Università di Pisa

Large-Scale adn Multi-structured Databases -
Project: Rotten Movies

Fabio Piras, Giacomo Volpi, Guillaume Quint

Contents

Chapter 1

Introduction

Rotten Movies is a web service where user can keep track of the general liking for movies, they can also share their thoughts with the world after signing up. In addition user can follow renowned top critic to be constantly updated with their latest review.

Guest user, as well as all other, can browse or search movies based on filters like: title, year of release and actor who worked in it. They can also view a Hall of Fame for the most positive review genres, production houses and years in which the best movie were released.

For this project, the application has been developed in Java Spring using Intellij to implement a web GUI. The application use a document database to store the main information about the movies, user, review and actor; a graph database is instead used to keep track of the relationship between normal user and top critic in terms of who follows who and between the movie and the user who reviewed it.

Chapter 2

Feasibility Study

The very first step to perform is to look up for what type of data we need to create the application and how to handle it by performing a feasibility study.

2.1 Dataset analysis and creation

Initially we search online for available dataset, we ended up with five different files coming from Kaggle and imdb with a combined storage of 4.18 Gb:

- title_principal.tsv
- name_basic.tsv
- title_basic.tsv
- title_crew.tsv
- rotten_movies.csv
- rotten_reviews.csv

The first three were found on the imdb site containing general data about the title of the entries, type (not all were movies), cast, crew and other useful information. The last two came from Kaggle and they contain data scraped from the rotten tomatoes site regarding the movies rating and their reviews. All of these dataset were organized in a relational manner.

The next step was to delete all useless information and perform a join operation to generate one single final dataset for the document database. These operations were performed through a python script with the help of the pandas library and Google Colab, the end result was a single json file of 270 Mb. The structure of the document is the following:

```
1 {
2     "_id" : ObjectId(<<id_field>>),
```

```
    "primaryTitle": "The first ever movie",
    "year": 1989,
    "runtimeMinutes": 70,
    "genres": ["Crime", "Drama", "Romantic"],
    "productionCompany": "Dingo Picture Production" ,
    "personnel": [
        {
            "name": "John Doe",
            "category": "producer",
            "job": "writer"
        },
        {
            "name": "Christopher Lee",
            "category": "actor",
            "character": ["The one"]
        },
        ...
    ],
    "top_critic_fresh_count": "4",
    "top_critic_rotten_count": 0,
    "user_fresh_count": 14,
    "user_rotten_count": 1,
    "top_critic_rating": 100,
    "top_critic_status": "Fresh",
    "user_rating": 93,
    "user_status": "Fresh",
    "review":
    [
        {
            "critic_name":      "AntonyE",
            "review_date":      "2018-12-07",
            "review_type":      "Rotten",
            "top_critic":       false,
            "review_content":"I really didn't liked it!",
            "review_score":     "1/10"
        },
        {
            "critic_name":      "AntonyE",
            "review_date":      "2015-12-07",
            "review_type":      "Fresh",
            "top_critic":       true,
            "review_content":"I really liked it!",
            "review_score":     "A-"
        },
        ...
    ]
}
```