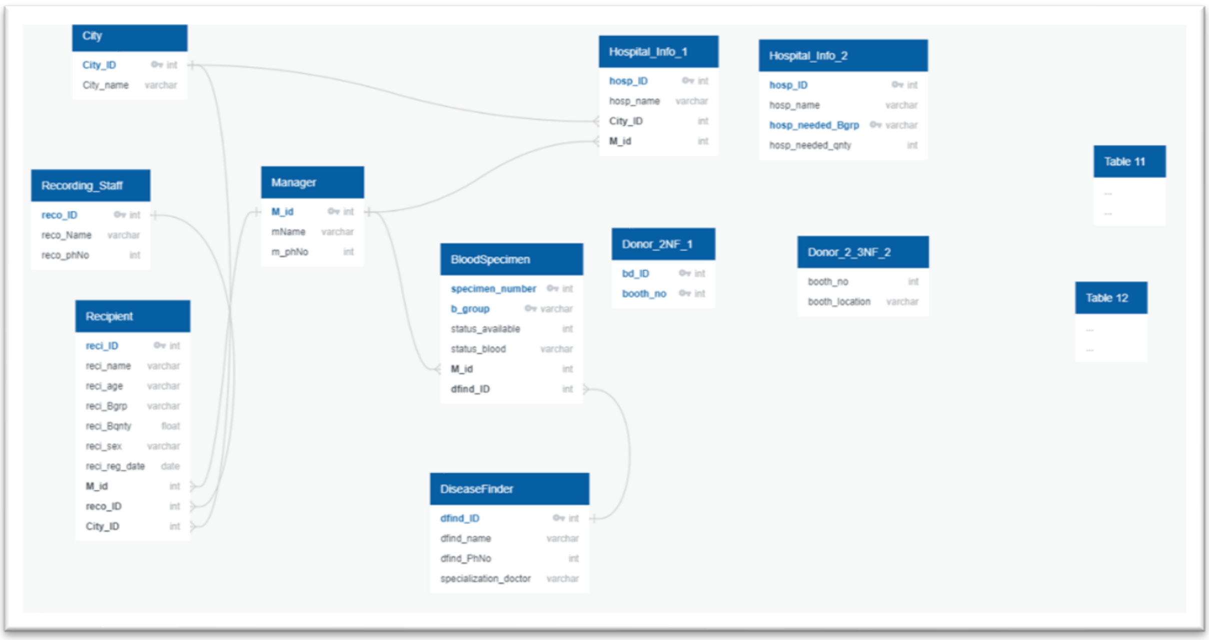
Donor	
bd_ID	int
bd_name	varchar
bd_age	int
bs_sex	varchar
bd_Bgrp	varchar
bd_reg_date	date
reco_ID	int
City_ID	int

Recipient	
reci_ID	int
reci_name	varchar
reci_age	varchar
reci_Bgrp	varchar
reci_Bqnty	float
reci_sex	varchar
reci_reg_date	date
M_id	int
reco_ID	int
City_ID	int

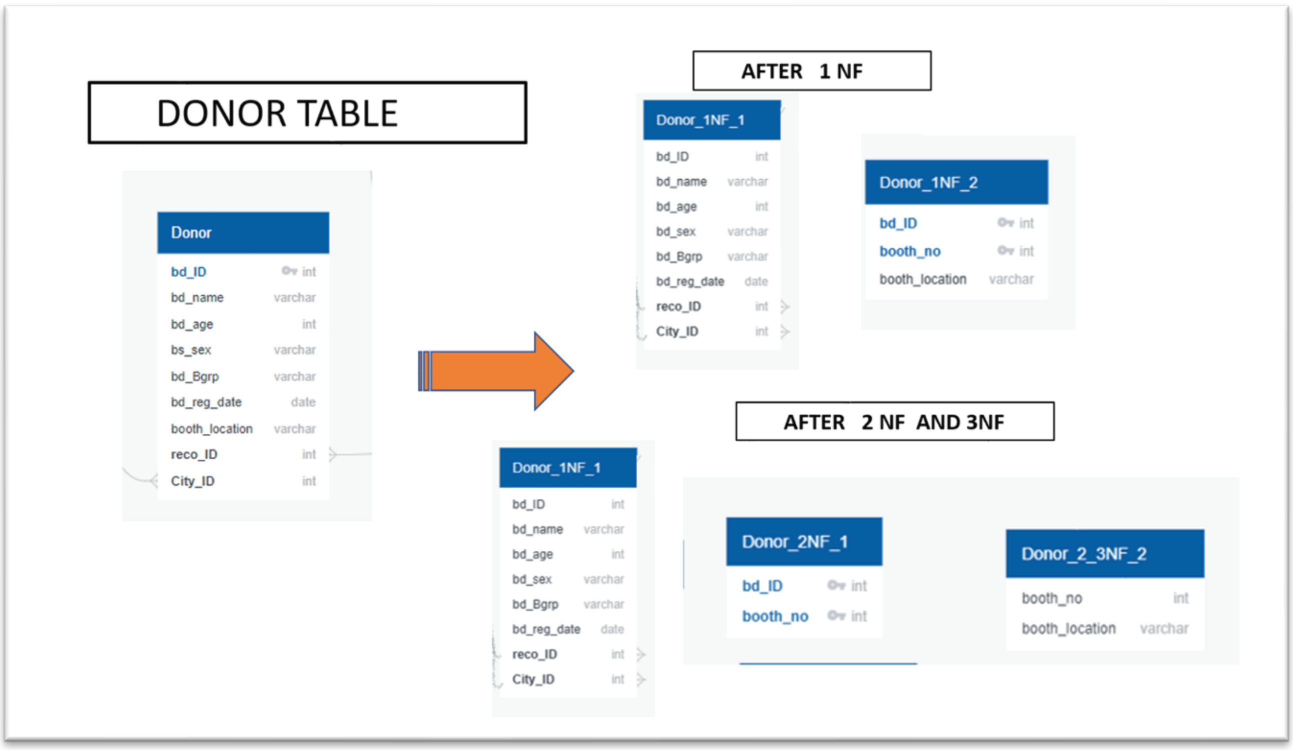
Hospital_Info_2	
hosp_ID	int
hosp_name	varchar
hosp_needed_Bgrp	varchar
hosp_needed_qnty	int

