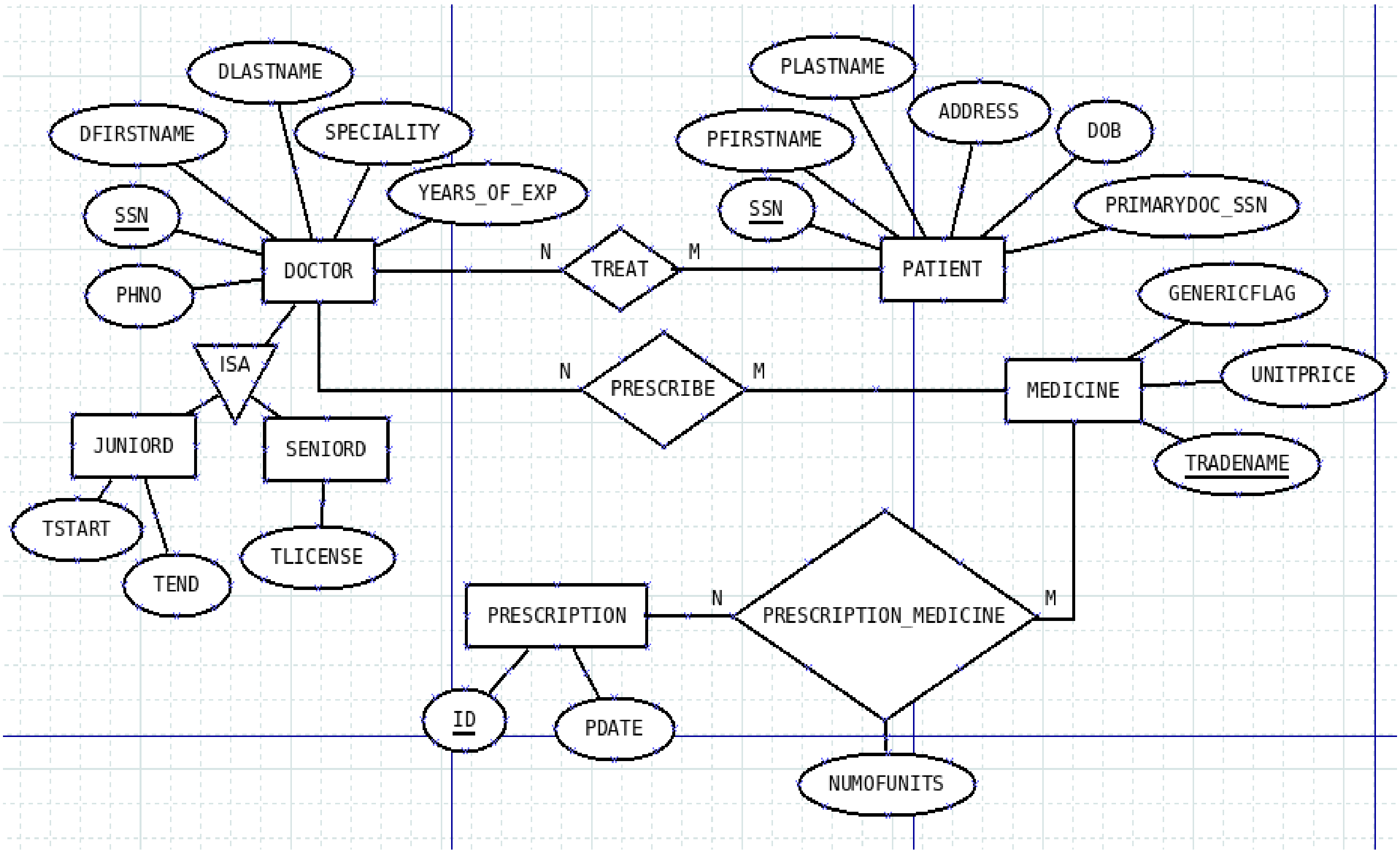
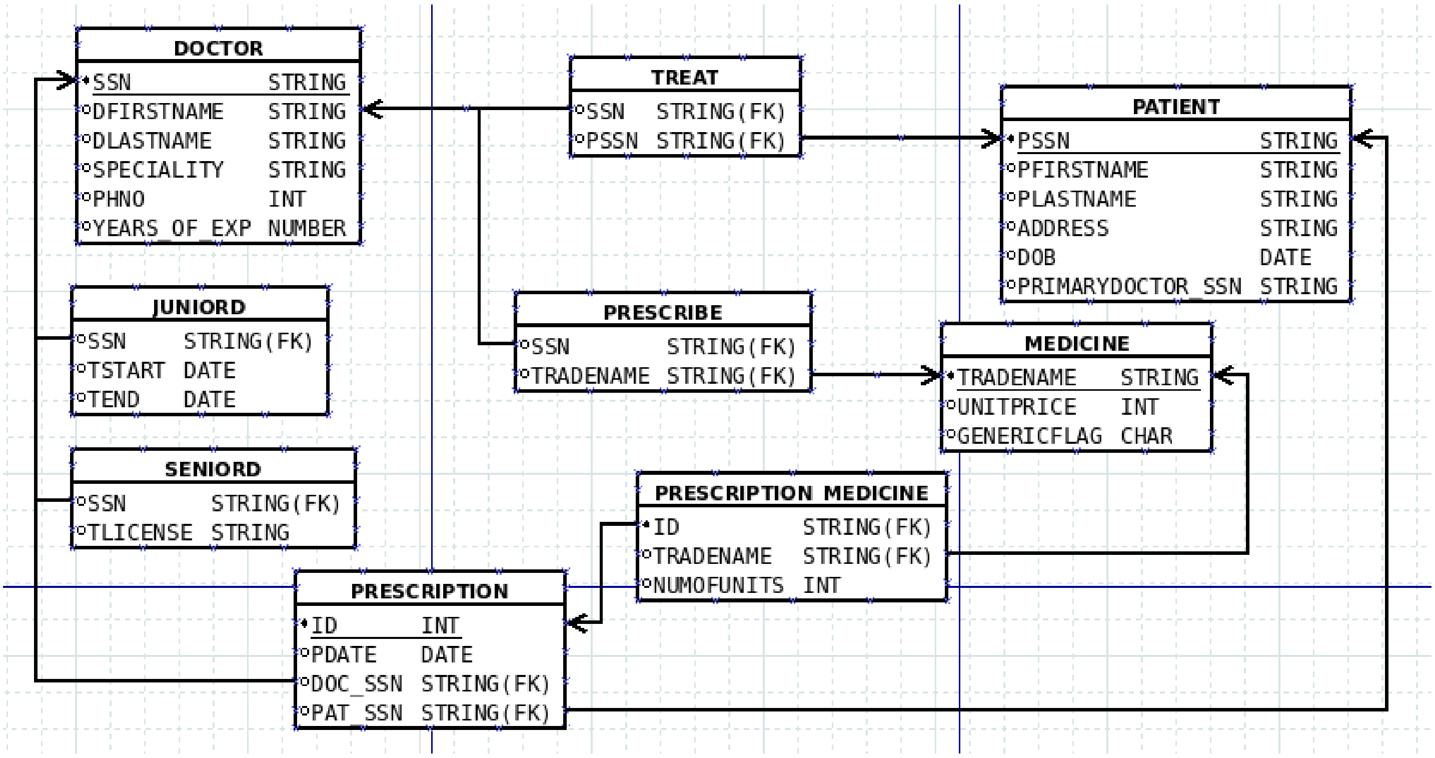
ASSIGNMENT – 7

