

## 1、描述

学院通过学院名称标识，数据库中记录学院名称和院长。

每个学院有多名教师，教师可以通过他们的职工号进行标识，数据库中记录教师的姓名和职工号。每个教师只能在一个学院工作。

学院下设若干本科专业，本科专业通过专业代码进行标识，数据库中记录专业代码和专业名称。每个本科专业只能归属一个学院。

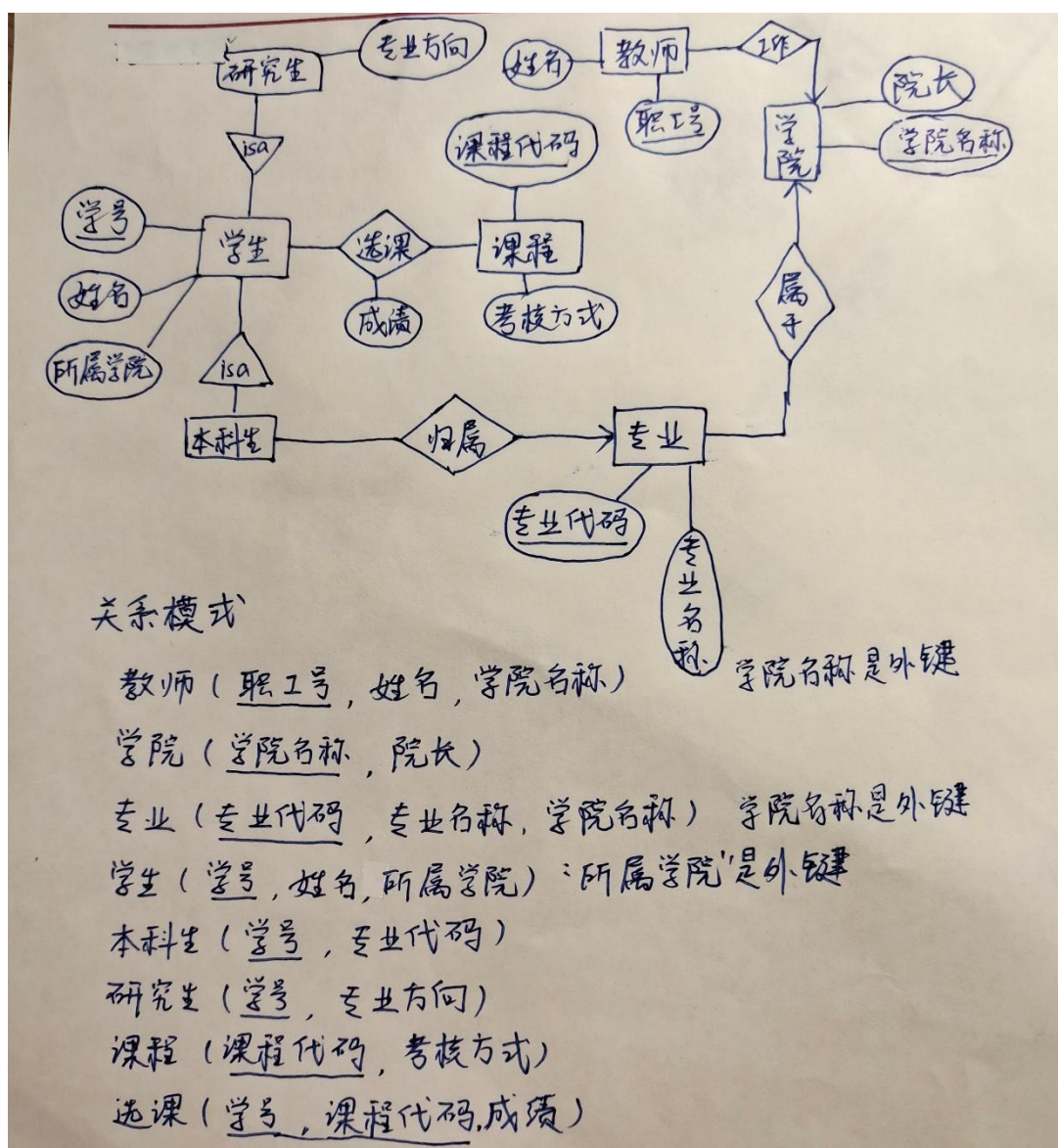
本科生通过学号进行标识，数据库中记录本科生的姓名、学号和所属学院。每个本科生只能归属于一个本科专业。

