实验设计报告

**实验课程名称 数据库课程设计**

**专 业： 计算机科学与技术**

**班 级： 计科三班**

**姓 名：**

**学 号：**

**实验学时：**

**指导教师： 胡艳荣**

**成 绩：**

目录

[一、数据库课程设计要求与目的 3](#_Toc90480938)

[二、需求分析 4](#_Toc90480939)

[2.1 需求分析 4](#_Toc90480940)

[2.2 系统功能模块图 5](#_Toc90480941)

[2.3数据字典 6](#_Toc90480942)

[2.4 事务 9](#_Toc90480943)

[2.5 触发器： 10](#_Toc90480944)

[三、概念结构设计 10](#_Toc90480945)

[3. 1 实体与联系 10](#_Toc90480946)

[3．2 局部E-R图 11](#_Toc90480947)

[3．3 全局E-R图 13](#_Toc90480948)

[四、逻辑结构设计 13](#_Toc90480949)

[4.1关系模式 13](#_Toc90480950)

[4.2建表属性与sql代码 14](#_Toc90480951)

[4.3插入数据的sql代码 17](#_Toc90480952)

[五、物理结构设计 20](#_Toc90480953)

[六、界面展示 21](#_Toc90480954)

[六、结论 25](#_Toc90480955)

[七、SQL代码块 27](#_Toc90480956)

[7.1：用户权限数据库代码 27](#_Toc90480957)

[7.2：用户权限设置： 32](#_Toc90480958)

[7.3：python代码 –SQL主代码 32](#_Toc90480959)

[八、python代码块： 43](#_Toc90480960)

**高校学籍管理系统**

# 一、数据库课程设计要求与目的

1. 目的

1.通过数据库课程设计，来进一加深对数据库开发与应用的了解，掌握sql语句与sql服务的要领，从而达到让我们实践的目的。

2.首先查阅书籍可以知道数据库课程设计数据库系统开发步骤的要领有：需求分析，概念设计，逻辑结构设计，物理结构设计，数据库实施阶段，由此我们可以展开以下高校学籍管理系统的设计。

1. 功能要求

实现学生信息、班级、院系、专业等的管理；

实现课程、学生成绩信息管理；

实现学生的奖惩信息管理；

创建规则用于限制性别项只能输入“男”或“女”；

创建视图查询各个学生的学号、姓名、班级、专业、院系；

创建存储过程查询指定学生的成绩单；

创建触发器当增加、删除学生和修改学生班级信息时自动修改相应班级学生人数；

建立数据库相关表之间的参照完整性约束。

# 二、需求分析

## 2.1 需求分析

学生学籍管理系统，旨在探索一种以互联网为平台的管理模式。这种新的管理模式，使教务管理突破时空限制，实现管理的网络化，提高管理效率和标准化水平。使学校管理者、教师和学生可以在任何时候、任何地点通过网络进行管理与查询，让管理者从繁重的工作中解脱出来，将主要精力转移到创造性的教学改革工作中。本系统主要完成对学生基本信息、教师信息、院系信息、专业信息、课程信息、成绩信息和奖惩信息等多种数据信息的管理，即对数据库中表的录入、修改、删除和查询等功能。

学籍管理系统的功能需求包括管理员、学生和教师对功能的需求的三大部分:

1. **管理员对功能的需求:**管理员权限最大，可以对学生、教师、课程进行管理，包括对学生学籍信息的增删改查，对教师信息的增删改查，以及对课程信息的增删改查等。

**2.学生对功能的需求:**学生只是利用此系统修改自己的登录密码，查

询自己的学籍信息，查询课程信息，进行选课，查询成绩。

**3.教师对功能的需求:**

教师利用该系统可以修改自己的登录密码，查询自己的信息，查询自己的授课信息，还对学生和课程进行管理，录入、修改学生的成绩。

## 2.2 系统功能模块图

其系统功能模块说明如下：

* 学生信息管理模块：该模块主要负责所有在校学生的个人基本信息。学生通过这个模块，可以查询自己在校的学籍信息，以及修改自己的登录密码等。管理员通过这个模块可以增加、删除、更改、查询学生的学籍信息。
* 学生课程管理模块：该模块主要负责管理全校师生的课程信息。主要功能包括增加、删除、更改、查询课程信息，其中只有管理员才具有对课程信息进行维护的权限（增加、删除、更改）。学生课程管理模块是选课管理模块的基础，只有在课程管理中添加选修课的信息，学生才能进行选课。针对选课模块，其功能行使者是学生，学生通过浏览全部的课程信息，选择符合自己要求的课程。
* 学生成绩管理模块：该模块的功能主要由老师所拥有。对于选择自己所教授课程的学生，老师具有给定该学生的专业分数和修改该学生的专业分数的权限，而学生只具有查询自己专业成绩的权限。
* 奖惩信息管理模块：该模块主要负责学生受奖励或受批评的公告信息。由学校管理员负责发布。
* 系统管理员管理模块：该模块由系统管理员全权拥有，负责管理学生、管理教师、管理课程等功能模块。

## 2.3数据字典

数据字典的数据项如下列表所示：

**表 1 院系信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 院系号 | varchar | 10 |
| 2 | 院系名 | varchar | 10 |
| 3 | 院长名 | varchar | 10 |

**表2 专业信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 专业号 | varchar | 10 |
| 2 | 专业名 | char | 10 |

**表3 学生基本信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 学号 | varchar | 20 |
| 2 | 姓名 | varchar | 20 |
| 3 | 性别 | Varchar | 2 |
| 4 | 年龄 | Int | 4 |
| 5 | 班级 | Varchar | 10 |
| 6 | 入学时间 | Datetime |  |

**表4 教师基本信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 教工号 | varchar | 10 |
| 2 | 姓名 | varchar | 10 |
| 3 | 性别 | varchar | 2 |
| 4 | 职称 | varchar | 10 |
| 5 | 所属院系 | varchar | 10 |
| 6 | 联系电话 | Varchar | 20 |

**表5 课程信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 课程号 | varchar | 10 |
| 2 | 课程名 | char | 10 |
| 3 | 学分 | Int | 2 |
| 4 | 学时 | Int | 2 |
| 5 | 上课时间 | varchar | 10 |
| 6 | 教师号 | Varchar | 10 |
| 7 | 教室号 | Varchar | 10 |

**表6 学生选课信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 学号 | varchar | 10 |
| 2 | 课程号 | varchar | 10 |
| 3 | 成绩 | int | 3 |

**表7 教师授课信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 教工号 | varchar | 10 |
| 2 | 课程号 | varchar | 10 |

**表8用户信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 用户 | varchar | 20 |
| 2 | 密码 | varchar | 20 |

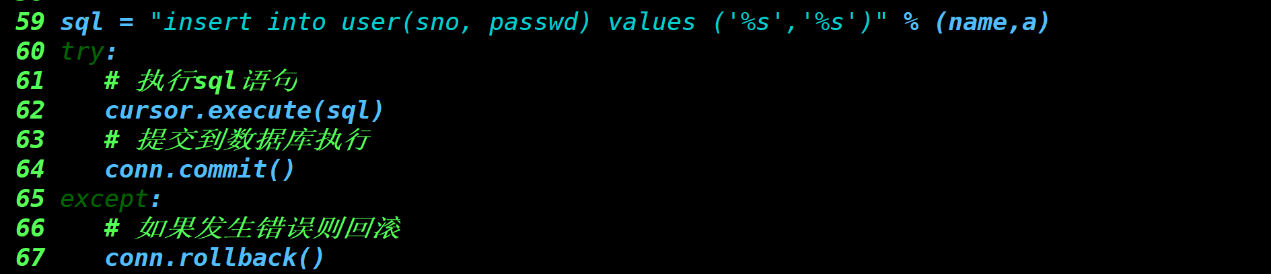
**表9奖惩情况表**

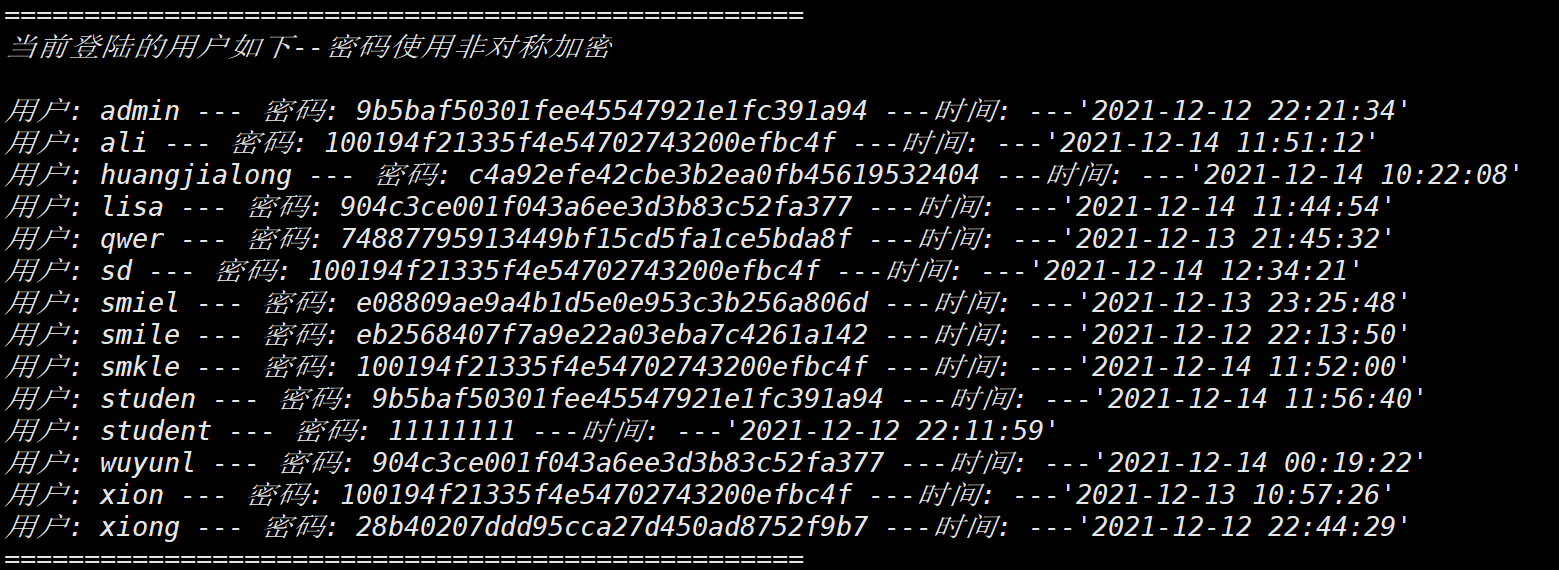
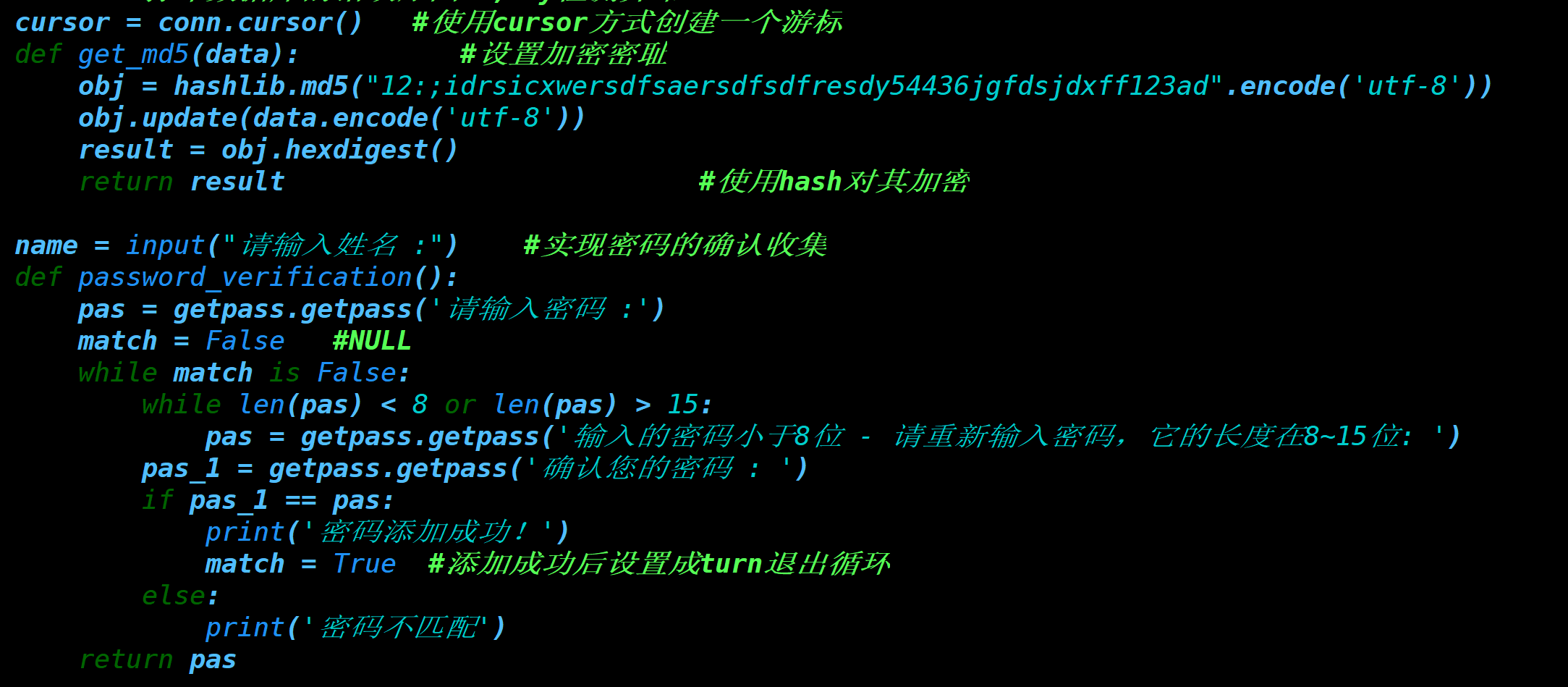
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 别名 | 数据类型 | 数据长度 |
| 1 | 学号 | varchar | 20 |
| 2 | 奖励 | varchar | 100 |
| 3 | 惩罚 | Varchar | 100 |

