

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY



DBMS PROJECT-1 **BLOOD BANK MANAGEMENT SYSTEM**

SUBMITTED TO

Ms.D.Hemavathi

Date:

Project Team:

Nitesh Reddy (RA111042010016)

Katha Sai Indra Reddy (RA111042010021)

BLOOD BANK MANAGEMENT SYSTEM

ABSTRACT

This project aims to develop a Blood Bank Management System. A Blood Bank Management System can be used in any clinic, hospital, labs or any emergency situation which requires blood units for survival. Our system can be used to find required type of blood in emergency situations from either blood bank or even blood donors.

Current system uses a grapevine communication for finding blood in cases of emergency, may it be by a donor or blood bank. The intentions of proposing such a system is to abolish the panic caused during an emergency due to unavailability of blood.

INTRODUCTION

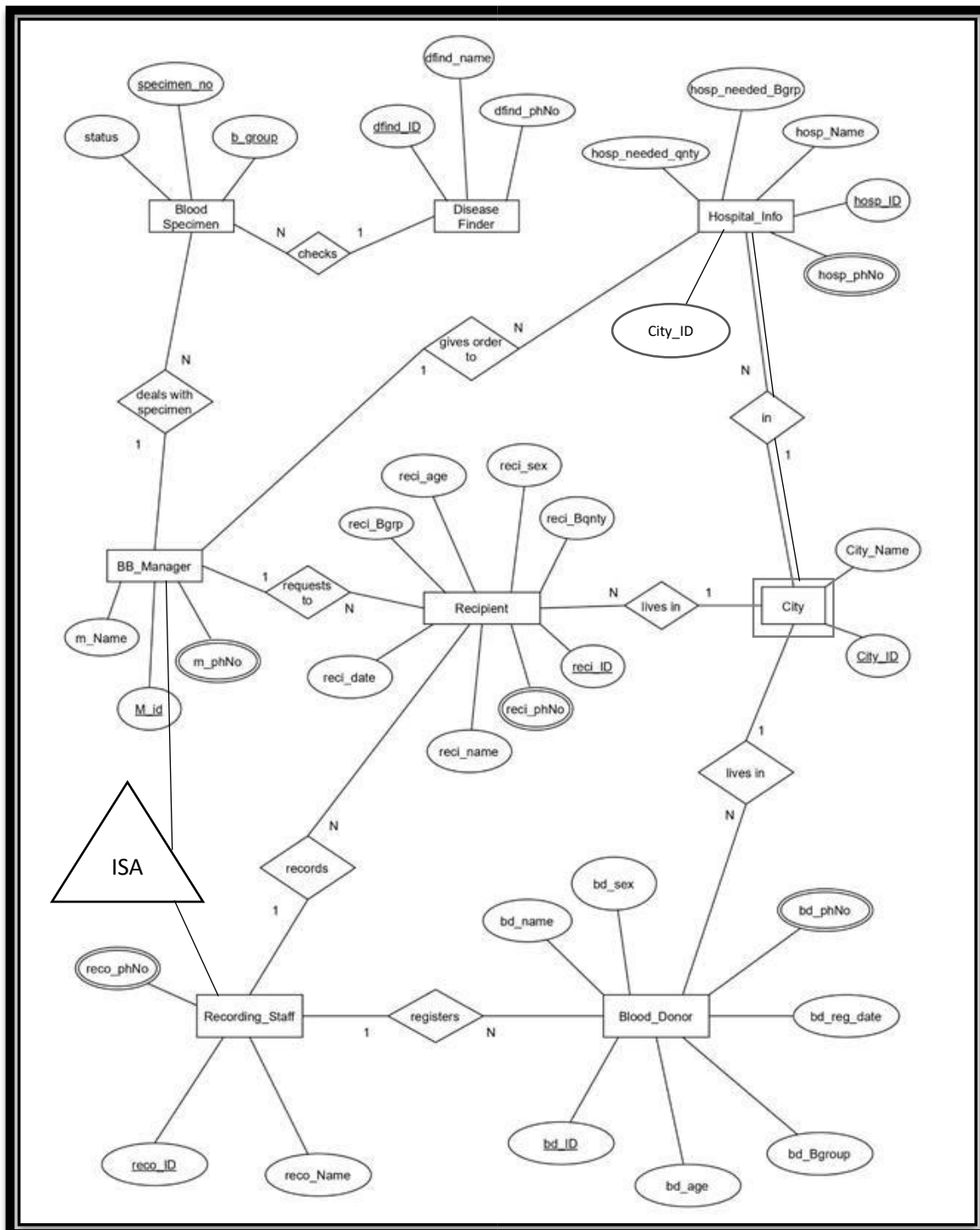
Blood banks collect, store and provide collected blood to the patients who are in need of blood. The people who donate blood are called 'donors'. The banks then group the blood which they receive according to the blood groups. They also make sure that the blood is not contaminated. The main mission of the blood bank is to provide the blood to the hospitals and health care systems which saves the patient's life. No hospital can maintain the health care system without pure and adequate blood.

