

云数据库 GaussDB

## 工具参考（分布式\_8.x）

文档版本 01  
发布日期 2024-07-01



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

---

# 目录

---

<b>1 gsql.....</b>	<b>1</b>
1.1 gsql 概述.....	1
1.2 使用指导.....	8
1.3 获取帮助.....	10
1.4 命令参考.....	11
1.5 元命令参考.....	16
1.6 常见问题处理.....	31
<b>2 gs_dump.....</b>	<b>36</b>
<b>3 gs_dumpall.....</b>	<b>48</b>
<b>4 gs_restore.....</b>	<b>54</b>

# 1 gsql

gsql是GaussDB提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，gsql还提供了若干[高级特性](#)，便于用户使用。

## 1.1 gsql 概述

### 基本功能

- **连接数据库：**默认只支持从服务器本机连接，如果需要连接到远端的数据库，必须在服务端进行配置。详细操作请参见《开发者指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节中的“远程连接数据库”。

#### 说明

gsql创建连接时，会有5分钟超时时间。如果在这个时间内，数据库未正确地接受连接并对身份进行认证，gsql将超时退出。

针对此问题，可以参考[常见问题处理](#)。

- **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。
- **执行元命令：**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。元命令的详细说明请参见[元命令参考](#)。

### 高级特性

gsql的高级特性如[表1-1](#)所示。

表 1-1 gsql 高级特性

特性名称	描述
变量	<p>gsql提供类似于Linux的shell命令的变量特性，可以使用gsql的元命令\set设置一个变量，格式如下：  <code>\set varname value</code></p> <p>删除由\set命令设置的变量请使用如下方式：  <code>\unset varname</code></p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• 变量只是简单的名称/值对，这里的值可以是任意长度。</li> <li>• 变量名称必须由字母（包括非拉丁字母）、数字和下划线组成，且对大小写敏感。</li> <li>• 如果使用\set varname的格式（不带第二个参数），则只是设置这个变量而没有给变量赋值。</li> <li>• 可以使用不带参数的\set来显示所有变量的值。</li> </ul> <p>变量的示例和详细说明请参见<a href="#">变量</a>。</p>
SQL代换	<p>利用gsql的变量特性，可以将常用的SQL语句设置为变量，以简化操作。</p> <p>SQL代换的示例和详细说明请参见<a href="#">SQL代换</a>。</p>
自定义提示符	<p>gsql使用的提示符支持用户自定义。可以通过修改gsql预留的三个变量PROMPT1、PROMPT2、PROMPT3来改变提示符。</p> <p>这三个变量的值可以由用户自定义，也可以使用gsql预定义的值。详情请参见<a href="#">提示符</a>。</p>
客户端操作历史记录	<p>gsql支持客户端操作历史记录，当客户端连接时指定“-r”参数，此功能被打开。可以通过\set设置记录历史的条数，例如，\set HISTSIZE 50，将记录历史的条数设置为50，\set HISTSIZE 0，不记录历史。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• 客户端操作历史记录条数默认设置为32条，最多支持记录500条。当客户端交互式输入包含中文字符时，只支持UTF-8的编码环境。</li> <li>• 出于安全考虑，将包含PASSWORD、IDENTIFIED、GS_ENCRYPT_AES128、GS_DECRYPT_AES128、GS_ENCRYPT、GS_DECRYPT、GS_ENCRYPT_BYTEA、GS_DECRYPT_BYTEA、PG_CREATE_PHYSICAL_REPLICATION_SLOT_EXTERN、SECRET_ACCESS_KEY、SECRETKEY、CREATE_CREDENTIAL、ACCESSKEY、SECRET_KEY等字符串（不区分大小写）的SQL语句记录识别为包含敏感信息的语句，不会记录到历史信息中，即不能通过上下翻回显。</li> </ul>

- 变量

可以使用gsql元命令\set设置一个变量。例如把变量foo的值设置为bar：

```
gaussdb=# \set foo bar
```

要引用变量的值，在变量前面加冒号。例如查看变量的值：

```
gaussdb=# \echo :foo
bar
```

这种变量的引用方法适用于规则的SQL语句和除\copy、\ef、\help、\sf、\!以外的元命令。

gsql预定义了一些特殊变量，同时也规划了变量的取值。为了保证和后续版本最大限度地兼容，请避免以其他目的使用这些变量。所有特殊变量见[表1-2](#)。

 说明

- 所有特殊变量都由大写字母、数字和下划线组成。
- 要查看特殊变量的默认值，请使用元命令 `\echo :varname` (例如 `\echo :DBNAME`)。

表 1-2 特殊变量设置

变量	设置方法	变量说明
DBNAME	<code>\set DBNAME dbname</code>	当前连接的数据库的名称。每次连接数据库时都会被重新设置。
ECHO	<code>\set ECHO all   queries</code>	<ul style="list-style-type: none"> <li>• 如果设置为all，只显示查询信息。等效于使用gsql连接数据库时指定-a参数。</li> <li>• 如果设置为queries，显示命令行和查询信息。等效于使用gsql连接数据库时指定-e参数。</li> </ul>
ECHO_HIDDEN	<code>\set ECHO_HIDDEN on   off   noexec</code>	<p>当使用元命令查询数据库信息 (例如 <code>\dg</code>) 时，此变量的取值决定了查询的行为：</p> <ul style="list-style-type: none"> <li>• 设置为on，先显示元命令实际调用的查询语句，然后显示查询结果。等效于使用gsql连接数据库时指定-E参数。</li> <li>• 设置为off，则只显示查询结果。</li> <li>• 设置为noexec，则只显示查询信息，不执行查询操作。</li> </ul>
ENCODING	<code>\set ENCODING encoding</code>	当前客户端的字符集编码。
FETCH_COUNT	<code>\set FETCH_COUNT variable</code>	<ul style="list-style-type: none"> <li>• 如果该变量的值为大于0的整数，假设为n，则执行SELECT语句时每次从结果集中取n行到缓存并显示到屏幕。</li> <li>• 如果不设置此变量，或设置的值小于等于0，则执行SELECT语句时一次性把结果都取到缓存。</li> </ul> <p><b>说明</b> 设置合理的变量值，将减少内存使用量。一般来说，设为100到1000之间的值比较合理。</p>
HISTCONTROL	<code>\set HISTCONTROL ignorespace   ignoredups   ignoreboth   none</code>	<ul style="list-style-type: none"> <li>• ignorespace: 以空格开始的行将不会写入历史列表。</li> <li>• ignoredups: 与以前历史记录里匹配的行不会写入历史记录。</li> <li>• ignoreboth、none或者其他值: 所有以交互模式读入的行都被保存到历史列表。</li> </ul> <p><b>说明</b> none表示不设置HISTCONTROL。</p>
HISTFILE	<code>\set HISTFILE filename</code>	此文件用于存储历史名列表。缺省值是 <code>~/.bash_history</code> 。

变量	设置方法	变量说明
HISTSIZE	\set HISTSIZE <i>size</i>	保存在历史命令里命令的个数。缺省值是500。
HOST	\set HOST <i>hostname</i>	已连接的数据库主机名称。
IGNOREEOF	\set IGNOREEOF <i>variable</i>	<ul style="list-style-type: none"> <li>若设置此变量为数值，假设为10，则在gsql中输入的前9次EOF字符（通常是Ctrl+C）都会被忽略，在第10次按Ctrl+C才能退出gsql程序。</li> <li>若设置此变量为非数值，则缺省为10。</li> <li>若删除此变量，则向交互的gsql会话发送一个EOF终止应用。</li> </ul>
LASTOID	\set LASTOID <i>oid</i>	最后影响的oid值，即为从一条INSERT或lo_import命令返回的值。此变量只保证在下一条SQL语句的结果显示之前有效。
ON_ERROR_ROLLBACK	\set ON_ERROR_ROLLBACK on   interactive   off	<ul style="list-style-type: none"> <li>如果是on，当一个事务块里的语句产生错误的时候，这个错误将被忽略而事务继续。</li> <li>如果是interactive，这样的错误只是在交互的会话里忽略。</li> <li>如果是off（缺省），事务块里一个语句生成的错误将会回滚整个事务。on_error_rollback-on模式是通过在一个事务块的每个命令前隐含地发出一个SAVEPOINT的方式来工作的，在发生错误的时候回滚到该事务块。</li> </ul>
ON_ERROR_STOP	\set ON_ERROR_STOP on   off	<ul style="list-style-type: none"> <li>on：命令执行错误时会立即停止，在交互模式下，gsql会立即返回已执行命令的结果。</li> <li>off（缺省）：命令执行错误时将会跳过错误继续执行。</li> </ul>
PORT	\set PORT <i>port</i>	正连接数据库的端口号。
USER	\set USER <i>username</i>	当前用于连接的数据库用户。
VERBOSITY	\set VERBOSITY terse   default   verbose	<p>这个选项可以设置为值terse、default、verbose之一以控制错误报告的冗余行。</p> <ul style="list-style-type: none"> <li>terse：仅返回严重且主要的错误文本以及文本位置（一般适合于单行错误信息）。</li> <li>default：返回严重且主要的错误文本及其位置，还包括详细的错误细节、错误提示（可能会跨越多行）。</li> <li>verbose：返回所有的错误信息。</li> </ul>

- SQL代换

像元命令的参数一样，gsql变量的一个关键特性是可以把gsql变量替换成正规的SQL语句。此外，gsql还提供为变量更换新的别名或其他标识符等功能。使用SQL代换方式替换一个变量的值可在变量前加冒号。例如：

```
gaussdb=# \set foo 'HR.areaS'
gaussdb=# select * from :foo;
 area_id | area_name
-----+-----
      4 | Middle East and Africa
      3 | Asia
      1 | Europe
      2 | Americas
(4 rows)
```

执行以上命令，将会查询HR.areaS表。

### 须知

变量的值是逐字复制的，甚至可以包含不对称的引号或反斜杠命令。所以必须保证输入的内容有意义。

- 提示符

通过表1-3的三个变量可以设置gsql的提示符，这些变量是由字符和特殊的转义字符所组成。

表 1-3 提示符变量

变量	描述	示例
PROMPT1	gsql请求一个新命令时使用的正常提示符。 PROMPT1的默认值为： %/%R%#	使用变量PROMPT1切换提示符： <ul style="list-style-type: none"> <li>提示符变为[local]： gaussdb=&gt; \set PROMPT1 %M [local:/tmp/gaussdba_mppdb]</li> <li>提示符变为name： gaussdb=&gt; \set PROMPT1 name name</li> <li>提示符变为=： gaussdb=&gt; \set PROMPT1 %R =</li> </ul>
PROMPT2	在一个命令输入期待更多输入时（例如，查询没有用一个分号结束或者引号不完整）显示的提示符。	使用变量PROMPT2显示提示符： gaussdb=# \set PROMPT2 TEST gaussdb=# select * from HR.areaS TEST; area_id   area_name -----+----- <ul style="list-style-type: none"> <li>1   Europe</li> <li>2   Americas</li> <li>4   Middle East and Africa</li> <li>3   Asia</li> </ul> (4 rows)



变量	描述	示例
PROMPT3	当执行COPY命令，并期望在终端输入数据时（例如，COPY FROM STDIN），显示提示符。	使用变量PROMPT3显示COPY提示符： gaussdb=# \set PROMPT3 '>>>>' gaussdb=# copy HR.areaS from STDIN; Enter data to be copied followed by a newline. End with a backslash and a period on a line by itself. >>>>1 aa >>>>2 bb >>>>\.

提示符变量的值是按实际字符显示的，但是，当设置提示符的命令中出现“%”时，变量的值根据“%”后的字符，替换为已定义的内容，已定义的提示符请参见表1-4。

表 1-4 已定义的替换

符号	符号说明
%M	主机的全名（包含域名），若连接是通过Unix域套接字进行的，则全名为[local]，若Unix域套接字不是编译的缺省位置，就是[local:/dir/name]。
%m	主机名删去第一个点后面的部分。若通过Unix域套接字连接，则为[local]。
%>	主机正在侦听的端口号。
%n	数据库会话的用户名。
%/	当前数据库名称。
%~	类似 %/，如果数据库是缺省数据库时输出的是波浪线~。
%#	如果会话用户是数据库系统管理员，使用#，否则用>。
%R	<ul style="list-style-type: none"> <li>对于PROMPT1通常是“=”，如果是单行模式则是“^”，如果会话与数据库断开（如果\connect失败可能发生）则是“!”。</li> <li>对于PROMPT2该序列被“-”、“*”、单引号、双引号或“\$”（取决于gsql是否等待更多的输入：查询没有终止、正在一个/* ... */注释里、正在引号或者美元符扩展里）代替。</li> </ul>
%x	事务状态： <ul style="list-style-type: none"> <li>如果不在事务块里，则是一个空字符串。</li> <li>如果在事务块里，则是“*”。</li> <li>如果在一个失败的事务块里则是“!”。</li> <li>如果无法判断事务状态时为“?”（比如没有连接）。</li> </ul>
%digits	指定字节值的字符将被替换到该位置。
%:name	gsql变量“name”的值。

符号	符号说明
%command	command的输出，类似于使用“^”替换。
%[... %]	提示可以包含终端控制字符，这些字符可以改变颜色、背景、提示文本的风格、终端窗口的标题。例如， gaussdb=> \set PROMPT1 '%[%033[1;33;40m%]%n@%/R%[%033[0m%]%' 这个句式的结果是在VT100兼容的可显示彩色的终端上的一个宽体(1;)黑底黄字(33;40)。

## 环境变量

表 1-5 与 gsql 相关的环境变量

名称	描述
COLUMNS	如果\set columns为0，则由此参数控制wrapped格式的宽度。这个宽度用于决定在自动扩展的模式下，是否要把宽输出模式变成竖线的格式。
PAGER	如果查询结果无法在一页显示，它们就会被重定向到这个命令。可以用\pset命令关闭分页器。典型的是用命令more或less来实现逐页查看。缺省值是平台相关的。 <b>说明</b> less的文本显示，受系统环境变量LC_CTYPE影响。
PSQL_EDITOR	\e和\ef命令使用环境变量指定的编辑器。变量是按照列出的先后顺序检查的。在Unix系统上默认的编辑工具是vi。
EDITOR	
VISUAL	
PSQL_EDITOR_LINENUMBER_ARG	当\e和\ef带上一行数字参数使用时，这个变量指定的命令行参数用于向编辑器传递起始行数。像Emacs或vi这样的编辑器，这只是个加号。如果选项和行号之间需要空白，在变量的值后加一个空格。例如： PSQL_EDITOR_LINENUMBER_ARG = '+' PSQL_EDITOR_LINENUMBER_ARG='--line ' Unix系统默认的是+。
PSQLRC	用户的.gsqlrc文件的交互位置。
SHELL	使用!命令跟shell执行的命令是一样的效果。
TMPDIR	存储临时文件的目录。缺省是/tmp。

