Gestión de accesos y permisos en bases de datos

1. Requisitos para conectarse a una base de datos

Para establecer una conexión a una base de datos, es fundamental cumplir con ciertos requisitos básicos que garantizan un acceso seguro y funcional. En primer lugar, se necesita contar con un *Sistema de Gestión de Bases de Datos (DBMS)*, como MySQL, PostgreSQL o SQL Server, el cual proporciona la infraestructura necesaria para el almacenamiento y manipulación de datos.

Adicionalmente, es imprescindible poseer *credenciales de autenticación*, que incluyen un nombre de usuario y una contraseña válidos, los cuales permitirán el acceso autorizado al DBMS. Asimismo, se requiere disponer de *información de conexión*, la cual consiste en detalles como la dirección del servidor (hostname o IP), el puerto de conexión y el nombre de la base de datos a la que se desea acceder.

Por último, es necesario contar con un *cliente o aplicación* que facilite la interacción con la base de datos, ya sea una línea de comandos, una aplicación de escritorio o una aplicación web.

2. Permisos a nivel sistema y a nivel objeto

Los permisos dentro de una base de datos pueden clasificarse en dos niveles principales: permisos a nivel sistema y permisos a nivel objeto.

- **Permisos a nivel sistema**
  Los permisos a nivel sistema afectan a toda la instancia del DBMS y pueden influir en todas las bases de datos gestionadas por el sistema. Este tipo de permisos controla operaciones globales como la creación o eliminación de bases de datos, la gestión de inicios de sesión y la configuración general del servidor.
  Por ejemplo, en SQL Server, el rol fijo de servidor sysadmin otorga permisos completos sobre el servidor, permitiendo la ejecución de cualquier tarea administrativa.
- **Permisos a nivel objeto**
  Por otro lado, los permisos a nivel objeto se aplican a elementos específicos dentro de una base de datos, como tablas, vistas, procedimientos almacenados o funciones. Estos permisos regulan acciones como la selección, inserción, actualización o eliminación de datos dentro de una tabla particular.
  Un ejemplo claro de este tipo de permisos es el permiso SELECT, el cual permite la lectura de los datos contenidos en una tabla.

3. Gestión de permisos

La administración de permisos en bases de datos se lleva a cabo mediante la asignación y revocación de derechos a usuarios o roles para realizar determinadas acciones en el sistema o en objetos específicos.

- **Otorgamiento de permisos**
  Para conceder permisos, se utiliza la sentencia GRANT, la cual otorga a un usuario o rol derechos específicos sobre una base de datos o uno de sus objetos. A continuación, se presentan algunos ejemplos de su uso en distintos sistemas de gestión de bases de datos:
  - ✓ *Ejemplo en MySQL:*

    ```
    GRANT SELECT, INSERT ON nombre_base_datos.nombre_tabla TO 'nombre_usuario'@'localh
    ```

    Este comando otorga permisos de SELECT e INSERT en la tabla nombre_tabla dentro de la base de datos nombre_base_datos al usuario nombre_usuario, permitiéndole conectarse desde localhost.

  - ✓ *Ejemplo en PostgreSQL:*

    ```
    GRANT SELECT, INSERT ON TABLE nombre_tabla TO nombre_usuario;
    ```

    En este caso, el usuario nombre_usuario recibe los permisos de SELECT e INSERT en la tabla nombre_tabla.

- **Revocación de permisos**
  Para retirar permisos previamente concedidos, se emplea la sentencia REVOKE. A continuación, se presenta un ejemplo en SQL Server:

  ```
  REVOKE SELECT, INSERT ON nombre_tabla FROM nombre_usuario;
  ```

  Este comando elimina los permisos de SELECT e INSERT otorgados previamente al usuario nombre_usuario sobre la tabla nombre_tabla.

4. Diferencias entre roles y usuarios

En el contexto de las bases de datos, se pueden distinguir dos conceptos fundamentales: los usuarios y los roles.

- **Usuarios**
  Un usuario es una entidad que representa a una persona o aplicación que interactúa directamente con la base de datos. Cada usuario cuenta con credenciales de autenticación y puede tener permisos asignados de forma individual.
- **Roles**
  Un rol, por otro lado, es una agrupación lógica de permisos que puede ser asignada a uno o más usuarios. El uso de roles facilita la administración de seguridad, ya que permite conceder y revocar permisos de manera centralizada en lugar de hacerlo usuario por usuario.
  Por ejemplo, en SQL Server existen roles predefinidos, como db_datareader, que otorga permisos para leer todos los datos de todas las tablas de usuario dentro de una base de datos.

La principal diferencia entre ambos conceptos radica en que los usuarios son entidades individuales que interactúan con la base de datos, mientras que los roles son conjuntos de permisos que pueden ser asignados a varios usuarios, facilitando la gestión de seguridad y accesos.

Referencias bibliográficas

[1] M. Guarnieri, S. Marinovic, y D. Basin, "Strong and Provably Secure Database Access Control," arXiv preprint arXiv:1512.01479, 2015. [En línea]. Disponible en: https://arxiv.org/abs/1512.01479

[2] Microsoft, "Introducción a los permisos de motor de base de datos - SQL Server," Microsoft Learn. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/getting-started-with-database-engine-permissions

[3] Microsoft, "Roles en el nivel de base de datos - SQL Server," Microsoft Learn. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/database-level-roles

[4] CertiDevs, "Tutorial SQL: Creación y manejo de usuarios y roles," CertiDevs. [En línea]. Disponible en: https://certidevs.com/tutorial-sql-usuarios-roles