HOSPITAL DATABASE SYSTEM

PROJECT REPORT

SUBMITTED TO:

PROF. SONIKA DAHIYA

SUBMITTED BY:

CHETAN PRAKASH(2K18/EE/057)

DEEPAK KUMAR(2K18/EE/060)

Table of Content

INTRODUCTION

Hospitals play a very significant role in our lives as they provide the medical facilities in order to reduce the suffering of the people who are facing various types of illness. The illness may happen because of ambient atmospheric conditions, increment in work pressure, psychological issues due to emotional stress. Keeping the record of the daily activities of the hospital is a tedious task manually, that's why a proficient database system is required to keep a record of all such activities.

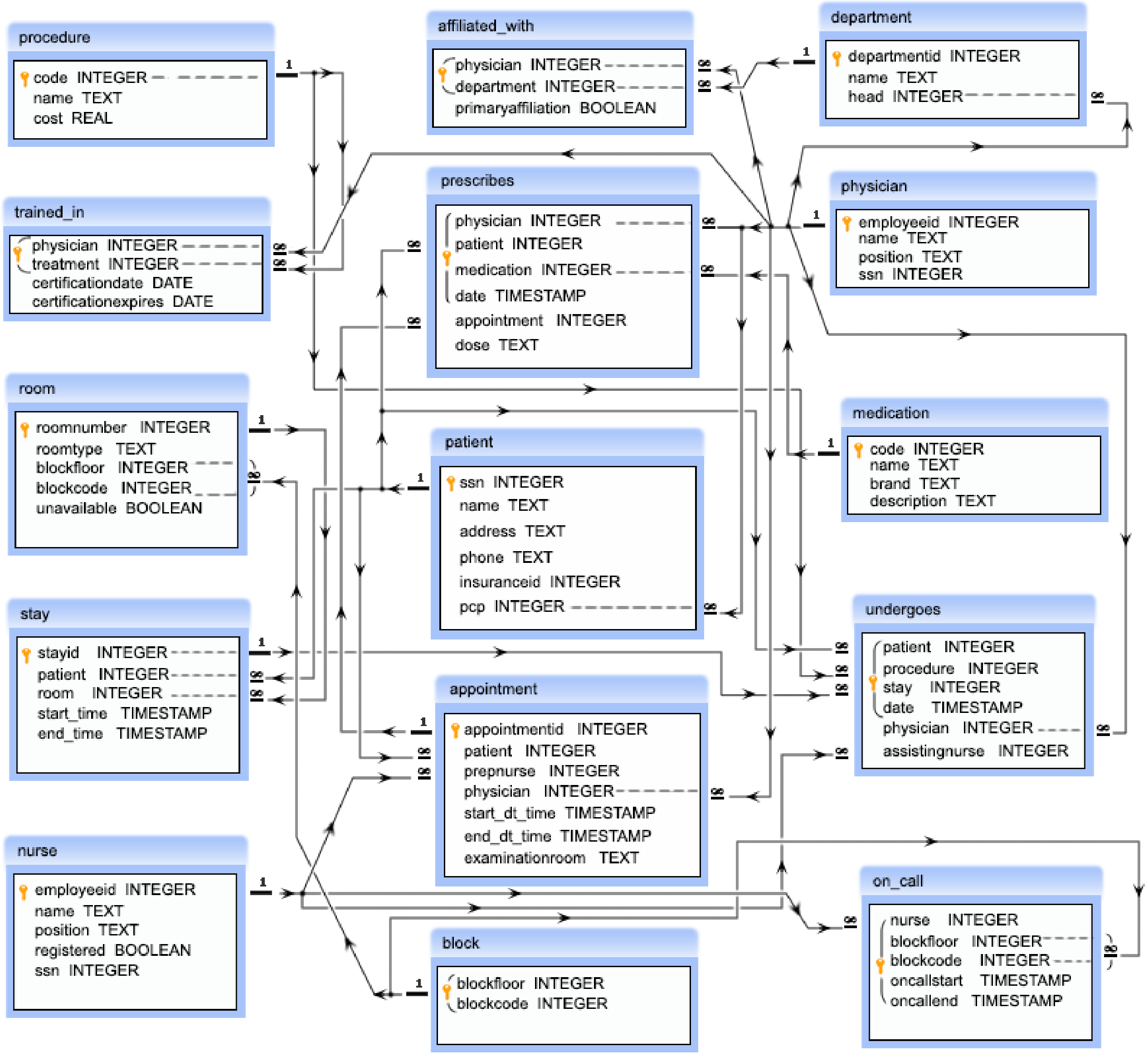
List of tables in hospital database system

- physician
- department
- affiliated_with
- procedure
- trained_in
- patient
- nurse
- appointment
- medication
- prescribes
- block
- room
- on_call
- stay
- undergoes

Objective:

- To develop a system that stores the record of all the clinical staff and their activities related to Hospital, i.e. Patients, Treatments, Nurses, Rooms, Doctors, Training etc.
- To understand the concept of Database development, running operations on them and displaying the information deemed necessary by the operator.
- Develop the understanding for Relation schemas, defining constraints in a table, inserting and updating the records in a table as per the user requirements.

