Hospital Database Management System

Ryan Ondocin

Jimit Mistry

Table of Contents

1. Project Summary

Hospitals are a fundamental institution in the world that are necessary for a healthy and productive society. They provide services that can accomodate for every physiological problem known to man which is why a ton of information is necessary for proper treatment. Though numerous parties are involved in this process, doctors have always played an integral role in assessing patients, prescribing medications, conducting surgeries and monitoring patient health. Hospitals are estimated to employ approximately 5.03 million people across the US which shows that the process digs much deeper than just doctor/patient care.[1] Registered Nurses constitute ~ 27% of the industry while the wheels are kept in motion by medical assistants, administrative staff, therapists, technicians and many more positions that make for an authentic healthcare experience. These ideas alone can help explain why many hospitals have thousands of employees on their payroll. Due to its location and prominence, our team will be using Upstate University Medical Hospital as a model for outlining business rules and creating an RDBMS. Upstate currently has 9,849 employees, spanning a wide variety of roles which is why our business should be treated as an enterprise.[2]

Healthcare is a field that demands efficiency when it comes to processing vast amounts of medical data and effectively treating patients. Within the last twenty years or so organizations have had to face the daunting task of transferring medical data from file processing systems into a robust relational database that can provide all parties with access to quick and reliable medical/financial information. HIPAA Privacy is a national standard that protects the dissemination of patient information between medical personnel which means that the data hospitals deal with is highly sensitive and demands file security/authorized access. This fact alone is enough to justify why file processing systems are antiquated in the context of modern-day healthcare. While many organizations have made the switch, this transition has posed numerous problems to some which result in setbacks for medical practitioners, hospital users and other staff.

In the context of healthcare, workflow congestion could literally translate to life or death for patients which means there is no room for inefficiencies when establishing an RDMS. Data management is crucial for assessing patient info, arranging patient/doctor schedules and accounting for the financial needs of the involved parties. The best method for attacking this issue is to establish a database management system that can adequately deal with problems relating to file security and information retrieval/updates.

---

[1] "Hospitals." *Data USA*, https://datausa.io/profile/naics/hospitals.

[2] "SUNY Upstate Facts at a Glance: About Upstate: SUNY Upstate Medical University." *About Upstate | SUNY Upstate Medical University*, http://www.upstate.edu/about/glance.php.

Our proposed solution is to create a relational database that efficiently, securely and succinctly processes medical information. The system would be able to digitally store patient medical records, appointment schedules, medical staff directories, room assignments and billing information. This approach would be less time consuming and more secure than a file processing system which affords the user a much better experience. Automating the process would also improve healthcare management and staff/patient interactions as greater precision would be conducted in regards to planning. The patient would be able to plan their appointments, view their billing statements and access their medical records on one secure platform *ahead* of time. The beauty in this approach is that by minimizing logistical congestion in the hospital, you can optimize the time spent being treated and cared for by medical personnel. Our system would also allow doctors to quickly retrieve/update medical notes which can ensure that the best diagnostic practices are being put to use.

That being said, our proposal is not nearly robust enough to account for all of the functions that occur in a fully operational medical unit. A few considerations we have chosen to omit are things such as hospital inventory and payroll management to name a few. Similarly, we will never be able to fully encompass all of the intricacies associated with insurance claim processing or financial assistance programs designed for accommodating certain patients due to their intrinsic complexities. We will, however, create a system that can fundamentally deal with the central processes that are vital for clinical administrations. Additionally, we will avoid dealing with unstructured data such as CT/PET-Scans/X-Rays/MRIs generated from lab results because of their difficulty in obtaining and interpreting on a high-level. In order to omit unstructured data from our relational database we will solely be dealing with medical records/diagnostics in a textual form.

Our system will be composed of doctors, patients, diagnoses, billing information, inpatient room planning and administrators. Admin will be able to administer billing statements, assign inpatients to rooms based on availability, confirm/modify/delete appointments and edit patient/staff details. Patients will be able to view their diagnosis/billing charges and schedule appointments. Doctors will be able to modify/view patient medical records, update their consultation fees and diagnose patients based on their symptoms. Given HIPAA considerations and the confidential nature of medical information, all users will simply not have equal access to the data. Patients will not have the ability to share their medical records with outside users nor will doctors. These safeguards will protect the management system from data leaks/breaches and ensure that the most ethical practices are utilized.

In order for our proposal to be helpful we are making the assumption that our ‘hospital’ is still using a file processing system. The data intake process is as follows. There is a medical form that is filled out ahead of time by the patient online. This form would capture the bare-essentials necessary for proper treatment to take place such as patient information and medical history(name, address, height, weight etc. etc..). Admin would be responsible for entering this information into a system that would update if the patient has visited a hospital before and be

created if it's a patient's first visit. Appointments are scheduled ahead of time, via a phone call by the patient. The administrator will determine when an appointment can take place based on the doctor's availability(offline). After an appointment has occurred the doctor provides a proper diagnosis to the patient and he/she enters this diagnosis into the system effectively updating the patient's medical records. Following the visit, the service charges will be totaled and issued to the patient one day after they have left the hospital. If the patient has been admitted then room availability is checked by an administrator and the patient is assigned a room. After the patient has been discharged their 'type' is changed from inpatient to outpatient by an administrator. The bill is administered the day after a patient has been discharged.

A few business rules used to guide our project are as follows:

Business Rules

1. Each doctor can have a different service/consultation fee.
2. Each inpatient can be assigned to only one room, but each room may have up to two patients.
3. Bills are issued the day after an appointment has occurred.
4. Patients who have been admitted are issued their bill the day after they have been discharged from the hospital.
5. On any particular day a doctor can see a maximum of ten patients.
6. Patients visiting the hospital for a routine checkup are considered to be outpatients.
7. Patients can schedule a maximum of three appointments per week.
8. We are assuming that each department can have only one type of doctor(cardiologist in cardiology department, nephrologist in nephrology department).
9. Hospital rooms charge a nightly fee for inpatients.

Entities and Attributes:

Administration Table:

Attribute Name	Field Type	Null/Not Null	Description
adminID - PK	INT	Not Null	Administrative Identification #
firstName	VARCHAR(20)	Not Null	
lastName	VARCHAR(30)	Not Null	
phoneNo	CHAR(10)	Not Null	
departmentName	VARCHAR(100)	Not Null	Medical department that administrator works.[3]
officeNo	VARCHAR(3)	Null	Where administrator works in hospital

Doctor Table:

Attribute Name	Field Type	Null/Not Null	Description
doctorID - PK	INT	Not Null	Doctor Identification #
firstName	VARCHAR(20)	Not Null	
lastName	VARCHAR(30)	Not Null	
departmentName	VARCHAR(100)	Not Null	Medical department that doctor works under
SSN	CHAR(9)	Not Null	Social Security Number of doctor
yearsOfPractice	NUMERIC	Null	How long the doctor has been practicing as a licensed physician
officeNo	VARCHAR(3)	Not Null	Office Number in where Physician is located
charge	NUMERIC	Not Null	charge ($USD) for each doctor's services

[3] https://www.upstate.edu/medicine/

Patient Table:

Attribute Name	Field Type	Null/Not Null	Description
patientID - PK	INT	Not Null	Patient ID#
firstName	VARCHAR(20)	Not Null	
lastName	VARCHAR(30)	Not Null	
DOB	DATE	Not Null	Date of birth (YYYY-MM-DD)
age	NUMERIC	Not Null	
patientWeight	INT	Null	Patient weight (lbs)
patientHeight	INT	Null	Patient height (cm)
gender	VARCHAR(2)	Not Null	Patient gender (M/F/NA)
streetNo	NUMERIC	Not Null	
streetName	VARCHAR(100)	Not Null	
city	VARCHAR(30)	Not Null	
stateName	CHAR(2)	Not Null	Two-letter abbreviations (NY/MA..etc.)
zip	NUMERIC(5)	Not Null	
phoneNo	CHAR(10)	Null	‘3255562716’
doctorID - FK to doctor	INT	Not Null	
patientType	CHAR(1)	Not Null	‘I’ = inpatient, ‘O’ = outpatient

Diagnosis Table

Attribute Name	Field Type	Null/Not Null	Description
diagnosisID - PK	INT	Not Null	Patient diagnosis#
doctorID - FK to doctor	INT	Not Null	Doctor responsible for diagnosis
patientID - FK to patient	INT	Not Null	Identification number of patient that was diagnosed
diagnosisDate	DATE	Not Null	Date diagnosis was made on patient: (YYYY-MM-DD)
diagnosis	TEXT	Null	Text Description of medical problem as well as recommendation moving forward
diagnosisCategory	VARCHAR(100)	Not Null	Description of medical ailments (latin name)

Inpatient Table:

Attribute Name	Field Type	Null/Not Null	Description
inpatientID - FK to patient	INT	Not Null	Unique identifier of the patient who has been admitted
admitDate	DATE	Not Null	Date patient has been admitted into the hospital (YYYY-MM-DD)
dischargeDate	DATE	Null	Date patient has been discharged from the hospital (YYYY-MM-DD)

Outpatient Table:

Attribute Name	Field Type	Null/Not Null	Description
outpatientID - FK to patient	INT	Not Null	Unique identifier of the patient who has been discharged
lastConsultDate	DATE	Null	Date of last consult: YYYY-MM-DD

Billing Table: Stores patient billing information

Attribute Name	Field Type	Null/Not Null	Descriptions
billingID - PK	INT	Not Null	
patientID - FK to patient	INT	Not Null	
doctorCharge	NUMERIC	Not Null	Total amount doctor has charged for services ($USD)
prescriptionCharge	NUMERIC	Not Null	Total charge of patient's prescription medications
roomCharge	NUMERIC	Not Null	Total charge of room following a patient's discharge from hospital
insuranceCoveragePercentage	NUMERIC(3)	Not Null	Percentage of total billing cost that patient's insurance covers
billingDate	DATE	Not Null	Date patients hospital bill was sent (day after appointment/discharge)

Room Table: Details about which rooms are available in the hospital as well as their fees

Attribute Name	Field Type	Null/Not Null	Description
roomID - PK	INT	Not Null	Room number where patient is located
roomStatus	VARCHAR(10)	Not Null	Status of room (Vacant/Occupied)
roomCharge	NUMERIC	Not Null	Cost of room ($USD) per night of patient stay

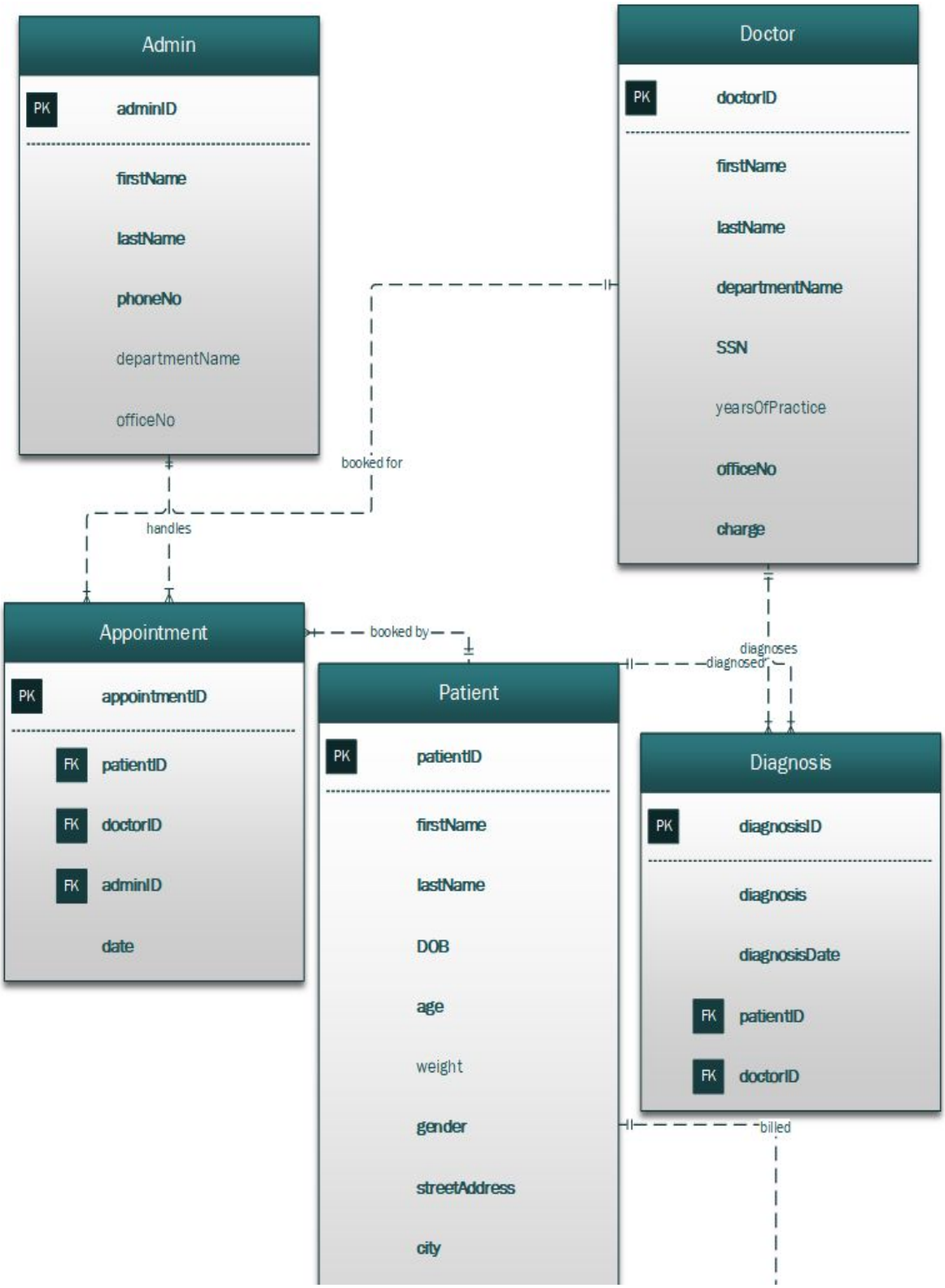
Room Inpatient Table: Breaks associative entity between Patient and Room Tables

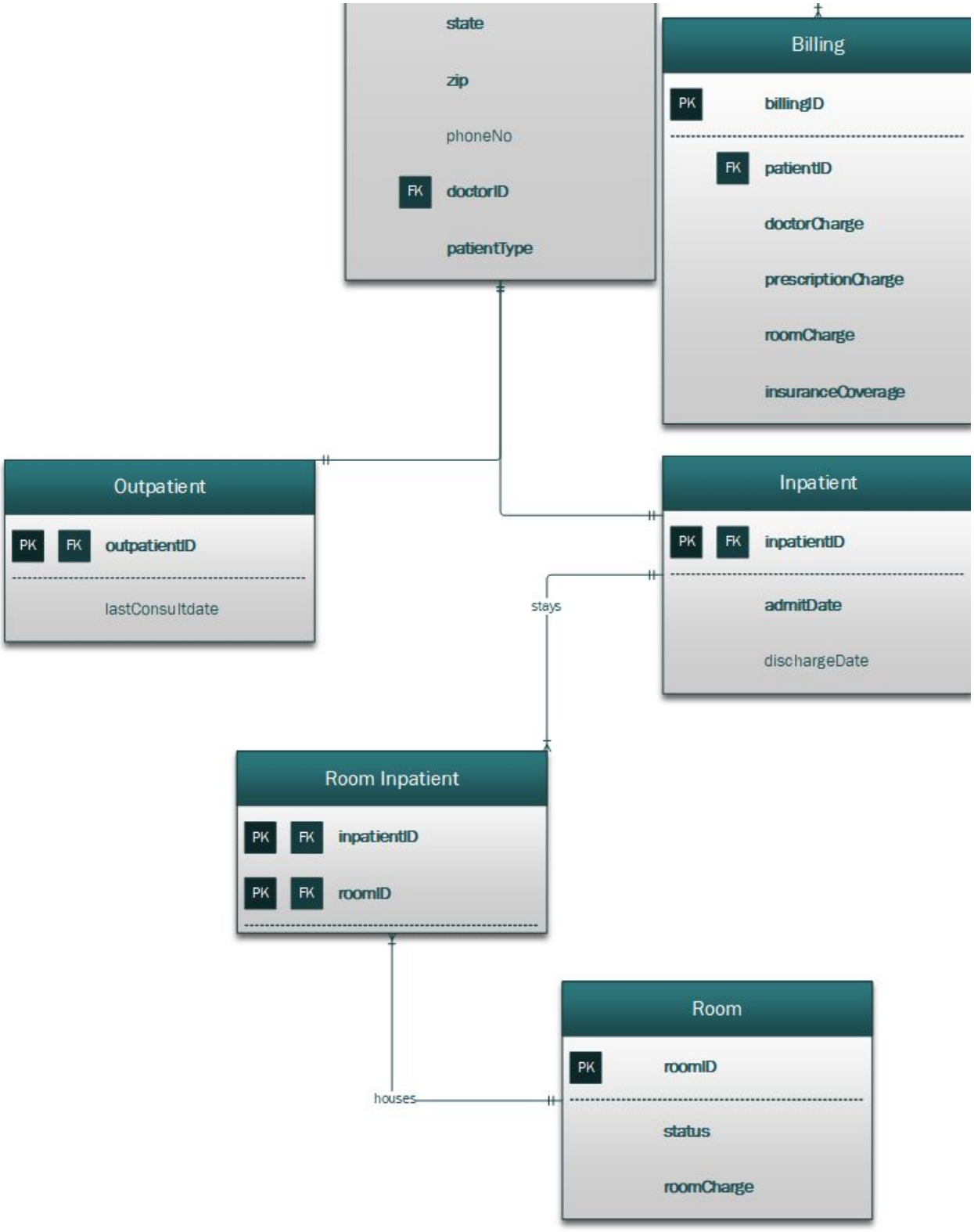
Attribute Name	Field Type	Null/Not Null	Description
inpatientID - PK, FK to inpatient	INT	Not Null	Unique identifier of patient who has been admitted
roomID - PK, FK to room	INT	Not Null	Room number where patient is located

