

Lecture 6, 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**有关视图的操作**，

能够使用SQL语句**创建需要的视图，对视图进行查询和取消视图。**

• 实验内容

1. 定义常见的视图形式，包括：
 - 行列子集视图
 - WITH CHECK OPTION的视图
 - 基于多个基表的视图
 - 基于视图的视图
 - 带表达式的视图
 - 分组视图
2. 考察 WITH CHECK OPTION这一语句在视图定义后产生的影响，包括对修改操作、删除操作、插入操作的影响
3. 讨论视图的数据更新情况，对子行列视图进行数据更新。
4. 使用DROP语句删除一个视图，由该视图导出的其他视图定义仍在数据字典中，但已不能使用，必须显式删除。同样的原因，删除基表时，由该基表导出的所有视图定义都必须显式删除。

- 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

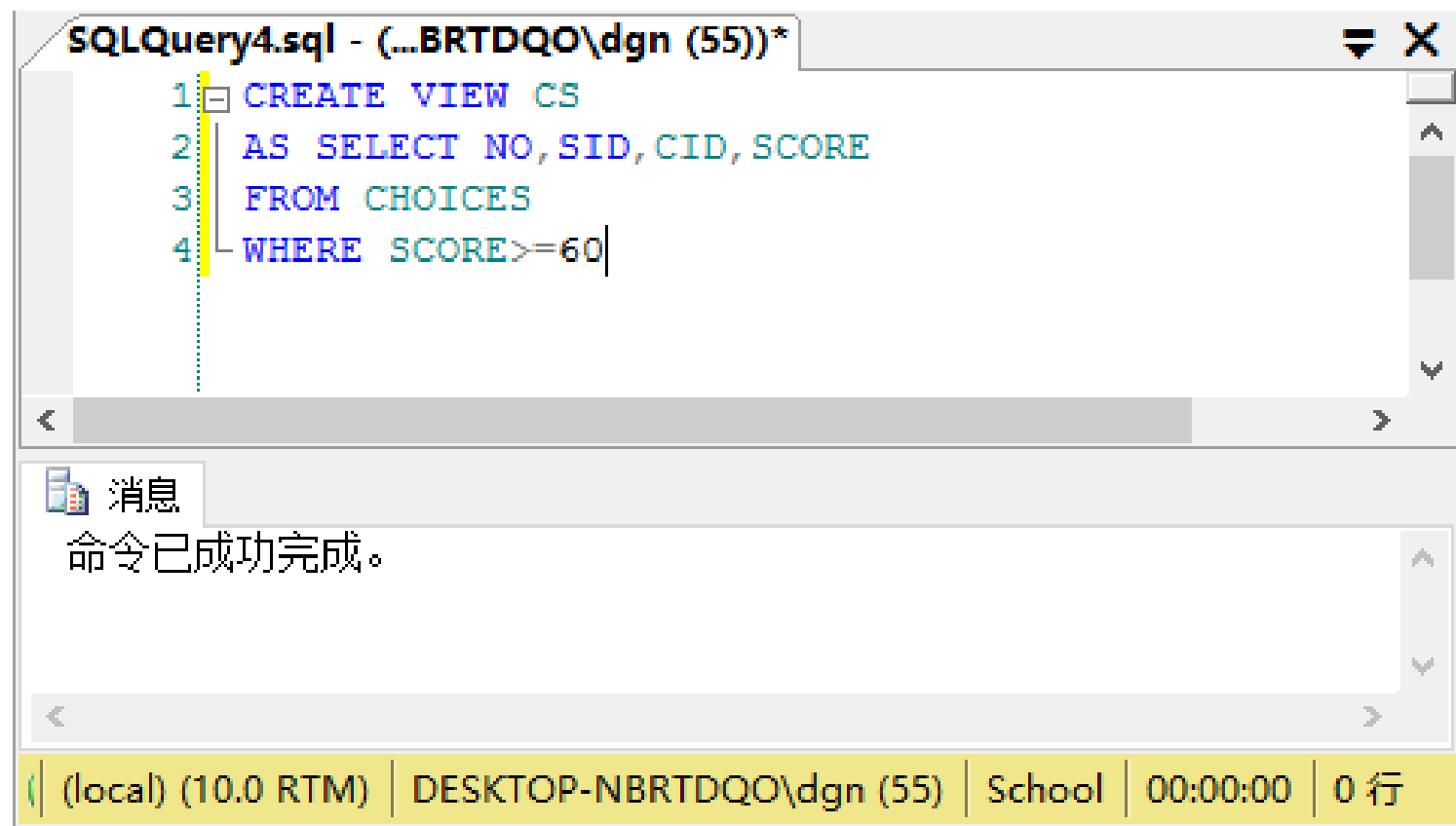
- students (sid, sname , email , grade)
- teachers (tid, tname , email , salary)
- courses (cid, cname , hour)
- choices (no, sid , tid , cid , score)

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. 创建一个行列子集视图CS，给出选课成绩合格的学生的编号，所选课程号和该课程成绩。



