

数据库系统

实验指导书

班级：1504201

姓名：殷悦

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 PB8.0 以上版本和 SQL SERVER 2000 以上版本上机环境编程测试。

为了达到这个目的，这里安排五个实验单元。

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）

本次实验的主要目的在于学会使用 SQL SERVER 环境建库的方法，学会 SQL SERVER 企业管理器和查询分析器的使用，通过数据库操纵环境，了解库的建立、表的建立、数据库的查询实现方法。实验数据如下：

Student

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

Course

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

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

实 验 报 告 1

一. SQL 语句实验题目:

1. 建立课程表
2. 建立学生表
3. 建立选课表
4. 查询所有学生的详细信息
5. 查询 1 班的学生学号及姓名
6. 查询‘刘晨’的出生年
7. 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)
8. 查询选修了 1 号课的学生姓名、性别、成绩
9. 查询没有先行课的课程的课号和课程名
10. 查询 2 班的所有女生的情况
11. 查询学分为 2 到 3 之间的课程号及课程名
12. 查询选修 1 或 2 号课的学生的班号、学号、姓名、课程名及成绩
13. 查询 2 班至少选修一门其先行课为 1 号课的学生的班号、学号、姓名、性别、系、课程号及成绩
14. 查询 2 号科成绩最高的学生班号、学号
15. 查询 1 班 2 号课成绩最低的学生班号、学号
16. 查询选修 2 号课且成绩不是最低的同学班号、学号

二、学习掌握 SQL SERVER 2000 环境数据库备份方法

练习 SQL SERVER 数据库备份与恢复技术方法。

二. 源程序清单和结果

1.

E073.Hello - dbo.Course		E073.Hello - dbo.Student		SQLQuery1.sql - E0..
	cno	cname	spno	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 2-1

2.

E073.Hello - dbo.Student		SQLQuery1.sql - E0...dministrator (53))*		E073.Hello - dbo.Course		E073.Hello -
	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 2-2

3.

E073.Hello - dbo.SC		E073.Hello - dbo.Course		E073.Hello - dbo.Student
	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 2-3

4. select * from Student;

SQLQuery1.sql - E0...dministrator (53)*E073.Hello - dbo.SC

select * from Student;

结果消息

	sclass	sno	sname	ssex	sage	Sdept
1	1	1	李勇	男	20	IS
2	1	2	刘晨	女	19	IS
3	1	3	刘朋	男	20	IS
4	2	1	王敏	女	18	MA
5	2	2	张锋	男	19	MA
6	2	3	李敏	男	20	MA

图 2-4

5. select sno,sname from Student where sclass=1;

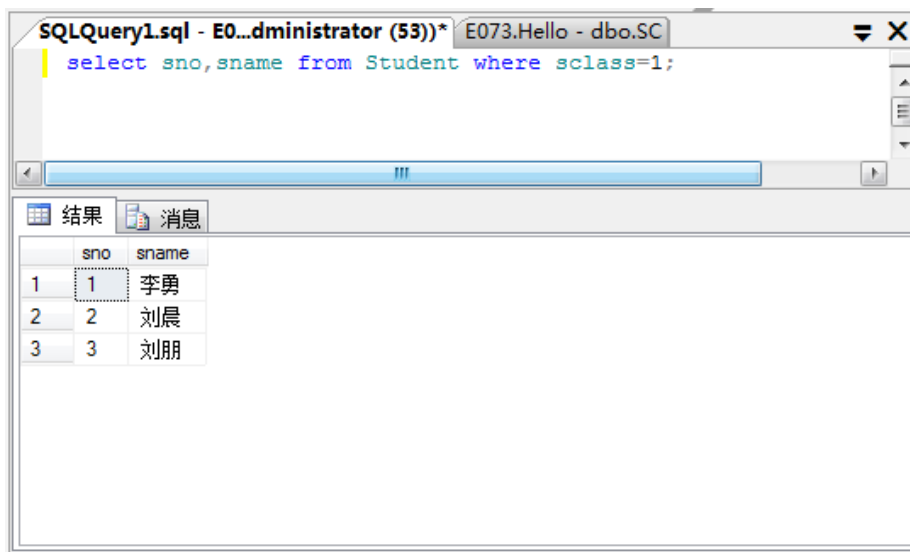


图 2-5

6. select 2017-sage Bir from Student where sname='刘晨';

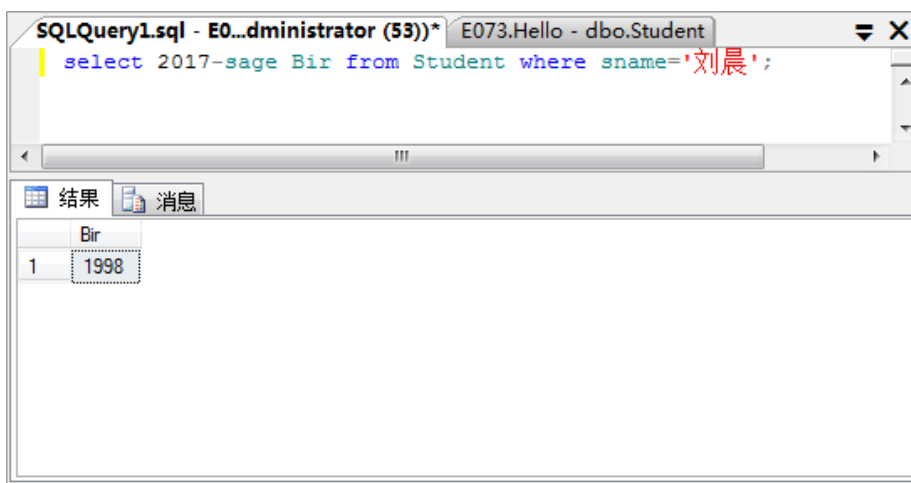


图 2-6

7.

select

Student.sclass,Student.sno,sname,Student.ssex,sage,Sdept,grade,cname,spno,ccredit from Student left join SC on (Student.sno=SC.sno and Student.sclass=SC.sclass) left join Course on(SC.cno=Course.cno)

where sname like '刘%';

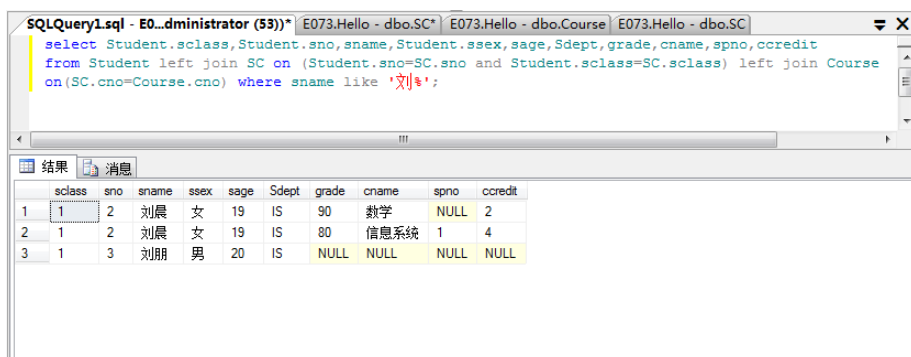


图 2-7

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

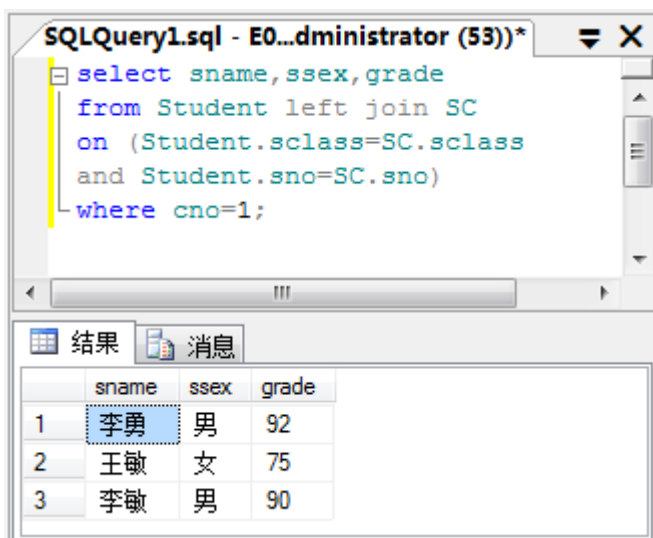


图 2-8

9. select cno,cname from Course where spno is null;

