# Contents

# 数据库实验 9 - 数据库完整性

## 实验目的

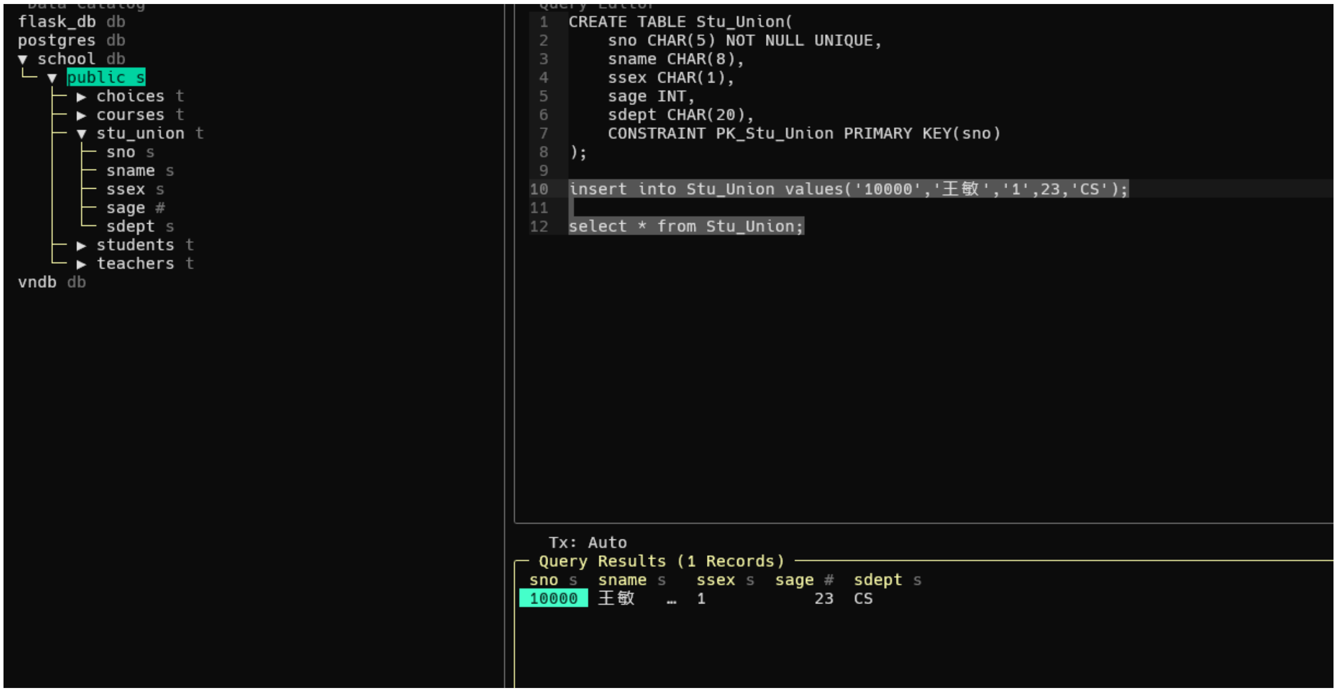
学习实体完整性的建立，以及实践违反实体完整性的结果；学习建立外键，以及利用 FOREIGN KEY... REFERENCES 子句以及各种约束保证参照完整性。

## 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：)

```
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;
```

- 创建并插入一条记录

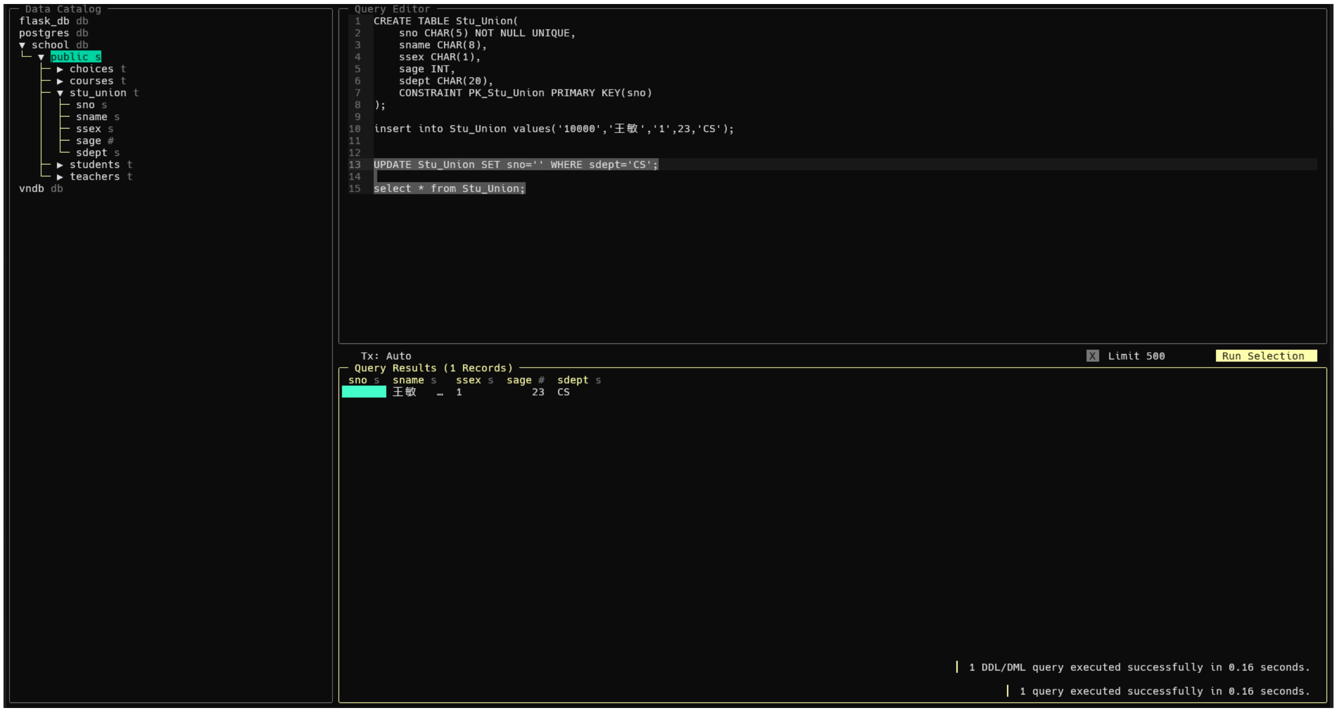


```
flask_db db
postgres db
▼ school db
└ ▼ public s
    ├ ▶ choices t
    ├ ▶ courses t
    ├ ▼ stu_union t
    │   ├ sno s
    │   ├ sname s
    │   ├ ssex s
    │   ├ sage #
    │   └ sdept s
    ├ ▶ students t
    └ ▶ teachers t
vndb db
```

```
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');

select * from Stu_Union;
```

```
Tx: Auto
Query Results (1 Records)
sno s  sname s   ssex s  sage #  sdept s
10000  王敏   …  1          23  CS
```

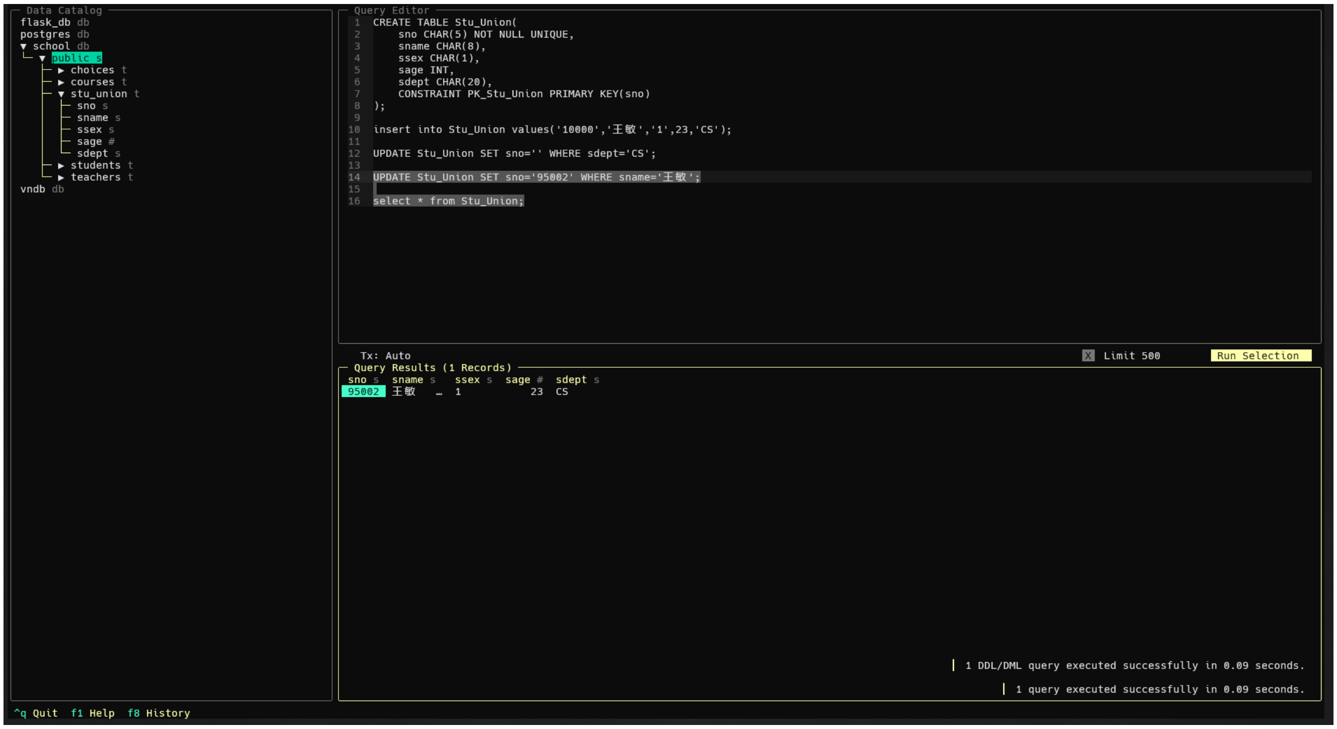
- 更新一条记录



```
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';

select * from Stu_Union;
```

```
Query Results (1 Records)
sno s  sname s   ssex s  sage #  sdept s
       王敏   …  1           23  CS
```

```
1 DDL/DML query executed successfully in 0.16 seconds.
1 query executed successfully in 0.16 seconds.
```



```
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;
```

