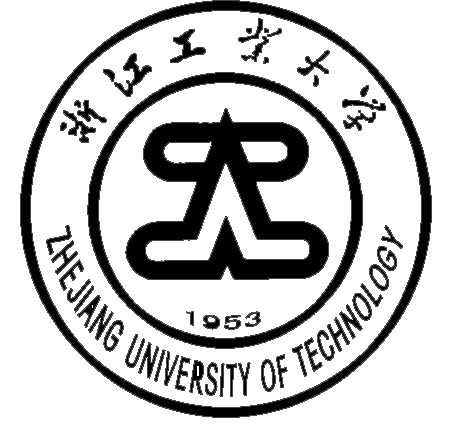


数据库系统课程设计报告

**2021短学期**



小组 **第18小组**

题目 **高校成绩管理数据库系统的设计与实现**

班级 软工1902班

学号 201906061609

姓名 吕锐

指导教师 熊丽荣

所在学院 计算机学院

提交日期 2021年7月23

目录

[1、需求分析 1](#_Toc22043)

[1.1 数据需求描述 2](#_Toc11655)

[1.2 系统功能需求 6](#_Toc29595)

[1.3 其他性能需求 7](#_Toc13257)

[2、概念结构设计 7](#_Toc5552)

[2.1 局部E-R图 7](#_Toc15258)

[2.2 全局E-R图 10](#_Toc3628)

[2.3 优化E-R图 11](#_Toc5109)

[3、逻辑结构设计 12](#_Toc27338)

[3.1 关系模式设计 12](#_Toc19374)

[3.2 数据类型定义 13](#_Toc24329)

[3.2 数据类型定义 13](#_Toc4185)

[3.3 关系模式的优化 15](#_Toc20977)

[4、物理结构设计 16](#_Toc1405)

[4.1 聚簇设计 16](#_Toc22288)

[4.2 索引设计 16](#_Toc24103)

[4.3 分区设计 16](#_Toc21538)

[5、数据库实施 17](#_Toc5215)

[5.1 基本表建立 17](#_Toc24905)

[5.2 视图的建立 20](#_Toc13589)

[5.3 索引的建立 22](#_Toc8132)

[5.4 触发器建立 22](#_Toc11375)

[5.5 建存储过程 23](#_Toc25466)

[6、应用系统开发与试运行 24](#_Toc7593)

[6.1 开发平台和开发环境介绍 24](#_Toc11741)

[6.2 前台界面与后台数据库连接说明、代码实现 24](#_Toc27369)

[6.3 系统各功能设计和运行界面截图 25](#_Toc11509)

[7、实验总结 39](#_Toc29875)

[7.1 遇到的问题和解决的办法 39](#_Toc18310)

[7.2 系统设计的不足 40](#_Toc5591)

[7.3 进一步改进思路和体会 40](#_Toc18357)

**高校成绩管理数据库系统的设计与实现**

**小组组长：吕锐** **小组成员：申屠克勤、郑昊、倪依涛**

**小组详细分工情况：**

需求分析：**吕锐**、**申屠克勤、郑昊、倪依涛**

概念结构设计及优化：**郑昊、倪依涛**

逻辑结构设计及优化：**吕锐**、**申屠克勤**

物理结构设计：**吕锐**、**申屠克勤、郑昊、倪依涛**

## 1、需求分析

计算机已经深入到日常工作和生活的方方面面，成为我们学习和工作的得力助手，比如文字处理、信息管理、辅助设计、图形图像处理、教育培训以及游戏娱乐等。随着越来越多的应用软件出现，人们对它的要求也越来越高；虽然现在世界上的各种软件层出不穷，但它们依然不能满足用户的各种特殊需要，所以人们仍是不得不开发适合特殊需求的软件。高校成绩管理系统记录了一个大学生成绩的系统,它的出现使得查询、更新、插入简单化，高效化，成本也随之大大减少。使用计算机对成绩信息的管理,具有手工管理所 无法比拟的优点：信息存储及时，检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。这些优点能够极大地提高学 生成绩管理的效率,也是高校成绩正规化管理的重要途径。

对于某校管理学生成绩的工作人员，根据实际工作需要，提出了以下数据和业务处理需求：

1、学校设置了各专业，在专业下开设班级，每个班级包含若干学生，学生信息至少需要包含学号、姓名、性别、年龄、生源所在地、已修学分总数等数据项；另外，需要有地区信息，用于统计某一地区的学生数；

2、课程信息表至少需包含课程编号、课程名称、任课教师、开课学期、学时、考试或考查、学分等数据项，课程根据班级开设。

3、教师信息至少需要包含教师编号、姓名、性别、年龄、职称、联系电话等数据项；

4、学生成绩至少需要学号，学期，课程名称，成绩，任课老师等数据项；

5、需要实现以下业务处理和查询功能：

学生成绩按每学年成绩统计

学生成绩名次排定

每门课程平均成绩统计

学生所学课程及学分统计

对每个学生输入成绩的时候，自动生成学生的已修学分总数

学生成绩查询

教师任课查询

班级课程开设查询

以SQL SERVER数据库管理系统为平台，通过设计数据库概念模型、逻辑模型以及利用标准SQL语言的数据库实现，掌握关系数据库系统的设计与实现方法，增强数据库设计和数据库应用系统开发能力。

