实验5、用户权限控制

5.1 实验目的

熟悉 SQL 的用户权限控制功能，能够使用 SQL 语句实现用户、角色的创建、修改与删除，用户权限的授予和回收，安全策略设置等。

5.2 实验内容

以下 XXX 表示自己名字汉语拼音首字母缩写。

(1) 为方便管理，采用系统管理员用户 XXXuser 登录数据库，完成初始的权限分配工作，然后用相应用户名登录数据库以验证权限分配的正确性。

(2) 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 ALL 、 PUBLIC 对所有用户授权。对不同的操作对象包括模式、数据库、视图、基本表等进行不同权限的授权。

(3) 使用 WITH GRANT OPTION 和 WITH ADMIN OPTION 子句授予用户传播该权限的权利。

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

(5) 使用 REVOKE 子句收回授权，取消授权的级联反应。

5.3 实验步骤

为简化数据库 XXXDB 的管理，老师的信息和学生的信息由校级教务教学部门来设置，课程信息和选课信息由院级教务教学部门来设置，教师可以设置学生成绩，学生查看课程成绩。以系统管理员 XXXuser 的身份登陆数据库 XXXDB ，分别建立如下用户和角色：用户包括校级教务处教学主管 JWC ，院级教学秘书 JxmsA 、 JxmsB ，教师 TeachA 、 TeachB 和学生 StuA 、 StuB 。角色包括 JwcRole 、 JxmsRole 和 TeacherRole 、 StudentRole 。验证权限分配之前，请备份好数据库，针对不同用户所具有的权限，分别以系统管理员 XXXuser 身份或以上用户身份登陆到数据库 XXXDB 中（用 gsql 客户端工具登陆更方便切换用户，具体命令可参考《0-2 zjut openGauss 数据库客户端工具》），设计相应的 SQL 语句，进行操作加以验证，并记录操作结果。

(1) 创建用户

/* 注意: CREATE USER 语句不是标准 SQL ，不同的 RDBMS 不同。

本实验用的是 openGauss 创建用户语句,同时查阅资料如何将系统权限授权给角色或用户，参考 GRANT 语法解释。

通过 CREATE USER 创建的用户，默认具有 LOGIN 权限；

通过 CREATE USER 创建用户的同时系统会在执行该命令的数据库中，为该用户创建一个同名的 SCHEMA ；其他数据库中，则不自动创建同名的 SCHEMA ；用户可使用 CREATE SCHEMA 命令，分别在其他数据库中，为该用户创建同名 SCHEMA 。

系统管理员在普通用户同名 schema 下创建的对象，所有者为 schema 的同名用户（非系统管理员）。*/

①创建校院两级教务教学主管用户，要求具有创建用户或角色的权利。

```
/* CREATE USER "JWC"加上双引号系统区分大小写，无引号则系统全部转换为小写*/
CREATE USER JWC WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE USER JxmsA WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE USER JxmsB WITH CREATEROLE PASSWORD 'Bigdata@123';
```

列出所有用户名称信息：

```
\du
\du+
```

```
postgres=# \du
                                   List of roles
 Role name |                            Attributes                             | Member of
-----------+-------------------------------------------------------------------+-----------
 ahwei     | Sysadmin                                                          | {}
 omm       | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}

postgres=# \du+
                                        List of roles
 Role name |                            Attributes                             | Member of | Description
-----------+-------------------------------------------------------------------+-----------+-------------
 ahwei     | Sysadmin                                                          | {}        |
 omm       | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}        |
```

```
postgres=# CREATE USER JWC WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# CREATE USER JxmsA WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# CREATE USER JxmsB WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name |                            Attributes                             | Member of
-----------+-------------------------------------------------------------------+-----------
 ahwei     | Sysadmin                                                          | {}
 jwc       | Create role                                                       | {}
 jxmsa     | Create role                                                       | {}
 jxmsb     | Create role                                                       | {}
 omm       | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}

postgres=# \du+
                                        List of roles
 Role name |                            Attributes                             | Member of | Description
-----------+-------------------------------------------------------------------+-----------+-------------
 ahwei     | Sysadmin                                                          | {}        |
 jwc       | Create role                                                       | {}        |
 jxmsa     | Create role                                                       | {}        |
 jxmsb     | Create role                                                       | {}        |
 omm       | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}        |
```