Relational Schema:



WORKING OF TABLES:

physician:

- employeeid – this will be a singular ID of a physician
- name – this will be the name of a physician
- position – this will be the designation of a physician

- ssn – this will be a security no. of a physician

department:

- departmentid – this will be a singular ID for a department

- name – this will be the name of a department

- head – this will be the ID of physician who is the head of a department, referencing to the column employeeid of the table doc

affiliated_with:

- physician – this will be the ID of the physicians that is referencing to the column employeeid of the physician table

- department – this will be the ID the department that is referencing to the column departmentid of the department table

- primaryaffiliation – this will be a logical column that indicate that whether or not the physicians are nevertheless to be related to or not

- **Note: The combinations of people working as physicians, department will be coming once in the table.**

procedure:

- code – this will be the distinctive ID of a procedure

- name – the name of the procedure

- cost – the value for the procedure

trained_in:

- physician – this will be ID of the physicians that is referencing to the column employeeid of the physician table

- treatment – this will be the ID of the procedure that is referencing to the column code of the procedure table

- certificationdate – this will be the beginning date of certification

- certificationexpires – this will be the end date of certification

- **Note: The combination of people working as physicians and treatment will be coming once in the table.**

patient:

- ssn – this will be a singular ID for every patient

- name – this will be the name of the patient

- address – this will be the address of the patient

- phone – this will be the contact no. of the patient

- insuranceid – this will be the insurance id of the patient

- pcp – this will be the ID of the physician who primarily checked up the patient that is referencing to the column employeeid of the physician table

nurse:

- employeeid – this will be the distinctive ID for a nurse

- name – name of the nurses

- position – the designation of the nurses

- registered – this will be a logical column that indicate that whether or not the nurses are registered for nursing or not

- ssn – this will be the security number allotted to a nurse

appointment:

- appointmentid – this will be the distinctive ID for an appointment

- patient – this will be the ID of every patient that is referencing to the ssn column of patient table

- prepnurse – the ID of the nurse who might attend the patient with the physician, that is referencing to the column employeeid of the nurse table

- physician – this will be the ID the physicians that is referencing to the employeeid column of the physician table

- start_dt_time – this will be the schedule date and approximate time to have a meeting with the physician

- end_dt_time – this will be the schedule date and approximate time to finish the meeting

• examinationroom – this is the place to have a meeting between a patient and a physician

medication:

• code – this will be the distinctive ID for a drugs

• name – this will be the name of the medication

• brand – this will be the company of the medication

• description – this will be the outline of the medication

prescribes:

• physician – this will be the ID of the physician referencing to the employeeid column of the physician table

• patient – this will be the ID of the patient that is referencing to the ssn column of the patient table

• medication – the ID of the medication that is referencing to the code of the medication table

• date – the date and time of the prescribed medication

• appointment – the prescription created by the physician to a patient who might taken a appointment that is referencing to column appointmentid of appointment table

• dose – the dose prescribed

• **Note: the mix of doc, patient, medication, date will be coming once in the table.**

block:

• blockfloor – ID designated to the floor

• blockcode - ID designated to the block

• **Note: The combination of blockfloor, blockcode will be coming once in the table.**

room:

• roomnumber – this will be the distinctive ID of a space

• roomtype – this will be category of room

• blockfloor - this will be the ground ID wherever the space in

- blockcode – this will be the ID of the block wherever the space in

- unavailable – this will be the logical column that indicate that whether or not the space is offered or not

- **Note: The of blockfloor, blockcode columns ar refercing to the combination of blockfloor and blockcode columns of the table block.**

on_call:

- nurse – this will be ID of the nurse that is referencing to the employeeid column of the table nurse

- blockfloor - this will be the ID of the ground

- blockcode – this will be the ID of block

- oncallstart - the beginning date and time of on call period

- oncallend – the ending date and time of on call period

- **Note: The mix of nurse, blockfloor, blockcode, oncallstart, oncallend will be coming once in the table and therefore the combination of blockfloor, blockcode columns are referring to the combination of blockfloor and blockcode columns of the table block .**

stay:

- stayid - this will be distinctive ID for the admission

- patient – this will be the ID of the patient that is referencing the ssn column of patient table

- area - this will be the ID of the space wherever the patient admitted and that is referencing to the spacenumber column of the room table

- start_time – this will be the time once a patient admitted

- end_time – this will be the time how much long a patient is staying

undergoes:

- patient - this will be ID of the patient that is referencing to the ssn column of the patient table

- procedure – this will be ID of the procedure and referencing to the code column of the procedure table

- stay - this will be the ID admission of a patient, that is referencing to the keepid column of the stay table

