

ADMINISTRACIÓN DE BASES DE DATOS



PostgreSQL

Fernando Zaragoza Hernández
fzaragozacu@gmail.com

Agenda



- ***Introducción***
- ***Instalación***
- ***Preparación de la Base de Datos***
- ***Inicio del servidor***
- ***Apagado del servidor***
- ***Usuarios y privilegios***
- ***Manejo de Bases de Datos***



Introducción



- ***Características principales de PostgreSQL***

- Open Source
- Licencia BSD (modificar, copiar y distribuir para cualquier propósito)
- Última versión 8.4.5 (9.0.0 – 9.0.1 – 9.4.4)
- Sitio oficial: <http://www.postgresql.org/>

- ***Límites de las bases de datos***

- Tamaño para una BD = ilimitado (existen BD de 32 TB)
- Tamaño de una tabla = 32 TB
- Tamaño para un registro = 400 GB
- Tamaño para un campo = 1GB
- Número de registros en una tabla = ilimitado
- Máximo de columnas en una tabla = 250 - 1600 (dependiendo de los tipos de columna)
- Número de índices en una tabla = ilimitado



Introducción



- ***Base de datos de Posgrado (Agosto 2015)***
 - Postgres 9.1.15
 - Código de caracteres SQL ASCII
 - 220 000 registros de alumnos
 - 2 831 000 registros de historias académicas



Instalación



1. **Descargar de: <http://www.postgresql.org>**
2. **Ejecutar los siguientes comandos:**

```
# groupadd postgres
# useradd -g postgres postgres
# tar -zxvf /home/instalacion/postgresql-x.x.x.tar.gz
# ln -s postgresql-8.x.x postgresql
```



Instalación (cont.)



```
# cd postgresql
# ./configure --prefix=/usr/local/pgsql
# make
# make install
# mkdir /usr/local/pgsql/data
# chown postgres /usr/local/pgsql/data
```





Instalación (cont.)

```
# su - postgres
# PATH=$PATH:/usr/local/pgsql/bin
# export POSTGRES_HOME=/usr/local/pgsql
# export PGDATA=/usr/local/pgsql/data
# export PGLIB=/usr/local/pgsql/lib
# export LD_LIBRARY_PATH=/usr/local/pgsql/lib
```





Preparación de la Base de datos

```
# /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

Opciones de *initdb*

- ***[-D, --pgdata=]DATADIR*** *location for this database cluster*
- ***-E, --encoding=ENCODING*** *set default encoding for new databases*
- ***--locale=LOCALE*** *initialize database cluster with given locale*
- ***--lc-collate, --lc-ctype, --lc-messages=LOCALE***
- ***--lc-monetary, --lc-numeric, --lc-time=LOCALE***



Inicio del servidor



```
# /usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start
```

```
# /usr/local/pgsql/bin/pg_ctl -l /usr/local/pgsql/data/errors.log  
-D /usr/local/pgsql/data/ -o -i start
```

```
# /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data
```





Inicio del servidor (cont.)

- ***Para verificar que el servidor se encuentra en operación, tenemos las siguientes opciones:***
 - Verificando el procesos en ejecución de *postmaster/postgres*
 - Verificando el puerto abierto
 - Verificando las bases de datos del servidor:
`#psql -l`



Apagado del servidor



- *Existen diferentes formas para detener el servidor.*
- *Utilizando el comando `pg_ctl` con la opción `stop`*
- *Matando el proceso de `postmaster/postgres` (no recomendado)*



Práctica 1



Manos a la máquina!!!



Usuarios y privilegios



- *Creación de usuarios*

Para crear un usuario se utiliza el comando SQL CREATE USER nombre_usuario