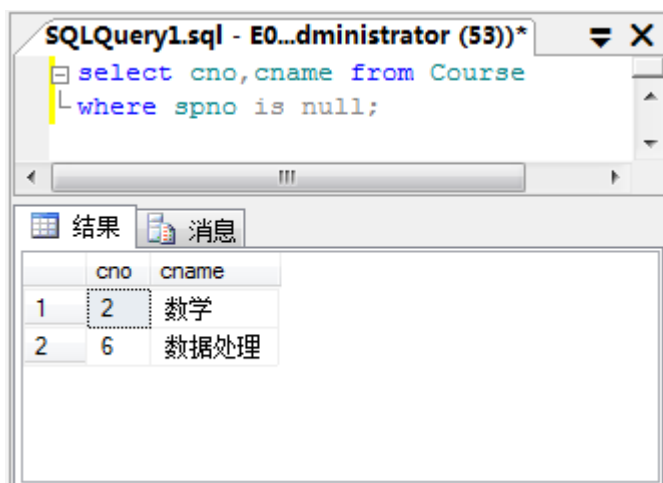


图 2-9

10. `select * from Student where sclass=2 and ssex='女';`

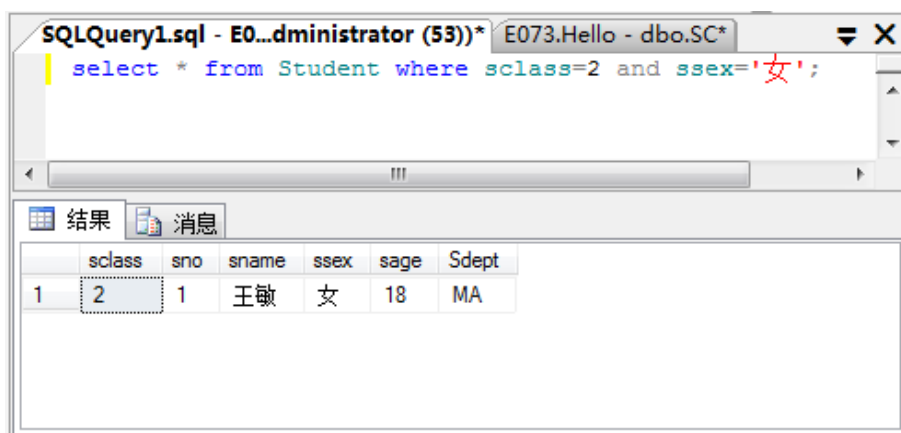


图 2-10

11. `select cno,cname from Course where ccredit<=3 and ccredit>=2;`

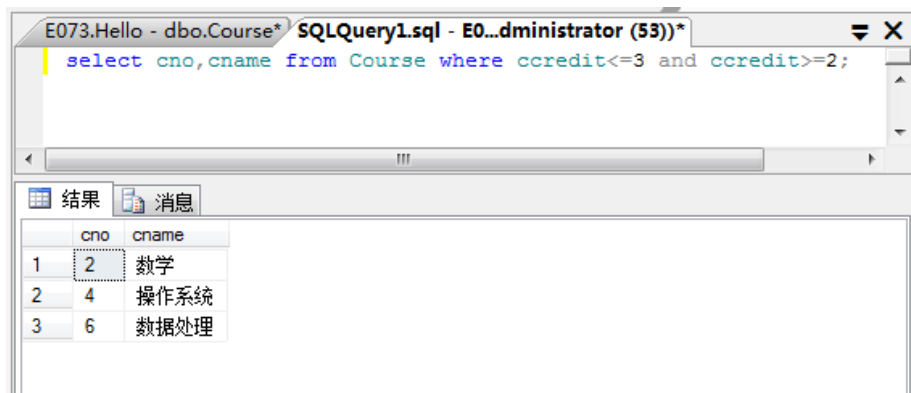


图 2-11

12.

```
select Student.sclass,Student.sno,sname,cname,grade from Student left
join SC on (Student.sno=SC.sno and Student.sclass=SC.sclass) left
join Course on(SC.cno=Course.cno) where SC.cno=1 or SC.cno=2;
```

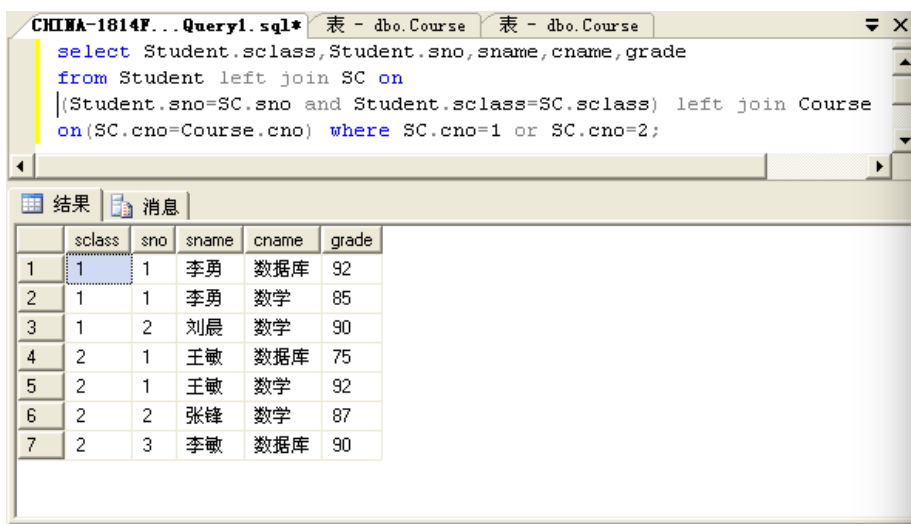


图 2-12

13. select

```
Student.sclass,Student.sno,sname,
Student.ssex,Sdept,SC.cno,grade
from Student left join SC
```

on (Student.sno=SC.sno and Student.sclass=SC.sclass)

left join Course on(SC.cno=Course.cno)

where exists

(select * from SC where cno in(

select cno from Course where spon=1));

14. select sclass,sno from SC where cno=2 and

grade=(select max(grade) from SC where cno=2);

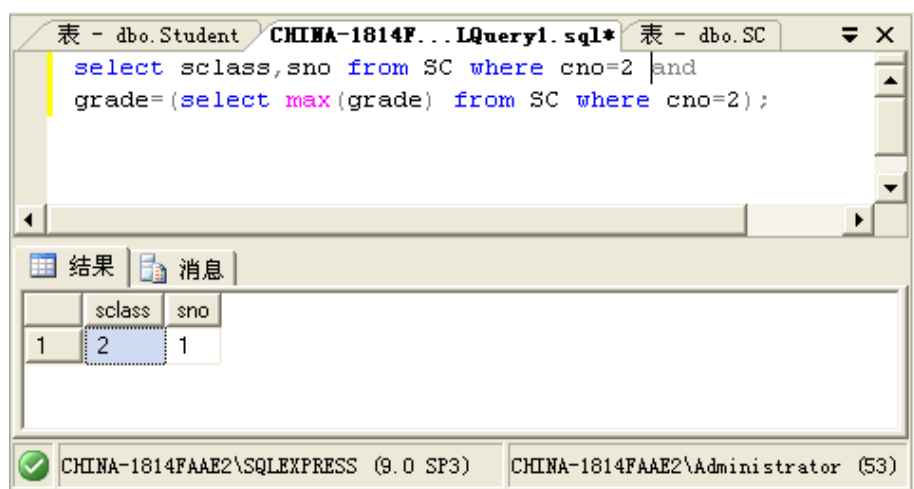


图 2-14

15. select sclass,sno from SC where sclass=1 and cno=2 and

grade=(select min(grade) from SC where sclass=1 and cno=2);

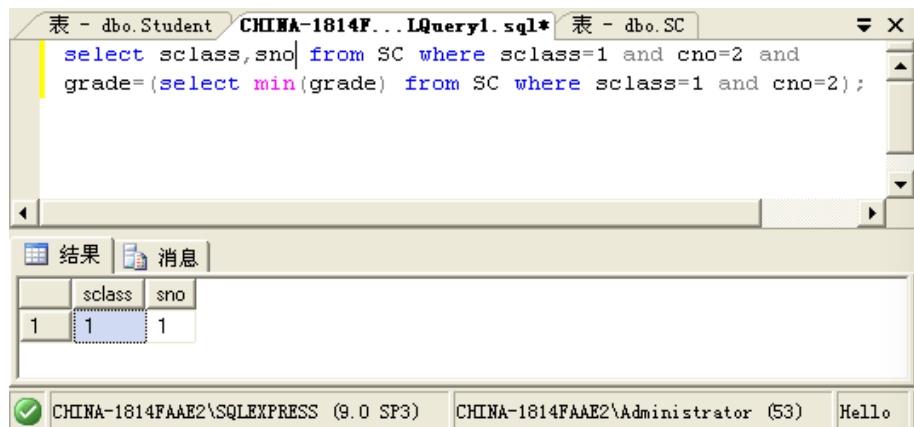
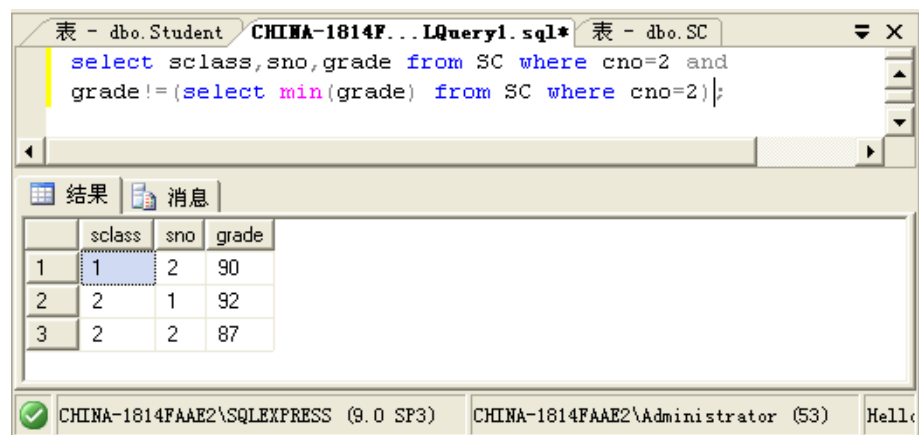


图 2-15

16. select sclass,sno,grade from SC where cno=2 and grade!=(select min(grade) from SC where cno=2);



The screenshot shows a SQL query execution window with the following components:

- Query Text:** `select sclass,sno,grade from SC where cno=2 and grade!=(select min(grade) from SC where cno=2);`
- Results Tab:** Displays a table with 3 rows and 4 columns (sclass, sno, grade).
- Message Tab:** Empty.
- Status Bar:** Shows connection details for CHINA-1814FAAE2\SQLEXPRESS (9.0 SP3) and CHINA-1814FAAE2\Administrator (53).

	sclass	sno	grade
1	1	2	90
2	2	1	92
3	2	2	87

图 2-16

实验 2: 学习 SQL 语句的数据操纵(4 学时)