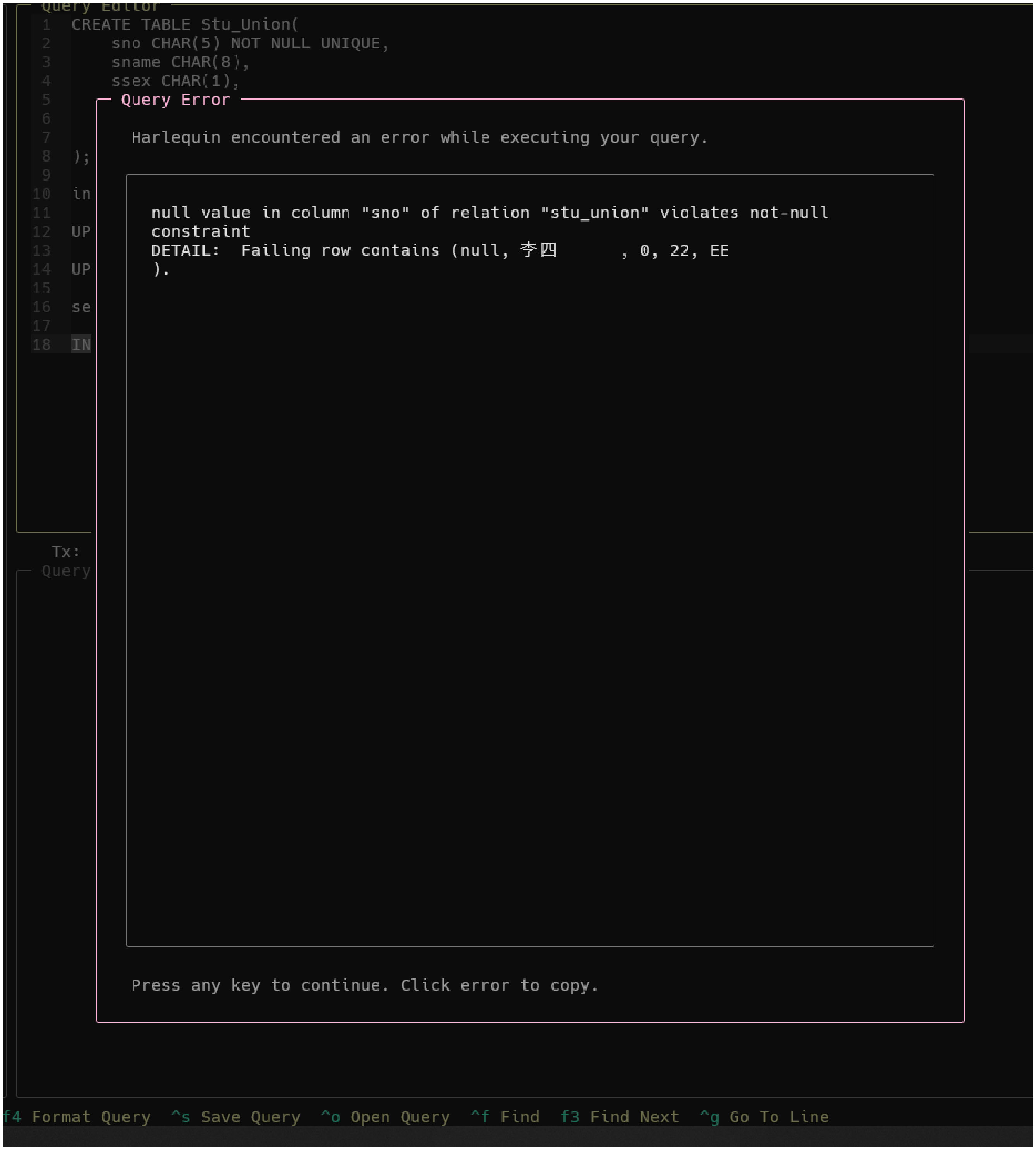
```
Query Results (1 Records)
sno s  sname s   ssex s  sage #  sdept s
95002  王敏   …  1           23  CS
```

```
1 DDL/DML query executed successfully in 0.09 seconds.
1 query executed successfully in 0.09 seconds.
```

2. 演示违反实体完整性的插入操作。(可截屏输出结果)

```
INSERT INTO Stu_Union VALUES(NULL,'李四','0',22,'EE');
```

执行结果如下:


Query Editor
Query Error
Tx:
Query
f4 Format Query  ^s Save Query  ^o Open Query  ^f Find  f3 Find Next  ^g Go To Line

```
CREATE TABLE Stu_Union(
    sno CHAR(5) NOT NULL UNIQUE,
    sname CHAR(8),
    ssex CHAR(1),



);

in

UP

UP

se

IN
```

Harlequin encountered an error while executing your query.

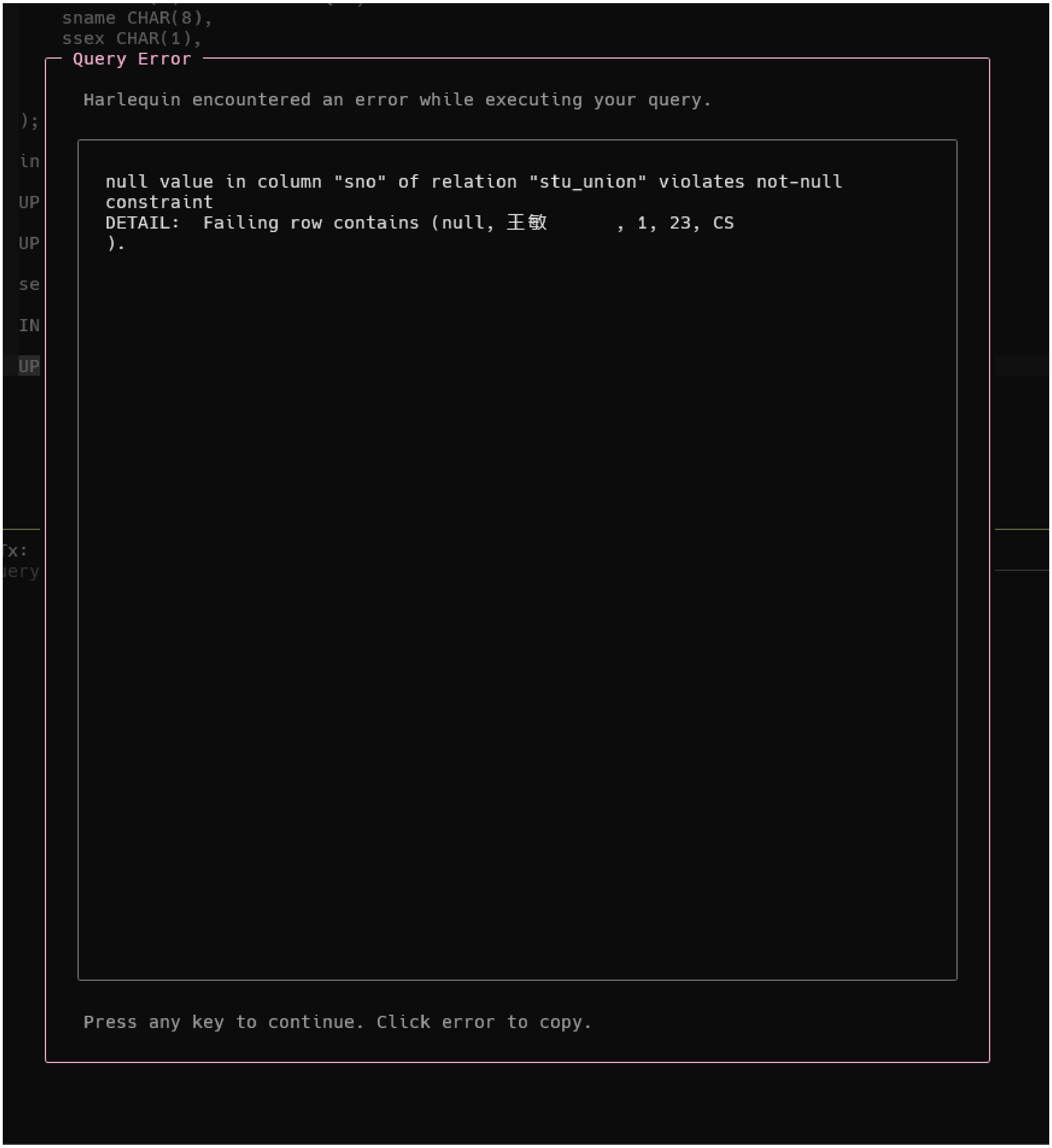
```
null value in column "sno" of relation "stu_union" violates not-null
constraint
DETAIL:  Failing row contains (null, 李四      , 0, 22, EE
).
```

