

# Unidad 2

---

## Control del acceso a las bases de datos

---

**Administración de Bases de Datos  
2º de ASIR**



Esta obra está bajo una licencia de Creative Commons.  
Autor: Jorge Sánchez Asenjo (año 2011) <http://www.jorgesanchez.net>  
e-mail: [info@jorgesanchez.net](mailto:info@jorgesanchez.net)

---

Esta obra está bajo una licencia de Reconocimiento-NoComercial-CompartirIgual de Creative Commons  
Para ver una copia de esta licencia, visite:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>  
o envíe una carta a:  
Creative Commons, 559 Nathan Abbot









## Reconocimiento-NoComercial-CompartirIgual 2.5 España

### Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



hacer obras derivadas

### Bajo las condiciones siguientes:



**Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciadador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



**No comercial.** No puede utilizar esta obra para fines comerciales.



**Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Apart from the remix rights granted under this license, nothing in this license impairs or restricts the author's moral rights.

Advertencia

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.  
Esto es un resumen legible por humanos del texto legal (la licencia completa) disponible en los idiomas siguientes:  
Catalán Castellano Euskera Gallego

Para ver una copia completa de la licencia, acudir a la dirección  
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>



## índice

<b>(2.1)</b> introducción	9
<b>(2.2)</b> control de usuarios en Oracle	9
(2.2.1) características de los usuarios de Oracle	9
(2.2.2) creación de usuarios en Oracle	10
(2.2.3) modificación de usuarios	11
(2.2.4) borrado de usuarios	11
(2.2.5) consultar usuarios	11
<b>(2.3)</b> control de privilegios en Oracle	11
(2.3.1) privilegios de sistema	11
(2.3.2) conceder privilegios	12
(2.3.3) revocar	12
(2.3.4) privilegios de objeto	13
(2.3.5) quitar privilegios de objeto	13
(2.3.6) mostrar información sobre privilegios	14
<b>(2.4)</b> administración de roles en Oracle	14
(2.4.1) creación de roles	14
(2.4.2) asignar y retirar privilegios a roles	14
(2.4.3) asignar roles	14
(2.4.4) roles predefinidos	15
(2.4.5) activar y desactivar roles	15
(2.4.6) borrar roles	15
(2.4.7) información sobre roles	15
<b>(2.5)</b> administración de perfiles de Oracle	16
(2.5.1) crear perfiles	17
(2.5.2) modificar perfiles	17
(2.5.3) borrar perfil	17
(2.5.4) asignar un perfil a un usuario	18
<b>(2.6)</b> usuarios y privilegios en MySQL	18
(2.6.1) cuentas de usuario en MySQL	18
(2.6.2) creación de usuarios	19
(2.6.3) borrado de usuarios	19
(2.6.4) consulta de los usuarios de MySQL	19
(2.6.5) modificar usuarios de MySQL	19
(2.6.6) cambiar de nombre a un usuario	19
(2.6.7) concesión de privilegios en MySQL	20
(2.6.8) revocación de permisos	22
(2.6.9) mostrar información sobre usuarios y privilegios	23



# (2)

## control de acceso a la base de datos

### (2.1) introducción

Todo acceso a una base de datos requiere conectar mediante un usuario y contraseña. Dicho usuario dará derecho a utilizar ciertos objetos de la base de datos, pero se puede restringir el uso de otros.

A los usuarios se les asigna una serie de privilegios que son los que dan permiso de uso a ciertos objetos. Para organizarse mejor la mayoría de Sistemas Gestores de Bases de Datos permiten agrupar permisos que normalmente se aplican conjuntamente en estructuras llamadas perfiles y roles, que en definitiva son un conjunto de permisos.

### (2.2) control de usuarios en Oracle

#### (2.2.1) características de los usuarios de Oracle

A los usuarios de Oracle se les puede asignar la configuración referida a:

- **Nombre de usuario.** No puede repetirse y como máximo debe tener 30 caracteres que sólo podrán contener letras del alfabeto inglés, números, el signo dólar y el signo de guión bajo (\_)
- **Configuración física.** Se refiere al espacio asociado al usuario para almacenar sus datos (lo que Oracle llama **tablespace**) y la cuota (límite de almacenamiento) que se le asigna.
- **Perfil asociado.** El perfil del usuario indica los recursos y configuración que tomará el usuario al sistema
- **Privilegios y roles.** Permiten especificar las acciones que se le permiten realizar al usuario.
- **Estado de la cuenta de usuario:**
  - **Abierta.** El usuario puede conectar y realizar sus acciones habituales
  - **Bloqueada.** EL usuario no podrá conectar mientras siga en estado bloqueado. El bloqueo lo realiza el DBA: `ALTER USER usuario ACCOUNT LOCK`
  - **Expirada.** La cuenta agotó el tiempo máximo asignado a ella. Para salir de este estado, el usuario/a debe resetear su contraseña de usuario.
  - **Expirada y bloqueada.**

