

数据库系统实验九

18340013 陈琮昊

一、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。

二、实验内容：

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

三、实验题目：

- 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。
- 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。
- 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。
- 4.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）
- 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）

四、实验过程与结果：

为了完成本次实验，首先要按照所给 PPT 把前面创建一些表的过程执行一遍。首先执行下面几段代码：

```

create table Stu_Union
(
    sno char(5) not null unique,
    sname char(8),
    ssex char(1),
    sage int,
    sdept char(20),
    constraint PK_Stu_Union primary key(sno)
);

insert into Stu_Union values ('10001', '李明', '0', 24, 'EE');
insert into Stu_Union values ('95002', '王敏', '1', 23, 'CS');
insert into Stu_Union values ('95003', '王浩', '0', 25, 'EE');
insert into Stu_Union values ('95005', '王杰', '0', 25, 'EE');
insert into Stu_Union values ('95009', '李明', '0', 25, 'EE');
select * from Stu_Union;

```

	sno	sname	ssex	sage	sdept
1	10001	李明	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	李明	0	25	EE

```
create table Course
(
    cno char(4) not null unique,
    cname varchar(50) not null,
    cpoints int,
    constraint PK primary key (cno)
);
insert Course values ('0001', 'ComputerNetworks', 2);
insert Course values ('0002', 'Database', 3);
select * from Course
```

	cno	cname	cpoints
1	0001	ComputerNetw...	2
2	0002	Database	3

<pre>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</pre>			
	sno	cno	grade
1	10001	0001	2
2	10001	0002	2
3	95002	0001	2
4	95002	0002	2

<pre>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;</pre>			
	card_id	stu_id	remained_money
1	05212222	800005753	200.50
2	05212567	800001216	100.25

<pre>create table ICBC_Card (bank_id char(20), stu_card_id char(14) references Stu_Card(card_id), restored_money decimal (10, 2), constraint PK_ICBC_Card primary key (bank_id)) insert into ICBC_Card values ('9558844022312', '05212567', 1500.10); insert into ICBC_Card values ('9558844023645', '05212222', 5000.30); select * from ICBC_Card;</pre>			
	bank_id	stu_card_id	restored_money
1	955884402231...	05212567	15000.10
2	955884402364...	05212222	50000.30

接下来完成本次实验任务:

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。

执行代码如下:

<pre>alter table SC drop constraint FK_SC_cno_3C69FB99; alter table SC drop constraint FK_SC_sno_3B75D760; alter table SC add constraint FK_SC_cno foreign key (cno) references Course(cno) on delete no action; alter table SC add constraint FK_SC_sno foreign key (sno) references Stu_Union(sno) on delete no action; delete from Stu_Union where sno = '10001'; select * from SC</pre>	
<p>MESSAGES</p> <p>[下午11:33:47] Started executing query at Line 1</p> <p>Msg 547, Level 16, State 0, Line 7</p> <p>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'。</p> <p>语句已终止。</p> <p>(4 rows affected)</p> <p>Total execution time: 00:00:00.078</p>	

可以看到此时操作执行失败, 因为操作与外码 sno 相冲突。因为给外码设定的 on delete no action 约束, 它规定了当 从表 中有匹配的记录的时候, 主表 中相应的候选码不允许进行 update 或者 delete 的操作。在这里, Stu_union 为主表, SC 为从表, 且 SC 中有数据 (前面创建的), 且其 sno 都是 '10001', 也就是说从表中有匹配的记录, 所以操作失败。

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。

执行代码如下:

<pre>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; insert into Stu_Union values ('10001', '李萌', '0', 24, 'EE'); insert into sc values ('10001', '0001', 2); insert into sc values ('10001', '0002', 2); delete from Stu_Union where sno = '10001'; select * from SC</pre>	
<p>[下午11:37:25] Started executing query at Line 1</p> <p>Msg 1761, Level 16, State 0, Line 3</p> <p>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'。</p> <p>Msg 1750, Level 16, State 0, Line 3</p> <p>无法创建约束。请参阅前面的错误消息。</p> <p>Total execution time: 00:00:00.016</p>	

通过报错可以看到两个约束删除成功, 但是在建立新的 on delete set NULL 时却失败了。这是由于在声明表 Stu_union 和表 Course 的时候, 分别限制了 sno 和 cno NOT NULL。而这里又要将它们设置为 NULL, 显然发生了冲突, 所以无法建立约束。

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。

执行代码如下：

```
begin transaction T3
alter table ICBC_Card drop constraint FK_ICBC_Card_stu_c__44F
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;
delete from students where sid = '800001216';
commit transaction T3
```

[下午11:44:26] Started executing query at Line 1
Msg 547, Level 16, State 0, Line 5
DELETE 语句与 REFERENCE 约束 'FK_CHOICES_STUDENTS' 冲突。该冲突发生于数据库 'School'，表 'dbo.CHICES'，column 'sid'。
语句已终止。
Total execution time: 00:00:00.091

这是因为数据库中原有表 CHOCIES 使用了外码关联 STUDENTS 表，采用 on delete no action。根据第1题可以知道，主表 STUDENTS 的候选码 sid 是不可删的，因此删除 STUDENTS 内数据会出错。

4.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）

执行下面的代码：

```
1 create table table_1(
2 s_id char(8),
3 s_name varchar(20),
4 help_sid char(8) NOT NULL,
5 constraint PK_help primary key (s_id)
6 )
7 ALTER table table_1
8 add constraint FK_table_1_help foreign key(help_sid
9 references table_1[s_id])
```

[下午11:46:22] Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:00.007

可以看到外键建立成功，说明结果是正确的！

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）

执行下面的代码：

```
1 create table table_2(
2 s_id char(8),
3 s_name varchar(20),
4 leader_id char(8),
5 constraint PK_table_2 primary key (s_id)
6 )
7 create table table_3(
8 s_id char(8),
9 s_name varchar(20),
10 monitor_id char(8),
11 constraint PK_table_3 primary key (s_id),
12 constraint FK_table_3 foreign key (monitor_id)
13 references table_2(s_id)
14 )
15 ALTER table table_2
16 add constraint FK_table_2 foreign key(leader_id)
17 references table_3(s_id)
```

[下午11:50:16] Started executing query at Line 1
Total execution time: 00:00:00.006

得到结果如下，可以看到，的确是互相参照。

五、实验体会：

这次实验比较麻烦，首先要按照课件所给创建一系列的表；然后在实验时碰到了一些bug还不知道怎么解决，最后问同学才发现漏掉了一个细节，才得以解决。在这次的实验中，动手操作了违反、不违反参照完整性地插入、删除数据的过程并对其结果进行了分析，此外还实现了表的自参照和互参照。在这个过程中，体会最深的是第三个实验任务：在设置了约束 on delete no action 后，其作用是当从表中有对应匹配的记录时，禁止主表中的候选码进行更新或者删除的操作。这一点一直困扰了我很久，后来通过上网查阅资料才理解了所表达的意思。总的来说是一次相对而言耗时比较长的实验，但在这个过程中发现问题、解决问题，可以说是收获满满，感觉一下子掌握了许多知识！