EHR System

“Electronic Health Record system of patient medical history, for a faster and better healthcare service”

Problem Statement :

Health data is more than just statistics or numbers. It can be collected, used, and shared in lots of different ways. But ignoring specific medical data can change the way you are going to treat, how your care is provided and what happens to you as a result.

It is imperative to maintain efficient software to handle the information of a Patient. This application provides a way to record this information and to access these in a simple way.

Aim :

We are aiming to build an electronic healthcare record system to keep a track of patient's records that will ease the process of storing and retrieval of information of a particular patient and prevent waste of time in case of emergency.The system would also provide excellent security of data at every level of user-system interaction and also provide robust & reliable storage and backup facilities. Our main objective is to lower the loss of medical records, make hospitals paperless so as to save trees and which in turn would diminish the need for double testing. Now as the records would be stored in a common database, it would thus reduce the time taken for diagnosis and treatment and lessen the time taken to communicate between departments. Our database would also make it easier for the doctor and the patients to check onto their own history of diseases, treatments, lab reports and prescriptions. This would help both the patient to make informed choices during their treatment based on their history and for the doctor to prescribe based on the patient's history. Our database would also make it easier for statistical analysis of a country's health situation as it would share the data with the administration. This would also help as it would make it easier to contact the administration of another hospital about the details of some patients which would make it easier to refer patients. Our system would also make it easier for the visitors to get the details of their respective patients and would show them what all timings are there for them to meet their loved ones. Last but not the least our database would make it easier to buy the prescribed medicine as that information could be shared with the pharmaceuticals. This would be helpful because it will Increase readability /legibility, hence reduce medical errors and handwritten prescribed errors and also would eliminate prescription forging.

Special Facility :

1. EHRs can also trigger warnings and reminders when a patient should get a new lab test or track prescriptions to see if a patient has been following doctors' orders.
2. Use of computer vision in uploading and scanning the patient reports and get some input for further use.
3. We would also keep a record of DGHS/CGHS and the patient's medical insurance facility to help patients.

Roles of the Stakeholders:

1. **Patients**: The patients will be able to look into their medical records and schedule their appointments. They will also have all of their records on a handy database which they can access from anywhere.

2. **Doctors/Nurses** : The doctors/ nurses would use the system for checking their appointments and for prescribing the patients medicine on the basis of their previous ailments.

3. **Government** : The government would use the database of the patients for surveying on different topics and to make doctors/nurses available to hospitals with shortage of staff.

4. **Visitors** : They would use the system to get the information of their patients and would also be shown what are the available time slots on which they can meet them.

5. **Pharmacist** : The patients and doctors would be able to see the medicines in stock. This would heavily decrease the man labour required.

6. **Hospital Departments**: This will make the transfer of patients from one department to the other as the flow of information is eased through our database.

Key Stakeholders:

1. Patient

a. What are the upcoming appointments ?
b. When they were diagnosed with a particular disease ?
c. What is the health history of the patient since the treatment in this hospital?
d. What are the previous and current prescriptions?
e. What are the details/contacts of nurses and doctors that are engaged with the treatment of the patient?
f. How many other specialists/doctors are available in the hospital?
g. How many and what type of rooms are available (semi private / private)?

2. Doctors/ Nurses

a. What are the details of the patient's who are going to get treated by the specific doctor?
b. What are the diseases patient suffered from in the past? (to treat them accordingly)
c. What are their appointments on a given day?
d. Are there labs in the hospital for blood tests, ct scans and any other prescribed tests?
e. What medicines are available in the hospital's pharmacy?

f. What are the working hours/shifts of the doctor/nurse?

3. Administration/ Government

a. What are the details of doctors and nurses?(Name,Currently Posted,DOJ,Department,Availability)
b. What kind of equipment is available to the patients?
c. How many patients are suffering with a particular disease in a given period in a given region?
d. What are the patient details,who had used the government medical insurance?
e. How many crates of medicines are supplied and used by the hospitals?
f. How many number of patients that are being treated at a selected government hospitals
g. Can check and update the prices of medicines.
h. What is the number of patients who came from other cities for their treatment at a hospital?

