MovieNet: A Social Network for Movie Enthusiasts

445 Course Project

Yanlei Diao
UMass Amherst
Overview

- 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 artists.
  - 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.
Project Requirements

- Schema design, data cleaning and loading (20%)
- Basic functionalities of a movie database (40%)
  - Searches: from simple to complex
  - User login and ratings
  - DB admin: updates, direct SQL queries
  - Constraints: schema design, data loading, user ratings
- Performance requirements (20%)
  - A few seconds for most queries
  - Normalization, physical tuning with indexes
- Web interface and extensions beyond above (20%)
  - Social networking among movie fans
  - Customized advertisements...
**Step 1: ER Diagram**

- **Entity sets**
  - Movies, Actors/Actresses, Directors, Producers, MPAA ratings, Genres, Keywords...
  - Deduplication: e.g., “John Smith I”, “John Smith II”
  - Users (login, name, age…), Friend network
  - Online movie company?

- **Relationship sets**
  - A movie has a cast, a director (optional), a producer (optional), genres (optional), keywords (optional), ratings (optional), reviews (optional)…
  - Users can rate and write reviews for movies, and be friends with each other…

- **Constraints**
  - A user can rate a movie once? Age constraint?
Data Sets

<table>
<thead>
<tr>
<th>Title</th>
<th>The Dark Knight Rises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2012</td>
</tr>
<tr>
<td>Running Time</td>
<td>165</td>
</tr>
<tr>
<td>MPAA Rating</td>
<td>PG-13</td>
</tr>
<tr>
<td>Genres</td>
<td>Action, Crime, Thriller</td>
</tr>
<tr>
<td>Key Words</td>
<td>action-hero, bat, billionaire, bomb, car-crash, catwoman</td>
</tr>
<tr>
<td>Producers</td>
<td>Nolan, Christopher (I), Roven, Charles, Thomas, Emma (I)</td>
</tr>
<tr>
<td>Directors</td>
<td>Nolan, Christopher (I)</td>
</tr>
<tr>
<td>Editors</td>
<td>Smith, Lee (II)</td>
</tr>
<tr>
<td>Actor</td>
<td>Bale, Christian, Bruce Wayne</td>
</tr>
<tr>
<td>Actress</td>
<td>Hathaway, Anne, Selina</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Life of Pi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2012</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
### Data Sets

<table>
<thead>
<tr>
<th>Users</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>email&lt;TAB&gt;user_name&lt;TAB&gt;password&lt;TAB&gt;age&lt;TAB&gt;gender&lt;TAB&gt;location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email&lt;TAB&gt;user_name&lt;TAB&gt;password&lt;TAB&gt;age&lt;TAB&gt;gender&lt;TAB&gt;location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:sheldon@def.com">sheldon@def.com</a></td>
<td>John Smith</td>
<td>45G5x$dd</td>
<td>21</td>
<td>Male</td>
</tr>
<tr>
<td><a href="mailto:jsmith@abc.com">jsmith@abc.com</a></td>
<td>John Smith</td>
<td>johnpass</td>
<td>50</td>
<td>Male</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ratings</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>user email&lt;TAB&gt;title&lt;TAB&gt;year&lt;TAB&gt;user_rating</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user email&lt;TAB&gt;title&lt;TAB&gt;year&lt;TAB&gt;user_rating</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:sheldon@def.com">sheldon@def.com</a></td>
<td>The Shawshank Redemption</td>
<td>1994</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:sheldon@def.com">sheldon@def.com</a></td>
<td>The Dark Knight Rises 2012</td>
<td>2012</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPAA</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPAA rating&lt;TAB&gt;definition&lt;TAB&gt;description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPAA rating&lt;TAB&gt;definition&lt;TAB&gt;description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>General Audiences.</td>
<td>All Ages Admitted.</td>
<td>A G-rated motion picture contains ...</td>
<td></td>
</tr>
</tbody>
</table>
Notes on the Data Set

- 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.
Upcoming Deadlines

- Groups formed on Feb 10
  - Please email help-cs445@edlab-mail.cs.umass.edu

- Proposal due on Feb 24
  1. Extension beyond basic functionality and performance requirements
  2. E/R diagram for your entire application
Raw Dataset (1)

- movies.txt (393,289,550 bytes)
  - Schema:
    - Each movie consists of multiple lines.
    - An empty line separates two movies.

