DATA BASE

MANEJO DE PERMISOS A NIVEL DE USUARIOS

GAUNA, OCTAVIO
GIMENEZ, MAXIMILIANO
MARIN, MATIAS
LOVATO, MATIAS

GRUPO 7

Tabla de Contenido:

1 Introducción:

En el ámbito de la gestión de datos de sistemas computacionales, las bases de datos SQL (Structured Query Language) han tenido una gran influencia en la organización y recuperación de la información de manera eficiente. Estas mismas, han ido evolucionado considerablemente a lo largo del tiempo, proporcionando a las organizaciones distintas ventajas como la capacidad de guardar y acceder a grandes volúmenes de datos de manera segura y efectiva. Sin embargo, para que esto sea así, la gestión de permisos es un aspecto fundamental que merece una atención continua y cuidadosa. Es por ello, que es importante diferenciar entre dos tipos que son, una a nivel **servidor** el cual abarca a los inicios de sesión y los roles dentro de este y la otra a nivel de **base de datos** asignados a usuarios y roles de la misma. [1], [2], [3]

El propósito de este trabajo de investigación es analizar en profundidad las prácticas actuales de manejo de permisos en bases de datos SQL, así como explorar algunas prácticas en este campo. Se examinarán aspectos clave, como la estructura de permisos en bases de datos, las formas de gestionarlos y las limitaciones que estos conllevan.

1.1 Objetivo General:

En esta investigación nos proponemos ahondar, concretamente, en el **Manejo de permisos a nivel de usuarios** con el objetivo de comprender, desarrollar y aplicar dichos conceptos dentro de bases de datos utilizando el motor de SQL Server para lograrlo.

1.2 Objetivos específicos:

1. Analizar la estructura de permisos en SQL Server a nivel de usuario y algunos de los tipos de permisos existentes como permisos de lectura, ejecución y administración.
2. Implementar diferentes roles de seguridad en una base de datos de SQL Server y asignar permisos específicos a dichos roles.
3. Crear usuarios de base de datos en SQL Server y asignarlos a roles con diferentes permisos.
4. Probar el funcionamiento de los permisos implementados mediante ejecución de consultas SQL con los diferentes usuarios creados.
5. Evaluar la efectividad de las políticas de seguridad implementadas a nivel de permisos de usuario.
6. Documentar el proceso de administración y prueba de permisos de usuarios implementado en la base de datos.

2 Marco Conceptual:

- **Usuarios a nivel de servidor:** Son aquellos que se encargan de la administración del servidor cómo seguridad o inicio de sesión, donde existen permisos ordenados jerárquicamente que son asignados a los mismos restringiendo su accionar, los cuales se pueden propagar a los permisos a nivel de base de datos. [2]
- **Usuarios a nivel de base de datos:** Son aquellos que se encargan de la administración a nivel de base de datos para los cuales existen permisos que delimitan sus funciones en toda la misma evitando su libre manipulación. Existen dos tipos de roles para los usuarios de base de datos: [3]
    - **Roles Fijos de Base de Datos:** son roles predefinidos proporcionados por SQL Server que permiten asignar permisos y tareas específicas a los usuarios en una base de datos, los cuales no pueden ser modificados ni eliminados. [3]
    - **Roles de Base de Datos Definidos por el Usuario:** son roles que los administradores o desarrolladores de bases de datos, a diferencia del anterior, pueden crear en estas para satisfacer requisitos específicos de seguridad y organización de la misma, los cuales pueden ser modificados o eliminados. [3]
- **Gestión de Bases de Datos**: La gestión de bases de datos abarca la administración, organización y manipulación de datos. El manejo de permisos es una parte fundamental de este proceso para garantizar la seguridad y la integridad de los datos.
- **Permisos de Usuario**: Los permisos de usuario son reglas que definen qué operaciones pueden realizar los usuarios en una base de datos. Esto incluye permisos para leer, escribir, modificar o eliminar datos.
- **Lectura** (SELECT): Los usuarios con permisos de lectura pueden consultar y visualizar datos en la base de datos, pero no pueden realizar cambios en ellos.
- **Escritura** (INSERT, UPDATE, DELETE): Los permisos de escritura permiten a los usuarios agregar nuevos registros, actualizar registros existentes o eliminar registros de la base de datos. Estos son permisos críticos, ya que pueden modificar la información almacenada.
- **Administración de Usuarios** (GRANT, REVOKE): Los administradores pueden otorgar y revocar permisos a otros usuarios. Esto es fundamental para mantener un control de acceso seguro.
- **Procedimientos almacenados:** Un procedimiento almacenado de SQL Server es un grupo de una o varias instrucciones Transact-SQL. [4]

3 Metodología Seguida:

3.1 Descripción de cómo se realizó el Trabajo Práctico:

Para iniciar el desarrollo de nuestro proyecto, planificamos y coordinamos el proyecto mediante una reunión a través de la plataforma Discord. Durante esta sesión, discutimos y trazamos un plan de acción que nos serviría como hoja de ruta para el trabajo.

