Hospital Database Management

CS 02-530 Database Design Project

Rowan University,Spring 2023
03/10/2023

Kavya Mistry

I. Domain Overview and Requirements

An essential tool for handling medical and administrative data in a healthcare business is a hospital database system. The system's functionality is intended to assist effective data management, enhance patient care, and simplify administrative procedures. Some of the common use cases are as described below:

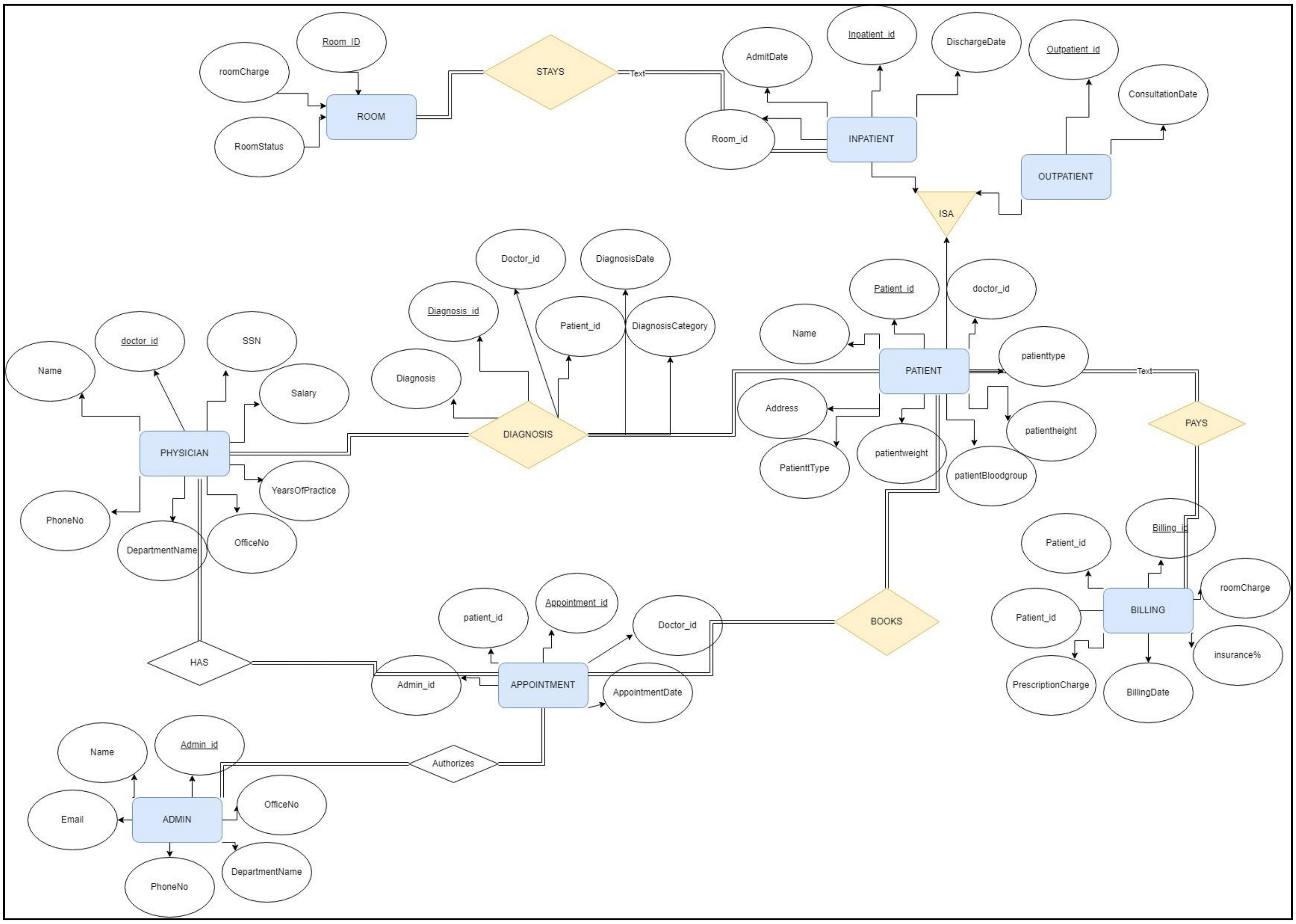
1. Registration and admission of patients
2. Management and scheduling of appointments
3. Prescription drug Charge
4. Medication administration/ Diagnosis
5. Patient discharge and ongoing care for Inpatients and Outpatients
6. Management of billing and insurance
7. Analytics and reporting

In my project, if the patient has previously visited a hospital, the system would be updated; if this is the patient's first visit, the system would be constructed. The patient calls and makes an appointment reservation in advance. Based on the doctor's availability, the administrator will choose a time for the appointment (offline). Following an appointment, the doctor gives the patient a correct diagnosis, which is then entered into the system, essentially updating the patient's medical data. One day after the patient leaves the hospital, the service costs will be added up following the visit and given to them. An administrator checks the availability of rooms if the patient has been admitted, and the patient is given a room. An administrator changes the patient's "type" from inpatient to outpatient after the patient has been released. The day following a patient's release, the bill is presented.

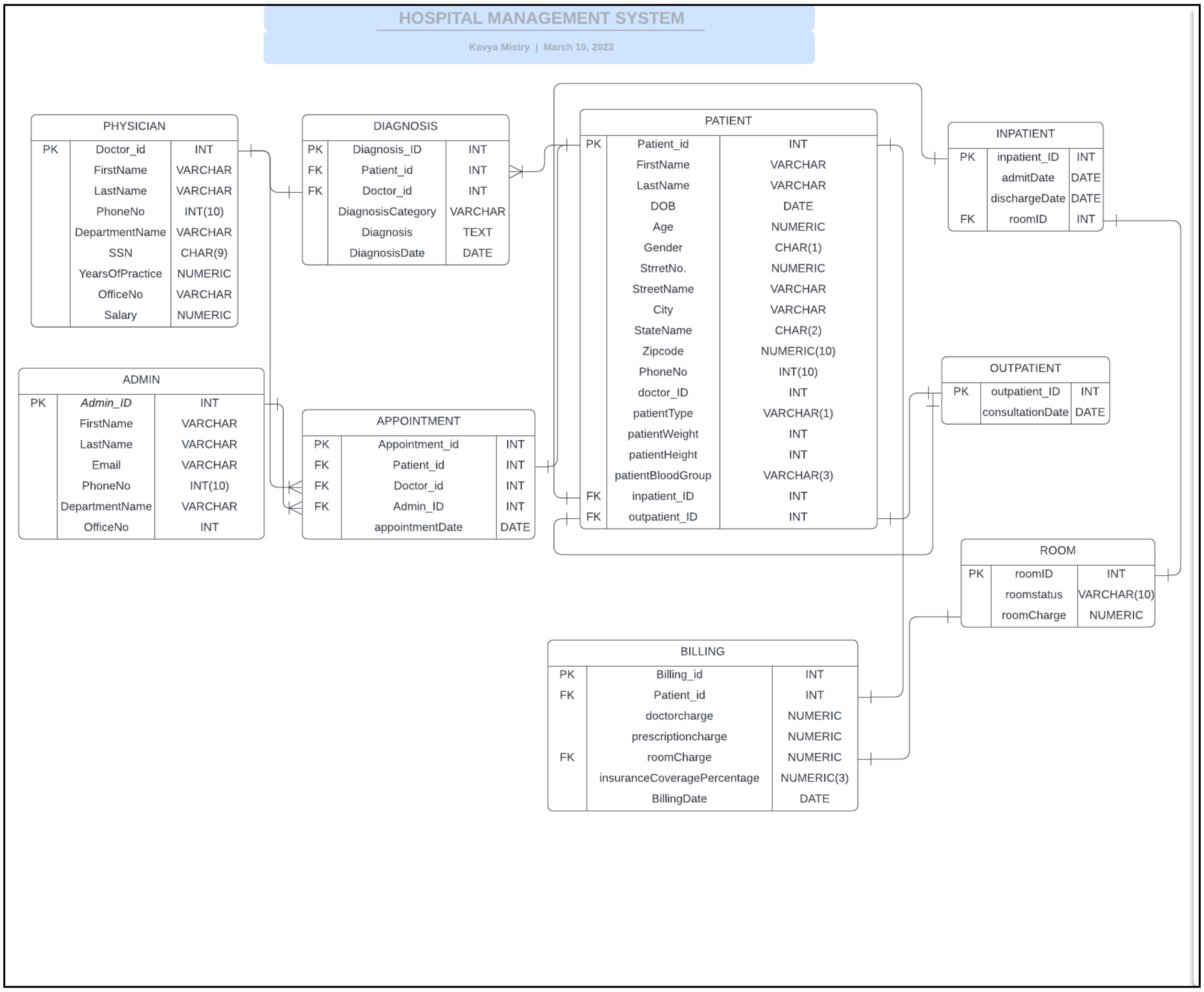
A few business rules to be assumed for the Hospital DBMS is as follows:

1. Each physician may have a unique service or consultation cost.
2. Just one room can be designated for each inpatient, however each room can accommodate up to two patients.
3. The day following an appointment, bills are sent out.
4. The day after a patient is released from the hospital after being hospitalized, they receive their bill.
5. A doctor is only allowed to visit a maximum of 10 patients on any given day.
6. Those who go to the hospital for a regular checkup are classified as outpatients.
7. A maximum of three appointments may be made each week by patients.
8. We are presuming that each department can only contain one type of physician (a cardiologist in the department of cardiology, a nephrologist in the department of nephrology).
9. Nightly fees are charged for hospital accommodations.

II. Entity Relationship Diagram



III. Relational Design Diagram



HOSPITAL MANAGEMENT SYSTEM

Kavya Mistry | March 10, 2023

PHYSICIAN

PK	Doctor_id	INT
	FirstName	VARCHAR
	LastName	VARCHAR
	PhoneNo	INT(10)
	DepartmentName	VARCHAR
	SSN	CHAR(9)
	YearsOfPractice	NUMERIC
	OfficeNo	VARCHAR
	Salary	NUMERIC

DIAGNOSIS

PK	Diagnosis_ID	INT
FK	Patient_id	INT
FK	Doctor_id	INT
	DiagnosisCategory	VARCHAR
	Diagnosis	TEXT
	DiagnosisDate	DATE

PATIENT

PK	Patient_id	INT
	FirstName	VARCHAR
	LastName	VARCHAR
	DOB	DATE
	Age	NUMERIC
	Gender	CHAR(1)
	StrretNo.	NUMERIC
	StreetName	VARCHAR
	City	VARCHAR
	StateName	CHAR(2)
	Zipcode	NUMERIC(10)
	PhoneNo	INT(10)
	doctor_ID	INT
	patientType	VARCHAR(1)
	patientWeight	INT
	patientHeight	INT
	patientBloodGroup	VARCHAR(3)
FK	inpatient_ID	INT
FK	outpatient_ID	INT

INPATIENT

PK	inpatient_ID	INT
	admitDate	DATE
	dischargeDate	DATE
FK	roomID	INT

ADMIN

PK	*Admin_ID*	INT
	FirstName	VARCHAR
	LastName	VARCHAR
	Email	VARCHAR
	PhoneNo	INT(10)
	DepartmentName	VARCHAR
	OfficeNo	INT

APPOINTMENT

PK	Appointment_id	INT
FK	Patient_id	INT
FK	Doctor_id	INT
FK	Admin_ID	INT
	appointmentDate	DATE

OUTPATIENT

PK	outpatient_ID	INT
	consultationDate	DATE

ROOM

PK	roomID	INT
	roomstatus	VARCHAR(10)
	roomCharge	NUMERIC

BILLING

PK	Billing_id	INT
FK	Patient_id	INT
	doctorcharge	NUMERIC
	prescriptioncharge	NUMERIC
FK	roomCharge	NUMERIC
	insuranceCoveragePercentage	NUMERIC(3)
	BillingDate	DATE

IV. Normalization

Normalization is the process of organizing data in a relational database to reduce redundancy and improve data integrity. The process involves breaking down large tables into smaller ones and establishing relationships between them. Normalization is usually done up to the third normal form (3NF), which is the most widely used normal form in practice.

