

上机实验三——基本表的建立和修改

三、实习内容：

1. 启动 MSSQL Server 服务，打开 Enterprise Manager 和 Query Analyzer。
2. 在 Query Analyzer 中用 CREATE TABLE 命令在实验二创建的 GradeManager 数据库中定义基本表：学生表（Student）、课程表（Course），利用 Enterprise Manager 的图形化功能建立班级表（Class）以及成绩表（Grade）。

create table Student

(Sno char(7) primary key ,

Sname varchar(20) not null ,

Ssex char(2) not null,

Sage Smallint ,

Clno char(5)not null);

作业点评：有同学出现了这样的 SQL 语句：

create table student

(sno char(7)primary key not null,

.....

说明：定义了 Primary key 以后，就不用定义为 Not Null 了，并且一般的次序应该是 not null primary key

create table Class

(Clno char(5) primary key,

Speciality varchar(20) not null,

Inyear char(4) not null,

Number integer ,

Monitor char(7));

create table Course

(Cno char(1) primary key,

Cname varchar(20) not null,

Credit Smallint);

```
create table Grade
(Sno char(7) references student(sno),
Cno char(1) references course(cno),
Gmark numeric,
Primary key(sno,cno));
```

四、针对以上四个表，用 SQL 语言完成以下各项操作。

- ① 给学生表增加一属性 Nation（民族），数据类型为 Varchar（20）；
- ② 删除学生表中新增的属性 Nation；
- ③ 向成绩表中插入记录（"2001110"，"3"，80）；
- ④ 修改学号为"2001110"的学生的成绩为 70 分；
- ⑤ 删除学号为"2001110"的学生的成绩记录；
- ⑥ 为学生表创建一个名为 IX_Class 的索引。

1. ALTER TABLE Student

```
ADD Nation varchar(20);
```

2. ALTER TABLE Student

```
DROP COLUMN Nation;
```

3. Insert into grade(sno,cno,gmark) values('2001110', '3', 80)

点评：运行该语句后，系统会出现该语句违反了参照完整性的错误信息，语句没有执行成功，原来是在我们定义 Grade 表时，定义了 Grade 的 Sno 属性要参照 Student 表的 Sno，而此时想插入的学号 2001110 是在 Student 表不存在的。大家要知道错误的原因。

4. UPDATE Grade

```
SET Gmark=70
```

```
WHERE Sno='2001110';
```

5. DELETE

```
FROM Grade
```

```
WHERE Sno='2001110';
```

6. CREATE INDEX IX_Class

```
ON Student(Cno Asc);
```

7. DROP INDEX Student.IX_Class;

五、思考题

在定义基本表语句时，NOT NULL 参数的使用有何作用？

答：Not Null 参数，可以保证在插入数据时，该属性列的取值不会空。

作业点评：有同学如此回答：SQL 可以很好的支持大数据量的存取，NOT NULL 参数的使用可以提高查询效率

这个回答有点偏题了，呵呵。

上机实验四——SELECT 语句的使用(一)

三、实习内容：

完成以下各项操作的 SQL 语句：

① 找出所有被学生选修了的课程号；

```
select distinct cno from grade;
```

该语句实现的路径应该是 Grade 表，Course 表中可能有某门课，一个学生都没有选修的。所以不能是查询 Course 表的 Cno。

② 找出 01311 班女学生的个人信息；

```
select * from student
```

```
where Ssex = '女' and clno = '01311';
```

③ 找出 01311 班、01312 班的学生姓名、性别、出生年份；

```
select Sname,Ssex,2009- Sage as birth
```

```
from student
```

```
where clno = '01311' or clno = '01312';
```

OR:

```
select Sname,Ssex,2009- Sage as birth
```

```
from student
```

```
where clno in('01311', '01312');
```

OR: (在 T-SQL 中,超大纲了,呵呵)

```
Select sname, sex, year(getdate())-sage
```

```
From student  
Where clno in ('01311','01312');
```

④ 找出所有姓李的学生的个人信息;

```
select * from student  
where Sname like '李%';
```

⑤ 找出学生李勇所在班级的学生人数;

```
Select count(*) from student  
Where clno in (Select clno from student  
Where sname="李勇");
```

