《数据库》

openGauss数据库安全指导手册



华为技术有限公司

目录

[前 言 3](#_Toc53423566)

[简介 3](#_Toc53423567)

[内容描述 3](#_Toc53423568)

[前置条件 3](#_Toc53423569)

[实验环境说明 3](#_Toc53423570)

[实验概览 4](#_Toc53423571)

[1 用户权限控制 5](#_Toc53423572)

[1.1 实验介绍 5](#_Toc53423573)

[1.1.1 关于本实验 5](#_Toc53423574)

[1.1.2 实验目的 5](#_Toc53423575)

[1.2 用户 5](#_Toc53423576)

[1.2.1 创建、修改、删除用户 5](#_Toc53423577)

[1.3 角色 7](#_Toc53423578)

[1.3.1 创建、修改、删除角色 7](#_Toc53423579)

[1.4 Schema 8](#_Toc53423580)

[1.4.1 创建、修改、删除Schema 8](#_Toc53423581)

[1.5 用户权限设置及回收 9](#_Toc53423582)

[1.5.1 将系统权限授权给用户或者角色 10](#_Toc53423583)

[1.5.2 将数据库对象授权给角色或用户 10](#_Toc53423584)

[1.5.3 将用户或者角色的权限授权给其他用户或角色 11](#_Toc53423585)

[1.5.4 权限回收 11](#_Toc53423586)

[1.6 安全策略设置 12](#_Toc53423587)

[1.6.1 设置账户安全策略 12](#_Toc53423588)

[1.6.2 设置账号有效期 13](#_Toc53423589)

[1.6.3 设置密码安全策略 14](#_Toc53423590)

[2 审计 17](#_Toc53423591)

[2.1 实验介绍 17](#_Toc53423592)

[2.1.1 关于本实验 17](#_Toc53423593)

[2.1.2 实验目的 17](#_Toc53423594)

[2.2 审计开、关 17](#_Toc53423595)

[2.3 查看审计结果 19](#_Toc53423596)

[2.4 维护审计日志 20](#_Toc53423597)

[3 附录一：openGauss数据库基本操作 22](#_Toc53423598)

[3.1 查看数据库对象 22](#_Toc53423599)

[3.2 其他操作 23](#_Toc53423600)

# 前 言

## 简介

本实验适用于 openGauss数据库，通过该实验可以顺利完成对数据库用户权限的控制及各种审计。

## 内容描述

本实验主要内容为用户权限控制（包括用户、角色、权限的授予、权限的回收）和各审计。

## 前置条件

openGauss数据库安全实验需要掌握openGauss数据库的基本操作和SQL语法，openGauss数据库支持SQL2003标准语法，数据库基本操作参见**附录一**。

