CSC 370 — Database Systems
Fall 2018
Assignment No. 6

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

Note 2 Working with other people is prohibited.

- Due date: Thursday, Aug 2, 2018 (Midnight)
- This assignment is worth $\frac{4}{3}$% of your total course mark.
- Summit electronically your solutions in a single text file.

Objectives

After completing this assignment, you will have experience:

- Create User Defined Functions.
- Creating a stored procedure
- Creating a trigger.

Your task, should you choose to accept it

Part 1

1. For this part you will learn about creating UDFs. These are very similar to Stored Procedures, but they return a result. See `https://www.postgresql.org/docs/9.4/static/xfunc-sql.html`.

   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 uvic-id (e.g. *dmg_myFunc*). The result of this UDF is a projection of the *id*, *year*, *rank* and *votes* of the **movies** (productions with attr equal NULL) 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).

   You can test your UDF using psql as follows:

```
SELECT * FROM dmg_myFunc('Nolan, Christopher (I)');

              id              | year | rank |  votes
------------------------------+------+------+---------
 Doodlebug (1997)             | 1997 |  7.1 |   11415
 Following (1998)             | 1998 |  7.6 |   66121
 Memento (2000)               | 2000 |  8.5 |  849849
 Insomnia (2002)              | 2002 |  7.2 |  219782
 Batman Begins (2005)         | 2005 |  8.3 |  986936
 The Exec (2006) {{SUSPENDED}} | 2006 |      |
 The Prestige (2006)          | 2006 |  8.5 |  849119
 The Dark Knight (2008)         | 2008 |    9 | 1685825
 Inception (2010)               | 2010 |  8.8 | 1476746
 The Dark Knight Rises (2012)   | 2012 |  8.5 | 1151061
 Interstellar (2014)            | 2014 |  8.6 |  937348
 Quay (2015)                    | 2015 |    8 |     357
 Dunkirk (2017)                 | 2017 |      |
(13 rows)
```

Part 2

For this assignment work in you own database. The name of your database is the name of you UVic netlink id.

- Create a table $Parts$ with the following schema:

  ```
  parts(pid integer, pname varchar(40), color varchar(20));
  ```

  Add some data to the table.

- Create a table called *partshistory* with exactly the same schema as the original table, plus three columns called:

  1. *operation* with type CHAR,
  2. *opwhen*, with type TIMESTAMP,
  3. *opuser*, with type CHAR(20).

- Create a TRIGGER that, whenever a record is added, updated or deleted from the $parts$ table, it makes a copy of that record into the table $parthistory$, and sets its *Operation* field to “U”, “D”, “I” (Updated, Deleted, Inserted, respectively), sets the field *When* to the current time, and the user name to the current user.

Use the Postgresql Programmer’s Guide. Make sure you use the one corresponding to the version of the DBMS we are using (9.3). The postgres documentation contains exactly the information you need to complete this assignment.

What to submit:

- Submit, via connex, one text file containing your solutions and a sample of the output of your program (directors).