

Laboratorio de Bases de Datos I

SESION 12

Guía para crear Triggers y gestionar roles de usuarios

Sergio Caro Álvaro

Ciencias de la Computación

Curso Académico 2017/2018

Índice

- Triggers.
 - Introducción
 - Creación de triggers con PGAdmin III
- Privilegios en PostgreSQL
 - Creando roles de grupo
 - Creando roles de usuario
 - Definiendo privilegios
 - Accediendo con usuarios
- Referencias

Índice

TRIGGERS
(1) Introducción

Universidad
de Alcalá

TRIGGERS
(2) Creación de Triggers en
PGAdmin III

Universidad
de Alcalá

Privilegios en PostgreSQL
Determinando privilegios a grupos y usuarios con PGAdmin III

Universidad
de Alcalá

Privilegios en PostgreSQL
(1) Creando roles de Grupo

Universidad
de Alcalá

Privilegios en PostgreSQL
(2) Creando roles de Usuario

Universidad
de Alcalá

Privilegios en PostgreSQL
(3) Definiendo privilegios

Universidad
de Alcalá

Privilegios en PostgreSQL
(4) Accediendo con usuarios

Universidad
de Alcalá

Referencias

Universidad
de Alcalá

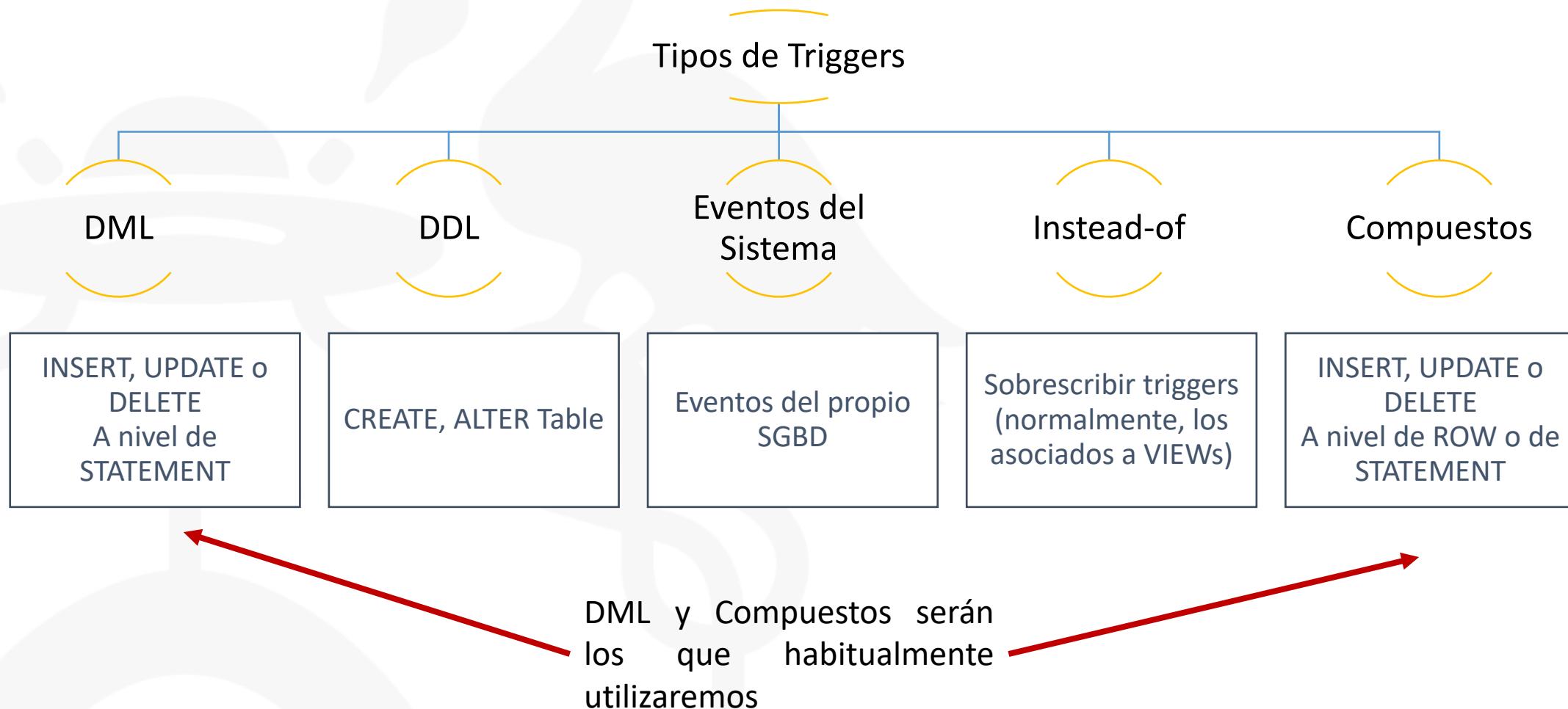
TRIGGERS

(1) Introducción

Triggers

- **Trigger** (disparador): Objetos asociados a tablas, guardados de la Base de Datos, que *se disparan* (es decir, se ejecutan) ante determinados eventos (INSERT, UPDATE, DELETE ...).
- ¿Por qué usarlos?
 - No todo se puede controlar con integridad referencial y/o checks.
 - Única forma para generar los atributos derivados.
 - Control, administración y seguridad.
- **NOTA:** no podemos llamar manualmente al trigger, se ejecutará automáticamente en respuesta a un evento.

Triggers. Tipos



TRIGGERS

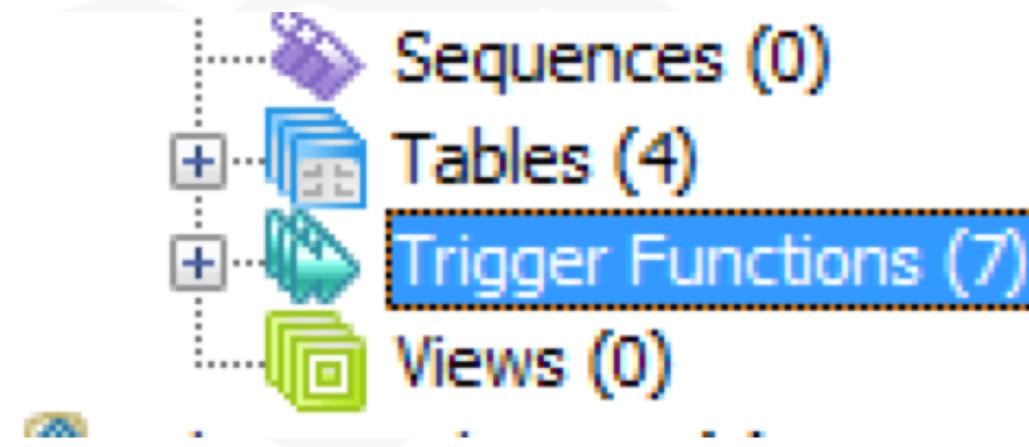
(2) Creación de Triggers en PGAdmin III

Consideraciones Previas

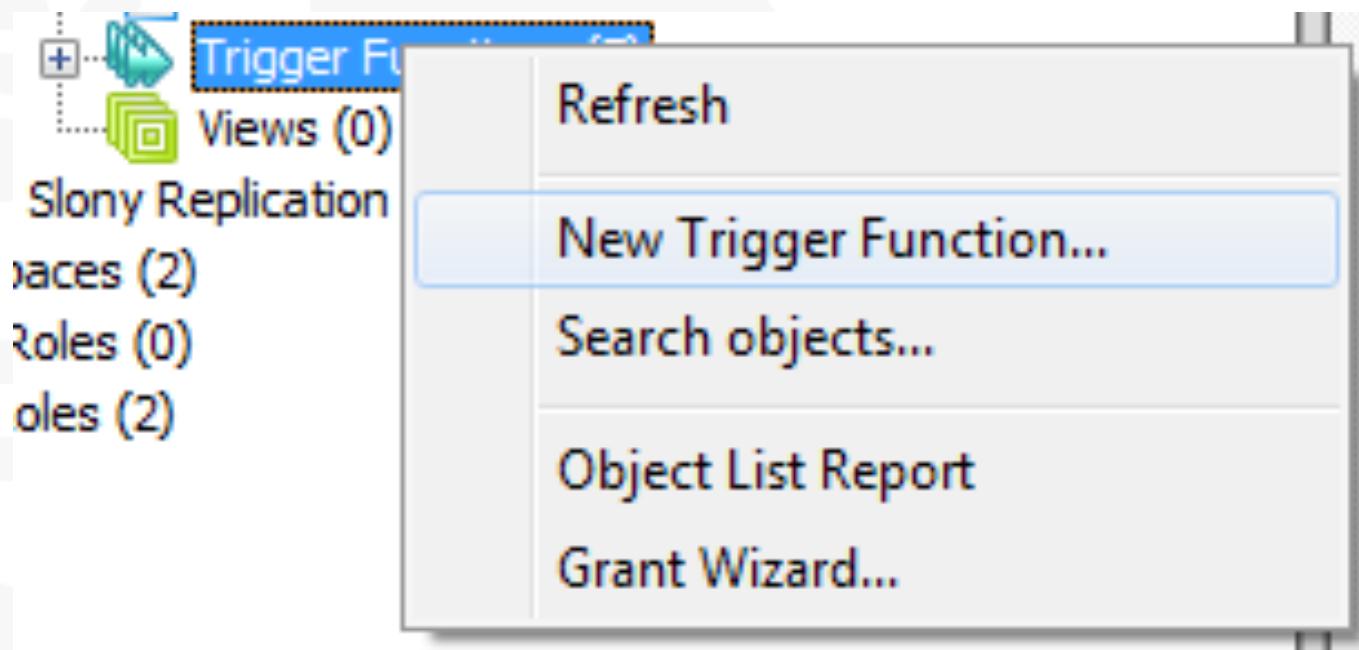
- Los triggers se asocian a una determinada BBDD, y solo aplica a esa BBDD.
- **Importante** el concepto de función (o procedimiento almacenado) y trigger:
 - Función: qué hacer cuando se ejecute el trigger.
 - Trigger: determinamos el evento a capturar y la función a ejecutar.

