# Enhanced Movie Database: Reviews, Searching, Rating the Movies

Liu Yihan (1170100177), Cai Haoming (117010002), Huang Zihao (117010103), Chen Haoyu (117010013), Cao Yuji (117010007)

**Abstract**

In decades, people fuel the online informalized data with vast streams of review and rating on various movie community websites. Therefore, the existence of those abundant data can not be ignored by any film archives which engage in the research of movies' overview. However, as a well-known movie community website providing plenty of useful movie information, Douban movie database is not able to serve as an effective source for aggregating movie reviews and research due to its limited data search and analysis methods. To address these problems effectively, an enhanced database is proposed for reviewing, searching, analyzing and rating movies in this report. In this database, to facilitate data searching and analysis, lossless decomposition is necessary. Therefore, it is designed in second normalized form and third normalized form. In addition, it provides adequate SQL queries that are used for more comprehensive analysis and data mining about movies. By applying this design, the film archives are able to do more specific operations and examine the data more extensively for their research.

## 1 Background and Motivation

**Background** Movies are highly popular on the Web. There are several web resources dedicated to movies and many others containing movie-related information. For the two famous movie database website Douban and IMDb, the Internet Movie Database (IMDb) claims to be "the world's most popular and authoritative source for movie, TV and celebrity content," offering a searchable database which includes more than two million films, television, and entertainment programs and attracting more than 150 million unique monthly visitors (IMDb, 2013). At the same time,9 Douban Movie ranks No.1 in the iOS App Store in 2012 with about 200 million registered users as of 2013 [1]. With such a large number of users, these movie databases attract lots of attention to the movie analysis study and research, which engages in movie reviews and ratings. For instance, Jie Yang and Brian Yecies designed an innovative collection and analytic tool, termed Douban-Learning, which is a big-data framework for analyzing Douban movie reviews [2] and Sandra Carvalho designed an emotional movie database (EMDB) based on IMDB data [3].

**Motivation** However, most movie database websites are only designed for registered members' rating and writing reviews, while both the EMDB and Douban-Learning needs to preprocess first to get the origin movie data and then for analysis use. Besides, the Web-only has limited search methods for basic information and poor analysis methods for top-rated movies. For researchers, such data can not be used directly and efficiently, and the scattered movie data still need to be aggregated for use. Hence, we designed an enhanced movie database for reviewing, searching, analyzing, and rating movies, especially for researchers.

# 2 Method

The report contains four phases:

- Movie data pre-processing: web crawler is applied in this part to get the origin movie data from the official web and Douban movie, and more than 90000 movies webpage are fetched.
- Database Design: design the backend database structure of the enhanced movie database, containing E-R diagram design and schema design which conforms to BCNF.
- Database implementation: process the data and split them into several tables for import to the database directly. Besides, this part also contains the website visualization and database querying design under the website.
- Analysis: the preliminary analysis based on the website visualization and the movie data, including the annual trend of rating, box office, counting, etc.

# 3 Data Collection

## 3.1 Web Crawler

The web crawler is a method that automatically simulates a human's browse using programs. No matter what kind of data it is, as long as it exists on a webpage, it can be fetched by the crawler program. Therefore, web crawlers are suitable for fetching massive and open-source information on a website in a short time.

## 3.2 Inconstant IP

Their tools contain checking IP addresses, browser's user-agent, request headers, etc. Among them, blocking IP addresses is the most common and efficient way to obstruct vicious requests. Douban uses this strategy, as well. Therefore, how to overpass this protection is one of the core parts of our program.

We could decrease the frequency of access to avoid websites to block our IP, but it would waste too much time. It seems that using an inconstant IP address is the only method we could choose. In our project, we bought available IP addresses from a proxy. With the help of inconstant IP addresses, we could fetch lots of data without being clocked. We use more than 40 thousand IP addresses to obtain almost 300 thousand items, such as actors, awards, and movies.


Figure 1: Raw data

## 3.3 Data Fetching

Libraries used to fetch web elements are various. Here we use an elegant library named Beautiful Soup. Calling 'find' in Beautiful Soup, the precise search could be implemented. Those massive raw files are required to further process to form the initial version of CSV files. We show this process in Figure 1 and Figure 2.

