**DBMS - Mini Project**

**Organ Donation and Procurement Management System**

Submitted By:

Name :Preethika Ajay Kumar

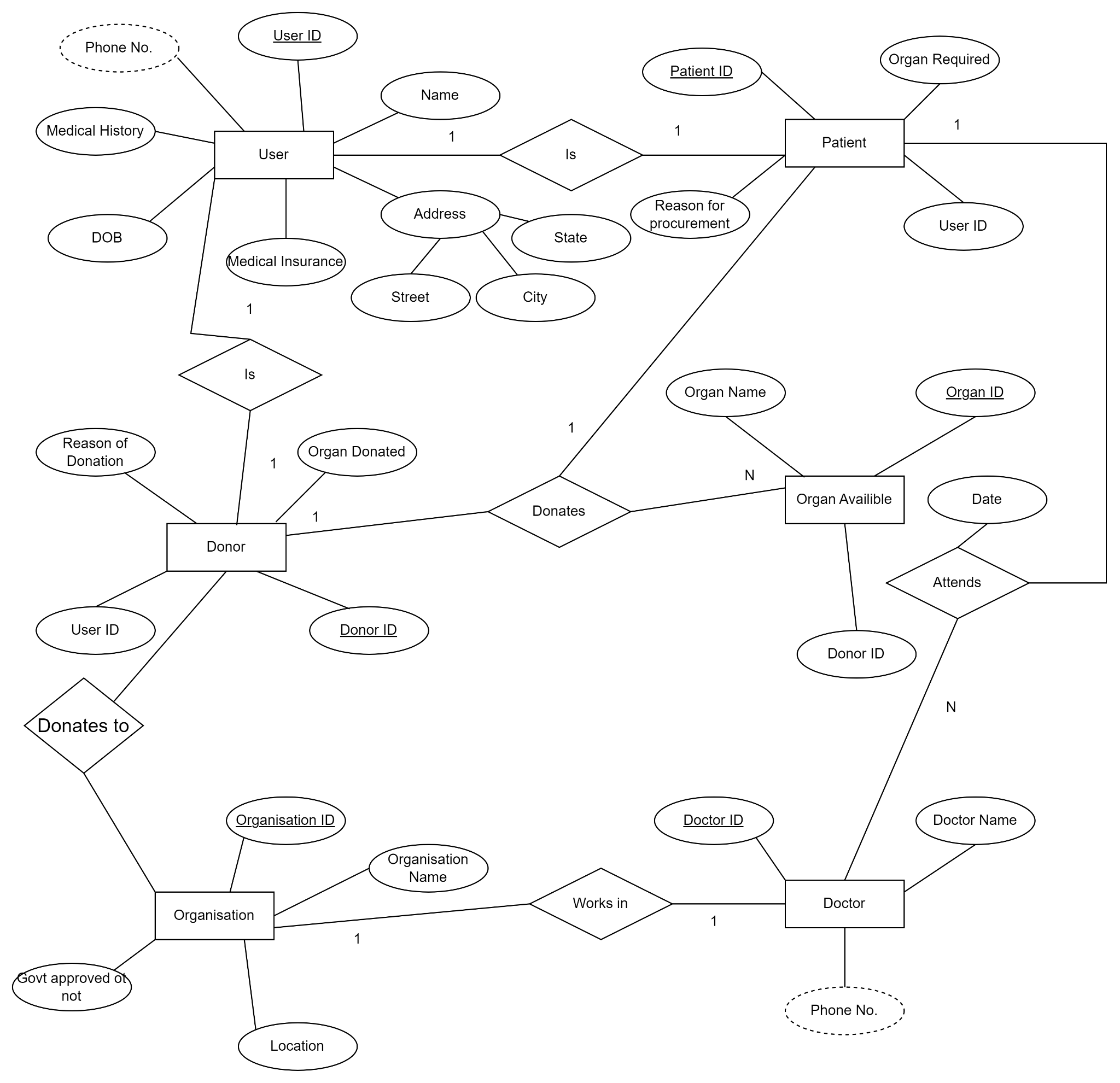
SRN : PES2UG20CS256

V Semester Section D

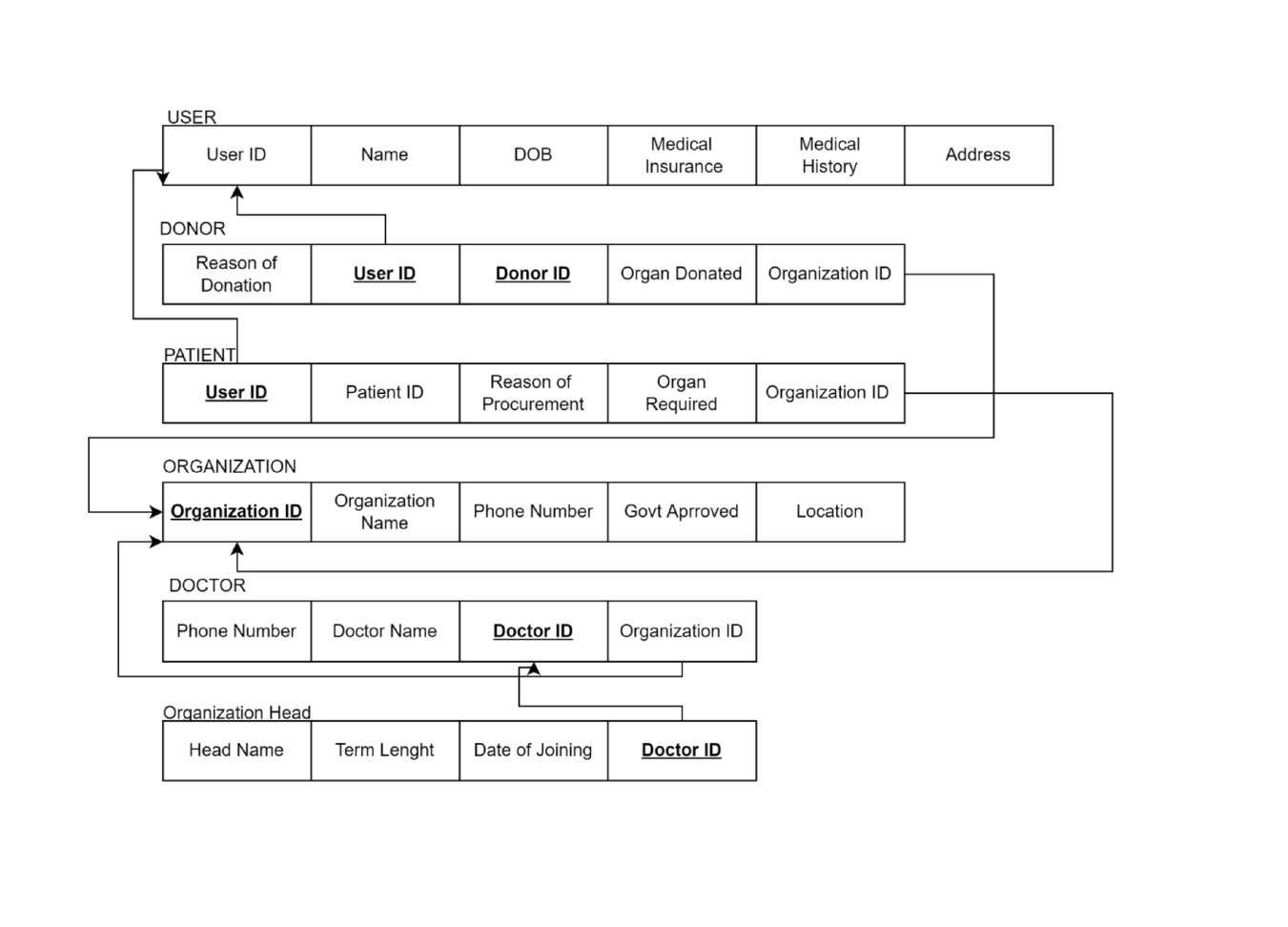
**Short Description and Scope of the Project**

Organ donation and procurement plays a crucial role in improving the lives of patients worldwide. One main problem that is present today is the fact that organs are not available in time to save patients’ lives. This system uses database technologies to manage a person’s donation of organs. Organ procurement facilities are also enlisted. The database will be designed to maintain a complete medical record of all patients, along with information based on geographic location. These features will enable easy procurement for patients in time of need.

**ER Diagram**

****

**Relational Schema**

****

**DDL statements - Building the database**

CREATE DATABASE DBMS\_proj\_final;

USE DBMS\_proj\_final;

#table 1

CREATE TABLE User(

User\_ID int NOT NULL,

Name varchar(20) NOT NULL,

Date\_of\_Birth date NOT NULL,

Medical\_insurance int,

Medical\_history varchar(20),

Street varchar(20),

City varchar(20),

State varchar(20),

Phno\_user varchar(10),

PRIMARY KEY(User\_ID)

);

#table 2

CREATE TABLE Organization(

Organization\_ID int NOT NULL,

Organization\_name varchar(20) NOT NULL,

Location varchar(20),

Government\_approved int, # 0 or 1

PRIMARY KEY(Organization\_ID)

);

#table 3

CREATE TABLE Doctor(

Doctor\_ID int NOT NULL,

Doctor\_Name varchar(20) NOT NULL,

organization\_ID int NOT NULL,

FOREIGN KEY(organization\_ID) REFERENCES Organization(organization\_ID) ON DELETE CASCADE,

PRIMARY KEY(Doctor\_ID)

);