### 数据需求描述

#### 数据流图

因为此次课程设计的主要是高校成绩管理系统，主要的功能模块有三个部分：学生、老师和管理员。因而可构建的数据流图如下图所示为：

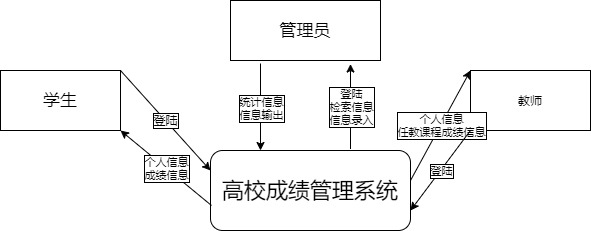


图1.1.1.1顶级数据流系统

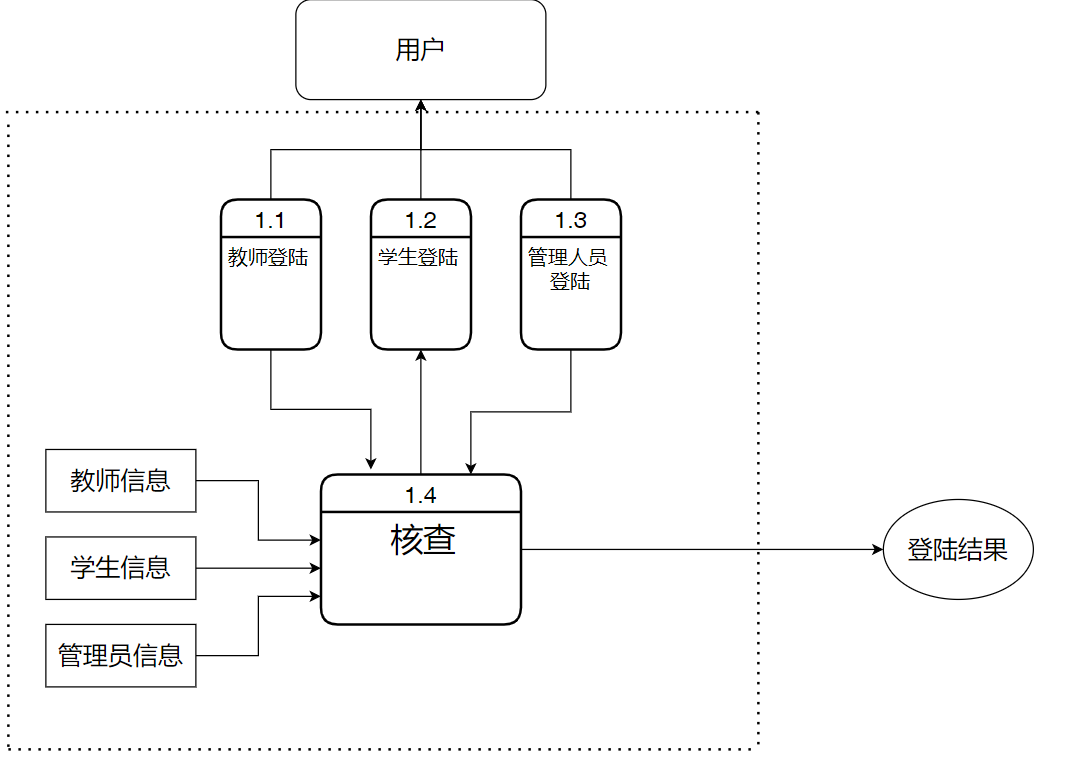


图1.1.1.2初级数据流

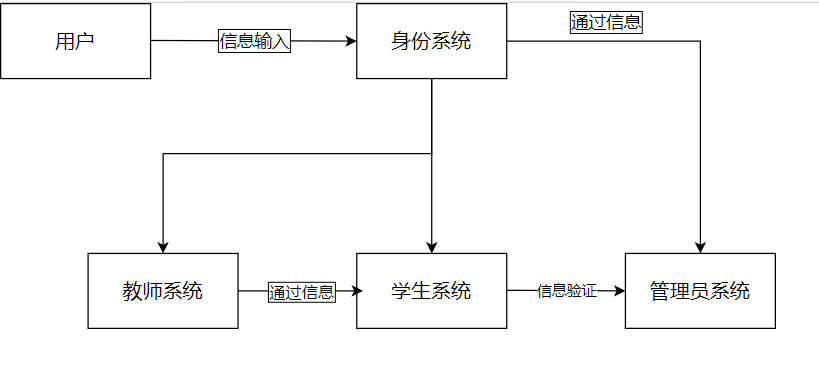


图1.1.1.3基础数据流向

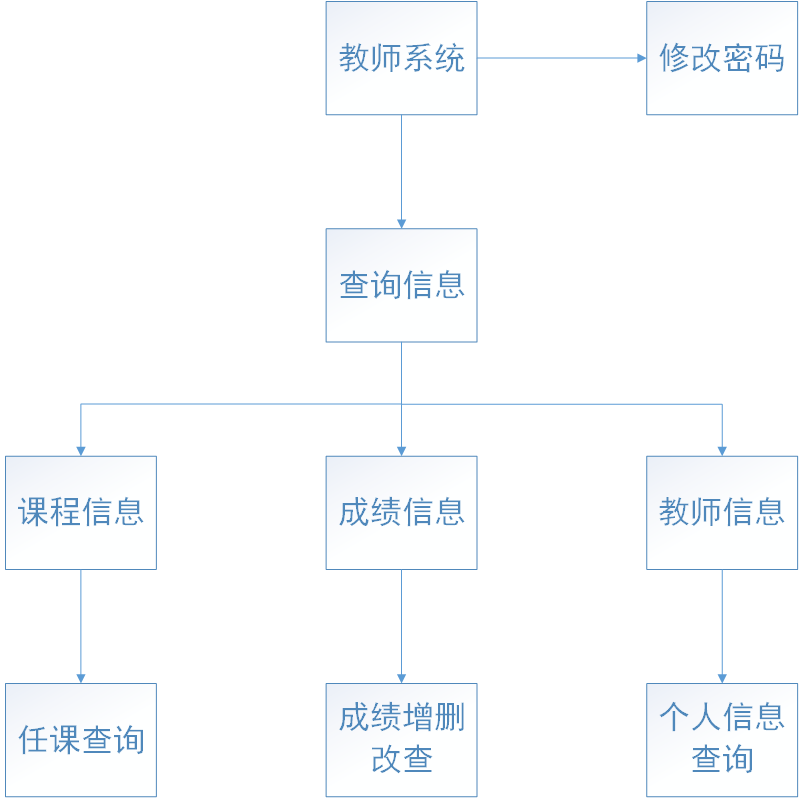


图1.1.1.4教师系统

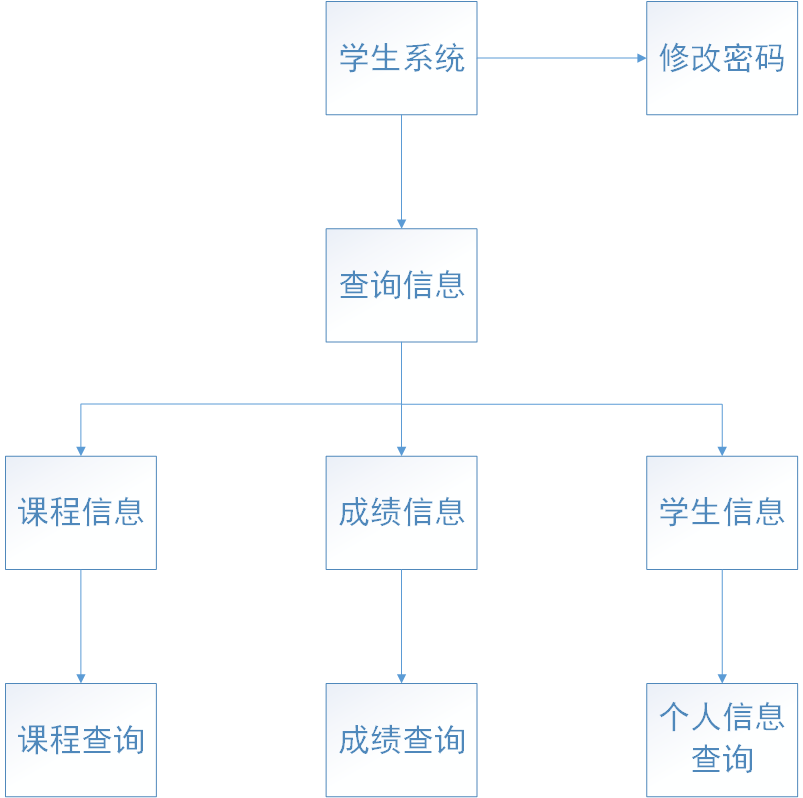


图1.1.1.5学生系统

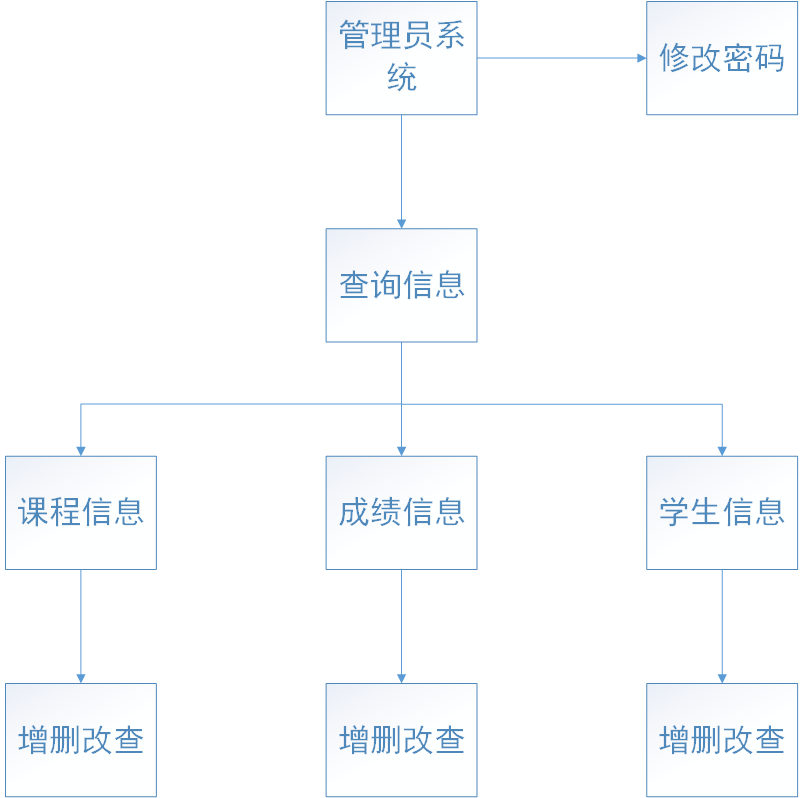


图1.1.1.6管理员系统

#### 数据字典

|  |  |
| --- | --- |
| 名称 | 学生 |
| 描述 | 记录学生信息 |
| 组成 | 学号、姓名、性别、年龄、生源地 |

|  |  |
| --- | --- |
| 名称 | 教师 |
| 描述 | 记录教师信息 |
| 组成 | 工号、姓名、性别、年龄、职称、电话 |

|  |  |
| --- | --- |
| 名称 | 课程 |
| 描述 | 记录课程信息 |
| 组成 | 课程名，开设学期，学时，学分，考核方法 |

|  |  |
| --- | --- |
| 名称 | 班级排课 |
| 描述 | 记录班级排课信息 |
| 组成 | 班级、课程 |
| 数据流来源 | 教务处 |
| 数据流去向 | 学生 |

|  |  |
| --- | --- |
| 名称 | 课程成绩 |
| 描述 | 记录学生课程成绩信息 |
| 组成 | 学号、课程、成绩 |
| 数据流来源 | 教师 |
| 数据流去向 | 学生 |

|  |  |
| --- | --- |
| 名称 | 教师任课 |
| 描述 | 记录教师课程授课信息 |
| 组成 | 工号、课程 |
| 数据流来源 | 教务处 |
| 数据流去向 | 教师 |

### 1.2 系统功能需求

此高校管理系统主要分为四个模块：用户登录模块、学生模块、教师模块和管理员模块。可采用层次数据模型进行表达，主要如下图所示：

图1.2.1 所有功能模块

该管理系统主要实现的功能有：

1. 学生信息管理
2. 教师信息管理
3. 班级信息管理
4. 课程信息管理
5. 学生成绩录入
6. 班级排课管理
7. 教师任课管理
8. 学生成绩查询
9. 学生成绩名次排定
10. 课程平均成绩统计
11. 学生生源地人数统计

### 1.3 其他性能需求

（1） 能够保证数据的独立性。数据和程序相互独立有利于加快软件开发速度，节省开发费用。

（2） 冗余数据少，数据共享程度高。

（3） 系统的用户接口简单，用户容易掌握，使用方便。

（4） 能够确保系统运行可靠，出现故障时能迅速排除；能够保护数据不受非受权者访问或破坏；能够防止错误数据的产生，一旦产生也能及时发现。

（5） 有重新组织数据的能力，能改变数据的存储结构或数据存储位置，以适应用户操作特性的变化，改善由于频繁插入、删除操作造成的数据组织零乱和时空性能变坏的状况。

（6） 具有可修改性和可扩充性。

（7） 能够充分描述数据间的内在联系。

## 2、概念结构设计

### 2.1 局部E-R图

（1）学生局部E=R图

学生包含学号、姓名、班级编号、性别、年龄、已修学分、生源地属性以及登录密码八个属性。因为一个班级包含多个学生，一个学生只属于一个班级，所以学生实体集和班级实体集有1：n联系；因为一个学生可以选修多门课程，一个课程可以被多个学生选修，所以学生和课程之间有n：m联系；因为一个学生包含自己的成绩统计、学分统计以及班级课表查询统计，而且该统计功能只属于一个学生，所以学生和统计有1：n联系。

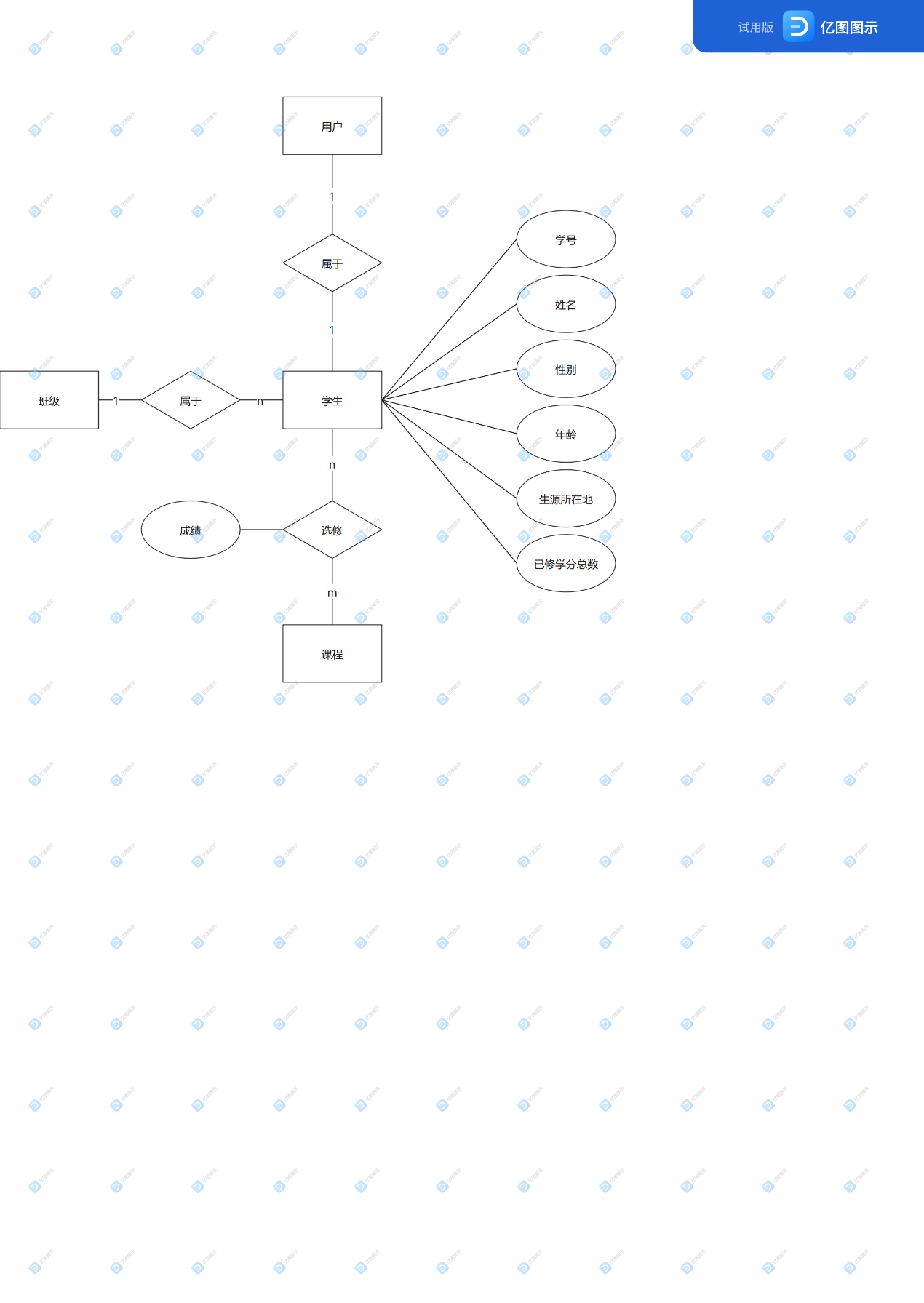


图2.1.1 学生局部E-R图

（2）教师局部E-R图

教师包含教师编号、姓名、性别、年龄、职称、联系电话、以及登录密码属性。因为课程可以在多个班开课、每个班可以开多门课，一门课可以被多个老师教，一个老师也可以教多门课，一个老师可以在多个班教课、一个班有多个老师授课。所以课程、教师、班级有n：m联系。

