

数据库系统

实验指导书

班级： 2006101

学号： 2200600226

姓名： 辛可

序言

数据库是数据管理的最新技术，是计算机科学的重要分支。通过学习重点让大家掌握数据库技术的程序设计思想和方法，学习开发管理系统的技术，并结合高级语言和数据库 SQL 语言上机环境编程测试。

为了达到这个目的，这里安排四个实验单元。

实验报告的内容

1. 题目

描述每个实验的内容是什么。

2. 需求分析

用 E-R 图描述数据库的模式设计及每个关系模式的建立；描述数据字典及程序数据流；每个事件、函数或过程的头和规格说明；

3. 源程序清单和结果

源程序要加注释，要有测试数据及结果。

实验 1 学习 SQL 语句与 SQL SERVER 的数据库环境 (4 学时)

一、实验目的

本次实验的主要目的在于学会使用 DBMS（如 SQL Server，MySQL，华为 OpenGauss 等）上机环境；掌握创建数据库和表的操作；掌握数据录入和简单的查询操作；掌握索引的建立和删除操作。

注：华为 OpenGauss 数据库的安装与使用见《华为 openGauss 数据库使用说明》。

二、实验要求

1. 要求学生独立完成实验内容；
2. 按照实验步骤完成实验后，撰写报告内容，并对操作结果进行截图。

三、实验内容及实验结果

建立数据库 ST，三个表名称及实验数据如下：

S

sclass	sno	sname	ssex	sage	Sdept
1	1	李勇	男	20	IS
1	2	刘晨	女	19	IS
1	3	刘朋	男	20	IS
2	1	王敏	女	18	MA
2	2	张锋	男	19	MA
2	3	李敏	男	20	MA

SC

sclass	sno	cno	grade
1	1	1	92
1	1	2	85
1	1	3	88
1	2	2	90
1	2	3	80
2	1	1	75
2	1	2	92
2	2	2	87
2	2	3	89
2	3	1	90

C

cno	cname	cpno	ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL 语言	6	4

1. 建立课程表 C，定义好实体完整性和参照完整性

```
create table C(cno varchar(10) primary key,
cname varchar(30) not null,
cpno varchar(10),
ccredit int) comment='课程表';
alter table C add foreign key(cpno) references C(cno);
insert into C values("1","数据库","5",4);
insert into C values("2","数学",NULL,2);
insert into C values("3","信息系统","1",4);
insert into C values("4","操作系统","6",3);
insert into C values("5","数据结构","7",4);
insert into C values("6","数据处理",NULL,2);
insert into C values("7","PASCAL 语言","6",4);
```

	cno	cname	cpno	ccredit
▶	1	数据库	5	4
	2	数学	NULL	2
	3	信息系统	1	4
	4	操作系统	6	3
	5	数据结构	7	4
	6	数据处理	NULL	2
	7	PASCAL语言	6	4
*	NULL	NULL	NULL	NULL

图 1-1

2. 建立学生表 S，定义好实体完整性和参照完整性

```
create table S(sclass varchar(10),sno varchar(10),
sname varchar(10) not null,ssex varchar(1),sage int,sdept char(4),
primary key(sclass,sno)) comment='学生表';
insert into S values("1","1","李勇","男",20,"IS");
insert into S values("1","2","刘晨","女",19,"IS");
insert into S values("1","3","刘朋","男",20,"IS");
insert into S values("2","1","王敏","女",18,"MA");
insert into S values("2","2","张锋","男",19,"MA");
insert into S values("2","3","李敏","男",20,"MA");
```

	sclass	sno	sname	ssex	sage	sdept
▶	1	1	李勇	男	20	IS
	1	2	刘晨	女	19	IS
	1	3	刘朋	男	20	IS
	2	1	王敏	女	18	MA
	2	2	张锋	男	19	MA
	2	3	李敏	男	20	MA
*	NULL	NULL	NULL	NULL	NULL	NULL

图 1-2

3. 建立选课表 SC，定义好实体完整性和参照完整性

```
create table SC(sclass varchar(10),
sno varchar(10),
cno varchar(10),
grade int,
primary key(sclass,sno)) comment='选课表';
alter table SC add foreign key(cno) references C(cno);
alter table SC add foreign key(sclass,sno) references S(sclass,sno);
insert into SC values("1","1","1",92);
insert into SC values("1","1","2",85);
insert into SC values("1","1","3",88);
insert into SC values("1","2","2",90);
insert into SC values("1","2","3",80);
insert into SC values("2","1","1",75);
insert into SC values("2","1","2",92);
insert into SC values("2","2","2",87);
insert into SC values("2","2","3",89);
insert into SC values("2","3","1",90);
```

	sclass	sno	cno	grade
▶	1	1	1	92
	1	1	2	85
	1	1	3	88
	1	2	2	90
	1	2	3	80
	2	1	1	75
	2	1	2	92
	2	2	2	87
	2	2	3	89
	2	3	1	90
•	NULL	NULL	NULL	NULL

图 1-3

4. 将 S 表的 ssex 字段设为不能为空 (not null)

```
alter table S modify ssex char(1) not null;
```

```
1 • use st;
2 • ALTER TABLE S MODIFY ssex char(1) NOT NULL;
```

图 1-4

5. 查询所有学生的详细信息 (包含学生、选课及课程信息)

```
select S.sclass,S.sno,sname,ssex,sage,sdept,
SC.cno,grade,cname,cpno,ccredit
from S,SC,C
where S.sclass=SC.sclass and S.sno=SC.sno and SC.cno=C.cno;
```

```

1 • use st;
2 • select S.sclass,S.sno,sname,ssex,sage,sdept,SC.cno,grade,cname,cpno,ccredit
3   from S,SC,C
4  where S.sclass=SC.sclass AND S.sno=SC.sno AND SC.cno=C.cno;
5

```

	sclass	sno	sname	ssex	sage	sdept	cno	grade	cname	cpno	ccredit
▶	1	1	李勇	男	20	IS	1	92	数据库	5	4
	1	1	李勇	男	20	IS	2	85	数学	HULL	2
	1	1	李勇	男	20	IS	3	88	信息系统	1	4
	1	2	刘晨	女	19	IS	2	90	数学	HULL	2
	1	2	刘晨	女	19	IS	3	80	信息系统	1	4
	2	1	王敏	女	18	MA	1	75	数据库	5	4
	2	1	王敏	女	18	MA	2	92	数学	HULL	2
	2	2	张锋	男	19	MA	2	87	数学	HULL	2
	2	2	张锋	男	19	MA	3	89	信息系统	1	4
	2	3	李敏	男	20	MA	1	90	数据库	5	4

图 1-5

6. 查询 1 班的学生学号及姓名

```
select sno,sname from S where sclass='1';
```

```

1 • use st;
2 • select sno,sname
3   from S
4  where sclass='1'
5

```

	sno	sname
▶	1	李勇
	2	刘晨
	3	刘朋

图 1-6

7. 查询‘刘晨’的出生年

```
select sname,2022-sage BIRTHYEAR from S where sname='刘晨';
```

```

1 • use st;
2 • SELECT sname,2022-sage BIRTHYEAR
3   FROM S
4  WHERE sname='刘晨';
5

```

	sname	BIRTHYEAR
▶	刘晨	2003

图 1-7

8. 查询姓‘刘’的学生的详细情况(包括学生表、选课表及课程表的全部信息)

```
select S.sclass,S.sno,sname,ssex,sage,sdept,
```

```

SC.cno,grade,cname,cpno,ccredit
from S left outer join SC on(S.sclass=SC.sclass and S.sno=SC.sno)
    left outer join C on(SC.cno=C.cno)
where sname like '刘%';

```

```

1 • use st;
2 • SELECT S.sclass,S.sno,sname,ssex,sage,Sdept,SC.cno,grade,cname,cpno,ccredit
3 FROM S LEFT OUTER JOIN SC ON(S.sclass=SC.sclass AND S.sno=SC.sno) LEFT OUTER JOIN C ON(SC.cno=C.cno)
4 WHERE sname LIKE '刘%';

