

# Lecture 10 数据库系统实验

Yubao Liu (刘玉葆)

School of Data and Computer Science

Sun Yat-sen University

- 本节课提纲

- 实验目的
- 实验内容
- 实验示例
- 练习

- 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。

- 实验内容
  - 不违反参照完整性的插入数据示例
  - 违反参照完整性的插入数据示例
  - 级联删除
  - 两张表的互相参照问题

- 实验示例

以 school数据库为例，在该数据库中存在4张表格，分别为

- students (sid, sname , email , grade)
- teachers (tid, tname , email , salary )
- courses (cid, cname , hour )
- choices (no, sid , tid , cid , score )

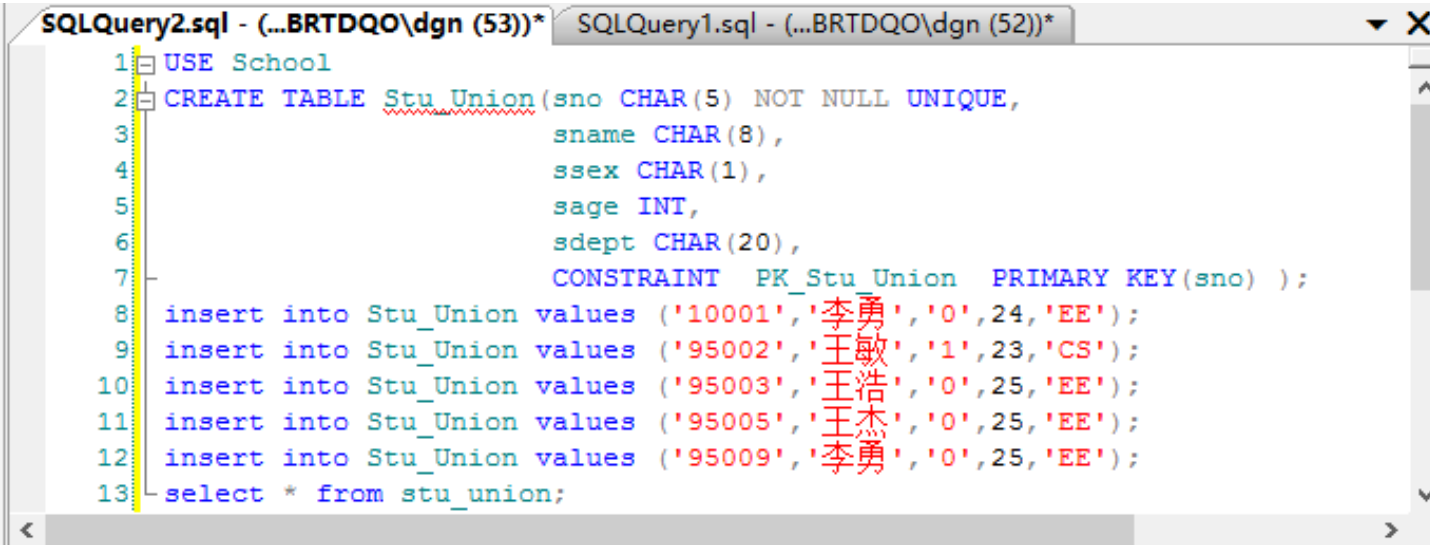
在数据库中，存在这样的关系，学生可以选择课程。一个课程对应一个教师。

在CHOICES表中保存学生的选课记录。

# • 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。建立表Course,令cno为主键。



The screenshot shows a SQL query editor with two tabs: 'SQLQuery2.sql - (...BRTDQO\dgn (53))\*' and 'SQLQuery1.sql - (...BRTDQO\dgn (52))\*'. The active tab contains the following SQL code:

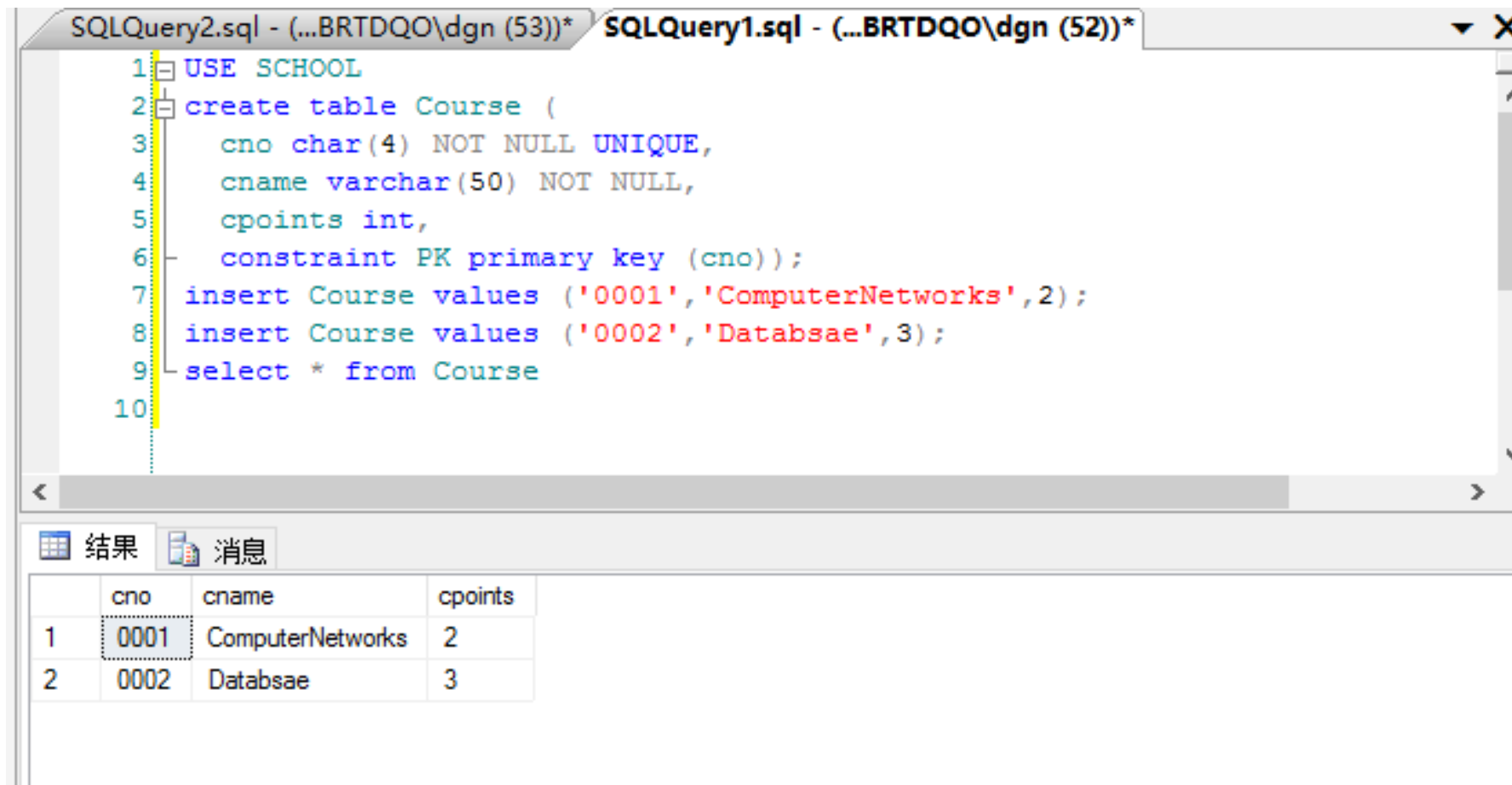
```
1 USE School
2 CREATE TABLE Stu_Union(sno CHAR(5) NOT NULL UNIQUE,
3                           sname CHAR(8),
4                           ssex CHAR(1),
5                           sage INT,
6                           sdept CHAR(20),
7                           CONSTRAINT PK_Stu_Union PRIMARY KEY(sno) );
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;
```

Below the query editor, there is a '结果' (Results) tab and a '消息' (Messages) tab. The '结果' tab is active, displaying the following table:

|   | 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    |

# • 实验准备

2)在数据库 school中建立表Course,令cno为主键。



```
SQLQuery2.sql - (...BRTDQO\dgn (53))*  SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 USE SCHOOL
2 create table Course (
3     cno char(4) NOT NULL UNIQUE,
4     cname varchar(50) NOT NULL,
5     cpoints int,
6     constraint PK primary key (cno));
7 insert Course values ('0001','ComputerNetworks',2);
8 insert Course values ('0002','Databsae',3);
9 select * from Course
10
```

|   | cno  | cname            | cpoints |
|---|------|------------------|---------|
| 1 | 0001 | ComputerNetworks | 2       |
| 2 | 0002 | Databsae         | 3       |

# 实验示例

1.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。

```
1 USE SCHOOL
2 CREATE TABLE SC(
3   sno CHAR(5) REFERENCES Stu_Union (sno) on delete cascade,
4   cno CHAR(4) REFERENCES Course(cno) on delete cascade,
5   grade INT,
6   CONSTRAINT PK_SC PRIMARY KEY (sno,cno)
7 );
8 insert into sc values ('95002','0001',2);
9 insert into sc values ('95002','0002',2);
10 insert into sc values ('10001','0001',2);
11 insert into sc values ('10001','0002',2);
12 Select * From SC;
13
```

