

# openGauss数据库基本操作及OS基本操作



# 前言

---

- 数据库在部署成功后，需要通过一些工具来便捷地连接数据库，对数据库进行各种操作和调试。openGauss提供了一些数据库连接工具。通过这些工具可以方便地连接数据库并对数据库进行操作。而生产数据库大部分都是基于Linux操作系统安装的，所以对Linux操作要有一定的能力。
- 本章主要介绍如何通过数据库连接工具gsql连接openGauss数据库及Linux操作系统基本命令。

# 目标

---

- 学完本课程后，您将能够：
  - 掌握数据库连接工具gsql的使用和常用语法；
  - 掌握Linux操作系统基本命令；

# 目录

---

## 1. openGauss数据库基本操作

- gsql

## 2. LINUX 操作系统基本操作

# 连接数据库

- 连接数据库的客户端工具包括gsql 、 Data Studio 、 应用程序接口（如ODBC和JDBC） 。
  - gsql是openGauss自带的客户端工具。使用gsql连接数据库，可以交互式地输入、编辑、执行SQL语句；
  - Data Studio是一个集成开发环境，帮助数据库开发人员便捷地构建应用程序，以图形化界面形式提供数据库关键特性；

# 查看数据库状态

- gsql是openGauss提供的在命令行下运行的数据库连接工具。此工具除了具备操作数据库的基本功能，还提供了若干高级特性，便于用户使用。
- 查看数据库状态（在数据库安装用户下执行，如：omm）
  - 使用“gs\_ctl status;”命令查询openGauss实例情况；

```
[omm@db1 ~]$ gs_ctl status -D /opt/software/openGauss/data  
[2021-08-03 16:38:30.982][259692][][gs_ctl]: gs_ctl status,datadir is /opt/software/openGauss/data  
gs_ctl: server is running (PID: 259642)  
/opt/software/openGauss/bin/gaussdb "-D" "/opt/software/openGauss/data"
```

- 如上server is running 表示数据库正在运行。
- 数据库数据路径为"/opt/software/openGauss/data"。



# 启动、关闭数据库服务

- 启动数据库服务（在数据库安装用户下执行，如：omm）

```
[omm@db1 ~]$ gs_ctl start -D /opt/software/openGauss/data
[2021-08-03 16:40:50.741][259784][][gs_ctl]: gs_ctl started,datadir is /opt/software/openGauss/data
[2021-08-03 16:40:50.795][259784][][gs_ctl]: waiting for server to start...
.0 LOG: [Alarm Module]Host Name: opengauss01
.....
```

- 关闭数据库服务（在数据库安装用户下执行，如：omm）

```
[omm@db1 ~]$ gs_ctl stop -D /opt/software/openGauss/data
[2021-08-03 16:38:40.509][259708][][gs_ctl]: gs_ctl stopped ,datadir is /opt/software/openGauss/data
waiting for server to shut down..... done
server stopped
```

# 数据库登录、退出

- 数据库登录

```
[omm@db1 ~]$ gsql -d postgres -p 5432 -r
gsql ((GaussDB Kernel V500R001C20 build 0cd52e37) compiled at 2021-08-03 16:28:11 commit 0 last mr debug)
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=#
```

其中postgres为需要连接的数据库名称，5432为数据库主节点的端口号

- 数据库退出

```
postgres=#\q
```



# 常用语法 (1)

- **执行SQL语句**：支持交互式地键入并执行SQL语句，也可以执行一个文件中指定的SQL语句。
  - 以创建数据库**db\_tpcc**为例。

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

- 通常，输入的命令在遇到分号的时候结束。如果输入的命令没有错误，结果就会输出到屏幕上。

## 常用语法 (2)

- **执行gsql元命令**：元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化SQL输出结果，以及连接到新的数据库等。
  - 以列出openGauss中所有的数据库和描述信息为例。

```
postgres=# \l
List of databases
  Name  | Owner | Encoding | Collate  | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----
db_tpcc | omm   | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
postgres | omm   | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
template0 | omm   | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/omm      +
template1 | omm   | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/omm      +
(4 rows)
```