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 name1<TAB>producer name2...
Directors<TAB>director name1<TAB>director name2...
Editors<TAB>editor name1<TAB>editor name2...
Actor<TAB>actor name1<TAB>role name1
Actor<TAB>actor name2<TAB>role name2
Actress<TAB>actress name1<TAB>role name3
Actress<TAB>actress name2<TAB>role name4
<EMPTY LINE>
...
<table>
<thead>
<tr>
<th>Title</th>
<th>The Dark Knight Rises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2012</td>
</tr>
<tr>
<td>Running Time</td>
<td>165</td>
</tr>
<tr>
<td>MPAA Rating</td>
<td>PG-13</td>
</tr>
<tr>
<td>Genres</td>
<td>Action, Crime, Thriller</td>
</tr>
<tr>
<td>Key Words</td>
<td>action-hero, bat, billionaire, bomb, car-crash, catwoman</td>
</tr>
<tr>
<td>Producers</td>
<td>Nolan, Christopher (I), Roven, Charles, Thomas, Emma (I)</td>
</tr>
<tr>
<td>Directors</td>
<td>Nolan, Christopher (I)</td>
</tr>
<tr>
<td>Editors</td>
<td>Smith, Lee (II)</td>
</tr>
<tr>
<td>Actor</td>
<td>Bale, Christian, Bruce Wayne</td>
</tr>
<tr>
<td>Actor</td>
<td>Oldman, Gary, Commissioner Gordon</td>
</tr>
<tr>
<td>Actress</td>
<td>Hathaway, Anne, Selina</td>
</tr>
<tr>
<td>Actress</td>
<td>Cotillard, Marion, Miranda</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Life of Pi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>2012</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Data Characteristics

- 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.
Raw Dataset (2)

- **users.txt** (64,753,924 bytes)
  - Schema:
    `<email, name, password, age, gender, location>`
  - Format:
    All the fields are separated by TAB.
<table>
<thead>
<tr>
<th>Email</th>
<th>First Name</th>
<th>Last Name</th>
<th>Password</th>
<th>Age</th>
<th>Gender</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:oaguilar0@xyz.net">oaguilar0@xyz.net</a></td>
<td>ORA</td>
<td>AGUILAR</td>
<td>EC0S3fu0</td>
<td>83</td>
<td>Female</td>
<td>New Jersey</td>
</tr>
<tr>
<td><a href="mailto:acampbell11@xyz.net">acampbell11@xyz.net</a></td>
<td>ANDRE</td>
<td>CAMPBELL</td>
<td>4!pKoNra</td>
<td>34</td>
<td>Male</td>
<td>Maryland</td>
</tr>
<tr>
<td><a href="mailto:chaile2@xyz.net">chaile2@xyz.net</a></td>
<td>CYNTHIA</td>
<td>HAILE</td>
<td>OSS2xRNA</td>
<td>77</td>
<td>Female</td>
<td>West Virginia</td>
</tr>
<tr>
<td><a href="mailto:ejones3@xyz.net">ejones3@xyz.net</a></td>
<td>ELAINE</td>
<td>JONES</td>
<td>FgKS6FLX</td>
<td>61</td>
<td>Female</td>
<td>Washington</td>
</tr>
<tr>
<td><a href="mailto:lrasmussen4@xyz.net">lrasmussen4@xyz.net</a></td>
<td>LILLIE</td>
<td>RASMUSSEN</td>
<td>8gfe5CJE</td>
<td>25</td>
<td>Female</td>
<td>California</td>
</tr>
<tr>
<td><a href="mailto:ejones145597@xyz.net">ejones145597@xyz.net</a></td>
<td>ELAINE</td>
<td>JONES</td>
<td>d6FIU90K</td>
<td>57</td>
<td>Female</td>
<td>New York</td>
</tr>
</tbody>
</table>
Raw Dataset (3)

- **ratings.txt** (137,156,280 bytes)
  - Schema:
    <user email, movie title, movie year, rating>
  - Format:
    All the fields are separated by TAB.