本次实验的主要目的在于学会使用 SQL 语言的增、删、改、查功能、多表间的连接操作、分组及视图操作。

实验数据如下:

Student

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

Course

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

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

实 验 报 告 2

一. 数据库 SQL 语言增、删、改、查功能

1. 查询选了 1 号课且选了 2 号课的学生的班号、学号
2. 查询选了 1 号课但不选 2 号课的学生的班号、学号
3. 查询 1 班平均分在 85 分以上的同学班号、学号、姓名、性别、系、各科课程号及成绩
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
5. 查询不选 1 号课的学生班号及学号
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
7. 统计学生选修课程的班号、学号及总学分
8. 统计 1 班选修 3 号课的学号及平均分
9. 把个人信息及选课信息插入到 Student 和 SC 表及新增一门“无机化学”课程信息
10. 删除选修 3 号课的所有选课信息并显示删除后的结果
11. 把选修 1 号课的所有男同学年龄增加 1 岁并显示最终学生 Student 信息
12. 把每个选课人的学号、班号及平均成绩插入到一个新表中。

二、视图 SQL 语言功能

- 1 使用企业管理器创建视图：在 ST 库中以“student”表为基础，建立信息系学生的视图 V_ISStudent
- 2 使用 SQL 语句创建视图：
 - ① 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;
 - ② 建立信息系建立信息系选修了 1 号课程且成绩在 90 分以上的学生的视图 V_IS_Score
 - ③ 将各系学生人数，平均年龄定义为视图 V_NUM_AVG。
- 3 查询以上所建的视图结果
- 4 查询选修了 1 号课程的信息系学生
- 5 在信息系学生的视图中找出年龄小于 20 岁的学生

6 将信息系学生视图 V_ISStudent 中学号一班 2 号的学生姓名改为“刘辰”

7 用 SQL 语句删除视图 S_C_GRADE

三、源程序代码和结果

1. select A.sclass,A.sno from SC A inner join SC B on (A.sclass=B.sclass and A.sno=B.sno) where A.cno=1 and B.cno=2;

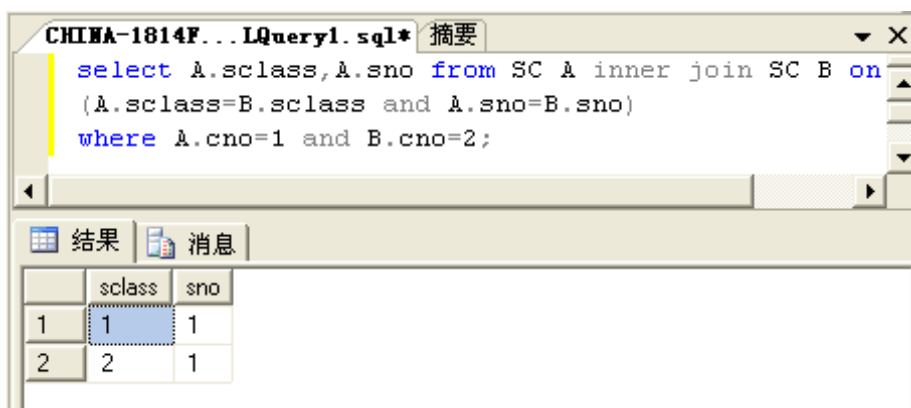


图 3-1

2. select sclass,sno from SC A where cno=1 and not exists (select * from SC where A.sclass=sclass and A.sno=sno and cno=2);

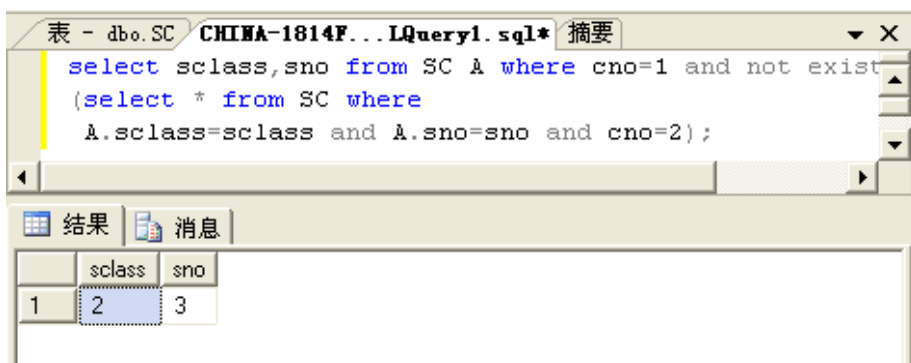


图 3-2

3. select SC.sclass,SC.sno,sname,ssex,sdept,cno,grade from SC right join (select Student.sclass,Student.sno,sname,ssex,sdept from Student right join (select sclass,sno from SC group by sclass,sno having

avg(grade)>85) as G on (Student.sclass=G.sclass and Student.sno=G.sno)) as E on (SC.sclass=E.sclass and SC.sno=E.sno);

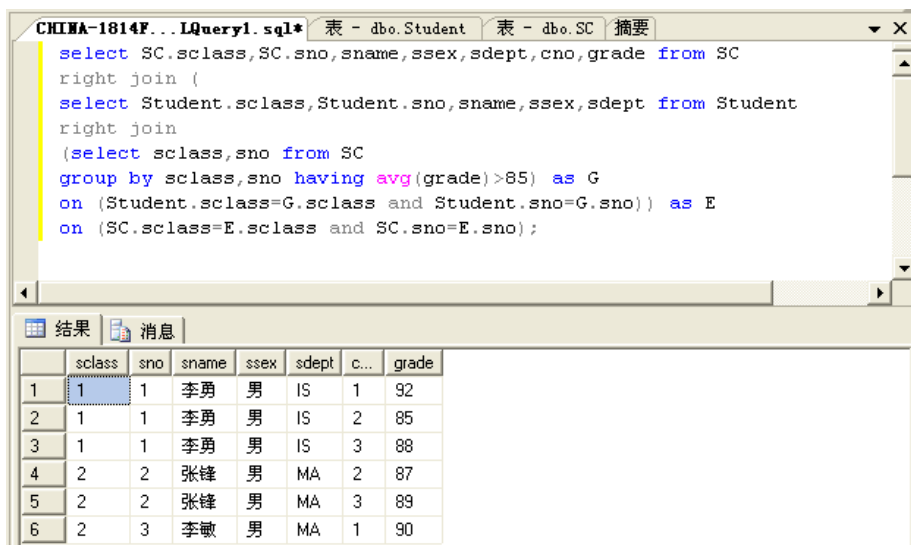


图 3-3

4. use hello;

select sclass,sno,sname from student 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=student.sno and sc1.cno=sc2.cno

))

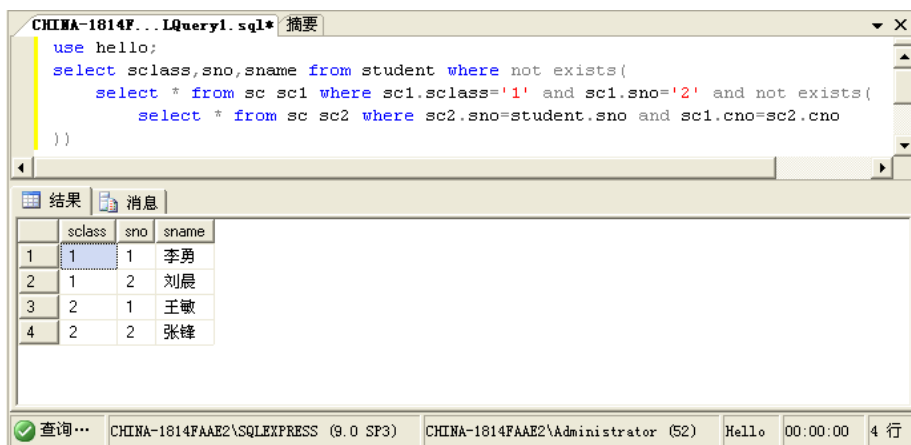


图 3-4

5. select sname,grade from SC left join Student on
 (SC.sclass=Student.sclass and SC.sno=Student.sno) where cno=2
 order by grade desc;

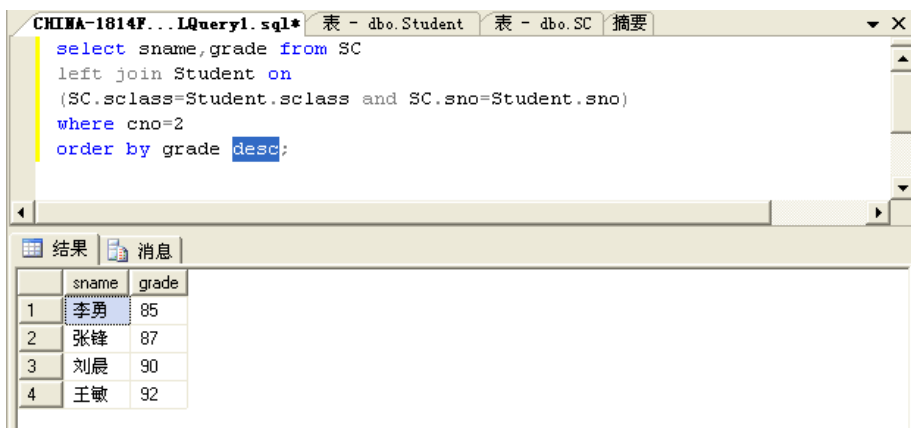


图 3-5

6. select sclass,sno,sum(sum(grade) sum from SC group by sclass,sno;

