

ACTIVIDAD

SQL: DCL y extensión procedimental, casos prácticos I

Desarrollo de Aplicaciones
Multiplataforma, Desarrollo de
Aplicaciones Web.

Módulo 02: Bases de datos



Actividad

SQL: DCL y extensión procedimental, casos prácticos I.

Objetivos

- Creación usuarios y asignación de permisos
- Creación de roles
- Creación de perfiles.

¿Cómo lo hago?

1. Rellena los datos que se piden en la tabla “Antes de empezar”.
2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc.
3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar.
4. Recuerda entregar la actividad en formato PDF a no ser que el profesor o profesora indique lo contrario.
5. Añade al documento el código de la aplicación y capturas de pantalla en las que se observe su funcionamiento.
6. Si la consulta pide una información determinada, por ejemplo una columna, vuestra consulta debe devolver solo el valor de esa columna.
7. Es importante entregar la tarea dentro del plazo establecido, aunque sea incompleta.
8. Se debe realizar con el gestor de Bases de datos Oracle.
9. Puedes adjuntar el fichero .sql
10. Comprime en un archivo zip, rar el código y el pdf.
11. Recuerda nombrar el archivo siguiendo estas indicaciones:
12. Ciclo_Módulo o crédito_Tema_ACT_número actividad_Nombre y apellido
 - a. Ejemplo: AF_M01_T01_ACT_01_Maria Garcia

Antes de empezar...

Nombre	Martí
Apellidos	Curto Vendrell
Módulo/Crédito	M02
UF (solo ciclos LOE)	UF3
Título de la actividad	SQL: DCL y extensión procedimental, casos prácticos I

1. Academia de Idiomas World

La academia de idiomas World de nuestra localidad nos ha pedido una pequeña base de datos para controlar las inscripciones a los cursillos de idiomas del próximo verano. Para simplificar el problema supondremos que un alumno solo puede apuntarse a un solo curso de idiomas.

De los cursos queremos guardar la siguiente información:

- Código del curso
- Nombre
- Horario
- Fecha inicio
- Fecha final
- Precio
- Profesor que lo imparte.

De los alumnos queremos guardar.

- Nombre
- Número de alumno (Clave primaria)
- Curso al que se inscribe.
- Fecha de inscripción

Para implementar la tarea debemos realizar las siguientes operaciones:

1. Crear un tablespace de 400MB llamado academia donde se almacenarán todos los datos.

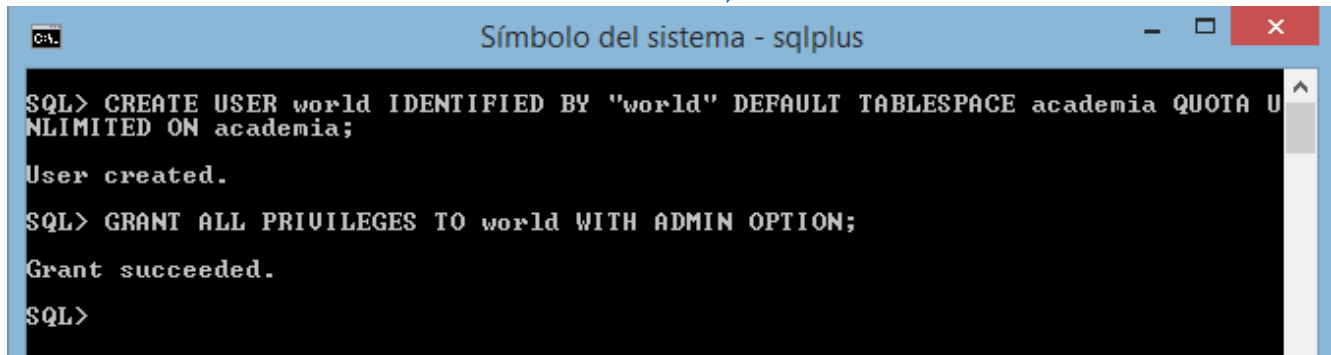
```
CREATE TABLESPACE academia DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\academia.dbf'
SIZE 400M;
```

```
C:\Users\User>sqlplus
SQL*Plus: Release 11.2.0.2.0 Production on Tue Nov 27 14:16:44 2020
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Enter user-name: system
Enter password:
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL> CREATE TABLESPACE academia DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\acad
emia.dbf' SIZE 400M;
Tablespace created.
SQL>
```

2. Crea un usuario llamado "world" que tenga todos los privilegios en el sistema Oracle. Comprobar que realmente tiene asignados estos permisos.

