

## 实验一：SQL 定义功能、数据插入

### 1. 建立教学数据库的三个基本表：

S(Sno,Sname,Sgender,Sage,Sdept) 学生（学号，姓名，性别，年龄，系）

SC(Sno,Cno,Grade) 选课（学号，课程号，成绩）

C(Cno,Cname,Cpno,Ccredit) 课程（课程号，课程名，先行课，学分）

```
create table S
(Sno char(20) PRIMARY KEY,
Sname char(50),
Sgender char(10),
Sage smallint,
Sdept char(20))
/
```

```
SQL> start f1
```

```
Table created.
```

```
SQL> desc S
```

Name	Null?	Type
SNO		CHAR(20)
SNAME		CHAR(50)
SGENDER		CHAR(10)
SAGE		NUMBER(38)
SDEPT		CHAR(20)

```
create table C
(Cno char(4) PRIMARY KEY,
Cname char(40) not null,
Cpno char(4),
Ccredit smallint,
FOREIGN KEY (Cpno) REFERENCES C(Cno))
/
```

```
SQL> create table C
2 (Cno char(4) PRIMARY KEY,
3 Cname char(40) not null,
4 Cpno char(4),
5 Ccredit smallint,
6 FOREIGN KEY (Cpno) REFERENCES C(Cno));

Table created.
```

```
SQL> desc C
```

Name	Null?	Type
CNO	NOT NULL	CHAR(4)
CNAME	NOT NULL	CHAR(40)
CPNO		CHAR(4)
CCREDIT		NUMBER(38)

```
create table SC
(Sno char(20),
Cno char(50),
Grade smallint,
PRIMARY KEY (Sno,Cno),
FOREIGN KEY (Sno) REFERENCES S(Sno),
FOREIGN KEY (Cno) REFERENCES C(Cno))
/
```

```
SQL> start f2
```

Table created.

```
SQL> desc SC
```

Name	Null?	Type
SNO	NOT NULL	CHAR(20)
CNO	NOT NULL	CHAR(50)
GRADE		NUMBER(38)

2. DROP TABLE、ALTER TABLE、CREATE INDEX、DROP INDEX 及 INSERT 语句输入数据。

(1) create index on

```
SQL> create unique index Cnameindex on C(Cname);
```

索引已创建。

(2) drop index:

```
SQL> drop index Cnameindex;
```

索引已删除。

转移到自己的电脑:

不小心发现 Sdept 打成了 Sbept,于是修改列名 alter table

```
SQL> alter table S rename column Sdept to Sdept;
表已更改。
SQL> desc S
名称 是否为空? 类型
-----
SNO NOT NULL CHAR(20)
SNAME CHAR(50)
SGENDER CHAR(10)
SAGE NUMBER(38)
SDEPT CHAR(20)
```

(3) INSERT:

insert into S(Sno,Sname,Sgender,Sage,Sdept) values ('161530301','应亦周', '女', 20, 'CS')  
/

```
SQL> edit f1
SQL> start f1
已创建 1 行。
```

继续插入数据:

insert into S(Sno,Sname,Sgender,Sage,Sdept) values ('161530302','刘杰', '女', 19, 'CS')  
/

insert into S(Sno,Sname,Sgender,Sage,Sdept) values ('101530303','王星  
宇', '男', 18, 'MA')  
/

insert into S(Sno,Sname,Sgender,Sage,Sdept) values ('111530303','庄明帅', '男', 19, 'IS')  
/

```
SQL> select * from S;
```

SNO	SNAME	SGENDER	SAGE	SDEPT
161530301	应亦周	女	20	CS
161530302	刘杰	女	19	CS
101530303	王星宇	男	18	MA
111530303	庄明帅	男	19	IS

通过 **DROP** 删除一行记录:

```
SQL> delete from S where Sno='161530301';
```

已删除 1 行。

用 **DELETE** TABLE 可以删除表中所有数据，但是不会删除表结构:

```
SQL> delete S;
```

已删除3行。

然后通过以下语句一次性插入所有记录:

```
insert into S(Sno,Sname,Sgender,Sage,Sdept)
(select '20125121','李勇','男',20,'CS' from dual)
union all(select '20125122','刘晨','女',19,'CS' from dual)
union all(select '20125123','王敏','女',18,'MS' from dual)
union all(select '20125125','张立','男',19,'IS' from dual)
/
```

```
SQL> start f1
```