# 常见用法示例 -- 执行SQL语句 (1)

- 创建表:

```
postgres=# CREATE TABLE customer_t1
( c_customer_sk      integer,
  c_customer_id      char(5),
  c_first_name       char(6),
  c_last_name        char(8)
);
```

- 向表中插入数据:

方式1:

```
postgres=# INSERT INTO customer_t1(c_customer_sk, c_customer_id, c_first_name) VALUES (3769, 'hello', 'Grace');
```

方式2:

```
postgres=# INSERT INTO customer_t1 VALUES (3769, 'hello', 'Grace');
```

方式3:

```
postgres=# INSERT INTO customer_t1 (c_customer_sk, c_first_name) VALUES (3769, 'Grace');
```

## 常见用法示例 -- 执行SQL语句 (2)

- 向表中插入多行数据:

```
postgres=# INSERT INTO customer_t1 (c_customer_sk, c_customer_id, c_first_name) VALUES (6885, 'maps', 'Joes'),  
(4321, 'tpcds', 'Lily'),  
(9527, 'world', 'James');
```

- 查看表数据:

```
postgres=# SELECT * FROM customer_t1;  
c_customer_sk | c_customer_id | c_first_name | c_last_name  
-----+-----+-----+-----  
3769 | hello      | Grace      |  
3769 | hello      | Grace      |  
3769 |           | Grace      |  
6885 | maps       | Joes       |
```

## 常见用法示例 -- 执行SQL语句 (3)

- 从指定表插入数据到当前表:

```
postgres=# CREATE TABLE customer_t2
( c_customer_sk      integer,
  c_customer_id      char(5),
  c_first_name       char(6),
  c_last_name        char(8)
);
postgres=# INSERT INTO customer_t2 SELECT * FROM customer_t1;
```

- 更新表中数据

```
postgres=# UPDATE customer_t2 SET c_customer_sk = 9876 WHERE c_customer_sk = 9527;
```

## 常见用法示例 -- 执行SQL语句 (4)

- 删除表中单条数据:

```
postgres=# DELETE FROM customer_t2 WHERE c_customer_sk = 9876;
```

- 删除表中所有的数据

```
postgres=# DELETE FROM customer_t2;
```

或者

```
postgres=# TRUNCATE TABLE customer_t2;
```

- 删除表:

```
postgres=# DROP TABLE customer_t2 CASCADE;
```

# 常见用法示例 -- 执行SQL语句 (5)

- 创建普通视图:

```
postgres=# CREATE OR REPLACE VIEW MyView AS SELECT * FROM customer_t1 WHERE c_customer_sk = 6885;
```

- 创建物化视图

```
postgres=# CREATE MATERIALIZED VIEW MV_MyView AS SELECT * FROM customer_t1 WHERE c_customer_sk = 6885;
```

- 物化视图使用场景：报表统计、大表统计等，定期固化数据快照，避免对多表重复跑相同的查询。
- 物化视图使用注意事项：  
    不可在临时表或全局临时表上创建。  
    当基表数据发生变化时，需要使用刷新命令保持物化视图与基表同步。



## 常见用法示例 -- 执行SQL语句 (6)

- 创建存储过程:

```
postgres=# create table t_test(c1 int, c2 int);

postgres=# create or replace procedure insert_data
is
a int;
b int;
begin
a=1;
b=2;
insert into t_test values(a,b);
insert into t_test values(b,a);
end;
/
```

# 常见用法示例 -- 执行SQL语句 (7)

- 调用存储过程:

```
postgres=# call insert_data();
```

- 查看执行结果:

```
postgres=# select * from t_test;
```

```
c1 | c2
```

```
----+----
```

```
1 | 2
```

```
2 | 1
```

```
(2 rows)
```

## 常见用法示例 -- 执行SQL语句 (8)

- 创建会话级全局临时表：
  - 建表语句，使用 ON COMMIT PRESERVE ROWS;
  - 数据会话级可见，其他会话看不到数据，但表结构可见。

```
postgres=# CREATE GLOBAL TEMPORARY TABLE t_test2(  
id integer,  
lbl text  
) ON COMMIT PRESERVE ROWS;
```

