BBDD

Bases de Datos – Gestión de bases de datos

IES Ciudad Escolar
Desarrollo de Aplicaciones Multiplataforma

Objetivos

- Gestión de usuarios y contraseñas (creación, modificación, bloqueos y eliminación de usuarios)
- Gestión y asignación de privilegios y roles.
- Crear, modificar y eliminar objetos de la base de datos: vistas e índices.

MySQL®

Lenguaje SQL

El lenguaje de programación **SQL es el lenguaje fundamental de los SGBD** relacionales y los elementos que lo componen son:

a) **DML (*Data Manipulation Language*)**: es el lenguaje que consulta o manipula los datos ya existentes de nuestra BD.

b) **DDL (*Data Definition Language*)**: permite la **definición, modificación y eliminación de las estructuras básicas** (BD, tablas, vistas, etc.) en un SGBD.

c) **TCL (*Transaction Control Language*)**: lenguaje que controla el procesamiento de las transacciones de la BD.

d) **DCL (*Data Control Language*)**: administra a los usuarios de la BD, concediendo o denegando los permisos oportunos.

Vistas

¿Qué es una vista?

- Las **Vistas** son tablas virtuales sin contenido que devuelven las filas resultantes de una sentencia SELECT.
- Son más eficientes que ejecutar directamente una sentencia SELECT dado que sus sentencias SELECT ya están **compiladas** y **almacenadas** en la BD.
- Se permite leer los datos de una vista a pesar de tener **acceso restringido a las tablas consultadas desde dicha vista**. Además, se limitan los campos y filas a mostrar y se facilita el acceso a la información a usuario inexpertos que requieran de consultas SELECT complejas.
- Una vista puede estar formada por tablas, otras vistas, subconsultas y todo tipo de joins.
- Para que sean actualizables o soporten nuevos registros, las vistas deben cumplir ciertas condiciones (como estar formada por una única tabla y no tener funciones de agrupación). Se debe tener en cuenta que los posibles cambios **afectarán a las tablas consultadas**.

CREATE VIEW <nom_vista> (campo1, campo2, …) AS <sentencia SELECT>;

Nota: Si no indicamos el nombre de las columnas, cada columna de la vista tendrá el mismo nombre que las columnas de la sentencia SELECT.

DROP VIEW [IF EXISTS] <nom _vista>;

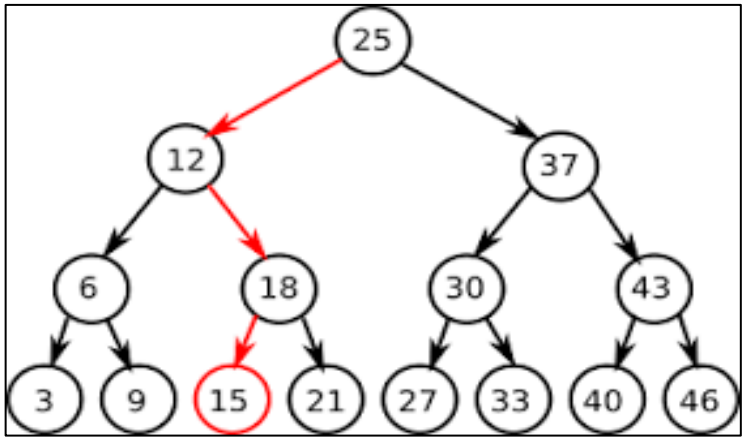
ALTER VIEW <nom _vista> (campo1, campo2, …) AS <sentencia SELECT>;

