

中国民航大学  
计算机科学与技术学院

2017-2018 学年第二学期

《数据库系统课程设计》

题目： 学院成绩管理系统

班级： XXXXXXXX

学号： XXXXXXXXXX

姓名： 詹金浩

教师： 曹老师

成绩：

一、	需求分析 .....	2
1.	用户角度需求分析 .....	2
2.	界面功能角度分析 .....	2
二、	系统功能分析 .....	3
1.	注册 .....	3
2.	登录 .....	3
3.	修改密码 .....	4
4.	我的主页 .....	4
5.	账号注销 .....	4
6.	退出系统 .....	4
7.	学生主页 .....	5
8.	班级管理员主页 .....	5
9.	学院管理员主页 .....	6
三、	系统图解 .....	8
1.	系统流程图图解 .....	8
2.	核心功能图解 .....	8
四、	数据库设计 .....	9
1.	实体分析 .....	9
2.	E-R 图 .....	9
3.	表结构 .....	10
4.	视图 .....	10
5.	触发器 .....	10
6.	储存过程 .....	12
7.	游标 .....	13
8.	函数 .....	13
9.	索引 .....	14
10.	关系完整性 .....	14
五、	后台开发 .....	15
1.	增加 .....	15
2.	删除 .....	15
3.	修改 .....	15
4.	查找 .....	15
5.	调用函数 .....	16
6.	调用存储过程 .....	16
7.	事务控制 .....	16
六、	总结 .....	17
1.	课程设计成果 .....	17
2.	不足及改进方法 .....	17
3.	收获 .....	17

## 一、需求分析

学院管理系统中学生成绩管理是重要的一环，本系统旨在让学生、班级管理员、学院管理员可以从不同角度查看学生的成绩信息。同时学院管理员可以进行信息的录入。三种用户分别拥有相应的权限。

### 1. 用户角度需求分析

用户分为三种：普通学生、班级管理员和学院管理员。其中：

班级学生可以查看自己已修课程的成绩，并且通过第几学期、课程是否通过、课程类别来组合查询自己的课程情况。下载自己成绩单。

班级管理员可以查看班级的总体成绩和班级学生的基本成绩信息。下载班级成绩单。以及进入自己所管理班级学生主页的权限。

学院管理员可以通过 **Excel** 文件录入学生信息、课程信息、学生成绩信息，还可以录入学生项目信息和学生竞赛信息。可以查看各专业基本成绩信息，查看每班的基本成绩信息，查看每班各学科详细成绩。分专业下载成绩单。以及进入班级管理员主页和学生主页的权限。

### 2. 界面功能角度分析

- 登录界面：提供三个权限的用户登录。
- 注册界面：提供普通学生和班级管理员注册。
- 修改密码界面：提供普通学生和班级管理员修改密码。
- 学生界面：查看自己已修课程的成绩，通过第几学期、课程是否通过、课程类别来组合查询自己的课程情况。已修学分和剩余应修学分。下载成绩单。
- 班级管理员界面：查看班级基本信息，查看班级各个学生的基本成绩。下载成绩单。进入班级学生界面。
- 学院管理员界面：分专业查看专业基本信息，每个专业各班的基本成绩，查询班级各科成绩。通过 **Excel** 文件批量录入学生信息、课程信息、成绩信息。单个录入学生项目、竞赛情况信息。进入班级管理员界面。

## 二、系统功能分析

系统的功能都是围绕从不同角度查看学生成绩信息开发的。同时进行了不同用户权限的控制和数据插入数据库之前的校验。

权限由低至高一共分为三层：班级学生、班级管理员、学院管理员。高级权限用户可享有较低级权限用户的所有服务。

### 1. 注册

在注册界面选择类别进行注册。普通学生选择班级学生，输入账号密码注册。班级管理员选择班级管理员，输入账号密码注册。学院管理员有默认账号，不提供注册服务。

错误提示及原因：

- 1) 学号不存在：学院管理员未导入此学号相关信息。
- 2) 学号已注册过：此学号已被注册。
- 3) 格式错误：账号、密码不和要求。

### 2. 登录

在登录界面选择类别进行登录。普通学生选择班级学生，输入账号密码登录。班级管理员选择班级管理员，输入账号密码登录。学院管理员选择任一个输入账号密码登录。类别选择错误会登录失败。