# 常见用法示例 -- 执行SQL语句 (9)

- 事务级全局临时表：
  - 建表语句，使用 ON COMMIT DELETE ROWS;
  - 数据事务级可见，事务提交后数据删除。

```
postgres=# CREATE GLOBAL TEMPORARY TABLE t_test3(  
id integer,  
lbl text  
) ON COMMIT DELETE ROWS;
```

# 常见用法示例 -- 执行SQL语句 (10)

- 通过系统表pg\_database查询数据库列表:

```
postgres=# SELECT datname FROM pg_database;
 datname
-----
 template1
 db_tpcc
 template0
 postgres
```

- 查看表空间:

```
postgres=# SELECT spcname FROM pg_tablespace;
 spcname
-----
 pg_default
 pg_global
(2 rows)
```

# 常见用法示例 -- 执行SQL语句 (11)

- 查看数据库用户列表:

```
postgres=# SELECT username,usesysid,usecreatedb FROM pg_user;  
username | usesysid | usecreatedb  
-----+-----+-----  
omm      |      10 | t  
(1 row)
```

- 要查看用户属性:

```
postgres=# SELECT * FROM pg_authid;
```

- 查看所有角色:

```
postgres=# SELECT * FROM PG_ROLES;
```

# 常见用法示例 -- 执行gsql元命令 (1)

- 连接数据库时，可以使用如下命令获取帮助信息。

```
gsql --help
```

- 显示如下帮助信息：

```
.....
```

```
Usage:
```

```
gsql [OPTION]... [DBNAME [USERNAME]]
```

```
General options:
```

```
-c, --command=COMMAND  run only single command (SQL or internal) and exit
```

```
-d, --dbname=DBNAME    database name to connect to (default: "postgres")
```

```
-f, --file=FILENAME    execute commands from file, then exit
```

```
.....
```



## 常见用法示例 -- 执行gsql元命令 (2)

- 连接到数据库后，可以使用如下命令获取帮助信息。

```
help
```

- 显示如下帮助信息：

```
You are using gsql, the command-line interface to gaussdb.
```

```
Type: \copyright for distribution terms
```

```
    \h for help with SQL commands
```

```
    \? for help with gsql commands
```

```
    \g or terminate with semicolon to execute query
```

```
    \q to quit
```

# 常见用法示例 -- 执行gsql元命令 (3)

- 查看gsql的帮助信息。

- 查看版权信息：

```
postgres=# \copyright
```

- 查看openGauss支持的SQL语句的帮助。

- 例如，查看openGauss支持的所有SQL语句：

```
postgres=# \h
Available help:
ABORT
ALTER AGGREGATE
... ..
```

## 常见用法示例 -- 执行gsq元命令 (4)

- 例如，查看CREATE DATABASE命令的参数可使用下面的命令：

```
postgres=# \help CREATE DATABASE
Command:  CREATE DATABASE
Description: create a new database
Syntax:
CREATE DATABASE database_name
    [ [ WITH ] { [ OWNER [=] user_name ] |
    [ TEMPLATE [=] template ] |
    [ ENCODING [=] encoding ] |
    [ LC_COLLATE [=] lc_collate ] |
    [ LC_CTYPE [=] lc_ctype ] |
    [ DBCOMPATIBILITY [=] compatibility_type ] |
    [ TABLESPACE [=] tablespace_name ] |
    [ CONNECTION LIMIT [=] connlimit ] } [...] ];
```

## 常见用法示例 -- 执行gsql元命令 (5)

- 查看gsql命令的帮助。
  - 例如，查看gsql支持的命令：

```
postgres=# \?
General
\copyright      show PostgreSQL usage and distribution terms
\g [FILE] or ;   execute query (and send results to file or |pipe)
\h(\help) [NAME] help on syntax of SQL commands, * for all commands
\q              quit gsql
... ..
```

# 常见用法示例 -- 执行gsq元命令 (6)

- 查看表的定义:

```
postgres=# \d customer_t1
Table "public.customer_t1"

 Column | Type   | Modifiers 
-----+-----+-----
c_customer_sk | integer | 
c_customer_id | character(5) | 
c_first_name | character(6) | 
c_last_name  | character(8) |
```

