**数据库设计**

**综合实训报告**

****

**课题 学生成绩管理系统**

**班级 计类20104班**

**学号 45420323**

**姓名 陈毅**

**指导教师 常会丽**

**2021 年 11月 25 日—2021 年 12月 31 日**

目录

[1.绪论 5](#_Toc16293)

[1.1项目背景 5](#_Toc16728)

[1.2设计目的 5](#_Toc10414)

[2.需求分析 6](#_Toc23953)

[2.1 信息需求 6](#_Toc11687)

[2.2 功能需求 6](#_Toc8915)

[2.3 安全性与完整性要求 6](#_Toc15862)

[2.4 内容需求 6](#_Toc15305)

[2.5系统设计 7](#_Toc22279)

[2.6.1六大功能管理模块划分 7](#_Toc32731)

[2.6 数据流图 8](#_Toc11175)

[2.7 数据字典 11](#_Toc13206)

[3.概念结构设计 15](#_Toc16639)

[3.1 局部E-R图 16](#_Toc26736)

[3.2 全局E-R图 19](#_Toc6899)

[4.逻辑结构设计 19](#_Toc5402)

[4.1将E-R图转化为关系模式 19](#_Toc3333)

[4.2将E-R图转化成表 20](#_Toc24532)

[5.物理结构设计 22](#_Toc11938)

[5.1建立视图 22](#_Toc18353)

[5.2建立索引 23](#_Toc31905)

[5.3建立触发器 24](#_Toc5267)

[6.数据库实施 26](#_Toc187)

[6.1 MySQL创建数据库和数据表 26](#_Toc29664)

[6.2 MySQL建立视图 28](#_Toc19040)

[6.3 MySQL建立索引 28](#_Toc24974)

[6.4 MySQL建立触发器 29](#_Toc12638)

[7. 实训设计总结 30](#_Toc24816)

# 1.绪论

## 1.1项目背景

现今使用计算机对学生信息进行管理，拥有手工管理所无法比拟的优点。例如：检索迅速、查找方便、可靠性高、存储量大、成本低等。这些优点能够极大地提高学生信息的效率，也是管理科学化、正规化的重要支撑。  
 学生成绩管理系统能够方便地查询和变更学生成绩的基本数据（包括信息数据和成绩数据），节省大量工作时间，有效地提高学生成绩管理的效率。而查询信息的直观显示将有助于系统的用户一目了然地掌握学生成绩的情况，以便于用户对成绩可行性操作。

## 1.2设计目的

在日常生活中，学生成绩信息的管理是每个老师都不可或缺的，成绩的录入、修改、导出以及计算综合分和平均分等，这些教学任务附带的工作往往给教师带来很大的负担，为了减轻教师上课负担，方便学生和老师的课堂教学，对当前学生成绩信息相关内容设计专门的软件进行管理，减轻教师管理学生成绩的工作量，在日常教学中具有积极的实践意义。

一直以来，学校对学生成绩管理的要求也越来越高、因此学校对学生信息的管理也更系统化、数字化。为了提高学生的实际应用数据库的能力。使学生锻炼简单的数据库开发能力、因此决定设计一个简单的学生成绩管理系统、包括学生信息、课程信息、以及班级信息和任课教师信息。能够存储相应学生评教信息。和课程安排提供的学生成绩管理各种功能分为面问学生。教师和教各员三部分，其中教务员可以录入学生，教师、班级、课程信息，每个专业、每个班级需要学习哪些课程，指定课程的任课教师。给每个班的每门课程安排指定的认课老师。教师可以查看学习该课程的学生名单。课程结束后。教师可以录入课程成绩。课程分两类。必修课和选修课。学生可以选修课程、每学期几门。学生可以查看自己各门课程的成绩、进行评教。给老师打分。该学生成绩管理系统类似于简单的学校教务系统、能对学校日常事务进行简单的管理。

# 2.需求分析

## 2.1 信息需求

对学校而言，学生成绩管理是管理工作中重要的一环，但是高校学生的成绩管理工作量大、繁杂，人工处理非常困难。因此，借助于强大计算机的处理能力，能够把人从繁重的成绩管理工作中解脱出来，并且更加准确、安全、清晰的管理环境。