#table 4

CREATE TABLE Patient(

Patient\_ID int NOT NULL,

organ\_req varchar(20) NOT NULL,

reason\_of\_procurement varchar(20),

Doctor\_ID int NOT NULL,

User\_ID int NOT NULL,

FOREIGN KEY(User\_ID) REFERENCES User(User\_ID) ON DELETE CASCADE,

FOREIGN KEY(Doctor\_ID) REFERENCES Doctor(Doctor\_ID) ON DELETE CASCADE,

PRIMARY KEY(Patient\_Id, organ\_req)

);

#table 5

CREATE TABLE Donor(

Donor\_ID int NOT NULL,

organ\_donated varchar(20) NOT NULL,

reason\_of\_donation varchar(20),

Organization\_ID int NOT NULL,

User\_ID int NOT NULL,

FOREIGN KEY(User\_ID) REFERENCES User(User\_ID) ON DELETE CASCADE,

FOREIGN KEY(Organization\_ID) REFERENCES Organization(Organization\_ID) ON DELETE CASCADE,

PRIMARY KEY(Donor\_ID, organ\_donated)

);

#table 6

CREATE TABLE Organ\_available(

Organ\_ID int NOT NULL AUTO\_INCREMENT,

Organ\_name varchar(20) NOT NULL,

Donor\_ID int NOT NULL,

FOREIGN KEY(Donor\_ID) REFERENCES Donor(Donor\_ID) ON DELETE CASCADE,

PRIMARY KEY(Organ\_ID)

);

**Populating the Database**

insert into Organization values

(1, 'Organization-1','New Delhi',1),

(2, 'Organization-2','Mumbai',0),

(3, 'Organization-3','Kolkata',0),

(4, 'Organization-4','Kolkata',1),

(5, 'Organization-5','Ahmedabad',1),

(6, 'Organization-6','Kolkata',0),

(7, 'Organization-7','Kolkata',0),

(8, 'Organization-8','Ahmedabad',0),

(9, 'Organization-9','Kolkata',1),

(10, 'Organization-10','Ahmedabad',1),

(11, 'Organization-11','Ahmedabad',1),

(12, 'Organization-12','Mumbai',0),

(13, 'Organization-13','Kolkata',0),

(14, 'Organization-14','Ahmedabad',1),

(15, 'Organization-15','Ahmedabad',0);

insert into Doctor values

(1,'Doctor-1',12),

(2,'Doctor-2',10),

(3,'Doctor-3',1),

(4,'Doctor-4',6),

(5,'Doctor-5',11),

(6,'Doctor-6',9),

(7,'Doctor-7',5),

(8,'Doctor-8',4),

(9,'Doctor-9',7),

(10,'Doctor-10',3),

(11,'Doctor-11',8),

(12,'Doctor-12',2),

(13,'Doctor-13',13),

(14,'Doctor-14',15),

(15,'Doctor-15',14);

insert into User values

( 1 ,'Name-1','1978-8-21',1,'NIL','Street-1','New Delhi','Delhi', '1212121212'),

( 2 ,'Name-2','1975-12-10',0,'NIL','Street-2','Mumbai','Maharashtra', '2121212121'),

( 3 ,'Name-3','1976-6-4',0,'NIL','Street-3','Mumbai','Maharashtra', '2323232323'),

( 4 ,'Name-4','1985-10-13',1,'NIL','Street-4','Ahmedabad','Gujarat', '3232323232'),

( 5 ,'Name-5','1983-10-12',1,'NIL','Street-5','Kolkata','West Bengal', '3434343434'),

( 6 ,'Name-6','1977-1-18',1,'NIL','Street-6','Kolkata','West Bengal', '4343434343'),

( 7 ,'Name-7','1976-2-26',0,'NIL','Street-7','New Delhi','Delhi', '4545454545'),

( 8 ,'Name-8','1973-4-12',1,'NIL','Street-8','Mumbai','Maharashtra', '5454545454'),

( 9 ,'Name-9','1976-11-1',0,'NIL','Street-9','Mumbai','Maharashtra', '5656565656'),

( 10 ,'Name-10','1978-11-18',1,'NIL','Street-10','New Delhi','Delhi', '6565656565'),

( 11 ,'Name-11','1975-1-6',1,'NIL','Street-11','Mumbai','Maharashtra', '6767676767'),

( 12 ,'Name-12','1983-11-1',1,'NIL','Street-12','Mumbai','Maharashtra', '7676767676'),

( 13 ,'Name-13','1983-1-9',1,'NIL','Street-13','New Delhi','Delhi', '7878787878'),

( 14 ,'Name-14','1975-10-12',1,'NIL','Street-14','Mumbai','Maharashtra', '8787878787'),

