数据库原理及应用实验报告

班级:软工1706 姓名：叶倩琳 学号：201706061330

# 实验7、实体完整性

****7.1 试验目的****

学习实体完整性的建立，以及实践违反实体完整性的结果。

****7.2 试验内容****

（1）   在数据库University\_Mis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。

（2）   演示违反实体完整性的插入操作。

（3）   演示违反实体完整性的更新操作。

（4）   演示事务的处理，包括事务的建立、处理以及出错时的事务回滚。

（5）   通过建立University\_Misarship表，插入数据，演示当与现有的数据环境不等时，无法建立实体完整性以及参照完整性。

****7.3 实验步骤****

以系统管理员或sa用户登录进入SSMS，在SSMS中新建查询窗口中输入如下命令，运行并观察和记录结果。

（1）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

CREATE TABLE Stu\_Union(Sno CHAR(8) NOT NULL UNIQUE,

                   Sname CHAR(8),

                   Ssex CHAR(1),

                   Sage INT,

                   Sdept CHAR(20),

                   CONSTRAINT PK\_Stu\_Union PRIMARY KEY(Sno));

INSERT Stu\_Union VALUES('S01','王兵','M',23,'CS');

UPDATE Stu\_Union SET Sno=' ' WHERE Sdept='CS';

UPDATE Stu\_Union SET Sno='S02' WHERE Sname='王兵';

SELECT \* FROM Stu\_union;

（2）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

INSERT Stu\_Union VALUES (‘S02’,’黄山’,’M’,23,’CS’);

（3）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

UPDATE Stu\_Union SET Sno =NULL WHERE Sno='S02';

（4）输入并执行如下SQL命令

①   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SET XACT\_ABORT ON

BEGIN TRANSACTION T1

INSERT INTO Stu\_union VALUES(‘S09’,’李永’,’M’,25,’EE’);

INSERT INTO Stu\_union VALUES (‘S03’,’黄浩’,’F’,25,’EE’);

INSERT INTO Stu\_union VALUES (‘S05’,’黄浩’,’F’,25,’EE’);

SELECT \* FROM Stu\_union;

COMMIT TRANSACTION T1

②   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SET  XACT\_ABORT ON

BEGIN TRANSACTION  T2

INSERT INTO Stu\_union VALUES (‘S07’,’李宁’,’M’,25,’EE’);

SELECT \* FROM Stu\_union;

INSERT INTO Stu\_union VALUES (‘S09’,’李靖’,’F’,22,’CS’);

COMMIT TRANSACTION T2

③   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SELECT \* FROM Stu\_union;

（5）输入并执行如下SQL命令

①   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

CREATE TABLE Scholarship

(

M\_ID VARCHAR(10), Stu\_id CHAR(8),R\_Money INT

)

INSERT INTO Scholarship VALUES(‘M01’,’ S07’,5000)

INSERT INTO Scholarship VALUES (‘M01’,’ S08’,8000)

SELECT \* FROM Scholarship

②   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

ALTER TABLE Scholarship ADD

CONSTRAINT PK\_Scholarship PRIMARY KEY(M\_ID)

③   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

ALTER TABLE Scholarship ADD

CONSTRAINT FK\_Scholarship FOREIGN KEY(Stu\_id) REFERENCES Students(Sno)

1.1  实验要求

(1) 实验之前请细细阅读实验总体要求与说明指导书

(2 ) 在SQL Server 2005或2008的SSMS环境中，完成以上实验(1)-(9)步所有SQL数据定义操作，并将其奇数步骤操作窗口剪贴到实验报告中。

****7.4实验结果****

（1）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

CREATE TABLE Stu\_Union(Sno CHAR(8) NOT NULL UNIQUE,

                   Sname CHAR(8),

                   Ssex CHAR(1),

                   Sage INT,

                   Sdept CHAR(20),

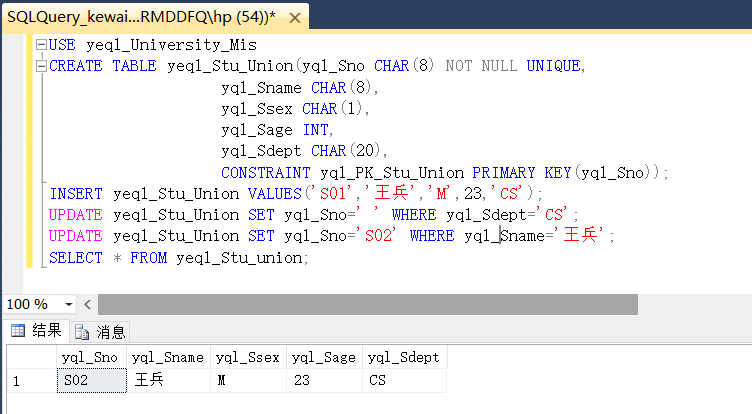
                   CONSTRAINT PK\_Stu\_Union PRIMARY KEY(Sno));

INSERT Stu\_Union VALUES('S01','王兵','M',23,'CS');

UPDATE Stu\_Union SET Sno=' ' WHERE Sdept='CS';

UPDATE Stu\_Union SET Sno='S02' WHERE Sname='王兵';

