# Charity Data Base project


**Team Members:**

**1- محمود محمد طه أبوعياد**
**2- حسام الدين أنسى**
**3- طارق فهيم**
**4- محمد على يوسف الخياط**


*2016*

# Abstraction

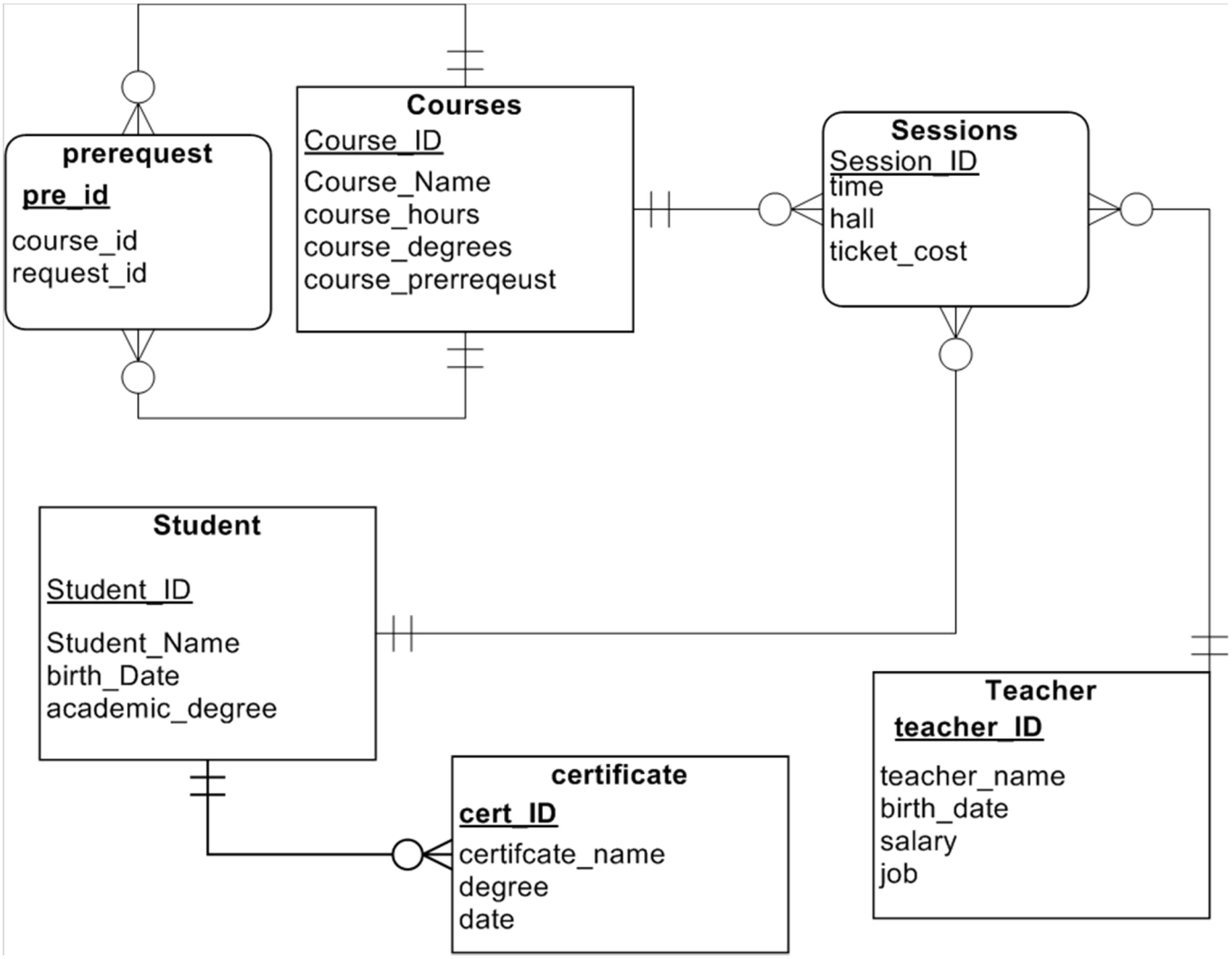
**Our project aims to store and restore all data needed for the charity system and safely update and delete from data**

