



Prácticas Introductorio PostgreSQL

Índice de contenido

1 Introducción.....	4
2 Módulos.....	4
2.1 Módulo 1.....	4
2.2 Módulo 2.....	4
2.3 Modulo 3.....	4
2.3.1 Ejercicio 1.....	4
2.3.2 Solución 1.....	4
2.4 Modulo 4.....	9
2.4.1 Ejercicio 1.....	9
2.4.2 Solución 1.....	10
2.4.3 Ejercicio 2.....	12
2.4.4 Solución 2.....	12
2.4.5 Ejercicio 3.....	14
2.4.6 Solución 3.....	14
2.4.7 Ejercicio 4.....	15
2.4.8 Solución 4.....	16
2.5 Modulo 5.....	17
2.5.1 Ejercicio 1.....	17
2.5.2 Solución 1.....	18
2.5.3 Ejercicio 2.....	18
2.5.4 Solución 2.....	18
2.5.5 Ejercicio 3.....	19
2.5.6 Solución 3.....	19
2.5.7 Ejercicio 4.....	20
2.5.8 Solución 4.....	20
2.6 Modulo 6.....	21
2.6.1 Ejercicio 1.....	21
2.6.2 Solución 1.....	22
2.7 Modulo 7.....	27
2.7.1 Ejercicio 1.....	27
2.7.2 Solución 1.....	27
2.7.3 Ejercicio 2.....	33
2.7.4 Solución 2.....	33
2.7.5 Ejercicio 3.....	43
2.7.6 Solución 3.....	43
2.7.7 Ejercicio 4.....	45
2.7.8 Solución 4.....	46
2.8 Modulo 8.....	46
2.8.1 Ejercicio 1.....	46
2.8.2 Soluciones.....	46
2.8.3 Ejercicio 2.....	47
2.8.4 Solución 2.....	47
2.8.5 Ejercicio 3.....	48
2.8.6 Solución 3.....	48
2.8.7 Ejercicio 4.....	48

Practicas Introductorio PostgreSQL

2.8.8 Solución 4.....	49
2.9 Modulo 9.....	49
2.9.1 Ejercicio 1.....	49
2.9.2 Solución 1.....	50
2.9.3 Ejercicio 2.....	50
2.9.4 Solución 2.....	51
2.9.5 Ejercicio 3.....	52
2.9.6 Solución 3.....	52
2.9.7 Ejercicio 4.....	54
2.9.8 Solución 4.....	54
2.10 Modulo 10.....	55
2.10.1 Ejercicio 1.....	55
2.10.2 Solución 1.....	55
2.10.3 Ejercicio 2.....	56
2.10.4 Solución 2.....	56
2.10.5 Ejercicio 3.....	56
2.10.6 Solución 3.....	57
2.10.7 Ejercicio 4.....	57
2.10.8 Solución 4.....	58
2.10.9 Ejercicio 5.....	59
2.10.10 Solución 5.....	60
2.10.11 Ejercicio 6.....	60
2.10.12 Solución 6.....	61
2.11 Modulo 11.....	64
2.11.1 Ejercicio 1.....	64
2.11.2 Solución 1.....	64
2.11.3 Ejercicio 2.....	65
2.11.4 Solución 2.....	65
2.11.5 Ejercicio 3.....	66
2.11.6 Solución 3.....	66
2.11.7 Ejercicio 4.....	67
2.11.8 Solución 4.....	67
2.12 Modulo 12.....	67
2.12.1 Ejercicio 1.....	67
2.12.2 Solución 1.....	67
2.12.3 Ejercicio 2.....	68
2.12.4 Solución 2.....	68
2.12.5 Ejercicio 3.....	68
2.12.6 Solución 3.....	68
2.12.7 Ejercicio 4.....	69
2.12.8 Solución 4.....	69
2.13 Modulo 13.....	70
2.13.1 Ejercicio 1.....	70
2.13.2 Solución 1.....	70

1 Introducción

El presente documento sirve de guía para la resolución de las prácticas que se encuentran en el curso de Introducción a PostgreSQL.

2 Módulos

2.1 Módulo 1

Sin ejercicios

2.2 Módulo 2

Sin ejercicios

2.3 Módulo 3

2.3.1 Ejercicio 1

- Elegir la plataforma en la que se desea instalar PostgreSQL.
- Descargar el instalador en un click de la web de EnterpriseDB para la plataforma elegida.
- Preparar la plataforma
- Instalar PostgreSQL
- Conectarse a PostgreSQL empleando psql

2.3.2 Solución 1

Instalar PostgreSQL

- Descargar el software PostgreSQL 9.4 de la página de www.enterprisedb.com
- Loguearse como usuario root
- Ejecutar los binarios

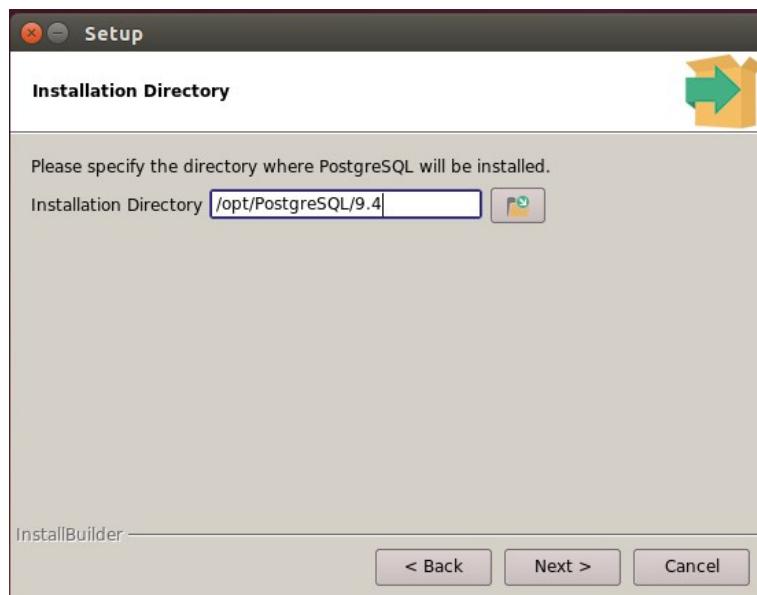
Prácticas Introductorio PostgreSQL

```
root@training-postgres:/home/postgres/material/installador# ./postgresql-9.4.4-3-linux.run
```

Pantalla de inicio de instalación.



Selección de la ubicación de los binarios de PostgreSQL.



Prácticas Introductorio PostgreSQL

Selección de la ubicación de la base del cluster.

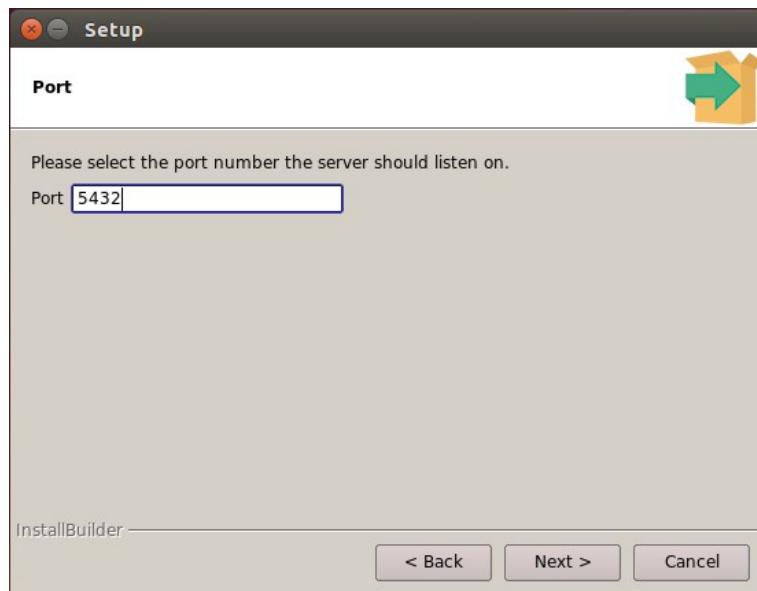


Password del superusuario de la base de datos (Usuario postgres).



Prácticas Introductorio PostgreSQL

Selección del puerto donde escuchará el postmaster.

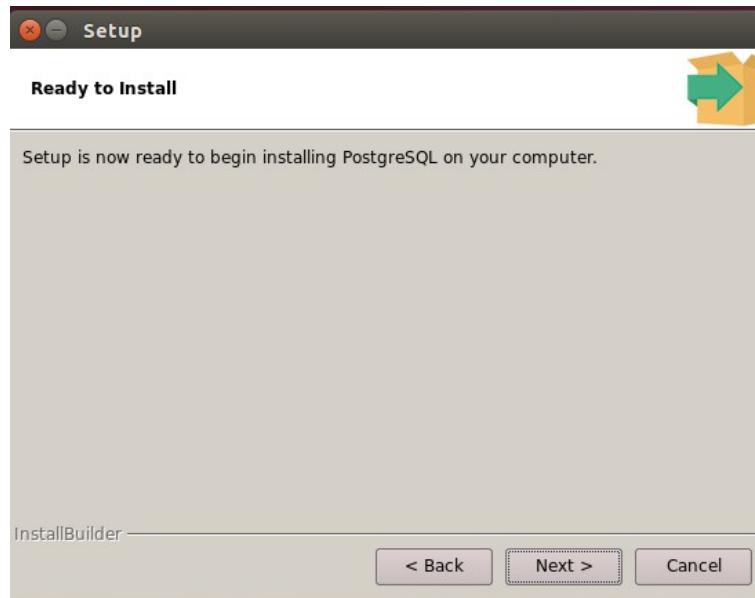


Selección del locale de la máquina.

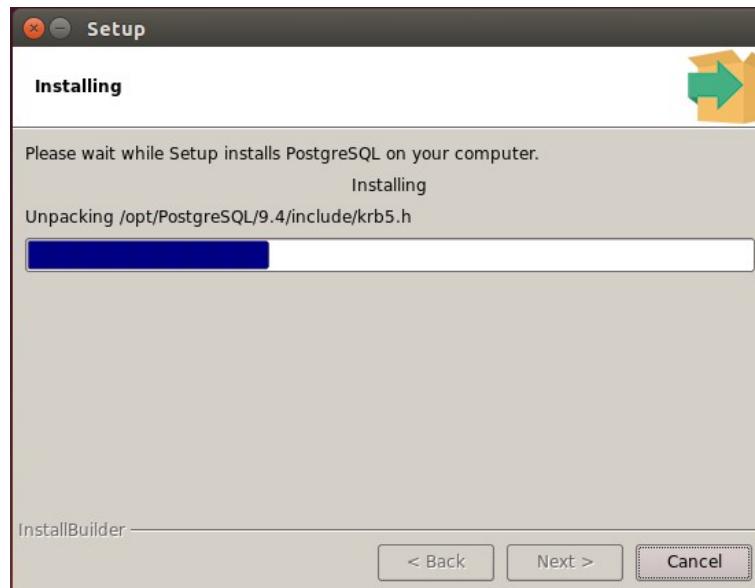


Prácticas Introductorio PostgreSQL

Resumen antes de la instalación.



Proceso de instalación.



Prácticas Introductorio PostgreSQL

Ejecución del stackbuilder para despliegue de paquetería. No ejecutar.



Conectarse a PostgreSQL empleando psql

Prueba de conexión mediante *psql*.

```
postgres@training-postgres:~$ psql
psql.bin (9.4.4)
Type "help" for help.
```

```
No entry for terminal type "xterm";
using dumb terminal settings.
postgres=#
```

2.4 Modulo 4

2.4.1 Ejercicio 1

- Estas trabajando como DBA. Se recomienda mantener una copia de backup de los ficheros de configuración antes de realizar ningún cambio.
- Permitir que el servidor de base de datos reciba 200 conexiones de usuario simultáneas.
- Reservar 10 conexiones para usuarios DBA.
- Máximo tiempo para completar la autentificación de cliente en 10 segundos.

2.4.2 Solución 1

Estas trabajando como DBA. Se recomienda mantener una copia de backup de los ficheros de configuración antes de realizar ningún cambio

```
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ cp postgresql.conf  
postgresql.conf_bak
```

Permitir que el servidor de base de datos reciba 200 conexiones de usuario simultáneas

```
# Modificaciones  
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf  
  
#max_connections = 100  
max_connections = 200  
  
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart  
Password:  
Restarting PostgreSQL 9.4:  
  
Password:  
waiting for server to shut down.... done  
server stopped  
waiting for server to start.... done  
server started  
  
PostgreSQL 9.4 restarted successfully  
  
# Comprobaciones  
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql  
psql.bin (9.4.4)  
Type "help" for help.  
  
No entry for terminal type "xterm";  
using dumb terminal settings.  
postgres=# show max_connections;  
max_connections  
-----  
200  
(1 row)  
  
postgres=#
```

Reservar 10 conexiones para usuarios DBA

```
# Modificaciones  
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf  
  
superuser_reserved_connections = 10
```

Practicas Introductorio PostgreSQL

```
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started

PostgreSQL 9.4 restarted successfully

# Comprobaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql
psql.bin (9.4.4)
Type "help" for help.

No entry for terminal type "xterm";
using dumb terminal settings.
postgres=# show superuser_reserved_connections ;
superuser_reserved_connections
-----
10
(1 row)

postgres=#

```

Máximo tiempo para completar la autentificación de cliente en 10 segundos

```
# Modificaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf

#authentication_timeout = 1min
authentication_timeout = 10s

postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started

PostgreSQL 9.4 restarted successfully

# Comprobaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql
psql.bin (9.4.4)
Type "help" for help.
```

```
No entry for terminal type "xterm";
using dumb terminal settings.
postgres=# show authentication_timeout ;
authentication_timeout
-----
10s
(1 row)

postgres=#
```

2.4.3 Ejercicio 2

- Realice los siguientes cambios y compruébelos
- Guarda los mensajes de error dentro de la ruta pg_log de tu instancia
- Loguea todas las queries que duran más de 5 segundos y sus tiempos
- Loguea los usuarios que se conectan/desconectan a la instancia
- Haz lo cambios arriba especificados y verifícalos
- Incluir mayor nivel de información en el log

2.4.4 Solución 2

Realice los siguientes cambios y compruébelos

Guarda los mensajes de error dentro de la ruta pg_log de tu instancia

Loguea todas las queries que duran más de 5 segundos y sus tiempos

Loguea los usuarios que se conectan/desconectan a la instancia

Incluir mayor nivel de información en el log

```
# Modificaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf

# Guardar mensajes de error
log_destination = 'stderr'
logging_collector = on
log_directory = 'pg_log'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'

# Loguear las queries que duran más de 5 segundos
log_min_duration_statement = 5000

# Loguear los usuarios que se conectan/desconectan a la base de datos
```

Prácticas Introductorio PostgreSQL

```
log_connections = on
log_disconnections = on
# Incluir mayor nivel de información en el log
log_line_prefix = '%t:%r:%d:%u:%p'
```

Haz lo cambios arriba especificados y verifícalos

```
# Reinicio para coger los cambios
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
PostgreSQL 9.4 restarted successfully

# Se comprueban los cambios
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql
psql.bin (9.4.4)
Type "help" for help.

No entry for terminal type "xterm";
using dumb terminal settings.
postgres=# show log_destination ;
log_destination
-----
stderr
(1 row)

postgres=# show logging_collector ;
logging_collector
-----
on
(1 row)

postgres=# show log_directory ;
log_directory
-----
pg_log
(1 row)

postgres=# show log_filename ;
log_filename
-----
postgresql-%Y-%m-%d_%H%M%S.log
(1 row)
```

Prácticas Introductorio PostgreSQL

```
postgres=# show log_min_duration_statement ;
log_min_duration_statement
-----
5s
(1 row)

postgres=# show log_connections ;
log_connections
-----
on
(1 row)

postgres=# show log_disconnections;
log_disconnections
-----
on
(1 row)

postgres=# show log_line_prefix ;
log_line_prefix
-----
%t:%r:%d:%u:%p
(1 row)

postgres=#
```

2.4.5 Ejercicio 3

- Realice los siguientes cambios de configuración
- Shared buffer a 256MB
- Effective Cache para los índices a 512MB
- Memoria de Mantenimiento a 64MB

2.4.6 Solución 3

Realice los siguientes cambios de configuración

Shared buffer a 256MB

Effective Cache para los índices a 512MB

Memoria de Mantenimiento a 64MB

```
# Modificaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf
```

Practicas Introductorio PostgreSQL

```
shared_buffers = 256MB
effective_cache_size = 512MB
maintenance_work_mem = 64MB

postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
PostgreSQL 9.4 restarted successfully

# Comprobaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql
psql.bin (9.4.4)
Type "help" for help.

No entry for terminal type "xterm";
using dumb terminal settings.
postgres=# show shared_buffers ;
shared_buffers
-----
256MB
(1 row)

postgres=# show effective_cache_size ;
effective_cache_size
-----
512MB
(1 row)

postgres=# show maintenance_work_mem ;
maintenance_work_mem
-----
64MB
(1 row)

postgres=#

```

2.4.7 Ejercicio 4

- La tarea de mantenimiento de Vacuum es fundamental en PostgreSQL. Cambiar algunos parámetros de configuración
- Autovacuum workers a 6

Prácticas Introductorio PostgreSQL

- Autovacuum threshold a 100
- Autovacuum scale factor a 0.3
- Auto Analyze threshold a 100
- Auto vacuum cost limit a 100

2.4.8 Solución 4

La tarea de mantenimiento de Vacuum es fundamental en PostgreSQL.
Cambiar algunos parámetros de configuración

Autovacuum workers a 6

Autovacuum threshold a 100

Autovacuum scale factor a 0.3

Auto Analyze threshold a 100

Auto vacuum cost limit a 100

```
# Modificaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ vim postgresql.conf

autovacuum_max_workers = 6
autovacuum_vacuum_threshold = 100
autovacuum_analyze_threshold = 100
autovacuum_vacuum_scale_factor = 0.3
autovacuum_vacuum_cost_limit = 100

postgres@training-postgres:/opt/PostgreSQL/9.4/data$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
PostgreSQL 9.4 restarted successfully

# Comprobaciones
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql
psql.bin (9.4.4)
Type "help" for help.

No entry for terminal type "xterm";
```

Practicas Introductorio PostgreSQL

```
using dumb terminal settings.  
postgres=# show autovacuum_max_workers;  
autovacuum_max_workers  
-----  
6  
(1 row)  
  
postgres=# show autovacuum_vacuum_threshold ;  
autovacuum_vacuum_threshold  
-----  
100  
(1 row)  
  
postgres=# show autovacuum_analyze_threshold ;  
autovacuum_analyze_threshold  
-----  
100  
(1 row)  
  
postgres=# show autovacuum_vacuum_scale_factor ;  
autovacuum_vacuum_scale_factor  
-----  
0.3  
(1 row)  
  
postgres=# show autovacuum_vacuum_cost_limit ;  
autovacuum_vacuum_cost_limit  
-----  
100  
(1 row)  
  
postgres=#
```

2.5Modulo 5

2.5.1 Ejercicio 1

- Una nueva web para alojar una base de datos de música online se está desarrollando
- Crea un usuario de base de datos *hoplastore* dentro de tu instancia
- Crea una base de datos *hoplastore* cuyo propietario sea *hoplastore*
- Logueate dentro de la base de datos *hoplastore* con el usuario *hoplastore* y crea el esquema *hoplastore*

2.5.2 Solución 1

Una nueva web para alojar una base de datos de música online se está desarrollando

Crea un usuario de base de datos *hoplastore* dentro de tu instancia

```
postgres@training-postgres:/opt/PostgreSQL/9.4/data$ psql  
psql.bin (9.4.4)  
Type "help" for help.
```

No entry for terminal type "xterm";
using dumb terminal settings.

```
postgres=# create user hoplastore password 'hoplastore';  
CREATE ROLE
```

Crea una base de datos *hoplastore* cuyo propietario sea *hoplastore*

```
postgres=# create database hoplastore owner hoplastore;  
CREATE DATABASE
```

Logueate dentro de la base de datos *hoplastore* con el usuario *hoplastore* y crea el esquema *hoplastore*

```
postgres=# \c hoplastore hoplastore  
Password for user hoplastore:  
You are now connected to database "hoplastore" as user "hoplastore".  
hoplastore=> create schema hoplastore;  
CREATE SCHEMA  
hoplastore=> \q
```

2.5.3 Ejercicio 2

- Un desarrollador de una aplicación web de música online quiere añadir un módulo de compra/venta y ha te solicitado separar todas las tablas usadas en transacciones online. En este momento, le has sugerido usar esquemas siguiendo las siguientes instrucciones:
- Crear un usuario *ebuy* con password 'lion' .
- Crear un esquema llamado *ebuy* cuyo propietario sea el usuario *ebuy*.
- Loguearse como el usuario *ebuy*, crear una tabla *sample1* y verificar si se ha almacenado en el esquema *ebuy*.