- date – this will be the date once a patient undergoes for a procedure

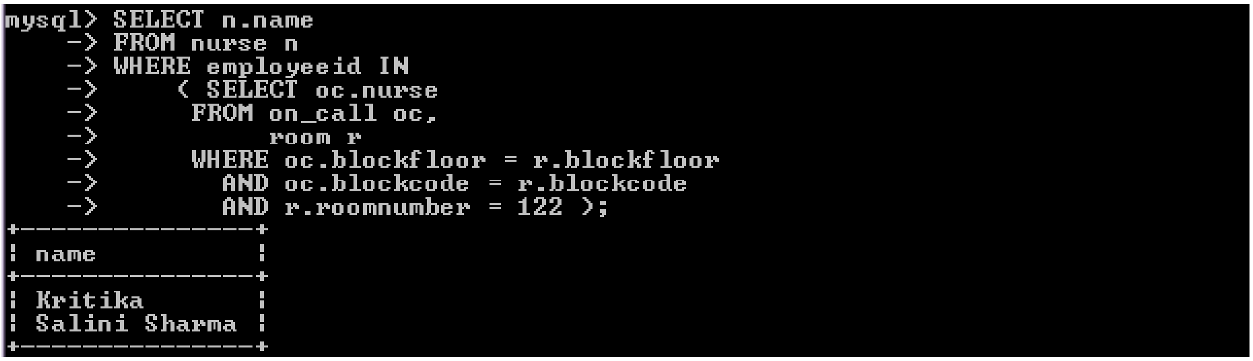
- physician – this will be the ID of a physician that is referencing to the column employeeid of the table physician

- assistingnurse – this will be the ID of a nurse who will be assisting, referencing to the column employeeid of the table nurse

- **Note: The combination ofpatient, procedure, stay, date will be coming once in the table.**

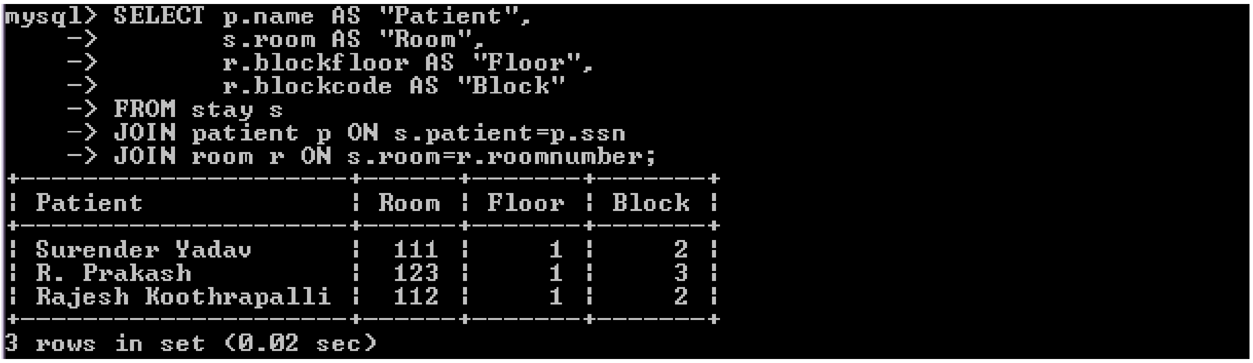
Some Example Queries:

1. Write a query in SQL to obtain the names of all the nurses who have ever been on call for room 122.



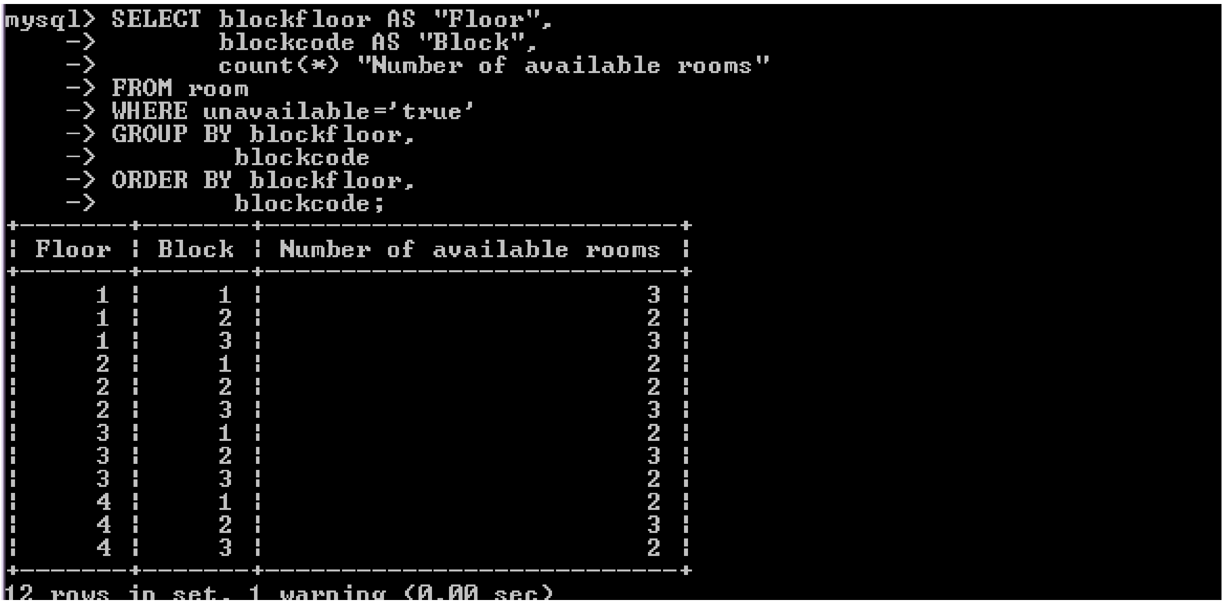
```
mysql> SELECT n.name
    -> FROM nurse n
    -> WHERE employeeid IN
    ->     ( SELECT oc.nurse
    ->      FROM on_call oc,
    ->           room r
    ->      WHERE oc.blockfloor = r.blockfloor
    ->        AND oc.blockcode = r.blockcode
    ->        AND r.roomnumber = 122 );
+----------------+
| name           |
+----------------+
| Kritika        |
| Salini Sharma  |
+----------------+
```

2.Write a query in SQL to obtain the name of the patients, their block, floor, and room number where they are admitted.