# Project components

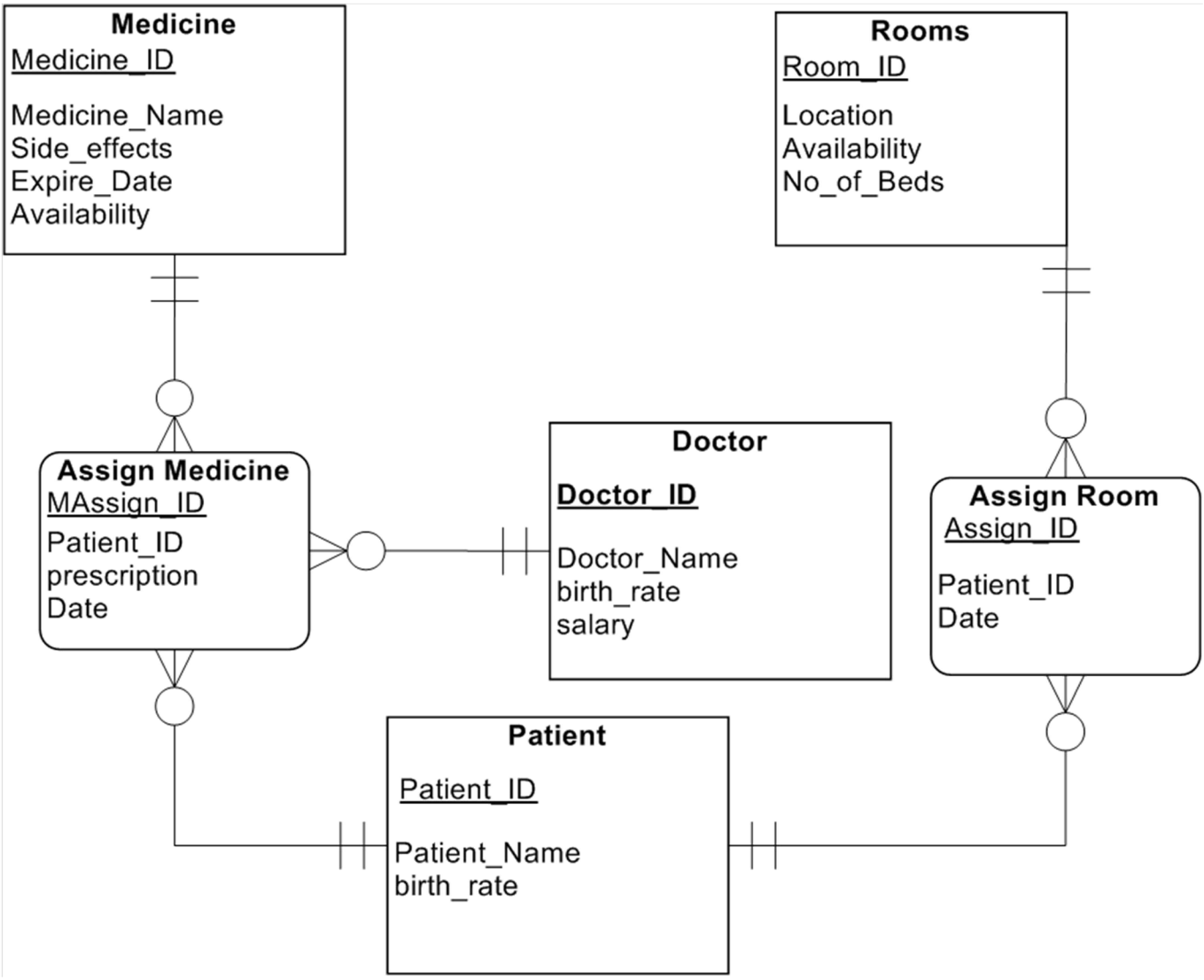
1- **ER diagrams**
2- **Relational diagrams**
3- **Documentation for each table**
4- **SQL scripts of the implementation of the ER diagrams**
5- **GUI to connect to the database with the user**

