

Practica 6-4-Solución

- 1.- Introducir algún valor en las tablas del **usuario1**

Podemos hacerlo dese SYSTEM o desde usuario1.

INSERT INTO usuario1.alumno **VALUES** (1111, 'Jose');

INSERT INTO usuario1.alumno **VALUES** (2222, 'Ana');

INSERT INTO usuario1.alumno **VALUES** (3333, 'Antonio');

INSERT INTO usuario1.alumno **VALUES** (4444, 'Maria');

INSERT INTO usuario1.profesor **VALUES** (1010,'Fernando');

INSERT INTO usuario1.profesor **VALUES** (2020,'Eva');

INSERT INTO usuario1.modulo **VALUES** ('PRO', 'Programacion',1010);

INSERT INTO usuario1.modulo **VALUES** ('ING', 'Ingles',2020);

INSERT INTO usuario1.modulo **VALUES** ('MAR', 'Marcas',1010);

INSERT INTO usuario1.matricula **VALUES** (1111,'PRO',6,'S');

INSERT INTO usuario1.matricula **VALUES** (1111,'MAR',7,'S');

INSERT INTO usuario1.matricula **VALUES** (2222,'PRO',8,'N');

INSERT INTO usuario1.matricula **VALUES** (2222,'MAR',6,'N')

INSERT INTO usuario1.matricula **VALUES** (2222,'ING',8,'N');

INSERT INTO usuario1.matricula **VALUES** (3333,'PRO',6,'S');

INSERT INTO usuario1.matricula **VALUES** (3333,'ING',8,'S');

INSERT INTO usuario1.matricula **VALUES** (4444,'PRO',6,'S');

- 2.- Crear un usuario llamado **Usuario2** con la misma contraseña y asignarle los roles o privilegios estándar, para que pueda conectarse y crear tablas.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE USER** Usuario2 **IDENTIFIED BY** Usuario2;

GRANT CONNECT TO usuario2;

GRANT RESOURCE TO usuario2;

- 3.- Realizar las acciones oportunas para que el usuario **Usuario2** pueda consultar (SELECT) la tabla ALUMNO del usuario **Usuario1**.

Usuario desde el que se realiza la acción: **Usuario1** o también desde **SYSTEM**

Instrucción SQL: **GRANT SELECT ON** Usuario1.alumno **TO** Usuario2;

- 4.- Acceder con dicho usuario (usuario2) a la BD y comprobar en su esquema las tablas que dispone el usuario.

Usuario desde el que se realiza la acción: **Usuario2**

Instrucción SQL: **SELECT * FROM TABS;**

Permite ver las tablas que tiene el usuario

NOTA: SELECT * FROM USER_TAB_PRIVS;

Permite ver los privilegios que tiene un usuario sobre tablas de otro usuario

- 5.- Volver a comprobar que tablas tiene el usuario **Usuario2**. ¿Por qué no aparece la tabla ALUMNO?

NO tiene tablas en su esquema, y no se muestra la tabla ALUMNO porque esta tabla pertenece al esquema de otro usuario Usuario1

- 6.- Comprobar que desde dicho usuario **Usuario2** puede consultar la tabla ALUMNO

Usuario desde el que se realiza la acción: **Usuario2**

Instrucción SQL: **SELECT * FROM Usuario1.alumno;**

Como la tabla está en otro esquema de usuario, es necesario indicar el esquema donde se encuentra dicha tabla.

- 7.- Permitir que el usuario **Usuario2** puede modificar, insertar y borrar (UPDATE, DELETE e INSERT) en la tabla ALUMNO

Usuario desde el que se realiza la acción: **Usuario1 o SYSTEM**

Instrucción SQL: **GRANT UPDATE, DELETE, INSERT ON Usuario1.alumno TO Usuario2;**

- 8.- Para probar que el **usuario2** tiene los permisos, insertar un nuevo alumno desde el usuario2 en la tabla alumno.

Usuario desde el que se realiza la acción: **Usuario2**

Instrucción SQL: **INSERT INTO usuario1.alumno VALUES (5555, 'Alvaro');**

- 9.- Permitir que el usuario **Usuario2** pueda otorgar el permiso de consulta sobre la tabla ALUMNO a otros usuarios.

Usuario desde el que se realiza la acción: **Usuario1**

Instrucción SQL: **GRANT SELECT ON usuario1.ALUMNO TO usuario2 WITH GRANT OPTION;**

10.- Crear un usuario llamado **Usuario3** con la misma contraseña y asignarle los roles o privilegios estándar, para que pueda conectarse y crear tablas.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE USER Usuario3 IDENTIFIED BY Usuario3;**
GRANT CONNECT TO usuario3;
GRANT RESOURCE TO usuario3;

11.- Permitir que el usuario2 otorgue al usuario3 el permiso de consulta a la tabla alumno del usuario1

Usuario desde el que se realiza la acción: **usuario2**

Instrucción SQL: **GRANT SELECT ON usuario1.alumno TO usuario3;**

12.- Comprobar que el usuario 3 puede consultar la tabla alumno.