¿Dónde ubicar la función?

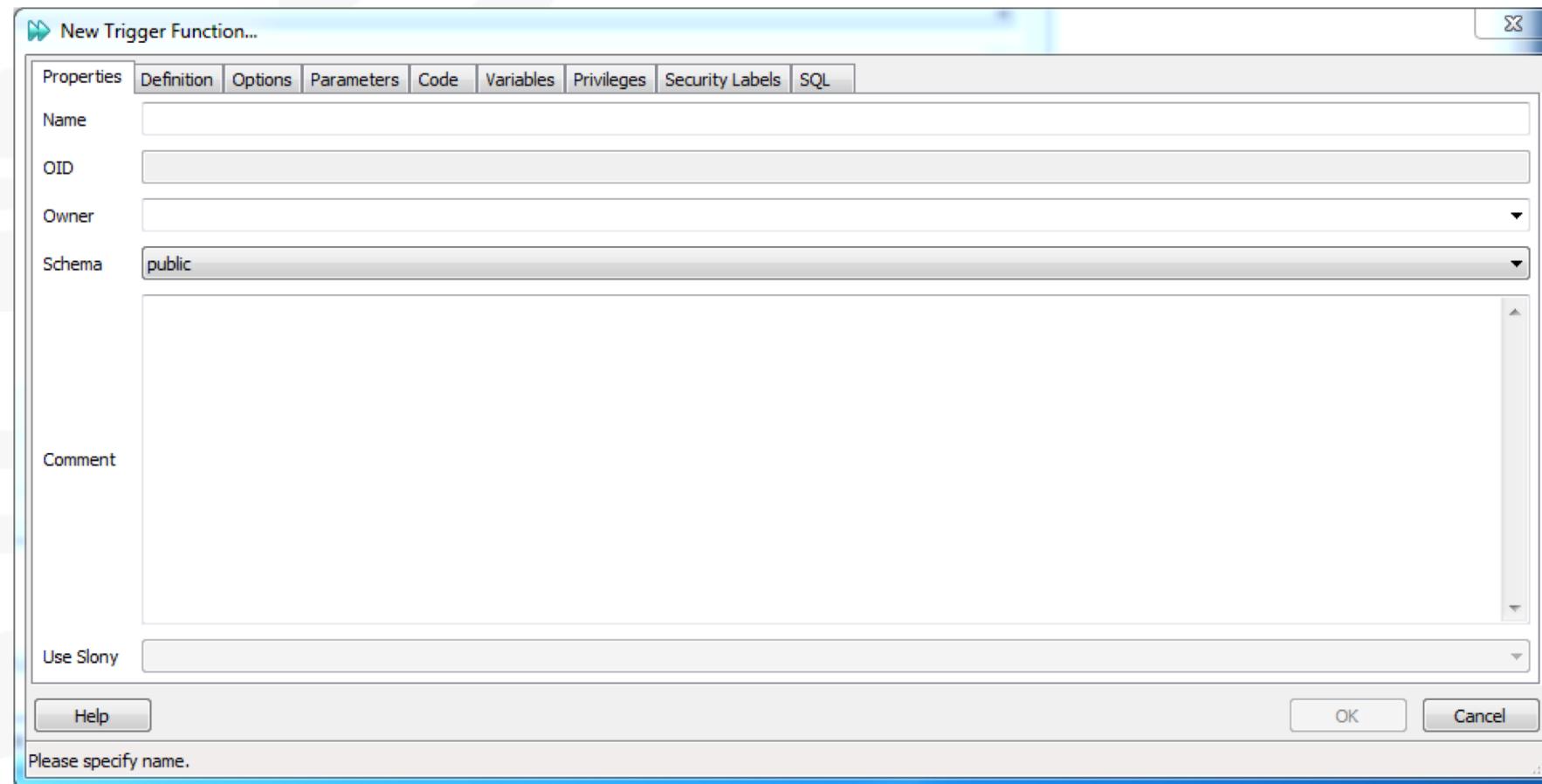
- <database> / Schemas / public / Trigger Functions



Creando la función (i)



Creando la función (ii)



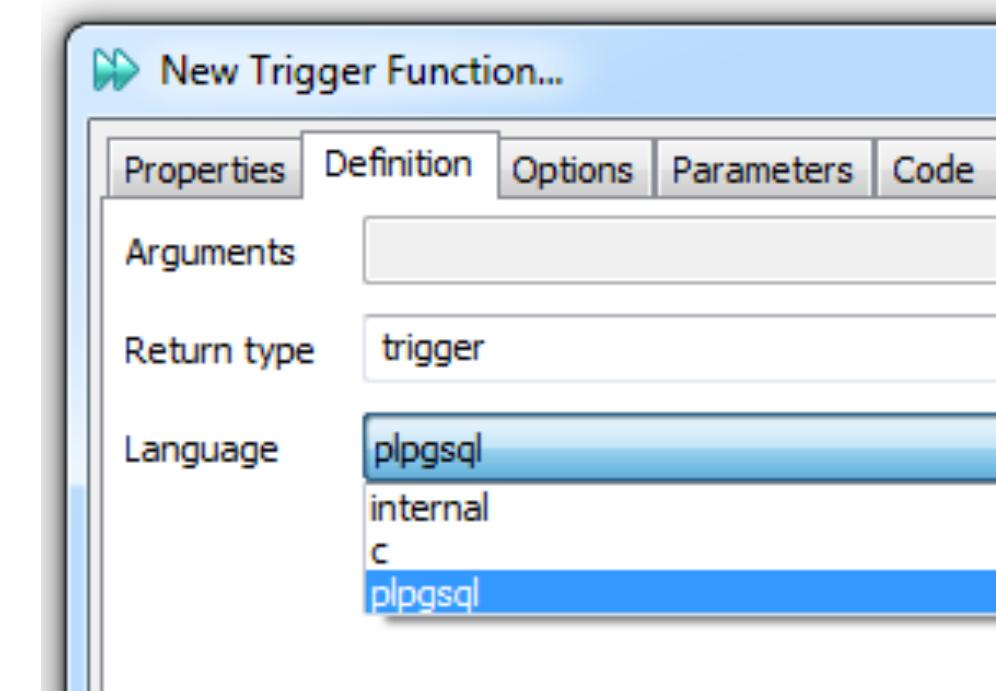
Creando la función (iii)

- Definimos el nombre de la función, el propietario (típicamente postgres) y el esquema (normalmente publico)

	Properties	Definition	Options	Parameters	Code	Variables
Name		nombre_de_la_funcion_asociada_al_trigger				
OID						
Owner		postgres				
Schema		public				

Creando la función (iv)

- El tipo de retorno es “trigger” (normalmente ya viene indicado).
- Aunque se puede escribir la función con varios lenguajes, desde aquí veremos plpgsql (habitual en PostgreSQL).



Creando la función (v)

- En la sección de “Code” escribimos la función. Recurrir a las referencias y a la bibliografía de BlackBoard para saber más sobre como escribir triggers con plpgsql.

New Trigger Function...

Properties Definition Options Parameters Code Variables

```
1 DECLARE          -- Declaracion de variables
2
3 BEGIN            --Cuerpo de la función
4
5
6          --Retorno
7      RETURN [ NULL | NEW ]
8 END;
```

Creando la función (vi)

- Una vez creada la función, debería aparecer en el listado de funciones asociadas a la base de datos (si no es así, forzar un “Refresh” de la lista).



nombre_de_la_funcion_asociada_al_trigger()

Asociando el Trigger



- Una vez creada la función, crearemos el trigger asociado a las condiciones del evento de disparo (CTR+E o símbolo de lupa), al que le asociaremos la función creada previamente. Recurrir a las referencias y a la bibliografía de BlackBoard para saber más sobre como escribir triggers.

A screenshot of an SQL editor window. The title bar says "SQL Editor". Below it is a toolbar with a magnifying glass icon. The main area has tabs for "SQL Editor" and "Graphical Query Builder", with "SQL Editor" selected. A "Previous queries" button is also present. The code area contains the following SQL script:

```
CREATE TRIGGER nombre_del_trigger BEFORE INSERT OR UPDATE OR DELETE
ON tabla FOR EACH ROW
EXECUTE PROCEDURE nombre_de_la_funcion_asociada_al_trigger();
```



Privilegios en PostgreSQL

Determinando privilegios a grupos y usuarios con PGAdmin III

Permisos. Introducción

- En SQL, los privilegios (o permisos) se pueden otorgar (y revocar) sobre tablas, atributos, vistas, funciones y secuencias.
- Via SQL, se realiza mediante *GRANT* y *REVOKE*.
- Hay que tener cuidado con el tema de permisos, y solo deberían poder hacerlo administradores del sistema y los dueños de los objetos.

```
GRANT privilege_name  
ON object_name  
TO {user_name | PUBLIC | role_name}  
[WITH GRANT OPTION];
```

```
REVOKE privilege_name  
ON object_name  
FROM {user_name | PUBLIC | role_name}
```

Permisos. Introducción.