| 中文姓名 | 英文姓名 | 性别 | 出生国家 | 出生省份 | 出生城市 | 星座 | 职业 | 更多外文名 | imdb编号 | 官方网站 | 现任配偶 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Null | Henry Hall | Null | 英国 | 伦敦 | 佩卡姆 | 金牛座 | 音乐 | Henry Robe | nm035564 | Null | Null |
| Null | Mark Freibu | Null | 美国 | 弗吉尼亚州 | 罗阿诺克 | 巨蟹座 | 制片人 / 导 | Stephen Ma | nm133043 | Null | Null |
| Null | Sammy Gle | Null | 英国 | Null | 伦敦 | 狮子座 | 演员 | Sammy Gle | nm032265 | http://www. | Null |
| Null | Heidi Heral | Null | 芬兰 | Null | 赫尔辛基 | 摩羯座 | 演员 | Heidi Helka | nm037842 | Null | Null |
| Null | Vida Hope | 女 | 英国 | Null | 利物浦 | 射手座 | 演员 | Null | nm039405 | Null | Null |
| Null | John Grillo | Null | 英国 | 赫特福德郡 | 沃特福德 | 射手座 | 演员 / 编剧 | Null | nm034203 | Null | Null |
| Null | Don Harris | Null | 美国 | 田纳西州 | 纳什维尔 | 天秤座 | 演员 | Null | nm036463 | Null | Null |
| 斯罗斯特·莱 | Þröstur Leó | 男 | Null | Null | Null | 金牛座 | 演员 | Thröstur Le | nm034827 | Null | Null |
| Null | David Hans | Null | 美国 | 加利福尼亚 | 圣莫尼卡 | 金牛座 | 演员 | Null | nm036123 | Null | Null |
| Null | Gidget Geir | Null | 美国 | 佛罗里达州 | 好莱坞 | 处女座 | 演员 | Brad Stewa | nm031198 | http://www. | Null |
| 马雷莎·加洛 | Maresa Ga | 女 | 意大利 | Null | 罗马 | 双子座 | 演员 / 副导 | Maresa Ma | nm030306 | Null | Null |
| Null | Beatriz Gal | Null | 西班牙 | Null | Null | 摩羯座 | 演员 | Beatriz Gal | nm030178 | Null | Null |
| Null | Julia Hede- | Null | 瑞典 | Null | 斯德哥尔摩 | 水瓶座 | 演员 / 摄影 | Julia Hede | nm037317 | Null | Null |
| Null | Daniela Ho | Null | 德国 | Null | 柏林 | 白羊座 | 演员 | Null | nm038915 | Null | Null |
| Null | Lenn Hjortz | Null | 瑞典 | 布莱金厄兰 | 卡尔斯克鲁 | 天秤座 | 副导演 / 演 | Null | nm038713 | Null | Null |
| Null | Maribel Gu | Null | 哥斯达黎加 | Null | 圣何塞 | 双子座 | 演员 | Maribel Fer | nm034532 | Null | Null |
| Null | Axel Hildeb | Null | 德国 | Null | 柏林 | 射手座 | 编剧 / 演员 | Axel Hildeb | nm038383 | Null | Null |
| Null | Cee Cee M | Null | 美国 | 北卡罗来纳 | 山核桃 | 金牛座 | 演员 | Cecelia P. H | nm036609 | http://www. | Null |
| Null | Freddy Sot | Null | 美国 | 得克萨斯州 | 埃尔帕索 | 巨蟹座 | 演员 / 编剧 | Alfred Soto | nm110025 | Null | Null |
| 奈杰尔·哈曼 | Nigel Harm | 男 | 英国 | 萨里 | 珀利 | 狮子座 | 演员 | Null | nm036342 | Null | Null |
| 朱莉·加菲尔 | Julie Garfie | 女 | 美国 | 加利福尼亚 | 洛杉矶 | 摩羯座 | 演员 | Julie Rober | nm030727 | Null | Null |
| Null | Jesús Hern | Null | 西班牙 | 安达卢西亚 | 韦尔瓦 | 巨蟹座 | 导演 / 演员 | Jesús Hern | nm037940 | Null | Null |
| 蒂姆·哈达威 | Tim Hardav | 男 | 美国 | 伊利诺伊州 | 芝加哥 | 处女座 | Null | Timothy Du | nm036204 | Null | Null |
| 朱塞佩·巴蒂 | Giuseppe B | 男 | 意大利 | Null | 乌迪内 | 巨蟹座 | 演员 / 配音 | Beppe Batt | nm006148 | Null | Null |

Figure 2: Initial version of processed data

# 4 Database Design

After we have all the data we need, we can start to design our database. We need to find a way that can store all the useful data in a structural way. The design needs two major steps. The first step is the E-R diagram design and the latter one is schema design.

## 4.1 E-R Diagram Design

As mentioned before, we got the data from different sources. In other words, they don't have a uniform data structure of a consistent data label. So, we decided to use a data-oriented way to design our database. Based on the data we had, we abstracted all the entities and attributes. We have several versions of the E-R diagram. The one shown in Figure 3 below is our final version.