设置级联删除

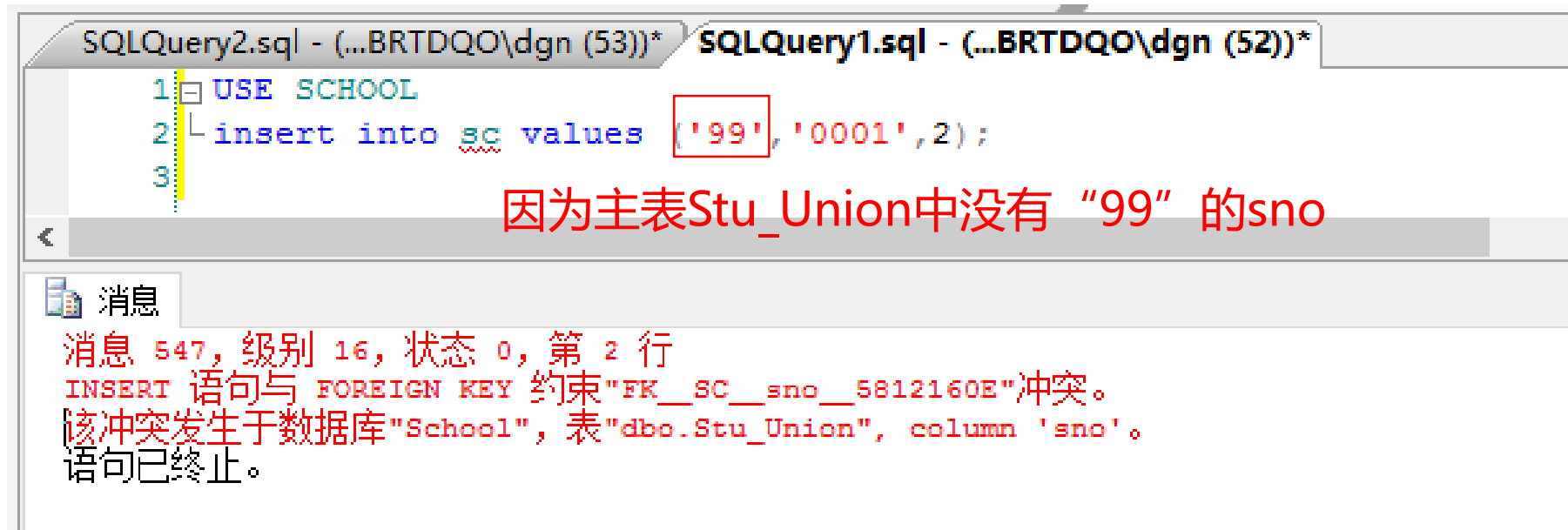
由于（sno，cno）联合作为主键，即使sno重复，但cno不重复，所以插入数据依然成功。

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



# • 实验示例

## 2.演示违反参照完整性的插入数据。



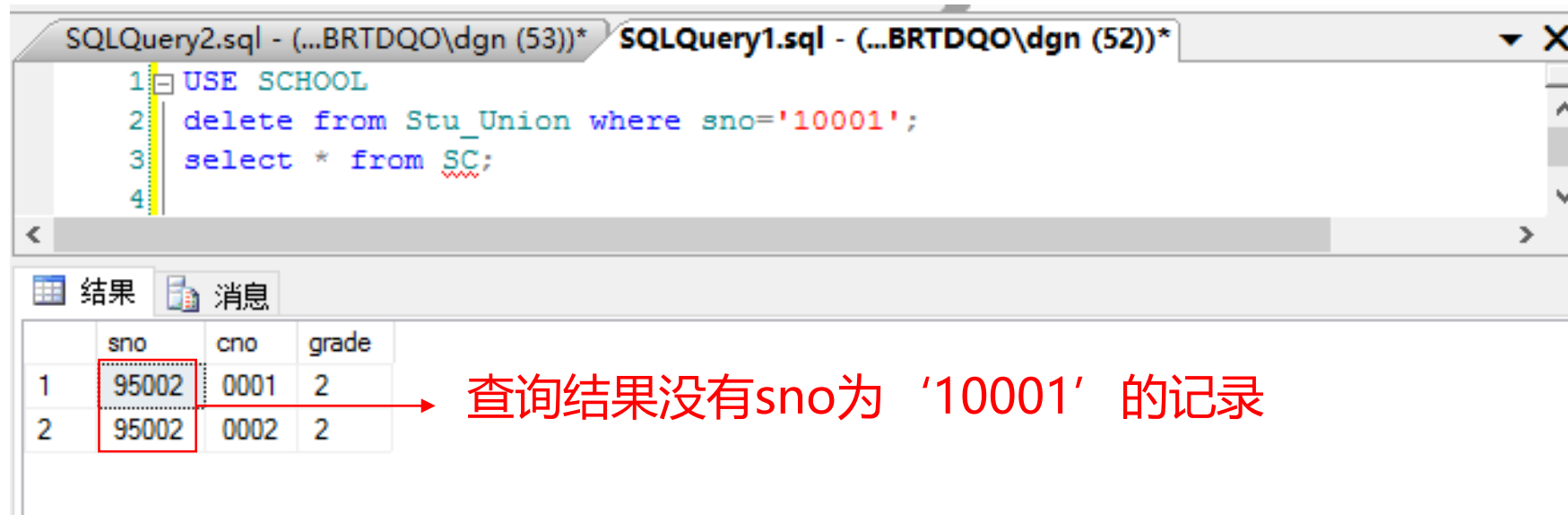
```
SQLQuery2.sql - (...BRTDQO\dgn (53))*  SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 USE SCHOOL
2 insert into sc values ('99','0001',2);
3
```

