数据库系统课程实验报告

实验名称：	数据库安全性
实验日期：	2022.11.25
实验地点：	四号楼
提交日期：	2022.11.28

学号：	22920202202877
姓名：	陈鑫蕾
专业年级：	数媒 2020 级
学年学期：	2022-2023 学年第一学期

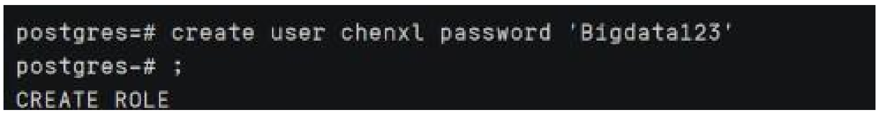
1. 实验目的

- 理解数据库系统用户（user）、权限（privilege）和角色（role）的概念和作用
- 熟练掌握用户的管理：创建、查看、删除和权限的授予与回收
- 熟练掌握通过数据字典查看用户权限、表和视图权限的方法
- 熟练掌握使用 Grant 命令给用户、角色授权的方法
- 熟练掌握使用 Revoke 命令回收已授权限的方法
- 熟练掌握角色定义、重命名和删除的方法
- 熟练掌握修改角色中权限的方法
- 理解视图的安全性作用

2. 实验内容和步骤

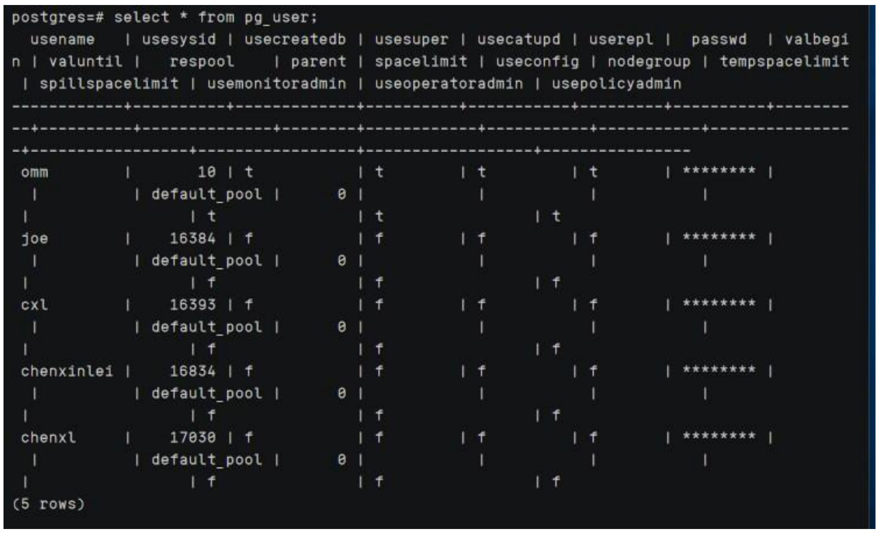
1.完成 https://bokai.blog.csdn.net/article/details/117912175 的内容。

(1)创建用户 chenxl



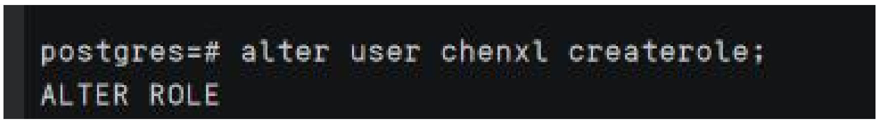
```
postgres=# create user chenxl password 'Bigdata123'
postgres-# ;
CREATE ROLE
```

（2）查看用户列表



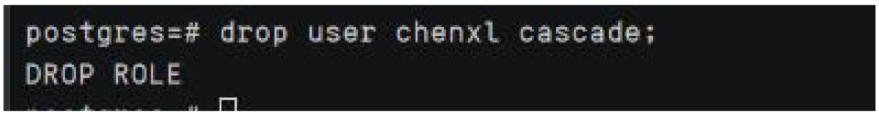
```
postgres=# select * from pg_user;
  usename    | usesysid | usecreatedb | usesuper | usecatupd | userepl |  passwd   | valbegi
n | valuntil |    respool     | parent | spacelimit | useconfig | nodegroup | tempspacelimit
 | spillspacelimit | usemonitoradmin | useoperatoradmin | usepolicyadmin
------------+----------+-------------+----------+-----------+---------+----------+--------
--+----------+--------------+--------+------------+-----------+-----------+--------------
-+-----------------+-----------------+------------------+----------------
 omm        |       10 | t           | t        | t         | t       | ******** |
  |          | default_pool |      0 |            |           |           |
 |           | t           | t                | t
 joe        |    16384 | f           | f        | f         | f       | ******** |
  |          | default_pool |      0 |            |           |           |
 |           | f           | f                | f
 cxl        |    16393 | f           | f        | f         | f       | ******** |
  |          | default_pool |      0 |            |           |           |
 |           | f           | f                | f
 chenxinlei |    16834 | f           | f        | f         | f       | ******** |
  |          | default_pool |      0 |            |           |           |
 |           | f           | f                | f
 chenxl     |    17030 | f           | f        | f         | f       | ******** |
  |          | default_pool |      0 |            |           |           |
 |           | f           | f                | f
(5 rows)
```

（3）为用户 chenxl 追加有创建角色的 CREATE ROLE 权限



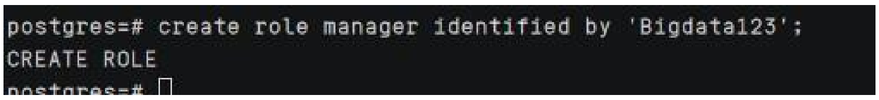
```
postgres=# alter user chenxl createrole;
ALTER ROLE
```

（4）删除用户



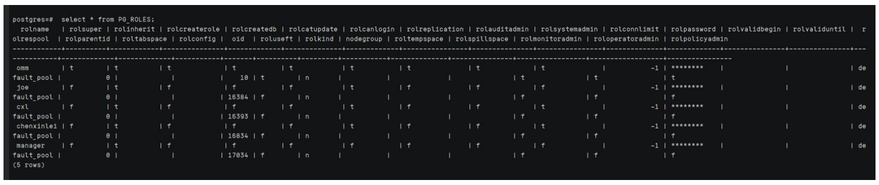
```
postgres=# drop user chenxl cascade;
DROP ROLE
```

（5）创建一个角色名为 manager，密码为 Bigdata123



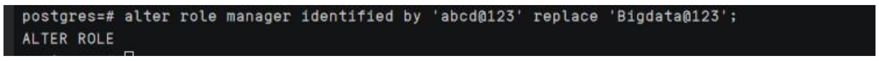
```
postgres=# create role manager identified by 'Bigdata123';
CREATE ROLE
```