错误提示及原因：

- 1) 账号或密码错误：账号密码不匹配或类别选择错误。

- 2) 学号未注册：此学号未注册过。
- 3) 格式错误：账号、密码不和要求。
- 4) 班级不存在：当前系统中未查找到需要注册的班级的信息。

### 3. 修改密码

修改密码的进入方式为在学生界面和班级管理员界面点击右上角导航栏后选择修改密码一项进入。学院管理员不提供修改密码服务。

错误提示及原因：

- 1) 旧密码错误：输入的旧密码和当前用户的密码不一致。
- 2) 格式错误：旧密码、新密码格式不和要求。

### 4. 我的主页

我的主页的进入方式为在自己主页点击右上角导航栏后选择我的主页一项进入。当前用户若跳转到其他页面时提供跳转至自己主页的服务。

### 5. 账号注销

账号注销的进入方式为在学生界面点击右上角导航栏后选择账号注销一项进入。用户一旦注销会删除他/她在本系统的所有信息。另外两类用户不提供注销服务。

### 6. 退出系统

退出系统的进入方式为在自己主页点击右上角导航栏后选择退出系统一项进入。用户一旦点击退出系统，会删除本次会话在服务器端保存的所有信息。页面会自动跳转至登录界面。

(3、4、5、6 图均见下图)



## 7. 学生主页

班级学生可从登录界面进入，班级管理员可以从班级管理员界面进入，学院管理员进入班级管理界面后再进入。

学生主页提供查看自己的基本成绩信息服务；由学期、类别、是否通过组合查询自己的成绩服务；已修学分、剩余应修学分服务；成绩单下载服务。



## 8. 班级管理员主页

班级管理员可从登录界面进入。学院管理员可从学院管理界面进入。

该主页提供查看本班级的基本成绩信息服务；查看每个同学的基本成绩信息服务；成绩单下载服务。点击学号可进入班级成员主页。

其中基本信息是整个班级的基本信息。各同学详细信息一项只能查看已注册过本系统的学生，无法查看到未注册的学生的详细成绩信息。

60x60 学院成绩管理系统 160341B

A : 班级信息

班级	160341B	班级人数	5
平均GPA	3.53	考试通过率	0.93
竞赛人次	0	项目人次	0

B : 班级成绩 下载成绩单

学号	姓名	绩点	已修学分	通过率
160341237	詹金浩	3.65	29.5	0.85
160341238	张家辉	2.48	19.5	1
160341239	袁振宇	4.23	19.5	1

Copyright © 2018 詹金浩 - 数据库课程设计

## 9. 学院管理员主页

学院管理员主页只能由学院管理员由登录界面进入。

该主页分成三个子页面：计科页面、信息页面、信息录入页面。其中：

- 1) 计科页面查看计科基本信息，计科各班级基本信息，计科某班级某课程详细信息。
- 2) 信安页面查看信安基本信息，信息各班级基本信息，信安某班级某课程详细信息。
- 3) 信息录入界面可以通过 **Excel** 文件批量录入学生信息、课程信息、成绩信息。可以逐个添加学生项目信息、学生竞赛信息。

说明：查看班级详细信息时只有存在选课记录的班级能被显示出来。无选课记录的班级不会显示。**Excel** 文件只能是 **xlsx** 文件，不能是 **xls** 文件。三种文件的格式要求必须和附录中提供的模板文件一致，且成绩文件中的学号和课程号录入必须先于成绩文件。

数据导入的提示及原因：

- 1) 文件数据已导入：整个文件中的数据全部正确导入数据库。
- 2) 文件不为空：未选择文件就点击上传按钮。
- 3) 文件格式不允许！：上传的文件不是 **xlsx** 文档。
- 4) 数据导入失败：以上两种原因的错误外，其他任何异常都会被归纳为“数据导入失败”。一般是由数据格式错误或者导入顺序不正确引起的问题。

