

Cursos SQL Server 2008 R2

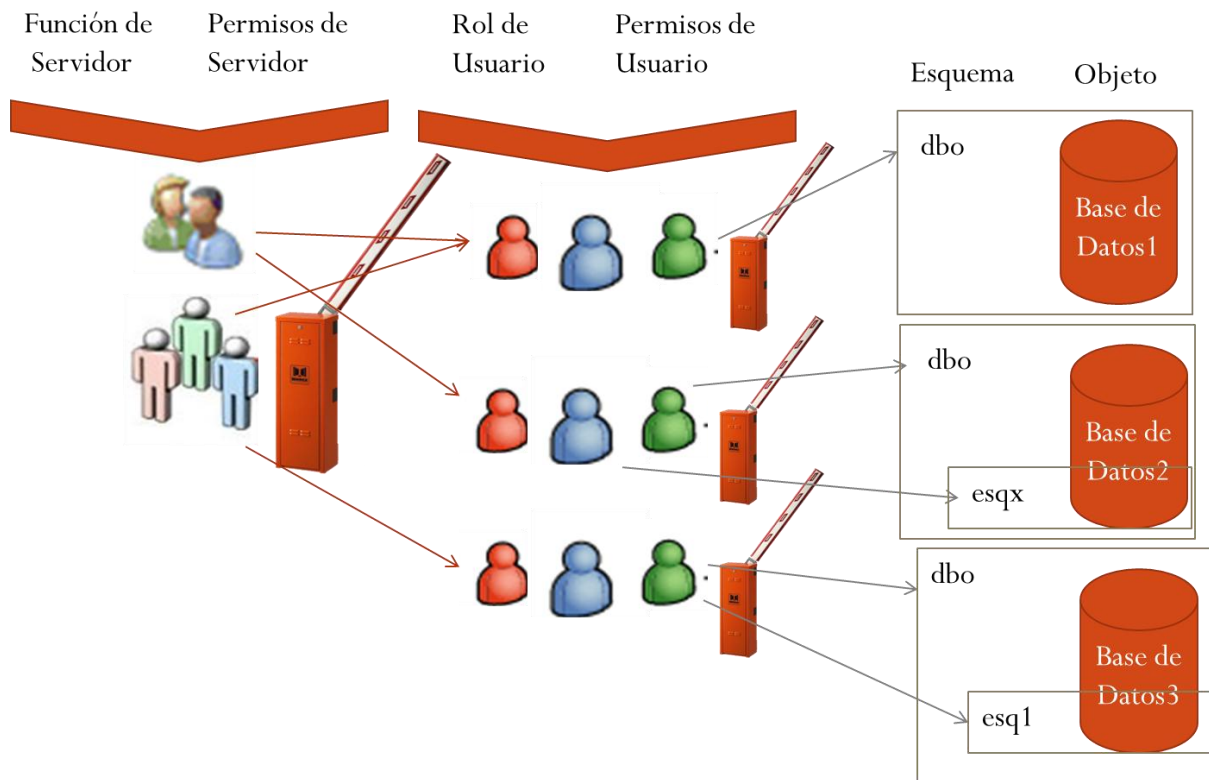
Seguridad - principios

Cursos SQL Server 2008 R2

Índice

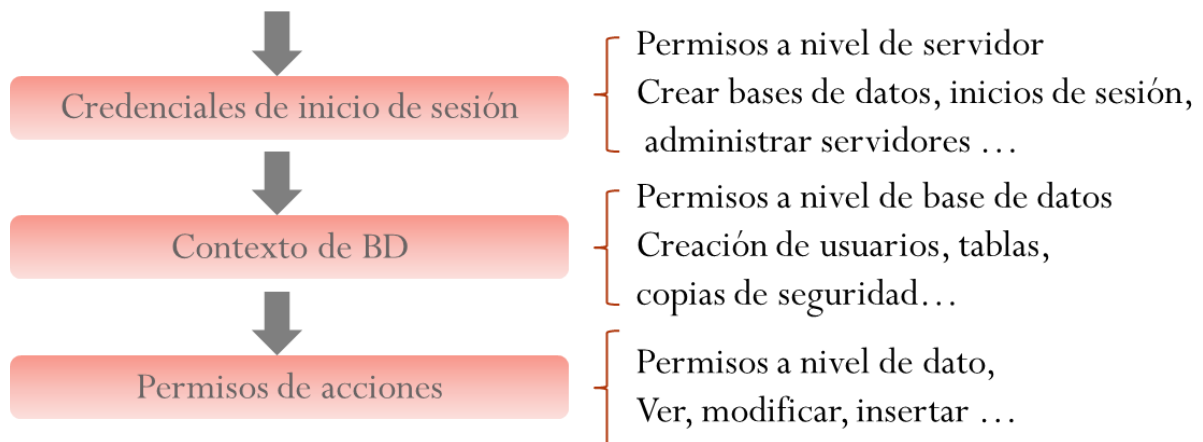
- Introducción
- Autenticación de usuarios
- Arquitectura de permisos
- Roles de servidor y roles de base de datos

DIAGRAMA GENERAL



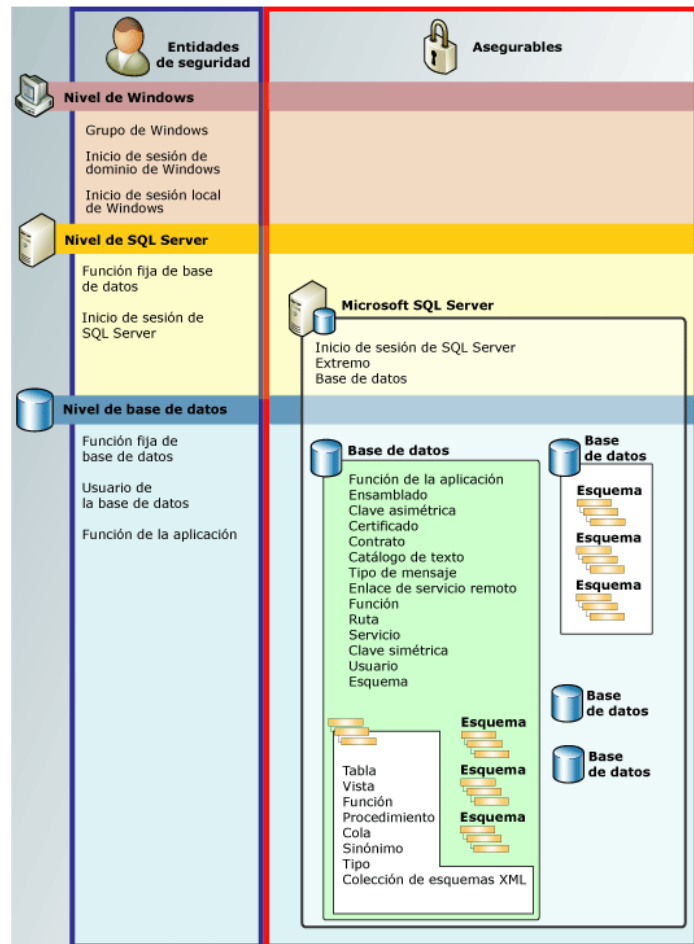
Introducción

El modelo de seguridad de SQL Server esta basado en permisos que se otorgan a los Principales, Individuales, Grupales y los procesos a los que pueden acceder.

