**“DataBASE PROJECT REPORT”**

**TOpic:**

**Hospital management system**

**GROUP MEMBERS:**

**NIMRAH TARIQ SE-003**

**ARooba khan se-011**

**HOSPITAL MANAGEMENT SYSTEM**

The project we have made is a hospital management system. It is a system to manage the records of employees and patients working in a hospital. Through this system the manual work is made much easier and the data is recorded much more efficiently.

**PROBLEMS OF STATEMENT:**

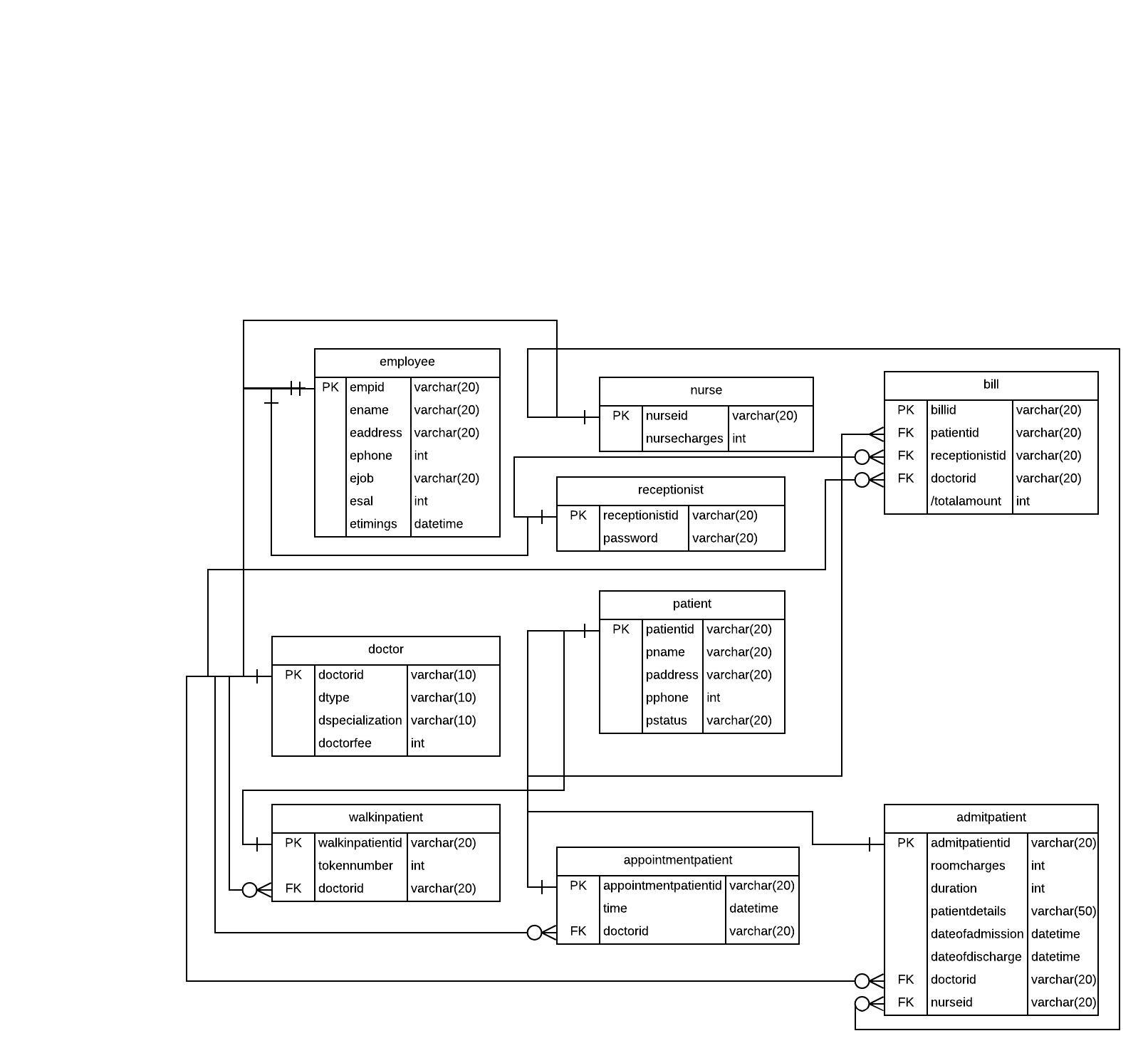
Following are some of the problems that we saw in manual management system of a hospital that inspired the making of this management system.

