数据库系统课程实验报告

实验名称:	实验六：数据库的安全性
实验日期:	2022/4/21
实验地点:	厦门大学德旺图书馆
提交日期:	2022/4/21

学号:	20420192201952
姓名:	庾晓萍
专业年级:	软工 2020 级
学年学期:	2021-2022 学年第二学期

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。

一、用户及角色

① 用户

· 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。

步骤：创建用户 jim，登录密码为 Bigdata@123。查看用户列表，为用

户 jim 追加有创建角色的 CREATE ROLE 权限，删除用户。

② 角色

·角色是拥有数据库对象和权限的实体。在不同的环境中角色可以认为是一个用户，一个组或者兼顾两者。

·在数据库中添加一个新角色，角色无登录权限。

·创建角色的用户必须具备 CREATE ROLE 的权限或者是系统管理员。

步骤：创建一个角色，名为 manager，密码为 Bigdata@123。查看角色，修改角色 manager 的密码为 abcd@123。修改角色 manager 为系统管理员，删除角色 manager。

```
postgres=# CREATE ROLE manager IDENTIFIED BY 'Bigdata@123';
CREATE ROLE
postgres=# SELECT * FROM PG_ROLES;
  rolname   | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate
| rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit |
rolpassword | rolvalidbegin | rolvaliduntil |  rolrespool  | rolparentid | roltab
space | rolconfig |  oid  | roluseft | rolkind | nodegroup | roltempspace | rolsp
illspace | rolmonitoradmin | roloperatoradmin | rolpolicyadmin
------------+----------+------------+---------------+-------------+--------------
+-------------+----------------+---------------+----------------+-------------+-
-----------+--------------+--------------+-------------+-------------+-------
------+-----------+-------+----------+---------+-----------+-------------+------
---------+-----------------+------------------+----------------
 omm        | t        | t          | t             | t           | t
| t           | t            | t             | t              |           -1 |
********    |              |               | default_pool |           0 |
      |           |    10 | t        | n       |           |             |
         | t               | t               | t
 yuxiaoping | f        | t          | f             | f           | f
| t           | f            | f             | f              |           -1 |
********    |              |               | default_pool |           0 |
      |           | 16576 | f        | n       |           |             |
         | f               | f               | f
 manager    | f        | t          | f             | f           | f
| f           | f            | f             | f              |           -1 |
********    |              |               | default_pool |           0 |
      |           | 16758 | f        | n       |           |             |
         | f               | f               | f
(3 rows)

postgres=# ALTER ROLE manager IDENTIFIED BY 'abcd@123' REPLACE 'Bigdata@123';
ALTER ROLE
postgres=# ALTER ROLE manager SYSADMIN;
ALTER ROLE
postgres=# DROP ROLE manager;
DROP ROLE
```

二、权限设置及回收

① 将系统权限授权给用户或者角色

步骤：创建名为 joe 的用户，将 sysadmin 权限授权给 joe，撤销 joe 用户的 sysadmin 权限。

```
postgres=# CREATE USER joe PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES TO joe;
ALTER ROLE
postgres=# REVOKE ALL PRIVILEGES FROM joe;
ALTER ROLE
```

② 将数据库对象授权给角色或用户

步骤：创建 tpcds 模式，tpcds 模式下创建一张 reason 表。

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

将模式 tpcds 的使用权限和表 tpcds.reason 的所有权限授权给用户 joe。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 joe。将数据库 postgres 的连接权限授权给用户 joe，并给予其在 postgres 中创建 schema 的权限，而且允许 joe 将此权限授权给其他用户。

```
postgres=# GRANT USAGE ON SCHEMA tpcds TO joe;
GRANT
postgres=# GRANT ALL PRIVILEGES ON tpcds.reason TO joe;
GRANT
postgres=# GRANT select (r_reason_sk,r_reason_id,r_reason_desc),update (r_reason_
desc) ON tpcds.reason TO joe;
GRANT
postgres=# GRANT create,connect on database postgres TO joe WITH GRANT OPTION;
GRANT
```

