

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 语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 语句收回授权，取消授权的级联反应。

5.3 实验步骤

用 SSMS 在数据库 chenht_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 chenht_Courses 的查询权限。以 USER1 的身份登陆到 SSMS，用 SQL 语言查询 chenht_Courses 和 chenht_Students 表，查询结果如何？
- (2) 授予用户 USER1 对表 chenht_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 chenht_Students 表，结果如何？（注意更新操作的授权）
- (3) 允许用户 USER2 在表 chenht_Reports 中插入元组，更新 chenht_Score 列，可以查询除了 chenht_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 chenht_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 chenht_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 **chenht_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 **chenht_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 **chenht_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 **chenht_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？

5.4 实验结果

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？

SQL 语句

```
/* user = [admin], database = chenht_University_Mis */
/* 用户授权 */
grant select on chenht_Courses to public

/* user = USER1, database = chenht_University_Mis */
/* 查询 chenht_Courses 表 */
select * from chenht_Courses
/* 查询 chenht_Students 表 */
select * from chenht_Students
```

实验结果

能够查询 **Course** 表，不能够查询 **Students** 表。

a 授权语句

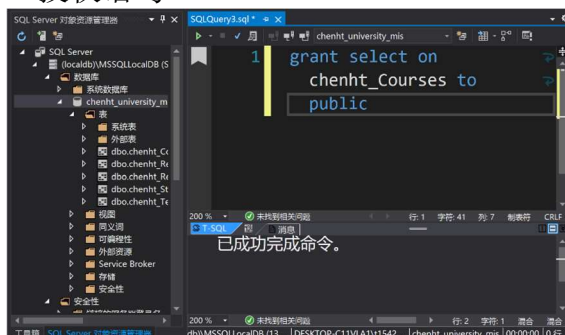


图 5-1 实验结果截图(1)

b 用 **USER1** 查询 **chenht_Courses**

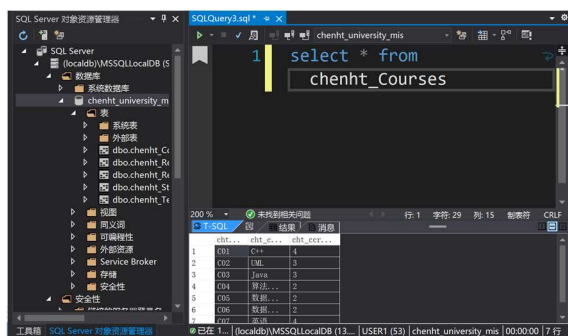


图 5-2 实验结果截图(2)

c 用 USER1 查询 chenht_Students

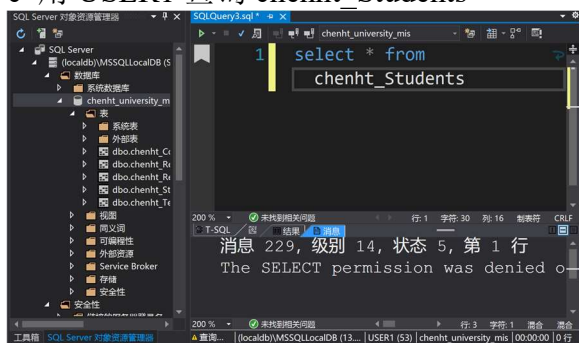


图 5-3 实验结果截图(3)

3) 允许用户 **USER2** 在表 **chenht_Reports** 中插入元组, 更新 **chenht_Score** 列, 可以查询除了 **chenht_Sno** 以外的所有列。以 **USER2** 的身份登录到 SSMS, 用 SQL 语言插入更新并查询 **chenht_Reports** 表, 结果如何? (注意更新操作的授权)

SQL 语句

```
/* user = [admin], database = chenht_University_Mis */
/* 授权 USER2 */
grant insert, update(cht_score), select(cht_tno, cht_cno,
cht_score)
on chenht_Reports to USER2

/* user = USER2, database = chenht_University_Mis */
/* insert :> chenht_Reports with USER2 */
insert into chenht_Reports
values ('S09', 'T01', 'C01', 93)

/* user = USER2, database = chenht_University_Mis */
/* update :> with USER2 */
update chenht_Reports set cht_score = 93
where cht_sno = 'S09' and cht_cno = 'C01'

/* user = USER2, database = chenht_University_Mis */
/* update :> chenht_Reports with USER2 */
```

```

update chenht_Reports set cht_score = cht_score + 1 where
cht_score < 100 and cht_cno = 'C01'

/* user = USER2, database = chenht_University_Mis */
/* select :> chenht_Reports with USER2 */
select cht_cno, count(*)re_count, avg(cht_score)avg_score
from chenht_Reports
group by cht_cno