```
template1=# \h create user  
CREATE USER user_name  
  [WITH PASSWORD password]  
  [CREATEDB | NOCREATEDB]  
  [CREATEUSER | NOCREATEUSER]  
  [IN GROUP group1, ...groupN]  
  [VALID UNTIL 'abstime'];
```





Usuarios y privilegios (cont.)

- *También es posible crear un usuario desde la línea de comandos*
createuser

Opciones:

-h, --host host

Especifica el nombre del host de la máquina sobre la que el postmaster corre.

-p, --puerto puerto

Especifica el puerto TCP/IP o el socket local Unix sobre el que el postmaster atiende a las conexiones.

-e, --echo

Muestra las consultas que createuser genera y envía al backend.

-q, --quiet

No muestra respuesta alguna.

-d, --createdb

Permite al nuevo usuario crear bases de datos.





Usuarios y privilegios (cont.)

-D, --no-createdb

Impide al nuevo usuario crear bases de datos.

-a, --adduser

Permite al nuevo usuario crear otros usuarios.

-A, --no-adduser

Impide al nuevo usuario crear otros usuarios.

-P, --pwprompt

Si se especifica este parámetro, createuser mostrará un mensaje preguntando por el password del nuevo usuario. Esto no es necesario si no planea usar autenticación por password.

-i, --sysid id_usuario

Le permite elegir otro id de usuario que no sea el que se da por defecto. Esto no es necesario, pero algunos lo prefieren.



Usuarios y privilegios (cont.)

- **Modificación de usuarios**

Para cambiar la contraseña y los privilegios de un usuario se utiliza el comando ALTER USER

Descripción

Se usa para cambiar los atributos de la cuenta de un usuario de PostgreSQL. Sólo un superusuario de una base de datos puede cambiar privilegios y fechas de caducidad de passwords con esta orden. Ordinariamente los usuarios sólo pueden cambiar su propia palabra clave.

ALTER USER user_name

[WITH PASSWORD 'password']

[CREATEDB | NOCREATEDB] [CREATEUSER | NOCREATEUSER]

[VALID UNTIL 'abstime']



Usuarios y privilegios (cont.)

- *Eliminación de usuarios*

DROP USER borra de la base de datos el usuario creado. Cabe aclarar que esta cláusula no borra tablas, vistas u otros objetos que pertenezcan al usuario. Si el usuario es dueño de una base de datos, se producirá un error.

DROP USER user_name;



Usuarios y privilegios (cont.)

- *Creación de grupos*

CREATE GROUP permite crear un grupo nuevo en la base de datos. Este comando solamente podrá ser ejecutado por un usuario administrativo.

CREATE GROUP name
[***WITH***
[***SYSID*** gid]
[***USER*** username [, ...]]]





Usuarios y privilegios (cont.)

- *Modificación de grupos*

ALTER GROUP se usa para añadir usuarios a un grupo o eliminarlos de un grupo. Sólo los administradores de bases de datos pueden usar este comando. Añadir un usuario a un grupo no crea ese usuario. Igualmente, eliminar a un usuario de un grupo no significa que se elimine al usuario en si mismo.

ALTER GROUP name ADD USER nombre de usuario1, ... ;
ALTER GROUP name DROP USER nombre de usuario1, ... ;



Usuarios y privilegios (cont.)



- *Eliminación de grupos*

DROP GROUP elimina el grupo especificado de la base de datos.
Los usuarios del grupo no se eliminan.

DROP GROUP nombre_grupo





Usuarios y privilegios (cont.)

- **Privilegios**

Cuando se crea un objeto en la base de datos, se le asigna un dueño. El dueño es el usuario que ejecutó la declaración de la creación. Para cambiar al dueño de una tabla, índice, secuencia, vista, se utiliza el comando ALTER TABLE. Por default, solo el dueño (o superusuario) puede hacer cualquier cosa con el objeto. Para permitir que otros usuarios lo utilicen, los privilegios deben ser concedidos (GRANT).

Hay varios privilegios : SELECT, INSERT, UPDATE, DELETE, RULE, REFERENCES, TRIGGER, CREATE, TEMPORARY, EXECUTE, USAGE, y ALL PRIVILEGES. El hecho de modificar o destruir un objeto es siempre el privilegio del dueño solamente, para ello se utiliza el comando GRANT.

Usuarios y privilegios (cont.)

- *Otorgamiento de privilegios*

GRANT → *otorga privilegios de acceso a un usuario, un grupo o a todos los usuarios.*

GRANT *privilege* [, ...] ***ON*** *object* [, ...]
TO { ***PUBLIC*** | ***GROUP*** *group* | *username* }

Nota:

psql \z *permite ver la información de los privilegios*



Usuarios y privilegios (cont.)