## 1.2 使用指导

### 前提条件

- 连接数据库时使用的用户需要具备访问数据库的权限。
- gsql须与数据库版本配套。

### 背景信息

使用gsql命令可以连接本机的数据库服务，也可以连接远程数据库服务。连接远程数据库服务时，需要在服务器上设置允许远程连接。详细操作请参见《开发者指南》中“数据库使用入门 > 连接数据库 > 使用gsql连接”章节中的“远程连接数据库”。

### 操作步骤

#### 步骤1 使用gsql连接到GaussDB服务器。

gsql工具使用-d参数指定目标数据库名、-U参数指定数据库用户名、-h参数指定主机名、-p参数指定端口号信息。

#### 说明

若未指定数据库名称，则使用初始化时默认生成的数据库名称；若未指定数据库用户名，则默认使用当前操作系统用户作为数据库用户名；当某个值没有前面的参数（-d、-U等）时，若连接的命令中没有指定数据库名（-d）则该参数会被解释成数据库名；如果已经指定数据库名（-d）而没有指定数据库用户名（-U）时，该参数则会被解释成数据库用户名。

示例1，使用omm用户连接到本机gaussdb数据库的8000端口。

```
gsql -d gaussdb -p 8000
```

示例2，使用jack用户连接到远程主机gaussdb数据库的8000端口。

```
gsql -h 10.180.123.163 -d gaussdb -U jack -p 8000
```

示例3，参数gaussdb和omm不属于任何选项时，分别被解释成了数据库名和用户名。

```
gsql gaussdb omm -p 8000
```

#### 等效于

```
gsql -d gaussdb -U omm -p 8000
```

详细的gsql参数请参见[命令参考](#)。

#### 步骤2 执行SQL语句。

以创建数据库human\_staff为例。

```
CREATE DATABASE human_staff;  
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。

#### 步骤3 执行gsql元命令。

以列出GaussDB中所有的数据库和描述信息为例。

```
gaussdb=# \l
          List of databases
-----+-----+-----+-----+-----+-----+
 Name      | Owner  | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----+
 human_resource | omm   | SQL_ASCII | C       | C      |
 postgres      | omm   | SQL_ASCII | C       | C      |
 template0     | omm   | SQL_ASCII | C       | C      | =c/omm      +
              |       |          |         |        | omm=CTc/omm
 template1     | omm   | SQL_ASCII | C       | C      | =c/omm      +
              |       |          |         |        | omm=CTc/omm
 human_staff   | omm   | SQL_ASCII | C       | C      |
(5 rows)
```

更多gsql元命令请参见[元命令参考](#)。

----结束

## 示例

以把一个查询分成多行输入为例。注意提示符的变化：

```
gaussdb=# CREATE TABLE HR.areaS(
gaussdb(# area_ID  NUMBER,
gaussdb(# area_NAME VARCHAR2(25)
gaussdb(# )tablespace EXAMPLE;
CREATE TABLE
```

查看表的定义：

```
gaussdb=# \d HR.areaS
          Table "hr.areas"
-----+-----+-----+
 Column |      Type      | Modifiers
-----+-----+-----+
 area_id | numeric        | not null
 area_name | character varying(25) |
```

向HR.areaS表插入四行数据：

```
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (1, 'Europe');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (2, 'Americas');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (3, 'Asia');
INSERT 0 1
gaussdb=# INSERT INTO HR.areaS (area_ID, area_NAME) VALUES (4, 'Middle East and Africa');
INSERT 0 1
```

切换提示符：

```
gaussdb=# \set PROMPT1 '%n@m %~%R%#'
omm@[local] gaussdb=#
```

查看表：

```
omm@[local] gaussdb=# SELECT * FROM HR.areaS;
 area_id | area_name
-----+-----
      1 | Europe
      4 | Middle East and Africa
      2 | Americas
      3 | Asia
(4 rows)
```

可以用\pset命令以不同的方法显示表：

```
omm@[local] gaussdb=# \pset border 2
Border style is 2.
omm@[local] gaussdb=# SELECT * FROM HR.areaS;
+-----+-----+
 area_id | area_name
```

```
| area_id | area_name |
+-----+-----+
| 1 | Europe |
| 2 | Americas |
| 3 | Asia |
| 4 | Middle East and Africa |
+-----+-----+
(4 rows)
omm@[local] gaussdb=# \pset border 0
Border style is 0.
omm@[local] gaussdb=# SELECT * FROM HR.areaS;
area_id area_name
-----
1 Europe
2 Americas
3 Asia
4 Middle East and Africa
(4 rows)
```

#### 使用元命令：

```
omm@[local] gaussdb=# \a \t \x
Output format is unaligned.
Showing only tuples.
Expanded display is on.
omm@[local] gaussdb=# SELECT * FROM HR.areaS;
area_id|2
area_name|Americas

area_id|1
area_name|Europe

area_id|4
area_name|Middle East and Africa

area_id|3
area_name|Asia
omm@[local] gaussdb=#
```

## 1.3 获取帮助

### 操作步骤

- 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: "omm")
-f, --file=FILENAME    execute commands from file, then exit
.....
```

- 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
```

## 任务示例

**步骤1** 使用如下命令连接数据库。

```
gsql -d gaussdb -p 8000
```

gaussdb为需要连接的数据库名称，8000为CN的端口号。

连接成功后，系统显示类似如下信息：

```
gsql ((GaussDB Kernel XXX.X.XXX build f521c606) compiled at 2021-09-16 14:55:22 commit 2935 last mr
6385 release)
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

**步骤2** 查看gsql的帮助信息。具体执行命令请参见[表1-6](#)。

**表 1-6** 使用 gsql 联机帮助

描述	示例
查看版权信息	<code>\copyright</code>
查看GaussDB支持的SQL语句的帮助	<p>查看GaussDB支持的SQL语句的帮助</p> <p>例如，查看GaussDB支持的所有SQL语句：</p> <pre>gaussdb=# \h Available help: ABORT ALTER AGGREGATE ALTER APP WORKLOAD GROUP ... ..</pre> <p>例如，查看CREATE DATABASE命令的参数可使用下面的命令：</p> <pre>gaussdb=# \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 ]}{...} ;</pre>
查看gsql命令的帮助	<p>例如，查看gsql支持的命令：</p> <pre>gaussdb=# \? General \copyright          show GaussDB Kernel 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 ... ..</pre>

----结束

## 1.4 命令参考

详细的gsql参数请参见[表1-7](#)、[表1-8](#)、[表1-9](#)和[表1-10](#)。

表 1-7 常用参数

参数	参数说明	取值范围
-c, -- command=CO MMAND	声明gsql要执行一条字符串命令然后退出。	-
-d, -- dbname=DBNA ME	指定想要连接的数据库名称。 另外, gsql允许使用扩展的DBNAME, 即 'postgres[ql]://[user[:password]@][netloc] [:port][,...]/[dbname][? param1=value1&...]'或'[key=value] [...]'形 式的连接串作为DBNAME, gsql将从连接串 中解析连接信息, 并优先使用这些信息。 <b>注意</b> gsql使用扩展的DBNAME创建连接时, 不支持指 定replication参数。	字符串。
-f, -- file=FILENAME	使用文件作为命令源而不是交互式输入。 gsql将在处理完文件后结束。如果 FILENAME是- (连字符), 则从标准输入读 取。	绝对路径或相对路 径, 且满足操作系 统路径命名规则。
-l, --list	列出所有可用的数据库, 然后退出。	-
-v, --set, -- variable=NAME =VALUE	设置gsql变量NAME为VALUE。 变量的示例和详细说明请参见 <a href="#">变量</a> 。	-
-X, --no-gsqlrc	不读取启动文件 (系统范围的gsqlrc或者用 户的~/.gsqlrc都不读取)。 <b>说明</b> 启动文件默认为~/.gsqlrc, 或通过PSQLRC环境 变量指定。	-
-1 ("one"), -- single- transaction	当gsql使用-f选项执行脚本时, 会在脚本的 开头和结尾分别加上START TRANSACTION/COMMIT用以把整个脚本 当作一个事务执行。这将保证该脚本完全执 行成功, 或者脚本无效。 <b>说明</b> 如果脚本中已经使用了START TRANSACTION, COMMIT, ROLLBACK, 则该选项无效。	-
-?, --help	显示关于gsql命令行参数的帮助信息然后退 出。	-
-V, --version	打印gsql版本信息然后退出。	-

表 1-8 输入和输出参数

参数	参数说明	取值范围
-a, --echo-all	在读取行时向标准输出打印所有内容。 <b>注意</b> 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-e, --echo-queries	把所有发送给服务器的查询同时回显到标准输出。 <b>注意</b> 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-E, --echo-hidden	回显由\d和其他反斜杠命令生成的实际查询。	-
-k, --with-key=KEY	使用gsql对导入的加密文件进行解密。 <b>须知</b> <ul style="list-style-type: none"> <li>对于本身就是shell命令中的关键字如单引号 (') 或双引号 (")，Linux shell会检测输入的单引号 (') 或双引号 (") 是否匹配。如果不匹配，shell认为用户没有输入完毕，会一直等待用户输入，从而不会进入到gsql程序。</li> <li>不支持解密导入存储过程和函数。</li> </ul>	-
-L, --logfile=FILENAME	除了正常的输出源之外，把所有查询输出记录到文件FILENAME中。 <b>注意</b> <ul style="list-style-type: none"> <li>使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。</li> <li>此参数只保留查询结果到相应文件中，主要目标是为了查询结果能够更好更准确地被其他调用者（例如自动化运维脚本）解析；而不是保留gsql运行过程中的相关日志信息。</li> </ul>	绝对路径或相对路径，且满足操作系统路径命名规则。
-m, --maintenance	允许在两阶段事务恢复期间连接集群。 <b>说明</b> 该选项是一个开发选项，禁止用户使用，只限专业技术人员使用，功能是：使用该选项时，gsql可以连接到备机，用于校验主备机数据的一致性。	-
-n, --no-libedit	关闭命令行编辑。	-
-o, --output=FILENAME	将所有查询输出重定向到文件FILENAME。	绝对路径或相对路径，且满足操作系统路径命名规则。
-q, --quiet	安静模式，执行时不会打印出额外信息。	缺省时gsql将打印许多其他输出信息。



参数	参数说明	取值范围
-s, --single-step	单步模式运行。意味着每个查询在发往服务器之前都要提示用户，用这个选项也可以取消执行。此选项主要用于调试脚本。 <b>注意</b> 使用此参数可能会暴露部分SQL语句中的敏感信息，如创建用户语句中的password信息等，请谨慎使用。	-
-S, --single-line	单行运行模式，这时每个命令都将由换行符结束，像分号那样。	-
-C, -C1, --enable-client-encryption=1	当使用-C参数连接本地数据库或者连接远程数据库时，可通过该选项打开密态数据库开关，此开关为密态等值查询基本能力开关。	-
-C3, --enable-client-encryption=3	当使用-C参数连接本地数据库或者连接远程数据库时，可通过该选项打开内存解密逃生通道开关，支持密态等值查询基本能力以及内存解密逃生通道能力。	-

表 1-9 输出格式参数

参数	参数说明	取值范围
-A, --no-align	切换为非对齐输出模式。	缺省为对齐输出模式。
-F, --field-separator=S TRING	设置域分隔符（默认为“ ”）。	-
-H, --html	打开HTML格式输出。	-
-P, --pset=VAR[=ARG]	在命令行上以\pset的风格设置打印选项。 <b>说明</b> 这里必须用等号而不是空格分隔名称和值。例如，把输出格式设置为LaTeX，可以键入-P format=latex	-
-R, --record-separator=S TRING	设置记录分隔符。	-
-r	开启在客户端操作中可以编辑的模式。	缺省为关闭。
-t, --tuples-only	只打印行。	-

参数	参数说明	取值范围
-T, --table-attr=TEXT	允许声明放在HTML table标签里的选项。 使用时请搭配参数“-H,--html”，指定为HTML格式输出。	-
-x, --expanded	打开扩展表格式模式。	-
-z, --field-separator-zero	设置非对齐输出模式的域分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-
-0, --record-separator-zero	设置非对齐输出模式的记录分隔符为空。 使用时请搭配参数“-A, --no-align”，指定为非对齐输出模式。	-
-2, --pipeline	使用管道传输密码，禁止在终端使用，必须和-c或者-f参数一起使用。	-

表 1-10 连接参数

参数	参数说明	取值范围
-h, --host=HOSTNAME	<p>指定正在运行服务器的主机名、Unix域套接字的路径、或者域名。接受以“,”分割的字符串来指定多个主机地址，支持指定多个主机地址，支持指定IPv6主机地址。</p> <p>当指定多个主机地址时，默认情况下会跳过故障的节点，自动选择第一个可用的主节点进行连接。可通过设置PGTARGETSESSIONATTRS环境变量的值来选择连接到不同类型的节点，变量与节点类型对应的关系如下：</p> <p><b>PGTARGETSESSIONATTRS环境变量的值--选择连接的节点类型</b></p> <p>read-write--可读写的节点 read-only--只读节点 primary或者不设定--主节点 standby--备节点 prefer-standby--首选备节点，没有备节点则转为any any--不进行角色检查</p> <p><b>说明</b> 主CN节点和主DN节点都是主节点，一般情况下DN节点都是只读节点，可以通过设置PGTARGETSESSIONATTRS为read-write来选择连接到CN节点。</p>	<p>如果省略主机名，gsql将通过Unix域套接字与本地主机的服务器相联，或者在没有Unix域套接字的机器上，通过TCP/IP与localhost连接。</p>

参数	参数说明	取值范围
-p, --port=PORT	指定数据库服务器的端口号。可以配置一个或多个，当配置一个时，所有的主机地址都使用同一个端口连接；当配置多个时，顺序与主机地址顺序相同，个数必须与主机地址数相等，当不相等时会报错。可以通过port参数修改默认端口号。	默认端口可通过编译参数来指定，不指定的话默认为5432。
-U, --username=USE RNAME	指定连接数据库的用户。 <b>说明</b> <ul style="list-style-type: none"> <li>通过该参数指定用户连接数据库时，需要同时提供用户密码用以身份验证。您可以通过交换方式输入密码，或者通过-W参数指定密码。</li> <li>用户名中包含有字符\$，需要在字符\$前增加转义字符才可成功连接数据库。</li> </ul>	字符串，默认使用与当前操作系统用户同名的用户。
-W, --password=PASS WORD	当使用-U参数连接本地数据库或者连接远端数据库时，可通过该选项指定密码。 <b>说明</b> <ul style="list-style-type: none"> <li>登录CN所在服务器后连接本地CN实例时，默认使用trust连接，会忽略此参数。</li> <li>用户密码中包含特殊字符“\”和“”时，需要增加转义字符才可成功连接数据库。</li> <li>如果用户未输入该参数，但是数据库连接需要用户密码，这时将出现交互式输入，请用户输入当前连接的密码。该密码最长长度为999字节，受限与GUC参数password_max_length的最大值。</li> </ul>	字符串。