Press any key to continue. Click error to copy.

3. 演示违反实体完整性的更新操作。

```
UPDATE Stu_Union SET sno = NULL WHERE sname = '王敏';
```

执行结果如下：



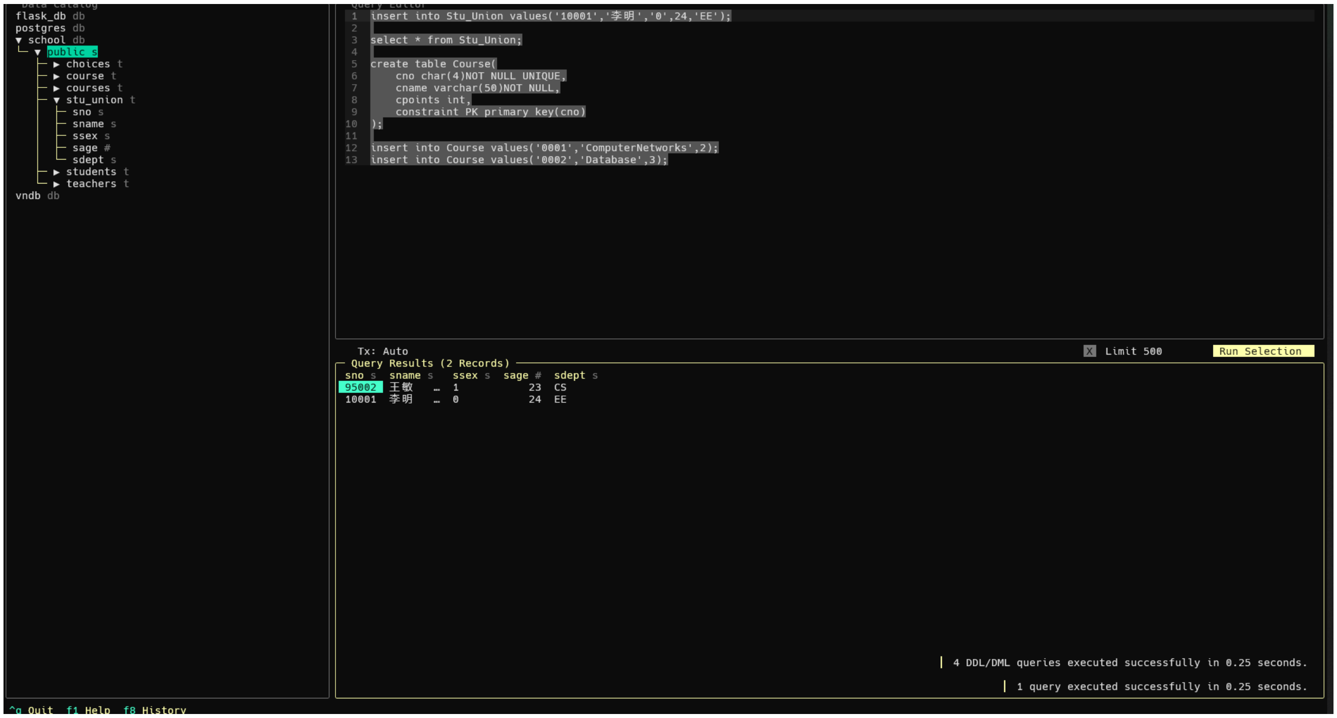
```
    sname CHAR(8),
    ssex CHAR(1),
```

```
Harlequin encountered an error while executing your query.

null value in column "sno" of relation "stu_union" violates not-null
constraint
DETAIL:  Failing row contains (null, 王敏      , 1, 23, CS
).

Press any key to continue. Click error to copy.
```

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','李明','0',24,'EE');

select * from Stu_Union;

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','Database',3);
```

执行结果如下:



```
flask_db db
postgres db
▼ school db
└ ▼ public s
  ├ ▶ choices t
  ├ ▶ course t
  ├ ▶ courses t
  ├ ▼ stu_union t
  │  ├ sno s
  │  ├ sname s
  │  ├ ssex s
  │  ├ sage #
  │  └ sdept s
  ├ ▶ students t
  └ ▶ teachers t
vndb db
```

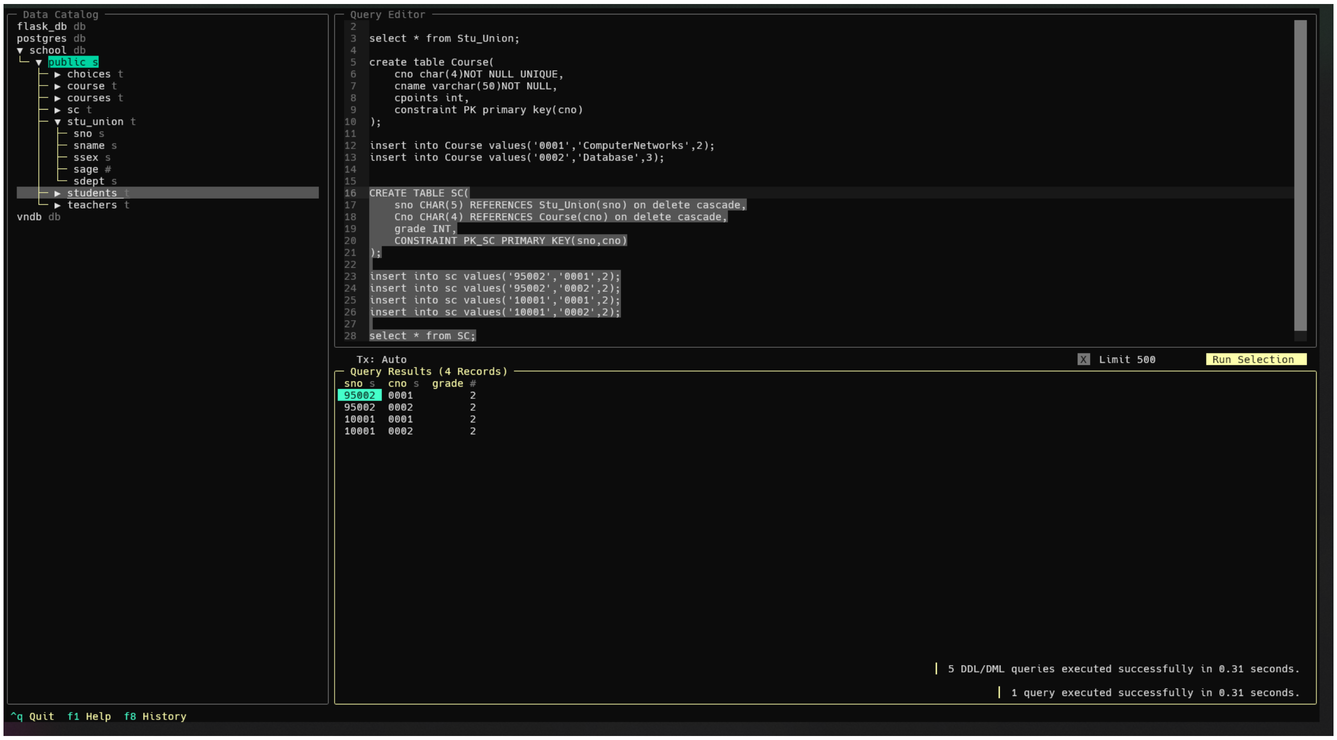
```
insert into Stu_Union values('10001','李明','0',24,'EE');

select * from Stu_Union;

