

# Lecture 12 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

- 实验示例

- 练习

- 实验目的

通过实验加深对数据完整性的理解，学会创建和使用触发器。

## • 实验示例

本次课实验是在上周实验课建立的worker表基础上进行实验。

1.为worker表建立触发器T1，当插入或是更新表中数据时，保证所操作的记录的sage值大于0。

```
2.4.1.sql - (local...P-NBRTDQO\dgn (62))  SQLQuery4.sql - (...BRTDQO\dgn (61))*  2.3.7.sql - (loca

1  use school
2  go
3  create trigger T1 on worker
4  for insert , update
5  as
6  if (select sage from inserted)<1
7  begin
8  print 'Sage must be a integer more than zero! Tansaction fail'
9  Rollback transaction
10 End
11 |
```

消息  
命令已成功完成。

# • 实验示例

2.演示违反T1触发器约束的插入操作。

```
2.4.4.sql - (local)...\NBRTDQO\dgn (65))* 2.4.2.sql - (local...P-NBRTDQO\dgn (64)) SQLC
1 use school
2 insert into worker values ('00004','李红','F',-10,'开发部')
3
```

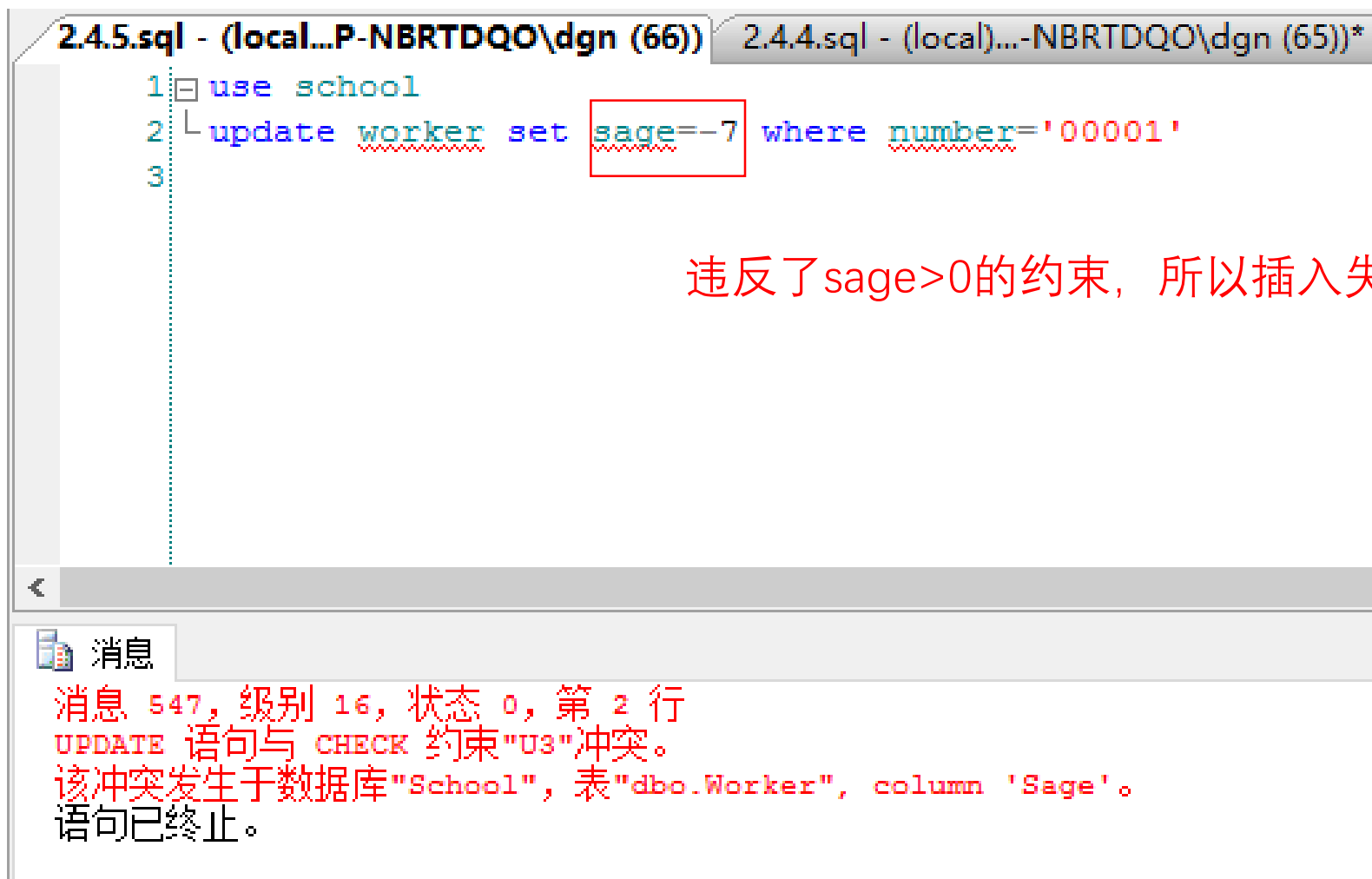
违反了sage>0的约束，所以插入失败。

## 消息

消息 547，级别 16，状态 0，第 2 行  
INSERT 语句与 CHECK 约束"U3"冲突。  
该冲突发生于数据库"School"，表"dbo.Worker"，column 'Sage'。  
语句已终止。

# 实验示例

## 3.演示违反T1触发器约束更新操作。



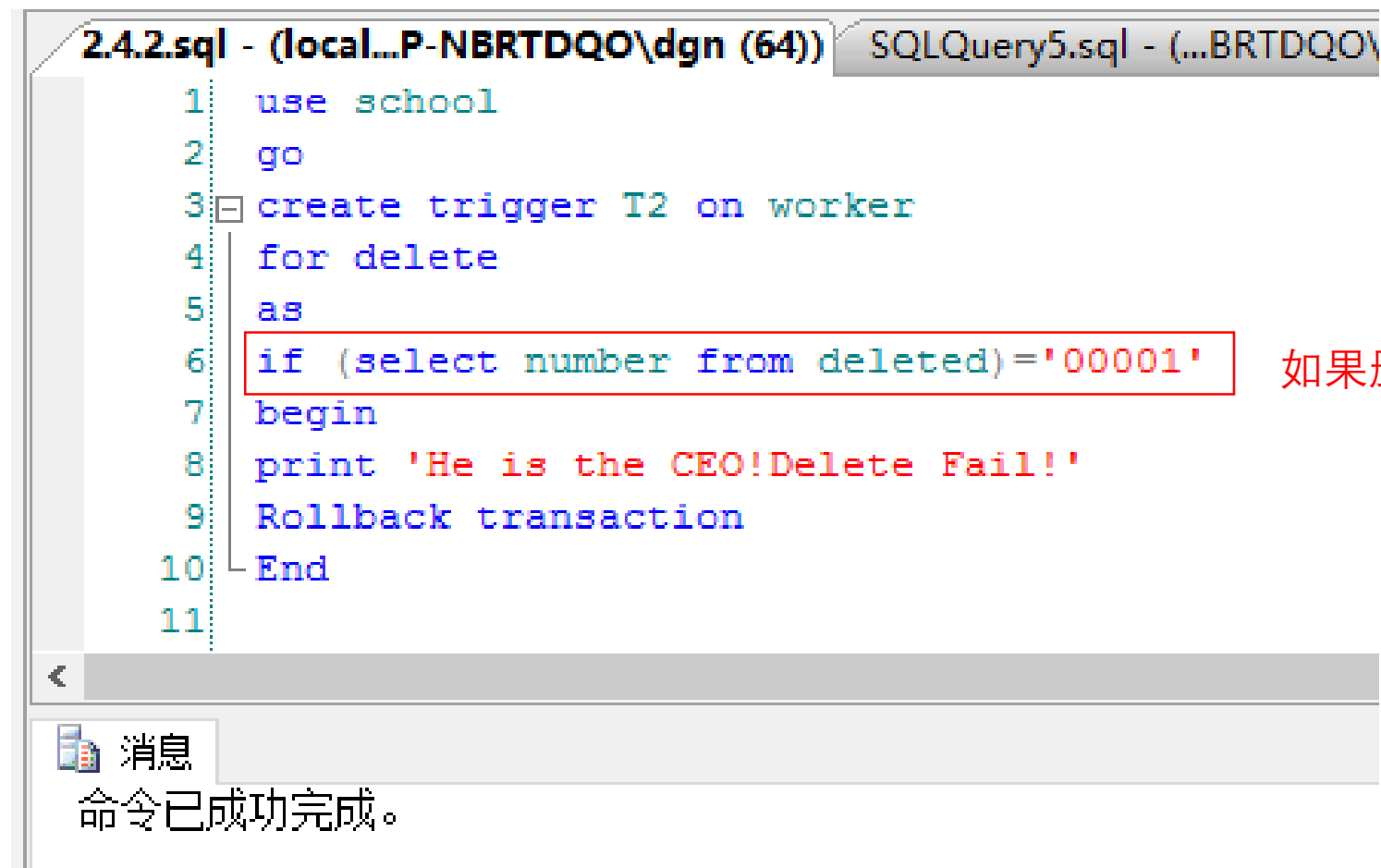
```
2.4.5.sql - (local...P-NBRTDQO\dgn (66)) 2.4.4.sql - (local)...-NBRTDQO\dgn (65))*  
1 use school  
2 update worker set sage=-7 where number='00001'  
3
```

