

# 数据库实践课程报告

题目： 动画信息管理系统

组员 1 姓名： 钱庭涵

组员 1 学号： 10152130122

完成时间： 2018.7.2

华东师范大学计算机科学技术系

## 目 录(供参考，可根据实际情况调整)

一.	系统需求分析 .....	1
1.	系统描述 .....	1
2.	需求分析和应用程序功能.....	2
二.	数据库概念设计 .....	10
1.	确定实体和属性.....	10
2.	E-R 图 .....	11
三.	数据库逻辑结构设计 .....	12
1.	关系模式设计.....	12
2.	基本表设计 .....	13
四.	数据库物理设计和实施 .....	17
1.	数据库的创建.....	17
2.	创建基本表 .....	18
3.	触发器设计 .....	22
4.	存储过程设计.....	23
5.	视图设计 .....	23
6.	其他数据库对象的设计.....	35
五.	应用程序设计 .....	36
1.	开发及运行环境介绍.....	36
2.	主要功能设计.....	36
3.	主要界面 .....	38
六.	心得体会 .....	42

(此目录由 WORD 自动生成。正文中的各类标题只须更改内容，不要更改格式。最后在本页中右击上方的目录区域，选择“更新域”命令。)

## 一. 系统需求分析

### 1. 系统描述

随着近些年来动画行业的蓬勃发展，每个季度播出的动画数量越来越多，种类越来越繁杂，动画制作公司、声优人数也在不断上升中。在每个季度如此多的动画中，不少喜欢日本动画的爱好者们不知应从哪里获取全面而客观的动画信息，以找到自己想看的动画。

在哔哩哔哩动画网上有不少漫评 up 主，但他们对于新番的介绍往往带有较强的主观色彩，不适用于每个人。但有一个 up 主会在每个季度出一个视频，介绍完整又详细的本季度新番信息，不带有主观的评价，而是客观的呈现介绍动画应有的一些信息。但该 up 主由于工作繁忙，于 2018 年 4 月更新了最后一期新番介绍视频之后，停止了更新。因此我想要设计一个动画信息管理系统。

动画信息管理系统包括动画信息及一系列与动画相关信息（如动画制作公司、动画原作公司及原作者、动画制作组、动画声优、动画音乐及歌手等）查询、按照特定条件查询推荐动画、查询喜欢的声优本季度配了多少动画等前台功能，以及所有动画相关信息的添加、修改、删除等后台管理功能。

本系统面向的用户分别为普通用户和管理员。普通用户只需输出自己的用户名即可登录，无需注册，只有浏览查询、获取推荐信息的权限。管理员需要切换到管理员登录方式进行登录，需要输入特定的用户名及相应的密码，除了浏览查询、获取推荐信息的权限，还可以进行所有动画相关信息的添加、修改、删除等数据管理操作。

动画信息管理系统为喜爱日本动画的爱好者们提供了便利，也省去了制作新番介绍视频的工作量 and 时间，只需往系统中添加新的信息，即可实现介绍与推荐。

## 2. 数据存储需求

### ①

动画信息管理系统数据库需要存储如下信息：

动画的名称、制作类型、改编类型、原作公司、原作者、制作公司、新番季度、首播时间、总集数、类别。

动画的原作公司和动画的制作公司的名称、地址、成立时间、代表作，原作公司还需存储相应的主要业务。

动画制作组中每个工作人员（STAFF）的姓名、代表作、制作动画、相应职位。

动画的声优（CV）列表，出演动画及角色。

动画的音乐列表，音乐名称、类型及演唱歌手。

原作者、声优、歌手或组合或乐队的姓名、性别、出生日期、职业、代表作。

### ②

系统内所有名称或姓名，均不考虑重名情况，能够输入数据库的即为唯一。

系统内所有代表作，与数据库内存储的动画无关，即代表作内动画的信息不需要被数据库包含。

### ③

动画的制作类型有“新作”、“续作”、“衍生”、“重制”四种，改编类型有“漫改”、“轻改”、“游戏改”、“原创”四种，新番季度的格式为 4 位年份+“年”+“春”/“夏”/“秋”/“冬”，如“2018 年春”即 2018 年 4 月番（2018 年 4 月前后开始播出的动画），“2018 年冬”即 2018 年 1 月番，部分未完结动画可能未公布总集数，设为默认值 0。

若改编类型为“原创”，则由动画制作公司独立制作，不存在原作公司，不考虑原作者的情况；

若改编类型为“游戏改”，则原作公司为游戏公司，游戏的原作者可能为一个团队，因此当原作者为一人时，记录原作者，当原作者为团队时则不考虑；

若改编类型为“漫改”，则原作公司为漫画出版社，原作者为漫画家；

若改编类型为“轻改”，则原作公司为轻小说出版社，原作者为小说家。

一部动画一定有一个制作公司，一个动画公司可以制作一个或多个动画。

一部动画不一定有原作公司和原作者，但至多有一个原作公司和一个原作者，一个原作公司可以出版多部动画的原作，一个原作者可以写多部被动画化的原作。

一部动画可以有多个类别，至少得有一个类别。一个类别包含多个动画，至少包含一部动画。

一部动画有一个或多个制作人员，一个制作人员也可以参与多部动画的制作。

一部动画可以有一个或多个声优参与配音，一个声优也可以参与多部动画的配音工作。

一部动画可以有一首或多首音乐，一首音乐只能属于一部动画。

#### ④

原作公司和动画公司必须有名称，其他信息未知时可为空。

成立时间可能有只知道年份、只知道年月等部分时间信息的情况。

#### ⑤

人物包括原作者、声优、歌手或组合或乐队。

性别未知时可为空，组合、乐队等多人团体的性别也置为空。

出生日期和代表作未知时可为空。组合、乐队等多人团体的出生日期为成立时间，或出道时间，或发售第一张专辑的时间。出生日期可能有只知道年份、只知道年月等部分时间信息的情况。

漫画原作者的职业为漫画家，轻小说原作者的职业为小说家，声优的职业为声优，其余人物的职业不固定。

人物中包含一项特殊的数据，名称为“纯音乐”，职业为“纯音乐”，其余信息为空，专门用来与无人声演唱的音乐建立联系。

⑥

系统中不记录无用的 **staff** 工作人员，即数据库中含有的工作人员一定参与了数据库中某部动画的制作，一部动画的一个职位可以有多个工作人员，一个工作人员在一部动画里也可以有多个职位。

**Staff** 代表作未知时可为空。

当一个工作人员在一部动画里有多个职位时，必须按照特定格式来记录，职位的种类有“导演”、“编剧”、“音乐”、“动画人设”，职位叠加时需要按顺序且中间用“、”分隔，即职位总共有以下 15 种可能：

