

## 实验 8、参照完整性

### 8.1 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。

### 8.2 实验内容

- (1) 为演示参照完整性，建立表 `chenht_Course`，令 `cht_Cno` 为其主键，并在 `chenht_Stu_Union` 中插入数据。为下面的实验步骤做预先准备。
- (2) 建立表 `chenht_SC`，令 `cht_Sno` 和 `cht_Cno` 分别为参照 `chenht_Stu_Union` 表以及 `chenht_Course` 表的外键，设定为级联删除，并令(`cht_Sno`, `cht_Cno`)为其主键。在不违反参照完整性的前提下，插入数据。
- (3) 演示违反参照完整性的插入数据。
- (4) 在 `chenht_Stu_Union` 中删除数据，演示级联删除。
- (5) 在 `chenht_Course` 中删除数据，演示级联删除。
- (6) 为了演示多重级联删除，建立 `chenht_Stu_Card` 表，令 `cht_Stu_id` 为参数 `chenht_Stu_Union` 表的外键，令 `cht_Card_id` 为其主键，并插入数据。
- (7) 为了演示多重级联删除，建立 `chenht_ICBC_Card` 表，令 `cht_Stu_card_id` 为参数 `chenht_Stu_Union` 表的外键，令 `cht_Card_id` 为其主键，并插入数据。
- (8) 通过删除 `chenht_Stu_Union` 表中的一条记录，演示三个表的多重级联删除。
- (9) 演示事务中进行多重级联删除失败的处理。修改 `chenht_ICBC_Card` 表的外键属性，使其变为 `On delete No action`，演示事务中通过删除 `chenht_Stu_Union` 表中的一条记录，多重级联删除失败，整个事务回滚到事务的初始状态。
- (10) 演示互参考问题及其解决方法。要建立教师授课和课程指定教师听课关系的两张表，规定一个教师可以授多门课，但是每个课程只能指定

一个教师去听课，所以要为两张表建立相互之间的参照关系。

### 8.3 实验步骤

以系统管理员或 sa 账号登录到 SSMS，在新建查询窗口中输入如下命令，运行并观察和记录结果。

- (1) [初始化 chenht\_Stu\_Union 和 chenht\_Course 表]在新建查询窗口中输入如下 SQL 语句：

```
USE chenht_University_Mis
INSERT chenht_Stu_Union Values('S01','李用','0',24,'FF')
SELECT * FROM chenht_Stu_Union;
CREATE TABLE chenht_Course(
    cht_Cno CHAR(4) NOT NULL UNIQUE,
    cht_Cname VARCHAR(50) NOT NULL,
    cht_Cpoints INT,
    CONSTRAINT cht_PK PRIMARY KEY(cht_Cno));
INSERT cht_Course VALUES('C01','ComputerNetworks',2);
INSERT Course VALUES('C02','ArtificialIntelligence',3);
```

- (2) [初始化 chenht\_SC 表]在新建查询窗口中输入如下 SQL 语句：

```
USE chenht_University_Mis
CREATE Table chenht_SC(
    cht_Sno CHAR(8),
    cht_Cno CHAR(4),
    cht_Scredit INT,
    CONSTRAINT cht_PK_SC PRIMARY KEY(cht_Sno,cht_Cno),
    CONSTRAINT cht_FK_SC_Sno FOREIGN KEY(cht_Sno) REFERENCES
chenht_Stu_Union (cht_Sno) ON DELETE CASCADE,
    CONSTRAINT cht_FK_SC_Cno FOREIGN KEY(cht_Cno) REFERENCES
chenht_Course (cht_Cno) ON DELETE CASCADE
);
INSERT INTO chenht_SC VALUES('S02','C01',2);
INSERT INTO chenht_SC VALUES ('S02','C02',2);
INSERT INTO chenht_SC VALUES ('S01','C01',2);
INSERT INTO chenht_SC VALUES ('S01','C02',2);
SELECT * FROM chenht_SC;
```

- (3) [在 chenht\_SC 插入一条数据，验证参照完整性]在新建查询窗口中输入如下 SQL 语句：

```
USE chenht_University_Mis
INSERT INTO chenht_SC values('S99','C99',2)
```

