软件系统开发综合实践项目说明书

教学管理系统

## 小组分工

|  |  |  |
| --- | --- | --- |
| 小组成员 | 分工 | 占比 |
| 55181029李康 | 数据库设计，管理员/教师/学生模块实现与调试，报告撰写 | 55% |
| 55181030秦培充 | 网页样式设计 | 15% |
| 55181031韦子龙 | 登录模块实现 | 15% |
| 55181024王嘉冬 | 视频录制及数据库数据源 | 15% |

## 需求说明

随着学校规模的扩大，越来越多的老师，学生乃至课程都加大了学校的管理难度，纸面上的管理几乎成为了不可能的事情，因此需要一个管理系统来协助管理。

## 设计思路

### 总体思路

我们设计了3个角色来实现教学管理系统。

1. 学生。学生可以在教学管理系统中选择课程，退出课程，查看自己的最新成绩，亦或是更改密码。但是他无法更改自己的其他信息诸如姓名，学院等。
2. 教师。教师可以在教学管理系统中开设课程，结束课程（决定学生通过与否），亦或是更改密码。我们认为教师不能随便删除课程，没有将删除课程的权限赋予教师。
3. 管理员。管理员具有对任何用户的增删改查权限，任意课程的删除权限，以及学院的开设权限。管理员还可以帮学生退课，但是不能帮老师结课，只能删除课程。

### 详细思路

管理员

我们将管理员功能分为了3个模块，用户管理，课程管理和学院管理。

用户管理：

对用户管理，即用户的增减，用户信息的更改，实际上也分为了对管理员管理和非管理员管理。非管理员管理即教师和学生，它们除了通用的编号和密码外，还拥有其他属性诸如姓名，性别，这些需要用户输入。而管理员只拥有编号和密码。

参考现实中的情况，为了实现快速的批量用户添加，对于非管理员用户，我们令学号/工号自动生成，还设置了默认密码，不需要在添加时手动输入，当然后续可以更改（不管是管理员还是学生本人）。对于管理员用户，可以自己设置账号（即编号），并且在创建时就可以输入密码。

课程管理：

课程管理包括对课程信息的修改和课程的删除。鉴于管理员身份的特殊性使它更像一个工具，因此管理员不应能开设课程。同时我们认为课程的属性对于学生来讲是一件很重要的事，一旦发布不应该能随意修改甚至于删除，因此我们只将课程修改和删除的权限赋予了管理员。

管理员可以在此界面输入课程名或教师名关键词以搜索课程。

学院管理：

教师

教师的功能可以分为开设课程，（自己的）课程管理，查看和修改个人信息。

开设课程：

教师可以在此界面输入课程信息以开设课程，但是课程一旦开设信息即不可修改，仅管理员拥有对于课程的修改和删除权限，原因参照上文。

课程管理：

教师在此界面可以浏览自己课程的全部信息，也可以进入课程查看该门课程的全部学生并逐一选择是否通过课程，当教师同意某个学生通过时，该学生的总学分会自动加上该门课程的分数。

查看和修改个人信息：

个人信息包括工号，姓名，性别，所属学院，职位和密码。参考现实的教学管理系统，为了信息的严谨和严密，只有密码可以由教师本人修改，只有管理员拥有其余所有信息（包括密码）的修改权限。

