

****

**数据库系统实现总结报告**

**王天博18373520、肖宜松18373494、袁劭涵18373488**

**2020年10月**

目录

[概述 1](#_Toc40377030)

[数据库设计部分 1](#_Toc40377030)

[一数据库需求分析 1](#_Toc40377042)

[（一）基本信息分析 1](#_Toc40377031)

[（二）系统需求分析 2](#_Toc40377031)

[（三）数据元素表分析 3](#_Toc40377031)

[（四）数据流图分析 9](#_Toc40377031)

[二数据库概念设计 11](#_Toc40377047)

[（一）实体设计 11](#_Toc40377031)

[（二）实体间关系设计 13](#_Toc40377031)

[（三）总体E-R图设计 17](#_Toc40377031)

[三数据库逻辑设计 17](#_Toc40377057)

[（一）关系模式分析 17](#_Toc40377031)

[（二）函数依赖分析 1](#_Toc40377031)8

[（三）范式分析 20](#_Toc40377031)

[数据库实现部分 1](#_Toc40377030)

[四数据库系统功能结构图 1](#_Toc40377042)

[（一）基本信息分析 1](#_Toc40377031)

[（二）系统需求分析 2](#_Toc40377031)

[（三）数据元素表分析 3](#_Toc40377031)

[（四）数据流图分析 9](#_Toc40377031)

[五基本表设计及相关设计说明 11](#_Toc40377047)

[（一）实体设计 11](#_Toc40377031)

[（二）实体间关系设计 13](#_Toc40377031)

[（三）总体E-R图设计 17](#_Toc40377031)

[六数据库代码实现说明 17](#_Toc40377057)

[（一）关系模式分析 17](#_Toc40377031)

[（二）函数依赖分析 1](#_Toc40377031)8

[（三）范式分析 20](#_Toc40377031)

[七结果展示 17](#_Toc40377057)

[（一）关系模式分析 17](#_Toc40377031)

[（二）函数依赖分析 1](#_Toc40377031)8

[总结与收获 1](#_Toc40377030)

概述

经过讨论，本小组决定此次数据库实验大作业的数据库设计方向为基于教务系统的校园关系数据库，主要的面向用户对象为学生和老师。为了尽可能模拟真实的学生校园生活，本小组在一般教务系统的基础之上增加了宿舍楼、教学楼、教室、社团等实体，增加学生与学生、学生与教师之间，甚至是学生与职工之间的联系，从而丰富了校园关系网，更为贴近学生的校园生活。

本小组选定该主题的原因包括如下几条：

① 相比于其他的数据关系网，如工厂、商场等，我们更为熟知学校关系网，能够较好地实现和复现。

② 在随机生成数据库数据的同时能够使生成的数据更加符合实际情况。

③ 校园关系网复杂度适中，基于校园关系网建立的数据库更能充分体现出数据库的各种功能及其作用。

基于上述因素，我们从数据库需求分析、概念设计和逻辑设计三方面分析本小组的数据库设计。

数据库设计部分

一 数据库需求分析

（一）基本信息分析

（1）学校中有多个系，描述系的信息有：系编号、系名，每个系只能有一个系主任。

（2）学校中有很多学生，描述学生的信息有：学号、姓名、性别、籍贯和年龄。

（3）学校中有很多教师，描述教师的信息有：教师编号、姓名、性别、年龄、籍贯、邮箱、教龄、职称等，每个教师只属于一个系，一个系有多个教师。

（4）学校中有很多班级，描述班级的信息有：班级编号、名称等，每个班有且仅有一个班主任和辅导员，每名老师只能担任一个班级的班主任，每名辅导员可以担任多个班级的辅导员。

（5）学校开设了很多课程，描述课程的信息有：课程编号、课程名称、学分等。每个院系可以开设多门课程，每门课程只能由一个院系开设。

（6）学校中有很多职工，描述职工的信息有：职工编号、姓名、性别、年龄、籍贯和工龄。

（7）学校中有很多社团，描述社团的信息有：名称、类型和星级等。每个社团有唯一一名指导老师。

（8）学校中藏有很多书籍，描述书籍的信息有：图书编号、书名、作者、出版社、出版年份和借阅状态。

（9）学校中有多个教学楼，描述教学楼的信息有：名称、所处区域、总楼层、建设年份。

（10）学校中有多个教室，描述教室的信息有：教室编号、楼层、类型等。每个教室只位于一个教学楼，一个教学楼包含多个教室。

（11）学校中有多个宿舍楼，描述宿舍楼的信息有：宿舍楼号、总楼层、寝室容量、所处区域、建设年份等。

（12）学校中有多个寝室，描述寝室的信息有：寝室号、楼层、朝向等。每个宿舍只能位于一栋宿舍楼，每个宿舍楼包含多个宿舍；每个宿舍只有一名宿舍长，每名学生只能担任一个宿舍的宿舍长。

（13）一个学生可以参加多个社团，每个社团可以有多名学生参加，学生在社团之中可以担任不同的职务。

（14）一个学生可以借阅多本图书，一本图书可以被多名同学借阅，但同一本书在同一时间只能被一名同学借阅，学生借阅图书会有借阅时间及借阅状态记录。

（15）一个学生只能属于一个班级，一个班级可以有多个学生，学生在班级中可以担任不同的职务。

（16）一个学生可以选修多门课程，一门课程可以被多个学生选修，当一个学生选定一门课时，就有一个确定的老师，修完该门课后会有确定的成绩。

（17）一个学生住在一间寝室，一间寝室可以有多名学生住宿，每名学生有自己对应的床位号。

（18）一名教师可以教授多门课程，一门课程可以由多名老师教授，每个老师教授的课程都有自己的上课地点和上课时间。

（19）一名职工可以管理多个教学楼和宿舍楼，一栋教学楼和宿舍楼可以有多名职工管理。

（二）系统需求分析

依据上述基本信息，结合本数据库的设计目标，我们将数据库的用户种类设计为如下几种，并针对不同种类的用户设计不同的用户需求：

（1）超级管理员

需要实现对用户表的管理，包括增加、删除用户，修改用户身份、密码等。

（2）图书管理员

需要实现对图书系统的管理，包括增删查改图书信息，以及学生借阅图书的信息。

（3）系管理员

需要实现对教务信息，学生、教师的个人信息及教学楼、教室等信息的管理，包括学生、老师个人信息的增删查改，学生选课、老师授课信息的增删查改，系、班级信息的增删查改等等。

（4）其他管理员

需要实现对除教学信息以外的其他信息的管理，包括宿舍楼寝室、社团信息管理，学生参加社团、住宿信息管理，以及职工的个人信息及职工对宿舍楼、教学楼责任信息的管理。

（3）学生

实现对学生个人信息的查询，如身份信息，系、班级、选课信息的查询；实现对授课老师的信息查找；除此之外还可以根据个人需求实现部分对其他学生信息的查询。

（4）老师

实现对老师个人信息的查询，如身份信息，系、班级、授课信息的查询;除此之外可以查看选修自己课程的学生的成绩及班级均分等。

（三）数据元素表分析

针对上面的功能需求，结合实际得出本系统要处理的数据元素表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **社团实体（Association）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **名称** | varchar | 50 | primary key |
| **类型** | varchar | 50 | notnull |
| **星级** | varchar | 50 | not null |
| **指导老师** | varchar | 50 | null |

表1.1.1社团数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **图书实体（Book）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **图书编号** | varchar | 50 | primary key |
| **书名** | varchar | 50 | not null |
| **作者** | varchar | 50 | notnull |
| **出版社** | varchar | 50 | null |
| **出版年份** | Int | 4 | null |
| **借阅状态** | varchar | 50 | not null |

表1.1.2 图书数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **班级实体（Class）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **班级编号** | varchar | 50 | primary key |
| **系编号** | varchar | 50 | foreignkey |
| **班主任** | varchar | 50 | foreignkey |
| **辅导员** | varchar | 50 | foreignkey |

表1.1.3班级数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **教室实体（Classroom）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **教室编号** | varchar | 50 | primary key |
| **教学楼名称** | varchar | 50 | foreignkey |
| **楼层** | int | 4 | not null |
| **类型** | varchar | 50 | null |

表1.1.4教室数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **课程实体（Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **课程编号** | varchar | 50 | primary key |
| **名称** | varchar | 50 | not null |
| **学分** | int | 4 | not null |
| **开设院系** | varchar | 50 | foreignkey |

表1.1.5课程数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **系实体（Department）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **系编号** | varchar | 50 | primary key |
| **系名** | varchar | 50 | not null |
| **系主任** | varchar | 50 | foreignkey |

表1.1.6 系数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **寝室实体（Dormitory）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **寝室号** | varchar | 50 | primary key |
| **宿舍楼号** | varchar | 50 | notnull |
| **楼层** | int | 4 | null |
| **朝向** | varchar | 50 | null |
| **宿舍长** | varchar | 50 | foreignkey |

表1.1.7 寝室数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **宿舍楼实体（DormitoryBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **楼号** | varchar | 50 | primary key |
| **总楼层** | Int | 4 | null |
| **所处区域** | varchar | 50 | not null |
| **建设年份** | varchar | 50 | not null |
| **寝室容量** | int | 4 | null |
| **类型** | varchar | 50 | not null |

表1.1.8 宿舍楼数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **职工实体（ClassRoom）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **职工编号** | varchar | 50 | primary key |
| **姓名** | varchar | 50 | not nul |
| **性别** | varchar | 50 | not nul |
| **年龄** | int | 4 | null |
| **籍贯** | varchar | 50 | null |
| **工龄** | int | 4 | null |

表1.1.9 职工数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **职工-教学楼（Staff-TeachingBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **职工编号** | varchar | 50 | primary key  foreign key |
| **教学楼编号** | varchar | 50 | primary key  foreign key |

表1.1.10 职工-教学楼数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **职工-宿舍楼（Staff-DormitoryBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **职工编号** | varchar | 50 | primary key  foreign key |
| **宿舍楼编号** | varchar | 50 | primary key  foreign key |

表1.1.11 职工-宿舍楼数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生实体（Student）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **学号** | varchar | 50 | primary key |
| **姓名** | varchar | 50 | not null |
| **性别** | varchar | 50 | not null |
| **籍贯** | varchar | 50 | not null |
| **年龄** | int | 4 | null |

表1.1.12 学生数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-社团（Student-Association）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **学号** | varchar | 50 | primary key  foreign key |
| **社团名称** | varchar | 50 | primary key  foreign key |
| **职位** | varchar | 50 | null |

表1.1.13社团数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-班级（Student-Class）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **学号** | varchar | 50 | primary key  foreign key |
| **班级编号** | varchar | 50 | primary key  foreign key |
| **职位** | varchar | 50 | not null |

表1.1.14学生-班级数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-课程（Student-Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **学号** | varchar | 50 | primary key  foreign key |
| **课程号** | Varchar | 50 | primary key  foreign key |
| **教师编号** | varchar | 50 | primary key  foreign key |
| **成绩** | Int | 4 | null |

表1.1.15学生-课程数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-寝室（Student-Dormitory）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **学号** | varchar | 50 | primary key  foreign key |
| **寝室** | varchar | 50 | primary key  foreign key |
| **床位** | Int | 4 | notnull |

表1.1.16学生-寝室数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **教师实体（Teacher）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **教师编号** | varchar | 50 | primary key |
| **姓名** | varchar | 50 | not null |
| **性别** | varchar | 50 | not null |
| **籍贯** | varchar | 50 | null |
| **年龄** | int | 4 | null |
| **邮箱** | varchar | 50 | null |
| **教龄** | int | 4 | null |
| **院系** | varchar | 50 | foreign key |
| **职称** | varchar | 50 | null |

表1.1.17 教师数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **教师-课程（Teacher-Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **课程号** | varchar | 50 | primary key  foreign key |
| **教师编号** | varchar | 50 | primary key  foreign key |
| **上课地点** | varchar | 50 | null |
| **上课时间** | varchar | 50 | null |