（6）查看角色



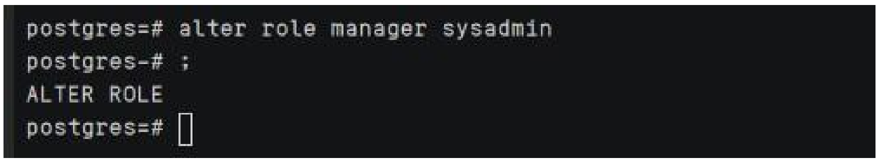
```
postgres=#  select * from PG_ROLES;
  rolname   | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rolpassword | rolvalidbegin | rolvaliduntil |  r
olrespool   | rolparentid | roltabspace | rolconfig |  oid   | roluseft | rolkind | nodegroup | roltempspace | rolspillspace | rolmonitoradmin | roloperatoradmin | rolpolicyadmin
-----------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+-------------+---------------+---------------+---
-----------+-------------+-------------+-----------+-------+----------+---------+-----------+--------------+---------------+-----------------+------------------+----------------
 omm        | t        | t          | t             | t           | t            | t           | t              | t             | t              |           -1 | ********    |               |               | de
fault_pool |         0 |             |           |    10 | t        | n       |           |              |               | t               | t                | t
 joe        | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | ********    |               |               | de
fault_pool |         0 |             |           | 16384 | f        | n       |           |              |               | f               | f                | f
 cxl        | f        | t          | f             | f           | f            | t           | f              | f             | t              |           -1 | ********    |               |               | de
fault_pool |         0 |             |           | 16393 | f        | n       |           |              |               | f               | f                | f
 chenxinlei | f        | t          | f             | f           | f            | t           | f              | f             | t              |           -1 | ********    |               |               | de
fault_pool |         0 |             |           | 16834 | f        | n       |           |              |               | f               | f                | f
 manager    | f        | t          | f             | f           | f            | f           | f              | f             | f              |           -1 | ********    |               |               | de
fault_pool |         0 |             |           | 17034 | f        | n       |           |              |               | f               | f                | f
(5 rows)
```

（7）修改角色 manager 密码为 abcd@123



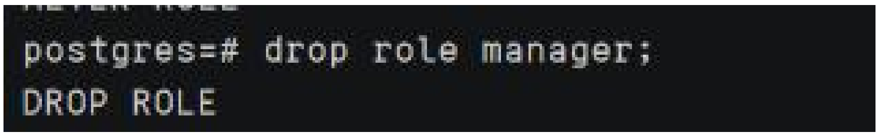
```
postgres=# alter role manager identified by 'abcd@123' replace 'Bigdata@123';
ALTER ROLE
```

（8）修改角色 manager 为系统管理员



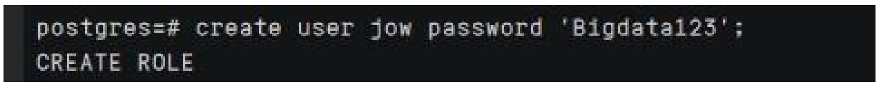
```
postgres=# alter role manager sysadmin
postgres-# ;
ALTER ROLE
postgres=# 
```

（9）删除角色 manager



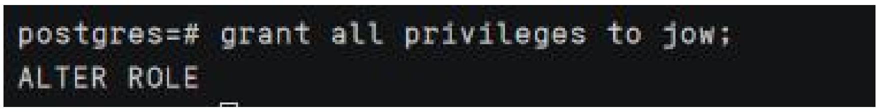
```
postgres=# drop role manager;
DROP ROLE
```

（10）创建名为 jow 的用户



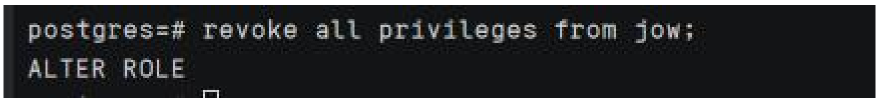
```
postgres=# create user jow password 'Bigdata123';
CREATE ROLE
```

（11）将 sysadmin 权限授权为 jow



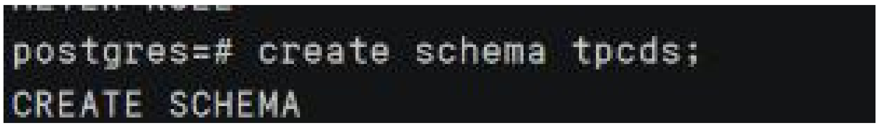
```
postgres=# grant all privileges to jow;
ALTER ROLE
```

（12）撤销 jow 用户的 syadmin 权限



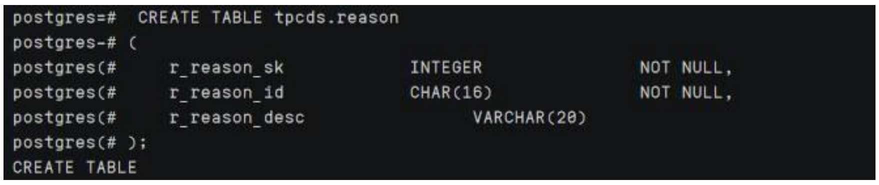
```
postgres=# revoke all privileges from jow;
ALTER ROLE
```

（13）创建 tpcds 模式



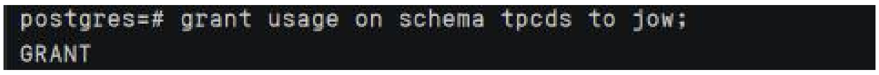
```
postgres=# create schema tpcds;
CREATE SCHEMA
```

（14）tpcds 模式下创建一张 reason 表

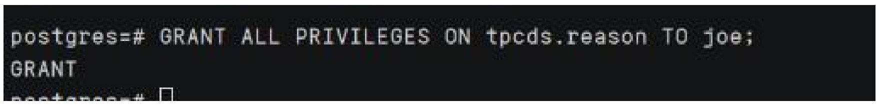


```
postgres=#  CREATE TABLE tpcds.reason
postgres-# (
postgres(#     r_reason_sk            INTEGER               NOT NULL,
postgres(#     r_reason_id            CHAR(16)              NOT NULL,
postgres(#     r_reason_desc                   VARCHAR(20)
postgres(# );
CREATE TABLE
```

（15）将模式 tpcds 的使用权限和表 tpcds.reason 的所有权限授权给用户 jow

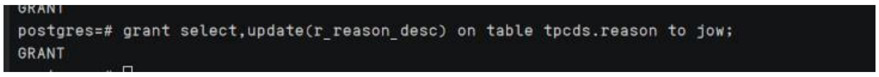


```
postgres=# grant usage on schema tpcds to jow;
GRANT
```



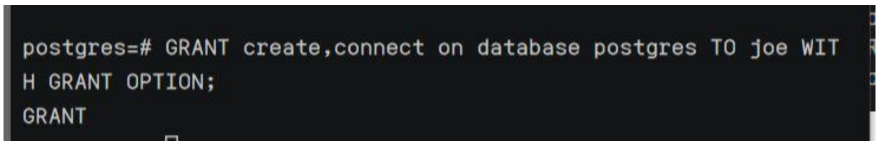
```
postgres=# GRANT ALL PRIVILEGES ON tpcds.reason TO joe;
GRANT
```