- WITH GRANT OPTION permite que al rol de grupo/usuario al que se le han concedido unos permisos pueda, a su vez, concederlos a otros roles de grupo/usuario.

Permisos. Introducción.

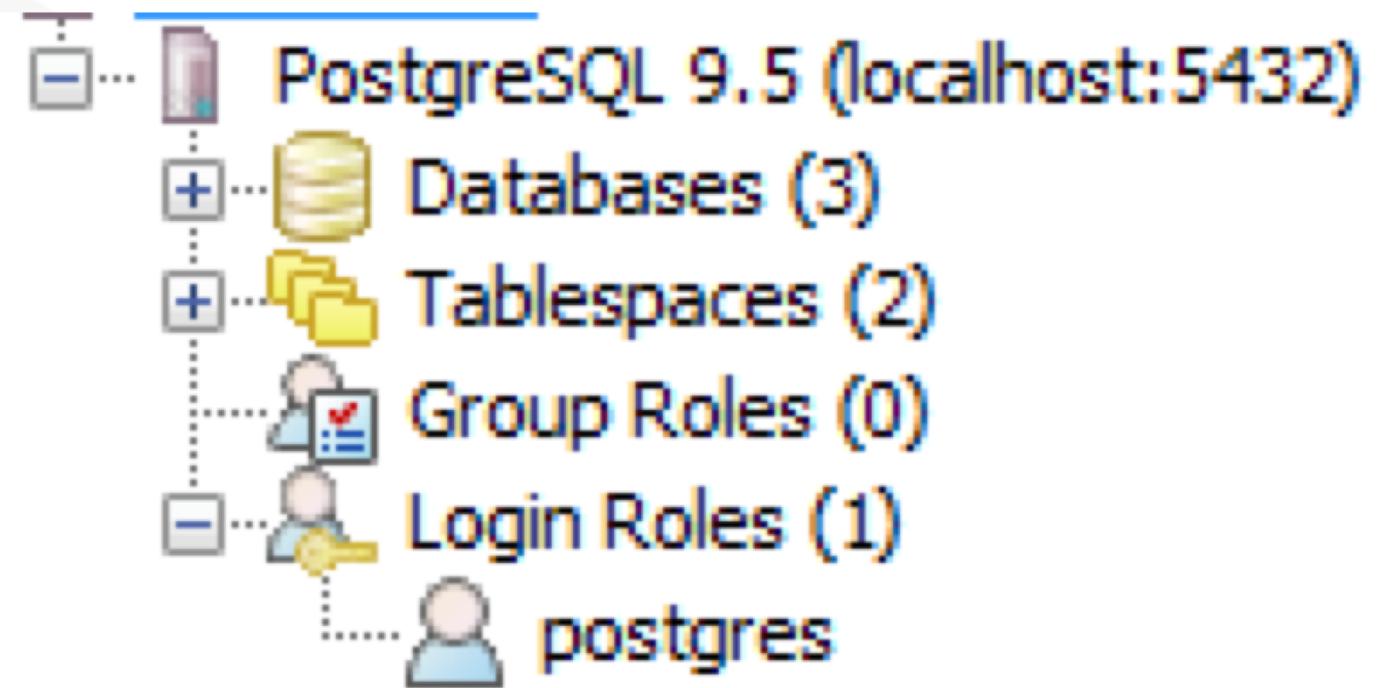
- De cara a la mantenibilidad, y a facilitar la operación del SGBD, es recomendable la creación de roles de grupo (a los que se asocian usuarios concretos) sobre los que se aplican las correspondientes restricciones de acceso.

Privilegios en PostgreSQL

(1) Creando roles de Grupo

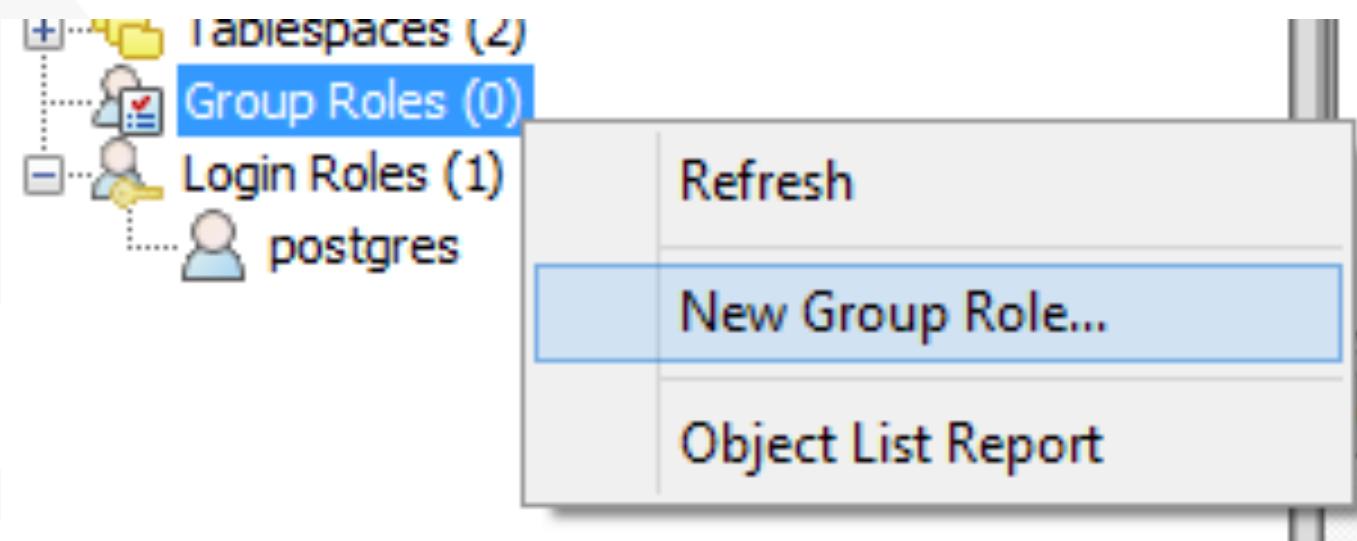
Creando roles de Grupo

- Los roles de grupo se guardan en la sección “*Group Roles*” y son propios para cada servidor.



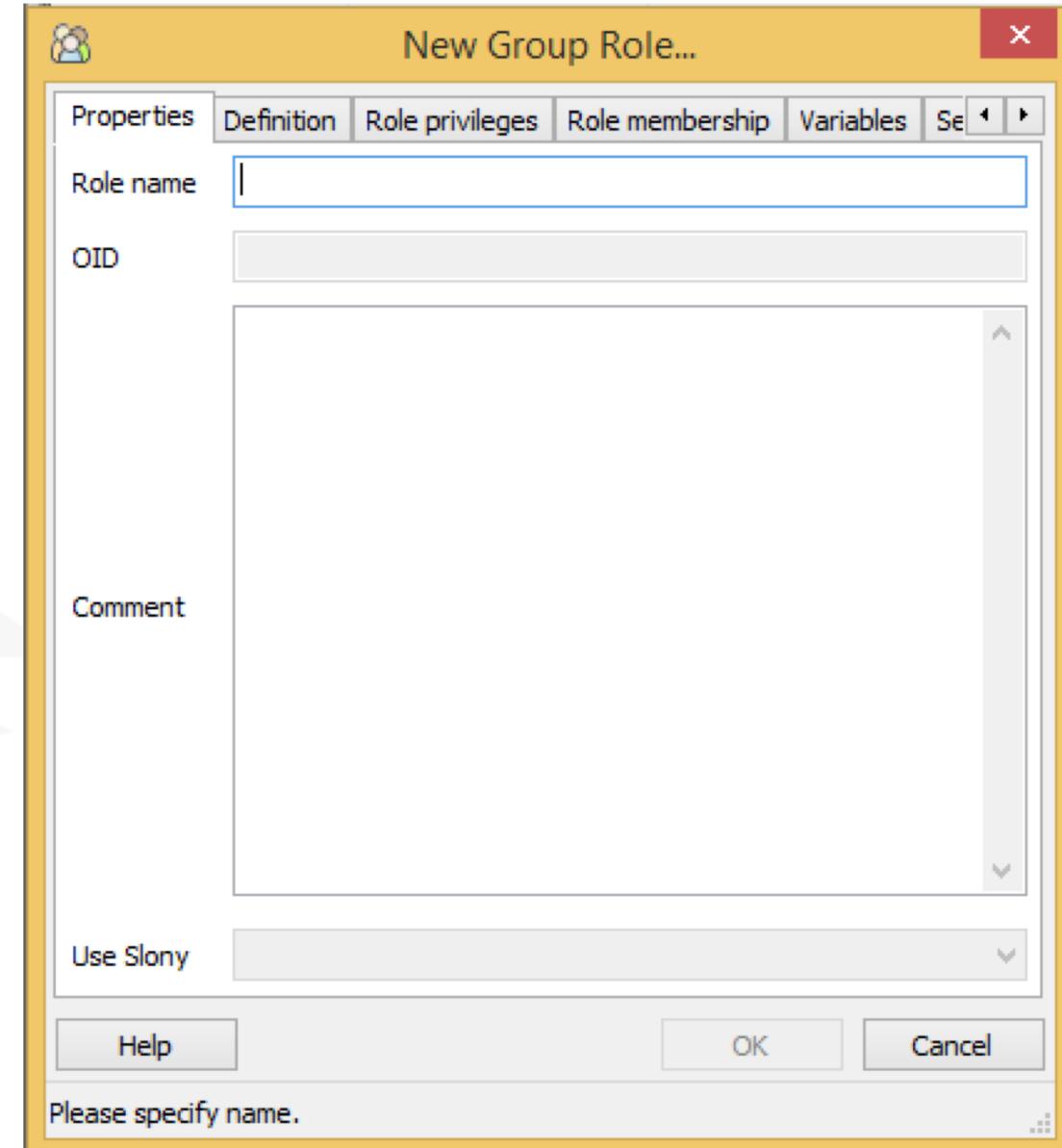
Creando roles de Grupo

- Para crear un nuevo rol de grupo, botón derecho en la sección “*Group Roles*” y seleccionar “*New Group Role...*”