## 1.5 元命令参考

介绍使用GaussDB数据库命令行交互工具登录数据库后，gsql所提供的元命令。所谓元命令就是在gsql里输入的任何以不带引号的反斜杠开头的命令。

### 注意事项

- 一个gsql元命令的格式是反斜杠后面紧跟一个动词，然后是任意参数。参数命令动词和其他参数以任意个空白字符间隔。
- 要在参数里面包含空白，必须用单引号把它引起来。要在这样的参数里包含单引号，可以在前面加一个反斜杠。任何包含在单引号里的内容都会被进一步进行类似C语言的替换：\n（新行）、\t（制表符）、\b（退格）、\r（回车）、\f（换页）、\digits（八进制表示的字符）、\xdigits（十六进制表示的字符）。
- 用""包围的内容被当做一个命令行传入shell。该命令的输出（删除了结尾的新行）被当做参数值。
- 如果不带引号的参数以冒号(:)开头，它会被当做一个gsql变量，并且该变量的值最终会成为真正的参数值。
- 有些命令以一个SQL标识的名称（比如一个表）为参数。这些参数遵循SQL语法关于双引号的规则：不带双引号的标识强制转换成小写，而双引号保护字母不进行

大小写转换，并且允许在标识符中使用空白。在双引号中，成对的双引号在结果名称中分析成一个双引号。比如，FOO"BAR"BAZ解析成fooBARbaz；而"Aweird""name"解析成A weird"name。

- 对参数的分析在遇到另一个不带引号的反斜杠时停止。这里会认为是一个新的元命令的开始。特殊的双反斜杠序列 (\\) 标识参数的结尾并将继续分析后面的SQL语句（如果存在）。这样SQL和gsql命令可以自由的在一行里面混合。但是在任何情况下，一条元命令的参数不能延续超过行尾。

## 元命令

元命令的详细说明请参见[表1-11](#)、[表1-12](#)、[表1-13](#)、[表1-14](#)、[表1-16](#)、[表1-18](#)、[表1-19](#)、[表1-20](#)、[表1-22](#)和[表1-23](#)。

### 须知

以下命令中所提到的FILE代表文件路径。此路径可以是绝对路径（如/home/gauss/file.txt），也可以是相对路径（file.txt，file.txt会默认在用户执行gsql命令所在的路径下创建）。

表 1-11 一般的元命令

参数	参数说明	取值范围
\copyright	显示GaussDB的版本和版权信息。	-
\g [FILE] or ;	执行查询（并将结果发送到文件或管道）。	-
\h(\help) [NAME]	给出指定SQL语句的语法帮助。	如果没有给出NAME，gsql将列出可获得帮助的所有命令。如果NAME是一个星号（*），则显示所有SQL语句的语法帮助。

参数	参数说明	取值范围
\parallel [on [num]] off]	<p>控制并发执行开关。</p> <ul style="list-style-type: none"> <li>on: 打开控制并发执行开关, 且最大并发数为num。</li> <li>off: 关闭控制并发执行开关。</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>不支持事务中开启并发执行以及并发中开启事务。</li> <li>不支持\d这类元命令的并发。</li> <li>并发select返回结果混乱问题, 此为客户可接受, core、进程停止响应不可接受。</li> <li>不推荐在并发中使用set语句, 否则导致结果与预期不一致。</li> <li>不支持创建临时表! 如需使用临时表, 需要在开启parallel之前创建好, 并在parallel内部使用。parallel内部不允许创建临时表。</li> <li>\parallel执行时最多会启动num个独立的gsql进程连接服务器。</li> <li>\parallel中所有作业的持续时间不能超过session_timeout, 否则可能会导致并发执行过程中断连。</li> </ul>	<p>num的默认值: 1024。</p> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>服务器能接受的最大连接数受max_connection及当前已有连接数限制。</li> <li>设置num时请考虑服务器当前可接受的实际连接数合理指定。</li> </ul>
\q	退出gsql程序。在一个脚本文件里, 只在脚本终止的时候执行。	-

表 1-12 查询缓存区元命令

参数	参数说明
\e [FILE] [LINE]	使用外部编辑器编辑查询缓冲区 (或者文件)。
\ef [FUNCNAME [LINE]]	使用外部编辑器编辑函数定义。如果指定了LINE (即行号), 则光标会指到函数体的指定行。
\p	打印当前查询缓冲区到标准输出。
\r	重置 (或清空) 查询缓冲区。
\w FILE	将当前查询缓冲区输出到文件。

表 1-13 输入/输出元命令

参数	参数说明
<code>\copy { table [ ( column_list ) ]   ( query ) } { from   to } { filename   stdin   stdout   pstdin   pstdout } [LOAD] [LOAD_DISCARD 'string'] [ with ] [ binary ] [ oids ] [ delimiter [ as ] 'character' ] [ null [ as ] 'string' ] [ useeof ] [ csv [ header ] [ quote [ as ] 'character' ] [ escape [ as ] 'character' ] [ force quote column_list   * ] [ force not null column_list ] ] [parallel integer]</code>	<p>在任何gsql客户端登录数据库成功后可以执行导入导出数据，这是一个运行SQL COPY命令的操作，但不是读取或写入指定文件的服务器，而是读取或写入文件，并在服务器和本地文件系统之间路由数据。这意味着文件的可访问性和权限是本地用户的权限，而不是服务器的权限，并且不需要数据库初始化用户权限。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>\COPY只适合小批量，格式良好的数据导入，导入数据应优先选择GDS或COPY。</li> <li>\COPY 可以指定数据导入时的客户端数量，从而实现数据文件的并行导入，目前并发数范围为[1, 8]。</li> <li>\COPY并行导入目前存在以下约束：临时表的并行导入不支持、在事务内的并行导入不支持、对二进制文件的并行导入不支持、数据导入支持AES128加密时不支持以及COPY选项中存在EOL。在这些情况下，即使指定了parallel参数，仍然会走非并行流程。</li> <li>\COPY的text格式和csv格式均支持header功能。</li> <li>其中LOAD功能为gs_loader进行语法转换后调用copy的标识，非主动调用功能。</li> <li>其中LOAD_DISCARD功能为gs_loader解析后discard文件路径，非主动调用功能。</li> </ul>
<code>\echo [STRING]</code>	把字符串写到标准输出。
<code>\i FILE</code>	从文件FILE中读取内容，并将其当作输入，执行查询。
<code>\i+ FILE KEY</code>	执行加密文件中的命令。
<code>\ir FILE</code>	和\i类似，只是相对于存放当前脚本的路径。
<code>\ir+ FILE KEY</code>	和\i+类似，只是相对于存放当前脚本的路径。
<code>\o [FILE]</code>	把所有的查询结果发送到文件里。
<code>\qecho [STRING]</code>	把字符串写到查询结果输出流里。

 说明

表1-14中的选项S表示显示系统对象，+表示显示对象附加的描述信息。PATTERN用来指定要被显示的对象名称。

表 1-14 显示信息元命令

参数	参数说明	取值范围	示例
\d[S+]	列出当前search_path中模式下所有的表、视图和序列。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	-	列出当前search_path中模式下所有的表、视图和序列。 gaussdb=# \d
\d[S+] NAME	列出指定表、视图和索引的结构。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	-	假设存在表a, 列出指定表a的结构。 gaussdb=# \dtable+ a
\d+ [PATTERN]	列出所有表、视图和索引。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的表、视图和索引。	列出所有名称以f开头的表、视图和索引。 gaussdb=# \d+ f*
\da[S] [PATTERN]	列出所有可用的聚集函数, 以及它们操作的数据类型和返回值类型。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的聚集函数。	列出所有名称以f开头可用的聚集函数, 以及它们操作的数据类型和返回值类型。 gaussdb=# \da f*
\db[+] [PATTERN]	列出所有可用的表空间。	如果声明了PATTERN, 只显示名称匹配PATTERN的表空间。	列出所有名称以p开头的可用表空间。 gaussdb=# \db p*
\dc[S+] [PATTERN]	列出所有字符集之间的可用转换。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的转换。	列出所有字符集之间的可用转换。 gaussdb=# \dc *
\dC[+] [PATTERN]	列出所有类型转换。PATTERN需要使用实际类型名, 不能使用别名。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的转换。	列出所有名称以c开头的类型转换。 gaussdb=# \dC c*

参数	参数说明	取值范围	示例
\dd[S] [PATTERN]	显示所有匹配PATTERN的描述。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	如果没有给出参数，则显示所有可视对象。“对象”包括：聚集、函数、操作符、类型、关系(表、视图、索引、序列、大对象)、规则。	列出所有可视对象。 gaussdb=# \dd
\ddp [PATTERN]	显示所有默认的使用权限。	如果指定了PATTERN，只显示名称匹配PATTERN的使用权限。	列出所有默认的使用权限。 gaussdb=# \ddp
\dD[S+] [PATTERN]	列出所有可用域。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN，只显示名称匹配PATTERN的域。	列出所有可用域。 gaussdb=# \dD
\det[+] [PATTERN]	列出所有的外部表。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN，只显示名称匹配PATTERN的表。	列出所有的外部表。 gaussdb=# \det
\des[+] [PATTERN]	列出所有的外部服务器。	如果声明了PATTERN，只显示名称匹配PATTERN的服务器。	列出所有的外部服务器。 gaussdb=# \des
\deu[+] [PATTERN]	列出用户映射信息。	如果声明了PATTERN，只显示名称匹配PATTERN的信息。	列出用户映射信息。 gaussdb=# \deu
\dew[+] [PATTERN]	列出封装的外部数据。	如果声明了PATTERN，只显示名称匹配PATTERN的数据。	列出封装的外部数据。 gaussdb=# \dew
\df[antw][S+] [PATTERN]	列出所有可用函数，以及它们的参数和返回的数据类型。a代表聚集函数，n代表普通函数，t代表触发器，w代表窗口函数。当search_path中不同模式存在同名对象时，只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN，只显示名称匹配PATTERN的函数。	列出所有可用函数，以及它们的参数和返回的数据类型。 gaussdb=# \df



参数	参数说明	取值范围	示例
\dF[+] [PATTERN]	列出所有的文本搜索配置信息。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的配置信息。	列出所有的文本搜索配置信息。 gaussdb=# \dF+
\dFd[+] [PATTERN]	列出所有的文本搜索字典。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的字典。	列出所有的文本搜索字典。 gaussdb=# \dFd
\dFp[+] [PATTERN]	列出所有的文本搜索分析器。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的分析器。	列出所有的文本搜索分析器。 gaussdb=# \dFp
\dFt[+] [PATTERN]	列出所有的文本搜索模板。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只显示名称匹配PATTERN的模板。	列出所有的文本搜索模板。 gaussdb=# \dFt
\dg[+] [PATTERN]	列出所有数据库角色。 <b>说明</b> 因为用户和群组的概念被统一为角色, 所以这个命令等价于\du。为了和以前兼容, 所以保留两个命令。	如果指定了PATTERN, 只显示名称匹配PATTERN的角色。	列出名称为“j?e”所有数据库角色 (“?”表示任一字符)。 gaussdb=# \dg j?e
\dl	\lo_list的别名, 显示一个大对象的列表。	-	列出所有的大对象。 gaussdb=# \dl
\dL[S+] [PATTERN]	列出可用的程序语言。	如果指定了PATTERN, 只列出名称匹配PATTERN的语言。	列出可用的程序语言。 gaussdb=# \dL
\dm[S+] [PATTERN]	列出物化视图。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果指定了PATTERN, 只列出名称匹配PATTERN的物化视图。	列出物化视图。 gaussdb=# \dm
\dn[S+] [PATTERN]	列出所有模式 (名称空间)。如果向命令追加+, 会列出每个模式相关的权限及描述。	如果声明了PATTERN, 只列出名称匹配PATTERN的模式名。缺省时, 只列出用户创建的模式。	列出所有名称以d开头的模式以及相关信息。 gaussdb=# \dn+ d*

参数	参数说明	取值范围	示例
\do[S] [PATTERN]	列出所有可用的操作符, 以及它们的操作数和返回的数据类型。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只列出名称匹配PATTERN的操作符。缺省时, 只列出用户创建的操作符。	列出所有可用的操作符, 以及它们的操作数和返回的数据类型。 gaussdb=# \do
\dO[S+] [PATTERN]	列出排序规则。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果声明了PATTERN, 只列出名称匹配PATTERN的规则。缺省时, 只列出用户创建的规则。	列出排序规则。 gaussdb=# \dO
\dp [PATTERN]	列出一列可用的表、视图以及相关的权限信息。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。  \dp显示结果如下: rolename=xxxx/yyyy --赋予一个角色的权限 =xxxx/yyyy --赋予public的权限  xxxx表示赋予的权限, yyyy表示授予这个权限的角色。权限的参数说明请参见表1-15。	如果指定了PATTERN, 只列出名称匹配PATTERN的表、视图。	列出一列可用的表、视图以及相关的权限信息。 gaussdb=# \dp
\drds [PATTERN1 [PATTERN2]]	列出所有修改过的配置参数。这些设置可以是针对角色的、针对数据库的或者同时针对两者的。PATTERN1和PATTERN2表示要列出的角色PATTERN和数据库PATTERN。	如果声明了PATTERN, 只列出名称匹配PATTERN的规则。缺省或指定*时, 则会列出所有设置。	列出数据库所有修改过的配置参数。 gaussdb=# \drds * dbname
\dT[S+] [PATTERN]	列出所有的数据类型。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。	如果指定了PATTERN, 只列出名称匹配PATTERN的类型。	列出所有的数据类型。 gaussdb=# \dT
\du[+] [PATTERN]	列出所有数据库角色。 <b>说明</b> 因为用户和群组的概念被统一为角色, 所以这个命令等价于\dg。为了和以前兼容, 所以保留两个命令。	如果指定了PATTERN, 则只列出名称匹配PATTERN的角色。	列出所有数据库角色。 gaussdb=# \du