表1.1.18教师-课程数据元素表

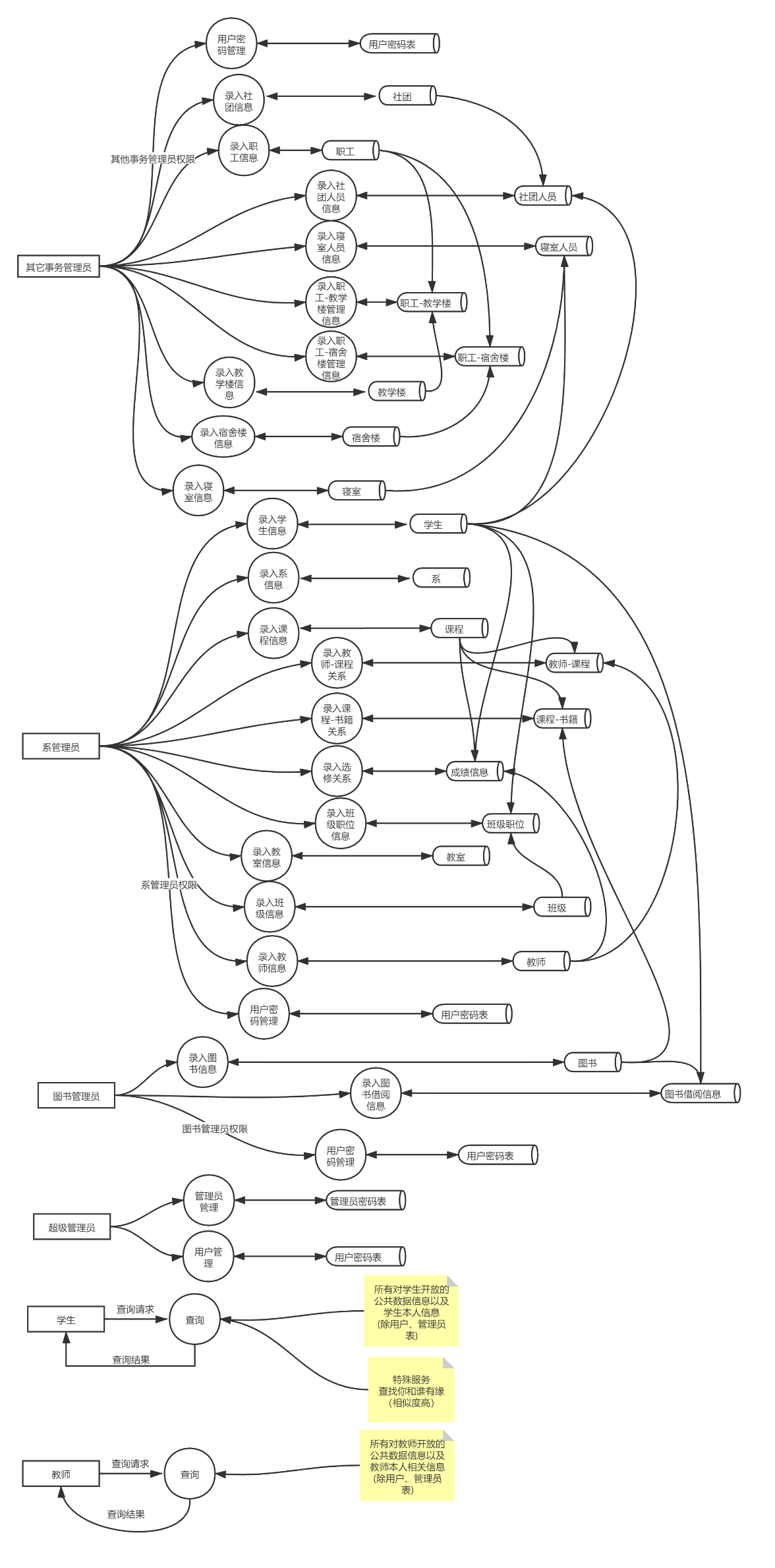
|  |  |  |  |
| --- | --- | --- | --- |
| **教学楼（TeachingBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **名称** | varchar | 50 | primary key |
| **所处区域** | varchar | 50 | null |
| **总楼层** | int | 4 | null |
| **建设年份** | int | 4 | null |

表1.1.19教学楼数据元素表

|  |  |  |  |
| --- | --- | --- | --- |
| **用户（User）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **登录名** | varchar | 50 | primary key |
| **密码** | varchar | 50 | not null |
| **类型** | varchar | 50 | not null |
| **编号** | varchar | 50 | not null |

表1.1.20用户数据元素表

（四）数据流图分析

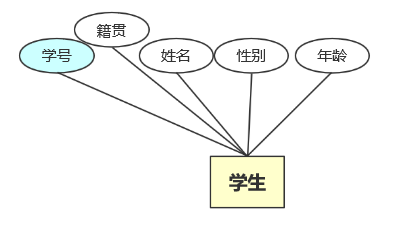


二 数据库概念设计

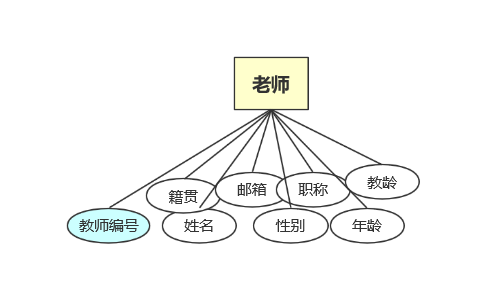
（一）实体设计

本系统中包括十二个实体：⑴学生实体；⑵老师实体；⑶班级实体；⑷系实体；⑸课程实体；⑹教室实体；⑺教学楼实体；⑻图书实体；⑼社团实体；⑽寝室实体；⑾宿舍楼实体；⑿职工实体。

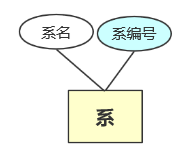
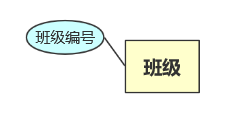
根据需求分析的结果，我们得到各个实体对应的ER属性图如下(属性中标记为蓝色的为该实体的主码)：



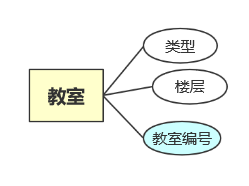
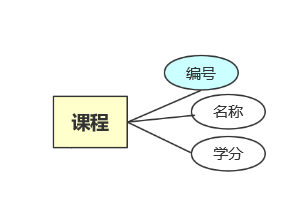
**图2.1.1学生实体属性图**



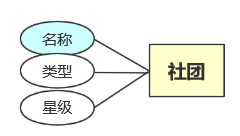
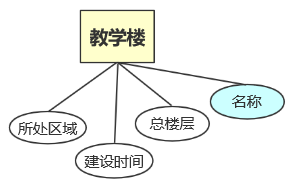
**图2.1.2老师实体属性图**

****

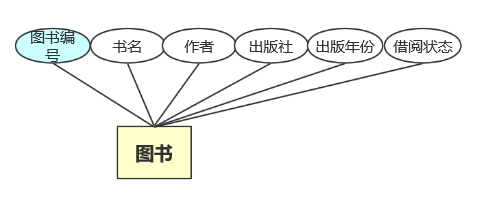
**图2.1.3班级实体属性图 图2.1.4系实体属性图**

****

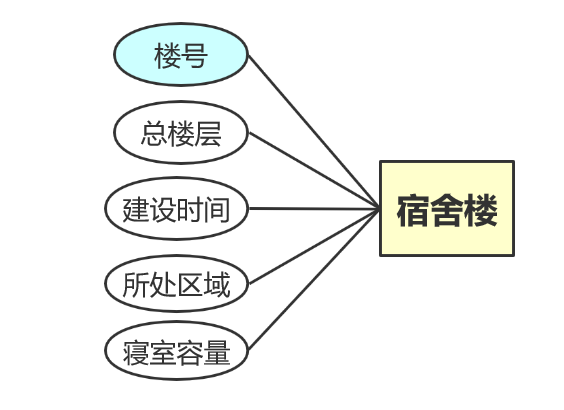
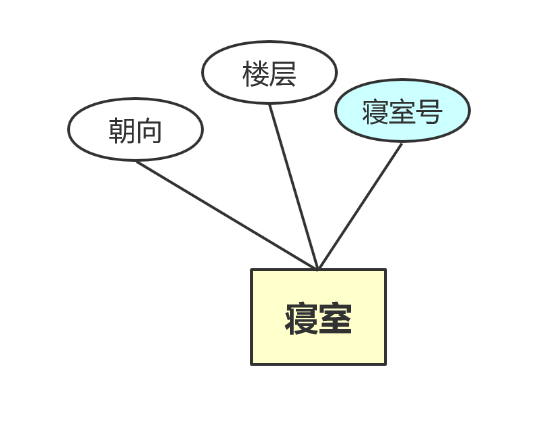
**图2.1.5课程实体属性图 图2.1.6教室实体属性图**

****

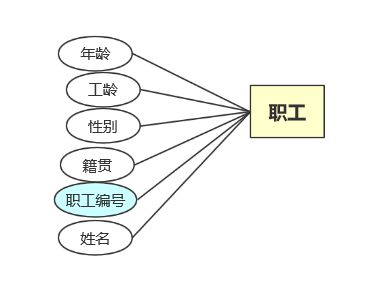
**图2.1.7教学楼实体属性图 图2.1.8社团实体属性图**

****

**图2.1.9图书实体属性图**



**图2.1.10寝室实体属性图 图2.1.11宿舍楼实体属性图**

****

**图2.1.12职工实体属性图**

（二）实体间关系设计

根据需求分析的结果，我们得到各个实体之间的关系如下所示：

**（1）学生-班级**



**图2.2.1学生-班级的辅导员关系**



**图2.2.2学生-班级的属于关系**

**（2）学生-社团**



**图2.2.3 学生-社团的参加关系**

**（3）学生-寝室**



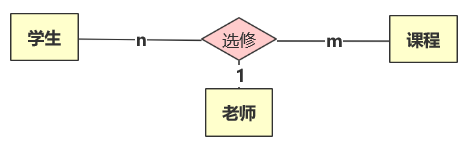
**图2.2.4 学生-寝室的住宿关系**

**（4）学生-图书**



**图2.2.5 学生-图书的借阅关系**

**（5）学生-课程-老师**



**图2.2.6学生-课程-老师的选修关系**

**（6）系-课程**



**图2.2.7 系-课程的开设关系**

**（7）系-老师**



**图2.2.8 系-老师的属于关系**



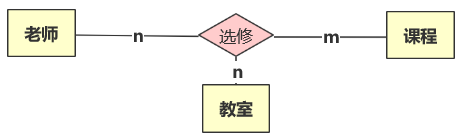
**图2.2.9 系-老师的系主任关系**

**（8）系-班级**



**图2.2.10系-班级的属于关系**

**（9）老师-课程-教室**



**图2.2.11 老师-课程-教室的选修关系**

**（10）老师-班级**



**图2.2.12老师-班级的班主任关系**

**（11）老师-社团**



**图2.2.13老师-社团的指导关系**

**（12）教室-教学楼**



**图2.2.14 教室-教学楼的位于关系**

**（13）寝室-宿舍楼**



**图2.2.15 寝室-宿舍楼的位于关系**

**（14）职工-教学楼**



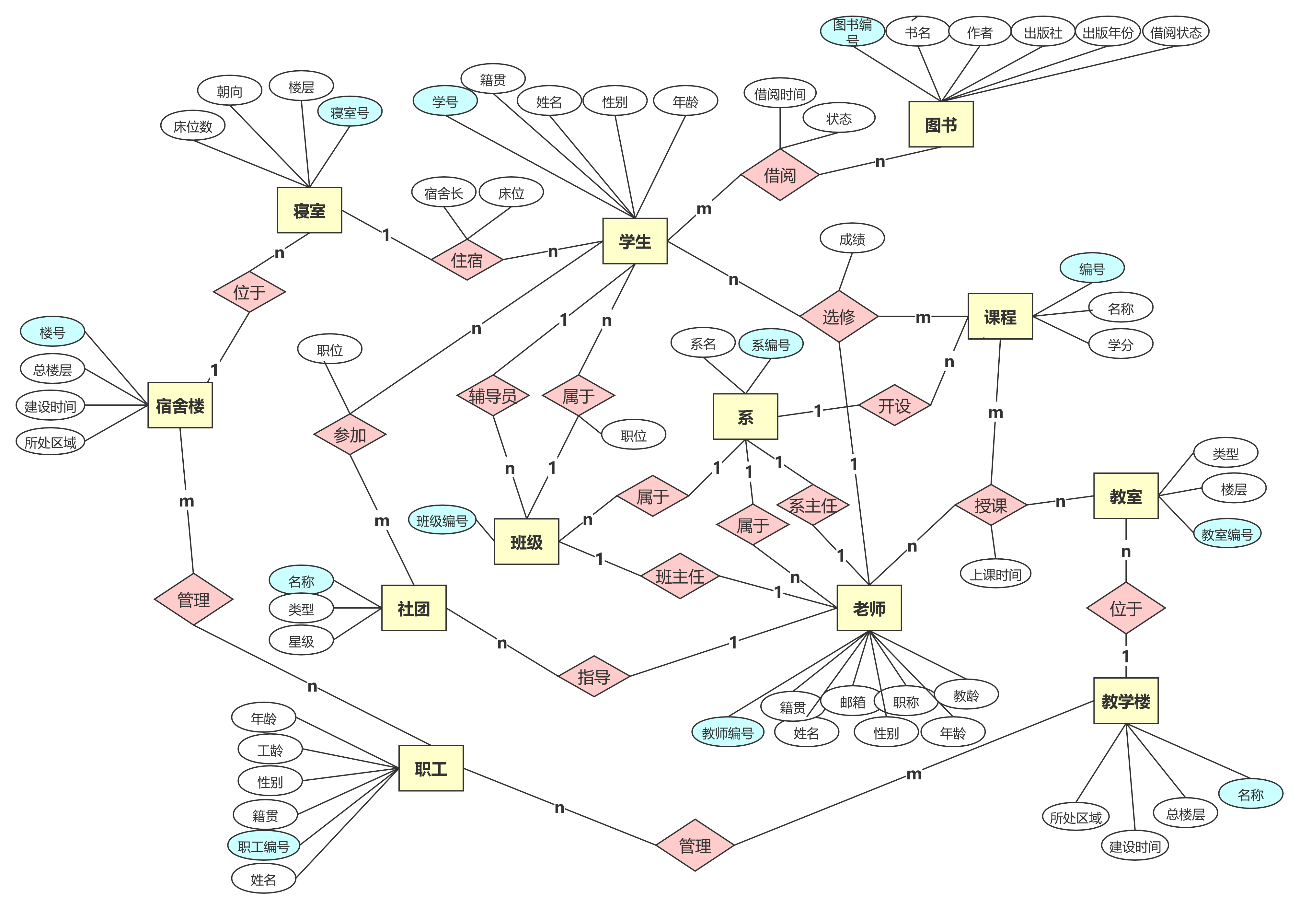
**图2.2.16 职工-教学楼的管理关系**

**（15）教室-宿舍楼**



**图2.2.17 职工-宿舍楼的管理关系**

（二）总体E-R图设计



**图2.3.1系统整体ER图**

三 数据库逻辑设计

（一）关系模式分析

根据第二部分的ER图，我们可以将图中的每一个实体、实体与实体之间的关系，分析得出一个个关系模式R(U)。其中关系的属性名集合U即为实体的属性集合。最终得到的所有关系模式如下所示：

