

数据库系统实验——SQL语言

王肇康 (091220108)

2011年10月04日

1 数据建立

1.1 建立基本表

使用的SQL代码如下

```
CREATE TABLE Student( SNO INT PRIMARY KEY, SNAME CHAR(8) NOT NULL, SEX  
CHAR(2), DEPTNO INT );  
  
CREATE TABLE Course( CNO INT, CNAME CHAR(20) NOT NULL, TNO INT, CREDIT  
INT, PRIMARY KEY(CNO,TNO) );  
  
CREATE TABLE SC ( SNO INT, CNO INT, PRIMARY KEY(SNO,CNO), GRADE INT );  
  
CREATE TABLE Teacher( TNO INT PRIMARY KEY, TNAME CHAR(8) NOT NULL, DEPTNO  
INT );  
  
CREATE TABLE Dept( DEPTNO INT PRIMARY KEY, DNAME CHAR(20) NOT NULL );
```

建立后的效果如图1所示。

1.2 修改基本表

使用的SQL代码如下:

```
ALTER TABLE Student ADD AGE INT;
```

```
ALTER TABLE Student MODIFY AGE SMALLINT;
```

效果如图所示。

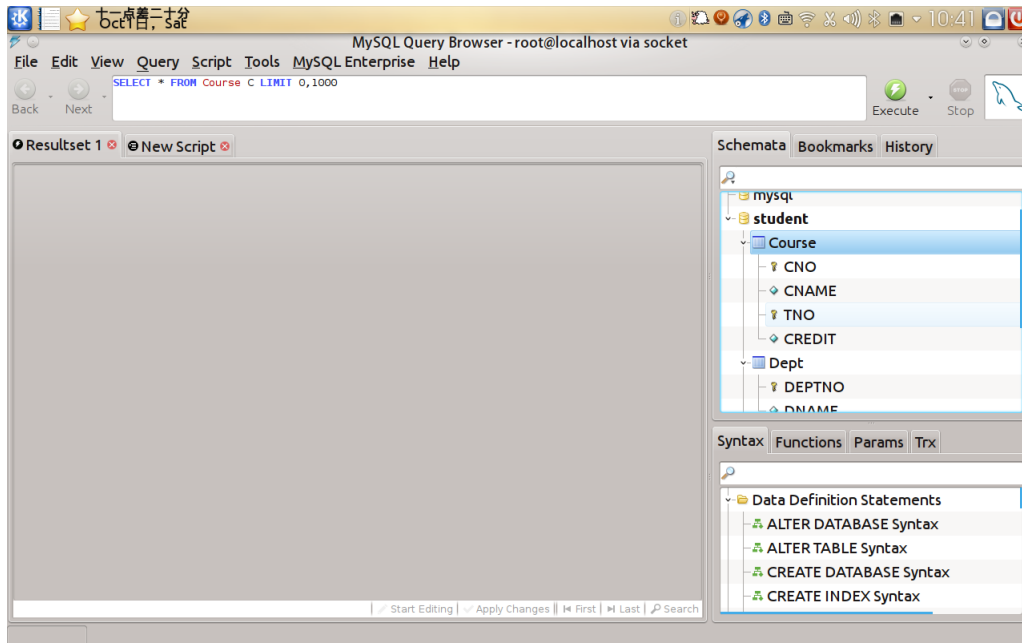


图 1：建立基本表

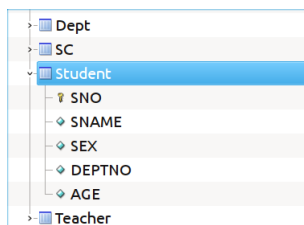


图 2：修改表

1.3 插入数据

使用的代码如下：

```
Insert Into Student
values(1001,'张天','m',10,20),(1002,'李兰','f',10,21),
(1003,'陈铭','m',10,21),(1004,'刘茜','f',20,21),
(1005,'马阳','m',20,22);
Insert Into Course
values(1,'数据结构',101,4),(2,'数据库',102,4),
(3,'离散数学',103,4),(4,'C语言程序设计',101,2);
Insert Into SC
values(1001,1,80),(1001,2,85),(1001,3,78),
(1002,1,78),(1002,2,82),(1002,3,86),
(1003,1,92),(1003,3,90),
(1004,1,87),(1004,4,90),
(1005,1,85),(1005,4,92);
Insert Into Teacher
values
(101,'张星',10),(102,'李珊',10),
(103,'赵应',10),(104,'刘田',20);
Insert Into Dept
values
(10,'计算机'),(20,'信息');
```

