

报告成绩	
演示成绩	
答辩成绩	
总评	

中北大学

课程设计说明书

数据库课程实验周

学生姓名： 刘洋 学 号： 1607084138

学 院： 大数据学院

专 业： 网络工程

题 目： 学生成绩管理系统

——课程管理及选课子系统模块

指导教师： 王丽芳

2018 年 6 月 20 日

1 系统设计

1.1 设计目标

实现“学生成绩管理系统”，包括教师端口，教务处端口，学生端口。

1.1.1 学生端口

学生基本信息查询、学生选课管理（对现有的课程信息进行查询，保存学生选课信息），学生成绩查询（教师通过教师端口录入成绩之后然后学生便可以通过各种查询方式显示出自己各科的成绩）

1.1.2 教师端口

成绩录入和成绩查询两部分。首先教师可以通过成绩录入对成绩进行添加和维护，然后可以查询自己所授课程的学生成绩。

1.1.3 教务处端口

对教师，学生，课程信息进行插入，删除，查询，修改。

1.2 需求描述

学生成绩管理系统是为学生老师以及教务处管理成绩服务的，该系统的基本需求如下：

1.2.1 信息要求

该系统主要记录学生、教师和课程的基本情况。

学生基本信息

学生基本信息包括：学号、姓名、性别、年龄、系别和密码。

教师基本信息

教师基本信息包括：教师号、姓名、性别和密码。

课程基本信息

课程基本信息包括：课程编号、课程名、学分、学时和备注。

1.2.2 安全性与完整性需求

1) 安全性要求

① 教务处管理员具有开启和关闭成绩录入功能，教师必需在规定的时间内

录入成绩，系统关闭后，教师不能继续录入成绩。

② 教务处管理员具有开启和关闭课程选课功能，学生必需在规定的时间内完成课程选课，系统关闭后学生不能继续进行选课。

③ 用户登录系统时，需要输入正确的用户名、密码和身份，方可登陆系统。

④ 学生登录系统后，只能查询自己个人信息、进行选课和查看自己的成绩。

⑤ 老师登录系统后，只能查询自己信息及本人录入的所授课成绩。

2) 完整性需求

①实体完整性

学号、教师号、课程编号分别为学生、教师、课程的主码。

②参照完整性

授课老师的教师号与教师基本信息中的教师号相对应，老师所授课的课程号与课程基本信息中的课程号相对应，选课学生学号与学生基本信息中的学号相对应，学生所选课的课程号与课程基本信息中的课程号相对应。

③用户定义的完整性

学生和教师信息中的性别为：男、女；

学生、老师和管理员信息中的密码：六到十个字符；

学生表中学号、姓名、性别、系别和密码五个属性非空，教师表中教师号、姓名和密码、性别四个属性非空，课程表中课程名、课程号、学时、学分四个属性非空，选课表中学号、课程号两个属性非空。

1.2.3 处理需求

该系统的基本业务要求为：

一名教师可以录入多个学生的成绩，教师按照讲授的课程录入对应学生的成绩；

一名教师可以讲授一门课程，一个课程可以被多名教师讲解；

一名学生可以选修多门课程，一个课程可以被多名学生选择。

根据学生管理业务情况，该系统共分成管理员、教师、学生三大系统模块。

1) 管理员子系统

① 录入学生信息；

② 修改学生信息；

③ 删除学生信息；

- ④ 录入老师信息;
- ⑤ 修改老师信息;
- ⑥ 删除老师信息;
- ⑦ 添加课程信息;
- ⑧ 修改课程信息;
- ⑨ 删除课程信息;
- ⑩ 维护选课关系信息。

