Project Report On

BLOOD BANK MANAGEMENT SYSTEM

Submitted by

M VAMSI KRISHNA (201EC269)

M GEETHA SRIDHAR (201ME320)

I SHANKAR VARMA (201EE224)

IV SEM B.Tech

in partial fulfillment for the award of the degree

of

Bachelor of Technology

In

Information Technology

At

under the guidance of

Ms. J R Shruti

Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

May 2022

CERTIFICATE

This is to certify that the project entitled **"BLOOD BANK MANAGEMENT SYSTEM "** has been submitted by '**M VAMSI KRISHNA'** student of second year B.Tech (ECE)**, 'M GEETHA SRIDHAR'** student of second year B.Tech (MECHANICAL)**, 'I SHANKAR VARMA'** student of second year B.Tech (EEE), National Institute of Technology Karnataka, Surathkal, during the even semester of the academic year 2021 - 2022, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal.

(Signature of the Guide)

Place:

Date:

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

Course code : IT 252

Course Title: Database Systems

Title of the Project: ***BLOOD BANK MANAGEMENT SYSTEM***

Details of Project Group

Name of the Student	*Register No.*	*Signature with Date*
1. M VAMSI KRISHNA	201EC269	
2. M GEETHA SRIDHAR	201ME320	
3. I SHANKAR VARMA	201EE224	

Name of Instructor: Ms J R Shruti

Signature of the Instructor:

Place:

Date:

1. PROBLEM STATEMENT:

Blood Bank Management System (BBMS) is a web based system that can assists the information of blood bag during its handling in the blood bank. With this system, the user of this system can key in the result of blood test that has been conducted to each of the blood bag received by the blood bank. The result of test will indicate whether the blood bag can be delivered to patient or not.

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received have to be managed thoroughly so that there will be no negative effect to the blood receiver once they received blood.

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 staffs and nurses only are informed about the blood donation schedule for each month on the whiteboard at the blood house. So they are using manual way in 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 whiteboard and it is written by using whiteboard marker.

To oversee these, the BBMS interface will be constructed to cater for the blood house staff to post about the blood donation events. These details can be viewed by the public so that they know and they can allocate some time to go and donate their blood. To ensure that the blood donation event schedule is informed among the blood house staff, there will be an interface for staff to be able to fill in details and list of location of the blood donation events for each month. By having this function in BBMS, it is easier for the blood staff to make any correction if there is any incorrect details and make any changes if there is any changes in location or specified date.

One of the factor of the public afraid to donate their blood is they believe in myths. The myths that they always believe are, if they donate their blood they will become fat and if they donate their blood, their blood will become less in total of amount and they will become pale. This BBMS should provide more information in order to educate the public so that they know blood donation will not give bad effects. By giving awareness to the public, this will increase volunteers to donate their blood.

Donor will be person who donates the blood. It will contain information such as name,age,gender,contact,blood group, etc.

The patient will be person who will receive the blood this will have the id , name , gender, blood type, age, status.

Blood bank has the data of blood type, DID, Address, Phno , Issues

Hospital consists of HNO, Name, Phno, Address

Requests consists of Rid, Blood type, Quantity
Blood Bank Manager consists of Emp id, Name, Phno, Email_id

2. ACTORS:

User
Manager
Receptionist

3. SOME SAMPLE QUERIES:

A Manager can

- Check list of Donors
- Check health status of Donor
- Blood group of Donors also can be checked
- Age group of Donors can be checked
- Can check the request from hospitals
- Get list of donors who donate with in 2 days of registration

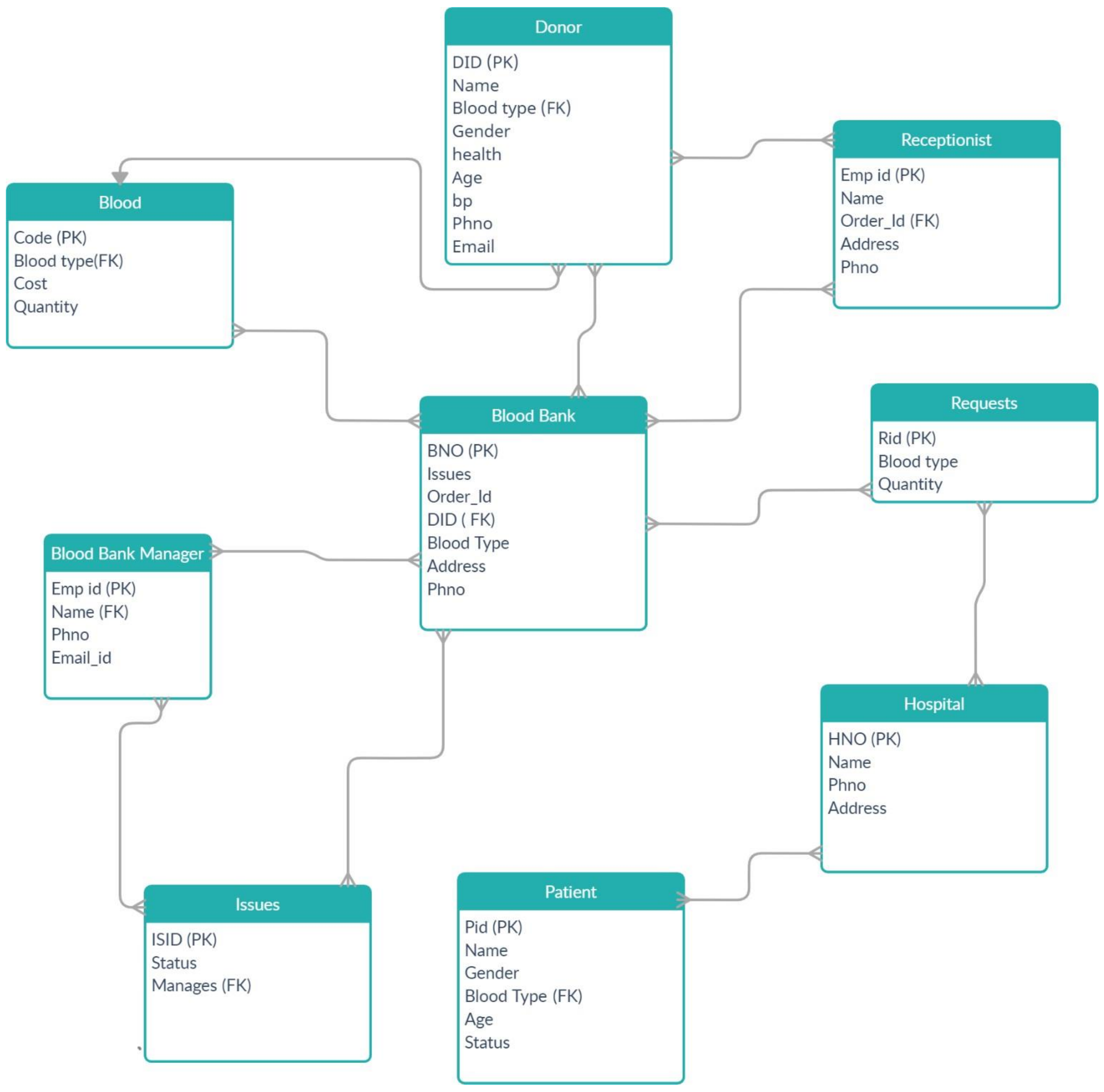
A User can

- Check there is vacancy or not
- See whether his application is approved or not

A Receptionist can

- Check for same blood samples
- Register Donors whose age is greater than 18.
- Can check for blood samples greater than 500 ml.

4. EER MODEL (CONCEPTUAL MODEL)



Normalization

Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.
Normalization is used for mainly two purposes,

- *Eliminating redundant(useless) data.*
- *Ensuring data dependencies make sense i.e data is logically stored.*

Problems Without Normalization

If a table is not properly normalized and have data redundancy then it will not only eat up extra memory space but will also make it difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if database is not normalized. To understand these anomalies let us take an example of a Student table.

Roll no.	***Name***	***Branch***	***HOD***	***Office_con.***
401	*jalam*	*CSE*	*Mr. ram*	*58882*
402	*kalam*	*CSE*	*Mr. ram*	*58882*
403	*donam*	*CSE*	*Mr. ram*	*58882*
404	*mahan*	*CSE*	*Mr. ram*	*58882*
405	*dokon*	*CSE*	*Mr. ram*	*58882*

*In the table above, we have data of 4 Computer Sci. students. As we can see, data for the fields branch, hod and office_con.is repeated for the students who are in the same branch in the college, this is **Data Redundancy.***

Insertion Anomaly

Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as NULL.
Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.

These scenarios are nothing but Insertion anomalies.

Updation Anomaly

What if Mr. X leaves the college? or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency. This is Updation anomaly.

Deletion Anomaly

In our Student table, two different informations are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is Deletion anomaly.

Normalization Rule

Normalization rules are divided into the following normal forms:

1. *First Normal Form*
2. *Second Normal Form*
3. *Third Normal Form*
4. *BCNF*

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. *It should only have single(atomic) valued attributes/columns.*
2. *Values stored in a column should be of the same domain*
3. *All the columns in a table should have unique names.*
4. *And the order in which data is stored, does not matter.*

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. *It should be in the First Normal form.*
2. *And, it should not have Partial Dependency.*

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. *It is in the Second Normal form.*
2. *And, it doesn't have Transitive Dependency.*

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

1. *R must be in 3rd Normal Form*
2. *and, for each functional dependency (X → Y), X should be a super Key.*

Current Database Model:

Donor:

DID	*Name*	*Blood_type*	*Gender*	*Health*	*Age*	*Phno*	*Email*

Blood:

Code	*Did*	*Cost*	*Quantity*

Receptionist:

Empno	*Did*	*Phno*	*Address*	*Bnkno*

Blood bank:

BnkNo	*Did*	*name*	*phno*	*Address*	*issues*

Blood Bank Manager:

Emp_id	*Name*	*Phno*	*Bno*

Hospital:

HNO	*Name*	*Phno*	*Address*	*Request_no*

Issues:

isid	*Status*	*empid*

Patients :

Pid	*Name*	*Gender*	*Blood type*	*Age*	*Hno*

Requests:

Rid	*Bno*	*Quantity*

Checking 1NF :

The necessary conditions for the table to be in 1NF are

1. *It should only have single(atomic) valued attributes/columns.*
2. *Values stored in a column should be of the same domain*
3. *All the columns in a table should have unique names.*
4. *And the order in which data is stored, does not matter.*

Name is the attribute which is conflicting but the data which we have taken are of just simple names that is no first,middle,last names.

Every table is in 1NF in this database.

Checking 2NF :

The necessary conditions for the table to be in 2NF are

1. *It should be in the First Normal form.*
2. *It should not have Partial Dependency.*

Here in all the tables there is only a candidate key in each table so there is no matter of partial dependency.

Checking 3NF :

The necessary conditions for the table to be in 3NF are

1. *It is in the Second Normal form.*
2. *And, it doesn't have Transitive Dependency.*

So, here all tables are in 2NF and now checking Transitive Dependency.
From Donor table,

Donor:

DID	*Name*	*Blood_type*	*Gender*	*Health*	*Age*	*Phno*	*Email*

DID ---> Email
Email ---> Name

Here there is transitive dependency.
To remove it dividing donor table into sub tables.

Donor:

<u>DID</u>	Bloodtype	Gender	Health	Age	Phno	Email

Details:

Email	Name

Checking BCNF :

The necessary conditions for the table to be in BCNF are

1. *R must be in 3rd Normal Form*
2. *for each functional dependency (X → Y), X should be a super Key.*

Firstly, all tables are in 3NF.

Here from details table email is a superkey and DID is a superkey checking for talbes details and donor.

So, All tables are in BCNF.

Finally, The normalised Database model is as follows:

Tables:

Donor:

DID	*Bloodtype*	*Gender*	*Health*	*Age*	*Phno*	*Email*

Details:

Email	*Name*

Receptionist:

Empno	*Did*	*Phno*	*Address*	*Bnkno*

Blood bank:

BnkNo	*Did*	*name*	*phno*	*Address*	*issues*

Blood Bank Manager:

Emp_id	*Name*	*Phno*	*Bno*

Hospital:

HNO	*Name*	*Phno*	*Address*	*Request_no*

Issues:

isid	*Status*	*empid*

Patients :

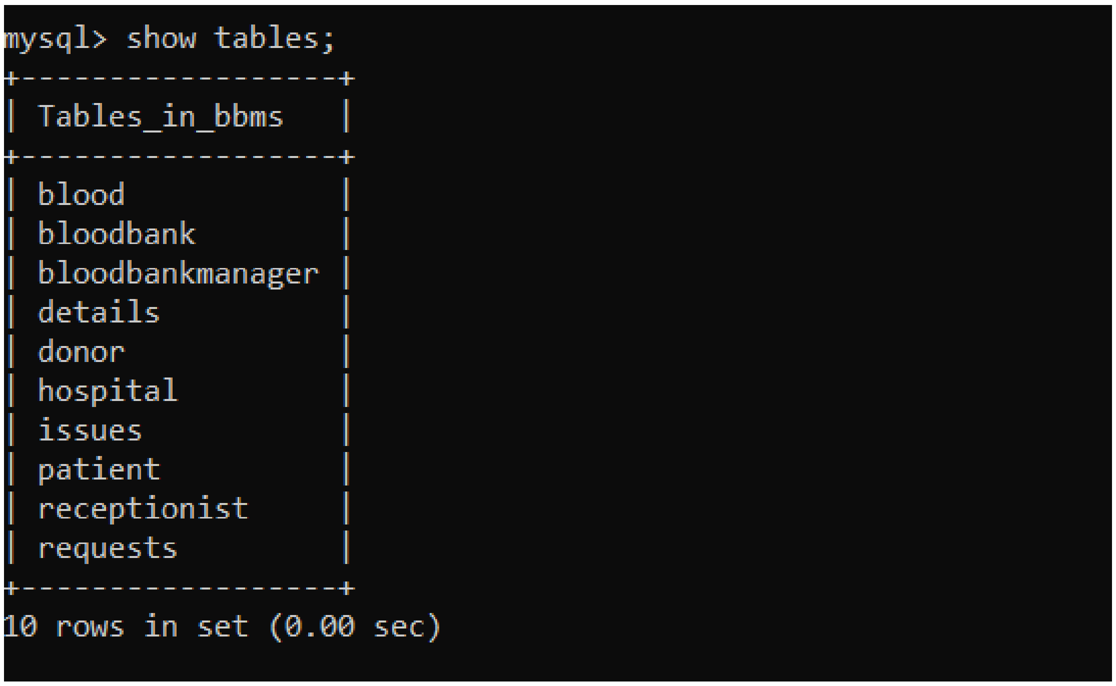
Pid	*Name*	*Gender*	*Blood type*	*Age*	*Hno*

