



**课 程 设 计**

课程名称 数据库系统课程设计

题目名称 学生学籍信息管理系统

专业班级 2015软件4班

学 号 3115005369

学生姓名 肖经达

指导教师 何晓桃

2018年 1 月 12 日

[1.引言 1](#_Toc503560674)

[1.1课程设计选题 1](#_Toc503560675)

[1.2课程设计的目的 1](#_Toc503560676)

[1.3本选题的使用技术 1](#_Toc503560677)

[2.需求分析 1](#_Toc503560678)

[2.1信息需求 1](#_Toc503560679)

[2.2功能需求 1](#_Toc503560680)

[2.3安全性与完整性需求 2](#_Toc503560681)

[2.4数据流图 2](#_Toc503560682)

[2.5数据字典 5](#_Toc503560683)

[3.概念结构设计 7](#_Toc503560684)

[3.1E-R图 7](#_Toc503560685)

[4.逻辑结构设计 8](#_Toc503560686)

[4.1关系模型 8](#_Toc503560687)

[4.2用户子模式 9](#_Toc503560688)

[4.3安全性设计和实现说明 10](#_Toc503560689)

[4.4完整性设计和实现说明 11](#_Toc503560690)

[4.5 触发器 12](#_Toc503560691)

[5.数据库物理设计 14](#_Toc503560692)

[5.1索引 14](#_Toc503560693)

[5.2数据的存放位置 15](#_Toc503560694)

[5.3数据库内容 15](#_Toc503560695)

[5.4模块设计 16](#_Toc503560696)

[6.数据库实施 17](#_Toc503560697)

[7.数据库运行与维护 17](#_Toc503560698)

[8.收获和体会 25](#_Toc503560699)

# 1.引言

系统功能基本要求：对所有学生从入学到毕业的全部过程中的基本信息进行管理：学籍变动（转学、休学等），基本信息采集、毕业处理等。

## **1.1课程设计选题**

学生学籍信息管理系统

## **1.2课程设计的目的**

随着我国教育体制改革的深入进行，教育系统得到了前所未有的发展。学生管理正在逐步迈向管理信息现代化。学校的规模不断扩大,学生数量急剧增加,有关学生的各种信息量也成倍增长。由于大学生类别和专业的设置的多样化，大学生的学籍管理历来是非常繁琐和复杂的的工作。面对庞大的信息量, 这时的人工管理几乎无法实现，在这种情况下用数据库进行管理变的尤为必要，这可以发挥计算机的优势来提高学生管理工作的效率。

学生学籍管理系统是应对学生人数增多、信息量增大的问题，实现管理的现代化、化，逐步摆脱学生学籍管理的人工管理方式，提高学籍管理效率而开发的。望该开发程序能够解决学生信息存储、学生各种信息查询等一系列功能。

## **1.3本选题的使用技术**

开发环境：Windows 7

开发工具：NetBeans IDE 8.2

辅助工具：visio 2016

数据库管理系统：mysql

运行环境：Windows 7

开发技术：JSP+mysql

# 2.需求分析

## **2.1信息需求**

需要对所有学生从入学到毕业期间的所有信息进行提取，分类。如转学/休学信息，入学信息，毕业信息，学生基本信息。可以创建四个基本表，转学/休学表，学生基本表，毕业表，，用户登录表

## **2.2功能需求**

系统功能基本要求：对所有学生从入学到毕业的全部过程中的基本信息进行管理：学籍变动（转学、休学等），基本信息采集、毕业处理等。

本系统是针对高等院校的学生学籍管理，因此学籍管理系统的用户包括系统管理员、

学生。主要涉及学生信息、毕业处理、学籍变动（转学、休学等）等多种数据信息。

 实现的具体功能如下：

**前台：**

1.信息查询管理

学生，管理员查询学生基本信息或转学/休学或毕业信息

**后台：**

1.信息查询管理

学生，管理员查询学生基本信息或转学/休学或毕业信息

2.学籍变动处理

系统管理员负责对学生转学、休学的录入、修改、删除等操作。

 3.毕业生信息管理

系统管理员负责对毕业学生的录入、修改、删除等操作。

以上是用户对系统的基本的功能要求，此外用户还要求系统的效率要高，查询

速度要快，比较小的冗余，易维护，具有较高的数据安全性。

**功能模块图**



## **2.3安全性与完整性需求**

安全性：

设置两类用户，一类系统管理员，一类学生。给它们设置不同的权限，系统管理员具有对学生基本信息，转学/休学信息，毕业信息增删改查的功能；学生只有对学生基本信息，转学/休学信息，毕业信息查询功能

