

Esquemas y permisos

1. ¿Qué es un esquema?:

Podemos decir que un esquema es un **contenedor de objetos de bases de datos** como tablas, vistas, funciones, procedimientos almacenados...

Ventajas principales del uso de esquemas:

- Mantener los **datos organizados**, podemos crear esquemas como si fueran carpetas con objetos de BD en su interior.
- Gestionar la **seguridad** (acceso y permisos) para todos los objetos de un esquema.

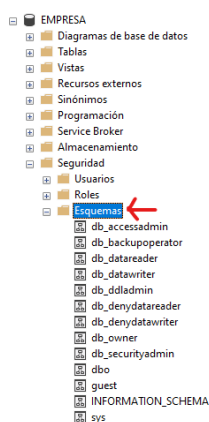
2. Esquema dbo:

El esquema dbo **es el esquema por defecto**, es decir, será el esquema asignado a un usuario si cuando lo creamos no le hemos asignado ninguno.

Cuando ese usuario crea objetos, a menos que el usuario indique otro esquema, los objetos se crearán en el esquema por defecto del usuario creador.

3. Consultar los esquemas de una BD:

Para consultar los esquemas de un BD podemos hacerlo por **entorno gráfico** en la carpeta de **Seguridad->Esquemas** de la base de datos, en este caso la BD EMPRESA:



También podemos lanzar en la BD la siguiente instrucción T-SQL: `select * from sys.schemas;`

```
1 use EMPRESA;
2 select * from sys.schemas;
```

	name	schema_id	principal_id
1	dbo	1	1
2	guest	2	2
3	INFORMATION_SCHEMA	3	3
4	sys	4	4
5	db_owner	16384	16384
6	db_accessadmin	16385	16385
7	db_securityadmin	16386	16386
8	db_ddladmin	16387	16387
9	db_backupoperator	16389	16389
10	db_datareader	16390	16390
11	db_datawriter	16391	16391
12	db_denysdatareader	16392	16392
13	db_denysdatawriter	16393	16393

Esquemas y permisos

4. Esquema para las vistas de la BD EMPRESA:

Necesitamos dar **acceso de sólo de lectura a algunos datos de nuestra base de datos EMPRESA**. Para ello vamos a crear un esquema y dentro de éste las vistas que darán acceso a los datos.

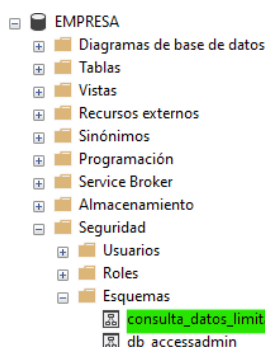
Posteriormente, crearemos un usuario con permiso de SELECT sobre el esquema.

5. Creación de un esquema en la BD EMPRESA para las vistas:

Nos conectamos al servidor de SQL Server como **sa** y creamos el esquema **consulta_datos_limitada**:

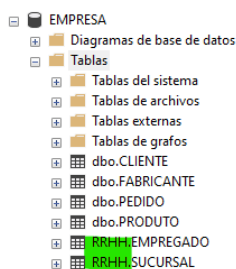
```
use EMPRESA;
--Es necesario indicar GO porque
--CREATE SCHEMA debe ser la 1ª instrucción en un lote
--de consultas
GO
create schema consulta_datos_limitada;
--Al no indicar propietario del esquema,
--se asigna dbo (usuario que está creando el esquema)
```

Comprobamos en el explorador de objetos como aparece como un esquema nuevo:

**6. Creación de varios esquemas en la BD EMPRESA para las tablas:****6.1. Vamos a crear un esquema RRHH para EMPREGADO y SUCURSAL.**

```
use EMPRESA;
GO
create schema RRHH;
GO
alter schema RRHH
transfer dbo.EMPREGADO;
GO
alter schema RRHH
transfer dbo.SUCURSAL;
```

Comprobamos en el explorador de objetos el cambio de esquema de las tablas:

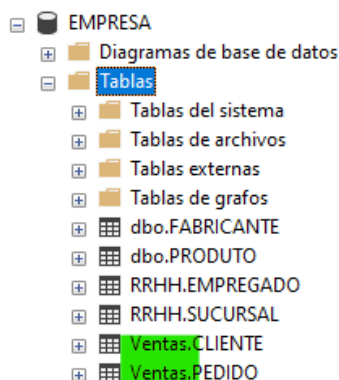


Esquemas y permisos

6.2. Vamos a crear un esquema Ventas para CLIENTE y PEDIDO.

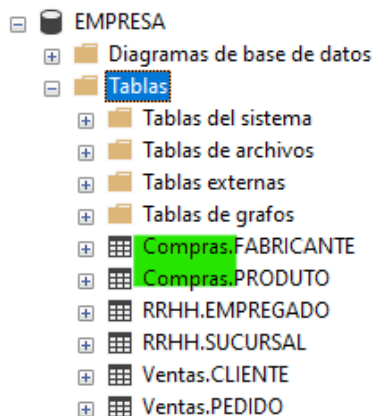
```
use EMPRESA;  
GO  
create schema Ventas;  
GO  
alter schema Ventas  
transfer dbo.CLIENTE;  
GO  
alter schema Ventas  
transfer dbo.PEDIDO;
```

Comprobamos en el explorador de objetos el cambio de esquema de las tablas:

**6.3. Vamos a crear un esquema Compras para FABRICANTE y PRODUTO.**

```
use EMPRESA;  
GO  
create schema Compras;  
GO  
alter schema Compras  
transfer dbo.FABRICANTE;  
GO  
alter schema Compras  
transfer dbo.PRODUTO;
```

Comprobamos en el explorador de objetos el cambio de esquema de las tablas:



7. Creamos las vistas en el esquema *consulta_datos_limitada*:

- **v_datos_vendedor:** Vista que contenga el nombre completo del empleado y la fecha de nacimiento (EMPREGADO), así como la ciudad y la región (SUCURSAL) de la sucursal en la que trabajan. El texto de la vista debe estar cifrado:

```
use EMPRESA;
GO
--Es necesario indicar GO porque
--CREATE VIEW debe ser la 1ª instrucción en un lote
--de consultas
create view consulta_datos_limitada.v_datos_vendedor
WITH ENCRYPTION
as
select e.nombre,e.apel, isnull(ape2,'') as ape2, s.ciudad, s.region
from RRHH.EMPREGADO e inner join RRHH.SUCURSAL s
on e.id_sucursal_trabaja=s.identificador;
```