## 实验环境说明

* 组网说明

本实验环境为华为云 ECS 服务器 + openGauss数据库。

* 设备介绍

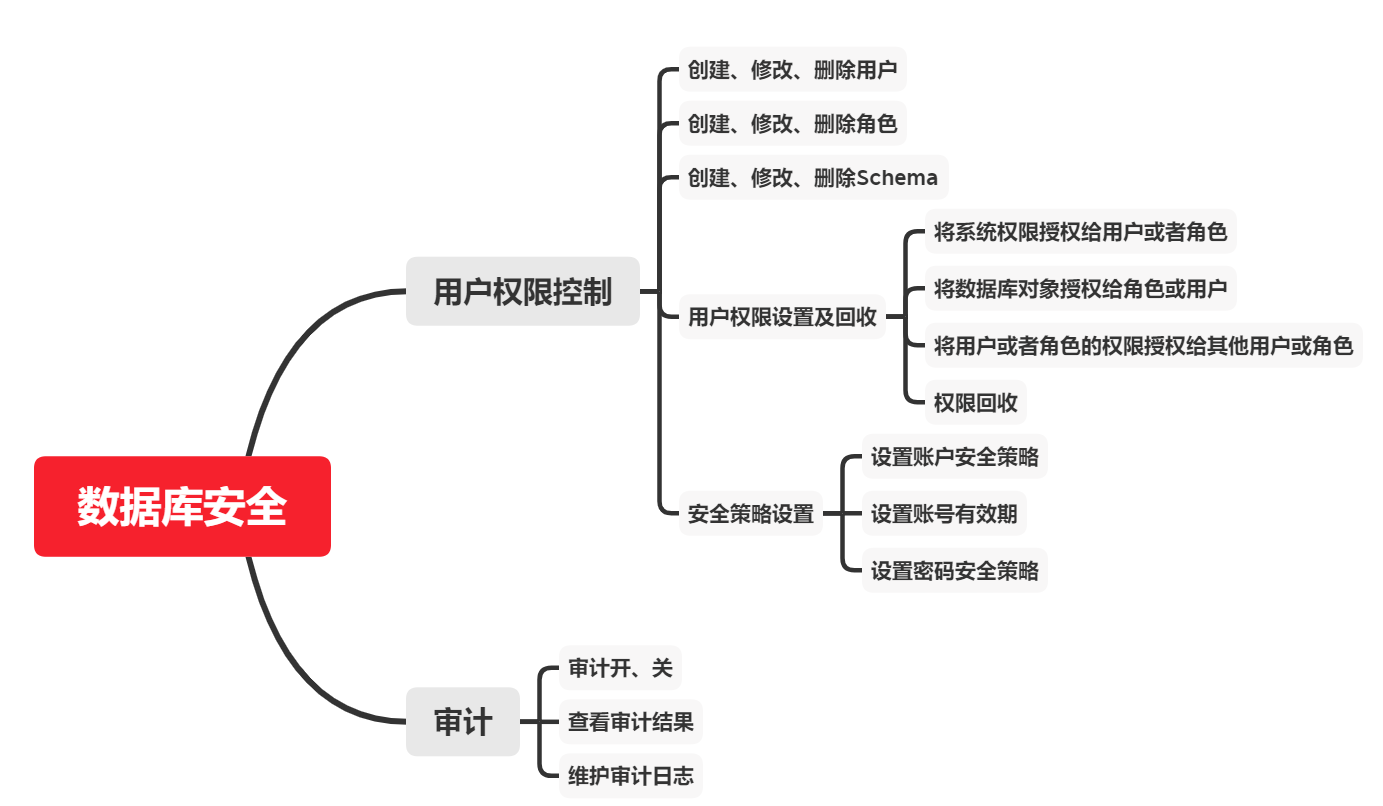
为了满足数据库原理与实践课程实验需要，建议每套实验环境采用以下配置：

设备名称、型号与版本的对应关系如下：

设备明细表

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 软件版本 |
| 数据库 | openGauss | openGauss 1.1.0 |
| 操作系统 | openEuler | openEuler 20.3LTS |

## 实验概览



本实验概览图

# 用户权限控制

## 实验介绍

### 关于本实验

本实验主要描述用户的创建管理、角色的创建管理、Schema的创建管理、用户权限设置、用户安全策略设置。

### 实验目的

* 掌握用户、角色、Schema的创建及管理；
* 掌握用户权限的授予各回收；
* 掌握用户安全策略如何设置。

## 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。

### 创建、修改、删除用户

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

连接数据库后，进入SQL命令界面。创建用户jim，登录密码为Bigdata@123。

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

CREATE ROLE

说明：密码规则如下：

* 密码默认不少于8个字符；
* 不能与用户名及用户名倒序相同；
* 至少包含大写字母（A-Z），小写字母（a-z），数字（0-9），非字母数字字符（限定为~!@#$%^&\*()-\_=+\|[{}];:,<.>/?）四类字符中的三类字符；
* 创建用户时，应当使用双引号或单引号将用户密码括起来。

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 | spillspacelimit

---------+----------+-------------+----------+-----------+---------+----------+---

omm | 10 | t | t | t | t | \*\*\*\*\*\*\*\* | | | default\_pool | 0 | | | |

|

**jim**  | 16389 | f | f | f | f | \*\*\*\*\*\*\*\* | | | default\_pool | 0 | | | |

|

(2 rows)