参数	参数说明	取值范围	示例
\dE[S+] [PATTERN] \di[S+] [PATTERN] \ds[S+] [PATTERN] \dt[S+] [PATTERN] \dv[S+] [PATTERN]	这一组命令, 字母E, i, s, t和v分别代表着外部表, 索引, 序列, 表和视图。可以以任意顺序指定其中一个或者它们的组合来列出这些对象。当search_path中不同模式存在同名对象时, 只显示search_path中位置靠前模式下的同名对象。例如: \dit列出所有的索引和表。在命令名称后面追加+, 则每一个对象的物理尺寸以及相关的描述也会被列出。	如果指定了PATTERN, 只列出名称匹配该PATTERN的对象。默认情况下只会显示用户创建的对象。通过PATTERN或者S修饰符可以把系统对象包括在内。	列出所有的索引和视图。 gaussdb=# \div
\dx[+] [PATTERN]	列出安装数据库的扩展信息。	如果指定了PATTERN, 则只列出名称匹配PATTERN的扩展信息。	列出安装数据库的扩展信息。 gaussdb=# \dx
\l[+]	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。	-	列出服务器上所有数据库的名称、所有者、字符集编码以及使用权限。 gaussdb=# \l
\sf[+] FUNCNAME	显示函数的定义。 <b>说明</b> 对于带圆括号的函数名, 需要在函数名两端添加双引号, 并且在双引号后面加上参数类型列表。参数类型列表两端添加圆括号。如果存在同名的函数, 则会返回多个函数的定义。	-	假设存在函数function_a和函数名带圆括号的函数func()name, 列出函数的定义。 gaussdb=# \sf function_a gaussdb=# \sf "func()name"(argtype1, argtype2)
\z [PATTERN]	列出数据库中所有表、视图和序列, 以及它们相关的访问特权。	如果给出任何pattern, 则被当成一个正则表达式, 只显示匹配的表、视图、序列。	列出数据库中所有表、视图和序列, 以及它们相关的访问特权。 gaussdb=# \z

表 1-15 权限的参数说明

参数	参数说明
r	SELECT: 允许对指定的表、视图读取数据。

参数	参数说明
w	UPDATE: 允许对指定表更新字段。
a	INSERT: 允许对指定表插入数据。
d	DELETE: 允许删除指定表中的数据。
D	TRUNCATE: 允许清理指定表中的数据。
x	REFERENCES: 允许创建外键约束, 分布式场景暂不支持。
t	TRIGGER: 允许在指定表上创建触发器。
X	EXECUTE: 允许使用指定的函数, 以及利用这些函数实现的操作符。
U	USAGE: <ul style="list-style-type: none"> <li>对于过程语言, 允许用户在创建函数时, 指定过程语言。</li> <li>对于模式, 允许访问包含在指定模式中的对象。</li> <li>对于序列, 允许使用nextval函数。</li> </ul>
C	CREATE: <ul style="list-style-type: none"> <li>对于数据库, 允许在该数据库里创建新的模式。</li> <li>对于模式, 允许在该模式中创建新的对象。</li> <li>对于表空间, 允许在其中创建表, 以及允许创建数据库和模式的时候把该表空间指定为其缺省表空间。</li> </ul>
c	CONNECT: 允许用户连接到指定的数据库。
T	TEMPORARY: 允许创建临时表。
A	ALTER: 允许用户修改指定对象的属性。
P	DROP: 允许用户删除指定的对象。
m	COMMENT: 允许用户定义或修改指定对象的注释。
i	INDEX: 允许用户在指定表上创建索引。
v	VACUUM: 允许用户对指定的表执行ANALYZE和VACUUM操作。
*	给前面权限的授权选项。

表 1-16 格式化元命令

参数	参数说明
\a	对齐模式和非对齐模式之间的切换。
\C [STRING]	把正在打印的表的标题设置为一个查询的结果或者取消这样的设置。
\f [STRING]	对于不对齐的查询输出, 显示或者设置域分隔符。

参数	参数说明
\H	<ul style="list-style-type: none"> <li>若当前模式为文本格式，则切换为HTML输出格式。</li> <li>若当前模式为HTML格式，则切换回文本格式。</li> </ul>
\pset NAME [VALUE]	设置影响查询结果表输出的选项。NAME的取值见表 1-17。
\t [on off]	切换输出的字段名的信息和行计数脚注。
\T [STRING]	指定在使用HTML输出格式时放在table标签里的属性。如果参数为空，不设置。
\x [on off auto]	切换扩展行格式。

表 1-17 可调节的打印选项

选项	选项说明	取值范围
border	value必须是一个数字。通常这个数字越大，表的边界就越宽线就越多，但是这个取决于特定的格式。	<ul style="list-style-type: none"> <li>在HTML格式下，取值范围为大于0的整数。</li> <li>在其他格式下，取值范围： <ul style="list-style-type: none"> <li>- 0: 无边框</li> <li>- 1: 内部分隔线</li> <li>- 2: 台架</li> </ul> </li> </ul>
expanded (或x)	在正常和扩展格式之间切换。	<ul style="list-style-type: none"> <li>当打开扩展格式时，查询结果用两列显示，字段名称在左、数据在右。这个模式在数据无法放进通常的“水平”模式的屏幕时很有用。</li> <li>在正常格式下，当查询输出的格式比屏幕宽时，用扩展格式。正常格式只对aligned和wrapped格式有用。</li> </ul>
fieldsep	声明域分隔符来实现非对齐输出。这样就可以创建其他程序希望的制表符或逗号分隔的输出。要设置制表符域分隔符，键入\pset fieldsep '\t'。缺省域分隔符是' '(竖条符)。	-
fieldsep_z ero	声明域分隔符来实现非对齐输出到零字节。	-
footer	用来切换脚注。	-

选项	选项说明	取值范围
format	设置输出格式。允许使用唯一缩写（这意味着一个字母就够了）。	取值范围： <ul style="list-style-type: none"> <li>unaligned：写一行的所有列在一条直线上中，当前活动字段分隔符分隔。</li> <li>aligned：此格式是标准的，可读性好的文本输出。</li> <li>wrapped：类似aligned，但是包装跨行的宽数据值，使其适应目标字段的宽度输出。</li> <li>html：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。</li> <li>latex：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。</li> <li>troff-ms：把表输出为可用于文档里的对应标记语言。输出不是完整的文档。</li> </ul>
null	打印一个字符串，用来代替一个null值。	缺省是什么都不打印，这样很容易和空字符串混淆。
numericlocale	切换分隔小数点左边的数值的区域相关的分组符号。	<ul style="list-style-type: none"> <li>on：显示指定的分隔符。</li> <li>off：不显示分隔符。</li> </ul> 忽略此参数，显示默认的分隔符。
pager	控制查询和gsql帮助输出的分页器。如果设置了环境变量PAGER，输出将被指向到指定程序，否则使用系统缺省。	<ul style="list-style-type: none"> <li>on：当输出到终端且不适合屏幕显示时，使用分页器。</li> <li>off：不使用分页器。</li> <li>always：当输出到终端无论是否符合屏幕显示时，都使用分页器。</li> </ul>
recordsep	声明在非对齐输出格式时的记录分隔符。	-
recordsep_zero	声明在非对齐输出到零字节时的记录分隔符。	-
tableattr (或T)	声明放在html输出格式中HTML table标签的属性（例如：cellpadding或bgcolor）。注意：这里可能不需要声明border，因为已经在\pset border里用过了。如果没有给出value，则不设置表的属性。	-

选项	选项说明	取值范围
title	为随后打印的表设置标题。这个可以用于给输出一个描述性标签。如果没有给出value，不设置标题。	-
tuples_only (或者 t)	在完全显示和只显示实际的表数据之间切换。完全显示将输出像列头、标题、各种脚注等信息。在 tuples_only 模式下，只显示实际的表数据。	-
feedback	切换是否输出结果行数。	-

表 1-18 连接元命令

参数	参数说明	取值范围
\c[onnect] [DBNAME]-USER -HOST -PORT -]	<p>连接到一个新的数据库。当数据库名称长度超过63个字节时，默认前63个字节有效，连接到前63个字节对应的数据库，但是gsql的命令提示符中显示的数据库对象名仍为截断前的名称。</p> <p><b>说明</b> 重新建立连接时，如果切换数据库登录用户，将可能会出现交互式输入，要求输入新用户的连接密码。该密码最长长度为999字节，受限与GUC参数password_max_length的最大值。</p>	-
\encoding [ENCODING]	设置客户端字符编码格式。	不带参数时，显示当前的编码格式。
\conninfo	输出当前连接的数据库的信息。	-

表 1-19 操作系统元命令

参数	参数说明	取值范围
\cd [DIR]	切换当前的工作目录。	绝对路径或相对路径，且满足操作系统路径命名规则。
\setenv NAME [VALUE]	设置环境变量NAME为VALUE，如果没有给出VALUE值，则不设置环境变量。	-
\timing [on off]	以毫秒为单位显示每条SQL语句的执行时间（不包括屏显打印时间）。	<ul style="list-style-type: none"> <li>on表示打开显示。</li> <li>off表示关闭显示。</li> </ul>
\! [COMMAND]	返回到一个单独的Unix shell或者执行Unix命令COMMAND。	-

表 1-20 变量元命令

参数	参数说明
\prompt [TEXT] NAME	提示用户用文本格式来指定变量名称。
\set [NAME [VALUE]]	<p>设置内部变量NAME为VALUE或者如果给出了多于一个值，设置为所有这些值的连接结果。如果没有给出第二个参数，只设变量不设值。</p> <p>有一些常用变量被gsql特殊对待，它们是一些选项设置，通常所有特殊对待的变量都是由大写字母组成(可能还有数字和下划线)。表1-21是一个所有特殊对待的变量列表。</p>
\unset NAME	不设置 (或删除) gsql变量名。

表 1-21 \set 常用命令

名称	命令说明	取值范围
\set VERBOSITY value	这个选项可以设置为值default, verbose, terse之一以控制错误报告的冗余行。	value取值范围: default, verbose, terse
\set ON_ERROR_STOP value	如果设置了这个变量，脚本处理将马上停止。如果该脚本是从另外一个脚本调用的，另一个脚本也会按同样的方式停止。如果最外层的脚本不是从一次交互的gsql会话中调用的而是用-f选项调用的，gsql将返回错误代码3，以示这个情况与致命错误条件的区别（错误代码为1）。	value取值范围为: on/off, true/false, yes/no, 1/0
\set AUTOCOMMIT [on off]	<p>设置当前gsql连接的自动提交行为，on为打开自动提交，off为关闭自动提交。默认情况下，gsql连接处于自动提交模式，每个单独的语句都被隐式提交。如果基于性能或者其它方面考虑，需要关闭自动提交时，需要用户自己显示的发出COMMIT命令来保证事务的提交。例如，在指定的业务SQL执行完之后发送COMMIT语句显式提交，特别是gsql客户端退出之前务必保证所有的事务已经提交。</p> <p><b>说明</b> gsql默认使用自动提交模式，若关闭自动提交，将会导致后面执行的语句都受到隐式事务包裹，数据库中不支持在事务中执行的语句不能在此模式下执行。</p>	<ul style="list-style-type: none"> <li>on表示打开自动提交。</li> <li>off表示关闭自动提交。</li> </ul>



表 1-22 大对象元命令

参数	参数说明
\lo_list	显示一个目前存储在该数据库里的所有GaussDB大对象和提供给大对象的注释。

表 1-23 全密态元命令

参数	参数说明
\send_token	全密态功能，传输密钥到服务端缓存，只在开启内存解密逃生通道的情况下使用。
\st	全密态功能，传输密钥到服务端缓存，只在开启内存解密逃生通道的情况下使用。
\clear_token	全密态功能，销毁服务端缓存的密钥，只在开启内存解密逃生通道的情况下使用。
\ct	全密态功能，销毁服务端缓存的密钥，只在开启内存解密逃生通道的情况下使用。
\key_info KEY_INFO	在全密态数据库特性中，用于设置用于访问外部密钥管理者的参数。

### 说明

分布式暂不支持全密态内存解密逃生通道。

M-Compatibility暂不支持全密态数据库。

## PATTERN

很多\d命令都可以用一个PATTERN参数来指定要被显示的对象名称。在最简单的情况下，PATTERN正好就是该对象的准确名称。在PATTERN中的字符通常会被变成小写形式（就像在SQL名称中那样），例如\dt FOO将会显示名为foo的表。把PATTERN放在双引号中可以阻止它被转换成小写形式。如果需要在PATTERN中包括一个真正的双引号字符，则需要把它写成两个相邻的双引号，这同样是符合SQL引用标识符的规则。例如，\dt "FOO""BAR"将显示名为FOO"BAR（不是foo"bar）的表。和普通的SQL名称规则不同，不能只在PATTERN的一部分周围放上双引号，例如\dt FOO"FOO"BAR将会显示名为fooFOObar的表。

不使用PATTERN参数时，\d命令会显示当前schema搜索路径中可见的全部对象，等价于用\*作为PATTERN。对象可见是指可以直接用名称引用该对象，而不需要用schema来进行限定。要查看数据库中所有的对象而不管它们的可见性，可以把\*\*用作PATTERN。

如果放在一个PATTERN中，\*将匹配任意字符序列（包括空序列），而?会匹配任意的单个字符（这种记号方法就像 Unix shell 的文件名PATTERN一样）。例如，\dt int\*会显示名称以int开始的表。但是如果被放在双引号内，\*和?就会失去这些特殊含义而变成普通的字符。

包含一个点号（.）的PATTERN被解释为一个schema名称模式后面跟上一个对象名称模式。例如，\dt foo\*.bar\*会显示名称以foo开始的schema中所有名称包括bar的表。

如果没有出现点号，那么模式将只匹配当前schema搜索路径中可见的对象。同样，双引号内的点号会失去其特殊含义并且变成普通的字符。

高级用户可以使用字符类等正则表达式记法，如[0-9]可以匹配任意数字。所有的正则表达式特殊字符都遵从POSIX正则表达式。以下字符除外：

- .会按照上面所说的作为一种分隔符。
- \*会被翻译成正则表达式记号.\*。
- ?会被翻译成.。
- \$则按字面意思匹配。

根据需要，可以通过书写?、(R+)、(R)和R?来分别模拟PATTERN字符.、R\*和R?。\$不需要作为一个正则表达式字符，因为PATTERN必须匹配整个名称，而不是像正则表达式的常规用法那样解释（换句话说，\$会被自动地追加到PATTERN上）。如果不希望该PATTERN的匹配位置被固定，可以在开头或者结尾写上\*。注意在双引号内，所有的正则表达式特殊字符会失去其特殊含义并且按照其字面意思进行匹配。另外，在操作符名称PATTERN中（即\do的PATTERN参数），正则表达式特殊字符也按照字面意思进行匹配。

## 1.6 常见问题处理

### 连接性能问题

- 开启log\_hostname，但是配置错误的DNS导致的连接性能问题。

连接数据库，通过“show log\_hostname”语句，检查数据库中是否开启log\_hostname参数。

如果开启了相关参数，那么数据库内核通过DNS反查客户端所在机器的主机名。如果数据库CN所在的主机配置了不正确的/不可达的DNS服务器，那么会导致数据库建立连接过程较慢。此参数的更多信息详见GUC参数log\_hostname。

- 数据库内核执行初始化语句较慢导致的性能问题。

此种情况定位较难，可以尝试使用Linux命令：strace。

```
strace gsql -U MyUserName -d gaussdb -h 127.0.0.1 -p 23508 -r -c '\q'  
Password for MyUserName:
```

在屏幕上打印出数据库的连接过程。比如较长时间停留在下面的操作上：

```
sendto(3, "Q\0\0\0\25SELECT VERSION()\0", 22, MSG_NOSIGNAL, NULL, 0) = 22  
poll([fd=3, events=POLLIN|POLLERR], 1, -1) = 1 ([fd=3, revents=POLLIN])
```