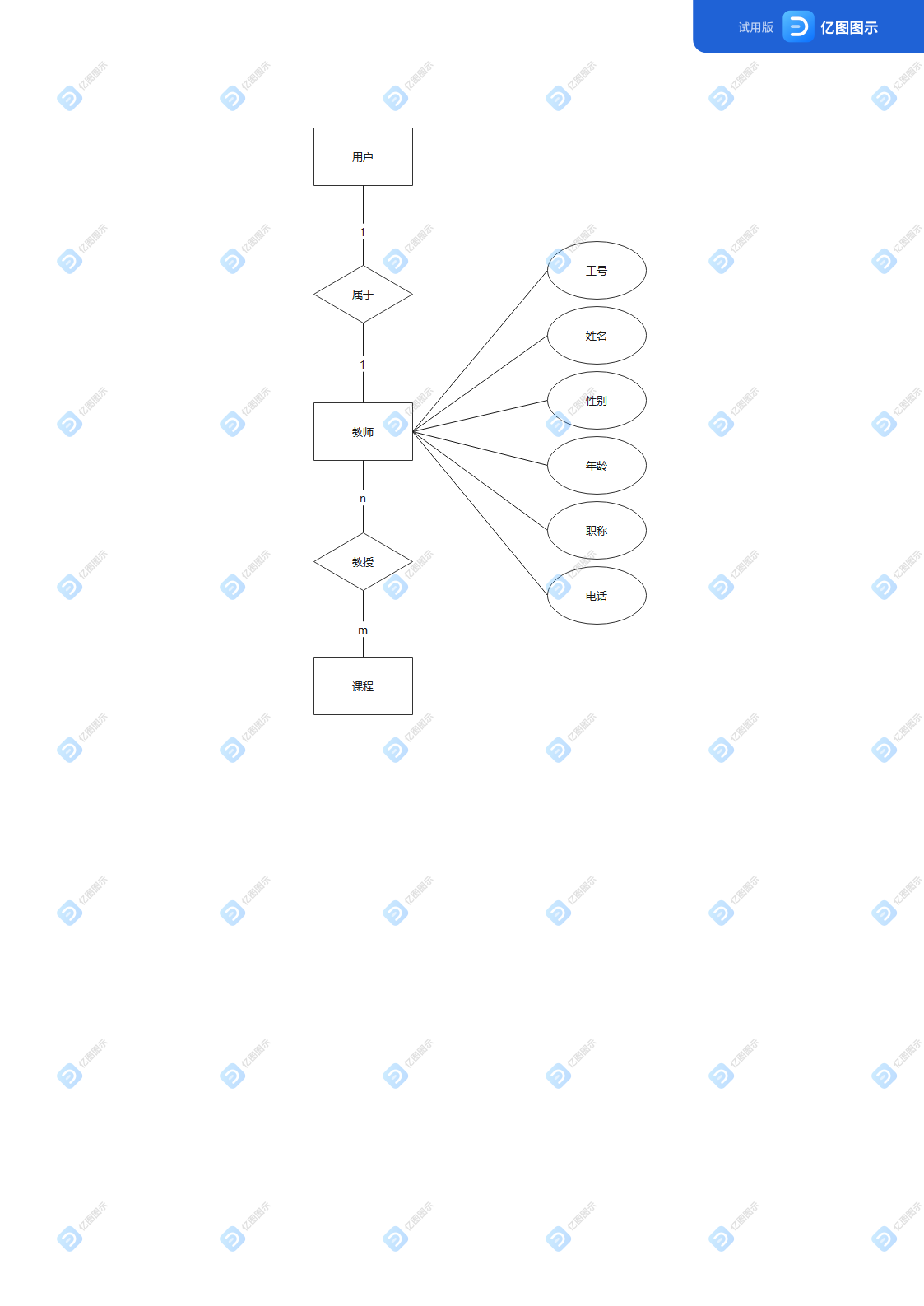


图2.1.2 教师局部E-R图

（3）成绩和课程局部E-R图

课程包含课程编号、课程名称、开课学期、学时、考试或考查、学分属性。因为一个学生可以选修多门课程，一个课程可以被多个学生选修，所以学生和课程之间有n：m联系；因为课程可以在多个班开课、每个班可以开多门课，一门课可以被多个老师教，一个老师也可以教多门课，一个老师可以在多个班教课、一个班有多个老师授课。所以课程、教师、班级有n：m联系。

成绩统计包含绩点、学期、排名属性。因为一个学生包含自己的成绩统计，一个成绩统计只属于一个学生，所以学生和成绩统计有1：n联系。

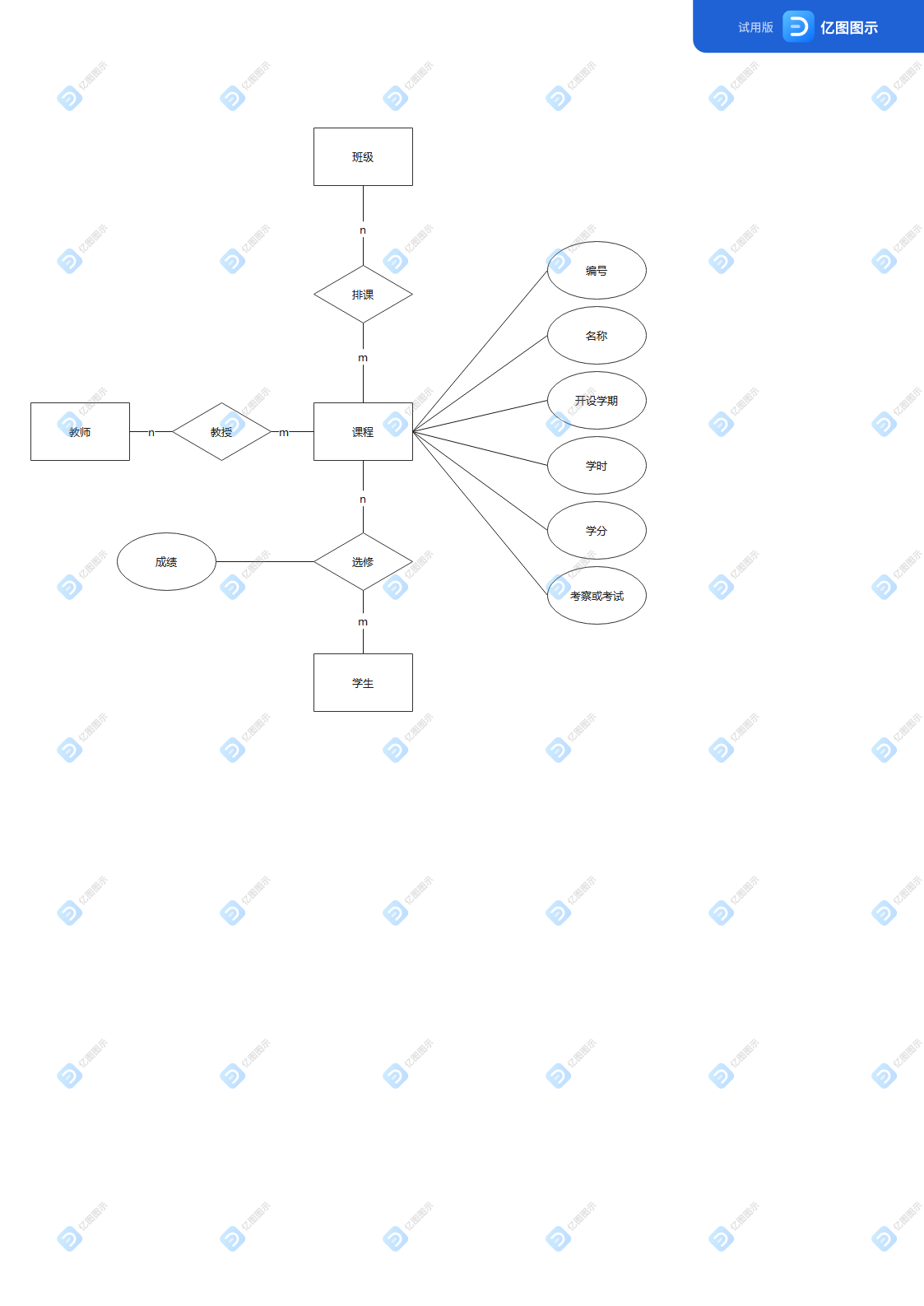


图2.1.3 成绩和课程局部E-R图

（4）专业和班级局部E-R图

班级包含班级编号和班级名称属性。因为一个班级只属于一个专业，一个专业包含多个班级，所以专业和班级有1：n的联系;因为一个班级包含多个学生，一个学生只属于一个班级，所以班级和学生是1：n的关系。

专业包含专业编号和专业名称属性。因为一个班级只属于一个专业，一个专业包含多个班级，所以专业和班级有1：n的联系。



图2.1.4 专业和班级局部E-R图

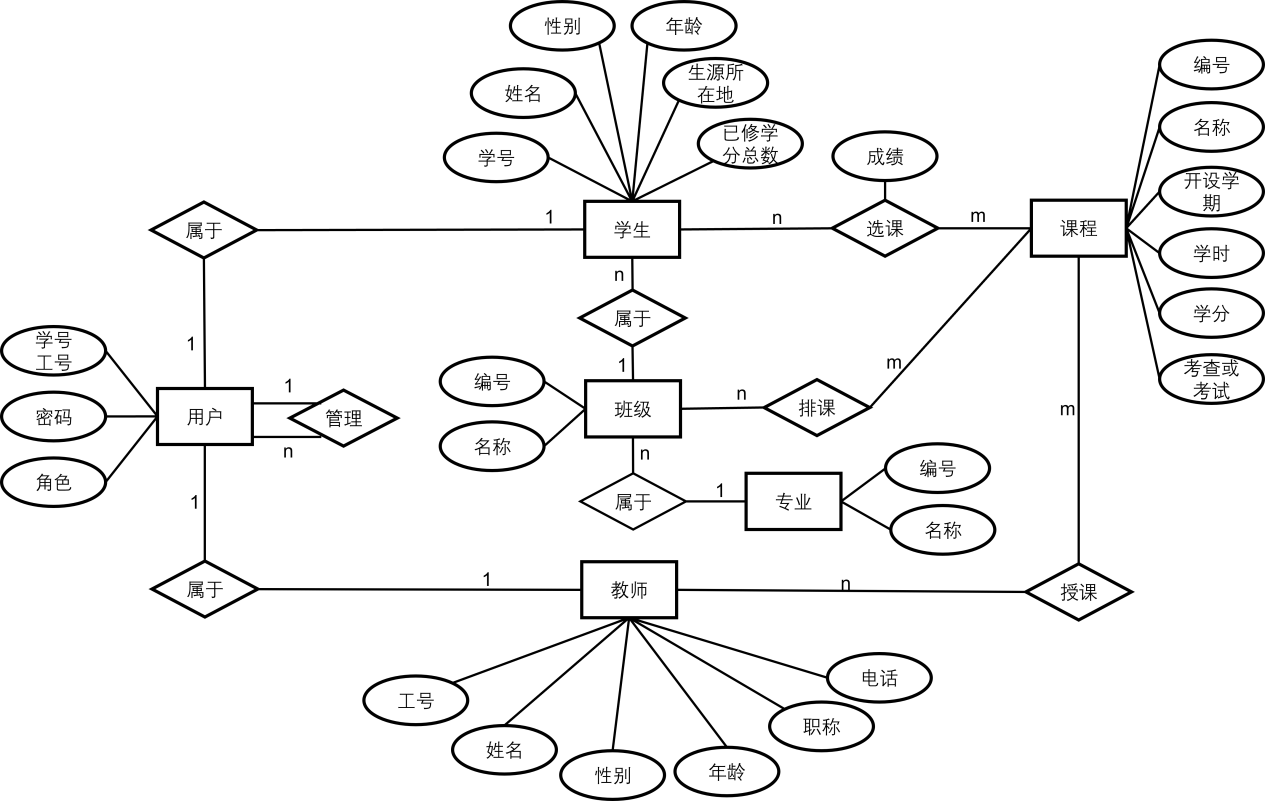
### 2.2 全局E-R图

学生拥有多个属性，且多个学生共同组成了一个班级，而多个班级又共同组成了一个专业。而一个班级又会同时开设多门课程，多门课程可能会同时由一位老师进行教授。每位学生会同时上多门课程。

同一门课可能需要多个教师上课，一个教师在一学期可能上多门课程。该联系在E-R图中表示为开课联系，具有平均成绩属性。教师任课查询、班级课程开设查询、每门课程平均成绩统计这三项功能可以通过该联系实现。

学生与课程之间存在选修联系，该联系具有成绩属性。

学生成绩按每学年成绩统计、学生所学课程及学分统计、学生成绩查询、对每个学生输入成绩的时候，自动生成学生的已修学分总数、学生成绩名次排定这五项功能可以通过该联系实现。

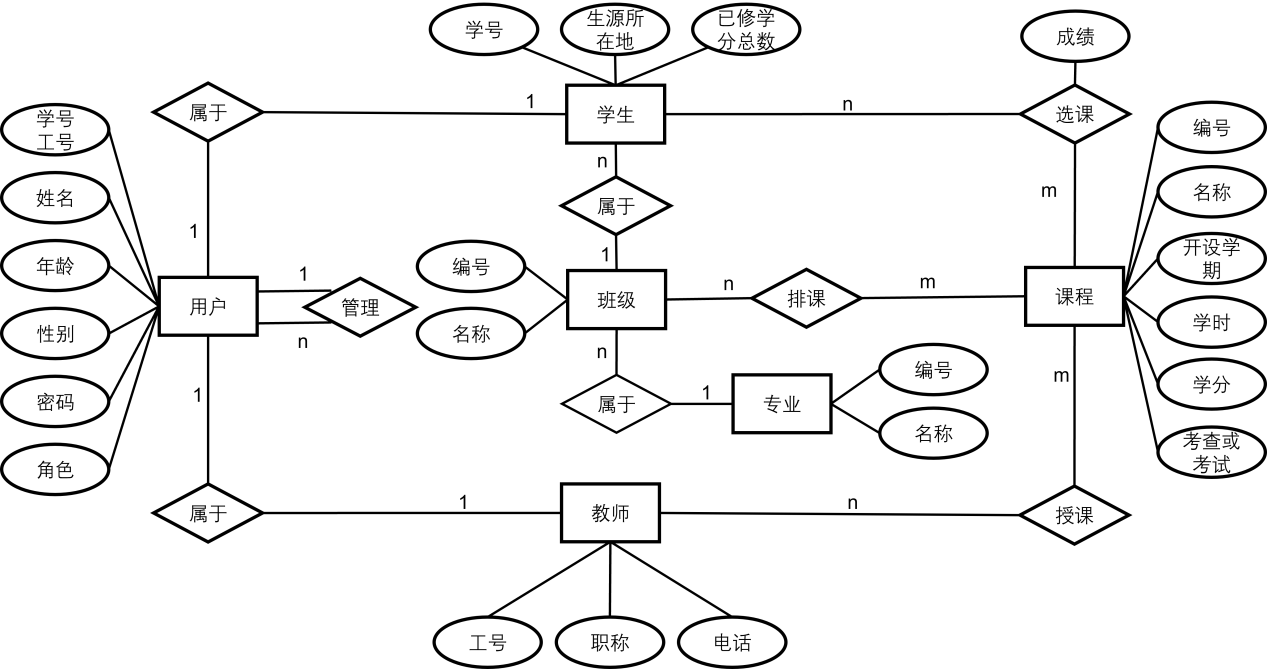


### 2.3 优化E-R图

学生与教师有多个属性重叠，将其合并并移动到用户表内。

消除各个局部E-R图之间的属性冲突、命名冲突、结构冲突

删去冗余的部分以提高范式等级



## 3、逻辑结构设计

### 3.1 关系模式设计

（1）建立学生关系模式

班级与学生之间是1对n联系，所以在学生关系模式中添加班级编号的属性

其中学号参照用户表中的编号，为外键。

得到关系模式：学生（学生编号，生源地区编号，已修学分，班级编号）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Student18(lvr\_Sno18,lvr\_Splace18,lvr\_SCredits18,lvr\_Ano18)