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','Database',3);
```

```
Tx: Auto                                              X  Limit 500      Run Selection
Query Results (2 Records)
sno s  sname s   ssex s  sage #  sdept s
95002  王敏   …  1           23  CS
10001  李明   …  0           24  EE

                                   | 4 DDL/DML queries executed successfully in 0.25 seconds.
                                       | 1 query executed successfully in 0.25 seconds.
```

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) 为主主键, 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)
);

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;
```

执行结果如下:



```

select * from Stu_Union;

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','Database',3);


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;
```

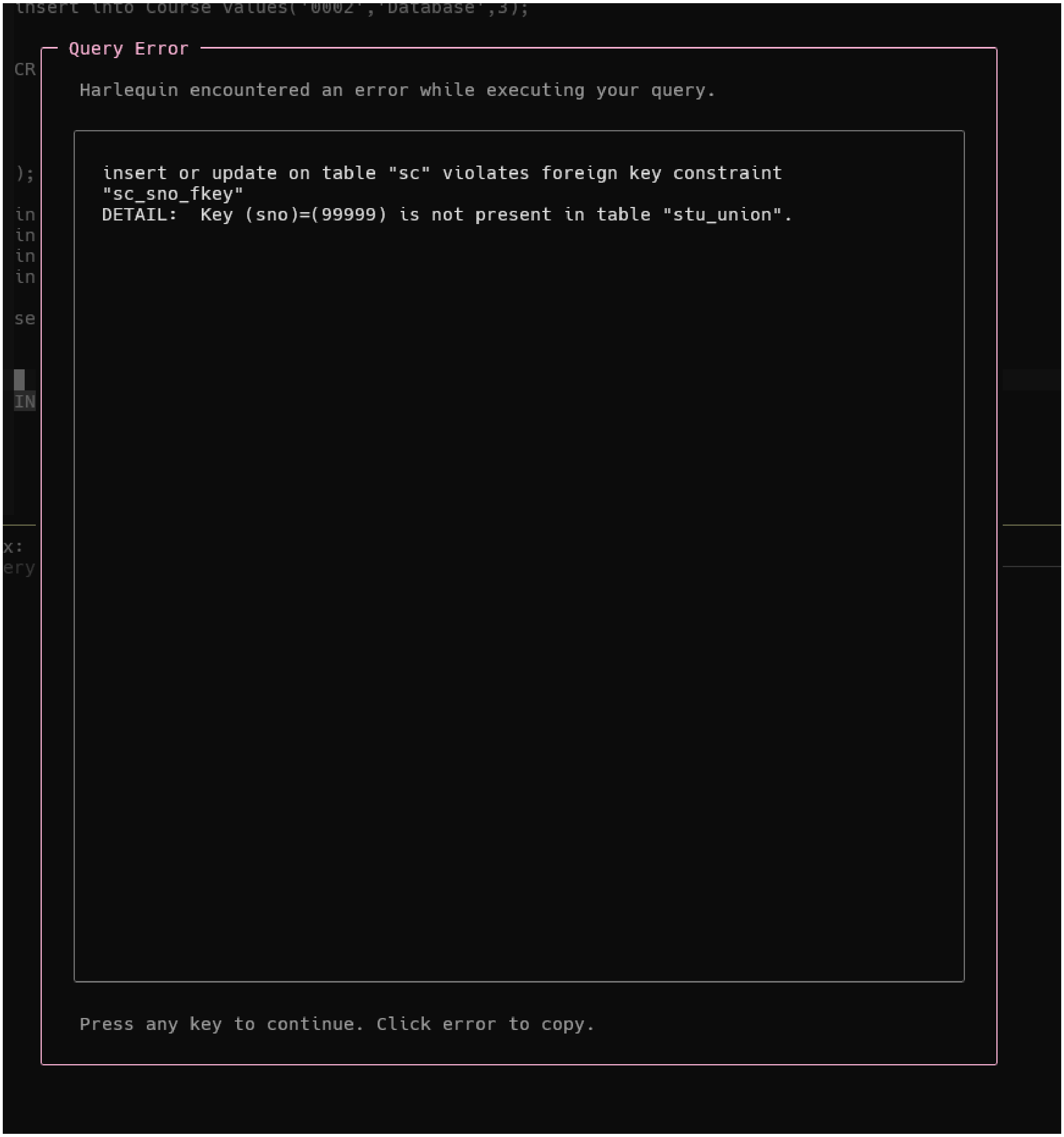
```
Query Results (4 Records)
sno s  cno s  grade #
95002  0001       2
95002  0002       2
10001  0001       2
10001  0002       2
```

```
5 DDL/DML queries executed successfully in 0.31 seconds.
1 query executed successfully in 0.31 seconds.
```

6. 演示违反参照完整性的插入操作。

```
INSERT INTO SC VALUES('99999','0001',2);
```

执行结果如下：



```
insert into Course values('0002','Database',3);
```

Query Error

Harlequin encountered an error while executing your query.

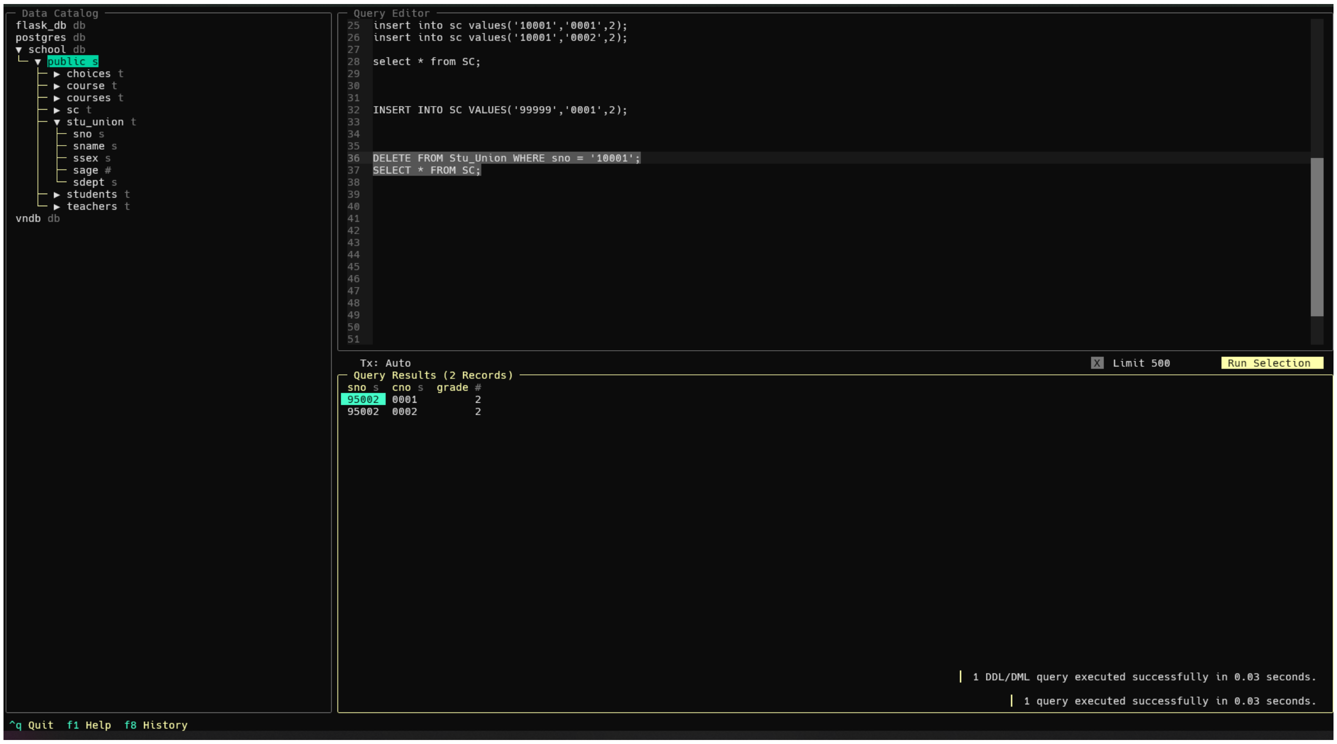
```
insert or update on table "sc" violates foreign key constraint
"sc_sno_fkey"
DETAIL:  Key (sno)=(99999) is not present in table "stu_union".
```

Press any key to continue. Click error to copy.