②创建教师和学生用户。

```
CREATE USER TeachA IDENTIFIED BY 'Bigdata@123';
CREATE USER TeachB IDENTIFIED BY 'Bigdata@123';
CREATE USER StuA IDENTIFIED BY 'Bigdata@123';
CREATE USER StuB IDENTIFIED BY 'Bigdata@123';
```

```
postgres=# CREATE USER TeachA  IDENTIFIED BY  'Bigdata@123';
CREATE ROLE
postgres=# CREATE USER TeachB  IDENTIFIED BY  'Bigdata@123';
CREATE ROLE
postgres=# CREATE USER StuA  IDENTIFIED BY  'Bigdata@123';
CREATE ROLE
postgres=# CREATE USER StuB  IDENTIFIED BY  'Bigdata@123';
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name |                            Attributes                             | Member c
-----------+-------------------------------------------------------------------+---------
 ahwei     | Sysadmin                                                          | {}
 jwc       | Create role                                                       | {}
 jxmsa     | Create role                                                       | {}
 jxmsb     | Create role                                                       | {}
 omm       | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}
 stua      |                                                                   | {}
 stub      |                                                                   | {}
 teacha    |                                                                   | {}
 teachb    |                                                                   | {}
```

(2) 创建角色并分配权限

①分别创建校院两级教务教学管理角色`JwcRole`和`JxmsRole`、教师角色`TeacherRole`和学生角色`StudentRole`。

```
postgres=# CREATE ROLE JwcRole WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# CREATE ROLE JxmsRole WITH CREATEROLE PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# CREATE ROLE TeacherRole WITH  PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# CREATE ROLE StudentRole WITH  PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# \du
                                   List of roles
  Role name  |                            Attributes                             | Member of
-------------+-------------------------------------------------------------------+-----------
 ahwei       | Sysadmin                                                          | {}
 jwc         | Create role                                                       | {}
 jwcrole     | Create role, Cannot login                                         | {}
 jxmsa       | Create role                                                       | {}
 jxmsb       | Create role                                                       | {}
 jxmsrole    | Create role, Cannot login                                         | {}
 omm         | Sysadmin, Create role, Create DB, Replication, Administer audit, UseFT | {}
 stua        |                                                                   | {}
 stub        |                                                                   | {}
 studentrole | Cannot login                                                      | {}
 teacha      |                                                                   | {}
 teachb      |                                                                   | {}
 teacherrole | Cannot login                                                      | {}
```

②把数据库XXXDB的模式XXXschema的使用权限授予所有角色与部分用户。

```
GRANT USAGE ON SCHEMA ahweischema to JwcRole WITH GRANT OPTION;
GRANT USAGE ON SCHEMA ahweischema to JxmsRole WITH GRANT OPTION;
GRANT USAGE ON SCHEMA ahweischema to TeacherRole WITH GRANT OPTION;
GRANT USAGE ON SCHEMA ahweischema to StudentRole WITH GRANT OPTION;
GRANT USAGE ON SCHEMA ahweischema to JWC,JxmsA,JxmsB, TeachA, TeachB,StuA,StuB WITH
GRANT OPTION;
```

```
ahweidb=# GRANT USAGE ON SCHEMA ahweischema to JxmsRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT USAGE ON SCHEMA ahweischema to TeacherRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT USAGE ON SCHEMA ahweischema to StudentRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT USAGE ON SCHEMA ahweischema to JWC,JxmsA,JxmsB, TeachA, TeachB,StuA,StuB WITH GRANT OPTION;
GRANT
ahweidb=# GRANT USAGE ON SCHEMA ahweischema to JwcRole WITH GRANT OPTION;
GRANT
ahweidb=#
```

③把 Teachers、Students、Courses 和 STC 的相应权限分别授权给不同的角色

```
GRANT ALL ON TABLE Teachers,Students to JwcRole WITH GRANT OPTION;
GRANT ALL ON TABLE Courses,STC to JxmsRole WITH GRANT OPTION;
GRANT SELECT,UPDATE ON TABLE STC to TeacherRole WITH GRANT OPTION;
GRANT SELECT ON TABLE STC to StudentRole WITH GRANT OPTION;
```

```
ahweidb=# GRANT ALL ON TABLE Teachers,Students to JwcRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT ALL ON TABLE Courses,STC to JxmsRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT ALL ON TABLE Courses,STC to JxmsRole WITH GRANT OPTION;
GRANT
ahweidb=# GRANT SELECT ON TABLE STC to StudentRole WITH GRANT OPTION;
GRANT
ahweidb=#
```