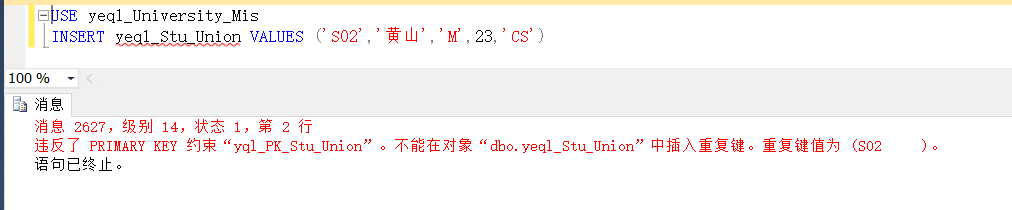
SELECT \* FROM Stu\_union;



（2）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

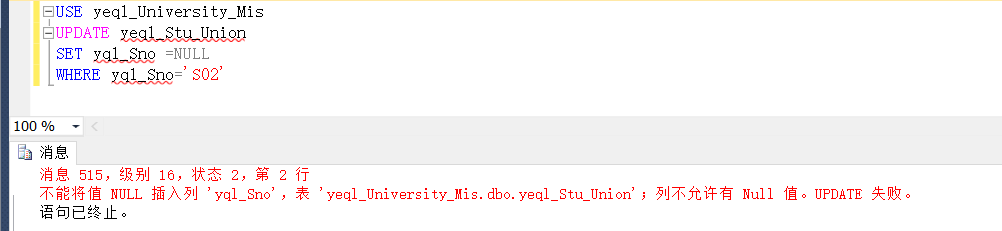
INSERT Stu\_Union VALUES (‘S02’,’黄山’,’M’,23,’CS’);



（3）   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

UPDATE Stu\_Union SET Sno =NULL WHERE Sno='S02';



（4）输入并执行如下SQL命令

①   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SET XACT\_ABORT ON

BEGIN TRANSACTION T1

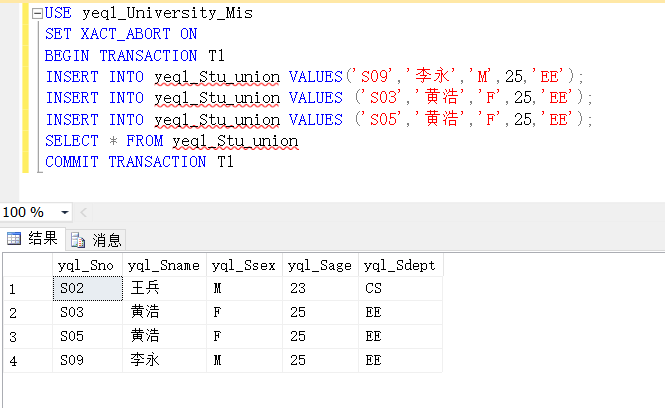
INSERT INTO Stu\_union VALUES(‘S09’,’李永’,’M’,25,’EE’);

INSERT INTO Stu\_union VALUES (‘S03’,’黄浩’,’F’,25,’EE’);

INSERT INTO Stu\_union VALUES (‘S05’,’黄浩’,’F’,25,’EE’);

SELECT \* FROM Stu\_union;

COMMIT TRANSACTION T1



②   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SET  XACT\_ABORT ON

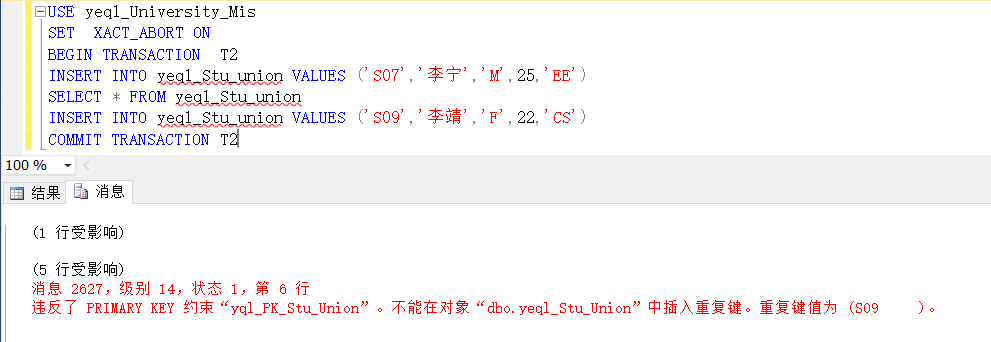
BEGIN TRANSACTION  T2

INSERT INTO Stu\_union VALUES (‘S07’,’李宁’,’M’,25,’EE’);

SELECT \* FROM Stu\_union;

INSERT INTO Stu\_union VALUES (‘S09’,’李靖’,’F’,22,’CS’);

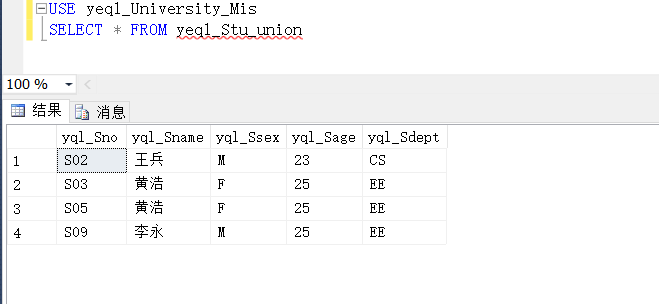
COMMIT TRANSACTION T2



③   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

SELECT \* FROM Stu\_union;