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.





DOCTOR	
◆SSN	STRING
○DFIRSTNAME	STRING
○DLASTNAME	STRING
○SPECIALITY	STRING
○PHNO	INT
○YEARS_OF_EXP	NUMBER

TREAT	
○SSN	STRING(FK)
○PSSN	STRING(FK)

PATIENT	
◆PSSN	STRING
○PFIRSTNAME	STRING
○PLASTNAME	STRING
○ADDRESS	STRING
○DOB	DATE
○PRIMARYDOCTOR_SSN	STRING

JUNIORD	
○SSN	STRING(FK)
○TSTART	DATE
○TEND	DATE

PRESCRIBE	
○SSN	STRING(FK)
○TRADENAME	STRING(FK)

MEDICINE	
◆TRADENAME	STRING
○UNITPRICE	INT
○GENERICFLAG	CHAR

SENIORD	
○SSN	STRING(FK)
○TLICENSE	STRING

PRESCRIPTION_MEDICINE	
◆ID	STRING(FK)
○TRADENAME	STRING(FK)
○NUMOFUNITS	INT

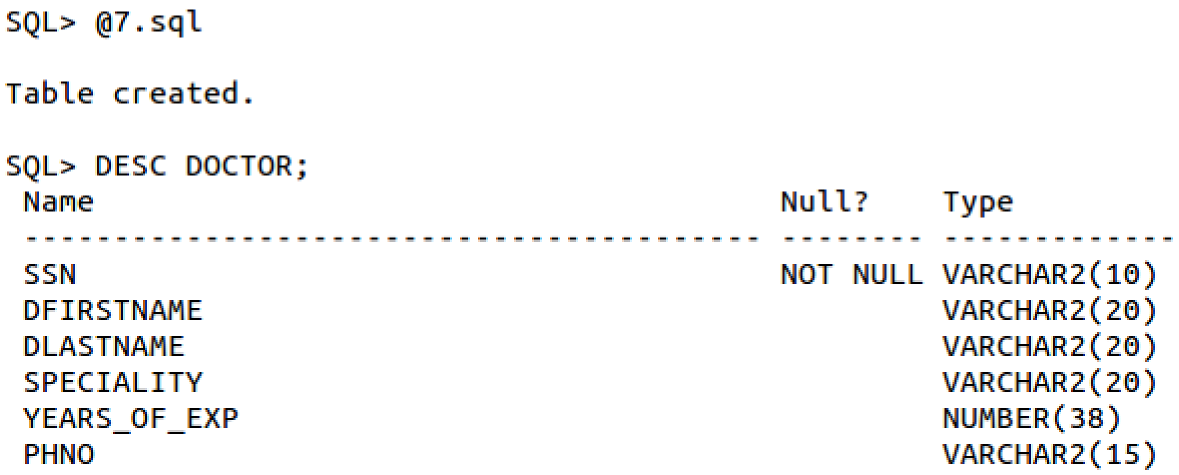
PRESCRIPTION	
◆ID	INT
○PDATE	DATE
○DOC_SSN	STRING(FK)
○PAT_SSN	STRING(FK)

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.