Primero creamos al usuario y le damos los permisos:

```
CREATE USER world IDENTIFIED BY "world" DEFAULT TABLESPACE academia QUOTA UNLIMITED ON academia;
GRANT ALL PRIVILEGES TO world WITH ADMIN OPTION;
```



```
Símbolo del sistema - sqlplus

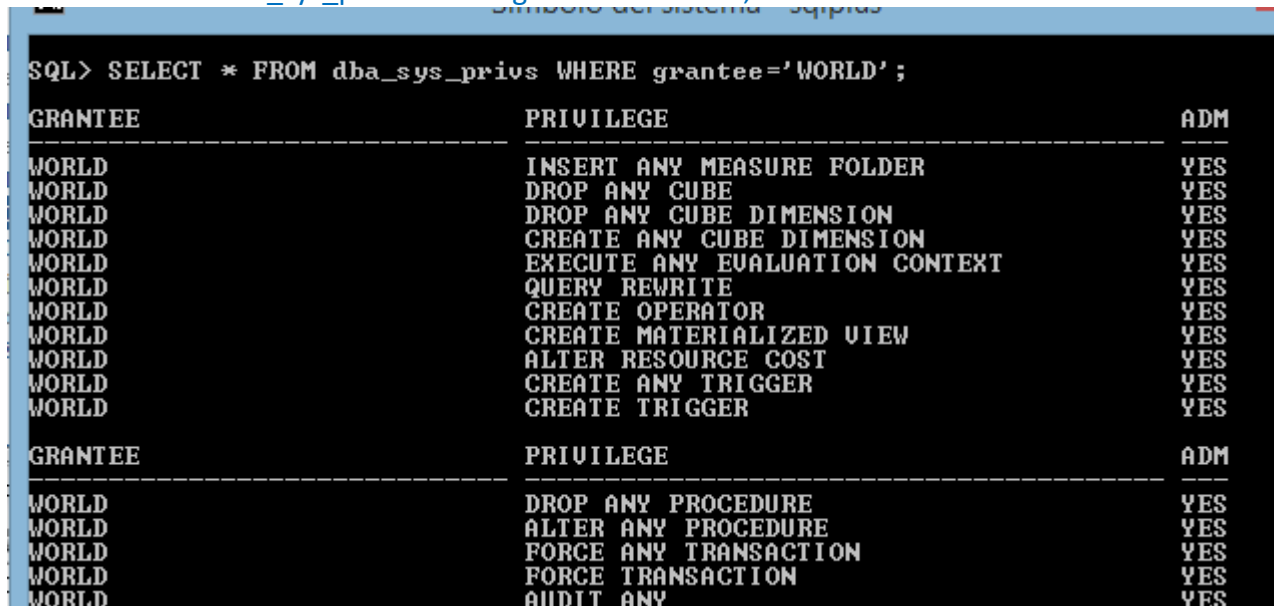
SQL> CREATE USER world IDENTIFIED BY "world" DEFAULT TABLESPACE academia QUOTA UNLIMITED ON academia;
User created.

SQL> GRANT ALL PRIVILEGES TO world WITH ADMIN OPTION;
Grant succeeded.

SQL>
```

Comprobamos que tiene todos los permisos:

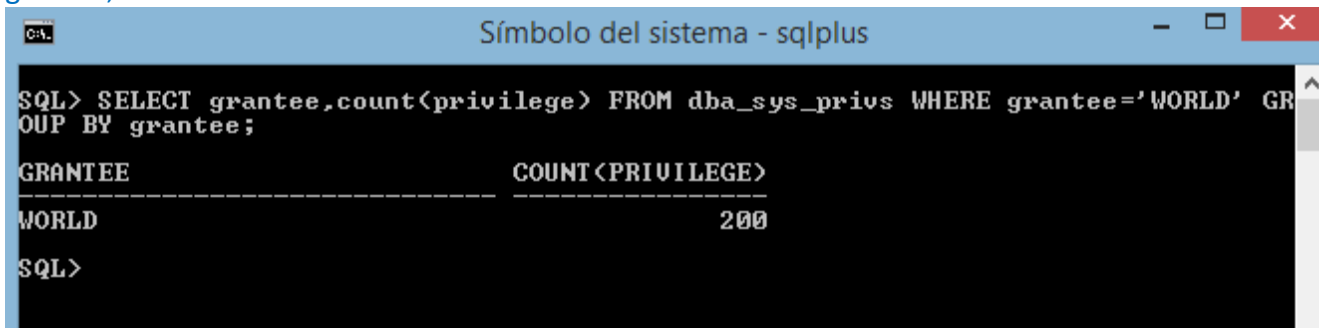
```
SELECT * FROM dba_sys_privs WHERE grantee='WORLD';
```



GRANTEE	PRIVILEGE	ADM
WORLD	INSERT ANY MEASURE FOLDER	YES
WORLD	DROP ANY CUBE	YES
WORLD	DROP ANY CUBE DIMENSION	YES
WORLD	CREATE ANY CUBE DIMENSION	YES
WORLD	EXECUTE ANY EVALUATION CONTEXT	YES
WORLD	QUERY REWRITE	YES
WORLD	CREATE OPERATOR	YES
WORLD	CREATE MATERIALIZED VIEW	YES
WORLD	ALTER RESOURCE COST	YES
WORLD	CREATE ANY TRIGGER	YES
WORLD	CREATE TRIGGER	YES
GRANTEE	PRIVILEGE	ADM
WORLD	DROP ANY PROCEDURE	YES
WORLD	ALTER ANY PROCEDURE	YES
WORLD	FORCE ANY TRANSACTION	YES
WORLD	FORCE TRANSACTION	YES
WORLD	AUDIT ANY	YES

Oracle tiene 200 permisos, vamos a ver que los tenga todos:

```
SELECT grantee,count(privilege) FROM dba_sys_privs WHERE grantee='WORLD' GROUP BY grantee;
```