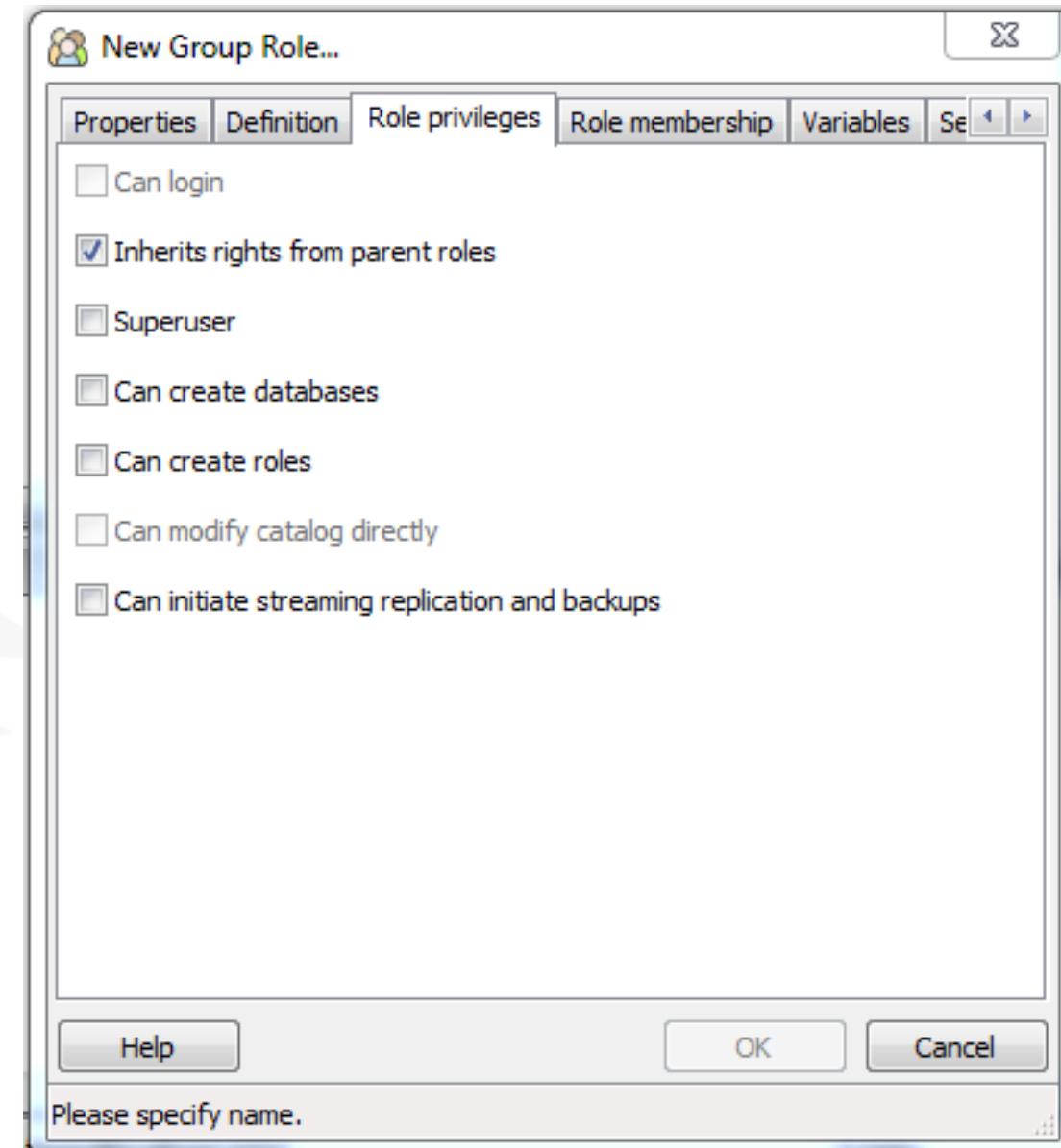
Creando roles de Grupo

- Con indicar el nombre del rol (en la pestaña “Properties”) es suficiente



Creando roles de Grupo

- De forma adicional, en la pestaña “*Role privileges*”, se pueden seleccionar una serie de privilegios de administración del servidor.

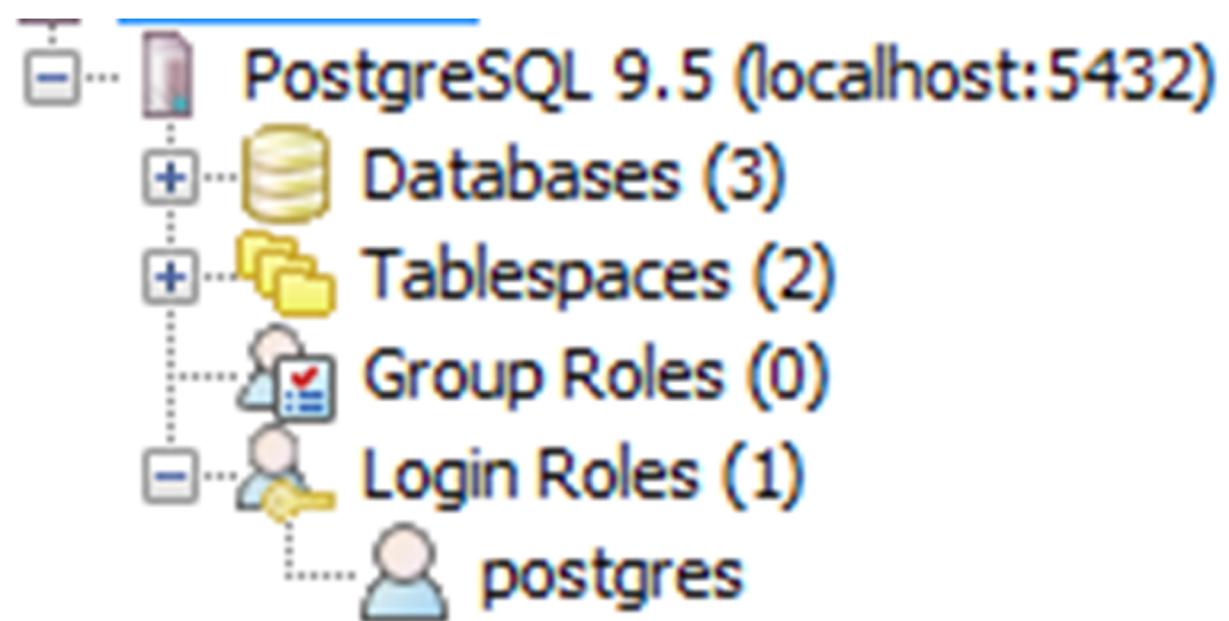


Privilegios en PostgreSQL

(2) Creando roles de Usuario

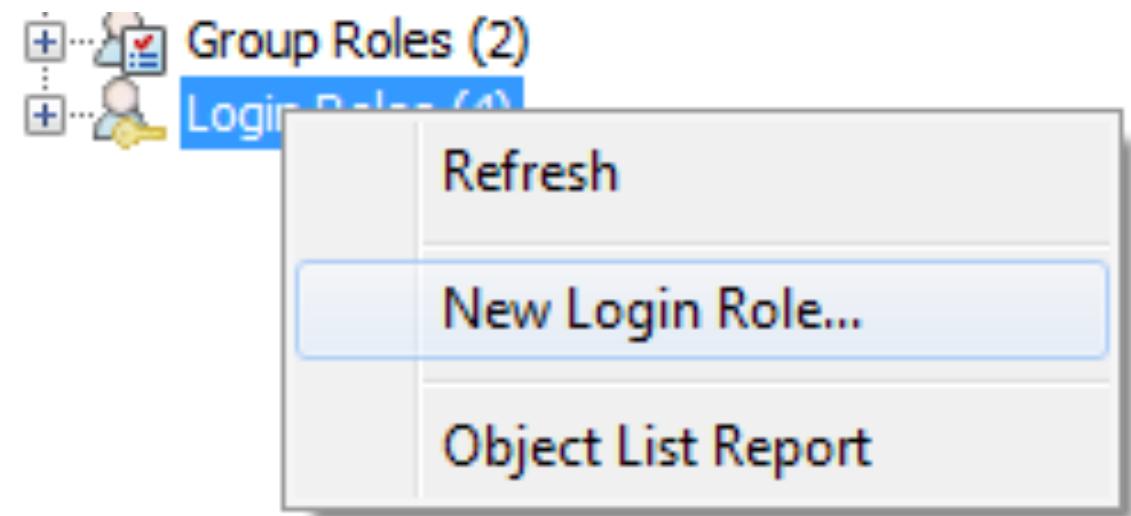
Creando roles de Usuario

- Los roles de usuario se guardan en la sección “*Login Roles*”.
- Al igual que con los roles de grupo, los roles de usuarios son específicos del servidor.



