

河南工业大学实验报告

课程 数据库系统原理及应用 实验名称: 数据库定义及数据基本查询

院 系 信息科学与工程学院 专业班级 物联网 1603

姓 名 郭治洪 学 号 201616070320

指导老师 许贺洋 日 期 2018.9

一. 实验目的

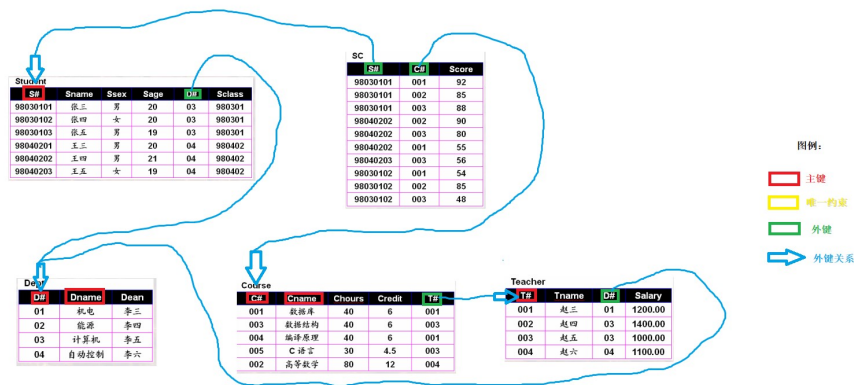
1. 掌握创建、删除数据库语句。
2. 掌握创建、修改和删除基本表语句。
3. 掌握单表数据查询功能,掌握基于单表的 ORDER BY 子句的用法。

二. 实验内容及要求

1. 运行 SQL Server 2008, 构建 SCT 数据库, 其表结构说明详见《实验二 数据库定义及数据基本查询 实验说明.doc》所示。
2. 按照《实验二 数据库定义及数据基本查询 实验说明.doc》中规定的实验任务写出 SQL 语句, 并对得出的结果进行截图。

三. 实验过程及结果

第 0 步 先把这些表关系做出一个关系图:



第 1 步 根据关系图和表属性建立数据库, 建立表:

注意这个建表一定要先建被依赖外键的表! 不然会提示错误!