The major concern each blood bank has is to monitor the quality of the blood and monitor the people who donate the blood, that is 'donors'. But this a tough job. The existing system will not satisfy the need of maintaining quality blood and keep track of donors. To overcome all these limitations we introduced a new system called 'Blood Donation Management System'.

The 'Blood Bank Management System' allows us to keep track of quality of blood and also keeps track of available blood when requested by the acceptor. The existing systems are Manual systems which are time consuming and not so effective. 'Blood Bank Management system' automates the distribution of blood. This database consists of thousands of records of each blood bank.

By using this system searching the available blood becomes easy and saves lot of time than the manual system. It will hoard, operate, recover and analyse information concerned with the administrative and inventory management within a blood bank. This system is developed in a manner that it is manageable, time effective, cost effective, flexible and much man power is not required.

ER Diagram:



INFORMATION OF ENTITIES:

1. Blood_donor:

```
SQL> create table Blood_donor(bd_id number(10) primary key, bd_name varchar2(25), bd_sex varchar2(25), bd_Bgroup varchar2(25) not null, bd_reg_date date, bd_phno number(15), bd_age number(10), check(bd_age>18));

Table created.

SQL> desc blood_donor;

```

Name	Null?	Type
BD_ID	NOT NULL	NUMBER(10)
BD_NAME		VARCHAR2(25)
BD_SEX		VARCHAR2(25)
BD_BGROUP	NOT NULL	VARCHAR2(25)
BD_REG_DATE		DATE
BD_PHNO		NUMBER(15)
BD_AGE		NUMBER(10)

The 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. Other than that name, age , sex , blood group, phone number and registration dates will be stored in database under Blood_donor entity.

2. Recipient:

```
SQL> create table Recipient( reci_id number(10) primary key, reci_name varchar2(25), reci_age number(10), reci_Bgroup varchar2(25) not null, reci_Bqnty number(10) not null, reci_sex varchar2(25), reci_reg_date date, reci_phno number(15));

Table created.

SQL> desc Recipient;

```

Name	Null?	Type
RECI_ID	NOT NULL	NUMBER(10)
RECI_NAME		VARCHAR2(25)
RECI_AGE		NUMBER(10)
RECI_BGROUP	NOT NULL	VARCHAR2(25)
RECI_BQNTY	NOT NULL	NUMBER(10)
RECI_SEX		VARCHAR2(25)
RECI_REG_DATE		DATE
RECI_PHNO		NUMBER(15)

The Recipient is the person who receives blood from blood bank, when blood is given to a recipient a recipient ID (reci_id) is generated and used as primary key for the recipient entity to identify blood recipients information. Along with it name, age, sex, blood group (needed), blood quantity(needed), phone number, and registration dates are also stored in the data base under recipient entity.

3.BB_manager:

```
SQL> create table BB_manager(m_id number(10) primary key,m_name varchar2(25),m_phno number(15));
```

Table created.

```
SQL> desc BB_manager;
```

Name	Null?	Type
M_ID	NOT NULL	NUMBER(10)
M_NAME		VARCHAR2(25)
M_PHNO		NUMBER(15)

The blood bank manager is the person who takes care of the available blood samples in the blood bank, he is also responsible for handling blood requests from recipients and hospitals. Blood manager has a unique identification number (m_id) used as primary key along with name and phone number of blood bank manager will be stored in data base under BB_manager entity.