4. Visitors

a. What time and date are available for the visitor to meet their patients?
b. What is the room number of their patient?
c. How many visitors visited the patient?
d. What are the names,age, gender of visitors?
e. Where do the visitors stay and their addresses?
f. What are the phone numbers and mail id's of the visitors?

5. Pharmacist

a. What are the prescribed medicines?
b. What are the records of medicines being sold at frequent intervals?
c. What are the records of billings?
d. What are the medicines that are in stock?

e. What are the discounts being offered to the customers?(selected customer)
f. What are the records of the payment defaulters?

6. Hospital Departments

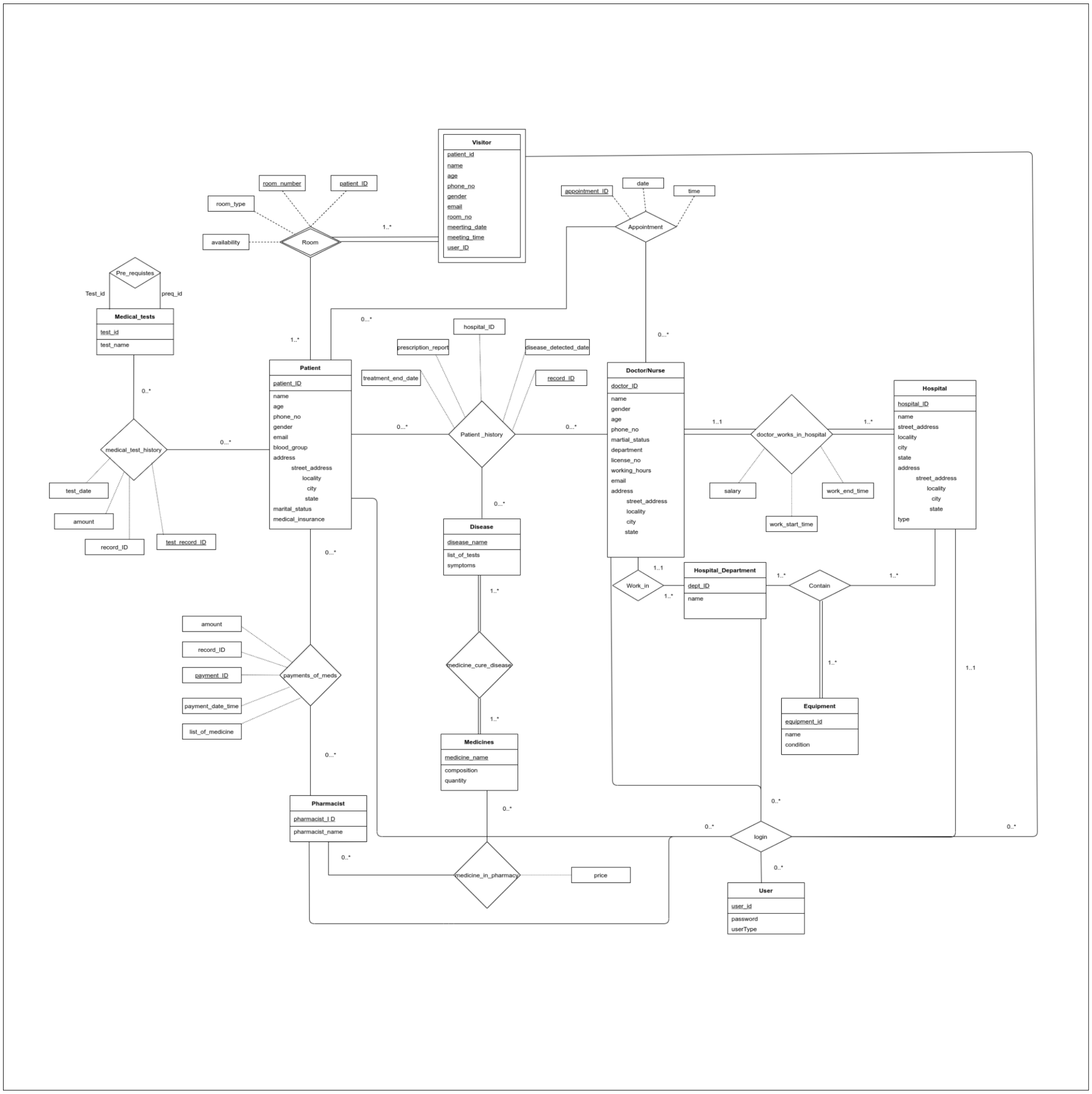
a. What are the medical records that can help medical practitioners?
b. What are the records of doctors, nurses, and patients?
c. What are the details of the equipment that are installed in the department?
d. What are the records of the payment defaulters that can help the hospital department?
e. What is the faulty equipment in the hospital departments?

