Tarea 5

Creación de Usuarios y Roles en un Manejador de Base de Datos

En un manejador de base de datos (DBMS), la creación de usuarios y roles es esencial para gestionar quién puede acceder a la base de datos y qué acciones puede realizar. Los usuarios en un DBMS son entidades que pueden conectarse a la base de datos y realizar operaciones. Para crear un usuario, se utiliza un comando específico del sistema de base de datos. Por ejemplo:

- En **Oracle**: CREATE USER nombre_usuario IDENTIFIED BY contraseña;.
- En **PostgreSQL**: CREATE USER nombre_usuario WITH PASSWORD 'contraseña';.
- En **MySQL**: CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'contraseña';.

Por otro lado, los **roles** son conjuntos de privilegios que se pueden asignar a usuarios para simplificar la administración de permisos. Un rol puede ser otorgado a múltiples usuarios, y un usuario puede tener múltiples roles. El comando para crear un rol es generalmente similar en distintos DBMS, se usa el mismo para los 3 sistemas anteriores:

CREATE ROLE nombre_rol;

Cuando se crean usuarios y roles, se pueden asignar distintas propiedades:

- **Permisos de acceso**: Determinan si el usuario o rol puede conectarse a la base de datos.
- **Perfil de recursos**: Controla el uso de recursos como CPU y tiempo de ejecución.
- **Espacio en disco**: Define las cuotas de espacio en tablaspaces (en Oracle) o en la base de datos en general.
- **Políticas de seguridad**: Configura aspectos como la caducidad de contraseñas y la longitud mínima.
- **Privilegios**: Especifica los privilegios a nivel de sistema o a nivel de objetos que el usuario o rol puede tener.

Privilegios a Nivel Sistema y a Nivel de Objetos

Privilegios a Nivel Sistema

Son aquellos que permiten realizar acciones a nivel de todo el sistema de la base de datos y que no están restringidos a objetos específicos. Estos privilegios suelen ser críticos y deben ser manejados con cuidado, ya que afectan la seguridad y estabilidad del sistema. Algunos ejemplos de estos privilegios son:

- **CREATE USER**: Permite crear nuevos usuarios en el sistema.
- **ALTER SYSTEM**: Permite cambiar parámetros de configuración de la base de datos, como la gestión de memoria o la configuración del sistema.
- **CREATE TABLESPACE**: Permite crear un nuevo tablespace, que es un contenedor lógico para almacenar datos en la base de datos.
- **DROP USER**: Permite eliminar usuarios del sistema, lo cual implica también la eliminación de todos los objetos propiedad del usuario.
- **CREATE SESSION**: Permite iniciar una sesión en la base de datos, lo que es necesario para que un usuario pueda conectarse y operar.

Privilegios a Nivel de Objetos

Los privilegios a nivel de objetos son aquellos que permiten realizar acciones sobre objetos específicos dentro de la base de datos, como tablas, vistas, procedimientos almacenados, etc. Estos privilegios son esenciales para definir qué puede hacer un usuario con los diferentes objetos dentro de la base de datos. Algunos de los ejemplos más utilizados de estos privilegios son:

- **SELECT ON nombre_tabla**: Permite leer datos de una tabla específica.
- **INSERT ON nombre_tabla**: Permite insertar datos en una tabla específica.
- **UPDATE ON nombre_tabla**: Permite actualizar datos en una tabla específica.
- **DELETE ON nombre_tabla**: Permite eliminar datos de una tabla específica.
- **EXECUTE ON nombre_procedimiento**: Permite ejecutar un procedimiento almacenado específico, lo cual es importante en aplicaciones que usan lógica de negocio en la base de datos.

Otorgar y Quitar Privilegios a un Usuario o Rol

Para otorgar privilegios a un usuario o rol, se utiliza el comando **GRANT**. Esto permite asignar tanto privilegios a nivel de sistema como a nivel de objetos a un usuario o rol, dependiendo de las responsabilidades del usuario. Algunos ejemplos son:

- **Otorgar un privilegio a nivel de objeto**: GRANT SELECT ON nombre_tabla TO nombre_usuario;
- **Otorgar un privilegio a nivel de sistema**: GRANT CREATE SESSION TO nombre_usuario;
- **Otorgar un rol a un usuario**: GRANT nombre_rol TO nombre_usuario;

Para revocar o quitar privilegios de un usuario o rol, se utiliza el comando **REVOKE**. Esto es importante para mantener la seguridad y controlar el acceso a ciertos recursos críticos cuando ya un usuario ya no es necesario en esa área o cuando cambia de función. Algunos ejemplos usándolo son:

- **Quitar un privilegio a nivel de objeto**: REVOKE SELECT ON nombre_tabla FROM nombre_usuario;
- **Quitar un privilegio a nivel de sistema**: REVOKE CREATE SESSION FROM nombre_usuario;
- **Quitar un rol de un usuario**: REVOKE nombre_rol FROM nombre_usuario;

Referencias

Oracle. (s.f.). *Oracle Database SQL Language Reference*. Oracle Help Center. https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/index.html

PostgreSQL Global Development Group. (s.f.). *PostgreSQL Documentation: SQL Commands*. PostgreSQL Documentation. https://www.postgresql.org/docs/current/sql-commands.html

Oracle Corporation. (s.f.). *MySQL 8.0 Reference Manual*. MySQL Documentation. https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html

Red Hat. (s.f.). *Guía de administración de bases de datos*. Red Hat Customer Portal. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/database_administration_guide/index

GeeksforGeeks. (s.f.). *SQL | CREATE ROLE Statement*. GeeksforGeeks. https://www.geeksforgeeks.org/sql-create-role-statement/