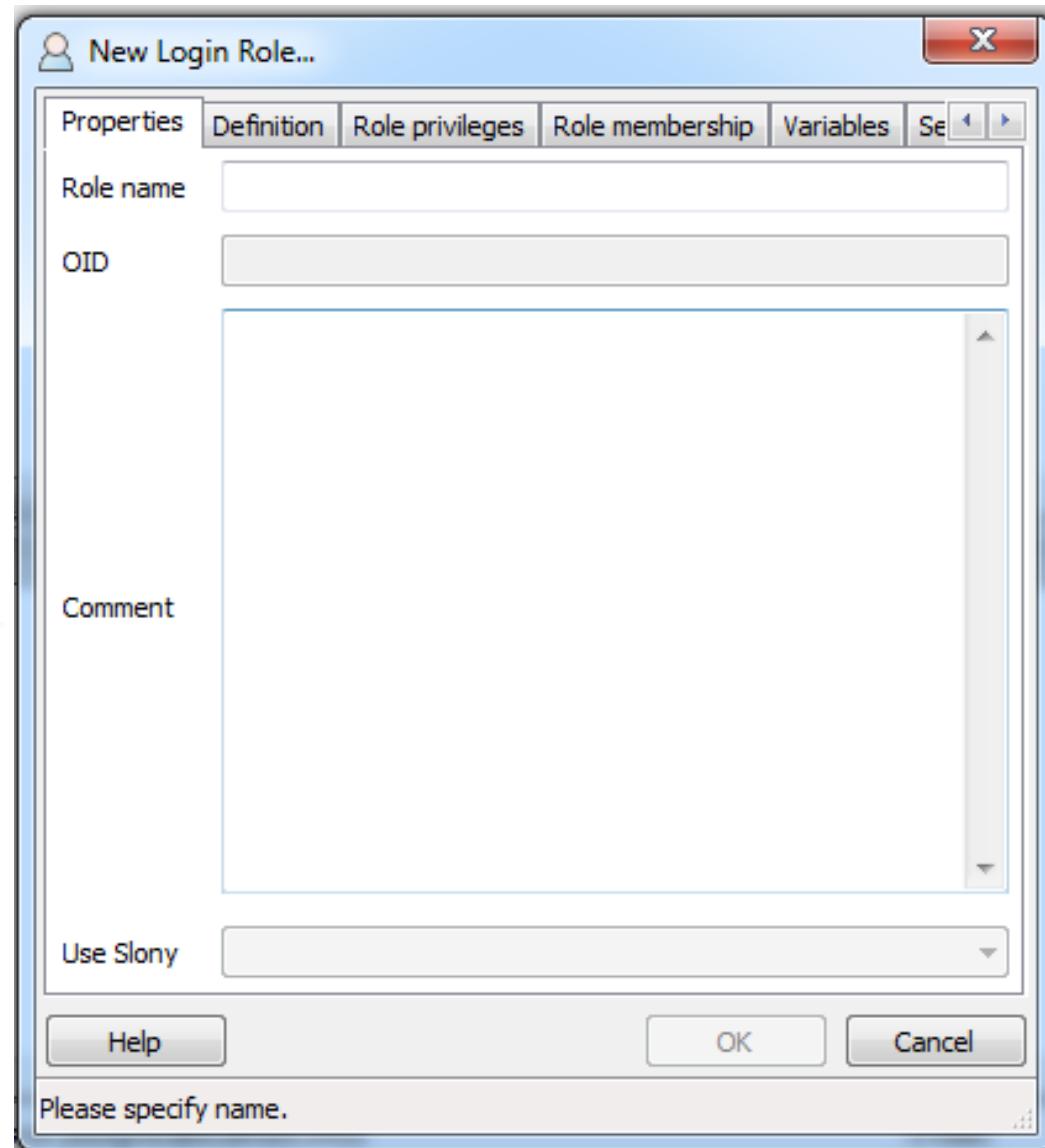
Creando roles de Usuario

- Para crear un nuevo rol de usuario, botón derecho en la sección “*Login Roles*” y seleccionar “*New Login Role...*”.



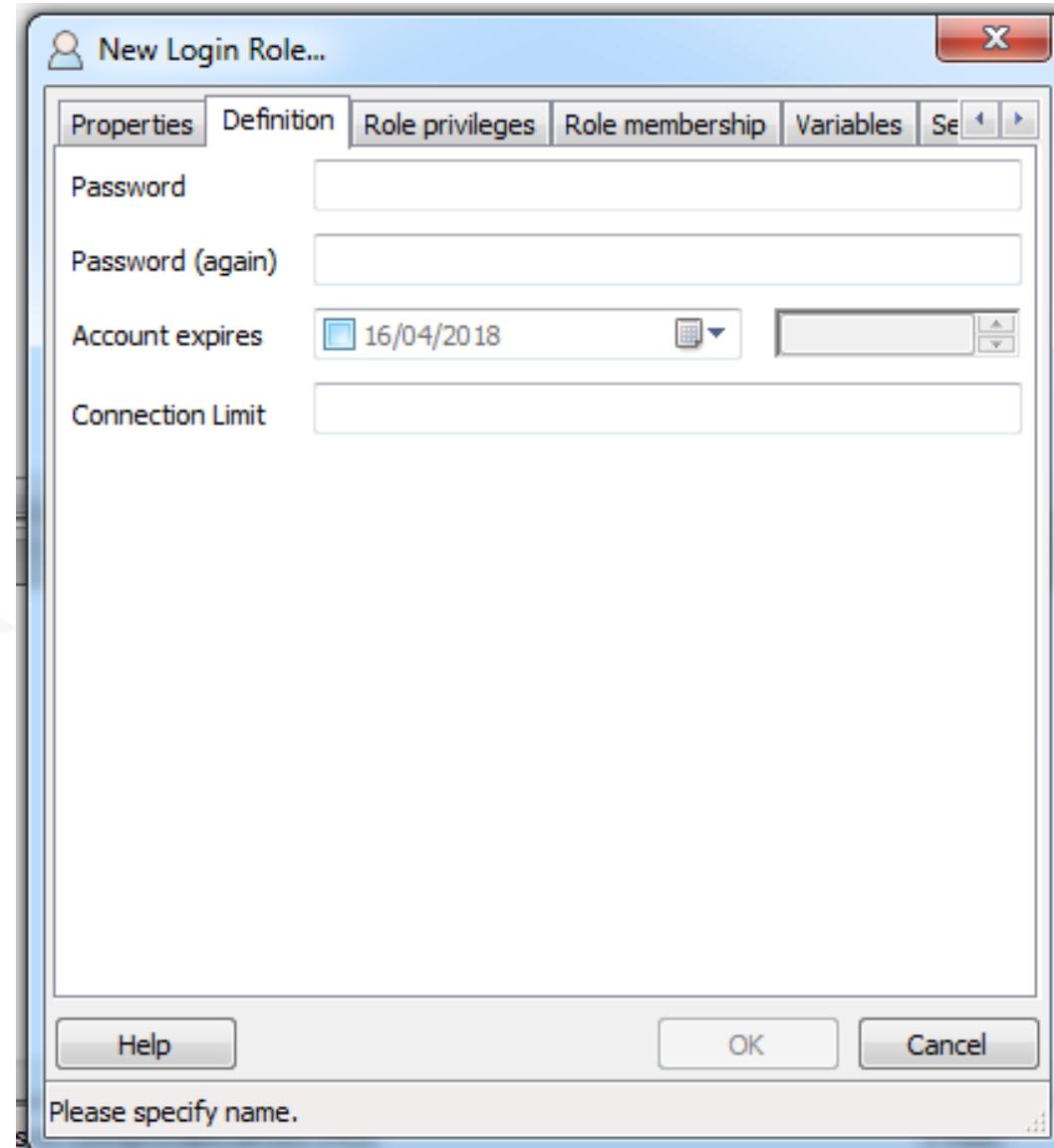
Creando roles de Usuario

- En la pestaña “Properties”, es suficiente con indicar el nombre del rol (“Role name”).
- “Role name” es el nombre de usuario que se usará para el log-in al servidor.