（16）将 tpcds.reason 表中 r_reason_sk、r_reason_id、r_reason_desc 列的查询权限，r_reason_desc 的更新权限授权给 jow



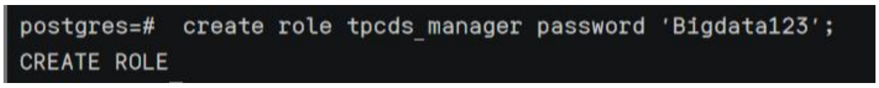
```
GRANT
postgres=# grant select,update(r_reason_desc) on table tpcds.reason to jow;
GRANT
```

（17）将数据库 postgres 的连接权限授权给用户 jow，并给予其在 postgres 中创建 schema 的权限，而且允许 jow 将此权限授权给其他用户



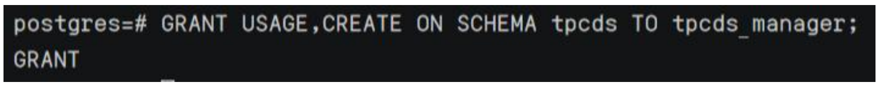
```
postgres=# GRANT create,connect on database postgres TO joe WIT
H GRANT OPTION;
GRANT
```

（18）创建角色 tpcds_manager



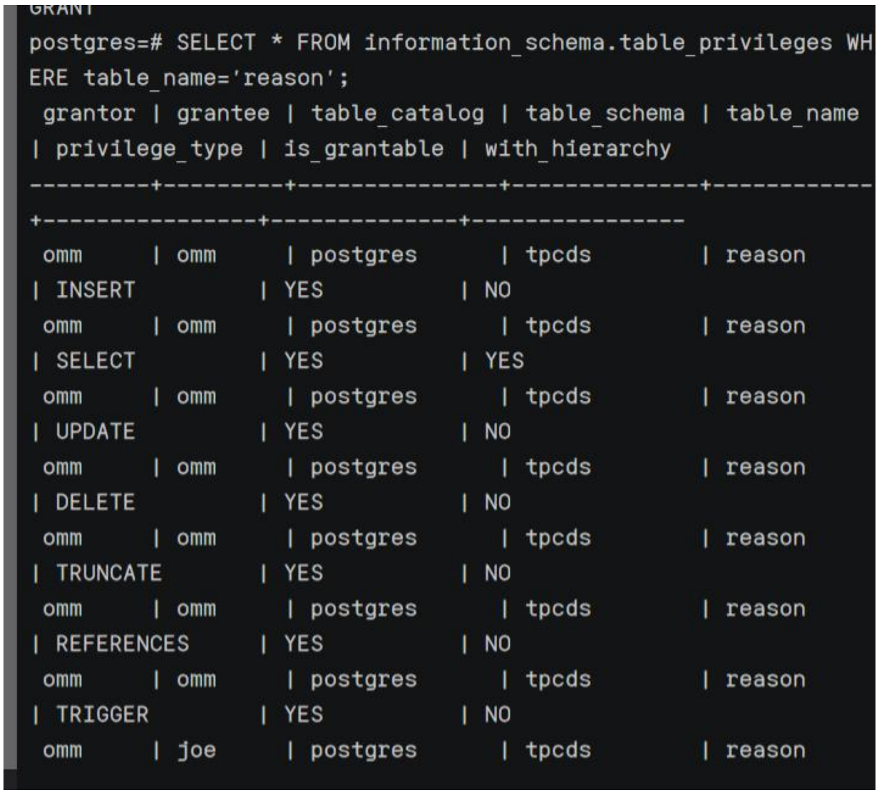
```
postgres=#  create role tpcds_manager password 'Bigdata123';
CREATE ROLE
```

（19）将模式 tpcds 的访问权限授权给角色 tpcds_manager，并授予该角色在 tpcds 下创建对象的权限，不允许该角色中的用户将权限授权给其人

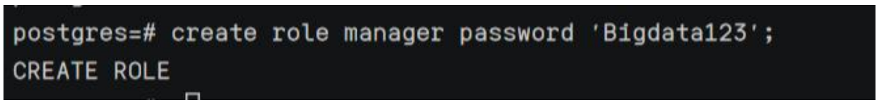


```
postgres=# GRANT USAGE,CREATE ON SCHEMA tpcds TO tpcds_manager;
GRANT
```

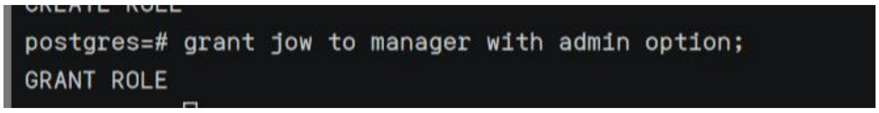
（20）查看表 reason 权限


```
GRANT
postgres=# SELECT * FROM information_schema.table_privileges WH
ERE table_name='reason';
 grantor | grantee | table_catalog | table_schema | table_name
| privilege_type | is_grantable | with_hierarchy
---------+---------+---------------+--------------+------------
+----------------+--------------+----------------
 omm     | omm     | postgres      | tpcds        | reason
| INSERT         | YES          | NO
 omm     | omm     | postgres      | tpcds        | reason
| SELECT         | YES          | YES
 omm     | omm     | postgres      | tpcds        | reason
| UPDATE         | YES          | NO
 omm     | omm     | postgres      | tpcds        | reason
| DELETE         | YES          | NO
 omm     | omm     | postgres      | tpcds        | reason
| TRUNCATE       | YES          | NO
 omm     | omm     | postgres      | tpcds        | reason
| REFERENCES     | YES          | NO
 omm     | omm     | postgres      | tpcds        | reason
| TRIGGER        | YES          | NO
 omm     | joe     | postgres      | tpcds        | reason
```

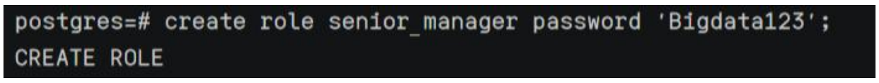
（21）创建角色 manager


```
postgres=# create role manager password 'Bigdata123';
CREATE ROLE
```

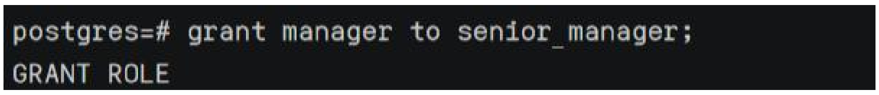
（22）将 joe 的权限授权给 manager，并允许该角色将权限授权给其他人


```
CREATE ROLE
postgres=# grant jow to manager with admin option;
GRANT ROLE
```

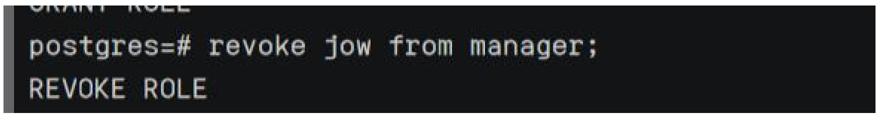
（23）创建用户 senior_manager


```
postgres=# create role senior_manager password 'Bigdata123';
CREATE ROLE
```

