Database Management Systems

Organ Donation Database

Team members:
Nandana Kandathil Rajeev 207248
Samhitha Reddy Vedire 207167

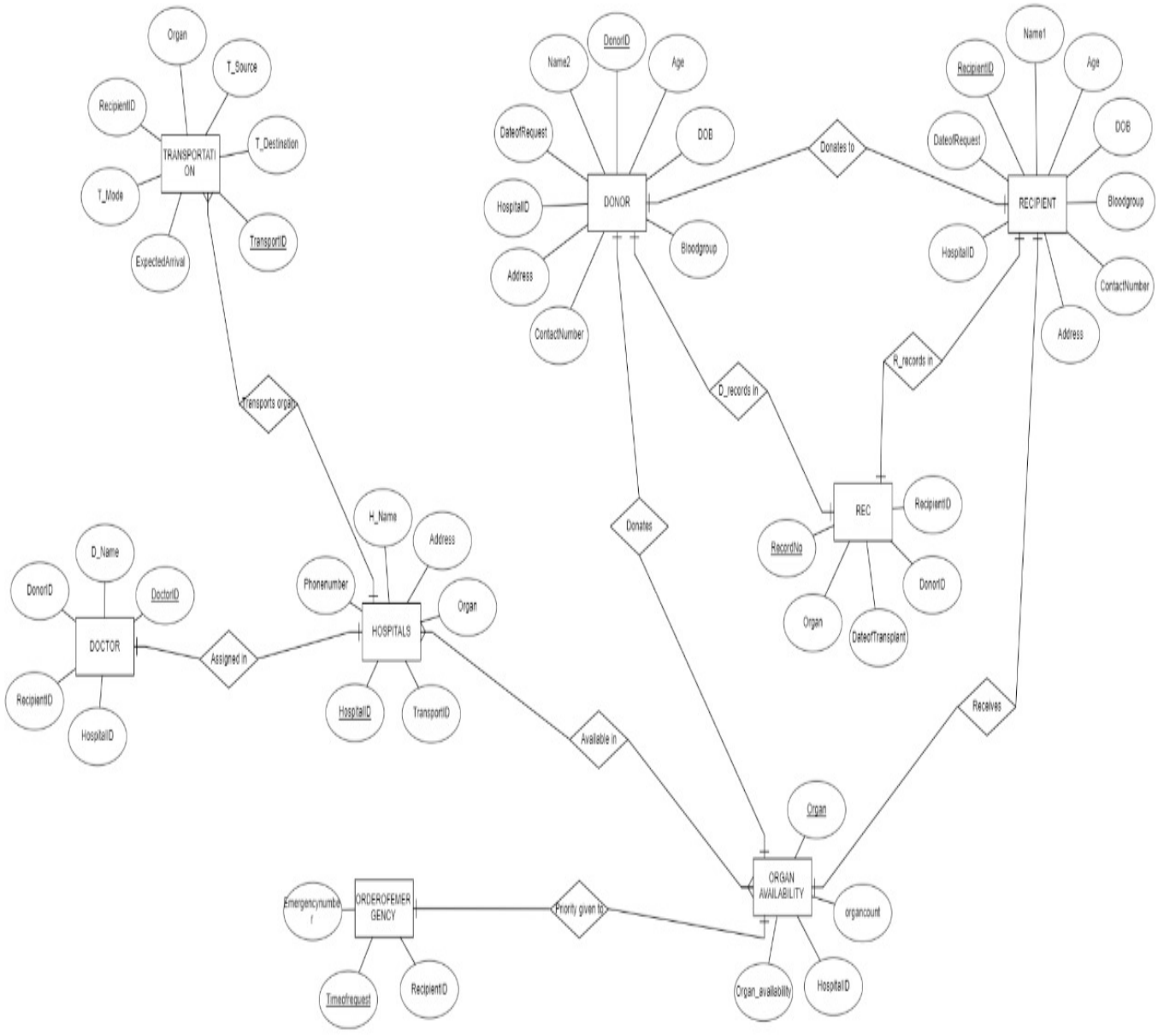
Problem Statement

To design a database management system to ease the tracking, organizing, and storing of information related to organ donation between patients and hospitals.