学生

学生的功能包括选课，（所选的）课程管理，查看和修改个人信息和查看最新成绩。

选课：

学生在此页面可以浏览和搜索所有的课程信息（不包括自己已选/已通过的课程）并选课。

课程管理：

学生在此页面可以浏览自己所选课程的信息以及自己通过与否并选择退出课程（已经通过的课程无法退出）。

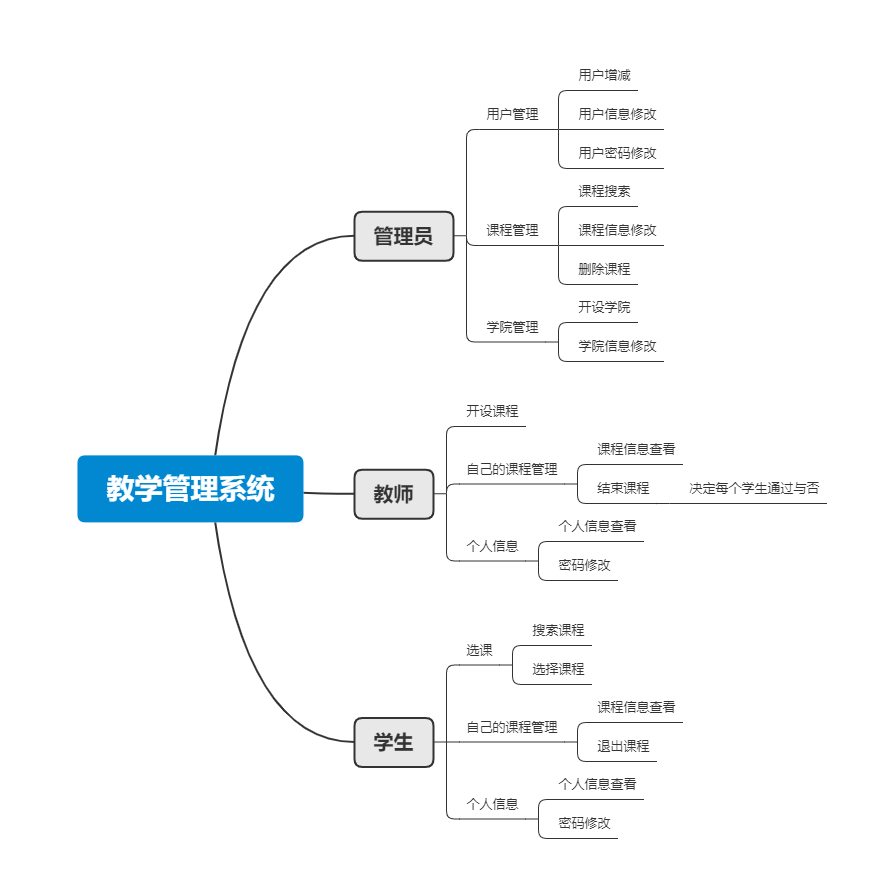
查看和修改个人信息：

个人信息包括学号，姓名，性别，所属学院，总学分和密码。只有密码可以由学生本人修改，只有管理员拥有其余所有信息（包括密码）的修改权限，原因同教师。

查看最新成绩：

学生可以在此页面查看自己所选的最新成绩（通过与否）和成绩发布时间。

### 结构

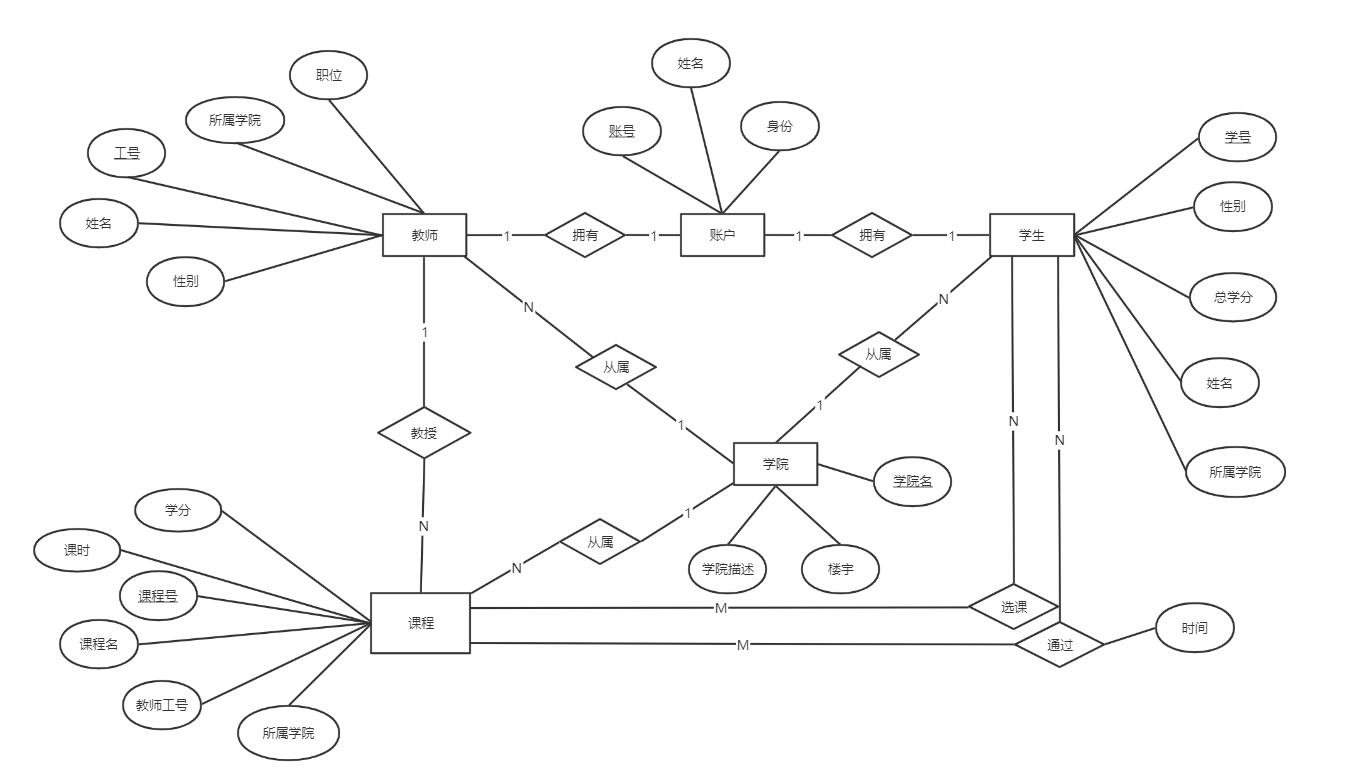


## 设计

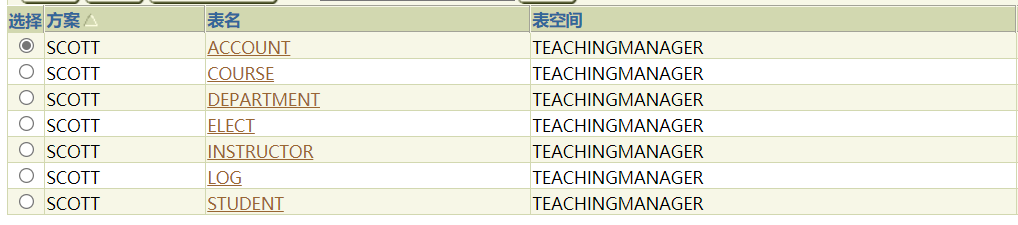
### 数据库设计

本实验使用oracle 11g数据库。

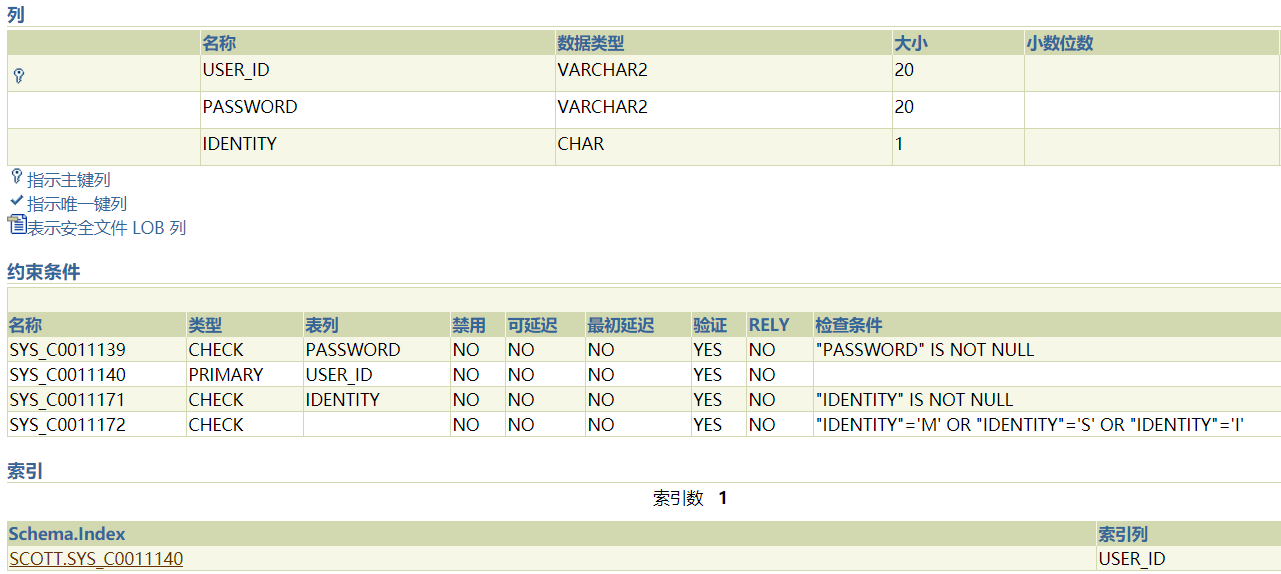
##### ER图



##### 数据库表、索引、约束、大对象



账户表

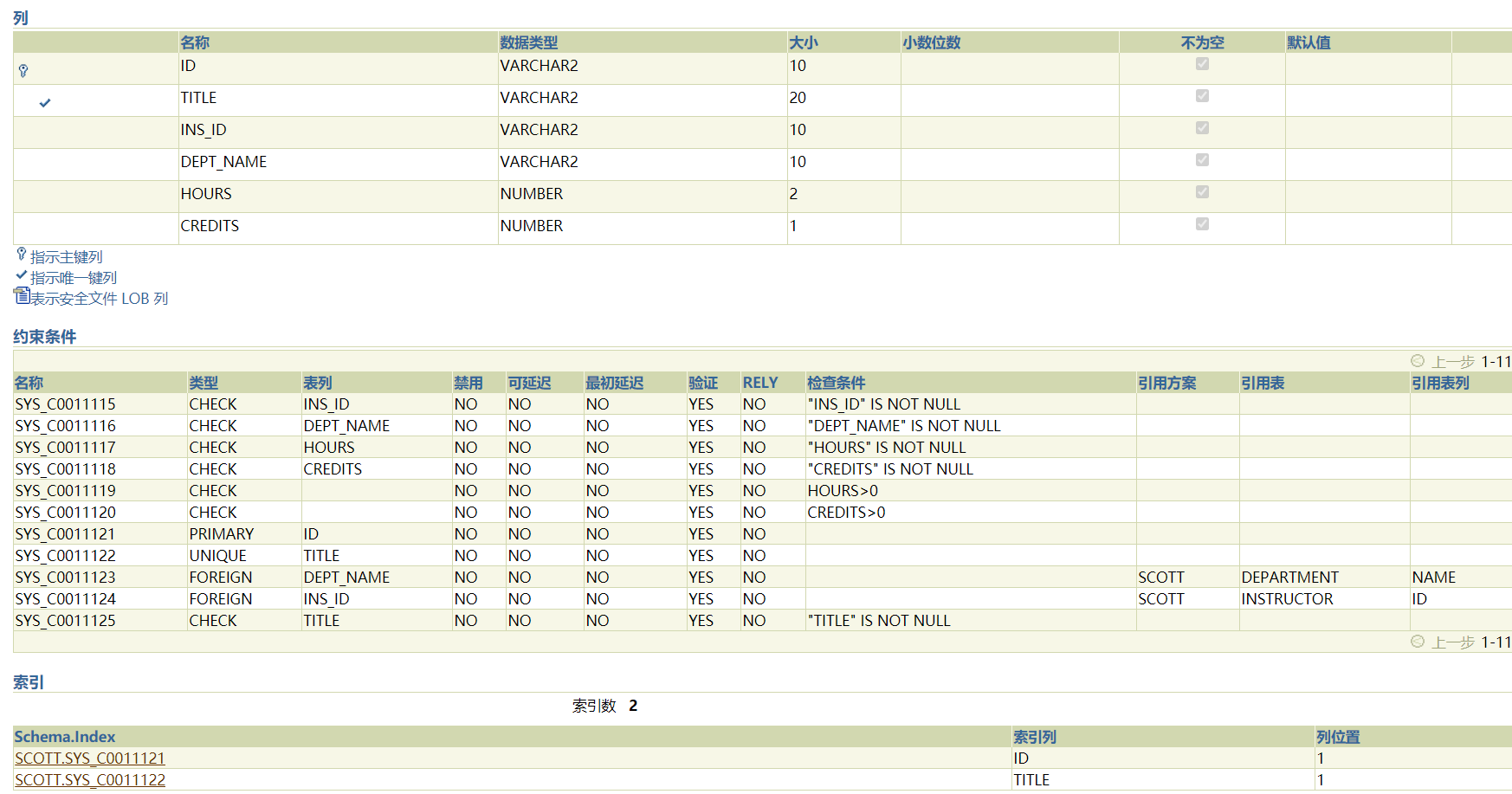


USER\_ID 账号/编号 主键（附带索引）

PASSWORD 密码 NOT NULL

IDENTITY 身份 限定‘M’,’S’,’I’，分别表示管理员，学生，教师

课程表



ID 课程号 主键（附带索引）

TITLE 课程名 UNIQUE（附带索引）

INS\_ID 教师工号 外键（引用INSTRUCTOR.ID）

DEPT\_NAME 所属学院 外键（引用DEPARTMENT.NAME）

HOURS 学时 >0 NOT NULL

CREDITS 学分 >0 NOT NULL

主键附带序列配合触发器实现主键自增自动分配

学院表

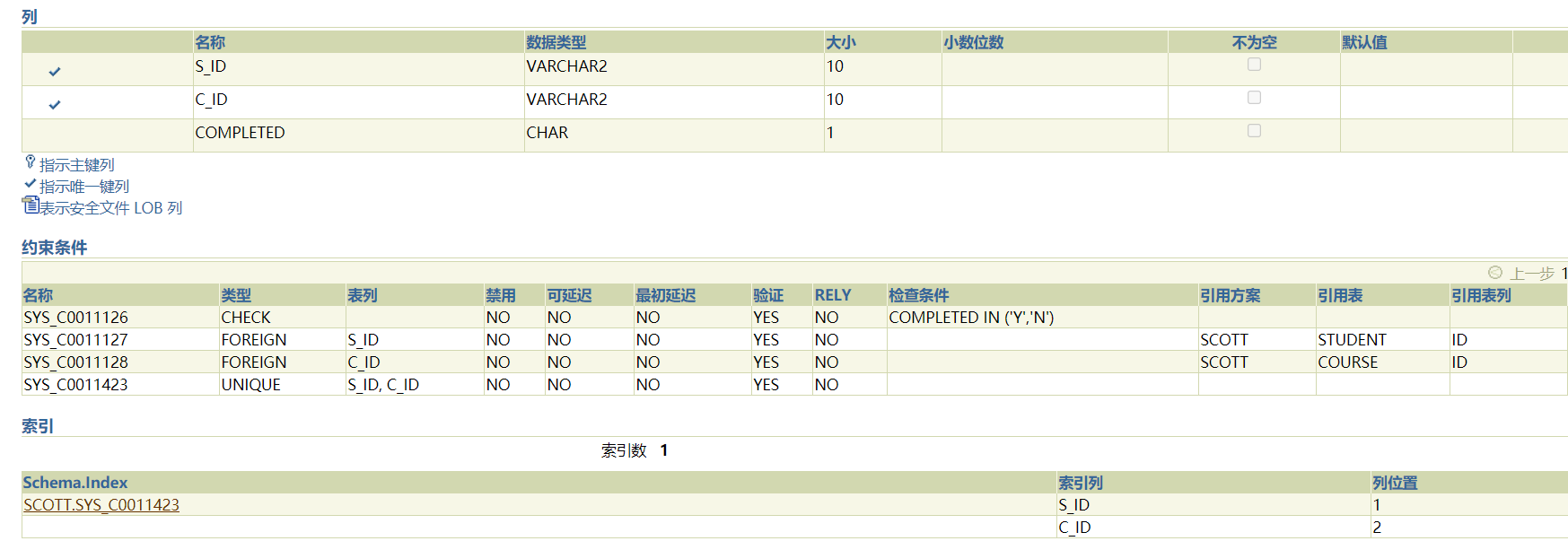