创建有“创建数据库”权限的用户，则需要加CREATEDB关键字。

postgres=# **CREATE USER dim CREATEDB PASSWORD 'Bigdata@123';**

CREATE ROLE

将用户jim的登录密码由Bigdata@123修改为Abcd@123。

postgres=# **ALTER USER jim IDENTIFIED BY 'Abcd@123' REPLACE 'Bigdata@123';**

ALTER ROLE

为用户jim追加有创建角色的CREATEROLE权限。

postgres=# **ALTER USER jim CREATEROLE;**

ALTER ROLE

锁定jim帐户。

postgres=# **ALTER USER jim ACCOUNT LOCK;**

ALTER ROLE

解锁jim帐户。

postgres=# **ALTER USER jim ACCOUNT UNLOCK;**

ALTER ROLE

删除用户。

postgres=# **DROP USER jim CASCADE;**

DROP ROLE

退出数据库

postgres=# **\q**

## 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。

### 创建、修改、删除角色

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

创建一个角色，名为manager，密码为Bigdata@123。

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

CREATE ROLE

创建一个角色，从2021年1月1日开始生效，到2021年6月1日失效。

postgres=# **CREATE ROLE miriam WITH LOGIN PASSWORD 'Bigdata@123' VALID BEGIN '2021-01-01' VALID UNTIL '2021-06-01';**

CREATE ROLE

修改角色manager的密码为abcd@123。

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

ALTER ROLE

修改角色manager为系统管理员。

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

ALTER ROLE

删除角色manager。

postgres=# **DROP ROLE manager;**

DROP ROLE

查看角色。

postgres=# **SELECT \* FROM PG\_ROLES;**

rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlogin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rol

password | rolvalidbegin | rolvaliduntil | rolrespool | rolparentid | roltabspace | rolconfig | oid | roluseft | rolkind | nodegroup | rolte

mpspace | rolspillspace

---------+----------+------------+---------------+-------------+--------------+-------------+----------------+---------------

omm | t | t | t | t | t | t | t | t | t | -1 | \*\*\*

\*\*\*\*\* | | | default\_pool | 0 | | | 10 | t | n | |

|

dim | f | t | f | t | f | t | f | f | f | -1 | \*\*\*

\*\*\*\*\* | | | default\_pool | 0 | | | 16393 | f | n | |

|

miriam | f | t | f | f | f | t | f | f | f | -1 | \*\*\*

\*\*\*\*\* | 2020-07-01 00:00:00+08 | 2020-12-01 00:00:00+08 | default\_pool | 0 | | | 16401 | f | n | |

|

(3 rows)

退出数据库

postgres=# **\q**

## Schema

Schema又称作模式。通过管理Schema，允许多个用户使用同一数据库而不相互干扰。

每个数据库包含一个或多个Schema。

在数据库创建用户时，系统会自动帮助用户创建一个同名Schema。

### 创建、修改、删除Schema

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

创建模式ds。

postgres=# **CREATE SCHEMA ds;**

CREATE SCHEMA

将当前模式ds更名为ds\_new。

postgres=# **ALTER SCHEMA ds RENAME TO ds\_new;**

ALTER SCHEMA

创建用户jack。

postgres=# **CREATE USER jack PASSWORD 'Bigdata@123';**

CREATE ROLE

将DS\_NEW的所有者修改为jack。

postgres=# **ALTER SCHEMA ds\_new OWNER TO jack;**

ALTER SCHEMA

查看Schema所有者。

postgres=# **SELECT s.nspname,u.usename AS nspowner FROM pg\_namespace s, pg\_user u WHERE s.nspowner = u.usesysid;**

nspname | nspowner

--------------------+----------

pg\_toast | omm

cstore | omm

dbe\_perf | omm

snapshot | omm

pg\_catalog | omm

public | omm

information\_schema | omm

dim | dim

jack | jack

ds\_new | jack

(10 rows)