CHINA-1814F...LQuery1.sql* 表 - dbo.Student 表 - dbo.SC 摘要

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

结果 消息

	sclass	sno	sum
1	1	1	265
2	2	1	167
3	1	2	170
4	2	2	176
5	2	3	90

图 3-6

7. select sclass,avg(grade) arg from SC where sclass=1 and cno=1
group by sclass;

CHINA-1814F...LQuery1.sql* 摘要

```
select sclass,avg(grade) arg  
from SC  
where sclass=1 and cno=1  
group by sclass;
```

结果 消息

	sclass	arg
1	1	92

图 3-7

8. insert into

Student(sclass,sno,sname,ssex,sage,Sdept)

values(1,4,'老王','男',30,'SI');

insert into

Course(cno,cname,spno,ccredit)

values(8,'无机化学','2',4);

insert into

SC(sclass,sno,cno,grade)

```
values(1,4,8,100);
```

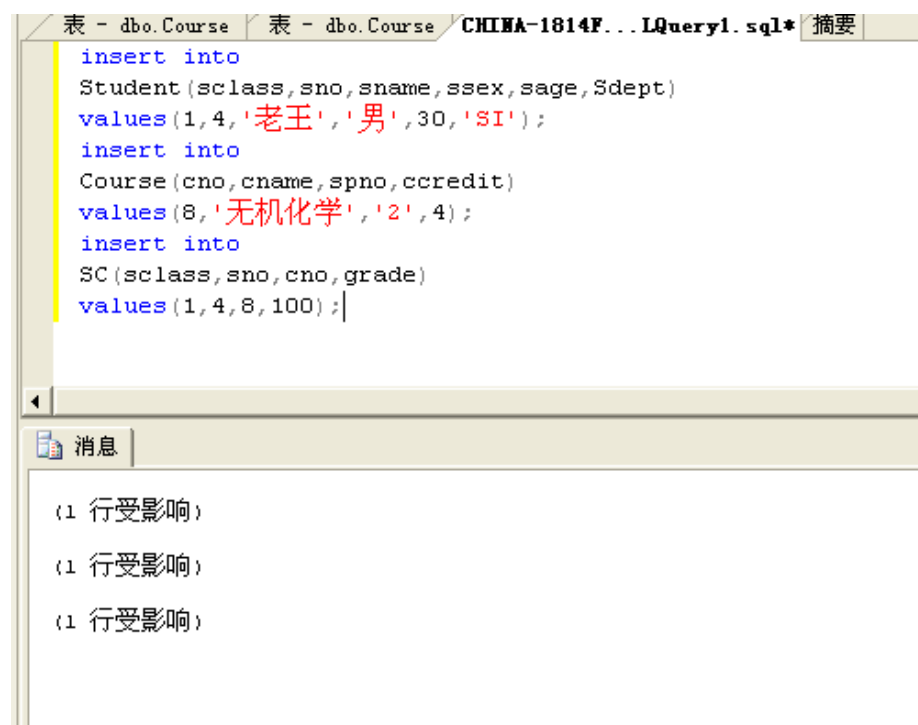
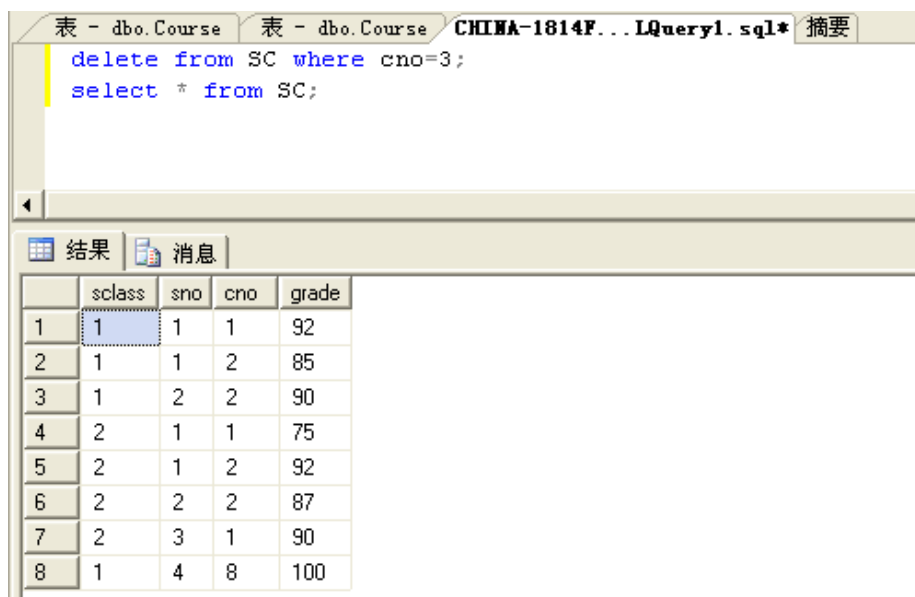


图 3-8

```
9. delete from SC where cno=3;
```

```
select * from SC;
```



The screenshot shows a SQL query window with the following content:

```
delete from SC where cno=3;  
select * from SC;
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	sclass	sno	cno	grade
1	1	1	1	92
2	1	1	2	85
3	1	2	2	90
4	2	1	1	75
5	2	1	2	92
6	2	2	2	87
7	2	3	1	90
8	1	4	8	100

图 3-9

10. update Student

```
set sage=sage+1
```

```
where ssex='男' and exists(
```

```
select * from SC where Student.sclass=SC.sclass and
```

```
Student.sno=SC.sno and cno=1);
```

```
select * from Student;
```

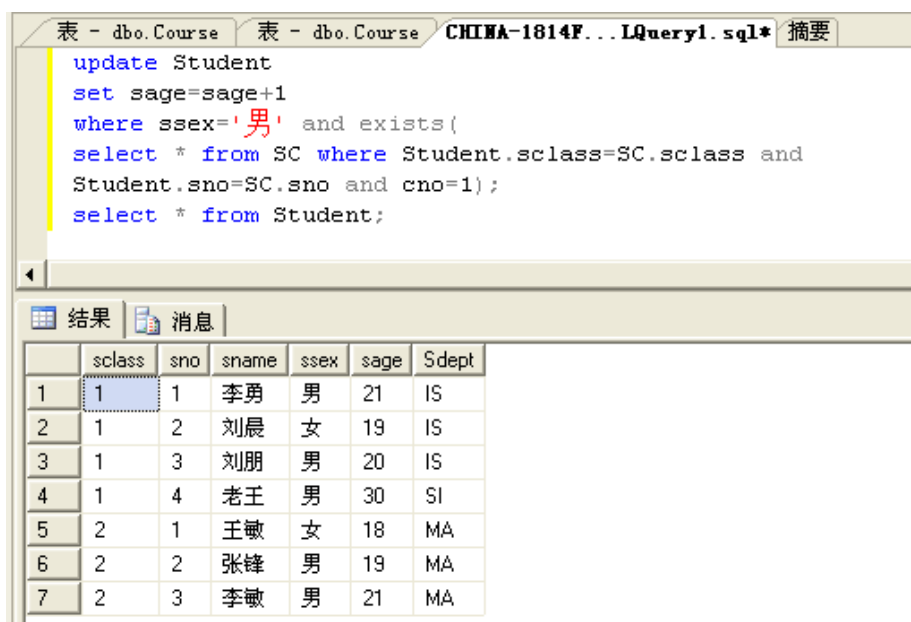


图 3-10

二. 1 使用企业管理器创建视图：在 ST 库中以“student”表为基础，建立信息系学生的视图 V_ISStudent

```
create view V_ISStudent (sclass,sno,sname,ssex,sage,Sdept) as
select sclass,sno,sname,ssex,sage,Sdept from Student where
Sdept='IS';
```

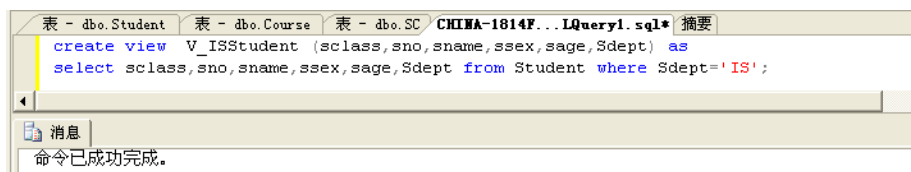


图 3-11

表 - dbo.Student 表 - dbo.Course 表

```
select * from V_ISStudent;
```

结果 消息

	sclass	sno	sname	ssex	sage	Sdept
1	1	1	李勇	男	21	IS
2	1	2	刘晨	女	19	IS
3	1	3	刘朋	男	20	IS

图 3-12

2 使用 SQL 语句创建视图：

① 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(sno,sclass,sname,cname,grade)
as
select Student.sno,Student.sclass,Student.sname,cname,grade
from Student
left join SC on Student.sno=SC.sno and Student.sclass=SC.sclass
left join Course on SC.cno=Course.cno;
```

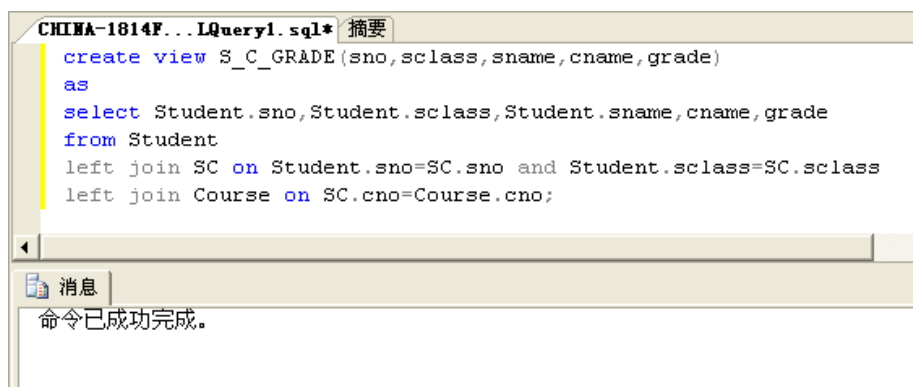


图 3-13

CHINA-1814F...LQuery1.sql* 摘要					
select * from S_C_GRADE;					
结果 消息					
	sno	sclass	sname	cname	grade
1	1	1	李勇	数据库	92
2	1	1	李勇	数学	85
3	2	1	刘晨	数学	90
4	3	1	刘朋	NULL	NULL
5	4	1	老王	无机化学	100
6	1	2	王敏	数据库	75
7	1	2	王敏	数学	92
8	2	2	张锋	数学	87
9	3	2	李敏	数据库	90

