

Usuarios, privilegios y roles de **Oracle**

Álvaro González Sotillo

1 de octubre de 2021

Índice

1. Introducción	1
2. <i>Tablespaces</i>	1
3. Usuarios	3
4. Privilegios	4
5. Roles	5
6. Perfiles	6
7. Referencias	7

1. Introducción

- Oracle puede utilizarse simultáneamente por varios procesos y clientes
- Cada uno puede tener distintos permisos y capacidades
 - Espacio de disco disponible
 - Gasto en CPU, red
 - Acceso a diferentes tablas de datos

2. *Tablespaces*

- **Oracle** almacena datos en los *tablespaces*
 - Conjuntos de ficheros
 - Normas para su tamaño: inicial, máximo, crecimiento
- Cada *tablespace* puede usarse para diferentes funciones
 - Datos de usuario o del sistema: *permanent tablespace*
 - Datos de recuperación: *undo tablespace*
 - Datos temporales: *temporary tablespace*

2.1. Recordatorio: Tipos de fichero según su uso

- Permanentes (*permanent*)
 - Datos que deben ser guardados
 - Ejemplo: Empleados contratados, nóminas pagadas, declaraciones de impuestos,...
- De movimiento (*undo*)
 - Cambios que deben ser incluidos en archivos permanentes
 - Ejemplo: un puesto de peaje debe guardar todos los pagos con tarjeta, y enviarlos juntos
- De maniobra (*temporary*)
 - Se utilizan como extensión a la RAM de un ordenador, se borran cuando el proceso termina
 - Ejemplo: caché de disco de los navegadores

2.2. ¿Por qué tantas normas?

- Disponibilidad
 - ¿Es mejor garantizar el espacio para las tablas?
 - ¿Es mejor ahorrar espacio mientras se pueda?
- Velocidad
 - Hacer crecer un fichero es lento
 - Un fichero que ha crecido poco a poco está disperso en el disco (y es más lento)
- Capacidad
 - Cada sistema de ficheros tiene un tamaño de fichero máximo

2.3. *Tablespaces* por defecto

- Por defecto, **Oracle** crea en una nueva base de datos
 - `users`: Tablespace asignado por defecto para los datos de todos los usuarios
 - `system`: Datos acerca de la instancia y del diccionario de datos
 - `sysaux`: Operaciones temporales del administrador que no caben en memoria
 - `undo (undotbs1)`: Datos para deshacer las transacciones (`rollback`)
 - `temp`: Operaciones temporales de usuarios que no caben en memoria

```
select tablespace_name, contents from dba_tablespaces;
```

Mas información en:

- https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7003.htm
- https://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm

2.4. Crear un *tablespace*

```
CREATE TABLESPACE ejemplo_tablespace
DATAFILE
'/tablespaces/ejemplo_1.dbf' SIZE 10M
AUTOEXTEND ON NEXT 200k MAXSIZE 14M,
'/tablespaces/ejemplo_2.dbf' SIZE 10M
AUTOEXTEND ON NEXT 200k MAXSIZE 14M;
```

Más en docs.oracle.com

2.5. ¿Por qué es tan complicado?

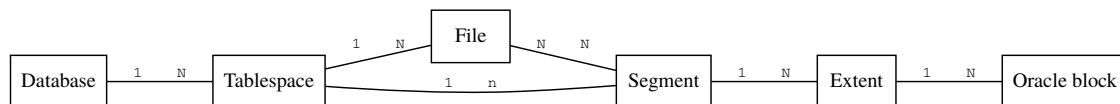
- Esta flexibilidad permite:
 - Que cada usuario tenga sus *tablespaces*
 - Que cada *tablespace* esté en discos distintos (rapidez)
 - Que un *tablespace* se localice en varios discos (rapidez, tamaño)
 - Mover *tablespaces* una vez creados

2.6. Ejercicio: Llena un *tablespace*

- Crea un *tablespace* con un tamaño inicial de 10MB, y un tamaño máximo de 14MB
- Crea una tabla sobre el *tablespace*
- Inserta datos en la tabla hasta conseguir el error ORA-01653

```
create table datos(valor varchar(2048)) tablespace ejemplo_tablespace;
begin for j in 1..1 loop for i in 1..1000
loop insert into datos values( 'datos ' || i ); end loop; commit; end loop; end; /
```

2.7. Conceptos de almacenamiento



Más información en [Oracle.com](https://www.oracle.com)

3. Usuarios

¿Qué usuario hemos utilizado con `sqlplus` hasta ahora?

- **Oracle** tiene dos modos de autenticar usuarios
 - Autenticación de sistema operativo
 - Autenticación con seguridad nativa de **oracle**
- Al instalarlo, elegimos que el grupo `wheel` era administrador

3.1. Creación de usuarios

```
CREATE USER usuario IDENTIFIED BY contraseña
DEFAULT TABLESPACE tablespace
TEMPORARY TABLESPACE tablespace
QUOTA UNLIMITED ON tablespace
QUOTA tamaño ON tablespace
ACCOUNT LOCK
ACCOUNT UNLOCK
```

3.2. Modificación de usuario

- Modificación de un usuario ya creado

```
ALTER USER usuario
cualquier opcion valida al crear usuario
```

- Borrado de usuario

```
DROP USER usuario
```

4. Privilegios

- Cada usuario puede tener unos permisos distintos
- Ya hemos visto dos permisos
 - En qué *tablespaces* se puede escribir
 - Cuántos datos se pueden escribir en esos *tablespaces*
 - Si una cuenta está bloqueada
- Pero hay más permisos
 - Veremos los *privilegios* de **Oracle**