Tras esta fase inicial, comenzamos con la investigación sobre el tema central de nuestro proyecto. Cada uno de los participantes se dedicó a investigar de manera individual, explorando diversas fuentes de información, documentos relevantes y estudios relacionados lo que nos permitió obtener una comprensión más profunda del tema en cuestión.

Una vez que contamos con los conocimientos necesarios, procedimos a la fase de implementación práctica de nuestro proyecto. En este punto, cada miembro del equipo se centró en desarrollar código y herramientas de manera individual, lo que nos brindó la oportunidad de sumergirnos en los detalles técnicos y entender a fondo las complejidades de lo que estábamos construyendo. Esta aproximación individual permitió a cada uno de nosotros adquirir conocimiento de la implementación, lo que a su vez contribuyó a la calidad y robustez del proyecto en su conjunto.

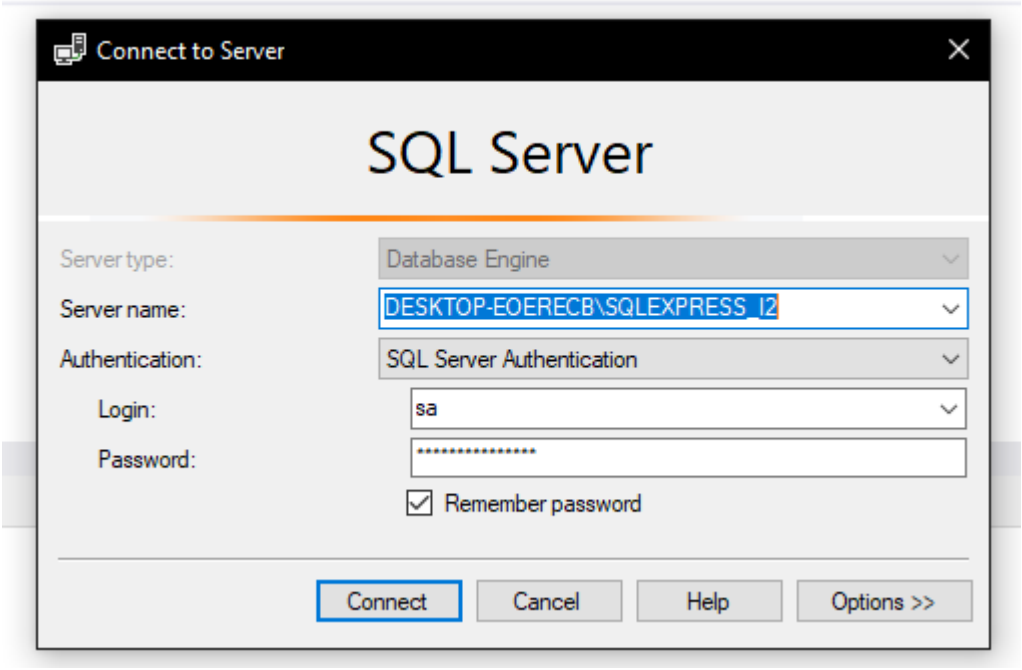
Finalmente, una vez que cada miembro había completado su contribución, unificamos nuestro trabajo y conocimientos para crear una solución coherente.

3.2 Herramientas (Instrumentos y procedimientos):

- **Discord:** Para las comunicaciones grupales.
- **WhatsApp:** Para determinar los avances realizados por cada integrante.
- **Chat GPT:** Para complementar a los conocimientos adquiridos en la investigación y en el ejercicio práctico.
- **GitHub:** Para ir subiendo los avances a nivel código que fuimos realizando en el equipo.
- **SQL Server Management Studio:** Como herramienta de administración y desarrollo de bases de datos proporcionada por Microsoft.
- **Google Docs:** Para realizar los aportes de información y redacción de cada

4 Desarrollo del Tema / Presentación de Resultado:

Para poder desarrollar la parte práctica y cumplir con los objetivos planteados en el informe que estamos llevado a cabo fuimos realizando una serie de pasos que se evidenciará a continuación.
Para empezar, iniciamos una instancia del motor SQL Server en modo mixto para poder manejar usuarios.