创建角色 tpcds_manager，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。查看表 reason 权限：

```
postgres=# CREATE ROLE tpcds_manager PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# GRANT USAGE,CREATE ON SCHEMA tpcds TO tpcds_manager;
GRANT
postgres=# SELECT * FROM information_schema.table_privileges WHERE 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     | INSERT         |
 NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | SELECT         |
 NO           | YES
 omm     | joe     | postgres      | tpcds        | reason     | UPDATE         |
 NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | DELETE         |
 NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | TRUNCATE       |
```

③ 将用户或者角色的权限授权给其他用户或角色

步骤：创建角色 manager，将 joe 的权限授权给 manager，并允许该角色将权限授权给其他人，创建用户 senior_manager，将用户 manager 的权限授权给该用户。

```
postgres=# CREATE ROLE manager PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=# GRANT joe TO manager WITH ADMIN OPTION;
GRANT ROLE
postgres=# CREATE ROLE senior_manager PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=#  GRANT manager TO senior_manager;
GRANT ROLE
postgres=#
```

④ 权限回收并清理用户

步骤：逐步回收 manager 权限，删除 manager 用户，逐步回收 joe 权限，逐步回收 tpcds_manager 权限，删除 tpcds_manager 用户，删除 senior_manager 用户，删除 joe 用户。

```
postgres=# REVOKE joe FROM manager;
REVOKE ROLE
postgres=# REVOKE manager FROM senior_manager;
REVOKE ROLE
postgres=# DROP USER manager;
DROP ROLE
postgres=# REVOKE ALL PRIVILEGES ON tpcds.reason FROM joe;
REVOKE
postgres=# REVOKE ALL PRIVILEGES ON SCHEMA tpcds FROM joe;
REVOKE
postgres=# REVOKE USAGE,CREATE ON SCHEMA tpcds FROM tpcds_manager;
REVOKE
postgres=# DROP ROLE tpcds_manager;
DROP ROLE
postgres=# DROP ROLE senior_manager;
DROP ROLE
postgres=# DROP USER joe 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。

```
# 创建视图 salesman，该视图只保存 employees 表中所有 job_title 为'Sales Representative'的雇员。
CREATE VIEW salesman AS
SELECT * FROM employees WHERE job_title='Sales Representative';
-- 查看
SELECT*FROM salesman;
```

```
sale=> CREATE VIEW salesman AS
sale-> SELECT * FROM employees WHERE job_title='Sales Representative';
CREATE VIEW
sale=> SELECT*FROM salesman;
 employee_id | first_name | last_name |             email             |       pho
ne         |      hire_date      | manager_id |      job_title
-------------+------------+-----------+-------------------------------+----------
----------+---------------------+------------+----------------------
          56 | Evie       | Harrison  | evie.harrison@example.com     | 011.44.13
44.486508 | 2016-11-23 00:00:00 |         46 | Sales Representative
          57 | Scarlett   | Gibson    | scarlett.gibson@example.com   | 011.44.13
45.429268 | 2016-01-30 00:00:00 |         47 | Sales Representative
          58 | Ruby       | Mcdonald  | ruby.mcdonald@example.com     | 011.44.13
45.929268 | 2016-03-04 00:00:00 |         47 | Sales Representative
          59 | Chloe      | Cruz      | chloe.cruz@example.com        | 011.44.13
45.829268 | 2016-08-01 00:00:00 |         47 | Sales Representative
          60 | Isabelle   | Marshall  | isabelle.marshall@example.com | 011.44.13
45.729268 | 2016-03-10 00:00:00 |         47 | Sales Representative
          61 | Daisy      | Ortiz     | daisy.ortiz@example.com       | 011.44.13
45.629268 | 2016-12-15 00:00:00 |         47 | Sales Representative
```

（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。

```
#创建基于 salesman 的视图，该视图存储salesman 的联系方式。
CREATE VIEW salesman_contacts AS
SELECT first_name, last_name, email, phone FROM salesman;
-- 查看
SELECT*FROM salesman_contacts;
```

