基于 MySQL 的影视作品信息检索平台

数据库系统及应用实践课程项目

TRIGGER(devil-crash (trigger) · GitHub)

BTDWZLLC(GitHub Dashboard)

目录

1. 简介

此项目是基于 MySQL 的影视作品信息检索平台，采用前后端分离架构，可实现远程

问，使用者可根据需要检索影视作品及相关从业人员信息，后期也可通过拓展数据库字段，以超链接形式访问外部资源或在本地存储，实现为简易的影视作品检索平台。

2. 总体设计

2.1 系统功能

使用者可根据需要检索影视作品及相关从业人员信息，后期也可通过拓展数据库字段，以超链接形式访问外部资源或在本地存储，实现为简易的影视作品检索平台。

检索功能包括模糊搜索以及标签匹配。

2.2 系统软硬件平台

2.2.1 系统开发平台（含开源/第三方工具）

后端开发平台为 idea（maven），采用 springboot 架构，相关依赖有 spring-boot-web-starter，mysql-connector-j 等，详情见附录。

前端 PyCharm（tkinter，simplejson，urllib）。

2.2.2 系统运行平台

采用 web 技术，平台依赖低。客户端无依赖；服务端中后端使用

java，任意平台均可部署，数据库端使用 mysql，部署对应平台版本即可。

2.3 关键技术

Springboot，前后端分离架构，RESTful 风格。

3. 数据库设计

3.1 数据处理

3.1.1 原始数据来源

Imdb 非商业数据集（IMDb Non-Commercial Datasets）结构如下

title.akas.tsv.gz

- titleId (string) - a tconst, an alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- title (string) – the localized title
- region (string) - the region for this version of the title
- language (string) - the language of the title
- types (array) - Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay". New values may be added in the future without warning
- attributes (array) - Additional terms to describe this alternative title, not enumerated
- isOriginalTitle (boolean) – 0: not original title; 1: original title

title.basics.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) - original title, in the original language
- isAdult (boolean) - 0: non-adult title; 1: adult title
- startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) – TV Series end year. '\N' for all other title types
- runtimeMinutes – primary runtime of the title, in minutes
- genres (string array) – includes up to three genres associated with the title

title.crew.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- directors (array of nconsts) - director(s) of the given title
- writers (array of nconsts) – writer(s) of the given title

title.episode.tsv.gz

- tconst (string) - alphanumeric identifier of episode
- parentTconst (string) - alphanumeric identifier of the parent TV Series
- seasonNumber (integer) – season number the episode belongs to
- episodeNumber (integer) – episode number of the tconst in the TV series

title.principals.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- nconst (string) - alphanumeric unique identifier of the name/person
- category (string) - the category of job that person was in
- job (string) - the specific job title if applicable, else '\N'
- characters (string) - the name of the character played if applicable, else '\N'

title.ratings.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- averageRating – weighted average of all the individual user ratings
- numVotes - number of votes the title has received

name.basics.tsv.gz

- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)– name by which the person is most often credited
- birthYear – in YYYY format
- deathYear – in YYYY format if applicable, else '\N'
- primaryProfession (array of strings)– the top-3 professions of the person
- knownForTitles (array of tconsts) – titles the person is known for

3.1.2 数据拆分与合并

数据表 title.basics.tsv 中 genres 为多值字段，且会在项目中作为检索标签，与作品唯一标识 tconst 为多对一关系，将其单独拆分（携带 tconst）。

