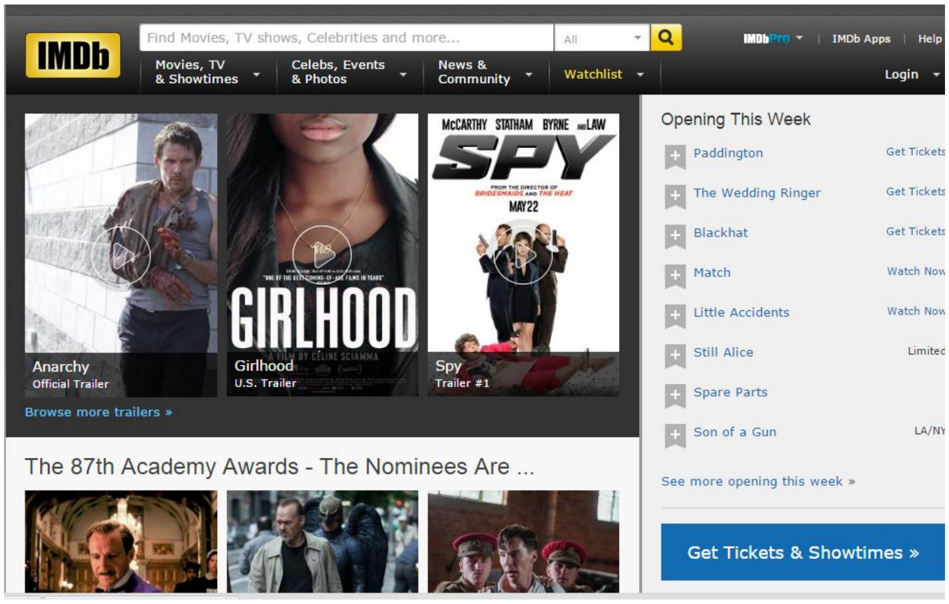
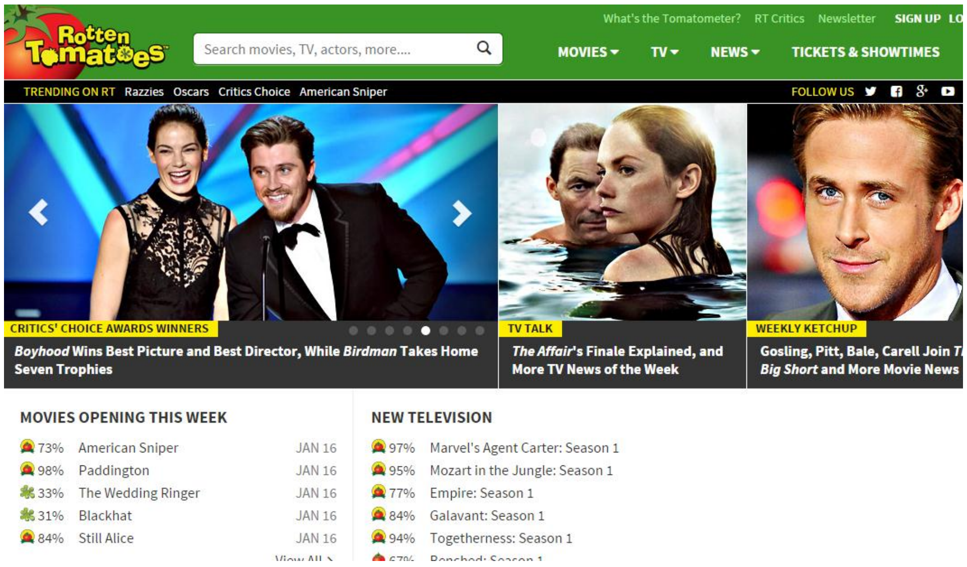
CS 4721-A Database Design I (Spring 2015)

The Programming Project

A Movie Recommender System based on IMDb and Rotten Tomatoes

Monday, April 27, 2015, at 11:59pm




Rotten Tomatoes
Search movies, TV, actors, more....
MOVIES
TV
NEWS
TICKETS & SHOWTIMES
TRENDING ON RT Razzies Oscars Critics Choice American Sniper
CRITICS' CHOICE AWARDS WINNERS
Boyhood Wins Best Picture and Best Director, While Birdman Takes Home Seven Trophies
TV TALK
The Affair's Finale Explained, and More TV News of the Week
WEEKLY KETCHUP
MOVIES OPENING THIS WEEK
73% American Sniper JAN 16
98% Paddington JAN 16
33% The Wedding Ringer JAN 16
31% Blackhat JAN 16
84% Still Alice JAN 16
NEW TELEVISION
97% Marvel's Agent Carter: Season 1
95% Mozart in the Jungle: Season 1
77% Empire: Season 1
84% Galavant: Season 1
94% Togetherness: Season 1

1. Project Description

In this project, you are required to design and implement a movie recommender system based on the real data obtained from

IMDb and Rotten Tomatoes,

which involves the following steps:

- determine the queries you want to implement;
- based on those queries, design a good database schema;
- create your database based on your schema;
- populate your database using the data I give you;
- implement the query module of your system;
- implement the front-end GUI of your system;
- and demo your system in the demo session.

In this project, you will learn how to use SQL, JDBC, and MySQL to implement a Java search engine driven by big data.