## 2.2 功能需求

能够进行数据库的数据定义、数据操纵、数据控制等处理功能。具体功能应包括：可提供课程安排、课程成绩数据的添加、插入、删除、更新、查询，学生及教职工基本信息查询的功能。

## 2.3 安全性与完整性要求

对于学生成绩管理系统数据库来讲，由于其主要数据是学生成绩，只能由本人以及所教老师及教务处知道，因此做好数据安全性是重中之重。另外，要求所有在校学生的信息都要录入其中，并且要设计好个别情况。

## 2.4 内容需求

学生信息管理系统属于应用系统。学生信息管理系统的使用者是教务处管理员和学生。它的功能就是教师通过输入用户名和口令登陆到教学管理数据库中，从而实现对各门选课人数限制，对学生信息的维护限制。该系统是一个集成多功能的信息管理系统，有教师资料管理、学生管理、班级管理、课程管理和教务功能模块。本系统将大大提高学生信息的管理效率，带来数据共享、网络查询和网络管理等便利，实现学生成绩管理的数字化和智能化。

## 2.5系统设计

### 2.6.1六大功能管理模块划分

教师管理模块：本模块主要对教师的资料信息进行查询，删除，修改和添加。  
 学生管理模块：本模块主要对学生资料信息进行相应的查询，删除，修改和添加。添加了学生所在院系资料与学生所在专业资料，同时可对院系资料与专业资料进行相应的添加，修改，查询。  
 课程管理模块：本模块主要对课程资料进行管理，对其进行查询，添加，修改和删除。又详细划分了课程类别资料，对其进行如上操作。  
 成绩管理模块：本模块对学生成绩进行查询操作，如发现成绩有输入错误，可对其进行相应的修改，还可对学生的成绩进行录入  
 班级管理模块：本模块班级资料信息进行管理，对其进行相应的查询，修改，删除和添加。

教务人员管理模块：本模块对学生、班级、教师、课程进行管理，对其进行相应的查询，修改，删除和添加。

根据实际问题，本系统要实现以下功能：  
（1）用户认证。本系统用户有教务处管理员和学生，所以用户必须通过认证才能登入系统。系统能够自动识别用户类型，能够给不同的用户分配不同的权限。教务处管理员用户成功登入后可以进入操作界面，学生成功登入后即可进入学生本人信息界面。  
（2）系统更新与维护。可以对系统和各项管理模块等进行维护。  
（3）信息管理。允许教务处管理员修改学生信息，允许学生在任何时候登入查询本人信息。