（5）输入并执行如下SQL命令

①   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

CREATE TABLE Scholarship

(

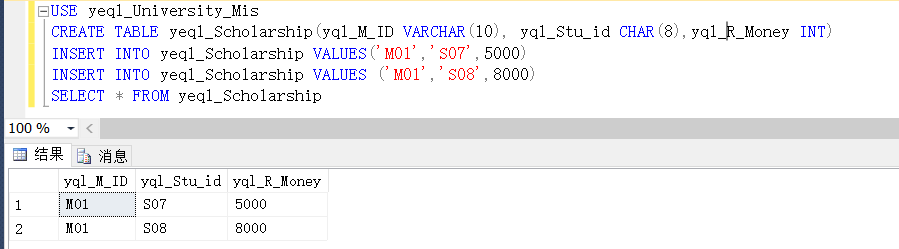
M\_ID VARCHAR(10), Stu\_id CHAR(8),R\_Money INT

)

INSERT INTO Scholarship VALUES(‘M01’,’ S07’,5000)

INSERT INTO Scholarship VALUES (‘M01’,’ S08’,8000)

SELECT \* FROM Scholarship

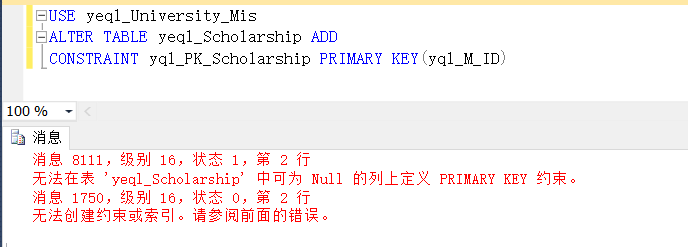


②   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

ALTER TABLE Scholarship ADD

CONSTRAINT PK\_Scholarship PRIMARY KEY(M\_ID)



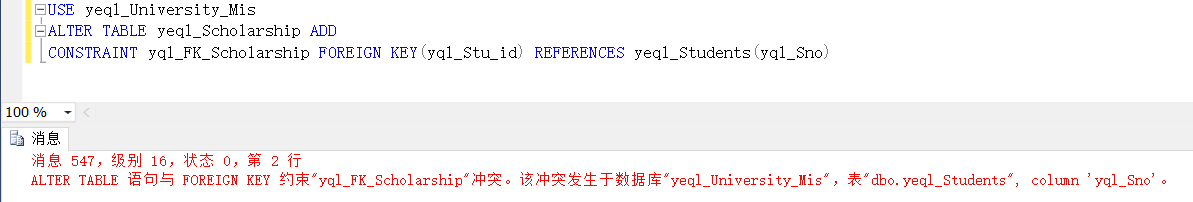
因为主键必须能唯一确定元组，但前面插入的元组中有两个有相同的yql\_M\_ID，说以yql\_M\_ID属性不能作为PRIMARY KEY。

③   在新建查询窗口中输入如下SQL语句：

USE University\_Mis

ALTER TABLE Scholarship ADD

CONSTRAINT FK\_Scholarship FOREIGN KEY(Stu\_id) REFERENCES Students(Sno)



NOTE：参与构造外键关系的列必须具有相同长度和小数位数，由于yeql\_Students.yql\_Sno的长度与外键yql\_FK\_Scholarship中的引用列yeql\_Scholarship.yql\_Stu\_id的长度不同，所以无法创建约束。

# 实验5、数据控制

****5.1 实验目的****

熟悉SQL的数据控制功能，能够使用SQL语句来向用户授予和收回权限。

****5.2 实验内容****

(1)  使用GRANT语句来对用户授权，对单个用户或多个用户授权，或使用保留字PUBLIC对所有用户授权。对不同的操作对象包括数据库、视图、基本表等进行不同权限的授权。

(2)  使用WITH GRANT OPTION字句授予用户传播该权限的权利。

(3)  当在授权时发生循环授权，考察DBS能否发现这个错误。如果不能，结合取消权限操作，查看DBS对循环授权的控制。

(4)  使用REVOKE子句收回授权，取消授权的级联反应。

****5.3 实验步骤****

用SSMS在数据库University\_Mis中建立三个用户USER1、USER2和USER3，他们在数据库中的角色是PUBLIC。请按以下要求，分别以管理员身份或这三个用户的身份登陆到数据库中，进行操作，并记录操作结果。

**(1)   授予所有用户对表Courses的查询权限。以USER1的身份登陆到**SSMS**，用SQL语言查询Courses和Students表，查询结果如何？**

(2)  授予用户USER1对表Students插入和更新的权限，但不授予删除权限，并且授予用户USER1传播这两个权限的权利。以USER1的身份登陆到SSMS，用SQL语言插入和更新Students表，结果如何？（注意更新操作的授权）

**(3)  允**许**用户USER2在表Reports中插入元组，更新Score列，可以查询除了Sno以外的所有列。以USER2的身份登陆到SSMS，用SQL语言插入更新并查询reports表，结果如何？**（注意更新操作的授权）

(4)  **用户USER1授予用户USER2对表Students插入和更新的权限，并且授予用户USER2传播插入和更新操作的权利。分别以USER1和user2的的身份登陆到SSMS，用SQL语言验证以上授权操作，结果如何？**