Símbolo del sistema - sqlplus

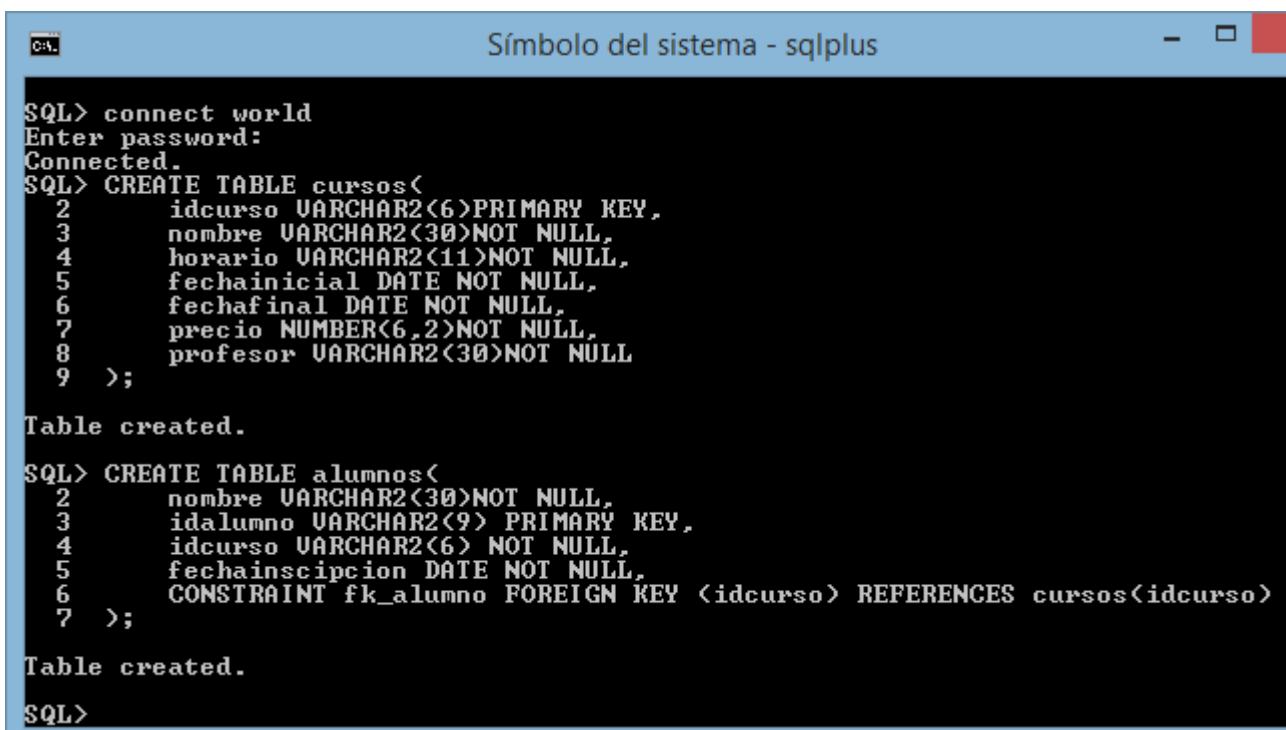
```
SQL> SELECT grantee,count(privilege) FROM dba_sys_privs WHERE grantee='WORLD' GR  
OUP BY grantee;
```

GRANTEE	COUNT(PRIIVILEGE)
WORLD	200

```
SQL>
```

- Utilizando el usuario de nombre "world" crea la tabla cursos y la tabla alumnos. Utilizar el tipo de campo y la longitud que creáis más adecuados para cada uno de los campos. Introduce datos en las tablas.

Nos conectamos con "world" y creamos las tablas:



Símbolo del sistema - sqlplus

```
SQL> connect world
Enter password:
Connected.
SQL> CREATE TABLE cursos(
2     idcurso VARCHAR2(6)PRIMARY KEY,
3     nombre VARCHAR2(30)NOT NULL,
4     horario VARCHAR2(11)NOT NULL,
5     fechainicial DATE NOT NULL,
6     fechafinal DATE NOT NULL,
7     precio NUMBER(6,2)NOT NULL,
8     profesor VARCHAR2(30)NOT NULL
9 );

Table created.
SQL> CREATE TABLE alumnos(
2     nombre VARCHAR2(30)NOT NULL,
3     idalumno VARCHAR2(9) PRIMARY KEY,
4     idcurso VARCHAR2(6) NOT NULL,
5     fechainscpcion DATE NOT NULL,
6     CONSTRAINT fk_alumno FOREIGN KEY (idcurso) REFERENCES cursos(idcurso)
7 );

Table created.
SQL>
```

Las rellenamos:

```
Símbolo del sistema - sqlplus

SQL> INSERT INTO cursos VALUES (<'MAT101','matematicas introducción','18:00-19:00',
', '21-01-08', date '21-06-20', 500.50, 'Jorge Perez')>;

1 row created.

SQL> INSERT INTO cursos VALUES (<'ENG101','ingles introducció','19:00-20:00','21-
01-08', '21-06-20', 650.50, 'Robet Smith')>;

1 row created.

SQL>
SQL> INSERT INTO alumnos VALUES(<'Miguel Perez', 'MPC01','MAT101','20-12-20')>;

1 row created.

SQL> INSERT INTO alumnos VALUES(<'Javier Alto Delgado', 'JAD01','MAT101','20-12-1
5')>;

1 row created.

SQL> INSERT INTO alumnos VALUES(<'Juan Bajo Delgado', 'JBD01','ENG101','20-12-15'
>);

1 row created.

SQL> INSERT INTO alumnos VALUES(<'Javier Alto Delgado', 'JAD02','ENG101','21-1-7'
>);
```

