Administración de Bases de Datos

Tema 5
(parte 1 de 3)

Seguridad en las Bases de Datos

Objetivos

- Conocer los diferentes mecanismos para gestionar la seguridad en el SGBD Oracle
- Aprender a gestionar usuarios, roles, privilegios y perfiles
- Aprender a gestionar la auditoría del SGBD Oracle

Contenidos

5.1. Introducción

5.2. Autenticación en la base de datos

5.3. Gestión de usuarios

5.4. Gestión de recursos con perfiles (*profiles*)

5.5. Administración de privilegios

5.6. Roles

5.7. Auditoría

5.8. Copia de seguridad y recuperación

5.1. Introducción

- Diseñar, crear e imponer procedimientos de seguridad permite proteger un bien corporativo trascendental: **los datos**
- Los mecanismos de seguridad que proporciona Oracle se dividen en:
  - **Autenticación**. Métodos para identificar quién está accediendo a la base de datos
  - **Autorización**. Proporciona acceso a diversos objetos de la base de datos después de habernos autentificado. Por ejemplo, privilegio para acceder a una determinada tabla, autorización para tener una sesión abierta durante sólo 15 minutos, etc.
  - **Auditoría**. Puede ser a nivel alto (por ejemplo, registrar los intentos correctos y fallidos de inicio de sesión de los usuarios) o de granularidad fina (por ejemplo, saber la columna en la que se ha realizado una operación de actualización)

5.2. Autenticación en la base de datos

- Cada BD Oracle dispone de una lista de usuarios válidos de la misma

- Para acceder a una base de datos, un usuario debe ejecutar una aplicación de la BD y conectarse a la instancia usando un nombre de usuario y una clave de acceso (opcional) definidas en la BD

- Oracle puede limitar el número de usuarios y el número de sesiones que pueden estar conectadas a una base de datos concurrentemente. Se establece en el fichero de inicialización mediante los parámetros:

  - **LICENSE_MAX_USERS:** especifica el número máximo de usuarios que pueden ser creados en la base de datos
  - **PROCESSES:** especifica el número máximo de procesos de usuario del sistema operativo que se pueden conectar simultáneamente a Oracle

- Los dos métodos más habituales para identificar a un usuario son:
  - Identificación por la base de datos
  - Identificación por el sistema operativo (autentificación externa)

Identificación por el Sistema Operativo

- Oracle no verifica la contraseña, únicamente controla que el nombre de usuario, a nivel de sistema operativo, corresponde a un nombre de usuario en la base de datos.
- Para identificar al usuario mediante el S.O. hay que establecer el parámetro **OS_AUTHENT_PREFIX** y usar el prefijo en los nombres de usuario de Oracle

Ejemplo: si OS_AUTHENT_PREFIX = OPS$ (valor por defecto), y un usuario autentificado por el S.O. como 'usu1' intenta conectar con la BD, Oracle comprueba si existe un usuario de la BD llamado 'OPS$usu1' y, si es así, permite la conexión

5.3. Gestión de usuarios

- Cuando se crea un usuario, el objetivo es establecer una cuenta segura y útil que tenga los **privilegios** y los **parámetros** adecuados

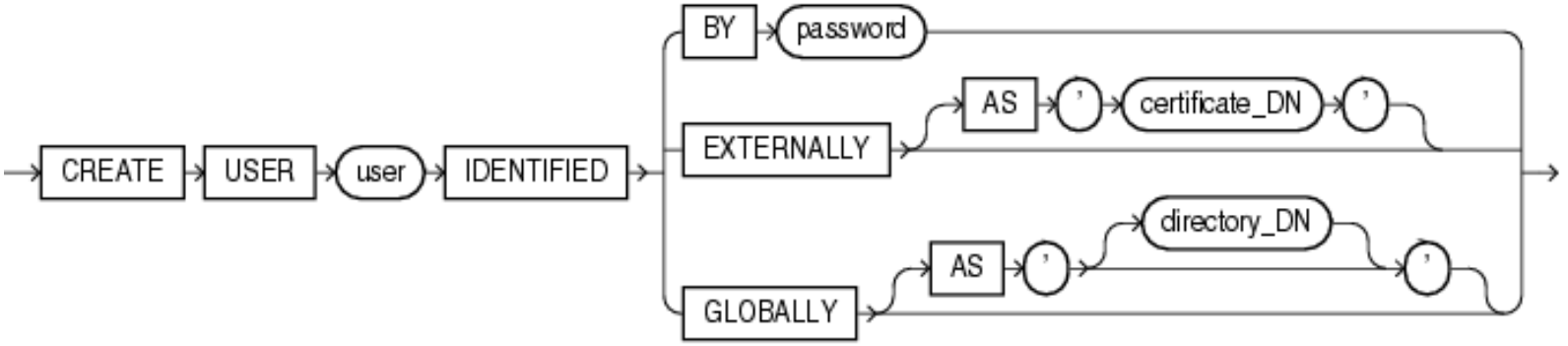
- Al crear una cuenta, ésta no tendrá ninguna función y los usuarios no podrán ni siquiera conectarse a la base de datos hasta que se le conceda los privilegios necesarios