(5)  收回对用户USER1对表Courses查询权限的授权。分别以USER1和USER2的身份登陆到SSMS，用SQL语言查询Courses表，查询结果如何？

**(6)  由上面（2）和（4）的授权，再由用户USER2对用户USER3授予表Students插入和更新的权限，并且授予用户USER3传播插入和更新操作的权力。这时候，如果由USER3对USER1授予表Students的插入和更新权限是否能得到成功？如果能够成功，那么如果有用户USER2取消USER3的权限，对USER1会有什么影响？如果再由DBA取消USER1的权限，对USER2有什么影响？**

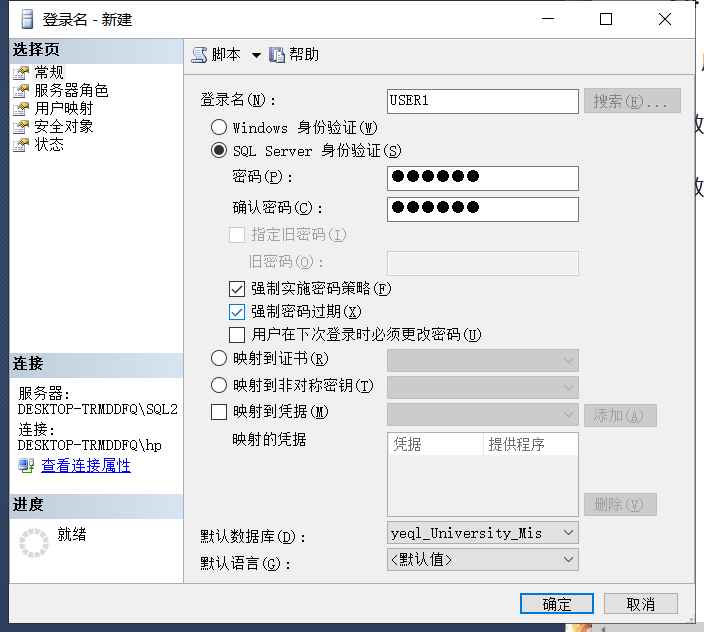
**（2）   实验要求**

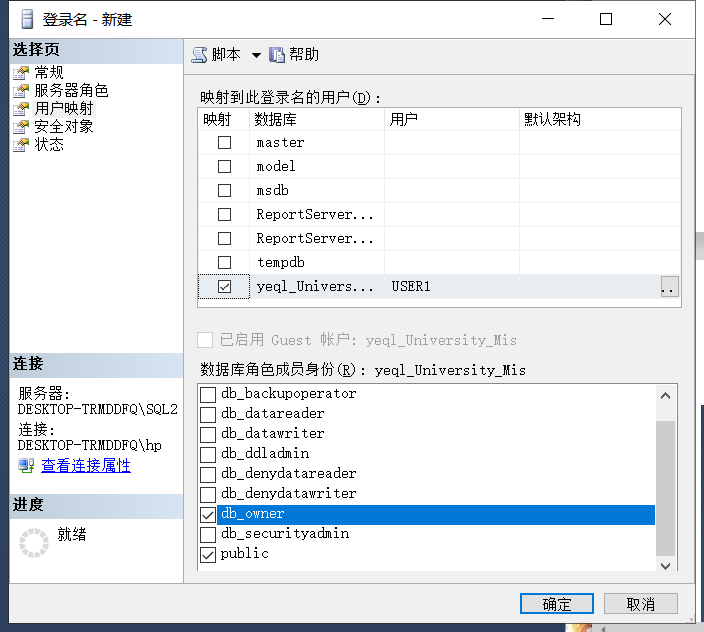
(1) 实验之前请细细阅读实验总体要求与说明指导书

(2 ) 在SQL Server 2005或2008的SSMS环境中，完成以上实验(1)-(6)步所有SQL数据定义操作，并将其中粗体步骤的操作窗口剪贴到实验报告中。

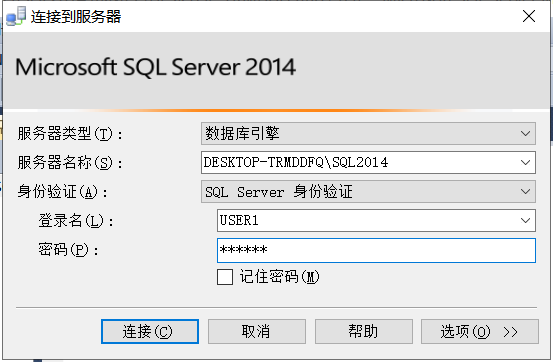
****5.4实验结果****

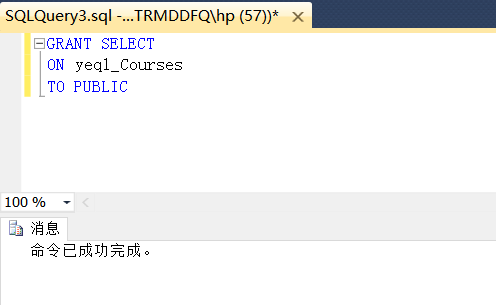
用SSMS在数据库University\_Mis中建立三个用户USER1、USER2和USER3，他们在数据库中的角色是PUBLIC。请按以下要求，分别以管理员身份或这三个用户的身份登陆到数据库中，进行操作，并记录操作结果。