（4）信息查询、录入和修改。针对学生、教师、班级、课程、教务人员进行信息的增删改查。

功能结构层次图如图1所示：



图1 功能结构层次图

## 2.6 数据流图



图2 顶层数据流图



图3 一层数据流图

## 2.7 数据字典

数据项

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 别名 | 数据类型 | 取值范围 | 取值含义 |
| Sno | 唯一标识每一位学生 | 学生学号 | Char(8) | 0~999999 | 前两位表示学校，第三、四位表学院，第五六位表年级，最后四位学生序号 |
| sname | 标识姓名 | 学生姓名 | Varchar(10) |  |  |
| sex | 标识性别 | 性别 | Enum(‘男’,’女’) | 男，女 |  |
| sage | 标识年龄 | 年龄 | Int | 0-100 |  |
| saccount | 标识每一个学生的账户 | 学生账户 | Char(20) |  |  |
| spassword | 标识每一个学生的密码 | 学生密码 | Varchar(50) | 有数字、字母、特殊、符号 |  |
| nativeplace | 标识每一个学生的所在地 | 学生籍贯 | Varchar(20) |  |  |
| iphone | 标识电话号码 | 电话号码 | Varchar(50) |  |  |
| sdept | 标识学生专业 | 专业 | Char(10) |  |  |
| sxueyuan | 标识学生学院 | 学院 | Char(20) |  |  |
| classat | 标识查询课程所在的班级 | 教学班号 | Char（10） |  |  |
| Rnum | 标识每个班的班级人数 | 人数 | Int（4) |  |  |
| identity card | 唯一标识每一位学生 | 身份证号 | Char(18) |  |  |
| class | 标识每一个班级 | 班级 | Char(4) |  |  |
| Classname | 标识每个班级的名称 | 班级名称 | Char(4) |  |  |
| cno | 唯一标识每一节课程 | 课程编号 | Char(20) |  |  |
| cname | 标识课程名 | 课程名称 | Char(20) |  |  |
| credit | 标识学生所修的学分 | 学分 | Decimal(3,1) | 1.0-5.0 |  |
| ctime | 标识学生所修的学时 | 学时 | Int(4) |  |  |
| Optional cource | 标识学生的选修课 | 选修课 | Char(20) |  |  |
| score | 标识每门课程学生的考试成绩 | 成绩 | Tinyint(2) | 0-100 |  |
| tno | 唯一标识每一位老师 | 教师工号 | Char(20) |  | 前两位表学校，第三、四位表学院，第五六位表年 |
| tname | 标识每一位老师的姓名 | 教师姓名 | Varchar(10) |  |  |
| taccount | 标识每一位老师的账户 | 教师账户 | Char(20) |  |  |
| tpassword | 标识每一位老师的登录密码 | 教师密码 | Varchar(50) |  |  |
| Scyear | 标识每一学年 | 学年 | date |  |  |
| scxueqi | 标识每一学期 | 学期 | Int（2） | 0-8 |  |
| ttno | 唯一标识教务处教师的工号 | 教务处工号，教务处用户名 | Char(10) | 0~9999999999 | 前两位表学校，第三、四位表学院，第五六位表年级，最后四位教务处人员排号 |
| ttname | 标识教务处教师的姓名 | 教务处教师姓名 | Varchar(10) |  |  |
| ttpassword | 唯一标识教务处教师登录的密码 | 教务处密码 | Varchar(50) | 含有数字字母特殊符号 |  |

数据结构

|  |  |  |
| --- | --- | --- |
| 数据结构名 | 含义说明 | 组成 |
| 学生 | 是学生信息管理子系统的主体数据结构，定义了每一个学生的有关信息 | 学号，姓名，性别，账户，籍贯，身份证号,密码 |
| 教师 | 是教师信息管理子系统的主体数据结构，定义了每一个教师的有关信息 | 工号，姓名，性别，账户 |
| 班级 | 是班级信息管理子系统的主体数据，定义班级的有关信息 | 班级编号，人数，班级名称，院系，年级 |
| 课程 | 是课程管理信息子系统的主体数据，定义课程的有关信息 | 课程号，课程名，学时,学分，学期 |
| 教务处 | 是教务处人员信息的主体系统，定义了一个有关于教务处人员的信息 | 教务处工号、教务处教师姓名、教务处密码 |

数据流

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据流名 | 说明 | 数据流来源 | 数据流去向 | 组成 |
| 课程导出 | 课程信息 | 教务人员导入 | 课程信息表存储 | 课程号，课程名，学时，前置课，学分，学期，成绩 |
| 账户信息 | 进行账户登录 | 输入的工号和密码 | 流入登录系统中 | 用户名，密码 |
| 信息导出 | 从信息表中调取用户名，密码信息 | 信息表 | 流入登陆系统中 | 用户名，密码 |
| 学生信息表 | 学生的基本信息 | 教务人员导入 | 学生信息表存储 | 学号，姓名，年龄，性别，账户，密码，身份证号 |
| 教师信息表 | 教师的基本信息 | 教务人员导入 | 教师信息表存储 | 工号，姓名，性别，账户 |
| 班级信息表 | 班级的基本信息 | 教务人员导入 | 班级信息表存储 | 人数，班级名称，院系，班级编号，年级 |
| 成绩录入 | 把成绩信息录入进去 | 录入成绩 | 成绩信息表 | 学生学号、课程号、成绩 |
| 课程录入 | 把课程信息录入进去 | 录入课程 | 课程信息表 | 课程号、课程名、学分、学时 |
| 学生成绩修改 | 把学生成绩进行修改 | 修改成绩 | 成绩信息表 | 学生学号、课程号、成绩、学期 |
| 教师信息修改 | 把教师信息进行修改 | 修改教师信息 | 教师信息表 | 教师工号、教师姓名、教师密码 |
| 班级信息录入 | 把班级信息录入进去 | 录入班级 | 班级信息表 | 教学班号、年级、人数 |