Requests:

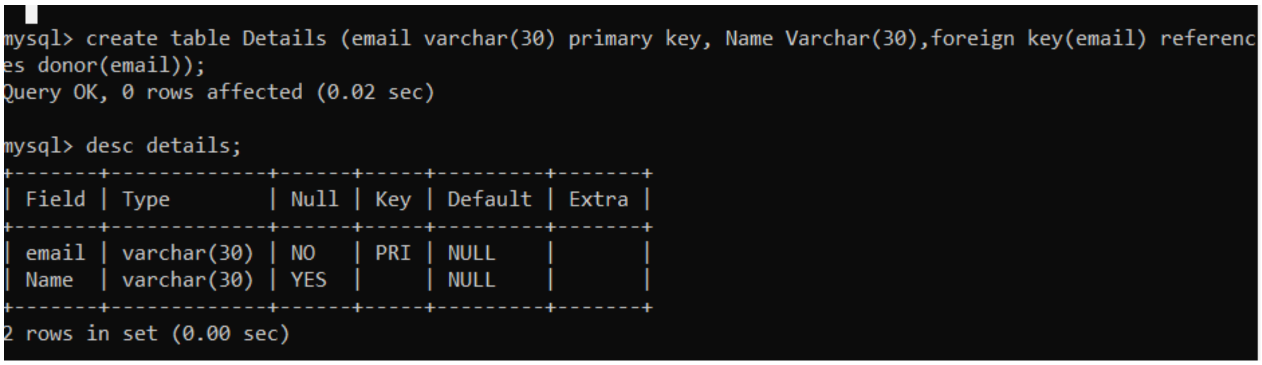
Rid	*Bno*	*Quantity*

Database is created as bbms.

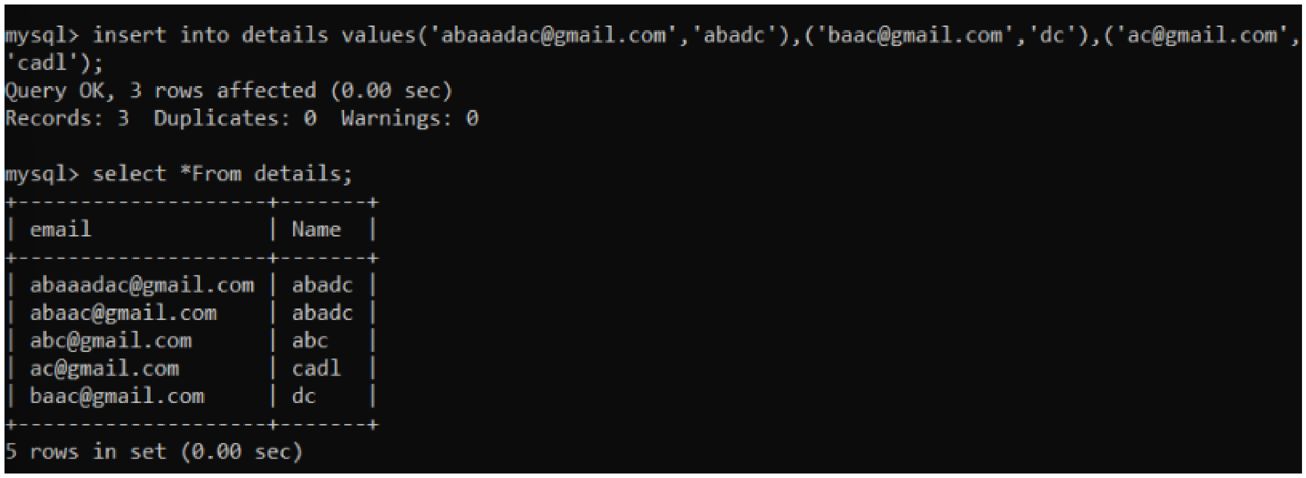
And tables are

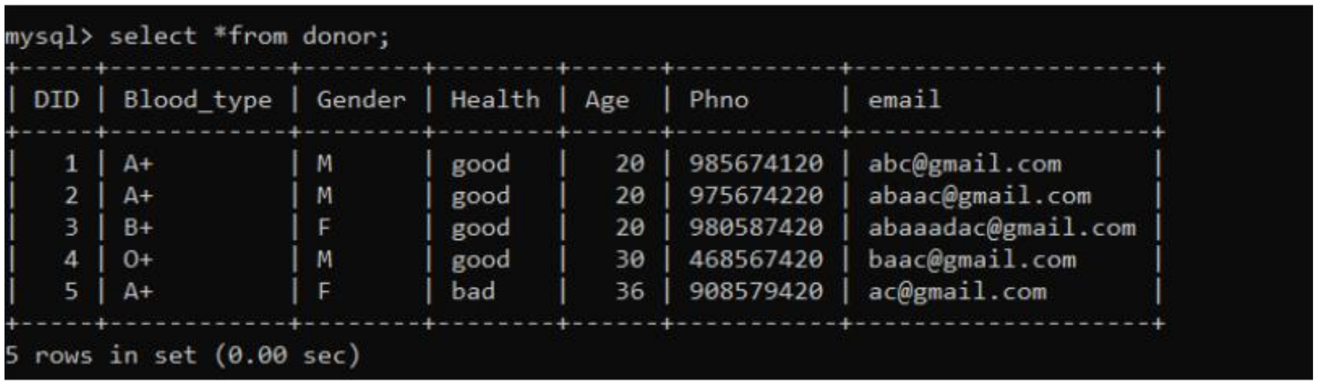


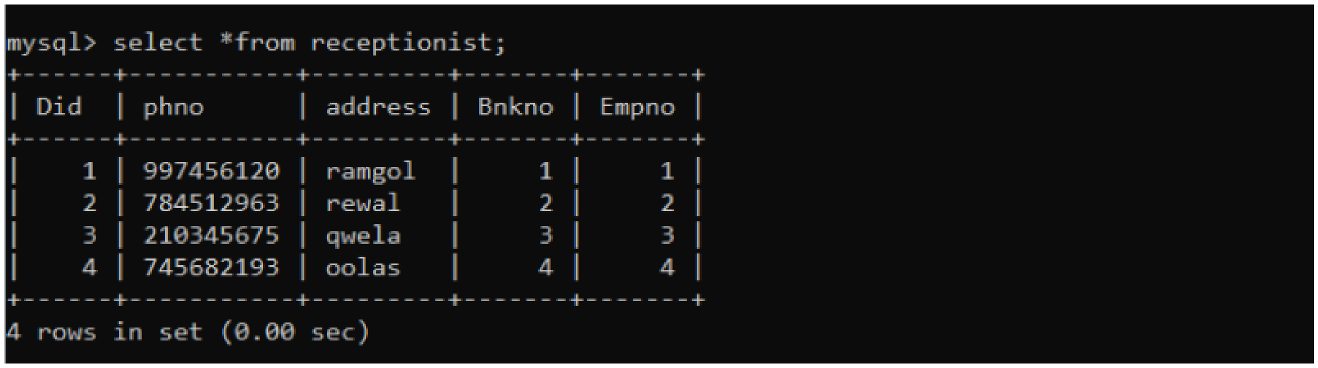
```
mysql> show tables;
+------------------+
| Tables_in_bbms   |
+------------------+
| blood            |
| bloodbank        |
| bloodbankmanager |
| details          |
| donor            |
| hospital         |
| issues           |
| patient          |
| receptionist     |
| requests         |
+------------------+
10 rows in set (0.00 sec)
```

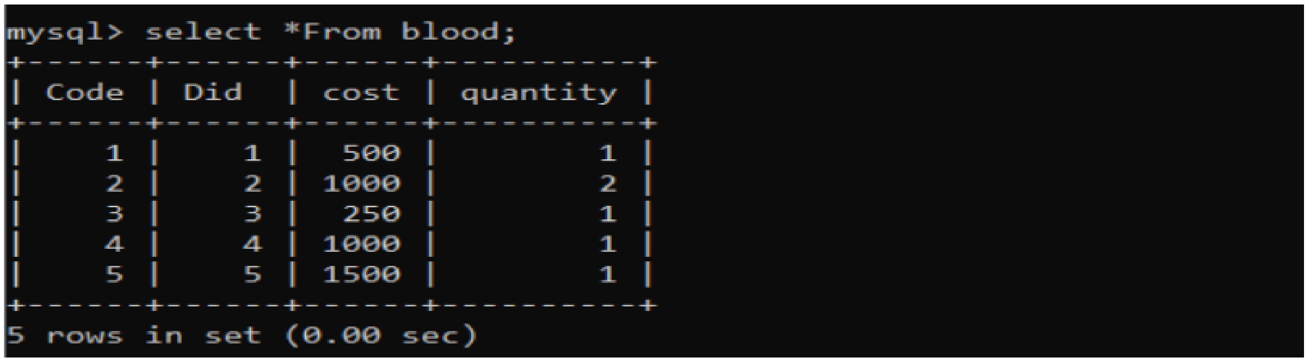


```
mysql> create table Details (email varchar(30) primary key, Name Varchar(30),foreign key(email) referenc
es donor(email));
Query OK, 0 rows affected (0.02 sec)

mysql> desc details;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| email | varchar(30) | NO   | PRI | NULL    |       |
| Name  | varchar(30) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

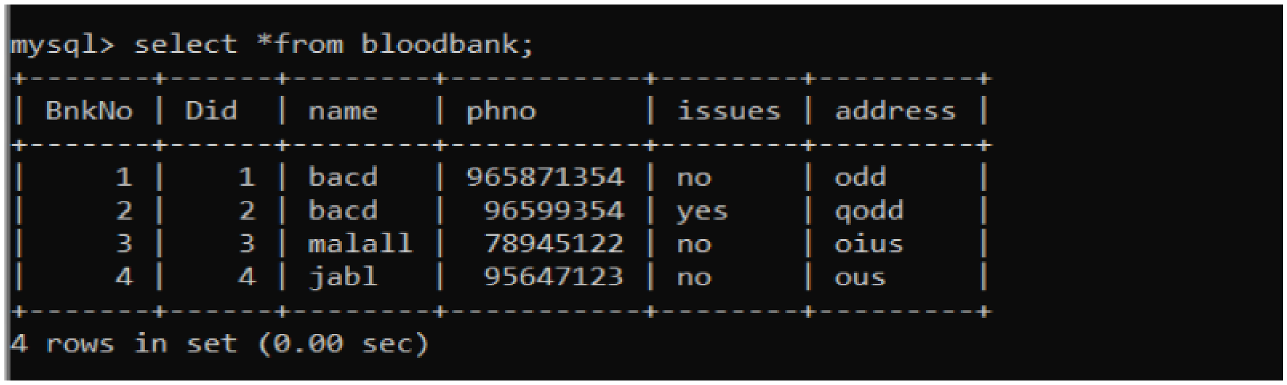

```
mysql> insert into details values('abaaadac@gmail.com','abadc'),('baac@gmail.com','dc'),('ac@gmail.com',
'cadl');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select *From details;
+--------------------+-------+
| email              | Name  |
+--------------------+-------+
| abaaadac@gmail.com | abadc |
| abaac@gmail.com    | abadc |
| abc@gmail.com      | abc   |
| ac@gmail.com       | cadl  |
| baac@gmail.com     | dc    |
+--------------------+-------+
5 rows in set (0.00 sec)
```