```

实验结果

可以更新和查询部分列

a 授予 USER2 权限

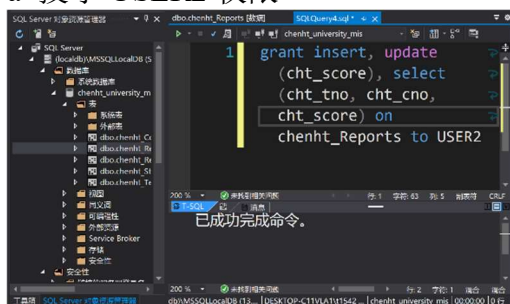


图 5-4 实验结果截图(4)

b 向 chenht_Reports 中插入数据

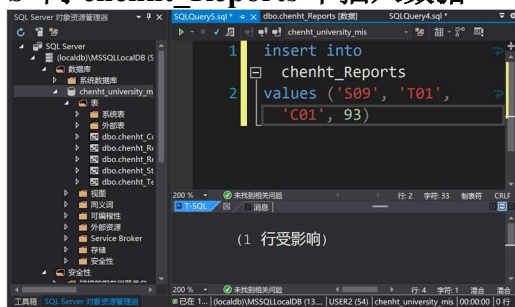


图 5-5 实验结果截图(5)

c 更新 cht_score (case1)

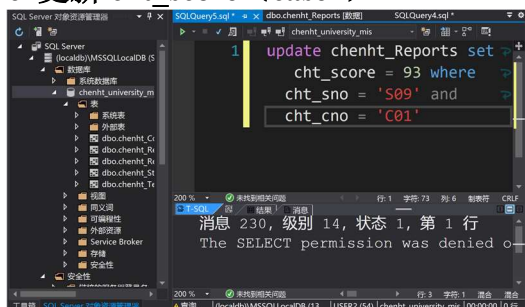


图 5-6 实验结果截图(6)

d 更新 cht_score (case2)



图 5-7 实验结果截图(7)

e 查询得到每个课程的平均成绩

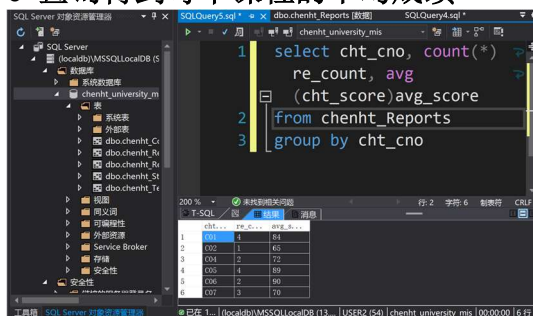


图 5-8 实验结果截图(8)

4) 用户 USER1 授予用户 USER2 对表 chenht_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 的身份登录到 SSMS，用 SQL 语言验证以上授权操作，结果如何？

SQL 语句

```
/* user = USER1, database = chenht_University_Mis*/
/* USER2 插入数据 */
insert into chenht_Students values ('S10', '王中波', 'WZB@zjut.edu.cn', 20.2, '男')

/* user = USER1 , database = chenht_University_Mis */
/* 授权给 USER2 */
grant insert,update on chenht_Students to USER2 with grant option

/* user = USER1, database = chenht_University_Mis*/
/* USER2 插入数据 */
insert into chenht_Students values ('S10', '王中波', 'WZB@zjut.edu.cn', 20.2, '男')
```

