Universidad Nacional Autónoma de México

Facultad de Ingeniería

Tarea 2

Bases de Datos (1644)

Profesor: Ing. Fernando Arreola Franco

Semestre 2026-2

Grupo: 1

Alumna: Cruz Basilio Ximena Carolina

No. de cuenta: 321116424

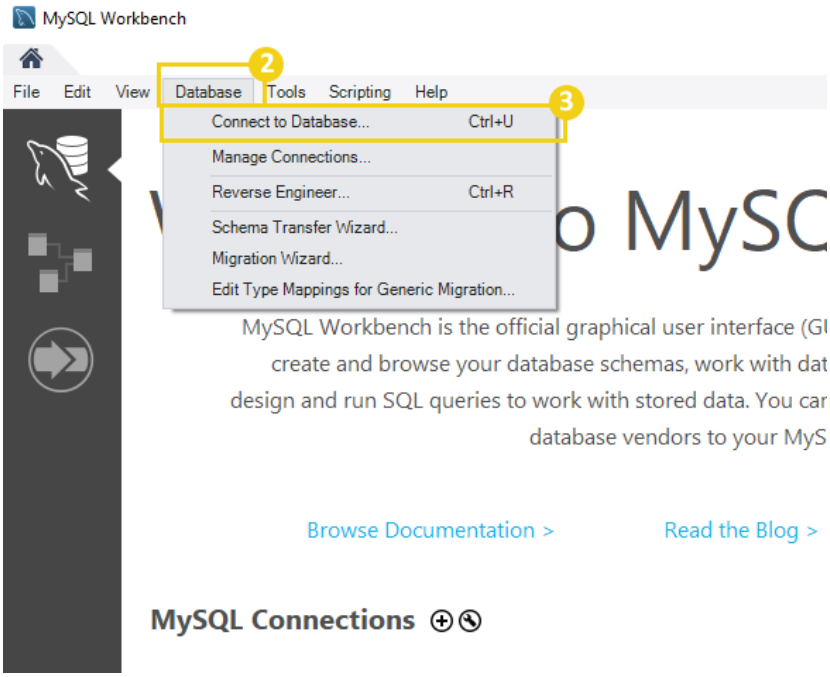
Fecha de entrega: 16 de febrero de 2026

I. ¿Que necesito para conectarme a una BD?

Para poder conectarme a una base de datos, primero necesito tener a la mano algunos datos básicos. Casi siempre me van a pedir el host o servidor (que puede ser una IP o dominio), el puerto que usa el motor, el nombre de la base de datos, el usuario y la contraseña. También es importante saber qué motor de base de datos se está utilizando (por ejemplo PostgreSQL, MySQL, SQL Server o MongoDB), porque de eso depende el driver o conector que debo instalar. Con esta información normalmente se arma una URL o cadena de conexión, que es la forma en que las aplicaciones se conectan a la base de datos.

Además de esos datos mínimos, a veces se requieren cosas extra. Por ejemplo, si la base de datos está en otra red, puede ser necesario usar una VPN o estar conectado a la misma red. También debo verificar que mi usuario tenga los permisos necesarios y que el firewall permita el acceso al puerto correspondiente. En algunos casos la conexión debe ser segura, así que se usa SSL/TLS, lo que implica tener certificados.

En el caso específico de Db2, antes de conectarme debo asegurarme de que esté instalado el driver compatible. Si la herramienta ya lo incluye, puedo usarlo directamente; si no, tengo que instalar el paquete de controladores de Db2. También debo revisar si la conexión será con o sin SSL, porque eso cambia la configuración y puede requerir un certificado. Si hay un firewall, hay que confirmar que estén abiertos los puertos necesarios (por ejemplo, 50000 o 50001). Finalmente, con todos los datos de conexión y credenciales reunidos, ya puedo establecer la conexión correctamente a la base de datos.



II. Permisos a Nivel Sistema y Objetos

Cuando se administra la seguridad de una base de datos, es fundamental distinguir dos tipos principales de permisos: los **permisos de sistema** y los **permisos de objeto**. Ambos determinan qué acciones puede realizar un usuario dentro del sistema, pero en niveles distintos.

II-A. Concepto general de permisos

Los permisos en una base de datos permiten controlar el acceso y las acciones que cada usuario puede realizar. Se asignan normalmente mediante roles y el comando `GRANT`. Su objetivo principal es proteger la información, evitar accesos no autorizados y asegurar que cada usuario tenga únicamente los privilegios necesarios para cumplir sus funciones.

II-B. Permisos de sistema

Los permisos de sistema controlan el acceso al sistema completo de gestión de bases de datos. Permiten ejecutar acciones generales relacionadas con la administración del sistema.