title.crew.tsv 中 directors 和 writers 均为多值字段且与 tconst 为多对一关系，将其分别携带 tconst 拆分为两表。

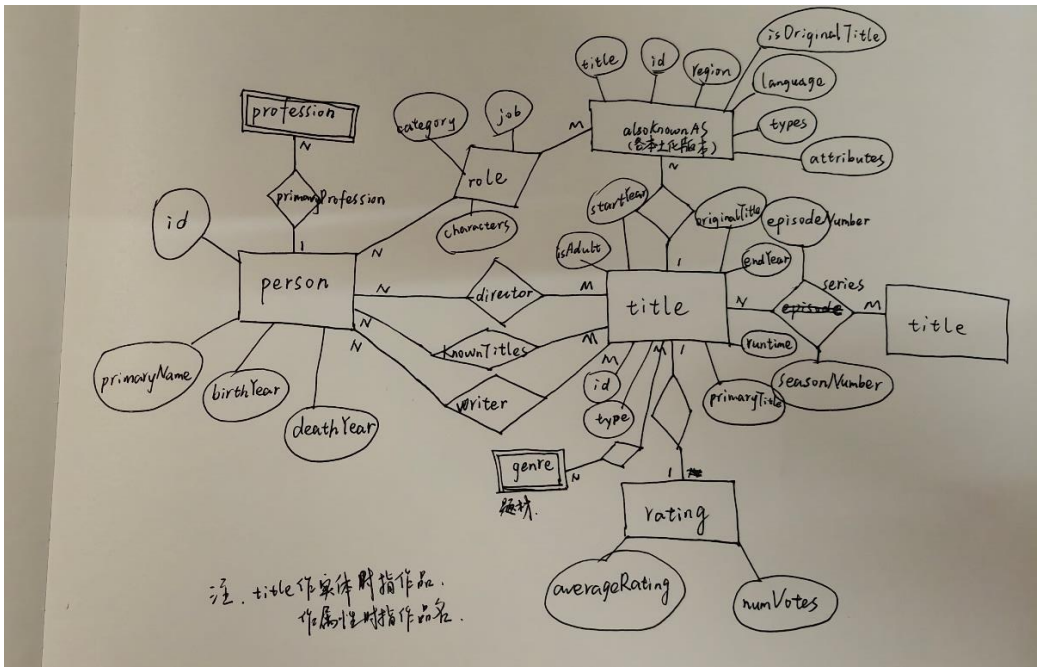
name.basics.tsv 中 primaryProfession 和 knownForTitles 为多值字段且与人员唯一标识 nconst 为多对一关系，将其分别携带 nconst 拆分为两表。