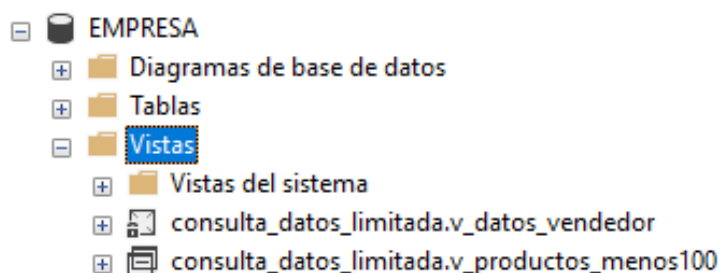
- **v_productos_menos100:** Vista que contenga todas las columnas de los productos cuyo precio sea inferior a 100€. Esta vista no permitirá hacer ningún tipo de actualización que vulnere la condición del where, por ejemplo, no debe permitir insertar productos con precio mayor o igual que 100, ni cambiar el precio de un producto para que sea ≥ 100 .

Las columnas de las vistas se llamarán: COD_FAB, ID_PROD, NOMBRE, IMPORTE, STOCK

Si se consulta la información de metadatos devuelva los de la vista y no de los objetos subyacentes.

```
use EMPRESA;
GO
create view consulta_datos_limitada.v_productos_menos100
(COD_FAB, ID_PROD, NOMBRE, IMPORTE, STOCK)--Aquí cambiamos los nombres de los
--campos de la vista. También se puede hacer con alias en el SELECT de la vista.
WITH VIEW_METADATA --Para que se consulten los metadatos de la vista
as
select cod_fabricante, identificador, descripcion,
prezo, existencias
from Compras.produto
where prezo < 100
WITH CHECK OPTION;--Para que las actualizaciones contra la vista
--verifiquen las condiciones de la vista.
--En este caso prezo < 100
```

Podemos comprobar que se han creado en el explorador de objetos. Fíjate que el icono de la vista que está cifrada es distinto y si intentamos ver su diseño con el menú contextual no lo permite:



Para consultar las vistas creadas **debemos acordarnos de indicar el esquema**. Siendo sa ejecutamos:

```
use EMPRESA;
select * from v_datos_vendedor; -- Nos indicará que el nombre v_datos_vendedor no existe

Mens. 208, Nivel 16, Estado 1, Línea 49
El nombre de objeto 'v_datos_vendedor' no es válido.

-- Para no obtener el error anterior debemos
-- hacer referencia a la vista
-- indicando el esquema dónde se creó
select * from consulta_datos_limitada.v_datos_vendedor;
```

IMPORTANTE: Podríamos NO indicar el esquema si el usuario con el que estamos conectados tiene ese esquema asignado como esquema por defecto.

Debemos tener en cuenta que **un usuario puede ser propietario de varios esquemas, pero sólo tiene asignado uno por defecto**.

8. Usuarios con acceso sólo de consulta a las vistas:

Para que un usuario, o varios sólo tengan acceso de consulta a las vistas procederemos del siguiente modo:

8.1. Creamos un rol de nombre *rol_vistas*:

```
use EMPRESA;
GO
create role rol_vistas;
```

8.2. Le damos permiso sólo de *SELECT* sobre el esquema de las vistas:

```
use EMPRESA;
GRANT SELECT
on SCHEMA::consulta_datos_limitada
TO rol_vistas;
```

Ahora cuando queramos que un usuario tenga acceso sólo a las vistas lo asignaremos al rol de nombre *rol_vistas*.

9. Creación de un rol con acceso completo a las tablas y a las vistas:

Para que un usuario, o varios tengan acceso completo a las tablas y a las vistas procederemos del siguiente modo:

9.1. Creamos un rol de nombre *rol_tablas*:

```
use EMPRESA;
GO
create role rol_tablas;
```

9.2. Le damos permisos de consulta y actualización sobre los esquemas de las tablas:

```
use empresa;
GO
GRANT SELECT, INSERT, UPDATE, DELETE
ON SCHEMA::RRHH
TO rol_tablas;
GO
GRANT SELECT, INSERT, UPDATE, DELETE
ON SCHEMA::Ventas
TO rol_tablas;
GO
GRANT SELECT, INSERT, UPDATE, DELETE
ON SCHEMA::Compras
TO rol_tablas;
```

Ahora cuando queramos que un usuario tenga acceso completo a tablas y vistas (normalmente los administradores) lo asignaremos al rol de nombre **rol_tablas**.

10. Creación de usuarios y asignación de roles:

Vamos a crear dos usuarios, uno **maria_jefa** que sólo puede consultar vistas y **anton_admin** que puede modificar y consultar tanto tablas como vistas.

10.1. Creamos los login y los usuarios (con el mismo nombre que el login) con BD por defecto EMPRESA:

Al coincidir el nombre del usuario con el del login no es necesario indicar FOR LOGIN en el create user.

```
--Usuario jefe departamento
create login maria_jefa
with password = 'maria_jefa',
    default_database=EMPRESA;

USE EMPRESA;
create user maria_jefa;

--Usuario administrador
create login anton_admin
with password = 'anton_admin',
    default_database=EMPRESA;
USE EMPRESA;
create user anton_admin;
```

10.2. Asignamos los usuarios a los roles que nos interesa según los permisos que queremos que tengan:

El usuario **maria_jefa** sólo puede consultar vistas por lo que tenemos que asignarlo al **rol_vistas**:

```
-- Usuario con permisos solo sobre vistas
EXEC sp_addrolemember rol_vistas, maria_jefa;
```

Esquemas y permisos

El usuario **anton_admin** sólo puede acceder a vistas y tablas por lo que tenemos que asignarlo a ambos roles **rol_vistas** y **rol_tablas**:

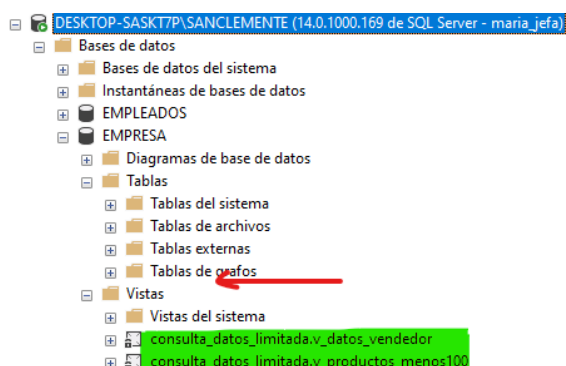
```
-- Usuario con permisos sobre tablas y vistas
EXEC sp_addrolemember rol_tablas, anton_admin;
EXEC sp_addrolemember rol_vistas, anton_admin;
```

11. Comprobación de permisos según los roles a los que pertenecen:

11.1. Conexión como maria_jefa y comprobación de permisos:

Nos conectamos como maria_jefa y abrimos una nueva consulta.