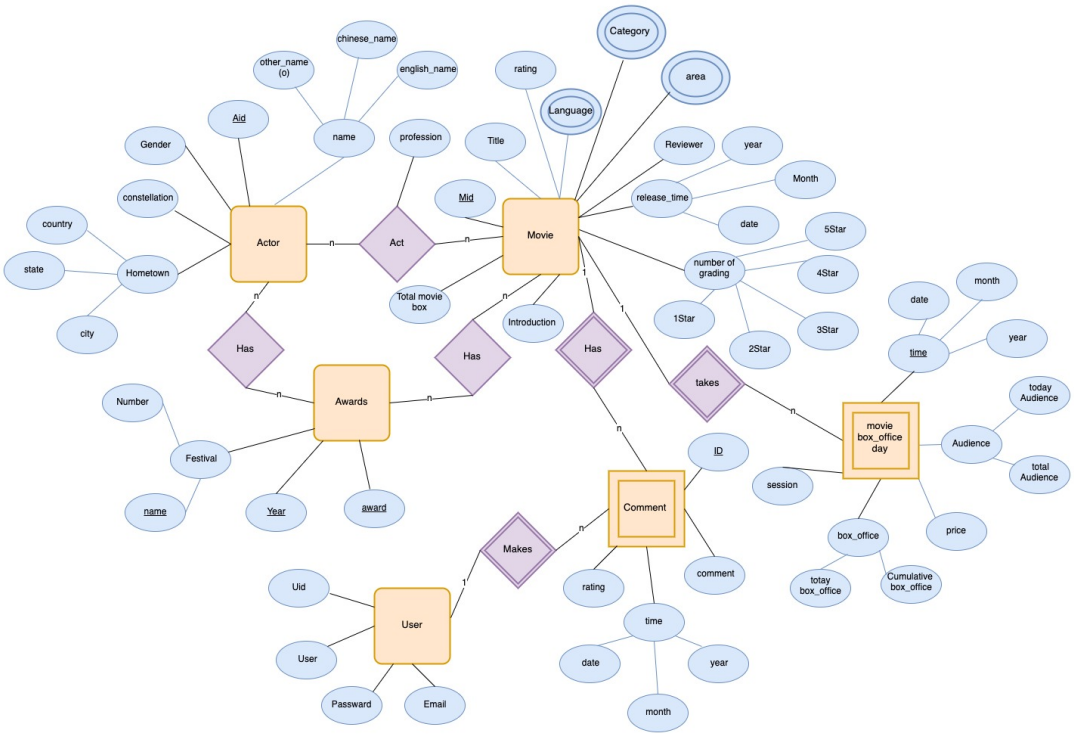



Figure 3: E-R diagram

There are six basic entities in our design. They are Movie, User, Comment, the Movie Box Office of each day, Awards and Actor. In these six entities, comment and movie box office of each day are weak entities. The comment depends on the user and a specific movie, and the movie box office of each day depends on a specific movie. Each of the six entities has some relationships connected with others. The relationships are described below:

- An actor can win several awards and an award can give to many actors that's an N to N relationship.
- A movie can also earn many awards and an award can also be given to many movies, which is also an N to N relationship.

- An actor can perform in many movies and a movie can also have many actors.
- A movie can have multiple Movie Box Offices of different days but a specific movie box office of one day only refers to a specific movie.
- Each movie can have multiple comments, but a specific comment can only be made to a specific movie.
- Each user can make several comments, but a specific comment can only be owned by a specific user.

For each of the six entities, it has its own attributes which are listed in the diagram.

