1.1.1 Xestión de usuarios dunha base de datos

1.1.1.1 Introdución

Unha das tarefas principais dun administrador de bases de datos é a xestión de usuarios.

Tódolos accesos a unha base de datos requiren a conexión mediante un usuario. Os usuarios teñen asignados unha serie de privilexios que son os que lles dan permiso de uso de certos obxectos da base de datos. Desta maneira todo usuario terá dereito a utilizar certos obxectos da base de datos e terá restrinxido o uso doutros.

Para unha maior simplicidade, a maioría de Sistemas Xestores de Bases de Datos permiten agrupar os permisos que se lles poden aplicar aos usuarios nunhas estruturas chamadas perfís e roles, que en definitiva son un conxunto de permisos.

Deste xeito, cando un usuario quere conectarse a unha base de datos, primeiro debe autentificarse, é dicir, probar que é quen di ser (normalmente mediante un contrasinal). Dita autentificación terá asociados uns privilexios (uns dereitos) e unhas restricións.

Polo tanto, será responsabilidade do administrador a creación de usuarios e a asignación aos mesmos dos distintos roles e privilexios que lles permitan desenvolver a súa actividade sen poñer en compromiso a seguridade da base de datos.

De forma xeral, non é unha boa práctica deixar que todos os usuario con acceso ao servidor teñan todos os privilexios. Para conservar a integridade dos datos e das estruturas, será conveniente que só algúns usuarios poidan realizar determinadas tarefas, e que outras, que requiran maior coñecemento sobre as estruturas de bases de datos e táboas, só poidan realizarse por un número limitado e controlado de usuarios.

1.1.1.2 Xestión de usuarios en MySQL

Niveles de privilexios

En MySQL existen cinco niveles distintos de privilexios:

- **Globais**: aplícanse ao conxunto de todas as bases de datos nun servidor. É o nivel máis alto de privilexios, no sentido de que o seu ámbito é o máis xeral.
- **De base de datos**: refírense a bases de datos individuais, e por extensión, a todos os obxectos que contén cada base de datos.
- **De táboa**: aplícanse a táboas individuais, e polo tanto, a todas as columnas desas táboa.
- **De columna**: aplícanse a unha columna nunha táboa concreta.
- **De rutina**: aplícanse aos procedementos almacenados.

Creación de usuarios

Aínda que a partir da versión 5.0.2 de MySQL existe unha sentenza para crear usuarios, CREATE USER, en versións anteriores úsase exclusivamente a sentenza GRANT para crealos.

A sintaxe de CREATE USER é:

```
CREATE USER usuario [IDENTIFIED BY [PASSWORD] 'contrasinal']
                    [, usuario [IDENTIFIED BY [PASSWORD] 'contrasinal']] ...
```

Usando GRANT pódese crear un usuario e ao mesmo tempo concederlle tamén os privilexios que terá, aínda que nas últimas versións de MySQL a recomendación é crear primeiro o novo usuario con CREATE USER e logo usar GRANT para darlle privilexios.

A sintaxe simplificada para usar GRANT é:

```
GRANT    tipo_privilexio [(lista_columnas)] [, tipo_privilexio [(lista_columnas)]] ...
```

```
ON  obxecto  TO   nome_usuario[ @equipo ] [ IDENTIFIED BY 'contrasinal' ]
[WITH GRANT OPTION]
```

- **tipo_privilexio**, representa os privilexios que se lle poden conceder aos usuarios, e dicir, o que se lle vai a permitir facer cos obxectos do servidor. A orden **show privileges** permite ver todos os tipos de privilexios posibles. Algúns dos máis utilizados son:
  - ALL [PRIVILEGES] Todos os privilexios, excepto GRANT OPTION
  - ALTER Modificar obxectos coa orden ALTER (táboas)
  - CREATE Crear obxectos coa orden CREATE (bases de datos ou táboas)
  - CREATE VIEW Crear vistas
  - DROP Borrar táboas con DROP TABLE
  - EXECUTE Executar procedementos almacenados
  - SELECT Facer consultas con SELECT
  - UPDATE Modificar datos das táboas
  - USAGE Sinónimo de 'sen privilexios'
  - GRANT OPTION Conceder privilexios a outros usuarios
- No caso de utilizar a opción WITH GRANT OPTION, se lle está dando ao usuario a posibilidade de ceder a outros usuarios os privilexios que se lle conceden a el.
- **obxecto**, representa sobre que cousas se conceden, ou retiran, os privilexios. Os máis utilizados son:
  - *.* Todas as táboas de todas a bases de datos
  - * Todas as táboas da base de datos activa
  - nome _bd.* Todas as táboas da base de datos nome_bd
  - nome_db.nome_táboa A táboa especificada da base de datos nome_bd

  Tamén se poden conceder ou retirar privilexios sobre funcións ou procedementos almacenados.
- **nome_usuario**, representa o usuario ao que se conceden os permisos. O nome do usuario é unha cadea de carácteres que só debe levar letras, números, e o guión baixo (_).
- **equipo**, pode ser o nome dun equipo, unha dirección IP, ou ben, o símbolo %, que representa calquera ordenador (excepto a máquina local). O símbolo % tamén se pode utilizar como un carácter comodín en combinación co nome do equipo, ou a dirección IP:

  Exemplos de usuarios:
  - **'administrador'@'localhost'** Usuario administrador cando se conecte desde o equipo local
  - **'administrador'@'%'** Usuario administrador cando se conecte desde calquera equipo da rede
  - **'julio'@'ordenador124'** Usuario julio cando se conecta desde o equipo co nome ordenador124
  - **'andres'@ '192.68.123.50'** Usuario andres cando se conecta desde o equipo coa IP 192.68.123.50
  - **'luis'@'192.68.%.%'** Usuario luis cando se conecta desde un equipo cunha IP que empeza por 192.68

- **contrasinal**, A cláusula IDENTIFIED BY permite asignarlle unha contrasinal ao usuario no momento en que se lle conceden os permisos de acceso. Se pode cambiar a contrasinal dun usuario usando algunha das seguintes sentencias SQL:

```
GRANT USAGE ON *.* TO  nome_usuario[ @equipo ]  IDENTIFIED BY 'contrasinal';
SET PASSWORD [FOR usuario]  =  PASSWORD('contrasinal');
```

  A función PASSWORD cifra o contrasinal, utilizando o sistema de cifrado de MySQL, que converte calquera cadea de texto nunha cadea de 41 carácteres en hexadecimal. Existen outros sistemas de cifrado que se poden utilizar facendo uso doutras funcións de cifrado, como por exemplo SHA1, ou MD5.

Aínda que sempre se deben conceder os mínimos privilexios necesarios, existen algúns privilexios que son especialmente perigosos. Por exemplo, nunca se debe conceder acceso de carácter global. Os seguintes privilexios poden resultar unha ameaza para a seguridade da base de datos:

Calquera privilexio sobre a base de datos mysql	Nesta base de datos se almacena información de todo o sistema de seguridade do servidor
ALTER	Un usuario podería modificar as táboas de privilexios, e inutilizalas
DROP	Un usuario podería borrar as táboas de privilexios, perdéndose a información das contas, o que impediría o acceso dos usuarios.
FILE	Os usuarios poderían crear un ficheiro con información das contas de usuario, que todo o mundo poda ler.
GRANT	Permite que un usuario poda ceder os seus privilexios a outros usuarios, que poden non ser tan fiables como el
PROCESS	As consultas realizadas poden ser vistas en modo texto, o que inclúe calquera que cambie ou defina contrasinais.
SHUTDOWN	Os usuarios con este privilexio poden parar o servidor, e deixar ao resto dos usuarios sen servizo

Por exemplo, para crear un usuario sen privilexios:

```
GRANT USAGE ON *.* TO anonimo IDENTIFIED BY 'clave';
```

Hai que ter en conta que o contrasinal débese introducir entre comiñas de forma obrigatoria. O usuario 'anonimo' poderá abrir unha sesión MySQL mediante unha orde:

```
mysql -h localhost -ou anonimo -p
```

Pero non poderá facer moito máis, xa que non ten privilexios. Non terá, por exemplo, oportunidade de facer seleccións de datos, de crear bases de datos ou táboas, inserir datos, etc.

Máis exemplos:

- Conceder permiso para executar os comandos *insert* e *delete* na táboa *titles* a *Mary* :

```
grant insert, delete
on titles
to mary
```

- Conceder permiso para executar o comando *update* nas columnas *price* e *advance* da táboa *titles* a *public*.

```
grant update (price, advance)
on titles
to public
```

- Conceder permiso a Mary e John para utilizar os comandos create database e create table.

```
grant create database, create table
to mary, john
```

- Conceder todos os permisos de acceso á táboa *titles* a todos os usuarios.

```
grant all on titles
to public
```

- Conceder permiso a *Mary* para utilizar o comando *update* na táboa *authors* e para conceder ese permiso a outros.

```
grant update on authors
to mary
with grant option
```

Revogar privilexios

Para revogar privilexios úsase a sentenza REVOKE.

```
REVOKE priv_type [(column_list)] [, priv_type [(column_list)]] ...
ON
FROM user [, user] ...
```

A sintaxe é similar á de GRANT, por exemplo, para revogar o privilexio SELECT da táboa *xente* da base de datos *proba* ao usuario *anonimo*, usarase a sentenza:

```
REVOKE SELECT ON proba.xente FROM anonimo;
```

Mostrar os privilexios dun usuario

Pódense ver os privilexios que se lle concederon a un usuario mediante a cláusula SHOW GRANTS. A saída desta sentenza é unha lista de sentenzas GRANT que se deben executar para conceder os privilexios que ten o usuario. Por exemplo:

```
mysql> SHOW GRANTS FOR anonimo;
-------------------------------------------------------------------
| Grants for anonimo@% |
-------------------------------------------------------------------
| GRANT USAGE ON *.* TO 'anonimo'@'%' IDENTIFIED BY PASSWORD '*5...' |
| GRANT SELECT ON `proba`.`xente` TO 'anonimo'@'%' |
-------------------------------------------------------------------
```

Borrar usuarios

Para eliminar usuarios úsase a sentenza DROP USER.

Por exemplo:

```
mysql> DROP USER anonimo;
Query OK, 0 rows affected (0.00 sec)
```

Tarefa 1. Xestionar usuarios en MySQL.