Ira A. Fulton Schools of Engineering
Arizona State University

CSE 511: Data Processing at Scale

Assignment 1: Create a Movie Database

Purpose

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.

The Predator — September 14, 2018
Christopher Robin — August 3, 2018
Searching — August 24, 2018
Mission: Impossible -- Fall... — July 27, 2018

Objectives

Learners will be able to:

- Create simple tables in a database with the appropriate datatypes for the columns.
- Recognize how to add rows to a table once it's been created.
- Produce relationship tables to link two or more tables.
- Distinguishing the use of foreign keys, primary keys, and other constraints.

Technology Requirements

- PostgreSQL

Assignment Description

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 necessarily have a genre.

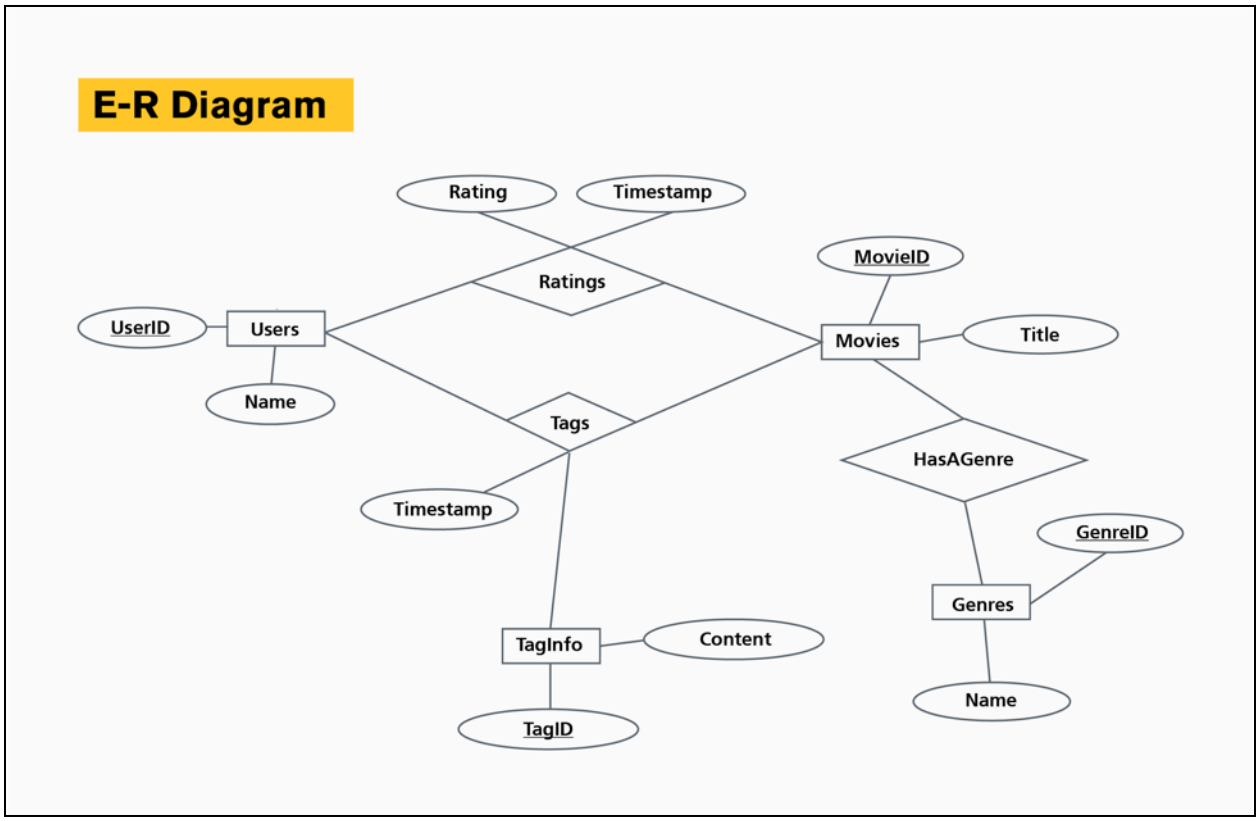
A user can give a 5-star scale rating (0-5) to a movie. For instance, User (ID 4) gave 4 stars to the movie "The GodFather". A user can only rate a movie once. The database needs to log each rating operation. The database should not allow any out-of-range movie ratings.

A user can also assign a tag to a movie. A user can tag a movie multiple times. For instance, User (ID 20) assigned a "very cool" tag to the 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.

Directions

In addition to the directions below, please review the "**Introduction**" and "**Submission and Feedback**" **videos** which are located in the "**Week 1: Overview**" section.