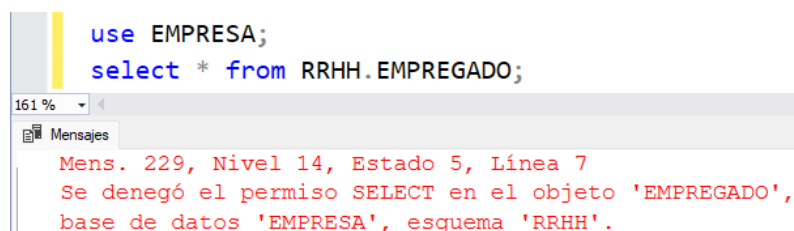
En el propio explorador de objetos comprobamos como al desplegar la carpeta de Tablas no aparecen las tablas de la BD pero sí las vistas:



Lanzamos una consulta sobre la vista **consulta_datos_limitada.v_datos_vendedor** y comprobamos cómo podemos consultarla sin problema.



La vista está definida sobre las tablas RRHH.EMPREGADO y RRHH.SUCURSAL. Si intentamos acceder a cualquiera de las dos tablas veremos que no nos lo permite, ya que el usuario **maria_jefa** sólo pertenece al rol **rol_vistas**:

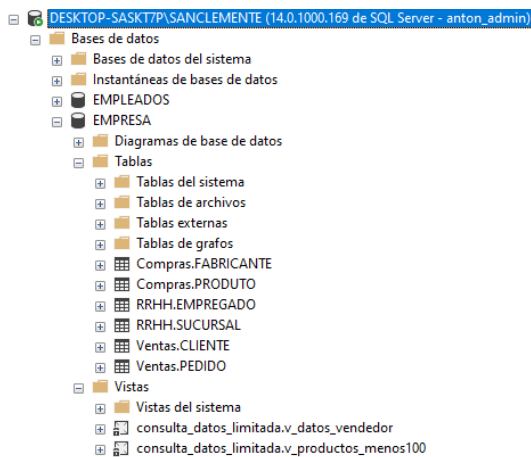


Esquemas y permisos

11.2. Conexión como *anton_admin* y comprobación de permisos:

Nos conectamos como **anton_admin** y abrimos una nueva consulta.

En el propio explorador de objetos comprobamos como al desplegar las carpetas de Tablas y Vistas aparecen todas:



Lanzamos una consulta sobre la vista *consulta_datos_limitada.v_datos_vendedor* y comprobamos cómo podemos consultarla sin problema.

```
use EMPRESA;
select * from consulta_datos_limitada.v_datos_vendedor;
```

	nome	ape1	ape2	cidade	rexion
1	DANIEL	GARCIA	XIL	VALENCIA	LESTE
2	SUSANNE	SMITH		A CORUÑA	OESTE
3	PAULA	CRUZ	SOUTO	VALENCIA	LESTE
4	MARCOS	CHANS	PÉREZ	VALENCIA	LESTE
5	ANTIA	GONZÁLEZ	FERREIRA	MURCIA	LESTE
6	MARTÍN	DELGADO	MONTERO	BARCELONA	LESTE
7	ANA	MARTINEZ	IGLESIAS	VIGO	OESTE
8	LARA	GARCIA	PAZOS	A CORUÑA	OESTE
9	MARIA	SEARA	JANEIRO	BARCELONA	LESTE
10	CARLOS	GRIMM		VALENCIA	LESTE

La vista está definida sobre las tablas RRHH.EMPREGADO y RRHH.SUCURSAL. Si intentamos acceder a cualquiera de las dos tablas veremos que sí nos lo permite, ya que el usuario **anton_admin** pertenece al rol **rol_vistas** y al **rol_tablas**:

```
use EMPRESA;
select * from RRHH.SUCURSAL;
```

	identificador	cidade	rexion	num_empleado_director	obxectivo
1	11	BARCELONA	LESTE	106	575000.00
2	12	VALENCIA	LESTE	104	800000.00
3	13	MURCIA	LESTE	105	350000.00
4	21	A CORUÑA	OESTE	108	725000.00
5	22	VIGO	OESTE	108	300000.00

Esquemas y permisos

El usuario **anton_admin** también podrá hacer actualizaciones por pertenecer al rol **rol_tablas**:

```

begin tran
select * from RRHH.SUCURSAL;
update RRHH.SUCURSAL
set cidade='LONDRES'
where identificador=11;
select * from RRHH.SUCURSAL;
rollback;

```

	identificador	cidade	rexion	num_empleado_director	objetivo
1	11	BARCELONA	LESTE	106	575000.00
2	12	VALENCIA	LESTE	104	800000.00
3	13	MURCIA	LESTE	105	350000.00
4	21	A CORUÑA	OESTE	108	725000.00
5	22	VIGO	OESTE	108	300000.00

	identificador	cidade	rexion	num_empleado_director	objetivo
1	11	LONDRES	LESTE	106	575000.00
2	12	VALENCIA	LESTE	104	800000.00
3	13	MURCIA	LESTE	105	350000.00
4	21	A CORUÑA	OESTE	108	725000.00
5	22	VIGO	OESTE	108	300000.00

12. Permisos de objeto:

Hemos visto en los apartados anteriores cómo dar permisos (**GRANT**) a un rol sobre un esquema.

Los permisos pueden darse también directamente a usuarios, pero también se pueden revocar (**REVOKE**) y denegar (**DENY**).

Se pueden dar o quitar permisos:

- **de instrucción**, o,
- **de objeto**: específicos sobre un objeto determinado (esquema, tabla, vista, procedimiento almacenado...)