删除用户jack和模式ds\_new。

postgres=# **DROP SCHEMA ds\_new;**

DROP SCHEMA

postgres=# **DROP USER jack;**

DROP ROLE

退出数据库

postgres=# **\q**

## 用户权限设置及回收

使用GRANT命令进行用户授权包括以下三种场景：

* 将系统权限授权给角色或用户。
* 将数据库对象授权给角色或用户。
* 将角色或用户的权限授权给其他角色或用户。

### 将系统权限授权给用户或者角色

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

创建名为joe的用户，并将sysadmin权限授权给joe。

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

CREATE ROLE

postgres=# **GRANT ALL PRIVILEGES TO joe;**

ALTER ROLE

### 将数据库对象授权给角色或用户

撤销joe用户的sysadmin权限，然后创建tpcds模式，并给tpcds模式下创建一张reason表。

postgres=# **REVOKE ALL PRIVILEGES FROM joe;**

ALTER ROLE

postgres=# **CREATE SCHEMA tpcds;**

CREATE SCHEMA

postgres=# **CREATE TABLE tpcds.reason**

**(**

**r\_reason\_sk INTEGER NOT NULL,**

**r\_reason\_id CHAR(16) NOT NULL,**

**r\_reason\_desc VARCHAR(20)**

**);**

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。

postgres=# **GRANT USAGE ON SCHEMA tpcds TO joe;**

GRANT

postgres=# **GRANT ALL PRIVILEGES ON tpcds.reason TO joe;**

GRANT

授权成功后，joe用户就拥有了tpcds.reason表的所有权限，包括增删改查等权限。

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=# **GRANT select (r\_reason\_sk,r\_reason\_id,r\_reason\_desc),update (r\_reason\_desc) ON tpcds.reason TO joe;**

GRANT

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

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。

postgres=# **CREATE ROLE tpcds\_manager PASSWORD 'Bigdata@123';**

CREATE ROLE

postgres=# **GRANT USAGE,CREATE ON SCHEMA tpcds TO tpcds\_manager;**

GRANT

### 将用户或者角色的权限授权给其他用户或角色

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。

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

CREATE ROLE

postgres=# **GRANT joe TO manager WITH ADMIN OPTION;**

GRANT ROLE

创建用户senior\_manager，将用户manager的权限授权给该用户。

postgres=# **CREATE ROLE senior\_manager PASSWORD 'Bigdata@123';**

CREATE ROLE

postgres=# **GRANT manager TO senior\_manager;**

GRANT ROLE

### 权限回收

撤销权限，并清理用户。

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

退出数据库

postgres=# **\q**

注意：实验完成后请尽量清理本实验的对象，以免影响与其它实验产生冲突。

## 安全策略设置

为了保证帐户安全，如果用户输入密码次数超过一定次数（failed\_login\_attempts），系统将自动锁定该帐户，默认值为10。次数设置越小越安全，但是在使用过程中会带来不便。

当帐户被锁定时间超过设定值（password\_lock\_time），则当前帐户自动解锁，默认值为1天。时间设置越长越安全，但是在使用过程中会带来不便。

### 设置账户安全策略

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

配置failed\_login\_attempts参数。

查看已配置的参数。

postgres=# **SHOW failed\_login\_attempts;**

failed\_login\_attempts

-----------------------

10

(1 row)

如果显示结果不为10，执行“\q”命令退出数据库。

然后在操作系统 omm 用户下执行如下命令设置成默认值10。

**gs\_guc reload -D /gaussdb/data/dbnode -c "failed\_login\_attempts=10"**

注意：/gaussdb/data/dbnode 指的是数据目录，要根据自己实际情况调整。

比如数据目录/gaussdb/data/db1，执行结果如下。

[omm@ecs-700c ~]$ **gs\_guc reload -D /gaussdb/data/db1 -c "failed\_login\_attempts=9"**

expected instance path: [/gaussdb/data/db1/postgresql.conf]

gs\_guc reload: failed\_login\_attempts=9: [/gaussdb/data/ecs-700c/postgresql.conf]

server signaled

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

配置password\_lock\_time参数。

查看已配置的参数。

[omm@ecs-c2f0 ~]$ **gsql -d postgres -p 26000 -r**

gsql ((openGauss 1.0.0 build 0bd0ce80) compiled at 2020-06-30 18:19:23 commit 0 last mr )

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

Type "help" for help.

postgres=# **SHOW password\_lock\_time;**

password\_lock\_time

--------------------

1

(1 row)

如果显示结果不为1，执行“\q”命令退出数据库。

然后在操作系统 omm 用户下执行如下命令设置成默认值1。

**gs\_guc reload -N all -I all -c "password\_lock\_time=1"**

执行结果如下：

[omm@ecs-700c ~]$ **gs\_guc reload -N all -I all -c "password\_lock\_time=1"**

NOTICE: password\_lock\_time and failed\_login\_attempts must have positive for lock and unlock functions to work as.

Begin to perform gs\_guc for all datanodes.

Total instances: 1. Failed instances: 0.

Success to perform gs\_guc!

