

数据库课程设计

设计一个**IMDB**风格的小型电影数据库管理系统。

# 一、课程设计题目

**小型电影数据库管理系统（MDB）**

# 二、课程设计目的

针对日益增长的电影观看需求，以及不断变化的电影信息以及上涨的电影数量，需要建立一个高效且精准的电影信息管理系统，这可以帮助有需求的观影者更快速找到自己想要看的电影，同时也能延展了解更多有关信息；并且能够根据数据库中的增删查改从而达到构建私人影片库的功能，更具个性化和具体化。

# 三、总体设计

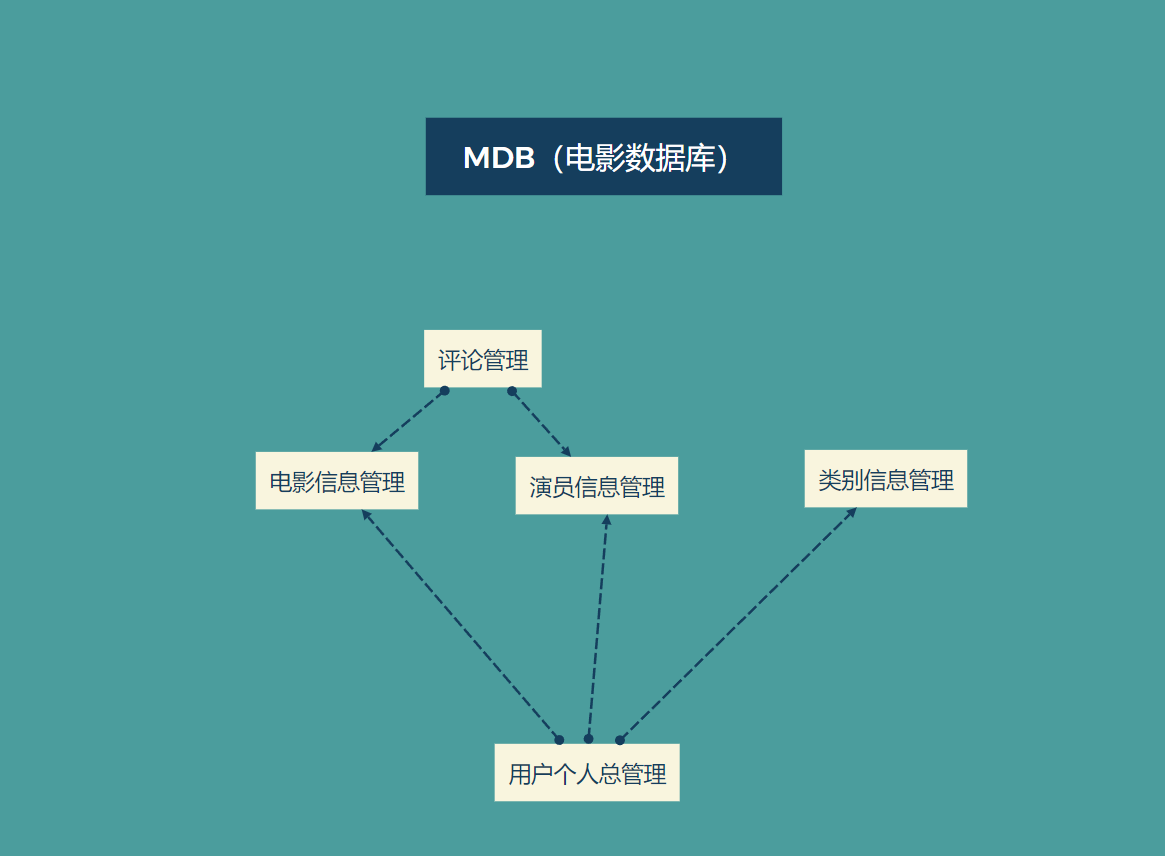
## 1.系统需求分析

### 1.1 系统功能分析

系统的基本功能包含：

* **电影信息管理**：存储电影的基本信息，如电影名称、上映年份、导演、主演等。
* **演员管理**：存储演员的信息，并与电影建立关系。
* **评论管理**：用户可以为电影提供评论。
* **用户个人管理**：注册的用户可以对数据库中的影片信息进行修改、查询等，同时能够自行管理上述四个功能的具体细节。

### 1.2 **数据库设计**



### 1.3 与其他系统的关系

MDB可以与各个平台的电影评论系统、电影发布系统、演员信息系统等建立关联，从而获得更全面的信息。

### 1.4 流程

该数据库系统可以实现多个单方面功能以及多方面联合功能，具体的实现以及流程如下方所展示。

#### 用户

* + 注册：进入登录界面->选择注册选项->输入用户名->输入密码->确认密码->点击确认->录入用户数据于数据库中->注册成功
  + 登录：进入登录界面->选择登录选项->输入用户名->输入密码->点击确认->将输入信息与库中信息进行寻找和比对->如果匹配成功，登录成功；如果匹配失败或未寻找到，登录失败
  + 退出：进入个人界面->点击退出登录->退出成功
  + 个人信息查看-基本信息（仅限邮箱待定）：进入个人界面->选择基本信息选项->进行查看
  + ....