The given database makes the entire process of storing and managing information during organ donation easier.
The database can store various information regarding the donors, recipients, hospitals, doctors, organs available, records regarding the patients, and transportation. These aspects of the database make it easier to create a relationship among various schemas in the database.

Hence, we can make the information handling process during organ donation faster, easier, and more convenient through this database management system.

ER Diagram:


Organ
T_Source
RecipientID
T_Destination
TRANSPORTATION
T_Mode
TransportID
ExpectedArrival
Transports organ
DonorID
Name2
Age
DateofRequest
DOB
HospitalID
DONOR
Bloodgroup
Address
ContactNumber
Donates to
Name1
RecipientID
Age
DateofRequest
DOB
RECIPIENT
Bloodgroup
HospitalID
ContactNumber
Address
D_records in
R_records in
H_Name
Address
Phonenumber
Organ
HOSPITALS
HospitalID
TransportID
D_Name
DonorID
DoctorID
DOCTOR
RecipientID
HospitalID
Assigned in
Donates
REC
RecordNo
RecipientID
DonorID
Organ
DateofTransplant
Available in
Receives
Organ
ORGAN AVAILABILITY
organcount
Organ_availability
HospitalID
Priority given to
Emergencynumber
ORDEROFEMERGENCY
Timeofrequest
RecipientID

Normalization

---

1. Donor

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, DonorID.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

2. Recipient

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, RecipientID.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

3. Record

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, RecordNo.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

4. OrganAvailability

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, Organ.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

5. OrderOfEmergency

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, TimeofRequest.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

6. Hospitals

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key, HospitalID.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

7. Doctor

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key,DoctorID.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

8. Transportation

All attributes and their values are atomic and therefore in 1NF.
All the attributes are entirely dependent on the primary key,TransportID.
Therefore, the table is in 2NF. The relation does not have any transitive dependencies too and it is in 2NF which makes the relation in 3NF.

Table Descriptions

---

1. Donor

The donor table gives the basic information about the donors that have registered with the database.

```
create table Donor
(
DonorID int primary key,
Name2 varchar(20),
Age int,
DOB varchar(20),
Bloodgroup varchar(20),
ContactNumber int,
Address varchar(20),
HospitalID int,
organ varchar(20),
foreign key(organ) references organavailability(organ)
);
```

```
insert into donor values(237, 'Neeti', 40, '12-12-1981', 'A+', 3892847567, '24 Concrete Lane', 937, 'Pancreas');
insert into donor values(244, 'Dean', 50, '13-09-1971', 'B+', 3457892349, '47 Timber Blvd', 944, 'Liver');
insert into donor values(289, 'Lucas', 67, '14-08-1954', 'B-', 2345890459, '369 Rose Lane', 989, 'Kidney');
insert into donor values(227, 'Shakti', 56, '29-01-1966', 'AB+', 1235679023, '456 Duck Road', 927, 'Liver');
insert into donor values(259, 'Charles', 79, '29-09-1959', 'A+', 1239045896, '19 Primrose Lane', 959, 'Heart');
insert into donor values(245, 'Ramesh', 59, '23-03-1963', 'B+', 3849574810, '75 Tudor Lane', 945, 'Kidney');
insert into donor values(241, 'Claire', 74, '25-07-1947', 'A+', 1234567891, '358 Star Lane', 941, 'Lungs');
insert into donor values(284, 'Santosh', 53, '24-02-1963', 'A-', 246813579, '949 Gold Road', 984, 'Kidney');
```