NOTA: a partir de esta parte se han creado 2 sinónimos públicos para facilitar los ejercicios con el siguiente código:

```
CREATE PUBLIC SYNONYM cursos FOR world.cursos;
CREATE PUBLIC SYNONYM alumnos FOR world.alumnos;
```



```
Símbolo del sistema - sqlplus

SQL> CREATE PUBLIC SYNONYM cursos FOR world.cursos;

Synonym created.

SQL> CREATE PUBLIC SYNONYM alumnos FOR world.alumnos;

Synonym created.
```

- 
- 
4. Crear dos usuarios “secre1” y “secre2” con password “world1234”, que se encarguen de la gestión de la academia (añadir, modificar, borrar, consultar) en la tabla de cursos y en la tabla de alumnos. Comprueba que los privilegios se han asignado de forma correcta y que puede hacer las operaciones asignadas.

En primer lugar, creamos los 2 usuarios. Como no nos especifican la memoria que pueden ocupar u otros limites, solamente las crearemos y les diremos que usen el tablespace academia.

```
CREATE USER secre1 IDENTIFIED BY "world1234" DEFAULT TABLESPACE academia;  
CREATE USER secre2 IDENTIFIED BY "world1234" DEFAULT TABLESPACE academia;
```

```
SQL> CREATE USER secre1 IDENTIFIED BY "world1234" DEFAULT TABLESPACE academia;  
User created.  
SQL> CREATE USER secre2 IDENTIFIED BY "world1234" DEFAULT TABLESPACE academia;  
User created.  
SQL>
```

Les damos los privilegios:

```
GRANT CREATE SESSION TO secre1, secre2;  
GRANT SELECT,INSERT,UPDATE,DELETE ON cursos TO secre1, secre2;  
GRANT SELECT,INSERT,UPDATE,DELETE ON alumnos TO secre1,secre2;
```

```
SQL> GRANT CREATE SESSION to secre1, secre2;  
Grant succeeded.  
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON cursos to secre1, secre2;  
Grant succeeded.  
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON alumnos to secre1,secre2;  
Grant succeeded.  
SQL>
```


Y comprobamos que privilegios tienen, en primer lugar, veremos que pueden iniciar sesión y después los privilegios sobre las tablas:

`SELECT * FROM dba_sys_privs WHERE grantee IN('SECRE1','SECRE2');`

Símbolo del sistema - sqlplus

```
SQL> SELECT * FROM dba_sys_privs WHERE grantee IN('SECRE1','SECRE2');
```

GRANTEE	PRIVILEGE	ADM
SECRE1	CREATE SESSION	NO
SECRE2	CREATE SESSION	NO



SQL>

`SELECT substr(grantee,1,10),substr(table_name,1,10),substr(privilege,1,10)
FROM dba_tab_privs WHERE grantee IN('SECRE1','SECRE2');`

	SUBSTR(GRANTEE,1,10)	SUBSTR(TABLE_NAME,1,10)	SUBSTR(PRIVILEGE,1,10)
1	SECRE2	ALUMNOS	UPDATE
2	SECRE1	ALUMNOS	UPDATE
3	SECRE2	ALUMNOS	SELECT
4	SECRE1	ALUMNOS	SELECT
5	SECRE2	ALUMNOS	INSERT
6	SECRE1	ALUMNOS	INSERT
7	SECRE2	ALUMNOS	DELETE
8	SECRE1	ALUMNOS	DELETE
9	SECRE2	CURSOS	UPDATE
10	SECRE1	CURSOS	UPDATE
11	SECRE2	CURSOS	SELECT
12	SECRE1	CURSOS	SELECT
13	SECRE2	CURSOS	INSERT
14	SECRE1	CURSOS	INSERT
15	SECRE2	CURSOS	DELETE
16	SECRE1	CURSOS	DELETE

Para una segunda prueba nos conectaremos con secre1 e intentaremos hacer un SELECT en cada tabla.

`CONNECT secre1;
SELECT nombre, idcurso FROM world.alumnos;
SELECT idcurso, nombre FROM world.cursos;`



```
SQL> connect secre1
Enter password:
Connected.
SQL> SELECT  nombre, idcurso FROM world.alumnos;

NOMBRE                                IDCURS
-----                                -
Miguel Perez                          MAT101
Javier Alto Delgado                   MAT101
Juan Bajo Delgado                     ENG101
Javier Alto Delgado                   ENG101

SQL> SELECT idcurso, nombre FROM world.cursos;

IDCURS  NOMBRE
-----  -
MAT101  matematicas introducción
ENG101  ingles introducció

SQL>
```

Como ya hemos hecho la comprobación anterior asumimos que tenemos los permisos para las otras operaciones en las 2 tablas y con los 2 usuarios.

