APUNTE PROYECTO INTEGRADOR BD I

TEMA: MANEJO DE PERMISOS A NIVEL DE USUARIOS DE BASE DE DATOS.

Índice

Usuarios.

¿Qué son los usuarios?

Al igual que en otros servicios, SQL Server ofrece la opción de crear diferentes tipos de usuarios. Un usuario es una entidad de seguridad de la base de datos. Dependiendo del usuario, puede haber o no un inicio de sesión ligado a él.

Un usuario y un inicio de sesión **NO SON LO MISMO.**

Un usuario puede tener permisos dentro de la base de datos, pero sin un inicio de sesión no podrá conectarse al servidor.

A continuación, se detallan los tipos de usuarios más comunes:

- **SQL User sin Login**: Usuario que existe solo en la base de datos y no tiene un inicio de sesión asociado a nivel de servidor.
- **SQL User con Login**: Usuario de la base de datos que está asociado a un inicio de sesión de SQL Server (Login). Este login puede ser de autenticación SQL o autenticación de Windows.
- **Windows User**: Usuario de Windows que tiene permisos de acceso a la base de datos a través de su cuenta de Active Directory.
- **Application Role (Rol de Aplicación)**: Rol que se utiliza dentro de una aplicación para ejecutar con permisos específicos, sin estar ligado a un usuario individual.

¿Qué son los inicios de sesión?

Un inicio de sesión es una entidad de seguridad o una entidad que puede ser autenticada por un sistema seguro. Se puede crear un inicio de sesión basado en una entidad de seguridad de Windows (como un usuario de dominio o un grupo de dominio de Windows) o se puede crear un inicio de sesión que no esté basado en una entidad de seguridad de Windows (como un inicio de sesión de SQL Server).

Los inicios de sesión deben estar asignados a un usuario para poder conectarse a una base de datos. Un inicio de sesión se puede asignar a bases de datos diferentes como usuarios diferentes, pero solo se puede asignar como un usuario en cada base de datos.

Alguno de los inicios de sesión de SQL Server son los siguientes:

- **SQL Login**: Usuario autenticado mediante SQL Server Authentication, donde se crea un nombre de usuario y una contraseña dentro de SQL Server.

- **Windows Login**: Usuario autenticado mediante Windows Authentication, utilizando las credenciales de Active Directory.
- **Login de Certificado**: Basado en certificados para la autenticación.
- **Login de Asimetría (Asymmetric Key Login)**: Basado en claves asimétricas para la autenticación.

¿Cómo se crea un usuario?

1. **Usuario que se autentica en la base de datos contenida (contained database users):** Estos usuarios se crean y se autentican **dentro de la base de datos especificada**, por lo que solamente tendrán acceso a esa base de datos. No dependen de un inicio de sesión. Se crean utilizando las siguientes cláusulas:

CREATE USER NombreUsuario WITH PASSWORD = 'Contraseña';

---

2. **Usuarios basados en inicios de sesión en la base de datos principal:** Estos usuarios dependen de un inicio de sesión. El inicio de sesión se crea a nivel del servidor y, luego, los usuarios a nivel de base de datos se asocian a ese inicio de sesión.
   Primero creamos el inicio de sesión:

CREATE LOGIN NombreLogin CON PASSWORD = 'Contraseña';

   Luego creamos el usuario dentro de la base de datos que se asocia con ese inicio de sesión

---

USE consorcio;

CREATE USER NombreUsuario FROM LOGIN NombreLogin;

---

3. **Usuarios sin inicio de sesión**: Estos son usuarios creados en la base de datos que no están asociados a ningún inicio de sesión en el servidor. Al no tener ningún login asociado en el servidor, son útiles para escenarios de acceso limitado o automatización.

CREATE USER NombreUsuario WITHOUT LOGIN;

Permisos.

¿Qué son los permisos?

En una base de datos existen "permisos", los cuales son derechos que determinan lo que un usuario puede o no puede hacer con el manejo de los datos, específicamente con las tablas, datos, procedimientos, etc.…

Permisos a nivel de usuario.

Los permisos a nivel de usuario pueden incluir algunos de los siguientes derechos:

1. **SELECT:** Permite ver o consultar datos (leer información de una tabla).
2. **INSERT:** Permite agregar nuevos registros a una tabla.
3. **UPDATE:** Permite modificar registros existentes.
4. **DELETE:** Permite eliminar registros de una tabla.
5. **EXECUTE:** Permite ejecutar procedimientos almacenados

Permisos a nivel de servidor.

los permisos a nivel de servidor determinan lo que un *inicio de sesión* puede hacer en la instancia completa del servidor, no solo dentro de una base de datos específica. Estos permisos son más amplios y afectan tareas administrativas y de gestión en toda la instancia del servidor.

Alguno de los más usados son los siguientes:

- **ALTER ANY LOGIN**: Permite crear, modificar y eliminar inicios de sesión en el servidor.
- **ALTER SERVER STATE**: Permite modificar el estado del servidor.
- **ALTER ANY DATABASE**: Permite crear, modificar o eliminar bases de datos en el servidor.
- **VIEW SERVER STATE**: Permite ver el estado general del servidor.
- **SHUTDOWN**: Permite detener la instancia de SQL Server.
- **CREATE ANY DATABASE**: Permite crear nuevas bases de datos en el servidor.
- **CREATE ENDPOINT**: Permite crear puntos de conexión de SQL Server.
- **VIEW ANY DATABASE**: Permite ver las bases de datos existentes en el servidor.
- **CREATE SERVER ROLE**: Permite crear roles a nivel de servidor.

