Summer 2011 CSCC43
Introduction to Databases

Assignment 3

Due Date: August 8th, 11:59p.m. (20%)

Teams:

- This assignment is to be done in teams of 2 (3 is unacceptable).

Electronic submission:

- Submit your assignment through portal. There is a link in the assignments folder for submission.

DBMS:

- This assignment must be done on PostgreSQL.

Late Assignment:

- Since this assignment is on the last of classes -- no late submissions will be accepted.

Part I: Stored Procedures & CLI (10%)

This part of the assignment builds on assignment 2.

Stored Procedures
You are going to write stored procedures, to run in PostgreSQL. Stored procedures for inserting into each of the 7 tables are required – one per table. You are also required to write stored procedure for the following queries: 10, 11 and 12.

Embedded SQL
You are required to write embedded SQL (i.e. SQL inside Java code) for the following queries: 7, 8 and 9.

Java Program
You are required to write a Java command line program (name the main class **DBDriver.java**) that will enable a user to interact with your database instance. The program should display a menu of 14 options – as follows:

```
Select one of the following options
   1. Insert in Movie table
   2. Insert in Director table
   3. Insert in Actor table
   4. ……
   13.
   14. Exit
```

Notes:

- Once the user select an item, ask for the values and do the corresponding action in the database. Do not exit unless the user inputs 14.
- You will need to check that the user has provided valid input for the inserts.
- All exceptions must be caught by your program and display a meaningful error message.
- When connecting to database, you will need to load username and password from key=value pair text file (name and location of file shown in Required Directory Structure at end of this document). You can use java.util.Properties class to read this text file.

Part II: XPath (10%)

For this part we will examine roughly 275MB of XML data describing a selection of clinical trials conducted in the USA during 2010.

The data are all publicly available at www.clinicaltrials.gov, and the search engine there may help you debug your queries. See also http://clinicaltrials.gov/ct2/info/linking. For your convenience, all the data you need are available at the portal as a compressed tarball (24MB download). To extract the files run from Cygwin (or on other platforms with recent builds of 'tar' and 'xz'):

```
$ tar xaf clinical-trials-2010.tar.xz
```

Each clinical trial is represented by a single .xml file (about 21,000 in all); there is also a 'collection.xml' which lists all the trials if you want to access them as a group using XQuery's collection function. The DTD describing the trials' XML layout is available at http://clinicaltrials.gov/ct2/html/images/info/public.dtd

During testing, you will want to use small subsets of the data for faster response time. To help with this, we have provided a python script (available on portal) which will take a list of file names from stdin and print a random sample of them to stdout in XML format similar to the included collections.xml. For example, to create a random selection of 10 files:

```
$ ls clinical-trials-2010/ | python random-subset.py 10
<collection>
  <doc href='NCT01043861.xml'/>
  <doc href='NCT01051336.xml'/>
  <doc href='NCT01096212.xml'/>
  <doc href='NCT01103557.xml'/>
```

```
  <doc href='NCT01113619.xml'/>
  <doc href='NCT01118260.xml'/>
  <doc href='NCT01121822.xml'/>
  <doc href='NCT01203592.xml'/>
  <doc href='NCT01208727.xml'/>
  <doc href='NCT01212861.xml'/>
</collection>
```

You will most likely want to work with sets of ~1000 files to get a good sample without taking too long for queries to run. For XML query processing, we will use the Java version of SaxonHE, available at http://saxon.sourceforge.net/. You will need access to Java JRE version 1.5 or later as well. Java is available at http://www.java.com/getjava/index.jsp. To run Saxon, you need to place your XPath/XQuery code in a file (normally with extension .xq) and run the following command:

```
$ java -Xmx500M -cp saxon9he.jar net.sf.saxon.Query query.xq
```

Please be aware that Saxon needs roughly 3MB memory for every 1MB of XML data, because they want you to pay for two basic features (projection and streaming) which would prevent the problem; if Java crashes with OutOfMemory errors, try adjusting the -Xmx flag to give the JVM access to more memory.

Your task is to write XPath expressions which give the following (*note that these do not necessarily apply to the clinical trials dataset*):

1) All country nodes residing anywhere in the document
2) The first address node of each order
3) The zip code of each address whose city is London
4) All image nodes having a width or height attribute
5) The nct_id of each clinical_study spanning at least five locations
6) The name and all aliases of each prisoner in a prison whose name attribute equals "Alcatraz"
7) Output the nct_id, last_name of the overall_official, and first listed condition for each study with a location in Toronto.
8) Output the nct_id and each Toronto-based address for any clinical_study spanning 5 or more locations where one or more of those locations is in a country besides the "United States" or "Canada" (note: some such studies might not have a location in Toronto).

Required Directory Structure

`Note: directory name bolded. Comments in italic.`

`<your-team-name>`

`README` `--` *names, ids, and UTORIDS of 2 students who worked on assignment*
`Makefile`

```
Config.props –- key=value text file containing database URL, and
          -- username/password used to connect to database.
sql
          create.sql      -- ddl file
          inserts.sql     -- your inserts file from A2
          drop.sql        -- deleting tables
          sprocs.sql      -- stored procedures file
libs
          -- any jar files you are using and required to run
          -- your submission go here.
classes
          -- this is where the Makefile should put the .class
src
          -- All your part I java files go here – including
          -- DBDriver
          DBDriver.java
xpath
          -- all files related to part II here
```

What to submit?

Zip the parent directory <your-team-name> and submit the zipped folder through [*http://portal.utoronto.ca/*](http://portal.utoronto.ca/)