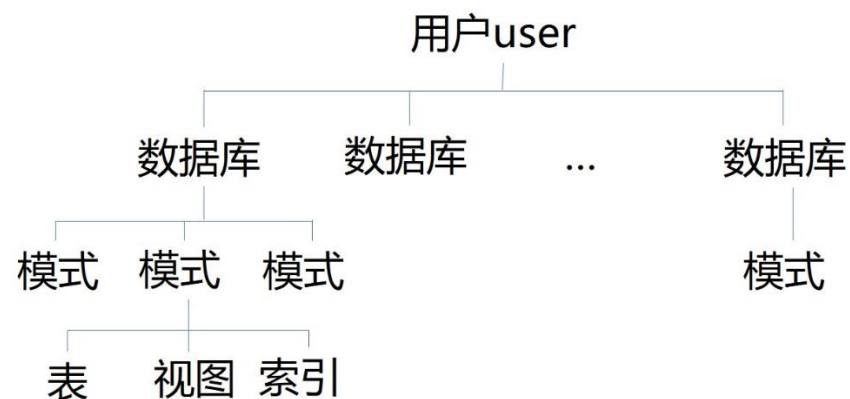


## 目录

1.用户，数据库，模式，数据库对象(表，视图，索引，触发器，函数，包，过程)之间的关系 .....	2
2.切换至 omm 操作系统用户环境 .....	2
3.数据库服务状态的检查、服务的启动和关闭 .....	2
4.数据库的连接和退出 .....	3
5.数据库的创建、查看和切换 .....	5
6. 用户密码的修改 .....	8
7.数据库用户的创建、查看、切换和删除 .....	8
8.模式的创建、设置、查看和删除 .....	13
9.基本表的查看.....	17
10.视图查看.....	20

11.查询缓冲区的清除 .....	20
12.参考资料.....	21

## 1.用户，数据库，模式，数据库对象(表，视图，索引，触发器，函数，包，过程)之间的关系



## 2.切换至 omm 操作系统用户环境

在数据库主节点服务器上，切换至 omm 操作系统用户环境。

```
[root@ecs-c9bf script]# su - omm
```

## 3.数据库服务状态的检查、服务的启动和关闭

用户登录 openGauss 数据库的前提：（1）数据库服务器已正常启动；（2）具有连接到数据库权限的合法用户。

(1) 查看服务是否启动

```
[omm@ecs-9a68 ~]$ gs_om -t status
```

```
-----  
cluster_state   : Normal  
redistributing  : No  
-----
```

(2) 启动服务 (如果没有启动, 即上述查询结果为: unavailable, 不是 normal)

```
[omm@ecs-c9bf ~]$ gs_om -t start
```

```
Starting cluster.
```

```
=====  
=====
```

```
Successfully started.
```

(3) 关闭服务

```
[omm@ecs-c9bf ~]$ gs_om -t stop
```

## 4.数据库的连接和退出

(1) gsql 是 openGauss 数据库提供的命令行方式的数据库连接工具, openGauss 数据库使用前必须先连接到数据库。

语法: gsql [-h 主机 IP] -d 目标数据库名 -p 端口号 -U 数据库用户名 -W 密码 -r

```
[omm@ecs-c9bf ~]$ gsql -d postgres -p 26000 -r
```

当结果显示为如下信息，则表示连接成功。

```
gsql ((openGauss 2.0.0 build 290d125f) compiled at 2021-03-31 02:59:43 commit 2143 last mr 131  
Non-SSL connection (SSL connection is recommended when requiring high-security)  
Type "help" for help.  
  
postgres=#
```

注: omm 账号的命令提示符为=#，普通用户的命令提示符为=>，可以通过这个细节分辨出当前用户是 omm 还是普通用户。

postgres 为 openGauss 安装完成后默认生成的数据库，初始可以连接到此数据库进行新数据库的创建，26000 为数据库主节点的端口号，需根据 openGauss 的实际情况做替换。

其它示例:

```
[omm@ecs-c9bf ~]$ gsql -h 10.180.123.163 -d postgres -U jack -p 26000
```

(2) 使用命令\q 退出当前数据库

退出 postgres 数据库。

```
postgres=# \q
```

使用新用户连接到此数据库。

```
[omm@ecs-c9bf ~]$ gsql -d db_tpcc -p 26000 -U joe -W Bigdata@123 -r
```

当结果显示为如下信息，则表示连接成功。

```
gsqll ((openGauss 2.0.0 build 290d125f) compiled at 2021-03-31 02:59:43 commit 2143 last mr 131  
Non-SSL connection (SSL connection is recommended when requiring high-security)  
Type "help" for help.  
  
db_tpcc=>
```

