**Clase 19**

**Sublenguaje DCL**

*EL LENGUAJE DCL*

*CONCEPTO GENERAL*

*DATA CONTROL LANGUAGE*

El **Lenguaje de Control de Datos** (DCL) permite definir diferentes usuarios dentro del motor de base de datos Mysql, y establecer para cada uno de ellos, permisos totales, parciales, o negar el acceso sobre los diferentes Objetos que conforman la Base de Datos.Al igual que con **DDL** y **DML**, **DCL** provee una serie de cláusulas y comandos para poder crear, renombrar y eliminar usuarios dentro de uno o más servidores de base de datos, como así también establecer y/o modificar un password o contraseña de acceso.

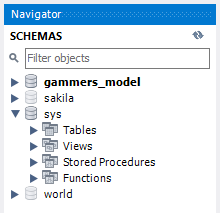
Definidos los usuarios a través de **DCL**, contamos con otro set de comandos y cláusulas para **permitir o revocar el acceso a diferentes Objetos de la base de datos**.

Entendemos por Objetos a: Tablas, Campos, Vistas, Stored Procedures y Funciones Almacenadas.

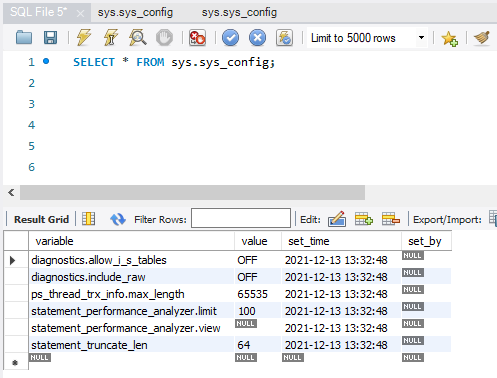
*BASE DE DATOS DEL SISTEMA*

*DB DE SISTEMA*

Mysql cuenta con lo que se conoce como **Base de Datos del Sistema**, donde se almacena información referente al motor de base de datos en sí, performance y demás cuestiones propias de un sistema de software.Cuando instalamos por primera vez a Mysql vemos que, por defecto, se crea la **base de datos SYS** junto al motor de Mysql. Despleguemos la misma para ver su contenido.

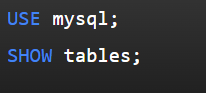


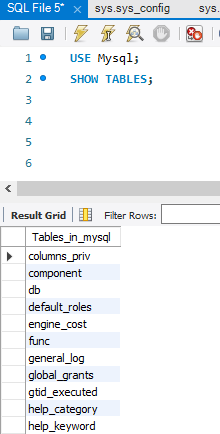
Veremos que apenas tiene una tabla visible, llamada **sys\_config**, con información sobre performance de carga. Si revisamos las Vistas y los Stored Procedures, encontraremos mucho más datos que aportan a la misma causa.



Además de **sys**, Mysql tiene otra base de datos de sistema llamada **mysql**. En la misma encontrarás otro set de tablas con información general para el sistema Mysql, más una tabla llamada **user**, donde se almacena la información de usuarios de la o las bases de datos de Mysql.

Puedes verificar esto mismo, directamente desde **Mysql Workbench**, abriendo una pestaña de script y escribiendo los comandos:

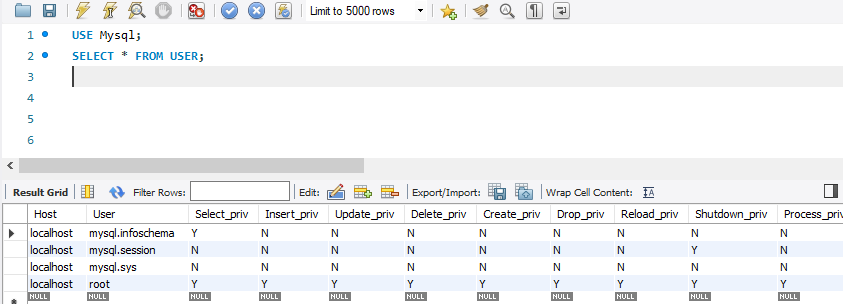




*LA TABLA mysql.user*

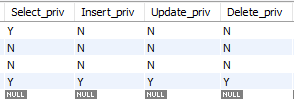
En **la tabla user**, Mysql almacena la información de usuarios junto a cada uno de los permisos o bloqueo para trabajar sobre la base de datos Mysql y los objetos de ésta

Si invocamos la misma mediante la instrucción **SELECT**, veremos información sobre el host o máquina, el usuario en cuestión, y la lista de permisos totales que podemos aplicar sobre cada usuario.