#### 电影

* + 查看-电影基本信息：进入电影界面->选择对应电影->进行相应的查看
  + 排序-按序列号id/字母/类别/评分进行正/倒序排序：进入电影界面->点击排序->选择相应排序方式->选择相应排序顺序
  + 查询-按电影名称
  + ...

#### 演员

* + 查看-演员基本信息：进入演员界面->选择对应演员->进行相应的查看
  + 排序...
  + 查询-按演员名称
  + ...

#### 用户-电影

* + 增删查改：进入电影/演员界面->点击相应功能按键->进行操作->数据更新
  + 进行评论/评分

#### 用户-演员

* + 如上
* ...

### 1.4 数据流程图



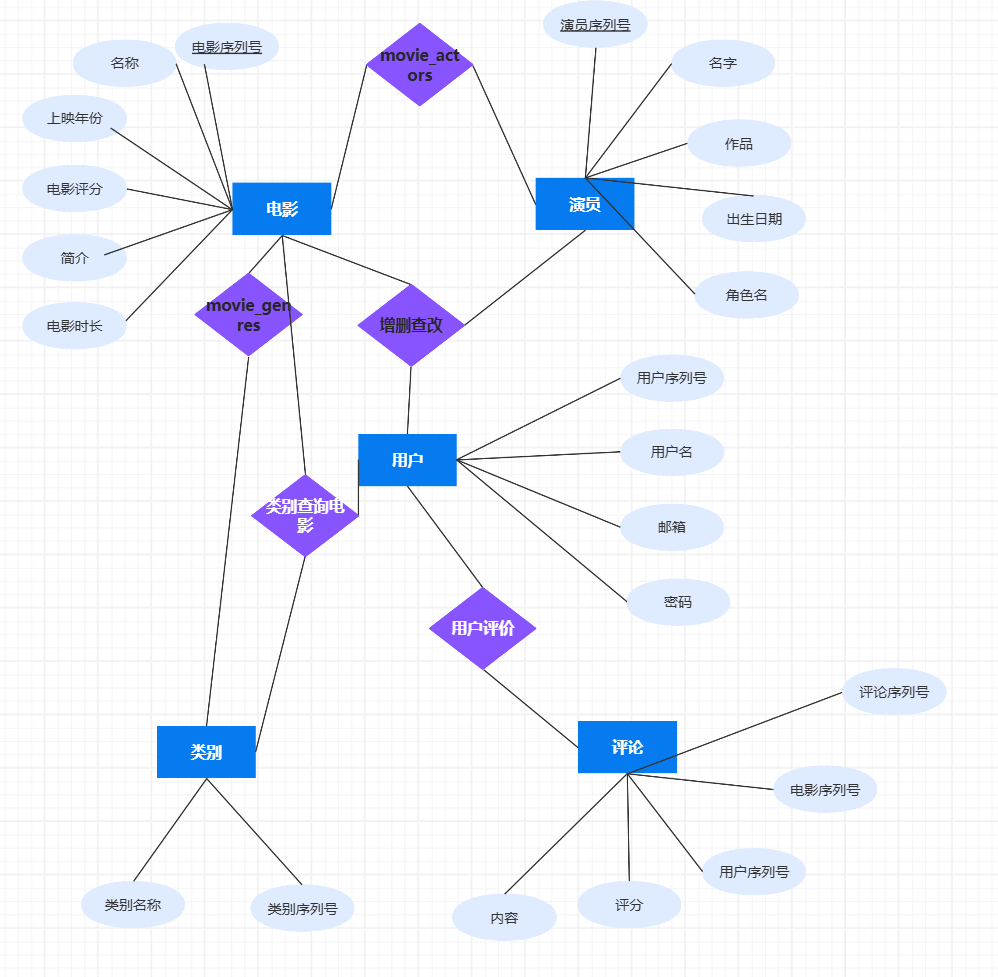
## 2 数据库设计 (movies)

### 2.1 数据库需求分析

根据系统的数据流程图，可以列出以下记录电影信息所需的数据项和数据结构：

* **电影**：电影ID、电影名称、类别、上映年份、导演、主要演员、简介、评分。
* **演员**：演员ID、姓名、性别、出生日期、国籍、出演电影、简介。
* **评论**：评论ID、用户ID、电影ID、评分、评论内容、评论日期。
* **类别**：类别ID、类别名称。
* **用户**：用户ID、用户名、密码、注册日期。

### 2.2 数据库概念结构设计

E-R图大概  


**电影表** (movies)

| 列名 | 数据类型 | 描述 |
| --- | --- | --- |
| movie\_id | INT | 主键，自增 |
| title | VARCHAR(255) | 电影名称 |
| release\_year | INT | 上映年份 |
| description | TEXT | 简介 |
| rating | DECIMAL(3, 2) | 电影评分（0-10） |
| duration | INT | 电影时长（分钟） |

**演员表 (actors)**

| 列名 | 数据类型 | 描述 |
| --- | --- | --- |
| actor\_id | INT | 主键，自增 |
| name | VARCHAR(100) | 名字 |
| works | VARCHAR(100) | 作品 |
| roles | VARCHAR(100) | 角色 |
| birthdate | DATE | 出生日期 |

