数据库系统课程实验报告

实验名称:	数据库的安全性
实验日期:	2022/4/21
实验地点:	四号楼
提交日期:	2022/4/24

学号:	22920202202879
姓名:	陈奕培
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）完成网页内容

1.切换到 omm

```
[root@ecs-4904 ~]# su - omm
Last login: Tue Apr 19 20:31:32 CST 2022 on pts/0
```

2.启动服务

```
[omm@ecs-4904 ~]$ gs_om -t start
```

3.连接到数据库

```
[omm@ecs-4904 ~]$ gsql -d postgres -p 26000
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:0
ommit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring hi
curity)
Type "help" for help.
```

4.用户

4.1 新建用户

```
postgres=# CREATE USER jim PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=#
```

4.2 查看用户列表

```
postgres=# SELECT * FROM pg_user;
 usename | usesysid | usecreatedb | usesuper | usecatupd | userepl |  passwd  | valbegin | valuntil |   respool    | parent | spacelimit | useconfig | nodegroup | tempspacelimit | spills
pacelimit | usemonitoradmin | useoperatoradmin | usepolicyadmin
---------+----------+-------------+----------+-----------+---------+----------+----------+----------+--------------+--------+------------+-----------+-----------+----------------+-------
---------+-----------------+------------------+---------------
 omm     |       10 | t           | t        | t         | t       | ******** |          |          | default_pool |      0 |            |           |           |                |
         | t               | t                |          | t
 xiaopei |    16384 | f           | f        | f         | f       | ******** |          |          | default_pool |      0 |            |           |           |                |
         | f               | f                |          | f
 jim     |    16831 | f           | f        | f         | f       | ******** |          |          | default_pool |      0 |            |           |           |                |
         | f               | f                |          | f
(3 rows)
```

4.3 为 jim 追加创建角色权限

```
postgres=# ALTER USER jim CREATEROLE;
ALTER ROLE
```

4.4 删除用户

```
postgres=# DROP USER jim CASCADE;
DROP ROLE
```

5.角色

5.1 创建角色

```
postgres=# CREATE ROLE manager IDENTIFIED BY 'Bigdata@123';
CREATE ROLE
```

5.2 查看角色

```
postgres=# SELECT * FROM pg_roles;
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rolpassword | rolvalidbegin
 | rolvaliduntil |  rolrespool  | rolparentid | roltabspace | rolconfig |  oid  | roluseft | rolkind | nodegroup | roltempspace | rolspillspace | rolmonitoradmin | roloperatoradmin | rol
policyadmin
--------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+-------------+---------------
-+---------------+--------------+-------------+-------------+-----------+-------+----------+---------+-----------+--------------+---------------+-----------------+------------------+----
----------
 omm     | t        | t          | t             | t           | t            | t           | t              | t             | t              |           -1 | ********    |
         |          | default_pool |           0 |             |              |    10 | t        | n       |           |              |               | t               | t                | t
 xiaopei | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | ********    |
         |          | default_pool |           0 |             |              | 16384 | f        | n       |           |              |               | f               | f                | f
 manager | f        | t          | f             | f           | f            | f           | f              | f             | f              |           -1 | ********    |
         |          | default_pool |           0 |             |              | 16835 | f        | n       |           |              |               | f               | f                | f
(3 rows)
```

5.3 修改角色密码

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

5.4 修改角色 manager 为系统管理员

```
postgres=# ALTER ROLE manager SYSADMIN;
ALTER ROLE
```

5.5 删除角色

```
ALTER ROLE
postgres=# DROP ROLE manager;
DROP ROLE
postgres=#
```

6.权限设置

6.1 创建用户 joe

```
postgres=# CREATE USER joe PASSWORD 'Bigdata@123';
CREATE ROLE
postgres=#
```

6.2 将 sysadmin 权限授权给 joe

```
postgres=# GRANT ALL PRIVILEGES TO joe;
ALTER ROLE
postgres=#
```

6.3 撤销 joe 的 sysadmin 权限

```
postgres=# REVOKE ALL PRIVILEGES FROM joe;
ALTER ROLE
postgres=#
```

7.将数据库对象授权给角色或用户

7.1 创建 tpcds 模式

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

7.2 创建表

```
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) );
CREATE TABLE
```