# 1- ER Diagrams

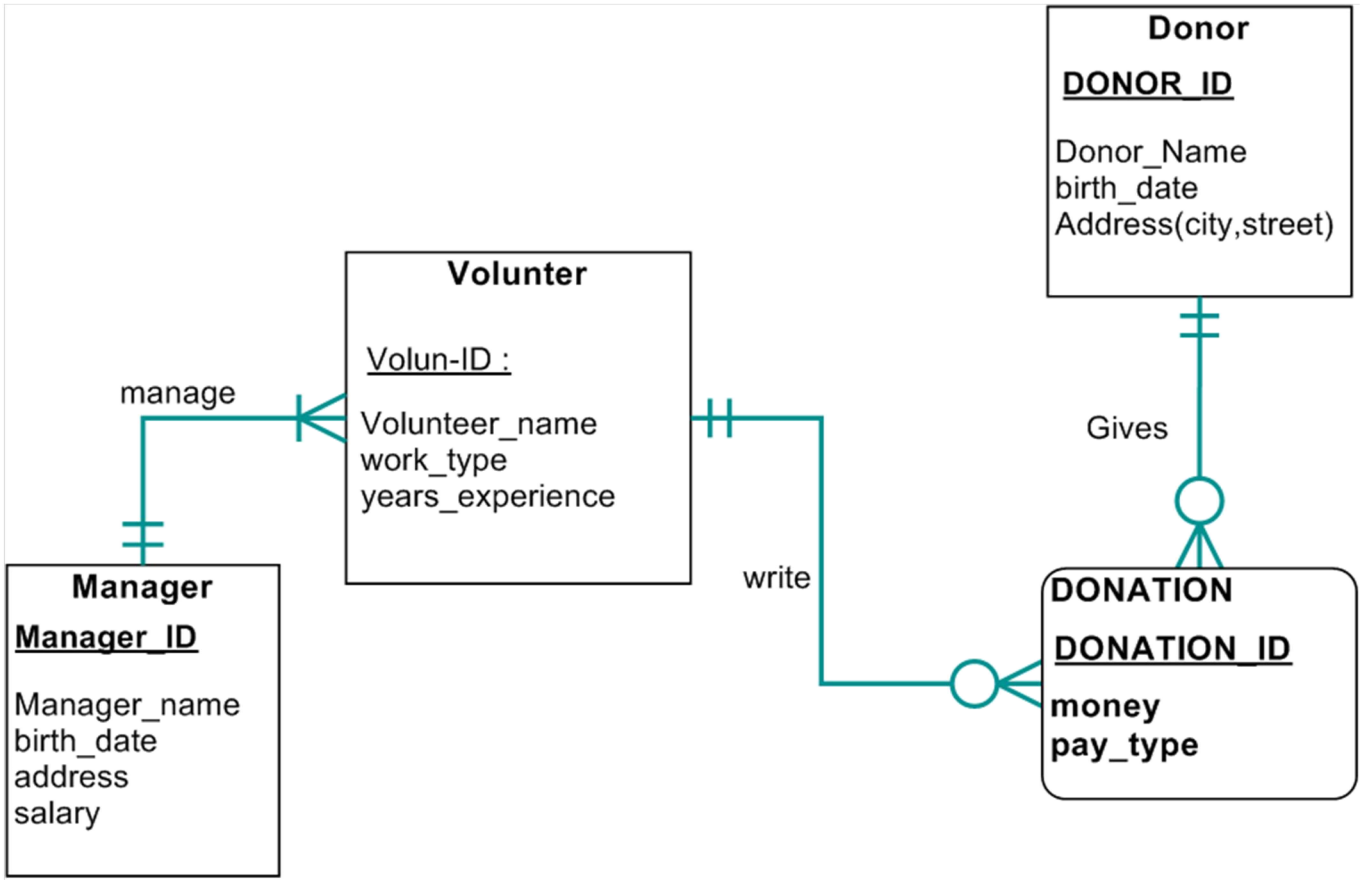
**a- For the educational sub-system**



**b- For the medical sub-system**

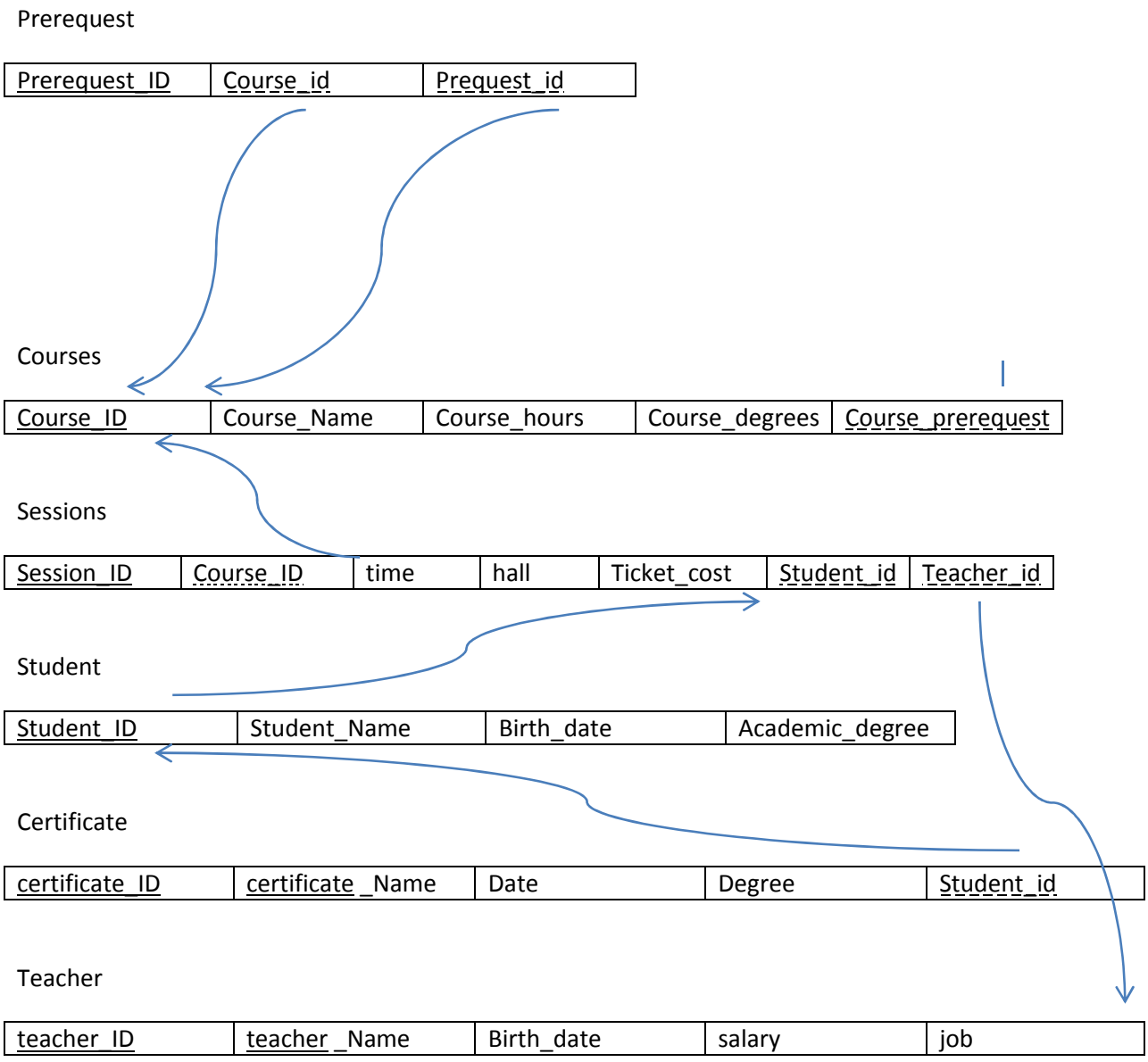


**c- For the Learning sub-system**

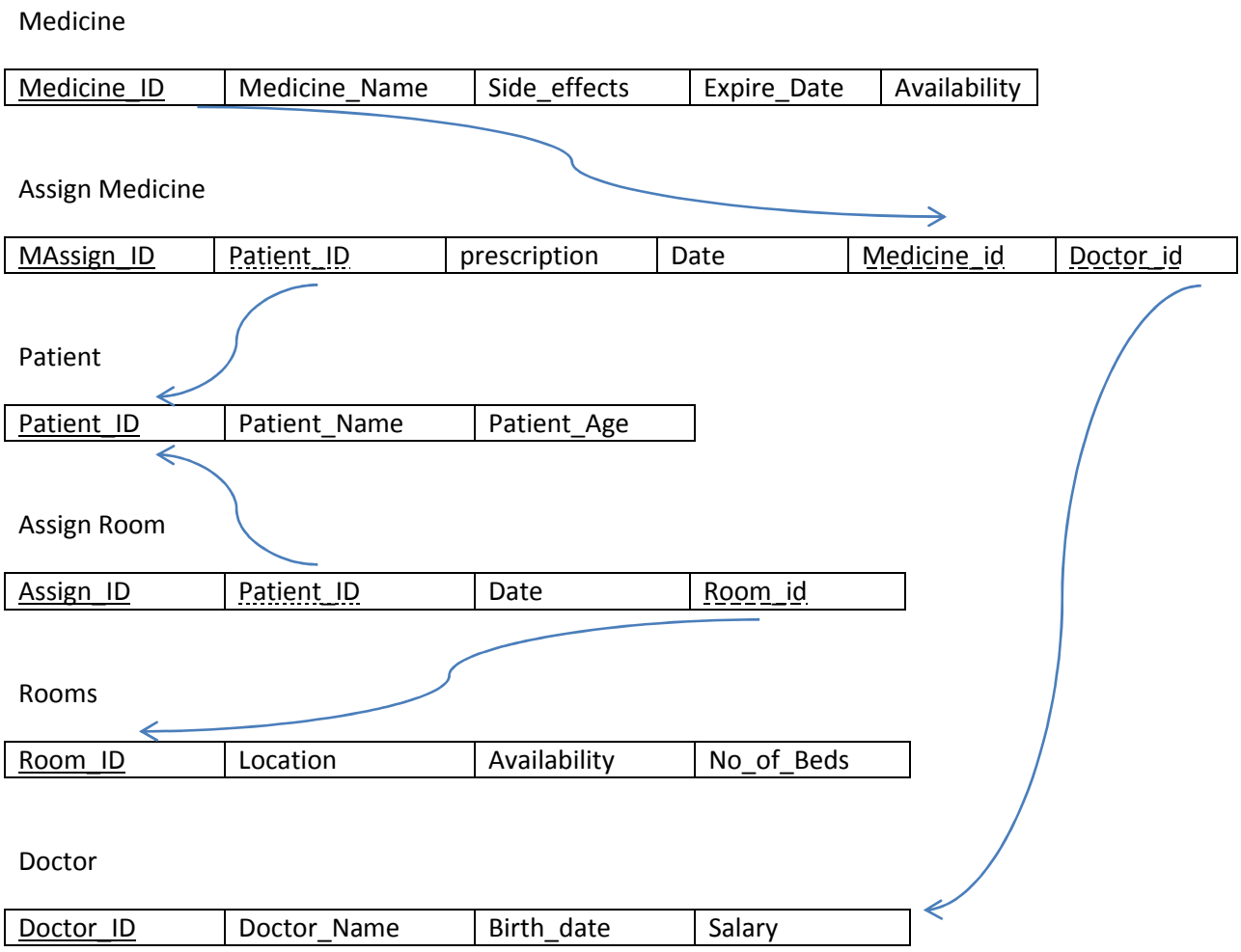


# 2- Relational diagrams

## Learning sub-system



# Medical sub-system



# Fund raising sub-system

## Manager

| Manager_ID | Manager_Name | Birth_date | Address | Salary |
|---|---|---|---|---|