- **Expirada en periodo de gracia.** Está en los últimos momentos de uso antes de pasar a estado de expirada
- **Métodos de autenticación.** Define la forma en la que el usuario verifica quién es. Posibilidades:
  - **Autenticación por el sistema operativo.** Sólo vale para administradores (a los que se asigne roles de **SYSDBA** o **SYSOPER**) para conectar con este tipo de usuario se usa **CONNECT / AS SYSDBA** (se puede usar **SYSOPER** en lugar de **SYSDBA**).
  - **Autenticación por archivo de contraseñas.** Se usa en los mismos casos que la anterior. Cuando no se considera que el Sistema Operativo sea muy seguro, se utiliza como opción. Para usar esta forma de autenticación los usuarios de tipo **SYSDBA** o **SYSOPER** indican su nombre de usuario y contraseña al conectar (opcionalmente indican el host al que se desean conectar) esos datos se contrastará con los del archivo de contraseñas utilizado. Esta forma (y la anterior) permite conectar la base de datos aunque no esté montada todavía la base de datos.
  - **Autenticación por contraseña.** Una contraseña se utilizará para autenticar al usuario. Se utiliza para todo tipo de usuarios, la contraseña se valida consultando el diccionario de datos; por lo que esta configuración requiere la base de datos montada y abierta (por ello no es válida para usuarios con rol de **SYSDBA** y **SYSOPER**).
  - **Autenticación externa.** Oracle delega la autenticación a un servicio externo que se asociará a Oracle. Ejemplos de servicios externos son **Kerberos** o **RADIUS**.
  - **Autenticación global.** Se trata de utilizar un servicio LDAP para realizar la autenticación. Oracle dispone de un servicio LDAP global integrado en **Oracle Applications** (plataforma de Oracle para la creación de aplicaciones) que se llama **Oracle Internet Directory**.

### (2.2.2) creación de usuarios en Oracle

La sentencia de creación de usuarios (que es estándar) es:

```
CREATE USER nombre IDENTIFIED BY 'contraseña' [OPCIONES]
```

Es una sentencia estándar, a la que se le pueden añadir múltiples cláusulas.

```
CREATE USER nombre {IDENTIFIED BY 'contraseña' | EXTERNALLY}
[DEFAULT TABLESPACE tableSpacePorDefecto]
[TEMPORARY TABLESPACE tableSpaceTemporal]
[QUOTA {cantidad [K|M] | UNLIMITED} ON tablespace
[QUOTA {cantidad [K|M] | UNLIMITED} ON tablespace [...]]
]
[PASSWORD EXPIRE]
[ACCOUNT {UNLOCK|LOCK}];
[PROFILE {perfil | DEFAULT}
```

Sólo la primera línea es obligatoria, el resto posee opciones por defecto que se aplican si no se especifica cada apartado (no hace falta especificar todos, sólo las líneas que nos interesen). Ejemplo:

```
CREATE USER jsanchez IDENTIFIED BY 'Caracola'
DEFAULT TABLESPACE 'Usuarios'
QUOTA 15M ON 'Usuarios' //Se dan 15MBytes de espacio en el tablespace
ACCOUNT LOCK; //La cuenta estará bloqueada
```

### (2.2.3) modificación de usuarios

Cada parámetro indicado anteriormente se puede modificar mediante la instrucción ALTER USER que se utiliza igual que CREATE USER. Ejemplo:

```
ALTER USER jsanchez QUOTA UNLIMITED ON usuarios
```

### (2.2.4) borrado de usuarios

Se realiza mediante:

```
DROP USER usuario [CASCADE]
```

La opción CASCADE elimina los objetos del esquema del usuario antes de eliminar al propio usuario. Es obligatorio si el esquema contiene objetos.

### (2.2.5) consultar usuarios

La vista administrativa DBA\_USERS muestra la lista y configuración de todos los usuarios del sistema.