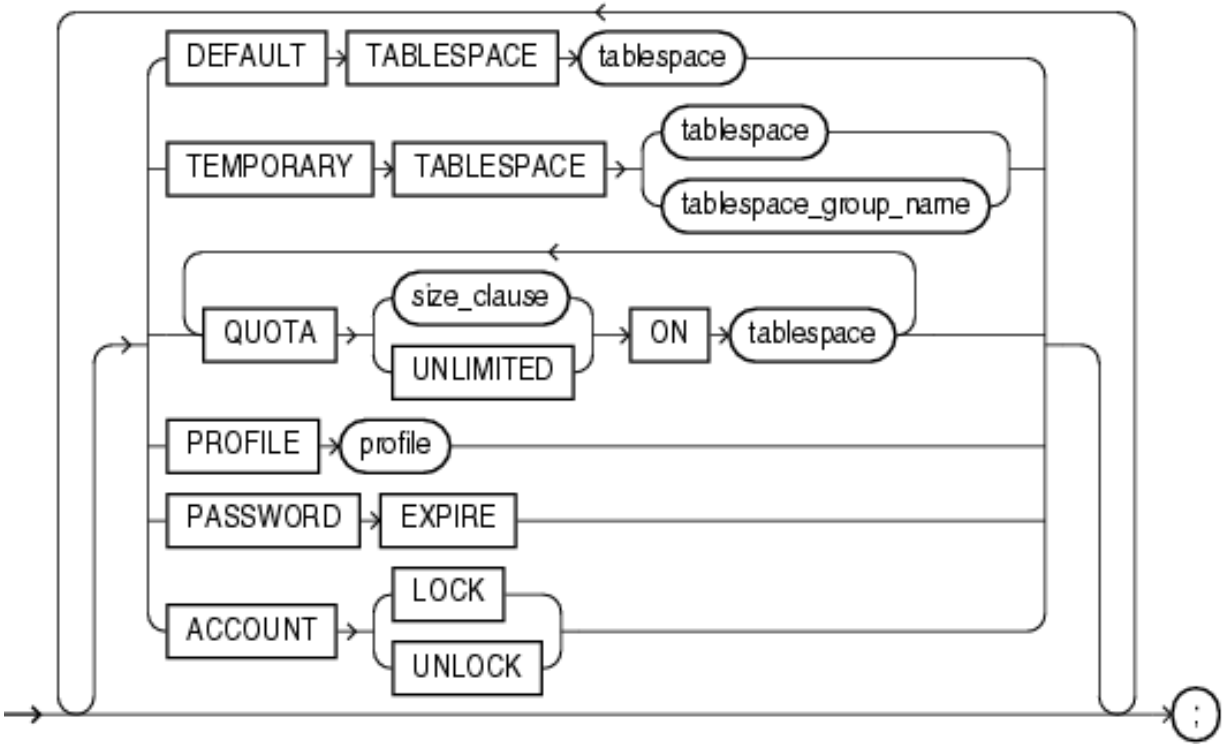
- Para poder crear un usuario es necesario tener el privilegio de sistema **CREATE USER**. Después de crear un usuario hay que concederle, al menos, el privilegio de **CREATE SESSION** o el rol **CONNECT** (los roles se explicarán más adelante)

- Todos los parámetros de una cuenta de usuario pueden especificarse en la sentencia CREATE USER

http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_8003.htm#SQLRF01503

Creación de usuarios





- Sentencia SQL:

```
CREATE USER <nombre>
IDENTIFIED BY <contraseña> | EXTERNALLY
[ DEFAULT TABLESPACE <espacio de tablas> ]
[ TEMPORARY TABLESPACE <espacio de tablas> ]
[ QUOTA entero [K | M] | UNLIMITED ON <espacio de tablas> ]
[ PROFILE <perfil> ]
[ PASSWORD EXPIRE ]
[ ACCOUNT LOCK | UNLOCK ];
```

- Significado de los parámetros (I):
  - **nombre**: nombre del usuario que se va a crear. El nombre de usuario sólo puede contener caracteres del juego de caracteres de la base de datos y puede tener como máximo 30 bytes de longitud. Dentro de una BD el nombre debe ser único. Cada usuario tiene asociado un esquema y dentro del mismo cada objeto debe tener un nombre único
  - **autenticación**: método que utiliza Oracle para autenticar el usuario.
    - **Contraseña**. Se necesita una contraseña para la conexión
    - **Externo**. Especifica que el sistema operativo verifica el usuario
  - **default tablespace**: especifica el espacio de tablas donde se crearán los objetos del esquema del usuario cuando, en la sentencia de creación del objeto no se indique ninguno en particular
    - Si se omite, los objetos se crean en el espacio de tablas permanente por defecto que se establece mediante la sentencia ***alter database default tablespace ...;*** Si éste no se ha especificado, el espacio por defecto es **SYSTEM** (¡¡ NO SE DEBE HACER !!)
  - **temporary tablespace**: especifica el espacio de tablas donde se almacenarán los segmentos temporales requeridos por el usuario para, por ejemplo, operaciones de ordenación
    - Si se omite, el espacio de tablas temporal por defecto es el **SYSTEM** (¡¡ NO SE DEBE HACER !!) a menos que se haya fijado el valor del espacio de tablas temporal por defecto mediante la sentencia ***alter database default temporary tablespace ...;***