## 4.2 Schema Design

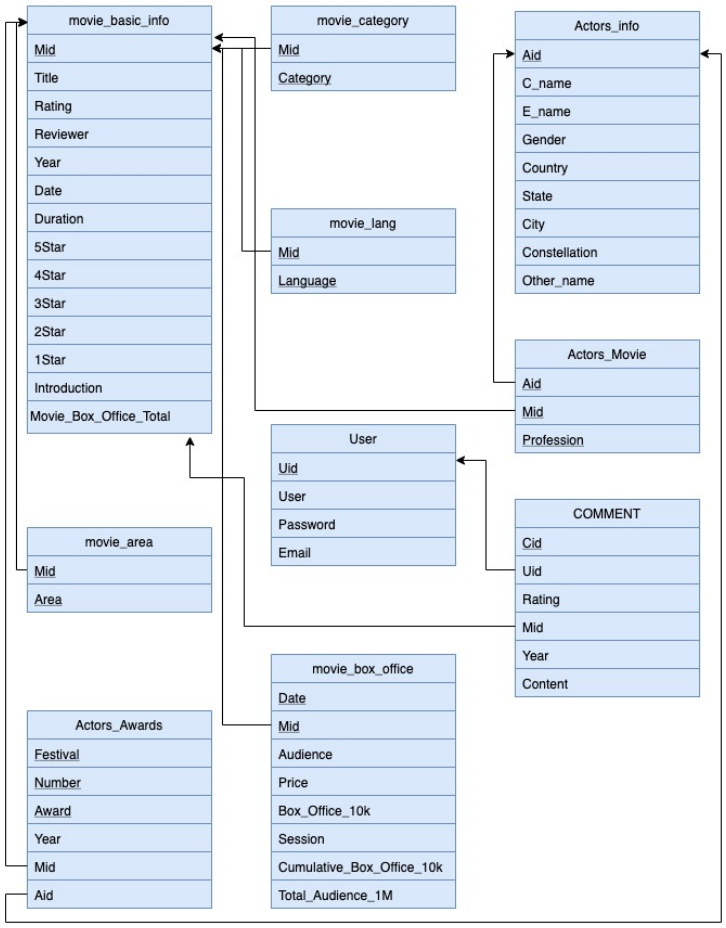



Figure 4: Schema Of the Database

Based on the E-R diagram shown above, our schema design is created. The Figure 4 shown below is our last version.

The last version of the schema satisfies the BCNF. To design the schema, we first make each entity in the E-R diagram to be a table, then decompose them into smaller tables, until each of them satisfies the BCNF. There is something special in the design: the table movie area, movie category and movie language only have two attributes that is because they are multivalued attributes of the movie entity. They have an N to N relationship to movies. To reduce the information redundancy, we make these 3 attributes to have their table. The FDs for each of the table are list below:

- movie_basic_info:

  $Mid \rightarrow Title, Rating, Reviewer, Year, Date, Duration, 5Star, 4Star, 3Star, 2Star, 1Star,$
  $introduction, Movie\_Box\_Office\_Total$

- Actor_Awards

  $Festival, Number, Award \rightarrow Year, Mid, Aid$
  $Festival, Year, Award \rightarrow Number, Mid, Aid$

- User:

  $Uid \rightarrow User, Password, Email$

- movie_box_office:

  $Date, Mid \rightarrow Audiences, Price, Box\_Office\_10k, Session,$
  $Cumulative\_Box\_Office\_10k, Total\_Audience\_1M$

- Actor_info:

  $Aid \rightarrow C\_name, E\_name, Gender, Country, State, City, Constellation, Other\_name$

- COMMENT:

  $Cid \rightarrow Uid, Rating, Mid, Year, Content$

# 5 Database Implementation

## 5.1 Data Processing

As mentioned above, we gather the data from Douban Website and other professional websites using web crawlers. It consists of movies' information, reviews' information, actors' information, and information about movie awards like Oscar. Moreover, the data could be dirty because of inaccurate, incomplete, or inconsistent input. After data collection, despite reallocating according to the schema of our database for the convenience of importing data to our database, we still need data cleansing to rewrite the data easily.

We implemented data cleaning first, which detected and removed corrupt or inaccurate records from the database and then modified the dirty data. For instance, the unit of the

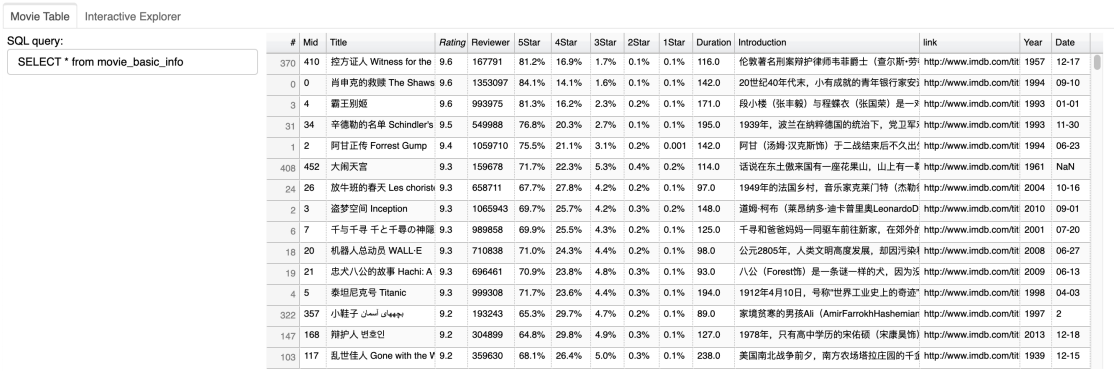
movie box did not conform to any format in MySQL. Therefore, we invoked eval function in python to replace the origin string format into an integer. After data cleaning, since the data were fetched from multiple datasets, we implemented data matching and data reorganization to combine data from different data sets and fulfill our requirements, which could be considered as a lossless decomposition of the original data with the Mid(Movie ID) attribute as the super key. As for linking the same movie from different data sets, we performed record matching based on their titles and the corresponding release years.

