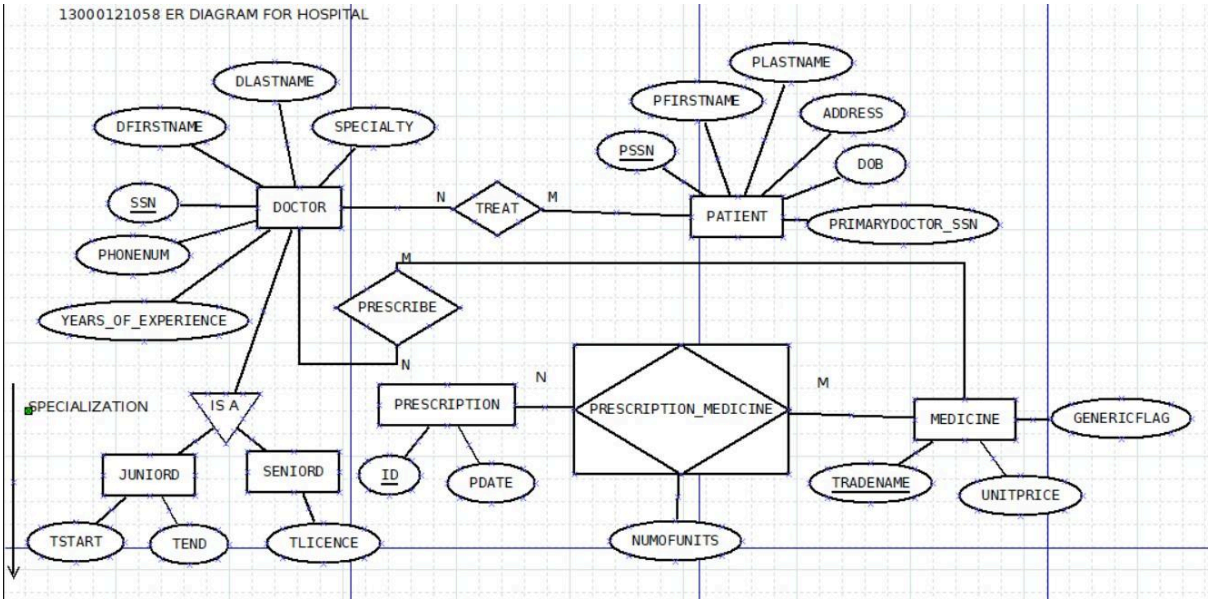
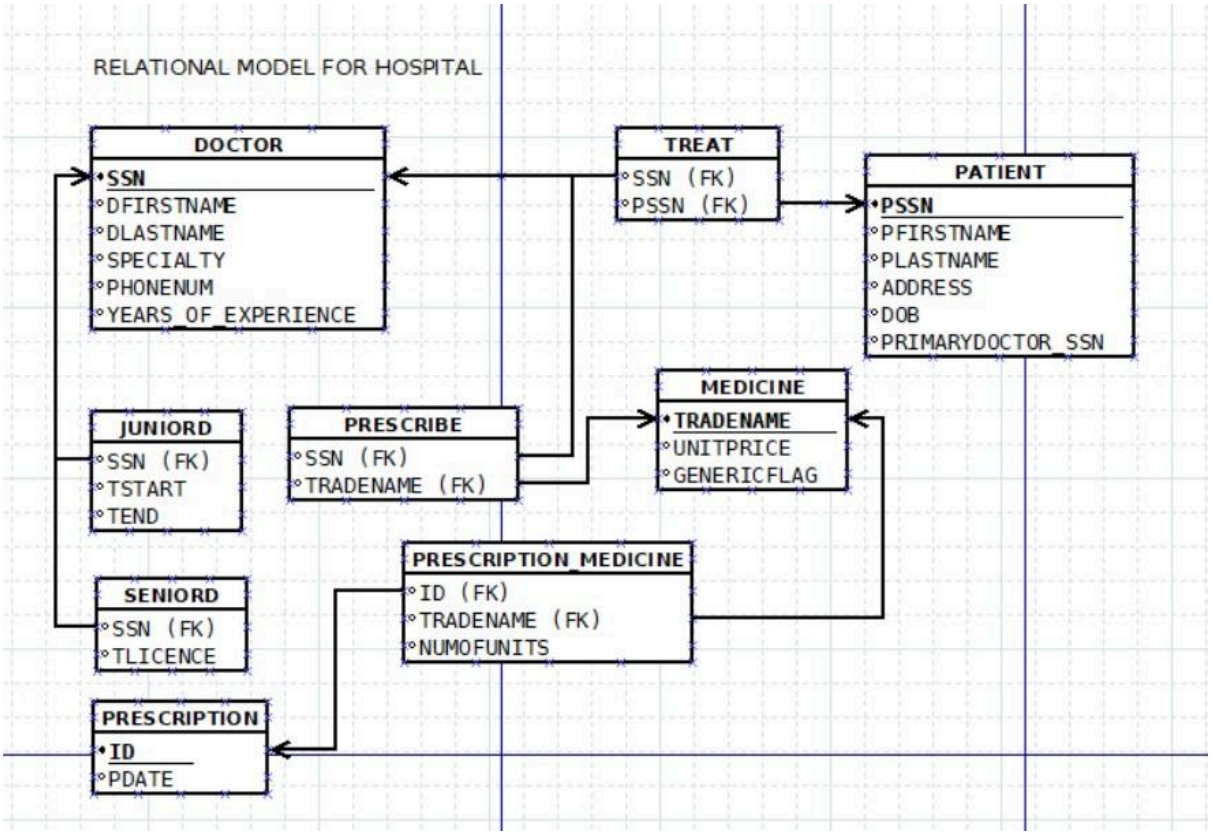
ASSIGNMENT 7