(以上功能见下三图)

计科成绩

信安成绩

信息录入

A : 专业信息

专业	计算机科学与技术	GPA	3.58
挂科门次	2	挂科率	0.06
竞赛人次	0	项目人次	0

B : 各班成绩

下载成绩单

班级	GPA	挂科门次	通过率	竞赛人次	项目人次
170341B	4.80	0	1.0	0	0
160341B	3.53	2	0.93	0	0
160341C	3.40	0	1.0	0	0

科目成绩

班级: 170341B

科目: 物理上-2017001

查询

(专业基本信息)

计科成绩

信安成绩

A : 专业信息

专业		3.35
挂科门次		0.0
竞赛人次		0

B : 各班成绩

下载成绩单

班级	GPA	挂科门次	通过率	竞赛人次	项目人次
170342B	3.80	0	1.0	0	0
160342B	3.42	0	1.0	0	0
160342A	3.20	0	1.0	0	0

科目成绩

班级: 160342B

科目: 物理上-2017001

查询

160342B - 2017001

学号	姓名	成绩	总GPA
160342239	熊三	90	4.23
160342237	熊大	80	3.53
160342238	熊二	70	2.48

关闭

(班级课程详细成绩)

计科成绩

信安成绩

信息录入

A : 基本信息

学生信息录入(xlsx文件):

课程信息录入(xlsx文件):

学生成绩录入(xlsx文件):

B : 附加信息

类别: 竞赛

学号: 160341237

名称:

负责人: 是

奖项: 国一

时间: 2015

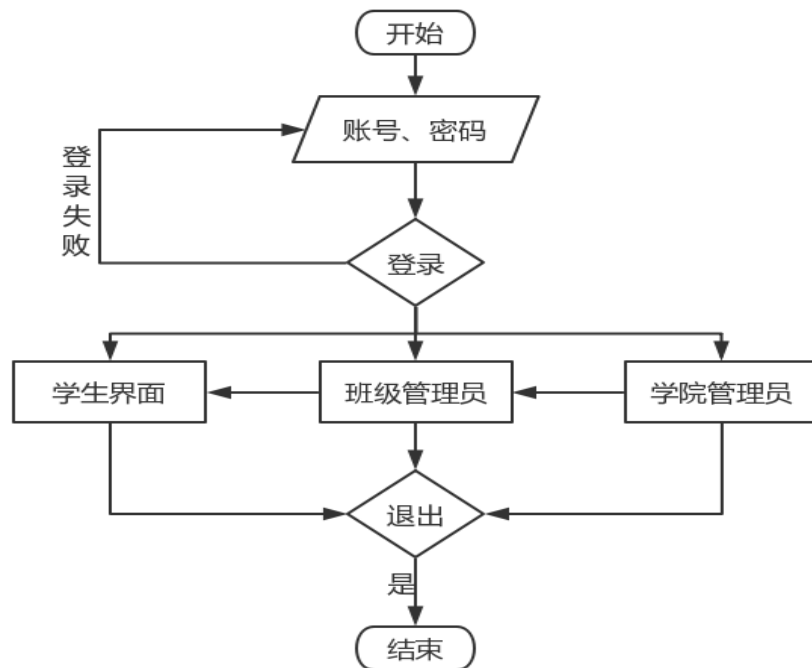
添加

(信息录入)