- Interesting metrics
  - **Popularity**: number of ratings of a movie
  - **Goodness**: average rating of a movie
  - Most people want to know about popular and good movies!
<table>
<thead>
<tr>
<th>Email</th>
<th>Movie</th>
<th>Year</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:bfrost422115@xyz.net">bfrost422115@xyz.net</a></td>
<td>Life of Pi</td>
<td>2012</td>
<td>7</td>
</tr>
<tr>
<td><a href="mailto:alal1492627@xyz.net">alal1492627@xyz.net</a></td>
<td>Life of Pi</td>
<td>2012</td>
<td>9</td>
</tr>
<tr>
<td><a href="mailto:rcrain695898@xyz.net">rcrain695898@xyz.net</a></td>
<td>Life of Pi</td>
<td>2012</td>
<td>8</td>
</tr>
<tr>
<td><a href="mailto:pcarter688965@xyz.net">pcarter688965@xyz.net</a></td>
<td>The Dark Knight Rises</td>
<td>2012</td>
<td>9</td>
</tr>
<tr>
<td><a href="mailto:nnunnemaker901280@xyz.net">nnunnemaker901280@xyz.net</a></td>
<td>The Dark Knight Rises</td>
<td>2012</td>
<td>8</td>
</tr>
</tbody>
</table>
Raw Dataset (4)

- mpaa.txt (4,378 bytes)
  - Schema:
    <MPAA rating, definition, description>
  - Format:
    All the fields are separated by TAB.
PG     Parental Guidance Suggested. Some Material May Not Be Suitable For Children. A PG-rate
eir younger children attend. The PG rating indicates, in the view of the Rating Board, that pare
nts should make that decision. The more mature themes in some PG-rated motion pictures may call
of violence or brief nudity. But these elements are not deemed so intense as to require that par
There is no drug use content in a PG-rated motion picture.
etermine whether their children under age 13 should view the motion picture, as some material ma
PG rating in theme, violence, nudity, sensuality, language, adult activities or other elements, a
picture by itself will not result in a rating greater than PG-13, although depictions of activitie
 motion picture. Any drug use will initially require at least a PG-13 rating. More than brief n
rated motion picture generally will not be sexually oriented. There may be depictions of vi
persistent violence. A motion picture's single use of one of the harsher sexually-derived words, e
More than one such expletive requires an R rating, as must even one of those words used in a m
icture PG-13 if, based on a special vote by a two-thirds majority, the Raters feel that most Amer
of the context or manner in which the words are used or because the use of those words in the m
Datasets Available at

- Small sample files for ER modeling:
  
  Edlab: /usr/net/ftp/pub/cs445/project/sample_data/

- Complete data sets:
  
  Edlab: /usr/net/ftp/pub/cs445/project/data/
Step 2: Relational Model

- An entity set translated to a table
  - Set the primary key
  - Can introduce an ID to be the key; use ‘auto_increment’

- A relationship set translated to a table or embedded in an existing table
  - Attributes (Not Null?)
  - Primary key
  - Foreign keys

- Implement complex constraints using constraints or in PHP
  - Triggers: [MySQL vs PostgreSQL](http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL#Constraints), [create-trigger.html](http://dev.mysql.com/doc/refman/5.4/en/create-trigger.html), textbook Ch 5.8
  - PHP: program as you want

```sql
CREATE TABLE RelName
{ AttrName Type … }
```
Step 3: Parse & Load Data Set

- Raw data sets
  - Raw data format: see the data description above
  - Location on Edlab: /usr/net/ftp/pub/cs445/project/data/

- Project database in MySQL:
  - Database name: FirstLetterOfEachLastName, sorted in lexicographic order
  - SQL command: `mysql -h cs445sql -p -u $username $group_DB_name`

- Tasks for parsing and loading into your defined tables
  1. Parse data into the format defined in your schema
  2. Change permission of directory and files to be publicly accessible
  3. Load parsed data files into your tables in MySQL

```
LOAD DATA INFILE 'complete_path_of_data_file'
INTO TABLE $table_name FIELDS TERMINATED BY '\t';
```
Tips on Data Loading

- Prepare a table in a text file outside MySQL. **DO NOT:**
  - run MySQL queries to preprocess the data, or
  - insert a tuple at a time

- Where to place your temporary data files:
  - Under your 445 directory (not your home directory), which gives you enough space

- Make sure to make your data files *publicly accessible*
More about Data Loading…

- If you decide to use movie_id (or person_id)
  - Declare the field to be ‘auto_increment’;
  - Load movies into table M; movie ids are auto generated;
  - Load each other data file (e.g., Acting) first without the movie id field (say, into Table A1).
  - Join A1 with the movie table M; insert all result tuples into a new table (A2) to be the real Acting table;
  - Delete the old table (A1).
If Data Loading is Slow…

