CS3431: Project Description
B-term, 2011
Homework 4 Solution

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)

- The **Doctor** relation has attributes Social Security Number (SSN), first and last names, specialty, the number of experience years, and the phone number.
- The **Patient** relation has attributes SSN, first and last names, address, date of birth (DOB), and the SSN of the patient's primary doctor.
- The **Medicine** relation has attributes trade name, unit price, and whether or not the medicine is generic (True or False).
- The **Prescription** relation has attributes the prescription id, the date in which the prescription is written, the SSN of the doctor who wrote the prescription, and the SSN of the patient to whom the prescription is written.
- The **Prescription_Medicine** relation stores the medicines written in each prescription along with their quantities (number of units).

Problem 1 (SQL Queries)

Q1: Report the first and last names of patients whose primary doctor is "John Smith".

```
Select  P.FirstName, P.LastName
From Patient P, Doctor D
Where  P.PrimaryDoctor_SSN = D.SSN
And  D.FirstName = 'John'
And  D.LastName = 'Smith';
```

Q2: For each medicine (TradeName), report the trade name, the number of prescriptions it is written in, and the total number of units prescribed.

```
Select  TradeName, count(*), sum(NumOfUnits)
From Prescription_Medicine
Group By TradeName;
```

Q3: Report the first and last names of the patient who has the most prescriptions.

```
Select
From Patient
Where SSN not in (
        Select A.ssn
        From
            (Select Patient_SSN As ssn, count(*) As cnt
             From Prescription P
             Group By Patient_SSN) A,
            (Select Patient_SSN As ssn, count(*) As cnt
             From Prescription P
             Group By Patient_SSN) B
        Where A.cnt < B.cnt
        );
```

Another solution:

```
Select
From Patient
Where SSN = (
        Select A.ssn
        From
            (Select Patient_SSN As ssn, count(*) As cnt
             From Prescription P
             Group By Patient_SSN
             Order By cnt Desc) A
         Where rownum = 1
        );
```

Q4: Report the first and last names of the patient who have more than 10 prescriptions. Order the list based on the last name and then the first name.

```
Select FirstName, LastName
From Patient
Where SSN in (
                Select Patient_SSN
                From Prescription P
                Group By Patient_SSN
                Having count(id) > 10)
Order By LastName, FirstName;
```

Problem 2 (SQL Views)

Q1: Create a view that reports the trade name, unit price, and the generic flag of the most expensive and cheapest medicines.

```
Create View MedicineView AS
Select TradeName, unitPrice, genericFlag
From Medicine
Where unitPrice in (
                  Select max(unitPrice) price
                  From Medicine
                  Union
                  Select min(unitPrice) price
                  From Medicine
                );
```

Q2: Using the view you created in the previous step, write a query that reports the SSN of doctors who prescribed the most expensive or the cheapest medicine (the query should reference the view and may reference other tables as well).

```
Select P.Doctor.SSN
From Prescription P, Prescription_Medicine PM, MedicinceView V
Where P.Id = PM.Prescription_Id
And PM.tradename = V.tradename;
```

Problem 3 (Triggers) [20 Points]

Write the needed trigger(s) that ensure that if NumOfUnits in Prescription_Medicine is less than or equal to zero or null, then the value should be automatically set to 1.

```
Create Or Replace Trigger
Before Insert or Update Of (NumOfUnits) On Prescription_Medicine
For Each Row

Begin
    If ((:new.NumOfUnits is null) or (:new.NumOfUnits <= 0)) Then
            :new.NumOfUnits := 1;
    End If;
End;
```

Problem 4 (Assertions) [20 Points (10 Points each sub-question)]

Q1: Write an assertion that makes sure that no patient takes 'Aspirin' and 'Vitamin' trade names in the same prescription.

```
Create Assertion lessThanFive AS
   (Not Exists(
          Select Prescription_id
          From Prescription_Medicine
          Where tradename = 'Aspirin'
         Intersect
          Select Prescription_id
          From Prescription_Medicine
          Where tradename = 'Vitamin'));
```

Q2: Write an assertion that makes sure that no doctor is the primary doctor to more than 5 patients.

```
Create Assertion lessThanFive AS
    (Not Exists(
          Select *
          From Patient
          Group By PrimaryDoctor_SSN
          Having count(*) > 5));
```