违反了sage>0的约束，所以插入失败。

消息 547, 级别 16, 状态 0, 第 2 行  
UPDATE 语句与 CHECK 约束"U3"冲突。  
该冲突发生于数据库"School", 表"dbo.Worker", column 'Sage'。  
语句已终止。

# • 实验示例

4.为worker表建立触发器T2，禁止删除编号为00001的记录。



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: '2.4.2.sql - (local...P-NBRTDQO\dgn (64))' and 'SQLQuery5.sql - (...BRTDQO\...)'. The main window displays a SQL script with the following lines:

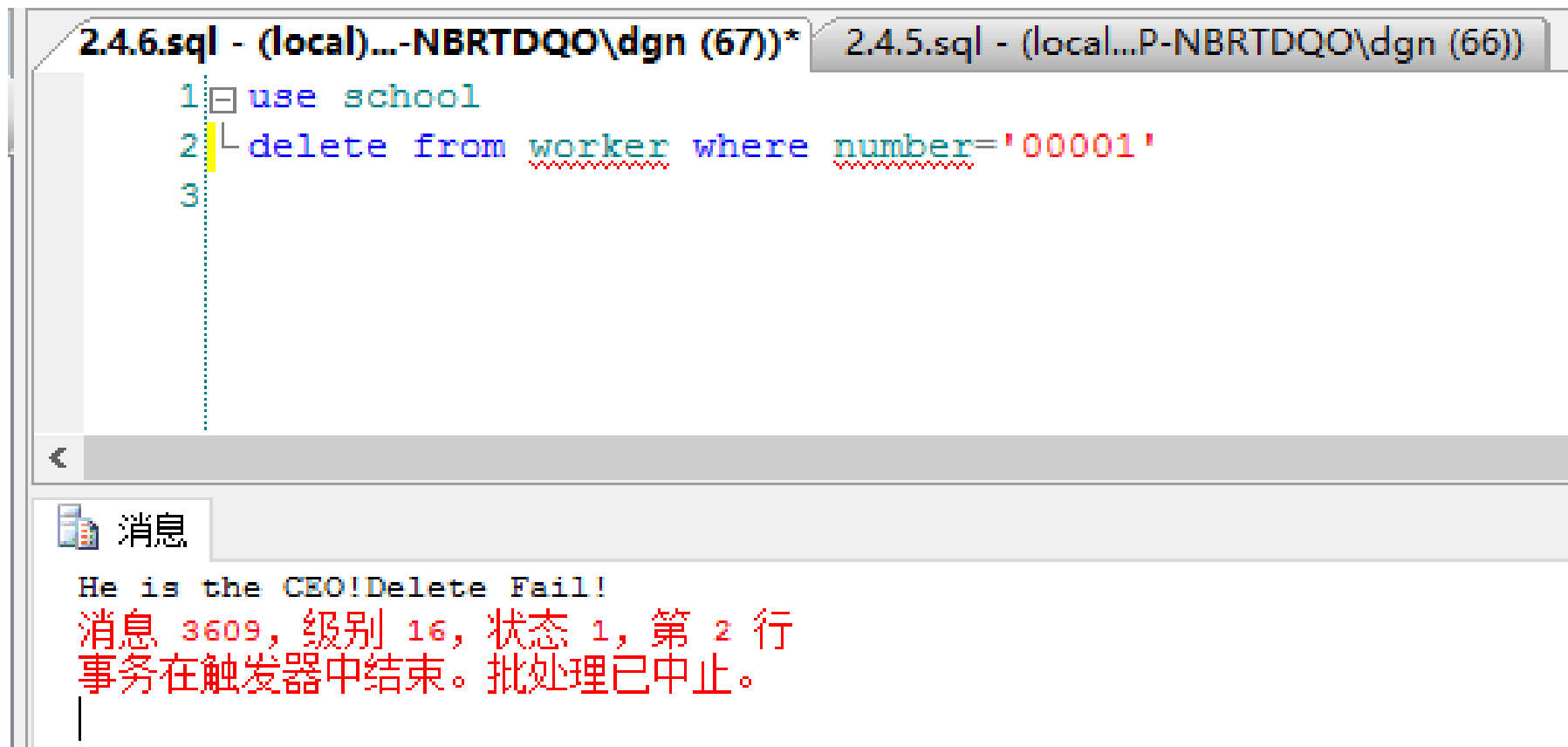
```
1 use school
2 go
3 create trigger T2 on worker
4 for delete
5 as
6 if (select number from deleted)='00001'
7 begin
8 print 'He is the CEO!Delete Fail!'
9 Rollback transaction
10 End
11
```