```

	sclass	sno	sname	ssex	sage	Sdept	cno	grade	cname	cpno	ccredit
▶	1	2	刘晨	女	19	IS	2	90	数学	NULL	2
	1	2	刘晨	女	19	IS	3	80	信息系统	1	4
	1	3	刘朋	男	20	IS	NULL	NULL	NULL	NULL	NULL

图 1-8

9. 查询选修了 1 号课的学生姓名、性别、成绩
- ```

select sname,ssex,grade
from S left outer join SC on(S.sclass=SC.sclass and S.sno=SC.sno)
where cno='1';

```

```

1 • use st;
2 • SELECT sname,ssex,grade
3 FROM S LEFT OUTER JOIN SC ON(S.sclass=SC.sclass AND S.sno=SC.sno)
4 WHERE cno='1';

```

|   | sname | ssex | grade |
|---|-------|------|-------|
| ▶ | 李勇    | 男    | 92    |
|   | 王敏    | 女    | 75    |
|   | 李敏    | 男    | 90    |

图 1-9

10. 查询没有先行课课程的课程号和课程名
- ```

select cno,cname from C where cpno is null;

```

```

1 • use st;
2 • SELECT cno,cname
3 FROM C
4 WHERE cpno IS NULL;

```

	cno	cname
▶	2	数学
	6	数据处理
*	NULL	NULL

图 1-10

11. 查询 2 班的所有女生的情况(包括学生表、选课表及课程表的全部信息)

```
select S.sclass,S.sno,sname,ssex,sage,sdept,
SC.cno,grade,cname,cpno,ccredit
from S left outer join SC on (S.sclass=SC.sclass and S.sno=SC.sno)
left outer join C on (SC.cno=C.cno)
where S.sclass='2' and ssex='女';
```

```
1 • use st;
2 • SELECT S.sclass,S.sno,sname,ssex,sage,Sdept,SC.cno,grade,cname,cpno,ccredit
3 FROM S LEFT OUTER JOIN SC ON(S.sclass=SC.sclass AND S.sno=SC.sno) LEFT OUTER JOIN C ON(SC.cno=C.cno)
4 WHERE S.sclass='2' AND ssex='女'
```

	sclass	sno	sname	ssex	sage	Sdept	cno	grade	cname	cpno	ccredit
▶	2	1	王敏	女	18	MA	1	75	数据库	5	4
	2	1	王敏	女	18	MA	2	92	数学	NULL	2

图 1-11

12. 查询学分为 2 到 3 之间的课程号及课程名

```
select cno,cname
from C
where ccredit>=2 and ccredit<=3;
```

```
1 • use st;
2 • SELECT cno,cname
3 FROM C
4 WHERE ccredit>=2 AND ccredit<=3;
```

	cno	cname
▶	2	数学
	4	操作系统
	6	数据处理
*	NULL	NULL

图 1-12

13. 查询选修 1 号课的学生的班号、学号、姓名、课程名及成绩, 要求成绩按照递减排序输出

```
select S.sclass,S.sno,sname,cname,grade
from S left outer join SC on(S.sclass=SC.sclass and S.sno=SC.sno)
left outer join C on(SC.cno=C.cno)
where SC.cno='1'
order by grade DESC;
```



```

1 • use st;
2 • SELECT S.sclass,S.sno,sname,cname,grade
3 FROM S LEFT OUTER JOIN SC ON(S.sclass=SC.sclass AND S.sno=SC.sno) LEFT OUTER JOIN C ON(SC.cno=C.cno)
4 WHERE SC.cno='1'
5 ORDER BY grade DESC;

```

	sclass	sno	sname	cname	grade
1	1	1	李勇	数据库	92
2	3	3	李敏	数据库	90
2	1	1	王敏	数据库	75

图 1-13

14. 查询 2 班至少选修一门其先行课为 1 号课的学生的学号、姓名、性别、系、课程号及成绩

```

select S.sclass,S.sno,sname,ssex,sdept,SC.cno,SC.grade
from S left outer join SC on(S.sclass=SC.sclass and S.sno=SC.sno)
left outer join C on(SC.cno=C.cno)
where SC.sclass='2' and (select count(*) from C where SC.cno=C.cno
and cpno='1')>=1;

```

```

1 • use st;
2 • SELECT S.sclass,S.sno,sname,ssex,Sdept,SC.cno,SC.grade
3 FROM S LEFT OUTER JOIN SC ON(S.sclass=SC.sclass AND S.sno=SC.sno)
4 WHERE SC.sclass='2' AND (SELECT count(*) FROM C WHERE SC.cno=C.cno AND cpno='1')>=1;

```

	sclass	sno	sname	ssex	Sdept	cno	grade
2	2	2	张锋	男	MA	3	89

图 1-14

15. 查询 2 号课成绩最高的学生班号、学号、姓名

```

select sclass,sno,sname from S
where (sclass,sno) in (select SC1.sclass,SC1.sno from
SC SC1 where SC1.cno='2' and SC1.grade=(select max(grade)
from SC SC2 group by SC2.cno having SC2.cno='2'));

```

```

1 • use st;
2 • SELECT sclass,sno,sname
3 FROM S
4 WHERE (sclass,sno) IN (SELECT SC1.sclass,SC1.sno FROM SC SC1 WHERE SC1.cno='2'
5 AND SC1.grade=(SELECT MAX(grade) FROM SC SC2 GROUP BY SC2.cno HAVING SC2.cno='2'));

```

	sclass	sno	sname
2	1	1	王敏

图 1-15

16. 查询 1 班 2 号课成绩最低的学生学号、姓名和性别

```
select sno,sname,ssex from S where (sclass,sno)
in (select SC1.sclass,SC1.sno
from SC SC1
where SC1.cno='2' and SC1.sclass='1'
and SC1.grade=(select min(grade)
from SC SC2
group by SC2.cno
having SC2.cno='2'));
```

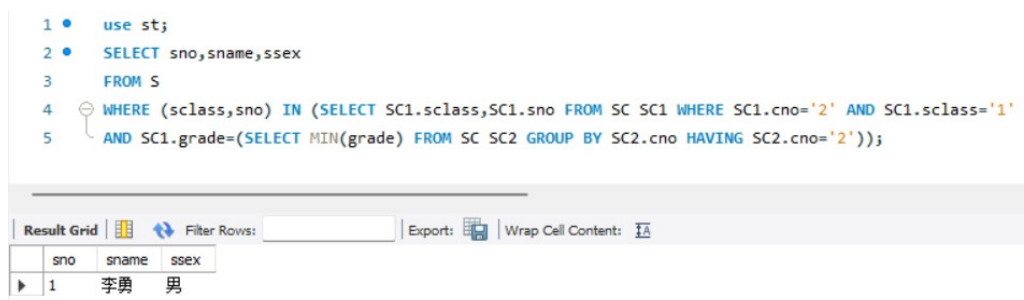


图 1-16

17. 查询选修 2 号课且成绩不是最低的同学班号、学号

```
select sclass,sno from S
where (sclass,sno) in
(select SC1.sclass,SC1.sno from
SC SC1 where SC1.cno='2'
and SC1.grade!=(select max(grade)
from SC SC2
group by SC2.cno
having SC2.cno='2'));
```

