数据库实验一

赵丰

091242045

实验环境： MySQL Server 5.5

实验步骤：

（一）数据定义

1. 建立基本表

[1] 创建学生表 Student ,由以下属性组成 :学号 SNO(INT 型 ,主键),姓名 SNAME(CHAR 型，长度为 8，非空)，性别 SEX（ CHAR 型，长度为 2)，所在系 DEPTNO（ INT 型)。

```
mysql> CREATE TABLE Student(
    ->  SNO INT PRIMARY KEY,
    ->  SNAME CHAR(8) NOT NULL,
    ->  SEX CHAR(2),
    ->  DEPTNO INT
    -> );
Query OK, 0 rows affected (0.12 sec)
```

[2] 创建课程表 Course ,由以下属性组成 :课程号 CNO(INT 型),课程名 CNAME(CHAR 型 ,长度为 20 ,非空),授课教师编号 TNO(INT 型),学分 CREDIT(INT 型)。其中(CNO，TNO ）为主键。

```
mysql> CREATE TABLE Course(
    ->  CNO INT,
    ->  CNAME CHAR(20) NOT NULL,
    ->  TNO INT,
    ->  CREDIT INT
    -> );
Query OK, 0 rows affected (0.12 sec)
```

[3] 创建学生选课表 SC，由以下属性组成：学号 SNO，课程号 CNO，成绩 GRADE。所有属性均为 INT 型，其中（ SNO，CNO ）为主键。

```
mysql> CREATE TABLE SC(
    ->  SNO INT,
    ->  CNO INT,
    ->  GRADE INT
    -> );
Query OK, 0 rows affected (0.19 sec)
```

[4] 创建教师表 Teacher ,由以下属性组成 :教师编号 TNO(INT 型 ,主键) ,教师姓名 TNAME（CHAR 型，长度为 8，非空），所在系 DEPTNO（INT 型）。

```
mysql> CREATE TABLE Teacher(
    ->  TNO INT PRIMARY KEY,
    ->  TNAME CHAR(8) NOT NULL,
    ->  DEPTNO INT
    -> );
Query OK, 0 rows affected (0.13 sec)
```

[5] 创建系表 Dept ,由以下属性组成 :系号 DEPTNO(INT 型 ,主键) ,系名 DNAME(CHAR 型，长度为 20，非空）。

```
mysql> CREATE TABLE Dept(
    ->  DEPTNO INT PRIMARY KEY,
    ->  DNAME CHAR(20) NOT NULL
    -> );
Query OK, 0 rows affected (0.13 sec)
```

2. 修改基本表

[1] 在 Student 表中加入属性 AGE（INT 型）。

```
mysql> ALTER TABLE Student ADD AGE INT;
Query OK, 0 rows affected (0.21 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

[2] 将 Student 表中的属性 AGE 类型改为 SMALLINT 型。

```
mysql> ALTER TABLE Student MODIFY AGE SMALLINT;
Query OK, 0 rows affected (0.23 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3. 删除基本表

[1] 在所有操作结束后删除 Student 表。

```
mysql> DROP TABLE Student;
Query OK, 0 rows affected (0.06 sec)
```

[2] 在所有操作结束后删除 Course 表。

```
mysql> DROP TABLE Course;
Query OK, 0 rows affected (0.08 sec)
```

[3] 在所有操作结束后删除 SC 表。

```
mysql> DROP TABLE SC;
Query OK, 0 rows affected (0.06 sec)
```

[4] 在所有操作结束后删除 Teacher 表。

```
mysql> DROP TABLE Teacher;
Query OK, 0 rows affected (0.05 sec)
```

[5] 在所有操作结束后删除 Dept 表。

```
mysql> DROP TABLE Dept;
Query OK, 0 rows affected (0.06 sec)
```

（二）数据操作

1. 插入数据

[1] 向 Student 表插入下列数据：

（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）。

```
mysql> INSERT INTO Student VALUES(1001,'张天','m',10,20);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO Student VALUES(1002,'李兰','f',10,21);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Student VALUES(1003,'陈铭','m',10,21);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Student VALUES(1004,'刘茜','f',20,21);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Student VALUES(1005,'马阳','m',20,22);
Query OK, 1 row affected (0.04 sec)
```