因为主表Stu\_Union中没有 “99” 的sno

消息 547, 级别 16, 状态 0, 第 2 行  
INSERT 语句与 FOREIGN KEY 约束“FK\_\_SC\_\_sno\_\_5812160E”冲突。  
该冲突发生于数据库“School”, 表“dbo.Stu\_Union”, column 'sno'。  
语句已终止。

# • 实验示例

3.在主表Stu\_Union中删除数据，演示级联删除。



The screenshot shows a SQL query window with two tabs. The active tab, 'SQLQuery1.sql - (...BRTDQO\dgn (52))\*', contains the following SQL code:

```
1 USE SCHOOL
2 delete from Stu_Union where sno='10001';
3 select * from SC;
4
```

Below the query window, the '结果' (Results) tab is selected, displaying a table with the following data:

|   | sno   | cno  | grade |
|---|-------|------|-------|
| 1 | 95002 | 0001 | 2     |
| 2 | 95002 | 0002 | 2     |

A red arrow points from the text '查询结果没有sno为 '10001' 的记录' (Query result does not have records with sno '10001') to the table, indicating that the delete operation was successful and no records with sno '10001' are present in the result set.

由于on delete cascade的连带删除作用，在主表Stu\_Union中删除某个学号，那么从表SC中对应这个学号为外键的所有记录也会跟着删除。

## • 实验示例

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Students表中一条记录，演示三个表的多重级联删除。

1)建立Stu\_Card表:

The screenshot shows a SQL query editor with two tabs: 'SQLQuery2.sql - (...BRTDQO\dgn (53))\*' and 'SQLQuery1.sql - (...BRTDQO\dgn (52))\*'. The active tab 'SQLQuery1.sql' contains the following SQL code:

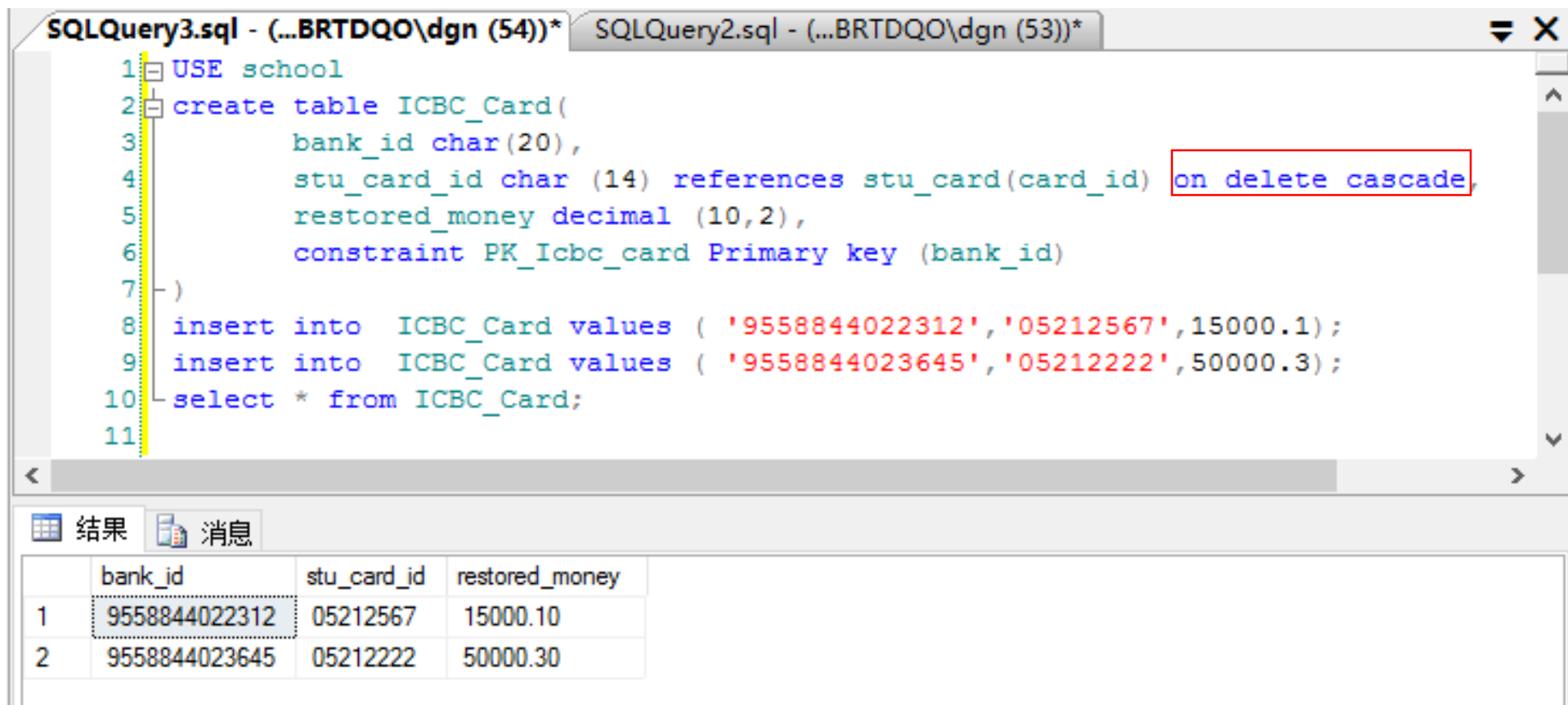
```
1 USE school
2 create table Stu_Card(
3     card_id char(14),
4     stu_id char(10) references students(sid) on delete cascade,
5     remained_money decimal(10,2),
6     constraint PK_stu_card Primary key (card_id)
7 )
8 insert into Stu_Card values ('05212567','800001216',100.25);
9 insert into Stu_Card values ('05212222','800005753',200.50);
10 select * from Stu_card;
```

