Prácticas de Administración de Bases de Datos

Grado en Ingeniería Informática

PRÁCTICA 10

Seguridad en las Bases de Datos
(privilegios, roles)

SOLUCIONES

OBJETIVOS

- Adquirir destreza en la gestión de la confidencialidad
- Aprender a gestionar privilegios y roles

Ejercicios de privilegios y roles

Para realizar estos ejercicios vamos a partir de una base de datos sencilla (con tres tablas). Esto nos servirá para recordar la sintaxis de la sentencia CREATE TABLE.

La base de datos mantiene información acerca de los alumnos de una academia de preparación para oposiciones donde se imparten cursos de diferentes áreas. La descripción de las tablas es la siguiente:

TABLA	COLUMNAS	TIPO	COMENTARIO
ESTUDIANTE	nombre	texto(40)	nombre del estudiante
	dni	texto(8)	clave primaria de la tabla
	fechaNac	fecha	fecha de nacimiento
	curso	texto(3)	código del curso en el que está matriculado un estudiante. Clave ajena de la tabla CURSO

CURSO	nomCurso	texto(20)	nombre del curso
	codCurso	texto(3)	clave primaria de la tabla
	profesor	texto(8)	nombre del profesor que imparte el curso. Clave ajena de la tabla PROFESOR
	horas	numérico(3)	número de horas del curso

PROFESOR	nombre	texto(40)	nombre del profesor
	dni	texto(8)	clave primaria de la tabla
	sueldo	real	sueldo del profesor

1. Mediante la sentencia SQL CREATE USER, crear un usuario (que será el propietario de las tablas), con las siguientes características:

   - El usuario debe ser identificado por Oracle
   - Nombre: **admin**
   - Contraseña: **admin**
   - El espacio de tablas por defecto debe ser ET_USUARIOS (se supone que ya lo habéis creado en práctica anterior)
   - El espacio de tabla temporal será TEMP
   - La cuota de usuario será de 2M sobre ET_USUARIOS
   - Asignar al usuario el perfil ESTUDIANTE_ABD creado anteriormente

```
CREATE USER admin
IDENTIFIED BY admin
DEFAULT TABLESPACE ET_USUARIOS
QUOTA 2M ON ET_USUARIOS
TEMPORARY TABLESPACE TEMP
PROFILE ESTUDIANTE_ABD;
```

2. Crear el rol ***"mi_conexion"*** con los privilegios para poder conectarse y crear tablas

```
CREATE ROLE mi_conexion;
GRANT CREATE SESSION, CREATE TABLE TO mi_conexion;
```

3. Desde el usuario SYS, otorgar al usuario ***"admin"*** el rol ***"mi_conexion"***

```
GRANT mi_conexion TO admin;
```

4. Comprobar en la vista correspondiente cuales son los roles asignados al usuario ***"admin"***

```
select * from dba_role_privs where grantee = 'ADMIN';
```

5. Consultar la tabla o vista adecuada para saber en qué espacios de tabla puede escribir el usuario ***"admin"***

```
select USERNAME, TABLESPACE_NAME, BYTES
from dba_ts_quotas
where username = 'ADMIN';
```

6. Con el usuario ***"admin"***, ejecutar el script de creación de las tablas **(es conveniente que, además de ejecutar el script, le echéis un vistazo a la sintaxis para recordar conceptos)**

```
drop table profesor cascade constraint;
create table PROFESOR (nombre varchar2(40) not null,
                       dni char(8),
                       sueldo number,
                       constraint profClave primary key(dni));


drop table curso cascade constraint;
create table CURSO (nomCurso varchar2(20) not null,
                    codCurso char(3),
                    profesor char(8),
                    horas number,
                    constraint cursosClave primary key (codCurso),
                    constraint cursosAjena foreign key (profesor) references
                PROFESOR(dni));


drop table alumno cascade constraint;
create table ESTUDIANTE (nombre varchar2(40) not null,
                         dni char(8),
                         fechaNac date,
                         curso char(3) not null,
                         constraint estClave primary key(dni),
                         constraint  estAjena  foreign  key  (curso)  references
                CURSO(codCurso));

commit;
```

