Exam 2 Replacement Project (Due: Nov. 15).

1. Please first complete your HW5.
2. Using the result of your HW5 (please use the standard solution instructor provided), create the schema (with necessary constraints) of your database for this project via an SQL script file under your Oracle account. The script file must be included in your submission.
3. Populate your database tables using another SQL script file so that each table contains at least 10 tuples (These tuples shall be carefully composed, meaningful and look like real). This script must also be included in your submission.
4. Develop and implement the following applications (using **Java** and **JDBC**) to support different university personnel to use the database:
    a. For staff members: *add/delete/update/search* records in anyone table. To use this function, the staff member will be prompted for entering his/her id for authentication. If a match is found from the Staff table, a pop-up menu or a relevant function button will instantly show up so that the user can proceed to do what s/he is supposed to do. Otherwise, nothing shows up, which prevents unauthorized users to do anything with the system and the database.

    b. For faculty: shall be able to *search* any information from the database in a variety of meaningful ways (be creative!). Similar to the above access control for staff members, a faculty member's id needs to be verified before s/he can access/search the system.

    c. For students: a student can enroll him/herself into a course if the course limit has not been reached; a student may search for any personal information of himself/herself, e.g., courses enrolled, exam scores of a course, the department that offers the course, and the faculty name (but not faculty id) who teaches the course, etc (generally, any information that makes sense for students to know shall be made searchable).

    d. Integrate the above three categories of database access functions into a properly designed GUI.

Notes:

- The recommended programming language is **Java**, but if you prefer a different language, you must get preapproved by the instructor and you must implement exactly the same set of functions as described above.
- The above description of the project provides an outline of the minimum requirements of a university database and application system. You are encouraged to stay creative and proactive when you design and implement your project, enriching the functions outlined above (but your enrichment must make sense).
- Check of enrollment limit can be done by a trigger or simply through your Java program (**note** Oracle does not support *Assertion* constraint).

- The full score of this project is 100 points, but students may earn **up to a total of 10 bonus points** if the project is "professionally done" with every item carefully thought out and designed.
- **Hints to receive the bonus points**: treat the project as a ***real*** project rather than a course project; make your finish look professional and impressive.
- Your submission must include a detailed "readme.txt" file summarizing the functions you implemented for each type of users, and documenting everything needed for the TA to easily set up and test your project.
- This project takes much longer than most students may think. So please ***start early!***
- **Checklist for your submission:**
    1. All source code necessary for the TA to set up and test your project
    2. A detailed readme.txt file with concise but sufficient instructions on setting up your project
    3. SQL script file for database schema creation
    4. SQL script file for data population