- *Revocación de privilegios*

REVOKE → *Revoca el privilegio de acceso a un usuario, a un grupo o a todos los usuarios*

REVOKE privilegio [, ...]

ON objeto [, ...]

FROM { PUBLIC | GROUP | nombre_usuario }



Práctica 2



Manos a la máquina!!!



Manejo de Bases de Datos



- *Creación de la base de datos*

CREATE DATABASE crea una nueva base de datos PostgreSQL. El creador pasa a ser el propietario de la nueva base de datos.

EJEMPLO:

Para crear una nueva base de datos:

template1=> create database mibase;

El comando createdb es un script shell construido y basado en el comando CREATE DATABASE.

createdb [options] dbname [descripcion]





Manejo de Bases de Datos (cont.)

El comando `createdb` es un script shell construido y basado en el comando `CREATE DATABASE`.

`createdb [options] dbname [descripcion]`

Opciones:

-h, --host host

-p, --port port

-U, --username username

-W, --password

Forza a que se teclee password.

-e, --echo

Muestra la consulta que `createdb` genera y envía al motor de la base de datos (backend)

-q, --quiet

No muestra ninguna respuesta.

-D, --location datadir

-O – Owner (dueño)

-E, --encoding encoding





Manejo de Bases de Datos (cont.)

- *Eliminación de la base de datos*

DROP DATABASE elimina las entradas de catálogo de una base de datos existente y borra el directorio que contiene los datos. Solamente puede ser ejecutado por el propietario de la base de datos (normalmente quien la creó).

DROP DATABASE name





Manejo de Bases de Datos (cont.)

- **Plantillas**

CREATE DATABASE realmente trabaja copiando una base de datos existente. Por default, copia la base de datos estándar del sistema llamada template1. Así esta base de datos es el "template" para crear nuevas bases de datos. Si nosotros agregamos objetos a la base de datos template1, estos objetos serán copiados en bases de datos creadas posteriormente por el usuario. Este comportamiento permite modificaciones al sistema estándar de objetos en bases de datos (site-local).





Manejo de Bases de Datos (cont.)

- ***Hay una segunda base de datos estándar del sistema llamada template0. template0 sin que nunca se cambie después initdb. Para dar la instrucción CREATE DATABASE para copiar template0 en ves de template1, podemos crear un usuario “virgen” en la base de datos que no contiene ningún site-local adicionado en template1. Esto es particularmente práctico al restaurar con pg_dump dump: el script dump debe restaurar una base de datos virgen para asegurarse de que reconstruya el contenido correcto de la base de datos descargada, sin ningún conflicto con las adiciones que pueden estar presentes en template1.***



Manejo de Bases de Datos (cont.)

Para crear una base de datos copiando template0 debemos usar:

Del ambiente SQL,

CREATE DATABASE dbname TEMPLATE template0;

o del shell:

#createdb -T template0 dbname



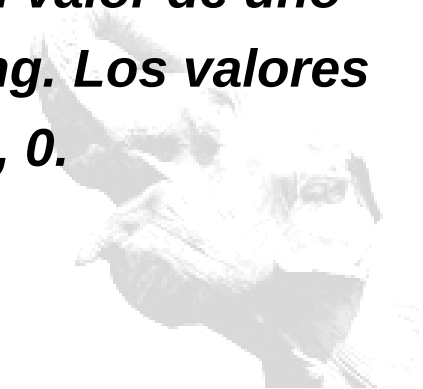


Manejo de Bases de Datos (cont.)

- *Configuración de la base de datos*

Existen muchos parámetros de configuración que afectan el comportamiento del sistema de base de datos.

Todos los nombres de los parámetros son sensibles a mayúsculas y minúsculas. Cada parámetro toma el valor de uno de estos cuatro tipos: boolean, integer, float y string. Los valores booleanos son ON, OFF, TRUE, FALSE, YES, NO, 1, 0.





Manejo de Bases de Datos (cont.)

- *Un forma de cambiar estos parámetros es modificar el archivo `postgresql.conf` en el directorio `data`. Un ejemplo de cómo luciría este archivo es:*

Comentario

`log_connections = yes`

`syslog = 2`

`search_path = '$user, public'`





Manejo de Bases de Datos (cont.)

- *Una segunda forma de establecer estos parámetros de configuración es pasarlos como una opción de postmaster a través de la línea de comandos:*

`postmaster -c log_connections=yes -c syslog=2`

Las opciones de la línea de comandos anulan cualquier ajuste existente en el archivo.





Manejo de Bases de Datos (cont.)

- ***Opciones de conexión***
 - tcpip_socket (booleano)
 - listen_addresses = 'localhost'
 - max_connections (integer)
 - superuser_reserved_connections (integer)
 - port (integer)
 - unix_socket_directory (string)
 - unix_socket_group (string)