In the case of a hospital database management system (DBMS), we have the following tables: Admin, Patient, Physician, Appointment, Diagnosis, Billing, Room, Inpatient and Outpatient.

First Normal Form (1NF)

The first step in normalization is to ensure that each table is in first normal form (1NF). To achieve 1NF, we need to ensure that each table has a primary key and that each attribute in the table is atomic (indivisible).
Admin table:
AdminID (PK), AdminName, AdminEmailAddress, AdminPhoneNo, OfficeNo, departmentName

Patient table:
PatientID (PK), PatientName, PatientAddress, PatientPhone, Gender, DateOfBirth, Age, PatientType,PatientWeight, PatientHeight, PatientBloodGroup, DoctorID (FK)

Physician table:
DoctorID (PK), PhysicianName, PhysicianPhone, SSN, Years Of Practice, Salary

Appointment table:

AppointmentID (PK), PatientID (FK), PhysicianID (FK), AppointmentDate, AdminID

Diagnosis table:
DiagnosisID (PK), DoctorID(FK),PatientID(FK),DiagnosisCategory, Diagnosis, DiagnosisDate

Billing table:
BillID (PK), Patient_id (FK) ,DoctorCharge,PrescriptionCharge,RoomCharge, InsuranceCoverage,Billing Date

Room table:
RoomID (PK), RoomCharge, RoomStatus

Inpatient table:
InpatientID (PK), PatientID (FK), AdmissionDate, DischargeDate, RoomID (FK)

Outpatient table:
OutpatientID (PK), PatientID (FK), ConsulationDate

Second Normal Form (2NF)

The next step is to ensure that each table is in second normal form (2NF). To achieve 2NF, we need to ensure that each non-key attribute is dependent on the entire primary key.
Admin table:
AdminID (PK), AdminName, AdminEmailAddress, AdminPhoneNo, OfficeNo, departmentName

Patient table:
PatientID (PK), PatientName, PatientAddress, PatientPhone, Gender, DateOfBirth, Age, PatientType,PatientWeight, PatientHeight, PatientBloodGroup, DoctorID (FK)

Physician table:
DoctorID (PK), PhysicianName PhysicianPhone, SSN, Years Of Practice, Salary

Appointment table:
AppointmentID (PK), PatientID (FK), PhysicianID (FK), AppointmentDate, AdminID

Diagnosis table:
DiagnosisID (PK), AppointmentID (FK), DiagnosisDetails

Billing table:
BillID (PK), Patient_id (FK) ,DoctorCharge,PrescriptionCharge,RoomCharge, InsuranceCoverage,Billing Date

Room table:
RoomID (PK), RoomCharge, RoomStatus

Inpatient table:
InpatientID (PK), PatientID (FK), AdmissionDate, DischargeDate, RoomID (FK)

Outpatient table:
OutpatientID (PK), PatientID (FK), ConsulationDate

Third Normal Form (3NF)

The final step is to ensure that each table is in the third normal form (3NF). To achieve 3NF, we need to ensure that each non-key attribute is dependent only on the primary key and not on any other non-key attribute.

Admin table:
AdminID (PK), AdminFirstName, AdminLastName AdminEmailAddress, AdminPhoneNo, OfficeNo, departmentName

Patient table:
PatientID (PK), PatientFirstName, PatientLastName, streetNo, streetName stateName, stateName, zip code, PatientPhone, Gender, DateOfBirth, Age, PatientType,PatientWeight, PatientHeight, PatientBloodGroup, DoctorID (FK)

Physician table:
DoctorID (PK), Physician First Name, Physician Last Name PhysicianPhone, SSN, Years Of Practice, Salary

Appointment table:
AppointmentID (PK), PatientID (FK), PhysicianID (FK), AppointmentDate, AdminID

Diagnosis table:
DiagnosisID (PK), DoctorID(FK),PatientID(FK),DiagnosisCategory, Diagnosis, DiagnosisDate

Billing table:
BillID (PK), Patient_id (FK) ,DoctorCharge,PrescriptionCharge,RoomCharge, InsuranceCoverage,Billing Date

Room table:
RoomID (PK), RoomCharge, RoomStatus

Inpatient table:
InpatientID (PK), PatientID (FK), AdmissionDate, DischargeDate, RoomID (FK)

Outpatient table:
OutpatientID (PK), PatientID (FK), ConsulationDate

V. SQL Statements

1. CREATE STATEMENTS

```
    /*
Hospital Management System
Kavya Mistry
*/

--DROP TABLE ADMIN;
CREATE TABLE ADMIN
(
admin_ID INT NOT NULL PRIMARY KEY, -- primary key column
firstName VARCHAR(20) NOT NULL,
lastName VARCHAR(30) NOT NULL,
phoneNo INT(10) NOT NULL,
emailAddress VARCHAR(25),
departmentName VARCHAR(100) CHECK( departmentName IN ('General
Internal Medicine', 'Cardiology', 'Dermatology', 'Endocrinology',
'Gastroenterology', 'Oncology', 'Epidemiology', 'Nephrology',
'Pharmacology', 'Pulmonology', 'Rheumatology', 'ER')),
officeNo VARCHAR(3)
);

--DROP TABLE PHYSICIAN;
CREATE TABLE PHYSICIAN
(
doctor_ID INT NOT NULL PRIMARY KEY, -- primary key column
firstName VARCHAR(20) NOT NULL,
lastName VARCHAR(30) NOT NULL,
phonenumber INT(10) NOT NULL,
departmentName VARCHAR(100) NOT NULL CHECK( departmentName
IN ('General Internal Medicine', 'Cardiology', 'Dermatology',
'Endocrinology', 'Gastroenterology', 'Oncology', 'Epidemiology',
'Nephrology', 'Pharmacology', 'Pulmonology', 'Rheumatology', 'ER')),
SSN CHAR(9) NOT NULL,
yearsOfPractice NUMERIC DEFAULT 1,
officeNo VARCHAR(3) NOT NULL,
salary NUMERIC NOT NULL CHECK(salary > 0)-- can be different for each
doctor according to experience
);

--DROP TABLE PATIENT;
CREATE TABLE PATIENT
(
patientID INT NOT NULL PRIMARY KEY, -- primary key column
firstName VARCHAR(20) NOT NULL,
lastName VARCHAR(30) NOT NULL,
DOB DATE NOT NULL, --CHECK(DOB <= date('now')),
age NUMERIC NOT NULL CHECK(age >= 0),
gender VARCHAR(2) NOT NULL CHECK(gender IN ('M', 'F', 'None')), --
assigned at birth
streetNo NUMERIC NOT NULL,
streetName VARCHAR(100) NOT NULL,
city VARCHAR(30) NOT NULL,
stateName CHAR(2) NOT NULL, -- Two letter abbreviation for stateName
zipcode NUMERIC(5) NOT NULL,
phoneNo CHAR(10),
doctor_ID INT NOT NULL,
patientType CHAR(1) NOT NULL CHECK(patientType IN ('O','I')),
patientHeight INT CHECK(patientHeight > 0), -- in centimeters (cm)
patientWeight INT CHECK(patientWeight > 0),
patientBloodGroup VARCHAR(3), -- in pounds (lbs)
FOREIGN KEY (doctor_ID) REFERENCES PHYSICIAN(doctor_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

```
--DROP TABLE APPOINTMENT;
CREATE TABLE APPOINTMENT
(
appointment_ID INT NOT NULL PRIMARY KEY,
patientID INT NOT NULL,
doctor_ID INT NOT NULL,
admin_ID INT NOT NULL,
appointmentDate DATE NOT NULL, --CHECK(appointmentDate > date()),
FOREIGN KEY (doctor_ID) REFERENCES PHYSICIAN(doctor_ID) ON
DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (patientID) REFERENCES PATIENT(patientID) ON
DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (admin_ID) REFERENCES ADMIN(admin_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);



--Drop table DIAGNOSIS;
CREATE TABLE DIAGNOSIS
(
diagnosis_ID INT NOT NULL PRIMARY KEY, -- primary key column
doctor_ID INT NOT NULL,
patientID INT NOT NULL,
diagnosisCategory VARCHAR(100) NOT NULL CHECK(diagnosisCategory
IN ('Hypertension','Hyperlipidemia','Diabetes','Back
pain','Anxiety','Obesity','Allergic rhinitis','Respiratory
problems','Hypothyroidism','Visual refractive
errors','Osteoarthritis','Myositis','Pain in joint','Acute maxillary
sinusitis','Major depressive disorder','Acute bronchitis','Asthma','Skin
Disease','Coronary atherosclerosis','Urinary tract
infection','Influenza','Tuberculosis','Viral infection','Seizure
Disorder','Cerebral Palsy','Tourette Syndrome','Attention Deficit
Disorder','Down Syndrome')),
diagnosis TEXT,
diagnosisDate DATE NOT NULL, --CHECK(diagnosisDate <= date())

FOREIGN KEY (doctor_ID) REFERENCES PHYSICIAN(doctor_ID) ON
DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (patientID) REFERENCES PATIENT(patientID) ON
DELETE CASCADE ON UPDATE CASCADE
);


Drop Table BILLING;
CREATE TABLE BILLING
(
billing_ID INT NOT NULL PRIMARY KEY, -- primary key column
patientID INT NOT NULL,
doctorCharge NUMERIC NOT NULL DEFAULT 0,
prescriptionCharge NUMERIC NOT NULL DEFAULT 0,
roomCharge NUMERIC NOT NULL DEFAULT 0,
insuranceCoveragePercentage NUMERIC(3) NOT NULL DEFAULT 0,
billingDate DATE NOT NULL, --CHECK(billingDate >= date())

FOREIGN KEY (patientID) REFERENCES PATIENT(patientID) ON
DELETE CASCADE ON UPDATE CASCADE
);


--Drop Table INPATIENT;
CREATE TABLE INPATIENT
(
inpatient_ID INT NOT NULL PRIMARY KEY, -- primary key column
admitDate DATE NOT NULL, --CHECK(admitDate <= date()),--inpatient is
a hospital patient who, in most cases, stays in the hospital overnight and meets
a set of clinical criteria.
```

```
dischargeDate DATE NULL,
roomID INT NOT NULL,

FOREIGN KEY (inpatient_ID) REFERENCES PATIENT(patientID) ON DELETE CASCADE ON UPDATE CASCADE
FOREIGN KEY (roomID) REFERENCES ROOM(roomID) ON DELETE CASCADE ON UPDATE CASCADE

);


--DROP TABLE OUTPATIENT;
CREATE TABLE OUTPATIENT
(
outpatient_ID INT NOT NULL PRIMARY KEY, -- primary key column
consultation_date DATE, --CHECK(consultation_date = date())--Outpatients are people who receive care or hospital services and return home the same day.