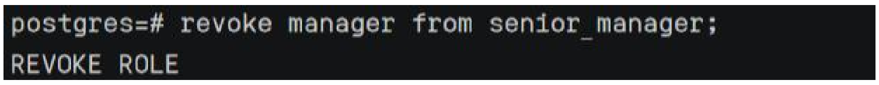
（24）将用户 manager 的权限授权给该用户


```
postgres=# grant manager to senior_manager;
GRANT ROLE
```

（25）回收 manager 权限

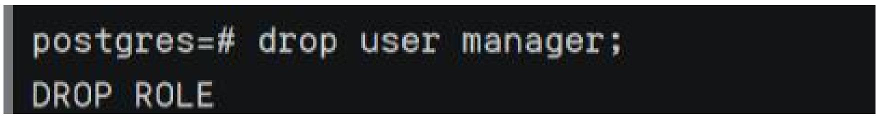


```
postgres=# revoke jow from manager;
REVOKE ROLE
```



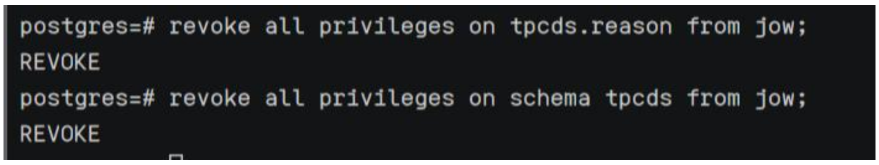
```
postgres=# revoke manager from senior_manager;
REVOKE ROLE
```

（26）删除用户 manager

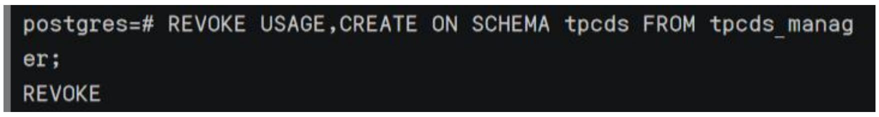


```
postgres=# drop user manager;
DROP ROLE
```

（27）回收 tpcds_manager 权限

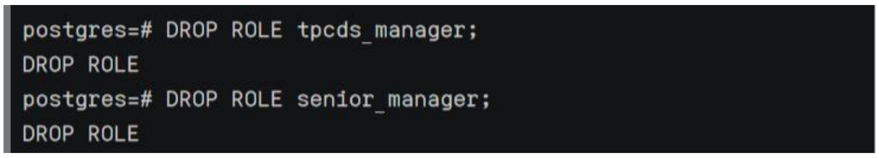


```
postgres=# revoke all privileges on tpcds.reason from jow;
REVOKE
postgres=# revoke all privileges on schema tpcds from jow;
REVOKE
```

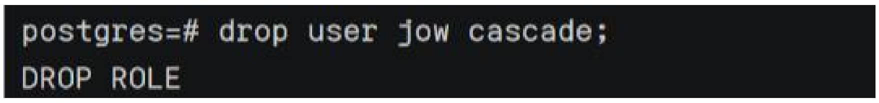


```
postgres=# REVOKE USAGE,CREATE ON SCHEMA tpcds FROM tpcds_manag
er;
REVOKE
```

（28）删除 tpcds_manager 用户，删除 senior_manager 用户，删除 jow 用户

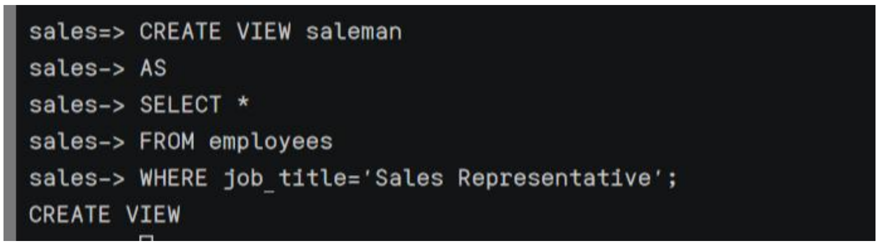


```
postgres=# DROP ROLE tpcds_manager;
DROP ROLE
postgres=# DROP ROLE senior_manager;
DROP ROLE
```



```
postgres=# drop user jow cascade;
DROP ROLE
```

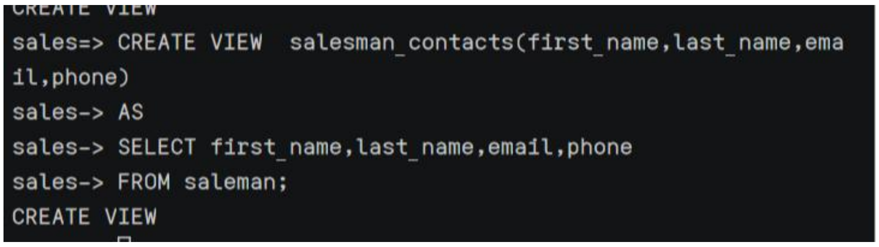
2.创建视图 salesman，该视图只保存 employees 表中所有 job_title 为 'Sales Representative'的雇员。



```
sales=> CREATE VIEW saleman
sales-> AS
sales-> SELECT *
sales-> FROM employees
sales-> WHERE job_title='Sales Representative';
CREATE VIEW
```

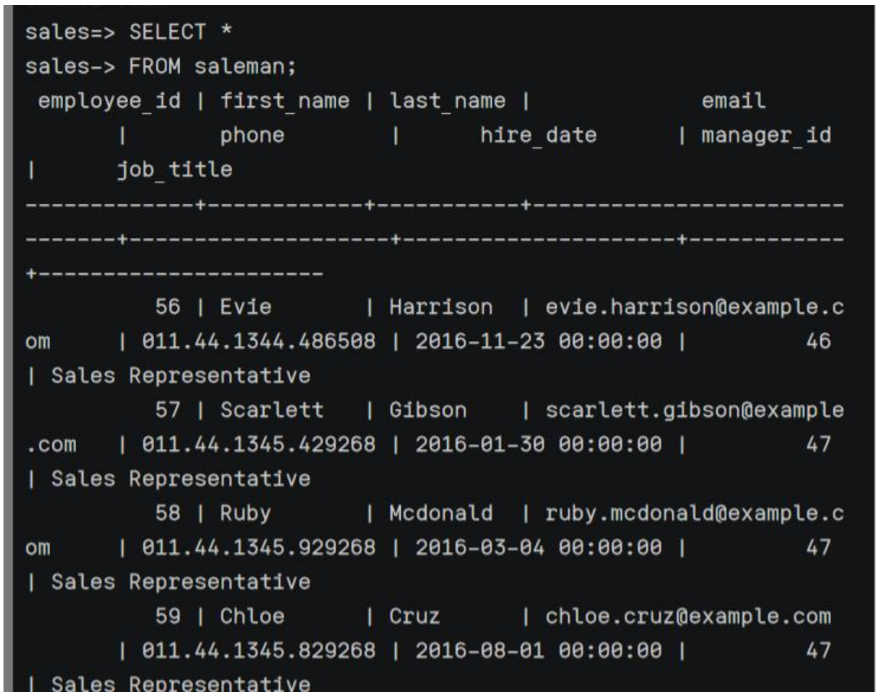
3.创建基于 salesman 的视图 salesman_contacts(first_name,last_name,email,phone)，该视图存储的

