CSE 532: Project 2
Talent Database
via the Object-Oriented Extensions of SQL

Spring 2016

Deadline: March 29

In this project, you will be designing and implementing the TDB database using the object-relational extensions of SQL — whatever is supported by PostgreSQL.

You will be using Java/JSP/Servlets to build an application front end to your database. Database connectivity should be done using Java/JDBC. The most convenient way to work with servlets is to use Eclipse, NetBeans, or a similar IDE.

You are required to maintain your project code in a **private** cloud Git code repository on Bitbucket.

You are allowed to work with a partner—see Section 8—but both partners must share the work equally and must be committing code to the repository regularly.

1 General Description

Please refer to Project 1 for the description of the TDB database.

2 Required Data

The data items required by your system are the same as in Project 1. However, unlike Project 1, you are to use the *object-relational* model and to utilize the object-relational features of PostgreSQL as much as possible. You should determine the relationships among tables, identify the key attributes, etc. In addition, you should associate indices with your tables to speed up processing of the given queries. Finally, you should specify and enforce referential integrity constraints and other CHECK-style constraints on the data whenever applicable.

3 Queries

The system will not maintain individual user accounts: we will assume open Web access where anyone can access the content without supplying credentials.

User queries. The queries are the same as in Project 1. However, since SQL does not support universal quantification, implement the corresponding queries using the means that SQL does support (not-exists, minus, subqueries). PostgreSQL does support the recursion feature, which is needed for some queries.

4 Interface

The interface to the system should be Web-based with the front end using JSP and the back-end using servlets. Nothing fancy is expected. A basic front page with links to invoke the required queries and to display their results would suffice. No need to enable data entry through the Web interface.

5 Code Repository

The source code of your project must be maintained in your cloud code repository. It is **important** to make frequent check-ins to the repository both as a demonstration of professionalism and also to document your history of project development. **Projects with no history will be rejected outright.** Those with insufficient history will get lower grade.

If you are working with a partner, you need a joint **private** repository on Bitbucket. Both partners must commit *equally* to that repository.

You must share the repository with me and the TA as you did in Project 1. The repository name shall have he following naming scheme:

```
CSE532p2-YourBitbucketId-YourLastName-YourPartner
```

Omit the "-YourPartner" part if working alone. Our Bitbucket ids are:

- mine: **kifer-sbu**
- TA's: `asaydin`

NOTE: Share ONLY with the above Ids. We have others on Bitbucket, but only the above will be checked for project submissions.

6 Documentation and Submission Instructions

All source files written by you should have a header with the following information:

```
/*****************************************************************************
                CSE532 -- Project 2

   File name:          XXXXX
   Author(s):          Name 1  (SBU Id)
                       Name 2  (SBU Id)
   Brief description:  ZZZZZZZZZZZZZZZZZZZZZZZZZZ
                       ZZZZZZZZZZZZZZZZZZZZZZZZZZ
*****************************************************************************/
```

The source should satisfy all the coding standards (must be well-structured, indented, commented, use reasonable naming schemes, be understandable).

Your deliverables must include a project report with the following items in this order:

- An Entity-Relationship (ER) diagram of the complete project.
- A clear description of the database scheme, including a discussion of your design decisions. Remember that your design should be *object-relational*, i.e., different from Project 1.
- Description of **integrity constraints**, including referential integrity and CHECK-constraints whenever applicable.
- All SQL CREATE TABLE/VIEW/TYPE commands used to build the database. These statements must include all the applicable FOREIGN KEY and CHECK statements.
- For each query mentioned in Section 3, the report must supply the appropriate SQL statement.

- All SQL queries must be presented in a *conveniently* readable form. A simple dump of your program's code will **not be accepted** and penalty will be applied.
- A brief user guide. Explain how to install and run your program.

To prepare the report, you can use any word processor, such as Open/Libre Office, MS Word, or LaTex. But **the only acceptable submission formats** for the project report are **PDF** or **plain text**.
The report and the source code must include the following statement at the top:

> **I (*or* We, *if working with a partner*) pledge my (*or* our) honor that all parts of this project were done by me (*or* us) alone and without collaboration with anybody else.**

7 Notifying That You Are Done

There is a Blackboard assignment, called Project 2. You should use that assignment to submit your project report (in PDF or TXT formats only) as an attachment.

When submitting the report, you will also see a text box. Please write something like "I am done" so that we'll know that this is not an intermediate version of the project report and that you are really done. If the project does not quite work or does not work at all, explain the problems in that same text box. We will then see if partial credit can be given.

In addition, if you were working with a partner, indicate that this is a joint project with such-and-such person (Name and NetID). Only one of the partners needs to submit the joint *project report*, but both partners must fill out the text box (for notification purposes). That is, while submitting, the first partner will attach the report *and* fill out the text box, while the second needs to fill out the text box only.

Note: do not submit your code through Blackboard — we will get it from your repository.

8 Teaming

You can do Project 2 in groups of at most two. If you choose to work alone, this will *not* reduce the scope of your project.

There are pitfalls in working with a partner whom you do not know well: it can be a frustrating experience to find out that your partner is a freeloader. It also sometimes happens that one of the partners gets involved in a case of academic dishonesty, and this may reflect negatively on you. So, choose your partner carefully!

The division of work between partners must be **vertical**, not horizontal. This means that the partners must collaborate on each part of the project and be on top of what the other partner is doing. Each student must be aware of and understand the techniques used by his/her partner. **Your final document must clearly state who did what part of the job.**

9 Planning Your Work

You really do not have much time and should start working right away—especially if you have gaps in the undergraduate database material, Java, have no experience with IDEs like Eclipse, or with source code control systems.

Begin your work with an Entity-Relationship design and the ER diagram. The diagram must then be translated into the object-relational model. This diagram and its object-relational rendering should become part of your project report.

The next step is to create your database and populate it with sample data (same as in Project 1). Then you should write and test the queries. Following that, start designing and implementing the Web interface. Finally, write and test the code needed to connect the interface to your database.

Use the test data set from Project 1, but you might need to add more information to ensure that your queries have enough data to work with.

10 Demos

You will need to demo your project to the TA in the week following the completion of this part of the project. There will be a signup wiki page in due time.