2 查询

2.1 单表查询

使用的SQL语句如下，一行对应一个问题：

```
select * from Student;
```

```
select SNAME from Student where SEX = 'f';
```

```
select * from SC where GRADE between 80 and 89 order by GRADE DESC;
```

```
SELECT DEPTNO,COUNT(*) FROM Student GROUP BY DEPTNO;
```

2.2 连接查询

使用SQL语句如下：

```
--连接查询
select Student.SNAME,Student.SNO
from Student,Dept
where Dept.DEPTNO = Student.DEPTNO

      AND Student.AGE <= 21
      AND Student.SEX = 'f'
      AND Dept.DNAME = '信息';
```

这里没有使用SQL语言自身提供的JOIN功能，而是在WHERE中显示指出了连接条件。

其结果如图所示。

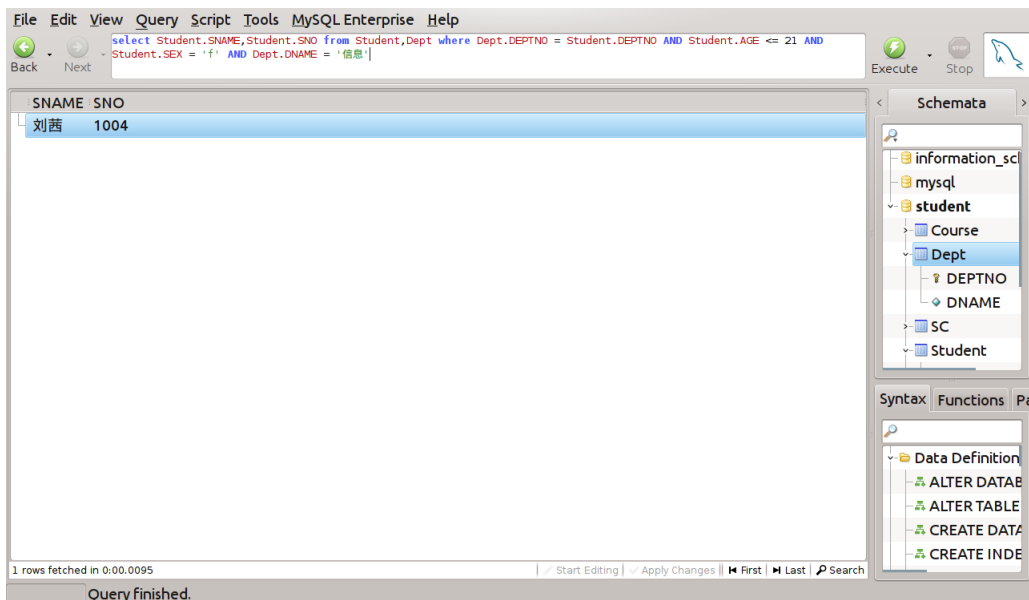


图 3：连接查询结果

2.3 嵌套查询

使用的SQL语句如下所示：

```
--嵌套查询
--1
SELECT Student.SNAME FROM Student
    WHERE Student.SNO IN
    ( SELECT SC.SNO FROM SC,Course
    WHERE SC.CNO = Course.CNO
    GROUP BY SC.SNO
    HAVING SUM(Course.CREDIT) < 10);
--2
SELECT Student.SNAME,HIGH_LIST.CNO,HIGH_LIST.GRADE
FROM Student NATURAL JOIN
    (SELECT SC.SNO,SC.CNO,SC.GRADE
    FROM SC NATURAL JOIN
        (SELECT SC.CNO,MAX(SC.GRADE) AS GRADE
        FROM SC GROUP BY SC.CNO) AS HIGHEST )
    AS HIGH_LIST
ORDER BY HIGH_LIST.CNO
```

