

# 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 实验四 数据更新和视图

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. 10

## 一. 实验目的

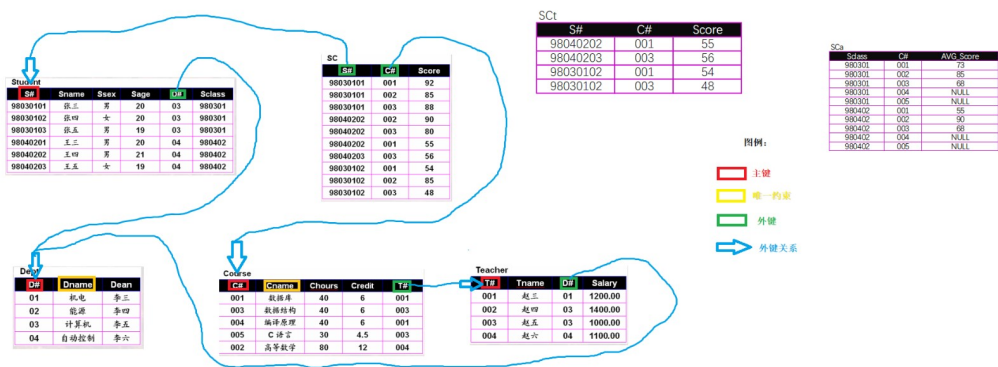
1. 掌握连接查询、嵌套查询（IN、EXISTS）的语句构造及其相互转化；
2. 理解和掌握 INSERT、UPDATE 和 DELETE 语法结构的各个组成成分，结合嵌套 SQL 子查询，分别设计几种不同形式的插入、修改和删除数据的语句，并调试成功；
3. 了解视图与基本表的联系与区别。

## 二. 实验内容及要求

1. 按照《实验四 数据更新和视图 实验说明.doc》中规定的实验任务写出 SQL 语句，并得出规定的结果。

## 三. 实验过程及结果

A 关系模式图



B 实验脚本

```
USE SCT; --不要忘了这句，要不然可能查不到
--别忘了这句!!!!
--别忘了!!!!
--不然可能出事情的!!!!
--一定要小心!!!!
--一定要谨慎!!!!
--记住!!!!
--（1）检索有薪水差额的任意两位教师的姓名及二者的工资差额；

Select DISTINCT T1.Tname, T2.Tname,( T1.Salary - T2.Salary) AS
[Difference]
From Teacher AS T1, Teacher AS T2
```

```

Where T1.Salary != T2.Salary;
-- (2) 检索“001”号课成绩比“002”号课成绩高的所有学生的姓名、001号课程成绩、002号
课程成绩及成绩差;
SELECT DISTINCT Student.Sname,
    SC1.Score AS [Score(001)],
    SC2.Score AS [Score(002)],
    (SC1.Score-SC2.Score) AS [Difference]
FROM SC AS SC1,
    SC AS SC2,
    Student
WHERE (SC1.C#='001' AND SC2.C#='002')
    AND (SC1.S#=SC2.S#)
    AND (SC1.Score>SC2.Score)
    AND (SC1.S#=Student.S#);
-- (3) 检索选修‘001’号课程有成绩差的任意两位同学的学号和成绩差;
SELECT DISTINCT SC1.S# AS [S1.S#],SC2.S# AS [S2.S#],
    (SC1.Score-SC2.Score) AS [Difference]
FROM SC AS SC1,
    SC AS SC2
WHERE (SC1.Score!=SC2.Score)
    AND (SC1.S#!=SC2.S#)
    AND (SC1.C#='001')
    AND (SC2.C#='001')
    AND (SC1.C#=SC2.C#);
-- (4) 检索选修了001号课程的学生的学号和姓名;
SELECT Student.Sname,SC.S#
FROM Student,SC
WHERE (Student.S#=SC.S#) AND (SC.C#='001');
-- (5) 检索有两门及以上不及格(超过一门)课程同学的学号及其平均成绩;
--这里平时成绩是他整个的平均成绩
SELECT S#,AVG(Score)
FROM SC
--括号内为不及格超过一门的同学的学号
WHERE S# IN(SELECT S#
    FROM SC
    WHERE Score <60
    GROUP BY S#
    HAVING COUNT(*) >=2)
GROUP BY S#;
-- (6) 找出001号课成绩最高的学生的学号;
SELECT S#
FROM SC
WHERE C#='001'
    AND Score=(SELECT MAX(Score) FROM SC);