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.013 seconds

	DONORID	NAME2	AGE	DOB	BLOODGROUP	CONTACTNUMBER	ADDRESS	HOSPITALID	ORGAN
1	237	Neeti	40	12-12-1981	A+	3892847567	24 Concrete Lane	937	Pancreas
2	244	Dean	50	13-09-1971	B+	3457892349	47 Timber Blvd	944	Liver
3	289	Lucas	67	14-08-1954	B-	2345890459	369 Rose Lane	989	Kidney
4	227	Shakti	56	29-01-1966	AB+	1235679023	456 Duck Road	927	Liver
5	259	Charles	79	29-09-1959	A+	1239045896	19 Primrose Lane	959	Heart
6	245	Ramesh	59	23-03-1963	B+	3849574810	75 Tudor Lane	945	Kidney
7	241	Claire	74	25-07-1947	A+	1234567891	358 Star Lane	941	Lungs
8	284	Santosh	53	24-02-1963	A-	246813579	949 Gold Road	984	Kidney

2. Recipient

The recipient table gives the basic information about the recipients that have put in a request for an organ.

```
create table Recipient
(
RecipientID int primary key,
Name1 varchar(20),
Age int,
DOB varchar(20),
Bloodgroup varchar(20),
ContactNumber int,
Address varchar(20),
HospitalID int,
DateofRequest varchar(20),
organ varchar(20),
foreign key(organ) references organavailability(organ)

);
```

```
insert into recipient  values(121, 'Franklin', 57, '14-03-1965', 'A+', 7392778738, '12 Baker Street', 921, '03-03-2022', 'Kidney');
insert into recipient  values(145, 'Rajesh', 44, '27-07-1977', 'O+', 8392746568, '45 Steel Street', 945, '02-04-2022', 'Heart');
insert into recipient  values(189, 'Roopa', 66, '23-08-1955', 'B-', 5673694829, '781 Flour Street', 989, '19-01-2022', 'Kidney');
insert into recipient  values(135, 'Harmony', 77, '27-09-1944', 'AB+', 3898465723, '98 Stone Lane', 935, '27-02-2002', 'Liver');
insert into recipient  values(184, 'Edmund', 49, '13-01-1972', 'A-', 4789375620, '345 Spring Blvd', 984, '14-02-2022', 'Lungs');
insert into recipient  values(139, 'Mohan', 81, '25-06-1940', 'B-', 8394729475, '237 Water Lane', 939, '31-03-2022', 'Pancreas');
insert into recipient  values(127, 'Susan', 58, '11-01-1964', 'A+', 7483938273, '79 Brookstone Blvd', 927, '01-01-2022', 'Heart');
insert into recipient  values(157, 'Sheela', 72, '31-10-1949', 'AB-', 6758493847, '69 Creek Road', 957, '21-03-2022', 'Kidney');
insert into recipient  values(144, 'Harry', 69, '17-09-1952', 'B+', 9283758493, '117 Frog Lane', 944, '13-01-2022', 'Liver')
insert into recipient  values(159, 'Katrina', 38, '09-01-1984', 'A+', 3489576846, '295 Lily Lane', 959, '26-04-2022', 'Pancreas');
```