1. Managing a hospital is a huge task. Patients are continuously coming and going, it isn’t an easy task to manage multiple incoming patients and assign them some token number or id.
2. It would be great if the doctor can get a clear status of what kind of patient he is dealing with. It should be clearly stated that whether if he is a walk-in patient, a patient who called to book an appointment or if it a patient who needs to be admitted in the hospital.
3. A full bio-data of the doctors in the hospital should be there as well so patients can have a look at the doctor itinerary. The working time of the specialists in the hospital along with their specialties should be stored in the records unambiguously.
4. The information about the receptionist and the nurses working in the hospital should be stored as well.
5. The bill should be generated according to the type of patient.

**TOOLS:**

We have created our EERD by using an online tool ***“LUCID CHARTS”***. And this EERD can be implemented in any database tool. We have used Oracle to implement it.

**ENHANCED RELATIONSHIP DIAGRAM:**

****

**ASSUMPTIONS:**

1. We have considered only three kinds of patients, “***Walk-in Patients***”, “***Admit Patients***” and “***Appointment Patients***”.
2. We first created a superclass, an entity known as “***Patient”***, and then the 3 kinds of patients were inherited from the superclass.
3. Then we made a superclass, an entity known as “***Employee”***, and further 3 entities were inherited from it.
4. The entities inherited from the Employee class ***are “Doctor”, “Nurse***” and “***Receptionist”***.
5. We’ve assumed that there will be multiple receptionists in the hospital who’d work in shifts.
6. The relationship between the two super classes that is Employee and Patient is “***many-to-many***”.
7. The participation of Employee is mandatory (1) because the doctor, nurse and receptionist will always be there.
8. The participation of Patient is optional (0) because it might happen sometimes that there are no patients as all.
9. There is a non-disjoint constraint (AND) in the Patient inheritance. A walk-in patient can also be and admit patient.
10. There is a non-disjoint constraint (OR) in the Employee inheritance. A doctor can neither be a nurse nor a receptionist, nor vice versa.
11. There is “***is-a”*** relationship between the subclasses and super classes like “Doctor/Nurse/Receptionist is an employee” and “Walk-in/Admit/Appointment is a patient”.
12. In ***“bill”*** table the derived attribute ***“total amount”*** holds for the doctor fee and admit patient related charges if any.
13. The relationship between the ***“doctor”*** and the subclasses of ***“patient”*** is a one to many.
14. ***“Nurse”*** has a one to many relationship with ***“admit”*** patients.
15. ***“Bill”*** has relationship of one-to many with ***“patient”, ”doctor”*** and ***“receptionist”*** because all of those entities plays a major role in making a bill unique for each specific patient

**NORMALIZATION:**

We are going to normalize our Employee table first:

**2st Normal Form:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| empid | ename | eaddress | ephone | ejob | esal | etimings |
| REC-01 | Ahmed | Street abc, Karachi | 033983774 | Receptionist | 15,000/- | 9:00:00 |
| NUR-2 | Kate | Sector 16, Islamabad | 032382748 | Nurse | 20,000/- | 12:00:00 |
| DOC-3 | Kamran | Street xyz, Lahore | 039734830 | Doctor | 70,000/- | 14:00:00 |

The above table is in the 2st normal form and the primary key is empid. The functional dependencies is:

1. fd1: empid 🡪 ejob

This above mentioned dependency is transitive dependency because ejob is dependent on empid via ename. So we can make another table for it and convert it into 3rd Normal Form as the transitive dependency would be removed.

Job(empid, ejob)

Employee(empid, ename, eaddress, ephone, esal)

We can observe that with the formation of a new table the ejob attribute is deleted from the Employee table and has been added into the new table that is the “Job” table. This reduced the redundancy from the Employee table and made it much easier to understand.

**3rd Normal Form:**

Now with the addition of another table now our employee table is in 3rd Normal form and now we will check if there is a need to create the BCNF or not. By examining the table, it is concluded that the table can’t be further normalized so 3rd Normal Form is the final normalized form, the employee table could have.