## Volunteer

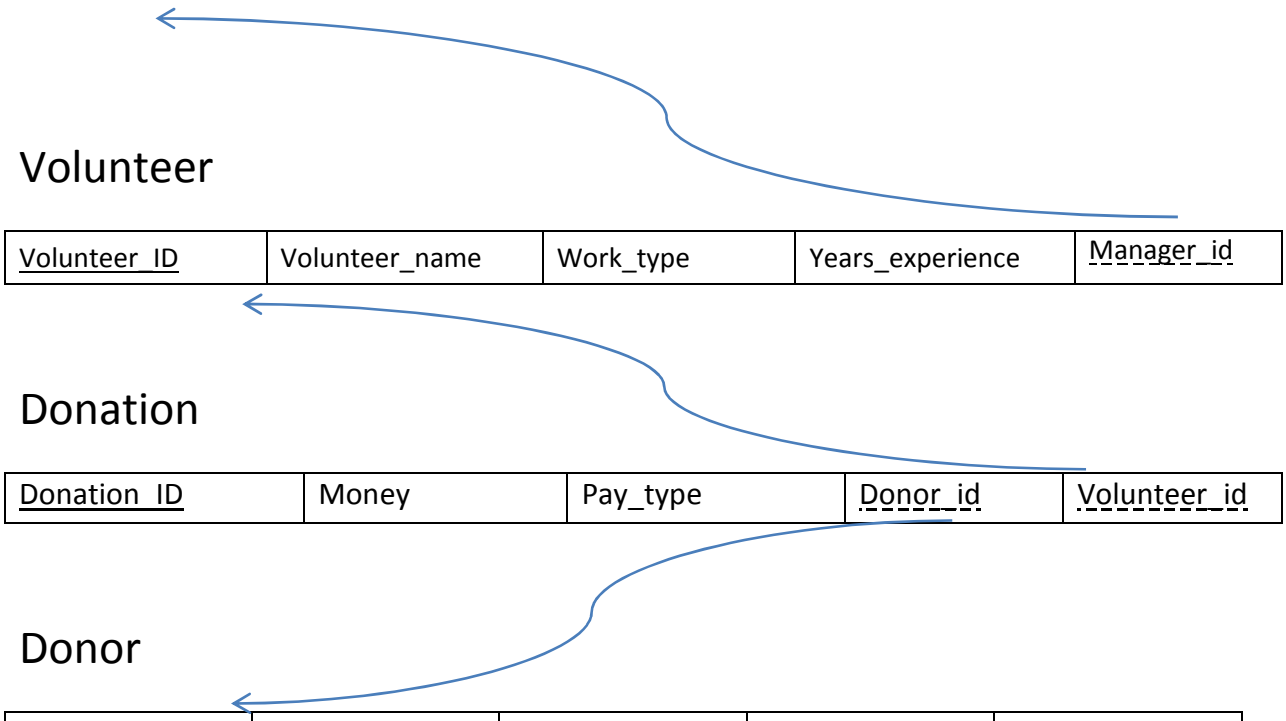
| Volunteer_ID | Volunteer_name | Work_type | Years_experience | Manager_id |
|---|---|---|---|---|

## Donation

| Donation_ID | Money | Pay_type | Donor_id | Volunteer_id |
|---|---|---|---|---|

## Donor

| Donor_ID | Donor_name | Address | Brith_date | Address |
|---|---|---|---|---|



# 3- Documentation of the system

# a- Learning sub-system

Courses

Here we get the course required for the case the system store all its data and courses

| Course | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Course_ID | Int (11) | Yes | | No | Identifier Key |
| Coures_Name | Varchar2(30) | | | No | Name of this course |
| Course_degrees | Varchar2(30) | | | No | Degrees of the course |

Student

Here we get the students required for the case the system store all its data and students

| student | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| student_ID | Int (11) | Yes | | No | Identifier Key |
| studnet_Name | Varchar2(30) | | | No | Get the name |
| academic_degrees | Varchar2(30) | | | No | Degrees of the course |
| Birth_date | Date | | | No | Identified the birth date |

Teacher

Here we get the teachers required for the case the system store all its data and teachers

| teacher | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| teacher_ID | Int (11) | Yes | | No | Identifier Key |
| teacher_Name | Varchar2(30) | | | No | Get the name |
| Salary | Int(11) | | | No | Identifies the salary |
| Birth_date | Date | | | No | Identified the birth date |

