

Lecture 5, Fall 2017/2018

# 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

熟悉SQL的**数据更新操作**，

能够使用SQL语句对数据库进行**数据插入、更新、删除操作**。

# • 实验内容

在本次实验中,主要的内容是如何使用SQL语句对数据进行更新。

- 使用 **INSERT INTO**语句插入数据,包括**插入一个元组**或将**子查询的结果插入到数据库中**两种方式。
- 使用 **SELECT INTO**语句,产生一个新表并插入数据。
- 使用 **UPDATE语句**可以修改指定表中满足 WHERE子句条件的元组,有三种修改的方式:修改某一个元组的值,修改多个元组的值,带子查询的修改语句。
- 使用 **DELETE语句**删除数据:删除某一个元组的值,删除多个元组的值,带子查询的删除语句。

- 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。

## • 实验示例

1. 向students表中插入元祖 ( 学号 : 700045678 , 姓名 : LiMing , email : [LX@cdemg.com](mailto:LX@cdemg.com) , 年级 : 1992 )



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query text is as follows:

```
1 INSERT  
2 INTO STUDENTS  
3 VALUES ('700045678', 'LiMing', 'LX@cdemg.com', 1992)
```

Below the query window, a message box indicates the result of the execution:

消息  
(1 行受影响)

The status bar at the bottom of the window displays the following information:

✓ 查询已... | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:01 | 0 行

## • 实验示例

2.对每个课程，求学生的选课人数和学生的平均成绩，并把结果存入数据库。  
使用select into和insert into两种方式实现。

1) Insert into方式。因为数据库school中没有这个表，所以需要**先建一个表，再插入数据**。

先建新表

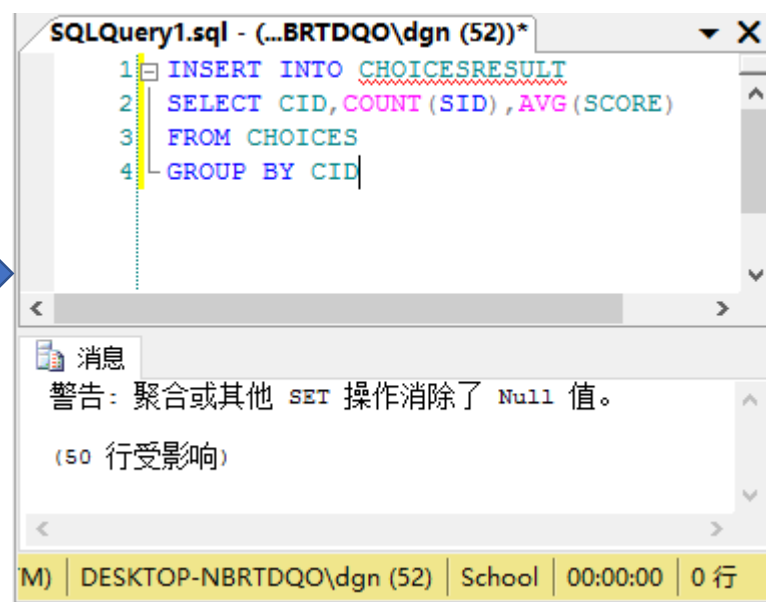


```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 CREATE TABLE CHOICESRESULT_new
2 (
3   CID CHAR(10),
4   STUDENTS SMALLINT,
5   AVGSCORE SMALLINT
6 )
```

消息  
命令已成功完成。

M) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行

再插入数据



```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 INSERT INTO CHOICESRESULT
2 SELECT CID, COUNT(SID), AVG(SCORE)
3 FROM CHOICES
4 GROUP BY CID
```

消息  
警告：聚合或其他 SET 操作消除了 Null 值。  
(50 行受影响)

M) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行

- 实验示例

2)Select into方式。Select into 与insert into 的区别在于：select into 可以同时建立一个新表，不需要像insert into那样需要先定义了表才能插入数据。

```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 SELECT CID AS CID,COUNT(SID) AS STUDENTS,AVG(SCORE) AS AVGSCORE
2 INTO CHOICESRESULT
3 FROM CHOICES
4 GROUP BY CID
```