## 5.2 Web Visualization

In order to provide a better way to demonstrate our database, a web-based front end application is implemented using Bokeh and Sqlite3 Library in Python. Bokeh is an interactive visualization library and Sqlite3 is a C-language library that implements a SQL database engine. In the application, the Bokeh library helps to demonstrate and visualize the data using various methods like data tables and diagrams. The Sqlite3 library contributes to process SQL queries in the database engine and extracts corresponding data. The commands to run this application is listed in Appendices Listing 3. To demonstrate our database, we implement this front end application, which consists of two Tabs: data table and iterative explorer diagram.

### 5.2.1 Data Table Tab

Data Table Tab extracts the data from the Movie database and demonstrates it according to the SQL query in the textInput widget. When a user inputs a SQL query and presses ENTER key, the SQL query is passed to the Bokeh server and being executed by the Sqlite3 engine to select the data. Then the data is returned back to the Bokeh serve, rendered by the Bokeh server and displayed on the web browser.




| # | Mid | Title | Rating | Reviewer | 5Star | 4Star | 3Star | 2Star | 1Star | Duration | Introduction | link | Year | Date |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 370 | 410 | 控方证人 Witness for the | 9.6 | 167791 | 81.2% | 16.9% | 1.7% | 0.1% | 0.1% | 116.0 | 伦敦著名刑案辩护律师韦菲爵士（查尔斯·劳 | http://www.imdb.com/tit | 1957 | 12-17 |
| 0 | 0 | 肖申克的救赎 The Shaws | 9.6 | 1353097 | 84.1% | 14.1% | 1.6% | 0.1% | 0.1% | 142.0 | 20世纪40年代末，小有成就的青年银行家安 | http://www.imdb.com/tit | 1994 | 09-10 |
| 3 | 4 | 霸王别姬 | 9.6 | 993975 | 81.3% | 16.2% | 2.3% | 0.2% | 0.1% | 171.0 | 段小楼（张丰毅）与程蝶衣（张国荣）是一对 | http://www.imdb.com/tit | 1993 | 01-01 |
| 31 | 34 | 辛德勒的名单 Schindler's | 9.5 | 549988 | 76.8% | 20.3% | 2.7% | 0.1% | 0.1% | 195.0 | 1939年，波兰在纳粹德国的统治下，党卫军 | http://www.imdb.com/tit | 1993 | 11-30 |
| 1 | 2 | 阿甘正传 Forrest Gump | 9.4 | 1059710 | 75.5% | 21.1% | 3.1% | 0.2% | 0.001 | 142.0 | 阿甘（汤姆·汉克斯饰）于二战结束后不久出 | http://www.imdb.com/tit | 1994 | 06-23 |
| 408 | 452 | 大闹天宫 | 9.3 | 159678 | 71.7% | 22.3% | 5.3% | 0.4% | 0.2% | 114.0 | 话说在东土傲来国有一座花果山，山上有一 | http://www.imdb.com/tit | 1961 | NaN |
| 24 | 26 | 放牛班的春天 Les choristes | 9.3 | 658711 | 67.7% | 27.8% | 4.2% | 0.2% | 0.1% | 97.0 | 1949年的法国乡村，音乐家克莱门特（杰勒 | http://www.imdb.com/tit | 2004 | 10-16 |
| 2 | 3 | 盗梦空间 Inception | 9.3 | 1065943 | 69.7% | 25.7% | 4.2% | 0.3% | 0.2% | 148.0 | 道姆·柯布（莱昂纳多·迪卡普里奥LeonardoD | http://www.imdb.com/tit | 2010 | 09-01 |
| 6 | 7 | 千与千寻 千と千尋の神隠 | 9.3 | 989858 | 69.9% | 25.5% | 4.3% | 0.2% | 0.1% | 125.0 | 千寻和爸爸妈妈一同驱车前往新家，在郊外发 | http://www.imdb.com/tit | 2001 | 07-20 |
| 18 | 20 | 机器人总动员 WALL·E | 9.3 | 710838 | 71.0% | 24.3% | 4.4% | 0.2% | 0.1% | 98.0 | 公元2805年，人类文明高度发展，却因污染 | http://www.imdb.com/tit | 2008 | 06-27 |
| 19 | 21 | 忠犬八公的故事 Hachi: A | 9.3 | 696461 | 70.9% | 23.8% | 4.8% | 0.3% | 0.1% | 93.0 | 八公（Forest饰）是一条谜一样的犬，因为泛 | http://www.imdb.com/tit | 2009 | 06-13 |
| 4 | 5 | 泰坦尼克号 Titanic | 9.3 | 999308 | 71.7% | 23.6% | 4.4% | 0.3% | 0.1% | 194.0 | 1912年4月10日，号称"世界工业史上的奇迹" | http://www.imdb.com/tit | 1998 | 04-03 |
| 322 | 357 | 小鞋子 بچه‌های آسمان | 9.2 | 193243 | 65.3% | 29.7% | 4.7% | 0.2% | 0.1% | 89.0 | 家境贫寒的男孩Ali（AmirFarrokhHashemian | http://www.imdb.com/tit | 1997 | 2 |
| 147 | 168 | 辩护人 변호인 | 9.2 | 304899 | 64.8% | 29.8% | 4.9% | 0.3% | 0.1% | 127.0 | 1978年，只有高中学历的宋佑硕（宋康昊饰） | http://www.imdb.com/tit | 2013 | 12-18 |
| 103 | 117 | 乱世佳人 Gone with the W | 9.2 | 359630 | 68.1% | 26.4% | 5.0% | 0.3% | 0.1% | 238.0 | 美国南北战争前夕，南方农场塔拉庄园的千金 | http://www.imdb.com/tit | 1939 | 12-15 |

