CSC 370 — Database Systems
Assignment No. 1

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

Note 2 Working with other people is prohibited.

A note on Academic Integrity and Plagiarim

Please review the following documents:

- Standards for Professional Behaviour, Faculty of Engineering:
  `https://www.uvic.ca/engineering/assets/docs/professional-behaviour.pdf`
- Policies Academic Integrity, UVic:
  `https://www.uvic.ca/students/academics/academic-integrity/`
- Uvic's Calendar section on Plagirism:
  `https://www.uvic.ca/calendar/undergrad/index.php#/policy/Sk_0xsM_V`

Note specifically:

> Plagiarism
> Single or multiple instances of inadequate attribution of sources should result in a failing grade for the work. A largely or fully plagiarized piece of work should result in a grade of F for the course.

A program you submit will be considered a **piece of work**. You are responsible for your own submission, but you could also be responsible if somebody plagiarizes your submission.

Objectives

After completing this assignment, you will have experience:

- Writing Relational Algebra Expressions.
- Creating SQL statements.
- Interacting with postgresql using its command line interface.

IMDB

The IMDB makes available a lot of information about movies. See `https://www.imdb.com/interfaces/` for a description of this information.

A. I have created a database called *imdb* that corresponds to this information. You should be able to read its relations. Look at their names and schemas, and their foreign key constraints. These are some specifics about the data:

- Productions. Correspond to all types of productions. Use the field *productiontype* to determine the type of production (movies, tvSeries, tvSpecial, tvEpisodes, etc).
- Persons. People who have been involved making movies. Referenced by Roles and Crew.
- Roles. This relation contains the most important actor/actresses of each movie (e.g. *major roles*). See Persons.
- Directors. Lists who the directors (there might be more than one) of a production are (do not use crew for this information).
- Episodes. A production type 'tvEpisode' is linked to a placeholder for the entire series. For example, the tvSeries called 'Detectorists' has the id *tt4082744*. However, each of its episodes has a different tuple in productions, each of type 'tvEpisode' (e.g. its first 3 episodes are *tt4088818, tt4095606, and tt4095612*). The relation Episodes links the episodes to their corresponding 'tvSeries' entry in productions.
- Ratings. The rating of productions. Note that a tvSeries has an "overall" rating and each episode has its own rating.
- Genres. The genres of each production.

B. To connect to the database:

- You need to login to one of the Linux computers in the faculty. For example: linux.csc.uvic.ca
- To connect to the database you should use: host *pgstudent.csc.uvic.ca*, database name *imdb*.Your username and password have been placed in your personal gitlab repo (the file is *id.txt*) for this course). For example, your database user id might be user120
- Connect to the DBMS using psql using the following command (replace user120 with your db userid):

```
psql -h pgstudent.csc.uvic.ca -U user120 imdb
```

- In psql, you can:
  - Change the password. **DO NOT REUSE a password**. Passwords in postgresql are not secure, therefore do not use a sensitive one.
  - List the relations using `\d`
  - List the schema of a relation using `\d relationName`
  - You can output the results of a query using `\o filename`. This file is saved in the computer where you are running psql from. You stop such output using `\o`
  - Read the manual for psql for more information.
- For this assignment, use only relations that are type *relation* (output of `\d`)

Your task, should you choose to accept it

Answer the following questions, both in relational algebra, and SQL. **Relational algebra queries should match SQL queries**. If it does not, your answer will be graded as incorrect and you will not receive any marks for it.

- Submit your answers via Brigthspace as indicated in class.

- Your query should use only the information provided in the question.
- There are some restrictions imposed in several questions below. If you violate a restriction, you will get zero in that question.

Part I

For this part of the assignment you are only allowed to use projection, selection, union, intersection and difference. No other operation should be used. **You should not use group by, cross products or joins**.

1 From 2000 (inclusive), in which *movies* has *Clint Eastwood* been both a director and had a major role?

Your query should return the following result.

id	title	originaltitle	productiontype	runtime	year	endyear
tt0186566	Space Cowboys	Space Cowboys	movie	130	2000	
tt0309377	Blood Work	Blood Work	movie	110	2002	
tt0405159	Million Dollar Baby	Million Dollar Baby	movie	132	2004	
tt1205489	Gran Torino	Gran Torino	movie	116	2008	
tt7959026	The Mule	The Mule	movie	116	2018	
tt1924245	Cry Macho	Cry Macho	movie	104	2021	

2 Which are the TV series (not episodes) with an average rating above 9.0 and at least 100,000 votes that were made since 2010 (year 2010 including).

