Blood and Organ Donation Management System

Project Design Report IST659 – Section M003

Instructor: Prof. Yang Wang

Submitted by: Tanya Shrivastava

Project Summary

The proposed project is for institutions which deal with organ and blood donations on a daily basis. For such an environment, information is vital and the data accuracy is of grave importance. Simply put, there is no scope for human error in such scenarios where a slip up could lead to the loss of life.

While considering organ donation, one must keep in mind that there are several kinds of donors: Living donors, Deceased donors, VCA (Vascularized Composite Allografts) donors, and pediatric donors. Living donations are valued more than deceased donations as the donor is alive and the quality of organ is considerably better. VCA donations involve multiple tissue types including skin, bone, nerves, and blood vessels where as pediatric donations refers to donation of child-sized organs.

Although these donors may seem different, the process of every donation is bound by the medical criteria that the organ donor and the organ receiver/patient must have a “compatible blood type”. Hence, Users must register in order to Donate or Receive an organ or blood.

A system that uses a domain for Patients, Donors and Employees (that work at the blood and organ donation institution) is proposed. The Donors fill a Donor Registration Form after registering as a user in order to make the donation request available to the system. As for the patients, they fill out a Patient Request Form against which all possible donations are matched to find a compatible blood type in order to process any kind of donation(Blood or Organ).

Employees can use the system to check for compatible blood type and match the patient to a donor. The system also considers the patient’s age and location while creating matches for organ or blood donation. For example, the best match for a patient who is 20 years old will be a donor whose age is within 5 years of his age. Similarly, another factor to consider is the location of the donor and patient. Obviously, most patients in need of organs or blood cannot travel. Thus, the system needs to find donors within the same city.

The aim of the system is to reduce the workload for the employees and thus minimize the chances of an error. Although the system might be complicated to deal with at first, it has a lot of scope and can reduce the number of employees required to carry out the same task which reduces the expenditure of the institution.

As the information involved in this problem statement is very hefty, a structured system should be developed to deal with it. Such a system design is reflected in this report along with the its implementation using SQL Server and Microsoft Access. These tools are used for their ability to easily transform a big database into a user friendly and interactive graphical user interface. The project also has a wider scope than described in the report and can be used to apply more constraints by adding more attributes.

Entity and Attribute Tables

1. Users - This entity captures information about the user who requests for an organ/blood (Patient), receives an organ/blood (Receiver), or Donates an organ/blood. User is a super-type for Patient, Receiver and Donor object.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Users | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ; UID given  by the B&OD Management System |
| OTHER ATTRIBUTES | UFName | VARCHAR(20) | NOT NULL | First Name of the User |
|  | ULName | VARCHAR(20) | NOT NULL | Last Name of the User |
|  | UAge | INT | NOT NULL | Age of the User |
|  | UPhone | VARCHAR(20) | NOT NULL | Contact Number of the  User |
|  | UAddress | VARCHAR(20) | NOT NULL | Address of the User |
|  | UCity | VARCHAR(20) | NOT NULL | City the user is currently residing in |
|  | UState | VARCHAR(20) | NOT NULL | State the user is currently residing in |
|  | UBloodtype | VARCHAR(20) | NOT NULL | Blood Group of the user |
|  | UType | VARCHAR(20) | NOT NULL | (Discriminator)  Whether the user is a Patient, Donor or Reciever |

1. Patient - This entity captures information about the User who makes a request for a donation (either organ or blood). It is a sub-type of User.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Patient | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY;  FOREIGN KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ; UID given  by the B&OD Management System |
| OTHER ATTRIBUTES | Condition | VARCHAR(20) | NOT NULL | The state of the Patient at the time of request  ( Critical, Serious, Stable, Intensive ) |

1. Donor - This entity captures information about the User who register to make a donation (either organ or blood). It is a sub-type of User.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Donor | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY;  FOREIGN KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ; UID given  by the B&OD Management System |
| OTHER ATTRIBUTES | DStatus | VARCHAR(20) | NOT NULL | The state of the Patient at the time of donation  ( Living, Deceased ) |

1. Receiver - This entity captures information about the User whose request for donation (either organ or blood) has been approved. It is a sub-type of User.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Receiver | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY;  FOREIGN KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ; UID given  by the B&OD Management System |
| OTHER ATTRIBUTES | Condition | VARCHAR(20) | NOT NULL | The state of the Receiver after the Donation  ( Critical, Serious, Stable, Intensive ) |

1. Patient Request – Contains the information about a Patient’s request for donation (Not Approved).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Receiver Request | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | RequestID | INT | NOT NULL | Unique Identifier for Patient Request Entity |
| FOREIGN KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ;UserID from User Table(of the Reciever) |
| FOREIGN KEY | EmployeeID | INT | NOT NULL | Unique Identifier for Employee Entity ; EmployeeID from Employee Table |
| OTHER ATTRIBUTES | RequestType | VARCHAR(20) | NOT NULL | The Patient request for which type of donation  (Liver, Kidney, Heart, Blood) |
|  | BloodType | VARCHAR(5) | NOT NULL | The blood type of the Patient. |
|  | RequestStatus | VARCHAR(20) | NULL | Reflect the status of the request(Approved, Null, Declined) |
|  | DateRequested | DATETIME() | NOT NULL | Timestamp of when the request was submitted |

1. Donor Registration – Contains the information about a Donor’s registration for a donation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Donor Registration | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | RegistrationID | INT | NOT NULL | Unique Identifier for Donor Registration Entity |
| FOREIGN KEY | UserID | INT | NOT NULL | Unique Identifier for User Entity ;UserID from User Table(of the Donor) |
| FOREIGN KEY | EmployeeID | INT | NOT NULL | Unique Identifier for Employee Entity ; EmployeeID from Employee Table |
| OTHER ATTRIBUTES | DonationType | VARCHAR(20) | NOT NULL | The Donor registers for which type of donation  (Liver, Kidney, Heart, Blood) |
|  | BloodType | VARCHAR(5) | NOT NULL | Blood Type of the Donor. |

1. Employee – Contains information about the employee who approves/rejects receiver request by matching them with Donor registration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  User | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | EmployeeID | INT | NOT NULL | Unique Identifier for Employee Entity |
| OTHER ATTRIBUTES | EFName | VARCHAR(20) | NOT NULL | First Name of the Employee |
|  | ELName | VARCHAR(20) | NOT NULL | Last Name of the Employee |
|  | EPhone | VARCHAR(10) | NOT NULL | Contact Number of the  Employee |
|  | EAddress | VARCHAR(20) | NOT NULL | Address of the Employee |

1. Donations – Contains the information about donation requests approved by the employee.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Donation | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | DonationID | INT | NOT NULL | Unique Identifier for Donation Entity |
| FOREIGN KEY | DonorID | INT | NOT NULL | Unique Identifier for User Entity ;UserID of the Donor from Donor Table |
| FOREIGN KEY | ReceverID | INT | NOT NULL | Unique Identifier for User Entity ;UserID of the Receiver from Receiver Table |
| FOREIGN KEY | EmployeeID | INT | NOT NULL | Unique Identifier for Employee Entity |
| OTHER ATTRIBUTES | DonationType | VARCHAR(20) | NOT NULL | type of donation  (Liver, Kidney, Heart, Blood) |

Entity Relationship Diagram



Creating Tables

-- User Table

Create table Users

(

UserID integer not null,

UFName varchar(20) not null,

ULName varchar(20) not null,

UAge integer not null,

UPhone varchar(10) not null,

UAddress varchar(20) not null,

UCity varchar (20) not null,

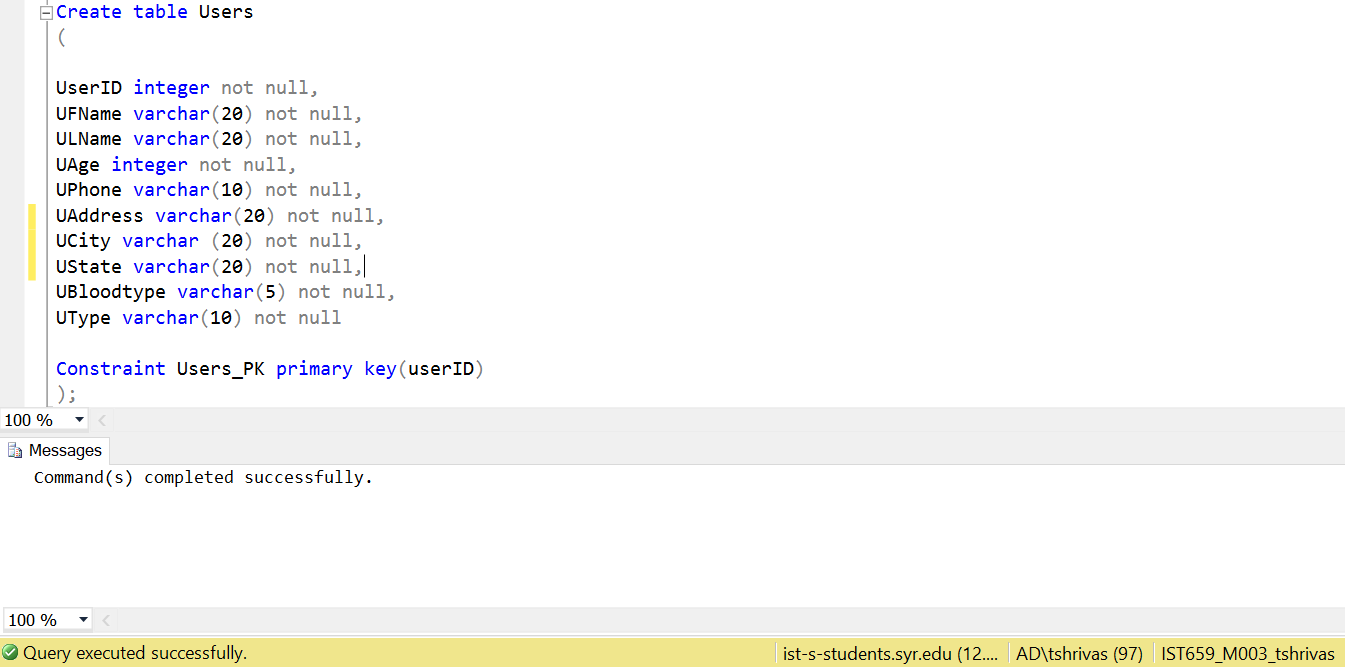
UState varchar(20) not null,

UBloodtype varchar(5) not null,

UType varchar(10) not null

Constraint Users\_PK primary key(userID)

);