The line 'if (select number from deleted)='00001'' is highlighted with a red rectangular box. To the right of this box, there is a red text annotation: '如果删除的是00001记录，则操作失败'.

At the bottom of the window, there is a '消息' (Messages) pane. It contains a single message: '命令已成功完成。' (Command completed successfully.).

# • 实验示例

5.演示违反T2触发器的约束的操作。



The screenshot shows a SQL IDE with two tabs. The active tab is '2.4.6.sql - (local)...-NBRTDQO\dgn (67))\*'. It contains the following SQL code:

```
1 use school
2 delete from worker where number='00001'
3
```

Below the code editor is a '消息' (Messages) pane. It displays the following error message:

```
He is the CEO!Delete Fail!
消息 3609, 级别 16, 状态 1, 第 2 行
事务在触发器中结束。批处理已中止。
```

因为违反T2触发器的约束，所以数据删除失败



- 实验示例

6.Worker表中的人员编号是不允许改变的。创建触发器T3实现更新中编号不可改变性。

```
2.4.3.sql - (local...P-NBRTDQO\dgn (69))  2.4.7.sql - (local...P-NBRTDC
1  use school
2  go
3  create trigger T3 on Worker
4  for update
5  as
6  if update(number)  不可更新编号
7  begin
8  print 'Everynumber cannot be changed!'
9  Rollback Transaction
10 End
```

<

消息  
命令已成功完成。

- 实验示例

7.演示违反触发器T3的操作。

2.4.3.sql - (local...P-NBRTDQO\dgn (69))

2.4.7.sql - (local...P-NBRTDQO\dgn (68))

```
1 use school
2 update worker set number='00007' where sex='F'
3 |
```

