

西安交通大学实验报告

课程名称：数据库基础与应用
学院：物理学院
班级：物试 2201
学号：2223711803

实验名称：数据库安全性
实验日期：2023.4.19
姓名：汪洋

1 实验目的

1. 认识数据库的安全性的意义。
2. 了解安全性的概念和技术。
3. 掌握用户管理、权限管理的设置方法。
4. 掌握备份和恢复单个数据库的方法。

2 实验任务

1. 用户和权限管理。
2. 备份和恢复数据库。

3 实验环境

1. 华为云服务器
2. openGauss 数据库
3. Windows 操作系统
4. Putty

4 实验步骤与结果

4.1 用户和权限管理

1. 创建名称为 bank 的数据库。

```
1 [omm@ecs-5138 ~]$ gs_om -t start
2 [omm@ecs-5138 ~]$ gsql -d postgres -U omm -p 26000 -r
3 postgres=# create database bank;
```



2. 创建数据库系统管理员 admin1 和 admin2。

```
1 create user admin1 with sysadmin password 'A13561973155a';
2 create user admin2 with sysadmin password 'A13561973155a';
```

```
omm@ecs-5138:~  
postgres=# create user admin1 with sysadmin password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=# create user admin2 with sysadmin password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=#
```

3. 创建数据库 bank 的数据库管理员 bankdbmanager，将 bank 数据库的所有者权限赋予 bankdbmanager。

```
1 create user bankdbmanager with sysadmin password 'A13561973155a';  
2 grant all privileges on database bank to bankdbmanager;
```

```
omm@ecs-5138:~  
postgres=# create user bankdbmanager with sysadmin password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=# grant all privileges on database bank to bankdbmanager;  
GRANT  
postgres=#
```

4. 为业务员 wang 创建账户。

```
1 create user wang with password 'A13561973155a';
```

```
omm@ecs-5138:~  
postgres=# create user wang with password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=#
```

5. 为部门管理人员创建角色 deptmanager。

```
1 create role deptmanager with password 'A13561973155a';
```

```
omm@ecs-5138:~  
postgres=# create role deptmanager with password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=#
```

6. 为部门经理 sun 创建账户。

```
1 create user sun with password 'A13561973155a';
```

```
omm@ecs-5138:~  
postgres=# create user sun with password 'A13561973155a';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=#
```

7. 新入职职员 zhang，为其创建 6 个月有效期的账户。

```
1 create user zhang with password 'A13561973155a' valid begin '2023-4-19' valid until '2023-10-19';
```

```
omm@ecs-5138:~  
postgres=# create user zhang with password 'A13561973155a' valid begin '2023-4-19' valid until '2023-10-19';  
NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.  
CREATE ROLE  
postgres=#
```

8. 以数据库 bank 的数据库管理员 bankdbmanager 身份连接数据库 bank。

```
1 postgres=# \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d bank -U bankdbmanager -W A13561973155a -r
```

```
omm@ecs-5138:~  
postgres=# \q  
[omm@ecs-5138 ~]$ gsql -p 26000 -d bank -U bankdbmanager -W A13561973155a -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.  
bank=>
```

9. 创建同名模式（如果需要）并设置默认搜索路径为 bankdbmanager。

```
1 create schema bankdbmanager;  
2 set search_path to bankdbmanager;
```

```
omm@ecs-5138:~  
bank=> create schema bankdbmanager;  
CREATE SCHEMA  
bank=> set search_path to bankdbmanager;  
SET  
bank=>
```

10. 在数据库 bank 中创建 worker 数据表，其关系模式为 worker(id,name,age,dept)，字段的数据类型和长度根据常用参数设置。

```
1 create table worker(id varchar(10), name varchar(20), age int, dept varchar(20));
```

```
omm@ecs-5138:~  
bank=> create table worker(id varchar(10), name varchar(20), age int, dept varchar(20));  
CREATE TABLE  
bank=>
```