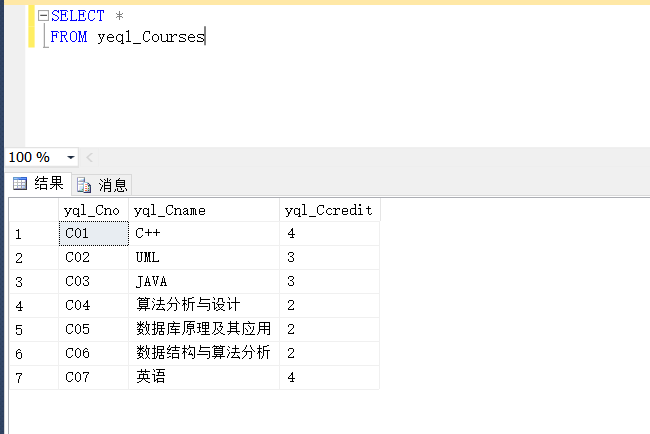


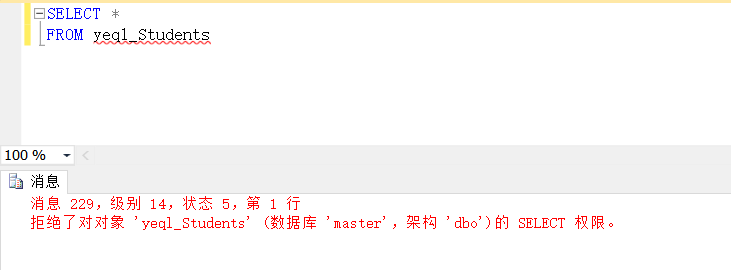
1. **授予所有用户对表Courses的查询权限。以USER1的身份登陆到**SSMS**，用SQL语言查询Courses和Students表，查询结果如何？**





USER1登录后，进行查询操作。

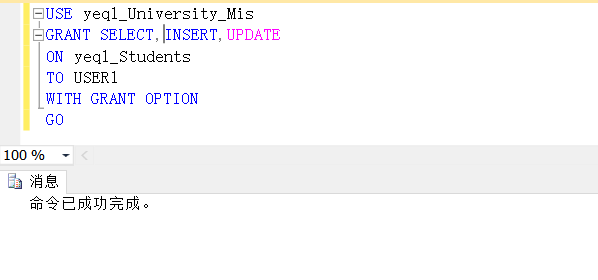




USER1访问yeql\_Students没有经过授权，所以无法访问。

1. 授予用户USER1对表Students插入和更新的权限，但不授予删除权限，并且授予用户USER1传播这两个权限的权利。以USER1的身份登陆到SSMS，用SQL语言插入和更新Students表，结果如何？（注意更新操作的授权）