[2] 向 Course 表插入下列数据：

（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）。

```
mysql> INSERT INTO Course VALUES(1,'数据结构',101,4);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Course VALUES(2,'数据库',102,4);
Query OK, 1 row affected (0.05 sec)
```

```
mysql> INSERT INTO Course VALUES(3,'离散数学',103,4);
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO Course VALUES(4,'C语言程序设计',101,2);
Query OK, 1 row affected (0.05 sec)
```

[3] 向 SC 表插入下列数据：

（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）。

```
mysql> INSERT INTO SC VALUES(1001,1,80);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO SC VALUES(1001,2,85);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO SC VALUES(1001,3,78);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO SC VALUES(1002,1,78);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO SC VALUES(1002,2,82);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO SC VALUES(1002,3,86);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO SC VALUES(1003,1,92);
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO SC VALUES(1003,3,90);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO SC VALUES(1004,1,87);
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO SC VALUES(1004,4,90);
Query OK, 1 row affected (0.04 sec)
```

```
mysql> INSERT INTO SC VALUES(1005,1,85);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO SC VALUES(1005,4,92);
Query OK, 1 row affected (0.13 sec)
```

[4] 向 Teacher 表插入下列数据：

（101，张星，10）、（102，李珊，10）、

（103，赵应，10）、（104，刘田，20）。

```
mysql> INSERT INTO Teacher VALUES(101,'张星',10);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Teacher VALUES(102,'李珊',10);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Teacher VALUES(103,'赵应',10);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO Teacher VALUES(104,'刘田',20);
Query OK, 1 row affected (0.08 sec)
```

[5] 向 Dept 表插入下列数据：

（10，计算机）、（20，信息）。

```
mysql> INSERT INTO Dept VALUES(10,'计算机');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO Dept VALUES(20,'信息');
Query OK, 1 row affected (0.08 sec)
```

2. 单表查询

[1] 查询所有学生的信息。

```
mysql> SELECT *
    -> FROM Student;
+------+-------+------+--------+------+
| SNO  | SNAME | SEX  | DEPTNO | AGE  |
+------+-------+------+--------+------+
| 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 |
+------+-------+------+--------+------+
5 rows in set (0.05 sec)
```

[2] 查询所有女生的姓名。

```
mysql> SELECT SNAME
    -> FROM Student
    -> WHERE SEX = 'f';
+--------+
| SNAME  |
+--------+
| 李兰   |
| 刘茜   |
+--------+
2 rows in set (0.04 sec)
```

[3] 查询成绩在 80 到 89 之间的所有学生的选课记录，查询结果按成绩的降序排列。

```
mysql> SELECT *
    -> FROM SC
    -> WHERE GRADE >= 80 AND GRADE <= 89
    -> ORDER BY GRADE DESC;
+------+------+-------+
| SNO  | CNO  | GRADE |
+------+------+-------+
| 1004 |    1 |    87 |
| 1002 |    3 |    86 |
| 1001 |    2 |    85 |
| 1005 |    1 |    85 |
| 1002 |    2 |    82 |
| 1001 |    1 |    80 |
+------+------+-------+
6 rows in set (0.05 sec)
```

[4] 查询各个系的学生人数。

```
mysql> SELECT DEPTNO,COUNT(SNO)
    -> FROM Student
    -> GROUP BY DEPTNO;
+--------+------------+
| DEPTNO | COUNT(SNO) |
+--------+------------+
|     10 |          3 |
|     20 |          2 |
+--------+------------+
2 rows in set (0.00 sec)
```

3. 连接查询

查询信息系年龄在 21 岁以下（含 21 岁）的女生姓名及其年龄。

```
mysql> SELECT SNAME,AGE
    -> FROM Student,Dept
    -> WHERE Student.DEPTNO = Dept.DEPTNO
    -> AND Dept.DNAME = '信息'
    -> AND AGE <= 21
    -> AND SEX = 'f';
+-------+------+
| SNAME | AGE  |
+-------+------+
| 刘茜  |   21 |
+-------+------+
1 row in set (0.00 sec)
```