Appointment Table: Details of Doctor/Patient Appointments

Attribute Name	Field Type	Null/Not Null	Description
appointmentID - PK	INT	Not Null	ID of scheduled appointment
patientID - FK to patient	INT	Not Null	ID of patient booked for an appointment
doctorID - FK to doctor	INT	Not Null	ID of doctor scheduled for patient with appointment
adminID - FK to administration	INT	Not Null	ID of administrator in charge of booking appointment
appointmentDate	DATE	Null	Date and time of scheduled appointment: YYYY-MM-DD

Relational Data Model:


Admin
PK
adminID
firstName
lastName
phoneNo
departmentName
officeNo
Doctor
PK
doctorID
firstName
lastName
departmentName
SSN
yearsOfPractice
officeNo
charge
booked for
handles
Appointment
PK
appointmentID
FK
patientID
FK
doctorID
FK
adminID
date
booked by
Patient
PK
patientID
firstName
lastName
DOB
age
weight
gender
streetAddress
city
diagnoses
diagnosed
Diagnosis
PK
diagnosisID
diagnosis
diagnosisDate
FK
patientID
FK
doctorID
billed


state
zip
phoneNo
FK doctorID
patientType
Billing
PK billingID
FK patientID
doctorCharge
prescriptionCharge
roomCharge
insuranceCoverage
Outpatient
PK FK outpatientID
lastConsultdate
Inpatient
PK FK inpatientID
admitDate
dischargeDate
stays
Room Inpatient
PK FK inpatientID
PK FK roomID
houses
Room
PK roomID
status
roomCharge

DATABASE SYSTEM INFRASTRUCTURE

The following tools were used to create our database management system:

1. **Microsoft Viso**: Utilized to create our entity relationship diagram. Entities and attributes were created and identified giving our database a foundation to build upon. Relationships and cardinalities were established between entities and primary and foreign keys were also defined.
2. **MySQL Community Server**: SQL Community server was used to store all tables and data in our RDBMS. Triggers were also deployed using the SQL server.
3. **MS Access:** Microsoft Access was used to create the forms associated with our user interface system. Tables that we created in the SQL server were also linked using Microsoft Access. Following this step, forms were created to display necessary information to users and to store user input. Following input, reports were generated using access to help users interpret information generated in our RDBMS.

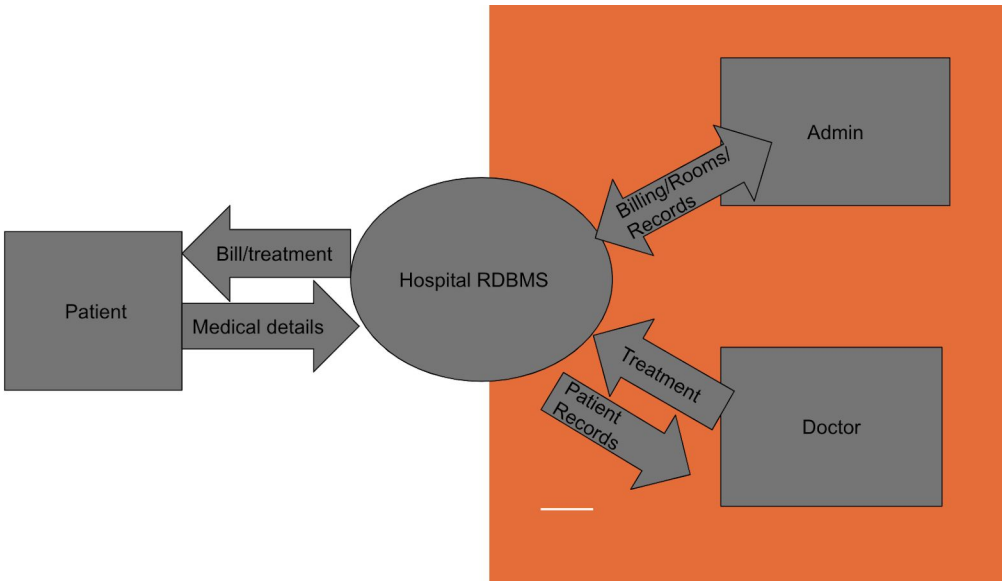


Figure 1: Logical Design of Database System

SQL SCRIPT FOR CREATING TABLES AND INSERTING DATA:

```
CREATE TABLE ADMINISTRATION
(
    adminID INT NOT NULL PRIMARY KEY, -- primary key column
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(30) NOT NULL,
    phoneNo CHAR(10) NOT NULL,
    departmentName VARCHAR(100) CHECK( departmentName IN ('General
Internal Medicine', 'Cardiology', 'Dermatology', 'Endocrinology',
'Gastroenterology', 'Oncology', 'Epidemiology', 'Nephrology',
'Pharmacology', 'Pulmonology', 'Rheumatology', 'ER')),
    officeNo VARCHAR(3)
    -- specify more columns here
);

CREATE TABLE DOCTOR
(
    doctorID INT NOT NULL PRIMARY KEY, -- primary key column
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(30) 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,
    charge NUMERIC NOT NULL CHECK(charge > 0)-- A flat hourly rate,
different for each doctor
);


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 DEFAULT GETDATE() CHECK(DOB <= GETDATE()),
    age NUMERIC NOT NULL CHECK(age >= 0),
    gender VARCHAR(2) NOT NULL CHECK(gender IN ('M', 'F', 'NA')), --
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
    zip NUMERIC(5) NOT NULL,
    phoneNo CHAR(10),
    doctorID 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), -- in pounds (lbs)

    CONSTRAINT patient_fk FOREIGN KEY (doctorID) REFERENCES
DOCTOR(doctorID)
);

CREATE TABLE APPOINTMENT
(
     appointmentID INT NOT NULL PRIMARY KEY,
     patientID INT NOT NULL,
     doctorID INT NOT NULL,
     adminID INT NOT NULL,
     appointmentDate DATE DEFAULT GETDATE() CHECK(appointmentDate <=
GETDATE()),

     CONSTRAINT appointment_fk1 FOREIGN KEY (doctorID) REFERENCES
DOCTOR(doctorID),
     CONSTRAINT appointment_fk2 FOREIGN KEY (patientID) REFERENCES
PATIENT(patientID),
     CONSTRAINT appointment_fk3 FOREIGN KEY (doctorID) REFERENCES
ADMINISTRATION(adminID)
);
```

```
CREATE TABLE DIAGNOSIS
(
    diagnosisID INT NOT NULL PRIMARY KEY, -- primary key column
    doctorID INT NOT NULL,
    patientID INT NOT NULL,
    diagnosisCategory VARCHAR(100) NOT NULL CHECK(diagnosisCategory IN
('Hypertension','Hyperlipidemia','Diabetes','Back
pain','Anxiety','Obesity','Allergic rhinitis','Reflux
esophagitis','Respiratory problems','Hypothyroidism','Visual refractive
errors','Osteoarthritis','myositis','Pain in joint','Acute
laryngopharyngitis','Acute maxillary sinusitis','Major depressive
disorder','Acute bronchitis','Asthma','Skin Disease','Coronary
atherosclerosis','Urinary tract
infection','Influenza','Tiberculosis','Viral infecton','Celiac
Disease','Seizure Disorder','Cerebral Palsy','Tourette
Syndrome','Attention Deficit Disorder','Down Syndrome','Crohns Disease')),
    diagnosis TEXT,
    diagnosisDate DATE NOT NULL CHECK(diagnosisDate <= GETDATE()),

    CONSTRAINT diagnosis_fk1 FOREIGN KEY (doctorID) REFERENCES
DOCTOR(doctorID),
    CONSTRAINT diagnosis_fk2 FOREIGN KEY (patientID) REFERENCES
PATIENT(patientID),
);

CREATE TABLE BILLING
(
    billingID INT NOT NULL PRIMARY KEY, -- primary key column
    patientID INT NOT NULL,
    doctorCharge NUMERIC NOT NULL DEFAULT 0, -- same as the rate of doctor
    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 <= GETDATE()),
```

```
    CONSTRAINT billing_fk FOREIGN KEY (patientID) REFERENCES
PATIENT(patientID),
);

CREATE TABLE INPATIENT
(
    inpatientID INT NOT NULL PRIMARY KEY, -- primary key column
    admitDate DATE NOT NULL DEFAULT GETDATE() CHECK(admitDate <=
GETDATE()),
    dischargeDate DATE,

    CONSTRAINT inpatient_fk FOREIGN KEY (inpatientID) REFERENCES
PATIENT(patientID),
);

CREATE TABLE OUTPATIENT
(
    outpatientID INT NOT NULL PRIMARY KEY, -- primary key column
    lastConsultDate DATE CHECK(lastConsultDate < GETDATE()),

    CONSTRAINT outpatient_fk FOREIGN KEY (outpatientID) REFERENCES
PATIENT(patientID),
);

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), -- charge per day
);

CREATE TABLE ROOM_INPATIENT
(
    inpatientID INT NOT NULL, -- primary key column
    roomID INT NOT NULL,
```

```
    CONSTRAINT room_inpatient_pk PRIMARY KEY (inpatientID,roomID),

    CONSTRAINT room_inpatient_fk1 FOREIGN KEY (inpatientID) REFERENCES
INPATIENT(INpatientID),
    CONSTRAINT room_inpatient_fk2 FOREIGN KEY (roomID) REFERENCES
ROOM(roomID),

);

--Insert Values into ADMINISTRATION Table
INSERT INTO ADMINISTRATION VALUES(1,'John', 'Dorian','1111111111','General
Internal Medicine','112');
INSERT INTO ADMINISTRATION VALUES(2,'Elliot',
'Reid','2222222222','Cardiology','223');
INSERT INTO ADMINISTRATION VALUES(3,'Christopher',
'Turk','3333333333','Dermatology','334');
INSERT INTO ADMINISTRATION VALUES(4,'Percival',
'Cox','4444444444','Oncology','445');
INSERT INTO ADMINISTRATION VALUES(5,'Bob',
'Kelso','5555555555','Gastroenterology','556');
INSERT INTO ADMINISTRATION VALUES(6,'Todd',
'Quinlan','6666666666','Endocrinology','667');
INSERT INTO ADMINISTRATION VALUES(7,'John',
'Wen','7777777777','Pulmonology','778');
INSERT INTO ADMINISTRATION VALUES(8,'Keith',
'Dudemeister','8888888888','Pharmacology','889');
INSERT INTO ADMINISTRATION VALUES(9,'Molly',
'Clock','9999999999','Nephrology','990');

--Insert Values into DOCTOR Table
INSERT INTO doctor VALUES(1,'Susan', 'Grey','General Internal
Medicine','111111111',10,'111',250);
INSERT INTO doctor VALUES(2,'Chris',
'Billinson','Cardiology','222222222',2,'222',300);
INSERT INTO doctor VALUES(3,'John',
'Noble','Dermatology','333333333',4,'333',125);
```

```
INSERT INTO doctor VALUES(4,'Beth',
'Rettinger','Oncology','444444444',7,'444',260);
INSERT INTO doctor VALUES(5,'Amy',
'Cote','Gastroenterology','555555555',1,'555',530);
INSERT INTO doctor VALUES(6,'Phil',
'Kinsella','Endocrinology','666666666',9,'666',120);
INSERT INTO doctor VALUES(7,'Patricia',
'Smith','Pulmonology','777777777',8,'777',550);
INSERT INTO doctor VALUES(8,'Jeffrey',
'Carpenter','Pharmacology','888888888',23,'888',400);
INSERT INTO doctor VALUES(9,'Amanda',
'Shock','Nephrology','999999999',1,'999',1000);

--Insert Values into PATIENT Table
INSERT INTO PATIENT VALUES(1,'Timothy',
'Gamble','1996-10-16',23,'M',123,'Cary
Road','Manlius','NY',13104,'3153453651',1,'I',172,180);
INSERT INTO PATIENT VALUES(2,'Chris',
'Richards','1990-11-12',27,'M',234,'Bridge
Avenue','Manlius','NY',13104,'3154256157',3,'I',150,164);
INSERT INTO PATIENT VALUES(3,'Chase',
'Roberts','1986-02-14',33,'M',345,'Lorraine
Avenue','Syracuse','NY',16802,'6157267893',2,'I',144,220);
INSERT INTO PATIENT VALUES(4,'Nancy',
'Frechette','1964-01-19',55,'F',456,'Carrier
Drive','Liverpool','NY',16803,'5152620092',5,'I',130,135);
INSERT INTO PATIENT VALUES(5,'Elvira',
'Robinson','2001-02-13',18,'F',567,'Taft
Lane','Fayetteville','NY',22222,'1236728172',4,'I',190,240);
INSERT INTO PATIENT VALUES(6,'Lucy',
'Puro','2004-02-04',15,'F',678,'Barksdale
Lane','Baldwinsville','NY',31215,'3334125263',6,'O',115,100);
INSERT INTO PATIENT VALUES(7,'Jamal',
'Badger','1997-07-21',22,'M',789,'Trillium
Trail','Manlius','NY',13104,'4447267281',9,'O',156,145);
```

```
INSERT INTO PATIENT VALUES(8,'Rick',
'Carlton','1972-01-01',47,'M',890,'Parker
Drive','Fayetteville','NY',22222,'7772891827',8,'O',174,210);
INSERT INTO PATIENT VALUES(9,'Sally',
'Baker','1952-04-20',67,'F',012,'Trout
Road','Syracuse','NY',16802,'7268880290',7,'O',189,214);

--Insert Values into APPOINTMENT Table
INSERT INTO APPOINTMENT VALUES(100000001,1,1,9,'2019-09-08');
INSERT INTO APPOINTMENT VALUES(100000002,2,3,2,'2019-10-04');--
INSERT INTO APPOINTMENT VALUES(100000003,3,6,8,'2019-10-02');
INSERT INTO APPOINTMENT VALUES(100000004,4,5,3,'2019-04-06');
INSERT INTO APPOINTMENT VALUES(100000005,5,4,5,'2019-08-16');--
INSERT INTO APPOINTMENT VALUES(100000006,6,2,6,'2019-05-11');--
INSERT INTO APPOINTMENT VALUES(100000007,7,8,7,'2019-11-10');--
INSERT INTO APPOINTMENT VALUES(100000008,8,7,4,'2019-11-09');--
INSERT INTO APPOINTMENT VALUES(100000009,9,9,1,'2019-07-16');

--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','2019-09-08');
INSERT INTO DIAGNOSIS VALUES(100000022,2,3,'Diabetes','Type II Diabetic,
must reduce sugar and intake and take Insulin once daily','2019-10-05');
INSERT INTO DIAGNOSIS VALUES(100000033,3,2,'Back Pain','Fractured L3
Vertebrae, perscribed Oxycontin, recommended 30 mg / daily for one
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','2019-08-16');
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','2019-04-07');
INSERT INTO DIAGNOSIS VALUES(100000066,6,6,'Obesity','Referred to
Dietician. Must reduce sugar intake and exercise regularly','2019-05-11');
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,'2019-09-09');
INSERT INTO BILLING VALUES(100000222,2,125,400,100,60,'2019-10-06');--
INSERT INTO BILLING VALUES(100000333,3,300,80,0,40,'2019-10-06');
INSERT INTO BILLING VALUES(100000444,4,530,0,0,10,'2019-04-08');
INSERT INTO BILLING VALUES(100000555,5,260,90,100,0,'2019-08-18');--
INSERT INTO BILLING VALUES(100000666,6,120,0,100,55,'2019-05-13');--
INSERT INTO BILLING VALUES(100000777,7,1000,15,100,100,'2019-11-12');--
INSERT INTO BILLING VALUES(100000888,8,400,0,100,15,'2019-11-11');--
INSERT INTO BILLING VALUES(100000999,9,550,100,0,45,'2019-11-11');

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

--Insert Values into Outpatient Table
INSERT INTO OUTPATIENT VALUES(6,'2019-09-08');--
INSERT INTO OUTPATIENT VALUES(7,'2019-10-02');--
INSERT INTO OUTPATIENT VALUES(8,'2019-04-06');--
INSERT INTO OUTPATIENT VALUES(9,'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);


--Insert Values into ROOM_INPATIENT Table
INSERT INTO ROOM_INPATIENT VALUES(1,1);
INSERT INTO ROOM_INPATIENT VALUES(2,2);
INSERT INTO ROOM_INPATIENT VALUES(3,3);
INSERT INTO ROOM_INPATIENT VALUES(4,4);
INSERT INTO ROOM_INPATIENT VALUES(5,5);
```

Output for SQL Scripts:

The following tables contain data that was created using the aforementioned SQL code. Access forms and triggers are also displayed near the end of this section:

1. Administration Table

	adminID	firstName	lastName	phoneNo	departmentName	officeNo
1	1	John	Dorian	1111111111	General Internal Medicine	112
2	2	Elliot	Reid	2222222222	Cardiology	223
3	3	Christopher	Turk	3333333333	Dermatology	334
4	4	Percival	Cox	4444444444	Oncology	445
5	5	Bob	Kelso	5555555555	Gastroenterology	556
6	6	Todd	Quinlan	6666666666	Endocrinology	667
7	7	John	Wen	7777777777	Pulmonology	778
8	8	Keith	Dudemeister	8888888888	Pharmacology	889
9	9	Molly	Clock	9999999999	Nephrology	990