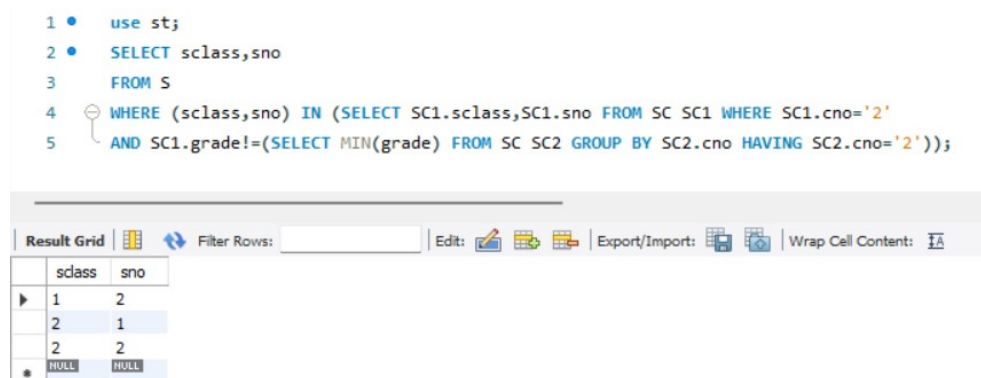


图 1-17

18. 查询每门课程的课程号及选修人数

```
select cno,count(sno) from SC group by cno;
```

```

1 • use st;
2 • SELECT cno,COUNT(sno)
3 FROM SC
4 GROUP BY cno;

```

	cno	COUNT(sno)
▶	1	3
	2	4
	3	3

图 1-18

19. 查询选修三门课的同学班号、学号、姓名、课程号、成绩

```

select S.sclass,S.sno,sname,SC1.cno,grade
from S left outer join SC SC1 on(S.sno=SC1.sno and
S.sclass=SC1.sclass)
where (select count(SC2.cno) from SC SC2 group by
SC2.sno,SC2.sclass having SC2.sno=SC1.sno and
SC2.sclass=SC1.sclass)=3;

```

```

1 • use st;
2 • SELECT S.sclass,S.sno,sname,SC1.cno,SC1.grade
3 FROM S LEFT OUTER JOIN SC SC1 ON (S.sno=SC1.sno AND S.sclass=SC1.sclass )
4 WHERE ( SELECT COUNT(SC2.cno)
5 FROM SC SC2
6 GROUP BY SC2.sno, SC2.sclass
7 HAVING SC2.sno=SC1.sno
8 AND SC2.sclass=SC1.sclass) = 3;

```

	sclass	sno	sname	cno	grade
▶	1	1	李勇	1	92
	1	1	李勇	2	85
	1	1	李勇	3	88

图 1-19

20. 在 S 表的 sage 字段添加约束，sage 的属性值在 18-30 岁之间
 alter table S add constraint sage check(18<=sage<=30);

21. 实现如下索引操作，如果不能成功建立索引，请分析原因

- (1) 在 S 表的 sname 列上建立普通降序索引。

```
create index sname_desc on S(sname DESC);
```

- (2) 在 C 表的 cname 列上建立唯一索引。

```
create unique index cname_unique on C(cname);
```

(3) 在 SC 表的 sno 列上建立聚集索引。

不能成功建立，因为 mysql 在创建主键时自动创建了主键的聚集索引

(4) 在 SC 表的 sclass(升序)，sno(升序)，cno (降序)三列上建立一个普通索引。

```
create index sclass_a_sno_a_cno_d on SC(sclass,sno,cno DESC);
```

(5) 删除索引： 将 C 表的 cname 列的唯一索引删掉。

```
drop index cname_unique on C;
```

四、实验总结

在实验中有哪些重要问题或者事件？你如何处理的？你的收获是什么？

问题：向 S 表中插入数据报错

解决：由于主码定义错误，导致主码重复，添加数据失败。使用了(sclass,sno)的组合主码后即可成功添加数据

收获：学会了 mysql workbench 链接 mysql 数据库，基础的 mysql 语句编写，学会了灵活处理不同的查询需求。

实验 2 学习 SQL 语句的数据操纵与视图（4 学时）

一、实验目的

本次实验的主要目的在于学会使用 SQL 语言的相关子查询及 SQL 语言的增、删、改、查功能及视图操作。

二、实验要求

1. 要求学生独立完成实验内容；
2. 按照实验步骤完成实验后，撰写报告内容，并对操作结果进行截图。

三、实验内容及实验结果

对实验 1 操作的数据库 ST 执行如下操作。

(一) SQL 相关子查询及增、删、改功能

1. 查询选了 1 号课且选了 2 号课的学生的班号、学号

```
select SC1.sclass,SC1.sno from SC SC1
where SC1.cno='1' and (SC1.sclass,SC1.sno) in (
select SC2.sclass,SC2.sno
from SC SC2 where SC2.cno='2');
```

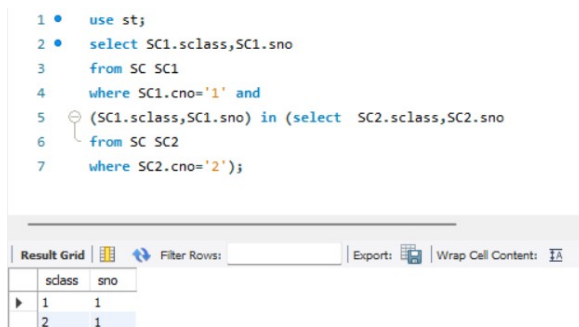


图 2-1

2. 查询选了 1 号课但不选 2 号课的学生的班号、学号

```
select SC1.sclass,SC1.sno from SC SC1
where SC1.cno='1' and (SC1.sclass,SC1.sno) not in (
select SC2.sclass,SC2.sno
from SC SC2 where SC2.cno='2');
```

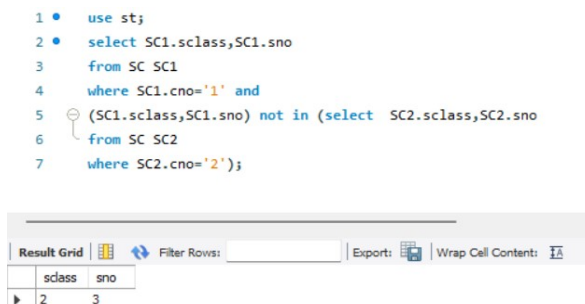


图 2-2

3. 查询 1 班平均分在 85 分以上的同学学号、姓名、性别、系、各科课程号及成绩

```
select S.sclass,S.sno,sname,ssex,sdept,SC.cno,grade
from S left outer join SC on (S.sno=SC.sclass and S.sclass=SC.sclass)
where S.sno in (select sno from SC SC1
where SC1.sclass='1'
and (select avg(SC2.grade) from SC SC2
where SC1.sno=SC2.sno
and SC1.sclass=SC2.sclass group by SC.sno)>85)
and S.sclass='1';
```

```

1 • use st;
2 • select S.sclass,S.sno,sname,ssex,sdept,SC.cno,grade
3   from S left outer join SC on (S.sno=SC.sno and S.sclass=SC.sclass)
4  where S.sno in (select sno from SC SC1 where SC1.sclass='1'
5   and (select avg(SC2.grade) from SC SC2 where
6    SC1.sno=SC2.sno and SC1.sclass=SC2.sclass group by SC.sno)>85)
7   and S.sclass='1';

```

sclass	sno	sname	ssex	sdept	cno	grade
1	1	李勇	男	IS	1	92
1	1	李勇	男	IS	2	85
1	1	李勇	男	IS	3	88

图 2-3

4. 查询至少选了 1 班 2 号同学所选课的所有班号、学号及同学姓名
- ```

select sclass,sno,sname from S
where not exists(select * from SC SC1 where SC1.sclass='1'
and SC1.sno='2' and not exists(select * from SC SC2
where SC2.sno=S.sno and SC2.sclass=S.sclass and SC2.cno=SC1.cno));

```

```

1 • use st;
2 • select sclass,sno,sname
3 from S
4 where not exists (select * from SC SC1 where SC1.sclass='1' and SC1.sno='2' and
5 not exists (select * from SC SC2 where SC2.sno=S.sno and SC2.sclass=S.sclass and SC2.cno=SC1.cno));

```

| sclass | sno  | sname |
|--------|------|-------|
| 1      | 1    | 李勇    |
| 2      | 2    | 张锋    |
| 1      | 2    | 刘晨    |
| NULL   | NULL | NULL  |