Association（名称，类型，星级，指导老师）

Book（图书编号，书名，作者，出版社，出版年份，借阅状态）

Class（班级编号，系编号，班主任，辅导员）

Classroom（教室编号，教学楼名称，楼层，类型）

Course（课程编号，名称，学分，开设院系）

Department（系编号，系名，系主任）

Dormitory（寝室号，宿舍楼号，楼层，朝向，床位数，宿舍长）

DormitoryBuliding（楼号，总楼层，所处区域，建设年份）

Staff（职工编号，姓名，性别，年龄，籍贯，工龄）

Student（学号，姓名，性别，籍贯，年龄）

Teacher（教师编号，姓名，性别，年龄，籍贯，邮箱，教龄，院系，职称）

TeachingBuilding（名称，所属区域，总楼层，建设年份）

User（登录名，密码，类型，编号）

Staff-Dormitory（职工编号，宿舍楼号）

Staff-TeachingBuliding（职工编号，教学楼号）

Student-Association（学号，社团名称，职位）

Student-Book（学号，图书编号，借阅时间，状态）

Student-Class（学生编号，班级，职位）

Student-Course（学号，课程号，教师编号，成绩）

Student-Dormitory（学号，寝室，床位）

Teacher-Course（课程号，教师编号，上课地点，上课时间）

（二）函数依赖分析

在分析关系模式的范式等级之前需要先行明确关系模式种的函数依赖，根据函数依赖判断关系模式属于的范式等级。最终分析得到的函数依赖关系如下（X→Y表示X确定Y，Y依赖于X）：

（1）社团实体

社团名称→（类型，星级，指导老师）

（2）图书实体

图书编号→（书名，作者，出版社，出版年份，借阅状态）

（3）班级实体

班级编号→（系编号，班主任，辅导员）

（4）教室实体

教室编号→（教学楼名称，楼层，类型）

（5）课程实体

课程编号←→课程名称→（学分，开设院系）

（6）系实体

系编号→（系名，系主任）

（7）寝室实体

寝室号→（宿舍楼号，楼层，朝向，床位数，宿舍长）

（8）宿舍楼实体

宿舍楼号→（总楼层，所处区域，建设年份）

（9）职工实体

职工编号→（姓名，性别，年龄，籍贯，工龄）

（10）学生实体

学号→（姓名，性别，籍贯，年龄）

（11）教师实体

教师编号→（姓名，性别，年龄，籍贯，邮箱，教龄，院系，职称）

（12）教学楼实体

教学楼名称→（所属区域，总楼层，建设年份）

（13）学生-社团

（学号，社团名称）→职位

（14）学生-图书

（学号，图书编号，借阅时间）→状态

（15）学生-班级

学生编号→（班级，职位）

（16）学生-课程-老师

（学号，课程号，教师编号）→成绩

（17）学生-寝室

学号→（寝室，床位）

（18）老师-课程

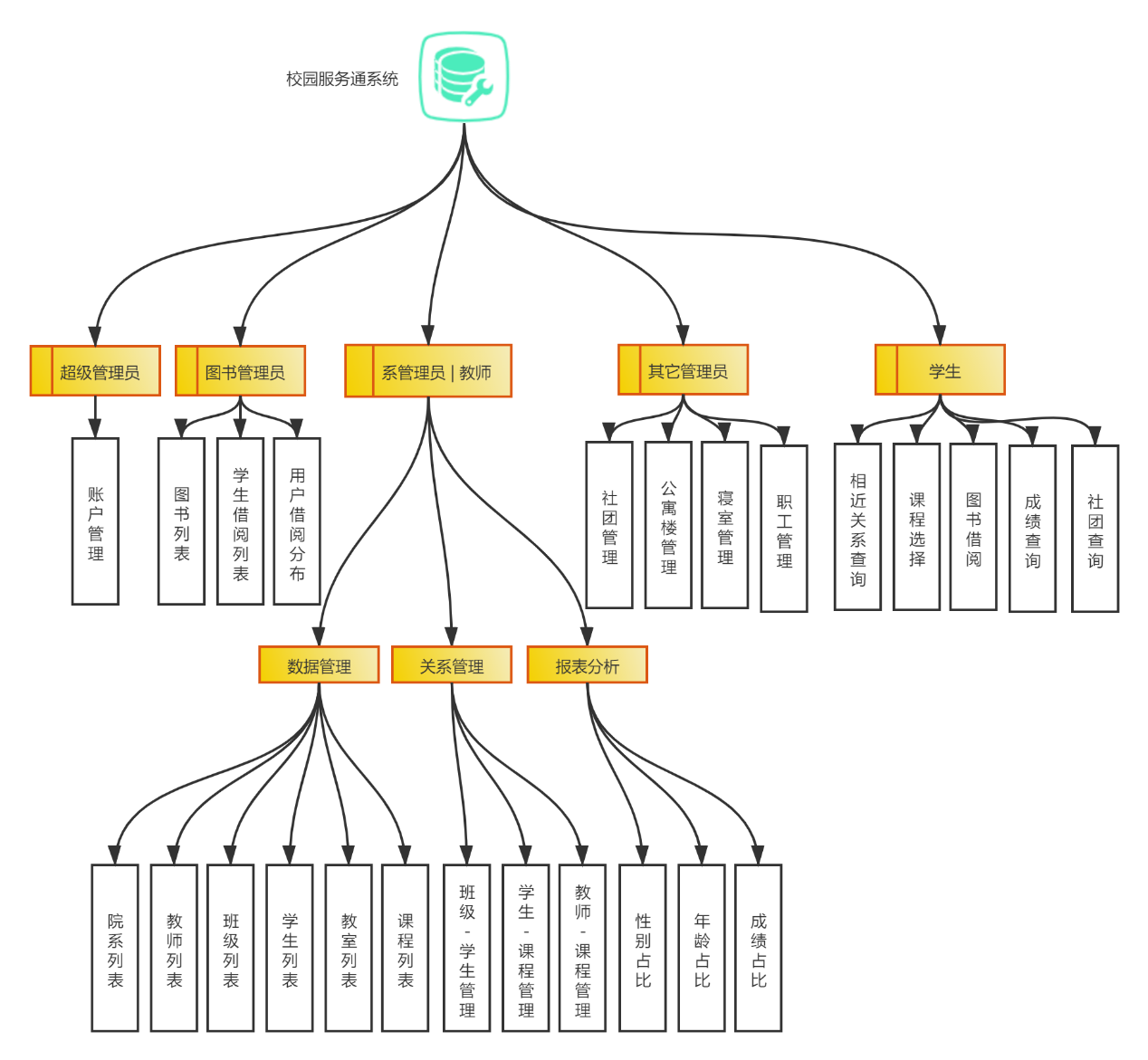
（课程号，教师编号）→（上课地点，上课时间）

（三）范式分析

经过分析每个关系的属性间的传递依赖性，可以发现每个关系的每一个非主 属性既不部分依赖于码，也不传递依赖于码，因此该数据库所有关系至少拥有 3NF。

数据库实现部分

四 系统功能结构图及安全性设计



五 基本表设计及相关说明

根据前文的数据元素分析及实体、实体与实体之间关系的分析，我们设计了如下基本表，对于每个基本表针对其数据特点设置完整性约束定义和索引定义。除此之外，为了追求与实际相符合，我们通过触发器实现所需的基本表间约束

（一）基本表设计及完整性约束

1.社团信息表（Association）

|  |  |  |  |
| --- | --- | --- | --- |
| **社团信息表（Association）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **名称** | varchar | 50 | notnull |
| **类型** | varchar | 50 | notnull |
| **星级** | varchar | 50 | not null |
| **指导老师** | varchar | 50 | null |

表1.1.1社团信息表

（1）实体完整性

**主码(PK):**(名称)

（2）参照完整性

**外码1(FK1)**:(指导老师) → Teacher(教师编号)

（3）用户定义完整性

**约束1(CK1)**:([星级]='三星' OR [星级]='二星' OR [星级]='一星')

**约束2(CK2)**:([类型]='人文类' OR [类型]='实践类' OR [类型]='科技类' OR [类型]='艺术类' OR [类型]='体育类')

2.图书信息表（Book）

|  |  |  |  |
| --- | --- | --- | --- |
| **图书信息表（Book）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **图书编号** | varchar | 50 | notnull |
| **书名** | varchar | 50 | not null |
| **作者** | varchar | 50 | notnull |
| **出版社** | varchar | 50 | null |
| **出版年份** | Int | 4 | null |
| **借阅状态** | varchar | 50 | not null |

表1.1.2 图书信息表

（1）实体完整性

**主码(PK):**(图书编号)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**:([图书编号] like 'B[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

**约束2(CK2)**:([借阅状态]='不可借阅' OR [借阅状态]='可借阅')

3.班级信息表（Class）

|  |  |  |  |
| --- | --- | --- | --- |
| **班级信息表（Class）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **班级编号** | varchar | 50 | notnull |
| **系编号** | varchar | 50 | notnull |
| **班主任** | varchar | 50 | notnull |
| **辅导员** | varchar | 50 | notnull |

表1.1.3班级信息表

（1）实体完整性

**主码(PK):**(班级编号)

（2）参照完整性

**外码1(FK1)**:(系编号) → Department(系编号)

**外码2(FK2)**:(辅导员) → Student(学号)

**外码3(FK3)**:(班主任) → Teacher(教师编号)

（3）用户定义完整性

**约束1(CK1)**:(substring([班级编号],(1),(2))=[系编号])

4.教室信息表（Classroom）

|  |  |  |  |
| --- | --- | --- | --- |
| **教室信息表（Classroom）** | | | |
| **数据项名** | **数据类型** | **长度** | **数据约束** |
| **教室编号** | varchar | 50 | notnull |
| **教学楼名称** | varchar | 50 | notnull |
| **楼层** | int | 4 | not null |
| **类型** | varchar | 50 | null |

表1.1.4教室信息表

（1）实体完整性

**主码(PK):**(教室编号)

（2）参照完整性

**外码1(FK1)**:(教学楼名称) → TeachingBuilding(名称)

（3）用户定义完整性

**约束1(CK1)**:([类型]='小' OR [类型]='中' OR [类型]='大')

5.课程信息表（Course）

|  |  |  |  |
| --- | --- | --- | --- |
| **课程信息表（Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **课程编号** | varchar | 50 | notnull |
| **名称** | varchar | 50 | not null |
| **学分** | int | 4 | not null |
| **开设院系** | varchar | 50 | notnull |

表1.1.5课程信息表

（1）实体完整性

**主码(PK):**(课程编号)

（2）参照完整性

**外码1(FK1)**:(开设院系) → Department(系编号)

（3）用户定义完整性

**约束1(CK1)**:([课程编号] like 'C[0-9][0-9]')

**约束1(CK1)**:([学分]<=(6))

6.系信息表（Department）

|  |  |  |  |
| --- | --- | --- | --- |
| **系信息表（Department）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **系编号** | varchar | 50 | notnull |
| **系名** | varchar | 50 | not null |
| **系主任** | varchar | 50 | null |

表1.1.6 系信息表

（1）实体完整性

**主码(PK):**(系编号)

（2）参照完整性

**外码1(FK1)**:(系主任) → Teacher(教师编号)

（3）用户定义完整性

**约束1(CK1)**:([系编号] like '[0-9][0-9]%')

7.寝室信息表（Dormitory）

|  |  |  |  |
| --- | --- | --- | --- |
| **寝室信息表（Dormitory）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **寝室号** | varchar | 50 | notnull |
| **宿舍楼号** | varchar | 50 | notnull |
| **楼层** | int | 4 | notnull |
| **朝向** | varchar | 50 | null |
| **宿舍长** | varchar | 50 | null |

表1.1.7 寝室信息表

（1）实体完整性

**主码(PK):**(寝室号)

（2）参照完整性

**外码1(FK1)**:(宿舍楼号) → DormitoryBuilding(楼号)

**外码2(FK2)**:(宿舍长) → Student(学号)

（3）用户定义完整性

**约束1(CK1)**:([朝向]='北' OR [朝向]='东' OR [朝向]='南' OR [朝向]='西')

8.宿舍楼信息表（DormitoryBuilding）

|  |  |  |  |
| --- | --- | --- | --- |
| **宿舍楼信息表（DormitoryBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **楼号** | varchar | 50 | notnull |
| **总楼层** | Int | 4 | notnull |
| **所处区域** | varchar | 50 | not null |
| **建设年份** | varchar | 50 | not null |
| **寝室容量** | int | 4 | notnull |
| **类型** | varchar | 50 | not null |

表1.1.8 宿舍楼信息表

（1）实体完整性

**主码(PK):**(楼号)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**:([总楼层]<(20))

**约束2(CK2)**:([建设年份]>=(1900) AND [建设年份]<=(2020))

**约束3(CK3)**:([所处区域]='大运村' OR [所处区域]='老北区' OR [所处区域]='新北区')

**约束4(CK4)**:([类型]='女' OR [类型]='男')

9.职工信息表（Staff）

|  |  |  |  |
| --- | --- | --- | --- |
| **职工信息表（Staff）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **职工编号** | varchar | 50 | notnull |
| **姓名** | varchar | 50 | not nul |
| **性别** | varchar | 50 | not nul |
| **年龄** | int | 4 | notnull |
| **籍贯** | varchar | 50 | null |
| **工龄** | int | 4 | null |