5. Se decide que el usuario “secre1” pueda crear nuevos usuarios, pero no podrá eliminar a ningún usuario. Comprobar que realmente tiene asignados estos permisos. Quitamos el permiso de borrar registros al usuario “secre2” sobre la tabla de cursos. Comprobación.

En este caso entendemos que solo tiene que crear los usuarios, no hace falta que inicien sesión ni puedan hacer nada más por el momento

`GRANT CREATE USER to secre1;`

Miramos que pueda crear usuarios:

`SELECT * FROM dba_sys_privs WHERE grantee='SECRE1';`

Todas las Filas Recuperadas: 2 en 0,0

	GRANTEE	PRIVILEGE	ADMIN_OPTION
1	SECRE1	CREATE USER	NO
2	SECRE1	CREATE SESSION	NO

`CONNECT secre1;`

`CREATE USER prueba IDENTIFIED BY "1234";`

`DROP USER prueba;`

```
SQL> GRANT CREATE USER to secre1;
Grant succeeded.

SQL> connect secre1
Enter password:
Connected.
SQL> CREATE USER prueba IDENTIFIED BY "1234";
User created.

SQL> DROP USER prueba;
DROP USER prueba
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

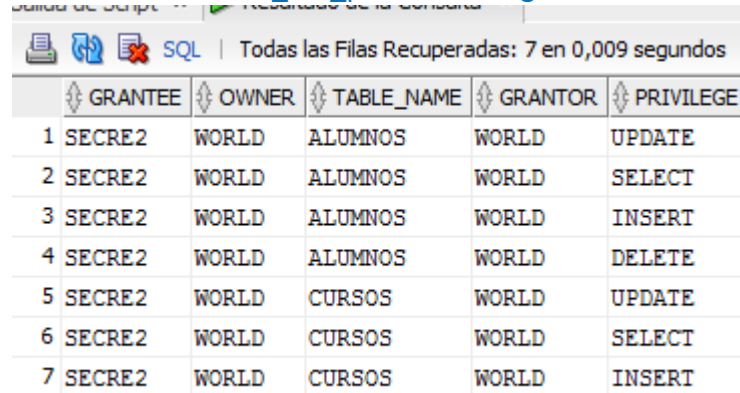
Ahora le quitaremos el permiso de registros a secre2 sobre la tabla cursos y lo comprobaremos:

`REVOKE DELETE ON cursos FROM secre2;`

```
SQL> REVOKE DELETE ON cursos FROM secre2;
Revoke succeeded.
```

Y lo comprobaremos

`SELECT * FROM dba_tab_privs WHERE grantee='SECRE2';`

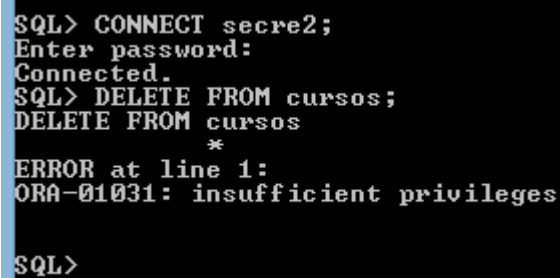


	GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE
1	SECRE2	WORLD	ALUMNOS	WORLD	UPDATE
2	SECRE2	WORLD	ALUMNOS	WORLD	SELECT
3	SECRE2	WORLD	ALUMNOS	WORLD	INSERT
4	SECRE2	WORLD	ALUMNOS	WORLD	DELETE
5	SECRE2	WORLD	CURSOS	WORLD	UPDATE
6	SECRE2	WORLD	CURSOS	WORLD	SELECT
7	SECRE2	WORLD	CURSOS	WORLD	INSERT

Vemos que en la tabla alumnos puede eliminar registros, pero en los cursos no. Si lo intentamos entrando en su sesión tampoco nos dejará:

`CONNECT secre2;`

`DELETE FROM cursos;`

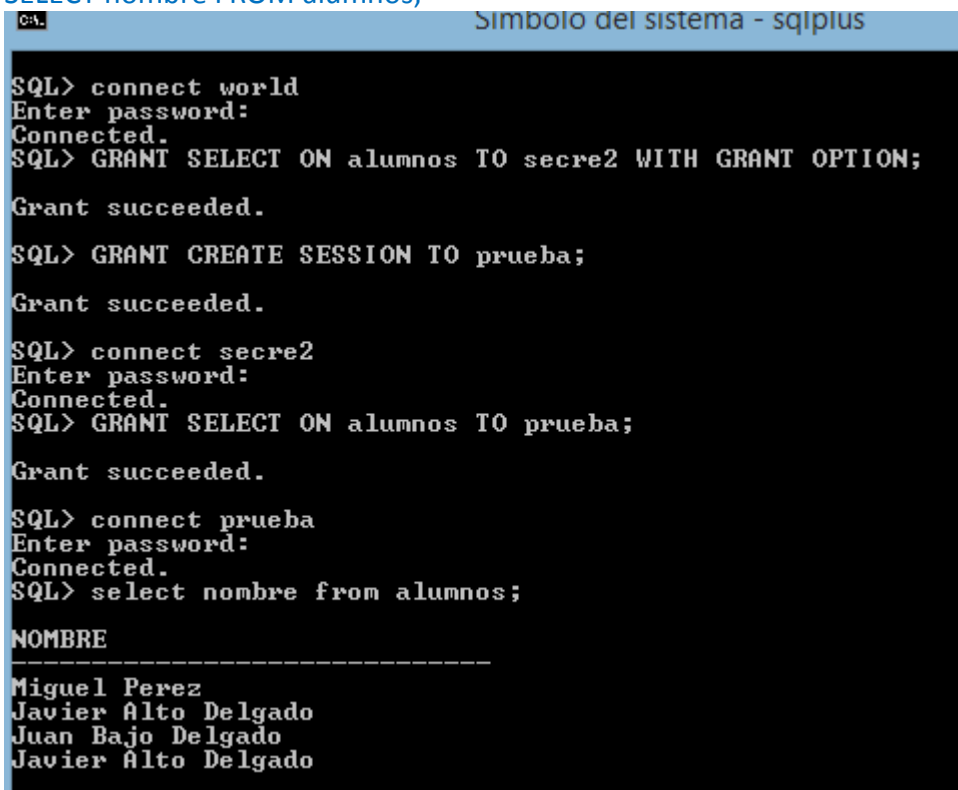