I.Design an ER diagram for an application that models a hospital doctors treat patients, prescribe tests, monitor progress etc. Analyse 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

1. Doctors can treat multiple patients, and patients can be treated by multiple doctors (many-to-many relationship).
2. Doctors can prescribe multiple tests and medications for patients.
3. Each patient's progress can be monitored through multiple progress reports.
4. Doctors can specialize in different medical fields such as surgery, pediatrics, cardiology, etc.
5. Patients can be categorized as inpatients or outpatients.
6. Tests and medications share common attributes such as ID, name, and price.
7. Each progress report includes details such as description and date.

II.Create 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,
FirstName VARCHAR2(50),
LastName VARCHAR2(50),
Specialty VARCHAR2(20),
YearsOfExperience INT,
PhoneNum VARCHAR2(15));

```
SQL> CREATE TABLE DOCTOR (
  2      SSN VARCHAR2(10) PRIMARY KEY,
  3      FirstName VARCHAR2(50),
  4      LastName VARCHAR2(50),
  5      Specialty VARCHAR2(20),
  6      YearsOfExperience INT,
  7      PhoneNum VARCHAR2(15)
  8  );

Table created.

SQL> DESC DOCTOR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SSN                                       NOT NULL VARCHAR2(10)
 FIRSTNAME                                          VARCHAR2(50)
 LASTNAME                                           VARCHAR2(50)
 SPECIALTY                                          VARCHAR2(20)
 YEARSOFEXPERIENCE                                  NUMBER(38)
 PHONENUM                                           VARCHAR2(15)

SQL>
```

CREATE TABLE PATIENT (
SSN VARCHAR2(10) PRIMARY KEY,
FirstName VARCHAR2(50),
LastName VARCHAR2(50),
Address VARCHAR2(25),
DOB DATE,
PrimaryDoctor_SSN VARCHAR(10),