7. Asignar cuota cero en el espacio de tablas por defecto para el usuario ***"admin"***. Vuelve a consultar la tabla o vista adecuada para saber en qué espacios de tabla puede escribir el usuario ***"admin".*** Comprobar si Siguen estando sus objetos

```
alter user admin quota 0 on ET_USUARIOS;

select USERNAME, TABLESPACE_NAME, BYTES
from dba_ts_quotas
where username = 'ADMIN';


    select owner, table_name
    from dba_tables
    where owner='ADMIN';
```

8. Cambiarle la cuota a 2 Megas sobre ET_USUARIOS al usuario admin y conectados como dicho usuario ejecutar el script ***InsertaDatos.sql***.

```
alter user admin quota 2M on ET_USUARIOS;
```

9. Desde el usuario ***"estudiante"*** intentar consultar los datos introducidos en las tablas. ¿Es posible?, ¿por qué?, ¿cuál es el mensaje de error?, ¿piensas que es intuitivo el mensaje de error?

```
select * from admin.estudiante;
```

10. Desde el usuario ***"admin"***, otorgar al usuario ***"estudiante"*** privilegios de consulta sobre la tabla **ESTUDIANTE**. Recordad validar cada operación para que sea efectiva para el otro usuario (comando ***commit***).

```
grant select on estudiante to estudiante;
commit;
```

11. Volver a intentar el ejercicio 9 con la tabla **ESTUDIANTE**

12. Desde los usuarios ***"admin"*** y ***"estudiante"***, consultar las tablas del diccionario de datos. Para cada tabla (USER_TAB_PRIVS, USER_TAB_PRIVS_MADE y USER_TAB_PRIVS_RECD) proporcionar una breve explicación de cada campo

13. Otorgar privilegio de consulta al usuario ***"estudiante"*** sobre todos los atributos de la tabla **PROFESOR** excepto del sueldo (recuerda que al privilegio SELECT no se le puede indicar, explícitamente, las columnas para consultar). Buscar un mecanismo que haga la misma función.

```
Si admin no tiene privilegios..
grant create view to admin

Creamos una vista con los dos campos

CREATE VIEW PROFESOR_001 AS
SELECT nombre, dni FROM PROFESOR;
commit;


Otorgamos el privilegio a la vista

GRANT SELECT ON PROFESOR_001 TO estudiante;
COMMIT;
```

Ahora, el usuario *estudiante* puede consultar los campos "dni" y "nombre" de la tabla PROFESOR (que para él es la vista PROFESOR_001)

```
SELECT * FROM admin.PROFESOR_001;
```

14. Otorgar privilegios de consulta al usuario ***"estudiante"*** sobre todos los atributos de la tabla **CURSO** y de inserción sobre todos los campos excepto el D.N.I. del profesor. Comentar el resultado obtenido al intentar realizar las siguientes operaciones desde la cuenta del usuario ***"estudiante"***:

    - Insertar una tupla completa en la tabla CURSO
    - Insertar una tupla para un CURSO sin poner el D.N.I. del profesor
    - Visualizar el nuevo contenido de la tabla CURSO

```
GRANT SELECT, INSERT(nomCurso,codCurso,horas)
ON CURSO TO estudiante;
commit;

INSERT INTO admin.CURSO VALUES ('Nuevo Curso','C04','29123123',300);

ERROR en línea 1:
ORA-01031: privilegios insuficientes

INSERT INTO admin.CURSO (nomCurso,codCurso,horas)
VALUES ('Nuevo Curso','C04',300);

1 fila creada.
```

Comprobación

```
SELECT * FROM admin.CURSO;

NOMCURSO             COD PROFESOR      HORAS
-------------------- --- -------- ----------
Constitución         C01 29123123        150
Informática Contable C02 29444555        350
Ofimática            C03 12365478        200
Nuevo Curso          C04                 300

4 filas seleccionadas.
```

15. Desde el usuario ***"admin"*** crear el rol ***"PRIVADO"*** identificado por la contraseña ***"priv"***

```
Si no tiene privilegio…
Desde system grant create role to admin
Ya en admin
create role privado identified by priv;
```