-- Patient Table

Create table Patient

(

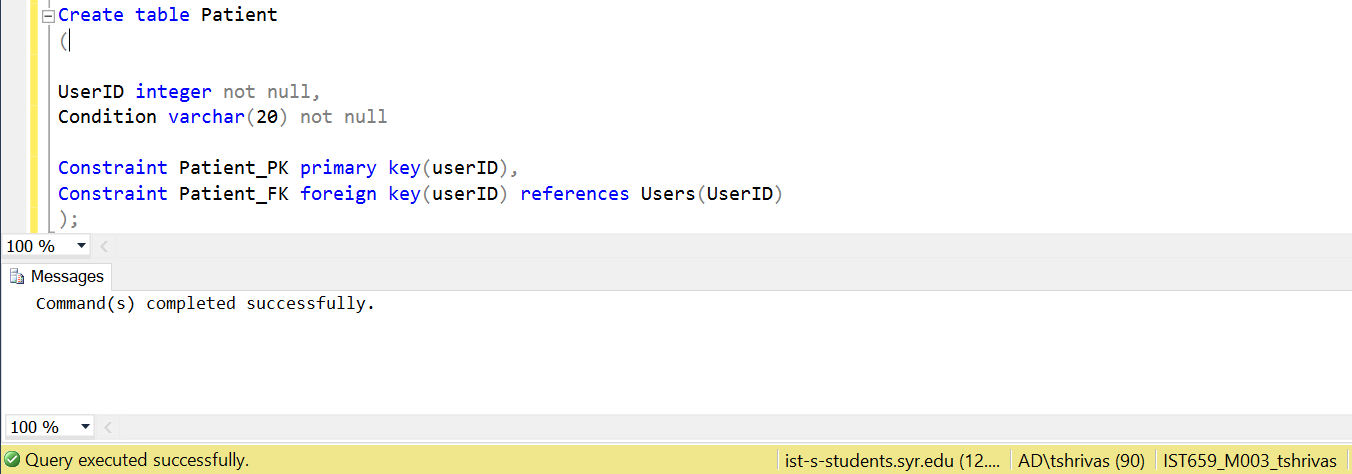
UserID integer not null,

Condition varchar(20) not null

Constraint Patient\_PK primary key(userID),

Constraint Patient\_FK foreign key(userID) references Users(UserID)

);



-- Donor Table

Create table Donor

(

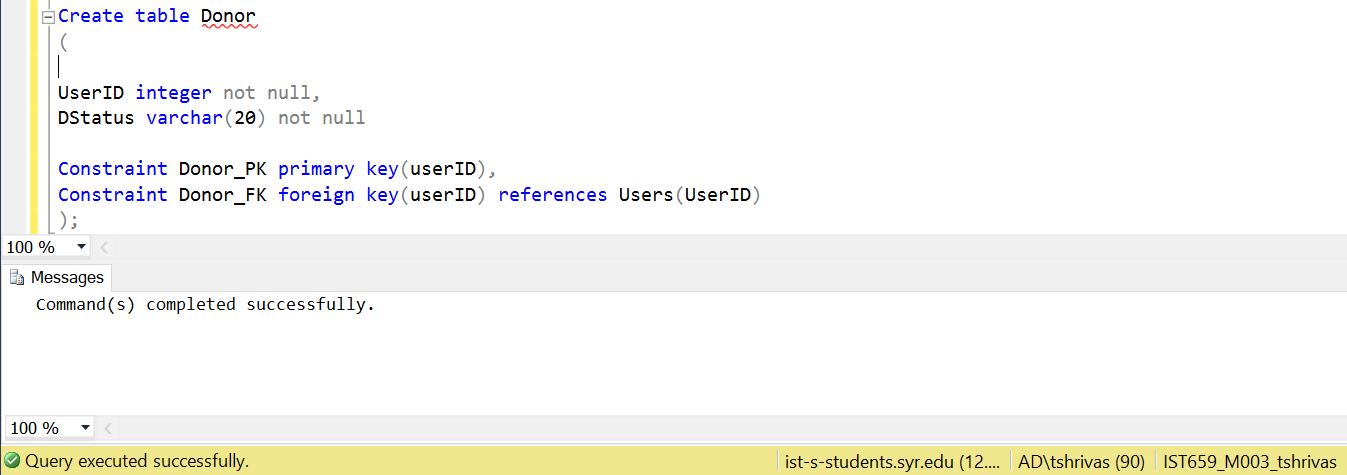
UserID integer not null,

DStatus varchar(20) not null

Constraint Donor\_PK primary key(userID),

Constraint Donor\_FK foreign key(userID) references Users(UserID)

);



-- Receiver Table

Create table Receiver

(

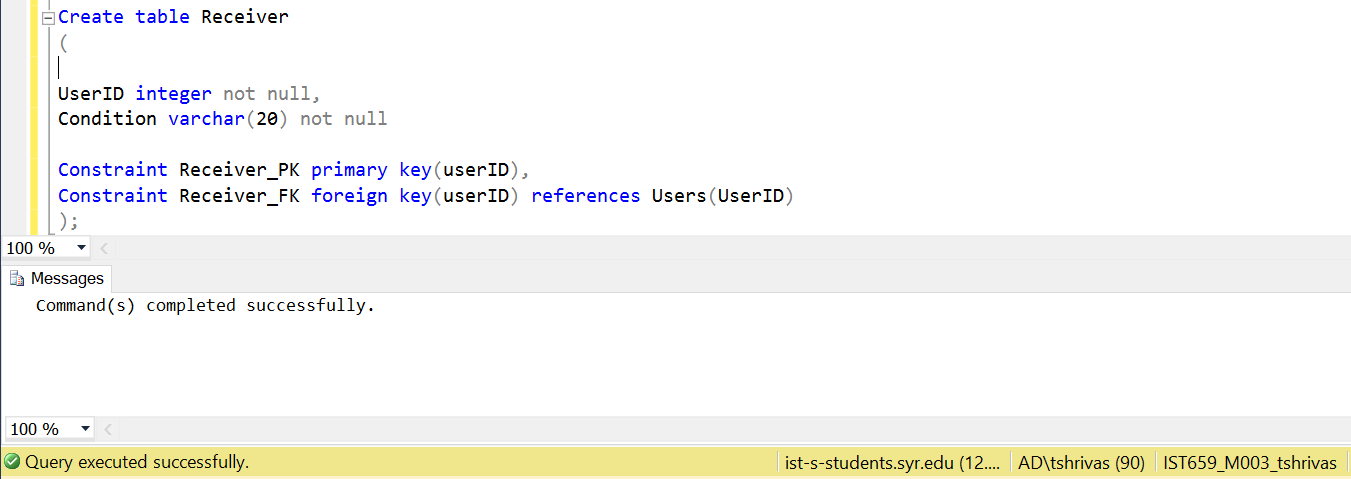
UserID integer not null,

Condition varchar(20) not null

Constraint Receiver\_PK primary key(userID),

Constraint Receiver\_FK foreign key(userID) references Users(UserID)

);



-- Patient Request Table

Create table PatientRequest

(

RequestID int not null,

UserID int not null,

EmployeeID int not null,

RequestType varchar(20) not null,

BloodType varchar(5) not null,

RequestStatus varchar(20),

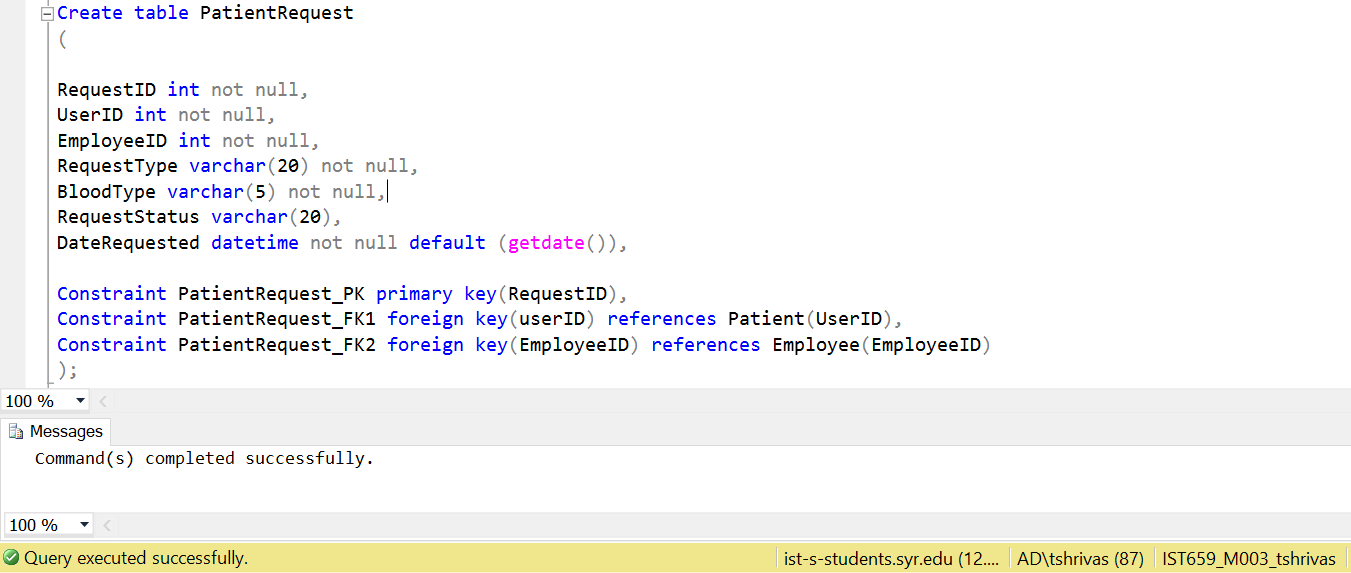
DateRequested datetime not null default (getdate()),

Constraint PatientRequest\_PK primary key(RequestID),

Constraint PatientRequest\_FK1 foreign key(userID) references Patient(UserID),

Constraint PatientRequest\_FK2 foreign key(EmployeeID) references Employee(EmployeeID)

);



-- Donor Registration Table

Create table DonorRegistration

(

RegistrationID int not null,

UserID int not null,

EmployeeID int not null,

DonationType varchar(20) not null,

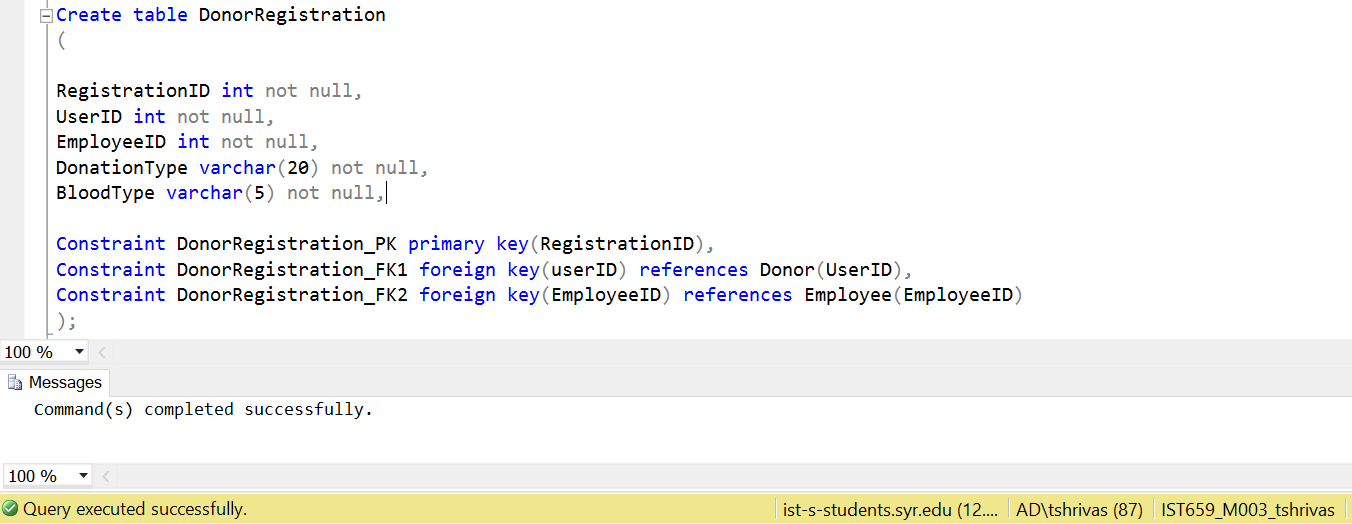
BloodType varchar(5) not null,

Constraint DonorRegistration\_PK primary key(RegistrationID),

Constraint DonorRegistration\_FK1 foreign key(userID) references Donor(UserID),

Constraint DonorRegistration\_FK2 foreign key(EmployeeID) references Employee(EmployeeID)

