

=====

Perfiles, roles y usuarios en Oracle 12c (12.2.0.1)

=====

Autor: M.IT.I Ambrosio Cardoso Jiménez

Fecha: 20-Enero-2020

Objetivo: Crear y manipular perfiles, grupos y usuarios en Oracle 12

Introducción.

La Seguridad es una de las tareas importantes que tiene un DBA y esta se realiza siguiendo una serie de estándares que incluyen estándares legales y de negocio.

La seguridad de los datos es uno de los aspectos que todo negocio debe de reconocer como algo importante; además, deberá desarrollar políticas acordes que les permitan proteger los datos de forma adecuada. Una de las formas de proteger los datos es por medio de privilegios que se conceden a los usuarios y que en el momento que el usuario realiza una conexión con la base de datos, estos privilegios concedidos se van a habilitar para permitirle realizar aquellas acciones que el usuario requiera.

La base de datos tiene dos tipos de privilegios:

- De sistema: estos privilegios le permiten al usuario realizar acciones específicas sobre la base de datos.
- De objetos: estos privilegios le permiten al usuario acceder y manipular objetos específicos.

Una de las preocupaciones de muchos administradores de base de datos es la posibilidad de que usuarios de aplicación y de base de datos tengan privilegios excesivos; más aún, que algunos de ellos no se estén utilizando; pero que en algún momento podrían ser aprovechados para ejecutar sentencias SQL que puedan dañar la base de datos o, por el contrario, extraer información para usos no deseados.

Perfiles.

- ◆ Un perfil de usuario es una forma de limitar los recursos que puede utilizar un usuario.
- ◆ Cada usuario puede tener un único perfil.
- ◆ Antes de asignar un perfil a un usuario es necesario que este perfil exista en la base de datos.
- ◆ Un perfil se asigna en la creación de un usuario **CREATE USER** o modificándolo **ALTER USER**

Los recursos que limitamos son recursos del kernel: uso de la CPU, duración de sesión, entre otros, también límites de uso de las claves de acceso (passwords): duración, intentos de acceso, reuso.

| Parámetros | Significado |
|----------------------------------|--|
| PASSWORD_LIFE_TIME | Tiempo de vida del Password |
| PASSWORD_GRACE_TIME | Número de días disponibles para cambiar el password antes de que se bloquee la cuenta. |
| PASSWORD_REUSE_TIME | Número de días que tienen que pasar para poder reutilizar un password. |
| PASSWORD_REUSE_MAX | Cantidad de días que debe transcurrir para reutilizar el mismo password |
| FAILED_LOGIN_ATTEMPTS | Número determinado de veces que el login del usuario puede fallar |
| PASSWORD_LOCK_TIME | Numero de días transcurridos para que la cuenta se bloquee |
| COMPOSITE_LIMIT | Suma del máximo de CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION y PRIVATE_SGA. Si este limite es excedido, Oracle aborta la sesión y regresa un error. |
| CONNECT_TIME | Tiempo permitido de conexión por sesión en minutos. |
| CPU_PER_CALL | Máximo tiempo de CPU por llamada en centésimas de segundo. |
| CPU_PER_SESSION | Máximo tiempo de CPU por sesión en centésimas de segundo. |
| IDLE_TIME | Tiempo máximo permitido sin actividad por el usuario antes de ser desconectado. Se expresa en minutos. |
| LOGICAL_READS_PER_CALL | Máximo número de bloques de base de datos leídos por llamada. |
| LOGICAL_READS_PER_SESSION | Máximo numero de bloques de base de datos leídos por sesión. |
| PRIVATE_SGA | Máxima cantidad de bytes de espacio privado reservado en la SGA. Se puede expresar en el formato enteroK para kilobytes o enteroM para megabytes. |
| SESSIONS_PER_USER | Máximo número de sesiones concurrentes permitidas por usuario |

```

CREATE PROFILE miperfil LIMIT
SESSIONS_PER_USER          2      -- 2 sesiones abiertas
CPU_PER_SESSION            10000   -- decimas de segundo
CPU_PER_CALL               1      -- decimas de segundo
CONNECT_TIME               UNLIMITED -- minutos
IDLE_TIME                  30      -- minutos
LOGICAL_READS_PER_SESSION  DEFAULT -- DB BLOCKS
LOGICAL_READS_PER_CALL    DEFAULT -- DB BLOCKS
-- COMPOSITE_LIMIT         DEFAULT --
PRIVATE_SGA                20M    --
FAILED_LOGIN_ATTEMPTS      3      -- 3 reintentos
PASSWORD_LIFE_TIME         30     -- dias
PASSWORD_REUSE_TIME        12     --
PASSWORD_REUSE_MAX         UNLIMITED --
PASSWORD_LOCK_TIME         DEFAULT -- dias
PASSWORD_GRACE_TIME        2      -- dias
PASSWORD_VERIFY_FUNCTION   NULL;