Recording_Staff	
reco_ID	int
reco_Name	varchar
reco_phNo	int

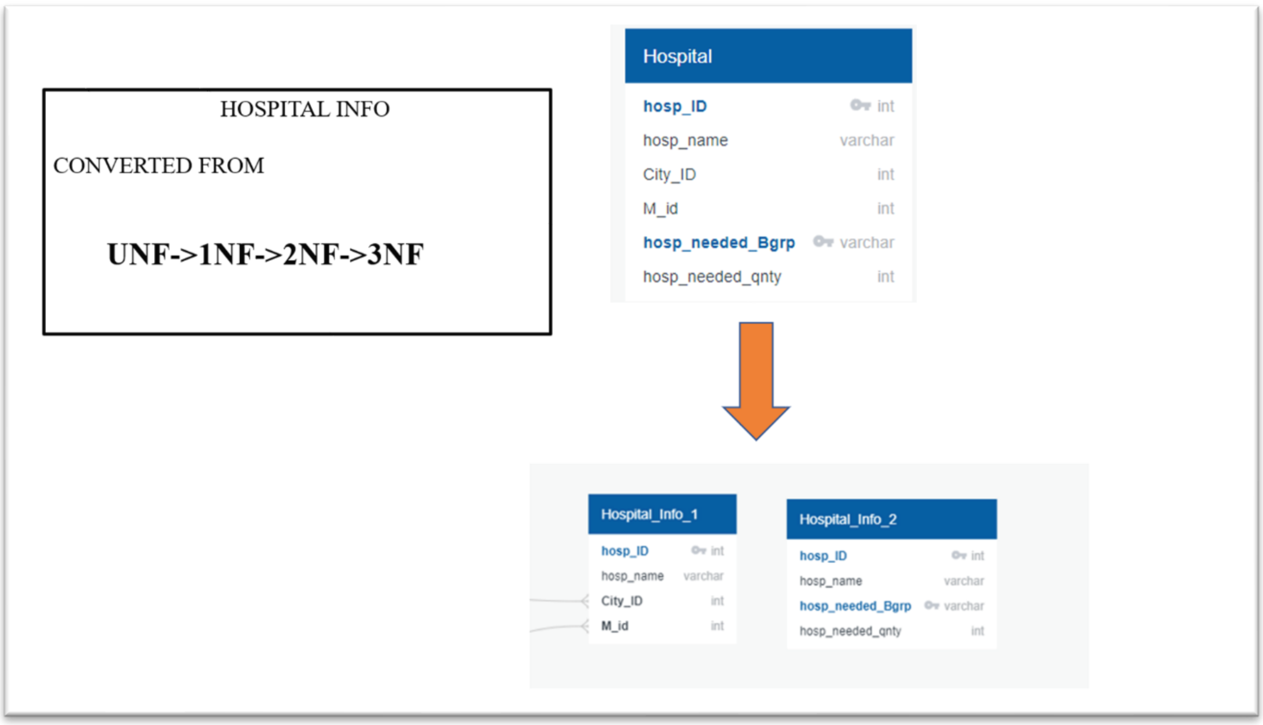
ER TO TABLES AFTER NORMALIZATION



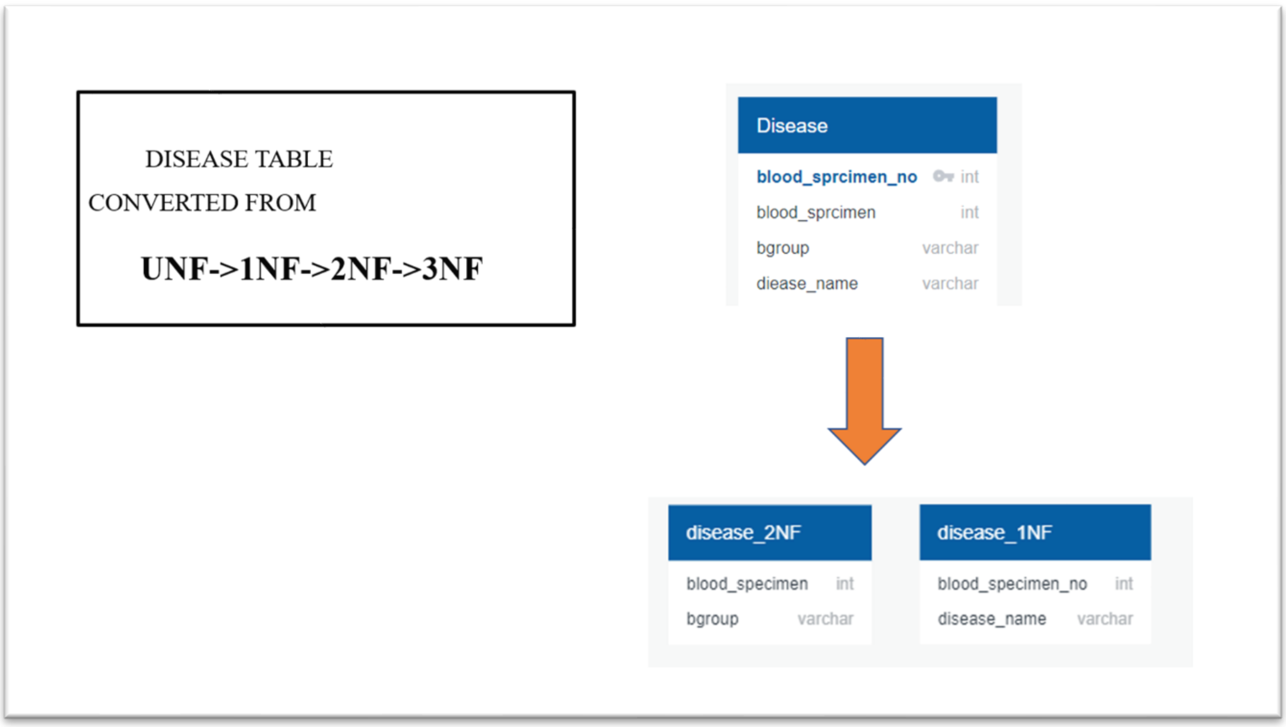
NORMALIZATION OF DONOR TABLE



NORMALIZATION OF HOSPITAL_INFO

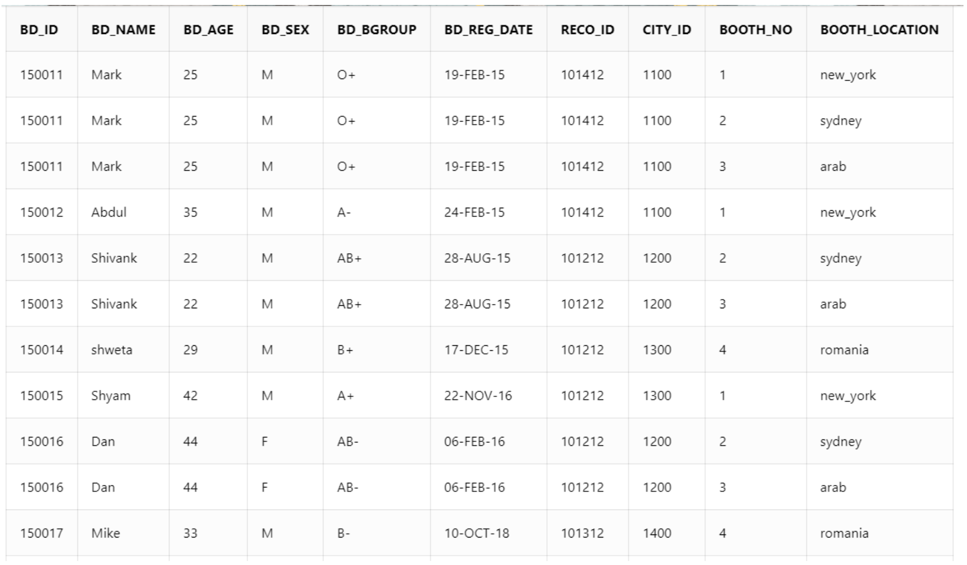


NORMALIZATION OF DISEASE



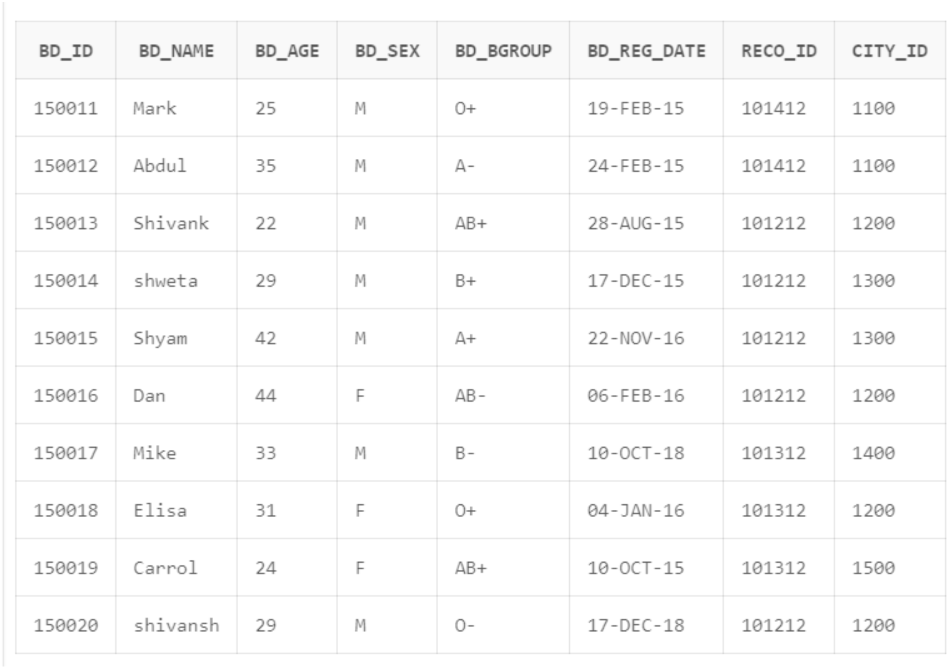
RELATIONAL SCHEMAS AFTER NORMALIZATION:

1. DONOR



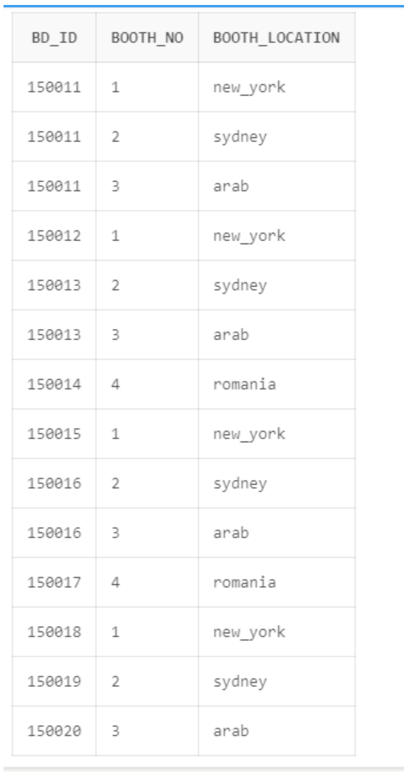
BD_ID	BD_NAME	BD_AGE	BD_SEX	BD_BGROUP	BD_REG_DATE	RECO_ID	CITY_ID	BOOTH_NO	BOOTH_LOCATION
150011	Mark	25	M	O+	19-FEB-15	101412	1100	1	new_york
150011	Mark	25	M	O+	19-FEB-15	101412	1100	2	sydney
150011	Mark	25	M	O+	19-FEB-15	101412	1100	3	arab
150012	Abdul	35	M	A-	24-FEB-15	101412	1100	1	new_york
150013	Shivank	22	M	AB+	28-AUG-15	101212	1200	2	sydney
150013	Shivank	22	M	AB+	28-AUG-15	101212	1200	3	arab
150014	shweta	29	M	B+	17-DEC-15	101212	1300	4	romania
150015	Shyam	42	M	A+	22-NOV-16	101212	1300	1	new_york
150016	Dan	44	F	AB-	06-FEB-16	101212	1200	2	sydney
150016	Dan	44	F	AB-	06-FEB-16	101212	1200	3	arab
150017	Mike	33	M	B-	10-OCT-18	101312	1400	4	romania