salesman 的联系方式。

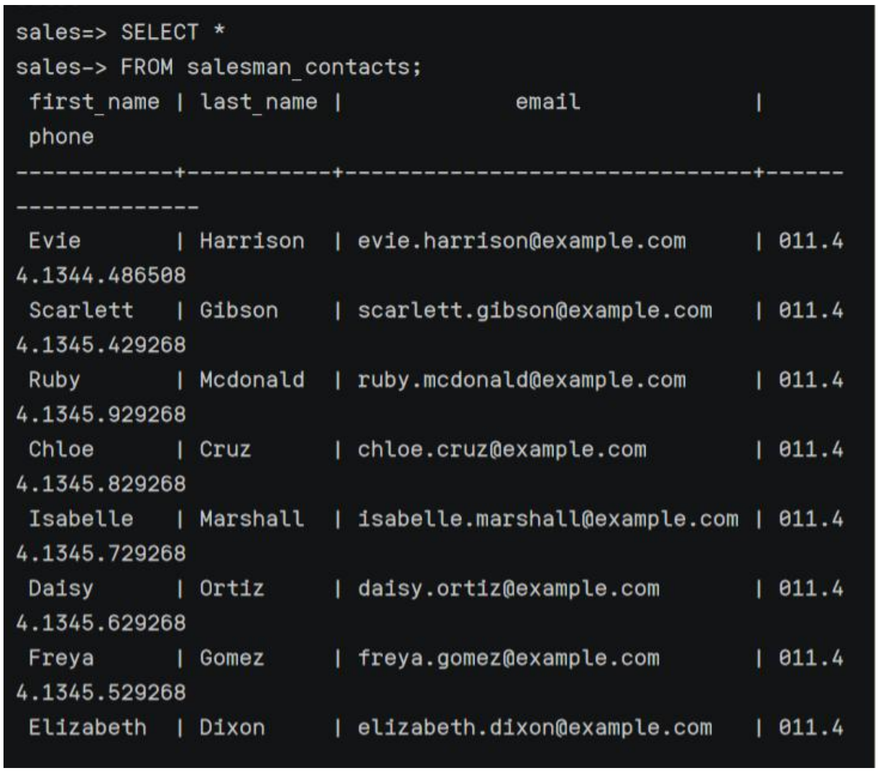


```
CREATE VIEW
sales=> CREATE VIEW  salesman_contacts(first_name,last_name,ema
il,phone)
sales-> AS
sales-> SELECT first_name,last_name,email,phone
sales-> FROM saleman;
CREATE VIEW
```

4.查询视图 salesman 和 salesman_contacts。

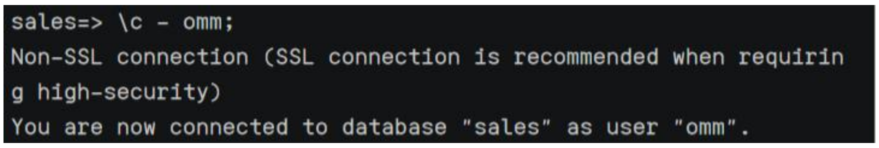


```
sales=> SELECT *
sales-> FROM saleman;
 employee_id | first_name | last_name |            email
       |       phone        |      hire_date      | manager_id
|      job_title
-------------+------------+-----------+------------------------
-------+--------------------+---------------------+------------
+----------------------
          56 | Evie       | Harrison  | evie.harrison@example.c
om      | 011.44.1344.486508 | 2016-11-23 00:00:00 |         46
| Sales Representative
          57 | Scarlett   | Gibson    | scarlett.gibson@example
.com    | 011.44.1345.429268 | 2016-01-30 00:00:00 |         47
| Sales Representative
          58 | Ruby       | Mcdonald  | ruby.mcdonald@example.c
om      | 011.44.1345.929268 | 2016-03-04 00:00:00 |         47
| Sales Representative
          59 | Chloe      | Cruz      | chloe.cruz@example.com
       | 011.44.1345.829268 | 2016-08-01 00:00:00 |         47
| Sales Representative
```



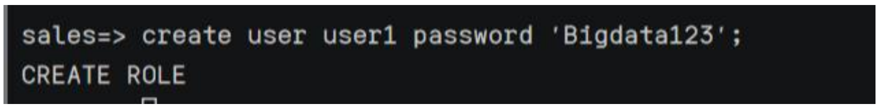
```
sales=> SELECT *
sales-> FROM salesman_contacts;
 first_name | last_name |              email             |
 phone
------------+-----------+--------------------------------+------
--------------
 Evie       | Harrison  | evie.harrison@example.com      | 011.4
4.1344.486508
 Scarlett   | Gibson    | scarlett.gibson@example.com    | 011.4
4.1345.429268
 Ruby       | Mcdonald  | ruby.mcdonald@example.com      | 011.4
4.1345.929268
 Chloe      | Cruz      | chloe.cruz@example.com         | 011.4
4.1345.829268
 Isabelle   | Marshall  | isabelle.marshall@example.com  | 011.4
4.1345.729268
 Daisy      | Ortiz     | daisy.ortiz@example.com        | 011.4
4.1345.629268
 Freya      | Gomez     | freya.gomez@example.com        | 011.4
4.1345.529268
 Elizabeth  | Dixon     | elizabeth.dixon@example.com    | 011.4
```

5.在当前窗口输入命令：\c - omm 切换到 omm 用户。



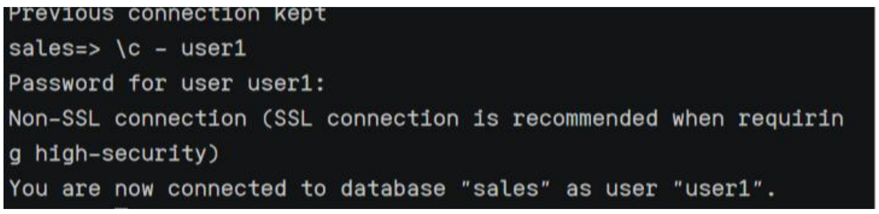
```
sales=> \c - omm;
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
You are now connected to database "sales" as user "omm".
```

6.创建新用户 user1。



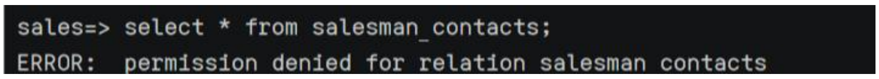
```
sales=> create user user1 password 'Bigdata123';
CREATE ROLE
```

7.在当前窗口输入命令：\c －user1 切换到 user1 用户。



```
Previous connection kept
sales=> \c - user1
Password for user user1:
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
You are now connected to database "sales" as user "user1".
```