NOTE:更新操作的授权必须先授权用户SELECT，否则用户登录时无法UPDATE。

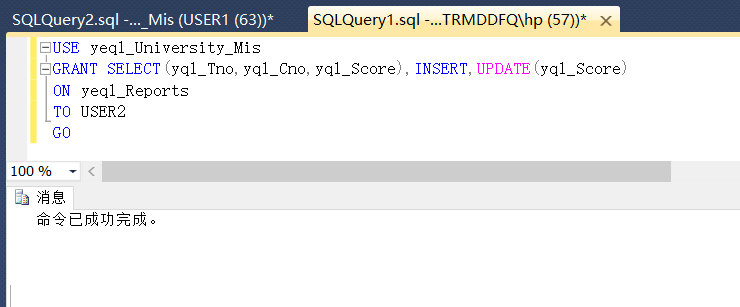


在USER1登录的情况下，进行SELSECT和UPDATE操作，成功。

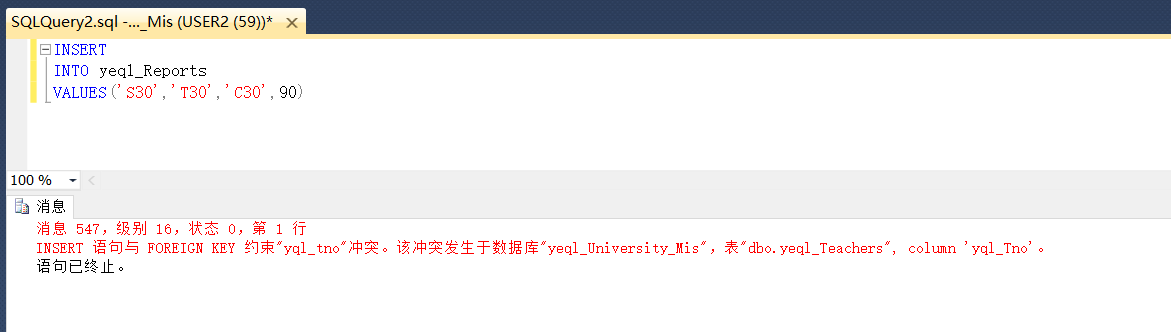


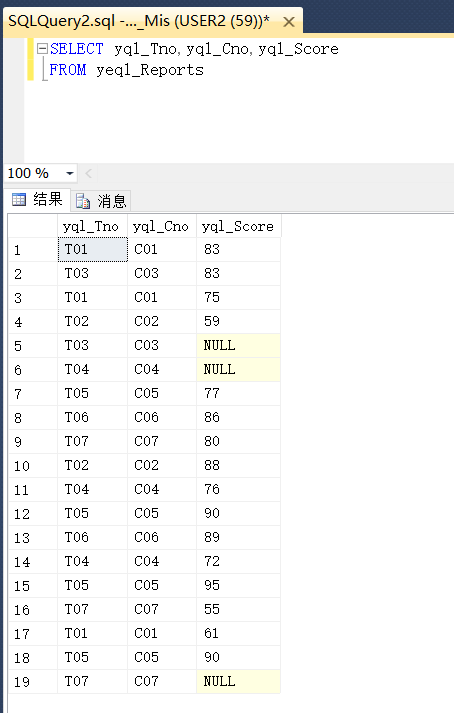
1. **允**许**用户USER2在表Reports中插入元组，更新Score列，可以查询除了Sno以外的所有列。以USER2的身份登陆到SSMS，用SQL语言插入更新并查询reports表，结果如何？**（注意更新操作的授权）

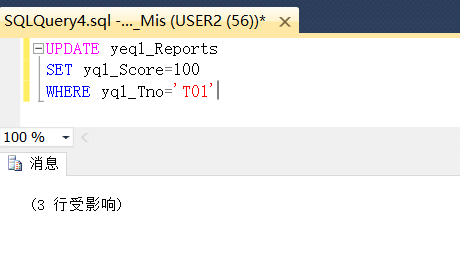
NOTE:更新的操作权限不仅要授权UPDEATE，还要授权SELECT

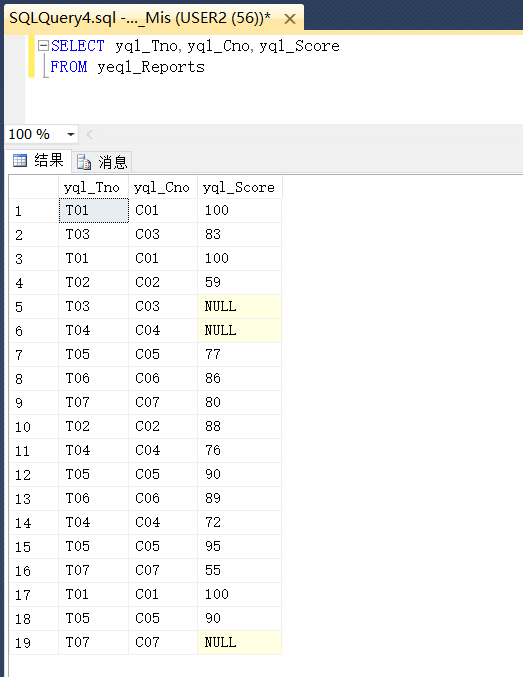


USER2登录后**插入更新并查询reports表**

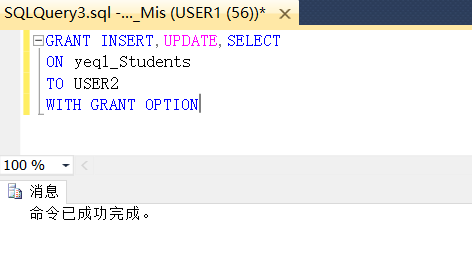






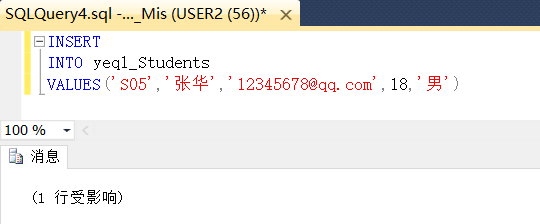


1. **用户USER1授予用户USER2对表Students插入和更新的权限，并且授予用户USER2传播插入和更新操作的权利。分别以USER1和user2的的身份登陆到SSMS，用SQL语言验证以上授权操作，结果如何？**

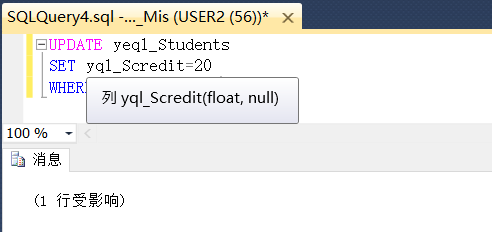


**user2的的身份登陆到SSMS，用SQL语言验证以上授权操作(对表Students插入和更新的权限，并且授予用户USER2传播插入和更新操作的权利)，得到结果：USER2能对表Students插入和更新，也能给其他用户授权对表Students的插入和更新操作，具体验证如下图所示。**