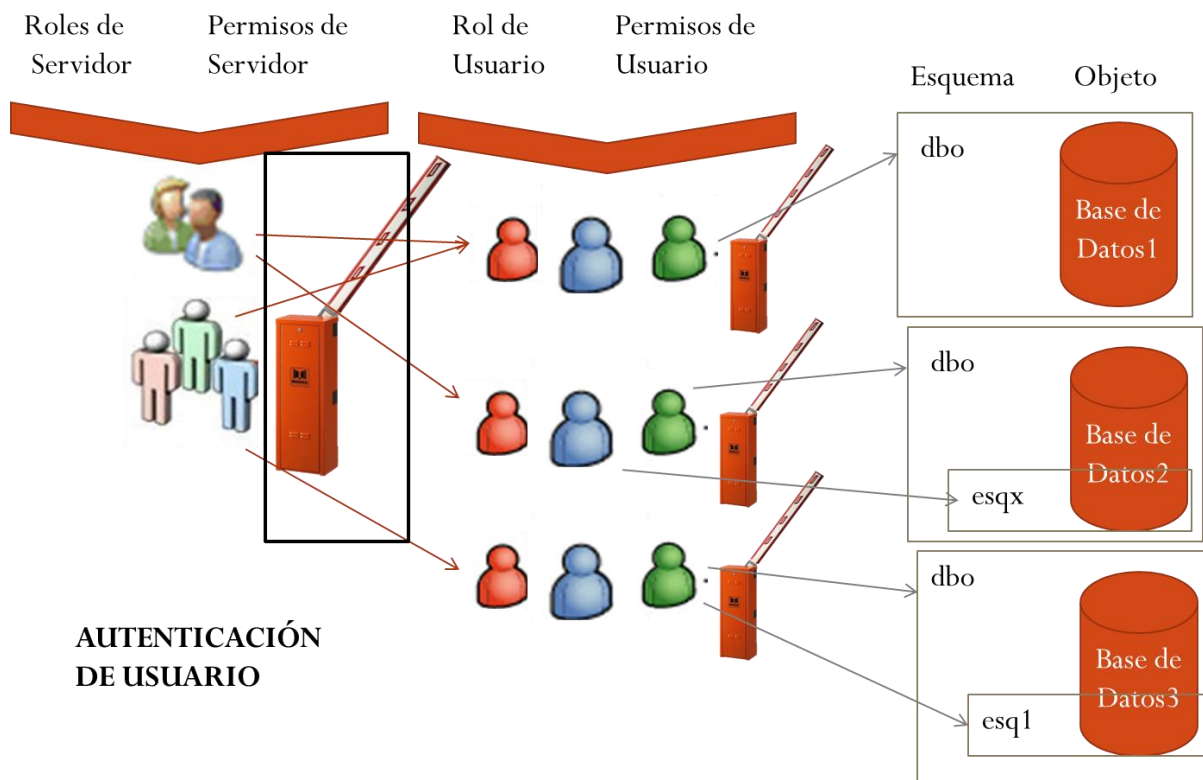


Principales o entidades de seguridad:
 Usuarios de Windows, usuarios de SQL Server, Usuarios de Bases de Datos.

Asegurables: Recursos que pueden ser protegidos.



Autenticación de usuario



SQL Server autentifica los permisos de todas las conexiones, por lo que todas las conexiones deben especificar el modo de autenticación y las credenciales.

Existen dos modos de autenticación:

- Autenticación de Windows
- Autenticación Mixta

Autenticación Windows

Solo los usuarios autenticados de Windows podrán acceder a la instancia de SQL Server por lo que se tendrá que añadir un usuario/grupo de Windows por cada usuario o grupo que queramos que acceda a la instancia de SQL Server.

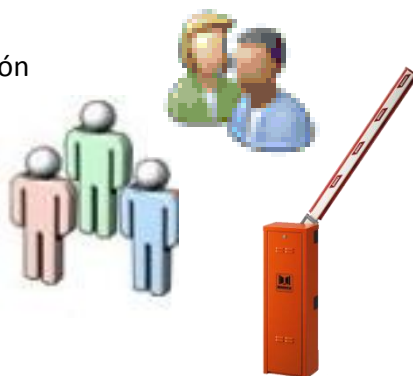
Esta es la opción recomendada y por defecto, ya que se aprovechan las políticas de seguridad del dominio Active Directory.



Autenticación mixta

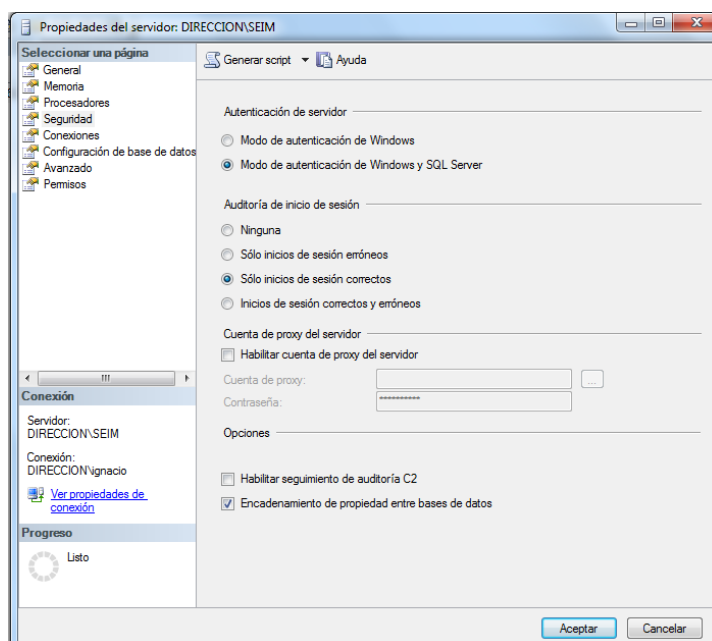
Se permite tanto la autenticación Windows como la autenticación SQL Server.

Se utiliza en los casos donde es necesario dar acceso a usuarios que no son a usuarios de Windows.

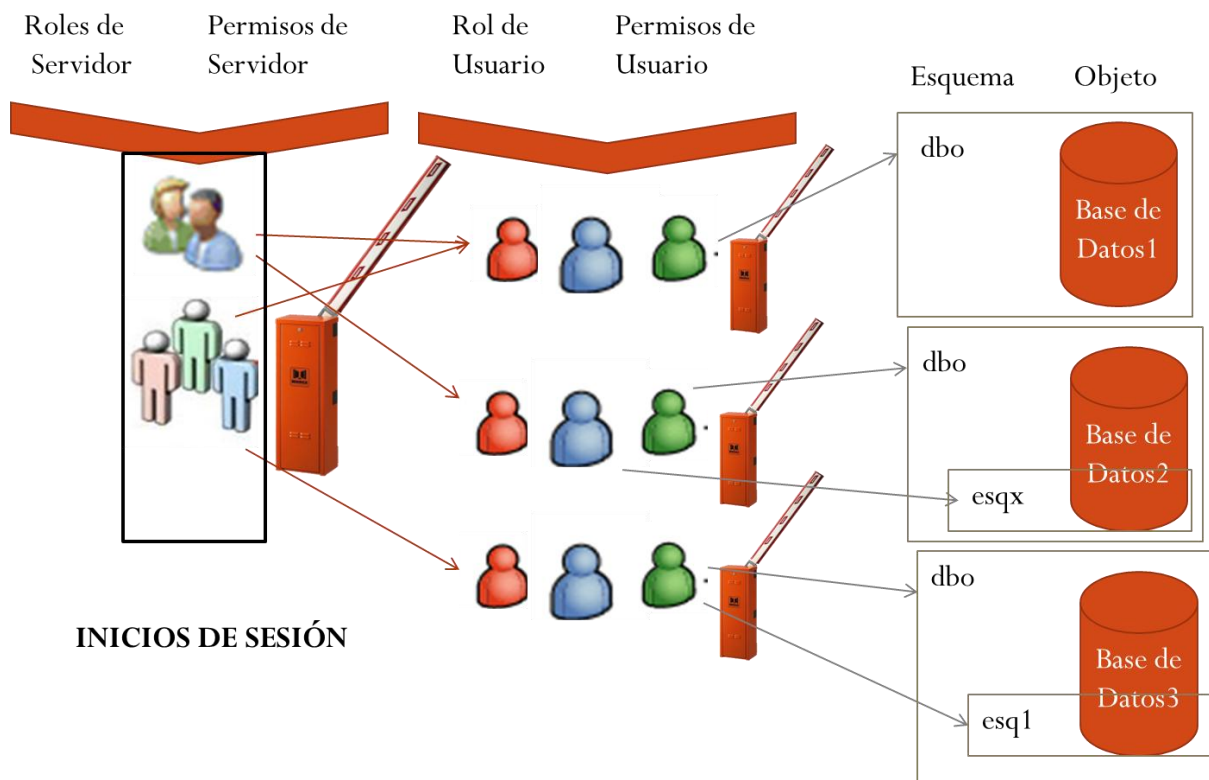


Cambio de modo de autenticación

1. Botón derecho en el servidor y seleccionar Propiedades.
2. Seleccionar Seguridad.
3. Seleccionar el tipo de autenticación que deseamos.
4. Seleccionar Guardar para guardar los cambios.
5. OK en el cuadro de dialogo que advierte que los cambios no serán efectivos mientras no se reinicie.
6. Botón derecho en el servidor y seleccionar reiniciar.



Inicios de sesión



Los inicios de sesión son principales de seguridad que permiten el acceso a SQL Server. Su creación se puede hacer de diferentes maneras.