NAME 学院名 主键（附带索引）

BUILDING 楼宇 NOT NULL

DESCRIPE 学院描述 CLOB大对象类型

选课关系表



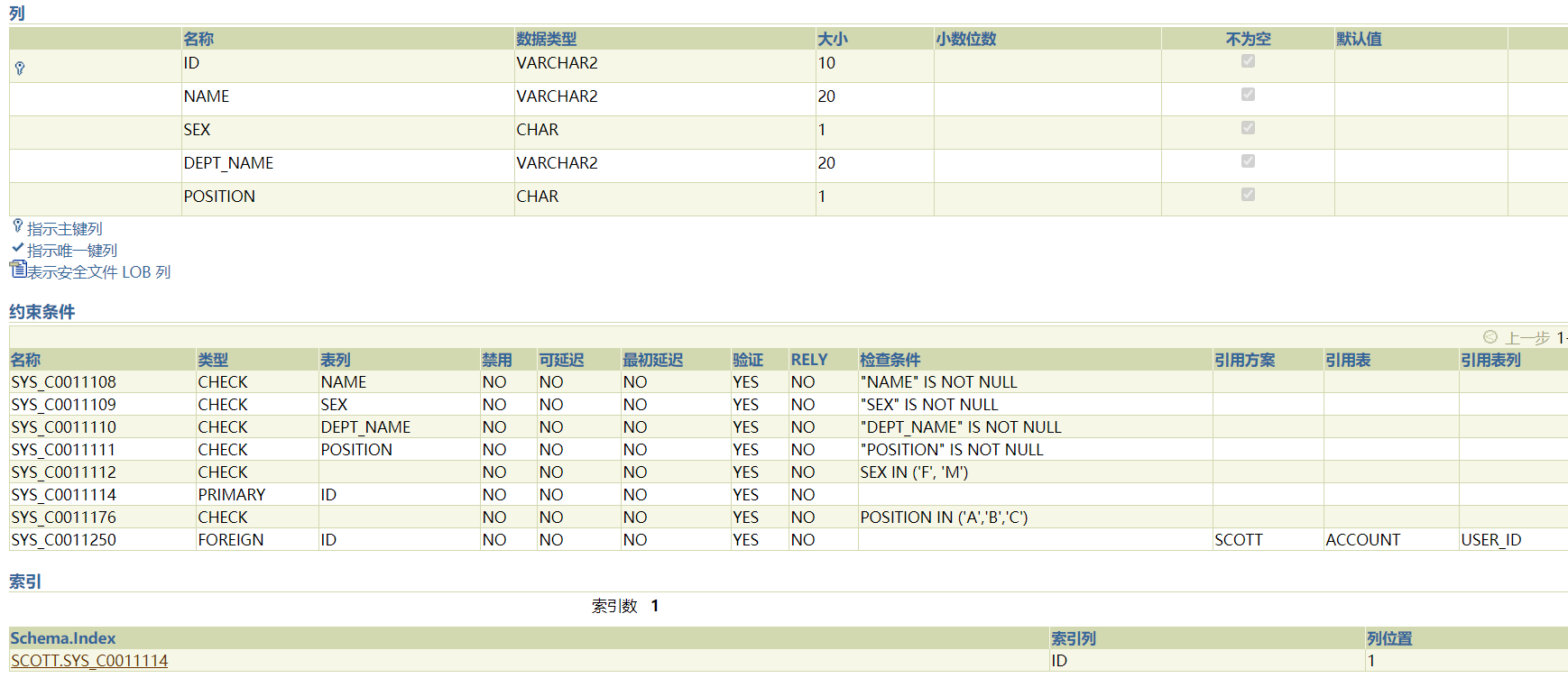
S\_ID 学生编号 外键（引用STUDENT.ID）

C\_ID 课程号 外键（引用COURSE.ID）

COMPLETED 课程通过与否 限定‘Y’,’N’，分别代表通过/未通过

S\_ID与C\_ID构成一个UNIQUE约束，并且具有共同索引

教师表



ID 教师工号 主键（附带索引），外键（引用ACCOUNT.USER\_ID）

NAME 姓名 NOT NULL

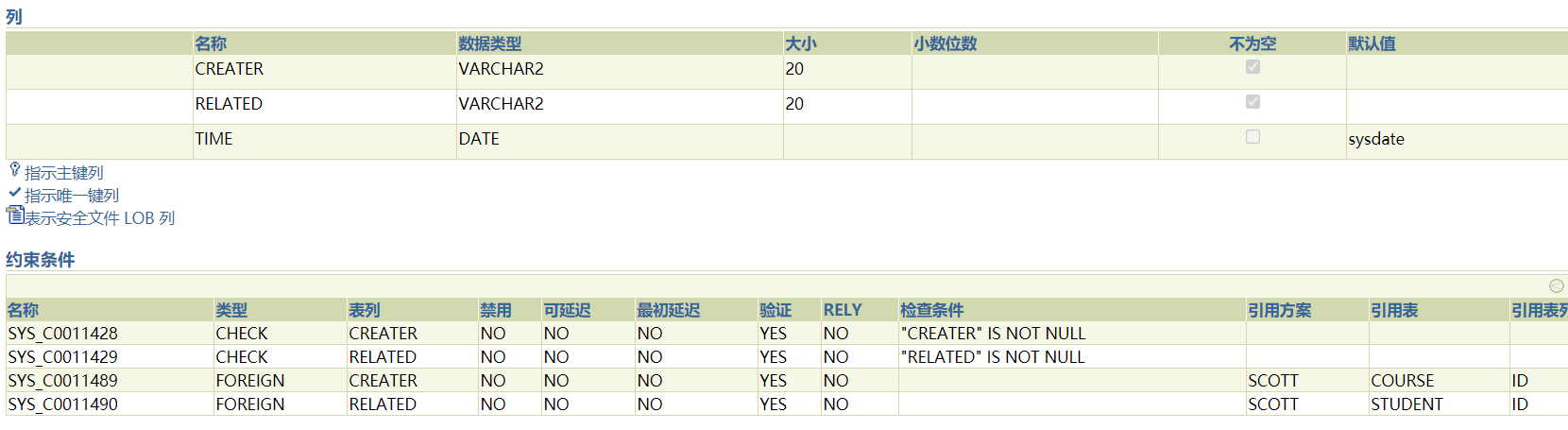
SEX 性别 限定‘F’,’M’，分别代表女性/男性

DEPT\_NAME 所属学院 外键（引用DEPARTMENT.NAME）

POSITION 职位 限定‘A’,’B’,’C’，分别代表教授，副教授，讲师

主键附带序列配合触发器实现主键自增自动分配

课程通过关系表

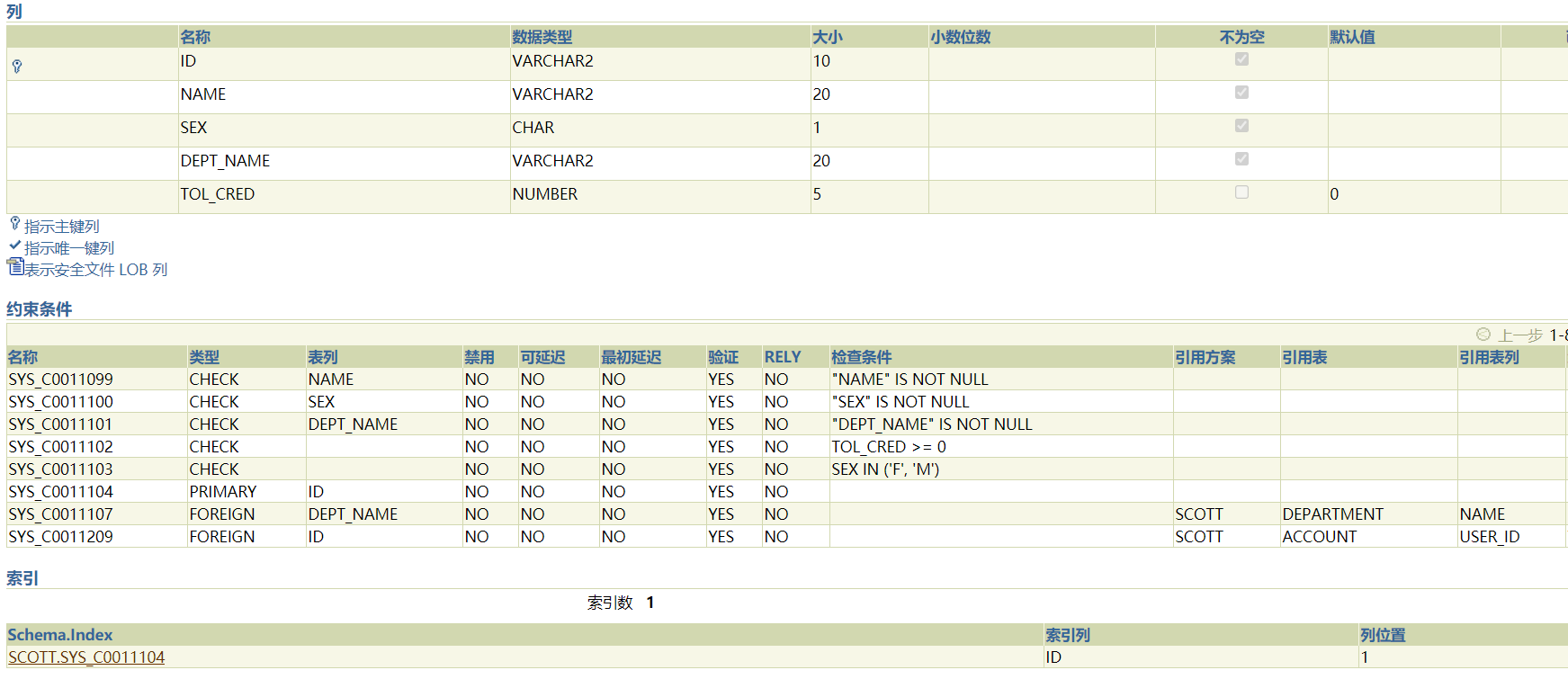


CREATER 课程号 NOT NULL

RELATED 学生学号 NOT NULL

TIME 成绩产生时间 DATE类型，默认值为系统时间

学生表



ID 学号 主键（附带索引）外键（引用ACCOUNT.USER\_ID）

NAME 姓名 NOT NULL

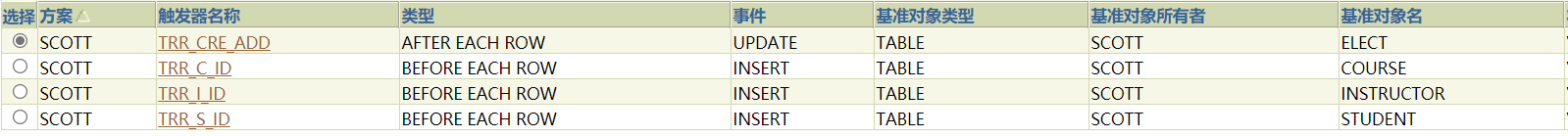
SEX 性别 限定’F’,’M’，分别代表女性/男性

DEPT\_NAME 所属学院名 外键（引用DEPARTMENT.NAME）

TOL\_CRED 总学分 >=0，默认值为0

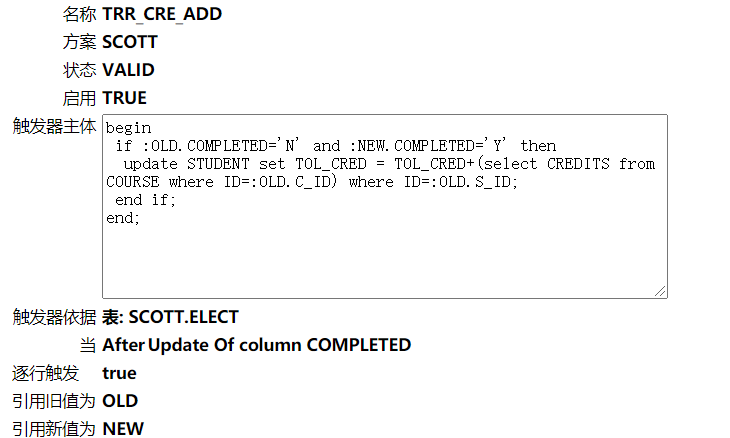
主键附带序列配合触发器实现主键自增自动分配

##### 触发器



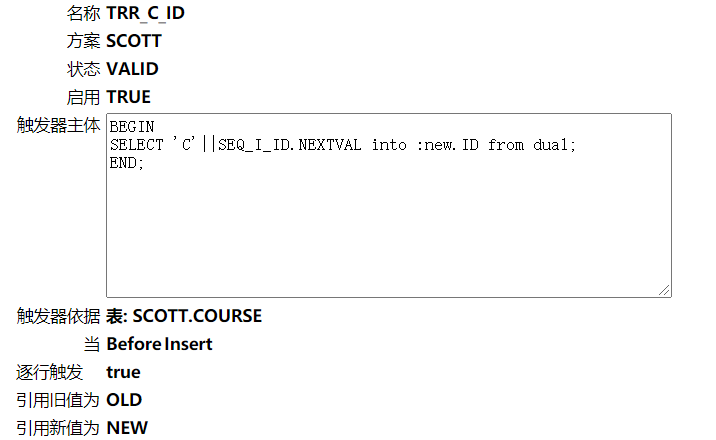