研究生通过学号进行标识，数据库中要录研究生的姓名、学号、所属学院和专业方向。

所有学生须修读一定课程，课程通过课程代码标识，数据库中记录课程代码和考核方式，学生通过选课系统选课，每名学生可选多门课程，并记录分数。

2、

a) 和 b) ER 图和关系模式



c) 用 SQL 语句创建上述关系模式。

```

CREATE TABLE Teacher(//1、教师表 Teacher
Tname CHAR(20),
Tnum CHAR(9) PRIMARY KEY, //职工号是主键
Tcolname CHAR(20),
//Tcolname是外键，被参照的是Collage表，被参照的列是Cname
FOREIGN KEY (Tcolname) REFERENCES Collage(Cname)
);
CREATE TABLE Collage(//2、学院表 Collage
Cname CHAR(30) PRIMARY KEY, //学院名称是主键
Cdean CHAR(10) //院长
);
CREATE TABLE Major(//3、专业表 Major
Mnum CHAR(9) PRIMARY KEY, //专业代码是主键
Mname CHAR(30),
Mcolname CHAR(30),
//Mcolname是外键，被参照的是Collage表，被参照的列是Cname
FOREIGN KEY (Mcolname) REFERENCES Collage(Cname)
);
CREATE TABLE Student(//4、学生表 Student
Snum CHAR(9) PRIMARY KEY, //学号是主键
Sname CHAR(20);
Scolname CHAR(30),
//Scolname是外键，被参照的是Collage表，被参照的列是Cname
FOREIGN KEY (Scolname) REFERENCES Collage(Cname)
);
CREATE TABLE Undergraduate(//5、本科生表 Undergraduate
Unum CHAR(9) PRIMARY KEY, //学号是主键
Umajornum CHAR(9)
);
CREATE TABLE Postgraduate(//6、研究生表 Postgraduate
Pnum CHAR(9) PRIMARY KEY, //学号是主键
Pmajorfield CHAR(20)
);
CREATE TABLE Course(//7、课程表 Course
Coursenum CHAR(9) PRIMARY KEY, //课程代码是主键
ExamineCHAR(20) //考核方式
);
CREATE TABLE SelectCourse(//8、选课表 SelectCourse
Studentnum CHAR(9),
Coursenum CHAR(9),
//"Studentnum"和"Coursenum"共同构成主键
PRIMARY KEY(Studentnum, Coursenum),
Score INT
);

```

d) 给出该数据库模式上 5 个查询语句样例

//1、单表查询, 查询化学学院的学生姓名

```
select Sname
from Student
where Scolname = '化学学院';
```

//2、多表连接查询, 查询同名教师的职工号对

```
select t1.Tnum, t2.Tnum
from Teacher t1, Teacher t2
where t1.Tname = t2.Tname;
```

//3、嵌套查询, 查询高数100分的学生学号和其他课程成绩

```
select *
from SelectCourse
where Studentnum in (select Studentnum
                      from SelectCourse
                      where Coursenum = 'gaoshu' and Score = '100';
                      );
```