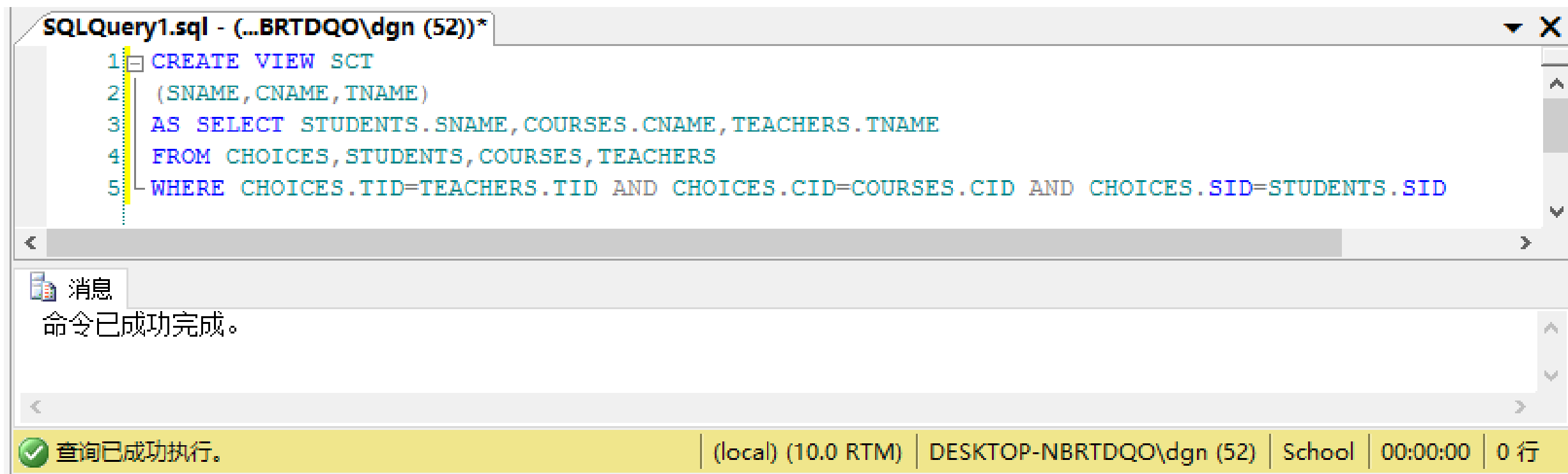
The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window titled "SQLQuery4.sql - (...BRTDQO\dgn (55))*" containing the following SQL code:

```
1 CREATE VIEW CS
2 AS SELECT NO, SID, CID, SCORE
3 FROM CHOICES
4 WHERE SCORE >= 60
```

The bottom pane shows a message box with the text "命令已成功完成。" (Command completed successfully.). The status bar at the bottom indicates the connection is to "(local) (10.0 RTM)", the database is "DESKTOP-NBRTDQO\dgn (55)", the schema is "School", the execution time is "00:00:00", and the result set contains "0 行" (0 rows).

- 实验示例

2.创建基于多个基表的视图SCT，这个视图由学生姓名和其所选修的课程名及该讲授该课的教师姓名构成。



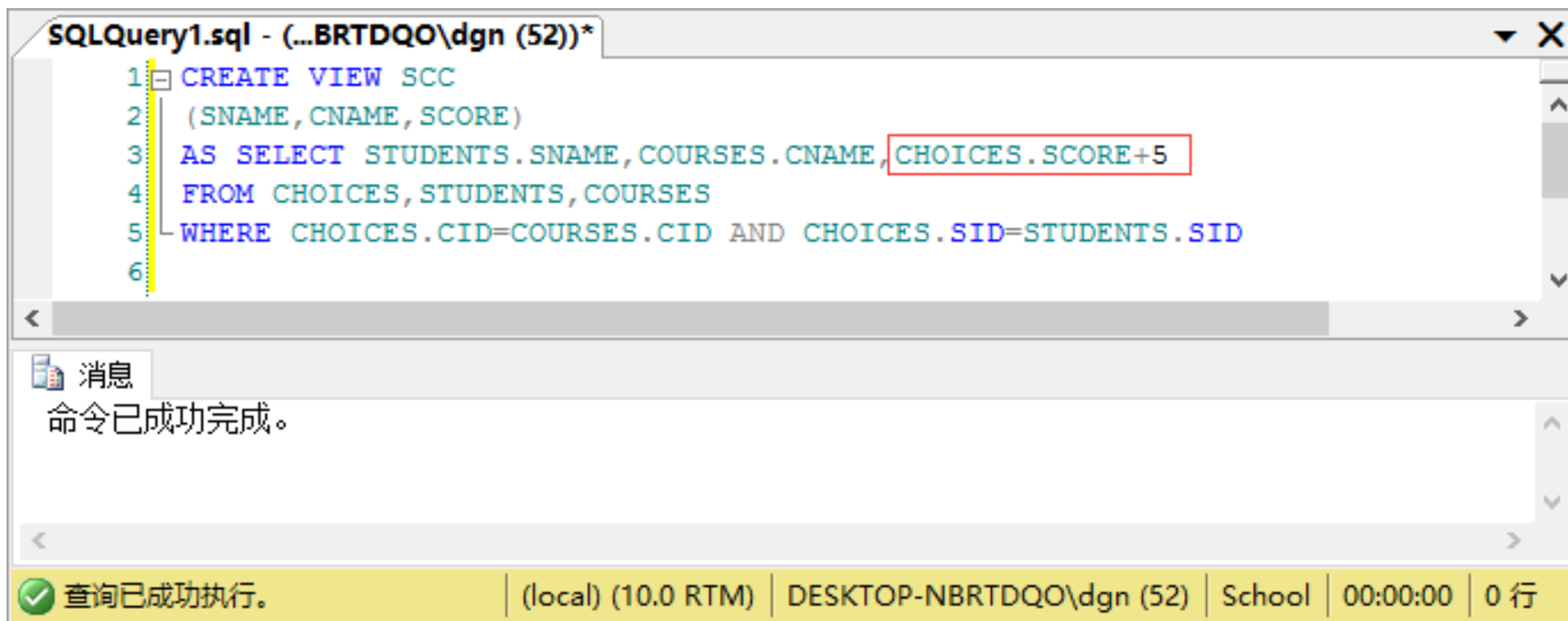
```
SQLQuery1.sql - (...BRTDQO\dgn (52))*  
1 CREATE VIEW SCT  
2 (SNAME, CNAME, TNAME)  
3 AS SELECT STUDENTS.SNAME, COURSES.CNAME, TEACHERS.TNAME  
4 FROM CHOICES, STUDENTS, COURSES, TEACHERS  
5 WHERE CHOICES.TID=TEACHERS.TID AND CHOICES.CID=COURSES.CID AND CHOICES.SID=STUDENTS.SID
```

