Roles

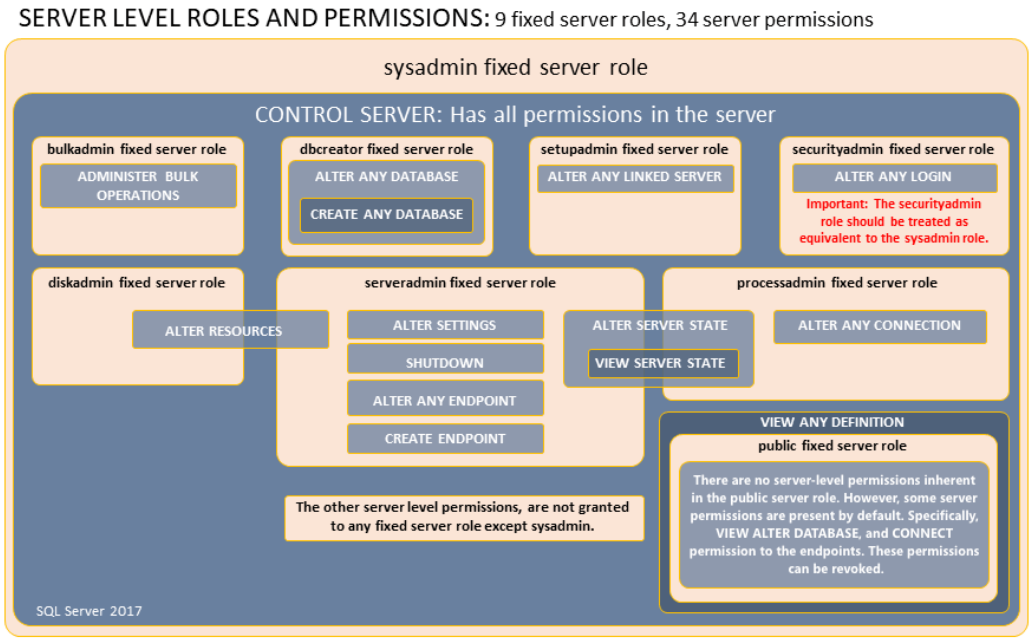
1. ¿Qué es un rol?:

Un rol es una **entidad de seguridad de SQL Server** que nos **permite agrupar permisos**, de manera similar a los grupos del SO Windows.

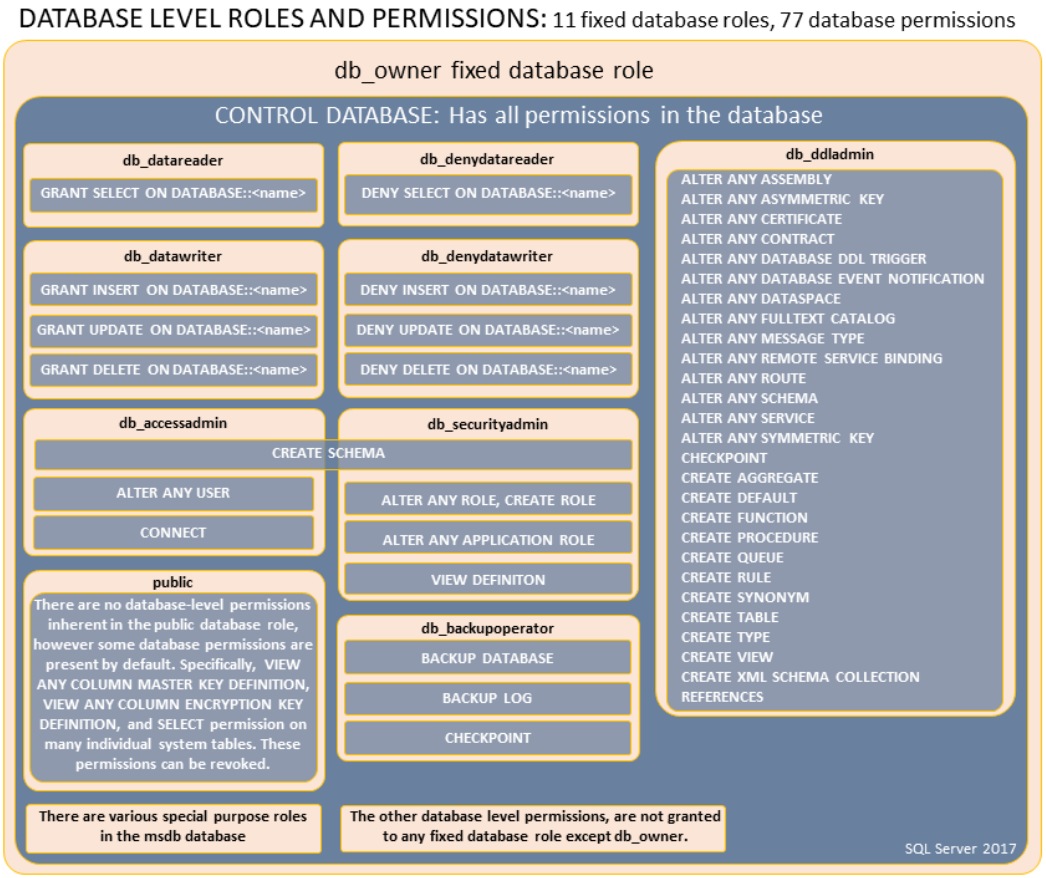
Es muy útil cuando tenemos muchos usuarios y además algunos de ellos comparten permisos, es decir pueden realizar las mismas operaciones en el servidor.

2. Tipos de roles:

- A nivel de **servidor**:
  - Roles **fijos**: vienen *predefinidos* en el servidor (por ejemplo sysadmin, serveradmin, processadmin, …, public). Todos los inicios de sesión pertenecen al rol public.
  - Roles **flexibles**: roles de servidor *creados por el usuario*. Se le agregarían permisos de nivel de servidor.