4.Recording_staff:

```
SQL> create table Recording_staff(reco_id number(10) primary key,reco_name varchar2(25),reco_phno number(15));
```

Table created.

```
SQL> desc Recording_staff;
```

Name	Null?	Type
RECO_ID	NOT NULL	NUMBER(10)
RECO_NAME		VARCHAR2(25)
RECO_PHNO		NUMBER(15)

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 and recorder's phone number will also be stored in the data base under Recording_staff entity.

5.Bloodspecimen:

```
SQL> create table Bloodspecimen(specimen_number number(10) primary key,b_group varchar2(25) not null,status varchar2(15) not null);
```

Table created.

```
SQL> desc Bloodspecimen;
```

Name	Null?	Type
SPECIMEN_NUMBER	NOT NULL	NUMBER(10)
B_GROUP	NOT NULL	VARCHAR2(25)
STATUS	NOT NULL	VARCHAR2(15)

In data base, under Blood specimen entity we will store the information of blood samples which are available in the blood bank. In this entity specimen_number and b_group together will be primary key along with status attribute which will show if the blood is contaminated or not.

6.Diseasefinder:

```
SQL> create table Diseasefinder(dfnd_ID number(10) primary key,dfnd_name varchar2(25),dfnd_phno number(15));
Table created.

SQL> desc Diseasefinder;
Name                                     Null?    Type
-----
DFIND_ID                                NOT NULL NUMBER(10)
DFIND_NAME                              VARCHA2(25)
DFIND_PHNO                               NUMBER(15)
```

In data base , under Diseasefinder entity we will store the information of the doctor who checks the blood for any kind of contaminations. To store that information we have unique identification number (dfnd_ID) as primary key. Along with name and phone number of the doctor will also be stored under same entity.

7.hospital_info:

```
SQL> desc hospital_info;
Name                                     Null?    Type
-----
HOSP_ID                                NOT NULL NUMBER(10)
HOSP_NAME                              VARCHA2(25)
HOSP_NEEDED_BGRP                        NOT NULL VARCHA2(25)
HOSP_NEEDED_BQNTY                       NOT NULL NUMBER(10)
CITY_ID                                NOT NULL NUMBER(10)
```

In the data base, under hospital_info entity we will store the information of hospitals. In this hosp_id and hosp_needed_bgrp together makes the primary key. We will store hospital name and the blood quantity required at the hospital.

8.city:

```
SQL> create table city(city_id number(10) not null,city_name varchar2(25) not null,foreign key(city_id) references hospital_info(hosp_id));
Table created.

SQL> desc city;
Name                                     Null?    Type
-----
CITY_ID                                NOT NULL NUMBER(10)
CITY_NAME                              NOT NULL VARCHA2(25)
```

This entity will store the information of cities where donors, recipients and hospitals are present. A unique identification number (city_id) will be used as primary key to identify the information about the city. Along with ID city names will also be stored under this entity.

RELATIONSHIP BETWEEN ENTITIES

1. City and hospital_Info:

Relationship = “in”

Type of relation = 1 to many

Explanation = A city can have many hospital in it. One hospital will belong in one city.

2. City and Blood_donor:

Relationship = “lives in”

Type of relation = 1 to many

Explanation = In a city, many donor can live. One donor will belong to one city.

3. City and Recipient:

Relationship = “lives in”

Type of relation = 1 to many

Explanation = In a city, many recipient can live. One recipient will belong to one city.

4. Recording_staff and Donor:

Relationship = “registers”

Type of relation = 1 to many

Explanation = One recording staff can register many donors. One donor will register with one recording officer.

5. Recording_staff and Recipient:

Relationship = “records”

Type of relation = 1 to many

Explanation = One recording staff can record many recipients. One recipient will be recorded by one recording officer.

6.Hospital_Info and BB_manager:

Relationship = “gives order to”

Type of relation = 1 to many

Explanation = One Blood bank manager can handle and process requests from many hospitals. One hospital will place request to on blood bank manager.

7.BB_manager and Bloodspecimen:

Relationship = “deales with specimen”

Type of relation = 1 to many

Explanation = One Blood bank manager can manage many blood specimen and one specimen will be managed by one manager.