2.5.4 Solución 2

Un desarrollador de una aplicación web de música online quiere añadir un

Prácticas Introductorio PostgreSQL

módulo de compra/venta y ha te solicitado separar todas las tablas usadas en transacciones online. En este momento, le has sugerido usar esquemas siguiendo las siguientes instrucciones:

Crear un usuario *ebuy* con password 'lion' .

```
postgres=# create user ebuy password 'lion';
CREATE ROLE
```

Crear un esquema llamado *ebuy* cuyo propietario sea el usuario *ebuy*.

```
postgres=# create schema ebuy authorization ebuy;
CREATE SCHEMA
```

Loguearse como el usuario *ebuy*, crear una tabla *sample1* y verificar si se ha almacenado en el esquema *ebuy*.

```
postgres=# \c postgres ebuy
Password for user ebuy:
You are now connected to database "postgres" as user "ebuy".
postgres=> create table sample1 (id numeric, name varchar);
CREATE TABLE
postgres=> \dt
      List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+
 ebuy  | sample1 | table | ebuy
(1 row)
```

2.5.5 Ejercicio 3

- Hopla Software proporciona una base de datos de test *hoplasoftware.sql* que se encuentra en la máquina de test o se proporcionará durante la práctica.
- Cargar la base de datos de prueba.
- Comprobar que se ha cargado correctamente.

2.5.6 Solución 3

Hopla Software proporciona una base de datos de test *hoplastore.sql* que se encuentra en la máquina de test o se proporcionará durante la práctica.

Cargar la base de datos de prueba.

```
postgres@training-postgres:~$ psql -f ~/material/labs/hoplastore.sql -d hoplastore -U
hoplastore
Password for user hoplastore:
```

Prácticas Introductorio PostgreSQL

```
...  
...  
...
```

Comprobar que se ha cargado correctamente.

```
postgres@training-postgres:~$ psql -U hoplastore -d hoplastore  
Password for user hoplastore:  
psql.bin (9.4.4)  
Type "help" for help.  
  
No entry for terminal type "xterm";  
using dumb terminal settings.  
hoplastore=> \dt  
      List of relations  
 Schema |   Name   | Type | Owner  
-----+-----+-----  
hoplastore | categories | table | hoplastore  
hoplastore | cust_hist | table | hoplastore  
hoplastore | customers | table | hoplastore  
hoplastore | dept     | table | hoplastore  
hoplastore | emp      | table | hoplastore  
hoplastore | inventory | table | hoplastore  
hoplastore | job_grd  | table | hoplastore  
hoplastore | jobhist   | table | hoplastore  
hoplastore | locations  | table | hoplastore  
hoplastore | orderlines | table | hoplastore  
hoplastore | orders    | table | hoplastore  
hoplastore | products   | table | hoplastore  
hoplastore | reorder   | table | hoplastore  
(13 rows)
```

2.5.7 Ejercicio 4

- Obtener una lista de las bases de datos empleando un comando SQL
- Obtener una lista de las bases de datos empleando un metacomando psql
- Obtener una lista de las tablas que se almacenan en la base de datos *hoplastore* y revisar que esquema y propietario tienen.

2.5.8 Solución 4

Obtener una lista de las bases de datos empleando un comando SQL

```
hoplastore=> select datname from pg_database;  
datname  
-----  
template1
```

Practicas Introductorio PostgreSQL

```
template0
postgres
hoplastore
(4 rows)
```

Obtener una lista de las bases de datos empleando un metacomando psql

```
hoplastore=> \l
      List of databases
   Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+
hoplastore | hoplastore | UTF8 | en_US.utf8 | en_US.utf8 |
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 |
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres      +
          |         |       |           |       | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres      +
          |         |       |           |       | postgres=CTc/postgres
(4 rows)
```

Obtener una lista de las tablas que se almacenan en la base de datos *hoplastore* y revisar que esquema y propietario tienen

```
hoplastore=> \dt
      List of relations
 Schema |  Name  | Type | Owner
-----+-----+-----+
hoplastore | categories | table | hoplastore
hoplastore | cust_hist | table | hoplastore
hoplastore | customers | table | hoplastore
hoplastore | dept     | table | hoplastore
hoplastore | emp      | table | hoplastore
hoplastore | inventory | table | hoplastore
hoplastore | job_grd  | table | hoplastore
hoplastore | jobhist  | table | hoplastore
hoplastore | locations | table | hoplastore
hoplastore | orderlines | table | hoplastore
hoplastore | orders    | table | hoplastore
hoplastore | products  | table | hoplastore
hoplastore | reorder   | table | hoplastore
(13 rows)
```

2.6Modulo 6

2.6.1 Ejercicio 1

- Conectarse a la base de datos *postgres*
- Cambiar a *hoplastore* con el usuario *hoplastore*
- Describe de la tabla *customers*

Prácticas Introductorio PostgreSQL

- Describe de la tabla *customers* incluyendo la descripción
- Lista todas las bases de datos
- Lista todos los esquemas
- Lista todos los tablespaces
- Ejecuta un comando SQL y salvarlo en un fichero de salida
- Hacer lo mismo que arriba pero salvándolo sin las cabeceras de las columnas
- Crear un script con algún comando SQL y ejecutarlo desde *psql*
- Activar el formato extendido y probar un comando SQL
- Listar tablas, vistas y secuencias con sus privilegios de acceso asociados

2.6.2 Solución 1

Conectarse a la base de datos *postgres*

```
postgres@training-postgres:~$ psql
psql.bin (9.4.4)
Type "help" for help.
```

```
No entry for terminal type "xterm";
using dumb terminal settings.
postgres=#
```

Cambiar a *hoplastore* con el usuario *hoplastore*

```
postgres=# \c hoplastore hoplastore
Password for user hoplastore:
You are now connected to database "hoplastore" as user "hoplastore".
```

Describe de la tabla *customers*

```
hoplastore=> \d customers
                                         Table "hoplastore.customers"
   Column      |      Type      |          Modifiers
+-----+-----+
customerid    | integer      | not null default
nextval('customers_customerid_seq'::regclass)
firstname     | character varying(50) | not null
lastname      | character varying(50) | not null
address1      | character varying(50) | not null
address2      | character varying(50) |
city          | character varying(50) | not null
state         | character varying(50) |
```

Prácticas Introductorio PostgreSQL

```

zip          | integer      |
country     | character varying(50) | not null
region       | smallint      | not null
email        | character varying(50) |
phone        | character varying(50) |
creditcardtype | integer      | not null
creditcard   | character varying(50) | not null
creditcardexpiration | character varying(50) | not null
username     | character varying(50) | not null
password     | character varying(50) | not null
age          | smallint      |
income        | integer      |
gender        | character varying(1) |
Indexes:
  "customers_pkey" PRIMARY KEY, btree (customerid)
  "ix_cust_username" UNIQUE, btree (username)
Referenced by:
  TABLE "cust_hist" CONSTRAINT "fk_cust_hist_customerid" FOREIGN KEY
  (customerid) REFERENCES customers(customerid) ON DELETE CASCADE
  TABLE "orders" CONSTRAINT "fk_customerid" FOREIGN KEY (customerid)
  REFERENCES customers(customerid) ON DELETE SET NULL

```

Describe de la tabla *customers* incluyendo la descripción

hoplastore=> \d+ customers			
		Table "hoplastore.customers"	
Column	Type	Modifiers	
Storage	Stats		
target	Description		
	+-----+		
+-----+			
customerid	integer not null default nextval('customers_customerid_seq'::regclass) plain		
firstname	character varying(50) not null		
extended			
lastname	character varying(50) not null		
extended			
address1	character varying(50) not null		
extended			
address2	character varying(50)		
extended			
city	character varying(50) not null		
extended			
state	character varying(50)		

Practicas Introductorio PostgreSQL

```
extended |
| zip          | integer           | plain |
| country      | character varying(50) | not null |
extended |
| region       | smallint          | not null | plain
|
| email        | character varying(50) |
extended |
| phone        | character varying(50) |
extended |
| creditcardtype | integer          | not null |
plain   |
| creditcard   | character varying(50) | not null |
extended |
| creditcardexpiration | character varying(50) | not null |
| extended |
| | username    | character varying(50) | not null | |
| | extended |
| | | password   | character varying(50) | not null |
| | | extended |
| | | age        | smallint          | plain |
| | | income     | integer           | plain |
| | | gender     | character varying(1) |
| | | extended |
| | |
Indexes:
  "customers_pkey" PRIMARY KEY, btree (customerid)
  "ix_cust_username" UNIQUE, btree (username)
Referenced by:
  TABLE "cust_hist" CONSTRAINT "fk_cust_hist_customerid" FOREIGN KEY
  (customerid) REFERENCES customers(customerid) ON DELETE CA
  SCADE
  TABLE "orders" CONSTRAINT "fk_customerid" FOREIGN KEY (customerid)
  REFERENCES customers(customerid) ON DELETE SET NULL
```

Lista todas las bases de datos

```
hoplastore=> \l
```

Practicas Introductorio PostgreSQL

```
          List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+
hoplastore | hoplastore | UTF8 | en_US.utf8 | en_US.utf8 |
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 |
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres      +
|           |           |           |           | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres      +
|           |           |           |           | postgres=CTc/postgres
(4 rows)
```

Lista todos los esquemas

```
hoplastore=> \dn
      List of schemas
  Name | Owner
-----+
hoplastore | hoplastore
public   | postgres
(2 rows)
```

Lista todos los tablespaces

```
hoplastore=> \db
      List of tablespaces
  Name | Owner | Location
-----+-----+-----+
pg_default | postgres |
pg_global  | postgres |
(2 rows)
```

Ejecuta un comando SQL y salvarlo en un fichero de salida

```
hoplastore=> \o /tmp/customer_data_header.txt
hoplastore=> select * from customers;
hoplastore=> \o
```

Hacer lo mismo que arriba pero salvándolo sin las cabeceras de las columnas

```
hoplastore=> \t
Tuples only is on.
hoplastore=> \o /tmp/customer_data.txt
hoplastore=> select * from customers;
hoplastore=> \o
```

Crear un script con algún comando SQL y ejecutarlo desde *psql*

Prácticas Introductorio PostgreSQL

```
# Creación del fichero
postgres@training-postgres:~$ vi /tmp/customer_sql.sh
#!/bin/bash
psql -U postgres -d hoplastore -c 'SELECT * FROM hoplastore.customers limit 1;'

# Ejecución
postgres@training-postgres:~$ ./customer_sql.sh
 customerid | firstname | lastname | address1 | address2 | city | state | zip
 | country | region | email
 | phone | creditcardtype | creditcard | creditcardexpiration | username |
password | age | income | gender
-----+-----+-----+-----+-----+-----+-----+
-----+-----+
-----+-----+-----+-----+-----+-----+-----+
-----+-----+
 1 | VKUUXF | ITHOMQJNYX | 4608499546 Dell Way | QSDPAGD | SD |
24101 | US | 1 | ITHOMQJNYX@d
ell.com | 4608499546 | 1 | 1979279217775911 | 2012/03 | user1 |
password | 55 | 100000 | M
(1 row)
```

Activar el formato extendido y probar un comando SQL

```
hoplastore=> \x
Expanded display is on.
hoplastore=> select * from emp limit 1;
-[ RECORD 1 ]-----
empno | 7369
ename | SMITH
job   | CLERK
mgr   | 7902
hiredate | 1980-12-17 00:00:00
sal   | 800.00
comm  |
deptno | 20
```

Listar tablas, vistas y secuencias con sus privilegios de acceso asociados

```
hoplastore=> \dp
                         Access privileges
 Schema |      Name      | Type | Access privileges | Column ac
cess privileges
-----+-----+-----+-----+
-----+
hoplastore | categories      | table |          |          |
hoplastore | categories_category_seq | sequence |          |          |
hoplastore | cust_hist       | table |          |          |
hoplastore | customers        | table |          |          |
hoplastore | customers_customerid_seq | sequence |          |          |
hoplastore | dept            | table |          |          |
```

hoplastore	emp	table		
hoplastore	inventory	table		
hoplastore	job_grd	table		
hoplastore	jobhist	table		
hoplastore	locations	table		
hoplastore	next_empno	sequence		
hoplastore	orderlines	table		
hoplastore	orders	table		
hoplastore	orders_orderid_seq	sequence		
hoplastore	products	table		
hoplastore	products_prod_id_seq	sequence		
hoplastore	reorder	table		
hoplastore	salesemp	view		
(19 rows)				

2.7 Modulo 7

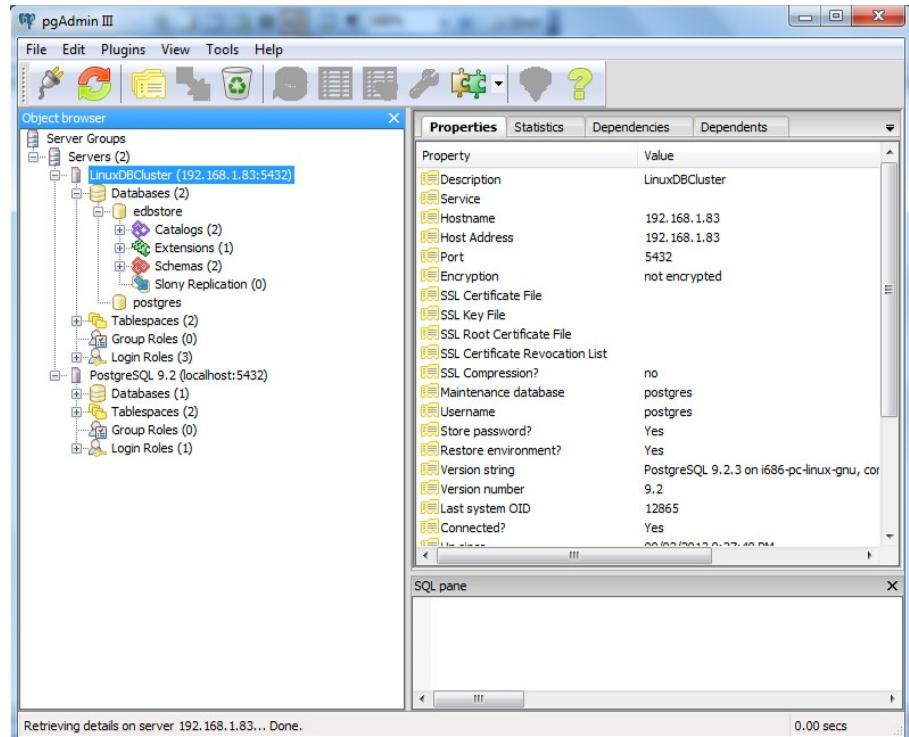
2.7.1 Ejercicio 1

- Abre PgAdmin III y conéctate a la base de datos por defecto de la instancia
- Crea un usuario llamado *pguser*
- Crea una base de datos *pgdb* cuyo propietario sea *pguser*
- Despues de la creación de *pgdb*, cambia el connection limit a 2
- Crea un esquema llamado *pguser* dentro de la base de datos *pgdb*. Su propietario será *pguser*

