Desarrollo de Aplicaciones Multiplataforma, Desarrollo de Aplicaciones Web.

SQL: DCL y extensión procedimental, casos prácticos I

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     1. 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.

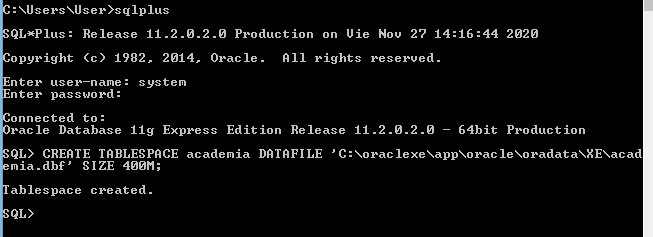
De los alumnos queremos guardar.

* 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;

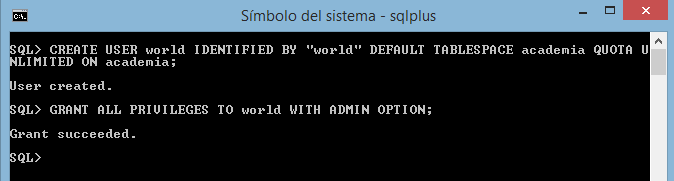


1. Crea un usuario llamado “world” que tenga todos los privilegios en el sistema Oracle. Comprobar que realmente tiene asignados estos permisos.

Primero creamos al usuario y le damos los permisos:

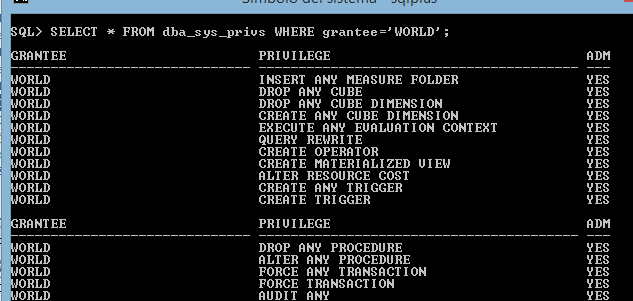
CREATE USER world IDENTIFIED BY "world" DEFAULT TABLESPACE academia QUOTA UNLIMITED ON academia;

GRANT ALL PRIVILEGES TO world WITH ADMIN OPTION;



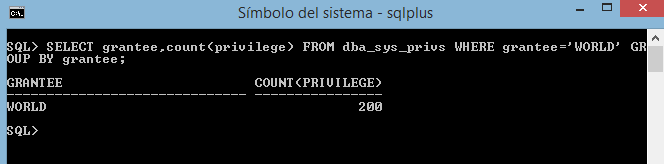
Comprobamos que tiene todos los permisos:

SELECT \* FROM dba\_sys\_privs WHERE grantee='WORLD';



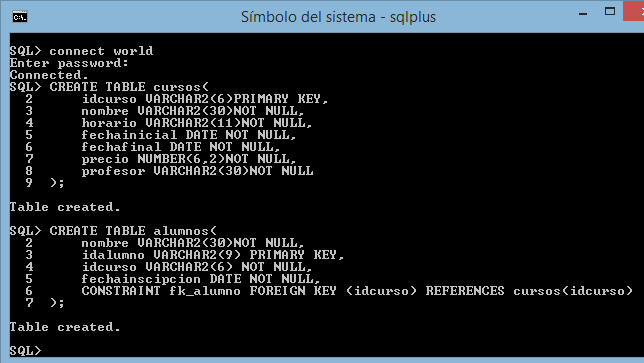
Oracle tiene 200 permisos, vamos a ver que los tenga todos:

SELECT grantee,count(privilege) FROM dba\_sys\_privs WHERE grantee='WORLD' GROUP BY grantee;

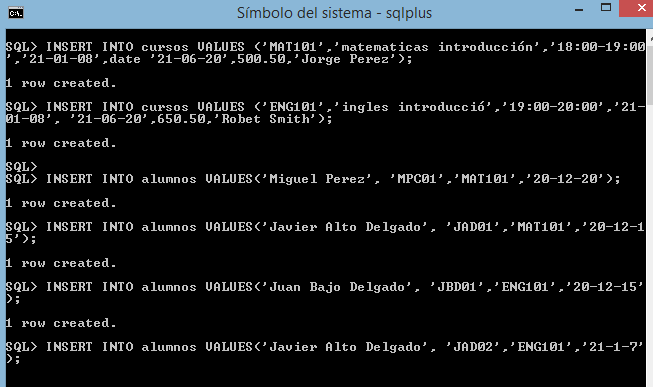


1. Utilizando el usuario de nombre “world” crea la tabla cursos y la tabla alumnos. Utilizar el tipo de campo y la longitud que creáis más adecuados para cada uno de los campos. Introduce datos en las tablas.