CONSTRAINT PFK1 FOREIGN KEY (PrimaryDoctor_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE);

```
SQL> CREATE TABLE PATIENT (
  2      SSN VARCHAR2(10) PRIMARY KEY,
  3      FirstName VARCHAR2(50),
  4      LastName VARCHAR2(50),
  5      Address VARCHAR2(25),
  6      DOB DATE,
  7      PrimaryDoctor_SSN VARCHAR(10),
  8      CONSTRAINT PFK1 FOREIGN KEY (PrimaryDoctor_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE
  9  );

Table created.

SQL> DESC PATIENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SSN                                       NOT NULL VARCHAR2(10)
 FIRSTNAME                                          VARCHAR2(50)
 LASTNAME                                           VARCHAR2(50)
 ADDRESS                                            VARCHAR2(25)
 DOB                                                DATE
 PRIMARYDOCTOR_SSN                                  VARCHAR2(10)

SQL>
```

CREATE TABLE MEDICINE (
TradeName VARCHAR2(20) PRIMARY KEY,
UnitPrice NUMBER(10,2),
GenericFlag CHAR(1));

```
SQL> CREATE TABLE MEDICINE (
  2      TradeName VARCHAR2(20) PRIMARY KEY,
  3      UnitPrice NUMBER(10,2),
  4      GenericFlag CHAR(1)
  5  );

Table created.

SQL> DESC MEDICINE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TRADENAME                                 NOT NULL VARCHAR2(20)
 UNITPRICE                                          NUMBER(10,2)
 GENERICFLAG                                        CHAR(1)

SQL>
```

CREATE TABLE PRESCRIPTION (
Id VARCHAR2(10) PRIMARY KEY,
PDate DATE,
Doctor_SSN VARCHAR2(10),
Patient_SSN VARCHAR2(10),
CONSTRAINT PNFK1 FOREIGN KEY (Doctor_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE,
CONSTRAINT PNFK2 FOREIGN KEY (Patient_SSN) REFERENCES PATIENT(SSN) ON DELETE CASCADE);

```
SQL> CREATE TABLE PRESCRIPTION (
  2      Id VARCHAR2(10) PRIMARY KEY,
  3      PDate DATE,
  4      Doctor_SSN VARCHAR2(10),
  5      Patient_SSN VARCHAR2(10),
  6      CONSTRAINT PNFK1 FOREIGN KEY (Doctor_SSN) REFERENCES DOCTOR(SSN) ON DELETE CASCADE,
  7      CONSTRAINT PNFK2 FOREIGN KEY (Patient_SSN) REFERENCES PATIENT(SSN) ON DELETE CASCADE
  8  );

Table created.

SQL> DESC PRESCRIPTION;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL VARCHAR2(10)
 PDATE                                              DATE
 DOCTOR_SSN                                         VARCHAR2(10)
 PATIENT_SSN                                        VARCHAR2(10)

SQL>
```

CREATE TABLE Prescription_Medicine (
Prescription_Id VARCHAR2(10),
TradeName VARCHAR2(20),
NumOfUnits NUMBER,
CONSTRAINT PMFK1 FOREIGN KEY (Prescription_Id) REFERENCES Prescription(Id) ON DELETE CASCADE,
CONSTRAINT PMFK2 FOREIGN KEY (TradeName) REFERENCES Medicine(TradeName) ON DELETE CASCADE);

```
SQL> CREATE TABLE Prescription_Medicine (
  2      Prescription_Id VARCHAR2(10),
  3      TradeName VARCHAR2(20),
  4      NumOfUnits NUMBER,
  5      CONSTRAINT PMFK1 FOREIGN KEY (Prescription_Id) REFERENCES Prescription(Id) ON DELETE CASCADE,
  6      CONSTRAINT PMFK2 FOREIGN KEY (TradeName) REFERENCES Medicine(TradeName) ON DELETE CASCADE
  7  );

Table created.

SQL> DESC PRESCRIPTION_MEDICINE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRESCRIPTION_ID                                    VARCHAR2(10)
 TRADENAME                                          VARCHAR2(20)
 NUMOFUNITS                                         NUMBER

SQL>
```

