数据库完整性

实验目的

学习实体完整性的建立,以及实践违反实体完整性的结果;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。

课内实验

1. 在数据库 school中建立表Stu_Union, 进行主键约束,在没有违反实体完整性的前提下插入并更新一条记录。(参考代码如下:)

此参考代码稍微有点问题,修改如下:

```
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('10000','王敏','1',23,'cs');

UPDATE Stu_Union SET sno='' WHERE sdept='CS';
UPDATE Stu_Union SET sno='95002' WHERE sname='王敏';

select * from Stu_Union;
```

插入成功:

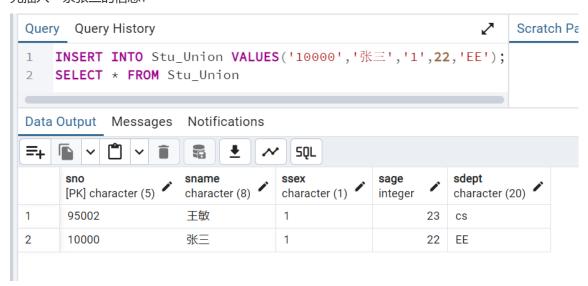


2. 演示违反实体完整性的插入操作。 (可截屏输出结果)



3. 演示违反实体完整性的更新操作。

先插入一条张三的信息:



这时候进行更新操作,发现也是报错:

4. 为演示参照完整性,建立表 Course,令 cno 为其主键,并在 Stu_Union 中插入数据。 为下面的实验步骤做预先准备。(参考代码如下:)

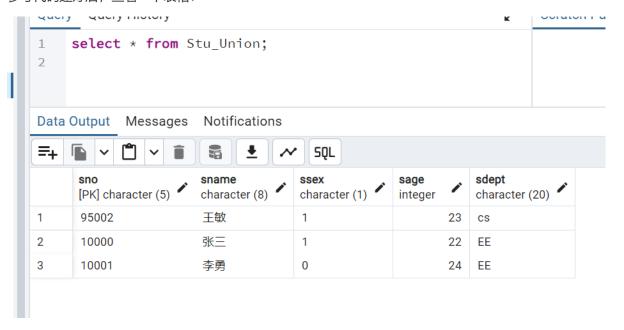
```
insert into Stu_Union values('10001','李勇','0',24,'EE');

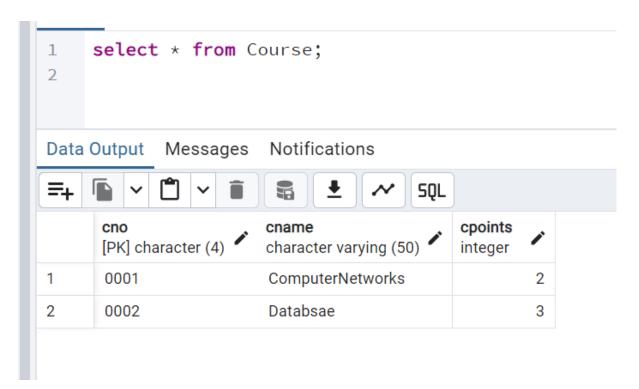
select * from Stu_Union; -- 修改一下

create table Course(
    cno char(4)NOT NULL UNIQUE,
    cname varchar(50)NOT NULL,
    cpoints int,
    constraint PK primary key(cno)
);

insert into Course values('0001','ComputerNetworks',2);
insert into Course values('0002','Databsae',3);
```

参考代码建好后, 查看一下表格:





得到创建成功

5. 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。(参考代码如下:)

```
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) -- 修改
);

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;
```

```
insert into sc values('95002','0002',2);
insert into sc values('10001','0001',2);
insert into sc values('10001','0002',2);

Select * From SC;
4
```

Data Output Messages Notifications

=+	□ ∨ □ ∨ ■ ■ ± ~ SQL			
	sno [PK] character (5)	cno [PK] character (4)	grade integer	Þ
1	95002	0001	2	2
2	95002	0002	2	2
3	10001	0001	2	2
1	10001	0002	2	2