FOREIGN KEY (outpatient_ID) REFERENCES PATIENT(patientID) ON DELETE CASCADE ON UPDATE CASCADE
);

--DROP TABLE ROOM;
CREATE TABLE ROOM
(
roomID INT NOT NULL PRIMARY KEY, -- primary key column
roomStatus VARCHAR(10) NOT NULL, --CHECK(roomStatus IN ('Vacant', 'Occupied')),
roomCharge NUMERIC NOT NULL --CHECK(roomCharge > 0)
);

--DROP TABLE ROOM_INPATIENT;
/*CREATE TABLE ROOM_INPATIENT
(
inpatientID INT NOT NULL, -- primary key column
roomID INT NOT NULL,
PRIMARY KEY (inpatientID,roomID),
FOREIGN KEY (inpatientID) REFERENCES INPATIENT(inpatient_ID),
FOREIGN KEY (roomID) REFERENCES ROOM(roomID)
);*/
```

2. INSERT STATEMENTS

```
--Insert Values into ADMIN Table
INSERT INTO ADMIN VALUES(1,'Kavya',
'Mehta','1111111111','kavyamehta@gmail.com','General Internal
Medicine','112');
INSERT INTO ADMIN VALUES(2,'Mistry',
'Singh','2222222222','mistry@gmail.com','Cardiology','223');
INSERT INTO ADMIN VALUES(3,'Abhinav',
'Goel','3333333333','abhig@gmail.com','Dermatology','334');
INSERT INTO ADMIN VALUES(4,'Dan',
'Cox','4444444444','dannn@yahoo.com','Oncology','445');
INSERT INTO ADMIN VALUES(5,'Sydney',
'Derran','5555555555','derransy@gmail.com','Gastroenterology','556');
INSERT INTO ADMIN VALUES(6,'Sam',
'Downey','6666666666','samjr@gmail.com','Endocrinology','667');
INSERT INTO ADMIN VALUES(7,'Jake',
'Wong','7777777777','wongjake@gmail.com','Pulmonology','778');
INSERT INTO ADMIN VALUES(8,'Katy',
'Nate','8888888888','mskate@gmail.com','Pharmacology','889');
INSERT INTO ADMIN VALUES(9,'Nina',
'Class','9999999999','iamnina@yahoo.com','Nephrology','990');

--Insert Values into PHYSICIAN Table
INSERT INTO PHYSICIAN VALUES(1,'Prince',
'Patel','1234567890','General Internal Medicine','111111111',10,'111',250);
INSERT INTO PHYSICIAN VALUES(2,'Nikita',
'Joshi','1234567678','Cardiology','222222222',2,'222',300);
```

```
INSERT INTO PHYSICIAN VALUES(3,'Payal',
'Parmar','1234563783','Dermatology','333333333',4,'333',125);
INSERT INTO PHYSICIAN VALUES(4,'Swapnil',
'Challuri','1234512345','Oncology','444444444',7,'444',260);
INSERT INTO PHYSICIAN VALUES(5,'Anna',
'Carles','1234524680','Gastroenterology','555555555',1,'555',530);
INSERT INTO PHYSICIAN VALUES(6,'Jill',
'Hade','1234590899','Endocrinology','666666666',9,'666',120);
INSERT INTO PHYSICIAN VALUES(7,'Will',
'Smith','12345000','Pulmonology','777777777',8,'777',550);
INSERT INTO PHYSICIAN VALUES(8,'Jeff',
'Carpenter','1234567676','Pharmacology','888888888',23,'888',400);
INSERT INTO PHYSICIAN VALUES(9,'Amy',
'Russ','5432154321','Nephrology','999999999',1,'999',1000);

--Insert Values into PATIENT Table
INSERT INTO PATIENT VALUES(1,'Tommy',
'Hillfigure','1996-10-10',26,'M',123,'Cary
Road','Manlius','NY',13104,'3153453651',1,'O',172,180,'B+');
INSERT INTO PATIENT VALUES(2,'Dwayne',
'Jonson','1990-11-15',33,'M',234,'Bridge
Avenue','Manlius','NY',13104,'3154256157',3,'I',150,164,'O+');
INSERT INTO PATIENT VALUES(3,'Hugh',
'Jackson','1986-02-09',37,'M',345,'Lorraine
Avenue','Syracuse','NY',16802,'6157267893',2,'O',144,220,'O-');
INSERT INTO PATIENT VALUES(4,'Chris',
'Hemsworth','1964-12-19',59,'M',456,'Carrier
Drive','Liverpool','NY',16803,'5152620092',5,'O',130,135,'B+');
INSERT INTO PATIENT VALUES(5,'Chris',
'Evans','2001-02-02',22,'M',567,'Taft
Lane','Fayetteville','NY',22222,'1236728172',4,'I',190,240,'AB+');
INSERT INTO PATIENT VALUES(6,'Robert',
'Downey','2004-02-04',19,'M',678,'Barksdale
Lane','Baldwinsville','NY',31215,'3334125263',6,'I',115,100,'O+');
```

```
INSERT INTO PATIENT VALUES(7,'Tom',
'Holland','1997-07-21',25,'M',789,'Trillium
Trail','Manlius','NY',13104,'4447267281',9,'I',156,145,'O-');
INSERT INTO PATIENT VALUES(8,'Billie',
'Eilish','1972-01-01',51,'F',890,'Parker
Drive','Fayetteville','NY',22222,'7772891827',8,'I',174,210,'O+');
INSERT INTO PATIENT VALUES(9,'Post',
'Malone','1952-04-20',70,'F',012,'Trout
Road','Syracuse','NY',16802,'7268880290',7,'O',189,214,'B+');

--Insert Values into APPOINTMENT Table
INSERT INTO APPOINTMENT VALUES(100000001,1,1,9,'2023-03-23');
INSERT INTO APPOINTMENT VALUES(100000002,2,3,2,'2023-03-11');
INSERT INTO APPOINTMENT VALUES(100000003,3,6,8,'2023-03-12');
INSERT INTO APPOINTMENT VALUES(100000004,4,5,3,'2023-03-15');
INSERT INTO APPOINTMENT VALUES(100000005,5,4,5,'2023-03-15');
INSERT INTO APPOINTMENT VALUES(100000006,6,2,6,'2023-03-14');
INSERT INTO APPOINTMENT VALUES(100000007,7,8,7,'2023-03-10');
INSERT INTO APPOINTMENT VALUES(100000008,8,7,4,'2023-03-23');
INSERT INTO APPOINTMENT VALUES(100000009,9,9,1,'2023-03-25');

--Insert Values into Diagnosis Table
INSERT INTO DIAGNOSIS VALUES(100000011,1,1,'Hypertension','High
Systolic BP. High Salt Diet, must reduce and take ACE
Inhibitors','2021-09-08');
INSERT INTO DIAGNOSIS VALUES(100000022,2,3,'Diabetes','Type II
Diabetic, must reduce sugar and intake and take Insulin once
daily','2022-10-05');
INSERT INTO DIAGNOSIS VALUES(100000033,3,2,'Asthma','Albuterol for
a month','2019-10-04');
INSERT INTO DIAGNOSIS VALUES(100000044,4,5,'Anxiety','Reduced
levels of Serotonin in the brain, perscribed Alazopram 0.5 mg for three
months. Take as needed','2022-08-16');
```

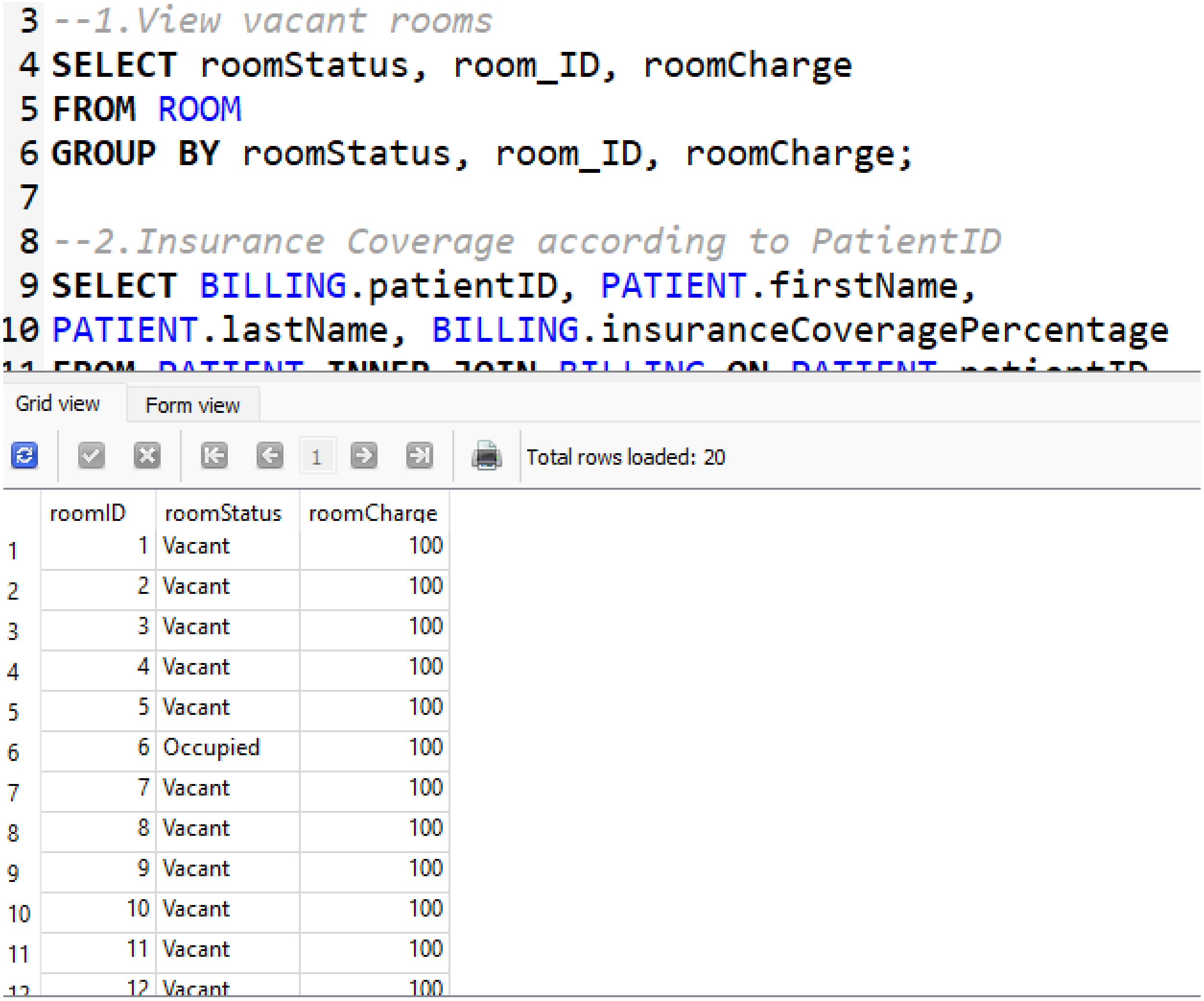
```
INSERT INTO DIAGNOSIS VALUES(100000055,5,4,'Allergic
rhinitis','Inflamed sinus, stuffy nose for 2 weeks. Take OTC Allegra, Benadryl
or Claritin from local pharmacy','2021-04-07');
INSERT INTO DIAGNOSIS VALUES(100000066,6,6,'Obesity','Referred to
Dietician. Must reduce sugar intake and exercise regularly','2020-05-11');
INSERT INTO DIAGNOSIS
VALUES(100000077,7,9,'Hypothyroidism','Hyperactive thyroid leading to
weight gain and lack of hunger. Take Levothyroxine: 10 mg/day for 3 months
and schedule an additional appointment within the year.','2019-11-10');
INSERT INTO DIAGNOSIS VALUES(100000088,8,8,'Osteoarthritis','Joint
Pain in left knee following lifting boxes. Take X-Ray of joint and increase
Calcium intake','2019-11-09');
INSERT INTO DIAGNOSIS VALUES(100000099,9,7,'Acute
bronchitis','Severe coughing fits. Prescribed inhaler from nearest
pharmacy','2019-11-10');