<

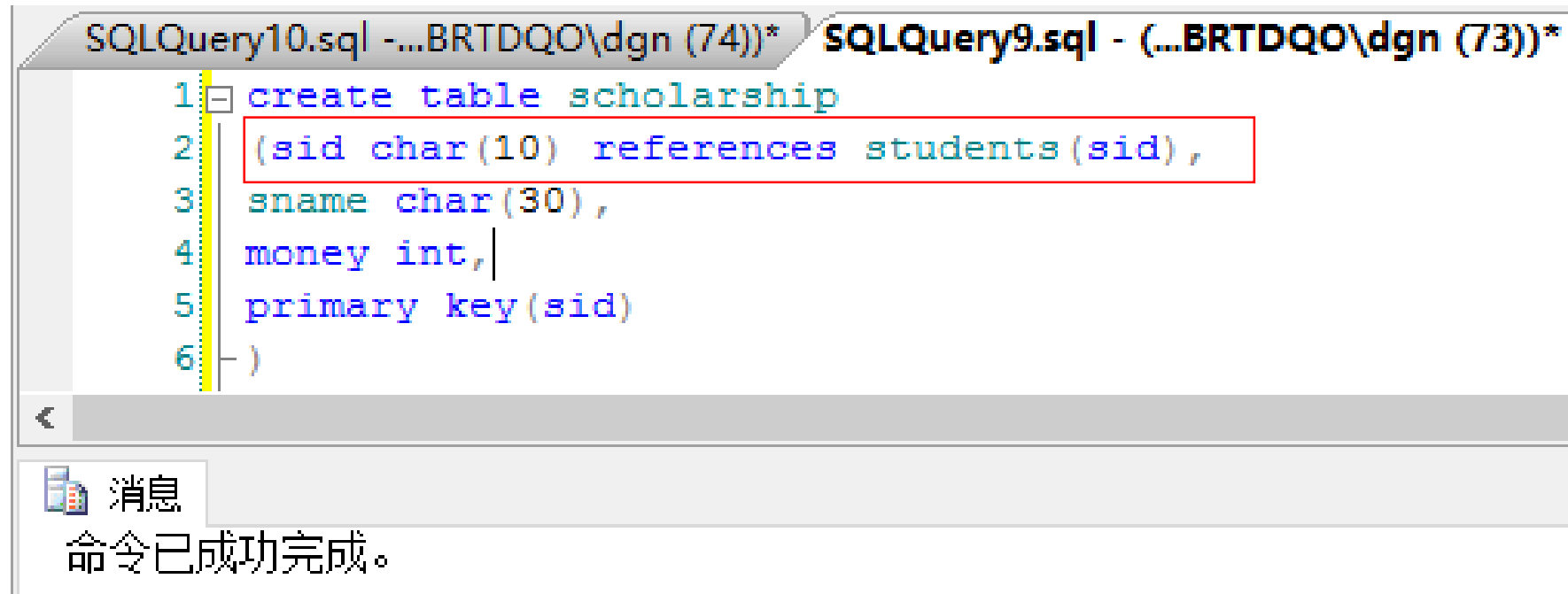
消息

Everynumber cannot be changed!  
消息 3609, 级别 16, 状态 1, 第 2 行  
事务在触发器中结束。批处理已中止。

- 实验示例

8.演示 INSTEAD OF 触发器使不可更新视图上支持更新。

1)首先创建scholarship表，其sid为参照表Students表的外键。



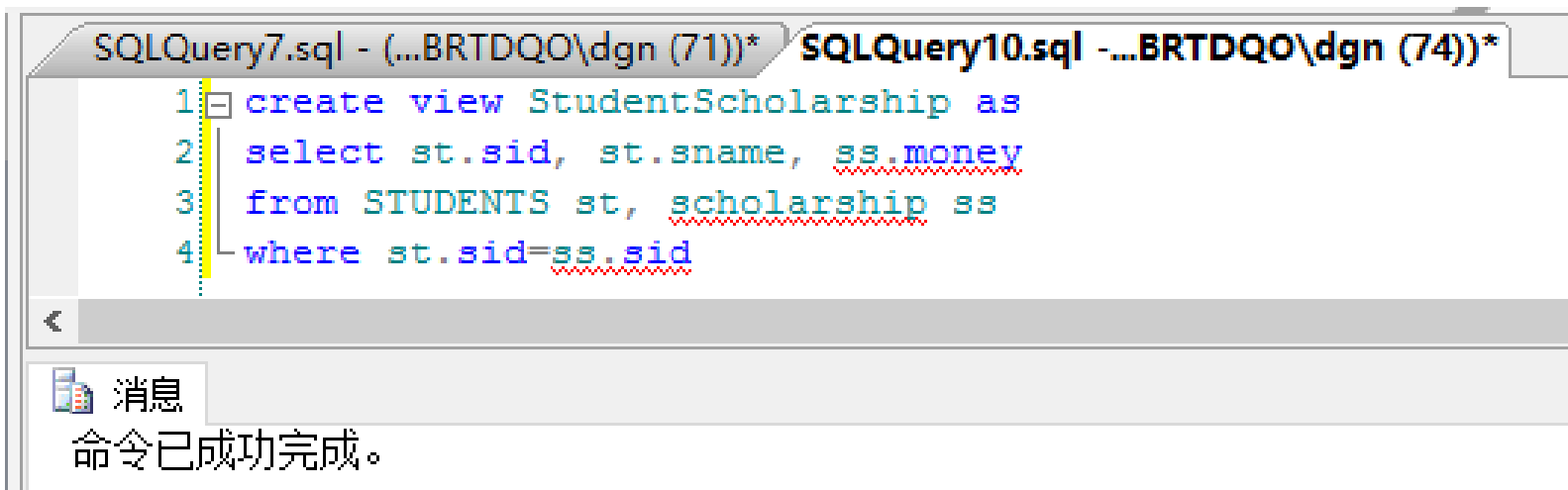
```
SQLQuery10.sql - ...BRTDQO\dgn (74))*  SQLQuery9.sql - (...BRTDQO\dgn (73))*  
1 create table scholarship  
2 (sid char(10) references students(sid),  