);



-- Employee Table

Create table Employee

(

EmployeeID int not null,

EFName varchar(20) not null,

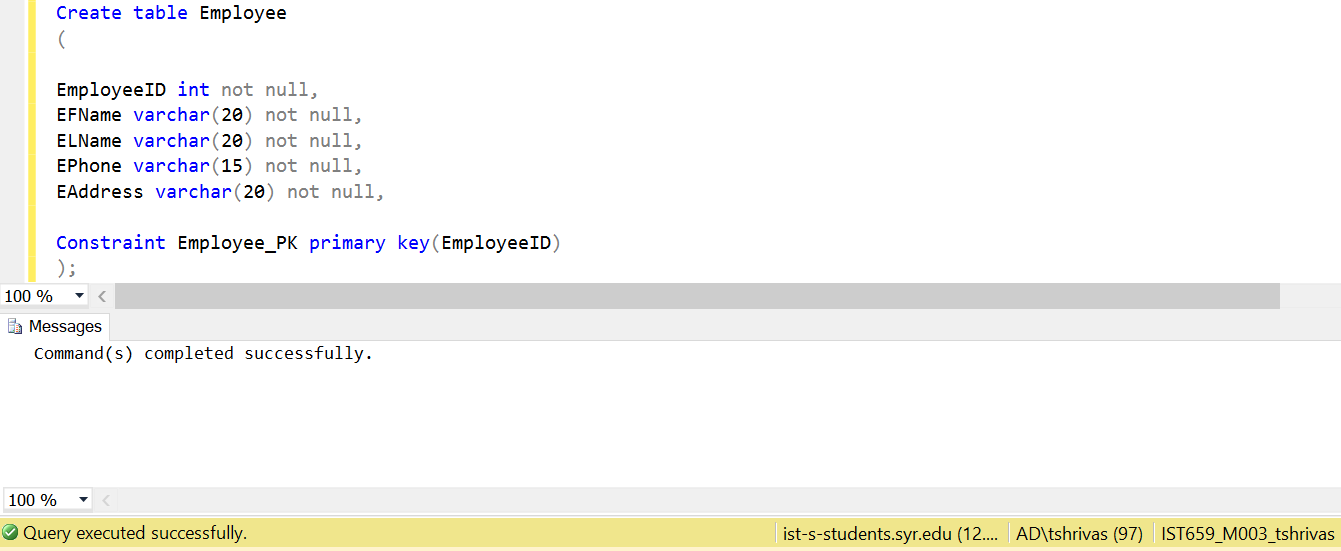
ELName varchar(20) not null,

EPhone varchar(15) not null,

EAddress varchar(20) not null,

Constraint Employee\_PK primary key(EmployeeID)

);



-- Donation Table

Create table Donation

(

DonationID int not null,

DonorID int not null,

ReceiverID int not null,

EmployeeID int not null,

DonationType varchar(20) not null,

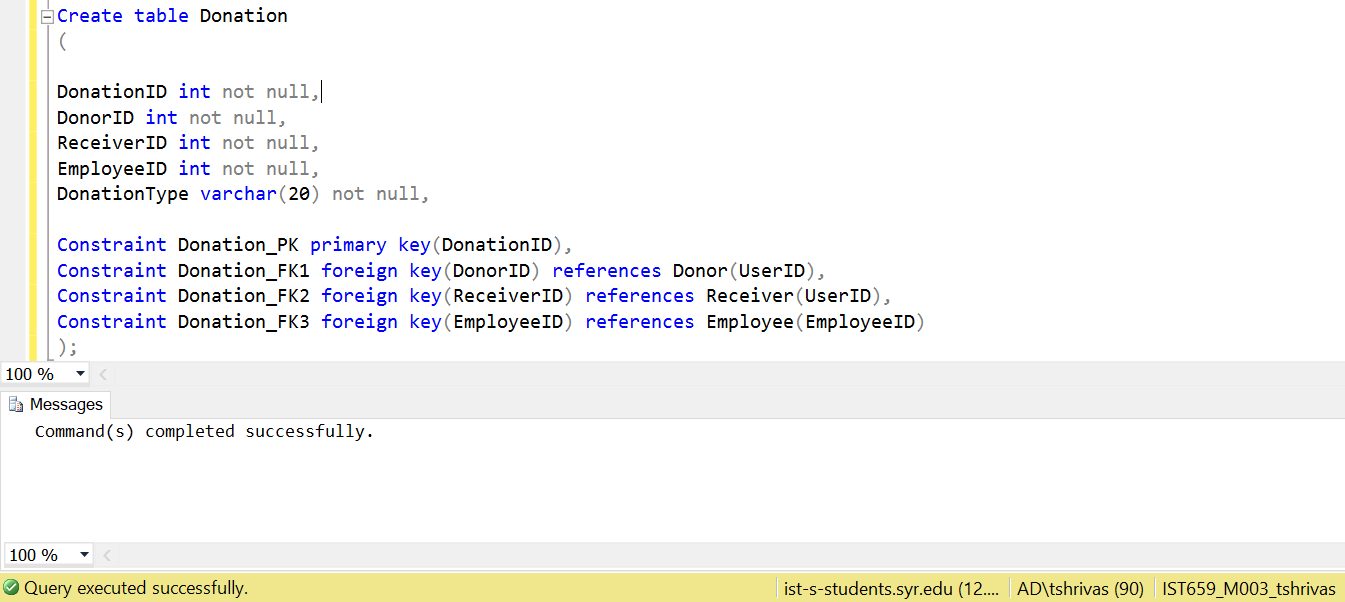
Constraint Donation\_PK primary key(DonationID),

Constraint Donation\_FK1 foreign key(DonorID) references Donor(UserID),

Constraint Donation\_FK2 foreign key(ReceiverID) references Receiver(UserID),

Constraint Donation\_FK3 foreign key(EmployeeID) references Employee(EmployeeID)

);



Inserting Values into Tables

-- populating Users

insert into Users values (1001,'Thomas','Cook',25,'315-672-3512','215 Westcott','Syracuse','NY','A+','Donor');

insert into Users values (1002,'Lily','Smith',37,'315-612-8767','734 University Ave','Syracuse','NY','O+','Donor');

insert into Users values (1003,'Amy','Welsh',24,'443-789-6877','89 Rock Place','Baltimore','MD','B+','Donor');

insert into Users values (1004,'Steve','Travis',28,'347-665-3565','5th Ave','NYC','NY','AB+','Donor');

insert into Users values (1005,'Andrea','Sacks',62,'315-452-7812','345 Thornton','Syracuse','NY','A-','Donor');

insert into Users values (1006,'Theo','Rhodes',49,'347-635-4646','23 Lexington Ave','NYC','NY','O-','Donor');

insert into Users values (1007,'Alice','Brown',19,'443-672-3512','34 North Drive','Baltimore','MD','B-','Donor');

insert into Users values (1008,'Kimberly','Bates',20,'443-685-3552','455 Apple Tree','Baltimore','MD','AB-','Donor');

insert into Users values (1009,'Bran','Whitte',23,'445-267-4556','58 Dupont Circle','Maryland City','MD','A-','Donor');

insert into Users values (1010,'Fred','Atkins',43,'445-873-9742','23 South Drive','Maryland City','MD','O-','Donor');

insert into Users values (1011,'Abe','Thompson',45,'315-672-6757','678 Westcott','Syracuse','NY','A-','Patient');

insert into Users values (1012,'Riley','Williams',55,'315-667-8767','325 University Ave','Syracuse','NY','O-','Patient');

insert into Users values (1013,'Isabella','Stewart',23,'443-744-7749','238 Rock Place','Baltimore','MD','B+','Patient');

insert into Users values (1014,'Noah','Reed',65,'347-667-5366','6th Ave','NYC','NY','AB+','Patient');

insert into Users values (1015,'Oliver','Coffey',45,'315-783-7882','887 Thornton','Syracuse','NY','A+','Patient');

insert into Users values (1016,'William','Darby',50,'347-655-4589','382 Lexington Ave','NYC','NY','O+','Patient');

insert into Users values (1017,'Mia','Moss',22,'443-645-4789','728 North Drive','Baltimore','MD','B-','Patient');

insert into Users values (1018,'Ben','Floyd',17,'443-366-5820','33 Apple Tree','Baltimore','MD','AB-','Patient');

insert into Users values (1019,'Mason','Powell',32,'445-209-4840','38 Dupont Circle','Maryland City','MD','A+','Patient');

insert into Users values (1020,'Lucas','Bennett',39,'445-087-9742','563 South Drive','Maryland City','MD','O+','Patient');

insert into Users values (1021,'Olivia','Smith',35,'315-676-2447','678 Westcott','Syracuse','NY','A-','Receiver');

insert into Users values (1022,'Ava','Jhonson',32,'315-234-4425','325 University Ave','Syracuse','NY','O-','Receiver');

insert into Users values (1023,'James','Jones',55,'443-754-4478','238 Rock Place','Baltimore','MD','B+','Receiver');

insert into Users values (1024,'Michael','Brown',16,'347-447-7589','6th Ave','NYC','NY','AB+','Receiver');

insert into Users values (1025,'Amelia','Davis',24,'315-478-3990','887 Thornton','Syracuse','NY','A+','Receiver');

insert into Users values (1026,'John','Miller',67,'347-477-5900','382 Lexington Ave','NYC','NY','O+','Receiver');

insert into Users values (1027,'Sophia','Cyrus',27,'443-282-1119','728 North Drive','Baltimore','MD','B-','Receiver');

insert into Users values (1028,'Mary','Wilson',83,'443-126-1220','33 Apple Tree','Baltimore','MD','AB-','Receiver');

insert into Users values (1029,'Emma','Turner',56,'445-119-4221','38 Dupont Circle','Maryland City','MD','A+','Receiver');

insert into Users values (1030,'Liam','Cooper',37,'445-031-9001','563 South Drive','Maryland City','MD','O+','Receiver');

insert into Users values (1031,'Tara','James',33,'315-676-1652','238 Thornton','Syracuse','NY','A-','Donor');

insert into Users values (1032,'Ash','Damons',23,'315-234-4783','82 Armoury Square','Syracuse','NY','O-','Donor');

insert into Users values (1033,'Fred','Philips',56,'443-735-3378','88 NorthDrive','Baltimore','MD','O-','Donor');

insert into Users values (1034,'Nick','Jordan',24,'347-444-4259','223 Union Square','NYC','NY','AB+','Donor');

insert into Users values (1035,'Patrick','Noel',56,'315-368-0090','790 University Ave','Syracuse','NY','A+','Donor');

insert into Users values (1036,'Lucas','Mayer',57,'347-470-2300','319 Madison Square','NYC','NY','O-','Donor');

