SOEN 363: Data Systems for Software Engineers

Assignment 2, Fall 2024

October 22, 2024

Date posted: Tuesday, October 22nd, 2024.

Date due: Friday, November 15th, 2024, by 23:59.

Weight: 5% of the overall grade.

Individual assignment. You must work strictly on your own.

Overview

In this assignment, you create a local database of movies and their information. The assignment targets writing SQL queries for data definition, data collection, as well as search queries.

While IMDB [1], TMDB [2], or any other public movie database may be used as the main source of the data, in this assignment you are required to create your own database and populate the data. The data population may be done through public APIs [3] such as IMDbOT [4]. Watchmode [5] provides a mapping between IMDB and TMDB IDs as well as streaming information. Most APIs return requested data in JSON format. Using a JSON Visualizer [6] would be extremely helpful in locating the info items that are returned by the API.

Example:

```
GET https://search.imdbot.workers.dev/?tt=tt0068646
```

You may optionally start with some pre-populated database (see ref. [7]), however, your database must strictly follow the following design.

> Movies are searchable by TMDB-ids and IMDB-ids. Both may be used for alternate search keys. Using TBDM-id is required, however, IMDB-id is optional. It is recommended that you use your own primary key. For each movie, store: title, plot, content rating (certificates), viewers rating (a floating point number, between 0 and 10, with 1 decimal digit after the decimal point), genres, actors, directors, release year, AKAs, countries, languages, keywords. Also, store the watchmode id for each movies. In theory not all movies have a watchmode id. Make sure your database only contains few movies with watch-mode id.

Note that both IMDbOT and the above Sample Movies Database are given as examples. You may find alternatives sites on the internet. You are allowed to use any data source that fits the purpose.

Implementation Platform

In this assignment, you use PortgreSQL [8] to implement the database tables.

DDL Queries

Provide answer to each of the following parts in separate `.sql` files.

Part 1. [20 pts] Provide the DDL queries for creating the database tables that correspond entities and their relationships.

- Use internal integer primary keys (no TMDB/IMDB id).
- Include referential integrity / unique constraints.
- Include full country name and short code for countries.
- Create foreign tables for genres, content rating, and keywords.
- Use singular names for tables / relations.
- Make sure your database include a few records with missing IMDB-ids. You may manually delete the IMDB data and/or use fake information.

Data Population

For the data population of the data, demonstrate:

Part 2. [15 pts] Consume a web service, either via a script or a small code that performs get operations for each movie in question.

Part 3. [15 pts] Create database instance via a database client (i.e. Psycopg2 for python [9]). You may chose any programming language as you wish. Using Python is not mandatory.

ERD

Part 4. [7 pts] Generate the ERD of your database using a reverse engineering tool i.s. pdadmin 10.

Database DDL

Part 5. [8 pts] Provide the DML queries that populates the data in the database. You may back up your database as SQL (see pddump [11] as an example). Make sure you create at least 50 movies. Remove unnecessary statements from your sql dump file. Note that the graders will run the DDL queries in the previous section to create your database. Make sure the queries are provided in correct order.

Use of Views

Part 6. [7 pts] Create the following view namely 'movie-summary' that displays tmdb key, imdb key, title, description / plot, content rating, runtime, number of keywords, number of countries. Provice the DDL SQL.

Queries

Part 7. [28 pts] Provide the following queries:

A) **[4 pts]** Find the total number of movies with and without IMDB id in the database. Use one query.

B) **[4 pts]** Pick an actor. Find all movies by that actor that is released between 2000 and 2020. List TMDB-id, IMDB-id, movie title, release date, and watchmode-id.

C) **[4 pts]** Find movies that have highest number of reviews. List top 3.

D) **[4 pts]** Find number of movies that are in more than one language.

E) **[4 pts]** For each language list how many movies are there in the database. Order by highest rank.

F) **[4 pts]** Find top 2 comedies (higher ratings).

G) **[4 pts]** Write a batch-update query that rounds up all the ratings.
IMPORTANT: Do not run the query. The grader will run the query and verify the result.

Submission

Make sure all above queries return data. Modify the data in your database, if necessary.

Submit your assignment electronically on Moodle: `https://moodle.concordia.ca`
Include your name and student ID in the submission. Make sure that you upload the assignment to the correct assignment box on Moodle. No email submissions are accepted. Assignments uploaded to the wrong system, wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle prior to the submission deadline. The deadline will not be extended.

References

1. https://www.imdb.com/
2. https://developer.themoviedb.org/reference/search-movie
3. https://github.com/public-apis/public-apis
4. https://github.com/SpEcHiDe/IMDbOT
5. https://rapidapi.com/meteoric-llc-meteoric-llc-default/api/watchmode/details
6. https://jsonviewer.stack.hu/
7. https://www.databasestar.com/sample-database-movies/
8. https://www.postgresql.org/
9. https://www.freecodecamp.org/news/postgresql-in-python/
10. https://www.pgadmin.org/docs/pgadmin4/development/erd_tool.html
11. https://www.dbvis.com/thetable/a-complete-guide-to-pg-dump-with-examples-tips-and-tricks/