TRR\_CRE\_ADD

功能：当ELECT表学生选课信息中的COMPLETED列的值由‘N’变成‘Y’时，该条选课信息对应的学生的学分会自动加上该条选课信息对应的学分

详细：

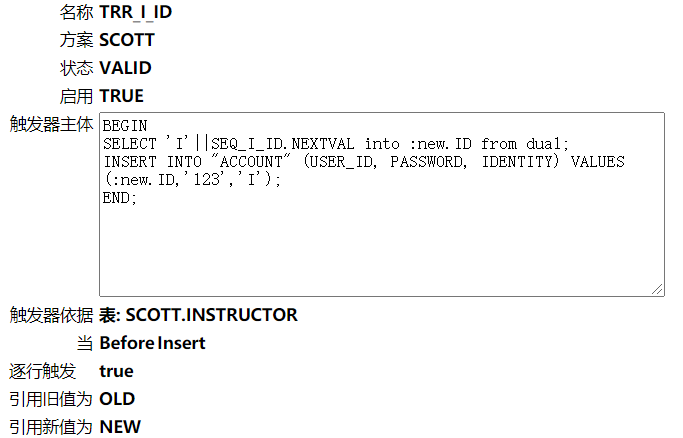
TRR\_C\_ID

功能：配合序列实现课程表的ID（编号）主键自增自动分配

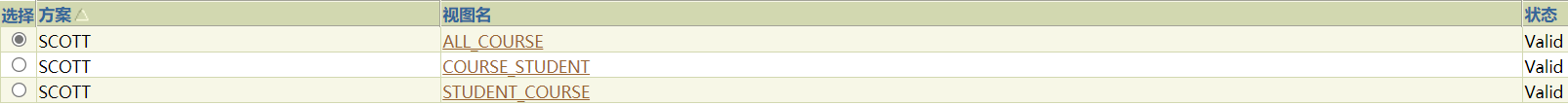
详细（以TRR\_C\_ID）为例：

TRR\_I\_ID/TRR\_S\_ID

功能：配合序列实现教师/学生表的ID（编号）主键自增和自动分配，同时在向教师/学生表中插入一行时，也会向账户表中插入默认密码为123的对应的一行。

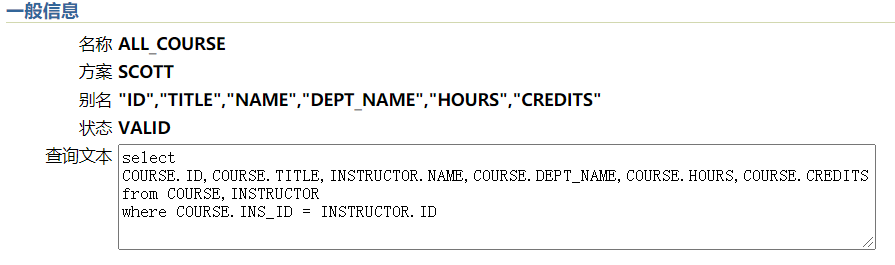
详细：

##### 视图



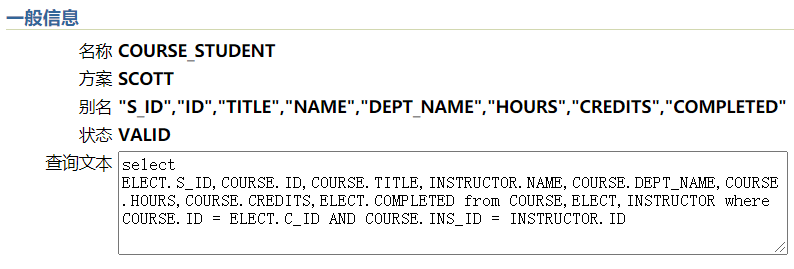
ALL\_COURSE

功能：连接课程表，教师表输出包含教师姓名在内的全部课程信息

详细：

COURSE\_STUDENT

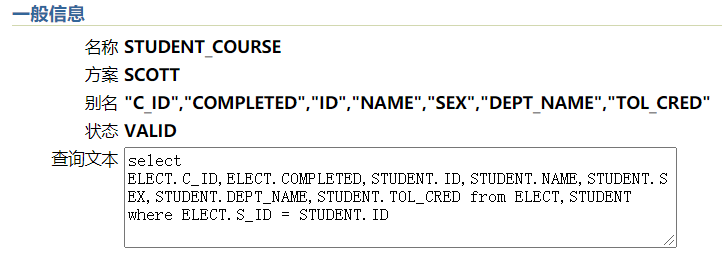
功能：连接选课关系表，课程表输出包含课程通过与否在内的某个学生的选课信息

详细：

STUDENT\_COURSE

功能：连接选课关系表，学生表输出选了某门课的全部学生的学生信息

详细：



##### 序列



SEQ\_C\_ID/SEQ\_I\_ID/SEQ\_S\_ID

功能：配合触发器实现课程，教师和学生表的主键自增和自动分配

详细（以SEQ\_C\_ID为例）：

### 程序设计

本实验为Dynamic Web 项目，所用jdk版本为1.7.0\_45

采用技术包括JSP,Servlet,JDBC,HTML,CSS,JavaScript

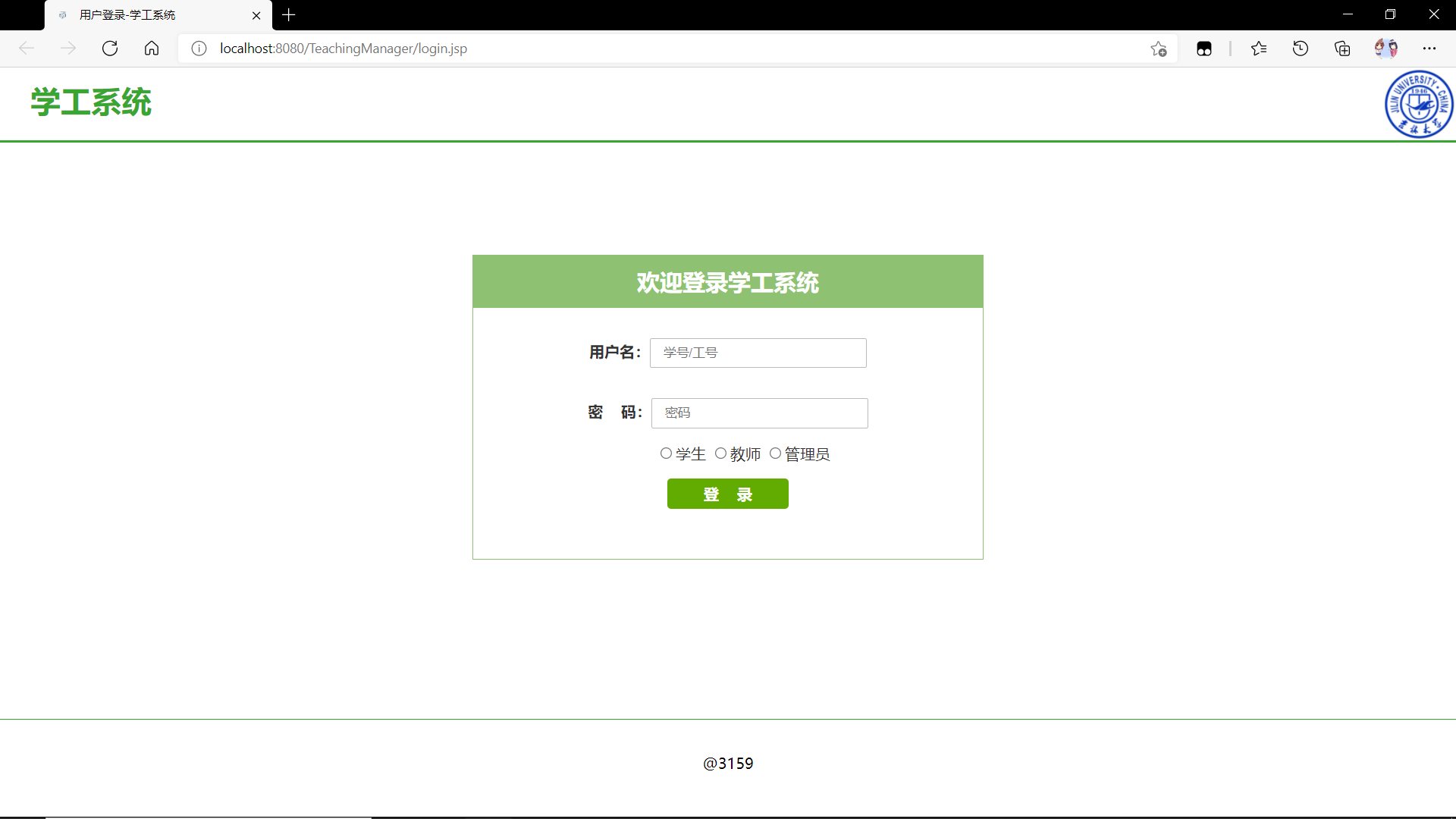
IDE为eclipse 2021-03(4.19.0)

