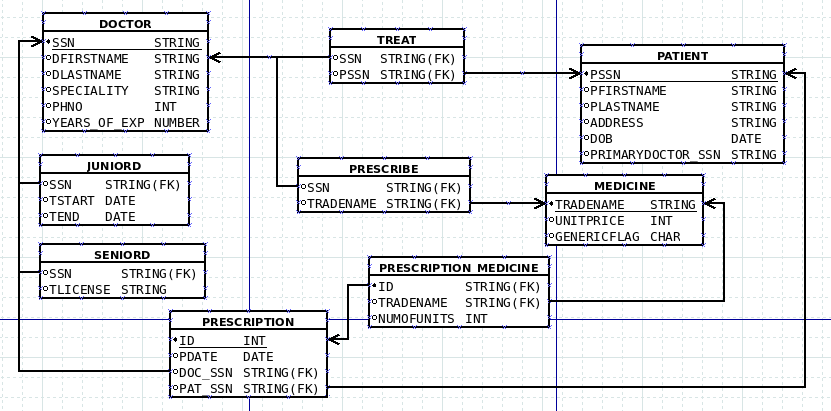
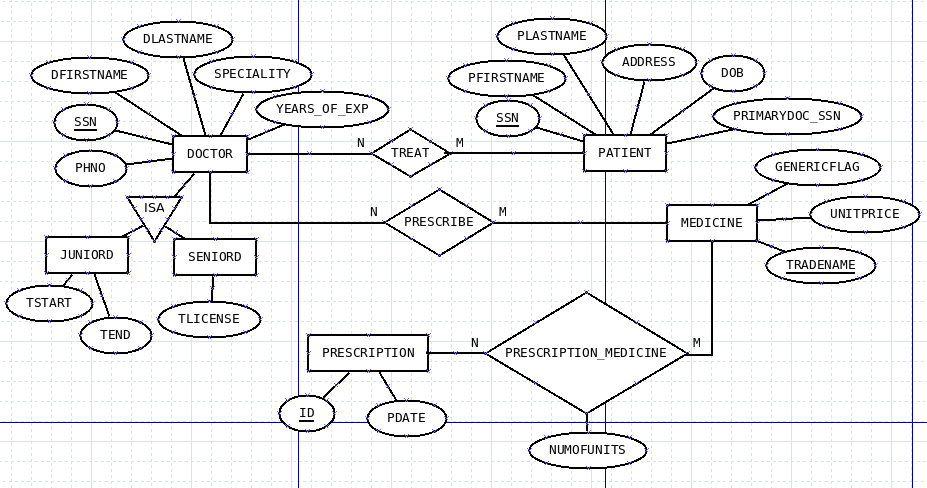
# ASSIGNMENT – 7

1. Design an ER diagram for an application that models a hospital doctors treat patients, prescribe tests, monitor progress etc. Analyze the requirements by identifying the entities, attributes, relationships, keys, constraints etc. Apply extended entity-relationship features to the design. Defend your design with proper assumptions and justifications. Map the ER model into a relational model.



# Assumptions and Justification:

* + Doctors can treat multiple patient, and patients can be treated by multiple doctors, which is many-to-many relationship.
  + Doctors can prescribe multiple tests and madications for patients.
  + Each patients progress can be mentioned through multiple pregress reports.
  + Doctors can specialize in different medical fields such as surgery, pediatrics etc.
  + Patients can be categorized as inpatients or outpatients.
  + Tests and medications share common attributes such as ID, name etc.
  + Each progress report includes details such as description and date.

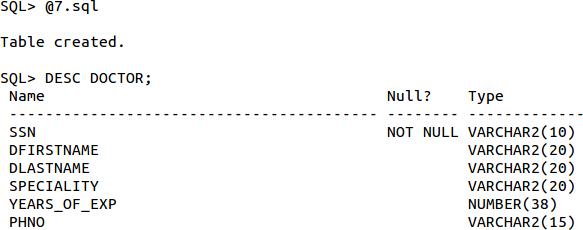
1. Create following tables, populate with data and construct queries (advanced) in SQL to extract information from the hospital doctor’s database.