(3) 为用户分配角色及权限

①为用户按角色分配并授予权限

```
GRANT JwcRole TO JWC WITH ADMIN OPTION;
GRANT JxmsRole TO JxmsA WITH ADMIN OPTION;
GRANT TeacherRole TO TeachA  ADMIN OPTION;
GRANT StudentRole TO StuA WITH ADMIN OPTION;
```

```
ahweidb=# GRANT JwcRole TO JWC WITH ADMIN OPTION;
GRANT ROLE
ahweidb=# GRANT JxmsRole TO JxmsA WITH ADMIN OPTION;
GRANT ROLE
ahweidb=# GRANT TeacherRole TO TeachA WITH ADMIN OPTION;
GRANT ROLE
ahweidb=# GRANT StudentRole TO StuA WITH ADMIN OPTION;
GRANT ROLE
```

②按用户单独赋予权限

```
GRANT SELECT ON TABLE Teachers,Students, Courses,STC to JxmsB,TeachB, StuB WITH
GRANT OPTION;
```

```
ahweidb=# GRANT SELECT ON TABLE Teachers,Students, Courses,STC to JxmsB,TeachB, StuB WITH GRANT OPTION;
GRANT
ahweidb=#
```

(4)分别以JWC、JxmsA、TeachA和StuA的身份登陆数据库XXXDB，验证按角色授予的权限，用SQL语言查询XXXschema.Teachers、XXXschema.Students、XXXschema.Courses和XXXschema.STC表，查询结果如何？并分析原因。

例子：`gsql -d ylhdb -p 26000 -r -U jwc  -W Bigdata@123`

其中`-U`是区分大小写的（命令行上的参数是区分大小写的）。

- `JWC`

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U jwc  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from ahweischema.Teachers;
ERROR:  permission denied for relation teachers
ahweidb=> select * from ahweischema.Students;;
ERROR:  permission denied for relation students
ahweidb=> select * from ahweischema.Courses;
ERROR:  permission denied for relation courses
ahweidb=> select * from ahweischema.STC;
ERROR:  permission denied for relation stc
```

- `JxmsA`

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U jxmsa  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from ahweischema.Teachers;
ERROR:  permission denied for relation teachers
ahweidb=> select * from ahweischema.Students;
ERROR:  permission denied for relation students
ahweidb=> select * from ahweischema.Courses;
ERROR:  permission denied for relation courses
ahweidb=> select * from ahweischema.STC;
ERROR:  permission denied for relation stc
```

- `TeachA`

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U teacha  -W Bigdata@1234
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from ahweischema.Teachers;
ERROR:  permission denied for relation teachers
ahweidb=> select * from ahweischema.Students;
ERROR:  permission denied for relation students
ahweidb=> select * from ahweischema.Courses;
ERROR:  permission denied for relation courses
ahweidb=> select * from ahweischema.STC;
ERROR:  permission denied for relation stc
```

- `StuA`

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U stua  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from ahweischema.Teachers;
ERROR:  permission denied for relation teachers
ahweidb=> select * from ahweischema.Students;
ERROR:  permission denied for relation students
ahweidb=> select * from ahweischema.Courses;
ERROR:  permission denied for relation courses
ahweidb=> select * from ahweischema.STC;
ERROR:  permission denied for relation stc
```

> 四个用户都无法查询表，因为没有赋予他们查询权限

```
set search_path=ahweischema,jwc,public;
```

```
ahweidb=> set search_path=ahweischema,jwc,public;
SET
ahweidb=>
```