Nos conectamos con “world” y creamos las tablas:



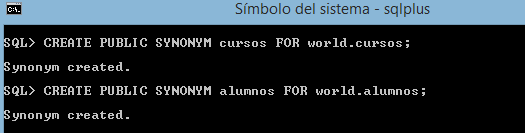
Las rellenamos:



**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;

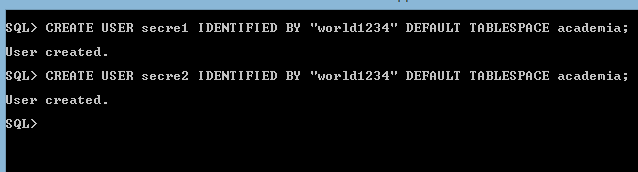


1. Crear dos usuarios “secre1” y “secre2” con password “world1234”, que se encarguen de la gestión de la academia (añadir, modificar, borrar, consultar) en la tabla de cursos y en la tabla de alumnos. Comprueba que los privilegios se han asignado de forma correcta y que puede hacer las operaciones asignadas.

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;

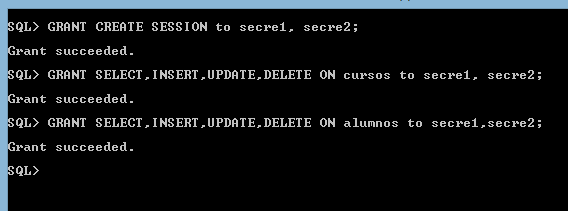


Les damos los privilegios:

GRANT CREATE SESSION TO secre1, secre2;

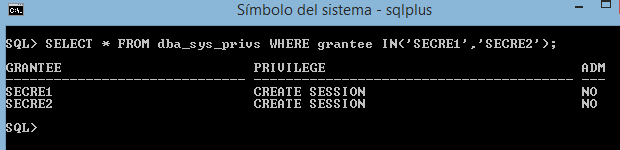
GRANT SELECT,INSERT,UPDATE,DELETE ON cursos TO secre1, secre2;

GRANT SELECT,INSERT,UPDATE,DELETE ON alumnos TO secre1,secre2;



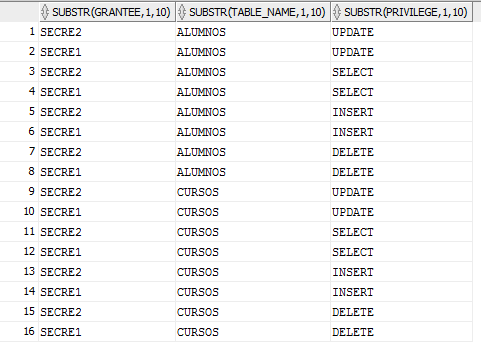
Y comprobamos que privilegios tienen, en primer lugar, veremos que pueden iniciar sesión y después los privilegios sobre las tablas:

SELECT \* FROM dba\_sys\_privs WHERE grantee IN('SECRE1','SECRE2');



SELECT substr(grantee,1,10),substr(table\_name,1,10),substr(privilege,1,10)

FROM dba\_tab\_privs WHERE grantee IN('SECRE1','SECRE2');

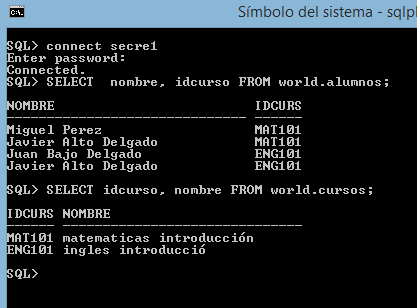


Para una segunda prueba nos conectaremos con secre1 e intentaremos hacer un SELECT en cada tabla.

CONNECT secre1;

SELECT nombre, idcurso FROM world.alumnos;

SELECT idcurso, nombre FROM world.cursos;



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.

1. Se decide que el usuario “secre1” pueda crear nuevos usuarios, pero no podrá eliminar a ningún usuario. Comprobar que realmente tiene asignados estos permisos.

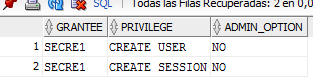
Quitamos el permiso de borrar registros al usuario “secre2” sobre la tabla de cursos. Comprobación.