## 2.4 事务

**针对于课程设计出现的可能问题，我使用了事务来解决。**

* **为数据库操作提供了一个从失败中恢复到正常状态的方法，同时提供了数据库即使在异常状态下仍能保持一致性的方法。**
* **当多个应用程序在并发访问数据库时，可以在这些应用程序之间提供一个隔离方法，以防止彼此的操作互相干扰。**

为了保持程序的完整性，我在代码中使用了大量的事务来保证数据不会丢失。  
同时，在python中，为了使数据更加的安全，我使用了单向加密（hash函数）确保一些SQL注入的发生。



## 2.5 触发器：

在统计人数上面使用了select count(sno) from student;对人数进行统计。同时为了防止增删改查对人数的影响，这样不由得就用到了触发器，在增加人数的同时，对数据进行跟新。

# 三、概念结构设计

## 3. 1 实体与联系

**实体 （属性）**

院系(**院系号**、院系名、院长名) ；

专业(**专业号**、专业名) ；

学生(**学号**、姓名、性别、出生日期、联系电话、入学时间)；

教师(**教工号**、姓名、性别、职称、院系号、联系电话)；

课程（**课程号**、课程名、学时、学分、上课时间，上课教室)；

选课（**学号**、**课程号**、成绩)。

用户（用户，密码）

奖惩情况（学号，奖励，惩罚）

教师授课（教工号，课程号）

**实体间的联系情况与转换规则：**

院系与教师的联系是工作 对应的联系情况是一对多；

教师与课程的联系是授课 对应的联系情况是多对多；

学生与专业的联系是属于 对应的联系情况是一对多；

学生与选课的联系是报课情况，对应的关系是多对多；

## 3．2 局部E-R图

学生

专业

院系

选课

教师授课

用户表

奖惩情况

课程信息

教师

## 3．3 全局E-R图

学生

成绩

教师

课程

查询

选课

授课

录入

# 四、逻辑结构设计

## 4.1关系模式

将E-R图转换为关系模型实际上就是要奖实体型、实体的属性和实体型之间的联系转换为关系模式，这种转换一般遵循如下原则：一个实体型转换为一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码。现将概念结构设计阶段设计好的基本

E-R图转换为关系模型，如下所示：

院系(**院系号**、院系名、院长名) ；

专业(**专业号**、专业名) ；

学生(**学号**、姓名、性别、出生日期、联系电话、入学时间)；

教师(**教工号**、姓名、性别、职称、院系号、联系电话)；

课程（**课程号**、课程名、学时、学分、上课时间，上课教室)；

选课（**学号**、**课程号**、成绩)。

用户（用户，密码）

奖惩情况（学号，奖励，惩罚）

教师授课（教工号，课程号）

## 4.2建表属性与sql代码

**院系表：**



CREATE TABLE `department` (

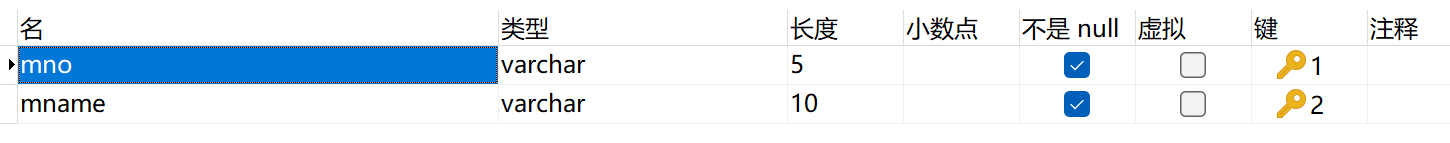
`dmpno` varchar(5) NOT NULL,

`dname` varchar(10) NOT NULL,

`dmphead` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**专业表：**

****

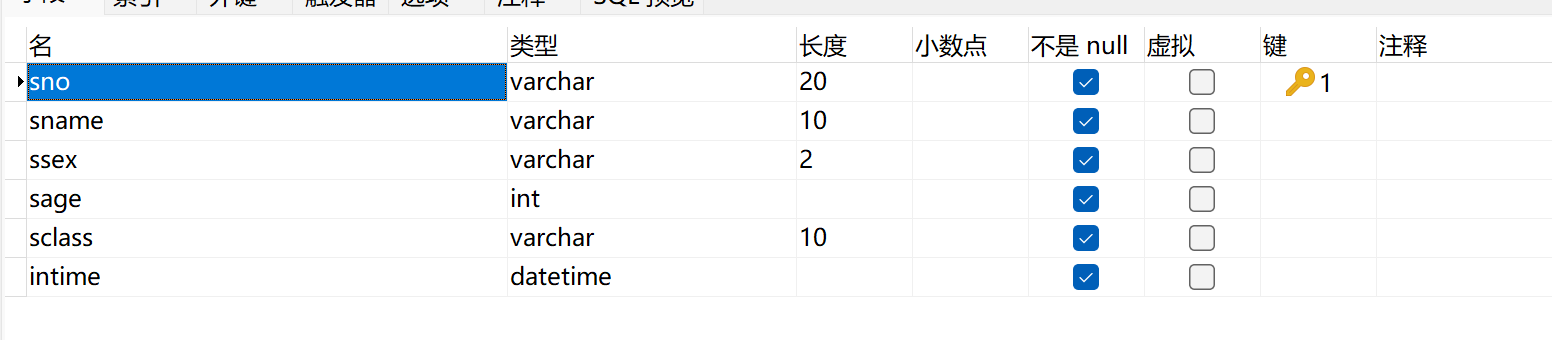
CREATE TABLE `major` (

`mno` varchar(5) NOT NULL,

`mname` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**学生表：**

****

CREATE TABLE `student` (

`sno` varchar(20) NOT NULL,

`sname` varchar(10) NOT NULL,

`ssex` varchar(2) NOT NULL,

`sage` int(4) NOT NULL,

`sclass` varchar(10) NOT NULL,

`intime` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**用户表：**

****

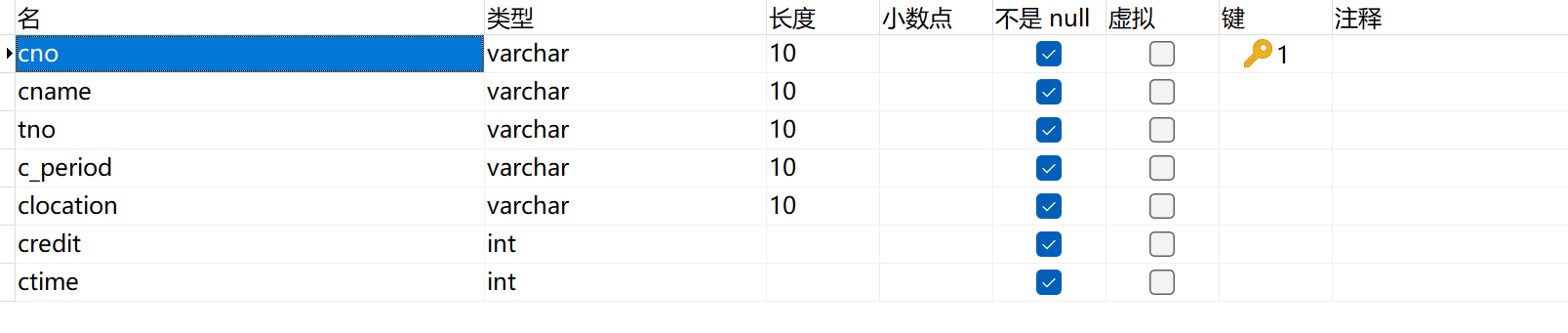
CREATE TABLE `account` (

`username` varchar(20) NOT NULL,

`passward` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**课程表：**

****

CREATE TABLE `course` (

`cno` varchar(10) NOT NULL,

`cname` varchar(10) NOT NULL,

`tno` varchar(10) NOT NULL,

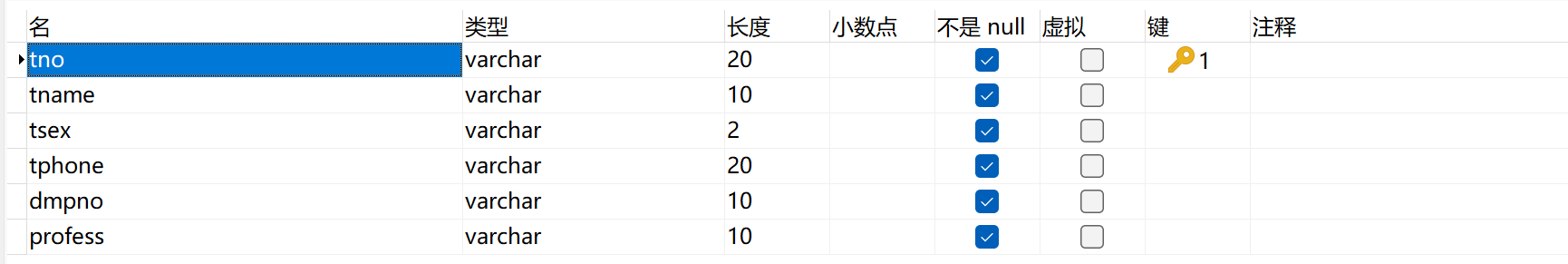
`c\_period` varchar(10) NOT NULL,

`clocation` varchar(10) NOT NULL,

`credit` int(2) NOT NULL,

`ctime` int(2) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**教师表：  
**

CREATE TABLE `teacher` (

`tno` varchar(20) NOT NULL,

`tname` varchar(10) NOT NULL,

`tsex` varchar(2) NOT NULL,

`tphone` varchar(20) NOT NULL,

`dmpno` varchar(10) NOT NULL,

`profess` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**教师任课表：**

****

CREATE TABLE `teacher\_course` (

`tno` varchar(10) NOT NULL,

`cno` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**选课表：**



CREATE TABLE `stu\_course` (

`sno` varchar(20) NOT NULL,

`cno` varchar(10) NOT NULL,

`grade` int(3) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

**奖惩情况表：**

****

CREATE TABLE `stu\_reward\_punishment` (

`sno` varchar(20) NOT NULL,

`reward` varchar(100) DEFAULT NULL,

`punishment` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

## 4.3插入数据的sql代码

"创建表格插入数据"

1. insert into student values ('202032120777', '张晓梅', '女', 20, '经管1701', '2017-09-01');
2. insert into student values ('201932120778', '李晓龙', '男', 21, '数学1601', '2020-09-01');
3. insert into student values ('202132120779', '黄飞红', '女', 20, '英语1801', '2018-09-01');
4. insert into student values ('202032120768', '李云隆', '男', 21, '数学1701', '2017-09-01');
5. insert into student values ('201932120720', '张阿红', '女', 19, '英语1901', '2019-09-01');
6. insert into student values ('201932120711', '马晕', '男', 22, '数学1601', '2020-09-01');
7. insert into student values ('202032120762', '鲁智深', '男', 21, '经管1701', '2017-09-01');
8. insert into student values ('202132120719', '孙文空', '女', 19, '英语1801', '2018-09-01');
9. insert into student values ('201932120710', '马花藤', '男', 22, '数学1601', '2020-09-01');
10. insert into course values ('c001', '微观经济学', '122118', '周四 第10-12节', '3-311', 3, 80);
11. insert into course values ('c002', '宏观经济学', '122124', '周五 第10-12节', '22-402', 6, 60);
12. insert into course values ('c003', '英语名著欣赏', '122125', '周三 第1-3节', '4-301', 2, 60);
13. insert into course values ('c004', '数学分析', '122110', '周二 第1-4节', '20-201', 6, 80);
14. insert into course values ('c005', '高等数学', '122110', '周四 第1-4节', '3-310', 6, 80);
15. insert into course values ('c006', '商务英语写作', '122122', '周四 第8-9节', '5-301', 4, 80);
16. insert into teacher values ('122118', '陈读', '男', '17398447611' , 'd001', '副教授');
17. insert into teacher values ('122124', '韩宏', '女', '17398440222' , 'd001', '教授');
18. insert into teacher values ('122125', '马冬梅', '女', '17398449118' , 'd002', '副教授');
19. insert into teacher values ('122110', '唐伯虎', '男', '17398443912' , 'd003', '副教授');
20. insert into teacher values ('122122', '谭英文', '男', '17398442121' , 'd002', '副教授');
21. insert into department values ('d001', '经济学院', '王元');
22. insert into department values ('d002', '英语学院', '张小龙');
23. insert into department values ('d003', '数学学院', '黄飞');
24. insert into major values ('m001', '经济学专业');
25. insert into major values ('m002', '英文专业');
26. insert into major values ('m003', '数学专业');
27. insert into stu\_course values ('202032120777', 'c001', null);
28. insert into stu\_course values ('202032120777', 'c002', null);
29. insert into stu\_course values ('201932120778', 'c004', null);
30. insert into stu\_course values ('201932120778', 'c005', null);
31. insert into stu\_course values ('202132120779', 'c003', null);
32. insert into stu\_course values ('202032120768', 'c004', 70);
33. insert into stu\_course values ('202032120762', 'c002', 80);
34. insert into stu\_course values ('202132120719', 'c003', 90);
35. insert into stu\_course values ('202132120719', 'c006', 92);
36. insert into teacher\_course values ('122118', 'c001');
37. insert into teacher\_course values ('122124', 'c002');
38. insert into teacher\_course values ('122125', 'c003');
39. insert into teacher\_course values ('122110', 'c004');
40. insert into teacher\_course values ('122110', 'c005');
41. insert into teacher\_course values ('122122', 'c006');
42. insert into class\_stu\_num values ('经管1701', 2);
43. insert into class\_stu\_num values ('数学1601', 3);
44. insert into class\_stu\_num values ('数学1701', 1);
45. insert into class\_stu\_num values ('英语1801', 2);
46. insert into class\_stu\_num values ('英语1901', 1);
47. insert into stu\_reward\_punishment(sno, reward) values ('202032120777', '校一等奖学金');
48. insert into stu\_reward\_punishment(sno, punishment) values ('202032120777', '全校通告批评');
49. insert into stu\_reward\_punishment(sno, punishment) values ('201932120711', '校一等奖学金');
50. insert into account values ('stu', '1224');
51. insert into account values ('stu1', '1224');
52. insert into account values ('teach', '1224');
53. insert into account values ('admin', '1224');
54. insert into account values ('student', '1224');
55. "用户的登陆信息表格"