## (2.3) control de privilegios en Oracle

Los privilegios son permisos que damos a los usuarios para que puedan realizar ciertas operaciones con la base de datos. En Oracle hay más de cien posibles privilegios. Se dividen en:

- **Privilegios de sistema.** Son permisos para modificar el funcionamiento de la base de datos. Son cambios, en definitiva, que afectan a todos los usuarios y usuarias.
- **Privilegios de objeto.** Son permisos que se aplican a un objeto concreto de la base de datos.

### (2.3.1) privilegios de sistema

Se comentan algunos de los privilegios de sistema más importantes

Privilegio	Significado
CREATE SESSION	Permite al usuario conectar con la base de datos
ALTER DATABASE	Permite modificar la estructura de la base de datos
ALTER SYSTEM	Permite modificar los parámetros y variables del sistema

Privilegio	Significado
CREATE TABLE	Permite crear tablas. Incluye la posibilidad de borrarlas.
GRANT ANY OBJECT PRIVILEGE	Permite conceder privilegios de los objetos del usuario hacia otros usuarios
CREATE ANY TABLE	Permite crear tablas en otros esquemas de usuario
DROP ANY TABLE	Permite borrar tablas de otros usuarios
SELECT ANY TABLE	Permite seleccionar datos en tablas de otros usuarios
INSERT ANY TABLE	Permite añadir datos en tablas de otros usuarios
UPDATE ANY TABLE	Permite eliminar datos en tablas de otros usuarios
DELETE ANY TABLE	Permite eliminar datos en tablas de otros usuarios

Funcionan parecido los privilegios para cualquier tipo de objeto de la base de datos, así igual que hay **CREATE TABLE**, se puede usar **CREATE VIEW** para las vistas o **INDEX**, **TRIGGER**, **PROCEDURE**, **SEQUENCE**, **SYNONYM**, **TYPE**,... para cada tipo de objeto que tengamos.

Hay dos privilegios especiales que permiten conceder nivel de DBA, son: **SYSDBA** y **SYSOPER**. **SYSDBA** es administrador total y **SYSOPER** es un poco rebajado: no puede crear usuarios con privilegios **SYSOPER** ni lanzar o parar la instancia de base de datos.

### (2.3.2) conceder privilegios

Se usa con la instrucción **GRANT** que funciona así:

```
GRANT privilegio1 [,privilegio2[,...]] TO usuario  
[WITH ADMIN OPTION];
```

La opción **WITH GRANT OPTION** permite que el usuario al que se le concede el privilegio puede conceder dicho privilegio a otros usuarios. Es, por tanto, una opción a utilizar con cautela.

Ejemplo:

```
GRANT CREATE SESSION, ALTER SESSION, CREATE TABLE,  
CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE,  
CREATE TRIGGER, CREATE PROCEDURE, CREATE TYPE  
TO jsanchez;
```

### (2.3.3) revocar

Se realiza con la instrucción **REVOKE** que funciona de esta forma:

```
REVOKE privilegio1 [,privilegio2 [,...]] FROM usuario;
```

Al revocar los privilegios, las acciones llevadas a cabo con ellos no se anulan.

### (2.3.4) privilegios de objeto

Se trata de privilegios que se colocan a un objeto para dar permiso de uso a un usuario.

Sintaxis:

```
GRANT {privilegio [(listaColumnas)] [,privilegio [(listaColumnas)] [,...]] |  
ALL [PRIVILEGES]}  
ON [esquema.]objeto  
TO {usuario | rol | PUBLIC} [, {usuario | rol | PUBLIC} [,...]]  
[WITH GRANT OPTION]
```

La opción **ALL** concede todos los privilegios posibles sobre el objeto. Se pueden asignar varios privilegios a la vez y también varios posibles usuarios. La opción **WITH GRANT OPTION** permite al usuario al que se le conceden los privilegios, que pueda, a su vez, concederlos a otro.

