Table creation code

DROP TABLE IF EXISTS Physician;

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 BIT 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 BIT 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,

Starto DATETIME NOT NULL,

Endo 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 BIT 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)

);

CREATE TABLE Physician (

EmployeeID INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Position TEXT NOT NULL,

SSN INTEGER NOT NULL

);

CREATE TABLE Department (

DepartmentID INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Head INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID)

);

CREATE TABLE Affiliated\_With (

Physician INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID),

Department INTEGER NOT NULL

CONSTRAINT fk\_Department\_DepartmentID REFERENCES Department(DepartmentID),

PrimaryAffiliation BOOLEAN NOT NULL,

PRIMARY KEY(Physician, Department)

);

CREATE TABLE Procedure (

Code INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Cost REAL NOT NULL

);

CREATE TABLE Trained\_In (

Physician INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID),

Treatment INTEGER NOT NULL

CONSTRAINT fk\_Procedure\_Code REFERENCES Procedure(Code),

CertificationDate DATETIME NOT NULL,

CertificationExpires DATETIME NOT NULL,

PRIMARY KEY(Physician, Treatment)

);

CREATE TABLE Patient (

SSN INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Address TEXT NOT NULL,

Phone TEXT NOT NULL,

InsuranceID INTEGER NOT NULL,

PCP INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID)

);

CREATE TABLE Nurse (

EmployeeID INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Position TEXT NOT NULL,

Registered BOOLEAN NOT NULL,

SSN INTEGER NOT NULL

);

CREATE TABLE Appointment (

AppointmentID INTEGER PRIMARY KEY NOT NULL,

Patient INTEGER NOT NULL

CONSTRAINT fk\_Patient\_SSN REFERENCES Patient(SSN),

PrepNurse INTEGER

CONSTRAINT fk\_Nurse\_EmployeeID REFERENCES Nurse(EmployeeID),

Physician INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID),

Start DATETIME NOT NULL,

End DATETIME NOT NULL,

ExaminationRoom TEXT NOT NULL

);

CREATE TABLE Medication (

Code INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL,

Brand TEXT NOT NULL,

Description TEXT NOT NULL

);

CREATE TABLE Prescribes (

Physician INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID),

Patient INTEGER NOT NULL

CONSTRAINT fk\_Patient\_SSN REFERENCES Patient(SSN),

Medication INTEGER NOT NULL

CONSTRAINT fk\_Medication\_Code REFERENCES Medication(Code),

Date DATETIME NOT NULL,

Appointment INTEGER

CONSTRAINT fk\_Appointment\_AppointmentID REFERENCES Appointment(AppointmentID),

Dose TEXT NOT NULL,

PRIMARY KEY(Physician, Patient, Medication, Date)

);

CREATE TABLE Block (

Floor INTEGER NOT NULL,

Code INTEGER NOT NULL,

PRIMARY KEY(Floor, Code)

);

CREATE TABLE Room (

Number INTEGER PRIMARY KEY NOT NULL,

Type TEXT NOT NULL,

BlockFloor INTEGER NOT NULL,

BlockCode INTEGER NOT NULL,

Unavailable BOOLEAN NOT NULL,

FOREIGN KEY(BlockFloor, BlockCode) REFERENCES Block

);

CREATE TABLE On\_Call (

Nurse INTEGER NOT NULL

CONSTRAINT fk\_Nurse\_EmployeeID REFERENCES Nurse(EmployeeID),

BlockFloor INTEGER NOT NULL,

BlockCode INTEGER NOT NULL,

Start DATETIME NOT NULL,

End DATETIME NOT NULL,

PRIMARY KEY(Nurse, BlockFloor, BlockCode, Start, End),

FOREIGN KEY(BlockFloor, BlockCode) REFERENCES Block

);

CREATE TABLE Stay (

StayID INTEGER PRIMARY KEY NOT NULL,

Patient INTEGER NOT NULL

CONSTRAINT fk\_Patient\_SSN REFERENCES Patient(SSN),

Room INTEGER NOT NULL

CONSTRAINT fk\_Room\_Number REFERENCES Room(Number),

Start DATETIME NOT NULL,

End DATETIME NOT NULL

);