- De forma gráfica en SSMS
- Usando CREATE LOGIN

Creación de un inicio de sesión de Windows

`CREATE LOGIN [Domain\User] FROM WINDOWS`

Para la creación de un inicio de sesión de SQL Server

`CREATE LOGIN login_name WITH PASSWORD='password'`

Para los inicios de sesión de SQL Server se pueden especificar las siguientes opciones:

- **MUST_CHANGE:** El inicio de sesión debe cambiar el password la próxima vez que inicie sesión.
- **CHECK_EXPIRATION:** SQL Server aplicará la política de expiración de Windows al inicio de sesión de SQL Server.
- **CHECK_POLICY:** SQL Server aplicará la política de passwords de Windows al inicio de sesión que creamos.

Password policies

Para garantizar la seguridad es mejor aplicar las opciones anteriores.



Creación y modificación

En el siguiente ejemplo crearemos un inicio de sesión y lo forzaremos a verificar la expiración y las políticas de contraseñas:

```
CREATE LOGIN Administrador WITH PASSWORD='Administrador', CHECK_EXPIRATION=ON,  
CHECK_POLICY =ON
```

Para cambiar las propiedades de un inicio de sesión usaremos la sentencia *ALTER LOGIN*.

Cambio de contraseñas de un inicio de sesión SQL Server:

```
ALTER LOGIN login_name WITH PASSWORD='password'
```

También se puede inhabilitar un inicio de sesión:

```
ALTER LOGIN login_name DISABLE
```

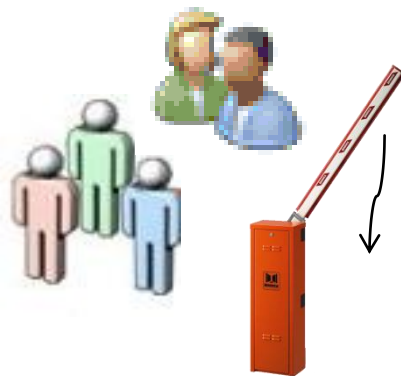
Borrado

Para borrar un inicio de sesión usaremos *DROP LOGIN*

```
DROP LOGIN login_name
```

Para borrar un inicio de sesión de Windows

```
DROP LOGIN [Domain\User]
```



Precaución

No se podrá borrar ningún inicio de sesión vinculado a un usuario que posea cualquier asegurable, objeto de servidor o tarea de SQL Server Agent.

Se debería primero inhabilitar al usuario(s), y una vez seguro de que la acción no afecta a SQL Server, borrarlo.

Si se borra un usuario de base de datos asociado a un inicio de sesión, SQL Server no borrará automáticamente el inicio de sesión, por lo que quedará huérfano.

Se puede borrar un inicio de sesión sin eliminar al usuario asociado a él. El problema es que ya no se podrá acceder al usuario a través del inicio de sesión.

Excepciones

El administrador de bases de datos debe en ocasiones gestionar excepciones al dar acceso a un grupo de Windows. Por ejemplo, de todo un grupo de Windows todos deben tener acceso a SQL Server excepto uno.

Para realizar esta tarea se debe crear un login de Windows para el grupo y después denegar el acceso a ese usuario en particular:

```
CREATE LOGIN [domain_name\group_name] FROM WINDOWS
```

```
DENY CONNECT SQL TO [domain_name\user_name]
```


Inicios de sesión preestablecidos

Al instalarse SQL Server se crean 2 inicios de sesión:

- La cuenta de servicio que se utiliza para iniciar el servicio SQL Server. Pueden cambiarse sus privilegios. Por defecto es la cuenta del Administrador del Sistema, pero se puede modificar.
- El inicio de sesión **sa**. No puede eliminarse ni modificarse. No estará disponible si el motor está configurada con autenticación de Windows.

Pueden realizar cualquier tarea en SQL Server (ambos pertenecen al rol de servidor sysadmin)

Visualización

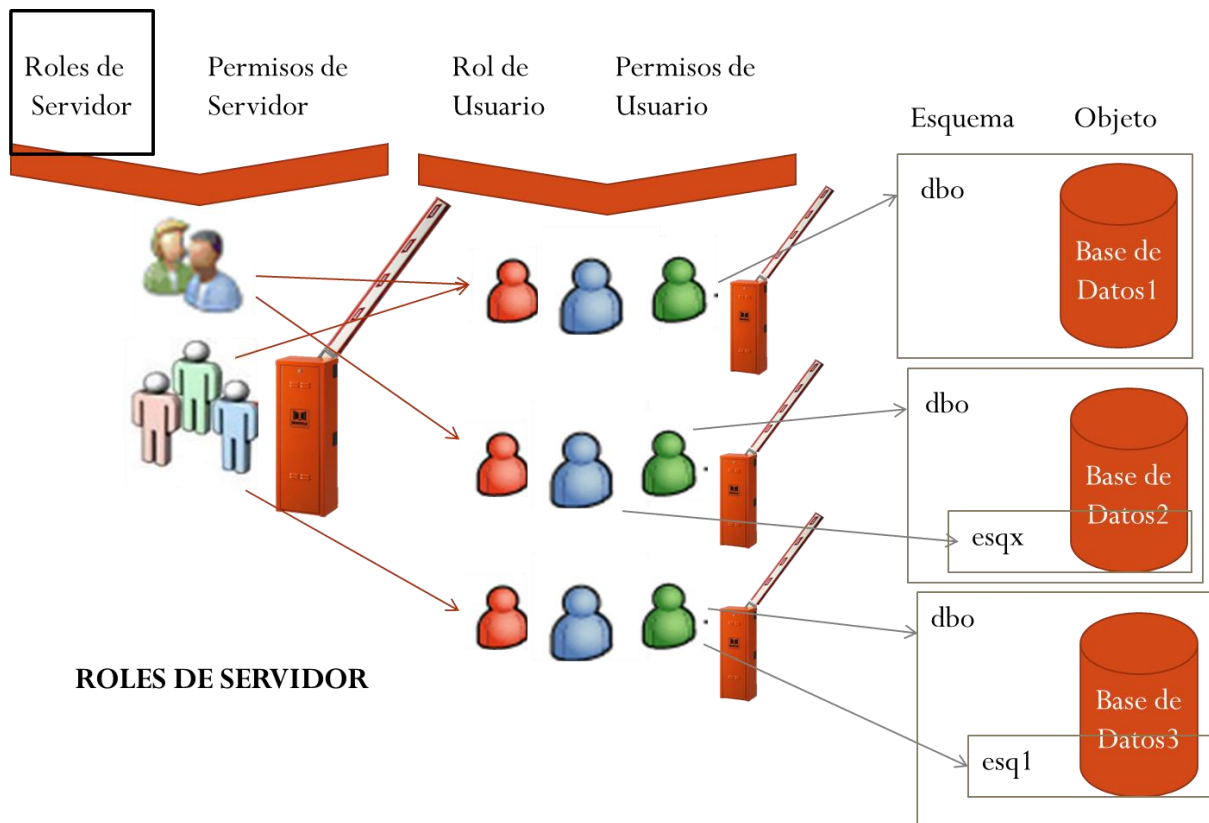
Para ver la información de los inicios de sesión tenemos dos tablas del diccionario:

- `sys.server_principals`: devuelve una fila por cada uno de los inicios de sesión.
- `sys.sql_logins`: Devuelve una fila por cada inicio de sesión con autenticación SQL.

Ejemplo:

2.1.1 Seguridad (Creación Inicios-Usuarios).sql

Roles de Servidor



Cada rol agrupa un conjunto de permisos a nivel de servidor y sirven par facilitar la administración de la seguridad.

Un inicio de sesión puede pertenecer a uno (por defecto y como mínimo **public**) o más roles de servidor y adquiere los permisos (o no permisos) de este.

Características.

- Son fijos
- No pueden modificarse sus permisos
- No pueden eliminarse
- No pueden añadirse nuevos roles de servidor
- Son independientes de las bases de datos

Lista de roles

- **Bulkadmin:** Diseñado para cuentas que tienen que hacer inserciones masivas en la base de datos.
- **Dbcreator:** Esta diseñado para las cuentas que deben crear, modificar, eliminar y restaurar bases de datos.
- **Diskadmin:** Administración de archivos de disco.
- **Processadmin:** Administra procesos de SQL Server. Pueden detener procesos.
- **Securityadmin:** Usuarios que tienen que administrar accesos, crear permisos y leer registros de errores. Puede denegar, otorgar y revocar accesos a nivel de servidor y de bases de datos, resetear contraseñas y acceder a registros de errores.
- **Serveradmin:** Diseñado para usuarios que tienen que manejar opciones de configuración a nivel de servidor. Así como apagarlo.
- **Setupadmin:** Diseñado para aquellos usuarios que tienen que administrar servidores vinculados, así como controlar procedimientos de inicio.
- **Sysadmin:** comprende a todos los roles anteriores.
- **Public:** es el rol asignado por defecto a una cuenta de inicio de sesión.