**

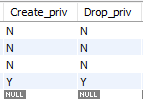
En esta imagen podemos ver los **Campos** que definen los permisos de usuario sobre

**DML**

**

*Cada permiso se define con un* ***Y*** *o* ***N****, si queremos brindarle al usuario Acceso o Restricción (respectivamente), sobre una acción determinada*

Además de **DML**, también encontramos las tablas que permiten o impiden al usuario realizar sentencias del **DDL**.



Y si recorremos el resto de la tabla, veremos en detalle que podemos tener un control específico sobre la ejecución, inserción, visualización y eliminación de objetos dentro de la o las bases de datos Mysql.

*GESTIONAR USUARIOS*

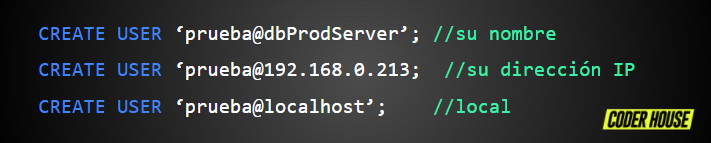
Veamos a continuación, cómo gestionar nuestros propios usuarios dentro de Mysql a través de los diferentes comandos que este motor de base de datos pone a nuestra disposición.

*CREATE USER*

A través de la sentencia **CREATE USER ‘nombre’**, creamos un nuevo usuario en la base de datos. Podemos definir solo su nombre, o el nombre de éste seguido del dominio al cual pertenece.

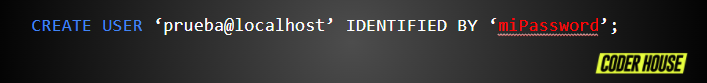
*CREATE USER + DOMINIO*

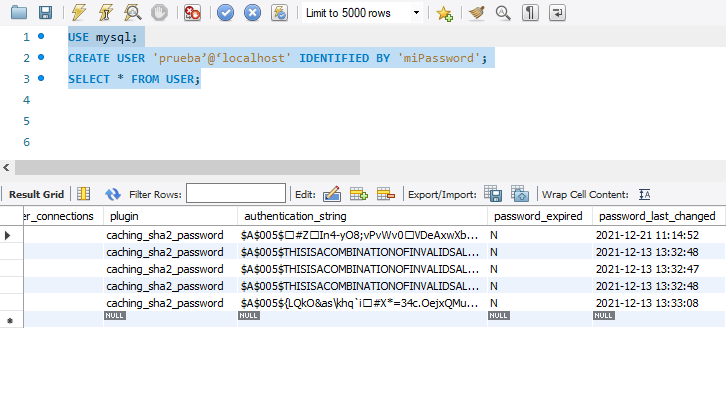
El dominio en sí hace referencia a la máquina o computadora donde se encuentra instalado Mysql. Podemos referirnos a ésta mediante **su nombre**, **IP**, o si es local, utilizando **localhost**.



*CREATE USER + DOMINIO + PASSWORD*

Si también deseamos especificar un password para el usuario en cuestión, podemos hacerlo incorporando a la sentencia, el comando **IDENTIFIED BY**. El password será visible en el comando en cuestión, pero se encriptará una vez almacenado en la tabla **user.**

****

****

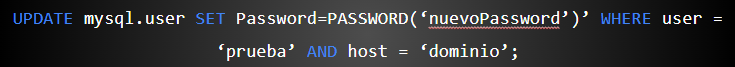
La columna **authentication\_string** permite validar que, el password ingresado, se almacena de manera encriptada.

*MODIFICAR UN PASSWORD*

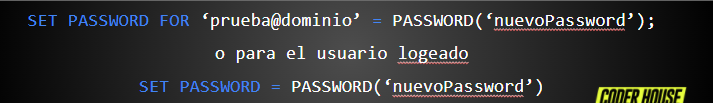
Podemos cambiar el password de un usuario, mediante el comando **ALTER USER:**

