#### 大学的信息管理系统的详细需求描述：

大学有许多学院，每个学院通过学院名称标识；

每个学院有多名教师，但他们其中只有1人可以担任院长；

教师可以通过他们的职工号进行标识，数据库中要记录每个教师的姓名、职称和工资，每个教师只能归属一个学院，院长同样只能归属一个学院，并负责管理他所在的学院；

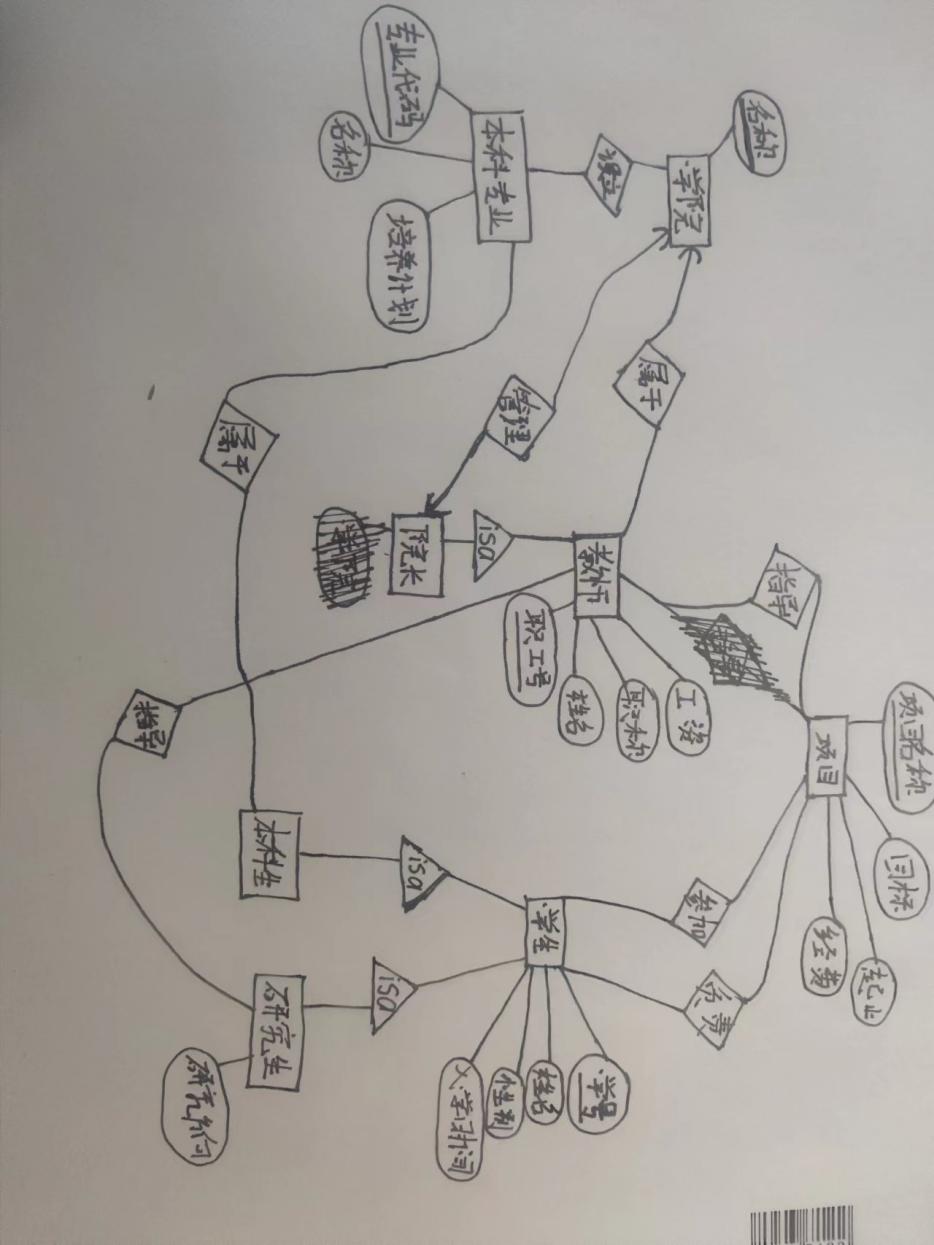
学院下设若干本科专业，本科专业通过专业代码进行标识，每个专业要有专业名称和培养目标，每个本科专业只能归属一个学院。