8.Recipient and BB_manager:

Relationship = “requests to”

Type of relation = 1 to many

Explanation = One recipient can request blood to one manager and one manager can handle requests from many recipients.

9.Disease_finder and Bloodspecimen:

Relationship = “checks”,

Type of relation = 1 to many

Explanation = A disease finder can check many blood samples. One blood sample is checked by one disease finder.

After inserting values into table

1.Blood_donor:

```
SQL> select * from Blood_donor;
```

BD_ID	BD_NAME	BD_SEX
BD_BGROU	BD_REG_DA	BD_PHNO
BD_AGE		
O+	150011 Mark	M
	19-JUL-15	9876543567
25		
A-	150012 Abdul	M
	24-DEC-15	8745364578
35		
AB+	150013 Shivank	M
	28-AUG-15	8764536278
22		

BD_ID	BD_NAME	BD_SEX
BD_BGROU	BD_REG_DA	BD_PHNO
BD_AGE		
B+	150014 shweta	M
	17-DEC-15	9.8746E+10
29		
A+	150015 Shyam	M
	22-NOV-16	9865436578
42		
AB-	150016 Dan	F
	06-FEB-16	9765434564
44		

BD_ID	BD_NAME	BD_SEX
BD_BGROU	BD_REG_DA	BD_PHNO
BD_AGE		
B-	150017 Mike	M
	15-OCT-16	8976547546
33		
O+	150018 Elisa	F
	04-JAN-16	8745342313
31		
AB+	150019 Carrol	F
	10-SEP-16	9871234356
24		

BD_ID	BD_NAME	BD_SEX
BD_BGROU	BD_REG_DA	BD_PHNO
BD_AGE		
O-	150020 shivansh	M
	17-DEC-16	9845343423
29		

10 rows selected.

2. Recipient:

```
SQL> select * from Recipient;
```

RECI_ID	RECI_NAME	RECI_AGE	RECI_BGROUP
RECI_BQNTY	RECI_SEX	RECI_REG_	RECI_PHNO
10001	Peter	25	B+
2	M	17-DEC-15	9845362718
10002	shivank	60	A+
1	M	16-DEC-15	8743241564
10003	akhil	35	AB+
1	M	10-OCT-15	8123465768
RECI_ID	RECI_NAME	RECI_AGE	RECI_BGROUP
RECI_BQNTY	RECI_SEX	RECI_REG_	RECI_PHNO
10004	Parker	66	B+
1	M	17-NOV-16	8765432345
10005	jojo	53	B-
1	M	17-APR-15	8976342109
10006	Preetham	45	O+
2	M	17-DEC-16	8909876756

RECI_ID	RECI_NAME	RECI_AGE	RECI_BGROUP
RECI_BQNTY	RECI_SEX	RECI_REG_	RECI_PHNO
10007	Swetha	22	AB+
1	F	20-JAN-16	6753425467
10008	Lance	30	A+
2	F	16-FEB-15	8912309865
10009	Swathi	25	B+
2	F	14-DEC-15	7865234567
RECI_ID	RECI_NAME	RECI_AGE	RECI_BGROUP
RECI_BQNTY	RECI_SEX	RECI_REG_	RECI_PHNO
10010	Marsh	25	AB-
4	M	17-OCT-16	7892341567

```
10 rows selected.
```

3.BB_manager:

```
SQL> select * from BB_manager;
```

M_ID	M_NAME	M_PHNO
101	shivank	9854323154
102	shwetanshu	8974325467
103	singh	7653214356
104	yusuf	8976543245
105	jackson	6754323456
106	akhil	9090989878
107	jojo	9076544545
108	stella	7676565643
109	monika	9083212354
110	himanshi	7656565643

10 rows selected.

4.Recording_staff:

```
SQL> select * from Recording_staff;
```

RECO_ID	RECO_NAME	RECO_PHNO
101012	Lekha	4536234526
101112	shivam	4352617634
101212	Walcot	4563872354
101312	jackson	4545342312
101412	Silva	4589765456
101512	Adrian	4321098765
101612	shivam	4678986545
101712	shyam	5643212354
101812	Jerry	5487612345
101912	Tim	4509812546

10 rows selected.