- By default, MySQL (InnoDB) builds a clustered index on the primary key and a secondary index on each foreign key
  - Using B+Tree insertion algorithm
  - We may have to run OPTIMIZE TABLE later
- Having enough memory to hold B+trees helps reduce random I/Os
- Turn off a few operations if data loading is still slow
  - set autocommit = 0
  - set unique_checks = 0
  - set foreign_key_checks = 0
If Data Loading Hangs…

- Background processes may hold locks of tables, and block new processes. Some useful commands:
  - Show background processes
    - “show processlist; ”
      (You can get process id, and status info of the processes)
  - Kill a process
    - “kill process_id; ”
      (Killing a long-running process may take long)

- Get some idea of the progress of loading
  - “show table status;”
    The Data_length field shows the current size of a table. It should be increasing during the loading process.
Project Update on Data Loading (Mar 10)

1. List of tables created, including attributes, primary keys, foreign keys, and other relevant constraints
   ▪ Can show the CREATE TABLE commands

2. Basic data characteristics of the tables, including the number of rows in each table, and the size of each table

   SHOW TABLE STATUS;

For details, see

Step 4: Test Queries & DB Tuning

- Test queries are released on the project web page
  - Selection
  - Join
  - Group by aggregation
  - Nested queries
- Test your database using the test queries.
  - If answer not correct, check schema & queries.
Query 1. Find all movies that have “Life” as a prefix in the title. (1,007 rows)

Query 2. Find all movies that “Pitt, Brad” acted in. (159 rows)

Query 3. Find the performers who acted in at least one movie directed by “Spielberg, Steven”. (2,665 rows)

Query 4. Show the number of user ratings for each rating value. (10 rows)

Query 5. Find the directors of all movies that are rated 10 by “DERRICK MYERS” whose age is 36. Show the name of each director. (4 rows)

Query 6. For each movie that is rated more than 5,000 times, show the title, year and the average age of the users who rate the movie. (17 rows)

Query 7. Find the users who are 17 or under 17 and rate at least one “NC-17” movie. Show the email, name, age and location of each user. (1,783 rows)
Query 8. Find the 5 best movies. The 5 best movies are the 5 movies with the highest average rating. Only the movies rated more than 1,000 times are considered. Show the title, year and average rating of each best movie. (5 rows)

Query 9. Find all the users of good taste. A user is of good taste if and only if the user rates at least 2 of the 5 best movies, and the user never rates the 5 best movies with rating less than 9. Show the email of each user. (238 rows)

Query 10. Now we consider only the ratings from the users of good taste. Find the 10 movies with the highest average rating. Only the movies rated more than 2 times by the users of good taste are considered. Show the title, year and average rating of each best movie. (10 rows)
Performance Tuning

- If performance is not good,
  - Add indexes:
    
    ```
    CREATE INDEX index_name ON table_name ( column_name(s) )
    ```
  - **Analyze table:** collect more precise statistics
  - **Rewrite queries:** eliminate nested subqueries
    - Use joins, but not subqueries, if possible
    - If needed, use derived tables in the From clause
  - **Store redundant information:**
    - Adding additional columns or tables (physically stored)
      - Make sure that redundant information is consistent with base data
    - Adding temporal tables within a connection
      
      ```
      CREATE TEMPORARY TABLE FROM SELECT ... 
      ```
  - **Revisit schema design:** normalization vs. denormalization
Report due on April 9

- Extend your report with
  - the set of test queries written in your schema,
  - the performance issues encountered,
  - the techniques employed to overcome performance issues,
  - the execution time of test queries using MySQL

Our performance target: <= 20 seconds
Step 5: Web Application Development

- PHP/MySQL tutorial
  

- PHP on Edlab
  

  - Group PHP directory:
    
    /courses/cs400/cs445/php-dirs/cs445_x_s15/www,
    
    where ‘x’ is the 2 or 3 letter group name, all in the upper case.

  - Upload your PHP files

  - Group URL:
    

  - Need username/password to access your group URL
    (Will be distributed via the course mailing list)
Implementing the Web Interface

- **Search interface**
  - Search by title, year, actor name, director name, etc.
- **User activities**
  - User login, rating of movies, update of avg. rating
- **Admin interface**
  - SQL queries are sent, including updates
  - Results are displayed with the response time

- More advanced features are welcome!
Implementing Your Extension

- Many ideas are possible:
  - Social networking among movie fans
  - Customized advertisements
  - Integrating with youtube, trailers...

- May need to generate new data, communication with the DB backend

- Start early…