```

```

--成绩等于最高值，可以用SELECT MAX(Score) FROM SC 选出
-- (7) 找出张三同学成绩最低的课程号；
SELECT Min(SC.Score) AS [张三最低的成绩]
FROM SC
WHERE SC.Score IN (SELECT SC.Score
                    FROM SC,Student
                    WHERE SC.S#=Student.S#
                    AND Student.Sname='张三');
-- (8) 列出没学过李明老师讲授任何一门课程的所有同学的姓名；
/*
    最内层查询：从Teacher表查询李明老师的工号 (T#)
    次内层查询：从Course表查询所有由李明老师的工号 (T#) 所教的课程号 (C#)
        联立条件：Course.T#=Teacher.T#
    次外层查询：从SC表查询所有没有由李明老师的工号 (T#) 所教的课程号 (C#) 的学
    生的学号 (S#)
        联立条件：Course.C#=SC.C#
    最外层查询：从Student表查询所有没有由李明老师的工号 (T#) 所教的课程号 (C#)
    的学生的学号 (S#) 的学生姓名 (Sname)
        联立条件：SC.S#=Student.S#
    其中联立条件是每个表的外键，即每层AND后面是与外层查询的表连接条件，即选择查
    询的内容
*/
SELECT Sname FROM Student --查询所有没选择查询由李明老师的工号所教的课程号
的学生的学号的学生姓名
WHERE EXISTS
(
    SELECT S# FROM SC --查询所有没选择查询由李明老师的工号所教的课程号的学
    生的学号
    WHERE NOT EXISTS
    (
        SELECT C# FROM Course --查询由李明老师的工号所教的课程号
        WHERE EXISTS
        (
            SELECT T# FROM Teacher --查询李明老师的工号
            WHERE Teacher.Tname='李明'
            AND Course.T#=Teacher.T#
        )
        AND Course.C#=SC.C#
    )
    AND SC.S#=Student.S#
);
-- (9) 列出学过98030101号同学学过所有课程的同学的学号；
/*
    内层查询：从SC表查询98030101号同学学过所有课程课程号

```

外层查询：从SC表查询学号学过98030101号同学学过所有课程课程号的同学

```
*/  
  
SELECT DISTINCT SC1.S# FROM SC AS SC1  
WHERE EXISTS  
(  
    SELECT SC2.C# FROM SC AS SC2  
    WHERE S#='98030101'  
        AND SC2.C#=SC1.C#  
);  
/*
```

!!! 所有涉及更改，插入，新建新表以及删除操作，都应该在第二次执行之前将结果重置为默认值，不然将会无法检验结果正确性。!!!

!!! 所有涉及增删改操作测试都应该在测试之前将所有已经更改（增加或者删除）的结果重置!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

```
*/
```

-- (10) 新建Table: SCt(S#, C#, Score), 将检索到的成绩不及格同学的记录新增到该表中;

--执行查询并且添加新表任务

```
SELECT S#,C#,Score  
INTO SCt --直接将结果插入新表  
FROM SC  
WHERE ( Score < 60 );
```

--检索插入的新表

```
SELECT * FROM SCt;
```

-- (11) 从SCt表中删除有两门不及格课程的所有同学;

```
DELETE FROM SCt
```

--WHERE 是要删除条件, 括号内内容应该为去掉重复的两门不及格课程的所有同学的学号

```
WHERE S# IN (SELECT DISTINCT SCt.S#  
            FROM SCt  
            WHERE Score <60  
            GROUP BY SCt.S#  
            HAVING COUNT(*) =2);
```

--检索删除后的新表

```
SELECT * FROM SCt;
```

--做增删改测试之后应该将更改（增加，删除）的值予以重置，否则将会影响第二次测试结果!!!

```
/*
```

!!! 所有涉及更改，插入，新建新表以及删除操作，都应该在第二次执行之前将结果重

置为默认值，不然将会无法检验结果正确性。!!!