--Insert Values into Billing Table
INSERT INTO BILLING VALUES(100000111,1,250,40,0,80,'2021-10-08');
INSERT INTO BILLING VALUES(100000222,2,125,400,100,60,'2022-10-06');
INSERT INTO BILLING VALUES(100000333,3,300,80,0,40,'2019-12-05');
INSERT INTO BILLING VALUES(100000444,4,530,0,0,10,'2022-09-19');
INSERT INTO BILLING VALUES(100000555,5,260,90,100,0,'2021-04-08');
INSERT INTO BILLING VALUES(100000666,6,120,0,100,55,'2020-05-12');
INSERT INTO BILLING
VALUES(100000777,7,1000,15,100,100,'2019-11-11');
INSERT INTO BILLING VALUES(100000888,8,400,0,100,15,'2019-11-10');
INSERT INTO BILLING VALUES(100000999,9,550,100,0,45,'2020-01-10');

--Insert Values into Inpatient Table
INSERT INTO INPATIENT VALUES(1,'2019-10-04','2019-10-05',1);
INSERT INTO INPATIENT VALUES(2,'2019-08-16','2019-08-17',2);
INSERT INTO INPATIENT VALUES(3,'2019-05-11','2019-05-12',3);
INSERT INTO INPATIENT VALUES(4,'2019-11-10','2019-11-11',4);
INSERT INTO INPATIENT VALUES(5,'2019-11-09','2019-11-10',5);
```

```
--Insert Values into Outpatient Table
INSERT INTO OUTPATIENT VALUES(1,'2019-09-08');
INSERT INTO OUTPATIENT VALUES(2,'2019-10-02');
INSERT INTO OUTPATIENT VALUES(3,'2019-04-06');
INSERT INTO OUTPATIENT VALUES(4,'2019-07-16');

--Insert Values into ROOM Table
INSERT INTO ROOM VALUES(1,'Vacant',100);
INSERT INTO ROOM VALUES(2,'Vacant',100);
INSERT INTO ROOM VALUES(3,'Vacant',100);
INSERT INTO ROOM VALUES(4,'Vacant',100);
INSERT INTO ROOM VALUES(5,'Vacant',100);
INSERT INTO ROOM VALUES(6,'Vacant',100);
INSERT INTO ROOM VALUES(7,'Vacant',100);
INSERT INTO ROOM VALUES(8,'Vacant',100);
INSERT INTO ROOM VALUES(9,'Vacant',100);
INSERT INTO ROOM VALUES(10,'Vacant',100);
INSERT INTO ROOM VALUES(11,'Vacant',100);
INSERT INTO ROOM VALUES(12,'Vacant',100);
INSERT INTO ROOM VALUES(13,'Vacant',100);
INSERT INTO ROOM VALUES(14,'Vacant',100);
INSERT INTO ROOM VALUES(15,'Vacant',100);
INSERT INTO ROOM VALUES(16,'Vacant',100);
INSERT INTO ROOM VALUES(17,'Vacant',100);
INSERT INTO ROOM VALUES(18,'Vacant',100);
INSERT INTO ROOM VALUES(19,'Vacant',100);
INSERT INTO ROOM VALUES(20,'Vacant',100);
```

3. SELECT STATEMENTS/USE CASES

--Testing USE cases

--1.View vacant rooms
SELECT roomStatus, room_ID, roomCharge
FROM ROOM
GROUP BY roomStatus, room_ID, roomCharge;

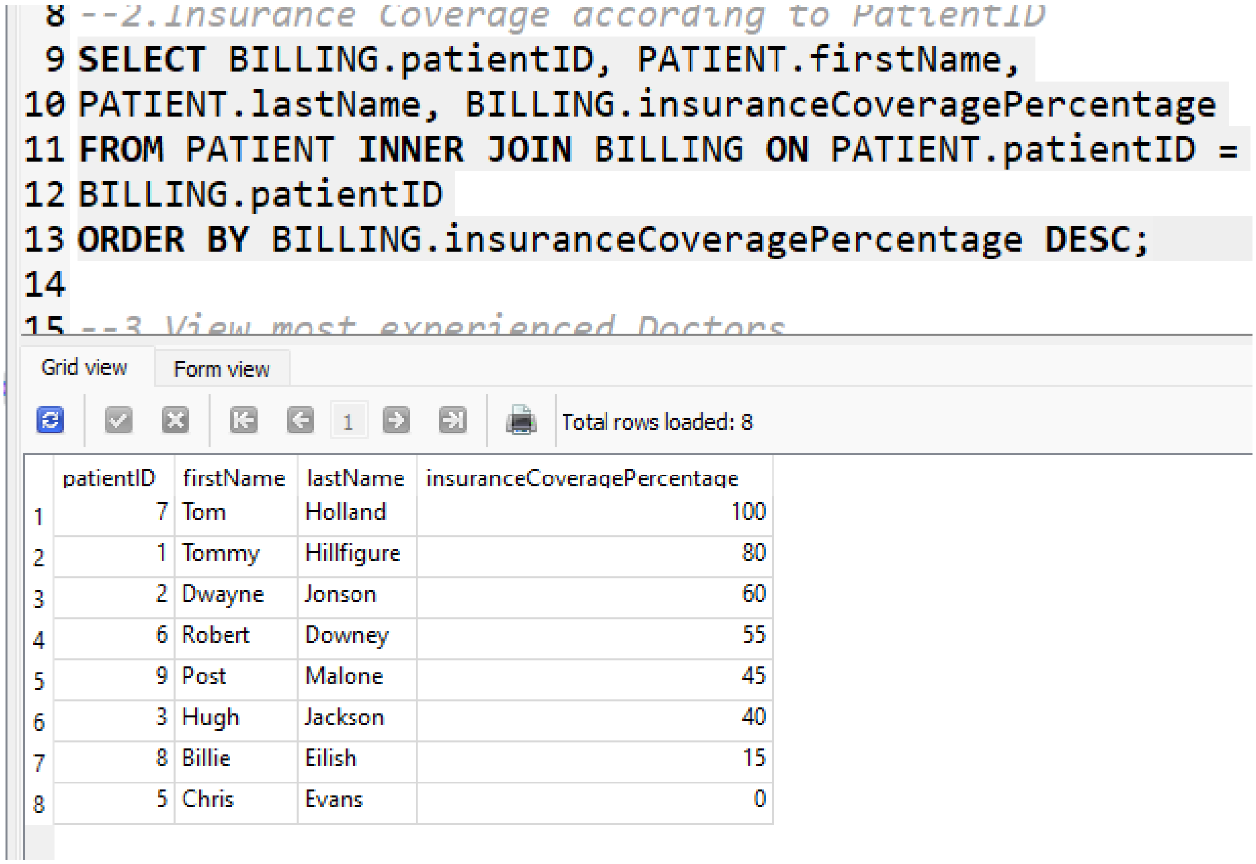


```
--1.View vacant rooms
SELECT roomStatus, room_ID, roomCharge
FROM ROOM
GROUP BY roomStatus, room_ID, roomCharge;

--2.Insurance Coverage according to PatientID
SELECT BILLING.patientID, PATIENT.firstName,
PATIENT.lastName, BILLING.insuranceCoveragePercentage
FROM PATIENT INNER JOIN BILLING ON PATIENT.patientID
```

Total rows loaded: 20

	roomID	roomStatus	roomCharge
1	1	Vacant	100
2	2	Vacant	100
3	3	Vacant	100
4	4	Vacant	100
5	5	Vacant	100
6	6	Occupied	100
7	7	Vacant	100
8	8	Vacant	100
9	9	Vacant	100
10	10	Vacant	100
11	11	Vacant	100
12	12	Vacant	100

--2.Insurance Coverage according to PatientID
SELECT BILLING.patientID, PATIENT.firstName,
PATIENT.lastName, BILLING.insuranceCoveragePercentage
FROM PATIENT INNER JOIN BILLING ON PATIENT.patientID =
BILLING.patientID
ORDER BY BILLING.insuranceCoveragePercentage DESC;

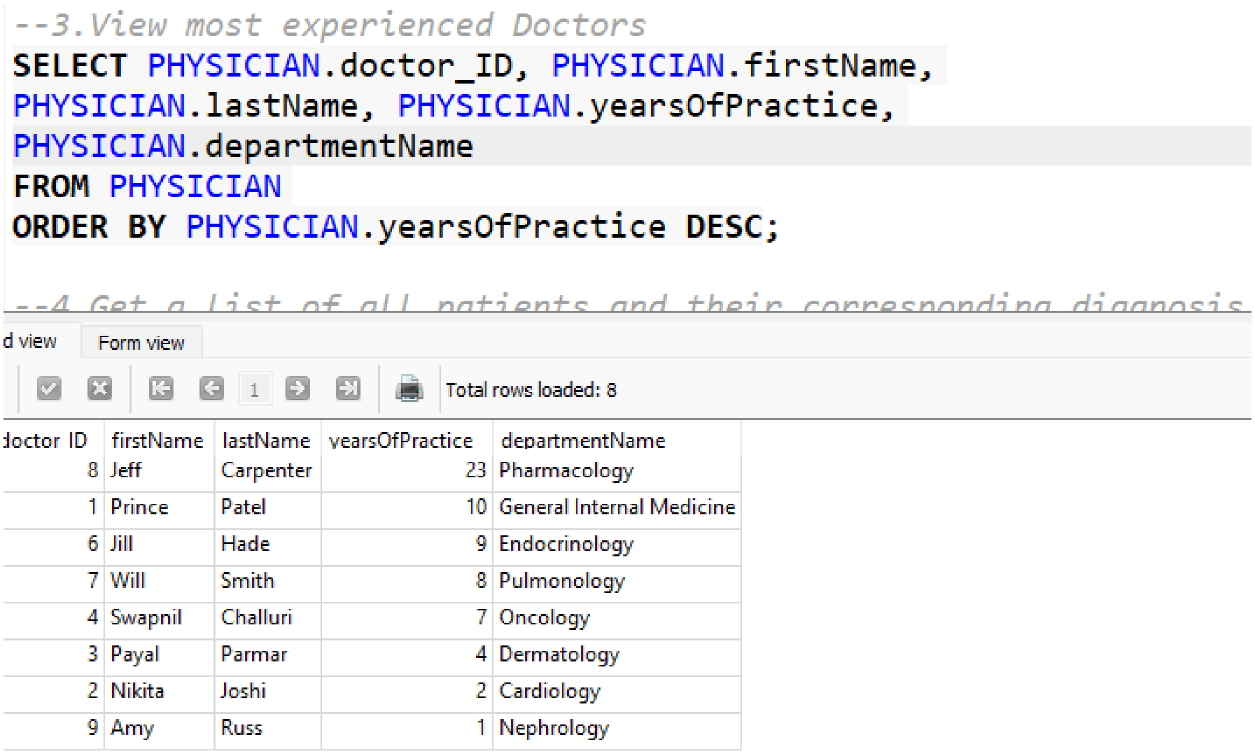