7. 在 Stu_Union 中删除数据，演示级联删除。

```
DELETE FROM Stu_Union WHERE sno = '10001';
SELECT * FROM SC;
```

执行结果如下：



```
Data Catalog
flask_db db
postgres db
▼ school db
└ ▼ public s
  ├ ▶ choices t
  ├ ▶ course t
  ├ ▶ courses t
  ├ ▶ sc t
  ├ ▼ stu_union t
  │ ├ sno s
  │ ├ sname s
  │ ├ ssex s
  │ ├ sage #
  │ └ sdept s
  ├ ▶ students t
  └ ▶ teachers t
vndb db
```

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

select * from SC;


INSERT INTO SC VALUES('99999','0001',2);


DELETE FROM Stu_Union WHERE sno = '10001';
SELECT * FROM SC;
```

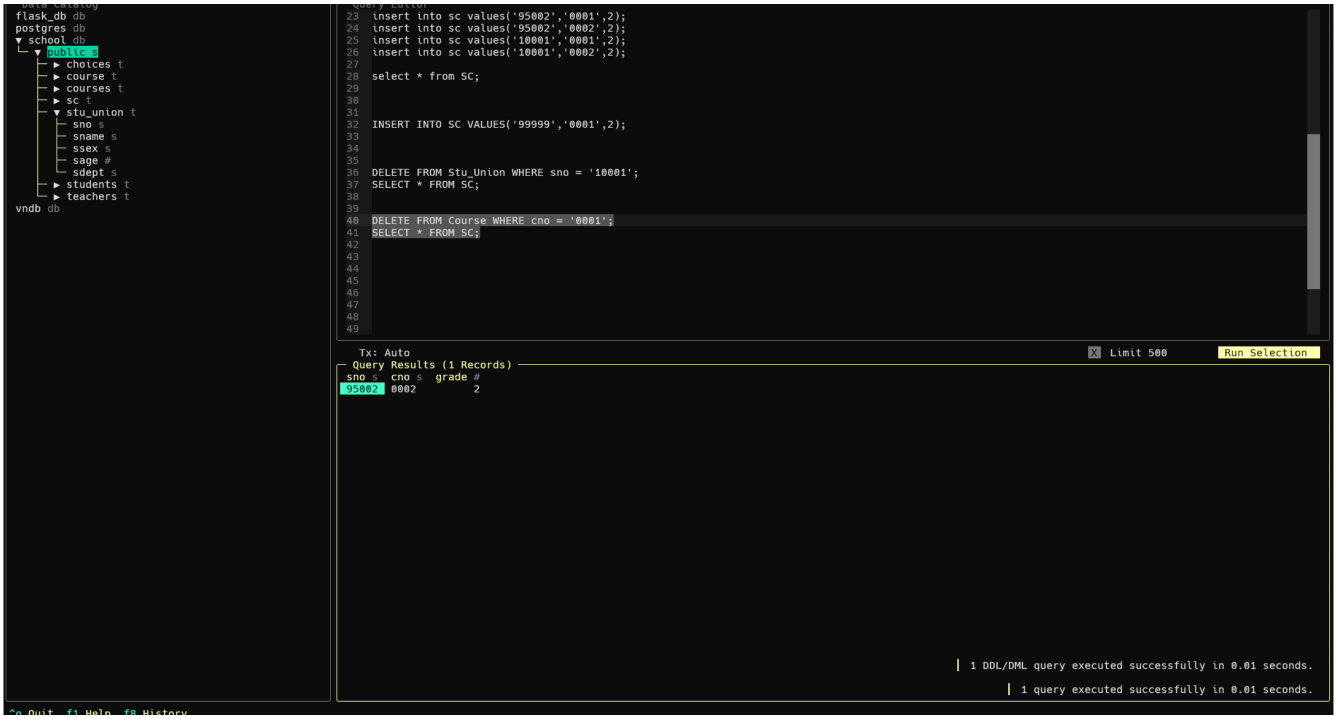
```
Tx: Auto                                   Limit 500     Run Selection
Query Results (2 Records)
sno s  cno s  grade #
95002  0001         2
95002  0002         2

| 1 DDL/DML query executed successfully in 0.03 seconds.
| 1 query executed successfully in 0.03 seconds.
^q Quit  f1 Help  f8 History
```

可见，在删除了 Stu_Union 表中的一条记录后，SC 表中的相应记录也被级联删除了。

8. 在 Course 中删除数据，演示级联删除。

```
DELETE FROM Course WHERE cno = '0001';
SELECT * FROM SC;
```

执行结果如下：



```
Data Catalog
flask_db db
postgres db
▼ school db
└ ▼ public s
  ├ ▶ choices t
  ├ ▶ course t
  ├ ▶ courses t
  ├ ▶ sc t
  ├ ▼ stu_union t
  │ ├ sno s
  │ ├ sname s
  │ ├ ssex s
  │ ├ sage #
  │ └ sdept s
  ├ ▶ students t
  └ ▶ teachers t
vndb db
```

```
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('99999','0001',2);


DELETE FROM Stu_Union WHERE sno = '10001';
SELECT * FROM SC;


DELETE FROM Course WHERE cno = '0001';
SELECT * FROM SC;
```

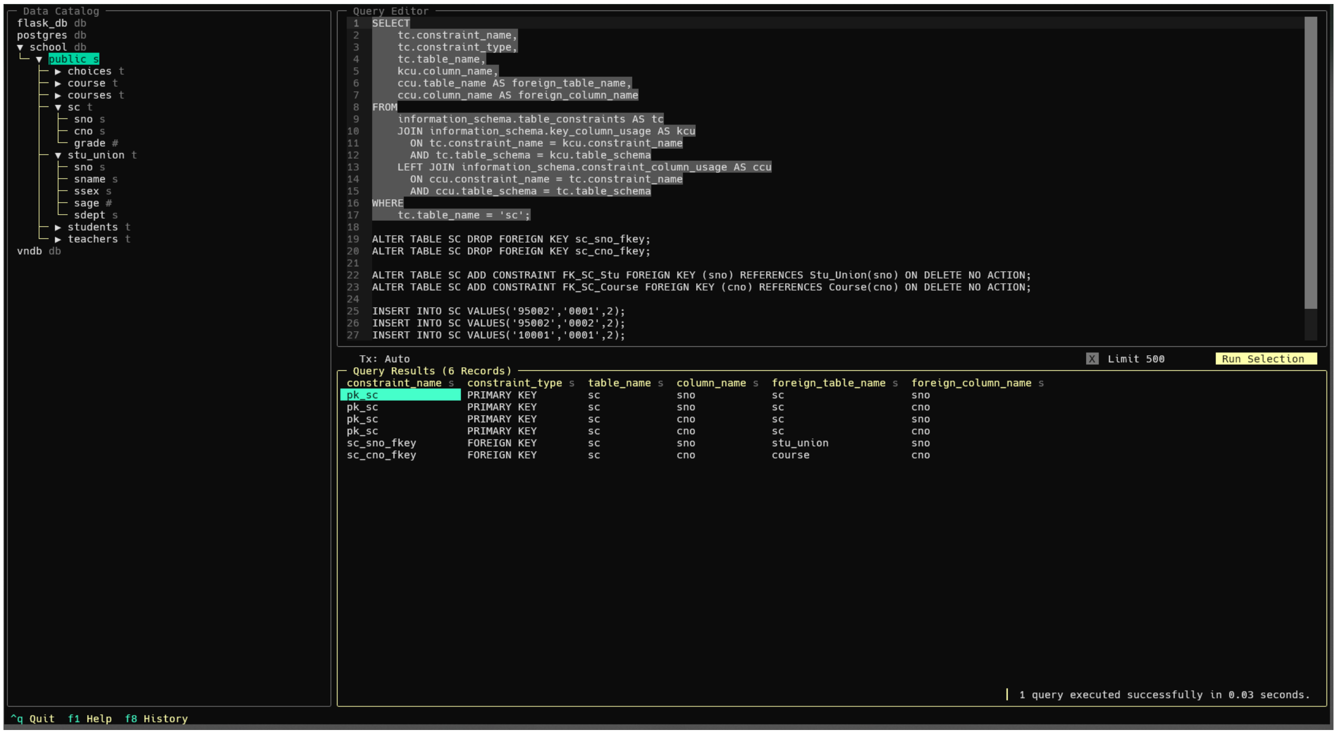
```
Tx: Auto                                   Limit 500     Run Selection
Query Results (1 Records)
sno s  cno s  grade #
95002  0002         2