Ejemplo de uso de GRANT con privilegios de objeto:

```
GRANT UPDATE, INSERT ON jsanchez.personas  
TO anozal;
```

Los privilegios posibles están en la siguiente tabla:

Privilegio	Aplicable a
SELECT	Tablas, vistas, secuencias, sinónimos
INSERT	Tablas, vistas, sinónimos
UPDATE	Tablas, vistas, sinónimos
DELETE	Tablas, vistas, sinónimos
ALTER	Tablas, secuencias
EXECUTE	Procedimientos, funciones, paquetes, sinónimos

### (2.3.5) quitar privilegios de objeto

Sintaxis:

```
REVOKE {privilegio1 [,privilegio2] [,...]} |  
ALL [PRIVILEGES]}  
ON [esquema.]objeto  
FROM {usuario | rol | PUBLIC} [, {usuario | rol | PUBLIC} [,...]]  
[CASCADE CONSTRAINTS]
```

**CASCADE CONSTRAINTS** elimina cualquier restricción que impida el borrado del privilegio.

### (2.3.6) mostrar información sobre privilegios

Vista	Significado
DBA_SYS_PRIVS	Privilegios de sistema asignados a usuarios y roles
DBA_TAB_PRIVS	Lista de todos los privilegios de todos los objetos de la base de datos
DBA_COL_PRIVS	Lista de todos los privilegios aplicados a columnas de la base de datos
SESSION_PRIVS	Privilegios en activo para el usuario y sesión actuales

## (2.4) administración de roles en Oracle

Los roles son privilegios aglutinados sobre un mismo nombre, bajo la idea de que ese conjunto denote un uso habitual sobre la base de datos. Gracias a los roles se facilita la asignación de privilegios a los usuarios. Un usuario puede tener asignados varios roles y viceversa.

### (2.4.1) creación de roles

Los roles se crean usando esta sintaxis

```
CREATE ROLE rol [NOT IDENTIFIED |  
IDENTIFIED {BY password | EXTERNALLY |  
GLOBALLY | USING package}];
```

La opción **IDENTIFIED** funciona igual que las formas de identificar un usuario, salvo la opción **PACKAGE** que hace que el rol sólo se pueda utilizar para el paquete de aplicaciones indicado. Por defecto un ROL no requiere identificación.

La instrucción **ALTER ROLE** permite modificar la configuración del rol (tiene las mismas opciones que **CREATE ROLE**)

### (2.4.2) asignar y retirar privilegios a roles

Se realiza con la instrucción **GRANT**. A los roles se les asignan privilegios igual que a los usuarios, pueden ser de sistema y/o de objeto.

Lógicamente se eliminan mediante **REVOKE**.

### (2.4.3) asignar roles

Se pueden asignar usuarios a un usuario e incluso a otro rol. La sintaxis es:

```
GRANT rol1 [,rol2 [...]]  
TO {usuario|rol|PUBLIC [, {usuario|rol|PUBLIC} [...]}  
[WITH ADMIN OPTION]
```

Al igual que en las instrucciones anteriores, **PUBLIC** asigna el rol a todos los usuarios y **WITH ADMIN OPTION** permite al usuario al que se le concede el rol, conceder él dicho rol a otros usuarios/as.

### (2.4.4) roles predefinidos

Oracle dispone de una serie de roles predefinidos que se pueden asignar a los usuarios. Hay más de cincuenta roles predefinidos. Los clásicos son:

rol	significado
<b>CONNECT</b>	Permite crear sesiones. Se mantiene por compatibilidad
<b>RESOURCE</b>	Permite crear tablas y código PL/SQL del tipo que sea. Se mantiene por compatibilidad
<b>DBA</b>	Permite casi todo, excepto manejar la instancia de la base de datos

### (2.4.5) activar y desactivar roles

Se realiza mediante **SET ROLE** que se encarga de desactivar (temporalmente) y activar roles. Su sintaxis:

#### SET ROLE

```
{ rol1 [IDENTIFIED BY contraseña]
  [,rol2 [IDENTIFIED BY contraseña] [,...]]
| ALL [EXCEPT rol1 [,rol2 [,...]]]
| NONE};
```

Las posibilidades son:

- Indicar una lista de roles que serán los que se activen (se usa cuando se habían desactivado)
- Indicar **ALL** para activar todos los roles, excepto aquellos que se indiquen en la cláusula **EXCEPT** que sirve para desactivarlos.
- **NONE** desactiva todos los roles, sólo los privilegios indicados directamente podrán ser utilizados por el usuario.

### (2.4.6) borrar roles

Lo hace la instrucción **DROP ROLE**, seguida del rol a borrar. Desde ese momento a los usuarios a los que se habían asignado el rol se les revoca.

### (2.4.7) información sobre roles

Existen varias vistas para examinar los roles.