**用户表 (users)**

| 列名 | 数据类型 | 描述 |
| --- | --- | --- |
| user\_id | INT | 主键，自增 |
| username | VARCHAR(100) | 用户名 |
| email | VARCHAR(100) | 电子邮件 |
| password | VARCHAR(255) | 密码 |

**管理员表 (admins)**

| 列名 | 数据类型 | 描述 |
| --- | --- | --- |
| admin\_id | INT | 主键，自增 |
| adminname | VARCHAR(100) | 用户名 |
| email | VARCHAR(100) | 电子邮件 |
| password | VARCHAR(255) | 密码 |

## 3. **各功能模块设计与实现**

### 3.1 各模块基本实现逻辑

#### **用户管理模块**

1. **注册**（管理员操作类似）
   * 用户通过填写用户名、电子邮件和密码进行注册。
   * 验证用户名和电子邮件的唯一性（确保在数据库中没有重复的用户名或电子邮件）。
   * 如果验证通过，将用户信息插入 users 表中。
2. **登录**
   * 用户通过输入用户名和密码登录。
   * 查询 用户 表，验证邮箱是否存在以及密码是否匹配。
   * 若成功匹配，则允许登录；否则，返回错误信息。
3. **退出登录**
   * 用户可以选择退出账户。
4. **查看个人信息**
   * 通过 user\_id 查询 users 表中的个人信息，包括用户名、电子邮件和注册日期等。

#### **电影信息管理模块**

1. **查看电影基本信息**
   * 通过 movie\_id 查询 movies 表，显示电影的标题、简介、上映年份、评分等基本信息。
2. **排序电影**
   * 根据用户选择的排序方式（如上映年份、评分、标题等），对 movies 表进行排序。
3. **按条件查询电影**
   * 用户可以通过电影名称、类别或上映年份等条件查找电影。
   * 如果按类别查询，需要联合查询 movie\_genres 表和 genres 表。

#### **演员信息管理模块**

1. **查看演员基本信息**
   * 通过 name 查询 actors 表，获取演员的名字、出生日期、国籍等信息。
2. **查询演员的出演电影**
   * 获取演员出演的所有电影信息。

#### **评论管理模块**

1. **添加评论**
   * 管理员可以为电影添加评论，数据将存储在 movies的rating中。
2. **查看评论**
   * 通过 movies 查询 ，可以获取评论。
3. **删除评论**
   * 用户可以删除自己发布的评论。

### 3.2 界面设计

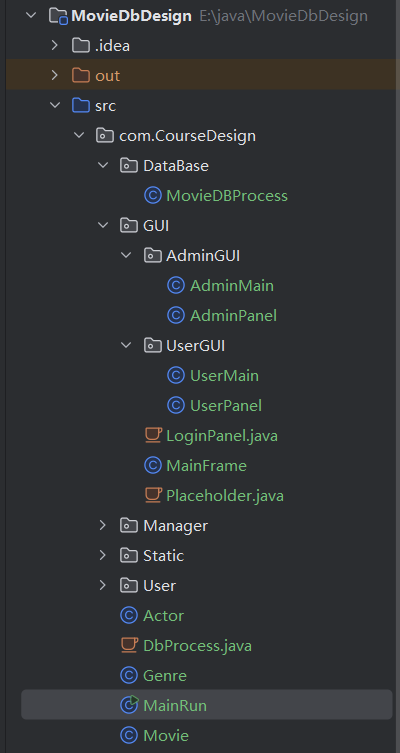
系统界面设计为一个主窗体，内含不同的标签页来完成5项主要功能

* 应用程序主窗体的创建（可进入不同的界面）  
  用户界面
* 普通用户登录界面
* 普通用户注册界面
* 普通用户详细信息界面
* 管理员登录界面
* 管理员注册界面
* 管理员详细信息界面  
  电影
* 电影详细信息界面
* 删除电影界面
* 添加电影界面  
  演员
* 演员详细信息界面
* 添加演员界面
* 删除演员界面  
  查询
* 查询电影界面
* 查询演员界面
* 电影信息输入界面
* 演员信息输入界面
* 评论信息输入界面
* 用户管理数据界面

