UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA
COORDINACIÓN DE COMPUTACIÓN

BASES DE DATOS

Tarea 2

Diseño Conceptual de una Base de Datos

Nombre del alumno: Yukioayax Canek Gabriel Hernández
Profesor: Fernando Arreola Franco
Grupo: 1
Semestre en curso: 2026-2

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

Para establecer una conexión con una base de datos desde una aplicación o cliente, se necesitan los siguientes elementos:

- **Controlador:** Es el puente entre la aplicación y el motor de base de datos como JDBC para Java, ODBC para entornos Windows, o conectores nativos como psycopg2 para PostgreSQL.
- **URL de conexión:** Dirección que identifica el motor, el servidor, el puerto y la base de datos específica, por ejemplo `jdbc:mysql://localhost:3306/mi_bd`.
- **Credenciales de acceso:** Nombre de usuario y contraseña válidos en el sistema gestor de bases de datos (SGBD).
- **Protocolo de comunicación:** TCP/IP, sockets locales, etc., según la configuración del servidor.
- **Parámetros adicionales:** Configuraciones como tiempo de espera, codificación de caracteres, SSL, etc., que pueden ser necesarias según el entorno.

En aplicaciones modernas también se suelen usar *connection pools* para optimizar el manejo de múltiples conexiones.

2. Permisos a nivel sistema y a nivel objeto

Los permisos en bases de datos determinan las operaciones que un usuario puede realizar, se clasifican en dos grandes grupos:

2.1. Permisos a nivel sistema

Son aquellos que permiten ejecutar acciones administrativas o sobre la estructura general de la base de datos, no están ligados a un objeto específico sino a capacidades globales; algunos ejemplos comunes son:

- `CREATE USER`, `ALTER USER`, `DROP USER` (gestión de usuarios).
- `CREATE TABLE`, `CREATE DATABASE`, `CREATE VIEW` (creación de objetos).
- `ALTER ANY TABLE`, `DROP ANY TABLE` (afectar tablas de cualquier esquema).
- `SELECT ANY TABLE` (consultar cualquier tabla).
- `EXECUTE ANY PROCEDURE` (ejecutar cualquier procedimiento almacenado).

Estos permisos suelen otorgarse a administradores o roles especiales.

2.2. Permisos a nivel objeto

Son los que se conceden sobre objetos concretos de la base de datos (tablas, vistas, procedimientos, etc.), determinan qué acciones puede realizar un usuario sobre ese objeto, algunos son:

- `SELECT`, `INSERT`, `UPDATE`, `DELETE` sobre una tabla o vista.
- `EXECUTE` sobre un procedimiento o función.
- `REFERENCES` para crear claves foráneas que apunten a una tabla.
- `ALTER` sobre una tabla (cambiar estructura).

Estos permisos se pueden otorgar a usuarios específicos o a roles.

3. ¿Cómo dar y quitar permisos?

La mayoría de los SGBD relacionales utilizan los comandos SQL `GRANT` y `REVOKE` para administrar permisos.

3.1. Otorgar permisos (GRANT)

Sintaxis general:

```
GRANT privilegio1, privilegio2, ...
ON objeto
TO usuario_o_role
[WITH GRANT OPTION];
```

`WITH GRANT OPTION` permite que el receptor pueda otorgar el mismo permiso a otros.
Ejemplos:

- Dar permiso de consulta e inserción en la tabla `Empleados` al usuario `juan`:

  ```
  GRANT SELECT, INSERT ON Empleados TO juan;
  ```

- Dar permiso de creación de tablas a nivel sistema al usuario `admin`:

  ```
  GRANT CREATE TABLE TO admin;
  ```

- Otorgar todos los permisos sobre la vista `Ventas` al rol `consultores`:

  ```
  GRANT ALL PRIVILEGES ON Ventas TO consultores;
  ```

3.2. Quitar permisos (REVOKE)

Sintaxis general:

```
REVOKE privilegio1, privilegio2, ...
ON objeto
FROM usuario_o_role;
```

Ejemplos:

- Quitar el permiso de eliminación en la tabla `Empleados` al usuario `juan`:

  ```
  REVOKE DELETE ON Empleados FROM juan;
  ```

- Revocar el permiso de sistema `CREATE TABLE` a `admin`:

  ```
  REVOKE CREATE TABLE FROM admin;
  ```

Es importante notar que si un permiso fue otorgado con `GRANT OPTION`, al revocarlo también se revocan los permisos que el usuario haya otorgado a otros (dependiendo del SGBD, esto puede ser en cascada o no).

4. Diferencia entre rol y usuario

En el contexto de administración de bases de datos, los conceptos de **usuario** y **rol** son fundamentales:

4.1. Usuario

Es una identidad individual que puede conectarse a la base de datos, cada usuario tiene credenciales (usuario/contraseña) y posee un esquema (conjunto de objetos) propio, los permisos se pueden asignar directamente a un usuario.

4.2. Rol

Es una colección de privilegios que se puede otorgar a usuarios u otros roles; los roles facilitan la administración de permisos al agruparlos según funciones laborales (`ROL_VENTAS`, `ROL_RRHH`).

4.3. Principales diferencias

- **Naturaleza:** Un usuario es una persona o aplicación que se conecta; un rol es una categoría de permisos.
- **Credenciales:** Los usuarios tienen contraseñas; los roles no inician sesión.
- **Pertenencia:** Los usuarios pueden tener asignados uno o varios roles; los roles pueden contener otros roles (jerarquía de roles).
- **Objetivo:** Los usuarios identifican quién realiza las acciones; los roles simplifican la asignación masiva de permisos.

En la práctica, se recomienda asignar permisos a roles y luego otorgar los roles a los usuarios, en lugar de asignar permisos directamente a cada usuario, esto facilita el mantenimiento y la escalabilidad del sistema de permisos.

Referencias

[1] C. J. Date, *An Introduction to Database Systems*, 8th ed. Massachusetts: Addison Wesley, 2003.

[2] R. Elmasri y S. Navathe, *Fundamentos de sistemas de bases de datos*, 3ra ed. Pearson Prentice Hall, 2003.

[3] D. Kroenke, *Procesamiento de bases de datos*, 8a ed. México: Pearson / Prentice Hall, 2003.

[4] P. Rob y C. Coronel, *Database Systems: Design, Implementation and Management*, 6th ed. Course Technology, 2004.

[5] A. De Miguel Martínez, M. Piattini et al., *Diseño de bases de datos relacionales*. México: Alfamaga, 2000.

[6] J. Johnson, *Bases de datos, modelos, lenguajes, diseño*. México: Oxford University Press, 2000.