Vista	Significado
<b>DBA_ROLES</b>	Muestra todos los roles de la base de datos
<b>DBA_ROLES_PRIVS</b>	Roles asignados a los usuarios
<b>ROLE_ROLE_PRIVS</b>	Roles asignados a otros roles
<b>DBA_SYS_PRIVS</b>	Privilegios de sistema asignados a usuarios y roles

Vista	Significado
ROLE_SYS_PRIVS	Privilegios de sistema asignados a roles
ROLE_TAB_PRIVS	Privilegios de objeto concedidos a roles
SESSION_ROLES	Roles en activo para el usuario actual

## (2.5) administración de perfiles de Oracle

Los perfiles permiten limitar los recursos que los usuarios usan de la base de datos. Hay un perfil llamado **DEFAULT** que se aplica automáticamente a todos los usuarios y que les da recursos ilimitados sobre la base de datos. Para limitar el número de recursos en principio se debe de activar a **TRUE** la variable de sistema **RESOURCE\_LIMIT** (que por defecto está a **FALSE**). Esto se hace:

```
ALTER SYSTEM SET RESOURCE_LIMIT=TRUE;
```

En realidad hay dos tipos de parámetros de los perfiles:

■ **Manejo de contraseñas**, los posibles cambios respecto a ese aspecto son:

Variable de perfil	Significado
FAILED_LOGIN_ATTEMPTS	Número consecutivo de errores en las contraseñas antes de bloquear la cuenta. Por defecto son 10
PASSWORD_LOCK_TIME	Número de días hasta que se bloquea una cuenta si se supera el límite de intentos al meter una contraseña. Por defecto es uno
PASSWORD_LIFE_TIME	Números de días que tiene vigencia una contraseña. Por defecto es 180
PASSWORD_GRACE_TIME	Días que la contraseña se la concede un periodo extra de gracia tras consumir su tiempo de vida. Por defecto es 7
PASSWORD_REUSE_TIME	Número de días que una contraseña puede ser reutilizada
PASSWORD_VERIFY_FUNCTION	Función a la que se invoca cuando se modifica una contraseña con el fin de verificar su validez en base a las reglas de complejidad que deseemos

■ **Manejo de recursos.**

Variable de perfil	Significado
SESSIONS_PER_USER	Número de conexiones de usuario concurrentes que se permiten.
CPU_PER_SESSION	Límite de tiempo (en centésimas de segundo) que se permite a un usuario utilizar la CPU antes de ser echado del sistema. De esa forma se evitan peligros de rendimiento
CPU_PER_CALL	Como la anterior pero referida a cada proceso

Variable de perfil	Significado
<b>PRIVATE_SGA</b>	Para conexiones en instalaciones de servidor compartido, número de KB que puede consumir cada sesión en la zona de memoria compartida ( <b>SGA</b> )
<b>CONNECT_TIME</b>	Minutos como máximo que se permite a una sesión
<b>IDLE_TIME</b>	Minutos máximos de inactividad de una sesión
<b>LOGICAL_READS_PER_SESSION</b>	Máximo número de bloques leídos en una sesión
<b>LOGICAL_READS_PER_CALL</b>	Máximo número de bloques leídos por un proceso
<b>COMPOSITE_LIMIT</b>	Máximo número de recursos consumidos por una sesión. Es la media ponderada de varios parámetros anteriores

### (2.5.1) crear perfiles

Sintaxis:

```
CREATE PROFILE perfil LIMIT parámetros
```

Los parámetros son los explicados anteriormente a los que se les indica un valor, o bien la palabra **DEFAULT** (significa que el parámetro toma su valor por defecto) o bien **UNLIMITED** para indicar que no tienen límite. Ejemplo:

```
CREATE PROFILE programador LIMIT  
SESSIONS_PER_USER UNLIMITED  
CPU_PER_SESSION UNLIMITED  
IDLE_TIME 15  
CONNECT_TIME 150  
FAILED_LOGIN_ATTEMPTS 5  
PASSWORD_LOCK_TIME 2;
```

### (2.5.2) modificar perfiles

La instrucción **ALTER PROFILE** funciona igual que **CREATE PROFILE** y es la encargada de hacer modificaciones a un perfil creado. Permite hacer modificaciones al perfil que se usa por defecto (**DEFAULT**).