导演

编剧

音乐

动画人设

导演、编剧

导演、音乐

导演、动画人设

编剧、音乐

编剧、动画人设

音乐、动画人设

导演、编剧、音乐

导演、编剧、动画人设

导演、音乐、动画人设

编剧、音乐、动画人设

导演、编剧、音乐、动画人设

⑦

数据库中已存储的一部动画的声优列表并不是参与这部动画配音的所有声优。但已存储的声优一定有一个配音角色。

⑧

数据库中只记录了一部动画的片头曲（OP）和片尾曲（ED），不记录插曲和背景音乐，因此数据库中音乐类型只有“OP”和“ED”两种，一部动画可以有多个 OP 和 ED。

一首音乐可以由多人演唱，也可能无人演唱，无人演唱时歌手存储为“纯音乐”。

### 3. 系统常做的查询与更新

经常做的查询，需要创建有关视图的：

- 一部动画非常全面的基本信息
- 每个季度新番数量，新作数量，续作数量，衍生动画数量，重制动画数量，漫改动画数量，轻改动画数量，游戏改动画数量，原创动画数量
- 一部动画的配音列表及声优代表作
- 一部动画的所有音乐及相应的所有歌手
- 一部动画的制作组列表及工作人员职位和代表作
- 包含新番年份的动画信息
- 数据库内所有新番季度及新番年份
- 季番列表、半年番列表、长篇动画列表
- 担任过导演职位的工作人员以及相应制作的动画信息
- 担任过编剧职位的工作人员以及相应制作的动画信息
- 担任过音乐职位的工作人员以及相应制作的动画信息
- 担任过动画人设职位的工作人员以及相应制作的动画信息
- 一个声优配音过的所有动画及相应的动画信息
- 一位歌手/组合/乐队演唱过的所有动画及相应的动画信息
- 一个工作人员制作过的所有动画及相应的动画信息
- 一个类别的所有动画

- 原作公司被动画化的所有动画及相应的动画信息
- 动画公司制作过的所有动画及相应的动画信息
- 原作者被动画化的所有原作作品及相应的动画信息
- 一个季度的所有动画音乐信息

关于添加、更新、删除

- 每到一个季度，需要大量添加本季度动画信息及所有其他相关信息
- 大量添加之后，可能需要更新，比如动画首播时间、总集数公布，动画类别的添加，动画跳票（不在本季度播出，改到其他季度），数据库内为空的信息的补充（如代表作为空、地址为空），动画换制作公司，动画声优追加，声优因生病退出动画配音等等
- 原作公司、动画公司、人物的相关信息只要不为空且完整（成立时间为完整年月日、出生日期为完整年月日、地址确切等），一般不更新，数据库内其余信息都可能更新。

#### 4. 应用程序功能

查询的主要功能如下：

##### 1) 动画查询

- a) 动画基本信息：查询某部动画的基本信息（包含动画名称、简介、制作类型、改编类型、原作公司、原作公司代表作、原作者、原作者代表作、动画类别、制作公司、制作公司代表作、新番季度、首播时间、总集数、staff 列表、staff 代表作、cast 列表、cast 代表作、动画音乐名称及类型、歌手）
- b) 制作类型：可限制放送时间（所有、特定年份、特定季度），查询某种制作类型的动画，因为用户可能只想看新作，因为没看过前作，所以不想看续作等
- c) 改编类型：可限制放送时间（所有、特定年份、特定季度），查询某种改编类型的动画，因为用户可能只想看有原作的动画，不想看原



创动画，无原作可考究

- d) 原作公司：可限制放送时间（所有、特定年份、特定季度），查询某个原作公司的动画，比如芳文社出品百合合作、key 社的游戏
- e) 原作者：可限制放送时间（所有、特定年份、特定季度），查询某个原作者的动画，比如想看麻枝准写的游戏的动画
- f) 制作公司：可限制放送时间（所有、特定年份、特定季度），查询某个制作公司的动画，比如想看看这个季度（或者这年、全部）A1 又出了什么动画，如果查询的是一个季度，就也返回当季度 A1 做的动画数量，毕竟开太多必会崩或者经费不足
- g) 季度：查询某个季度的新番，或者某个季度的旧番，及显示当季度动画数量，每种制作类型的数量，每种改编类型的数量
- h) 首播时间：查询某段首播时间的动画，比如想看看 4 月第一周都有什么动画开播了
- i) 总集数：可限制放送时间（所有、特定年份、特定季度），查询某个范围总集数的动画，比如不想看太长的，只想看 12 集 13 集的那种季番，想一天一口气看完
- j) STAFF：可限制放送时间（所有、特定年份、特定季度），根据职位，查询某个 staff 制作的动画，比如导演是新房昭之的、编剧是大河内一楼的、编剧是冈田磨里的、音乐是泽野弘之配的、音乐是横山克的
- k) CV：可限制放送时间（所有、特定年份、特定季度），查询某个声优参与配的动画，也返回配的角色姓名
- l) 演唱：可限制放送时间（所有、特定年份、特定季度），查询某个歌手/组合/声优/乐队唱了主题曲的动画，主题曲包括 OP 和 ED
- m) 类别：查询某种类别的动画，可以限定多种类别（最多四种），将返回符合四种类别中任一种的动画，并且注明查询结果动画的所有类别

## 2) 原作公司查询

- a) 原作公司基本信息：查询某个原作公司的基本信息

- b) 被动画化的作品：可限制放送时间（所有、特定年份、特定季度），查询某个原作公司被动画化的作品，显示该公司当前放送时间限制范围内作品数量
  - c) 主要业务：根据主要业务，查询表内所有漫画出版社、轻小说出版社、游戏公司
- 3) 动画公司查询
- a) 动画公司基本信息：查询某个动画公司的基本信息
  - b) 制作动画：可限制放送时间（所有、特定年份、特定季度），查询某个动画公司制作的动画作品，显示该公司当前放送时间限制范围内作品数量
- 4) 人物查询
- a) 人物基本信息：查询某个人物的基本信息
  - b) 职业：查询表内所有漫画家、小说家、声优，声优的查询可以限制性别
  - c) 作品参与：根据参与方式，查询某个季度作品被动画化了 or 参与配音了的所有漫画家/小说家/声优/歌手等，及配音的动画数量，从多到少排序
- 5) Staff 查询
- a) Staff 基本信息：查询某个 staff 的基本信息，担任过的职位
  - b) 职位：根据职位选择，查询哪个职位都有哪些人
  - c) 作品参与：可限制放送时间（所有、特定年份、特定季度），查询某个 staff 都担任过哪些动画的哪些职位，及参与动画数量，降序排
- 6) 音乐查询
- a) 音乐基本信息：查询某首音乐的基本信息（名称、歌手、类型、出自动画）
  - b) 季度：查询某个季度的所有新番动画音乐
  - c) 歌手：可限制放送时间（所有、特定年份、特定季度），查询某个歌手在唱过的音乐（包括名称、类型、出自动画）