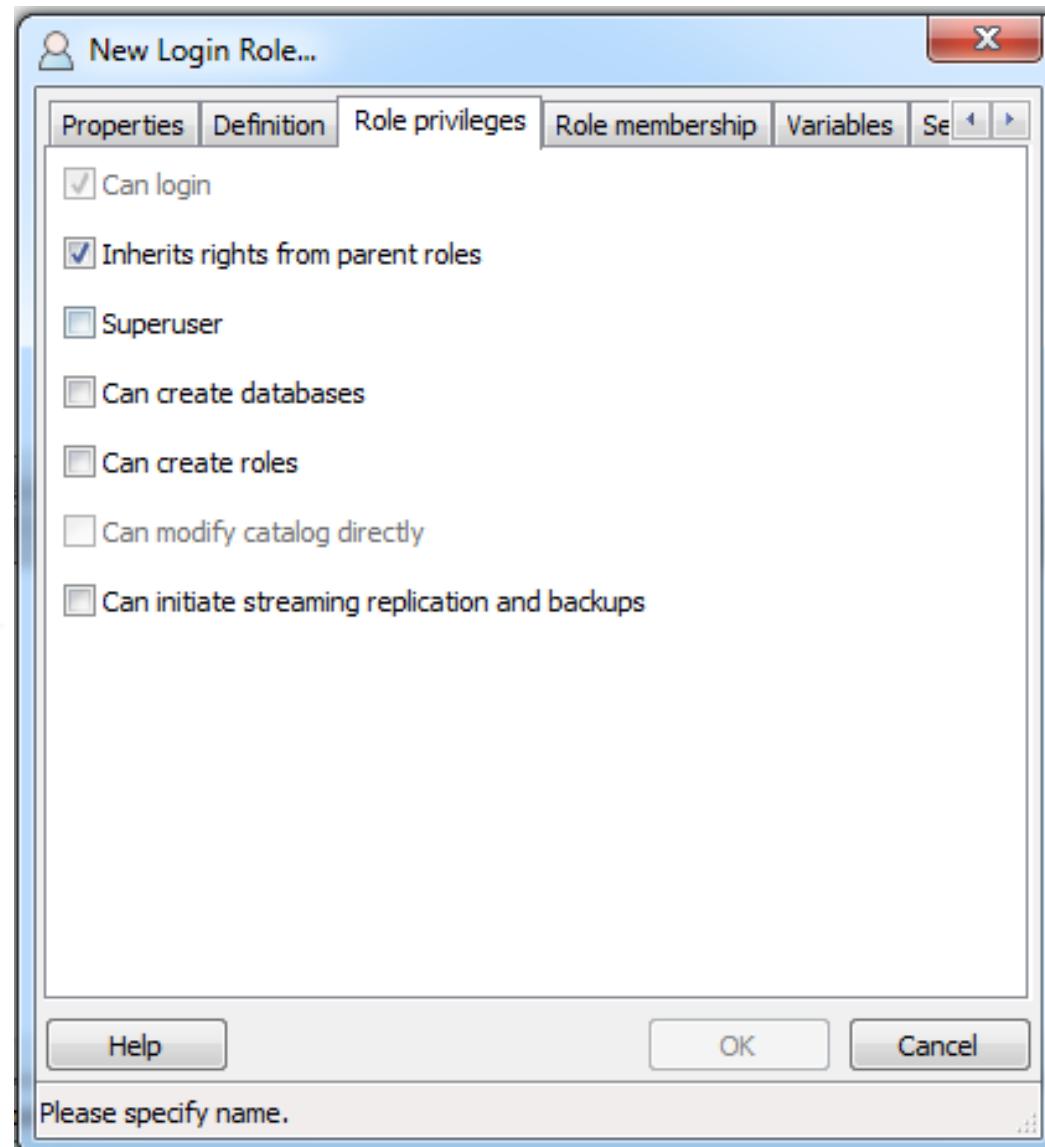
Creando roles de Usuario

- En la pestaña “*Definition*”, se puede especificar la contraseña de acceso para ese usuario (altamente recomendable).
- De forma opcional, podemos indicar cuándo caducará la cuenta.
- También se puede indicar el número de conexiones simultaneas permitidas (“-1” o nada para indicar que no hay límite).



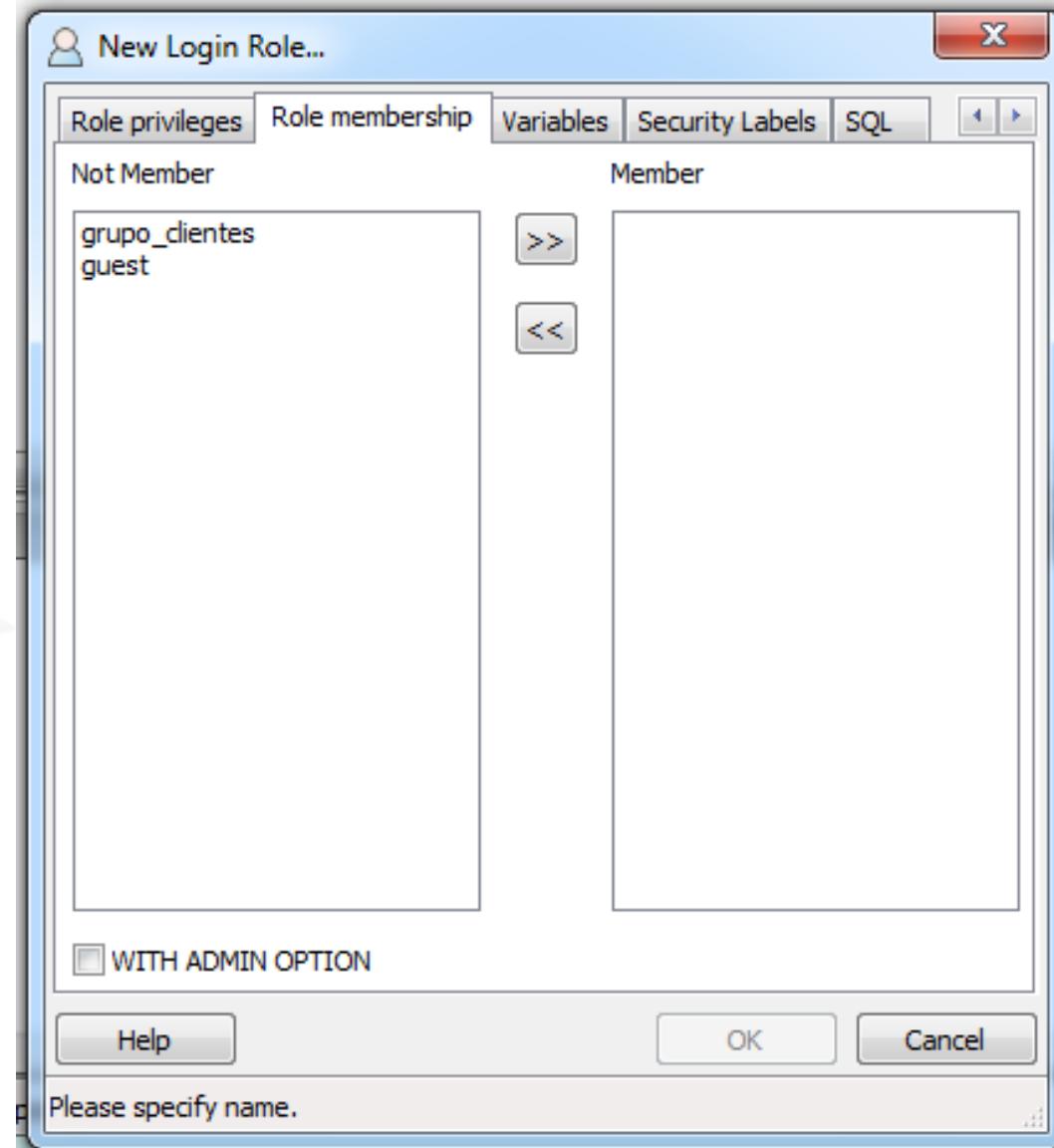
Creando roles de Usuario

- Al igual que con el rol de grupo, en la pestaña “Role privileges”, se pueden seleccionar una serie de privilegios de administración del servidor.
- En este caso, de forma específica al usuario (no afecta al rol de grupo).



Creando roles de Usuario

- Al crear un usuario, lo importante es la sección “*Role membership*”, para indicar a cual(es) grupo(s) pertenece el usuario.