1. 建立教师关系模式

其中教师编号参照用户表中的编号，为外键。

关系模式：教师（教师编号，职称，电话）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Teacher18(lvr\_Tno18,lvr\_Ttitle18,lvr\_Tphone18)

1. 建立课程关系模式

关系模式：课程（课程编号，课程名称，学分，学时，开课学期，考核方式）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Course18(lvr\_Cno18,lvr\_Cname18,lvr\_Ccredits18,lvr\_Cclasshour18,lvr\_Cterm18,lvr\_Cexamineway18)

1. 建立专业关系模式

关系模式：专业（专业编号，专业名称）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Major18(lvr\_Mno18,lvr\_Mname18)

（5）建立班级关系模式

专业与班级之间是1对n联系，所以在班级的关系模式中添加专业编号属性得到关系模式：行政班级（行政班级编号，班级名称，专业编号）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_AdminClass18(lvr\_Ano18,lvr\_Aname18,lvr\_Mno18)

（6）建立用户关系模式

E-R图中管理员具有编号、姓名、年龄、性别、密码、角色属性

得到关系模式：管理员账号（编号，姓名，年龄，性别，密码，身份）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_User18(lvr\_Uno18,lvr\_Uname18,lvr\_Uage18,lvr\_Usex18,lvr\_Upasswords18,lvr\_Uchar18)

（7）建立授课关系模式

生成关系模式：授课（课程编号，教师编号）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Teaching(lvr\_Cno18,lvr\_Tno18)

1. 建立学生成绩关系模式

生成关系模式：成绩（课程编号，学生编号，成绩）

使用英文或拼音字母命名方式，转换得到关系模式为：

Lvr\_Report(lvr\_Cno18,lvr\_Sno18,lvr\_Rscore18)

（9）建立排课关系模式

生成关系模式：排课（班级编号，课程编号）

使用英文或拼音字母命名方式，转换得到关系模式为

Lvr\_Lesson(lvr\_Ano,lvr\_Cno)

### 3.2 数据类型定义

### 3.2 数据类型定义

（1）专业表

|  |  |  |  |
| --- | --- | --- | --- |
| **专业 Major** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 专业编号 | NCHAR | 10 | 主键、非空、唯一 |
| 专业名称 | NCHAR | 10 | 非空 |

（2）班级表

|  |  |  |  |
| --- | --- | --- | --- |
| **班级 Class** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 班级编号 | NCHAR | 10 | 主键、非空、唯一 |
| 班级名称 | NCHAR | 10 | 非空 |
| 专业编号 | NCHAR | 10 | 非空，外键 |

（3）用户表

|  |  |  |  |
| --- | --- | --- | --- |
| **用户 User** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 用户编号 | NCHAR | 10 | 主键、非空、唯一 |
| 姓名 | NCHAR | 10 | 非空 |
| 性别 | NCHAR | 10 | 非空 |
| 年龄 | INT |  | 非空 |
| 密码 | NCHAR | 256 | 非空 |
| 角色 | NCHAR | 10 | 非空 |

（4）学生表

|  |  |  |  |
| --- | --- | --- | --- |
| **学生 Student** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 学号 | NCHAR | 10 | 主键、非空、唯一、外键 |
| 生源所在地 | NCHAR | 10 | 非空 |
| 已修学分 | INT |  | 默认值0 |
| 班级编号 | NCHAR | 10 | 非空，外键 |

（5）教师表

|  |  |  |  |
| --- | --- | --- | --- |
| **教师 Teacher** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 工号 | NCHAR | 10 | 主键、非空、唯一、外键 |
| 职称 | NCHAR | 10 | 非空 |
| 电话 | NCHAR | 10 | 非空 |

（6）课程表

|  |  |  |  |
| --- | --- | --- | --- |
| **课程 Course** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 课程编号 | NCHAR | 10 | 主键、非空、唯一 |
| 课程名称 | NCHAR | 10 | 非空 |
| 开设学期 | NCHAR | 10 | 非空 |
| 学时 | INT |  | 非空 |
| 学分 | INT |  | 非空 |
| 考试或考查 | NCHAR | 10 | 非空 |

（7）开课表

|  |  |  |  |
| --- | --- | --- | --- |
| **开课 ClassCourse** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 班级编号 | NCHAR | 10 | 主键、非空、外键 |
| 课程编号 | NCHAR | 10 | 主键、非空、外键 |

（8）选课成绩表

|  |  |  |  |
| --- | --- | --- | --- |
| **选课成绩 StudentCourse** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 学号 | NCHAR | 10 | 主键、非空、外键 |
| 课程编号 | NCHAR | 10 | 主键、非空、外键 |
| 成绩 | INT |  |  |

（9）教师授课表

|  |  |  |  |
| --- | --- | --- | --- |
| **教师授课 TeacherCourse** | | | |
| **数据项名** | **数据类型** | **数据长度** | **完整性约束** |
| 工号 | NCHAR | 10 | 主键、非空、外键 |
| 课程编号 | NCHAR | 10 | 主键、非空、外键 |

### 3.3 关系模式的优化

用户Lvr\_User18表的函数依赖为：用户编号→姓名，性别，年龄，密码，角色；其中候选键为用户编号。经判断为BCNF。

教师Lvr\_Teacher18表的函数依赖为：工号→职称，电话；其中候选键为工号。经判断为BCNF。

学生Lvr\_Student18表的函数依赖为：学号→生源所在地，所修学分，班级编号；其中候选键为学号。经判断为BCNF。

班级Lvr\_Class18表的函数依赖为：班级编号→班级名称，专业编号；其中候选键为班级编号。经判断为BCNF。

专业Lvr\_Major18表的函数依赖为：专业编号→专业名称；其中候选键为专业编号。经判断为BCNF。

课程Lvr\_Course18表的函数依赖为：课程编号→课程名称，开设学期，学时，学分，考试或考查；其中候选键为课程编号。经判断为BCNF。

班级课程排课Lvr\_ClassCourse18表无函数依赖。经判断为BCNF。

学生课程成绩Lvr\_StudentCourse18表的函数依赖为：学生编号，课程编号→成绩；其中候选键为学生编号，课程编号。经判断为BCNF。

教师课程任课Lvr\_TeacherCourse18表无函数依赖。经判断为BCNF。

## 4、物理结构设计

### 4.1 聚簇设计

聚簇是将有关的数据元组集中存放于一个物理块内或若干相邻物理块内或同一柱面内，以提高查询效率的数据存储结构。

当一个关系按照某些属性列建立聚簇后，关系中的元组都按照聚簇属性列的顺序存放在磁盘的一个物理块或若干相邻物理块内，因此，对这些属性列的查询特别有效，可以明显提高查询效率。

学生关系模式（Lvr\_Student18）的学生编号（lvr\_Sno18），

教师关系模式（Lvr\_Teacher18）的教师编号（lvr\_Tno18），

课程关系模式（Lvr\_Course18）的课程编号（lvr\_Cno18），

班级关系模式Lvr\_AdminClass18的课程编号（lvr\_Ano18)，

专业关系模式Lvr\_Major18的专业编号（lvr\_Mno18），

用户关系模式Lvr\_Admin18的用户编号（lvr\_Uno18），

这些属性都是其所在关系模式的主键，会被大范围的搜索，在许多查询中也都要用到。因此设置为主键。在mysql有主键的情况下，主键就是聚簇索引

### 4.2 索引设计

索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息。

对于一个确定的关系，通常在下列情况下可以考虑建立索引：

1. 在主属性列和外键属性列上通常都可以分别建立索引，不仅有助于唯一性检测和完整性检查，而且可以加快连接查询的速度
2. 以查询为主的关系可以建立尽可能多的索引。
3. 对等值连接，但满足条件的元组较少的查询可考虑建立索引。

如果查询可以从索引直接得到结果而不必访问关系，则对此种查询可建立索引。例如，为查询某个属性的MIN,MAX,AVG,SUM,COUNT等函数值，可在该属性列上建立索引。

根据以上要求，对班级表中的专业编号和学生表中的班级编号这种不作为主键但需要频繁查询的建立索引。

班级表（Lvr\_AdminClass18）对专业编号（Lvr\_Mno18）建立索引

学生表（Lvr\_Student18）对班级编号（Lvr\_Ano18）建立索引

### 4.3 分区设计

关于数据库文件和日志文件的分区问题。

磁盘分区设计的一般原则：

减少访问冲突，提高I/O并发性。多个事物并发访问同一磁盘时，会产生磁盘访问冲突而导致效率低下，如果事务访问数据均能分布于不同磁盘上，则I/O可并发执行，从而提高数据库访问速度。

分散热点数据，均衡I/O负担。在数据库中数据访问的频率是不均匀的，那些经常被访问的数据成为热点数据，此类数据宜分散存在于不同的磁盘上，以均衡各个磁盘的负荷，充分发挥多磁盘的并行操作的优势。

保证关键数据快速访问，缓解系统瓶颈。在数据库中有些数据如数据字典等的访问频率很高，为保证对它的访问不直接影响整个系统的效率，可以将其存放在某一固定磁盘上，以保证其快速访问。