2. DONOR_1NF_1 TABLE:



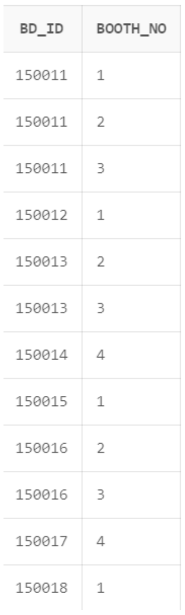
BD_ID	BD_NAME	BD_AGE	BD_SEX	BD_BGROUP	BD_REG_DATE	RECO_ID	CITY_ID
150011	Mark	25	M	O+	19-FEB-15	101412	1100
150012	Abdul	35	M	A-	24-FEB-15	101412	1100
150013	Shivank	22	M	AB+	28-AUG-15	101212	1200
150014	shweta	29	M	B+	17-DEC-15	101212	1300
150015	Shyam	42	M	A+	22-NOV-16	101212	1300
150016	Dan	44	F	AB-	06-FEB-16	101212	1200
150017	Mike	33	M	B-	10-OCT-18	101312	1400
150018	Elisa	31	F	O+	04-JAN-16	101312	1200
150019	Carrol	24	F	AB+	10-OCT-15	101312	1500
150020	shivansh	29	M	O-	17-DEC-18	101212	1200

3. DONOR_1NF_2 TABLE:



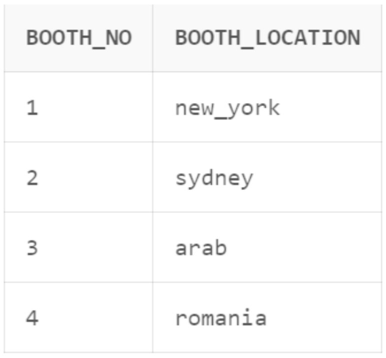
BD_ID	BOOTH_NO	BOOTH_LOCATION
150011	1	new_york
150011	2	sydney
150011	3	arab
150012	1	new_york
150013	2	sydney
150013	3	arab
150014	4	romania
150015	1	new_york
150016	2	sydney
150016	3	arab
150017	4	romania
150018	1	new_york
150019	2	sydney
150020	3	arab

4. DONOR_2NF_1



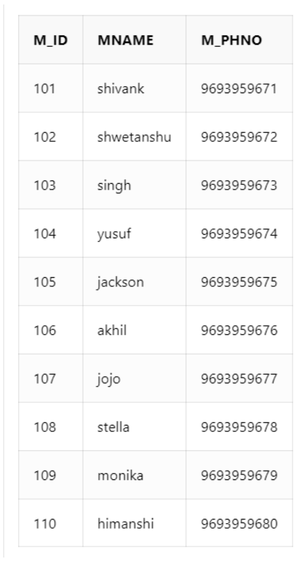
BD_ID	BOOTH_NO
150011	1
150011	2
150011	3
150012	1
150013	2
150013	3
150014	4
150015	1
150016	2
150016	3
150017	4
150018	1

5. DONOR_2NF_2:



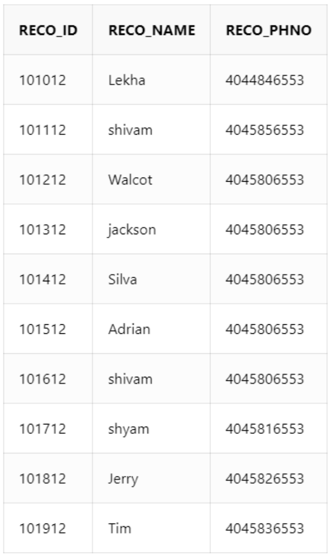
BOOTH_NO	BOOTH_LOCATION
1	new_york
2	sydney
3	arab
4	romania

6. MANAGER TABLE:



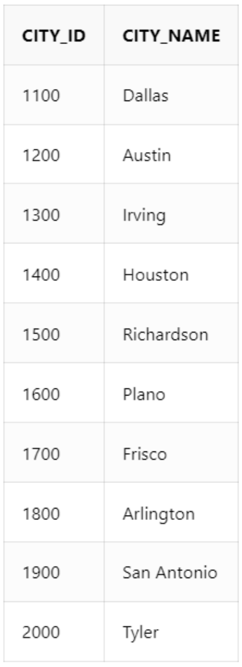
M_ID	MNAME	M_PHNO
101	shivank	9693959671
102	shwetanshu	9693959672
103	singh	9693959673
104	yusuf	9693959674
105	jackson	9693959675
106	akhil	9693959676
107	jojo	9693959677
108	stella	9693959678
109	monika	9693959679
110	himanshi	9693959680

7. RECORDING STAFF TABLE:



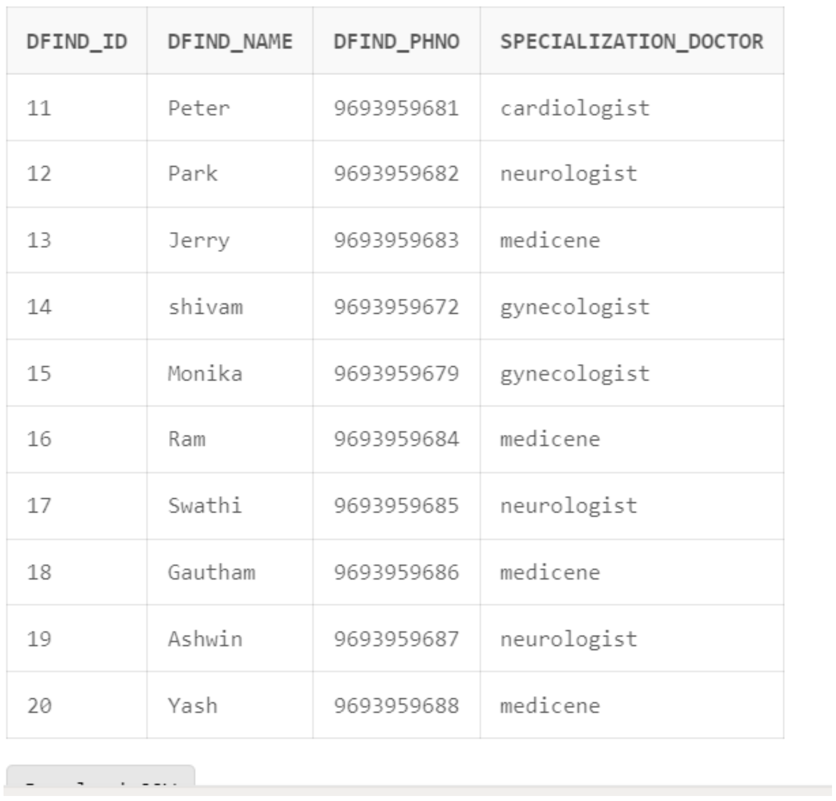
RECO_ID	RECO_NAME	RECO_PHNO
101012	Lekha	4044846553
101112	shivam	4045856553
101212	Walcot	4045806553
101312	jackson	4045806553
101412	Silva	4045806553
101512	Adrian	4045806553
101612	shivam	4045806553
101712	shyam	4045816553
101812	Jerry	4045826553
101912	Tim	4045836553