CREATE TABLE JUNIORD (
SSN VARCHAR2(10),
TSTART DATE,
TEND DATE,
CONSTRAINT JFK1 FOREIGN KEY (SSN) REFERENCES DOCTOR(SSN));

```
SQL> CREATE TABLE JUNIORD (
  2     SSN VARCHAR2(10),
  3     TSTART DATE,
  4     TEND DATE,
  5     CONSTRAINT JFK1 FOREIGN KEY (SSN) REFERENCES DOCTOR(SSN)
  6  );

Table created.

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

SQL> INSERT ALL
  2  INTO JUNIORD VALUES ('123456789', TO_DATE('2023-01-01', 'YYYY-MM-DD'), TO_DATE('2024-12-31', 'YYYY-MM-DD'))
  3  INTO JUNIORD VALUES ('234567890', TO_DATE('2023-02-01', 'YYYY-MM-DD'), TO_DATE('2024-11-30', 'YYYY-MM-DD'))
  4  INTO JUNIORD VALUES ('345678901', TO_DATE('2023-03-01', 'YYYY-MM-DD'), TO_DATE('2024-10-31', 'YYYY-MM-DD'))
  5  SELECT * FROM DUAL;

3 rows created.

SQL> SELECT * FROM JUNIORD;

SSN        TSTART    TEND
---------- --------- ---------
123456789  01-JAN-23 31-DEC-24
234567890  01-FEB-23 30-NOV-24
345678901  01-MAR-23 31-OCT-24

SQL>
```

CREATE TABLE SENIORD (
SSN VARCHAR2(10),
TLICENCE VARCHAR2(10),
CONSTRAINT SFK1 FOREIGN KEY (SSN) REFERENCES DOCTOR(SSN));

```
SQL> CREATE TABLE SENIORD (
  2     SSN VARCHAR2(10),
  3     TLICENCE VARCHAR2(10),
  4     CONSTRAINT SFK1 FOREIGN KEY (SSN) REFERENCES DOCTOR(SSN)
  5  );

Table created.

SQL> DESC SENIORD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SSN                                                VARCHAR2(10)
 TLICENCE                                           VARCHAR2(10)

SQL> INSERT ALL
  2  INTO SENIORD VALUES ('456789012', 'G45678')
  3  INTO SENIORD VALUES ('567890123', 'G56789')
  4  SELECT * FROM dual;

2 rows created.

SQL> SELECT * FROM SENIORD;

SSN        TLICENCE
---------- ----------
456789012  G45678
567890123  G56789

SQL>
```

CREATE TABLE TREAT (
DSSN VARCHAR2(10),

PSSN VARCHAR2(10),
CONSTRAINT TRFK1 FOREIGN KEY (DSSN) REFERENCES DOCTOR(SSN),
CONSTRAINT TRFK2 FOREIGN KEY (PSSN) REFERENCES PATIENT(SSN));

```
SQL> CREATE TABLE TREAT (
  2     DSSN VARCHAR2(10),
  3     PSSN VARCHAR2(10),
  4     CONSTRAINT TRFK1 FOREIGN KEY (DSSN) REFERENCES DOCTOR(SSN),
  5     CONSTRAINT TRFK2 FOREIGN KEY (PSSN) REFERENCES PATIENT(SSN)
  6  );

Table created.

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

SQL> INSERT ALL
  2  INTO TREAT VALUES ('123456789', '111111111')
  3  INTO TREAT VALUES ('234567890', '222222222')
  4  INTO TREAT VALUES ('345678901', '333333333')
  5  INTO TREAT VALUES ('456789012', '444444444')
  6  INTO TREAT VALUES ('567890123', '555555555')
  7  SELECT * FROM dual;

5 rows created.

SQL> SELECT * FROM TREAT;

DSSN       PSSN
---------- ----------
123456789  111111111
234567890  222222222
345678901  333333333
456789012  444444444
567890123  555555555

SQL>
```