该成绩管理系统由于程序较小，所以不进行分区设计。

## 5、数据库实施

### 5.1 基本表建立

**①学生表**

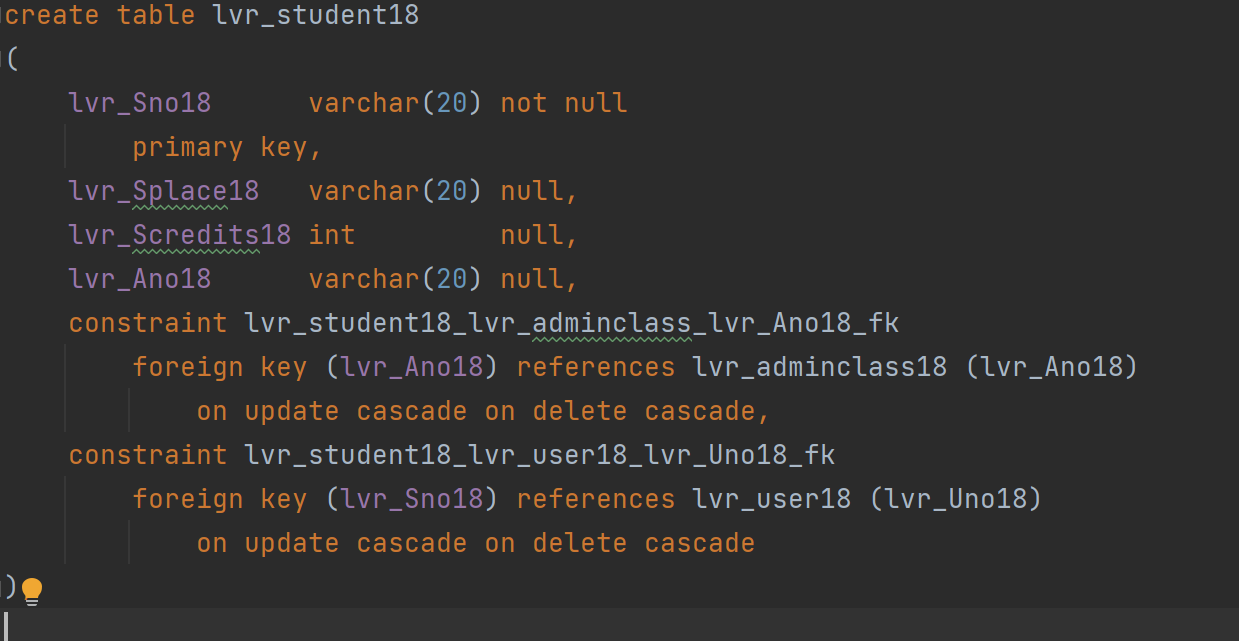


图5-1 学生表

**②班级表**

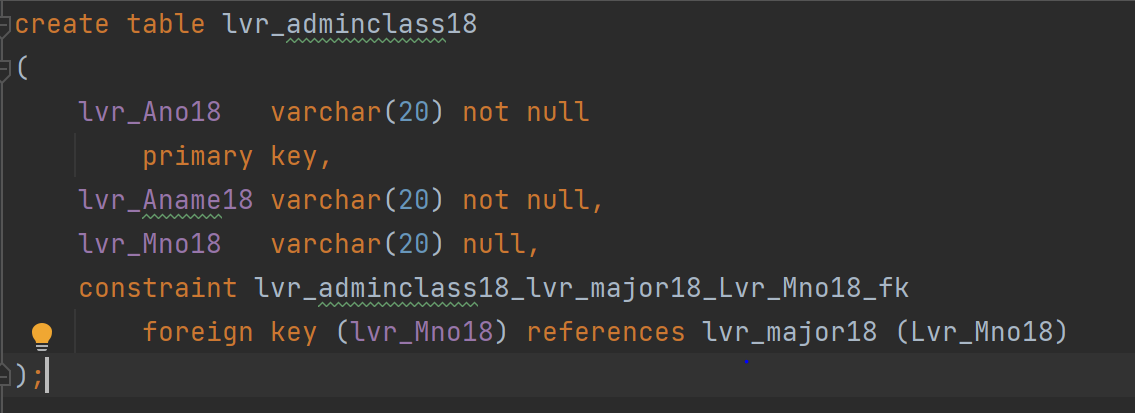


图5-2 班级表

**③专业表**

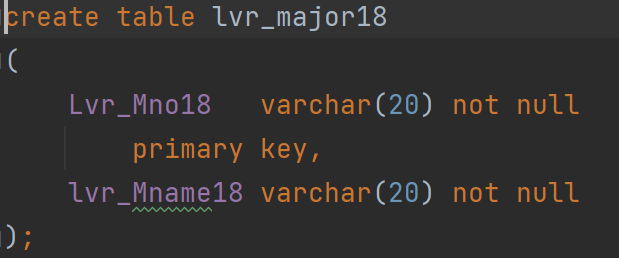


图5-3 专业表

**④教师表**

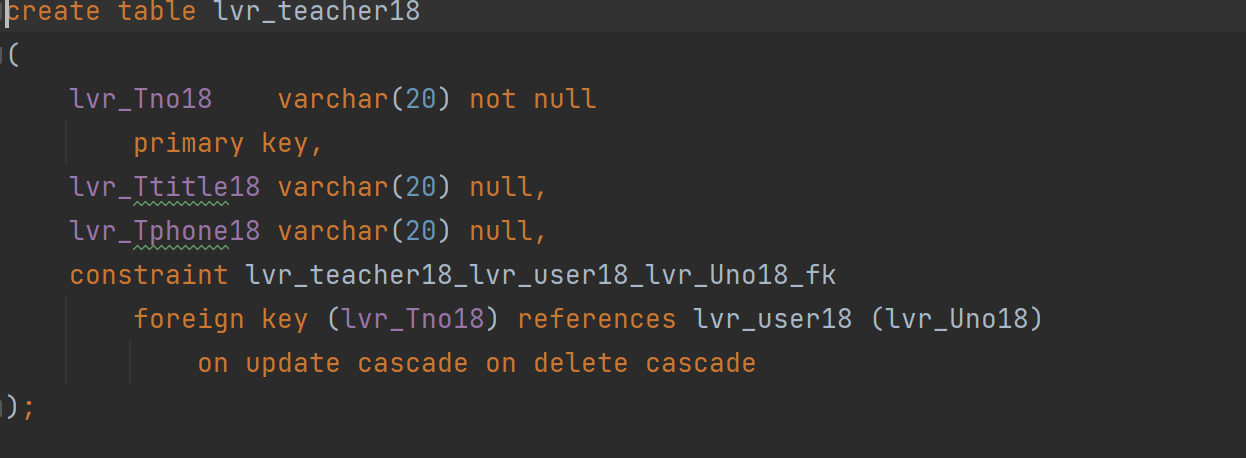


图5-4 教师表

**⑤课程表**

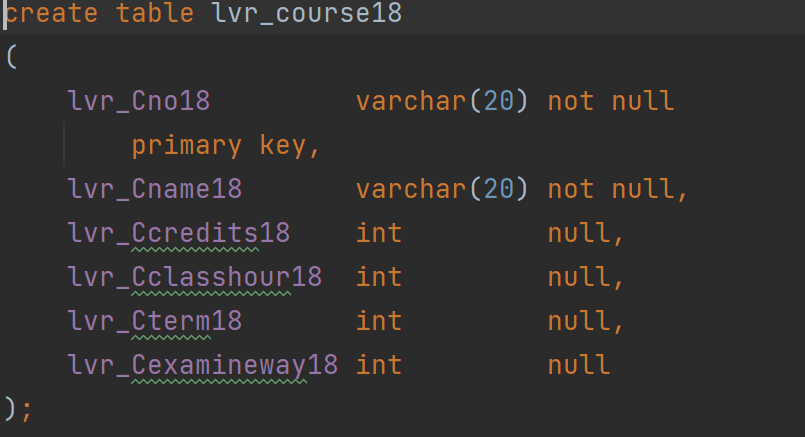


图5-5 课程表

插入数据：

INSERT

**⑥成绩表**

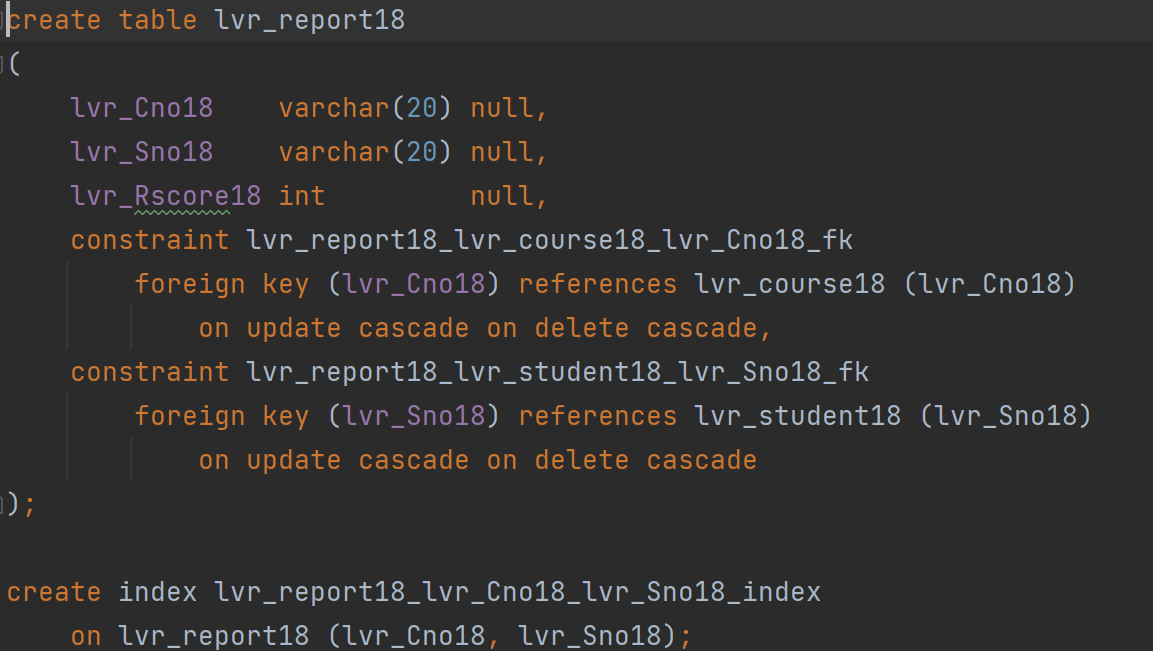


图5-6 成绩表

**⑦授课表**

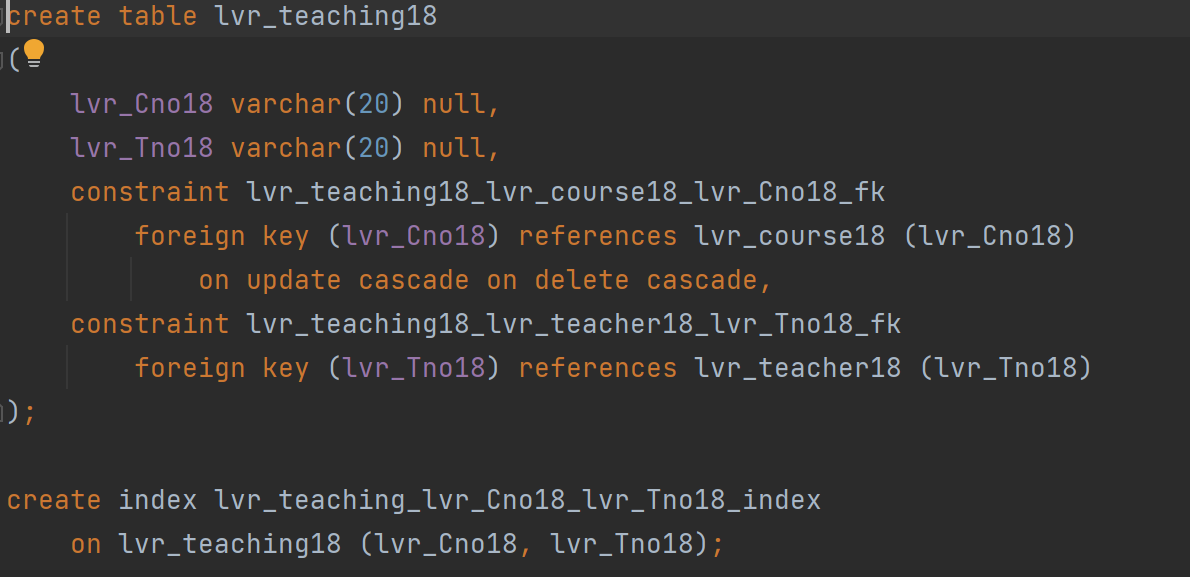


图5-7 授课表

**⑧用户表**

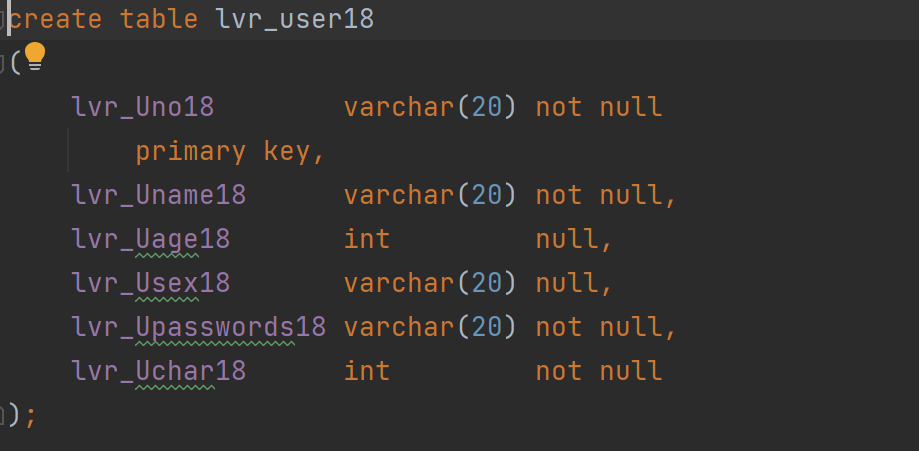


图5-8 管理员表

