MovieNet: A Social Network for Movie Enthusiasts

445 Course Project

MovieNet is a social network for movie enthusiasts, containing a database of movies, actors/actresses, directors, etc., and a social network of movie enthusiasts. Users of MovieNet can search for movies and actors/actresses. They can also rate movies. They can further be friends with each other based on movie interests or other similarities. An online movie store can pay to join MovieNet and publish ads that are customized to each user based on her movie interest.

The MovieNet that you design and develop should meet the requirements, including the minimum functionalities, good performance for common user operations, and a significant extension the project, e.g., an integration of a movie database with a social network or an online movie store. Grading of the project combines four specific requirements as follows:

Grading scheme:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>Data parsing and schema design (Section 1)</td>
</tr>
<tr>
<td>40%</td>
<td>Correct implementation of minimum functionalities (Section 2)</td>
</tr>
<tr>
<td>20%</td>
<td>Performance of required features (Section 3)</td>
</tr>
<tr>
<td>20%</td>
<td>A major extension of the project (Section 4)</td>
</tr>
</tbody>
</table>

The four requirements are explained in detail in Sections 1-4, respectively. The data set that we provide for building the movie database is described in Section 5.

1 Data Parsing and Schema Refining

The movie dataset is given as .txt files delimited by tab (‘\t’) and contains incomplete information. It is part of the project to properly parse the data, decide if you want to decompose the big flat dataset into multiple relations, and figure out how. In your final project report, please include:

1. concrete examples of incomplete records in the dataset,
2. what you have done to parse them, and
3. your schema design.

2 Minimum Functionality Requirements

The minimum requirements regard the Internet movie database that you are asked to build. The Internet movie database must support the following features.

2.1 Searches

Your application should support three types of searches. These searches can be performed by any user without having to login (i.e., without being registered on the site).

A. Simple searches based on properties of movies. For this category, your application
should support the following queries:

- Find all movies that were released in the input year range.
- Find movies with title given as input. Users should be able to search for both exact matches and substring matches (using the LIKE construct in SQL).

**B. Complex searches involving multiple entities.** For instance, the join operator and aggregates should be properly supported. For example,

- Find names of actors/actresses who performed in more than 10 MPAA rated movies in a given year.
- Find names of movies that have been rated by more than 1000 users.

**C. Top K searches**

- Show top k (k≥1) movies based on average user rating. In case of a tie, show any top k movies.

### 2.2 User Login and Rating of movies

As mentioned before, users can rate a movie in the database. The database stores the average rating of a movie based on up-to-date rating information.

**User login**

For users to rate movies, they need to register to the site and create logins for themselves. After they login to the site with the correct user name and password, they can rate movies.

**Rating of movies**

A user can rate a movie in the range of 1-10, and is allowed to rate a movie at most once. Every time a new rating is generated, the average rating for the movie is updated in the database. *(Hint: you may want to store the average rating of a movie and the count of users who have rated this movie so far for such updates). It is up to you, the owner of the database, to decide if you allow a user to update/delete his rating of a movie.*

### 2.3 SQL Interface for the Administrator

The database administrator is a special user with the user name “admin”. She has unrestricted access to all the data in the database.

**Searches by the Administrator**

Besides the search interface for regular users (i.e., using text boxes on a search web page), you web site should also support the direct input of SQL queries for use by the administrator.

**Updates by the Administrator**

Information about the movies, actors/actresses, and ratings, etc. in the movie database can only be updated with an administrator login. *(Hint: you can use the same SQL interface for both searches and updates by the administrator.)*
2.4 Constraints
While many of the constraints to which you need to pay attention are described above, they are summarized and re-iterated below.

Parsing and Loading the dataset
- In the loading process, you may also want to create values of other attributes that are in the schema of your database but not in the initial data sets. If an attribute is present in more than one table in your schema, please make sure that values of this attribute are consistent across all tables.

Updating the database
- After the initial loading, only the administrator is allowed to add tuples representing new movies or new actors/actresses, etc. to the relevant tables, and to update or delete existing tuples about movies and actors/actresses, etc.
- The database can also be updated with the information of user login when a user registers to the site, and the information of user ratings.