!!! 所有涉及增删改操作测试都应该在测试之前将所有已经更改（增加或者删除）的结果重置!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 否则之前结果会影响第二次执行结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

!!! 应该在第一次测试后就重置，而不是第二次开始前重置，否则会同样影响结果!!!

\*/

**DROP TABLE** SCt;

-- (12) 将张三同学001号课的成绩置为其班级该门课的平均成绩。

--题意：查询张三同学班级001号课的平均成绩并将其置为张三同学001号课的成绩

--首先创建这个表出来

**SELECT DISTINCT** Student.Sclass **AS** [Sclass],  
Course.C# **AS** [C#]

**INTO** SCa

**FROM** Student, Course;

--在这个表中添加AVG\_Score列

**ALTER TABLE** SCa

**ADD** AVG\_Score **float**(1);

--检索新建结果

**SELECT \* FROM** SCa;

--计算每个班每门课的平均值插入结果到SCa表

**UPDATE** SCa **SET** SCa.AVG\_Score = Temp.AVG\_Score

**FROM**(**SELECT DISTINCT** Student.Sclass **AS** [Sclass], SC.C# **AS**  
[C#], **AVG**(SC.Score) **AS** [AVG\_Score]

**FROM** Student, SC

**WHERE** Student.S#=SC.S#

**GROUP BY** Student.Sclass, SC.C#) **AS** Temp

**WHERE** SCa.Sclass=Temp.Sclass **AND** SCa.C#=Temp.C#;

--检索更改结果

**SELECT \* FROM** SCa;

--将张三同学001号课的成绩置为其班级该门课的平均成绩

--用该班该门课的平均成绩覆盖张三同学001号课的成绩

**UPDATE** SC **SET** SC.Score = TEMP.Score

--括号内查询的是张三该班001号课平均的成绩

--UPDATE是张三同学的001号课的成绩

**FROM** (**SELECT** SCa.AVG\_Score **AS** [Score]

**FROM** SCa, Student

**WHERE** Student.Sclass=SCa.Sclass

**AND** Student.Sname='张三'

**AND** SCa.C#='001') **AS** TEMP, Student

**WHERE** Student.S#=SC.S#

```

        AND Student.Sname='张三'
        AND SC.C#='001';
--检索更改结果
SELECT * FROM SC;
/*
    !!! 所有涉及更改，插入，新建新表以及删除操作，都应该在第二次执行之前将结果重置为默认值，不然将会无法检验结果正确性。!!!
    !!! 所有涉及增删改操作测试都应该在测试之前将所有已经更改（增加或者删除）的结果重置!!!
    !!! 否则之前结果会影响第二次执行结果!!!
    !!! 否则之前结果会影响第二次执行结果!!!
    !!! 否则之前结果会影响第二次执行结果!!!
    !!! 应该在第一次测试后前就重置，而不是第二次开始前重置，否则会同样影响结果!!!
    !!! 应该在第一次测试后前就重置，而不是第二次开始前重置，否则会同样影响结果!!!
    !!! 应该在第一次测试后前就重置，而不是第二次开始前重置，否则会同样影响结果!!!
*/
--重置张三同学001号课的成绩（不重置会影响第二次的的结果）
UPDATE SC SET SC.Score = 92
FROM Student
WHERE Student.S#=SC.S#
        AND Student.Sname='张三'
        AND SC.C#='001';
--检索更改结果
SELECT * FROM SC;
-- 重置为增删改前默认的状态（不重置会影响第二次的的结果）
DROP TABLE SCa;