完整性：

学生的学号非空且唯一。

转学的编号非空且唯一。

休学的编号非空且唯一。

毕业学生的编号非空且唯一。

转学/休学表中的学号作为外码应与学生基本表中学号一致

毕业表中的学号作为外码应与学生基本表中学号一致

创建一些触发子，级联更新，删除。

## **2.4数据流图**

顶层图：

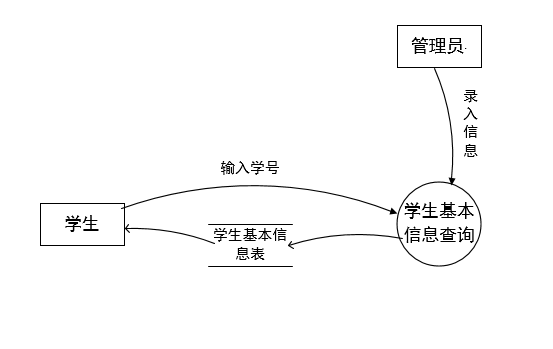


0层图

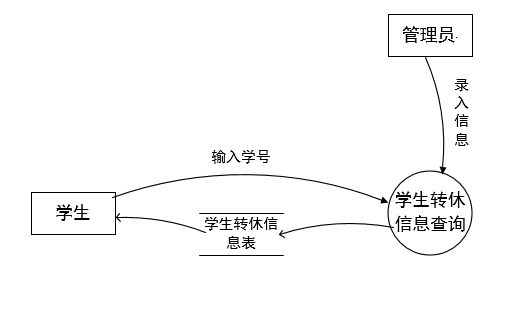


1层图

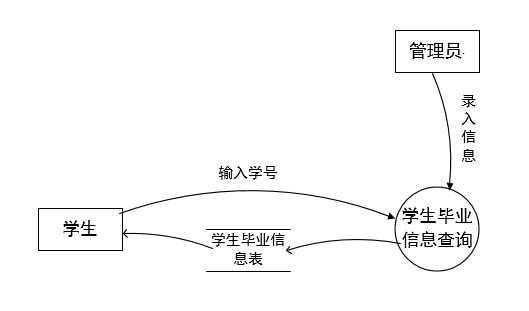
学生基本信息流程图



学生转休信息流程图



学生毕业信息流程图



## **2.5数据字典**

1．数据项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 别名 | 数据类型 | 数据长度 | 取值范围 | 简述 |
| 学号 | Sno | 字符型 | 9位 | 数字组成 | 惟一标识学生身份 |
| 姓名 | Sname | 字符型 | 10位 | 只能由中文组成 | 学生名唯一 |
| 性别 | Ssex | 字符型 | 2位 | 男|女 |  |
| 出生日期 | Sbirth | 日期 |  | 1900.1至今 |  |
| 籍贯 | Snative | 字符型 | 10位 |  | 出生省份 |
| 学院 | Sdept | 字符型 | 20位 | 中文 |  |
| 入学时间 | Sent | 日期 |  |  | 学生入学的时间 |
|  |  |  |  |  |  |
| 编号 | Bno | 字符型 | 3位 | 数字组成 | 变动编号 |
| 学号 | Sno | 字符型 | 9位 | 数字组成 | 惟一标识学生身份 |
| 变动时间 | Btime | 日期 |  | 1900.1至今 |  |
| 变动类型 | Btype | 字符型 | 4位 | 退学/休学 | 退学/休学 |
|  |  |  |  |  |  |
| 毕业编号 | Gno | 字符型 | 3位 | 数字组成 | 毕业编号 |
| 学号 | Sno | 字符型 | 9位 | 数字组成 | 惟一标识学生身份 |
| 毕业时间 | Gtime | 日期 |  | 1900.1至今 |  |
| 所修学分 | Gcredit | 整型 |  | smallint | 需大于32 |