本实验采用多个Javabean类实现java对象与数据表的一对一映射，同时方便了在前端的操作（只需对javabean对象操作，无需频繁地与数据库交互）

网页通常逻辑为用户填写表单传至servlet，servlet调用jdbc函数对数据库进行修改，最终携带执行信息返回原网页。在网页中为方便查看大量使用了列表，实现逻辑为查询数据库构成数据表对应Javabean类型的List集合，返回前端后使用jsp循环<td>标签读取List集合每项值形成表格。列表中往往每一行都包含了操作按钮，实现逻辑为提交表单时传递该按钮的值至servlet，servlet识别该按钮的值判断接下来对数据库进行何种操作。

在表单中填写的数据必须满足数据库约束，否则servlet跳转回原界面时将携带错误信息并以调用js函数以弹窗显示，同理成功执行时也会弹出成功提示，下面不再赘述

##### 登录页面



在此页面输入账号，密码以及选择身份即可登录，该页面验证的是ACCOUNT表（账户表）的信息。验证功能由service.login包下的loginServlet实现，比对账号，密码和身份与表中信息匹配则跳转到对应模块，否则设置request级错误信息变量并跳转回登陆页面，登录页面读取错误信息并显示。

考虑了三种错误信息：用户名/密码为空，密码错误，权限不足（身份选择错误），参见演示视频。

若选择用户/教师身份登录，登录成功后会将ID传至session级变量以供后续网页及子网页使用。而管理员并无私有属性，账户和密码仅是一个验证的工具，因此不必设置变量。

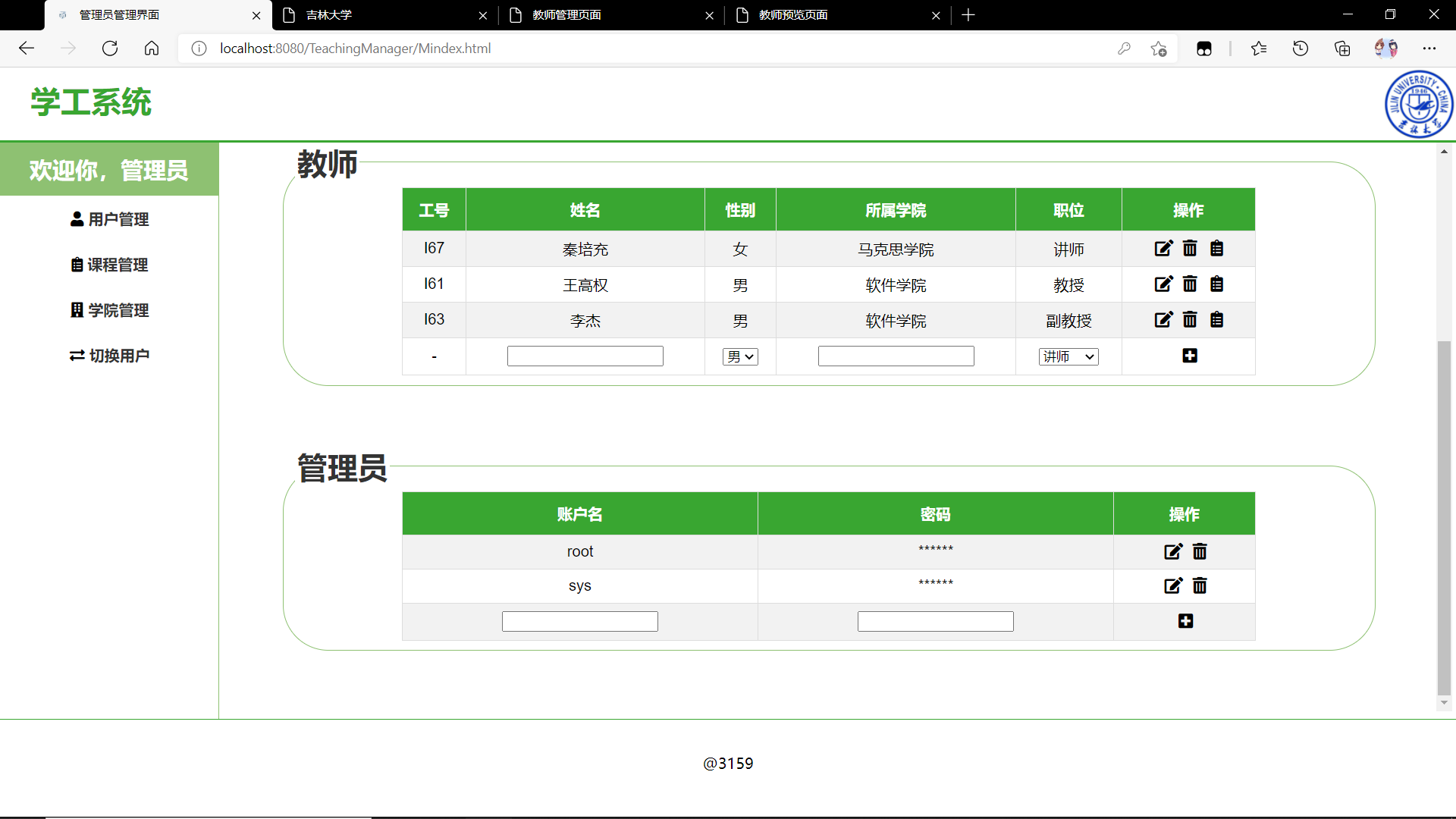
每次进入登陆页面时，调用session.invalidate()以清空session级变量。

##### 管理员模块

以管理员账户登录管理员页面，网页采用div+css+iframe布局，左侧为导航栏，右侧为iframe框架，点击左侧导航栏可切换iframe内的网页。



用户管理



如图，在右侧网页以三个表格分别列出了所有学生，教师，管理员的全部信息，并且对应不同的操作。学生/教师可以添加/删除/编辑课程信息/编辑个人信息，管理员可以添加/删除/编辑个人信息。

在此界面下的表单输入将被提交至包service.managerservlet.user下的servlet进行处理（按照管理员/教师/学生分为3个servlet）。

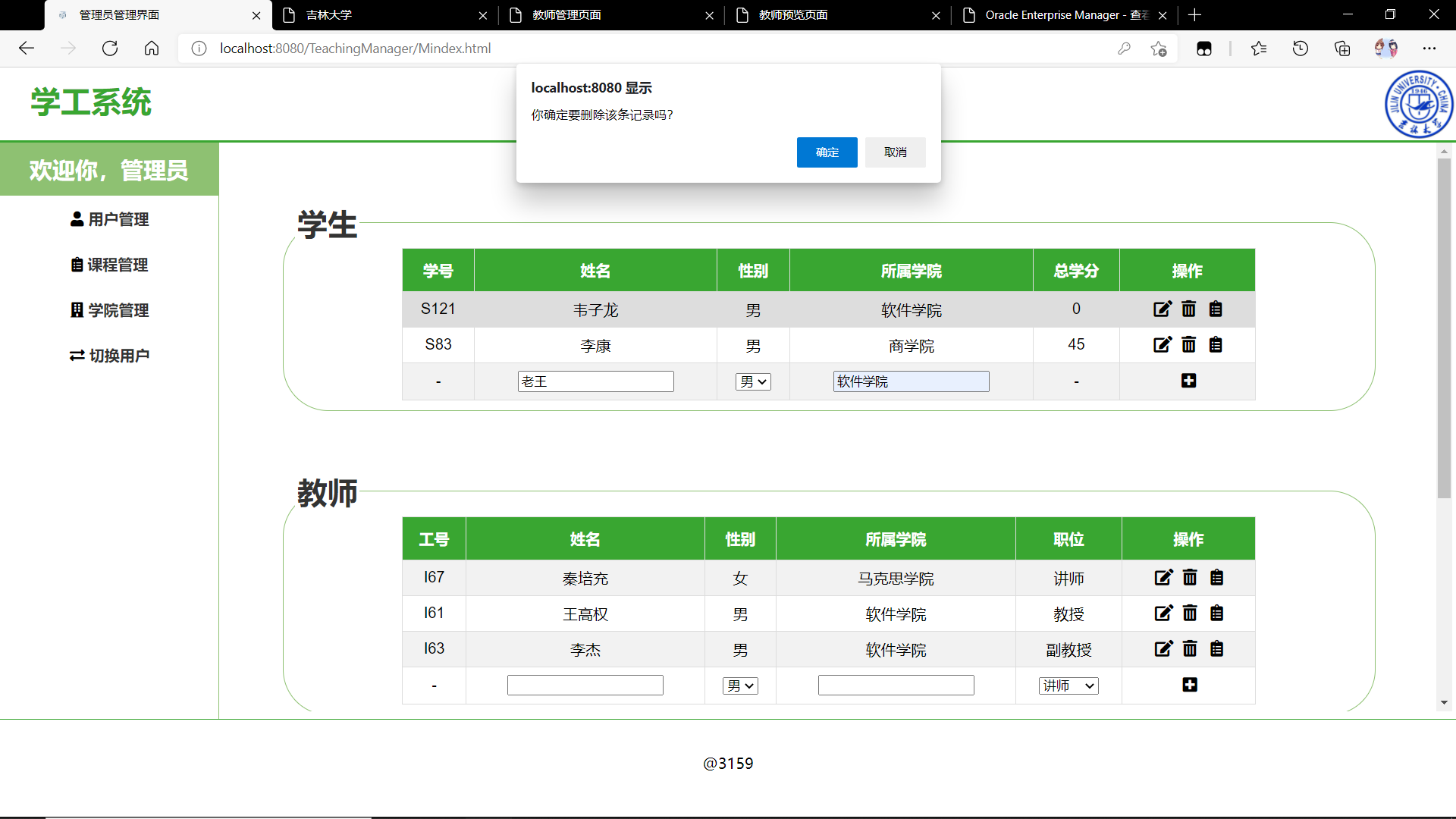
点击表格中一行的编辑按钮，按钮的onclick事件调用js函数将弹出一个表单，在此表单中输入信息可以编辑对应用户的信息，点击确认后表单将被提交。Servlet中提取不为空的表项的值调用事先定义jdbc函数更新数据表，即在该表单中，不必输入全部数据，对于不必要修改的数据可以不输入，以下教师和课程信息编辑同理，servlet执行完毕后将返回原界面。