根据用户的需求，需建立三个视图，分别是**查看学生视图、查看课程视图、查询选课视图。**

# 五、物理结构设计

数据库在物理设备上的存储结构与存取方法成为数据库的物理结构，它依赖于选定的数据库管理系统。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，就是数据库的物理设计。

 **5.1 、数据库的物理设计：**

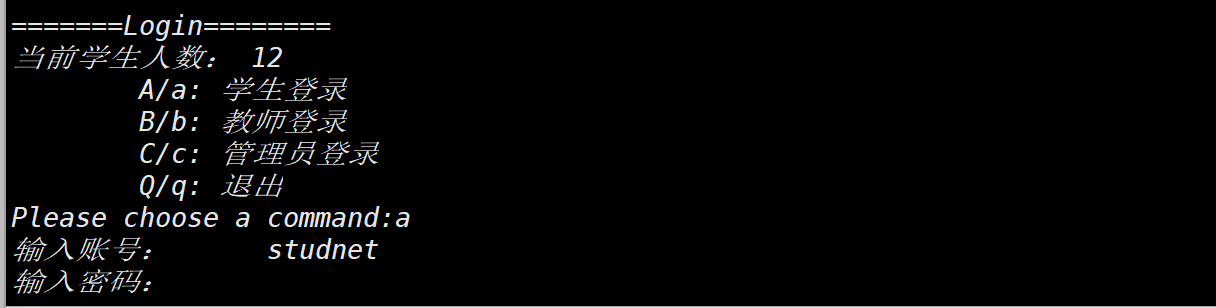
* 1. 确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构；
  2. 对物理进行评价，评价的重点是时间和空间效率。

 **5.2、关系数据库物理设计：**

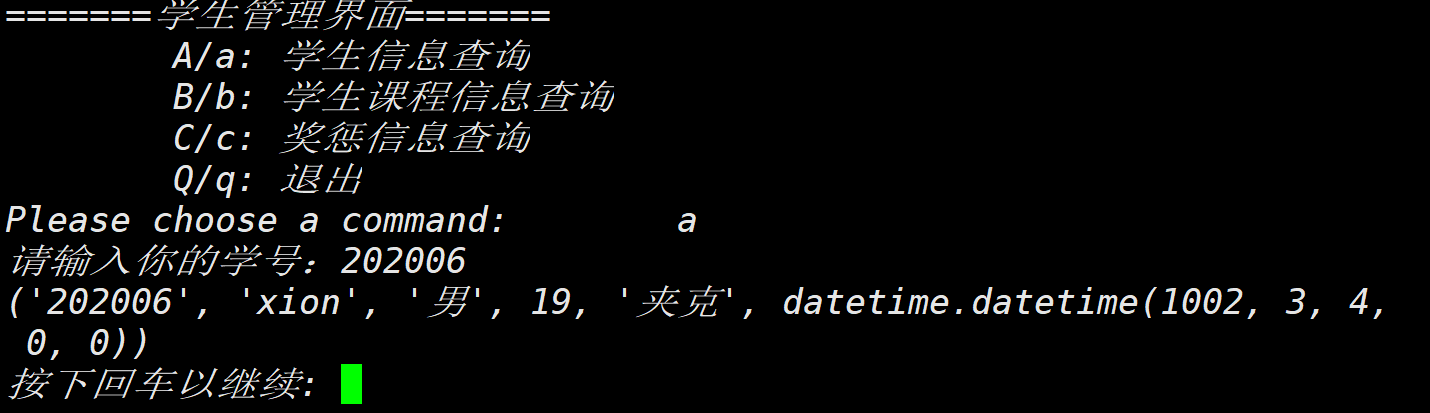
1. 为关系模式选择存取方法；常用的存储方法有索引法（B+树法）、聚簇法和HASH方法。  
   在我的设计中，使用了hash方法，对存储的数据进行单向加密和保护。
2. 设计关系、索引等数据库文件的物理存储结构。
3. 确定数据库物理结构主要指确定数据的存放位置和存储结构，包括：确定关系、索引、聚簇、日志、备份等的存储安排和存储结构，确定系统配置等。
4. 评价屋里数据库的方法完全依赖于所选用的DBMS，主要是从定量估算各种方案的存储空间、存储时间和维护代价入手，对估算结果进行权衡、比较，选择出一个较优的合理的物理结构。