## 5.数据库的创建、查看和切换

(1) 创建数据库命令: create database ....;

```
postgres=# CREATE DATABASE db_tpcc OWNER joe;
```

当结果显示为如下信息, 则表示创建成功。

```
CREATE DATABASE
```

(2) 数据库的查看

- 使用 gsql 元命令 \l 显示当前用户能看到的所有数据库

```
demo=> \l
```

```
              List of databases
  Name      | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
demo       | whj   | UTF8     | C       | C     |
postgres   | omm   | UTF8     | C       | C     |
template0   | omm   | UTF8     | C       | C     | =c/omm          +
            |       |          |         |       | omm=CTc/omm
template1   | omm   | UTF8     | C       | C     | =c/omm          +
            |       |          |         |       | omm=CTc/omm
(4 rows)
```

- 查询当前数据库命令: `select current_catalog;`

```
demo=> select current_catalog;
current_database
-----
demo
(1 row)
```

- 通过查询系统表: `pg_database`

如果不清楚 `pg_database` 表包含哪些字段可先使用 `gspl` 元命令 `\d` 查看 `pg_database` 表结构

```
demo=> \d pg_database
      Table "pg_catalog.pg_database"
      Column      |      Type      | Modifiers
-----+-----+-----
 datname          | name           | not null
 datdba           | oid            | not null
 encoding         | integer        | not null
 datcollate       | name           | not null
 datatype         | name           | not null
 datistemplate    | boolean        | not null
 dataallowconn    | boolean        | not null
 datconndeflimit  | integer        | not null
 datlastsysoid    | oid            | not null
 datfrozenxid     | xid32          | not null
 dattablespace    | oid            | not null
 datcompatibility | name           | not null
 dataacl          | aclitem[]      |
 datfrozenxid64   | xid            |
Indexes:
    "pg_database_datname_index" UNIQUE, btree (datname) TABLESPACE pg_global, tablespace "pg_global"
    "pg_database_oid_index" UNIQUE, btree (oid) TABLESPACE pg_global, tablespace "pg_global"
Replica Identity: NOTHING
Tablespace: "pg_global"
```

- 使用命令： **SELECT datname FROM pg\_database;** 查看 pg\_database 表中当前用户 **有权查看** 的所有数据库列表

(3) 从当前数据库切换到另外一个数据库的命令： **\c database\_name**



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

- 当前用户为 whj，当前模式为 demo，现在要切换到用户 whj 有权访问的数据库 postgres，需要输入用户 whj 的密码，如果 whj 无权访问该数据库，则 openGauss 会提示相应错误。上述反馈表明切换成功。

## 6. 用户密码的修改

修改数据库 omm 用户密码，新密码假设修改为 Bigdata@123（建议用户自定义密码）。

```
postgres=# alter role omm identified by 'Bigdata@123' replace 'GaussDB@123';
```

注意：alter role omm identified by '新密码' replace '原密码';

显示如下信息即为修改成功：

```
ALTER ROLE
```

## 7. 数据库用户的创建、查看、切换和删除

（1）默认只有 openGauss 安装时创建的管理员用户可以访问初始数据库，您还可以创建其他数据库用户帐号。

```
postgres=# CREATE USER joe WITH PASSWORD "Bigdata@123";
```

或

```
postgres=# CREATE USER joe identified by "Bigdata@123";
```

当结果显示为如下信息，则表示创建成功。

```
CREATE ROLE
```

如上创建了一个用户名为 joe，密码为 Bigdata@123 的用户。

(2) 使用 select user 命令可以查看当前用户

```
demo=> select user,
current_user
-----
whj
(1 row)
```

或使用 SQL 命令：select username from pg\_user; --pg\_user 是系统表

(3) 使用 gsql 元命令 \c - username 命令从当前用户切换到另外一个用户

```

demo=# \c - whj
Password for user whj:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "demo" as user "whj".
demo=> create user user2 password 'Bigdata123';
CREATE ROLE
demo=> select  username, usesuper from pg_user;
 username | usesuper
-----+-----
 omm      | t
 whj      | f
 user1    | f
 user2    | f
(4 rows)

```

(4) 使用命令 `drop user username;` 删除用户

需要注意的是，在某个数据库下创建的在用户在其它数据库下被删除会报错。示例如下：

a. 在自建数据库 demo 下用户 whj 成功创建新用户 user2

```

demo=# \c - whj
Password for user whj:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "demo" as user "whj".
demo=> create user user2 password 'Bigdata123';
CREATE ROLE
demo=> select  username, usesuper from pg_user;
 username | usesuper
-----+-----
 omm      | t
 whj      | f
 user1    | f
 user2    | f
(4 rows)