3  sname char(30),  
4  money int,  
5  primary key(sid)  
6 )
```

<

消息  
命令已成功完成。

## • 实验示例

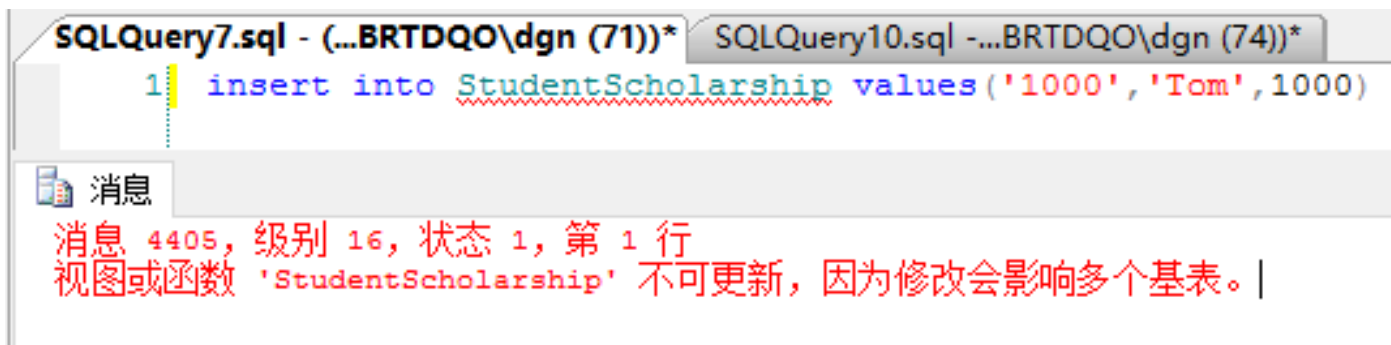
- 2) 创建一个不可更新的视图，该视图基于students和scholarship两个表。  
(如果视图中存在多个表的连接关系，则不可更新)。



```
SQLQuery7.sql - (...BRTDQO\dgn (71))* SQLQuery10.sql - ...BRTDQO\dgn (74))*
1 create view StudentScholarship as
2 select st.sid, st.sname, ss.money
3 from STUDENTS st, scholarship ss
4 where st.sid=ss.sid
```

