CSCI 3357: Database System Implementation

Homework Assignment 0

Due Wednesday August 28

Every assignment this semester will involve running the SimpleDB database system. It is essential that you get SimpleDB installed and become comfortable with its use as soon as possible. Please do the following tasks by next class. Let me know if you run into difficulties. You need not turn anything in.

1. Ensure that the Eclipse IDE is on Your Computer

If you do not have the Eclipse IDE on your computer, go to *eclipse.org* to download and install Eclipse for Java. If you have a different IDE then that is fine, but you are responsible for getting the SimpleDB code to work with it.

2. Download the SimpleDB Source Code

Download the file *SimpleDB_3.3.zip* from the Documents module of the course website and extract its contents on your computer. The extracted folder should have subfolders named *simpledb* and *simpleclient*. (It will also have a folder named *derbyclient*, but that can be ignored.)

3. Create a Project for the SimpleDB Engine

A. In Eclipse, create a new Java project named *SimpleDBEngine*.

- You will need to specify the location. I recommend using the default location, which tells Eclipse to create a folder named *SimpleDBEngine* within its workspace.
- You should specify "Create separate folders for sources and class files". Eclipse will create folders named *src* and *bin* within the *SimpleDBEngine* folder.
- Click on the Finish button.

B. Use the operating system to copy the entire downloaded *simpledb* folder to the folder *SimpleDBEngine/src* in your Eclipse workspace.

- When you are done, the *src* folder should have one child folder, namely *simpledb*. The *simpledb* folder should have the child folders *buffer*, *file*, etc.

C. In Eclipse, execute Project/refresh (F5) to compile all the source files.

- The *bin* folder will now contain a *class* file for each source file.

4. Create a Project for the Client Code

A. In Eclipse, create a new Java project named *SimpleDBClients*.

- Configure the project the same as the first two bullet points of part A above.
- Instead of clicking "Finish", click "Next" to get to the Java Settings window.
- Click on the Projects tab. Then click Add, and click the box for the *SimpleDBEngine* project. (Doing so adds the SimpleDB source code to the project's class path. Otherwise, the client code will not be able to resolve references to the SimpleDB classes.)
- Now you can click the Finish button.

B. Use the operating system to copy the contents of the downloaded *simpleclient* folder into the *SimpleDBClients/src* folder in your Eclipse workspace. Do not copy the enclosing *simpleclients* folder. The src folder should have four items: two folders and two files.

C. In Eclipse, refresh the project as in the previous step C.

5. Run the Embedded Client Programs

A. Create the student database in embedded mode.

- Look at the programs in the *embedded* folder of the *SimpleDBClients* project.
- Run *CreateStudentDB* . It will create a database named *studentdb* having the same tables as in Figure 1.1 of the text.
- Refresh the project. You should see a folder for the studentdb database in the project window. Feel free to examine its contents.
- Run *StudentMajor*. It should open a console window and display 9 records showing the names of the students and their majors.
- Run the *ChangeMajor* client, which will change the *MajorId* value of Amy's record in the STUDENT table. Re-run the *StudentMajor* program to verify this.
- Run *CreateStudentDB* again. Technically, you shouldn't do this, but do it anyway just for fun. Re-run *StudentMajor*. What happened to the database?

B. Delete the database and re-create it.

- From the Eclipse client project, delete the folder containing the files for the *studentdb* database. You just destroyed the database!
- From Eclipse, re-run *CreateStudentDB*. You just re-created the database. Rerun *StudentMajors* to verify that it is back to normal.
- If you want to see the database folder in the Eclipse project list, refresh the project.

6. Run the SimpleDB Engine as a Server

A. Create a run configuration for the server program.

- Go to “Run Configurations” in the Eclipse Run menu. Add a new configuration to your *SimpleDBEngine* project, called “SimpleDB Server”. In the field for the main class, enter “simpledb.server.StartServer”.
- By default, the server will use a database named “studentdb”. This is what I recommend. But if you want to use a differently-named database, use the Arguments tab in the configuration to enter the database name.

B. Run the *SimpleDB Server* configuration you just created. A console window should appear indicating that the SimpleDB server is running.

C. The server creates its database in a different location from the embedded client. The folder for this database lives in the *SimpleDBEngine* project. Refresh the project to see it in the Eclipse project window.

7. Run the Server-based Client Programs

Look at the programs in the network folder of the *SimpleDBClients* project.

- While the server is running, run the *CreateStudentDB* and *StudentMajor* clients. They should print the same output as in step 6.
- Go to the console window for the server, and shut it down (by clicking on the red square near the top of the console window). Run *StudentMajor* again. You should get an error message.
- Rerun the server, then run *StudentMajor*. It should now work.
- Run the *ChangeMajor* network client. Now you have two slightly different student databases. In the embedded database, Amy is a math major. In the network database, Amy is a drama major.

8. Run the *SimpleIJ* Client Demo

- Run the program *SimpleIJ*, which is in the “default package” folder for the *SimpleDBClients* project.

- The first thing it will ask for is a connection. Enter the following string, which will establish a connection to the embedded database.

```
jdbc:simpledb:studentdb
```

- The client will now repeatedly ask you to enter SQL queries, one per line. Type the following query, which should print the name and majorid of all students.

```
select sname, majorid from student
```

Note that Amy has majorid = 30 in this database.

- Type “exit” to terminate the program.

- Assuming that the server is still running, re-run *SimpleIJ*. This time, enter the following network connection string

  ```
  jdbc:simpledb://localhost
  ```

  This will connect you to the network database (If your server is not running, it will show an error message). Type the following query, and note that Amy has majorid = 20 in this database.

  ```
  select sname, majorid from student
  ```

- If you know SQL, try entering some other queries into SimpleIJ. You can figure out the names of the tables and their fields by looking at Figure 1.1 of the text. Section 1.5 of the text describes the subset of SQL supported by SimpleDB. What happens when you try to execute an SQL statement that SimpleDB doesn't support?

- Type “exit” to terminate the program.

- Shut down the server (by clicking on the red square near the top of the console window).

9. Finish Up

A. Write a 10 page paper describing what you learned.

- Just kidding. Don’t do that.

- On the other hand, if you don’t understand what I asked you to do, then try again. If you still are perplexed, then ask me.

B. Configuring a system is often deceptively difficult.

- The point of this assignment was to ensure that your system is properly configured, and to get you totally comfortable using it.

- Later homework assignments will ask you to change the code for the system. The time you spend now getting comfortable with the system will make it possible for you to debug your code confidently and effectively.