Script Output × Query Result ×

SQL | All Rows Fetched: 10 in 0.006 seconds

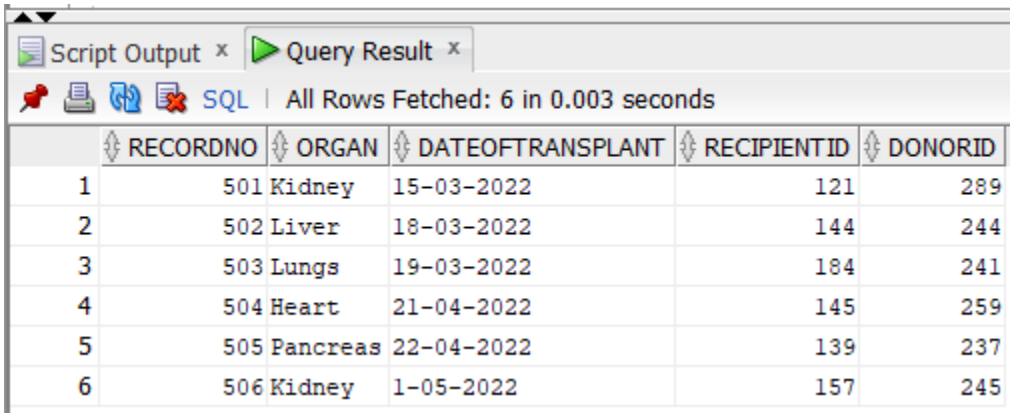
	RECIPIENTID	NAME1	AGE	DOB	BLOODGROUP	CONTACTNUMBER	ADDRESS	HOSPITALID	DATEOFREQUEST	ORGAN
1	121	Franklin	57	14-03-1965	A+	7392778738	12 Baker Street	921	03-03-2022	Kidney
2	145	Rajesh	44	27-07-1977	O+	8392746568	45 Steel Street	945	02-04-2022	Heart
3	189	Roopa	66	23-08-1955	B-	5673694829	781 Flour Street	989	19-01-2022	Kidney
4	135	Harmony	77	27-09-1944	AB+	3898465723	98 Stone Lane	935	27-02-2002	Liver
5	184	Edmund	49	13-01-1972	A-	4789375620	345 Spring Blvd	984	14-02-2022	Lungs
6	139	Mohan	81	25-06-1940	B-	8394729475	237 Water Lane	939	31-03-2022	Pancreas
7	127	Susan	58	11-01-1964	A+	7483938273	79 Brookstone Blvd	927	01-01-2022	Heart
8	157	Sheela	72	31-10-1949	AB-	6758493847	69 Creek Road	957	21-03-2022	Kidney
9	144	Harry	69	17-09-1952	B+	9283758493	117 Frog Lane	944	13-01-2022	Liver
10	159	Katrina	38	09-01-1984	A+	3489576846	295 Lily Lane	959	26-04-2022	Pancreas

3. Rec

The record table stores information about organ transplants that have already happened between a donor and a recipient.

```
create table Rec
(
RecordNo int primary key,
Organ varchar(20),
DateofTransplant varchar(20),
RecipientID int,
DonorID int,
foreign key(RecipientID) references Recipient(RecipientID),
foreign key(DonorID) references Donor(DonorID)
);
```

```
insert into rec values(501,'Kidney','15-03-2022',121,289);
insert into rec values(502,'Liver','18-03-2022',144,244);
insert into rec values(503,'Lungs','19-03-2022',184,241);
insert into rec values(504,'Heart','21-04-2022',145,259);
insert into rec values(505,'Pancreas','22-04-2022',139,237);
insert into rec values(506,'Kidney','1-05-2022',157,245);
```


Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.003 seconds

	RECORDNO	ORGAN	DATEOFTRANSPLANT	RECIPIENTID	DONORID
1	501	Kidney	15-03-2022	121	289
2	502	Liver	18-03-2022	144	244
3	503	Lungs	19-03-2022	184	241
4	504	Heart	21-04-2022	145	259
5	505	Pancreas	22-04-2022	139	237
6	506	Kidney	1-05-2022	157	245

4. OrganAvailability

The OrganAvailability table gives information about the type of organs and the number available.

```
create table Organavailability
(
  organ varchar(20) primary key,
  organcount int,
  Organ_avaliability varchar(20),

);
```

```
insert into Organavailability values('Kidney', 2, 'yes');
insert into Organavailability values('Heart', 1, 'yes');
insert into Organavailability values('Liver', 3, 'yes');
insert into Organavailability values('Lungs', 2, 'yes');
insert into Organavailability values('Pancreas',4, 'yes');
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.006 seconds

	ORGAN	ORGANCOUNT	ORGAN_AVALIABILITY
1	Kidney	2	yes
2	Heart	1	yes
3	Liver	3	yes
4	Lungs	2	yes
5	Pancreas	4	yes

5. OrderOfEmergency

The OrderOfEmergency table lists the recipients in order of the date of request. In case two recipients have the same date of request, the level of emergency is decided based on the time of request.