Based on the movie database design, the movie database includes multiple tables. For this assignment, you need to create seven tables: users, movies, taginfo, genres, ratings, tags, hasagenre. Then load the corresponding data into the tables.



1. The description of the tables is as follows:
    a. users: userid (int, primary key), name (text)
    b. movies: movieid (integer, primary key), title (text)
    c. taginfo: tagid (int, primary key), content (text)
    d. genres: genreid (integer, primary key), name (text)
    e. ratings: userid (int, foreign key), movieid (int, foreign key), rating (numeric), timestamp (bigint, seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970)
    f. 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).
    g. 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 for

the database by yourself. The key information in the requirement is just a skeleton and not complete; attributes can be primary keys and foreign keys at the same time.

Test Data:

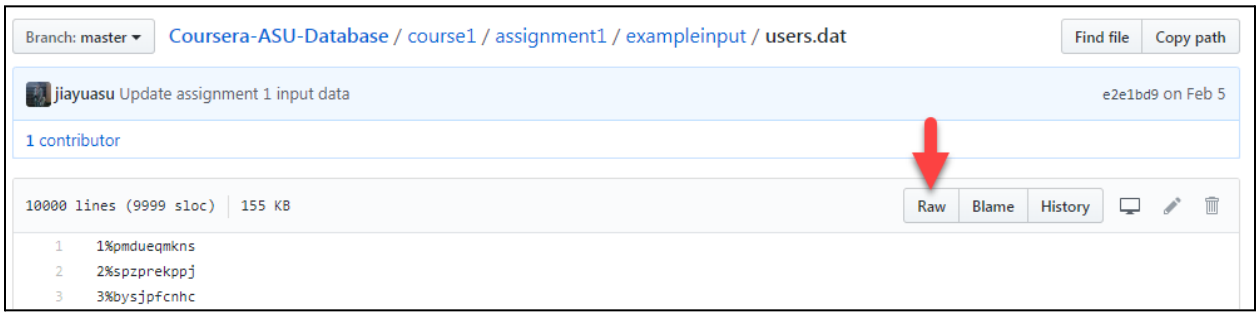
We provide some test data for you to try. The delimiter of all files is “%”. Note that the grading system will use more test data and test cases to try your SQL scripts.

Data link:

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

To download the data:

1. Go to the file you want to download.
2. Click it to view the contents within the GitHub UI.
3. In the top right, right-click the "Raw" button.
4. Right-click and select "Save as".



Assignment Tips:

1. All table names and attribute names **must be in lowercase letters and match the specification**.
2. You can use COPY FROM / INSERT to load data to test your tables but don’t include the test data in the submission. Information on test data is provided in the Test Data section below.
3. **Your SQL script should NOT contain the commands to create a database, change the 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.

4. Remember that you may make edits to your submission and resubmit as many times as you would like. If you have trouble submitting your assignment, we encourage you to ask questions to your peers as many of them may have encountered similar problems and found a solution.

Submission Directions for Assignment Deliverables

Submit a **single plain text file "solution.sql"**. This file should contain the SQL scripts for creating the seven tables.

When you are ready to submit:

1. Go to "**Programming Assignment: Assignment 1: Create Movie Recommendation Database**".
2. Click on the "**My submissions**" tab (located at the top of the page under the deadline date).
3. Click on "**Create submission**."
4. Upload one file for the assignment and click "**Submit**."
    a. If there is an error in your assignment, you may make corrections and resubmit.

Test Cases:

Upon submission the autograder will test your code against the following test cases:

- TEST CASE 1: Insert normal data
- TEST CASE 2: Insert non-exist foreign key
- TEST CASE 3: Insert duplicate rating
- TEST CASE 4: Insert a hasagenre record that contains wrong genre id
- TEST CASE 5: Insert a rating larger than 5

Evaluation

There are five test cases for a total of 1 point, so each test case is worth 0.2 points. **If your .sql fails, you will see the corresponding .sql error logs that indicate *where* the error occurred.** In the end, if the submission runs correctly, you will see feedback "You passed 5/5 tests."

The test cases are executed in a simultaneous manner. If you pass any 2 of the 5 test cases, you would receive 40% of the total marks.

Learner Checklist

Prior to submitting, read through the Learner Checklist to ensure you are ready to submit your best work.

- ☐ Did you title your file correctly and convert it into a single **.sql file**?
- ☐ Did you answer all of the questions to the best of your ability?
- ☐ Did you make sure your answers directly address the prompt(s) in an organized manner that is easy to follow?
- ☐ Did you proofread your work?