```
--2.Insurance Coverage according to PatientID
SELECT BILLING.patientID, PATIENT.firstName,
PATIENT.lastName, BILLING.insuranceCoveragePercentage
FROM PATIENT INNER JOIN BILLING ON PATIENT.patientID =
BILLING.patientID
ORDER BY BILLING.insuranceCoveragePercentage DESC;

--3 View most experienced Doctors
```

Total rows loaded: 8

	patientID	firstName	lastName	insuranceCoveragePercentage
1	7	Tom	Holland	100
2	1	Tommy	Hillfigure	80
3	2	Dwayne	Jonson	60
4	6	Robert	Downey	55
5	9	Post	Malone	45
6	3	Hugh	Jackson	40
7	8	Billie	Eilish	15
8	5	Chris	Evans	0

--3.View most experienced Doctors
SELECT PHYSICIAN.doctor_ID, PHYSICIAN.firstName,
PHYSICIAN.lastName, PHYSICIAN.yearsOfPractice,
PHYSICIAN.departmentName
FROM PHYSICIAN
ORDER BY PHYSICIAN.yearsOfPractice DESC;

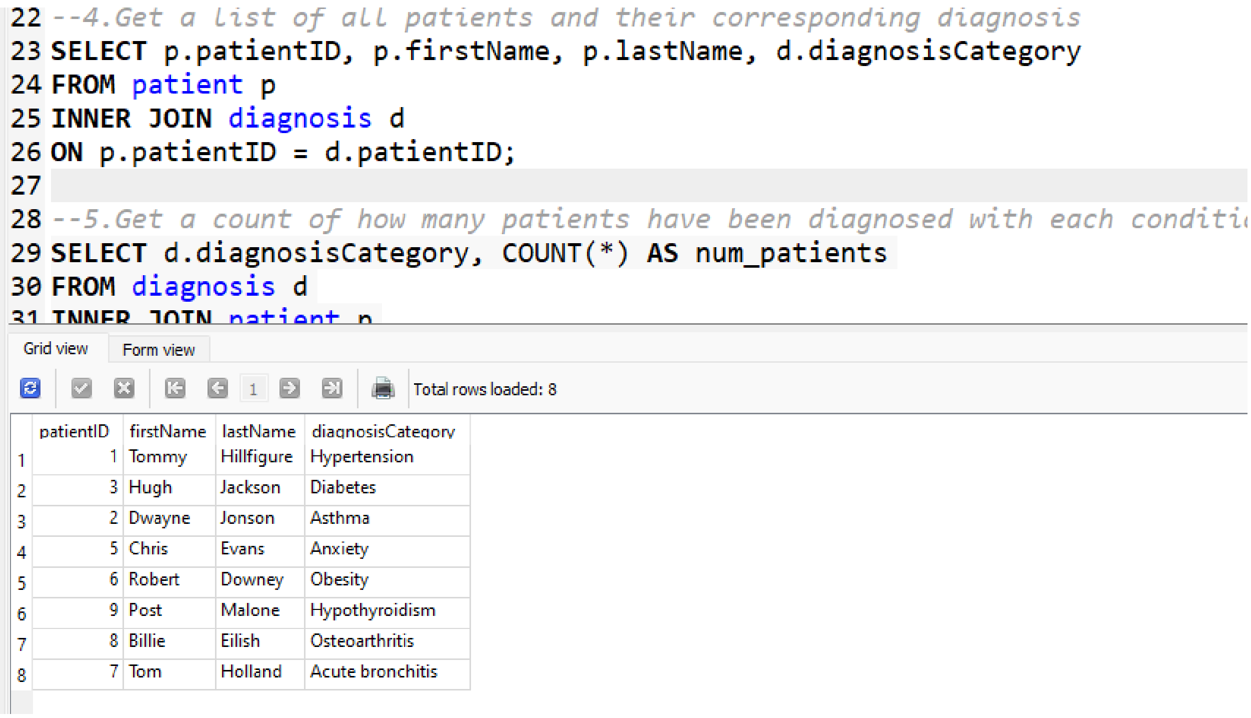


```
--3.View most experienced Doctors
SELECT PHYSICIAN.doctor_ID, PHYSICIAN.firstName,
PHYSICIAN.lastName, PHYSICIAN.yearsOfPractice,
PHYSICIAN.departmentName
FROM PHYSICIAN
ORDER BY PHYSICIAN.yearsOfPractice DESC;

--4 Get a list of all patients and their corresponding diagnosis
```

Total rows loaded: 8

doctor ID	firstName	lastName	yearsOfPractice	departmentName
8	Jeff	Carpenter	23	Pharmacology
1	Prince	Patel	10	General Internal Medicine
6	Jill	Hade	9	Endocrinology
7	Will	Smith	8	Pulmonology
4	Swapnil	Challuri	7	Oncology
3	Payal	Parmar	4	Dermatology
2	Nikita	Joshi	2	Cardiology
9	Amy	Russ	1	Nephrology

```
--4.Get a list of all patients and their corresponding diagnosis
SELECT p.patientID, p.firstName, p.lastName, d.diagnosisCategory
FROM patient p
INNER JOIN diagnosis d
ON p.patientID = d.patientID;
```



```
--4.Get a list of all patients and their corresponding diagnosis
SELECT p.patientID, p.firstName, p.lastName, d.diagnosisCategory
FROM patient p
INNER JOIN diagnosis d
ON p.patientID = d.patientID;

--5.Get a count of how many patients have been diagnosed with each conditi
SELECT d.diagnosisCategory, COUNT(*) AS num_patients
FROM diagnosis d
INNER JOIN patient p
```

Total rows loaded: 8

	patientID	firstName	lastName	diagnosisCategory
1	1	Tommy	Hillfigure	Hypertension
2	3	Hugh	Jackson	Diabetes
3	2	Dwayne	Jonson	Asthma
4	5	Chris	Evans	Anxiety
5	6	Robert	Downey	Obesity
6	9	Post	Malone	Hypothyroidism
7	8	Billie	Eilish	Osteoarthritis
8	7	Tom	Holland	Acute bronchitis

--5.Get a count of how many patients have been diagnosed with each condition:

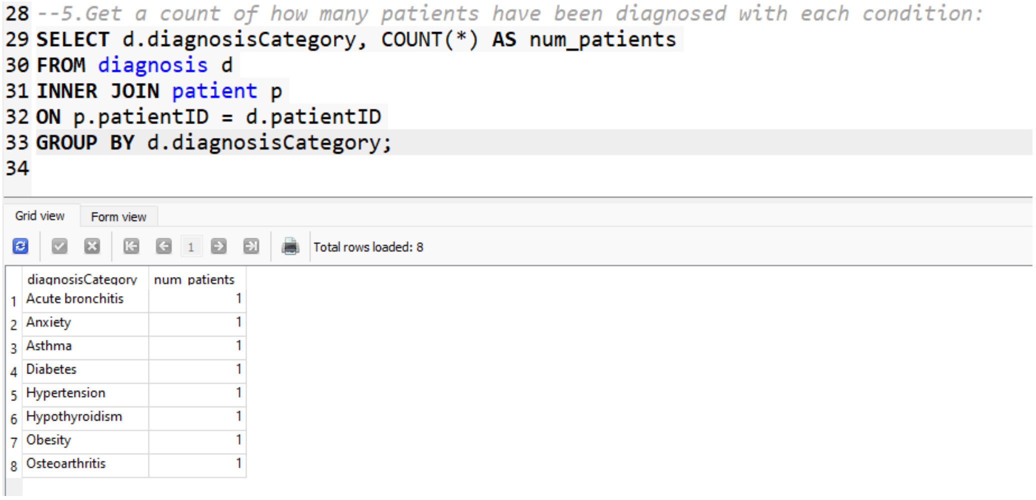
SELECT d.diagnosisCategory, COUNT(*) AS num_patients

FROM diagnosis d

INNER JOIN patient p

ON p.patientID = d.patientID

GROUP BY d.diagnosisCategory;



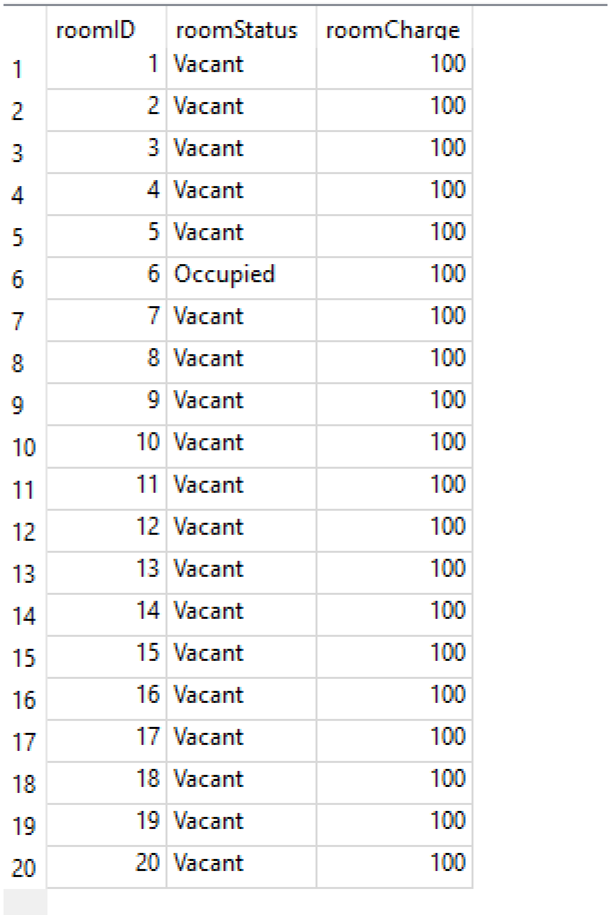
```
--5.Get a count of how many patients have been diagnosed with each condition:
SELECT d.diagnosisCategory, COUNT(*) AS num_patients
FROM diagnosis d
INNER JOIN patient p
ON p.patientID = d.patientID
GROUP BY d.diagnosisCategory;
```

Total rows loaded: 8

	diagnosisCategory	num_patients
1	Acute bronchitis	1
2	Anxiety	1
3	Asthma	1
4	Diabetes	1
5	Hypertension	1
6	Hypothyroidism	1
7	Obesity	1
8	Osteoarthritis	1

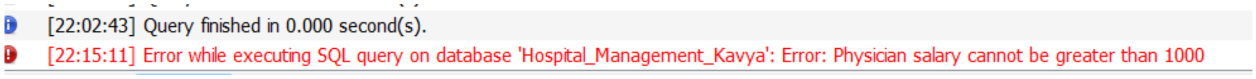
3. TRIGGER STATEMENTS and TESTING

```
--1.trigger function that updates the room status from "vacant" to "occupied"
when a new patient is admitted to the room
CREATE TRIGGER update_room_status12
AFTER INSERT ON INPATIENT
BEGIN
UPDATE ROOM
SET roomStatus = 'Occupied'
WHERE roomID = NEW.roomID;
END;