2．数据结构

|  |  |
| --- | --- |
| 编号 | 01 |
| 名称 | 学生基本信息 |
| 描述 | 描述学生姓名，年龄等的基本信息 |
| 组成 | 基本={学号，姓名，性别，出生日期，籍贯，学院，入学时间} |
| 位置 | 学生基本表 |

|  |  |
| --- | --- |
| 编号 | 02 |
| 名称 | 学生转休信息 |
| 描述 | 描述学生转学/休学基本信息 |
| 组成 | 转休={编号，学号，变动时间，变动类型} |
| 位置 | 学生转休表 |

|  |  |
| --- | --- |
| 编号 | 03 |
| 名称 | 学生毕业信息 |
| 描述 | 描述学生毕业处理基本信息 |
| 组成 | 毕业={毕业编号，学号，毕业时间，所修学分} |
| 位置 | 学生毕业表 |

3．部分数据流图

|  |  |
| --- | --- |
| 名称 | 录入基本信息 |
| 含义 | 管理员录入学生基本信息 |
| 来源 | 管理员获得的基本学生信息 |
| 去向 | 学生基本表 |
| 组成 | 学号，姓名，性别，出生日期，籍贯，学院，入学时间 |

|  |  |
| --- | --- |
| 名称 | 录入转休信息 |
| 含义 | 管理员录入学生转学/休学信息 |
| 来源 | 管理员获得的转学/休学学生信息 |
| 去向 | 学生转休表 |
| 组成 | 编号，学号，变动时间，变动类型 |

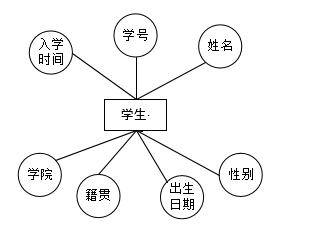
|  |  |
| --- | --- |
| 名称 | 录入毕业信息 |
| 含义 | 管理员录入学生毕业信息 |
| 来源 | 管理员获得的毕业学生信息 |
| 去向 | 学生毕业表 |
| 组成 | 毕业编号，学号，毕业时间，所修学分 |

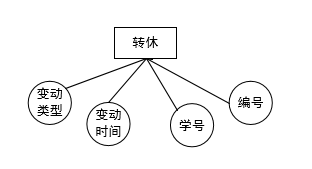
4．部分处理过程

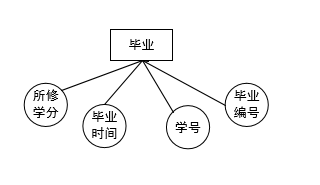
|  |  |
| --- | --- |
| 名称 | 查询基本信息 |
| 含义 | 对学生的基本信息进行查询 |
| 输入数据流 | 学生基本表 |
| 处理 | 显示输出学生基本信息 |
| 输出数据流 | 学生基本查询信息 |
| 名称 | 查询转休信息 |
| 含义 | 对学生的转学/休学信息进行查询 |
| 输入数据流 | 学生转休表 |
| 处理 | 显示输出学生转休信息 |
| 输出数据流 | 学生转学/休学查询信息 |

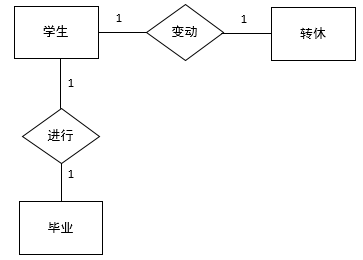
# 3.概念结构设计

## **3.1E-R图**

学生实体ER图

转休实体ER图

毕业实体ER图

总体ER图

# 4.逻辑结构设计

## **4.1关系模型**

建立的表：

用户表：（用户名，密码，用户类型）3NF

学生基本表：（学号，姓名，性别，出生日期，籍贯，学院，入学时间）3NF

学生转休表：（编号，学号，变动时间，变动类型）3NF

学生毕业表：（毕业编号，学号，毕业时间，所修学分）3NF

学生基本表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 属性 | 数据类型 | 数据长度 | 备注 |
| 学号 | Sno | char | 9位 | 主码 |
| 姓名 | Sname | char | 10位 | 唯一 |
| 性别 | Ssex | char | 2位 | 男/女 |
| 出生日期 | Sbirth | date |  | 非空 |
| 籍贯 | Snative | char | 10位 | 非空 |
| 学院 | Sdept | char | 20位 | 非空 |
| 入学时间 | Sent | date |  | 非空 |