管理数据的主要功能如下：

- 1) 原作公司：原作公司表添加、修改、删除
- 2) 动画公司：动画公司表添加、修改、删除
- 3) 人物：人物表添加、修改、删除
- 4) 动画（添加）：动画表添加
- 5) 动画（删改）：动画表修改、删除
- 6) 动画类别：动画类别表添加、修改、删除
- 7) 动画制作组：制作表和 staff 表同步添加、修改、删除，数据库中不存储无用的 staff 信息，即要求每个 staff 一定要参与数据库中某部动画的制作，否则删除 staff 表中该条信息
- 8) 动画声优：cast 表添加、修改、删除
- 9) 动画音乐：音乐表添加、修改、删除
- 10) 音乐演唱：演唱表添加、修改、删除

反馈功能：（来不及实现了 QAQ）

用户可以对每条信息进行匿名建议反馈，提出应该增加/删除的整条信息（如某部新番信息公布而系统内还没更新，如某部番突然跳票，某部番追加 cast 等等），可以补充的信息（比如补充一些系统中没有代表作的 staff 的代表作），错误或不准确需要修改的信息。

管理员可以查看用户提交的所有反馈信息（但看不到用户的用户名），管理员可以一键同意、一键否决用户的修改信息，也可以点击修改（在用户的提交信息的基础上修改信息再执行）。

二次反馈：用户提交的建议反馈和管理员处理的结果存储在数据库中，用户可以查询自己的历史提交反馈记录，以及处理结果（待处理、采纳、修改后采纳、未采纳）。管理员可以查询所有历史反馈记录，以及处理结果，或是修改方案，但看不到是哪个用户提交的。

## 二. 数据库概念设计

### 1. 确定实体和属性

分析动画信息管理系统的需求，将系统中设计的人、物进行抽象，得到了系统的实体如下：

- 1) 原作公司实体集。属性包括：名称、地址、成立时间、主要业务、代表作
- 2) 动画公司实体集。属性包括：名称、地址、成立时间、代表作
- 3) 人物实体集。属性包括：姓名、性别、出生日期、职业、代表作
- 4) staff 实体集。属性包括：姓名、代表作
- 5) 动画实体集。属性包括：名称、制作类型、改编类型、新番季度、首播时间、总集数
- 6) 分类实体集。属性包括：类别
- 7) 音乐实体集。属性包括：名称、类型