```
create table Orderofemergency
(
Timeofrequest varchar(20) primary key,
Emergencynumber int,
RecipientID int
);
```

```
insert into orderofemergency values('13:01:57', 1, 127);
insert into orderofemergency values('11:34:45', 2, 144);
insert into orderofemergency values('10:23:02', 3, 189);
insert into orderofemergency values('09:42:15', 4 , 184);
insert into orderofemergency values('14:09:56', 5, 135);
insert into orderofemergency values('12:10:59', 6, 121);
insert into orderofemergency values('09:56:41', 7, 157);
insert into orderofemergency values('13:49:13', 8, 139);
insert into orderofemergency values('21:31:04', 9, 145);
insert into orderofemergency values('23:03:45', 10, 159);
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.002 seconds

	TIMEOFREQUEST	EMERGENCYNUMBER	RECIPIENTID
1	13:01:57	1	127
2	11:34:45	2	144
3	10:23:02	3	189
4	09:42:15	4	184
5	14:09:56	5	135
6	12:10:59	6	121
7	09:56:41	7	157
8	13:49:13	8	139
9	21:31:04	9	145
10	23:03:45	10	159

6. Hospitals

The Hospitals table gives the basic information about the different hospitals registered with the database.

```
create table Hospitals
   (
    HospitalID int primary key,
    H_name varchar(20),
    Phonenumber int,
    Address varchar(20),
    Organ varchar(20),
    TransportID int,
    foreign key(TransportID) references Transportation(TransportID)
   );
```

```
insert into hospitals values(921, 'Hospital21', 2121212121, 'Road 21', 'Heart', 721);
insert into hospitals values(945, 'Hospital45', 4545454545, 'Road45', 'Kidney', 745);
insert into hospitals values(989, 'Hospital89', 8989898989, 'Road89', 'Lungs', 789);
insert into hospitals values(935, 'Hospital35', 3535353535, 'Road35', 'Heart', 735);
insert into hospitals values(984, 'Hospital84', 8484848484, 'Road84', 'Pancreas', 784);
insert into hospitals values(939, 'Hospital39', 3939393939, 'Road39', 'Lungs', 739);
insert into hospitals values(927, 'Hospital27', 2727272727, 'Road27', 'Liver', 727);
insert into hospitals values(957, 'Hospital57', 5757575757, 'Road57', 'Kidney', 757);
insert into hospitals values(944, 'Hospital44', 4444444444, 'Road44', 'Liver', 744);
insert into hospitals values(959, 'Hospital59', 5959595959, 'Road59', 'Lungs', 759);
insert into hospitals values(937, 'Hospital37', 3737373737, 'Road37', 'Heart', 737);
insert into hospitals values(941, 'Hospital41', 4141414141, 'Road41', 'Pancreas', 741);
```

Script Output x | Query Result x

SQL | All Rows Fetched: 12 in 0.003 seconds

	HOSPITALID	H_NAME	PHONENUMBER	ADDRESS	ORGAN	TRANSPORTID
1	945	Hospital45	4545454545	Road45	Kidney	745
2	927	Hospital27	2727272727	Road27	Liver	727
3	921	Hospital21	2121212121	Road 21	Heart	721
4	989	Hospital89	8989898989	Road89	Lungs	789
5	935	Hospital35	3535353535	Road35	Heart	735
6	984	Hospital84	8484848484	Road84	Pancreas	784
7	939	Hospital39	3939393939	Road39	Lungs	739
8	957	Hospital57	5757575757	Road57	Kidney	757
9	944	Hospital44	4444444444	Road44	Liver	744
10	959	Hospital59	5959595959	Road59	Lungs	759
11	937	Hospital37	3737373737	Road37	Heart	737
12	941	Hospital41	4141414141	Road41	Pancreas	741