数据存储

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据存储名 | 说明 | 流入的数据流 | 流出的数据流 | 组成 |
| 学生信息表 | 存储学生信息 | 学生信息录入 | 学生登录 | 学号，姓名，年龄，性别，账户，密码，身份证号 |
| 教师信息表 | 存储教师信息 | 教师信息录入 | 教师登录 | 工号，姓名，性别，账户 |
| 班级信息表 | 存储班级信息 | 班级信息录入 | 查找班级 | 人数，班级名称，院系，班级编号，年级 |
| 成绩信息表 | 存储成绩信息 | 成绩信息录入 | 查询成绩 | 学生学号、课程号、成绩、学期 |
| 课程信息表 | 存储课程信息 | 课程信息录入 | 查找课程 | 课程号，课程名，学时，学分，学期 |

数据处理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 处理过程名 | 说明 | 输入数据流 | 输出数据流 | 处理 |
| 录入成绩 | 将所选择的班级，进行成绩录入 | 所选班级的信息 | 该班级里该学生在这门课程的成绩 | 将所选班级，进行成绩录入 |
| 登录 | 用户名密码校验的过程 | 用户名，密码 | 个人的信息 | 将输入的用户名和密码与信息表中的用户名密码，进行校验 |
| 查询成绩并导出 | 进行成绩的查取 | 学生的信息 | 成绩的信息 | 通过输入的学生信息导出成绩表里关于该学生的成绩 |
| 查询课程并导出 | 进行课程的查取 | 老师的信息 | 课程的信息 | 通过输入的老师信息导出课程表里关于该老师的课程 |
| 学生信息导入或导出 | 学生信息存入表或导出表 | 学生信息表 | 学生信息表 | 记录学生学号，姓名，年龄，性别，账户，密码，身份证号 |
| 班级信息导入或导出 | 班级信息存入表或导出表 | 班级信息表 | 班级信息表 | 记录人数，成绩，教学班， |
| 课程信息导入或导出 | 课程信息存入表或导出表 | 课程信息表 | 课程信息表 | 记录学分，学时 |

# 3.概念结构设计

概念结构设计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。

根据学生成绩信息管理数据库设计需求抽象出学生、教师、课程、班级，教务人员五个实体，对五个实体做简化处理，默认一门课程仅被一位老师讲授。因简化后关系结构比较简单，对5个实体之间的关系进行分析如下：

一位学生会被多位老师教导，一位老师会教导多位学生，所有学生与教师之间是多对多（m:n）的关系；

一位学生可能会选修多门课程，一门课程会被多位学生选修，所以学生与课程之间是多对多（m:n）的关系；

一位教师会教授多个班级，一个班级会被多个教师教导，所以教师与班级是多对多（m:n）的关系；

一位教师会讲授多门课程，一门课程会被一位教师讲授，所以教师与课程的关系是一对多（1:n）的关系；

一门课程拥有多项成绩，一项成绩仅被一门课程拥有，所以课程与成绩的关系是一对多（1:n）的关系；

## 3.1 局部E-R图



图4 学生实体



图五 教师实体



图6 班级实体



图7 教务人员实体



图8 课程实体

## 3.2 全局E-R图



全局E-R图

# 4.逻辑结构设计

逻辑结构是独立于任何一种数据模型的信息结构。逻辑结构的任务是把概念结构设计阶段设计好的基本E-R图转化为宜选用的DBMS所支持的数据模型相符合的逻辑结构，并对其进行优化。