8. CITY TABLE:



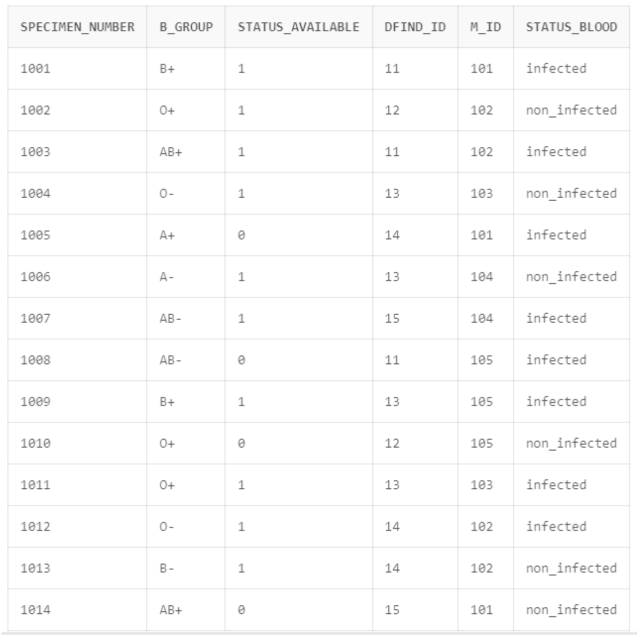
CITY_ID	CITY_NAME
1100	Dallas
1200	Austin
1300	Irving
1400	Houston
1500	Richardson
1600	Plano
1700	Frisco
1800	Arlington
1900	San Antonio
2000	Tyler

9. DISEASE FINDER:



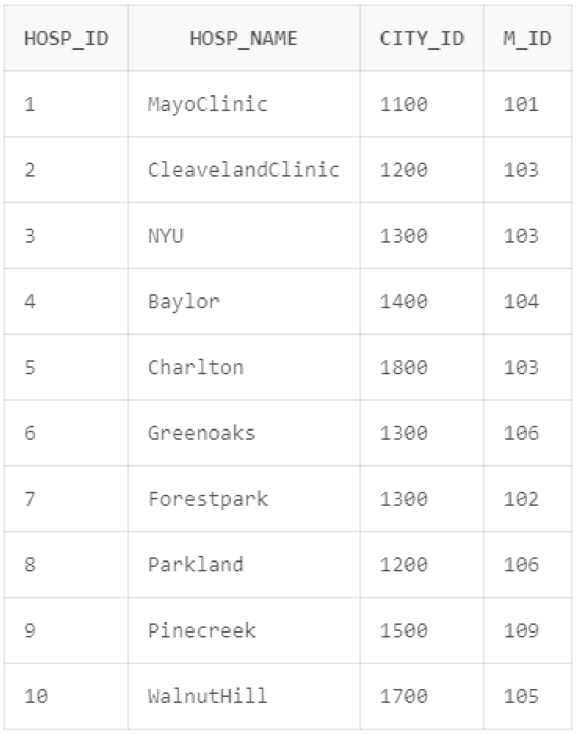
DFIND_ID	DFIND_NAME	DFIND_PHNO	SPECIALIZATION_DOCTOR
11	Peter	9693959681	cardiologist
12	Park	9693959682	neurologist
13	Jerry	9693959683	medicene
14	shivam	9693959672	gynecologist
15	Monika	9693959679	gynecologist
16	Ram	9693959684	medicene
17	Swathi	9693959685	neurologist
18	Gautham	9693959686	medicene
19	Ashwin	9693959687	neurologist
20	Yash	9693959688	medicene

10. BLOOD SPECIMEN:



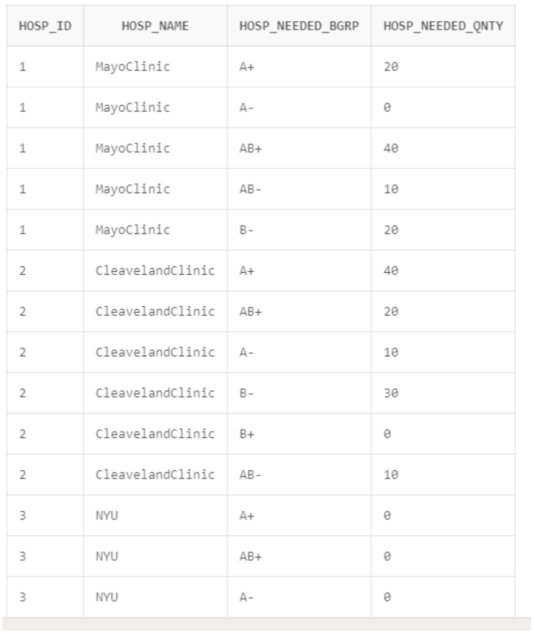
SPECIMEN_NUMBER	B_GROUP	STATUS_AVAILABLE	DFIND_ID	M_ID	STATUS_BLOOD
1001	B+	1	11	101	infected
1002	O+	1	12	102	non_infected
1003	AB+	1	11	102	infected
1004	O-	1	13	103	non_infected
1005	A+	0	14	101	infected
1006	A-	1	13	104	non_infected
1007	AB-	1	15	104	infected
1008	AB-	0	11	105	infected
1009	B+	1	13	105	infected
1010	O+	0	12	105	non_infected
1011	O+	1	13	103	infected
1012	O-	1	14	102	infected
1013	B-	1	14	102	non_infected
1014	AB+	0	15	101	non_infected

11. HOSPITAL_INFO_1:



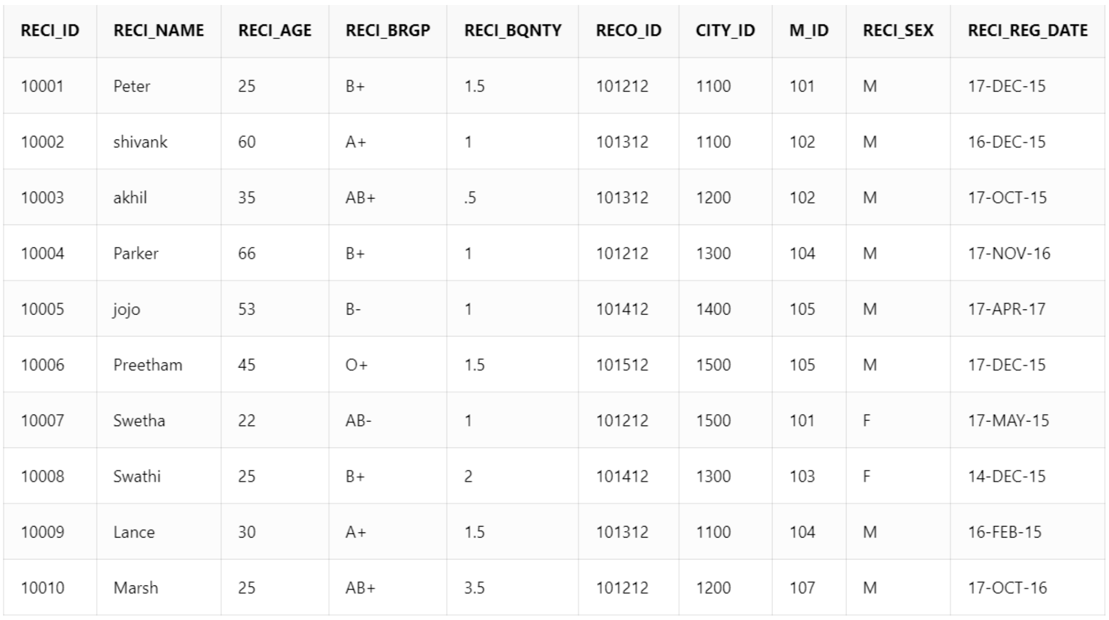
HOSP_ID	HOSP_NAME	CITY_ID	M_ID
1	MayoClinic	1100	101
2	CleavelandClinic	1200	103
3	NYU	1300	103
4	Baylor	1400	104
5	Charlton	1800	103
6	Greenoaks	1300	106
7	Forestpark	1300	102
8	Parkland	1200	106
9	Pinecreek	1500	109
10	WalnutHill	1700	105

12. HOSPITAL_INFO_2:



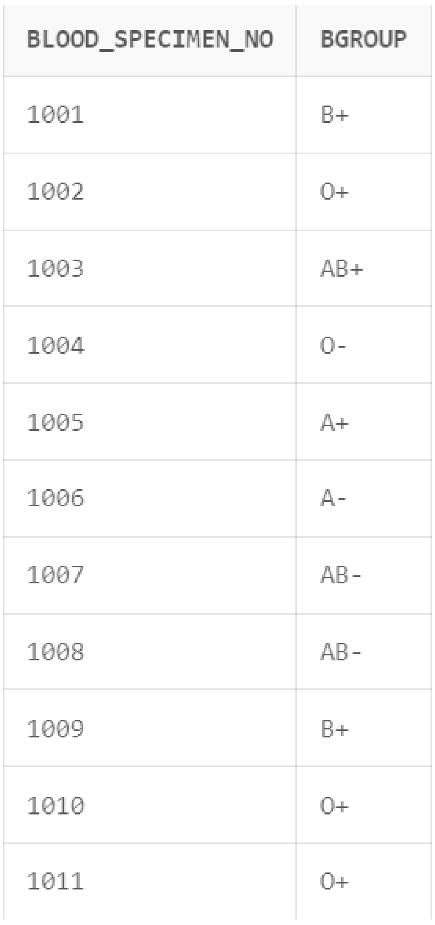
HOSP_ID	HOSP_NAME	HOSP_NEEDED_BGRP	HOSP_NEEDED_QNTY
1	MayoClinic	A+	20
1	MayoClinic	A-	0
1	MayoClinic	AB+	40
1	MayoClinic	AB-	10
1	MayoClinic	B-	20
2	CleavelandClinic	A+	40
2	CleavelandClinic	AB+	20
2	CleavelandClinic	A-	10
2	CleavelandClinic	B-	30
2	CleavelandClinic	B+	0
2	CleavelandClinic	AB-	10
3	NYU	A+	0
3	NYU	AB+	0
3	NYU	A-	0