Assumption

1. Admin and Government are the same in our database. They will have the access to all the information in our database so as to monitor the Hospital's activity.
2. User table would contain the user id and password of all the stakeholders i.e, govt/admin , doctors/nurses , patients, visitors, pharmacist,hospital dept.
3. We are making this data-base to be used in a hospital. So the entity named hospital would just be storing the details of the hospital and not the details of many different hospitals.

E-R Diagram

Link -



https://drive.google.com/file/d/1FuBNLd3F-ZcIrsAnyswDqH16tBz-CrH_/view?usp=sharing

Relation Schemas and Tables:

- **Patient**
  a. patientID : integer(20), primary key, not null
  b. userID : varchar(20), references User(userID),not null ,on delete cascade, on update cascade
  c. patientName: varchar(100) , not null
  d. age: integer(3) , not null , check (age>0)
  e. phoneNo: varchar(10) , not null
  f. gender: varchar(10) , not null, check in(“M”,”F”,”O”)
  g. email: varchar(100)
  h. bloodGroup: varchar(50) , not null
  i. streetAddress : varchar(100)
  j. locality : varchar(100), not null
  k. city : varchar(100), not null
  l. state : varchar(100), not null
  m. address :
     i. streetAddress
     ii. locality
     iii. city
     iv. state

     not null
  n. medicalInsurance : varchar(100) , not null , check in (“Yes” ,“No”)

- **Doctors/Nurses**
  a. doctorID : integer(20), primary key, not null
  b. userID : varchar(20), references User(userID),not null ,on delete cascade, on update cascade
  c. doctorName: varchar(100) , not null
  d. age: integer(3) , not null
  e. phoneNo: varchar(10), not null
  f. gender: varchar(10) , not null, check in(“M”,”F”,”O”)
  g. department: varchar(50) , not null
  h. workingHours: varchar(50) , not null
  i. licenseNumber: varchar(50), not null

    j. email: varchar(100)
    k. address :
        i. streetAddress
        ii. locality
        iii. city
        iv. state
    not null

- **Visitors**
    a. patientID : integer(20), references Patients(patientID), primary key ,not null ,on delete cascade, on update cascade
    b. userID : varchar(20), references User(userID),not null ,primary key ,on delete cascade, on update cascade
    c. visitorName: varchar(100) , not null
    d. age: integer(3) , not null
    e. phoneNo: varchar(10), not null
    f. gender: varchar(10) ,
    g. email: varchar(100)
    h. roomNo : integer(10) , not null
    i. meetingDate : date , not null
    j. meetingTime : time , not null

- **Hospital**
    a. hospitalID : varchar(20), primary key, not null
    b. userID : varchar(20), references User(userID),not null ,on delete cascade, on update cascade
    c. hospitalName : varchar(100), not null
    d. streetAddress : varchar(100) not null
    e. locality : varchar(100) not null
    f. city : varchar(100), not null
    g. state : varchar(100), not null
    h. address :
        i. streetAddress
        ii. locality
        iii. city

iv. state

not null

h. type : varchar(100), check in (“Government”, ”Private”)

- **Pharmacist**
  a. pharmacistID : integer(20), primary key, not null
  b. pharmacistName : varchar(100), not null
  c. userID : varchar(20), references User(userID),not null ,on delete cascade, on update cascade

- **Rooms**
  a. roomNumber: Integer(10),not null
  b. roomType: varchar(20)
  c. availability: varchar(20), check in (“available, not available”)
  d. patientID : integer(20), references Patients(patientID), not null
  e. Primary key (patientID, roomNumber)