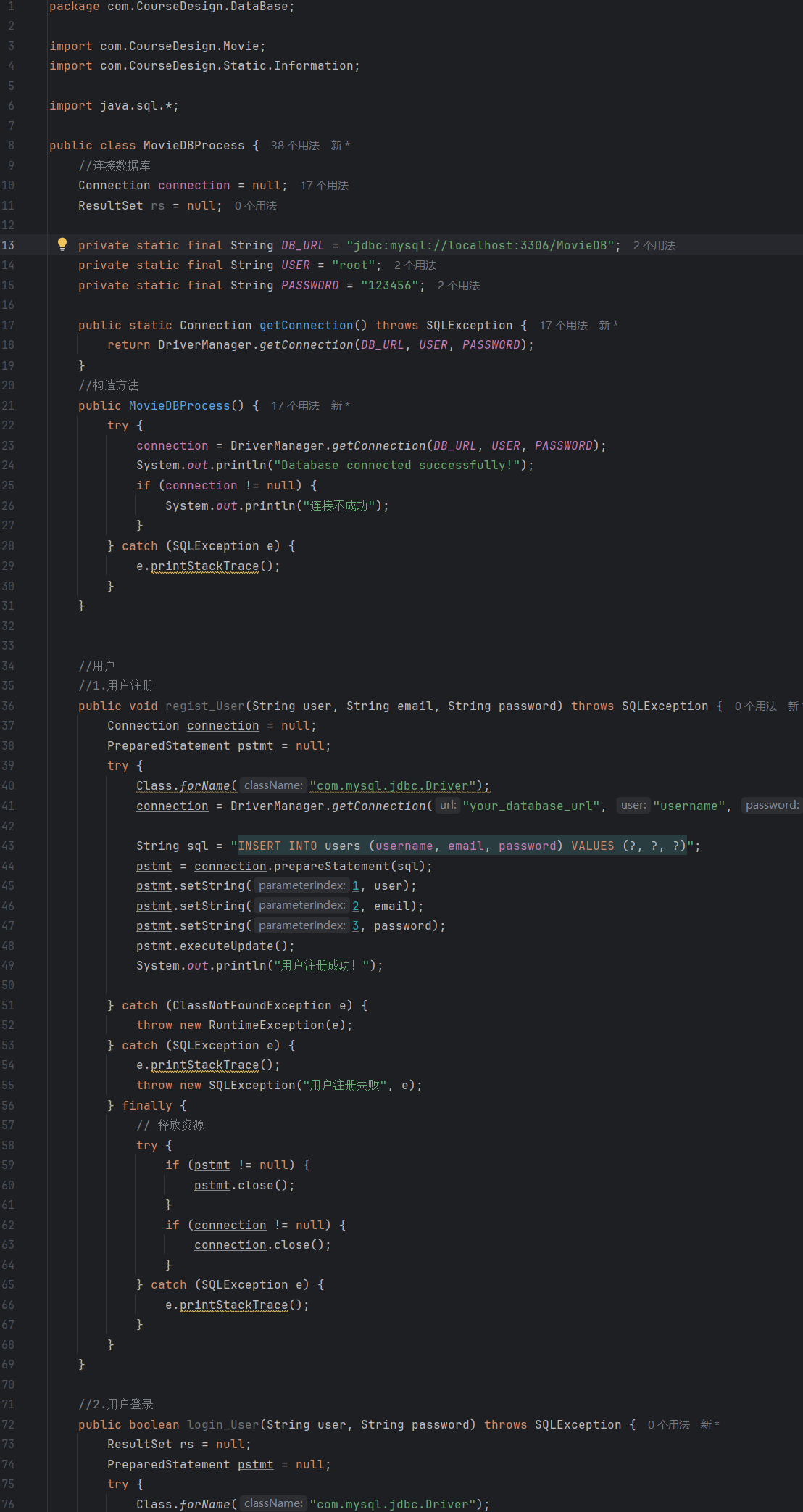
# 四、详细设计

### 代码分模块实现以及注释解释

代码文件目录：

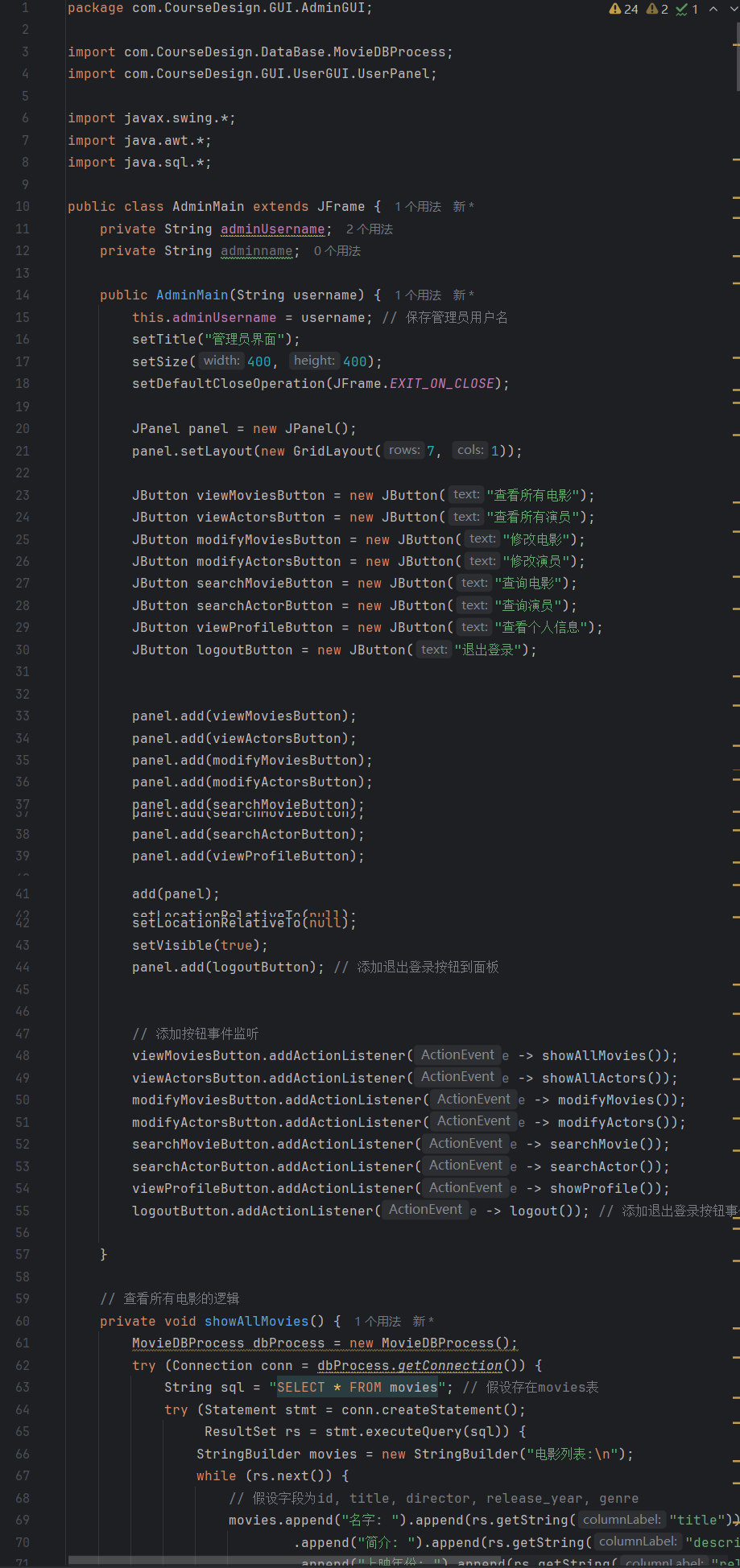


MovieDBProcess部分（JDBC部分）

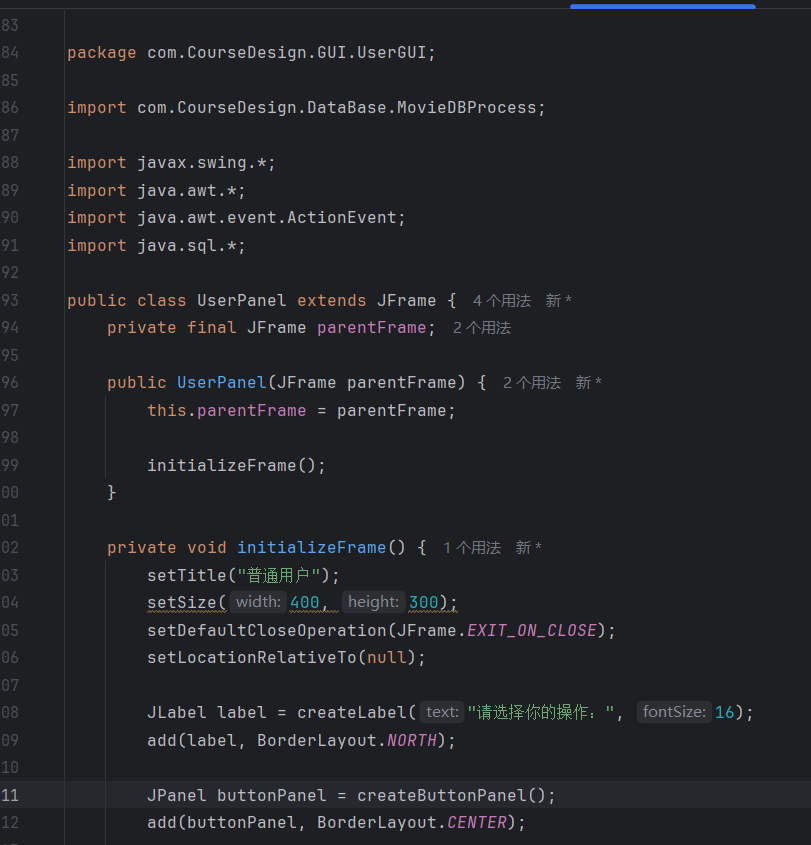


GUI部分（界面设计）

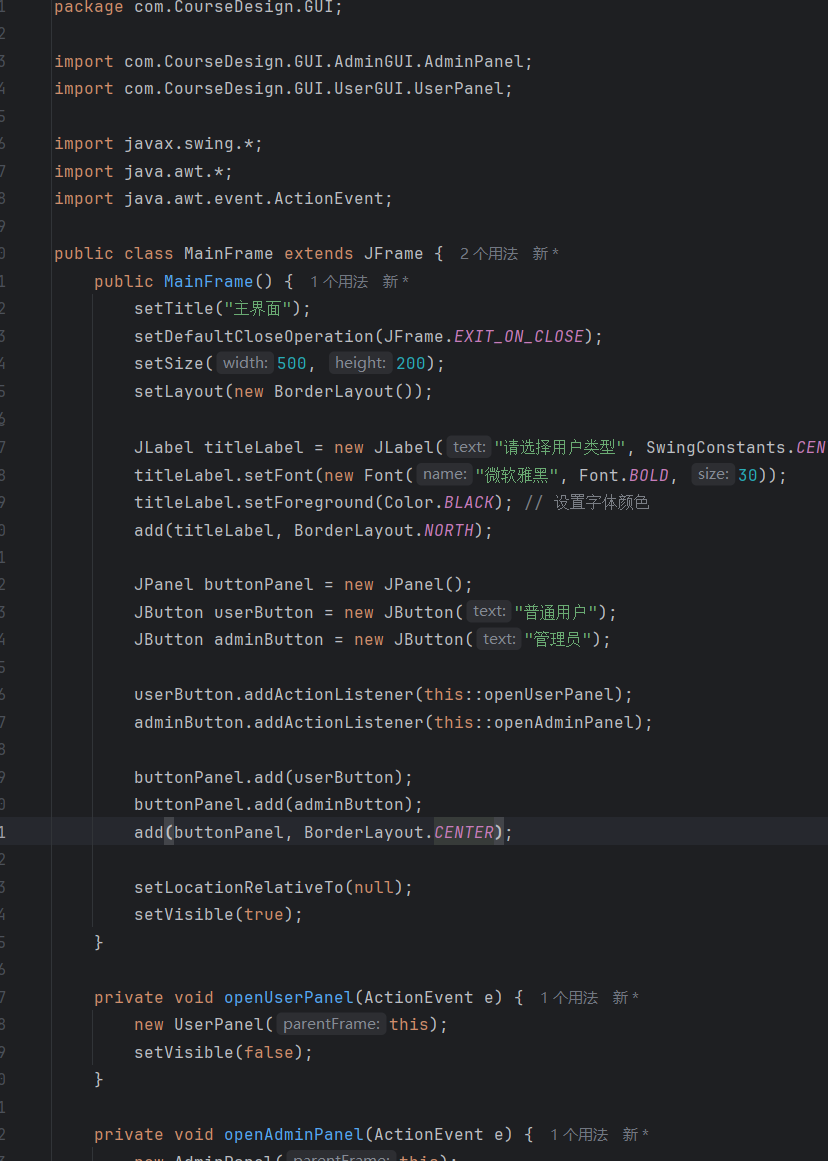
- 管理员



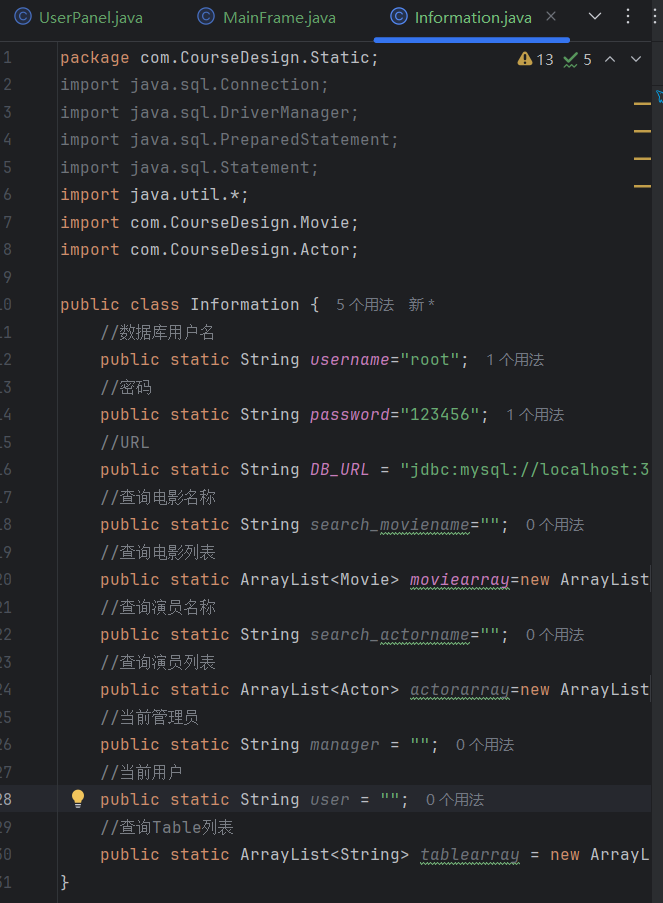