学生转休表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 属性 | 数据类型 | 数据长度 | 备注 |
| 编号 | Bno | char | 3位 | 主码 |
| 学号 | Sno | char | 9位 | 外码非空 |
| 变动时间 | Btime | date |  | 非空 |
| 变动类型 | Btype | char | 4位 | 转学/休学 |

学生毕业表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 属性 | 数据类型 | 数据长度 | 备注 |
| 毕业编号 | Gno | char | 3位 | 主码 |
| 学号 | Sno | char | 9位 | 外码非空 |
| 毕业时间 | Gtime | date |  | 非空 |
| 所修学分 | Gcredit | smallint |  | 非空大于32 |

用户表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 属性 | 数据类型 | 数据长度 | 备注 |
| 用户名 | Yname | char | 9位 | 主码 |
| 密码 | Ypass | char | 9位 | 非空 |
| 用户类型 | Ytype | char | 4位 | 非空 |

## **4.2用户子模式**

1．学生基本信息视图

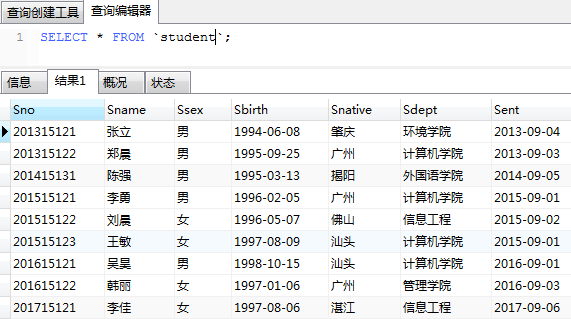
create view student

as

select Sno,Sname,Ssex,Sbirth,Snative,Sdept,Sent

from xjd\_student;

查询学生基本信息内容：student为视图



2．学生转学/休学视图

create view zx

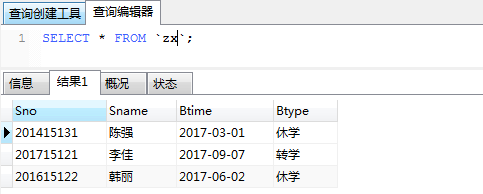
as

select xjd\_zx.Sno,Sname,Btime,Btype

from xjd\_student,xjd\_zx

where xjd\_student.Sno=xjd\_zx.Sno;

查询学生转学/休学信息：zx为视图



3．学生毕业信息视图

create view gra

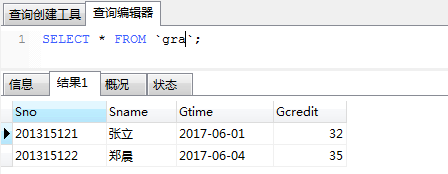
as

select xjd\_by.Sno,Sname,Gtime,Gcredit

from xjd\_student,xjd\_by

where xjd\_student.Sno=xjd\_by.Sno;

查询毕业学生信息：gra为视图



## **4.3安全性设计和实现说明**

(1)当前用户为dbo,是系统管理员，可以通过sql语句赋予dbo对数据库xjd中

   表xjd\_student,xjd\_zx,xjd\_by的所有特权，不过这是默认的，例如：

学生基本信息

grant all privileges

   on xjd\_student

   to dbo

   with grant option

学生转学/休学

grant all privileges

   on xjd\_zx

   to dbo

   with grant option

学生毕业信息

grant all privileges

   on xjd\_by

   to dbo

   with grant option

(2)先在管理器中建立一个登录用户，再在xjd数据库中建立一个用户sh,假设他

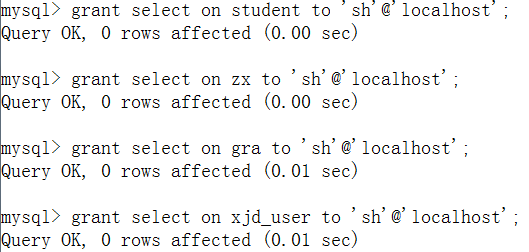
   是一名学生，则可以赋予他对所有表的select权限。



grant select on student to sh;

grant select on zx to sh;

grant select on gra to sh;