```

## C 实验结果

Tname	Tname	Difference
赵六	赵三	-100
赵六	赵四	-300
赵六	赵五	100
赵三	赵六	100
赵三	赵四	-200
赵三	赵五	200
赵四	赵六	300
赵四	赵三	200
赵四	赵五	400
赵五	赵六	-100
赵五	赵三	-200
赵五	赵四	-400

(12 行受影响)

Sname	Score(001)	Score(002)	Difference
张三	92	85	7

(1 行受影响)

S1.S#	S2.S#	Difference
98030101	98030102	38
98030101	98040202	37
98030102	98030101	-38
98030102	98040202	-1
98040202	98030101	-37
98040202	98030102	1

(6 行受影响)

Sname	S#
张三	98030101
王四	98040202
张四	98030102

(3 行受影响)

S#
98030102 62.33333333333333

(1 行受影响)

S#
98030101

(1 行受影响)

张三最低的成绩
85

(1 行受影响)

Sname

-----

张三

张四

王四

王五

(4 行受影响)

S#

-----

98030101

98030102

98040202

98040203

(4 行受影响)

(4 行受影响)

S#	C#	Score
----	----	-------

-----

98040202	001	55
----------	-----	----

98040203	003	56
----------	-----	----

98030102	001	54
----------	-----	----

98030102	003	48
----------	-----	----

(4 行受影响)

(2 行受影响)

S#	C#	Score
----	----	-------

-----

98040202	001	55
----------	-----	----

98040203	003	56
----------	-----	----

(2 行受影响)

(10 行受影响)

Sclass	C#	AVG_Score
--------	----	-----------

-----

980301	001	NULL
--------	-----	------



```
980301 002 NULL
980301 003 NULL
980301 004 NULL
980301 005 NULL
980402 001 NULL
980402 002 NULL
980402 003 NULL
980402 004 NULL
980402 005 NULL
```

(10 行受影响)

(6 行受影响)

```
Sclass C#  AVG_Score
-----
```

```
980301 001  73
980301 002  85
980301 003  68
980301 004 NULL
980301 005 NULL
980402 001  55
980402 002  90
980402 003  68
980402 004 NULL
980402 005 NULL
```

(10 行受影响)

(1 行受影响)

```
S#      C#    Score
-----
```

```
98030101 001  73
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)

(1 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)

#### 四. 实验中的问题及心得

这个实验，讲真有点难度，但是我很喜欢研究。

说说遇得的坑吧：

比如刚开始查询忘加选择数据库查询，造成查询结果错误，后来发现是之前创建同样没有指定数据库创建表在系统 master 数据库里了，而且这个问题我还犯了不止一遍，我觉得我应该避免，这就相当于删除表没有加条件一样可怕，如果有重要数据再没备份不会操作，那么就要大难临头了，所以我必须细心而又认真，避免这种错误，另外权限越大，我们责任也越大，所以分级权限和及时备份还是有必要的。

另外做后面的题，多重嵌套查询，也是测了很多遍，一次又一次，不过我总结要从内层向外层一点点查询，注意联立条件，理解清表的关系模型图（就是我上面画的那个），这点真的看的很方便。

做到后面最后一题，由于我不是很懂题意，所以我就用多表联合查询把查询结果到一个新表中，同时我也研究如何更改这个表的某一列的值，学习了 INSERT INTO 也可除 UPDATE 修改表的值（根据查询结果联合条件插入），不过这一题我觉得还需要和老师交流反馈。

然后和老师交流终于理解了题意，同时也修改了自己不正确的地方。

另外我还意识到增删改操作会影响数据库，造成第二次查询等结果就会不一样，因此如果测试要及时恢复默认值，同时也要提醒我小心，明白在干什么，知道后果，另外也要会恢复。

虽然坑很多，但是我砸了大笔时间学习，debug，我觉得我学的深入些，对我未来有好处，所以我会继续努力。

就这样吧，干了几天了，有点累，早点睡吧。

另外感谢老师对我的帮助。

谢谢。