图 3-14

② 建立信息系建立信息系选修了 1 号课程且成绩在 90 分以上的学生的视图 V_IS_Score

```

SELECT          dbo.Student.sclass,      dbo.Student.sno,
dbo.Student.sname,      dbo.Student.ssex,      dbo.Student.sage,
dbo.Student.Sdept, dbo.Course.cname,
                dbo.Course.cpno,      dbo.Course.ccredit,
dbo.SC.grade
FROM            dbo.Student LEFT OUTER JOIN
                dbo.SC      ON      dbo.Student.sclass      =
dbo.SC.sclass AND dbo.Student.sno = dbo.SC.sno LEFT OUTER
JOIN
                dbo.Course      ON      dbo.SC.cno      =
dbo.Course.cno
WHERE          (dbo.Student.Sdept = 'IS') AND (dbo.SC.grade > 90)

```

视图 - dbo.V_IS_Score* 视图 - dbo.V_IS_Score CHINA-1814F...QLQuery1.sql 摘要

SELECT dbo.Student.sclass, dbo.Student.sno, dbo.Student.sname, dbo.Student.ssex, dbo.Student.sage, dbo.Student.Sdept, dbo.Course.cname, dbo.Course.cno, dbo.Course.ccredit, dbo.SC.grade
FROM dbo.Student LEFT OUTER JOIN
dbo.SC ON dbo.Student.sclass = dbo.SC.sclass AND dbo.Student.sno = dbo.SC.sno LEFT OUTER JOIN
dbo.Course ON dbo.SC.cno = dbo.Course.cno
WHERE (dbo.Student.Sdept = 'IS') AND (dbo.SC.grade > 90)

列	别名	表	输出	排序类型	排序顺序	筛选器	或...	或
sclass		Student	✓					
sno		Student	✓					
sname		Student	✓					
ssex		Student	✓					

class	sno	sname	ssex	sage	Sdept	cname	cpno	ccredit	grade
1	1	李勇	男	21	IS	数据库	3	4	92
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 3-15

③ 将各系学生人数, 平均年龄定义为视图 V_NUM_AVG。
create view V_NUM_AVG(Sdept,sum_num,avr_age)

as

select Sdept,count(sno),avg(sage)

from Student

group by Sdept;

表 - dbo.Student CHINA-1814F...Query1.sql* 摘要

```
create view V_NUM_AVG (Sdept,sum_num,avr_age)
as
select Sdept,count(sno),avg(sage)
from Student
group by Sdept;
```

消息
命令已成功完成。

图 3-16

表 - dbo.Student CHINA-1814F...Query1.sql* 摘要

```
select * from V_NUM_AVG;
```

结果 消息

	Sdept	sum_num	avr_age
1	IS	3	19
2	MA	3	19

图 3-17

3 查询以上所建的视图结果

1-1

表 - dbo.Student 表 - dbo.Course 表

```
select * from V_ISStudent;
```

结果 消息

	sclass	sno	sname	ssex	sage	Sdept
1	1	1	李勇	男	21	IS
2	1	2	刘晨	女	19	IS
3	1	3	刘朋	男	20	IS

图 3-18

2-1

CHINA-1814F... LQuery1.sql* 摘要

```
select * from S_C_GRADE;
```

结果 消息

	sno	sclass	sname	cname	grade
1	1	1	李勇	数据库	92
2	1	1	李勇	数学	85
3	2	1	刘晨	数学	90
4	3	1	刘朋	NULL	NULL
5	4	1	老王	无机化学	100
6	1	2	王敏	数据库	75
7	1	2	王敏	数学	92
8	2	2	张锋	数学	87
9	3	2	李敏	数据库	90

图 3-19

2-2

CHINA-1814F... LQuery1.sql* 摘要

```
select * from V_IS_Score;
```

结果 消息

	sclass	sno	sname	ssex	sage	Sdept	cname	cp...	cce...	grade
1	1	1	李勇	男	21	IS	数据库	5	4	92

图 3-20

2-3

CHINA-1814F... LQuery1.sql* 摘要

```
select * from V_NUM_AVG;
```

结果 消息

	sclass	sno	sname	ssex	sage	Sdept	cname	cp...	cce...	grade
1	1	1	李勇	男	21	IS	数据库	5	4	92

图 3-21

4 查询选修了 1 号课程的信息系学生

```
select Student.sclass,Student.sno,sname,grade from SC
left join Student on SC.sno=Student.sno and
SC.sclass=Student.sclass
left join Course on SC.cno=Course.cno
where SC.cno=1 and Sdept='IS';
```

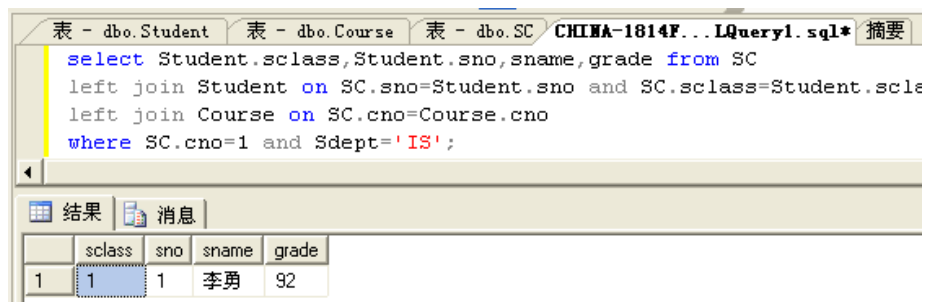


图 3-22

5 在信息系学生的视图中找出年龄小于 20 岁的学生

```
select * from V_ISStudent where sage<20;
```

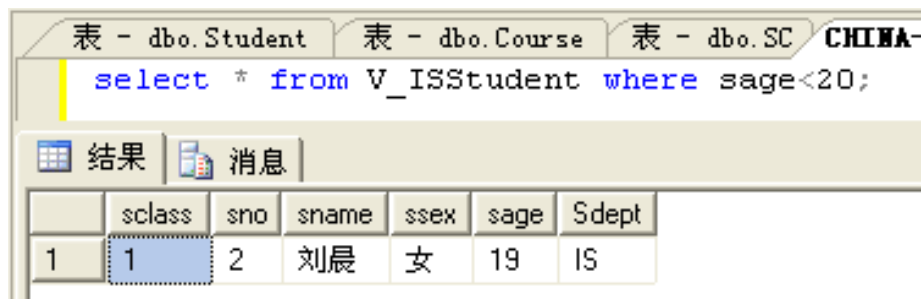


图 3-23

6 将信息系学生视图 V_ISStudent 中学号一班 2 号的学生姓名改为“刘辰”

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

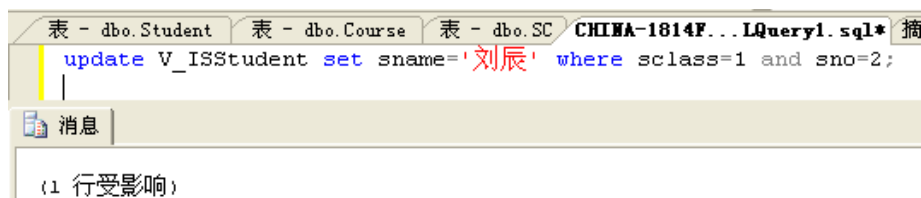
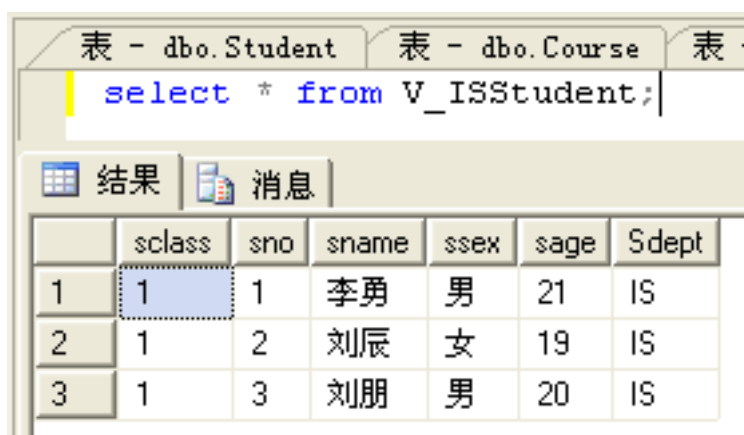


图 3-24

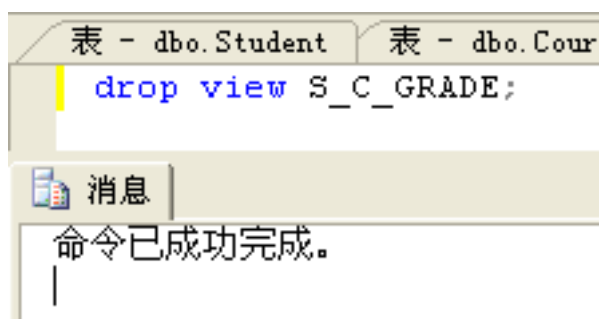


The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query: `select * from V_ISStudent;`. The bottom pane shows the results of this query in a table format. The table has seven columns: `sclass`, `sno`, `sname`, `ssex`, `sage`, and `Sdept`. There are three rows of data.

	sclass	sno	sname	ssex	sage	Sdept
1	1	1	李勇	男	21	IS
2	1	2	刘辰	女	19	IS
3	1	3	刘朋	男	20	IS

图 3-25

7 用 SQL 语句删除视图 S_C_GRADE



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query: `drop view S_C_GRADE;`. The bottom pane shows the message: "命令已成功完成。" (Command completed successfully.)