CREATE TABLE PRESCRIBE (
DSSN VARCHAR2(10),
MTRADENAME VARCHAR2(20),
CONSTRAINT PRFK1 FOREIGN KEY (DSSN) REFERENCES DOCTOR(SSN),
CONSTRAINT PRFK2 FOREIGN KEY (MTRADENAME) REFERENCES MEDICINE(TradeName));

```
SQL> CREATE TABLE PRESCRIBE (
  2     DSSN VARCHAR2(10),
  3     MTRADENAME VARCHAR2(20),
  4     CONSTRAINT PRFK1 FOREIGN KEY (DSSN) REFERENCES DOCTOR(SSN),
  5     CONSTRAINT PRFK2 FOREIGN KEY (MTRADENAME) REFERENCES MEDICINE(TradeName)
  6  );

Table created.

SQL> DESC PRESCRIBE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DSSN                                               VARCHAR2(10)
 MTRADENAME                                         VARCHAR2(20)

SQL> INSERT ALL
  2  INTO PRESCRIBE VALUES ('123456789', 'Aspirin')
  3  INTO PRESCRIBE VALUES ('234567890', 'Paracetamol')
  4  INTO PRESCRIBE VALUES ('345678901', 'Ibuprofen')
  5  INTO PRESCRIBE VALUES ('456789012', 'Amoxicillin')
  6  INTO PRESCRIBE VALUES ('567890123', 'Vitamin')
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM PRESCRIBE;

DSSN       MTRADENAME
---------- --------------------
123456789  Aspirin
234567890  Paracetamol
345678901  Ibuprofen
456789012  Amoxicillin
567890123  Vitamin

SQL>
```

INSERT ALL
INTO DOCTOR VALUES ('123456789', 'John', 'Smith', 'Cardiology', 15, '123-456-7890')
INTO DOCTOR VALUES ('234567890', 'Jane', 'Doe', 'Pediatrics', 10, '234-567-8901')
INTO DOCTOR VALUES ('345678901', 'David', 'Johnson', 'Orthopedics', 20, '345-678-9012')
INTO DOCTOR VALUES ('456789012', 'Emily', 'Brown', 'Oncology', 8, '456-789-0123')
INTO DOCTOR VALUES ('567890123', 'Michael', 'Davis', 'Internal Medicine', 12, '567-890-1234')
SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO DOCTOR VALUES ('123456789', 'John', 'Smith', 'Cardiology', 15, '123-456-7890')
  3  INTO DOCTOR VALUES ('234567890', 'Jane', 'Doe', 'Pediatrics', 10, '234-567-8901')
  4  INTO DOCTOR VALUES ('345678901', 'David', 'Johnson', 'Orthopedics', 20, '345-678-9012')
  5  INTO DOCTOR VALUES ('456789012', 'Emily', 'Brown', 'Oncology', 8, '456-789-0123')
  6  INTO DOCTOR VALUES ('567890123', 'Michael', 'Davis', 'Internal Medicine', 12, '567-890-1234')
  7  SELECT * FROM dual;

5 rows created.
```

INSERT ALL
INTO PATIENT VALUES ('111111111', 'Alice', 'Johnson', '123 Main St',
TO_DATE('1990-05-15', 'YYYY-MM-DD'), '123456789')

INTO PATIENT VALUES ('222222222', 'Bob', 'Williams', '456 Elm St',
TO_DATE('1985-08-20', 'YYYY-MM-DD'), '234567890')
INTO PATIENT VALUES ('333333333', 'Carol', 'Miller', '789 Oak St',
TO_DATE('1975-12-10', 'YYYY-MM-DD'), '345678901')
INTO PATIENT VALUES ('444444444', 'David', 'Wilson', '101 Pine St',
TO_DATE('2000-03-25', 'YYYY-MM-DD'), '456789012')
INTO PATIENT VALUES ('555555555', 'Emma', 'Brown', '202 Cedar St',
TO_DATE('1998-09-30', 'YYYY-MM-DD'), '567890123')
SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO PATIENT VALUES ('111111111', 'Alice', 'Johnson', '123 Main St', TO_DATE('1990-05-15', 'YYYY-MM-DD'), '123456789')
  3  INTO PATIENT VALUES ('222222222', 'Bob', 'Williams', '456 Elm St', TO_DATE('1985-08-20', 'YYYY-MM-DD'), '234567890')
  4  INTO PATIENT VALUES ('333333333', 'Carol', 'Miller', '789 Oak St', TO_DATE('1975-12-10', 'YYYY-MM-DD'), '345678901')
  5  INTO PATIENT VALUES ('444444444', 'David', 'Wilson', '101 Pine St', TO_DATE('2000-03-25', 'YYYY-MM-DD'), '456789012')
  6  INTO PATIENT VALUES ('555555555', 'Emma', 'Brown', '202 Cedar St', TO_DATE('1998-09-30', 'YYYY-MM-DD'), '567890123')
  7  SELECT * FROM dual;

5 rows created.

SQL>
```