```
mysql> select *from donor;
+-----+------------+--------+--------+------+-----------+--------------------+
| DID | Blood_type | Gender | Health | Age  | Phno      | email              |
+-----+------------+--------+--------+------+-----------+--------------------+
|   1 | A+         | M      | good   |   20 | 985674120 | abc@gmail.com      |
|   2 | A+         | M      | good   |   20 | 975674220 | abaac@gmail.com    |
|   3 | B+         | F      | good   |   20 | 980587420 | abaaadac@gmail.com |
|   4 | O+         | M      | good   |   30 | 468567420 | baac@gmail.com     |
|   5 | A+         | F      | bad    |   36 | 908579420 | ac@gmail.com       |
+-----+------------+--------+--------+------+-----------+--------------------+
5 rows in set (0.00 sec)
```

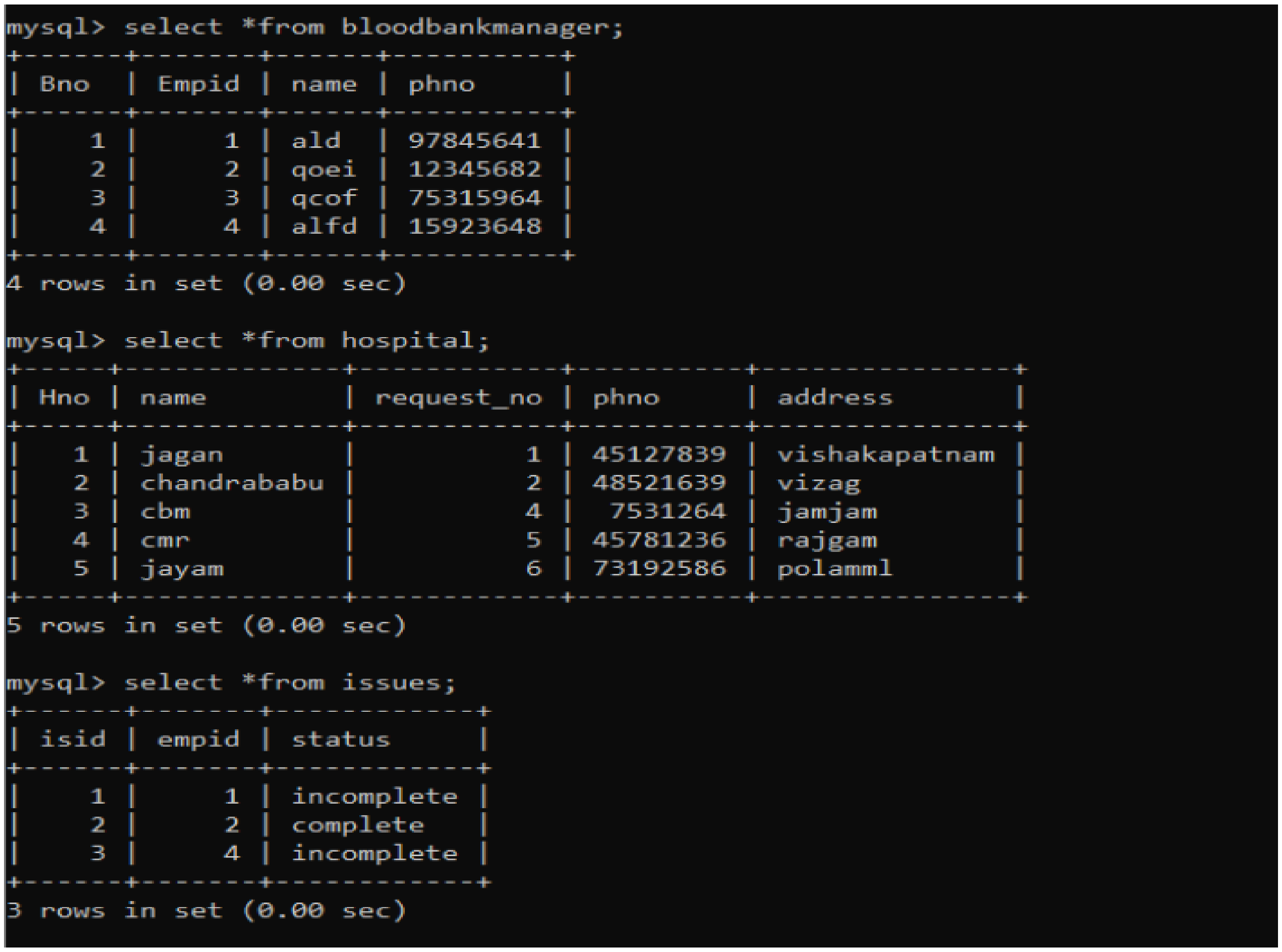

```
mysql> select *from receptionist;
+------+-----------+---------+-------+-------+
| Did  | phno      | address | Bnkno | Empno |
+------+-----------+---------+-------+-------+
|    1 | 997456120 | ramgol  |     1 |     1 |
|    2 | 784512963 | rewal   |     2 |     2 |
|    3 | 210345675 | qwela   |     3 |     3 |
|    4 | 745682193 | oolas   |     4 |     4 |
+------+-----------+---------+-------+-------+
4 rows in set (0.00 sec)
```



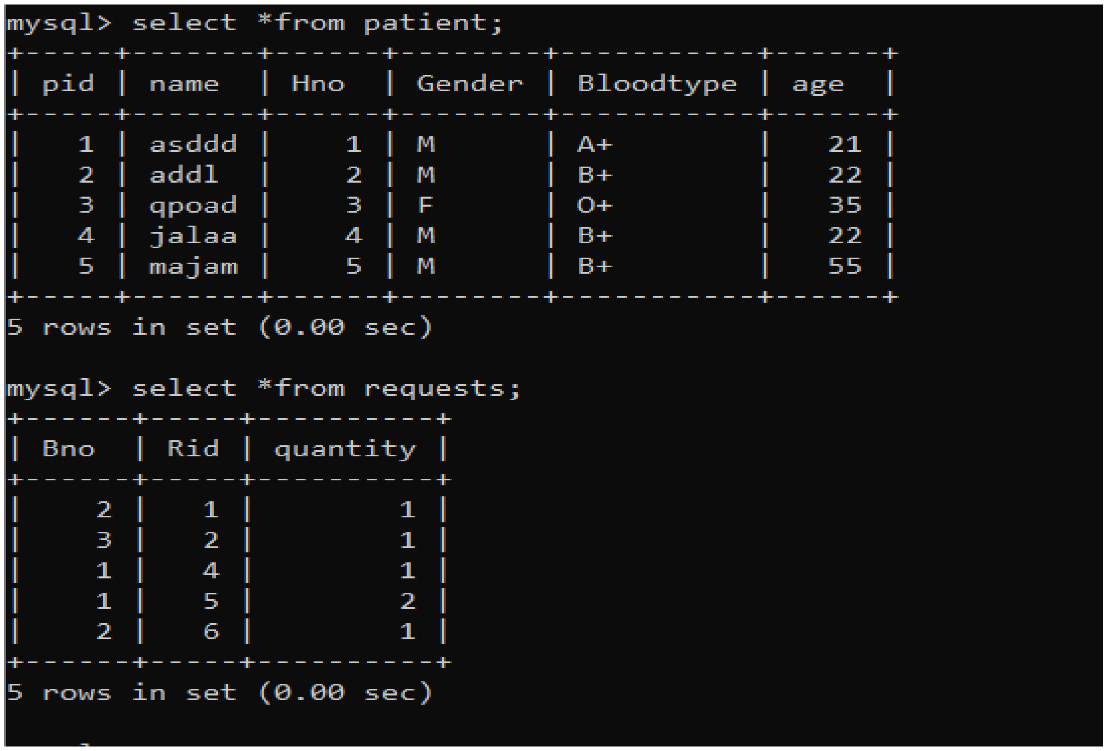
```
mysql> select *From blood;
+------+------+------+----------+
| Code | Did  | cost | quantity |
+------+------+------+----------+
|    1 |    1 |  500 |        1 |
|    2 |    2 | 1000 |        2 |
|    3 |    3 |  250 |        1 |
|    4 |    4 | 1000 |        1 |
|    5 |    5 | 1500 |        1 |
+------+------+------+----------+
5 rows in set (0.00 sec)
```



```
mysql> select *from bloodbank;
+-------+------+--------+-----------+--------+---------+
| BnkNo | Did  | name   | phno      | issues | address |
+-------+------+--------+-----------+--------+---------+
|     1 |    1 | bacd   | 965871354 | no     | odd     |
|     2 |    2 | bacd   |  96599354 | yes    | qodd    |
|     3 |    3 | malall |  78945122 | no     | oius    |
|     4 |    4 | jabl   |  95647123 | no     | ous     |
+-------+------+--------+-----------+--------+---------+
4 rows in set (0.00 sec)
```

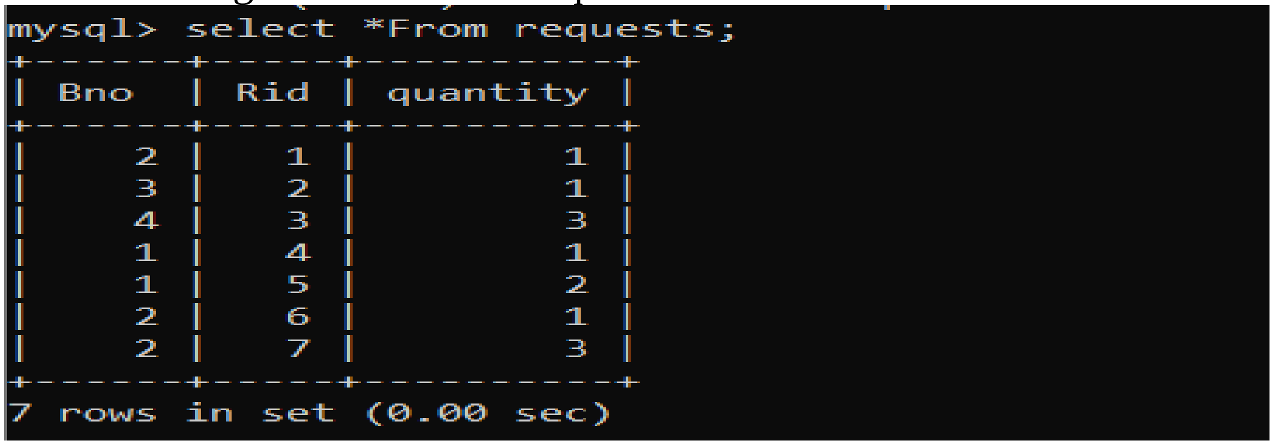


```
mysql> select *from bloodbankmanager;
+------+-------+------+----------+
| Bno  | Empid | name | phno     |
+------+-------+------+----------+
|    1 |     1 | ald  | 97845641 |
|    2 |     2 | qoei | 12345682 |
|    3 |     3 | qcof | 75315964 |
|    4 |     4 | alfd | 15923648 |
+------+-------+------+----------+
4 rows in set (0.00 sec)

mysql> select *from hospital;
+-----+-------------+------------+----------+---------------+
| Hno | name        | request_no | phno     | address       |
+-----+-------------+------------+----------+---------------+
|   1 | jagan       |          1 | 45127839 | vishakapatnam |
|   2 | chandrababu |          2 | 48521639 | vizag         |
|   3 | cbm         |          4 |  7531264 | jamjam        |
|   4 | cmr         |          5 | 45781236 | rajgam        |
|   5 | jayam       |          6 | 73192586 | polamml       |
+-----+-------------+------------+----------+---------------+
5 rows in set (0.00 sec)

mysql> select *from issues;
+------+-------+------------+
| isid | empid | status     |
+------+-------+------------+
|    1 |     1 | incomplete |
|    2 |     2 | complete   |
|    3 |     4 | incomplete |
+------+-------+------------+
3 rows in set (0.00 sec)
```



```
mysql> select *from patient;
+-----+--------+-----+--------+-----------+------+
| pid | name   | Hno | Gender | Bloodtype | age  |
+-----+--------+-----+--------+-----------+------+
|   1 | asddd  |   1 | M      | A+        |   21 |
|   2 | addl   |   2 | M      | B+        |   22 |
|   3 | qpoad  |   3 | F      | O+        |   35 |
|   4 | jalaa  |   4 | M      | B+        |   22 |
|   5 | majam  |   5 | M      | B+        |   55 |
+-----+--------+-----+--------+-----------+------+
5 rows in set (0.00 sec)

mysql> select *from requests;
+------+-----+----------+
| Bno  | Rid | quantity |
+------+-----+----------+
|    2 |   1 |        1 |
|    3 |   2 |        1 |
|    1 |   4 |        1 |
|    1 |   5 |        2 |
|    2 |   6 |        1 |
+------+-----+----------+
5 rows in set (0.00 sec)
```

After inserting some values in requests:

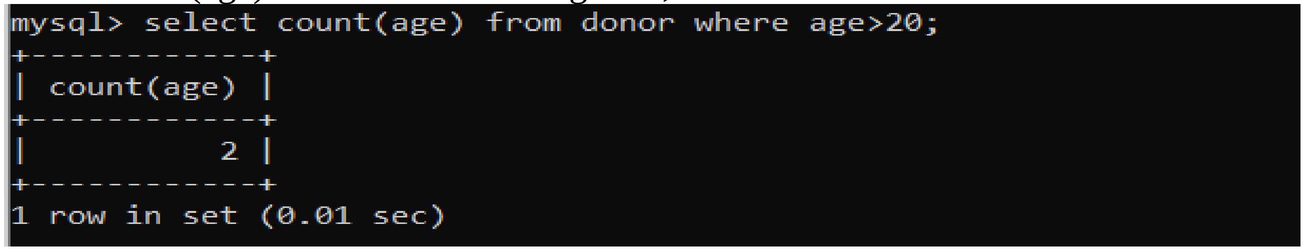