2. 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)
);
```

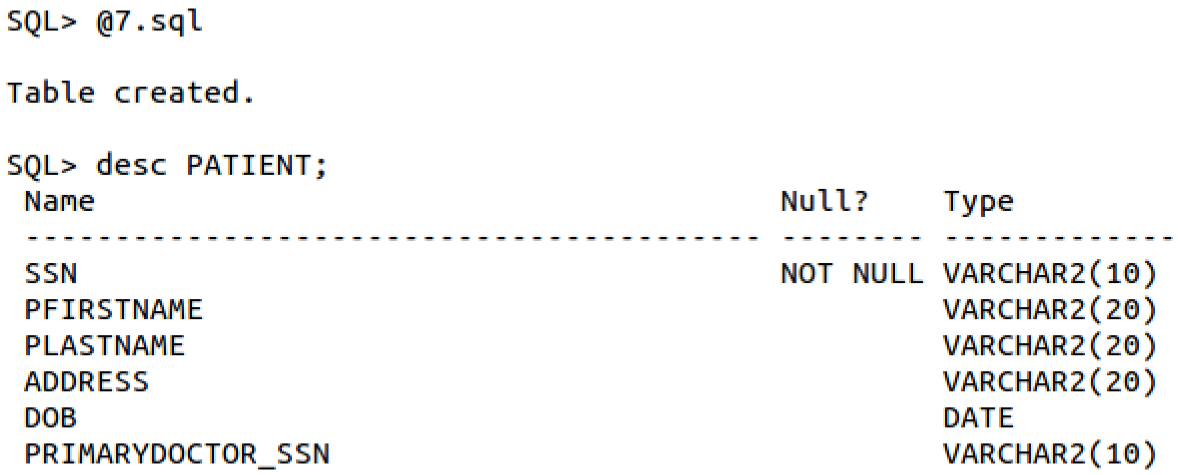


```
SQL> @7.sql

Table created.

SQL> DESC DOCTOR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 SSN                                       NOT NULL VARCHAR2(10)
 DFIRSTNAME                                         VARCHAR2(20)
 DLASTNAME                                          VARCHAR2(20)
 SPECIALITY                                         VARCHAR2(20)
 YEARS_OF_EXP                                       NUMBER(38)
 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
);
```

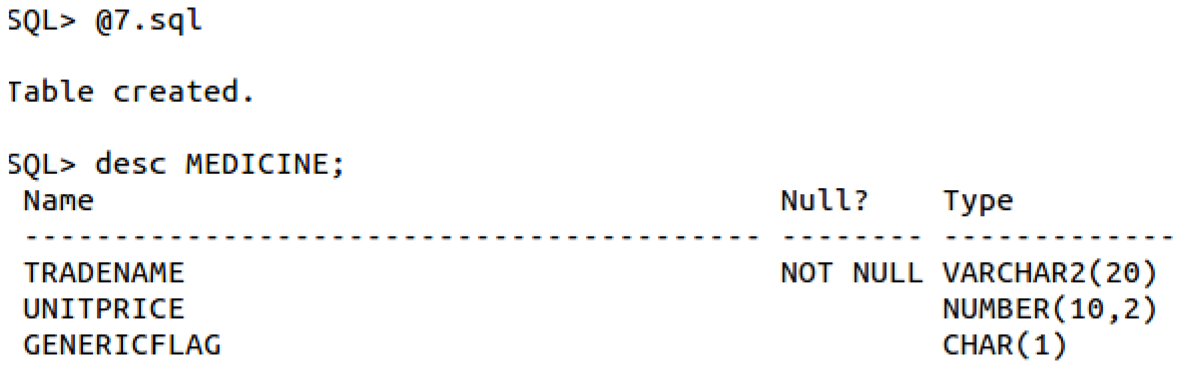


```
SQL> @7.sql

Table created.

SQL> desc PATIENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 SSN                                       NOT NULL VARCHAR2(10)
 PFIRSTNAME                                         VARCHAR2(20)
 PLASTNAME                                          VARCHAR2(20)
 ADDRESS                                            VARCHAR2(20)
 DOB                                                DATE
 PRIMARYDOCTOR_SSN                                  VARCHAR2(10)
```