消息
命令已成功完成。

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

3.创建带表达式的视图SCC，由学生姓名、选课名称，和所有课程成绩比原来多5分这几个属性构成。



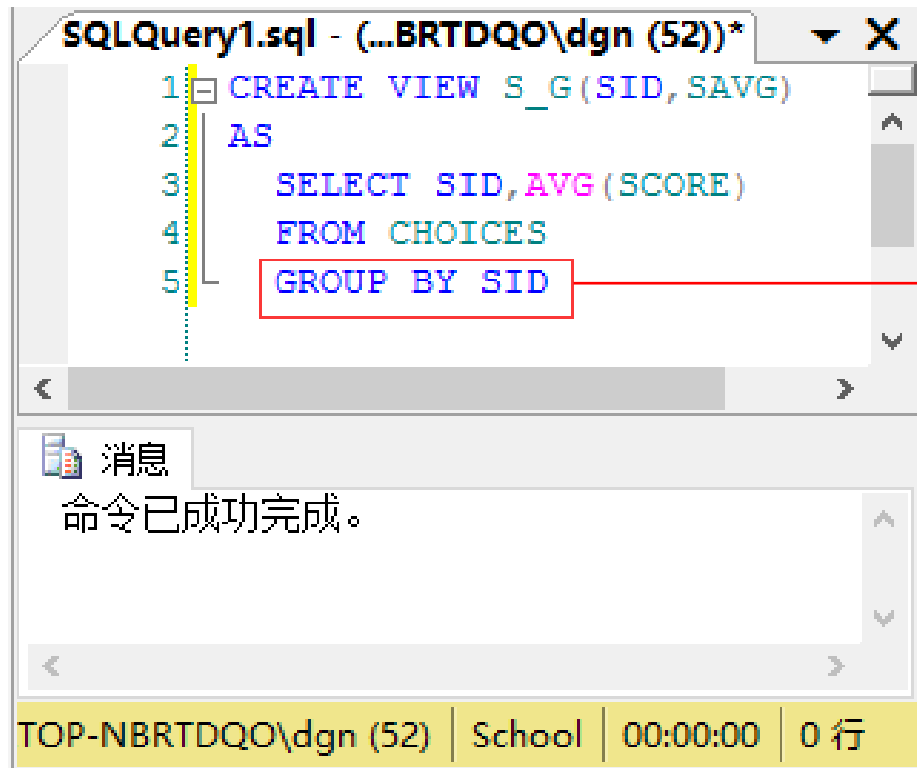
```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 CREATE VIEW SCC
2 (SNAME, CNAME, SCORE)
3 AS SELECT STUDENTS.SNAME, COURSES.CNAME, CHOICES.SCORE+5
4 FROM CHOICES, STUDENTS, COURSES
5 WHERE CHOICES.CID=COURSES.CID AND CHOICES.SID=STUDENTS.SID
6
```

消息
命令已成功完成。

✓ 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

4.创建分组视图S_G，将学生学号以其平均成绩定义为一个视图。



```
SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 CREATE VIEW S_G (SID, SAVG)
2 AS
3     SELECT SID, AVG(SCORE)
4     FROM CHOICES
5     GROUP BY SID
```