## 4.1将E-R图转化为关系模式

E-R图向关系模型转化要解决的问题是如何将实体型和实体间的联系转化为关系模式，如何确定这些关系模式的属性和码。

(1)抽象出系统实体

学生成绩管理子系统实体的关系：

学生（学号，姓名，年龄，性别，籍贯，账户，身份证号，**课程号，教师工号，班级编号**）

教师（工号，姓名，性别，账户,**班级编号**，**课程号**）

班级（班级编号，人数，班级名称，院系，年级，**教师工号，学号**）

课程（课程号，课程名，学时，选修课，学分，学期，成绩，**教师工号，学号**）

教务处（教务处工号，教务处姓名，教务处密码，**班级编号，学号**）

（2）每个实体之间的主码关系：

讲授（教师工号，班级编号）

教授（教师工号，学号）

选课（学号，课程号）

授课（教师工号，课程号）

属于（学号，班级编号）

管理（教务处工号，班级编号，学号）

## 4.2将E-R图转化成表

设计学生成绩管理数据库，包括课程、学生、教师、班级，教务处五个关系,其关系模式中对每个实体定义的属性表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| Sno | Char | 8 | 主键 |
| Sname | Varchar | 10 | 非空 |
| sage | Int | 4 |  |
| sex | Enum(‘男’,’女’) | 2 |  |
| saccount | char | 10 | 非空 |
| nativeplace | char | 20 |  |
| identitycard | char | 18 | 唯一、非空 |

学生表的表结构：

教师表的表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| tno | Char | 8 | 主键 |
| tname | char | 10 | 非空 |
| Sex | Enum(‘男’,’女’) | 2 |  |
| taccount | char | 10 | 非空 |

课程表的表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| Cno | Char | 8 | 主键 |
| cname | Char | 10 | 非空 |
| ctime | int | 2 |  |
| credit | Decimal | 4 |  |

班级表的表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| classat | Char | 10 | 主键 |
| Calssname | Char | 10 | 非空 |
| rnum | int | 4 |  |
| sdept | char | 20 |  |

教务表的表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| ttno | Char | 10 | 主键 |
| ttname | Varchar | 10 |  |
| ttpassword | Varchar | 20 | 非空 |

学习情况表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| sno | Char | 8 | 主键 |
| cno | Char | 8 | 主键 |
| score | Tinyint | 2 |  |

讲授情况表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 字段大小 | 约束 |
| cno | Char | 8 | 主键 |
| tno | Char | 8 | 主键 |
| scxueqi | int | 4 |  |

# 5.物理结构设计

## 5.1建立视图

（1）创建视图class\_view,显示各个系里每个班级的详细信息。

create view class\_view(classat,classname,sdept)

as select classat,classname,sdept

from class

group by sdept;

（2）创建一个学生视图s\_sc，要求显示学生学号，姓名，性别，年龄，账号，身份证号等信息的学生信息视图：

create view s\_sc as

select s.sno,s.sname,sex,sage,saccount,identitycard

from s;

（3）创建视图grade\_view,显示成绩不及格的学生学号、姓名、课程名和成绩信息：

create view grade\_view(sno,sname,cname,score)

as select s.sno,sname,cname,c.score

from s,c,sc

where s.sno=sc.sno and c.cno=sc.cno and c.score<60;

(4)创建课程信息c\_js视图(课程号、课程名、学时、学期、学分、工号)

create view c\_impro as

select c.cno,cname,credit,ctime,scxueqi,tno

from c,js

where c.cno=js.cno

## 5.2 建立索引

（1）分别对学号，工号，课程号建立索引：

create unique index in\_sno on s(sno);

create unique index in\_tno on t(tno);

create unique index in\_cno on c(cno);

（2）S表按学号升序建唯一索引，C表按课程号升序建唯一索引，SC表按学号升序和课程号降序建唯一索引：

create unique index stu\_idx\_sno on s(sno asc);

create unique index stu\_idx\_cno on c(cno asc);

create unique index stu\_idx\_cno\_sno on sc(sno asc,cno desc);

## 5.3建立触发器

（1）创建一个更新后触发器tsupdate，实现在更新学生表的sno时，级联更新对于sc中的相关记录的sno值：

create trigger tsupdate after update

on s for each row

begin

declare newsno char(10);

declare oldsno char(10);

set newsno=new.sno;

set oldsno=old.sno;

if newsno<>oldsno then

update sc set sno=newsno where sno=oldsno;

end if;

end $$

（2）创建一个触发器，当class表中增加一个学生，该班级表中相应的系部的人数加1；

delimiter $$

create trigger insert\_after\_trigger after insert on class for each row

begin

update class set rnum=rnum+1 where sdept=new.sdept;

end$$

（3)创建一个触发器insert\_before\_trigger，当SC表录入成绩时，grade值要求在0到100之间：

delimiter $$

create trigger insert\_before\_trigger before insert

on sc for each

row

begin

if(new.score>=0&&new.score<=100)then

set new.score=new.score;