- **Equipment**
  a. equipID : varchar(20) , primary key, not null
  b. equipName : varchar(100), not null
  c. equipCondition : varchar(50), check in (“working, not working”)

- **MedicalTests**
  a. testID : integer(20), primary key, not null
  b. testName : varchar(100), not null
  c. preRequisites : varchar(100), not null

- **Contain**
  a. hospitalID :varchar(20), references Hospitals(hospitalID)
  b. equipID : varchar(20) , reference Equipment(equipID)
  c. departID : varchar(20) , reference Department(departID)

d. Primary key (hospitalID,equipID , departID)

- **Appointment**
  a. appointmentID : varchar(20) , primary key, not null
  b. appointDate : date , not null
  c. appointTime : time , not null
  d. patientID : integer(20), references Patients(patientID), not null
  e. doctorID : integer(20), references Doctors/Nurses(doctorID), not null

- **MedicalTestsHistory**
  a. testRecordID : varchar(20), primary key, not null
  b. testID : integer(20), references MedicalTests(testID), not null, on delete cascade, on update cascade
  c. recordID : integer(20), references PatientHistory(recordID), not null, on delete cascade, on update cascade
  d. patientID : integer(20), references Patient(patientID), not null
  e. testDate : date, not null
  f. amountPaid : numeric(10,2), not null, check (amount>0)

- **Diseases**
  a. diseaseName : varchar(100), not null
  b. listOfTests: varchar(500)
  c. symptoms : varchar(500)
  d. medicineName : varchar(100), references Medicines (medicineName), not null, delete on cascade, update on cascade
  e. primary key(diseaseName,medicineName)

- **PatientHistory**
  a. recordID : integer(20), primary key, not null
  b. diseaseName : varchar(100), references Diseases(diseaseName), not null , on update cascade
  c. diseaseDetectedDate : date , not null

d. treatmentEndDate : date ,
e. patientID : integer(20), references Patients(patientID), not null, on delete cascade, on update cascade
f. doctorID : integer(20), references Doctors(doctorID), not null, on delete cascade, on update cascade
g. hospitalID : varchar(20), references Hospitals(HospitalID), , not null, on delete cascade, on update cascade
h. prescriptionReport : varchar(20), not null

- **Medicines**
  a. medicineName : varchar(100), primary key, not null
  b. composition: varchar(500), not null
  c. quantity : integer(20) , not null

- **MedicinesInPharmacy**
  a. medicineName : varchar(100), references Medicines(medicineName), not null ,delete on cascade, update on cascade
  b. pharmacistID : integer(20), references Pharmacist(pharmacistID), not null, delete on cascade, update on cascade
  c. price : numeric(10,2), not null
  d. primary key(pharmacistID, medicineName)