Certificate

Here we get the Certificate s required for the case the system store all its data and Certificates

| Certificate | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Certificate _ID | Int (11) | Yes | | No | Identifier Key |
| Certificate _Name | Varchar2(30) | | | No | Get the name |
| academic_degrees | Varchar2(30) | | | No | Degrees of the course |
| Date_get | Date | | | No | Identified the date |
| Student_id | Int(11) | | Yes | No | Students owns the certificate |

sessions

Here we get the sessions required for the case the system store all its data and sessions

| session | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| session _ID | Int (11) | Yes | | No | Identifier Key |
| Hall | Varchar2(30) | | | No | Get the hall |
| Ticket_cost | int(11) | | | No | Cost of session |
| Time | Date | | | No | Identified the date |
| Student_id | Int(11) | | Yes | No | Students owns the certificate |
| Teacher_id | Int(11) | | Yes | No | Teacher teaches the session |
| Course_id | Int(11) | | Yes | No | Course taught in the session |

# b- Medical sub-system

Medicine

Here we get the Medicine required for the case the doctor is up to and check whether it has side effects or not , depending on the case we assign the medicine if available and this medicine goes to the assigned medicine table for that particular patient

| Medicine | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Medicine_ID | Int (11) | Yes | | No | Identifier Key |
| Medicine_Name | Varchar2(30) | | | No | Name of this Medicine |
| Side_Effects | Varchar2(30) | | | No | Side effects of this Medicine |
| Expire_Date | Int(8) | | | No | Expiration date of this Medicine |
| Availability | Text | | | No | Whether it's available or not |

Assigned Medicine

here we receive this assigned medicine by a prescription from the doctor and add it to that particular patient with the unique id he/she gets and this assigned medicine id is written in that patients medical history for further recommendations

| Assigned Medicine | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| MAssign_ID | Int (11) | Yes | | No | Identifier Key |
| Patient_ID | Int (11) | | Yes | No | Foreign Key for Patient Table |
| Prescription | Varchar2(30) | | | No | Prescription of this Medicine to be included in the history of the patient |
| Date | Int(8) | | | No | Date of Assigning this medicine |
| Medicine_id | Int(11) | | Yes | No | Identified the medicine to be assigned |
| Doctor_id | Int(11) | | Yes | No | Identifies the doctor gave you the medicine |

Patient

This is one of the most important tables as without it there would be no medical services given to patients as there will be no patients , we reserve the patient's name ,age for identification and assign each patient and id for further services and to ease the process as for the medical history is really important to recommend the best way to treat this patient whether it's by a surgery , medicine … etc

| Patient |
|---|

| Field | Type | Primary Key | Secondary Key | Null | Comments |
|---|---|---|---|---|---|
| Patient_ID | Int (11) | Yes | | No | Identifier Key |
| Patient_Name | text | | | No | Patient Name |
| Birth_date | date | | | No | Patient Age |

Assign Room

Here we check the assigned rooms for the patient with 2 keys one from the room and the other from the patient to make sure this room is taken or available and for history of this room as occupied

| Assign Room | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Assign_ID | Int (11) | Yes | | No | Identifier Key |
| Patient_ID | Int (11) | | Yes | No | Patient Foreign ID |
| Date | Int (8) | | | No | Data of Room Assignment |
| Room_id | Int(11) | | yes | no | Identifies the room to be assign |

Rooms

In this table we check if there are any available rooms , how many beds in a room and the location of each room in the premises

| Rooms | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Room_ID | Int (11) | Yes | | No | Identifier Key |
| Location | Text | | | No | Location of the room in the Hospital or caring center |
| Availability | Varchar2(30) | | | No | To Check if there are any available rooms |
| No_of_Beds | Int(5) | | | No | To determine how many beds in a room for variations |

Doctors

in this table we get the required for the identification of each doctor for the medical history of each patient along with the financial matter of the financial department and the services given within a period of time

| Doctors | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Doctor_ID | Int (11) | Yes | | No | Identifier Key |
| Doctor_Name | text | | | No | Name of Doctor for Medical History |
| Birth_date | Int(8) | | | No | To Check for validation of employment |
| Salary | Int(5) | | | No | For Financial Purposes |

# c- Fund raising sub-system

Donor

Here we get the Donor s required for the case the system store all its data and Donors

| teacher | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Donor r_ID | Int (11) | Yes | | No | Identifier Key |
| Donor_Name | Varchar2(30) | | | No | Get the name |
| address | Varchar2(30) | | | No | Here we get the address |
| Birth_date | Date | | | No | Identified the birth date |

volunteer

Here we get the volunteers required for the case the system store all its data and volunteers

| teacher | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| volunteers _ID | Int (11) | Yes | | No | Identifier Key |
| volunteers _Name | Varchar2(30) | | | No | Get the name |
| Work_type | Varchar2(30) | | | No | Type of work done |
| Years_experience | Int(11) | | | No | Year of experience in work |
| Manager_id | Int(11) | | Yes | No | Specifies the manger |

Manager

Here we get the Managers required for the case the system store all its data and Managers

| teacher | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Manager_ID | Int (11) | Yes | | No | Identifier Key |
| Manager_Name | Varchar2(30) | | | No | Get the name |
| address | Varchar2(30) | | | No | Here we get the address |
| Birth_date | Date | | | No | Identified the birth date |