13. RECIPIENT:



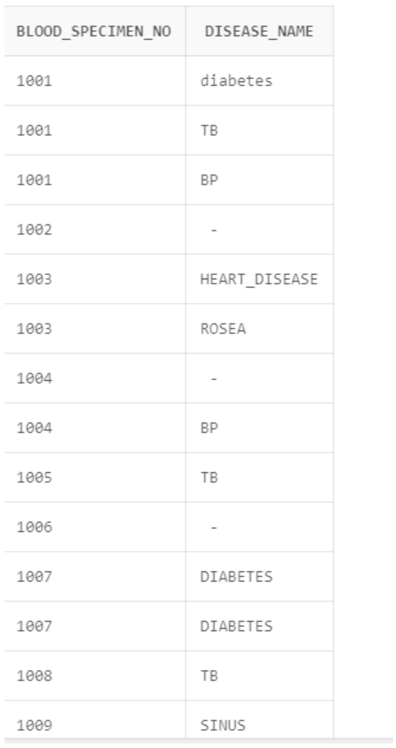
RECI_ID	RECI_NAME	RECI_AGE	RECI_BRGP	RECI_BQNTY	RECO_ID	CITY_ID	M_ID	RECI_SEX	RECI_REG_DATE
10001	Peter	25	B+	1.5	101212	1100	101	M	17-DEC-15
10002	shivank	60	A+	1	101312	1100	102	M	16-DEC-15
10003	akhil	35	AB+	.5	101312	1200	102	M	17-OCT-15
10004	Parker	66	B+	1	101212	1300	104	M	17-NOV-16
10005	jojo	53	B-	1	101412	1400	105	M	17-APR-17
10006	Preetham	45	O+	1.5	101512	1500	105	M	17-DEC-15
10007	Swetha	22	AB-	1	101212	1500	101	F	17-MAY-15
10008	Swathi	25	B+	2	101412	1300	103	F	14-DEC-15
10009	Lance	30	A+	1.5	101312	1100	104	M	16-FEB-15
10010	Marsh	25	AB+	3.5	101212	1200	107	M	17-OCT-16

14. disease_1NF TABLE:



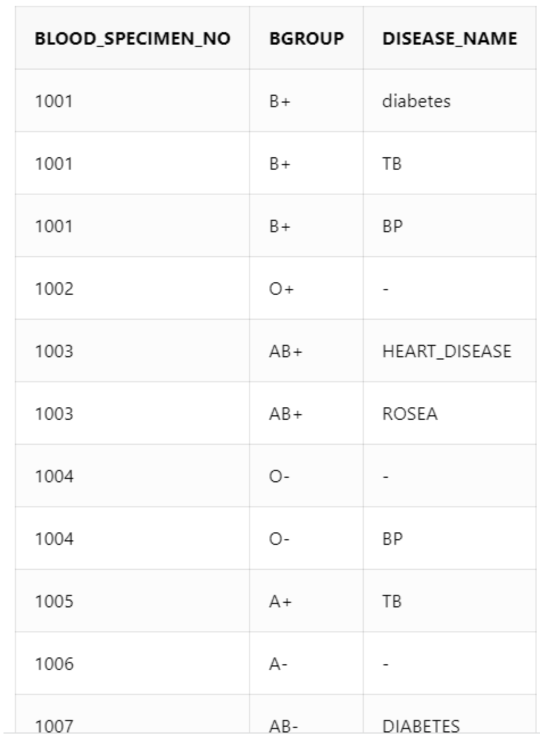
BLOOD_SPECIMEN_NO	BGROUP
1001	B+
1002	O+
1003	AB+
1004	O-
1005	A+
1006	A-
1007	AB-
1008	AB-
1009	B+
1010	O+
1011	O+

15. disease_2NF TABLE:



BLOOD_SPECIMEN_NO	DISEASE_NAME
1001	diabetes
1001	TB
1001	BP
1002	-
1003	HEART_DISEASE
1003	ROSEA
1004	-
1004	BP
1005	TB
1006	-
1007	DIABETES
1007	DIABETES
1008	TB
1009	SINUS

16. Disease TABLE:

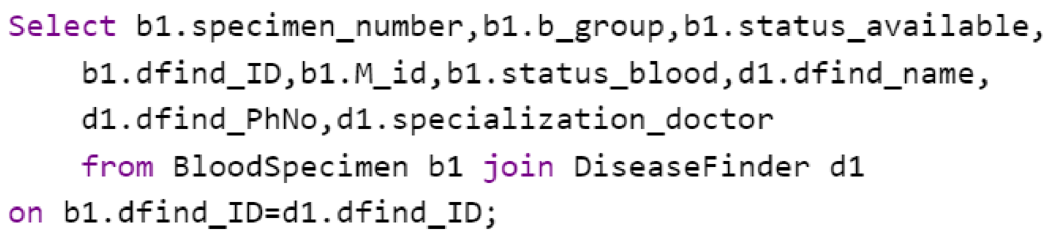


BLOOD_SPECIMEN_NO	BGROUP	DISEASE_NAME
1001	B+	diabetes
1001	B+	TB
1001	B+	BP
1002	O+	-
1003	AB+	HEART_DISEASE
1003	AB+	ROSEA
1004	O-	-
1004	O-	BP
1005	A+	TB
1006	A-	-
1007	AB-	DIABETES

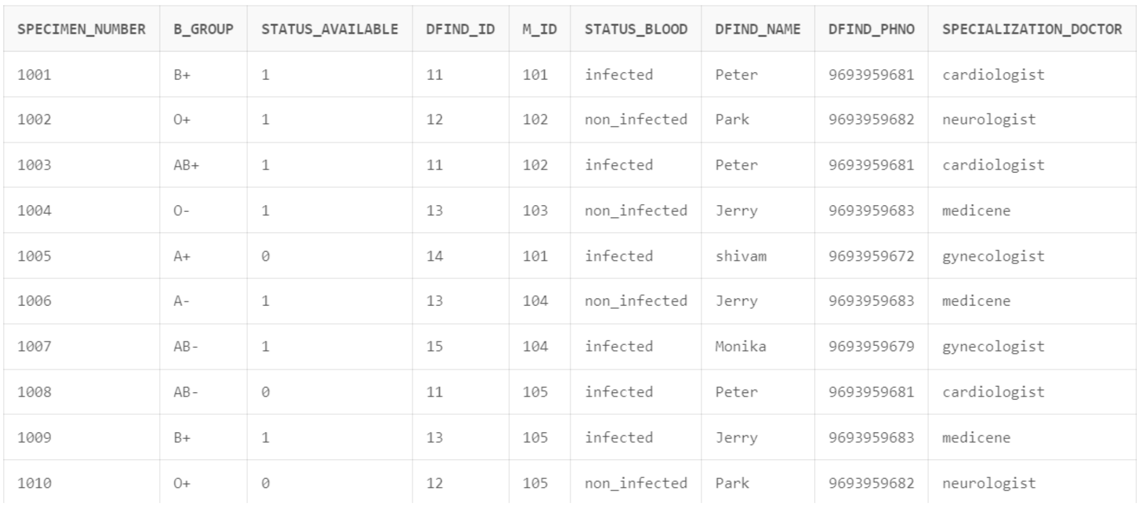
QUERIES:

USING JOINS:

1.