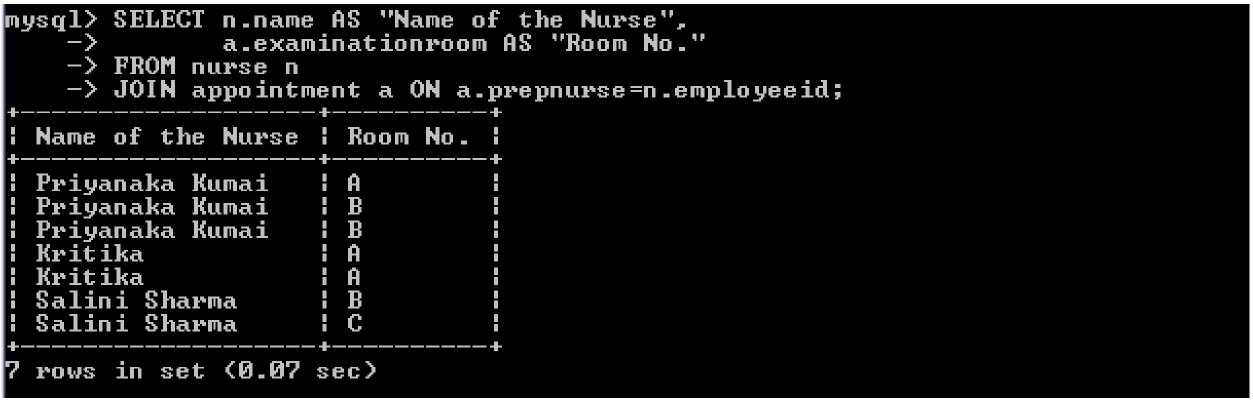
```
mysql> SELECT p.name AS "Patient",
    ->        s.room AS "Room",
    ->        r.blockfloor AS "Floor",
    ->        r.blockcode AS "Block"
    -> FROM stay s
    -> JOIN patient p ON s.patient=p.ssn
    -> JOIN room r ON s.room=r.roomnumber;
+---------------------+------+-------+-------+
| Patient             | Room | Floor | Block |
+---------------------+------+-------+-------+
| Surender Yadav      |  111 |     1 |     2 |
| R. Prakash          |  123 |     1 |     3 |
| Rajesh Koothrapalli |  112 |     1 |     2 |
+---------------------+------+-------+-------+
3 rows in set (0.02 sec)
```

3. Write a query in SQL to count the number of unavailable rooms for each block in each floor.



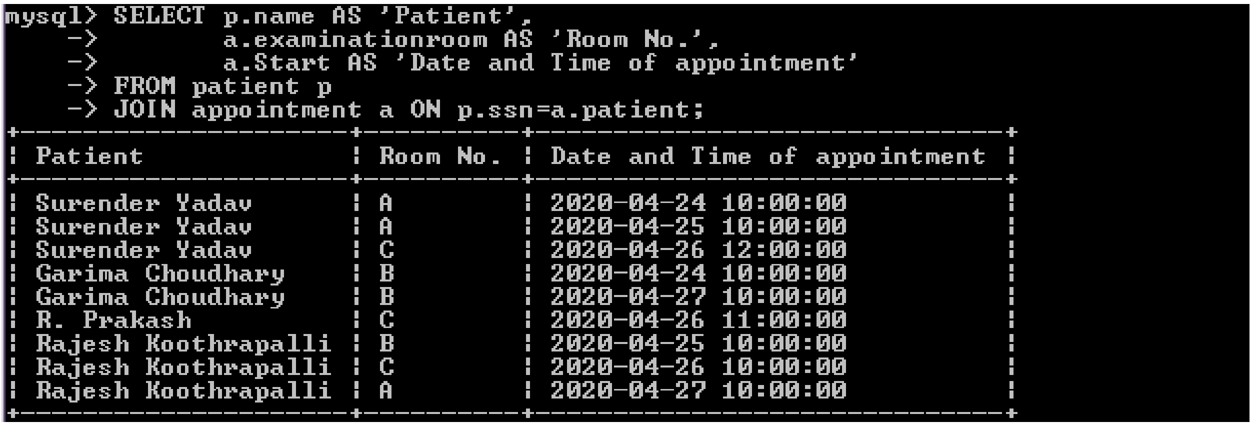
```
mysql> SELECT blockfloor AS "Floor",
    ->        blockcode AS "Block",
    ->        count(*) "Number of available rooms"
    -> FROM room
    -> WHERE unavailable='true'
    -> GROUP BY blockfloor,
    ->          blockcode
    -> ORDER BY blockfloor,
    ->          blockcode;
+-------+-------+---------------------------+
| Floor | Block | Number of available rooms |
+-------+-------+---------------------------+
|     1 |     1 |                         3 |
|     1 |     2 |                         2 |
|     1 |     3 |                         3 |
|     2 |     1 |                         2 |
|     2 |     2 |                         2 |
|     2 |     3 |                         3 |
|     3 |     1 |                         2 |
|     3 |     2 |                         3 |
|     3 |     3 |                         2 |
|     4 |     1 |                         2 |
|     4 |     2 |                         3 |
|     4 |     3 |                         2 |
+-------+-------+---------------------------+
12 rows in set, 1 warning (0.00 sec)
```