Privilegios en PostgreSQL

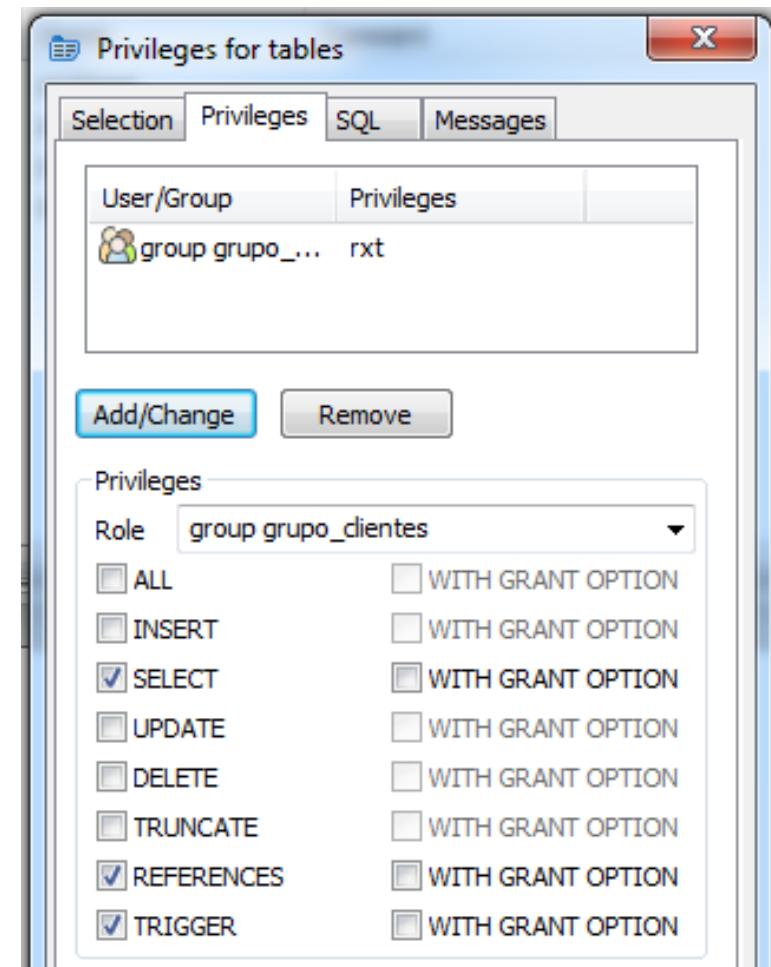
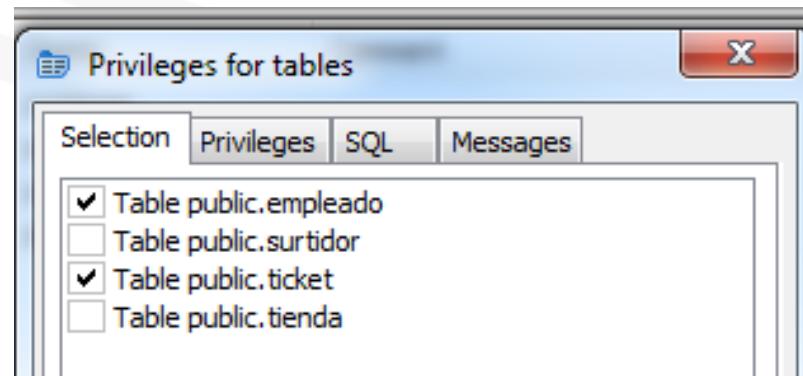
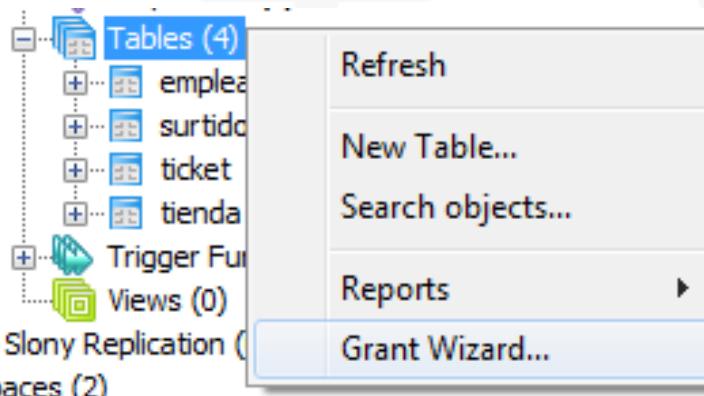
(3) Definiendo privilegios

Definiendo privilegios

- En términos generales, se pueden asignar permisos a todos los elementos de la base de datos.
- Principalmente, botón derecho en cualquier elemento (base de datos, tabla, atributo) y acceder a las propiedades. En las pestañas “*Privileges*” y/o “*Default Privileges*” se pueden realizar la asignación de permisos.

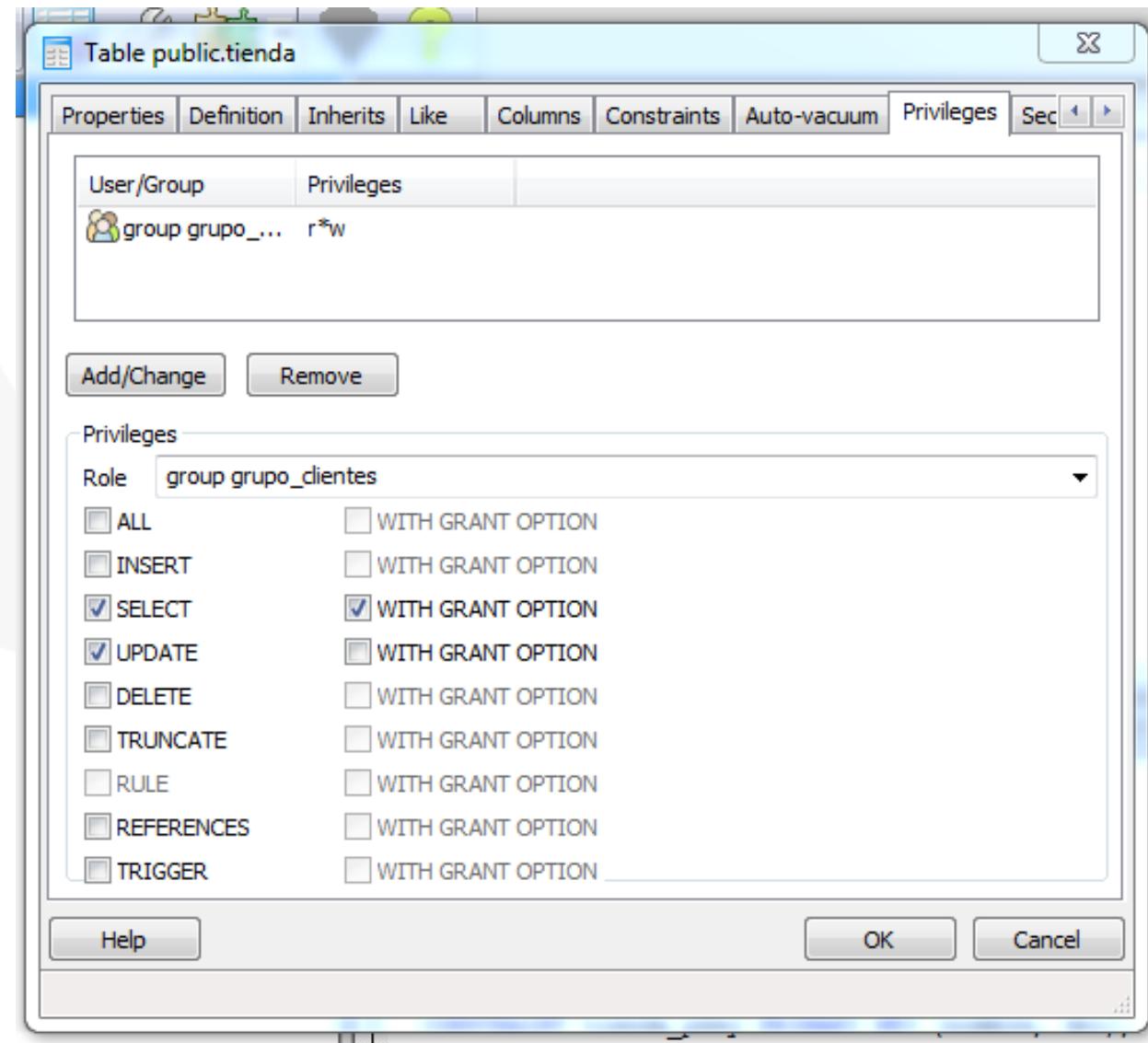
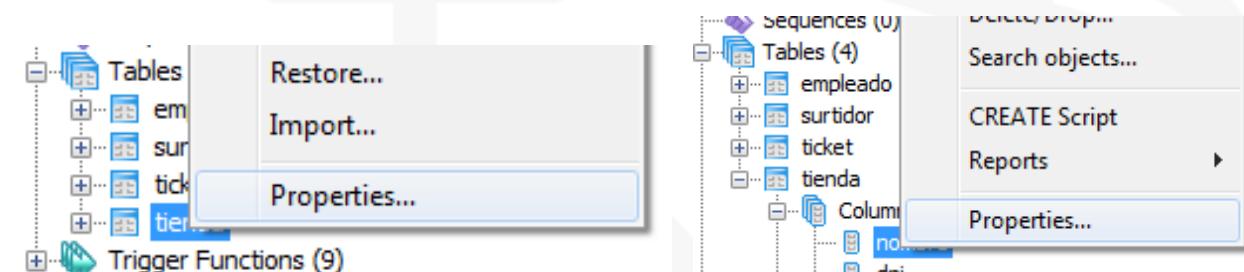
Definiendo privilegios

- Si queremos hacer algo rápido y común a todas las tablas, botón derecho en “Tables” y seleccionar “Grant Wizard...”. Seleccionamos las tablas que queremos modificar y, en la pestaña “Privileges” seleccionar los permisos deseados para un determinado rol de grupo.
 - Importante, una vez seleccionados los permisos, dar al botón “Add/Change” para que se guarden los permisos.



Definiendo privilegios

- La otra opción, si queremos algo más específico, es ir elemento a elemento (tabla o columna). Botón derecho y seleccionar “Properties”. En la pestaña “Privileges” seleccionar los permisos deseados para un determinado rol de grupo.
 - Importante, una vez seleccionados los permisos, dar al botón “Add/Change” para que se guarden los permisos.