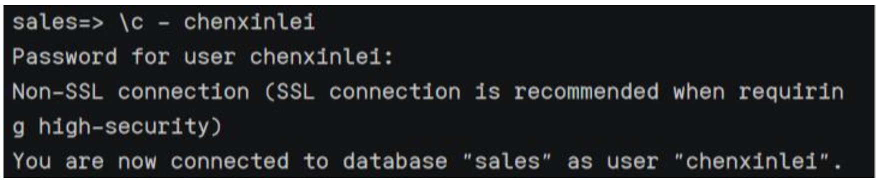
8.发布查询命令：select * from salesman_contacts;观察结果。



```
sales=> select * from salesman_contacts;
ERROR:  permission denied for relation salesman_contacts
```

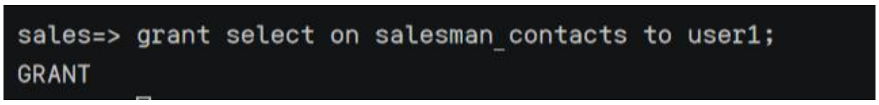
报错：无权限

9.发布命令：\c － chenxinlei 切换到 chenxinlei 用户



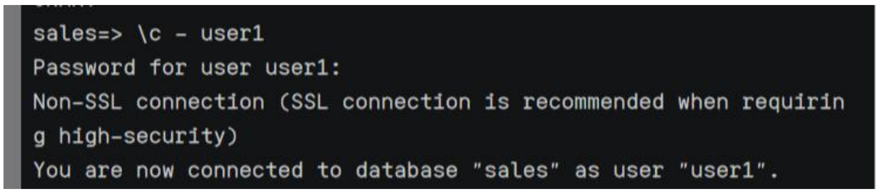
```
sales=> \c - chenxinlei
Password for user chenxinlei:
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
You are now connected to database "sales" as user "chenxinlei".
```

10.在当前 chenxinlei 用户下输入命令：grant select on alesman_contacts to user1；实现授权操作。

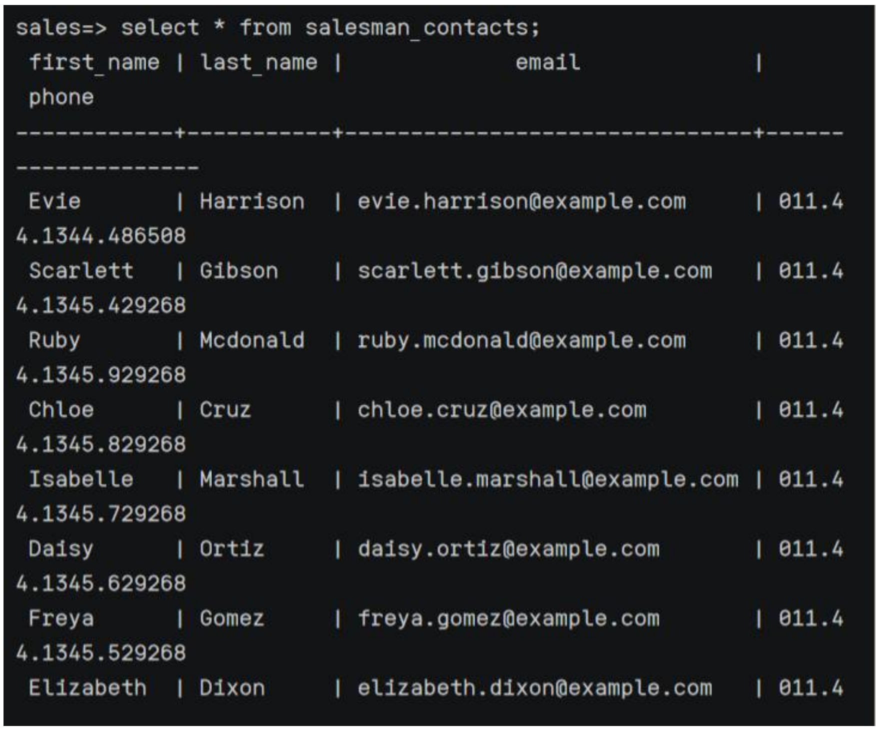


```
sales=> grant select on salesman_contacts to user1;
GRANT
```

11.依次重复步骤（7）和（8），比较两次查询的结果。



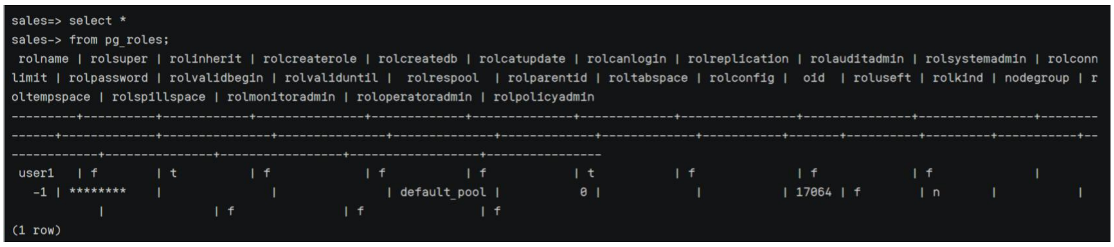
```
sales=> \c - user1
Password for user user1:
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
You are now connected to database "sales" as user "user1".
```



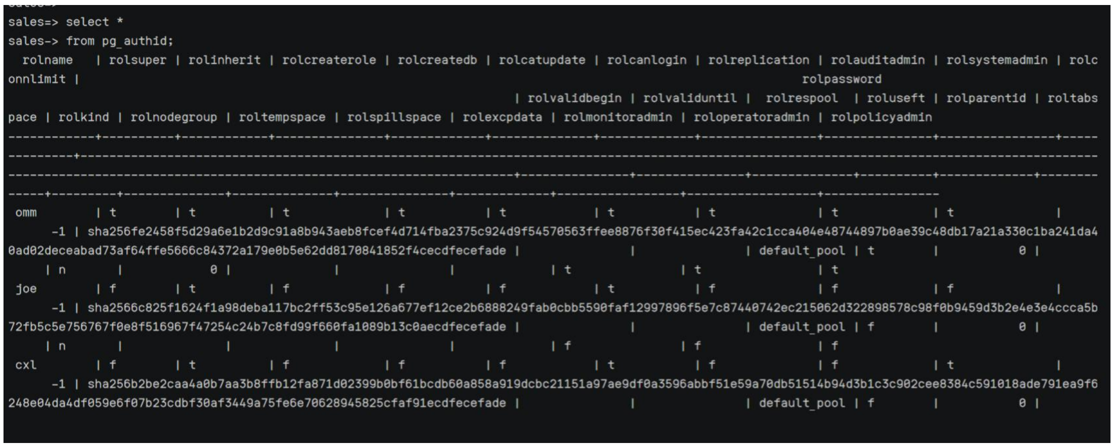
```
sales=> select * from salesman_contacts;
 first_name | last_name |              email              |
 phone
------------+-----------+---------------------------------+------
--------------
 Evie       | Harrison  | evie.harrison@example.com       | 011.4
4.1344.486508
 Scarlett   | Gibson    | scarlett.gibson@example.com     | 011.4
4.1345.429268
 Ruby       | Mcdonald  | ruby.mcdonald@example.com       | 011.4
4.1345.929268
 Chloe      | Cruz      | chloe.cruz@example.com          | 011.4
4.1345.829268
 Isabelle   | Marshall  | isabelle.marshall@example.com   | 011.4
4.1345.729268
 Daisy      | Ortiz     | daisy.ortiz@example.com         | 011.4
4.1345.629268
 Freya      | Gomez     | freya.gomez@example.com         | 011.4
4.1345.529268
 Elizabeth  | Dixon     | elizabeth.dixon@example.com     | 011.4
```