- Significado de los parámetros (II):
  - **quota**: indica la cantidad de espacio que un usuario puede disponer en un determinado espacio de tablas. Por defecto, un usuario no tiene cuota en ningún espacio de almacenamiento. Una sentencia **CREATE USER** puede tener **múltiples cláusulas quota** para múltiples espacios de tablas. **UNLIMITED** permite al usuario reservar espacio sin límite
    - En la práctica, sólo es necesario dar cuotas a los usuarios que tengan necesidad de crear segmentos (desarrolladores, usuarios avanzados, la cuenta sobre la que va a ejecutarse una determinada aplicación, etc.)
  - **profile**: especifica el perfil que se desea asignar al usuario. Si se omite, se asigna el perfil DEFAULT (los perfiles se explicarán más adelante)
  - **password expire**: obliga al usuario a cambiar la contraseña en la primera conexión con la base de datos
  - **account**: estado de la cuenta
    - **lock**. Bloquea la cuenta de usuario y desactiva el acceso a la cuenta
    - **unlock**. Desbloquea la cuenta de usuario y activa el acceso a la cuenta (valor por defecto)

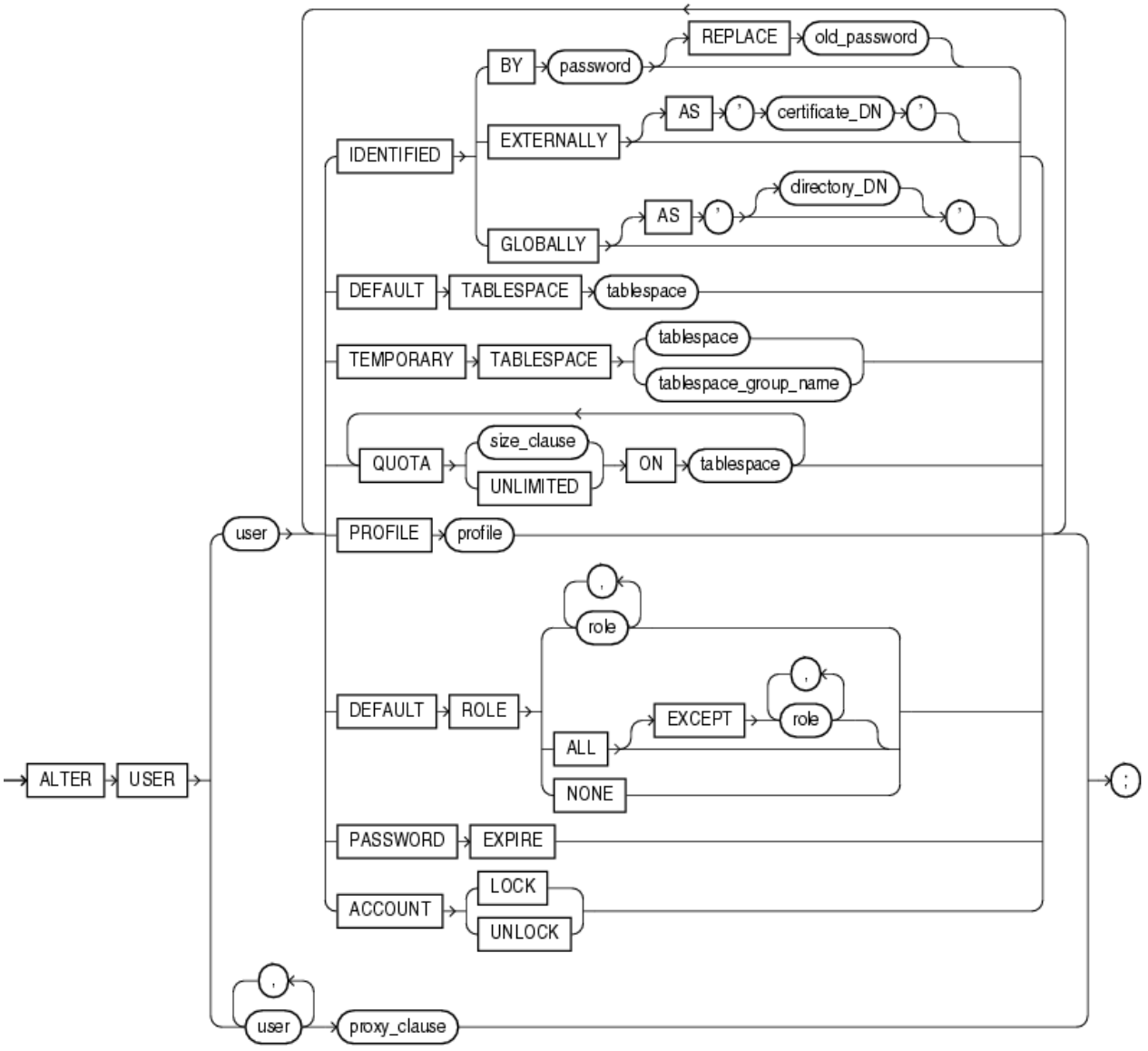
- Ejemplos de sentencias SQL para crear usuarios:

```
CREATE USER usuario_1
    IDENTIFIED BY password
    DEFAULT TABLESPACE ET_usuarios
    QUOTA 10M ON ET_usuarios
    TEMPORARY TABLESPACE temp
    PROFILE perfil_usuarios
    PASSWORD EXPIRE;
```

```
CREATE USER ops$usuario_2
   IDENTIFIED EXTERNALLY
   DEFAULT TABLESPACE ET_usuarios
   QUOTA 5M ON ET_usuarios
   PROFILE app_user;
```

Modificación de usuarios

- Entre otras cosas, es posible modificar las propiedades de seguridad de los usuarios o quitar a un usuario el derecho a consumir más cantidad de espacio en un espacio de tablas
- Se realiza mediante la sentencia **ALTER USER**
- Los cambios producidos afectarán a las futuras sesiones pero no a la actual



http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_4003.htm#SQLRF53555

- Ejemplos de sentencias de modificación de usuarios:

```
ALTER USER usuario_1
   DEFAULT TABLESPACE ET_usuarios_antiguos
   QUOTA 5M ON ET_usuarios_antiguos;
```