Con el modo mixto creamos 2 usuarios a nivel servidor.

```
-- se necesita tener una instancia MIXTA para realizar esto
-- Crear dos usuarios de base de datos:
CREATE LOGIN UsuarioAdmin WITH PASSWORD = 'pwAdmin';
CREATE LOGIN UsuarioSoloLectura WITH PASSWORD = 'pwSoloLectura';
```

Luego, limitamos a los usuarios para que uno sea administrador y el otro tenga permiso de **solo lectura.**

```
--Asignar permisos al usuario de administrador:
USE base_consorcioPI;
CREATE USER UsuarioAdmin FOR LOGIN UsuarioAdmin;
ALTER ROLE db_owner ADD MEMBER UsuarioAdmin;

--Asignar permisos al usuario de solo lectura:
USE base_consorcioPI;
CREATE USER UsuarioSoloLectura FOR LOGIN UsuarioSoloLectura;
GRANT EXECUTE TO UsuarioSoloLectura;
```

Creamos un **procedimiento almacenado** donde se insertan cierta cantidad de registros con datos.

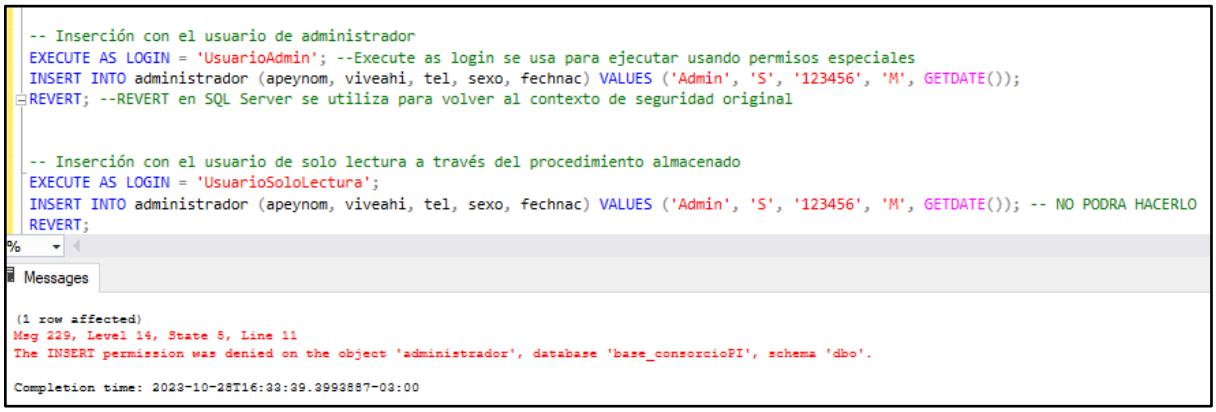
```
CREATE PROCEDURE procedimiento_insert_administrador
AS
BEGIN
    -- Aqui va la logica del procedimiento
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('GONZ♦LEZ JUAN', 'S', '3624235689', 'M', '19801015');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('RODR♦GUEZ MAR♦A', 'N', '3624236689', 'F', '19751203');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('L♦PEZ CARLOS', 'S', '3624237689', 'M', '19790822');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('MART♦NEZ LUC♦A', 'N', '3624238689', 'F', '19820614');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('S♦NCHEZ ANDR♦S', 'S', '3624239689', 'M', '19740520');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('FERN♦NDEZ ELENA', 'N', '3624240689', 'F', '19790110');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('D♦AZ RODRIGO', 'S', '3624241689', 'M', '19831007');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('PEREZ ANA', 'N', '3624242689', 'F', '19720430');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('G♦MEZ LAURA', 'S', '3624243689', 'F', '19810719');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('TORRES JOS♦', 'S', '3624244689', 'M', '19780711');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('RODRIGUEZ HUGO', 'N', '3624245689', 'M', '19810502');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('SILVA MARTINA', 'S', '3624246689', 'F', '19760625');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('LOPEZ JUAN', 'N', '3624247689', 'M', '19800217');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('MARTINEZ MARIA', 'S', '3624248689', 'F', '19731028');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('SANCHEZ PEDRO', 'N', '3624249689', 'M', '19830906');
    Insert into administrador(apeynom,viveahi,tel,sexo,fechnac) values ('GOMEZ ISABEL', 'S', '3624250689', 'F', '19770723');

END;
```