5. Bloodspecimen:

```
SQL> select * from Bloodspecimen;
```

SPECIMEN_NUMBER	B_GROUP	STATUS
1001	B+	1
1002	O+	1
1003	AB+	1
1004	O-	1
1005	A+	0
1006	A-	1
1007	AB-	1
1008	AB-	0
1009	B+	1
1010	O+	0

```
10 rows selected.
```

6. Diseasefinder:

```
SQL> select * from Diseasefinder;
```

DFIND_ID	DFIND_NAME	DFIND_PHNO
11	peter	9843251763
12	Park	8973423456
13	Jerry	8976342345
14	shivam	9098123423
15	Monika	7864312345
16	Ram	8989343428
17	Swathi	9988664343
18	Gautham	9090231345
19	Ashwin	6754329809
20	Yash	9124569876

```
10 rows selected.
```

7.hopital_info:

```
SQL> select * from hospital_info;
```

HOSP_ID	HOSP_NAME	HOSP_NEEDED_BGRP	HOSP_NEEDED_BQNTY
1	Mayoclinic	A+	20
2	CleavelandClinic	AB+	40
3	NYU	O+	30
4	Baylor	O-	10
5	Chariton	AB-	30
6	Greenoaks	B-	40
7	Forestpark	A-	20
8	Parkland	B+	30
9	Pinecreek	AB+	10
10	WalnutHill	O+	40

10 rows selected.

8.city:

```
SQL> select * from city;
```

CITY_ID	CITY_NAME
11	Delhi
12	Punjab
13	Hyd
14	Vizag
1500	Vijayawada
1600	Chennai
1700	Mumbai
1800	Jaipur
1900	Benguluru
20	Ranchi

10 rows selected.

Relational Operators:

```
SQL> select * from Blood_donor where bd_bgroup='A+';
```

BD_ID	BD_NAME	BD_SEX	
BD_BGROUP	BD_REG_DA	BD_PHNO	BD_AGE
150015	Shyam	M	
A+	22-NOV-16	9865436578	42

```
SQL> select * from Blood_donor where bd_age<25;
```

BD_ID	BD_NAME	BD_SEX	
BD_BGROUP	BD_REG_DA	BD_PHNO	BD_AGE
150013	Shivank	M	
AB+	28-AUG-15	8764536278	22
150019	Carrol	F	
AB+	10-SEP-16	9871234356	24

```
SQL> select * from Blood_donor where bd_sex<>'M';
```

BD_ID	BD_NAME	BD_SEX	
BD_BGROUP	BD_REG_DA	BD_PHNO	BD_AGE
150016	Dan	F	
AB-	06-FEB-16	9765434564	44
150018	Elisa	F	
O+	04-JAN-16	8745342313	31
150019	Carrol	F	
AB+	10-SEP-16	9871234356	24

```
SQL> select * from hospital_info where hosp_needed_bqnty>=30;
```

HOSP_ID	HOSP_NAME	HOSP_NEEDED_BGRP	HOSP_NEEDED_BQNTY
2	CleavelandClinic	AB+	40
3	NYC	O+	30
5	Chariton	AB-	30
6	Greennoaks	B-	40
8	Parkland	B+	30
10	WalnutHill	O+	40

```
6 rows selected.
```

Arithmetic Operator:

```
SQL> select specimen_number+status as sum from Bloodspecimen;
```

SUM
1002
1003
1004
1005
1005
1007
1008
1008
1010
1010

```
10 rows selected.
```

```
SQL> select hosp_needed_bqnty-hosp_id as sub from hospital_info;
```

SUB
19
38
27
6
25
34
13
22
1
30

```
10 rows selected.
```

Logical Operators:

```
SQL> select * from Recipient where reci_sex='M' AND reci_bgroup='B+';
```

RECI_ID	RECI_NAME	RECI_AGE	RECI_BGROUP
10001	Peter	25	B+
2	M	17-DEC-15	9845362718
10004	Parker	66	B+
1	M	17-NOV-16	8765432345

```
SQL> select * from Bloodspecimen where b_group='B+' OR status=0;
```