- 查看表的属性

```
postgres=# \d+ tablename
```

# 常见用法示例 -- 执行gsq元命令 (7)

- 列举表:

```
postgres=# \dt
List of relations
Schema | Name      | Type | Owner | Storage
-----+-----+-----+-----+-----
public | customer_t1 | table | omm   | {orientation=row,compression=no}
public | t_test      | table | omm   | {orientation=row,compression=no}
public | t_test2     | table | omm   | {orientation=row,compression=no,on_commit_delete_rows=false}
public | t_test3     | table | omm   | {orientation=row,compression=no,on_commit_delete_rows=true}
(4 rows)
```

- 列举所有表、视图和索引:

```
postgres=# \d+
```

## 常见用法示例 -- 执行gsq元命令 (8)

- 列举schema:

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

- 查看索引:

```
postgres=# \di
```



# 常见用法示例 -- 执行gsq元命令 (9)

- 切换数据库:

```
postgres=# \c db_tpcc
```

```
Non-SSL connection (SSL connection is recommended when requiring high-security)
```

```
You are now connected to database "db_tpcc" as user "omm".
```

```
db_tpcc=#
```

- 切换用户:

```
postgres=# \c - jim
```

```
Password for user jim:
```

```
Non-SSL connection (SSL connection is recommended when requiring high-security)
```

```
You are now connected to database "postgres" as user "jim".
```

```
postgres=>
```

# 常见用法示例 -- 执行gsq元命令 (10)

- 查看表空间:

```
postgres=# \db
      List of tablespaces
  Name  | Owner | Location
-----+-----+-----
pg_default | omm  |
pg_global | omm  |
(2 rows)
```

- 退出数据库:

```
postgres=# \q
```

# 目录

---

1. openGauss数据库基本操作
- 2. LINUX 操作系统基本操作**
  - 相关基础命令

# linux命令行操作技巧

- **Tab 键补全**：可以用来自动补全命令或文件名，省时准确。
  - 未输入命令状态下，连续两次 tab 键列出所有可用命令；
  - 已输入部分命令名或文件名，按次 tab 键自动补全。
- **光标**
  - Up 方向键上（可以调出输入历史执行记录，快速执行命令）；
  - down 方向键下（配合 up 选择历史执行记录）；
  - Home 移动光标到本行开头；                      shift + G 移动光标到文件尾；
  - Ctl + C 终止当前程序；                      Ctl + L 清理屏幕显示；

# 常见命令 (1)

- **vi/vim**

- 文本编辑器，若文件存在则是编辑，若不存在则是创建并编辑文本。

- 命令语法：

```
vim [参数]
```

- 参数说明：可编辑的文件名。

- 命令示例：

- 编辑名为clusterconfig的xml文本：

```
vim clusterconfig.xml
```

## 常见命令 (2)

- vi/vim编辑器有以下三种模式：
  - 正常模式：其它模式下按Esc或Ctrl+[进入，左下角显示文件名或为空。
  - 插入模式：正常模式下按i键进入，左下角显示--INSERT--。
  - 可视模式：正常模式下按v键进入，左下角显示--VISUAL--。
- 退出命令（正常模式下）：
  - :wq 保存并退出。
  - :q! 强制退出并忽略所有更改。
  - :e! 放弃所有修改，并打开原有文件。

# 常见命令 (3)

- **cd**

- 显示当前目录的名称，或切换当前的目录（打开指定目录）。

- 命令语法：

```
cd [参数]
```

- 参数说明：

- 无参数：切换用户当前目录。
    - .：表示当前目录；
    - ..：表示上一级目录；
    - ~：表示home目录；
    - /：表示根目录。



# 常见命令 (4)

- cd 命令示例:

- 切换到usr目录下的bin目录中:

```
cd /usr/bin
```

- 切换到用户home目录:

```
cd ~
```

- 切换到根目录下:

```
cd /
```

- 切换到当前目录上一级目录(cd后面接两个.):

```
cd ..
```

# 常见命令 (5)

- **mv**

- 文件或目录改名(move (rename) files)或将文件或目录移入其它位置，经常用来备份文件或目录。

- 命令语法：

```
mv [选项] 参数1 参数2
```

- 常用选项：

- -b：若需覆盖文件，则覆盖前先行备份。

- 参数说明：

- 参数1：源文件或目录。

- 参数2：目标文件或目录。

## 常见命令 (6)

- mv 命令示例:

- 将文件python重命名为python.bak:

```
mv python python.bak
```

- 将/physical/backup目录下的所有文件和目录移到/data/dbn1目录下:

```
mv /physical/backup/* /data/dbn1
```

# 常见命令 (7)

- **yum**

- Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载和安装。

- 命令语法: `yum [options] [command] [package ...]`

- 常用选项:

- -h: 查看帮助;
    - -y: 当安装过程提示选择全部为 "yes";
    - -q: 不显示安装的过程。

# 常见命令 (8)

- yum参数说明:

- command: 要进行的操作。
- package: 安装的包名。

- yum 命令示例:

- 更新所有软件命令:

```
yum check-update
```

- 安装指定的软件:

```
yum install -y libaio-devel flex bison ncurses-devel glibc-devel patch lsb_release wget python3
```

# 常见命令 (9)

- **wget**

- wget是Linux下下载文件的最常用命令。wget支持HTTP,HTTPS和FTP协议,支持自动下载,即可以在用户退出系统后在后台执行,直到下载结束。

- 命令语法: `wget [选项] [URL]`

- 常用选项:

- -c: 接着下载没下载完的文件;
    - -b: 启动后转入后台执行;
    - -P: 指定下载目录;
    - -O: 变更下载文件名;
    - --ftp-user --ftp-password: 使用FTP用户认证下载。

# 常见命令 (10)

- wget参数说明:
  - 指定的文件下载URL地址。
- wget 命令示例:
  - 下载openGauss数据库安装文件到当前文件夹:

```
wget https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit.tar.gz
```

- 使用wget断点续传:

```
wget -c https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit.tar.gz
```

# 常见命令 (11)

- ln

- 为某一个文件在另外一个位置建立一个同步的链接。
- 当需要在不同的目录，用到相同的文件时，就不需要在每一个需要要的目录下都放一个必须相同的文件，我们只要在某个固定的目录，放上该文件，然后在其它的目录下用ln命令链接（link）它就可以，不必重复的占用磁盘空间。
- 命令语法： `ln [选项] 参数1 参数2`
- 常用选项：（不带选项为硬链接）
  - -b --删除，覆盖以前建立的链接；
  - -d --允许超级用户制作目录的硬链接；
  - -s --软链接(符号链接)。



# 常见命令 (12)

- ln 参数说明:

- 参数1: 源文件或目录;
- 参数2: 被链接的文件或目录。

- 命令示例:

- 为python3文件创建软链接/usr/bin/python, 如果python3丢失/usr/bin/python将失效:

```
ln -s python3 /usr/bin/python
```

- 为python3创建硬链接/usr/bin/python, python3与/usr/bin/python的各项属性相同:

```
ln python3 /usr/bin/python
```

# 常见命令 (13)

- **mkdir**

- 创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

- 命令语法: `mkdir [选项] [参数]`

- 常用选项:

- -p --可以是一个路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录（递归）；
    - -v --每次创建新目录都显示信息；
    - -m --设定权限<模式> (类似chmod)，而不是rwxrwxrwx减umask。

# 常见命令 (14)

- mkdir 参数说明:
  - 需要创建的目录。
- 命令示例:
  - 创建一个空目录:

```
mkdir test
```

- 递归创建多个目录:

```
mkdir -p /opt/software/openGauss
```

- 创建权限为777的目录(目录的权限为rwxrwxrwx) :

```
mkdir -m 777 test
```

# 常见命令 (15)

- **chmod**

- 更改文件权限。

- 命令语法: `chmod [选项] <mode> <file...>`

- 常用选项:

- -R, ---以递归的方式对目前目录下的所有文件与子目录进行相同的权限变更。

- 参数说明:

- mode: 权限设定字串, 详细格式如下: `[ugoa...][[+ -=][rwxX]...][,...],`

- 其中, [ugoa...]: u 表示该档案的拥有者, g 表示与该档案的拥有者属于同一个群体(group)者, o 表示其他以外的人, a 表示所有 (包含上面三者); [+ -=]: + 表示增加权限, - 表示取消权限, = 表示唯一设定权限; [rwxX]: r 表示可读取, w 表示可写入, x 表示可执行, X表示只有当该档案是个子目录或者该档案已经被设定过为可执行。