```
mysql> select *From requests;
+------+-----+----------+
| Bno  | Rid | quantity |
+------+-----+----------+
|    2 |   1 |        1 |
|    3 |   2 |        1 |
|    4 |   3 |        3 |
|    1 |   4 |        1 |
|    1 |   5 |        2 |
|    2 |   6 |        1 |
|    2 |   7 |        3 |
+------+-----+----------+
7 rows in set (0.00 sec)
```

Queries:

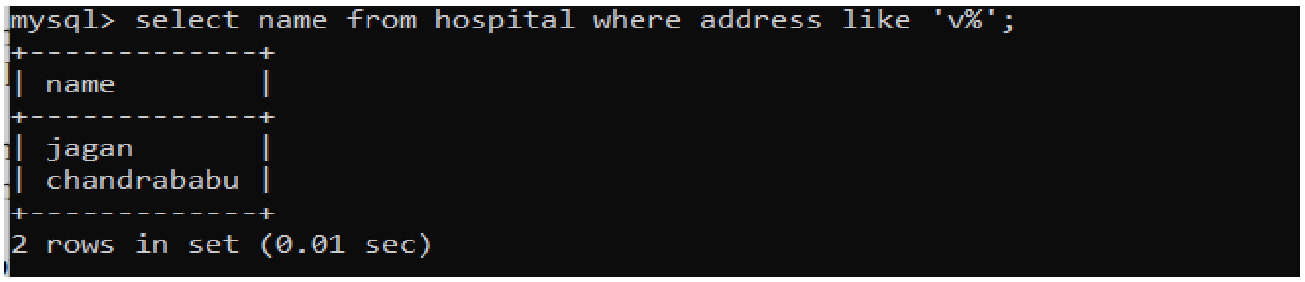
Simple Queries:

To display total number of donors who have age >20 Query:
 select count(age) from donor where age>20;



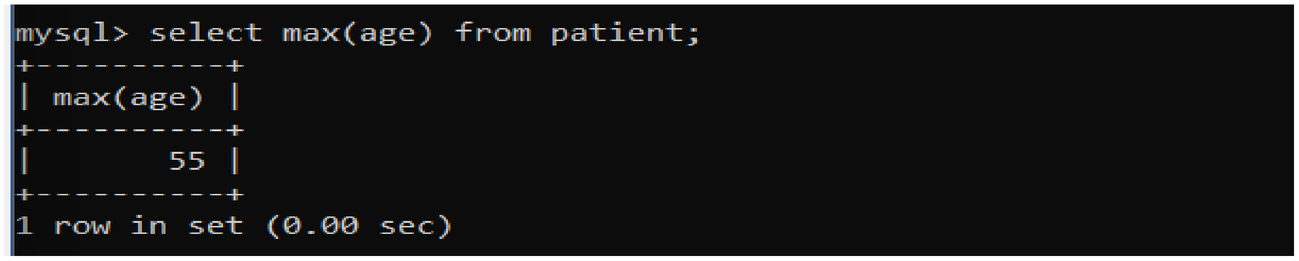
```
mysql> select count(age) from donor where age>20;
+------------+
| count(age) |
+------------+
|          2 |
+------------+
1 row in set (0.01 sec)
```

To display hospital names whose address starts with letter v
Query: select name from hospital where address like'v%';

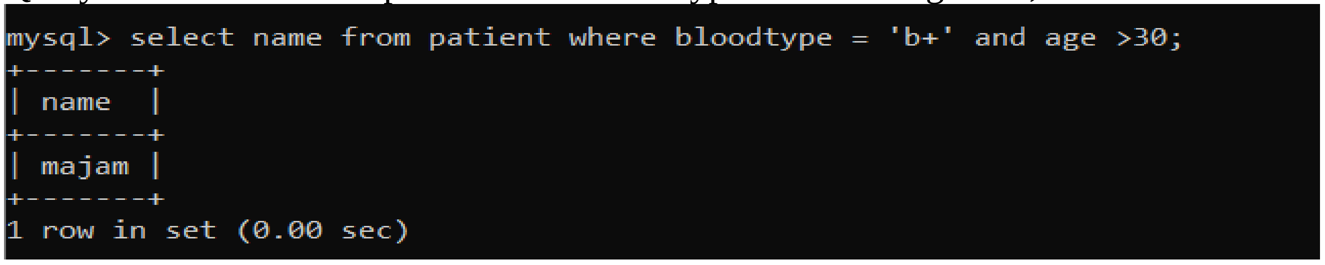


```
mysql> select name from hospital where address like 'v%';
+-------------+
| name        |
+-------------+
| jagan       |
| chandrababu |
+-------------+
2 rows in set (0.01 sec)
```

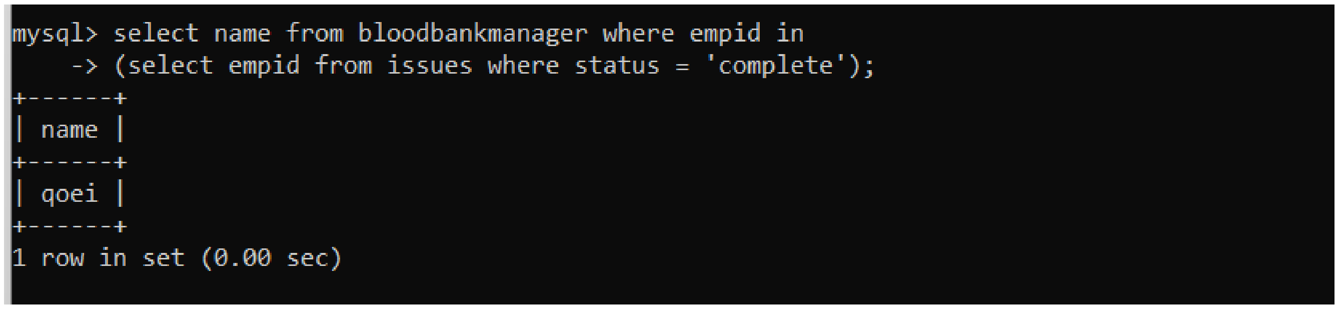
To display age of patient who is oldest
Query: select max(age) from patient;


```
mysql> select max(age) from patient;
+----------+
| max(age) |
+----------+
|       55 |
+----------+
1 row in set (0.00 sec)
```

To display Names of patients whose blood group is b+ and age is > 30
Query: select name from patient where bloodtype = 'b+' and age >30;

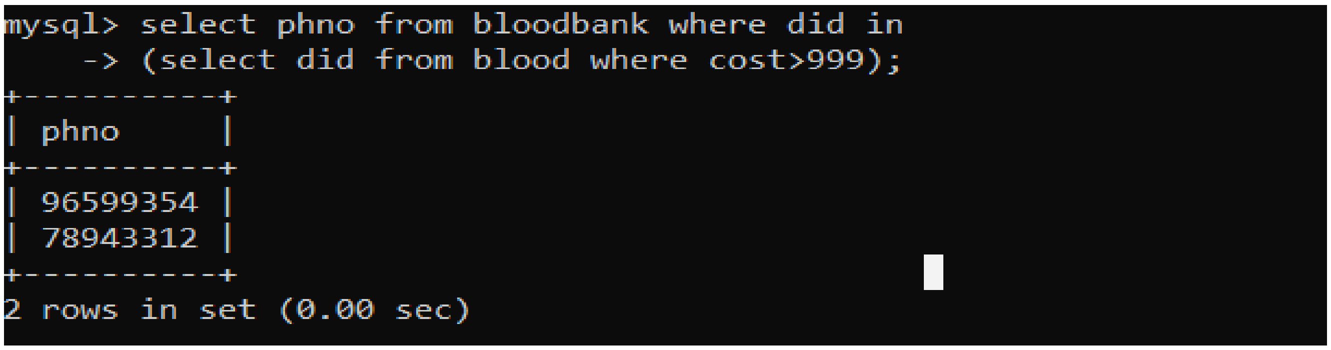

```
mysql> select name from patient where bloodtype = 'b+' and age >30;
+-------+
| name  |
+-------+
| majam |
+-------+
1 row in set (0.00 sec)
```

To display Name of the manager who solved the issue
Query : select name from bloodbankmanager where empid in
(select empid from issues where status = 'complete');

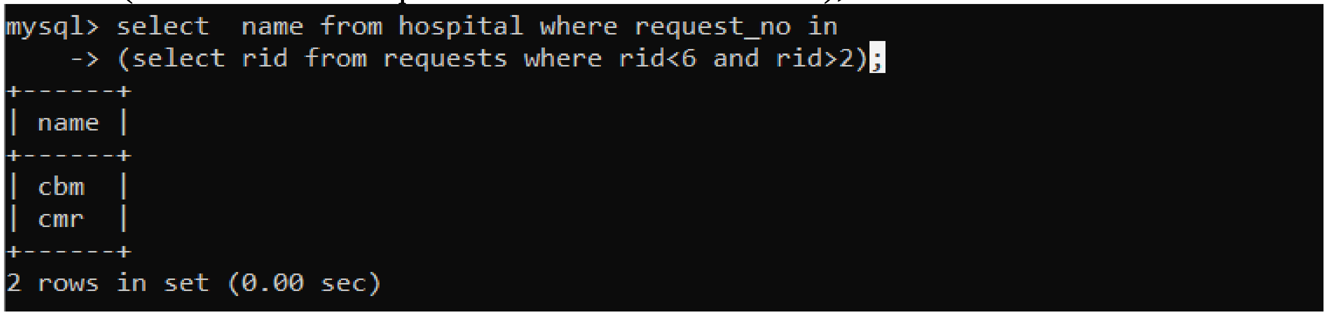

```
mysql> select name from bloodbankmanager where empid in
    -> (select empid from issues where status = 'complete');
+------+
| name |
+------+
| qoei |
+------+
1 row in set (0.00 sec)
```

Complex Queries:

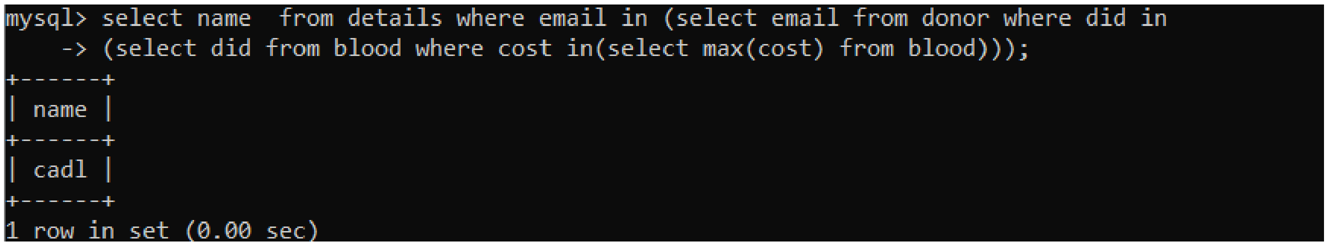
To display phone of the donor who paid more than 999 rupees;
Query: select phno from bloodbank where did in
(select did from blood where cost>999)


```
mysql> select phno from bloodbank where did in
    -> (select did from blood where cost>999);
+----------+
| phno     |
+----------+
| 96599354 |
| 78943312 |
+----------+
2 rows in set (0.00 sec)
```

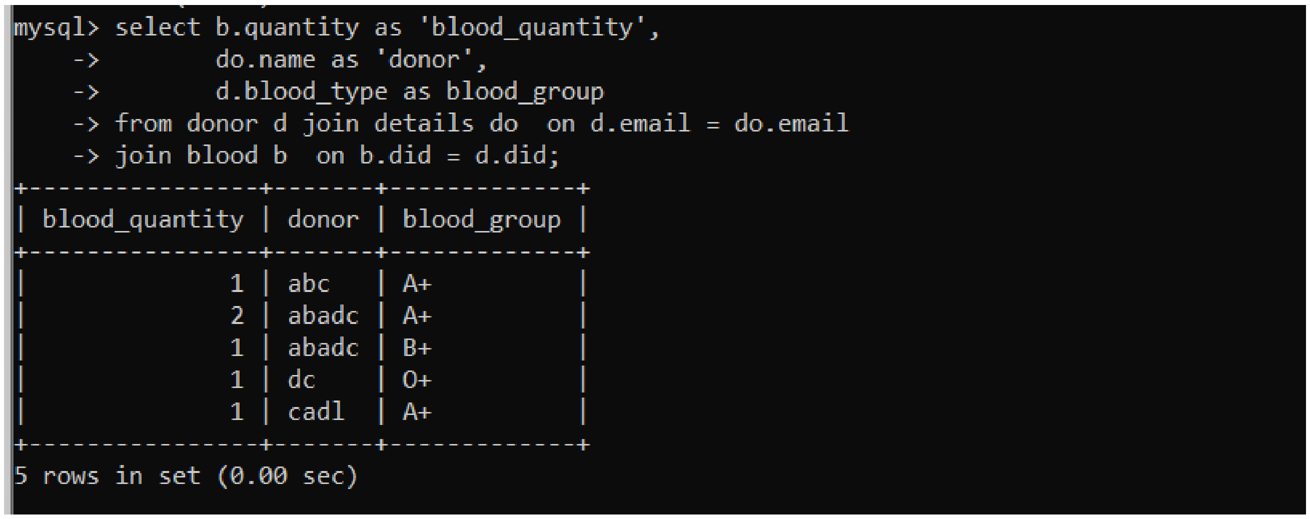
To display name of the hospital where the request id are in between 2 and 6 Query:
select name from hospital where request_no in
(select rid from requests where rid<6 and rid>2);


```
mysql> select  name from hospital where request_no in
    -> (select rid from requests where rid<6 and rid>2);
+------+
| name |
+------+
| cbm  |
| cmr  |
+------+
2 rows in set (0.00 sec)
```

To display name of the donor where his blood cost is highest.
Query: select name from details where email in (select email from donor where did in
(select did from blood where cost in(select max(cost) from blood)));

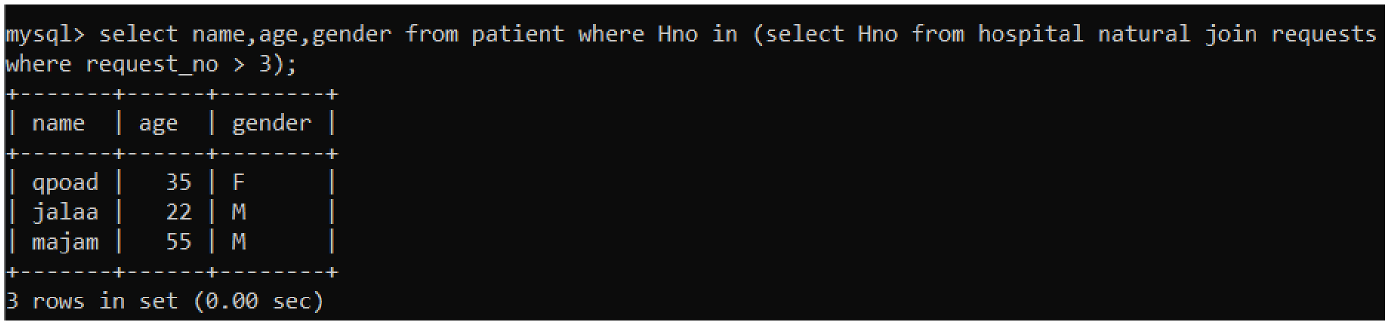