消息

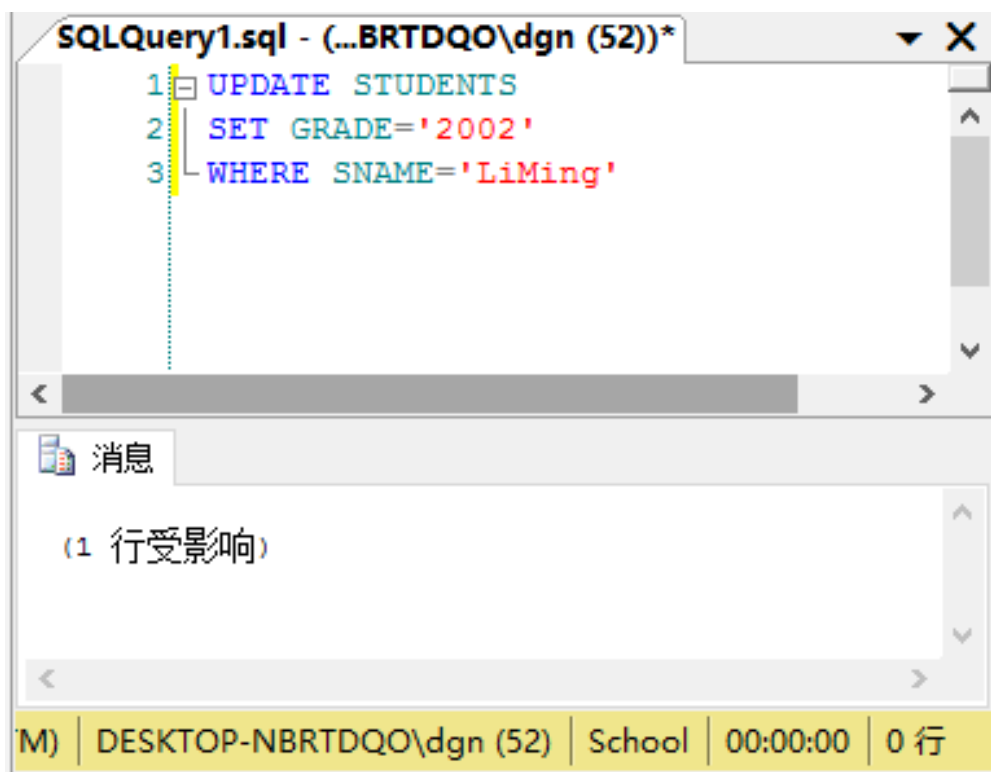
警告: 聚合或其他 SET 操作消除了 Null 值。  
(50 行受影响)

查询已成功执行。 (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (52) School 00:00:01 0 行