# 常见命令 (16)

- chmod 命令示例:

- 设置所有用户可读取文件 cluterconfig.xml:

```
chmod ugo+r cluterconfig.xml
```

或

```
chmod a+r cluterconfig.xml
```

- 设置当前目录下的所有档案与子目录皆设为任何人可读写:

```
chmod -R a+rw *
```

# 常见命令 (17)

- 数字权限使用格式:
- 这种使用方式中, 规定数字4、2和1表示读、写、执行权限, 即 $r=4, w=2, x=1$ 。
  - 例:  $rw x = 7 \ (4+2+1)$  ;  $rw = 6 \ (4+2)$  ;  $r-x = 5 \ (4+0+1)$  ;  $r-- = 4 \ (4+0+0)$  ;  $--x = 1 \ (0+0+1)$  ;
- 每个文件都可以针对三个粒度, 设置不同的 $rw x$ (读写执行)权限。即我们可以用三个8进制数字分别表示 拥有者、群组、其它组(  $u$ 、  $g$ 、  $o$ )的权限详情, 并用 $chmod$ 直接加三个8进制数字的方式直接改变文件权限。

# 常见命令 (18)

- 语法格式为：`chmod <abc> file...`
  - 其中，a,b,c各为一个数字，分别代表User、Group、及Other的权限，相当于简化版的 `chmod u=权限,g=权限,o=权限 file...`，而此处的权限将用8进制的数字来表示User、Group、及Other的读、写、执行权限。
- `chmod` 命令示例：
  - 赋予cluterconfig.xml文件可读可写可执行权限（所有权限）：  

```
chmod 777 cluterconfig.xml
```
  - 赋予/opt/software/openGauss目录下所有文件及其子目录 用户所有权限组可读可执行权限，其他用户可读可执行权限：

```
chmod -R 755 /opt/software/openGauss
```

# 常见命令 (19)

- **chown**

- 利用 chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。只有系统管理者(root)才有这样的权限。

- 命令语法：`chown [选项] user[:group] file...`

- 常用选项：

- -c : 显示更改的部分的信息；
    - -f : 忽略错误信息；
    - -R : 处理指定目录以及其子目录下的所有文件。



# 常见命令 (20)

- 参数说明:

- user : 新的文件拥有者的使用者 ID。
- group : 新的文件拥有者的使用者组(group)。
- file: 文件。

- chown 命令示例:

- 将文件 clusterconfig.xml 的拥有者设为omm，群体的使用者dbgrp:

```
chown omm:dbgrp /opt/software/openGauss/clusterconfig.xml
```

- 将目前目录下的所有文件与子目录的拥有者皆设为omm，群体的使用者dbgrp:

```
chown -R omm:dbgrp *
```

# 常见命令 (21)

- **cp**

- 复制文件或者目录。

- 命令语法: `cp [选项] 参数1 参数2`

- 常用选项:

- -f --如果目标文件无法打开则将其移除并重试(当 -n 选项存在时则不需再选此项);
- -n --不要覆盖已存在的文件(使前面的 -i 选项失效);
- -l --覆盖前询问(使前面的 -n 选项失效);
- -p --保持指定的属性(默认: 模式,所有权,时间戳), 如果可能保持附加属性: 环境、链接、xattr 等;
- -R,-r --复制目录及目录内的所有项目。

# 常见命令 (22)

- 参数说明:
  - 参数1: 源文件。
  - 参数2: 目标文件。
- cp 命令示例:
  - 将home目录中的abc文件复制到opt目录下:

```
cp /home/abc /opt
```

注: 目标文件存在时, 会询问是否覆盖。这是因为cp是cp -i的别名。

# 常见命令 (23)

- **rm**

- 删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。
- rm是一个危险的命令，使用的时候要特别当心，否则整个系统就会毁在这个命令（比如在/（根目录）下执行rm \* rf）。所以，我们在执行rm之前最好先确认一下在哪个目录，到底要删除什么东西，操作时保持高度清醒的头脑。
- 命令语法：`rm [选项] 文件`
- 常用选项：
  - -f --忽略不存在的文件，从不给出提示；
  - -r --指示rm将参数中列出的全部目录和子目录均递归地删除。