Now we will normalize our Patient table:

**2nd Normal Form:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| patientID | pName | pAddress | pPhone | pStatus |
| WIP-1 | Ali | Street abc, Karachi | 103020048 | Walkin |
| APP-6 | Saleem | Sector 12, Lahore | 433467789 | Appointment |
| AP-76 | Tahira | Block 12, Karachi | 980230830 | Admit |

The above table is in the 2st normal form and the primary key is patientID. The functional dependency is:

1. patientID 🡪 pstatus

This above mentioned dependency is transitive dependency because pStatus is dependent on patientID via pName. So we can make another table for it and convert it into 3rd Normal Form as the transitive dependency would be removed.

Status(patientID, pStatus)

Patient(patientID, pName, pAddress, pPhone)

We can observe that with the formation of a new table the pStatus attribute is deleted from the Patient table and has been added into the new table that is the “Status” table. This reduced the redundancy from the Patient table and made it much easier to understand.

**3rd Normal Form:**

Now with the addition of another table now our patient table is in 3rd Normal form and now we will check if there is a need to create the BCNF or not. By examining the table, it is concluded that the table can’t be further normalized so 3rd Normal Form is the final normalized form, the patient table could have.

**FUNCTIONALITY:**

Our system will provide the following functionalities:

1. It will assign an ID to the patients as per their status as to keep their record.
2. It will be feasible for the patients to book an appointment over the phone.
3. The receptionist would schedule appointment, assign token number for the patients by considering the timings of the doctor.
4. The receptionist would generate the bill.
5. It will keep the records of all the employees and patients.
6. Receptionist will enroll the admit patients in to rooms and assign them doctors and nurses.
7. It will give a token number to the walk-in patients and assign a specific time for the appointment patients.
8. The nurses will be assigned the admit patients to whom they are responsible to look after.

**DATABASE CODE:**

CREATE TABLE employee

(

empid varchar2(20) not null,

ename varchar2(20) not null,

eaddress varchar2(100) null,

ephone number(11) not null,

ejob varchar2(20) not null,

esal number(20) not null,

etimings timestamp not null,

CONSTRAINT employee\_pk PRIMARY KEY (empid)

);

CREATE TABLE doctor

(

doctorid varchar2(20) not null,

dtype varchar2(20) not null,

dspecialization varchar2(100) not null,

doctorfee number(20) not null,

CONSTRAINT doctor\_pk PRIMARY KEY (doctorid)

);

Alter table doctor add CONSTRAINT fk\_employee FOREIGN KEY (doctorid) REFERENCES employee (empid);

CREATE TABLE nurse

(

nurseid varchar2(20) not null,

nursecharges number(20) not null,

CONSTRAINT nurse\_pk PRIMARY KEY (nurseid)

);

Alter table nurse add CONSTRAINT fk\_employee\_nurse FOREIGN KEY (nurseid) REFERENCES employee (empid);

CREATE TABLE receptionist

(

receptionistid varchar2(20) not null,

password varchar2(20) not null,

CONSTRAINT receptionist\_pk PRIMARY KEY (receptionistid)

);

Alter table receptionist add CONSTRAINT fk\_employee\_recep FOREIGN KEY (receptionistid) REFERENCES employee (empid);

CREATE TABLE walkinpatient

(

walkinpateintid varchar2(20) not null,

tokennumber number(20) not null,

doctorid varchar2(20) not null,

CONSTRAINT walkinpatient\_pk PRIMARY KEY (walkinpateintid)

);

Alter table walkinpatient add CONSTRAINT fk\_walkin\_patient FOREIGN KEY (WALKINPATEINTID) REFERENCES patient (patientid);

Alter table add walkinpatient add CONSTRAINT fk\_doctor FOREIGN KEY (doctorid) REFERENCES doctor (doctorid);

CREATE TABLE appointmentpatient

(

appointmentpatientid varchar2(20) not null,

time timestamp not null,

doctorid varchar2(20) not null,

CONSTRAINT appointmentpatient\_pk PRIMARY KEY (appointmentpatientid)

);

Alter table appointmentpatient add CONSTRAINT fk\_patient FOREIGN KEY (appointmentpatientid) REFERENCES patient (patientid);

Alter table appointmentpatient add CONSTRAINT fk\_doctor FOREIGN KEY (doctorid) REFERENCES doctor (doctorid)