INSERT INTO INPATIENT VALUES(6,'2023-03-09','',6);
```



	roomID	roomStatus	roomCharge
1	1	Vacant	100
2	2	Vacant	100
3	3	Vacant	100
4	4	Vacant	100
5	5	Vacant	100
6	6	Occupied	100
7	7	Vacant	100
8	8	Vacant	100
9	9	Vacant	100
10	10	Vacant	100
11	11	Vacant	100
12	12	Vacant	100
13	13	Vacant	100
14	14	Vacant	100
15	15	Vacant	100
16	16	Vacant	100
17	17	Vacant	100
18	18	Vacant	100
19	19	Vacant	100
20	20	Vacant	100

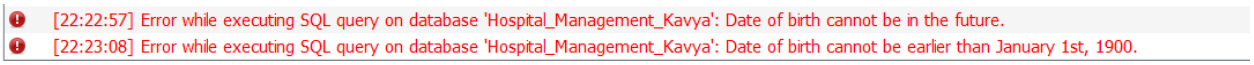
```
INSERT INTO INPATIENT VALUES(6,'2023-03-09','',6);

--2.trigger function to check the physician salary is not greater than 1000
CREATE TRIGGER check_physician_salary
BEFORE INSERT ON physician
FOR EACH ROW
WHEN NEW.salary > 1000
BEGIN
SELECT RAISE(ABORT, 'Error: Physician salary cannot be greater than 1000');
END;

INSERT INTO PHYSICIAN VALUES(10,'Usha', 'Patel','1432154321','Nephrology','999999998',5,'1000',2000);
```



```
[22:02:43] Query finished in 0.000 second(s).
[22:15:11] Error while executing SQL query on database 'Hospital_Management_Kavya': Error: Physician salary cannot be greater than 1000
```

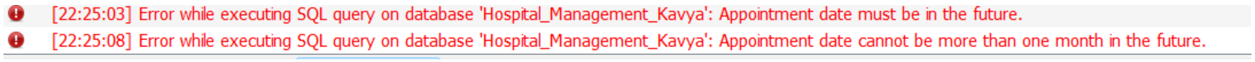
```
--Create Trigger function for checking DOB constraint
CREATE TRIGGER check_date_of_birth
BEFORE INSERT ON PATIENT
FOR EACH ROW
BEGIN
SELECT
CASE
WHEN NEW.DOB > date('now')
THEN RAISE(ABORT, 'Date of birth cannot be in the future.')
WHEN NEW.DOB < '1900-01-01'
THEN RAISE(ABORT, 'Date of birth cannot be earlier than January 1st, 1900.')
ELSE NULL
END;
END;
```

```
INSERT INTO PATIENT VALUES(1,'Jess',
'Hillfigure','2023-03-15',26,'M',123,'Cary
Road','Manlius','NY',13104,'3153453651',1,'O',172,180,'B+');
INSERT INTO PATIENT VALUES(2,'Nick',
'Jonson','1889-11-01',33,'M',234,'Bridge
Avenue','Manlius','NY',13104,'3154256157',3,'I',150,164,'O+');
```



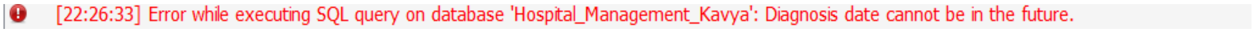
```
[22:22:57] Error while executing SQL query on database 'Hospital_Management_Kavya': Date of birth cannot be in the future.
[22:23:08] Error while executing SQL query on database 'Hospital_Management_Kavya': Date of birth cannot be earlier than January 1st, 1900.
```

```
--Create Trigger function for Appointment Date constraint
CREATE TRIGGER check_appointment_date
BEFORE INSERT ON APPOINTMENT
FOR EACH ROW
BEGIN
SELECT
CASE
WHEN NEW.appointmentDate <= date('now')
THEN RAISE(ABORT, 'Appointment date must be in the future.')
WHEN NEW.appointmentDate > date('now', '+1 month')
THEN RAISE(ABORT, 'Appointment date cannot be more than one month in
the future.')
ELSE NULL
END;
END;

INSERT INTO APPOINTMENT VALUES(100000011,8,7,4,'2022-03-23');
INSERT INTO APPOINTMENT VALUES(100000012,9,9,1,'2023-04-25');
```



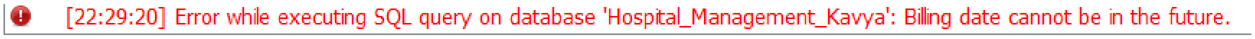
```
[22:25:03] Error while executing SQL query on database 'Hospital_Management_Kavya': Appointment date must be in the future.
[22:25:08] Error while executing SQL query on database 'Hospital_Management_Kavya': Appointment date cannot be more than one month in the future.
```

```
--Create Trigger for Diagnosis Date constraint
CREATE TRIGGER check_diagnosis_date
BEFORE INSERT ON DIAGNOSIS
FOR EACH ROW
BEGIN
SELECT
CASE
WHEN NEW.diagnosisDate > date('now')
THEN RAISE(ABORT, 'Diagnosis date cannot be in the future.')
ELSE NULL
END;
END;

INSERT INTO DIAGNOSIS VALUES(100000039,3,2,'Asthma','Albuterol for a month','2023-10-04');
```



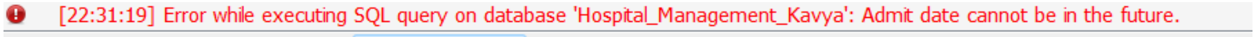
```
[22:26:33] Error while executing SQL query on database 'Hospital_Management_Kavya': Diagnosis date cannot be in the future.
```

```
--Create Trigger for Billing Date Constraint
CREATE TRIGGER check_billing_date
BEFORE INSERT ON BILLING
FOR EACH ROW
BEGIN
SELECT
CASE
WHEN NEW.billingDate > date('now')
THEN RAISE(ABORT, 'Billing date cannot be in the future.')
ELSE NULL
END;
END;

INSERT INTO BILLING VALUES(100000567,9,550,100,0,45,'2023-11-10');
```



```
[22:29:20] Error while executing SQL query on database 'Hospital_Management_Kavya': Billing date cannot be in the future.
```

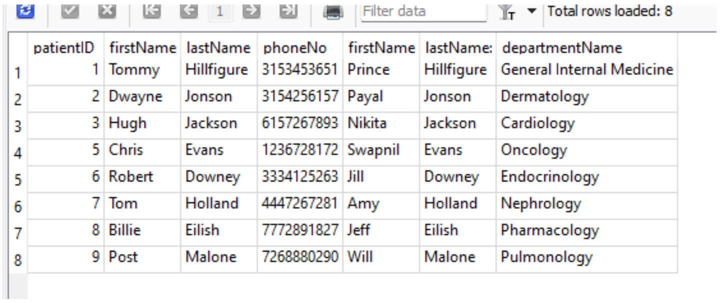
```
--Create Trigger for AdmitDate and Discharge Date
CREATE TRIGGER check_AdmitDate_date
BEFORE INSERT ON INPATIENT
FOR EACH ROW
BEGIN
SELECT
CASE
WHEN NEW.admitDate > date('now')
THEN RAISE(ABORT, 'Admit date cannot be in the future.')
ELSE NULL
END;
END;

INSERT INTO INPATIENT VALUES(5,'2023-11-09',' ',5);
```



```
[22:31:19] Error while executing SQL query on database 'Hospital_Management_Kavya': Admit date cannot be in the future.
```

4. VIEW STATEMENTS

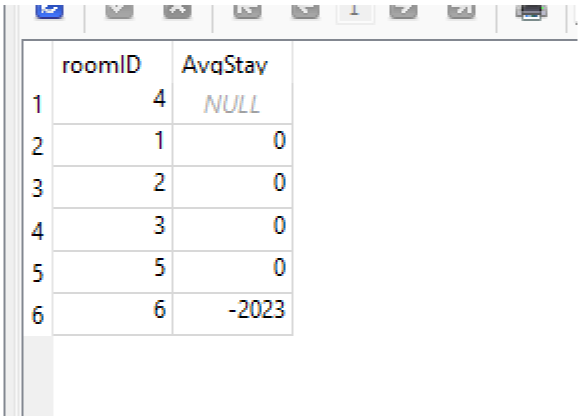
```
--View to show all patient information along with their assigned doctors
CREATE VIEW PatientDoctorView AS
SELECT p.patientID, p.firstName, p.lastName, p.phoneNo,
d.firstName,p.lastName, d.departmentName
FROM Patient p
LEFT JOIN PHYSICIAN d ON p.doctor_id = d.doctor_id;
```



Total rows loaded: 8

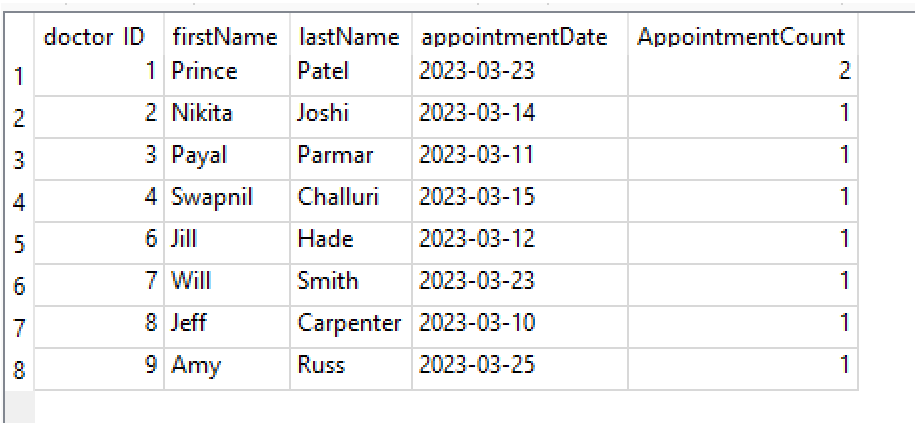
	patientID	firstName	lastName	phoneNo	firstName	lastName	departmentName
1	1	Tommy	Hillfigure	3153453651	Prince	Hillfigure	General Internal Medicine
2	2	Dwayne	Jonson	3154256157	Payal	Jonson	Dermatology
3	3	Hugh	Jackson	6157267893	Nikita	Jackson	Cardiology
4	5	Chris	Evans	1236728172	Swapnil	Evans	Oncology
5	6	Robert	Downey	3334125263	Jill	Downey	Endocrinology
6	7	Tom	Holland	4447267281	Amy	Holland	Nephrology
7	8	Billie	Eilish	7772891827	Jeff	Eilish	Pharmacology
8	9	Post	Malone	7268880290	Will	Malone	Pulmonology

```
--View to show the average length of stay for each ward
CREATE VIEW RoomAvgStay AS
SELECT r.roomID, AVG((a.dischargeDate) - (a.admitDate)) AS AvgStay
FROM ROOM r
LEFT JOIN INPATIENT a ON r.roomID = a.roomID
GROUP BY a.inpatient_ID;
```



	roomID	AvgStay
1	4	NULL
2	1	0
3	2	0
4	3	0
5	5	0
6	6	-2023

```
--View to show the number of appointments per doctor per day
CREATE VIEW DoctorAppointmentCount AS
SELECT d.doctor_ID, d.firstName, d.lastName, a.appointmentDate,
COUNT(*) AS AppointmentCount
FROM PHYSICIAN d
LEFT JOIN Appointment a ON d.doctor_ID = a.doctor_ID
GROUP BY d.doctor_ID, a.appointmentDate;

