Introduction to Database Systems
Homework 1: SQL

Jorge Quiané

February 2023

1 Instructions

Write SQL queries to retrieve the information requested below from your PostgreSQL database. Then enter each query, as well as the numerical answer to each question, in learnIT. The query should not return anything except the answer. As this is a learning exercise, you should take time to test variants and make the queries as simple as you possibly can. The queries should also be well and consistently formatted, and as readable as possible, as SQL queries are generally part of your code base.

Although you can use subqueries, you should not answer the question using a sequence of several queries, and you should not use any "magic constants" in your queries. Note that in many cases, some (incorrect) variants of some queries could return the same results as a correct query. The fact that a query returns the correct answer thus does not necessarily mean that the query itself is correct. Therefore, you must take care to insert additional data to test your queries adequately. In short, the query must work for **any instance of the schema**.

All the queries ask you to count the rows that satisfy some criteria. You should start by first creating a query that returns all rows that satisfy the given criteria. Once your query is ready, you can then turn it into a counting query, either by replacing the output columns with a `count` statement, or by enveloping the query with a counting query:

```
select count(*)
from (
    select ...
) tmp;
```

In the exam, you will also get directions along the following lines (but given that this is not an exam, you need not worry about partial credit): Queries must work for any database instance and should avoid system-specific language features (including `LIMIT 1`). Queries should not return anything except the answer; a query that returns more information will not receive full points, even if the answer is part of the returned result. A sequence of several queries that answer the question will not receive full points, but subqueries and views can be used. Queries should be as simple as possible; queries that are unnecessarily complex may not get full points, despite returning the correct answer. If you are unable to complete the query, you can still submit your attempt, along with a brief description of the problem with the query, and you might still get partial points.

2 Database description

In this homework, you will work with the database `imdb`. To start working with the database, you must first create it and then run the `HW1-DB.sql` script available on LearnIT using `psql` on your laptop:

```
psql [-U postgres] [-q] your_database_name < HW1-DB.sql
```

The database contains information on movies and people. Most of the data is actually part of the real IMDb database, but some information is fabricated. The following is the relevant part of the database schema (the relations `person` and `movie` have some additional attributes that are not used in the homework).

> `person`(<u>id</u>, name, gender, ...)
> `movie`(<u>id</u>, title, year, ...)
> `genre`(<u>genre</u>, category)
> `movie_genre`(movieId, genre)
> `role`(<u>role</u>)
> `involved`(<u>movieId</u>, <u>personId</u>, <u>role</u>)

All relations have a defined primary key (all underlined attributes) except `movie_genre`. In particular, the `involved` relation has a composite primary key consisting of all three attributes. Foreign keys are defined where appropriate. The `role` relation is a lookup table with a list of all possible roles. The meaning of other relations and attributes should be self-explanatory.

3 Passing grade

The maximum grade is 40 points. You get 4 points for each correct SQL query and 1 point for each correct numerical answer. To pass the homework, you need to have *at least 5/40* points. However, the more you do, the more feedback you get and the more you learn.

4 Queries

In the following, you are given the correct answer of a very similar query but not the correct answer for the query you should produce in the end. Start by finding the correct form for the given answer and then convert the query to the requested answer. As outlined above, each query should return only a single numerical answer. Answer each of the following questions using a single SQL query on the `imdb` database:

(1) The `person` relation contains 51,052 entries with a registered gender. How many records are there with gender 'f'?

(2) In the database, there are 365 movies for which the average height of all the people involved is less than 165 centimeters (ignoring people with unregistered height). What is the number of movies for which the average height of all people involved is greater than 195 centimeters?

(3) The `movie_genre` relation does not have a primary key, which can lead to a movie having more than one entry with the same genre. What is the highest number of such duplicated entries in one movie?

(4) According to the information in the database, 476 different people acted in movies directed by 'Francis Ford Coppola'. How many different people acted in movies directed by 'Roger Spottiswoode'?

(5) Of all the movies produced in 2002, there are 12 that have no registered entry in `involved`. How many movies produced in 2011 have no registered entry in `involved`?

(6) In the database, the number of people who have acted in exactly one movie that they have also self-directed is 603. What is the maximum number of times a single person has both acted and directed in the same movie?

(7) Of all the movies produced in 2002, there are 282 that have entries registered in `involved` for *all* roles defined in the `roles` relation. How many movies produced in 2011 have entries registered in `involved` for all roles defined in the `roles` relation? *Note: This is a relational division query which must work for any schema; you can not use the fact that currently there are only 2 different roles to write a 'magic number'.*

(8) The number of people who have played a role in movies of *all* genres in the category 'Newsworthy' is 156. How many people have played a role in movies of all genres in the category 'Popular'?