Create Movie Recommendation Database

Due Jan 25 by 11:59pm **Points** 20
Submitting a text entry box or a file upload
Available Jan 11 at 12am - Jan 25 at 11:59pm 15 days

This assignment was locked Jan 25 at 11:59pm.

Introduction

You have learned how to design a movie recommendation database. The assignment will give you an opportunity to create such a database from scratch and build applications on top of this database.

Background

In this database, a movie has two attributes: id, title. A possible movie record is as follows: 54796, 2 Days in Paris (2007).

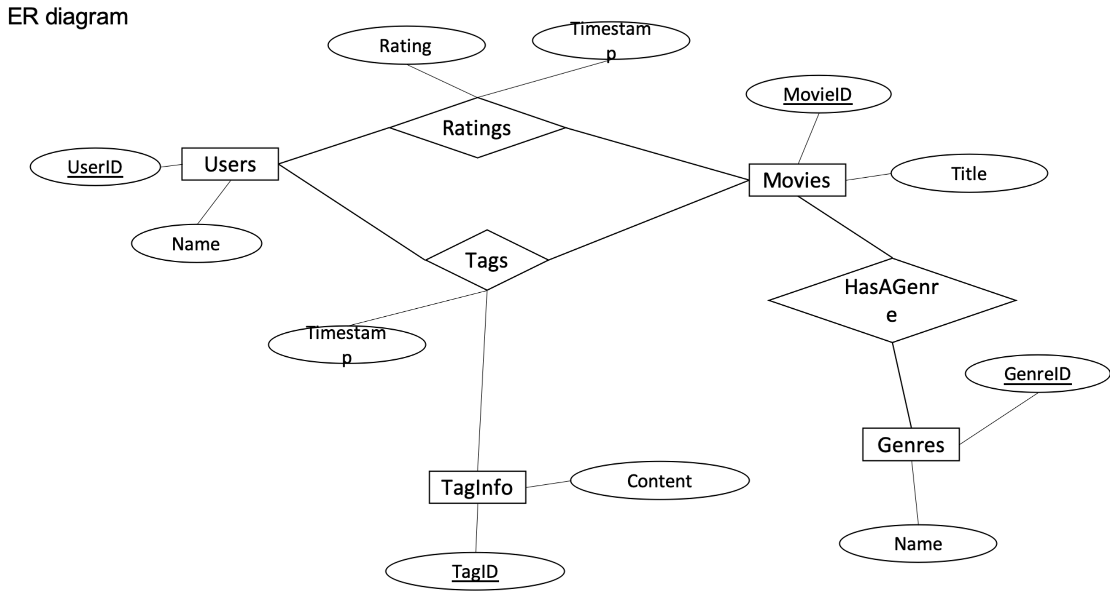
A movie can be categorized into multiple genres. A genre is selected from Action, Adventure, Animation, Children’s, Comedy, Crime and so on. A movie may not have a genre.

A user can give a 5-star scale rating (0-5) to a movie. For instance, User (ID 4) gave 4 stars to Movie “God Father”. A user can only rate a movie once. The database needs to log each rating operation. The database should not allow any out-of-range ratings

A user can also assign a tag to a movie. A user can tag a movie multiple times. For instance, User (ID 20) assigned “very cool” tag to Movie “Mission: Impossible – Ghost Protocol”. Two days later, he added a new tag “unbelievable” to the same movie. Each tag is typically a single word or short phrase. The meaning, value and purpose of a particular tag are determined by each user. The database needs to log each tagging operation.

Requirement

According to the database design made by you, the movie database includes multiple tables. In particular, you need to consider seven tables: users, movies, taginfo, genres, ratings, tags, hasagenre. In this phase, you must create these tables and load the corresponding data into these tables.



1. The description of the tables is as follows. You should also check the requirement in the given graphic description:

users: userid (int, primary key), name (text)

movies: movieid (integer, primary key), title (text)

taginfo: tagid (int, primary key), content (text)

genres: genreid (integer, primary key), name (text)

ratings: userid (int, foreign key), movieid (int, foreign key), rating (numeric), timestamp (bigint, seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970)

tags: userid (int, foreign key), movieid (int, foreign key), tagid (int, foreign key), timestamp (bigint, seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970).

hasagenre: movieid (int, foreign key), genreid (int, foreign key)

2. The requirement only tells you the name and data type of each attribute in each table. You need to figure out the primary keys, foreign keys, constraints or other necessary settings by yourself. The key information in the requirement is not complete and attributes can be primary keys and foreign keys at the same time.

Remarks

1. All table names and attribute names must **be in lowercase letters and exactly same with the specification**.

2. You can use COPY FROM / INSERT to load data and test your tables but don't put it in the submission.

3. **Your SQL script should NOT contain the commands to create database, change database or set encoding. Please do not use any commands to change Postgres DB settings.** This may crash the grading environment! Your script should just simply create tables. I will decide the working database and all other parameters.

Test data

We provide some test data for you to try. The delimiter of all files is "%". But the grading system will use more test data and test cases to try your SQL script.

Data link:

https://github.com/jiayuasu/Coursera-ASU-Database/tree/master/course1/assignment1/exampleinput (https://github.com/jiayuasu/Coursera-ASU-Database/tree/master/course1/assignment1/exampleinput)

Submission

Submit a single plain text file "solution.sql"