( 15 ,'Name-15','1977-9-23',0,'NIL','Street-15','Ahmedabad','Gujarat', '8989898989');

insert into Patient values

(1,'Heart','Reason-1',3,12),

(2,'Kidney','Reason-2',2,13),

(3,'Pancreas','Reason-3',7,8),

(4,'Kidney','Reason-4',8,7),

(5,'Heart','Reason-5',4,11),

(6,'Lung','Reason-6',1,14),

(7,'Intestine','Reason-7',10,5),

(8,'Intestine','Reason-8',12,3),

(9,'Lung','Reason-9',11,4),

(10,'Kidney','Reason-13',6,9),

(11,'Kidney','Reason-11',9,6),

(12,'Pancreas','Reason-12',5,10),

(13,'Intestine','Reason-13',13,2),

(14,'Heart','Reason-14',14,1),

(15,'Kidney','Reason-15',15,15);

insert into Donor values

(1,'Heart','Reason-1',7,8),

(2,'Pancreas','Reason-2',9,6),

(3,'Pancreas','Reason-3',1,14),

(4,'Intestine','Reason-4',6,9),

(5,'Kidney','Reason-5',8,7),

(6,'Pancreas','Reason-6',2,13),

(7,'Kidney','Reason-7',5,10),

(8,'Kidney','Reason-8',3,12),

(9,'Heart','Reason-9',15,15),

(10,'Heart','Reason-10',4,11),

(11,'Kidney','Reason-11',11,4),

(12,'Pancreas','Reason-12',4,12),

(13,'Pancreas','Reason-13',12,3),

(14,'Pancreas','Reason-14',10,5),

(15,'Heart','Reason-15',5,4),

(16,'Intestine','Reason-16',14,1),

(17,'Intestine','Reason-17',13,2),

(18,'Intestine','Reason-18',2,7),

(19,'Pancreas','Reason-19',9,9),

(20,'Intestine','Reason-20',1,2);

insert into organ\_available values

(1,'Heart',9),

(2,'Pancreas',4),

(3,'Pancreas',5),

(4,'Intestine',6),

(5,'Kidney',12),

(6,'Pancreas',8),

(7,'Kidney',3),

(8,'Kidney',11),

(9,'Heart',1),

(10,'Heart',10),

(11,'Kidney',13),

(12,'Pancreas',2),

(13,'Pancreas',7),

(14,'Pancreas',14),

(15,'Heart',15),

(16,'Intestine',3),

(17,'Intestine',9),

(18,'Intestine',7),

(19,'Pancreas',6);

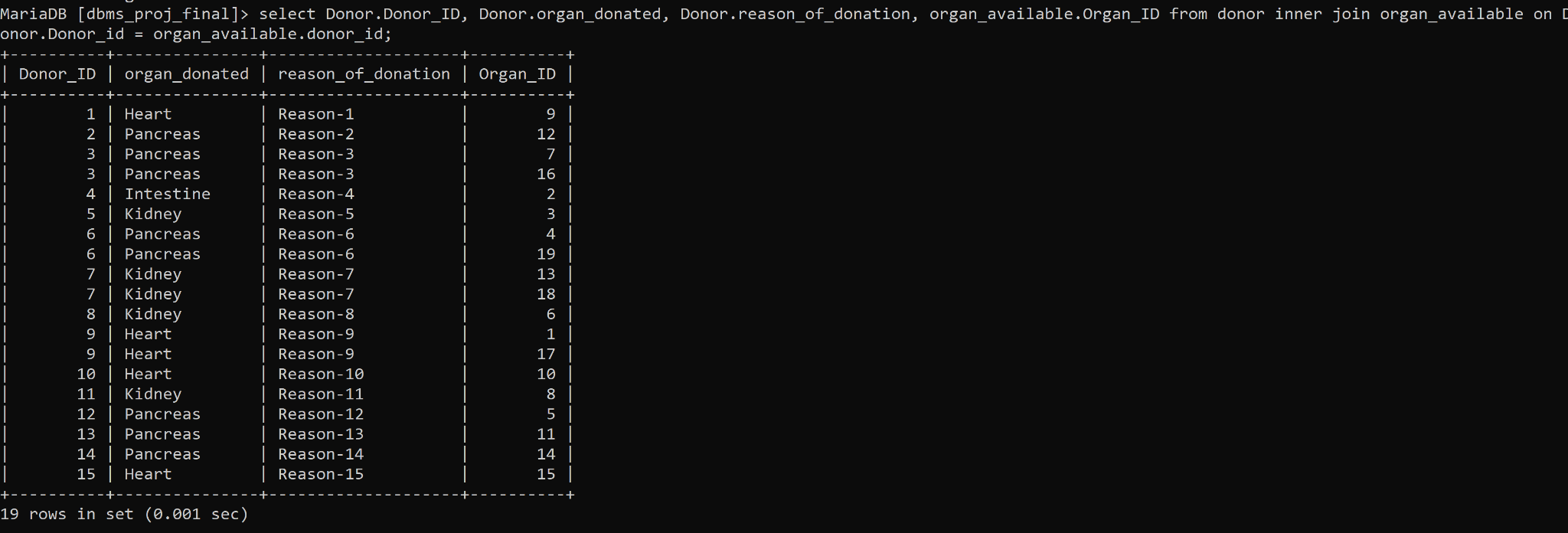
**Join Queries**

Showcase at least 4 join queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