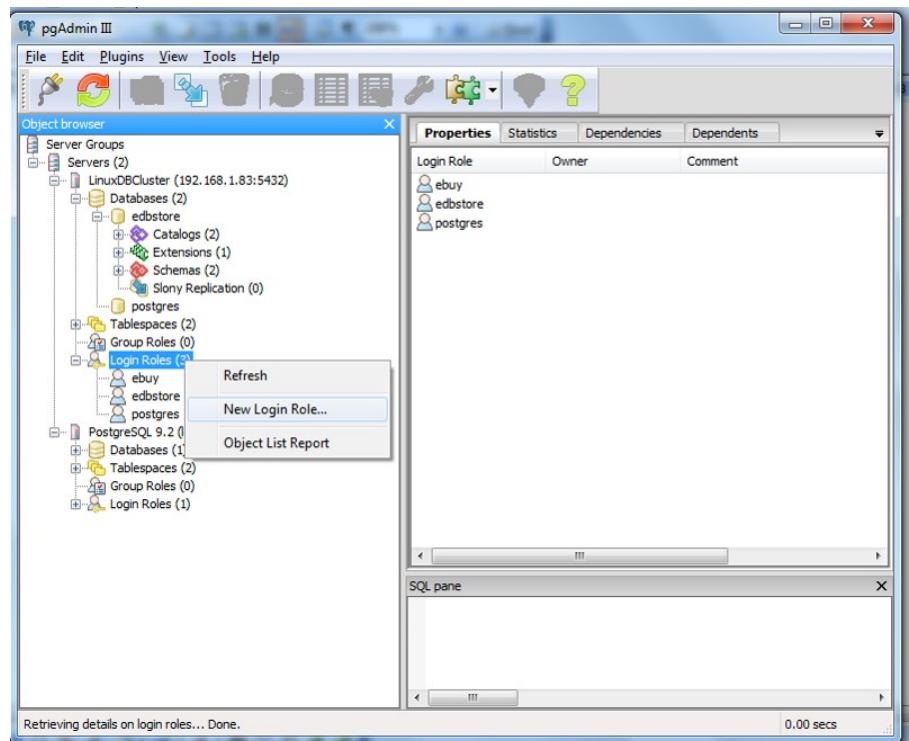
2.7.2 Solución 1

Abre PgAdmin III y conéctate a la base de datos por defecto de la instancia

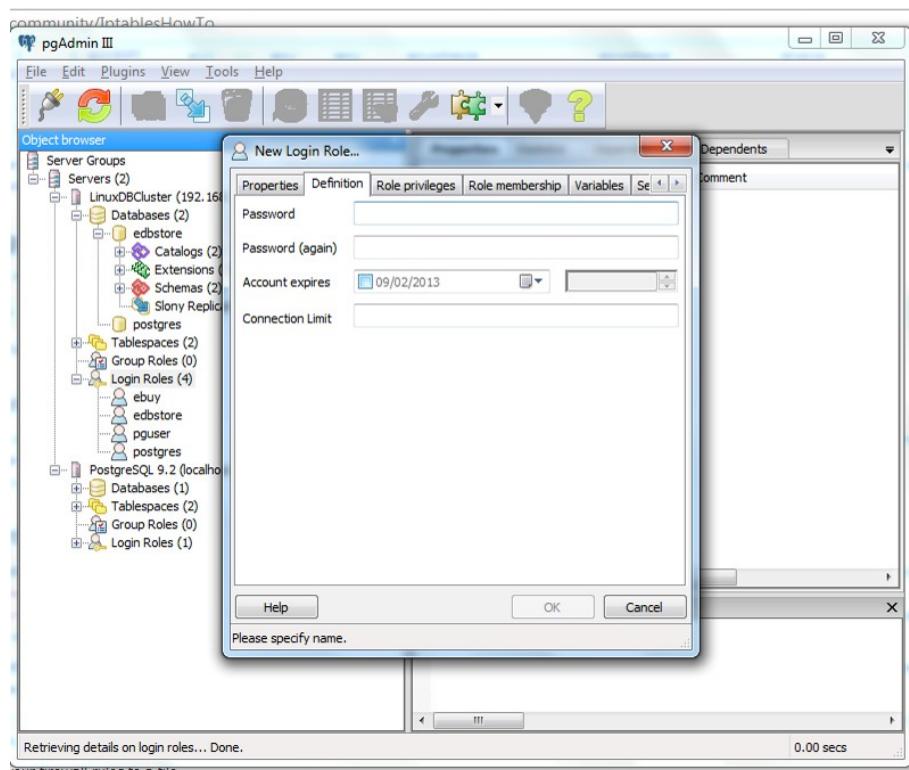
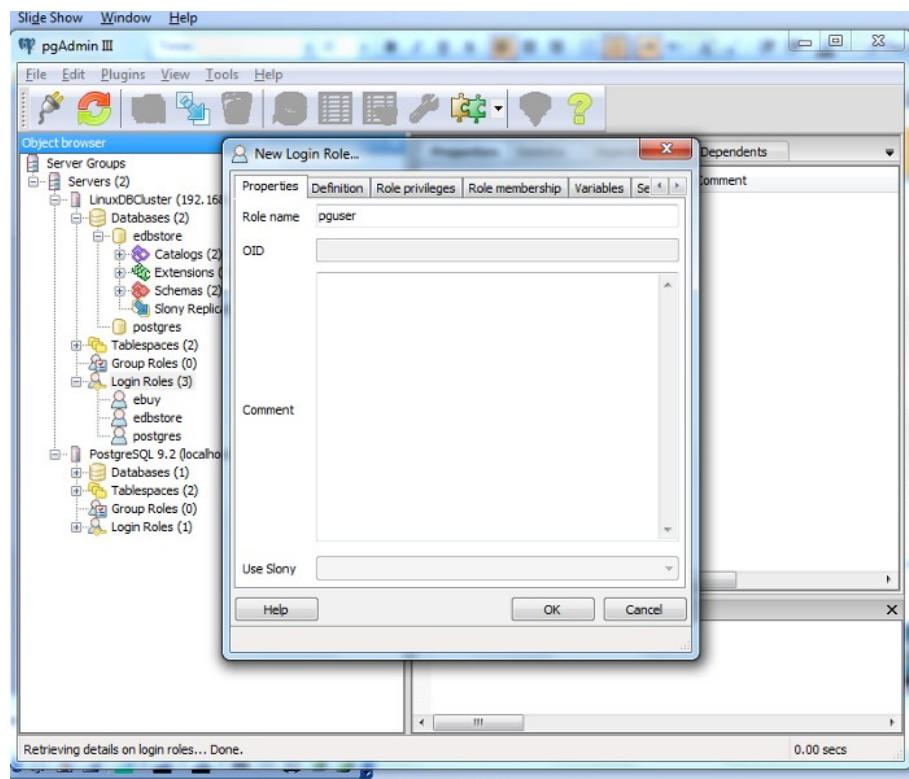
Prácticas Introductorio PostgreSQL



Crea un usuario llamado *pguser*

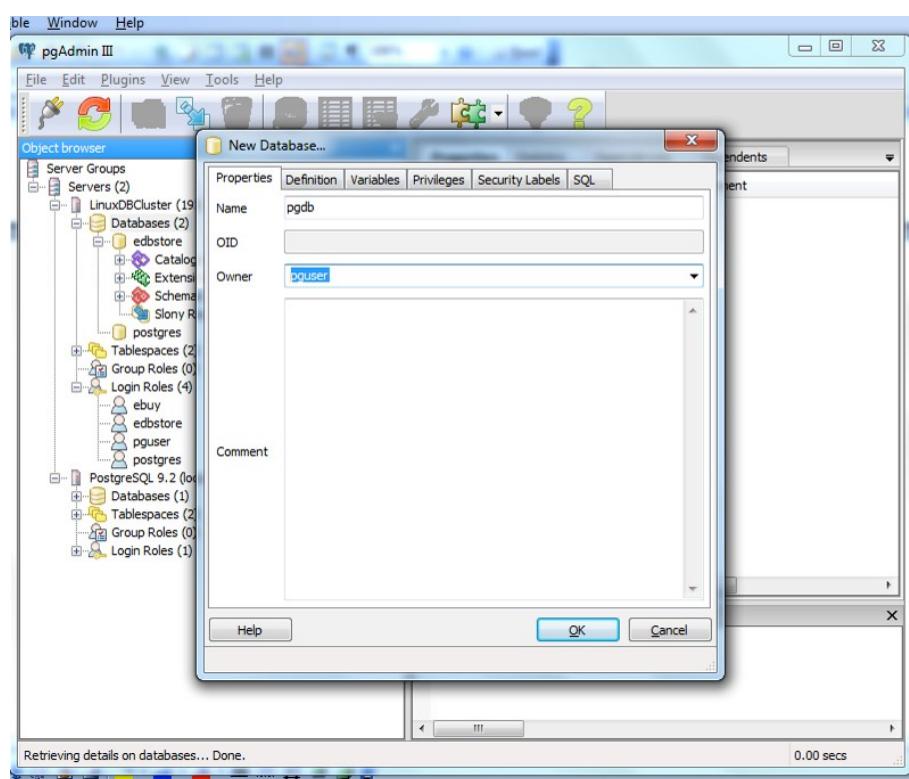
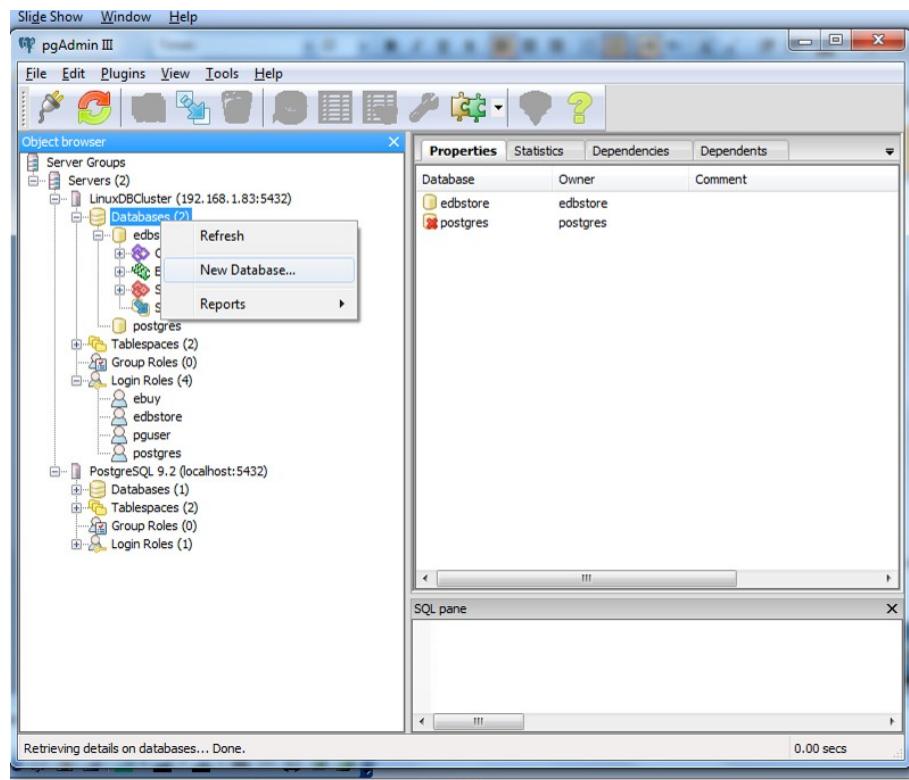


Prácticas Introductorio PostgreSQL



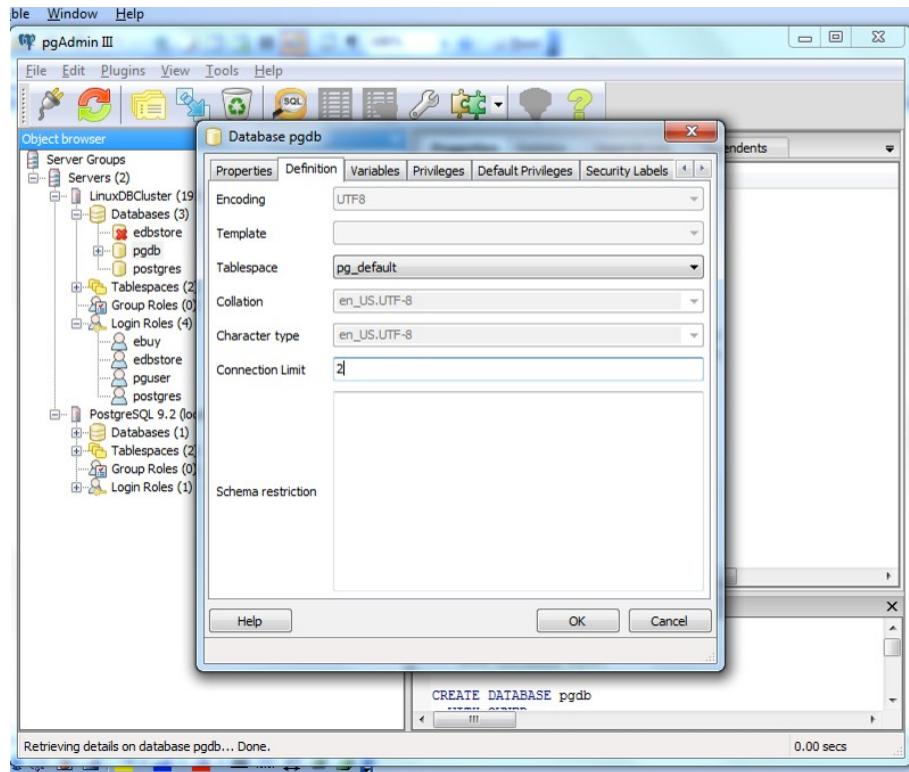
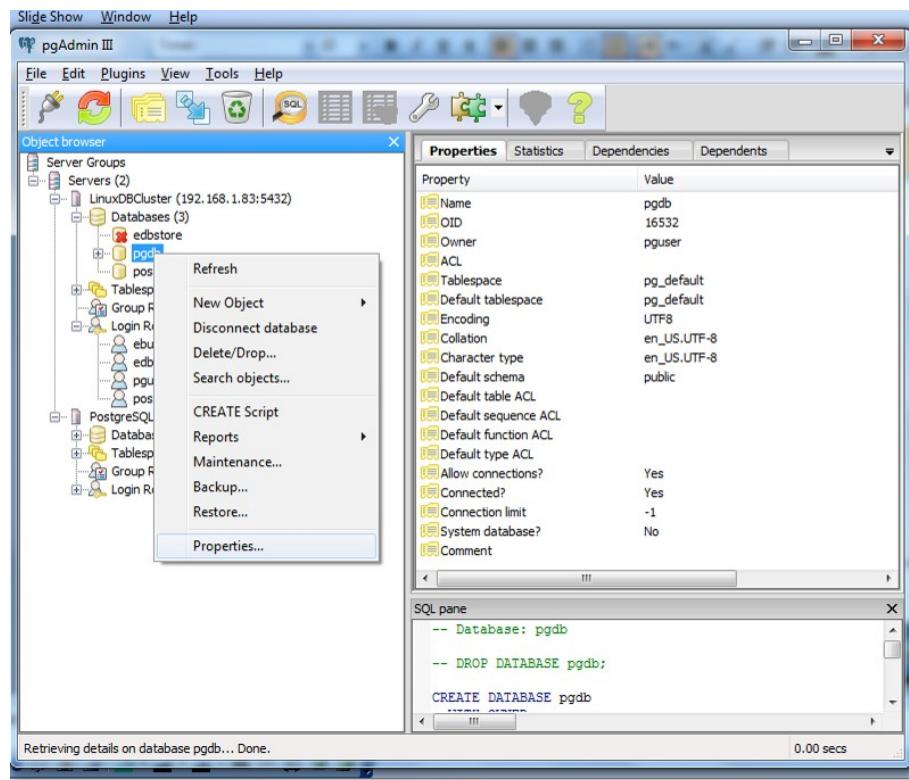
Crea una base de datos *pgdb* cuyo propietario sea *pguser*

Prácticas Introductorio PostgreSQL



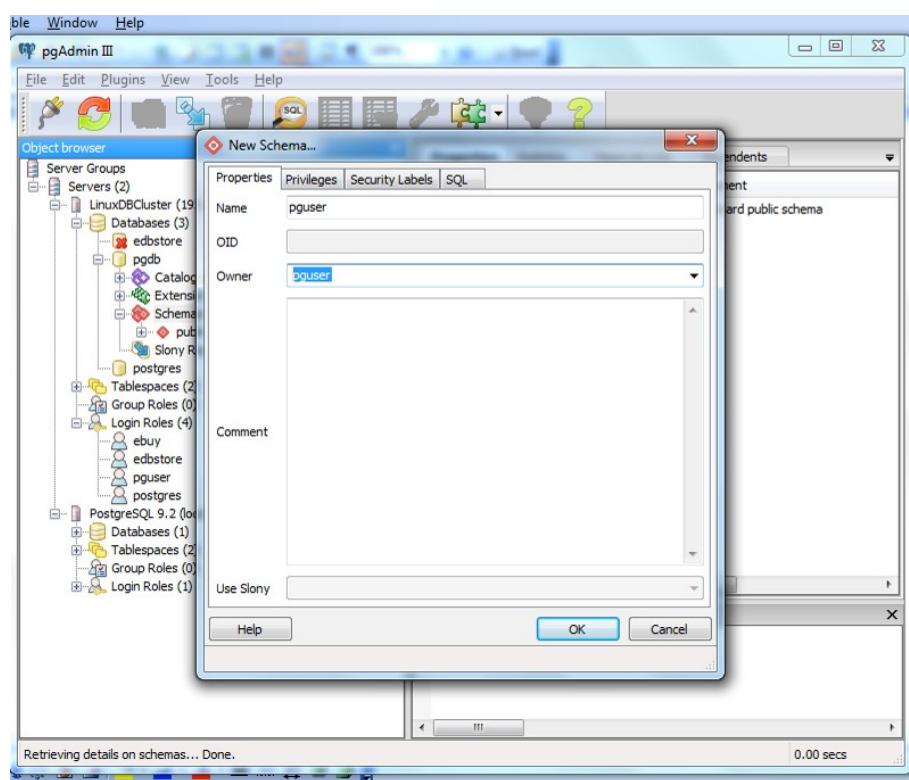
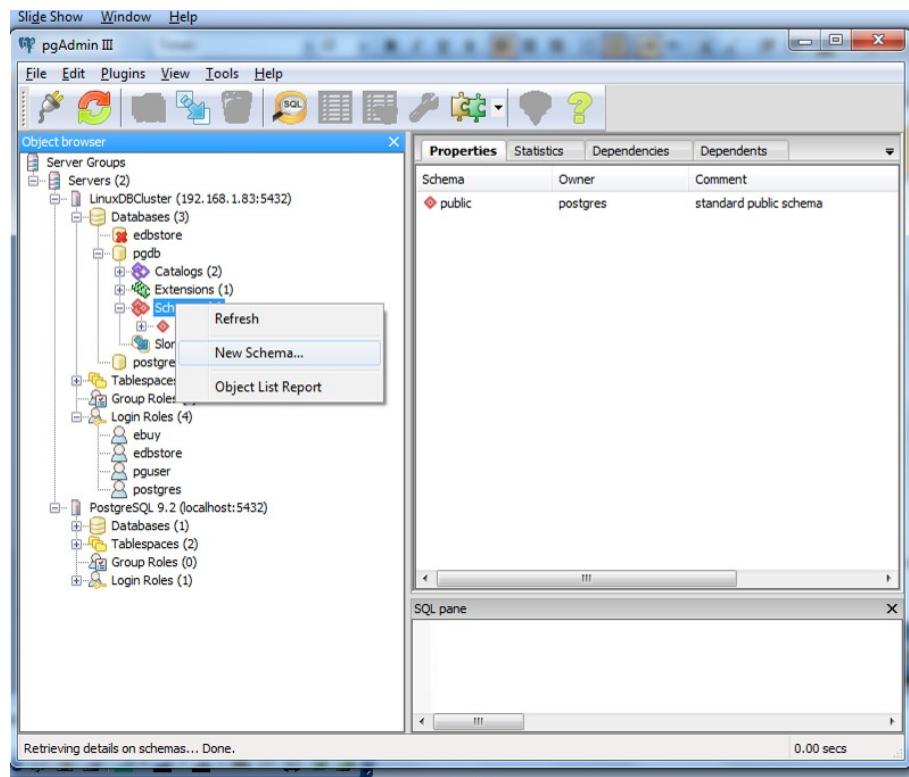
Después de la creación de *pgdb*, cambia el connection limit a 2

Prácticas Introductorio PostgreSQL



Crea un esquema llamado *pguser* dentro de la base de datos *pgdb*. Su propietario será *pguser*

Prácticas Introductorio PostgreSQL



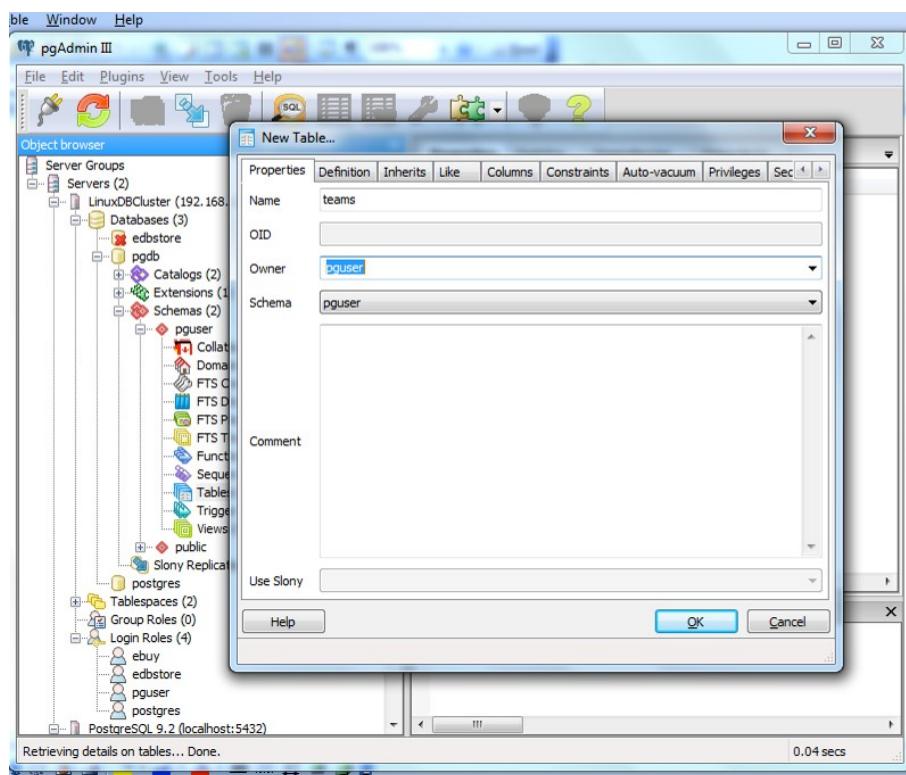
2.7.3 Ejercicio 2

- Ya has creado la base de datos *pgdb* con el esquema *pguser* bajo el propietario *pguser*. Crea los siguientes objetos:
 - Tabla: *Teams* con columnas *TeamID*, *TeamName*, *TeamRatings*
 - Secuencia: *seq_teamid* empezando en el valor 1 incrementando por 1
 - Columnas: Cambiar el valor por defecto para la columna *TeamID* a *seq_teamid*
 - Constraint: *TeamRatings* debe estar entre 1 y 10
 - Índice: Primary key en *TeamID*
 - Vista: Muestra todos los teams en orden ascendente de sus ratings. Nombre de la vista *vw_top_teams*