Usuario desde el que se realiza la acción: **usuario3**

Instrucción SQL: **SELECT * FROM usuario1.alumno;**

13.- Instrucciones para permitir al usuario **Usuario2** acceder a consultar (SELECT) todas las tablas del usuario Usuario1

Usuario desde el que se realiza la acción: **usuario1**

Instrucción SQL: Sería hacer un **GRANT** a cada una de las tablas

GRANT SELECT ON usuario1.modulo to usuario2;
GRANT SELECT ON usuario1.profesor to usuario2;
GRANT SELECT ON usuario1.matricula to usuario2;

14.- Realizar las acciones necesarias para que el usuario **Usuario2** ya no pueda ni actualizar ni insertar en las tablas del esquema del usuario Usuario1

Usuario desde el que se realiza la acción: **Usuario1**

Instrucción SQL: **REVOKE DELETE, INSERT, SELECT, UPDATE ON usuario1.alumno FROM Usuario2;**

Sería realizar el mismo proceso sobre cada una de las tablas de Usuario1
Pero como no los tienes el resto de las tablas no se los podemos quitar

15.- Descargar el fichero **provincias.sql** y lanzar su ejecución dentro del esquema **Usuario2** para que se cree una tabla **pronorte**. Comprobar cuantas tablas son propiedad del usuario Usuario2

- Ejecutar **SQLPLUS** y conectarse con el usuario2.
- Ejecutar **start nombrerefichero** con ruta por ejemplo: **start e:provincias.sql;**



- 16.- Conéctate con **Usuario2**. ¿Por qué no se muestran las tablas ALUMNOS, CURSOS, PROFESOR, MATRICULA?

Son tablas de otro esquema de usuario, no del usuario Usuario2

- 17.- Comprobar que se muestra si ejecutamos uno de los siguientes comandos:

```
select * from USER_TAB_PRIVS_RECD;  
select * from USER_TAB_PRIVS;
```

Muestran los privilegios que tiene otorgado el usuario usuario2

- 18.- Realizar las acciones oportunas para que la tabla **pronorte** pueda ser consultada por cualquier usuario

Usuario desde el que se realiza la acción: **Usuario2**

Instrucción SQL: **GRANT SELECT ON Usuario2.pronorte TO PUBLIC**

- 19.- Dar privilegio al usuario **Usuario1** para que pueda crear tablas en cualquier esquema.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **GRANT CREATE ANY TABLE TO Usuario1;**

- 20.- Comprobar si el usuario **Usuario1** puede crear una tabla llamada MITABLA con un campo cualquiera, en el esquema del usuario Usuario3

Usuario desde el que se realiza la acción: **Usuario1**

Instrucción SQL: **CREATE TABLE** usuario3.mitabla

```
(  
    Campo varchar2(20)  
)
```

- 21.- Quitarle los permisos al usuario **Usuario1** para que no pueda crear tablas en cualquier esquema.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **REVOKE CREATE ANY TABLE FROM Usuario1;**

- 22.- Quitar los permisos necesarios para que la tabla **pronorte** no pueda ser consultada por cualquier usuario

Usuario desde el que se realiza la acción: **Usuario2**

Instrucción SQL: **REVOKE SELECT ON pronorte FROM PUBLIC**

23.- Crea los siguientes usuarios: usuario4, usuario5, usuario6. Asignarle los roles o privilegios estándar (CONNECT y RESOURCE.)

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE USER Usuario4 IDENTIFIED BY Usuario4;**

CREATE USER Usuario5 IDENTIFIED BY Usuario5;

CREATE USER Usuario6 IDENTIFIED BY Usuario6;

GRANT CONNECT,RESOURCE TO usuario4, usuario5, usuario6;

24.- Crea un rol que se llame **lectura** y otorgale los privilegios de consulta a la tabla pronorte.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE ROLE lectura;**

GRANT SELECT ON usuario2.pronorte TO lectura;

25.- Crea un rol que se llame **escritura** y otorgale los privilegios de consulta y modificación a la tabla alumnos y a la tabla pronorte.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE ROLE escritura;**

GRANT SELECT, UPDATE ON usuario1.alumno TO escritura;

GRANT SELECT, UPDATE ON usuario2.pronorte TO escritura;

26.- Otorga el **rol lectura** a los usuarios usuario4, usuario5 y usuario6.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **GRANT lectura TO usuario4, usuario5, usuario6;**

27.- Otorga el **rol escritura** a los usuarios usuario5, usuario6

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **GRANT escritura TO usuario5, usuario6;**