The 'on delete cascade' text in line 4 is highlighted with a red rectangle. Below the query editor, there is a '结果' (Results) tab and a '消息' (Messages) tab. The '结果' tab is active, displaying a table with the following data:

|   | card_id  | stu_id    | remained_money |
|---|----------|-----------|----------------|
| 1 | 05212222 | 800005753 | 200.50         |
| 2 | 05212567 | 800001216 | 100.25         |

# 实验示例

## 2)建立表ICBC\_Card表:



The screenshot shows a SQL query editor with two tabs: 'SQLQuery3.sql - (...BRTDQO\dgn (54))\*' and 'SQLQuery2.sql - (...BRTDQO\dgn (53))\*'. The active tab contains the following SQL code:

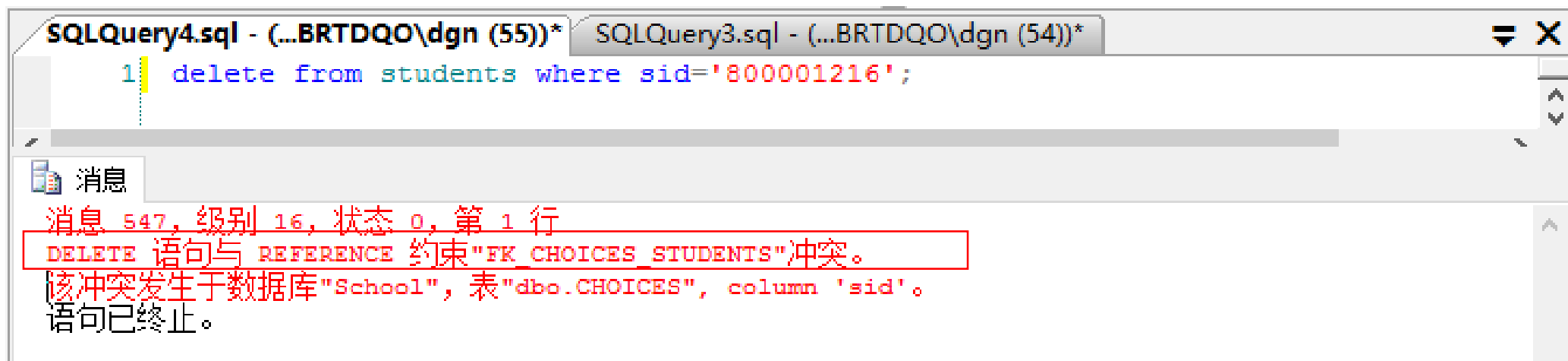
```
1 USE school
2 create table ICBC_Card(
3     bank_id char(20),
4     stu_card_id char (14) references stu_card(card_id) on delete cascade,
5     restored_money decimal (10,2),
6     constraint PK_Icbc_card Primary key (bank_id)
7 )
8 insert into ICBC_Card values ( '9558844022312', '05212567', 15000.1);
9 insert into ICBC_Card values ( '9558844023645', '05212222', 50000.3);
10 select * from ICBC_Card;
11
```

The text 'on delete cascade,' is highlighted with a red box. Below the query editor, there is a section with two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with the following data:

|   | bank_id       | stu_card_id | restored_money |
|---|---------------|-------------|----------------|
| 1 | 9558844022312 | 05212567    | 15000.10       |
| 2 | 9558844023645 | 05212222    | 50000.30       |

# • 实验示例

3)通过删除Students表中一条记录，演示三个表的多重级联删除。



删除数据出错原因：由于数据库中原有表Choices使用了外键关联Students表，采用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）。所以直接在students中删除数据会出错。

# 实验示例

要演示多重级联删除，必须去除表choices原有约束，然后建立新的外键约束，并将其外键设置为级联删除。操作如下：

SQLQuery4.sql - (...BRTDQO\dgn (55))\*

SQLQuery3.sql - (...BRTDQO\dgn (54))\*

SQLQuery2.sql - (...BRTDQO\dgn (53))\*

```
1 USE school
2 alter table choices drop [FK_CHOICES_STUDENTS];
3 alter table choices add
4 CONSTRAINT [FK_CHOICES_STUDENTS] FOREIGN KEY
5 (
6     [sid]
7 ) REFERENCES [dbo].[STUDENTS] (
8     [sid]
9 ) on delete cascade;
10 delete from students where sid='800001216';
11 select * from stu_card;
12 select * from icbc_card;
```