2. Doctor Table

Results Messages

	doctorID	firstName	lastName	departmentName	SSN	yearsOfPractice	officeNo	charge
1	1	Susan	Grey	General Internal Medicine	111111111	10	111	250
2	2	Chris	Billinson	Cardiology	222222222	2	222	300
3	3	John	Noble	Dermatology	333333333	4	333	125
4	4	Beth	Rettinger	Oncology	444444444	7	444	260
5	5	Amy	Cote	Gastroenterology	555555555	1	555	530
6	6	Phil	Kinsella	Endocrinology	666666666	9	666	120
7	7	Patricia	Smith	Pulmonology	777777777	8	777	550
8	8	Jeffrey	Carpenter	Pharmacology	888888888	23	888	400
9	9	Amanda	Shock	Nephrology	999999999	1	999	1000

3. Patient Table

	patientID	firstName	lastName	DOB	age	gender	streetNo	streetName	city	stateName	zip	phoneNo	doctorID	patientType	patientHeight	patientWeight
1	1	Timothy	Gamble	1996-10-16	23	M	123	Cary Road	Manlius	NY	13104	3153453651	1	I	172	180
2	2	Chris	Richards	1990-11-12	27	M	234	Bridge Avenue	Manlius	NY	13104	3154256157	3	I	150	164
3	3	Chase	Roberts	1986-02-14	33	M	345	Lorraine Avenue	Syracuse	NY	16802	6157267893	2	I	144	220
4	4	Nancy	Frechette	1964-01-19	55	F	456	Carrier Drive	Liverpool	NY	16803	5152620092	5	I	130	135
5	5	Elvira	Robinson	2001-02-13	18	F	567	Taft Lane	Fayetteville	NY	22222	1236728172	4	I	190	240
6	6	Lucy	Puro	2004-02-04	15	F	678	Barksdale Lane	Baldwinsville	NY	31215	3334125263	6	O	115	100
7	7	Jamal	Badger	1997-07-21	22	M	789	Trillium Trail	Manlius	NY	13104	4447267281	9	O	156	145
8	8	Rick	Carlton	1972-01-01	47	M	890	Parker Drive	Fayetteville	NY	22222	7772891827	8	O	174	210
9	9	Sally	Baker	1952-04-20	67	F	12	Trout Road	Syracuse	NY	16802	7268880290	7	O	189	214

4. Appointment Table

Results | Messages

	appointmentID	patientID	doctorID	adminID	appointmentDate
1	100000001	1	1	9	2019-09-08
2	100000002	2	3	2	2019-10-04
3	100000003	3	6	8	2019-10-02
4	100000004	4	5	3	2019-04-06
5	100000005	5	4	5	2019-08-16
6	100000006	6	2	6	2019-05-11
7	100000007	7	8	7	2019-11-10
8	100000008	8	7	4	2019-11-09
9	100000009	9	9	1	2019-07-16

5. Diagnosis Table

Results | Messages

	diagnosisID	doctorID	patientID	diagnosisCategory	diagnosis	diagnosisDate
1	100000011	1	1	Hypertension	High Systolic BP. High Salt Diet, must reduce and...	2019-09-08
2	100000022	2	3	Diabetes	Type II Diabetic, must reduce sugar and intake an...	2019-10-05
3	100000033	3	2	Back Pain	Fractured L3 Vertebrae, perscribed Oxycontin, re...	2019-10-04
4	100000044	4	5	Anxiety	Reduced levels of Serotonin in the brain, perscrib...	2019-08-16
5	100000055	5	4	Allergic rhinitis	Inflamed sinus, stuffy nose for 2 weeks. Take OT...	2019-04-07
6	100000066	6	6	Obesity	Referred to Dietician. Must reduce sugar intake a...	2019-05-11
7	100000077	7	9	Hypothyroidism	Hyperactive thyroid leading to weight gain and lac...	2019-11-10
8	100000088	8	8	Osteoarthritis	Joint Pain in left knee following lifting boxes. Take...	2019-11-09
9	100000099	9	7	Acute bronchitis	Severe coughing fits. Prescribed inhaler from nea...	2019-11-10

6. Billing Table

Results | Messages

	billingID	patientID	doctorCharge	prescriptionCharge	roomCharge	insuranceCoveragePercentage	billingDate
1	100000111	1	250	40	0	80	2019-09-09
2	100000222	2	125	400	100	60	2019-10-06
3	100000333	3	300	80	0	40	2019-10-06
4	100000444	4	530	0	0	10	2019-04-08
5	100000555	5	260	90	100	0	2019-08-18
6	100000666	6	120	0	100	55	2019-05-13
7	100000777	7	1000	15	100	100	2019-11-12
8	100000888	8	400	0	100	15	2019-11-11
9	100000999	9	550	100	0	45	2019-11-11

7. Outpatient Table

Results Messages

	outpatientID	lastConsultDate
1	6	2019-09-08
2	7	2019-10-02
3	8	2019-04-06
4	9	2019-07-16

8. Inpatient Table

Results Messages

	inpatientID	admitDate	dischargeDate
1	1	2019-10-04	2019-10-05
2	2	2019-08-16	2019-08-17
3	3	2019-05-11	2019-05-12
4	4	2019-11-10	2019-11-11
5	5	2019-11-09	2019-11-10

9. Room Table

	roomID	roomStatus	roomCharge
1	1	Occupied	100
2	2	Vacant	100
3	3	Vacant	100
4	4	Vacant	100
5	5	Vacant	100
6	6	Vacant	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

10. Room Inpatient Table

	inpatientID	roomID
1	1	1
2	2	1
3	2	2
4	3	3
5	4	4
6	5	5

TRIGGERS:

This trigger activates on ROOM_INPATIENT INSERT or UPDATE queries. This trigger allows the admin to assign a room to a patient only if the roomStatus of ROOM Table is 'Vacant'. As each room can have 2 patients, as soon as there are two patients in a single room, the trigger updates the roomStatus in ROOM table to 'Occupied'

```
CREATE TRIGGER assignRoom_trigger
ON ROOM_INPATIENT
FOR INSERT, UPDATE AS
IF @@ROWCOUNT >=1

    BEGIN
    DECLARE @roomID INT
    DECLARE @inpatientID INT
    DECLARE @r_count INT

    SELECT @roomID=roomID, @inpatientID=inpatientID from inserted

    IF EXISTS (SELECT * FROM ROOM WHERE roomStatus = 'Vacant' and roomID =
@roomID)
        BEGIN
            SELECT @r_count = COUNT(*) FROM ROOM_INPATIENT WHERE roomID =
@roomID GROUP BY roomID
            IF @r_count = 2
                BEGIN
                    UPDATE ROOM
                    SET roomStatus = 'Occupied'
                    WHERE roomID = @roomID AND roomStatus <> 'Occupied'
                END
            ELSE
                BEGIN
                    IF @r_count >= 3
                        RAISERROR (15602,-1,-1, 'The room already contains
2 patients, please assign a different room');
                END
```

```
            END
    ELSE
        RAISERROR (15603,-1,-1, 'The room is already occupied, please
check which rooms are available');
    END
```

STORED PROCEDURE:

This procedure can be used to assign rooms to inpatients without writing the sql query.

```
CREATE PROCEDURE assignRoom
    @inpatientID int,
    @roomID int

AS
    BEGIN

        INSERT INTO ROOM_INPATIENT VALUES (@inpatientID, @roomID)
        SELECT *
        FROM ROOM
        WHERE roomID = @roomID
    END
GO
```

Major Data Questions

1. **What is the total cost of a hospital visit for any given patient?**
   SQL Query:

```
SELECT dbo_BILLING.billingID, dbo_BILLING.patientID,
dbo_PATIENT.patientType, dbo_BILLING.billingDate,
dbo_BILLING.doctorCharge, dbo_BILLING.prescriptionCharge,
dbo_BILLING.roomCharge, IIF(dbo_INPATIENT.dischargeDate IS NOT
NULL,ABS(DATEDIFF("d", dbo_INPATIENT.admitDate,
dbo_INPATIENT.dischargeDate)),NULL) AS NumberOfDays,
IIF(dbo_INPATIENT.dischargeDate IS NOT
NULL,dbo_BILLING.roomCharge*(ABS(DATEDIFF("d",
dbo_INPATIENT.admitDate, dbo_INPATIENT.dischargeDate))),0) AS
TotalRoomCharge,
dbo_BILLING.insuranceCoveragePercentage,
```

```
(1-dbo_BILLING.insuranceCoveragePercentage/100)*(IIF(dbo_INPATIENT.d
ischargeDate IS NOT NULL,
dbo_BILLING.roomCharge*(ABS(DATEDIFF("d", dbo_INPATIENT.admitDate,
dbo_INPATIENT.dischargeDate))),0)+
dbo_BILLING.doctorCharge+dbo_BILLING.prescriptionCharge) AS
TotalBill
FROM (dbo_PATIENT INNER JOIN dbo_BILLING ON dbo_PATIENT.patientID =
dbo_BILLING.patientID)
INNER JOIN dbo_INPATIENT ON dbo_PATIENT.patientID =
dbo_INPATIENT.inpatientID;
UNION
SELECT dbo_BILLING.billingID, dbo_BILLING.patientID,
dbo_PATIENT.patientType, dbo_BILLING.billingDate,
dbo_BILLING.doctorCharge,
dbo_BILLING.prescriptionCharge, NULL, NULL,NULL,
dbo_BILLING.insuranceCoveragePercentage,
(1-dbo_BILLING.insuranceCoveragePercentage/100)*(dbo_BILLING.doctorC
harge+dbo_BILLING.prescriptionCharge) AS TotalBill
FROM (dbo_PATIENT INNER JOIN dbo_BILLING ON dbo_PATIENT.patientID =
dbo_BILLING.patientID)
INNER JOIN dbo_OUTPATIENT ON dbo_PATIENT.patientID =
dbo_OUTPATIENT.outpatientID;
```

Output Datasheet

billingID	patientID	patientType	billingDate	doctorCharge	prescriptionC	roomCharge	NumberOfDa	TotalRoomCl	insuranceCo	TotalBill
100000111	1	I	9/9/2019	250	40	0	1	0	80	58
100000222	2	I	10/6/2019	125	400	100	1	100	60	250
100000333	3	I	10/6/2019	300	80	0	1	0	40	228
100000444	4	I	4/8/2019	530	0	0	1	0	10	477
100000555	5	I	8/18/2019	260	90	100	1	100	0	450
100000666	6	O	5/13/2019	120	0				55	54
100000777	7	O	11/12/2019	1000	15				100	0
100000888	8	O	11/11/2019	400	0				15	340
100000999	9	O	11/11/2019	550	100				45	357.5

Output Report

totalBill_query_report

billingID	patientID	patientType	billingDate	doctorCharge	prescriptionCha	roomCharge	NumberOfDays	talRoomCharge	insuranceCover	TotalBill
100000111	1	I	9/9/2019	250	40	0	1	0	80	58
100000222	2	I	10/6/2019	125	400	100	1	100	60	250
100000333	3	I	10/6/2019	300	80	0	1	0	40	228
100000444	4	I	4/8/2019	530	0	0	1	0	10	477
100000555	5	I	8/18/2019	260	90	100	1	100	0	450
100000666	6	O	5/13/2019	120	0				55	54
100000777	7	O	11/12/2019	1000	15				100	0
100000888	8	O	11/11/2019	400	0				15	340
100000999	9	O	11/11/2019	550	100				45	357.5

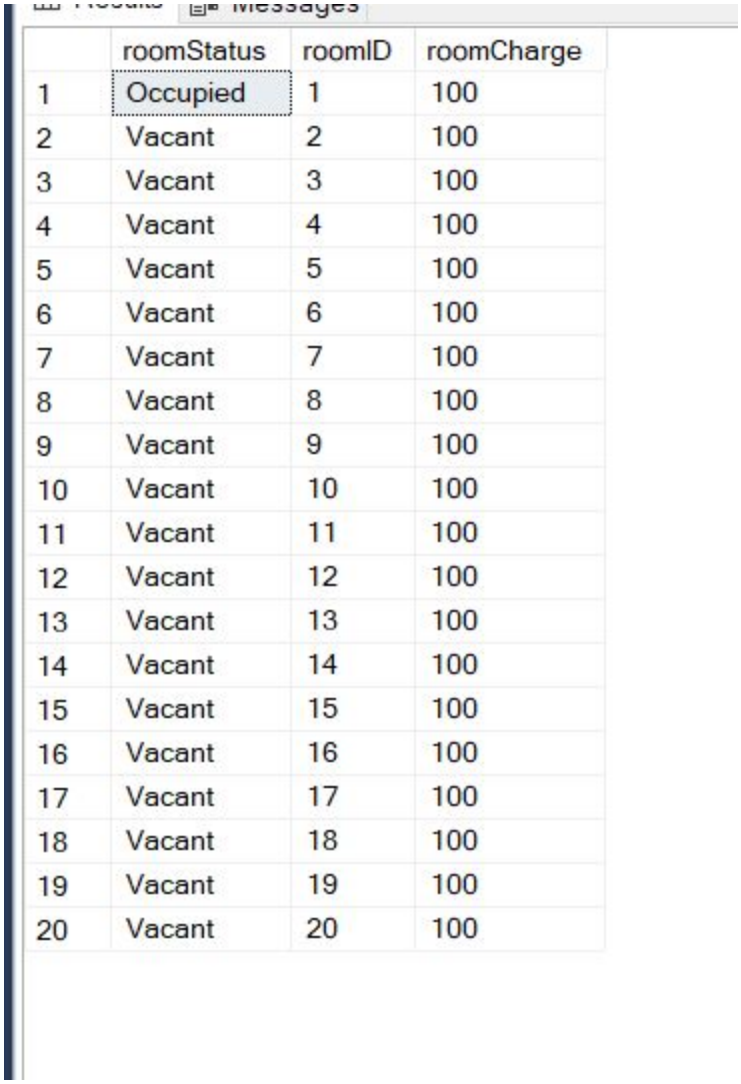
Tuesday, December 3, 2019 Page 1 of 1

2. How can administrative assistants ensure that enough rooms are available for all inpatients?

SQL Query:

```
SELECT roomStatus, roomID, roomCharge
FROM ROOM
GROUP BY roomStatus, roomID, roomCharge
```

Output Datasheet



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

Output Report

dbo_ROOM_report

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

3. **Which patient has the maximum Insurance Coverage(%)?**
   SQL Query

```
SELECT dbo_BILLING.patientID,  dbo_PATIENT.firstName,
dbo_PATIENT.lastName, dbo_BILLING.insuranceCoveragePercentage
FROM dbo_PATIENT INNER JOIN dbo_BILLING ON dbo_PATIENT.patientID =
dbo_BILLING.patientID
ORDER BY dbo_BILLING.insuranceCoveragePercentage DESC;
```

Output Datasheet

patientID	firstName	lastName	insuranceCov
7	Jamal	Badger	100
1	Timothy	Gamble	80
2	Chris	Richards	60
6	Lucy	Puro	55
9	Sally	Baker	45
3	Chase	Roberts	40
8	Rick	Carlton	15
4	Nancy	Frechette	10
5	Elvira	Robinson	0

Output Report

Maximum_Insurance

insuranceCoveragePercentage	patientID	firstName	lastName
100	7	Jamal	Badger
80	1	Timothy	Gamble
60	2	Chris	Richards
55	6	Lucy	Puro
45	9	Sally	Baker
40	3	Chase	Roberts
15	8	Rick	Carlton
10	4	Nancy	Frechette
0	5	Elvira	Robinson

4. **How many doctors are there in a department?**

SQL Query:

```
SELECT dbo_DOCTOR.departmentName, Count(dbo_DOCTOR.doctorID) AS CountOfdoctor
FROM dbo_DOCTOR
GROUP BY dbo_DOCTOR.departmentName;
```

Output Datasheet

departmentName	CountOfdoctor
Cardiology	1
Dermatology	1
Endocrinology	1
Gastroenterology	1
General Internal Medici	1
Nephrology	1
Oncology	1
Pharmacology	1
Pulmonology	1

Output Report

NumberOfDoctors_report

departmentName	CountOfdoctor
Cardiology	1
Dermatology	1
Endocrinology	1
Gastroenterology	1
General Internal Medicine	1
Nephrology	1
Oncology	1
Pharmacology	1
Pulmonology	1

5. **Which doctor is the most experienced based on years of practice and number of appointments?**
   SQL Query:

```
SELECT dbo_DOCTOR.doctorID, dbo_DOCTOR.firstName,
dbo_DOCTOR.lastName, dbo_DOCTOR.yearsOfPractice,
dbo_DOCTOR.departmentName
FROM dbo_DOCTOR
ORDER BY dbo_DOCTOR.yearsOfPractice DESC;
```

   Output Datasheet

doctorID	firstName	lastName	yearsOfPract	departmentN
8	Jeffrey	Carpenter	23	Pharmacology
1	Susan	Grey	10	General Interna
6	Phil	Kinsella	9	Endocrinology
7	Patricia	Smith	8	Pulmonology
4	Beth	Rettinger	7	Oncology
3	John	Noble	4	Dermatology
2	Chris	Billinson	2	Cardiology
5	Amy	Cote	1	Gastroenterolo
9	Amanda	Shock	1	Nephrology