```
sale=> CREATE VIEW salesman_contacts AS
sale-> SELECT first_name, last_name, email, phone FROM salesman;
CREATE VIEW
sale=> SELECT*FROM salesman_contacts;
 first_name | last_name |             email              |       phone
------------+-----------+--------------------------------+-------------------
 Evie       | Harrison  | evie.harrison@example.com      | 011.44.1344.486508
 Scarlett   | Gibson    | scarlett.gibson@example.com    | 011.44.1345.429268
 Ruby       | Mcdonald  | ruby.mcdonald@example.com      | 011.44.1345.929268
 Chloe      | Cruz      | chloe.cruz@example.com         | 011.44.1345.829268
 Isabelle   | Marshall  | isabelle.marshall@example.com  | 011.44.1345.729268
 Daisy      | Ortiz     | daisy.ortiz@example.com        | 011.44.1345.629268
 Freya      | Gomez     | freya.gomez@example.com        | 011.44.1345.529268
 Elizabeth  | Dixon     | elizabeth.dixon@example.com    | 011.44.1644.429262
 Florence   | Freeman   | florence.freeman@example.com   | 011.44.1346.229268
 Alice      | Wells     | alice.wells@example.com        | 011.44.1346.329268
 Charlotte  | Webb      | charlotte.webb@example.com     | 011.44.1346.529268
 Sienna     | Simpson   | sienna.simpson@example.com     | 011.44.1346.629268
 Matilda    | Stevens   | matilda.stevens@example.com    | 011.44.1346.729268
 Evelyn     | Tucker    | evelyn.tucker@example.com      | 011.44.1343.929268
 Eva        | Porter    | eva.porter@example.com         | 011.44.1343.829268
 Millie     | Hunter    | millie.hunter@example.com      | 011.44.1343.729268
 Sofia      | Hicks     | sofia.hicks@example.com        | 011.44.1343.629268
 Lucy       | Crawford  | lucy.crawford@example.com      | 011.44.1343.529268
 Elsie      | Henry     | elsie.henry@example.com        | 011.44.1343.329268
 Imogen     | Boyd      | imogen.boyd@example.com        | 011.44.1644.429267
 Layla      | Mason     | layla.mason@example.com        | 011.44.1644.429266
 Rosie      | Morales   | rosie.morales@example.com      | 011.44.1644.429265
 Maya       | Kennedy   | maya.kennedy@example.com       | 011.44.1644.429264
 Esme       | Warren    | esme.warren@example.com        | 011.44.1644.429263
 Grace      | Ellis     | grace.ellis@example.com        | 011.44.1344.987668
 Lily       | Fisher    | lily.fisher@example.com        | 011.44.1344.498718
 Sophia     | Reynolds  | sophia.reynolds@example.com    | 011.44.1344.478968
 Sophie     | Owens     | sophie.owens@example.com       | 011.44.1344.345268
 Poppy      | Jordan    | poppy.jordan@example.com       | 011.44.1344.129268
 Phoebe     | Murray    | phoebe.murray@example.com      | 011.44.1346.129268
(30 rows)
```

（4）查询视图 salesman 和 salesman_contacts。

```
sale=> CREATE VIEW salesman AS
sale-> SELECT * FROM employees WHERE job_title='Sales Representative';
CREATE VIEW
sale=> SELECT*FROM salesman;
 employee_id | first_name | last_name |             email              |       pho
ne        |      hire_date      | manager_id |      job_title
-------------+------------+-----------+--------------------------------+----------
----------+---------------------+------------+---------------------
          56 | Evie       | Harrison  | evie.harrison@example.com      | 011.44.13
44.486508 | 2016-11-23 00:00:00 |         46 | Sales Representative
          57 | Scarlett   | Gibson    | scarlett.gibson@example.com    | 011.44.13
45.429268 | 2016-01-30 00:00:00 |         47 | Sales Representative
          58 | Ruby       | Mcdonald  | ruby.mcdonald@example.com      | 011.44.13
45.929268 | 2016-03-04 00:00:00 |         47 | Sales Representative
          59 | Chloe      | Cruz      | chloe.cruz@example.com         | 011.44.13
45.829268 | 2016-08-01 00:00:00 |         47 | Sales Representative
          60 | Isabelle   | Marshall  | isabelle.marshall@example.com | 011.44.13
45.729268 | 2016-03-10 00:00:00 |         47 | Sales Representative
          61 | Daisy      | Ortiz     | daisy.ortiz@example.com        | 011.44.13
45.629268 | 2016-12-15 00:00:00 |         47 | Sales Representative
```

