# **CSCI X370: Database Management**

**Spring 2017**

## **Project 1: Relational Algebra Operators**

Due: Jan 27 (11:59 pm)

Implement the following relational algebra operators:

Select, Project, Union, Minus and Join.

Some of the operators are partially implemented in Table.java (at ELC) and what you need to do is to fully implement those operators. Tuples are stored in comparable lists (ArrayList). You can see some test cases in the main function of Table.java; however, since functions are partially implemented, it will not give the proper output. So, you should be able to see the desired output after the completion of implementation. Download the starter source code from the ELC and compile it. Upon making sure that it is working properly, you can start implementing. If you compile and run MovieDB.java, you will see the tables retrieved based on the queries, which uses partially implemented operators.

Every operator has a method to be implemented and those methods have other methods to be called and to be implemented. So, it may require implementing more than one method to complete an operator. While implementing, also pay attention to the domain (data types) since this could be important to be able to implement some of the operators. The methods may have some parts to be deleted or uncommented right after the implementation; so, please read comments within the code.

Union operator will simply merge the two tables (except the duplicate tuples in intersection). Minus operator will output the tuples that do not exist in the second table. Project operator will retrieve the elements based on the column positions. For Select, data type conversion is important (in String2Type.java). Furthermore, you need to make comparisons per the conditions given in the query (postfix).

Use a (Java) TreeMap for an index.

Your program must be thoroughly documented (generate javadoc). Use the @author tag for each class and method. Each method should have a single author. The coding workload should be split roughly among 4 (or 5) team-members. We will check this by examining the @author tags and peer evaluations. Please make sure that the output of your program is easy to understand. Provide a flag for turning on/off your tracing/debugging messages in your program's output – if necessary.

**Programming language:** Java is required for the project.

**What to submit:** Please submit

* all source code
* all the javadoc files
* a readme file

The readme file should contain: your names, how to compile and run your code and other specifications you want to make. Please pack all your files in a zip package with the file name: "project1" + last names of group members. For example: project1\_chen\_kim\_wong\_allen.zip

**How to Submit:**

Submit your ".zip" file using ELC. Only team leaders need to make a submission. **Every student needs to submit a peer-evaluation form within 24 hours of the project submission deadline**.

Do not place your solution on a public web site. Submit your own work and follow the course misconduct policy.