This part contains reference information for PostgreSQL client applications and utilities. Not all of these commands are of general utility; some might require special privileges. The common feature of these applications is that they can be run on any host, independent of where the database server resides.

When specified on the command line, user and database names have their case preserved — the presence of spaces or special characters might require quoting. Table names and other identifiers do not have their case preserved, except where documented, and might require quoting.

D:\files\opensource\postgresql\package\postgresql-10.9-1-linux-x64-binaries\pgsql\doc\postgresql\html\reference-client.html

# Clusterdb 重新聚族已有索引

cluster a PostgreSQL database

clusterdb is a utility for reclustering tablesin a PostgreSQL database. It finds tablesthat have previously been clustered, and clusters them again on the sameindex that was last used. Tables that have never been clustered are notaffected.

clusterdb is a wrapper around the SQLcommand CLUSTER.There is no effective difference between clustering databases viathis utility and via other methods for accessing the server.

To cluster the database test:

|  |
| --- |
| $ clusterdb test |

To cluster a single table foo in a database named xyzzy:

|  |
| --- |
| $ clusterdb --table=foo xyzzy |

可以用对应的SQL：

Cluster the table employees on the basis ofits index employees\_ind:

|  |
| --- |
| CLUSTER employees USING employees\_ind; |

Cluster the employees table using the sameindex that was used before:

|  |
| --- |
| CLUSTER employees; |

Cluster all tables in the database that have previously been clustered:

|  |
| --- |
| CLUSTER; |

# Createdb 创建数据库，可以根据模板数据库进行创建

create a new PostgreSQL database

To create the database demo using the defaultdatabase server:

|  |
| --- |
| $ createdb demo |

To create the database demo using theserver on host eden, port 5000, using the template0 template database, here is thecommand-line command and the underlying SQL command:

|  |
| --- |
| $ createdb -p 5000 -h eden -T template0 -e demo |

CREATE DATABASE demo TEMPLATE template0;

使用SQL创建

|  |
| --- |
| CREATE DATABASE *name*  [ [ WITH ] [ OWNER [=] *user\_name* ]  [ TEMPLATE [=] *template* ]  [ ENCODING [=] *encoding* ]  [ LC\_COLLATE [=] *lc\_collate* ]  [ LC\_CTYPE [=] *lc\_ctype* ]  [ TABLESPACE [=] *tablespace\_name* ]  [ ALLOW\_CONNECTIONS [=] *allowconn* ]  [ CONNECTION LIMIT [=] *connlimit* ]  [ IS\_TEMPLATE [=] *istemplate* ] ] |

# createuser

define a new PostgreSQL user account

To create a user joe on the default databaseserver:

|  |
| --- |
| $ createuser joe |

To create a user joe on the default databaseserver with prompting for some additional attributes:

|  |
| --- |
| $ createuser --interactive joe  Shall the new role be a superuser? (y/n) n  Shall the new role be allowed to create databases? (y/n) n  Shall the new role be allowed to create more new roles? (y/n) n |

To create the same user joe using theserver on host eden, port 5000, with attributes explicitly specified,taking a look at the underlying command:

|  |
| --- |
| $ createuser -h eden -p 5000 -S -D -R -e joe  CREATE ROLE joe NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN; |

To create the user joe as a superuser,and assign a password immediately:

|  |
| --- |
| $ createuser -P -s -e joe  Enter password for new role: xyzzy  Enter it again: xyzzy  CREATE ROLE joe PASSWORD 'md5b5f5ba1a423792b526f799ae4eb3d59e' SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN; |

In the above example, the new password isn't actually echoed when typed,but we show what was typed for clarity. As you see, the password isencrypted before it is sent to the client.

# dropdb

remove a PostgreSQL database

To destroy the database demo on the defaultdatabase server:

$ dropdb demo

To destroy the database demo using theserver on host eden, port 5000, with verification and a peekat the underlying command:

$ dropdb -p 5000 -h eden -i -e demo

Database "demo" will be permanently deleted.

Are you sure? (y/n) y

DROP DATABASE demo;

# dropuser

remove a PostgreSQL user account

To remove user joe from the default databaseserver:

$ dropuser joe

To remove user joe using the server on host eden, port 5000, with verification and a peek at the underlyingcommand:

$ dropuser -p 5000 -h eden -i -e joe

Role "joe" will be permanently removed.