Mantenimiento de roles de servidor

Si queremos obtener información de los inicios de sesión de cada rol de servidor usaremos la vista de catálogo

`sys.server_role_members`

Para dar a un inicio de sesión un rol de servidor.

`EXECUTE sp_addsrvrolemember login_name, fixed_server_role`

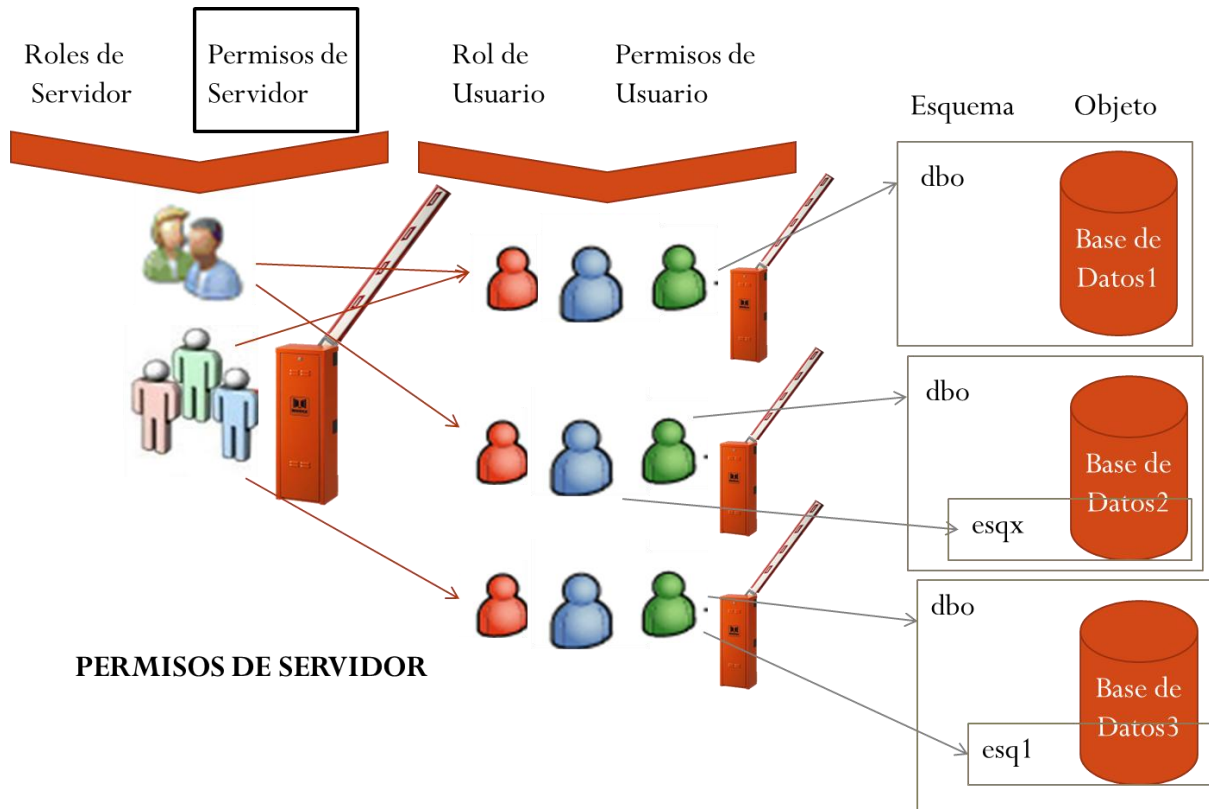
Para quitar al inicio de sesión del rol del servidor.

`sp_dropsrvrolemember login_name, fixed_server_role`

Ejemplo:

(2.1.2 Seguridad (Roles Nivel Servidor).sql)

Permisos de servidor



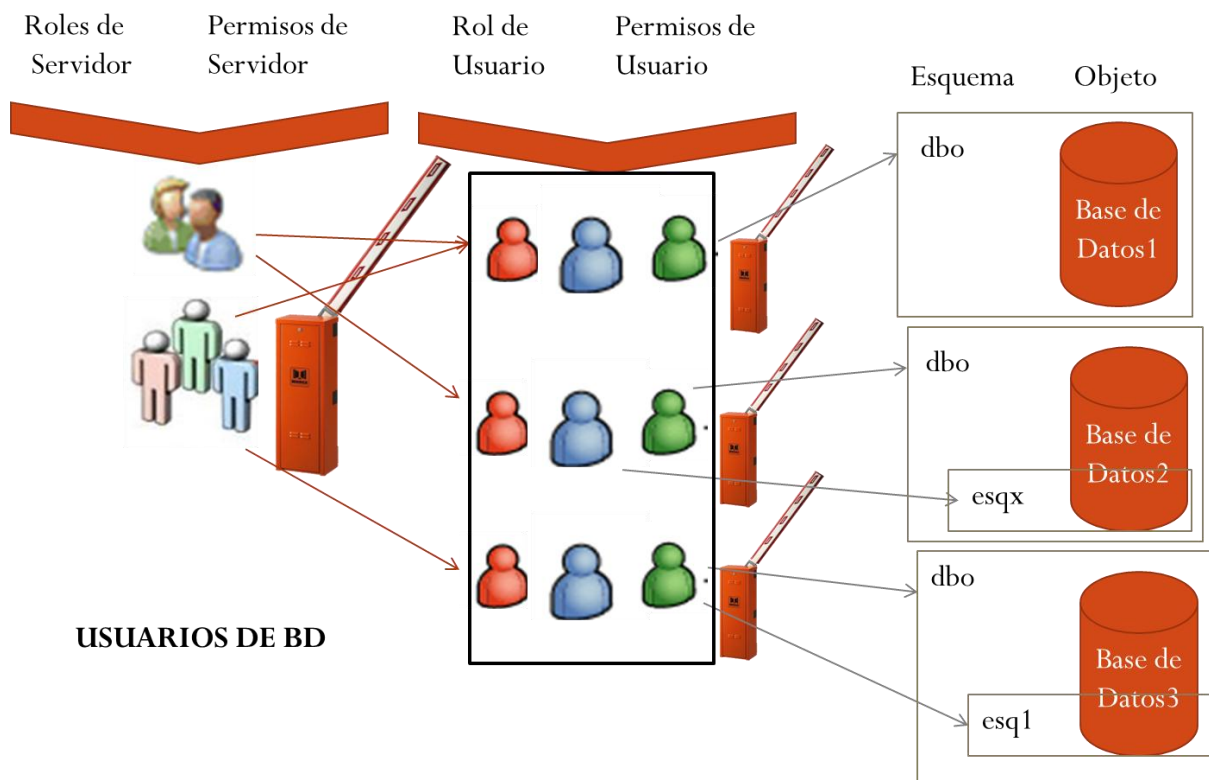
Se aplican a nivel de instancia. Los permisos de servidor sólo pueden concederse si la base de datos actual es la master. Se pueden consultar mediante la vista de catálogo *sys.server_permissions*

Ejemplos:

- Conexión al servidor (instancia)
- Crear o modificar (uso de DDL's)
- Acceso a recursos externos a SQL Server (extremos)

Ejemplo – Práctica

Usuarios de BD



Una vez configurado el modo de autenticación y creados los inicios de sesión, se debe dar a cada uno el acceso apropiado a la base de datos. Lo haremos creando un usuario de base de datos asignado a un inicio de sesión. Puede haber varios usuarios en distintas bases de datos asignados a un mismo inicio de sesión. Los usuarios de base de datos pueden ser asignados a roles de bases de datos.