```
mysql> select name  from details where email in (select email from donor where did in
    -> (select did from blood where cost in(select max(cost) from blood)));
+------+
| name |
+------+
| cadl |
+------+
1 row in set (0.00 sec)
```

To display to bloodgroup,blood quantity (sample taken by) and donor name. Query :
select b.quantity as 'blood_quantity',
do.name as 'donor',
d.blood_type as blood_group
from donor d join details do on d.email = do.email join
blood b on b.did = d.did;



```
mysql> select b.quantity as 'blood_quantity',
    ->        do.name as 'donor',
    ->        d.blood_type as blood_group
    -> from donor d join details do  on d.email = do.email
    -> join blood b  on b.did = d.did;
+----------------+-------+-------------+
| blood_quantity | donor | blood_group |
+----------------+-------+-------------+
|              1 | abc   | A+          |
|              2 | abadc | A+          |
|              1 | abadc | B+          |
|              1 | dc    | O+          |
|              1 | cadl  | A+          |
+----------------+-------+-------------+
5 rows in set (0.00 sec)
```

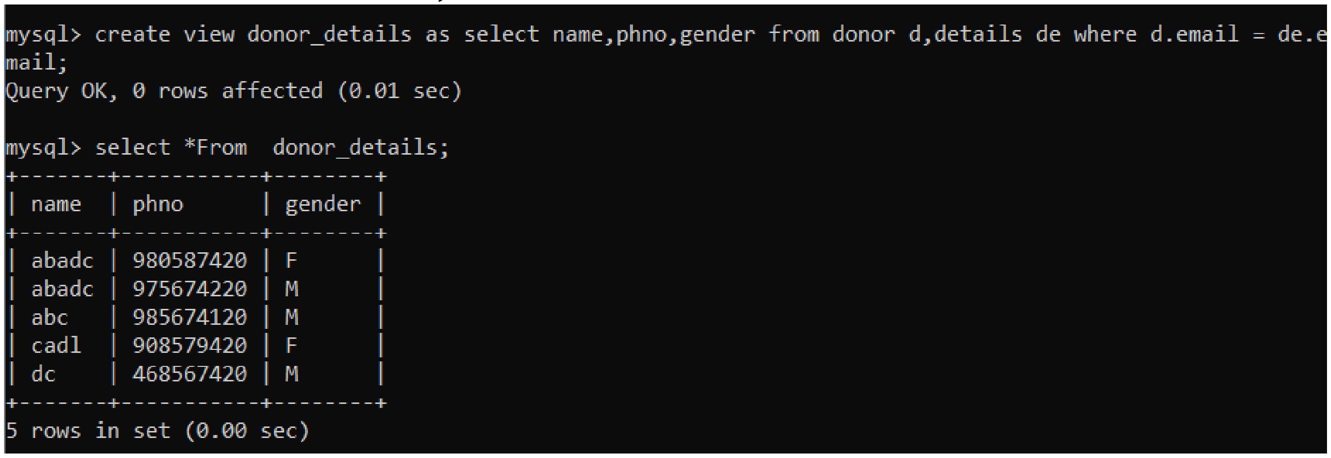
To display name,age,gender of patient whose hospital request_no are greater than 2.
Query: select name,age,gender from patient where Hno in
(select Hno from hospital natural join requests where request_no >
3);



```
mysql> select name,age,gender from patient where Hno in (select Hno from hospital natural join requests
where request_no > 3);
+-------+------+--------+
| name  | age  | gender |
+-------+------+--------+
| qpoad |   35 | F      |
| jalaa |   22 | M      |
| majam |   55 | M      |
+-------+------+--------+
3 rows in set (0.00 sec)
```

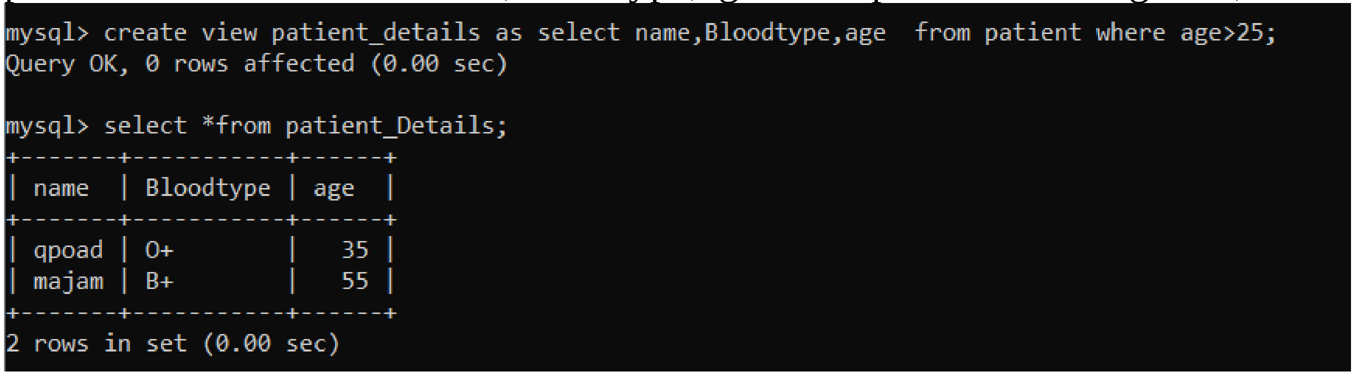
Views:

Create a view donor_details which has name, phno,gender of donor. create view donor_details as select name,phno,gender from donor d,details de where d.email = de.email;



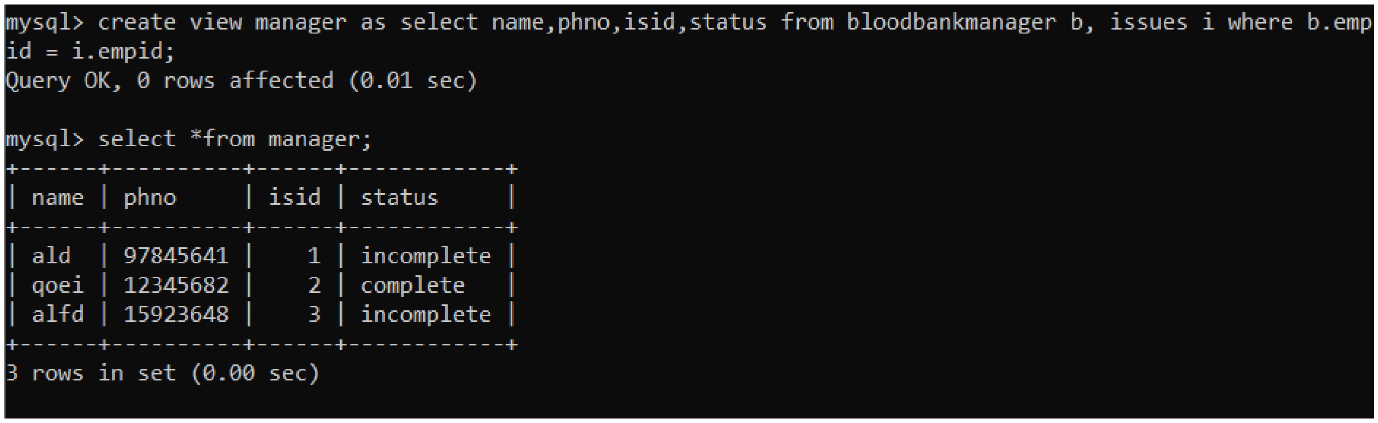
```
mysql> create view donor_details as select name,phno,gender from donor d,details de where d.email = de.e
mail;
Query OK, 0 rows affected (0.01 sec)

mysql> select *From  donor_details;
+-------+-----------+--------+
| name  | phno      | gender |
+-------+-----------+--------+
| abadc | 980587420 | F      |
| abadc | 975674220 | M      |
| abc   | 985674120 | M      |
| cadl  | 908579420 | F      |
| dc    | 468567420 | M      |
+-------+-----------+--------+
5 rows in set (0.00 sec)
```

Create a view patient_details with name,bloodgroup,age and to make confidential of patient age who are having age less than 25 create view patient_details as select name,Bloodtype,age from patient where age>25;



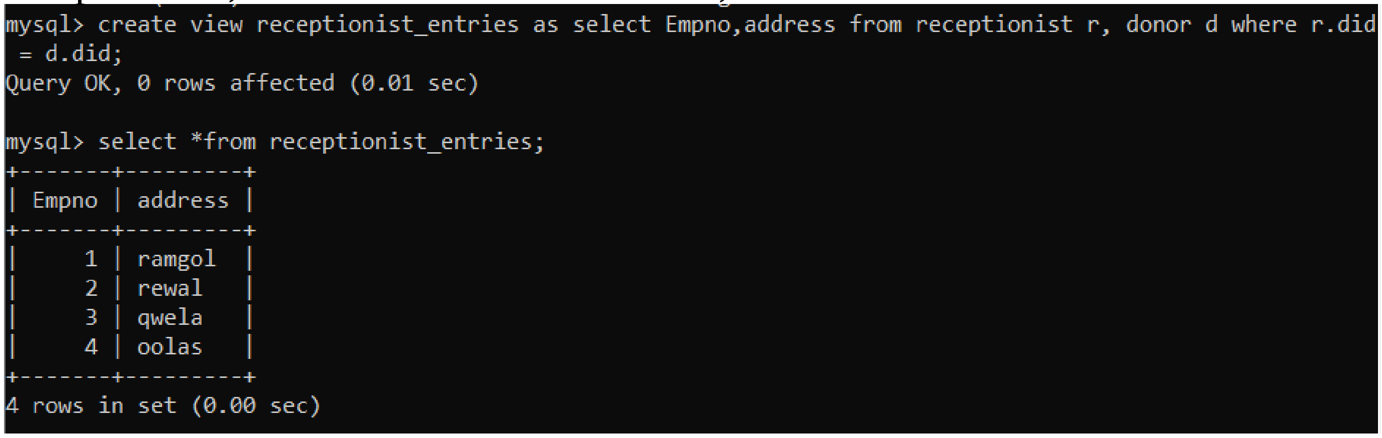
```
mysql> create view patient_details as select name,Bloodtype,age  from patient where age>25;
Query OK, 0 rows affected (0.00 sec)

mysql> select *from patient_Details;
+-------+-----------+------+
| name  | Bloodtype | age  |
+-------+-----------+------+
| qpoad | O+        |   35 |
| majam | B+        |   55 |
+-------+-----------+------+
2 rows in set (0.00 sec)
```

Create a view manager with name,phno,isid,status of issue whether it is completed or not.
create view manager as select name,phno,isid,status from bloodbankmanager b, issues i where b.empid = i.empid;


```
mysql> create view manager as select name,phno,isid,status from bloodbankmanager b, issues i where b.emp
id = i.empid;
Query OK, 0 rows affected (0.01 sec)

mysql> select *from manager;
+------+----------+------+------------+
| name | phno     | isid | status     |
+------+----------+------+------------+
| ald  | 97845641 |    1 | incomplete |
| qoei | 12345682 |    2 | complete   |
| alfd | 15923648 |    3 | incomplete |
+------+----------+------+------------+
3 rows in set (0.00 sec)
```

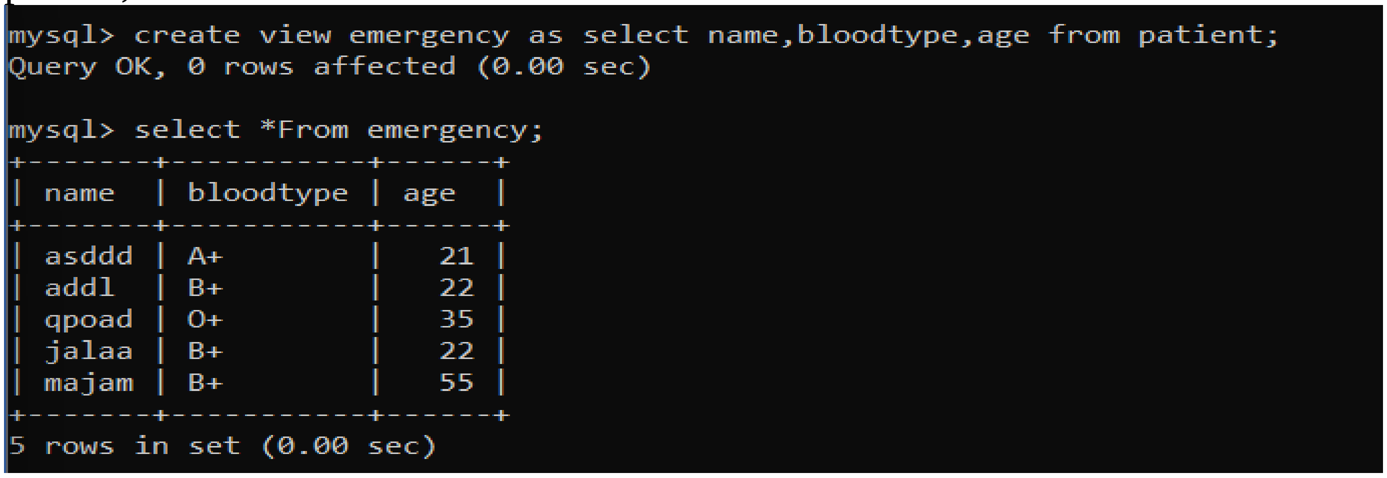
Create a view receptionist_entries with only donor address details with respective employee id.

create view receptionist_entries as select Empno,address from receptionist r, donor d where r.did = d.did;


