CSC 370 — Database Systems
Spring 2014
Assignment No. 4

Note 1 **This assignment is to be done individually**

Note 2 Working with other people is prohibited.

- Due date: Tuesday, Nov 13, 2014, at the beginning of the class.
- This assignment is worth 1% of your total course mark.
- Summit electronically your UDF and the source code of the program. Submit a paper copy at the beginning of the class.

Objectives

After completing this assignment, you will have experience:

- Create User Defined Functions.
- Create programs that interact with the DBMS.

Your task, should you choose to accept it

1. Write a User Defined Function that takes one parameter: a string. This string is a *pid*. Call the UDF anything you want (e.g. *myFunc*) but prefix it with your initials (e.g. *dmg_myFunc*). The result of this UDF is a projection of the *id*, *year*, *rank* and *votes* of the **movies** directed by the given pid (see below), ordered by year. Make sure you do a left or full join between tables (so you show movies even if they do not have a rank).
2. Our customer wants a report that, given a *pid*, finds all the movies that that person has directed (*productions* with *attr* is null), ordered by year, and then by title (if two have the same year). The report should create an HTML file that contains a table that looks like this (borders are optional) when the pid is *Nolan, Christopher (I)*.

Director: Nolan, Christopher (I)

id	**year**	**rank**	**votes**
Doodlebug (1997)	1997	7.2	7006
Following (1998)	1998	7.6	50506
Memento (2000)	2000	8.5	629051
Insomnia (2002)	2002	7.2	175822
Batman Begins (2005)	2005	8.3	686987
The Exec (2006) SUSPENDED	2006		
The Prestige (2006)	2006	8.5	582740
The Dark Knight (2008)	2008	9	1202013
Inception (2010)	2010	8.8	989812
The Dark Knight Rises (2012)	2012	8.6	798512
Interstellar (2014)	2014		

Total: 11 movies

You can write your program in any language you choose provided it can be executed in linux.csc.uvic.ca.

These are the requirements:

(a) Your program takes as its first parameter the pid of the director. **Do not ask for it in an interactive manner**. For example, if your program is in perl, and it is called *program.pl*. I want to run your program as (note the quotes around the parameter):

(b) Your program should **interactively** ask for username and password from the user (in that order).

```
perl ./program.pl 'Nolan, Christopher (I)'
```

(c) Your program should output the HTML to standard output. **Do not write the data to a file**. So I can run your program as:

```
perl ./program.pl 'Nolan, Christopher (I)' > output.html
```

(d) Your program can only use the UDF that you defined in part 1 (see above). **You cannot write any queries that reference directly any of the IMDB tables.**

(e) The listing should contain the movies directed by such *pid*, ordered by year.

(f) It should generate valid HTML.

(g) **Do not read all the tuples at once.** It is bad practice to read all tuples at once (unless you know in advance how many you will read). You should read one tuple at a time because you do not know how many tuples you might read.

(h) Important: **Make sure your program does not suffer from SQL injections**. See Wikipedia for details.

What to submit:

- Submit, via connex, 2 files, in one zip file (or tar file).

  1. File Number 1, called *README.TXT* should describe how to compile and run your program. Remember, it should compile and run in linux.csc.uvic.ca.
  2. File Number 2 should be your program. Call it whatever you want.

- Submit a printout of your program before the beginning of class.