(5) 用系统管理员`XXXuser`授予用户`JxmsB`对表`XXXschema.Students`插入和更新的权限，但不授予删除权限，并且授予用户`JxmsB`传播这两个权限的权利。以`JxmsB`的身份登陆，用`SQL`语言插入和更新`Students`表，结果如何？（注意更新操作的授权）

```
GRANT insert,update ON TABLE ahweischema.Students to JxmsB WITH GRANT OPTION;
```

```
ahweidb=# GRANT insert,update ON TABLE ahweischema.Students to JxmsB WITH GRANT OPTION;
GRANT
ahweidb=#
```

```
GRANT Insert ON TABLE Teachers,Students, Courses,STC to JxmsB,TeachB, StuB WITH
GRANT OPTION;
```

```
ahweidb=# GRANT Insert ON TABLE Teachers,Students, Courses,STC to JxmsB,TeachB, StuB WITH GRANT OPTION;
GRANT
```

```
set search_path=ylhschema,jwc,public;
```

```
ahweidb=# set search_path=ahweischema,jwc,public;
SET
ahweidb=#
```

JxmsB身份登录：

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U jxmsb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

SQL语言插入：

```
ahweidb=> insert into students (sno, sname, semail, scredit, ssex) values ('S666', '阿伟', '520Wjw@zjut.edu.cn', 100.0, '男');
INSERT 0 1
ahweidb=> select * from students;
 sno  | sname  |         semail         | scredit | ssex
------+--------+-------------------------+---------+------
 S01  | 王建平 | WJP@zjut.edu.cn         |    23.1 | 男
 S02  | 刘华   | LH@zjut.edu.cn          |    24.6 | 女
 S03  | 范林军 | FLJ@zjut.edu.cn         |    16.6 | 女
 S04  | 李伟   | LW@zjut.edu.cn          |    15.8 | 男
 S26  | 黄河   | HUanghe@zjut.edu.cn     |    13.4 | 男
 S52  | 长江   | Changjiang@zjut.edu.cn  |    12.4 | 男
 S100 | 刘华   |                         |         | 男
 S666 | 阿伟   | 520Wjw@zjut.edu.cn      |   100.0 | 男
(8 rows)
```

SQL语言更新：

```
ahweidb=> update students set scredit = 100;
UPDATE 8
ahweidb=> select * from students;
 sno  | sname  |         semail          | scredit | ssex
------+--------+-------------------------+---------+------
 S01  | 王建平 | WJP@zjut.edu.cn         |   100.0 | 男
 S02  | 刘华   | LH@zjut.edu.cn          |   100.0 | 女
 S03  | 范林军 | FLJ@zjut.edu.cn         |   100.0 | 女
 S04  | 李伟   | LW@zjut.edu.cn          |   100.0 | 男
 S26  | 黄河   | HUanghe@zjut.edu.cn     |   100.0 | 男
 S52  | 长江   | Changjiang@zjut.edu.cn  |   100.0 | 男
 S100 | 刘华   |                         |   100.0 | 男
 S666 | 阿伟   | 520Wjw@zjut.edu.cn      |   100.0 | 男
(8 rows)
```

> 结果：可以查询或更新`Student`表

(6) 用系统管理员`XXXuser`授予允许用户`TeachB`在表`STC`中插入元组，更新`Score`列，可以查询除了`Sno`以外的所有列。以`TeachB的身份登陆，用SQL语言插入更新并查询STC表，结果如何？（注意更新操作的授权）操作如下SQL语句，结果如何？为什么?

```
SELECT * FROM XXXschema.STC;
```

