
Prácticas de FBD con PostgreSQL

Índice

TODO Introducción	3
Instalación de PostgreSQL	3
Arch Linux/derivados	3
Ubuntu/derivados	3
Fedora/derivados	3
macOS	3
Documentación general	4
Manejo básico.	4
Creación de usuario y base de datos	4
Manejo de <code>psql</code>	4
Tutorial	5

TODO Introducción

Instalación de PostgreSQL

Puesto que el objeto de esta guía es evitar el uso de *software* privativo para actividades académicas, se asume que estás usando tu distribución Linux o SO libre favorito.

Se proporciona información para instalar PostgreSQL en algunos sistemas.

Arch Linux/derivados

Existe un paquete oficial, llamado `postgresql`. La [wiki de Arch](#) tiene la documentación necesaria para una configuración básica.

Ubuntu/derivados

También existe una [guía oficial de Ubuntu](#).

Fedora/derivados

Para Fedora se puede seguir [guía para instalar en Fedora](#), teniendo en cuenta de que solo hace falta llegar hasta el 2.4; cambiando de usuario a su postgres ya se puede usar `psql` sin problema.

macOS

Si por desgracia utilizas un sistema operativo privativo como es macOS, puedes utilizar `homebrew` para instalar PostgreSQL.

Si no tienes `homebrew` instalado, puedes hacerlo con el comando

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Una vez que tengas `homebrew`, instalar PostgreSQL es tan sencillo como hacer

```
brew update  
brew install postgresql
```

Para ejecutar el servidor de PostgreSQL utilizamos el comando

```
postgres -D /usr/local/var/postgres
```

Documentación general

Disponible en la [página de PostgreSQL](#).

Nota: el enlace anterior lleva a la documentación de la versión 9.6.x, hay enlaces equivalentes para todas las versiones. Puedes comprobar la versión que tienes instalada con la orden

```
postgres --version
```

Manejo básico.

Creación de usuario y base de datos

Si siguiendo alguna de las guías anteriores no has creado el usuario `postgres` en tu sistema, hazlo ahora. Tras ello, identifícate con ese usuario para tener acceso a las órdenes de administración del SGBD:

```
useradd postgres  
su - postgres
```

A continuación, crea un usuario de Postgres y base de datos:

```
createuser --interactive  
createdb database_name
```

Manejo de `psql`

Cuando nos identificamos con el usuario `postgres` tenemos acceso a la orden `psql`, que es una línea de comandos interactiva para introducir órdenes al SGBD.

Nota: `psql` no evalúa las órdenes hasta que encuentra un punto y coma. Si se pulsa *enter* antes de escribir un punto y coma, asumirá que se quiere escribir una orden en varias líneas.

Nota 2: en mi experiencia, `psql` no guarda el historial de órdenes entre sesiones. Para habilitar esa funcionalidad, tuve que crear el archivo `.psql_history` en el directorio *home* del usuario `postgres`, y hacer que dicho archivo pertenezca a ese usuario con `chown`.

Una vez que hemos comprobado que `psql` está funcionando, podemos optar por seguir el tutorial, como se describe en la sección siguiente.

Creación de tablas

La orden para crear tablas tiene la siguiente sintaxis básica:

```
CREATE TABLE <nombre_table> (  
  <atributo1> <tipo1>,  
  <atributo2> <tipo2>,  
  ...,  
)
```

Descripción de una tabla

Para obtener la descripción de una tabla, podemos usar la orden

```
\d+ <nombre_tabla>
```

Tutorial

PostgreSQL incorpora unos archivos de órdenes SQL por defecto. Están disponibles en la [distribución de código fuente](#). Para usarlo, descargamos la distribución de código fuente correspondiente a nuestra versión, la descomprimos y dentro del directorio descomprimido, nos vamos a `src/tutorial` y ejecutamos `make`.

Ejecutamos

```
psql -s <nombredb>
```

donde la base de datos `nombredb` ha sido creada anteriormente. La opción `-s` activa el modo *single-step*, que muestra la orden que se va introducir para su confirmación antes de enviarla al servidor.

En `psql`, ejecutamos `\i basics.sql`, lo cual comienza el tutorial. Además, hay otros archivos `.sql` que podemos ejecutar cuando terminemos con este.