- (4) [在 chenht\_Stu\_Union 删除一条数据，验证级联删除]在新建查询窗口中输入如下 SQL 语句：

```
USE chenht_University_Mis
```

```
DELETE FROM chenht_Stu_Union WHERE cht_Sno='S01';
```

```
SELECT * FROM chenht_SC;
```

- (5) [在 chenht\_Course 删除一条数据, 验证级联删除]在新建查询窗口中输入如下 SQL 语句:

```
USE chenht_University_Mis
```

```
DELETE FROM chenht_Course WHERE cht_Cno='C02';
```

```
SELECT * FROM chenht_SC;
```

- (6) [初始化 cht\_Stu\_Card 表]在新建查询窗口中输入如下 SQL 语句:

```
/* 在这之前, 将 chenht_Students 和 chenht_Reports 中 cht_Sno 的类型更改为  
cht_Sno, 以匹配外键 */
```

```
USE chenht_University_Mis
```

```
CREATE TABLE chenht_Stu_Card(
```

```
    cht_Card_id CHAR(14),
```

```
    cht_Sno CHAR(8),
```

```
    cht_Remained_money DECIMAL(10,2),
```

```
    Constraint cht_PK_Stu_Card PRIMARY KEY(cht_Card_id),
```

```
    Constraint cht_FK_Stu_Card_Sno FOREIGN KEY(cht_Sno) REFERENCES che  
nht_Students(cht_Sno) ON DELETE CASCADE
```

```
)
```

```
INSERT INTO chenht_Stu_Card VALUES('05212567','S03',400.25);
```

```
INSERT INTO chenht_Stu_Card VALUES('05212222','S09',600.50);
```

```
SELECT * FROM chenht_Stu_card;
```

- (7) [初始化 cht\_ICBC\_Card 表]在新建查询窗口中输入如下 SQL 语句:

```
USE chenht_University_Mis
```

```
CREATE TABLE chenht_ICBC_Card(
```

```
    cht_Bank_id CHAR(20),
```

```
    cht_Stu_card_id CHAR(14),
```

```
    cht_Restored_money DECIMAL(10,2),
```

```
    constraint cht_PK_ICBC_Card PRIMARY KEY(cht_Bank_id),
```

```
    constraint cht_FK_ICBC_Card_Stu_id FOREIGN KEY(cht_Stu_card_id) RE  
FERENCES chenht_Stu_card(cht_Card_id) ON DELETE CASCADE
```

```
)
```

```
INSERT INTO chenht_ICBC_Card VALUES('9558844022312','05212567',15000.1  
);
```

```
INSERT INTO chenht_ICBC_Card VALUES('9558844023645','05212222',50000.3  
);
```

```
SELECT * FROM chenht_ICBC_Card;
```

- (8) [chenht\_Report 外键添加级联删除, 验证级联删除]在新建查询窗口中输入如下 SQL 语句:

```
USE chenht_University_Mis
```

```
ALTER TABLE chenht_Reports DROP [cht_FK_Reports_Students];
```

```
ALTER TABLE chenht_Reports ADD
```

```
CONSTRAINT [cht_FK_Reports_Students] FOREIGN KEY
```

```
(
```

```
    [cht_Sno]
```

```

    ) REFERENCES [dbo].[chenht_Students] (
        [cht_Sno]
    ) ON DELETE CASCADE;
DELETE FROM chenht_Students WHERE cht_Sno='S03';
SELECT * FROM chenht_Stu_card;
SELECT * FROM chenht_ICBC_Card;

```

(9) [chenht\_ICBC\_Card 修改外键为不采取操作, 并进行验证]在新建查询窗口中输入如下 SQL 语句:

```

ALTER TABLE chenht_ICBC_Card
DROP CONSTRAINT cht_FK_ICBC_Card_Stu_id;
ALTER TABLE chenht_ICBC_Card
ADD CONSTRAINT cht_FK_ICBC_Card_Stu_id FOREIGN KEY (cht_Stu_card_id)
REFERENCES chenht_Stu_card(cht_Card_id) ON DELETE NO ACTION;