-用户



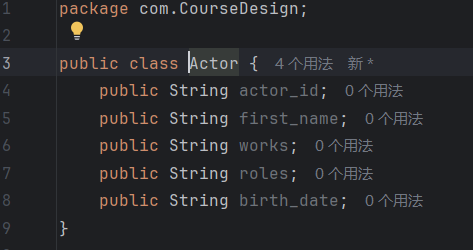
MainFrame部分（主界面）



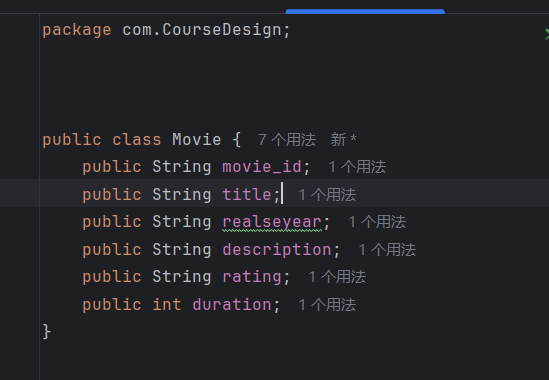
# Information部分



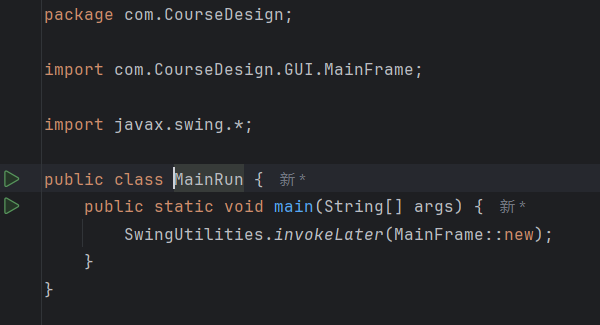
Actor部分



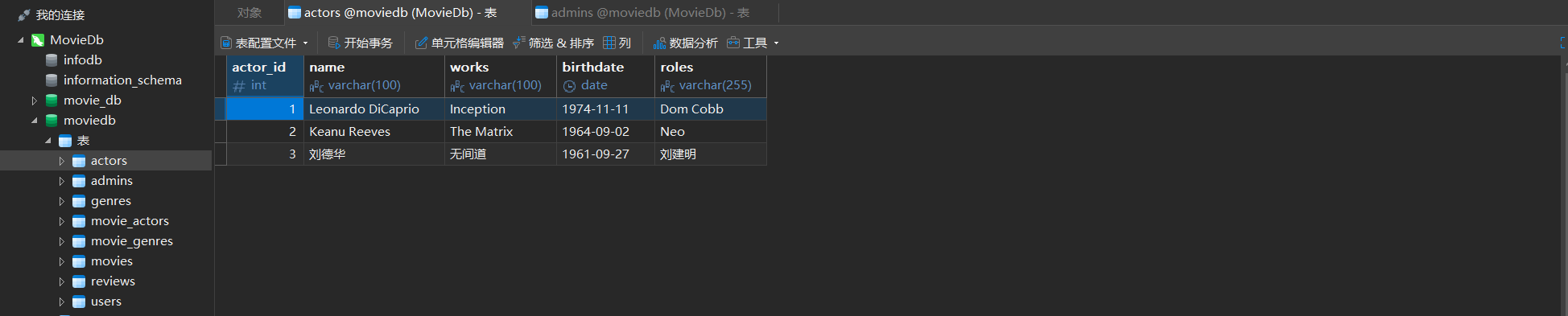
Movie部分

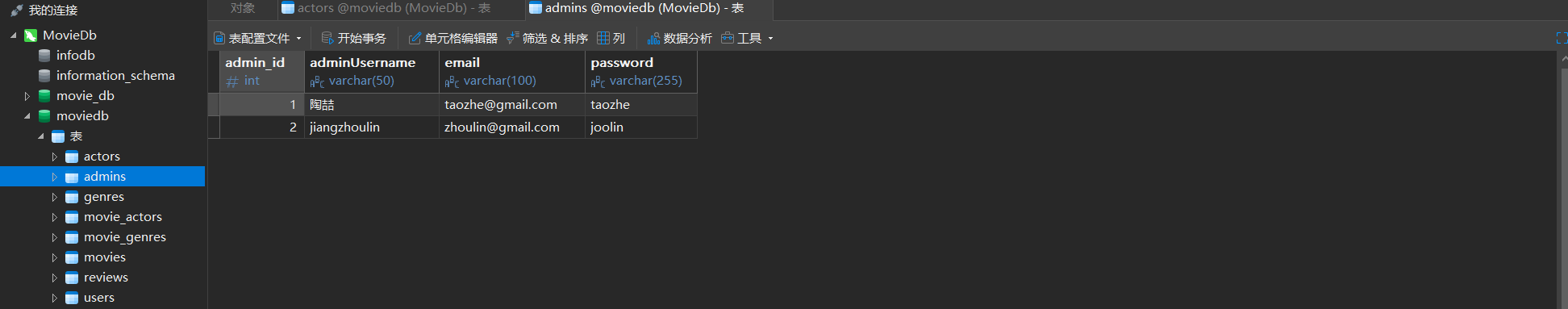


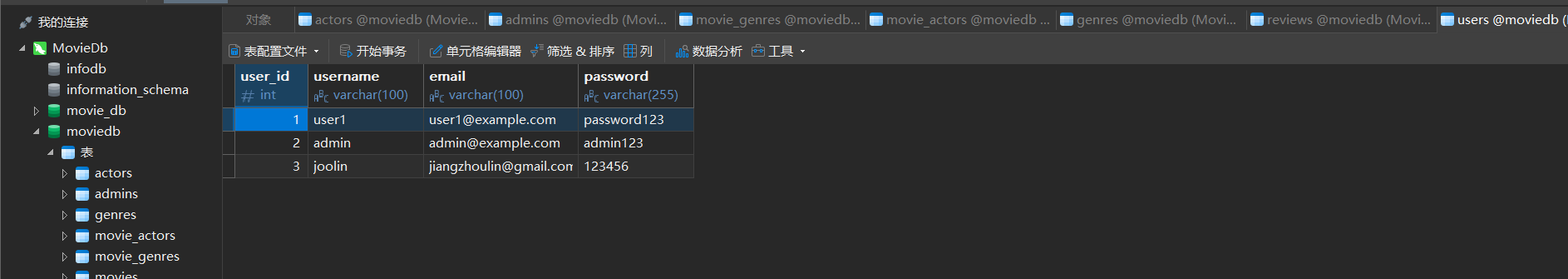