| 1 DDL/DML query executed successfully in 0.01 seconds.
| 1 query executed successfully in 0.01 seconds.
^q Quit  f1 Help  f8 History
```

可见，在删除了 Course 表中的一条记录后，SC 表中的相应记录也被级联删除了。

## 自我实践

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。

- 先查看一下外键约束的名称：

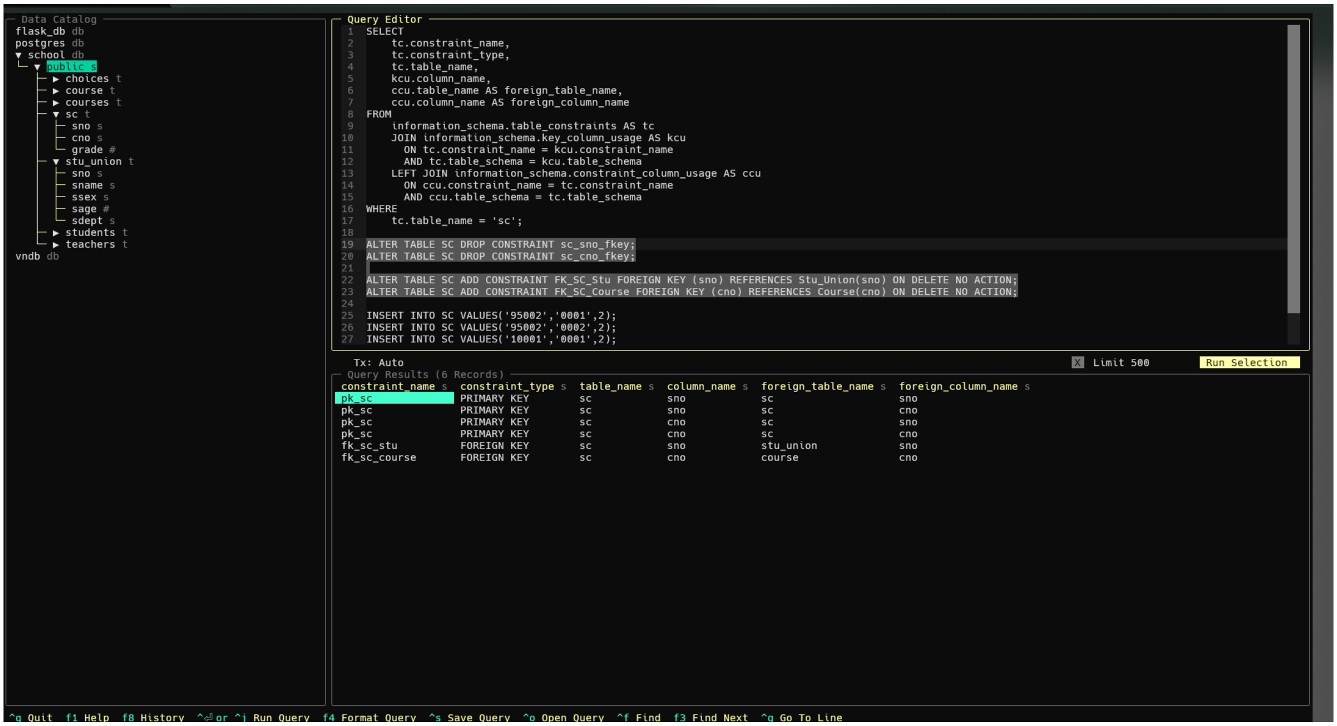


```
SELECT
    tc.constraint_name,
    tc.constraint_type,
    tc.table_name,
    kcu.column_name,
    ccu.table_name AS foreign_table_name,
    ccu.column_name AS foreign_column_name
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
      ON tc.constraint_name = kcu.constraint_name
      AND tc.table_schema = kcu.table_schema
    LEFT JOIN information_schema.constraint_column_usage AS ccu
      ON ccu.constraint_name = tc.constraint_name
      AND ccu.table_schema = tc.table_schema
WHERE
    tc.table_name = 'sc';

ALTER TABLE SC DROP FOREIGN KEY sc_sno_fkey;
ALTER TABLE SC DROP FOREIGN KEY sc_cno_fkey;

ALTER TABLE SC ADD CONSTRAINT FK_SC_Stu FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON DELETE NO ACTION;
ALTER TABLE SC ADD CONSTRAINT FK_SC_Course FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE NO ACTION;

INSERT INTO SC VALUES('95002','0001',2);
INSERT INTO SC VALUES('95002','0002',2);
INSERT INTO SC VALUES('10001','0001',2);
```

```
Query Results (6 Records)
constraint_name s  constraint_type s  table_name s  column_name s  foreign_table_name s  foreign_column_name s
pk_sc              PRIMARY KEY        sc            sno            sc                    sno
pk_sc              PRIMARY KEY        sc            sno            sc                    cno
pk_sc              PRIMARY KEY        sc            cno            sc                    sno
pk_sc              PRIMARY KEY        sc            cno            sc                    cno
sc_sno_fkey        FOREIGN KEY        sc            sno            stu_union             sno
sc_cno_fkey        FOREIGN KEY        sc            cno            course                cno

                                                              1 query executed successfully in 0.03 seconds.
```

- 更改 SC 表约束：

```
ALTER TABLE SC DROP CONSTRAINT sc_sno_fkey;
ALTER TABLE SC DROP CONSTRAINT sc_cno_fkey;

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


Data Catalog
Query Editor
Tx: Auto
Limit 500
Run Selection
Query Results (6 Records)
^q Quit  f1 Help  f8 History  ^⏎ or ^j Run Query  f4 Format Query  ^s Save Query  ^o Open Query  ^f Find  f3 Find Next  ^g Go To Line

```
flask_db db
postgres db
▼ school db
└ ▼ public s
  ├ ▶ choices t
  ├ ▶ course t
  ├ ▶ courses t
  ├ ▼ sc t
  │ ├ sno s
  │ ├ cno s
  │ └ grade #
  ├ ▼ stu_union t
  │ ├ sno s
  │ ├ sname s
  │ ├ ssex s
  │ ├ sage #
  │ └ sdept s
  ├ ▶ students t
  └ ▶ teachers t
vndb db
```

```sql
SELECT
    tc.constraint_name,
    tc.constraint_type,
    tc.table_name,
    kcu.column_name,
    ccu.table_name AS foreign_table_name,
    ccu.column_name AS foreign_column_name
FROM
    information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu
      ON tc.constraint_name = kcu.constraint_name
      AND tc.table_schema = kcu.table_schema
    LEFT JOIN information_schema.constraint_column_usage AS ccu
      ON ccu.constraint_name = tc.constraint_name
      AND ccu.table_schema = tc.table_schema
WHERE
    tc.table_name = 'sc';

ALTER TABLE SC DROP CONSTRAINT sc_sno_fkey;
ALTER TABLE SC DROP CONSTRAINT sc_cno_fkey;

ALTER TABLE SC ADD CONSTRAINT FK_SC_Stu FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON DELETE NO ACTION;
ALTER TABLE SC ADD CONSTRAINT FK_SC_Course FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE NO ACTION;

INSERT INTO SC VALUES('95002','0001',2);
INSERT INTO SC VALUES('95002','0002',2);
INSERT INTO SC VALUES('10001','0001',2);
```

```
constraint_name s  constraint_type s  table_name s  column_name s  foreign_table_name s  foreign_column_name s
pk_sc              PRIMARY KEY        sc            sno            sc                    sno
pk_sc              PRIMARY KEY        sc            sno            sc                    cno
pk_sc              PRIMARY KEY        sc            cno            sc                    sno
pk_sc              PRIMARY KEY        sc            cno            sc                    cno
fk_sc_stu          FOREIGN KEY        sc            sno            stu_union             sno
fk_sc_course       FOREIGN KEY        sc            cno            course                cno
```