7. Doctor

The doctor table gives information about the doctors working in each hospital.

```
create table Doctor
(
   DoctorID int primary key,
   D_name varchar(20),
   HospitalID int,
   RecipientID int,
   DonorID int,
   foreign key(HospitalID) references Hospitals(HospitalID)
 );
```

```
insert into doctor values(421, 'Doctor21', 921, 121, 289);
insert into doctor values(444, 'Doctor44', 944, 144, 244);
insert into doctor values(484, 'Doctor84', 984, 184, 241);
insert into doctor values(445, 'Doctor45', 945, 145, 259);
insert into doctor values(439, 'Doctor39', 939, 139, 237);
insert into doctor values(457, 'Doctor57', 957, 157, 245);
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.002 seconds

	DOCTORID	D_NAME	HOSPITALID	RECIPIENTID	DONORID
1	421	Doctor21	921	121	289
2	444	Doctor44	944	144	244
3	484	Doctor84	984	184	241
4	445	Doctor45	945	145	259
5	439	Doctor39	939	139	237
6	457	Doctor57	957	157	245

8. Transportation

The Transportation table is used to track the transportation of an organ from a particular source to a particular destination. Only organs currently in active transit are tracked, while the rest of the table has null values. When another organ has to be transported, values are given to the tuple with the required source and destination.

```
create table Transportation
 (
   TransportID int primary key,
   T_Source varchar(20),
   T_Destination varchar(20),
   Organ varchar(20),
   RecipientID int,
   T_Mode varchar(20),
   ExpectedArrival varchar(20)
 );
```

```
insert into transportation values(745, 'Hospital45', 'Hospital89', 'Kidney',189,'By road','13-05-2022');
insert into transportation values(727,'Hospital27','Hospital35','Liver',135,'Flight','14-05-2022');
insert into transportation values(721,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(789,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(735,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(784,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(739,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(757,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(744,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(759,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(737,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
insert into transportation values(741,'NULL', 'NULL', 'NULL',0,'NULL','NULL');
```

Script Output × Query Result ×

SQL | All Rows Fetched: 12 in 0.005 seconds

	TRANSPORTID	T_SOURCE	T_DESTINATION	ORGAN	RECIPIENTID	T_MODE	EXPECTEDARRIVAL
1	745	Hospital45	Hospital89	Kidney	189	By road	13-05-2022
2	727	Hospital27	Hospital35	Liver	135	Flight	14-05-2022
3	721	NULL	NULL	NULL	0	NULL	NULL
4	789	NULL	NULL	NULL	0	NULL	NULL
5	735	NULL	NULL	NULL	0	NULL	NULL
6	784	NULL	NULL	NULL	0	NULL	NULL
7	739	NULL	NULL	NULL	0	NULL	NULL
8	757	NULL	NULL	NULL	0	NULL	NULL
9	744	NULL	NULL	NULL	0	NULL	NULL
10	759	NULL	NULL	NULL	0	NULL	NULL
11	737	NULL	NULL	NULL	0	NULL	NULL
12	741	NULL	NULL	NULL	0	NULL	NULL

9. Availablein

This relation shows the hospital ID and the organ that the corresponding hospital has.

```
create table Availablein
(
   HospitalID int,
   Organ varchar(20),
   foreign key (HospitalID) references Hospitals(HospitalID),
   foreign key (Organ) references Organavailability(Organ)