MainRun部分



数据库部分

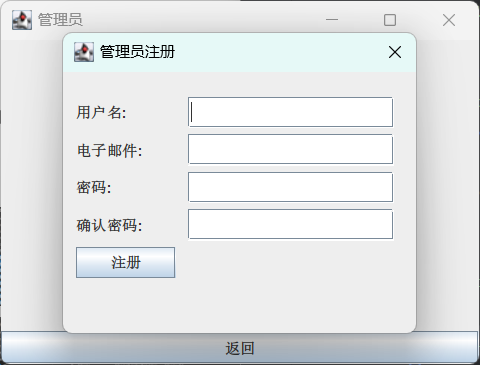




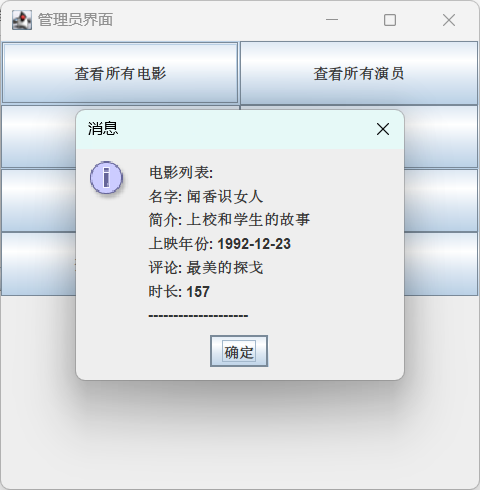


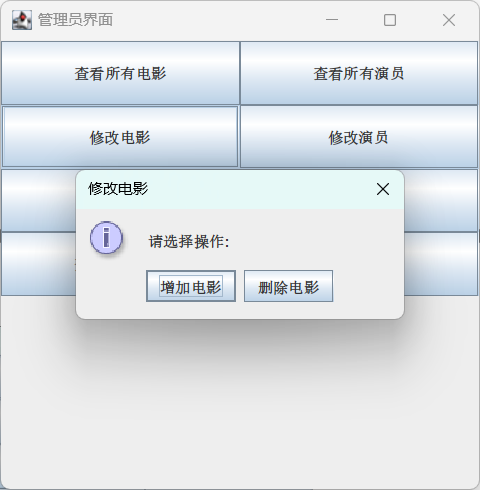
# 五、结果与分析

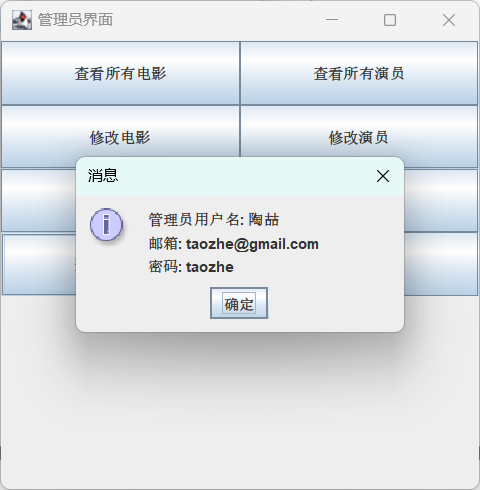
实验结果-界面展示：











# 六、小结与心得体会

经过两周的课程设计与学习，基本了解与掌握了例如JDBC、MySQL的基本应用、Java语法的深入学习