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。

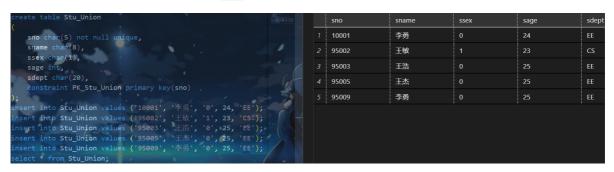
3.建立事务 T3 ,修改 ICBC_Card 表的外键属性,使其变为 on delete set NULL ,尝试删除 students 表中一条记录。观察结果,并分析原因。

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)

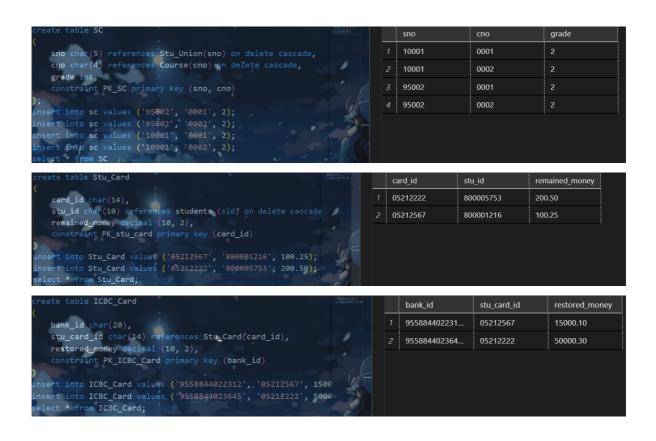
5.学校学生会的每个部门都有一个部长,每个部长领导多个部员,每个部只有一个部员有评测部长的权利,请给出体现这两种关系(领导和评测)的两张互参照的表的定义。(两个表互相参照的问题)

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 Course | ENTER NO. | | cno | cname | cpoints |
|--|-----------|---|------|--------------|---------|
| <pre>cno char(4) not null unique,</pre> | | 1 | 0001 | ComputerNetw | 2 |
| <pre>cname varchar(50) not null, cpoints int,</pre> | | 2 | 0002 | Database | 3 |
| constraint PK primary key (cno)); insert Course values ('0001', 'ComputerNetworks', 2); | | | | | |
| insert Course values ('0002', 'Database', 3); select * from Course | | | | | |



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:



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:



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:



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:



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:



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、删除数据的过程和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!