```
mysql> create view receptionist_entries as select Empno,address from receptionist r, donor d where r.did
 = d.did;
Query OK, 0 rows affected (0.01 sec)

mysql> select *from receptionist_entries;
+-------+---------+
| Empno | address |
+-------+---------+
|     1 | ramgol  |
|     2 | rewal   |
|     3 | qwela   |
|     4 | oolas   |
+-------+---------+
4 rows in set (0.00 sec)
```

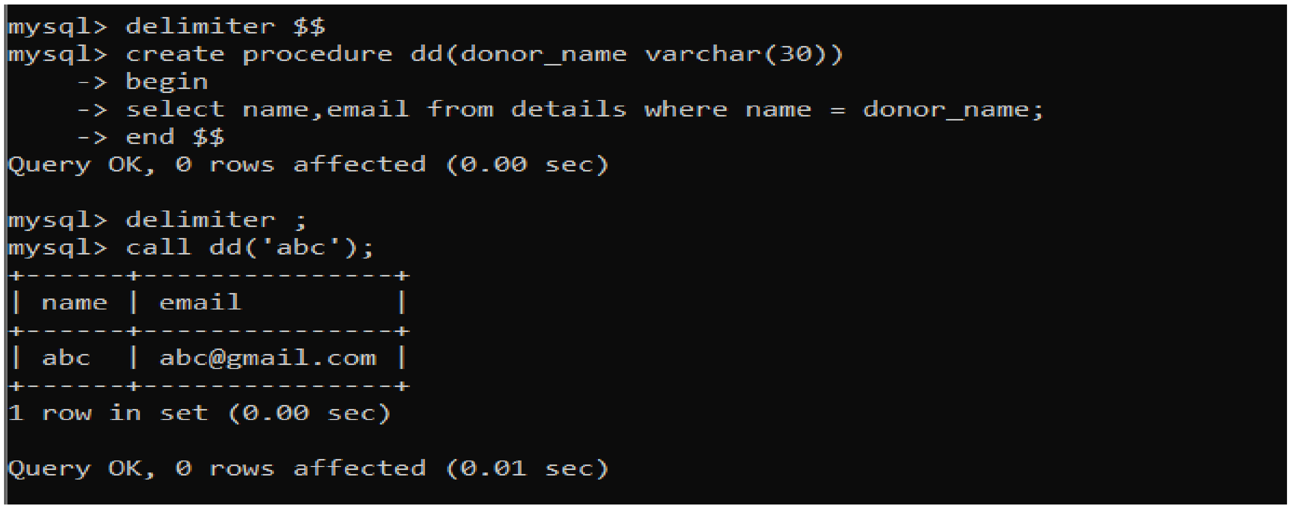
Create view emergency details of name,bloodtype,age of patients.

create view emergency as select name,bloodtype,age from patient;


```
mysql> create view emergency as select name,bloodtype,age from patient;
Query OK, 0 rows affected (0.00 sec)

mysql> select *From emergency;
+-------+-----------+------+
| name  | bloodtype | age  |
+-------+-----------+------+
| asddd | A+        |   21 |
| addl  | B+        |   22 |
| qpoad | O+        |   35 |
| jalaa | B+        |   22 |
| majam | B+        |   55 |
+-------+-----------+------+
5 rows in set (0.00 sec)
```

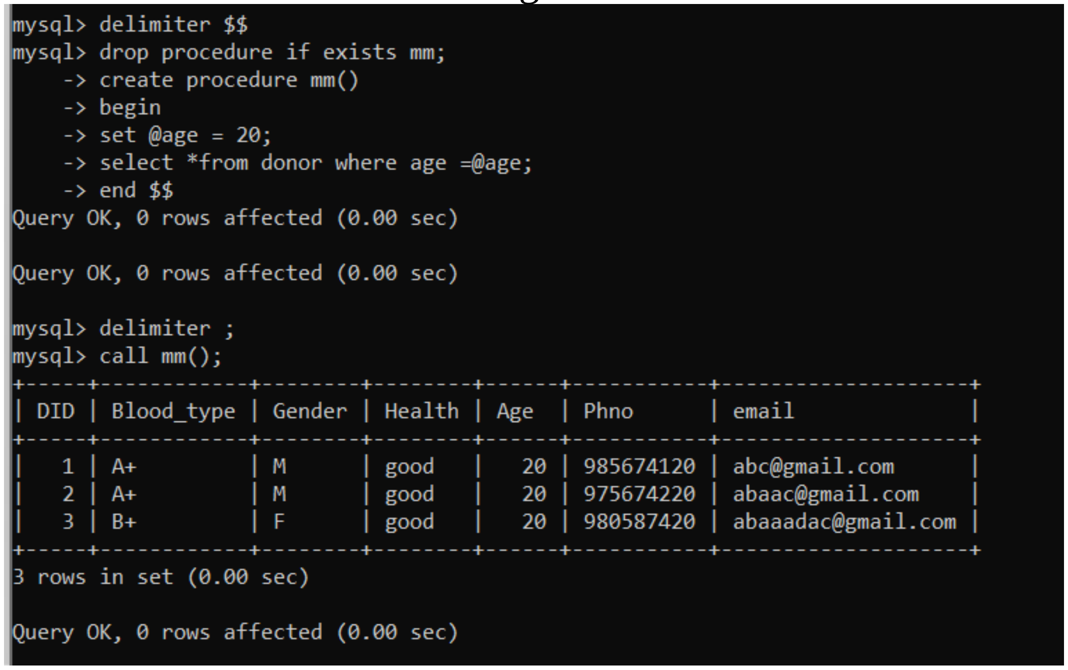
Stored Procedures:

To check email of a donor.



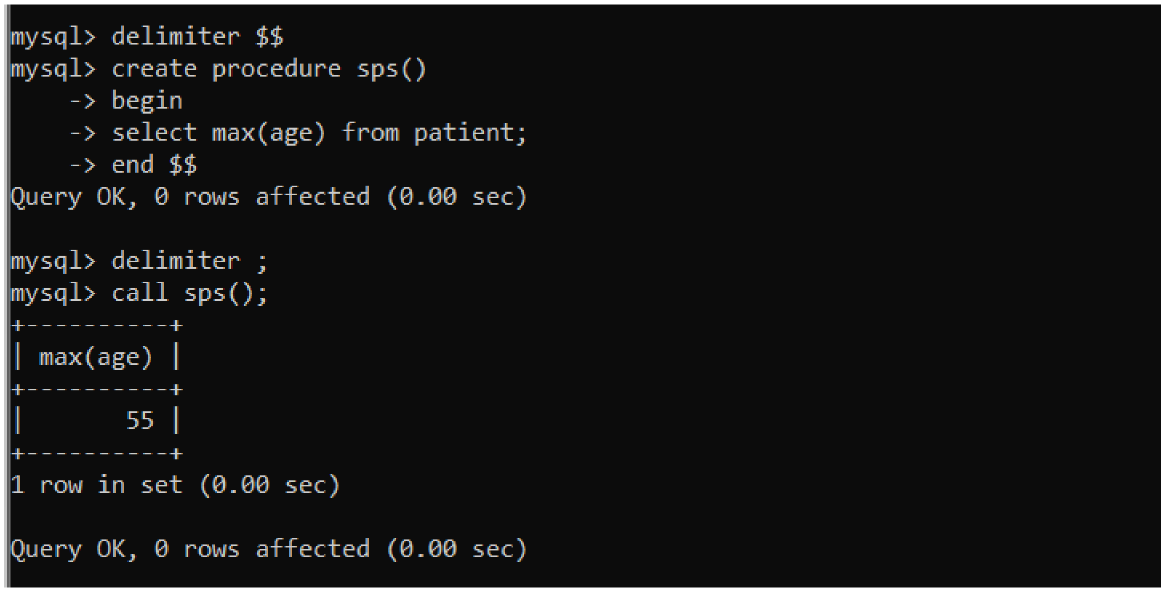
```
mysql> delimiter $$
mysql> create procedure dd(donor_name varchar(30))
    -> begin
    -> select name,email from details where name = donor_name;
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call dd('abc');
+------+---------------+
| name | email         |
+------+---------------+
| abc  | abc@gmail.com |
+------+---------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

To find details of donor with age =20.



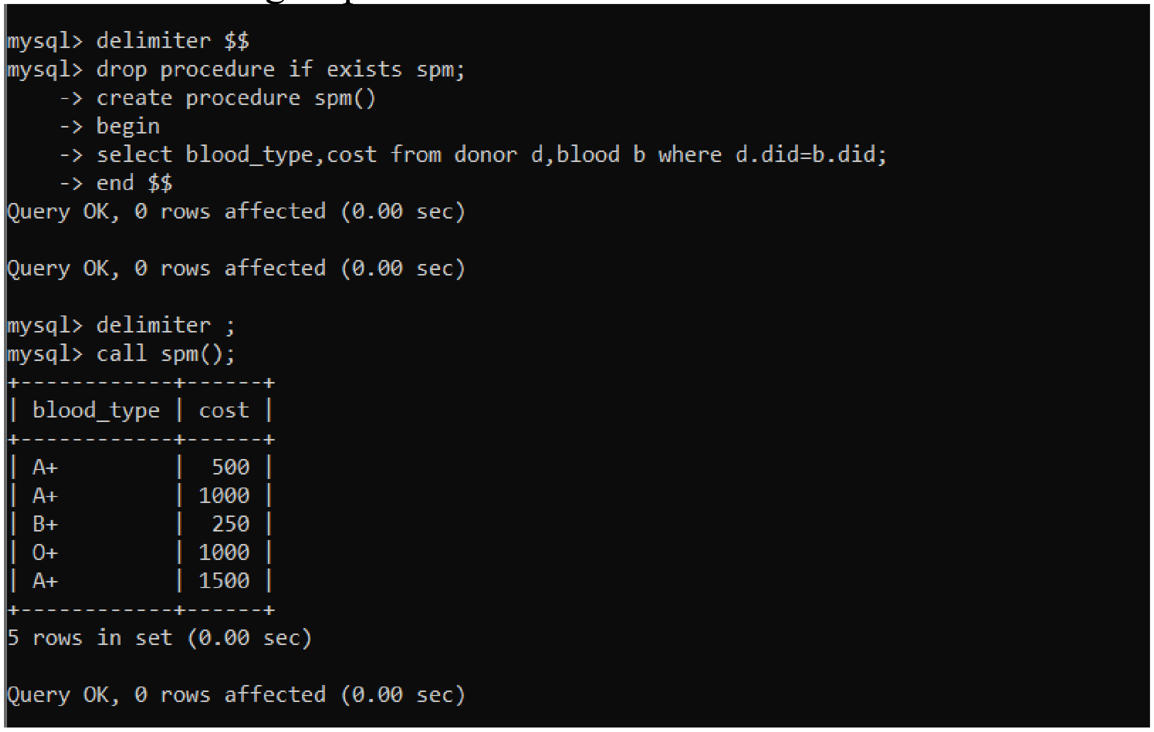
```
mysql> delimiter $$
mysql> drop procedure if exists mm;
    -> create procedure mm()
    -> begin
    -> set @age = 20;
    -> select *from donor where age =@age;
    -> end $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call mm();
+-----+------------+--------+--------+------+-----------+--------------------+
| DID | Blood_type | Gender | Health | Age  | Phno      | email              |
+-----+------------+--------+--------+------+-----------+--------------------+
|   1 | A+         | M      | good   |   20 | 985674120 | abc@gmail.com      |
|   2 | A+         | M      | good   |   20 | 975674220 | abaac@gmail.com    |
|   3 | B+         | F      | good   |   20 | 980587420 | abaaadac@gmail.com |
+-----+------------+--------+--------+------+-----------+--------------------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

To find the oldest patient age.



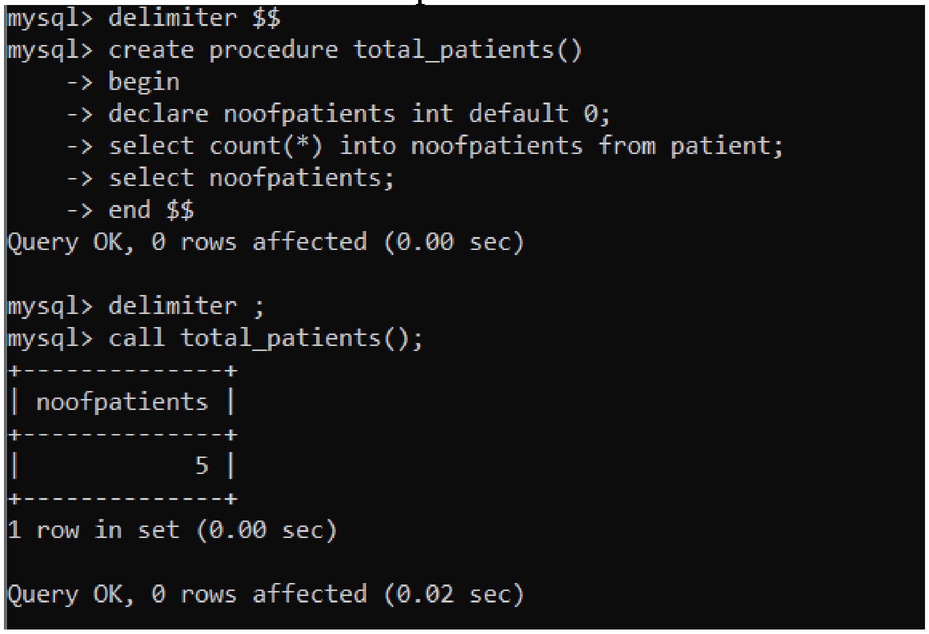
```
mysql> delimiter $$
mysql> create procedure sps()
    -> begin
    -> select max(age) from patient;
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call sps();
+----------+
| max(age) |
+----------+
|       55 |
+----------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

