Fall 2010 CS 3200 Class Project: Milestone 6

The goals for this milestone are (1) to work with constraints and triggers and (2) to get started with JDBC.

This milestone is to be completed individually (i.e., no teams). You can discuss problems with other students, but you have to create all deliverables yourself from scratch. In particular, it is not allowed to copy somebody else's code or text and modify it.

The report for this milestone is due on Wednesday, **November 10 at 5pm**. For late submissions you will lose one percentage point per hour after the deadline. This milestone is worth 15% of your overall homework score. Please email the deliverables to both me and Yue. You should receive a confirmation email from either of us. If you have not received a confirmation email within 12 hours after submitting your solution or by the time of the deadline, whichever comes first, you need to email us immediately to make sure we actually received your submission. (Of course, if you submit too close to the deadline, you might receive a confirmation sometime within the next 30-60 minutes after you submitted.) If you need to send multiple files, please create a single zip file. Many other attachments types, in particular rar files, are rejected by the CCIS mail server.

Database Simplification

One goal of the data generator assignment was to make sure there is interesting data to test the queries. The other was to give everybody another opportunity to evaluate the database design. Sometimes all tables are in BCNF, but when inserting data, some aspects of the design are still awkward. We are glad that somebody asked this question: Why do we have the Date table at all? It does not add any information, but just collects all date values and creates useless foreign-key relationships. That is exactly the right approach to design: we don't want to blindly follow some default approach, but need to constantly ask if things could be improved. This results in the first task:

1. Delete the useless Date table.

Constraints and Triggers

In class we discussed integrity constraints (CHECK), assertions (ASSERTION), and triggers (TRIGGER). Recall that Milestone 2 described several application constraints that we have not yet added to our database. The goal for this milestone is to make sure that we enforce the missing constraints as much as possible. We assume the following use of our system: Before performing an action, users try to enter the information about this action into the database. If a constraint or trigger causes an error, the real-world action is cancelled. For example, if a person tries to pass an item to a non-friend of the original owner, she first tries to enter this information into the database. We want the database to prevent this update (which would tell the user to not perform this item exchange.)

2. Go through **all** the requirements of the project scenario in Milestone 2 (except the trivial ones that are constraining the data type of an attribute) and do the following:
    a. Briefly state the requirement. E.g., "Each dorm building has between 10 and 40 rooms."

   b. State if our current database implementation already enforces this requirement.
      i. If it does, state in one sentence (maybe two), **why**.
      ii. If it does not, argue in one sentence (maybe two) which of CHECK, ASSERTION, or TRIGGER you would use to enforce it, and **why**.
   c. If the requirement is not enforced yet, add your enforcement option of choice to the database. Notice that SQL Server might not support all options we discussed in class. In that case, try to find an alternative solution that achieves the same effect: you want to prevent updates that violate the constraints.
   d. If the data in your tables violates a constraint you want to add, make the corresponding changes to create a consistent database.

JDBC

3. Write a simple Java program that does the following:
   a. Connect to your SQL Server database.
   b. Run query 1 from HW 5 and print all result rows to the screen (using System.out.println).
   c. Add a new student to the Students table who lives one of the buildings.
   d. Delete another student from the Students table. This student should live in a different building.
   e. Run query 1 from HW 5 again and print all result rows to the screen.
   f. Close the connection to the database.

Deliverables

Your report should contain the following items:

1. A discussion of everything requested for part 2.
2. The SQL code showing clearly the CHECK, ASSERTION, or TRIGGER statements that you added to the database.
3. The source code of your JDBC program in a separate file so that it is easy for us to run it.
4. A screenshot of the output of your JDBC program.