```

/\*\*在新建查询窗口中输入如下 SQL 语句: \*\*/

```

Begin Transaction Del
DELETE FROM chenht_Stu_Card WHERE cht_Card_id = '05212222';
SELECT * FROM chenht_Stu_card;
SELECT * FROM chenht_ICBC_card;
Commit Transaction Del

```

/\*\*在新建查询窗口中输入如下 SQL 语句: \*\*/

```

USE University_Mis
SELECT * FROM chenht_Stu_card;
SELECT * FROM chenht_ICBC_card;

```

#10-12: 演示互参考问题及解决方法。

(10)在新建查询窗口中输入如下 SQL 语句:

```

USE chenht_University_Mis
CREATE TABLE chenht_Listen_course(
    cht_Tno CHAR(6),
    cht_Tname VARCHAR(20),
    cht_Cno CHAR(4)
    CONSTRAINT cht_PK_listen_course PRIMARY KEY(cht_Tno)
    CONSTRAINT cht_FK_listen_course FOREIGN KEY(cht_Cno)
    REFERENCES chenht_Teach_course(cht_Cno)
)
CREATE TABLE chenht_Teach_course(
    cht_Cno CHAR(4),
    cht_Cname VARCHAR(30),
    cht_Tno CHAR(6)
    CONSTRAINT cht_PK_Teach_course PRIMARY KEY(cht_Cno)
    CONSTRAINT cht_FK_Teach_course FOREIGN KEY(cht_Tno)
    REFERENCES chenht_Listen_course(cht_Tno)
)

```

(11) 在新建查询窗口中输入如下 SQL 语句:

```

USE chenht_University_Mis
CREATE TABLE chenht_Listen_course(

```

```

cht_Tno CHAR(6),
cht_Tname VARCHAR(20),
cht_Cno CHAR(4)
CONSTRAINT cht_PK_listen_Course PRIMARY KEY(cht_Tno)
)

```

(12) 在新建查询窗口中输入如下 SQL 语句:

```

USE chenht_University_Mis
CREATE TABLE chenht_Teach_course(
    cht_Cno CHAR(4),
    cht_Cname VARCHAR(30),
    cht_Tno CHAR(6)
    CONSTRAINT cht_PK_teach_course PRIMARY KEY(cht_Cno)
    CONSTRAINT cht_FK_teach_course FOREIGN KEY(cht_Tno)
    REFERENCES chenht_Listen_course(cht_Tno)
)
ALTER TABLE chenht_Listen_course
    ADD CONSTRAINT cht_FK_listen_course FOREIGN KEY(cht_Cno)
    REFERENCES chenht_Teach_course(cht_Cno);

```

#### 8.4 实验结果

3) 演示违反参照完整性的插入数据。

SQL 语句:

```

USE chenht_University_Mis
INSERT INTO chenht_SC values('S99','C99',2)

```

实验结果:

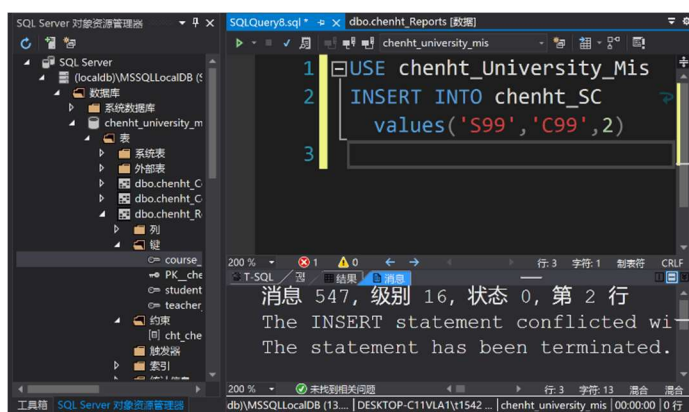


图 8-1 实验结果截图(1)

4) 在 chenht\_Stu\_Union 中删除数据, 演示级联删除。

SQL 语句:

```

USE chenht_University_Mis
DELETE FROM chenht_Stu_Union WHERE cht_Sno='S01';
SELECT * FROM chenht_SC;

```

实验结果:

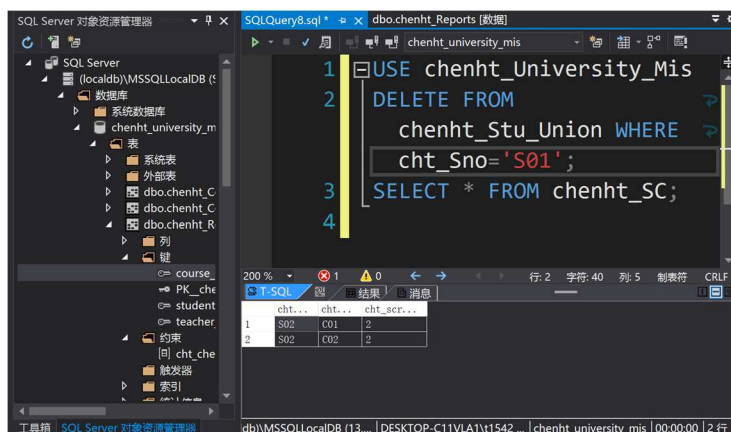


图 8-2 实验结果截图(2)

7) 为了演示多重级联删除, 建立 ICBC\_Card 表, 令 Stu\_card\_id 为参数 chenht\_Stu\_Union n 表的外键, 令 Card\_id 为其主键, 并插入数据。

SQL 语句:

```
USE chenht_University_Mis
CREATE TABLE chenht_ICBC_Card(
    cht_Bank_id CHAR(20),
    cht_Stu_card_id CHAR(14),
    cht_Restored_money DECIMAL(10,2),
    constraint cht_PK_ICBC_Card PRIMARY KEY(cht_Bank_id),
    constraint cht_FK_ICBC_Card_Stu_id FOREIGN KEY(cht_Stu_card_id) REFERENCES chenht_Stu_card(cht_Card_id) ON DELETE CASCADE
)
INSERT INTO chenht_ICBC_Card VALUES('9558844022312','05212567',15000.1);
INSERT INTO chenht_ICBC_Card VALUES('9558844023645','05212222',50000.3);
SELECT * FROM chenht_ICBC_Card;
```

实验结果:

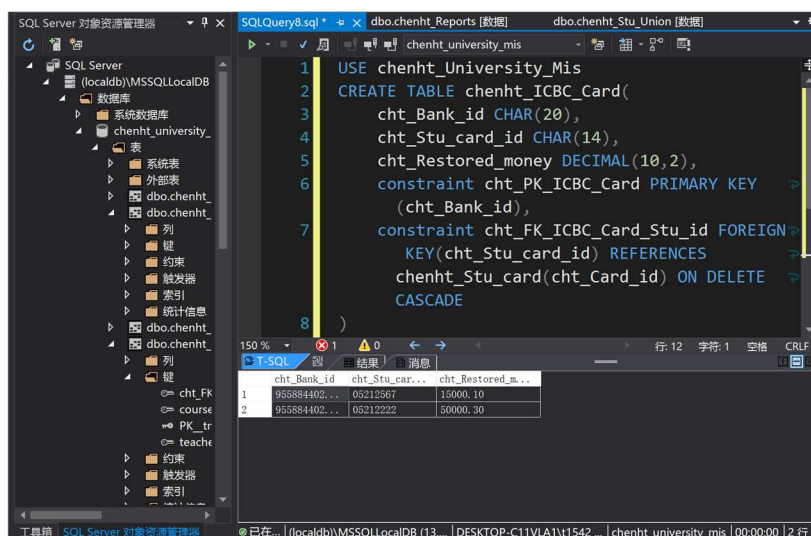


图 8-3 实验结果截图(3)

9) 演示事务中进行多重级联删除失败的处理。修改 ICBC\_Card 表的外键属

性，使其变为 On delete No action，演示事务中通过删除 chenht\_Stu\_Union 表中的一条记录，多重级联删除失败，整个事务回滚到事务的初始状态。