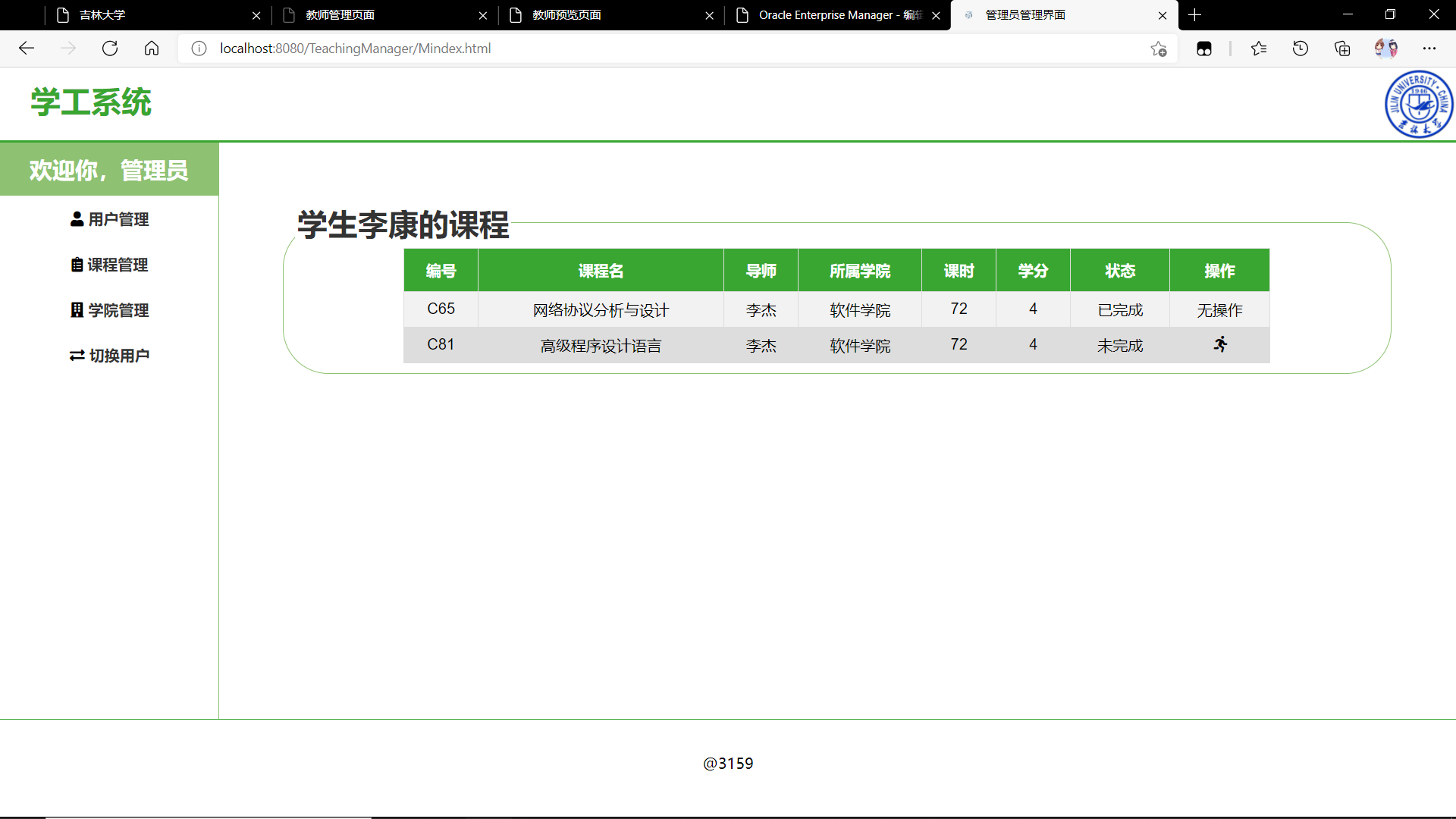
考虑到管理员身份的功能性远大于用于浏览的作用，我们没有将添加功能额外放出，而是直接在列表最末添加一行可以输入的空行，随后直接点击该行操作的添加按钮即可快速的添加。添加功能同样是以表单形式提交至servlet处理。如果是教师/学生，数据库操作为在INSTRUCTOR/STUDENT表中插入一行，随后触发器会在ACCOUNT表中同样插入对应的一行，初始默认密码为123。如果是管理员，数据库操作为直接在ACCOUNT中插入一行。



点击表格中一行的删除按钮，将弹出提示框再次确认删除操作，选择是后同样提交一个表单，servlet识别提交按钮的value值为删除后调用jdbc删除数据表中对应行后返回原界面。



点击教师/学生的编辑课程按钮，可以快速的进入教师的所教课程列表/学生的所选课程列表进行编辑，进行帮助学生退课与编辑/删除教师课程的操作。教师还可在此页面搜索自己的课程，该列表的操作会因课程状态而产生差异，学生已经修完的课程无法退课。





具体操作参见演示视频。

课程管理

课程管理界面是一个具有搜索功能的界面，若不输入关键词将显示所有课程（首次进入即如此）。搜索框支持多个关键字以空格隔开，支持课程名与教师名搜索。在输入关键词点击提交后，servlet会将字符串以split()分割为字符串数组，并以此作为参数传至jdbc函数。Jdbc函数以此字符串数组拼接sql语句进行搜索。

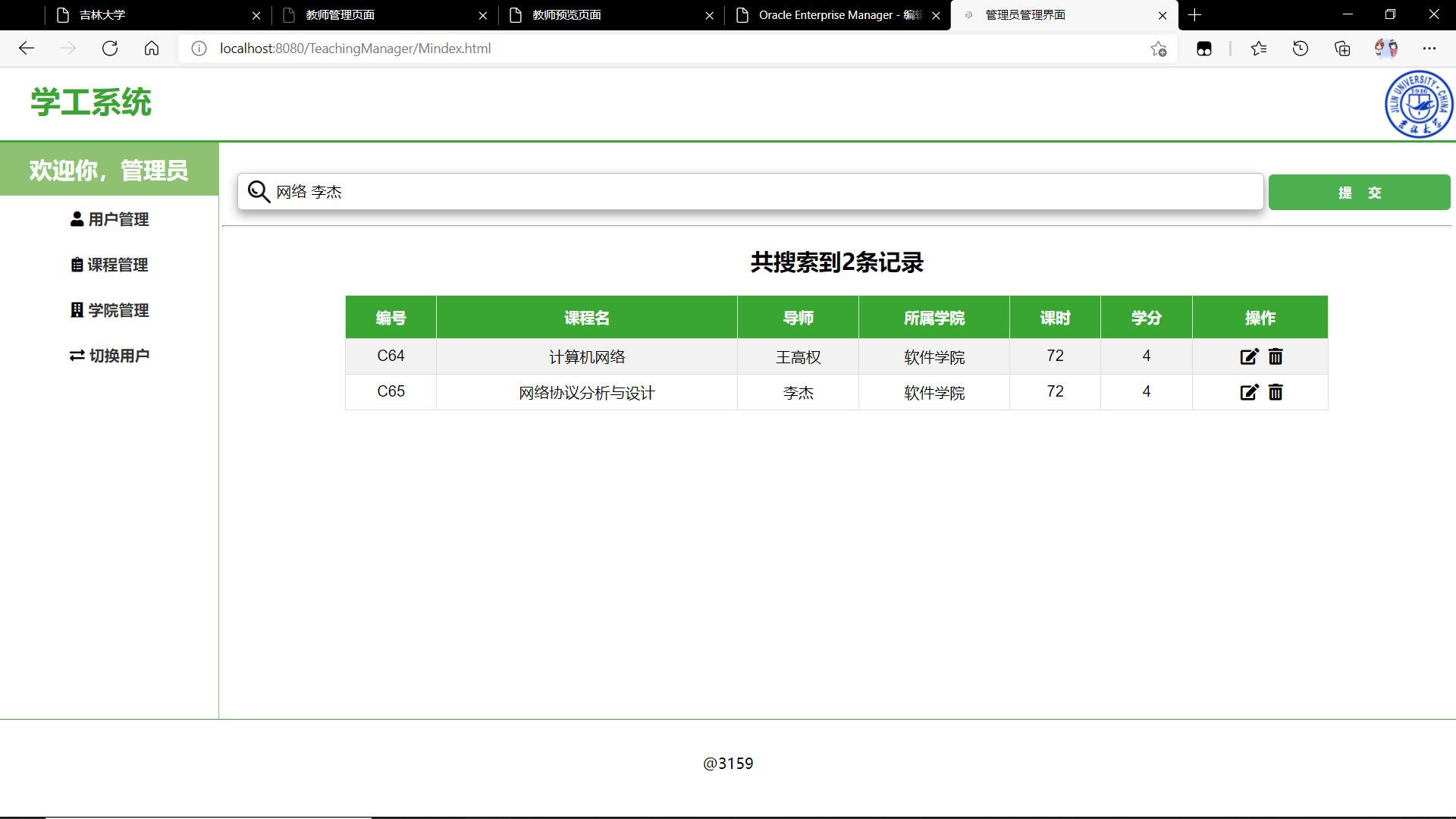
Sql语句采用全模糊形式进行select，这种方式效率不高且不能有效利用索引，但符合功能要求。

对搜索的课程列表可以编辑和删除，同样，对于不必要修改的数据可以不填写。

在教师管理中有管理教师所授课程的按钮，实际上就是预置了教师名关键词进入该网页实现的，同样可以在此页面实现一样的功能。但是考虑到系统的方便性，我们仍在用户管理处设置了快速入口。