   Output Report

ExperienceDoctor_report1

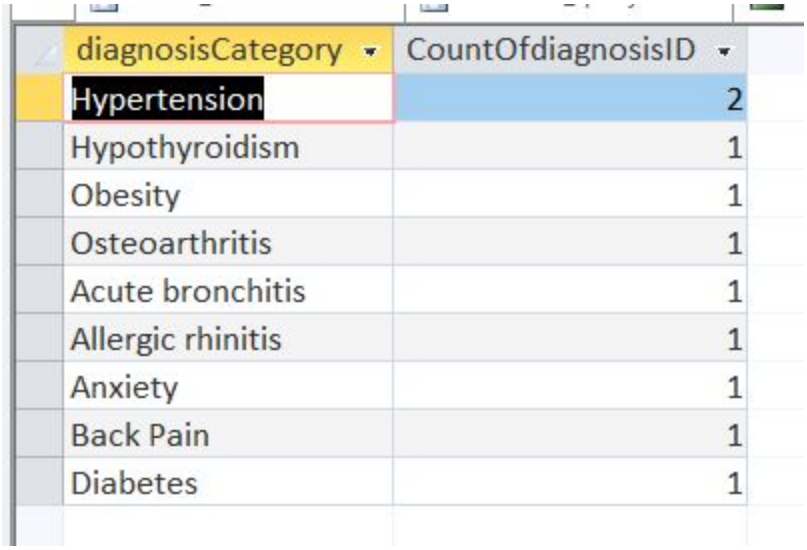
doctorID	firstName	lastName	yearsOfPractice	departmentName
8	Jeffrey	Carpenter	23	Pharmacology
1	Susan	Grey	10	General Internal Medicine
6	Phil	Kinsella	9	Endocrinology
7	Patricia	Smith	8	Pulmonology
4	Beth	Rettinger	7	Oncology
3	John	Noble	4	Dermatology
2	Chris	Billinson	2	Cardiology
5	Amy	Cote	1	Gastroenterology
9	Amanda	Shock	1	Nephrology

6. **What diseases are most common?**

SQL Query:

```
SELECT dbo_DIAGNOSIS.diagnosisCategory, Count(dbo_DIAGNOSIS.diagnosisID)
AS CountOfdiagnosisID
FROM dbo_DIAGNOSIS
GROUP BY dbo_DIAGNOSIS.diagnosisCategory
ORDER BY Count(dbo_DIAGNOSIS.diagnosisID) DESC;
```

Output Datasheet



diagnosisCategory	CountOfdiagnosisID
Hypertension	2
Hypothyroidism	1
Obesity	1
Osteoarthritis	1
Acute bronchitis	1
Allergic rhinitis	1
Anxiety	1
Back Pain	1
Diabetes	1

Output Report

Common_diagnosis_query_report

CountOfdiagnosisID	diagnosisCategory
2	Hypertension
1	Hypothyroidism
1	Obesity
1	Osteoarthritis
1	Acute bronchitis
1	Allergic rhinitis
1	Anxiety
1	Back Pain
1	Diabetes

Interfaces:

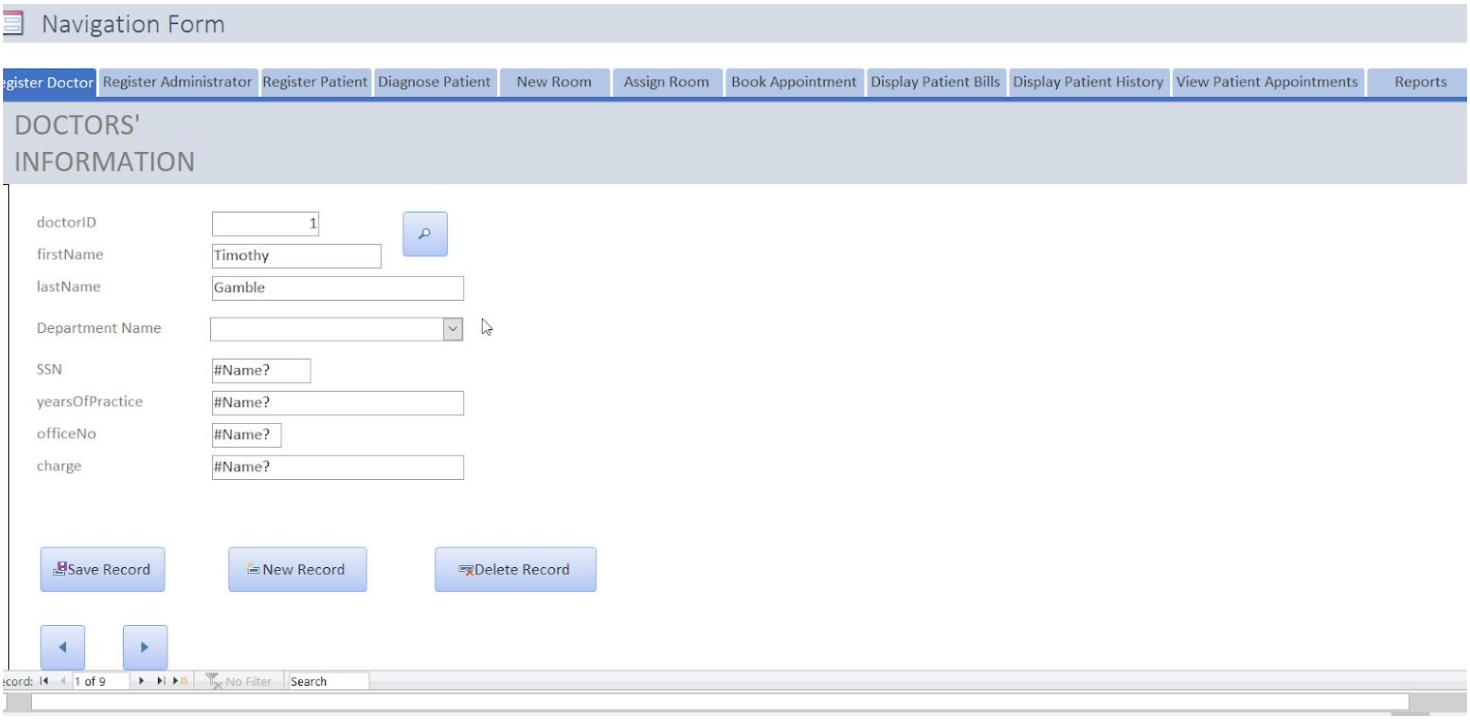
Our GUI has three levels of users:

- Admin: Schedule appointments/verify patient information/book rooms/administer bills
- Doctors: view/update personal and patient medical information
- Patients: view treatment notes/update personal information/view medical costs'

FORMS:

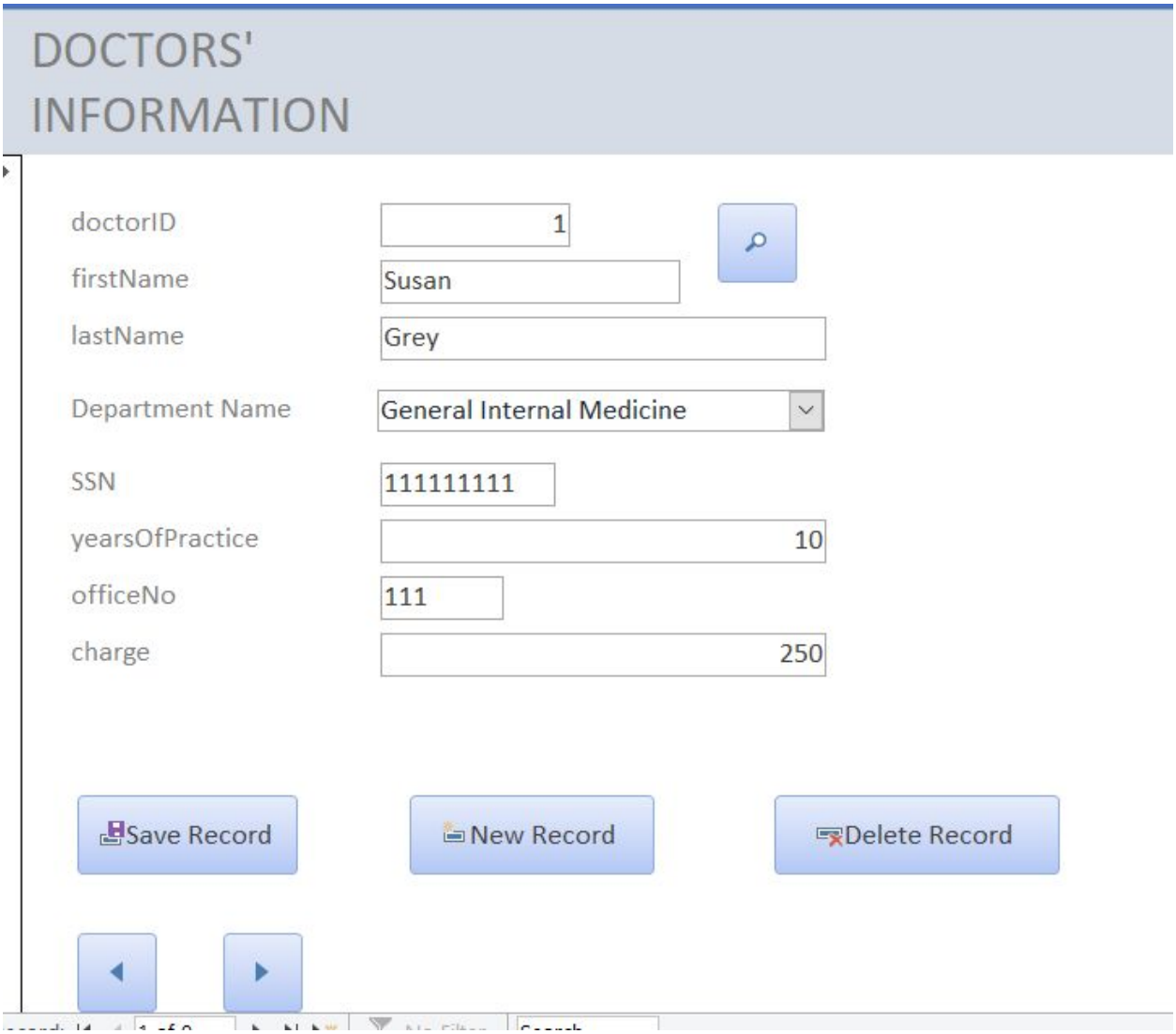
1. Navigation Form

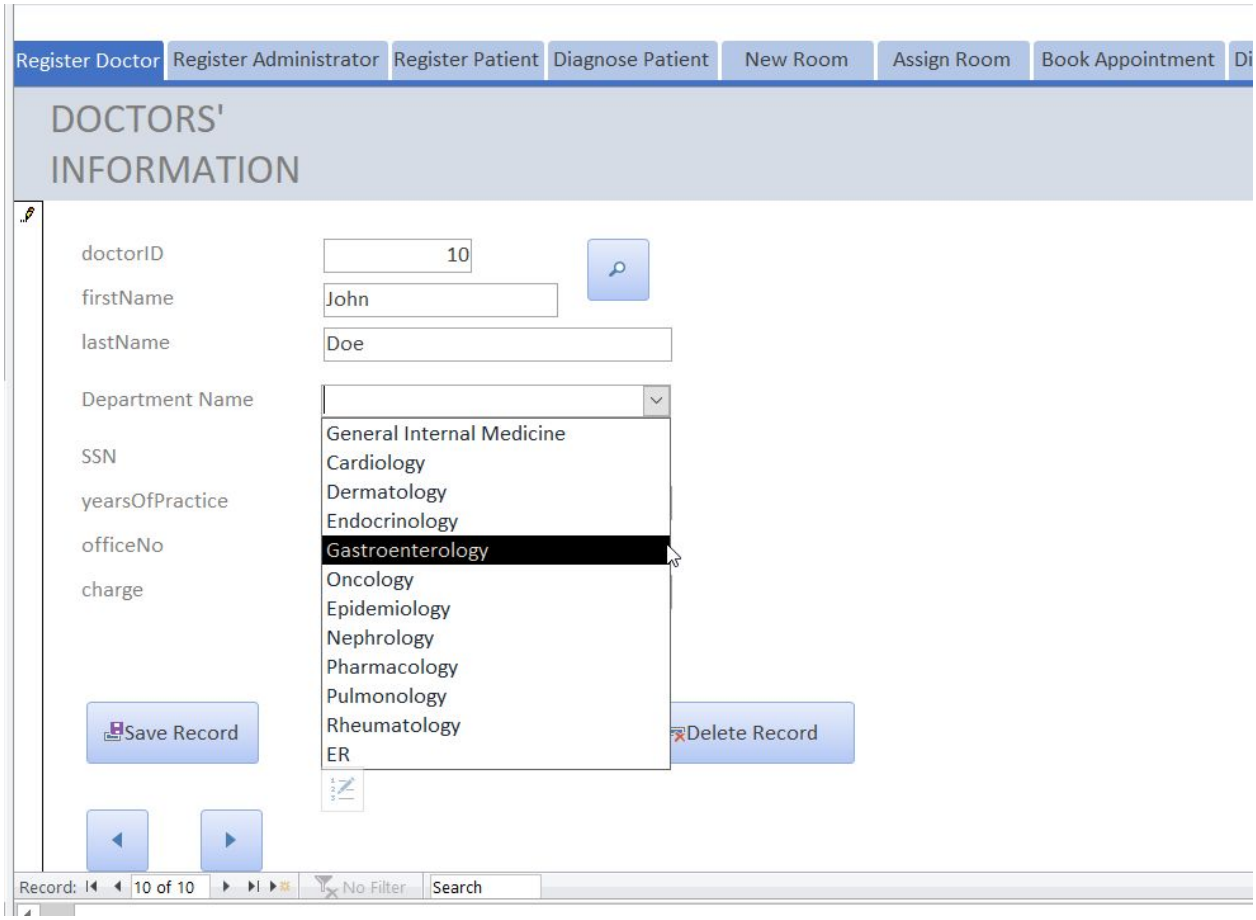
For our Interface, there is a Navigation Form, through which the user can switch between different forms and also look at the generated reports.



2. Register a Doctor

   We use the Doctors' Information form to register new doctors.