To know bloodgroup and it’s cost.



```
mysql> delimiter $$
mysql> drop procedure if exists spm;
    -> create procedure spm()
    -> begin
    -> select blood_type,cost from donor d,blood b where d.did=b.did;
    -> end $$
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call spm();
+------------+------+
| blood_type | cost |
+------------+------+
| A+         |  500 |
| A+         | 1000 |
| B+         |  250 |
| O+         | 1000 |
| A+         | 1500 |
+------------+------+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

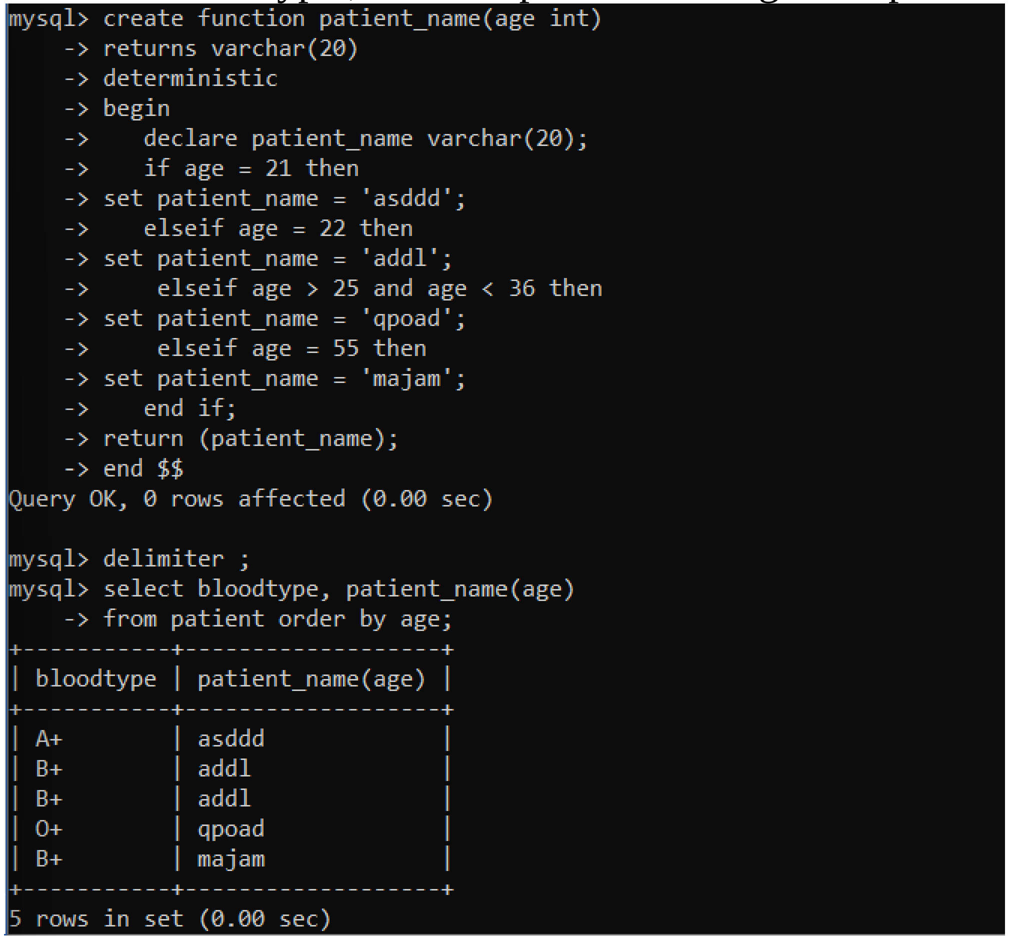
To find total number of patients who need blood.



```
mysql> delimiter $$
mysql> create procedure total_patients()
    -> begin
    -> declare noofpatients int default 0;
    -> select count(*) into noofpatients from patient;
    -> select noofpatients;
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call total_patients();
+--------------+
| noofpatients |
+--------------+
|            5 |
+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

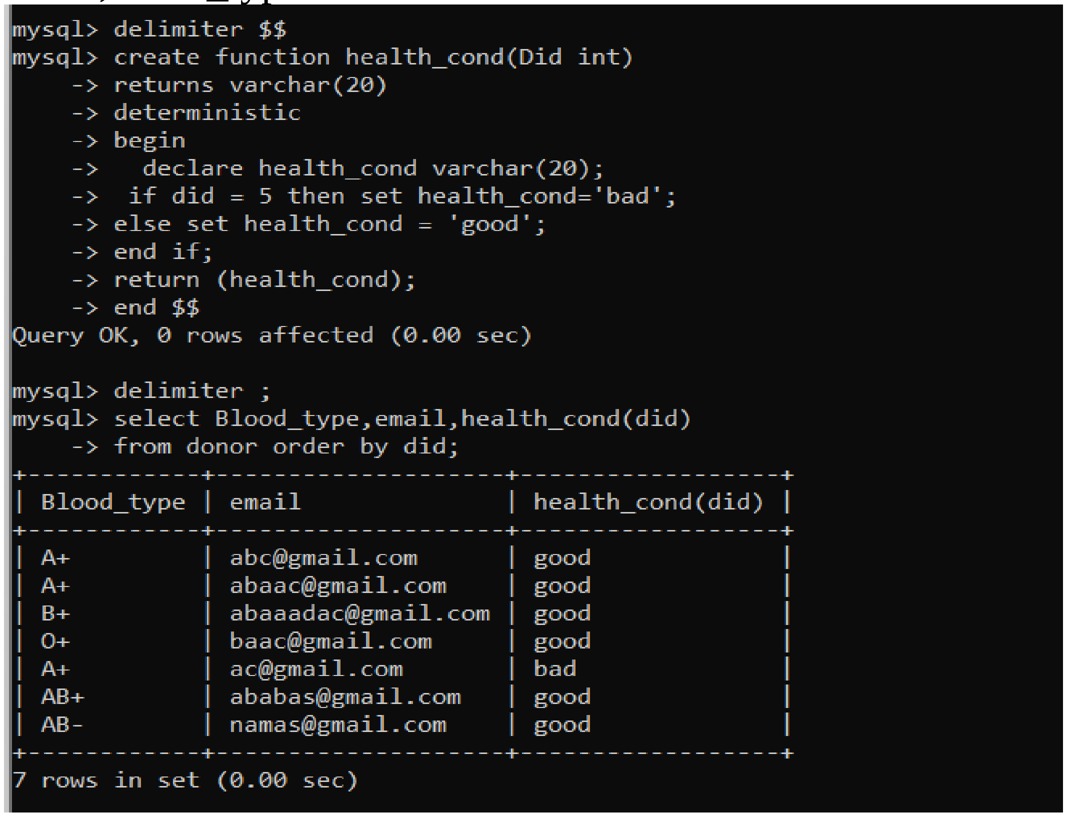
Stored Functions:

To know bloodtype , name of patient from age as input.



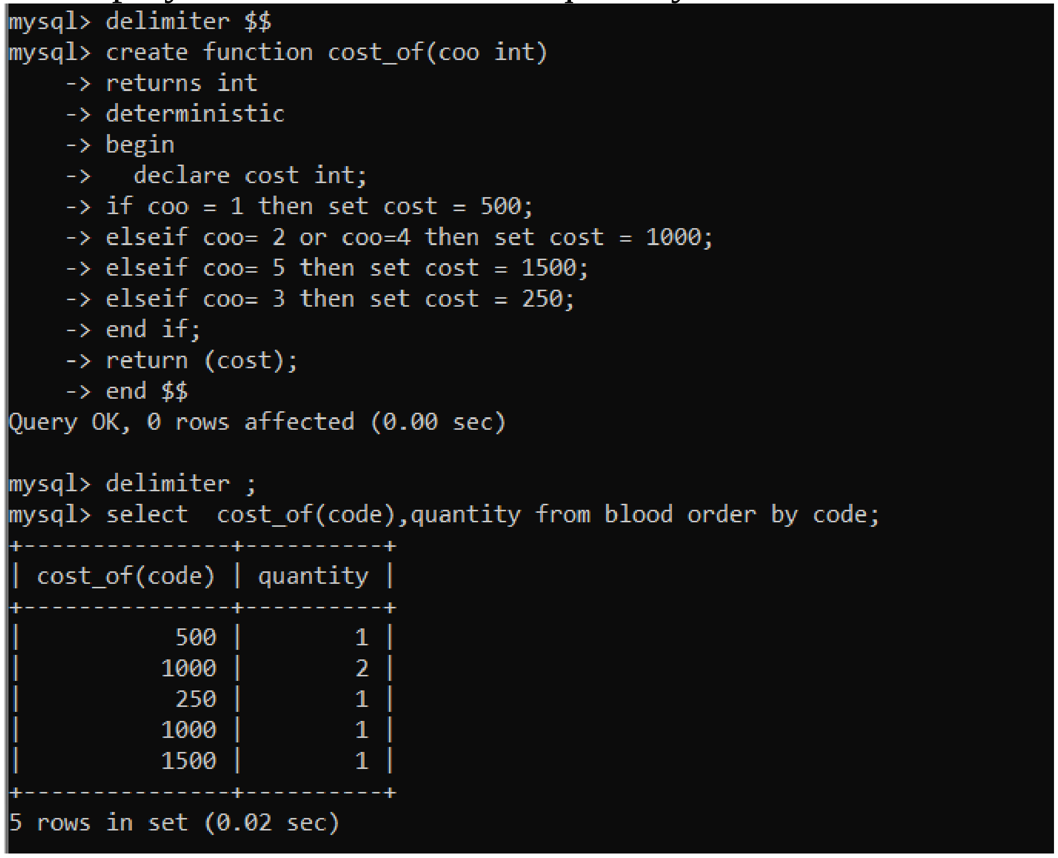
```
mysql> create function patient_name(age int)
    -> returns varchar(20)
    -> deterministic
    -> begin
    ->    declare patient_name varchar(20);
    ->    if age = 21 then
    -> set patient_name = 'asddd';
    ->    elseif age = 22 then
    -> set patient_name = 'addl';
    ->     elseif age > 25 and age < 36 then
    -> set patient_name = 'qpoad';
    ->     elseif age = 55 then
    -> set patient_name = 'majam';
    ->    end if;
    -> return (patient_name);
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select bloodtype, patient_name(age)
    -> from patient order by age;
+-----------+-------------------+
| bloodtype | patient_name(age) |
+-----------+-------------------+
| A+        | asddd             |
| B+        | addl              |
| B+        | addl              |
| O+        | qpoad             |
| B+        | majam             |
+-----------+-------------------+
5 rows in set (0.00 sec)
```

To know the status of donor’s health condition with email,blood_type



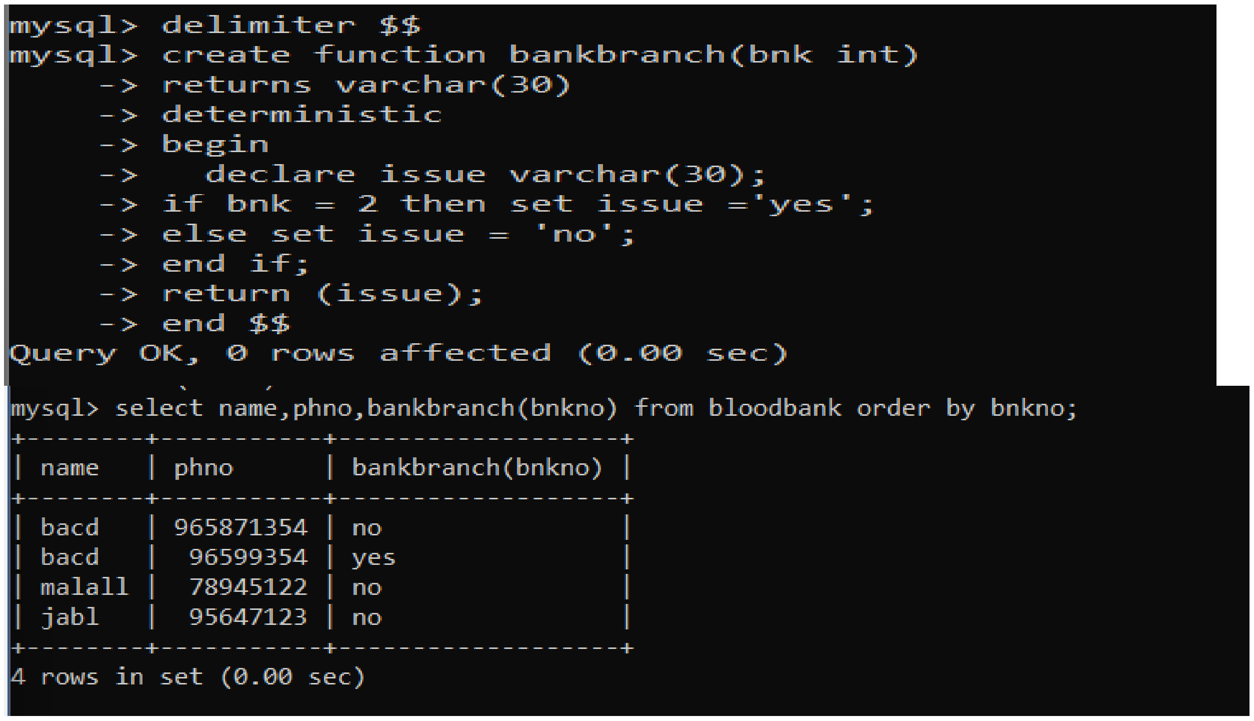
```
mysql> delimiter $$
mysql> create function health_cond(Did int)
    -> returns varchar(20)
    -> deterministic
    -> begin
    ->   declare health_cond varchar(20);
    ->  if did = 5 then set health_cond='bad';
    -> else set health_cond = 'good';
    -> end if;
    -> return (health_cond);
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select Blood_type,email,health_cond(did)
    -> from donor order by did;
+------------+--------------------+------------------+
| Blood_type | email              | health_cond(did) |
+------------+--------------------+------------------+
| A+         | abc@gmail.com      | good             |
| A+         | abaac@gmail.com    | good             |
| B+         | abaaadac@gmail.com | good             |
| O+         | baac@gmail.com     | good             |
| A+         | ac@gmail.com       | bad              |
| AB+        | ababas@gmail.com   | good             |
| AB-        | namas@gmail.com    | good             |
+------------+--------------------+------------------+
7 rows in set (0.00 sec)
```