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:

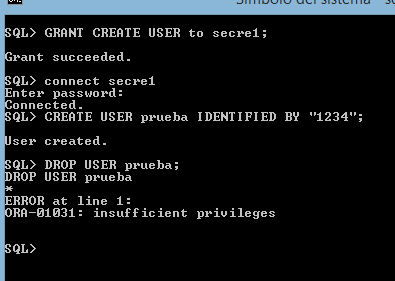
SELECT \* FROM dba\_sys\_privs WHERE grantee='SECRE1';



CONNECT secre1;

CREATE USER prueba IDENTIFIED BY "1234";

DROP USER prueba;



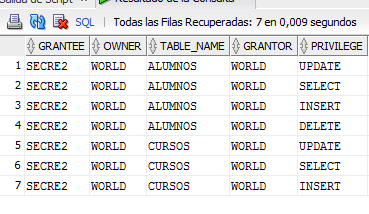
Ahora le quitaremos el permiso de registros a secre2 sobre la tabla cursos y lo comprobaremos:

REVOKE DELETE ON cursos FROM secre2;



Y lo comprobaremos

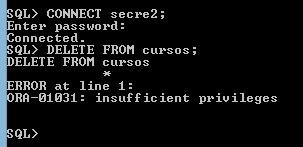
SELECT \* FROM dba\_tab\_privs WHERE grantee='SECRE2';



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;



1. 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;

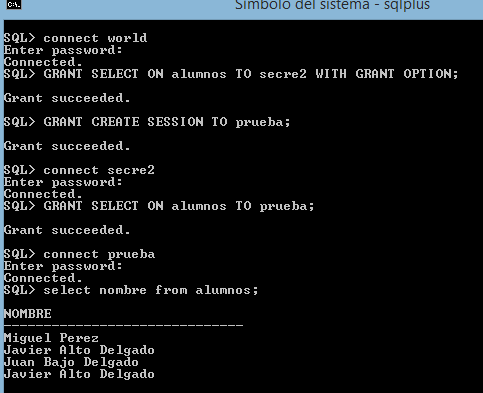
GRANT SELECT ON alumnos TO secre2 WITH GRANT OPTION;

GRANT CREATE SESSION TO prueba;

CONNECT secre2;

GRANT SELECT ON alumnos TO prueba;

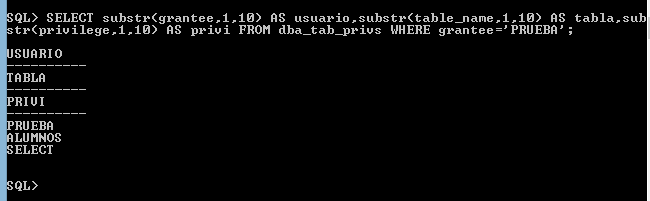
SELECT nombre FROM alumnos;



Comprobaremos que solo tiene ese privilegio con el código:

SELECT substr(grantee,1,10) AS usuario,substr(table\_name,1,10) AS tabla,sub

str(privilege,1,10) AS privi FROM dba\_tab\_privs WHERE grantee='PRUEBA';



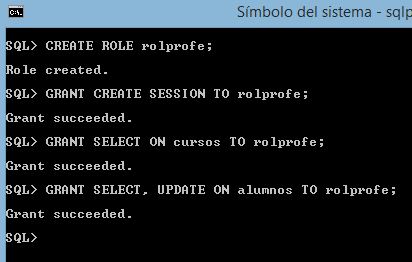
1. Crea un rol llamado rolprofe con las siguientes características: Puede iniciar sesión, leer la tabla de cursos y leer y modificar la tabla de alumnos (no puede ni borrar ni añadir).

CREATE ROLE rolprofe;

GRANT CREATE SESSION TO rolprofe;

GRANT SELECT ON cursos TO rolprofe;

GRANT SELECT, UPDATE ON alumnos TO rolprofe;



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

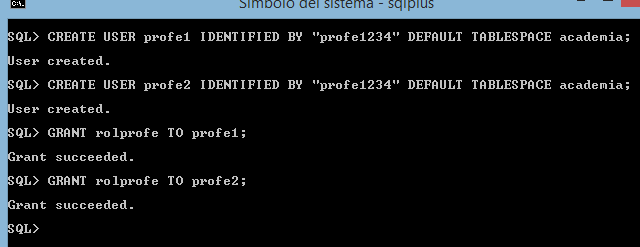
Primero crearemos los usuarios y les daremos el rol desde el usuario “world”:

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;

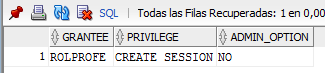


Ahora comprobaremos que tienen los privilegios que hemos dado;

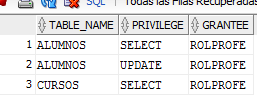
Lo podemos hacer de 2 formas, mirando la información de los archivos de la base de datos o probando si somos capaces de conectarnos y hacer las acciones que se nos pide.