## **4.4完整性设计和实现说明**

创建学生基本表

create table xjd\_student(

Sno char(9) primary key,

Sname char(10) unique,

Ssex char(2) not null,

Sbirth date not null,

Snative char(10) not null,

Sdept char(20) not null,

Sent date not null

);

创建学生转休表

create table xjd\_zx(

Bno char(3) primary key,

Sno char(9) not null,

Btime date not null,

Btype char(4) not null,

foreign key(Sno) references xjd\_student(Sno)

on delete cascade on update cascade

);创建学生毕业表

CREATE TABLE xjd\_by (

Gno CHAR(3) PRIMARY KEY,

Sno CHAR(9) NOT NULL,

Gtime DATE NOT NULL,

Gcredit SMALLINT NOT NULL,

FOREIGN KEY (Sno)

REFERENCES xjd\_student (Sno)

on delete cascade on update cascade

);创建用户表

create table xjd\_user(

Yname char(3) primary key,

Ypass char(9) not null,

Ytype char(4) not null

);

实体完整性：通过设置主键都已实现xjd\_student中Sno为主码，xjd\_zx中Bno为主码，xjd\_by中Gno为主码。

参照完整性：xjd\_zx中Sno与xjd\_by中Sno为外码。对应xjd \_student中的Sno。

用户自定义完整性：mysql不支持check，建立触发子

## **4.5 触发器**

学生基本表中性别只能取男女

delimiter #

create trigger T1 before insert on xjd\_student

for each row begin

if

new.Ssex <> '男'

and new.Ssex <> '女' then

delete from xjd\_student where Ssex = new.Ssex;

end if;

end;

#

create trigger T2 before update on xjd\_student

for each row begin

if

new.Ssex <> '男'

and new.Ssex <> '女' then

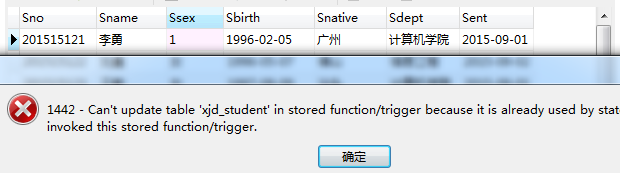
delete from xjd\_student where Ssex = new.Ssex;

end if;

end;

#

delimiter ;



学生转休表中Btype只能取转学或休学

delimiter #

create trigger T5 before insert on xjd\_zx

for each row begin

if

new.Btype <> '转学'

and new. Btype <> '休学' then

delete from xjd\_zx where Btype = new. Btype;

end if;

end;

#

create trigger T6 before update on xjd\_zx

for each row begin

if

new. Btype <> '转学'

and new. Btype <> '休学' then

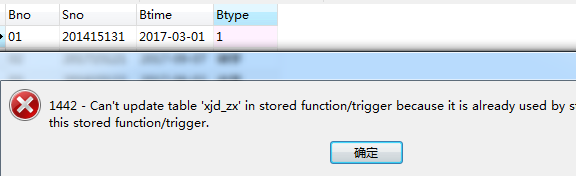
delete from xjd\_zx where Btype = new. Btype;

end if;

end;

#

delimiter ;



学生毕业表中学分要大于或等于32

delimiter #

create trigger T3 before insert on xjd\_by

for each row begin

if

new.Gcredit <32

then

delete from xjd\_by where Gcredit = new.Gcredit;

end if;

end;

#

create trigger T4 before update on xjd\_by

for each row begin

if

new.Gcredit <32

then

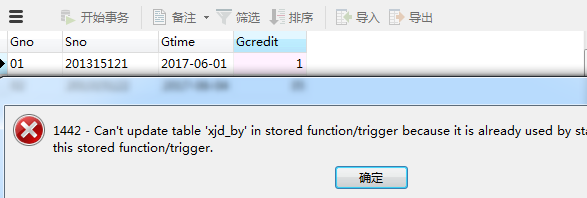
delete from xjd\_by where Gcredit = new.Gcredit;

end if;

end;

#

delimiter ;