 - unix_socket_permissions (integer)
 - virtual_host (string)





Manejo de Bases de Datos (cont.)

- ***Seguridad y autenticación***
 - authentication_timeout (integer)
 - ssl (boolean)
 - krb_server_keyfile (string)
 - db_user_namespace (boolean)





Manejo de Bases de Datos (cont.)

- **Memoria**
 - shared_buffers (integer)
 - sort_mem (integer)
 - vacuum_mem (integer)





Manejo de Bases de Datos (cont.)

- *Eliminación de una base de datos*

Se usa drop database para eliminar la base de datos.

DROP DATABASE → Elimina una base de datos existente

DROP DATABASE name





Manejo de Bases de Datos (cont.)

- ***Conjunto de Caracteres***
- ***El soporte local de caracteres se refiere al uso de preferencias culturales con respecto a los alfabetos, ordenamientos, formato de números, etc. PostgreSQL utiliza los estándares ISO C y POSIX proporcionadas por el sistema operativo del servidor.***





Manejo de Bases de Datos (cont.)

- ***El soporte local de caracteres se inicializa automáticamente cuando se crea un cluster de la base de datos usando initdb. Si se desea utilizar un soporte local de caracteres diferente, se puede indicar a initdb exactamente que soporte se desea con la opción -- locale. Por ejemplo:***

initdb -- locale=sv_SE

- ***Este ejemplo fija el soporte de caracteres local al sueco (sv).***
- ***Si más de un juego de caracteres locales es válido entonces las especificaciones podrían verse de la siguiente manera: cs_CZ.ISO8859-2.***



Manejo de Bases de Datos (cont.)

- De vez en cuando es útil mezclar reglas de varios juegos de caracteres, por ejemplo, utilizar las reglas inglesas de colocación para los mensajes en español. Para soportar esto, existe un juego de subcategorías que controlan ciertos aspectos de las reglas del soporte de caracteres.

LC_COLLATE	Cadena de ordenamiento
LC_CTYPE	Clasificación de los caracteres
LC_MESSAGES	Lenguaje de los mensajes
LC_MONETARY	Formato de las cantidades monetarias
LC_NUMERIC	Formato de números
LC_TIME	Formato de fechas y tiempos





Manejo de Bases de Datos (cont.)

- ***Por ejemplo, para fijar el juego de caracteres local al francés canadiense, pero usando las reglas de EUA para el formato de las cantidades monetarias initdb quedaría de la siguiente manera:***

initdb -- locale=fr_CA -- lc-monetary=en_US

- ***Si usted quisiera que el sistema se comportara como si no tuviera soporte de caracteres local, es necesario utilizar el juego de caracteres especial C o POSIX.***





Manejo de Bases de Datos (cont.)

- *Soporte de juego de caracteres*
- *El soporte de juego de caracteres en PostgreSQL permite almacenar texto en diferentes juegos de caracteres, incluyendo juegos de caracteres single-byte tales como la ISO 8859 y del multiple-byte tales como EUC (Extended Unix Code), Unicode. Todos los juegos de caracteres se pueden utilizar transparentemente a través del servidor. El juego de caracteres por default es seleccionado mientras se inicializa el cluster de la base de datos PostgreSQL.*
- *La siguiente tabla muestra los juegos de caracteres soportados.*





Manejo de Bases de Datos (cont.)

Nombre	Descripción
SQL_ASCII	ASCII
EUC_JP	Japonés EUC
EUC_CN	Chino EUC
EUC_KR	Coreano EUC
JOHAB	Coreano EUC (base Hangle)
EUC_TW	Taiwán EUC
UNICODE	Unicode (UTF-8)
MULE_INTERNAL	Código Interno Mule
LATIN1	ISO 8859-1/ECMA 94 (alfabeto latino no.1)
LATIN2	ISO 8859-2/ECMA 94 (alfabeto latino no.2)
LATIN3	ISO 8859-3/ECMA 94 (alfabeto latino no.3)
LATIN4	ISO 8859-4/ECMA 94 (alfabeto latino no.4)
LATIN5	ISO 8859-9/ECMA 128 (alfabeto latino no.5)
LATIN6	ISO 8859-10/ECMA 144 (alfabeto latino no.6)
LATIN7	ISO 8859-13 (alfabeto latino no.7)
LATIN8	ISO 8859-14 (alfabeto latino no.8)
LATIN9	ISO 8859-15 (alfabeto latino no.9)
LATIN10	ISO 8859-16/ASRO SR 14111(alfabeto latino no.10)
ISO_8859_5	ISO 8859-5/ECMA 113 (Latino/Cirílico)
ISO_8859_6	ISO 8859-6/ECMA 114 (Latino/Árabe)
ISO_8859_7	ISO 8859-7/ECMA 118 (Latino/Griego)
ISO_8859_8	ISO 8859-8/ECMA 121 (Latino/Hebreo)
KOI8	KOI8-R(U)
WIN	Windows CP1251
ALT	Windows CP866
WIN1256	Windows CP1256 (Árabe)
TCVN	TCVN-5712/Windows CP1258 (Vietnamita)
WIN874	Windows CP874 (Tailandés)





Manejo de Bases de Datos (cont.)

- *Fijar el juego de caracteres por default:*