CREATE TABLE DOCTOR(

SSN VARCHAR2(10) PRIMARY KEY, DFIRSTNAME VARCHAR2(20), DLASTNAME VARCHAR2(20), SPECIALITY VARCHAR2(20), YEARS\_OF\_EXP INT,

PHNO VARCHAR2(15)

);



CREATE TABLE PATIENT(

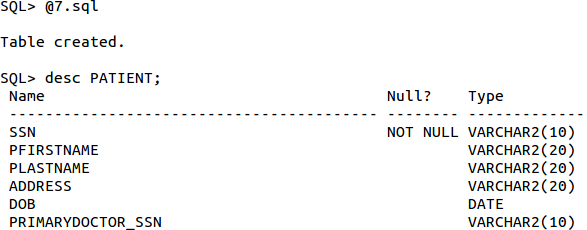
SSN VARCHAR2(10) PRIMARY KEY, PFIRSTNAME VARCHAR2(20), PLASTNAME VARCHAR2(20), ADDRESS VARCHAR2(20),

DOB DATE,

PRIMARYDOCTOR\_SSN VARCHAR2(10),

FOREIGN KEY(PRIMARYDOCTOR\_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE

);

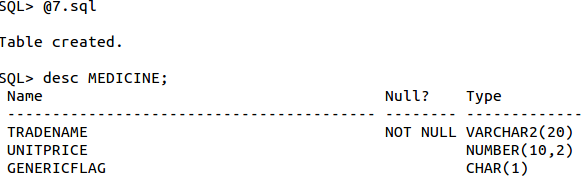


CREATE TABLE MEDICINE(

TRADENAME VARCHAR2(20) PRIMARY KEY, UNITPRICE NUMBER(10,2),

GENERICFLAG CHAR(1)

);

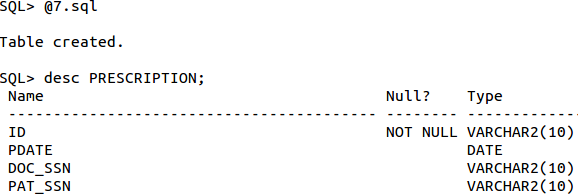


CREATE TABLE PRESCRIPTION(

ID VARCHAR2(10) PRIMARY KEY, PDATE DATE,

DOC\_SSN VARCHAR2(10), PAT\_SSN VARCHAR2(10),

FOREIGN KEY(DOC\_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE, FOREIGN KEY(PAT\_SSN) REFERENCES PATIENT(SSN) ON DELETE CASCADE

);

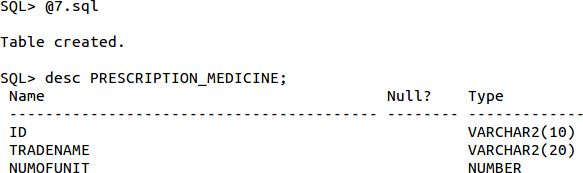
CREATE TABLE PRESCRIPTION\_MEDICINE( ID VARCHAR2(10),

TRADENAME VARCHAR2(20), NUMOFUNIT NUMBER,

FOREIGN KEY(ID) REFERENCES PRESCRIPTION(ID) ON DELETE CASCADE, FOREIGN KEY(TRADENAME) REFERENCES MEDICINE(TRADENAME) ON DELETE

CASCADE

);

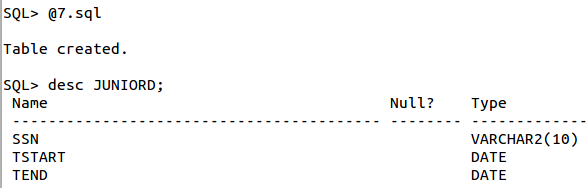


CREATE TABLE JUNIORD( SSN VARCHAR2(10), TSTART DATE,

TEND DATE,

FOREIGN KEY(SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE

);