CREATE TABLE admitpatient

(

admitpatientid varchar2(20) not null,

roomcharges number(20) not null,

duration number(20) not null,

patientdetails varchar2(50) not null,

dateofadmission date not null,

dateofdischarge date not null,

doctorid varchar2(20) not null,

nurseid varchar2(20) not null,

CONSTRAINT admitpatient\_pk PRIMARY KEY (admitpatientid)

);

Alter table admitpatient add CONSTRAINT fk\_patient FOREIGN KEY (admitpatientid) REFERENCES patient (patientid)

Alter table admitpatient add CONSTRAINT fk\_doctor FOREIGN KEY (doctorid) REFERENCE doctor (doctorid)

Alter table admitpatient add CONSTRAINT fk\_nurse FOREIGN KEY (nurseid) REFERENCES nurse (nurseid)

CREATE TABLE bill

(

billid varchar2(20) not null,

patientid varchar2(20) not null,

receptionistid varchar2(20) not null,

doctorid varchar2(20) not null,

totalamount number(20) not null,

CONSTRAINT bill\_pk PRIMARY KEY (billid)

);

Alter table bill add CONSTRAINT fk\_doctor\_bill FOREIGN KEY (doctorid) REFERENCES doctor (doctorid);

Alter table bill add CONSTRAINT fk\_receptionist\_bill FOREIGN KEY (receptionistid) REFERENCES receptionist (receptionistid);

Alter table bill add CONSTRAINT fk\_patient FOREIGN KEY (patientid) REFERENCE patient (patientid);

CREATE TABLE patient

(

patientid varchar2(20) not null,

pname varchar2(20) not null,

paddress varchar2(20) null,

pphone number(20) not null,

pstatus varchar2(20) not null,

CONSTRAINT patient\_pk PRIMARY KEY (patientid)

);

**QUERIES:**

For the retrieval and insertion of data into the database, the queries we added in the front end coding are:

1. "select \* from receptionist where receptionistid = '" + textBox1.Text + "' and password = '" + textBox2.Text + "'"
2. "select \* from employee where empid = '" + textBox5.Text + "'"
3. "insert into employee (empid, ename, eaddress, enumber,ejob,esal,etimings) values (" + textBox5.Text + ",'" + textBox4.Text + "','" + textBox3.Text + "','" + textBox2.Text + "','" + textBox1.Text + "','" + textBox6.Text + "','" + textBox7.Text + "')"
4. "select \* from employee"
5. "select ephone,ename,eaddress,ejob,esal,etimings from employee where empid = " + textBox5.Text + ""
6. "select \* from doctor where doctorid = '" + textBox5.Text + "'"
7. "insert into doctor (doctorid, dtype, dspecialization, doctorfee) values (" + textBox5.Text + ",'" + textBox8.Text + "','" + textBox9.Text +"','" + textBox10.Text + "')"
8. "select \* from doctor"
9. "select pphone,pname,paddress,pstatus from patient where patientid = " + textBox1.Text + ""
10. "select timings,ename from employee where empid ='" + textBox7.Text + "'"
11. "select \* from appointmentpatient where appointmentpatientid = '" + textBox1.Text + "'"
12. "insert into appointmentpatient (appointmentpatientid, time,doctorid) values (" + textBox1.Text + ",'" + textBox6.Text + "','" + textBox7.Text + "')"
13. "select \* from appointmentpatient"
14. "insert into admitpatient (admitpatientid, doctorid,dateofadmission,dateofdischarge,patientdetails,roomcharges,duration,nurseid) values (" + textBox1.Text + ",'" + textBox7.Text + "','" + textBox12.Text + "','" + textBox13.Text + "','" + textBox10.Text + "','"+ textBox6.Text +"','"+ textBox11.Text +"','"+ textBox14.Text +"')"
15. "select timings,ename from employee where empid ='" + textBox14.Text + "'"
16. "insert into nurse (nurseid,nursecharges) values (" + textBox5.Text + ",'" + textBox8.Text + "')"
17. "select ephone,ename,eaddress,ejob,esal,etimings from employee where empid = " + textBox5.Text + ""
18. "insert into receptionist (receptionistid,password) values (" + textBox5.Text + ",'" + textBox8.Text + "')"
19. "insert into walkinpatient (walkinpatientid, tokennumber,doctorid) values (" + textBox1.Text + ",'" + textBox6.Text + "','" + textBox7.Text +"')"