## 2. E-R 图

系统 E-R 图如图 2-1 所示：

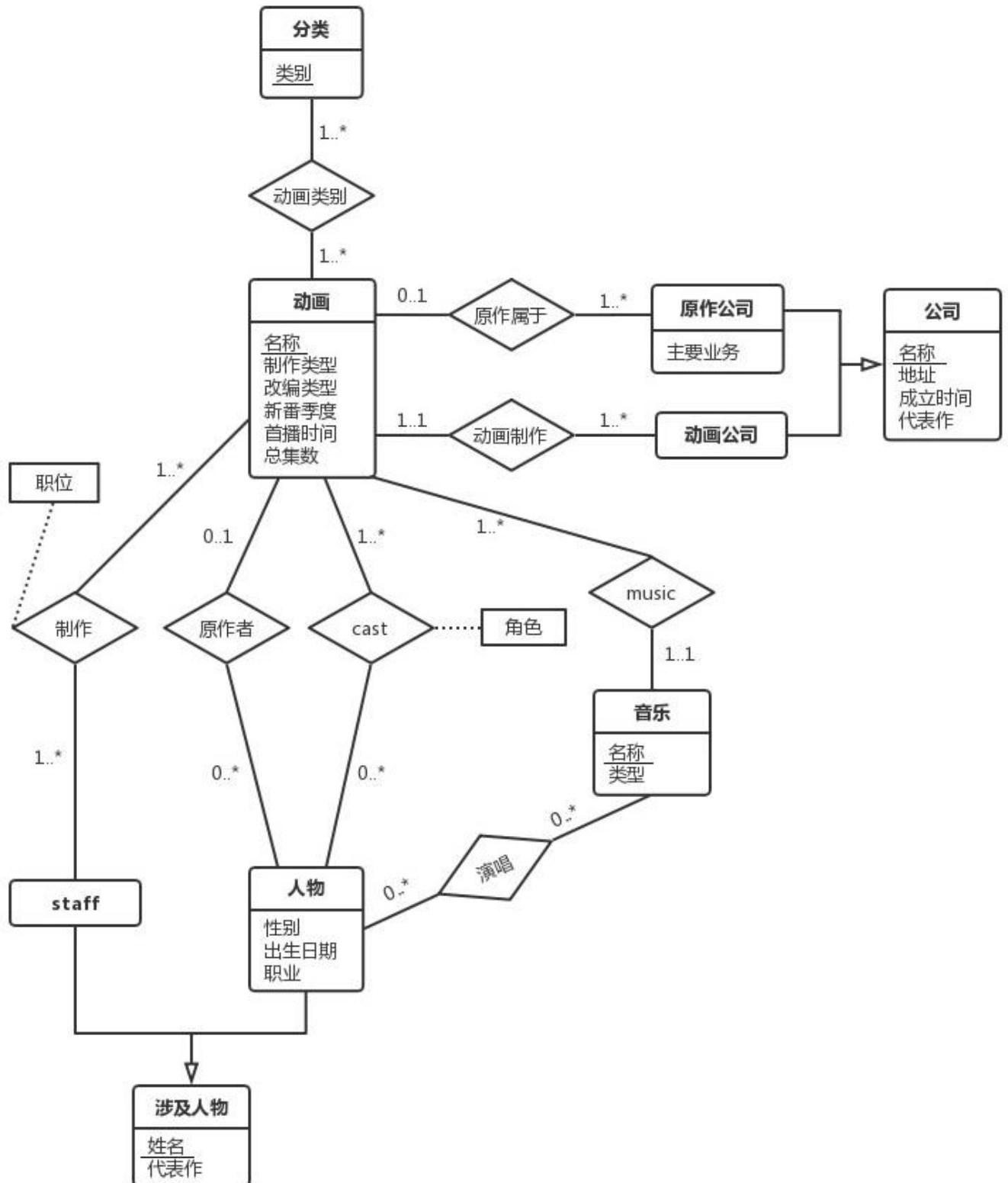


图 2-1 E-R 图

### 三. 数据库逻辑结构设计

#### 1. 关系模式设计

根据概念结构设计得到的 E-R 图和转换规则，得到如下关系模式（主键用下划线标出，外键予以说明）：

原作公司（名称，地址，成立时间，主要业务，代表作）

动画公司（名称，地址，成立时间，代表作）

人物（姓名，性别，出生日期，职业，代表作）

staff（姓名，代表作）

动画（名称，制作类型，改编类型，原作公司，原作者，制作公司，新番季度，首播时间，总集数）

Foreign key 原作公司 reference 原作公司 的 名称

Foreign key 原作者 reference 人物 的 姓名

Foreign key 制作公司 reference 动画公司 的 名称

动画类别（名称，类别）

Foreign key 名称 reference 动画 的 名称

制作（姓名，动画，职位）

Foreign key 姓名 reference staff 的 姓名

Foreign key 动画 reference 动画 的 名称

cast（姓名，动画，角色）

Foreign key 姓名 reference 人物 的 姓名

Foreign key 动画 reference 动画 的 名称

音乐（名称，类型，动画）

Foreign key 动画 reference 动画 的 名称

演唱（姓名，音乐）

Foreign key 姓名 reference 人物 的 姓名

Foreign key 音乐 reference 音乐 的 名称

## 2. 基本表设计

基本表设计如表 3-1~3-10 所示。

表 3-1：原作公司的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
名称	nvarchar(30)	否			主码	
地址	nvarchar(60)	是				
成立时间	nvarchar(30)	是				
主要业务	nvarchar(80)	是				
代表作	nvarchar(80)	是				

表 3-2：动画公司的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
名称	nvarchar(30)	否			主码	
地址	nvarchar(60)	是				
成立时间	nvarchar(30)	是				
代表作	nvarchar(80)	是				

表 3-3：人物的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
姓名	nvarchar(30)	否			主码	
性别	nvarchar(6)	是	“男”或“女” 或为空			若人物为组合等 多人团体，则性 别为空
出生日期	nvarchar(30)	是				
职业	nvarchar(20)	否				
代表作	nvarchar(80)	是				

表 3-4: staff 的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
姓名	nvarchar(30)	否			主码	
代表作	nvarchar(80)	是				

表 3-5: 动画的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
名称	nvarchar(40)	否			主码	
制作类型	nvarchar(6)	否	“新作”或“续作”或“衍生”或“重制”			
改编类型	nvarchar(8)	否	“漫改”或“轻改”或“游戏改”或“原创”			
原作公司	nvarchar(30)	是	若改编类型为“原创”，则为空，否则，不为空		外键	来自于原作公司的名称
原作者	nvarchar(30)	是	若改编类型为“原创”，则为空，若改编类型为“漫改”或“轻改”，则不为空		外键	来自于人物的姓名
制作公司	nvarchar(30)	否			外键	来自于动画公司的名称
新番季度	nvarchar(10)	否	格式为 4 位年份+“年”+“春”或“夏”或“秋”或“冬”			
首播时间	date	否				
总集数	smallint	否	大于等于 0	0		



表 3-6: 动画类别的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
名称	nvarchar(40)	否			主码、外码	来自于动画的名称
类别	nvarchar(10)	否			主码	

表 3-7: 制作的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
姓名	nvarchar(30)	否			主码、外码	来自于 staff 的姓名
动画	nvarchar(40)	否			主码、外码	来自于动画的名称
职位	nvarchar(30)	否	种类有”导演”、”编剧”、”音乐”、”动画人设”，或四种种类叠加，中间用”、”分隔			

表 3-8: cast 的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
姓名	nvarchar(30)	否			主码、外码	来自人物的姓名
动画	nvarchar(40)	否			主码、外码	来自动画的名称
角色	nvarchar(40)	否				

表 3-9：音乐的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
名称	nvarchar(60)	否			主码	
类型	nvarchar(5)	否	“OP” 或”ED”			
动画	nvarchar(40)	否			外码	来自动画的 名称

表 3-10：用户信息表的设计

属性名	数据类型	是否可空	列约束	默认值	键	解释
姓名	nvarchar(30)	否			主码、外码	来自人物的 姓名
音乐	nvarchar(60)	否			主码、外码	来自音乐的 名称

## 四. 数据库物理设计和实施

### 1. 数据库的创建

使用 MS SQL Server Express 建立动画信息管理系统的数据库，数据库基本信息如下：

表 4-1：动画信息管理系统的数据库参数表

选项		参数
数据库名称		Animation_MS
数据文件	逻辑文件名	动画信息管理系统
	物理文件名	动画信息管理系统.mdf
	初始容量	72MB
	最大容量	无限制
	增长量	64MB
日志文件	逻辑文件名	动画信息管理系统_log
	物理文件名	动画信息管理系统_log.ldf
	初始容量	8MB
	最大容量	2097152MB
	增长量	64MB

## 2. 创建基本表

（给出创建每个基本表的 SQL 语句，应有如下实现：表名、列名、主键、外键、参照完整性约束，对每个列，应有数据类型和长度、默认、是否可空等信息）

--创建原作公司表

```
create table 原作公司
(
    名称 nvarchar(30) primary key,
    地址 nvarchar(60),
    成立时间 nvarchar(30),
    主要业务 nvarchar(80),
    代表作 nvarchar(80)
);
```

--创建动画公司表

```
create table 动画公司
(
    名称 nvarchar(30) primary key,
    地址 nvarchar(60),
    成立时间 nvarchar(30),
    代表作 nvarchar(80)
);
```

--创建人物表

```
create table 人物
(
    姓名 nvarchar(30) primary key,
    性别 nvarchar(6) check(性别='男' or 性别='女'),
    出生日期 nvarchar(30),
```

```
职业 nvarchar(20) not null,  
代表作 nvarchar(80)  
);
```

--创建 staff 表

```
create table staff  
(  
姓名 nvarchar(30) primary key,  
代表作 nvarchar(80)  
);
```

--创建动画表

```
create table 动画  
(  
名称 nvarchar(40) primary key,  
制作类型 nvarchar(6) check(制作类型 in ('新作', '续作', '衍生', '重制')),  
改编类型 nvarchar(8) check(制作类型 in ('漫改', '轻改', '游戏改', '原创')),  
原作公司 nvarchar(30),  
原作者 nvarchar(30),  
制作公司 nvarchar(30) not null,  
新番季度 nvarchar(10) not null check(新番季度 like [0-9][0-9][0-9][0-9]年[春夏秋冬]),  
首播时间 date not null,  
总集数 smallint not null default 0 check(总集数>=0),  
foreign key (原作公司) references 原作公司(名称),  
foreign key (原作者) references 人物(姓名),  
foreign key (制作公司) references 动画公司(名称)  
);
```

--动画表改编类型的 check 约束

alter table 动画

add constraint adapted\_type check

(case when 改编类型 = '原创' then

case when 原作公司 is null and 原作者 is null then 1 else 0 end

when 改编类型 = '游戏改' then

case when 原作公司 is not null then 1 else 0 end

else

case when 原作公司 is not null and 原作者 is not null then 1 else 0 end

end = 1);

--创建动画类别表

create table 动画类别

(

名称 nvarchar(40),

类别 nvarchar(10),

primary key (名称, 类别),

foreign key (名称) references 动画(名称)