CREATE TABLE Undergoes (

Patient INTEGER NOT NULL

CONSTRAINT fk\_Patient\_SSN REFERENCES Patient(SSN),

Procedure INTEGER NOT NULL

CONSTRAINT fk\_Procedure\_Code REFERENCES Procedure(Code),

Stay INTEGER NOT NULL

CONSTRAINT fk\_Stay\_StayID REFERENCES Stay(StayID),

Date DATETIME NOT NULL,

Physician INTEGER NOT NULL

CONSTRAINT fk\_Physician\_EmployeeID REFERENCES Physician(EmployeeID),

AssistingNurse INTEGER

CONSTRAINT fk\_Nurse\_EmployeeID REFERENCES Nurse(EmployeeID),

PRIMARY KEY(Patient, Procedure, Stay, Date)

);

Sample dataset

INSERT INTO Physician VALUES(1,'John Dorian','Staff Internist',111111111);

INSERT INTO Physician VALUES(2,'Elliot Reid','Attending Physician',222222222);

INSERT INTO Physician VALUES(3,'Christopher Turk','Surgical Attending Physician',333333333);

INSERT INTO Physician VALUES(4,'Percival Cox','Senior Attending Physician',444444444);

INSERT INTO Physician VALUES(5,'Bob Kelso','Head Chief of Medicine',555555555);

INSERT INTO Physician VALUES(6,'Todd Quinlan','Surgical Attending Physician',666666666);

INSERT INTO Physician VALUES(7,'John Wen','Surgical Attending Physician',777777777);

INSERT INTO Physician VALUES(8,'Keith Dudemeister','MD Resident',888888888);

INSERT INTO Physician VALUES(9,'Molly Clock','Attending Psychiatrist',999999999);

INSERT INTO Department VALUES(1,'General Medicine',4);

INSERT INTO Department VALUES(2,'Surgery',7);

INSERT INTO Department VALUES(3,'Psychiatry',9);

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 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 Patient VALUES(100000001,'John Smith','42 Foobar Lane','555-0256',68476213,1);

INSERT INTO Patient VALUES(100000002,'Grace Ritchie','37 Snafu Drive','555-0512',36546321,2);

INSERT INTO Patient VALUES(100000003,'Random J. Patient','101 Omgbbq Street','555-1204',65465421,2);

INSERT INTO Patient VALUES(100000004,'Dennis Doe','1100 Foobaz Avenue','555-2048',68421879,3);

INSERT INTO Nurse VALUES(101,'Carla Espinosa','Head Nurse',1,111111110);

INSERT INTO Nurse VALUES(102,'Laverne Roberts','Nurse',1,222222220);

INSERT INTO Nurse VALUES(103,'Paul Flowers','Nurse',0,333333330);

INSERT INTO Appointment VALUES(13216584,100000001,101,1,'2008-04-24 10:00','2008-04-24 11:00','A');

INSERT INTO Appointment VALUES(26548913,100000002,101,2,'2008-04-24 10:00','2008-04-24 11:00','B');

INSERT INTO Appointment VALUES(36549879,100000001,102,1,'2008-04-25 10:00','2008-04-25 11:00','A');

INSERT INTO Appointment VALUES(46846589,100000004,103,4,'2008-04-25 10:00','2008-04-25 11:00','B');

INSERT INTO Appointment VALUES(59871321,100000004,NULL,4,'2008-04-26 10:00','2008-04-26 11:00','C');

INSERT INTO Appointment VALUES(69879231,100000003,103,2,'2008-04-26 11:00','2008-04-26 12:00','C');

INSERT INTO Appointment VALUES(76983231,100000001,NULL,3,'2008-04-26 12:00','2008-04-26 13:00','C');

INSERT INTO Appointment VALUES(86213939,100000004,102,9,'2008-04-27 10:00','2008-04-21 11:00','A');

INSERT INTO Appointment VALUES(93216548,100000002,101,2,'2008-04-27 10:00','2008-04-27 11:00','B');

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 Prescribes VALUES(1,100000001,1,'2008-04-24 10:47',13216584,'5');

INSERT INTO Prescribes VALUES(9,100000004,2,'2008-04-27 10:53',86213939,'10');

INSERT INTO Prescribes VALUES(9,100000004,2,'2008-04-30 16:53',NULL,'5');