```
CREATE TABLE MEDICINE(
   TRADENAME VARCHAR2(20) PRIMARY KEY,
   UNITPRICE NUMBER(10,2),
   GENERICFLAG CHAR(1)
);
```

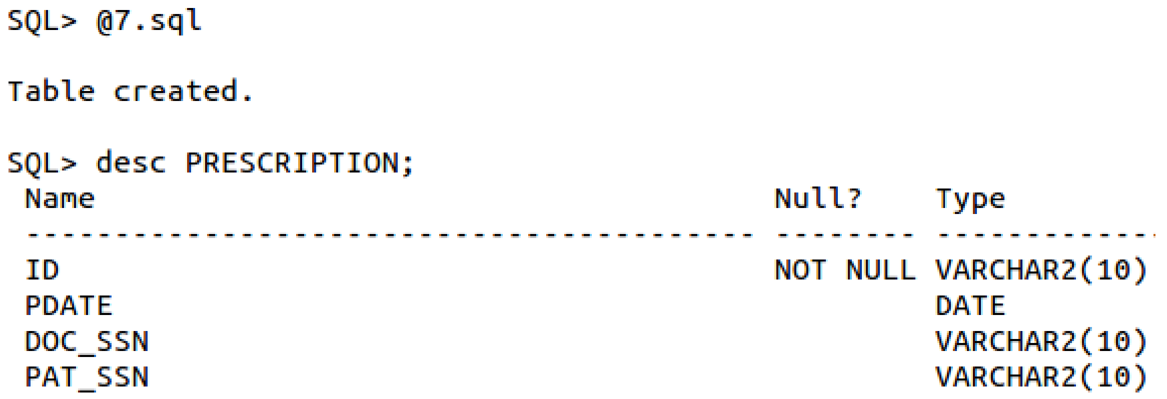


```
SQL> @7.sql

Table created.

SQL> desc MEDICINE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 TRADENAME                                 NOT NULL VARCHAR2(20)
 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
);
```

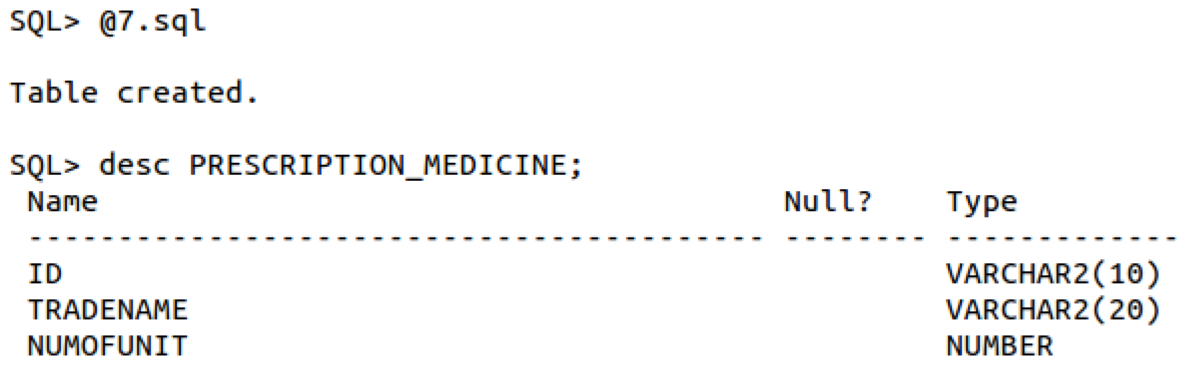


```
SQL> @7.sql

Table created.

SQL> desc PRESCRIPTION;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 ID                                        NOT NULL VARCHAR2(10)
 PDATE                                              DATE
 DOC_SSN                                            VARCHAR2(10)
 PAT_SSN                                            VARCHAR2(10)
```

```
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
);
```

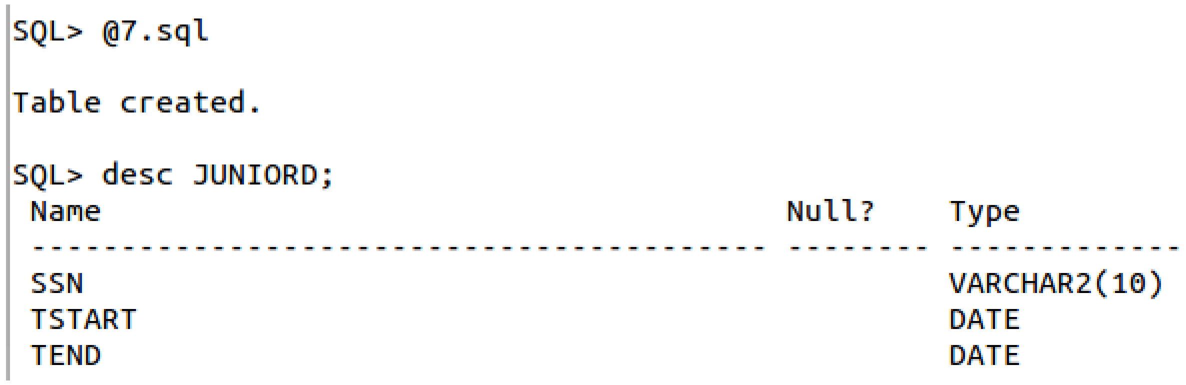


```
SQL> @7.sql

Table created.