2. The Novelty of Our Recommender System

Our movie recommender system differs from most of the existing movie recommender systems in the following ways:

- Our system allows user to search movies by tags. Tags are user-generated metadata about movies. Each tag is typically a single word, or a short phrase. Tag can be considered as a subjective indication of how a user judges a movie.
- Our system provides a “See the Top Popular Movies” feature.
- Our system provides a “See the Top Popular Directors” feature.
- Our system provides a “See the Top Popular Actors” feature.

3. What is inside Our Big Data?

It contains all kind of information, including movie ratings and tags, about more than 10,000 movies. The readme file about the data is available in the zipped data package. Please understand the data first before you start to work on this project. To view big data file, please refer to my slides for how to use glogg.

Some statistics about our data:

- 2113 users
- 10197 movies

- 20 movie genres
- 20809 movie genre assignments
- avg. 2.040 genres per movie

- 4060 directors
- 95321 actors
- avg. 22.778 actors per movie

- 72 countries

- 10197 country assignments
- avg. 1.000 countries per movie

- 47899 location assignments
- avg. 5.350 locations per movie

- 13222 tags
- 47957 tag assignments (tas), i.e. tuples [user, tag, movie]
- avg. 22.696 tas per user
- avg. 8.117 tas per movie

- 855598 ratings
- avg. 404.921 ratings per user
- avg. 84.637 ratings per movie

4. The Supported Queries

Your system must support the following types of queries:

Query 1:

Provide a “See the Top Popular Movies” feature. Show a ranking of the top 20 movies with the highest Rotten Tomatoes audience scores without regard to their genres. Each movie should show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture and its IMDb cover picture.

Query 2:

For any movie title specified by user, show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture, its IMDb cover picture and all the user tags associated to that movie.

Query 3:

For any movie genre name specified by user, show the top 20 movies in that particular genre with the highest Rotten Tomatoes audience scores. Again, each movie should show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture and its IMDb cover picture.

Query 4:

For any director name specified by user, show all the movies directed by him or her. Again, each movie should show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture and its IMDb cover picture.

Query 5:

For any actor name specified by user, show all the movies which he or she appears in. Again, each movie should show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture and its IMDb cover picture.

Query 6:

For any tag name specified by user, show all the movies associated to that particular tag. Again, each movie should show its title, year, its Rotten Tomatoes audience score, its Rotten Tomatoes cover picture and its IMDb cover picture.

Query 7:

Provide a “See the Top Popular Directors” feature. Show a ranking of the top 20 directors with the highest average Rotten Tomatoes audience scores of all the movies he or she has directed.

Query 8:

Provide a “See the Top Popular Actors” feature. Show a ranking of the top 20 actor with the highest average Rotten Tomatoes audience scores of all the movies which he or she has appeared in.

Bonus points:

For any type of queries not mentioned above, based on your implementation, I will give you from 0 to 10 bonus points. Some possibilities include:

- Search movie by filming location
- Search movie by country
- Search all the ratings of movies given by a particular user
- Search all the ratings of a particular movie given by all users

5. Database Design and Implementation

Based on the queries you want to support in your system, design your database schema. Add primary keys and foreign keys to your schema appropriately. If you store everything in one huge table, you will get zero credit for the schema part. Once you finish your schema, create your database based on your schema.

6. Data Extraction

Based on the queries you want to support in your system, extract the data from our original data files to your database. Please refer to my slides on big data processing for how to parse data to Java programs from big data files. Because I give you big data,

extract only the data you need to populate your database!

7. System Implementation

Implement your system using Java+ JDBC + MySQL. Please connect your java applications to MySQL by using JDBC. Please refer to my slides on JDBC for help.

8. GUI Implementation

Give your system a user-friendly GUI for users to use your system.

9. Give a Great Name to Your Movie Recommender

Each group should include no more than five members. By **Monday March 9, 2015**, each team leader must send me (hachen@valdosta.edu) an email that describes the following:

1. Names of all the members in your group
2. The name of your movie recommender system

10. Materials to Hand In

1. All your Java source code for implementing your system. Include all your code in a zipped file named as "CS4721_Movie_Recommender.zip".

2. Schema.txt. This is the file which states the schema (showing all the primary keys and foreign keys) of your database.

3. SQL.txt. This file contains all the SQL statements you used to implement the 8 queries.

4. Bonus.txt (optional). This is the file which states the additional features in your system to claim bonus points. Based on your work, the instructor will give your group bonus points.

5. Members.txt. This is the file which states the names of all your teammates and the task division over all members within your group.

11. My Grading Breakdown

Database Schema	10 %
SQL Statements for 8 Queries	10 %
System Functionality and GUI	80 %

Please see the demo evaluation form for details.

12. Deadline

All the materials above should be submitted on BlazeView **individually** before

Monday, April 27, 2015, at 11:59pm.

NO late submissions will be accepted.

13. How to Demo Your System

Each group has 15 minutes to demo your system. Demo your system as follows:

1) Show your database schema.
2) Show your data using MySQL Workbench
3) Demo all the eight queries one by one. For each query, briefly talk about the SQL statement(s) in your code to implement that query.
4) The workload division over group members and what you have learned from this project

14. Help Sessions

Meetings for programming project

Monday, March 30, 2015	**12:00pm -12:50pm**
Monday, April 6, 2015	**12:00pm -12:50pm**

15. Demo Sessions

Monday, April 20, 2015	**12:00pm -12:50pm**
Wednesday, April 22, 2015	**12:00pm -12:50pm**
Friday, April 24, 2015	**12:00pm -12:50pm**