```
sale=> CREATE VIEW salesman_contacts AS
sale-> SELECT first_name, last_name, email, phone FROM salesman;
CREATE VIEW
sale=> SELECT*FROM salesman_contacts;
 first_name | last_name |             email              |       phone
------------+-----------+--------------------------------+--------------------
 Evie       | Harrison  | evie.harrison@example.com      | 011.44.1344.486508
 Scarlett   | Gibson    | scarlett.gibson@example.com    | 011.44.1345.429268
 Ruby       | Mcdonald  | ruby.mcdonald@example.com      | 011.44.1345.929268
 Chloe      | Cruz      | chloe.cruz@example.com         | 011.44.1345.829268
 Isabelle   | Marshall  | isabelle.marshall@example.com | 011.44.1345.729268
 Daisy      | Ortiz     | daisy.ortiz@example.com        | 011.44.1345.629268
 Freya      | Gomez     | freya.gomez@example.com        | 011.44.1345.529268
 Elizabeth  | Dixon     | elizabeth.dixon@example.com    | 011.44.1644.429262
 Florence   | Freeman   | florence.freeman@example.com   | 011.44.1346.229268
 Alice      | Wells     | alice.wells@example.com        | 011.44.1346.329268
 Charlotte  | Webb      | charlotte.webb@example.com     | 011.44.1346.529268
 Sienna     | Simpson   | sienna.simpson@example.com     | 011.44.1346.629268
 Matilda    | Stevens   | matilda.stevens@example.com    | 011.44.1346.729268
 Evelyn     | Tucker    | evelyn.tucker@example.com      | 011.44.1343.929268
 Eva        | Porter    | eva.porter@example.com         | 011.44.1343.829268
 Millie     | Hunter    | millie.hunter@example.com      | 011.44.1343.729268
 Sofia      | Hicks     | sofia.hicks@example.com        | 011.44.1343.629268
 Lucy       | Crawford  | lucy.crawford@example.com      | 011.44.1343.529268
 Elsie      | Henry     | elsie.henry@example.com        | 011.44.1343.329268
 Imogen     | Boyd      | imogen.boyd@example.com        | 011.44.1644.429267
 Layla      | Mason     | layla.mason@example.com        | 011.44.1644.429266
 Rosie      | Morales   | rosie.morales@example.com      | 011.44.1644.429265
 Maya       | Kennedy   | maya.kennedy@example.com       | 011.44.1644.429264
 Esme       | Warren    | esme.warren@example.com        | 011.44.1644.429263
 Grace      | Ellis     | grace.ellis@example.com        | 011.44.1344.987668
 Lily       | Fisher    | lily.fisher@example.com        | 011.44.1344.498718
 Sophia     | Reynolds  | sophia.reynolds@example.com    | 011.44.1344.478968
 Sophie     | Owens     | sophie.owens@example.com       | 011.44.1344.345268
 Poppy      | Jordan    | poppy.jordan@example.com       | 011.44.1344.129268
 Phoebe     | Murray    | phoebe.murray@example.com      | 011.44.1346.129268
(30 rows)
```

（5）在当前窗口输入命令：\c - omm 切换到 omm 用户。

```
sale=> \c - omm
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "sale" as user "omm".
sale=#
```

（6）创建新用户 user1。

```
sale=# CREATE USER user1 PASSWORD 'user1@123';
CREATE ROLE
```

（7）在当前窗口输入命令：\c – user1 切换到 user1 用户。

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

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

```
sale=> select * from salesman_contacts;
ERROR:  relation "salesman_contacts" does not exist on dn_6001
LINE 1: select * from salesman_contacts;
                      ^
```

（9）发布命令：\c – yuxiaoping 切换到 whj 用户—此处的 whj 应替换为你们自己创建的用户。密码是 yuxiaoping@123

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

（10）在当前 whj 用户下输入命令：grant select on salesman_contacts to user1；实现授权操作。由于 opengauss 报错，实际上使用了 grant all on schema icebear to user1;。