CREATE TABLE SENIORD( SSN VARCHAR2(10), TLICENSE VARCHAR2(10),

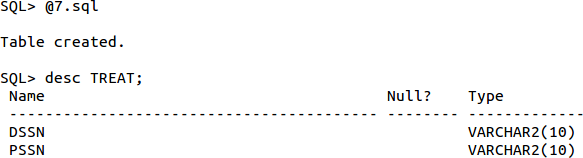
FOREIGN KEY(SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE

);



CREATE TABLE TREAT( DSSN VARCHAR2(10), PSSN VARCHAR2(10),

FOREIGN KEY(DSSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE, FOREIGN KEY(PSSN) REFERENCES PATIENT(SSN) ON DELETE CASCADE

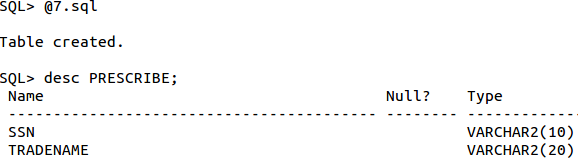
);

CREATE TABLE PRESCRIBE( SSN VARCHAR2(10), TRADENAME VARCHAR2(20),

FOREIGN KEY(SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE, FOREIGN KEY(TRADENAME) REFERENCES MEDICINE(TRADENAME) ON DELETE

CASCADE

);



Inserting Values:

INSERT INTO DOCTOR VALUES('123','JOHN','SMITH','CARDIOLOGY',15, 884399);

INSERT INTO DOCTOR VALUES('124','A','B','PEDIATRICS',10, 884399);

INSERT INTO DOCTOR VALUES('125','C','D','CARDIOLOGY',16, 884399);

INSERT INTO DOCTOR VALUES('126','E','F','PEDIATRICS',25, 884399);

INSERT INTO DOCTOR VALUES('127','G','H','CARDIOLOGY',9, 884399);



INSERT INTO PATIENT VALUES('1234','JOHN','SMITH','FDVNK',TO\_DATE('2003-07-09','YYYY-MM- DD'),'123');

INSERT INTO PATIENT VALUES('1235','GG','GHOSH','FDVNK',TO\_DATE('2003-07-09','YYYY-MM-

DD'),'124');

INSERT INTO PATIENT VALUES('1236','GG','PATIL','FDVNK',TO\_DATE('2003-07-09','YYYY-MM- DD'),'125');

INSERT INTO PATIENT VALUES('1237','BEN','SMITH','FDVNK',TO\_DATE('2003-07-09','YYYY-MM- DD'),'126');

INSERT INTO PATIENT VALUES('1238','DUCK','DUCKINS','FDVNK',TO\_DATE('2003-07-09','YYYY- MM-DD'),'127');



INSERT INTO MEDICINE VALUES('ASPIRIN',10.00,'Y');

INSERT INTO MEDICINE VALUES('P650',5.00,'Y');

INSERT INTO MEDICINE VALUES('S890',15.00,'N');

INSERT INTO MEDICINE VALUES('VITAMIN',15.00,'Y');

INSERT INTO MEDICINE VALUES('HOTST',10.00,'N');



INSERT INTO PRESCRIPTION VALUES('001',TO\_DATE('2025-03-27','YYYY-MM-DD'),'123','1234');

INSERT INTO PRESCRIPTION VALUES('002',TO\_DATE('2025-01-26','YYYY-MM-DD'),'124','1235'); NSERT INTO PRESCRIPTION VALUES('003',TO\_DATE('2025-03-23','YYYY-MM-DD'),'125','1236'); INSERT INTO PRESCRIPTION VALUES('004',TO\_DATE('2025-03-27','YYYY-MM-DD'),'126','1237'); INSERT INTO PRESCRIPTION VALUES('005',TO\_DATE('2025-03-23','YYYY-MM-DD'),'127','1238');



