#########################################

## Jakob Roberts - v00484900

## CSC370

## Assignment 5

#########################################

####################################################################################

1.

1. SELECT from MovieStar

SELECT from MovieExec

1. SELECT from MovieExec

SELECT from Movies

SELECT from StarsIn

1. SELECT from Movies

SELECT from Sutdio

INSERT to Studio

1. SELECT from StarsIn

DELETE in StarsIn

1. SELECT from Studio

SELECT from MovieExec

UPDATE to MovieExec

1. SELECT from MovieStar

INSERT to MovieStar

UPDATE to MovieStar

1. SELECT from Studio

SELECT from MovieExec

UPDATE to Studio

UPDATE to MovieExec

INSERT to Studio

INSERT to MovieExec

####################################################################################

2.

|  |  |  |
| --- | --- | --- |
| **Step** | **By** | **Action** |
| 1 | A | GRANT p TO B WITH GRANT OPTION |
| 2 | A | GRANT p TO C |
| 3 | B | GRANT p TO D WITH GRANT OPTION |
| 4 | D | GRANT p TO C, E WITH GRANT OPTION |
| 5 | B | REVOKE p FROM D CASCADE |
| 6 | A | REVOKE p FROM C CASCADE |

Show the grant diagrams after steps (4) through (6) of the sequence of actions listed in Fig. 10.6. Assume A is the owner of the relation to which privilege p refers.

####################################################################################

3.

One way to check is by doing:

SELECT prosrc FROM pg\_proc WHERE proname ~’jmr’;

OR, the actual code to make it was:

CREATE OR REPLACE FUNCTION jmr(character varying)

RETURNS TABLE(id character varying,year integer,rank double precision,votes integer)

AS $$

SELECT id,year,rank,votes

FROM directors NATURAL JOIN productions NATURAL LEFT JOIN ratings

WHERE attr IS NULL AND pid =$1

ORDER BY year,title desc;

$$ LANGUAGE sql;

####################################################################################

4.

import getpass

import psycopg2

import sys

def main():

output = sys.stdout

sys.stdout = open('/dev/tty','w')

if(len(sys.argv) ==2):

input = sys.argv[1]

command = "SELECT \* FROM JMR('" + input + "');"

print "Connecting to: studentdb.csc.uvic.ca @ DB: imdb"

print "--------------Please provide input--------------"

username = raw\_input('Username: ')

pswrd = getpass.getpass('Password: ')

try:

conn = psycopg2.connect(host="studentdb.csc.uvic.ca",dbname="imdb",user=username,password=pswrd)

cur = conn.cursor('cursor\_of\_doom')

cur.execute(command)

#rows = cur.fetchall()

sys.stdout = output

print "<!DOCTYPE html>\n<html>\n<head>\n<style>\ntable, th, td {\n\tborder: 2px solid black;\n\tborder-collapse: collapse;\n}\nth, td {\n\tpadding: 3px;\n}\n</style>\n</head>\n<body>\n\n<table style=\"width:100\"%>\n <tr>\n <td><strong>id</strong></td>\n <td align=\"right\"><strong>year</strong></td>\n <td align=\"right\"><strong>rank</strong></td>\n <td align=\"right\"><strong>votes</strong></td>\n </tr>"

print "Director:",input,"<br><br>\n"

count=0

row = cur.fetchone()

while row is not None:

count+=1

print "<tr>"

for i in range(0,4):

if(row[i]==None):

print " <td>"+""+"</td>"

elif(i==0):

print " <td>",row[i],"</td>"

else:

print " <td align=\"right\">",row[i],"</td>"

print "</tr>"

row = cur.fetchone()

print "</table>\n<br>Total:",count,"movies"

print "\n</body>\n</html>"

except:

print "Database Error!!"

else:

print "Correct usage: python a5python.py <pid>"

conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Director: Nolan, Christopher (I)

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **year** | **rank** | **votes** |
| Doodlebug (1997) | 1997 | 7.2 | 8815 |
| Following (1998) | 1998 | 7.6 | 57298 |
| Memento (2000) | 2000 | 8.5 | 717017 |
| Insomnia (2002) | 2002 | 7.2 | 194816 |
| Batman Begins (2005) | 2005 | 8.3 | 800210 |
| The Prestige (2006) | 2006 | 8.5 | 691835 |
| The Exec (2006) {{SUSPENDED}} | 2006 |  |  |
| The Dark Knight (2008) | 2008 | 9.0 | 1394726 |
| Inception (2010) | 2010 | 8.8 | 1193532 |
| The Dark Knight Rises (2012) | 2012 | 8.5 | 939715 |
| Interstellar (2014) | 2014 | 8.8 | 544268 |

Total: 11 movies