**⑨开课表**

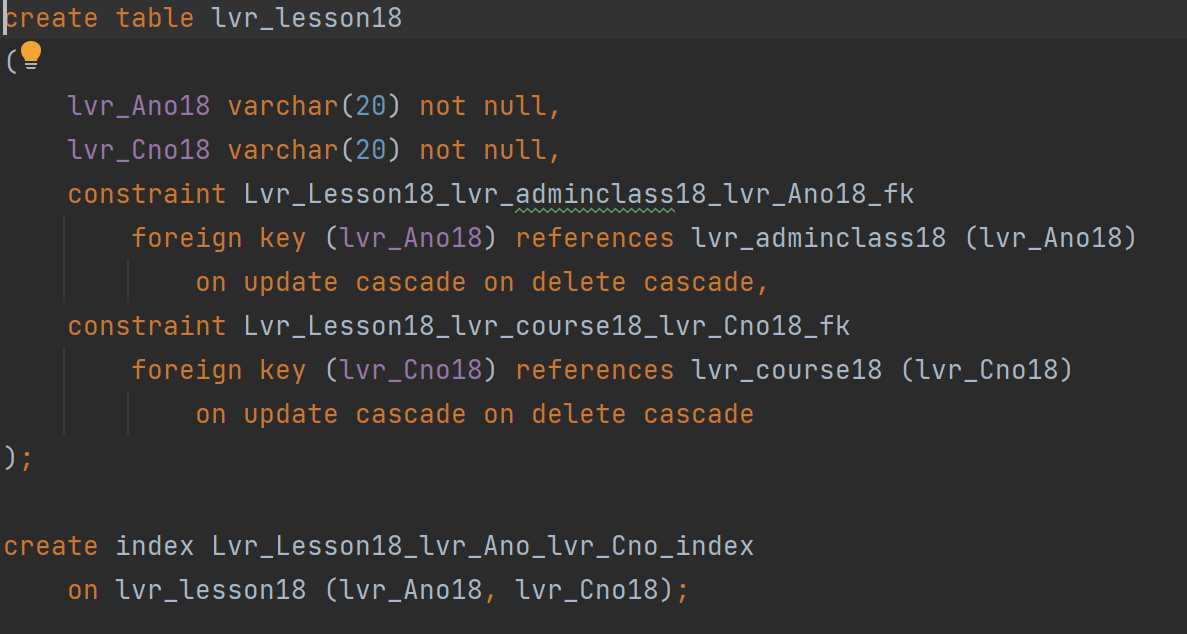


图5-9 开课表

### 5.2 视图的建立

**①**学生成绩统计视图

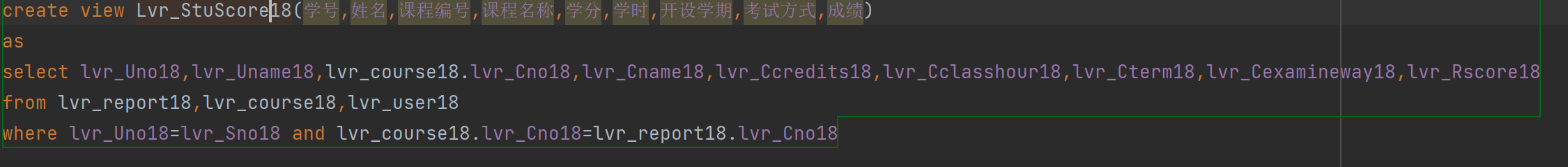
****

图5-9 学生成绩统计视图

**②**每门课平均成绩视图

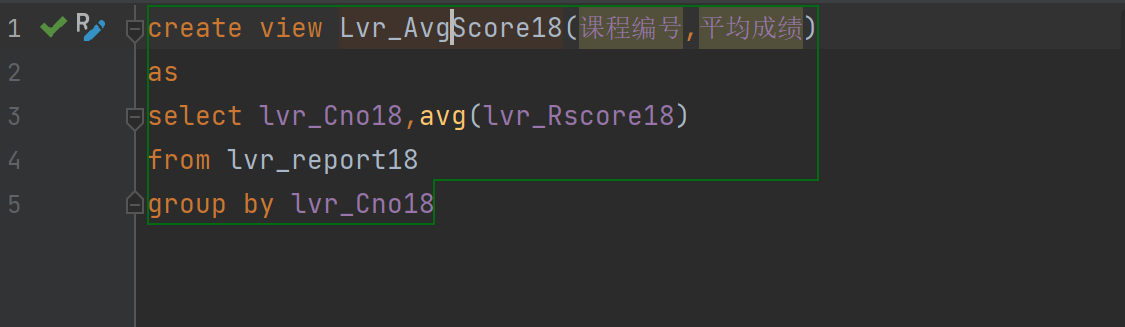


图5-10 每门课平均成绩视图

**③**学生所学课程及学分统计视图



图5-11 学生所学课程及学分统计视图

**④**教师任课情况查询视图

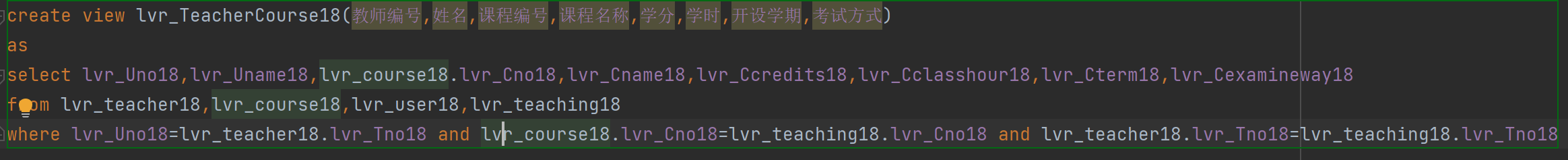
****

图5-12 教师任课情况查询视图

**⑤**班级开课查询视图



图5-13 班级开课查询视图

**⑥**地区学生数统计视图

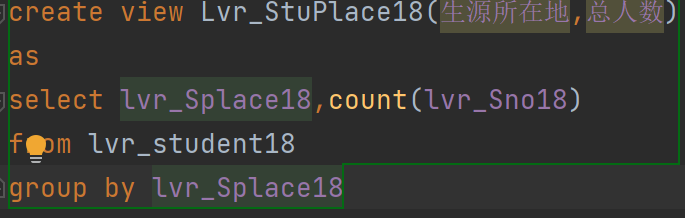


图5-14 地区学生数统计视图

**⑦**平均成绩按学年统计视图

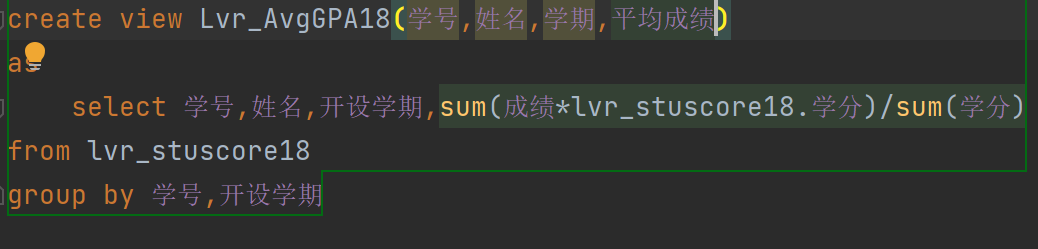
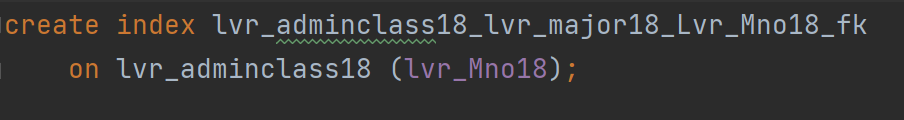
****

