浙江工业大学

数据库原理及应用实验报告

(2021 级)

实验题目　实验 6　参照完整性

学生姓名　温家伟

学生学号　202103151422

学科(专业)　数据科学与大数据技术(分析方向)2101 班

所在学院　理学院

提交日期　2023 年 4 月 14 日

1.1 实验目的

学习建立外键，以及利用`FOREIGN KEY…REFERENCES`子句以及各种约束保证参照完整性。

1.2 实验内容

1.2.1 为演示参照完整性，建立表:

```
学生表    Students(Sno,Sname, Semail,Scredit,Sroom);
教师表    Teachers(Tno,Tname,Temail,Tsalary);
课程表    Courses(Cno,Cname,Ccredit);
成绩表    STC(Sno,Tno,Cno, Score);
```

注意：

```
CONSTRAINT FK_Stu_STC FOREIGN KEY(Sno) REFERENCES Students(sno) ON DELETE
CASCADE,
CONSTRAINT FK_Tea_STC FOREIGN KEY(Tno) REFERENCES Teachers,
CONSTRAINT FK_Cou_STC FOREIGN KEY(Cno) REFERENCES Courses
```

提示：可以用一下命令修改约束

```
ALTER TABLE stc Drop CONSTRAINT FK_Stu_STC;
ALTER TABLE stc ADD CONSTRAINT FK_Stu_STC FOREIGN KEY(Sno) REFERENCES
Students(sno);
```

并插入相应数据。为下面的实验步骤做预先准备。

查看STC表中定义了什么约束：`\d STC`;

```
SELECT
tc.constraint_name
FROM
information_schema.table_constraints AS tc
JOIN information_schema.key_column_usage AS kcu ON tc.constraint_name =
kcu.constraint_name
JOIN information_schema.constraint_column_usage AS ccu ON ccu.constraint_name =
tc.constraint_name
WHERE constraint_type = 'FOREIGN KEY' AND tc.table_name = 'stc';
```

1.2.2认识参照完整性：

在不违反参照完整性的前提下，插入数据；在违反参照完整性的前提下，插入数据；在违反参照完整性的前提下，删除数据；

1.2.3 演示级联删除。

1.2.4 针对课程表courses来重复1.2.3的工作。

1.2.5 综合应用。

1.3 实验步骤

以系统管理员或sa账号登录查询分析器，在查询分析器窗体中输入如下命令，运行并观察和记录结果。

1.3.1 在查询分析器中输入如下SQL语句：

```
Set search_path =XXXschema,public;
```

1.3.2 认识参照完整性。输入如下SQL语句：

```
insert into stc values('222','T05','C95',90);
```

```
 public | teachers | table | omm    | {orientation=row,compression=no}
(4 rows)

ahweidb=# insert into stc values('222','T05','C95',90);
ERROR:  insert or update on table "stc" violates foreign key constraint "fk_stu_rep"
DETAIL:  Key (sno)=(222) is not present in table "students".
ahweidb=#
```

分析：违反了参照完整性，因为被参照关系表students中不存在Sno为'222'的记录。

在学生表中插入学号222，姓名jxmsc，的记录；

```
insert into students(sno,sname) values('222', 'jxmsc');
insert into stc values('222','T05','C95',90);
```

```
ahweidb=# insert into students(sno,sname) values('222', 'jxmsc');
INSERT 0 1
ahweidb=# insert into stc values('222','T05','C95',90);
ERROR:  insert or update on table "stc" violates foreign key constraint "fk_cou_rep"
DETAIL:  Key (cno)=(C95) is not present in table "courses".
ahweidb=#
```

分析：违反了参照完整性，因此被参照关系表courses中不存在cno为'C95'的记录。

```
insert into stc values('222','T05','C05',90);
```

```
DETAIL:  Key (cno)=(C95) is not present in table "courses".
ahweidb=# insert into stc values('222','T05','C05',90);
INSERT 0 1
ahweidb=#
```

下面考察删除被引用的学生记录：

```
delete from students where sno='222';
```

```
ahweidb=# delete from students where sno='222';
ERROR:  update or delete on table "students" violates foreign key constraint "fk_stu_rep" on table "stc"
DETAIL:  Key (sno)=(222) is still referenced from table "stc".
ahweidb=#
```