```
SQL> CONNECT secre2;
Enter password:
Connected.
SQL> DELETE FROM cursos;
DELETE FROM cursos
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

6. El usuario “world” concede al usuario “secre2”, la posibilidad de asignar el permiso de lectura (SELECT) de datos a otros usuarios sobre la tabla de alumnos. Comprobar que realmente tiene asignados estos permisos.

```
CONNECT world;  
GRANT SELECT ON alumnos TO secre2 WITH GRANT OPTION;  
GRANT CREATE SESSION TO prueba;  
CONNECT secre2;  
GRANT SELECT ON alumnos TO prueba;  
SELECT nombre FROM alumnos;
```



```
SQL> connect world  
Enter password:  
Connected.  
SQL> GRANT SELECT ON alumnos TO secre2 WITH GRANT OPTION;  
  
Grant succeeded.  
  
SQL> GRANT CREATE SESSION TO prueba;  
  
Grant succeeded.  
  
SQL> connect secre2  
Enter password:  
Connected.  
SQL> GRANT SELECT ON alumnos TO prueba;  
  
Grant succeeded.  
  
SQL> connect prueba  
Enter password:  
Connected.  
SQL> select nombre from alumnos;  
  
NOMBRE  
-----  
Miguel Perez  
Javier Alto Delgado  
Juan Bajo Delgado  
Javier Alto Delgado
```

Comprobaremos que solo tiene ese privilegio con el código:

```
SELECT substr(grantee,1,10) AS usuario,substr(table_name,1,10) AS tabla,substr(privilege,1,10) AS privi FROM dba_tab_privs WHERE grantee='PRUEBA';
```

```
SQL> SELECT substr(grantee,1,10) AS usuario,substr(table_name,1,10) AS tabla,substr(
privilege,1,10) AS privi FROM dba_tab_privs WHERE grantee='PRUEBA';
```

USUARIO	TABLA	PRIVI
PRUEBA	ALUMNOS	SELECT

```
SQL>
```

7. Crea un rol llamado rolprofe con las siguientes características: Puede iniciar sesión, leer la tabla de cursos y leer y modificar la tabla de alumnos (no puede ni borrar ni añadir).

```
CREATE ROLE rolprofe;
GRANT CREATE SESSION TO rolprofe;
GRANT SELECT ON cursos TO rolprofe;
GRANT SELECT, UPDATE ON alumnos TO rolprofe;
```

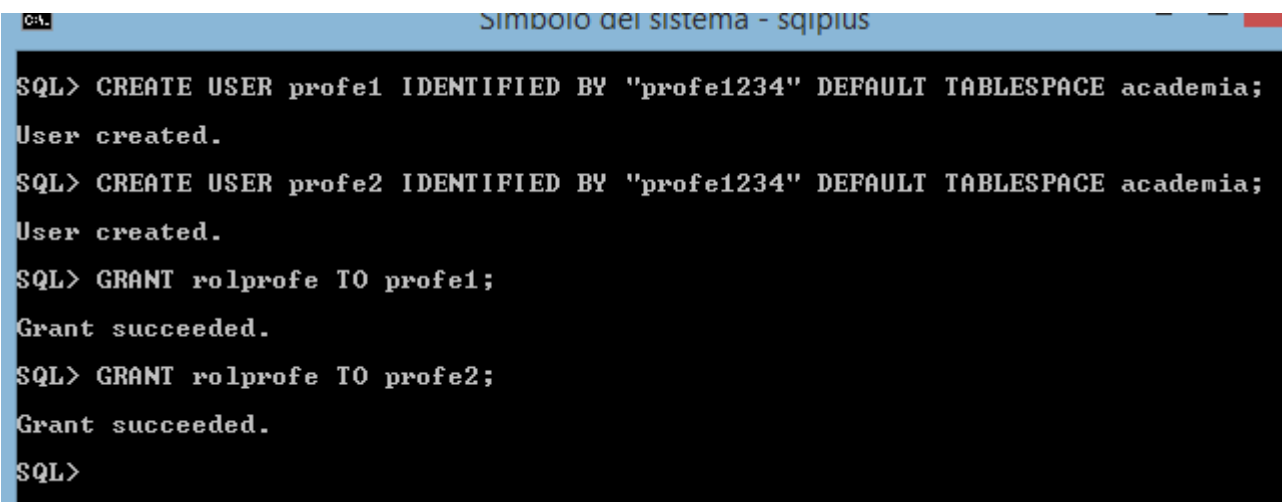
Símbolo del sistema - sqlp

```
SQL> CREATE ROLE rolprofe;
Role created.
SQL> GRANT CREATE SESSION TO rolprofe;
Grant succeeded.
SQL> GRANT SELECT ON cursos TO rolprofe;
Grant succeeded.
SQL> GRANT SELECT, UPDATE ON alumnos TO rolprofe;
Grant succeeded.
SQL>
```