To display cost of blood for the quantity.



```
mysql> delimiter $$
mysql> create function cost_of(coo int)
    -> returns int
    -> deterministic
    -> begin
    ->   declare cost int;
    -> if coo = 1 then set cost = 500;
    -> elseif coo= 2 or coo=4 then set cost = 1000;
    -> elseif coo= 5 then set cost = 1500;
    -> elseif coo= 3 then set cost = 250;
    -> end if;
    -> return (cost);
    -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select  cost_of(code),quantity from blood order by code;
+---------------+----------+
| cost_of(code) | quantity |
+---------------+----------+
|           500 |        1 |
|          1000 |        2 |
|           250 |        1 |
|          1000 |        1 |
|          1500 |        1 |
+---------------+----------+
5 rows in set (0.02 sec)
```

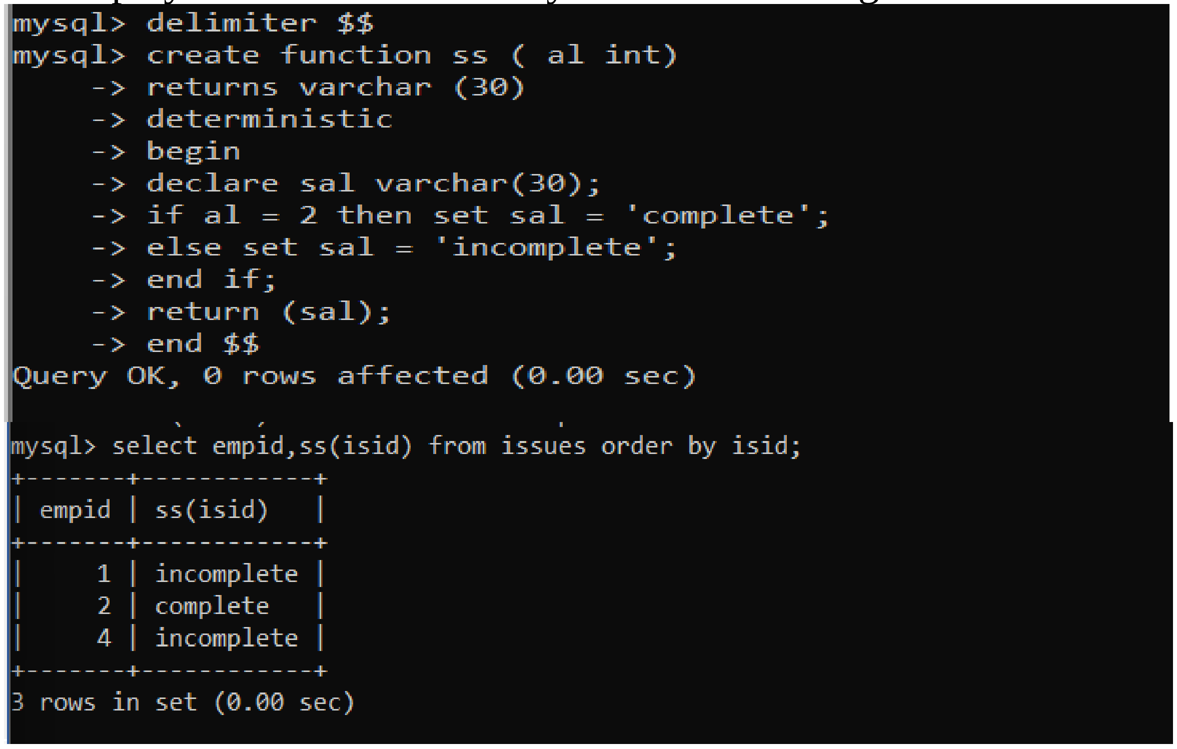
To find issue of bankbranch whether it is there or not.



```
mysql> delimiter $$
mysql> create function bankbranch(bnk int)
    -> returns varchar(30)
    -> deterministic
    -> begin
    ->   declare issue varchar(30);
    -> if bnk = 2 then set issue ='yes';
    -> else set issue = 'no';
    -> end if;
    -> return (issue);
    -> end $$
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select name,phno,bankbranch(bnkno) from bloodbank order by bnkno;
+--------+-----------+-------------------+
| name   | phno      | bankbranch(bnkno) |
+--------+-----------+-------------------+
| bacd   | 965871354 | no                |
| bacd   |  96599354 | yes               |
| malall |  78945122 | no                |
| jabl   |  95647123 | no                |
+--------+-----------+-------------------+
4 rows in set (0.00 sec)
```

To display the status of issues by bloodbank manager.

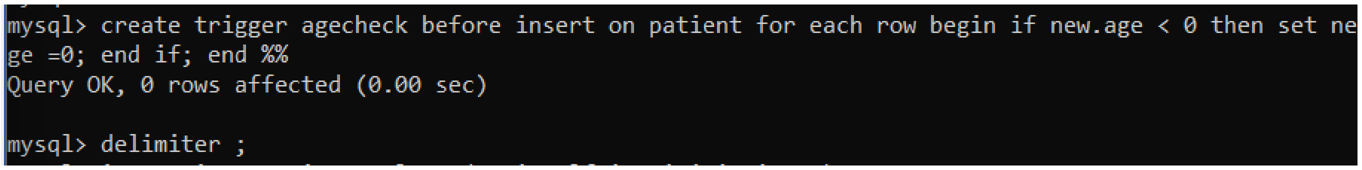


```
mysql> delimiter $$
mysql> create function ss ( al int)
    -> returns varchar (30)
    -> deterministic
    -> begin
    -> declare sal varchar(30);
    -> if al = 2 then set sal = 'complete';
    -> else set sal = 'incomplete';
    -> end if;
    -> return (sal);
    -> end $$
Query OK, 0 rows affected (0.00 sec)
```

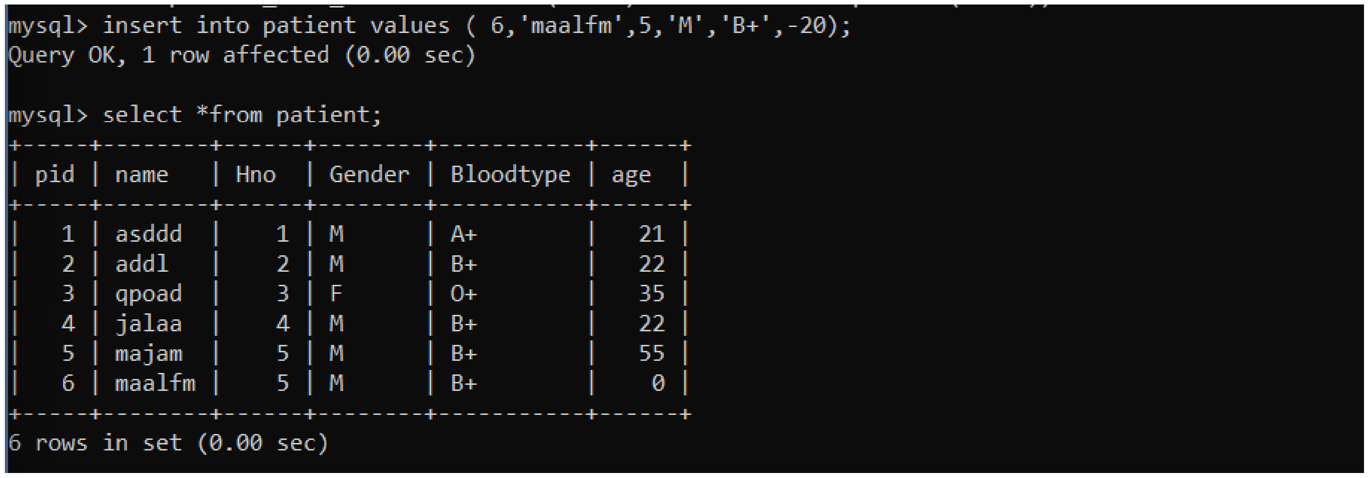
```
mysql> select empid,ss(isid) from issues order by isid;
+-------+------------+
| empid | ss(isid)   |
+-------+------------+
|     1 | incomplete |
|     2 | complete   |
|     4 | incomplete |
+-------+------------+
3 rows in set (0.00 sec)
```

Triggers

Create trigger check age which is given is greater than zero then only insert the values to patient table.

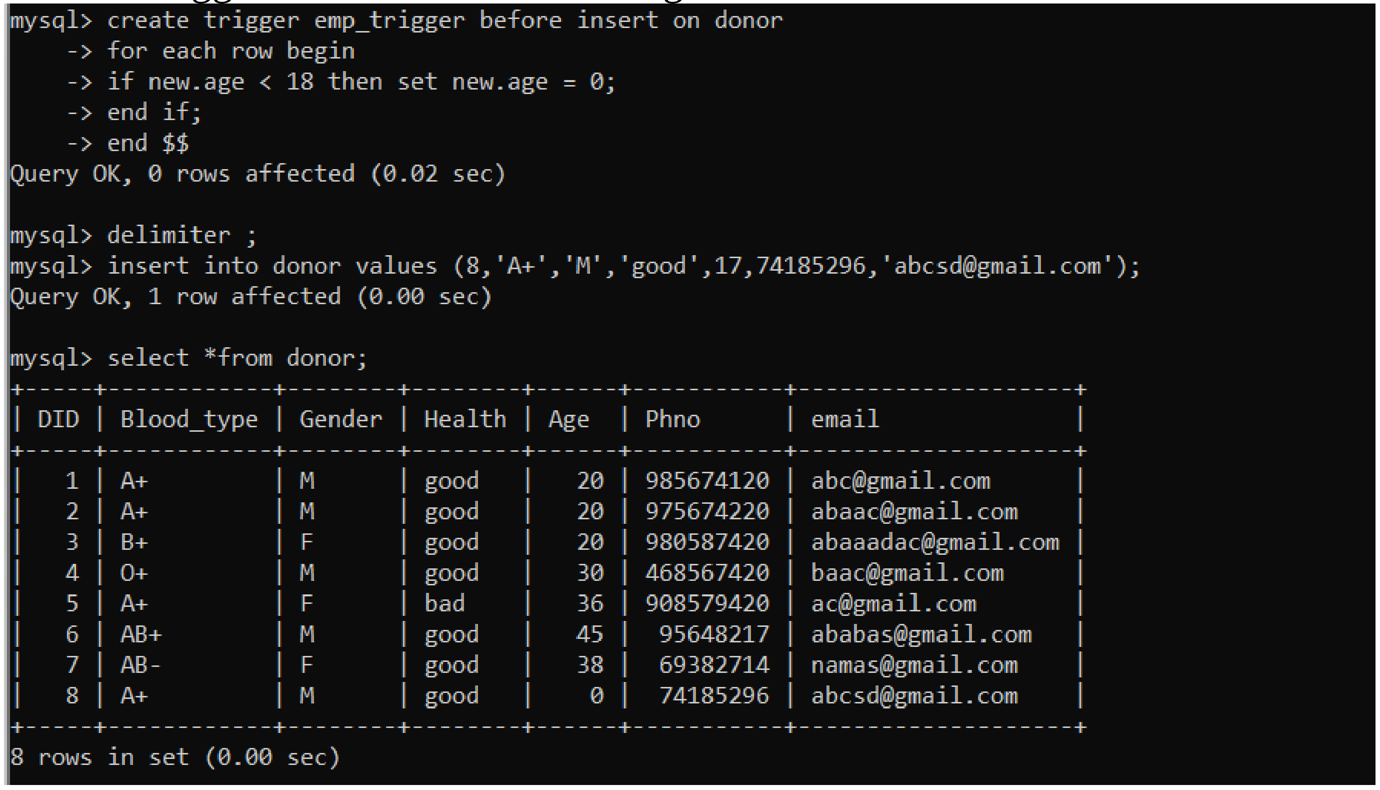


```
mysql> create trigger agecheck before insert on patient for each row begin if new.age < 0 then set ne
ge =0; end if; end %%
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
```



```
mysql> insert into patient values ( 6,'maalfm',5,'M','B+',-20);
Query OK, 1 row affected (0.00 sec)

mysql> select *from patient;
+-----+--------+------+--------+-----------+------+
| pid | name   | Hno  | Gender | Bloodtype | age  |
+-----+--------+------+--------+-----------+------+
|   1 | asddd  |    1 | M      | A+        |   21 |
|   2 | addl   |    2 | M      | B+        |   22 |
|   3 | qpoad  |    3 | F      | O+        |   35 |
|   4 | jalaa  |    4 | M      | B+        |   22 |
|   5 | majam  |    5 | M      | B+        |   55 |
|   6 | maalfm |    5 | M      | B+        |    0 |
+-----+--------+------+--------+-----------+------+
6 rows in set (0.00 sec)
```

Create a trigger not to allow donor of age below 18.



```
mysql> create trigger emp_trigger before insert on donor
    -> for each row begin
    -> if new.age < 18 then set new.age = 0;
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> insert into donor values (8,'A+','M','good',17,74185296,'abcsd@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> select *from donor;
+-----+------------+--------+--------+------+-----------+--------------------+
| DID | Blood_type | Gender | Health | Age  | Phno      | email              |
+-----+------------+--------+--------+------+-----------+--------------------+
|   1 | A+         | M      | good   |   20 | 985674120 | abc@gmail.com      |
|   2 | A+         | M      | good   |   20 | 975674220 | abaac@gmail.com    |
|   3 | B+         | F      | good   |   20 | 980587420 | abaaadac@gmail.com |
|   4 | O+         | M      | good   |   30 | 468567420 | baac@gmail.com     |
|   5 | A+         | F      | bad    |   36 | 908579420 | ac@gmail.com       |
|   6 | AB+        | M      | good   |   45 |  95648217 | ababas@gmail.com   |
|   7 | AB-        | F      | good   |   38 |  69382714 | namas@gmail.com    |
|   8 | A+         | M      | good   |    0 |  74185296 | abcsd@gmail.com    |
+-----+------------+--------+--------+------+-----------+--------------------+
8 rows in set (0.00 sec)
```