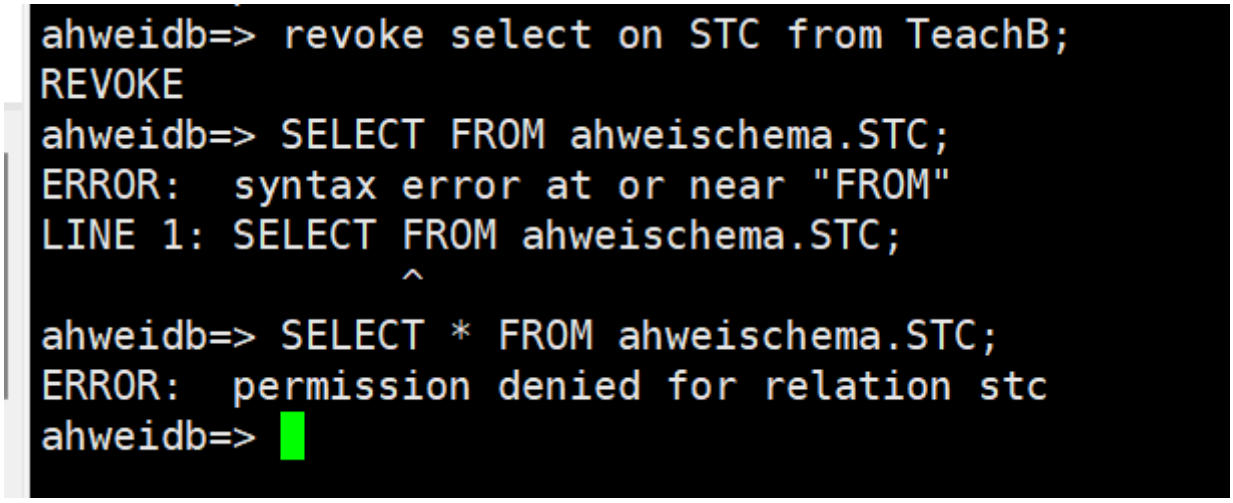
```
ahweidb=# select * from ahweischema.stc;
 sno | tno | cno | score
-----+-----+-----+-------
 S01 | T01 | C01 |  83.0
 S01 | T03 | C03 |  85.0
 S02 | T01 | C01 |  75.0
 S02 | T02 | C02 |  45.0
 S02 | T03 | C03 |
 S02 | T04 | C04 |
 S02 | T05 | C05 |  70.0
 S02 | T04 | C06 |  83.0
 S02 | T05 | C07 |  90.0
 S02 | T01 | C08 |  83.0
 S02 | T02 | C09 |  77.0
 S02 | T07 | C10 |  83.0
 S02 | T06 | C11 |  88.0
 S03 | T01 | C08 |  63.0
 S03 | T02 | C02 |  93.0
 S03 | T01 | C01 |  78.0
 S04 | T06 | C06 |  89.0
 S04 | T05 | C05 |  93.0
 S26 | T07 | C10 |  45.0
 S26 | T04 | C04 |  86.0
 S52 | T07 | C10 |  91.0
 S52 | T06 | C11 |  90.0
 S52 | T05 | C05 |
 S52 | T01 | C08 |  64.0
 S52 | T02 | C09 |  81.0
 222 | T05 | C05 |  90.0
(26 rows)
```

> 由于`TeachB`在前面`GRANT SELECT ON TABLE Teachers,Students, Courses,STC to JxmsB,TeachB, StuB WITH GRANT OPTION`;他有查询STC任何字段的权限。

```
UPDATE ahweischema.STC SET Score= Score+0.5 WHERE Sno='S01';
GRANT insert,update(score),select(tno,cno,score) on ahweischema.STC to TeachB;
```

```
ahweidb=> UPDATE ahweischema.STC SET Score= Score+0.5 WHERE Sno='S01';
ERROR:  permission denied for relation stc
ahweidb=> GRANT insert,update(score),select(tno,cno,score) on ahweischema.STC to TeachB;
ERROR:  permission denied for relation stc
ahweidb=>
```

为了真正起到不让`TeachB`查看`sno`的目的，必须以系统管理员身份执行：`revoke select on STC from TeachB`此时`SELECT * FROM ahweischema.STC`;




```
ahweidb=> select * from stc;
ERROR:  permission denied for relation stc
```

而`select tno,cno,score from STC;`才可以。

```
ahweidb=> select tno,cno,score from STC;
 tno | cno | score
-----+-----+-------
 T01 | C01 |  83.0
 T03 | C03 |  85.0
 T01 | C01 |  75.0
 T02 | C02 |  45.0
 T03 | C03 |
 T04 | C04 |
 T05 | C05 |  70.0
 T04 | C06 |  83.0
 T05 | C07 |  90.0
 T01 | C08 |  83.0
 T02 | C09 |  77.0
 T07 | C10 |  83.0
 T06 | C11 |  88.0
 T01 | C08 |  63.0
 T02 | C02 |  93.0
 T01 | C01 |  78.0
 T06 | C06 |  89.0
 T05 | C05 |  93.0
 T07 | C10 |  45.0
 T04 | C04 |  86.0
 T07 | C10 |  91.0
 T06 | C11 |  90.0
 T05 | C05 |
 T01 | C08 |  64.0
 T02 | C09 |  81.0
 T05 | C07 |  53.0
 T05 | C03 |  83.0
(27 rows)

ahweidb=>
```

(7) 用户JxmsB授予用户TeachB对表Students插入和更新的权限，并且授予用户TeachB传播插入操作的权利。分别以JxmsB和TeachB的身份登陆，用SQL语言验证以上授权操作，结果如何?