### 设置账号有效期

创建新用户时，需要限制用户的操作期限（有效开始时间和有效结束时间）。

不在有效操作期内的用户需要重新设定帐号的有效操作期。

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

创建用户并制定用户的有效开始时间和有效结束时间。

postgres=# **CREATE USER joe WITH PASSWORD 'Bigdata@123' VALID BEGIN '2020-07-10 08:00:00' VALID UNTIL '2022-10-10 08:00:00';**

CREATE ROLE

重新设定帐号的有效期。

postgres=# **ALTER USER joe WITH VALID BEGIN '2020-11-10 08:00:00' VALID UNTIL '2021-11-10 08:00:00';**

ALTER ROLE

退出数据库

postgres=# **\q**

### 设置密码安全策略

用户密码存储在系统表pg\_authid中，为防止用户密码泄露，openGauss对用户密码可进行加密存储、密码复杂度设置、密码重用天数设置、密码有效期限设置等。

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

配置的加密算法。

查看已配置的参数。

postgres=# **SHOW password\_encryption\_type;**

password\_encryption\_type

--------------------------

2

(1 row)

如果显示结果为0或1，执行“\q”命令退出数据库。

然后在操作系统 omm 用户下执行如下命令将其设置为安全的加密算法。

**gs\_guc reload -N all -I all -c "password\_encryption\_type=2"**

注意说明：

* 当参数password\_encryption\_type设置为0时，表示采用md5方式对密码加密。md5为不安全的加密算法，不建议使用。
* 当参数password\_encryption\_type设置为1时，表示采用sha256和md5方式对密码加密。其中包含md5为不安全的加密算法，不建议使用。
* 当参数password\_encryption\_type设置为2时，表示采用sha256方式对密码加密，为默认配置。

配置密码安全参数。

查看已配置的参数。

postgres=# **SHOW password\_policy;**

password\_policy

-----------------

1

(1 row)

如果显示结果不为1，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值1。

**gs\_guc reload -N all -I all -c "password\_policy=1"**

注意说明：

* 参数password\_policy设置为1时表示采用密码复杂度校验，默认值；
* 参数password\_policy设置为0时表示不采用任何密码复杂度校验，设置为0会存在安全风险，不建议设置为0，即使需要设置也要将所有openGauss节点中的password\_policy都设置为0才能生效。

配置密码重用。

查看不可重用天数已配置的参数。

postgres=# **SHOW password\_reuse\_time;**

password\_reuse\_time

---------------------

60

(1 row)

如果显示结果不为60，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值60。

**gs\_guc reload -N all -I all -c "password\_reuse\_time=60"**

查看不可重用次数已配置的参数。

postgres=# **SHOW password\_reuse\_max;**

password\_reuse\_max

0

(1 row)

如果显示结果不为0，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值0。

**gs\_guc reload -N all -I all -c "password\_reuse\_max = 0"**

注意说明：

* 不可重用天数默认值为60天，不可重用次数默认值是0；
* 这两个参数值越大越安全，但是在使用过程中会带来不便，其默认值符合安全标准，用户可以根据需要重新设置参数，提高安全等级。

配置密码有效期限。

数据库用户的密码都有密码有效期（[password\_effect\_time](https://opengauss.org/zh/docs/1.0.0/docs/Developerguide/%E5%AE%89%E5%85%A8%E5%92%8C%E8%AE%A4%E8%AF%81%EF%BC%88postgresql-conf%EF%BC%89.html#zh-cn_topic_0237124696_zh-cn_topic_0059778664_sfcc6124115734728917a548a8bd8f0d4)），当达到密码到期提醒天数（[password\_notify\_time](https://opengauss.org/zh/docs/1.0.0/docs/Developerguide/%E5%AE%89%E5%85%A8%E5%92%8C%E8%AE%A4%E8%AF%81%EF%BC%88postgresql-conf%EF%BC%89.html#zh-cn_topic_0237124696_zh-cn_topic_0059778664_s1beab889ab8d49848ef28bf60c10d8f7)）时，系统会在用户登录数据库时提示用户修改密码。

配置password\_effect\_time参数。

查看已配置的参数。

postgres=# **SHOW password\_effect\_time;**

password\_effect\_time

----------------------

90

(1 row)

如果显示结果不为90，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值90（不建议设置为0）。

**gs\_guc reload -N all -I all -c "password\_effect\_time = 90"**

配置password\_notify\_time参数。

查看已配置的参数。

postgres=# **SHOW password\_notify\_time;**

password\_notify\_time

----------------------

7

(1 row)