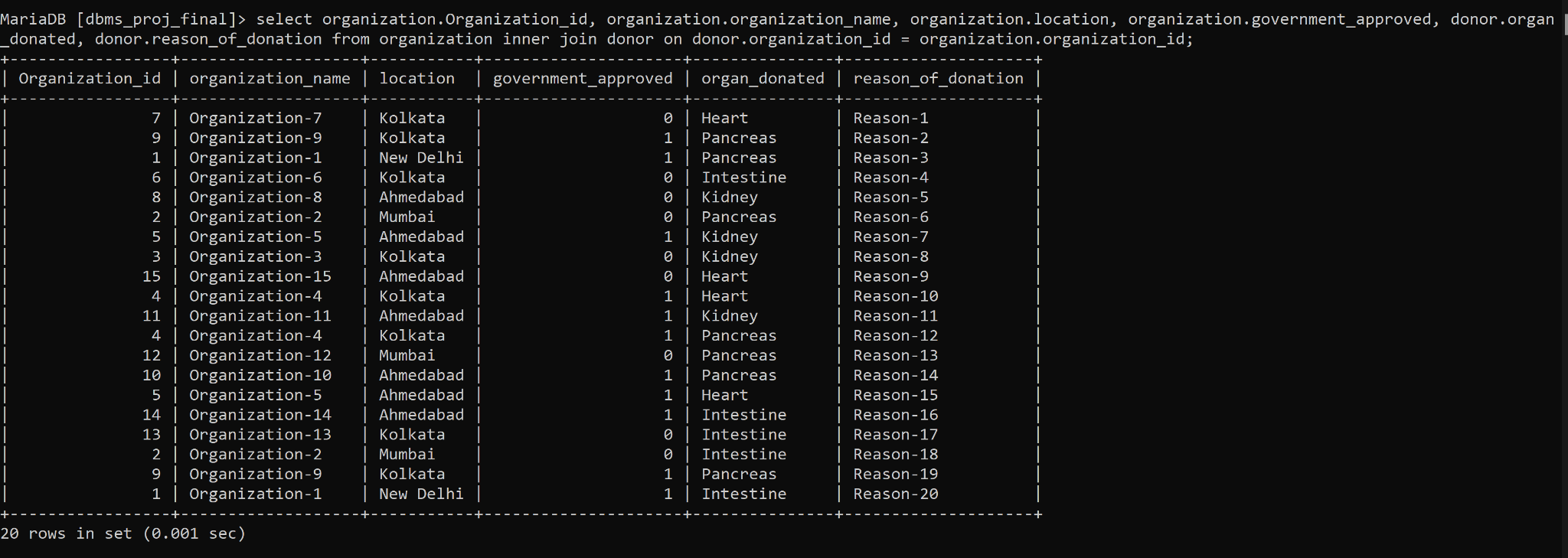
Join of Organ\_available and Donor, so as to get all the organs donated by a donor along with organ\_id

**select Donor.Donor\_ID, Donor.organ\_donated, Donor.reason\_of\_donation, organ\_available.Organ\_ID from donor inner join organ\_available on Donor.Donor\_id = organ\_available.donor\_id;**



Join to find the location where the donor has donated the organ.

**select organization.Organization\_id, organization.organization\_name, organization.location, organization.government\_approved, donor.organ\_donated, donor.reason\_of\_donation from organization inner join donor on donor.organization\_id = organization.organization\_id;**