1. 以jxmsb身份登录系统

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U jxmsb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

2. `Grant insert on table ahweischema.students to TeachB with grant option;`

```
ahweidb=> Grant insert on table ahweischema.students to TeachB with grant option;
GRANT
```

3. `Grant update on table ahweischema.students to TeachB;`

```
ahweidb=> Grant update on table ahweischema.students to TeachB;
GRANT
```

4. 退出，以TeachB登录

```
ahweidb=> \q
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U teachb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

(8) 收回用户`JxmsB`对表`Courses`查询权限的授权。分别以`JxmsB`和`TeachB`的身份登陆，用`SQL`语言查询`Courses`表，查询结果如何?

```
ahweidb=> revoke select on courses from jxmsb;
REVOKE
```

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U jxmsb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from courses;
 cno |      cname       | ccredit
-----+------------------+---------
 C01 | C++              |     4.0
 C02 | UML              |     4.0
 C03 | JAVA             |     3.0
 C04 | 算法分析与设计   |     3.0
 C05 | 数据库原理及应用 |     3.0
 C06 | 数据结构与算法   |     4.0
 C07 | 计算机组成原理   |     4.0
 C08 | 英语             |     6.0
 C09 | 数字生活         |     2.0
 C10 | 音乐鉴赏         |     2.0
 C11 | 体育1            |     2.0
(11 rows)
```

```
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U teachb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> select * from courses;
 cno |      cname       | ccredit
-----+------------------+---------
 C01 | C++              |     4.0
 C02 | UML              |     4.0
 C03 | JAVA             |     3.0
 C04 | 算法分析与设计   |     3.0
 C05 | 数据库原理及应用 |     3.0
 C06 | 数据结构与算法   |     4.0
 C07 | 计算机组成原理   |     4.0
 C08 | 英语             |     6.0
 C09 | 数字生活         |     2.0
 C10 | 音乐鉴赏         |     2.0
 C11 | 体育1            |     2.0
(11 rows)
```

(9)由上面（6）和（7）的授权，再由用户TeachB对用户StuB授予表Students插入和更新的权限，并且授予用户StuB传播插入操作的权力。这时候，如果由StuB对JxmsB授予表Students的插入和更新权限是否成功？如果能够成功，那么如果由用户TeachB取消StuB的权限，对JxmsB会有什么影响？如果再由XXXuser取消JxmsB的权限，对TeachB有什么影响？

- TeachB执行：

```
Grant insert on table ahweischema.students to stub with grant option;
Grant update on table ahweischema.students to stub;
```

```
ahweidb=> Grant insert on table ahweischema.students to stub with grant option;
GRANT
ahweidb=> Grant update on table ahweischema.students to stub;
WARNING:  no privileges were granted for "students"
GRANT
ahweidb=>
```

- StuB执行：

```
Grant insert on table ahweischema.students to jxmsb;
```

```
ahweidb=> Grant insert on table ahweischema.students to jxmsb;
GRANT
```

- TeachB取消StuB的权限：

```
revoke insert on table ahweischema.students from stub;
```

```
ERROR:  dependent privileges exist
HINT:  Use CASCADE to revoke them too.
```

```
ahweidb=> \q
[omm@db1 ~]$ gsql -d ahweidb -p 26000 -r -U teachb  -W Bigdata@123
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

ahweidb=> revoke insert on table ahweischema.students from stub;
ERROR:  dependent privileges exist
HINT:  Use CASCADE to revoke them too.
```

```
revoke insert on table ahweischema.students from stub CASCADE;
```

```
ahweidb=> revoke insert on table ahweischema.students from stub CASCADE;
REVOKE
ahweidb=>
```

`JxmsB`的插入权限没有受到影响，原因，这是两次权利，`jxmsb`还有`ahweiuser`授权的`insert`权限:
如果再由`ahweiuser`取消`JxmsB`的权限，也必须用`CASCADE`。