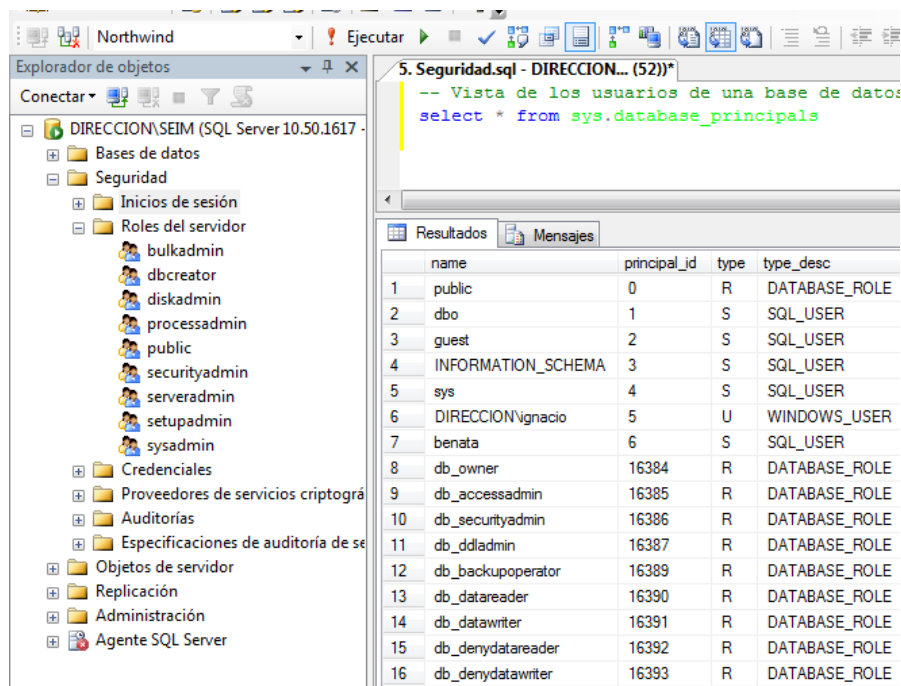
Usuarios por defecto en una Base de Datos

- **dbo:** Propietario por defecto de los objetos que se creen. No puede ser eliminado de la Base de Datos.
- **guest:** Permite el acceso a la base de datos a usuarios con un inicio de sesión que no tiene cuenta en la Base de Datos. Tendrán los permisos dados a guest.
- **information_schema:** Permite ver los metadatos de SQL Server.
- **sys:** Permite consultar las tablas y vistas del sistema, procedimientos extendidos y otros objetos del catálogo del sistema. Es su propietario.

Los tres últimos elementos del apartado anterior son usados por el sistema y están deshabilitados.

Mostrar usuarios de una base de datos

- `Select * from sys.database_principals`



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Roles del servidor' folder is expanded, showing a list of server roles including bulkadmin, dbcreator, diskadmin, processadmin, public, securityadmin, serveradmin, setupadmin, and sysadmin. On the right, a query window titled '5. Seguridad.sql - DIRECCION... (52)*' displays the results of the query `select * from sys.database_principals`. The results are shown in a table with columns: name, principal_id, type, and type_desc. The table lists 16 database principals, including public, dbo, guest, INFORMATION_SCHEMA, sys, DIRECCION\vignacio, benata, db_owner, db_accessadmin, db_securityadmin, db_ddladmin, db_backupoperator, db_datareader, db_datawriter, db_denydatareader, and db_denydatawriter.

name	principal_id	type	type_desc
public	0	R	DATABASE_ROLE
dbo	1	S	SQL_USER
guest	2	S	SQL_USER
INFORMATION_SCHEMA	3	S	SQL_USER
sys	4	S	SQL_USER
DIRECCION\vignacio	5	U	WINDOWS_USER
benata	6	S	SQL_USER
db_owner	16384	R	DATABASE_ROLE
db_accessadmin	16385	R	DATABASE_ROLE
db_securityadmin	16386	R	DATABASE_ROLE
db_ddladmin	16387	R	DATABASE_ROLE
db_backupoperator	16389	R	DATABASE_ROLE
db_datareader	16390	R	DATABASE_ROLE
db_datawriter	16391	R	DATABASE_ROLE
db_denydatareader	16392	R	DATABASE_ROLE
db_denydatawriter	16393	R	DATABASE_ROLE

Permitir que un usuario acceda a la base de datos

Para permitir que un inicio de sesión acceda a la base de datos tenemos que crear un usuario de base de datos para cada inicio de sesión que necesita acceder a la base de datos o asignarle uno existente:

En caso de no especificar el nombre de inicio de sesión, SQL Server tratará de crear un usuario de base de datos asignado al inicio de sesión con el mismo nombre.

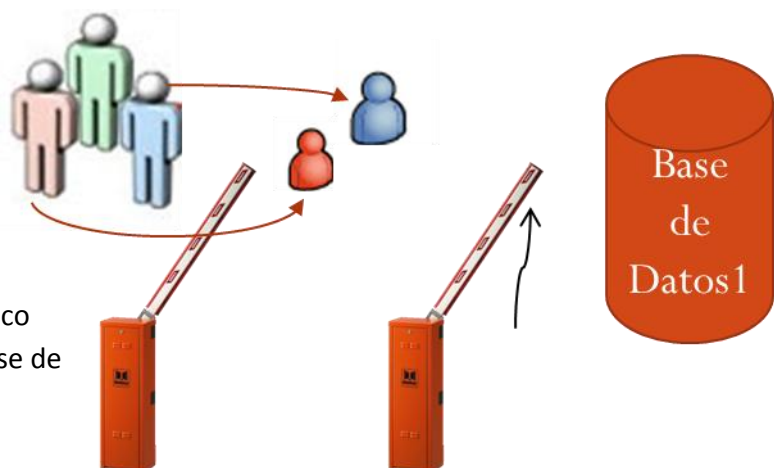
Para modificar un usuario de BD

`ALTER USER`

Para eliminar un usuario de BD

`DROP USER`

Es posible también administrar los usuarios de BD con el entorno grafico en la carpeta seguridad de cada base de datos.



Procedimientos almacenados de sistema

- Para conceder permisos

sp_grantdbaccess y sp_adduser

- Para quitar el acceso

sp_revokedbaccess

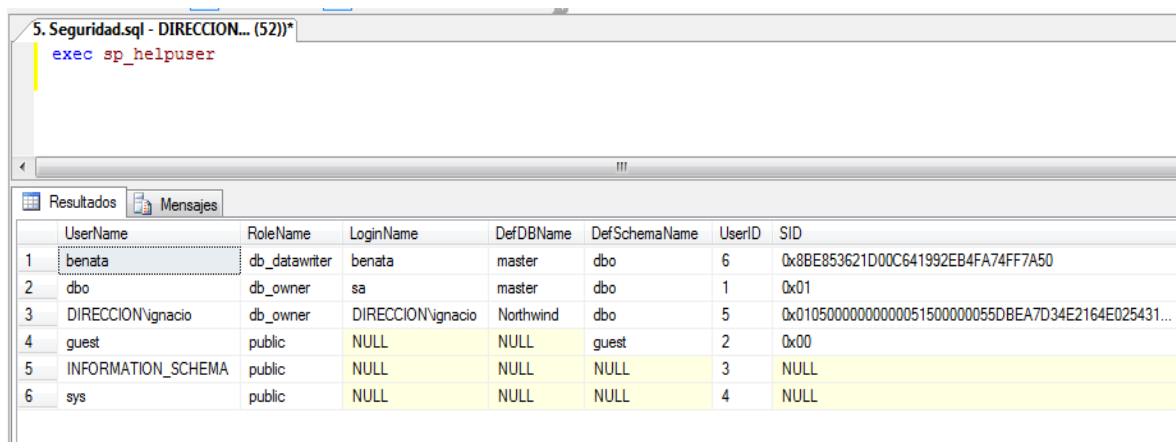
- Para reasignar inicios de sesión con usuarios