此时可以查询 salesman_contacts,因为已经给予 user1 查询权限

12.查看与角色、权限相关的系统表和系统视图：pg_roles，pg_authid。



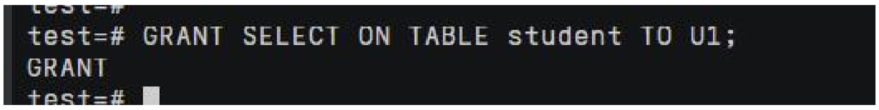
```
sales=> select *
sales-> from pg_roles;
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconn
limit | rolpassword | rolvalidbegin | rolvaliduntil |  rolrespool  | rolparentid | roltabspace | rolconfig |  oid  | roluseft | rolkind | nodegroup | r
oltempspace | rolspillspace | rolmonitoradmin | roloperatoradmin | rolpolicyadmin
---------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------
------+-------------+---------------+---------------+--------------+-------------+-------------+-----------+-------+----------+---------+-----------+--
------------+---------------+------------------+------------------+----------------
 user1   | f        | t          | f             | f           | f            | t           | f              | f             | f              |
   -1 | ********    |             |               | default_pool |           0 |             |           | 17064 | f        | n       |           |
           |               | f                | f                | f
(1 row)
```



```
sales=> select *
sales-> from pg_authid;
  rolname   | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolc
onnlimit |                                                                                   rolpassword
                                                     | rolvalidbegin | rolvaliduntil |  rolrespool  | roluseft | rolparentid | roltabs
pace | rolkind | rolnodegroup | roltempspace | rolspillspace | rolexcpdata | rolmonitoradmin | roloperatoradmin | rolpolicyadmin
-----------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+-----
---------+----------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------+---------------+---------------+--------------+----------+-------------+--------
-----+---------+--------------+--------------+---------------+-------------+-----------------+------------------+----------------
 omm       | t        | t          | t             | t           | t            | t           | t              | t             | t              |
      -1 | sha256fe2458f5d29a6e1b2d9c91a8b943aeb8fcef4d714fba2375c924d9f54570563ffee8876f30f415ec423fa42c1cca404e48744897b0ae39c48db17a21a330c1ba241da4
0ad02deceabad73af64ffe5666c84372a179e0b5e62dd8170841852f4cecdfecefade |               |               | default_pool | t        |           0 |
    | n       |            0 |              |               |             | t               | t                | t
 joe       | f        | t          | f             | f           | f            | t           | f              | f             | f              |
      -1 | sha2566c825f1624f1a98deba117bc2ff53c95e126a677ef12ce2b6888249fab0cbb5590faf12997896f5e7c87440742ec215062d322898578c98f0b9459d3b2e4e3e4ccca5b
72fb5c5e756767f0e8f516967f47254c24b7c8fd99f660fa1089b13c0aecdfecefade |               |               | default_pool | f        |           0 |
    | n       |              |              |               |             | f               | f                | f
 cxl       | f        | t          | f             | f           | f            | t           | f              | f             | t              |
      -1 | sha256b2be2caa4a0b7aa3b8ffb12fa871d02399b0bf61bcdb60a858a919dcbc21151a97ae9df0a3596abbf51e59a70db51514b94d3b1c3c902cee8384c591018ade791ea9f6
248e04da4df059e6f07b23cdbf30af3449a75fe6e70628945825cfaf91ecdfecefade |               |               | default_pool | f        |           0 |
```

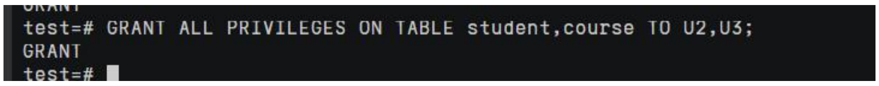
13.在完成（1）的基础上，重做教材中的[例 4.1-例 4.13]，因为 openGauss 的语法与教材上的不完全一致，可以通过以上实操加深对 openGauss 安全性控制机制的理解。

1.将 student 表的查询权限授给 u1



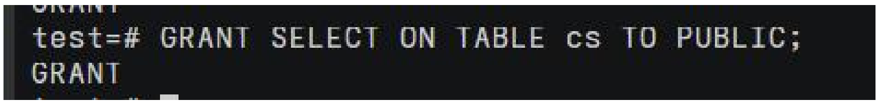
```
test=# GRANT SELECT ON TABLE student TO U1;
GRANT
test=#
```

2.对 student 和 course 的所有权限授予 u2 和 u3



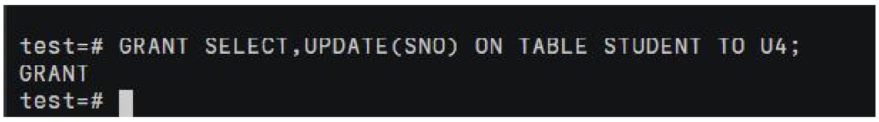
```
test=# GRANT ALL PRIVILEGES ON TABLE student,course TO U2,U3;
GRANT
test=#
```

3.对 sc 表的查询权限授予所有用户



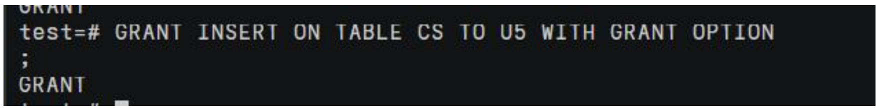
```
test=# GRANT SELECT ON TABLE cs TO PUBLIC;
GRANT
```

4.将查询 student 和修改学号的权限授予 u4



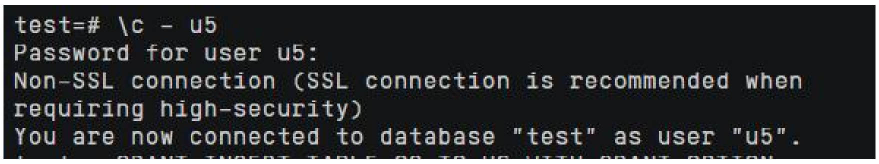
```
test=# GRANT SELECT,UPDATE(SNO) ON TABLE STUDENT TO U4;
GRANT
test=#
```