Figure 5: Data table tab in web visualization

### 5.2.2 Iterative Explorer Diagram

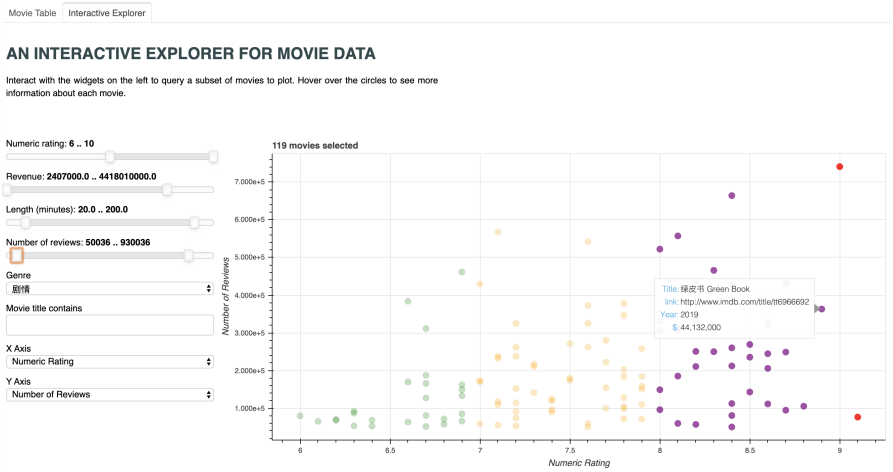



Figure 6: Iterative explorer tab in web visualization

Iterative Explorer Diagram demonstrates the movies with a circle diagram where each movie is represented by a circle in the diagram and the darker the color, and higher rating the movie is. By selecting various criteria in the left column, the Bokeh will automatically compose these different constraints and instruct the Sqlite3 database to extract the corresponding data and visualize them on the diagram. In addition, a user is able to select different attributes on the X and Y axis to explore the different relationships between rating, length of movies, revenue and so forth.

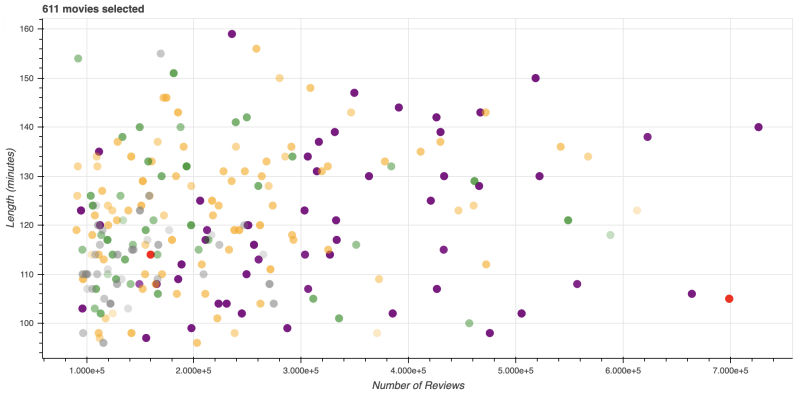



Figure 7: Diagram of the review number and length of movie

# 6 Data Analysis

Here we firstly create some views to simplify the subsequent querying. These views temporarily store matches which we find between movies and their related entities. Listing 1 shows the creation of this view.

