

T3: CONFIDENCIALIDAD y SEGURIDAD

Introducción

+ Consideraciones generales de seguridad

- ↳ Técnicas generales: buenas prácticas, permisos, prevenir SQL inyecciones,...
- ↳ Seguridad en la instalación: ficheros de datos, log,...
- ↳ Control de acceso y seguridad (dentro ): usuarios, tablas, vistas,...
- ↳ Seguridad de la red: restricción de acceso por ubicación al cliente
- ↳ Copia de seguridad: datos, ficheros de config y logs, recuperación errores

+ Consejos de Seguridad en MySQL

- 1) NO dar acceso a la tabla `user` (excepto root)
- 2) NO conceder más privilegios de los necesarios
- 3) Nunca conceder privilegios a todos los hosts
- 4) GRANT 2 REVOKE → SHOW GRANTS ver q cuentas tienen acceso sobre qué
- 5) NO almacenar contraseñas → hash claves
User fun: hash unidireccionales: `SHA2()`, `SHA1()`, `MD5()` y guardar valor hash
- 6) NO utilizar contraseñas con palabras de diccionario
- 7) Utilizar firewalls
- 8) Cifrar la info.

+ Medidas de Seguridad en MySQL

- ↳ Listas de control de Acceso (ACls - Access Control List)
- ↳ Conexiones seguras con SSL (Secure Sockets Layer)
entre servidores → clientes
- ↳ cliente → contraseña → debe coincidir el hash con el valor almacenado en la tabla `user`

Protegiendo al ROOT

+ Usuarios anónimos (no recomendados)

↳ no suelen tener ni usuarios ni contraseñas

+ Comprobando cuentas `Empty` (donde si estén)

```
mysql> SELECT User, Host, HEX(authentication_string)as Password FROM mysql.user;
```

```
mysql> SELECT User, Host, Password FROM mysql.user; (antes v5.7.6)
```

| User | Host | Password |
|------|-----------|---|
| root | localhost | SASQHISTHSACQHNMTHQINVALISALANDPSSQHHTHHSNEVERBRESEED |

Si `password` está vacío:

- Asignar contraseñas para `root`
- Asignar contraseñas para `usuarios anónimos`

+ Cambiando contraseña al `root`

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

```
mysql> SET PASSWORD [FOR 'root'@'localhost'] = 'new_password';
```

+ Cambiando contraseña a `Cuentas anónimas`

```
mysql> ALTER USER ''@'localhost' IDENTIFIED BY 'new_password';
```

```
mysql> SET PASSWORD [FOR ''@'localhost'] = 'new_password';
```

Sintaxis para nombres de usuario: 'Palabra'@'localhost'

Se permite el uso de comodines '%' y '_' en el host_name

'pepito' = pepito@%
'pepito'@localhost'
'pepito@localhost'
'pepito'@%.miweb.com'
'pepito'@192.168.1._'

User_name

- Cadena no vacía

- '%'@'localhost' → usuario anónimo

Host_name

- Nombre de dominio (www.web.es)

- IP4 ó IP6 (192.168.1.1 ó ::1)

- IP + máscara (192.168.1.0/255.255.255.0)

| Campo | Largo máxima (caracteres) |
|--------------|---------------------------|
| host | 60 255 |
| user | 32 (hasta v5.7.8) |
| db | 64 |
| table_name | 64 |
| column_name | 64 |
| routine_name | 64 |

Gestión de usuarios

⇒ **F(x) principal**: autenticar usuario y asociar privilegios

⇒ con el est. de privilegios de MySQL no se puede...

+ Especificar explícitamente → usuario no tiene permiso/acceso

↳ solo almacenar lo que se le permite

+ Especifica → crear y borrar tablas ↳ No puede < borrar 

+ Contraseñas no se aplican globalmente a una cuenta

⇒ Sentencias SQL Privilegios: CREATE USER, GRANT, y REVOKE

⇒ Sistemas de privilegios de acceso de MySQL:

+ (internamente)  almacena info priv. re l tablas renombradas 

+ Tabla leída → almacena en memoria los privilegios que el arrancar 

+ identidad usuario del host que realiza la conexión

+ ver privilegios:

```
SHOW GRANTS FOR user; (** sintaxis ***)
```

```
SHOW GRANTS FOR 'root'@'localhost';
```

⇒ Sist de control MySQL → 2 etapas

1. Servidor → acepta/rechaza conexiones
2. (si conexión) → verifica toda sentencia
(los privilegios usuario)

identidad
contraseña
y cuenta (activa o bloq.)

