

Pasos para empezar a usar PostgreSQL instalado en el ldc

1. Ingrese al ldc mediante un terminal remoto

a) Si está en Linux

- `ssh <USERNAME>@<HOSTNAME>`
- `<PASSWORD>`

b) Si está en Windows

- SSH Secure Shell Client
- Quick Connet
- Host Name <HOSTNAME>
- User Name <USERNAME>
- Connect
- Password <PASSWORD>
- Ok

2. En el terminal remoto

- Ingrese a PostgreSQL con la base de datos global “postgres”
 - `psql -h uxtal ldc.usb.ve “postgres” “<PGUSR>”`
 - `<PGPASS>`
- Dentro de PostgreSQL, cree su propia base de datos
 - `CREATE DATABASE “<PGDBNAME>” OWNER “<PGUSR>”;`
- Salga de PostgreSQL
 - `\q`
- Vuelva a entrar a PostgreSQL, ahora con su propia base de datos
 - `psql -h uxtal ldc.usb.ve “<PGDBNAME>” “<PGUSR>”`
- Cambie su password en PostgreSQL
 - `ALTER USER “<PGUSR>” WITH PASSWORD ‘<NUEVOPASSWORD>’;`
 -
- Salga de PostgreSQL
 - `\q`
- A partir de ahora puede usar PostgreSQL con su base de datos y su nuevo password

Cómo transferir los scripts entre su máquina local y su cuenta en el ldc

Sus scripts de comandos en SQL debe editarlos en la máquina local, usando el editor de código que tenga disponible, como puede ser notepad++ o gedit o emacs, o cualquier otro de su preferencia. Para correr estos scripts en PostgreSQL debe transferirlos a su cuenta en el ldc.

b) Si está en Linux

- Hacia el ldc desde la máquina local
 - `scp <ARCHIVO> <USERNAME>@<HOSTNAME>:<PATHREMOTO>`
- Desde el ldc hacia la máquina local
 - `scp <USERNAME>@<HOSTNAME>:<ARCHIVO> <PATHLOCAL>`

c) Si está en Windows

- SSH Secure File Transfer Client
- Quick Connet
- Host Name <HOSTNAME>
- User Name <USERNAME>
- Connect
- Password <PASSWORD>
- Ok
- puede copiar de una máquina a otra por interfaz gráfica

Algunos comandos útiles de PostgreSQL

- \h Ayuda con los comandos SQL.
- \? Ayuda para los comandos psql, de “slash” internos.
- \q Salir.
- \l: Listar las bases de datos.
- \c nombre_basedatos: conectar a nombre_base-datos.
 - ejemplo: \c postgres
- \d : Listar las tablas pertenecientes a la base de datos.
- \d nombre_tabla : Describe la table nombre_tabla.
 - ejemplo: \d detencion
- \i **nombre_archivo**
 - Lee las entradas del archive **nombre_archivo** y las ejecutas como si hubiesesido ingresadas por consola.
- \p
 - Imprime la consulta que está actualmente en el buffer usando la salida estándar.
- \s [**nombre_archivo**]
 - Imprime o guarda todo el historial de comandos en nombre_archivo.
- \w { **nombre_archivo** | **nombre_comando** }
 - Extrae la consulta que se encuentra actualmente el buffer hacia el **nombre_archivo** o lo envía mediante pipe a **nombre_comando** perteneciente a los comandos de Unix.
- \! [**nombre_comando**]
 - Ejecuta el **comando** perteneciente a los comandos de Unix. Ejemplo: \! ls -la
- \o **nombre_archivo**;
 - Almacena el resultado de las consultas en un archivo.
 - Ejemplo: si se quiere almacenar la fecha de hoy en el archivo resultado.log y luego visualizar el contenido de este archivo
 - \o resultado.log;
 - *SELECT now();*
 - \! cat resultado.log;
 - si deseamos redireccionar los resultados de las consultas hacia la consola de nuevo, debemos ejecutar \o sin argumentos.
- \s [**nombre_archivo**]
 - Imprime o guarda todo el historial de comandos en nombre_archivo.
 - Ejemplo:
 - \s comandos.log
 - Si no se borra el buffer, el comando \s almacenará todo el historial completo, incluyendo aquellos comandos ejecutados en una conexión anterior, si se desea evitar esto, es necesario limpiar el buffer
- \r Limpiar o reiniciar el buffer