  );
```

```
insert into availablein values(921, 'Heart');
insert into availablein values(945, 'Kidney');
insert into availablein values(989, 'Lungs');
insert into availablein values(935, 'Heart');
insert into availablein values(984, 'Pancreas');
insert into availablein values(939, 'Lungs');
insert into availablein values(927, 'Liver');
insert into availablein values(957, 'Kidney');
insert into availablein values(944, 'Liver');
insert into availablein values(959, 'Lungs');
insert into availablein values(937, 'Heart');
insert into availablein values(941, 'Pancreas');
```

Script Output | Query Result

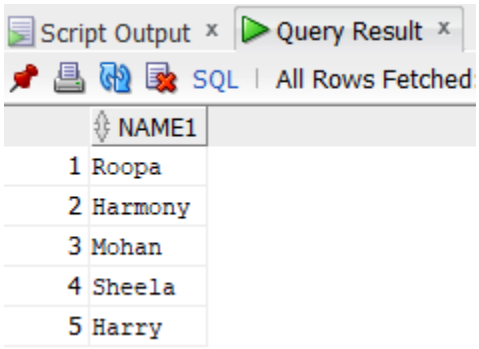
SQL | All Rows Fetched: 12 in 0.004 seconds

	HOSPITALID	ORGAN
1	921	Heart
2	945	Kidney
3	989	Lungs
4	935	Heart
5	984	Pancreas
6	939	Lungs
7	927	Liver
8	957	Kidney
9	944	Liver
10	959	Lungs
11	937	Heart
12	941	Pancreas

Queries

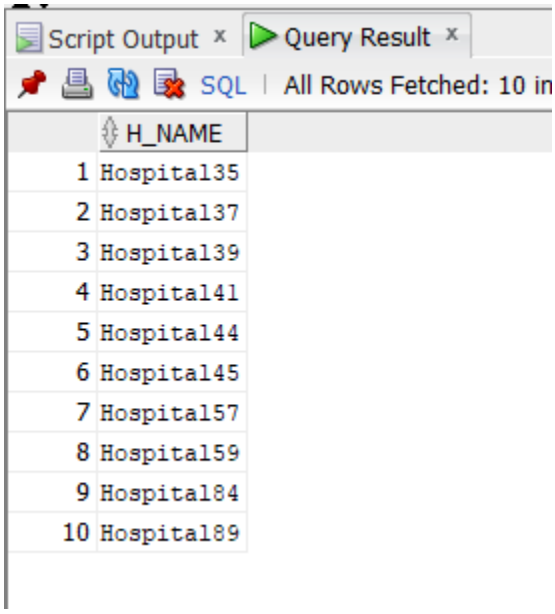
---

1. Find the patients who are older than 60.
SELECT NAME FROM RECIPIENT WHERE AGE>60;



	NAME1
1	Roopa
2	Harmony
3	Mohan
4	Sheela
5	Harry

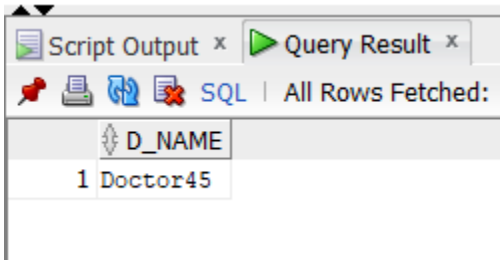
2. Find the names of the hospital whose IDs are greater than 930.
SELECT H_NAME FROM HOSPITALS WHERE HOSPITALID>930;



	H_NAME
1	Hospital35
2	Hospital37
3	Hospital39
4	Hospital41
5	Hospital44
6	Hospital45
7	Hospital57
8	Hospital59
9	Hospital84
10	Hospital89

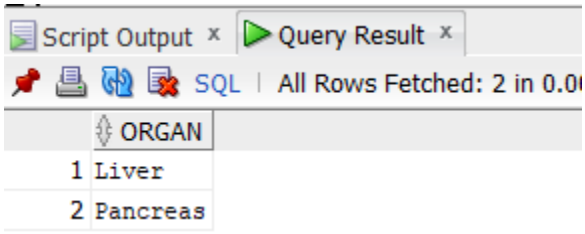
3. Find the name of the doctor whose patient recipient ID was 145.

SELECT D_NAME FROM DOCTOR WHERE RECIPIENTID=145;



4. Find the names of the organs whose count is greater than 2.

SELECT ORGAN FROM ORGANAVAILABILITY WHERE ORGANCOUNT>2;



5. Print all the organs that are available in decreasing order of their count.

SELECT ORGAN FROM ORGANAVAILABILITY ORDER BY ORGANCOUNT DESC;