4.1. Privilegios de Oracle

Privilegio	Objeto sobre el que se aplica
Create, alter, drop	Table, sequence, view, user, synonym, session
select, update, delete, insert	Sobre campos de tablas y filas

4.2. Sintaxis de Grant

```
grant PRIVILEGIO1,PRIVILEGIO2,...,PRIVILEGION
on OBJETO
to USUARIO
with grant option;
```

```
create table alumnos (...);
create user profesor ...;
grant select on alumnos to profesor;
```

Fuente: docs.oracle.com

4.3. Ejercicio

- Crea un usuario CONPERMISOS
 - Que tenga privilegios de connect y resource
 - Utilízalo para crear una tabla DATOS (TEXTO varchar2(255), numero integer)
 - Inserta datos (puede que necesite cuota)
- Crea un usuario LIMITADO
- Haz que CONPERMISOS de privilegios a LIMITADO para que:
 - Pueda leer todos los campos de la tabla DATOS
 - Pueda actualizar el campo NUMERO de tabla DATOS
 - Pero no pueda modificar el campo TEXTO, ni borrar filas, ni insertar filas

4.4. Ejercicio

- Haz que el usuario CONPERMISOS tenga una cuota de 100k en el tablespace USERS
- Llena toda su cuota insertando filas en la tabla DATOS
- ¿Qué ocurre?

4.5. Quitar privilegios

- Los privilegios se quitan con revoke
- Cuando un usuario pierde un privilegio, los pierden también todos los que recibieron el mismo privilegio a través de él
 - Por la cláusula with grant option

```
connect sys/*****
grant select on unatabla to unusuario with grant option;

connect unusuario/*****
grant select on unatabla to otrousuario;

connect sys/*****
revoke select on unatabla from unusuario;

-- AQUI NI unusuario NI otrousuario TIENEN PRIVILEGIO SOBRE unatabla
```

5. Roles

- Asignar todos los privilegios a un usuario es trabajoso, pero factible
- ¿Qué ocurre si tenemos que manejar a muchos usuario?
- Los **roles** permiten dar nombre a un grupo de privilegios
 - Se pueden asignar privilegios a un rol
 - Y después asignar ese rol a varios usuarios

5.1. Sintaxis de roles

```
create role NOMBREROL;  
grant PRIVILEGIOS on OBJETOS to NOMBREROL;  
grant NOMBREROL to USUARIO;
```

Fuente: docs.oracle.com

5.2. Ejercicio

- Imagina que
 1. Creas un rol con sus permisos
 2. Le asignas privilegios
 3. Lo asignas al usuario USUARIOANTES
 4. Quitas algún privilegio del rol
 5. Asignas el rol al usuario USUARIODESPUES
- El usuario USUARIODESPUES, ¿tiene más, menos o los mismos privilegios que USUARIOANTES?
 - O lo que es lo mismo, ¿los permisos del rol se *copian* al usuario o se *enlazan*?

6. Perfiles

- Un *profile* es un conjunto de limitaciones sobre el sistema **Oracle**
- No limita acceso a datos, sino al propio SGBD y sistema operativo

6.1. Creación de perfiles

```
CREATE PROFILE nombreperfil LIMIT  
SESSIONS_PER_USER          UNLIMITED  
CPU_PER_SESSION             UNLIMITED  
CPU_PER_CALL                 3000  
CONNECT_TIME                45  
IDLE_TIME                    300  
LOGICAL_READS_PER_SESSION   DEFAULT  
LOGICAL_READS_PER_CALL      1000  
PRIVATE_SGA                  15K  
COMPOSITE_LIMIT              5000000;  
  
ALTER SYSTEM SET resource_limit = TRUE scope = BOTH
```

- Nota: Según la fuente, los tiempos se miden en días. Se pueden especificar fracciones de día.
 - Pero a mí me funcionan como minutos

Fuente: docs.oracle.com

6.2. Asignación de perfil a un usuario

- En la creación (`create user`), o posteriormente

```
alter user USUARIO profile NOMBREDEPERFIL
```

6.3. Ejercicio

- Haz que el usuario `LIMITADO`
 - se quede sin sesión tras 1 minuto de inactividad
 - se quede sin sesión a los 2 minutos de conectarse, aunque no haya estado inactivo

6.4. Ejercicio

- Utiliza las vistas de **Oracle** para conocer los límites del profile por defecto.

6.5. Ejercicio

- Usa la opción `PASSWORD_VERIFY_FUNCTION` para evitar que las contraseñas sean más largas de tres caracteres

```
Create or Replace Function trescaracteresomenos (in_username in varchar2, in_newpassword in varchar2, in_oldpassword
in varchar2) return boolean as pwd_okay boolean; begin -- * -- you can write your own password validation here
... -- * if LENGTH(in_newpassword) >3 then raise_application_error(-20001, 'LA CONTRASEÑA ES MUY LARGA');
end if; return true; end; /
```

```
CREATE PROFILE EJEMPLO LIMIT CONNECT_TIME 1 IDLE_TIME 1 PASSWORD_VERIFY_FUNCTION
trescaracteresomenos;
```

7. Referencias

- Formatos:
 - [Transparencias](#)
 - [PDF](#)
 - [EPUB](#)
- Creado con:
 - [Emacs](#)
 - [org-re-reveal](#)
 - [Latex](#)
- Alojado en [Github](#)