SQL 语句：

```
ALTER TABLE chenht_ICBC_Card
DROP CONSTRAINT cht_FK_ICBC_Card_Stu_id;
ALTER TABLE chenht_ICBC_Card
ADD CONSTRAINT cht_FK_ICBC_Card_Stu_id FOREIGN KEY (cht_Stu_card_id)
REFERENCES chenht_Stu_card(cht_Card_id) ON DELETE NO ACTION;

/**在新建查询窗口中输入如下 SQL 语句：**/
Begin Transaction Del
DELETE FROM chenht_Stu_Card WHERE cht_Card_id = '05212222';
SELECT * FROM chenht_Stu_card;
SELECT * FROM chenht_ICBC_card;
Commit Transaction Del

/**在新建查询窗口中输入如下 SQL 语句：**/
USE chenht_University_Mis
SELECT * FROM chenht_Stu_card;
SELECT * FROM chenht_ICBC_card;
```

实验结果：

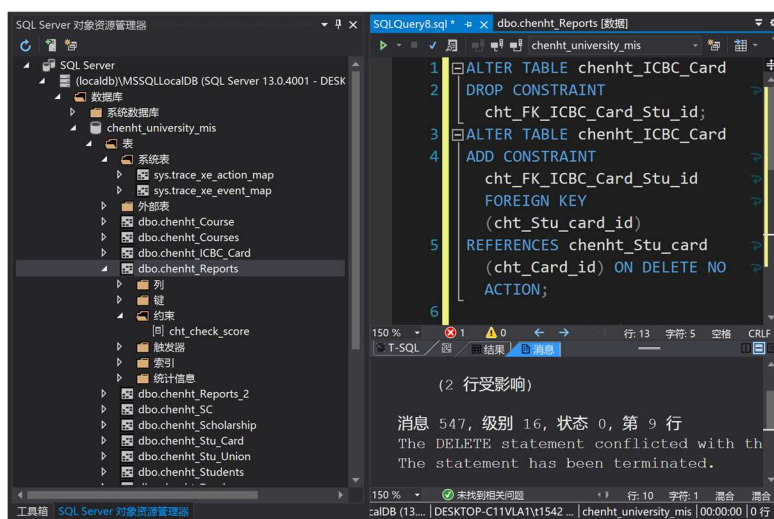


图 8-4 实验结果截图(4)

#### 10) 建立互参考模型（错误示范）

SQL 语句：

```
USE chenht_University_Mis
CREATE TABLE chenht_Listen_course(
    cht_Tno CHAR(6),
    cht_Tname VARCHAR(20),
    cht_Cno CHAR(4)
    CONSTRAINT cht_PK_listen_course PRIMARY KEY(cht_Tno)
    CONSTRAINT cht_FK_listen_course FOREIGN KEY(cht_Cno)
```



```

REFERENCES chenht_Teach_course(cht_Cno)
)
CREATE TABLE chenht_Teach_course(
    cht_Cno CHAR(4),
    cht_Cname VARCHAR(30),
    cht_Tno CHAR(6)
    CONSTRAINT cht_PK_Teach_course PRIMARY KEY(cht_Cno)
    CONSTRAINT cht_FK_Teach_course FOREIGN KEY(cht_Tno)
    REFERENCES chenht_Listen_course(cht_Tno)
)

```

实验结果:

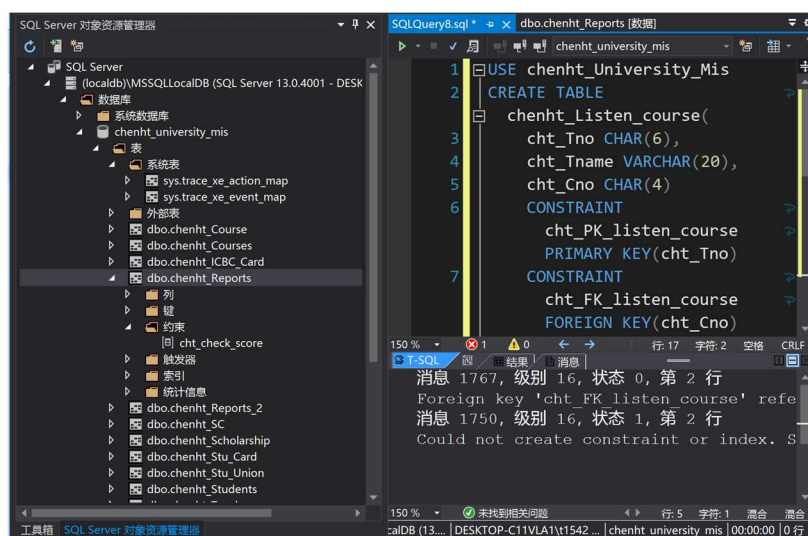


图 8-5 实验结果截图(5)