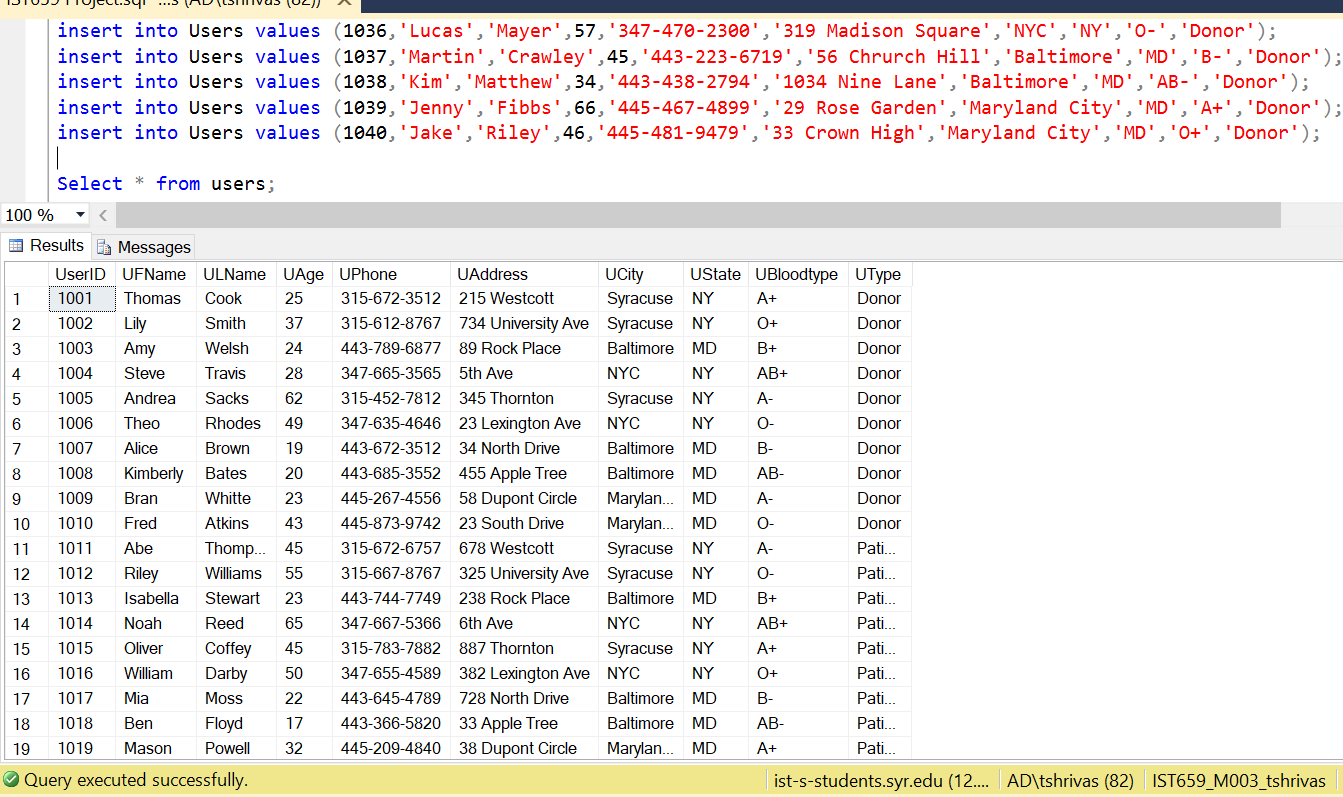
insert into Users values (1037,'Martin','Crawley',45,'443-223-6719','56 Chrurch Hill','Baltimore','MD','B-','Donor');

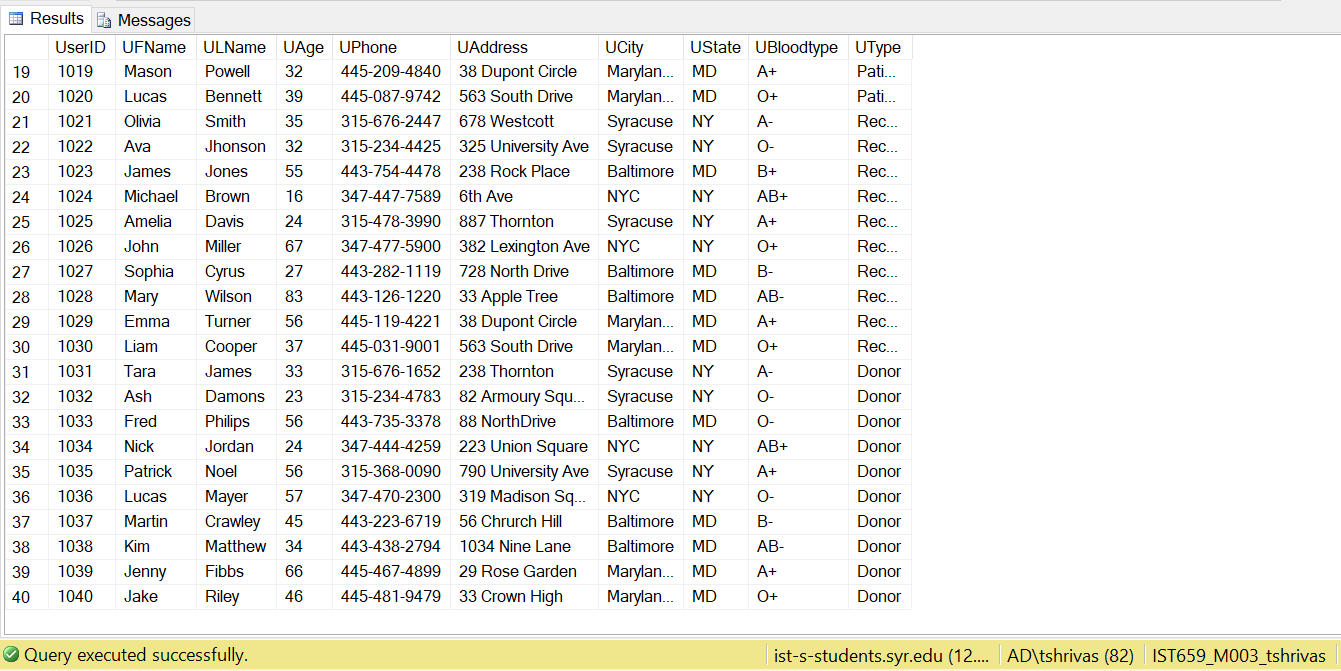
insert into Users values (1038,'Kim','Matthew',34,'443-438-2794','1034 Nine Lane','Baltimore','MD','AB-','Donor');

insert into Users values (1039,'Jenny','Fibbs',66,'445-467-4899','29 Rose Garden','Maryland City','MD','A+','Donor');

insert into Users values (1040,'Jake','Riley',46,'445-481-9479','33 Crown High','Maryland City','MD','O+','Donor');

Select \* from users;





-- populating Patient

insert into Patient values (1011, 'Critical');

insert into Patient values (1012, 'Stable');

insert into Patient values (1013, 'Serious');

insert into Patient values (1014, 'Serious');

insert into Patient values (1015, 'Stable');

insert into Patient values (1016, 'Critical');

insert into Patient values (1017, 'Serious');

insert into Patient values (1018, 'Stable');

insert into Patient values (1019, 'Stable');

insert into Patient values (1020, 'Intensive');

Select \* from Patient;



-- populating Donor

insert into Donor values (1001, 'Living');

insert into Donor values (1002, 'Living');

insert into Donor values (1003, 'Living');

insert into Donor values (1004, 'Living');

insert into Donor values (1005, 'Deceased');

insert into Donor values (1006, 'Living');

insert into Donor values (1007, 'Living');

insert into Donor values (1008, 'Living');

insert into Donor values (1009, 'Deceased');

insert into Donor values (1010, 'Living');

insert into Donor values (1031, 'Living');

insert into Donor values (1032, 'Living');

insert into Donor values (1033, 'Living');

insert into Donor values (1034, 'Living');

insert into Donor values (1035, 'Living');

insert into Donor values (1036, 'Living');

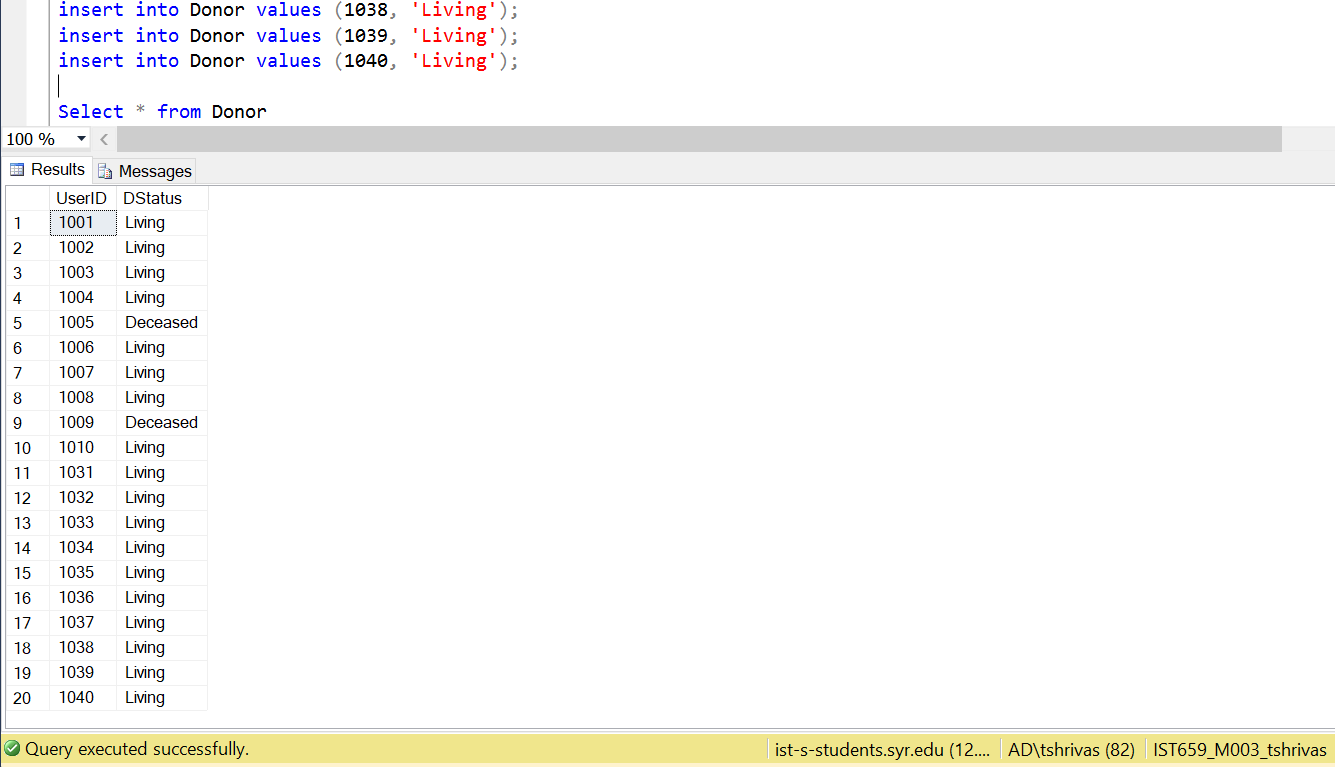
insert into Donor values (1037, 'Living');

insert into Donor values (1038, 'Living');

insert into Donor values (1039, 'Living');

insert into Donor values (1040, 'Living');

Select \* from Donor



-- populating Receiver

insert into Receiver values (1021, 'Stable');

insert into Receiver values (1022, 'Stable');

insert into Receiver values (1023, 'Stable');

insert into Receiver values (1024, 'Stable');

insert into Receiver values (1025, 'Critical');

insert into Receiver values (1026, 'Stable');