- 重新插入数据（当然，其它表格也需要恢复删除的数据，我直接用备份了，这里略过）：

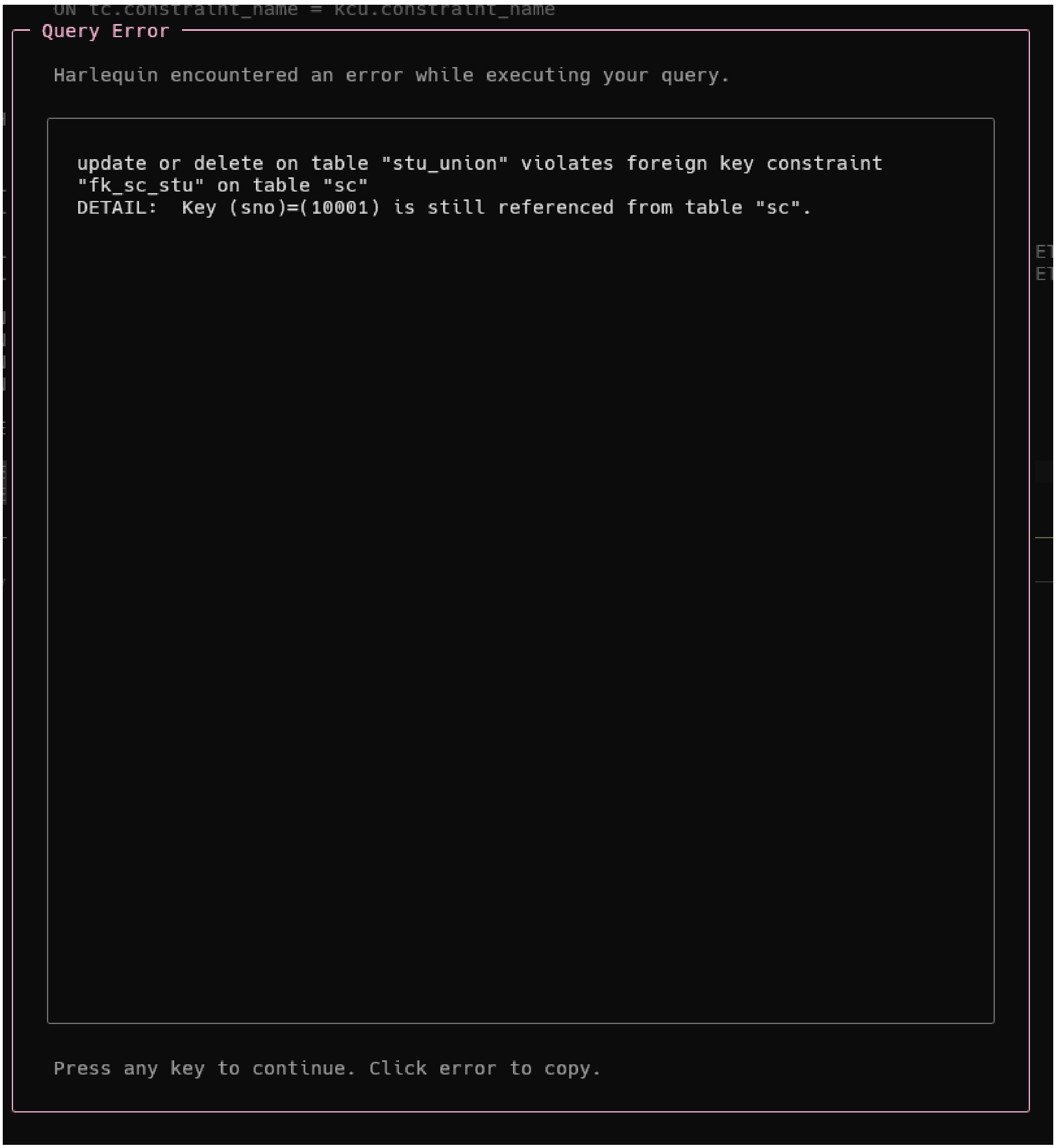
```sql
INSERT INTO SC VALUES('95002','0001',2);
INSERT INTO SC VALUES('95002','0002',2);
INSERT INTO SC VALUES('10001','0001',2);
INSERT INTO SC VALUES('10001','0002',2);

SELECT * FROM SC;
```

009-3

- 重复课内实验 7：

```sql
DELETE FROM Stu_Union WHERE sno = '10001';
SELECT * FROM SC;
```

执行结果如下：



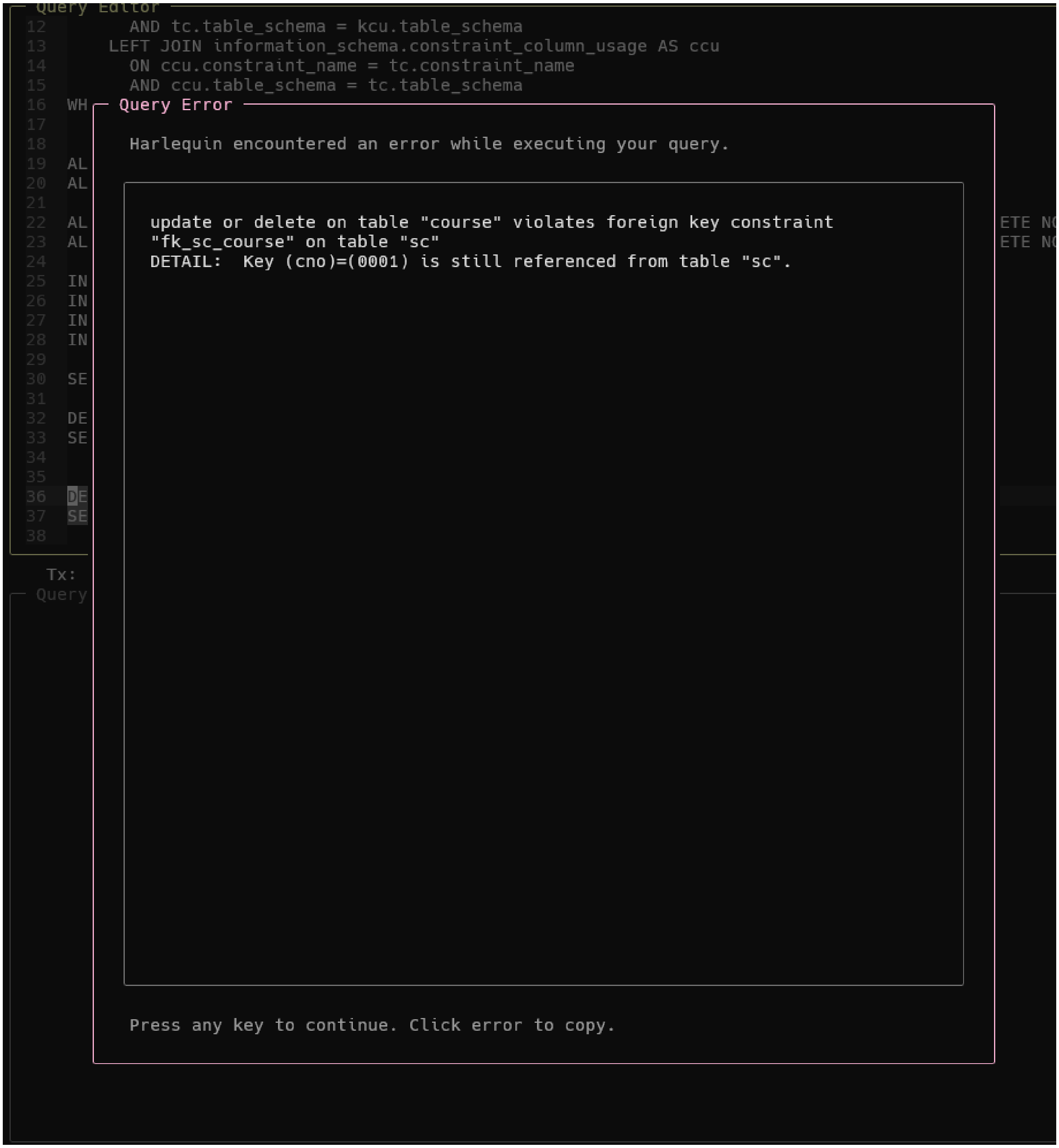
```
ON tc.constraint_name = kcu.constraint_name
```

```
Harlequin encountered an error while executing your query.

update or delete on table "stu_union" violates foreign key constraint
"fk_sc_stu" on table "sc"
DETAIL:  Key (sno)=(10001) is still referenced from table "sc".

Press any key to continue. Click error to copy.
```

可见，由于在表格 SC 中设置了 on delete no action，当尝试删除 Stu_Union 中的一行数据时，约束会阻止从 SC 中删除条目。为了保持数据完整性，防止在 SC 中出现悬挂引用，数据库将会阻止删除操作。

- 重复课内实验 8：

执行结果如下：



```
      AND tc.table_schema = kcu.table_schema
    LEFT JOIN information_schema.constraint_column_usage AS ccu
      ON ccu.constraint_name = tc.constraint_name
      AND ccu.table_schema = tc.table_schema
WH

AL
AL

AL                                                   ETE NO
AL                                                   ETE NO

IN
IN
IN
IN

SE

DE
SE


DE
SE
```

Harlequin encountered an error while executing your query.

```
update or delete on table "course" violates foreign key constraint
"fk_sc_course" on table "sc"
DETAIL:  Key (cno)=(0001) is still referenced from table "sc".
```

Press any key to continue. Click error to copy.