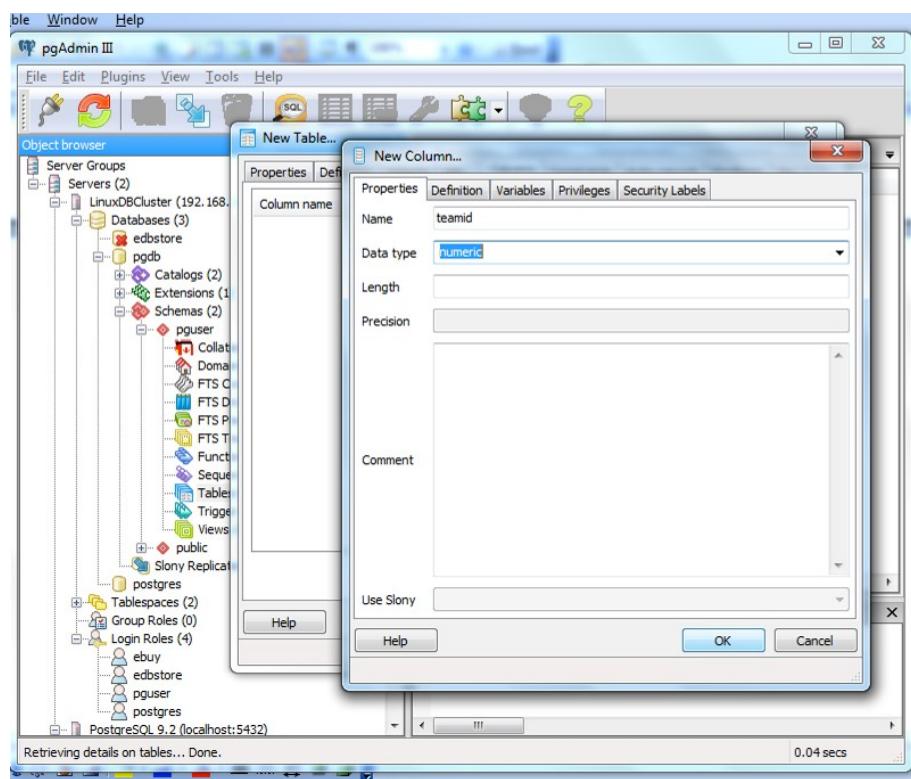
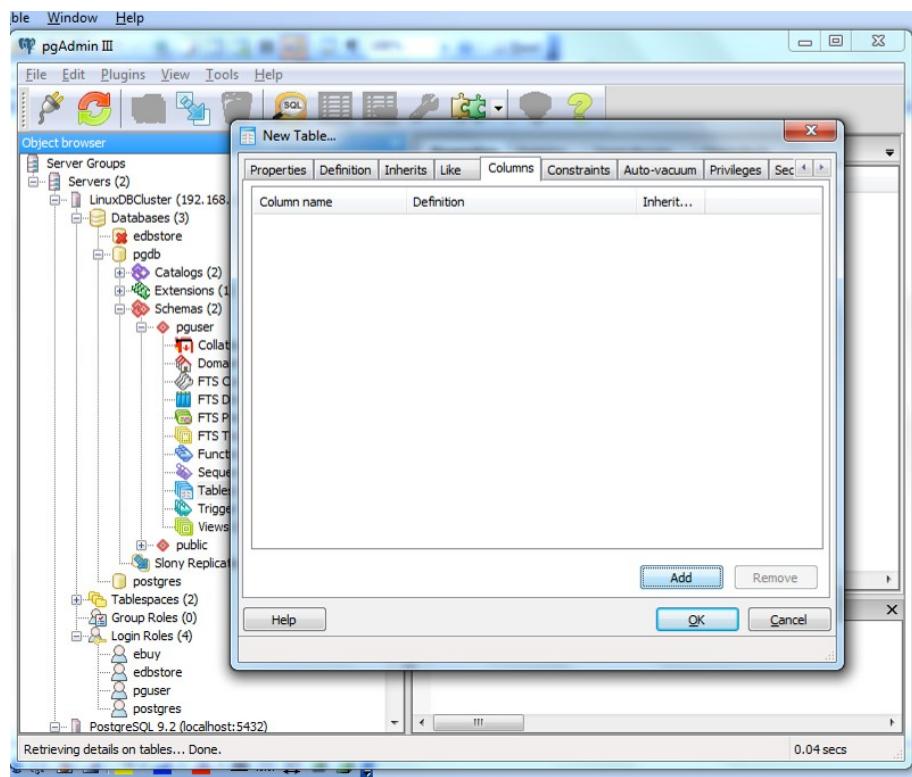
2.7.4 Solución 2

Ya has creado la base de datos *pgdb* con el esquema *pguser* bajo el propietario *pguser*. Crea los siguientes objetos:

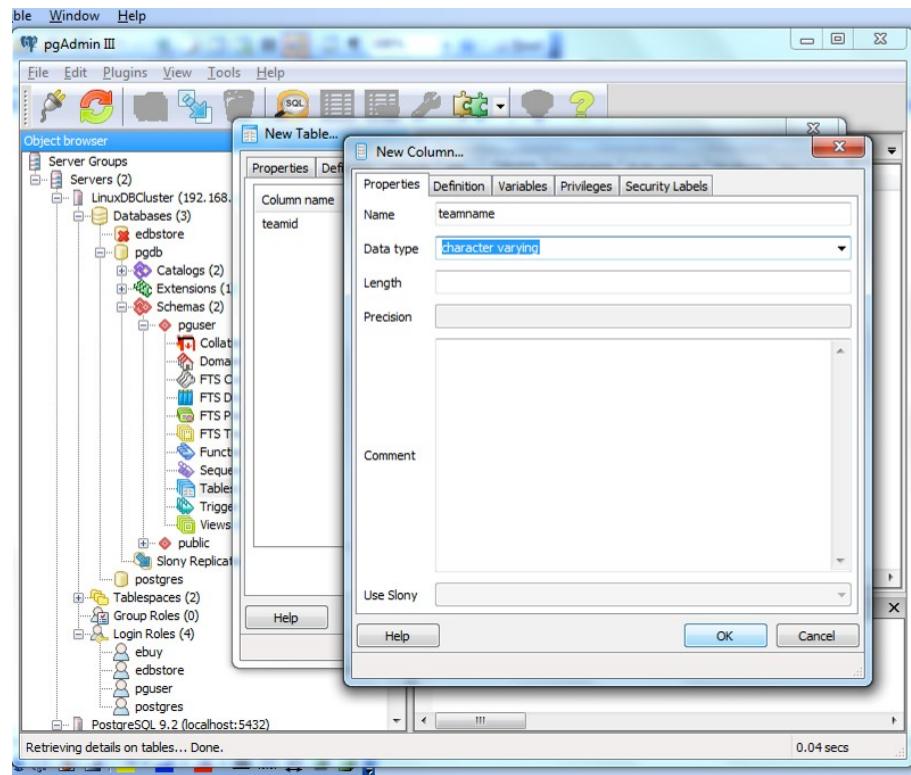
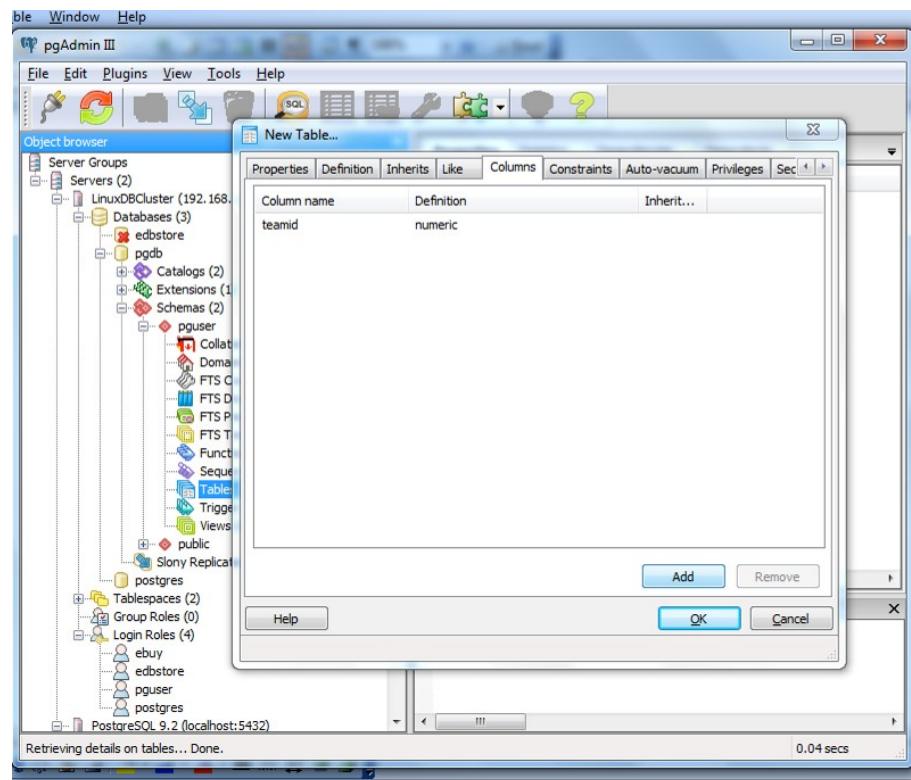
Tabla: *Teams* con columnas *TeamID*, *TeamName*, *TeamRatings*



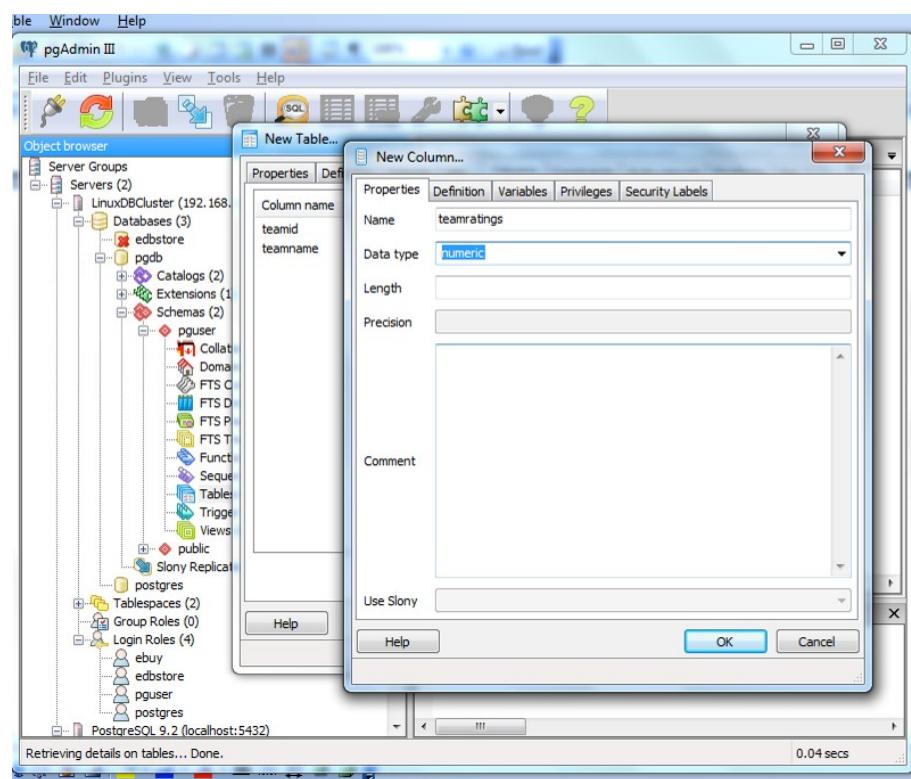
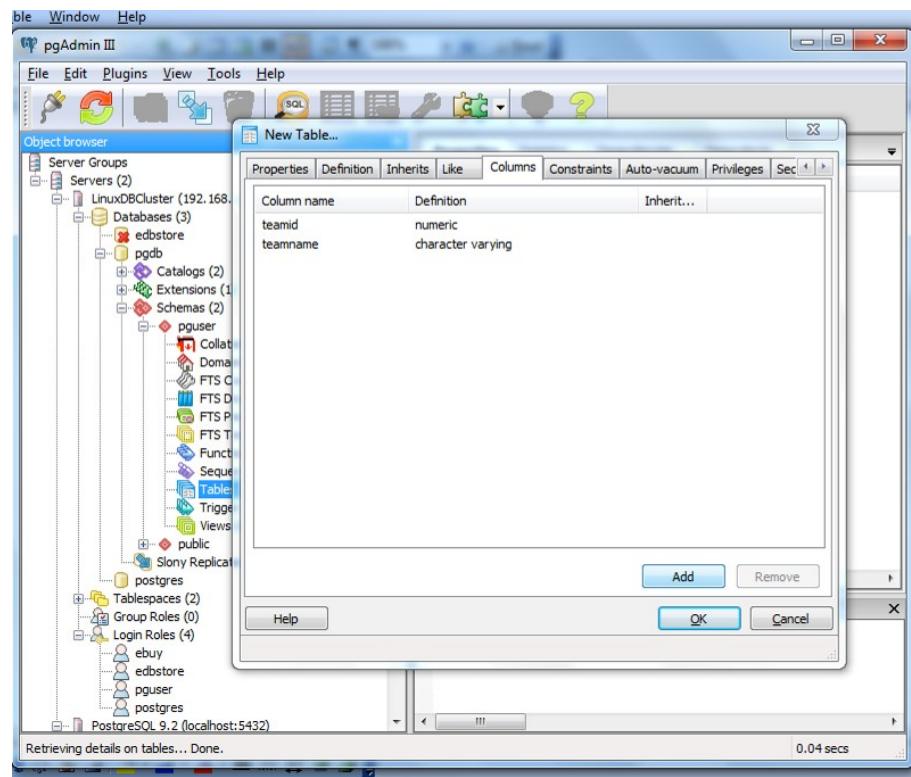
Prácticas Introductorio PostgreSQL