表1.1.9 职工信息表

（1）实体完整性

**主码(PK):**(职工编号)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**:([职工编号] like 'S[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

**约束2(CK2)**:([性别]='女' OR [性别]='男')

**约束3(CK3)**:(([年龄]-[工龄])>=(18))

10.职工-教学楼（Staff-TeachingBuilding）

|  |  |  |  |
| --- | --- | --- | --- |
| **职工-教学楼（Staff-TeachingBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **职工编号** | varchar | 50 | notnull |
| **教学楼编号** | varchar | 50 | notnull |

表1.1.10 职工-教学楼信息表

（1）实体完整性

**主码(PK):**(职工编号，宿舍楼号)

（2）参照完整性

**外码1(FK1)**:(宿舍楼号) → DormitoryBuilding(楼号)

**外码2(FK2)**:(职工编号) → Staff(职工编号)

（3）用户定义完整性

**无**

11.职工-宿舍楼（Staff-DormitoryBuilding）

|  |  |  |  |
| --- | --- | --- | --- |
| **职工-宿舍楼（Staff-DormitoryBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **职工编号** | varchar | 50 | notnull |
| **宿舍楼编号** | varchar | 50 | notnull |

表1.1.11 职工-宿舍楼信息表

（1）实体完整性

**主码(PK):**(职工编号，教学楼名称)

（2）参照完整性

**外码1(FK1)**:(教学楼名称) → TeachingBuilding(名称)

**外码2(FK2)**:(职工编号) → Staff(职工编号)

（3）用户定义完整性

**无**

12.学生信息表（Student）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生信息表（Student）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **姓名** | varchar | 50 | not null |
| **性别** | varchar | 50 | not null |
| **籍贯** | varchar | 50 | not null |
| **年龄** | int | 4 | null |

表1.1.12 学生信息表

（1）实体完整性

**主码(PK):**(学号)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**:([学号] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

**约束2(CK2)**:([性别]='女' OR [性别]='男')

**约束3(CK3)**:([年龄]<(100) AND [年龄]>(0))

13.学生-社团（Student-Association）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-社团（Student-Association）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **社团名称** | varchar | 50 | notnull |
| **职位** | varchar | 50 | null |

表1.1.13社团信息表

（1）实体完整性

**主码(PK):**(学号，社团名称)

（2）参照完整性

**外码1(FK1)**:(学号) → Student(学号)

**外码2(FK2)**:(社团名称) → Association(名称)

（3）用户定义完整性

**约束1(CK1)**:([职位]='社员' OR [职位]='副社长' OR [职位]='社长')

14.学生-书籍（Student-Book）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-书籍（Student-Book）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **图书编号** | varchar | 50 | notnull |
| **借阅时间** | varchar | 50 | notnull |
| **状态** | varchar | 50 | not null |

（1）实体完整性

**主码(PK):**(学号，图书编号，借阅时间)

（2）参照完整性

**外码1(FK1)**:(图书编号) → Book(图书编号)

**外码2(FK2)**:(学号) → Student(学号)

（3）用户定义完整性

**约束1(CK1)**:([状态]='未归还' OR [状态]='已归还')

15.学生-班级（Student-Class）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-班级（Student-Class）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **班级编号** | varchar | 50 | notnull |
| **职位** | varchar | 50 | not null |

表1.1.14学生-班级信息表

（1）实体完整性

**主码(PK):**(学号)

（2）参照完整性

**外码1(FK1)**:(学号) → Student(学号)

**外码2(FK2)**:(班级) → Class(班级编号)

（3）用户定义完整性

**约束1(CK1)**:([职位]='普通学生' OR [职位]='生活委员' OR [职位]='学习委员' OR [职位]='副班长' OR [职位]='班长')

16.学生-课程（Student-Course）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-课程（Student-Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **课程号** | Varchar | 50 | notnull |
| **教师编号** | varchar | 50 | notnull |
| **成绩** | Int | 4 | null |

表1.1.15学生-课程信息表

（1）实体完整性

**主码(PK):**(学号，课程号)

（2）参照完整性

**外码1(FK1)**:(学号) → Student(学号)

**外码2(FK2)**:(课程号) → Course(课程编号)

**外码3(FK3)**:(教师编号) → Teacher(教师编号)

（3）用户定义完整性

**约束1(CK1)**:([成绩]>=(0) AND [成绩]<=(100))

17.学生-寝室（Student-Dormitory）

|  |  |  |  |
| --- | --- | --- | --- |
| **学生-寝室（Student-Dormitory）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **学号** | varchar | 50 | notnull |
| **寝室** | varchar | 50 | notnull |
| **床位** | Int | 4 | notnull |

表1.1.16学生-寝室信息表

（1）实体完整性

**主码(PK):**(学号)

（2）参照完整性

**外码1(FK1)**:(学号) → Student(学号)

**外码2(FK2)**:(寝室) → Dormitory(寝室号)

（3）用户定义完整性

**约束1(CK1)**:(寝室，床位) UNIQUE

18.教师信息表（Teacher）

|  |  |  |  |
| --- | --- | --- | --- |
| **教师信息表（Teacher）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **教师编号** | varchar | 50 | notnull |
| **姓名** | varchar | 50 | not null |
| **性别** | varchar | 50 | not null |
| **籍贯** | varchar | 50 | null |
| **年龄** | int | 4 | null |
| **邮箱** | varchar | 50 | null |
| **教龄** | int | 4 | null |
| **院系** | varchar | 50 | notnull |
| **职称** | varchar | 50 | null |

表1.1.17 教师信息表

（1）实体完整性

**主码(PK):**(教师编号)

（2）参照完整性

**外码1(FK1)**:(院系) → Department(系编号)

（3）用户定义完整性

**约束1(CK1)**:([教师编号] like 'T[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

**约束2(CK2)**:([性别]='女' OR [性别]='男')

**约束3(CK3)**:([年龄] >(0) AND [年龄]<(100))

**约束4(CK4)**:(([年龄]-[教龄])>=(18))

**约束5(CK5)**:([邮箱] like '%@%')

**约束6(CK6)**:([职称]='讲师' OR [职称]='副教授' OR [职称]='教授')

19.教师-课程（Teacher-Course）

|  |  |  |  |
| --- | --- | --- | --- |
| **教师-课程（Teacher-Course）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **课程号** | varchar | 50 | notnull |
| **教师编号** | varchar | 50 | notnull |
| **上课地点** | varchar | 50 | not null |
| **上课时间** | varchar | 50 | not null |

表1.1.18教师-课程信息表

（1）实体完整性

**主码(PK):**(课程号，教师编号)

（2）参照完整性

**外码1(FK1)**:(上课地点) → Classroom(教室编号)

**外码2(FK2)**:(课程号) → Course(课程编号)

**外码3(FK3)**:(教师编号) → Teacher(教师编号)

（3）用户定义完整性

**约束1(CK1)**:[上课时间] = 周几上/下午第几节

20.教学楼信息表（TeachingBuilding）

|  |  |  |  |
| --- | --- | --- | --- |
| **教学楼信息表（TeachingBuilding）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **名称** | varchar | 50 | not null |
| **所处区域** | varchar | 50 | not null |
| **总楼层** | int | 4 | not null |
| **建设年份** | int | 4 | null |

表1.1.19教学楼信息表

（1）实体完整性

**主码(PK):**(名称)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**:([所处区域]='新区' OR [所处区域]='老区')

**约束2(CK2)**:([总楼层]<=(15))

**约束3(CK3)**:([建设年份]>=(1900) AND [建设年份]<=(2020))

21.用户信息表（User）

|  |  |  |  |
| --- | --- | --- | --- |
| **用户信息表（User）** | | | |
| **数据项名** | **数据类型** | **长度** | **允许null值** |
| **登录名** | varchar | 50 | notnull |
| **密码** | varchar | 50 | not null |
| **类型** | varchar | 50 | not null |
| **编号** | varchar | 50 | null |

表1.1.20用户信息表

（1）实体完整性

**主码(PK):**(登录名)

（2）参照完整性

**无**

（3）用户定义完整性

**约束1(CK1)**: ([类型]='超级管理员' OR [类型]='图书管理员' OR [类型]='系管理员' OR [类型]='其他管理员' OR [类型]='学生' OR [类型]='老师')

（二）触发器设计

（1）modify\_classroom\_floor

**作用基本表：**Classroom, TeachingBuilding

**限制动作**：insert, update

**触发器功能：**限制教室所在楼层不能高于所在教学楼的总楼层数

**具体代码实现：**

ALTERTRIGGER [dbo].[modify\_classroom\_floor]

ON [dbo].[Classroom]

AFTERinsert,update

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SETNOCOUNTON;

Declare @floor int;

Declare @height int;

select @floor =(select楼层from inserted)

select @height =(select总楼层from TeachingBuilding

where名称in(select教学楼名称from inserted)

)

if (@floor > @height)begin

Declare @cnt int;

select @cnt =(selectcount(\*)from deleted)

if (@cnt > 0)begin

update Classroom

set Classroom.楼层=(select楼层from deleted)

where Classroom.教室编号=(select教室编号from deleted)

end

elsebegin

deletefrom Classroom

where Classroom.教室编号=(select教室编号from inserted)

end

end

-- Insert statements for trigger here

END

（2）modify\_dorm\_floor

**作用基本表：**Dormitory, DormitoryBuilding

**限制动作**：insert, update

**触发器功能：**限制寝室所在楼层不能高于所在宿舍楼的总楼层数

**具体代码实现：**

ALTERTRIGGER [dbo].[modify\_dorm\_floor]

ON [dbo].[Dormitory]

AFTERinsert,update

AS

BEGIN

SETNOCOUNTON;

Declare @floor int;

Declare @height int;

select @floor =(select楼层from inserted)

select @height =(select总楼层from DormitoryBuilding

where楼号in(select宿舍楼号from inserted)

)

if (@floor > @height)begin

Declare @cnt int;

select @cnt =(selectcount(\*)from deleted)

if (@cnt > 0)begin

update Dormitory

set dormitory.楼层=(select楼层from deleted)

where dormitory.寝室号=(select寝室号from deleted)

end

elsebegin

deletefrom Dormitory

where dormitory.寝室号=(select寝室号from inserted)

end

end

END

（3）set\_master

**作用基本表：**Dormitory, Student-Dormitory

**限制动作**：insert, update

**触发器功能：**限制一个寝室的寝室长必须是住在该寝室同学

**具体代码实现：**

ALTERTRIGGER [dbo].[set\_master]

ON [dbo].[Dormitory]

AFTERinsert,update

AS

BEGIN

SETNOCOUNTON;

Declare @master varchar(50)

Declare @cnt int;

select @cnt =(selectcount(\*)from inserted where宿舍长isNULL)

if (@cnt = 0)begin

select @master =(select宿舍长from inserted)

select @cnt =(selectcount(\*)from [Student-Dormitory]

where [Student-Dormitory].学号= @master and

[Student-Dormitory].寝室=(select寝室号from inserted)

)

if (@cnt = 0)begin

update Dormitory

set宿舍长=NULL

where寝室号=(select寝室号from inserted)

end

end

END

（4）update\_borrow

**作用基本表：**Book, Student-Book

**限制动作**：insert, update

**触发器功能：**实现借书状态与图书借阅状态的同步转换。当借书记录中出现某一条的未归还的借书记录，需将该图书的借阅状态改为不可借阅；当借书记录的未归还变成已归还，则将该图书的借阅状态改为可借阅。

**具体代码实现：**

ALTERTRIGGER [dbo].[update\_borrow]

ON [dbo].[Student\_Book]

AFTERinsert,update

AS

BEGIN

SETNOCOUNTON;

declare @book varchar;

declare @state varchar;

select学号,图书编号from inserted

update book

set借阅状态='可借阅'

where book.图书编号in(

select图书编号from inserted

where状态='已归还'

)

update book

set借阅状态='不可借阅'

where book.图书编号in(

select图书编号from inserted

where状态='未归还'

)

END

（5）judge\_pos

**作用基本表：**Student\_Class

**限制动作**：insert, update

**触发器功能：**限制一个班级中每种类型的班干部最多只能有一个同学担任

**具体代码实现：**

ALTERTRIGGER [dbo].[judge\_pos]

ON [dbo].[Student\_Class]

AFTERinsert,update

AS

BEGIN

Declare @pos varchar(50)

select @pos =(select职位from inserted)

if (@pos !='普通学生')begin

Declare @cnt int

select @cnt =(selectcount(\*)from [Student\_Class] where [Student\_Class].职位= @pos and [Student\_Class].班级=(select班级from inserted))

if (@cnt > 1)begin

select @cnt =(selectcount(\*)from deleted)

if (@cnt > 0)begin

update [Student\_Class]

set [Student\_Class].职位=(select职位from deleted)

where [Student\_Class].学生编号=(select学生编号from deleted)

end

elsebegin

deletefrom [Student\_Class]

where [Student\_Class].学生编号=(select学生编号from inserted)

end

end

end

END

（6）judge\_club\_pos

**作用基本表：**Student\_Association

**限制动作**：insert, update

**触发器功能：**限制每个社团中，社长和副社长最多只能有一个同学担任

**具体代码实现：**

ALTERTRIGGER [dbo].[judge\_club\_pos]