```

Ejemplo:

```
CREATE PROFILE lim_prueba
```

```
LIMIT idle_time          2
```

```
failed_login_attempts 3;
```

Puede estar hasta 2 minutos sin hacer nada, transcurrido ese tiempo se desconectará automáticamente y solo puede fallar 3 veces en la autenticación

Crear usuario y asignar perfil **lim_prueba**

```

CREATE USER usrtmp PROFILE lim_prueba IDENTIFIED BY pwdUsrTmp
DEFAULT TABLESPACE practica_datos;

```

Conceder privilegios

```
GRANT CONNECT, CREATE SESSION, CREATE TABLE TO usrtmp;
```

Probar perfil

```
SQL> SELECT TO_CHAR(sysdate,'DD-MM-YYYY HH24:MI:SS') AS TIEMPO FROM DUAL;
TIEMPO
-----
27-03-2018 12:38:22

SQL> SELECT TO_CHAR(sysdate,'DD-MM-YYYY HH24:MI:SS') AS TIEMPO FROM DUAL;
SELECT TO_CHAR(sysdate,'DD-MM-YYYY HH24:MI:SS') AS TIEMPO FROM DUAL
*
ERROR at line 1:
ORA-02396: exceeded maximum idle time, please connect again

SQL> █
```

Figura 1. Tiempo excedido de inactividad

Si no tuviera efecto la configuración del perfil, es decir, que a pesar de haber transcurrido el tiempo el usuario sigue ejecutando las sentencias y no genera el error. Ejecute la siguiente instrucción desde la cuenta de SYS.

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

Y nuevamente vuelva a probar.

```
SQL> conn usrtmp/intento1
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL> conn usrtmp/intento2
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn usrtmp/intento3
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn usrtmp/intento4
ERROR:
ORA-28000: the account is locked

SQL> █
```

Figura 2. Intento de conexión con password incorrecto.

Nota: Para desbloquear la cuenta de un usuario se debe hacer con la cuenta de SYS o con la cuenta de un usuario con esos privilegios.

```
ALTER USER usrtmp ACCOUNT UNLOCK;
```

Ejercicio

Modificar el perfil lim_prueba restringiendo a 3 sesiones por usuario y compruebe que funciona correctamente

```
ALTER PROFILE lim_prueba LIMIT SESSIONS_PER_USER 3;
```

Roles.

Un rol es una forma de agrupar permisos (o privilegios) para asignarlos luego a los usuarios u a otro rol.

Cada usuario puede tener varios roles.

Sintaxis

```
CREATE ROLE <nombre_rol> [IDENTIFIED BY Contraseña];
```

Ejemplos:

```
CREATE ROLE programador;
```

```
CREATE ROLE abd; -- Administrador de base de datos
```

Conceder Privilegios

```
GRANT CREATE SESSION TO programador;  
GRANT SELECT, INSERT on JOEL.JUGADOR TO programador;
```

Crear otro Rol

```
CREATE ROLE otro_role;
```

Conceder los privilegios de programador a otro_rol;

```
GRANT programador TO otro_role;
```

Usuarios.