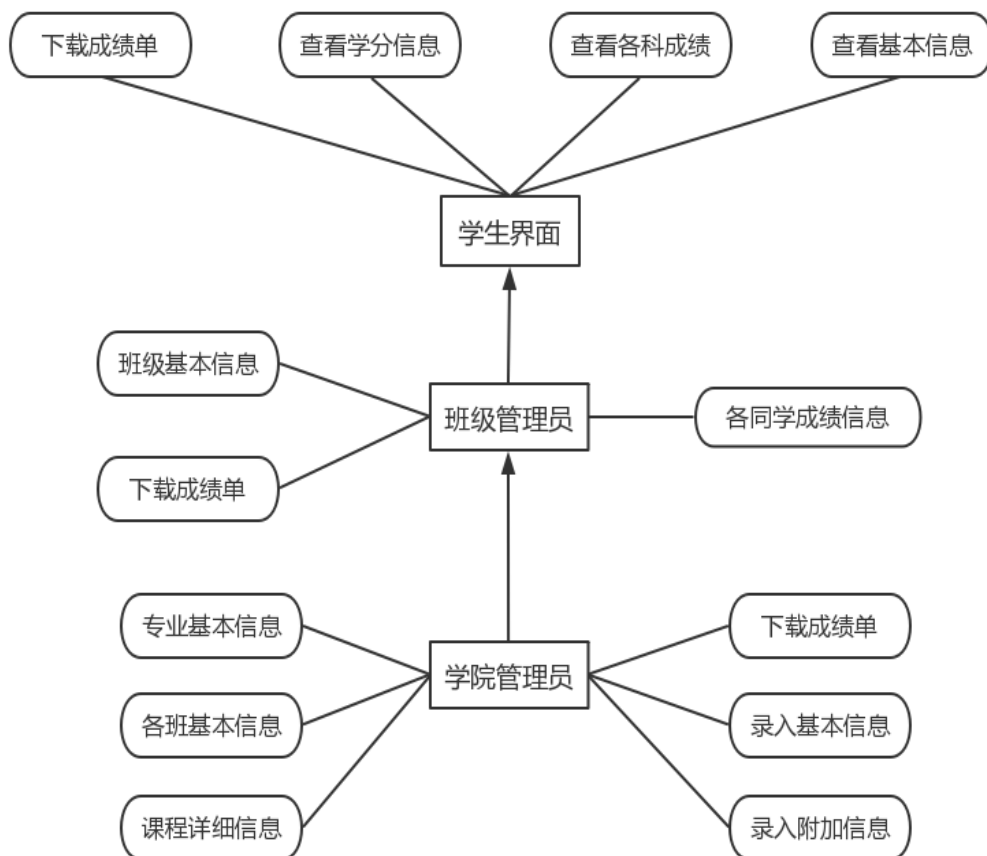


### 三、 系统图解

#### 1. 系统流程图图解



#### 2. 核心功能图解



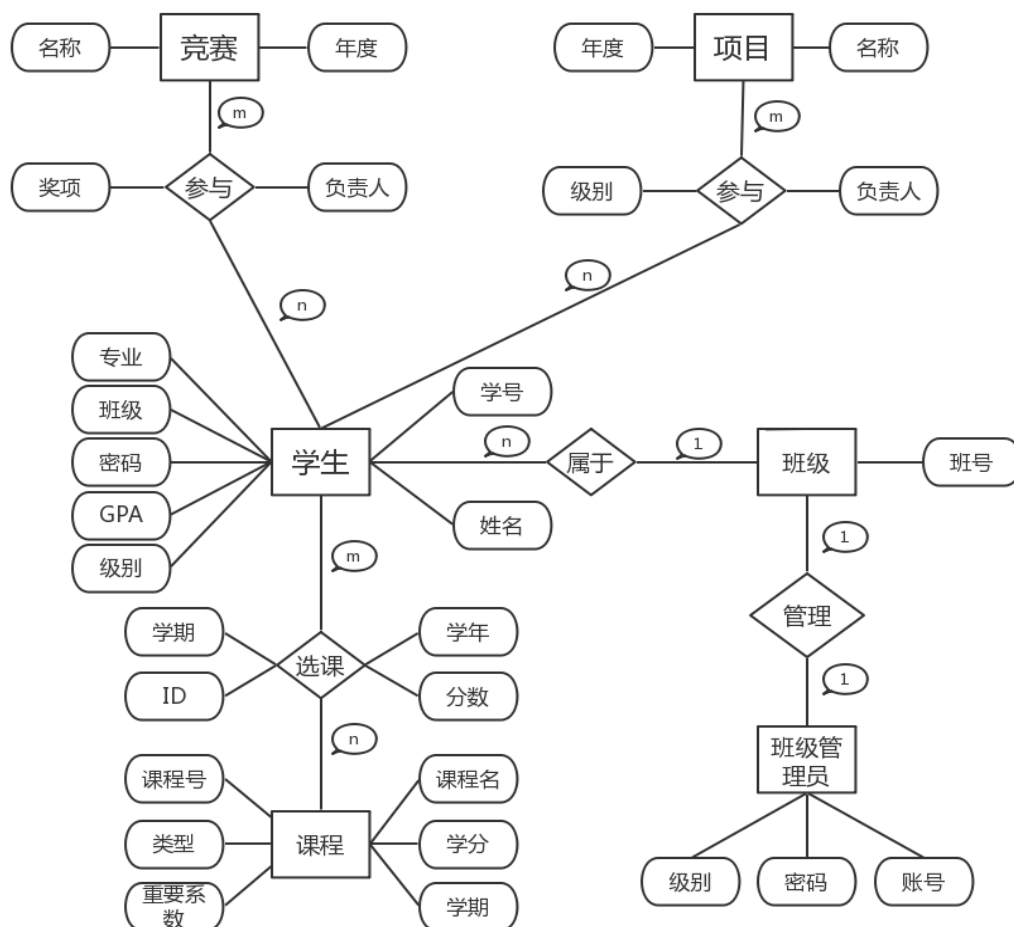
## 四、 数据库设计

### 1. 实体分析

- 1) 学生：学号、姓名、专业、班级、密码、GPA、级别；
- 2) 课程：课程号、课程名、类型、学分、重要系数；
- 3) 班级：班号；
- 4) 班级管理员：账号、密码、级别；
- 5) 竞赛：名称、年度；
- 6) 项目：名称、年度；

一个学生可以选多门课，一门课可以被多个学生选择，每次选课有成绩、选课号、学期、学年。一个班级有多个学生，每个学生只有一个班级。一个班级有一个管理员，一个管理员管理一个班级。一个项目可以有多个学生参与，一个学生可以参与多个项目，每次参与项目有级别、是否是负责人。一个竞赛可以被多个学生参与，一个学生可以参与多个竞赛，每次参与有奖项、是否是负责人。

### 2. E-R 图



### 3. 表结构

- 1) 学生：学号、密码、姓名、专业、班级、GPA、级别；
- 2) 课程：课程号、课程名、类型、重要系数、学分、年度；
- 3) 选课：选课号、学生学号、课程号、分数、年度、学期；
- 4) 班级管理员：账号、密码、班号、级别；
- 5) 竞赛和项目：ID、名称、年度、奖项/级别、是否是负责人、学号、类别；

### 4. 视图

系统中一共建了四个视图，以其中一个举例说明。班级管理员界面可以看见自己所管理班级学生的基本成绩信息来至于视图 `classstudentscores`。此视图是由学生表、课程表和选课表连接得到的。具体定义如下：

```
SELECT
    `scores`.`stu_id` AS `id`,
    `studentinfos`.`name` AS `name`,
    `studentinfos`.`pwd` AS `pwd`,
    `studentinfos`.`cclass` AS `cclass`,
    `studentinfos`.`gpa` AS `gpa`,
    sum(`courses`.`credit`) AS `allcre`,
    `getStudentPassRate` (`scores`.`stu_id`) AS `passrate`
FROM(
    (`studentinfos` JOIN `scores`)
    JOIN `courses`
)
WHERE (
    (
        `studentinfos`.`id` = `scores`.`stu_id`
    )
    AND (
        `scores`.`cou_id` = `courses`.`id`
    )
    AND (`scores`.`score` > 59)
)
GROUP BY
    `scores`.`stu_id`
```