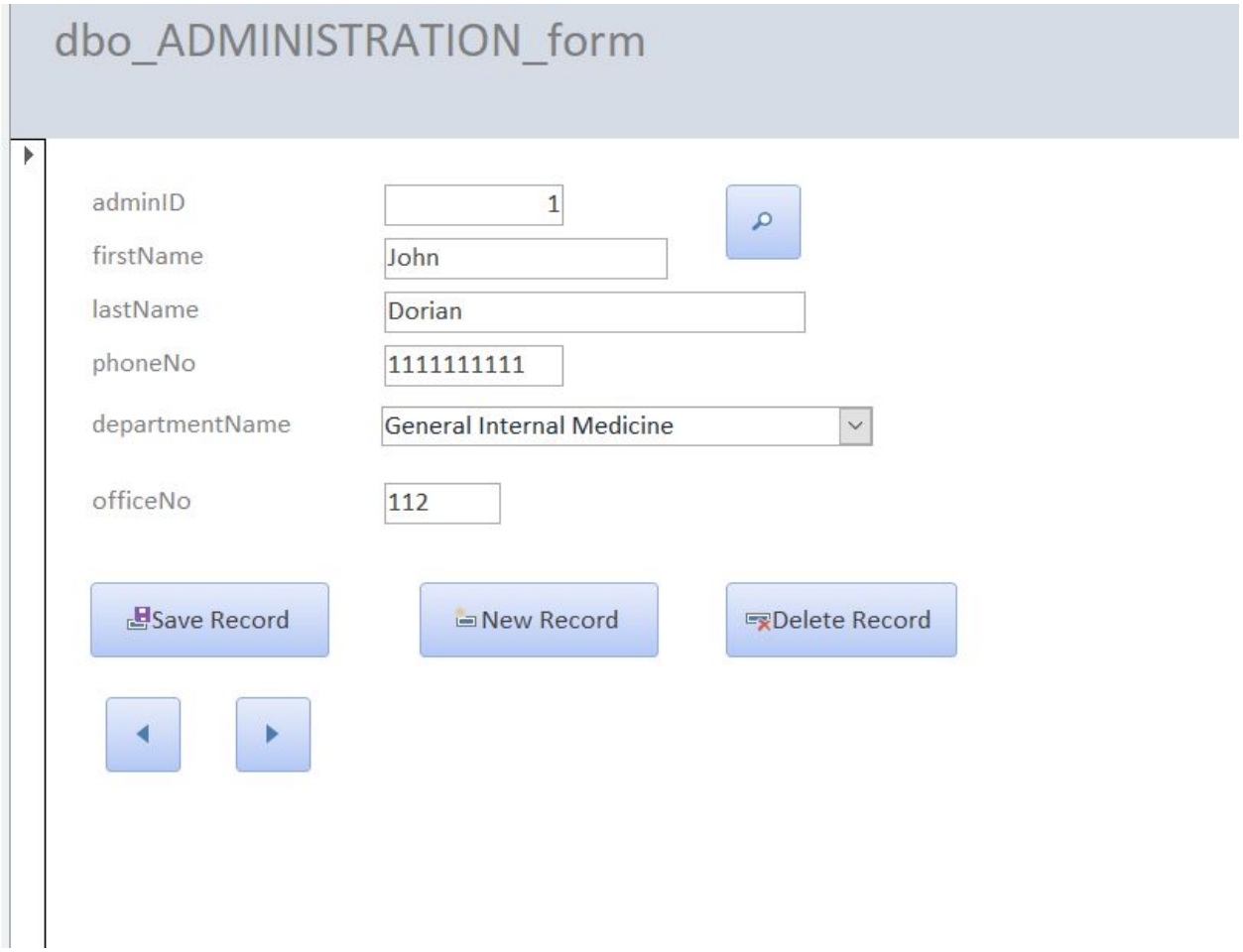


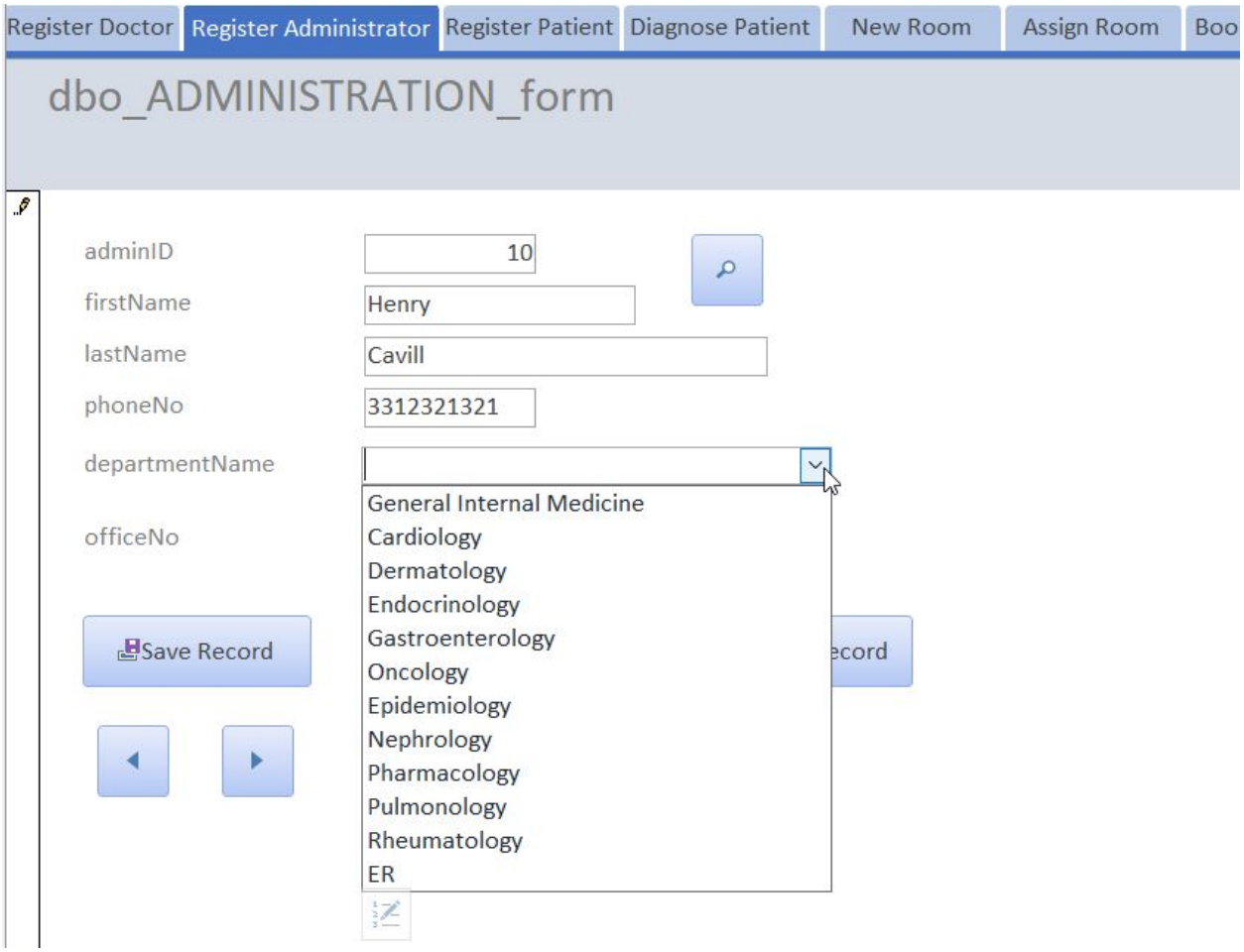


	doctorID	firstName	lastName	departmentName	SSN	yearsOfPractice	officeNo	charge
1	1	Susan	Grey	General Internal Medicine	111111111	10	111	250
2	2	Chris	Billinson	Cardiology	222222222	2	222	300
3	3	John	Noble	Dermatology	333333333	4	333	125
4	4	Beth	Rettinger	Oncology	444444444	7	444	260
5	5	Amy	Cote	Gastroenterology	555555555	1	555	530
6	6	Phil	Kinsella	Endocrinology	666666666	9	666	120
7	7	Patricia	Smith	Pulmonology	777777777	8	777	550
8	8	Jeffrey	Carpenter	Pharmacology	888888888	23	888	400
9	9	Amanda	Shock	Nephrology	999999999	1	999	1000
10	10	John	Doe	Gastroenterology	313213213	35	332	500

3. Register an Administrator

   The Administrator form is used to register any new users to the database.



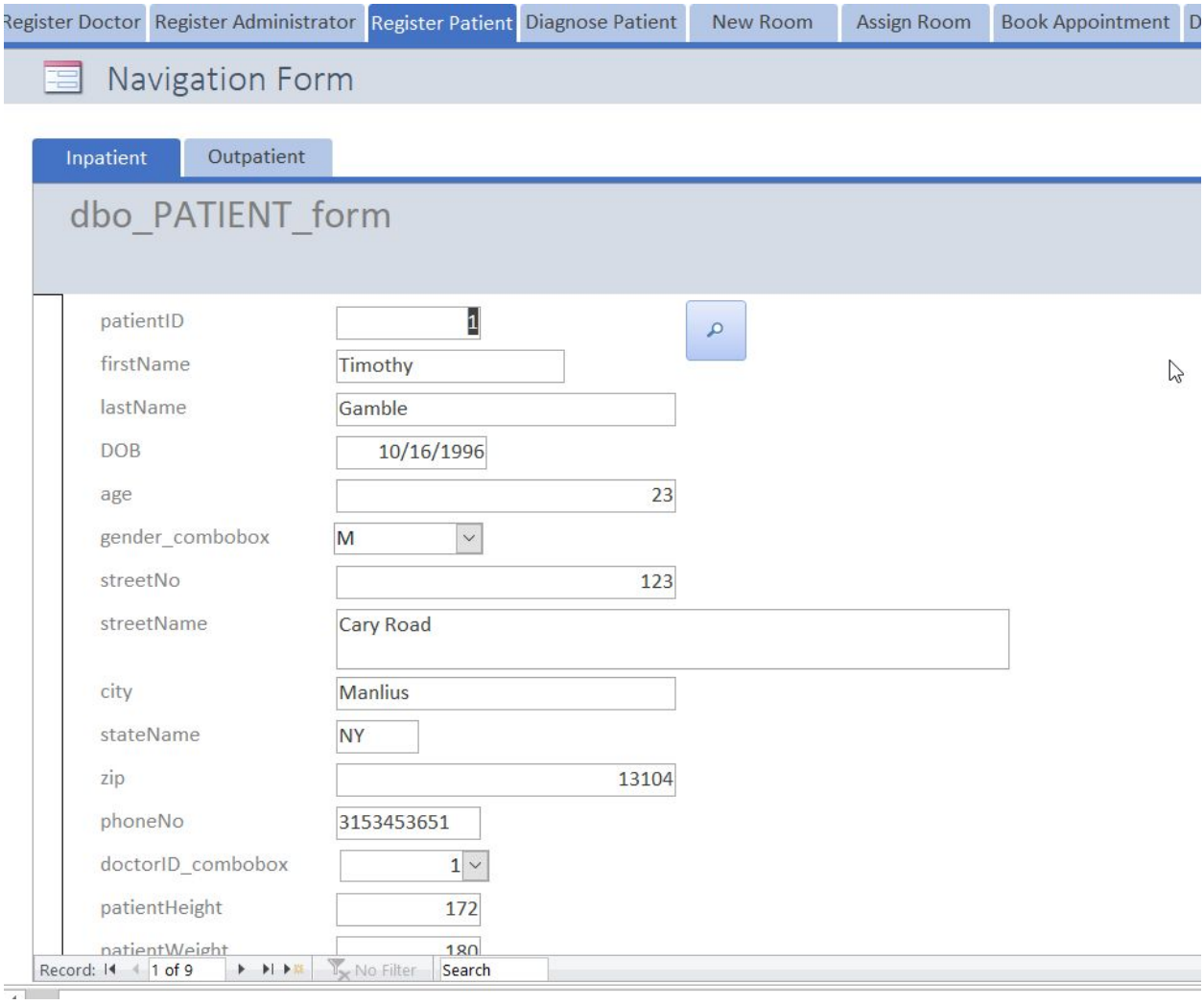


	adminID	firstName	lastName	phoneNo	departmentName	officeNo
1	1	John	Dorian	1111111111	General Internal Medicine	112
2	2	Elliot	Reid	2222222222	Cardiology	223
3	3	Christopher	Turk	3333333333	Dermatology	334
4	4	Percival	Cox	4444444444	Oncology	445
5	5	Bob	Kelso	5555555555	Gastroenterology	556
6	6	Todd	Quinlan	6666666666	Endocrinology	667
7	7	John	Wen	7777777777	Pulmonology	778
8	8	Keith	Dudemeister	8888888888	Pharmacology	889
9	9	Molly	Clock	9999999999	Nephrology	990
10	10	Henry	Cavill	3312321321	Dermatology	222

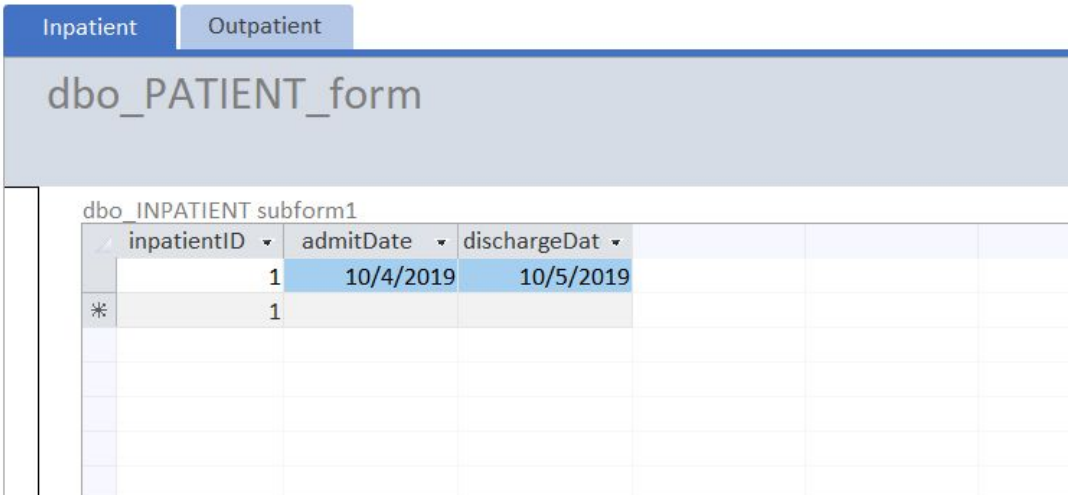
4. Register a Patient

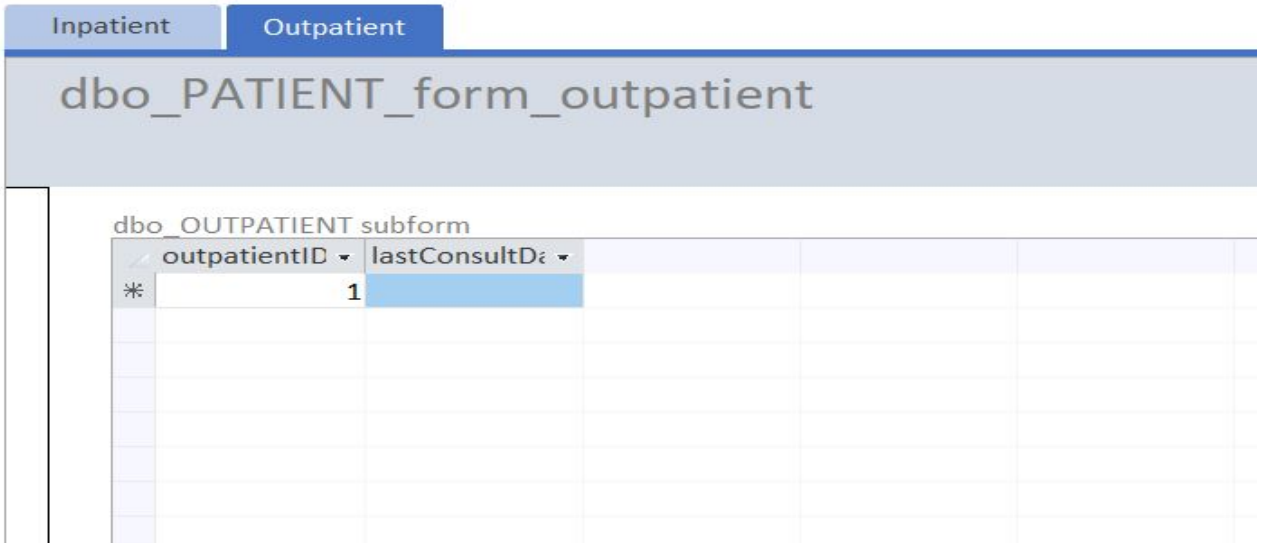
   Whenever a patient needs to book and appointment, first there should be a record of their details, and while also mentioning if the patient is an inpatient or an outpatient.

   So when the user fills out the patient forms, the user has to select from two different forms depending on whether the patient is an inpatient or an outpatient.



The different forms contain different subforms, which contain particular information regarding inpatient and outpatients.