4. Write a query in SQL to find the name of the nurses and the room scheduled, where they will assist the physicians.



```
mysql> SELECT n.name AS "Name of the Nurse",
    ->        a.examinationroom AS "Room No."
    -> FROM nurse n
    -> JOIN appointment a ON a.prepnurse=n.employeeid;
+-------------------+----------+
| Name of the Nurse | Room No. |
+-------------------+----------+
| Priyanaka Kumai   | A        |
| Priyanaka Kumai   | B        |
| Priyanaka Kumai   | B        |
| Kritika           | A        |
| Kritika           | A        |
| Salini Sharma     | B        |
| Salini Sharma     | C        |
+-------------------+----------+
7 rows in set (0.07 sec)
```

5.Write a query in SQL to find the name of the patients and the number of the room where they have to go for their treatment.



```
mysql> SELECT p.name AS 'Patient',
    ->        a.examinationroom AS 'Room No.',
    ->        a.Start AS 'Date and Time of appointment'
    -> FROM patient p
    -> JOIN appointment a ON p.ssn=a.patient;
+---------------------+----------+------------------------------+
| Patient             | Room No. | Date and Time of appointment |
+---------------------+----------+------------------------------+
| Surender Yadav      | A        | 2020-04-24 10:00:00          |
| Surender Yadav      | A        | 2020-04-25 10:00:00          |
| Surender Yadav      | C        | 2020-04-26 12:00:00          |
| Garima Choudhary    | B        | 2020-04-24 10:00:00          |
| Garima Choudhary    | B        | 2020-04-27 10:00:00          |
| R. Prakash          | C        | 2020-04-26 11:00:00          |
| Rajesh Koothrapalli | B        | 2020-04-25 10:00:00          |
| Rajesh Koothrapalli | C        | 2020-04-26 10:00:00          |
| Rajesh Koothrapalli | A        | 2020-04-27 10:00:00          |
+---------------------+----------+------------------------------+
```