Índices

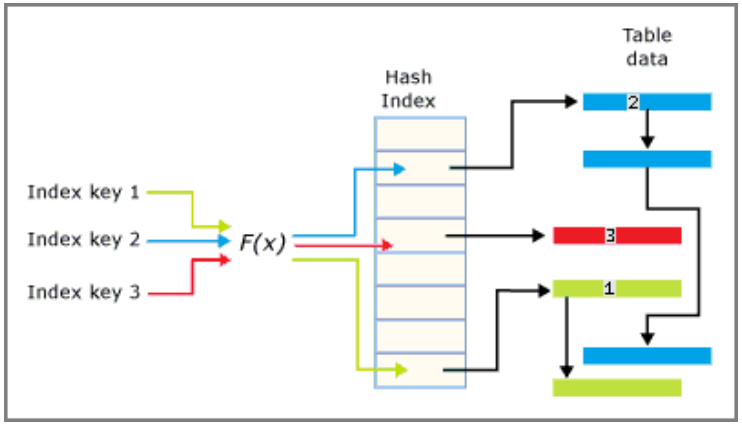
¿Qué es un índice?

- Los **Índices** son recursos asociados a las tablas que ayudan a indexar el contenido de las columnas para facilitar la búsquedas de contenido cuando se ejecutan consultas sobre esas tablas.
- Buscar un valor específico en la columna *no indexada* de una tabla supone recorrer toda la tabla comparando registro a registro hasta encontrar el valor que coincida con el valor buscado.
- Para tablas con pocas filas no es un problema, pero cuando el número de filas es muy elevado, las operaciones de búsqueda se hacen eternas y consumen muchos recursos.
- Cuando marcamos uno o varios campos de una tabla como **PRIMARY KEY** (PK) se crea un Índice cuyo valor es único y no puede ser NULL.
- Cuando marcamos uno o varios campos de una tabla con la restricción **UNIQUE** también se crea un índice asociado a esos campos automáticamente.

Tipos de índices



Arboles b



Hash

Índices (InnoDB)

- Si queremos añadir un Índice *ordinario* que permita duplicados en uno o varios campos de una tabla, podemos usar cualquiera de las dos siguientes sentencias:

```
ALTER TABLE nombre_tabla ADD INDEX idx_nombre(col);
CREATE [UNIQUE] INDEX idx_nombre ON nombre_tabla (col1,…);
```

- Si queremos eliminar un Índice ordinario, debemos modificar la tabla asociada mediante la sentencia:

```
ALTER TABLE nombre_tabla DROP INDEX idx_nombre;
DROP INDEX idx_nombre ON nombre_tabla;
```

- Para ver los índices de una determinada tabla podemos ejecutar la sentencia:

```
SHOW INDEXES FROM nombre_tabla;
```

La creación de índices aunque mejora los tiempos de respuesta en las sentencias SELECT, **sobrecarga las sentencias INSERT, DELETE y UPDATE** sobre la tabla relacionada dado que los índices deben actualizarse tras cada una de esas sentencia.

Optimización de consultas (explain)

- Las consultas que realizamos contra una base de datos pueden ser más o menos eficientes (buscamos minimizar el tiempo de respuesta) en función de qué condiciones establezcamos.
- Un mecanismo para evaluar cómo de buena es una sentencia de SQL, es utilizar la sentencia **EXPLAIN.**
- Con EXPLAIN podremos recuperar cómo ejecuta una determinada sentencia SELECT, DELETE, INSERT o UPDATE el SGBD.

```
EXPLAIN SELECT … FROM nombre_tabla …;
EXPLAIN UPDATE nombre_tabla SET …;
EXPLAIN DELETE FROM …;
EXPLAIN INSERT INTO …;
```

- La sentencia EXPLAIN devuelve una tabla con una serie de filas con información sobre cada una de las tablas empleadas en la consulta a la que acompaña. Los campos más representativos para ver la eficiencia de la sentencia evaluada son:
    - **possible_keys**: Posibles índices para localizar las filas de la tabla. Si esta columna es NULL, significa que no hay índices que puedan ser usados para mejorar la consulta.
    - **key**: índice que MySQL ha decido usar para ejecutar la consulta. Es posible que el nombre del índice no esté presente en la lista de possible_keys.
    - **rows**: estimación del número de filas que MySQL examinará para localizar las filas de la tabla.