title.akas.tsv 中多值字段 types 和 attributes 不作为检索依据，故拆分，当作整块文本处理。

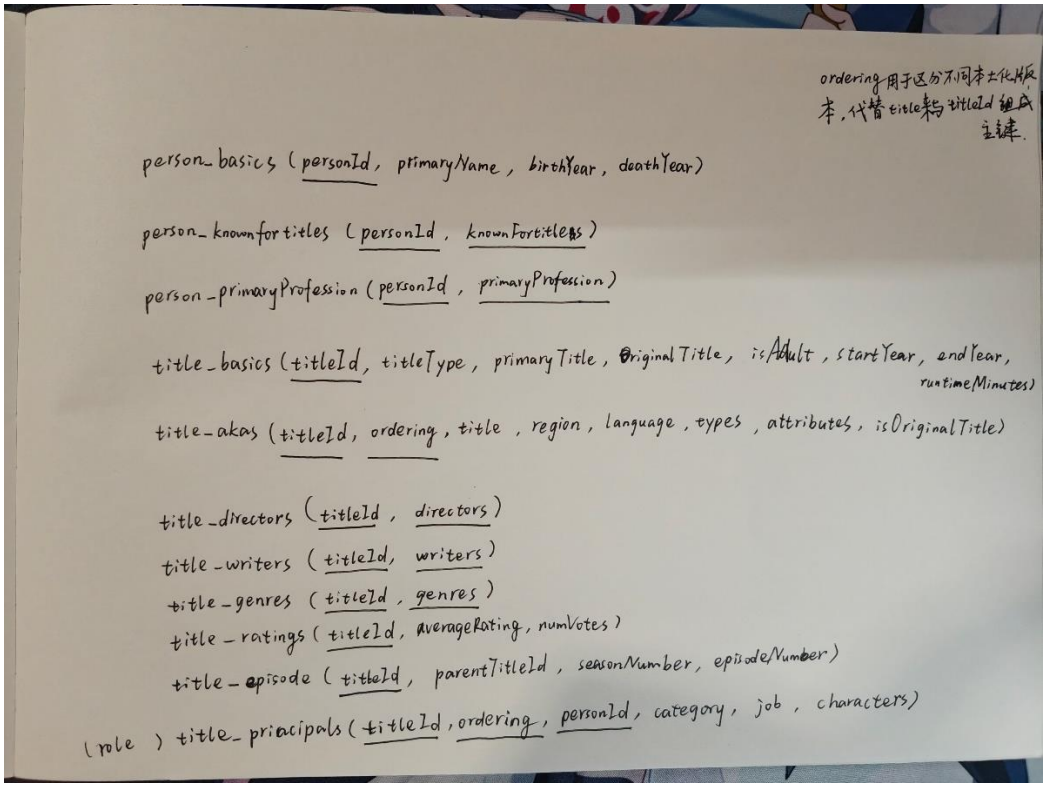
数据拆分由 java 代码实现，见附录。

3.2 ER 图及关系模型

3.2.1 ER 图



3.2.2 关系模型



3.3 SQL 语句

根据名称模糊检索影视作品

```
select titleId, primaryTitle, originalTitle, titleType, startYear, endYear, averageRating, numVotes, count(distinct title)
from
    (title_akas natural join title_basics) natural join title_ratings
where
    titleType!='tvEpisode' and title like '%title%'
group by
    titleId
```

根据唯一标识检索影视作品 genre 属性

```
select genres from title_genres where titleId='titleId'
```

根据唯一标识检索影视作品 writer

```
select * from title_writers where titleId='titleId'
```

根据唯一标识检索作品 director

select * from title_directors where titleId='**titleId**'

根据唯一标识检索作品 actor

select personId from title_principals

where (category='actor' or category='actress') and titleId='**titleId**'

根据唯一标识检索作品剧集信息

select seasonNumber, count(episodeNumber) episodes

from title_episode

where parentTitleId='**titleId**'

group by seasonNumber

根据名字模糊检索从业人员

select * from person_basics where primaryName like '%**name**%'

根据唯一标识检索人物职业

select primaryProfession from person_primaryprofession where personId='**personId**'

根据唯一标识检索人物著名作品

select knwonForTitles from person_knownfortitles where personId='**personId**'

检索作品所有种类

select distinct titleType from title_basics

检索作品所有风格题材

select distinct genres from title_genres

根据 titleType，genre，startYear 三属性筛选作品

select titleId, primaryTitle, originalTitle, titleType, startYear, endYear, averageRating, numVotes

from

    title_basics natural join title_ratings natural join title_genres

where

    titleType!='tvEpisode' and titleType='**titleType**'

and genres='**genre**' and startYear=**startYear**

order by

startYear desc, averageRating desc

根据唯一标识检索作品

select titleId, primaryTitle, originalTitle, titleType, startYear, endYear, averageRating, numVotes

from title_basics natural join title_ratings

where titleId='**titleId**'

根据唯一标识检索人物

select * from person_basics where personId='**personId**'