INSERT ALL
INTO MEDICINE VALUES ('Aspirin', 10.00, 'Y')
INTO MEDICINE VALUES ('Vitamin', 5.00, 'Y')
INTO MEDICINE VALUES ('Paracetamol', 15.00, 'N')
INTO MEDICINE VALUES ('Amoxicillin', 20.00, 'N')
INTO MEDICINE VALUES ('Ibuprofen', 8.00, 'Y')
SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO MEDICINE VALUES ('Aspirin', 10.00, 'Y')
  3  INTO MEDICINE VALUES ('Vitamin', 5.00, 'Y')
  4  INTO MEDICINE VALUES ('Paracetamol', 15.00, 'N')
  5  INTO MEDICINE VALUES ('Amoxicillin', 20.00, 'N')
  6  INTO MEDICINE VALUES ('Ibuprofen', 8.00, 'Y')
  7  SELECT * FROM dual;

5 rows created.

SQL>
```

INSERT ALL
INTO PRESCRIPTION VALUES ('00001', TO_DATE('2024-03-19', 'YYYY-MM-DD'),
'123456789', '111111111')
INTO PRESCRIPTION VALUES ('00002', TO_DATE('2024-03-20', 'YYYY-MM-DD'),
'234567890', '222222222')
INTO PRESCRIPTION VALUES ('00003', TO_DATE('2024-03-21', 'YYYY-MM-DD'),
'345678901', '333333333')

INTO PRESCRIPTION VALUES ('00004', TO_DATE('2024-03-22', 'YYYY-MM-DD'), '456789012', '444444444')
INTO PRESCRIPTION VALUES ('00005', TO_DATE('2024-03-23', 'YYYY-MM-DD'), '567890123', '555555555')
SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO PRESCRIPTION VALUES ('00001', TO_DATE('2024-03-19', 'YYYY-MM-DD'), '123456789', '111111111')
  3  INTO PRESCRIPTION VALUES ('00002', TO_DATE('2024-03-20', 'YYYY-MM-DD'), '234567890', '222222222')
  4  INTO PRESCRIPTION VALUES ('00003', TO_DATE('2024-03-21', 'YYYY-MM-DD'), '345678901', '333333333')
  5  INTO PRESCRIPTION VALUES ('00004', TO_DATE('2024-03-22', 'YYYY-MM-DD'), '456789012', '444444444')
  6  INTO PRESCRIPTION VALUES ('00005', TO_DATE('2024-03-23', 'YYYY-MM-DD'), '567890123', '555555555')
  7  SELECT * FROM dual;

5 rows created.

SQL>
```

INSERT ALL
INTO Prescription_Medicine VALUES ('00001', 'Aspirin', 2)
INTO Prescription_Medicine VALUES ('00002', 'Vitamin', 1)
INTO Prescription_Medicine VALUES ('00003', 'Paracetamol', 3)
INTO Prescription_Medicine VALUES ('00004', 'Amoxicillin', 2)
INTO Prescription_Medicine VALUES ('00005', 'Ibuprofen', 2)
SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO Prescription_Medicine VALUES ('00001', 'Aspirin', 2)
  3  INTO Prescription_Medicine VALUES ('00002', 'Vitamin', 1)
  4  INTO Prescription_Medicine VALUES ('00003', 'Paracetamol', 3)
  5  INTO Prescription_Medicine VALUES ('00004', 'Amoxicillin', 2)
  6  INTO Prescription_Medicine VALUES ('00005', 'Ibuprofen', 2)
  7  SELECT * FROM dual;

5 rows created.