Source Code:

```
create database hospital_project;
use hospital_project;
CREATE TABLE Physician (
  EmployeeID INTEGER NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Position VARCHAR(30) NOT NULL,
  SSN INTEGER NOT NULL,
  CONSTRAINT pk_physician PRIMARY KEY(EmployeeID)
);

DROP TABLE IF EXISTS Department;
CREATE TABLE Department (
  DepartmentID INTEGER NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Head INTEGER NOT NULL,
  CONSTRAINT pk_Department PRIMARY KEY(DepartmentID),
  CONSTRAINT fk_Department_Physician_EmployeeID FOREIGN KEY(Head) REFERENCES Physician(EmployeeID)
);


DROP TABLE IF EXISTS Affiliated_With;
CREATE TABLE Affiliated_With (
  Physician INTEGER NOT NULL,
  Department INTEGER NOT NULL,
  PrimaryAffiliation BOOLEAN NOT NULL,
  CONSTRAINT fk_Affiliated_With_Physician_EmployeeID FOREIGN KEY(Physician) REFERENCES Physician(EmployeeID),
  CONSTRAINT fk_Affiliated_With_Department_DepartmentID FOREIGN KEY(Department) REFERENCES
Department(DepartmentID),
  PRIMARY KEY(Physician, Department)
);

DROP TABLE IF EXISTS Procedures;
CREATE TABLE Procedures (
  Code INTEGER PRIMARY KEY NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Cost REAL NOT NULL
);

DROP TABLE IF EXISTS Trained_In;
CREATE TABLE Trained_In (
  Physician INTEGER NOT NULL,
  Treatment INTEGER NOT NULL,
  CertificationDate DATETIME NOT NULL,
  CertificationExpires DATETIME NOT NULL,
  CONSTRAINT fk_Trained_In_Physician_EmployeeID FOREIGN KEY(Physician) REFERENCES Physician(EmployeeID),
  CONSTRAINT fk_Trained_In_Procedures_Code FOREIGN KEY(Treatment) REFERENCES Procedures(Code),
  PRIMARY KEY(Physician, Treatment)
);
```

```
DROP TABLE IF EXISTS Patient;
CREATE TABLE Patient (
  SSN INTEGER PRIMARY KEY NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Address VARCHAR(30) NOT NULL,
  Phone VARCHAR(30) NOT NULL,
  InsuranceID INTEGER NOT NULL,
  PCP INTEGER NOT NULL,
  CONSTRAINT fk_Patient_Physician_EmployeeID FOREIGN KEY(PCP) REFERENCES Physician(EmployeeID)
);

DROP TABLE IF EXISTS Nurse;
CREATE TABLE Nurse (
  EmployeeID INTEGER PRIMARY KEY NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Position VARCHAR(30) NOT NULL,
  Registered BOOLEAN NOT NULL,
  SSN INTEGER NOT NULL
);

DROP TABLE IF EXISTS Appointment;
CREATE TABLE Appointment (
  AppointmentID INTEGER PRIMARY KEY NOT NULL,
  Patient INTEGER NOT NULL,
  PrepNurse INTEGER,
  Physician INTEGER NOT NULL,
  Start DATETIME NOT NULL,
  End DATETIME NOT NULL,
  ExaminationRoom TEXT NOT NULL,
  CONSTRAINT fk_Appointment_Patient_SSN FOREIGN KEY(Patient) REFERENCES Patient(SSN),
  CONSTRAINT fk_Appointment_Nurse_EmployeeID FOREIGN KEY(PrepNurse) REFERENCES Nurse(EmployeeID),
  CONSTRAINT fk_Appointment_Physician_EmployeeID FOREIGN KEY(Physician) REFERENCES Physician(EmployeeID)
);

DROP TABLE IF EXISTS Medication;
CREATE TABLE Medication (
  Code INTEGER PRIMARY KEY NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Brand VARCHAR(30) NOT NULL,
  Description VARCHAR(30) NOT NULL
);


DROP TABLE IF EXISTS Prescribes;
CREATE TABLE Prescribes (
  Physician INTEGER NOT NULL,
  Patient INTEGER NOT NULL,
  Medication INTEGER NOT NULL,
  Date DATETIME NOT NULL,
  Appointment INTEGER,
  Dose VARCHAR(30) NOT NULL,
  PRIMARY KEY(Physician, Patient, Medication, Date),
  CONSTRAINT fk_Prescribes_Physician_EmployeeID FOREIGN KEY(Physician) REFERENCES Physician(EmployeeID),
  CONSTRAINT fk_Prescribes_Patient_SSN FOREIGN KEY(Patient) REFERENCES Patient(SSN),
  CONSTRAINT fk_Prescribes_Medication_Code FOREIGN KEY(Medication) REFERENCES Medication(Code),
  CONSTRAINT fk_Prescribes_Appointment_AppointmentID FOREIGN KEY(Appointment) REFERENCES
Appointment(AppointmentID)
);

DROP TABLE IF EXISTS Block;
CREATE TABLE Block (
  BlockFloor INTEGER NOT NULL,
  BlockCode INTEGER NOT NULL,
  PRIMARY KEY(BlockFloor, BlockCode)
);

DROP TABLE IF EXISTS Room;
CREATE TABLE Room (
  RoomNumber INTEGER PRIMARY KEY NOT NULL,
  RoomType VARCHAR(30) NOT NULL,
  BlockFloor INTEGER NOT NULL,
  BlockCode INTEGER NOT NULL,
  Unavailable BOOLEAN NOT NULL,
  CONSTRAINT fk_Room_Block_PK FOREIGN KEY(BlockFloor, BlockCode) REFERENCES Block(BlockFloor, BlockCode)
);

DROP TABLE IF EXISTS On_Call;
CREATE TABLE On_Call (
  Nurse INTEGER NOT NULL,
  BlockFloor INTEGER NOT NULL,
  BlockCode INTEGER NOT NULL,
  OnCallStart DATETIME NOT NULL,
  OnCallEnd DATETIME NOT NULL,
  PRIMARY KEY(Nurse, BlockFloor, BlockCode, OnCallStart, OnCallEnd),
  CONSTRAINT fk_OnCall_Nurse_EmployeeID FOREIGN KEY(Nurse) REFERENCES Nurse(EmployeeID),
  CONSTRAINT fk_OnCall_Block_Floor FOREIGN KEY(BlockFloor, BlockCode) REFERENCES Block(BlockFloor, BlockCode)
);

DROP TABLE IF EXISTS Stay;
CREATE TABLE Stay (
  StayID INTEGER PRIMARY KEY NOT NULL,
  Patient INTEGER NOT NULL,
  Room INTEGER NOT NULL,
  StayStart DATETIME NOT NULL,
  StayEnd DATETIME NOT NULL,
  CONSTRAINT fk_Stay_Patient_SSN FOREIGN KEY(Patient) REFERENCES Patient(SSN),
  CONSTRAINT fk_Stay_Room_Number FOREIGN KEY(Room) REFERENCES Room(RoomNumber)
);

DROP TABLE IF EXISTS Undergoes;
CREATE TABLE Undergoes (
  Patient INTEGER NOT NULL,
  Procedures INTEGER NOT NULL,
  Stay INTEGER NOT NULL,
  DateUndergoes DATETIME NOT NULL,
  Physician INTEGER NOT NULL,
  AssistingNurse INTEGER,
  PRIMARY KEY(Patient, Procedures, Stay, DateUndergoes),
  CONSTRAINT fk_Undergoes_Patient_SSN FOREIGN KEY(Patient) REFERENCES Patient(SSN),
  CONSTRAINT fk_Undergoes_Procedures_Code FOREIGN KEY(Procedures) REFERENCES Procedures(Code),
  CONSTRAINT fk_Undergoes_Stay_StayID FOREIGN KEY(Stay) REFERENCES Stay(StayID),
  CONSTRAINT fk_Undergoes_Physician_EmployeeID FOREIGN KEY(Physician) REFERENCES Physician(EmployeeID),
  CONSTRAINT fk_Undergoes_Nurse_EmployeeID FOREIGN KEY(AssistingNurse) REFERENCES Nurse(EmployeeID)
);
```