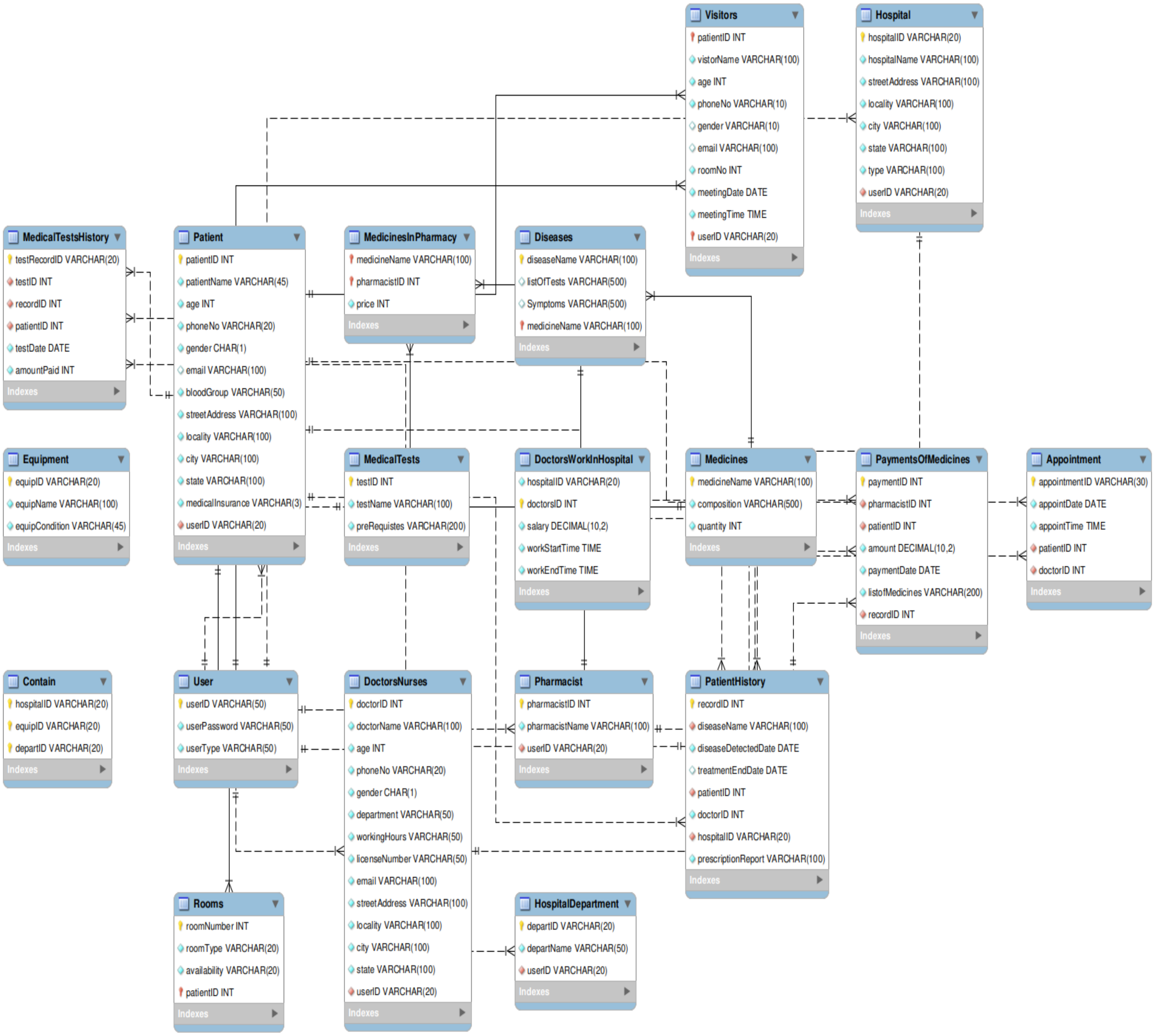
- **DoctorsWorkInHospital**
  a. hospitalID : varchar(20), references Hospitals(Hospital_ID) , not null
  b. doctorID : integer(20), references Doctors(doctor_ID), primary key ,not null
  c. salary : numeric(10,2), not null, check(Salary>0)
  d. workStartTime : time(0), not null
  e. workEndTime : time(0), not null

- **PaymentsOfMedicines**
    a. paymentID : integer(20), primary key, not null
    b. pharmacistID : integer(20), reference Pharmacist(pharmacistID), not null, on delete cascade, on update cascade
    c. patientID : integer(20), references Patient(patientID), not null, on delete cascade, on update cascade
    d. amount : numeric(10,2), not null, check (Amount>0)
    e. paymentDate : date, not null
    f. listOfMedicines : varchar(200), not null
    g. recordID : integer(20), references PatientHistory(recordID), not null ,on delete cascade, on update cascade

- **Hospital Department**
    a. departID : varchar(5), primary key, not null
    b. departName : varchar(40)
    c. userID : varchar(20), references User(userID),not null ,on delete cascade, on update cascade

- **User**
    a. userID : varchar(20), primary key, not null
    b. password : varchar(20), not null
    c. userType : varchar(20), not null, check in (“Patient, Doctor/Nurses,Hospital Department”)

Schema Diagram:



Queries involving various relational algebraic Operations:

1. Patients who don't have medical insurance

   SQL : SELECT * FROM patient WHERE medicalInsurance = ‘No’