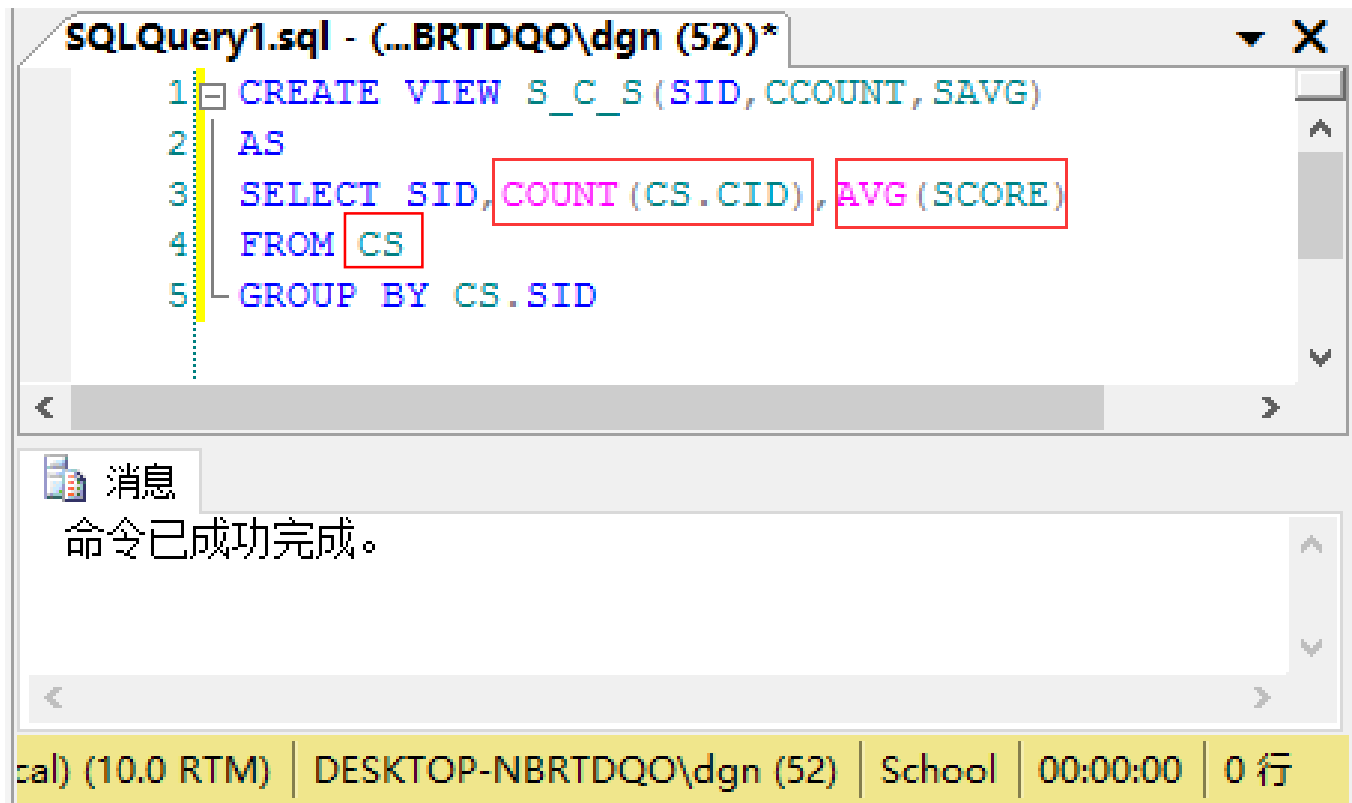
消息
命令已成功完成。

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

注意：由于属性SAVG是由**分组统计**得到的，因此这样的视图是**不允许更新的**。

• 实验示例

5. 创建一个**基于视图的视图**S_C_S，基于（1）中建立的视图，定义一个包括学生编号，学生所选课程数目和平均成绩的视图。



The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))*". The editor contains the following SQL code:

```
1 CREATE VIEW S_C_S (SID, CCOUNT, SAVG)
2 AS
3 SELECT SID, COUNT (CS.CID), AVG (SCORE)
4 FROM CS
5 GROUP BY CS.SID
```

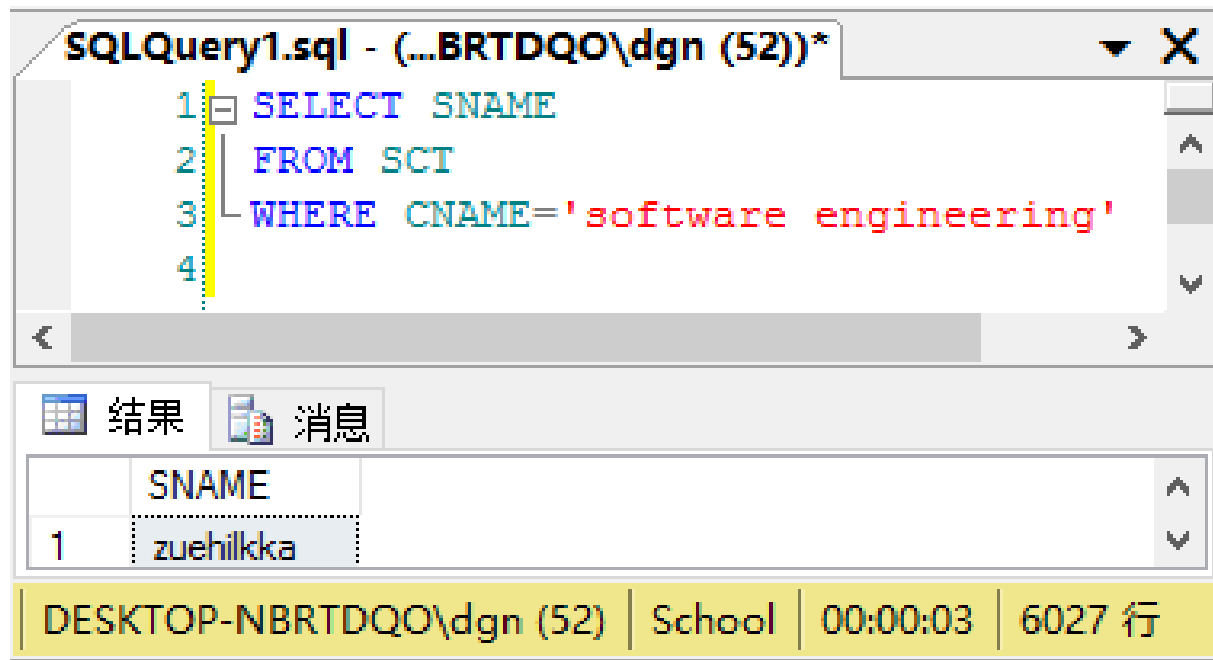
The code is displayed in a monospaced font with syntax highlighting. The table name "CS" in the FROM clause is highlighted with a red box. The function "COUNT" and its argument "CS.CID" are also highlighted with a red box. The function "AVG" and its argument "SCORE" are highlighted with a red box. The table name "S_C_S" in the CREATE VIEW statement is highlighted with a red box. The column names "SID", "CCOUNT", and "SAVG" are also highlighted with a red box.