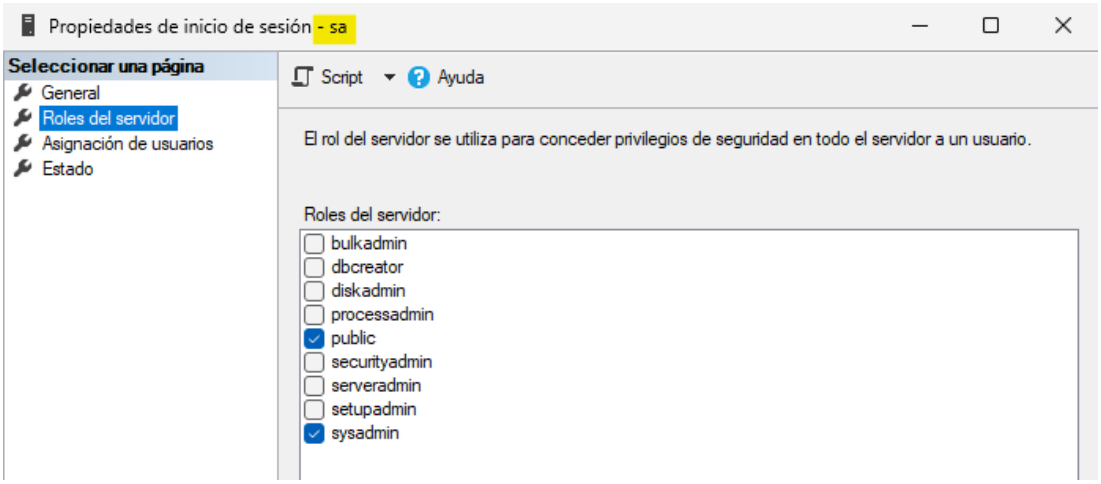
- A nivel de **base de datos**:
  - Roles **fijos**: vienen *predefinidos* en cada base de datos (por ejemplo db_owner, db_ddladmin, db_datawriter, db_datareader,…, public). Todos los usuarios de una BD pertenecen al rol de base de datos public.
  - Roles **flexibles**: roles de bases de datos *creados por el usuario*. Se le agregarían permisos de nivel de base de datos.



3. Comprobamos los roles a los que pertenece el inicio de sesión sa:

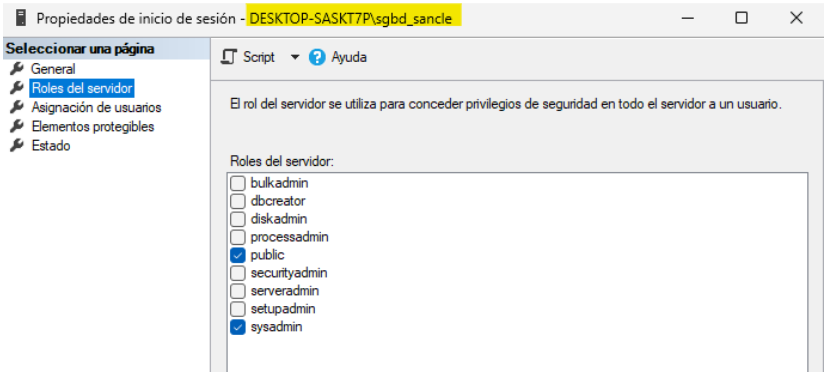
Nos conectamos como sa y consultamos las propiedades del login. En la ventana que aparece escogemos la página **Roles del servidor** y comprobamos que sa pertenece al rol public y aunque intentemos desmárcalo no nos lo permite.

Aunque nos deja desmarcar el rol sysadmin cuando le demos a Aceptar nos dará un error, ya que el usuario sa debe obligatoriamente pertenecer al rol de servidor sysadmin.



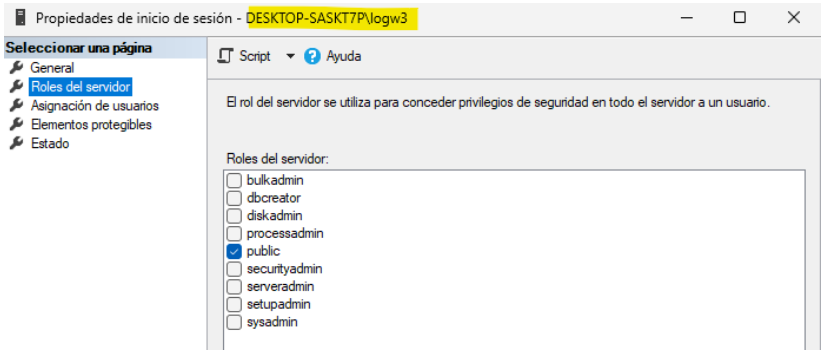
4. Comprobamos los roles a los que pertenece el inicio de sesión de Windows sgbd_sancle:

Nos conectamos como sa y consultamos las propiedades del login. En la ventana que aparece escogemos la página **Roles del servidor** y comprobamos que **sgbd_sancle** pertenece al rol public y también a sysadmin, ya que durante la instalación de SQL Server indicamos que sería administrador del servidor de BD.



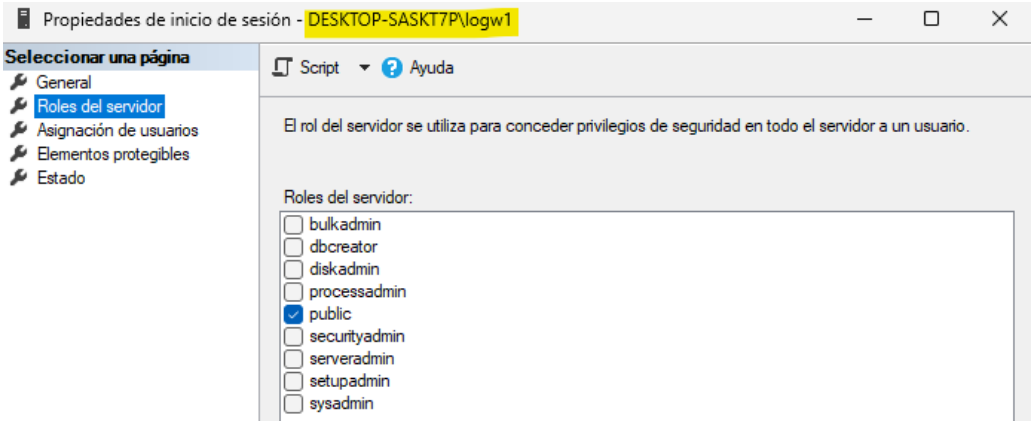
5. Comprobamos los roles a los que pertenece el inicio de sesión de Windows logw3:

Como no es administrador del servidor de BD por defecto sólo pertenece al rol **public**.