# 常见命令 (24)

- 参数说明：
  - 需要删除的文件或目录。
- rm 命令示例：
  - 删除文件：

```
rm qwe
```

注：输入rm qwe命令后，系统会询问是否删除，输入y后就会删除文件，不想删除文件则输入n。

- 强制删除某个文件：

```
rm -rf clusterconfig.log
```

# 常见命令 (25)

- **cat**

- 连接文件并在标准输出上输出。这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。
- 命令语法：
- 常用选项：`cat [选项] [参数]`
  - -E --在每行结束显示\$;
  - -n --由1开始对给所有输出行编号;
  - -b 或 --number-nonblank: 和 -n 相似，只不过对于空白行不编号;
  - -v --使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外。

# 常见命令 (26)

- 参数说明：
  - 可操作的文件名。
- cat 命令示例：
  - 显示testfile文件的内容：

```
cat testfile
```

- 向/etc/profile中追加内容（输入EOF表示结束追加）：

```
cat >>/etc/profile<<EOF  
>export LD_LIBRARY_PATH=$packagePath/script/gspylib/clib:$LD_LIBRARY_PATH  
>EOF
```

- 注：EOF是end of file的缩写，表示“文字流”（stream）的结尾。“文字流”可以是文件（file），也可以是标准输入（stdin）。

# 常见命令 (27)

- groupadd
  - 用于创建一个新的工作组。
  - 命令语法: `groupadd [-g gid [-o]] [-r] group`
  - 常用选项:
    - -g: 指定新建工作组的 id;
    - -r: 创建系统工作组, 系统工作组的组ID小于 500;
    - -o: 允许添加组 ID 号不唯一的工作组。
- 命令示例:
  - 创建一个新的组, 并添加组 ID。

```
groupadd -g 1000 dbgrp
```



# 常见命令 (28)

- useradd
  - 可用来建立用户帐号。
  - 命令语法: `useradd [option] username`
  - 常用选项:
    - -d <登入目录> 指定用户登入时的目录。
    - -g <群组> 初始群组。
    - -G <群组> 非初始群组。
    - -m 自动创建用户的家目录。
    - -M 不要创建用户的家目录。
    - -N 不要创建以用户名称为名的群组。
    - -s 指定用户登入后所使用的[shell](#)。

# 常见命令 (29)

- 参数说明:

- 要创建的用户名。

- 命令示例:

- 添加一般用户:

```
useradd omm
```

- 为添加的用户指定相应的用户组、登入目录、用户ID:

```
useradd -g dbgrp -u 1000 -d /home/omm omm
```

帐号建好之后，可用 passwd 设定帐号的密码；可用 userdel 删除帐号。

# 思考题

---

1. （判断）元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化SQL输出结果，以及连接到新的数据库等。（ ）
  - A. 正确
  - B. 错误
2. （单选）查看表的定义的元命令是。（ ）
  - A. \l
  - B. \q
  - C. \d
  - D. \h

# 本章总结

---

- gsql是openGauss提供的数据库连接工具，通过它可以对openGauss数据库进行各种操作，其执行命令方式包括二种：一、SQL命令；二、元命令。
- Linux操作系统常用命令包括文件编辑、用户及权限等等。

# 更多信息

---

**共同打造更优秀的国产数据库！**  
**共同构筑良好的国产数据库生态！**

官网: <https://opengauss.org>  
邮箱: [contact@opengauss.org](mailto:contact@opengauss.org)



微信公众号



添加微信小助手  
即可加入官方社群交流

# 缩略语表

缩略语	英文全称	解释
ODBC	Open Database Connectivity	开放数据库互连
JDBC	Java DataBase Connectivity	java数据库连接

# Thank you.

把数字世界带入每个人、每个家庭、  
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and  
organization for a fully connected,  
intelligent world.

**Copyright©2020 Huawei Technologies Co., Ltd.  
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