);

--创建制作表

create table 制作

(

姓名 nvarchar(30),

动画 nvarchar(40),

职位 nvarchar(30) not null check(职位 in ('导演', '编剧', '音乐', '动画人设', '导演、  
编剧', '导演、音乐', '导演、动画人设', '编剧、音乐', '编剧、动画人设', '音乐、动  
画人设', '导演、编剧、音乐', '导演、编剧、动画人设', '导演、音乐、动画人设', '  
编剧、音乐、动画人设', '导演、编剧、音乐、动画人设'))),

```
primary key (姓名, 动画),  
foreign key (姓名) references staff(姓名),  
foreign key (动画) references 动画(名称)  
);
```

--创建 cast 表

```
create table cast  
(  
    姓名 nvarchar(30),  
    动画 nvarchar(40),  
    角色 nvarchar(40) not null,  
    primary key (姓名, 动画),  
    foreign key (姓名) references 人物(姓名),  
    foreign key (动画) references 动画(名称)  
);
```

--创建音乐表

```
create table 音乐  
(  
    名称 nvarchar(60) primary key,  
    类型 nvarchar(5) not null check(类型='OP' or 类型='ED'),  
    动画 nvarchar(40) not null,  
    foreign key (动画) references 动画(名称)  
);
```

--创建演唱表

```
create table 演唱  
(
```

```
姓名 nvarchar(30),
音乐 nvarchar(60),
primary key(姓名, 音乐),
foreign key (姓名) references 人物(姓名),
foreign key (音乐) references 音乐(名称)
);
```

### 3. 触发器设计

（如有触发器定义，给出定义 SQL 语句和功能说明）

功能：删除动画时级联删除动画类别、cast、演唱、音乐（制作由于跟 staff 相连且关系复杂，需要手动先删除掉动画制作组）

SQL 语句：

```
create trigger 删除动画时的级联删除
on 动画
instead of delete
as
begin
    declare @动画 nvarchar(40)
    set @动画 = (select 名称 from deleted)
    delete from 动画类别 where 名称=@动画
    delete from cast where 动画=@动画
    delete from 演唱 where exists
        (select 名称 from 音乐 where 演唱.音乐 = 音乐.名称 and 动画=@动画)
    delete from 音乐 where 动画=@动画
    delete from 动画 where 名称=@动画
end
go
```



#### 4. 存储过程设计

（如有存储过程定义，给出定义 SQL 语句和功能说明）

功能：删除音乐时级联删除演唱

SQL 语句：

```
create procedure pr_删除音乐时级联删除演唱
    @音乐 nvarchar(60)
as
begin
    delete from 演唱 where 音乐=@音乐
    delete from 音乐 where 名称=@音乐
end
go
```

#### 5. 视图设计

（如有视图定义，给出定义 SQL 语句和功能说明）

功能：动画基本信息查询，便于在动画所有相关信息查询时显示

SQL 语句：

```
create view 动画基本信息
as
(select 动画.名称,动画.制作类型,动画.改编类型,动画.原作公司,原作公司.代表作
as 原作公司代表作,动画.原作者,
    人物.代表作 as 原作者代表作,动画.制作公司,动画公司.代表作 as 制作公
司代表作,动画.新番季度,动画.首播时间,动画.总集数
```

```
from 动画,原作公司,人物,动画公司
where 动画.原作公司 = 原作公司.名称 and 动画.原作者 = 人物.姓名 and 动画.制作公司 = 动画公司.名称)
union
(select 动画.名称,动画.制作类型,动画.改编类型,动画.原作公司,原作公司.代表作
as 原作公司代表作,动画.原作者,
    null as 原作者代表作,动画.制作公司,动画公司.代表作 as 制作公司代表作,
    动画.新番季度,动画.首播时间,动画.总集数
from 动画,原作公司,动画公司
where 动画.原作公司 = 原作公司.名称 and 动画.原作者 is null and 动画.制作公司 = 动画公司.名称)
union
(select 动画.名称,动画.制作类型,动画.改编类型,动画.原作公司,null as 原作公司代表作,动画.原作者,
    null as 原作者代表作,动画.制作公司,动画公司.代表作 as 制作公司代表作,
    动画.新番季度,动画.首播时间,动画.总集数
from 动画,动画公司
where 动画.改编类型 = '原创' and 动画.制作公司 = 动画公司.名称)
```

功能：查询得到每个季度番剧总数量以及各种不同制作类型和改编类型的动画数量

SQL 语句：

```
create view 每季度番剧数量
as
with type1 as
    (select 新番季度,count(*) as 新作数量 from 动画 where 制作类型 = '新作'
group by 新番季度),
type2 as
```

```
(select 新番季度,count(*) as 续作数量 from 动画 where 制作类型 = '续作'
group by 新番季度),
type3 as
(select 新番季度,count(*) as 衍生数量 from 动画 where 制作类型 = '衍生'
group by 新番季度),
type4 as
(select 新番季度,count(*) as 重制数量 from 动画 where 制作类型 = '重制'
group by 新番季度),
type5 as
(select 新番季度,count(*) as 漫改数量 from 动画 where 改编类型 = '漫改'
group by 新番季度),
type6 as
(select 新番季度,count(*) as 轻改数量 from 动画 where 改编类型 = '轻改'
group by 新番季度),
type7 as
(select 新番季度,count(*) as 游戏改数量 from 动画 where 改编类型 = '游戏改'
group by 新番季度),
type8 as
(select 新番季度,count(*) as 原创数量 from 动画 where 改编类型 = '原创'
group by 新番季度),
total as
(select type1.新番季度,
case when type1.新作数量 is not null then type1.新作数量 else 0 end as
新作数量,
case when type2.续作数量 is not null then type2.续作数量 else 0 end as
续作数量,
case when type3.衍生数量 is not null then type3.衍生数量 else 0 end as
衍生数量,
case when type4.重制数量 is not null then type4.重制数量 else 0 end as
```