8. Crea dos usuarios "profe1" y "profe2" con password "profe1234" y le asignas el rol anterior. Comprueba que tienen los permisos correspondientes.

Primero crearemos los usuarios y les daremos el rol desde el usuario "world":

```
CREATE USER profe1 IDENTIFIED BY "profe1234" DEFAULT TABLESPACE academia;
CREATE USER profe2 IDENTIFIED BY "profe1234" DEFAULT TABLESPACE academia;
GRANT rolprofe TO profe1;
GRANT rolprofe TO profe2;
```



```

SQL> CREATE USER profe1 IDENTIFIED BY "profe1234" DEFAULT TABLESPACE academia;
User created.
SQL> CREATE USER profe2 IDENTIFIED BY "profe1234" DEFAULT TABLESPACE academia;
User created.
SQL> GRANT rolprofe TO profe1;
Grant succeeded.
SQL> GRANT rolprofe TO profe2;
Grant succeeded.
SQL>

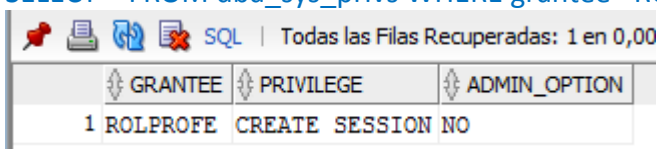
```

Ahora comprobaremos que tienen los privilegios que hemos dado;

Lo podemos hacer de 2 formas, mirando la información de los archivos de la base de datos o probando si somos capaces de conectarnos y hacer las acciones que se nos pide.

Primero probaremos mediante los archivos de la base de datos:

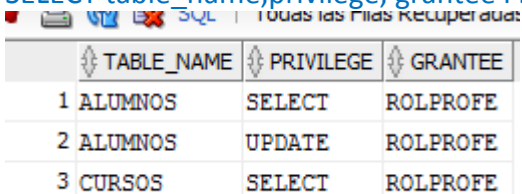
`SELECT * FROM dba_sys_privs WHERE grantee='ROLPROFE';`



Todas las Filas Recuperadas: 1 en 0,00

	GRANTEE	PRIVILEGE	ADMIN_OPTION
1	ROLPROFE	CREATE SESSION	NO

`SELECT table_name, privilege, grantee FROM dba_tab_privs WHERE grantee='ROLPROFE';`



Todas las Filas Recuperadas:

	TABLE_NAME	PRIVILEGE	GRANTEE
1	ALUMNOS	SELECT	ROLPROFE
2	ALUMNOS	UPDATE	ROLPROFE
3	CURSOS	SELECT	ROLPROFE

Ahora mediante el método de prueba/error a ver si nos deja ejecutar las acciones.

```

CONNECT profe1;
SELECT idcurso FROM cursos;
UPDATE alumnos SET idalumno='99999999X' WHERE idalumno='MPC01';
SELECT idalumno FROM alumnos;

```

```

SQL> CONNECT profe1;
Enter password:
Connected.
SQL> SELECT idcurso FROM cursos;

IDCURS
-----
ENG101
MAT101

SQL> UPDATE alumnos SET idalumno='99999999X' WHERE idalumno='MPC01';
1 row updated.

SQL> SELECT idalumno FROM alumnos;

IDALUMNO
-----
99999999X
JAD01
JAD02
JBD01
SQL>

```

Comprobamos que no podemos borrar ni añadir:

```

DELETE FROM alumnos WHERE idalumno='99999999X';
INSERT INTO alumnos VALUES('Javier Alto Delgado', 'JAD02','ENG101','21-1-7');

