UD 05

SQL. Acceso a la información

Contenido del tema

- **Introducción**
- Gestión de cuentas de usuario sobre bases de datos
  - Creación de usuarios
  - Modificación de usuarios
  - Eliminación de usuarios
- Gestión de roles sobre usuarios
  - Asignación de roles
  - Revocación de roles
- Gestión de privilegios sobre objetos
  - Asignación de privilegios
  - Revocación de privilegios
- Vistas
  - Creación
  - Borrado

Introducción

La administración de una base de datos puede ser una tarea muy compleja debido a las innumerables posibilidades que las herramientas de MySQL proporcionan a sus usuarios.

- Bases de datos de gran tamaño con múltiples accesos desde distintos equipos.
- Bases de datos distribuidas.
- Parametrización de comportamiento, que redunda en un mejor aprovechamiento pero también en una mayor complejidad.

El administrador de la base de datos suele ser una persona con experiencia y grandes conocimientos en bases de datos que debe resolver los problemas de los usuarios y los que el sistema va planteando. Tareas:

- Instalación de MySQL
- Diseño y creación de la base de datos
- Arranque y parada de la base de datos
- Crear y controlar usuarios
- Gestionar espacio
- Hacer copias de seguridad
- Recuperar la base de datos en caso de fallos

Introducción

La gestión de seguridad tiene mucho que ver con la gestión de usuarios y con la concesión y supresión de privilegios a los usuarios.

El administrador de la base de datos es el responsable de permitir o denegar el acceso a los usuarios a determinados objetos o recursos de la base de datos.

Se puede clasificar la seguridad de la base de datos en dos categorías: seguridad del sistema y seguridad de los datos.

- Seguridad del Sistema: Incluye los mecanismos que controlan el acceso y uso de la base de datos a nivel del sistema. Por ejemplo: cada vez que se conecta un usuario a la base de datos, los mecanismos de seguridad comprobarán si éste está autorizado.
- Seguridad de los Datos: Incluye mecanismos que controlan el acceso y uso de la base de datos a nivel de objetos. Por ejemplo, cada vez que un usuario acceda a un objeto (tabla, vista, etc), los mecanismos de seguridad comprobarán si el usuario puede acceder a ese objeto y qué tipo de operación puede hacer con él (INSERT, SELECT, etc.).

Contenido del tema

- Introducción
- **Gestión de cuentas de usuario sobre bases de datos**
  - Creación de usuarios
  - Modificación de usuarios
  - Eliminación de usuarios
- Gestión de roles sobre usuarios
  - Asignación de roles
  - Revocación de roles
- Gestión de privilegios sobre objetos
  - Asignación de privilegios
  - Revocación de privilegios
- Vistas
  - Creación
  - Borrado

Gestión de cuentas de usuario

Un **usuario** es un nombre definido en la base de datos que se puede conectar y acceder a determinados objetos según ciertas condiciones que define el administrador.

Estos usuarios acceden a la BD ejecutando una aplicación de bases de datos o que conecte con la BD.

Durante la instalación de la base de datos de MySQL se crea un primer usuario denominado usuario **root**. Éste será el usuario con mayor privilegio del sistema y solo se recomiendo su utilización a la hora de la instalación del sistema.

Las operaciones que podremos realizar con los usuarios son básicamente tres:

- Creación: CREATE USER nombre IDENTIFIED BY contraseña
- Modificación: ALTER USER nombre IDENTIFIED BY contraseña
- Eliminación: DROP USER nombre [CASCADE]

Contenido del tema

- Introducción
- Gestión de cuentas de usuario sobre bases de datos
  - Creación de usuarios
  - Modificación de usuarios
  - Eliminación de usuarios
- **Gestión de roles sobre usuarios**
  - Asignación de roles
  - Revocación de roles
- Gestión de privilegios sobre objetos
  - Asignación de privilegios
  - Revocación de privilegios
- Vistas
  - Creación
  - Borrado

Gestión de roles sobre usuarios

Un rol es un conjunto de privilegios predefinidos que se pueden asociar de forma directa a un usuario. Para conceder un rol se utiliza el comando GRANT con la siguiente sintaxis:

```
GRANT nombre_rol TO nombre_usuario [IDENTIFIED BY contraseña] [WITH AMDIN OPTION]
```

Dónde IDENTIFIED BY indica la contraseña de la cuenta de usuario y WITH ADMIN OPTION indica que el usuario pueda asignar dicho rol a otros usuarios.