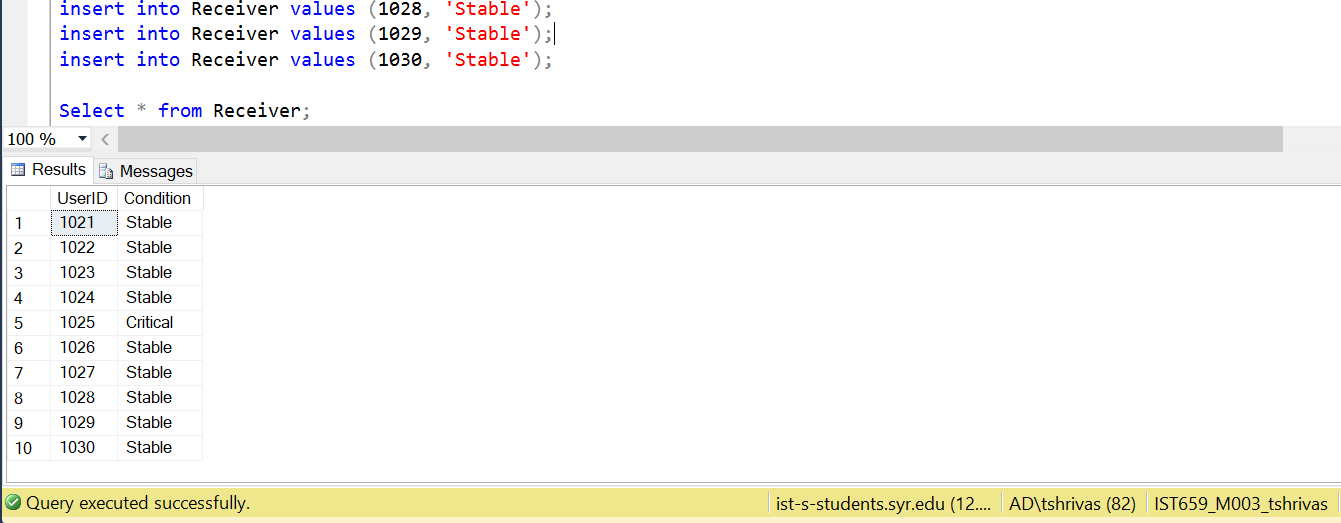
insert into Receiver values (1027, 'Stable');

insert into Receiver values (1028, 'Stable');

insert into Receiver values (1029, 'Stable');

insert into Receiver values (1030, 'Stable');

Select \* from Receiver;



-- populating Employee

insert into Employee values (2001,'Rebecca','Winters','315-334-3110','515 Maryland Ave');

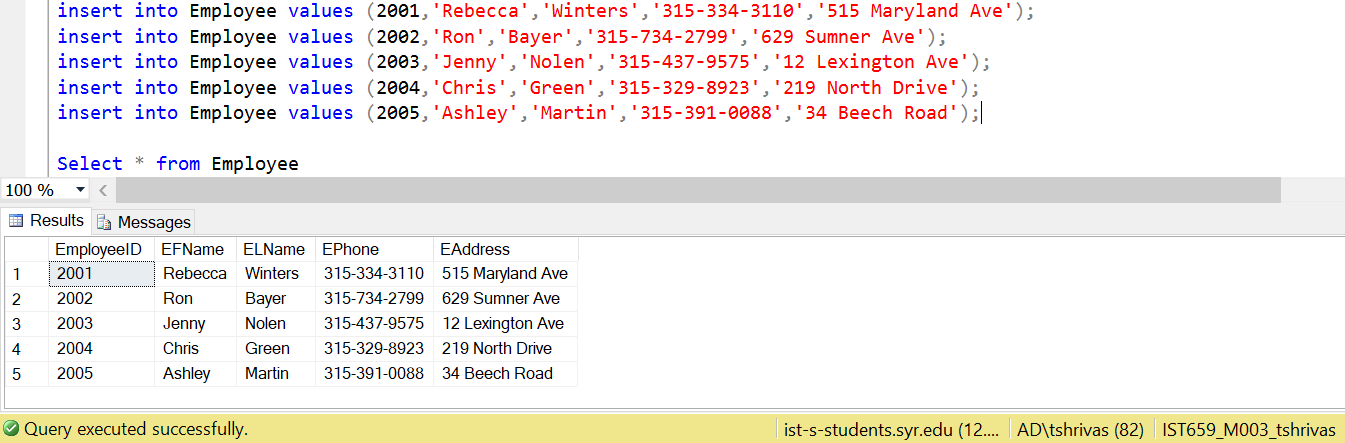
insert into Employee values (2002,'Ron','Bayer','315-734-2799','629 Sumner Ave');

insert into Employee values (2003,'Jenny','Nolen','315-437-9575','12 Lexington Ave');

insert into Employee values (2004,'Chris','Green','315-329-8923','219 North Drive');

insert into Employee values (2005,'Ashley','Martin','315-391-0088','34 Beech Road');

Select \* from Employee;



-- populating DonorRegisteration

insert into DonorRegistration values (3001,1001,2001,'Blood','A+');

insert into DonorRegistration values (3002,1002,2002,'Blood','O+');

insert into DonorRegistration values (3003,1003,2003,'PartLiver','B+');

insert into DonorRegistration values (3004,1004,2004,'Blood','AB+');

insert into DonorRegistration values (3005,1005,2005,'Heart','A-');

insert into DonorRegistration values (3006,1006,2001,'PartLiver','O-');

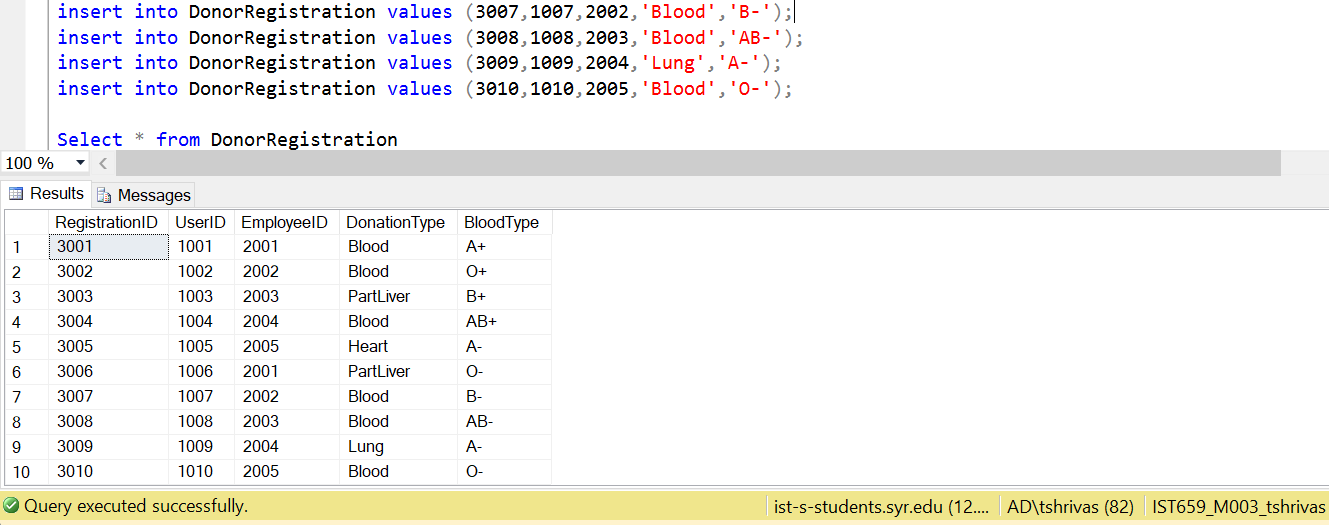
insert into DonorRegistration values (3007,1007,2002,'Blood','B-');

insert into DonorRegistration values (3008,1008,2003,'Blood','AB-');

insert into DonorRegistration values (3009,1009,2004,'Lung','A-');

insert into DonorRegistration values (3010,1010,2005,'Blood','O-');

Select \* from DonorRegistration



-- populating PatientRequest

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4001,1011,2001,'Blood','A-');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4002,1012,2001,'Blood','O-');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4003,1013,2001,'Blood','B+');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4004,1014,2002,'Heart','AB+');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4005,1015,2003,'PartLiver','A+');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4006,1016,2004,'PartLiver','O+');

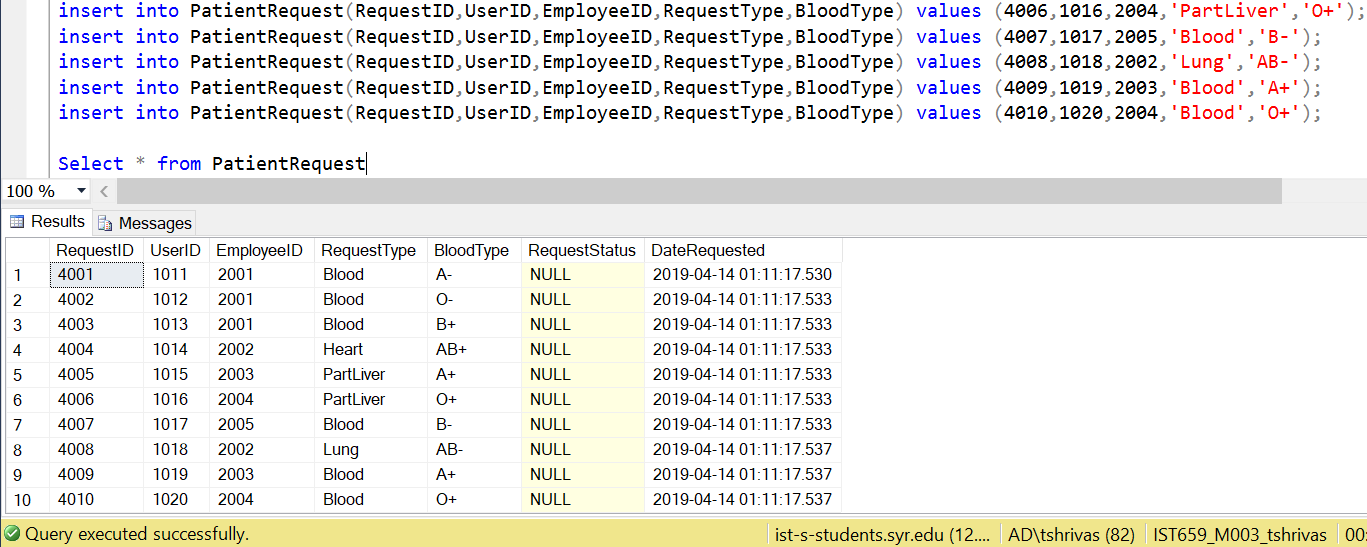
insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4007,1017,2005,'Blood','B-');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4008,1018,2002,'Lung','AB-');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4009,1019,2003,'Blood','A+');

insert into PatientRequest(RequestID,UserID,EmployeeID,RequestType,BloodType) values (4010,1020,2004,'Blood','O+');

Select \* from PatientRequest



-- populating Donations

insert into Donation values (5001,1031,1021,2001,'Blood');

insert into Donation values (5002,1032,1022,2002,'Blood');

insert into Donation values (5003,1033,1023,2003,'Blood');

insert into Donation values (5004,1034,1024,2004,'Blood');

insert into Donation values (5005,1035,1025,2005,'Blood');

insert into Donation values (5006,1036,1026,2001,'Blood');