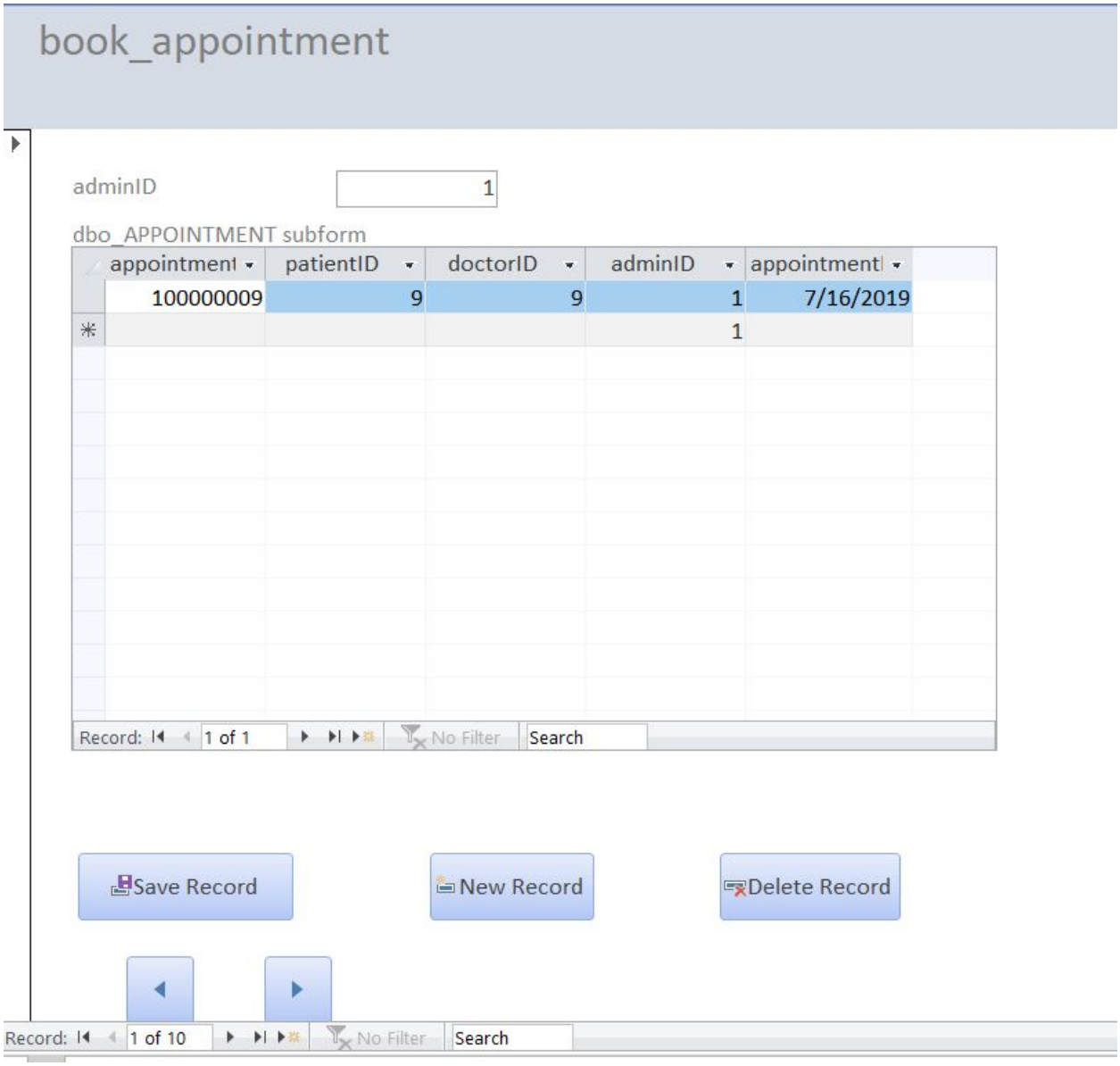




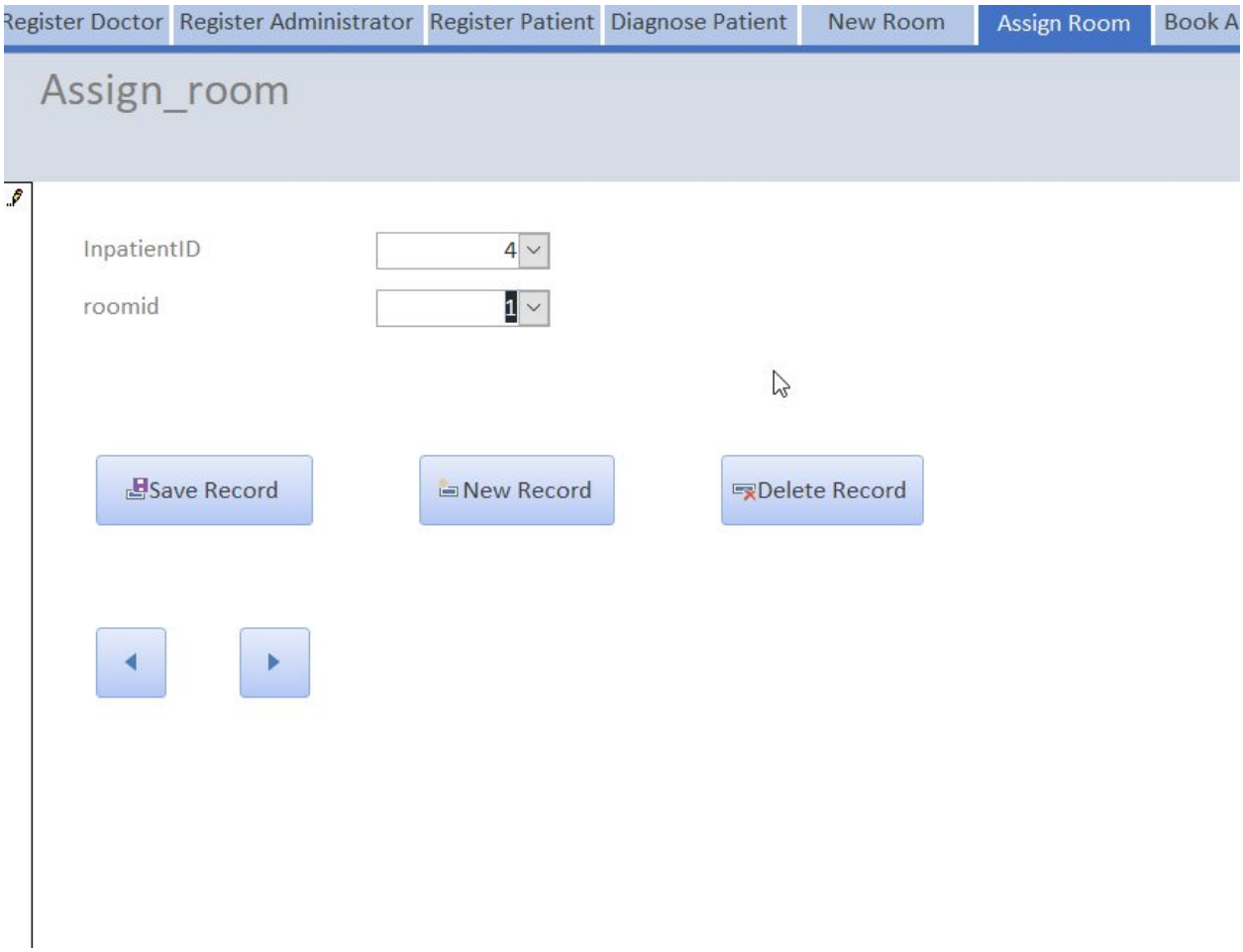
5. Book an Appointment

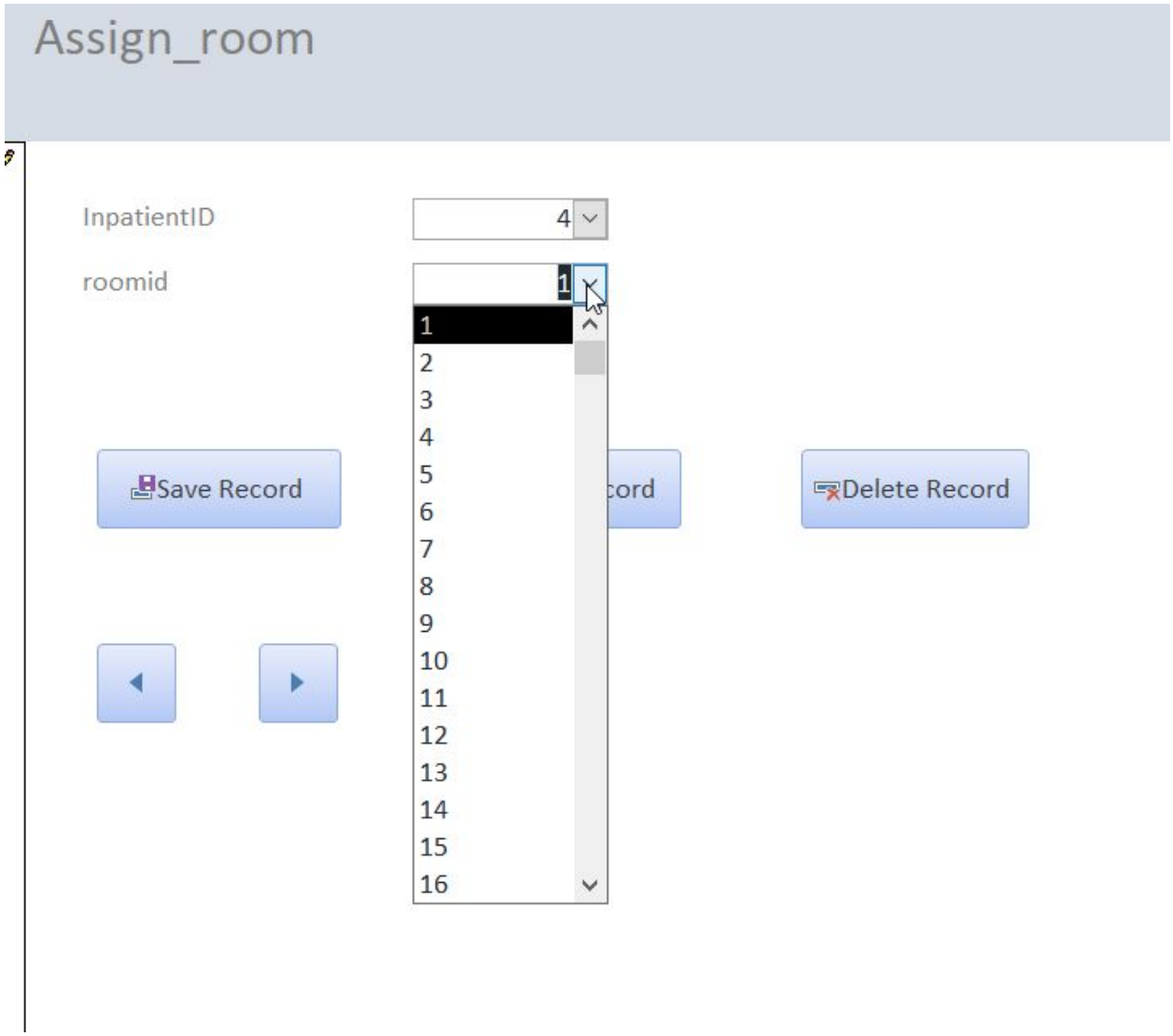
   The administrator can confirm the appointment of the patient, and then add the details into the database. So, first the admin can input the adminID, and all the appointments booked by that adminID will be displayed.

   Here the administrator can add the details of the appointment under that adminID.

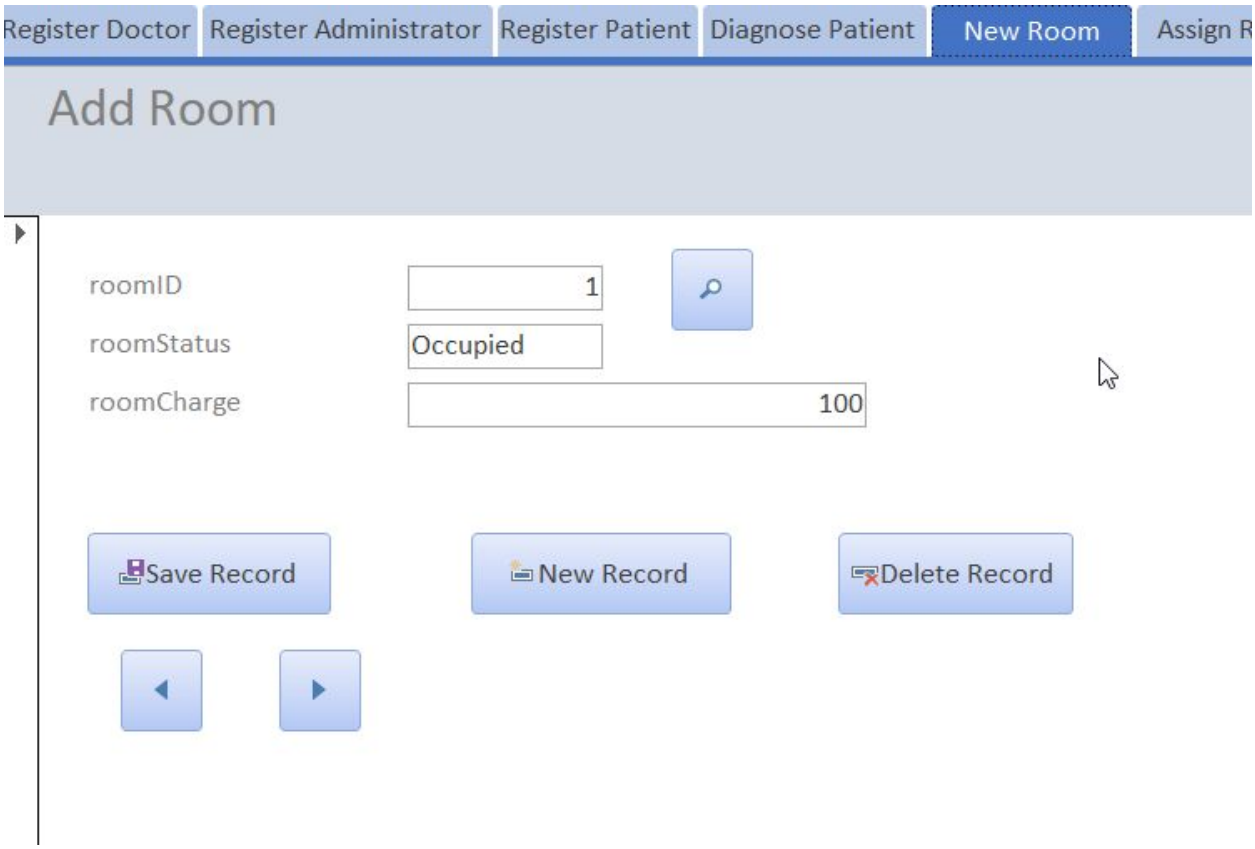


6. Assign a Room to a Patient
   The Administrator assigns room to the patient who are of Inpatient class in the database. The administrator has to select the inpatientID from the existing records of inpatients from a dropdown menu, and then select a Room by roomID dropdown from an existing list of rooms in the hospital.

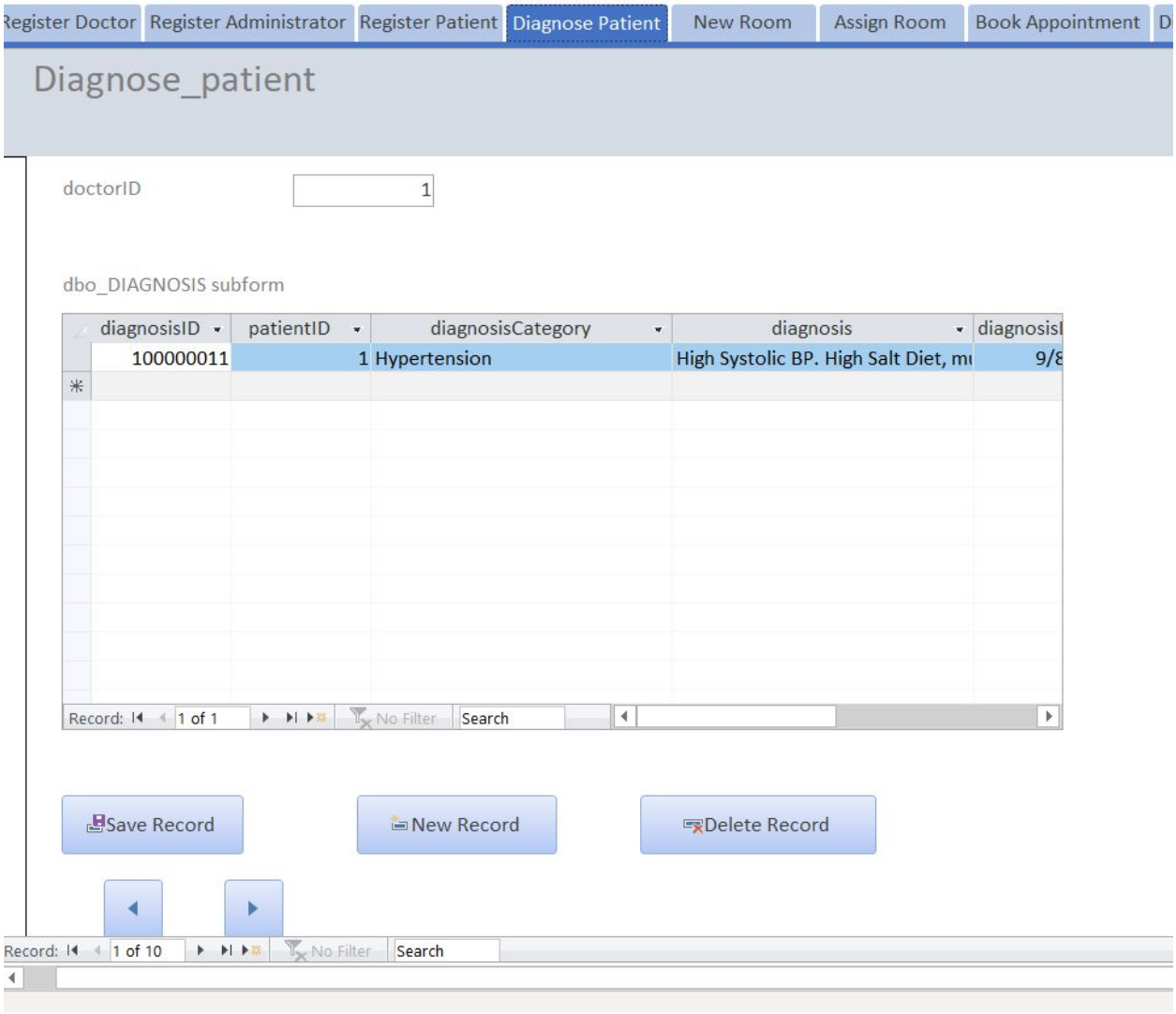



Assign_room
InpatientID
4
roomid
1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Save Record
cord
Delete Record

7. Register a New Room
   If there is ever the need to add a new Room for the patients, or to delete a Room from the system, then the administrator can use this Form to do that.

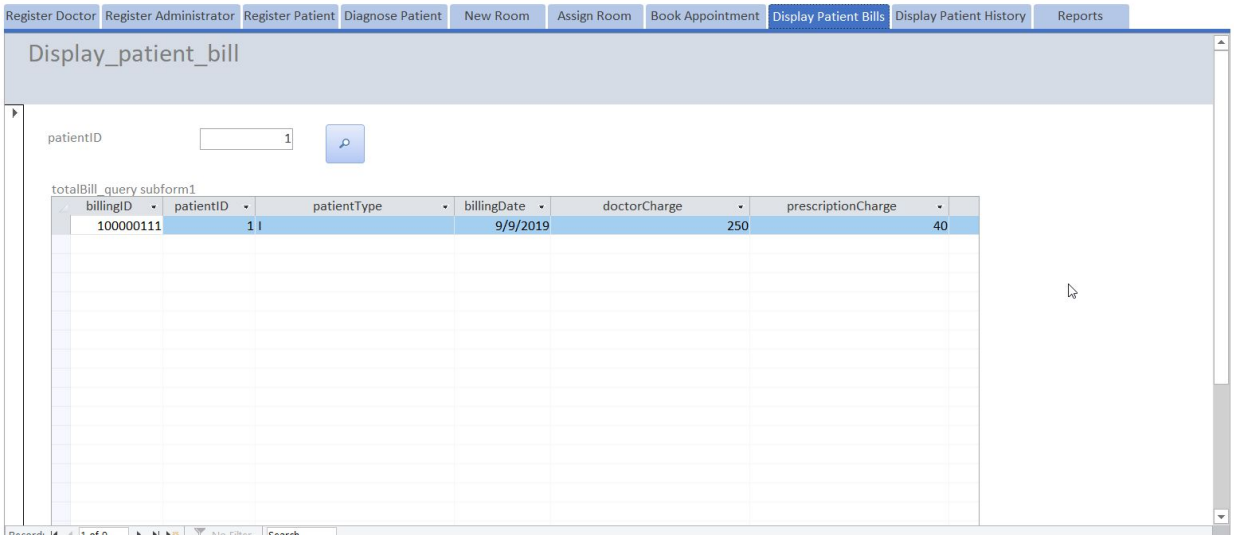


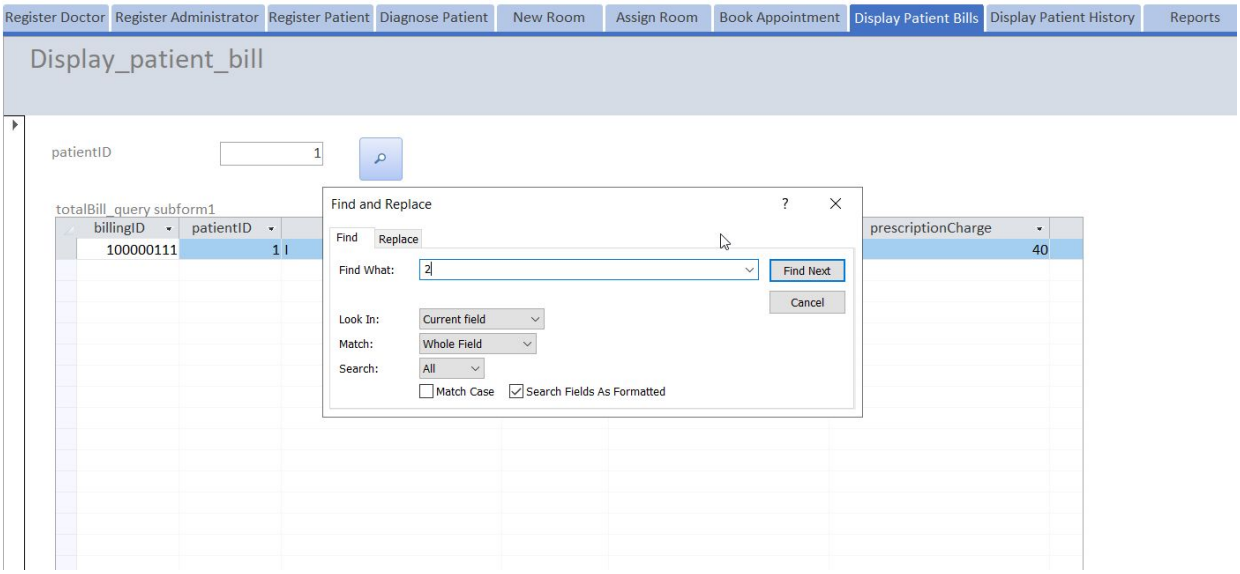
8. Diagnose Patient
   After the appointment between the doctor and the patient, the doctor needs to store a record of the diagnosis for the patient, for that diagnosis date. This is done through the Diagnose Patient Form. The doctor inputs the doctorID, and all of his/her patients will be displayed in a subform. Through this, the doctor can add a new diagnosis for a new or a recurring patient in the database.