学院培养本科生，本科生通过他们的学号进行标识，数据库库中要记录本科生的姓名、性别和入学时间。每个本科生只能归属于一个本科专业。

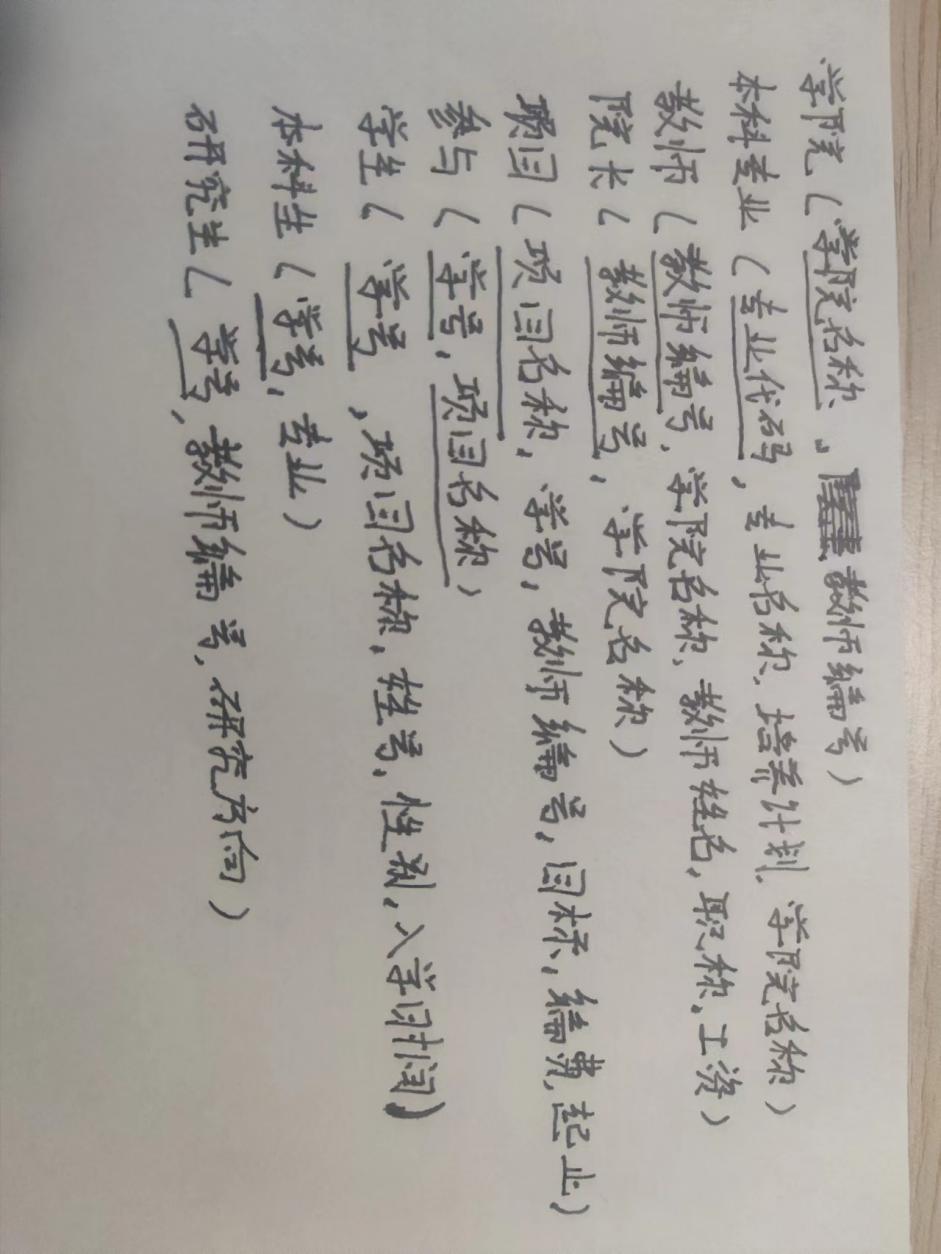
学院培养研究生，研究生通过他们的学号进行标识，数据库库中要记录研究生的姓名、性别、入学时间和专业方向。每位研究生必须有一位导师进行指导。

大学设立项目经费，鼓励学生在学期间申请科技创新项目，具体要求为：每个项目必须有名称、目标、起止时间和经费数额，每个项目可以由多个学生参加，但必须明确其中一名学生为项目负责人，必须有一名指导教师。学校允许一名教师指导多个项目，也允许一名学生参加多个项目，但不允许一名学生做多个项目的负责人。

#### **概念模型ER图：**



#### ER图转换成关系模式：



#### Sql语句创建关系模式：

create table college

(

CollegeName char(256) not null,

TeacherIndex numeric(8,0) not null,

primary key (CollegeName)

);

create table dean

(

TeacherIndex numeric(8,0) not null,

CollegeName char(256) not null,

primary key (TeacherIndex)

);

create table graduate

(

StudentIndex numeric(8,0) not null,

TeacherIndex numeric(8,0) not null,

direction char(256),

primary key (StudentIndex)