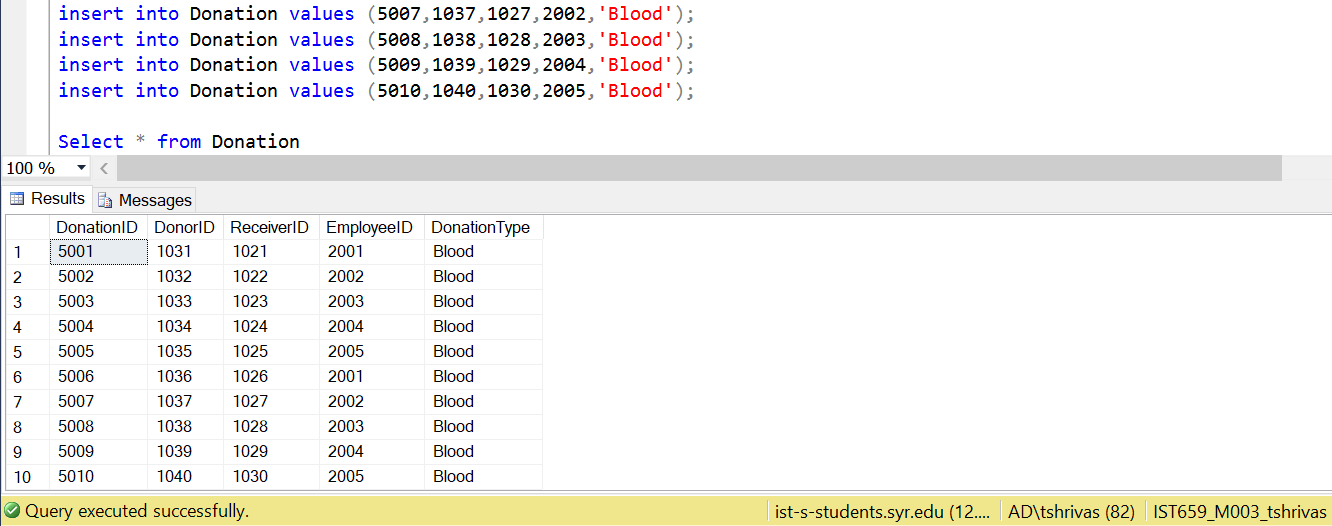
insert into Donation values (5007,1037,1027,2002,'Blood');

insert into Donation values (5008,1038,1028,2003,'Blood');

insert into Donation values (5009,1039,1029,2004,'Blood');

insert into Donation values (5010,1040,1030,2005,'Blood');

Select \* from Donation



Major Data Questions

-- MATCHING DONORS AND PATIENTS

-- creating blood constrain table

create table bloodchart

(

DonorBloodtype varchar(5) not null,

ReceiverBloodtype varchar(5) not null

);

-- inserting values according to the blood constraint

insert into bloodchart values('A+','A+');

insert into bloodchart values('A+','AB+');

insert into bloodchart values('O+','O+');

insert into bloodchart values('O+','A+');

insert into bloodchart values('O+','B+');

insert into bloodchart values('O+','AB+');

insert into bloodchart values('B+','B+');

insert into bloodchart values('B+','AB+');

insert into bloodchart values('AB+','AB+');

insert into bloodchart values('A-','A+');

insert into bloodchart values('A-','A-');

insert into bloodchart values('A-','AB+');

insert into bloodchart values('A-','AB-');

insert into bloodchart values('O-','A+');

insert into bloodchart values('O-','O+');

insert into bloodchart values('O-','B+');

insert into bloodchart values('O-','AB+');

insert into bloodchart values('O-','A-');

insert into bloodchart values('O-','O-');

insert into bloodchart values('O-','B-');

insert into bloodchart values('O-','AB-');

insert into bloodchart values('B-','B+');

insert into bloodchart values('B-','B-');

insert into bloodchart values('B-','AB+');

insert into bloodchart values('B-','AB-');

insert into bloodchart values('AB-','AB+');

insert into bloodchart values('AB-','AB-');

select \* from bloodchart

-- CREATING A VIEW FOR THE MATCH OF DONOR AND PATIENTS

-- join of donor register with bloodchart by Donor's bloodtype

-- join of the above with Patient request by Patient bloo

GO

create view matchblood as

(select dr.UserID as DonorID,bloodchart.DonorBloodtype,dr.DonationType,

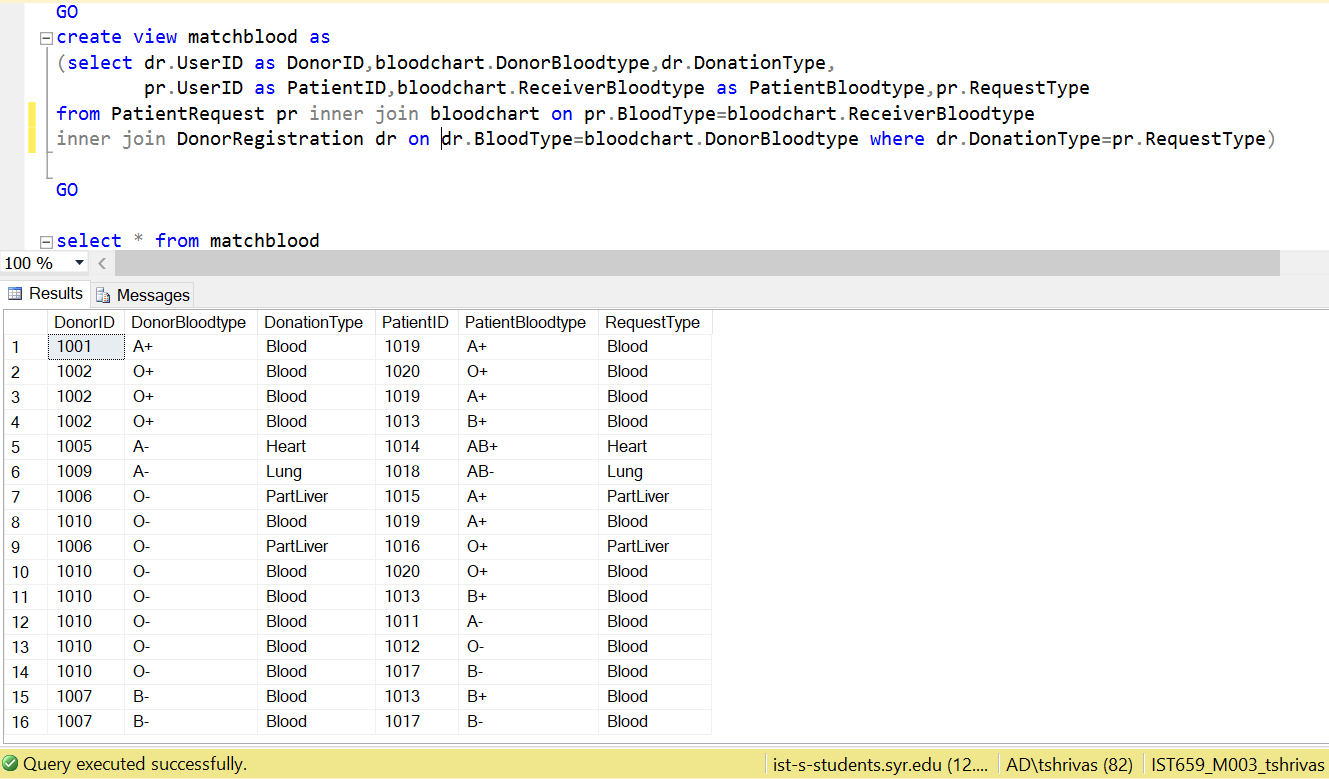
pr.UserID as PatientID,bloodchart.ReceiverBloodtype as PatientBloodtype,pr.RequestType

from PatientRequest pr inner join bloodchart on pr.BloodType=bloodchart.ReceiverBloodtype inner join DonorRegistration dr on

dr.BloodType=bloodchart.DonorBloodtype where dr.DonationType=pr.RequestType)

GO

select \* from matchblood



**Question 1**

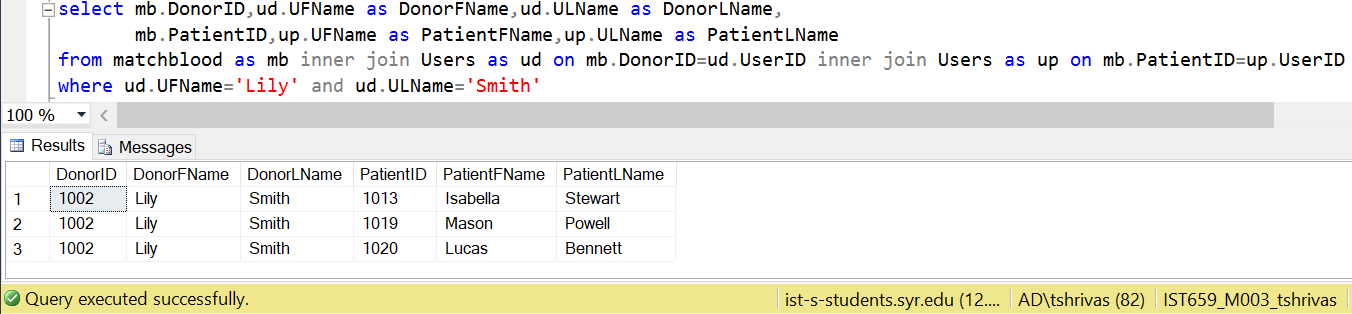
-- Given a Donor(Lily Smith) , which Receivers can accept the Donation?

select mb.DonorID,ud.UFName as DonorFName,ud.ULName as DonorLName,

mb.PatientID,up.UFName as PatientFName,up.ULName as PatientLName

from matchblood as mb inner join Users as ud on mb.DonorID=ud.UserID inner join Users as up on mb.PatientID=up.UserID

where ud.UFName='Lily' and ud.ULName='Smith'



**Question 2**

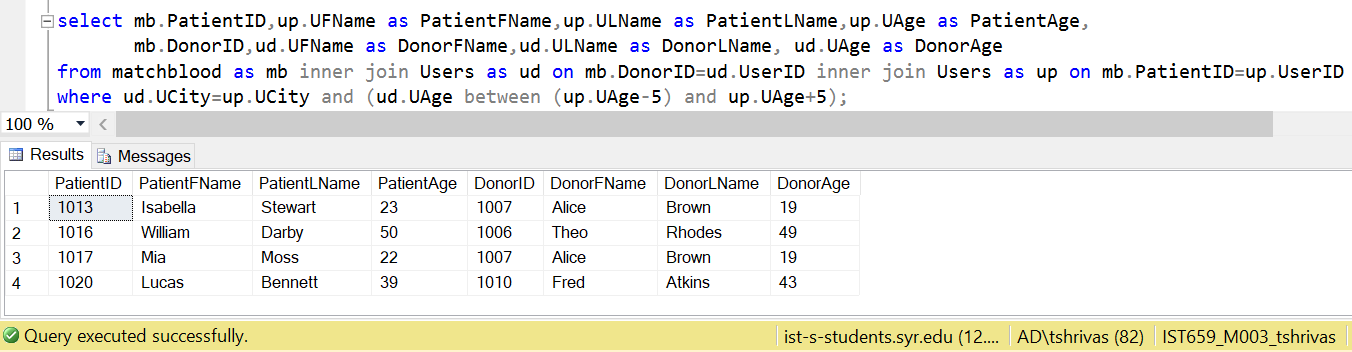
-- Given a Receiver who wants a donor around their age, which Donors can they accept the Donations from?

select mb.PatientID,up.UFName as PatientFName,up.ULName as PatientLName,up.UAge as PatientAge,

mb.DonorID,ud.UFName as DonorFName,ud.ULName as DonorLName, ud.UAge as DonorAge

from matchblood as mb inner join Users as ud on mb.DonorID=ud.UserID inner join Users as up on mb.PatientID=up.UserID

where ud.UCity=up.UCity and (ud.UAge between (up.UAge-5) and up.UAge+5);

****

**Question 3**

-- What are the matches with Location constraint(same city) in place?