# 5.数据库物理设计

## **5.1索引**

(1)所有表中的主键自动被设为索引项

(2)由于属性“学号”经常在连接操作中出现 转休表，毕业表中Sno查询次数很高，故在Sno上建立索引

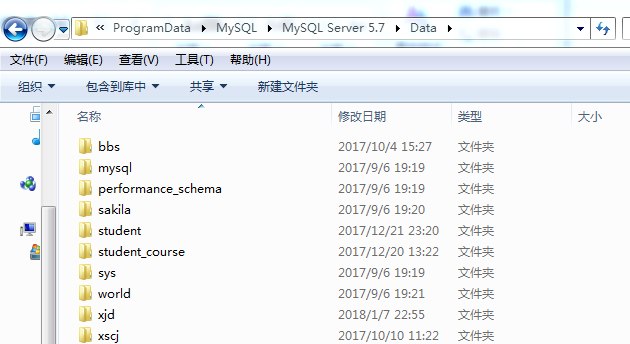
create unique index zxsno on xjd\_zx(Sno);

create unique index bysno on xjd\_by(Sno);

(3)学生表中的“姓名”经常作为各种信息查询活动内容被查询，可以建立相应索引

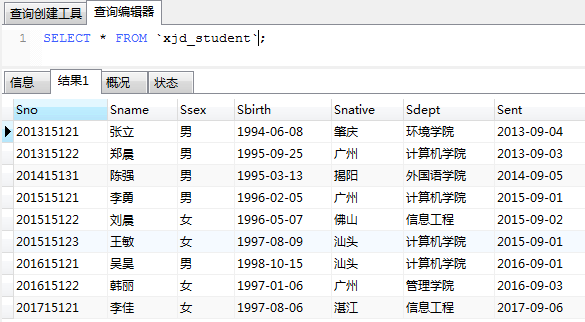
create unique index sname on xjd\_student(Sname);

## **5.2数据的存放位置**

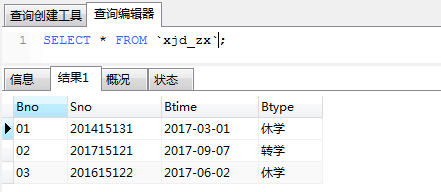


## **5.3数据库内容**

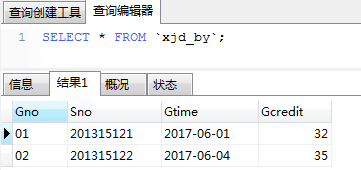
学生基本表内容：



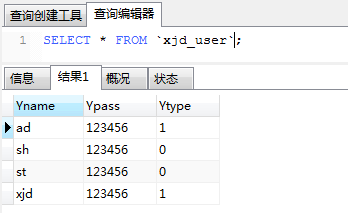
学生转休表内容：



学生毕业表内容：



用户表内容：



## **5.4模块设计**

P1

学生基本信息管理

P2

学生转休信息管理

P3

学生毕业信息管理

管理员

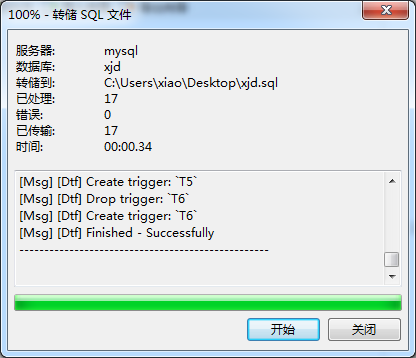
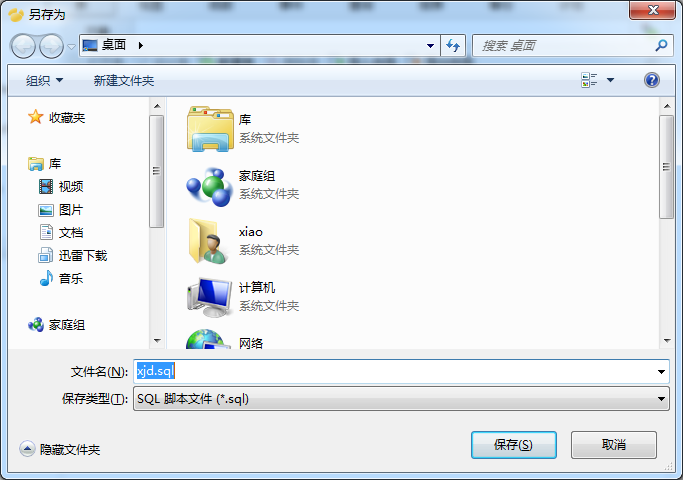
学生基本信息表