//4、 EXISTS查询

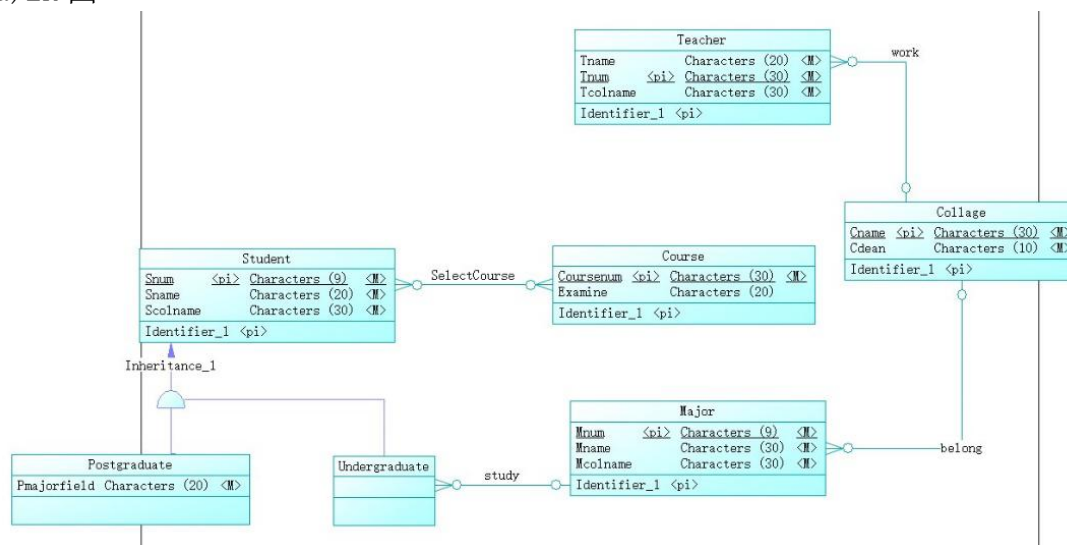
```
select Snum
from Student s1
where not exists (select *
                  from Student
                  where Scolname = s1.Scolname and Snum <> s1.Snum;
                  );
```

//5、聚合操作查询, 查询高数平均成绩

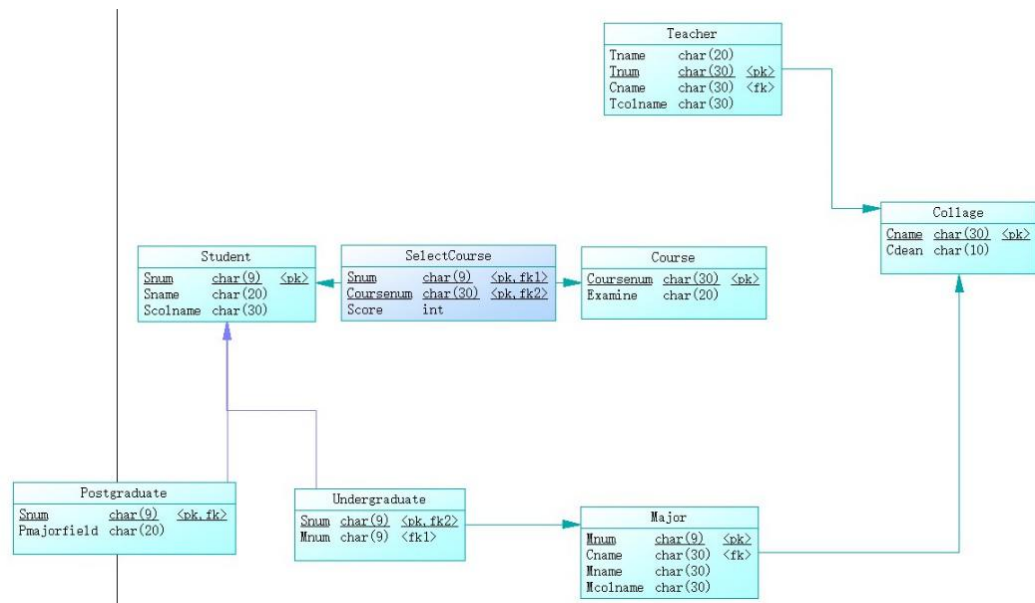
```
select avg(Score)
from SelectCourse
where Coursenum = 'gaoshu';
```