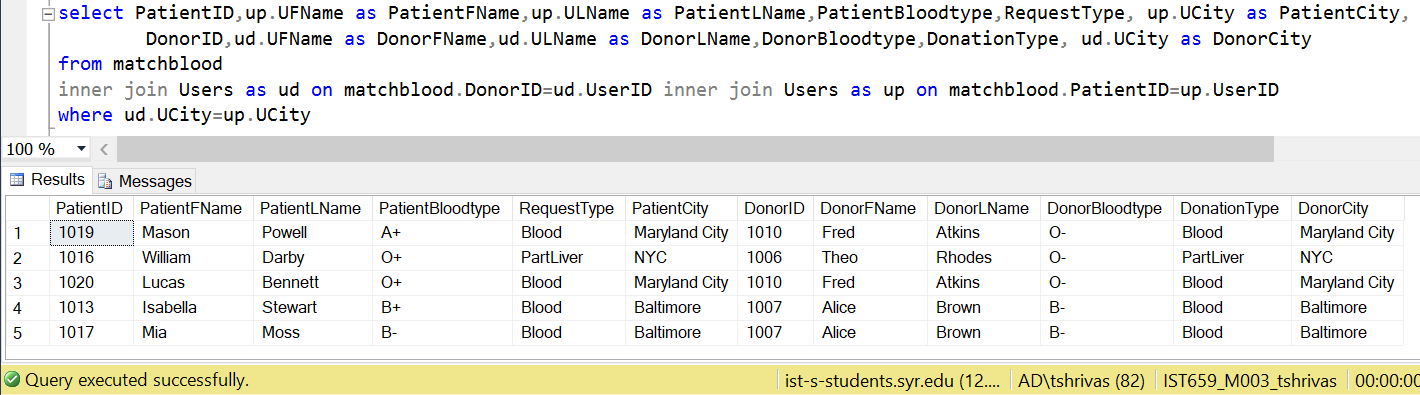
select PatientID,up.UFName as PatientFName,up.ULName as PatientLName,PatientBloodtype,RequestType, up.UCity as PatientCity,

DonorID,ud.UFName as DonorFName,ud.ULName as DonorLName,DonorBloodtype,DonationType, ud.UCity as DonorCity

from matchblood

inner join Users as ud on matchblood.DonorID=ud.UserID inner join Users as up on matchblood.PatientID=up.UserID

where ud.UCity=up.UCity



**Question 4**

-- Find a match for patients in critical condition.

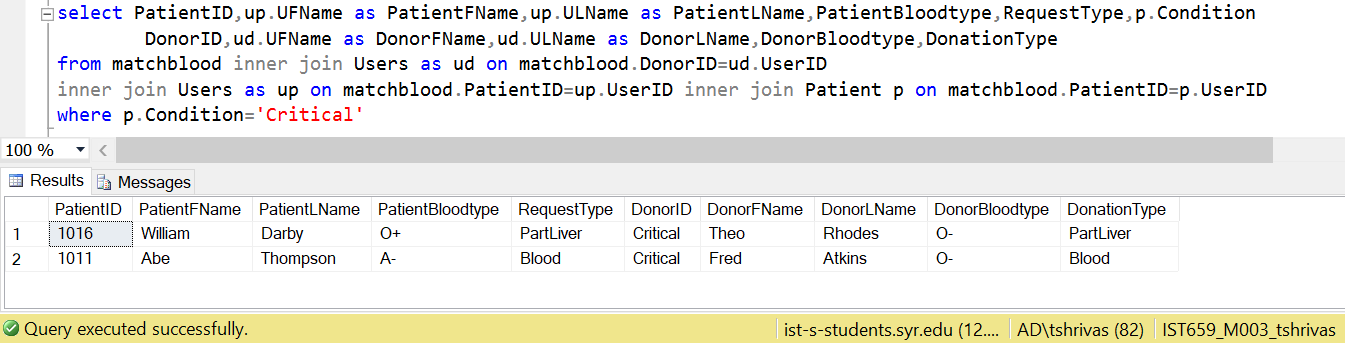
select PatientID,up.UFName as PatientFName,up.ULName as PatientLName,PatientBloodtype,RequestType,p.Condition

DonorID,ud.UFName as DonorFName,ud.ULName as DonorLName,DonorBloodtype,DonationType

from matchblood inner join Users as ud on matchblood.DonorID=ud.UserID

inner join Users as up on matchblood.PatientID=up.UserID inner join Patient p on matchblood.PatientID=p.UserID

where p.Condition='Critical'



**Question 5**

-- Find a match for the patient has been the longest on the waiting list?

select PatientID,up.UFName as PatientFName,up.ULName as PatientLName,PatientBloodtype,RequestType,

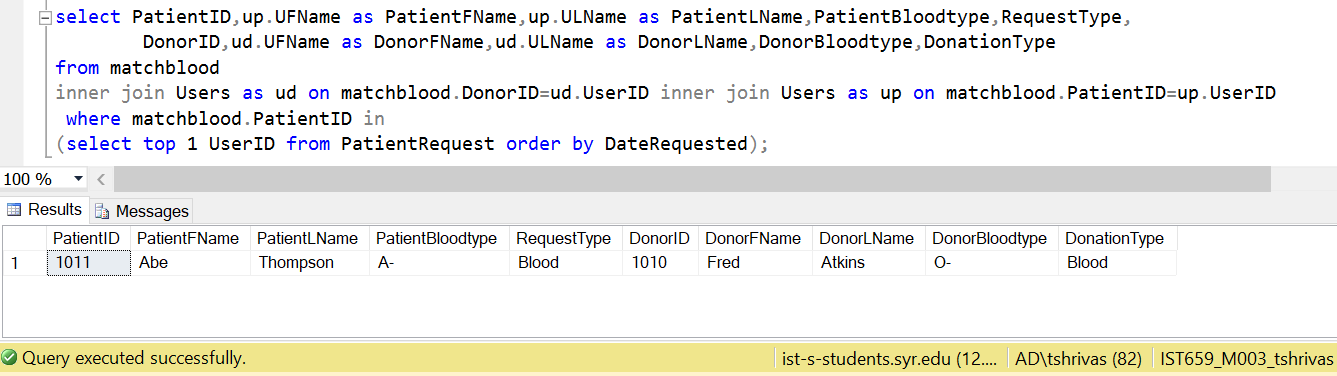
DonorID,ud.UFName as DonorFName,ud.ULName as DonorLName,DonorBloodtype,DonationType

from matchblood

inner join Users as ud on matchblood.DonorID=ud.UserID inner join Users as up on matchblood.PatientID=up.UserID

where matchblood.PatientID in

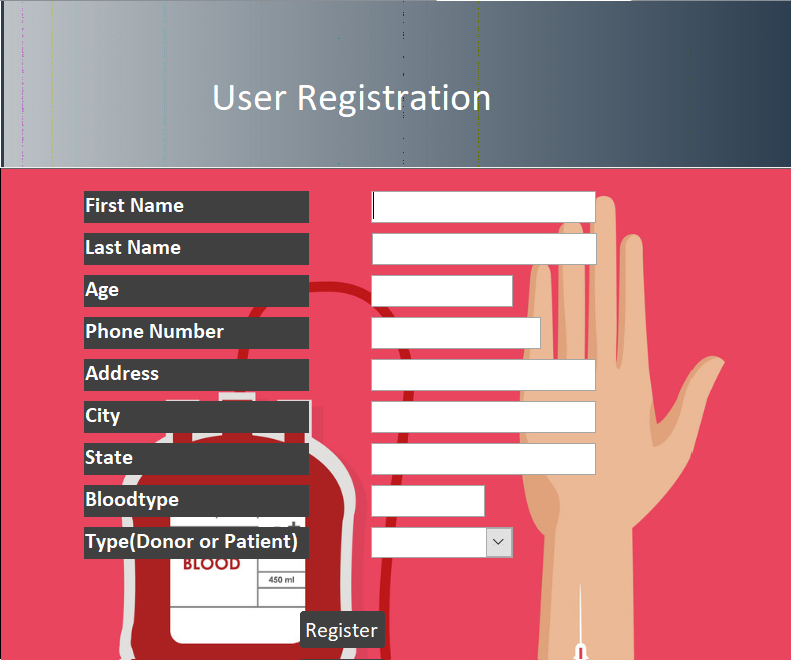
(select top 1 UserID from PatientRequest order by DateRequested);



Implementation - Forms

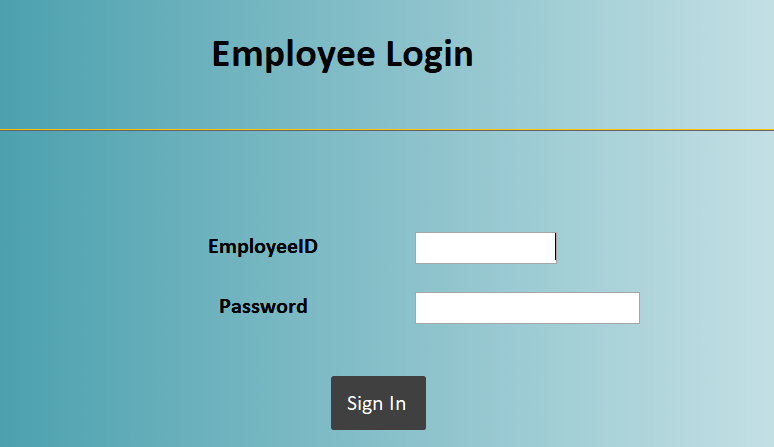
1. **User Registration Form-**

Takes in the User details and adds Records to the User Table.

****

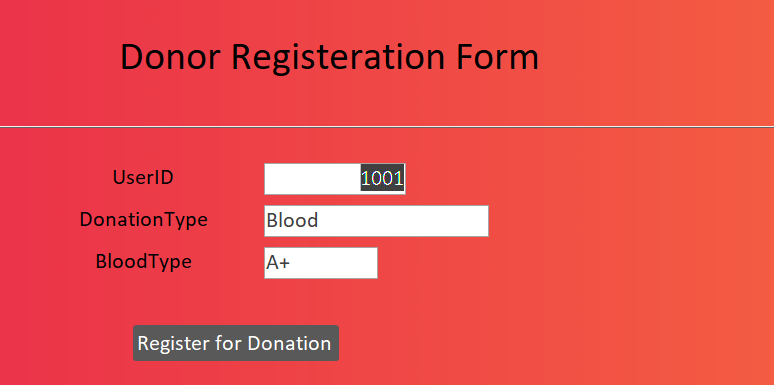
1. **Employee Login-**

Logs in an Employee and redirects them to the Matching Page.

****

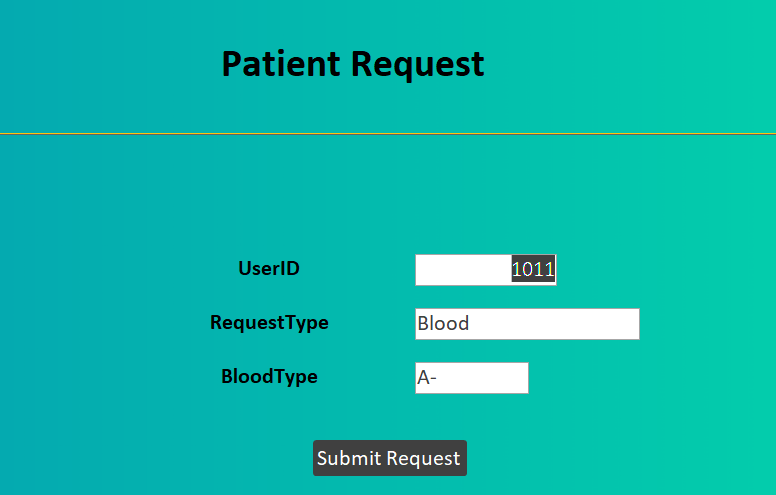
1. **Donor Registration Form-**

Takes in the Donor Registration Details and adds them as an extra record in the Donor Registration Table.