学生转休信息表

学生毕业信息表

学生

# 6.数据库实施



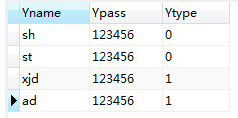
以附录形式给出创建数据库及数据库对象的SQL脚本文件

# 7.数据库运行与维护

系统使用说明：2－3个程序运行的主要窗口的截图；具有代表性的功能的实现方法

的说明和主要代码（要有较完整的注释），要求能够体现程序设计中的重点和难点及设计者的能力及主要工作。

系统测试方案和测试报告。



用户的登录帐号和密码，0表示学生，只有查询视图权限，上面安全性设计有授予权限。1表示管理员，是dbo系统管理员用户。具有增删改查权限。

**登录界面：**



界面实现自主存取控制方法：

<%

String username =new String(request.getParameter("userName").getBytes("ISO-8859-1"),"UTF-8");

String password =new String(request.getParameter("password").getBytes("ISO-8859-1"),"UTF-8");

String chioce =new String(request.getParameter("systemrole").getBytes("ISO-8859-1"),"UTF-8");

/\*获取登录界面输入的用户名，密码和（学生或管理员）\*/

if(chioce.equals("administer"))//如果选择管理员，用系统管理员root登录数据库

{

Connection con = null;

Statement st = null;

ResultSet rs = null;

if(userName.equals("")) {

response.sendRedirect("login.jsp");

}

try{

Class.forName("com.mysql.jdbc.Driver");

/\*url后面加的?useUnicode=true&characterEncoding=gbk，是为了处理向数据库中添加数据时出现乱码的问题。\*/

String url="jdbc:mysql://localhost:3306/xjd?useUnicode=true&characterEncoding=gbk";

con=DriverManager.getConnection(url,"root","138248");//连接数据库

st=con.createStatement();

String query="select \* from xjd\_user where Yname='" + userName + "'"+"and Ypass='"+password+"'"+"and Ytype='"+1+"'";

//检索xjd\_user表，对Ytype=1（管理员）检索

rs=st.executeQuery(query);

if(rs.next()){

response.sendRedirect("../main/admain.jsp");//成功，转向管理员登录界面

}else{

response.sendRedirect("login.jsp");//失败，重新转向登录界面

}

}catch(Exception e){

e.printStackTrace();

}finally{

rs.close();

st.close();

con.close();

}

}

else (chioce.equals("student"))//否则为student用户，用创建的user，sh登录数据库

{

Connection con = null;

Statement st = null;

ResultSet rs = null;

if(userName.equals("")) {

response.sendRedirect("login.jsp");

}

try{

Class.forName("com.mysql.jdbc.Driver");

/\*url后面加的?useUnicode=true&characterEncoding=gbk，是为了处理向数据库中添加数据时出现乱码的问题。\*/

String url="jdbc:mysql://localhost:3306/xjd?useUnicode=true&characterEncoding=gbk";

con=DriverManager.getConnection(url,"sh","123456");

st=con.createStatement();

String query="select \* from xjd\_user where Yname='" + userName + "'"+"and Ypass='"+password+"'"+"and Ytype='"+0+"'";

//检索xjd\_user表，对Ytype=0（学生）检索

rs=st.executeQuery(query);

if(rs.next()){

response.sendRedirect("../main/stmain.jsp");//成功，转向学生登录界面

}else{

response.sendRedirect("login.jsp");//失败，重新转向登录界面

}

}catch(Exception e){

e.printStackTrace();

}finally{

rs.close();

st.close();

con.close();