Your query should return the following result.

#+RESULTS:

id	title	originaltitle	productiontype	runtime	year	endyear
tt0944947	Game of Thrones	Game of Thrones	tvSeries	57	2011	2019
tt1475582	Sherlock	Sherlock	tvSeries	88	2010	2017
tt14392248	Aspirants	Aspirants	tvSeries	45	2021	
tt12392504	Scam 1992: The Harshad Mehta Story	Scam 1992: The Harshad Mehta Story	tvSeries	54	2020	2020
tt2861424	Rick and Morty	Rick and Morty	tvSeries	23	2013	

Part II

For this part of the assignment you can use all operations **except group-by**.

1 What is the production (or productions, as they might be more than one) directed by 'Christopher Nolan' with the highest rating.

Your query should return the following result.

averagerating	title	year
9	The Dark Knight	2008

2 Which directors have directed two or more movies with a rating of at least 8.5 and at least 500,000 votes?

Your query should return the following result.

director	title	year
Alfred Hitchcock	Psycho	1960
Alfred Hitchcock	Rear Window	1954
Christopher Nolan	Inception	2010
Christopher Nolan	Interstellar	2014
Christopher Nolan	The Dark Knight	2008
Christopher Nolan	The Prestige	2006

```
| David Fincher        | Fight Club                                          | 1999 |
| David Fincher        | Se7en                                               | 1995 |
| Francis Ford Coppola | The Godfather                                       | 1972 |
| Francis Ford Coppola | The Godfather Part II                               | 1974 |
| Frank Darabont       | The Green Mile                                      | 1999 |
| Frank Darabont       | The Shawshank Redemption                            | 1994 |
| Martin Scorsese      | Goodfellas                                          | 1990 |
| Martin Scorsese      | The Departed                                        | 2006 |
| Peter Jackson        | The Lord of the Rings: The Fellowship of the Ring | 2001 |
| Peter Jackson        | The Lord of the Rings: The Return of the King     | 2003 |
| Peter Jackson        | The Lord of the Rings: The Two Towers             | 2002 |
| Quentin Tarantino    | Django Unchained                                    | 2012 |
| Quentin Tarantino    | Pulp Fiction                                        | 1994 |
| Ridley Scott         | Alien                                               | 1979 |
| Ridley Scott         | Gladiator                                           | 2000 |
| Robert Zemeckis      | Back to the Future                                  | 1985 |
| Robert Zemeckis      | Forrest Gump                                        | 1994 |
| Steven Spielberg     | Saving Private Ryan                                 | 1998 |
| Steven Spielberg     | Schindler's List                                    | 1993 |
```

Part III

What episodes of the *tvMiniSeries* with *title Ahsoka* do not have an average rating?

Your query should return the following result.

```
| id         | episodeof  | season | episode |
|------------+------------+--------+---------|
| tt27484659 | tt13622776 |      1 |       6 |
| tt27484664 | tt13622776 |      1 |       7 |
| tt27484667 | tt13622776 |      1 |       8 |
```

1. Answer the question with a query that does not use joins or cross-products.
2. Answer the question without using the following operators in the Selection: IN, EXISTS, ALL, or ANY

Hints

1. Remember to add a semicolon at the end of your sql statement. Postgres will process a statement only after it finds a semicolon.
2. **Your relational algebra should match your SQL queries**.
3. psql will allow you to use emacs commands to traverse its history. You can also configure it to use vi commands. Read the documentation of psql.
4. In postgresql, usernames are case sensitive.
5. Once you are logged in, you need to type the semicolon at the end or every query. If you don't, psql will not process your query and way for more input. If your prompt is different from `<username>=`, then your query is not yet complete.
6. You can always use `Ctrl-c` to stop your query (either you are in the middle of typing it, or it is being processed.
7. To exist psql type `Ctrl-d`. It is bad practice to simply kill the terminal, since the dbms will still hold resources for a while until it can detect that the client is dead.

What to submit

Submit via Brightspace, under the **Assignment 1**. Further instructions in Brightspace.

- For each query, its corresponding Relational Algebra, and SQL query