2)教师子系统

- ① 查看自己教授的课程;
- ② 查看选择自己课程的学生;
- ④ 给学生课程打分;
- ⑤ 修改学生分数。

3)学生子系统

- ① 查看个人信息;
- ② 浏览课程;
- ③ 查看课程信息;
- ④ 查看自己课程分数;
- ⑤ 进行选课。

1.3 开发和运行环境选择

开发工具： 前台开发语言为 Java，后台数据库为 MySQL

运行环境： windows 2000 及更高的操作系统

2 数据库设计

2.1 数据库概念结构设计

数据库概念结构设计完成后，E-R 图如图 2.1 所示：

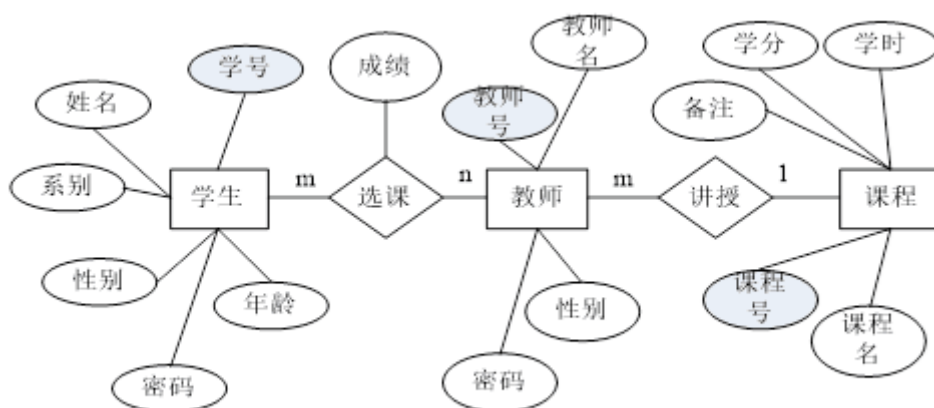


图 2.1 基本 E-R 图

2.2 数据库逻辑结构设计

根据 E-R 图设计出表结构如表 2.1、表 2.2、表 2.3、表 2.4 所示：

表 2.1 学生表(Student)

中文含义	字段名称	数据类型	长度	是否为空	备注
学号	Sno	Varchar	10	No	主码
姓名	Sname	Varchar	10	No	
性别	Ssex	Varchar	2	No	男 or 女
年龄	Sage	Varchar	3	Yes	大于 0，小于 100
系别	Sdept	Varchar	20	No	
密码	Spassword	Varchar	10	No	默认 123456789

课程设计说明书

表 2.2 教师表 (Teacher)

中文含义	字段名称	数据类型	长度	是否为空	备注
教师号	Tno	Varchar	10	No	主码
姓名	Tname	Varchar	10	No	
性别	Tsex	Varchar	2	No	男 or 女
密码	Tpassword	Varchar	10	No	最大 10 位字符
课程号	Cno	Varchar	10	Yes	外码

表 2.3 课程表 (Course)

中文含义	字段名称	数据类型	长度	是否为空	备注
课程号	Cno	Varchar	10	No	主码
课程名	Cname	Varchar	10	No	
学时	Ctime	Varchar	10	No	
学分	Ccredit	Varchar	10	No	
备注	Cremarks	Varchar	50	Yes	

表 2.4 选课表 (st)

中文含义	字段名称	数据类型	长度	是否为空	备注
选课成绩	Grade	Float	0	No	
学号	Sno	Varchar	10	No	主码
课程号	Tno	Varchar	10	No	主码