7.3 将模式的使用权限和表的所有权限授予 joe

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

7.4 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

```
postgres=#
postgres=# GRANT select (r_reason_sk,r_reason_id,r_reason_desc),update (r_reason_desc) ON
tpcds.reason TO joe;
GRANT
postgres=#
```

7.5 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 create,connect on database postgres TO joe WITH GRANT OPTION;
GRANT
```

7.6 创建角色 tpcds_manager

```
postgres=# CREATE ROLE tpcds_manager PASSWORD 'Bigdata@123';
CREATE ROLE
```

7.7 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
```

7.8 查看表 reason 的权限

```
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       | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | REFERENCES     | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | TRIGGER        | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | ALTER          | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | DROP           | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | COMMENT        | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | INDEX          | NO           | NO
 omm     | joe     | postgres      | tpcds        | reason     | VACUUM         | NO           | NO
(19 rows)
```

8.将用户或者角色的权限授权给其他用户或者角色

8.1 创建角色 manager

```
postgres=# CREATE ROLE manager PASSWORD 'Bigdata@123';
CREATE ROLE
```

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

```
postgres=# GRANT joe TO manager WITH ADMIN OPTION;
GRANT ROLE
```

8.3 创建用户 senior_manager

```
postgres=# CREATE ROLE senior_manager PASSWORD 'Bigdata@123';
CREATE ROLE
```

8.4 将用户 manager 的权限授权给该用户

```
postgres=# GRANT manager TO senior_manager;
GRANT ROLE
```

9.权限回收

9.1 逐步回收 manager 的权限

```
postgres=# REVOKE joe FROM manager;
REVOKE ROLE
postgres=# REVOKE manager FROM senior_manager;
REVOKE ROLE
```

9.2 删除 manager 用户

```
REVOKE ROLE
postgres=# DROP USER manager;
DROP ROLE
```

9.3 逐步回收 joe 权限

```
DROP ROLE
postgres=# REVOKE ALL PRIVILEGES ON tpcds.reason FROM joe;
REVOKE
postgres=# REVOKE ALL PRIVILEGES ON SCHEMA tpcds FROM joe;
REVOKE
```

9.4 逐步回收 tpcds_manager 权限

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

9.5 删除 tpcds_manager 用户

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

9.6 删除 senior_manager 用户

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

9.7 删除 joe 用户

```
DROP ROLE
postgres=# DROP USER joe CASCADE;
DROP ROLE
postgres=#
```

（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。

1.切换到 sales 数据库

```
postgres=# ^Z
[2]+  Stopped                 gsql -d postgres -p 26000
[omm@ecs-4904 ~]$ gsql -d sales -p 26000
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:
Non-SSL connection (SSL connection is recommended when requiring
Type "help" for help.

sales=#
```

2.创建视图

```
sales=# CREATE VIEW salesman 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。

```
sales=# CREATE VIEW  salesman_contacts(first_name,last_name,email,phone) AS
sales-# SELECT first_name,last_name,email,phone
sales-# FROM salesman;
CREATE VIEW
sales=#
```

（4）查询视图 salesman 和 salesman_contacts

```
sales=# SELECT * FROM salesman LIMIT 10;
 employee_id | first_name | last_name |           email            |       phone
 |      hire_date      | manager_id |      job_title
-------------+------------+-----------+----------------------------+-------------------
-+---------------------+------------+----------------------
          55 | Evie       | Harrison  | evie.harrison@example.com     | 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.com     | 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
          60 | Isabelle   | Marshall  | isabelle.marshall@example.com | 011.44.1345.729268
 | 2016-03-10 00:00:00 |         47 | Sales Representative
          61 | Daisy      | Ortiz     | daisy.ortiz@example.com       | 011.44.1345.629268
 | 2016-12-15 00:00:00 |         47 | Sales Representative
          62 | Freya      | Gomez     | freya.gomez@example.com       | 011.44.1345.529268
 | 2016-11-03 00:00:00 |         47 | Sales Representative
          80 | Elizabeth  | Dixon     | elizabeth.dixon@example.com   | 011.44.1644.429262
 | 2016-01-04 00:00:00 |         50 | Sales Representative
          64 | Florence   | Freeman   | florence.freeman@example.com  | 011.44.1346.229268
 | 2016-03-19 00:00:00 |         48 | Sales Representative
          65 | Alice      | Wells     | alice.wells@example.com       | 011.44.1346.329268
 | 2016-01-24 00:00:00 |         48 | Sales Representative
(10 rows)
```