**

También podemos hacerlo mediante la sentencia **UPDATE:**

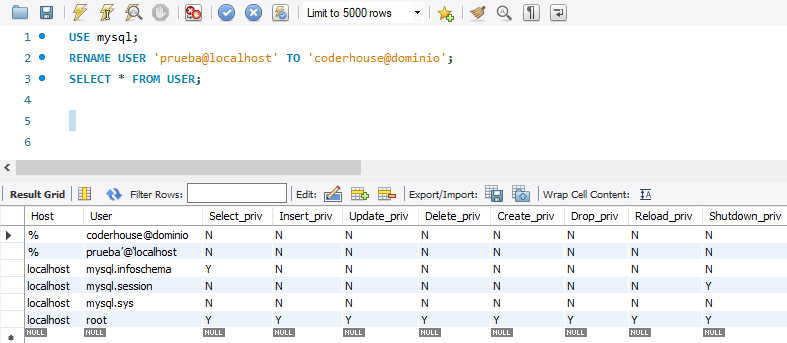
**

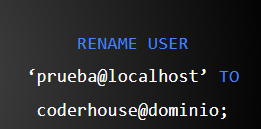
Si en algún momento debemos trabajar con una **base de datos** **Mysql v5.7.5 o inferior**, debemos utilizar el comando **SET PASSWORD.**

**

*RENOMBRAR Y ELIMINAR USUARIOS*

También podemos renombrar un usuario, una vez creado, utilizando el comando **RENAME USER**.

**

**

*ELIMINAR UN USUARIO*

Finalmente, si debemos eliminar un usuario, recurrimos al comando **DROP USER:**

**

*PERMISOS DEFINIDOS POR DEFECTO*

Por cada usuario que creamos, solo definimos un nombre de usuario y sobre qué SERVIDOR trabajará el mismo.

Nos queda por delante comenzar a definir los permisos que cada usuario tendrá, sobre un dominio, base de datos, tabla(s) y/o campo(s) específicos.

*¡Veamos entonces cómo hacerlo!*

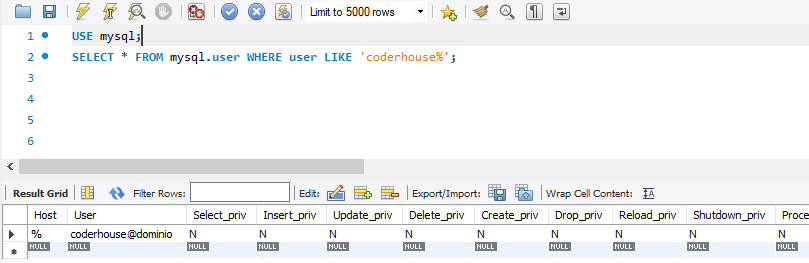
*VERIFICAR EL USUARIO CREADO*

Si queremos verificar los permisos de un usuario específico, podemos realizar una consulta de selección filtrando específicamente por el usuario en cuestión.

Ejecutemos para ello, la sentencia **SELECT** sobre la tabla **user**, filtrando por el usuario en particular:

**

Como podemos apreciar, en cada campo de la tabla, el parámetro correspondiente al permiso está seteado en **N**, lo cual indica que no tiene establecido ningún permiso.

**

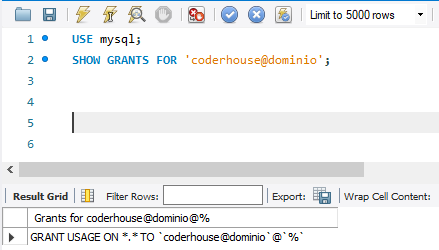
*ESTABLECER PERMISOS SOBRE OBJETOS MYSQL*

*SENTENCIA GRANT*

A través de sentencia **GRANT**, podemos definir en detalle, los permisos de escritura, modificación, eliminación, y/o lectura de datos y creación, alteración y borrado de objetos de la base de datos, para un usuario

**SHOW GRANTS** nos permite detallar sobre el usuario específico, cuáles permisos tiene asociado sobre los diferentes objetos de la DB.

**

**

Su resultado no detalla **ningún permiso** sobre sentencias DML, aún para el usuario seleccionado.

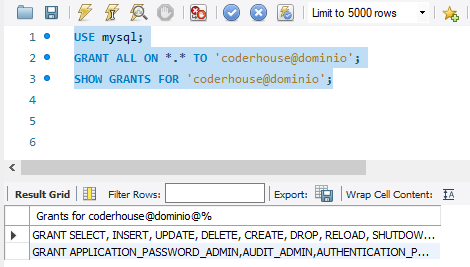
*Veamos entonces cómo comenzar a otorgarlos*.

*OTORGARLE TODOS LOS PERMISOS*

Para otorgarle permisos completos a un usuario, sobre todos los objetos de todos los dominios, utilizamos la sentencia **GRANT ALL**.