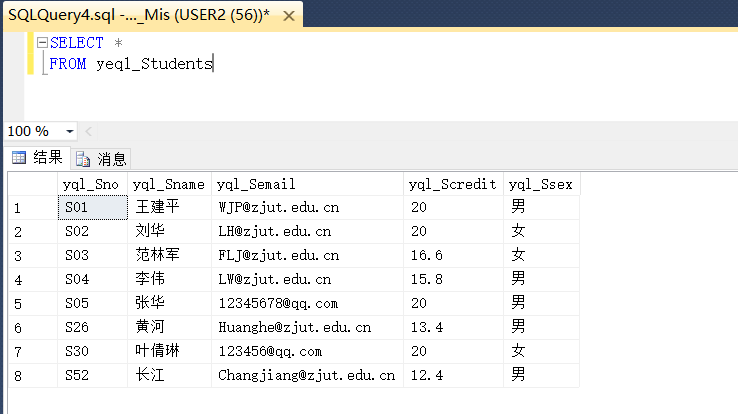
**USER2能对表Students插入**



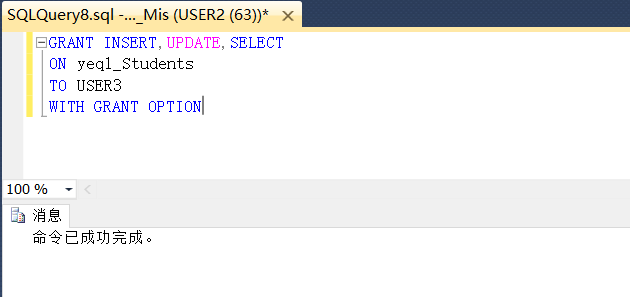
**USER2能对表Students更新**

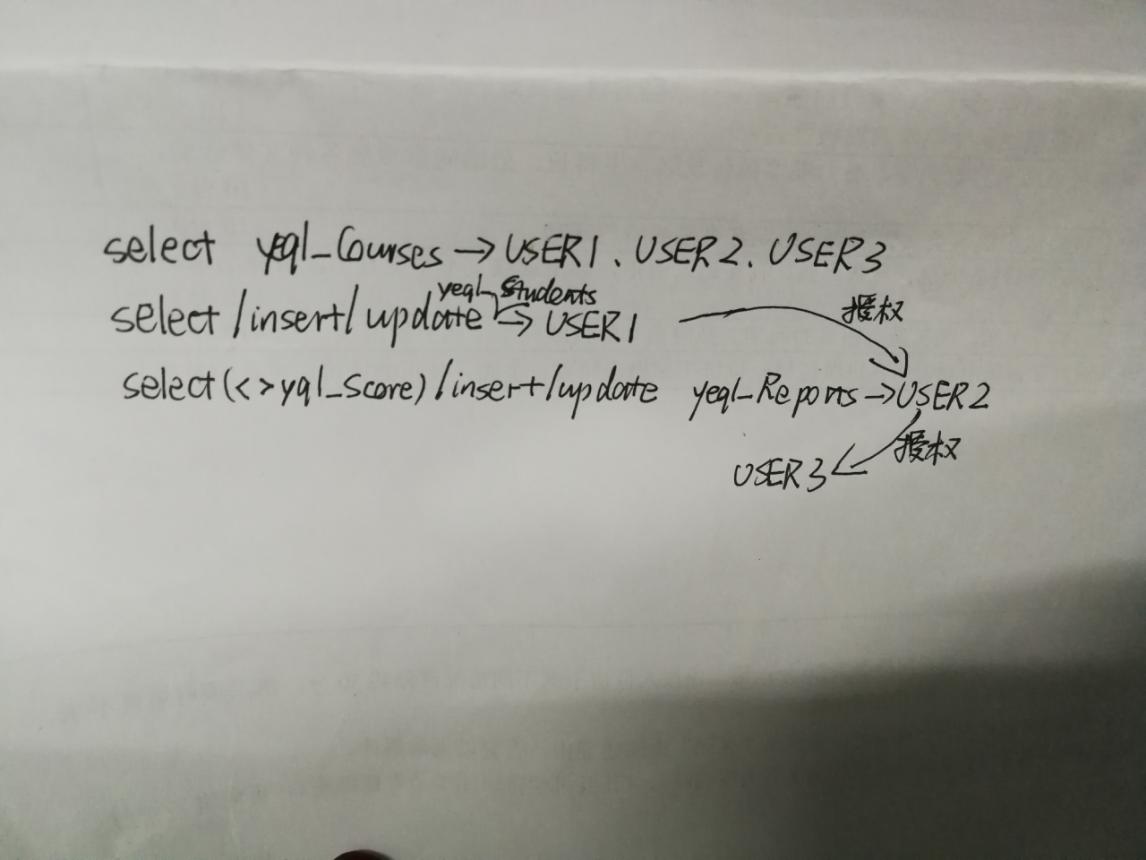


**USER2能对表Students查询**

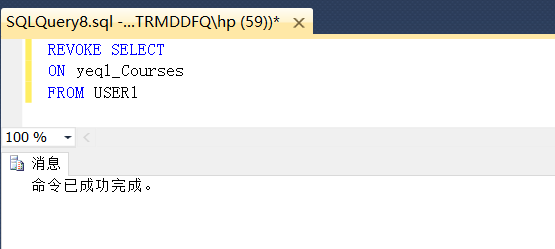


**USER2能给其他用户授权对表Students的插入和更新操作**





(5)  收回对用户USER1对表Courses查询权限的授权。分别以USER1和USER2的身份登陆到SSMS，用SQL语言查询Courses表，查询结果如何？



**结果：USER1拒绝了**对表Courses的查询；USER2可以对表Courses进行查询，因为USER2对表Courses的查询是管理员授权的。

USER2:

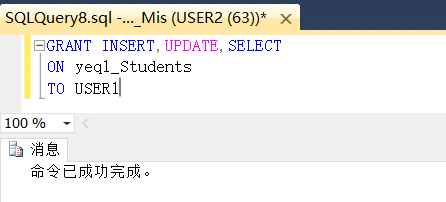


**（6）  由上面（2）和（4）的授权，再由用户USER2对用户USER3授予表Students插入和更新的权限，并且授予用户USER3传播插入和更新操作的权力。这时候，如果由USER3对USER1授予表Students的插入和更新权限是否能得到成功？如果能够成功，那么如果有用户USER2取消USER3的权限，对USER1会有什么影响？如果再由DBA取消USER1的权限，对USER2有什么影响？**