重制数量,

case when type5.漫改数量 is not null then type5.漫改数量 else 0 end as  
漫改数量,

case when type6.轻改数量 is not null then type6.轻改数量 else 0 end as  
轻改数量,

case when type7.游戏改数量 is not null then type7.游戏改数量 else 0 end  
as 游戏改数量,

case when type8.原创数量 is not null then type8.原创数量 else 0 end as  
原创数量

from type1 full outer join type2 on type1.新番季度 = type2.新番季度

full outer join type3 on type1.新番季度 = type3.新番季度

full outer join type4 on type1.新番季度 = type4.新番季度

full outer join type5 on type1.新番季度 = type5.新番季度

full outer join type6 on type1.新番季度 = type6.新番季度

full outer join type7 on type1.新番季度 = type7.新番季度

full outer join type8 on type1.新番季度 = type8.新番季度)

select 新番季度,(新作数量+续作数量+衍生数量+重制数量) as 番剧数量,新作数  
量,续作数量,衍生数量,重制数量,

漫改数量,轻改数量,游戏改数量,原创数量

from total

功能：查询给一部动画配音的所有声优及其代表作

SQL 语句：

create view 动画配音列表

as

select cast.动画,cast.角色,cast.姓名,人物.代表作

from cast,人物

where cast.姓名 = 人物.姓名

功能：查询一部动画的所有音乐及音乐各自的歌手，编写自定义函数来将一首音乐对应的歌手都连接成一个字符串，用”、”隔开

SQL 语句：

```
create function getsinger(@音乐 nvarchar(60))
```

```
returns @音乐及歌手 table
```

```
(
```

```
    歌手 nvarchar(255)
```

```
)
```

```
begin
```

```
    insert @音乐及歌手
```

```
        select 姓名 as 歌手
```

```
        from 演唱
```

```
        where 音乐 = @音乐
```

```
    return
```

```
end
```

```
go
```

```
create view 动画音乐列表
```

```
as
```

```
select 音乐.动画,音乐.类型,音乐.名称,stuff((
```

```
    select '&'+歌手
```

```
    from getsinger(音乐.名称)
```

```
    for xml path("")),1,1,") as 歌手
```

```
from 音乐
```

功能：查询得到一部动画的制作组工作人员以及他们的代表作

SQL 语句：

```
create view 动画制作组列表
as
select 制作.动画,制作.职位,制作.姓名,staff.代表作
from 制作,staff
where 制作.姓名 = staff.姓名
```

功能，计算得到一个带年份的动画表，便于根据新番年份进行查询

SQL 语句：

```
create view 带年份的动画表
as
select *,left(动画.新番季度,4) as 年份
from 动画
```

功能：将所有新番年份和新番季度合并到一个列表中，便于前端下拉列表框的选择放送限制时间

SQL 语句：

```
create view 限制时间
as
(select distinct 新番季度 as 限制时间
from 带年份的动画表)
union
(select distinct 年份 as 限制时间
from 带年份的动画表)
```

功能：记录所有季番信息

SQL 语句：

```
create view 季番表
as
select *
from 带年份的动画表
where 总集数 <= 14
```

功能：记录所有半年番信息

SQL 语句：

```
create view 半年番表
as
select *
from 带年份的动画表
where 总集数 > 14 and 总集数 <= 26
```

功能：记录所有长篇动画信息

SQL 语句：

```
create view 长篇表
as
select *
from 带年份的动画表
where 总集数 > 26
```

功能：计算获得数据库中制作组职位为导演的工作人员，其代表作，以及对应制作的动画信息

SQL 语句：

```
create view 制作组导演表
```

```
as
```

```
select 姓名,代表作,动画制作组列表.动画,制作类型,改编类型,原作公司,原作者,  
制作公司,新番季度,首播时间,总集数,年份
```

```
from 动画制作组列表,带年份的动画表
```

```
where 职位 like '%导演%' and 动画制作组列表.动画 = 带年份的动画表.名称
```

功能：计算获得数据库中制作组职位为编剧的工作人员，其代表作，以及对应制作的动画信息

SQL 语句：

```
create view 制作组编剧表
```

```
as
```

```
select 姓名,代表作,动画制作组列表.动画,制作类型,改编类型,原作公司,原作者,  
制作公司,新番季度,首播时间,总集数,年份
```

```
from 动画制作组列表,带年份的动画表
```

```
where 职位 like '%编剧%' and 动画制作组列表.动画 = 带年份的动画表.名称
```

功能：计算获得数据库中制作组职位为音乐的工作人员，其代表作，以及对应制作的动画信息

SQL 语句：

```
create view 制作组音乐表
```

```
as
```

```
select 姓名,代表作,动画制作组列表.动画,制作类型,改编类型,原作公司,原作者,  
制作公司,新番季度,首播时间,总集数,年份
```



from 动画制作组列表,带年份的动画表  
where 职位 like '%音乐%' and 动画制作组列表.动画 = 带年份的动画表.名称

功能：计算获得数据库中制作组职位为动画人设的工作人员，其代表作，以及对应制作的动画信息

SQL 语句：

```
create view 制作组动画人设表
as
select 姓名,代表作,动画制作组列表.动画,制作类型,改编类型,原作公司,原作者,
制作公司,新番季度,首播时间,总集数,年份
from 动画制作组列表,带年份的动画表
where 职位 like '%动画人设%' and 动画制作组列表.动画 = 带年份的动画表.名称
```

功能：记录声优配过的动画、角色及动画信息

SQL 语句：

```
create view 声优配过的动画及信息
as
select 姓名,角色,名称,制作类型,改编类型,原作公司,原作者,制作公司,新番季度,
首播时间,总集数,年份
from 动画配音列表,带年份的动画表
where 动画配音列表.动画 = 带年份的动画表.名称
```

功能：记录歌手演唱过的动画及动画信息

SQL 语句：

```
create view 歌手演唱动画信息
as
select distinct 演唱.姓名,带年份的动画表.名称,制作类型,改编类型,原作公司,原作者,制作公司,新番季度,首播时间,总集数,年份
from 音乐,演唱,带年份的动画表
where 音乐.名称 = 演唱.音乐 and 音乐.动画 = 带年份的动画表.名称
```

功能：按照类别查询一个动画，并得到这个动画的所有类别

SQL 语句：

```
create view 类别动画类别
as
select a1.类别 as 类别 1,a1.名称,a2.类别 as 类别 2
from 动画类别 as a1,动画类别 as a2
where a1.名称 = a2.名称
```

功能：记录原作公司被动画化作品及其信息，同时记录原作公司的信息

SQL 语句：

```
create view 原作公司被动画化作品
as
select 原作公司.名称,地址,成立时间,主要业务,代表作,带年份的动画表.名称 as 动画,改编类型,新番季度,年份
from 原作公司,带年份的动画表
where 原作公司.名称 = 带年份的动画表.原作公司
```

功能：记录动画公司制作的动画作品及其信息，同时记录动画公司的信息

SQL 语句：

```
create view 动画公司制作作品
as
select 动画公司.名称,地址,成立时间,代表作,带年份的动画表.名称 as 动画,新番
季度,年份
from 动画公司,带年份的动画表
where 动画公司.名称 = 带年份的动画表.制作公司
```

功能：记录原作者被动画化的作品及其信息，同时记录原作者的信息

SQL 语句：

```
create view 原作者参与动画
as
select 人物.姓名,性别,出生日期,代表作,改编类型,新番季度,count(*) as 参与数量
from 动画,人物
where 动画.原作者 = 人物.姓名
group by 人物.姓名,性别,出生日期,代表作,改编类型,新番季度
```