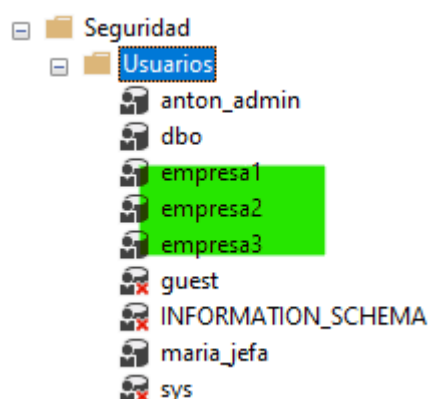
12.1. Creamos 2 usuarios nuevos en la BD EMPRESA para hacer pruebas:

Vamos a crear los usuarios **empresa1**, **empresa2** y **empresa3**.

```

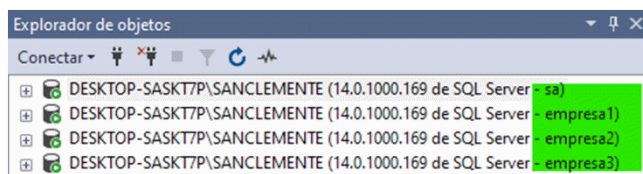
--empresa1
create login empresa1
with password = 'empresa1',
    default_database=EMPRESA;
USE EMPRESA;
create user empresa1;
--empresa2
create login empresa2
with password = 'empresa2',
    default_database=EMPRESA;
USE EMPRESA;
create user empresa2;
--empresa3
create login empresa3
with password = 'empresa3',
    default_database=EMPRESA;
USE EMPRESA;
create user empresa3;

```



Esquemas y permisos

Debemos tener 4 conexiones: una con **sa** conectado, otra con **empresa1**, otra con **empresa2** y otra con **empresa3**.

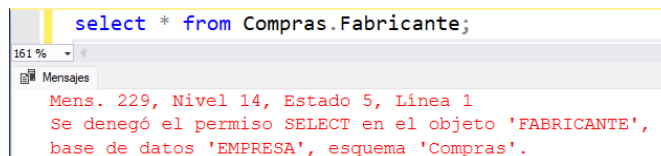


Abriremos una nueva consulta de cada conexión:

UD11_PracticaGuiad....EMPRESA (sa (51))* SQLQuery6.sql - DE...SA (empresa1 (59)) SQLQuery7.sql - DE...SA (empresa2 (62)) SQLQuery8.sql - DE...SA (empresa3 (66))

12.2. Dar permisos a usuarios actuales y futuros:

Ahora mismo aunque tienen BD por defecto EMPRESA al no tener ningún permiso directo ni heredado (pertenecen al rol **public** y éste no tiene permisos), no pueden realizar ni un simple SELECT. Probamos con empresa1:



Supongamos que queremos que todos los **usuarios, los actuales y los futuros**, tengan permisos de *consulta sobre las vistas* (que estarán siempre en el esquema **consulta_datos_limitada**).

Siendo **sa** ejecutamos la siguiente instrucción:

```
USE EMPRESA;
GRANT SELECT
ON SCHEMA::consulta_datos_limitada
TO public;
```

IMPORTANTE: Aunque ya tenemos un rol creado, **rol_vistas**, que permite hacer SELECT en el esquema de vistas, NO PODEMOS asignar ese rol a public, porque **SQL Server no permite asignar roles a public**, sólo permisos.

Como usuario **empresa1** comprobamos como ahora podemos consultar la vista de productos de menos de 100€ **consulta_datos_limitada.v_productos_menos100** pero no la tabla con los datos de los productos que se usa en el select de la vista:

```
select * from consulta_datos_limitada.v_productos_menos100;
```

COD_FAB	ID_PROD	NOMBRE	IMPORTE	STOCK
1	ASU	Tarjeta gráfica SVGA Asus NVIDIA GeForce 210 Sil...	34.90	20
2	KIN	HD SSD 120GB 2.5 SATA3 v300	54.80	150
3	KIN	DDR3 4GB PC1600 CL11 DIMM SRX8	36.80	20
4	KIN	DDR3 SO DIMM 4GB PC1333 CL9 SR	39.60	0
5	LOG	mk270 combo teclado con rato óptico	26.00	5
6	LOG	rato óptico logitech b100 negro	9.90	20
7	LOG	HD Webcam C270	30.90	12
8	LOG	USB Headset H540	59.90	37
9	LOG	3D PRO Joystick	51.90	30
10	LOG	Bluetooth Audio Adapter	39.99	7
11	SAIM	Tarjeta de memoria SD PRO Clase 10 UHS-I de 16 GB	26.50	140
12	SAIM	Cable USB a micro USB	11.99	120
13	SAIM	Funda Book Cover para Samsung Galaxy Tab S 8.4" ...	59.90	30
14	SEA	HDD 1TB 7200rpm 64MB SATA3 6gbps	52.80	2
15	SEA	HD 2.5 500GB 8MB 5400rpm SATA2	46.40	15
16	TOS	Pen Drive Daichi 32 GB 3.0 azul	16.95	28
17	TOS	Rato Toshiba W30 Óptico sen Fios negro	19.90	22
18	TOS	Disco Duro interno Toshiba MQ Series 1TB 2.5' SATA	69.90	24

```
select * from Compras.Producto;
```

Mens. 208, Nivel 16, Estado 1, Línea 6
El nombre de objeto 'Compras.Producto' no es válido.

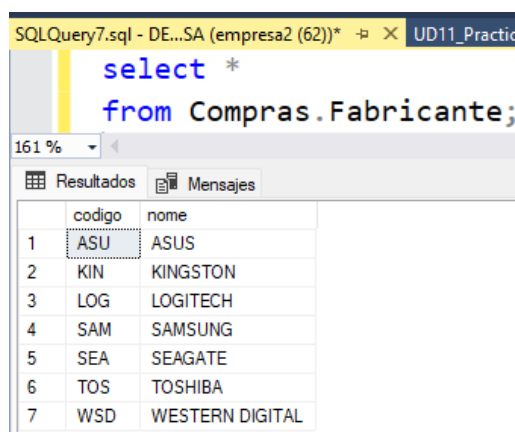
Puedes repetir la consulta sobre cualquiera de las dos vistas en las pestañas de **empresa2** y **empresa3** para ver cómo los usuarios tienen permiso para consultarlas, por pertenecer al rol **public**.