3、

a) ER 图



b) 关系模型图



c) 使用 PowerDesigner 工具，生成创建数据库的 SQL 语句。截图如下：



```

/*=====
/* DBMS name:    MySQL 5.0                */
/* Created on:   2022/4/5 20:11:31        */
/*=====

```

```

drop table if exists Collage;

drop table if exists Course;

drop table if exists Major;

drop table if exists Postgraduate;

drop table if exists SelectCourse;

drop table if exists Student;

drop table if exists Teacher;

drop table if exists Undergraduate;

```

```

/*=====
/* Table: Collage                */
/*=====
create table Collage
(
    Cname          char(30) not null,
    Cdean          char(10) not null,
    primary key (Cname)
);

```

```

/*=====
/* Table: Course                */
/*=====
create table Course
(
    Coursenum      char(30) not null,
    Examine        char(20),
    primary key (Coursenum)
);

```

```

/*=====
/* Table: Major                */
/*=====
create table Major
(
    Mnum           char(9) not null,
    Cname          char(30),
    Mname          char(30) not null,
    Mcolname       char(30) not null,
    primary key (Mnum)
);

```

```

/*=====
/* Table: Postgraduate         */
/*=====
create table Postgraduate
(
    Snum           char(9) not null,
    Pmajorfield    char(20) not null,
    primary key (Snum)
);

```

```

/*=====
/* Table: SelectCourse */
/*=====
create table SelectCourse
(
    Snum          char(9) not null,
    Coursenum     char(30) not null,
    Score         int,
    primary key (Snum, Coursenum)
);

/*=====
/* Table: Student */
/*=====
create table Student
(
    Snum          char(9) not null,
    Sname         char(20) not null,
    Scolname      char(30) not null,
    primary key (Snum)
);

/*=====
/* Table: Teacher */
/*=====
create table Teacher
(
    Tname         char(20) not null,
    Tnum          char(30) not null,
    Cname         char(30),
    Tcolname      char(30) not null,
    primary key (Tnum)
);

/*=====
/* Table: Undergraduate */
/*=====
create table Undergraduate
(
    Snum          char(9) not null,
    Mnum          char(9),
    primary key (Snum)
);

```

```

alter table Major add constraint FK_belong foreign key (Cname)
references Collage (Cname);

```

```

alter table Postgraduate add constraint FK_Inheritance_2 foreign key (Snum)
references Student (Snum);

```

```

alter table SelectCourse add constraint FK_SelectCourse foreign key (Snum)
references Student (Snum);

```

```

alter table SelectCourse add constraint FK_SelectCourse2 foreign key (Coursenum)
references Course (Coursenum);

```

```

alter table Teacher add constraint FK_work foreign key (Cname)
references Collage (Cname);

```

```

alter table Undergraduate add constraint FK_Inheritance_1 foreign key (Snum)
references Student (Snum);

```

```

alter table Undergraduate add constraint FK_study foreign key (Mnum)
references Major (Mnum);

```

#### 4. 分析比较采用上述两种方法

a) 两种关系模式的设计是否存在差异？如有差异，这种差异是否对后期的实现带来不同的影响？

有差异。

因为在使用 **PowerDesigner** 工具生成关系模式图时，是先画出 **ER** 图，然后将 **ER** 图转为关系模型图。所以 **PowerDesigner** 生成的关系模式给 **Teacher**、**Major** 表自动添加了外键。

对后期实现没有影响。

b) **PowerDesigner** 工具生成的 **SQL** 语句有什么样的特点？为什么会出现一些附加语句？它的作用是什么？

特点：

关键字都是小写；在表的最后说明主键；没有说明外键。

附加语句：

在开始出现的“`drop table if exists Course;`”语句，如果存在一个 **Course** 表，就把它删除。防止出现两个同名的表。

结尾的“`alter table Major add constraint FK_belong foreign key (Cname)references Collage (Cname);`”语句，表明为 **Major** 表添加了外键约束。