SQL> desc PRESCRIPTION_MEDICINE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 ID                                                 VARCHAR2(10)
 TRADENAME                                          VARCHAR2(20)
 NUMOFUNIT                                          NUMBER
```

```
CREATE TABLE JUNIORD(
    SSN VARCHAR2(10),
    TSTART DATE,
    TEND DATE,
    FOREIGN KEY(SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE
);
```

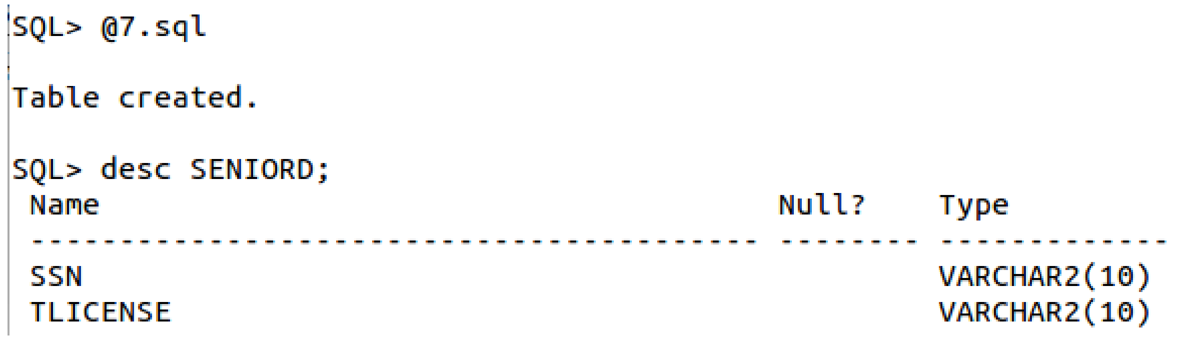


```
SQL> @7.sql

Table created.

SQL> desc JUNIORD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 SSN                                                VARCHAR2(10)
 TSTART                                             DATE
 TEND                                               DATE
```

```
CREATE TABLE SENIORD(
    SSN VARCHAR2(10),
    TLICENSE VARCHAR2(10),
    FOREIGN KEY(SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE
);
```

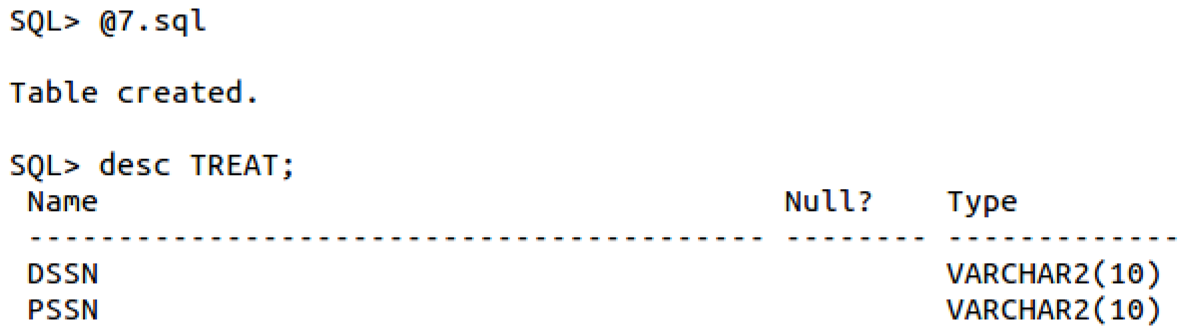


```
SQL> @7.sql

Table created.

SQL> desc SENIORD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 SSN                                                VARCHAR2(10)
 TLICENSE                                           VARCHAR2(10)
```

```
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
);
```

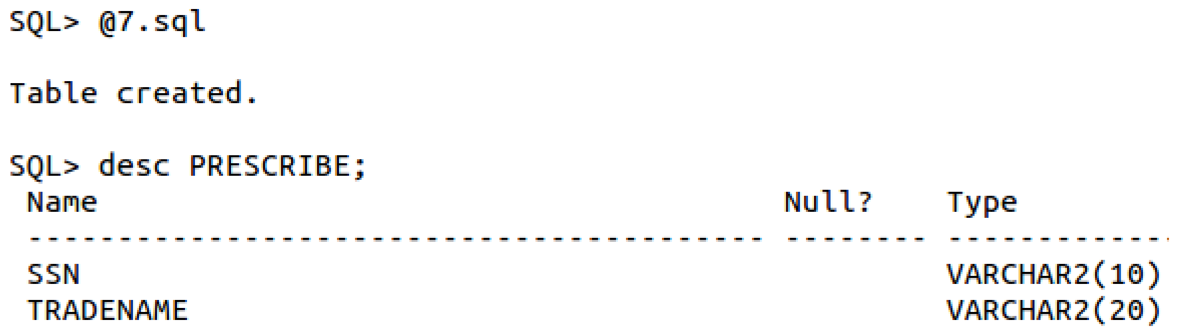


```
SQL> @7.sql

Table created.

SQL> desc TREAT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 DSSN                                               VARCHAR2(10)
 PSSN                                               VARCHAR2(10)
```

```
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
);
```



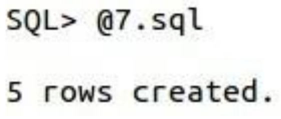
```
SQL> @7.sql

Table created.