Sintaxis

```
CREATE USER nombre
IDENTIFIED BY contraseña
[DEFAULT TABLESPACE nombreTableSpace]
[TEMPORARY TABLESPACE nombreTemp]
[QUOTA INT {K|M} | UNLIMITED ON nombreTableSpace]
[PROFILE perfil]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}]
;
```

| | |
|----------------------|---|
| CREATE USER | Nos permite especificar el nombre del usuario |
| IDENTIFIED BY | Nos permite especificar su contraseña |
| DEFAULT TABLESPACE | Definimos el tablespace por defecto. Si no se especifica se asigna USERS |
| TEMPORARY TABLESPACE | Definimos el tablespace temporal. Si no especificamos ninguno se asigna TEMP |
| QUOTA | Espacio que el usuario podrá utilizar en el sistema. Si no se especifica el espacio por defecto es 0 con lo cual el usuario no podrá crear nada |
| PROFILE | Permite especificar el perfil por defecto. Si no se especifica se asigna DEFAULT. Es utilizado para controlar el acceso a los recursos, por ejemplo, el número de sesiones concurrentes, uso de CPU, etc. |
| PASSWORD EXPIRE | Especifica que la contraseña asignada al usuario expirará, de esta forma, el propio usuario o el DBA deberá asignar una nueva antes de acceder al sistema |
| ACCOUNT LOCK/UNLOCK | Podemos decidir si el usuario tendrá la cuenta bloqueada o no de forma inicial |

Ejemplos:

```
CREATE USER rosalba
IDENTIFIED BY pwdDeRosalba
DEFAULT TABLESPACE practica_datos
TEMPORARY TABLESPACE tmp
QUOTA 50M ON practica_datos
PASSWORD EXPIRE
ACCOUNT UNLOCK;
```

Modificación de Usuarios.

Una vez hemos creado un usuario podemos hacer ciertas modificaciones. La sintaxis es muy parecida a la vista anteriormente en la creación de usuarios.

Sintaxis

```
ALTER USER NombreUsuario  
IDENTIFIED BY contraseña  
[DEFAULT TABLESPACE nombreTableSpace]  
[TEMPORARY TABLESPACE nombreTemp]  
[QUOTA INT {K|M} UNLIMITED ON nombreTableSpace]  
[PASSWORD EXPIRE]  
[ACCOUNT {LOCK | UNLOCK}]  
[PROFILE perfil];
```

Ejemplo:

```
ALTER USER rosalba  
QUOTA UNLIMITED ON practica_datos;
```

Borrar usuarios

Podemos realizar una eliminación de usuarios con el comando DROP USER. Una vez que se borra un usuario, si se vuelve a crear otro usuario con el mismo nombre no se podrían heredar los objetos del anterior usuario. La razón por la que esto ocurriría es que Oracle asigna automáticamente un ID interno a cada usuario con el que lo identifica de forma única y aunque los usuarios tengan nombre idéntico, su ID siempre será distinto.

Sintaxis

```
DROP USER nombreUsuario [CASCADE];
```

Nota: CASCADE nos indica que vamos a borrar también **TODOS** sus objetos antes de proceder a la eliminación del usuario evitando posibles errores. Si el usuario posee algún objeto solo podremos eliminarlo con CASCADE.

Privilegios

Privilegios

-sobre objetos

- DELETE
- EXECUTE
- FLUSH
- INSERT
- INDEX
- LOAD
- REFRESH
- REFERENCES
- SELECT
- UNLOAD
- UPDATE

-del Sistema

- ADMIN
- ALTER ANY CACHE GROUP
- ALTER ANY INDEX
- ALTER ANY MATERIALIZED VIEW
- ALTER ANY PROCEDURE
- ALTER ANY SEQUENCE
- ALTER ANY TABLE
- ALTER ANY VIEW
- CACHE MANAGER
- CREATE ANY CACHE GROUP
- CREATE ANY INDEX
- CREATE ANY MATERIALIZED VIEW
- CREATE ANY PROCEDURE
- CREATE ANY SEQUENCE
- CREATE ANY SYNONYM
- CREATE ANY TABLE
- CREATE ANY VIEW
- CREATE CACHE GROUP
- CREATE MATERIALIZED VIEW
- CREATE PROCEDURE
- CREATE PUBLIC SYNONYM
- CREATE SEQUENCE
- CREATE SESSION
- CREATE SYNONYM
- CREATE TABLE
- CREATE VIEW
- DELETE ANY TABLE
- DROP ANY CACHE GROUP
- DROP ANY INDEX
- DROP ANY MATERIALIZED VIEW
- DROP ANY TABLE
- DROP ANY VIEW
- DROP ANY PROCEDURE
- DROP ANY SEQUENCE
- DROP ANY SYNONYM
- DROP PUBLIC SYNONYM
- EXECUTE ANY PROCEDURE
- FLUSH ANY CACHE GROUP
- INSERT ANY TABLE
- LOAD ANY CACHE GROUP
- REFRESH ANY CACHE GROUP
- SELECT ANY SEQUENCE
- SELECT ANY TABLE
- UNLOAD ANY CACHE GROUP
- UPDATE ANY TABLE
- XLA