NOTA: el usuario **usuario_1** podrá seguir creando objetos en el espacio de tablas **ET_usuarios**, pero deberá especificarlo explícitamente en sentencias como **CREATE TABLE** o **CREATE INDEX**

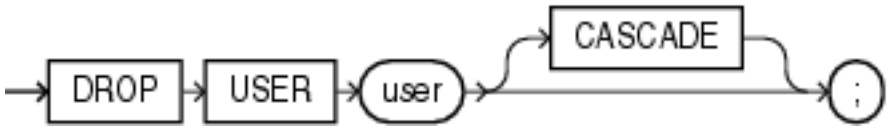
```
ALTER USER usuario_3
   IDENTIFIED BY nueva_clave
   PASSWORD EXPIRE;
```

Para modificar la contraseña de un usuario

```
ALTER USER usuario_7
   ACCOUNT LOCK;
```

Para bloquear la cuenta de un usuario

Eliminación de usuarios



- Un usuario puede ser eliminado mediante la sentencia **DROP USER**

- Mediante la cláusula **CASCADE** se elimina el usuario y todos sus objetos asociados, así como las claves ajenas que dependan de sus tablas
  - Implicaciones que tiene sobre otros esquemas el borrado del usuario y de su propio esquema:
    - Se invalidan vistas o sinónimos para objetos en el esquema borrado
    - Se invalidan procedimientos almacenados, funciones, o paquetes que consulten objetos pertenecientes al esquema eliminado
    - Las vistas materializadas en otros esquemas basados en tablas pertenecientes al esquema borrado no podrán refrescarse
    - Se borran todos los disparadores del esquema
    - No se eliminan roles creados por el usuario

http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9008.htm#SQLRF54874

Información sobre los usuarios

- Las principales vistas para obtener información de los usuarios son **DBA_USERS** y **DBA_TS_QUOTAS** y sus columnas más significativas son:

  - **DBA_USERS**:
    - **USERNAME**: nombre del usuario
    - **USER_ID**: identificador de usuario
    - **PASSWORD**: contraseña (encriptada) del usuario
    - **ACCOUNT_STATUS**: estado de la cuenta
    - **LOCK_DATE**: fecha de bloqueo de la cuenta (si está bloqueada)
    - **EXPIRY_DATE**: fecha de caducidad de la contraseña
    - **DEFAULT_TABLESPACE**: espacio de tablas por defecto del usuario
    - **TEMPORARY_TABLESPACE**: espacio de tablas temporal del usuario
    - **CREATED**: fecha de creación del usuario
    - **PROFILE**: perfil del usuario

- **DBA_TS_QUOTAS**:
  - **TABLESPACE_NAMES**: nombre del espacio de tablas
  - **USERNAME**: nombre del usuario que tiene una cuota en el espacio de tablas
  - **BYTES**: espacio, en bytes, utilizado actualmente por el usuario
  - **MAX_BYTES**: cuota, en bytes, del usuario en el espacio de tablas (-1 si cuota es UNLIMITED)
  - **BLOCKS**: espacio, en bloques, actualmente utilizados por el usuario
  - **MAX_BLOCKS**: cuota, en bloques, del usuario en el espacio de tablas (-1 si cuota es UNLIMITED)

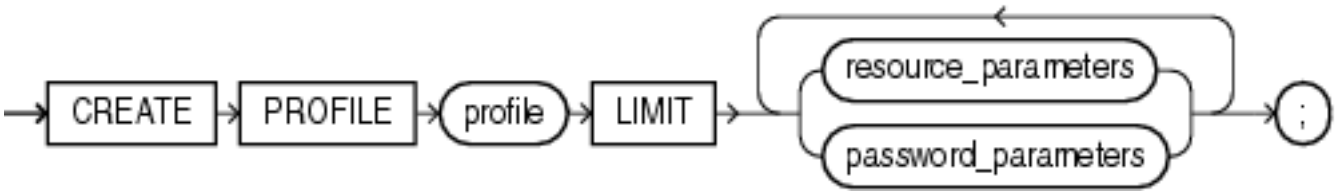
5.4. Gestión de recursos con perfiles (*profiles*)