### 5. 触发器

系统中使用到一次触发器。在学院管理员录入学生成绩时，触发器根据新录入的学生成绩的系统中已有的学生选课成绩计算出录入之后的学生 GPA。由于三种权限的用户都要使用 GPA，所以此触发器处在整个系统的核心位置。定义如下：

```
BEGIN
  DECLARE myavg DOUBLE;
  DECLARE allcre DOUBLE;
  DECLARE temp DOUBLE;
  DECLARE sum DOUBLE DEFAULT 0.0;
  DECLARE t INTEGER;
  DECLARE t1 DOUBLE;
  DECLARE t2 DOUBLE;
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT
    scores.score, courses.coefficient, courses.credit
  FROM studentinfos, scores, courses
  WHERE studentinfos.id = scores.stu_id AND scores.cou_id =
courses.id AND stu_id = new.stu_id;
  DECLARE CONTINUE HANDLER FOR NOT found SET done = TRUE;
  SELECT SUM(courses.credit) INTO allcre
  FROM studentinfos, scores, courses
  WHERE studentinfos.id = scores.stu_id AND scores.cou_id =
courses.id AND stu_id = new.stu_id AND score >= 60;
  OPEN cur;
  read_loop :
  LOOP
    FETCH cur INTO t, t1, t2;
    IF done THEN LEAVE read_loop; END IF;
    IF (t < 60) THEN
      SET temp = 0.0;
    END IF;
    IF (t >= 60) THEN
      SET temp = 1.0;
    END IF;
    IF (t >= 64) THEN
      SET temp = 1.6;
    END IF;
    IF (t >= 66) THEN
      SET temp = 1.7;
    END IF;
    IF (t >= 68) THEN
      SET temp = 2.0;
    END IF;
    IF (t >= 72) THEN
      SET temp = 2.3;
    END IF;
    IF (t >= 75) THEN
      SET temp = 2.7;
    END IF;
```

```

    IF (t >= 78) THEN
        SET temp = 3.0;
    END IF;
    IF (t >= 82) THEN
        SET temp = 3.3;
    END IF;
    IF (t >= 85) THEN
        SET temp = 3.7;
    END IF;
    IF (t >= 90) THEN
        SET temp = 4.0;
    END IF;
    SET sum = temp * t1 * t2 + sum;
END LOOP;
CLOSE cur;
SET myavg = sum / allcre;
UPDATE studentinfos SET gpa = myavg WHERE id = new.stu_id;
END

```

#### 6. 储存过程

系统使用到一次存储过程。在获得班级总体的 GPA 时，使用了带输入输出参数的储存过程。参数：IN `classID` varchar(20),OUT `avgpa` double。定义如下：

```

BEGIN
    DECLARE cgpa DOUBLE;
    DECLARE sum DOUBLE DEFAULT 0.0;
    DECLARE done int DEFAULT FALSE;
    DECLARE num int;
    DECLARE cur cursor for select gpa from
        studentinfos where cclass = classID;
    DECLARE CONTINUE HANDLER for not found set done = TRUE;
    set sum=0.0;
    set num=0;
    open cur;
    read_loop:LOOP
    FETCH CUR INTO cgpa;
    if done then leave read_loop; end if;
    set sum = sum + cgpa;
    set num = num+1;
    end loop;
    close cur;
    set avgpa = sum/num;
    SELECT sum,avgpa;
END

```

## 7. 游标

系统中使用到了两次游标及游标的循环。以触发器中的游标举例说明：

```
.....
DECLARE t INTEGER;
DECLARE t1 DOUBLE;
DECLARE t2 DOUBLE;
DECLARE done INT DEFAULT FALSE;
  DECLARE cur cursor for SELECT
    scores.score, courses.coefficient, courses.credit
    from studentinfos, scores, courses
      where studentinfos.id = scores.stu_id and scores.cou_id =
        courses.id and stu_id = new.stu_id;
DECLARE CONTINUE HANDLER for not found set done = TRUE;
SELECT SUM(courses.credit) into allcre
  from studentinfos, scores, courses
    where studentinfos.id = scores.stu_id and scores.cou_id =
courses.id and stu_id = new.stu_id and score>=60;
open cur;
  read_loop:loop
  FETCH cur into t, t1, t2;
.....
```