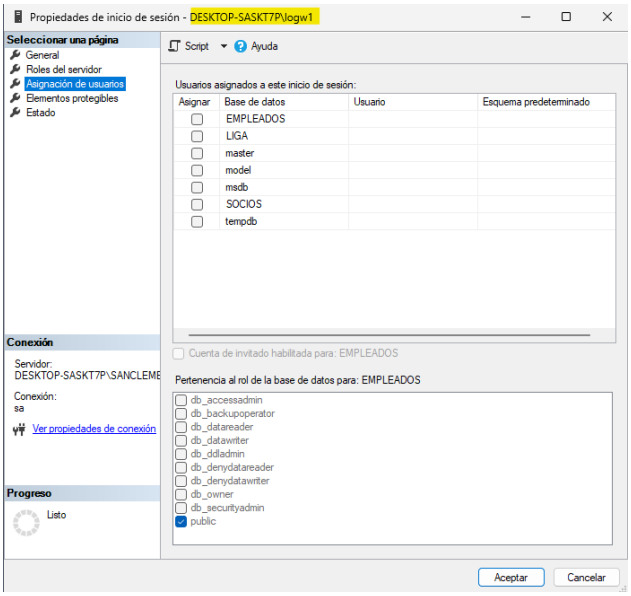


6. Convertir un inicio de sesión en administrador de SQL Server:

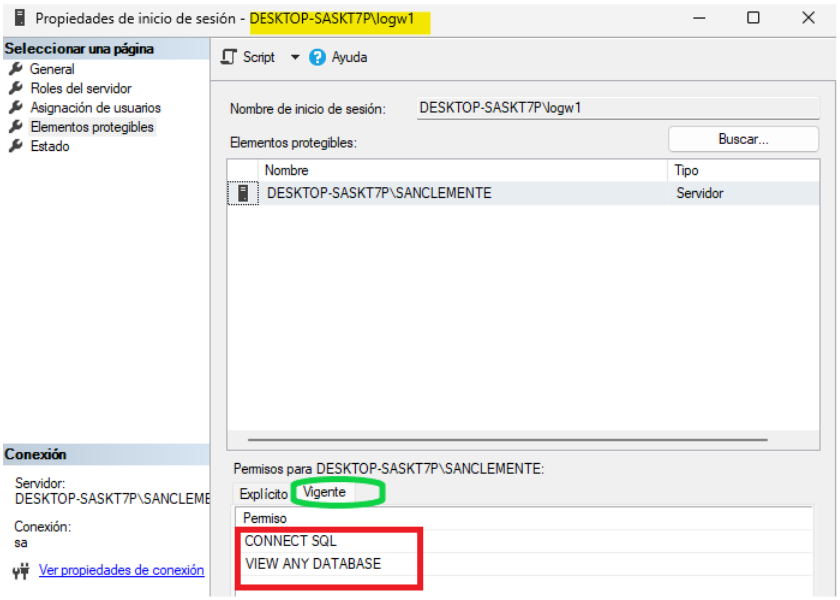
Consultando los permisos del inicio de sesión de Windows logw1 comprobaremos que pertenece al rol de servidor public (se consulta en la página **Roles de servidor**).



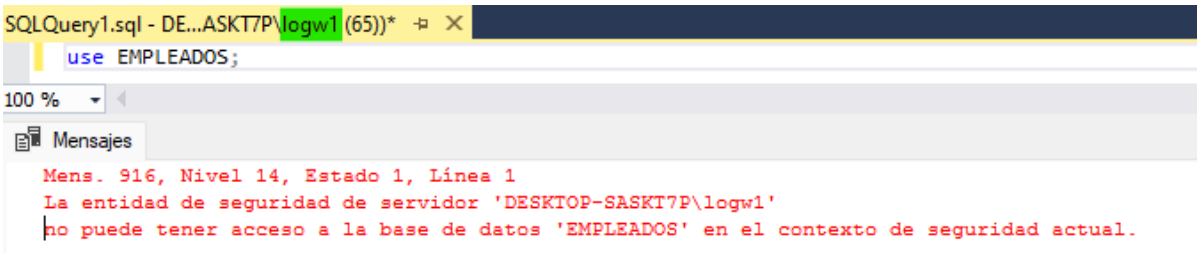
No tiene usuarios asignados en ninguna bd (se consulta en la página **Asignación de usuarios**).



Sus permisos vigentes son de conexión y de poder ver las BD (se consulta en la pestaña *Vigente* de la página **Elementos protegibles**).

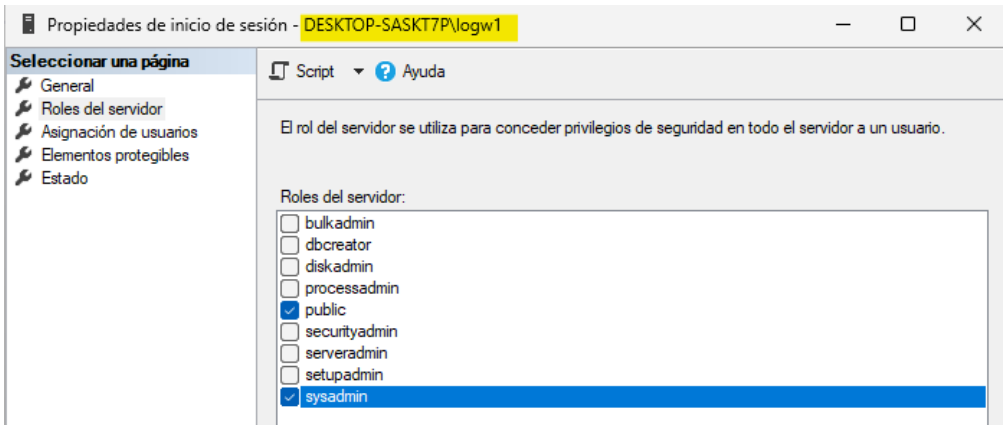


Vamos a comprobar cómo el login logw1 no puede hacer ni un use EMPLEADOS.