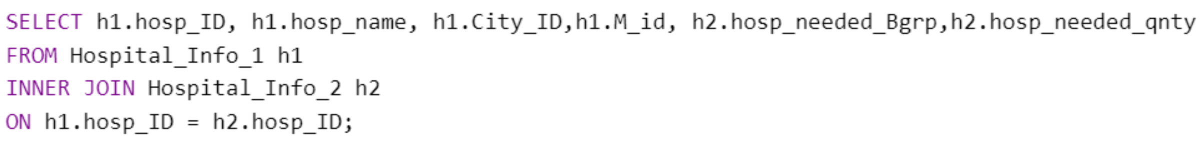


```
Select b1.specimen_number,b1.b_group,b1.status_available,
    b1.dfind_ID,b1.M_id,b1.status_blood,d1.dfind_name,
    d1.dfind_PhNo,d1.specialization_doctor
    from BloodSpecimen b1 join DiseaseFinder d1
on b1.dfind_ID=d1.dfind_ID;
```

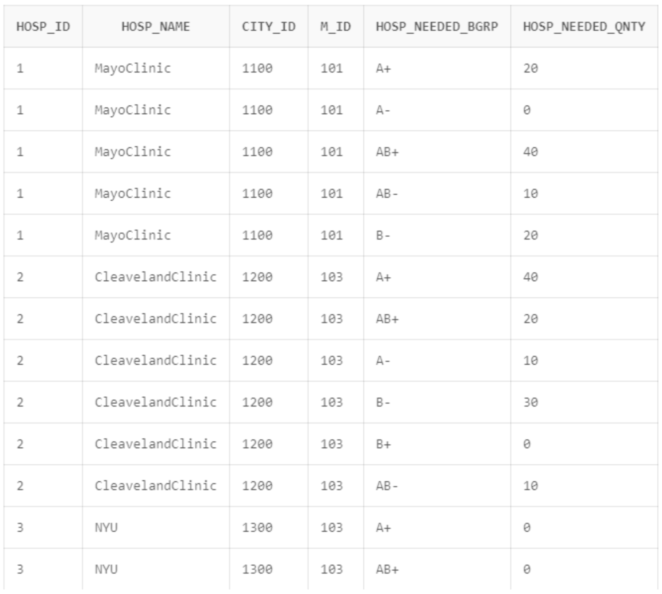


SPECIMEN_NUMBER	B_GROUP	STATUS_AVAILABLE	DFIND_ID	M_ID	STATUS_BLOOD	DFIND_NAME	DFIND_PHNO	SPECIALIZATION_DOCTOR
1001	B+	1	11	101	infected	Peter	9693959681	cardiologist
1002	O+	1	12	102	non_infected	Park	9693959682	neurologist
1003	AB+	1	11	102	infected	Peter	9693959681	cardiologist
1004	O-	1	13	103	non_infected	Jerry	9693959683	medicene
1005	A+	0	14	101	infected	shivam	9693959672	gynecologist
1006	A-	1	13	104	non_infected	Jerry	9693959683	medicene
1007	AB-	1	15	104	infected	Monika	9693959679	gynecologist
1008	AB-	0	11	105	infected	Peter	9693959681	cardiologist
1009	B+	1	13	105	infected	Jerry	9693959683	medicene
1010	O+	0	12	105	non_infected	Park	9693959682	neurologist

2.



```
SELECT h1.hosp_ID, h1.hosp_name, h1.City_ID,h1.M_id, h2.hosp_needed_Bgrp,h2.hosp_needed_qnty
FROM Hospital_Info_1 h1
INNER JOIN Hospital_Info_2 h2
ON h1.hosp_ID = h2.hosp_ID;
```



HOSP_ID	HOSP_NAME	CITY_ID	M_ID	HOSP_NEEDED_BGRP	HOSP_NEEDED_QNTY
1	MayoClinic	1100	101	A+	20
1	MayoClinic	1100	101	A-	0
1	MayoClinic	1100	101	AB+	40
1	MayoClinic	1100	101	AB-	10
1	MayoClinic	1100	101	B-	20
2	CleavelandClinic	1200	103	A+	40
2	CleavelandClinic	1200	103	AB+	20
2	CleavelandClinic	1200	103	A-	10
2	CleavelandClinic	1200	103	B-	30
2	CleavelandClinic	1200	103	B+	0
2	CleavelandClinic	1200	103	AB-	10
3	NYU	1300	103	A+	0
3	NYU	1300	103	AB+	0

3.

```
select Donor.bd_name,Recipient.reci_name,reco_Name from
Recording_Staff
inner join Donor on Recording_Staff.reco_ID = Donor.reco_ID inner join Recipient
on Recording_Staff.reco_ID = Recipient.reco_ID where Donor.bd_Bgroup =
Recipient.reci_Brgp;
```

BD_NAME	RECI_NAME	RECO_NAME
Shivank	Marsh	Walcot
Shivank	Marsh	Walcot
shweta	Peter	Walcot
shweta	Parker	Walcot
Dan	Swetha	Walcot
Dan	Swetha	Walcot
Carrol	akhil	jackson

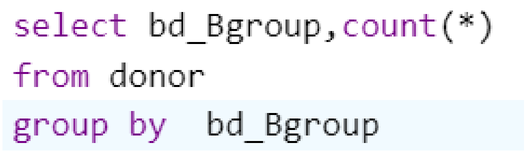
4.

```
SELECT d.bd_name
FROM Donor d
INNER JOIN City c ON d.City_ID = c.City_ID
WHERE c.City_name = 'Austin' AND d.bd_reg_date >= '19-FEB-2015';
```

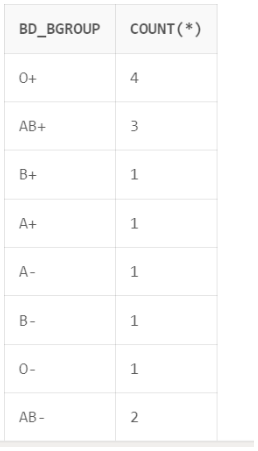
BD_NAME
Shivank
Shivank
Dan
Dan
Elisa
shivansh

USING GROUP BY

1.

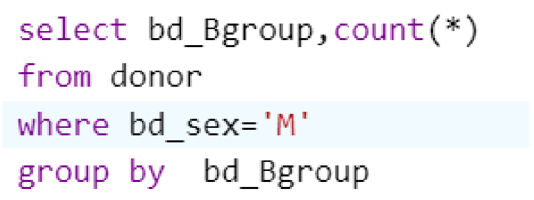


```sql
select bd_Bgroup,count(*)
from donor
group by  bd_Bgroup
```

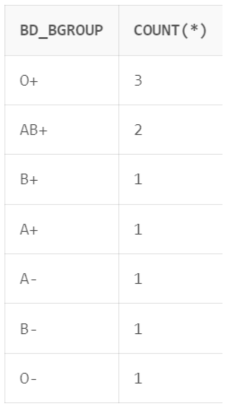


BD_BGROUP	COUNT(*)
O+	4
AB+	3
B+	1
A+	1
A-	1
B-	1
O-	1
AB-	2

2.

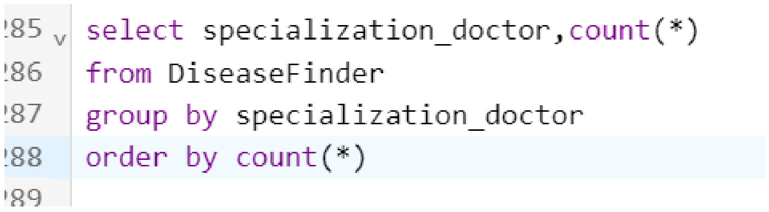


```sql
select bd_Bgroup,count(*)
from donor
where bd_sex='M'
group by  bd_Bgroup
```

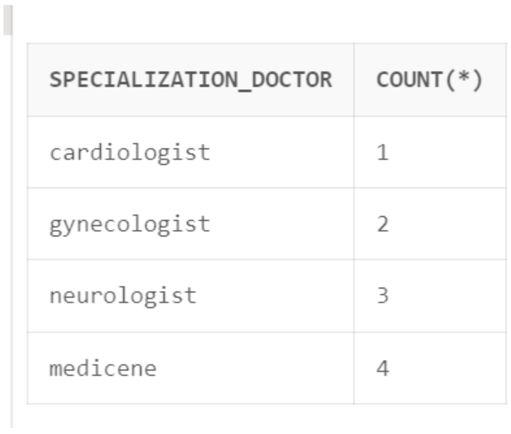


BD_BGROUP	COUNT(*)
O+	3
AB+	2
B+	1
A+	1
A-	1
B-	1
O-	1

3.



```sql
select specialization_doctor,count(*)
from DiseaseFinder
group by specialization_doctor
order by count(*)
```

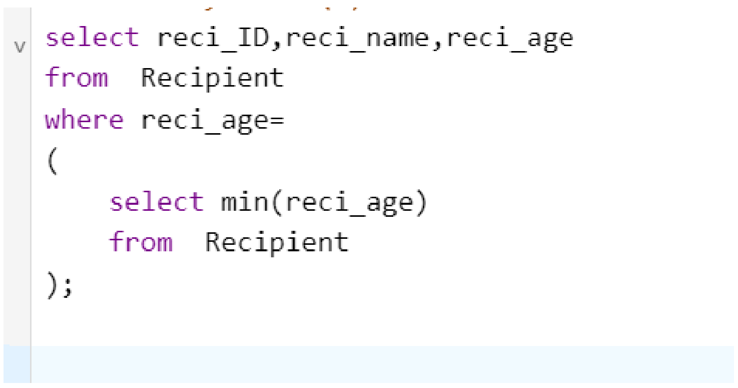


SPECIALIZATION_DOCTOR	COUNT(*)
cardiologist	1
gynecologist	2
neurologist	3
medicene	4

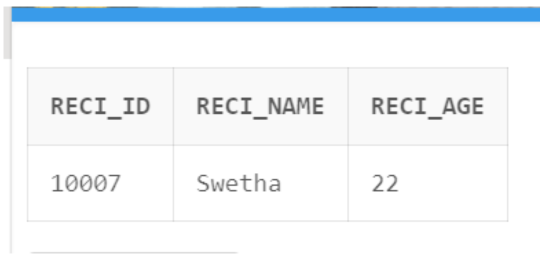
USING SUBQUERIES

SINGLE ROW SUBQUERY

1.