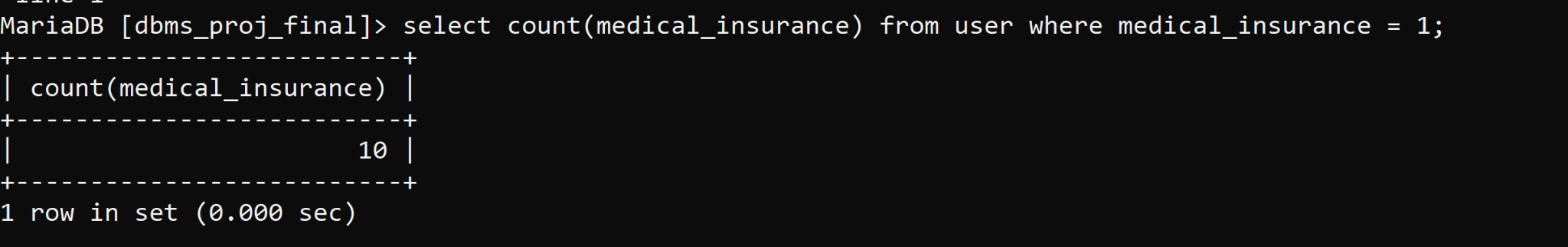
**Aggregate Functions**

Showcase at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

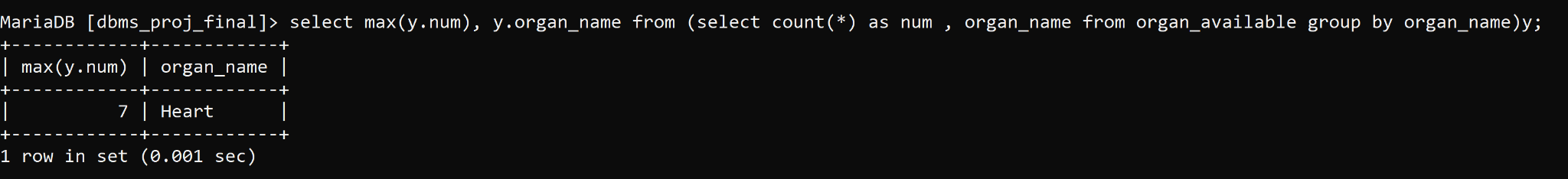
To get the count of number of users that have medical insurance:

**select count(medical\_insurance) from user where medical\_insurance = 1;**



To select the organ which has maximum number of available units for donation

**select max(y.num), y.organ\_name from (select count(\*) as num , organ\_name from organ\_available group by organ\_name)y;**



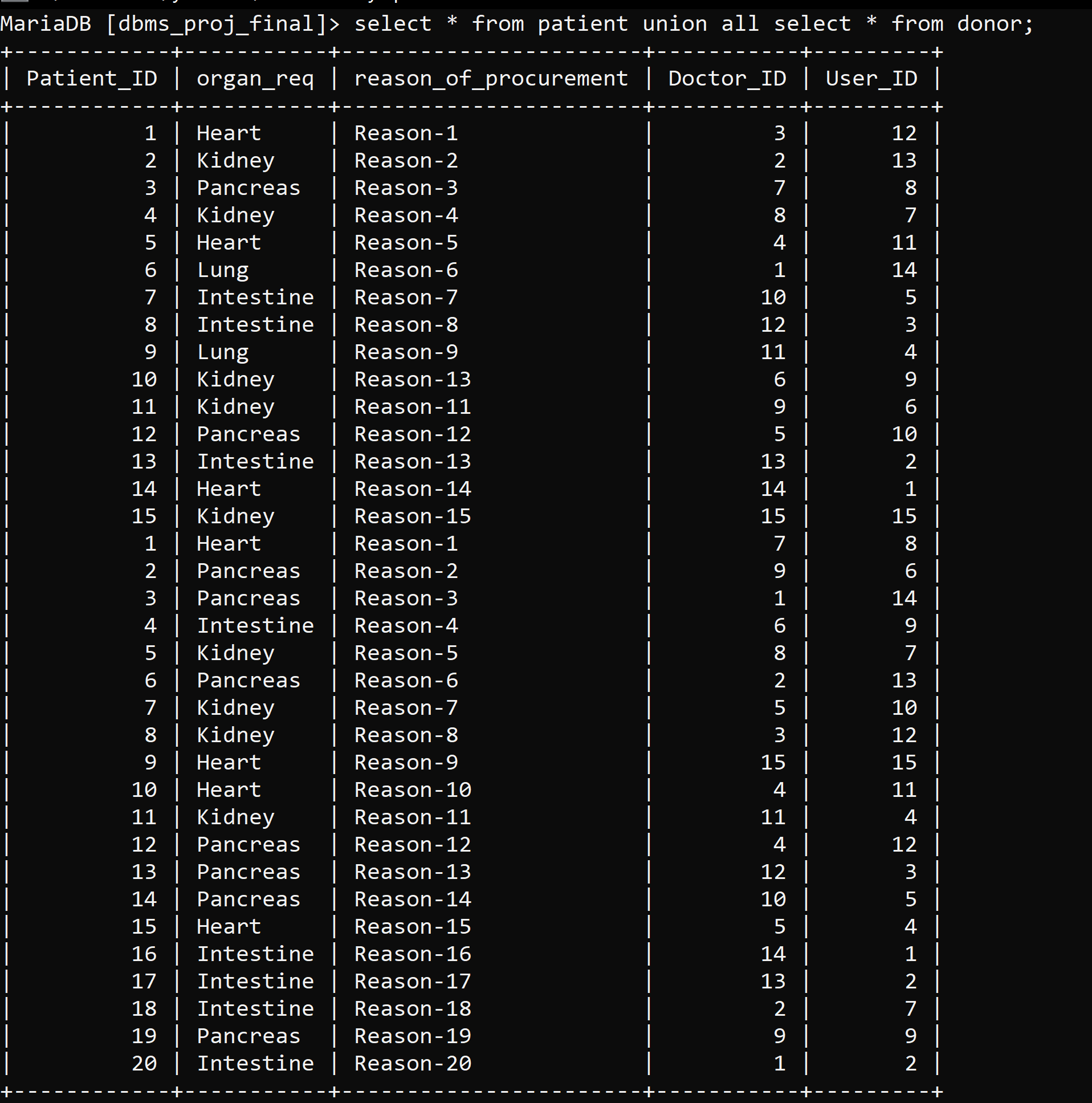
**Set Operations**

Showcase at least 4 Set Operations queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

