

Usuarios, privilegios y roles de **Oracle**

Álvaro González Sotillo

October 10, 2024

Contents

1	Introducción	1
2	<i>Tablespaces</i>	1
3	Usuarios	4
4	Privilegios	4
5	Roles	6
6	Perfiles	7
7	Referencias	8

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;
```

Más 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 Ejercicio: *tablespace* por defecto

- Consulta el *tablespace* por defecto de los usuarios (`dba_users`)
- Cambia el *tablespace* por defecto de un usuario (`alter user`)
- Consulta el *tablespace* por defecto por (`DATABASE_PROPERTIES`)
- Cambia el *tablespace* por defecto por defecto (`alter database`)
 - Nota: esto cambia el *tablespace* por defecto de todos los usuarios existentes
- Cambiar una tabla de *tablespace*: `ALTER TABLE ... MOVE TABLESPACE`

```
ALTER TABLE <TABLE NAME to be moved> MOVE TABLESPACE <destination TABLESPACE NAME>
```

2.8 Ejercicio: más *tablespaces*

1. Crea un *tablespace* PRUEBA1
 - inicialmente 10M, máximo 20M
2. Crea un usuario
 - no le digas *tablespace* por defecto
 - pero que al crearlo su *tablespace* por defecto sea PRUEBA1
3. Crea una tabla
 - `MISDATOS (DATOS VARCHAR(255))`
4. Llena la tabla hasta que no quede espacio en PRUEBA1
5. Crea un *tablespace* PRUEBA2
 - tamaño inicial igual al máximo, 30M
6. mueve MISDATOS a PRUEBA2
7. Vuelve a llenar la tabla hasta que se llene PRUEBA2

```
create tablespace PRUEBA1 DATAFILE '/tablespaces/prueba1.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M
MAXSIZE 20M;
```

```
ALTER DATABASE DEFAULT TABLESPACE prueba1;
```

```
alter session set "oraclescript"=true;
```

```
create user usuarioprueba1 identified by a; grant connect,resource to usuarioprueba1; alter user usuarioprueba1
quota unlimited on prueba1;
```

```
connect usuarioprueba1/a create table misdatos(datos varchar(255));
```

```
select * from dba_tables where TABLE_NAME='MISDATOS';
```

```
alter session set current_schema=prueba1;
```

```
BEGIN LOOP insert into misdatos values('hola'); commit; end loop;
```

```
END; /
```

```
create tablespace PRUEBA2 DATAFILE '/tablespaces/prueba2.dbf' SIZE 30M;
```

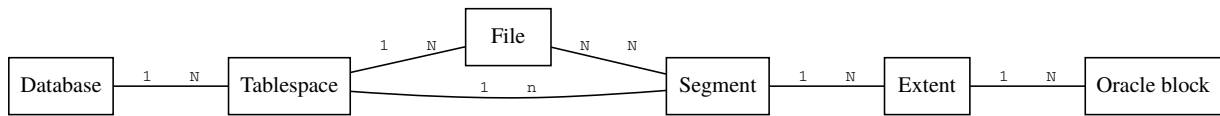
```
alter user usuarioprueba1 quota unlimited on prueba2;
```

```
connect usuarioprueba1/a alter table misdatos move tablespace prueba2;
```

```
begin LOOP for i in 1 .. 10 loop insert into misdatos values('hola'); end loop; commit; end loop;
```

```
END; /
```

2.9 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 autentificar usuarios
 - Autenticación de sistema operativo (parámetro `os_authent_prefix`)
 - Autenticación con seguridad nativa de **oracle**
- Al instalarlo, elegimos que el grupo `wheel` era administrador

```
SQL> SHOW PARAMETER OS_authent_prefix
```

```
NAME TYPE VALUE OS_authent_prefix string ops$ SQL>  
- UNIX CREATE USER ops$tim_hall IDENTIFIED EXTERNALLY; GRANT CREATE SESSION TO ops$tim_hall;
```

3.1 Creación de usuarios

```
CREATE USER usuario IDENTIFIED BY contrasena  
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 select, update, delete, insert	Table, sequence, view, user, synonym, session, procedure 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;

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

4.6 Privilegios de sistema

- No se aplican sobre tablas/objetos concretos
 - CREATE ANY TABLE, CREATE_ANY_VIEW,...
 - CREATE SESSION, CREATE SYNONYM,..
 - UNLIMITED TABLESPACE, SYSDBA,...
 -

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 NOMBROL;  
grant PRIVILEGIOS on OBJETOS to NOMBROL;  
grant NOMBROL to USUARIO;
```

Fuente: docs.oracle.com

5.2 Ejercicio

Se pueden asignar privilegios a PUBLIC, para que todos los usuarios tengan dicho privilegio. Decide si PUBLIC es un usuario o un rol, y compruébalo en las tablas de diccionario.

5.3 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*?

5.4 ¿Qué privilegios tengo?

- Un usuario puede tener muchos permisos otorgados directamente y a través de un rol
- Además, **algunos roles son por defecto**, pero otros hay que activarlos con **SET ROLE**

```
select * from session_roles;  
select * from session_privs;
```

5.5 Consulta recursivas

```
select * from dba_role_privs connect by prior granted_role = grantee start with grantee = '&USER' order by 1,2,3;

select * from dba_sys_privs where grantee = '&USER' or grantee in (
  select granted_role from dba_role_privs connect by prior granted_role = grantee start with grantee = '&USER'
)
order by 1,2,3;

select * from dba_tab_privs where grantee = '&USER' or grantee in (
  select granted_role from dba_role_privs connect by prior granted_role = grantee start with grantee = '&USER'
)
order by 1,2,3,4;
```

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                  1
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

7 Referencias

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