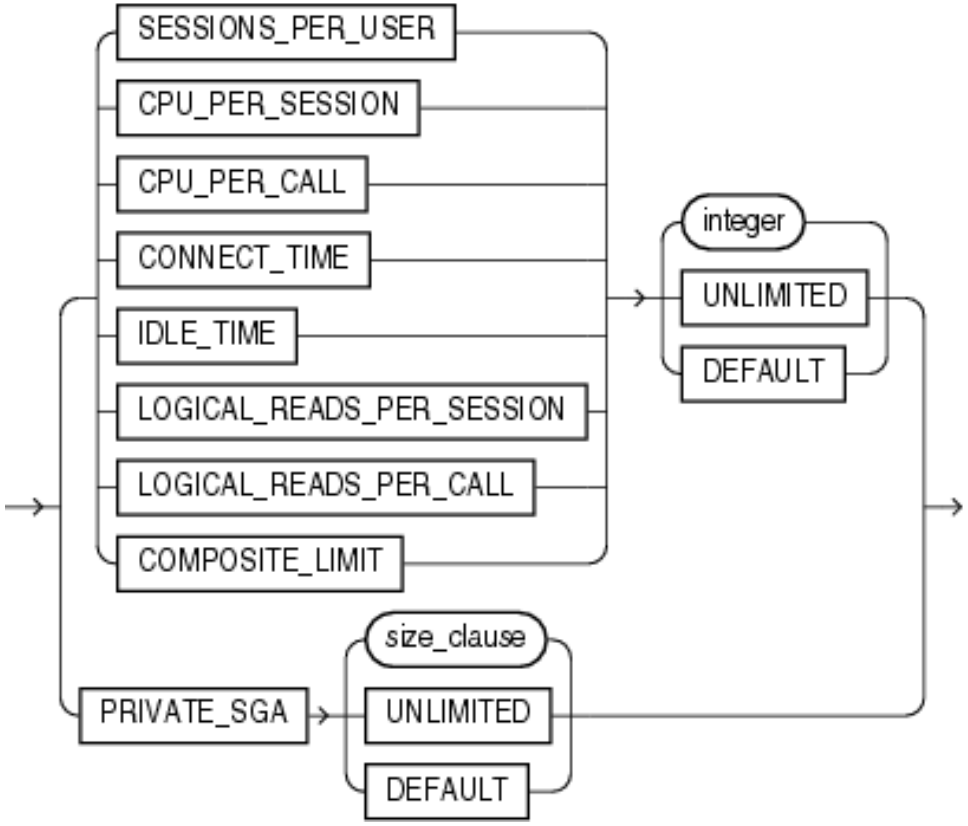
- Es la forma más sencilla de gestionar los recursos puestos a disposición de los usuarios de la base de datos
- Un **perfil** es un conjunto de límites de recursos con un nombre asignado. A cada usuario se le puede asignar un perfil. Además, se puede asignar un perfil por defecto para todos los usuarios de la BD que no posean perfil
- Los perfiles permiten gestionar tareas como:
  - Limitar el consumo de recursos del sistema
  - Limitar el consumo de recursos de la base de datos
  - Poner restricciones relativas a las contraseñas
- Cada BD crea, de forma automática, el perfil **DEFAULT**. Los límites de este perfil se utilizan en los siguientes casos:
  - cuando a un usuario no se le asigna explícitamente un perfil
  - para los límites no especificados en un perfil

- Inicialmente todos los límites del perfil **DEFAULT** están definidos como **UNLIMITED**
- Para prevenir el consumo ilimitado de recursos, el ABD debe modificar los límites del perfil **DEFAULT** después de la creación de la BD (sentencia **ALTER PROFILE**)
- La asignación de un perfil se puede realizar durante la creación del usuario (**CREATE USER**) o posteriormente (**ALTER USER**)
- Un usuario sólo puede tener un perfil asignado a la vez

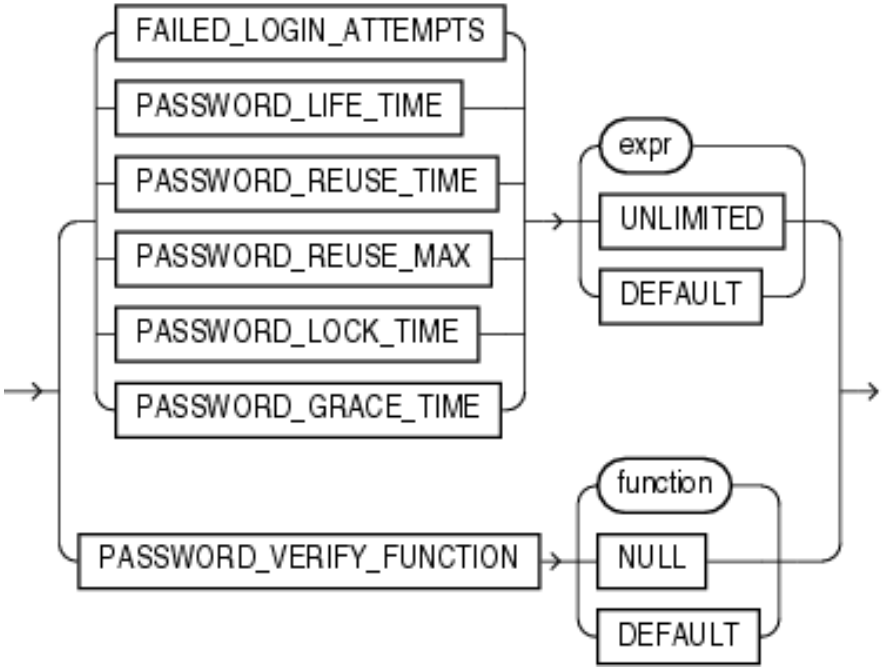
- Por defecto, el control de la limitación de los recursos no está activa (las funcionalidades de gestión de contraseñas siempre están habilitadas). Habilitar o deshabilitar la limitación de recursos mediante perfiles puede hacerse de dos formas:
  - Mediante el parámetro de inicialización **RESOURCE_LIMIT**, asignando valores **TRUE** o **FALSE** (por defecto)
  - Mediante la sentencia **ALTER SYSTEM SET RESOURCE_LIMIT = TRUE | FALSE**

Creación de perfiles





resource_parameters



password_parameters

http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_6010.htm#SQLRF01310

- Para crear un perfil es necesario tener el privilegio de sistema **CREATE PROFILE**