Below the editor, a message box titled "消息" (Message) displays the text "命令已成功完成。" (Command completed successfully.).

The status bar at the bottom of the window shows the following information: "cal) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行" (cal) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 rows).

- 实验示例

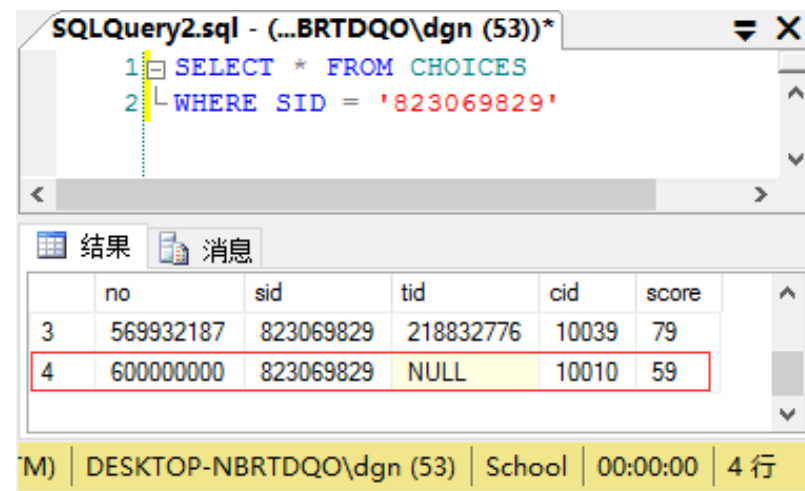
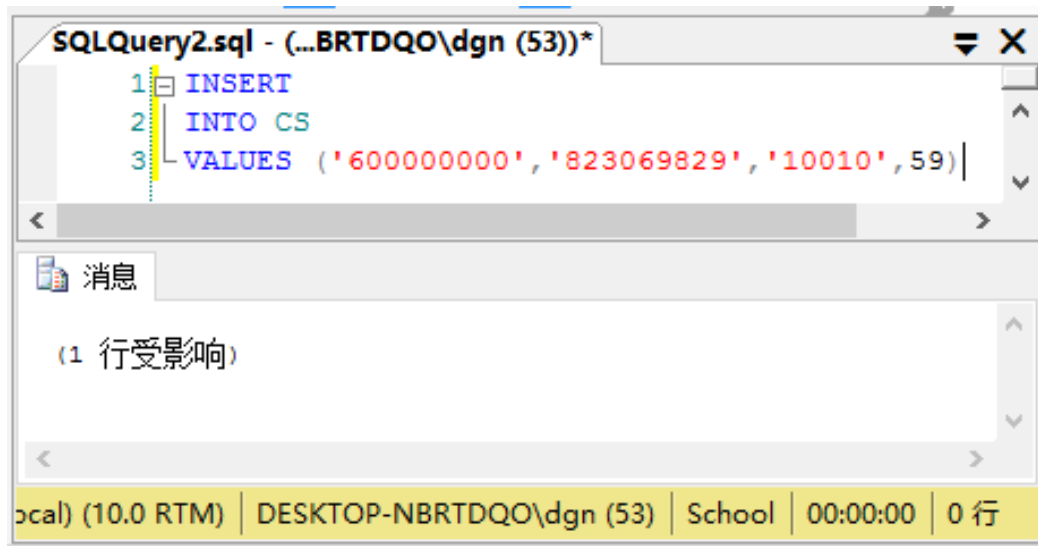
6.查询所有选修课程Software Engineering的学生姓名。



注意：视图的查询操作与基本表查询操作相同。
但由于视图本身已经是由单表或多表映射而成的虚表，所以**一般不再将视图与其他表做连接查询。**

• 实验示例

7.插入元组 ('600000000' , '823069829' , '10010' ,59) 到视图CS中。
若视图CS定义中存在WITH CHECK OPTION , 会对插入操作有何影响？