### (2.5.3) borrar perfil

En este caso es **DROP PROFILE** seguida del nombre del perfil a eliminar. Se puede usar la palabra **CASCADE** para eliminar todas las restricciones que impidan crear el perfil. Sintaxis:

```
DROP PROFILE perfil [CASCADE]
```

### (2.5.4) asignar un perfil a un usuario

Cada usuario tiene un solo perfil. La instrucción de creación de usuarios ya dispone de apartado para indicar el perfil que se asigna. Pero si lo deseamos hacer después disponemos de la instrucción **ALTER USER** con la que podemos indicar el perfil para el usuario. Ejemplo:

```
ALTER USER jsanchez PROFILE programador;
```

## (2.6) usuarios y privilegios en MySQL

### (2.6.1) cuentas de usuario en MySQL

En MySQL el nombre de un usuario está compuesto por el nombre seguido del signo @ y después el ordenador desde el que dicho usuario se conecta, porque se asume que no es lo mismo el usuario [pepe@192.168.1.35](#) que [pepe@192.168.1.36](#), es decir hay diferentes usuarios de nombre *pepe* y tendrán por tanto diferentes privilegios según de qué *pepe* hablemos en base a la máquina o la red desde la que se conectan.

A partir de esta idea, cuando un usuario se conecta primero se comprueba si tiene permiso para hacerlo (suponiendo que la contraseña sea correcta). Después cada operación que intenta realizar será controlada para saber si se permite o no.

La tabla **mysql.user**, es decir: tabla **user** de la base de datos **mysql** que contiene la información sobre los usuarios de mysql. En ella se observa una de las particularidades de MySQL, los usuarios usan un nombre seguido del host. De esa forma dos usuarios pueden parecer iguales pero al variar la parte del host, se convierten en dos usuarios diferentes. Por ejemplo, [usuario@192.168.1.10](#) sería diferente de [usuario@192.168.1.11](#)

En MYSQL el nombre de usuario debe de cumplir:

- Tener un máximo de 16 caracteres
- Debe comenzar por letra
- No puede repetirse para el mismo host
- Si el usuario lleva espacios en blanco, se coloca entre comillas simples

Para la parte que se refiere al host, es posible usar:

- Direcciones IP, como [192.168.1.10](#)
- Nombres de host, como [localhost](#) o cualquier otro nombre reconocido por nuestro servidor DNS
- Direcciones IP con máscara, como [192.168.1.0/255.255.255.0](#) (sólo son válidas de 8,16,24 o 32 bits)
- Direcciones IP con el comodín %, por ejemplo [192.168.%.%](#) representa cualquier máquina de la red 192.168.0.0
- En los dos últimos puntos anteriores, el host se escribe entre comillas simples.

Cada usuario tiene asociada una contraseña así como una serie de operaciones posibles para realizar.

### (2.6.2) creación de usuarios

Desde la versión 5.0.2 de MySQL es posible utilizar el comando estándar **CREATE USER**. La sintaxis es:

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

Se pueden crear usuarios sin indicar contraseñas. El nombre de usuario debe incluir el host (como se comentó en el apartado anterior), de otro modo usará el host '%' que representa a cualquier máquina (es un usuario global).

No es obligatorio el apartado **IDENTIFIED BY** que permite indicar la contraseña; si no se hace uso de él, la contraseña del usuario queda en blanco (situación nada recomendable). La contraseña se puede indicar en texto plano o a través de la función **PASSWORD** indicado el resultado de aplicar la función **PASSWORD()** (de esa forma se oculta el texto plano).

Los usuarios así creados no tienen privilegios asociados,

### (2.6.3) borrado de usuarios

Se realiza mediante la instrucción:

```
DROP USER usuario [,...];
```

Si el usuario tiene sesión abierta, no se cierra la sesión. Se aplicará el comando al cierre de la sesión de dicho usuario.

### (2.6.4) consulta de los usuarios de MySQL

La tabla **mysql.user** contiene la lista completa de usuarios. Modificar esta tabla permite modificar los usuarios, por lo que las instrucciones **INSERT**, **DELETE** o **UPDATE** en esta tabla añaden, modifican o eliminan usuarios; aunque no se recomienda ni **INSERT** ni **DELETE** (al existir las instrucciones de creación y eliminación de usuario estándares).

Las principales columnas de esa tabla son **user**, **host** y **password**.