便可以确定是数据库执行“SELECT VERSION()”语句较慢。

在连接上数据库后，便可以通过执行“explain performance select version()”语句来确定初始化语句执行较慢的原因。更多信息，详见《开发者指南》中“SQL调优指南 > SQL执行计划介绍”章节。

另外还有一种场景不太常见：由于数据库CN所在机器的磁盘满或故障，此时所查询等受影响，无法进行用户认证，导致连接过程挂起。解决此问题清理数据库CN的数据盘空间便可。

- TCP连接创建较慢问题。

此问题可以参考上面的初始化语句较慢排查的做法，通过strace侦听，如果长时间停留在：

```
connect(3, {sa_family=AF_FILE, path="/home/test/tmp/gaussdb_llt1/s.PGSQL.61052"}, 110) = 0
```

或者

```
connect(3, {sa_family=AF_INET, sin_port=htons(61052), sin_addr=inet_addr("127.0.0.1")}, 16) = -1  
EINPROGRESS (Operation now in progress)
```

说明客户端与数据库端建立物理连接过慢，此时应当检查网络是否存在不稳定、网络吞吐量太大的问题。

- 资源负载满导致连接较慢的问题。

原因分析：当CPU、内存、I/O负载中的任意一项接近100%时，会出现gsql连接慢的现象。

问题解决：

- a. 通过top命令等确认CPU使用率；通过free命令确认内存使用情况；通过iostat命令确认I/O负载；还可以通过cm\_agent中的监控日志，以及数据库运维平台中的监测记录进行检查。
- b. 针对短时间内大量慢查询导致的峰值负载场景，可通过[数据库服务器的端口号+1]端口连接，查询pg\_stat\_activity视图；针对慢查询，可以使用系统函数pg\_terminate\_backend进行查杀会话。
- c. 针对业务量长期超负载情况（即无明显慢查询，或慢查询查杀后但新的查询依然会变成慢查询），应考虑降低业务负载、增加数据库资源的方式进行优化。

## 创建连接故障

- gsql: could not connect to server: No route to host

此问题一般是指定了不可达的地址或者端口导致的。请检查-h参数与-p参数是否添加正确。

- gsql: FATAL: Invalid username/password,login denied.

此问题一般是输入了错误的用户名和密码导致的，请联系数据库管理员，确认用户名和密码的正确性。

- gsql: FATAL: Forbid remote connection with trust method!

数据库由于安全问题，禁止远程登录时使用trust模式。这时需要修改gs\_hba.conf里的连接认证信息。请联系管理员处理。

### 说明

请不要修改gs\_hba.conf中数据库集群主机的相关设置，否则可能导致数据库功能故障。建议业务应用部署在数据库集群之外，而非集群内部。

- 在CN所在的主机连接数据库，添加“-h 127.0.0.1”可以连接，去掉后无法连接问题。

通过执行SQL语句“show unix\_socket\_directory”检查数据库CN使用的Unix套接字目录，是否与shell中的环境变量\$PGHOST一致。

如果检查结果不一致，那么修改PGHOST环境变量到GUC参数unix\_socket\_directory指向的目录。

- The "libpq.so" loaded mismatch the version of gsql, please check it.

此问题是由于环境中使用的libpq.so的版本与gsql的版本不匹配导致的，请通过“ldd gsql”命令确认当前加载的libpq.so的版本，并通过修改LD\_LIBRARY\_PATH环境变量来加载正确的libpq.so。

- gsql: symbol lookup error: xxx/gsql: undefined symbol: libpqVersionString

此问题是由于环境中使用的libpq.so的版本与gsql的版本不匹配导致的（也有可能是环境中存在PostgreSQL的libpq.so），请通过“ldd gsql”命令确认当前加载的libpq.so的版本，并通过修改LD\_LIBRARY\_PATH环境变量来加载正确的libpq.so。

- gsql: connect to server failed: Connection timed out  
Is the server running on host "xx.xxx.xxx.xxx" and accepting TCP/IP connections on port xxxx?  
此问题是由于网络连接故障造成。请检查客户端与数据库服务器间的网络连接。如果发现从客户端无法PING到数据库服务器端，则说明网络连接出现故障。请联系网络管理人员排查解决。

```
ping -c 4 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
From 10.10.10.1: icmp_seq=2 Destination Host Unreachable
From 10.10.10.1 icmp_seq=2 Destination Host Unreachable
From 10.10.10.1 icmp_seq=3 Destination Host Unreachable
From 10.10.10.1 icmp_seq=4 Destination Host Unreachable
--- 10.10.10.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 2999ms
```

- gsql: FATAL: permission denied for database "gaussdb"  
DETAIL: User does not have CONNECT privilege.  
此问题是由于用户不具备访问该数据库的权限，可以使用如下方法解决。
  - a. 使用管理员用户dbadmin连接数据库。  
`gsql -d gaussdb -U dbadmin -p 8000`
  - b. 赋予该用户访问数据库的权限。  
`GRANT CONNECT ON DATABASE gaussdb TO user1;`

### 📖 说明

实际上，常见的许多错误操作也可能产生用户无法连接上数据库的现象。如用户连接的数据库不存在，用户名或密码输入错误等。这些错误操作在客户端工具也有相应的提示信息。

```
gsql -d gaussdb -p 8000
gsql: FATAL: database "gaussdb" does not exist
```

```
gsql -d gaussdb -U user1 -p 8000
Password for user user1:
gsql: FATAL: Invalid username/password,login denied.
```

- gsql: FATAL: sorry, too many clients already, active/non-active: 197/3.  
此问题是由于系统连接数量超过了最大连接数量。请联系数据库DBA进行会话连接数管理，释放无用会话。  
关于查看用户会话连接数的方法如[表1-24](#)。  
会话状态可以在视图PG\_STAT\_ACTIVITY中查看。无用会话可以使用函数pg\_terminate\_backend进行释放。

```
select datid,pid,state from pg_stat_activity;
 datid |   pid   | state
-----+-----+-----
 13205 | 139834762094352 | active
 13205 | 139834759993104 | idle
(2 rows)
```

其中pid的值即为该会话的线程ID。根据线程ID结束会话。

```
SELECT PG_TERMINATE_BACKEND(139834759993104);
```

显示类似如下信息，表示结束会话成功。

```
PG_TERMINATE_BACKEND
-----
t
(1 row)
```

表 1-24 查看会话连接数

描述	命令
查看指定用户的会话连接数上限。	<p>执行如下命令查看连接到指定用户USER1的会话连接数上限。其中-1表示没有对用户user1设置连接数的限制。</p> <pre>SELECT ROLNAME,ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME='user1'; rolname   rolconlimit -----+----- user1              -1 (1 row)</pre>
查看指定用户已使用的会话连接数。	<p>执行如下命令查看指定用户USER1已使用的会话连接数。其中，1表示USER1已使用的会话连接数。</p> <pre>SELECT COUNT(*) FROM dv_sessions WHERE USERNAME='user1';  count ----- 1 (1 row)</pre>
查看指定数据库的会话连接数上限。	<p>执行如下命令查看连接到指定数据库gaussdb的会话连接数上限。其中-1表示没有对数据库gaussdb设置连接数的限制。</p> <pre>SELECT DATNAME,DATCONNLIMIT FROM PG_DATABASE WHERE DATNAME='gaussdb';  datname   datconlimit -----+----- gaussdb            -1 (1 row)</pre>
查看指定数据库已使用的会话连接数。	<p>执行如下命令查看指定数据库gaussdb上已使用的会话连接数。其中，1表示数据库gaussdb上已使用的会话连接数。</p> <pre>SELECT COUNT(*) FROM PG_STAT_ACTIVITY WHERE DATNAME='gaussdb'; count ----- 1 (1 row)</pre>
查看所有用户已使用会话连接数。	<p>执行如下命令查看所有用户已使用的会话连接数。</p> <pre>SELECT COUNT(*) FROM dv_sessions;  count ----- 10 (1 row)</pre>

- gsql: wait xxx.xxx.xxx.xxx:xxxx timeout expired

gsql在向数据库发起连接的时候，会有5分钟超时机制，如果在这个超时时间内，数据库未能正常的对客户端请求进行校验和身份认证，那么gsql会退出当前会话的连接过程，并报出如上错误。

一般来说，此问题是由于连接时使用的-h参数及-p参数指定的连接主机及端口有误（即错误信息中的xxx部分），导致通信故障；极少数情况是网络故障导致。要排除此问题，请检查数据库的主机名及端口是否正确。

- gsql: could not receive data from server: Connection reset by peer.

同时，检查CN日志中出现类似如下日志“FATAL: cipher file "/data/coordinator/server.key.cipher" has group or world access”，一般是由于数据目录或部分关键文件的权限被误操作篡改导致。请参照其他正常实例下的相关文件权限修改。

- gsql: FATAL: GSS authentication method is not allowed because XXXX user password is not disabled.

目标CN的gs\_hba.conf里配置了当前客户端IP使用"gss"方式来做认证，该认证算法不支持用作客户端的身份认证，请修改到"sha256"后再试。请联系管理员处理。

#### 说明

- 请不要修改gs\_hba.conf中数据库集群主机的相关设置，否则可能导致数据库功能故障。
- 建议业务应用部署在数据库集群之外，而非集群内部。

## 其他故障

- 出现因“总线错误”（Bus error）导致的core dump或异常退出。

一般情况下出现此种问题，是进程运行过程中加载的共享动态库（在Linux为.so文件）出现变化；或者进程二进制文件本身出现变化，导致操作系统加载机器的执行码或者加载依赖库的入口发生变化，操作系统出于保护目的将进程终止，产生core dump文件。

解决此问题，请重试。同时请尽可能避免在升级等运维操作过程中，在集群内部运行业务程序，避免升级时因替换文件产生此问题。

#### 说明

此故障的core dump文件的可能堆栈是dl\_main及其子调用，它是操作系统用来初始化进程做共享动态库加载的。如果进程已经初始化，但是共享动态库还未加载完成，严格意义上来说，进程并未完全启动。

# 2 gs\_dump

## 背景信息

gs\_dump是GaussDB用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等）。支持导出的数据库可以是默认数据库postgres，也可以是自定义数据库。

gs\_dump工具在进行数据导出时，其他用户可以访问集群数据库（读或写）。

gs\_dump工具支持导出完整一致的数据。例如，T1时刻启动gs\_dump导出A数据库，那么导出数据结果将会是T1时刻A数据库的数据状态，T1时刻之后对A数据库的修改不会被导出。

gs\_dump支持将数据库信息导出至纯文本格式的SQL脚本文件或其他归档文件中。

- 纯文本格式的SQL脚本文件：包含将数据库恢复为其保存时的状态所需的SQL语句。通过[gsql](#)运行该SQL脚本文件，可以恢复数据库。即使在其他主机和其他数据库产品上，只要对SQL脚本文件稍作修改，也可以用来重建数据库。
- 归档格式文件：包含将数据库恢复为其保存时的状态所需的数据，可以是tar格式、目录归档格式或自定义归档格式，详见[表2-1](#)。该导出结果必须与[gs\\_restore](#)配合使用来恢复数据库，gs\_restore工具在导入时，系统允许用户选择需要导入的内容，甚至可以在导入之前对等待导入的内容进行排序。

gs\_dump支持SSL加密通信，使用方式同gsq方式。

使用gs\_dump前请确保gs\_dump版本与数据库版本保持一致，高版本gs\_dump不保证完全兼容低版本内核数据。

## 主要功能

gs\_dump可以创建四种不同的导出文件格式，通过[-F或者--format=]选项指定，具体如[表2-1](#)所示。



表 2-1 导出文件格式

格式名称	-F的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含 SQL 语句和命令。命令可以由 gsql 命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，一般推荐纯文本格式。	使用 gsql 工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，推荐自定义归档格式。	使用 gs_restore 可以选择要从自定义归档/目录归档/tar 归档导出文件中导入相应的数据库对象。
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和 blob 对象对应的数据文件。	-	
tar 归档格式	t	tar 归档文件支持从导出文件中恢复所有或所选数据库对象。tar 归档格式不支持压缩且对于单独表大小应小于 8GB。	-	

### 📖 说明

可以使用 gs\_dump 工具将文件压缩为目录归档或自定义归档导出文件，减少导出文件的大小。生成目录归档或自定义归档导出文件时，默认进行中等级别的压缩。gs\_dump 程序无法压缩已归档导出文件。

### 注意事项

- 禁止修改 -F c/d/t 格式导出的文件和内容，否则可能无法恢复成功。对于 -F p 格式导出的文件，如有需要，可谨慎编辑导出的文件。
- 为了保证数据一致性和完整性，gs\_dump 会对需要转储的表设置共享锁。如果表在别的事务中设置了共享锁，gs\_dump 会等待锁释放后锁定表。如果无法在指定时间内锁定某个表，转储会失败。用户可以通过指定 --lock-wait-timeout 选项，自定义等待锁超时时间。
- 不支持加密导出存储过程和函数。
- 对于物化视图，本工具仅支持物化视图定义的导出，在导入后需手动执行 REFRESH 命令来进行数据恢复。
- 对于临时对象，本工具仅支持导出全局临时表。
- 本工具不支持在备机上使用。
- 由于 DN 上系统表中分布信息不完整，所以在 DN 使用 gs\_dump 时，不会转储表的分布信息。



- gs\_dump导出分区索引时，部分索引分区的属性无法导出，比如索引分区的 unusable状态。可以通过查询系统表PG\_PARTITION或者查询视图ADM\_IND\_PARTITIONS获取索引分区的具体属性，通过ALTER INDEX命令可以手动设置索引分区属性。
- 对于定时任务，本工具仅支持导出在MYSQL兼容性数据库中，通过CREATE EVENT创建的定时任务或通过高级包创建的非周期性定时任务。

#### 📖 说明

普通用户不支持导出DIRECTORY、SYNONYM，若普通用户进行相关导出，会提示“WARNING: xx not dumped because current user is not a superuser”。

## 语法

```
gs_dump [OPTION]... [DBNAME]
```

#### 📖 说明

“dbname”前面不需要加短或长选项。“dbname”指定要连接的数据库。

例如：

不需要-d，直接指定“dbname”。

```
gs_dump -p port_number testdb -f dump1.sql
```

或者

```
export PGDATABASE=testdb
```

```
gs_dump -p port_number -f dump1.sql
```