INSERT INTO PRESCRIPTION\_MEDICINE VALUES('001','P650',5);

INSERT INTO PRESCRIPTION\_MEDICINE VALUES('002','ASPIRIN',2);

INSERT INTO PRESCRIPTION\_MEDICINE VALUES('003','S890',3);

INSERT INTO PRESCRIPTION\_MEDICINE VALUES('004','VITAMIN',1);

INSERT INTO PRESCRIPTION\_MEDICINE VALUES('005','P650',2);



INSERT INTO JUNIORD VALUES('123',TO\_DATE('2004-09-14','YYYY-MM-DD'),TO\_DATE('2004-09- 14','YYYY-MM-DD'));

INSERT INTO JUNIORD VALUES('124',TO\_DATE('2004-09-14','YYYY-MM-DD'),TO\_DATE('2004-09- 14','YYYY-MM-DD'));

INSERT INTO JUNIORD VALUES('125',TO\_DATE('2004-09-14','YYYY-MM-DD'),TO\_DATE('2004-09- 14','YYYY-MM-DD'));

INSERT INTO SENIORD VALUES('126','VBHKX');

INSERT INTO SENIORD VALUES('127','CDHKX');



INSERT INTO TREAT VALUES('123','1234');

INSERT INTO TREAT VALUES('124','1235');

INSERT INTO TREAT VALUES('125','1236');

INSERT INTO TREAT VALUES('126','1237');

INSERT INTO TREAT VALUES('127','1238');



INSERT INTO PRESCRIBE VALUES ('123', 'P650');

INSERT INTO PRESCRIBE VALUES ('123', 'VITAMIN');

INSERT INTO PRESCRIBE VALUES ('125', 'P650');

INSERT INTO PRESCRIBE VALUES ('126', 'P650');

INSERT INTO PRESCRIBE VALUES ('127', 'VITAMIN');



1.List the trade name of generic medicine with unit price less than $50.

SELECT TRADENAME FROM MEDICINE WHERE GENERICFLAG = 'Y' AND UNITPRICE < 50;



2.List the first and last name of patients whose primary doctor named ʻJohn Smithʼ.

SELECT p.PFIRSTNAME, p.PLASTNAME

FROM PATIENT p

JOIN DOCTOR d ON p.PRIMARYDOCTOR\_SSN = d.SSN

WHERE d.DFIRSTNAME = 'John' AND d.DLASTNAME = 'Smith';

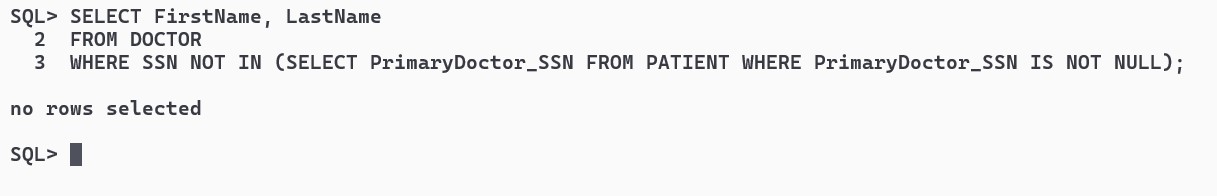


3.List the first and last name of doctors who are not primary doctors to any patient.

SELECT FirstName, LastName

FROM DOCTOR

WHERE SSN NOT IN (SELECT PrimaryDoctor\_SSN FROM PATIENT WHERE PrimaryDoctor\_SSN IS NOT NULL);



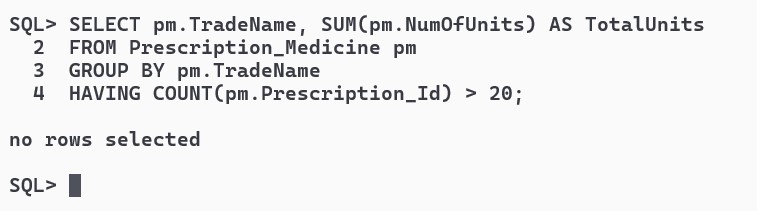
4.For medicines written in more than 20 prescriptions, report the trade name and the total number of units prescribed.