## 8. 函数

系统中共使用了 6 次函数。以获得班级挂科率函数举例说明：

参数：`classID` varchar(20);

返回类型：double

定义：

```
BEGIN
  DECLARE sum DOUBLE;
  DECLARE notPassSum DOUBLE;
  select COUNT(*) into sum from scores,studentinfos where
scores.stu_id = studentinfos.id and studentinfos.cclass = classID;
  select COUNT(*) into notPassSum from scores,studentinfos
    where scores.stu_id = studentinfos.id and
      studentinfos.cclass = classID and scores.score < 60;
  if(sum = 0) THEN
    return 0;
  end if;
  RETURN notPassSum/sum;
END
```

## 9. 索引

系统中除了 Navicat 自动添加的索引之外，使用到一次索引。在选课表中给成绩加上降序索引，因为后台获得的数据很多都是按成绩降序获得的。

定义：create INDEX scoredesc on scores(score DESC);

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
scores	0	PRIMARY	1	id	A	55	(Null)	(Null)		BTREE
scores	1	cou_id	1	cou_id	A	14	(Null)	(Null)		BTREE
scores	1	stu_id	1	stu_id	A	7	(Null)	(Null)		BTREE
scores	1	scoresdesc	1	score	A	27	(Null)	(Null)	YES	BTREE

## 10. 关系完整性

- 1) 实体完整性：主键均不为空且不重复；
- 2) 域的完整性：成绩信息中的成绩、年度、学期默认值为 NULL。
- 3) 参照完整性：成绩信息中的学号参照学生信息中的学号，成绩信息中的课程号参照课程信息中的课程号，项目竞赛信息中的学号参照学生信息中的学号。删除和更新采取的操作都是级联。

成绩表的完整定义如下：

```
CREATE TABLE `scores` (
  `id` INT (20) NOT NULL AUTO_INCREMENT,
  `stu_id` VARCHAR (20) NOT NULL,
  `cou_id` VARCHAR (20) NOT NULL,
  `score` INT (20) DEFAULT NULL,
  `year` INT (20) DEFAULT NULL,
  `term` INT (20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `cou_id` (`cou_id`),
  KEY `stu_id` (`stu_id`),
  KEY `scoresdesc` (`score`),
  CONSTRAINT `scores_ibfk_1` FOREIGN KEY (`cou_id`) REFERENCES
`courses` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `scores_ibfk_2` FOREIGN KEY (`stu_id`) REFERENCES
`studentinfos` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE = INNODB AUTO_INCREMENT = 443 DEFAULT CHARSET = utf8
```

## 五、 后台开发

数据库一端除了使用基本的增删查改，还使用到了视图、触发器、存储过程、游标、函数、索引等技术。但如索引等技术不能在后台代码中很好的体现出来。所以将能在后台中明显体现的技术列举如下。

例中的代码为 Dao 层的代码，即 sql 语句。事务的控制 Service 层中。

### 1. 增加

系统中多次使用到了增加，以导入学生信息时的增加为例说明：

```
public void insert(StudentInfos studentInfo) throws SQLException{
    String sql = "insert into studentinfos values(?,?,?,?,?,?,?)";
    queryRunner.update(sql, studentInfo.getId(), studentInfo.getPwd(),
        studentInfo.getName(), studentInfo.getMajor(), studentInfo.getCclass(),
        studentInfo.getGpa(), studentInfo.getLevel());
}
```

### 2. 删除

系统中多次使用到了删除，以学生注销时的级联删除为例：

```
public void signOffAccount(String id) throws SQLException{
    String sql1 = "delete from candp where stuid = ?";
    String sql2 = "delete from scores where stu_id = ?";
    String sql3 = "delete from studentinfos where id = ?";
    queryRunner.update(sql1, id);
    queryRunner.update(sql2, id);
    queryRunner.update(sql3, id);
}
```

### 3. 修改

系统中使用到的修改不多，以修改密码为例：

```
public void updatePwd(String id, String pwd) throws SQLException{
    String sql = "update studentinfos set pwd = ? where id = ?";
    queryRunner.update(sql, pwd, id);
}
```