- Significado de los parámetros de recursos (I):
  - Cada uno de estos parámetros puede tomar un **valor entero**, **UNLIMITED** o **DEFAULT**:
    - **SESSIONS_PER_USER:** número máximo de sesiones simultáneas permitidas para un usuario
    - **CPU_PER_SESSION:** cantidad total de tiempo de CPU permitido en una sesión. El límite se expresa en centésimas de segundos
    - **CPU_PER_CALL:** límite de tiempo de CPU permitido para una llamada (análisis, ejecución o recuperación). El límite se expresa en centésimas de segundos
    - **CONNECT_TIME:** tiempo máximo permitido para una sesión. El límite se expresa en minutos

Significado de los parámetros de recursos (II):

- **IDLE_TIME:** tiempo máximo de inactividad continua permitido en una sesión. Las consultas y otras operaciones de larga ejecución no están sujetas a este límite. El límite se expresa en minutos

- **LOGICAL_READS_PER_SESSION:** número total de lecturas de bloques de datos permitidas en una sesión. El límite incluye los bloques leídos desde la memoria y desde el disco

- **LOGICAL_READS_PER_CALL:** número máximo de lecturas de bloques de datos permitidas para una llamada (análisis, ejecución o recuperación) para procesar una sentencia SQL

- **PRIVATE_SGA:** cantidad máxima de espacio privado que una sesión puede asignar en el conjunto compartido del Área Global del Sistema (SGA). El límite de SGA privada se aplica sólo si está utilizando una arquitectura de servidor compartido. El límite se expresa en K o M

- **COMPOSITE_LIMIT:** coste de recursos total para una sesión. El coste de recursos para una sesión es la suma ponderada del tiempo de CPU utilizado en la sesión, el tiempo de conexión, el número de lecturas realizadas en la sesión y la cantidad de espacio de SGA privada asignado

- Significado de los parámetros de contraseñas (I):
  - **FAILED_LOGIN_ATTEMPTS:** limita el número de intentos fallidos de conexión antes de bloquear la cuenta
  - **PASSWORD_LIFE_TIME:** número de días de validez de la contraseña. La contraseña caduca si no se cambia en este periodo. Si se indica un valor para PASSWORD_GRACE_TIME, la clave expira si no se cambia en este periodo. Si no se indica valor para PASSWORD_GRACE_TIME (por defecto UNLIMITED), se genera un aviso pero el usuario puede seguir conectándose
  - **PASSWORD_REUSE_TIME:** número mínimo de días que deben pasar para poder utilizar otra vez la misma contraseña
  - **PASSWORD_REUSE_MAX:** número de cambios de contraseña necesarios para poder volver a usar la misma contraseña

NOTA: PASSWORD_REUSE_TIME y PASSWORD_REUSE_MAX se deben usar conjuntamente. Si se indica un entero para ambos parámetros, el usuario no puede reutilizar la contraseña hasta que ha cambiado el número de veces indicado en PRM durante el periodo indicado por PRT. Si alguno de los dos tiene valor UNLIMITED, nunca se podrá reutilizar la contraseña

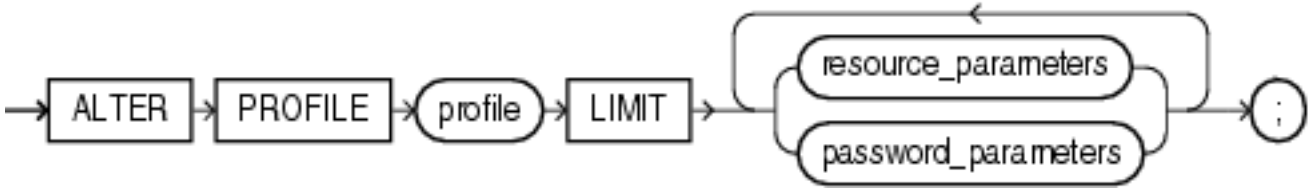
```
CREATE PROFILE mi_perfil LIMIT
   PASSWORD_REUSE_TIME 20
   PASSWORD_REUSE_MAX 5;
```

Los usuarios con este perfil podrán reutilizar su contraseña después de 20 días, siempre que la contraseña haya sido modificada al menos 5 veces

- Significado de los parámetros de contraseñas (II):
  - **PASSWORD_LOCK_TIME:** especifica el número de días que la cuenta está bloqueada después de fallar el número especificado de intentos de conexión. Después de este periodo de tiempo, la cuenta se desbloqueará de manera automática. Si se especifica UNLIMITED, sólo el administrador de la base de datos puede desbloquear la cuenta. Valor 1 en el perfil DEFAULT
  - **PASSWORD_GRACE_TIME:** periodo de gracia donde se permite la conexión pero se notifica la necesidad de cambiarla. Si no se cambia dentro de este periodo de tiempo, la cuenta caduca y la contraseña deberá ser cambiada para que el usuario pueda iniciar una sesión
  - **PASSWORD_VERIFY_FUNCTION:** permite que se utilice un bloque PL/SQL para la verificación de contraseña cuando los usuarios a los que se les asigna este perfil se conectan a una base de datos. Si se indica NULL no se usa función alguna

Modificación de perfiles

- La asignación de límites de recursos puede modificarse con la sentencia SQL **ALTER PROFILE**, estando en posesión del privilegio correspondiente. Los valores modificados no afectan a las sesiones en curso



http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_2007.htm#SQLRF00813

Eliminación de perfiles

- La asignación de límites de recursos puede eliminarse con la sentencia SQL **DROP PROFILE**, estando en posesión del privilegio correspondiente. El borrado de un perfil no afecta a las sesiones en curso

