Gestión de Accesos en Bases de Datos

Tarea 2

Sergio Armando Calzada Lozada

19 de agosto de 2025

1 ¿Qué requiero para conectarme a una base de datos?

Para conectarse a una base de datos es necesario cumplir con ciertos requisitos técnicos y de seguridad:

- Tener instalado el **cliente adecuado** para el motor de base de datos (por ejemplo: MySQL Workbench, psql, SQL Server Management Studio).
- Contar con la **dirección del servidor** (host), el **puerto**, el **nombre de la base de datos** y las **credenciales de acceso**.
- Permisos a nivel de red (firewall o VPN) si la base de datos se encuentra en otro servidor.
- Un **usuario válido** con permisos de autenticación y autorización adecuados.

2 Permisos a nivel sistema y a nivel objeto

Permisos a nivel sistema

Son privilegios globales que permiten realizar tareas administrativas sobre el motor de base de datos completo. Ejemplos:

- Crear usuarios.
- Asignar roles.
- Crear y eliminar bases de datos.
- Controlar parámetros de configuración del servidor.

Permisos a nivel objeto

Son privilegios que se aplican a objetos específicos dentro de una base de datos. Ejemplos:

- Permisos de `SELECT`, `INSERT`, `UPDATE` o `DELETE` sobre una tabla.
- Permisos de ejecución sobre procedimientos almacenados.
- Permisos de uso sobre esquemas o vistas.

3 ¿Cómo dar y quitar permisos?

El control de permisos suele realizarse con las sentencias `GRANT` y `REVOKE` (en SQL estándar):

Dar permisos

```
GRANT SELECT, INSERT ON empleados TO usuario1;
```

Este comando otorga a `usuario1` permisos para leer e insertar datos en la tabla `empleados`.

Quitar permisos

```
REVOKE INSERT ON empleados FROM usuario1;
```

Este comando elimina el permiso de inserción para `usuario1` en la tabla `empleados`.

4 Diferencia entre rol y usuario

- **Usuario:** representa a una persona o aplicación que se conecta a la base de datos. Tiene un nombre de usuario y una contraseña, y puede poseer permisos propios.
- **Rol:** es un conjunto de permisos que se pueden asignar a uno o varios usuarios. Facilita la administración de seguridad, ya que en lugar de asignar permisos uno por uno a cada usuario, se otorgan a un rol y luego se asocia ese rol a múltiples usuarios.

Ejemplo

```
-- Crear un rol
CREATE ROLE analista;

-- Otorgar permisos al rol
GRANT SELECT ON ventas TO analista;

-- Asignar el rol a un usuario
GRANT analista TO usuario2;
```

De esta manera, `usuario2` hereda todos los permisos del rol `analista`.

5 Fuentes de información

- Coronel, C., Morris, S., & Rob, P. (2017). *Database Systems: Design, Implementation, & Management* (12th ed.). Cengage Learning.
- Oracle. (2024). *Database SQL Language Reference*. Recuperado de: `https://docs.oracle.com/`
- PostgreSQL Global Development Group. (2025). *PostgreSQL Documentation*. Recuperado de: `https://www.postgresql.org/docs/`
- Microsoft. (2025). *SQL Server Security Documentation*. Recuperado de: `https://learn.micros`