图 2-4

5. 查询不选 1 号课的学生班号及学号
- ```

select sclass,sno from S
where not exists(select * from SC where SC.sclass=S.sclass
and SC.sno=S.sno and SC.cno='1');

```

```

1 • use st;
2 • select sclass,sno
3   from S
4  where not exists (select * from SC where SC.sclass=S.sclass and SC.sno=S.sno and SC.cno='1');

```

sclass	sno
2	2
1	3
1	2
NULL	NULL

图 2-5

6. 查询选 2 号课的学生名字及相应 2 号课成绩，按成绩从高到低排序

```
select sname,grade
from S left outer join SC on (S.sclass=SC.sclass and S.sno=SC.sno)
where cno='2' order by grade DESC;
```

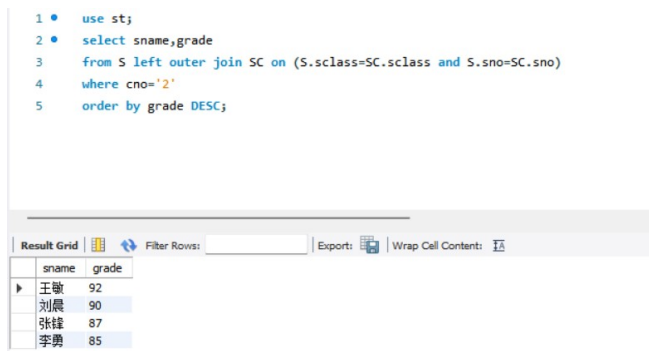


图 2-6

7. 统计选修课程的学生班号、学号及总学分

```
select sclass,sno,sum(ccredit) from SC,C
where SC.cno=C.cno group by sclass,sno;
```

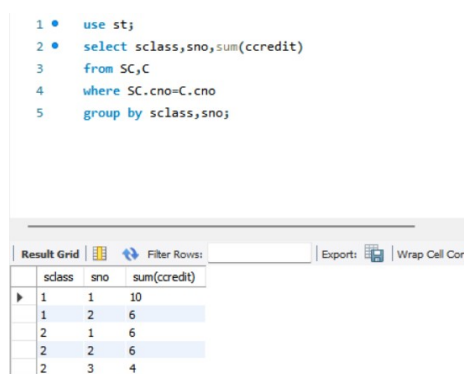


图 2-7

8. 统计 1 班选修 3 号课的学号及平均分

```
select SC1.sno,avg(SC1.grade) from SC SC1
where (SC1.sclass,SC1.sno) in ( select SC2.sclass,SC2.sno
from SC SC2 where SC2.cno='3' and SC2.sclass='1') group by sno;
```

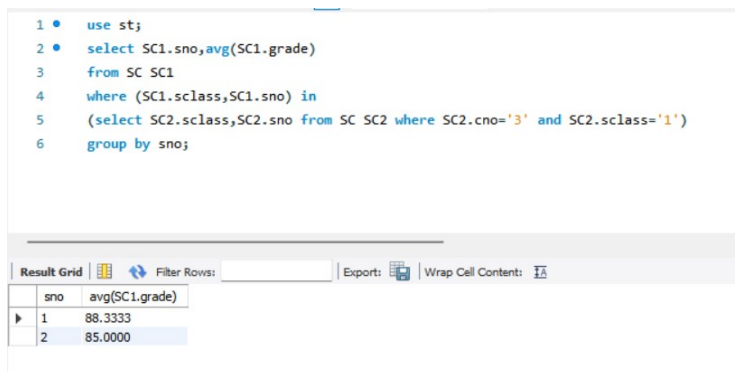


图 2-8

9. 在 SC 表中插入选课信息：1 班，4 号学生，选修 3 号课程。该信息是否能成功插入 SC 表？若不能请说明原因。

不能 不满足外码约束

10. 把个人信息(班级号:2、学号: 4、姓名: 张三、性别: 男、年龄: 17、所在系: IS)及选课信息(班级号:2、学号: 4、课程号: 4、成绩: 95)插入到 S 和 SC 表及新增加一门“无机化学”课程信息。该插入操作执行时有问题吗？如果有问题请说明原因，并改正使其能够插入成功。

有问题 第一条年龄不满足之前定义过的年龄字段约束

删除年龄约束：

```
alter table S drop constraint sage;
```

插入信息：

```
insert into S values("2","4","张三","男",17,"IS");
```

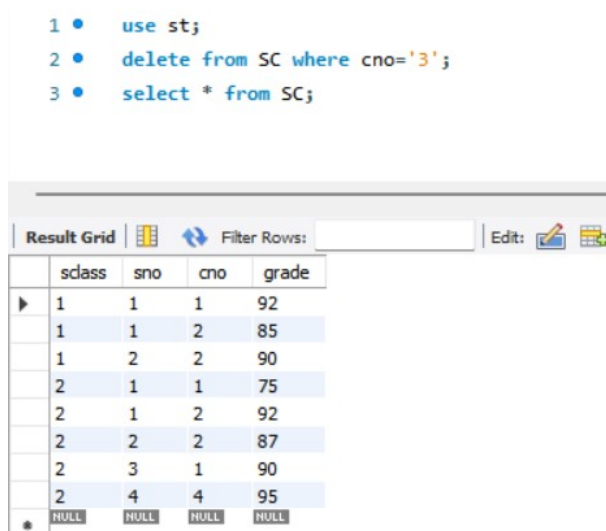
```
insert into SC values("2","4","4",95);
```

```
insert into C values("8","无机化学",NULL,NULL);
```

11. 删除选修 3 号课的所有选课信息并显示删除后的结果

```
delete from SC where cno='3';
```

```
select * from SC;
```



```
1 • use st;
2 • delete from SC where cno='3';
3 • select * from SC;
```

	sclass	sno	cno	grade
▶	1	1	1	92
	1	1	2	85
	1	2	2	90
	2	1	1	75
	2	1	2	92
	2	2	2	87
	2	3	1	90
	2	4	4	95
•	NULL	NULL	NULL	NULL

图 2-9

12. 把选修 1 号课的所有男同学年龄增加 1 岁并显示最终 S 表信息

```
update S set sage=sage+1 where (sclass,sno) in
```

```
(select SC.sclass,SC.sno from SC where cno='1');
```


select * from S;

```

1 • use st;
2 • update s
3   set sage=sage+1 where (sclass,sno) in
4     (select SC.sclass,SC.sno from SC where cno='1');
5 • select * from S

```

Result Grid						
Filter Rows:						
	sclass	sno	sname	ssex	sage	sdept
▶	1	1	李勇	男	21	IS
	1	2	刘晨	女	19	IS
	1	3	刘朋	男	20	IS
	2	1	王敏	女	19	MA
	2	2	张锋	男	19	MA
	2	3	李敏	男	21	MA
	2	4	张三	男	17	IS
✱	NULL	NULL	NULL	NULL	NULL	NULL

图 2-10

13. 把每个选课人的学号、班号及平均成绩插入到一个新表中。

create table Avg_Grade

(sclass varchar(10),sno varchar(10),avg_grade float,

primary key(sclass,sno));

insert into Avg_Grade

select sclass,sno,avg(grade) from SC group by sclass,sno;

select * from Avg_Grade;

```

1 • use st;
2 • create table Avg_Grade
3   (sclass varchar(10),sno varchar(10),avg_grade float,primary key (sclass,sno));
4 • insert into Avg_Grade
5   select sclass,sno,AVG(grade) from SC
6   group by sclass,sno;
7 • select * from Avg_Grade;

```

Result Grid			
Filter Rows:			
	sclass	sno	avg_grade
▶	1	1	88.5
	1	2	90
	2	1	83.5
	2	2	87
	2	3	90
	2	4	95
✱	NULL	NULL	NULL

图 2-11

(二) 视图 SQL 语言功能

1 创建视图:

(1) 在 ST 库中以 S 表为基础,建立信息系(IS 系) 学生的视图 V_IS_Student

```
create view V_IS_Student as
select * from S where sdept='IS';
```