4. 嵌套查询

[1] 查询修课总学分在 10 分以下的学生姓名。

```
mysql> SELECT SNAME
    -> FROM Student
    -> WHERE SNO IN
    ->  (SELECT SNO
    ->   FROM SC,Course
    ->   WHERE SC.CNO = Course.CNO
    ->   GROUP BY SNO
    ->          HAVING SUM(CREDIT) < 10);
+-------+
| SNAME |
+-------+
| 陈铭  |
| 刘茜  |
| 马阳  |
+-------+
3 rows in set (0.05 sec)
```

[2] 查询各门课程取得最高成绩的学生姓名及其成绩。

```
mysql> SELECT CNO,SNAME,GRADE
    -> FROM Student,SC SC1
    -> WHERE Student.SNO = SC1.SNO AND SC1.GRADE IN
    ->  (SELECT MAX(GRADE)
    ->   FROM SC SC2
    ->   WHERE SC1.CNO = SC2.CNO
    ->   GROUP BY CNO);
+------+-------+-------+
| CNO  | SNAME | GRADE |
+------+-------+-------+
|    2 | 张天  |    85 |
|    1 | 陈铭  |    92 |
|    3 | 陈铭  |    90 |
|    4 | 马阳  |    92 |
+------+-------+-------+
4 rows in set (0.00 sec)
```

[3] 查询选修了 1001 学生选课的全部课程的学生学号。

```
mysql> SELECT SNO
    -> FROM Student
    -> WHERE NOT EXISTS
    ->  (SELECT *
    ->   FROM SC SC1
    ->   WHERE SC1.SNO = 1001 AND NOT EXISTS
    ->   (SELECT *
    ->    FROM SC SC2
    ->    WHERE SC2.SNO = Student.SNO AND SC2.cno = SC1.cno));
+------+
| SNO  |
+------+
| 1001 |
| 1002 |
+------+
2 rows in set (0.00 sec)
```

[4] 查询选修了张星老师开设的全部课程的学生姓名。

```
mysql> SELECT SNAME
    -> FROM Student
    -> WHERE NOT EXISTS
    -> (SELECT *
    ->  FROM Course
    ->  WHERE TNO IN
```

```
    ->  WHERE TNO IN
    ->  (SELECT TNO
    ->   FROM Teacher
    ->   WHERE TNAME = '张星') AND NOT EXISTS
    ->   (SELECT *
    ->    FROM SC
    ->    WHERE SC.SNO = Student.SNO AND SC.CNO = Course.CNO));
+--------+
| SNAME  |
+--------+
| 刘茜   |
| 马阳   |
+--------+
2 rows in set (0.00 sec)
```

5. 修改数据

将张星老师数据结构课的学生成绩全部加 2 分。

```
mysql> UPDATE SC
    -> SET GRADE = GRADE + 2
    -> WHERE SC.CNO IN
    -> (SELECT CNO
    ->  FROM Course
    ->  WHERE Course.CNAME = '数据结构' AND Course.TNO IN
    ->  (SELECT TNO
    ->   FROM Teacher
    ->   WHERE Teacher.TNAME = '张星'));
Query OK, 5 rows affected (0.09 sec)
Rows matched: 5  Changed: 5  Warnings: 0
```

6. 删除数据

删除马阳同学的所有选课记录。

```
mysql> DELETE FROM SC
    -> WHERE SC.CNO IN
    -> (SELECT SNO
    ->  FROM Student
    ->  WHERE Student.SNAME='马阳');
Query OK, 0 rows affected (0.05 sec)
```

（三）视图操作

1. 建立视图

在插入数据的 Student 基本表上为计算机科学与技术系的学生记录建立一个视图

CS_STUDENT。

```
mysql> CREATE VIEW CS_STUDENT(SNO,SNAME,SEX,DEPTNO,AGE)
    -> AS (SELECT *
    ->     FROM Student
    ->     WHERE DEPTNO = '计算机');
Query OK, 0 rows affected (0.11 sec)
```

2. 删除视图

在操作结束后删除视图 CS_STUDENT。

```
mysql> DROP VIEW CS_STUDENT;
Query OK, 0 rows affected (0.00 sec)
```

实验问题与解决：

问题：MySQL command line client 里无法正常显示中文。

原因：字符编码没有设置正确。

解决：将 MySQL 安装目录下的配置文件 my.ini 中的两处 default-character-set=utf8 改为 default-character-set=gbk 即可。