Cambios Privilegios

Efecto inmediato: GRANT, REVOKE, SET PASSWORD, RENAME USER

Reinicio servidor: INSERT, UPDATE, DELETE
e recarguen Priv ↳ FLUSH PRIVILEGES

Tablas con información sobre permisos

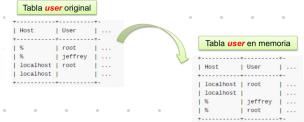
| | |
|---------------------------|---|
| <code>user</code> | Cuentas de usuario, privilegios globales y otras columnas |
| <code>db</code> | Privilegios a nivel de base de datos |
| <code>tables_priv</code> | Privilegios a nivel de tabla |
| <code>columns_priv</code> | Privilegios a nivel de columna (campo) |
| <code>procs_priv</code> | Privilegios sobre procedimientos almacenados y funciones |

T3: CONFIDENCIALIDAD y SEGURIDAD

(Gestión de usuarios)

Verificación de conexiones

servidor carga → Memoria: tabla user
[Tenería Filas]



Creación de cuentas de usuarios

1. sentencias (preferencia)
 - CREATE_USER
 - GRANT

2. Manipulando tablas de permisos
 - INSERT, UPDATE & DELETE

```
mysql> CREATE USER 'francis'@'localhost' IDENTIFIED BY 'frank';
--> GRANT ALL PRIVILEGES ON *.* TO 'francis'@'localhost' WITH GRANT OPTION;
mysql> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
--> GRANT RELOAD, PROCESS ON *.* TO 'admin'@'localhost';
mysql> CREATE USER 'jeffrey'@'localhost';
--> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
--> ON bankaccount
--> TO 'jeffrey'@'localhost';
```

Borrado de cuentas de usuarios:

```
mysql> DROP USER IF EXISTS user_1;
mysql> DROP USER 'jean'@'localhost';
```

Renombrar cuentas

```
mysql> RENAME USER old_user TO new_user; -- old_user TO new_user;
```

privilegios a nivel de tabla (db-tabla)

- tabla especificada
- tabla afectada: mysql.tabla_priv

```
mysql> GRANT ALL ON mydb.tabla TO 'someuser'@'somehost';
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON mydb.tabla TO 'someuser'@'somehost';
```

Privilegios a nivel de campo/columna)

- privilegio (columna) en db-tabla)
- tabla afectada: mysql.columns_priv

```
mysql> GRANT SELECT (col1, INSERT (col1, col2)
--> ON mydb.tabla TO 'someuser'@'somehost';
```

Permisos Posibles para GRANT y REVOKE

- **ALL PRIVILEGES** → proporciona todos los permisos excepto GRANT OPTION
- **ALTER** → permite el uso de ALTER TABLE
- **ALTER ROUTINE** → permite modificar o borrar procedimientos almacenados
- **CREATE** → permite el uso de CREATE TABLE
- **CREATE ROUTINE** → permite crear procedimientos almacenados
- **CREATE TEMPORARY TABLES** → permite el uso de CREATE TEMPORARY TABLE
- **CREATE USER** → permite el uso de CREATE USER, DROP USER, RENAME USER y REVOKE ALL PRIVILEGES
- **CREATE VIEW** → permite crear y modificar vistas
- **DELETE** → permite el uso de DELETE
- **DROP** → permite eliminar bases de datos, tablas y vistas
- **EVENT** → permite el uso del programador de eventos (Event Scheduler)
- **EXECUTE** → permite ejecutar procedimientos almacenados
- **FILE** → permite al usuario leer y/o escribir ficheros en el servidor
- **GRANT OPTION** → permite poner y/o quitar permisos de otras cuentas de usuario
- **INDEX** → permite crear y borrar índices
- **INSERT** → permite el uso de INSERT
- **LOCK TABLES** → permite el uso de LOCK TABLES en tablas para las que se tenga permiso de SELECT
- **PROCESS** → permite al usuario ver todos los procesos con SHOW PROCESSLIST
- **REFERENCES** → permite la creación de claves foráneas
- **RELOAD** → permite el uso de las operaciones FLUSH
- **REPLICATION CLIENT** → permite al usuario preguntar dónde están los servidores maestro o esclavo
- **REPLICATION SLAVE** → permite a los esclavos de replicación leer el fichero binario de log del maestro
- **SELECT** → permite el uso de SELECT
- **SHOW DATABASES** → permite el uso de SHOW DATABASES
- **SHOW VIEW** → permite el uso de SHOW CREATE VIEW
- **SHUTDOWN** → permite el uso de "mysqld shutdown" (parar el servidor)
- **SUPER** → permite el uso de otras operaciones administrativas como CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL y el comando "mysqldadmin debug"
- **TRIGGER** → permite operaciones con TRIGGERS (disparadores)
- **UPDATE** → permite el uso de UPDATE
- **USAGE** → sinónimo de "no privileges"