sp_change_users_login

- Ver información de los usuarios

sp_helpuser

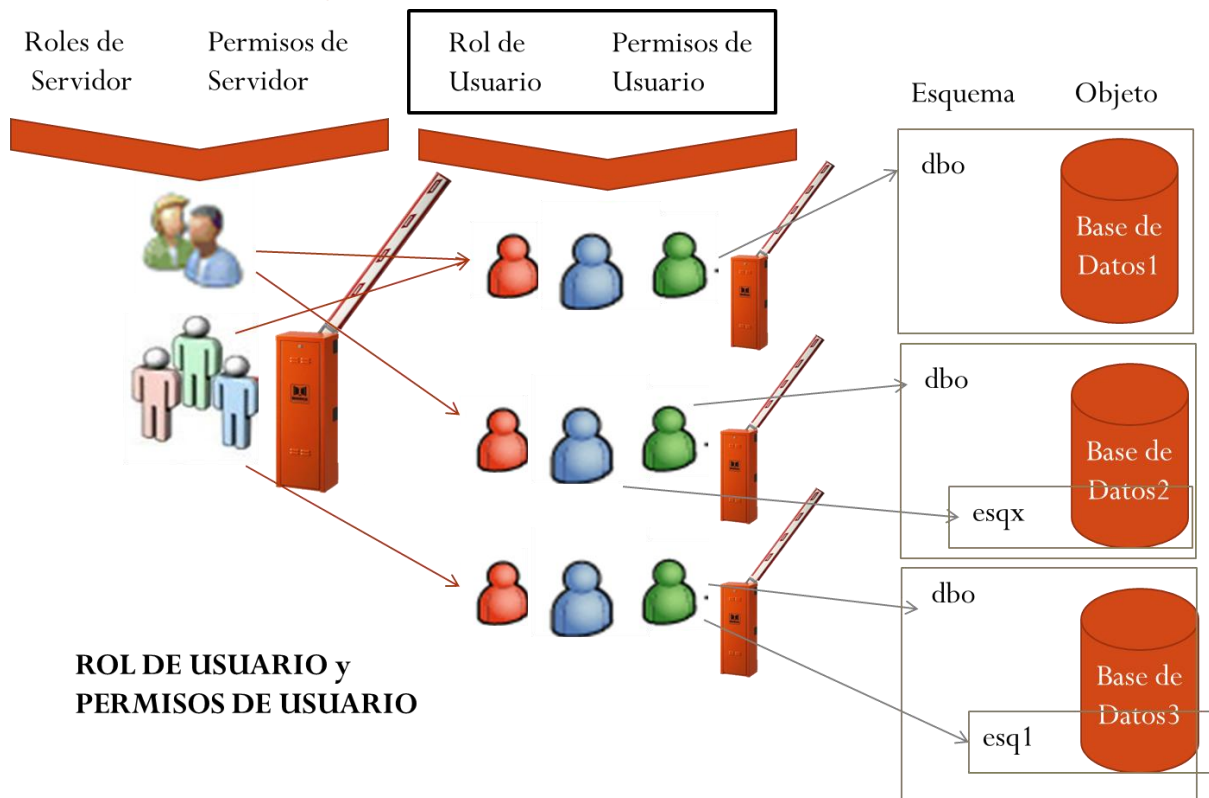
Ver información de los usuarios



The screenshot shows a SQL Server Enterprise Manager window with the title '5. Seguridad.sql - DIRECCION... (52))*'. The query editor contains the command 'exec sp_helpuser'. Below the editor, the 'Resultados' (Results) tab is active, displaying a table with user information. The table has columns: UserName, RoleName, LoginName, DefDBName, DefSchemaName, UserID, and SID. The results are as follows:

	UserName	RoleName	LoginName	DefDBName	DefSchemaName	UserID	SID
1	benata	db_datawriter	benata	master	dbo	6	0x8BE853621D00C641992EB4FA74FF7A50
2	dbo	db_owner	sa	master	dbo	1	0x01
3	DIRECCION\vgnacio	db_owner	DIRECCION\vgnacio	Northwind	dbo	5	0x01050000000000051500000055DBEA7D34E2164E025431...
4	guest	public	NULL	NULL	guest	2	0x00
5	INFORMATION_SCHEMA	public	NULL	NULL	NULL	3	NULL
6	sys	public	NULL	NULL	NULL	4	NULL

Roles de Usuario y Permisos de usuario



Gestión de Roles de bases de datos

En caso de tener muchos usuarios la gestión de los mismos puede ser muy laboriosa. Para ayudar en esta tarea, existen unos roles de usuario para agrupar usuarios de bases de datos.

Fixed Database Role	Database-Level Permission
<i>db_accessadmin</i>	Granted: <i>ALTER ANY USER, CREATE SCHEMA</i>
<i>db_accessadmin</i>	Granted with <i>GRANT</i> option: <i>CONNECT</i>
<i>db_backupoperator</i>	Granted: <i>BACKUP DATABASE, BACKUP LOG, CHECK-POINT</i>
<i>db_datareader</i>	Granted: <i>SELECT</i>
<i>db_datawriter</i>	Granted: <i>DELETE, INSERT, UPDATE</i>

Fixed Database Role	Database-Level Permission
<i>db_ddladmin</i>	Granted: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES
<i>db_denydatareader</i>	Denied: SELECT
<i>db_denydatawriter</i>	Denied: DELETE, INSERT, UPDATE
<i>db_owner</i>	Granted with GRANT option: CONTROL
<i>db_securityadmin</i>	Granted: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION

También es posible crear un rol de base de datos personalizado y agrupar a los usuarios de base de datos en ese rol. Estos roles:

- Agrupan un conjunto de permisos
- No tienen permisos predefinidos

Los permisos de estos roles se establecen por:

- Pertenencia a otros roles
- Concesión de Permisos de sentencias
- Concesión de Permisos específicos de objetos

Los pueden gestionar: sysadmin, db_owner y db_securityadmin

Ejemplo

Se puede agrupar a los usuarios del departamento de contabilidad en un rol de base de datos. De esta forma, asignaremos los permisos a ese rol de base de datos siendo aplicados a todos los usuarios de ese rol.

Para crear un rol:

```
CREATE ROLE role_name
```

Para modificar su nombre:

```
ALTER ROLE statement
```

Para eliminarlo:

```
DROP ROLE
```

Se pueden gestionar los roles utilizando SSMS en el apartado seguridad de cada base de datos.

Asignación de un rol de base de datos a un usuario de la base de datos. Tenemos dos opciones:

- EXECUTE *sp_addrolemember* role_name, user_name
- Vía SSMS modificando las propiedades del usuario.

Es posible agrupar roles. Por ejemplo, si queremos agrupar a los gestores del departamento de contabilidad en un rol llamado AccountingMgr. Es posible unir los roles Accounting y AccountingMgr (que solo tendrá los permisos extra).

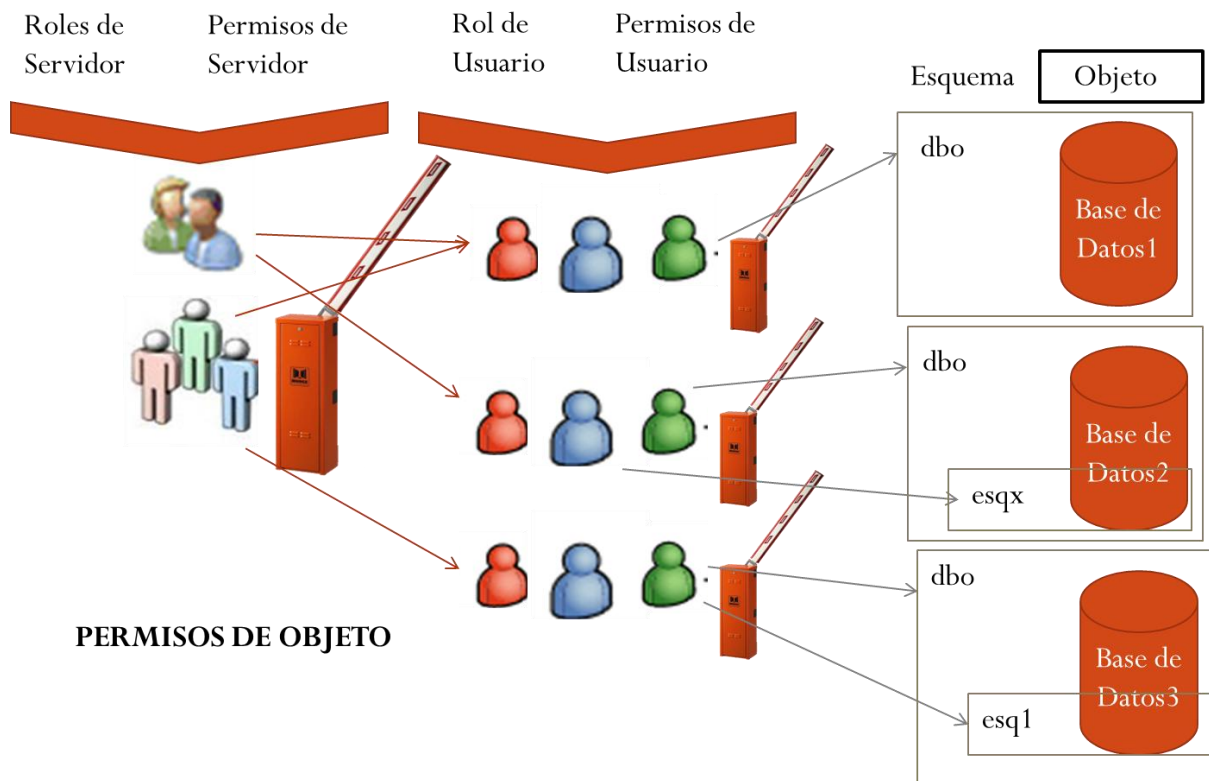
Obtención de información de usuarios de un rol. Podemos consultar la tabla de sistema:

sys.database_role_members

Ejemplo

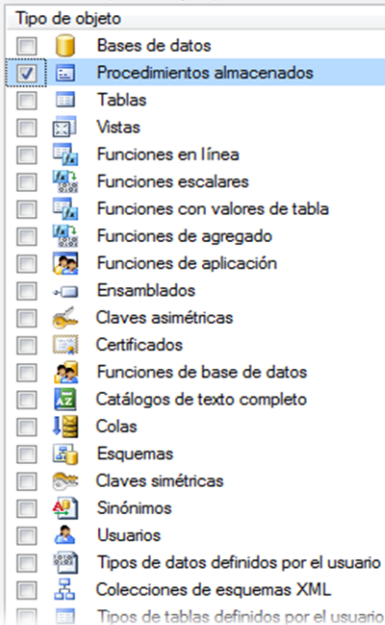
(2.1.3 Seguridad (Consultas Roles y Usuarios).sql)