-- 物联网1603 郭治洪 201616070320 数据库实验二

-- 创建数据库 学生选课数据库

```
CREATE DATABASE SCT;
```

```
GO
```

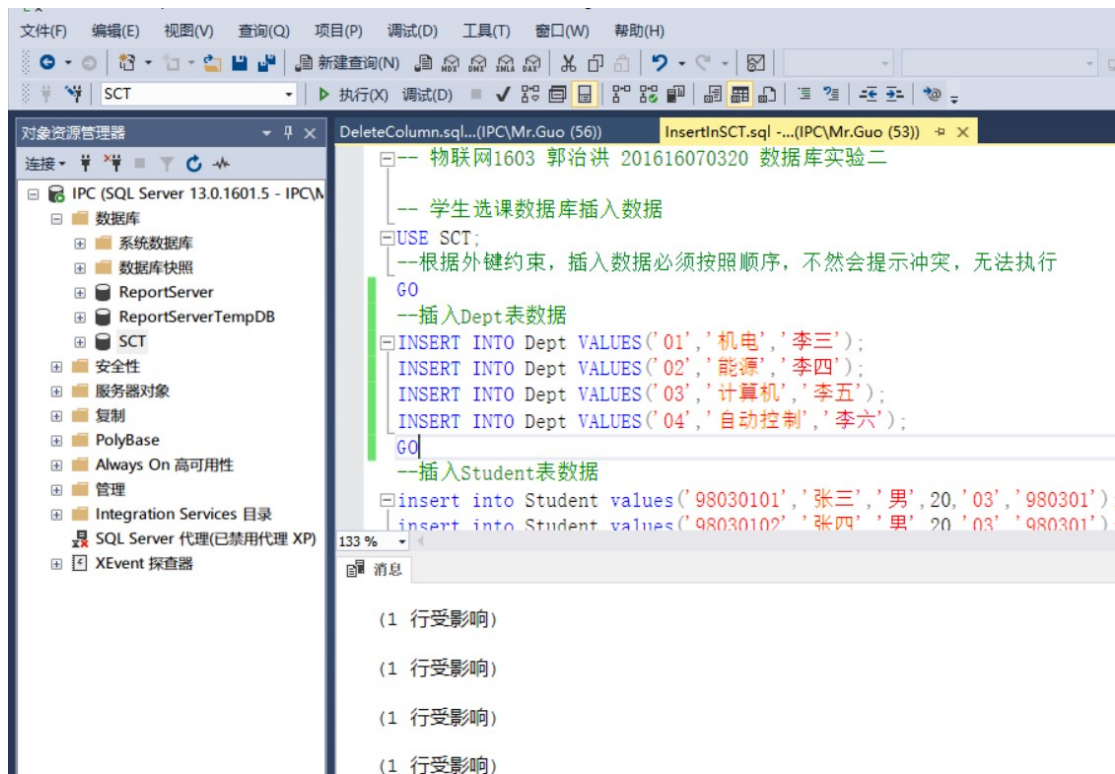
```
USE SCT; /*选中SCT数据库*/
```

```

GO
--创建院系表
CREATE TABLE Dept
(
    D# char(2) PRIMARY KEY NOT NULL,
    Dname char(10) NOT NULL,
    Dean char(10) NOT NULL, --系主任
);
GO
--创建教师表
CREATE TABLE Teacher
(
    T# char(3) PRIMARY KEY NOT NULL,
    Tname char(10) NOT NULL,
    D# char(2) NOT NULL,
    Salary float(2),
    FOREIGN KEY(D#) REFERENCES Dept(D#) --外键
);
GO
--创建课程表
CREATE TABLE Course
(
    C# char(3) PRIMARY KEY NOT NULL,
    Cname char(12) NOT NULL,
    Chours integer NOT NULL,
    Credit float(1) NOT NULL,
    T# char(3) NOT NULL,
    FOREIGN KEY(T#) REFERENCES Teacher(T#)
);
GO
--创建学生表
CREATE TABLE Student
(
    S# char(8) PRIMARY KEY NOT NULL, --主键为学号
    Sname char(10) NOT NULL,
    Ssex char(2) NOT NULL,
    Sage integer NOT NULL,
    D# char(2) NOT NULL,
    Sclass char(6) NOT NULL,
    FOREIGN KEY(D#) REFERENCES Dept(D#) --外键约束
);
GO
--创建选课表
CREATE TABLE SC

```

```
(
    S# char(8) NOT NULL,
    C# char(3) NOT NULL,
    Score float(1)
    FOREIGN KEY(S#) REFERENCES Student(S#),
    FOREIGN KEY(C#) REFERENCES Course(C#)
);
```



第 2 步 插入数据到表中

同样这个插入有顺序不然会提示错误！

然后查询插入的数据：

```
-- 物联网1603 郭治洪 201616070320 数据库实验二

-- 学生选课数据库插入数据
USE SCT;
--根据外键约束，插入数据必须按照顺序，不然会提示冲突，无法执行
GO
--插入Dept表数据
INSERT INTO Dept VALUES('01','机电','李三');
INSERT INTO Dept VALUES('02','能源','李四');
INSERT INTO Dept VALUES('03','计算机','李五');
```

```

INSERT INTO Dept VALUES ('04', '自动控制', '李六');
GO
--插入Student表数据
insert into Student values ('98030101', '张三', '男', 20, '03', '980301');
insert into Student values ('98030102', '张四', '男', 20, '03', '980301');
insert into Student values ('98030103', '张五', '男', 19, '03', '980301');
insert into Student values ('98040201', '王三', '男', 20, '04', '980402');
insert into Student values ('98040202', '王四', '男', 21, '04', '980402');
insert into Student values ('98040203', '王五', '女', 19, '04', '980402');
GO
--插入Teacher表数据
INSERT INTO Teacher VALUES ('001', '赵三', '01', 1200.00);
INSERT INTO Teacher VALUES ('002', '赵四', '03', 1400.00);
INSERT INTO Teacher VALUES ('003', '赵五', '03', 1000.00);
INSERT INTO Teacher VALUES ('004', '赵六', '04', 1100.00);
GO
--插入Course表数据
INSERT INTO Course VALUES ('001', '数据库', 40, 6, '001');
INSERT INTO Course VALUES ('003', '数据结构', 40, 6, '003');
INSERT INTO Course VALUES ('004', '编译原理', 40, 6, '001');
INSERT INTO Course VALUES ('005', 'C 语言', 30, 4.5, '003');
INSERT INTO Course VALUES ('002', '高等数学', 80, 12, '004');
GO
--插入SC表数据
INSERT INTO SC VALUES ('98030101', '001', 92);
INSERT INTO SC VALUES ('98030101', '002', 85);
INSERT INTO SC VALUES ('98030101', '003', 88);
INSERT INTO SC VALUES ('98040202', '002', 90);
INSERT INTO SC VALUES ('98040202', '003', 80);
INSERT INTO SC VALUES ('98040202', '001', 55);
INSERT INTO SC VALUES ('98040203', '003', 56);
INSERT INTO SC VALUES ('98030102', '001', 54);
INSERT INTO SC VALUES ('98030102', '002', 85);
INSERT INTO SC VALUES ('98030102', '003', 48);
GO

```

结果

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	男	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402

98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

(6 行受影响)

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

(4 行受影响)

T#	Tname	D#	Salary
001	赵三	01	1200
002	赵四	03	1400
003	赵五	03	1000
004	赵六	04	1100

(4 行受影响)

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
002	高等数学	80	12	004
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003

(5 行受影响)

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85

98030102 003 48

(10 行受影响)

第 4 步 根据需要查询需要的信息

```
USE SCT;
-- (1) 检索教师表中所有工资少于1500元或者工资大于2000元，并且是03系的教师姓名和工资。
SELECT Tname,Salary
FROM Teacher
WHERE D# = '03' AND ( Salary <1500 OR Salary >2000 );
-- (2) 检索学生表(Student)中所有年龄大于19岁的学生的年龄及姓名。
SELECT Sname,Sage
FROM Student
WHERE Sage>19
-- (3) 在选课表中，检索成绩大于80分的所有学生的学号和姓名，要求确保结果的唯一性。
--DISTINCT 关键字用来去重
SELECT distinct Student.S#,Student.Sname
FROM SC,Student
WHERE ( Score > 80 ) AND (Student.S#=SC.S#)
-- (4) 检索选课表中所有不及格的学生的学号、课程号和成绩。
SELECT S#,C#,Score
FROM SC
WHERE ( Score < 60 )
-- (5) 检索选课表的所有信息，并按成绩由高到低的顺序进行排序。
SELECT *
FROM SC
ORDER BY Score DESC ;
```

查询结果:

Tname	Salary
-------	--------

赵四	1400
赵五	1000

(2 行受影响)

Sname	Sage
-------	------

张三	20
张四	20

王三	20
王四	21

(4 行受影响)

S#	Sname

98030101	张三
98030102	张四
98040202	王四

(3 行受影响)

S#	C#	Score

98040202	001	55
98040203	003	56
98030102	001	54
98030102	003	48

(4 行受影响)

S#	C#	Score

98030101	001	92
98040202	002	90
98030101	003	88
98030101	002	85
98030102	002	85
98040202	003	80
98040203	003	56
98040202	001	55
98030102	001	54
98030102	003	48

(10 行受影响)

附录：删除表中的所有数据

注意这个也要求有顺序，先删除有外键的表。

--删掉表中所有的行

--先删关联关系最少的表

USE SCT;

```
DELETE FROM SC;  
DELETE FROM Course;  
DELETE FROM Teacher;  
DELETE FROM Student;  
DELETE FROM Dept;
```

四. 实验中的问题及心得

由于主外键的原因，MSSQL 建立表删除表,插入数据修改数据都必须按照一定的顺序，所以搞清表的结构是很有必要的!!!

这就是我做的实验这么慢的原因!!!