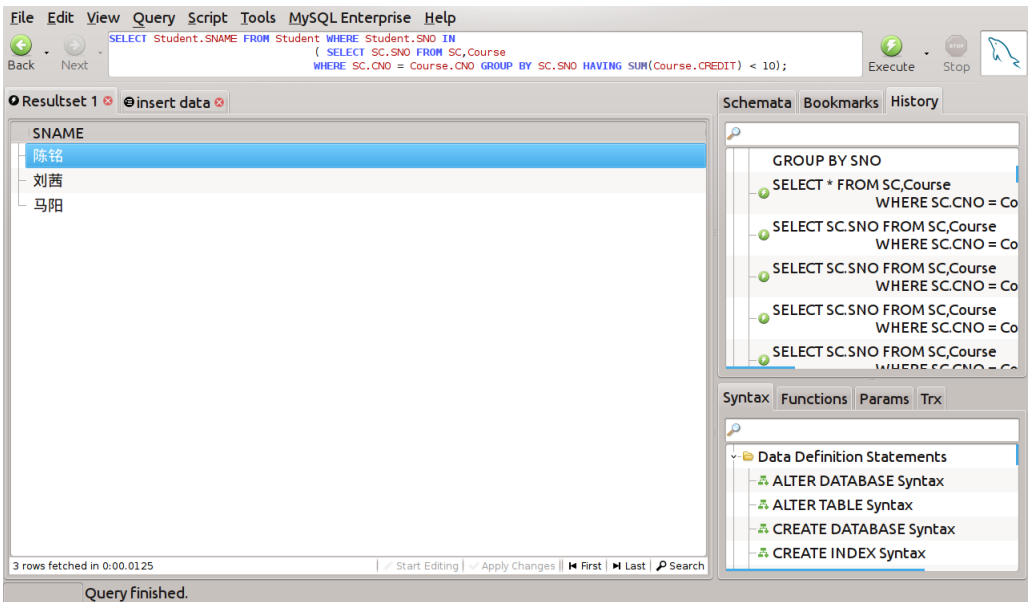
```
--3
SELECT A.SNO FROM SC A NATURAL JOIN (
    SELECT B.CNO FROM SC B
    WHERE B.SNO = 1001) AS C
GROUP BY A.SNO
HAVING COUNT(A.CNO) >=
    (SELECT COUNT(*) FROM
        (SELECT B.CNO FROM SC B
        WHERE B.SNO = 1001) AS D);

--4
SELECT Student.SNAME
FROM Student
WHERE Student.SNO IN (
    SELECT SC.SNO FROM SC NATURAL JOIN (
        SELECT Course.CNO
        FROM Course
        WHERE Course.TNO IN (
            SELECT TNO
            FROM Teacher
            WHERE TNAME = '张星')
        ) AS ZX_COURSE
    GROUP BY SC.SNO
    HAVING COUNT(*) >= (
        SELECT COUNT(*)
        FROM Course
        WHERE Course.TNO IN (
            SELECT TNO
            FROM Teacher
            WHERE TNAME = '张星')
        )
    )
)
```