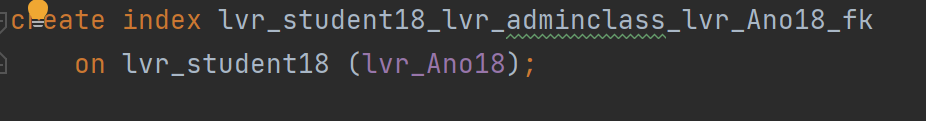
图5-15 平均成绩按学年统计视图

### 5.3 索引的建立

1、班级表的专业编号索引



2、学生表的班级编号索引



### 5.4 触发器建立

**①**当插入用户信息后将信息同步到学生表



图5-16 当插入学生信息后将信息同步到用户表

**②**当插入用户信息后将信息同步到教师表

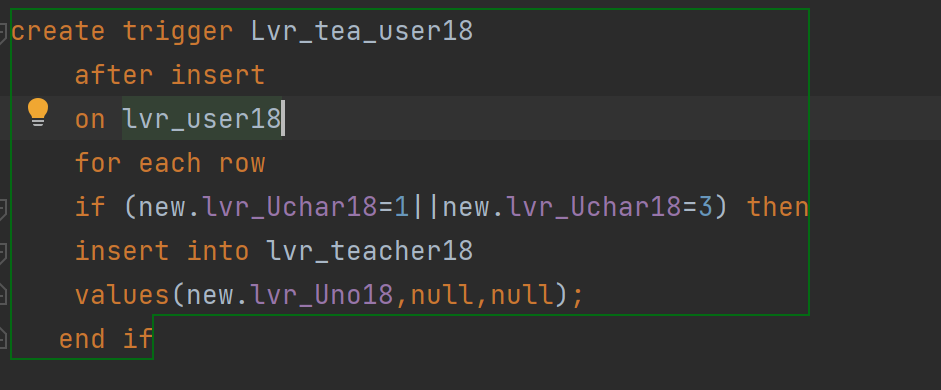
****

图5-17 当插入教师信息后将信息同步到用户表

**③**当插入成绩时自动计算已修学分

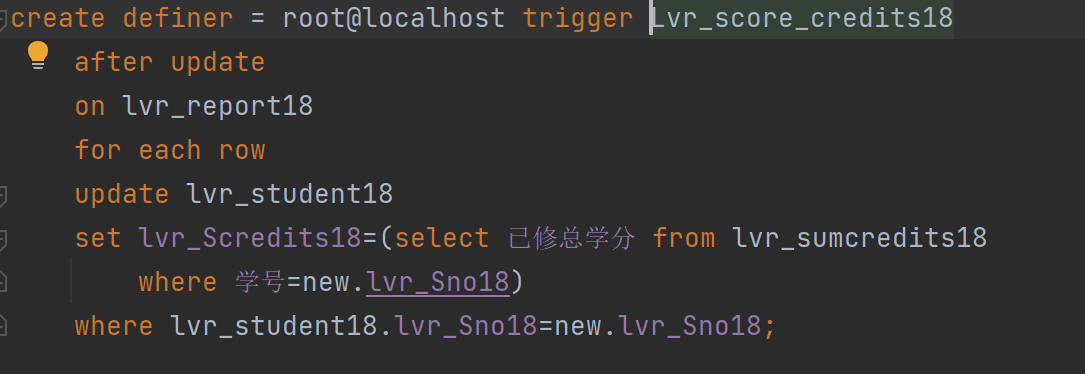


图5-18 当插入教师信息后将信息同步到用户表

### 5.5 建存储过程

**①**修改成绩的存储过程

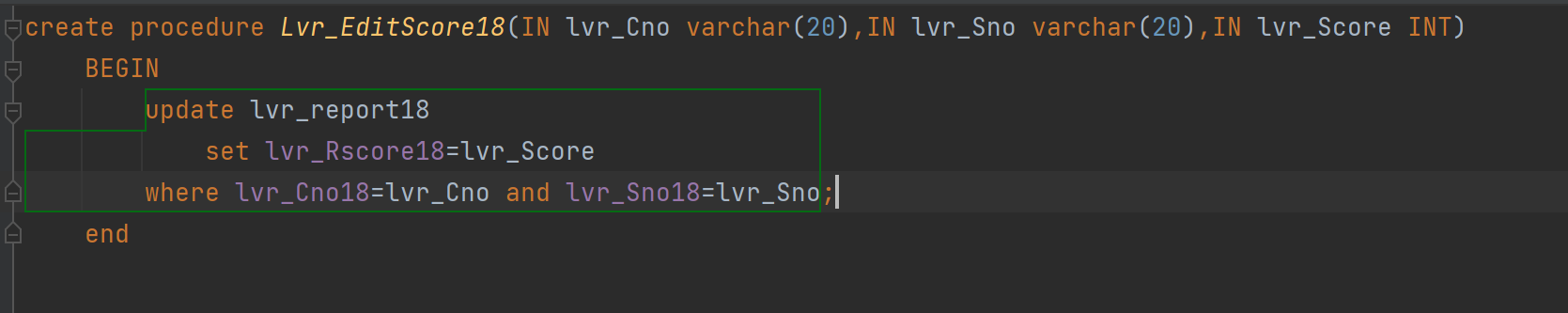
****

图5-19 修改成绩的存储过程

**②**建立修改教师表内容的存储过程

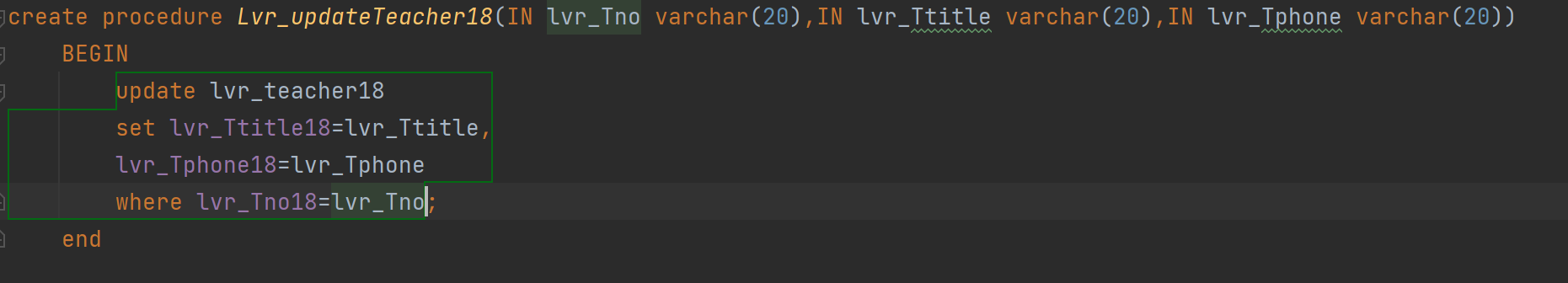
****

图5-20 建立修改教师表内容的存储过程

**③**建立修改学生表内容的存储过程

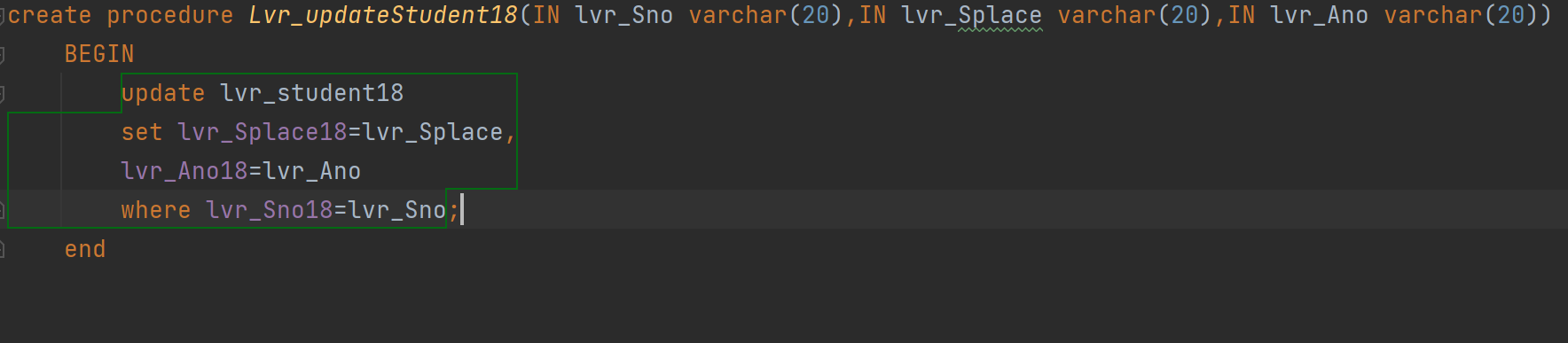
****

图5-20 建立修改学生表内容的存储过程

## 6、应用系统开发与试运行

### 6.1 开发平台和开发环境介绍

开发系统：Windows10 PC

环境：JDK1.8的Java编译环境 Tomcat10.0.6运行环境

开发软件：Intellij IDEA 2020.3.3

### 6.2 前台界面与后台数据库连接说明、代码实现

使用jsp进行前台界面的开发，使用dao和mvc设计模式对数据连接进行封装，建立数据池，使用DAO封装Data Source，同时将对数据库的操作，基本的方法也封装到各个model对应的DAO内部，

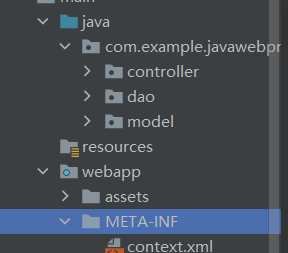
#### 代码实现文件详情

##### Context.xml文件：

<?xml version="1.0" encoding="utf-8"?>  
<Context reloadable="true">  
 <Resource  
  
 name="jdbc/lvrmis18"  
 type="javax.sql.DataSource"  
 maxTotal="99"  
 maxIdle="2"  
 driverClassName="com.mysql.cj.jdbc.Driver"  
 url="jdbc:mysql://localhost:3306/lvrmis18?useSSL=false&amp;useUnicode=true&amp;characterEncoding=utf8"  
 username="root"  
 password="manstein2000"  
 maxWaitMillis="5000"/>  
</Context>

##### Dao.java文件：

public interface Dao {  
  
 public static DataSource getDataSource(){  
 DataSource dataSource = null;  
 try {  
 Context context = new InitialContext();  
 dataSource = (DataSource)context.lookup("java:comp/env/jdbc/lvrmis18");  
  
 }catch(NamingException ne){  
 System.*out*.println(ne);  
 }  
 return dataSource;  
 }  
  
 public default Connection getConnection() throws  
  
 com.example.javawebproject.dao.DaoException {  
 DataSource dataSource = *getDataSource*();  
 Connection conn = null;  
 try{  
 conn = dataSource.getConnection();  
 }catch(SQLException sqle){  
 System.*out*.println(sqle);  
 }  
 return conn;  
 }  
}