With those views, we could dig deep to find tendencies or some correlations in movie fields in decades. Figure 8 shows the rank of a ratio where ratio equals to

$$\frac{\text{number of actors whose average rating exceeds 7.25 in one specific country}}{\text{number of actors in this country}}$$

This rank of ratio reveals the different countries' movies' influence. The queries is shown in Listing 2. In this ranking, we could find those countries which are not abundant in the movie industry obtain relatively high rankings because only their famous actors and movies are recorded in open source databases. The reason China gets a relatively low ranking is that China is a brand new movie market. Considering that quality always fails to catch up with quantity, this ranking is quite reasonable and indicates that there is still a long distance to develop for Chinese movies.

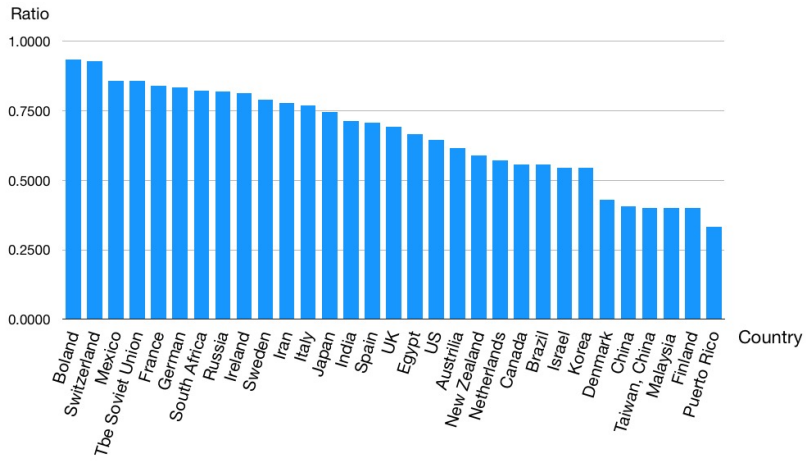



Figure 8: Rank of country about ratio of actors/actresses whose average rating grades exceed 7.25

The Figure 9 could confirm the booming movie market. After counting the annual number and rating of movies, the tendency of the movie industry is evident. The barriers are falling for small firms. The amount of film increases rapidly, but the quality decreases step by step.

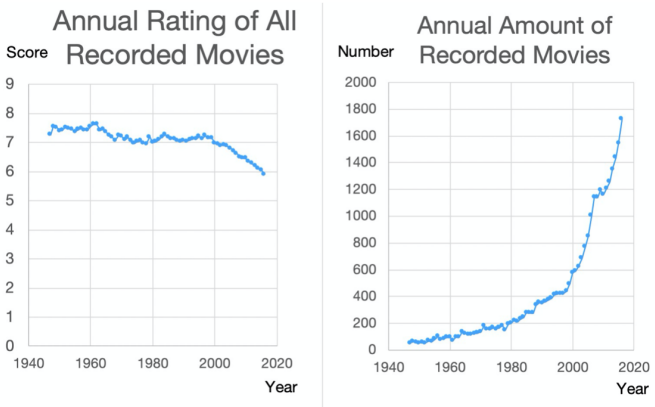



Figure 9: Annual rating of all recorded movies and annual amount of recorded movies

# 7 Self-evaluation

## 7.1 What did we learn from the experience?

Since we realized the entire process of building a database system, we have learned and practiced many related objects, including:

- Analyze the requirements in a realistic situation.
- Identify and improve a database system design in Boyce-Codd normal form.
- Produce E-R diagram based on our database design.
- Data collection with web crawlers.
- Produce SQL queries for practical operations.
- Data analysis based on our database.
- Implement a visual website to demonstrate the database.

## 7.2 What difficulties have we solved?

In this process, we also encountered some difficulties. Since the data is from several sources, therefore, part of the movies has been assigned different IDs in the different datasets. We need to match and link the same movie from the different datasets based on their attributes. Justifying our database designs are in good normalized form is also a challenge. To optimize our design, we make full use of the knowledge learned in the course and achieve a good result.

# 8 Conclusion and Future Work

In conclusion, we build an Enhanced Movie Database as an valuable source for aggregating movie reviews, searching, rating, and research. There is a large amount of realistic data in our database, which all come from several professional movie data websites, such as Douban and China Movie Data Information Network. The database is in a high-quality design, which is in Boyce-Codd normal form. We also provide adequate SQL queries for specific searching and selection. We show the value of our database by these queries that would have required exploring multiple web pages to answer. Through our database system, the researchers become much simpler to implement more comprehensive analysis and data mining.

For future work, there are other unused data, such as movie release type, producer company, and other multi-value attributes. Adding new entities and relationships can help enrich our database. Besides, because the size of our database is large, we will continue to work on improving the IO efficiency of our system for better user experience. On the other hand, although we provide some cases above, we are confident that researchers could utilize our project for more widely and more deeply application of meaningful machine learning and social analytics. For instance, whether rating and box office of a movie is related to the number of actresses in that movie. We can also cluster all kinds of attributes of movies to find the common characteristics of high-quality/low-quality movies. These both can be engaging research topics. Therefore, using our database to conduct a profound study on movies may be an interesting way to explore.