```
sale=> SET SEARCH_PATH To icebear,public;
SET
sale=> 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。

```
sale=> select * from salesman_contacts;
 first_name | last_name |              email              |        phone
------------+-----------+---------------------------------+--------------------
 Evie       | Harrison  | evie.harrison@example.com       | 011.44.1344.486508
 Scarlett   | Gibson    | scarlett.gibson@example.com     | 011.44.1345.429268
 Ruby       | Mcdonald  | ruby.mcdonald@example.com       | 011.44.1345.929268
 Chloe      | Cruz      | chloe.cruz@example.com          | 011.44.1345.829268
 Isabelle   | Marshall  | isabelle.marshall@example.com   | 011.44.1345.729268
 Daisy      | Ortiz     | daisy.ortiz@example.com         | 011.44.1345.629268
 Freya      | Gomez     | freya.gomez@example.com         | 011.44.1345.529268
 Elizabeth  | Dixon     | elizabeth.dixon@example.com     | 011.44.1644.429262
 Florence   | Freeman   | florence.freeman@example.com    | 011.44.1346.229268
 Alice      | Wells     | alice.wells@example.com         | 011.44.1346.329268
 Charlotte  | Webb      | charlotte.webb@example.com      | 011.44.1346.529268
 Sienna     | Simpson   | sienna.simpson@example.com      | 011.44.1346.629268
 Matilda    | Stevens   | matilda.stevens@example.com     | 011.44.1346.729268
 Evelyn     | Tucker    | evelyn.tucker@example.com       | 011.44.1343.929268
 Eva        | Porter    | eva.porter@example.com          | 011.44.1343.829268
 Millie     | Hunter    | millie.hunter@example.com       | 011.44.1343.729268
 Sofia      | Hicks     | sofia.hicks@example.com         | 011.44.1343.629268
 Lucy       | Crawford  | lucy.crawford@example.com       | 011.44.1343.529268
 Elsie      | Henry     | elsie.henry@example.com         | 011.44.1343.329268
 Imogen     | Boyd      | imogen.boyd@example.com         | 011.44.1644.429267
 Layla      | Mason     | layla.mason@example.com         | 011.44.1644.429266
 Rosie      | Morales   | rosie.morales@example.com       | 011.44.1644.429265
 Maya       | Kennedy   | maya.kennedy@example.com        | 011.44.1644.429264
 Esme       | Warren    | esme.warren@example.com         | 011.44.1644.429263
 Grace      | Ellis     | grace.ellis@example.com         | 011.44.1344.987668
 Lily       | Fisher    | lily.fisher@example.com         | 011.44.1344.498718
 Sophia     | Reynolds  | sophia.reynolds@example.com     | 011.44.1344.478968
 Sophie     | Owens     | sophie.owens@example.com        | 011.44.1344.345268
 Poppy      | Jordan    | poppy.jordan@example.com        | 011.44.1344.129268
 Phoebe     | Murray    | phoebe.murray@example.com       | 011.44.1346.129268
(30 rows)
```