16. Desde el usuario ***"admin"*** asignarle el rol ***"PRIVADO"*** al usuario ***"estudiante"***

```
grant privado to estudiante;
```

17. Comprobar los roles asignados al usuario ***"estudiante"***. ¿El rol ***"PRIVADO"*** es un rol por defecto?

```
select * from dba_role_privs where grantee='ESTUDIANTE';
```

18. Desde el usuario ***"admin"***, hacer que el rol ***"PRIVADO"*** no esté activo para el usuario ***"estudiante"*** cuando se conecte

```
alter user estudiante default role all except privado;
```

19. Comprobar de nuevo los roles asignados al usuario ***"estudiante"***. ¿El rol ***"PRIVADO"*** es ahora un rol por defecto?

```
select * from dba_role_privs where grantee='ESTUDIANTE';
```

20. Con el usuario ***"estudiante"*** comprobar en la vista apropiada del diccionario de datos los roles activos en su sesión. Desde el propio usuario, activar el rol ***"PRIVADO"*** y volver a hacer la comprobación

```
select * from session_roles;
set role privado identified by priv;
select * from session_roles;
```

21. La política de seguridad de la empresa está centralizada en el administrador de la base de datos, que debe decidir qué tipos de privilegios deben concederse a cada usuario para el acceso a las tablas **PROFESOR**, **CURSO** y **ESTUDIANTE**. Después de realizar un estudio exhaustivo, el ABD ha decidido crear los siguiente grupos:

Grupo Estudiantes

Formado por los estudiantes de la academia. Estos sólo podrán consultar datos no significativos de la BD:

- Nombre de los estudiantes
- Información de los cursos excepto el dni del profesor
- Nombre de los profesores

Grupo Profesores

Este grupo, además de tener los privilegios del grupo de alumnos, también podrá modificar y eliminar información de la tabla PROFESOR

Grupo Gestores

Este grupo, al que pertenece el usuario ***"admin"***, debe tener todos los privilegios sobre todas las tablas

Teniendo en cuenta esta información, desde el usuario ***"admin"*** realizar los siguientes ejercicios:

22. Crear todos los roles necesarios para implementar esta política de seguridad, asignándole a cada rol los privilegios convenientes en cada caso.

```
1. ROL PARA LOS ESTUDIANTES
CREATE ROLE rol_estudiante;

Creamos una vista con el campo "nombre"

CREATE VIEW ESTUDIANTE_NOMBRE AS
SELECT nombre FROM ESTUDIANTE;
commit;


Otorgamos el privilegio de consulta al rol

GRANT SELECT ON ESTUDIANTE_NOMBRE TO rol_estudiante;
COMMIT;

Hacemos la misma operación para las otras tablas

CREATE VIEW CURSO_SIN_PROFESOR AS
SELECT nomCurso, codCurso, horas FROM CURSO;
commit;

GRANT SELECT ON CURSO_SIN_PROFESOR TO rol_estudiante;
COMMIT;

CREATE VIEW PROFESOR_NOMBRE AS
SELECT nombre FROM PROFESOR;
commit;

GRANT SELECT ON PROFESOR_NOMBRE TO rol_estudiante;
COMMIT;

2. ROL PARA LOS PROFESORES
CREATE ROLE rol_profesor;

GRANT rol_estudiante TO rol_profesor;
GRANT UPDATE, DELETE ON PROFESOR TO rol_profesor;



3. ROL PARA LOS GESTORES
CREATE ROLE rol_gestor;
```

```
GRANT ALL PRIVILEGES ON ESTUDIANTE TO rol_gestor;
GRANT ALL PRIVILEGES ON CURSO TO rol_gestor;
GRANT ALL PRIVILEGES ON PROFESOR TO rol_gestor;
```

23. Para verificar que se ha realizado bien el ejercicio anterior, crear un nuevo usuario y asignar/revocar diferentes roles, comprobando que cuando lo tiene asignado puede hacer las operaciones que se piden pero no puede hacer otras que tenga prohibidas