a) Privilegios sobre objetos

Sintaxis

```
GRANT privilegio1 [[,privilegio2, ...] | ALL]
[(columna1[,columna2,...])]
[ON usuario[.objeto] | ANY TABLE]
TO {nombreUsuario | rol | PUBLIC}
[WITH GRANT OPTION];
```

- **ON:** Objeto sobre el que se aplica los privilegios
- **TO:** Usuario al que se concede la lista de privilegios
- **ALL:** Permite asignar todos los permisos
- **PUBLIC:** Asigna el privilegio o privilegios a todos los usuarios del sistema (también a los futuros)
- **WITH GRANT OPTION:** Permite que el usuario que lo reciba pueda conceder permisos a otros usuarios

Algunos de los privilegios mas usados son:

| Permiso | TABLA | VISTA | SECUENCIA | PROCEDIMIENTO |
|---------|-------|-------|-----------|---------------|
| ALTER | X | | X | |
| UPDATE | X | | | |
| DELETE | X | X | | |
| EXECUTE | | | | X |
| INSERT | X | X | | |
| SELECT | X | X | X | |

Nota: En la tabla se especifican algunos de los privilegios que podemos conceder y sobre qué objetos se utilizan.

Ejemplos:

```
GRANT SELECT ON JOEL.JUGADOR TO rosalba;
GRANT UPDATE ANY TABLE TO rosalba;
GRANT SELECT, INSERT, UPDATE ON JOEL.ESTADIO TO public;
GRANT INSERT(jugador_id,nombre) ON JOEL.JUGADOR TO rosalba;
```

b) Privilegios sobre sistema

Podemos también asignar ciertos permisos sobre el sistema a los usuarios que tengamos creados en nuestra base de datos. Los privilegios del sistema no se asignan sobre objetos concretos sino que especifican qué acciones se podrán realizar sobre el sistema gestor de base de datos.

Sintaxis

```
GRANT permiso1[,permiso2,...]  
TO nombreUsuario[,nombreUsuario2,...] | nombreRol;
```

Dónde permiso puede ser alguno de los siguientes:

- **create**
 - **session:** Permite conectarse a la base de datos
 - **table:** Permite crear tablas
 - **sequence:** Permite crear secuencias
 - **view:** Permite crear vistas
 - **trigger:** Permite crear disparadores
 - **procedure:** Permite crear procedimientos
 - **profile:** Permite crear perfiles
 - **synonym:** Permite crear sinónimos
- **execute any procedure:** Permite ejecutar cualquier procedimiento
- **create**
 - **user:** Permite crear usuarios. **WITH ADMIN OPTIONS** permite que el nuevo usuario tenga permisos administrativos, por ejemplo, para crear nuevos usuarios.
 - **role:** Permite crear roles
- **drop**
 - **table:** Permite eliminar tables
 - **sequence:** Permite eliminar secuencias
 - **view:** Permite eliminar vistas
 - **trigger:** Permite eliminar disparadores
 - **procedure:** Permite eliminar procedimientos
 - **profile:** Permite eliminar perfiles
 - **synonym:** Permite eliminar sinónimos
 - **user:** Permite eliminar usuarios
 - **role:** Permite eliminar roles
 - **session:** Permite eliminar sesiones
- **grant**
 - **privilege:** Permite asignar privilegios
 - **role:** Permite asignar roles