Are you sure? (y/n) y

DROP ROLE joe;

# ecpg

embedded SQL C preprocessor

If you have an embedded SQL C source file named prog1.pgc, you can create an executableprogram using the following sequence of commands:

ecpg prog1.pgc

cc -I/usr/local/pgsql/include -c prog1.c

cc -o prog1 prog1.o -L/usr/local/pgsql/lib -lecpg

# pg\_basebackup

take a base backup of a PostgreSQL cluster

To create a base backup of the server at mydbserverand store it in the local directory /usr/local/pgsql/data:

$ pg\_basebackup -h mydbserver -D /usr/local/pgsql/data

To create a backup of the local server with one compressedtar file for each tablespace, and store it in the directory backup, showing a progress report while running:

$ pg\_basebackup -D backup -Ft -z -P

To create a backup of a single-tablespace local database and compressthis with bzip2:

$ pg\_basebackup -D - -Ft -X fetch | bzip2 > backup.tar.bz2

(This command will fail if there are multiple tablespaces in thedatabase.)

To create a backup of a local database where the tablespace in /opt/ts is relocatedto ./backup/ts:

$ pg\_basebackup -D backup/data -T /opt/ts=$(pwd)/backup/ts

# pgbench

run a benchmark test on PostgreSQL

# pg\_config

retrieve information about the installed version of PostgreSQL

To reproduce the build configuration of the current PostgreSQLinstallation, run the following command:

eval ./configure `pg\_config --configure`

The output of pg\_config --configure containsshell quotation marks so arguments with spaces are representedcorrectly. Therefore, using eval is requiredfor proper results.

显示编译的时候的配置参数：

|  |
| --- |
| [root@centos7 ~]# /opt/pgsql/bin/pg\_config --configure  '--with-icu' '--enable-debug' '--with-libs=/opt/local/Current/lib' '--with-includes=/opt/local/Current/include/libxml2:/opt/local/Current/include' '--prefix=/mnt/hgfs/pginstaller.auto/server/staging\_cache/linux-x64' '--with-ldap' '--with-openssl' '--with-perl' '--with-python' '--with-tcl' '--with-tclconfig=/opt/local/edb/languagepack-10/Tcl-8.6/lib' '--with-pam' '--enable-thread-safety' '--with-libxml' '--with-ossp-uuid' '--docdir=/mnt/hgfs/pginstaller.auto/server/staging\_cache/linux-x64/doc/postgresql' '--with-libxslt' '--with-libedit-preferred' '--with-gssapi' 'LD\_LIBRARY\_PATH=/opt/local/Current/lib' 'CFLAGS=-O2 -DMAP\_HUGETLB=0x40000' 'ICU\_CFLAGS=-I/opt/local/Current/include' 'ICU\_LIBS=-L/opt/local/Current/lib -licuuc -licudata -licui18n' |

# pg\_dump

extract a PostgreSQL database into a script file or other archive file

dump成sql文件

To dump a database called mydb into a SQL-script file:

$ pg\_dump mydb > db.sql

To reload such a script into a (freshly created) database named newdb:

$ psql -d newdb -f db.sql

To dump a database into a custom-format archive file:

$ pg\_dump -Fc mydb > db.dump

To dump a database into a directory-format archive:

$ pg\_dump -Fd mydb -f dumpdir

To dump a database into a directory-format archive in parallel with5 worker jobs:

$ pg\_dump -Fd mydb -j 5 -f dumpdir

To reload an archive file into a (freshly created) database named newdb:

$ pg\_restore -d newdb db.dump

To dump a single table named mytab:

$ pg\_dump -t mytab mydb > db.sql

To dump all tables whose names start with emp in the detroit schema, except for the table named employee\_log:

$ pg\_dump -t 'detroit.emp\*' -T detroit.employee\_log mydb > db.sql

To dump all schemas whose names start with east or west and end in gsm, excluding any schemas whosenames contain the word test:

$ pg\_dump -n 'east\*gsm' -n 'west\*gsm' -N '\*test\*' mydb > db.sql

The same, using regular expression notation to consolidate the switches:

$ pg\_dump -n '(east|west)\*gsm' -N '\*test\*' mydb > db.sql

To dump all database objects except for tables whose names begin with ts\_:

$ pg\_dump -T 'ts\_\*' mydb > db.sql

To specify an upper-case or mixed-case name in -t and relatedswitches, you need to double-quote the name; else it will be folded tolower case (see Patterns). Butdouble quotes are special to the shell, so in turn they must be quoted.Thus, to dump a single table with a mixed-case name, you need somethinglike

$ pg\_dump -t "\"MixedCaseName\"" mydb > mytab.sql

# pg\_dumpall

extract a PostgreSQL database cluster into a script file

To dump all databases:

$ pg\_dumpall > db.out

To reload database(s) from this file, you can use:

$ psql -f db.out postgres

(It is not important to which database you connect here since thescript file created by pg\_dumpall willcontain the appropriate commands to create and connect to the saveddatabases.)