Para convertir logw1 en usuario administrador de SQL Server lo asignamos al rol sysadmin.

6.1. Desde entorno gráfico: siendo sa abrimos las propiedades del inicio de sesión y en la página Roles del servidor marcamos ***sysadmin***:



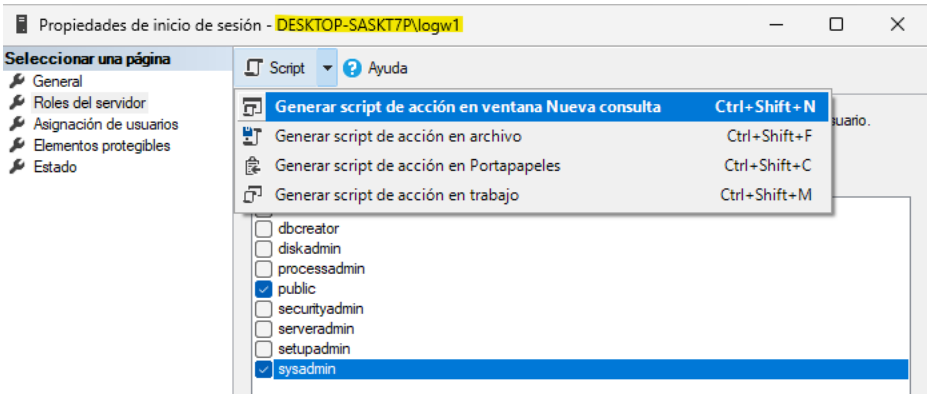
6.2. Con T-SQL: Siendo sa ejecutamos la instrucción:

```
execute sp_addsrvrolemember "DESKTOP-SASKT7P\logw1", sysadmin;
```

NOTA: *sp_addsrvrolemember* es un procedimiento almacenado del sistema que permite asignar un inicio de sesión a un rol de servidor.

6.3. Generamos el script T-SQL desde entorno gráfico:

Desde la opción Script generamos el código T-SQL:

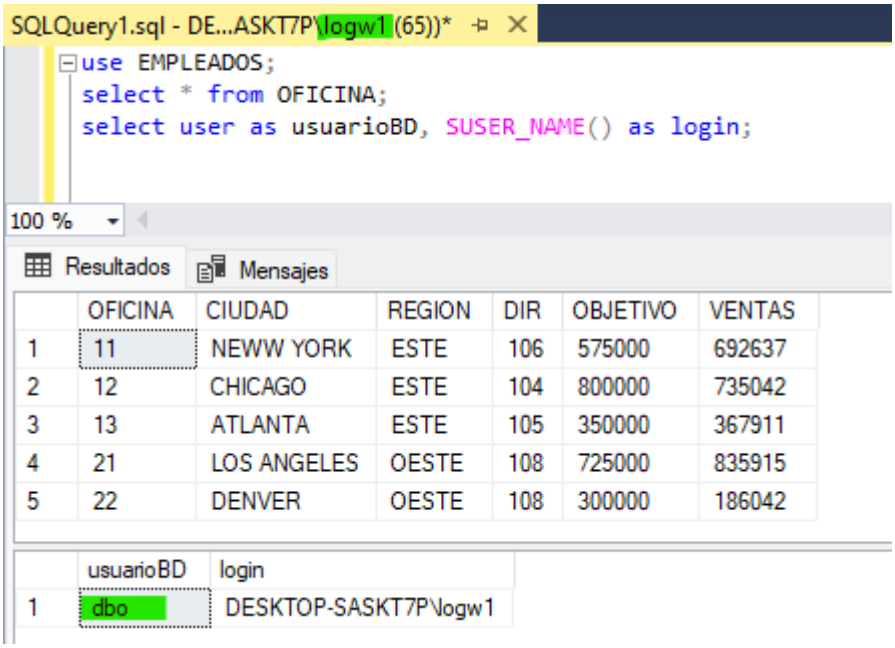


En una nueva ventana de consulta aparecerá el código siguiente que podemos ejecutar:

```
ALTER SERVER ROLE [sysadmin] ADD MEMBER [DESKTOP-SASKT7P\logw1]
GO
```

NOTA: Usar [] es equivalente a usar comillas.

Vamos a comprobar cómo el inicio de sesión logw1 ahora SÍ puede hacer use EMPLEADOS y consultar, por ejemplo, la tabla OFICINA:



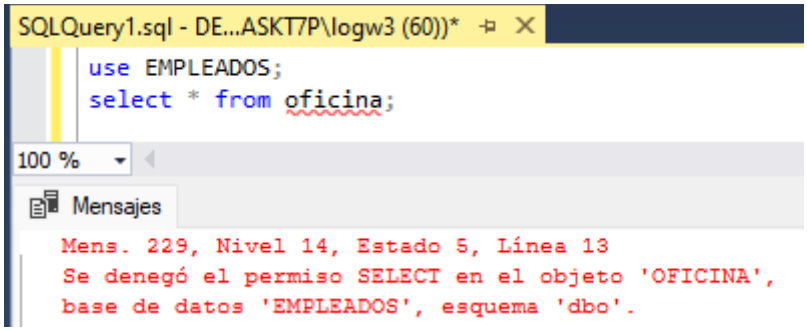
	OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
1	11	NEWW YORK	ESTE	106	575000	692637
2	12	CHICAGO	ESTE	104	800000	735042
3	13	ATLANTA	ESTE	105	350000	367911
4	21	LOS ANGELES	OESTE	108	725000	835915
5	22	DENVER	OESTE	108	300000	186042

	usuarioBD	login
1	dbo	DESKTOP-SASKT7P\logw1

El login se conecta a la BD EMPLEADOS como dbo. El usuario dbo de la BD EMPLEADOS (como en todas) pertenece al rol de bd ***db_owner***. Este rol tiene el permiso CONTROL sobre la BD por lo que puede realizar todas las actividades de configuración y mantenimiento en la base de datos, incluido eliminarla.