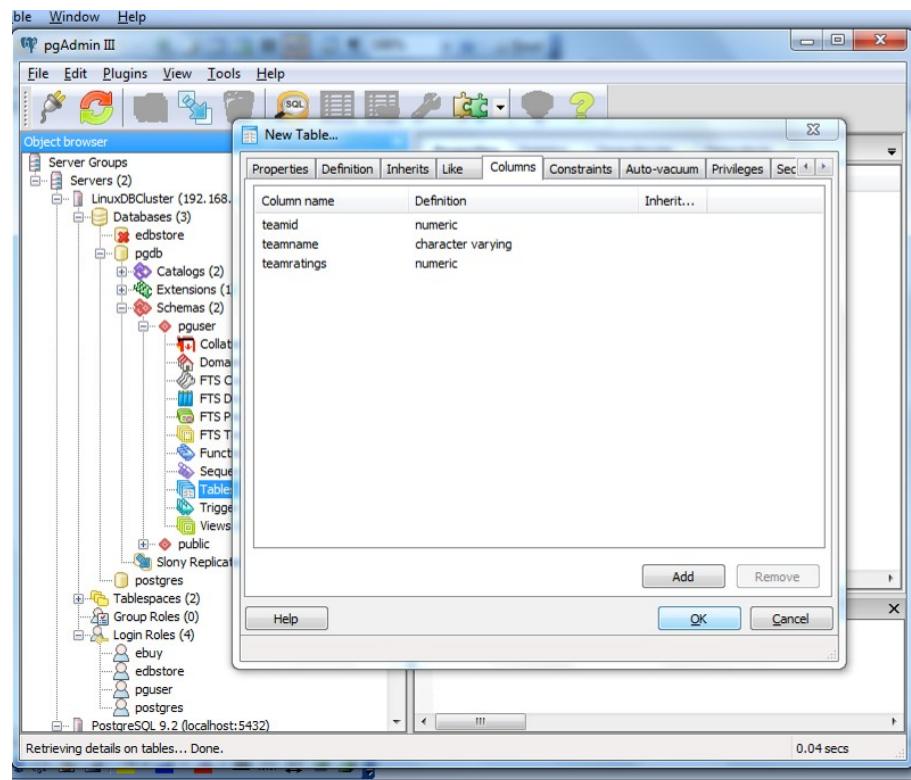
Prácticas Introductorio PostgreSQL



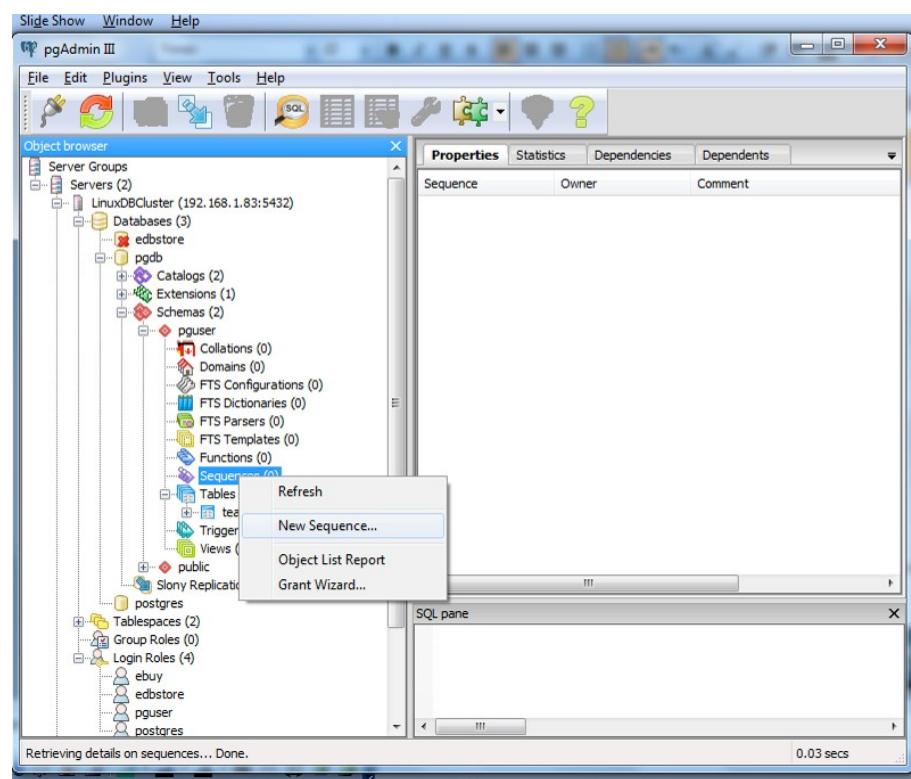
Prácticas Introductorio PostgreSQL



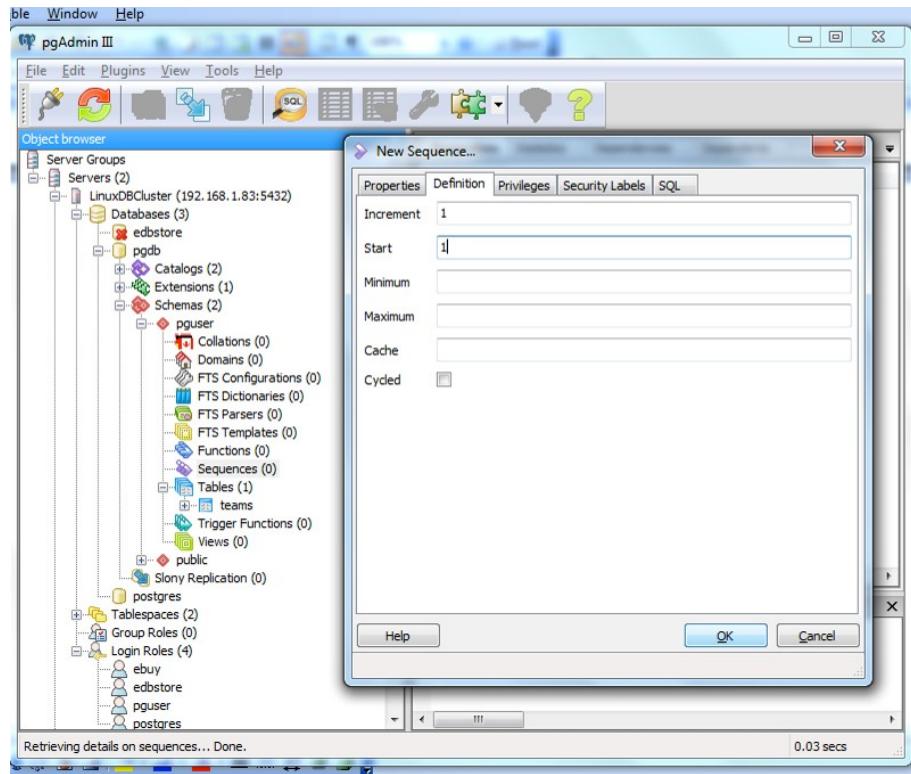
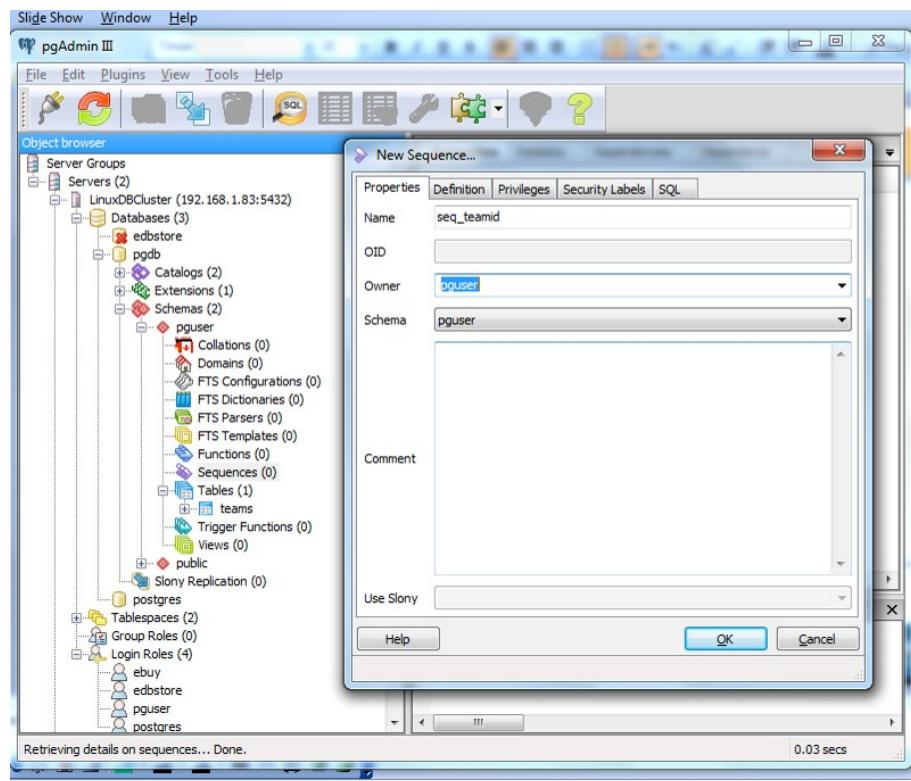
Prácticas Introductorio PostgreSQL



Secuencia: *seq_teamid* empezando en el valor 1 incrementando por 1

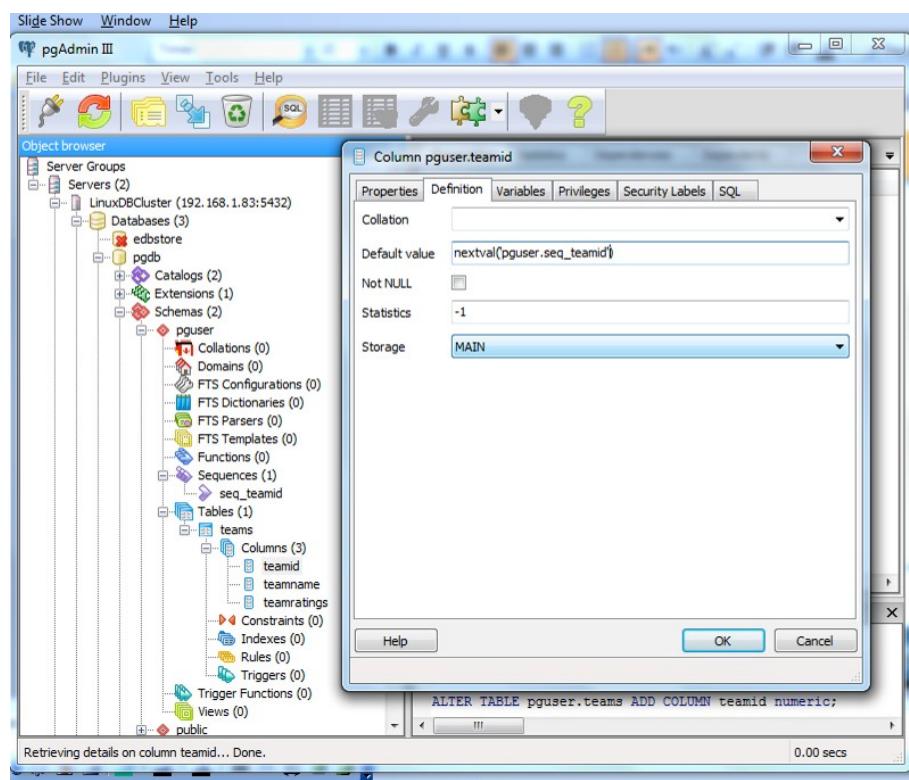
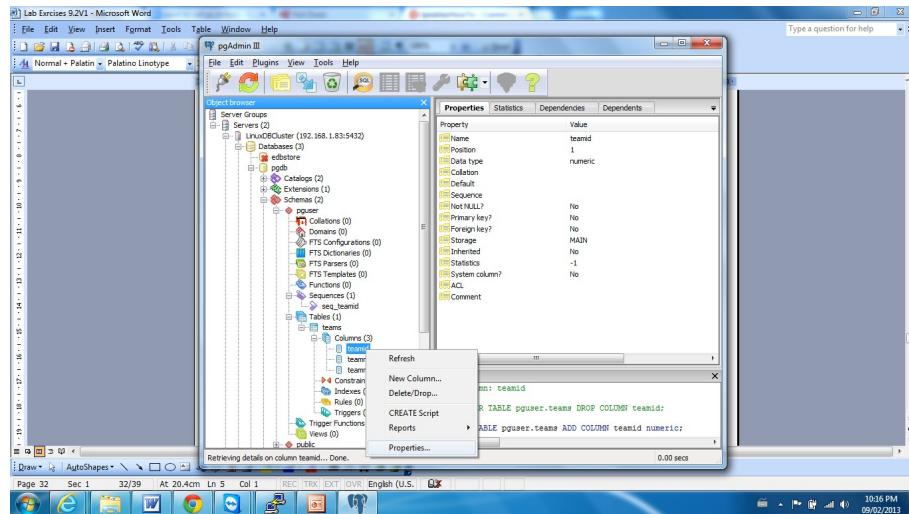


Prácticas Introductorio PostgreSQL



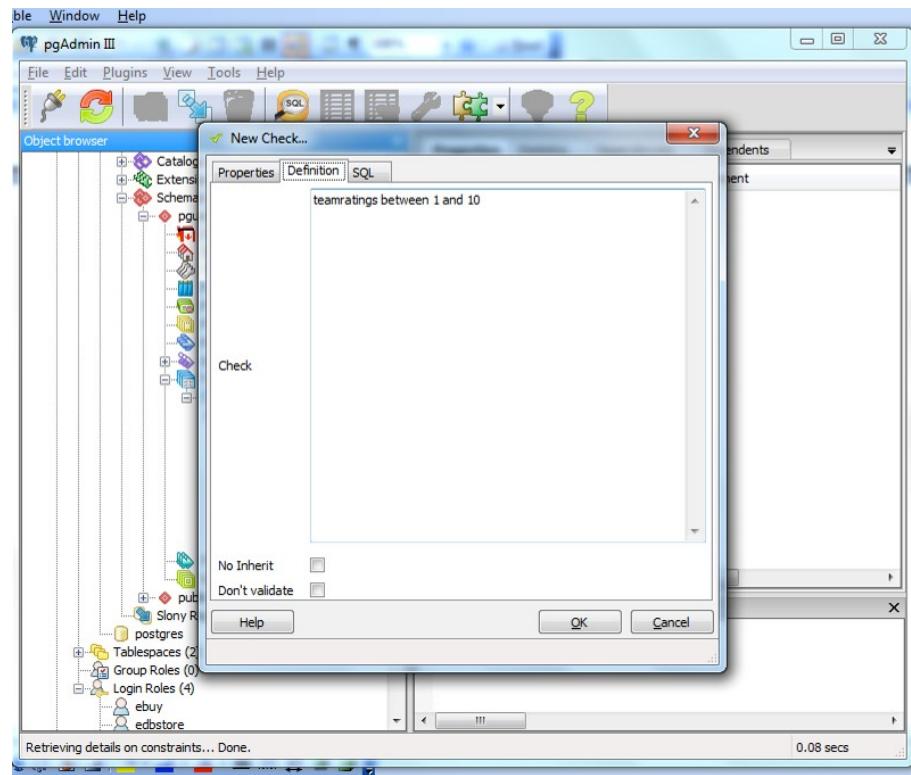
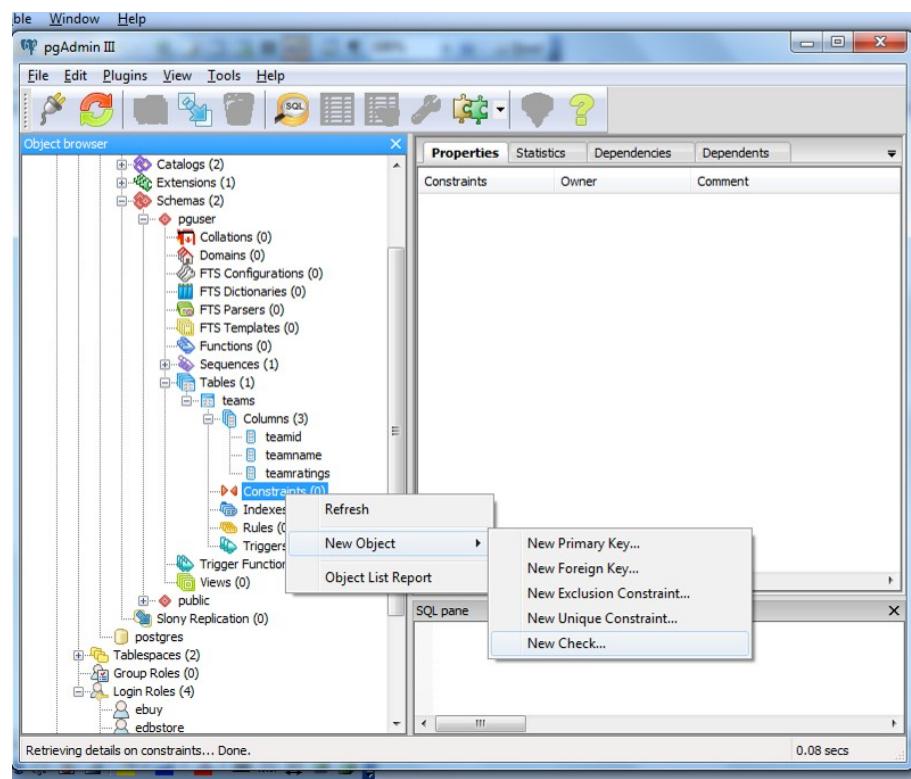
Prácticas Introductorio PostgreSQL

Columnas: Cambiar el valor por defecto para la columna TeamID a seq_teamid



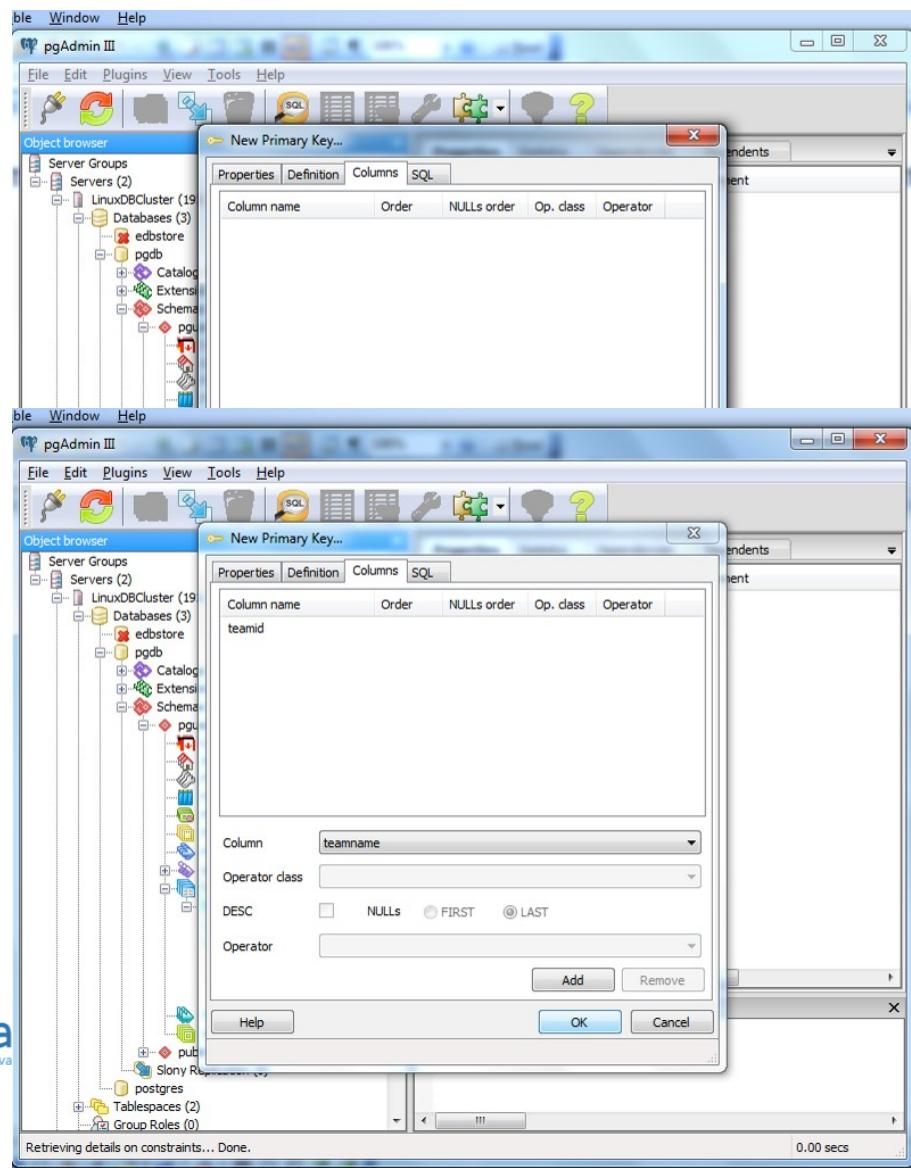
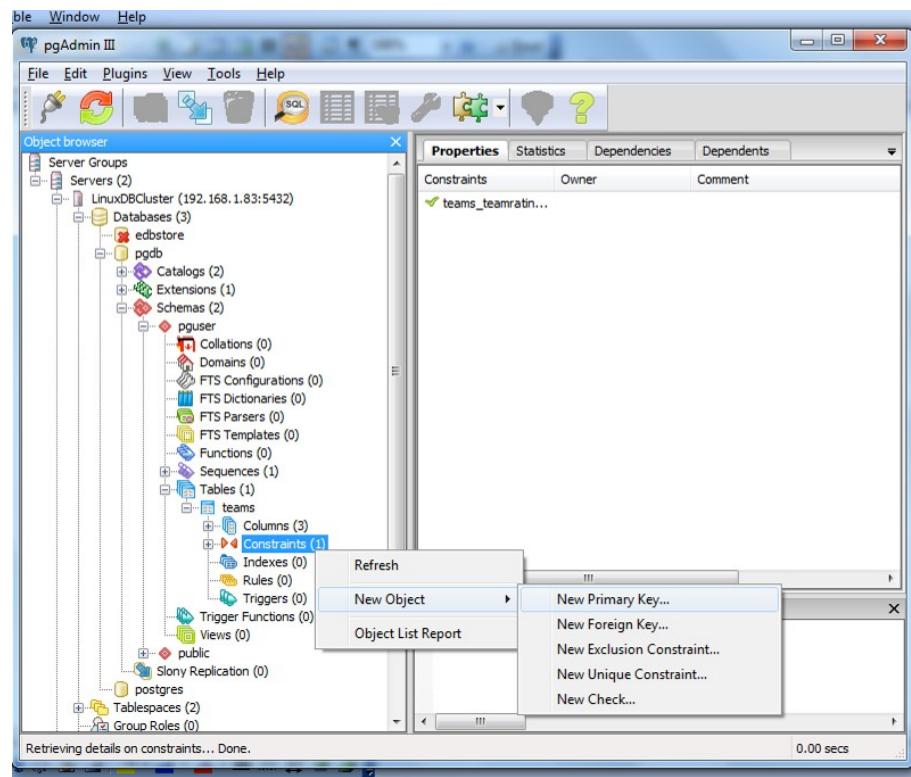
Constraint: *TeamRatings* debe estar entre 1 y 10

