

CS561: Advanced Topics In Database Systems
Spring-2012

Solution of Homework 3

Question 1

1)

1	{A},{B},{C},{D},{E},{F},{G}	{A,5/8},{B,4/8},{C,5/8},{D,6/8},{F,4/8},{G,5/8}
2	{A,B},{A,C},{A,D},{A,F},{A,G}, {B,C},{B,D},{B,F},{B,G}, {C,D},{C,F},{C,G}, {D,F},{D,G}, {F,G}	{A,B,3/8},{A,C,3/8},{A,D,4/8}, {C,D,4/8},{C,G,3/8}, {D,F,4/8},{D,G,3/8}
3	{A,C,D},{C,D,G}	{A,C,D,3/8}

2)

Support (A,BD)=2/8

Confidence (A,BD)= Support(A,BD)/Support(A)= 2/8 / 5/8 = 2/5

Support (BD,AC)=1/8

Confidence (BD,AC)= Support(BD,AC)/Support(BD)=1/8 / 2/8= 1/2

Support (A,CD)=3/8

Confidence (A,CD)= Support(A,CD)/ Support(A)= 3/8 / 5/8= 3/5

Question 2

1) Creating Tables

```
--DOCTOR
CREATE TABLE Doctor (
  SSN                NUMBER(9) NOT NULL,
  FirstName          VARCHAR2(255) NOT NULL,
  LastName           VARCHAR2(300) NOT NULL,
  Speciality         VARCHAR2(2000),
  YearsOfExperience  NUMBER(2) DEFAULT 0 NOT NULL,
  City               VARCHAR2(255) NOT NULL
);
ALTER TABLE Doctor ADD CONSTRAINT PK_Doctor_SSN PRIMARY KEY(SSN);
-- =====

--PATIENT
CREATE TABLE Patient (
  SSN                NUMBER(9) NOT NULL,
  FirstName          VARCHAR2(255) NOT NULL,
  LastName           VARCHAR2(300) NOT NULL,
  Address            VARCHAR2(2000),
  DOB                DATE NOT NULL,
  PrimaryDoctor_SSN NUMBER(9) NOT NULL
);
ALTER TABLE Patient ADD CONSTRAINT PK_Patient_SSN PRIMARY KEY(SSN);
ALTER TABLE Patient ADD CONSTRAINT FK_Doctor_SSN FOREIGN KEY(PrimaryDoctor_SSN)
REFERENCES Doctor(SSN);
-- =====

--MEDICINE
CREATE TABLE Medicine(
  TradeName          VARCHAR2(255),
  UnitPrice          NUMBER(14,2) DEFAULT 0 NOT NULL,
  GenericFlag        NUMBER(1) DEFAULT 0 );
ALTER TABLE Medicine ADD CONSTRAINT PK_Tradename PRIMARY KEY(TradeName);
-- =====

--PRESCRIPTION
CREATE TABLE Prescription (
  Id                 NUMBER(14) NOT NULL,
  Prescription_Date  DATE NOT NULL,
  Doctor_SSN         NUMBER(9) NOT NULL,
  Patient_SSN        NUMBER(9) NOT NULL,
  TotalCost          NUMBER(14,2) NOT NULL);

ALTER TABLE Prescription ADD CONSTRAINT PK_Id PRIMARY KEY(Id);
```

```

ALTER TABLE Prescription ADD CONSTRAINT FK_Prescription_Doctor_SSN FOREIGN KEY(Doctor_SSN)
REFERENCES Doctor(SSN);
ALTER TABLE Prescription ADD CONSTRAINT FK_Prescription_Patient_SSN FOREIGN KEY(Patient_SSN)
REFERENCES Patient(SSN);
-- =====

--PRESCRIPTION_MEDICINE
CREATE TABLE Prescription_Medicine (
    Prescription_Id          NUMBER(9),
    TradeName                VARCHAR2(255) NOT NULL,
    NumOfUnits               NUMBER(9,2)
);
ALTER TABLE Prescription_Medicine ADD CONSTRAINT PK_Prescription_Id_TrdName PRIMARY
KEY(Prescription_Id,TradeName);
ALTER TABLE Prescription_Medicine ADD CONSTRAINT FK_Prescription_Medicine_Id FOREIGN
KEY(Prescription_Id) REFERENCES Prescription(Id);
ALTER TABLE Prescription_Medicine ADD CONSTRAINT FK_Presc_Med_TrdName FOREIGN
KEY(TradeName) REFERENCES Medicine(TradeName);
-- =====

```

2) Needed Triggers (The code is left for your exercise)

- Two triggers are needed (they can be merged into one piece of code)
 - Before Insert (per row) on Patient table: Makes sure the newly inserted PrimaryDoctor_SSN does not already have 5 patients.
 - Before Update (per row) on Patient Table (PrimaryDoctor_SSN) column: Makes sure that the updated PrimaryDoctor_SSN does not already have 5 patients.
- Several triggers are needed as follows:
 - After Insert|Update|Delete (per row) on Prescription_Medicine table: Makes sure whenever a new record is inserted, updated, or deleted, the prescription_id of that row will have its TotalCost re-computed and updated.
 - After Update (per row) on Medicine Table: Makes sure if the UnitPrice has changed, then the TotalCost of any prescription containing the updated TradeName is re-computed.

3) OLAP Queries

- **Query 1**

Select D.city, M.TradeName, Sum(NumOfUnits) TotalUnits, Sum(NumOfUnits * UnitPrice) TotalCost
From Doctor D, Prescription P, Prescription_Medicine PM, Medicine M
Where D.SSN = P.Doctor_SSN
And P.Id = PM.Prescription_Id
And PM.TradeName = M.TradeName
Group By D.city, M.TradeName;

- **Query 2**

Select D.Specialty, D.YearsOfExperience, Count(P.Id) PrescriptionCount,
Count(Distinct Patient_SSN) PatientCount
From Doctor D, Prescription P
Where D.SSN = P.Doctor.SSN
Group By D.Specialty, D.YearsOfExperience;