Donation

Here we get the Donations required for the case the system store all its data and Donations

| teacher | | | | | |
|---|---|---|---|---|---|
| Field | Type | Primary Key | Secondary Key | Null | Comments |
| Donation_ID | Int (11) | Yes | | No | Identifier Key |
| Pay_type | Varchar2(30) | | | No | How to be paid |
| Money | Int(11) | | | No | Amount of money |
| Volunteer_id | Int(11) | | Yes | No | Specifies the volunteer |
| donor_id | Int(11) | | Yes | No | Specifies the donor |

# 4- SQL codes

# 1- Learning sub-system

```
create table student (
student_id number(11) primary key,
student_name varchar2(30) ,
birth_date date not null ,
academic_degree number(11) );



create table teacher (
teacher_id number(11) primary key,
teacher_name varchar2(30) ,
birth_date date not null ,
job varchar2(30) );



create table certificate (
certificate_id number(11) primary key,
certificate_name varchar2(30) ,
date_get date not null,
```

```
degree varchar2(30) ,
student_id number(11) ,
FOREIGN KEY (student_id) references student(student_id)
);

create table course (
course_id number(11) primary key,
course_name varchar2(30) ,
course_degree number(11) ,
course_hours number(11) );

create table prerequest (
prerequest_id number(11) primary key,
course_id number(11) ,
prequest_id number(11) ,
FOREIGN KEY (course_id) references course(course_id) ,
FOREIGN KEY (prequest_id) references course(course_id)
 );

create table sessions (
certificate_id number(11) primary key,
cost number(11) ,
```

```
date_get date not null,
hall varchar2(30) ,
course_id number(11) ,
student_id number(11) ,
teacher_id number(11) ,

FOREIGN KEY (student_id) references student(student_id) ,
FOREIGN KEY (course_id) references course(course_id) ,
FOREIGN KEY (teacher_id) references teacher(teacher_id)
);
```

# 2- Medical sub-system

```
create table medicine (
 medicine_id number(11) not null  PRIMARY KEY ,  medicine_name varchar2(30) not null ,
 side_effects varchar2(30) ,
 expire_date date not null,
 availability number(11) not null);
```

```
create table room(
room_id number(11) not null PRIMARY KEY,
```

```
location varchar2(30) not null,
availability number(11) not null, no_of_bed number(11) not null);


create table doctors(
doctor_id number(11) not null  PRIMARY KEY, doctor_name varchar2 (30) not null,
birth_date date not null,
salary number(11) not null);


create table patients(
patient_id number(11) not null PRIMARY KEY, patient_name varchar2(30) not null,
birth_date date not null);


create table assign_room(
assign_id number(11)  PRIMARY KEY,
patient_id number(11),
room_id number(11),
assign_date date ,
FOREIGN KEY (patient_id) REFERENCES patients(patient_id) ,
FOREIGN KEY (room_id) REFERENCES room(room_id)
);
```

```
create table assign_medicine(
assign_id number(11)  PRIMARY KEY,
patient_id number(11),
medicine_id number(11),
doctor_id number(11),
assign_date date ,
prescription varchar2(30),

FOREIGN KEY (patient_id) REFERENCES patients(patient_id) ,

FOREIGN KEY (medicine_id) REFERENCES medicine(medicine_id) ,

FOREIGN KEY (doctor_id) REFERENCES doctors(doctor_id)

);
```