已创建4行。

改成了书上的数据:

```
SQL> select * from S;
```

SNO	SNAME
20125121	李勇
男	20 CS
20125122	刘晨
女	19 CS
20121523	王敏
女	18 MS

SNO	SNAME
20125125	张立
男	19 IS

同样插入数据到其他的表:

```
SQL> start f1
```

已创建7行。

```
insert into C(Cno,Cname,Cpno,Ccredit)
(select '1','数据库','5',4 from dual)
union all(select '2','数学','',2 from dual)
union all(select '3','信息系统','1',4 from dual)
union all(select '4','操作系统','6',3 from dual)
union all(select '5','数据结构','7',4 from dual)
union all(select '6','数据处理','',2 from dual)
union all(select '7','PASCAL 语言','6',4 from dual)
/
```

```
SQL> start f1
insert into SC(Sno,Cno,Grade) values (<'20121521','1',92>
*
第 1 行出现错误:
ORA-02291: 违反完整约束条件 (SYSTEM.SYS_C0011107) - 未找到父项关键字

SQL> select * from S;
```

SNO	SNAME
20125121	李勇
男	20 CS

出现了错误，用 **ALTER** 修改字段值：

```
SQL> update S set sno='20121521' where sno='20125121';  
已更新 1 行。
```

```
insert into SC(Sno,Cno,Grade)  
(select '20121521','1',92 from dual)  
union all(select '20121521','2',85 from dual)  
union all(select '20121521','3',88 from dual)  
union all(select '20121522','2',90 from dual)  
union all(select '20121522','3',80 from dual)  
/
```

## 实验二：数据查询

1. 查询选修 1 号课程的学生学号与姓名。

```
select distinct S.Sno,Sname from S,SC  
where S.Sno in (select Sno from SC where S.Sno=SC.Sno and SC.Cno='1')  
/
```

```
SQL> start f1
```

SNO	SNAME
20121521	李勇

2. 查询选修课程名为数据结构的学生学号与姓名。

```
select Sno,Sname from S  
where Sno in  
(select Sno from SC  
where Cno =  
(select Cno from C where Cname='数据结构'))  
/
```

```
SQL> start f1
```

未选定行

```
SQL> insert into SC (Sno,Cno,Grade) values ('20121525','5',91);  
已创建 1 行。
```

SNO	SNAME
20121525	张立

3. 查询不选 1 号课程的学生学号与姓名。

```
select Sno,Sname from S
where Sno in
(select Sno from SC
where Cno not in
(select Cno from C where Cno='1'))
/
```



A screenshot of a SQL query result in a terminal window. The prompt is 'SQL> start f1'. The result is a table with two columns: 'SNO' and 'SNAME'. The data rows are: 20121522 (刘晨), 20121521 (李勇), and 20121525 (张立).

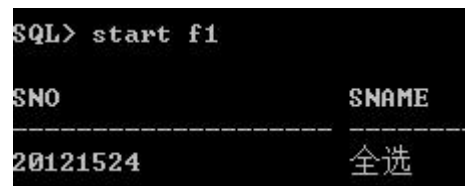
SNO	SNAME
20121522	刘晨
20121521	李勇
20121525	张立

4. 查询学习全部课程学生姓名。

```
insert into S (Sno,Sname,Sgender,Sage,Sdept) values ('20121524','全选','男',23,'CS')
/
```

```
insert into SC(Sno,Cno,Grade)
(select '20121524','1',91 from dual)
union all(select '20121524','2',92 from dual)
union all(select '20121524','3',93 from dual)
union all(select '20121524','4',94 from dual)
union all(select '20121524','5',95 from dual)
union all(select '20121524','6',96 from dual)
union all(select '20121524','7',97 from dual)
/
```

```
select Sno,sname from S
where not exists
(select * from C
where not exists
(select * from SC
where Sno=S.Sno
and Cno=C.Cno))
/
```



A screenshot of a SQL query result in a terminal window. The prompt is 'SQL> start f1'. The result is a table with two columns: 'SNO' and 'SNAME'. The data row is: 20121524 (全选).

SNO	SNAME
20121524	全选

5. 查询所有学生除了选修 1 号课程外所有成绩均及格的学生的学号和平均成绩，其结果按平均成绩的降序排列。

```

select Sno,AVG(Grade) from SC
where Cno<>'1' and Grade>=60
group by Sno
order by AVG(Grade) desc
/

```

```
SQL> start f1
```

SNO	AVG<GRADE>
20121524	94.5
20121525	91
20121521	86.5
20121522	85

6. 查询选修数据库原理成绩第 2 名的学生姓名。

```

select Sname from S where Sno in
(select Sno from SC where Grade in
(select MAX(Grade) from
(
select * from SC where Grade not in
((select MAX(Grade) from
(select * from SC where Cno in
(select Cno from C where Cname='数据结构')) t)) and Cno in
(select Cno from C where Cname='数据结构')
) t2) and Cno in (select Cno from C where Cname='数据结构'))
/

