CS 386/586 Winter 2013 Assignment 5

Assigned: Wednesday, February 13, 2013

Due: Wednesday, March 6, 2013 at midnight

General Information:
Submission: you **must post a note** to the "**instructors**" with the **attachment** (listed below) in Piazza. You **must** put your note with the attachment in the **turn-in-assign-here5** folder.

Assignment:

1. Create two relations where student is identified by id and advisor is identified by id:

   Student (id, first, last, advisor_id)
   Advisor (id, first, last, student_id)

   Each student must have an advisor and each advisor must have a student assistance – identified by student-id. Define the required foreign key and not null constraints.

   Demonstrate that you cannot add a new student with a new advisor in the student table. Also, demonstrate that you cannot add a new advisor with a new student in the advisor table. (You also can't add a student with a null advisor or an advisor with a null student assistant. We'll learn about how to overcome these problems in about two weeks.)

   Provide SQL statements and screen shots.

2. Consider the following relations:

   Title (ISBN, title, author, yearPublished, publisher)
   Copy (ISBN, copyNum, yearPurchased, onLoan)
   Member (id, first, last, age, gender)
   Loan (memberId, ISBN, copyNum, outDate, dueDate)

   The Title relation has one row for each book in the library listed, with its ISBN, title, year it is published, and publisher. A book can have many physical copies. Each copy has a copy number and is listed in the Copy relation. A member of this library has a unique ID and is able to check out a copy of a book, represented by a row in the Loan table.

   a. Write SQL statements to create the tables with appropriate primary key and foreign key constraints. Give screen shots of the results.
   b. Write tuple or attribute level CHECK constraints or Assertions to ensure that each of the following requirement is met:
      i. All the members should be at least 10 years old and should have a valid gender (of 'm' or 'f').
      ii. The year purchased for a book copy should be greater than or equal to the year the book was published.
      iii. The total copies of books loaned to a member below age 18 should not exceed 5.
   c. Write a trigger for the following requirement and implement it in the PostgreSQL: Whenever a members loans a copy of a book, it should be marked as onLoan = 'Y'.

Provide your code, screen shots for your implementation and screen shots for the execution of the trigger. You will have to add the required data in your database.

3. Write a stored procedure for correcting clearance IDs of agents assigned to a particular mission (similar to Assignment 2).

   The stored procedure should take a mission ID as an input. It should then check the access_id required for that mission and make sure that all the agents assigned to the mission have a security clearance that is less than or equal to the mission clearance. If not, update the agent's clearance to the required level. For example, if the mission clearance (i.e., access_id) is 3 and the agent's clearance_id is 4, then the agent's clearance_id should be changed to 3. If agent's clearance_id is 2, it should not be changed. The stored procedure should return the agents whose clearance ID was modified in the form of a cursor/ table with the following columns: AgentId, FirstName, LastName, OldClearanceID, NewClearanceID.

   You can either write one stored procedure for the entire functionality or you can write more than one stored procedures where one procedure invokes another (making it like an application) or you can write a stored procedure(s) and invoke it(them) from your C/ Java program in Assignment 2.

   You need to turn in:
   - Your code
   - Instructions on how to invoke your stored procedure (if it has to be invoked through phpgadmin in order to test)
   - Procedure/ Program output for input mission ID: 1 and 5, use the data provided for Assignment 2.
   - An extract of your agent table (.csv) after executing your stored procedure for input mission IDs 1 and 5.