# pg\_isready 看看数据库是否准备好

check the connection status of a PostgreSQL server

Standard Usage:

$ pg\_isready

/tmp:5432 - accepting connections

$ echo $?

0

Running with connection parameters to a PostgreSQL cluster in startup:

$ pg\_isready -h localhost -p 5433

localhost:5433 - rejecting connections

$ echo $?

1

Running with connection parameters to a non-responsive PostgreSQL cluster:

$ pg\_isready -h someremotehost

someremotehost:5432 - no response

$ echo $?

# pg\_receivewal 获取数据库的write-ahead日志

stream write-ahead logs from a PostgreSQL server

To stream the write-ahead log from the server at mydbserver and store it in the local directory /usr/local/pgsql/archive:

$ pg\_receivewal -h mydbserver -D /usr/local/pgsql/archive

# pg\_recvlogical

control PostgreSQL logical decoding streams

# pg\_restore 回复数据库

restore a PostgreSQL database from an archive file created by pg\_dump

Assume we have dumped a database called mydb into acustom-format dump file:

$ pg\_dump -Fc mydb > db.dump

To drop the database and recreate it from the dump:

$ dropdb mydb

$ pg\_restore -C -d postgres db.dump

The database named in the -d switch can be any database existingin the cluster; pg\_restore only uses it to issue the CREATE DATABASE command for mydb. With -C, data is always restored into the database name that appearsin the dump file.

To reload the dump into a new database called newdb:

$ createdb -T template0 newdb

$ pg\_restore -d newdb db.dump

Notice we don't use -C, and instead connect directly to thedatabase to be restored into. Also note that we clone the new databasefrom template0 not template1, to ensure it isinitially empty.

To reorder database items, it is first necessary to dump the table ofcontents of the archive:

$ pg\_restore -l db.dump > db.list

The listing file consists of a header and one line for each item, e.g.:

;

; Archive created at Mon Sep 14 13:55:39 2009

; dbname: DBDEMOS

; TOC Entries: 81

; Compression: 9

; Dump Version: 1.10-0

; Format: CUSTOM

; Integer: 4 bytes

; Offset: 8 bytes

; Dumped from database version: 8.3.5

; Dumped by pg\_dump version: 8.3.8

;

;

; Selected TOC Entries:

;

3; 2615 2200 SCHEMA - public pasha

1861; 0 0 COMMENT - SCHEMA public pasha

1862; 0 0 ACL - public pasha

317; 1247 17715 TYPE public composite pasha

319; 1247 25899 DOMAIN public domain0 pasha

Semicolons start a comment, and the numbers at the start of lines refer to theinternal archive ID assigned to each item.

Lines in the file can be commented out, deleted, and reordered. For example:

10; 145433 TABLE map\_resolutions postgres

;2; 145344 TABLE species postgres

;4; 145359 TABLE nt\_header postgres

6; 145402 TABLE species\_records postgres

;8; 145416 TABLE ss\_old postgres

could be used as input to pg\_restore and would only restoreitems 10 and 6, in that order:

$ pg\_restore -L db.list db.dump

# psql

PostgreSQL interactive terminal

# reindexdb

reindex a PostgreSQL database

To reindex the database test:

$ reindexdb test

To reindex the table foo and the index bar in a database named abcd:

$ reindexdb --table=foo --index=bar abcd

# vacuumdb 垃圾收集

garbage-collect and analyze a PostgreSQL database

To clean the database test:

$ vacuumdb test

To clean and analyze for the optimizer a database named bigdb:

$ vacuumdb --analyze bigdb

To clean a single table foo in a database named xyzzy, and analyze a single column bar of the table for the optimizer:

$ vacuumdb --analyze --verbose --table='foo(bar)' xyzzy