功能：记录声优参与配音的动画作品及其信息，同时记录声优的信息

SQL 语句：

```
create view 声优参与动画
as
select 人物.姓名,性别,出生日期,代表作,新番季度,count(*) as 参与数量
from 动画,人物,cast
where 动画.名称 = cast.动画 and cast.姓名 = 人物.姓名
group by 人物.姓名,性别,出生日期,代表作,新番季度
```

功能：记录歌手/组合/乐队参与演唱音乐对应的动画作品及其信息，同时记录歌手/组合/乐队的信息

SQL 语句：

```
create view 歌手组合参与动画
as
select 人物.姓名,性别,出生日期,代表作,新番季度,count(*) as 参与数量
from 动画,人物,音乐,演唱
where 音乐.名称 = 演唱.音乐 and 演唱.姓名 = 人物.姓名 and 音乐.动画 =
动画.名称
group by 人物.姓名,性别,出生日期,代表作,新番季度
```

功能：记录 staff 参与制作的动画作品及其信息，同时记录 staff 的信息

SQL 语句：

```
create view staff 参与动画
as
select 姓名,代表作,动画,职位,新番季度,年份
from 动画制作组列表,带年份的动画表
where 动画制作组列表.动画 = 带年份的动画表.名称
```

功能：记录每个季度新番的动画音乐

SQL 语句：

```
create view 季度新番动画音乐
as
select 动画音乐列表.名称,类型,动画,歌手,新番季度
from 动画音乐列表,动画
where 动画音乐列表.动画 = 动画.名称
```

功能：记录歌手演唱过的所有音乐出自的动画信息

SQL 语句：

```
create view 演唱动画音乐
```

```
as
```

```
select 演唱.姓名,动画音乐列表.名称,类型,动画,歌手,新番季度,年份
```

```
from 动画音乐列表,演唱,带年份的动画表
```

```
where 动画音乐列表.动画 = 带年份的动画表.名称 and 演唱.音乐 = 动画音乐  
列表.名称
```

## 6. 其他数据库对象的设计

（如有设计安全机制、定义索引和维护规则等，请另起小结进行说明）

## 五. 应用程序设计

### 1. 开发及运行环境介绍

系统使用的开发工具是 Wxpython

### 2. 主要功能设计

（简要说明主要的功能实现，可给出关键代码，辅以流程图进行说明等）

- ① 在登录页面，输入登录信息进行登录，普通用户无需输入密码，只要输入用户名即可，管理员则需输入相应的账号名和密码。（输入不符合条件时会跳出错误信息提示框）

管理账号：QTH

密码：animation

- ② 登录菜单栏中可以切换普通用户/管理员两种登录方式，关于菜单栏中可以查看作者信息。
- ③ 登录之后进入主界面，工具栏中共有 9 项，前 6 项是查询信息主体，如动画查询，就是根据输入或选择的限制条件查询符合的动画信息。第 7 项是管理数据，可以对系统内数据进行增删改，当登录方式为普通用户登录时，这个工具栏为灰化不可点击状态。第 8 项为返回登录界面，可以返回登录界面以选择其他登录方式。第 9 项为退出系统，可以直接退出整个系统。对于第 8 项、第 9 项以及直接点击右上方 X 按钮，都有二次确认提示框跳出。
- ④ 在每项菜单栏的左侧列表框中，可以选择不同的查询方式，如“动画查询”->“制作类型”可以根据制作类型，选择相应的制作类型及放送时间限制，查询得到所有符合条件的动画及基本信息。具体查询功能可见本报告上文中对系统功能的描述，或查看实际系统。
- ⑤ 在管理数据中，可以对数据库内 10 个表进行增删改操作，在对应文本框内输出或在对应下拉列表中选择数据，点击添加按钮进行添加。进行修

改操作时，需要先进行查询，再在查询结果的基础上进行修改，点击修改按钮。根据查询得到的信息，可以在下方选择相应的数据，点击删除按钮进行删除。当添加、查询、修改、删除成功或失败时，系统都会跳出对应的提示信息，如“添加成功”、“删除失败”。

- ⑥ 特殊情况：本系统将 `staff` 表与制作表一起进行增删改操作，不在 `staff` 表中保留无用信息，即数据库中每个 `staff` 必须参与数据库中至少一部动画的制作，若当删除或修改制作表时，存在某个 `staff` 在制作表中没有制作过任何动画，那么就在 `staff` 表中删除该 `staff` 的基本信息。
- ⑦ 管理数据界面中，带\*的为必填内容，部分输入限制条件及要求写在界面输入框的右侧。
- ⑧ 考虑到用户可能不知道完整的信息，如用户知道 `A1` 这个动画公司，想进行查询，在动画公司名称里输入“`A1`”，但却没有得到结果，这是因为这个动画公司的正式全名为 `A-1 Pictures`，数据库内存储的也是这个名称，而“`A1`”无法匹配。因此在部分查询、添加、修改、删除功能中，我设置了下拉框供用户进行选择，而不是提供空白的文本框让用户输入，这样可以减少用户查询不到结果的可能性。

### 3. 主要界面

登录界面如图 5-1 所示。



The screenshot shows a window titled "动画信息管理系统" (Animation Information Management System). The window has a menu bar with "登录(L)" (Login) and "关于(B)" (About). The main content area displays the title "动画信息管理系统" in large characters. Below the title, there are two input fields: "用户名:" (Username) and "密码:" (Password). To the right of the password field is a "确定" (Confirm) button. Below the password field, there is a note: "【注】普通用户登录无需输入密码" (Note: Ordinary users do not need to enter a password to log in).

图 5-1 登陆界面



查询界面如图 5-2、5-3 所示，图 5-2 为点击工具栏后显示页面，图 5-3 为在左侧列表框选择对应查询方式之后的页面。



图 5-2 选择工具栏中查询方式后界面（普通用户登录）



图 5-3 选择左侧列表框中查询方式后界面（普通用户登录）

管理数据界面如图 5-4、5-5 所示。

动画信息管理系统

动画查询

原作公司查询

动画公司查询

人物查询

staff查询

音乐查询

管理数据

返回登录界面

退出系统

原作公司

动画公司

人物

动画(添加)

动画(删改)

动画类别

动画制作组

动画声优

动画音乐

音乐演唱

请输入想要添加的数据:

【注】带\*为必填项, 修改操作要求先进行查询, 再在查询结果基础上进行修改

名称:

\*

地址:

成立时间:

主要业务:

代表作:

添加

请选择原作公司名称:

名称:

\*

地址:

成立时间:

主要业务:

代表作:

查询

修改

请选择原作公司名称:

名称:

\*

删除

图 5-4 管理原作公司表数据（管理员登录）

动画信息管理系统

动画查询

原作公司查询

动画公司查询

人物查询

staff查询

音乐查询

管理数据

返回登录界面

退出系统

原作公司

动画公司

人物

动画(添加)