SPECIMEN_NUMBER	B_GROUP	STATUS
1001	B+	1
1005	A+	0
1008	AB-	0
1009	B+	1
1010	O+	0

```
SQL> select * from Bloodspecimen where specimen_number between 1005 and 1010;
```

SPECIMEN_NUMBER	B_GROUP	STATUS
1005	A+	0
1006	A-	1
1007	AB-	1
1008	AB-	0
1009	B+	1
1010	O+	0

6 rows selected.

```
SQL> select * from Recording_staff where reco_name LIKE '%h%';
```

RECO_ID	RECO_NAME	RECO_PHNO
101012	Lekha	4545372687
101112	shivam	4567382764
101612	shivam	4102938475
101712	shyam	7653728910

SQL FUNCTIONS:

```
SQL> select initcap(city_name) from city;
```

```
INITCAP(CITY_NAME)
```

```
-----
```

```
Delhi  
Punjab  
Hyd  
Vijayawada  
Mumbai  
Chennai  
Kolkata  
Bengaluru
```

```
SQL> select upper(city_name) from city;
```

```
UPPER(CITY_NAME)
```

```
-----
```

```
DELHI  
PUNJAB  
HYD  
VIJAYAWADA  
MUMBAI  
CHENNAI  
KOLKATA  
BENGALURU
```

```
8 rows selected.
```

```
SQL> select hosp_id,AVG(hosp_needed_bqnty) from hospital_info where hosp_id IN(1,2,3,4,5) GROUP BY hosp_id HAVING min(hosp_needed_bqnty)>1;
```

```
HOSP_ID  AVG(HOSP_NEEDED_BQNTY)
```

```
-----
```

1	2
2	3
3	4
4	3
5	2

JOINING TABLES:

1.INNER JOIN:

```
SQL> select diseasefinder.dfind_name,diseasefinder.dfind_phno,city.city_name from diseasefinder INNER JOIN city on diseasefinder.dfind_id=city.city_id;
```

DFIND_NAME	DFIND_PHNO	CITY_NAME
peter	9874343423	Delhi
park	8989745362	Punjab
Jerry	8963425674	Hyd
Shivam	989765423	Vizag
Yash	8724316537	Ranchi

2.LEFT JOIN:

```
SQL> select diseasefinder.dfind_name,diseasefinder.dfind_phno,city.city_name from diseasefinder LEFT JOIN city on diseasefinder.dfind_id=city.city_id;
```

DFIND_NAME	DFIND_PHNO	CITY_NAME
peter	9874343423	Delhi
park	8989745362	Punjab
Jerry	8963425674	Hyd
Shivam	989765423	Vizag
Monika	8989654321	
Ram	9132764527	
Swathi	8143094526	
Gautham	9025371928	
Ashwin	9024357654	
Yash	8724316537	Ranchi

10 rows selected.

3.RIGHT JOIN:

```
SQL> select diseasefinder.dfind_name,diseasefinder.dfind_phno,city.city_name from diseasefinder RIGHT JOIN city on diseasefinder.dfind_id=city.city_id;
```

DFIND_NAME	DFIND_PHNO	CITY_NAME
peter	9874343423	Delhi
park	8989745362	Punjab
Jerry	8963425674	Hyd
Shivam	989765423	Vizag
		Vijayawada
		Chennai
		Mumbai
		Jaipur
		Benguluru
Yash	8724316537	Ranchi

10 rows selected.

4.FULL JOIN:

```
SQL> select diseasefinder.dfind_name,diseasefinder.dfind_phno,city.city_name from diseasefinder FULL JOIN city on diseasefinder.dfind_id=city.city_id;
```

DFIND_NAME	DFIND_PHNO	CITY_NAME
peter	9874343423	Delhi
park	8989745362	Punjab
Jerry	8963425674	Hyd
Shivam	989765423	Vizag
		Vijayawada
		Chennai
		Mumbai
		Jaipur
		Benguluru
Yash	8724316537	Ranchi
Monika	8989654321	
DFIND_NAME	DFIND_PHNO	CITY_NAME
Ashwin	9024357654	
Swathi	8143094526	
Gautham	9025371928	
Ram	9132764527	

15 rows selected.