Lenguaje SQL

El lenguaje de programación **SQL es el lenguaje fundamental de los SGBD** relacionales y los elementos que lo componen son:

a) **DML (*Data Manipulation Language*)**: es el lenguaje que consulta o manipula los datos ya existentes de nuestra BD.

b) **DDL (*Data Definition Language*)**: permite la **definición, modificación y eliminación de las estructuras básicas** (BD, tablas, vistas, etc.) en un SGBD.

c) **TCL (*Transaction Control Language*)**: lenguaje que controla el procesamiento de las transacciones de la BD.

d) **DCL (*Data Control Language*)**: administra a los usuarios de la BD, concediendo o denegando los permisos oportunos.

DCL (Data Control Language)

DCL (Data Control Language) o Lenguaje de Control de Datos es un lenguaje proporcionado por el SGBD que incluye una serie de comandos SQL que permiten al administrador **controlar el acceso a los datos** contenidos en la base de datos.

Las sentencias DCL por excelencia son:

- `GRANT`: se utiliza para otorgar privilegios y roles a usuarios y roles.
- `REVOKE`: se utiliza para quitar privilegios y roles a usuarios y roles.

La sintaxis de dichas sentencias es:

- Para dar permisos a un usuario: GRANT <Lista privilegios> ON <elementos> TO <usuario>;
- Para visualizar los permisos (o roles) de un usuario: SHOW GRANTS FOR <usuario>;
- Para quitar permisos a un usuario: REVOKE <Lista privilegios> ON <elementos> FROM <usuario>;

https://dev.mysql.com/doc/refman/8.0/en/grant.html
https://dev.mysql.com/doc/refman/8.0/en/revoke.html

Privilegios o Permisos

En general, en MySQL diferenciamos dos grandes grupos de privilegios:
1. **Global Privileges**: Privilegios administrativos del servidor que aplican a todas las bases de datos.
2. **Database Privileges**: Privilegios que aplican a una base de datos específica

Los privilegios de MySQL más comunes se suelen agrupar de la siguiente manera:

1. Privilegios sobre objetos (bases de datos, tablas, vistas, rutinas…)
   a. SELECT - permite consultar registros de una tabla o vista
   b. INSERT - permite insertar registros en tablas
   c. UPDATE - permite modificar registros en tablas
   d. DELETE - permite borrar registros de tablas (necesario también para hacer TRUNCATE)
   e. EXECUTE - permite ejecutar un procedimiento almacenado o función
   f. SHOW VIEW - permite ver la consulta con la que se creó la vista (SHOW CREATE VIEW)
   g. LOCK TABLES - permite bloquear tablas completas durante la gestión de concurrencia (precisa del permiso SELECT).

SHOW privileges;

2. Privilegios DDL (más comunes)
   a. CREATE - permite crear bases de datos, tablas, vistas, etc
   b. ALTER - permite modificar la estructura de una tabla o modificar su nombre (precisa de los permisos INSERT y CREATE)
   c. INDEX - permite crear o borrar índices
   d. CREATE VIEW - permite crear vistas
   e. CREATE ROUTINE - permite crear funciones y procedimientos almacenados
   f. ALTER ROUTINE - permite modificar funciones y procedimientos almacenados
   g. DROP - permite borrar bases de datos, tablas, vistas, etc.
   h. TRIGGER - permite crear y borrar Triggers en una tabla

3. Otros privilegios (más comunes)
   a. GRANT OPTION - permite dar o quitar los permisos propios a otros usuarios (SOLO los propios)
   b. CREATE ROLE & DROP ROLE – permite crear o borrar un rol.
   c. CREATE USER – permite crear un usuario.

https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html

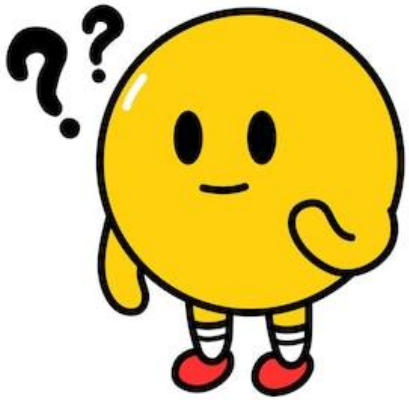
Privilegios ¿Cuándo comienzan a aplicar?