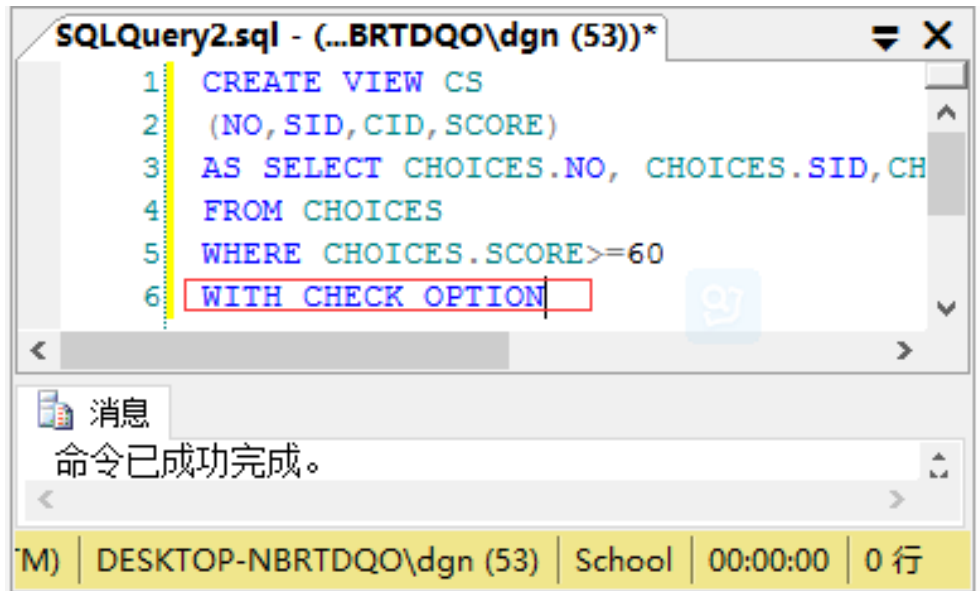


视图CS中没有WITH CHECK OPTION时，插入成功。

并且在基本表CHOICES中也相应插入该元组('600000000','823069829' , NULL , '10010',59)

• 实验示例

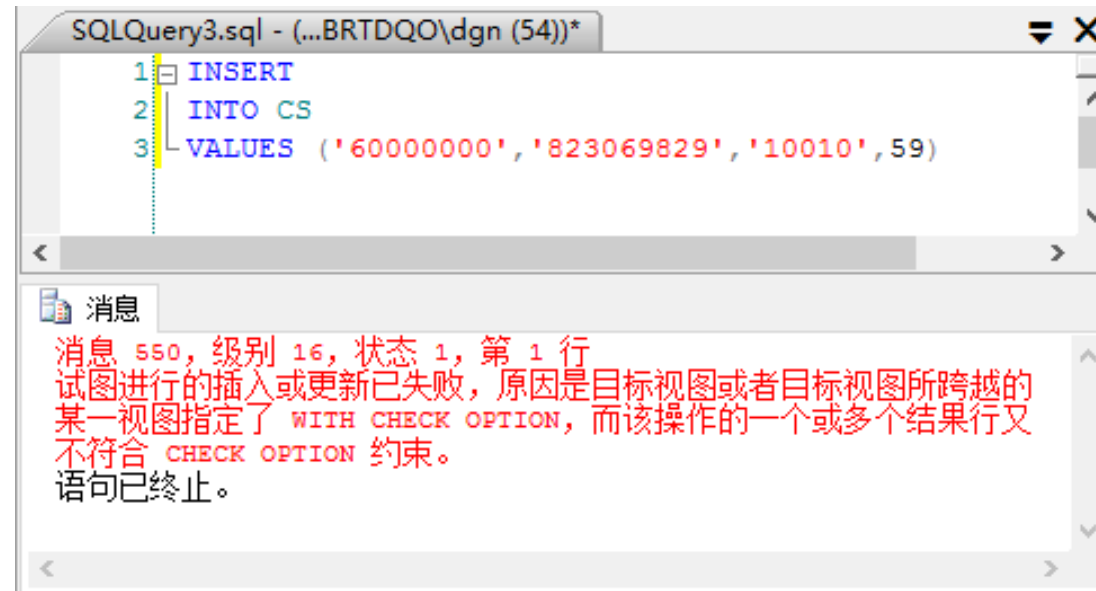
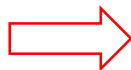
若视图CS定义中存在WITH CHECK OPTION，则数据插入失败。



```
SQLQuery2.sql - (...BRTDQO\dgn (53))*  
1 CREATE VIEW CS  
2 (NO, SID, CID, SCORE)  
3 AS SELECT CHOICES.NO, CHOICES.SID, CH  
4 FROM CHOICES  
5 WHERE CHOICES.SCORE >= 60  
6 WITH CHECK OPTION
```

