Lab Assignment 4

(due Friday May 29, 2020 before midnight)

1 Introduction

In the first part of this assignment, you are asked to specify a number of additional constraints for the schema to ensure that the data inserted in the database are meaningful. You are also asked to specify triggers to ensure that deletions are propagated through the tables. The documentation found in http://www.postgresql.org/docs/8.3/static/ddl-constraints.html may be useful. Since we already have our schema declared, the ALTER TABLE ... ADD CONSTRAINT command should be used appropriately.

2 Add Constraints

In this section, you are asked to add some constraints for database. If some of these constraints have been set in your lab assignment 1, then just list your commands again.

C1 Add a constraint such that actors must have at least two strings (a first name and last name), and there can be no digits. You must also verify there is at least one vowel, and two consonants in each string. Each string must also start with a capital letter.

C2 Add a constraint such that actors cannot be older than 100 years old, and no younger than 10.

C3 Add a constraint such that actors must be either Male, Female or Non-binary.

C4 Add a constraint such that a review_text is no longer than 2000 characters, and there aren't more than 20 sentences, and there are less than 1000 words. Here you can assume that a period indicates a new sentence, and that a space indicates a new word. You should also validate that the first letter of the review_text is a capital letter.

C5 Add a constraint such that rating and imdb_rating are both floats in the range of 1 through 10.

C6 Set or alter some constraints so that when a record is deleted from the movies and actors tables, all records in the other three tables referring to the record are also deleted. Finally, delete the record from movies table where movieid equals 7 and delete the record from actors where the actorid equals 5.

C7 Write an insert into command with data of your choosing, that fails C1 - C5. The insert into should only fail because of the added constraint.

3 Check Functional Dependencies

In this section, you are asked to issue SQL commands to check for functional dependencies in the database tables.

C8 Issue an sql query that checks whether or not the functional dependency {title, genre} $\rightarrow$ {director} hold in the Movies table. Your query should return empty set if the functional dependency holds. Describe (textually, in your submission file) how sql checks this condition.

C9 Issue an sql query that checks whether or not the functional dependency {name, gender} $\rightarrow$ {birthyear} hold in the Actors table. Your query should return empty set if the functional dependency holds. Describe (textually, in your submission file) how sql checks this condition.

C10 Make at least one update or insertion that violates C8 and C9, and use your sql query to verify the violation.

4 Inserting New Data

Q1 Insert data associated with your favorite movie(s) in each of the five tables. Only one new entry per table is required, but feel free to add as many as you want. You should only increment your movieid and actorid, i.e., the entries you're inserting must have a movieid > 9 and an actorid > 13.

Q2 Return a list of all the remaining movie titles, with an extra column titled "New Movie", which is True if the movieid is > 9, and False otherwise. You should also include the imdb_id and the Reviews rating associated with each movie.

Q3 Return a list of all the remaining actor names, with an extra column titled "New Actor", which is True if the name is the same as one of the actors you've inserted, else False. Note: You can hardcode the actor names in this query.

Q4 Return a list of all the remaining movieroles, with an extra column titled "New Role", which is True if the role is associated with a movie or actor you've inserted, else False.

Q5 Write a query which combines at least four of the commands you've learned so far (join, in, exists, group by, order by, union, row_number, split_part, regex_split_to_table, case, or an aggregation function) to return some interesting results. **Important: This query SHOULD NOT match any of the previous queries. You MUST indicate the intended output of your query as a comment above the command and results in your submission file.**

5 Submission

Save the commands issued (do not include the output of these commands) in a file named Lab4.txt. Upload this file to the Lab Assignment 4 submission page on Canvas by 11:59pm on Friday May 29th.