Prácticas Introductorio PostgreSQL



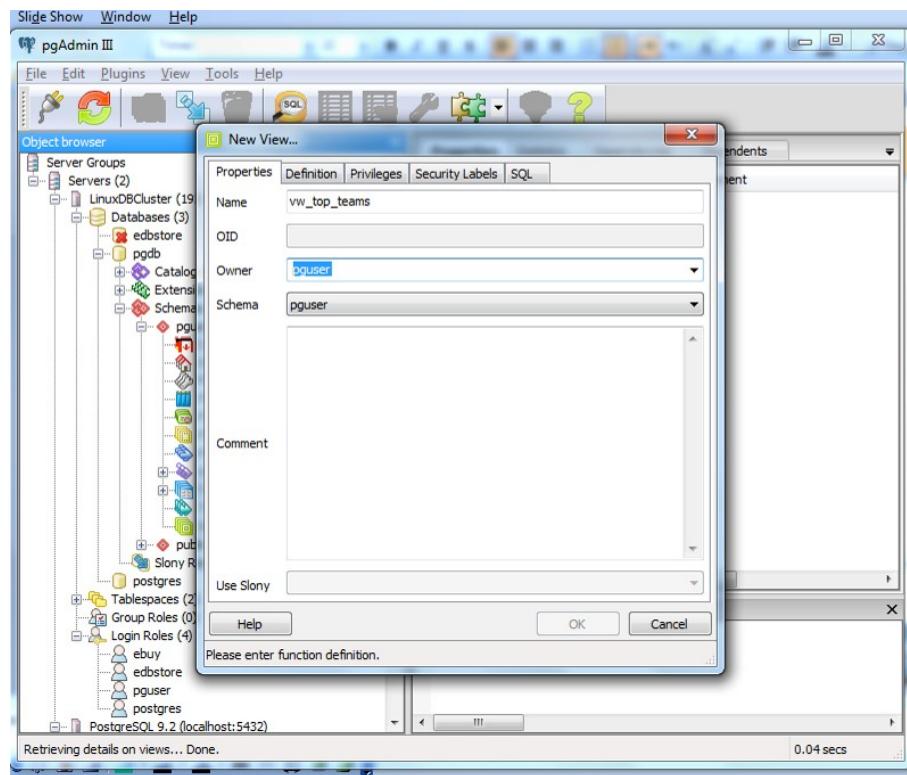
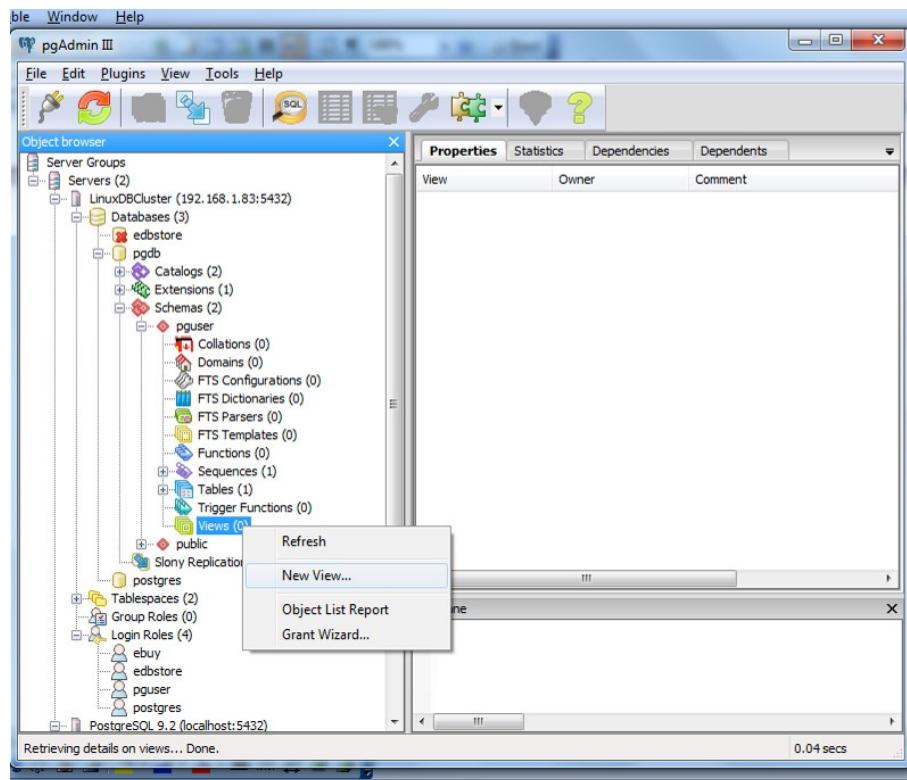
Índice: Primary key en TeamID

Prácticas Introductorio PostgreSQL

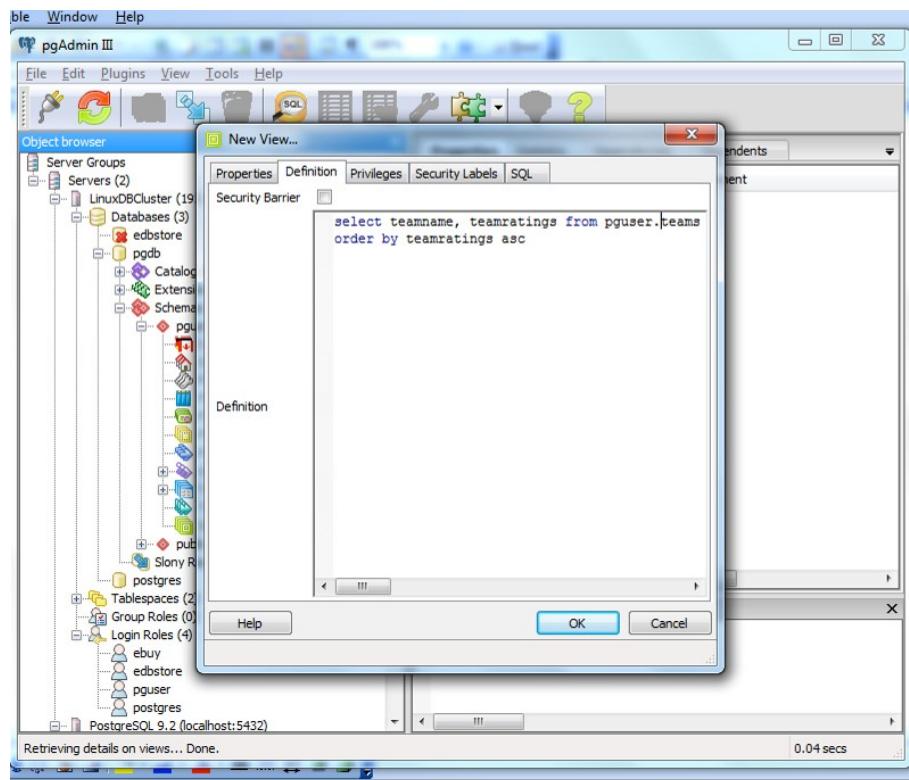


Practicas Introductorio PostgreSQL

Vista: Muestra todos los teams en orden ascendente de sus ratings. Nombre de la vista `vw_top_teams`



Prácticas Introductorio PostgreSQL



2.7.5 Ejercicio 3

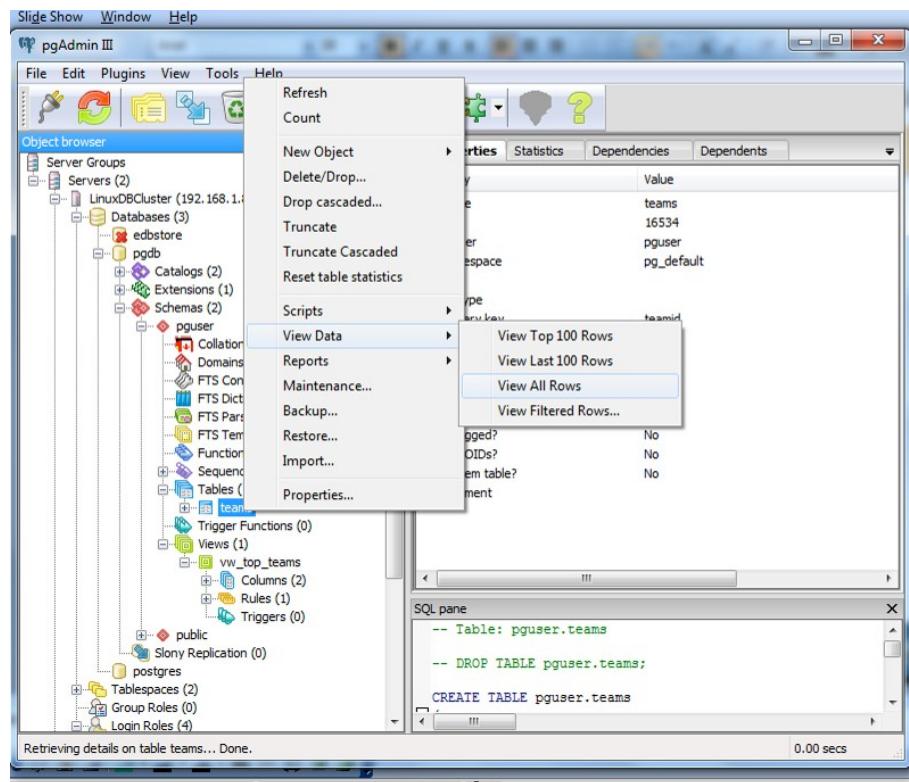
- Ver todas las filas de la tabla *teams*
- Usando el Edit data window que se abrió en el paso anterior, insertar las siguientes filas en la tabla *teams*

TeamID	TeamName	TeamRatings
Auto generated	Oilers	1
Auto generated	Rangers	6
Auto generated	Canucks	8
Auto generated	Blackhawks	5
Auto generated	Bruins	2

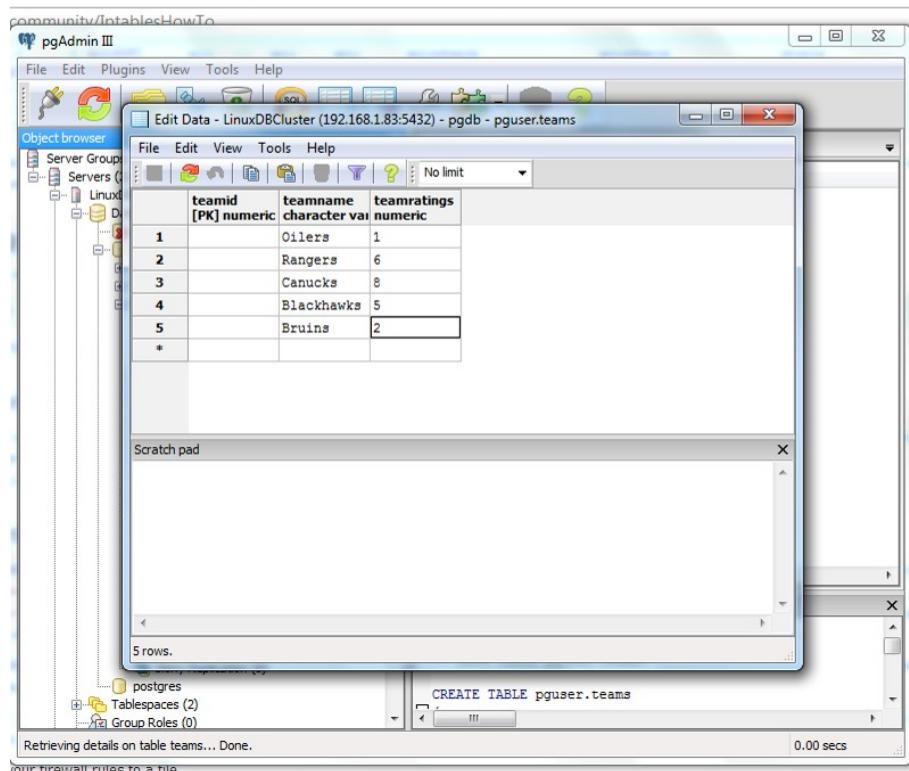
2.7.6 Solución 3

Ver todas las filas de la tabla *teams*

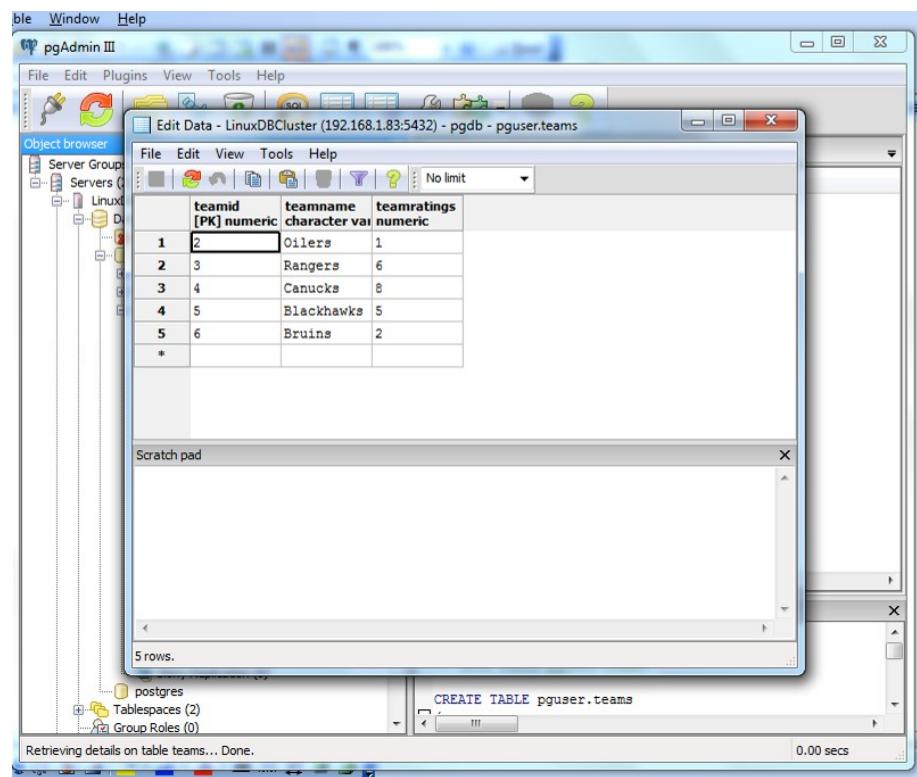
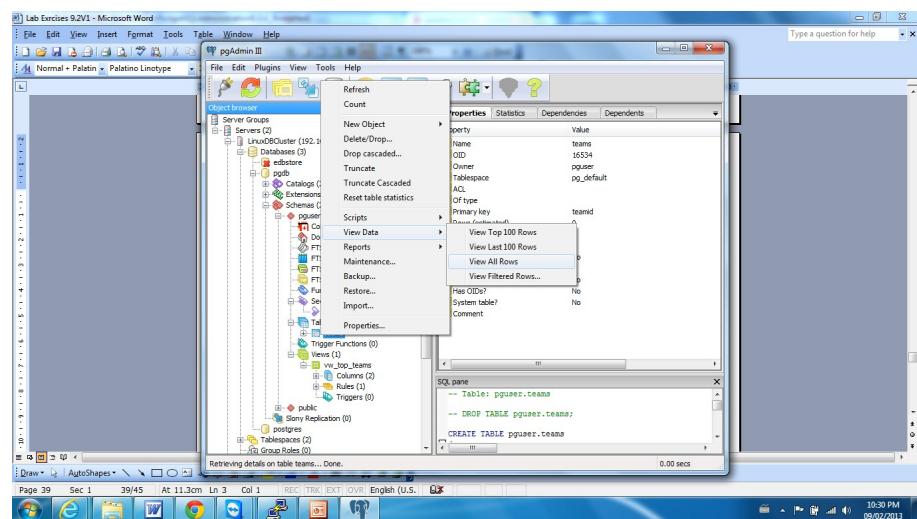
Prácticas Introductorio PostgreSQL



Usando el Edit data window que se abrió en el paso anterior, insertar las siguientes filas en la tabla teams



Prácticas Introductorio PostgreSQL



2.7.7 Ejercicio 4

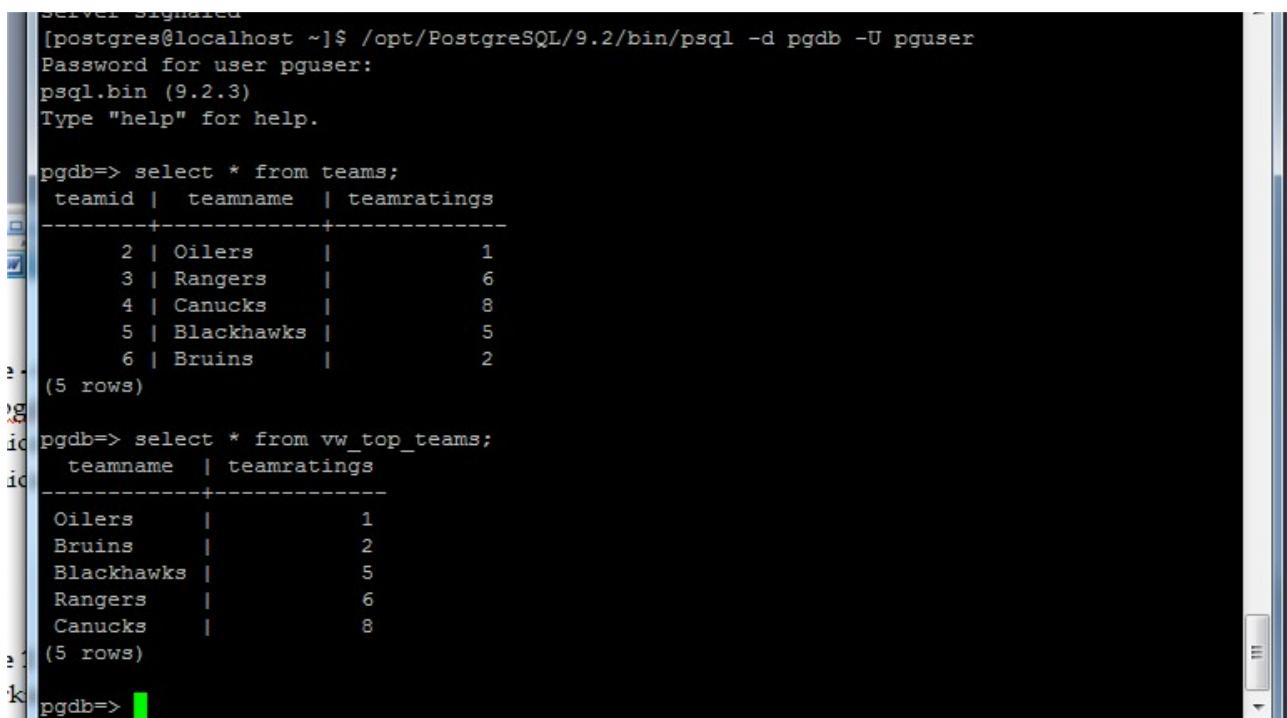
- Conectarse a la base de datos *pgdb* usando el query tool
- Usando el query builder gráfico, recuperar todas las filas de la table *teams*
- Usando el query builder gráfico, recuperar todas las filas de la vista *vw_top_teams*

2.7.8 Solución 4

Conetarse a la base de datos **pgdb** usando el query tool

Usando el query builder gráfico, recuperar todas las filas de la table **teams**

Usando el query builder gráfico, recuperar todas las filas de la vista
vw_top_teams



```
server: signalled
[postgres@localhost ~]$ /opt/PostgreSQL/9.2/bin/psql -d pgdb -U pguser
Password for user pguser:
psql.bin (9.2.3)
Type "help" for help.

pgdb=> select * from teams;
 teamid | teamname   | teamratings
-----+-----+
      2 | Oilers     |       1
      3 | Rangers    |       6
      4 | Canucks    |       8
      5 | Blackhawks |       5
      6 | Bruins     |       2
(5 rows)

pgdb=> select * from vw_top_teams;
 teamname   | teamratings
-----+-----+
 Oilers     |       1
 Bruins     |       2
 Blackhawks |       5
 Rangers    |       6
 Canucks    |       8
(5 rows)

pgdb=>
```

2.8 Modulo 8

2.8.1 Ejercicio 1

- Tu estás trabajando como DBA de PostgreSQL. Tu servidor dispone de 2 tarjetas de red con Ips 192.168.30.10 y 10.4.2.10. La primera es usada para una LAN interna y la segunda es usada para que los usuarios se conecten a un web server. Tu servidor debería aceptar conexiones TCP/IP desde ambas interfaces
- Configurar tu servidor para aceptar conexiones de la red interna y externa

2.8.2 Soluciones

Tu estás trabajando como DBA de PostgreSQL. Tu servidor dispone de 2 tarjetas de red con Ips 192.168.30.10 y 10.4.2.10. La primera es usada para una LAN interna y la segunda es usada para que los usuarios se conecten a un web server. Tu servidor debería aceptar conexiones TCP/IP desde ambas

Practicas Introductorio PostgreSQL

interfaces

Configurar tu servidor para aceptar conexiones de la red interna y externa

```
postgres@training-postgres:~$ vi /opt/PostgreSQL/9.4/data/postgresql.conf  
listen_addresses = '*'  
  
postgres@training-postgres:~$ vi /opt/PostgreSQL/9.4/data/pg_hba.conf  
host all all 192.168.30.10 md5  
host all all 10.4.2.10 md5  
  
postgres@training-postgres:~$ service postgresql-9.4 restart  
Password:  
Restarting PostgreSQL 9.4:  
  
Password:  
waiting for server to shut down.... done  
server stopped  
waiting for server to start.... done  
server started  
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory  
PostgreSQL 9.4 restarted successfully
```

2.8.3 Ejercicio 2

- Estas trabajando como DBA de PostgreSQL. Un desarrollador te muestra el siguiente error:

psql: could not connect to server: Connection refused
(0x0000274D/10061)

Is the server running on host 192.168.30.22" and accepting
TCP/IP connections on port 5432?
- Localiza el error

2.8.4 Solución 2

Estas trabajando como DBA de PostgreSQL. Un desarrollador te muestra el siguiente error:

**psql: could not connect to server: Connection refused
(0x0000274D/10061)**
**Is the server running on host 192.168.30.22" and accepting
TCP/IP connections on port 5432?**

Localiza el error

El desarrollador está tratando de conectarse a un puerto equivocado. En ese puerto no se están escuchando conexiones de PostgreSQL.