，通过对这些知识的进一步掌握，从而助力完成了最终的课程设计，其中有不解有疑惑，在与老师和同学之间的积极讨论之后都有了较好的解决方案。这次的数据库课程设计让我理解了一个项目、一个管理系统的诞生要经历什么，以及其中需要解决的问题又有哪些。

在本次课程设计中，我以 **Java** 为主要编程语言，结合 **Swing** 图形化界面开发框架和 **MySQL** 数据库，完成了一个功能较为完善的电影数据库管理系统。本系统具备电影信息的添加、查看、更新、删除、搜索等功能，同时实现了简单的分类筛选与统计操作。这次课程设计让我从理论知识的学习，深入到实际项目的开发中，收获颇丰。

#### 课程设计的主要收获

1. **理论与实践相结合**
2. **系统开发的完整流程**
3. **问题解决能力的提升**

#### 心得体会

1. **强化了面向对象编程的理解**
2. **理解了软件开发中的协同工作**
3. **实际开发中重视细节**
4. **动手能力与独立思考能力的提升**

#### 总结

通过本次课程设计，我不仅巩固了 Java 编程的基础知识，还掌握了 GUI 与数据库的结合应用，同时深入理解了软件开发流程。从最初的需求分析到最终系统的实现与优化，这是一段充满挑战但又收获满满的经历。未来，我将继续努力，进一步拓展自己的知识面和实践能力，为更复杂的项目做好准备。