### (2.6.5) modificar usuarios de MySQL

El comando **UPDATE** sobre la tabla de usuarios, **mysql.user**, es la forma habitual de hacerlo, pero necesitamos invocar al comando **FLUSH PRIVILEGES** para que los cambios se realicen al instante. Ejemplos:

```
UPDATE mysql.user SET host='192.168.1.%' WHERE user='opersys';  
UPDATE mysql.user SET password=PASSWORD('123456')  
WHERE user='clara';  
FLUSH PRIVILEGES;
```

### (2.6.6) cambiar de nombre a un usuario

Se usa el comando no estándar, **RENAME USER**, de esta forma:

```
RENAME USER nombreAntiguo TO nombreNuevo  
[,nombreAntiguo2 TO nombreNuevo2 [,...]
```

## (2.6.7) concesión de privilegios en MySQL

En MySQL es el comando estándar GRANT el que permite la concesión de privilegios. La sintaxis es extensa:

```
GRANT tipoDePrivilegio[(listaColumnas1)], tipoDePrivilegio[(listaColumnas2)][,...]  
ON [tipoDeObjeto]{tabla | * | *.* | baseDeDatos.* | baseDeDatos.tabla}  
TO usuario1 [IDENTIFIED BY [PASSWORD] 'contraseña'][, usuario2...]  
[WITH opción [,opción2[,...]]]
```

El *tipoDeObjeto* puede ser:

- TABLE
- FUNCTION
- PROCEDURE

Si no se indica tipo de objeto, se entiende que nos referimos a una tabla (que es lo habitual).

Las *opciones* del apartado **WITH** son:

- **GRANT OPTION**. Que permite que el usuario al que se le conceden los privilegios pueda, a su vez, concederles a otros.
- **MAX\_QUERIES\_PER\_HOUR *n***. Permite indicar el máximo número de consultas (*n*) a la hora que se le permiten al usuarios.
- **MAX\_UPDATES\_PER\_HOUR *n*** Máximo número de operaciones de modificación de datos permitidas en una hora.
- **MAX\_CONNECTIONS\_PER\_HOUR *n***. Máximo número de conexiones que se le permiten hacer al usuario en una hora.
- **MAX\_USER\_CONNECTIONS *n***. Conexiones concurrentes que como máximo el usuario puede mantener abiertas.

En todas las opciones anteriores si se le da a *n* un valor de **0**, entonces se traduce como **ilimitado**. Es decir **MAX\_USER\_CONNECTIONS 0** permite tener ilimitadas conexiones simultáneas.

El objeto al que se le aplican los privilegios puede ser:

- Una tabla de la base de datos actual. De ella se indica el nombre simplemente.
- Una tabla de una base de datos concreta. Se indica la base de datos y la tabla separadas por un punto.
- Todas las tablas de la base de datos actual. Se indica un asterisco.
- Todas las tablas de una base de datos. Se indica el nombre de la base de datos seguida de un punto y un asterisco.
- Todas las tablas de todas las bases de datos. Dos asteriscos separados por un punto.

Además existen cinco niveles de aplicación de los permisos:

- **Nivel global.** Se aplican a todas las bases de datos. Por lo que se deben aplicar obligatoriamente al objeto **\*\***. Ejemplos de privilegios que se aplican obligatoriamente a este nivel son **CREATE USER** o **SHUTDOWN**
- **Nivel de bases de datos.** Se aplican a todos los objetos de una base de datos. Se deben de aplicar usando el asterisco en las tablas pero un nombre de base de datos.
- **Nivel de tabla.** Se aplican sólo a una tabla.
- **Nivel de columna.** Se aplican a columnas de una tabla
- **Nivel de rutina.** Son privilegios que se aplican a rutinas.

La cláusula **IDENTIFIED BY** permite incluso crear el usuario al que se da permisos; de hecho antes de la versión 5 de MySQL era la forma de crear usuarios.

Los posibles permisos (apartado **tiposDePrivilegio** de la sintaxis de GRANT) que se pueden otorgar son:

Permiso	Significado
<b>ALL [PRIVILEGES]</b>	Otorga todos los privilegios
<b>ALTER</b>	Permite el uso de <b>ALTER TABLE</b>
<b>ALTER ROUTINE</b>	Modifica o borra rutinas almacenadas
<b>CREATE</b>	Permite crear tablas
<b>CREATE ROUTINE</b>	Crea rutinas almacenadas
<b>CREATE TEMPORARY TABLES</b>	Permite el uso de <b>CREATE TEMPORARY TABLE</b>
<b>CREATE USER</b>	Permite el uso de <b>CREATE USER, DROP USER, RENAME USER, y REVOKE ALL PRIVILEGES</b>
<b>CREATE VIEW</b>	Permite crear vistas
<b>DELETE</b>	Permite eliminar filas de tablas
<b>DROP</b>	<b>DROP</b> Permite el uso de <b>DROP TABLE</b>
<b>EXECUTE</b>	Permite ejecutar rutinas
<b>FILE</b>	Permite importar y exportar datos mediante <b>SELECT ... INTO OUTFILE y LOAD DATA INFILE</b>
<b>INDEX</b>	Permite crear y borrar índices
<b>INSERT</b>	Permite añadir filas
<b>LOCK TABLES</b>	Permite el uso de <b>LOCK TABLES</b> en tablas para las que tenga el permiso <b>SELECT</b>
<b>PROCESS</b>	Permite el uso de <b>SHOW FULL PROCESSLIST</b>
<b>RELOAD</b>	Permite el uso de <b>FLUSH</b>
<b>REPLICATION CLIENT</b>	Permite al usuario preguntar dónde están los servidores maestro o esclavo