动画(删改)

动画类别

动画制作组

动画声优

动画音乐

音乐演唱

请选择动画名称:

【注】带\*为必填项, 修改操作要求先进行查询, 再在查询结果基础上进行修改

名称:

\*

制作类型:

\*

"新作"或"续作"或"衍生"或"重制"

改编类型:

\*

"漫改"或"轻改"或"游戏改"或"原创"

原作公司:

原作者:

制作公司:

\*

新番季度:

\*

格式为4位年份+'年'+ '春'/'夏'/'秋'/'冬'

首播时间:

\*

时间格式为年月日, 如: 20180101

总集数:

\*

填入数字, 未和时填入0

查询

修改

请选择动画名称:

名称:

\*

删除

请先确保删除相应的动画制作组

图 5-5 管理动画表数据（修改和删除）（管理员登录）

40

查询示例如图 5-6 所示。



图 5-6 查询动画基本信息（管理员登录）

修改示例如图 5-7 所示。

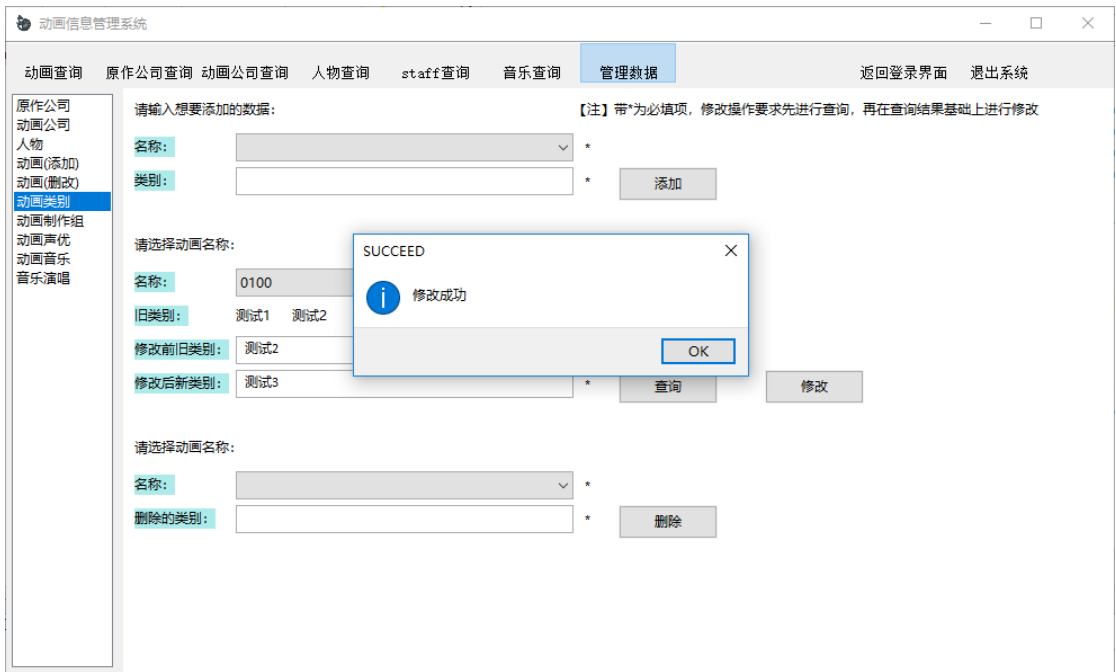


图 5-7 修改动画“0100”的旧类别“测试2”为“测试3”（管理员登录）

## 六. 心得体会

（包括课程设计中遇到的困难或问题、解决方法，对于数据库理论以及数据库应用开发的收获和体会等。）

首先在选题方面上是我的个人兴趣，在设计 ER 图、关系模式、数据库功能的时候感到很开心，想要实现很多功能，考虑到很多情况，根据条件给用户推荐动画等等，用户还可以进行匿名反馈，用户知道的消息可能比管理员知道的多，用户将增删改信息反馈给管理员，管理员确定无误后进行增删改操作，进一步保证了数据的准确和全面。

因为想要制作一个真正实用的系统，我想要搜集所有真实信息，真正的实现动画信息管理和查询推荐。在搜集数据的过程中，我也对我喜欢的东西有了更多的了解，知道了以前不知道的动画公司、原作公司，时不时有“啊原来这部动画是这个公司做的啊？！”、“原来这个角色是他配的？！”、“原来这首音乐是她唱的？！”这样的惊喜收获。

但是遇到了严峻的问题，就是这样的真实信息搜集实在是太费时间，为了查到有些信息，我要花很多时间去浏览很多页面，实在查不到的我才会置为空，以及我本来想要搜集尽可能多的动画信息（初步设想是 16 年、17 年、18 年上半年所有新番动画），做一个非常有用全面的系统，并且加入一些我喜欢的、推荐的旧番（16 年之前的），但是搜集信息过于花费时间，并且为了保证查询的多样性（总不能每个查询都只有一种结果、每个对应都是一对一吧），我先是将范围缩减到 20 个知名动画公司的近三年所有动画，又缩减到 8 个知名动画公司的近三年所有动画，又缩减到这 8 家公司 2018 年的所有动画，并且对声优也做了限制（只取主要角色和知名声优配音的角色）。我不能再在搜集数据上花费时间了。（为了节约时间、也对原先的 ER 图做了简化）

因为不熟悉 C#、HTML、PHP，我选择了自己熟悉的 PYTHON，本想用 QT 设计+Pyqt 实现前端功能，但折腾了半天也没弄出个基本来，就还是选择了熟悉的 Wxpython，不过后来事实证明，使用 Wxpython 也不是一件容易的事，为了实现我想要的功能（窗口切换、面板分隔、面板内置滚动条、面板内表格等），我做了很多尝试和修改，有些功能实现了，有些没实现放弃了，以及想要做的匿名

反馈也没有足够的时间去实现，只能在报告中写一下自己的设想了。（用户可以帮助数据库数据完善，管理员可以减少工作量，也一定程度上弥补了管理员信息的不足或错误，感觉是个很好的功能的 QAQ）

使用 C#的同学都说简单快捷，使用 HTML+PHP 的同学都说好麻烦好多页面要写好多 BUG，我本以为 Wxpython 会简单一些，但是写完了觉得跟使用 HTML+PHP 的同学的麻烦程度差不多 QAQ。

还有就是这学期实在是作业量太多，没有足够的时间，从第一次检查换到第二次，又拖到第三次，真的非常对不起老师 QAQ，感觉很麻烦老师，以及因为来不及写完报告，最后作业也迟交了 QAQ。真的是非常抱歉!!!

最后感谢一下前面报告里提到的那位 B 站 up 主——長門有 C，感谢他之前的那些新番介绍视频！还有感谢帅气可爱的徐飞老师，这一学期辛苦啦！