CS 166: Project Phase 3 Requirements

May 18, 2015

1 Introduction

In this phase you are provided a package including SQL schema, dataset, and a template user interface in Java. The dataset consists of data records that should be loaded into your database system. The template user interface in Java shows you how to connect to the Postgres database and how to execute SQL queries.

Additional extra points will be awarded to projects that use GUI interfaces and properly handle exceptional situation by providing user-friendly error messages.

Please follow the below steps to get started:

1. Make sure that your environment is setup correctly: variables $DB_NAME, $PGDATA, $PGPORT are defined, Postgres instance is started correctly

2. Download CS166_Project.zip from iLearn

3. In the download directory execute the following command:

   ```
   unzip CS166_Project.zip
   ```

   You will see that a directory named "CS166_Project" will be created. This directory contains all necessary files to get started. More specifically it contains the following things:

   - **CS166_Project/data** - contains the files, which will be used to populate your database. The data file is in excel format. You might have to do some formatting changes before you can load the files to your database.
   - **CS166_Project/sql/src/create_tables.sql** - SQL script creating the database relational schema. It also includes the commands to drop these tables. Primary Key constraint is already added. Please add the foreign key constraint before you start.
   - **CS166_Project/sql/src/create_indexes.sql** - SQL script which creates database indexes. Initially is empty, you should add all your indexes to this file.
   - **CS166_Project/sql/src/load_data.sql** - SQL script for loading the data in your tables. You need to write this after formatting the data file. **Note that the file paths have to be changed to absolute paths in order to make it work**.
   - **CS166_Project/sql/scripts/create_db.sh** - shell script, which you should to setup your database.

- **CS166_Project/java/src/ProfNetwork.java** - A basic java User Interface to your Postgres database. You should modify this program and write all your SQL-specific code there.
- **CS166_Project/java/scripts/compile.sh** - compiles&runs your java code.
- **CS166_Project/java/lib/pg73jdbc3.jar** - The Postgres JDBC driver, which is necessary for your Java code.

4. Execute **CS166_Project/sql/scripts/create_db.sh** to load your database
5. Execute **CS166_Project/java/scripts/compile.sh** to compile and run your Java client.

1.1 Java console application

If this is the first time you work with Java there is no need to be worried. You are provided with a template client written in Java. You are expected to extend this basic client and add the functionality, required for a complete system. An Introduction to Java/JDBC can also be found in your Textbook (Sections 6.2 – 6.3 of Database Management Systems (Third edition)).

In this phase we basically want to create an interactive console for non-sql users (i.e. users of our Online Network). You should therefore pay special attention to making the interface as intuitive as possible for regular user.

In order to run the template Java program you should first start the Postgres database as it was described in the Postgres Manual (or the quick start guide). Then you should execute shell script *CS166_Project/java/scripts/compile.sh*

The script will compile and run Java source for the file *CS166_Project/java/src/ProfNetwork.java*. If you will add other Java source files to your project, please modify the script accordingly.

2 Functionality of the Online Messenger

Below you will find the basic requirements for the functionality of the system. On top of them you may implement as many additional features as you want. Throughout the project you should keep in mind that our users are not SQL-aware. You should therefore make the user interface as intuitive as possible.

Below we provide the basic operations you must implement in your system.

1. Users
   - New User Registration: when a new user comes to the system, he/she can setup a new account through your interface, by providing necessary information.
   - User Login/Logout: user can use his/her username and password to login into the system. Once he/she logs in, a session will be maintained for him/her until logout or exit from system (an example has been provided in the Java source code).
   - Change Password: user can change the password.
   - search People: Search people by name
   - Sends Connection Request: Remember you can only send request within 3 levels of connection. For a new user who does not have any connections can make upto 5 new connections without the 3 level rule applying.
   - Accepts or Rejects Connection Request Connection Request

- View Friends and then go to a friends profile. From here they can again view friends and goto another friends profile. Everytime you view a profile you can either send a connection request(depending on level of connection) or send a message to them.
- Send a message to anyone on network.
- View Messages and then option to delete messages

3 Extra Credit

1. Good User Interface.

   A good user interface will bring more users to your application, and also more points in this phase. A user interface is good if it is:

   - Easy for users to explore features;
   - Robust in exceptional situations, like unexpected inputs;
   - Graphic interface supporting all required features.

2. Triggers and Stored Procedures.

   Instead of processing the workflow step by step, triggers and stored procedures can be used to handle a sequence of SQL command and to execute these sequences on some table events (inserts, updates, etc).

   To submit your triggers and stored procedures with your project, please include them in the following location *project/sql/src/triggers.sql*

3. Performance Tuning

   The performance can be improved if index are used properly. **You will receive points only if the index will actually be used in your queries**. You should defend why you have chosen to use an index at some particular table. For your submission, you should put index declarations in *project/sql/src/create˙indexes.sql*

4. Any Other Fancy Stuff...

   Please feel free to show any of your ideas to improve your project! The only thing required will be clear documentation!

4 Submission Guideline

1. Project Report

   You should provide a high level description (1-2 pages) of your implementation. You should describe which part each of you was working on and any problems\findings, you found along your way. In this document you should also include whatever is asked in the below individual descriptions.

   You should submit a single zip archive via iLearn at least 1 hour before your presentation and notify TA about that via email. Your report should have both the project partners names, sids, group id. Please put this information in all your source code as well. Put them as comments on the top of the page. The available slots for the presentation will be posted through iLearn later. Presentations will start from Jun 8 onwards.

2. Files

   The following files should be submitted:

   (a) Create Tables, Bulkload Data Scripts.

   If you have any schema or tables modifications you should include your changes into the files and leave necessary comments for them. Modify the following files: *CS166_Project/sql/src/c* *CS166_Project/sql/src/load_data.sql*

   (b) System Implementation.

   Submit your source file(s). You should make sure that your code can compiles and run successfully. **Any special requirement for compiling and running should be stated clearly in your project report, or a README file which comes with your source code**. Please put all your source code within the *CS166_Project/java/src* directory.

   (c) Other scripts, like triggers, stored procedures, and indexes.

   You should provide descriptions for these features and include all your scripts within the *CS166_Project/sql/src* directory.