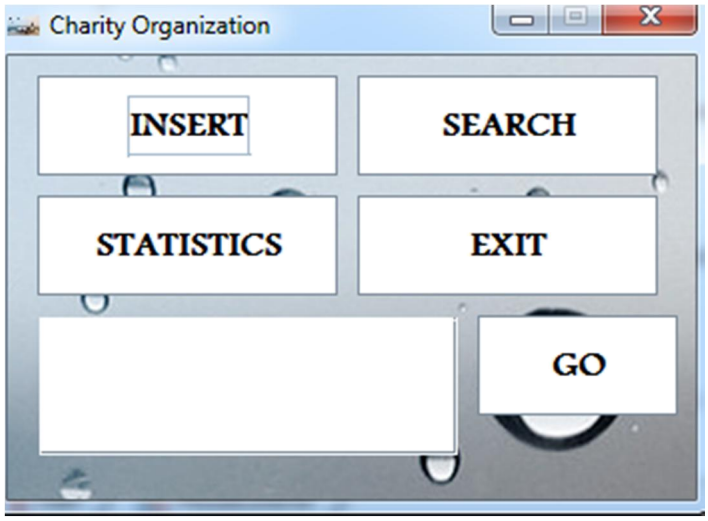
# 3- fund raisig sub-system

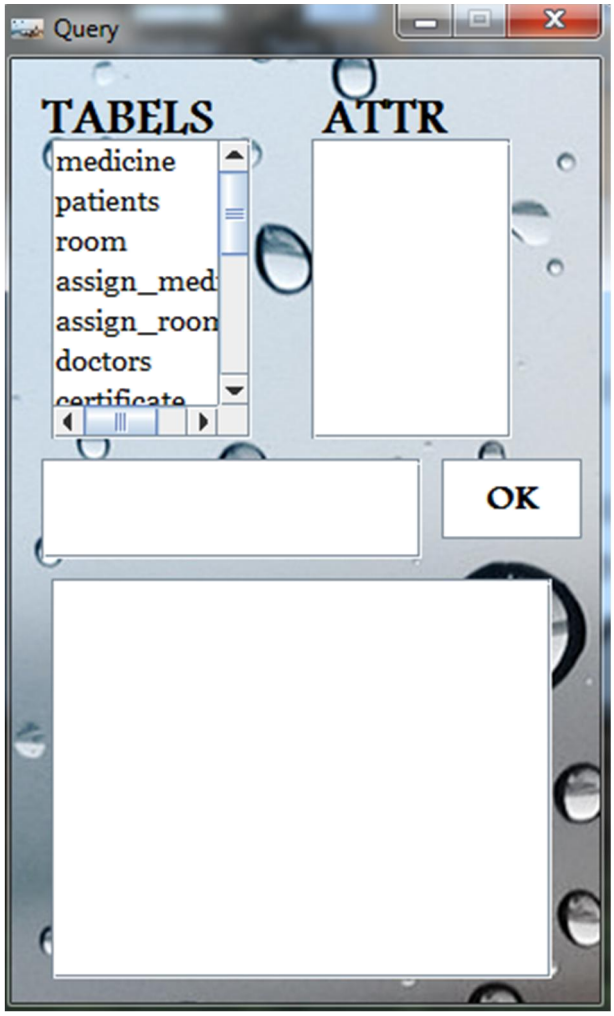
```
create table Donor (
Donor_ID number(11) primary key,
```

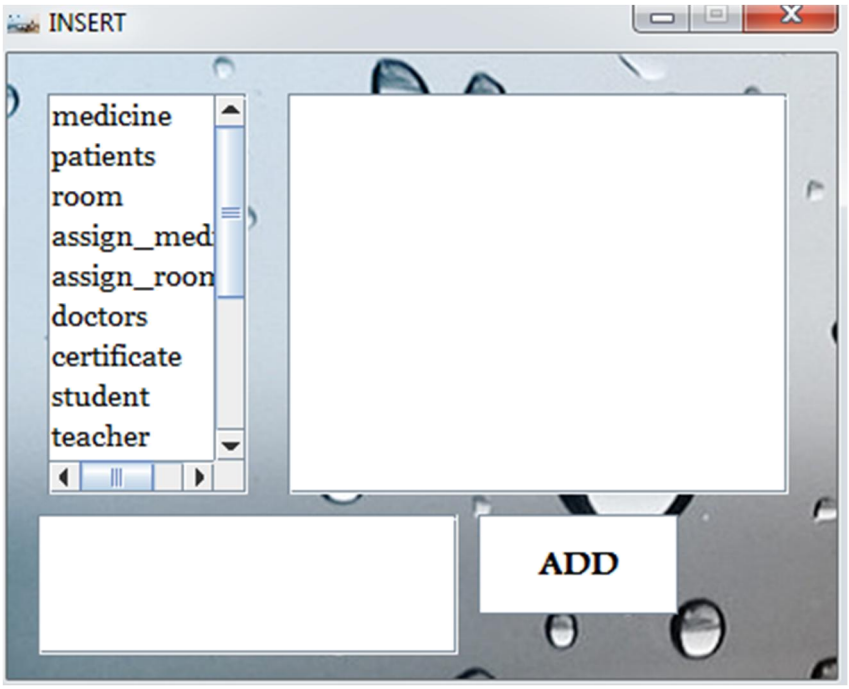
```
address varchar2(30) not null ,
birth_date date
);

create table Manager (
Manager_ID number(11) primary key,
Manager_name varchar2(30) not null,
address varchar2(30) ,
Salary number(11)

);

create table volunteer (
volunteer_ID number(11) primary key,
volunteer_name varchar2(30) not null,
year_experience varchar2(30) ,
manager_id number(11),
FOREIGN KEY (manager_id) REFERENCES manager(manager_id)
);
```

```
create table Donation (
donation_ID number(11) primary key,
money number(11) not null ,
pay_type varchar2(30) ,
Donor_id number(11) ,
volunteer_id number(11) ,
FOREIGN KEY (Donor_id) REFERENCES Donor(Donor_id),
FOREIGN KEY (volunteer_id) REFERENCES  Volunteer(volunteer_id)

);
```

# Screen shots of the gui with java


Charity Organization
INSERT
SEARCH
STATISTICS
EXIT
GO


Query
TABELS
ATTR
medicine
patients
room
assign_med
assign_roon
doctors
certificate
OK


INSERT
medicine
patients
room
assign_med
assign_roon
doctors
certificate
student
teacher
ADD