如果显示结果不为7，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值7（不建议设置为0）。

**gs\_guc reload -N all -I all -c "password\_notify\_time = 7"**

用户权限控制实验结束。

# 审计

## 实验介绍

### 关于本实验

数据库安全对数据库系统来说至关重要，openGauss将用户对数据库的所有操作写入审计日志，数据库安全管理员可以利用这些日志信息，重现导致数据库现状的一系列事件，找出非法操作的用户、时间和内容等。

本实验主要描述如何来设置数据库审计，主要包括审计开关、查看审计结果、维护审计日志。

### 实验目的

掌握如何设置数据库审计及审计日志的查看。

## 审计开、关

审计总开关[audit\_enabled](https://opengauss.org/zh/docs/1.0.0/docs/Developerguide/%E5%AE%A1%E8%AE%A1%E5%BC%80%E5%85%B3.html#zh-cn_topic_0237124745_zh-cn_topic_0059777744_sb3d1b703f24c49f096b36087a60fdfcd)支持动态加载。在数据库运行期间修改该配置项的值会立即生效，无需重启数据库。默认值为on，表示开启审计功能。

除了审计总开关，各个审计项也有对应的开关。只有开关开启，对应的审计功能才能生效。

各审计项的开关支持动态加载。在数据库运行期间修改审计开关的值，不需要重启数据库便可生效。

审计总开关设置。

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

查看已配置审计总开关参数。

postgres=# **show audit\_enabled;**

audit\_enabled

---------------

on

(1 row)

如果显示结果不为on，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值on（不建议设置为off）。

**gs\_guc reload -N all -I all -c "audit\_enabled = on"**

审计项开关设置。

查看已配置审计项参数。

以下以数据库启动、停止、恢复和切换审计项为例。

postgres=# **show audit\_database\_process;**

audit\_database\_process

------------------------

1

(1 row)

如果显示结果不为1，执行“\q”命令退出数据库。

然后在操作系统 omm下执行如下命令设置成默认值1。

**gs\_guc reload -N all -I all -c " audit\_database\_process = 1"**

注意说明：

* 用户登录、注销审计 。

参数：audit\_login\_logout。

默认值为7，表示开启用户登录、退出的审计功能。设置为0表示关闭用户登录、退出的审计功能。不推荐设置除0和7之外的值。

* 数据库启动、停止、恢复和切换审计。

参数：audit\_database\_process。

默认值为1，表示开启数据库启动、停止、恢复和切换的审计功能。

* 用户锁定和解锁审计 。

参数：audit\_user\_locked。

默认值为1，表示开启审计用户锁定和解锁功能。

* 用户访问越权审计。

参数：audit\_user\_violation。

默认值为0，表示关闭用户越权操作审计功能。

* 授权和回收权限审计 。

参数：audit\_grant\_revoke。

默认值为1，表示开启审计用户权限授予和回收功能。

* 数据库对象的CREATE，ALTER，DROP操作审计。

参数：audit\_system\_object。

默认值为12295，表示只对DATABASE、SCHEMA、USER、DATA SOURCE这四类数据库对象的CREATE、ALTER、DROP操作进行审计。

* 具体表的INSERT、UPDATE和DELETE操作审计。

参数：audit\_dml\_state。

默认值为0，表示关闭具体表的DML操作（SELECT除外）审计功能。

* SELECT操作审计。

参数：audit\_dml\_state\_select。

默认值为0，表示关闭SELECT操作审计功能。

* COPY审计。

参数：audit\_copy\_exec。

默认值为0，表示关闭copy操作审计功能。

* 存储过程和自定义函数的执行审计 。

参数：audit\_function\_exec。

默认值为0，表示不记录存储过程和自定义函数的执行审计日志。

* SET审计 参数：audit\_set\_parameter。

默认值为1，表示记录set操作审计日志。

## 查看审计结果

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

查询审计记录。

postgres=# **SELECT time,type,result,username,object\_name FROM pg\_query\_audit('2021-01-01 10:00:00','2021-08-15 09:47:33');**

time | type | result | username | object\_name

------------------------+---------------+--------+----------+-----------------

2020-07-10 10:00:59+08 | user\_logout | ok | omm | postgres

2020-07-10 10:02:39+08 | user\_logout | ok | omm | postgres

2020-07-10 10:08:35+08 | login\_success | ok | omm | postgres

2020-07-10 10:08:35+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 10:08:40+08 | user\_logout | ok | omm | postgres