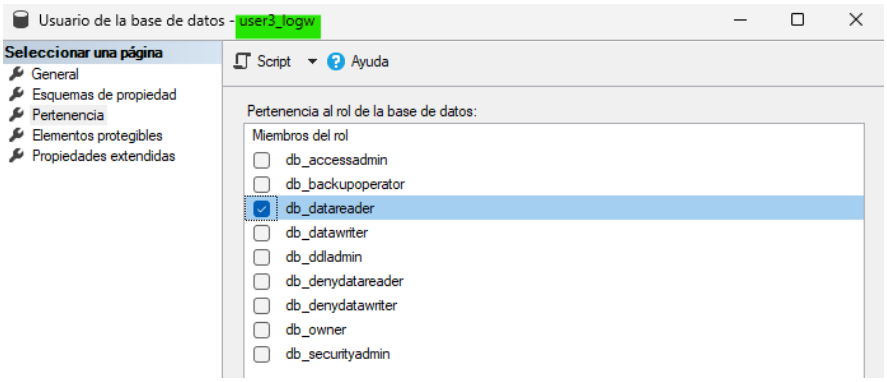
7. Hacer que un usuario de bd pueda hacer operaciones de lectura pero no de escritura:

Vamos a conectarnos con el login de Windows **logw3**. En la práctica guiada 01 a este login le asociamos el usuario **user3_logw** en la bd EMPLEADOS. Este usuario no tiene permisos actualmente, por lo que no podemos realizar ni un select sobre oficina:



Vamos a asignarle un rol de BD que le permita consultar (hacer SELECT) en los objetos de BD (también se podría hacer asignándole el permiso de SELECT con GRANT).

7.1. Desde entorno gráfico: siendo sa abrimos las propiedades del usuario de la BD EMPLEADOS y en la página Pertenencia marcamos el rol ***db_datareader***:



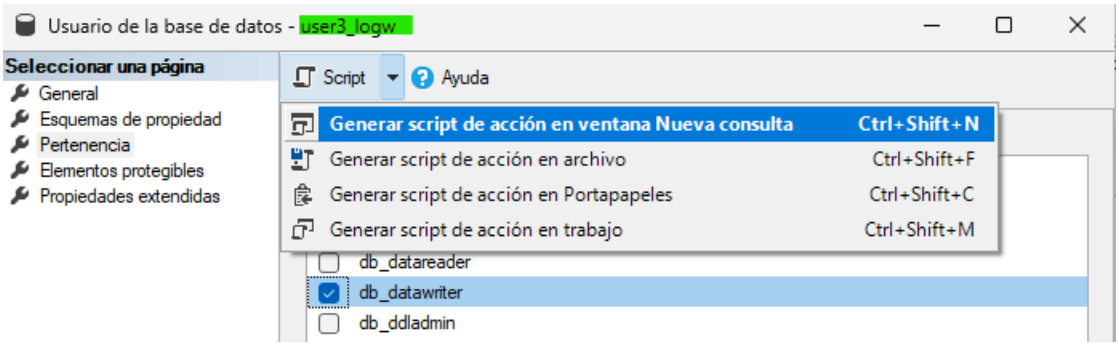
7.2. Con T-SQL: Siendo sa ejecutamos la instrucción:

```
use EMPLEADOS;
execute sp_addrolemember db_datareader, user3_logw;
```

IMPORTANTE: Al procedimiento *sp_addrolemember* primero se le pasa el rol y luego el usuario, a diferencia de *sp_**addsrv**rolemember* al que primero se le pasa el login y después el rol de servidor.

7.3. Generamos el script T-SQL desde entorno gráfico:

Desde la opción Script generamos el código T-SQL:



En una nueva ventana de consulta aparecerá el código siguiente que podemos ejecutar:

```
USE [EMPLEADOS]
GO
ALTER ROLE [db_datareader] ADD MEMBER [user3_logw ]
GO
```

7.4. Comprobación de permisos de user3_logw:

Comprobamos qué operaciones puede hacer ahora user3_logw en la BD EMPLEADOS .

Para ello lanzamos primero una consulta en la que se lean datos (SELECT). Debería poder ejecutar cualquier SELECT puesto que pertenece al rol ***db_datareader***.

Para comprobarlo lanzamos un SELECT sobre OFICINA y vemos como SÍ devuelve resultados.

```
USE EMPLEADOS;
select * from OFICINA;
```

100 %

Resultados | Mensajes

	OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
1	11	NEWW YORK	ESTE	106	575000	692637
2	12	CHICAGO	ESTE	104	800000	735042
3	13	ATLANTA	ESTE	105	350000	367911
4	21	LOS ANGELES	OESTE	108	725000	835915
5	22	DENVER	OESTE	108	300000	186042

Intentaremos hacer una operación de escritura, que no debe poder hacer porque no le hemos dado ese permiso directamente, no tampoco lo obtiene por heredarlo de un rol.

Intentamos modificar un campo en la tabla OFICINA y comprobamos que no nos lo permite porque nos deniega el permiso.

```
USE EMPLEADOS;
update OFICINA
set ciudad= 'NEW YORK'
where oficina=11;
```