Sub queries:

```
SQL> select bd_name,bd_sex from Blood_donor where bd_age>(select bd_age from Blood_donor where bd_name='shivansh');
```

BD_NAME	BD_SEX
Abdul	M
Shyam	M
Dan	F
Mike	M
Elisa	F

```
SQL> select specimen_number,b_group from Bloodspecimen where status=(select status from Bloodspecimen where specimen_number=1001);
```

SPECIMEN_NUMBER	B_GROUP
1001	bgroup
1002	O+
1003	AB+
1004	O-
1006	A-
1007	AB-
1009	B+

7 rows selected.

```
SQL> select hosp_name,hosp_needed_bgrp,hosp_needed_bqnty from hospital_info where hosp_needed_bqnty<(select hosp_needed_bqnty from hospital_info where hosp_name='WalnutHill');
```

HOSP_NAME	HOSP_NEEDED_BGRP	HOSP_NEEDED_BQNTY
Mayoclinic	A+	20
NYC	O+	30
Baylor	O-	10
Chariton	AB-	30
Forestpark	A-	20
Parkland	B+	30
Pinecreek	AB+	10

7 rows selected.

```
SQL> select bd_id,bd_name,bd_bgroup,bd_phno from Blood_donor where bd_age<(select bd_age from Blood_donor where bd_name='Abdul') AND bd_sex=(select bd_sex from Blood_donor where bd_name='Abdul');
```

BD_ID	BD_NAME	BD_BGROUP	BD_PHNO
150011	Mark	O+	6767432872
150013	Shivank	AB+	8989454532
150017	Mike	B-	7812095634
150020	shivansh	O-	9845357860

```
SQL> select bd_id,bd_name,bd_bgroup,bd_phno from Blood_donor where bd_age>(select bd_age from Blood_donor where bd_name='Abdul') AND bd_sex=(select bd_sex from Blood_donor where bd_name='Abdul');
```

BD_ID	BD_NAME	BD_BGROUP	BD_PHNO
150015	Shyam	A+	9090343476

```
SQL> select bd_id,bd_name,bd_sex,bd_bgroup from Blood_donor where bd_age>(select MIN(bd_age) from Blood_donor);
```

	BD_ID	BD_NAME	BD_SEX
BD_BGROUP			
O+	150011	Mark	M
A-	150012	Abdul	M
B+	150014	shweta	F
	BD_ID	BD_NAME	BD_SEX
BD_BGROUP			
A+	150015	Shyam	M
AB-	150016	Dan	F
B-	150017	Mike	M

	BD_ID	BD_NAME	BD_SEX
BD_BGROUP			
O+	150018	Elisa	F
AB+	150019	Carrol	F
O-	150020	shivansh	M

9 rows selected.

```
SQL> select bd_id,bd_name,bd_sex,bd_bgroup from Blood_donor where bd_age=(select MAX(bd_age) from Blood_donor);
```

	BD_ID	BD_NAME	BD_SEX
BD_BGROUP			
AB-	150016	Dan	F

```
SQL> select reci_id,reci_name,reci_sex,reci_bqnty,reci_age from Recipient where reci_age>ANY(select reci_age from Recipient where reci_sex='M');
```

RECI_ID	RECI_NAME	RECI_SEX	RECI_BQNTY
10002	Shivank	M	1
10003	akhil	M	1
1004	Parker	M	1
1006	Lance	F	2

```
SQL> select reci_id,reci_name,reci_sex,reci_bqnty,reci_age from Recipient where reci_age>ANY(select reci_age from Recipient where reci_sex='M')AND reci_bqnty<>2;
```

RECI_ID	RECI_NAME	RECI_SEX	RECI_BQNTY
10002	Shivank	M	1
10003	akhil	M	1
1004	Parker	M	1

```
SQL> select reci_id,reci_name,reci_sex,reci_bqnty,reci_age from Recipient where reci_age<ANY(select reci_age from Recipient where reci_sex='M')AND reci_bqnty<>2;
```