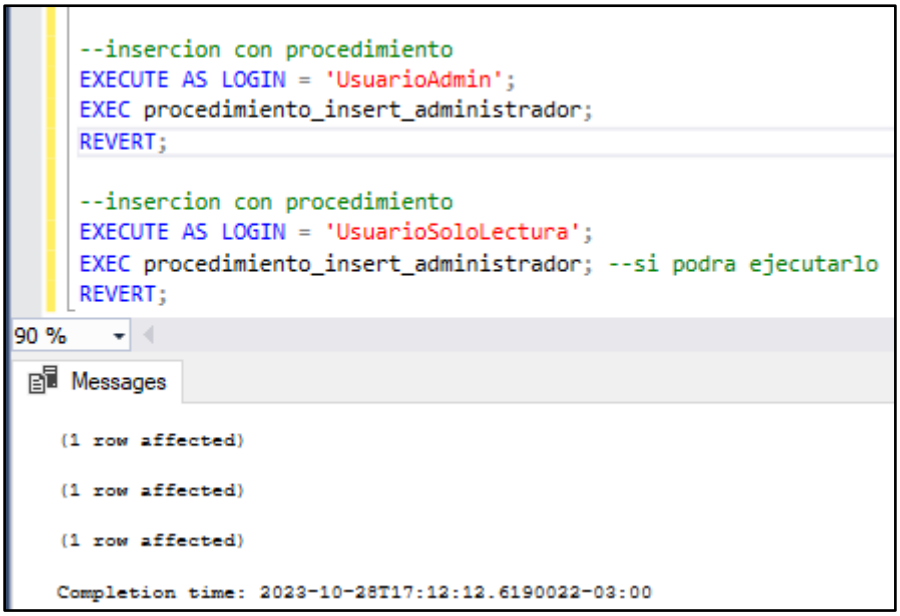
Con la sentencia **GRANT** permitimos al usuario de solo lectura la utilización del **procedimiento almacenado**.

```
--damos acceso al usuario de solo lectura al procedimiento
GRANT EXECUTE ON dbo.procedimiento_insert_administrador TO UsuarioSoloLectura;
```

En la siguiente imagen, ejecutamos un insert con ambos usuarios y podemos observar que el administrador si puede insertar, pero el “UsuarioSoloLectura” **no tiene permiso** para realizar la operación.



En cambio, si ejecutamos el **procedimiento almacenado** al que le dimos permiso de usar al “UsuarioSoloLectura” este si podrá ejecutarse correctamente al igual que el administrador. Esto significa que, a pesar de que "UsuarioSoloLectura" inicialmente tenía permisos limitados de **solo lectura**, al otorgarle permisos de ejecución en el procedimiento almacenado específico, ahora tiene la capacidad de ejecutar ese procedimiento almacenado, lo que le permite realizar acciones más allá de la lectura de datos.



4.1 Conclusión del Desarrollo:

Luego de realizar la sección práctica del tema planteado en el proyecto hemos comprendido la gran importancia de los permisos a nivel de base de datos, ya que al brindarle el rol de administrador a un usuario identificamos que es capaz de realizar una gran cantidad de funciones solamente en la respectiva base de datos a la que se le asignó dichos permisos, como escritura, modificación y eliminación de registros, entre otras cosas, cosa que no es posible cuando se asigna a un usuario el permiso de sólo lectura, ya que este podrá ver la información pero no realizar ninguna de las acciones anteriormente mencionadas exceptuando que a este se le brinde la posibilidad de ejecución de un procedimiento almacenado que en este caso sólo podrá ejecutar las instrucciones que se encuentren dentro de este, cosa que en nuestro caso es la inserción de registros.

5 Conclusión:

Los permisos de usuario en una base de datos son elementos esenciales que nos ayudan a garantizar la seguridad, integridad y privacidad de los datos. Estas reglas establecen de manera precisa las acciones que los usuarios pueden llevar a cabo, tales como la lectura, escritura, modificación o eliminación de información. La adecuada gestión de los permisos es crucial para proteger la información sensible y garantizar un uso eficiente y controlado de los recursos de la base de datos. Al establecer y mantener cuidadosamente los permisos de usuario, se puede lograr un gran equilibrio entre la accesibilidad de los datos y la protección de los mismos, lo que es fundamental en entornos empresariales y de seguridad de la información.

6 Bibliografía:

[1] "Microsoft. 'Permissions (Database Engine)'. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/permissions-database-engine?view=sql-server-ver16 [Accedido: 30 de octubre de 2023]."

[2] "Microsoft. 'Server-Level Roles (SQL Server)'. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-ver16 [Accedido: 30 de octubre de 2023]."

[3] "Microsoft. 'Database-Level Roles (SQL Server)'. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-ver16 [Accedido: 30 de octubre de 2023]."

[4] "Microsoft. 'Stored Procedures (Database Engine)'. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver16 [Accedido: 30 de octubre de 2023]."