Total rows: 4 of 4 Query complete 00:00:00.058 Ln 5, Col 16

6. 演示违反参照完整性的插入数据。



违反外键约束,因为'95003'不在Stu_Union表中

7. 在 Stu_Union 中删除数据, 演示级联删除。

```
Query Query History

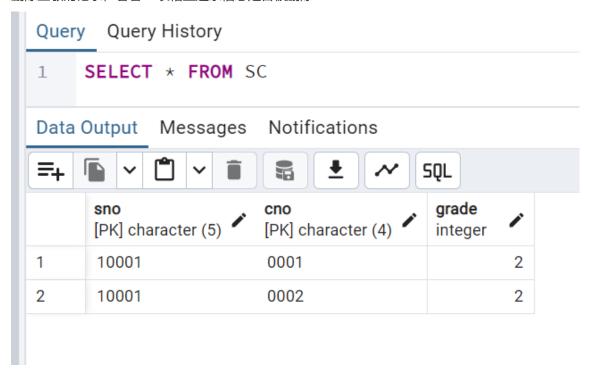
1    DELETE FROM Stu_Union WHERE sno='95002';

Data Output    Messages    Notifications

DELETE 1

Query returned successfully in 48 msec.
```

删除王敏的记录,看看SC表格里这条信息是否被删除:



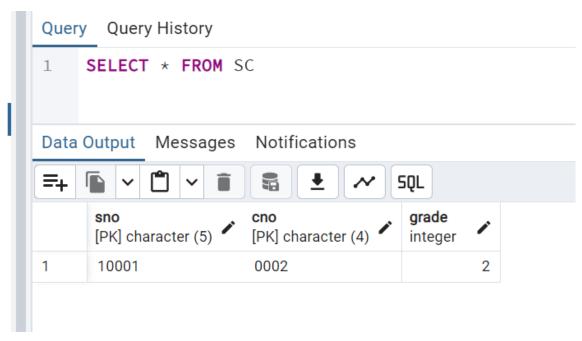
可以看到SC里对应的信息也被删除了

8. 在Course 中删除数据,演示级联删除。



Query returned successfully in 50 msec.

删除成功



可以看到SC里的信息也被删除了

自我实践(有问题部分用灰体字标出)

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。

重复课内实验中7.和8.,观察结果,分析原因。

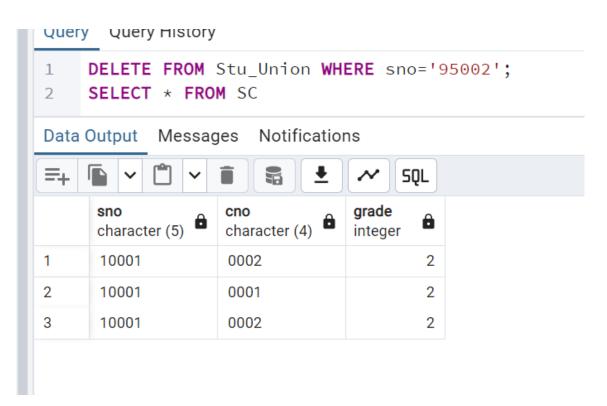
在这里,因为我直接进行约束删除,所以删除约束后能正常进行删除数据:

```
ALTER TABLE SC DROP CONSTRAINT PK_SC;

ALTER TABLE SC

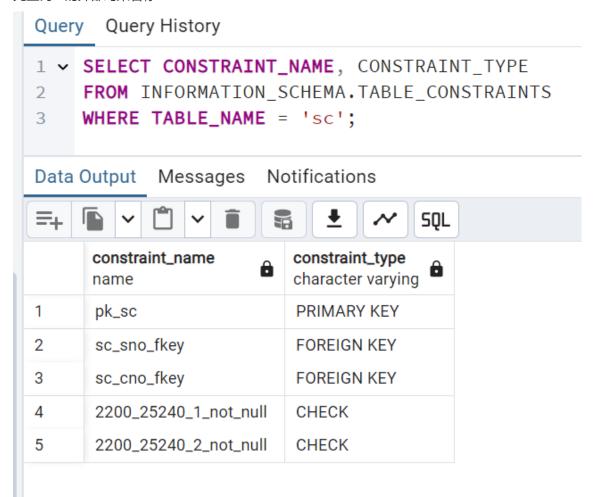
ADD CONSTRAINT FK_SC_Stu_Union FOREIGN KEY (Sno) REFERENCES Stu_Union(sno) ON DELETE NO ACTION,

ADD CONSTRAINT FK_SC_Course FOREIGN KEY (Cno) REFERENCES Course(cno) ON DELETE NO ACTION;
```



这样是不对的,因为在前面我们没有直接定义外部约束,我没有找到sc的外部约束,删错约束了, 所以导致删除没有被约束到,正确应该如下:

先查询sc的外部约束名称:



我们可以看到名称,然后进行更改操作:

```
ALTER TABLE SC DROP CONSTRAINT SC_SNO_fkey;
ALTER TABLE SC DROP CONSTRAINT SC_CNO_fkey;

ALTER TABLE SC
ADD CONSTRAINT FK_SC_Stu_Union FOREIGN KEY (Sno) REFERENCES Stu_Union(sno) ON DELETE NO ACTION,
ADD CONSTRAINT FK_SC_Course FOREIGN KEY (Cno) REFERENCES Course(cno) ON DELETE NO ACTION;
```

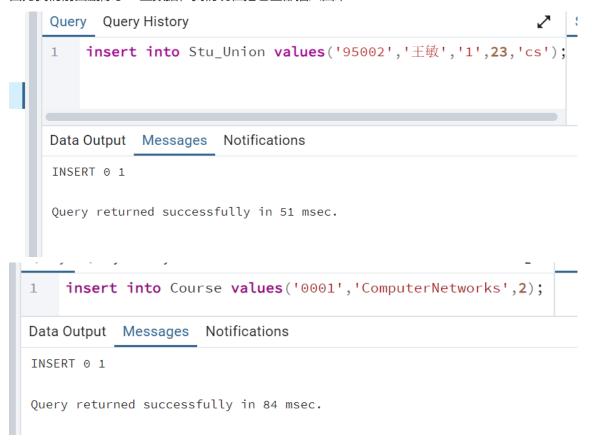
Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 60 msec.

更改成功

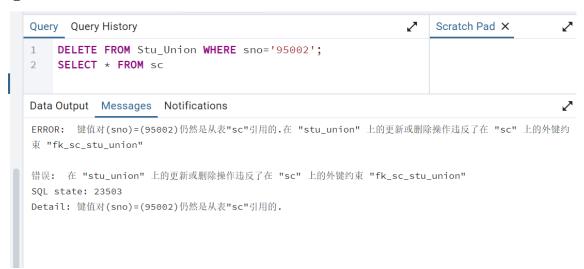
因为我们前面删除了一些数据,我们现在把它全部插入回来:



insert into sc values('95002','0001',2); 1 insert into sc values('95002','0002',2); 2 insert into sc values('10001','0001',2); 3 insert into sc values('10001','0002',2); 4 5 Select * From SC; 6 Notifications Data Output Messages SQL =+ grade sno cno **a** character (5) character (4) integer 1 10001 0002 2 2 2 95002 0001 2 3 95002 0002 2 10001 0001 5 10001 0002 2

然后复现七、八步:

七:



即可发现删除操作被约束到了

八:



同上

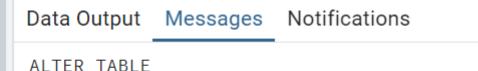
2. 使用 alter table 语句将 SC 表中的 on delete cascade 改为 on delete set NULL,重 新插入 SC 的数据。

重复课内实验中7.和8.,观察结果,分析原因。

```
ALTER TABLE SC DROP CONSTRAINT FK_SC_Stu_Union;
ALTER TABLE SC DROP CONSTRAINT FK_SC_Course;

ALTER TABLE SC
ADD CONSTRAINT FK_SC_Stu_Union FOREIGN KEY (Sno) REFERENCES Stu_Union(sno) ON DELETE SET NULL,
ADD CONSTRAINT FK_SC_Course FOREIGN KEY (Cno) REFERENCES Course(cno) ON DELETE SET NULL;
```

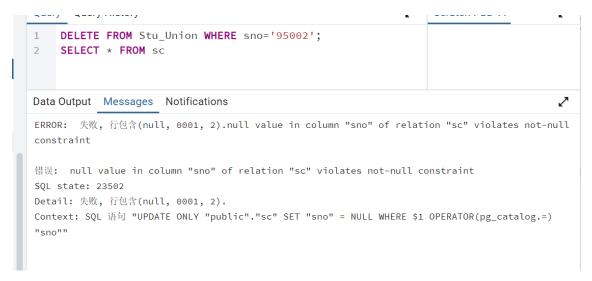
更改成功:



Query returned successfully in 53 msec.

前面没有删除数据,所以我们直接复现七、八:

七:



说明删除并将 sno 设为空违反了主键表 Stu_Union 的实体完整性。

八:

```
Data Output Messages Notifications

ERROR: 失败, 行包含(95002, null, 2).null value in column "cno" of relation "sc" violates not-null constraint

错误: null value in column "cno" of relation "sc" violates not-null constraint

SQL state: 23502
Detail: 失败, 行包含(95002, null, 2).
Context: SQL 语句 "UPDATE ONLY "public"."sc" SET "cno" = NULL WHERE $1 OPERATOR(pg_catalog.=) "cno""
```

说明删除并将 cno 设为空违反了主键表 Course 的实体完整性。

至此,实验结束。