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 On\_Call VALUES(101,1,1,'2008-11-04 11:00','2008-11-04 19:00');

INSERT INTO On\_Call VALUES(101,1,2,'2008-11-04 11:00','2008-11-04 19:00');

INSERT INTO On\_Call VALUES(102,1,3,'2008-11-04 11:00','2008-11-04 19:00');

INSERT INTO On\_Call VALUES(103,1,1,'2008-11-04 19:00','2008-11-05 03:00');

INSERT INTO On\_Call VALUES(103,1,2,'2008-11-04 19:00','2008-11-05 03:00');

INSERT INTO On\_Call VALUES(103,1,3,'2008-11-04 19:00','2008-11-05 03:00');

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,'2008-05-02',3,101);

INSERT INTO Undergoes VALUES(100000001,2,3215,'2008-05-03',7,101);

INSERT INTO Undergoes VALUES(100000004,1,3217,'2008-05-07',3,102);

INSERT INTO Undergoes VALUES(100000004,5,3217,'2008-05-09',6,NULL);

INSERT INTO Undergoes VALUES(100000001,7,3217,'2008-05-10',7,101);

INSERT INTO Undergoes VALUES(100000004,4,3217,'2008-05-13',3,103);

INSERT INTO Trained\_In VALUES(3,1,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(3,2,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(3,5,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(3,6,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(3,7,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(6,2,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(6,5,'2007-01-01','2007-12-31');

INSERT INTO Trained\_In VALUES(6,6,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,1,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,2,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,3,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,4,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,5,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,6,'2008-01-01','2008-12-31');

INSERT INTO Trained\_In VALUES(7,7,'2008-01-01','2008-12-31');

## Exercises

1. Obtain the names of all physicians that have performed a medical procedure they have *never* been certified to perform.

SELECT Name

FROM Physician

WHERE EmployeeID IN

(

SELECT Physician FROM Undergoes U WHERE NOT EXISTS

(

SELECT \* FROM Trained\_In

WHERE Treatment = Procedure

AND Physician = U.Physician

)

);

SELECT P.Name FROM

Physician AS P,

(SELECT Physician, Procedure FROM Undergoes

EXCEPT

SELECT Physician, Treatment FROM Trained\_in) AS Pe

WHERE P.EmployeeID=Pe.Physician]

SELECT Name

FROM Physician

WHERE EmployeeID IN

(

SELECT Undergoes.Physician

FROM Undergoes

LEFT JOIN Trained\_In

ON Undergoes.Physician=Trained\_In.Physician

AND Undergoes.Procedures=Trained\_In.Treatment

WHERE Treatment IS NULL

);

SELECT Physician.Name

FROM Physician

INNER JOIN Undergoes ON Physician.EmployeeID = Undergoes.Physician

LEFT JOIN Trained\_In ON Undergoes.Procedure = Trained\_In.Treatment AND Physician.EmployeeID = Trained\_In.Physician

WHERE Trained\_In.Treatment IS NULL

GROUP BY Physician.EmployeeID

2. Same as the previous query, but include the following information in the results: Physician name, name of procedure, date when the procedure was carried out, name of the patient the procedure was carried out on.

SELECT P.Name AS Physician, Pr.Name AS Procedure, U.Date, Pt.Name AS Patient

FROM Physician P, Undergoes U, Patient Pt, Procedure Pr

WHERE U.Patient = Pt.SSN

AND U.Procedure = Pr.Code

AND U.Physician = P.EmployeeID

AND NOT EXISTS

(

SELECT \* FROM Trained\_In T

WHERE T.Treatment = U.Procedure

AND T.Physician = U.Physician

);

SELECT P.Name,Pr.Name,U.Date,Pt.Name FROM

Physician AS P,

Procedure AS Pr,

Undergoes AS U,

Patient AS Pt,

(SELECT Physician, Procedure FROM Undergoes

EXCEPT

SELECT Physician, Treatment FROM Trained\_in) AS Pe

WHERE P.EmployeeID=Pe.Physician AND Pe.Procedure=Pr.Code AND Pe.Physician=U.Physician AND Pe.Procedure=U.Procedure AND U.Patient=Pt.SSN

3. Obtain the names of all physicians that have performed a medical procedure that they are certified to perform, but such that the procedure was done at a date (Undergoes.Date) after the physician's certification expired (Trained\_In.CertificationExpires).