Permiso	Significado
<b>REPLICATION SLAVE</b>	Necesario para los esclavos de replicación (para leer eventos del log binario desde el maestro)
<b>SELECT</b>	Permite consultar datos
<b>SHOW DATABASES</b>	Permite consultar la lista completa de bases de datos
<b>SHOW VIEW</b>	Permite el uso de <b>SHOW CREATE VIEW</b>
<b>SHUTDOWN</b>	Permite el uso de <b>mysqladmin shutdown</b> , que cierra la instancia de MySQL
<b>SUPER</b>	Permite el uso de los comandos <b>CHANGE MASTER, KILL, PURGE MASTER LOGS</b> , y <b>SET GLOBAL</b> ; además se permite al usuario que el comando <b>mysqladmin debug</b> le permita conectar (una vez) incluso si se ha superado el número máximo de conexiones.
<b>UPDATE</b>	Permite la modificación de datos
<b>USAGE</b>	Sin privilegios, el estado que tiene un usuario que se acaba de crear con <b>CREATE USER</b> .
<b>GRANT OPTION</b>	Permite dar permisos

Ejemplos:

```
GRANT INSERT,DELETE,UPDATE,SELECT ON almacen.pedidos TO ana;
#Permite a la usuaria ana@% permisos de modificación adición ,
#borrado y consulta sobre la tabla pedidos de la base de datos almacen
GRANT INSERT,DELETE,UPDATE,SELECT ON almacen.pedidos TO mario
WITH GRANT OPTION;
#Igual que la anterior para el usuario mario@% que además podrá el
#mismo conceder esos permisos
GRANT SELECT ON almacen.* TO felipe@192.168.1.32;
#al usuario indicado se le permite consultar todas las tablas del almacen
GRANT ALL ON almacen.* TO clara;
#clara puede hacer cualquier operación sobre la tabla de almacenes
GRANT CREATE ON almacen.* TO julian IDENTIFIED BY 'Caswq1209';
#Crea o modifica (si existe) el usuario Julian con la contraseña
indicada y permiso de creación de tablas en la base de datos almacén
```

## (2.6.8) revocación de permisos

El comando estándar **REVOKE** se encarga de quitar los permisos concedidos a un usuario concreto. La sintaxis del comando es:

```
REVOKE tipoDePrivilegio [(listaColumnas1)]
[,tipoDePrivilegio[(listaDeColumnas2)]] [...]
```

```
ON [tipoDeObjeto]{tabla | * | *.* | baseDeDatos.* | baseDeDatos.tabla}  
FROM usuario1[, usuario2[,...]]
```

Ejemplo:

```
REVOKE SHUTDOWN ON *.* FROM alberto@localhost;  
REVOKE INSERT,DELETE,UPDATE ON almacen.pedidos FROM clara;  
REVOKE ALL PRIVILEGES, GRANT OPTION FROM ana, elena;
```

La última instrucción elimina todos los privilegios.

### (2.6.9) mostrar información sobre usuarios y privilegios

La información sobre usuarios y privilegios se encuentra, fundamentalmente en las tablas **user** (lista de usuarios), **host** (lista de hosts) y **db** (bases de datos del sistema). La instrucción **DESCRIBE** seguida del nombre de la tabla (por ejemplo *mysql.db*) permite observar las columnas de la tabla y así saber qué consultar en ellas.

Otras tablas interesantes son:

- **tables\_priv**. Privilegios concedidos a las tablas, de cada tabla aparecen los usuarios que pueden operar con ella y los privilegios concretos que se les ha concedido.
- **columns\_priv**. Privilegios concedidos a las columnas.

Todas las tablas anteriores se pueden manipular para conceder o quitar permisos, aunque no es muy lógico que esas operaciones se hagan así.