Para añadir, quitar o modificar los permisos en MySQL existen dos procedimientos:

1. Realizar los cambios mediante comandos (GRANT, REVOKE, SET PASSWORD o RENAME USER). Esta es la opción más recomendable y con la que comienzan a aplicar automáticamente.

2. Realizar los cambios mediante sentencias SQL (INSERT, UPDATE o DELETE) directamente sobre la tabla *user* o *db* de la BD **mysql**. Esta opción no se recomienda y exige informar al SGBD que recargue en memoria la información de privilegios. Para ello, se precisa reiniciar el servidor o bien ejecutar la sentencia FLUSH PRIVILEGES.

```
select * from mysql.user;
select * from mysql.db;
```



Roles

- Los **roles** son colecciones de permisos o privilegios a las que se les asigna un nombre.
- Los privilegios o permisos se asignan a los roles igual que se asignan a las cuentas de usuario.
- Cuando se asigna un rol a un usuario, el usuario pasa a tener los privilegios del correspondiente rol.
- Cuando tenemos varios usuarios que deben compartir los mismos privilegios, crear un rol es la opción más eficiente al poder asignar de una vez un conjunto de privilegios además de evitar posibles descuidos (añadir de más o de menos privilegios)
- Las sentencias más comunes de MySQL para la gestión de roles son:
  - Para crear un rol: CREATE ROLE <nombre_del_rol>;
  - Para borrar un rol: DROP ROLE <nombre_del_rol>;
  - Para añadir privilegios a un rol existente: GRANT <Lista privilegios> ON <recursos> TO <nombre_del_rol>;
  - Para quitar privilegios a un rol existente: REVOKE <Lista privilegios> ON < recursos> FROM <nombre_del_rol>;
  - Para asignar un rol a un usuario: GRANT <nombre_del_rol> TO <usuario>;
  - Para que un rol asignado a un usuario aplique por defecto: SET DEFAULT ROLE <nombre_del_rol> TO <usuario>;
  - Para que un usuario seleccione uno de sus roles: SET ROLE <nombre_del_rol>
  - Para fijar un rol por defecto a un usuario: SET DEFAULT ROLE <nombre_del_rol>
  - Para que un usuario sepa el rol que le está aplicando: SELECT CURRENT_ROLE();
  - Para quitar un rol a un usuario: REVOKE <nombre_del_rol> FROM <usuario>;
  - Para visualizar los privilegios de un rol: SHOW GRANTS FOR <nombre_del_rol>;
  - Para visualizar los roles asignados a un usuario: SHOW GRANTS FOR <usuario>;

recursos: bases de datos, tablas, vistas… (se pueden usar comodines (*) para seleccionar todos los elementos de una BD

Usuarios y contraseñas

- Las gestión de usuarios en MySQL se lleva a cabo mediante las siguientes sentencias:

  - Para crear un usuario:

    CREATE USER [IF NOT EXISTS] <usuario>@<ubicación> IDENTIFIED BY <contraseña>;

  - Para borrar un usuario:

    DROP USER [IF NOT EXISTS] <usuario>@<ubicación>;

  - Para bloquear o desbloquear a un usuario:

    ALTER USER <usuario>@<ubicación> ACCOUNT LOCK;

    ALTER USER <usuario>@<ubicación> ACCOUNT UNLOCK;

  - Para renombrar a un usuario:

    RENAME USER <usuario>@<ubicación> TO <usuario2>@<ubicación2>;

  - Para cambiar la contraseña de un usuario:

    ALTER USER <usuario>@<ubicación> IDENTIFIED BY <nueva_contraseña>;