Ejemplo:

```
GRANT CREATE SESSION, CREATE TABLE TO ingrid;  
GRANT CREATE PROCEDURE, EXECUTE ANY PROCEDURE TO ingrid, rosalba;  
GRANT CREATE USER TO cardoso WITH ADMIN OPTION;
```

```
GRANT DBA TO ingrid; -- Concede privilegios (Administrador de la base de datos)
```

Quitar privilegios

a) Quitar privilegios sobre objetos

```
REVOKE permiso1[,permiso2,...] | ALL [PRIVILEGES]
ON [usuario.]objeto
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];
```

Ejemplo:

```
REVOKE INSERT on empleados FROM juanito;
```

b) Quitar privilegios sobre sistema

```
REVOKE permiso1[,permiso2,...] | ALL [PRIVILEGES]
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];
```

Ejemplo:

```
REVOKE ALL PRIVILEGES FROM laura;
REVOKE CREATE VIEW FROM rosa;
```

Autenticación de usuarios desde el Sistema operativo (Autenticación externa)

Oracle permite usuarios creados desde la base de datos y usuarios del sistema operativo, Los usuarios que hemos creado en los ejercicios anteriores todos han sido en la base de datos. En este apartado vamos a enlazar usuarios del sistema operativo con la base de datos de oracle.

Cuando un usuario conecta con la base de datos se verifica que el nombre de usuario es el mismo que el nombre de usuario del sistema operativo para permitir la validación.

No se almacenan las cuentas en la base de datos de ninguna forma. Estas cuentas están siempre referidas con **ops\$**. Sin embargo, a partir de la versión 10g se puede configurar **OS_AUTHENT_PREFIX** en el spfile

Paso 1. Crear usuarios en el sistema operativo (linux), con privilegios de root

```
# adduser -s /bin/bash rutilio
# passwd rutilio
```

Restringir horario de conexión

```
vi /etc/security/time.conf
```

agregar esta instrucción:

```
#--- Permitir acceso a rutilio todos los días 08:00 de la mañana a 17 hrs  
*;*;rutilio;A!0800-1700
```

Editar el siguiente archivo

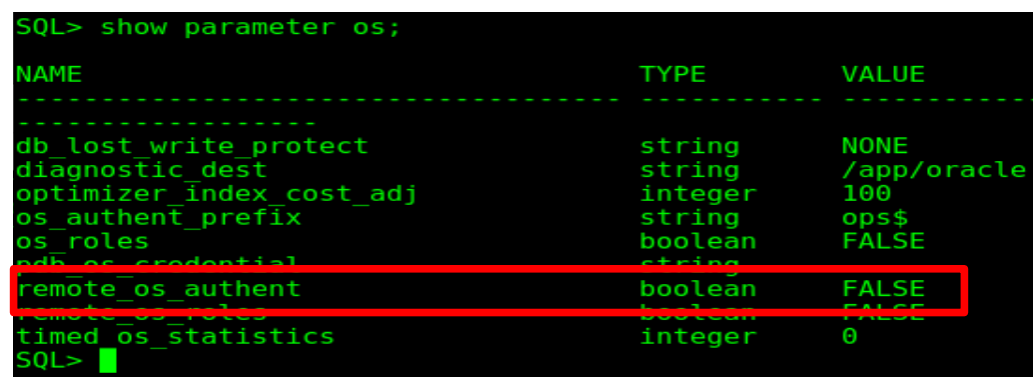
```
vi /etc/pam.d/system-auth
```

```
.  
.   
.   
  
#-----  
#--- Agregado por cardoso  
account    required    pam_time.so    #--- Agregar solo esta línea el resto ya aparece  
#-----  
account    required    pam_unix.so  
account    sufficient  pam_localuser.so  
account    sufficient  pam_succeed_if.so uid < 1000 quiet  
account    required    pam_permit.so  
.   
.   
. 
```

Paso 2. Crear el usuario en la base de datos, para ello es necesario revisar algunas configuraciones de la base de datos para evitar posibles errores:

a) Revisar el parámetro `remote_os_authent`

```
SQL> show parameter os;
```



| NAME | TYPE | VALUE |
|--------------------------|---------|-------------|
| db_lost_write_protect | string | NONE |
| diagnostic_dest | string | /app/oracle |
| optimizer_index_cost_adj | integer | 100 |
| os_authent_prefix | string | ops\$ |
| os_roles | boolean | FALSE |
| os_roles | string | |
| remote_os_authent | boolean | FALSE |
| remote_os_roles | boolean | FALSE |
| timed_os_statistics | integer | 0 |