消息
命令已成功完成。

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



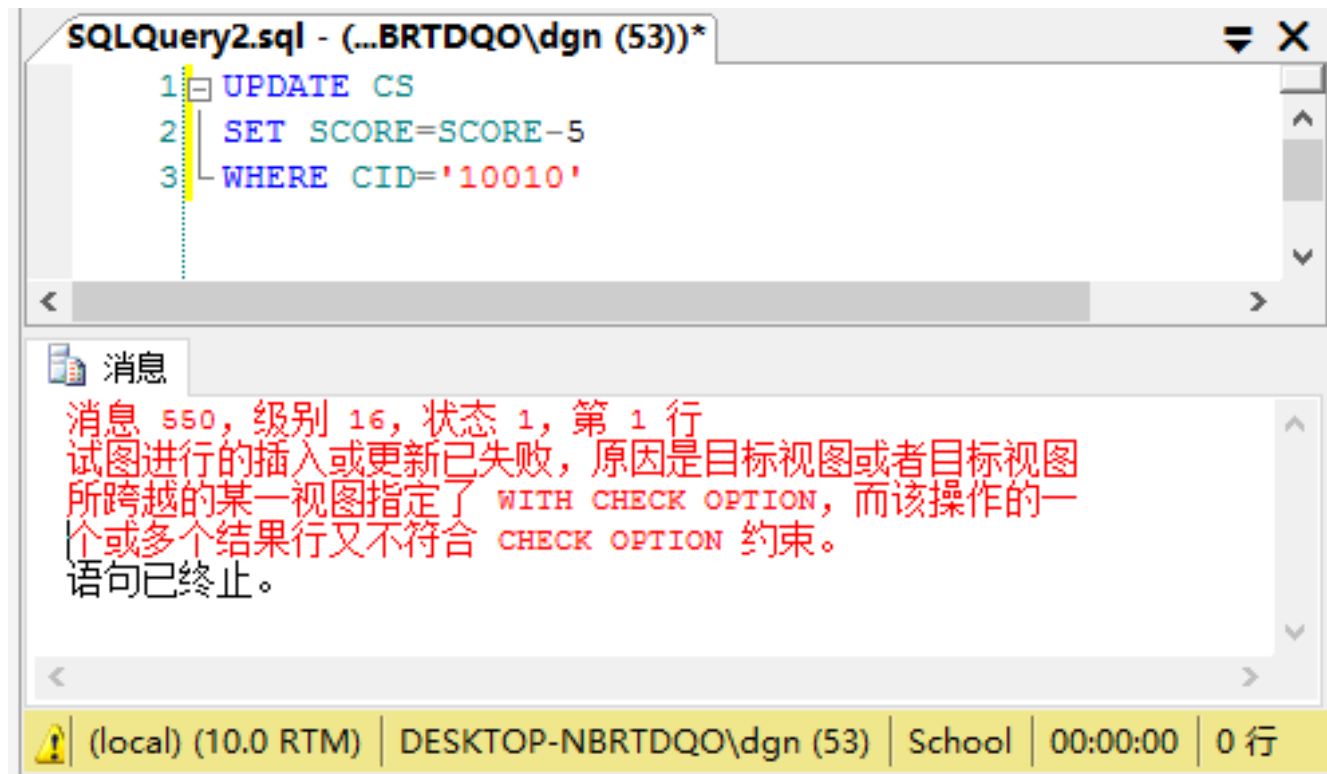
```
SQLQuery3.sql - (...BRTDQO\dgn (54))*  
1 INSERT  
2 INTO CS  
3 VALUES ('60000000', '823069829', '10010', 59)
```

消息
消息 550, 级别 16, 状态 1, 第 1 行
试图进行的插入或更新已失败, 原因是目标视图或者目标视图所跨越的
某一视图指定了 WITH CHECK OPTION, 而该操作的一个或多个结果行又
不符合 CHECK OPTION 约束。
语句已终止。

数据插入失败的原因：**WITH CHECK OPTION会对插入或更新操作进行检查**，由于新插入操作 score=59，不满足视图CS定义 “score>60” 的条件，所以数据插入失败。

• 实验示例

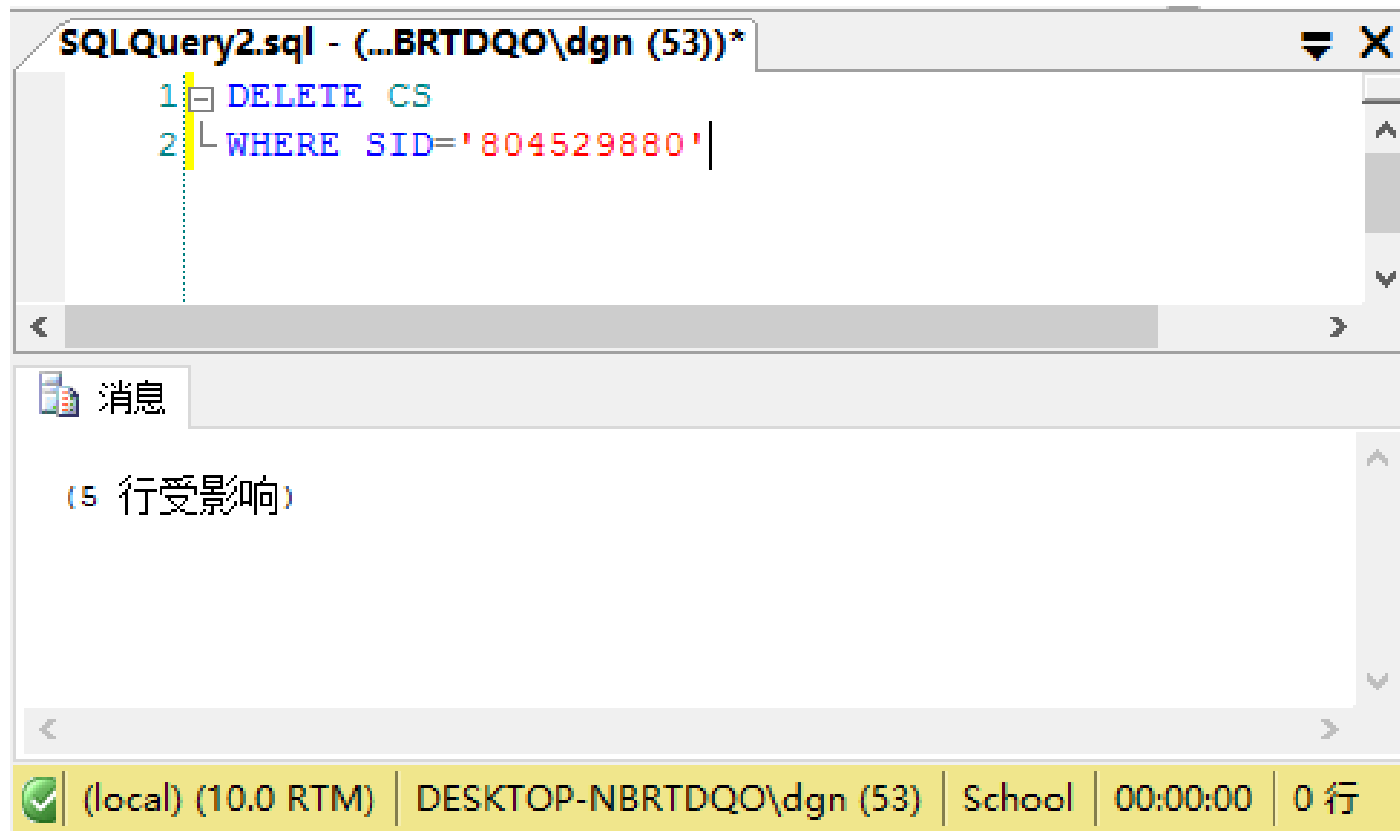
8.将视图CS(包含定义WITH CHECK OPTION) 中所有课程编号为10010的课程成绩都减五分，是否可以正确执行？