# 六、界面展示

**6.1用户界面：(后台使用mysql语句可以显示统计出的学生人数)**



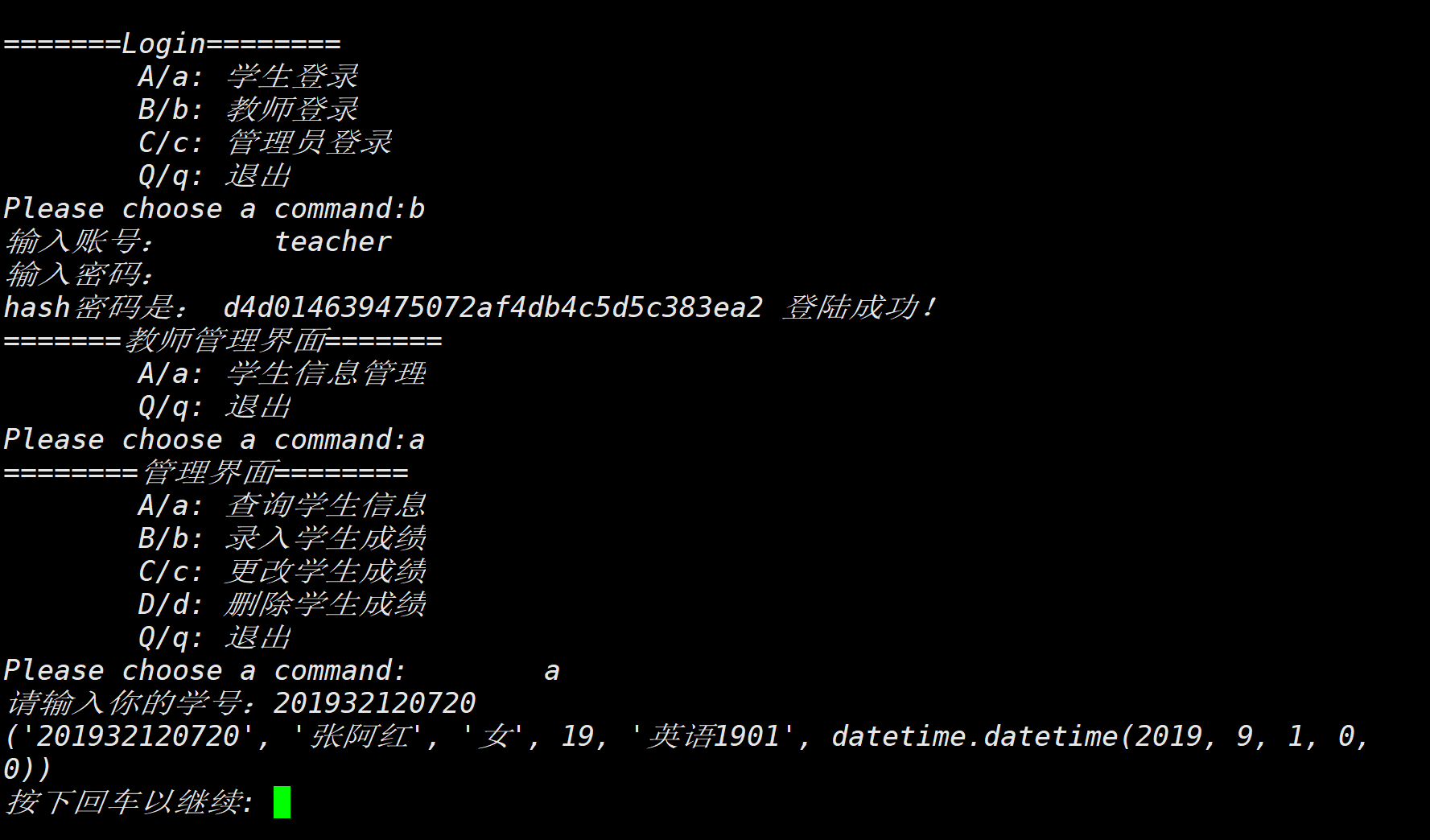
**6.2学生管理界面：（密码11111111）**



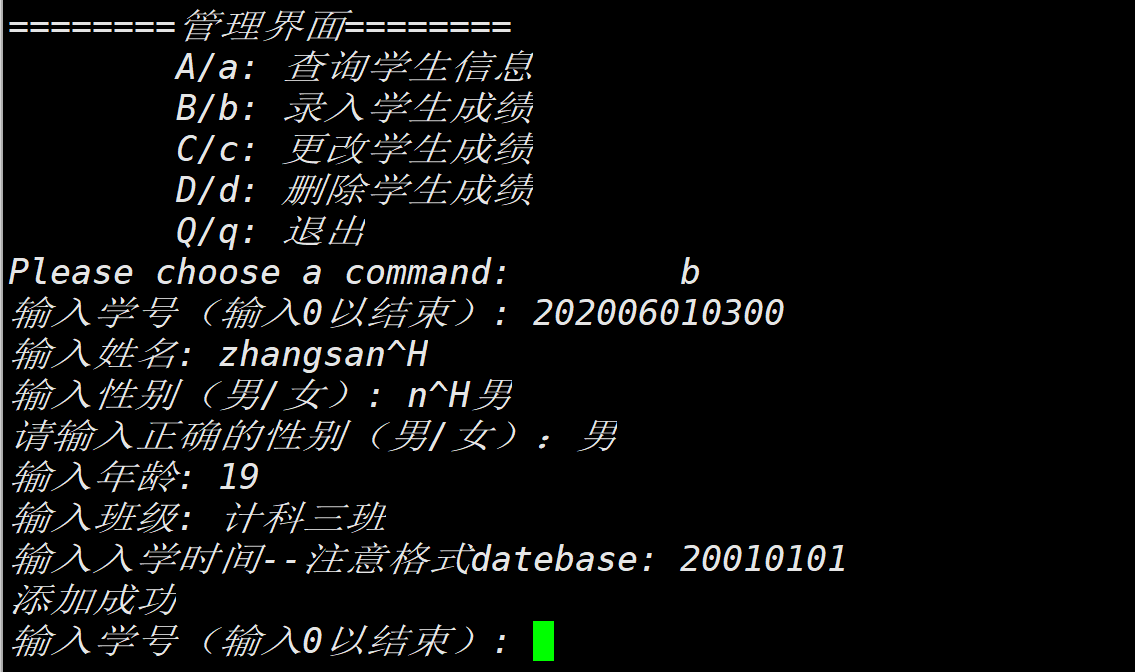


**6.3教师界面：（密码22222222）**

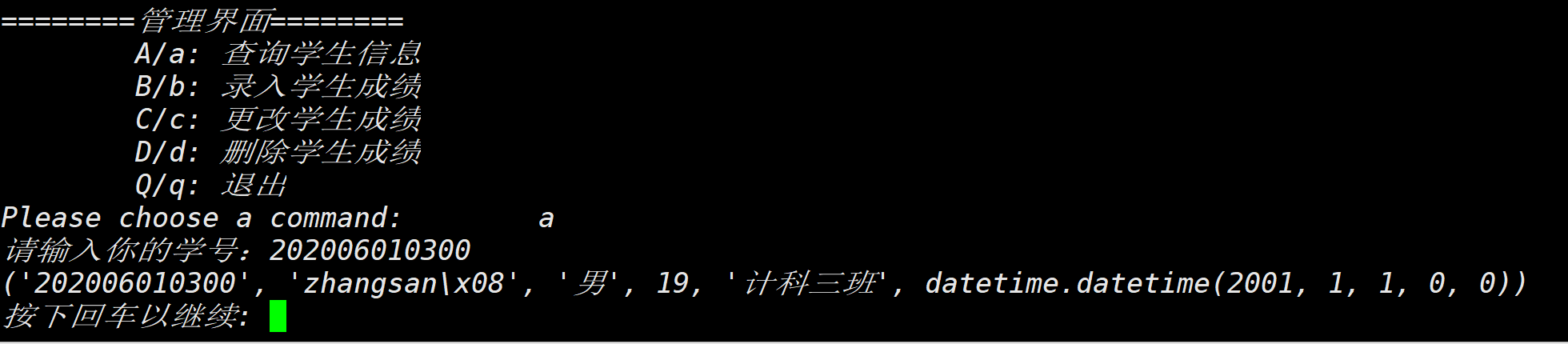
a:查询学生信息



B:录入学生信息：

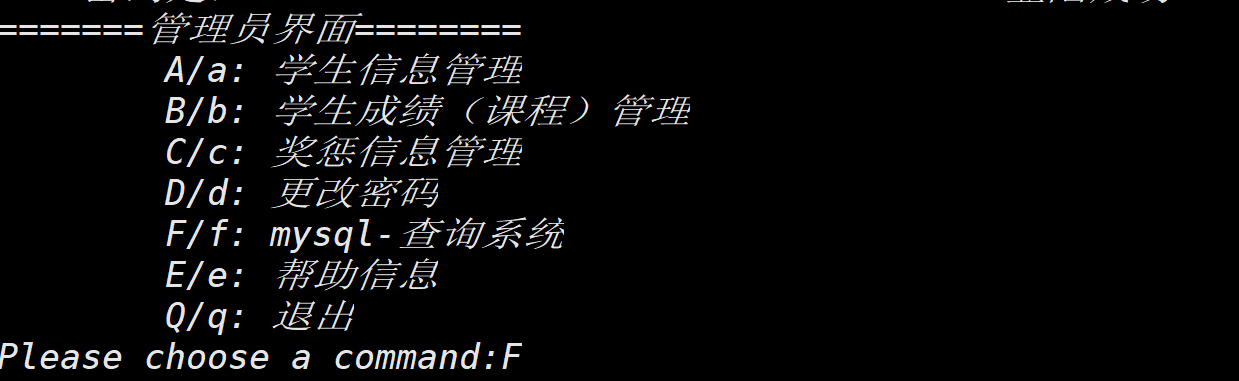


查询，添加成功！

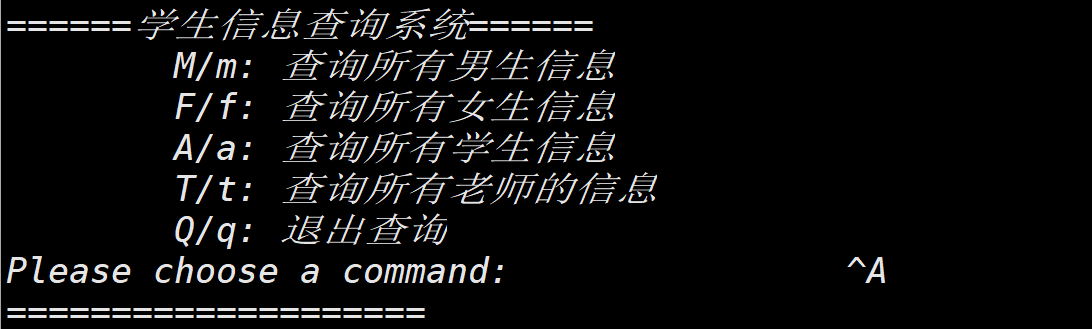


**6.4管理员界面：（密码：00000000）**

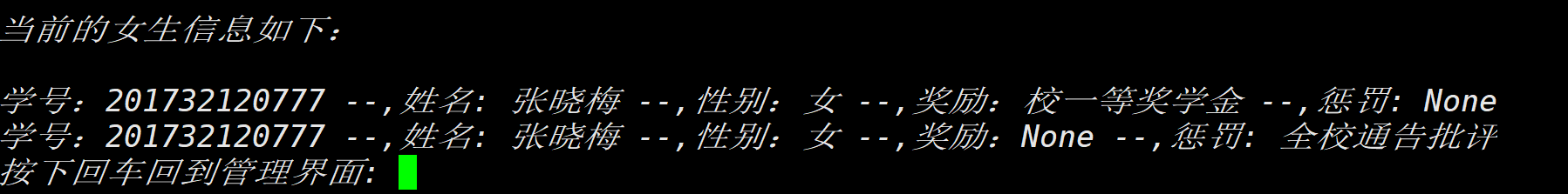
界面：



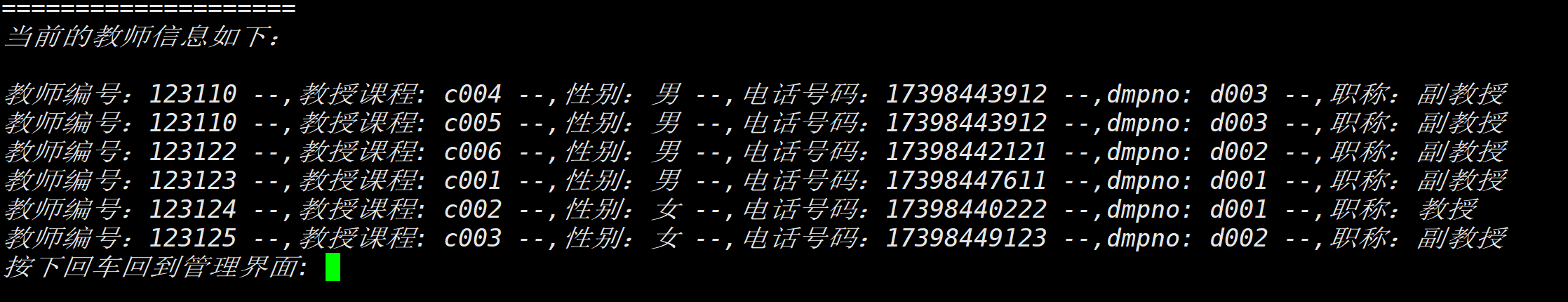
F/f查询系统：



查询所有女生：

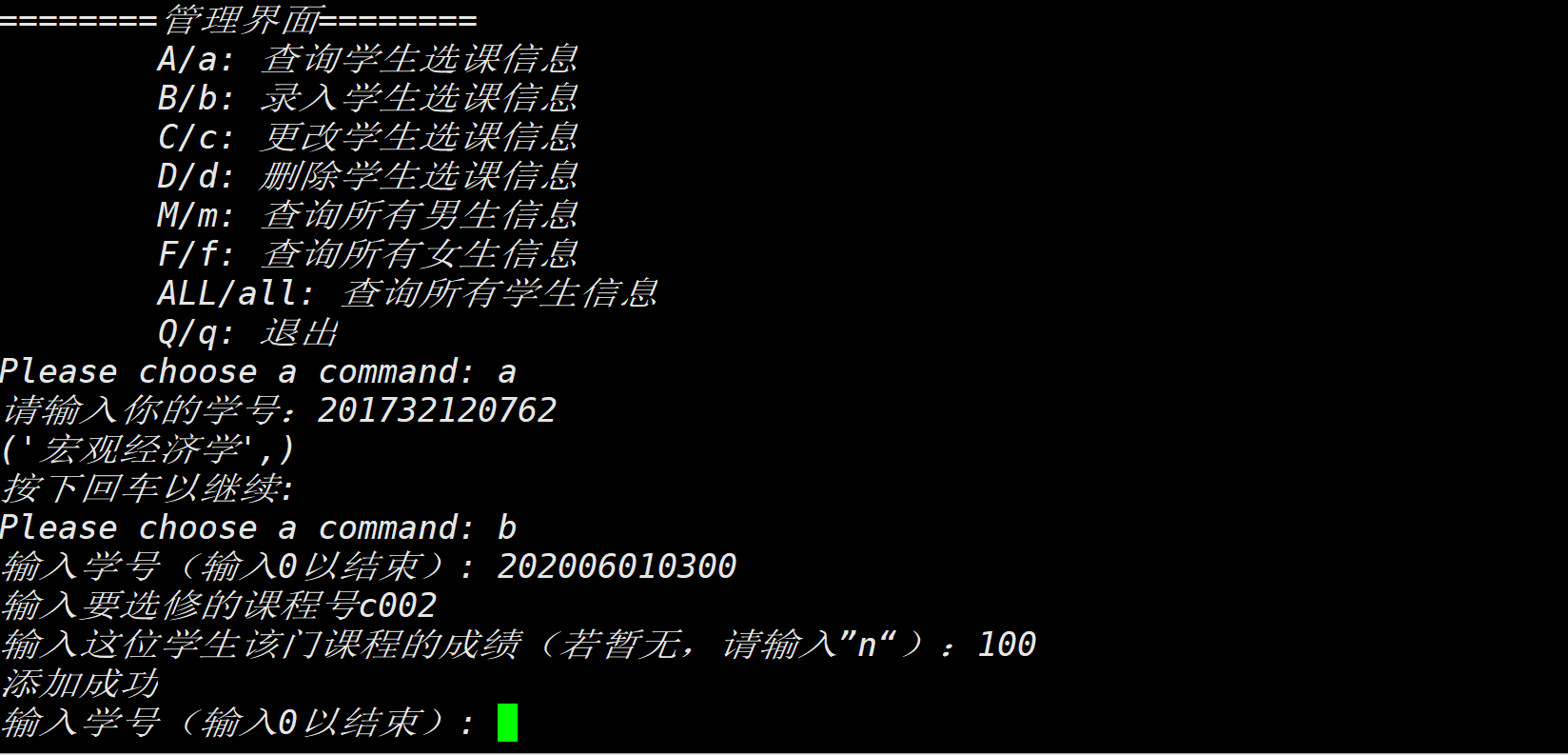


查询所有老师信息：

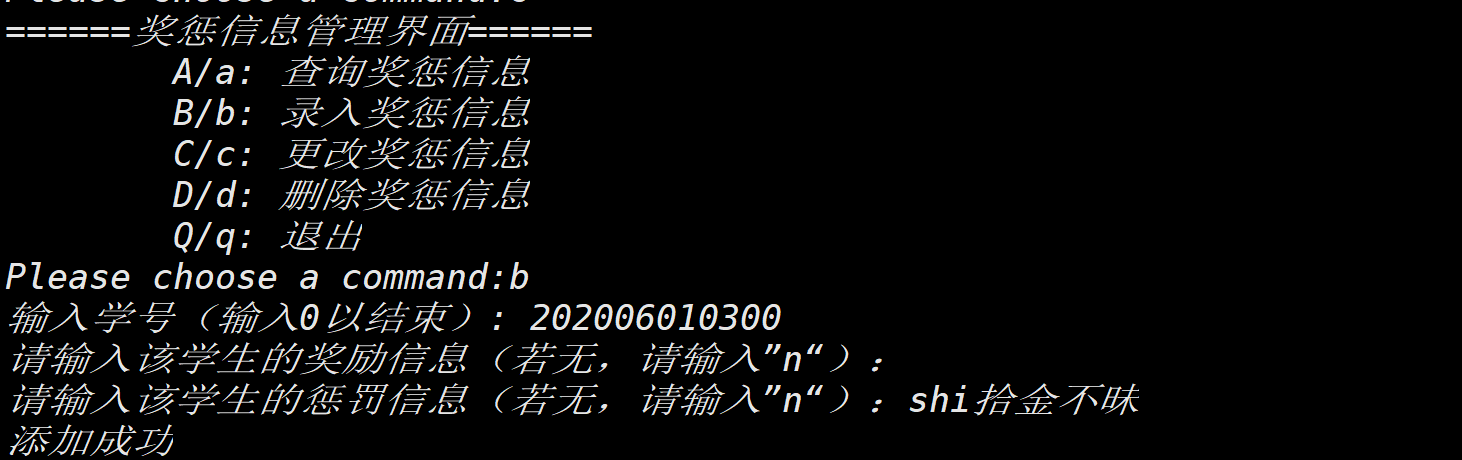


学生成绩管理系统：

查询和录入学生信息：202006010300



奖惩界面：



# 六、结论

经过这个系统的设计，我不仅仅是融会贯通了书中的知识，更重要的是在学习的过程中，所有的问题要自己去面对，有问题也只有靠自己去解决，在学习和解决这些困难的过程中提高了我学习的能力、解决问题的能力和实际工作的能力，学到了许多书本以外的知识。

在本学期的数据库设计中，通过对高校的学籍管理系统的设计，我学习到独立完成作业的重要性，锻炼了在编写实际运用数据库的实现，深刻感受到计算机学习的实用性和未来工作的巨大信息，但是由于学习效果不佳，在一些问题的处理和考虑的方面存在很大的缺陷和漏洞，希望在进一步的学习中能更好处理好相关问题。这次课题设计不能堪称完美，甚至来说还很不健全，但我会在以后的时间里去尽量的完善它，不断的对它进行升级和完善，解决系统可能会出现的。