具体演示参见演示视频。

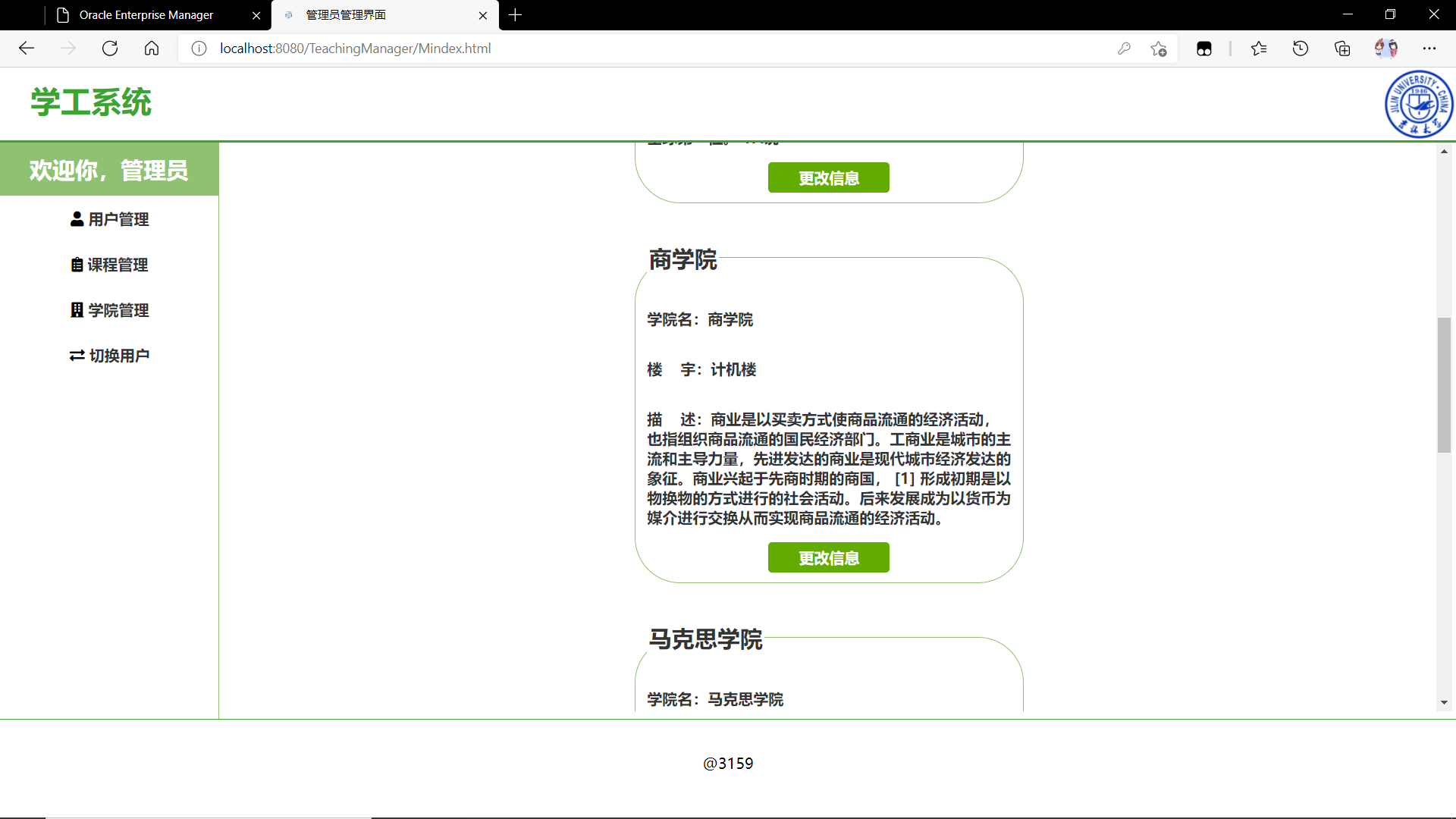




学院管理

在此界面会以多个<fieldset>列出所有学院的详细信息，点击对应的修改按钮会调用js函数弹出表单，提交表单至servlet后即可修改对应学院的信息。在学院信息中，“描述”信息是一个CLOB类型，可以容纳大量文字，因此在编辑表单中，该信息的输入用一个多行文本框<textarea>实现。

在后端数据库操作上，从结果集getClob()得到Clob类型数据后，以getSubString()转换为String类型，再掉用相应的set函数将该数据传入该表对应的已经实例化的Javabean类。插入Clob数据时，servlet从表单得到字符串形式的数据，作为参数调用数据库操作函数。将String类型的描述作为参数初始化一个StringReader类型，随后以PrepareStatement.serCharacterStream()将该字节流插入至事先定义好的sql语句的？占位符中。



点击底部开设学院按钮，同样弹出一个输入表单，输入学院信息提交表单至servlet后可以添加新的学院（描述允许为空）

具体操作参加演示视频。

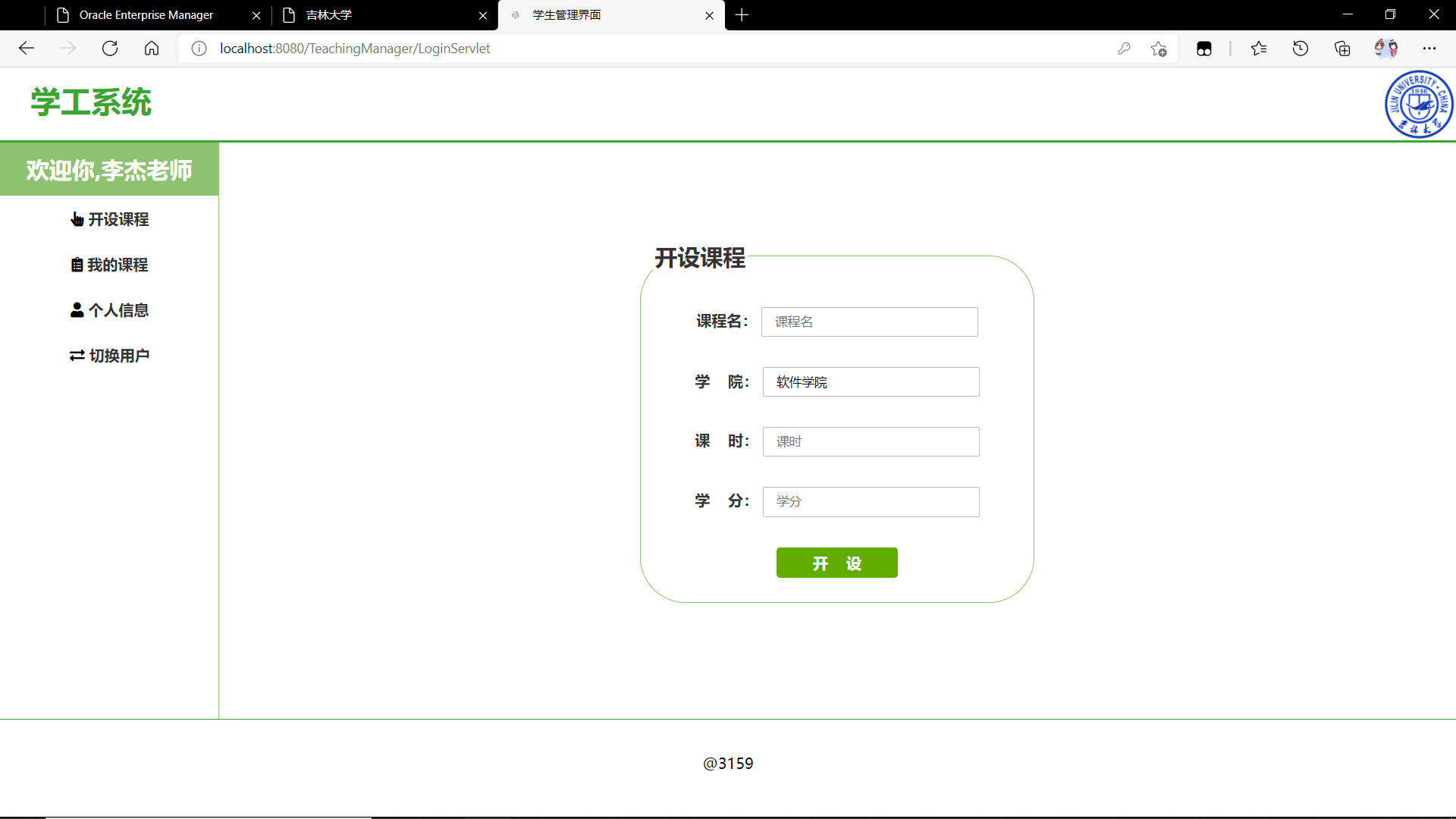


##### 教师模块

布局与管理员相似，与管理员有所差异的是进入时会读取登录时设置的session级变量ID来查询数据库得到教师姓名，在左侧显示问候消息。

开设课程

从左侧导航栏选择开设课程页签，右侧显示开设课程页面。页面包含一个课程信息表单，在此表单填写课程信息可以开设课程。课程的所属学院默认为该教师的所属学院，也允许后续修改。



我的课程

从左侧导航栏选择我的课程页签，右侧显示我的课程页面。页面包含一个该教师所开设的课程列表和课程数统计。

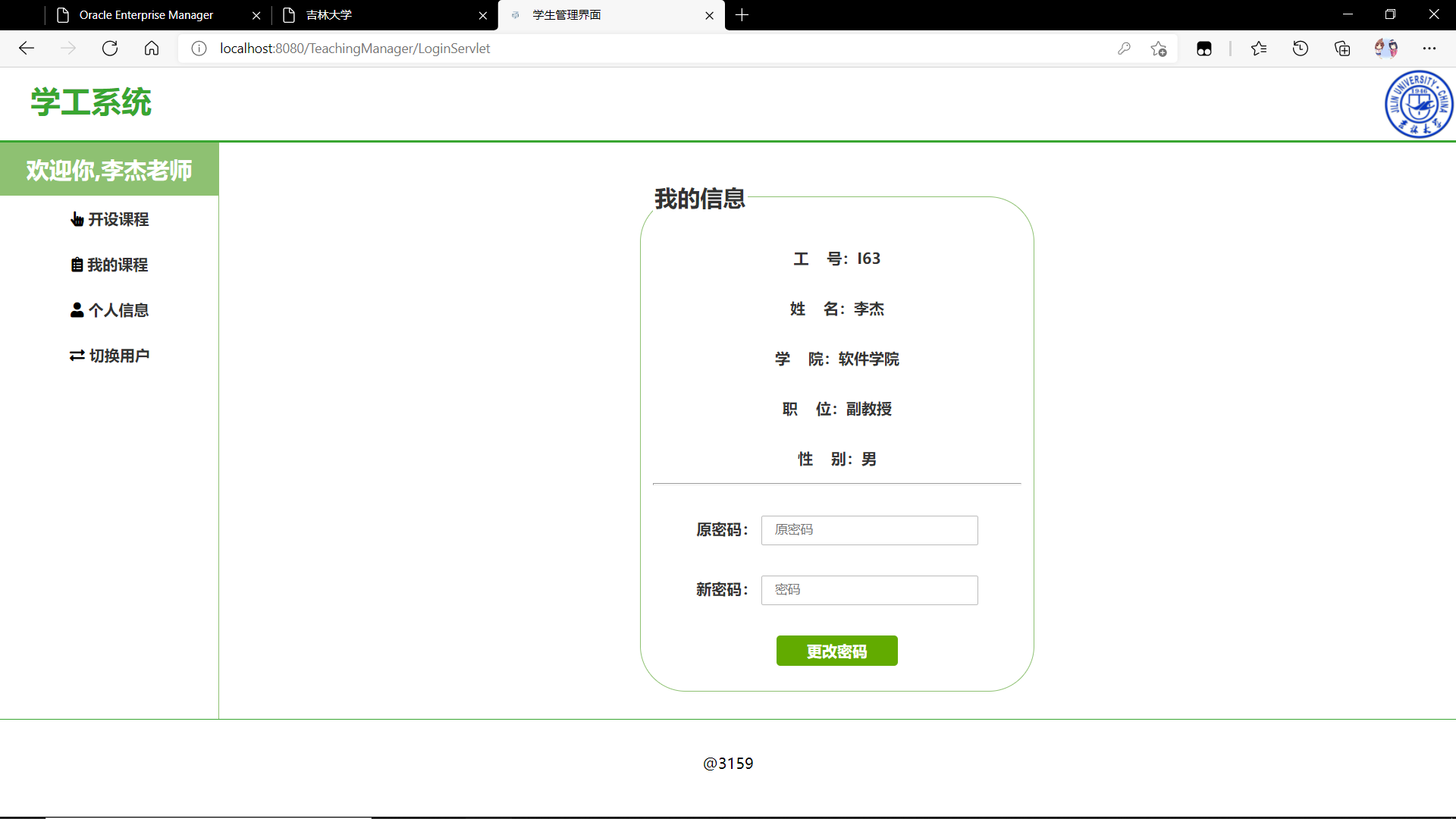


在列表一行的操作列，点击详细按钮进入管理该课程的学生页面。在此界面会以列表形式显示所有选了该门课程的学生信息，同时操作列可以判定某名学生通过该课程（以通过的学生不会显示操作）。判定通过该课程操作为将选课信息表中的COMPLETED列的值改为’Y’，触发器会自动为该学生的总学分加上相应的值。判断学生成绩通过时，同时也会往学生课程通过关系表插入一行，以供学生界面查看最新成绩模块使用。