Need of user login
- A user can search the database without having to login.
- But login is needed to be able to rate movies.

Ratings of movies
- To ensure fair rating, each user is allowed to rate a movie only once.

3 Performance Requirements
Your application is expected to run with reasonable performance. While there are no specific requirements in terms of response time, most operations on the web site, including user registration, user login, searches, and updates, should be completed with no obvious delay.

If you observe significant delay in common operations, it is time for you to review your design and implementation to identify the causes of slow performance. As general guidelines, you may want to consider the following things:

- Can you refine the database schema to improve performance? You may want to apply the theory of normalization that we learned in class.

- Can you create additional indexes to improve performance? Refer to the lecture on physical design and tuning for a detailed discussion of this topic. The SQL command for creating an index is CREATE INDEX index_name ON table_name ( column_name(s) ).

- Did you implement the functionalities of your application in the right place of your program? For example, a join between two tables should be performed in the database backend. Performing the join in your PHP code is a mistake—even though you may still provide correct answers, the performance of your application will evitably suffer.
4 Project Enhancement

Your application should have a significant extension beyond the basic functionalities described above. The following list includes several ideas for the project enhancement. Students can feel free to refine these ideas and implement different variants of them—this is the part of the project where you have the most flexibility. If you believe that you have a brilliant new idea, please come to discuss with the instructor. If your idea is deemed feasible for implementation within a semester and to involve a similar amount of work as other ideas, it will qualify as an extension to the basic functionalities.

To facilitate the extension of your project, the 445 staff will provide you with a large user table with synthetic user data including the following attributes:

- user email (unique),
- user name,
- password,
- age,
- gender,
- location

Social networking among movie enthusiasts

Users of the movie database would like to be friends with each other based on their movie interests (according to their ratings), age and location of residence, etc. In addition, friends would like to see each other’s activities, such as the movies watched recently, the rating of the movie, the review of the movie, and so on. One feature of the MovieNet is to send in-real updates about such friend activities. That is, the updates about a MovieNet user will show up immediately in the windows of her friends online. This update feature is similar to that supported by facebook, youtube, windows live etc., offering a great opportunity for users to know what friends are doing.

In your implementation, you should develop an extension of the database to generate the structure of the social network (i.e., randomly selecting 5-10 friends for each user), store the updates about each user, and display the updates on her friends’ side soon after the updates occur.

Customized advertisements

Online advertisements are important for an online movie store to make profit, but MovieNet users would only want to receive advertisements that match their interests. Advertising based on user movie preference, her movie search history, or her web page visit history has been a popular strategy. Your application can even use the user friends’ movie interests to help online advertising.

In your implementation, you might need to store user movie search history and web page visit history (in addition to the movie preference already stored in her profile), do some analysis of all of this information in order to figure which movies may interest the user, and display those movies in the user window. If the advertising strategy is purely based on user search/visit history, it probably will be random at the beginning and become more relevant as information gets accumulated over time. If a customized ad indeed interests the user (e.g., she clicked on it), the
online store will pay MovieNet for this ad. When the user actually requests to buy the movie, then the online store makes a profit.

Below are some other (smaller) ideas. Please feel free to integrate them with the extensions outlined above or to significantly expand these ideas into a major extension of your the project.

- You can create a user community where a user may maintain a list of his favorite movies. Other users can view these lists. You can further allow users to find the common set of favorite movies among them.
- Your application can also have a notification service, e.g., using XML/RSS feeds to report to interested users the latest addition to the movie database or top 10 movies this week.
- You may want to augment the security of your application, e.g, developing mechanisms to protect the database from a variety of attacks.

5 Data Sets

You will be provided with several data sets in the form of text files. It is an important initial step of your project to parse the data in those files and load it into the database that you will create based on your designed schema. After the data is loaded into the database, your movie database is ready to accept requests for search, update, and deletion.

The format of each data set is explained in order below. All of them are tab-delimited.

