

PostgreSQL

O PostgreSQL quando instalado cria um usuário chamado "postgres", que possui uma shell com permissões totais de gerenciamento do banco de dados. Como "root", digite:

```
# su - postgres
```

Agora você possui uma série de comandos que podem ser usados para gerenciar o PostgreSQL:

- pg_dumpall
- psql
- createuser
- dropdb
- dropuser
- clusterdb
- vacuumdb
- createdb
- pg_dump
- pg_restore
- readpgenv
- psql-wm

psql

Nome

psql -- terminal interativo do PostgreSQL

Sinopse

psql [opção...] [nome_do_banco_de_dados [nome_do_usuario]]

Descrição

O psql é um cliente no modo terminal do PostgreSQL. Permite digitar comandos interativamente, submetê-los para o PostgreSQL e ver os resultados. Como alternativa, a entrada pode vir de um arquivo. Além disso, disponibiliza vários meta-comandos e diversas funcionalidades semelhantes às do interpretador de comandos (shell) para facilitar a criação de scripts e automatizar um grande número de tarefas.

Para entrar no shell psql a partir do prompt faça:

```
cd C:\Program Files\PostgreSQL\9.1\bin
```

```
C:\Program Files\PostgreSQL\9.1\bin> psql -U nome_do_usuario -d nome_do_banco
```

Forma geral:

psql -h host -P port -U user -W (perguntar pela senha)

Agora como exemplo veremos alguns comandos do psql partiremos do principio que você esteja no banco de dados de nome "apolo" após efetuar os comandos acima.

Acessando o banco de dados de nome apolo

psql -U postgres -d apolo

Obs.:

apolo → banco de dados

postgres → usuario

Listar todos os bancos de dados

apolo=# \l

Sair do console do psql

apolo=# \q

Listar usuarios

apolo=# \dg

Listar tabelas do banco de dados

apolo=# \d

Listar uma tabela especifica do banco de dados

apolo=# TABLE nome_da_tabela;

Criando dump do banco de dados do apolo no Windows

cd C:\Program Files\PostgreSQL\9.1\bin

C:\Program Files\PostgreSQL\9.1\bin>

C:\Program Files\PostgreSQL\9.1\bin>pg_dump.exe -U postgres -Fc apolo > apolo.dump

Resumo:

C:\Program Files\PostgreSQL\9.1\bin>pg_dump.exe -U nome_do_usuario -Fc nome_do_banco.dump > nome_do_banco

-F: tipo de dump (que pode ser c,t de tar)

Obs.:

Para que o comando acima seja executado será necessário executar como Administrador ou o seu usuário terá que ter as mesmas permissões que o Administrador. Uma solução para isto seria executar o prompt da seguinte forma:

Iniciar > prompt > botão direito sobre o ícone > executar como administrador

Fora do console psql, ou seja, você esta no prompt mas executando comandos deste shell

psql -U nomeusuario -d nomebanco -> conectar à console psql no banco de dados

Exemplo:

C:\Program Files\PostgreSQL\9.1\bin>psql -U postgres -d apolo

psql -l -> lista os bancos de dados

Exemplo:

C:\Program Files\PostgreSQL\9.1\bin>psql -U postgres -l

psql -version -> mostra versão do PostgreSQL

Exemplo:

C:\Program Files\PostgreSQL\9.1\bin>psql -U postgres -version

Comandos no Linux

Acesse o servidor de banco:

gutocarvalho@destiny:~\$ ssh pgsq01 -l gutocarvalho

Torne-se root

gutocarvalho@pgsq01:~\$ sudo -i

Torne-se postgresql

root@pgsql:~\$ su postgres

Acesse o shell do postgres

postgres@pgsq01:~\$ psql

Agora estamos no shell do psql e no banco postgres

postgres=#

Agora estamos no shell do psql e no banco apolo

apolo=#

Agora estamos no shell do psql e no banco agencia itau

agencia_itau=#

Criando usuario com senha

postgres=# CREATE USER fernandao WITH PASSWORD 'senha';

Lista usuários existentes, as permissões e grupos que o usuário pertence

postgres=# \dg

Criando o banco

postgres=# CREATE DATABASE eleicoes_carga;

Excluir o banco criado

postgres=# DROP DATABASE nome_do_banco ;

Ajuda sobre os comandos:

postgres=# nome_comando - -help

Mudando de um banco para outro

postgres=# \c apolo

Resultado:

apolo=#

Verificando versão

psql -U postgres -version

Dando permissões ao usuário para acessar o banco

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE projeto_carga to fernandao;
```

Adicionando permissões ao usuário para acessar o banco