Or: Select number from class

```
Where clno in (Select clno from student  
Where sname="李勇");
```

Or: Select number from class, student

```
Where sname='李勇' and class.clno=student.clno;
```

⑥ 找出课程名为操作系统的平均成绩、最高分、最低分;

```
Select avg(gmark), Max(gmark), Min(gmark)  
From grade, course  
Where cname="操作系统" and grade.cno=course.cno;
```

Or: Select avg(gmark), Max(gmark), Min(gmark)

```
From grade
```

```
Where cno=
```

```
(Select cno from course where cname="操作系统");
```

⑦ 找出选修了课程的学生人数;

```
select count(distinct Sno)
```

```
from grade;
```

⑧ 找出选修了课程操作系统的学生人数。

```
Select count(sno) From grade
```

Where cno=

(Select cno from course Where cname="操作系统");

Or: Select count(sno) From grade, course

Where grade.cno=course.cno and cname="操作系统";

四、思考题:

什么情况下需要使用关系的别名？别名的作用范围是什么？

答：一般几种两种情况下使用别名：1. 关系名太长，为了简化；2. 为了做自身连接查询。3. 在同一个 SQL 语句中，为了对某个表进行 2 次扫描。

别名的作用范围是当前该 SQL 语句，离开当前 SQL 语句，别名就不存在。

上机实验五——SELECT 语句的使用(二)

三、实习内容:

完成以下各项操作的 SQL 语句：

- ① 找出与李勇在同一个班级的学生信息；
- ② 找出选修了课程操作系统的学生学号和姓名；
- ③ 找出年龄介于学生李勇和 25 岁之间的学生信息；（已知李勇年龄小于 25 岁）
- ④ 找出所有没有选修 1 号课程的学生姓名
- ⑤ 查询选修了 3 号课程的学生学号及其成绩，并按成绩的降序排列；
- ⑥ 求每个课程号及相应的选课人数；
- ⑦ 查询选修了 3 门以上课程的学生学号。

- ① 找出与李勇在同一个班级的学生信息；

Select * from student

Where clno=

(Select clno from student where sname='李勇');

- ② 找出选修了课程操作系统的学生学号和姓名；

Select sno, sname from student

Where sno in

(Select sno from grade

```
Where cno =  
        ( Select cno from course  
          Where cname ='操作系统'));
```

- ③ 找出年龄介于学生李勇和 25 岁之间的学生信息;

```
Select * from student  
  
Where sage between  
        (Select sage from student  
          Where sname='李勇') And 25;
```

- ④ 找出所有没有选修 1 号课程的学生姓名;

```
Select sname from student  
  
Where not exist  
        (Select sno from grade  
          where cno='1' and sno=student.sno);  
  
OR: Select sname from student  
      Where sno not in  
            (Select sno from grade  
              Where cno='1');
```

- ⑤ 查询选修了 3 号课程的学生学号及其成绩，并按成绩的降序排列;

```
Select sno,gmark from grade  
  
Where cno='3'  
  
Order by gmark desc;
```

- ⑥ 求每个课程号及相应的选课人数;

```
Select cno,count(sno)  
  
From grade  
  
Group by cno;
```

- ⑦ 查询选修了 3 门以上课程的学生学号。

```
Select sno from grade  
Group by sno Having count(*)>3;
```

⑧ 找出选修了全部课程的学生姓名。

(提示: 可找出这样的学生, 没有一门课程是他不选修的。)

解一:

```
Select sname from student  
Where not exists  
    (Select * from course  
     where not exists  
         (Select * from grade  
          Where sno=student.sno and cno=course.cno));
```

解二:

```
Select sname from student  
Where sno in  
    (Select sno from grade  
     Group by sno  
     Having count(*)=Select count(*) from course);
```

```
OR:  Select sname from student  
      Where (Select count(cno) from grade      Group by sno )  
            =(Select count(*) from course);
```

类似表达的还有:

--1.

```
SELECT Sname FROM Student  
WHERE (select count(cno) from course)  
      =(SELECT count(sno) FROM Grade  
        GROUP BY Sno)
```