ON [dbo].[Student-Association]

AFTERinsert,update

AS

BEGIN

SETNOCOUNTON;

Declare @pos varchar(50)

select @pos =(select职位from inserted)

if (@pos !='社员')begin

Declare @cnt int

select @cnt =(selectcount(\*)from [student-association]

where [student-association].职位= @pos and [student-association].社团名称=(select社团名称from inserted)

)

if (@cnt > 1)begin

select @cnt =(selectcount(\*)from deleted)

if (@cnt > 0)begin

update [student-association]

set [student-association].职位=(select职位from deleted)

where [student-association].学号=(select学号from deleted)

end

elsebegin

deletefrom [student-association]

where [student-association].学号=(select学号from inserted)

end

end

end

END

（7）judge\_sex

**作用基本表：**Student\_Dormitory, Dormitory, DormitoryBuilding

**限制动作**：insert, update

**触发器功能：**限制男生只能入住男生公寓，女生只能入住女生公寓

**具体代码实现：**

ALTERTRIGGER [dbo].[judge\_sex]

ON [dbo].[Student-Dormitory]

AFTERinsert,update

AS

BEGIN

SETNOCOUNTON;

Declare @sex varchar(50)

Declare @bsex varchar(50)

select @sex =(select student.性别from student, inserted

where inserted.学号= Student.学号

)

select @bsex =(select DormitoryBuilding.类型from Dormitory, inserted, DormitoryBuilding

where inserted.寝室= Dormitory.寝室号and DormitoryBuilding.楼号= Dormitory.宿舍楼号

)

if (@sex != @bsex)begin

deletefrom [Student-Dormitory]

where学号=(select学号from inserted)

end

END

六 代码实现及存储过程说明

（一）实现环境说明

本次数据库系统实现采用前后端分离开发方式，开发语言为Java， 开发jdk为Java1.8.0\_241，开发平台为IDEA。

（二）主要技术支持

前端采用vue框架，后端采用springboot框架，（袁哥阐述一下前后端连接原理）

（1）Java连接数据库

（三）后端代码实现存储过程说明

后端Sprintboot框架的数据传递模式采用的是Controller – (Entity) – Mapper 两层映射信息传递模式，其中Controller层是用于定义与前端的信息交互接口，接收前端的请求传递给底部的Mapper层，并将Mapper层返回的数据信息传递给前端。Mapper层用于将前端的请求转化为相应的SQL语句，按照存储过程的类型执行以直接访问并操作数据库。Entity实体类用于作为Controller层与Mapper层之间信息传递的载体，将Mapper层返回的数据库信息封装为一个个实体类，返回给Controller层，Controller层进一步将其解析为json格式并传递给前端。

在实现中，我们对数据库中每一个基本表表都新建了一个Entity实体，用于传递基本表的信息，实体包含一个构造函数以及所有属性的get和set方法（由于实体中所包含的属性与基本表的列一一对应，因此在后续的代码说明中不做阐述）。与其相应的也建立Mapper和Controller层用于前后端的数据传递。

除了每个基本表对应的Entity实体之外，为了查询方便，我们针对不同的前端查询需求新建一个对应的Query实体，用于存储后端返回的所需查询的数值，并在Mapper和Controller中加以使用。下面按照不同的基本表对**Mapper中实现的存储过程**进行说明。