/*说明：步骤（2）-（11）的主要目的是用于验证视图的作用：被授权用户只能查询在权限范围内的数据，范围外的数据不可访问*/

（12)查看与角色、权限相关的系统表和系统视图:pg_roles,pg_authid。pg_roles 视图提供访问数据库角色有关信息的接口。这个视图只是 pg_authid 表的公开可读部分的视图化，同时把口令字段用空白填充。

```
sale=> SELECT *FROM pg_roles;
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rolpass
word | rolvalidbegin | rolvaliduntil |  rolrespool  | rolparentid | roltabspace | rolconfig |  oid  | roluseft | rolkind | nodegroup | roltempspace | rolspillspace |
rolmonitoradmin | roloperatoradmin | rolpolicyadmin
---------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+--------
-----+---------------+---------------+--------------+-------------+-------------+-----------+-------+----------+---------+-----------+--------------+---------------+-
----------------+------------------+----------------
 user1   | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | *******
*    |               |               | default_pool |           0 |             |           | 16788 | f        | n       |           |              |               |
f               | f                | f
(1 row)
```

```
postgres=# SELECT * FROM pg_authid;
  rolname   | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit |
                                                                    rolpassword
                                 | rolvalidbegin | rolvaliduntil |  rolrespool  | roluseft | rolparentid | roltabspace | rolkind | rolnodegroup | roltempspace
| rolspillspace | rolexcpdata | rolmonitoradmin | roloperatoradmin | rolpolicyadmin
------------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+-----
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------+---------------+---------------+--------------+----------+-------------+-------------+---------+--------------+--------------
+---------------+-------------+-----------------+------------------+----------------
 omm        | t        | t          | t             | t           | t            | t           | t              | t             | t              |           -1 | sha2
562fb9095861d39b85f1e9baf05f792f9193b21db69a786d2b230d8819d10cdf6bf1af723b75c8a8082915c0c308014130c6663f8519162be5e69acfd4fbd046a71fd2bf82224b1a360da14ceb3c17863a27dd
bc9d4cceb537a8e689cac4250a56ecdfecefade |               |               | default_pool | t        |           0 |             | n       |              |            0 |
|               |             | t               | t                | t
 yuxiaoping | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | sha2
56580d42e87ee71f757e1e0ca70d08616e05a1e85f43f639c61aec1f007aa334a6eb9f3bbc18be1252ea6177d8d87a01bac35e1d7628bf10729a874a7f9447db849f08049666353bfbda0e43dca415d7ef61c8
d39877d378691b9a22371383873becdfecefade |               |               | default_pool | f        |           0 |             | n       |              |              |
|               |             | f               | f                | f
 user1      | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | sha2
56732a4d0ca93ad4f34261b86bc6163ba009f7385371b19008938a541bdd40bb0077ea98d9483877f22ab35e17ea947be30a60abb87f1b00ecb6ad1e10739f30691103d5094e91127589282dda72babebb233b
f2f0c36fd821ea92144b89ea2082ecdfecefade |               |               | default_pool | f        |           0 |             | n       |              |              |
|               |             | f               | f                | f
(3 rows)
```

（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 创建表格

```
postgres=# GRANT ALL PRIVILEGES TO yuxiaoping;
ALTER ROLE
postgres=# \q
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U yuxiaoping -W yuxiaoping@123  -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sale=> CREATE TABLE Student
sale-> (Sno CHAR(9)PRIMARY KEY,
sale(> Sname CHAR(20)UNIQUE,
sale(> Ssex CHAR(2),
sale(> Sage SMALLINT,
sale(> Sdept CHAR(20));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "student_pkey" for table "student"
NOTICE:  CREATE TABLE / UNIQUE will create implicit index "student_sname_key" for table "student"
CREATE TABLE
sale=> CREATE TABLE Course
sale-> (Cno CHAR(4) PRIMARY KEY,
sale(> Cname CHAR(40) NOT NULL,
sale(> Cpno CHAR(4),
sale(> Ccredit SMALLINT,
sale(> FOREIGN KEY(Cpno)REFERENCES Course(Cno));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "course_pkey" for table "course"
CREATE TABLE
sale=> CREATE TABLE SC
sale-> (Sno CHAR(9),
sale(> Cno CHAR(4),
sale(> Grade SMALLINT,
sale(> PRIMARY KEY(Sno,Cno),
sale(> FOREIGN KEY(Sno)REFERENCES Student(Sno),
sale(> FOREIGN KEY(Cno)REFERENCES Course(Cno));
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "sc_pkey" for table "sc"
CREATE TABLE
sale=>
```

② 创建角色，授权

```
sale=> GRANT SELECT ON TABLE Student TO user1;
GRANT
sale=> GRANT ALL PRIVILEGES ON TABLE Student,Course TO user1;
GRANT
sale=> GRANT SELECT ON TABLE SC TO PUBLIC;
GRANT
sale=> GRANT UPDATE(Sno),SELECT ON TABLE Student TO PUBLIC;
GRANT
sale=> GRANT UPDATE(Sno),SELECT ON TABLE Student TO user1;
GRANT
```

```
sale=> GRANT INSERT ON TABLE SC TO user1 WITH GRANT OPTION;
GRANT
sale=> Create USER user2 PASSWORD 'user2@123';
CREATE ROLE
sale=> Create USER user3 PASSWORD 'user3@123';
CREATE ROLE
sale=> Create USER user4 PASSWORD 'user4@123';
CREATE ROLE
sale=> \q
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U user1 -W user1@123  -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sale=> GRANT INSERT ON TABLE SC TO user2 WITH GRANT OPTION;
GRANT
sale=> \q
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U user2 -W user2@123  -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sale=> GRANT INSERT ON TABLE SC TO user3;
GRANT
```