环境变量：PGDATABASE

## 参数说明

通用参数：

- -f, --file=FILENAME  
将输出发送至指定文件或目录。如果省略该参数，则使用标准输出。如果输出格式为(-F c/-F d/-F t)时，必须指定-f参数。如果-f的参数值含有目录，要求目录对当前用户具有读写权限。
- -F, --format=c|d|t|p  
选择输出格式。格式如下：
  - p|plain：输出一个文本SQL脚本文件（默认）。
  - c|custom：输出一个自定义格式的归档，并且以目录形式输出，作为gs\_restore输入信息。该格式是最灵活的输出格式，因为能手动选择，而且能在恢复过程中将归档项重新排序。该格式默认状态下会被压缩。
  - d|directory：该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和blob对象对应的数据文件。
  - t|tar：输出一个tar格式的归档形式，作为gs\_restore输入信息。tar格式与目录格式兼容；tar格式归档形式在提取过程中会生成一个有效的目录格式归档形式。但是，tar格式不支持压缩且对于单独表有8GB的大小限制。此外，表数据项的相应排序在恢复过程中不能更改。
- -v, --verbose  
指定verbose模式。该选项将导致gs\_dump向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。
- -V, --version

打印gs\_dump版本, 然后退出。

- -Z, --compress=0-9  
指定使用的压缩比级别。  
取值范围: 0~9
  - 0表示无压缩。
  - 1表示压缩比最小, 处理速度最快。
  - 9表示压缩比最大, 处理速度最慢。针对自定义归档格式, 该选项指定单个表数据片段的压缩, 默认方式是以中等级别进行压缩。tar归档格式和纯文本格式目前不支持压缩。
- --lock-wait-timeout=TIMEOUT  
请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表, 就选择失败。可以以任何符合SET statement\_timeout的格式指定超时时间。
- -?, --help  
显示gs\_dump命令行参数帮助, 然后退出。

转储参数:

- -a, --data-only  
只输出数据, 不输出模式(数据定义)。转储表数据、大对象和序列值。
- -b, --blobs  
该参数为扩展预留接口, 不建议使用。
- -c, --clean  
在将创建数据库对象的指令输出到备份文件之前, 先将清理(删除)数据库对象的指令输出到备份文件中。(如果目标数据库中没有任何对象, gs\_dump或gs\_restore工具可能会输出一些提示性的错误信息。)  
该选项只对文本格式有意义。针对归档格式, 可以在调用gs\_restore时指定选项。
- -C, --create  
备份文件以创建数据库和连接到创建的数据库的命令开始(如果命令脚本是这种方式执行, 无所谓在运行脚本之前连接的是哪一个数据库)。  
该选项只对文本格式有意义。针对归档格式, 可以在调用gs\_restore时指定选项。
- -E, --encoding=ENCODING  
以指定的字符集编码创建转储。默认情况下, 以数据库编码创建转储。(得到相同结果的另一个办法是将环境变量“PGCLIENTENCODING”设置为所需的转储编码。)
- -n, --schema=SCHEMA  
只转储与模式名称匹配的模式, 此选项包括模式本身和所有它包含的对象。如果该选项没有指定, 所有在目标数据库中的非系统模式将会被转储。写入多个-n选项来选择多个模式。此外, 根据gs\_dump的\d命令所使用的相同规则, 模式参数可被理解成一个pattern, 所以多个模式也可以通过在该pattern中写入通配符来选择。使用通配符时, 注意给pattern打引号, 防止shell扩展通配符。

#### 说明

- 当-n已指定时, gs\_dump不会转储已选模式所附着的任何其他数据库对象。因此, 无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当-n指定时, 非模式对象不会被转储。

转储支持多个模式的转储。多次输入-n schemaname转储多个模式。

例如:

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -n sch1 -n sch2
```

在上面这个例子中, sch1和sch2会被转储。

- -N, --exclude-schema=SCHEMA

不转储任何与模式pattern匹配的模式。Pattern将参照针对-n的相同规则来理解。可以通过输入多次-N, 不转储与任何pattern匹配的模式。

当同时输入-n和-N时, 会转储与至少一个-n选项匹配、与-N选项不匹配的模式。如果有-N没有-n, 则不转储常规转储中与-N匹配的模式。

转储过程支持排除多个模式。

在转储过程中, 输入-N exclude schema name排除多个模式。

例如:

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -N sch1 -N sch2
```

在上面这个例子中, sch1和sch2在转储过程中会被排除。

- -o, --oids

转储每个表的对象标识符 (OIDs), 作为表的一部分数据。该选项用于应用以某种方式参照了OID列的情况。如果不是以上这种情况, 请勿使用该选项。

- -O, --no-owner

不输出设置对象的归属这样的命令, 以匹配原始数据库。默认情况下, gs\_dump 会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的数据库对象的归属。如果脚本正在运行, 该语句不会执行成功, 除非是由系统管理员触发 (或是拥有脚本中所有对象的同一个用户)。通过指定-O, 编写一个任何用户都能存储的脚本, 且该脚本会授予该用户拥有所有对象的权限。

该选项只对文本格式有意义。针对归档格式, 可以在调用gs\_restore时指定选项。

- -s, --schema-only

只转储对象定义 (模式), 而非数据。

- -S, --sysadmin=NAME

该参数为扩展预留接口, 不建议使用。

- -t, --table=TABLE

指定转储的表 (或视图、或序列、或外表) 对象列表, 可以使用多个-t选项来选择多个表, 也可以使用通配符指定多个表对象。

当使用通配符指定多个表对象时, 注意给pattern打引号, 防止shell扩展通配符。

当使用-t时, -n和-N没有任何效应, 这是因为由-t选择的表的转储不受那些选项的影响。

### 📖 说明

- -t参数选项个数必须小于等于100。
- 如果-t参数选项个数大于100，建议使用参数--include-table-file来替换。
- 当-t已指定时，gs\_dump不会转储已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。
- -t tablename只转储在默认搜索路径中可见的表。-t \*.tablename转储数据库下所有模式下的tablename表。-t schema.table转储特定模式中的表。
- -t tablename不会导出表上的触发器信息。
- 对于表中包含大写字母的表，在使用-t参数指定导出时需对表名添加\"来导出。如对于表"abC"，导出需指定-t \"abC\"；如对于表schema."abC"，导出需指定-t schema.\"abC\"。
- -t "" 不匹配任何表。

例如：

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -t schema1.table1 -t schema2.table2
```

在上面这个例子中，schema1.table1和schema2.table2会被转储。

- --include-table-file=FILENAME  
指定需要dump的表文件。
- -T, --exclude-table=TABLE  
不转储的表（视图、序列、外表）对象列表，可以使用多个-t选项来选择多个表，也可以使用通配符指定多个表对象。  
当同时输入-t和-T时，会转储在-t列表中，而不在-T列表中的表对象。

例如：

```
gs_dump -h host_name -p port_number testdb -f backup/bkp_shl2.sql -T table1 -T table2
```

在上面这个例子中，table1和table2在转储过程中会被排除。

- --exclude-table-file=FILENAME  
指定不需要dump的表文件。

### 📖 说明

同--include-table-file，其内容格式如下：

```
schema1.table1  
schema2.table2  
.....
```

- -x, --no-acl  
防止转储访问权限（授权/撤销命令），只影响acl对象，不影响privilege对象。
- -q, --target  
指定导出兼容其他版本数据库的文本文件，目前支持v1和v5参数，指定其他参数不会报错，但不会生效。v1参数用于导出GaussDB v5版本数据库的数据为兼容GaussDB v1版本数据库的文本文件。v5参数用于导出GaussDB v5版本数据库的数据为GaussDB v5版本数据库格式的文本文件，减少了导入GaussDB v5版本数据库时的可能的报错情况。  
在使用v1参数时，建议和--exclude-guc="enable\_cluster\_resize"，--exclude-function，--exclude-with等选项共用，否则导入到GaussDB v1版本数据库时可能报错。
- -g, --exclude-guc

导出的文本文件中，不包括相关guc参数的set命令，目前只支持enable\_cluster\_resize，指定其他参数不会报错，但不会生效。

- --exclude-function  
不导出函数和存储过程。
- --exclude-with  
导出的表定义，末尾不添加WITH(orientation=row, compression=on) 这样的描述。
- --binary-upgrade  
该参数为扩展预留接口，不建议使用。
- --binary-upgrade-usermap="USER1=USER2"  
该参数为扩展预留接口，不建议使用。
- --column-inserts|--attribute-inserts  
以INSERT命令带列名 (INSERT INTO表 (列、...) 值...) 方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致失败语句对应的数据丢失，而非整个表内容。
- --disable-dollar-quoting  
该选项将禁止在函数体前使用美元符号\$，并强制使用SQL标准字符串语法对其进行引用。
- --include-alter-table  
dump后的表删除列。
- --disable-triggers  
该参数为扩展预留接口，不建议使用。
- --exclude-table-data=TABLE  
指定不转储任何匹配表pattern的表这方面的数据。依照针对-t的相同规则理解该pattern。  
可多次输入--exclude-table-data来排除匹配任何pattern的表。当用户需要特定表的定义但不需要其中的数据时，这个选项很有帮助。  
排除数据库中所有表的数据，参见-s, --schema-only。
- --inserts  
发出INSERT命令 (而非COPY命令) 时转储数据。这会导致恢复缓慢。  
但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致对应的一行数据丢失，而非整个表内容。注意如果重排列顺序，可能会导致恢复整个失败。列顺序改变时，--column-inserts选项不受影响，虽然会更慢。
- --no-security-labels  
该参数为扩展预留接口，不建议使用。
- --no-tablespaces  
不输出选择表空间的命令。使用该选项，无论默认表空间是哪一个，在恢复过程中所有对象都会被创建。  
该选项只对文本格式有意义。针对归档格式，可以在调用gs\_restore时指定选项。
- --no-unlogged-table-data  
该参数为扩展预留接口，不建议使用。

- `--non-lock-table`  
该参数仅供软件间接调用。
- `--quote-all-identifiers`  
强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。
- `--section=SECTION`  
指定已转储的名称区段（pre-data、data、和post-data）。
- `--serializable-deferrable`  
转储过程中使用可串行化事务，以确保所使用的快照与之后的数据库状态一致；要实现该操作需要在无异常状况的事务流中等待某个点，因为这样才能保证转储成功，避免引起其他事务出现serialization\_failure要重新再做。  
但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候，加载一个数据库的复制作为报告或其他只读加载共享的转储是有帮助的。没有这个选项，转储会反映一个与任何事务最终提交的序列化执行不一致的状态。  
如果当gs\_dump启动时，读写事务仍处于非活动状态，即便使用该选项也不会对其产生影响。如果读写事务处于活动状态，转储的开始时间可能会延迟一段不确定的时间。
- `--use-set-session-authorization`  
输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功，而ALTER OWNER需要的权限则低得多。但是SET SESSION AUTHORIZATION部分支持直接使用密文密码，因为使用此参数导出的脚本可能无法正常恢复，不建议使用此参数导出。

#### 📖 说明

SET SESSION AUTHORIZATION使用范围：

- 系统管理员可以通过SET SESSION AUTHORIZATION语句切换到普通用户，无法切换到初始用户、其他sysadmin、opadmin、monadmin、poladmin和auditadmin。
- 其他用户无法通过SET SESSION AUTHORIZATION语句切换用户。
- `--with-encryption=AES128`  
指定转储数据需用AES128进行加密。
- `--with-key=KEY`  
AES128密钥规则如下：  
密钥长度为8~16个字符。  
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。

#### 📖 说明

不支持加密导出存储过程和函数。

- `--with-salt=RANDVALUES`  
gs\_dumpall使用此参数传递随机值。
- `--include-nodes`  
将TO NODE/TO GROUP语句包含在已转储的CREATE TABLE语句中。该参数只对外生效。

- --include-extensions  
在转储中包含扩展。

#### 须知

扩展功能为内部使用功能，不建议用户使用。

- --include-depend-objs  
备份结果包含依赖于指定对象的对象信息。该参数需要同-t/--include-table-file参数关联使用才会生效。
- --exclude-self  
备份结果不包含指定对象自身的的信息。该参数需要同-t/--include-table-file参数关联使用才会生效。
- --pipeline  
使用管道传输密码，禁止在终端使用。
- --dont-overwrite-file  
文本、tar、以及自定义格式情况下会重写现有文件。这对目录格式不适用。

例如：

设想这样一种情景，即当前目录下backup.sql已存在。如果在输入命令中输入-f backup.sql选项时，当前目录恰好也生成backup.sql，文件就会被重写。

如果备份文件已存在，且输入--dont-overwrite-file选项，则会报告附带‘转储文件已经存在’信息的错误。

```
gs_dump -p port_number testdb -f backup.sql -F plain --dont-overwrite-file
```

#### 说明

- -s/--schema-only和-a/--data-only不能同时使用。
- -c/--clean和-a/--data-only不能同时使用。
- --inserts/--column-inserts和-o/--oids不能同时使用，因为INSERT命令不能设置OIDs。
- --role和--rolepassword必须一起使用。
- --binary-upgrade-usermap和--binary-upgrade必须一起使用。
- --include-depend-objs/--exclude-self需要同-t/--include-table-file参数关联使用才会生效。
- --exclude-self必须同--include-depend-objs一起使用。
- --with-encryption=AES128仅支持-F p/plain。
- --with-key=KEY仅支持-F p/plain。
- --with-salt=RANDVALUES由gs\_dumpall调用，不需要用户手动输入。

连接参数：

- -h, --host=HOSTNAME  
指定主机名称。如果数值以斜杠开头，则被用作到Unix域套接字的路径。缺省从PGHOST环境变量中获取（如果已设置），否则，尝试一个Unix域套接字连接。  
该参数只针对集群外，对集群内本机只能用127.0.0.1。  
例如：主机名  
环境变量：PGHOST
- -p, --port=PORT



指定主机端口。在开启线程池情况下，建议使用 pooler port，即主机端口+1。

环境变量：PGPORT

- -U, --username=NAME

指定所连接主机的用户名，跨节点执行不支持使用初始用户。

环境变量：PGUSER

- -w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是trust，则不会对系统管理员进行密码验证，即无需输入-W选项；如果不加此参数，并且不是系统管理员，则会提示交互式输入，为了系统安全，推荐使用交互式输入密码方式。

- --role=ROLENAME

指定创建转储使用的角色名。选择该选项，会使gs\_dump连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs\_dump要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。

- --rolepassword=ROLEPASSWORD

指定角色名的密码。

## 说明

如果某数据库集群有任何本地数据要添加到template1数据库，请谨慎将gs\_dump的输出恢复到一个真正的空数据库中，否则可能会因为被添加对象的定义被复制，出现错误。要创建一个无本地添加的空数据库，需从template0而非template1复制，例如：

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

tar归档形式的文件大小不得超过8GB（tar文件格式的固有限制）。tar文档整体大小和任何其他输出格式没有限制，操作系统可能对此有要求。

由gs\_dump生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此，建议从某转储文件恢复之后运行ANALYZE以确保最佳效果。转储文件不包含任何ALTER DATABASE...SET命令，这些设置由gs\_dumpall转储，还有数据库用户和其他完成安装设置。

## 示例

使用gs\_dump转储数据库为SQL文本文件或其它格式的操作，如下所示。

示例中“backup/MPPDB\_backup.sql”表示导出的文件，其中backup表示相对于当前目录的相对目录；“37300”表示数据库服务器端口；“testdb”表示要访问的数据库名。

### 说明

导出操作时，请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例1：执行gs\_dump，导出testdb数据库全量信息，导出的MPPDB\_backup.sql文件格式为纯文本格式。