2.8.5 Ejercicio 3

- Se ha unido un nuevo desarrollador al equipo. Su ID es 89. Crea un nuevo usuario con nombre *dev89* y password *password89*. Asigne permisos para poder conectarse a la base de datos *hoplastore* y ver todas las tablas.

2.8.6 Solución 3

Se ha unido un nuevo desarrollador al equipo. Su ID es 89. Crea un nuevo usuario con nombre *dev89* y password *password89*. Asigne permisos para poder conectarse a la base de datos *hoplastore* y ver todas las tablas

```
postgres=# create user dev89 password 'password89';
CREATE ROLE
postgres=# grant connect on database hoplastore TO dev89;
GRANT
postgres=# \c hoplastore postgres
You are now connected to database "hoplastore" as user "postgres".
hoplastore=# GRANT USAGE ON SCHEMA hoplastore TO dev89;
GRANT
hoplastore=# GRANT SELECT,INSERT,DELETE,UPDATE ON ALL TABLES IN SCHEMA
hoplastore TO dev89;
GRANT
hoplastore=# \c hoplastore dev89
Password for user dev89:
You are now connected to database "hoplastore" as user "dev89".
hoplastore=> select * from dept;
ERROR: relation "dept" does not exist
LINE 1: select * from dept;
          ^
hoplastore=> select * from hoplastore.dept;
 deptno | dname   | loc
-----+-----+
 10 | ACCOUNTING | NEW YORK
 20 | RESEARCH   | DALLAS
 30 | SALES      | CHICAGO
 40 | OPERATIONS | BOSTON
(4 rows)
```

2.8.7 Ejercicio 4

- Un nuevo desarrollador se incorpora a e-music corp. Tiene una IP

Prácticas Introductorio PostgreSQL

192.168.30.89. No puede conectarse desde su máquina a la instancia de PostgreSQL debido al siguiente error:

FATAL: no pg_hba.conf entry for host "1.1.1.89", user "dev89",
database "edbstore", SSL off

- Configura el servidor para que el desarrollador pueda conectarse

2.8.8 Solución 4

Un nuevo desarrollador se incorpora a e-music corp. Tiene una IP 192.168.30.89. No puede conectarse desde su máquina a la instancia de PostgreSQL debido al siguiente error:

**FATAL: no pg_hba.conf entry for host "1.1.1.89", user "dev89",
database "edbstore", SSL off**

Configura el servidor para que el desarrollador pueda conectarse

```
postgres@training-postgres:~$ vi /opt/PostgreSQL/9.4/data/pg_hba.conf
host all all 192.168.30.89 md5

postgres@training-postgres:~$ service postgresql-9.4 restart
Password:
Restarting PostgreSQL 9.4:

Password:
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
PostgreSQL 9.4 restarted successfully
```

2.9 Modulo 9

2.9.1 Ejercicio 1

- Inicializa una sesión psql
- Accede a la base de datos *hoplastore* con el usuario *hoplastore*
- Indica si los siguientes comandos funcionan correctamente o no

SELECT ename, job, sal AS Salary FROM emp;

SELECT * FROM emp;

SELECT empno, ename, sal * 12 ANNUAL SALARY FROM emp;

2.9.2 Solución 1

Inicializa una sesión psql

Accede a la base de datos **hoplastore** con el usuario **hoplastore**

```
postgres@training-postgres:~$ psql -d hoplastore -U hoplastore  
Password for user hoplastore:  
psql.bin (9.4.4)  
Type "help" for help.
```

```
No entry for terminal type "xterm";  
using dumb terminal settings.  
hoplastore=>
```

Indica si los siguientes comandos funcionan correctamente o no

SELECT ename, job, sal AS Salary FROM emp;

TRUE

SELECT * FROM emp;

TRUE

SELECT empno, ename, sal * 12 ANNUAL SALARY FROM emp;

TRUE

2.9.3 Ejercicio 2

- Escribe una consulta para las siguientes necesidades:
- El departamento de HR necesita un informe de todos los empleados. Escribe una query que muestre nombre, número de departamento y nombre del departamento para todos los empleados
- Crea un informe que muestre el nombre de los empleados y el número de empleado con el nombre de su jefe y el número de su jefe. Etiqueta las columnas como Employee, Emp#, Manager y Mgr# respectivamente
- Crea un informe para el departamento de HR que muestre todos los empleados con nombre de empleado, número de departamento que coincidan con uno dado (por ejemplo ename = 'SMITH'), Poner unas etiquetas adecuadas

2.9.4 Solución 2

Escribe una consulta para las siguientes necesidades:

El departamento de HR necesita un informe de todos los empleados. Escribe una query que muestre nombre, número de departamento y nombre del departamento para todos los empleados

```
hoplastore=> select ename, emp.deptno, dname from emp join dept on
emp.deptno=dept.deptno;
ename | deptno | dname
-----+-----+
SMITH | 20 | RESEARCH
ALLEN | 30 | SALES
WARD | 30 | SALES
JONES | 20 | RESEARCH
MARTIN | 30 | SALES
BLAKE | 30 | SALES
CLARK | 10 | ACCOUNTING
SCOTT | 20 | RESEARCH
KING | 10 | ACCOUNTING
TURNER | 30 | SALES
ADAMS | 20 | RESEARCH
JAMES | 30 | SALES
FORD | 20 | RESEARCH
MILLER | 10 | ACCOUNTING
(14 rows)
```

Crea un informe que muestre el nombre de los empleados y el número de empleado con el nombre de su jefe y el número de su jefe. Etiqueta las columnas como Employee, Emp#, Manager y Mgr# respectivamente

```
hoplastore=> select e.empno as "Emp#", e.ename as "Employee", e.mgr as
"Manager", m.ename as "Mgr#" from emp e join
emp m on e.mgr=m.empno;
Emp# | Employee | Manager | Mgr#
-----+-----+
7369 | SMITH | 7902 | FORD
7499 | ALLEN | 7698 | BLAKE
7521 | WARD | 7698 | BLAKE
7566 | JONES | 7839 | KING
7654 | MARTIN | 7698 | BLAKE
7698 | BLAKE | 7839 | KING
7782 | CLARK | 7839 | KING
7788 | SCOTT | 7566 | JONES
7844 | TURNER | 7698 | BLAKE
7876 | ADAMS | 7788 | SCOTT
7900 | JAMES | 7698 | BLAKE
7902 | FORD | 7566 | JONES
7934 | MILLER | 7782 | CLARK
(13 rows)
```

Crea un informe para el departamento de HR que muestre todos los empleados con nombre de empleado, número de departamento que coincidan con uno dado (por ejemplo ename = 'SMITH'), Poner unas etiquetas adecuadas

```
hoplastore=> select ename, deptno from emp where deptno = (select deptno from emp where ename='SMITH');
ename | deptno
-----+
SMITH | 20
JONES | 20
SCOTT | 20
ADAMS | 20
FORD | 20
(5 rows)
```

2.9.5 Ejercicio 3

- Escribe una query que muestre el número de empleado y nombre de todos los empleados que trabajen en un departamento con un empleado cuyo nombre contenga la “E”
- Cambia el nombre del empleado *7566* a *Drexler*
- Cambia el salario a 1.000 \$ para todos los empleados quienes cobren menos de 900 \$
- Verifica tus cambios sobre la tabla
- Borra el usuario *MILLER* de la tabla *EMP*

2.9.6 Solución 3

Escribe una query que muestre el número de empleado y nombre de todos los empleados que trabajen en un departamento con un empleado cuyo nombre contenga la “E”

```
hoplastore=> select ename, deptno from emp where deptno IN (select deptno from emp where ename like '%E%');
ename | deptno
-----+
SMITH | 20
ALLEN | 30
WARD | 30
JONES | 20
MARTIN | 30
BLAKE | 30
```

Practicas Introductorio PostgreSQL

```
CLARK | 10
SCOTT | 20
KING | 10
TURNER | 30
ADAMS | 20
JAMES | 30
FORD | 20
MILLER | 10
(14 rows)
```

Cambia el nombre del empleado **7566** a **DREXLER**

```
hoplastore=> UPDATE emp set ename='DREXLER' where empno=7566;
UPDATE 1
```

Cambia el salario a **1.000 \$** para todos los empleados quienes cobren menos de **900 \$**

```
hoplastore=> UPDATE emp set sal=1000.00 where sal<900.00;
UPDATE 1
```

Verifica tus cambios sobre la tabla

```
hoplastore=> table emp;
   empno | ename | job | mgr |      hiredate      |    sal    | comm | deptno
-----+-----+-----+-----+-----+-----+-----+
 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 00:00:00 | 1600.00 | 300.00 |  30
 7521 | WARD  | SALESMAN | 7698 | 1981-02-22 00:00:00 | 1250.00 | 500.00 | 30
 7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 00:00:00 | 1250.00 | 1400.00 | 30
 7698 | BLAKE  | MANAGER | 7839 | 1981-05-01 00:00:00 | 2850.00 |        | 30
 7782 | CLARK  | MANAGER | 7839 | 1981-06-09 00:00:00 | 2450.00 |        | 10
 7788 | SCOTT  | ANALYST | 7566 | 1987-04-19 00:00:00 | 3000.00 |        | 20
 7839 | KING    | PRESIDENT |      | 1981-11-17 00:00:00 | 5000.00 |        | 10
 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 00:00:00 | 1500.00 | 0.00  | 30
 7876 | ADAMS  | CLERK   | 7788 | 1987-05-23 00:00:00 | 1100.00 |        | 20
 7900 | JAMES  | CLERK   | 7698 | 1981-12-03 00:00:00 | 950.00 |        | 30
 7902 | FORD   | ANALYST | 7566 | 1981-12-03 00:00:00 | 3000.00 |        | 20
 7934 | MILLER | CLERK   | 7782 | 1982-01-23 00:00:00 | 1300.00 |        | 10
 7566 | DREXLER | MANAGER | 7839 | 1981-04-02 00:00:00 | 2975.00 |        | 20
 7369 | SMITH  | CLERK   | 7902 | 1980-12-17 00:00:00 | 1000.00 |        | 20
(14 rows)
```

Borra el usuario **MILLER** de la tabla **EMP**

```
hoplastore=> delete from emp where ename='MILLER';
DELETE 1
```

2.9.7 Ejercicio 4

- Crea la tabla *EMP2* basada en la estructura de la tabla *EMP*. Incluye solo las columnas *empno*, *ename*, *sal* y *deptno*. Renombra las columnas en la nueva tabla como *ID*, *FIRST_NAME*, *SALARY* y *DEPTID* respectivamente.
- El departamento de HR quiere ocultar cierta información en la tabla *EMP*. Ellos quieren una vista llamada *EMPVU* basada en las columnas número de empleado, nombre del empleado y número de departamento. Desean que la columna nombre de empleado se renombre como *EMPLOYEE*.
- Confirma que la vista está correcta. Muestra el contenido de la vista *EMPVU*.
- Usando la vista *EMPVU*, escribe una query para el departamento de ventas que muestre todos los empleados y sus números de departamento.

2.9.8 Solución 4

Crea la tabla *EMP2* basada en la estructura de la tabla *EMP*. Incluye solo las columnas *empno*, *ename*, *sal* y *deptno*. Renombra las columnas en la nueva tabla como *ID*, *FIRST_NAME*, *SALARY* y *DEPTID* respectivamente.

```
hoplastore=> CREATE TABLE emp2(ID, FIRST_NAME, SALARY, DEPTID) AS select  
empno, ename, sal, deptno from emp;  
SELECT 13
```

El departamento de HR quiere ocultar cierta información en la tabla *EMP*. Ellos quieren una vista llamada *EMPVU* basada en las columnas número de empleado, nombre del empleado y número de departamento. Desean que la columna nombre de empleado se renombre como *EMPLOYEE*.

```
hoplastore=> CREATE VIEW empvu as select empno, ename as employee, deptno  
from emp;  
CREATE VIEW
```

Confirma que la vista está correcta. Muestra el contenido de la vista *EMPVU*.

```
hoplastore=> select * from empvu;  
empno | employee | deptno  
-----+-----+  
7499 | ALLEN | 30  
7521 | WARD | 30  
7654 | MARTIN | 30  
7698 | BLAKE | 30  
7782 | CLARK | 10  
7788 | SCOTT | 20  
7839 | KING | 10  
7844 | TURNER | 30
```

Practicas Introductorio PostgreSQL

```
7876 | ADAMS   | 20
7900 | JAMES   | 30
7902 | FORD    | 20
7566 | DREXLER | 20
7369 | SMITH   | 20
(13 rows)
```

Usando la vista *EMPVU*, escribe una query para el departamento de ventas que muestre todos los empleados y sus números de departamento.

```
hoplastore=> select employee,deptno from empvu;
employee | deptno
-----+-
ALLEN   | 30
WARD     | 30
MARTIN   | 30
BLAKE    | 30
CLARK    | 10
SCOTT    | 20
KING     | 10
TURNER   | 30
ADAMS    | 20
JAMES    | 30
FORD     | 20
DREXLER  | 20
SMITH    | 20
(13 rows)
```

2.10 Modulo 10

2.10.1 Ejercicio 1

- Se debe aplicar una politica de backup para la base de datos *hoplastore*.
- Como *root*, crear la carpeta */pgbackup* y asignar permisos de propietario al usuario *postgres* usando la utilidad *chown*.
- Realizar un dump completo de la base de datos *hoplastore* con la utilidad *pg_dump*. El dump debe estar en formato plano.
- Renombrar el backup como *hoplastore_full.sql*.

2.10.2 Solución 1

Se debe aplicar una politica de backup para la base de datos *hoplastore*.

Como *root*, crear la carpeta */pgbackup* y asignar permisos de propietario al usuario *postgres* usando la utilidad *chown*.

Prácticas Introductorio PostgreSQL

Realizar un dump completo de la base de datos *hoplastore* con la utilidad *pg_dump*. El dump debe estar en formato plano.

Renombrar el backup como *hoplastore_full.sql*.

```
root@training-postgres:~# mkdir /pgbackup
root@training-postgres:~# chown postgres:postgres /pgbackup/
root@training-postgres:~# su - postgres
postgres@training-postgres:~$ pg_dump -f /pgbackup/hoplastore_full.sql -U postgres
hoplastore
```

2.10.3 Ejercicio 2

- Realiza un backup del esquema hoplastore de la base de datos *hoplastore* y renómbralo como *hoplastore_schema.sql*.
- Ahora realiza un backup solo de los datos de la base de datos *hoplastore* completa. Usa el comando para extraer los *inserts* en lugar de *copy* y renómbralo como *hoplastore_data.sql*.
- Realiza un backup completo de la tabla *customers* y renómbralo como *hoplastore_customers.sql*.

2.10.4 Solución

Realiza un backup del esquema hoplastore de la base de datos *hoplastore* y renómbralo como *hoplastore_schema.sql*.

```
postgres@training-postgres:~$ pg_dump -n hoplastore -f
/pgbackup/hoplastore_schema.sql -U postgres hoplastore
```

Ahora realiza un backup solo de los datos de la base de datos *hoplastore* completa. Usa el comando para extraer los *inserts* en lugar de *copy* y renómbralo como *hoplastore_data.sql*.

```
postgres@training-postgres:~$ pg_dump --inserts --disable-triggers -f
/pgbackup/hoplastore_data.sql -U postgres hoplastore
```

Realiza un backup completo de la tabla *customers* y renómbralo como *hoplastore_customers.sql*.

```
postgres@training-postgres:~$ pg_dump -t hoplastore.customers -f
/pgbackup/hoplastore_customers.sql -U postgres hoplastore
```

2.10.5 Ejercicio 3

- Realiza un backup completo de la base de datos *hoplastore* en formato

Prácticas Introductorio PostgreSQL

comprimido usando la utilidad *pg_dump*, renómbralo como *hoplastore_full_fc.dmp*.

- Realiza un backup completo del cluster usando la utilidad *pg_dumpall*. Recuerda que *pg_dumpall* solo soporta formato texto plano. Renómbralo como *hopladata.sql*.

2.10.6 Solución 3

Realiza un backup completo de la base de datos *hoplastore* en formato comprimido usando la utilidad *pg_dump*, renómbralo como *hoplastore_full_fc.dmp*.

```
postgres@training-postgres:~$ pg_dump -Fc -f /pgbackup/hoplastore_full_fc.dmp -U postgres hoplastore
```

Realiza un backup completo del cluster usando la utilidad *pg_dumpall*. Recuerda que *pg_dumpall* solo soporta formato texto plano. Renómbralo como *hopladata.sql*.

```
postgres@training-postgres:~$ pg_dumpall -p 5432 -U postgres > /pgbackup/hopladata.sql
```

NOTA: Si no se ha configurado pgpass o se ha incluido el modo de autenticación trust. Puede solicitar la contraseña por cada base de datos que va a copiar.

2.10.7 Ejercicio 4

- En el siguiente ejercicio se demostrará la habilidad de restaurar una base de datos.
- Elimina la base de datos *hoplastore*.
- Crea una base de datos *hoplastore* con el owner *hoplastore*.
- Restaura el full dump que se tomó en uno de los ejercicios anteriores *hoplastore_full.sql*. Verifica que se han creado los objetos y sus propietarios.
- Elimina la base de datos *hoplastore*.
- Crea una base de datos *hoplastore* con el owner *hoplastore*.
- Restaura el full dump comprimido que se tomó en uno de los ejercicios anteriores *hoplastore_full_fc.sql*. Verifica que se han creado los objetos y sus propietarios.

2.10.8 Solución 4

En el siguiente ejercicio se demostrará la habilidad de restaurar una base de datos.