- Para eliminar un perfil que está asignado al menos a un usuario debe usarse la opción CASCADE
- Si se borra un perfil asociado a un usuario, a éste se le asigna, de forma automática, el perfil DEFAULT

http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_8026.htm#SQLRF01529

Tablas y vistas relacionadas con usuarios y perfiles

ALL_USERS	Usuarios visibles al usuario actual
DBA_TS_QUOTAS USER_TS_QUOTAS	Cuotas de espacio para usuarios
DBA_USERS	Usuarios de la base de datos
USER_PASSWORD_LIMITS	Parámetros de contraseña asignados al usuario
USER_RESOURCE_LIMITS	Parámetros de recursos asignados al usuario
USER_USERS	Usuarios de la base de datos
V$SESSION	Información sobre sesiones
V$SESSTAT	Estadísticas de sesión
DBA_PROFILES	Información acerca de los perfiles

5.5. Administración de privilegios

- Un privilegio es un derecho para ejecutar un tipo particular de sentencia SQL o acceder a un objeto de otro usuario
- Ejemplos:
  - derechos para conectar con la base de datos
  - derechos para crear una tabla
  - derechos para seleccionar tuplas de una tabla perteneciente a otro usuario
- Los privilegios se otorgan a los usuarios para que estos puedan realizar las tareas requeridas por su trabajo con la base de datos
- Es recomendable concederlos únicamente cuando un determinado privilegio es absolutamente necesarios para efectuar una tarea

- Un usuario puede recibir privilegios de dos formas distintas:
  - **de forma explícita,** otorgando un derecho concreto a un usuario particular. Por ejemplo, el usuario "*antúnez*" puede recibir el privilegio de insertar tuplas en la tabla NÓMINAS, que es propiedad del usuario "*martín*"
  - **por medio de roles,** que pueden ser otorgados a distintos usuarios de la base de datos. Por ejemplo, los privilegios de seleccionar, insertar, actualizar y eliminar tuplas en la tabla NÓMINAS pueden ser otorgados al **rol** llamado GESTIÓN, el cual puede ser otorgado a los usuarios "*gest_001*" y "*gest_004*"

- La gestión de roles es la vía más sencilla y segura de gestionar los privilegios de los distintos usuarios
- Existen dos categorías de privilegios:
  - **de sistema:** permiten realizar determinadas acciones en la base de datos (crear espacios de almacenamiento, crear usuarios, …) o en cualquier esquema
  - **de objeto:** permiten a los usuarios acceder y manipular o ejecutar objetos concretos (tablas, vistas, secuencias, procedimientos, funciones o paquetes)