Figura 3. Valor de `remote_os_authent`

En la figura 3. Se aprecia el valor de remote_os_authent a FALSE y es necesario cambiarlo a TRUE, para ello se debe ejecutar la siguiente sentencia:

```
SQL> ALTER SYSTEM set remote_os_authent=TRUE SID='*' SCOPE=SPFILE;
```

SID='*' para indicar esa operación afecte a todas instancias, se puede indicar a una sola por ejemplo **practica**, SCOPE permite indicar el ámbito; con el valor SPFILE cambia el valor en el spfile, por lo que hasta que no se reinicie la instancia de base de datos no tendrán efecto los cambios

Mediante el **scope** podemos indicar las siguientes tres opciones:

- **memory**: Cambia el valor solo a la instancia que se esta ejecutando, en el caso que se reinicie los cambios no tendrán efecto.
- **spfile**: Cambia el valor en el spfile, por lo que hasta que no se reinicie la instancia de base de datos no tendrán efecto los cambios
- **both**: Realiza el cambio tanto en la instancia que esta corriendo como en el spfile, haciendo permanentes los cambios.

Si al ejecutar la sentencia anterior marca error como el siguiente

ERROR at line 1:

ORA-32001: write to SPFILE requested but no SPFILE is in use

es necesario ejecutar esta instrucción:

```
SQL>show parameter pfile
NAME      TYPE        VALUE
-----
spfile     string
```

Si en la columna VALUE está en blanco, entonces ejecuta:

```
SQL>CREATE spfile FROM pfile;
File created.
SQL>shutdown immediate
SQL>startup
```

Volver a ejecutar las sentencias

```
SQL> ALTER SYSTEM set remote_os_authent=TRUE SID='*' SCOPE=SPFILE;
SQL> shutdown immediate
SQL> startup
```

Vea la figura 4.

```

SQL> ALTER SYSTEM SET remote_os_authent=TRUE SID='*' SCOPE=SPFILE;
ALTER SYSTEM SET remote_os_authent=TRUE SID='*' SCOPE=SPFILE
*
ERROR at line 1:
ORA-32001: write to SPFILE requested but no SPFILE is in use

SQL> show parameter pfile;

NAME                                TYPE        VALUE
-----
pfile                                string
SQL> CREATE SPFILE FROM pfile;

File created.

SQL> ALTER SYSTEM SET remote_os_authent=TRUE SID='*' SCOPE=SPFILE;
ALTER SYSTEM SET remote_os_authent=TRUE SID='*' SCOPE=SPFILE
*
ERROR at line 1:
ORA-32001: write to SPFILE requested but no SPFILE is in use

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area  423624704 bytes
Fixed Size                  8621472 bytes
Variable Size             335544928 bytes
Database Buffers           71303168 bytes
Redo Buffers                8155136 bytes
Database mounted.
Database opened.
SQL> ALTER SYSTEM SET remote_os_authent=TRUE SID='*' SCOPE=SPFILE;

System altered.

SQL> █

```

Figura 4. Modificación del parámetro remote_os_authent

```

SQL> show parameter os;

NAME                                TYPE        VALUE
-----
db_write_protect                   string      NONE
diagnostic_dest                     string      /app/oracle
optimizer_index_cost_adj           integer     100
os_authent_prefix                   string      ops$
os_roles                           boolean     FALSE
pdb_os_credential                   string
remote_os_authent                   boolean     TRUE
remote_os_roles                     boolean     FALSE
timed_os_statistics                 integer     0
SQL> █

```

Figura 5. remote_os_authent modificado

b) Crear el usuario

```

SQL> CREATE USER ops$rutilio IDENTIFIED EXTERNALLY;
User Created.
SQL>

```

Nota: Es importante que la cuenta del usuario comience con el valor de os_authent_prefix, como se muestra en la figura 5 con el valor de **ops\$**

c) Conceder privilegios

```

SQL> GRANT CREATE SESSION TO ops$rutilio;

