Usuarios en SQL Server

Gestión de permisos de usuarios en SQL Server

Miguel Marañón Quero

2024

Contenido

[Introducción 2](#_Toc151456625)

[Tabla de referencias cruzadas 2](#_Toc151456626)

[Creación de los inicios de sesión 2](#_Toc151456627)

[Creación de Usuarios y permisos 4](#_Toc151456628)

[Secretaría 4](#_Toc151456629)

[Profesores 5](#_Toc151456630)

[Alumnos 6](#_Toc151456631)

[Creación y manipulación de roles 7](#_Toc151456632)

[Roles del sistema 7](#_Toc151456633)

[Roles personalizados 7](#_Toc151456634)

# Introducción

A la hora de manipular una base de datos de cara a su uso por parte de usuarios ajenos a la administración de esta, necesitamos tener claro cuales van a ser los permisos que se les va a conceder a cada usuario para que puedan ejercer sus labores correctamente sin poner en peligro la integridad de la propia base de datos ni permitiéndoles el acceso a información que no deberían poder ver.

Las bases de datos tienen tres elementos principales de cara a esta gestión: los inicios de sesión, los usuarios y los roles. Cada elemento se superpone al siguiente siendo los inicios de sesión los primeros en ser creados al permitir a alguien poder acceder a la base de datos en primera instancia y los roles los últimos al ser los elementos que vamos a utilizar para poder gestionar los permisos de los usuarios.

# Tabla de referencias cruzadas

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Alumno | | | | Asignatura | | | | Matrícula | | | |
| permisos | S | I | U | D | S | I | U | D | S | I | U | D |
| Secretaria |  |  |  |  |  |  |  |  |  |  |  |  |
| Profesor |  |  |  |  |  |  |  |  |  |  | nota |  |
| Alumno |  |  |  |  | nom |  |  |  | nota |  |  |  |

# Creación de los inicios de sesión

Los inicios de sesión pueden ser de dos tipos, autenticación de Windows o autenticación de SQL Server. El primero se caracteriza por estar asociado a un usuario de Windows, no lo vamos a utilizar ya que nuestro temario se centra en el uso directo del gestor de base de datos por lo que el método de autenticación de SQL Server será el que mas se adapte a nuestras necesidades.

Lo primero que necesitamos para crear un inicio de sesión es un nombre de usuario y una contraseña. Una vez creados vamos a pensar en algo muy importante: ¿qué necesidades ocupa el inicio de sesión?

Vamos a ponernos en el caso de la base de datos de una escuela, como nos dicta el ejercicio 3 del temario. Desde una vista general tenemos dos tipos de inicio de sesión:

* Logins de la **base de datos** (secretaría, profesorado y alumnos)
* Logins del **servidor** (administrador y operador de copias de seguridad)

Ahora que tenemos diferenciados los dos tipos de inicios de sesión, los podemos crear, solo que hay una diferencia muy importante para poder crearlos bien que viene dada por su tipo de login (base de datos o servidor)

* Los de la base de datos usaran usuarios dentro de la base de datos escuela y se crearan dentro de la base de datos master
* Los del servidor se crearán dentro de la base de datos master

Esto se podrá establecer poniendo USE master justo antes de la orden de creación en la consulta.

Captura de pantalla de un celular

Descripción generada automáticamente

Aquí tenemos el fragmento del código que usaremos para crear los inicios de sesión, si nos vamos al apartado de seguridad ubicado en la raíz de directorios, podemos encontrar todos los inicios de sesión que creemos en esta ruta:

**WN11\_MIGUEL\Seguridad\Inicios de sesión**

Para continuar, vamos a crear el resto de los inicios de sesión que vamos a necesitar (en este caso los ubicados directamente en ella)

Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

# Creación de Usuarios y permisos

Ahora que tenemos nuestros inicios de sesión, necesitamos crear usuarios para que la base de datos pueda ser visualizada o manipulada desde fuera. Para ello necesitamos crear un usuario asociado a cada [inicio de sesión de la base de datos](#_Creación_de_los).

Es muy importante que los usuarios se creen dentro de la base de datos escuela, de lo contrario podrían no tener acceso a ella y habría que darselo de forma manual, lo haremos escribiendo la orden USE escuela justo antes de la creación

Para crear estos usuarios simplemente tenemos que escribir la línea CREATE USER seguido del nombre y luego FOR LOGIN seguido del inicio de sesión que habíamos creado para este usuario. Tenemos 3 distintos:

* Secretaría: puede leer y escribir la base de datos en su totalidad
* Profesores: puede leer toda la base de datos, pero solo manipular la variable notas de la tabla matrícula.
* Alumnos: solo pueden consultar su información personal, asignaturas en las que están matriculados y sus notas.

## Secretaría

Vamos a crear el usuario de Secretaría, siendo el mas sencillo primero:

Texto

Descripción generada automáticamente

Podemos ver que utilizamos la orden ALTER ROLE, esto lo estamos haciendo para conceder los permisos que necesita el personal de secretaría, pero como lo hacemos mediante roles eso lo veremos en el siguiente punto.

Vamos a seguir con la creación de usuarios para meternos en la manipulación de permisos.

## Profesores

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Como podemos ver, tenemos dos opciones para manipular los permisos, la primera otorga directamente los permisos al usuario, siendo este un método simple pero que si necesitamos aplicárselo a 10 usuarios distintos es menos efectivo.