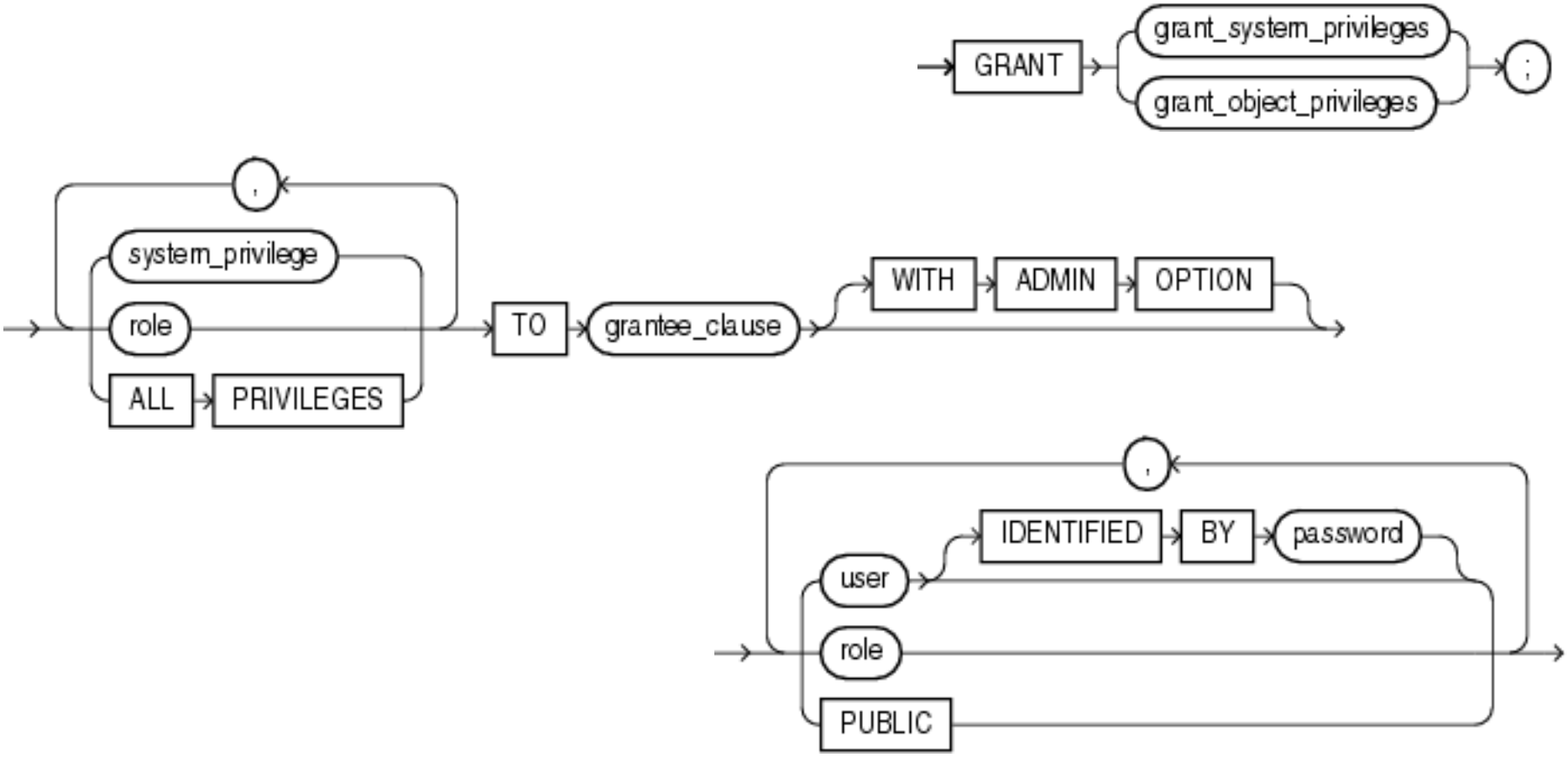
Privilegios de sistema

- Un **privilegio de sistema** permite realizar una acción determinada sobre un tipo de objeto. Existen más de 170 privilegios de sistema en la versión 11g. Todos ellos pueden consultarse en la tabla **SYSTEM_PRIVILEGE_MAP**
- Pueden clasificarse en:
  - Privilegios sobre operaciones del sistema: CREATE SESSION, CREATE TABLESPACE, …
  - Privilegios que permiten el manejo de objetos en el esquema propio de un usuario: CREATE TABLE, CREATE PROCEDURE, …
  - Privilegios que permiten el manejo de objetos en cualquier esquema: CREATE **ANY** TABLE, DELETE **ANY** TABLE, …
- Algunas consideraciones:
  - No existe el privilegio CREATE INDEX (se concede junto con CREATE TABLE)
  - Privilegios como CREATE TABLE o CREATE PROCEDURE incluyen el borrado de dichos objetos

Todos los privilegios de sistema en: http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9013.htm#SQLRF54937

Otorgar y revocar privilegios de sistema

- Se pueden otorgar y revocar privilegios a usuarios y roles mediante las sentencias **GRANT** y **REVOKE** respectivamente



http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9013.htm#SQLRF01603

- **system_privilege** indica cualquiera de los privilegios del sistema
- **role** es el nombre de algún rol definido
- **ALL PRIVILEGES** permite asignar todos los privilegios de sistema a la vez
- La opción **WITH ADMIN OPTION** otorga a los receptores del privilegio el derecho de otorgar y revocar, a su vez, estos privilegios a usuarios o roles, modificar el rol o eliminarlo
- Con una sola sentencia **GRANT** se puede asignar una lista de privilegios del sistema o roles a varios usuarios o roles a la vez o a todos los usuarios (opción **PUBLIC**)

- Para otorgar o revocar privilegios del sistema, se debe tener algún privilegio de sistema con la opción **ADMIN OPTION** o tener el privilegio **GRANT ALL PRIVILEGES**
  - **Ejemplo:** un usuario al que se le haya otorgado el privilegio de sistema CREATE ANY TABLE con la opción WITH ADMIN OPTION tiene la capacidad, a su vez, de otorgar o revocar dicho privilegio a cualquier otro usuario o rol

- Ejemplos de asignación de privilegios de sistema:

```
GRANT CREATE SESSION TO usu_002;
```

Otorga el privilegio de conexión a la base de datos al usuario ***usu_002***

```
GRANT
     CREATE SESSION,
     CREATE PROCEDURE,
     CREATE TABLE
   TO usu_004
   WITH ADMIN OPTION;
```

Otorga varios privilegios al usuario ***usu_004*** de forma que éste también puede otorgar dichos privilegios

- La sentencia **REVOKE** permite revocar un privilegio de sistema
- Para que un usuario pueda revocar un privilegio de sistema o rol se le debe haber concedido el privilegio con la opción WITH ADMIN OPTION
- De la misma forma, se puede revocar cualquier privilegio de sistema teniendo el privilegio de sistema **GRANT ALL PRIVILEGES**
- Algunas consideraciones:
  - Sólo se pueden revocar privilegios que hayan sido concedidos directamente con la sentencia **GRANT**. No puede usarse **REVOKE** para quitar privilegios recibidos mediante la cláusula **PUBLIC**
  - Si un privilegio ha sido asignado a un usuario y a PUBLIC, la revocación del privilegio para el usuario no tiene efecto ya que el usuario puede continuar ejerciendo el privilegio
  - Si se ejecuta para quitar privilegios del sistema a PUBLIC, los privilegios **sólo serán revocados** a aquellos usuarios que los recibieron por medio de una concesión PUBLIC y no a aquellos que lo recibieron explícitamente
  - Cuando varios usuarios han concedido el mismo privilegio a otro usuario, éste último sólo perderá este privilegio si todos los usuarios que se lo concedieron revocan el privilegio al usuario
  - Si se revoca un privilegio de sistema a un usuario, toma efecto inmediatamente

- Ejemplos

```
REVOKE DROP ANY TABLE FROM usu_003, usu_004;
```

A partir de la ejecución de esta sentencia, los usuarios ***usu_003*** y ***usu_004*** no podrán borrar tablas salvo en sus propios esquemas

```
REVOKE ALL PRIVILEGES FROM usu_009;
```

Revoca todos los privilegios a ***usu_009***