(2) 建立一个每个学生的学号、班号、姓名、选修的课名及成绩的视图 S_C_GRADE;

```
create view S_C_GRADE as
select S.sno,S.sclass,sname,cname,grade
from S left outer join SC on(S.sno=SC.sno and S.sclass=SC.sclass)
left outer join C on(SC.cno=C.cno);
```

(3) 建立信息系(IS 系) 选修了 1 号课程且成绩在 90 分以上的学生的视图 V_IS_Score

```
create view V_IS_Score as
select * from S where sdept='IS' and (S.sclass,S.sno) in
(select SC.sclass,SC.sno from SC where cno='1' and grade>90);
```

(4) 将各系学生人数, 平均年龄定义为视图 V_NUM_AVG。

```
create view V_NUM_AVG as
select sdept,count(*),avg(sage) from S group by sdept;
```

2 查询以上所建的视图结果

```
select * from V_IS_Student;
select * from S_C_GRADE;
select * from V_IS_Score;
select * from V_NUM_AVG;

1 • use st;
2 • select * from V_IS_Student;
3 • select * from S_C_GRADE;
4 • select * from V_IS_Score;
5 • select * from V_NUM_AVG;
```

图 2-12

	sclass	sno	sname	ssex	sage	sdept
▶	1	1	李勇	男	21	IS
	1	2	刘晨	女	19	IS
	1	3	刘朋	男	20	IS
	2	4	张三	男	17	IS

图 2-13

	sno	sclass	sname	cname	grade
▶	1	2	王敏	数据库	75
	1	2	王敏	数学	92
	3	2	李敏	数据库	90
	1	1	李勇	数据库	92
	1	1	李勇	数学	85
	2	2	张峰	数学	87
	4	2	张三	操作系统	95
	3	1	刘朋	NULL	NULL
	2	1	刘晨	数学	90

V_IS_Student 2 S_C_GRADE 3 × V_IS_Score 4 V_NUM_AVG 5

图 2-14

Result Grid						
	sclass	sno	sname	ssex	sage	sdept
▶	1	1	李勇	男	21	IS

图 2-15

Result Grid			
	sdept	count(*)	avg(sage)
▶	IS	4	19.2500
	MA	3	19.6667

图 2-16

- 3 在上面建立的视图上查询选修了 1 号课程的信息系学生
- ```
select * from V_IS_Student
where (sclass,sno) in (select SC.sclass,SC.sno from SC where cno='1');
```

```

1 • use st;
2 • select * from V_IS_Student
3 where (sclass,sno) in
4 (select SC.sclass,SC.sno from SC where cno='1');

```

| Result Grid |        |     |       |      |      |       |
|-------------|--------|-----|-------|------|------|-------|
|             | sclass | sno | sname | ssex | sage | sdept |
| ▶           | 1      | 1   | 李勇    | 男    | 21   | IS    |

图 2-17

- 4 在信息系学生的视图中找出年龄小于 20 岁的学生
- ```
select * from V_IS_Student where sage<20;
```

```

1 • use st;
2 • select * from V_IS_Student
3   where sage<20;

```

Result Grid						
	sclass	sno	sname	ssex	sage	sdept
▶	1	2	刘晨	女	19	IS
	2	4	张三	男	17	IS

图 2-18

- 5 将信息系学生视图 V_IS_Student 中 1 班 2 号的学生姓名改为“刘辰”

```
update V_IS_Student
set sname='刘辰'
where sclass='1' and sno='2';
```

- 6 删除视图 S_C_GRADE

```
drop view S_C_GRADE;
```

四、实验总结

在实验中有哪重要问题或者事件？你如何处理的？你的收获是什么？

收获：学会了 mysql 数据库基本的增删改查和视图的操作，自然语言向关系代数语言和 sql 语言的转化

实验 3 高级语言与数据库的连接及 SQL 语言查询编程（4 学时）

一、实验目的

1. 本次实验的主要目的是掌握高级语言与数据库的连接方法与编程技巧。
2. SQL 语言与高级语言的联合编程。

二、实验要求

1. 要求学生独立完成实验内容，画出 E-R 图及程序功能图；
2. 按照实验步骤完成实验后，撰写报告内容，并对操作结果进行截图，写出主要关键程序代码。

三、实验内容及实验结果与主要代码

1. 学习高级语言与数据库的连接方法，写出数据库的连接语句。

链接数据库：

```
connection = pymysql.connect(host='localhost', port=3306, user='root',
                             password='root', database='korako', charset='utf8',
                             cursorclass=pymysql.cursors.DictCursor)
```

2. 采用 SQL 语言编程如下界面功能的查询。学会使用 SQL 编程对数据库进行单表精确查询、模糊查询的方法，给出主要程序代码（要有注释）和运行结果截图。

结果显示列表	输入家庭地址:
<div>95001 王明 24 威海</div> <div>⋮</div>	<input type="text"/>
	<input type="button" value="查询"/>

E-R 图:

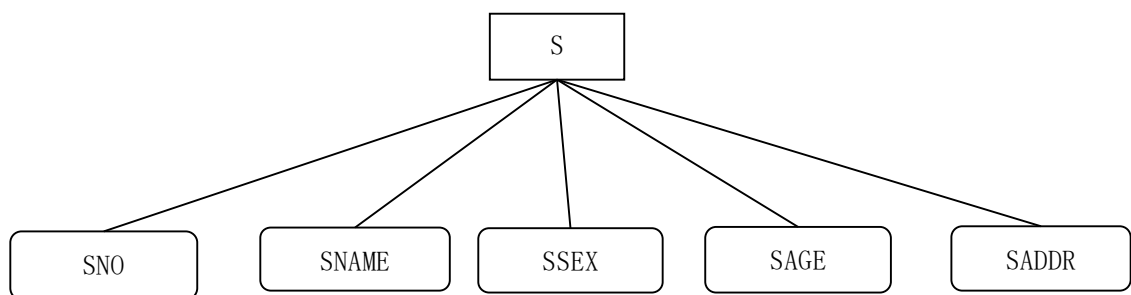


图 3-1

程序功能图:

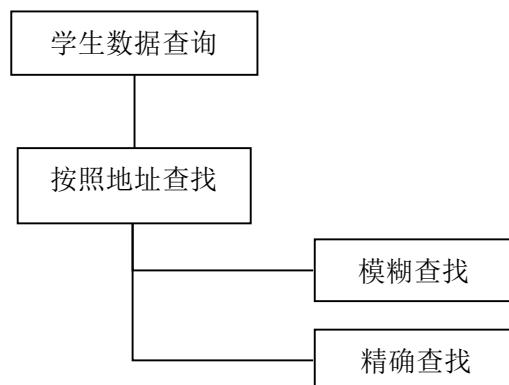


图 3-2

建立学生表: S(SNO,SNAME,SSEX,SAGE,SADDR)

```
import pymysql
```

```
try:
```

```
    cursor = connection.cursor()
```

```
    sql = "create table S (SNO varchar(10) primary key, SNAME varchar(20), SSEX  
varchar(1), SAGE int, SADDR varchar(20))"
```

```
    cursor.execute(sql)
```

运行截图:



图 3-3

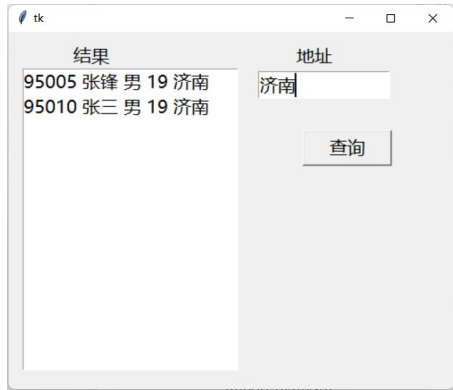


图 3-4

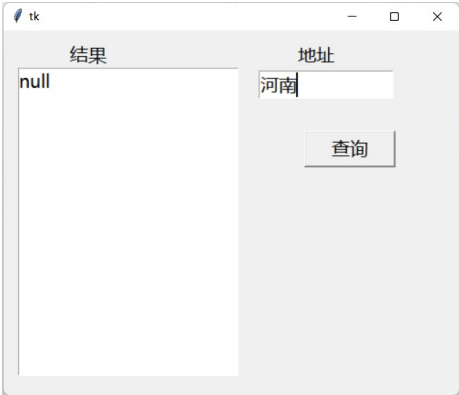


图 3-5



图 3-6

```
import pymysql
import tkinter as tk
from fileinput import close
class kora:
    db = pymysql.connect(host="localhost", user="root", password="root",
database="korako")
    cursor = db.cursor()

    def __init__(self):
        self.root = tk.Tk()      #创建窗口
        self.root.geometry("500x400")
    def select(self):
        address = self.Information.get()
        search_str = "select * from S where SADDR like '" + "%" + address + "%'"
        kora.cursor.execute(search_str)      #执行 sql 语句
        info = kora.cursor.fetchall()      #获取所有结果
        self.listbox.delete(0, tk.END)      #清空列表
        if len(info) == 0:                  #若结果为空返回 null
            self.listbox.insert(0, "null")
        for i in info:
            self.listbox.insert(tk.END, i)   #插入数据

    def interface(self):
        self.Information = tk.StringVar()
        self.entry = tk.Entry(self.root, width=12, bd=1, font=("微软雅黑",15)
            textvariable=self.Information) # 定义输入框
        self.entry.place(x=280, y=44)
        self.label1 = tk.Label(self.root, text="输入地址", font=("微软雅黑",15))
        self.label1.place(x=319, y=11)
        self.Button1 = tk.Button(self.root, text="查询", font=("微软雅黑",15)
            command=self.select) # 定义按钮
        self.Button1.place(x=330, y=106, relheight=0.1, relwidth=0.2)
        self.label2 = tk.Label(self.root, text="结果列表", font=("微软雅黑",15))
        self.label2.place(x=70, y=10)
```

```
self.listbox = tk.Listbox(self.root, width=20, height=12) # 定义输出列表
self.listbox.place(x=15, y=40)

def __del__(self):
    kora.cursor.close()
    kora.db.close()

if __name__ == "__main__":
    kora = kora()
    kora.interface()
    kora.root.mainloop()
```

四、实验总结

在实验中有重要问题或者事件？你如何处理的？你的收获是什么？

收获：学会了使用 `pymysql` 链接 `mysql` 数据库，和嵌入式的 `mysql` 语句封装和执行。

实验 4 SQ 综合应用编程（4 学时）

一、实验目的

本次实验的主要目的是掌握数据库编程，学会使用 SQL 编程对数据库进行增、删、改、备份的方法。

二、实验要求

1. 要求学生独立完成实验内容，画出 E-R 图及程序功能图；
2. 按照实验步骤完成实验后，撰写报告内容，并对操作结果进行截图，写出主要关键程序代码。

三、实验内容、实验结果与主要程序代码

基于实验一的三个表采用数据库编程实现数据库的录入、修改、删除和备份等管理功能，并能实现基于学号查询显示学生基本信息、课程名、成绩信息。

（一） 画出 E-R 图及程序功能分析设计图

E-R 图:

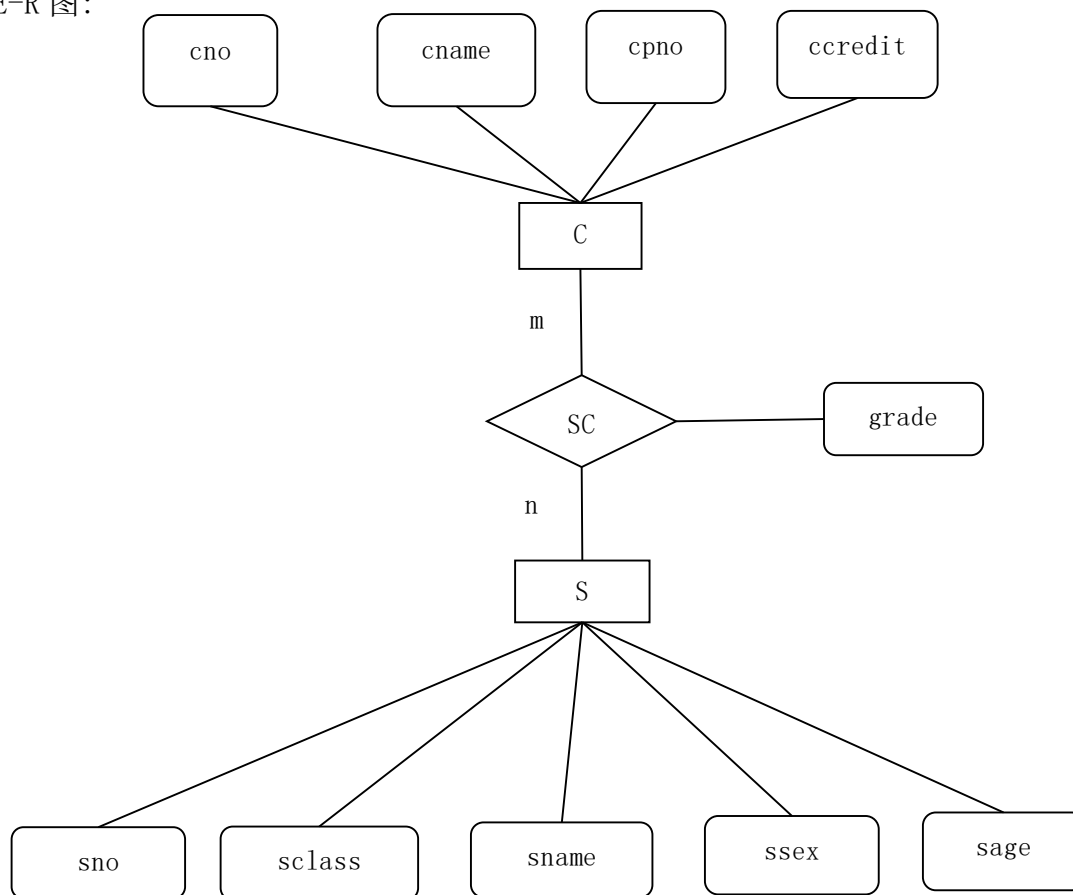


图 4-1

程序功能图:

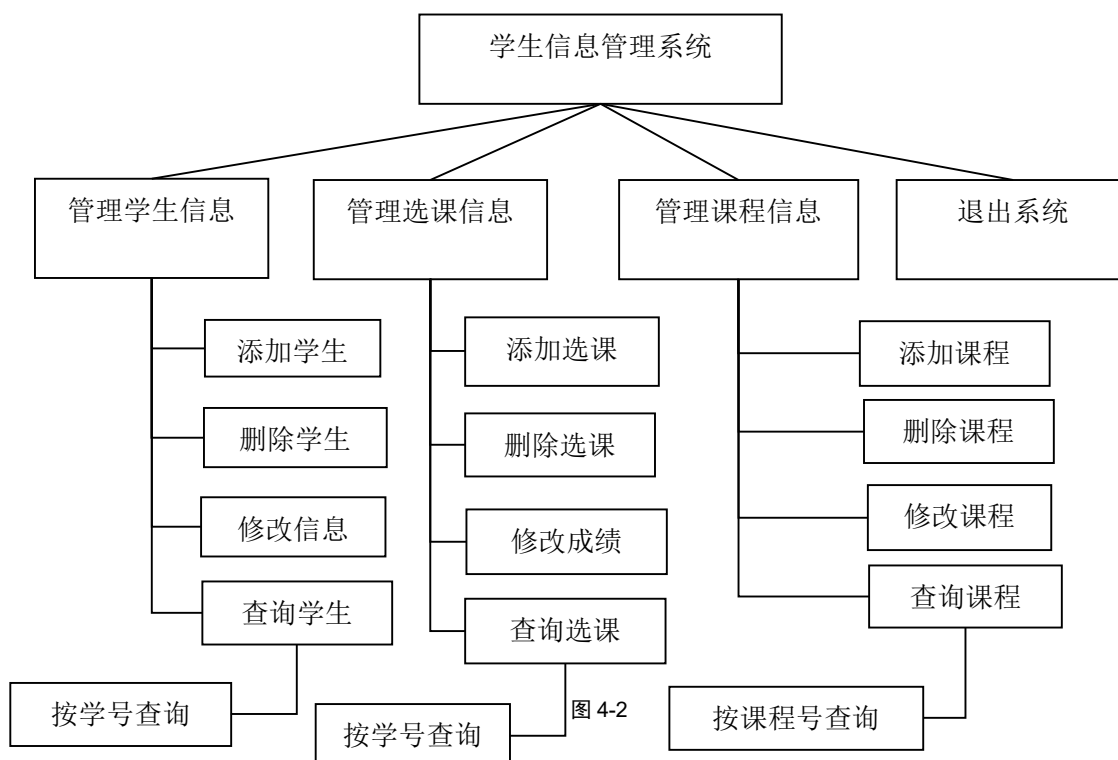


图 4-2

(二) 功能实现界面图及主要程序代码 (要有注释)

运行截图:

```

test4_packed x
C:\Users\86178\Anaconda2\envs\TF2.1\python.exe D:/Pycharm/database/test4_packed.py
请选择以下菜单号:
=====
1 登录学生信息管理系统
2 退出学生信息管理系统
=====
输入菜单号: 1
学生信息管理系统
=====
1 操作学生表
2 操作学生选课表
3 操作课程表
4 返回上级菜单
=====
输入菜单号: 4
请选择以下菜单号:
=====
1 登录学生信息管理系统
2 退出学生信息管理系统
=====
输入菜单号: 2
感谢使用学生信息管理系统!

Process finished with exit code 0

```

图 4-3

```

=====
1 按学号查询学生记录
2 查询全部学生记录
3 返回上级菜单
=====
请输入查询的菜单号: 2
      1  1  李勇 男  21  IS  数据库  92
      1  1  李勇 男  21  IS  数学  85
      1  2  刘辰 女  19  IS  数学  90
      2  1  王敏 女  20  MA  数据库  75
      2  1  王敏 女  20  MA  数学  92
      2  2  张锋 男  19  MA  数学  87
      2  3  李敏 男  21  MA  数据库  90
      2  4  张三 男  17  IS  操作系统  95
查询学生记录
=====
1 按学号查询学生记录
2 查询全部学生记录
3 返回上级菜单
=====
请输入查询的菜单号: 1
请输入班号: 2
请输入学号: 3
2 3 李敏 男 21 MA 数据库 90
查询成功
是否继续查询(y/n): n

```

图 4-4

```

=====
1 增加学生记录
2 查询学生记录
3 修改学生记录
4 删除学生记录
5 显示所有的学生记录
6 返回上级菜单
=====
输入菜单号: 3
请输入想要修改学生的班号: 1
请输入想要修改学生的学号: 1
=====
1 修改姓名
2 修改性别
3 修改年龄
4 修改班级
5 修改系
6 返回上级菜单
=====
请输入菜单号: 3
请输入修改后的年龄: 23
成功!

```

图 4-5

```

=====
1 按学号查询学生记录
2 查询全部学生记录
3 返回上级菜单
=====
请输入查询的菜单号: 1
请输入班号: 1
请输入学号: 1
1 1 李勇 男 23 IS 数据库 92
查询成功

```

图 4-6

```

import pymysql
conn = pymysql.connect(host="localhost", user="root", password="root",
database="st")
def main():
    cursor = conn.cursor(pymysql.cursors.SSCursor)
    return cursor
def print_menu(a, b):
    print(a)
    print('=====')
    i = 1
    for items in b:
        if items == 0:
            break
        else:
            print(i, items)

```

```

        i = i + 1

    print('=====')
def check_1(temp1):
    if temp1 == 1:
        conn.commit()
        print('成功! ')
    else:
        print('失败! ')
def jump_menu(i, func):
    a = func[i-1]
    a()
def login():
    while True:
        print_menu('学生信息管理系统', list(['操作学生表', '操作学生选课表', '操作课程表', '返回上级菜单']))
        opt1 = int(input('输入菜单号: '))
        if opt1 <= 3:
            jump_menu(opt1, func=[menu_s, menu_sc, menu_c])
        else:
            mainmenu()
def menu_s():
    while True:
        print_menu('学生信息管理系统', list(['增加学生记录', '查询学生记录', '修改学生记录', '删除学生记录',
                                                '显示所有的学生记录', '返回上级菜单']))
        mc2 = int(input('输入菜单号: '))
        func = [s_add, s_query, s_update, s_delete, s_print]
        if mc2 <= 5:
            jump_menu(mc2, func)
        else:
            break
def s_add(): # 插入学生记录
    cursor = main()
    sclass = input('班号: ')

```

```

sno = input('学号: ')
sname = input('姓名: ')
ssex = input('性别: ')
sage = int(input('年龄: '))
sdept = input('系: ')
add = cursor.execute('insert into S values(%s,%s,%s,%s,%s,%s)',
                      (sclass, sno, sname, ssex, sage, sdept))

check_1(add)

def s_query():
    while True:
        print_menu('查询学生记录', list(['按学号查询学生记录', '查询全部学生
记录', '返回上级菜单']))
        mc3 = int(input('请输入查询的菜单号: '))
        if mc3 <= 2:
            jump_menu(mc3, func=[s_q_by_sno, s_q_all])
        else:
            menu_s()

def s_q_by_sno():
    cursor = main()
    choice_cls = input('请输入班号: ')
    choice_id = input('请输入学号: ')
    cursor.execute('select S.sclass,S.sno,sname,ssex,sage,sdept,cname,grade '
                  'from S,SC,C where S.sno=%s and S.sclass=%s'
                  'and S.sclass=SC.sclass and S.sno=SC.sno and SC.cno=C.cno',
                  (choice_id, choice_cls))
    students = cursor.fetchall()
    for stu in students:
        print(stu[0], stu[1], stu[2], stu[3], stu[4], stu[5], stu[6], stu[7])
        print('查询成功')
        re = input('是否继续查询(y/n): ')
        if re == 'y':
            s_q_by_sno()
        else:
            menu_s()

def s_q_all():

```

```

cursor = main()
cursor.execute('select S.sclass,S.sno,sname,ssex,sage,sdept,cname,grade '
               'from S,SC,C where S.sclass=SC.sclass and S.sno=SC.sno and
SC.cno=C.cno')
students = cursor.fetchall()
for student in students:
    print('\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}'
          .format(student[0], student[1],
student[2], student[3], student[4],
student[5],
student[6], student[7]))
def s_update():
    cursor = main()
    curr = input('请输入想要修改学生的班号: ')
    cur = input('请输入想要修改学生的学号: ')
    print_menu(' ', list(['修改姓名', '修改性别', '修改年龄', '修改班级', '修改系', '返回上级菜单']))
    mc2 = int(input('请输入菜单号: '))
    if mc2 == 1:
        name = input('请输入修改后的名字: ')
        a = cursor.execute('update S set sname=%s where sno=%s and sclass=%s',
(name, cur, curr))
        check_1(a)
    elif mc2 == 2:
        gender1 = input('请输入修改后的性别: ')
        a = cursor.execute('update S set ssex=%s where sno=%s and sclass=%s',
(gender1, cur, curr))
        check_1(a)
    elif mc2 == 3:
        age1 = input('请输入修改后的年龄: ')
        a = cursor.execute('update S set sage=%s where sno=%s and sclass=%s',
(age1, cur, curr))
        check_1(a)
    elif mc2 == 4:
        class1 = input('请输入修改后的班级: ')
        a = cursor.execute('update S set sclass=%s where sno=%s and sclass=%s',