}

}

%>

**学生登录界面**



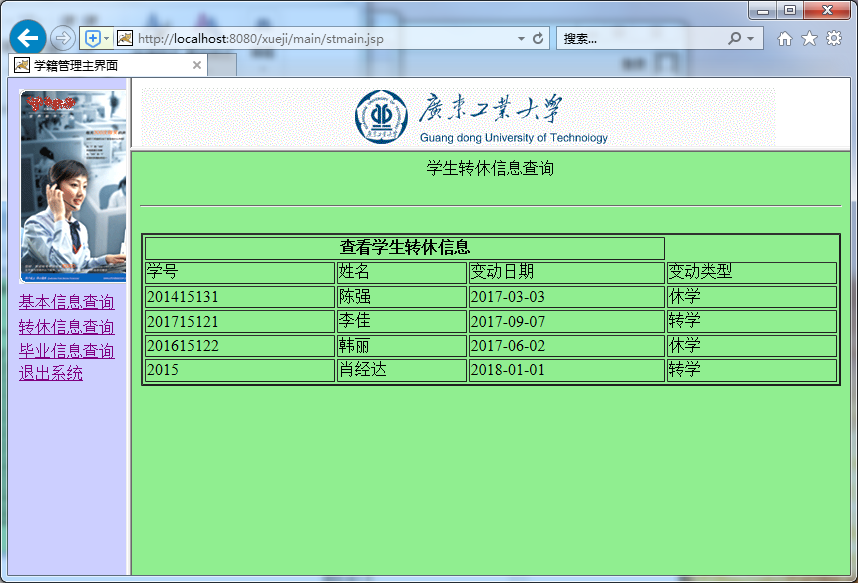
**学生基本信息查询**

通过sh，user查询student视图完成



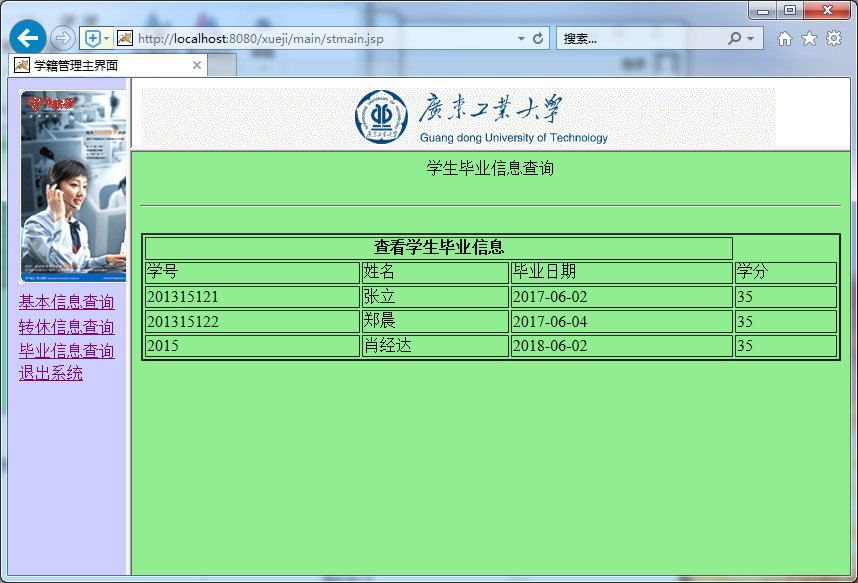
**学生转休信息查询**

通过sh，user查询zx视图完成



**学生毕业信息查询**

通过sh，user查询gra视图完成



**管理员登录界面**

即dbo用户



**基本信息管理界面**

信息查询通过dbo，user查询student视图完成



**信息删除界面** 通过dbo删除xjd\_student记录



**删除成功**



**信息添加界面** 通过dbo insert xjd\_student 完成



**添加成功**



添加一些不合法的界面



**信息修改** 通过dbo update xjd\_student 完成（学号修改成201515131等）



要是输入不合法数据



# 8.收获和体会

这次课设时间紧，任务重，由于大二暑假时学过一点javaweb，开发技术便选了JSP+mysql，这几天的课设时间真的学到了很多东西，学会了数据库的设计，概念模型设计，模式分解，jsp界面开发等，由于第一次接触数据库系统开发，在数据库连接，自主存取控制上花费了一天时间，最后还是有些不足，但是基本实现了自主存取控制。系统的优点就是由于用jsp做的，界面感觉还是可以的，数据库也实现了第三范式。不足：没有实现学生账号，密码登录，即进去只可以查看自己学号信息，其他信息没权限查询，没有用javabean实现，没有将jsp界面代码和java代码分开，也没有加框架。以后可以在这方面拓展。完善设想：可以增加多些功能，添加框架等，可以参考广工教务系统来做界面和功能。