INSERT INTO APPOINTMENT VALUES(1000000010,1,1,9,'2023-03-23');
```

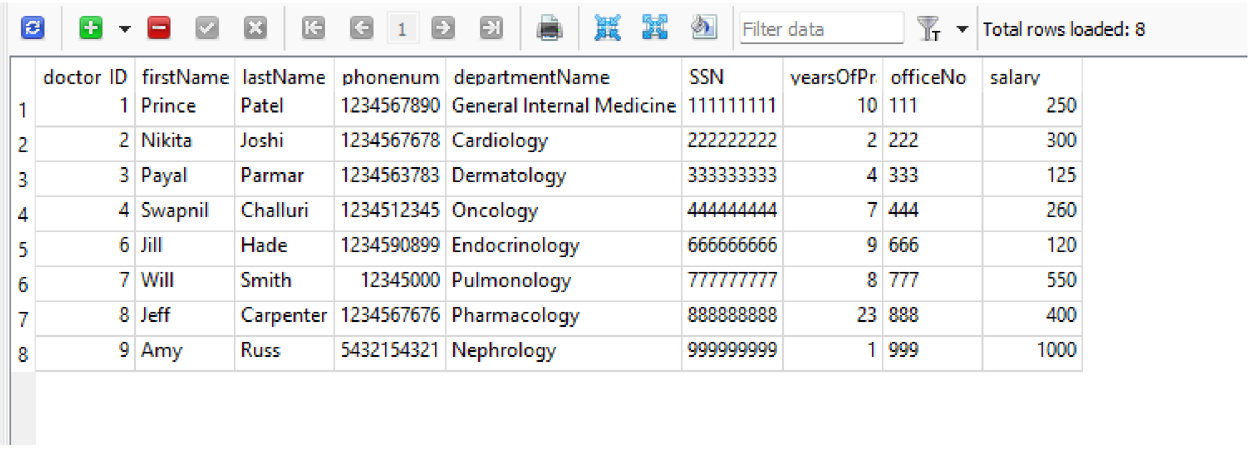


	doctor ID	firstName	lastName	appointmentDate	AppointmentCount
1	1	Prince	Patel	2023-03-23	2
2	2	Nikita	Joshi	2023-03-14	1
3	3	Payal	Parmar	2023-03-11	1
4	4	Swapnil	Challuri	2023-03-15	1
5	6	Jill	Hade	2023-03-12	1
6	7	Will	Smith	2023-03-23	1
7	8	Jeff	Carpenter	2023-03-10	1
8	9	Amy	Russ	2023-03-25	1

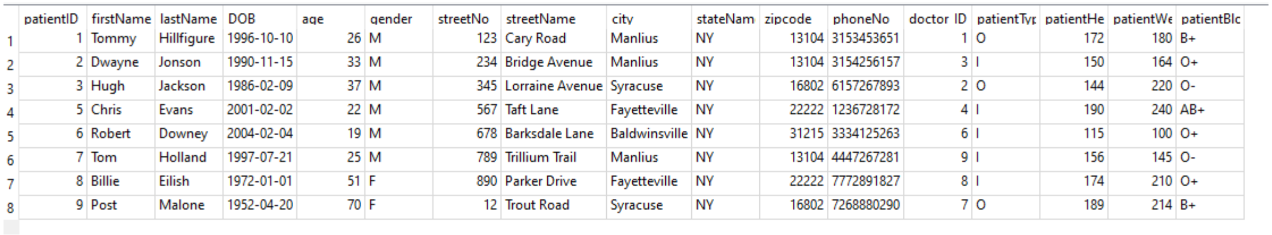
VI. UNIT TESTS

1. Checking ON DELETE CASCADE FOR PHYSICIAN TABLE

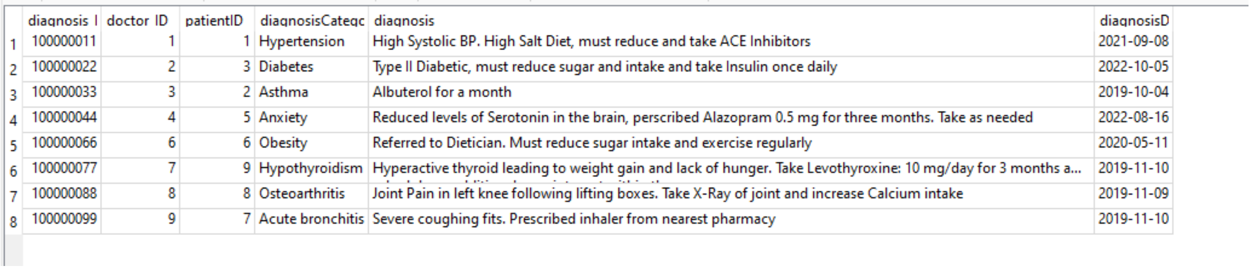
BEFORE:



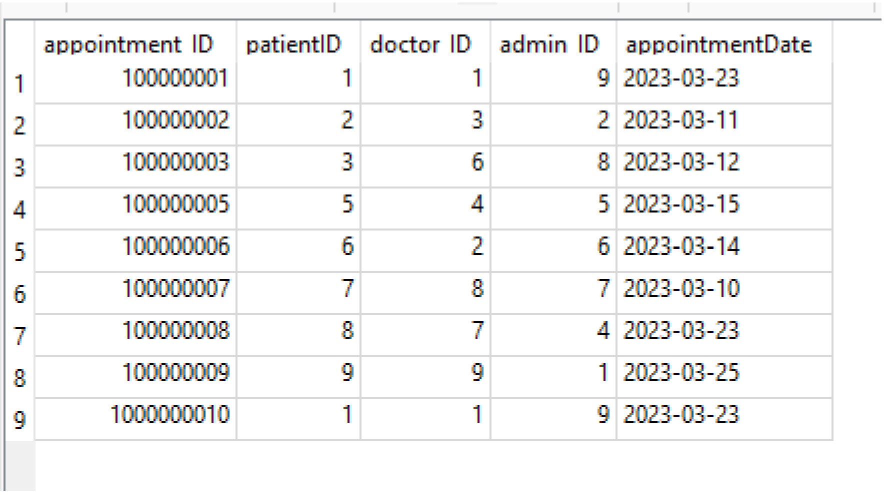
	doctor ID	firstName	lastName	phonenum	departmentName	SSN	yearsOfPr	officeNo	salary
1	1	Prince	Patel	1234567890	General Internal Medicine	111111111	10	111	250
2	2	Nikita	Joshi	1234567678	Cardiology	222222222	2	222	300
3	3	Payal	Parmar	1234563783	Dermatology	333333333	4	333	125
4	4	Swapnil	Challuri	1234512345	Oncology	444444444	7	444	260
5	6	Jill	Hade	1234590899	Endocrinology	666666666	9	666	120
6	7	Will	Smith	12345000	Pulmonology	777777777	8	777	550
7	8	Jeff	Carpenter	1234567676	Pharmacology	888888888	23	888	400
8	9	Amy	Russ	5432154321	Nephrology	999999999	1	999	1000



	patientID	firstName	lastName	DOB	age	gender	streetNo	streetName	city	stateNam	zipcode	phoneNo	doctor ID	patientTyp	patientHe	patientWe	patientBlo
1	1	Tommy	Hillfigure	1996-10-10	26	M	123	Cary Road	Manlius	NY	13104	3153453651	1	O	172	180	B+
2	2	Dwayne	Jonson	1990-11-15	33	M	234	Bridge Avenue	Manlius	NY	13104	3154256157	3	I	150	164	O+
3	3	Hugh	Jackson	1986-02-09	37	M	345	Lorraine Avenue	Syracuse	NY	16802	6157267893	2	O	144	220	O-
4	5	Chris	Evans	2001-02-02	22	M	567	Taft Lane	Fayetteville	NY	22222	1236728172	4	I	190	240	AB+
5	6	Robert	Downey	2004-02-04	19	M	678	Barksdale Lane	Baldwinsville	NY	31215	3334125263	6	I	115	100	O+
6	7	Tom	Holland	1997-07-21	25	M	789	Trillium Trail	Manlius	NY	13104	4447267281	9	I	156	145	O-
7	8	Billie	Eilish	1972-01-01	51	F	890	Parker Drive	Fayetteville	NY	22222	7772891827	8	I	174	210	O+
8	9	Post	Malone	1952-04-20	70	F	12	Trout Road	Syracuse	NY	16802	7268880290	7	O	189	214	B+

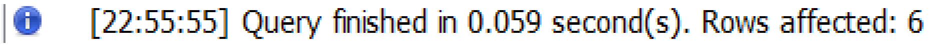


	diagnosis I	doctor ID	patientID	diagnosisCategc	diagnosis	diagnosisD
1	100000011	1	1	Hypertension	High Systolic BP. High Salt Diet, must reduce and take ACE Inhibitors	2021-09-08
2	100000022	2	3	Diabetes	Type II Diabetic, must reduce sugar and intake and take Insulin once daily	2022-10-05
3	100000033	3	2	Asthma	Albuterol for a month	2019-10-04
4	100000044	4	5	Anxiety	Reduced levels of Serotonin in the brain, perscribed Alazopram 0.5 mg for three months. Take as needed	2022-08-16
5	100000066	6	6	Obesity	Referred to Dietician. Must reduce sugar intake and exercise regularly	2020-05-11
6	100000077	7	9	Hypothyroidism	Hyperactive thyroid leading to weight gain and lack of hunger. Take Levothyroxine: 10 mg/day for 3 months a...	2019-11-10
7	100000088	8	8	Osteoarthritis	Joint Pain in left knee following lifting boxes. Take X-Ray of joint and increase Calcium intake	2019-11-09
8	100000099	9	7	Acute bronchitis	Severe coughing fits. Prescribed inhaler from nearest pharmacy	2019-11-10

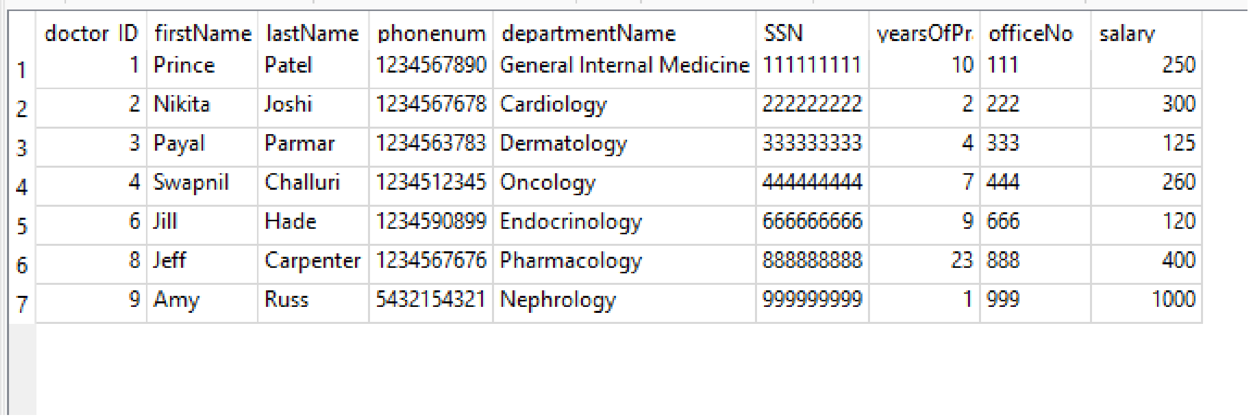


	appointment ID	patientID	doctor ID	admin ID	appointmentDate
1	100000001	1	1	9	2023-03-23
2	100000002	2	3	2	2023-03-11
3	100000003	3	6	8	2023-03-12
4	100000005	5	4	5	2023-03-15
5	100000006	6	2	6	2023-03-14
6	100000007	7	8	7	2023-03-10
7	100000008	8	7	4	2023-03-23
8	100000009	9	9	1	2023-03-25
9	1000000010	1	1	9	2023-03-23

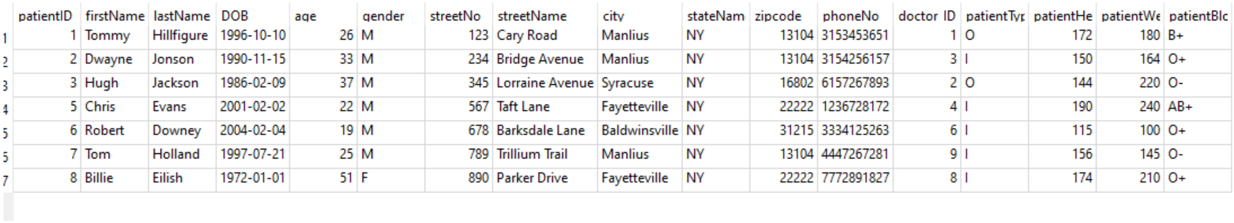
After:
DELETE FROM PHYSICIAN WHERE doctor_ID = 7;



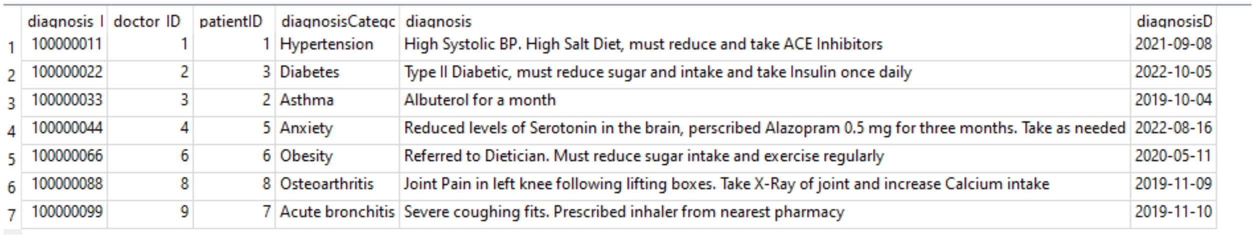
```
[22:55:55] Query finished in 0.059 second(s). Rows affected: 6
```



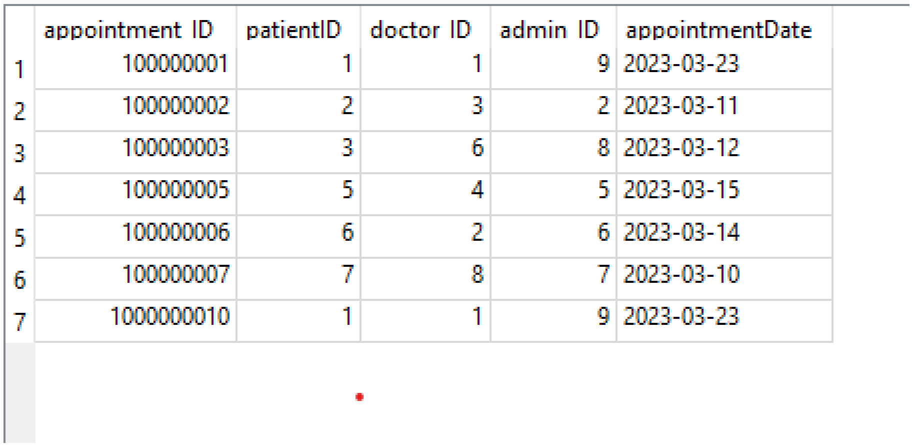
	doctor ID	firstName	lastName	phonenum	departmentName	SSN	yearsOfPr	officeNo	salary
1	1	Prince	Patel	1234567890	General Internal Medicine	111111111	10	111	250
2	2	Nikita	Joshi	1234567678	Cardiology	222222222	2	222	300
3	3	Payal	Parmar	1234563783	Dermatology	333333333	4	333	125
4	4	Swapnil	Challuri	1234512345	Oncology	444444444	7	444	260
5	6	Jill	Hade	1234590899	Endocrinology	666666666	9	666	120
6	8	Jeff	Carpenter	1234567676	Pharmacology	888888888	23	888	400
7	9	Amy	Russ	5432154321	Nephrology	999999999	1	999	1000



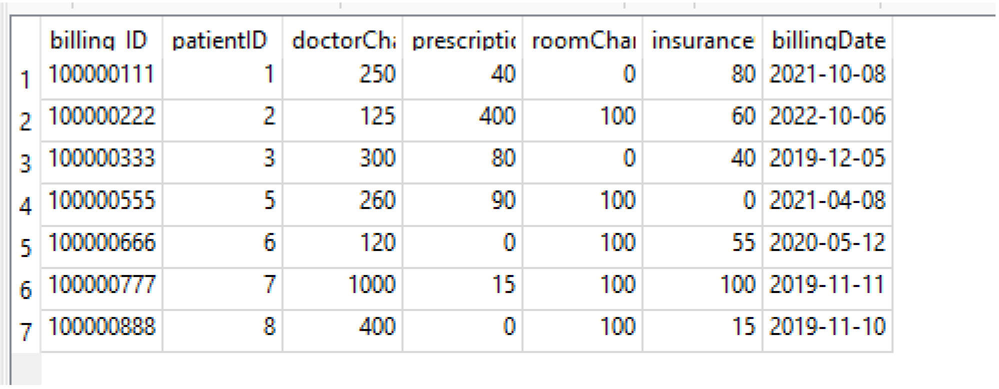
	patientID	firstName	lastName	DOB	age	gender	streetNo	streetName	city	stateNam	zipcode	phoneNo	doctor ID	patientTyp	patientHe	patientWe	patientBlc
1	1	Tommy	Hillfigure	1996-10-10	26	M	123	Cary Road	Manlius	NY	13104	3153453651	1	O	172	180	B+
2	2	Dwayne	Jonson	1990-11-15	33	M	234	Bridge Avenue	Manlius	NY	13104	3154256157	3	I	150	164	O+
3	3	Hugh	Jackson	1986-02-09	37	M	345	Lorraine Avenue	Syracuse	NY	16802	6157267893	2	O	144	220	O-
4	5	Chris	Evans	2001-02-02	22	M	567	Taft Lane	Fayetteville	NY	22222	1236728172	4	I	190	240	AB+
5	6	Robert	Downey	2004-02-04	19	M	678	Barksdale Lane	Baldwinsville	NY	31215	3334125263	6	I	115	100	O+
6	7	Tom	Holland	1997-07-21	25	M	789	Trillium Trail	Manlius	NY	13104	4447267281	9	I	156	145	O-
7	8	Billie	Eilish	1972-01-01	51	F	890	Parker Drive	Fayetteville	NY	22222	7772891827	8	I	174	210	O+



	diagnosis I	doctor ID	patientID	diagnosisCategc	diagnosis	diagnosisD
1	100000011	1	1	Hypertension	High Systolic BP. High Salt Diet, must reduce and take ACE Inhibitors	2021-09-08
2	100000022	2	3	Diabetes	Type II Diabetic, must reduce sugar and intake and take Insulin once daily	2022-10-05
3	100000033	3	2	Asthma	Albuterol for a month	2019-10-04
4	100000044	4	5	Anxiety	Reduced levels of Serotonin in the brain, perscribed Alazopram 0.5 mg for three months. Take as needed	2022-08-16
5	100000066	6	6	Obesity	Referred to Dietician. Must reduce sugar intake and exercise regularly	2020-05-11
6	100000088	8	8	Osteoarthritis	Joint Pain in left knee following lifting boxes. Take X-Ray of joint and increase Calcium intake	2019-11-09
7	100000099	9	7	Acute bronchitis	Severe coughing fits. Prescribed inhaler from nearest pharmacy	2019-11-10



	appointment ID	patientID	doctor ID	admin ID	appointmentDate
1	100000001	1	1	9	2023-03-23
2	100000002	2	3	2	2023-03-11
3	100000003	3	6	8	2023-03-12
4	100000005	5	4	5	2023-03-15
5	100000006	6	2	6	2023-03-14
6	100000007	7	8	7	2023-03-10
7	1000000010	1	1	9	2023-03-23

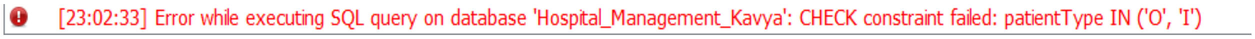


	billing ID	patientID	doctorChi	prescriptic	roomChai	insurance	billingDate
1	100000111	1	250	40	0	80	2021-10-08
2	100000222	2	125	400	100	60	2022-10-06
3	100000333	3	300	80	0	40	2019-12-05
4	100000555	5	260	90	100	0	2021-04-08
5	100000666	6	120	0	100	55	2020-05-12
6	100000777	7	1000	15	100	100	2019-11-11
7	100000888	8	400	0	100	15	2019-11-10

2. Testing CHECK constraints