SQL> desc PRESCRIBE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 SSN                                                VARCHAR2(10)
 TRADENAME                                          VARCHAR2(20)
```

Inserting Values:

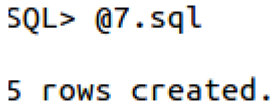
```
INSERT INTO DOCTOR VALUES('123','JOHN','SMITH','CARDIOLOGY',15, 884399);
INSERT INTO DOCTOR VALUES('124','A','B','PEDIATRICS',10, 884399);
INSERT INTO DOCTOR VALUES('125','C','D','CARDIOLOGY',16,
884399);
INSERT INTO DOCTOR VALUES('126','E','F','PEDIATRICS',25,
884399);
INSERT INTO DOCTOR VALUES('127','G','H','CARDIOLOGY',9,
884399);
```



```
SQL> @7.sql

5 rows created.
```

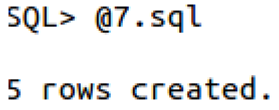
```
INSERT INTO PATIENT VALUES('1234','JOHN','SMITH','FDVNK',TO_DATE('2003-07-09','YYYY-
MM- DD'),'123');
INSERT INTO  PATIENT  VALUES('1235','GG','GHOSH','FDVNK',TO_DATE('2003-07-09','YYYY-MM-
DD'),'124');
INSERT INTO PATIENT VALUES('1236','GG','PATIL','FDVNK',TO_DATE('2003-07-
09','YYYY-MM- DD'),'125');
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');
```



```
SQL> @7.sql

5 rows created.
```

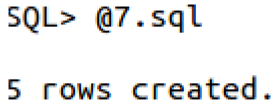
```
INSERT INTO MEDICINE VALUES('ASPIRIN',10.00,'Y');
INSERT INTO MEDICINE VALUES('P650',5.00,'Y');
INSERT INTO MEDICINE VALUES('S890',15.00,'N');
INSERT INTO MEDICINE
VALUES('VITAMIN',15.00,'Y');
INSERT INTO MEDICINE
VALUES('HOTST',10.00,'N');
```



```
SQL> @7.sql

5 rows created.
```

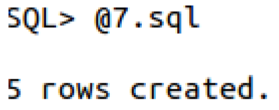
```
INSERT INTO PRESCRIPTION VALUES('001',TO_DATE('2025-03-27','YYYY-MM-DD'),'123','1234');
INSERT INTO PRESCRIPTION VALUES('002',TO_DATE('2025-01-26','YYYY-MM-DD'),'124','1235');
NSERT INTO PRESCRIPTION VALUES('003',TO_DATE('2025-03-23','YYYY-MM-DD'),'125','1236');
INSERT INTO PRESCRIPTION VALUES('004',TO_DATE('2025-03-27','YYYY-MM-DD'),'126','1237');
INSERT INTO PRESCRIPTION VALUES('005',TO_DATE('2025-03-23','YYYY-MM-DD'),'127','1238');
```



```
SQL> @7.sql

5 rows created.
```

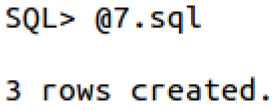
```
INSERT INTO PRESCRIPTION_MEDICINE VALUES('001','P650',5);
INSERT INTO PRESCRIPTION_MEDICINE VALUES('002','ASPIRIN',2);
INSERT INTO PRESCRIPTION_MEDICINE VALUES('003','S890',3);
INSERT INTO PRESCRIPTION_MEDICINE VALUES('004','VITAMIN',1);
INSERT INTO PRESCRIPTION_MEDICINE VALUES('005','P650',2);
```



```
SQL> @7.sql

5 rows created.
```

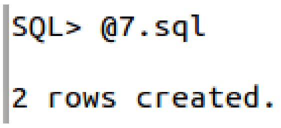
```
INSERT INTO JUNIORD VALUES('123',TO_DATE('2004-09-14','YYYY-MM-DD'),TO_DATE('2004-09-14','YYYY-MM-DD'));
INSERT INTO JUNIORD VALUES('124',TO_DATE('2004-09-14','YYYY-MM-DD'),TO_DATE('2004-09- 14','YYYY-MM-DD'));
INSERT INTO JUNIORD VALUES('125',TO_DATE('2004-09-14','YYYY-MM-DD'),TO_DATE('2004-09- 14','YYYY-MM-DD'));
```



```
SQL> @7.sql

3 rows created.
```

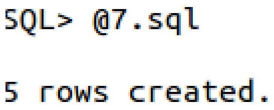
```
INSERT INTO SENIORD VALUES('126','VBHKX');
INSERT INTO SENIORD VALUES('127','CDHKX');
```



```
SQL> @7.sql

2 rows created.
```

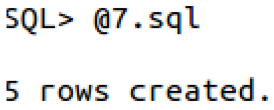
```
INSERT INTO TREAT VALUES('123','1234');
INSERT INTO TREAT VALUES('124','1235');
INSERT INTO TREAT VALUES('125','1236');
INSERT INTO TREAT VALUES('126','1237');
INSERT INTO TREAT VALUES('127','1238');
```



```
SQL> @7.sql

5 rows created.
```

```
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');
```

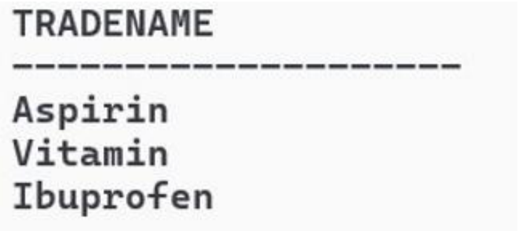