实验结果

更新授权符合预期的效果

a. 授权前使用 USER2 进行插入数据操作

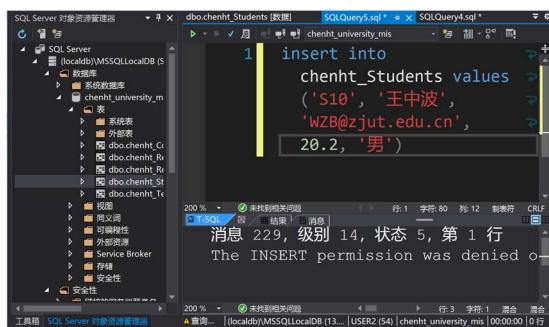


图 5-9 实验结果截图(9)

b. 授权

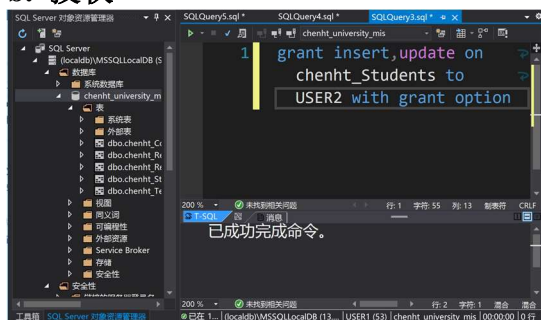


图 5-10 实验结果截图(10)

c. 授权后使用 USER2 进行插入数据操作

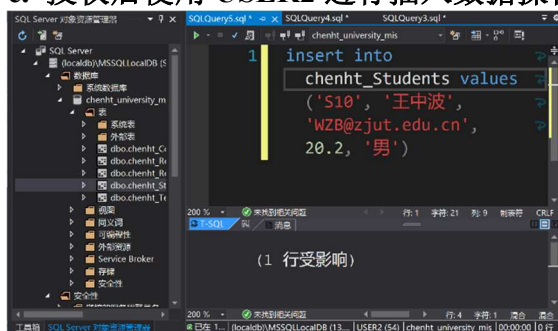


图 5-11 实验结果截图(11)

6) 由上面 (2) 和 (4) 的授权, 再由用户 USER2 对用户 USER3 授予表 chenht_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 chenht_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?

SQL 语句

实验结果

允许循环授权, USER3 授权 USER1 后, USER1 能够插入, USER2 收回 USER3 级联授权后, USER1 能够插入, DBA 收回 USER1 的级联授权后, USER2 不能够插入。

a. 授权(USER2->USER3)

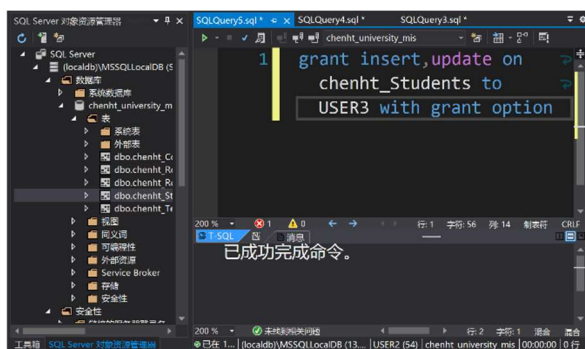


图 5-12 实验结果截图(12)

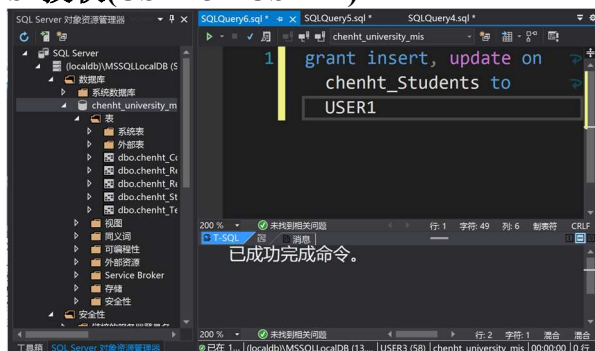
b 授权(USER3->USER1)

图 5-13 实验结果截图(13)

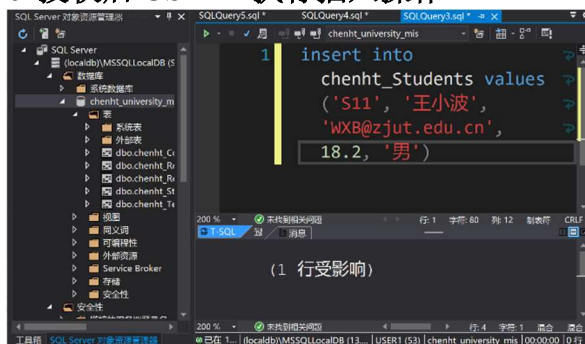
c 授权后 USER1 执行插入操作

图 5-14 实验结果截图(14)