RECI_ID	RECI_NAME	RECI_SEX	RECI_BQNTY
10002	Shivank	M	1
10003	akhil	M	1

PL/SQL:

```
SQL> DECLARE
  2   CURSOR city_cursor IS
  3     SELECT *
  4     FROM city;
  5
  6   v_city_id city.city_id%TYPE;
  7   v_city_name city.city_name%TYPE;
  8 BEGIN
  9   OPEN city_cursor;
 10
 11   LOOP
 12     FETCH city_cursor INTO v_city_id, v_city_name;
 13     EXIT WHEN city_cursor%NOTFOUND;
 14
 15     DBMS_OUTPUT.PUT_LINE('City ID: ' || v_city_id || ' | City Name: ' || v_city_name);
 16   END LOOP;
 17
 18   CLOSE city_cursor;
 19 END;
 20 /
City ID: 1 | City Name: New York
City ID: 2 | City Name: Los Angeles
City ID: 3 | City Name: Chicago
City ID: 4 | City Name: Houston
City ID: 5 | City Name: Miami
City ID: 6 | City Name: Seattle
City ID: 7 | City Name: Boston
City ID: 8 | City Name: San Francisco
City ID: 9 | City Name: Dallas
City ID: 10 | City Name: Philadelphia
```

PL/SQL procedure successfully completed.

```
SQL> DECLARE
  2   CURSOR c_hospital_info IS
  3     SELECT hosp_id, hosp_name, hosp_needed_bgrp, hosp_needed_bqnty
  4     FROM hospital_info;
  5
  6   v_hosp_id hospital_info.hosp_id%TYPE;
  7   v_hosp_name hospital_info.hosp_name%TYPE;
  8   v_hosp_needed_bgrp hospital_info.hosp_needed_bgrp%TYPE;
  9   v_hosp_needed_bqnty hospital_info.hosp_needed_bqnty%TYPE;
 10 BEGIN
 11   OPEN c_hospital_info;
 12
 13   LOOP
 14     FETCH c_hospital_info INTO v_hosp_id, v_hosp_name, v_hosp_needed_bgrp, v_hosp_needed_bqnty;
 15
 16     EXIT WHEN c_hospital_info%NOTFOUND;
 17
 18     DBMS_OUTPUT.PUT_LINE('Hospital ID: ' || v_hosp_id || ' | Hospital Name: ' || v_hosp_name || ' | Hosp Needed Blood Group: ' ||
 19 | v_hosp_needed_bgrp || ' | Hosp Needed Blood Quantity: ' || v_hosp_needed_bqnty);
 19   END LOOP;
 20
 21   CLOSE c_hospital_info;
 22 END;
 23 /
```

```
Hospital ID: 1 | Hospital Name: ABC Hospital | Hosp Needed Blood Group: A+ |  
Hosp Needed Blood Quantity: 20  
Hospital ID: 2 | Hospital Name: XYZ Hospital | Hosp Needed Blood Group: O- |  
Hosp Needed Blood Quantity: 10  
Hospital ID: 3 | Hospital Name: PQR Hospital | Hosp Needed Blood Group: B+ |  
Hosp Needed Blood Quantity: 5  
Hospital ID: 4 | Hospital Name: LMN Hospital | Hosp Needed Blood Group: B- |  
Hosp Needed Blood Quantity: 15  
Hospital ID: 5 | Hospital Name: QRS Hospital | Hosp Needed Blood Group: A- |  
Hosp Needed Blood Quantity: 5  
Hospital ID: 6 | Hospital Name: TUV Hospital | Hosp Needed Blood Group: AB+ |  
Hosp Needed Blood Quantity: 10  
Hospital ID: 7 | Hospital Name: JKL Hospital | Hosp Needed Blood Group: O+ |  
Hosp Needed Blood Quantity: 25  
Hospital ID: 8 | Hospital Name: DEF Hospital | Hosp Needed Blood Group: B+ |  
Hosp Needed Blood Quantity: 12  
Hospital ID: 9 | Hospital Name: MNO Hospital | Hosp Needed Blood Group: AB- |  
Hosp Needed Blood Quantity: 8  
Hospital ID: 10 | Hospital Name: STU Hospital | Hosp Needed Blood Group: A+ |  
Hosp Needed Blood Quantity: 18
```

```
PL/SQL procedure successfully completed.
```