个人信息

该界面显示该教师的全部个人信息以供浏览（不可修改），并可修改密码，但仅有管理员拥有修改除密码之外信息的权限。修改密码部分与管理员不同的是，修改自己密码时需要额外输入原密码，servlet会进行比对，比对正确后才会调用jdbc函数修改账户表中对应的一行。



具体操作参见演示视频。

##### 学生模块

布局与其余模块类似，与教师同样会在进入时提出session级的ID变量以供查找学生姓名在左侧显示问候信息。右侧网页默认为最新成绩页面。

网上选课

该页面复用了管理员模块的课程管理页面，仅在操作上有所不同，但该页面不会显示学生已选的课程，允许搜索课程，不输入关键词时，将显示所有未选过的课程。在课程列表的右侧操作列点击选课，在选课表中插入相应的一行（COMPLETED值为’N’，即未通过），选课成功。



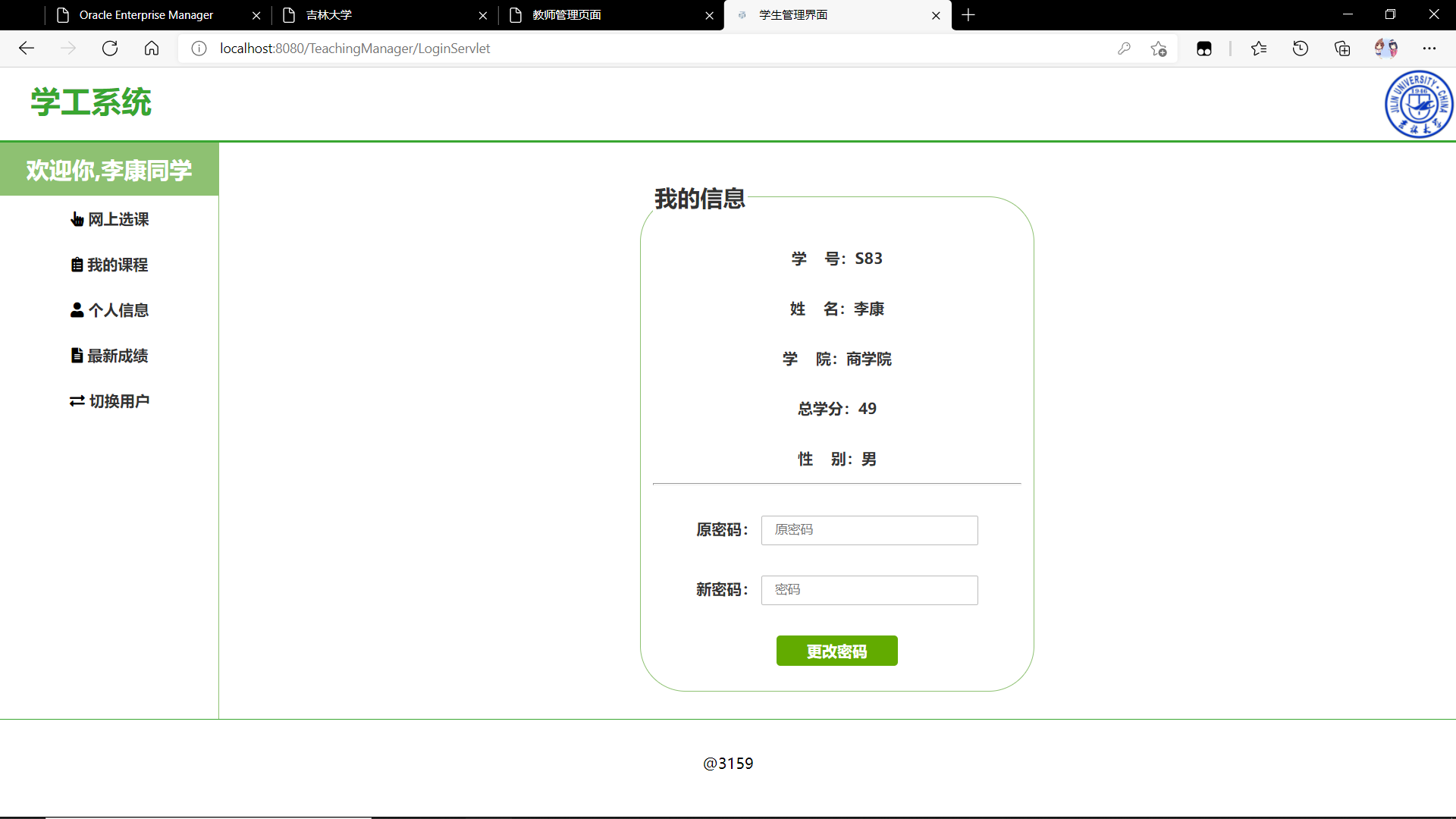
我的课程

该页面显示目前该学生选择的所有课程，并且在操作列提供了退课操作，在数据库选课表中删除相应的一行，操作列根据目前学生的课程完成情况而有所不同，学生已经通过的课程无法退课，不显示操作。管理员也能帮学生退课，实际上即复用了该页面。



个人信息

个人信息模块与教师的个人信息模块类似，同样显示了该学生的全部信息以供浏览，但是学生本人除密码外也无权修改，只有管理员有相应的权限。修改密码需要输入原密码，具体可参考教师同模块，实际上学生和教师使用了同一个jsp页面，在jsp中根据传入ID判断该显示学生信息或是教师信息。



最新成绩

该界面读取学生课程通过关系表的信息并以表格形式按时间倒序显示，可供学生了解最新的已通过成绩信息。



具体参见演示视频。

### 总结

在程序设计之初，我们对程序功能，即每个用户应当做什么做出了仔细的考量，没有过分赋予用户权限。比方说教师/学生，他们的个人信息能够自己修改的只有密码，乍一看似乎很不合理，但是我们做的是一个教学管理系统，需要考虑的是严谨性和保证数据可靠，而不是某个社交/购物网站，用户可以随意修改自己的昵称（姓名）/出生年月乃至性别，因而做出了这样的设计。

基于同样的理由我们也没有给教师赋予随意修改课程信息的权限，课程一旦发布课程信息除管理员外皆不可修改，因为课程课时/学分的变动对于教学来说是很大的影响，即使需要修改在正常使用中这种情况也应报至教务处（管理员）处理，当然我们没有设计教师与管理员的沟通渠道，但仅将修改权限赋予管理员也能起到一样的效果。

程序中大量使用了表格，表格往往依靠jsp循环实现，表格每一行的操作列都有操作按钮，都可以提交表单。那么如何在servlet中识别该进行何种操作乃至该队表格中哪一行进行操作？我们知道在提交表单时，按钮本身的值也会被提交，因此我们在servlet中识别按钮的值来判断应对数据库进行的操作，同时表单中往往有一行<input type=”hidden”>元素，该元素不会在页面显示，但是同样会被提交，我们通常令该元素的value值为表格应操作行的ID值，比如工号/学号/课程号，servlet识别该值以判断该对表格中哪一行进行操作。

关于程序的异常处理部分（主要为用户输入数据格式有误）则主要依靠数据库本身的约束来处理，servlet中捕获到SQLException异常后返回错误代码至前端网页，由js函数读取错误代码后由弹窗提示错误信息，若无错误该代码为0，前端会提示执行成功。将数据输入规约全部抛由后端数据库检验我们认为是很没有效率的一件事，至少应当在前端（由js等）检验一部分，这是程序应当改进的地方。

在程序设计初期我们花费了一些时间来设计CSS样式表来美化界面，并且在网上寻找了一些图标库来美化我们的按钮/导航栏页签，事实证明这是值得的（我们对于我们的界面设计比较满意）。CSS样式表可以由很多元素复用，因为一个网站的元素（表格等）和布局等往往都是相似乃至相同的，通过这次实验我理解到div+css对于网站美化是一件效率极高的事情。

##### 管理员模块

我们认为管理模块是最难实现的部分，因为他的功能最多，而且其余的两个模块可以大量复用他的功能，因此我们最先设计管理员，设计完管理员后后续工作就不是那么困难。

本实验的管理员并非像很多程序那样具有一切功能，反而具有诸多限制，比方说不能开设课程。这实际上是由管理员的属性决定的，在数据库设计时，我们设计了一个账户表，在此之上联系并设计了学生表和教师表，但是却并没有管理员表。这意味着管理员是没有任何私有属性的（我们认为也不需要），只存在账户名/密码的管理员账户实际上就是一个进入管理员账户的验证工具，自然是不能开课。这一点在管理员页面与其余页面的设计上也有所体现，管理员页面的任何功能不需要任何有关于管理员的信息，不像学生/教师需要自己的信息以方便查看自己的课程。

另外还有限制不能删除学院等是由于这个改动我们觉得过于重大，因为学院关联了太多数据（老师/学生/课程），因而没有加入这个功能。而现实中直接删除学院的情况也是极少数，多半是合并/改名，而这些修改学院信息功能足以应付。

##### 教师模块/学生模块

这两个模块大量复用了管理员模块的界面，但是复用往往不是完全一致。比方说课程管理，管理员需要显示所有课程，但是教师/学生只需要显示自己教/选的课程，除此之外在操作列上也有大的差异。这一点我们利用了jsp来识别界面应如何显示。我们知道登录教师/学生页面时他们的ID也会随之传过来，而管理员不会。这个ID在设计之初我们就约定教师以I开头，学生以S开头，课程以C开头，并以序列+触发器实现了ID的自动分配。因此在进入这种多用途界面时，往往先提取ID值，若为空则为管理员页面，否则由首字母决定。