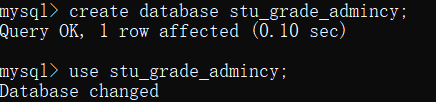
else insert into sc values(0);

end if;

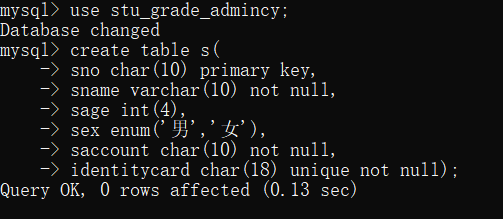
end $$

# 6.数据库实施

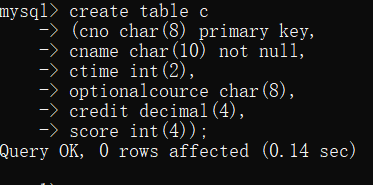
## 6.1 MySQL创建数据库和数据表



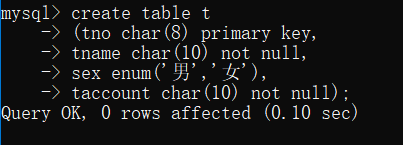
学生表：



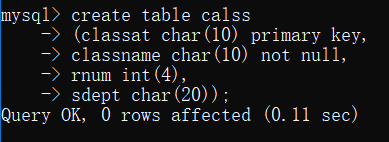
课程表：



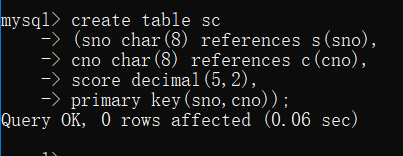
教师表：



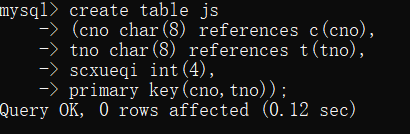
班级表：



学习情况表：

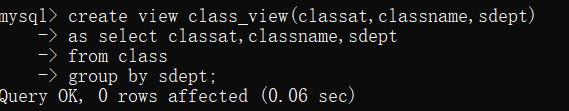


讲授表：

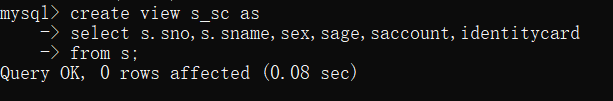


## 6.2 MySQL建立视图

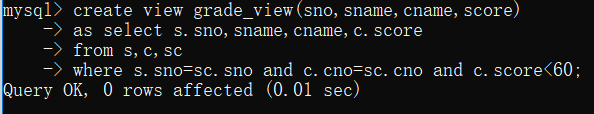
（1）创建视图class\_view,显示各个系里每个班级的详细信息。



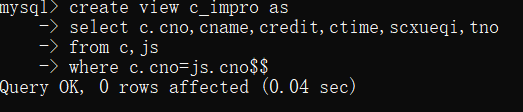
（2）创建一个学生视图s\_sc，要求显示学生学号，姓名，性别，年龄，账号，身份证号等信息的学生信息视图。