--2.

```
SELECT Sname FROM Student  
WHERE Sno=ANY
```

```
(SELECT Sno FROM Grade
GROUP BY Sno
HAVING(COUNT(*)=
(SELECT COUNT(*) FROM Course)))
ORDER BY Sname;
```

五、思考题：

1. 用 UNION 或 UNION ALL 将两个 SELECT 命令结合为一个时，结果有何不同？
2. 当既能用连接词查询又能用嵌套查询时，应该选择哪种查询较好？为什么？
3. 库函数能否直接使用在：SELECT 选取目标、HAVING 子句、WHERE 子句、GROUP BY 列名中？

上机实验六——SQL 的存储操作

三、实习内容：

完成以下各项操作的 SQL 语句：

②删除 01311 班全体学生的选课记录；

③学生李勇已退学，从数据库中删除有关他的记录；

①对每个班，求学生的平均年龄，并把结果存入数据库。

①将 01311 班的全体学生的成绩置零；

```
update grade
```

```
set gmark=0
```

```
where sno in (select sno from student where clno='01311')
```

②删除 01311 班全体学生的选课记录；

```
delete
```

```
from grade
```

```
where sno in (select sno
```

```
from student
```

```
where clno = '01311' ;
```

③ 学生李勇已退学，从数据库中删除有关他的记录。

注意：数据的一致性 删除表的次序：Grade / class /student


```
delete from grade  
where sno in  
    (select sno from student  
     where sname ='李勇');
```

```
update class  
set number = number -1  
where clno =  
    (select clno from student  
     where sname ='李勇');
```

```
delete from student  
where sname ='李勇';
```

作业点评：有同学的语句如下：

```
delete from student a join Grade b on a.Sno=b.Sno join Course c on c.Cno=b.Cno  
where Sname='李勇';
```

注意：微软的定义是一个 Delete 语句只能对一个表进行删除操作！

④对每个班，求学生的平均年龄，并把结果存入数据库；

解法一：建表、插入

```
create table avgage  
(clno char(5),  
 avgage smallint);  
insert into avgage select clno,avg(sage) from student group by clno;
```

解法二：建视图

```
create view v_avgage  
as  
select clno , avg(sage) as avg_age  
from student
```

解法三：修改 Class 表后插入

```

alter table class add avgage smallint
go
update class
set avgage = (select age(sage) from student
              group by clno
              having student.clno = class.clno)

```

解法四：直接建表插入

```

select clno, avg(sage) as avg_age into avgage
from student

```

五、思考题：

DROP 命令和 DELETE 命令的本质区别是什么？

19. 视图操作

```

create view Stu_00312_1          --1
as select *
from Student
where Student.Sno in(
    select Sno
    from Grade
    where Grade.Cno='1')and Student.Clno='00312'

create view Stu_00312_2          --2
as select *
from Student
where Student.Sno in(
    select Sno
    from Grade
    where Grade.Cno='1'and Grade.Gmark<=60)and Student.Clno='00312'

```

```

create view Stu_year          --3
as select Sno ,Sname,2005-Sage birthyear
from Student
select Sname          --4
from Stu_year
where birthyear>1983
select Sno,Sname,2005-Sage  --5
from Stu_01312_2

```

20.完整性约束说明:

```

Create table student
(sno char(7) primary key,
sname varchar(20) not null,
ssex char(2) not null check(ssex in('男','女')) default ('男'),
sage smallint check(sage<65 and sage>14),
clno char(5) not null)

alter table course add primary key(cno);
alter table course add constraint PK_course primary key(cno);
alter table course add constraint chk_cred
    check(credit in (1,2,3,4,5,6));
-----

alter table class add primary key(clno);
alter table class add check(number>1 and number<100);
alter table class add foreign key (monitor) references    student(sno);
-----

alter table student add foreign key(clno) references class(clno);
alter table grade add primary key(sno,cno);
alter table grade add foreign key (cno) references course(cno);
alter table grade add constraint chk_mark check(gmark>=0 and Gmark<=100);

```