Limitar recursos asignados al usuario

- ↳ MySQL **permite** establecer límites uso recursos servidor

- Num consultas de cuenta por hora
- Num actualizaciones de cuenta por hora
- Num veces q una cuenta **conectarse** servidor por hora
- Num de conexiones simultáneas al servidor desde una cuenta

- ↳ Se puede establecer en la Creación:

```
mysql> CREATE USER 'francis'@'localhost' IDENTIFIED BY 'frank'
--> WITH MAX_QUERIES_PER_HOUR 20
--> MAX_UPDATES_PER_HOUR 10
--> MAX_CONNECTIONS_PER_HOUR 5
--> MAX_USER_CONNECTIONS 2;
```

- ↳ Modificar después:

```
mysql> ALTER USER 'francis'@'localhost' WITH MAX_QUERIES_PER_HOUR 100;
mysql> ALTER USER 'francis'@'localhost' WITH MAX_CONNECTIONS_PER_HOUR 10;
```

Bloquear cuentas de usuarios

ACCOUNT LOCK | UNLOCK

durante creación → AFTER

```
mysql> CREATE USER 'someuser'@'somehost' IDENTIFIED BY 'pass' ACCOUNT LOCK;
```

```
mysql> ALTER USER 'someuser1'@'somehost1' ACCOUNT LOCK;
mysql> ALTER USER 'someuser2'@'somehost2' ACCOUNT UNLOCK;
```

Eliminación de permisos

```
mysql> REVOKE priv_type ((column_list)), priv_type ((column_list))
--> ON (object_type priv_level
--> FROM user_or_role , user_or_role);
mysql> REVOKE ALL PRIVILEGES , GRANT OPTION
--> FROM user_or_role , user_or_role;
```

Ejemplos:

```
mysql> REVOKE INSERT ON *.* FROM 'jeffrey'@'localhost';
mysql> REVOKE 'role1', 'role2' FROM 'user1'@'localhost', user2;
mysql> REVOKE SELECT ON world.* FROM 'role3';
```

```
mysql> REVOKE ALL PRIVILEGES, GRANT OPTION
--> FROM 'jeffrey'@'localhost', 'someuser'@'somehost';
```

Privilegios Globales (*.*)

- todos los privilegios de todas las BBDD del servidor
- tabla afectada: mysql.user

```
mysql> GRANT ALL ON *.* TO 'someuser'@'somehost';
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
```

Privilegios a nivel BD (db.*)

- tablas 1 BD → se usa sólo * no Priv: BD actual
- tabla afectada: mysql.db

```
mysql> GRANT ALL ON mydb.* TO 'someuser'@'somehost';
mysql> GRANT SELECT, INSERT ON mydb.* TO 'someuser'@'somehost';
```

T3: CONFIDENCIALIDAD y SEGURIDAD

(Gestión de usuarios)

Roles → colección de priv. identif. con un nombre

• un rol → revocar conceder > priv.

• un usuario ↔ 1:N roles

```
mysql> CREATE ROLE [IF NOT EXISTS] role [, role] ...
```

```
mysql> CREATE ROLE 'app_developer', 'app_read', 'app_write';
mysql> CREATE ROLE 'webapp'@'localhost';
```

```
mysql> GRANT ALL ON app_db.* TO 'app_developer';
mysql> GRANT SELECT ON app_db.* TO 'app_read';
mysql> GRANT INSERT, UPDATE, DELETE ON app_db.* TO 'app_write';
```

Asignación roles → usuarios

```
mysql> CREATE USER 'dev1'@'localhost' IDENTIFIED BY 'devipass';
mysql> CREATE USER 'read_user1'@'localhost' IDENTIFIED BY 'read_user1pass';
mysql> CREATE USER 'read_user2'@'localhost' IDENTIFIED BY 'read_user2pass';
mysql> CREATE USER 'rw_user1'@'localhost' IDENTIFIED BY 'rw_user1pass';
```

```
mysql> GRANT 'app_developer' TO 'dev1'@'localhost';
mysql> GRANT 'app_read' TO 'read_user1'@'localhost',
       'read_user2'@'localhost';
mysql> GRANT 'app_write' TO 'rw_user1'@'localhost';
```

Comprobación → priv. usando roles

```
mysql> SHOW GRANTS FOR 'dev1'@'localhost';
```

```
mysql> SHOW GRANTS FOR 'dev1'@'localhost' USING 'app_developer';
mysql> SHOW GRANTS FOR 'read_user1'@'localhost' USING 'app_read';
mysql> SHOW GRANTS FOR 'rw_user1'@'localhost' USING 'app_write', 'app_write';
```

Quitar no priv a los roles

```
mysql> REVOKE role FROM user;
```

```
mysql> REVOKE INSERT, UPDATE, DELETE ON app_db.* FROM 'app_write';
```

