COSC 416A – Lab #11

Berkeley DB

Slides can be accessed here: http://sdrv.ms/X9JOM0

BerkeleyDB is installed on **gpu1.ddl.ok.ubc.ca** at "/srv/berkeleydb".

It comes with a SQLite wrapper with command shell that can be accessed via **/srv/berkeleydb/bin/ dbsql <db-name>**. Where <db-name> can be your student id.

NOTE: Your database will be created in the directory you call the shell from and some files will be generated there. I suggest creating a new directory in your home directory (/home/<your student #>) for all of your BerkeleyDB files.

Once in to the **dbsql** shell you can type ".help" for information on SQLite commands.

.databases Will show current databases. Verify the location of your new database <db-name>.

Part 1 - DBSQL shell (10 marks):

For this part you will need to download and extract this zip file:

https://docs.google.com/file/d/0B1eE7wLtgG9fRnFuTDM0NUZ6aWM/edit?usp=sharing

to your new berkeleydb directory.

From this directory start up the dbsql shell with the following command with your student # where it says <std#>:

```
/srv/berkeleydb/bin/dbsql <std#>
```

1. **(1 mark)**
   From the shell, create 3 tables:

   players(id, fname, lname, bdate, sex, pic)

   games(id, name, publisher, releasedate, rating)

   playerGames(pid, gid, score)

2. **(2 marks)**
   Import the data from the text files into the tables. Hint: You will have to change the input/ output mode with ".mode". Use ".help" in the shell for more information.

3. **(2 marks)** Find the number of players for each game. Output should be game id, game name, playerCount, ordered by playerCount descending. Note that no player has two scores for the same game, and every game has been played at least once.
4. **(3 marks)** Find the player with the top score for each game. Output the player's id, full name and the game id and name.
5. **(2 marks)** Find the players with 2 high scores in any games. There should be 6. Output the player's id, fullname, and # of high scores.

Part 2 - JDBC SQLite Java (15 marks):

For this part you will need to complete the follow java file, and compile/ run on gpu1.ddl.ok.ubc.ca.

https://docs.google.com/file/d/0B1eE7wLtgG9faWhOZldmQTZ1WTA/edit?usp=sharing

(Updated April 3rd, 3:45pm. Second progress bar should work correctly now. This does not affect your answer.)

Suppose you had backed up your MySQL database to a text file of tab separated values like the .txt files: games.txt, players.txt, player_games.txt.

We are now migrated to a BerkeleyDB database and need to write a program to copy the contents of the text file **player_games.txt** to a new table **playerGames (pid, gid, score)** and run some queries on it.

Once we have loaded the data into the table, we want to run a query to find the highest score for each player and the game id that they achieved the score on. With the result set from this query we are going to output the contents to a new table **topScores (pid, gid, score)**.

Finally we will output the contents of **topScores** to verify we have copied the data correctly.

Use **Lab11.java** as a template to start your program. You will need to modify or insert code wherever there is a //TODO comment.

To (re)compile with the SQLite library use the following command:

```
javac –classpath /srv/berkeleydb/jar/sqlite.jar Lab11.java
```

To run your program with java use the following command:

```
java –classpath .:/srv/berkeleydb/jar/sqlite.jar –Djava.library.path=/
srv/berkeleydb/lib \Lab11
```

NOTE: Since this program creates tables each time it is ran you must delete your old database between running it. This can be accomplished with "rm <db-name>" in your Berkeleydb working directory.