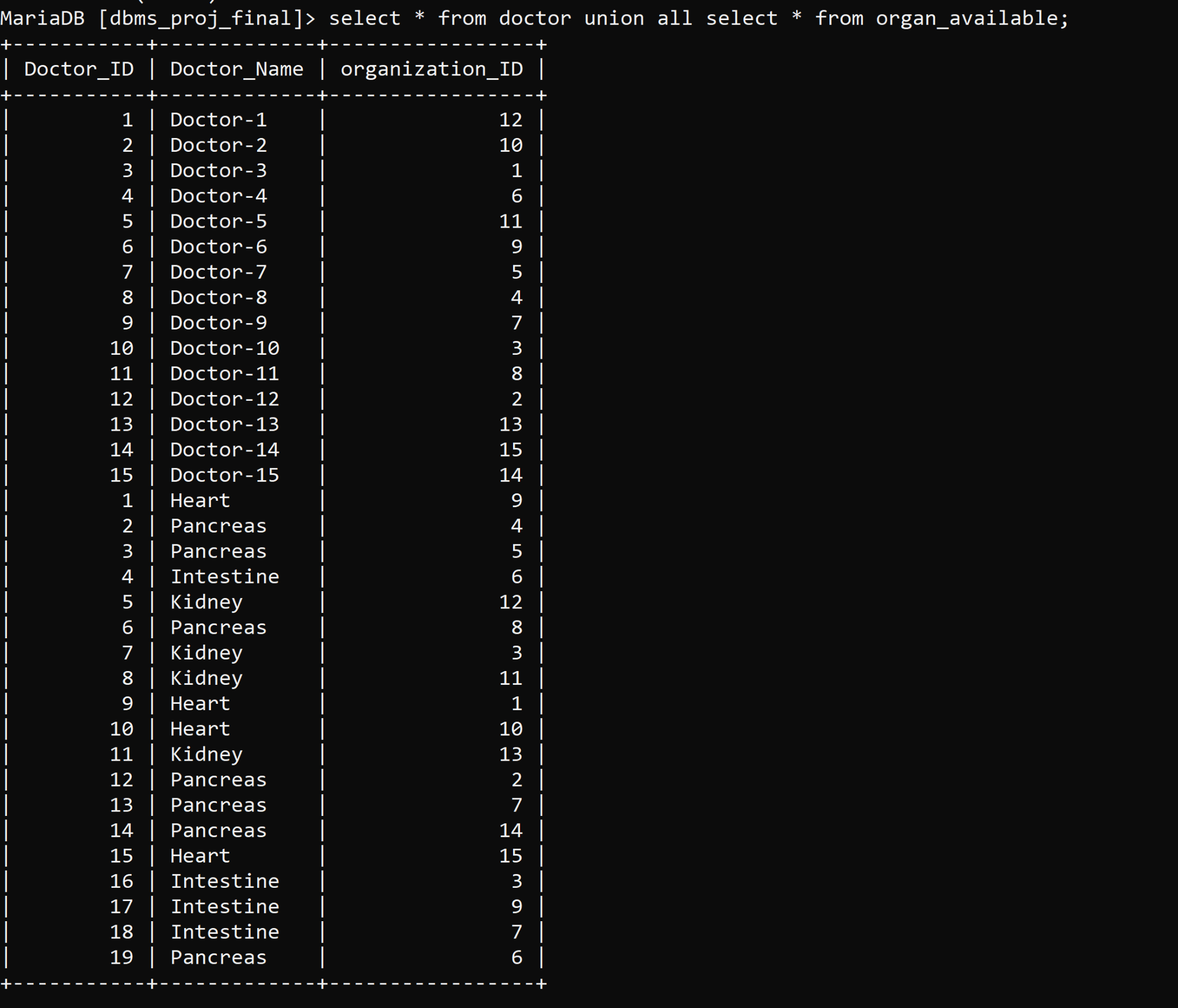
List of Donors and Patients

**select \* from patient union all select \* from donor;**



Union of doctor and organ\_available

**select \* from doctor union all select \* from organ\_available;**



**Functions and Procedures**

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

**DELIMITER //**

**CREATE PROCEDURE get\_loc()**

**BEGIN**

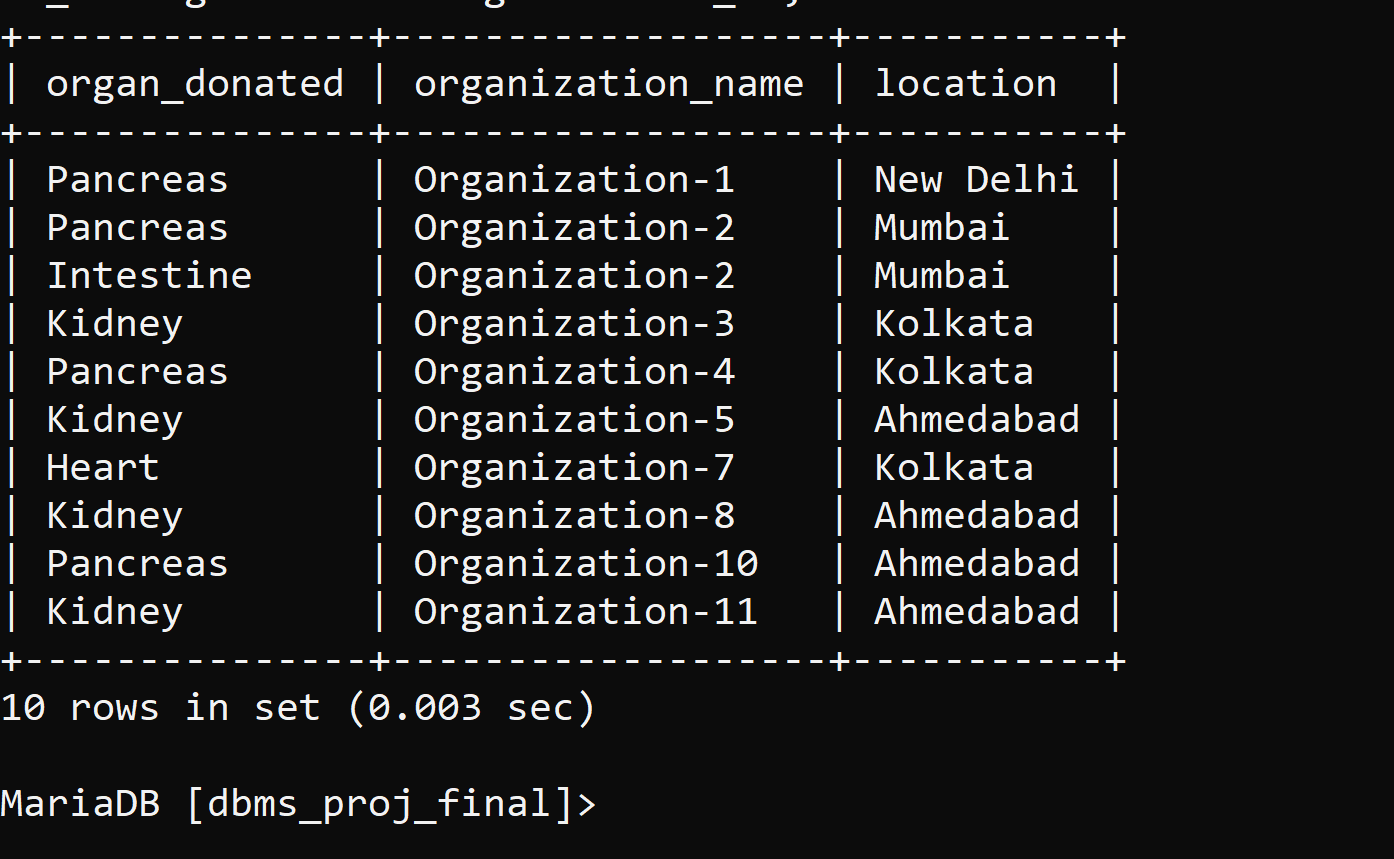
**SELECT donor.organ\_donated, organization.organization\_name, organization.location FROM donor INNER JOIN organization ON donor.organization\_id=organization.organization\_id;**

**END;**

**//**

**DELIMITER;**

**CALL get\_loc()**



**Triggers and Cursors**

Create a Trigger and a Cursor. State the objective. Run and display the results.

Trigger to get a insert name of users with medical insurance into another table.

**delimiter |**

**CREATE TRIGGER med\_ins BEFORE INSERT ON user**

**FOR EACH ROW**

**BEGIN**

**IF NEW.medical\_insurance = 1 THEN**

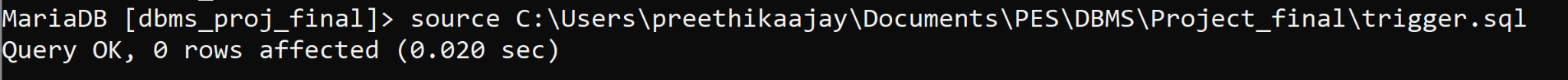
**INSERT INTO med\_insurance SET name = NEW.name;**

**END IF;**

**END;**

**|**

**delimiter ;**



**Developing a Frontend**

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table 2. There should be an window to accept and run any SQL statement and display the result