**

El uso de **\*.\*** refiere a **objetos.dominio**.

**

A través de la pestaña **Action Output** veremos la correcta aplicación de los permisos sobre el usuario referenciado.

*OTORGAR PERMISOS SOBRE TABLA(S)*

*OTORGAR PERMISOS SOBRE UNA TABLA*

Para otorgarle permisos a un usuario sobre una tabla específica de una base de datos puntual, debemos referenciar la sentencia de la siguiente forma:

**

*PERMISOS EN MÁS DE UNA TABLA*

Deberemos definir una línea específica por cada tabla en la que un usuario específico tendrá permisos:

**

*OTORGAR PERMISOS SELECTIVOS*

Si por ejemplo deseamos que un usuario tenga permisos selectivos referidos al DML sobre una tabla, debemos estructurar la cláusula de la siguiente forma:

**

*OTORGAR PERMISOS SELECTIVOS*

De esta manera, definimos que un usuario pueda realizar determinadas acciones sobre una tabla.

El usuario puede, por ejemplo sólo leer registros, agregar, o realizar todo tipo de operaciones menos eliminar.

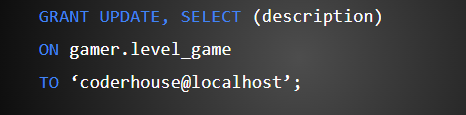
*Tengamos presente para esto, pensar bien la relación operativa entre las diferentes acciones, previo a establecer las mismas.*

*OTORGAR PERMISOS SOBRE COLUMNAS*

También podemos ir con más profundidad, estableciendo ciertos permisos sobre determinadas columnas de una tabla.

*Por ejemplo así podemos permitir que un usuario modifique solamente ciertos campos y evitar cambios sobre otros claves, como ser un documento de identidad o importes monetarios, entre otros*

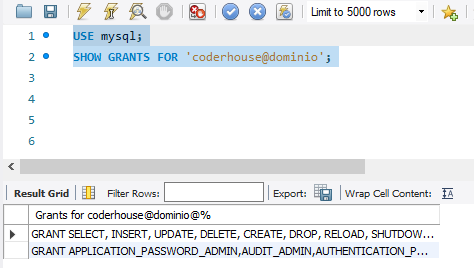
Para esto, debemos definir cada campo específico separado por una coma, tal como hacemos una consulta del tipo SELECT, definiendo qué campos visualizar.

**

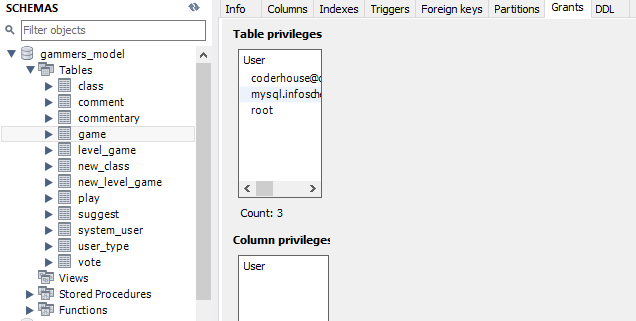
*VERIFICAR PERMISOS ESTABLECIDOS*

*PERMISOS ESTABLECIDOS*

Ejecutando nuevamente el comando **SHOW GRANTS** para el usuario en cuestión, podremos ver los diferentes permisos que le han sido otorgados al mismo.

**

También, accediendo a las propiedades de la tabla mediante **TABLE INSPECTOR**, podremos ver un detalle de los permisos por usuario establecidos.

**

*SENTENCIA REVOKE: QUITAR PERMISOS*

*SENTENCIA REVOKE*

Tal como existe una sentencia para otorgar diferentes permisos a un usuario de base de datos, también existe la sentencia que le quita dichos permisos. Se llama **REVOK**. y funciona de igual forma a **GRANT**, pero a la inversa.

*Veamos unos ejemplos a continuación…*

*QUITAR TODOS LOS PERMISOS*

Para quitarle todos los permisos a un usuario, sobre todos los objetos de todos los dominios, utilizamos la sentencia **REVOKE ALL**.

**

El uso de **\*.\*** refiere a **objetos.dominio**.

*QUITAR UN PERMISO DETERMINADO*

Para quitarle un permiso específico a un usuario, por ejemplo actualizar registros, sobre todos los objetos de todos los dominios, realizamos lo siguiente.

**