3 学生成绩管理系统（课程管理及学生选课子系统）详细设计

课程管理及学生选课子系统实现功能：完成与课程有关的功能。对课程表的维护和完成学生选课功能。

3.1 学生选课

学生在学生页面点击后进入选课界面，勾选课程。将选择的课程保存到数据库。程序运行流程图如图 3.1 所示：

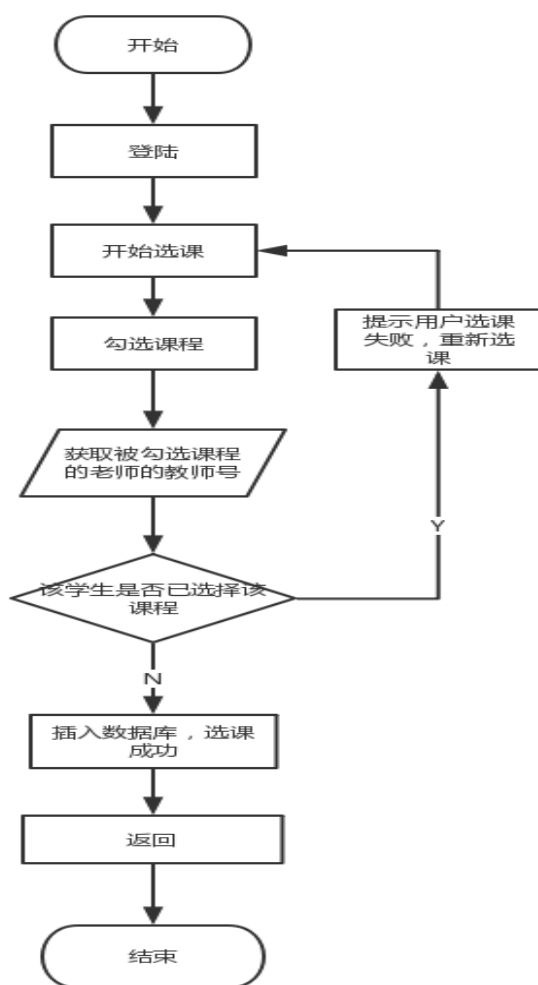


图 3.1. 选课流程图

课程设计说明书

该部分运行截图如图 3.2、图 3.3、图 3.4 所示



图 3.2 勾选课程截图

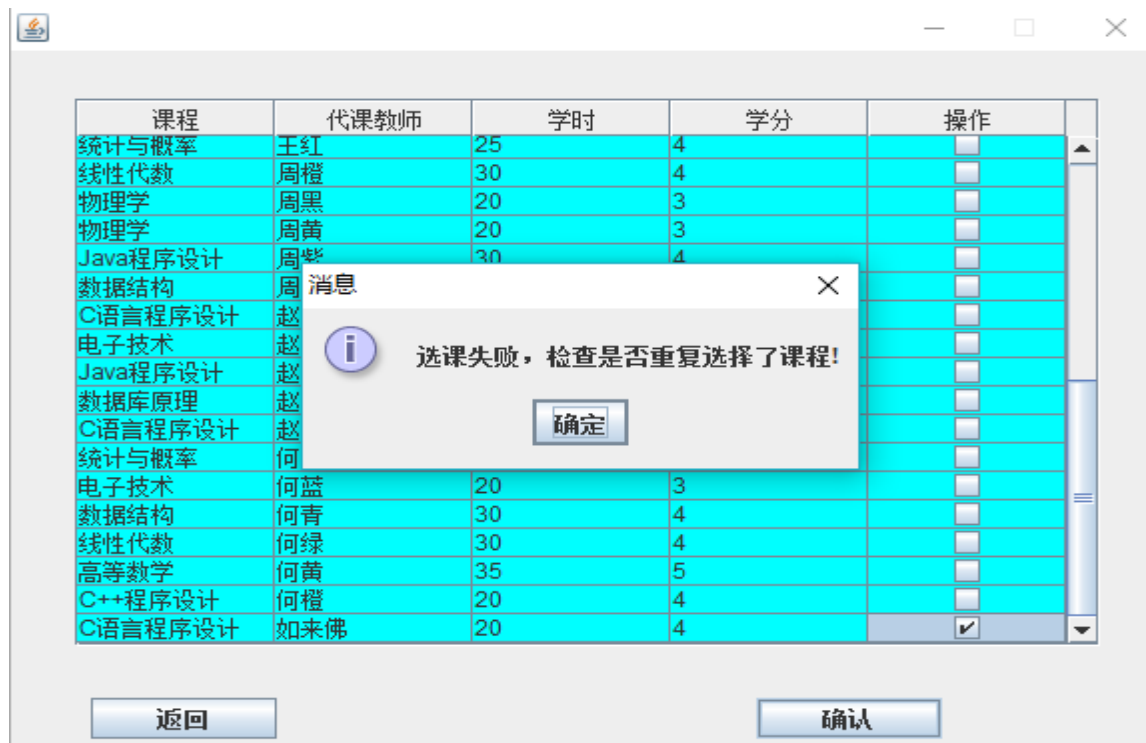


图 3.3 选择重复课程提示截图

课程	代课教师	成绩
物理学	沙僧	0.0
电路	王小橙	0.0
计算机网络	何小紫	0.0
数值分析	何小青	0.0
Linux编程	何小绿	0.0
毛泽东思想概论	何小红	0.0
微机原理	赵小黄	0.0
近代史纲要	周小红	0.0
统计与概率	何紫	0.0
线性代数	何绿	0.0
高等数学	何黄	0.0
C++程序设计	何橙	0.0
C语言程序设计	如来佛	70.0

图 3.4 运行截图

3.1.1 界面控制及核心数据库操作代码

```

public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("确认")){
        List<String> tnos = new ArrayList<>();
        for(int i=0;i<model.getRowCount();i++){

            System.out.println(model.getValueAt(i,model.getColumnCount()-1))

        }

        if((boolean)model.getValueAt(i,model.getColumnCount()-1)){
            tnos.add(teachers.get(i).getTno());
        }
    }
}
    
```

```
int a = stService.addBatch(sno, tnos);
if(a==1) {
    JOptionPane.showMessageDialog(this.frame, "选课成功!");
    new StudentView(sno).frame.setVisible(true);
    this.frame.dispose();
}else if(a==-1) {
    JOptionPane.showMessageDialog(this.frame, "选课失败，检查是否重复选择了课程!");
}else if(a==0) {
    JOptionPane.showMessageDialog(this.frame, "请选择课程!");
}
}else if(e.getActionCommand().equals("返回")) {
    new StudentView(sno).frame.setVisible(true);
    this.frame.dispose();
}
}
}

public boolean addOne(String sno,String tno) {
    Connection conn = getConnection();
    try {
        PreparedStatement ps = conn.prepareStatement("insert into
st(Sno,Tno) values(?,?)");
        ps.setString(1, sno);
        ps.setString(2, tno);
        if(ps.executeUpdate()>0) {
            return true;
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
```

```
        e.printStackTrace();
    }finally{
        try {
            conn.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    return false;
}
```

3.2 课程管理

进入管理员界面后可以进行选择对课程的操作。进入课程增加页面后输入需要添加的课程信息提交后即可保存。进入查询更改信息界面后，先会显示查询出来的课程信息，然后可以在列表直接更改，点击提交后即可保存。进入删除课程页面后先输入要删除的课程号，点击提交即可删除。程序运行流程图如图 3.5 所示：