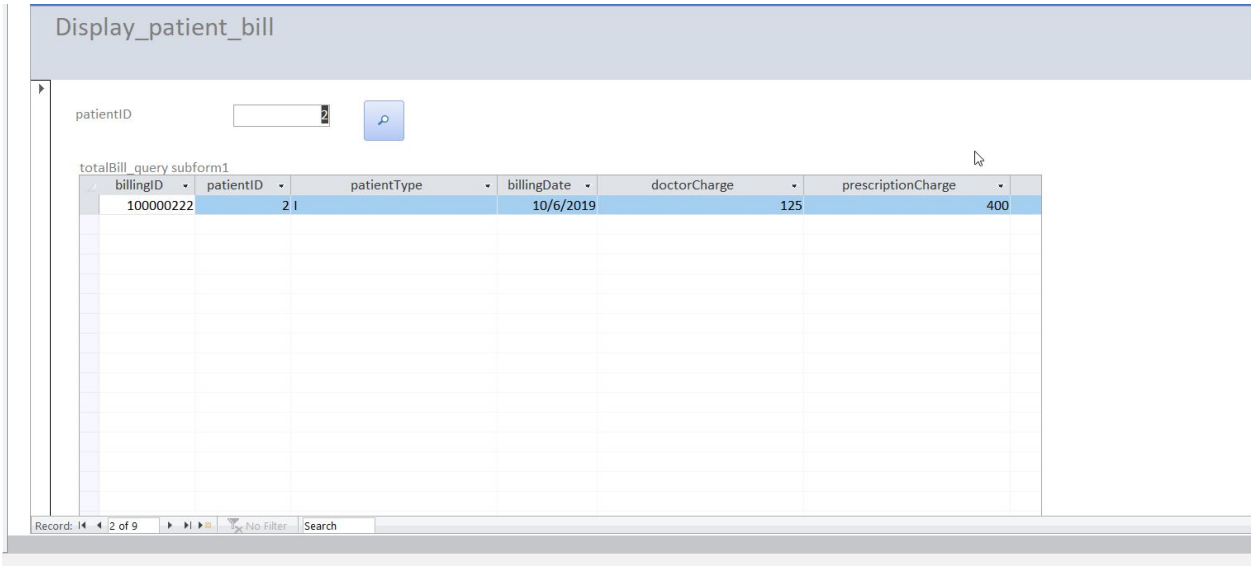


9. Create a Bill for the patient

   The billing information of the patient is stored in the BILLING form, and all the bills for a particular patient can be accessed by the users. Then the latest bill can be created through the billing subform.

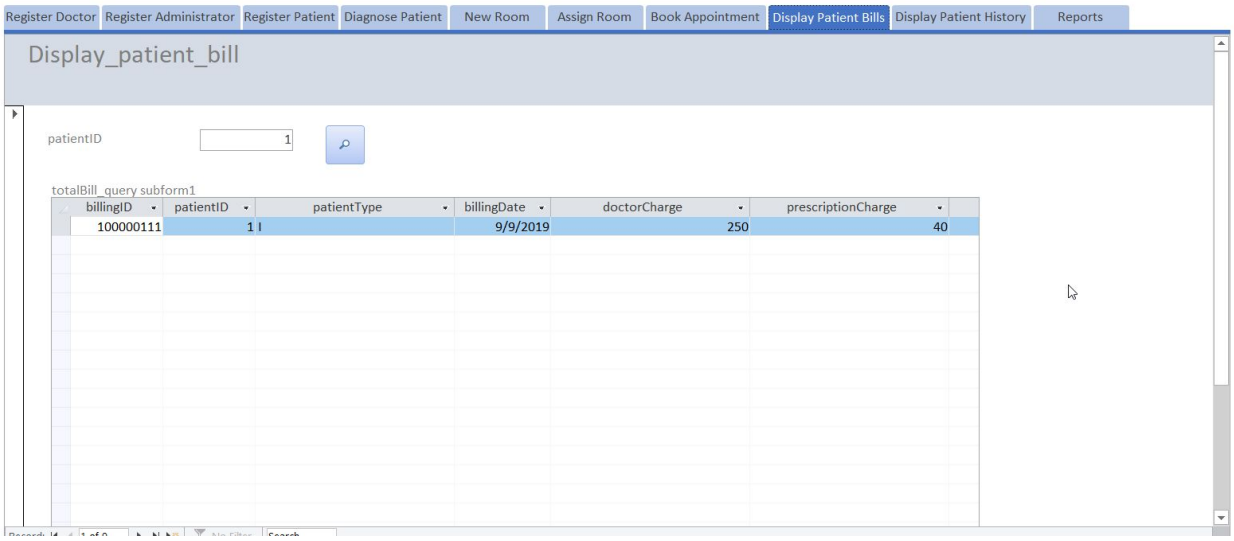

Register Doctor
Register Administrator
Register Patient
Diagnose Patient
New Room
Assign Room
Book Appointment
Display Patient Bills
Display Patient History
Reports
Display_patient_bill
patientID
1
totalBill_query subform1
billingID
patientID
patientType
billingDate
doctorCharge
prescriptionCharge
100000111
1
9/9/2019
250
40

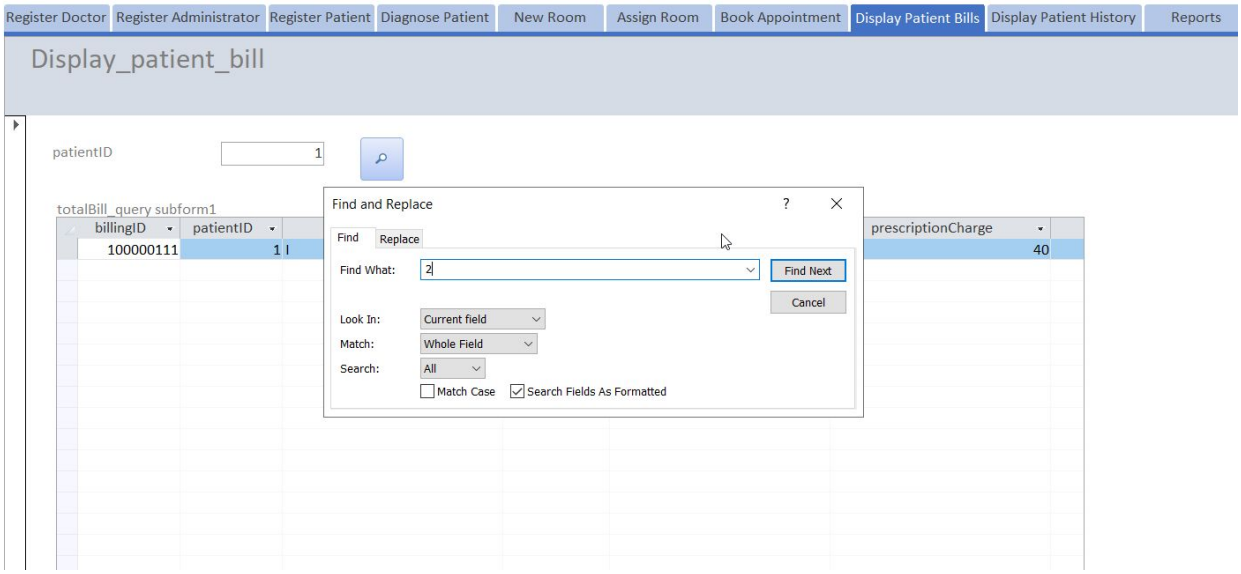

Register Doctor
Register Administrator
Register Patient
Diagnose Patient
New Room
Assign Room
Book Appointment
Display Patient Bills
Display Patient History
Reports
Display_patient_bill
patientID
1
totalBill_query subform1
billingID
patientID
prescriptionCharge
100000111
1
40
Find and Replace
Find
Replace
Find What:
2
Find Next
Cancel
Look In:
Current field
Match:
Whole Field
Search:
All
Match Case
Search Fields As Formatted

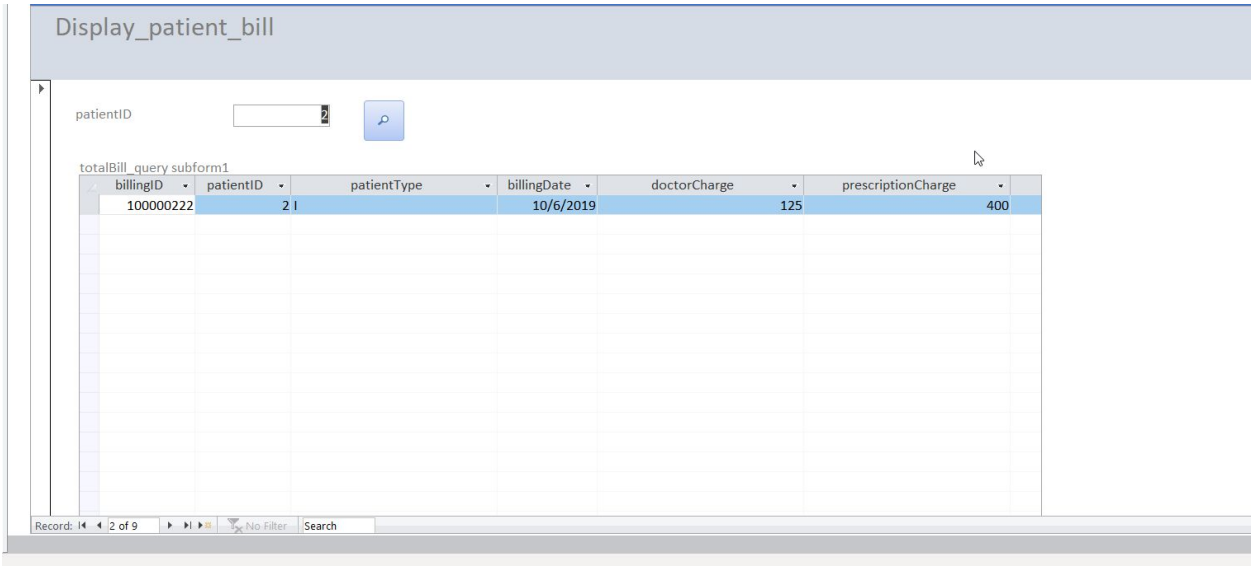


10. Display Patient Bill:

The patient can view the bills that are associated with that patientID.

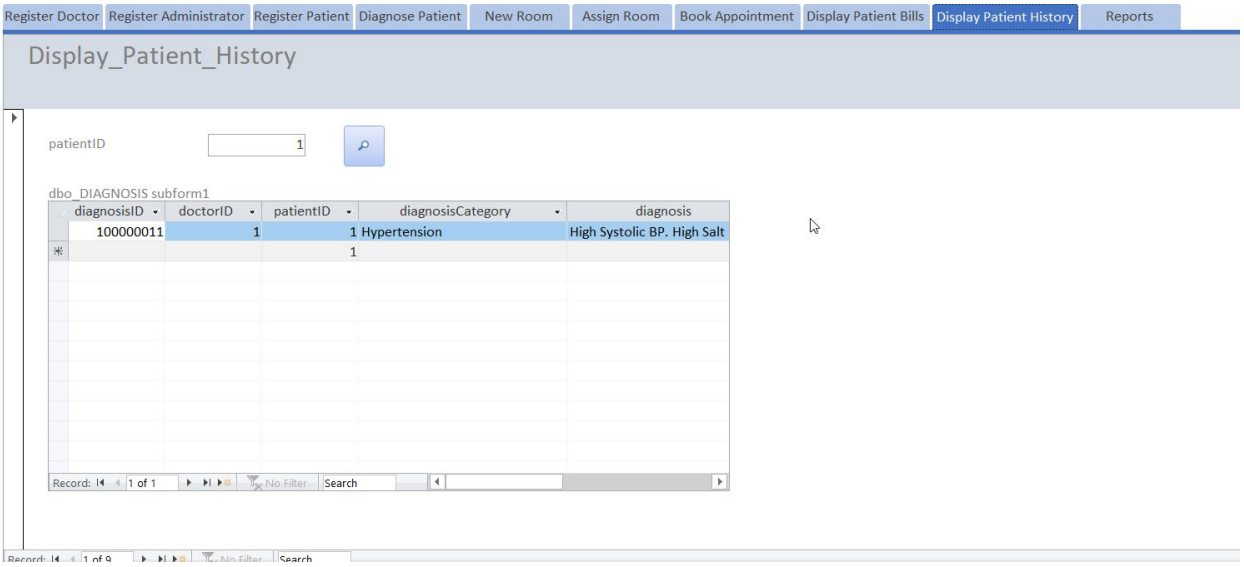


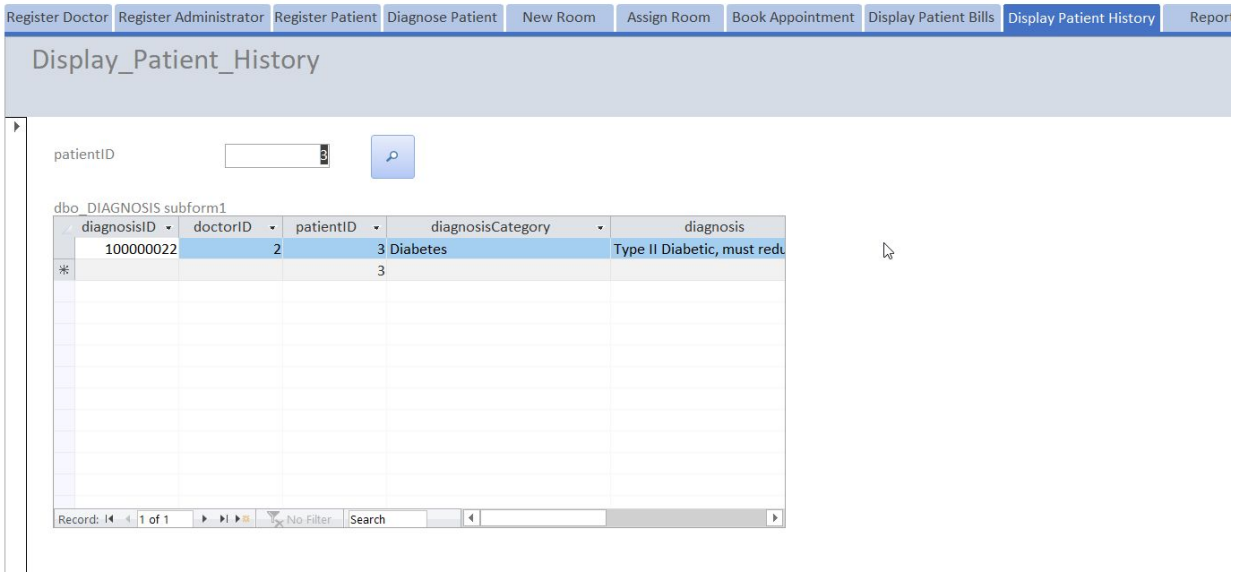

Register Doctor
Register Administrator
Register Patient
Diagnose Patient
New Room
Assign Room
Book Appointment
Display Patient Bills
Display Patient History
Reports
Display_patient_bill
patientID
1
totalBill_query subform1
billingID
patientID
prescriptionCharge
100000111
1
40
Find and Replace
Find
Replace
Find What:
2
Find Next
Cancel
Look In:
Current field
Match:
Whole Field
Search:
All
Match Case
Search Fields As Formatted