结果

消息

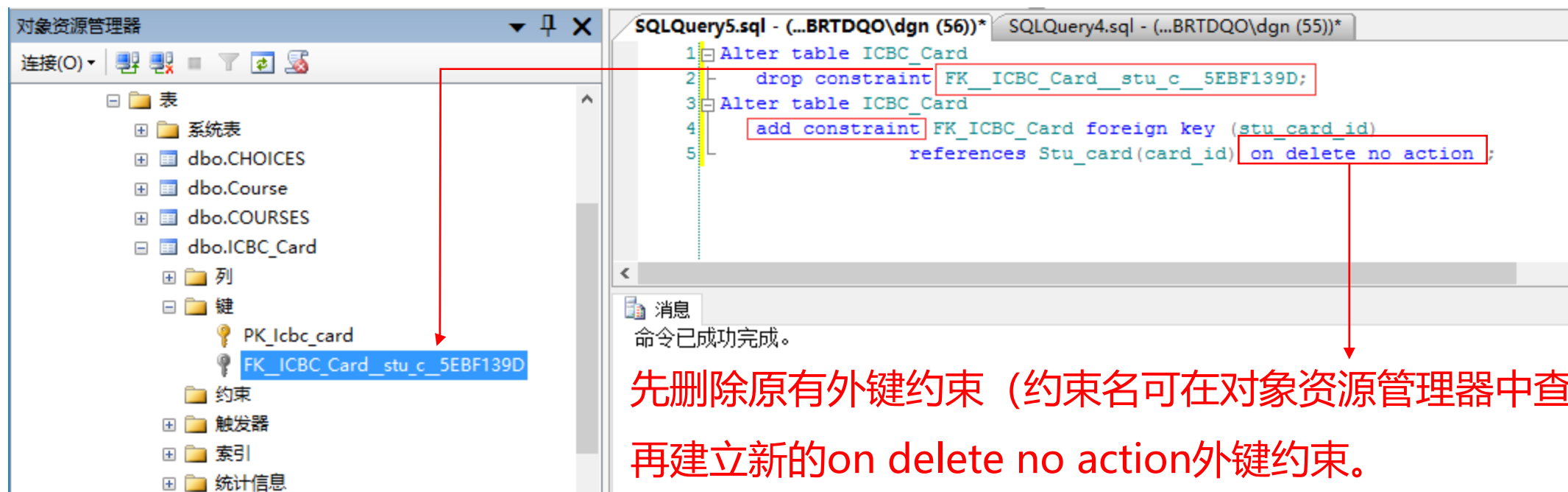
|   | card_id  | stu_id    | remained_money |
|---|----------|-----------|----------------|
| 1 | 05212222 | 800005753 | 200.50         |

|   | bank_id       | stu_card_id | restored_money |
|---|---------------|-------------|----------------|
| 1 | 9558844023645 | 05212222    | 50000.30       |

表stu\_card和表icbc\_card中的数据已级联删除

# 实验示例

5.演示事务中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students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。