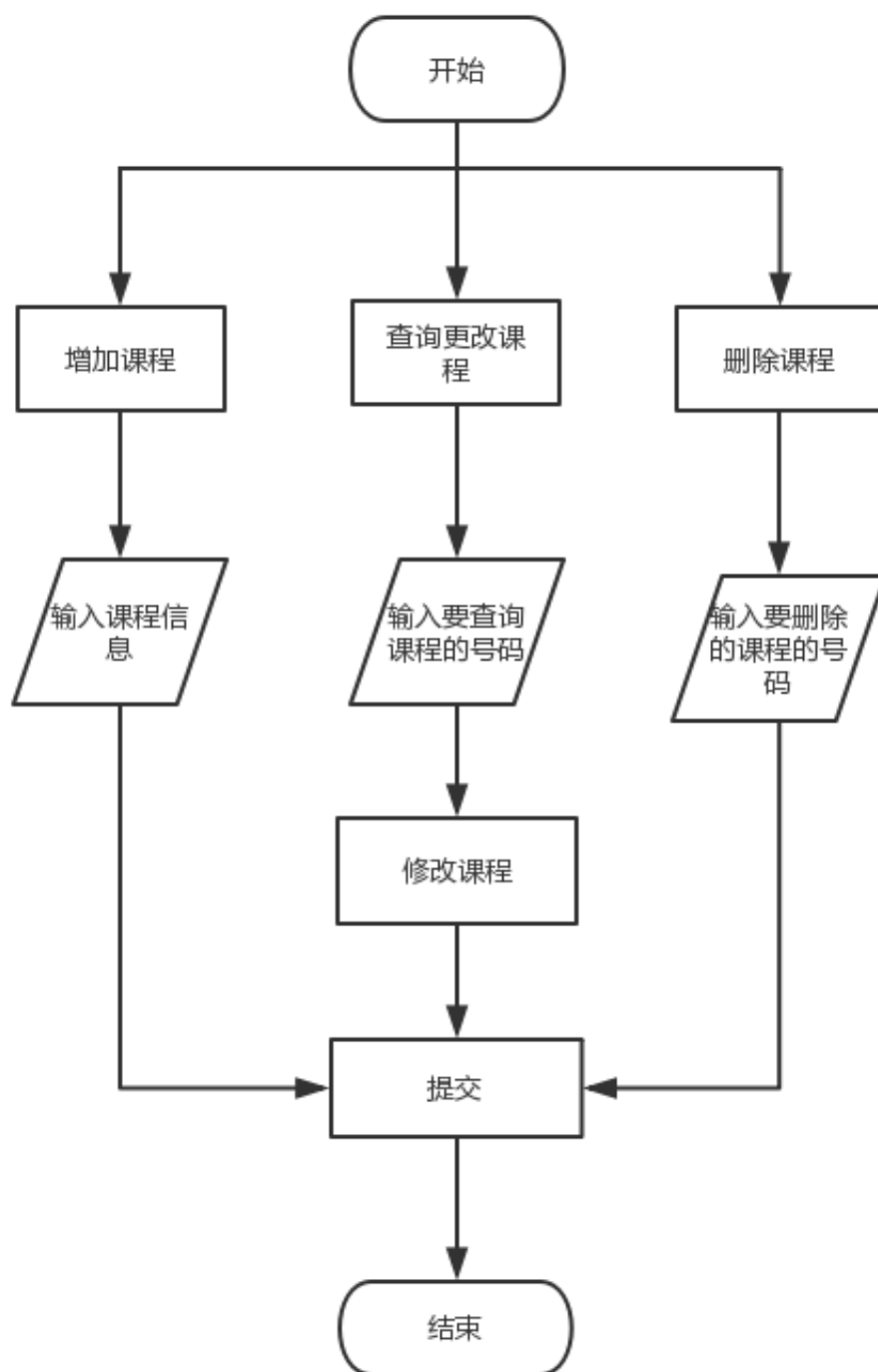
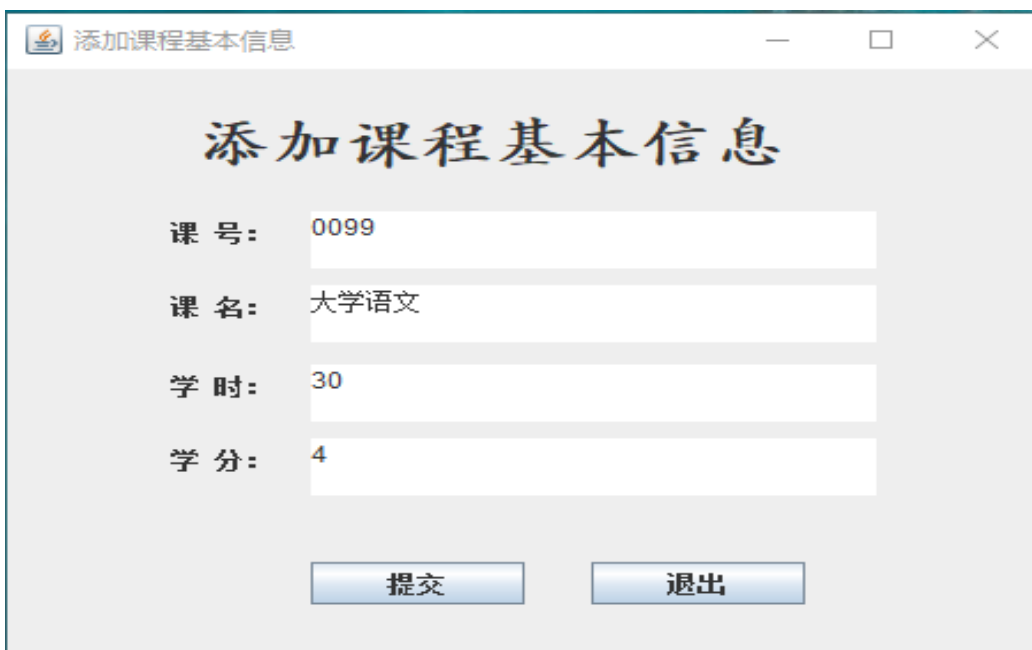