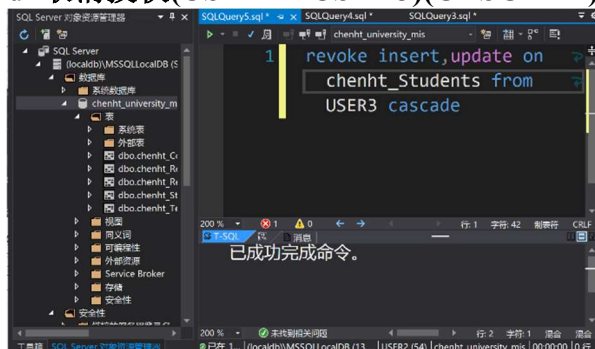
d 取消授权(USER2<-USER3)(CASCADE)

图 5-15 实验结果截图(15)

e 取消授权后 USER1 执行插入操作

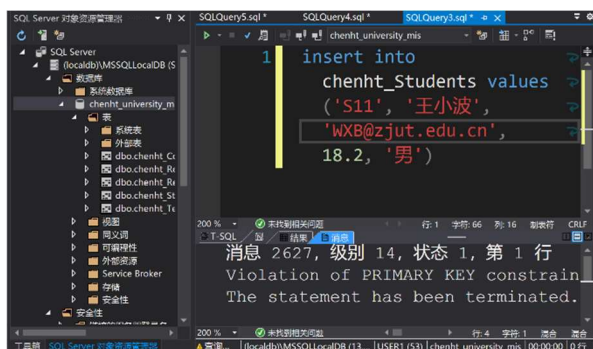


图 5-16 实验结果截图(16)

f 取消授权([DBA]<-USER1)

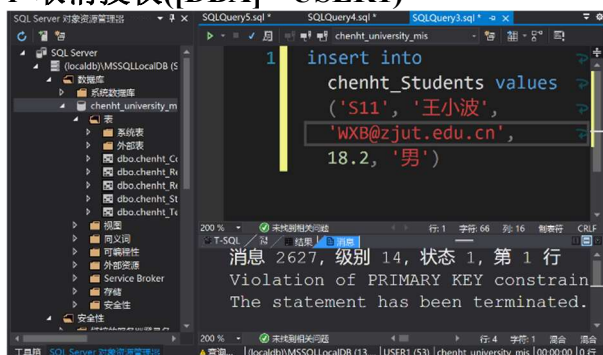


图 5-17 实验结果截图(17)

g 取消授权后 USER1 执行插入操作

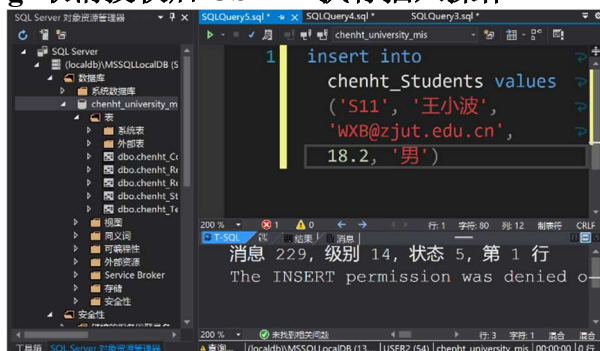


图 5-18 实验结果截图(18)

5.5 实验体会

1) 实验反思

- (关于 SELECT 权限) SELECT 权限不仅仅受限 SELECT 查询语句。只要用到读操作的场合，通通需要检查 SELECT 权限。例如 WHERE 语句的条件表达式。
- (GRANT/CASCADE) 传播权限授予后，被授予的角色可以把相应的权限分发给其他用户，由于传播权限不存在死循环情况。在 SQLServer，允许传播授权。从 2 个及以上覆盖的权限中删除至 1 个并不会影响相应的权限。有 GRANT 的权限授予必须加上 CASCADE 收回权限，此时收回权限后，被授予及被授予授予的用户均会被取回权限。
- (权限关系) INSERT 和 UPDATE 权限均与 SELECT 权限有关。如果 SELECT 权限的约束越高，则在执行 INSERT 和 UPDATE 操作时，就会越受限。

2) 实验收获

通过本次实验，体会了创建登录名、创建用户映射、授予和收回权限等各种

SQL 语法和概念。特别是对 GRANT 和 CASCADE 关键字有了更加深刻的理解。能够在多用户的环境下切换，并设计一个合理的多用户数据库。