11. 在数据表中插入一条数据并查询，name 使用自己的名字。

```
1 insert into worker values('2223711803', '汪洋', 18, '行政');
```

```
omm@ecs-5138:~  
bank=> insert into worker values('2223711803', '汪洋', 18, '行政');  
INSERT 0 1  
bank=>
```

12. 将 worker 表的查询权限授予角色 deptmanager。

```
1 grant select on worker to deptmanager;
```

```
omm@ecs-5138:~  
bank=> grant select on worker to deptmanager;  
GRANT  
bank=>
```

13. 将角色 deptmanager 的权限授予用户 sun。

```
1 grant deptmanager to sun;
```

```
omm@ecs-5138:~  
bank=> grant deptmanager to sun;  
GRANT ROLE  
bank=>
```

14. 将 worker 表的所有权限授予用户 wang。

```
1 grant all privileges on worker to wang;
```

```
omm@ecs-5138:~  
bank=> grant all privileges on worker to wang;  
GRANT  
bank=>
```

15. 将模式 bankdbmanager 的使权限分别授予用户 wang 和角色 deptmanager。

```
1 grant usage on schema bankdbmanager to wang, deptmanager;
```

```
omm@ecs-5138:~  
bank=> grant all privileges on schema bankdbmanager to wang, deptmanager;  
GRANT  
bank=>
```

16. 以 omm 连接数据库:

```
1 bank=> \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d postgres -U omm -r
```

```
omm@ecs-5138:~  
postgres=# select username ,usesysid from pg_user;  
-----+-----  
username | usesysid  
-----+-----  
omm      |      10  
root     |     16702  
admin1   |     16719  
admin2   |     16723  
bankdbmanager |     16735  
wang     |     16739  
sun      |     16746  
zhang    |     16754  
(8 rows)  
postgres=#
```

查看用户名和用户 ID (pg_user 表)。

```
1 select username, usesysid from pg_user;
```

```
omm@ecs-5138:~  
postgres=# select rolname, rolcreatedb, rolsystemadmin from pg_authid;  
-----+-----+-----  
rolname  | rolcreatedb | rolsystemadmin  
-----+-----+-----  
omm      | t           | t  
root     | f           | f  
admin1   | f           | t  
admin2   | f           | t  
bankdbmanager | f           | t  
wang     | f           | f  
deptmanager | f           | f  
sun      | f           | f  
zhang    | f           | f  
(9 rows)  
postgres=#
```

查看用户名, 是否拥有数据库创建权限, 是否系统管理员 (pg_authid 表)。

```
1 select rolname, rolcreatedb, rolsystemadmin from pg_authid;
```

```
omm@ecs-5138:~  
postgres=# select nspname, a.username from pg_user a right outer join pg_namespace  
e b on a.usesysid=b.nspowner;  
-----+-----  
nspname  | username  
-----+-----  
pg_toast | omm  
cstore   | omm  
pkg_service | omm  
dbe_perf | omm  
snapshot | omm  
pg_catalog | omm  
public   | omm  
information_schema | omm  
admin1   | admin1  
admin2   | admin2  
bankdbmanager | bankdbmanager  
wang     | wang  
sun      | sun  
zhang    | zhang  
(14 rows)  
postgres=#
```

查看模式名, 拥有者名 (pg_user 和 pg_namespace 表)。

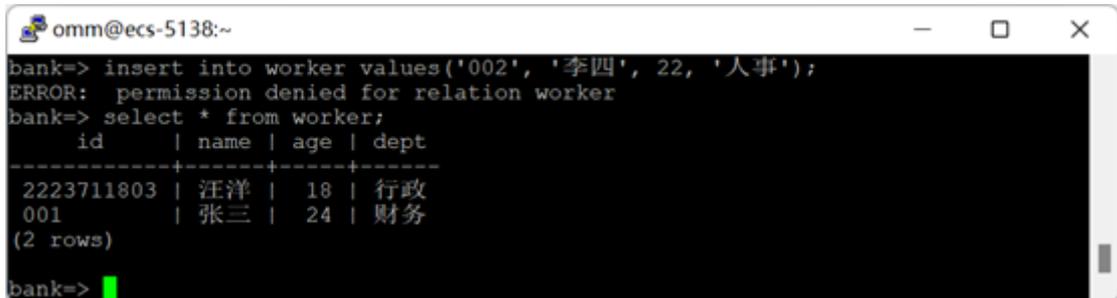
```
1 select nspname, a.username from pg_user a right outer join pg_namespace b on a.usesysid=b.  
   nspowner;
```



```
omm@ecs-5138:~  
bank=> select * from worker;  
   id   | name | age | dept  
-----+-----+-----+-----  
2223711803 | 汪洋 | 18 | 行政  
001    | 张三 | 24 | 财务  
(2 rows)  
bank=>
```

