Introduction

The assignment will create a movie recommendation 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 me, the movie database includes multiple tables. In particular, consider seven tables: users, movies, taginfo, genres, ratings, tags, hasagenre. In this phase, I have created these tables and loaded the corresponding data into these tables.

1. The description of the tables is as follows.

 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)

Test data

The delimiter of all the data files is “%”. But the grading system will use more test data and test cases to try your SQL script.