1. 创建视图grade\_view,显示挂科的学生学号、姓名、课程名和成绩信息。

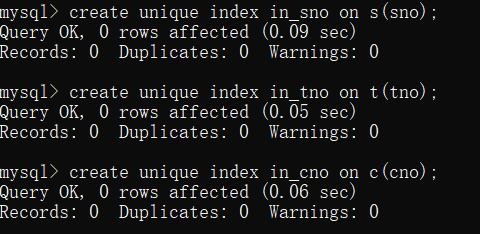
****

**(4)**创建课程信息c\_impro视图(课程号、课程名、学时、学期、学分、工号)

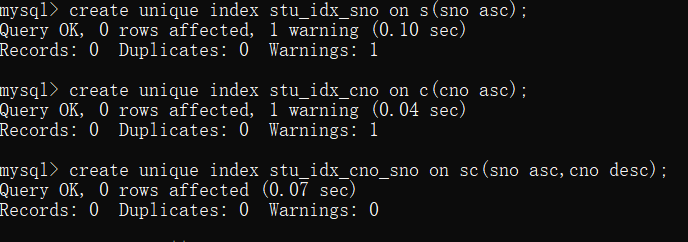


## 6.3 MySQL建立索引

（1）分别对学号，工号，课程号建立索引：

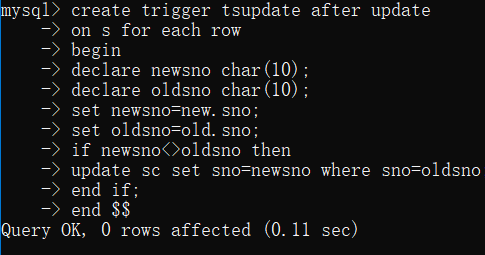


（2）S表按学号升序建唯一索引，C表按课程号升序建唯一索引，SC表按学号升序和课程号降序建唯一索引：

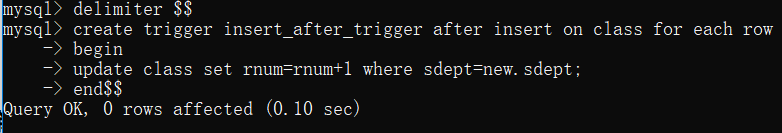


## 6.4 MySQL建立触发器

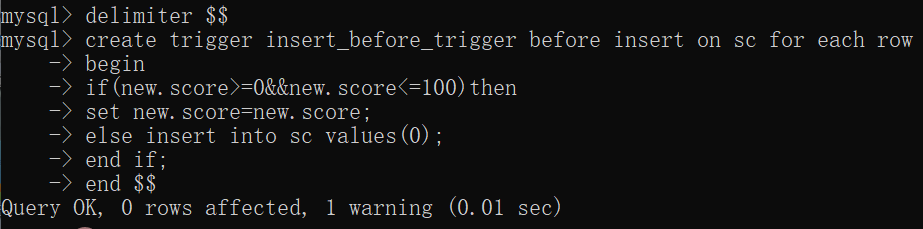
（1）创建一个更新后触发器tsupdate，实现在更新学生表的sno时，级联更新对于sc中的相关记录的sno值：

****

（2）创建一个触发器，当class表中增加一个学生，该班级表中相应的系部的人数加1；

****

（3）创建一个触发器insert\_before\_trigger，当SC表录入成绩时，grade值要求在0到100之间：



# 实训设计总结

本次数据库实训使我收获颇多，首先我们小组设计的是学生成绩管理系统，针对学生管理系统的求分析阶段，概念结构设计阶段、逻辑结构设计阶段、物理结构设计阶段、数据库实施阶段、数据库运行和维护阶段等我们做了一系列的准备工作，经过大家的共同努力和合作，学生成绩管理系统已经成功完成了。经过这次的数据库系统设计，自己也总结了关于学生成绩管理系统存在的一些问题，还有与数据库原理课程的一些区别，不过对于自身来说收获还是非常丰富的，先有理论知识作为“地基”，再加上结合数据库实践，更加综合了我的关于数据库运用的综合实力，让我学习到了更多。

首先，更近一步的了解了数据库的基本操作，在实训之前，关于数据库的认识大多是对理论知识上的学习，还有就是通过对应题目在MySQL命令端用数据库语言进行查询、修改、创建表、索引、触发器等操作来加深对理论知识的理解为后面的系统设计打下了基础。然而，对于我来说，令我比较头疼的就是对概念的理解，数据类型的灵活运用怎样才能做到精益求精，减少数据的冗余等。从不熟练到慢慢从增删改查操作中不断总结规律，形成我自己的一套数据库系统设计体系。

这次的学生成绩管理系统，在自己所掌握的知识下，进行了综合设计，完全体现了自己在数据库语言和数据库原理课程学习情况，充分地为自己以后更深入了解数据库语言奠下了深厚的基础。在系统设计的过程中，我从中发现，学习数据库要有细心和有耐性，对于数据的处理和运用是不可忽视的，在现今的大数据时代里，数据的重要性是何其重要。在本次实训中更以自身所学和不断从外界所学相结合才能设计一套高品质的数据库系统。最重要的是，这次数据库项目的设计实训，为我们以后程序开发的道路打下了扎实的基础。