```

b.管理员 omm 在数据库 demo 下删除 user2 是成功的

```

demo=> \c - omm
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "demo" as user "omm".
demo=# drop user user2;
DROP ROLE
demo=# select  username, usesuper from pg_user;
 username | usesuper
-----+-----
 omm      | t
 whj      | f
 user1    | f
(3 rows)

```

c.但在另外一个数据库 postgres 下删除用户 user2 则失败（先登录 postgres 数据库）

```
[omm@ecs-5801 ~]$ gsql -d postgres -p 26000 -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.

postgres=# drop user user2;
ERROR:  role "user2" cannot be dropped because some objects depend on it
DETAIL:  1 object in database demo
postgres=# █

postgres=# drop user user2;
ERROR:  role "user2" cannot be dropped because some objects depend on it
DETAIL:  1 object in database demo
postgres=# \c - whj
Password for user whj:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "whj".
postgres=> drop user user2;
ERROR:  role "user2" cannot be dropped because some objects depend on it
DETAIL:  1 object in database demo
postgres=> █
```

d.重新登录到 demo 数据库，再删除用户 user2，成功

```
postgres=> \q
[omm@ecs-5801 ~]$ gsql -d demo -p 26000 -U whj -W Bigdata123 -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.

demo=> drop user user2;
DROP ROLE
demo=> select username from pg_user;
 username
-----
 omm
 whj
 user1
(3 rows)
```

## 8.模式的创建、设置、查看和删除

(1) 模式的创建 create schema 命令

创建名为 joe 的 SCHEMA，并设置 joe 为当前的 schema。

```
db_tpcc=> CREATE SCHEMA joe AUTHORIZATION joe;
```

当结果显示为如下信息，则表示创建 SCHEMA 成功。

```
CREATE SCHEMA
```

(2) 设置当前模式命令 SET search\_path TO

将默认搜索路径设为 joe。

```
db_tpcc => SET search_path TO joe;
```

(3) 使用 gsql 的元命令 \dn 查看模式

```
postgres=> \dn
List of schemas
Name | Owner
-----+-----
whj   | whj
whj2  | whj
(2 rows)
```

一般情况下，omm 创建的新用户会在该用户下自动产生一个同名的模式。如果在新用户下使用命令 \dn 是没有显示模式名的。

新用户 whj 是 omm 在数据库 postgres 下创建的，但在用户 whj 下使用命令 \dn 查询不到模式 whj，下图（前图为当前用户的查询结果）。

```
demo=> select user
current_user
-----
whj
(1 row)
```

```
demo=> \dn
List of schemas
Name | Owner
-----+-----
(0 rows)
```

但切换到另外一个数据库 postgres 下，则可以看到。

```
demo=> \c postgres
Password for user whj:
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "postgres" as user "whj".
postgres=> \dn
List of schemas
Name | Owner
-----+-----
whj  | whj
(1 row)
```

这意味着，whj 模式是在数据库 postgres 下的。也可以通过 omm 用户下查看到。

```
[omm@ecs-5801 ~]$ gsql -d postgres -p 26000 -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.

postgres=# \dn
      List of schemas
      Name          | Owner
-----+-----
 cstore             | omm
 dbe_perf            | omm
 pkg_service         | omm
 public              | omm
 snapshot            | omm
 whj                  | whj
(6 rows)
```

(4) 使用 gsql 元命令 \d+ 查询模式



```
demo=> \d+
```

List of relations						
Schema	Name	Type	Owner	Size	Storage	Description
public	contacts	table	whj	72 kB	{orientation=row,compression=no}	
public	countries	table	whj	16 kB	{orientation=row,compression=no}	
public	customers	table	whj	112 kB	{orientation=row,compression=no}	
public	discounts	table	whj	16 kB	{orientation=row,compression=no}	
public	employees	table	whj	48 kB	{orientation=row,compression=no}	
public	inventories	table	whj	184 kB	{orientation=row,compression=no}	
public	locations	table	whj	16 kB	{orientation=row,compression=no}	
public	order_items	table	whj	152 kB	{orientation=row,compression=no}	
public	orders	table	whj	16 kB	{orientation=row,compression=no}	
public	palette_a	table	whj	8192 bytes	{orientation=row,compression=no}	
public	palette_b	table	whj	8192 bytes	{orientation=row,compression=no}	
public	product_categories	table	whj	16 kB	{orientation=row,compression=no}	
public	products	table	whj	72 kB	{orientation=row,compression=no}	
public	regions	table	whj	16 kB	{orientation=row,compression=no}	
public	salesman	view	whj	0 bytes		
public	salesman_contacts	view	whj	0 bytes		
public	warehouses	table	whj	16 kB	{orientation=row,compression=no}	