1.Association

1）.查找所有社团信息



2）.按照社团名称查找社团信息



3）.按照指导老师查找社团信息



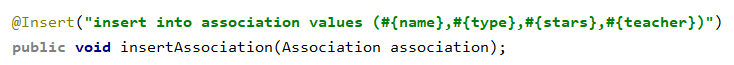
4）.按照社团类别查找社团信息



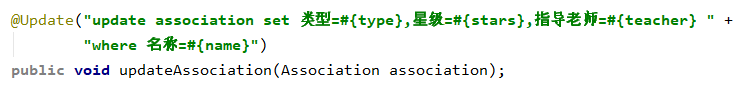
5）.按照星级查找社团信息



6）.插入一条新的社团信息



7）.更新一条社团信息



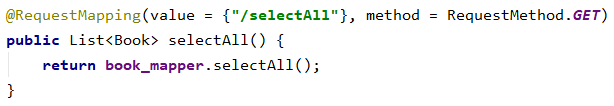
8）.删除一条社团信息



2.Book

1）.查看所有图书信息





2）.按照图书编号查找图书信息



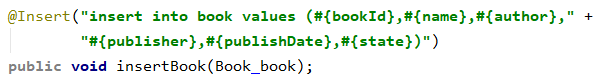
3）.按照图书编号或书名模糊查找图书信息



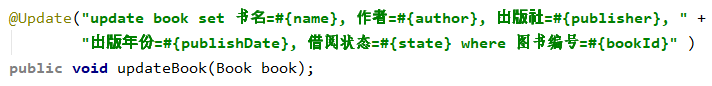
4）.按照借阅状态查找图书信息



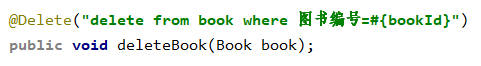
5）.插入一条新的图书信息



6）.更新一条图书信息



7）.删除一条社团信息



3.Class

1）.查看所有班级信息



2）.按照班级编号查找班级信息



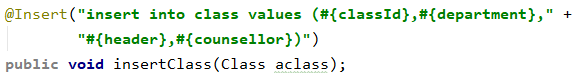
3）.按照系编号查找班级信息



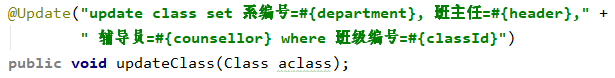
4）.按照辅导员查找班级信息



5）.插入一条新的班级信息



5）.更新一条班级信息



6）.删除一条班级信息



4.Classroom

1）.查看所有教室信息



2）.按照教室编号查找教室信息



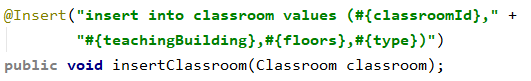
3）.按照教学楼名称查找教室信息



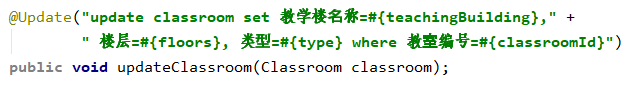
4）.按照类型查找教室信息



5）.插入一条新的教室信息



6）.更新一条教室信息



7）.删除一条教室信息

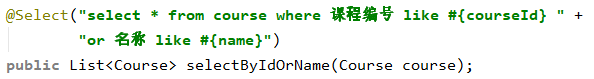


5.Course

1）.查看所有课程信息



2）.按照课程编号或名称模糊查找课程信息



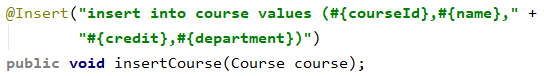
3）.按照名称查找课程信息



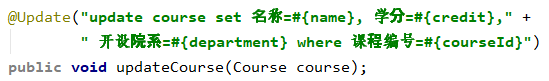
4）.按照开设院系查找课程信息



5）.插入一条新的课程信息



6）.更新一条课程信息



7）.删除一条课程信息

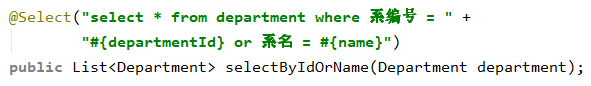


6.Department

1）.查看所有系信息



2）.按照系编号或系名模糊查找系信息



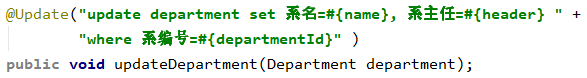
3）.按照系名查找系信息



4）.插入一条新的系信息



5）.更新一条系信息



6）.删除一条系信息



7.Dormitory

1）.查看所有寝室信息



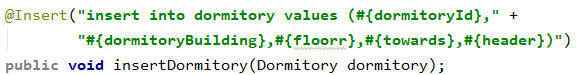
2）.按照寝室号查找寝室信息



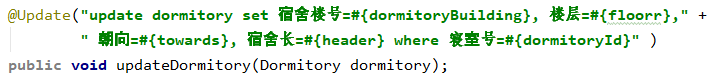
3）.按照宿舍楼号查找寝室信息



4）.插入一条新的寝室信息



5）.更新一条寝室信息



6）.删除一条寝室信息

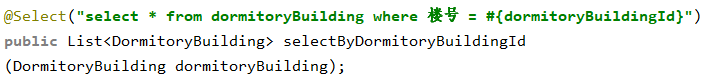


8.DormitoryBuilding

1）.查看所有宿舍楼信息



2）.按照楼号查找宿舍楼信息



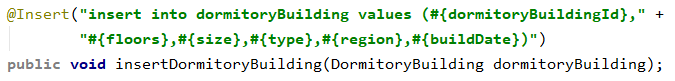
3）.按照类型查找宿舍楼信息



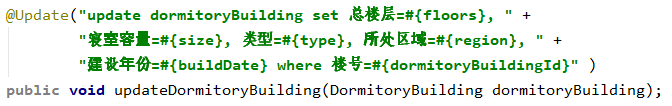
4）.按照所属区域查找宿舍楼信息



5）.插入一条新的宿舍楼信息



6）.更新一条宿舍楼信息



7）.删除一条宿舍楼信息



9.Staff

1）.查看所有职工信息



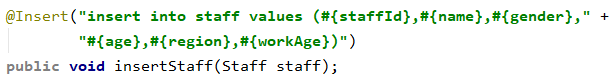
2）.按照职工编号查找职工信息



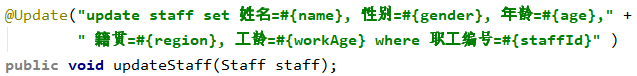
3）.按照性别查找职工信息



4）.插入一条新的职工信息



5）.更新一条职工信息



6）.删除一条职工信息

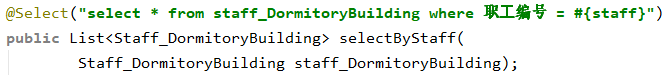


10.Staff\_DormitoryBuilding

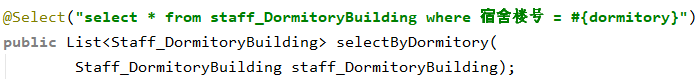
1）.查看所有职工管理宿舍楼信息



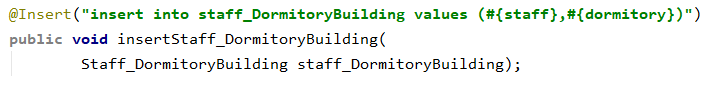
2）.按照职工编号查找职工管理宿舍楼信息



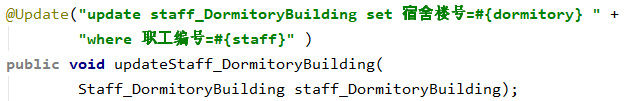
3）.按照宿舍楼号查找职工管理宿舍楼信息



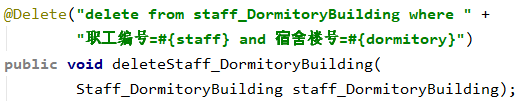
4）.插入一条新的职工管理宿舍楼信息



5）.更新一条职工管理宿舍楼信息



6）.删除一条职工管理宿舍楼信息

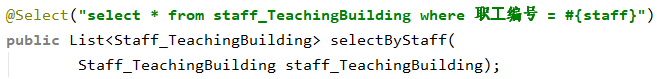


11.Staff\_TeachingBuilding

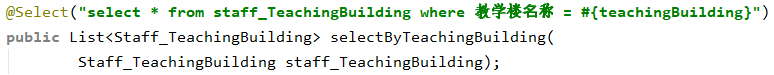
1）.查看所有职工管理教学楼信息



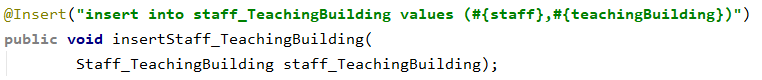
2）.按照职工编号查找职工管理教学楼信息



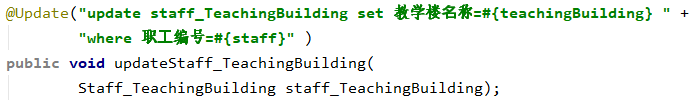
3）.按照教学楼名称查找职工管理教学楼信息



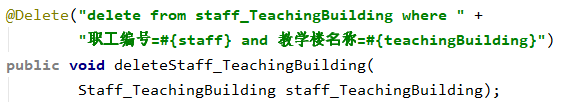
4）.插入一条新的职工管理教学楼信息



5）.更新一条职工管理教学楼信息



6）.删除一条职工管理教学楼信息

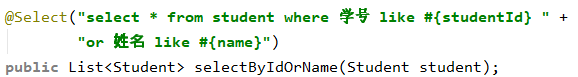


12.Student

1）.查看所有学生信息



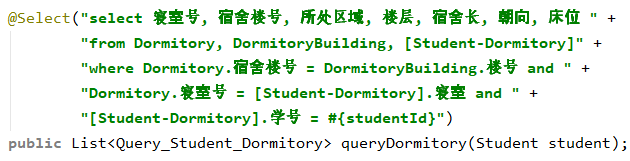
2）.按照学号或姓名模糊查找学生信息



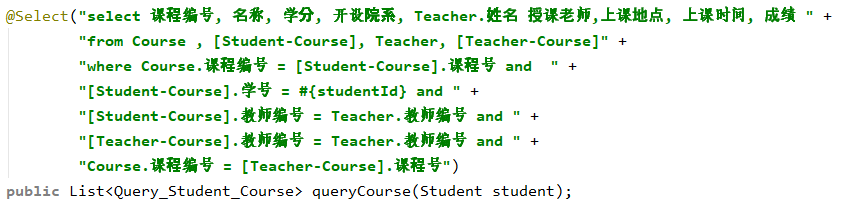
3）.按照姓名查找学生信息



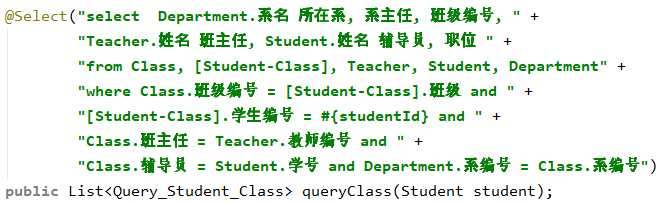
4）.查找学生的宿舍相关信息



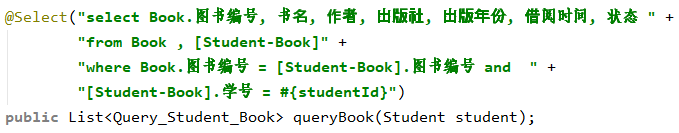
5）.查找学生的选课信息



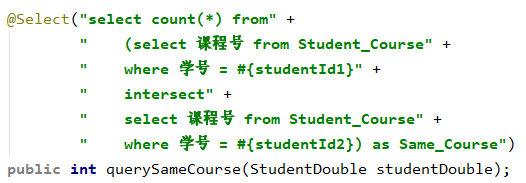
6）.查找学生所在的班级信息



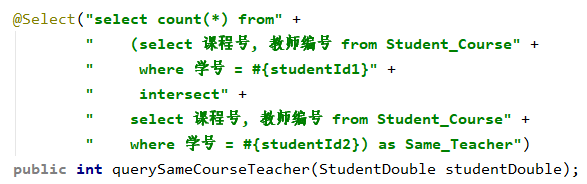
7）.查找学生的图书借阅信息



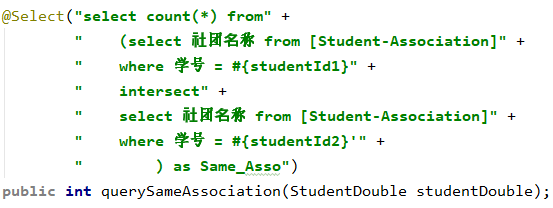
8）.查看两个学生选修相同的课程数量



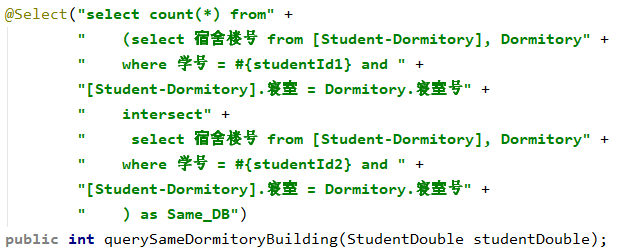
9）.查看两个学生选修同一个老师同一个课程的数量



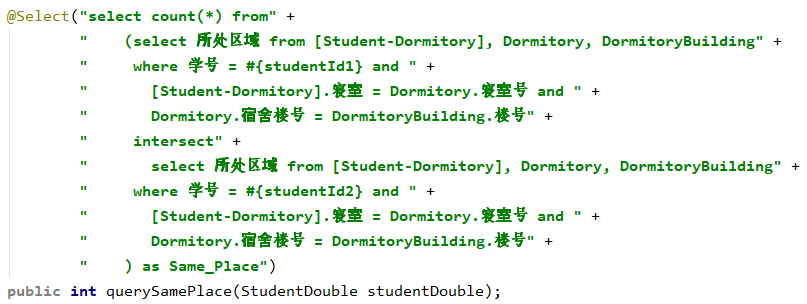
10）.查看两个学生参加相同社团的数量



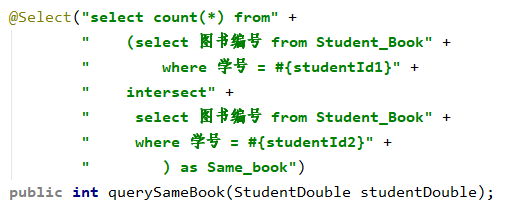
11）.查看两个学生是否住在同一个宿舍楼



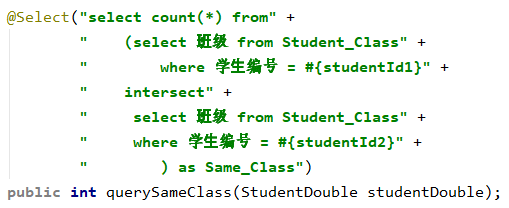
12）.查看两个学生是否住在同一个区域



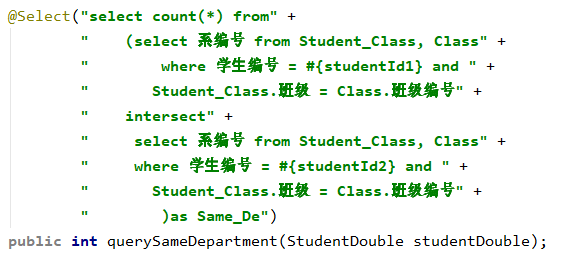
13）.查看两个学生借阅相同图书的个数



14）.查看两个学生是否在同一班级



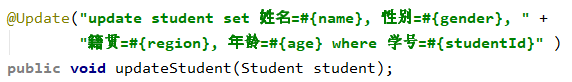
15）.查看两个学生是否在同一个系



16）.插入一条新的学生信息



17）.更新一条学生信息



18）.删除一条学生信息



13.Student\_Book

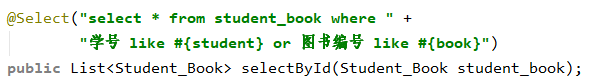
1）.查看所有学生借阅图书信息



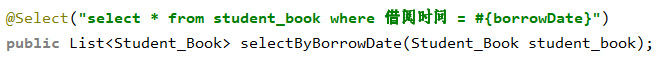
2）.按照学号查找学生借阅图书信息



3）.按照学号或图书编号模糊查找学生借阅图书信息



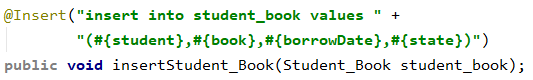
4）.按照借阅时间查找学生借阅图书信息



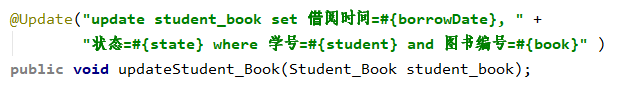
5）.按照借阅状态查找学生借阅图书信息



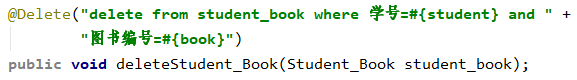
6）.插入一条新的学生借阅图书信息



7）.更新一条学生借阅图书信息

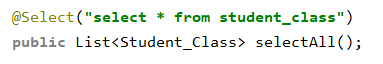


8）.删除一条学生借阅图书信息



14.Student\_Class

1）.查看所有学生所处班级信息



2）.按照学号查找学生所处班级信息



3）.按照班级查找学生所处班级信息



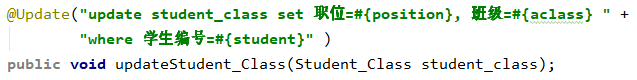
4）.按照职位查找学生所处班级信息



5）.插入一条新的学生所处班级信息



6）.更新一条学生所处班级信息



7）.删除一条学生所处班级信息



15.Student\_Course

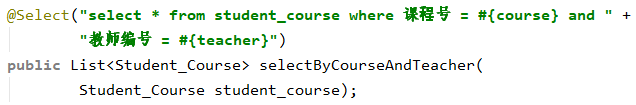
1）.查看所有学生选修课程信息



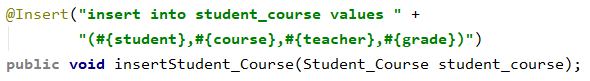
2）.按照学号查找学生选修课程信息



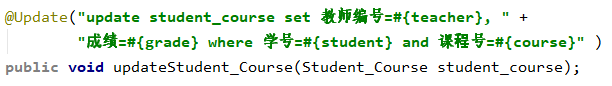
3）.按照课程号和教师编号学生选修课程信息



4）.插入一条新的学生选修课程信息



5）.更新一条学生选修课程信息



6）.删除一条学生选修课程信息

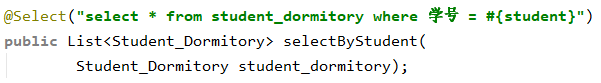


16.Student\_Dormitory

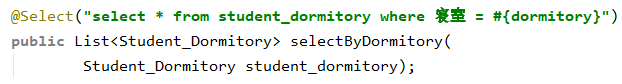
1）.查看所有学生住宿信息



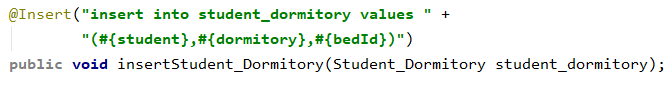
2）.按照学号查找学生住宿信息



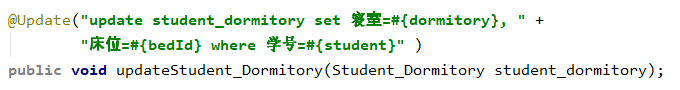
3）.按照寝室查找学生住宿信息



4）.插入一条新的学生住宿信息



5）.更新一条学生住宿信息



6）.删除一条学生住宿信息

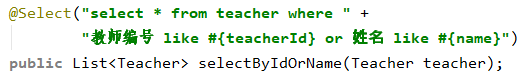


17.Teacher

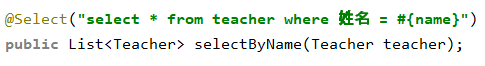
1）.查看所有教师信息



2）.按照教师编号或姓名模糊查找教师信息



3）.按照姓名查找教师信息



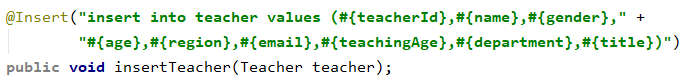
4）.按照性别查找教师信息



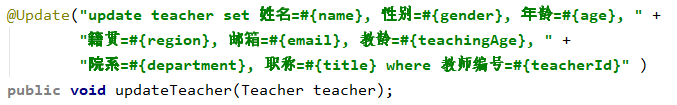
5）.按照院系查找教师信息



6）.插入一条新的教师信息



7）.更新一条教师信息



8）.删除一条教师信息

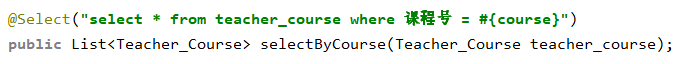


18.Teacher\_Course

1）.查看所有教师授课信息



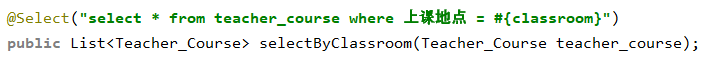
2）.按照课程号查找教师授课信息



3）.按照教师编号查找教师授课信息



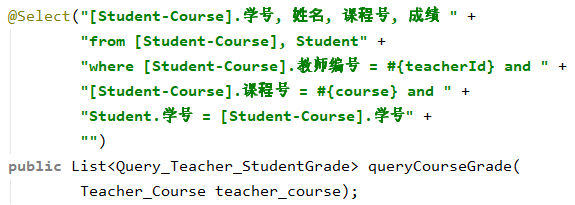
4）.按照上课地点查找教师授课信息



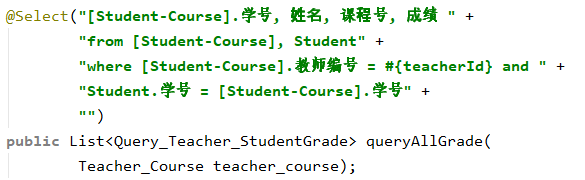
5）.按照上课时间查找教师授课信息



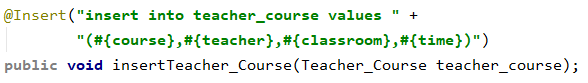
6）.查找所有选修该教师教授某门课程的所有学生成绩信息



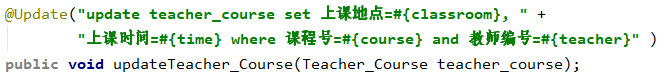
7）.查找所有选修该教师教授课程的所有学生成绩信息



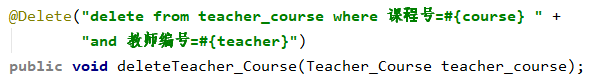
8）.插入一条新的教师授课信息



9）.更新一条教师授课信息



10）.删除一条教师授课信息



19.TeachingBuilding

1）.查看所有教学楼信息



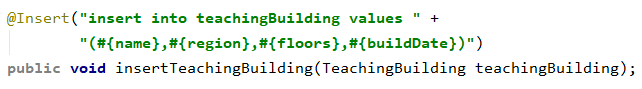
2）.按照名称查找教学楼信息



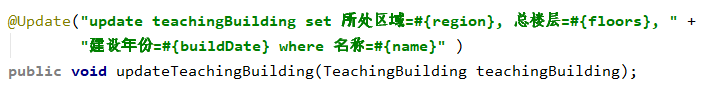
3）.按照所处区域查找教学楼信息



4）.插入一条新的教学楼信息



5）.更新一条教学楼信息



6）.删除一条教学楼信息



20.User

1）.查看所有用户信息



2）.按照登录名查找用户信息



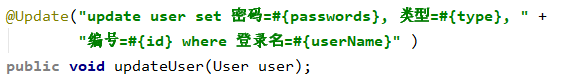
3）.按照类型查找用户信息



4）.插入一条新的用户信息



5）.更新一条用户信息



6）.删除一条用户信息

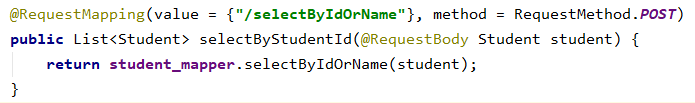


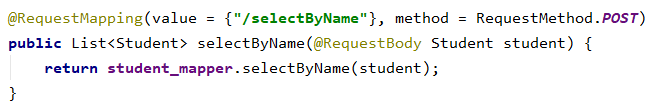
对于Mapper中的每一个存储过程，我们在Controller层中都建立一个相对应的前端交互入口，对于大多数的存储过程来说，Controller层中的交互入口是完全类似的，对部分特殊的查询过程需要特殊处理。在此处我们以Student表中的存储过程为例，简要说明存储过程到前端交互入口的映射。