SELECT Name

FROM Physician

WHERE EmployeeID IN

(

SELECT Physician FROM Undergoes U

WHERE Date >

(

SELECT CertificationExpires

FROM Trained\_In T

WHERE T.Physician = U.Physician

AND T.Treatment = U.Procedure

)

);

SELECT P.Name FROM

Physician AS P,

Trained\_In T,

Undergoes AS U

WHERE T.Physician=U.Physician AND T.Treatment=U.Procedure AND U.Date>T.CertificationExpires AND P.EmployeeID=U.Physician

4. Same as the previous query, but include the following information in the results: Physician name, name of procedure, date when the procedure was carried out, name of the patient the procedure was carried out on, and date when the certification expired.

SELECT P.Name AS Physician, Pr.Name AS Procedure, U.Date, Pt.Name AS Patient, T.CertificationExpires

FROM Physician P, Undergoes U, Patient Pt, Procedure Pr, Trained\_In T

WHERE U.Patient = Pt.SSN

AND U.Procedure = Pr.Code

AND U.Physician = P.EmployeeID

AND Pr.Code = T.Treatment

AND P.EmployeeID = T.Physician

AND U.Date > T.CertificationExpires;

5. Obtain the information for appointments where a patient met with a physician other than his/her primary care physician. Show the following information: Patient name, physician name, nurse name (if any), start and end time of appointment, examination room, and the name of the patient's primary care physician.

SELECT Pt.Name AS Patient, Ph.Name AS Physician, N.Name AS Nurse, A.Start, A.End, A.ExaminationRoom, PhPCP.Name AS PCP

FROM Patient Pt, Physician Ph, Physician PhPCP, Appointment A LEFT JOIN Nurse N ON A.PrepNurse = N.EmployeeID

WHERE A.Patient = Pt.SSN

AND A.Physician = Ph.EmployeeID

AND Pt.PCP = PhPCP.EmployeeID

AND A.Physician <> Pt.PCP;

6. The Patient field in Undergoes is redundant, since we can obtain it from the Stay table. There are no constraints in force to prevent inconsistencies between these two tables. More specifically, the Undergoes table may include a row where the patient ID does not match the one we would obtain from the Stay table through the Undergoes.Stay foreign key. Select all rows from Undergoes that exhibit this inconsistency.

SELECT \* FROM Undergoes U

WHERE Patient <>

(

SELECT Patient FROM Stay S

WHERE U.Stay = S.StayID

);

Obtain the names of all the nurses who have ever been on call for room 123.

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.Number = 123

);

7. The hospital has several examination rooms where appointments take place. Obtain the number of appointments that have taken place in each examination room.

N.b. The solution below fails in MS SQL Server Management Studio, with the following message:

Msg 306, Level 16, State 2, Line 473

The text, ntext, and image data types cannot be compared or sorted, except when using IS NULL or LIKE operator.

SELECT ExaminationRoom, COUNT(AppointmentID) AS Number FROM Appointment

GROUP BY ExaminationRoom;

8. Obtain the names of all patients (also include, for each patient, the name of the patient's primary care physician), such that \emph{all} the following are true:

* The patient has been prescribed some medication by his/her primary care physician.
* The patient has undergone a procedure with a cost larger that $5,000
* The patient has had at least two appointment where the nurse who prepped the appointment was a registered nurse.
* The patient's primary care physician is not the head of any department.

SELECT Pt.Name, PhPCP.Name FROM Patient Pt, Physician PhPCP

WHERE Pt.PCP = PhPCP.EmployeeID

AND EXISTS

(

SELECT \* FROM Prescribes Pr

WHERE Pr.Patient = Pt.SSN

AND Pr.Physician = Pt.PCP

)

AND EXISTS

(

SELECT \* FROM Undergoes U, Procedure Pr

WHERE U.Procedure = Pr.Code

AND U.Patient = Pt.SSN

AND Pr.Cost > 5000

)

AND 2 <=

(

SELECT COUNT(A.AppointmentID) FROM Appointment A, Nurse N

WHERE A.PrepNurse = N.EmployeeID

AND N.Registered = 1

)

AND NOT Pt.PCP IN

(

SELECT Head FROM Department

);