Permisos de objeto



Permisos de objeto

Seleccionar tipos de objeto para encontrar:



Son permisos específicos de cada objeto que se asignan a cada usuario o rol. Los permisos se asignan permitiendo o denegando las acciones que se pueden realizar sobre cada uno de los objetos.

Los permisos se pueden:

- Denegar
- Revocar
- Conceder

WITH GRANT: Indica que el receptor también podrá conceder el permiso especificado a otras entidades de seguridad.

Revocación y denegación

La diferencia se encuentra en la herencia de permisos

La denegación de un permiso garantiza que aunque dicho permiso sea otorgado por otra vía, no se lleve a cabo el acceso.

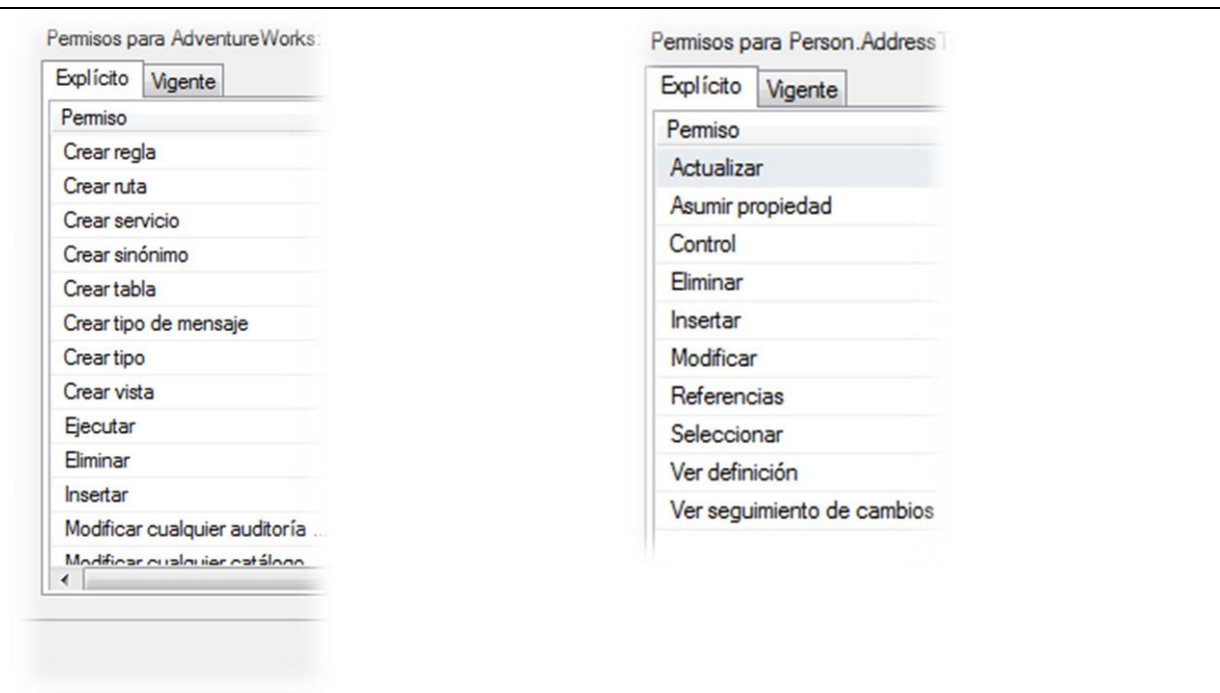
Denegación + Concesión = Denegación

La revocación no garantiza lo anterior.

Revocación + Concesión = Concesión

Por ejemplo, los permisos que se pueden conceder sobre una *base de datos*.

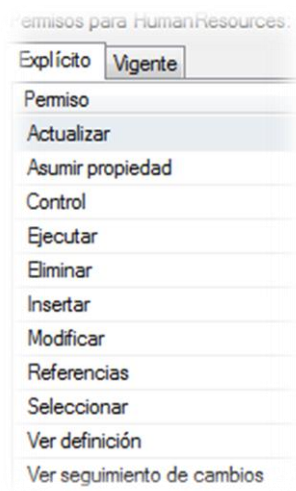
Por ejemplo, los permisos que se pueden conceder sobre una *tabla*.



Permisos de esquema

Son permisos asociados al esquema.

Ejemplo de los permisos que se pueden conceder sobre un Esquema.



Permisos. Ejemplo.

1. Queremos crear un **rol** llamado **Contabilidad** en la base de datos AdventureWorks que solo pueda Insertar (INSERT), Seleccionar (SELECT), Actualizar (UPDATE) y Eliminar (DELETE) datos relacionados con la contabilidad. Las tablas de contabilidad son las que pertenecen a los esquemas Purchasing y Sales.
2. Como AdventureWorks tiene agrupadas las tablas por esquemas podremos crear de manera más fácil el rol **Contabilidad**, permitiendo realizar las acciones sobre los esquemas que contienen las tablas.
3. Creamos un usuario para el inicio de sesión contabilidad y le asignamos el rol. Por lo que todos los usuarios que entran con este inicio de sesión podrán acceder a la base de datos AdventureWorks con las opciones que hemos dado a ese rol.

Usuarios huérfanos

Son usuarios que no tienen un inicio de sesión. Esto puede pasar al eliminar un inicio de sesión y no eliminar su usuario de Base de Datos.

Ejemplo de asignación

UsuarioBD es un usuario huérfano e *Inicio* es una cuenta de inicio de sesión.

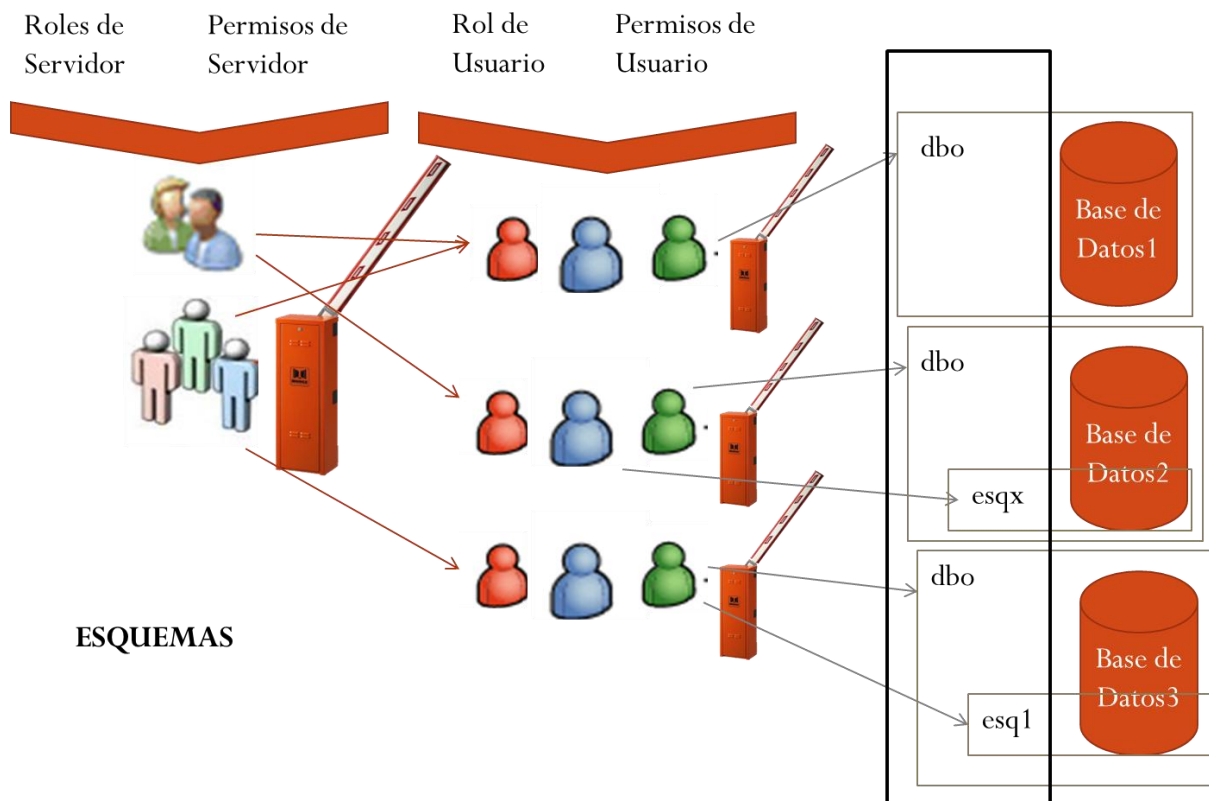
El siguiente mandato vincula al usuario *usuarioBD* con el inicio de Sesión denominado *Inicio*:

```
sp_change_users_login 'Update_On', 'UsuarioBD', 'Inicio'
```