```
gs_dump -U omm -f backup/MPPDB_backup.sql -p 37300 testdb -F p
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: The total objects number is
356.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: [100.00%] 356 objects have
been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: dump database testdb
successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 09:49:17]: total time: 1274 ms
```

使用gsql程序从纯文本导出文件中导入数据。

示例2: 执行gs\_dump, 导出testdb数据库全量信息, 导出的MPPDB\_backup.tar文件格式为tar格式。

```
gs_dump -U omm -f backup/MPPDB_backup.tar -p 37300 testdb -F t
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:24]: The total objects number is
1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: [100.00%] 1369 objects have
been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: dump database testdb
successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:02:53]: total time: 50086 ms
```

示例3: 执行gs\_dump, 导出testdb数据库全量信息, 导出的MPPDB\_backup.dmp文件格式为自定义归档格式。

```
gs_dump -U omm -f backup/MPPDB_backup.dmp -p 37300 testdb -F c
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:05:40]: The total objects number is
1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: [100.00%] 1369 objects have
been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: dump database testdb
successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:06:03]: total time: 36620 ms
```

示例4: 执行gs\_dump, 导出testdb数据库全量信息, 导出的MPPDB\_backup文件格式为目录格式。

```
gs_dump -U omm -f backup/MPPDB_backup -p 37300 testdb -F d
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:04]: The total objects number is
1369.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: [100.00%] 1369 objects have
been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: dump database testdb
successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:16:23]: total time: 33977 ms
```

示例5: 执行gs\_dump, 导出testdb数据库信息, 但不导出/home/MPPDB\_temp.sql中指定的表信息。导出的MPPDB\_backup.sql文件格式为纯文本格式。

```
gs_dump -U omm -p 37300 testdb --exclude-table-file=/home/MPPDB_temp.sql -f backup/
MPPDB_backup.sql
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:01]: The total objects number is
1367.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: [100.00%] 1367 objects have
been dumped.
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: dump database testdb
successfully
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-27 10:37:22]: total time: 37017 ms
```

示例6: 执行gs\_dump, 仅导出依赖于指定表testtable的视图信息。然后创建新的testtable表, 再恢复依赖其上的视图。

备份仅依赖于testtable的视图

```
gs_dump -U omm -s -p 37300 testdb -t PUBLIC.testtable --include-depend-objs --exclude-self -f backup/
MPPDB_backup.sql -F p
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: The total objects number is
```

```
331.  
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: [100.00%] 331 objects have  
been dumped.  
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: dump database testdb  
successfully  
gs_dump[user='omm'][localhost][port='37300'][testdb][2018-06-15 14:12:54]: total time: 327 ms
```

#### 修改testtable名称

```
gsql -p 37300 testdb -r -c "ALTER TABLE PUBLIC.testtable RENAME TO testtable_bak;"
```

#### 创建新的testtable表

```
CREATE TABLE PUBLIC.testtable(a int, b int, c int);
```

#### 还原依赖于testtable的视图

```
gsql -p 37300 testdb -r -f backup/MPPDB_backup.sql
```

## 相关命令

[gs\\_dumpall](#), [gs\\_restore](#)

# 3 gs\_dumpall

## 背景信息

gs\_dumpall是GaussDB用于导出所有数据库相关信息工具，它可以导出集群数据库的所有数据，包括默认数据库postgres的数据、自定义数据库的数据、以及集群所有数据库公共的全局对象。

gs\_dumpall工具在进行数据导出时，其他用户可以访问集群数据库（读或写）。

gs\_dumpall工具支持导出完整一致的数据。例如，T1时刻启动gs\_dumpall导出整个集群数据库，那么导出数据结果将会是T1时刻该集群数据库的数据状态，T1时刻之后对集群数据库的修改不会被导出。

gs\_dumpall在导出整个集群所有数据库时分为两部分：

- gs\_dumpall自身对所有数据库公共的全局对象进行导出，包括有关数据库用户和组，表空间以及属性（例如，适用于数据库整体的访问权限）信息。
- gs\_dumpall通过调用gs\_dump来完成集群中各数据库的SQL脚本文件导出，该脚本文件包含将数据库恢复为其保存时的状态所需要的全部SQL语句。

以上两部分导出的结果为纯文本格式的SQL脚本文件，使用gsql运行该脚本文件可以恢复集群数据库。

gs\_dumpall支持SSL加密通信，使用方式同gsql方式。

使用gs\_dumpall前请确保gs\_dumpall版本与gs\_dump版本、数据库版本保持一致，高版本gs\_dumpall不保证完全兼容低版本内核数据。

## 注意事项

- 禁止修改导出的文件和内容，否则可能无法恢复成功。
- 为了保证数据一致性和完整性，gs\_dumpall会对需要转储的表设置共享锁。如果某张表在别的事务中设置了共享锁，gs\_dumpall会等待此表的锁释放后锁定此表。如果无法在指定时间内锁定某张表，转储会失败。用户可以通过指定--lock-wait-timeout选项，自定义等待锁超时时间。
- 由于gs\_dumpall读取所有数据库中的表，因此必须以数据库集群管理员身份进行连接，才能导出完整文件。在使用gsql执行脚本文件导入时，同样需要管理员权限，以便添加用户和组，以及创建数据库。导入备份前，验证其安全性，防止管理员权限被利用。

- 使用gs\_dumpall导出所有数据库对象，并希望新的实例环境上进行导入时，需要保证导出和导入时使用用户的名称和权限相同，否则会出现名称不一致或权限不足的报错。
- 对于定时任务，本工具仅支持导出在MySQL兼容性数据库中，通过CREATE EVENT创建的定时任务或通过高级包创建的非周期性定时任务。

## 语法

```
gs_dumpall [OPTION]...
```

## 参数说明

通用参数：

- -f, --filename=FILENAME  
将输出发送至指定文件。如果这里省略，则使用标准输出。
- -v, --verbose  
指定verbose模式。该选项将导致gs\_dumpall向转储文件输出详细的对象注解和启动/停止次数，向标准错误流输出处理信息。
- -V, --version  
打印gs\_dumpall版本，然后退出。
- --lock-wait-timeout=TIMEOUT  
请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表，就选择失败。可以以任何符合SET statement\_timeout的格式指定超时时间。
- -?, --help  
显示gs\_dumpall命令行参数帮助，然后退出。

转储参数：

- -a, --data-only  
只转储数据，不转储模式（数据定义）。
- -c, --clean  
在重新创建数据库之前，执行SQL语句清理（删除）这些数据库。针对角色和表空间的转储命令已添加。
- -g, --globals-only  
只转储全局对象（角色和表空间），无数据库。
- -o, --oids  
转储每个表的对象标识符（OIDs），作为表的一部分数据。该选项用于应用以某种方式参照了OID列的情况。如果不是以上这种情况，请勿使用该选项。
- -O, --no-owner  
不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs\_dumpall会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。如果脚本正在运行，该语句不会执行成功，除非是由系统管理员触发（或是拥有脚本中所有对象的同一个用户）。通过指定-O，编写一个任何用户都能存储的脚本，且该脚本会授予该用户拥有所有对象的权限，因为没有使用ALTER OWNER或SET SESSION AUTHORIZATION语句，导入时一直使用执行用户权限，所以在导入前请确认转储文件内是否存在风险。例如，检查转储文件中是否包含提权语句，且该语句管理员是否知晓。

- `-r, --roles-only`  
只转储角色，不转储数据库或表空间。
- `-s, --schema-only`  
只转储对象定义（模式），而非数据。
- `-S, --sysadmin=NAME`  
该参数为扩展预留接口，不建议使用。
- `-t, --tablespaces-only`  
只转储表空间，不转储数据库或角色。
- `-x, --no-privileges`  
防止转储访问权限（授权/撤销命令）。
- `--column-inserts|--attribute-inserts`  
以INSERT命令带列名（INSERT INTO表（列、…）值…）方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令，所以在重新加载某行时出现的错误只会导致失败语句对应的数据丢失，而非整个表内容。
- `--disable-dollar-quoting`  
该选项将禁止在函数体前使用美元符号\$，并强制使用SQL标准字符串语法对其进行引用。
- `--disable-triggers`  
该参数为扩展预留接口，不建议使用。
- `--inserts`  
发出INSERT命令（而非COPY命令）时转储数据。这会导致恢复缓慢。注意如果重排列顺序，可能会导致恢复整个失败。--column-inserts选项更加安全，虽然可能更慢些。
- `--no-security-labels`  
该参数为扩展预留接口，不建议使用。
- `--no-tablespaces`  
请勿输出创建表空间的命令，也请勿针对对象选择表空间。使用该选项，无论默认表空间是哪一个，在恢复过程中所有对象都会被创建。
- `--no-unlogged-table-data`  
该参数为扩展预留接口，不建议使用。
- `--include-alter-table`  
导出表中已删除的列信息。
- `--quote-all-identifiers`  
强制对所有标识符加引号。为了向后续版本迁移，且其中可能涉及引入额外关键词，在转储相应数据库时该选项会有帮助。
- `--dont-overwrite-file`  
不重写当前文件。
- `--use-set-session-authorization`  
输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER命令来确定对象所有权。这样令转储更加符合标准，但是如果转储文件中的对象的历史有些问题，那么可能不能正确恢复。并且，使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功，而ALTER OWNER需要的权限则低得多。但是SET SESSION AUTHORIZATION部分支持使

用密文密码随意切换用户及权限，因为使用此参数导出的脚本可能无法正常恢复，不建议使用此参数导出。

### 📖 说明

SET SESSION AUTHORIZATION使用范围：

- 系统管理员可以通过SET SESSION AUTHORIZATION语句切换到普通用户，无法切换到初始用户、其他sysadmin、opradmin、monadmin、poladmin和auditadmin。
- 其他用户无法通过SET SESSION AUTHORIZATION语句切换用户。
- --with-encryption=AES128  
指定转储数据需用AES128进行加密。
- --with-key=KEY  
AES128密钥规则如下：  
密钥长度为8~16个字符。  
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。
- --include-extensions  
如果include-extensions参数被设置，将备份所有的CREATE EXTENSION语句。

---

### 须知

扩展功能为内部使用功能，不建议用户使用。

- 
- --include-templatedb  
转储过程中包含模板库。
  - --dump-nodes  
转储过程中包含节点和Node Group。
  - --include-nodes  
将TO NODE语句包含在已转储的CREATE TABLE命令中。
  - --include-buckets  
该参数为扩展预留接口，不建议使用。
  - --dump-wrm  
存储过程中包含负载资源管理器，具体包括资源池、负载组以及负载组映射。
  - --binary-upgrade  
该参数为扩展预留接口，不建议使用。

### 📖 说明

M-Compatibility数据库不支持该选项，导出M-Compatibility数据库时会跳过该选项。

- --binary-upgrade-usermap="USER1=USER2"  
该参数为扩展预留接口，不建议使用。
- --non-lock-table  
该参数仅供软件间接口调用。
- --tablespaces-postfix  
该参数为扩展预留接口，不建议使用。

- --parallel-jobs  
指定备份进程并发数，取值范围为1~1000。
- --pipeline  
使用管道传输密码，禁止在终端使用。

#### 📖 说明

- -g/--globals-only和-r/--roles-only不能同时使用。
- -g/--globals-only和-t/--tablespaces-only不能同时使用。
- -r/--roles-only和-t/--tablespaces-only不能同时使用。
- -s/--schema-only和-a/--data-only不能同时使用。
- -r/--roles-only和-a/--data-only不能同时使用。
- -t/--tablespaces-only和-a/--data-only不能同时使用。
- -g/--globals-only和-a/--data-only不能同时使用。
- --tablespaces-postfix和--binary-upgrade必须一起使用。
- --binary-upgrade-usermap和--binary-upgrade必须一起使用。
- --parallel-jobs和-f/--file必须一起使用。

#### 连接参数：

- -h, --host=HOSTNAME  
指定主机的名称。如果取值是以斜线开头，它将用作Unix域套接字的目录。默认值取自PGHOST环境变量；如果没有设置，将启动某个Unix域套接字建立连接。  
该参数只针对集群外，对集群内本机只能用127.0.0.1。  
环境变量：PGHOST
- -l, --database=DATABASENAME  
指定所连接的转储全局对象的数据库名称，并去寻找还有其他哪些数据库需要被转储。如果没有指定，会使用postgres数据库，如果postgres数据库不存在，会使用template1。
- -p, --port=PORT  
指定服务器所侦听的TCP端口或本地Unix域套接字后缀，以确保连接。默认值设置为PGPORT环境变量。  
在开启线程池情况下，建议使用 pooler port，即侦听端口+1。  
环境变量：PGPORT
- -U, --username=NAME  
所连接的用户名，跨节点执行不支持使用初始用户。  
环境变量：PGUSER
- -w, --no-password  
不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。
- -W, --password=PASSWORD  
指定用户连接的密码。如果主机的认证策略是trust，则不会对系统管理员进行密码验证，即无需输入-W选项；如果不加此参数，并且不是系统管理员，则会提示交互式输入，为了系统安全，推荐使用交互式输入密码方式。

- `--role=ROLENAME`  
指定创建转储使用的角色名。选择该选项，会使gs\_dumpall连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs\_dumpall要求的权限时，该选项会起到作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录，而使用该选项能够在不违反该规定的情况下完成转储。
- `--rolepassword=ROLEPASSWORD`  
指定具体角色用户的角色密码。

## 说明

- 由于gs\_dumpall内部调用gs\_dump，所以一些诊断信息参见[gs\\_dump](#)。
- 一旦恢复，建议在每个数据库上运行ANALYZE，优化程序提供有用的统计数据。
- gs\_dumpall恢复前需要所有必要的表空间目录为空；否则，对于处在非默认位置的数据库，数据库创建会失败。

## 示例

使用gs\_dumpall一次导出集群的所有数据库。

### 说明

gs\_dumpall仅支持纯文本格式导出。所以只能使用gsql恢复gs\_dumpall导出的转储内容。