```

```
SQL> start f1
```

SNAME
张立

7. 查询所有 3 个学分课程中有 3 门以上（含 3 门）课程获 80 分以上（含 80 分）的学生的姓名。

```

select Sname from S where Sno in
(select Sno from SC
where Cno in
(select Cno from C where Ccredit>=3)
and Grade>=80
group by Sno
having COUNT(*)>=3)
/

```



```

SQL> insert into SC values (&Sno','&Cno',&Grade);
输入 sno 的值: 20121521
输入 cno 的值: 4
输入 grade 的值: 80
原值      1: insert into SC values (&Sno','&Cno',&Grade)
新值      1: insert into SC values (&Sno','&Cno',&Grade)

已创建 1 行。

SQL> commit
2 ;

提交完成。

SQL> start f1

SNAME
-----
本勇
全选

```

#### 8. 查询选课门数唯一的学生的学号。

```

select Sno from SC
group by Sno
having count(*)=1
/

```

```

SNO
-----
20121525

```

#### 9. SELECT 语句中各种查询条件的实验。

(1) 查询所有课程的平均（以 avggrade 命名）分以及对应的课程名

```

select C.Cname,t.avggrade from C,
(select Cno,avg(Grade) avggrade from SC
group by SC.Cno) t
where C.Cno=t.Cno
/

```

CNAME	AUGGRADE
数据库	91.5
数学	89
信息系统	87
操作系统	87
数据结构	93
数据处理	96
PASCAL语言	97

已选择7行。

### 实验三：数据修改、删除

1. 把 1 号课程的非空成绩提高 10%。

之前：

SNO	GRADE	CNO
20121521	92	1
20121524	91	1
20121526		1

update SC

set Grade=Grade\*1.1

where Cno='1'

/

之后：

```
SQL> start f1
```

已更新3行。

```
SQL> select * from SC where Cno='1';
```

SNO	GRADE	CNO
20121521	101	1
20121524	100	1
20121526		1

Rollback 撤销。

2. 在 SC 表中删除课程名为数据结构的成绩的元组。

delete

from SC

where Cno in

(select Cno from C

where Cname='数据结构')

/

```
SQL> start f1

已删除2行。

SQL> select * from SC where Cno='5';

未选定行
```

显示已经全部被删除。

### 3. 在 S 和 SC 表中删除学号为 20121522 的所有数据。

经查阅，oracle 不允许一条语句删除多张表的部分数据。所以采用两条语句。

```
delete
from SC
where Sno='20121522'
/
delete
from S
where Sno='20121522'
/
```

## 实验四：视图的操作

### 1. 建立男学生的视图，属性包括学号、姓名、选修课程名和成绩。

```
create view ls_boy
as
select S.Sno,Sname,Cname,Grade
from S,SC,C
where S.Sno=SC.Sno and SC.Cno=C.Cno
/
```

```
视图已创建。

SQL> select view_name from user_views;

VIEW_NAME
-----
AQ$DEF$_AQCALL
AQ$DEF$_AQERROR
AQ$_DEF$_AQCALL_F
AQ$_DEF$_AQERROR_F
IS_BOY
MVIEW_EVALUATIONS
MVIEW_EXCEPTIONS
MVIEW_FILTER
MVIEW_FILTERINSTANCE
MVIEW_LOG
MVIEW_RECOMMENDATIONS
```

### 2. 在男学生视图中查询平均成绩大于 80 分的学生学号与姓名。

```
select Sno,Sname,avg(grade) from Is_boy
group by Sno,Sname
having avg(grade)>80
/
```

SNO	SNAME
-----	
AUG<GRADE>	
-----	
20121521	李勇
86.25	
20121524	全选
94	
20121525	张立
91	
SNO	SNAME
-----	-----
AUG<GRADE>	
-----	
20121522	刘晨
85	

### 实验五：库函数，授权控制

1. 计算每个学生有成绩的课程门数、平均成绩。

```
select t.sno,t.num,t.avgg from
(select Sno,count(Cno) num,avg(Grade) avgg from SC
group by Sno) t
/
```

SNO	NUM	AUGG
-----	-----	-----
20121521	4	86.25
20121525	1	91
20121526	1	
20121522	2	85
20121524	7	94

2. 使用 **GRANT** 语句，把对基本表 S、SC、C 的使用权限授给其它用户。

```
SQL> conn system/system;
已连接。
SQL> grant S,SC,C to "myuser";
```

3. 实验完成后，撤消建立的基本表和视图。

```
SQL> drop view Is_boy;
```

视图已删除。

### 实验六：综合实验：实现一个小型管理信息系统 10 学时

#### 问题描述：

熟练掌握 Visual C++、Pro\*C 或 Java 访问数据库的方法，设计和实现学生通讯录或学生选课的一个小型管理信息系统。要求具有数据的增加、删除、修改和查询的基本功能，并尽可能提供较多的查询功能，用户界面要友好。课程结束前提交实验报告和程序。