(1) movies.txt
(2) mpaa.txt
(3) users.txt
(4) ratings.txt

(1) movies.txt
The file contains real-world movie data. Each movie consists of multiple lines. An empty line separates two movies. The format is:

Title<TAB>the title
Year<TAB>the year
Running Time<TAB>length in minutes
MPAA Rating<TAB>the rating
Genres<TAB>genre1<TAB>genre2...
Key Words<TAB>word1<TAB>word2...
Producers<TAB>producer namel<TAB>producer name2...
Directors<TAB>director namel<TAB>director name2...
Editors<TAB>editor namel<TAB>editor name2...
Actor<TAB>actor namel<TAB>role name1
Actor<TAB>actor name2<TAB>role name2
Actress<TAB>actress namel<TAB>role name3
Actress<TAB>actress name2<TAB>role name4
<EMPTY LINE>
Title<TAB>the title of another movie
...<EMPTY LINE>
Title<TAB>the title of another movie
...

The content of the file looks like:

Title  The Dark Knight Rises
Year   2012
Running Time  165
MPAA Rating   PG-13
Genres  Action Crime  Thriller
Key Words action-hero  bat  billionaire  car-crash  catwoman
Producers Nolan, Christopher (I)  Roven, Charles  Thomas, Emma (I)
Directors Nolan, Christopher (I)
Editors  Smith, Lee (II)

Actor  Bale, Christian  Bruce Wayne
Actor  Oldman, Gary  Commissioner Gordon
Actress Hathaway, Anne  Selina
Actress Cotillard, Marion  Miranda

Title  Life of Pi
Year   2012
Running Time  127
MPAA Rating   PG
Genres  Adventure Drama
Key Words christian  india  lifeboat  starvation  tiger
Producers Lee, Ang  Netter, Gil  Womar, David
Directors Lee, Ang
Editors  Squyres, Tim
Actor  Sharma, Suraj  Pi Patel
Actor  Khan, Irrfan  Adult Pi Patel
Actor  Belur, Gautam  Pi Patel (5 Years)
Actress Tabu (I)  Gita Patel

Some properties of this real movie data set may be different from the ideal scenario you have seen in Assignment 1:

- A movie can be uniquely identified by the combination of its title and year. A movie always has a line for “Title”, and a line for “Year”. For some movies, the year is 0, which means the actual year is unknown.

- All the attribute lines, except “Title” and “Year”, could be missing for a movie. For example, the directors or producers may be unknown for a movie.

- A movie may have multiple values for “Genres”, and the values are listed in the same line. The same for “Key Words”, “Producers”, “Directors” and “Editors”.

- The actors/actresses of a movie are listed in separate lines. Each line contains the name of the actor/actress after the first TAB, and the name of the roll after the second TAB. An actor/actress may play multiple rolls in a movie. Multiple actors/actresses may play the same roll in a movie. If a roll name is “\N”, it means the actual roll name is unknown.

- A person can be an actor/actress, a producer, a director and an editor at the same time in the movies (even in the same movie). A person can be uniquely identified by her name. In the case that multiple persons have the same name, special tags (such as roman numbers) have been appended to the name of each person in the data set to remove ambiguity.

Note that you can use the content in this file for personal and non-profit use only. You should not
distribute the content in this file.

(2) mpaa.txt
The file contains the description of MPAA ratings. Each line contains 3 fields separated by a tab.

- MPAA rating
- definition
- description

A line in the file looks like:

G General Audiences. All Ages Admitted. A G-rated motion picture contains nothing in theme ...

(3) users.txt
The file contains synthetic data of users. Each line contains 6 fields separated by a tab.

- email (unique)
- name
- password
- age
- gender
- location

A line in the file looks like:

sheldon@def.com John Smith 45G5x$dd 21 Male Massachusetts

(4) ratings.txt
The file contains synthetic data of user ratings. Each line contains 4 fields separated by a tab.

- user email
- movie title
- movie year
- rating (1-10)

A line in the file looks like:

sheldon@def.com The Shawshank Redemption 1994 6
The text files for this project will be uploaded to the “/usr/net/ftp/pub/cs445/project/” on the edlab machines.