);

create table "join"

(

StudentIndex numeric(8,0) not null,

ProjectName char(256) not null,

primary key (StudentIndex, ProjectName)

);

create table majior

(

MajorIndex numeric(8,0) not null,

CollegeName char(256) not null,

MajorName char(256),

TrainingPlan char(256),

primary key (MajorIndex)

);

create table project

(

ProjectName char(256) not null,

StudentIndex numeric(8,0),

TeacherIndex numeric(8,0) not null,

target char(256),

appropriation float,

time float,

primary key (ProjectName)

);

create table student

(

StudentIndex numeric(8,0) not null,

ProjectName char(256),

StudentName char(256),

gender char(256),

RuxueTime numeric(8,0),

Attribute\_17 char(10),

primary key (StudentIndex)

);

create table teacher

(

TeacherIndex numeric(8,0) not null,

CollegeName char(256),

TeacherName char(256),

Job char(256),

wage float,

primary key (TeacherIndex)

);

create table undergraduate

(

StudentIndex numeric(8,0) not null,

MajorIndex numeric(8,0) not null,

primary key (StudentIndex)

);

alter table college add constraint FK\_administer2 foreign key (TeacherIndex)

references dean (TeacherIndex);

alter table dean add constraint FK\_Inheritance\_1 foreign key (TeacherIndex)

references teacher (TeacherIndex);

alter table dean add constraint FK\_administer foreign key (CollegeName)

references college (CollegeName);

alter table graduate add constraint FK\_Inheritance\_3 foreign key (StudentIndex)

references student (StudentIndex);

alter table graduate add constraint FK\_guide2 foreign key (TeacherIndex)

references teacher (TeacherIndex);

alter table "join" add constraint FK\_join foreign key (StudentIndex)

references student (StudentIndex);

alter table "join" add constraint FK\_join2 foreign key (ProjectName)

references project (ProjectName);

alter table majior add constraint FK\_belongsto2 foreign key (CollegeName)

references college (CollegeName);

alter table project add constraint FK\_Beresponsible2 foreign key (StudentIndex)

references student (StudentIndex);

alter table project add constraint FK\_guide foreign key (TeacherIndex)

references teacher (TeacherIndex);

alter table student add constraint FK\_Beresponsible foreign key (ProjectName)

references project (ProjectName);

alter table teacher add constraint FK\_Belongsto1 foreign key (CollegeName)

references college (CollegeName);

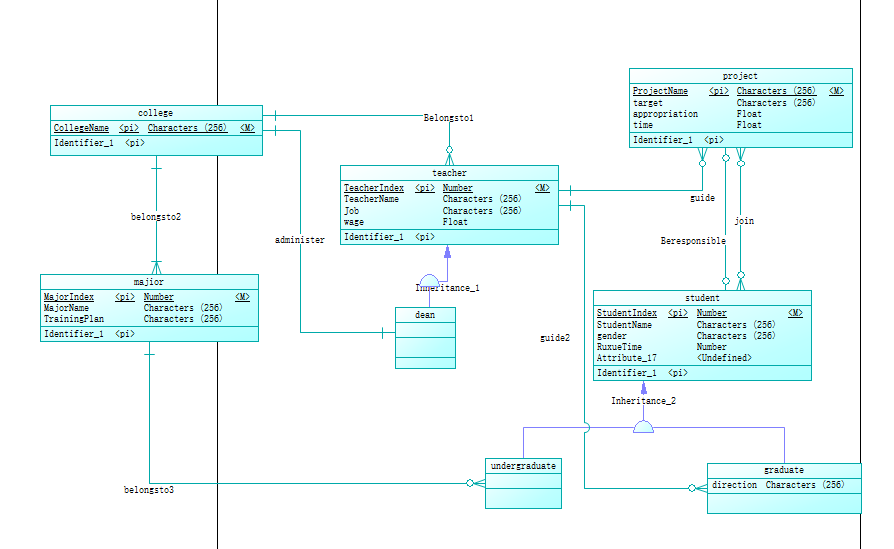
alter table undergraduate add constraint FK\_Inheritance\_2 foreign key (StudentIndex)

references student (StudentIndex);