- **GRANT**: Otorga un permiso específico a un inicio de sesión.
- **REVOKE**: Elimina un permiso previamente otorgado o denegado.
- **DENY**: Niega un permiso específico a un inicio de sesión, incluso si pertenece a un rol que tiene ese permiso.

¿Por qué son importantes estos permisos?

Nos permiten un control personalizado, se puede definir exactamente qué partes de la base de datos puede ver o modificar cada persona.

Este control de acceso es crucial para asegurar que solo las personas autorizadas puedan ver o modificar los datos, garantizando la integridad y la seguridad de la información almacenada.

Por ejemplo, un usuario podría tener permiso de lectura en una tabla (para ver los datos) pero no de escritura (para insertar o modificar información).

¿Cómo se otorgan/eliminan estos permisos?

Los administradores de la base de datos pueden otorgar estos permisos individualmente a cada usuario, mediante la cláusula *GRANT* y para eliminar se utiliza la cláusula *REVOKE*.

GRANT *SELECT, INSERT, UPDATE, DELETE ON consorcio TO Admin;*

Ejemplo para otorgar todos los permisos a un administrador en la tabla consorcio

---

REVOKE *INSERT ON consorcio FROM Admin;*

Ejemplo para eliminar el permiso de insertar a un administrador en la tabla consorcio

¿Qué ventajas nos dan los permisos?

- Seguridad: Solo los usuarios autorizados tienen acceso a funciones específicas, minimizando el riesgo de pérdida o modificación indebida de datos.
- Flexibilidad: Los permisos se pueden ajustar según las responsabilidades de cada usuario en la organización.

- Auditoría: Es más fácil rastrear quién tiene acceso a qué y controlar cómo se utiliza esa información.

Roles.

¿Qué son los roles?

Los roles son conjuntos predefinidos de permisos que pueden ser asignados a usuarios para facilitar la gestión de permisos. Al igual que los inicios de sesión, existen roles a nivel de Base de datos y a nivel del Servidor.

Roles predeterminados en Base de Datos:

- db_owner: Tiene todos los permisos sobre la base de datos.
- db_securityadmin: Gestiona roles y permisos dentro de la base de datos.
- db_accessadmin: Administra el acceso de los usuarios a la base de datos.
- db_backupoperator: Puede realizar copias de seguridad de la base de datos.
- db_datareader: Puede leer todos los datos de todas las tablas de la base de datos.
- db_datawriter: Puede escribir datos en todas las tablas de la base de datos.
- db_ddladmin: Puede ejecutar comandos DDL (crear, modificar tablas, procedimientos, etc.).
- db_denydatareader: Deniega el permiso de lectura en todas las tablas.
- db_denydatawriter: Deniega el permiso de escritura en todas las tablas.

Roles predeterminados a nivel de Servidor:

- sysadmin: Tiene control total sobre la instancia de SQL Server.
- serveradmin: Gestiona la configuración del servidor.
- securityadmin: Gestiona permisos y accesos.
- setupadmin: Gestiona la configuración de enlaces a otros servidores y configuraciones relacionadas.
- processadmin: Gestiona los procesos que se ejecutan en la instancia.
- bulkadmin: Puede ejecutar operaciones de carga masiva de datos.
- diskadmin: Administra los archivos físicos del servidor.
- dbcreator: Puede crear, alterar y eliminar bases de datos.

¿Cómo se crean los roles?

Rol en base de datos

Una forma más fácil de asignar permisos a muchos usuarios es creando roles con los permisos necesarios y asignando dichos usuarios a cada rol. Por ejemplo, si quisiéramos tener un rol de “Lectura”. En lugar de asignar el permiso de lectura uno por uno a cada usuario, puedes crear un rol y agregar usuarios a ese rol, de la siguiente forma:

Primero: Seleccionamos la base de datos donde existirá este rol, en caso de que no seleccionar ninguna, se utilizará la que esté por defecto:

USE consorcio;

Segundo: Creamos el rol de Lectores.

CREATE ROLE Lectores;

Tercero: Otorgamos el permiso de lectura al rol.

---

*GRANT **SELECT** ON SCHEMA::dbo TO Lectores;*

Cuarto: Agregamos usuarios al rol

EXEC sp_addrolemember 'Lectores', 'Usuario1';

EXEC sp_addrolemember 'Lectores', 'Usuario2';

De esta manera, se estaría creando un rol en la base de datos “consorcio” con el permiso de leer datos, y se agregaron dos usuarios.

Rol a nivel del servidor

Para crear un rol a nivel del servidor se usan las siguientes clausulas:

Primero creamos el rol y le asignamos un nombre

CREATE SERVER ROLE AdminServidor;

Luego asignamos los permisos al rol, por ejemplo, si queremos para ver todas las bases de datos:

---

*GRANT **VIEW ANY DATABASE** TO AdminServidor;*

Si queremos agregar un usuario al rol, usamos las siguientes cláusulas:

ALTER SERVER ROLE AdminServidor ADD MEMBER UsuarioAdmin;