5.把对 cs 表插入的权限授予 u5，并且可以再次授权

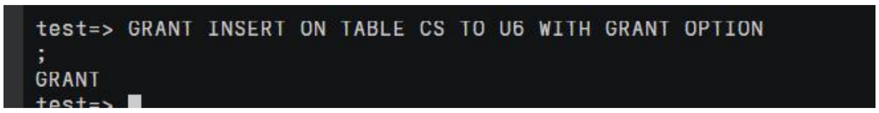


```
test=# GRANT INSERT ON TABLE CS TO U5 WITH GRANT OPTION
;
GRANT
```

6.切换到 u5，授权给 u6

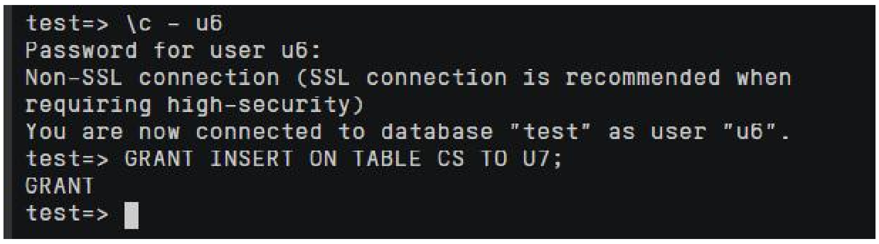


```
test=# \c - u5
Password for user u5:
Non-SSL connection (SSL connection is recommended when
requiring high-security)
You are now connected to database "test" as user "u5".
```



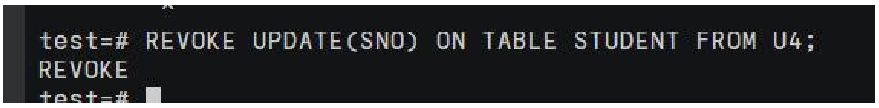
```
test=> GRANT INSERT ON TABLE CS TO U6 WITH GRANT OPTION
;
GRANT
```

7.切换到 u6，授权给 u7



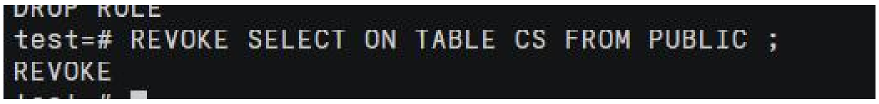
```
test=> \c - u6
Password for user u6:
Non-SSL connection (SSL connection is recommended when
requiring high-security)
You are now connected to database "test" as user "u6".
test=> GRANT INSERT ON TABLE CS TO U7;
GRANT
test=>
```

8.把用户 u4 的权限收回



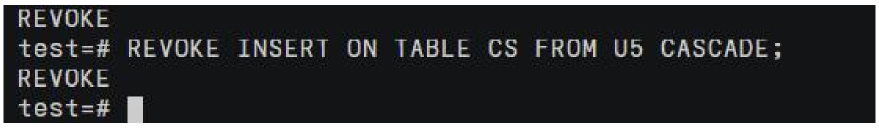
```
test=# REVOKE UPDATE(SNO) ON TABLE STUDENT FROM U4;
REVOKE
```

9.收回对 cs 的查询权限



```
test=# REVOKE SELECT ON TABLE CS FROM PUBLIC ;
REVOKE
```

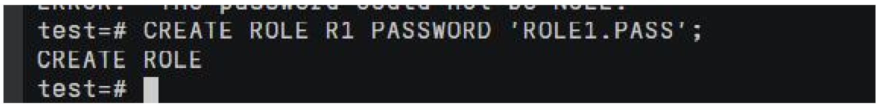
10.把 u5 的权限收回



```
REVOKE
test=# REVOKE INSERT ON TABLE CS FROM U5 CASCADE;
REVOKE
test=#
```

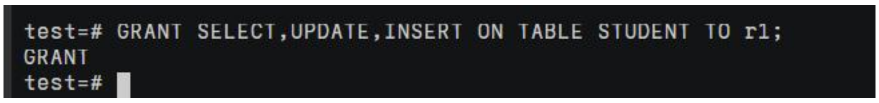
11.创建角色来授权用户

11.1 新建角色

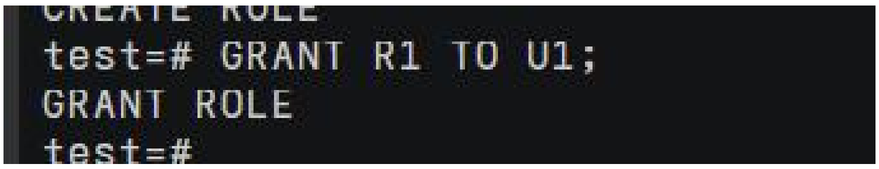


```
test=# CREATE ROLE R1 PASSWORD 'ROLE1.PASS';
CREATE ROLE
test=#
```

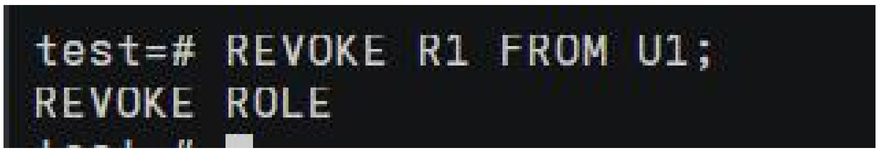
11.2 授权角色


```
test=# GRANT SELECT,UPDATE,INSERT ON TABLE STUDENT TO r1;
GRANT
test=#
```

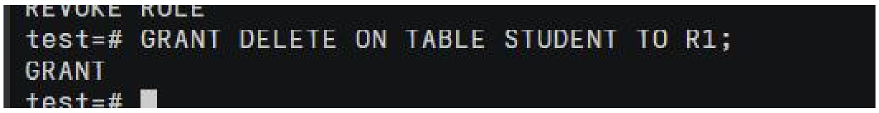
11.3 授权


```
CREATE ROLE
test=# GRANT R1 TO U1;
GRANT ROLE
test=#
```

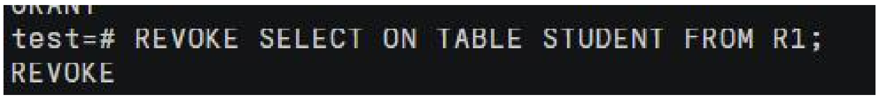
11.4 回收


```
test=# REVOKE R1 FROM U1;
REVOKE ROLE
```

12.修改权限


```
REVOKE ROLE
test=# GRANT DELETE ON TABLE STUDENT TO R1;
GRANT
test=#
```

13.修改权限


```
GRANT
test=# REVOKE SELECT ON TABLE STUDENT FROM R1;
REVOKE
```

思考:

只有拥有 DBA 权限的用户才能创建新用户

角色是拥有数据库对象和权限的实体,在不同的环境中可以认为是一个或一组用户

Grant 和 revoke 可以实现角色权限的修改

3. 实验总结

3.1 完成的工作

创建视图

创建用户，角色，对他们赋予不同的权限，收回他们的权限

3.2 对实验的认识

掌握了视图的创建；

新建用户、新建角色；

对用户和角色赋予不同的权限；

把用户的权限赋予角色；

角色的权限赋予用户；

理解视图的安全性作用；

3.3 遇到的困难及解决方法

无