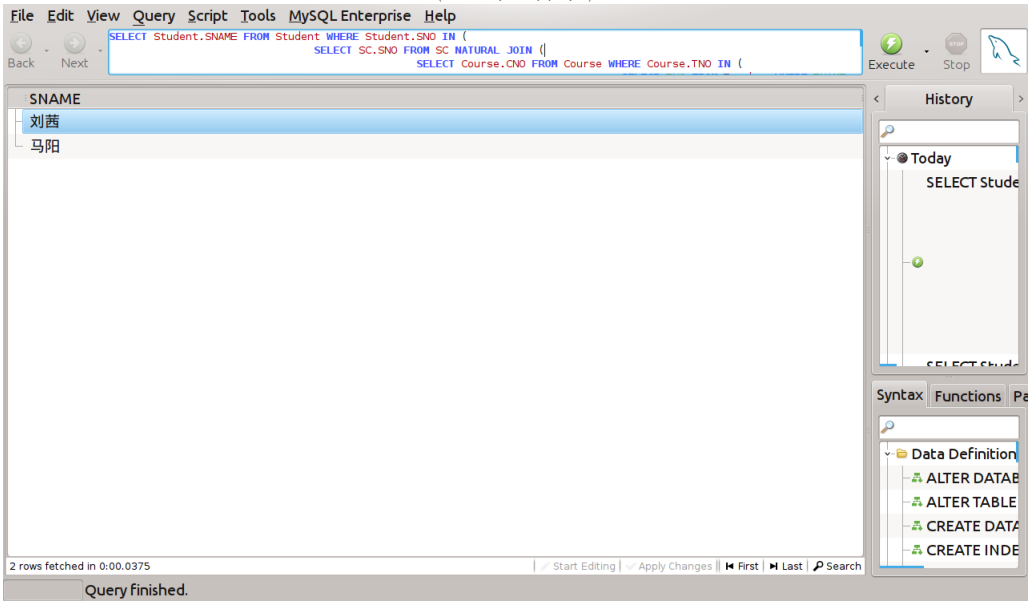
其中第一、四个查询的查询结果如图所示。

2.4 数据修改

数据修改使用的SQL语句如下：



(查询1结果)



查询4结果

图 4： 嵌套查询结果

```
--修改张星老师的数据结构课程成绩
UPDATE SC
SET GRADE = GRADE + 2 WHERE SC.CNO IN
(
    SELECT Course.CNO
    FROM Course
    WHERE Course.CNAME = '数据结构'
    AND Course.TNO IN
        (SELECT Teacher.TNO
        FROM Teacher
        WHERE TNAME = '张星'
        )
);
```

2.5 数据删除

使用语句如下:

```
--delete:
DELETE FROM SC
WHERE SC.SNO IN (
    SELECT Student.SNO
    FROM Student
    WHERE Student.SNAME = '马阳' );
```

3 视图

3.1 建立

使用语句如下:


```
--view:  
CREATE VIEW CS_STUDENT  
AS SELECT *  
FROM Student  
WHERE Student.DEPTNO IN (  
SELECT DEPTNO  
FROM Dept  
WHERE DNAME = '计算机');
```

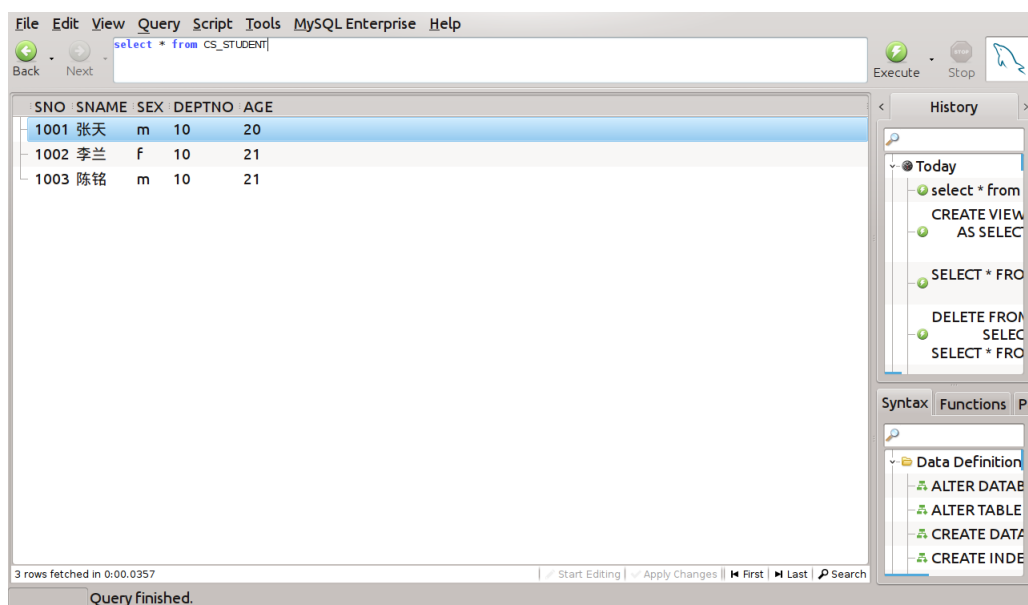


图 5：建立新视图

3.2 视图删除

```
DROP VIEW CS_STUDENT;
```

4 表删除

```
DROP TABLE Student,Course,SC,Teacher,Dept;
```

5 遇到的问题和发现

1. 在处理字符串的时候，最初传入的中文均变成??，后来发现是MySQL (5.1.x) 处理字符编码时默认是西文字符，所以不能识别中文。在使用可视化工具调整为UTF-8之后，就可以正常显示了。