# 实验示例

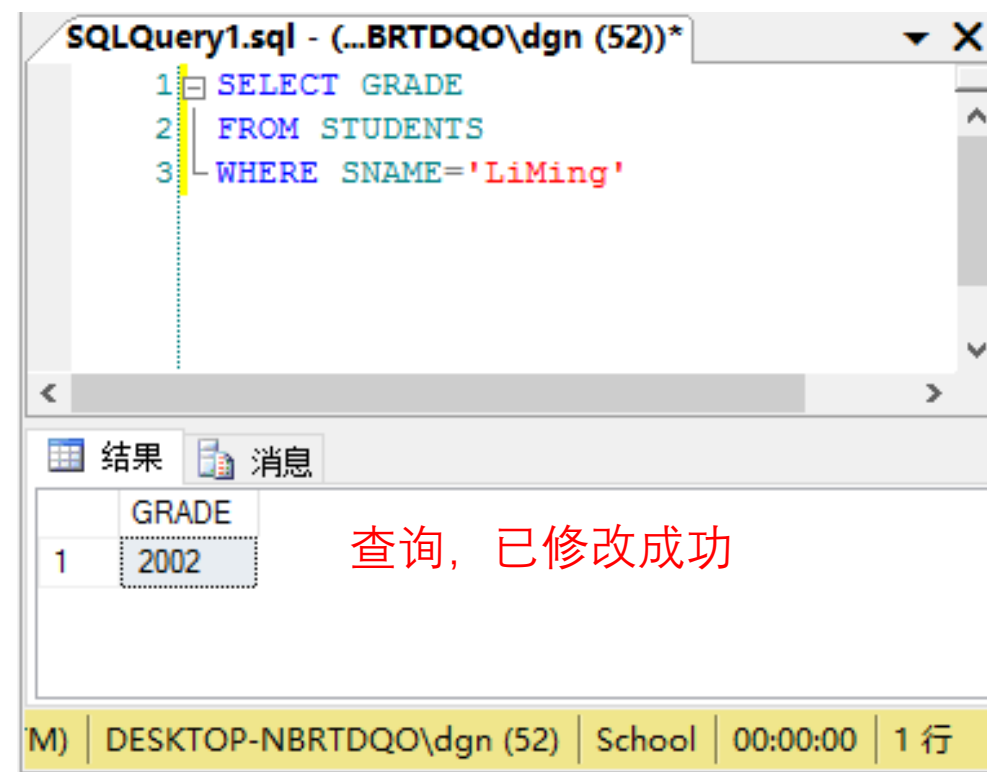
3. 在students表中，将姓名为“LiMing”的学生的年级改为“2002”。  
(修改指定元组，可以使用where子句来实现。)



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query text is:

```
1 UPDATE STUDENTS
2 SET GRADE='2002'
3 WHERE SNAME='LiMing'
```

Below the query window, the "消息" (Messages) tab is selected, showing the result: "(1 行受影响)" (1 row affected). The status bar at the bottom indicates "M) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行" (0 rows).



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))\*". The query text is:

```
1 SELECT GRADE
2 FROM STUDENTS
3 WHERE SNAME='LiMing'
```

Below the query window, the "结果" (Results) tab is selected, showing the result:

	GRADE
1	2002

Next to the result table, the text "查询，已修改成功" (Query, modified successfully) is displayed in red. The status bar at the bottom indicates "M) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 1 行" (1 row).

# 实验示例

4.在teachers表中，将所有教师的工资多加500元。

修改前

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT salary
2 FROM TEACHERS
3 order by salary DESC
```

结果 消息

	salary
1	4999
2	4999

KTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 15000 行

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

```
1 UPDATE TEACHERS
2 SET SALARY=SALARY+500
```

消息

(15000 行受影响)

DESKTOP-NBRTDQO\dgn (53) | School | 00:00:02 | 0 行

修改成功

SQLQuery1.sql - (...BRTDQO\dgn (52))\*