```
postgres=# ALTER USER fernandao SUPERUSER;
```

```
postgres=# ALTER USER fernandao CREATEDB;
```

```
postgres=# ALTER USER fernandao CREATEROLE
```

```
postgres=# CREATE USER aluno <aperte a tecla tab 2X, para autocomplete>;
```

Alterando owner do banco

```
postgres=# ALTER DATABASE projeto_carga OWNER TO fernandao;
```

Liberando acesso de uma maquina ao banco

após isto basta liberar o ip o cara no pg_hba.conf, veja um exemplo:

```
# IPv4 local connections:
host      all             all             127.0.0.1/32          md5
host      all             all             172.16.10.20/32       md5 # gandalf
```

Configurando postgres para listar em todas as interfaces

Tenha certeza que o postgresql está rodando em todas as interfaces, veja isso em postgresql.conf

```
listen_addresses = '0.0.0.0'          # what IP address(es) to listen on;
```

Será necessário reiniciar o banco.

dump

Dump de um banco inteiro - formato sql puro

```
pg_dump -U usuario -h ip_servidor nomedobanco > nomedobanco.sql
```

Dump de um banco inteiro - formato binario

pg_dump -U usuario -h ip_servidor -Fc -f banco.dump nome_do_banco

pg_dump -U usuario -Fc nome_do_banco.dump > nome_do_banco

Dump de uma tabela do banco

pg_dump -U usuario -h ip_servidor -Fc -t tabela -f tabela.pgdump nome_do_banco

Dump de um tabela de um schema

pg_dump -U postgres -h 127.0.0.1 -Fc -n schema -t tabela -f tabela.pgdump banco

restore

Restaurar um banco com dump (Estando como root)

psql -U nome_de_usuario -d nome_do_banco -f /nome_do_arquivo_de_dump.dump

Exemplo:

psql -U postgresql -d apolo -f /apolo_2013_10_14.dump

Restaurar um banco com dump em formato SQL

psql -U usuario -h ip_do_servidor -f nomeodabanco.sql nomedobanco

Restaurar um banco com dump em formato binário

pg_restore -U usuario -h ip_servidor -d nome_do_banco arquivo.dump

Restaurar somente os dados:

pg_restore -U usuario -h ip_servidor -a -d nome_do_banco arquivo.dump

Restaurar somente um determinado esquema

pg_restore -U usuario -h ip_servidor -n meu_esquema -d nome_do_banco arquivo.dump

Restaurar apenas os dados de uma tabela especifica

pg_restore -U usuario -h ip_servidor -a -t tabela_especifica -d nome_do_banco arquivo.dump

exemplos

Fazendo dump de tabela da radioagencia

```
pg_dump -Fc -U drupal -h 127.0.0.1 -t drupal_webform -f arquivo -d drupal drupal_webform
```

Restaurando tabela da radioagencia

```
pg_restore -Fc -U drupal -h 127.0.0.1 -t drupal_webform -d drupal drupal_webform
```

Observe o formato do DUMP, ele precisa ser tipo C ou tipo F para que o pg_restore consiga recuperar, se for plain text vai dar erro.

Todo administrador de sistemas pelo menos uma vez ficou com aquele servidor encostado com meia dúzia de clientes que acabaram sendo esquecidos entre várias migrações.

E é aí que começa o problema, saíram várias versões novas do postgresql e o dump que você faz no servidor antigo não restaura no novo... o que fazer?

Não estou dizendo que isto irá funcionar em todos os casos, mas em 2 bombas que eu encarei por aqui hoje pelo menos funcionou.

No momento do dump, utilize o usuário postgres (ou equivalente, com nível de administrador, se rodar com usuário restrito pode dar erro na hora de importar alguma coisa – pelo menos comigo deram vários). E sete os seguintes parâmetros:

pg_dump -U postgres -F t -x -i -O -v -f -W arquivodebackup.tar -d bancodedados

Distinguindo os parâmetros:

-U: usuário

-F: tipo de dump (neste caso t de tar, foi a única forma de resolver os problemas de caracteres)

-x: não envia informações de privilégios (grant/revoke)

-i: ignora versão

-O: não envia dados do owner

-v: verbose, caso dê algum erro no momento de gerar você vê na tela.

-f: arquivo de destino

-d: banco de dados

-W: força o pedido senha

Para fazer o restore utilize obrigatoriamente o usuário que irá utilizar o banco posteriormente, pois lembre-se no momento do backup e do restore mandamos ignorar estas informações.

pg_restore -d bancodedados -F t -i -v -O -x -U usuario -W