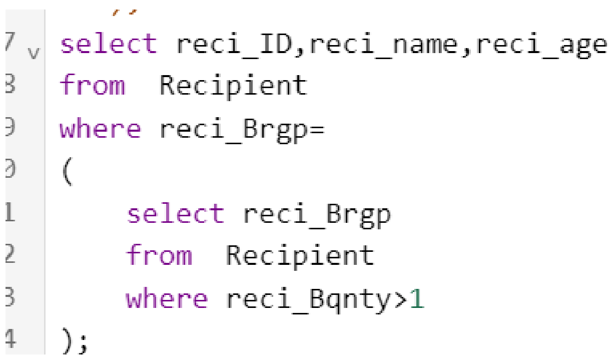
```
select reci_ID,reci_name,reci_age
from  Recipient
where reci_age=
(
    select min(reci_age)
    from  Recipient
);
```



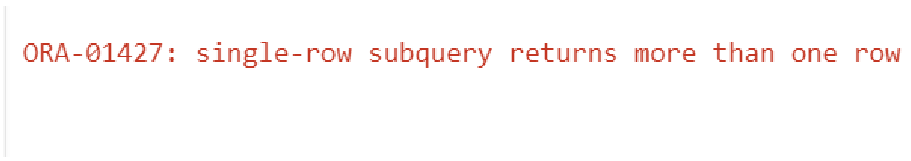
RECI_ID	RECI_NAME	RECI_AGE
10007	Swetha	22

MULTIROW SUBQUERY

1.



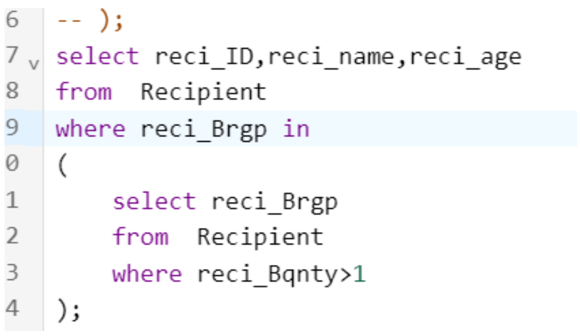
```
select reci_ID,reci_name,reci_age
from  Recipient
where reci_Brgp=
(
    select reci_Brgp
    from  Recipient
    where reci_Bqnty>1
);
```



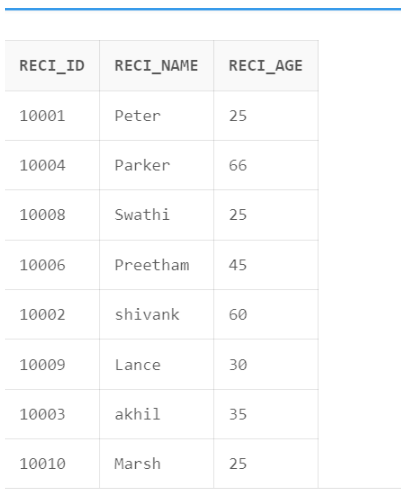
```
ORA-01427: single-row subquery returns more than one row
```

OPERATOR	MEANING
IN	Equals to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

2.

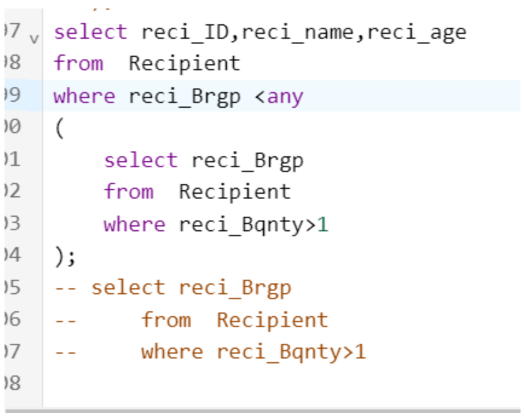


```
-- );
select reci_ID,reci_name,reci_age
from  Recipient
where reci_Brgp in
(
    select reci_Brgp
    from  Recipient
    where reci_Bqnty>1
);
```



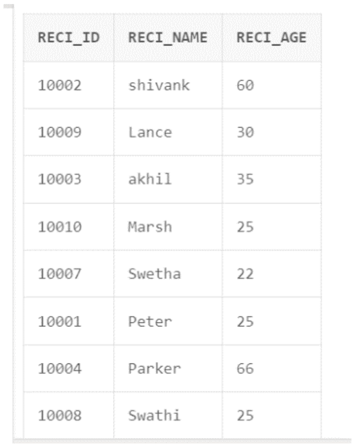
```
RECI_ID  RECI_NAME  RECI_AGE
10001    Peter      25
10004    Parker     66
10008    Swathi     25
10006    Preetham   45
10002    shivank    60
10009    Lance      30
10003    akhil      35
10010    Marsh      25
```

3.



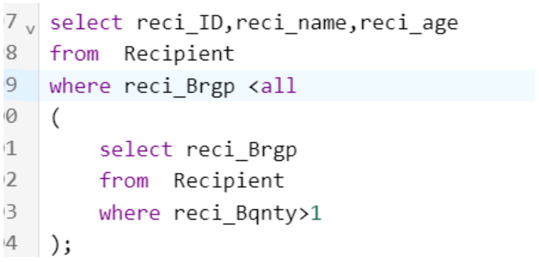
```
select reci_ID,reci_name,reci_age
from  Recipient
where reci_Brgp <any
(
    select reci_Brgp
    from  Recipient
    where reci_Bqnty>1
);
-- select reci_Brgp
--     from  Recipient
--     where reci_Bqnty>1

```

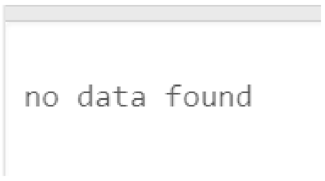


RECI_ID	RECI_NAME	RECI_AGE
10002	shivank	60
10009	Lance	30
10003	akhil	35
10010	Marsh	25
10007	Swetha	22
10001	Peter	25
10004	Parker	66
10008	Swathi	25

4.



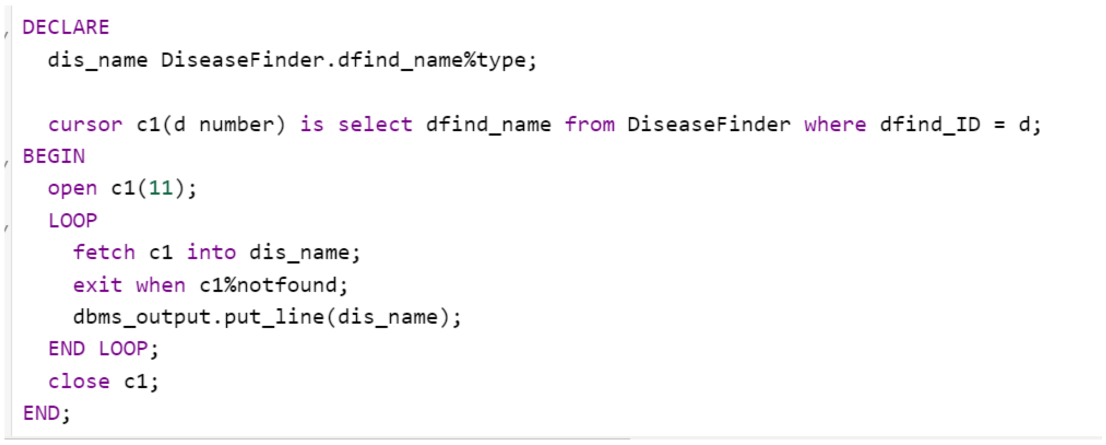
```
select reci_ID,reci_name,reci_age
from  Recipient
where reci_Brgp <all
(
    select reci_Brgp
    from  Recipient
    where reci_Bqnty>1
);
```



```
no data found
```

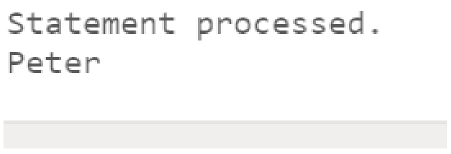
USING PL/SQL

1.