```
gs_dumpall -U omm -f backup/bkp2.sql -p 37300
gs_dump[user='omm'][localhost][port='37300'][dbname='testdb'][2018-06-27 09:55:09]: The total objects
number is 2371.
gs_dump[user='omm'][localhost][port='37300'][dbname='testdb'][2018-06-27 09:55:35]: [100.00%] 2371
objects have been dumped.
gs_dump[user='omm'][localhost][port='37300'][dbname='testdb'][2018-06-27 09:55:46]: dump database
dbname='testdb' successfully
gs_dump[user='omm'][localhost][port='37300'][dbname='testdb'][2018-06-27 09:55:46]: total time: 55567
ms
gs_dumpall[user='omm'][localhost][port='37300'][2018-06-27 09:55:46]: dumpall operation successful
gs_dumpall[user='omm'][localhost][port='37300'][2018-06-27 09:55:46]: total time: 56088 ms
其中： backup/bkp2.sql表示导出的文件， 37300表示数据库服务器端口， omm为用户名。
```

## 相关命令

[gs\\_dump](#)， [gs\\_restore](#)



# 4 gs\_restore

## 背景信息

gs\_restore是GaussDB提供的针对gs\_dump导出数据的导入工具。通过此工具可将gs\_dump导出生成的文件进行导入。

主要功能包含：

- 导入到数据库  
如果连接参数中指定了数据库，则数据将被导入到指定的数据库中。其中，并行导入必须指定连接的密码。
- 导入到归档文件  
如果参数指定"-l"，则生成归档文件，其中包含简略的数据总结。

gs\_restore支持SSL加密通信，使用方式同gsqll方式。

使用gs\_restore前请确保gs\_restore版本与gs\_dump版本、数据库版本保持一致。

## 命令格式

```
gs_restore [OPTION]... FILE
```

### 说明

- FILE没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件，需输入dbname或-l选项。不允许用户同时输入dbname和-l选项。
- gs\_restore默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议选择使用“-c”和“-e”参数。“-c”表示在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象；“-e”表示当发送SQL语句到数据库时如果出现错误请退出，默认状态下会继续，且在导入后会显示一系列错误信息。
- 在进行导入时，如果schema对象的owner拥有OPRADMIN的系统权限，那么在导入时需要使用初始用户。

## 参数说明

通用参数：

- -d, --dbname=NAME  
连接数据库dbname并直接导入到该数据库中。

- -f, --file=FILENAME  
指定生成归档的输出文件，使用-l时列表的输出文件。  
默认是标准输出。

#### 📖 说明

-f不能同-d一起使用。

- -F, --format=c|d|t  
指定归档格式。由于gs\_restore会自动决定格式，因此不需要指定格式。  
取值范围：
  - c/custom: 该归档形式为gs\_dump的自定义格式。
  - d/directory: 该归档形式是一个目录归档形式。
  - t/tar: 该归档形式是一个tar归档形式。
- -l, --list  
列出归档形式内容。这一操作的输出可用作-L选项的输入。注意如果像-n或-t的过滤选项与-l使用，过滤选项将会限制列举的项目（即归档形式内容）。
- -v, --verbose  
指定verbose模式。
- -V, --version  
打印gs\_restore版本，然后退出。
- -?, --help  
显示gs\_restore命令行参数帮助，然后退出。

导入参数：

- -a, -data-only  
只导入数据，不导入模式（数据定义）。gs\_restore的导入是以追加方式进行的。
- -c, --clean  
在重新创建数据库对象前，清理（删除）已存在于将要还原的数据库中的数据库对象。如果目标数据库中没有删除操作涉及的对象，可能会输出一些提示性的错误信息。
- -C, --create  
导入数据库之前会先使用CREATE DATABASE创建数据库（指定该选项后，-d指定的数据库仅用以执行CREATE DATABASE命令，所有数据依然会导入到创建的数据库中）。
- -e, --exit-on-error  
当发送SQL语句到数据库时如果出现错误，请退出。默认状态下会继续，且在导入后会显示一系列错误信息。
- -I, --index=NAME  
只导入已列举的index的定义。允许导入多个index。如果多次输入-I index导入多个index。  
例如：  

```
gs_restore -h host_name -p port_number -d testdb -I Index1 -I Index2 backup/MPPDB_backup.tar
```

  
在上面这个例子中，Index1和Index2会被导入。
- -j, --jobs=NUM

运行gs\_restore最耗时的部分（如加载数据、创建index、或创建约束）使用并发任务。该选项能大幅缩短导入时间，即将一个大型数据库导入到某一多处理器的服务器上。

每个任务可能是一个进程或一个线程，这由操作系统决定。每个任务与服务器进行单独连接。

该选项的最优值取决于服务器的硬件设置、客户端、以及网络。还包括这些因素，如CPU核数量、硬盘设置。建议是从增加服务器上的CPU核数量入手，更大的值（服务器上CPU核数量）在很多情况下也能导致数据文件更快地被导入。相应的，过高的值会由于超负荷反而导致性能降低。

该选项只支持自定义归档格式。输入文件必须是常规文件（不能是像pipe的文件）。如果是通过脚本文件，而非直接连接数据库服务器，该选项可忽略。而且，多任务不能与--single-transaction选项一起使用。

### 📖 说明

此参数适用于多表/多索引/多约束的情况。实际使用过程中，创建的进程数（或线程数）与表、索引、约束等的数量有关，最高并发不会超过给定的jobs数。

- -L, --use-list=FILENAME

只导入列举在list-file中的那些归档形式元素，导入顺序以它们在文件中的顺序为准。注意如果像-n或-t的过滤选项与-L使用，它们将会进一步限制导入的项目。

一般情况下，list-file是通过编辑前面提到的某个-l参数的输出创建的。文件行的位置可更改或直接删除行，也可使用分号(;)在行的开始注出。见下文的举例。

- -n, --schema=NAME

只导入已列举的模式中的对象。

该选项可与-t选项一起用以导入某个指定的表。

多次输入-n *schemaname*可以导入多个模式。

例如：

```
gs_restore -h host_name -p port_number -d testdb -n sch1 -n sch2 backup/MPPDB_backup.tar
```

在上面这个例子中，sch1和sch2会被导入。

- -O, --no-owner

不输出设置对象的归属这样的命令，以匹配原始数据库。默认情况下，gs\_restore会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。除非是由系统管理员（或是拥有脚本中所有对象的同一个用户）进行数据库首次连接的操作，否则语句会失败。使用-O选项，任何用户名都可用于首次连接，且该用户拥有所有已创建的对象。

- -P, --function=NAME(args)

只导入已列举的函数。请按照函数所在转储文件中的目录，准确拼写函数名称和参数。

当-P单独使用时，表示导入文件中所有'function-name(args)'函数；当-P同-n一起使用时，表示导入指定模式下的'function-name(args)'函数；多次输入-P，而仅指定一次-n，表示所有导入的函数默认都是位于-n模式下的。

可以多次输入-n *schema-name* -P 'function-name(args)'同时导入多个指定模式下的函数。

例如：

```
./gs_restore -h host_name -p port_number -d testdb -n test1 -P 'Func1(integer)' -n test2 -P 'Func2(integer)' backup/MPPDB_backup.tar
```

在上面这个例子中，test1模式下的函数Func1(i integer)和test2模式下的函数Func2(j integer)会被一起导入。

- `-s, --schema-only`  
只导入模式（数据定义），不导入数据（表内容）。当前的序列值也不会导入。
- `-S, --sysadmin=NAME`  
该参数为扩展预留接口，不建议使用。
- `-t, --table=NAME`  
只导入已列举的表定义、数据或定义和数据。该选项与`-n`选项同时使用时，用来指定某个模式下的表对象。`-n`参数不输入时，默认为PUBLIC模式。多次输入`-n <schemaname> -t <tablename>`可以导入指定模式下的多个表。

例如：

导入PUBLIC模式下的table1

```
gs_restore -h host_name -p port_number -d testdb -t table1 backup/MPPDB_backup.tar
```

导入test1模式下的test1和test2模式下test2

```
gs_restore -h host_name -p port_number -d testdb -n test1 -t test1 -n test2 -t test2 backup/MPPDB_backup.tar
```

导入PUBLIC模式下的table1和test1 模式下test1

```
gs_restore -h host_name -p port_number -d testdb -n PUBLIC -t table1 -n test1 -t table1 backup/MPPDB_backup.tar
```

#### 须知

- `-t`不支持`schema_name.table_name`的输入格式，指定此格式不会报错，但不会生效。
  - 当`-t`已指定时，`gs_restore`不会导入已选表所附着的任何其他数据库对象。因此，无法保证某个指定表的转储结果能够自行成功地导入到一个空数据库中。
  - `-t tablename`不会导入表上的触发器信息。
- 
- `-T, --trigger=NAME`  
该参数为扩展预留接口。
  - `-x, --no-privileges/--no-acl`  
防止导入访问权限（`grant/revoke`命令）。
  - `-1, --single-transaction`  
执行导入作为一个单独事务（即把命令包围在`BEGIN/COMMIT`中）。该选项确保要么所有命令成功完成，要么没有改变应用。该选项意为`--exit-on-error`。
  - `--disable-triggers`  
该参数为扩展预留接口，不建议使用。
  - `--no-data-for-failed-tables`  
默认状态下，即使创建表的命令失败（如表已经存在），表数据仍会被导入。使用该选项，像这种表的数据会被跳过。如果目标数据库已包含想要的表内容，这种行为会有帮助。  
该选项只有在直接导入到某数据库中时有效，不针对生成SQL脚本文件输出。
  - `--no-security-labels`  
该参数为扩展预留接口，不建议使用。

- `--no-tablespaces`  
不输出选择表空间的命令。使用该选项，无论默认表空间是哪一个，在导入过程中所有对象都会被创建。
- `--section=SECTION`  
导入已列举的区段（如pre-data、data、或post-data）。
- `--use-set-session-authorization`  
该选项用来进行文本格式的备份。  
输出SET SESSION AUTHORIZATION命令，而非ALTER OWNER命令，用以决定对象归属。该选项使转储更加兼容标准，但通过参考转储中对象的记录，导入过程可能会有问题。使用SET SESSION AUTHORIZATION的转储要求必须是系统管理员，同时在导入前还需参考"SET SESSION AUTHORIZATION"，手工对导出文件的密码进行修改验证，只有这样才能进行正确的导入操作，相比之下，ALTER OWNER对权限要求较低。
- `--pipeline`  
使用管道传输密码，禁止在终端使用。

## 须知

- 如果安装过程中有任何本地数据要添加到template1数据库，请谨慎将gs\_restore的输出载入到一个真正的空数据库中；否则可能会因为被添加对象的定义被复制，而出现错误。要创建一个无本地添加的空数据库，需从template0而非template1复制，例如：

```
CREATE DATABASE foo WITH TEMPLATE template0;
```

- gs\_restore不能选择性地导入大对象；例如只能导入那些指定表的对象。如果某个归档形式包含大对象，那所有大对象都会被导入。如果此归档对象通过-L、-t或其他选项被排除，那么所有大对象一个都不会被导入。

## 说明

1. `-d/--dbname` 和 `-f/--file` 不能同时使用。
2. `-s/--schema-only` 和 `-a/--data-only`不能同时使用。
3. `-c/--clean` 和 `-a/--data-only`不能同时使用。
4. 使用`--single-transaction`时，`-j/--jobs`必须为单任务。
5. `--role` 和 `--rolepassword`必须一起使用。

### 连接参数：

- `-h, --host=HOSTNAME`  
指定的主机名称。如果取值是以斜线开头，他将用作Unix域套接字的目录。默认值取自PGHOST环境变量；如果没有设置，将启动某个Unix域套接字建立连接。  
该参数只针对集群外，对集群内本机只能用127.0.0.1。  
环境变量：PGHOST
- `-p, --port=PORT`  
指定服务器所侦听的TCP端口或本地Unix域套接字后缀，以确保连接。默认值设置为PGPORT环境变量。  
在开启线程池情况下，建议使用 pooler port，即侦听端口+1。

环境变量：PGPORT

- -U, --username=NAME

所连接的用户名，跨节点执行不支持使用初始用户。

环境变量：PGUSER

- -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给出，则连接尝试将会失败。该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

- -W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是trust，则不会对系统管理员进行密码验证，即无需输入-W参数；如果不加此参数，并且不是系统管理员，则会提示交互式输入，为了系统安全，推荐使用交互式输入密码方式。

- --role=ROLENAME

指定导入操作使用的角色名。选择该参数，会使gs\_restore连接数据库后，发起一个SET ROLE角色名命令。当所授权用户（由-U指定）没有gs\_restore要求的权限时，该参数会起作用，即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录，而使用该参数能够在不违反该规定的情况下完成导入。

- --rolepassword=ROLEPASSWORD

指定具体角色用户的角色密码。

## 示例

特例：执行gs\_restore程序，使用如下选项导入由gs\_dump/gs\_dumpall生成导出文件夹（纯文本格式）的MPPDB\_backup.sql文件到testdb数据库。

```
gs_restore -d testdb -p 8000 -f /home/omm/test/MPPDB_backup.sql
SET
SET
SET
SET
SET
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
SET
CREATE INDEX
REVOKE
REVOKE
GRANT
GRANT
total time: 30476 ms
```

示例中“-f”后的是导出的文件，“8000”表示数据库服务器端口；“testdb”表示要访问的数据库名。

gs\_restore用来导入由gs\_dump生成的导出文件。

示例1：执行gs\_restore，将导出的MPPDB\_backup.dmp文件（自定义归档格式）导入到testdb数据库。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb
restore operation successful
total time: 13053 ms
```

示例2: 执行gs\_restore, 将导出的MPPDB\_backup.tar文件 (tar格式) 导入到testdb数据库。

```
gs_restore backup/MPPDB_backup.tar -p 8000 -d testdb
restore operation successful
total time: 21203 ms
```

示例3: 执行gs\_restore, 将导出的MPPDB\_backup文件 (目录格式) 导入到testdb数据库。

```
gs_restore backup/MPPDB_backup -p 8000 -d testdb
restore operation successful
total time: 21003 ms
```

示例4: 执行gs\_restore, 使用自定义归档格式的MPPDB\_backup.dmp文件来进行如下导入操作。导入PUBLIC模式下所有对象的定义和数据。在导入时会先删除已经存在的对象, 如果原对象存在跨模式的依赖则需手工强制干预。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -n PUBLIC
Error while PROCESSING TOC:
Error from TOC entry 313; 1259 337399 TABLE table1 gaussdba
could not execute query: ERROR: cannot drop table table1 because other objects depend on it
DETAIL: view t1.v1 depends on table table1
HINT: Use DROP ... CASCADE to drop the dependent objects too.
Command was: DROP TABLE IF EXISTS public.table1;
```

手工删除依赖, 导入完成后再重新创建。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -n PUBLIC
restore operation successful
total time: 2203 ms
```

示例5: 执行gs\_restore, 使用自定义归档格式的MPPDB\_backup.dmp文件来进行如下导入操作。只导入PUBLIC模式下表table1的定义。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -c -s -n PUBLIC -t table1
restore operation successful
total time: 21000 ms
```

示例6: 执行gs\_restore, 使用自定义归档格式的MPPDB\_backup.dmp文件来进行如下导入操作。只导入PUBLIC模式下表table1的数据。

```
gs_restore backup/MPPDB_backup.dmp -p 8000 -d testdb -e -a -n PUBLIC -t table1
restore operation successful
total time: 20203 ms
```

## 相关命令

[gs\\_dump](#), [gs\\_dumpall](#)