



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Bases de Datos

Grupo: 01

Tarea 02. Requisitos de una base de datos

Profesor: FERNANDO ARREOLA FRANCO

Alumno: López Romero David Baruc

No. de Cuenta: 321018067

Semestre: 2026-2

Ciudad Universitaria, CDMX
Febrero 2026

1. Requisitos de conexión de una Base de Datos

Para cada base de datos se cuenta con requisitos en específico, sin embargo, también hay algunos requisitos comunes aplicables a todas las bases de datos:

- **Driver o cliente:** es el software que permita a nuestra computadora hablar el idioma de la base de datos (Ejemplos: JDBC, ODBC, OCI)
- **url** de la dirección donde vive la base de datos:
 - Host: la IP del servidor
 - Puerto: el canal de comunicación
 - SID o Service Name: es el nombre en específico de la base de datos a la que se quiere entrar
- **Credenciales:** Usuario y contraseña validas
- **Permisos de red:** Verificar que no hay firewalls bloqueando el puerto

2. Permisos a nivel sistema y Objeto

En SQL Server la seguridad se encuentra estructurada en un doble nivel, el primero es la autenticación, en el se verifica que un usuario que esta intentando conectarse al servidor SQL tenga acceso al sistema, es decir, la 'persona' es quien dice ser y tiene concedido el permiso. EN el segundo nivel se verifica que el usuario tiene los permisos adecuados.

Los permisos a su vez se especifican en dos niveles, en el primero, permisos a nivel de objeto; se incluyen derechos de lectura, modificación y borrado de objetos. Por otro lado, los permisos de sistema son concedidos sobre instrucciones SQL que crean objetos en la base de datos, como bases, tablas, vistas, etc.

3. Otorgar y revocar permisos

En SQL la manera de otorgar o revocar permisos a un usuario, login o grupo se hace mediante el uso de las siguientes instrucciones:

```
-- Otorgar permiso
GRANT [privilegio] ON [objeto] TO [usuario];

-- Revocar permiso
REVOKE [privilegio] ON [objeto] FROM [usuario];
```

GRANT concede permisos sobre un elemento protegible a una entidad de seguridad. **REVOKE** revoca los permisos concedidos y denegados en una base de datos.

4. Diferencia entre 'role' y 'usuario'

En el caso de PostgreSQL los usuarios y roles no son entidades del todo separadas, ya que PostgreSQL maneja todo como role, la diferencia esta en si un role tiene privilegios de login (inicio de sesion) o no

- **user:** es básicamente un role con privilegios de login, es decir, es una cuenta que puede conectar a la base de datos utilizando un usuario y contraseña.
- **role:** es un concepto que puede representar usuarios, grupos o incluso una colección de privilegios. Los roles pueden ser otorgados a otros roles, lo cual permite gestión flexible entre los permisos (Poder crear una jerarquía de permisos).

El uso de roles trae varias ventajas entre ellas:

- En lugar de agregar permisos individuales a cada usuario se pueden definir roles para variedad de trabajos o sets de permisos
- Mejora la administración, ya que agregar o remover el acceso a usuario se vuelve tan simple como otorgar o revocar roles.
- Ayuda en la escalabilidad, ya que a medida que la base de datos crece, los permisos basados en roles facilitan la incorporación de nuevos usuarios o incluso ajustar los permisos sin interrumpir otras cuentas.

Referencias

- [1] ManageEngine, "Prerequisites for db connectivity." Analytics Plus Help Documentation. Accedido: 15-feb-2026.
- [2] IBM Corporation, "Permisos de objeto y sistema." IBM Netcool/OMNIbus 8.1.0 Documentation. Accedido: 15-feb-2026.
- [3] Universidad de Navarra, "Curso de sql: Sentencias de control." Accedido: 15-feb-2026.
- [4] Microsoft, "Grant (transact-sql)." Microsoft Learn. Accedido: 15-feb-2026.
- [5] Microsoft, "Revoke (transact-sql)." Microsoft Learn. Accedido: 15-feb-2026.
- [6] J. RamCloud, "Postgresql users and roles explained: A complete guide for access control," *Medium*. Accedido: 15-feb-2026.