```
1 SELECT salary
2 FROM TEACHERS
3 order by salary DESC
```

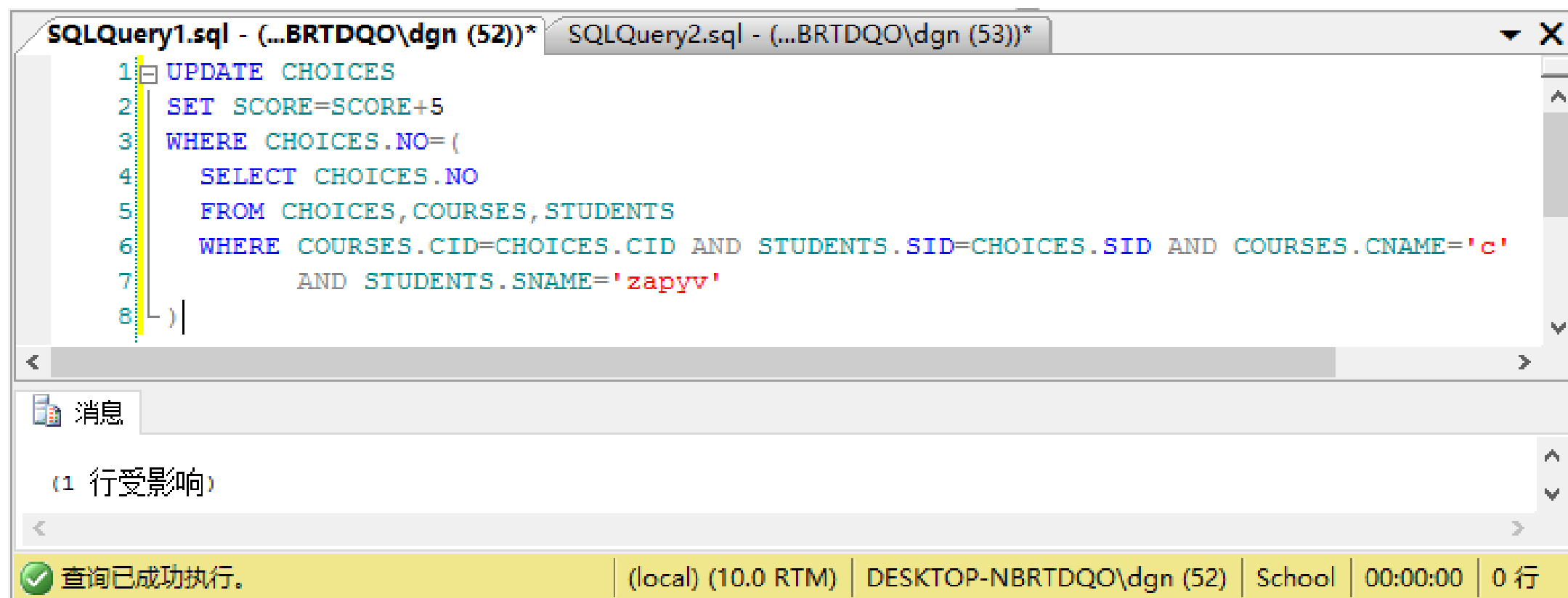
结果 消息

	salary
1	5499
2	5499

KTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 15000 行

## • 实验示例

5.将姓名为zapyv的学生的课程“C”的成绩加上5分。  
(要修改的元祖,要通过一个where子句来实现)



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

```
1 UPDATE CHOICES
2 SET SCORE=SCORE+5
3 WHERE CHOICES.NO=(
4     SELECT CHOICES.NO
5     FROM CHOICES,COURSES,STUDENTS
6     WHERE COURSES.CID=CHOICES.CID AND STUDENTS.SID=CHOICES.SID AND COURSES.CNAME='c'
7     AND STUDENTS.SNAME='zapyv'
8 )
```

Below the code editor, a message box indicates the execution result:

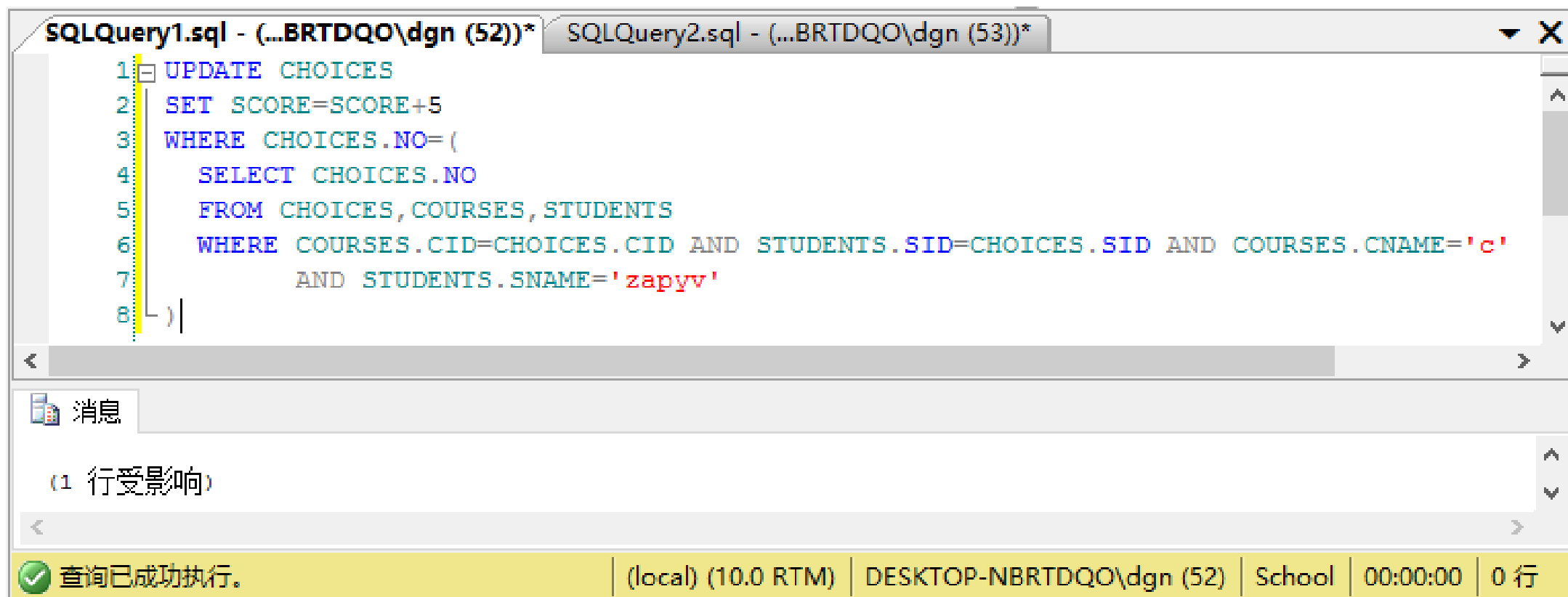
消息  
(1 行受影响)

The status bar at the bottom shows: 查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行

## • 实验示例

6.在students表中，删除姓名为“LiMing”的学生信息。

注意：使用delete语句删除元祖，必须在where子句中明确指出删除对象，如果省略where子句，则使整张表数据都删除！使用delete语句时，一定写好where子句！)



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