17. 以用户 wang 连接数据库 bank，在 worker 表中插入数据，查询信息。

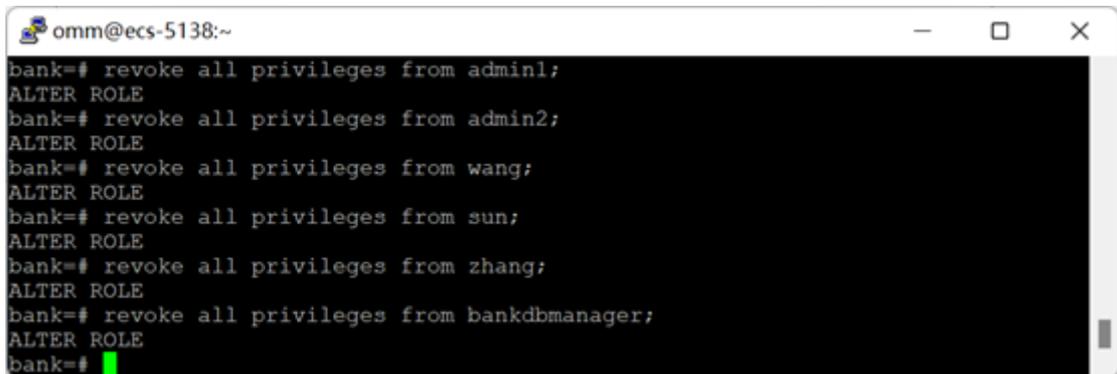
```
1 postgres=# \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d bank -U wang -W A13561973155a -r  
3 set search_path to bankdbmanager;  
4 insert into worker values('001', '张三', 24, '财务');  
5 select * from worker;
```



```
omm@ecs-5138:~  
bank=> insert into worker values('002', '李四', 22, '人事');  
ERROR: permission denied for relation worker  
bank=> select * from worker;  
   id   | name | age | dept  
-----+-----+-----+-----  
2223711803 | 汪洋 | 18 | 行政  
001    | 张三 | 24 | 财务  
(2 rows)  
bank=>
```

18. 以用户 sun 连接数据库 bank，在 worker 表中插入数据，查询数据，观察结果。

```
1 bank=# \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d bank -U sun -W A13561973155a -r  
3 insert into worker values('002', '李四', 22, '人事');  
4 select * from worker;
```



```
omm@ecs-5138:~  
bank=# revoke all privileges from admin1;  
ALTER ROLE  
bank=# revoke all privileges from admin2;  
ALTER ROLE  
bank=# revoke all privileges from wang;  
ALTER ROLE  
bank=# revoke all privileges from sun;  
ALTER ROLE  
bank=# revoke all privileges from zhang;  
ALTER ROLE  
bank=# revoke all privileges from bankdbmanager;  
ALTER ROLE  
bank=#
```

19. 以 bankdbmanager 连接系统，回收所有权限。

```
1 bank=# \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d bank -U bankdbmanager -W A13561973155a -r  
3 revoke all privileges from admin1;  
4 revoke all privileges from admin2;  
5 revoke all privileges from wang;  
6 revoke all privileges from sun;  
7 revoke all privileges from zhang;  
8 revoke all privileges from bankdbmanager;
```

```
omm@ecs-5138:~  
postgres=# drop database bank;  
DROP DATABASE  
postgres=# drop user admin1;  
DROP ROLE  
postgres=# drop user admin2;  
DROP ROLE  
postgres=# drop user wang;  
DROP ROLE  
postgres=# drop user sun;  
DROP ROLE  
postgres=# drop user zhang;  
DROP ROLE  
postgres=# drop user bankdbmanager;  
DROP ROLE  
postgres=#
```