```
initdb -E EUC_JP
```

- *También se puede crear una base de datos con un juego de caracteres diferente, por ejemplo:*

```
createdb -E EUC_KR koreano
```

o

```
CREATE DATABASE koreano WITH ENCODING 'EUC_KR';
```



Práctica 3



Manos a la máquina!!!





Mantenimiento de la Base de Datos

- ***Existen algunas tareas de mantenimiento que se deben realizar periódicamente para asegurar que el servidor PostgreSQL se ejecute sin problemas. Las tareas discutidas aquí son repetitivas y pueden fácilmente automatizarse usando las herramientas estándares de Unix, tal es el caso de los scripts de cron, sin embargo, es responsabilidad del administrador de la base de datos habilitar los scripts apropiados y comprobar que se ejecuten satisfactoriamente.***
- ***Una tarea obvia del mantenimiento es la creación de respaldos de los datos bajo un determinado horario. Sin un respaldo reciente, no existe ninguna posibilidad de recuperar los datos después de una catástrofe, por ejemplo fuego, falla del disco, entre otros.***



Mantenimiento de la Base de Datos

- **VACUUM**
 - El comando VACUUM de PostgreSQL se debe correr regularmente por varias razones:
 - Para recuperar espacio en disco ocupado por actualizaciones o registros borrados.
 - Para actualizar la estadística de datos usada por el planificador de consultas en PostgreSQL.
 - ***La tarea recomendada para la mayoría de los sitios es programar un VACUUM a la base de datos una vez al día en las horas de menos trabajo, es recomendable utilizar VACUUM y no VACUUM completo para la recuperación del espacio.***



Mantenimiento de la Base de Datos

- *Actualización del planeador estadístico*
- *El planificador de consultas de PostgreSQL hace uso de la información estadística sobre el contenido de tablas para optimizar las consultas. Estas estadísticas son recopiladas por el comando ANALYZE, que se puede invocar así mismo o como un paso opcional en VACUUM. Es importante contar con estadísticas exactas, ya que de lo contrario el rendimiento de las consultas puede no ser el mejor.*



Mantenimiento de la Base de Datos



- *Rutina de indexado*
- *En algunas ocasiones es recomendable reconstruir los índices con el comando REINDEX. Sin embargo, A partir de la versión 7.4 se ha reducido sustancialmente la necesidad de esta actividad comparada con versiones anteriores.*





Mantenimiento de la Base de Datos

- *Respaldos y restauración de la información*

En Postgres las bases de datos se deben respaldar regularmente. El procedimiento es muy simple, es importante tener un bosquejo básico de las técnicas de respaldo.

Hay dos funciones fundamentales para trabajar con PostgreSQL:

- Respaldo de SQL
- Respaldo a nivel del sistema de archivos



Mantenimiento de la Base de Datos



- *Respaldo a nivel de sistema de archivos*

Una alternativa para respaldar, es copiar directamente los archivos donde PostgreSQL almacena los datos de la base de datos. Se puede utilizar cualquier método para los respaldos generalmente del sistema de archivos, se hace de la siguiente manera:

```
tar -cf backup.tar /usr/local/pgsql/data
```



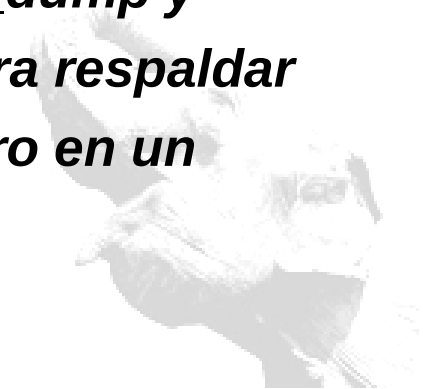


Mantenimiento de la Base de Datos

- *Respaldos SQL-dump*

La idea básica del método de SQL-dump es generar un archivo del texto con sentencias SQL ,el objetivo es reconstruir la base de datos en el mismo estado que era a la hora de la respaldo

PostgreSQL nos proporciona las herramientas pg_dump y pg_dumpall, las cuales nos ayudan tanto como para respaldar las tablas, como para copiarlas de un sistema a otro en un formato transportable.





Mantenimiento de la Base de Datos

- *pg_dump* se emplea para respaldar una base de datos o una tabla en particular.
- *pg_dumpall* respalda todas las bases de datos en el sistema.

El uso básico de este comando es:

pg_dump dbname > outfile

EJEMPLO

La forma más general de emplear pg_dump para respaldar sólo una tabla, es la siguiente:

pg_dump -t mitabla mibase > respaldo.sql





Mantenimiento de la Base de Datos

- *Restauración de un respaldo*

Los archivos del texto creados por el pg_dump deben ser leídos adentro por programa psql. La forma general del comando para restaurar una respaldo es:

psql dbname -f infile

Una vez que esté restaurado, es necesario ejecutar ANALYZE en cada base de datos, el optimizador saca estadísticas. o correr vacuumdb - a - z para analizar todas las bases de datos.



Mantenimiento de la Base de Datos



- *Respaldo y restauración en un solo comando*

pg_dump y psql tienen capacidad para leer y escribir en pipe (|) esto permite respaldar una base de datos directamente a partir de un servidor a otro; por ejemplo:

pg_dump - h host1 dbname |psql - h host2 dbname



Mantenimiento de la Base de Datos



- *Respaldo y compresión*

Se puede utilizar herramientas de Unix para trabajar con problemas de espacio, es común que se tenga una tabla, o base de datos que utilice mas espacio de lo permitido por el sistema operativo. Con pg_dump puede escribir a la salida estándar por lo tanto, se puede pasar a un programa que comprime datos.

```
pg_dump dbname | gzip > filename.gz
```



Ejercicio Final



Manos a la máquina!!!