Elimina la base de datos *hoplastore*.

```
postgres@training-postgres:~$ dropdb hoplastore
```

Crea una base de datos *hoplastore* con el owner *hoplastore*.

```
postgres@training-postgres:~$ createdb -O hoplastore hoplastore
```

Restaura el full dump que se tomó en uno de los ejercicios anteriores *hoplastore_full.sql*. Verifica que se han creado los objetos y sus propietarios.

```
postgres@training-postgres:~$ psql -f /pgbackup/hoplastore_full.sql -d hoplastore -U hoplastore
```

```
postgres@training-postgres:~$ psql -U hoplastore -d hoplastore
Password for user hoplastore:
psql.bin (9.4.4)
Type "help" for help.
```

```
No entry for terminal type "xterm";
using dumb terminal settings.
```

```
hoplastore=> \dt
      List of relations
 Schema |   Name    | Type | Owner
-----+-----+-----+
 hoplastore | categories | table | hoplastore
 hoplastore | cust_hist | table | hoplastore
 hoplastore | customers | table | hoplastore
 hoplastore | dept      | table | hoplastore
 hoplastore | emp       | table | hoplastore
 hoplastore | emp2      | table | hoplastore
 hoplastore | inventory | table | hoplastore
 hoplastore | job_grd   | table | hoplastore
 hoplastore | jobhist   | table | hoplastore
 hoplastore | locations  | table | hoplastore
 hoplastore | orderlines | table | hoplastore
 hoplastore | orders     | table | hoplastore
 hoplastore | products   | table | hoplastore
 hoplastore | reorder    | table | hoplastore
(14 rows)
```

Elimina la base de datos *hoplastore*.

```
postgres@training-postgres:~$ dropdb hoplastore
```

Crea una base de datos *hoplastore* con el owner *hoplastore*.

Practicas Introductorio PostgreSQL

```
postgres@training-postgres:~$ createdb -O hoplastore hoplastore
```

Restaura el full dump comprimido que se tomó en uno de los ejercicios anteriores *hoplastore_full_fc.sql*. Verifica que se han creado los objetos y sus propietarios.

```
postgres@training-postgres:~$ pg_restore -d hoplastore  
/pgbackup/hoplastore_full_fc.dmp
```

```
postgres@training-postgres:~$ psql -U hoplastore -d hoplastore  
Password for user  
hoplastore:  
psql.bin (9.4.4)  
Type "help" for help.
```

```
No entry for terminal type "xterm";  
using dumb terminal settings.  
hoplastore=> \dt  
      List of relations  
 Schema |   Name   | Type | Owner  
-----+-----+-----  
hoplastore | categories | table | hoplastore  
hoplastore | cust_hist | table | hoplastore  
hoplastore | customers | table | hoplastore  
hoplastore | dept     | table | hoplastore  
hoplastore | emp      | table | hoplastore  
hoplastore | emp2     | table | hoplastore  
hoplastore | inventory | table | hoplastore  
hoplastore | job_grd  | table | hoplastore  
hoplastore | jobhist  | table | hoplastore  
hoplastore | locations | table | hoplastore  
hoplastore | orderlines| table | hoplastore  
hoplastore | orders    | table | hoplastore  
hoplastore | products  | table | hoplastore  
hoplastore | reorder   | table | hoplastore  
(14 rows)
```

2.10.9 Ejercicio 5

- Crea un directorio que se llame */opt/arch* cuyo propietario sea *postgres*.
- Configura el cluster de postgres para que se ejecute en modo archive y el archive log sea el directorio arriba creado.
- Realiza un full online backup de tu cluster en */pgbackup* usando la utilidad *pg_basebackup*.

2.10.10 Solución 5

Crea un directorio que se llame **/opt/arch** cuyo propietario sea **postgres**.

```
root@training-postgres:~# mkdir /opt/arch  
root@training-postgres:~# chown postgres:postgres /opt/arch/  
root@training-postgres:~# su - postgres
```

Configura el cluster de **postgres** para que se ejecute en modo archive y el archive log sea el directorio arriba creado.

```
postgres@training-postgres:~$ vi /opt/PostgreSQL/9.4/data/postgresql.conf
```

```
wal_level = archive  
archive_mode = on  
archive_command = 'cp %p /opt/arch/%f'  
max_wal_senders = 3
```

```
postgres@training-postgres:~$ vi /opt/PostgreSQL/9.4/data/pg_hba.conf
```

```
local  replication  postgres          md5  
host   replication  postgres  127.0.0.1/32  md5  
host   replication  postgres  ::1/128       md5
```

```
postgres@training-postgres:~$ service postgresql-9.4 restart
```

```
Password:
```

```
su: Authentication failure
```

```
Restarting PostgreSQL 9.4:
```

```
Password:
```

```
waiting for server to shut down.... done
```

```
server stopped
```

```
waiting for server to start.... done
```

```
server started
```

```
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
```

```
PostgreSQL 9.4 restarted successfully
```

Realiza un full online backup de tu cluster en **/pgbackup** usando la utilidad **pg_basebackup**.

```
postgres@training-postgres:~$ pg_basebackup -h localhost -D /pgbackup/data  
NOTICE: pg_stop_backup complete, all required WAL segments have been archived
```

2.10.11 Ejercicio 6

- Una instancia puede sufrir diferentes tipos de fallos. Recupera tu cluster de las siguientes simulaciones:
- Recupera desde la pérdida del fichero *postgresql.conf*.

Prácticas Introductorio PostgreSQL

- Recupera desde la pérdida de la table *emp*.
- Recupera desde el borrado por error de la tabla *cust_hist*.

2.10.12 Solución 6

Una instancia puede sufrir diferentes tipos de fallos. Recupera tu cluster de las siguientes simulaciones:

Recupera desde la pérdida del fichero *postgresql.conf*.

```
postgres@training-postgres:~$ service postgresql-9.4 stop
Stopping PostgreSQL 9.4:
Password:
waiting for server to shut down.... done
server stopped

postgres@training-postgres:~$ cp /pgbackup/data/postgresql.conf
/opt/PostgreSQL/9.4/data/postgresql.conf

postgres@training-postgres:~$ service postgresql-9.4 start
Password:
Starting PostgreSQL 9.4:

Password:
waiting for server to start.... done
server started
touch: cannot touch '/var/lock/subsys/postgresql-9.4': No such file or directory
PostgreSQL 9.4 started successfully
```

Recupera desde la pérdida de la table *emp*.

```
postgres@training-postgres:~$ psql -d hoplastore -U hoplastore
Password for user hoplastore:
psql.bin (9.4.4)
Type "help" for help.

No entry for terminal type "xterm";
using dumb terminal settings.

hoplastore=> select count(*) from emp;
count
-----
 13
(1 row)

hoplastore=> drop table emp;
ERROR: cannot drop table emp because other objects depend on it
DETAIL: view empvu depends on table emp
view salesemp depends on table emp
```

Practicas Introductorio PostgreSQL

```
constraint jobhist_ref_emp_fk on table jobhist depends on table emp  
HINT: Use DROP ... CASCADE to drop the dependent objects too.
```

```
hoplastore=> drop table emp cascade;  
NOTICE: drop cascades to 3 other objects  
DETAIL: drop cascades to view empvu  
drop cascades to view salesemp  
drop cascades to constraint jobhist_ref_emp_fk on table jobhist  
DROP TABLE
```

```
postgres@training-postgres:~$ pg_restore -Fc -n hoplastore -t emp -v -d hoplastore -U  
hoplastore /pgbackup/hoplastore_full_fc.dmp  
pg_restore: connecting to database for restore  
Password:  
pg_restore: creating TABLE emp  
pg_restore: processing data for table "emp"  
pg_restore: setting owner and privileges for TABLE emp  
pg_restore: setting owner and privileges for TABLE DATA emp
```

```
postgres@training-postgres:~$ psql -d hoplastore -U hoplastore  
Password for user hoplastore:  
psql.bin (9.4.4)  
Type "help" for help.
```

No entry for terminal type "xterm";
using dumb terminal settings.

```
hoplastore=> \dt emp  
      List of relations  
 Schema | Name | Type | Owner  
-----+-----+-----+  
hoplastore | emp | table | hoplastore  
(1 row)
```

```
hoplastore=> select count(*) from emp;  
 count  
-----  
 13  
(1 row)
```

Recupera desde el borrado por error de la tabla **cust_hist**.

```
# Se comprueba que la tabla tiene datos
```

```
hoplastore=> select count(*) from cust_hist;  
 count  
-----  
 60350  
(1 row)
```

```
# Se comprueba el momento temporal actual (anterior al borrado accidental de la
```

Practicas Introductorio PostgreSQL

```
tabla)

hoplastore=> select now();
      now
-----
2016-02-14 11:31:49.798849+01
(1 row)

# Se borra la tabla por accidente

hoplastore=> drop table cust_hist ;
DROP TABLE

# Se para el servicio

postgres@training-postgres:~$ service postgresql-9.4 stop
Stopping PostgreSQL 9.4:
Password:
waiting for server to shut down.... done
server stopped

# Se hace una copia de seguridad

root@training-postgres:~# mv /opt/PostgreSQL/9.4/data/ /pgbackup/crashdata

# Se copia la base del cluster del backup físico tomado previamente

root@training-postgres:~# cp -rp /pgbackup/data/ /opt/PostgreSQL/9.4/data

# Se copian los últimos WAL logs que están en la carpeta de crash

root@training-postgres:~# cp -p /pgbackup/crashdata/data/pg_xlog/0*
/opt/PostgreSQL/9.4/data/pg_xlog/

# Se edita el fichero de recuperación

root@training-postgres:~# vi /opt/PostgreSQL/9.4/data/recovery.conf

restore_command='cp /opt/arch/%f %p'
recovery_target_time='2016-02-14 11:31:49'

# Se cambian los permisos

root@training-postgres:~# chown postgres:postgres
/opt/PostgreSQL/9.4/data/recovery.conf

# Se arranca el servicio y el cluster se pone en modo recuperación

postgres@training-postgres:~$ service postgresql-9.4 start
Password:
Starting PostgreSQL 9.4:
```

Practicas Introductorio PostgreSQL

```
Password:  
waiting for server to start..... done  
server started  
PostgreSQL 9.4 started successfully  
  
# Se comprueba la existencia de la tabla con datos hasta el momento anterior al  
borrado  
  
hoplastore=> select count(*) from cust_hist;  
count  
-----  
60350  
(1 row)
```

2.11 Modulo 11

2.11.1 Ejercicio 1

- Crea un explain plan para las siguientes queries. Crear un índice sobre la columna empno antes de comenzar.
- *Select * from emp where empno = 7566*
- *Select * from emp where empno > 7566*
- *Select * from emp where empno is not null 7566*

2.11.2 Solución 1

Crea un explain plan para las siguientes queries. Crear un índice sobre la columna empno antes de comenzar.

```
# Se crea un índice para esta prueba  
  
hoplastore=> create index inx_emp_empno on emp (empno);  
CREATE INDEX
```

Select * from emp where empno = 7566

```
hoplastore=> explain Select * from emp where empno = 7566;  
          QUERY PLAN  
-----  
Seq Scan on emp  (cost=0.00..1.16 rows=1 width=146)  
  Filter: (empno = 7566::numeric)  
(2 rows)
```

Select * from emp where empno > 7566

```
hoplastore=> explain Select * from emp where empno > 7566;
   QUERY PLAN
```

```
-----  
Seq Scan on emp (cost=0.00..1.16 rows=4 width=146)  
  Filter: (empno > 7566::numeric)  
(2 rows)
```

Select * from emp where empno is not null 7566

```
hoplastore=> explain Select * from emp where empno is not null;
   QUERY PLAN
```

```
-----  
Seq Scan on emp (cost=0.00..1.13 rows=13 width=146)  
  Filter: (empno IS NOT NULL)  
(2 rows)
```

2.11.3 Ejercicio 2

- En el ejercicio previo se puede observar que a pesar de tener índice las queries van por *seq scan* al resaltarles los costes inferiores. Configura el servidor para forzar a usar a la sesión actual a emplear *index scan*.

2.11.4 Solución 2

En el ejercicio previo se puede observar que a pesar de tener índice las queries van por *seq scan* al resaltarles los costes inferiores. Configura el servidor para forzar a usar a la sesión actual a emplear *index scan*.

```
# Configuración de parámetros a nivel de sesión para evitar en la medida de lo  
posible el seq scan atribuyéndole costes más altos.
```

```
hoplastore=> set enable_seqscan=false;  
SET  
hoplastore=> set enable_bitmapscan=false;  
SET
```

```
# Ejecución del explain de nuevo
```

```
hoplastore=> explain select * from emp where empno=7566;
   QUERY PLAN
```

```
-----  
Index Scan using inx_emp_empno on emp (cost=0.14..8.15 rows=1 width=146)  
  Index Cond: (empno = 7566::numeric)  
(2 rows)
```

2.11.5 Ejercicio 3

- Revisando las estadísticas, observas que algunas tablas no son automáticamente mantenidas por el *autovacuum*. Decides crear una tarea de mantenimiento para algunas de estas tablas. Escribe un script que realice el mantenimiento de los siguientes ítems:
- Reclamar el espacio obsoleto de la tabla *customers*
- Actualizar las estadísticas en las tablas *emp* y *dept*
- Marcar todas las filas obsoletas en la tabla *orders* para ser reutilizadas
- Ejecutar el script sobre la base de datos *hoplastore*

2.11.6 Solución 3

Revisando las estadísticas, observas que algunas tablas no son automáticamente mantenidas por el *autovacuum*. Decides crear una tarea de mantenimiento para algunas de estas tablas. Escribe un script que realice el mantenimiento de los siguientes ítems:

Reclamar el espacio obsoleto de la tabla *customers*

Actualizar las estadísticas en las tablas *emp* y *dept*

Marcar todas las filas obsoletas en la tabla *orders* para ser reutilizadas

```
# Se crea el script  
  
postgres@training-postgres:~$ vi /tmp/hoplastore_mnts_script.sql  
  
# Se añaden las tareas  
  
VACUUM FULL customers;  
ANALYZE emp;  
ANALYZE dept;  
VACUUM orders;
```

Ejecutar el script sobre la base de datos *hoplastore*

```
postgres@training-postgres:~$ psql -f /tmp/hoplastore_mnts_script.sql -d hoplastore  
-U hoplastore  
Password for user hoplastore:  
VACUUM  
ANALYZE  
ANALYZE  
VACUUM
```

2.11.7 Ejercicio 4

- El índice *ix_orderlines_orderid* sobre las columnas (*orderid*, *orderlineid*) de la tabla *orderlines* está teniendo un rendimiento muy bajo. Escribe un comando para reindexarlo de manera que mejore su comportamiento.

2.11.8 Solución 4

El índice *ix_orderlines_orderid* sobre las columnas (*orderid*, *orderlineid*) de la tabla *orderlines* está teniendo un rendimiento muy bajo. Escribe un comando para reindexarlo de manera que mejore su comportamiento.

```
hoplastore=> reindex index ix_orderlines_orderid ;  
REINDEX
```

2.12 Modulo 12

2.12.1 Ejercicio 1

- Estas trabajando con varios esquemas dentro de una base de datos. Necesitas conocer cuales son los esquemas que se encuentran en el search path actualmente. Escribe una consulta para conseguir esta información.

2.12.2 Solución 1

Estas trabajando con varios esquemas dentro de una base de datos. Necesitas conocer cuales son los esquemas que se encuentran en el search path actualmente. Escribe una consulta para conseguir esta información.

```
hoplastore=> select current_schemas(true);  
    current_schemas  
-----  
{pg_catalog,hoplastore,public}  
(1 row)  
  
hoplastore=> show search_path ;  
    search_path  
-----  
"$user",public  
(1 row)
```

2.12.3 Ejercicio 2

- Se requiere determinar los nombres y definiciones de todas las vistas que se encuentran en el esquema *hoplastore*. Crea un informe que devuelva esta información.
- Después de hacer cambios en el fichero *postgresql.conf* debe hacerse un reload de la configuración. Empleando un comando desde *psql* recárgala.

2.12.4 Solución 2

Se requiere determinar los nombres y definiciones de todas las vistas que se encuentran en el esquema *hoplastore*. Crea un informe que devuelva esta información.

```
hoplastore=> select * from pg_views where schemaname='hoplastore';
(No rows)
```

Después de hacer cambios en el fichero *postgresql.conf* debe hacerse un reload de la configuración. Empleando un comando desde *psql* recárgala.

```
hoplastore=> \c hoplastore postgres
You are now connected to database "hoplastore" as user "postgres".
hoplastore=# select pg_reload_conf();
-[ RECORD 1 ]---+
pg_reload_conf | t
```

2.12.5 Ejercicio 3

- Crea un informe de todos los usuarios que están actualmente conectados, lo que están haciendo y desde cuánto están conectados.
- Elimina uno de ellos empleando una función de PostgreSQL.

2.12.6 Solución 3

Crea un informe de todos los usuarios que están actualmente conectados, lo que están haciendo y desde cuánto están conectados.

```
hoplastore=# select datname, pid, usename, backend_start, xact_start, state, query
from pg_stat_activity;
   datname  | pid | usename |      backend_start      |      xact_start      |
   state    |          |           |           |           |
-----+-----+-----+-----+-----+-----+
+-----+
```

Prácticas Introductorio PostgreSQL

```
postgres | 10681 | postgres | 2016-02-14 12:33:35.434686+01 |
| idle   |
hoplastore | 10654 | postgres | 2016-02-14 12:31:43.806095+01 | 2016-02-14
12:36:09.776099+01 | active | select datname, pid, username, backend_start,
xact_start, state, query from pg_stat_activity;
(2 rows)
```

Elimina uno de ellos empleando una función de PostgreSQL.

```
# Se elimina la conexión psql a la BBDD postgres

hoplastore=# select pg_terminate_backend (10681);
pg_terminate_backend
-----
t
(1 row)

# Se comprueba que se ha muerto

hoplastore=# select datname, pid, username, backend_start, xact_start, state, query
from pg_stat_activity;
  datname  | pid | username |      backend_start      |      xact_start      | state
           |     |          |-----+-----+-----+-----+
+-----+
hoplastore | 10654 | postgres | 2016-02-14 12:31:43.806095+01 | 2016-02-14
12:37:08.2259+01 | active | select datname, pid, username, backend_start, xact_start,
state, query from pg_stat_activity;
(1 row)
```

2.12.7 Ejercicio 4

- Escribe una query que muestre el tamaño de la base de datos *hoplastore*. El tamaño debe ser mostrado en una unidad legible.

2.12.8 Solución 4

Escribe una query que muestre el tamaño de la base de datos *hoplastore*. El tamaño debe ser mostrado en una unidad legible.

```
hoplastore=# select pg_size_pretty(pg_database_size('hoplastore'));
pg_size_pretty
-----
22 MB
(1 row)
```

2.13 Modulo 13

2.13.1 Ejercicio 1

- Descarga los datos desde la tabla *emp* a un fichero csv, con las cabeceras de las columnas y delimitados por un pipe (|).
- Crea una table *copyemp* con la misma estructura que *emp*.
- Carga los datos en *copyemp* desde el fichero csv del paso uno.

2.13.2 Solución 1

Descarga los datos desde la tabla *emp* a un fichero csv, con las cabeceras de las columnas y delimitados por un pipe (|).

```
hoplastore=# COPY hoplastore.emp to '/tmp/emp.csv' with csv header;  
COPY 13
```

Crea una table *copyemp* con la misma estructura que *emp*.

```
hoplastore=# CREATE TABLE copyemp (like hoplastore.emp);  
CREATE TABLE
```

Carga los datos en *copyemp* desde el fichero csv del paso uno.

```
hoplastore=# copy copyemp from '/tmp/emp.csv' with csv header;  
COPY 13
```

```
hoplastore=# select count(*) from copyemp ;  
count  
-----  
 13  
(1 row)
```