**成绩管理系统**

**一、需求分析阶段**

**1．1.信息需求**

学院对每一门课程的成绩实施透明、严格管理，并要求在每次考试结束后提交详细的考试分析报告。现有的考试成绩管理完全采用纸质的方式手工实现，容易出错且造成一定的资源浪费。为了提高考试成绩管理的效率，减少手工实现出错的机率，需要实现一个综合的成绩管理系统。

**1.2.功能需求**

成绩管理系统的具体要求如下：

* 以教师为单位配备系统：要求为每一个代课教师均配备一个成绩管理系统，需要同时管理本科生与研究生的课程成绩。
* 与成绩管理系统相关的教师信息包括工资号、教师姓名和职称（助教、讲师、副教授、教授）。
* 与成绩管理系统相关的学生信息包括学号、姓名、性别。
* 教师每年可能会带多门课程，也可能一门课程都没有带。
* 学生可能会选修多门课程。对于必修课和限选课，必须进行选择；对于任选课程，按照学分要求自行选择。（成绩管理系统不考虑选课过程，即认为课程均已确定）
* 每个学生均属于一个班级，具有一个专业方向，课程可能以专业方向为单位进行开设，也可能以班级为单位进行开设。
* 每门课程包括课程编号（学院对每一门课程的唯一编号，例如本科生课程《数据库系统概论》的编号为SE3121017，研究生课程与本科生课程编号不同，例如：研究生课程《软件工程》的课程编号为1051030）、课程名称、课程类型（必修、限选和任选）、学时（X+Y的形式，其中X和Y为数字。例如《数据库系统概论》的学时为38+16）、学分、课程性质（本科生、研究生）。

每门课程每学年都要进行具体实施。在课程教学大纲的指导下，每年的课程教授内容可能略微不同，每门课程的内容由教学大纲的章节确定。除此以外，学院还要求详细记录每次课程实施的内容。

**1.3. 安全性与完整性要求**

（1）系统应该设置访问用户的标志以鉴别是否是合法用户，要求合法用户设置密码，保证用户身份的安全性。

（2）系统对不同用户设置不同权限，区分不同用户，如学生，教师。

（3）各种信息记录的完整性，不能为空。

（4）各种数据相互联系的正确性。

**二、概念结构设计阶段**

概念结构设计阶段是整个数据库设计的关键，它通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型。

设计学生成绩管理数据库包括学生、成绩、课程、教师、大纲五个关系。

## 2.1 确定联系集及E-R图

根据前面对系统进行的分析，已经初步了解了学生成绩管理系统的数据处理流程，找出与系统有关的各个实体及其相互联系如下：

①、标示实体集：学生、成绩、课程、教师、大纲五个关系。

②、标示联系集：

老师和课程：一个老师可以教授多门课程，一门课程只能给一个老师教学；

存在“授课”的关系：N：1

学生和课程：一个学生可以选修多门课程，一门课程提供给多个学生选修；

存在“选课”的关系：N : M

关系grade的主键是{student\_id，course\_id，grade\_id}显然同一个学生在同一个学期不允许修读同一门课程多次

关系outline的outline\_id属性参照本关系的主键course\_id和teacher\_id属性。这里假设一门课程最多只需要安排一位老师。

③、标示属性集：

学生（学号，姓名，性别）

老师（老师工号，老师姓名，老师职位）

大纲（大纲编号，大纲内容，大纲方式，老师工号，课程号）

课程（课程号，课程名，学分，课时数，教学性质，教学方式）

成绩（成绩编号，学号，课程号，平时成绩，大作业成绩，作业成绩，成绩）

## 2.2 画出E-R图

学生信息图如下：



图2-2-1学生实体集的E-R图

教师信息图如下：



图2-2-2老师实体集的E-R图

大纲信息图如下：



图2-2-3教学大纲实体集的E-R图

课程信息图如下：



图2-2-4课程实体集的E-R图

成绩信息图如下：



图2-2-5成绩实体集的E-R图

## 2.3成绩管理系统总E-R图

根据上面对实体联系的分析，可以画出E-R图如下:



图2-2-6成绩管理图实体集的E-R图

# 三、逻辑数据库设计阶段

## 3.1 学生信息表(student)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 描述 | 数据类型 | 字段限制 |
| student\_id | 学号 | char(10) | Primarykey |
| student\_name | 学生姓名 | varchar(10) | Not null |
| sex | 性别 | char(2) | Not null |

## 3.2 教师信息表(teacher)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 描述 | 数据类型 | 字段权限 |
| teacher\_id | 教师工号 | char(10) | Primarykey |
| teacher\_name | 教师姓名 | varchar(10) | Not null |
| teacher\_working | 教师工作 | varchar(10) | Not null |

## 3.3 大纲信息表（outline）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属 性 | 数据类型 | 字段权限 |
| outline\_id | 大纲编号 | char(10) | Primarykey |
| outline\_content | 大纲内容 | varchar(30) | Not null |
| outline\_style | 大纲形式 | char(10) | Not null |
| teacher\_id | 教师工号 | char(10) | Not null |
| course\_id | 课程编号 | char(10) | Not null |

## 3.4 课程表（course）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属 性 | 数据类型 | 字段权限 |
| Course\_id | 学号 | char(10) | Primarykey |
| Course\_name | 课程号 | char(10) | Not null |
| Course\_type | 开课学期 | char(10) | Not null |
| Course\_time | 成绩 | int | Not null |
| Course\_grade | 课程学分 | int | Not null |
| Course\_nature | 课程性质 | varchar(10) | Not null |

## 3.5 成绩表（grade）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属 性 | 数据类型 | 字段权限 |
| Grade\_id | 成绩编号 | char(10) | Primarykey |
| Grade\_working | 大作业成绩 | int | Not null |
| Grade\_final | 最终成绩 | int | Not null |
| Grade\_usual | 平时成绩 | int | Not null |
| Grade\_participation | 作业成绩 | int | Not null |
| Student\_id | 学号 | char(10) | Not null |
| Course\_id | 课程编号 | char(10) | Not null |

# 四、建表

## 4.1 创建数据库

打开SQL工具“查询分析器”

在查询窗口中键入下列SQL语句

create database student\_grade

执行上述SQL语句即可新建一名为student\_grade的数据库。

## 4.2 创建数据表

一个完整的数据库不可缺少的就是数据表，若干个数据表的集合成一个数据库。数据表主要用来存放一定格式的记录，数据表中的行被称为记录，列被称为字段。创建数据表的过程其实就是定义字段的过程。

在此学生成绩管理系统中需要创建五个表，即学生信息表、老师信息表、大纲信息表、课程信息表和成绩表。

创建数据库后，为student\_grade数据库添加数据表，步骤如下。

新建查询窗口

在查询窗口中键入下列SQL语句

**create** **table** teacher( #创建教师表

teacher\_id **char**(10) **primary** **key**, #教师工号

teacher\_name **varchar**(10) **not** **null**, #教师姓名

teacher\_work **varchar**(10) **not** **null** #教师职称

);

执行上述SQL语句即可创建教师信息表相关表格

**create** **table** student( #创建学生表

student\_id **char**(10) **primary** **key**, #学号

student\_name **varchar**(10) **not** **null**, #学生姓名

student\_sex **char**(2) **not** **null** #学生性别

);

执行上述SQL语句即可创建学生信息表相关表格

**create** **table** course( #创建课程表

course\_id **char**(10) **primary** **key**, #课程编号

course\_name **varchar**(10) **not** **null**, #课程名称

course\_type **varchar**(10) **not** **null**, #课程形式

course\_time **int** **not** **null**, #课程学时

course\_grade **int** **not** **null**, #课程成绩

course\_nature **varchar**(10) **not** **null** #课程性质

);