因为有些score减5分后，会<60，不满足WITH CHECK OPTION的条件。所以数据更新失败。

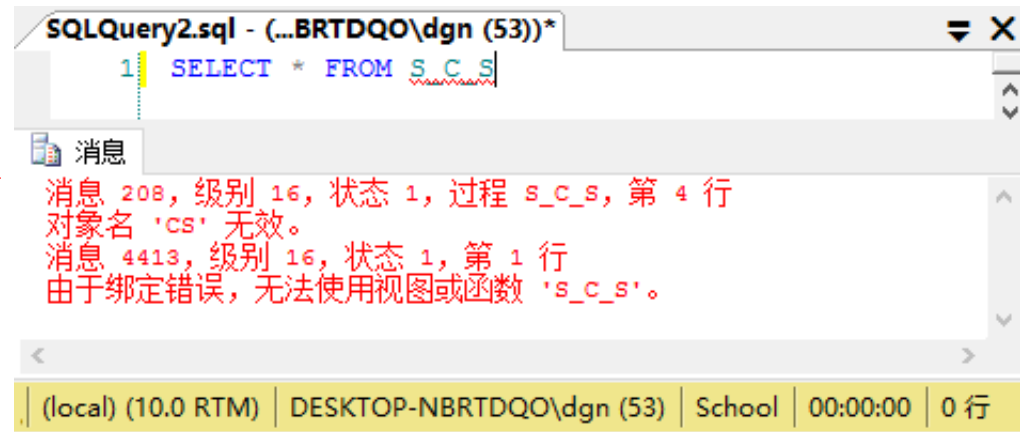
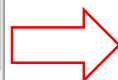
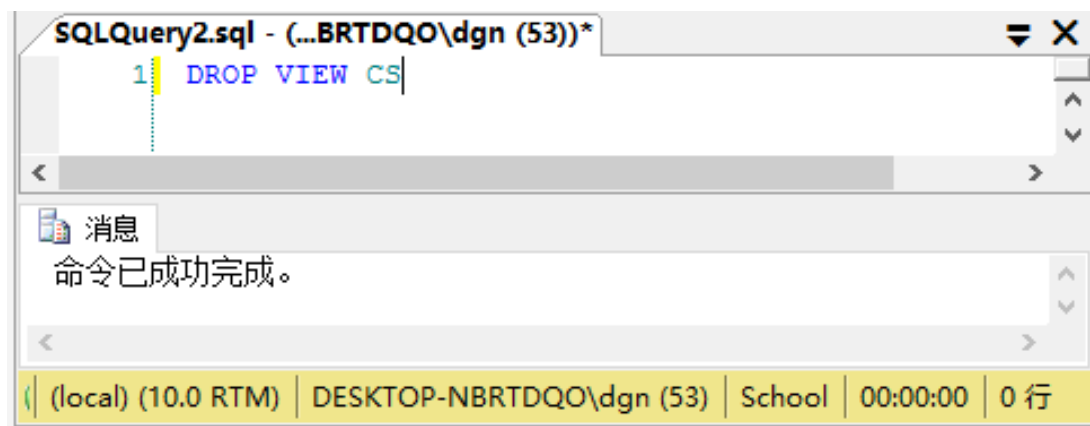
- 实验示例

9.在视图CS(包含WITH CHECK OPTION)中删除编号为 '804529880 '的学生记录。



• 实验示例

10.取消视图视图CS



注意：视图CS取消，那么由视图CS导出的视图S_C_S定义虽然还保留在数据字典里，但S_C_S已失效，需要drop view S_C_S显示删除掉。

• 练习

- (1)定义选课信息和课程名称的视图 VIEWC;
- (2)定义学生姓名与选课信息的视图 VIEWS;
- (3)定义年级低于1998的学生的视图S1(SID, SNAME, GRADE);
- (4)查询学生为 “ uxjof” 的学生的选课信息;
- (5)查询选修课程 “UML” 的学生的编号和成绩;
- (6)向视图S1插入记录(60000001,Lily,2001);
- (7)定义包括更新和插入约束的视图S1,尝试向视图插入记录(60000001,Lily,1997),删除所有年级为1999的学生记录,讨论更新和插入约束带来的影响。
- (8)在视图 VIEWS中将姓名为 “ uxjof” 的学生的选课成绩都加上5分。
- (9)取消以上建立的所有视图。