```
SQL> @7.sql

5 rows created.
```

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

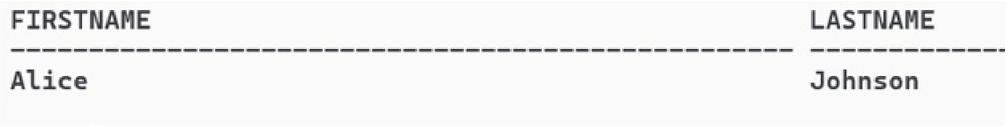
```
SELECT TRADENAME FROM MEDICINE WHERE GENERICFLAG = 'Y' AND UNITPRICE < 50;
```



```
TRADENAME
--------------------
Aspirin
Vitamin
Ibuprofen
```

2.List the first and last name of patients whose primary doctor named ‘John Smith’.

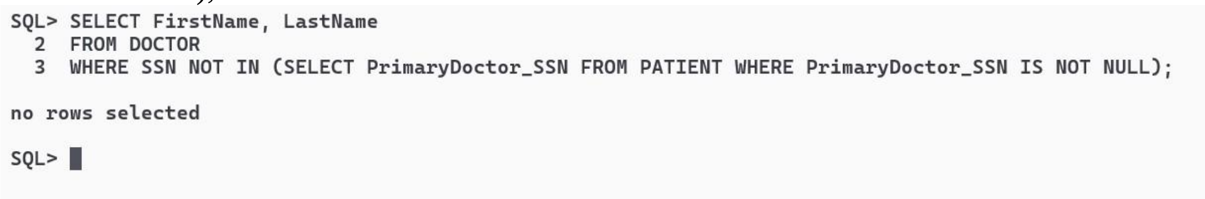
SELECT p.PFIRSTNAME, p.PLASTNAME
FROM PATIENT p
JOIN DOCTOR d ON p.PRIMARYDOCTOR_SSN = d.SSN
WHERE d.DFIRSTNAME = 'John' AND d.DLASTNAME = 'Smith';



```
FIRSTNAME                                          LASTNAME
-------------------------------------------------- ------------
Alice                                              Johnson
```

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);



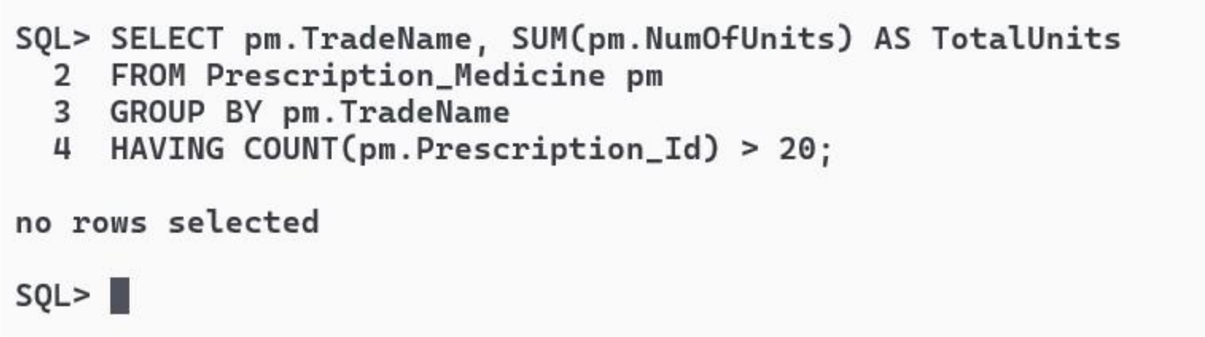
```
SQL> SELECT FirstName, LastName
     FROM DOCTOR
     WHERE SSN NOT IN (SELECT PrimaryDoctor_SSN FROM PATIENT WHERE PrimaryDoctor_SSN IS NOT NULL);

no rows selected

SQL>
```

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;



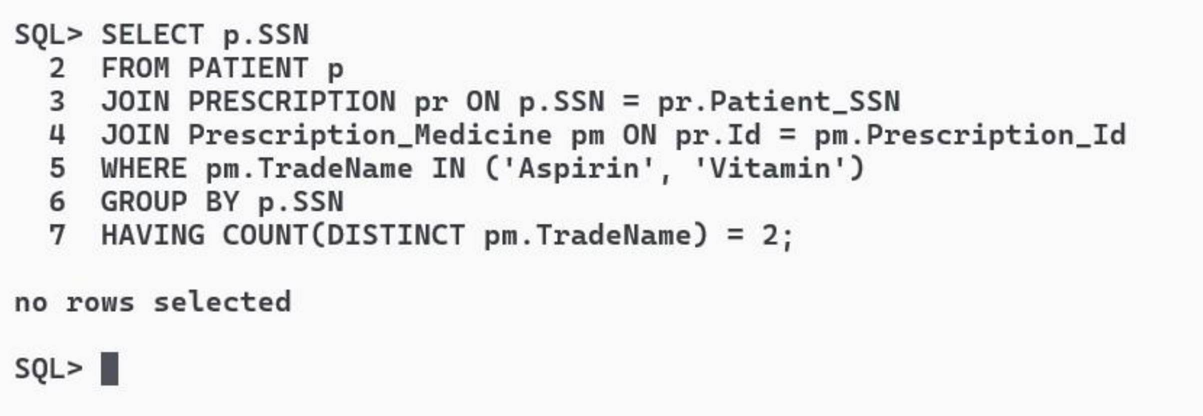
```
SQL> SELECT pm.TradeName, SUM(pm.NumOfUnits) AS TotalUnits
     FROM Prescription_Medicine pm
     GROUP BY pm.TradeName
     HAVING COUNT(pm.Prescription_Id) > 20;