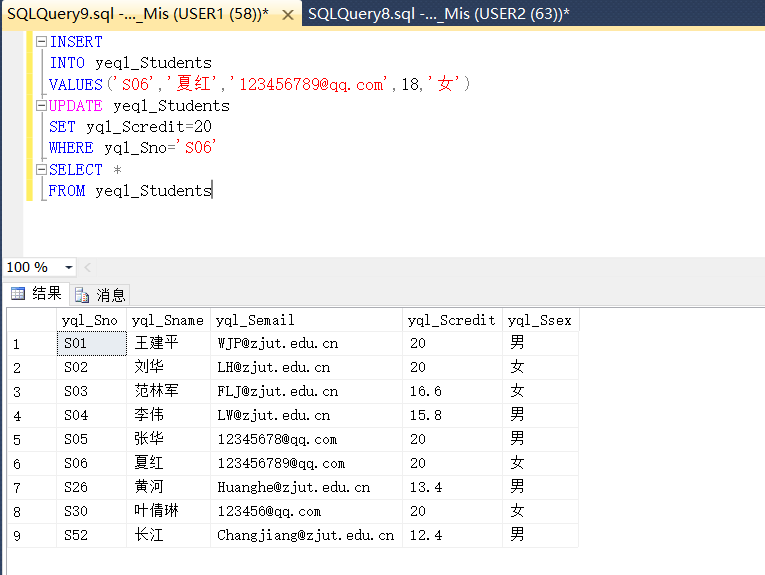
**由用户USER2对用户USER3授予表Students插入和更新的权限，并且授予用户USER3传播插入和更新操作的权力：**

# 

**再由USER3对USER1授予表Students的插入和更新权限能得到成功：**



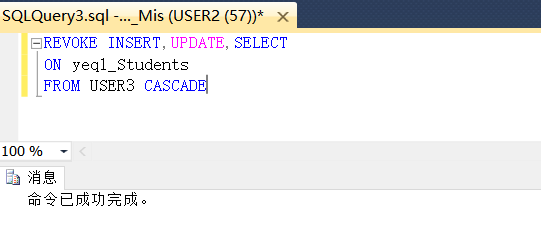
在USER1中进行验证：

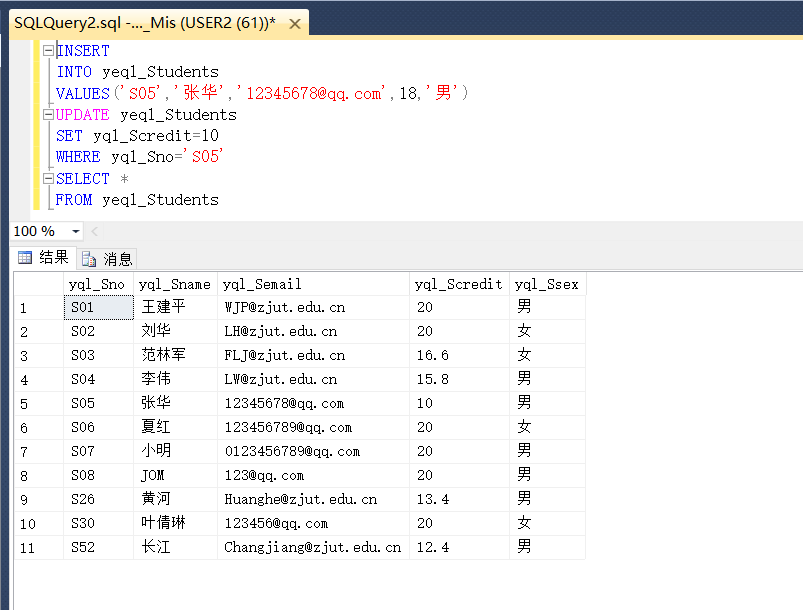


**问：如果有用户USER2取消USER3的权限，对USER1会有什么影响？**

**答：USER3赋予USER1的相应权限也被收回，但最初管理员赋予USER1的权限还在。所以USER2撤回权限后，在USER1中依然可以对表Students的插入和更新。**

**USER2撤回对USER3的权限：**

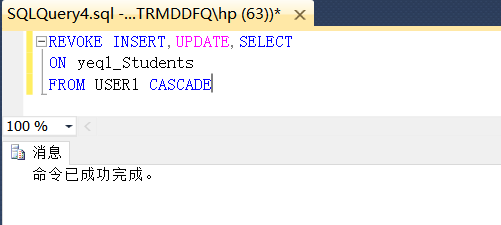




**问：如果再由DBA取消USER1的权限，对USER2有什么影响？**

**答：DBA取消USER1插入、更新yeql\_Students表的权限，USER2就不能对yeql\_Students表进行插入、更新，因为这个操作是USER1授权的。但USER2还能对yeql\_Reports表进行插入、更新，因为这个操作是DBA授权的。具体验证如下。**

**由DBA取消USER1的权限：**



**s**

**在USER2中不可以对表Students的插入和更新。**

**在USER2中依然可以对表yeql\_Reports进行插入和更新：**