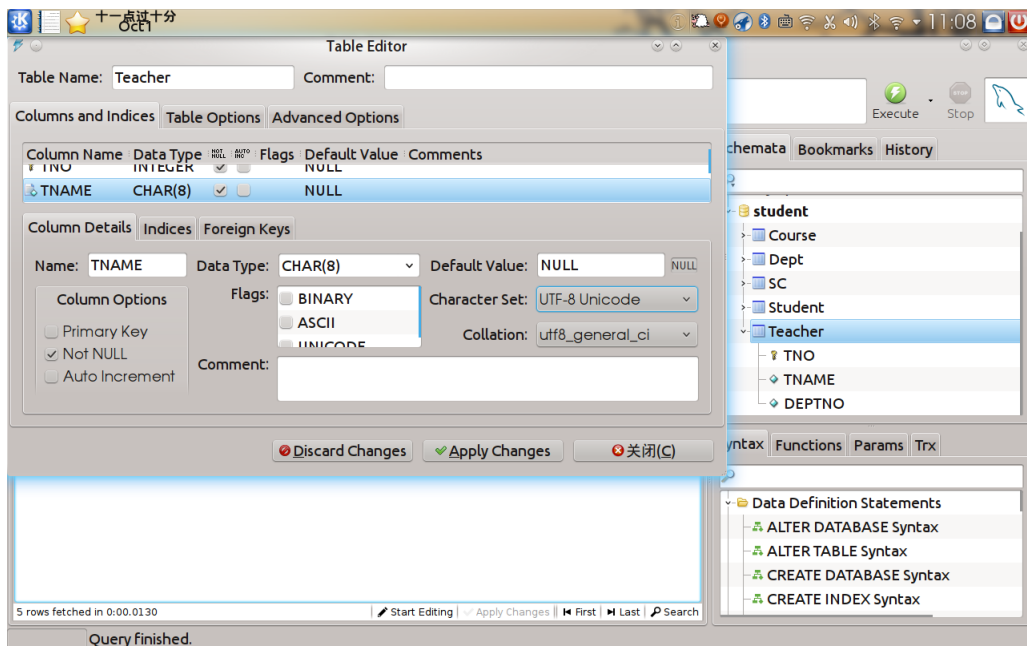


图 6：转换字符的编码

1. 在看讲义和查找资料时，发现SQL-92标准对于连结操作已经在FROM子句中提供了INNER JOIN等操作符。在TSQL中，推荐使用这些连结操作连结操作符以在WHERE子句中将连结操作条件和逻辑判断条件区别开，使SQL语句更容易阅读。
2. 在查询过程中可以发现，当对某个表执行第一次查询时，查询需要的时间较多，而此后在同一个表上进行类似的查询时，所需时间大大减少。这说明了数据库系统对一些最近查过的表进行了缓存，使其操作速度提升。
3. 有关关系代数中的除法在SQL语句中的实现：SQL语句不直接支持关系代数中的除运算，因此需要其他的方式来实现。在网上搜集资料后，发现有三种实现方式：
 - (a) 使用“相关子查询”：使用两层嵌套的带有NOT EXISTS条件的嵌套子查询实现。这种方式一直不是很理其原理；
 - (b) 使用COUNT的方式在连结后判断是否满足除法的条件，在本次实验中采用的即是此种方式；
 - (c) 将除关系的关系代数转变为笛卡儿乘和减法操作(1)，在本次实验中因为EXCEPTION操作一直没有通过而放弃了。

设 $T=R \div S$ ，设 R' 为 R 中不包涵于 S 的属性

$$T = \prod_{R'} R - \prod_{R'} (\prod_{R'} R \times S - R) \tag{1}$$