消息  
命令已成功完成。

测试：在该视图中插入数据失败



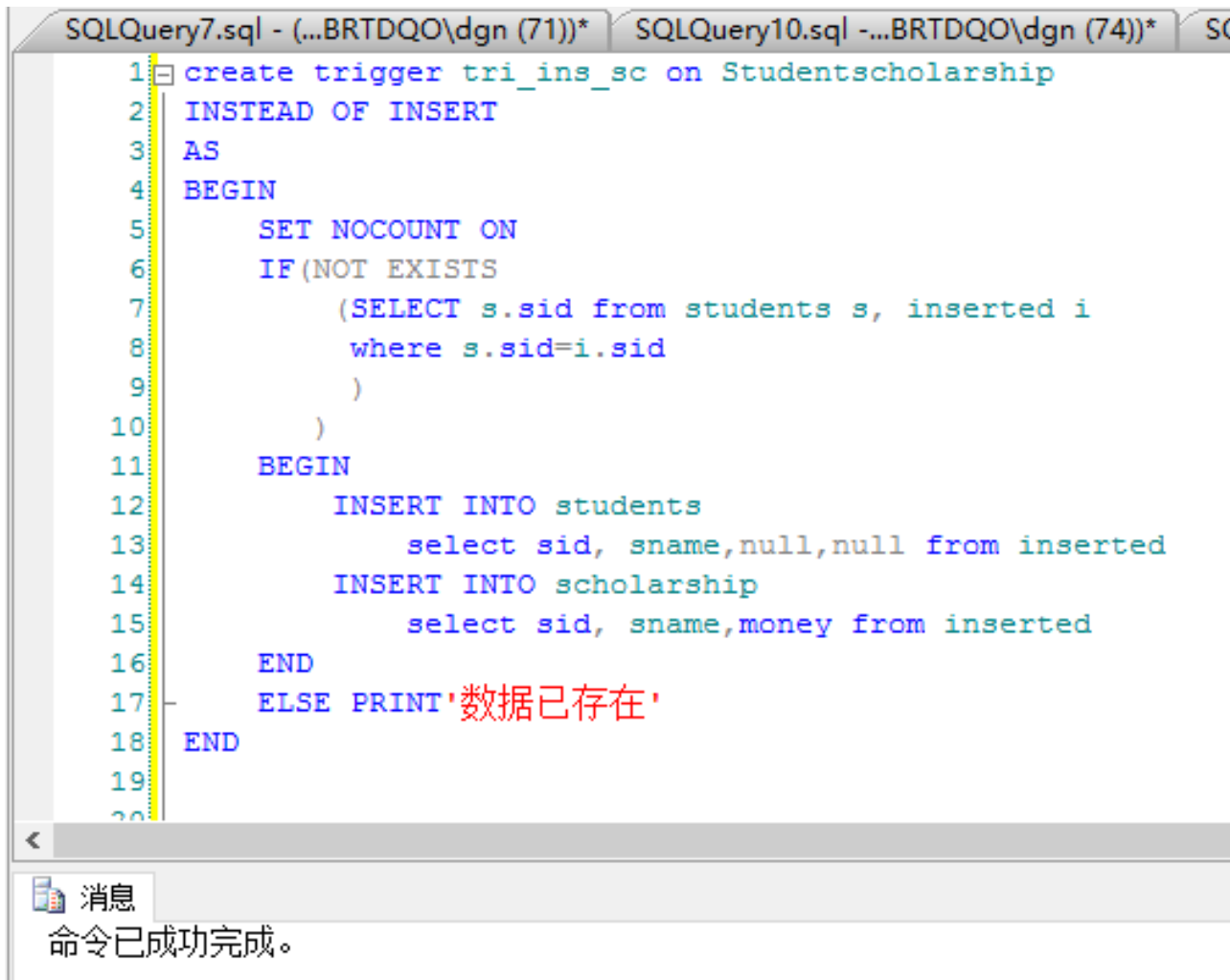
```
SQLQuery7.sql - (...BRTDQO\dgn (71))* SQLQuery10.sql - ...BRTDQO\dgn (74))*
1 insert into StudentScholarship values ('1000', 'Tom', 1000)
```