100 %

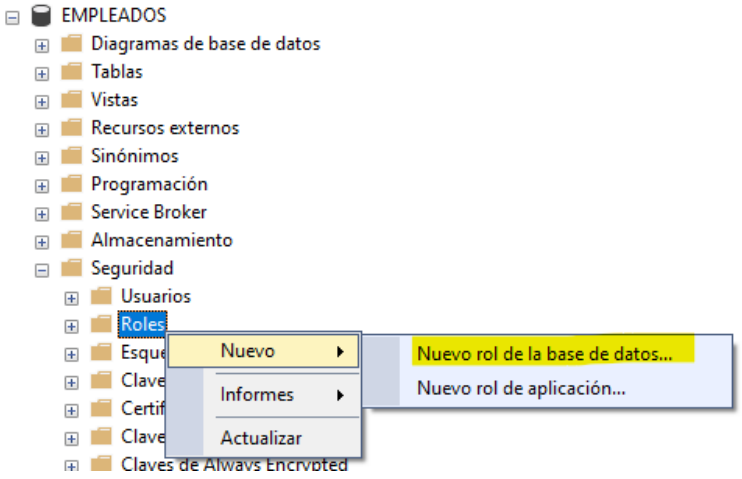
Mensajes

```
Mens. 229, Nivel 14, Estado 5, Línea 6
Se denegó el permiso UPDATE en el objeto 'OFICINA',
base de datos 'EMPLEADOS', esquema 'dbo'.
```

8. Rol de BD definido por el usuario:

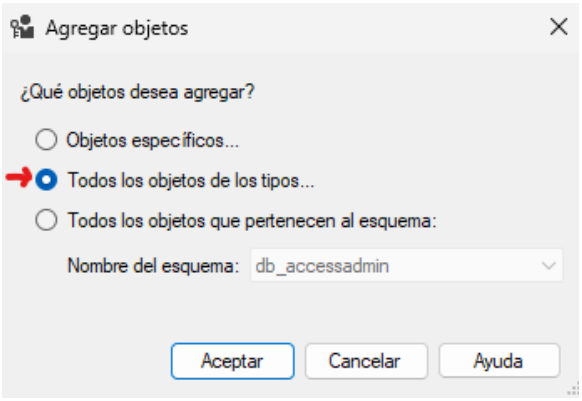
Siendo sa vamos a crear un rol de BD en la BD EMPLEADOS llamado **rol_update_emple**.

8.1. Desde entorno gráfico: siendo sa accedemos desde el explorador de objetos a la carpeta Roles de la BD EMPLEADOS:

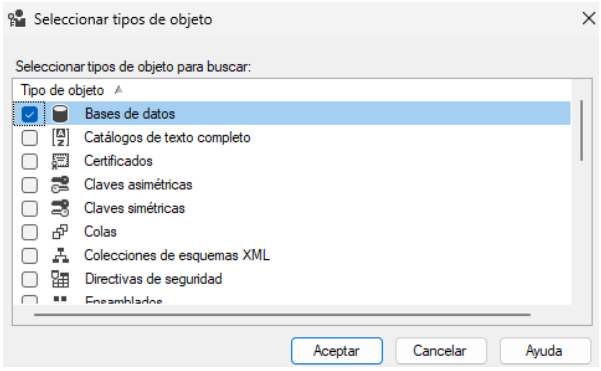


En la ventana que aparece indicamos el nombre del rol y como propietario escogerá el usuario con el que estamos creando ese rol (dbo).

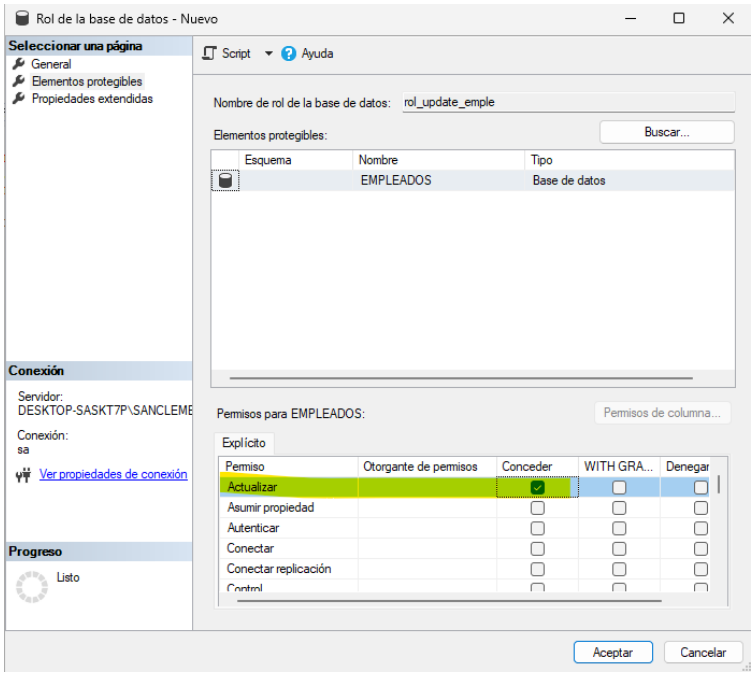
En la página **Elementos protegibles** Hacemos clic en **Buscar**:



Como tipo de objeto en la siguiente ventana escogemos *Bases de datos*:



Aparecerá la BD EMPLEADOS y marcamos el permiso *Actualizar*:

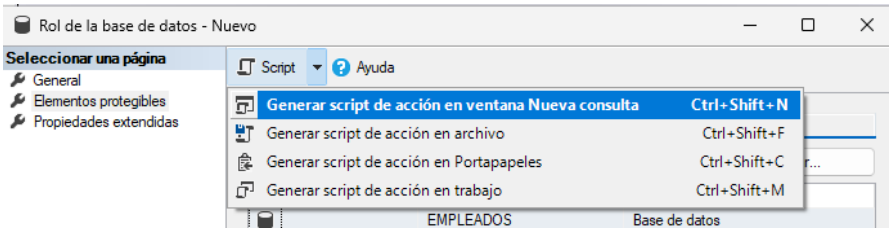


8.2. Con T-SQL: Siendo sa ejecutamos la instrucción:

```
use EMPLEADOS;
--Creamos el rol
create role rol_update_emple;
--Asignamos permisos al rol
GRANT UPDATE TO rol_update_emple;
```

8.3. Generamos el script T-SQL desde entorno gráfico:

Desde la opción Script generamos el código T-SQL:



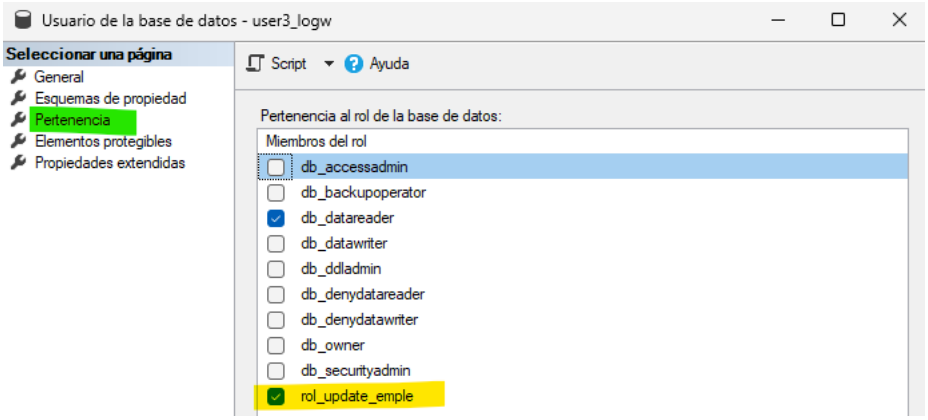
En una nueva ventana de consulta aparecerá el código siguiente que podemos ejecutar:

```
USE [EMPLEADOS]
GO
CREATE ROLE [rol_update_emple]
GO
use [EMPLEADOS]
GO
GRANT UPDATE TO [rol_update_emple]
GO
```

8.4. Asignamos el usuario user3_logw al rol rol_update_emple:

Hemos creado un rol de BD con permiso de UPDATE sobre la BD EMPLEADOS. Vamos a asignar el usuario **user3_logw** al rol **rol_update_emple**.

Siendo sa podemos hacerlo **por entorno gráfico** en las propiedades del usuario en la página **Pertenencia:**



O también con una de las siguientes instrucciones T-SQL:

Opción 1:	**Opción 2:**
USE EMPLEADOS; **ALTER ROLE** rol_update_emple **ADD** MEMBER user3_logw;	use EMPLEADOS; execute sp_addrolemember rol_update_emple, user3_logw;

8.5. Comprobación de permisos de user3_logw:

Vamos a comprobar si ahora puede realizar la modificación en OFICINA. En una ventana de consulta iniciada siendo el inicio de sesión logw3 ejecutamos lo siguiente:

```
use EMPLEADOS;
begin tran
select * from OFICINA where oficina=11;
update OFICINA
set ciudad='NEW YORK'
where oficina=11;
select * from OFICINA where oficina=11;
rollback;
```

100 %

Resultados

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
11	NEWW YORK	ESTE	106	575000	692637

(1 fila afectada)

(1 fila afectada) --Hace referencia a la fila modificada

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
11	NEW YORK	ESTE	106	575000	692637

(1 fila afectada)

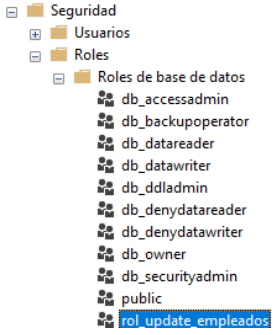
Comprobamos como ya no da error y actualiza la fila porque el usuario pertenece al rol **rol_update_emple.**

9. Cambiar el nombre de un rol:

Vamos a cambiar el nombre al rol **rol_update_emple** y le vamos a llamar **rol_update_empleados**. Siendo sa ejecutaremos la siguiente instrucción T-SQL:

```
USE EMPLEADOS;
ALTER ROLE rol_update_emple  WITH NAME= rol_update_empleados;
```

Comprobamos el cambio de nombre en la carpeta de Seguridad->Roles de la BD EMPLEADOS:

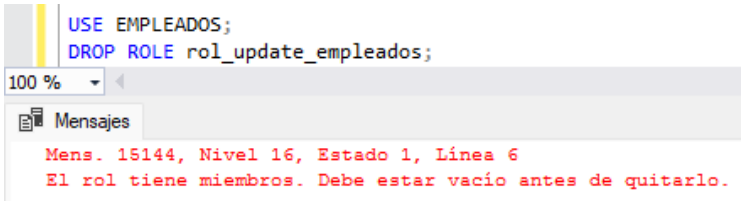


10. Eliminar un rol:

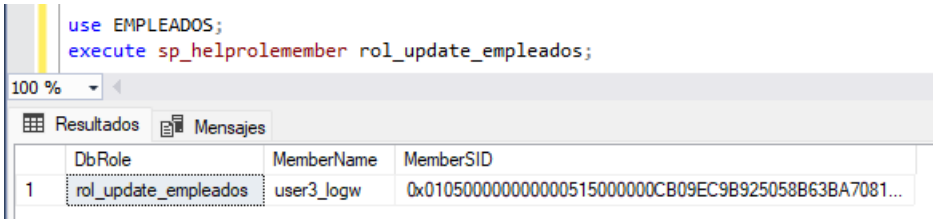
Vamos a eliminar el **rol_update_empleados**. Siendo sa ejecutaremos la siguiente instrucción T-SQL:

```
USE EMPLEADOS;
DROP ROLE rol_update_empleados;
```

Dará un error porque para eliminar un rol este debe estar vacío, es decir no debe tener usuarios asociados:



Comprobamos qué usuarios tiene el rol ejecutando el procedimiento almacenado del sistema **sp_helprolemember**:

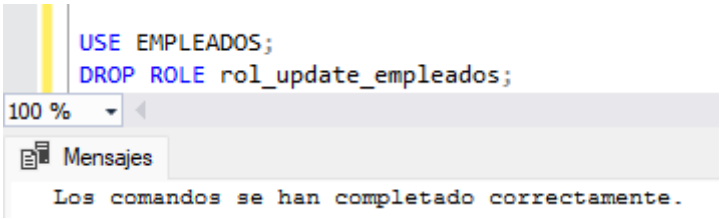


Después de comprobar que el único miembro del rol **rol_update_empleados** es **user3_logw**, procedemos a quitarlo del rol. Siendo sa podemos hacerlo **por entorno gráfico** desmarcando el rol en las propiedades del usuario en la página **Pertenencia.**

O también con una de las siguientes instrucciones T-SQL:

Opción 1:	**Opción 2:**
USE EMPLEADOS; **ALTER ROLE** rol_update_empleados **DROP** MEMBER user3_logw;	use EMPLEADOS; execute sp_droprolemember rol_update_empleados, user3_logw;

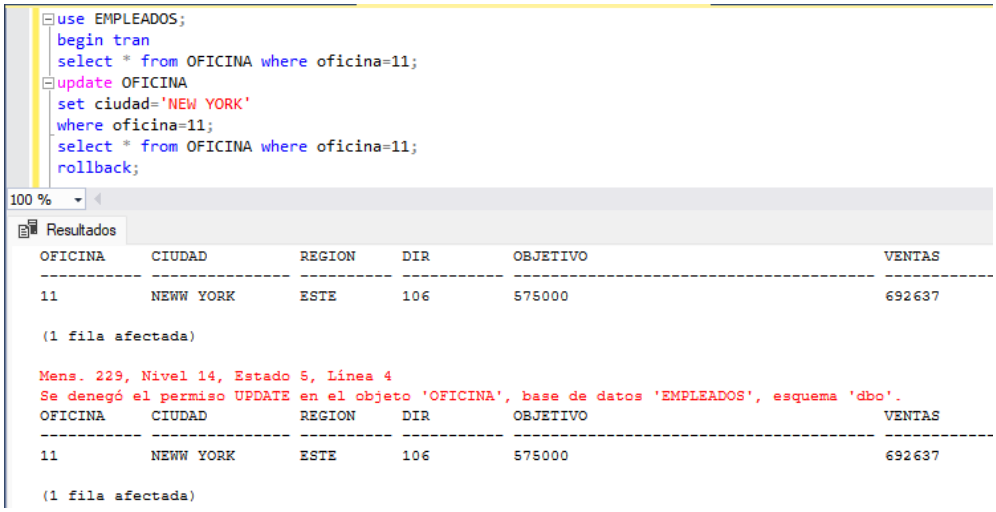
Una vez eliminado el usuario del rol podremos ejecutar la instrucción de borrado del rol:



11. Rol public:

Todos los usuarios de la BD pertenecen al rol **public**, los actuales y los que podamos crear.

Después de borrar el rol **rol_update_empleados**, el usuario **user3_logw** ya no puede realizar instrucciones UPDATE, ya que recibía ese permiso a través del rol. Lo comprobamos lanzando un update sobre OFICINA.



```
use EMPLEADOS;
begin tran
select * from OFICINA where oficina=11;
update OFICINA
set ciudad='NEW YORK'
where oficina=11;
select * from OFICINA where oficina=11;
rollback;
```

100 %

Resultados

```
OFICINA     CIUDAD          REGION     DIR         OBJETIVO                                VENTAS
----------- --------------- ---------- ----------- --------------------------------------- -------------
11          NEWW YORK       ESTE       106         575000                                  692637

(1 fila afectada)

Mens. 229, Nivel 14, Estado 5, Línea 4
Se denegó el permiso UPDATE en el objeto 'OFICINA', base de datos 'EMPLEADOS', esquema 'dbo'.
OFICINA     CIUDAD          REGION     DIR         OBJETIVO                                VENTAS
----------- --------------- ---------- ----------- --------------------------------------- -------------
11          NEWW YORK       ESTE       106         575000                                  692637

(1 fila afectada)
```

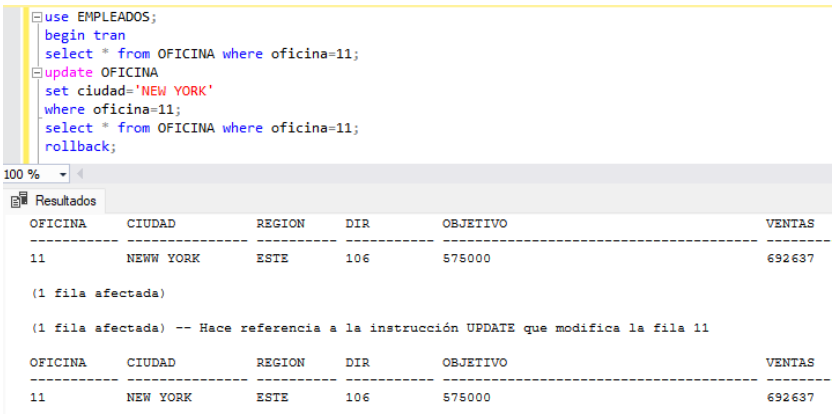
Supongamos que queremos que todos los usuarios de la BD EMPLEADOS puedan hacer UPDATE sobre la tabla OFICINA. Podríamos coger cada usuario y hacer GRANT update ON OFICINA TO *usuario* para cada uno de ellos. Si lo hacemos así y cuando creamos un usuario nuevo debemos acordarnos de darle el permiso.

IMPORTANTE: Si queremos que **todos los usuarios ACTUALES y FUTUROS** de nuestra BD tengan unos permisos determinados, en lugar de otorgarlos a cada usuario, los concedemos al rol **public**.

En este caso debemos ejecutar la siguiente instrucción T-SQL:

```
USE EMPLEADOS;
GRANT update ON OFICINA TO public;
```

El usuario user3_logw ya puede realizar la instrucción UPDATE sobre OFICINA porque hereda el permiso del rol **public**:



```
use EMPLEADOS;
begin tran
select * from OFICINA where oficina=11;
update OFICINA
set ciudad='NEW YORK'
where oficina=11;
select * from OFICINA where oficina=11;
rollback;
```

100 %

Resultados

```
OFICINA     CIUDAD          REGION     DIR         OBJETIVO                                VENTAS
----------- --------------- ---------- ----------- --------------------------------------- --------
11          NEWW YORK       ESTE       106         575000                                  692637

(1 fila afectada)

(1 fila afectada) -- Hace referencia a la instrucción UPDATE que modifica la fila 11

OFICINA     CIUDAD          REGION     DIR         OBJETIVO                                VENTAS
----------- --------------- ---------- ----------- --------------------------------------- --------
11          NEW YORK        ESTE       106         575000                                  692637
```