Inserting Data

```
use HOSPITAL;
INSERT INTO Medication VALUES(1,'Procrastin-X','X','N/A');
INSERT INTO Medication VALUES(2,'Thesisin','Foo Labs','N/A');
INSERT INTO Medication VALUES(3,'Awakin','Bar Laboratories','N/A');
INSERT INTO Medication VALUES(4,'Crescavitin','Baz Industries','N/A');
INSERT INTO Medication VALUES(5,'Melioraurin','Snafu Pharmaceuticals','N/A');

INSERT INTO Block VALUES(1,1);
INSERT INTO Block VALUES(1,2);
INSERT INTO Block VALUES(1,3);
INSERT INTO Block VALUES(2,1);
INSERT INTO Block VALUES(2,2);
INSERT INTO Block VALUES(2,3);
INSERT INTO Block VALUES(3,1);
INSERT INTO Block VALUES(3,2);
INSERT INTO Block VALUES(3,3);
INSERT INTO Block VALUES(4,1);
INSERT INTO Block VALUES(4,2);
INSERT INTO Block VALUES(4,3);

INSERT INTO Room VALUES(101,'Single',1,1,0);
INSERT INTO Room VALUES(102,'Single',1,1,0);
INSERT INTO Room VALUES(103,'Single',1,1,0);
INSERT INTO Room VALUES(111,'Single',1,2,0);
INSERT INTO Room VALUES(112,'Single',1,2,1);
INSERT INTO Room VALUES(113,'Single',1,2,0);
INSERT INTO Room VALUES(121,'Single',1,3,0);
INSERT INTO Room VALUES(122,'Single',1,3,0);
INSERT INTO Room VALUES(123,'Single',1,3,0);
INSERT INTO Room VALUES(201,'Single',2,1,1);
INSERT INTO Room VALUES(202,'Single',2,1,0);
INSERT INTO Room VALUES(203,'Single',2,1,0);
INSERT INTO Room VALUES(211,'Single',2,2,0);
INSERT INTO Room VALUES(212,'Single',2,2,0);
INSERT INTO Room VALUES(213,'Single',2,2,1);
INSERT INTO Room VALUES(221,'Single',2,3,0);
INSERT INTO Room VALUES(222,'Single',2,3,0);
INSERT INTO Room VALUES(223,'Single',2,3,0);
INSERT INTO Room VALUES(301,'Single',3,1,0);
INSERT INTO Room VALUES(302,'Single',3,1,1);
INSERT INTO Room VALUES(303,'Single',3,1,0);
INSERT INTO Room VALUES(311,'Single',3,2,0);
INSERT INTO Room VALUES(312,'Single',3,2,0);
INSERT INTO Room VALUES(313,'Single',3,2,0);
INSERT INTO Room VALUES(321,'Single',3,3,1);
INSERT INTO Room VALUES(322,'Single',3,3,0);
INSERT INTO Room VALUES(323,'Single',3,3,0);
INSERT INTO Room VALUES(401,'Single',4,1,0);
INSERT INTO Room VALUES(402,'Single',4,1,1);
INSERT INTO Room VALUES(403,'Single',4,1,0);
INSERT INTO Room VALUES(411,'Single',4,2,0);
INSERT INTO Room VALUES(412,'Single',4,2,0);
INSERT INTO Room VALUES(413,'Single',4,2,0);
INSERT INTO Room VALUES(421,'Single',4,3,1);
INSERT INTO Room VALUES(422,'Single',4,3,0);
INSERT INTO Room VALUES(423,'Single',4,3,0);

INSERT INTO Physician VALUES('1', 'Aakash Choudhary', 'Staff Internist', '123654563');
INSERT INTO Physician VALUES('2', 'Saurav Yadav ', 'Attending Physician', '195478945');
INSERT INTO Physician VALUES('3', 'Kartik Pandey', 'Surgical Attending Physician', '154564654');
INSERT INTO Physician VALUES('4', 'Kuldeep Choudhary', 'Senior Attending Physician', '186654654');
INSERT INTO Physician VALUES('5', 'Manju Patwal', 'Head Chief of Medicine', '176656656');
INSERT INTO Physician VALUES('6', 'Gaurav Kumar', 'Surgical Attending Physician', '198654566');
INSERT INTO Physician VALUES('7', 'Richa Batra', 'Surgical Attending Physician', '197985352');
INSERT INTO Physician VALUES('8', 'Kamika Chouhan', 'MD Resident', '124478935');
INSERT INTO Physician VALUES('9', 'Sunita Singh', 'Attending Psychiatrist', '156565487');

INSERT INTO Affiliated_With VALUES(1,1,1);
INSERT INTO Affiliated_With VALUES(2,1,1);
INSERT INTO Affiliated_With VALUES(3,1,0);
INSERT INTO Affiliated_With VALUES(3,2,1);
INSERT INTO Affiliated_With VALUES(4,1,1);
INSERT INTO Affiliated_With VALUES(5,1,1);
INSERT INTO Affiliated_With VALUES(6,2,1);
INSERT INTO Affiliated_With VALUES(7,1,0);
INSERT INTO Affiliated_With VALUES(7,2,1);
INSERT INTO Affiliated_With VALUES(8,1,1);
INSERT INTO Affiliated_With VALUES(9,3,1);

INSERT INTO Department VALUES(1,'General Medicine',4);
INSERT INTO Department VALUES(2,'Surgery',7);
INSERT INTO Department VALUES(3,'Psychiatry',9);

INSERT INTO Appointment VALUES(13216584,100000001,101,1,'2020-04-24 10:00','2020-04-24 11:00','A');
INSERT INTO Appointment VALUES(26548913,100000002,101,2,'2020-04-24 10:00','2020-04-24 11:00','B');