12.3. Dar permisos con la posibilidad de concederlos a otros:

Vamos a darle a **empresa2** el permiso de SELECT sobre la tabla *Compras.Fabricante* de tal modo que puede darle ese permiso a otros usuarios. Siendo **sa** ejecutamos la siguiente instrucción:

```
use EMPRESA;  
grant select  
on Compras.Fabricante  
to empresa2  
with grant option;
```

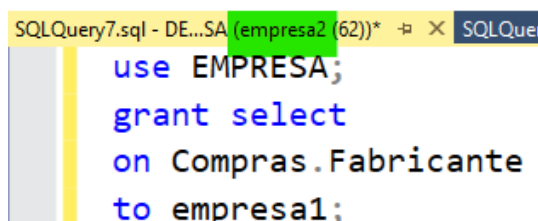
Nos conectamos como **empresa2** y comprobamos como ahora podemos consultar la tabla *Compras.Fabricante*.



```
SQLQuery7.sql - DE...SA (empresa2 (62))* X UD11_Practic  
select *  
from Compras.Fabricante;
```

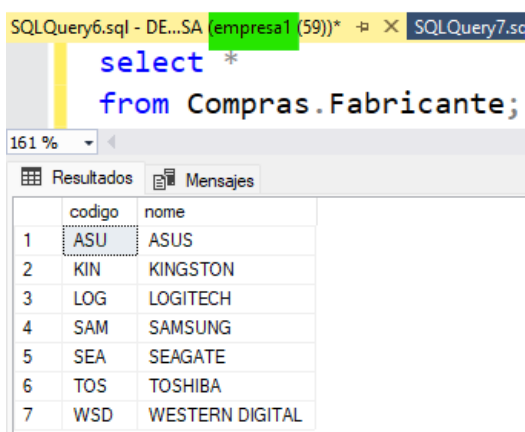
	codigo	nome
1	ASU	ASUS
2	KIN	KINGSTON
3	LOG	LOGITECH
4	SAM	SAMSUNG
5	SEA	SEAGATE
6	TOS	TOSHIBA
7	WSD	WESTERN DIGITAL

Seguimos en la sesión de **empresa2** y daremos permiso de SELECT en *Compras.Fabricante* al usuario **empresa1**:



```
SQLQuery7.sql - DE...SA (empresa2 (62))* X SQLQuer  
use EMPRESA;  
grant select  
on Compras.Fabricante  
to empresa1;
```

En la sesión de **empresa1** comprobamos como ahora SÍ puede consultar los fabricantes:



```
SQLQuery6.sql - DE...SA (empresa1 (59))* X SQLQuery7.sc  
select *  
from Compras.Fabricante;
```

	codigo	nome
1	ASU	ASUS
2	KIN	KINGSTON
3	LOG	LOGITECH
4	SAM	SAMSUNG
5	SEA	SEAGATE
6	TOS	TOSHIBA
7	WSD	WESTERN DIGITAL

12.4. Quitar permisos que se han dado con GRANT OPTION:

Siendo **sa** le quitamos el permiso de SELECT a **empresa2**. Al haberlo dado con GRANT OPTION hay que quitarlo con **CASCADE**. Esto hace que no sólo **empresa2** pierda el permiso, sino también todos los usuarios a los que se lo haya concedido, en este caso **empresa1**.

```
UD11_PracticaGuiad....EMPRESA (sa (51))* SQLQuery7.sql - DE...SA (empresa2 (62))* SQLQuery6.sql - DE...SA

revoke select
on Compras.Fabricante
from empresa2
cascade;--Si se ha dado el permiso con GRANT OPTION
--hay que quitarlo con CASCADE
```

Comprobamos como **empresa2** NO puede consultar los fabricantes.

```
SQLQuery7.sql - DE...SA (empresa2 (62))* SQLQuery6.sql - DE...SA (empresa1 (59))* UD11_PracticaG

select *
from Compras.Fabricante;
```

161 %

Mensajes

Mens. 229, Nivel 14, Estado 5, Línea 1
Se denegó el permiso SELECT en el objeto 'FABRICANTE',
base de datos 'EMPRESA', esquema 'Compras'.

Ahora **empresa1** tampoco podrá consultar los fabricantes porque recibió el permiso de **empresa2** y éste ya no lo tiene.

```
SQLQuery6.sql - DE...SA (empresa1 (59))* SQLQuery7.sql - DE...SA (empresa2 (62))* UD11_PracticaG

select *
from Compras.Fabricante;
```

161 %

Mensajes

Mens. 229, Nivel 14, Estado 5, Línea 6
Se denegó el permiso SELECT en el objeto 'FABRICANTE',
base de datos 'EMPRESA', esquema 'Compras'.

12.5. Quitar la posibilidad de hacer GRANT pero no el permiso:

Vamos a volver a dar a **empresa2** el permiso de SELECT sobre los fabricantes y con la posibilidad de concederlo a otros). También siendo **empresa2** concederemos el permiso a **empresa1**. Es decir, repite las instrucciones GRANT del apartado 12.3.

Supongamos que nos damos cuenta de que **empresa2** hace un mal uso de su privilegio de conceder el permiso (GRANT OPTION FOR), pero queremos que siga consultando datos de fabricantes.

Esquemas y permisos

Para **quitarle sólo el permiso de concederlo a los demás**, debemos usar la instrucción **revoke** con **GRANT OPTION FOR permiso**. Para quitarle a empresa2 el permiso de conceder SELECT sobre Fabricantes , realizaremos la siguiente instrucción siendo **sa**:

```
revoke GRANT OPTION FOR SELECT
on Compras.Fabricante
from empresa2
cascade;
```

Ahora podremos comprobar que **empresa2** puede seguir haciendo SELECT de *Compras.Fabricante* pero **no puede dar el permiso**.