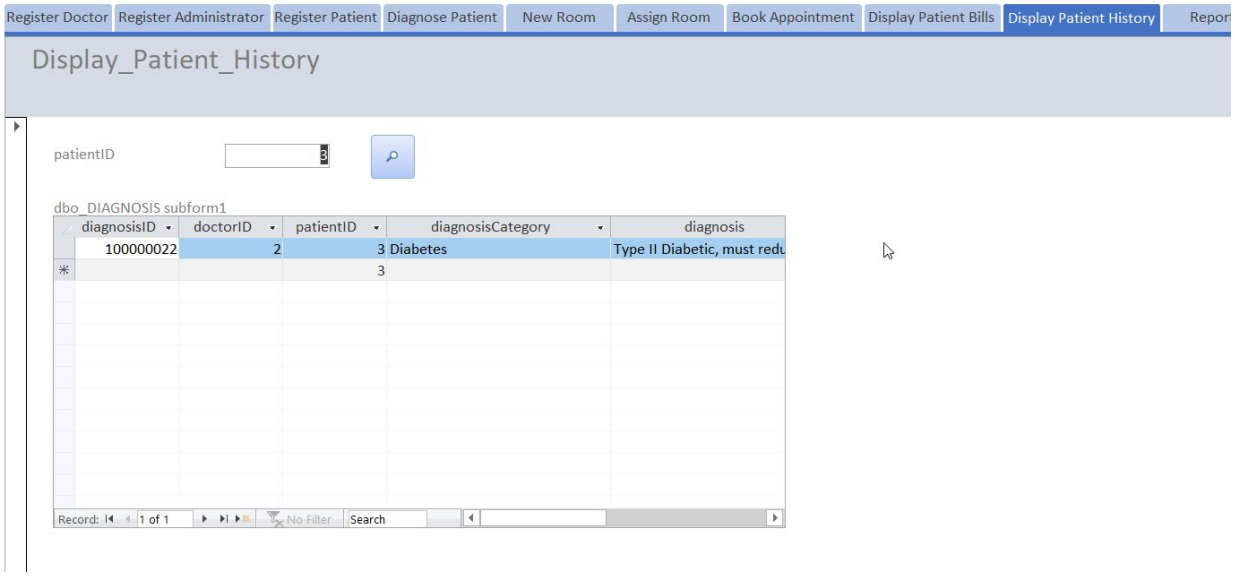

Display_patient_bill
patientID
2
totalBill_query subform1
billingID
patientID
patientType
billingDate
doctorCharge
prescriptionCharge
100000222
2
10/6/2019
125
400
Record:
2 of 9
No Filter
Search

11. Display Patient History

This can be used by the patients and doctors to view the past history of patients diseases

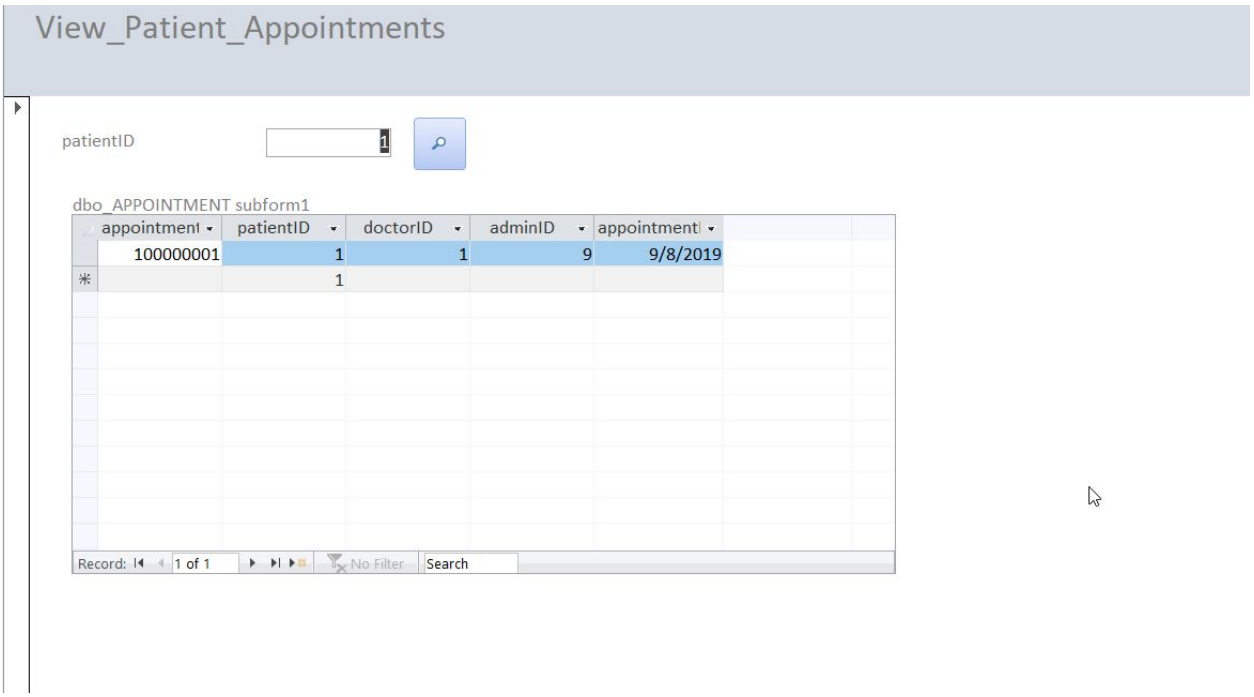


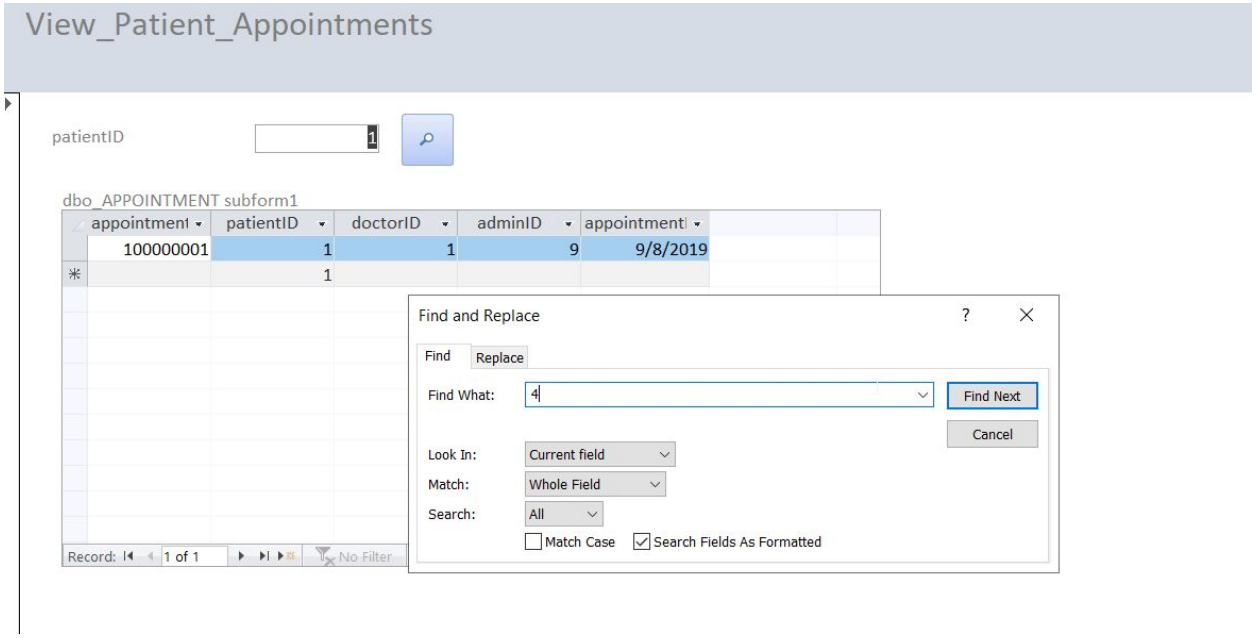


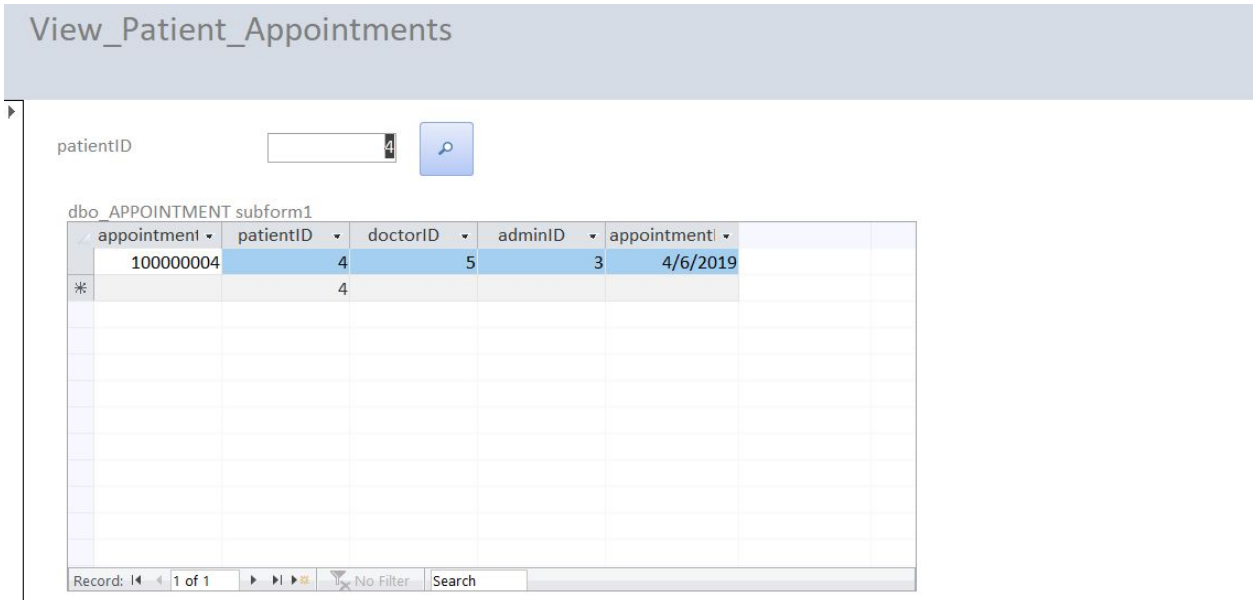


12. View Patient Appointments

This form can be used to display the appointments of the patients, by the patients and the doctors




View_Patient_Appointments
patientID
1
dbo_APPOINTMENT subform1
appointment
patientID
doctorID
adminID
appointment
100000001
1
1
9
9/8/2019
1
Record: 1 of 1
No Filter
Find and Replace
Find
Replace
Find What:
4
Find Next
Cancel
Look In:
Current field
Match:
Whole Field
Search:
All
Match Case
Search Fields As Formatted


View_Patient_Appointments
patientID
4
dbo_APPOINTMENT subform1
appointment
patientID
doctorID
adminID
appointment
100000004
4
5
3
4/6/2019
4
Record: 1 of 1
No Filter
Search

REPORTS:

1. Room Availability Report:
   This report shows the available and the occupied rooms in the hospital, in groups defined by the Status of the Room: i.e. Occupied and Vacant

dbo_ROOM_report

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

2. Invoice of Each Patient Report
   This report displays the invoice of every bill of the patients. It displays the individual amounts charged by the different aspects of the appointment/treatment, and also shows the Total Room Charge for all Inpatients, which is then all added to display the total Bill.
   The insurance covers some amount of the total bill, so the total bill displayed is the additional amount that is to be paid by the patient.

totalBill_query_report

billingID	patientID	patientType	billingDate	doctorCharge	prescriptionCha	roomCharge	NumberOfDays	talRoomCharge	insuranceCover	TotalBill
100000111	1	I	9/9/2019	250	40	0	1	0	80	58
100000222	2	I	10/6/2019	125	400	100	1	100	60	250
100000333	3	I	10/6/2019	300	80	0	1	0	40	228
100000444	4	I	4/8/2019	530	0	0	1	0	10	477
100000555	5	I	8/18/2019	260	90	100	1	100	0	450
100000666	6	O	5/13/2019	120	0				55	54
100000777	7	O	11/12/2019	1000	15				100	0
100000888	8	O	11/11/2019	400	0				15	340
100000999	9	O	11/11/2019	550	100				45	357.5

Tuesday, December 3, 2019 Page 1 of 1

3. Maximum Bill Amount paid by patient

Maximum_bill_report

patientType	TotalBill	billingID	patientID
I			
	477	100000444	4
	450	100000555	5
	250	100000222	2
	228	100000333	3
	58	100000111	1
O			
	357.5	100000999	9
	340	100000888	8
	54	100000666	6
	0	100000777	7

Tuesday, December 3, 2019 Page 1 of 1

This report displays the Bill amount for each patient type, arranged in descending order of Total Bill Amount paid by the patient.

4. Experienced Doctors Report
   The report displays the list of doctors in descending order of Years of Practice

ExperienceDoctor_report1

doctorID	firstName	lastName	yearsOfPractice	departmentName
8	Jeffrey	Carpenter	23	Pharmacology
1	Susan	Grey	10	General Internal Medicine
6	Phil	Kinsella	9	Endocrinology
7	Patricia	Smith	8	Pulmonology
4	Beth	Rettinger	7	Oncology
3	John	Noble	4	Dermatology
2	Chris	Billinson	2	Cardiology
5	Amy	Cote	1	Gastroenterology
9	Amanda	Shock	1	Nephrology

Tuesday, December 3, 2019 Page 1 of 1

5. Number of Doctors per Department

NumberOfDoctors_report

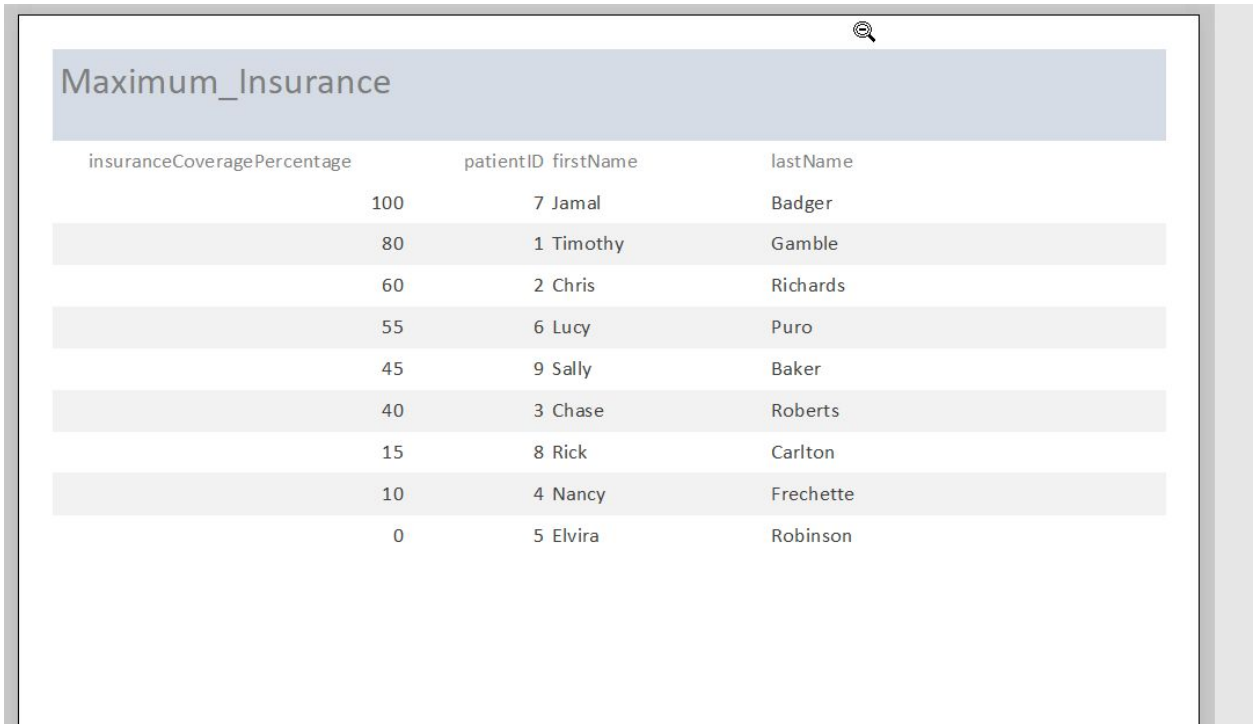
departmentName	CountOfdoctor
Cardiology	1
Dermatology	1
Endocrinology	1
Gastroenterology	1
General Internal Medicine	1
Nephrology	1
Oncology	1
Pharmacology	1
Pulmonology	1

Tuesday, December 3, 2019 Page 1 of 1

The report displays the number of doctors per department.

6. Maximum Insurance Coverage

This report displays the patients with maximum insurance coverage percentage by arranging the records in descending order of insurance Coverage Percentage



Maximum_Insurance

insuranceCoveragePercentage	patientID	firstName	lastName
100	7	Jamal	Badger
80	1	Timothy	Gamble
60	2	Chris	Richards
55	6	Lucy	Puro
45	9	Sally	Baker
40	3	Chase	Roberts
15	8	Rick	Carlton
10	4	Nancy	Frechette
0	5	Elvira	Robinson

7. **Which diseases are most common?**

This report shows the count of diseases based on the disease category to figure out the most common diseases that the hospital needs to be prepared for.

Common_diagnosis_query_report

CountOfdiagnosisID	diagnosisCategory
2	Hypertension
1	Hypothyroidism
1	Obesity
1	Osteoarthritis
1	Acute bronchitis
1	Allergic rhinitis
1	Anxiety
1	Back Pain
1	Diabetes