3.4 优化数据库

对于单属性主键的表中，对检索字段加索引，对于多属性主键的表中，对非主键的检索字段以及主键中的检索字段加索引，提高查询效率。

Ps. 理想情况下项目应用场景为读密集场景，且为提供给普通用户修改表的接口、权限，表的维护更新由管理员在数据库端本地完成，故添加索引以空间换时间。

添加索引的字段如下（均为单字段索引）

Person_basics---primaryName

Title_akas---title

Title_basics---titleType, startYear

Title_episode---parentTitleId

Title_genres---genres

Title_principals---catefory, titleId

4. 后端设计

接口文档见附录

4.1 数据格式

```
public class FigureBriefInfo {
    String personId;
    String primaryName;
    int birthYear;
    int deathYear;
```

```
public class FigureDetailInfo extends FigureBriefInfo{
    List<String> primaryProfessions;     //职业/工作
    List<String> knownForTitles;         //著名作品
```

```
public class TitleBriefInfo {
    String titleId;
    String primaryTitle;
    String originalTitle;
    String titleType;       //类型 short, tvSeries, movie...
    List<String> genres;    //风格/体裁 Comedy, Romance, Sport...
    int startYear;          //若非series则只有start
    int endYear;
    float averageRating;    //评分
    int numVotes;           //评分人数
```

```java
public class TitleDetailInfo extends TitleBriefInfo{
    List<Season> seasons;   //series属性，每季编号及其集数
    List<String> writers;
    List<String> directors;
    List<String> actors;
```

```java
public class Season {
    int seasonNumber;
    int episodes;
```

4.2 接口

与数据库端通过 jdbc 连接

与前端通过 json 传输数据，接口以 url 方式实现

接口如下：

Search：

Title：

数据结构 List<TitleBriefInfo>

http://...:8080/search/title?title=...

Figure:

List<FigureBriefInfo>

http://...:8080/search/figure?name=...

Filter(only title):