```

d) Probar la conexión. Abrir otra consola e iniciar sesión con la cuenta de rutilio

```
$ su rutilio
password:xxxxx

$ export ORACLE_HOME=/app/oracle/product/12.2.0/dbhome_1
$ export ORACLE_BASE=/app/oracle
$ export ORACLE_SID=practica
$cd $ORACLE_HOME/bin
$./sqlplus /
```

Si al ejecutar muestra el siguiente error:

```
SQL*Plus: Release 12.2.0.1.0 Production on Wed Mar 28 12:18:51 2018
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
ERROR:
ORA-12547: TNS:lost contact
```

entonces abra una terminal con cuenta de root y escriba lo siguiente:

```
[rutilio@localhost bin]$ ls -lrt oracle
-rwxr-x--x. 1 oracle oinstall 407988856 mar 13 18:54 oracle
[rutilio@localhost bin]$ su
Contraseña:
[root@localhost bin]# cd /app/oracle/product/12.2.0/dbhome_1/bin
[root@localhost bin]# chmod 6751 oracle
[root@localhost bin]# ls -lrt oracle
-rwsr-s--x. 1 oracle oinstall 407988856 mar 13 18:54 oracle
```

Volver a la ventana de rutilio y ejecutar nuevamente ./sqlplus /

```
[rutilio@localhost bin]$ ./sqlplus /
SQL*Plus: Release 12.2.0.1.0 Production on Wed Mar 28 12:38:00 2018
Copyright (c) 1982, 2016, Oracle. All rights reserved.
ERROR:
ORA-00942: table or view does not exist
Error accessing PRODUCT_USER_PROFILE
Warning: Product user profile information not loaded!
You may need to run PUPBLD.SQL as SYSTEM

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL>
```

Nota: El resultado anterior muestra un ERROR **ORA-00942: table or view does not exist**, esto se resuelve de la siguiente manera:

Conectarse con cuenta de SYS en la misma terminal

```
SQL>CONN SYSTEM  
password: XXXXX
```

Ejecutar la siguiente instrucción

```
SQL>@?/sqlplus/admin/pupbld.sql  
.  
.  
Synonym created.  
Synonym dropped.  
Synonym created.  
Session altered.  
  
SQL> exit
```

Nuevamente ejecutar ./sqlplus /

```
[rutilio@localhost bin]$ ./sqlplus /  
  
SQL*Plus: Release 12.2.0.1.0 Production on Wed Mar 28 12:42:18 2018  
Copyright (c) 1982, 2016, Oracle. All rights reserved.  
  
Last Successful login time: Wed Mar 28 2018 12:38:00 -07:00  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
```

Ejercicios:

1. Ejecuta la sentencia **SELECT * FROM dba_sys_privs**, comente en el grupo que información contiene. Además pruebe a filtrar los datos por la columna **GRANTEE**
2. Compara el resultado del ejercicio 1 con la siguiente **SELECT * FROM user_sys_privs**
3. Ejecute la misma sentencia del ejercicio 2 pero con otro usuario por ejemplo en la sesión de **alondra**, compare resultados.
4. Ejecute estas dos instrucciones **SELECT * FROM ALL_USERS** y **SELECT * FROM DBA_USERS**, compare resultados
5. Con la cuenta de **SYS**, ejecute la siguiente instrucción:
CREATE TABLE examen (examen_id NUMBER GENERATED ALWAYS AS IDENTITY, estudiante VARCHAR2 (100) NOT NULL, calificacion NUMBER (5,2), CONSTRAINT pkexamen PRIMARY KEY (examen_id));
- 6). Crea una cuenta con privilegios de inserción sobre la tabla examen e inserte al menos 3 filas
- 7). Crea una cuenta con privilegios de actualización sobre la tabla examen y actualiza una fila
- 8). Crea una cuenta de solo lectura sobre la tabla examen e intente borrar una fila
- 9). Crea una cuenta con privilegios de eliminación sobre la tabla examen intente insertar y actualizar
- 10). Crea una cuenta con todos los privilegios sobre la tabla examen y compruebe que los permisos sean efectivos.