[**文件地址：**http://xxw.nsddd.top/#s/7o0lHPdQ](file:///C:\Users\smile\.kodcloud\56c7d04c55fcfc539d6dce9d14231fbd\editor\6fce3f600bb2d4fc578d9dcc235c317d\文件地址：http:\xxw.nsddd.top\#s/7o0lHPdQ)

[**项目代码：**http://xxw.nsddd.top/#s/7pModeuw](http://xxw.nsddd.top/#s/7pModeuw)

# 七、SQL代码块

## 7.1：用户权限数据库代码

7.1.1：管理员admin:

1. *-- phpMyAdmin SQL Dump*
2. *-- version 5.1.1*
3. *-- https://www.phpmyadmin.net/*
4. *--*
5. *-- 主机： localhost*
6. *-- 生成日期： 2021-12-14 18:01:25*
7. *-- 服务器版本： 5.7.34-log*
8. *-- PHP 版本： 7.4.24*
9. SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";
10. START TRANSACTION;
11. SET time\_zone = "+00:00";
12. */\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;
13. */\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;
14. */\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;
15. */\*!40101 SET NAMES utf8mb4 \*/*;
16. *--*
17. *-- 数据库： `admin`*
18. *--*
19. *-- --------------------------------------------------------*
20. *--*
21. *-- 表的结构 `user`*
22. *--*
23. CREATE TABLE `user` (
24. `username` varchar(20) NOT NULL,
25. `passward` varchar(20) NOT NULL
26. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
27. *--*
28. *-- 转存表中的数据 `user`*
29. *--*
30. INSERT INTO `user` (`username`, `passward`) VALUES
31. ('admin', '00000000'),
32. ('张三', '00000000'),
33. ('李四', '00000000'),
34. ('王五', '00000000');
35. *--*
36. *-- 转储表的索引*
37. *--*
38. *--*
39. *-- 表的索引 `user`*
40. *--*
41. ALTER TABLE `user`
42. ADD PRIMARY KEY (`username`);
43. COMMIT;
44. */\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/*;
45. */\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/*;
46. */\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/*;

7.1.2:学生student：

1. *-- phpMyAdmin SQL Dump*
2. *-- version 5.1.1*
3. *-- https://www.phpmyadmin.net/*
4. *--*
5. *-- 主机： localhost*
6. *-- 生成日期： 2021-12-14 18:01:52*
7. *-- 服务器版本： 5.7.34-log*
8. *-- PHP 版本： 7.4.24*
9. SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";
10. START TRANSACTION;
11. SET time\_zone = "+00:00";
12. */\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;
13. */\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;
14. */\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;
15. */\*!40101 SET NAMES utf8mb4 \*/*;
16. *--*
17. *-- 数据库： `student`*
18. *--*
19. *-- --------------------------------------------------------*
20. *--*
21. *-- 替换视图以便查看 `student\_view`*
22. *-- （参见下面的实际视图）*
23. *--*
24. CREATE TABLE `student\_view` (
25. `username` varchar(20)
26. ,`passward` varchar(20)
27. );
28. *-- --------------------------------------------------------*
29. *--*
30. *-- 表的结构 `user`*
31. *--*
32. CREATE TABLE `user` (
33. `username` varchar(20) NOT NULL,
34. `passward` varchar(20) NOT NULL
35. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
36. *--*
37. *-- 转存表中的数据 `user`*
38. *--*
39. INSERT INTO `user` (`username`, `passward`) VALUES
40. ('student', '11111111'),
41. ('张三', '11111111'),
42. ('李四', '11111111'),
43. ('王五', '11111111');
44. *-- --------------------------------------------------------*
45. *--*
46. *-- 视图结构 `student\_view`*
47. *--*
48. DROP TABLE IF EXISTS `student\_view`;
49. CREATE ALGORITHM=UNDEFINED DEFINER=`student`@`%` SQL SECURITY DEFINER VIEW `student\_view`  AS SELECT `user`.`username` AS `username`, `user`.`passward` AS `passward` FROM `user` ;
50. *--*
51. *-- 转储表的索引*
52. *--*
53. *--*
54. *-- 表的索引 `user`*
55. *--*
56. ALTER TABLE `user`
57. ADD PRIMARY KEY (`username`);
58. COMMIT;
59. */\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/*;
60. */\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/*;
61. */\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/*;

7.1.3老师teacher

1. *-- phpMyAdmin SQL Dump*
2. *-- version 5.1.1*
3. *-- https://www.phpmyadmin.net/*
4. *--*
5. *-- 主机： localhost*
6. *-- 生成日期： 2021-12-14 18:02:17*
7. *-- 服务器版本： 5.7.34-log*
8. *-- PHP 版本： 7.4.24*
9. SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";
10. START TRANSACTION;
11. SET time\_zone = "+00:00";
12. */\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;
13. */\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;
14. */\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;
15. */\*!40101 SET NAMES utf8mb4 \*/*;
16. *--*
17. *-- 数据库： `teacher`*
18. *--*
19. *-- --------------------------------------------------------*
20. *--*
21. *-- 表的结构 `user`*
22. *--*
23. CREATE TABLE `user` (
24. `username` varchar(20) NOT NULL,
25. `passward` varchar(20) NOT NULL
26. ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
27. *--*
28. *-- 转存表中的数据 `user`*
29. *--*
30. INSERT INTO `user` (`username`, `passward`) VALUES
31. ('teacher', '22222222'),
32. ('张三', '22222222'),
33. ('李四', '22222222'),
34. ('王五', '22222222');
35. *--*
36. *-- 转储表的索引*
37. *--*
38. *--*
39. *-- 表的索引 `user`*
40. *--*
41. ALTER TABLE `user`
42. ADD PRIMARY KEY (`username`);
43. COMMIT;
44. */\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/*;
45. */\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/*;
46. */\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/*;

## 7.2：用户权限设置：

1. CREATE USER 'admin'@'%' IDENTIFIED WITH mysql\_native\_password AS '\*\*\*';
2. GRANT ALL PRIVILEGES ON \*.\* TO 'admin'@'%' REQUIRE NONE WITH GRANT OPTION MAX\_QUERIES\_PER\_HOUR 0,
3. MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0 MAX\_USER\_CONNECTIONS 0;
4. CREATE DATABASE IF NOT EXISTS `admin`;
5. GRANT ALL PRIVILEGES ON `admin`.\* TO 'admin'@'%';
6. "root用户，管理员"
7. REATE USER 'student'@'%' IDENTIFIED WITH mysql\_native\_password AS '\*\*\*';GRANT SELECT ON \*.\* TO 'student'@'%',
8. REQUIRE NONE WITH MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0 MAX\_USER\_CONNECTIONS 0;
9. CREATE DATABASE IF NOT EXISTS `student`;GRANT ALL PRIVILEGES ON `student`.\* TO 'student'@'%';
10. "student学生表"
11. CREATE USER 'teacher'@'%' IDENTIFIED WITH mysql\_native\_password AS '\*\*\*';
12. GRANT SELECT, INSERT, UPDATE, DELETE, FILE, EXECUTE ON \*.\* TO 'teacher'@'%' ,
13. REQUIRE NONE WITH MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0 MAX\_USER\_CONNECTIONS 0;
14. CREATE DATABASE IF NOT EXISTS `teacher`;
15. GRANT ALL PRIVILEGES ON `teacher`.\* TO 'teacher'@'%';
16. "teacher"