2020-07-10 10:08:42+08 | login\_success | ok | omm | postgres

2020-07-10 10:08:42+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 10:14:56+08 | login\_success | ok | omm | postgres

2020-07-10 10:14:56+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 10:16:33+08 | ddl\_database | ok | omm | mydb

2020-07-10 10:22:42+08 | user\_logout | ok | omm | postgres

2020-07-10 10:26:51+08 | user\_logout | ok | omm | postgres

2020-07-10 10:29:08+08 | login\_success | ok | omm | postgres

2020-07-10 10:29:08+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 10:48:23+08 | user\_logout | ok | omm | postgres

2020-07-10 10:55:02+08 | login\_success | ok | omm | postgres

2020-07-10 10:55:02+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 10:55:48+08 | login\_success | ok | omm | postgres

2020-07-10 10:55:48+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 11:17:25+08 | login\_success | ok | omm | postgres

2020-07-10 11:17:25+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 11:31:06+08 | user\_logout | ok | omm | postgres

2020-07-10 11:34:05+08 | login\_success | ok | omm | postgres

2020-07-10 11:34:05+08 | set\_parameter | ok | omm | connection\_info

2020-07-10 11:40:11+08 | login\_success | ok | omm | postgres

2020-07-10 11:40:11+08 | **set\_parameter** | ok | omm | connection\_info

退出数据库

postgres=# **\q**

## 维护审计日志

设置自动删除审计日志。

启动服务器，再使用gsql客户端以管理员用户身份连接postgres数据库，假设端口号为26000。

**gsql -d postgres -p 26000 -r**

配置审计文件占用磁盘空间的大小（audit\_space\_limit）。

查看已配置的参数。

postgres=# **SHOW audit\_space\_limit;**

audit\_space\_limit

-------------------

1GB

(1 row)

如果显示结果不为1GB（1024MB），执行“\q”命令退出数据库。

然后在操作系统 omm用户下执行如下命令设置成默认值1024MB。

**gs\_guc reload -N all -I all -c "audit\_space\_limit=1024MB"**

配置审计文件个数的最大值（audit\_file\_remain\_threshold）。

查看已配置的参数。

postgres=# **SHOW audit\_file\_remain\_threshold;**

audit\_file\_remain\_threshold

-----------------------------

1048576

(1 row)

如果显示结果不为1048576，执行“\q”命令退出数据库。

然后在操作系统 omm用户执行如下命令设置成默认值1048576。

**gs\_guc reload -N all -I all -c "audit\_file\_remain\_threshold=1048576"**

审计实验结束。

# 附录一：openGauss数据库基本操作

## 查看数据库对象

* 查看帮助信息：

postgres=# \?

* 切换数据库：

*postgres*=# \c dbname

* 列举数据库：

使用\l元命令查看数据库系统的数据库列表。

postgres=# \l

使用如下命令通过系统表pg\_database查询数据库列表。

*postgres*=# SELECT datname FROM pg\_database;

* 列举表：

postgres=# \dt

* 列举所有表、视图和索引：

postgres=# \d+

* 使用gsql的\d+命令查询表的属性。

*postgres*=# \d+ tablename

* 查看表结构：

*postgres*=# \d tablename

* 列举schema：

postgres=# \dn

* 查看索引：

postgres=# \di

* 查询表空间：

使用gsql程序的元命令查询表空间。

postgres=# \db

检查pg\_tablespace系统表。如下命令可查到系统和用户定义的全部表空间。

*postgres*=# SELECT spcname FROM pg\_tablespace;

* 查看数据库用户列表：

*postgres*=# SELECT \* FROM pg\_user;

* 要查看用户属性：

*postgres*=# SELECT \* FROM pg\_authid;

* 查看所有角色：

*postgres*=# SELECT \* FROM PG\_ROLES;

## 其他操作

* 查看openGauss支持的所有SQL语句。

postgres=#\h

* 切换数据库：

*postgres*=# \c dbname

* 切换用户：

*postgres*=# \c – username

* 退出数据库：

postgres=# \q