1. 设有如图所示的关系S、SC和C，试用关系代数表达式表示下列查询语句。

|  |  |  |
| --- | --- | --- |
| S | C | SC |
| |  |  |  |  | | --- | --- | --- | --- | | S# | SNAME | AGE | SEX | | 1 | 李强 | 23 | 男 | | 2 | 刘丽 | 22 | 女 | | 5 | 张友 | 22 | 男 | | |  |  |  | | --- | --- | --- | | C# | CNAME | TEACHER | | k1 | C语言 | 王华 | | k5 | 数据库原理 | 程军 | | k8 | 编译原理 | 程军 | | |  |  |  | | --- | --- | --- | | S# | C# | GRADE | | 1 | k1 | 83 | | 2 | k1 | 85 | | 5 | k1 | 92 | | 2 | k5 | 90 | | 5 | k5 | 84 | | 5 | k8 | 80 | |

1. 查询“李强”同学的考试成绩，输出学号、姓名、课程号和成绩

**Πs#,sNAME,c#,grade(σsname ='李强'(S∞SC))**

2. 查询选修“C语言”课程的学生的学号

**Πs# (σcname ='c语言'(C∞SC))**

3. 查询选修“程军”老师课程的学生的考试情况，输出学号、课程号和成绩

**Πs#,c#,grade (σteacher ='程军'(C∞SC))**

4. 查询“数据库原理”课程考试不及格的学生，输出学号和姓名

**Πs#,sname (σcname ='数据库原理' and grade<60(S∞SC∞C))**

2. 电影数据库模式：

Movies(title,year,length,genre,studioName,producerC)

MovieStar(name, address, gender, birthdate)

StarsIn(movieTitle, movieYear, starName)

MovieExec(name, address, cert, netWorth)

Studio(name, address, presC)

1. 创建MovieStar表.

create table MovieStar(name char(20) primary key,

address char(50),

gender char(1),

birthdate date);

2. 在MovieStar表中插入一条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 周润发 | 香港市何山路88号 | 男 | 1955-05-18 |

Insert into MovieStar values (**'**周润发**'**, **'**香港市何山路88号**'**, **'**男**'**,date **'**1955-05-18**'**);

3. 用alter table命令增加自定义约束con，要求Movies表里电影的年份必须在1900之后

alter table Movies add con check(year>=1900)

4. 创建视图Viewstar，用于保存参演电影“Titanic”的演员信息，包括姓名、性别和出生日期。

create view Viewstar as

select name,gender,birthdate

from MovieStar

where name in(select starName from StarsIn

where movieTitle =’ Titanic’);

5. 查询姓名中带有“明”字的男影星。

select \*

from moviestar

where name like '%明%';

6. 统计各电影公司经理的净资产值，并按资产值的降序排序。

7. 统计每个电影公司投资拍摄的电影数量。

select studioName,count(\*) as 电影数量

from Movies

group by studioName；

8. 查询与成龙合作过的女影星。（子查询）

Select name

From MovieStar

Where gender=‘F’ and name IN

( Select starName

From StarsIn

Where movieTitle IN

( select movieTilte

from StarsIn

where starName=‘成龙’ ));