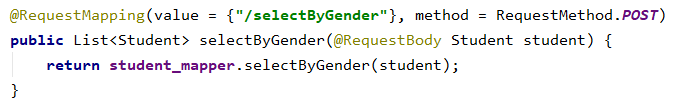
1.基础查询操作

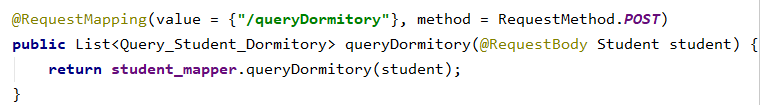
对于所有基本表的查询语句完全一致，直接在controller中调用Mapper中对应的接口即可，并将mapper接口返回的值返回到前端。

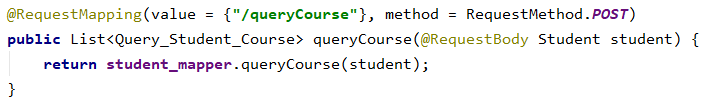


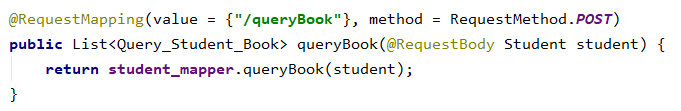








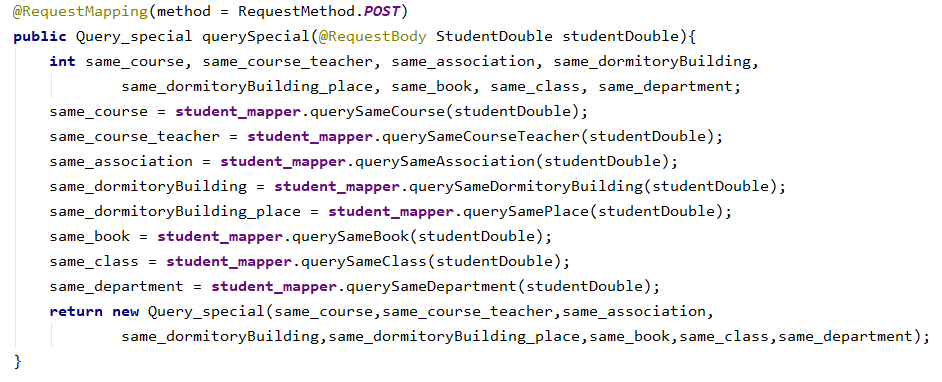




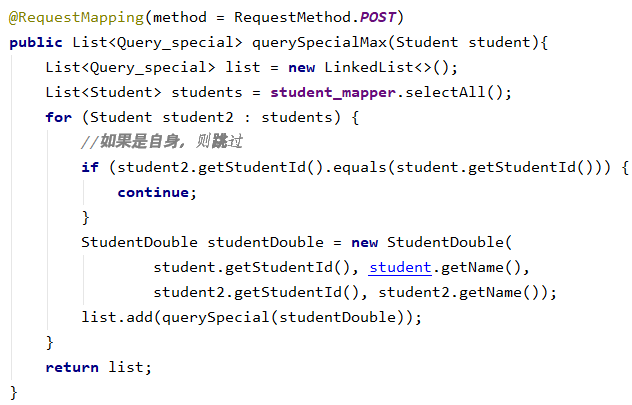
2.特殊查询操作

对于如Student表中的部分特殊查询功能，需要将Mapper中的多个存储过程合并在一起，合成一个新的查询类返回给前端。

如对于查询两个学生之间的联系时，需要同时合并查询相同选课数、查询相同老师选课数等存储操作，并给前端返回一个Query\_special类

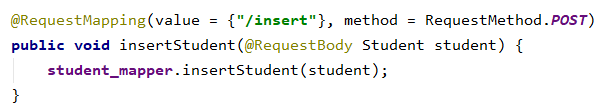


对于查询与该学生联系最紧密的学生时，需要对数据库中每一个同学进行上述关系查询，返回一个Query\_special类的list.



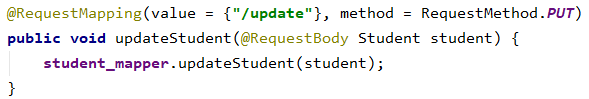
3.插入操作

对于所有基本表的插入完全一致，直接在controller中调用Mapper中对应的接口即可，根据前端传入的新表项插入到后端数据库中，无返回值。



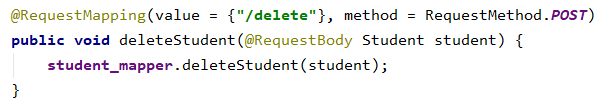
4.更新操作

对于所有基本表的更新完全一致，直接在controller中调用Mapper中对应的接口即可，根据前端传入的新表项更新数据库中对应的表项，无返回值。



5.删除操作

对于所有基本表的删除完全一致，直接在controller中调用Mapper中对应的接口即可，根据前端传入的表项删除数据库中对应的表项，无返回值。



（三）前端代码实现说明

以系管理页面为例，它作为一个总的基本页面，在导航栏放置系管理下属的各页面栏目，导向具体页面，比如教师列表、班级列表页面。



管理页面下属子页面分为三大类，数据管理，关系管理，报表分析。

数据管理中展示的为基本表的相关功能，查询，删除，修改，添加。关系管理则是涉及到两个表的相关约束，要在符合约束的条件下进行增删改查。以学生列表为例，在页面布置上由头部、搜索框、学生列表、添加按钮以及隐藏的添加信息框组成。在学生列表项的右侧布置有修改、删除按钮。

头部签出，修改个人信息，回到主页功能



按照学号或姓名进行查询的搜索功能



按每行表项的各属性信息进行展示



学生信息表框

报表分析中则是对相关数据以饼状图、柱状图、折线图的形式展现出来，形成对比，方便用户更好的观察数据分布。利用echarts组件，定义好数据样式，由后端将相应数据传回前端即可显示。

以成绩分布的饼状图为例。通过搜索课程名字或课程编号，传回成绩信息即可按照五级形式展示成绩分布饼状图。

课程成绩分布饼状图

以性别分布的柱状图为例。通过搜索系名字或系编号，传回男女人数信息即可显示以系为x轴的双列柱状图。

各系男女比例分布柱状图

以图书借阅分布的折线图为例。可以按月份统计图书的借阅情况，并在一个折线图内同时分布三条曲线，统计性别对应的借阅次数。



图书借阅按月分布折线图

七 结果展示

7.1网站主页

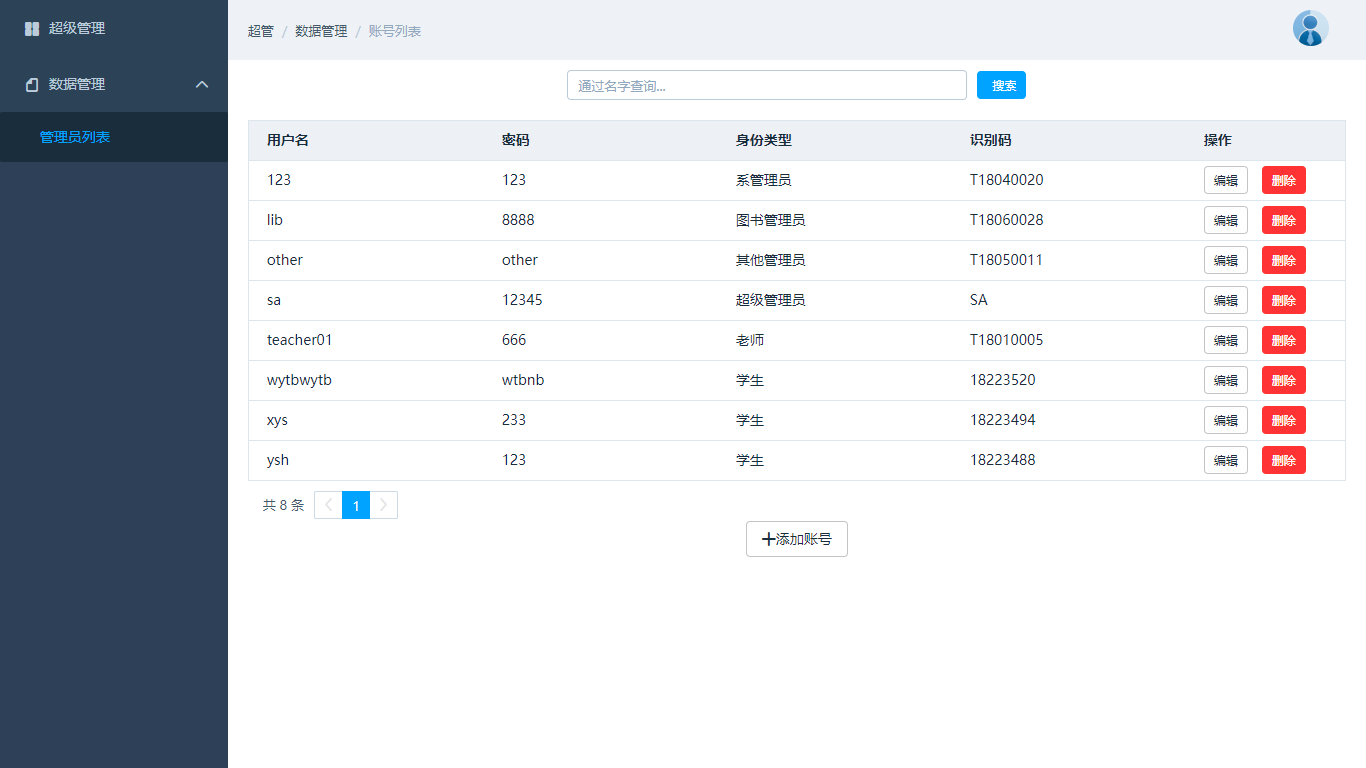
系统名为校园服务通，通过npm run dev指令，将前端运行后，即可进入数据库网站登陆页面。

对于每一位用户（教师、学生、管理员），可以通过统一认证账号的形式登陆，进入自己对应的管理或查询页面，如果是管理员，则可对自己权限内的表进行添加、修改、删除等操作，如果是学生，则可进行选课、借书、查看自己的成绩等操作，但其无权查看别人的成绩。

若一名用户为超级管理员，则他可以在登陆界面以任何身份登陆，根据选择的身份可以进入到对应的页面。

（一）超级管理员

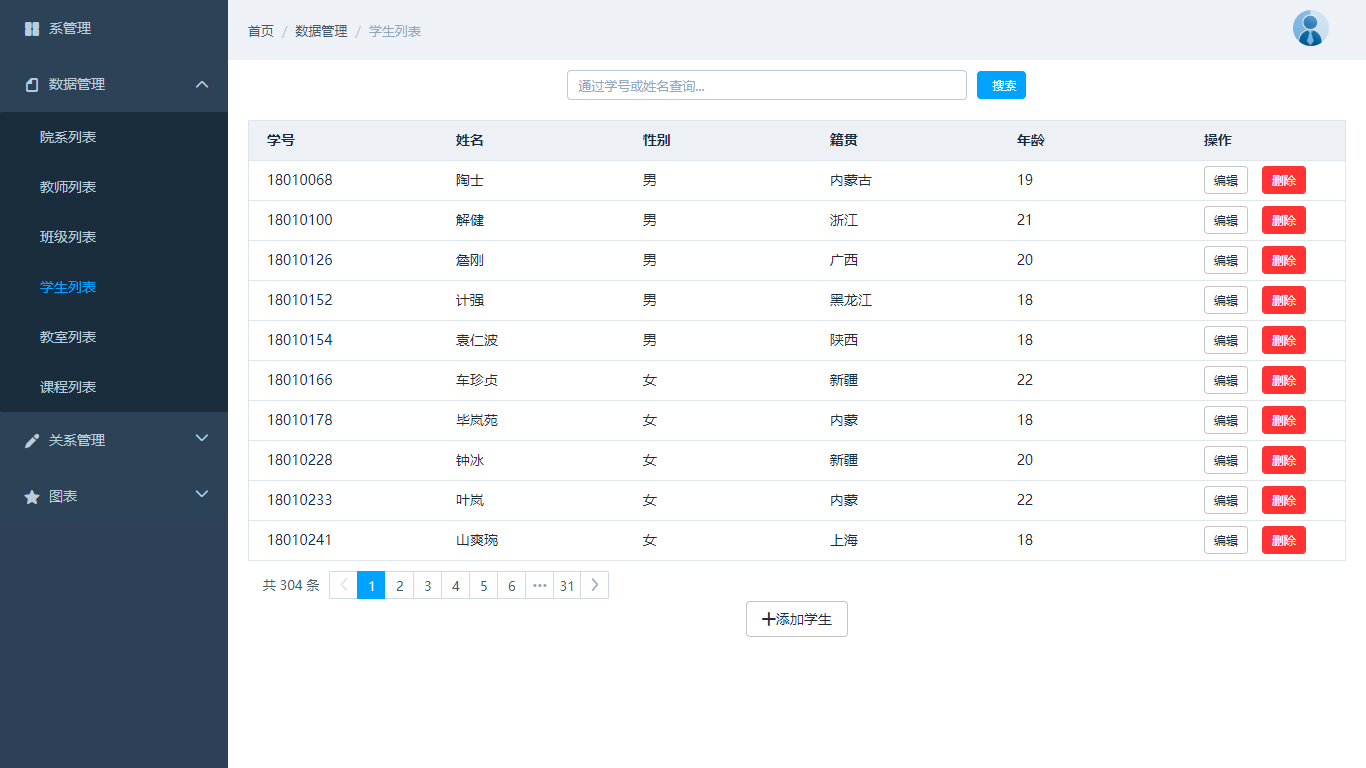
只有超级管理员才能这种身份进行登陆。若选择超级管理员身份登陆，则会进入此页面，对用户进行管理。超级管理员可以执行新增用户，更改用户身份、删除用户等操作。