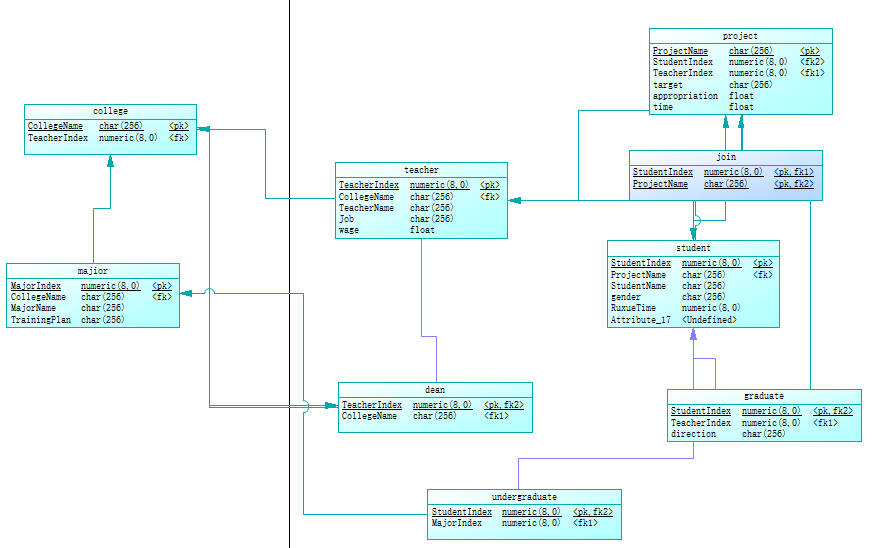
alter table undergraduate add constraint FK\_belongsto3 foreign key (MajorIndex)

references majior (MajorIndex);

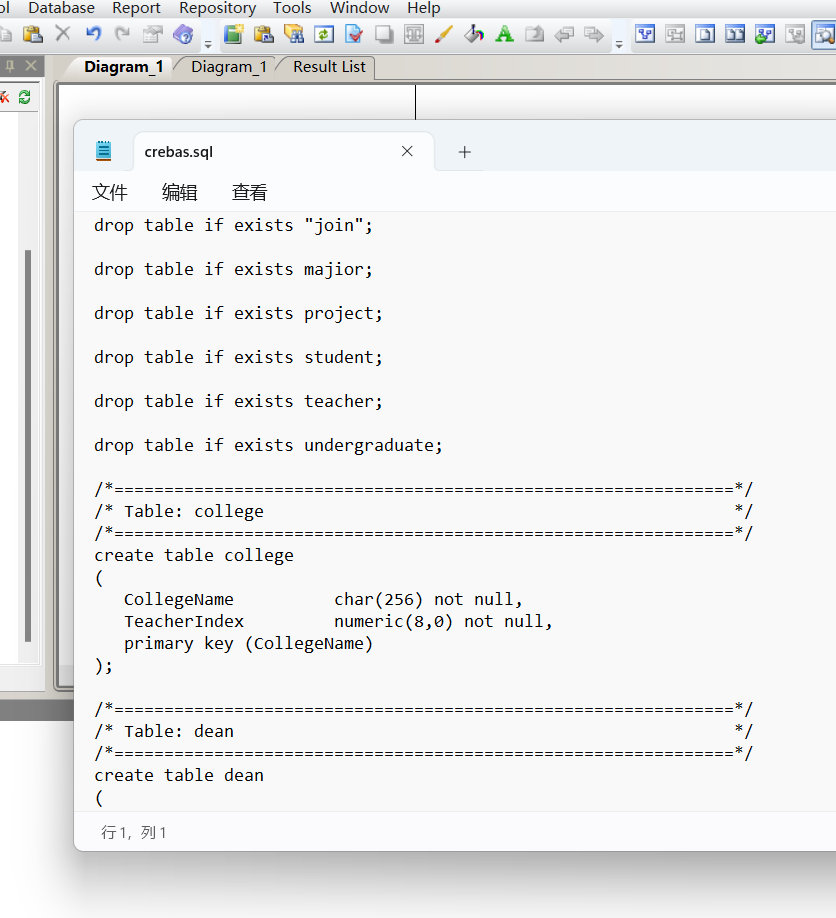
#### PowerDesign的ER图：



#### PowerDesign的关系模型：



#### PowerDesign生成创建数据库的SQL语句



#### 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？

当两个实体间有附加属性时，以ER图转为关系模式的方法，最终只会将关系转为一张表，而powerdesigner在多对多时将关系转为三张表。这种差异可能最终导致后期查询时powerdesigner产生的表的数量过多，且查询效率较低，需要查询多张表才能得到结果；而ER转为关系模式只将关系转为了一张表，查询效率较高。同时powerdesigner对外键有着明显的约束，即对外键的更新和删除受到限制，对后期的修改有着明显的影响。

#### 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？

powerdesigner工具生成的sql语句拥有大量注释语句，并且主键的声明都在表内声明，外键在表外统一添加。

为什么出现附加语句：出现的注释语句可以展现建立表的步骤。在声明外键时出现的附加语句是为了保证实体完整性，对外键的更新和删除进行限制来保证实体完整性。