```

```

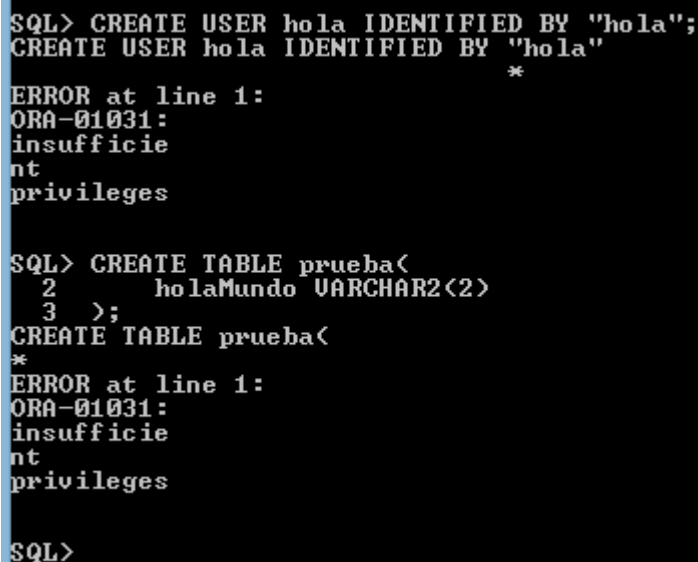
SQL> DELETE FROM alumnos WHERE idalumno='99999999X';
DELETE FROM alumnos WHERE idalumno='99999999X'
*
ERROR at line 1:
ORA-01031:
insufficie
nt
privileges

SQL> INSERT INTO alumnos VALUES('Javier Alto Delgado', 'JAD02','ENG101','21-1-7'
);
INSERT INTO alumnos VALUES('Javier Alto Delgado', 'JAD02','ENG101','21-1-7')
*
ERROR at line 1:
ORA-01031:
insufficie
nt
privileges

```


Intentaremos otro tipo de privilegios como el de crear usuarios o tablas:

```
CREATE USER hola IDENTIFIED BY "hola";  
CREATE TABLE prueba(  
    holaMundo VARCHAR2(2)  
);
```



```
SQL> CREATE USER hola IDENTIFIED BY "hola";  
CREATE USER hola IDENTIFIED BY "hola"  
*  
ERROR at line 1:  
ORA-01031:  
insufficie  
nt  
privileges  
  
SQL> CREATE TABLE prueba(  
    2      holaMundo VARCHAR2(2)  
    3  );  
CREATE TABLE prueba(  
*  
ERROR at line 1:  
ORA-01031:  
insufficie  
nt  
privileges  
  
SQL>
```

9. Crearemos un perfil para los profesores llamado perfilprofe que tenga un tiempo máximo de conexión de 1 hora, dos conexiones simultáneas y le obligue a cambiar la contraseña cada 30 días. Asigna este perfil al usuario profe1.

En primer lugar, activaremos los perfiles ya que por defecto están desactivados.



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



```
SQL> ALTER SYSTEM SET RESOURCE_LIMIT=TRUE;  
System altered.  
SQL>
```

Ahora crearemos el perfil:

```
CREATE PROFILE perfilprofe LIMIT  
    CONNECT_TIME 60  
    SESSIONS_PER_USER 2  
    PASSWORD_LIFE_TIME 30;
```

```

C:\>
SQL> CREATE PROFILE perfilprofe LIMIT
2     CONNECT_TIME 60
3     SESSIONS_PER_USER 2
4     PASSWORD_LIFE_TIME 30;

Profile created.

SQL>

```

Podemos comprobar los recursos que tiene este perfil mediante el código:


```
SELECT resource_name,substr(limit,1,10) FROM dba_profiles WHERE profile='PERFILPROFE';
```

	RESOURCE_NAME	LIMIT
1	COMPOSITE_LIMIT	DEFAULT
2	SESSIONS_PER_USER	2
3	CPU_PER_SESSION	DEFAULT
4	CPU_PER_CALL	DEFAULT
5	LOGICAL_READS_PER_SESSION	DEFAULT
6	LOGICAL_READS_PER_CALL	DEFAULT
7	IDLE_TIME	DEFAULT
8	CONNECT_TIME	60
9	PRIVATE_SGA	DEFAULT
10	FAILED_LOGIN_ATTEMPTS	DEFAULT
11	PASSWORD_LIFE_TIME	30
12	PASSWORD_REUSE_TIME	DEFAULT
13	PASSWORD_REUSE_MAX	DEFAULT
14	PASSWORD_VERIFY_FUNCTION	DEFAULT
15	PASSWORD_LOCK_TIME	DEFAULT
16	PASSWORD_GRACE_TIME	DEFAULT

Para activar los perfiles tendremos que usar la instrucción:

Ahora lo asignamos a profe1:

```
ALTER USER profe1 PROFILE perfilprofe;
```



```

SQL> ALTER USER profe1 PROFILE perfilprofe;

User altered.

SQL>

```

Vamos a comprobar que funciona, imaginamos que tenemos tiempo libre y probamos que al volver

pada una hora intentamos hacer un SELECT, vemos que se ha aplicado bien el perfil ya que nos tenemos que volver a conectar.

```
SQL> connect profel
Enter password:
Connected.
SQL> select * from alumnos
      2 ;
select * from alumnos
      *
ERROR at line 1:
ORA-02399: exceeded maximum connect time, you are being logged off

SQL>
```

Y si intentamos más conexiones a la base de datos nos indica un mensaje de error.

```
-- comprobamos
CONNECT profel;
SELECT idcurso FROM cursos;
UPDATE alumnos SET idalumno='99999999X' WHERE idalumno='MPC01';
SELECT idalumno FROM alumnos;
DELETE FROM alumnos WHERE idalumno='99999999X';
```

Salida de Script x

Tarea terminada en 4,393 segundos

Error que empieza en la línea: 96 del comando :

```
connect ...
Informe de error -
Fallo de conexión
USER          = profel
URL            = jdbc:oracle:thin:@localhost:1521:xe
Mensaje de Error = ORA-02391: exceeded simultaneous SESSIONS_PER_USER limit
Commit
```