no rows selected

SQL>
```

5.List the SSN of patients who have ‘Aspirin’ and ‘Vitamin’ 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;



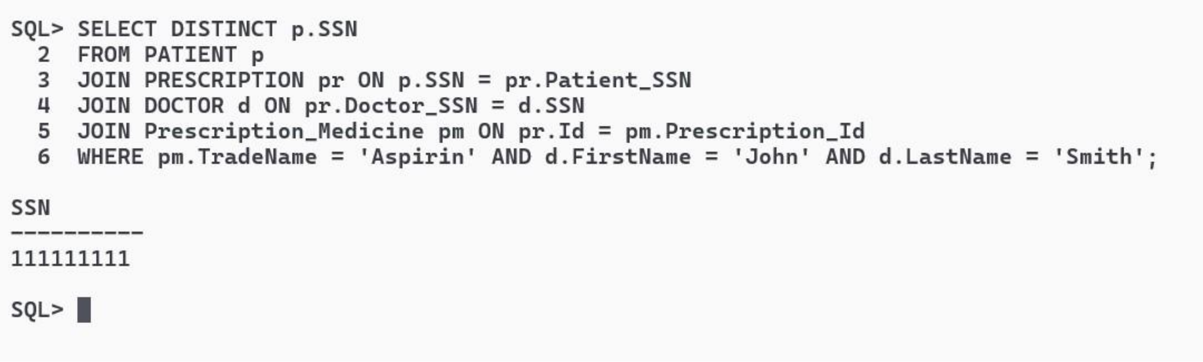
```
SQL> SELECT p.SSN
     FROM PATIENT p
     JOIN PRESCRIPTION pr ON p.SSN = pr.Patient_SSN
     JOIN Prescription_Medicine pm ON pr.Id = pm.Prescription_Id
     WHERE pm.TradeName IN ('Aspirin', 'Vitamin')
     GROUP BY p.SSN
     HAVING COUNT(DISTINCT pm.TradeName) = 2;

no rows selected

SQL>
```

6.List the SNN of distinct patients who have ‘Aspirin’ prescribed to them by doctor named ‘John Smith’.

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';



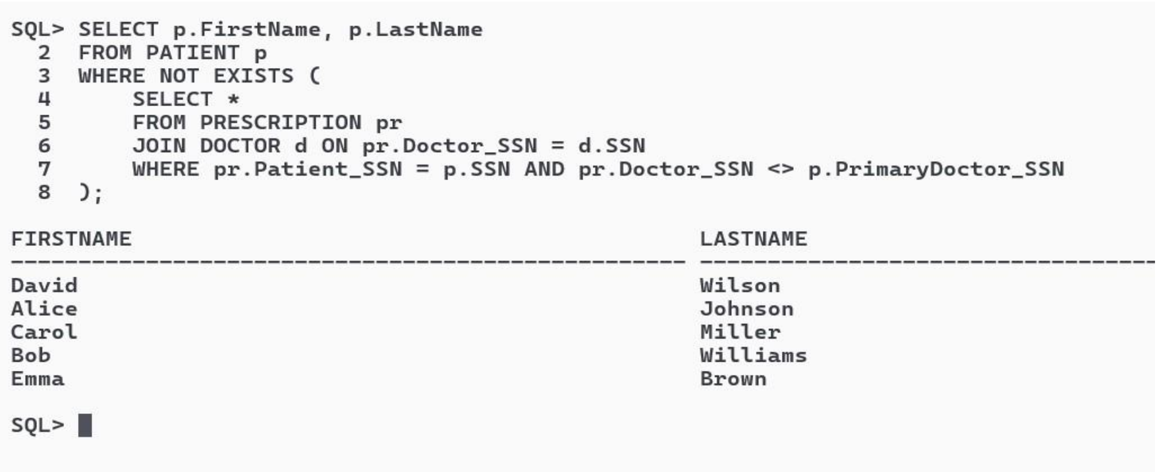
```
SQL> SELECT DISTINCT p.SSN
     FROM PATIENT p
     JOIN PRESCRIPTION pr ON p.SSN = pr.Patient_SSN
     JOIN DOCTOR d ON pr.Doctor_SSN = d.SSN
     JOIN Prescription_Medicine pm ON pr.Id = pm.Prescription_Id
     WHERE pm.TradeName = 'Aspirin' AND d.FirstName = 'John' AND d.LastName = 'Smith';

SSN
----------
111111111

SQL>
```

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);



```
SQL> 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
     );

FIRSTNAME                                          LASTNAME
-------------------------------------------------- ----------------------------------
David                                              Wilson
Alice                                              Johnson
Carol                                              Miller
Bob                                                Williams
Emma                                               Brown

SQL>
```