表 - dbo.Student	表 - dbo.Cour
<code>drop view S_C_GRADE;</code>	
消息	
命令已成功完成。	

图 3-26

实验 3：学习 PB 与 SQL SEREVR 数据库的连接方法

本次实验的主要目的是掌握 PB8.0 以上环境及语句的使用，学习 PB 与 SQL SEREVR 数据库的连接方法与编程技巧。

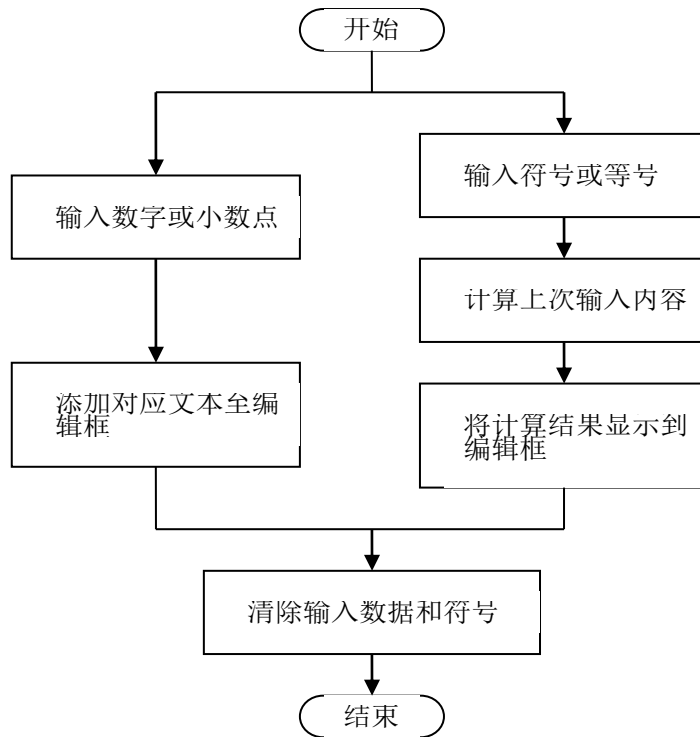
实 验 报 告 3

一、题目

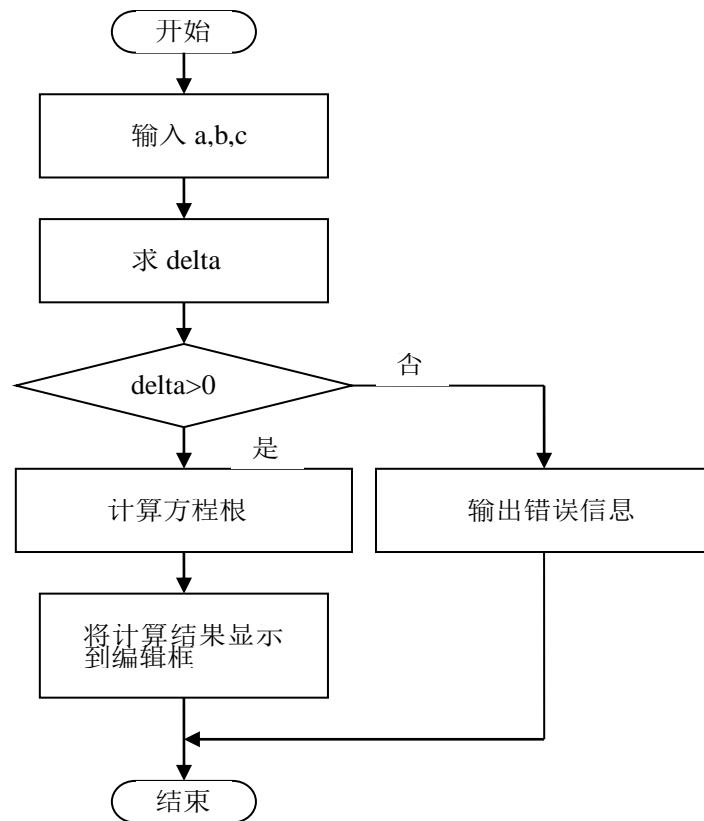
题目见 PB 实用教程（第 2、3、4 版）中实验 2、实验 4 及实验 5，编程序，数据库采用 SQL SERVER，不采用书中 ASA 数据库，然后上机连接调试运行。

二、画出每个实验功能图

计算器功能图：



计算方程根：



三、源程序清单和结果

实验二：

2-1.自制计算器

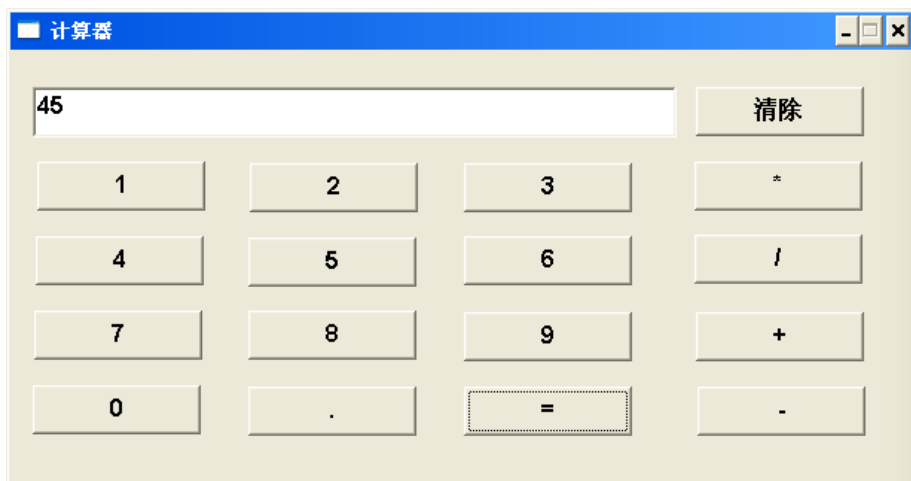


图 2-1

定义变量

Decimal add1

Char s1

Boolean flag

数字和点代码

IF flag THEN

sle_1.text=""

flag=false

END IF

sle_1.text=sle_1.text+THIS.text//给编辑框添加文本

运算符号代码

CHOOSE CASE s1//根据符号执行计算

CASE '*'

sle_1.text=String(add1*dec(sle_1.text))

CASE '/'

sle_1.text=String(add1/dec(sle_1.text))


```
CASE '+'
    sle_1.text=String(add1+dec(sle_1.text))
CASE '-'
    sle_1.text=String(add1 - dec(sle_1.text))
END CHOOSE
Add1=Dec(sle_1.text)
s1=THIS.text
flag=true
等号代码
CHOOSE CASE s1//根据符号执行计算
    CASE '*'
        sle_1.text=String(add1*dec(sle_1.text))
    CASE '/'
        sle_1.text=String(add1/dec(sle_1.text))
    CASE '+'
        sle_1.text=String(add1+dec(sle_1.text))
    CASE '-'
        sle_1.text=String(add1 - dec(sle_1.text))
END CHOOSE
flag=true
s1=""
清除代码
sle_1.text=""
Add1=0
s1=""
```

flag=true

2-2.解方程

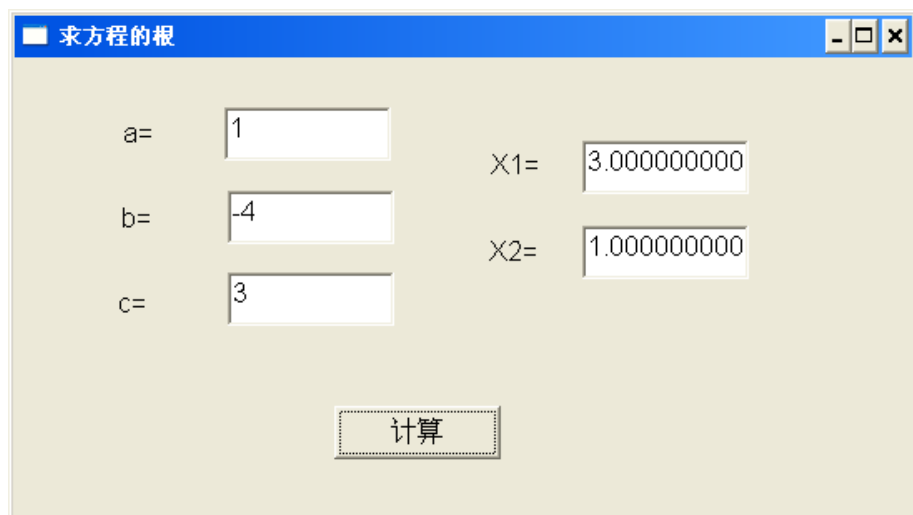


图 2-2

定义变量：

Decimal a

Decimal b

Decimal c

Decimal delta

Decimal x1

Decimal x2

代码：

```
a=dec(sle_a.text)
```

```
b=dec(sle_b.text)
```

```
c=dec(sle_c.text)
```

```
delta=b*b - 4*a*c//求 delta
```

```

sle_x1.text=String(delta)

if delta < 0 then//delta 小于零无根

    messagebox("error","delta is less than zero!")

    return

end if

x1=(sqrt(delta) - b)/2*a
x2=(- sqrt(delta) - b)/2*a
sle_x1.text=String(x1)
sle_x2.text=String(x2)