Al hacer esto le estamos dando permiso para poder utilizar el comando UPDATE dentro de una consulta, mas concretamente en la variable nota de la tabla Matricula, llegados a este punto esta todo correcto.

La otra forma es mas interesante, si necesitamos otorgar este permiso concreto de manipular la variable nota a varios usuarios, podemos crear un procedimiento almacenado que se encargue de hacerlo, así evitamos otorgar el comando UPDATE poniendo más seguridad al sistema.

En su lugar vamos a otorgar el comando EXECUTE para poder utilizar un procedimiento almacenado, luego tendremos que crear el procedimiento que queremos.

Texto

Descripción generada automáticamente

Como podemos ver, el procedimiento hace lo mismo que el permiso UPDATE, le introducimos un numero de matrícula, un ID de asignatura y una nota. Todo esto lo utiliza para actualizar la tabla matricula sustituyendo la variable nota por la que se ha introducido en el procedimiento solo en la línea en la que coincida el numero de la matricula introducido y el ID de la asignatura introducida.

## Alumnos

Para la creación de alumnos vamos a seguir el mismo proceso que hemos usado hasta ahora.

Texto

Descripción generada automáticamente

Como podemos ver, estamos concediendo un permiso a dos elementos: el primer elemento es la tabla alumno, dándole permiso para ver la tabla entera sin problema alguno. El segundo elemento es una vista, la cual se va a encargar de dar únicamente la información que queremos que el alumno pueda ver (solo pueden consultar las asignaturas en las que están matriculados y sus notas). La vista en cuestión es:

Texto

Descripción generada automáticamente

Para poder visualizar que los usuarios han sido creados Correctamente, podemos irnos a la siguiente ruta:

**WN11\_MIGUEL\Bases de datos\escuela\Seguridad\Usuarios**

Texto

Descripción generada automáticamente con confianza baja

# Creación y manipulación de roles

Los roles son un elemento esencial, el tercero en nuestra lista. Con la creación de roles podemos ahorrarnos muchísimo trabajo a la hora de otorgar permisos, haciendo que un rol pueda contener una serie de permisos A, C y D mientras que otro rol tenga los permisos B, E y un ultimo rol tenga el F y G. en vez de conceder estos grupos de permisos a los usuarios, podemos añadirlos directamente a los roles para que así puedan gozar de los mismos permisos siendo el proceso mucho más rápido y seguro.

## Roles del sistema

Como podemos ver, tras crear el usuario “SecretaríaSQL” estamos añadiéndolo a una serie de roles del sistema. Estos roles vienen creados ya dentro del gestor de base de datos, no hay que confundirlos con los roles que nosotros vamos a crear más tarde para poder manipular los permisos. El funcionamiento de estos roles es limitado para poder personalizar bien nuestros permisos, pero vienen muy bien cuando un usuario necesita control total sobre una acción. Cada rol tiene asociado una serie de ordenes de consulta, por ejemplo, otorgar el rol db\_datareader otorga permisos para poder usar SELECT sobre cualquier tabla de la base de datos.

Hay dos roles que vamos a estar utilizando mucho de aquí en adelante, el db\_datareader y db\_datawriter. Estos roles tal y como su nombre indican permiten leer y escribir todo el contenido de una base de datos sin excepción, como nuestro personal de secretaría necesita tener acceso a toda la base de datos y poder manipularla sin problemas, vamos a otorgarle estos roles del sistema.

Hay otros que por ejemplo excluyen de permisos, como puede ser el db\_denydatareader y db\_denydatawriter. Estos roles aplican una negación a los anteriores haciendo que, si un usuario los tiene, no se le permita leer o escribir. Se utilizan muchísimo para aplicarlos a grupos de usuarios enteros que pertenecen a un grupo mas grande. Siendo el grupo A el grande y el B el pequeño que pertenece al grupo grande, añadimos al rol db\_datawriter al grupo A para luego añadir al rol db\_denydatawriter al grupo B. De esta forma todos los miembros del grupo A podrán editar la base de datos, pero excluimos a los que también sean del grupo B sin perjudicar al resto.

## Roles personalizados

Si queremos crear nuestros propios roles, necesitamos tener en cuenta para qué los vamos a usar. Como he dicho antes, los roles se utilizan de forma optima para agrupar usuarios, como no es nuestro caso en este ejercicio no necesitamos ningún rol, pero para poder ver como funcionan vamos a crear uno para el usuario profesor y así poder ver todo el proceso.

Lo primero que hay que hacer es crear el rol con la orden CREATE ROLE seguido del nombre del rol. Luego podemos hacer dos cosas:

* Añadir el rol nuevo a un rol del sistema, dándole los permisos por herencia de esta forma (ALTER ROLE db\_dataX ADD MEMBER RolProfesorSQL)
* Otorgar permisos al rol de forma manual (GRANT x ON y(z) TO Profesor SQL).

Texto

Descripción generada automáticamente

Como podemos ver, hemos hecho que el rol tenga acceso total a la lectura de la base de datos y escritura sobre la variable nota de la tabla matrícula.

También podemos otorgar el permiso EXECUTE en el procedimiento almacenado que habíamos creado, haciendo que no este ligado directamente al usuario profesor, sino que esté ligado al rol al cual pertenece el usuario.

