



中山大学数据科学与计算机学院

移动信息工程专业-数据库系统

本科生实验报告

(2017-2018 学年秋季学期)

课程名称：数据库系统实验

教学班级	15M1	专业（方向）	移动互联网
学号	15352408	姓名	张镓伟

一、实验目的

1. 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。

二、实验内容

- 1.不违反参照完整性的插入数据实例。
- 2.违反参照完整性的插入数据实例。
- 3.级联删除。
- 4.两张表的互相参照问题。

三、实验准备

为了演示参照完整性，先建立两个表，为下面的实验示例做准备。

- 1) 在数据库 school 中建立表 Stu_Union,设置 sno 为主键。

```
SQLQuery2.sql - (...BRTDQO\dgn (53))* SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 USE School
2 CREATE TABLE Stu_Union(sno CHAR(5) NOT NULL UNIQUE,
3                          sname CHAR(8),
4                          ssex CHAR(1),
5                          sage INT,
6                          sdept CHAR(20),
7                          CONSTRAINT PK_Stu_Union PRIMARY KEY(sno) );
8 insert into Stu_Union values ('10001','李勇','0',24,'EE');
9 insert into Stu_Union values ('95002','王敏','1',23,'CS');
10 insert into Stu_Union values ('95003','王浩','0',25,'EE');
11 insert into Stu_Union values ('95005','王杰','0',25,'EE');
12 insert into Stu_Union values ('95009','李勇','0',25,'EE');
13 select * from stu_union;
```

	sno	sname	ssex	sage	sdept
1	10001	李勇	0	24	EE
2	95002	王敏	1	23	CS
3	95003	王浩	0	25	EE
4	95005	王杰	0	25	EE
5	95009	李勇	0	25	EE

- 2) 在数据库 school 中建立表 Course,设置 cno 为主键。



```
SQLQuery 1.341 - (10.01.2020) (53)
USE School
CREATE TABLE Course(
    cno CHAR(4) NOT NULL UNIQUE,
    cname CHAR(50),
    cpoints int,
    CONSTRAINT PK primary key (cno)
);
insert into Course values ('0001','ComputerNetworks',2);
insert into Course values ('0002','Database',3);
select * from Course;
```

	cno	cname	cpoints
1	0001	ComputerNetworks	2
2	0002	Database	3

3) 建立表 SC, 令 sno 和 cno 分别为参照 Stu_Union 表以及 Course 表的外键, 设定为级联删除, 并令 (sno, cno) 为其主键。在不违反参照完整性的前提下, 插入数据。

```
USE School
CREATE TABLE SC(
    sno char(5) REFERENCES Stu_Union (sno) on delete cascade,
    cno char(4) REFERENCES Course(cno) on delete cascade,
    grade INT,
    CONSTRAINT PK_SC PRIMARY KEY (sno,cno)
);
insert into SC values ('95002','0001',2);
insert into SC values ('95002','0002',2);
insert into SC values ('10001','0001',2);
insert into SC values ('10001','0002',2);
select * from SC;
```

	sno	cno	grade
1	10001	0001	2
2	10001	0002	2
3	95002	0001	2
4	95002	0002	2

4) 为了演示多重级联删除, 建立 Stu_Card 表, 令 card_id 为主键, 并令 stu_id 为参照 student 表的外键, 并插入数据。再建立表 ICBC_Card 表, 令 card_id 为主键, 令 stu_card_id 为参照 Stu_Card 表的外键, 并插入数据。通过删除 Stu_Union 表中一条记录, 演示三个表的多重级联删除。

```
USE School
CREATE TABLE Stu_Card(
    card_id char(14),
    stu_id char(10) references students(sid) on delete cascade,
    remained_money decimal (10,2),
    constraint PK_stu_card Primary key (card_id)
);
insert into Stu_Card values ('05212567','800001216',100.25);
insert into Stu_Card values ('05212222','800005753',200.50);
select * from Stu_Card;
```

	card_id	stu_id	remained_money
1	05212222	800005753	200.50
2	05212567	800001216	100.25

5) 建立表 ICBC_Card 表。

```
USE School
CREATE TABLE ICBC_Card(
    bank_id char(20),
    stu_card_id char(14) references stu_card(card_id) on delete cascade,
    restored_money decimal(10,2),
    constraint PK_Icbc_card Primary key (bank_id)
);
insert into ICBC_Card values ('9558844022312','05212567',15000.1);
insert into ICBC_Card values ('9558844023645','05212222',50000.3);
select * from ICBC_Card ;
```

	bank_id	stu_card_id	restored_money
1	9558844022312	05212567	15000.10
2	9558844023645	05212222	50000.30