2. Details of patients who took more than a year to recover from any disease

   ```
   Select * from Patient where  patientID in (SELECT patientID FROM
   PatientHistory where DATEDIFF(treatmentEndDate,diseaseDetectedDate)
   >365 ) ;
   ```

3. Doctors/Nurses who are going to retire in 5 years

   ```
   SELECT * FROM DoctorsNurses WHERE age>=55 ;
   ```

4. Records of a patients which have symptoms of ‘headache’ and ‘fever’

   ```
   SELECT *FROM PatientHistory WHERE diseaseName IN (Select
   diseaseName FROM Diseases WHERE Symptoms LIKE  '%fever%' OR
   Symptoms LIKE '%headache%' );
   ```

5. The government wants to find out the list of all medicines which are not available in any of the pharmacist

   ```
   SELECT medicineName FROM Medicines WHERE medicineName NOT
   IN (SELECT DISTINCT medicineName FROM MedicinesInPharmacy) ;
   ```

6. The pharmacist wants to check a record with recordID 1009 that the patient claims is corresponding to his prescription

   ```
   SELECT * FROM patientHistory WHERE recordID = 1009;
   ```

7. The government wants to find out the most detected disease

```
SELECT diseaseName, COUNT(*) as count FROM PatientHistory GROUP
BY diseaseName ORDER BY count DESC LIMIT 1 ;
```

8. A hospital wants to know the number of patients that have come to the hospital for their treatment from outside their own city “River Forest”

```
SELECT COUNT(*) as count FROM Patient WHERE city NOT IN ('River
Forest') ;
```

9. Details of the visitors who came in pair/group to visit a patient

```
SELECT vistorName , age, phoneNo, gender, email FROM Visitors
WHERE patientID IN((SELECT patientID FROM Visitors GROUP BY
patientID HAVING COUNT(*) > 1 ) );
```

10. How many private rooms are available at the moment.

```
SELECT COUNT(roomNumber) AS count FROM Rooms WHERE
roomType = 'private' AND availability = 'Available';
```

11. A patientID 10 wants to find out the amount he spent on medical tests when suffered from “chicken-pox”

```
SELECT amountPaid FROM MedicalTestsHistory WHERE recordID IN
(SELECT recordID FROM PatientHistory WHERE patientID = 10 AND
diseaseName = 'chicken-pox') ;
```

Embedded SQL and Advanced Aggregate Function Queries:

1. A Hospital wants to find out the total average salary of doctors/nurses in its hospital for each of its department

```
SELECT  HospitalDepartment.departName ,
AVG(DoctorsWorkInHospital.salary) as average
```

```
FROM DoctorsWorkInHospital
INNER JOIN HospitalDepartment ON DoctorsWorkInHospital.doctorID =
HospitalDepartment.doctorID
GROUP BY HospitalDepartment.departName;
```

2. A Pharmacist with ID 8 wants to find out his monthly , quarterly and yearly sales in the year 2006

   ```
   SELECT YEAR(paymentDate) as year, QUARTER(paymentDate) as
   quarter, MONTH( paymentDate ) as month, SUM(amount) as sales FROM
   PaymentsOfMedicines WHERE pharmacistID = 8  AND YEAR
   ( paymentDate ) = 2006 GROUP BY year, quarter, month WITH ROLLUP ;
   ```

3. A Hospital wants to rank its doctors based on the number of patient they treated

   ```
   SELECT  DoctorsNurses.doctorID ,DoctorsNurses.doctorName ,
   COUNT(PatientHistory.doctorID) as 'count' , RANK() OVER (ORDER BY
   COUNT(PatientHistory.doctorID) desc) as rankOfDoctor
   FROM DoctorsNurses
   INNER JOIN PatientHistory ON PatientHistory.doctorID =
   DoctorsNurses.doctorID
   GROUP BY PatientHistory.doctorID ORDER BY 'count' DESC;
   ```

4. A Hospital wants to know the maximum and minimum costing medicines that each pharmacists have

   ```
   SELECT PharmacistID , min(price) AS 'min' , max(price) AS 'max' FROM
   MedicinesInPharmacy GROUP BY pharmacistID ;
   ```