SELECT pm.TradeName, SUM(pm.NumOfUnits) AS TotalUnits

FROM Prescription\_Medicine pm

GROUP BY pm.TradeName

HAVING COUNT(pm.Prescription\_Id) > 20;



5.List the SSN of patients who have ʻAspirinʼ and ʻVitaminʼ trade names in one prescription.

SELECT p.SSN

FROM PATIENT p

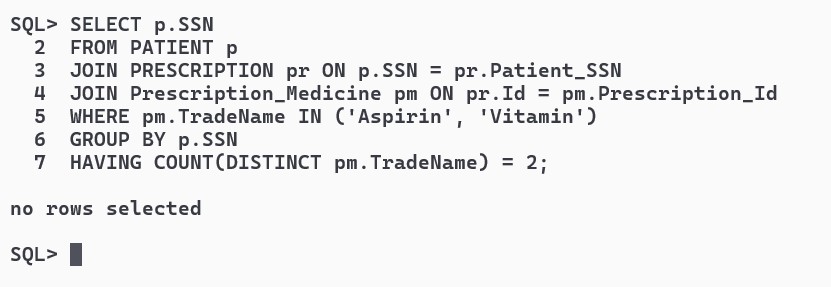
JOIN PRESCRIPTION pr ON p.SSN = pr.Patient\_SSN

JOIN Prescription\_Medicine pm ON pr.Id = pm.Prescription\_Id

WHERE pm.TradeName IN ('Aspirin', 'Vitamin')

GROUP BY p.SSN

HAVING COUNT(DISTINCT pm.TradeName) = 2;



6.List the SNN of distinct patients who have ʻAspirinʼ prescribed to them by doctor named ʻJohn Smithʼ.

SELECT DISTINCT p.SSN

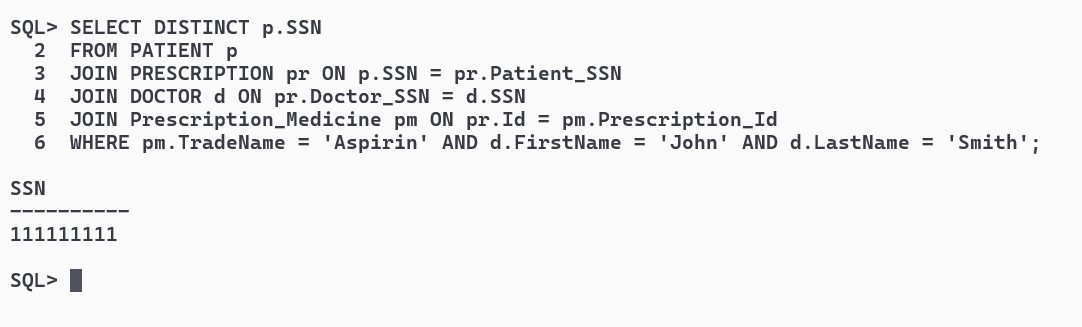
FROM PATIENT p

JOIN PRESCRIPTION pr ON p.SSN = pr.Patient\_SSN

JOIN DOCTOR d ON pr.Doctor\_SSN = d.SSN

JOIN Prescription\_Medicine pm ON pr.Id = pm.Prescription\_Id

WHERE pm.TradeName = 'Aspirin' AND d.FirstName = 'John' AND d.LastName = 'Smith';



7.List the first and last name of patients who have no prescriptions written by doctors other than their primary doctors.

SELECT p.FirstName, p.LastName FROM PATIENT p WHERE NOT EXISTS (

SELECT \* FROM PRESCRIPTION pr JOIN DOCTOR d ON pr.Doctor\_SSN = d.SSN

WHERE pr.Patient\_SSN = p.SSN AND pr.Doctor\_SSN <> p.PrimaryDoctor\_SSN);