```

实验四：数据库连接

数据库连接截图：

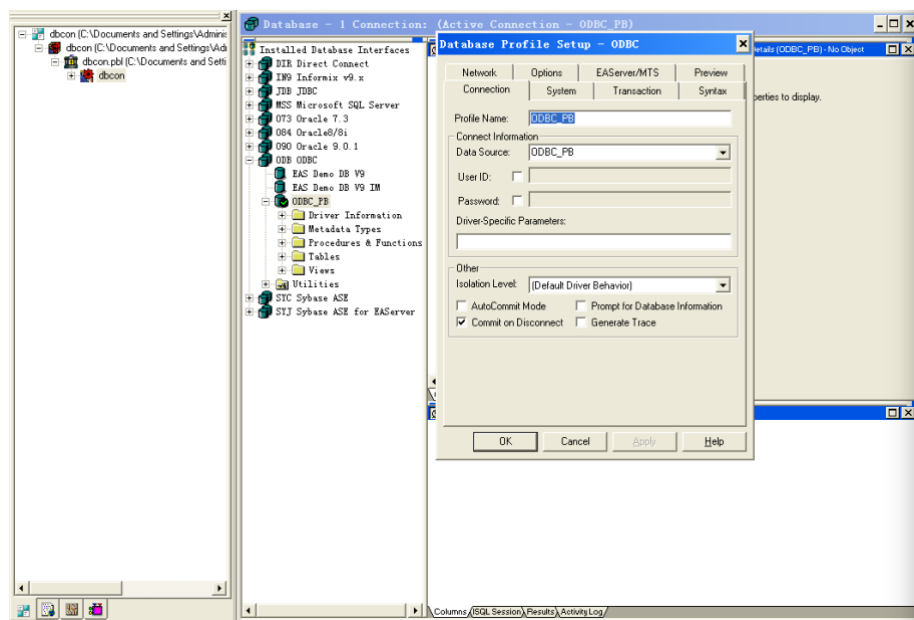


图 2-3

数据库表项：

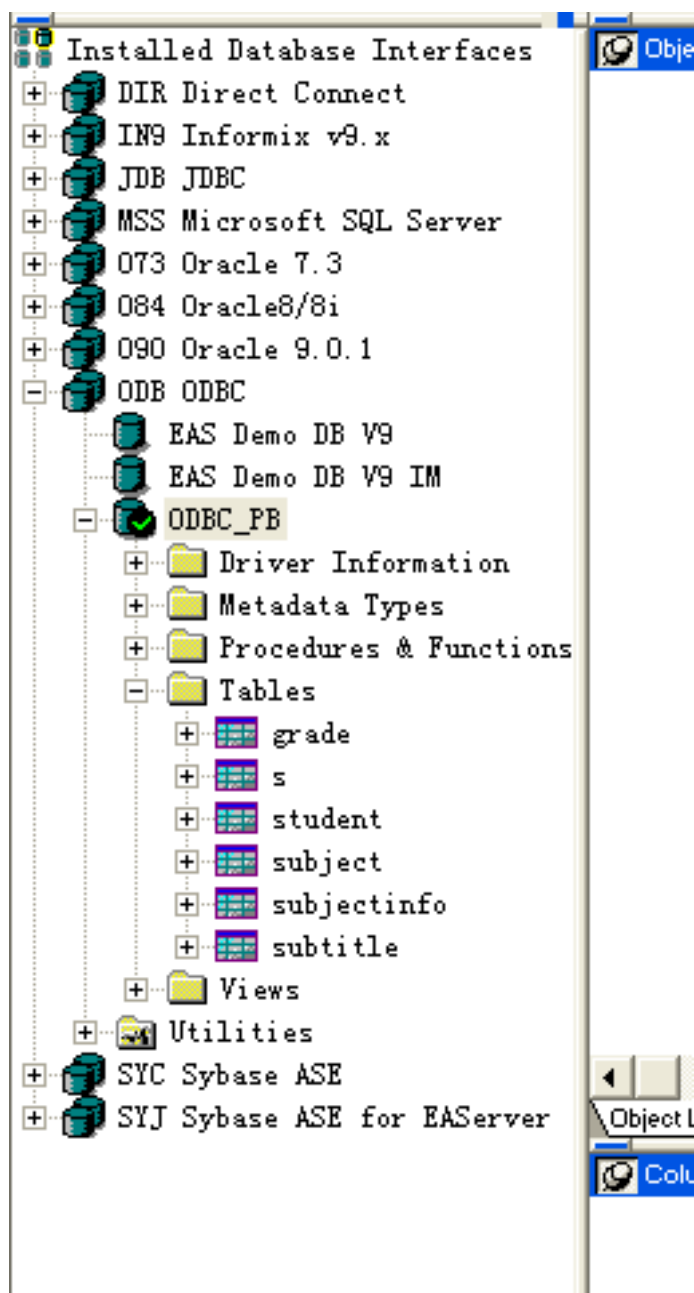


图 2-4

ODBC 连接:

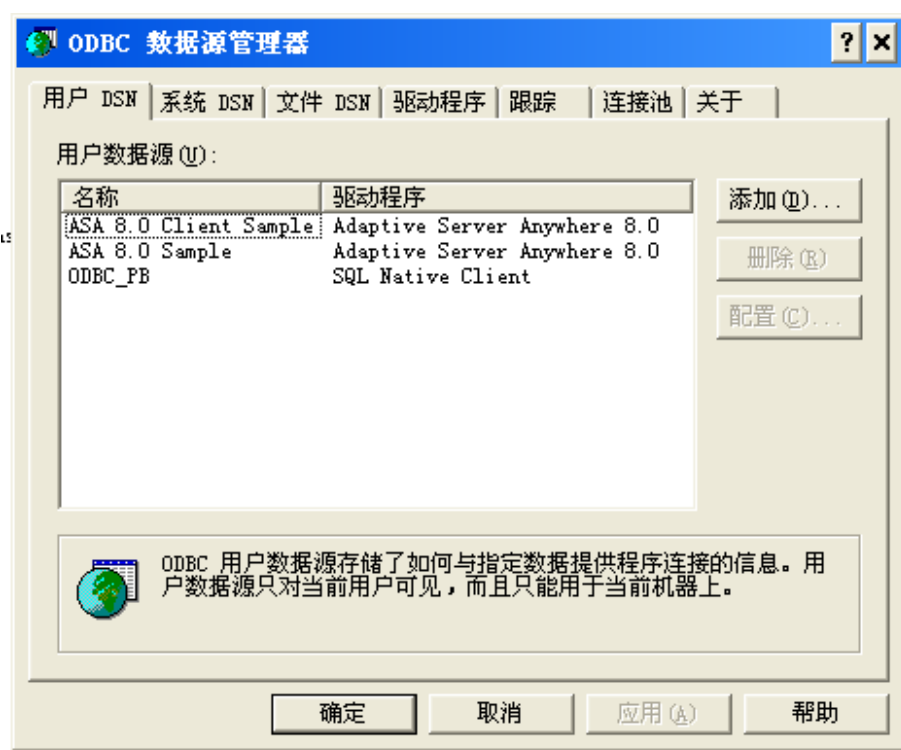


图 2-5

实验五：学生管理系统



图 2-6

数据库连接代码：

SetPointer(hourglass!)

//调用下面的函数

IF parent.wf_connect(sle_userid.text,sle_password.text)=-1 THEN

 MessageBox("连接数据库错误!", "连接失败"+sqlca.sqlerrtext)

 HALT

ELSE

 CLOSE(PARENT)

 Open(w_1)

END IF

取消代码：

//关闭窗口

HALT

wf_connect 函数：

//连接数据库

String ls_database

userid=Trim(userid)

password=Trim(password)

IF password="" THEN RETURN -1

ls_database="ConnectionString='DSN=ODBC_PB;'"

sqlca.dbparm=ls_database+"UID="+userid+";PWD="+password+""

CONNECT USING sqlca;

RETURN sqlca.SQLCode

// Profile ODBC_PB

```
//SQLCA.DBMS = "ODBC"
```

```
//SQLCA.AutoCommit = False
```

```
//SQLCA.DBParm = "ConnectionString='DSN=ODBC_PB'"
```

登录成功

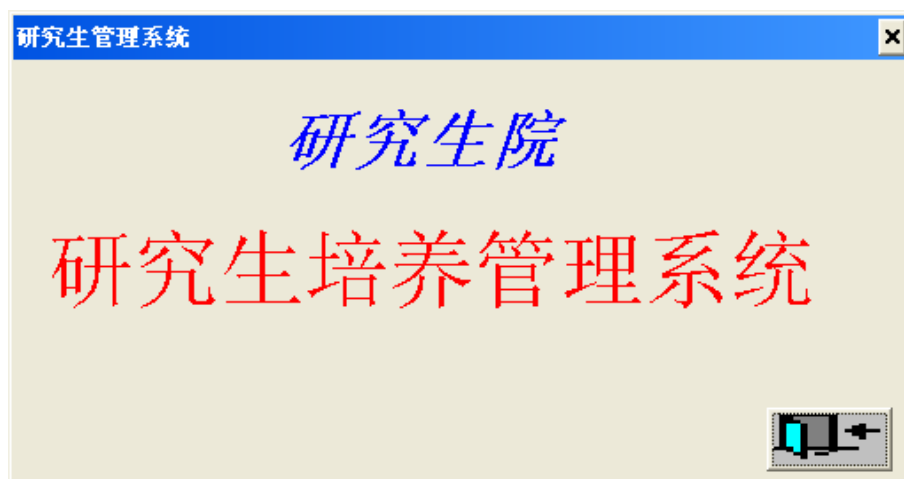


图 2-7

退出 w_1 窗口退出按钮

```
Close(Parent)
```

应用程序 Open

```
sqlca.DBMS="ODBC"
```

```
Open(w_2)
```

应用程序 Close

```
Disconnect using SQLCA;
```

实验4 学习嵌入SQL及PB常用功能控件的 编程使用（4学时）

本次实验的主要目的是掌握嵌入SQL及PB常用功能控件，学会使用嵌入SQL对数据库进行单表精确查询、模糊查询的方法。

实 验 报 告 4

一、题目：

见PB实用教程（第2、3、4版）中实验6，另加如下编程题目。编程序，然后上机调试运行。采用嵌入SQL语言用游标实现如下界面功能的查询。

1、建立学生表及系表：
S(SNO, SNAME, SSEX, SAGE, DNO, SADDR), DEPT(DNO, DEPARTMENT)