```
patientType CHAR(1) NOT NULL CHECK(patientType IN ('O','I')),
```

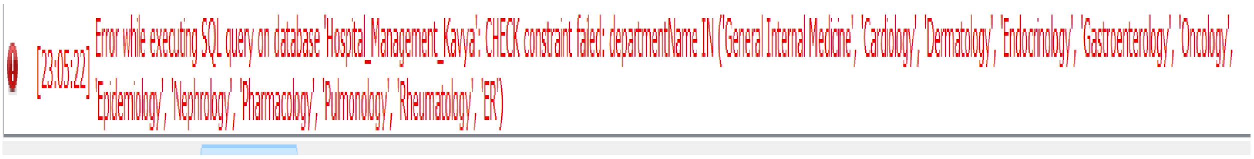
```
INSERT INTO PATIENT VALUES(15,'Terry',
'Mikes','1952-04-20',70,'F',012,'Trout
Road','Syracuse','NY',16802,'7268880290',7,'K',189,214,'B+');
```



```
[23:02:33] Error while executing SQL query on database 'Hospital_Management_Kavya': CHECK constraint failed: patientType IN ('O', 'I')
```

```
departmentName VARCHAR(100) NOT NULL CHECK( departmentName
IN ('General Internal Medicine', 'Cardiology', 'Dermatology',
'Endocrinology', 'Gastroenterology', 'Oncology', 'Epidemiology',
'Nephrology', 'Pharmacology', 'Pulmonology', 'Rheumatology', 'ER')),
```

```
INSERT INTO PHYSICIAN VALUES(15,'Shikha ',
'Panchal','1234567678','ABC','222222222',2,'222',300);
```



```
[23:05:22] Error while executing SQL query on database 'Hospital_Management_Kavya': CHECK constraint failed: departmentName IN ('General Internal Medicine', 'Cardiology', 'Dermatology', 'Endocrinology', 'Gastroenterology', 'Oncology',
'Epidemiology', 'Nephrology', 'Pharmacology', 'Pulmonology', 'Rheumatology', 'ER')
```

VII. CONCLUSION

In conclusion, the Hospital DBMS project developed in SQLite has proven to be a useful and efficient tool for managing the vast amount of data generated in a hospital setting. The project has demonstrated the ability to store, organize, and retrieve information related to patients, doctors, appointments, treatments, and medications.

Overall, the project has met its objectives of improving the efficiency, accuracy, and accessibility of hospital data management. Future enhancements to the system may include integration with other hospital systems, such as electronic health records, to facilitate seamless data exchange and improve patient care.