20. 以 omm 连接系统，删除本实验创建的数据库和用户。

```
1 bank=# \q  
2 [omm@ecs-5138 ~]$ gsql -p 26000 -d postgres -U omm -r  
3 drop database bank;  
4 drop user admin1;  
5 drop user admin2;  
6 drop user wang;  
7 drop user sun;  
8 drop user zhang;  
9 drop user bankdbmanager;
```

```
omm@ecs-5138:~  
postgres=# drop database bank;  
DROP DATABASE  
postgres=# drop user admin1;  
DROP ROLE  
postgres=# drop user admin2;  
DROP ROLE  
postgres=# drop user wang;  
DROP ROLE  
postgres=# drop user sun;  
DROP ROLE  
postgres=# drop user zhang;  
DROP ROLE  
postgres=# drop user bankdbmanager;  
DROP ROLE  
postgres=#
```

4.2 备份数据库

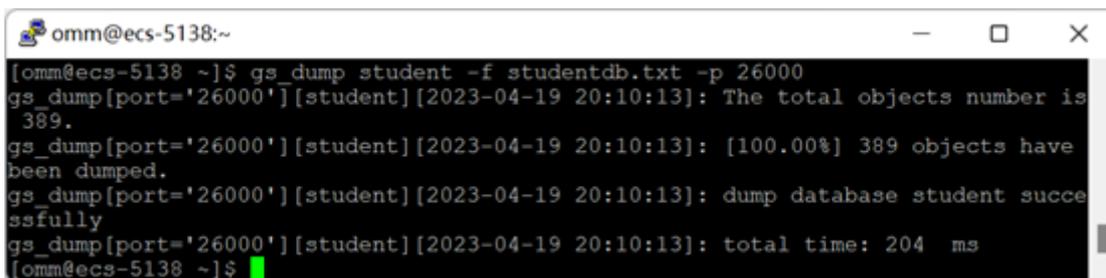
1. 以 omm 创建一个名称为 student 的数据库，切换到 student 数据库，创建模式 student，在该模式下创建数据表 stu(name varchar(20))，在其中插入两行数据（含自己的名字），查询。

```
1 create database student;  
2 \c student  
3 create schema student;  
4 set search_path to student;  
5 create table stu(name varchar(20));  
6 insert into stu values('汪洋');  
7 insert into stu values('张三');  
8 select * from stu;
```

```
omm@ecs-5138:~  
student=# select * from stu;  
 name  
-----  
 汪洋  
 张三  
(2 rows)  
student=#
```

2. 在操作系统下使用下列命令备份 student 数据库。

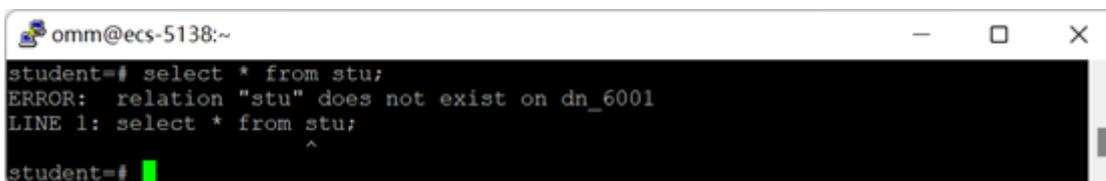
```
1 student=# \q
2 [omm@ecs-5138 ~]$ gs_dump student -f studentdb.txt -p 26000
```



```
omm@ecs-5138:~$ gs_dump student -f studentdb.txt -p 26000
gs_dump[port='26000'][student][2023-04-19 20:10:13]: The total objects number is
389.
gs_dump[port='26000'][student][2023-04-19 20:10:13]: [100.00%] 389 objects have
been dumped.
gs_dump[port='26000'][student][2023-04-19 20:10:13]: dump database student succe
ssfully
gs_dump[port='26000'][student][2023-04-19 20:10:13]: total time: 204 ms
omm@ecs-5138 ~]$
```

3. 以 omm 连接数据库系统，删除数据库 student 的模式 student 下的数据表 stu，查询。

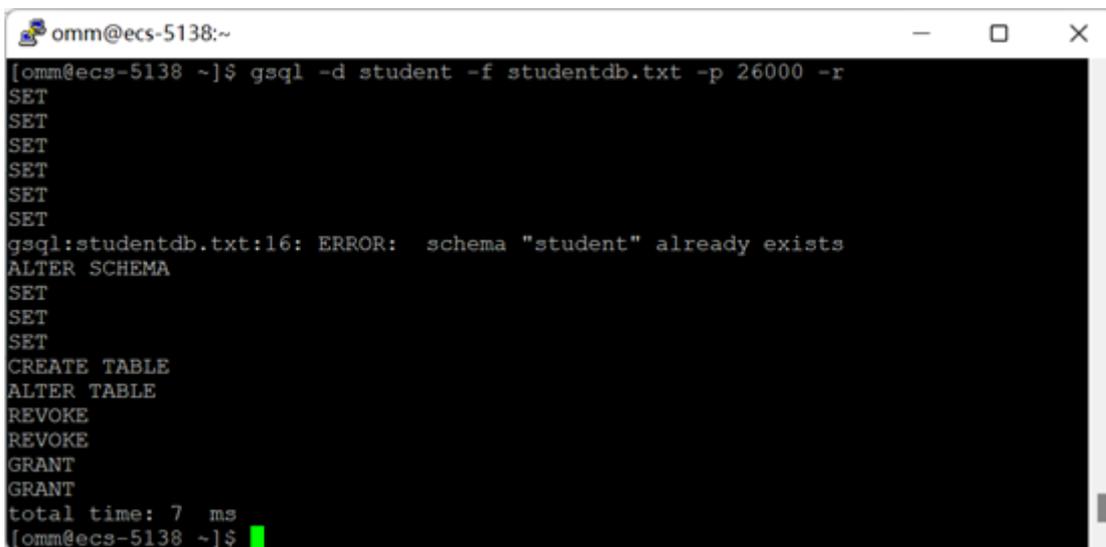
```
1 [omm@ecs-5138 ~]$ gsql -p 26000 -d student -U omm -r
2 set search_path to student;
3 drop table stu;
4 select * from stu;
```



```
omm@ecs-5138:~$ gsql -p 26000 -d student -U omm -r
student=# select * from stu;
ERROR:  relation "stu" does not exist on dn_6001
LINE 1: select * from stu;
                    ^
student=#
```

4. 在操作系统下，使用下列命令恢复数据库的内容。

```
1 student=# \q
2 [omm@ecs-5138 ~]$ gsql -d student -f studentdb.txt -p 26000 -r
```



```
omm@ecs-5138:~$ gsql -d student -f studentdb.txt -p 26000 -r
SET
SET
SET
SET
SET
SET
SET
gsql:studentdb.txt:16: ERROR:  schema "student" already exists
ALTER SCHEMA
SET
SET
SET
CREATE TABLE
ALTER TABLE
REVOKE
REVOKE
GRANT
GRANT
total time: 7 ms
omm@ecs-5138 ~]$
```

5. 登录系统，查询。

```
1 [omm@ecs-5138 ~]$ gsql -p 26000 -d student -U omm -r
2 set search_path to student;
3 select * from stu;
```



```
omm@ecs-5138:~$ gsql -p 26000 -d student -U omm -r
student=# select * from stu;
 name
-----
 汪洋
 张三
(2 rows)
student=#
```

6. 删除创建的数据库。

```
1 \c postgres
2 drop database student;
```

5 实验总结

过于简单，不想写了。