Para revocar los roles concedidos a un usuario se usa el comando REVOKE con la siguiente sintaxis:

```
REVOKE nombre_rol FROM nombre_usuario
```

Se muestra a continuación una lista con los roles disponibles en MySQL

AUDIT_ADMIN	Permite que el usuario pueda realizar políticas de auditoría a través de AUDIT y NOAUDIT
AUDIT_VIEWER	Permite ver y analizar las auditorías de datos
CAPTURE_ADMIN	Permite crear y gestionar privilegios de análisis de políticas
DBFS_PROFILE	Permite el acceso a los objetos y paquetes del sistema de ficheros
DELETE_CATALOG_ROLE	Se puede borrar registros en la tabla de auditoría del sistema
EXECUTE_CATALOG_ROLE	Se puede usar EXECUTE sobre los objetos del diccionario de datos
EXP_FULL_DATABASE	Permite exportar copias completas e incrementales
JAVA_ADMIN	Permite administrar las tablas de políticas para las aplicaciones Java
OLAP_DBA	Permite administrar los objetos dimensionados para Oracle OLAP
OLAP_USER	Permite desarrollar aplicaciones que crean objetos dimensionados.

Contenido del tema

- Introducción
- Gestión de cuentas de usuario sobre bases de datos
  - Creación de usuarios
  - Modificación de usuarios
  - Eliminación de usuarios
- Gestión de roles sobre usuarios
  - Asignación de roles
  - Revocación de roles
- **Gestión de privilegios sobre objetos**
  - Asignación de privilegios
  - Revocación de privilegios
- Vistas
  - Creación
  - Borrado

Gestión de privilegios sobre objetos

Los privilegios sobre los objetos determinan como un usuario puede tratar los objetos de lo que no es propietario. Por ejemplo, un usuario que trabaja en el departamento de ventas no debería poder tocar el campo Stock de la tabla Producto pero sin embargo sí podría hacer consultas. Por otro lado, algunos usuarios del departamento de incidencias si pueden modificar los valores de la tabla Precios.

Igual que en el caso de los roles el comando que se usa para otrogar estos privilegios es GRAN pero con una sintaxis diferente

GRANT {ALL [PRIVILEGES] | privilegio} ON objeto TO usuario [WITH GRANT OPTION]

Donde ALL otorga todos los privilegios y WITH GRANT OPTION permite además a dicho usuario otorgarlo a otros.

Para revocar dichos privilegios también se usa el comando REVOKE con la siguiente sintaxis

REVOKE {ALL [PRIVILEGES] | privilegio} ON objeto FROM usuario

Vistas

Las vistas son un recurso que se suelen asociar a la gestión de acceso y seguridad en las bases de datos. Permiten crear una “ventana” o “vista” sobre una determinada consulta mostrando al usuario solo aquello que se quiere mostrar protegiendo el resto de información.

Se pueden usar como un objeto y permitir que determinados usuarios tengan acceso a dichas vistas.

Para crear vistas, su sintaxis:

```
CREATE OR REPLACE VIEW nombre_vista AS
SELECT campo1, campo2... FROM tabla WHERE condicion
```

Con la anterior instruccion podemos crear la vista o reemplazarla en caso de que exista. Para su eliminación se procederá a utilizar el comando DROP

```
DROP VIEW nombre_vista
```