Eliminar roles

```
mysql> DROP ROLE 'app_read', 'app_write';
```

Contraseñas

• Cambiar contraseña de un usuario ↗

a) Con **ALTER USER** (recomendada)

```
mysql> ALTER USER 'someuser'@'somehost' IDENTIFIED BY 'newpassword';
```

• Si se está conectado con una cuenta no anónima, cambiar la contraseña propia sería:

```
mysql> ALTER USER USER() IDENTIFIED BY 'password';
```

b) Con **SET PASSWORD**

```
mysql> SET PASSWORD [FOR user] = 'newpassword'
      [REPLACE 'current_password']
      [RETAIN CURRENT_PASSWORD]
```

```
mysql> SET PASSWORD FOR 'jeffrey'@'localhost' = 'newpassword';
```

```
mysql> SET PASSWORD = 'newpassword';
```

c) Con **mysqladmin** desde la línea de comandos del SO:

```
shell> mysqladmin -u user_name -h host_name -p password "newpassword"
```

↳ Capacidades de gestión de contraseñas → ? MySQL

- Extracción contraseñas
- Restricciones en la reutilización de contraseñas
- Cambio de contraseña → Previa verificación de la actual
- Contraseña durales
- Comprobación / validación → fortalecer contraseñas

Cambiar contraseña → Versiones anteriores

- Con **SET PASSWORD** y la función **PASSWORD()**

```
mysql> SET PASSWORD FOR 'jeffrey'@'localhost' = PASSWORD('newpassword');
mysql> SET PASSWORD = PASSWORD('newpassword');
mysql> SET PASSWORD FOR 'jeffrey'@'localhost' = ...          (antes de la v5.7.6)
       ''4C8989346EAE75B8670AD8EA7A7FC116A935EF74'
```

- Con **GRANT USAGE**: (deprendido)

```
mysql> GRANT USAGE ON .* TO 'someuser'@'somehost'
      IDENTIFIED BY 'newpassword';
```

- Con **UPDATE**:

```
shell> mysql -u root -p mysql
mysql> UPDATE user SET authentication_string = PASSWORD('newpassword')
      WHERE host = 'somehost' AND user = 'someuser';
mysql> FLUSH PRIVILEGES;
```

Cifrado de información

info → guardadas → cifrada → ⇒ → fin cifrado (reversibles irreversibles)

| Función | Descripción |
|--|---|
| AES_ENCRYPT(string, clave, vector_inicial) | Cifra un string utilizando AES (Advanced Encryption Standard) |
| AES_DECRYPT(string, clave, vector_inicial) | Decifra un string cifrado previamente con AES |
| COMPRESS(string) | Devuelve un string binario como resultado de la compresión |
| UNCOMPRESS(string) | Descomprime un string previamente comprimido |
| MD5(string) | Calcula un resumen (checksum) utilizando MD5 128-bit (32 hex digits) |
| PASSWORD(string) (eliminada desde 8.0.11) | Calcula y devuelve un password cifrado mediante una función hash |
| SHA1(string), SHA(string) | Calcula un resumen (checksum) utilizando SHA1 - SHA 160-bit (40 hex digits) |
| SHA2(string, longitud) | Calcula un resumen (checksum) utilizando SHA2 del tamaño en bits indicado por longitud. Valores posibles: 224, 256, 384, 512 (o que equivale a 256) |
| RANDOM_BYTES(longitud) | Devuelve un string binario de longitud (1-1024) bytes |

Vistas

consultas almacenadas

CREATE | ALTER | DROP VIEW

```
CREATE [OR REPLACE]
       [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
       [DEFINER = user]
       [SQL SECURITY { DEFINER | INVOKER }]
VIEW view_name [(column_list)] AS select_statement
      [WITH [CASCADED | LOCAL] CHECK OPTION]
```

```
mysql> CREATE TABLE t (qty INT, price INT);
mysql> INSERT INTO t VALUES (3, 50), (5, 60);
mysql> CREATE VIEW v AS SELECT qty, price, qty*price AS value FROM t;
mysql> SELECT * FROM v;
+---+---+-----+
| qty | price | value |
+---+---+-----+
| 3   | 50    | 150   |
| 5   | 60    | 300   |
+---+---+-----+
mysql> SELECT * FROM v WHERE qty = 5;
+---+---+-----+
| qty | price | value |
+---+---+-----+
| 5   | 60    | 300   |
+---+---+-----+
```

↳ No expone tablas direct. a los usuarios (seguridad ✓)

↳ Almacena en el servidor → ✓ consumo recursos y ↑ eficiencia

↳ ✓ Guardar consultas complejas

↳ Las vistas no admiten parámetros