### 4. 查找

系统中多次使用到了查找。以查询班级某课程详细信息为例：

```
public ArrayList<StudentCourseScoresBean> getCourseScores(String id) throws SQLException{
    String sql = "select name as name, type as type, score as score, coefficient as coefficient, score as gpa, "
        + "credit as credit from courses, scores where courses.id = scores.cou_id and stu_id = ?";
    ArrayList<StudentCourseScoresBean> query = (ArrayList<StudentCourseScoresBean>)
        queryRunner.query(sql, new BeanListHandler<StudentCourseScoresBean>(StudentCourseScoresBean.class), id);
    return query;
}
```



## 5. 调用函数

系统多次调用了函数，以获得班级挂科率为例：

```
public Double getClassNotPassRate(String id) throws SQLException{
    String sql = "select getClassNotPassRate(?) as rate";
    Double rate = (Double) queryRunner.query(sql, new ScalarHandler("rate"), id);
    return rate;
}
```

## 6. 调用存储过程

系统中使用到了一次存储过程，代码如下：

```
private static final String DRIVER_CLASS = "com.mysql.jdbc.Driver";
private static final String URL = "jdbc:mysql:///College-Achievement-Management-System";
private static final String USERNAME = "root";
private static final String PASSWORD = "3306";
public Double getClassGpa(String id) throws SQLException, ClassNotFoundException{
    Class.forName(DRIVER_CLASS);
    Connection connection = (Connection) DriverManager.getConnection(URL, USERNAME, PASSWORD);
    String sql = "{CALL getClassGpa(?,?)}";
    CallableStatement cstm = connection.prepareCall(sql);
    cstm.setString(1, id);
    cstm.registerOutParameter(2, Types.DOUBLE);
    cstm.execute();
    return cstm.getDouble(2);
}
```

## 7. 事务控制

系统中多次使用到了事务的控制，以增加成绩信息为例。

```
public void add(ScoresInfos si) {
    Session session = HibernateUtils.getCurrentSession();
    Transaction tx = session.beginTransaction();
    try {
        cid.add(si);
    } catch (SQLException e) {
        e.printStackTrace();
        tx.rollback();
    }
    tx.commit();
}
```

说明：Transaction 是一个事务控制的类。beginTransaction()是用来开启事务的方法。rollback()事务回滚的方法。commit()是事务提交的方法。

## 六、 总结

### 1. 课程设计成果

此次数据库原理课程设计没有选择实验指导书上的题目，而是自己想了一个题目，但题目灵感来自于实验指导书上的学校成绩管理系统。因为综合自己能力和时间考虑，做学校级别的难度太大，所以选择做了学院级别的成绩管理系统。

该系统以 **Java Web** 技术为基础，实现了学院成绩管理的基本功能。学院管理员可以进行信息的录入，专业信息的查询。班级管理员可以进行班级信息的查询。学生可以进行自己成绩的查询。而且高权限用户兼有低权限用户的功能。

除了功能的开发，对于非法访问也进行了处理，如低权限用户访问高权限用户页面等情况做了拦截处理。提高了系统的健壮性。

### 2. 不足及改进方法

系统的健壮性不够。尤其是在通过 **Excel** 文件导入数据的时候，对相应错误处理不够完善，用户只知道有问题，不知道到底哪里出了问题。没有使用 **Spring** 层框架，导致代码冗余情况较为严重。没有使用到高级查找如外链接的技术。

改进方法为不通过 **Excel** 文件导入数据，通过 **csv** 文件导入数据更方便快捷，而且容错性较好。使用 **Spring** 框架减小代码冗余性。增加系统功能以使用到外链接的功能。

### 3. 收获

通过本次数据库系统课程设计掌握了基础的以 **Java** 为核心的网站开发技术。掌握了数据库的分析设计。并且综合应用数据库系统原理课上讲的增删查改、视图、存储过程、游标、触发器、函数、关系完整性等知识到数据库系统课程设计中。提升了自己的专业竞争力。