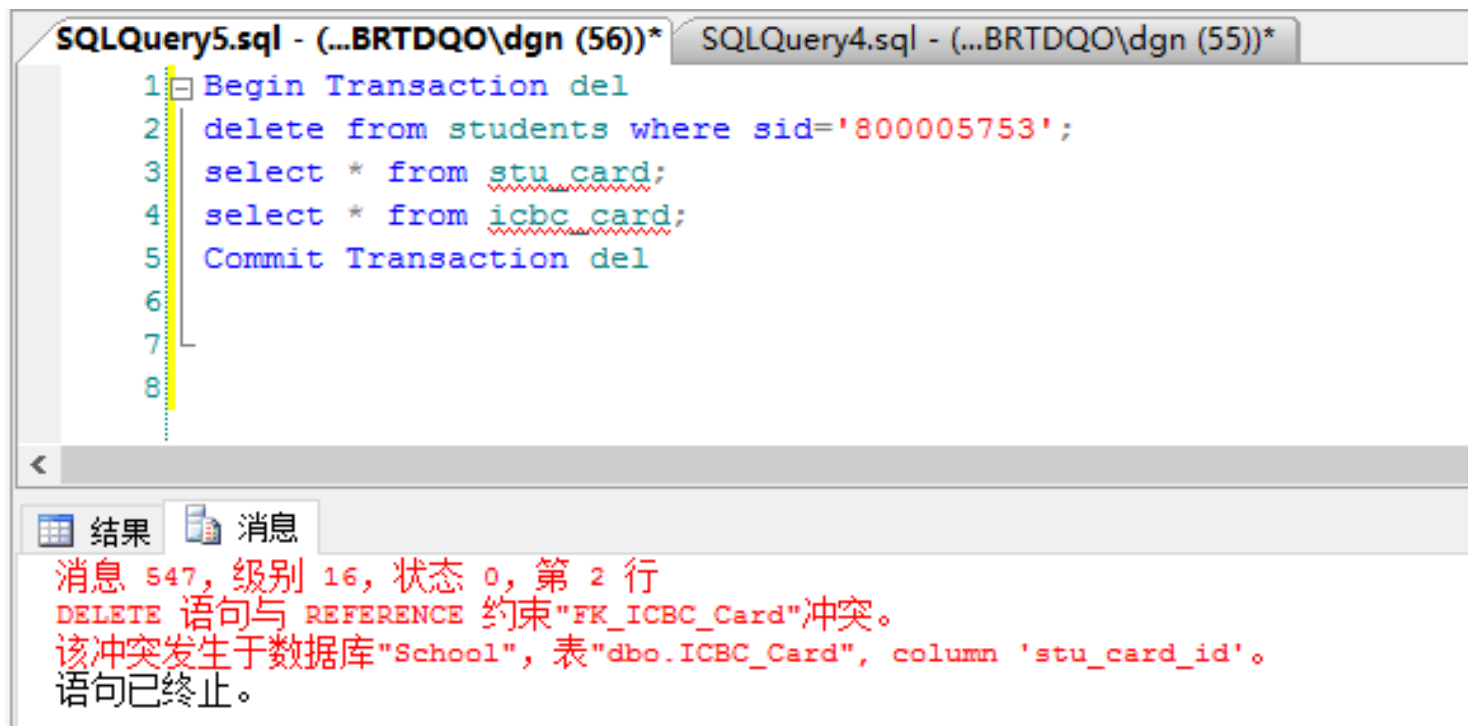


先删除原有外键约束（约束名可在对象资源管理器中查看），再建立新的on delete no action外键约束。

（ on delete no action意味着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）

# • 实验示例

建立事务del:



The screenshot shows a SQL query editor with two tabs: 'SQLQuery5.sql - (...BRTDQO\dgn (56))\*' and 'SQLQuery4.sql - (...BRTDQO\dgn (55))\*'. The active tab contains the following SQL code:

```
1 Begin Transaction del
2 delete from students where sid='800005753';
3 select * from stu_card;
4 select * from icbc_card;
5 Commit Transaction del
6
7
8
```

Below the code editor, there is a '消息' (Messages) pane showing an error message:

消息 547, 级别 16, 状态 0, 第 2 行  
DELETE 语句与 REFERENCE 约束"FK\_ICBC\_Card"冲突。  
该冲突发生于数据库"School", 表"dbo.ICBC\_Card", column 'stu\_card\_id'。  
语句已终止。

由于ICBC\_Card表中的外键属性是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ICBC\_Card无法执行，于是整个事务回滚。



## • 实验示例

验证事务回滚：再次查询表Stu\_card和ICBC\_Card，发现两个表的数据都没有被删除。

```
SQLQuery5.sql - (...BRTDQO\dgn (56))* SQLQuery4.sql - (...BRTDQO\dgn (55))*
1 USE School
2 select * from stu_card;
3 select * from icbc_card;
4
```