## 11-12) 建立互参考模型（正确示范）

SQL 语句:

```

USE chenht_University_Mis
CREATE TABLE chenht_Listen_course(
    cht_Tno CHAR(6),
    cht_Tname VARCHAR(20),
    cht_Cno CHAR(4)
    CONSTRAINT cht_PK_listen_Course PRIMARY KEY(cht_Tno)
)

USE chenht_University_Mis
CREATE TABLE chenht_Teach_course(
    cht_Cno CHAR(4),
    cht_Cname VARCHAR(30),
    cht_Tno CHAR(6)
    CONSTRAINT cht_PK_teach_course PRIMARY KEY(cht_Cno)
    CONSTRAINT cht_FK_teach_course FOREIGN KEY(cht_Tno)
    REFERENCES chenht_Listen_course(cht_Tno)
)

ALTER TABLE chenht_Listen_course
    ADD CONSTRAINT cht_FK_listen_course FOREIGN KEY(cht_Cno)

```



REFERENCES chenht\_Teach\_course(cht\_Cno);

实验结果:

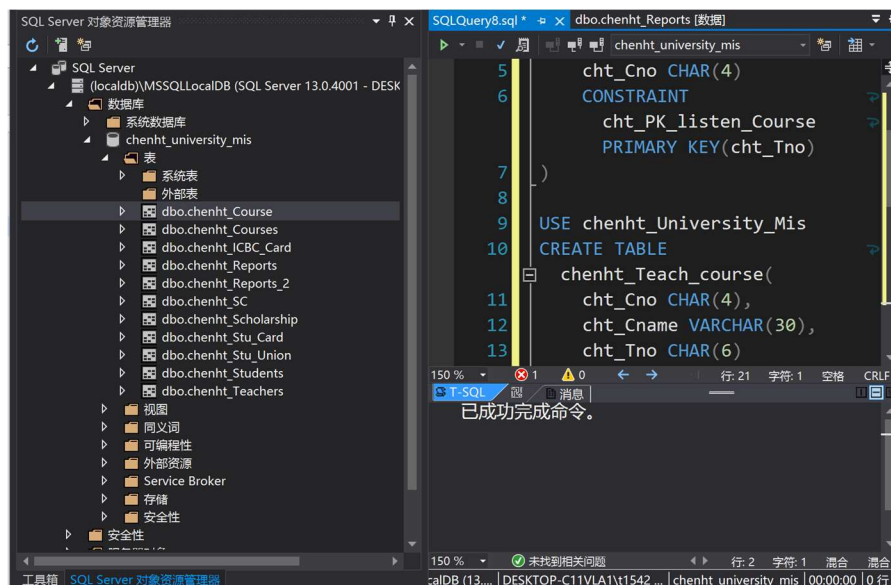


图 8-6 实验结果截图(6)

## 8.5 实验体会

### 1) 实验反思

- a (参照完整性数据一致性) 由于参照完整性是使用“参照来实现一致性的”，因此在设计数据库时连个参照的对象所对应的属性的类型应当保持一致，否则将无法建立外键（参照完整性）。
- b (ALTER) 在更改数据模式(ALTER)时，更改后的数据模式也应当保持原来数据的参照关系。例如在之前的表中 chenht\_Reports 参照 chenht\_Students 以 cht\_sno 字段。如果需要修改 cht\_sno 的类型。应该先删除外键，更改两个字段类型，重新建立外键这几个步骤。直接更改数据类型无法正常执行。
- c (级联更新和级联删除) 为保持数据的一致性，一般在外键参照关系中使用级联更新和级联删除，级联更新和级联删除具有传递性。
- d (互参照) 互参照关系应当先建立表，之后使用 ALTER 命令再建立外键。否则会有找不到表的问题。

### 2) 实验收获

通过该实验，对数据库参照的完整性以及外键的功能和作用，数据库的数据完整性保护有了一定的理解。能够设计具有参照完整性的一个数据库。