Entre los permisos de sistema más comunes se encuentran:

- `CREATE DATABASE`: permite crear nuevas bases de datos.
- `ALTER DATABASE`: permite modificar bases de datos existentes.
- `DROP DATABASE`: permite eliminar bases de datos.
- `CREATE USER`: permite crear usuarios.
- `CREATE SESSION`: permite conectarse a la base de datos.
- `GRANT ANY PRIVILEGE`: permite otorgar permisos a otros usuarios.

Algunos permisos incluyen la palabra `ANY`, lo que indica que se aplican a cualquier esquema dentro de la base de datos. Por ejemplo:

- `CREATE ANY TABLE`
- `DROP ANY TABLE`

Estos permisos son muy poderosos y deben asignarse con precaución, ya que permiten modificar el sistema completo.

II-C. Permisos de objeto

Los permisos de objeto controlan el acceso a elementos específicos dentro de la base de datos, como tablas, vistas, registros o procedimientos almacenados. Son más específicos y permiten un control más detallado sobre los datos.

Los permisos de objeto más comunes son:

- **SELECT**: permite leer datos.
- **INSERT**: permite agregar datos.
- **UPDATE**: permite modificar datos.
- **DELETE**: permite eliminar datos.

Por ejemplo, un desarrollador puede tener permisos para leer e insertar datos en una tabla, pero no para eliminarlos. Esto ayuda a proteger la información sensible.

II-D. Ejemplo en plataformas empresariales

En sistemas como Salesforce, los permisos de objeto permiten a los usuarios:

- Leer registros

- Crear registros
- Modificar registros
- Eliminar registros

También existen permisos más amplios como:

- **Ver todos los registros**: permite ver todos los datos de un objeto sin importar las reglas de seguridad.
- **Modificar todos los registros**: permite modificar y eliminar todos los registros.

Estos permisos pueden respetar o sobrescribir las reglas de seguridad dependiendo del nivel otorgado.

II-E. Diferencia entre permisos de sistema y de objeto

Tipo de permiso	Controla
Permisos de sistema	El sistema completo de la base de datos
Permisos de objeto	Objetos específicos (tablas, registros, vistas)

III. ¿CÓMO DAR Y QUITAR PERMISOS?

En la administración de seguridad de una base de datos, los comandos de **DCL (Data Control Language)** se utilizan para gestionar permisos y controlar *quién puede hacer qué* dentro del sistema. En particular, los comandos más importantes son:

- `GRANT`: otorga permisos.
- `REVOKE`: revoca (quita) permisos.

Una regla clave es que el usuario que ejecuta `GRANT` o `REVOKE` debe contar con privilegios suficientes (típicamente un administrador o "súper usuario").

III-A. Tipos de permisos: sistema vs. objeto

En general, los permisos se clasifican en **permisos del sistema** y **permisos de objeto**. Esta distinción es importante porque define el alcance del control de acceso.

III-A1. Permisos del sistema: Los **permisos del sistema** son privilegios *globales* que controlan el acceso al sistema gestor de base de datos. Incluyen acciones administrativas como crear usuarios, roles o bases de datos, y ejecutar operaciones de alto nivel.

Ejemplos (según la lista proporcionada):

- `ISQL`, `ISQL WRITE`
- `ALTER SYSTEM DROP CONNECTION`, `ALTER SYSTEM SHUTDOWN`
- `ALTER SYSTEM BACKUP`, `ALTER SYSTEM SET PROPERTY`
- `CREATE DATABASE`, `CREATE USER`, `CREATE ROLE`
- `ALTER USER`, `DROP USER`
- `GRANT ROLE`, `REVOKE ROLE`

III-A1a. Sintaxis para revocar permisos del sistema:

```
REVOKE system_permission,...
FROM ROLE 'role_name',... ;
```

donde `role_name` es el nombre del rol desde el cual se revoca el permiso.

III-A1b. Ejemplo:

```
revoke create table from role 'DDL_Admin';
```

III-A2. Permisos de objeto: Los **permisos de objeto** controlan el acceso a objetos específicos dentro de la base de datos (por ejemplo, tablas, vistas, procedimientos, triggers, etc.). Son más *granulares*, lo que permite asignar permisos según necesidades concretas.

Permisos comunes (según el objeto):

- **TABLA**: `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `ALTER`, `DROP`, `CREATE INDEX`, `DROP INDEX`
- **VIEW**: `SELECT`, `UPDATE`, `DELETE`, `ALTER`, `DROP`
- **DATABASE**: `DROP`, `CREATE TABLE`, `CREATE VIEW`
- **SQL PROCEDURE / EXTERNAL PROCEDURE**: `EXECUTE`, `ALTER`, `DROP`
- **TRIGGER GROUP**: `CREATE TRIGGER`, `ALTER`, `DROP`

III-A2a. Sintaxis para revocar permisos de objeto:

```
REVOKE object_permission,...
ON permission_object object_name
FROM ROLE 'role_name',... ;
```

donde `permission_object` representa el tipo de objeto (por ejemplo `TABLE`, `DATABASE`, `VIEW`) y `object_name` es el nombre del objeto.