The screenshot shows two SQL query windows. The first window, titled 'SQLQuery6.sql - DE...SA (empresa1 (59))', contains the following SQL code:

```
select *
from Compras.Fabricante;

use EMPRESA;
grant select
on Compras.Fabricante
to empresa1;
```

The second window, titled 'SQLQuery7.sql - DE...SA (empresa2 (62))', shows the results of a SELECT query on 'Compras.Fabricante'. The results are displayed in a table with columns 'codigo' and 'nome'.

codigo	nome
ASU	ASUS
KIN	KINGSTON
LOG	LOGITECH
SAM	SAMSUNG
SEA	SEAGATE
TOS	TOSHIBA
WSD	WESTERN DIGITAL

Below the results, it indicates '(7 filas afectadas)'. A red error message is displayed at the bottom of the window:

```
Mens. 15151, Nivel 16, Estado 1, Línea 5
No se puede buscar el objeto 'FABRICANTE'
porque no existe o el usuario no tiene permiso.
```

Y comprobamos que **empresa1** no puede hacer consultas de fabricantes, porque el permiso se lo había concedido **empresa2**.

The screenshot shows a SQL query window titled 'SQLQuery6.sql - DE...SA (empresa1 (59))' with the following SQL code:

```
select *
from Compras.Fabricante;
```

The window shows a red error message in the 'Mensajes' pane:

```
Mens. 229, Nivel 14, Estado 5, Línea 1
Se denegó el permiso SELECT en el objeto 'FABRICANTE',
base de datos 'EMPRESA', esquema 'Compras'.
```

Esquemas y permisos

12.6. Denegar permisos:

Vamos a crear un rol para cada uno de los esquemas de las tablas y le asignaremos permisos de SELECT, INSERT, UPDATE y DELETE. Los roles se llamarán:

- **rol_rrhh**
- **rol_compras**
- **rol_ventas**

Siendo **sa** creamos los roles del siguiente modo:

```
use EMPRESA;
GO
--rol_rrhh
create role rol_rrhh;
grant SELECT, INSERT, UPDATE, DELETE
on schema::rrhh
to rol_rrhh;
--rol_compras
create role rol_compras;
grant SELECT, INSERT, UPDATE, DELETE
on schema::compras
to rol_compras;
--rol_ventas
create role rol_ventas;
grant SELECT, INSERT, UPDATE, DELETE
on schema::ventas
to rol_ventas;
```

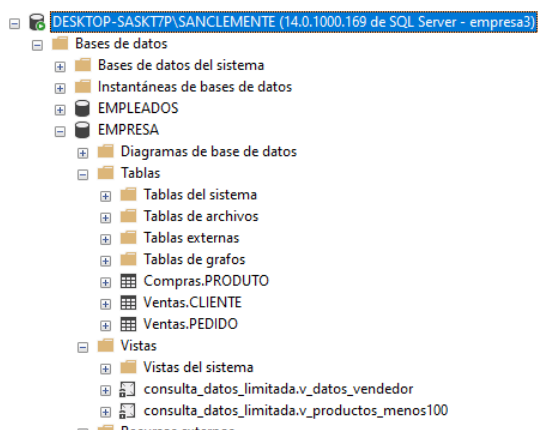
El usuario **empresa3** trabaja en el departamento de ventas y además es la persona encargada de controlar el stock del almacén, por lo que tiene que poder consultar las existencias de los productos. Siendo **sa** vamos a asignar al usuario **empresa3** al rol **rol_ventas**.

```
exec sp_addrolemember rol_ventas, empresa3;
```

Para que pueda ver las existencias de los productos, siendo **sa** vamos a concederle el permiso de SELECT sobre todas las columnas de la tabla *Compras.Produto*, menos el precio. Además también va a poder modificar la columna de existencias.

```
grant select(cod_fabricante, identificador, descripcion, existencias),
update(existencias)
on Compras.PRODUTO
to empresa3;
```

Nos conectamos como **empresa3** y comprobamos qué puede hacer. Desde el explorador de objetos comprobamos las tablas puede ver. También puede ver las vistas por pertenecer al rol **public**:



Esquemas y permisos

```
SQLQuery1.sql - DE...SA (empresa3 (55)) * x UD11_PracticaGuiad...EMPRESA (sa (52)) *
use EMPRESA;
select * from Compras.Producto; --Da error por la columna precio
select cod_fabricante, identificador, descripcion, existencias
from Compras.Producto; --OK
```

100 %

Resultados

Mens. 230, Nivel 14, Estado 1, Líneas 2
Se denegó el permiso SELECT en la columna 'precio' del objeto 'PRODUCTO', base de datos 'EMPRESA', esquema 'Compras'.

	cod_fabricante	identificador	descripcion	existencias
ASU	AK47A	Portátil convertible 2 en 1 ASUS 10,1" T100TA-DK048H Intel Quad Core Atom Bay Trail-T Z3775		10
ASU	AK48A	Tablet 8" ME581C-1B007A Wi-Fi 16 GB		11
ASU	XK48A	Tarxeta gráfica SVGA Asus NVIDIA GeForce 210 Silent DI/1GD3/V2 (LP)		20
KIN	11003	HD SSD 120GB 2.5 SATA3 v300		150
KIN	11089	DDR3 4GB PC1600 CL11 DIMM. SRX8		20
KIN	11672	DDR3 SO DIMM 4GB PC1333 CL9 SR		0
LOG	11002	mk270 combo teclado con rato óptico		5
LOG	11003	rato óptico logitech b100 negro		20
LOG	11004	HD Webcam C270		12
LOG	1100X	USB Headset H540		37
LOG	1100Y	3D PRO Joystick		30
LOG	1100Z	Bluetooth Audio Adapter		7
SAM	9A44G	Tarxeta de memoria SD PRO Clase 10 UHS-I de 16 GB		140
SAM	9A44L	Multifunción Láser Xpress C460W Wi-Fi		120
SAM	9A44R	Cable USB a micro USB		120
SEA	9A45C	Funda Book Cover para Samsung Galaxy Tab S 8,4" marrón		30
SEA	11200	HDD 1TB 7200rpm 64MB SATA3 6gbps		2
SEA	11400	HD 2.5 500GB 8MB 5400rpm SATA2		15
TOS	178CO	Pen Drive Daichi 32 GB 3.0 azul		28
TOS	178CO	Portátil Satellite Click 2 Pro 13,3" P30W-B-108 Intel Core i5 4210U		50
TOS	178CO	Disco Duro portátil Toshiba 2 TB USB 3.0		90
TOS	278HA	Rato Toshiba W30 Óptico sen fios negro		22
TOS	287PA	Disco Duro interno Toshiba MQ Series 1TB 2,5" SATA		24
TOS	287XA	Portátil 13,3" Satellite Z30-A-1DG Intel Core i5 4210U		320