## 7.3：python代码 –SQL主代码

1. *-- phpMyAdmin SQL Dump*
2. *-- version 5.1.1*
3. *-- https://www.phpmyadmin.net/*
4. *--*
5. *-- 主机： localhost*
6. *-- 生成日期： 2021-12-14 18:02:44*
7. *-- 服务器版本： 5.7.34-log*
8. *-- PHP 版本： 7.4.24*
9. SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";
10. START TRANSACTION;
11. SET time\_zone = "+00:00";
12. */\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;
13. */\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;
14. */\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;
15. */\*!40101 SET NAMES utf8mb4 \*/*;
16. *--*
17. *-- 数据库： `python`*
18. *--*
19. *-- --------------------------------------------------------*
20. *--*
21. *-- 表的结构 `account`*
22. *--*
23. CREATE TABLE `account` (
24. `username` varchar(20) NOT NULL,
25. `passward` varchar(20) NOT NULL
26. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
27. *--*
28. *-- 转存表中的数据 `account`*
29. *--*
30. INSERT INTO `account` (`username`, `passward`) VALUES
31. ('admin', '1234'),
32. ('stu', '1234'),
33. ('student', '1234'),
34. ('teacher', '1234');
35. *-- --------------------------------------------------------*
36. *--*
37. *-- 表的结构 `class\_stu\_num`*
38. *--*
39. CREATE TABLE `class\_stu\_num` (
40. `class` varchar(10) NOT NULL,
41. `stu\_num` int(11) DEFAULT '0'
42. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
43. *--*
44. *-- 转存表中的数据 `class\_stu\_num`*
45. *--*
46. INSERT INTO `class\_stu\_num` (`class`, `stu\_num`) VALUES
47. ('经管1701', 2),
48. ('数学1601', 3),
49. ('数学1701', 1),
50. ('英语1801', 2),
51. ('英语1901', 1);
52. *-- --------------------------------------------------------*
53. *--*
54. *-- 表的结构 `course`*
55. *--*
56. CREATE TABLE `course` (
57. `cno` varchar(10) NOT NULL,
58. `cname` varchar(10) NOT NULL,
59. `tno` varchar(10) NOT NULL,
60. `c\_period` varchar(10) NOT NULL,
61. `clocation` varchar(10) NOT NULL,
62. `credit` int(2) NOT NULL,
63. `ctime` int(2) NOT NULL
64. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
65. *--*
66. *-- 转存表中的数据 `course`*
67. *--*
68. INSERT INTO `course` (`cno`, `cname`, `tno`, `c\_period`, `clocation`, `credit`, `ctime`) VALUES
69. ('c001', '微观经济学', '123123', '周四 第10-12节', '25-101', 3, 80),
70. ('c002', '宏观经济学', '123124', '周五 第10-12节', '22-402', 6, 60),
71. ('c003', '英语名著欣赏', '123125', '周三 第1-3节', '16-301', 2, 60),
72. ('c004', '数学分析', '123110', '周二 第1-4节', '20-201', 6, 80),
73. ('c005', '高等数学', '123110', '周四 第1-4节', '20-301', 6, 80),
74. ('c006', '商务英语写作', '123122', '周四 第8-9节', '15-201', 4, 80);
75. *-- --------------------------------------------------------*
76. *--*
77. *-- 表的结构 `department`*
78. *--*
79. CREATE TABLE `department` (
80. `dmpno` varchar(5) NOT NULL,
81. `dname` varchar(10) NOT NULL,
82. `dmphead` varchar(10) NOT NULL
83. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
84. *--*
85. *-- 转存表中的数据 `department`*
86. *--*
87. INSERT INTO `department` (`dmpno`, `dname`, `dmphead`) VALUES
88. ('d001', '经济学院', '王元'),
89. ('d002', '英语学院', '张小龙'),
90. ('d003', '数学学院', '黄飞');
91. *-- --------------------------------------------------------*
92. *--*
93. *-- 替换视图以便查看 `jxj`*
94. *-- （参见下面的实际视图）*
95. *--*
96. CREATE TABLE `jxj` (
97. `sno` varchar(20)
98. ,`sname` varchar(10)
99. ,`ssex` varchar(2)
100. ,`reward` varchar(100)
101. ,`punishment` varchar(100)
102. );
103. *-- --------------------------------------------------------*
104. *--*
105. *-- 替换视图以便查看 `jxjf`*
106. *-- （参见下面的实际视图）*
107. *--*
108. CREATE TABLE `jxjf` (
109. `sno` varchar(20)
110. ,`sname` varchar(10)
111. ,`ssex` varchar(2)
112. ,`reward` varchar(100)
113. ,`punishment` varchar(100)
114. );
115. *-- --------------------------------------------------------*
116. *--*
117. *-- 替换视图以便查看 `jxjm`*
118. *-- （参见下面的实际视图）*
119. *--*
120. CREATE TABLE `jxjm` (
121. `sno` varchar(20)
122. ,`sname` varchar(10)
123. ,`ssex` varchar(2)
124. ,`reward` varchar(100)
125. ,`punishment` varchar(100)
126. );
127. *-- --------------------------------------------------------*
128. *--*
129. *-- 表的结构 `major`*
130. *--*
131. CREATE TABLE `major` (
132. `mno` varchar(5) NOT NULL,
133. `mname` varchar(10) NOT NULL
134. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
135. *--*
136. *-- 转存表中的数据 `major`*
137. *--*
138. INSERT INTO `major` (`mno`, `mname`) VALUES
139. ('m001', '经济学专业'),
140. ('m002', '英文专业'),
141. ('m003', '数学专业');
142. *-- --------------------------------------------------------*
143. *--*
144. *-- 表的结构 `student`*
145. *--*
146. CREATE TABLE `student` (
147. `sno` varchar(20) NOT NULL,
148. `sname` varchar(10) NOT NULL,
149. `ssex` varchar(2) NOT NULL,
150. `sage` int(4) NOT NULL,
151. `sclass` varchar(10) NOT NULL,
152. `intime` datetime NOT NULL
153. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
154. *--*
155. *-- 转存表中的数据 `student`*
156. *--*
157. INSERT INTO `student` (`sno`, `sname`, `ssex`, `sage`, `sclass`, `intime`) VALUES
158. ('201632120710', '马花藤', '男', 23, '数学1601', '2016-09-01 00:00:00'),
159. ('201632120711', '马晕', '男', 23, '数学1601', '2016-09-01 00:00:00'),
160. ('201632120778', '李晓龙', '男', 21, '数学1601', '2016-09-01 00:00:00'),
161. ('201732120762', '鲁智深', '男', 21, '经管1701', '2017-09-01 00:00:00'),
162. ('201732120768', '李云隆', '男', 21, '数学1701', '2017-09-01 00:00:00'),
163. ('201732120777', '张晓梅', '女', 20, '经管1701', '2017-09-01 00:00:00'),
164. ('201832120719', '孙文空', '女', 19, '英语1801', '2018-09-01 00:00:00'),
165. ('201832120779', '黄飞红', '女', 20, '英语1801', '2018-09-01 00:00:00'),
166. ('201932120720', '张阿红', '女', 19, '英语1901', '2019-09-01 00:00:00'),
167. ('202005', 'smile', '男', 19, '计科三班', '2006-04-30 00:00:00'),
168. ('202006', 'xion', '男', 19, '夹克', '1002-03-04 00:00:00'),
169. ('202006010300', 'zhangsan', '男', 19, '计科三班', '2001-01-01 00:00:00'),
170. ('张三s', '2022320', '男', 20, '计科三班', '2020-01-03 00:00:00');
171. *-- --------------------------------------------------------*
172. *--*
173. *-- 替换视图以便查看 `student\_view`*
174. *-- （参见下面的实际视图）*
175. *--*
176. CREATE TABLE `student\_view` (
177. `username` varchar(20)
178. ,`passward` varchar(20)
179. );
180. *-- --------------------------------------------------------*
181. *--*
182. *-- 表的结构 `stu\_course`*
183. *--*
184. CREATE TABLE `stu\_course` (
185. `sno` varchar(20) NOT NULL,
186. `cno` varchar(10) NOT NULL,
187. `grade` int(3) DEFAULT NULL
188. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
189. *--*
190. *-- 转存表中的数据 `stu\_course`*
191. *--*
192. INSERT INTO `stu\_course` (`sno`, `cno`, `grade`) VALUES
193. ('201632120778', 'c004', NULL),
194. ('201632120778', 'c005', NULL),
195. ('201732120762', 'c002', 80),
196. ('201732120768', 'c004', 70),
197. ('201732120777', 'c001', NULL),
198. ('201732120777', 'c002', NULL),
199. ('201832120719', 'c003', 90),
200. ('201832120719', 'c006', 92),
201. ('201832120779', 'c003', NULL),
202. ('202006010300', 'c002', 100);
203. *-- --------------------------------------------------------*
204. *--*
205. *-- 表的结构 `stu\_reward\_punishment`*
206. *--*
207. CREATE TABLE `stu\_reward\_punishment` (
208. `sno` varchar(20) NOT NULL,
209. `reward` varchar(100) DEFAULT NULL,
210. `punishment` varchar(100) DEFAULT NULL
211. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
212. *--*
213. *-- 转存表中的数据 `stu\_reward\_punishment`*
214. *--*
215. INSERT INTO `stu\_reward\_punishment` (`sno`, `reward`, `punishment`) VALUES
216. ('201732120777', '校一等奖学金', NULL),
217. ('201732120777', NULL, '全校通告批评'),
218. ('201632120711', NULL, '校一等奖学金'),
219. ('202006010328', 'student good', 'lose off'),
220. ('202006010300', '', 'shi拾金不昧'),
221. ('202006010300', '拾金不昧，乐于助人', ''),
222. ('q', '', '');
223. *-- --------------------------------------------------------*
224. *--*
225. *-- 表的结构 `teacher`*
226. *--*
227. CREATE TABLE `teacher` (
228. `tno` varchar(20) NOT NULL,
229. `tname` varchar(10) NOT NULL,
230. `tsex` varchar(2) NOT NULL,
231. `tphone` varchar(20) NOT NULL,
232. `dmpno` varchar(10) NOT NULL,
233. `profess` varchar(10) NOT NULL
234. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
235. *--*
236. *-- 转存表中的数据 `teacher`*
237. *--*
238. INSERT INTO `teacher` (`tno`, `tname`, `tsex`, `tphone`, `dmpno`, `profess`) VALUES
239. ('123110', '唐伯虎', '男', '17398443912', 'd003', '副教授'),
240. ('123122', '谭英文', '男', '17398442121', 'd002', '副教授'),
241. ('123123', '陈读', '男', '17398447611', 'd001', '副教授'),
242. ('123124', '韩宏', '女', '17398440222', 'd001', '教授'),
243. ('123125', '马冬梅', '女', '17398449123', 'd002', '副教授');
244. *-- --------------------------------------------------------*
245. *--*
246. *-- 表的结构 `teacher\_course`*
247. *--*
248. CREATE TABLE `teacher\_course` (
249. `tno` varchar(10) NOT NULL,
250. `cno` varchar(10) NOT NULL
251. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
252. *--*
253. *-- 转存表中的数据 `teacher\_course`*
254. *--*
255. INSERT INTO `teacher\_course` (`tno`, `cno`) VALUES
256. ('123110', 'c004'),
257. ('123110', 'c005'),
258. ('123122', 'c006'),
259. ('123123', 'c001'),
260. ('123124', 'c002'),
261. ('123125', 'c003');
262. *-- --------------------------------------------------------*
263. *--*
264. *-- 表的结构 `user`*
265. *--*
266. CREATE TABLE `user` (
267. `sno` varchar(30) NOT NULL,
268. `passwd` varchar(50) NOT NULL,
269. `datetime` datetime DEFAULT CURRENT\_TIMESTAMP
270. ) ENGINE=InnoDB DEFAULT CHARSET=gbk;
271. *--*
272. *-- 转存表中的数据 `user`*
273. *--*
274. INSERT INTO `user` (`sno`, `passwd`, `datetime`) VALUES
275. ('123', '100194f21335f4e54702743200efbc4f', '2021-12-14 16:57:38'),
276. ('admin', '9b5baf50301fee45547921e1fc391a94', '2021-12-12 22:21:34'),
277. ('ali', '100194f21335f4e54702743200efbc4f', '2021-12-14 11:51:12'),
278. ('asd', '100194f21335f4e54702743200efbc4f', '2021-12-14 17:16:40'),
279. ('dsa', '100194f21335f4e54702743200efbc4f', '2021-12-14 17:11:27'),
280. ('fsad', '100194f21335f4e54702743200efbc4f', '2021-12-14 17:00:08'),
281. ('huangjialong', 'c4a92efe42cbe3b2ea0fb45619532404', '2021-12-14 10:22:08'),
282. ('lisa', '904c3ce001f043a6ee3d3b83c52fa377', '2021-12-14 11:44:54'),
283. ('qwer', '74887795913449bf15cd5fa1ce5bda8f', '2021-12-13 21:45:32'),
284. ('sd', '100194f21335f4e54702743200efbc4f', '2021-12-14 12:34:21'),
285. ('sda', '2ae7a2005507312383f0a2872827b91b', '2021-12-14 16:41:12'),
286. ('smiel', 'e08809ae9a4b1d5e0e953c3b256a806d', '2021-12-13 23:25:48'),
287. ('smil', '100194f21335f4e54702743200efbc4f', '2021-12-14 16:55:37'),
288. ('smile', 'eb2568407f7a9e22a03eba7c4261a142', '2021-12-12 22:13:50'),
289. ('smkle', '100194f21335f4e54702743200efbc4f', '2021-12-14 11:52:00'),
290. ('studen', '9b5baf50301fee45547921e1fc391a94', '2021-12-14 11:56:40'),
291. ('student', '11111111', '2021-12-12 22:11:59'),
292. ('wuyunl', '904c3ce001f043a6ee3d3b83c52fa377', '2021-12-14 00:19:22'),
293. ('xion', '100194f21335f4e54702743200efbc4f', '2021-12-13 10:57:26'),
294. ('xiong', '28b40207ddd95cca27d450ad8752f9b7', '2021-12-12 22:44:29');
295. *-- --------------------------------------------------------*
296. *--*
297. *-- 视图结构 `jxj`*
298. *--*
299. DROP TABLE IF EXISTS `jxj`;
300. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `jxj`  AS SELECT `a`.`sno` AS `sno`, `a`.`sname` AS `sname`, `a`.`ssex` AS `ssex`, `b`.`reward` AS `reward`, `b`.`punishment` AS `punishment` FROM (`student` `a` join `stu\_reward\_punishment` `b` on((`a`.`sno` = `b`.`sno`))) ;
301. *-- --------------------------------------------------------*
302. *--*
303. *-- 视图结构 `jxjf`*
304. *--*
305. DROP TABLE IF EXISTS `jxjf`;
306. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `jxjf`  AS SELECT `a`.`sno` AS `sno`, `a`.`sname` AS `sname`, `a`.`ssex` AS `ssex`, `b`.`reward` AS `reward`, `b`.`punishment` AS `punishment` FROM (`student` `a` join `stu\_reward\_punishment` `b` on((`a`.`sno` = `b`.`sno`))) WHERE (`a`.`ssex` = '女') ;
307. *-- --------------------------------------------------------*
308. *--*
309. *-- 视图结构 `jxjm`*
310. *--*
311. DROP TABLE IF EXISTS `jxjm`;
312. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `jxjm`  AS SELECT `a`.`sno` AS `sno`, `a`.`sname` AS `sname`, `a`.`ssex` AS `ssex`, `b`.`reward` AS `reward`, `b`.`punishment` AS `punishment` FROM (`student` `a` join `stu\_reward\_punishment` `b` on((`a`.`sno` = `b`.`sno`))) WHERE (`a`.`ssex` = '男') ;
313. *-- --------------------------------------------------------*
314. *--*
315. *-- 视图结构 `student\_view`*
316. *--*
317. DROP TABLE IF EXISTS `student\_view`;
318. CREATE ALGORITHM=UNDEFINED DEFINER=`smile`@`%` SQL SECURITY DEFINER VIEW `student\_view`  AS SELECT `account`.`username` AS `username`, `account`.`passward` AS `passward` FROM `account` ;
319. *--*
320. *-- 转储表的索引*
321. *--*
322. *--*
323. *-- 表的索引 `account`*
324. *--*
325. ALTER TABLE `account`
326. ADD PRIMARY KEY (`username`);
327. *--*
328. *-- 表的索引 `class\_stu\_num`*
329. *--*
330. ALTER TABLE `class\_stu\_num`
331. ADD PRIMARY KEY (`class`);
332. *--*
333. *-- 表的索引 `course`*
334. *--*
335. ALTER TABLE `course`
336. ADD PRIMARY KEY (`cno`);
337. *--*
338. *-- 表的索引 `department`*
339. *--*
340. ALTER TABLE `department`
341. ADD PRIMARY KEY (`dmpno`);
342. *--*
343. *-- 表的索引 `major`*
344. *--*
345. ALTER TABLE `major`
346. ADD PRIMARY KEY (`mno`,`mname`);
347. *--*
348. *-- 表的索引 `student`*
349. *--*
350. ALTER TABLE `student`
351. ADD PRIMARY KEY (`sno`);
352. *--*
353. *-- 表的索引 `stu\_course`*
354. *--*
355. ALTER TABLE `stu\_course`
356. ADD PRIMARY KEY (`sno`,`cno`);
357. *--*
358. *-- 表的索引 `teacher`*
359. *--*
360. ALTER TABLE `teacher`
361. ADD PRIMARY KEY (`tno`);
362. *--*
363. *-- 表的索引 `teacher\_course`*
364. *--*
365. ALTER TABLE `teacher\_course`
366. ADD PRIMARY KEY (`tno`,`cno`);
367. *--*
368. *-- 表的索引 `user`*
369. *--*
370. ALTER TABLE `user`
371. ADD PRIMARY KEY (`sno`),
372. ADD UNIQUE KEY `sno` (`sno`);
373. COMMIT;
374. */\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/*;
375. */\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/*;
376. */\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/*;

# 八、python代码块：

1. *# coding=utf-8*
2. *#!use/bin/env python3*
3. #使用环境：Linux--Ubuntu
4. import pymysql             *#使用pymysql板块*
5. import hashlib
6. import getpass             *#隐藏密码*
7. try:
8. conn = pymysql.connect(
9. host = '127.0.0.1',
10. user='smile',
11. passwd='1234',
12. db='python',
13. port= 3306,      *#端口号*
14. charset='utf8'   *#字符数*
15. )
16. print("-------------python-连接成功------------------------")
17. *#REVOKE ALL PRIVILEGES ON `python`.\* FROM 'smile'@'%';  ---给smile用户访问权限*
18. *#GRANT ALL PRIVILEGES ON `python`.\* TO 'smile'@'%' WITH GRANT OPTION;*
19. *#CREATE USER 'student'@'%' IDENTIFIED WITH mysql\_native\_password AS '\*\*\*';   创建student用户*
20. *#GRANT ALL PRIVILEGES ON \*.\* TO 'student'@'%' REQUIRE NONE WITH GRANT OPTION MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0 MAX\_USER\_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `python`.\* TO 'student'@'%';*
21. except pymysql.Error as e:
22. print("数据库连接失败",e)
23. *#e：打印数据库的错误原因  ,try检测异常*
24. cursor = conn.cursor()   *#使用cursor方式创建一个游标*
25. def get\_md5(data):          *#设置加密密耻*
26. obj = hashlib.md5("12:;idrsicxwersdfsaersdfsdfresdy54436jgfdsjdxff123ad".encode('utf-8'))
27. obj.update(data.encode('utf-8'))
28. result = obj.hexdigest()
29. return result                          *#使用hash对其加密*
30. name = input("请输入姓名 :")    *#实现密码的确认收集*
31. def password\_verification():
32. pas = getpass.getpass('请输入密码 :')
33. match = False   *#NULL*
34. while match is False:
35. while len(pas) < 8 or len(pas) > 15:
36. pas = getpass.getpass('输入的密码小于8位 - 请重新输入密码，它的长度在8~15位: ')
37. pas\_1 = getpass.getpass('确认您的密码 : ')
38. if pas\_1 == pas:
39. print('密码添加成功！')
40. match = True  *#添加成功后设置成turn退出循环*
41. else:
42. print('密码不匹配')
43. return pas
44. a=password\_verification()
45. pas = a
46. a=get\_md5(pas)   *#将a传入*
47. sql = "insert into user(sno, passwd) values ('%s','%s')" % (name,a)
48. try:
49. *# 执行sql语句*
50. cursor.execute(sql)
51. *# 提交到数据库执行*
52. conn.commit()
53. except:
54. *# 如果发生错误则回滚*
55. conn.rollback()
56. print("==================================================")
57. print("当前登陆的用户如下--密码使用非对称加密\n")
58. cursor.execute("select \* from user")
59. re = cursor.fetchall()
60. for row in re:
61. fname = row[0]
62. passwd = row[1]
63. date = row[2]
64. print("用户: %s --- 密码: %s ---时间: ---'%s'"%\
65. (fname,passwd,date))
66. print("==================================================")
67. print("==================================================")
68. print("||||你的姓名是： ",name)
69. print("||||hash密码是： ",a)
70. print("==================================================")
71. *#将用户插入到数据表user中*
72. try:
73. conn1 = pymysql.connect(
74. host = '127.0.0.1',
75. user='student',
76. passwd='11111111',
77. db='student',
78. port= 3306,      *#端口号*
79. charset='utf8'   *#字符数*
80. )
81. print("-------------student-连接成功---------------------")
82. except pymysql.Error as e:
83. print("student数据库连接失败",e)
84. cursor1 = conn1.cursor()   *#使用cursor方式创建一个游标*
85. try:
86. conn2 = pymysql.connect(
87. host = '127.0.0.1',
88. user='teacher',
89. passwd='22222222',
90. db='teacher',
91. port= 3306,      *#端口号*
92. charset='utf8'   *#字符数*
93. )
94. print("-------------teacher-连接成功---------------------")
95. except pymysql.Error as e:
96. print("teacher数据库连接失败",e)
97. cursor2 = conn2.cursor()   *#使用cursor方式创建一个游标*
98. try:
99. conn3 = pymysql.connect(
100. host = '127.0.0.1',
101. user='admin',
102. passwd='00000000',
103. db='admin',
104. port= 3306,      *#端口号*
105. charset='utf8'   *#字符数*
106. )
107. print("-------------admin-连接成功-----------------------")
108. except pymysql.Error as e:
109. print("admin数据库连接失败",e)
110. cursor3 = conn3.cursor()   *#使用cursor方式创建一个游标*
111. *#cursor.execute("PRAGMA foreign\_keys = ON")*
112. *#执行函数，强制外键约束*
113. *#使用execute方式执行SQL查询*
114. *#使用fetchone（）方式获取单条数据*
115. def login\_menu():  *#admin管理   shudent  :学生账户 passwd1234*
116. print('\n')
117. usage = ('\tA/a: 学生登录',
118. '\tB/b: 教师登录',
119. '\tC/c: 管理员登录',
120. '\tQ/q: 退出')
121. print('Login'.center(20, '='))
122. mycount()
123. for u in usage:    *#设置登陆的页面*
124. print(u)      *# 输出usage*
125. *#-----------------------------------------------------------#*
126. def stu\_login():        *#-学生登陆-*
127. username = input("输入账号：\t")
128. passward = getpass.getpass("输入密码：\t")
129. passward=get\_md5(passward)
130. sql\_username = 'select \* from user'  *#执行查询语句,查询用户信息表，登陆用户*
131. cursor1.execute(sql\_username)  *#execute执行sql的查询操作，查出对应账户和密码记录*
132. username\_list = cursor1.fetchall()
133. *#和fetchone不同的是，fetchone返回单个元组，单挑记录，而fetchall返回多条记录，存入list*
134. target = None       *# 设置为一个空对象，不同于None*
135. for rec in username\_list:    *#日常遍历*
136. if username in rec:   *#如果输入符合要求，则退出循环*
137. target = rec   *#相匹配跳出循环*
138. break
139. while True:
140. a = str(target[0])
141. a = get\_md5(a)
142. b = str(target[1])
143. b = get\_md5(b)
144. if username == target[0] and passward == b:
145. print("hash密码是：",b,"\t登陆成功！")
146. break
147. else:
148. print("账号或密码错误，请重新输入。")
149. username = input("输入账号：")
150. passward = getpass.getpass("输入密码：")
151. conn1.close()
152. def teach\_login():   *#设置教师的登陆界面 ，使用的数据库是teacher*
153. username = input("输入账号：\t")
154. passward = getpass.getpass("输入密码：\t")
155. passward = get\_md5(passward)            *#对密码加密*
156. sql\_username = 'select \* from user'
157. cursor2.execute(sql\_username)  *#将查询结果保存*
158. username\_list = cursor2.fetchall()  *#将返回记录保存在列表*
159. target = False  *# 对应客户的账号密码*
160. for rec in username\_list:
161. if username in rec:
162. target = rec
163. break
164. while True:
165. a = str(target[0])
166. a = get\_md5(a)
167. b = str(target[1])
168. b = get\_md5(b)
169. if username == target[0] and passward == b:
170. print("hash密码是：",b,"登陆成功！")
171. break
172. else:
173. print("账号或密码错误，请重新输入。")
174. username = input("输入账号:\t")
175. passward = getpass.getpass("输入密码：\t")
176. passward = get\_md5(passward)
177. conn2.close()
178. def admin\_login():    *#管理员登陆界面*
179. username = input("输入账号:\t")
180. passward = getpass.getpass("输入密码:\t")
181. passward = get\_md5(passward)
182. sql\_username = 'select \* from user'
183. cursor3.execute(sql\_username)
184. username\_list = cursor3.fetchall()
185. target = None  *# 对应客户的账号密码*
186. for rec in username\_list:
187. if username in rec:
188. target = rec
189. break
190. while True:
191. a = str(target[0])
192. a = get\_md5(a)
193. b = str(target[1])
194. b = get\_md5(b)
195. if username == target[0] and passward == b:
196. print("hash密码是：",b,"\t登陆成功！")
197. break
198. else:
199. print("账号或密码错误，请重新输入。")
200. username = input("输入账号：\t")
201. passward = getpass.getpass("输入密码:\t")
202. passward = get\_md5(passward)
203. admin\_menu()   *#调用admin菜单函数*
204. conn3.close()
205. *#------------------------------------------------------#*
206. def stu\_menu():  *#学生student*
207. stu\_usage = ('\tA/a: 学生信息查询',
208. '\tB/b: 学生课程信息查询',
209. '\tC/c: 奖惩信息查询',
210. '\tQ/q: 退出')
211. print("学生管理界面".center(20, '='))
212. for u in stu\_usage:
213. print(u)
214. def teach\_menu():
215. teach\_usage = ('\tA/a: 学生信息管理',
216. '\tQ/q: 退出')
217. print("教师管理界面".center(20, '='))
218. for u in teach\_usage:
219. print(u)
220. def admin\_menu():  *#管理员界面*
221. admin\_usage = ('\tA/a: 学生信息管理',
222. '\tB/b: 学生成绩（课程）管理',
223. '\tC/c: 奖惩信息管理',
224. '\tD/d: 更改密码',
225. '\tF/f: mysql-查询系统',
226. '\tE/e: 帮助信息',
227. '\tQ/q: 退出')
228. print("管理员界面".center(20, '='))
229. for u in admin\_usage:
230. print(u)
231. *#------------------------------------------------------#*
232. def manage\_table\_stu():  *#教师界面*
233. """学生信息管理"""
234. stu\_usage = ('\tA/a: 查询学生信息',
235. '\tB/b: 录入学生成绩',
236. '\tC/c: 更改学生成绩',
237. '\tD/d: 删除学生成绩',
238. '\tQ/q: 退出')
239. print("管理界面".center(20, '='))
240. for u in stu\_usage:
241. print(u)  *#遍历功能*
242. while True:
243. command = input('Please choose a command:\t')
244. if command in ('A', 'a'):
245. table\_stu\_select()
246. elif command in ('B', 'b'):
247. table\_stu\_insert()
248. elif command in ('C', 'c'):
249. table\_stu\_update()
250. elif command in ('D', 'd'):
251. table\_stu\_delete()
252. elif command in ('Q', 'q'):
253. break  *#直接退出*
254. def manage\_table\_stu\_course():  *#学生选课表的信息查询*
255. teach\_usage = ('\tA/a: 查询学生选课信息',
256. '\tB/b: 录入学生选课信息',
257. '\tC/c: 更改学生选课信息',
258. '\tD/d: 删除学生选课信息',
259. '\tM/m: 查询所有男生信息',
260. '\tF/f: 查询所有女生信息',
261. '\tALL/all: 查询所有学生信息',
262. '\tQ/q: 退出')
263. print("管理界面".center(20, '='))
264. for u in teach\_usage:
265. print(u)
266. while True:
267. command = input('Please choose a command: ')
268. if command in ('A', 'a'):
269. table\_stu\_course\_select()   *#函数调用*
270. elif command in ('B', 'b'):
271. table\_stu\_course\_insert()
272. elif command in ('C', 'c'):
273. table\_stu\_course\_update()
274. elif command in ('D', 'd'):
275. table\_stu\_course\_delete()
276. elif command in ('Q', 'q'):
277. break
278. *#--------------------------------------------------------------*
279. def manage\_table\_stu\_reward\_punishment():
280. usage = ('\tA/a: 查询奖惩信息',
281. '\tB/b: 录入奖惩信息',
282. '\tC/c: 更改奖惩信息',
283. '\tD/d: 删除奖惩信息',
284. '\tQ/q: 退出')
285. print("奖惩信息管理界面".center(20, '='))
286. for u in usage:
287. print(u)
288. while True:
289. command = input('Please choose a command:')
290. if command in ('A', 'a'):
291. table\_stu\_reward\_punishment\_select()
292. elif command in ('B', 'b'):
293. table\_stu\_reward\_punishment\_insert()
294. elif command in ('C', 'c'):
295. table\_stu\_reward\_punishment\_update()
296. elif command in ('D', 'd'):
297. table\_stu\_reward\_punishment\_delete()
298. elif command in ('Q', 'q'):
299. break
300. else:
301. print("请输入合法的选项。")
302. input("按下回车以继续: ")
303. def table\_stu\_select():
304. """查询学生表中指定学生的所有信息"""
305. stu\_id = input('请输入你的学号：')
306. sql = "select \* from student where sno = '%s'"% (stu\_id) *#注意%s需要'%s' ---使用正则表达式*
307. cursor.execute(sql)   *#执行sql语句*
308. list = cursor.fetchall()
309. if not list:
310. print('\tDatabase has no record at this time.')
311. else:
312. for rec in list:
313. print(rec)
314. input("按下回车以继续: ")
315. def table\_stu\_insert():
316. """往student表中新增数据"""
317. while True:
318. sno = input('输入学号（输入0以结束）: ')
319. if sno == '0':
320. break
321. sname = input('输入姓名: ')
322. ssex = input('输入性别（男/女）: ')        *#*
323. while ssex not in ('男', '女'):
324. ssex = input("请输入正确的性别（男/女）：")
325. sage = input('输入年龄: ')
326. sclass = input('输入班级: ')
327. intime = input('输入入学时间--注意格式datebase: ')
328. sql = "insert into student values ('%s','%s','%s','%s','%s', '%s')" % (sno, sname, ssex, sage, sclass, intime)
329. cursor.execute(sql)
330. conn.commit()
331. print("添加成功")
332. sno = None
333. def table\_stu\_update():
334. while True:
335. sno = input('输入需要修改信息的学生的学号（输入0以结束): ')
336. if sno == '0':
337. break
338. print("修改开始")
339. sname = input('输入姓名: ')
340. ssex = input('输入性别（男/女）: ')
341. while ssex not in ('男', '女'):
342. ssex = input("请输入正确的性别（男/女）：")
343. sage = input('输入年龄: ')
344. sclass = input('输入班级: ')
345. intime = input('输入入学时间: ')
346. sql = """update student set
347. sname='%s', ssex='%s', sage=%s, sclass='%s', intime='%s'
348. where sno=%s""" % (sname, ssex, sage, sclass, intime, sno)
349. cursor.execute(sql)
350. conn.commit()
351. print("修改成功")
352. def table\_stu\_delete():
353. while True:
354. sno = input('输入要删除的信息对应的学号（输入0以结束）: ')
355. if sno == '0':
356. break
357. sql = "delete from student where sno = '%s'" % sno
358. cursor.execute(sql)
359. conn.commit()
360. print("删除成功")
361. def table\_stu\_course\_select():      *#查找学生信息*
362. stu\_id = input('请输入你的学号：')
363. sql = 'select cname from course where cno in (select cno from stu\_course where sno = %s)' % (stu\_id)
364. cursor.execute(sql)
365. list = cursor.fetchall()
366. if not list:
367. print('暂无此记录')
368. input("按下回车以继续: ")
369. else:
370. for rec in list:
371. print(rec)
372. input("按下回车以继续: ")
373. def table\_stu\_course\_insert():    *#插入课表*
374. while True:
375. sno = input('输入学号（输入0以结束）: ')
376. if sno == '0':
377. break
378. cno = input("输入要选修的课程号")
379. list = table\_course\_select()
380. cno\_list = []
381. for rec in list:
382. cno\_list.append(rec[0])
383. while cno not in cno\_list:
384. cno = input("没有这门课，请输入正确的课程编号（输入“q”以退出）：")
385. if cno == 'q':
386. break
387. grade = input("输入这位学生该门课程的成绩（若暂无，请输入”n“）：")
388. if grade == 'n':
389. grade = ''
390. sql = "insert into stu\_course(sno, cno, grade) values ('%s','%s','%s')" % (
391. sno, cno, grade)
392. cursor.execute(sql)
393. conn.commit()            *#提交事务*
394. print("添加成功")
395. def table\_stu\_course\_update():
396. while True:
397. sno = input('输入需要修改信息的学生的学号（输入0以结束): ')
398. if sno == '0':
399. break
400. cno = input('输入该学生所选的课程编号: ')
401. print("修改开始")
402. grade = input("输入该学生该门课成绩（若暂无，请输入”n“）：")
403. if grade == 'n':
404. grade = ''
405. sql = """update stu\_course set grade=%s
406. where sno='%s' and cno='%s'""" % (grade, sno, cno)
407. cursor.execute(sql)
408. conn.commit()
409. print("修改成功")
410. def table\_stu\_course\_delete():         *# 删除学生的信息*
411. while True:
412. sno = input('输入要删除的学生对应的学号（输入0以结束）: ')
413. if sno == '0':
414. break
415. cno = input('输入该学生所选的课程编号: ')
416. sql = "delete from stu\_course where sno = '%s' and cno = '%s'" % (sno, cno)
417. cursor.execute(sql)
418. conn.commit()         *#提交修改,事务，提交*
419. print("删除成功")
420. def table\_stu\_reward\_punishment\_select():   *#学生的奖惩*
421. cursor = conn.cursor()
422. stu\_id = input('请输入学号：')     *#查询奖赏的表*
423. sql = "select \* from stu\_reward\_punishment where sno ='%s'" % (stu\_id)
424. cursor.execute(sql)
425. list = cursor.fetchall()
426. if not list:
427. print('该学生暂无奖惩记录')
428. input("按下回车以继续: ")
429. else:
430. reward = []
431. punishment = []
432. for rec in list:
433. if rec[2] != None:
434. reward.append(rec[2])
435. if rec[3] != None:
436. punishment.append(rec[3])
437. if reward:      *#将空的列表来接收添加的数据，添加的数据判断*
438. print('奖项: ', reward)
439. if punishment:
440. print('惩罚: ', punishment)
441. input("按下回车回到管理界面: ")
442. def table\_stu\_reward\_punishment\_insert():
443. while True:
444. sno = input('输入学号（输入0以结束）: ')
445. if sno == '0':
446. break
447. reward = input("请输入该学生的奖励信息（若无，请输入”n“）：")
448. punishment = input("请输入该学生的惩罚信息（若无，请输入”n“）：")
449. if reward == 'n':
450. reward = ''
451. if punishment == 'n':
452. punishment = ''
453. sql = "insert into stu\_reward\_punishment(sno, reward, punishment) values ('%s','%s','%s')" % (sno, reward, punishment)
454. cursor.execute(sql)
455. conn.commit()
456. print("添加成功")
457. def table\_stu\_reward\_punishment\_update():
458. while True:
459. sno = input('输入需要修改信息的学生的学号（输入0以结束): ')
460. if sno == '0':
461. break
462. print("修改开始")
463. reward = input("请输入该学生的奖励信息（若无，请输入”n“）：")
464. punishment = input("请输入该学生的惩罚信息（若无，请输入”n“）：")
465. if reward == 'n':
466. reward = ''
467. if punishment == 'n':
468. punishment = ''
469. sql = """update stu\_reward\_punishment set
470. reward = '%s', punishment = '%s'
471. where sno = '%s'""" % (reward, punishment, sno)
472. cursor.execute(sql)
473. conn.commit()
474. print("修改成功")
475. def table\_stu\_reward\_punishment\_delete():
476. while True:
477. sno = input('输入要删除的信息对应的学号（输入0以结束）: ')
478. if sno == '0':
479. break
480. flag = input("请输入要删除奖励（输入”1“）还是惩罚（输入”2“）？: ")
481. if flag == '1':
482. sql = "delete from stu\_reward\_punishment where reward is not null and sno = %s" % sno
483. if flag == '2':
484. sql = "delete from stu\_reward\_punishment where punishment is not null and sno = %s" % sno
485. else:
486. while flag not in ('1', '2'):
487. flag = input("请输入合法的选项：")
488. cursor.execute(sql)
489. conn.commit()
490. print("删除成功")
491. def table\_course\_select():
492. sql = 'select \* from course'
493. cursor.execute(sql)
494. list = cursor.fetchall()
495. return list
496. def table\_account\_update():
497. while True:
498. username = input('输入需要修改的账号（输入0以结束): ')
499. if username == '0':
500. break
501. print("修改开始")
502. passward = input('输入新密码: ')
503. sql = """update account set passward=%s
504. where username='%s'""" % (passward, username)
505. cursor.execute(sql)
506. conn.commit()
507. print("修改成功")
508. def update\_passward():  *#管理员用户修改密码*
509. while True:
510. usage = ('\tA/a: 修改密码',
511. '\tQ/q: 退出')
512. print("更改密码".center(20, '='))
513. for u in usage:
514. print(u)
515. command = input('Please choose a command:\t')
516. if command in ('A', 'a'):
517. table\_account\_update()
518. input("按下回车回到管理界面: ")
519. elif command in ('Q', 'q'):
520. break
521. else:
522. print("请输入合法的选项。")
523. input("按下回车以继续: ")
524. def find\_admin():    *#针对管理员的查询系统*
525. while True:
526. usage = ('\tM/m: 查询所有男生信息',
527. '\tF/f: 查询所有女生信息',
528. '\tA/a: 查询所有学生信息',
529. '\tT/t: 查询所有老师的信息',
530. '\tQ/q: 退出查询')
531. print("学生信息查询系统".center(20,'='))
532. for u in usage:
533. print(u)
534. command = input('Please choose a command:\t')
535. if command in ('T', 't'):
536. sql = 'select distinct teacher.tno,teacher\_course.cno,teacher.tsex,\
537. teacher.tphone,teacher.dmpno,teacher.profess\
538. from teacher join teacher\_course where teacher\_course.tno = teacher.tno'
539. *#查询老师信息，跨数据库查询，且distinct显示教师号唯一*
540. cursor.execute(sql)
541. print("="\*20)
542. print("当前的教师信息如下：\n")
543. re = cursor.fetchall()
544. for row in re:
545. tno = row[0]  *#姓名*
546. cno = row[1]
547. tsex = row[2]
548. tphone = row[3]
549. dmpon = row[4]
550. prosess = row[5]
551. print("教师编号：%s --,教授课程: %s --,性别：%s --,电话号码：%s --,dmpno: %s --,职称：%s "%\
552. (tno,cno,tsex,tphone,dmpon,prosess))
553. input("按下回车回到管理界面: ")
554. elif command in ('M', 'm'):
555. sql = 'select \* from jxjm'
556. *#创建视图jxj create view jxj as select a.sno,a.sname,a.ssex,b.reward,b.punishment*
557. *#from student a join stu\_reward\_punishment b on a.sno = b.sno while a.ssex = '男';*
558. *#查询所有男生信息信息，跨数据库查询*
559. cursor.execute(sql)
560. print("="\*20)
561. print("当前的男生信息如下：\n")
562. re = cursor.fetchall()
563. for row in re:
564. tno = row[0]  *#姓名*
565. cno = row[1]
566. tsex = row[2]
567. tphone = row[3]
568. dmpon = row[4]
569. print("学号：%s --,姓名: %s --,性别：%s --,奖励：%s --,惩罚: %s"%\
570. (tno,cno,tsex,tphone,dmpon))
571. input("按下回车回到管理界面: ")
572. elif command in ('F', 'f'):
573. sql = 'select \* from jxjf'
574. *#创建视图jxj create view jxj as select a.sno,a.sname,a.ssex,b.reward,b.punishment*
575. *#from student a join stu\_reward\_punishment b on a.sno = b.sno while a.ssex = '女';*
576. *#查询所有男生信息信息，跨数据库查询*
577. cursor.execute(sql)
578. print("="\*20)
579. print("当前的女生信息如下：\n")
580. re = cursor.fetchall()
581. for row in re:
582. tno = row[0]  *#姓名*
583. cno = row[1]
584. tsex = row[2]
585. tphone = row[3]
586. dmpon = row[4]
587. print("学号：%s --,姓名: %s --,性别：%s --,奖励：%s --,惩罚: %s"%\
588. (tno,cno,tsex,tphone,dmpon))
589. input("按下回车回到管理界面: ")
590. elif command in ('A', 'a'):
591. sql = 'select \* from jxj'
592. *#创建视图jxj create view jxj as select a.sno,a.sname,a.ssex,b.reward,b.punishment*
593. *#from student a join stu\_reward\_punishment b on a.sno = b.sno;*
594. *#查询所有信息，跨数据库查询*
595. cursor.execute(sql)
596. print("="\*20)
597. print("当前的所有学生信息如下：\n")
598. re = cursor.fetchall()
599. for row in re:
600. tno = row[0]  *#姓名*
601. cno = row[1]
602. tsex = row[2]
603. tphone = row[3]
604. dmpon = row[4]
605. print("学号：%s --,姓名: %s --,性别：%s --,奖励：%s --,惩罚: %s"%\
606. (tno,cno,tsex,tphone,dmpon))
607. input("按下回车回到管理界面: ")
608. elif command in ('Q', 'q'):
609. break
610. else:
611. print("="\*20)
612. print("请输入合法的选项。")
613. input("按下回车以继续: ")
614. *#统计学生人数功能：*
615. def mycount():
616. sql = 'select count(sno) from student'
617. cursor.execute(sql)
618. count = cursor.fetchall()   *#取一条数据*
619. count = count[0]
620. print("当前学生人数： %s"%count)
621. *#主函数运行函数*
622. def main():
623. while True:
624. login\_menu() *# 运行登录界面*
625. command = input('Please choose a command:')
626. if command in ('A', 'a'):
627. """以学生身份登录"""
628. stu\_login()
629. while True:
630. stu\_menu()
631. command1 = input('Please choose a command:\t')
632. if command1 in ('A', 'a'):
633. table\_stu\_select()
634. elif command1 in ('B', 'b'):
635. table\_stu\_course\_select()
636. elif command1 in ('C', 'c'):
637. table\_stu\_reward\_punishment\_select()
638. elif command1 in ('Q', 'q'):
639. break
640. else:
641. print("请输入合法的选项。")
642. input("按下回车以继续: ")
643. elif command in ('B', 'b'):
644. *#以老师身份登陆*
645. teach\_login()
646. while True:
647. teach\_menu()
648. command4 = input('Please choose a command:')
649. if command4 in ('A', 'a'):
650. manage\_table\_stu()
651. elif command4 in ('Q', 'q'):
652. break
653. else:
654. print("请输入合法的选项。")
655. input("按下回车以继续: ")
656. elif command in ('C', 'c'):
657. *#以管理员身份登陆*
658. admin\_login()
659. while True:
660. command2 = input('Please choose a command:')
661. if command2 in ('A', 'a'): *#*
662. manage\_table\_stu()
663. elif command2 in ('B', 'b'):
664. manage\_table\_stu\_course()
665. elif command2 in ('C', 'c'):
666. manage\_table\_stu\_reward\_punishment()
667. elif command2 in ('D', 'd'):
668. update\_passward()
669. elif command2 in ('F','f'):
670. *#统计学生人数*
671. find\_admin()
672. elif command2 in ('E', 'e'):
673. print("请访问百度查找问题！")
674. elif command2 in ('Q', 'q'):
675. break
676. else:
677. print("请输入合法的选项。")
678. input("按下回车以继续: ")
679. elif command in ('Q', 'q'):
680. print("good bey!")
681. *#关闭游标*
682. cursor.close()
683. *#关闭连接*
684. conn.close()
685. break
686. else:
687. print("请输入合法的选项。")
688. input("按下回车以继续: ")
689. main()