四、 实验过程及结果

(1) 用 alter table 语句将 SC 表中的 on delete cascade 改为 on delete no action,重新插入 SC 的数据（按照实验一）。再删除 Stu_Union 中 sno 为‘10001’的数据。观察结果，并分析原因。

```
SQLQuery1.sql - (...PUT freedom (5.3))
delete from Stu_Union
where sno='10001'
```

消息 547, 级别 16, 状态 0, 第 1 行
DELETE 语句与 REFERENCE 约束“FK_SC_sno”冲突。该冲突发生于数据库“School”, 表“dbo.SC”, column ‘sno’。
语句已终止。

结果及原因:

数据库删除失败，因为 SC 表中的外键属性改为了 on delete no action（即当从表中有匹配的记录时，主表中相应的候选键不允许 update/delete 操作）所以多重级联删除到 SC 无法执行，于是整个事务回滚。

(2) 用 alter table 语句将 SC 表中的 on delete no action 改为 on delete set NULL,重新插入 SC 的数据（按照实验一）。再删除 Stu_Union 中 sno 为‘10001’的数据。观察结果，并分析原因。

SQL 语句及结果:

```
Alter table SC
drop constraint FK_SC_cno;
Alter table SC
drop constraint FK_SC_sno;
Alter table SC
add constraint FK_SC_cno foreign key (cno)
references Course(cno) on delete set NULL
Alter table SC
add constraint FK_SC_sno foreign key (sno)
references Stu_Union(sno) on delete set NULL
```

消息 1761, 级别 16, 状态 0, 第 5 行
由于一个或多个引用列不可为 Null, 因此无法使用 SET NULL 引用操作创建外键 “FK_SC_cno”。
消息 1750, 级别 16, 状态 0, 第 5 行
无法创建约束。请参阅前面的错误消息。

分析: 因为 Course.cno 和 Stu_Union.sno 都设置了不允许 NULL, 所以无法创建这



种外键。

(3) 建立事务 T3，修改 ICBC_Card 表的外键属性，使其变为 on delete set NULL, 尝试删除 students 表中一条记录。观察结果，并分析原因。

SQL 语句及结果：

```
Alter table ICBC_Card
    drop constraint FK_ICBC_Card_stu_c_46E78A0C;
Alter table ICBC_Card
    add constraint FK_ICBC_Card_stu_c foreign key (stu_card_id)
        references stu_card(card_id) on delete set NULL
```

消息
命令已成功完成。

再修改一下 choices 的约束

```
alter table choices add
    Constraint FK_CHOICES_STUDENTS FOREIGN KEY (sid)
        references students(sid) on delete cascade
```

消息
命令已成功完成。

删除：

```
Begin Transaction T3
    delete from STUDENTS where sid='800001216'
    select * from ICBC_Card
Commit Transaction T3
```

结果 消息

	bank_id	stu_card_id	restored_money
1	9558844022312	NULL	15000.10
2	9558844023645	05212222	50000.30

```
select * from stu_Card
```

结果 消息

	card_id	stu_id	remained_money
1	05212222	800005753	200.50

结果分析：从 students 表删除了 sid= '800001216' 的学生，由于级联删除，stu_card 表中对应记录也被删除，而 ICBC_Card 引用了 stu_card 的外键，因为设置了 set NULL，stu_card 项被删除后，ICBC_Card 对应那一项变成了 NULL。

(4) 用创建一个班里的学生互助表，规定：包括学生编号，学生姓名，学生的帮助对象，每个学生有且仅有一个帮助对象，帮助对象也必须是班里的学生。（表的自参照问题）。

SQL 语句及结果：



```
SQLQuery1.sql - (1...0 KB, freedom (32))
USE School
create table help(
    sid char(10) not null primary key,
    sname varchar(30),
    help_id char(10) references students(sid) on delete cascade,
    FOREIGN KEY (sid) references students(sid)
)
--
消息
命令已成功完成。
```

- (5) 学校学生会的每个部门都有一个部长，每个部长领导多个部员，每个部只有一个部员有评测部长的权利，请给出体现这两种关系（领导和评测）的两张互参照的表的定义。（两个表互相参照的问题）。

SQL 语句及结果：

模仿教师授课和课程指定教师听课关系的互相参照表的建立方式，先定义表示部员检测部长的 `members` 表，但是不定义外键属性，再定义完整的表示一个部长领导多个部员的 `monitor` 表，最后用 `alter table` 的命令定义 `members` 表的外键属性

```
USE School
create table members(
    sid char(10),
    sname varchar(30),
    myleader char(10),
    constraint PK_leader primary key(sid)
)
create table monitor(
    sid char(10),
    sname varchar(30),
    mymember char(10),
    constraint PK_monitor primary key(sid),
    constraint PF_monitor foreign key(mymember ) references members(sid)
)
alter table members
    add constraint FK_members foreign key(myleader)
        references monitor(sid)
--
消息
命令已成功完成。
```