Primero probaremos mediante los archivos de la base de datos:

SELECT \* FROM dba\_sys\_privs WHERE grantee='ROLPROFE';



SELECT table\_name,privilege, grantee FROM dba\_tab\_privs WHERE grantee='ROLPROFE';



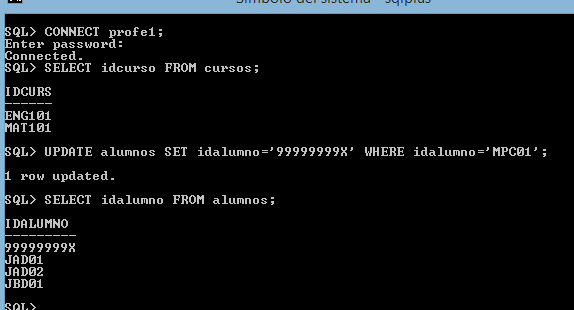
Ahora mediante el método de prueba/error a ver si nos deja ejecutar las acciones.

CONNECT profe1;

SELECT idcurso FROM cursos;

UPDATE alumnos SET idalumno='99999999X' WHERE idalumno='MPC01';

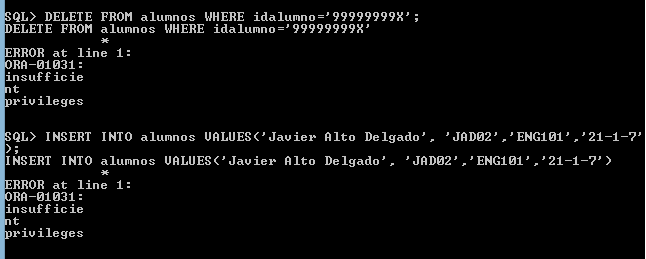
SELECT idalumno FROM alumnos;



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');



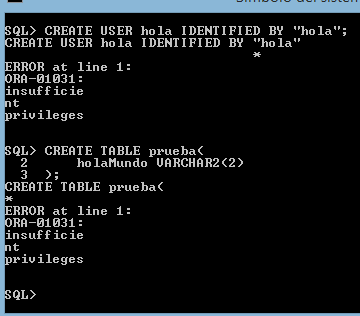
Intentaremos otro tipo de privilegios como el de crear usuarios o tablas:

CREATE USER hola IDENTIFIED BY "hola";

CREATE TABLE prueba(

holaMundo VARCHAR2(2)

);



1. 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;



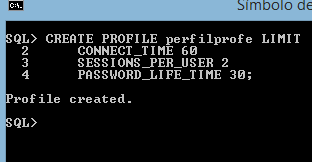
Ahora crearemos el perfil:

CREATE PROFILE perfilprofe LIMIT

CONNECT\_TIME 60

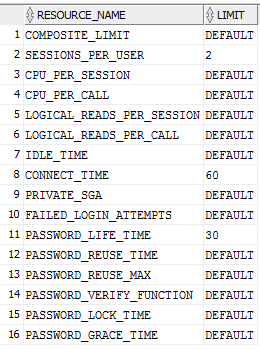
SESSIONS\_PER\_USER 2

PASSWORD\_LIFE\_TIME 30;



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

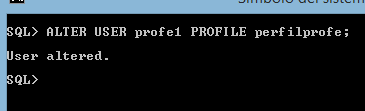
SELECT resource\_name,substr(limit,1,10) FROM dba\_profiles WHERE profile='PERFILPROFE';



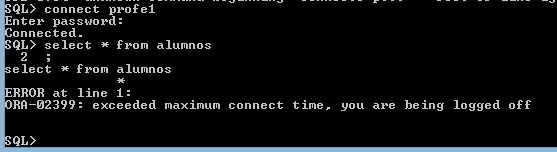
Para activar los perfiles tendremos que usar la instrucción:

Ahora lo asignamos a profe1:

ALTER USER profe1 PROFILE perfilprofe;



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.



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