```
1 UPDATE CHOICES
2 SET SCORE=SCORE+5
3 WHERE CHOICES.NO=(
4     SELECT CHOICES.NO
5     FROM CHOICES,COURSES,STUDENTS
6     WHERE COURSES.CID=CHOICES.CID AND STUDENTS.SID=CHOICES.SID AND COURSES.CNAME='c'
7     AND STUDENTS.SNAME='zapyv'
8 )
```

Below the code editor, a message box indicates the execution result:

消息

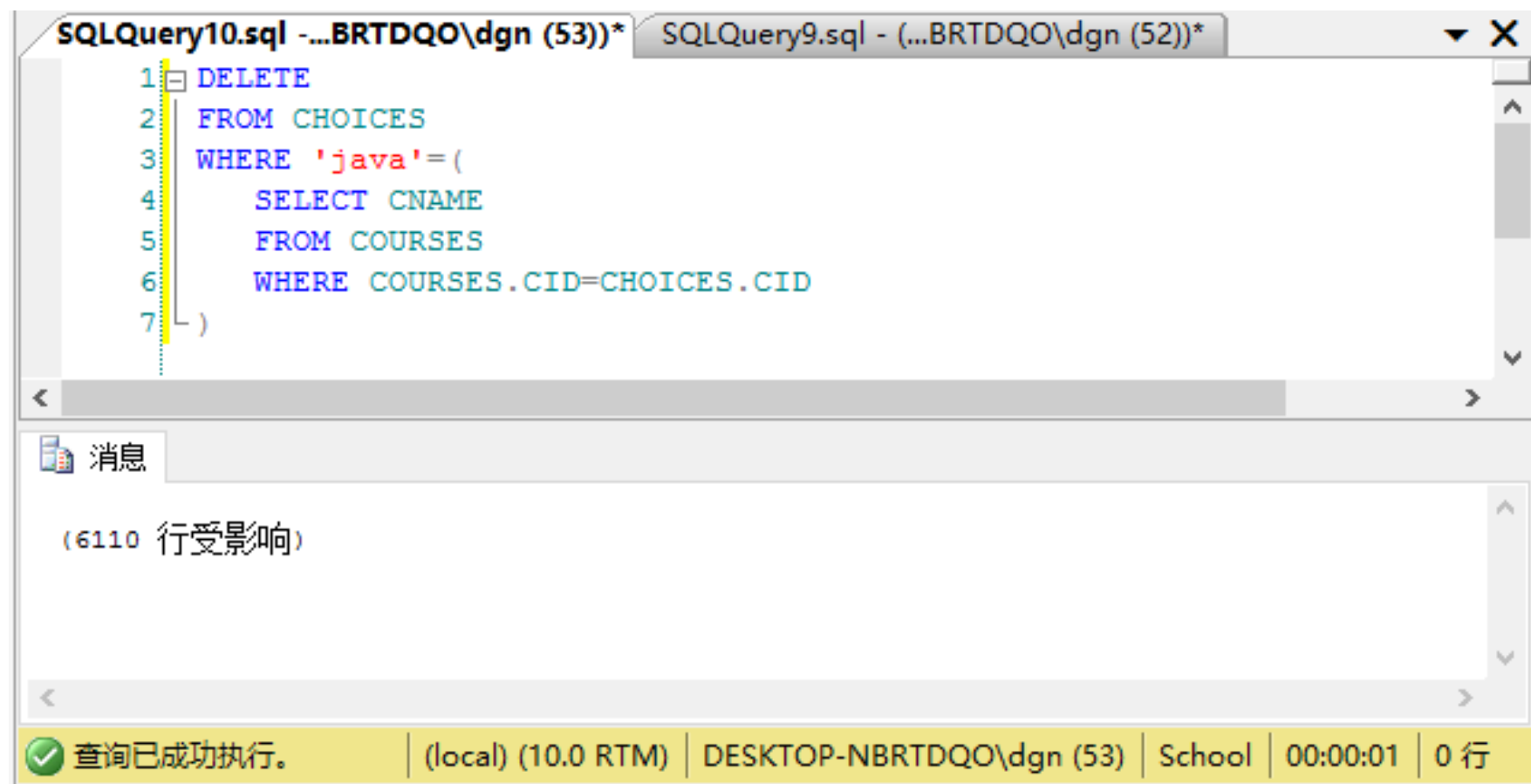
(1 行受影响)

At the bottom, a status bar shows the execution status and details:

✓ 查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行

# • 实验示例

7. 删除所有选修课程 “java” 的选课记录。



The screenshot shows a SQL query execution window with two tabs: 'SQLQuery10.sql - (...BRTDQO\dgn (53))\*' and 'SQLQuery9.sql - (...BRTDQO\dgn (52))\*'. The active tab displays a SQL query:

```
1 DELETE
2 FROM CHOICES
3 WHERE 'java'=(
4     SELECT CNAME
5     FROM COURSES
6     WHERE COURSES.CID=CHOICES.CID
7 )
```

Below the query editor, a message box titled '消息' (Message) displays the result: '(6110 行受影响)' (6110 rows affected).

The status bar at the bottom indicates: '查询已成功执行。' (Query executed successfully.) | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:01 | 0 行

# • 实验示例

8.在courses表中删去课时<48的操作。



报错原因：delete不满足表定义的约束条件。

因为COURSES表中cid属性，在CHOICES表中作为外键存在，所以存在约束“FK\_CHOICES\_COURSES”。所以该约束限制了对表COURSES的删除操作。

## • 练习

- (1)向 STUDENTS表插入编号是 “800022222” 且姓名是 “ WangLan” 的元组。
- (2)向 TEACHERS表插入元组( “200001000” , “LXL” , “s4zrck@pew.net” , “3024” )。
- (3)将 TEACHERS表中编号为 “200010493” 的老师工资改为4000。
- (4)将 TEACHERS表中所有工资小于2500的老师工资改为2500。
- (5)将编号为 “200016731” 的老师讲授的课程全部改成由姓名为 “rnupx” 的老师讲授。
- (6)更新编号 “800071780” 的学生年级为 “2001” 。
- (7)删除没有学生选修的课程。
- (8)删除年级高于1998的学生信息。
- (9)删除没有选修课程的学生信息。
- (10)删除成绩不及格的选课记录 。