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

二、需求分析设计

实验六：

1. 功能分析

(1) 用户分析：设置管理员登录界面

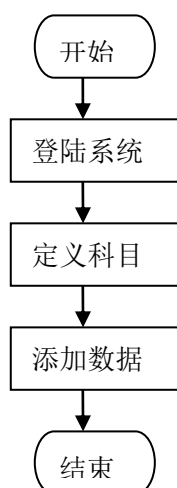
(2) 数据库连接：构造所需数据库

(3) 实现学生信息的增加：

学号，姓名，民族，出生日期，家庭住址，联系电话，个人简介，性别，党员信息等信息添加

2. 系统基本信息流程和模型

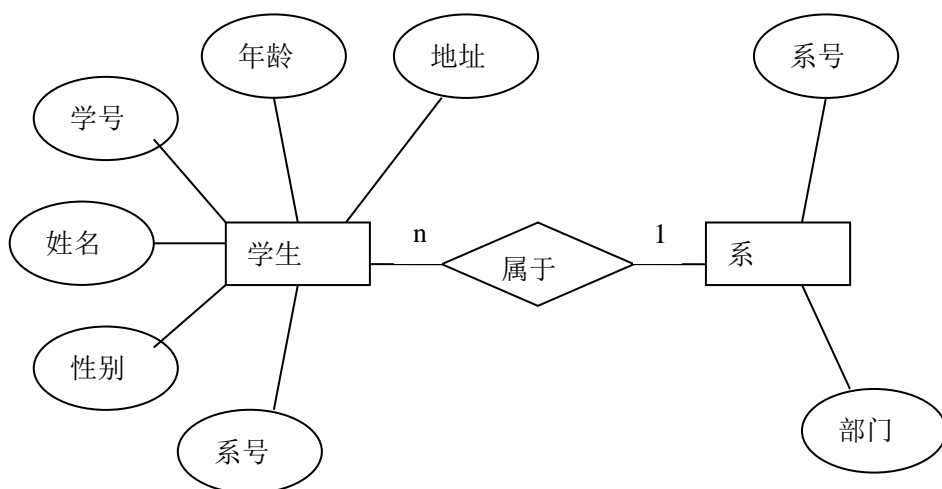
(1) 学生信息处理功能模型：



E-R:

查询信息：

E-R 图



表名：S							
字段名	标题	字段类	默认值	必填	允许空	大小	描述
SNO	学号	整型	0	是	否		学生唯一标识
SNAME	姓名	文本		是	是	4	学生姓名
SAGE	年龄	整型	0	是	是		学生年龄
DNO	系号	整型	0	是	否		学生所属系
SADDR	地址	文本		是	是	50	学生所在地址

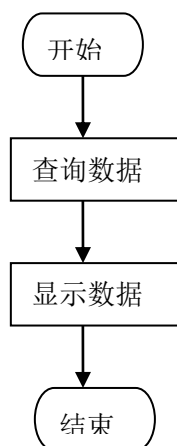
表名：DEPT							
字段名	标题	字段类	默认值	必填	允许空	大小	描述
DNO	系号	整型	0	是	否		系唯一标识
DEPARTMENT	部门	文本		是	是	10	系名

1. 功能分析

- (1) 用户分析：搜索查看信息
- (2) 数据库连接：构造所需数据库

2. 系统基本信息

- (1) 学生信息处理功能模型：



三、源程序清单和结果

实验六：

1. 登录界面



图 3-1

登录代码同上

2. 插入信息

图 3-2

ddlNation 的 constructor 函数：

//窗口打开时为下拉菜单添加数据

THIS.additem("汉族");

THIS.additem("回族");

THIS.additem("维吾尔族");

THIS.additem("藏族");

THIS.additem("苗族");

THIS.additem("黎族");

退出按钮：

close(Parent)

确定按钮：

Date ld_birthday

String ls_name,ls_nation,ls_sex,ls_home,ls_tel,ls_party

//输入合法行检查

IF sle_id.text="" OR isNull(sle_id.text)THEN

 MessageBox("缺少数据","请输入学号")

 sle_id.setfocus()

 RETURN

ELSEIF sle_name.text="" or isNull(sle_name.text) THEN

 MessageBox("缺少数据","请输入学生姓名")

 sle_name.setfocus()

 RETURN

END IF

IF sle_birthday.text<>"" and not isNull(sle_birthday) THEN

 IF isDate(sle_birthday.text)THEN

 ld_birthday=date(sle_birthday.text)

 ELSE

 MessageBox("输入数据格式错误","请使用“年-月-日”的日期格式")

 sle_birthday.setfocus()

 RETURN

 END IF

END IF

ll_id=long(sle_id.text)

//判断选项框

IF rb_cy.checked=TRUE THEN

 ls_party=rb_cy.text

```
ELSEIF rb_cp.checked=TRUE THEN
```

```
    ls_party=rb_cp.text
```

```
ELSE
```

```
    ls_party=rb_none.text
```

```
END IF
```

```
IF rb_man.checked=TRUE THEN
```

```
    ls_sex=rb_man.text
```

```
ELSEIF rb_woman.checked =TRUE THEN
```

```
    ls_sex=rb_woman.text
```

```
END IF
```

```
IF ddlb_nation.text="none" THEN
```

```
    ls_nation=""
```

```
ELSE
```

```
    ls_nation=trim(ddlb_nation.text)
```

```
END IF
```

```
//查询学号是否有重复
```

```
SELECT student.stud_id
```

```
    INTO :ll_i
```

```
    FROM student
```

```
    WHERE student.stud_id=:ll_id;
```

```
IF ll_i<>0 THEN
```

```
    MessageBox("错误信息","学号第"+String(ll_i)+"号重号!请改正.")
```

```
sle_id.setFocus()

Return

END IF

ls_name=trim(sle_name.text)
ls_home=trim(sle_home.text)
ls_tel=trim(sle_tel.text)
//插入数据库
INSERT INTO "student"
(   "stud_id",
    "name",
    "birthday",
    "sex",
    "nation",
    "home",
    "tel",
    "party",
    "resume")
VALUE
(   :ll_id,
    :ls_name,
    :ld_birthday,
    :ls_sex,
    :ls_nation,
    :ls_home,
    :ls_tel,
```



```

:ls_party
:mle_resume.text);
//初始化窗口状态
sle_id.text=""
sle_name.text=""
sle_birthday.text=""
mle_resume.text=""
sle_home.text=""
sle_tel.text=""
sle_id.setFocus()

```

表 - dbo.Student		表 - dbo.Student		表 - dbo.S		摘要			▼ X	
	stud_id	name	birthdat	sex	nation	home	tel	party	resume()	
▶	150120526	殷悦	1996-12-...	男	汉族	山东威海 ...	17863137381	团员 ...	数据库测试	
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

图 3-3

3.搜索信息

结果

结果显示列表

学号	姓名	性别	年龄	地址
150120526	殷悦	男	22	威海

输入家庭地址

威海

查询

退出

图 3-4

ODBC 和数据库连接



图 3-5

查询按钮函数：

```
//打开数据库
```

```
// Profile ODBC_PB
```

```
SQLCA.DBMS = "ODBC"
```

```
SQLCA.AutoCommit = False
```

```
SQLCA.DBParm = "ConnectionString='DSN=ODBC_PB'"
```

```
connect;
```

```
string sno,sname,ssex,sage,saddr,home
```

```
lb_list.reset()
```

```
lb_list.additem(" 学号    姓名  性别  年龄  地址")
```

```
home=trim(sle_addr.text)
```

```
declare cs cursor for
```

```
select sno,sname,ssex,sage,saddr
```

```
from s
```

```
where saddr=:home;
```

```
open cs;
```

```
if(sqlca.sqlcode=-1)then//判断是否连接成功
```

```
    MessageBox("error","error")
```

```
Else//连接成功就插入数据
```

```
    fetch cs into :sno,:sname,:ssex,:sage,:saddr;
```

```
    do while sqlca.sqlcode=0
```

```
        lb_list.additem(sno+" "+sname+" "+ssex+" "+sage+" "
```

```
        "+saddr)
```

```
        fetch cs into :sno,:sname,:sage,:saddr;
```

```
    loop
```

```
end if
```

```
close cs;
```

退出按钮:

HALT

程序 open 函数

```
open(w_search)
```

数据库:

表 - dbo. DEPT		表 - dbo. S	摘要
	DNO	DEPARTMENT	
▶	1	计算机	
*	NULL	NULL	

表 - dbo.S 摘要						
	sno	sname	ssex	sage	saddr	dno
▶	150120526	殷悦	男	22	威海	1
*	NULL	NULL	NULL	NULL	NULL	NULL