执行上述SQL语句即可创建课程信息表相关表格

**create** **table** outline( #创建大纲表

outline\_id **char**(10) **primary** **key**, #大纲序号

outline\_content **varchar**(20) **not** **null**, #大纲内容

outline\_style **char**(10) **not** **null**, #大纲类型

teacher\_id **char**(10) **not** **null**, #教师工号

course\_id **char**(10) **not** **null** #课程编号

**foreign** **key**(teacher\_id) **references** teacher(teacher\_id),

**foreign** **key**(course\_id) **references** course(course\_id)

);

执行上述SQL语句即可创建大纲表相关表格

**create** **table** grade( #创建成绩表

grade\_id **int** **not** **null**, #成绩表编号

grade\_working **int** **not** **null**, #大作业成绩

grade\_final **int** **not** **null**, #最终成绩

grade\_ususl **int** **not** **null**, #平时成绩

grade\_participation **int** **not** **null**, #作业成绩

student\_id **char**(10) **not** **null**, #学号

course\_id **char**(10) **not** **null**, #课程编号

**primary** **key**(grade\_id),

**foreign** **key**(student\_id) **references** student(student\_id),

**foreign** **key**(course\_id) **references** course(course\_id)

);

执行上述SQL语句即可创建成绩表相关表格

# 五、数据库的运行和维护

## 5.1 定义

基本表的创建、修改及删除；索引的创建和删除。

### 5.1.1 基本表的创建，建表语句

**create** **table** student( #创建学生表

student\_id **char**(10) **primary** **key**, #学号

student\_name **varchar**(10) **not** **null**, #学生姓名

student\_sex **char**(2) **not** **null** #学生性别

);

### 5.1.2 基本表的删除

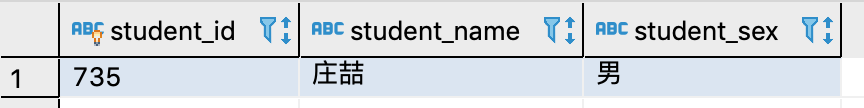
**drop** **table** **if** **exists** grade;

## 5.2 数据操作

### 5.2.1 单表查询：

**select** \* **from** student; #查询学生表的信息

可以查找学生表中所有学生的信息：



### 5.2.2 连接查询

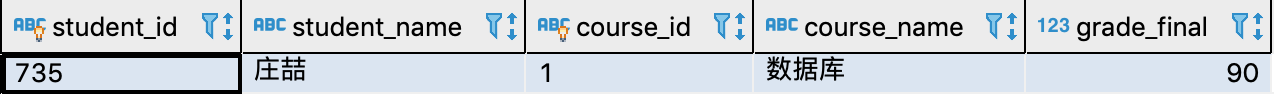
查询学生的学号、姓名，所选课程的课程号、课程名和成绩：

这个具体的操作需要联系3个表的信息（学生表，成绩表，课程表）：

**select** student.student\_id,student\_name,course.course\_id,course\_name,grade\_final

**from** student,grade,course

**where** student.student\_id='735' **and** course.course\_id='1';



### 5.2.3 嵌套查询

查询选修了“数据库”这门课的所有学生的学号、姓名、性别。

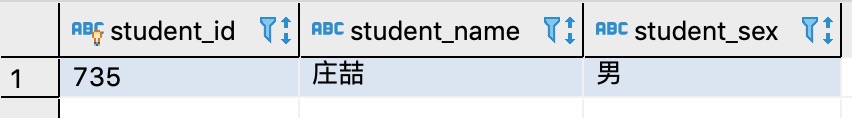
**select** student.student\_id,student\_name,student\_sex

**from** student **where** student\_id **in**(

**select** student\_id **from** grade

**where** course\_id='1'

);

d

## 5.3 数据库更新操作

### 5.3.1 插入数据

向Student表中添加一项记录：

姓名：小林、学号：736、性别：女:

**insert** **into** student **values**('736','小林','女');

## 5.4 为数据库建立索引

### 5.4.1 索引的建立

**create** **index** studentNameSM **on** student(student\_name);