CS 210 Spring 2012

Project 4 (Binary Files) Instructions

1. Project 4 will use the information on table definitions created in Project 3 and will develop mechanisms to store and retrieve data from binary files.
2. There will be one binary file per table. I suggest you use the directory structure you created for Project 3 in order to store the files. If you haven't created a directory structure, this is a great time to do it, otherwise the user (who could that be?) will have a large number of files and this will be difficult to deal with.
3. A type structure should be developed to manipulate the data being input. Upon reading the data from the insert statement, the data should be converted to the appropriate type. All manipulation of data should be done in the appropriate type (and not, of course, as strings). As this project progresses, more type-specific activity will be required.
4. For this project, you must create JUnit test methods as appropriate for this project.
5. Javadocs will also be required unless you have received a Javadoc waiver.
6. The following commands should be implemented for this project:
   a) Insert. Inserts a row of data into the table identified in the insert statement. Note that the insert statement must apply to an existing table, and the number, order and type of data to be inserted must match that of the fields as determined by the table's define table statement. If any of these criteria are not met, the data should not be inserted and the user notified.
   b) Print *tablename:* This prints out all data in the table. The first row of the printout should be a list of the columns for the table in question, in order of their creation. Each subsequent row should list all data from a record stored in the table, and all non-deleted rows should be printed out. I recommend that you separate each column with a tab. Otherwise, structural organization is simply not an issue. The columns do not need to be aligned on the page, and if the data from a single record is too long to print on a single row of the printout, it is acceptable for the output from one record to wrap to two or more rows. Please note that this command is primarily used for debugging and testing.
7. Note that “Drop” (implemented in project 2) should be expanded to delete the binary data file when called.
8. Submissions should have all current code loaded into subversion. JUnit tests are a mandatory component of this project, and in fact are almost essential to develop and debug a binary file reader and writer. Submissions are made by emailing a working jar file labeled with your name to cs210.submissions@gmail.com, and should include all required class files, all source code for the application including source code for all JUnit tests, and all javadocs as required.