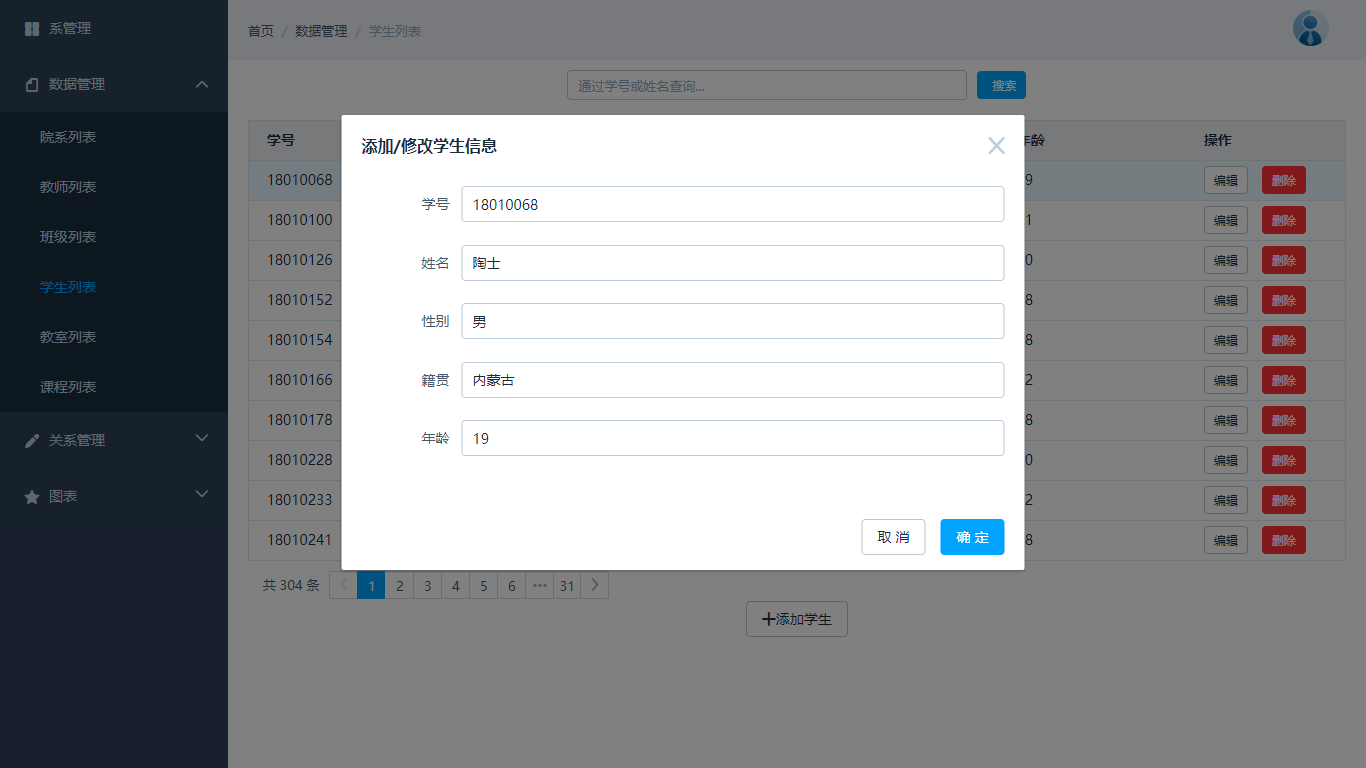


（二）系管理员

系管理员拥有院系列表、教师列表、班级列表、学生列表、教室列表和课程列表的增、删、改、查权限，也能编辑这些表之间的关系。大多数的表拥有“搜索”栏，通过在搜索栏以编号或名称的方式进行模糊匹配，即可查询所需要的信息。例如在学生列表中输入“王”，即可查找所有姓名中包含“王”的学生。

在图表页面，系管理员可以查看性别、年龄和成绩分布。若在查询栏中输入系编号或课程编号，还能够查看某一个系的性别、年龄分布或某一特定课程的成绩分布。





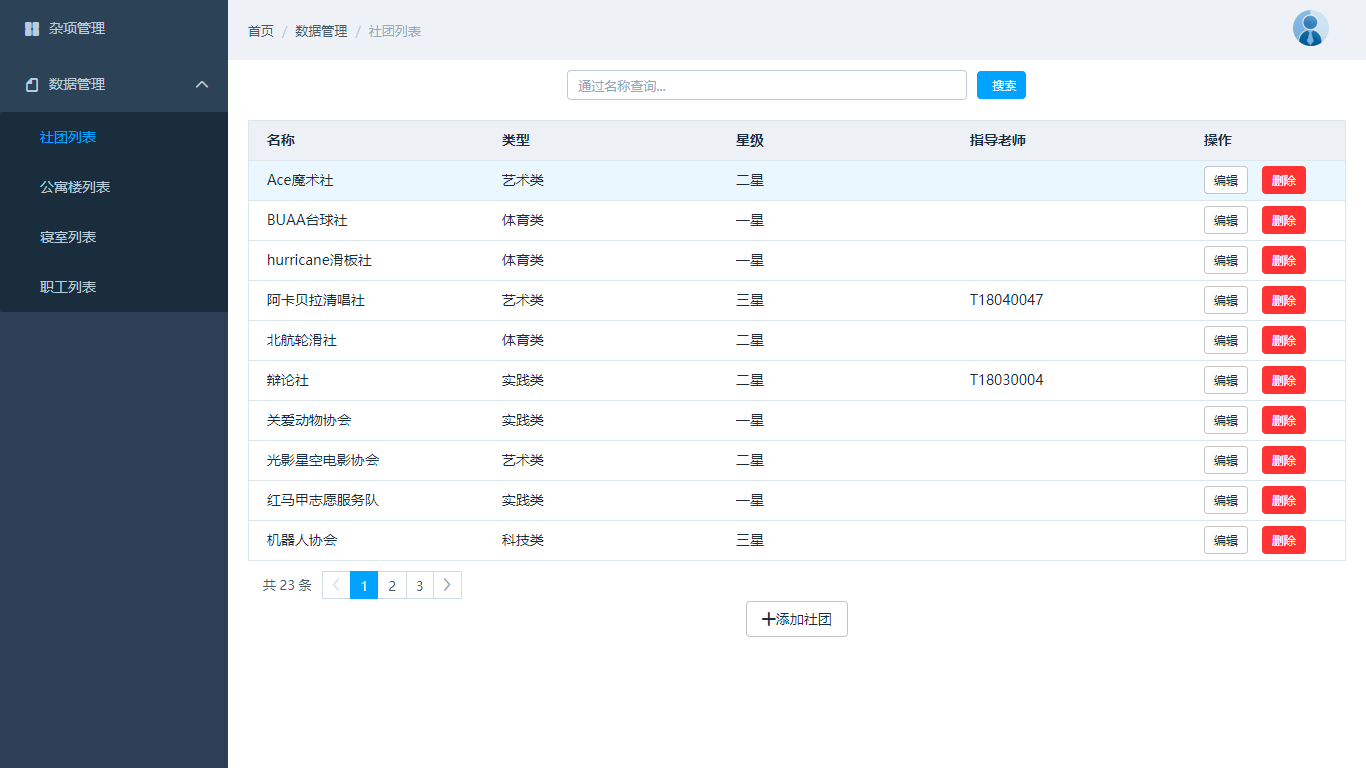
（三）图书管理员

图书管理员可以查看图书列表和学生借阅列表，进行新增图书、删除图书、修改借阅状态等操作，在学生借阅列表中，如果修改一条借书记录为“已归还”状态，图书列表中对应书目的状态将自动从“不可借阅”切换为“可借阅”。在报表分析页面，还可以查看当日借还书数据与总数据，以及查看近期借书情况的折线图。



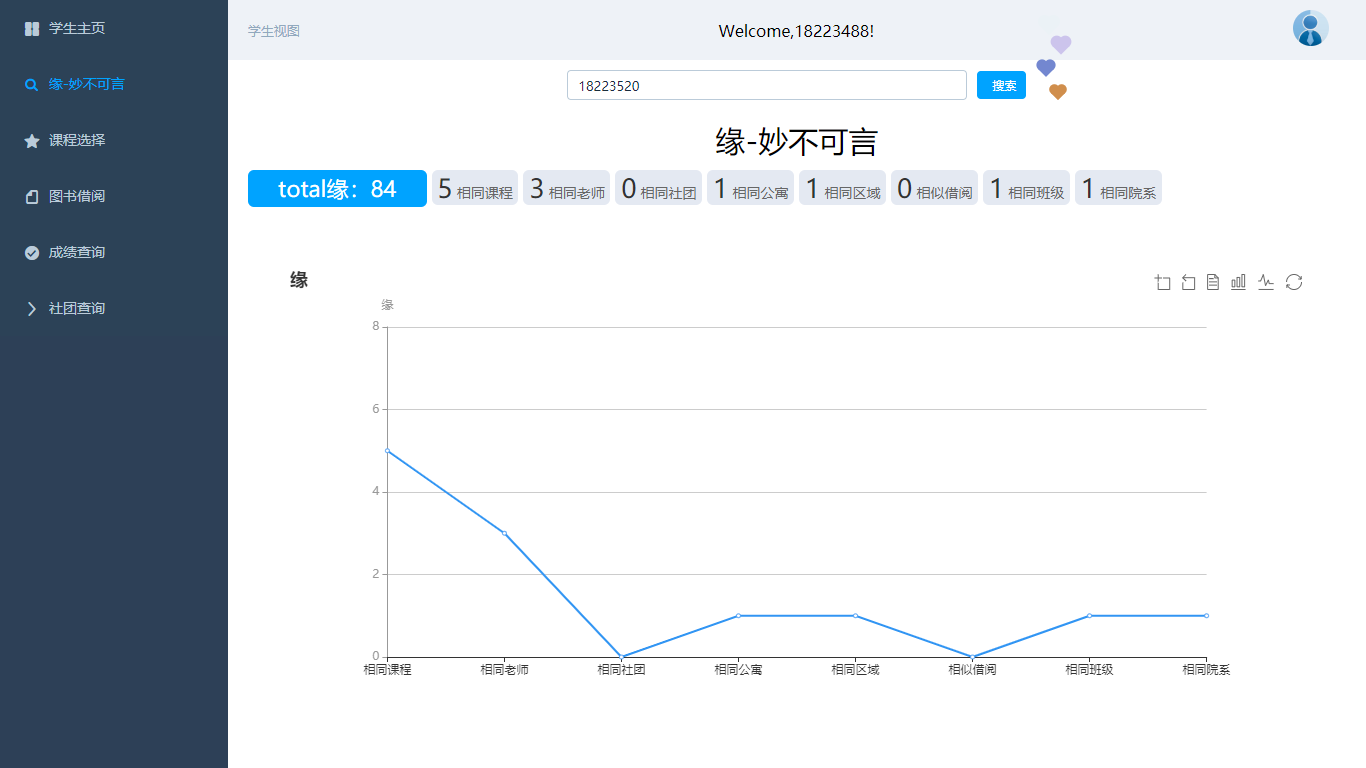
（四）其他管理员

其他管理员也是管理员身份的一种，他可以管理杂项信息，例如社团、公寓楼、寝室和职工列表。这些页面同样提供了通过编号名称进行模糊匹配查询的功能。



（五）学生

学生登陆后，即进入到学生界面，该界面顶端显示了学生的登陆名，学生在该页面可以进行选课、借书等操作。缘分计算则是充分利用数据库中的所有信息，学生可以通过输入某人的学号来查询与他的缘关系，在多个表中查出与该学生相关的信息，如相同课程，相同老师，相同社团，相同公寓，相同区域，相似借阅，相同班级，相同院系，由此八项完成对缘的计算。该页还添加了在点击时出现动态爱心特效的功能。







总结与收获