<

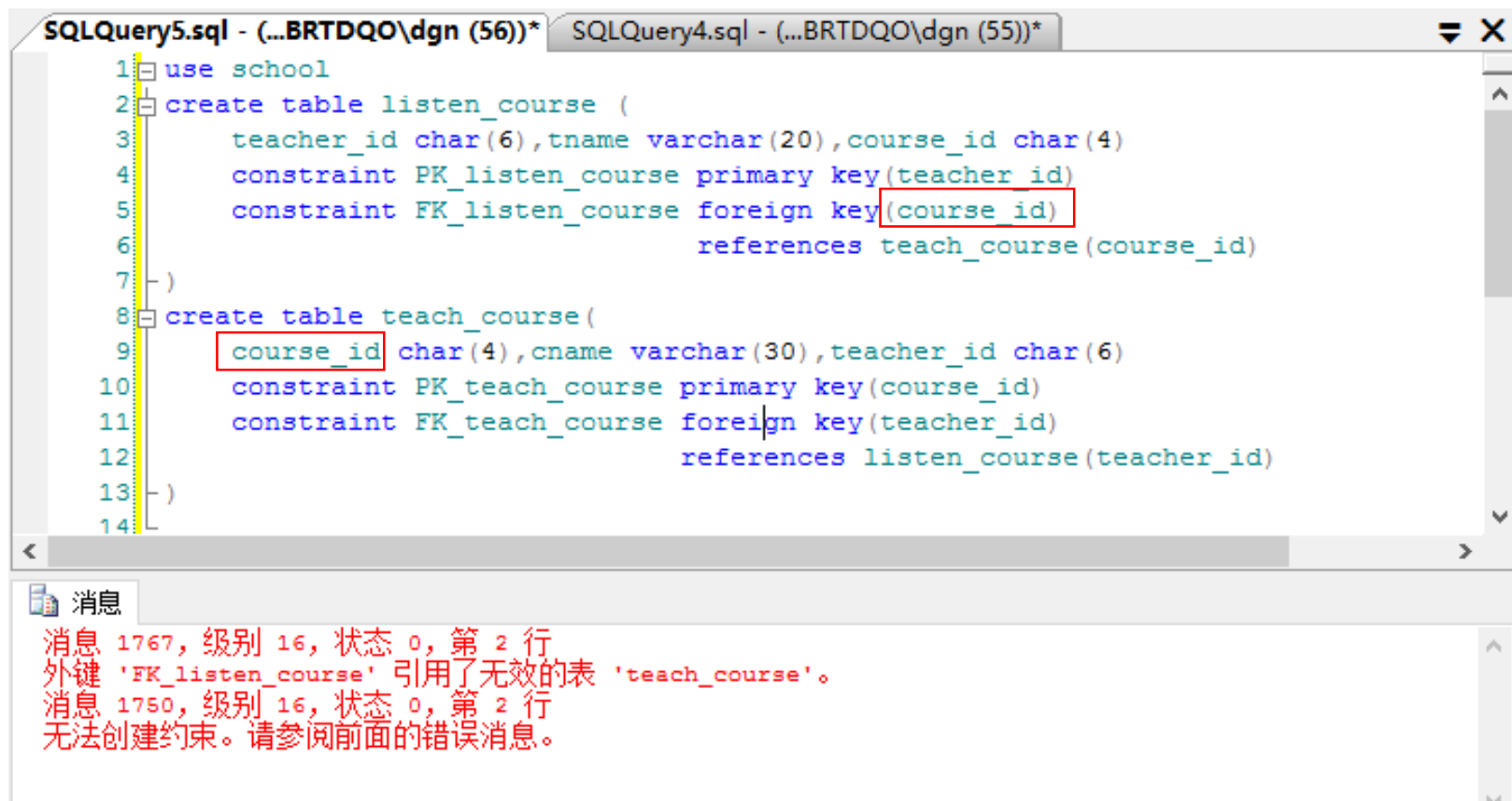
结果 消息

|   | card_id  | stu_id    | remained_money |
|---|----------|-----------|----------------|
| 1 | 05212222 | 800005753 | 200.50         |

|   | bank_id       | stu_card_id | restored_money |
|---|---------------|-------------|----------------|
| 1 | 9558844023645 | 05212222    | 50000.30       |

# 实验示例

7.演示互参照问题及其解决办法。要建立教师授课和课程指定教师听课关系的两张表，规定一个教师可授多门课，但每个课程只能指定一个教师去听课，所以要为两张表建立互相之间的参照关系。



```
SQLQuery5.sql - (...BRTDQO\dgn (56))* SQLQuery4.sql - (...BRTDQO\dgn (55))*
1 use school
2 create table listen_course (
3     teacher_id char(6), tname varchar(20), course_id char(4)
4     constraint PK_listen_course primary key(teacher_id)
5     constraint FK_listen_course foreign key(course_id)
6         references teach_course(course_id)
7 )
8 create table teach_course(
9     course_id char(4), cname varchar(30), teacher_id char(6)
10    constraint PK_teach_course primary key(course_id)
11    constraint FK_teach_course foreign key(teacher_id)
12        references listen_course(teacher_id)
13 )
14
```

消息 1767, 级别 16, 状态 0, 第 2 行  
外键 'FK\_listen\_course' 引用了无效的表 'teach\_course'。  
消息 1750, 级别 16, 状态 0, 第 2 行  
无法创建约束。请参阅前面的错误消息。

两张表互参照，会出现无法定义的问题。

解决办法：  
先定义listen\_course表，但不定义外键属性。  
再定义完整的teacher\_course表，用alter table的命令定义listen\_course的外键属性。

# • 实验示例

解决:

```
SQLQuery5.sql - (...BRTDQO\dgn (56))* SQLQuery4.sql - (...BRTDQO\dgn (55))*
1 use school
2 create table listen_course (
3     teacher_id char(6), tname varchar(20), course_id char(4)
4     constraint PK_listen_course primary key(teacher_id)
```

消息  
命令已成功完成。

先定义listen\_course表，但是不定义外键属性。

```
SQLQuery5.sql - (...BRTDQO\dgn (56))* SQLQuery4.sql - (...BRTDQO\dgn (55))*
1 use school
2 create table teach_course(
3     course_id char(4), cname varchar(30), teacher_id char(6)
4     constraint PK_teach_course primary key(course_id)
5     constraint FK_teach_course foreign key(teacher_id)
6     references listen_course(teacher_id)
7 )
8 alter table listen_course
9     add constraint FK_listen_course foreign key(course_id)
10     references teach_course(course_id);
11
```

消息  
命令已成功完成。

再定义完整的teacher\_course表，用alter table的命令定义listen\_course的外键属性。

互参照定义成功。

## • 练习

- (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）