****

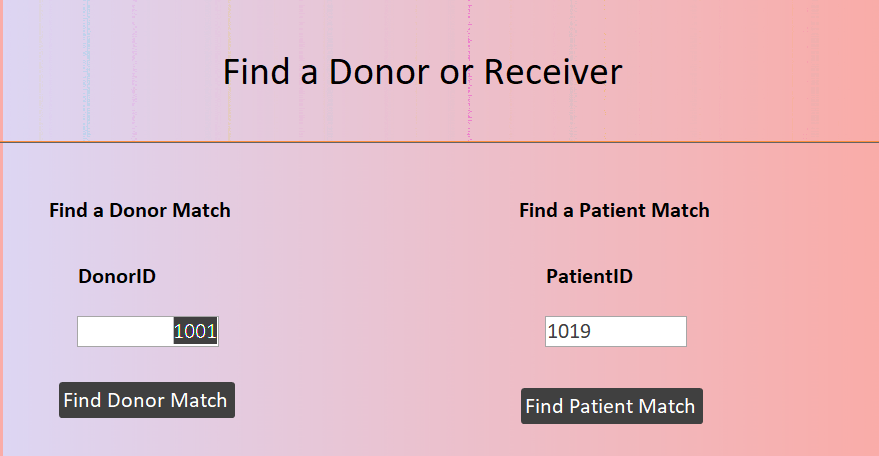
1. **Patient Request Form-**

Takes in the Patient Request for a Donation and adds the record to the Patient Request Table.

****

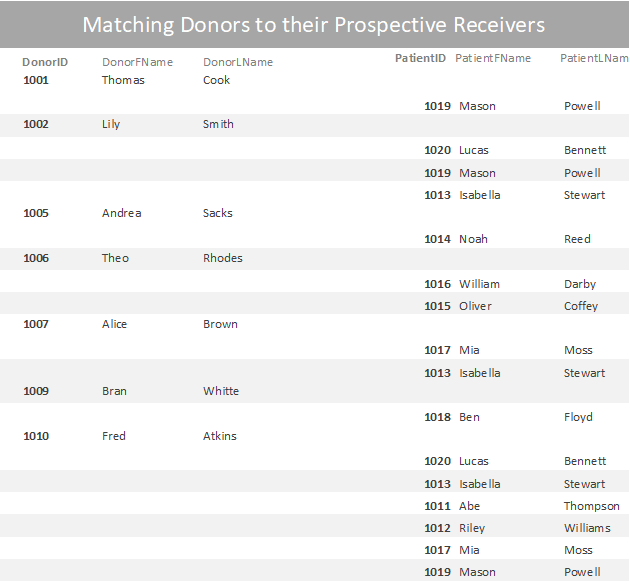
1. **Employee Match Form-**

Finds a match for either a donor or a patient depending on the form the employee fills and the button he/she presses.

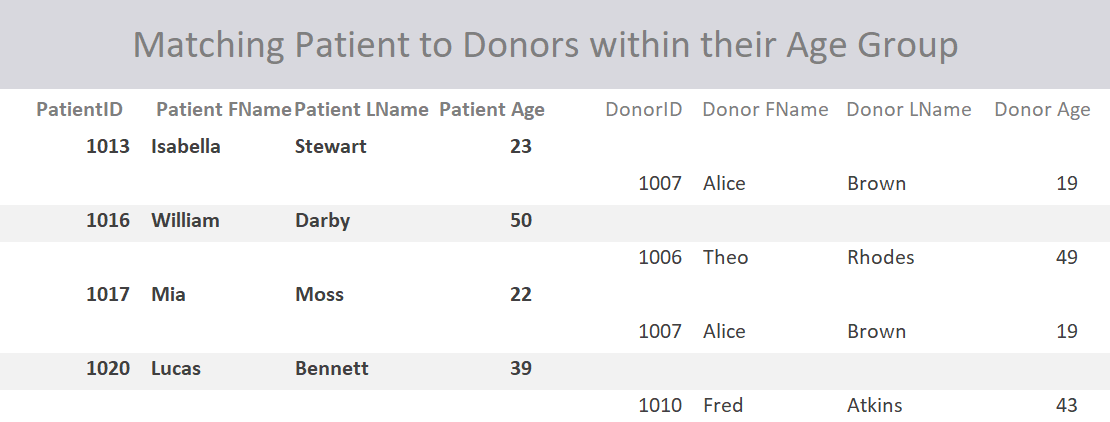
****

Implementation - Reports

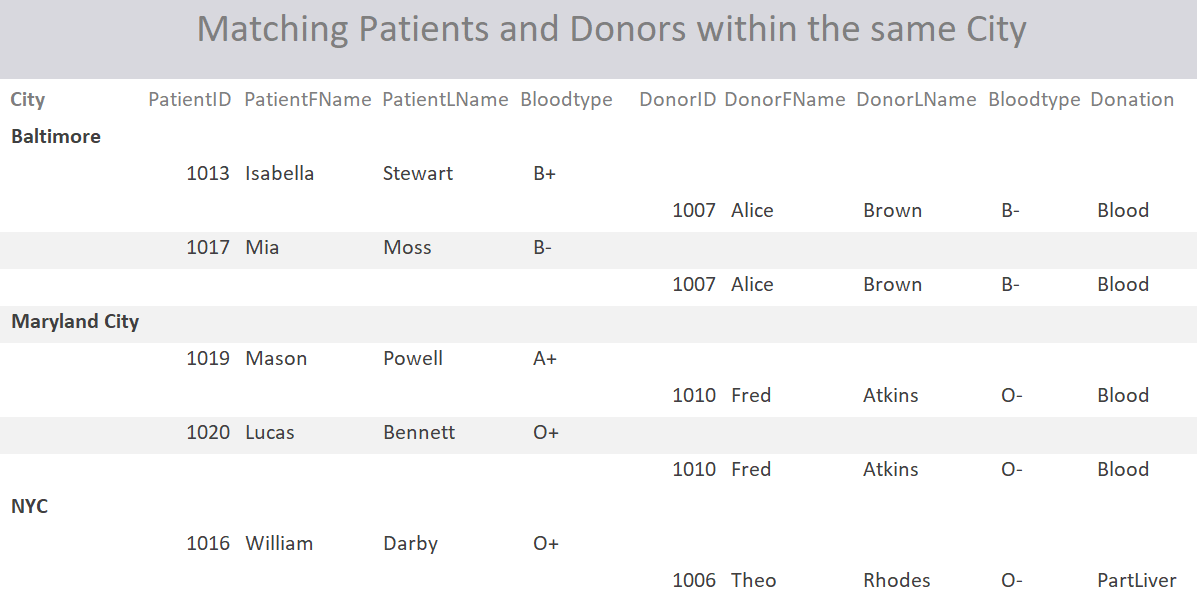
1. **Matching Donors to prospective Receivers (Patients)**

****

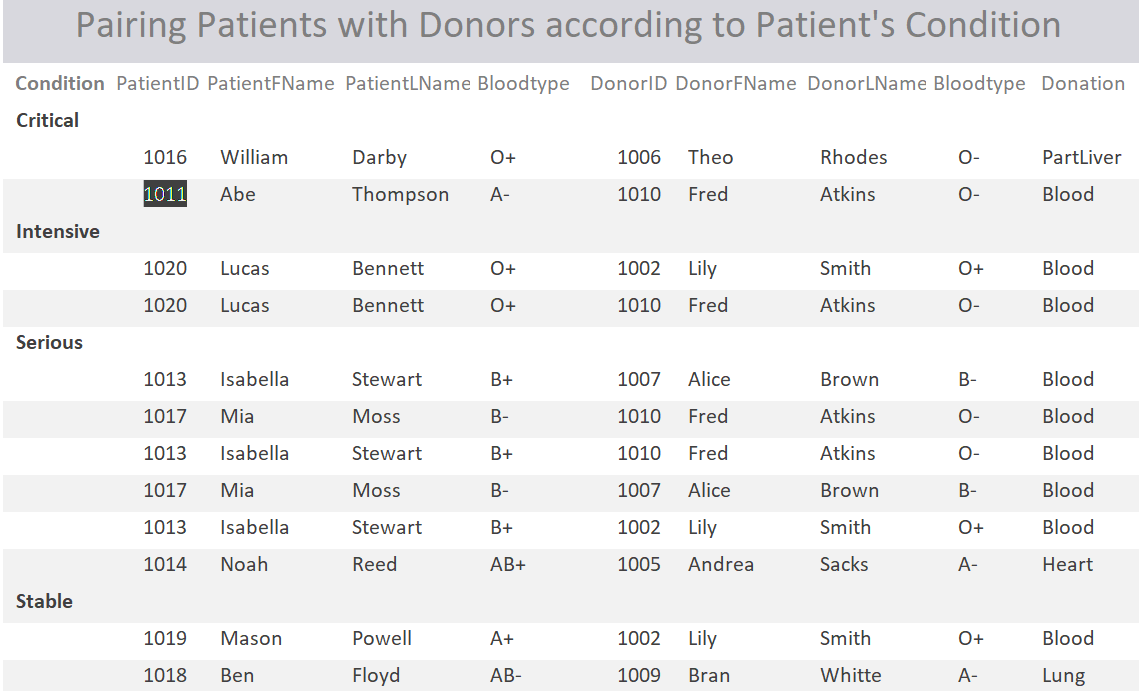
1. **Matching Patients and Donors with respect to Age Groups**

****

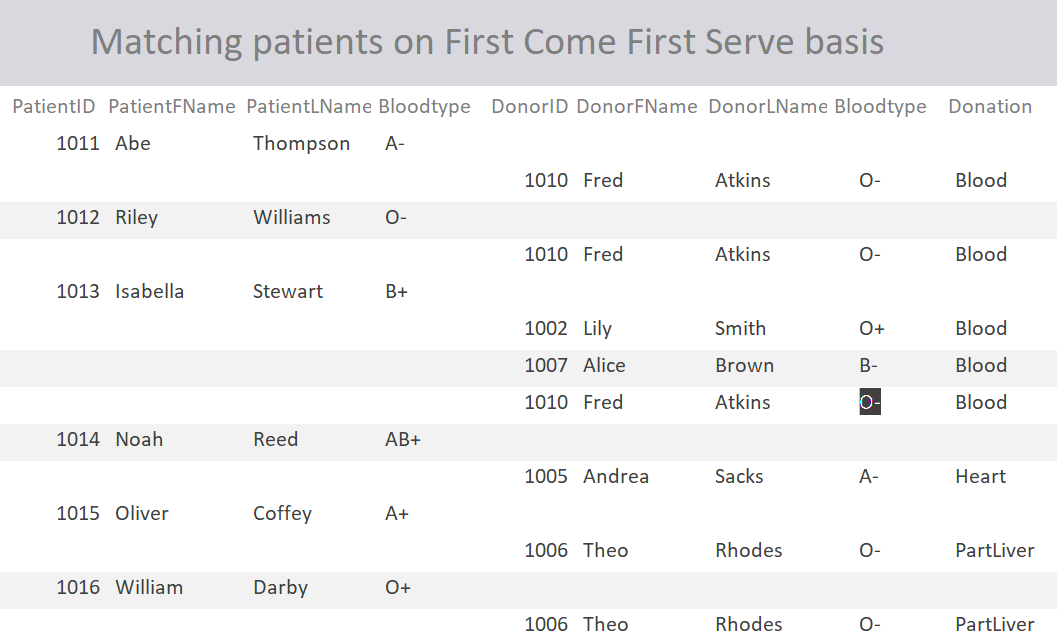
1. **City-wise Matching**



1. **Finding Matches depending on the Patient’s Condition**

****

1. **Finding Matches on First Come First Serve Basis for Patients**

****