筛选条目 type 和 genre 通过以下接口获取

List<String>

/titleTypes

/genres

筛选：

List<TitleBriefInfo>

/filter?titleType=…&genre=…&startYear=…

titleType, genre 为字符串 null 时表示不选此项，startYear 则需为-1

初步搜索筛选出的为 briefInfo，选定某个特定作品/人物后查看 detailInfo（通过唯一 ID）

Detail 接口：

TitleDetailInfo

/view/title?title?titleId=…

FigureDetailInfo

/view/figure?personId=…

作品详细信息中会展示相关人物简略信息，人物详细信息则包含著名

作品简略信息，

DetailInfo 传输时只包含唯一 ID，简略信息通过如下接口获取

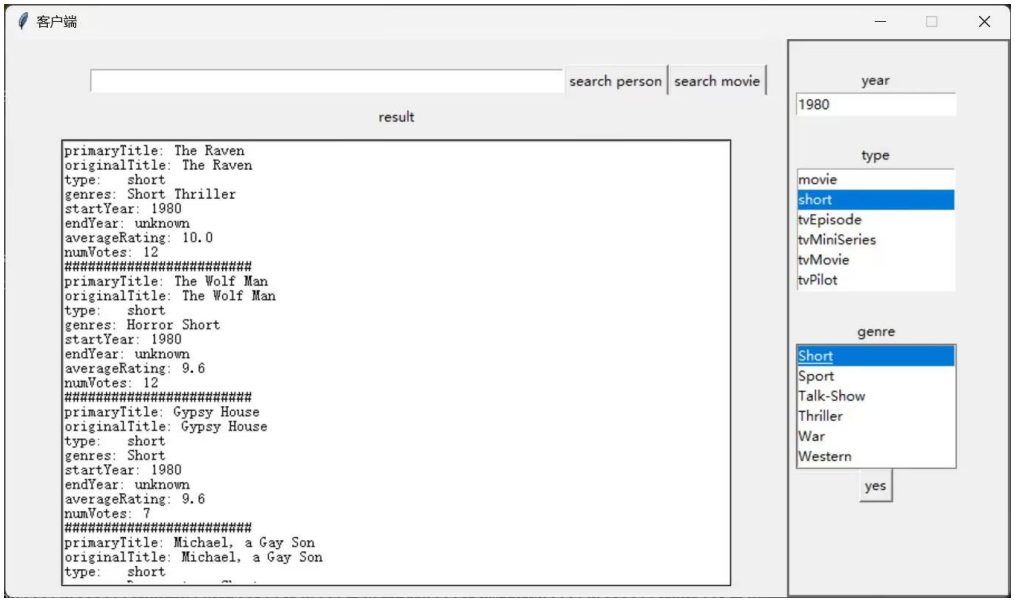
/attach/title?titleId=…

/attach/figure?personId=…

5. 前端设计

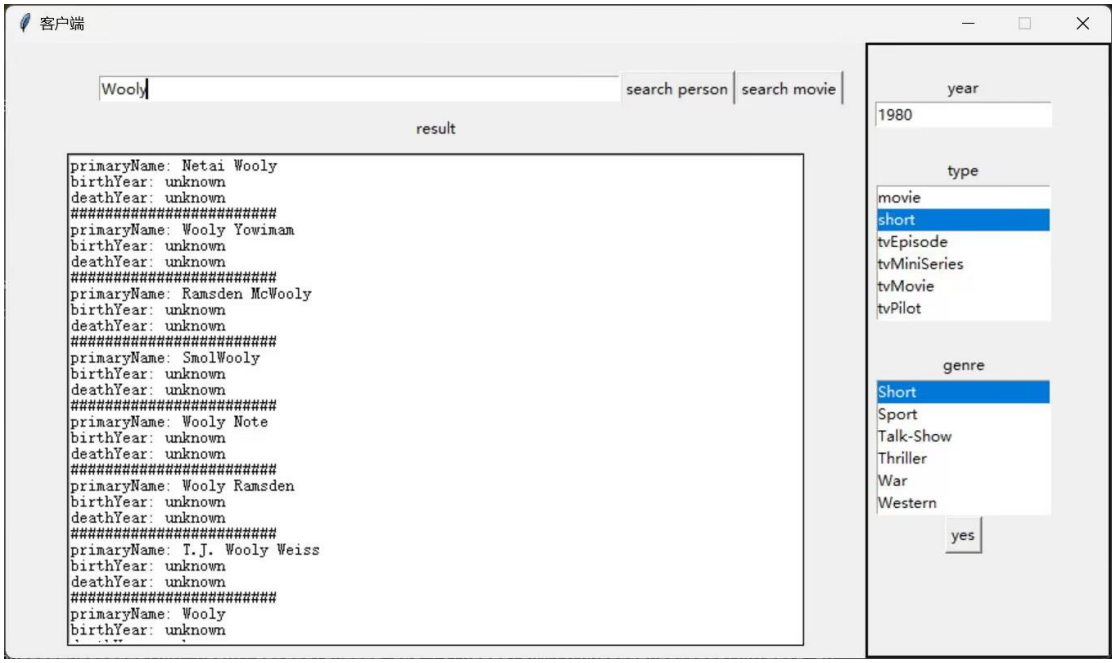
5.1 UI 设计

筛选作品



搜索作品

搜索人物



6. 系统安装及使用说明

数据库：MySQL5.7，任意平台，数据及建表语句见附录。

后端：jdk8，idea2020.3.4，maven3.6.1，其余相关依赖见附录，平台无关但建议 windows。MySQL 连接配置在

\dbproj\src\main\resources\dbproj\src\main\java\com\example\dbproj\dao\DBop.java 中函数 getConnection()中更改即可。

前端：/main.py 使用 PyCharm 运行即可。

7. 附录

附录所有项目文件地址：GitHub - devil-crash/simple-database-course-project

7.1 数据处理项目

/db_apart_file，java 项目，可使用 idea 运行，用于处理 imdb 数据集，输出 tsv 文件用于导入数据表。具体用法见项目中 README.txt。

/cutFile.java，取文件前 500 行，用于取测试用例

7.2 前后端接口文档

interface.docx

7.3 数据及建表语句

数据表及建表语句在 /data/ 中其中建表语句为 createTable.sql 其余.sql 文件为数据表，可使用 navicat 导入

7.4 后端项目依赖

Jdk8 和 maven 需要配置系统变量 JAVA_HOME 及 MAVEN_HOME，path 下配置各自的/bin，maven 在 /apache-maven-3.6.1，idea 中需要配置 maven，相关依赖在 /repository/ 中，依赖的项目配置文件为 \dbproj\src\main\resources\dbproj\pom.xml。idea 中 Maven 配置如下