```

```

(class1, cur, curr))
    b = cursor.execute('update SC set sclass=%s where sno=%s and sclass=%s',
(class1, cur, curr))
    check_1(a)
    check_1(b)
elif mc2 == 5:
    major1 = input('请输入修改后的专业: ')
    a = cursor.execute('update S set sdept=%s where sno=%s and sclass=%s',
(major1, cur, curr))
    check_1(a)
else:
    menu_s()
def s_delete():
    print_menu(' ', list(['删除学生所有信息', '回到初始界面']))
    mc4 = int(input('Input menu number:'))
    if mc4 == 1:
        s_delete_by_sno()
    else:
        menu_s()
def s_delete_by_sno():
    cursor = main()
    cls = input('输入想要删除学生的班号: ')
    sid = input('输入想要删除学生的学号: ')
    delete_s = cursor.execute('delete from S where sno=%s and sclass=%s', (sid, cls))
    delete_sc = cursor.excute('delete from SC where sno=%s and sclass=%s', (sid,
cls))
    if delete_s == 1 & delete_sc == 1:
        conn.commit()
        print('删除成功! ')
    else:
        print('删除失败! ')
def s_print():
    cursor = main()
    cursor.execute('select * from S')
    students = cursor.fetchall()

```

```
for stu in students:
    print(stu[0], stu[1], stu[2], stu[3], stu[4], stu[5])
def menu_sc():
    while True:
        print_menu('学生信息管理系统', list(['增加学生选课', '查询学生选课', '修改学生选课', '删除学生选课', '返回上级菜单']))
        opt_sc = int(input('输入菜单号: '))
        if opt_sc <= 4:
            jump_menu(opt_sc, func=[sc_add, sc_query, sc_update, sc_delete])
        else:
            login()
def sc_add():    # 插入选课记录
    cursor = main()
    sclass = input('班号: ')
    sno = input('学号: ')
    cno = input('课程: ')
    grade = int(input('成绩: '))
    add = cursor.execute('insert into SC values (%s,%s,%s,%s)', (sclass, sno, cno, grade))
    check_1(add)
def sc_query(): # 查询选课信息菜单
    while True:
        print_menu('查询选课记录', list(['按学号查询学生选课记录', '显示全部选课信息', '返回上级菜单']))
        opt_q_sc = int(input('请输入查询的菜单号: '))
        if opt_q_sc <= 2:
            jump_menu(opt_q_sc, func=[sc_q_by_sno, sc_q_all])
        else:
            menu_sc()
def sc_q_by_sno():
    cursor = main()
    choice_cls = input('请输入班号: ')
    choice_id = input('请输入学号: ')
    cursor.execute('select * from SC where sno =%s and sclass=%s', (choice_id, choice_cls))
```



```

students = cursor.fetchall()
for stu in students:
    print(stu[0], stu[1], stu[2], stu[3])
    print('查询成功')
    re = input('是否继续查询(y/n): ')
    if re == 'y':
        sc_q_by_sno()
    else:
        return
def sc_q_all():
    cursor = main()
    cursor.execute('select * from SC')
    students = cursor.fetchall()
    for student in students:
        print('\t{}\t{}\t{}\t{}'.format(student[0], student[1], student[2], student[3]))
def sc_update():
    cursor = main()
    curr = input('请输入想要修改学生的班号: ')
    cur = input('请输入想要修改学生的学号: ')
    print_menu(' ', list(['修改课程号', '修改成绩', '返回上级菜单']))
    mc2 = int(input('请输入菜单号: '))
    if mc2 == 1:
        name = input('请输入修改后的课程号: ')
        a = cursor.execute('update SC set cno=%s where sno=%s and sclass=%s',
(name, cur, curr))
        check_1(a)
    elif mc2 == 2:
        gender1 = input('请输入修改后的成绩: ')
        a = cursor.execute('update SC set grade=%s where sno=%s and sclass=%s',
(gender1, cur, curr))
        check_1(a)
    else:
        menu_sc()
def sc_delete():
    print_menu(' ', list(['删除选课所有信息', '回到初始界面']))

```

```
mc4 = int(input('Input menu number:'))
if mc4 == 1:
    sc_delete_by_sno()
else:
    menu_sc()
def sc_delete_by_sno():
    cursor = main()
    cls = input('输入想要删除学生的班号: ')
    sid = input('输入想要删除学生的学号: ')
    cid = input('输入想要删除学生的课程号: ')
    delete = cursor.execute('delete from SC where sno =%s and sclass=%s and
cno=%s', (sid, cls, cid))
    check_1(delete)
def menu_c():
    while True:
        print_menu('学生信息管理系统', list(['增加课程', '查询课程', '修改课程', '
删除课程', '返回上级菜单']))
        opt_c = int(input('输入菜单号: '))
        if opt_c <= 4:
            jump_menu(opt_c, func=[c_add, c_query, c_update, c_delete])
        else:
            login()
def c_add():
    cursor = main()
    cno = input('课程号: ')
    cname = input('课程名: ')
    cpno = input('先修课程号: ')
    ccredit = int(input('学分: '))
    add = cursor.execute('insert into C values (%s,%s,%s,%s)', (cno, cname, cpno,
ccredit))
    check_1(add)
def c_query():
    while True:
        print_menu('查询课程记录', list(['按课程号查询课程信息', '显示全部课
程信息', '返回上级菜单']))
```

```
    opt_q_sc = int(input('请输入查询的菜单号: '))
    if opt_q_sc <= 2:
        jump_menu(opt_q_sc, func=[c_q_by_cno, c_q_all])
    else:
        break
def c_q_by_cno():
    cursor = main()
    cid = input('请输入课程号: ')
    cursor.execute('select * from C where cno =%s', cid)
    students = cursor.fetchall()
    for stu in students:
        print(stu[0], stu[1], stu[2], stu[3])
        print('查询成功')
        re = input('是否继续查询(y/n): ')
        if re == 'y':
            c_q_by_cno()
        else:
            return
def c_q_all():
    cursor = main()
    cursor.execute('select * from C')
    students = cursor.fetchall()
    for student in students:
        print('\t{}\t{}\t{}\t{}'.format(student[0], student[1], student[2], student[3]))
def c_update():
    cursor = main()
    curr = input('请输入想要修改学生的课程号: ')
    print_menu(' ', list(['修改课程名', '修改先修课程号', '修改学分', '返回上级菜单']))
    mc2 = int(input('请输入菜单号: '))
    if mc2 == 1:
        ccc = input('请输入修改后的课程名: ')
        a = cursor.execute('update C set cname=%s where cno =%s', (ccc, curr))
        check_1(a)
    elif mc2 == 2:
```

```

        name = input('请输入修改后的先修课程号: ')
        a = cursor.execute('update C set cpno=%s where cno =%s', (name, curr))
        check_1(a)
    elif mc2 == 3:
        gender1 = input('请输入修改后的学分: ')
        a = cursor.execute('update C set ccredit=%s where cno =%s', (gender1, curr))
        check_1(a)
    else:
        menu_c()
def c_delete():
    print_menu(' ', list(['删除课程所有信息', '回到初始界面']))
    mc4 = int(input('Input menu number:'))
    if mc4 == 1:
        c_delete_by_cno()
    else:
        menu_c()
def c_delete_by_cno():
    cursor = main()
    cid = input('输入想要删除学生的课程号: ')
    delete = cursor.execute('delete from C where cno =%s', cid)
    check_1(delete)
def mainmenu():
    print_menu('请选择以下菜单号:', list(['登录学生信息管理系统', '退出学生信
息管理系统']))
    mc1 = int(input('输入菜单号: '))
    if mc1 == 1:
        login()
    elif mc1 == 2:
        print('感谢使用学生信息管理系统! ')
        quit()
mainmenu()

```

四、实验总结

在实验中有那些重要问题或者事件？你如何处理的？你的收获是什么？

收获:学会了使用 pymysql 编写数据库操作程序，通过高级语言实现数据库的增删改查等操作。