(24 filas afectadas)

Consultamos *ventas.cliente*. Tiene que poder consultarla porque pertenece al rol **rol_ventas**.

```
--Consultamos los datos de una tabla del esquema ventas.
--El permiso lo tiene por pertenecer al rol_ventas
select * from ventas.cliente where limite_de_credito>30000;
```

100 %

Resultados

numero	nome	num_empleado_asignado	limite_de_credito
1101	PC CAIXA, SL	106	65000.50
1102	APPS INFOR, SL	101	65000.90
1103	PC MAX	105	50000.00
1105	INFORMÁTICA SANTI	101	45000.00
1106	INFOR MAX	102	65000.00
1107	O TEU PC	110	35000.00
1108	PIP INFORMÁTICA	109	55000.00
1111	PC OK	103	50000.00
1112	HW & SW OK	108	50000.00
1117	TODO INFOR	106	35000.00
1118	TODO HW	108	60000.00
1120	PC POR PEZAS, SL	102	50000.00
1121	MERCA PC	103	45000.00
1123	MERCADO PC	102	40000.00
1124	INFOR REPARACIONS	NULL	40000.00

(15 filas afectadas)

REVOKE: Siendo **sa** vamos a quitarle a **empresa3** el permiso de consulta sobre *ventas.cliente*.

```
revoke select
on ventas.cliente
from empresa3;
```

Aparentemente podemos pensar que al quitarle el permiso ya no podrá consultar la tabla:

```
select *
from ventas.cliente
where num_empleado_asignado is null;
```

133 %

Resultados Mensajes

	numero	nome	num_empleado_asignado	limite_de_credito
1	1124	INFOR REPARACIONS	NULL	40000.00

Como vemos **sí puede realizar la consulta** ya que hemos quitado el permiso de manera explícita con **REVOKE** pero sigue estando vigente porque lo sigue heredando del rol **rol_ventas**.

Esquemas y permisos

Por entorno gráfico, vamos a consultar los permisos de **empresa3** en la tabla *ventas.cliente*:

Permisos explícitos: No tiene permisos que se hayan concedido directamente.

Esquema	Nombre	Tipo
Compras	PRODUTO	Tabla
Ventas	CLIENTE	Tabla

Permiso	Otorgante de permisos	Conceder	WITH GRA...	Denegar
Actualizar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asumir propiedad		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Control		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eliminar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insertar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modificar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Referencias		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seleccionar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ver definición		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ver seguimiento de ca...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Permisos vigentes: Aquí aparecen todos los permisos que tiene el usuario para ese objeto, tanto recibidos directamente como heredados de algún rol. **Como vemos sigue teniendo el permiso de SELECT.**

Esquema	Nombre	Tipo
Compras	PRODUTO	Tabla
Ventas	CLIENTE	Tabla

Permiso	Columna
DELETE	
INSERT	
SELECT	
SELECT	limite_de_credito
SELECT	nome
SELECT	num_empleado_asignado
SELECT	numero
UPDATE	
UPDATE	limite_de_credito
UPDATE	nome
UPDATE	num_empleado_asignado
UPDATE	numero

DENY: Siendo **sa** vamos a DENEGARLE a **empresa3** el permiso de select sobre *ventas.cliente*.

```
deny select
on ventas.cliente
to empresa3; --Con DENY debemos poner TO
              --Con REVOKE podemos usar TO/FROM
```

Al quitarle el permiso con DENY ya no podrá consultar la tabla, aunque herede el permiso del rol. Lo comprobamos lanzando la misma consulta de antes para buscar los clientes sin empleado asignado:

```
SQLQuery1.sql - DE...SA (empresa3 (55))* x UD11_PracticaGuiad...EMPRESA (sa (52))*
select *
from ventas.cliente
where num_empleado_asignado is null;

Mensajes
Mens. 229, Nivel 14, Estado 5, Línea 9
Se denegó el permiso SELECT en el objeto 'CLIENTE',
base de datos 'EMPRESA', esquema 'Ventas'.
```


Esquemas y permisos

Por entorno gráfico, vamos a consultar los permisos que tiene después del DENY **empresa3** en la tabla **ventas.cliente**:

Permisos explícitos: Aparece que SELECT ha sido denegado.

empresa3

Script Ayuda

Nombre de usuario: empresa3

Elementos protegibles:

Esquema	Nombre	Tipo
Compras	PRODUCTO	Tabla
Ventas	CLIENTE	Tabla

Permisos para Ventas.CLIENTE:

Permiso	Otorgante de permisos	Conceder	WITH GRA...	Denegar
Insertar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modificar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Referencias		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Seleccionar		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ver definición		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ver seguimiento de ca...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Permisos vigentes: Vemos como ya no aparece el SELECT en la lista de permisos.

empresa3

Script Ayuda

Nombre de usuario: empresa3

Elementos protegibles:

Esquema	Nombre	Tipo
Compras	PRODUCTO	Tabla
Ventas	CLIENTE	Tabla

Permisos para Ventas.CLIENTE:

Permiso	Columna
DELETE	
INSERT	
UPDATE	
UPDATE	limite_de_credito
UPDATE	nome
UPDATE	num_empleado_asignado
UPDATE	numero

12.7. Cuándo usar REVOKE y cuándo DENY:

Usaremos **REVOKE** cuando:

- Queremos limpiar permisos explícitos que ya no son necesarios.
- Estamos ajustando los permisos y queremos permitir que los permisos heredados sigan aplicándose.
- Queremos quitar permisos explícitos para administrar permisos más centralizados a través de roles.

Usaremos **DENY** cuando:

- Necesitamos asegurarnos de que un usuario no tenga un permiso específico bajo ninguna circunstancia.
- Queremos implementar una política de seguridad estricta donde ciertos usuarios no deben tener acceso a ciertos objetos, independientemente de los roles.

13. Permisos de instrucción:

Hasta ahora hemos concedido, revocado o denegado permisos de objeto, es decir sobre una tabla, vista, esquema... Ahora vamos a gestionar permisos de instrucción.

Los permisos de instrucción no están vinculados a objetos como tablas o vistas, sino a acciones generales.