```
sales=# SELECT * FROM salesman_contacts LIMIT 10;
 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
(10 rows)
```

（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 requiring high-security)
You are now connected to database "sales" as user "omm".
```

（6）创建新用户 user1

```
ERROR:  password must contain at least three kinds of
sales=# CREATE USER user1 PASSWORD '123@@abc';
CREATE ROLE
sales=#
```

（7）在当前窗口输入命令：\c – user1 切换到 user1 用户

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

（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
sales=>
```

显示权限不够

（9）切换到 yipei

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

（10）授权

```
sales=> GRANT SELECT ON salesman_contacts TO user1;
GRANT
sales=>
```

（11）重复 7，8

```
sales=> \c - user1
Password for user user1:
Non-SSL connection (SSL connection is recommended when requiring 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      | 011.44.1344.486508
 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
```

（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 * FROM PG_ROLES
sales-> ;
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rol
 | rolvaliduntil |  rolrespool  | rolparentid | roltabspace | rolconfig |  oid  | rolusesft | rolkind | nodegroup | roltempspace | rolspillspace | rolmonitoradmin |
policyadmin
---------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+----
-+---------------+--------------+-------------+-------------+-----------+-------+-----------+---------+-----------+--------------+---------------+-----------------+-
----------
 user1   | f        | t          | f             | f           | f            | t           | f              | f             | f              |              -1 | ***
 |               | default_pool |           0 |             |           | 16865 | f         | n       |           |              |               | f
(1 row)
```

```
sales=# SELECT * FROM pg_authid;
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit |
                                                                rolpassword                                                                                 | r
olvalidbegin | rolvaliduntil |  rolrespool  | roluseft | rolparentid | roltabspace | rolkind | rolnodegroup | roltempspace | rolspillspace | rolexcpdata | rolmonitoradmin | roloperatorad
min | rolpolicyadmin
---------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------+----------------+--------------+----
------------------------------------------------------------------------------------------------------------------------------------------------------------+---
-------------+---------------+--------------+----------+-------------+-------------+---------+--------------+--------------+---------------+-------------+-----------------+---
----+----------------
 omm     | t        | t          | t             | t           | t            | t           | t              | t             | t              |           -1 | sha25603b189ea4b1e3ad8c2f09
91303641e9262ed2bb306a3894f97bf89d50ef8e013c8479342042392 8d497c7af0bf0c7358d6bbf3a713b7a57ee000ec707b3f7cccd5486e49677 5d894352f7705 8e9207cef42fd8fccf52f600e504e2855 1d642cfecdfecefade |
             |               | default_pool | t        |           0 |             | n       |            0 |               |                |              | t               | t
    | t
 xiaopei | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | sha256dd47945b794c9ba9f6211
4afe7797d268eddcf75f2e7488115b2189b2241d8760d0f824e42cb5f0d1c58555384ca99e068cf2735a7d595e7507f51255e6f95b489af63266158f723463b44152407678cc4ef380a288a8eb561be7dbbfb598d5ecdfecefade |
             |               | default_pool | f        |           0 |             | n       |                |               |                |              | f               | f
    | f
 user1   | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | sha256b827b336034b604730cf1
1c1ab5fbf754615eaa8d37b6348dca37e22d7ee2d87bc44183f756db2265700bc49d16e62d8cbd1ae2e5b3e8f6fd98416f1ea64429bf3d54e823a0952e089f2eef45a1e99e815a7eec6abfa07cbe5c8ad8d05e2055671ecdfecefade |
             |               | default_pool | f        |           0 |             | n       |                |               |                |              | f               | f
    | f
 yipei   | f        | t          | f             | f           | f            | t           | f              | f             | f              |           -1 | sha2568e46a26ad116ed0537215
abc5e070f22063c631bfb85bce8ecb61240edbdae4043019983e77eab1f0a4c23571694ef55e447cc1f1df8f41faf1621a584018adc54109e3c928c32456d8b389817afa72d49c5a43bee0c6606ebf21344150ebf56ecdfecefade |
             |               | default_pool | f        |           0 |             | n       |                |               |                |              | f               | f
    | f
(4 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.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

```
DROP ROLE
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

```
ERROR:  The password should not be NULL.
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
```

实验思考

- 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