INSERT INTO Appointment VALUES(36549879,100000001,102,1,'2020-04-25 10:00','2020-04-25 11:00','A');
INSERT INTO Appointment VALUES(46846589,100000004,103,4,'2020-04-25 10:00','2020-04-25 11:00','B');
INSERT INTO Appointment VALUES(59871321,100000004,NULL,4,'2020-04-26 10:00','2020-04-26 11:00','C');
INSERT INTO Appointment VALUES(69879231,100000003,103,2,'2020-04-26 11:00','2020-04-26 12:00','C');
INSERT INTO Appointment VALUES(76983231,100000001,NULL,3,'2020-04-26 12:00','2020-04-26 13:00','C');
INSERT INTO Appointment VALUES(86213939,100000004,102,9,'2020-04-27 10:00','2020-04-21 11:00','A');
INSERT INTO Appointment VALUES(93216548,100000002,101,2,'2020-04-27 10:00','2020-04-27 11:00','B');

INSERT INTO Patient VALUES('100000001', 'Surender Yadav', '42 Subhas Nagar', '9854673215', '68476213', '1');
INSERT INTO Patient VALUES('100000002', 'Garima Choudhary', '37 Utttam Nagar', '9999546897', '36546321', '2');
INSERT INTO Patient VALUES('100000003', 'R. Prakash', '101 Dwarka', '9821020727', '65465421', '2');
INSERT INTO Patient VALUES('100000004', 'Rajesh Koothrapalli', 'C-110 Rohini', '8824841892', '68421879', '3');

INSERT INTO Nurse VALUES('101', 'Priyanaka Kumari', 'Head Nurse', '1', '111111110');
INSERT INTO Nurse VALUES('102', 'Kritika', 'Nurse', '1', '222222220');
INSERT INTO Nurse VALUES('103', 'Salini Sharma', 'Nurse', '0', '333333330');

INSERT INTO On_Call VALUES(101,1,1,'2020-11-04 11:00','2020-11-04 19:00');
INSERT INTO On_Call VALUES(101,1,2,'2020-11-04 11:00','2020-11-04 19:00');
INSERT INTO On_Call VALUES(102,1,3,'2020-11-04 11:00','2020-11-04 19:00');
INSERT INTO On_Call VALUES(103,1,1,'2020-11-04 19:00','2020-11-05 03:00');
INSERT INTO On_Call VALUES(103,1,2,'2020-11-04 19:00','2020-11-05 03:00');
INSERT INTO On_Call VALUES(103,1,3,'2020-11-04 19:00','2020-11-05 03:00');

INSERT INTO Prescribes VALUES(1,100000001,1,'2020-04-24 10:47',13216584,'5');
INSERT INTO Prescribes VALUES(9,100000004,2,'2020-04-27 10:53',86213939,'10');
INSERT INTO Prescribes VALUES(9,100000004,2,'2020-04-30 16:53',NULL,'5');

INSERT INTO Procedures VALUES(1,'Reverse Rhinopodoplasty',1500.0);
INSERT INTO Procedures VALUES(2,'Obtuse Pyloric Recombobulation',3750.0);
INSERT INTO Procedures VALUES(3,'Folded Demiophtalmectomy',4500.0);
INSERT INTO Procedures VALUES(4,'Complete Walletectomy',10000.0);
INSERT INTO Procedures VALUES(5,'Obfuscated Dermogastrotomy',4899.0);
INSERT INTO Procedures VALUES(6,'Reversible Pancreomyoplasty',5600.0);
INSERT INTO Procedures VALUES(7,'Follicular Demiectomy',25.0);

INSERT INTO Stay VALUES(3215,100000001,111,'2008-05-01','2008-05-04');
INSERT INTO Stay VALUES(3216,100000003,123,'2008-05-03','2008-05-14');
INSERT INTO Stay VALUES(3217,100000004,112,'2008-05-02','2008-05-03');

INSERT INTO Undergoes VALUES(100000001,6,3215,'2020-05-02',3,101);
INSERT INTO Undergoes VALUES(100000001,2,3215,'2020-05-03',7,101);
INSERT INTO Undergoes VALUES(100000004,1,3217,'2020-05-07',3,102);
INSERT INTO Undergoes VALUES(100000004,5,3217,'2020-05-09',6,NULL);
INSERT INTO Undergoes VALUES(100000001,7,3217,'2020-05-10',7,101);
INSERT INTO Undergoes VALUES(100000004,4,3217,'2020-05-13',3,103);

INSERT INTO Trained_In VALUES(3,1,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(3,2,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(3,5,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(3,6,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(3,7,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(6,2,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(6,5,'2018-01-01','2019-12-31');
INSERT INTO Trained_In VALUES(6,6,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,1,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,2,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,3,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,4,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,5,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,6,'2019-01-01','2020-12-31');
INSERT INTO Trained_In VALUES(7,7,'2019-01-01','2020-12-31');
```