# References

[1] Alexa, "douban.com Competitive Analysis, Marketing Mix and Traffic."

[2] J. Yang and B. Yecies, "Mining chinese social media ugc: a big-data framework for analyzing douban movie reviews," *Journal of Big Data*, vol. 3, no. 1, p. 3, 2016.

[3] S. Carvalho, J. Leite, S. Galdo-Álvarez, and Ó. F. Gonçalves, "The emotional movie database (emdb): A self-report and psychophysiological study," *Applied psychophysiology and biofeedback*, vol. 37, no. 4, pp. 279–294, 2012.

# Appendices

## SQL Queries

```
-- Movie, Actor, Rating
CREATE VIEW movie_actor_rating AS
SELECT movie_basic_info.Mid, title, rating, Aid, profession
FROM movie_basic_info, Actors_Movie
WHERE movie_basic_info.Mid = Actors_Movie.Mid
```

Listing 1: Pre-created Views Of Quick Querying

```
CREATE VIEW country_725actor AS
SELECT COUNT(Actors_info.Aid), Country
FROM movie_actor_rating, Actors_info
WHERE movie_actor_rating.Aid = Actors_info.Aid AND profession = "
    Actor"
AND rating > 7.25
GROUP BY Country;

CREATE VIEW country_0actor AS
SELECT COUNT(Actors_info.Aid), Country
FROM movie_actor_rating, Actors_info
WHERE movie_actor_rating.Aid = Actors_info.Aid AND profession = "
    Actor"
GROUP BY E_name, C_name, Country;

SELECT number725_Actors/number0_Actors,country_725actor.country
FROM country_725actor, country_0actor
WHERE country_0actor.country = country_725actor.country
```

Listing 2: SQL Queries of Figure 8

## Labor Distribution

- Liu Yihan (117010007): Web crawler & Design and normalization of E-R diagram and schema
- Cai Haoming (117010002): Web crawler & Raw data preprocessing & Data mining and analysis & PPT Creating
- Huang Zihao (117010103): Raw data preprocessing & Primary design and normalization of E-R Diagram and schema
- Chen Haoyu (117010013): Participation in the design of E-R diagram and normalization & Data processing
- Cao Yuji (117010007): Web visualization implementation & Primary design and normalization of E-R diagram & Database building & Report formatting

## Source Code

Our project structure is demonstrated in Figure 10 as a tree diagram. "database.db" is our database with Sqlite database extension. "src" is the folder containing our source codes. Commands to run the web visualization application is listed in Listing 3 and commands to view our database is listed in Listing 4.

```
#!/bin/bash
pwd # ensure your current directory is in our project
pip install Bokeh # install Bokeh Python Library if there isn't
cd src/Web_Visualization
bokeh serve --show . # run the application
```

Listing 3: Commands to run web visualization application

```
#!/bin/bash
pwd # ensure your current directory is in our project
sqlite3 database/movies.db
```

Listing 4: Commands to view the database in terminal

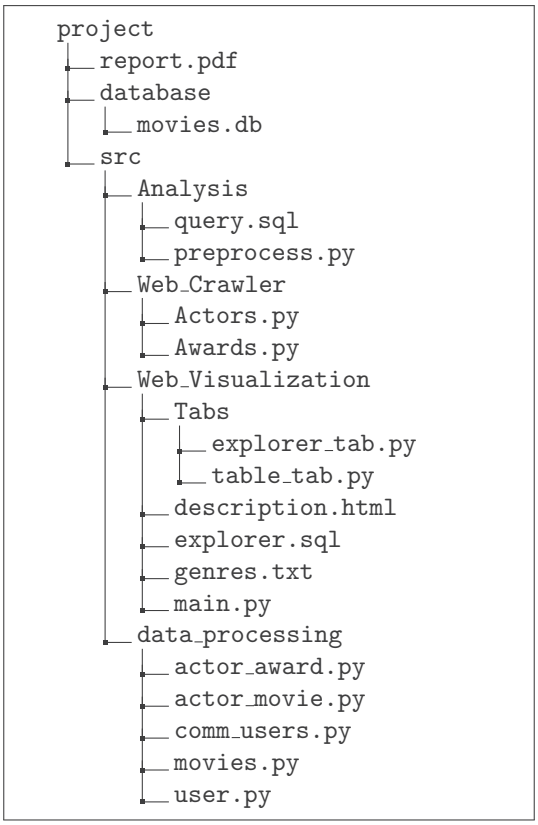



Figure 10: Project Structure