③ 回收限权

```
sale=> REVOKE UPDATE(Sno) ON TABLE Student FROM user1;
WARNING:  no privileges could be revoked for column "sno" of relation "student"
REVOKE
```

```
sale=> REVOKE SELECT ON TABLE SC FROM PUBlic;
WARNING:  no privileges could be revoked for "sc"
WARNING:  no privileges could be revoked for column "xc_node_id" of relation "sc"
WARNING:  no privileges could be revoked for column "tableoid" of relation "sc"
WARNING:  no privileges could be revoked for column "cmax" of relation "sc"
WARNING:  no privileges could be revoked for column "xmax" of relation "sc"
WARNING:  no privileges could be revoked for column "cmin" of relation "sc"
WARNING:  no privileges could be revoked for column "xmin" of relation "sc"
WARNING:  no privileges could be revoked for column "ctid" of relation "sc"
WARNING:  no privileges could be revoked for column "sno" of relation "sc"
WARNING:  no privileges could be revoked for column "cno" of relation "sc"
WARNING:  no privileges could be revoked for column "grade" of relation "sc"
REVOKE
sale=> REVOKE INSERT ON TABLE SC FROM user1 CASCADE;
REVOKE
sale=>
```

3. 实验总结

3.1 实验思考

·具有什么权限才能创建新用户?

答: 用户必须拥有数据库创建的权限或者是数据库的系统管理员权限才能创建数据库。创建用户必须有 dba 的权限。

·角色的作用是什么?

答：数据库角色是被命名的一组与数据库操作相关的权限，角色是权限的集合，因此，可以为一组有相同权限的用户创建一个角色，使用角色来管理数据库限权可以简化授权的过程。

·如何实现角色所含权限的修改，请设计样例验证之

答：给角色赋予对象权限，使用 GRANT。。要撤销或重新授予角色对权限， 可通过 REVOKE 实现。如下面的代码：

```
sale=> GRANT INSERT ON TABLE SC TO user2 WITH GRANT OPTION;
GRANT
sale=> \q
[omm@ecs-ad18 ~]$ gsql -d sale -p 26000 -U user2 -W user2@123  -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

sale=> GRANT INSERT ON TABLE SC TO user3;
GRANT
```

```
sale=> REVOKE INSERT ON TABLE SC FROM user1 CASCADE;
REVOKE
sale=>
```

3.2 对实验的认识

通过实验我对 openGauss 中的一些语句更熟悉了。如

SET SEARCH_PATH TO icebear, public;可以将搜索路径设置为 icebear、public，首先搜索 icebear。如 SELECT * FROM customer_t1;可以用来查询表 customer_t1 的所有数据。gsql -d sale -p 26000 -U yuxiaoping -W yuxiaoping@123 -r 或者 gsql -d sale -p 26000 -U user1 -W user1@123 -r 可以用来将新用户连接到数据库。可以使用 gsql -d postgres -p 26000 -r 连接到 postgres。gs_om -t start 可以开启数据库。

3.3 遇到的困难及解决方法

要更改当前会话的默认 Schema，请使用 SET 命令。执行如下命令

SET SEARCH_PATH To icebear,public;将搜索路径设置为 myschema、public，首先搜索 myschema。

```
sale=> SET SEARCH_PATH TO icebear, public;
SET
```

高斯默认有 session 超时时间，若想要 session 一直保持，需要修改配置项：ALTER DATABASE sale SET session_timeout TO 0;

```
postgres=# ALTER DATABASE postgres SET session_timeout TO 0;
ALTER DATABASE
```