(17 rows)

- \d+命令比\dn 命令显示的内容更全面，它把当前用户下有权访问的当前数据库模式 schema 及其包含的数据库对象（表名、类型、属主、大小和存储方式）都列举出来。

#### (5) 模式删除命令 DROP SCHEMA

```
postgres=> drop schema whj2;  
ERROR:  cannot drop schema whj2 because other objects depend on it  
DETAIL:  table whj2.test2 depends on schema whj2  
HINT:   Use DROP ... CASCADE to drop the dependent objects too.  
postgres=>
```

有依赖时报错，默认为 RESTRICT。可使用命令 DROP SCHEMA...CASCADE;删除 SCHEMA 下的所有对象

- 无法删除当前模式。

参考: <https://docs.opengauss.org/zh/docs/3.1.1/docs/Developerguide/DROP-SCHEMA.html>

## 9.基本表的查看

使用系统视图 `pg_tables` 可以查看基本表的更多信息，包括模式名、表名、属主、表创建的时间戳。

(1) 使用 gsql 元命令 \d 查看表结构

```
demo=> \d pg_tables
View "pg_catalog.pg_tables"
  Column      |      Type      | Modifiers
-----+-----+-----
 schemaname   | name           |
 tablename    | name           |
 tableowner   | name           |
 tablespace   | name           |
 hasindexes   | boolean        |
 hasrules     | boolean        |
 hastriggers  | boolean        |
 tablecreator | name           |
 created      | timestamp with time zone |
 last_ddl_time | timestamp with time zone |
```

(2) 使用 SQL 命令 `select ... from pg_tables where....limit n;` 查看表的相关信息。

```
demo=> select schemaname,tablename,tableowner,created from pg_tables limit 20;
```

schemaname	tablename	tableowner	created
pg_catalog	pg_statistic	omm	
pg_catalog	pg_type	omm	
pg_catalog	pg_ts_dict	omm	
public	regions	whj	2022-04-11 22:38:16.510393+08
pg_catalog	pg_job_proc	omm	
public	locations	whj	2022-04-11 22:42:20.78232+08
public	countries	whj	2022-04-11 22:40:43.774355+08
public	product_categories	whj	2022-04-11 22:42:20.86076+08
public	order_items	whj	2022-04-11 22:42:20.966482+08
public	contacts	whj	2022-04-11 22:42:20.920358+08
public	orders	whj	2022-04-11 22:42:20.942015+08
public	employees	whj	2022-04-11 22:42:20.827448+08
public	customers	whj	2022-04-11 22:42:20.900683+08
public	products	whj	2022-04-11 22:42:20.880307+08
public	inventories	whj	2022-04-11 22:42:24.858252+08
public	warehouses	whj	2022-04-11 22:42:20.806506+08
public	discounts	whj	2022-04-13 20:30:56.360672+08
public	palette_a	whj	2022-04-15 22:05:34.362261+08
public	palette_b	whj	2022-04-15 22:06:03.55424+08
pg_catalog	pg_authid	omm	

(20 rows)

- limit 20 表明输出结果数为 20。若无 limit 限制，则全部输出。

## 10.视图查看

使用 gsql 命令 `\dv` 显示当前用户能看到的视图。

```
demo=> \dv
          List of relations
 Schema |          Name          | Type | Owner | Storage
-----+-----+-----+-----+-----
 public | salesman               | view |       |
 public | salesman_contacts     | view |       |
 public | v_region              | view |       |
(3 rows)
```

## 11.查询缓冲区的清除

为提示 SQL 查询的效率，查询语句在执行后会留在缓冲区，而不立即释放，可以使用 gsql 的 `\p` 和 `\r` 参数来控制是否清除。

`\r` 清除缓存中的 select 语句

`\p` 显示查询缓存中的内容

```
postgres=> \r
Query buffer reset (cleared).
postgres=> select * from test;
 a
---
(0 rows)

postgres=> \p
select * from test;
postgres=> \r
Query buffer reset (cleared).
postgres=> \p
Query buffer is empty.
```

## 12.参考资料

- openGauss 工具参考
- openGauss 管理员指南
- openGauss 开发者指南