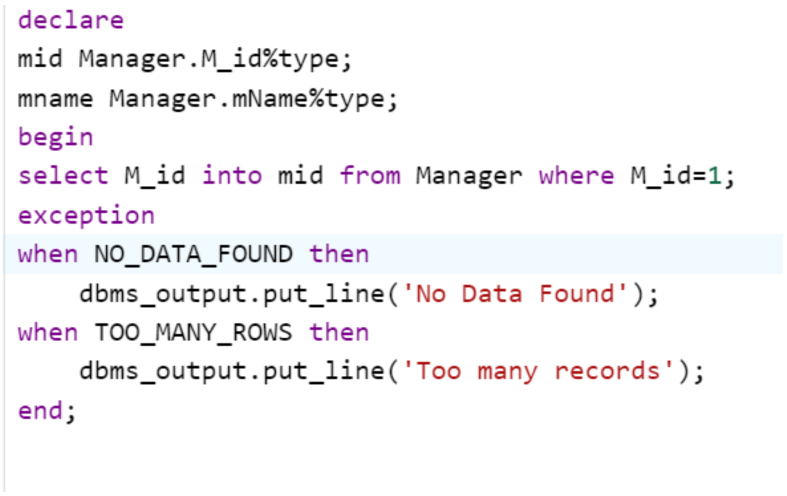
```
DECLARE
  dis_name DiseaseFinder.dfind_name%type;

  cursor c1(d number) is select dfind_name from DiseaseFinder where dfind_ID = d;
BEGIN
  open c1(11);
  LOOP
    fetch c1 into dis_name;
    exit when c1%notfound;
    dbms_output.put_line(dis_name);
  END LOOP;
  close c1;
END;
```

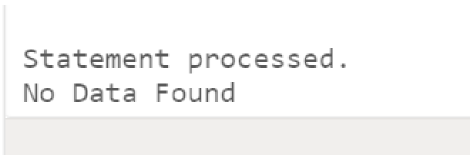


```
Statement processed.
Peter
```

2.

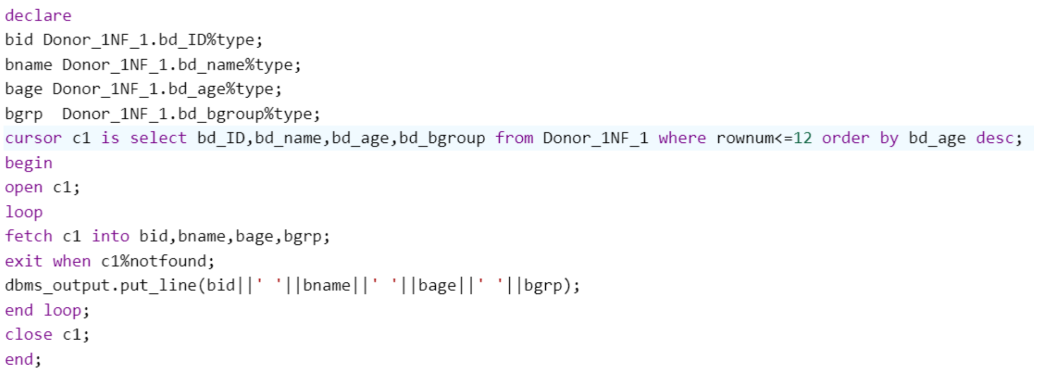


```
declare
mid Manager.M_id%type;
mname Manager.mName%type;
begin
select M_id into mid from Manager where M_id=1;
exception
when NO_DATA_FOUND then
    dbms_output.put_line('No Data Found');
when TOO_MANY_ROWS then
    dbms_output.put_line('Too many records');
end;
```

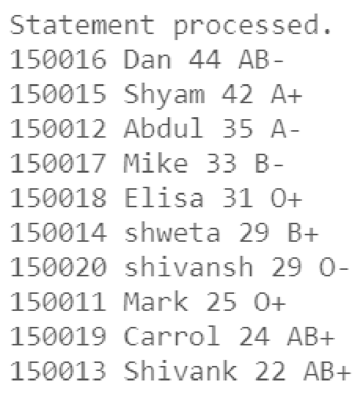


```
Statement processed.
No Data Found
```

3.

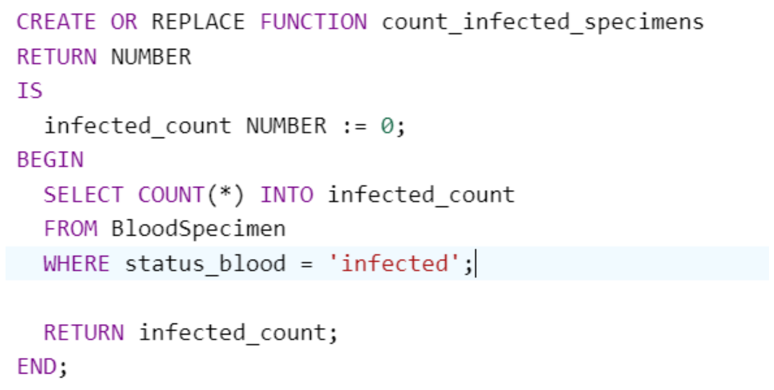


```
declare
bid Donor_1NF_1.bd_ID%type;
bname Donor_1NF_1.bd_name%type;
bage Donor_1NF_1.bd_age%type;
bgrp  Donor_1NF_1.bd_bgroup%type;
cursor c1 is select bd_ID,bd_name,bd_age,bd_bgroup from Donor_1NF_1 where rownum<=12 order by bd_age desc;
begin
open c1;
loop
fetch c1 into bid,bname,bage,bgrp;
exit when c1%notfound;
dbms_output.put_line(bid||' '||bname||' '||bage||' '||bgrp);
end loop;
close c1;
end;
```



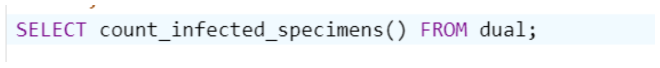
```
Statement processed.
150016 Dan 44 AB-
150015 Shyam 42 A+
150012 Abdul 35 A-
150017 Mike 33 B-
150018 Elisa 31 O+
150014 shweta 29 B+
150020 shivansh 29 O-
150011 Mark 25 O+
150019 Carrol 24 AB+
150013 Shivank 22 AB+
```

4.

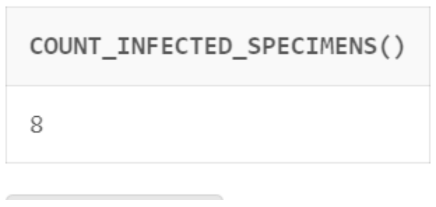


```
CREATE OR REPLACE FUNCTION count_infected_specimens
RETURN NUMBER
IS
  infected_count NUMBER := 0;
BEGIN
  SELECT COUNT(*) INTO infected_count
  FROM BloodSpecimen
  WHERE status_blood = 'infected';

  RETURN infected_count;
END;
```



```
SELECT count_infected_specimens() FROM dual;
```



COUNT_INFECTED_SPECIMENS()
8

CONCLUSION

Our project well addressed the limitations of the existing system. We designed well organized database management system which is a challenging job in this era. We have built a database for a Blood Bank using Microsoft SQL Server. Before implementing the database, in the design phase, we have explored various features, operations of a blood bank to figure out required entities, attributes and the relationship among entities to make an efficient Entity Relationship Diagram (ERD). After analysing all the requirements, we have created our ERD and then converted the ERD to relational model and normalized the tables. Using Microsoft SQL Server we have created the tables for our database and inserted some sample values in the tables. Finally, we have executed sample queries on our database to check its performance to retrieve useful information accurately and speedily.

We have combined all concepts including **Generalization**, **Specialization, PL/SQL, Subqueries, Joins, Procedures, Cursors**

REFERNCES

1. SQL server tutorial: https://www.tutorialspoint.com/ms_sql_server/ 2.
2. Tool for ER diagram: Creately- Creately
3. TOOL for ER to Tables: QuickDBD (quickdatabasediagrams.com)
4. Notes from Prof. Hakki C Cankaya, University of Texas at Dallas, Dallas, Texas
5. Blood Donation Management: Recent trends – Sunil K Joseph, Johnson Michael, Prem Jose Vazhacharickal

Signature of Faculty member