分析：违反了参照完整性，因为参照关系表STC引用了被参照表students中的学号222。

1.3.3 级联删除实验：

删除原先的约束STC与students之间的引用关系：

```
ALTER TABLE STC DROP CONSTRAINT FK_Stu_STC;
delete from students where sno='222';
```

删除成功；

```
xxxdb=# delete from students where sno='222';
DELETE 1
```

查看STC中有无'222'的选课记录。

```
ylhdb=# select * from stc where sno='222';
sno | tno | cno | score
-----+-----+-----+-------
222 | T05 | C05 | 90.0
(1 row)
```

```
ahweidb=# ALTER TABLE stc DROP CONSTRAINT fk_stu_rep;
ALTER TABLE
ahweidb=# delete from students where sno='222';
DELETE 1
ahweidb=# select * from stc where sno='222';
 sno | tno | cno | score
-----+-----+-----+-------
 222 | T05 | C05 |  90.0
(1 row)
```

恢复students中的记录：

```
insert into students (sno,sname) values('222', 'jxmsc');
```

考察级联删除：

```
ALTER TABLE STC ADD CONSTRAINT FK_Stu_STC FOREIGN KEY(sno) REFERENCES
Students(sno) MATCH SIMPLE ON DELETE CASCADE;
//no action on update no action;
xxxdb=# DELETE FROM Students WHERE Sno='222';
DELETE 1
```

检查是否级联把STC中Sno='222'所有选课记录删除了：

```
xxxdb=# select * from STC where sno='222';
sno | tno | cno | score
-----+-----+-----+-------
(0 rows)
```

```
ahweidb=# insert into students (sno,sname) values('222', 'jxmsc');
INSERT 0 1
ahweidb=# ALTER TABLE STC ADD CONSTRAINT FK_Stu_STC FOREIGN KEY(sno) REFERENCES Students(sno) MATCH SIMPLE ON DELETE CASCADE;
ALTER TABLE
ahweidb=# DELETE FROM Students WHERE Sno='222';
DELETE 1
ahweidb=# select * from STC where sno='222';
 sno | tno | cno | score
-----+-----+-----+-------
(0 rows)

ahweidb=#
```

分析：由于ON DELETE CASCADE 的连带删除作用，当Students中删除某个学号时，STC中对应这个学号为外键的所有记录都要被删除。

危险游戏。

输入如下SQL语句：

```
Begin Transaction;
DELETE FROM Students WHERE sno ='S01';
Select * from STC WHERE sno ='S01';
Rollback;
```

```
ahweidb=# Begin Transaction;
BEGIN
ahweidb=# DELETE FROM Students WHERE sno ='S01';
DELETE 1
ahweidb=# Select * from STC WHERE sno ='S01';
 sno | tno | cno | score
-----+-----+-----+-------
(0 rows)

ahweidb=# Rollback;
ROLLBACK
ahweidb=#
```

1.3.4 针对课程表courses来重复（3）的工作。

```
ahweidb=# ALTER TABLE stc DROP CONSTRAINT fk_cou_rep;
ALTER TABLE
ahweidb=# delete from courses where cno = 'C01';
DELETE 1
ahweidb=# select * from stc where cno = 'c01';
 sno | tno | cno | score
-----+-----+-----+-------
(0 rows)

ahweidb=# select * from stc where cno = 'C01';
 sno | tno | cno | score
-----+-----+-----+-------
 S01 | T01 | C01 |  83.0
 S02 | T01 | C01 |  75.0
 S03 | T01 | C01 |  78.0
(3 rows)
```

```
ahweidb=# select * from STC where cno = 'C01';
 sno | tno | cno | score
-----+-----+-----+-------
(0 rows)
```

分析：由于ON DELETE CASCADE的连带删除作用，当Course中删除某个课号时，STC中对应这个课号为外键的所有记录都要被删除。

1.3.5 针对所建模式，如何插入数据?

```
学生表    Students(Sno,Sname, Semail,Scredit,Sroom);
教师表    Teachers(Tno,Tname,Temail,Tsalary);
课程表    Courses(Cno,Cname,Ccredit);
成绩表    STC(Sno,Tno,Cno, Score);
```

先插入Students、Teachers、Courses三张表，因为他们无关联，最后插入STC表。