图 3.5. 课程管理流程图

该部分运行截图如图 3.6、图 3.7、图 3.8



添加课程基本信息

课号: 0099

课名: 大学语文

学时: 30

学分: 4

提交 退出

图 3.6. 添加课程截图



查询课程基本信息

课程号: 查询

0002	C++程序设计	20	4
0003	高等数学	35	5
0004	线性代数	30	4
0005	数据结构	30	4
0006	电子技术	20	3
0007	统计与概率	25	4
0008	数据库原理	30	4
0009	Java程序设计	30	4
0010	物理学	20	3
0011	毛泽东思想概论	30	4
0012	马克思主义原理	30	4
0013	计算机网络	30	4

保存 退出

图 3.7 查询课程截图



图 3.8 删除课程截图

3.2.1 界面控制及核心数据库操作代码

```
button_3.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Course course;
        for(int i=0;i<table.getRowCount();i++) {
            String cno = table.getValueAt(i, 0).toString();
            String cname = table.getValueAt(i, 1).toString();
            String ctime = table.getValueAt(i, 2).toString();
            String ccredit = table.getValueAt(i, 3).toString();
            course = new Course(cno, cname, ctime, ccredit);
            cours.add(course);
        }
        Connection conn = null;
        conn = ConnectionFactory.getInstance().makeConnection();
        try {
```

```

        conn.setAutoCommit(false);
    } catch (SQLException e2) {
        e2.printStackTrace();
    }
    for (Course cour:cours) {
        try {
            PreparedStatement ps = conn.prepareStatement("update
course\r\n" +
                "set Cname = ?, Ctime = ?, Ccredit = ?\r\n" +
                "where Cno = ?");
            ps.setString(4, cour.getCno());
            ps.setString(1, cour.getCname());
            ps.setString(2, cour.getCtime());
            ps.setString(3, cour.getCcredit());
            ps.execute();
            conn.commit();
        } catch (SQLException e1) {
            try {
                conn.rollback();
                JOptionPane.showMessageDialog(null, "请检查所输入的数据",
"更新错误");
                JOptionPane.ERROR_MESSAGE);
            } catch (SQLException e2) {
                e2.printStackTrace();
            }
            e1.printStackTrace();
        }
        JOptionPane.showMessageDialog(frame, "保存成功");
    }
}

```

4 系统测试

4.1 学生选课

学号为 1607084110 的学生之前没有选课，现在为他选择教师号为 10010 的教师教授的物理学和教师号为 0034 的教师教授的电路两门课程。执行的 sql 语句为：

```
insert into st(sno,tno) values( '1607084110' , '10010' );
```

```
insert into st(sno,tno) values( '1607084110' , '0034' );
```

执行完 sql 语句后，再查询 st 表，结果如图 4.1 所示：

```
MySQL 5.7 Command Line Client - Unicode
34 rows in set (0.00 sec)

mysql> select * from st where sno='1607084110';
+----+-----+-----+
| Sno | Tno  | Grade |
+----+-----+-----+
| 1607084110 | 0034 | NULL  |
| 1607084110 | 101010 | NULL  |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

图 4.1 系统测试图

可以看到记录已插入成功。插入完成运行界面如图 4.2 所示：

姓名: 陈橙 学号: 1607084110 系别: PH

[修改密码](#)

课程	代课教师	成绩
物理学	沙僧	0.0
电路	王小橙	0.0

[返回](#) [选课](#)

图 4.2 选课测试图

4.2 课程管理

查询出所有课程，并对课程号为 0023 做出修改。执行 SQL 语句为：