Ejemplo:

2.1.4 Seguridad (Usuarios huérfanos).sql

Esquemas



Un Esquema es un objeto contenedor de otros objetos.

Los objetos contenidos dentro de un esquema pueden ser tablas, procedimientos almacenados, vistas y triggers, conformando el conjunto un único espacio de nombres.

El mayor beneficio de los esquemas es la posibilidad de separar los usuarios de los objetos. Por lo que los cambios de usuario no producen cambios en la aplicación.

Cada esquema pertenece (es propiedad de) a un usuario o rol, por lo que en el caso de necesitar eliminar a un usuario solo se tendrá que transferir la propiedad del esquema a un nuevo usuario o rol.

Sintaxis

Crear un esquema:

```
CREATE SCHEMA schema_name AUTHORIZATION owner
```

Modificar un esquema:

```
ALTER SCHEMA
```

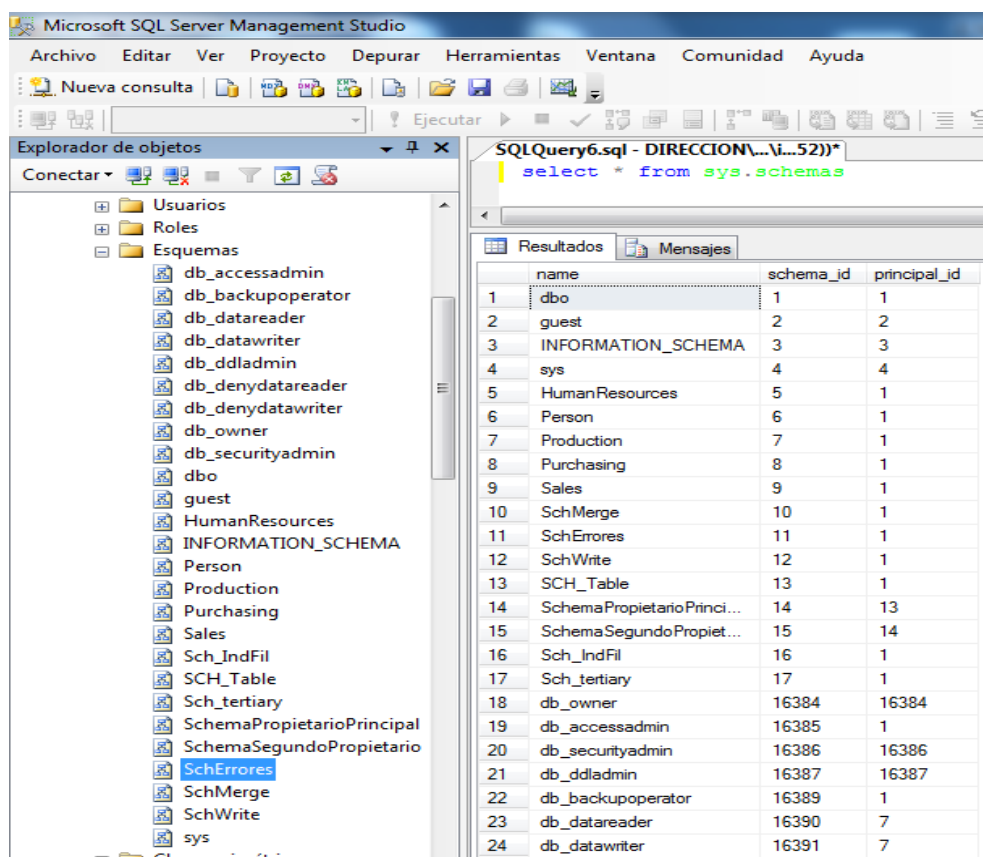
Borrar un esquema:

```
DROP SCHEMA
```

Para recuperar información de un esquema:

- sys.schemas

Esquemas de AdventureWorks



The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the 'Explorador de objetos' (Object Explorer) displays the 'Esquemas' (Schemas) folder under the 'AdventureWorks' database. The list of schemas includes: db_accessadmin, db_backupoperator, db_datareader, db_datawriter, db_ddladmin, db_denydatareader, db_denydatawriter, db_owner, db_securityadmin, dbo, guest, HumanResources, INFORMATION_SCHEMA, Person, Production, Purchasing, Sales, Sch_IndFil, SCH_Table, Sch_tertiary, SchemaPropietarioPrincipal, SchemaSegundoPropietario, SchErrores, SchMerge, SchWrite, and sys. The 'SchErrores' schema is highlighted. On the right, the 'SQLQuery6.sql' window shows the query 'select * from sys.schemas'. The 'Resultados' (Results) pane displays the output of the query as a table with three columns: 'name', 'schema_id', and 'principal_id'.

	name	schema_id	principal_id
1	dbo	1	1
2	guest	2	2
3	INFORMATION_SCHEMA	3	3
4	sys	4	4
5	HumanResources	5	1
6	Person	6	1
7	Production	7	1
8	Purchasing	8	1
9	Sales	9	1
10	SchMerge	10	1
11	SchErrores	11	1
12	SchWrite	12	1
13	SCH_Table	13	1
14	SchemaPropietarioPrinci...	14	13
15	SchemaSegundoPropiet...	15	14
16	Sch_IndFil	16	1
17	Sch_tertiary	17	1
18	db_owner	16384	16384
19	db_accessadmin	16385	1
20	db_securityadmin	16386	16386
21	db_ddladmin	16387	16387
22	db_backupoperator	16389	1
23	db_datareader	16390	7
24	db_datawriter	16391	7

Esquema predeterminado

Especifica el esquema al que pertenecerán los objetos creados por este usuario, excepto si se indica lo contrario.

Los usuarios de la base de datos tienen un esquema predeterminado (dbo). Se puede cambiar de esquema predeterminado.

Esquema – Mover objetos

Al mover un objeto de un esquema a otro esquema, todos los permisos asociados al elemento protegible (el objeto) serán quitados.

2.1.5 Seguridad (Transferencia entre Esquemas)

Práctica – Permisos nivel objeto

Crearemos un inicio de sesión y un usuario de base de datos para **Peter**. **Peter** necesita tener acceso a los objetos del esquema *HumanResources* de *AdventureWorks*.

1. Crearemos el Inicio de sesión y el usuario de base de datos PETER que accede a la base de datos Adventure Works.

```
CREATE LOGIN Peter WITH PASSWORD='Pa$$w0rd'
```

```
GO
```

```
USE AdventureWorks
```

```
GO
```

```
CREATE USER Peter FROM LOGIN Peter
```

2. Damos permiso a *Peter* para leer en objetos de base de datos pertenecientes al esquema *HumanResources*

```
GRANT SELECT ON SCHEMA::[HumanResources] TO [Peter]
```

3. Cerramos la conexión, abrimos una nueva y entramos como PETER.

Ejecutamos:

```
USE AdventureWorks
```

```
GO
```

```
SELECT * FROM HumanResources.Employee
```

Consultas de Diccionario

En este guión *.sql* tenemos algunas consultas útiles relacionadas con el el tema expuesto.

2.1.6 Seguridad (Consultas Diccionario)

Más información

C/ Miracruz, 10 (Bº de Gros) 20001 Donostia

Telf.: 943 275819

email: seim@centroseim.com