### 6.3 系统各功能设计和运行界面截图

#### 6.3.1登录模块

选择登录页面，此页面可以选择登录身份，即选择管理员登录，学生登录还是教师登录：

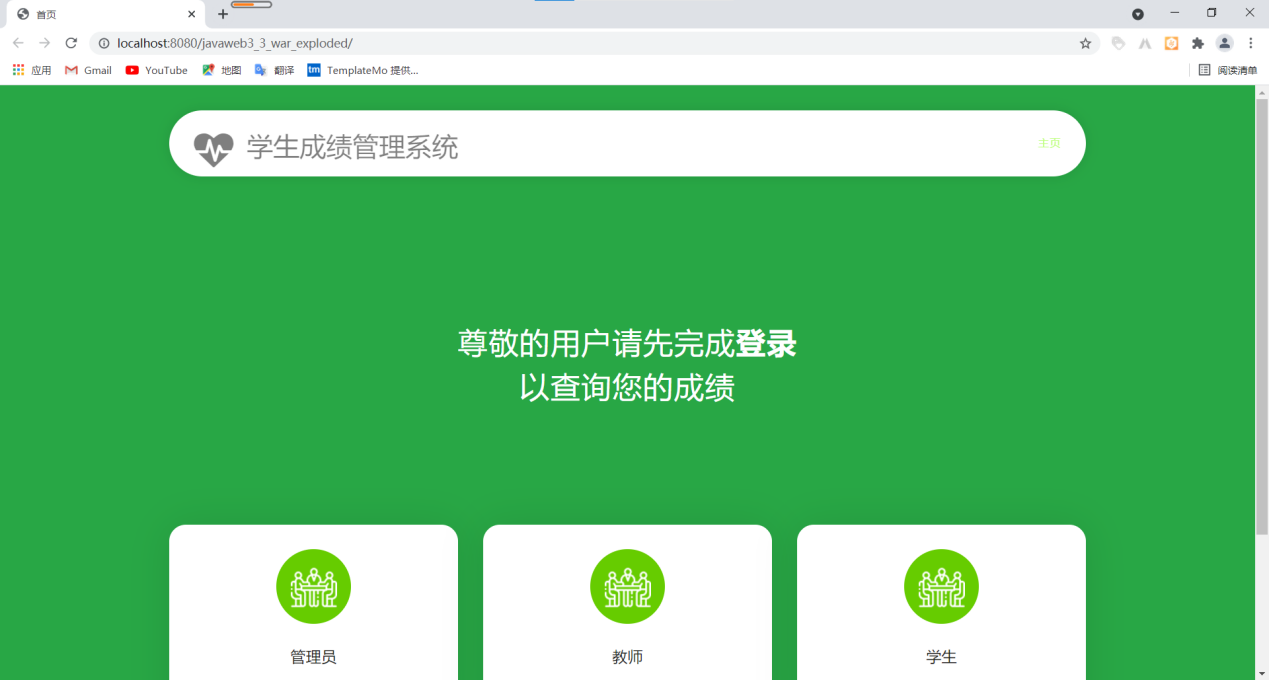


图6-1 用户登录页面

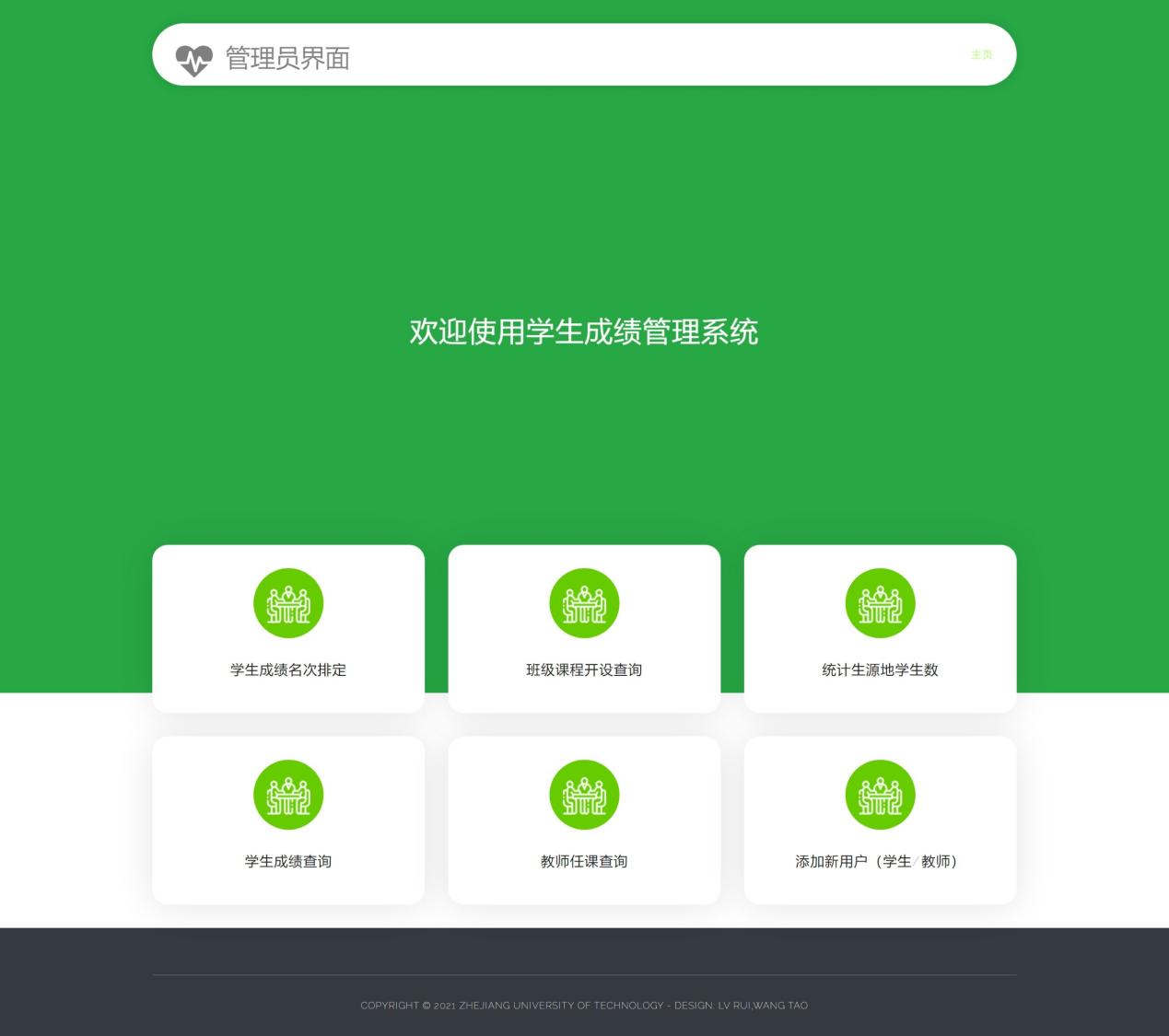
#### 6.3.2管理员模块

**管理员登录界面**



输入编号和密码进行登录，并在loginActionServlet中连接数据库中的登录表，验证用户名密码是否正确

**管理员登录主界面**



**查询学生成绩名次**

**点击 学生成绩名次排定 进入**



可进行升序降序选择和学期选择

可根据学期按照课程成绩进行学期选择，类型为下拉框，防止客户随意输入

**班级课程开设查询**

**点击 班级课程开设 查询**

**进入如下界面**



可通过班级编号搜索开课情况，不输数字默认为显示所以开课情况

**统计生源地学生数**

**点击 统计生源地学生数 进入如下页面**

#### 

可通过地区名查询

支持模糊查询

**学生成绩查询**

**点击 学生成绩查询 进入如下页面**

#### 

可通过学生编号查询

可根据学期按照课程成绩进行学期选择，类型为下拉框，防止客户随意输入

**教师任课查询**

**点击 教师任课查询 进入如下页面**

#### 

可通过教师编号查询

可根据学期按照课程成绩进行学期选择，类型为下拉框，防止客户随意输入

**添加新用户**

**点击 添加新用户 进入如下页面**

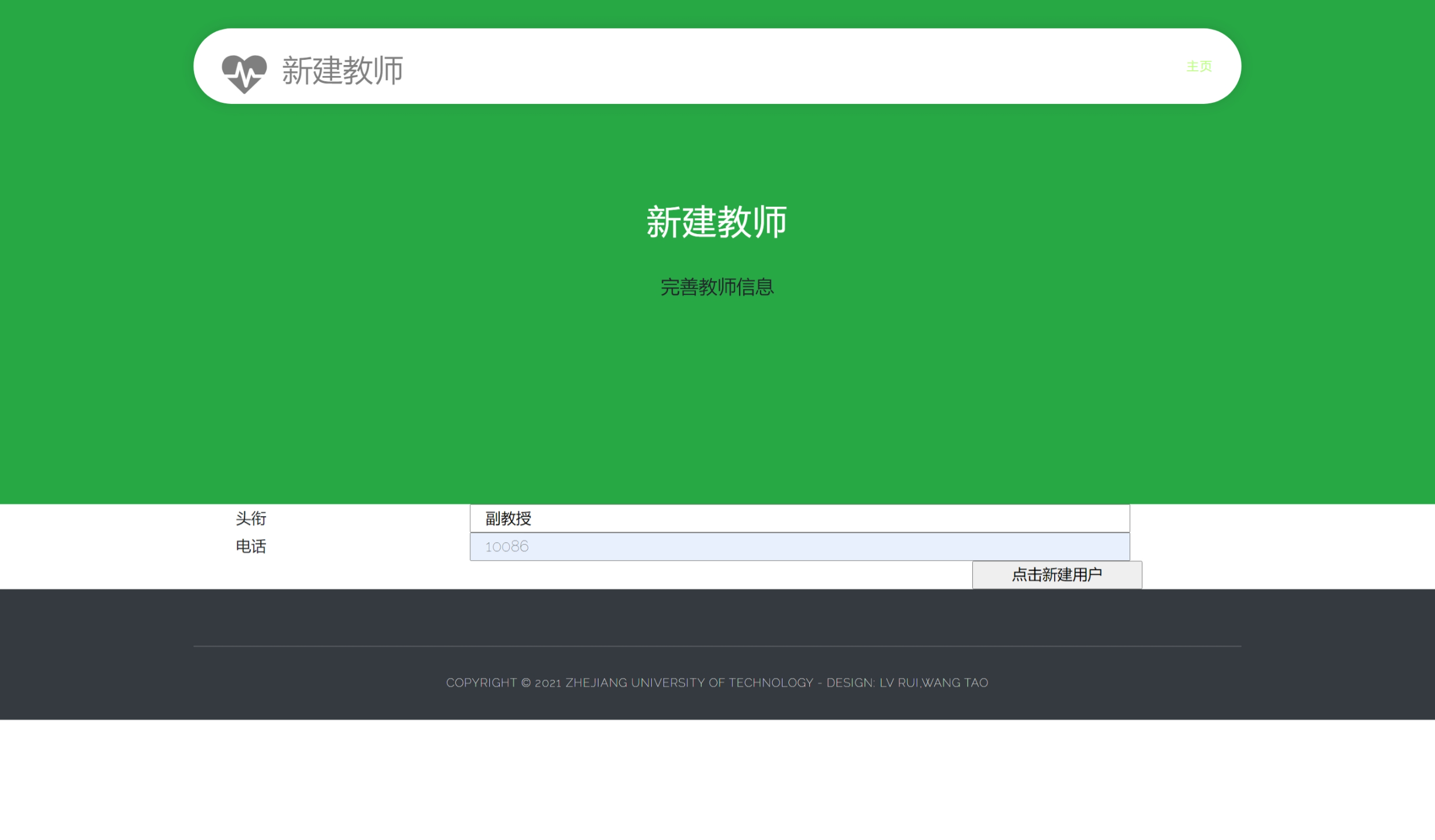
#### 

根据选择的不同角色会转跳到不同的详细信息填写界面

学生或学生管理员进入新增学生界面



教师或教师管理员进入新增学生界面



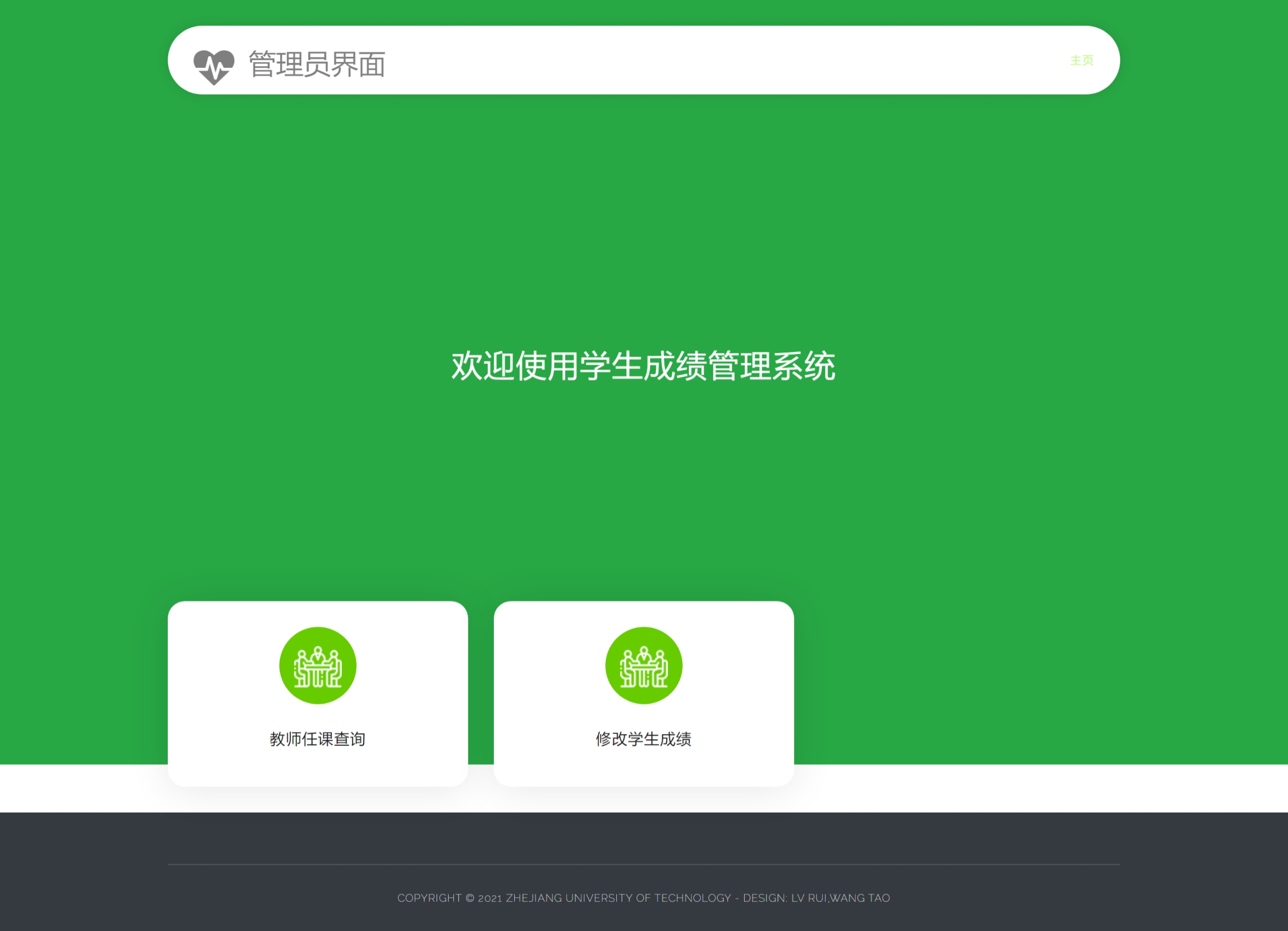
#### 6.3.2教师模块

**教师登录界面**



输入编号和密码进行登录，并在loginActionServlet中连接数据库中的登录表，验证用户名密码是否正确

**管理员登录主界面**



**查询排课情况**

**点击 教师任课查询 进入**



返回登录账号对应老师被安排教学的课程以及班级

可根据学期按照课程成绩进行学期选择，类型为下拉框，防止客户随意输入

**查询修改学生成绩**

**点击 修改学生成绩成绩 进入如下界面**

可通过学号查询



可根据查询结果更改学生成绩，同时会同步到学生成绩视图和更新该学生所修总学分

修改成功会有提示



#### 6.3.3学生模块

**学生登录界面**



输入编号和密码进行登录，并在loginActionServlet中连接数据库中的登录表，验证用户名密码是否正确

**管理员登录主界面**



**学生所学课程及学分统计**

**点击 学生所学课程及学分统计 进入如下页面**



点击主页返回登录账号

**查询课程成绩**

**点击查询课程成绩进入如下界面**



可根据学期按照课程成绩进行学期选择，类型为下拉框，防止客户随意输入

## 7、实验总结

### 7.1 遇到的问题和解决的办法

**DAO设计与视图设计**

我在设计基本表的时候，为了顾及信息的详细和全面，设计了很多基本表。

但是这对Java里选用数据，特别是查询带来了很大的麻烦，查询很多地方出现了重复并且无法做到一表一DAO的设计思想导致代码复用率低下，可读性也很差。所以我制作了大量视图。因为有很多信息的交叉，所以我不得不将一个个的表格信息抽取出来先整理到草稿纸上，再根据查询要求设计成相关视图。很多视图甚至需要反复设计、修改。幸好经过长时间的设计修改，终于完成了视图的设计，大大提示了代码的复用率和可读性。

**触发器设计**

在刚开始设计触发器的时候，对它和外键联级删除、更新的区别不是特别有概念。后来实际操作后发现，联级只能算系统自带的最基础的触发，而触发器的完整性约束功能更强，能加强几张表之间的联级修改。

### 7.2 系统设计的不足

**输入问题**

对于输入的数据没有做全面的约束和限制，也没有支持所有输入选择都通过下拉框实现。

**增删改不完全**

虽然实现了对老师学生和用户的增加修改但限于时间没有能完成对所有表都进行增删改。

### 7.3 进一步改进思路和体会

1. 在进行数据设计的时候，只考虑了几个最简单的数据输入，可能只能用于课程设计，并不能用于实际。

2. 可供用户选择的筛选条件还不够多，应增加班级专业下拉框来查询成绩学分信息等

3. 界面的空间布局不是完全统一，美化有提高空间。

4. 增加更多的提示，提高用户体验。

**体会**

需求、设计E-R图，确定关系模式，建立表格以及优化数据库，在这个过程中，我们又重新翻开书本，回忆起当初课上所教过的内容，这是一个学习\_再学习的过程。接下来就是进行索引、触发器、存储过程的设计，在结合我们建立表的属性特征，以及我们所要进行的增删查改操作的频率，设置相应的索引，建立触发器来对数据进行约束，建立存储过程来进行相应的设计。这部分我们小组集思广益，经过大家不断推敲，建立起了属于我们的数据模型。

从系统的分析，到整体的设计，再到数据库的建立，以及java前端页面的编写。数据库驱动的加载。可以说整个的项目开发都是由一个人完成。这是在本学期学习了java课程和数据库课程之后的一次很宝贵的实战经验总结。之前对两门课程的学习只是简单的基础学习，没有设计到整个系统的开发。但是在这次的实践中对于整个系统开发有了更深的认识。从专业知识学习的角度说，首先我学会了作为一个系统的整个数据库的建立和完善。学会将SQL语句嵌入java中，对数据库进行操作。也学会了如何利用java的Mysql驱动连接数据库，这为之后的java的其他开发奠定了坚实的基础。其次数据库可课程设计更是提升了java的实战能力。在本次课设中，通过主函数中对其他各个累的调用和 其他设计，更深层次的了解或者说是掌握了java的面向对象这一概念。此次开发总共涉及了自己编写的大大小小的十几个java文件，但正是由于面向对象的思想，对于java的操作却没有想象的那么复杂。系统的维护和改动也只需要在相应的子文件里面做改动。本次实战中具有突出的是在java的主界面的编写中，为了对于各个子文件（子页面）的操作更加的简便，也为了主菜单格式的统一，我使用了树目录结构的方式，使得主界面的操作灵活方便。也使得各个操作在主界面的显示样式更加的统一方便好看。总体来说本次的数据库课程设计从整体上独立完成了一个系统的开发。为之后多项专业课的学习打下了坚实的基础，也是一个检验自己java与数据库知识掌握程度的机会，在此次课程设计中收获颇丰，再次有类似的学习。