Algunos permisos de instrucción son:

- **CREATE DATABASE**
- **BACKUP DATABASE**
- **CREATE TABLE**
- **CREATE VIEW**
- **CREATE ROLE**
- **ALTER ANY LOGIN**
- **SELECT, INSERT, UPDATE, DELETE, EXECUTE...**

13.1. Creamos un usuario en la BD SOCIOS:

En la BD SOCIOS vamos a crear un usuario **socios1** y nos conectamos como **socios1** al servidor:

```
--socios1
create login socios1
with password = 'socios1',
    default_database=SOCIOS;
USE SOCIOS;
create user socios1;
```

El usuario no puede hacer ni un SELECT sobre una de las tablas de la BD:

```
use SOCIOS;
select * from SOCIO;
```

133 %

Mensajes

Mens. 229, Nivel 14, Estado 5, Línea 2
Se denegó el permiso SELECT en el objeto 'SOCIO',
base de datos 'SOCIOS', esquema 'dbo'.

13.2. Concedemos el permiso de consulta de cualquier objeto en la BD SOCIOS:

Siendo **sa** vamos a conceder el permiso de SELECT pero esta vez no vamos a indicar ningún objeto:

```
use SOCIOS;
grant select
to socios1;
```

En este caso no necesitamos la cláusula ON, ya que no tenemos que indicar ningún objeto.

Esquemas y permisos

Comprobamos como ahora el usuario **socios1** puede hacer cualquier SELECT en la BD SOCIOS.

SQLQuery1.sql - DE...CIOS (socios1 (56))* - X UD11_PracticaGuiad...E.SOCIOS (sa (52))*

```
use SOCIOS;
select * from ACTIVIDADE;
select * from AULA;
```

133 %

Resultados Mensajes

identificador	nome	data_ini	data_fin	num_prazas	prezo	observacions	num_profesorado_imparte	num_aula
1	10	TENIS PARA PRINCIPIANTES	2014-02-10	2014-10-10	15	301.55	Precisase raqueta e 1 bote de pelotas	100
2	20	REPOSTARIA	2015-02-15	2015-02-15	60	46.50	NULL	200
3	30	XADREZ	2014-03-20	2014-05-20	30	74.40	NULL	100
4	40	INICIACIÓN Á INFORMÁTICA	2015-03-01	2015-04-01	60	0.00	NULL	300

numero	descripcion	superficie	estado
1	1	PISTA DE TENIS	270 B
2	2	COCINA	100 R
3	3	AULA TALLER	150 B
4	4	AULA SUR	80 M
5	5	AULA NORTE	50 B

13.3. Concedemos el permiso de crear bases de datos:

Vamos a comprobar si el usuario **socios1** puede crear una base de datos nueva. No es administrador, ni pertenece a un rol con ese permiso ni tampoco se le ha concedido explícitamente, así que no debería poder:

SQLQuery1.sql - DE...CIOS (socios1 (56))* - X UD11_PracticaGuiad...E.S

```
create database pruebas_permisos;
```

133 %

Mensajes

Mens. 262, Nivel 14, Estado 1, Línea 5
Se ha denegado el permiso CREATE DATABASE en la base de datos 'master'.

Efectivamente comprobamos como siendo **socios1** no podemos crear una BD.

Vamos a concederle el permiso para que sí pueda crear bases de datos. Siendo **sa** ejecutamos las siguientes instrucciones:

```
use master;--Para este tipo de permisos
--debemos tener seleccionada la BD master
--sino dará error
create user socios1
for login socios1;--Debemos tener creado el usuario en master
--sino dará error
grant create database
to socios1;
```

Comprobamos ahora como el usuario **socios1** Sí puede crear una base de datos nueva:

SQLQuery1.sql - DE...CIOS (socios1 (56))* - X UD11_PracticaGuiad...TE.maste

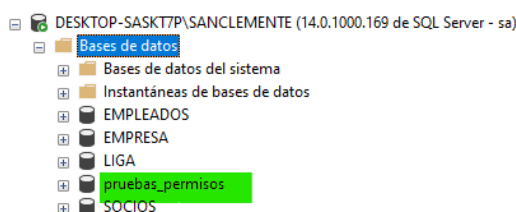
```
create database pruebas_permisos;
```

133 %

Mensajes

Los comandos se han completado correctamente.

Comprobamos en el explorador de objetos que la BD se ha creado:



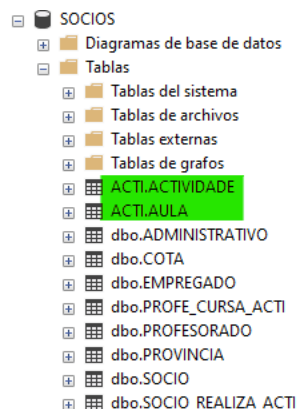
14. Borrado de esquemas:

Para borrar un esquema el esquema no puede contener objetos.

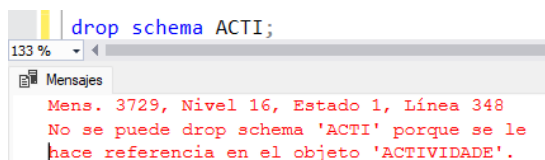
14.1. Borrado de un esquema con objetos:

En la BD SOCIOS creamos el esquema **ACTI** y movemos **AULA** y **ACTIVIDAD** para el esquema nuevo:

```
use SOCIOS;
GO
create schema ACTI;
GO
alter schema ACTI
transfer dbo.ACTIVIDADE;
GO
alter schema ACTI
transfer dbo.AULA;
```

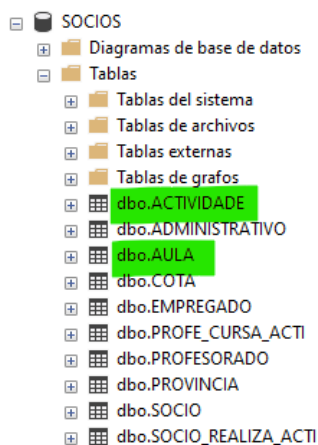


Intentamos borrar el esquema y no lo permite:

**14.2. Borrado de un esquema vacío:**

Vamos a volver a dejar las tablas **AULA** y **ACTIVIDAD** en el esquema **dbo**:

```
use SOCIOS;
GO
alter schema dbo
transfer ACTI.ACTIVIDADE;
GO
alter schema dbo
transfer ACTI.AULA;
```



Intentamos borrar el esquema y ahora Sí lo permite porque el esquema está vacío:

