

CSCI 3357: Database System Implementation
Homework Assignment 0
Due Thursday August 31

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. Do not turn anything in.

The following tasks supersede the instructions given in Chapter 1 of the text.

1. Ensuring 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. From the installer, select “Eclipse IDE for Java Developers”. I don’t mind if you choose to use a different IDE, but you are responsible for getting the SimpleDB code to work with it.

2. Downloading the SimpleDB Source Code

Download the file *SimpleDB_3.4.zip* from the course website and extract its contents on your computer. The extracted folder should have subfolders named *simplifiedb* and *simpleclient*. (It will also have a folder named *derbyclient*, but that can be ignored.)

3. Creating a Project for the SimpleDB Engine

- A. In Eclipse, create a new Java project named *SimpleDBEngine*.
 - Go to the File menu and select “New>Java Project”.
 - You will need to specify the location. I recommend using the default location, which tells Eclipse to create a folder named *SimpleDBEngine* in 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.
 - At the bottom, ensure that “Create module-info.java file” is unchecked. We will not be using modules.
 - Click on the Finish button.
- B. Using the Finder (for Macintosh) or File Explorer (for Windows), copy the entire downloaded *simplifiedb* folder to the folder *SimpleDBEngine/src* in your Eclipse workspace.
 - When you are done, the *src* folder should have one child folder, named *simplifiedb*. The *simplifiedb* folder should have the child folders *buffer*, *file*, etc.

C. In Eclipse, execute the File>Refresh menu option (or press F5) to compile all the source files.

- In your workspace, the *bin* folder for your project will now contain a *class* file for each source file.

4. Creating 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, and select "Classpath". Then click Add, **click the box** for the *SimpleDBEngine* project, and click OK. This adds the SimpleDB source code to the project's class path. If you don't do this, the client code will not be able to resolve references to the SimpleDB classes.
- Now you can click the Finish button.

B. Use the Finder (or File Explorer, as appropriate) 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. Running the Embedded Client Programs

A. Create the student database in embedded mode, as follows:

- From Eclipse, 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. The actions taken by the engine will be displayed in the console window.
- Using the Finder (or File Explorer), go to your Eclipse workspace and examine the *SimpleDBClients* folder. It should now contain a folder named *studentdb*. This folder is where SimpleDB stores the contents of the database you just created. (Take a look!) Any time that you make a change to the database, that change will be reflected in the contents of this folder.
- Go back to Eclipse and refresh the *SimpleDBClients* project. The project window will now display the same *studentdb* folder that you saw in the previous step.
- 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. This causes duplicate records to be inserted into the database. To verify this, re-run *StudentMajor* and note that there are two records for each student. Interestingly, the records for Amy have different majors; do you see why?

- B. Delete the database and re-create it, as follows:
- From the Eclipse client project, delete the folder containing the files for the *studentdb* database. Congratulations! You just destroyed the database.
 - Another way to destroy the database is to use the Finder (or File Explorer) to delete the *studentdb* folder from the Eclipse workspace. Then refresh the project, so that Eclipse knows that this deletion occurred.
 - Now re-create the database, by simply re-running *CreateStudentDB*. Refresh the project if you want Eclipse to display the folder for the new database.

6. Running the SimpleDB Engine as a Server

- A. Create a run configuration for the server program, as follows:
- Go to “Run Configurations” in the Eclipse Run menu. Select “Java Application”, and then press the “New Configuration” button. In the resulting window, type “SimpleDB Server” as the configuration name, “SimpleDBEngine” as the project, and “simpledb.server.StartServer” as the main class.
 - By default, the server will use the database named “studentdb”. This is what I recommend. But if you want to use a differently-named database, type your desired database name into the *Arguments* tab of the window.

B. Run the *SimpleDB Server* configuration you just created. A console window should appear indicating that the SimpleDB server created a new database and is ready.

C. In Step 5A you saw that embedded databases created from the *SimpleDBClient* project live in the *SimpleDBClient* folder of the Eclipse workspace. Similarly, the server-based database you just created lives in the *SimpleDBEngine* folder of the workspace. If you refresh that project, the database folder will appear in the project window.

7. Running the Server-based Client Programs

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

- While the server is running, first run the *CreateStudentDB* client and then *StudentMajor*. Although their output should be the same as in step 5A, it will be displayed in separate console windows — one window for the server, and one for the client. By default, Eclipse displays these windows on top of each other. To toggle between them, click the “Display selected console window” icon (which is the 10th icon from the left at the top right of the console window).
- Shut down the server by going to its console window and clicking on the red square near its top. Run *StudentMajor* again. You should get an error message.
- Rerun the server, then run *StudentMajor*. It should now work.
- Run the *ChangeMajor* network client. As a result, the two databases in your Eclipse workspace are now slightly different. In the embedded database, Amy is a math major. In the network database, Amy is a drama major.

8. Running the *SimpleIJ* Client Demo

- Run the program *SimpleIJ*, which is in the “default package” folder for the *SimpleDBClients* project.
- Its console window displays “Connect>”. This message asks you to enter a connection string. Enter the following string, which will establish a connection to the embedded database:

```
jdbc:simpledb:studentdb
```

- The client now displays “SQL>”, which asks you to enter an SQL query. Type the following query to print the name and majorid of all students.

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

Pressing the Return key will cause the program to execute the query and print its results. You should note that the record for Amy has majorid = 20.

- Type “exit” to terminate the program.
- Assuming that the server is still running, re-run *SimpleIJ*. This time, enter the following network connection string, which will connect you to the network database. (If your server is not running, it will show an error message).

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

- Type the same query as before. Note that in this database Amy has majorid = 30.
- If you know SQL, enter some other queries into SimpleIJ. (Keep in mind that the program expects the entire query to be on a single line.) 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 execute an SQL statement that SimpleDB doesn't support?
- Type “exit” to terminate the program. Then shut down the server (by clicking on the red square near the top of its console window).

9. Finishing 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 is 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 future code confidently and effectively.