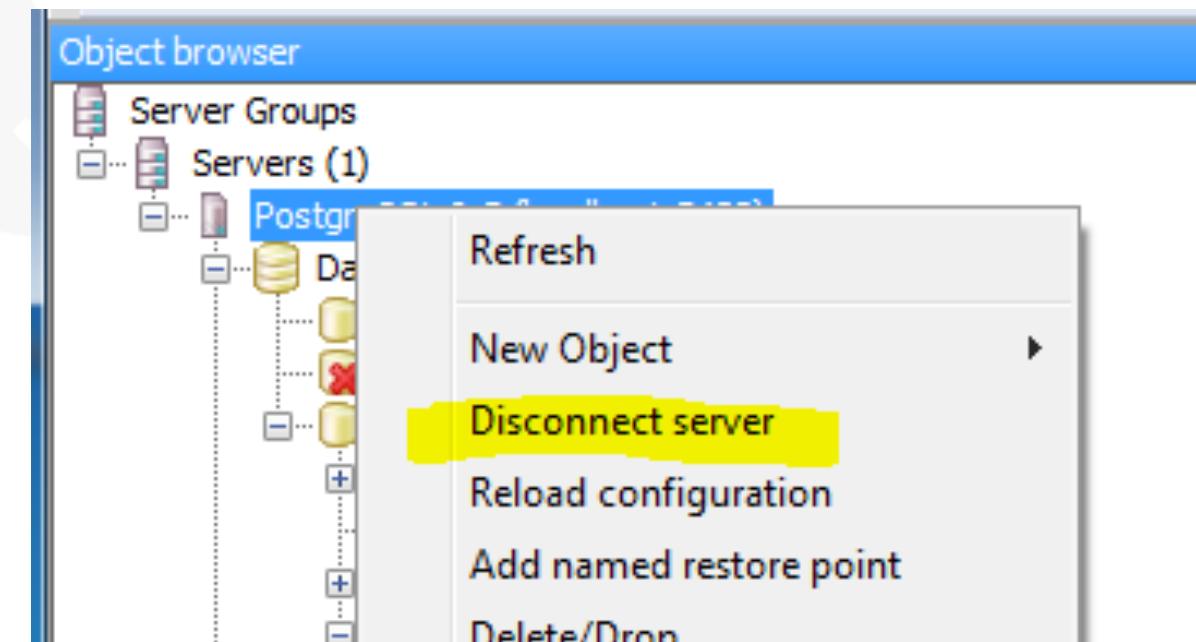


Privilegios en PostgreSQL

(4) Accediendo con usuarios

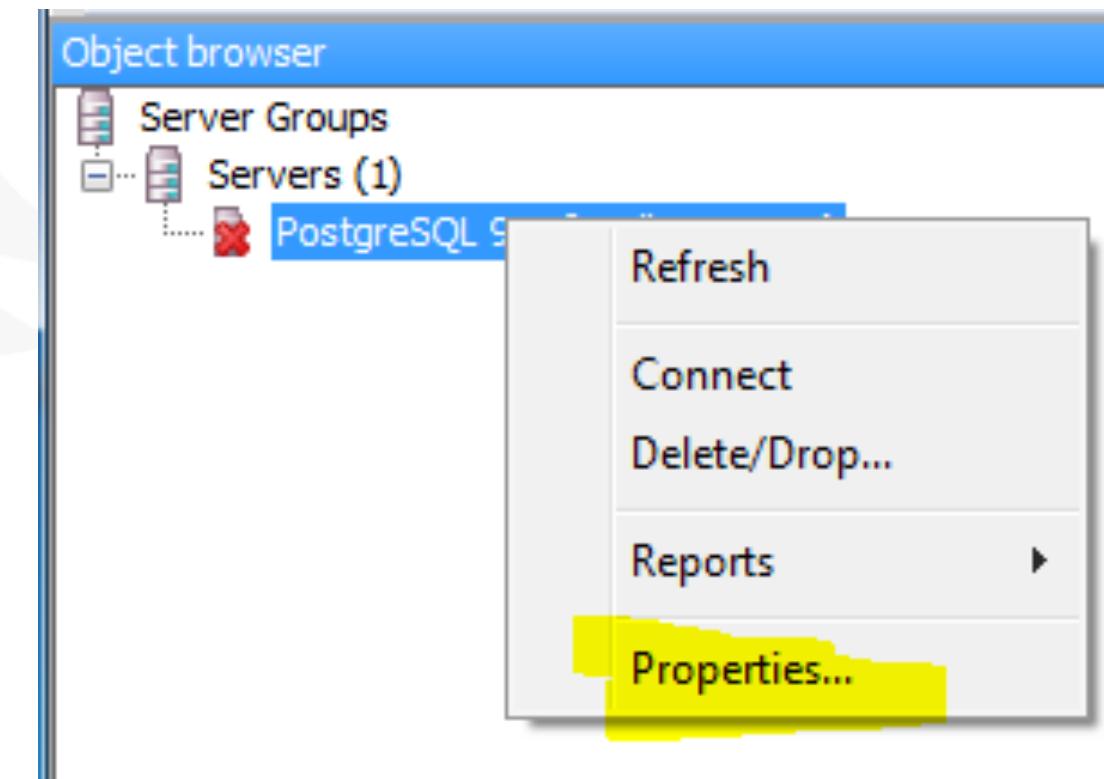
¿Cómo pruebo los usuarios? (i)

- Dada una conexión, nos desconectamos del servidor. Botón derecho en el servidor en el que nos encontramos actualmente y pulsar “*Disconnect Server*”.



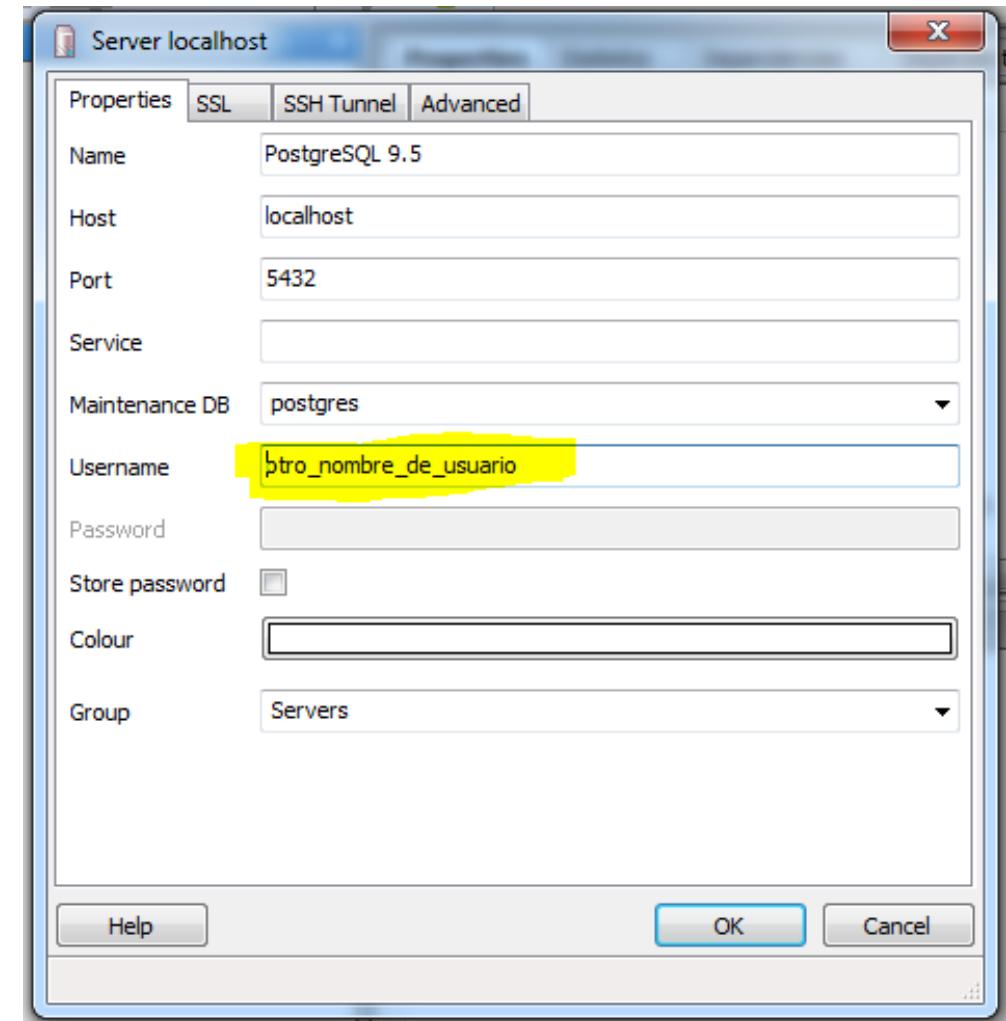
¿Cómo pruebo los usuarios? (ii)

- Tras la desconexión, botón derecho en el servidor y accedemos a las propiedades.



¿Cómo pruebo los usuarios? (iii)

- En la ventana emergente, cambiar el campo “*Username*” con el nombre del rol que hayamos creado previamente y pulsar “OK”.
- Tras guardar, conectar de forma habitual con el Servidor.



Referencias

Referencias

- Tutoriales de Triggers:
 - <https://e-mc2.net/es/disparadores-triggers-en-postgresql>
 - <https://elbauldelprogramador.com/plsql-disparadores-o-triggers/>
 - <http://www.rebellionrider.com/pl-sql-tutorials/triggers-in-oracle-database/introduction-of-triggers-in-oracle-database.htm>
- Permisos
 - <http://beginner-sql-tutorial.com/sql-grant-revoke-privileges-roles.htm>
 - <https://www.postgresql.org/docs/9.0/static/sql-grant.html>