SQL>
```

INSERT ALL
INTO JUNIORD VALUES ('123456789', TO_DATE('2023-01-01', 'YYYY-MM-DD'), TO_DATE('2024-12-31', 'YYYY-MM-DD'))
INTO JUNIORD VALUES ('234567890', TO_DATE('2023-02-01', 'YYYY-MM-DD'), TO_DATE('2024-11-30', 'YYYY-MM-DD'))
INTO JUNIORD VALUES ('345678901', TO_DATE('2023-03-01', 'YYYY-MM-DD'), TO_DATE('2024-10-31', 'YYYY-MM-DD'))
SELECT * FROM DUAL;

INSERT ALL
INTO SENIORD VALUES ('456789012', 'G45678')
INTO SENIORD VALUES ('567890123', 'G56789')
SELECT * FROM dual;

```
INSERT ALL
INTO TREAT VALUES ('123456789', '111111111')
INTO TREAT VALUES ('234567890', '222222222')
INTO TREAT VALUES ('345678901', '333333333')
INTO TREAT VALUES ('456789012', '444444444')
INTO TREAT VALUES ('567890123', '555555555')
SELECT * FROM DUAL;
```

```
INSERT ALL
INTO PRESCRIBE VALUES ('123456789', 'Aspirin')
INTO PRESCRIBE VALUES ('234567890', 'Paracetamol')
INTO PRESCRIBE VALUES ('345678901', 'Ibuprofen')
INTO PRESCRIBE VALUES ('456789012', 'Amoxicillin')
INTO PRESCRIBE VALUES ('567890123', 'Vitamin')
SELECT * FROM DUAL;
```

III.Consider the following relations run the following SQL queries :
Doctor(SSN, FirstName, LastName, Specialty, YearsOfExperience, PhoneNum)
Patient(SSN, FirstName, LastName, Address, DOB, PrimaryDoctor_SSN)
Medicine(TradeName, UnitPrice, GenericFlag)
Prescription(Id, Date, Doctor_SSN, Patient_SSN)
Prescription_Medicine(Prescription Id, TradeName, NumOfUnits)

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

```
SELECT TradeName
FROM MEDICINE
WHERE GenericFlag = 'Y' AND UnitPrice < 50;
```

```
SQL> SELECT TradeName
  2  FROM MEDICINE
  3  WHERE GenericFlag = 'Y' AND UnitPrice < 50;

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

SQL>
```

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

```
SELECT p.FirstName, p.LastName
FROM PATIENT p
JOIN DOCTOR d ON p.PrimaryDoctor_SSN = d.SSN
```

WHERE d.FirstName = 'John' AND d.LastName = 'Smith';

```
SQL> SELECT p.FirstName, p.LastName
  2  FROM PATIENT p
  3  JOIN DOCTOR d ON p.PrimaryDoctor_SSN = d.SSN
  4  WHERE d.FirstName = 'John' AND d.LastName = 'Smith';

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

SQL>
```

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
  2  FROM DOCTOR
  3  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
  2  FROM Prescription_Medicine pm
  3  GROUP BY pm.TradeName
  4  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
  2  FROM PATIENT p
  3  JOIN PRESCRIPTION pr ON p.SSN = pr.Patient_SSN
  4  JOIN Prescription_Medicine pm ON pr.Id = pm.Prescription_Id
  5  WHERE pm.TradeName IN ('Aspirin', 'Vitamin')
  6  GROUP BY p.SSN
  7  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
  2  FROM PATIENT p
  3  JOIN PRESCRIPTION pr ON p.SSN = pr.Patient_SSN
  4  JOIN DOCTOR d ON pr.Doctor_SSN = d.SSN
  5  JOIN Prescription_Medicine pm ON pr.Id = pm.Prescription_Id
  6  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
  2  FROM PATIENT p
  3  WHERE NOT EXISTS (
  4      SELECT *
  5      FROM PRESCRIPTION pr
  6      JOIN DOCTOR d ON pr.Doctor_SSN = d.SSN
  7      WHERE pr.Patient_SSN = p.SSN AND pr.Doctor_SSN <> p.PrimaryDoctor_SSN
  8  );

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

SQL>
```