消息  
消息 4405, 级别 16, 状态 1, 第 1 行  
视图或函数 'StudentScholarship' 不可更新, 因为修改会影响多个基表。

## • 实验示例

3) 为了解决上述问题，创建一个INSTEAD OF 触发器：

【原理：触发器将insert语句进行分解，避免了一次对多个基表进行操作】

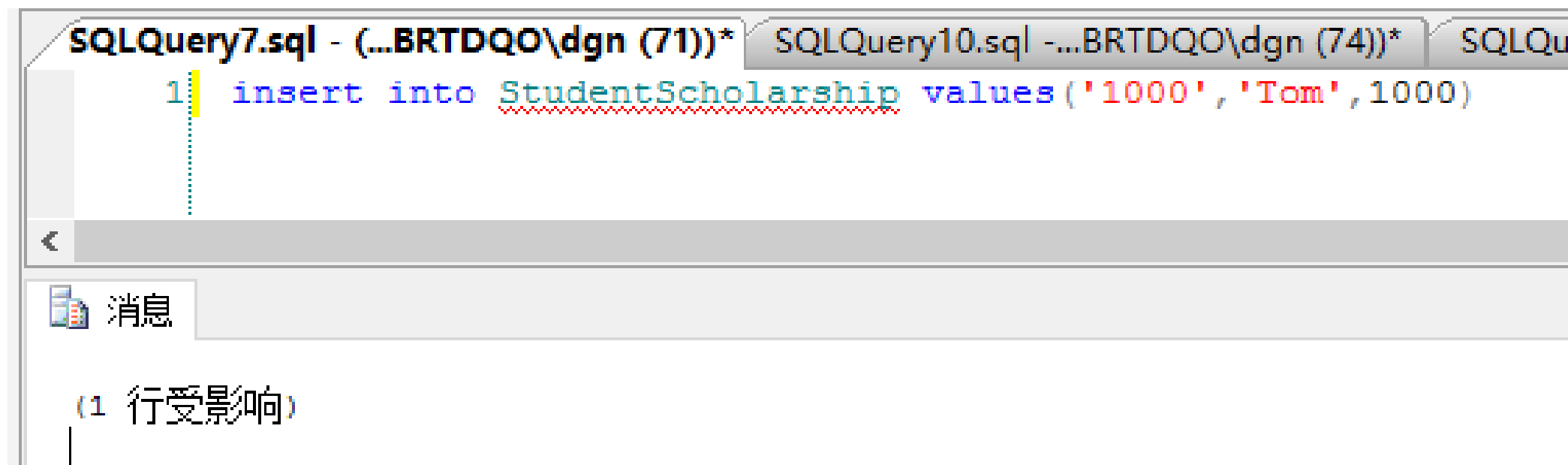


```
SQLQuery7.sql - (...BRTDQO\dgn (71))*  SQLQuery10.sql - ...BRTDQO\dgn (74))*  SC
1 create trigger tri_ins_sc on Studentscholarship
2 INSTEAD OF INSERT
3 AS
4 BEGIN
5     SET NOCOUNT ON
6     IF (NOT EXISTS
7         (SELECT s.sid from students s, inserted i
8          where s.sid=i.sid
9         )
10    )
11    BEGIN
12        INSERT INTO students
13            select sid, sname,null,null from inserted
14        INSERT INTO scholarship
15            select sid, sname,money from inserted
16    END
17    ELSE PRINT '数据已存在'
18 END
19
20
```

消息  
命令已成功完成。

- 实验示例

4) 创建INSTEAD OF 触发器后，验证视图支持插入数据。



创建 INSTEAD OF 触发器后，视图支持插入数据。

## • 练习

以下1-5练习均在worker表上进行（请按照示例中worker建表）

- (1) 建立一个在worker表上的触发器T4，要求插入记录的sage值必须比表中已记录的最大sage值大。
- (2) 演示违反触发器T4的操作，即插入一条比表中已记录的最大sage值小的记录。
- (3) 建立一个在worker表上的触发器T5，要求当更新一个记录的时候，表中记录的sage值要比老记录的sage值大，因为一般工资级别只能升不能降。
- (4) 演示违反触发器T5的操作。
- (5) 为worker表建立触发器T6，禁止修改编号为00001的记录。
- (6) 建立基于students和stu\_card两个表的视图，创建INSTEADOF触发器使该视图支持更新操作，并演示更新操作。（注：stu\_card的建表见PPT10参照完整性page10）