III-A2b. Ejemplo:

```
revoke drop on database testdb from role 'DDL_Admin';
```

III-B. Ejemplo práctico: administrador vs. usuario con pocos permisos

Un escenario común (como en la explicación del video) es:

1. Crear un **súper usuario** (administrador) con control total.
2. Crear un usuario normal (por ejemplo, `consultor`) inicialmente **sin privilegios**.
3. El usuario sin privilegios no puede realizar acciones como `CREATE DATABASE` ni acceder a datos sin `SELECT`.
4. El administrador usa `GRANT` para dar permisos mínimos necesarios, por ejemplo:
   ```
   GRANT SELECT ON clientes TO consultor;
   ```
5. Cuando el usuario termina, el administrador usa `REVOKE` para retirar permisos:
   ```
   REVOKE SELECT ON clientes FROM consultor;
   ```

Este enfoque aplica el principio de **mínimo privilegio**: otorgar sólo los permisos necesarios y por el tiempo necesario.

IV. Usuarios, roles y grupos en la administración de seguridad

En los sistemas de administración de bases de datos y aplicaciones empresariales, la seguridad se gestiona mediante la asignación de **usuarios**, **roles** y **grupos**. Estos elementos permiten controlar qué datos pueden ver los usuarios y qué acciones pueden realizar dentro del sistema.

IV-A. Usuarios

Un **usuario** es una cuenta individual que representa a una persona o sistema que accede a la aplicación o base de datos. Cada usuario puede iniciar sesión, tener una contraseña y preferencias propias.

Algunas tareas relacionadas con usuarios incluyen:

- Crear o eliminar usuarios.
- Cambiar contraseñas.
- Configurar preferencias personales (por ejemplo, vistas o filtros).
- Asignar roles y permisos.

Los usuarios pueden administrarse de forma centralizada en un repositorio federado (por ejemplo, LDAP o la consola administrativa de WebSphere), lo que permite gestionar cuentas desde un solo lugar.

Por ejemplo:

- Grupo de administradores
- Grupo de analistas
- Grupo de solo lectura

Cuando se asigna un rol a un grupo, todos los usuarios dentro del grupo heredan ese rol.

IV-B. Roles

Un **rol** define el nivel de acceso que tienen los usuarios o grupos dentro del sistema. Determina:

- qué datos pueden ver
- qué acciones pueden realizar
- qué funciones pueden ejecutar

Por ejemplo, un rol puede permitir:

- solo lectura
- lectura y escritura
- administración completa

Los roles se asignan a usuarios o grupos. Una vez asignado un rol, el usuario puede acceder a los recursos del sistema según los permisos definidos en ese rol.

IV-C. Diferencia entre usuario y rol

Usuario	**Rol**
Es una cuenta individual	Es un conjunto de permisos
Representa a una persona o sistema	Define qué puede hacer el usuario
Tiene contraseña y preferencias	No inicia sesión por sí mismo
Se le asignan roles	Se asigna a usuarios o grupos

REFERENCIAS

[1] YouTube, “Comandos dcl en sql server: Grant y revoke,” YouTube, 2026, consultado: 15 de febrero de 2026. [Online]. Available: https://www.youtube.com/watch?v=B7RVdd0y1kA&t=70s

[2] IBM, “Connecting to db2 database,” IBM Documentation, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://www.ibm.com/docs/es/db2w-as-a-service?topic=database-connecting-db2

[3] ——, “System and object permissions in netcool omnibus,” IBM Documentation, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://www.ibm.com/docs/es/netcoolomnibus/8.1.0?topic=roles-system-object-permissions

[4] Ediciones ENI, “Oracle 12c: Administración — gestión de usuarios y permisos,” Ediciones ENI, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://www.ediciones-eni.com/libro/oracle-12c-administracion-9782746095168/gestion-de-usuarios-y-sus-permisos

[5] FasterCapital, “Dcl para administradores de bases de datos: gestión de permisos y seguridad,” FasterCapital, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://fastercapital.com/es/contenido/DCL-para-administradores-de-bases-de-datos--gestion-de-permisos-y-seguridad.html

[6] IBM, “Select command (aggregate),” IBM Documentation, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://www.ibm.com/docs/es/netcoolomnibus/8.1.0?topic=reference-select-command-aggregate

[7] Microsoft, “Getting started with database engine permissions,” Microsoft Learn, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/getting-started-with-database-engine-permissions?view=sql-server-ver17

[8] ——, “Connection string syntax (ado.net),” Microsoft Learn, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-syntax

[9] Oracle Corporation, “Mysql connector/j jdbc url format,” MySQL Documentation, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://dev.mysql.com/doc/connector-j/en/connector-j-reference-jdbc-url-format.html

[10] PostgreSQL Global Development Group, “libpq — connection strings,” PostgreSQL, s.f., consultado: 15 de febrero de 2026. [Online]. Available: https://www.postgresql.org/docs/current/libpq-connect.html