与上述相同。

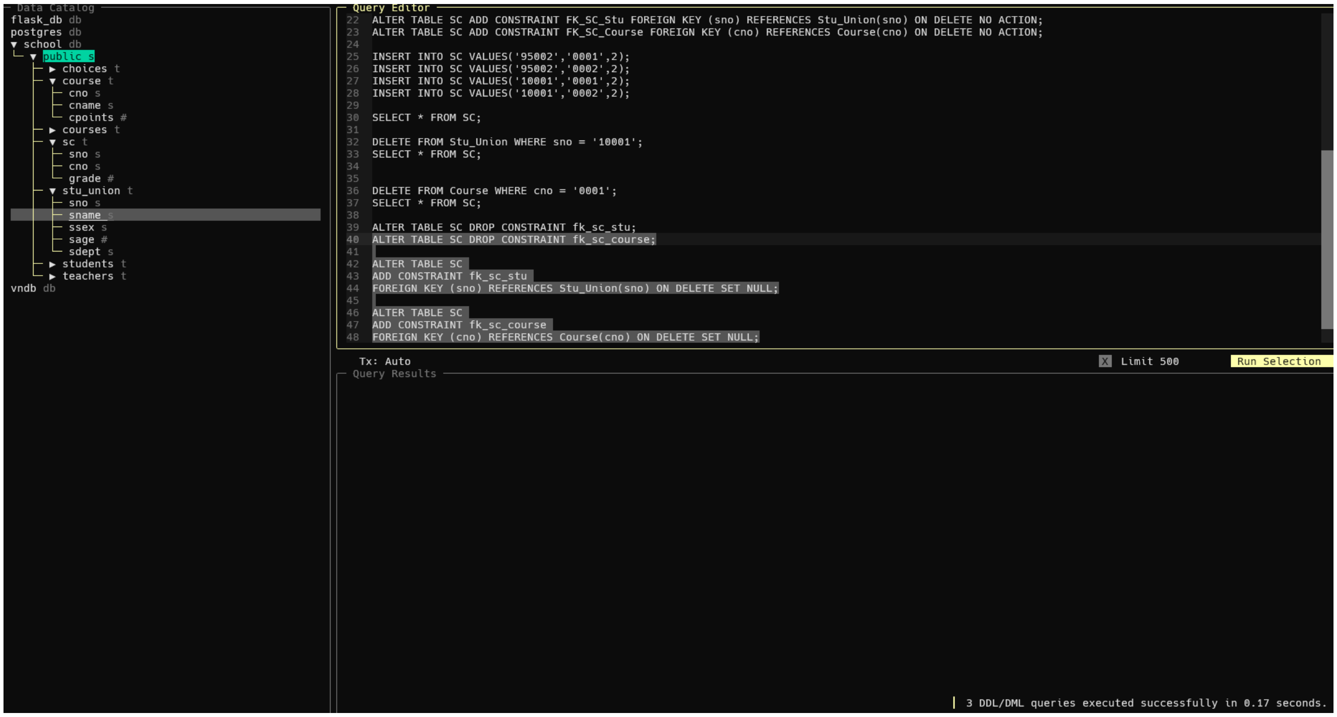
2. 使用 alter table 语句将 SC 表中的 on delete cascade 改为 on delete set null，重新插入 SC 的数据。重复课内实验中 7. 和 8.，观察结果，分析原因。

- 更改约束

```
ALTER TABLE SC DROP CONSTRAINT fk_sc_stu;
ALTER TABLE SC DROP CONSTRAINT sk_sc_course;

ALTER TABLE SC
ADD CONSTRAINT fk_sc_stu
FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON DELETE SET NULL;
```

```
ALTER TABLE SC
ADD CONSTRAINT fk_sc_course
FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE SET NULL;
```

执行结果如下：

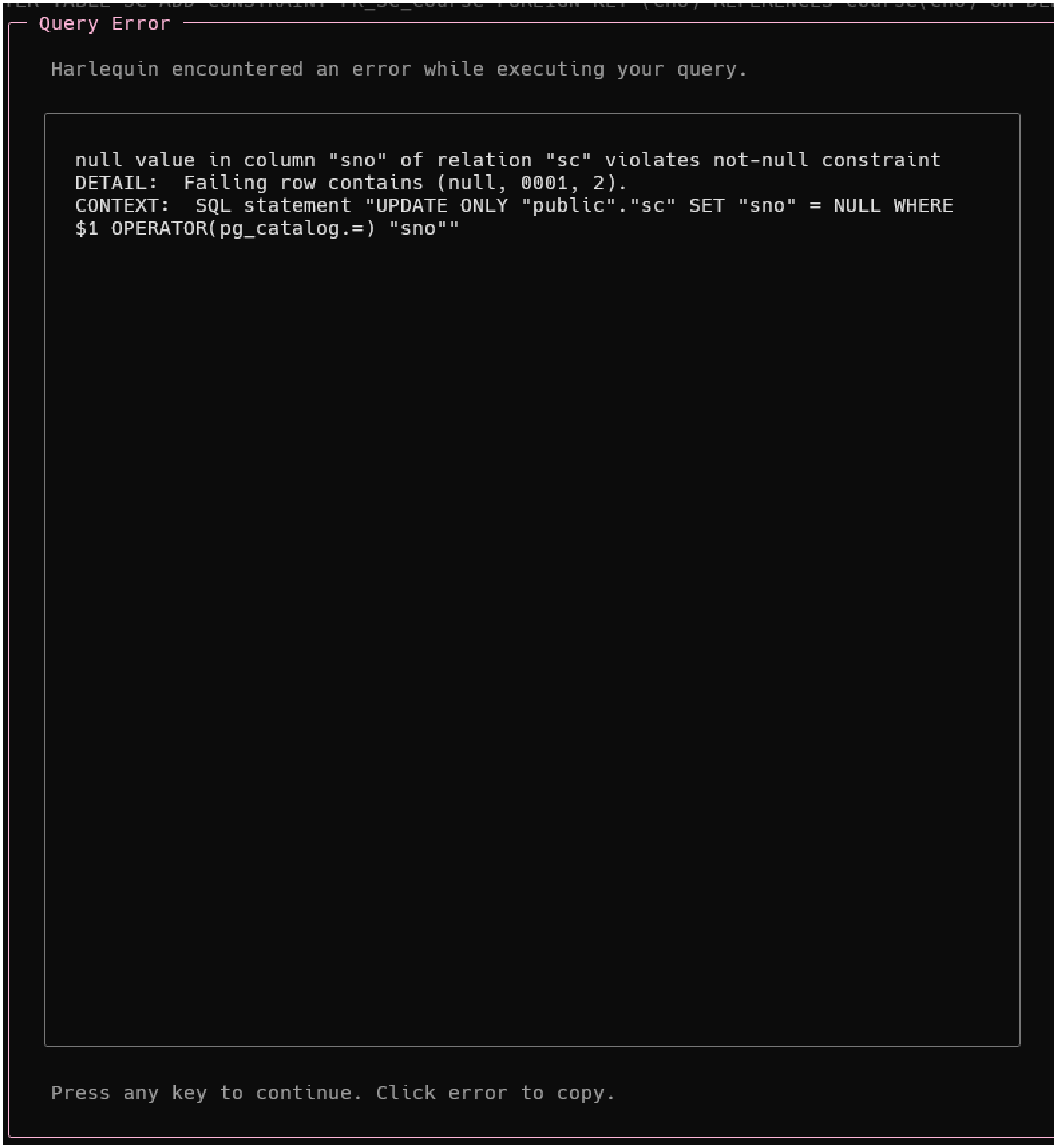


```
ALTER TABLE SC ADD CONSTRAINT FK_SC_Stu FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON DELETE NO ACTION;
ALTER TABLE SC ADD CONSTRAINT FK_SC_Course FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE NO ACTION;

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;

DELETE FROM Stu_Union WHERE sno = '10001';
SELECT * FROM SC;


DELETE FROM Course WHERE cno = '0001';
SELECT * FROM SC;

ALTER TABLE SC DROP CONSTRAINT fk_sc_stu;
ALTER TABLE SC DROP CONSTRAINT fk_sc_course;

ALTER TABLE SC
ADD CONSTRAINT fk_sc_stu
FOREIGN KEY (sno) REFERENCES Stu_Union(sno) ON DELETE SET NULL;

ALTER TABLE SC
ADD CONSTRAINT fk_sc_course
FOREIGN KEY (cno) REFERENCES Course(cno) ON DELETE SET NULL;
```

```
3 DDL/DML queries executed successfully in 0.17 seconds.
```

- 重复课内实验中 7. 和 8.

执行结果如下：



```
Harlequin encountered an error while executing your query.

null value in column "sno" of relation "sc" violates not-null constraint
DETAIL:  Failing row contains (null, 0001, 2).
CONTEXT:  SQL statement "UPDATE ONLY "public"."sc" SET "sno" = NULL WHERE
$1 OPERATOR(pg_catalog.=) "sno""

Press any key to continue. Click error to copy.
```

当试图从 Stu_Union 中删除一个条目时，由于设置了 on delete set null，数据库会将 SC 表中与该条目相关的行的 sno 字段设置为 null，而不会删除该行。然而，此处的 sno 字段为 SC 表主键的一部分，它不能被设置为 null，因此，数据库会阻止这个操作。