```
update course set Ctime= '30'
```

```
where Cno= '0023' ;
```

修改完成之后运行界面如图 4.3、图 4.5、图 4.5 所示：



查询课程基本信息

课程号:

0012	马克思主义原理	30	4
0013	计算机网络	30	4
0014	Linux编程	30	4
0015	密码学	25	3
0016	数值分析	35	4
0017	微机原理	30	4
0018	电路	25	3
0019	操作系统	35	4
0020	大学生创业基础	15	2
0021	近代史纲要	20	2
0022	大学生心理学	20	2
0023	形势与政策	20	2

图 4.3 修改课程测试图

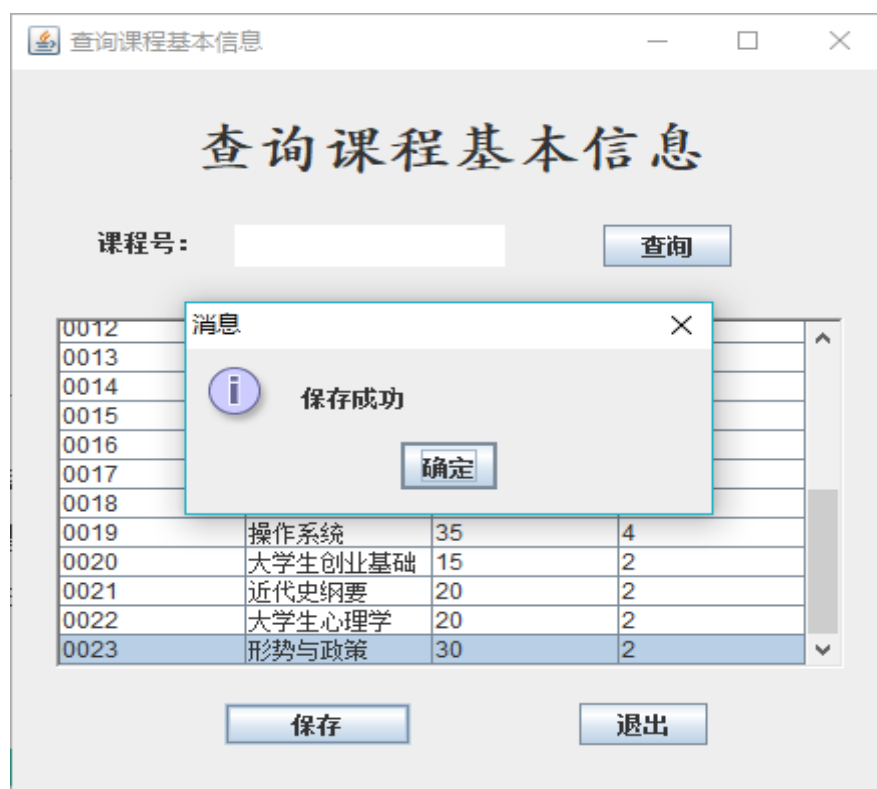


查询课程基本信息

课程号:

0012	马克思主义原理	30	4
0013	计算机网络	30	4
0014	Linux编程	30	4
0015	密码学	25	3
0016	数值分析	35	4
0017	微机原理	30	4
0018	电路	25	3
0019	操作系统	35	4
0020	大学生创业基础	15	2
0021	近代史纲要	20	2
0022	大学生心理学	20	2
0023	形势与政策	30	2

图 4.4 修改课程测试图



查询课程基本信息

课程号:

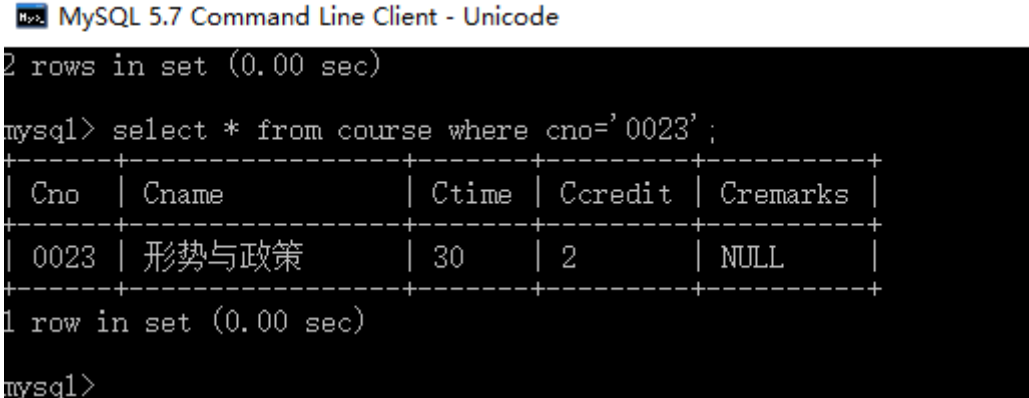
消息

保存成功

0012			
0013			
0014			
0015			
0016			
0017			
0018			
0019	操作系统	35	4
0020	大学生创业基础	15	2
0021	近代史纲要	20	2
0022	大学生心理学	20	2
0023	形势与政策	30	2

图 4.5 修改课程测试图

在数据库查询结果如图 4.6 所示:



```

MySQL 5.7 Command Line Client - Unicode
2 rows in set (0.00 sec)

mysql> select * from course where cno='0023';
+-----+-----+-----+-----+-----+
| Cno   | Cname      | Ctime | Ccredit | Cremarks |
+-----+-----+-----+-----+-----+
| 0023  | 形势与政策 | 30    | 2       | NULL     |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
  
```

图 4.6 修改课程结果测试图

5 个人体会

通过此次课程设计，加深了对 **Java** 语言和 **SQL** 语言的理解和掌握，初步掌握了设计开发一个软件的步骤。设计软件必须要遵守规范，严格按照数据库设计步骤执行。需求分析和概念结构设计必须精心设计，要考虑得全面，这两个步骤是基础，这两个步骤的设计决定了整个软件开发的基础与成败。此次课程设计虽然已经完成，但我们的软件还有一些漏洞，界面太简陋，不够友好，还未完善修补，功能太少，不能查询学生成绩平均分，高低分等。这也说明我们能力的不足，离设计一个完整的商业项目还有很大的能力差距，需要加强学习。

参 考 文 献

- [1] 王珊, 萨师煊. 数据库系统概论(第四版). 北京: 高等教育出版社, 2006: 10-56.
- [2] 耿祥义, 张跃平. Java2 实用教程 (第三版). 北京: 清华大学出版社, 2006: 31-59.
- [3] 李刚. 疯狂 Java 讲义. 北京: 人民邮电出版社, 2016: 156-278.
- [4] 姜中华, 刘小春. Java 数据库应用程序设计. 北京: 机械工业出版社, 2010: 48-49.
- [5] 郑阿奇. SQL Server 实用教程 (第 2 版) 北京: 电子工业出版社, 2004 :57-69
- [6] 郑阿奇. MySQL 数据库教程. 北京: 人民邮电出版社, 2009: 156-158
- [7] 姜桂红, 刘树淑, 孙勇. 数据库技术与应用. 北京: 科学技术出版社, 2010: 103-105
- [8] 李刚. Java 数据库技术详解. 北京: 化学工业出版社, 2014: 123-125
- [9] (美国)多纳休. Java 数据库编程宝典. 北京: 电子工业出版社, 2009: 154-156
- [10] 王国辉、吕海涛、李钟尉. Java 数据库系统开发案例精选. 北京: 人民邮电出版社, 2007: 98-103
- [11] (美)费舍儿 (Fisher, M), (美)埃利斯 (Ellis, J), (美)布鲁斯 (Bruce, J). JDBC API 教程与参考手册. 北京: 清华大学出版社, 2011: 89-100
- [12] 张晓东. Java 数据库高级教程. 北京: 清华大学出版社, 2010: 145-149