

Ejercicios propuestos de SQL Oracle

- 1 Crear un *tablespace* de nombre COMPRAS asociándole un fichero en disco llamado 'COMPRAS.ORA' de 5 Megabytes.

```
SQL> CREATE TABLESPACE COMPRAS DATAFILE 'COMPRAS.ORA' SIZE 5M,  
2  
SQL>  
SQL>
```

- 2 Modificar el *tablespace* del ejercicio 1 para que pueda autoextenderse automáticamente, sin límite de espacio en disco.

```
SQL> create tablespace COMPRAS datafile 'COMPRAS.ORA' size 5M;  
Tablespace creado.  
SQL> alter tablespace COMPRAS add datafile 'COMPRAS1.ORA' size 1M autoexten  
next 1M maxsize unlimited;  
Tablespace modificado.
```

-
- 3 Crear un rol que tenga los siguientes privilegios: INSERT y SELECT en DEPART y EMPLE, CREATE SESSION, CREATE DATABASE LINK, CREATE TABLE y CREATE VIEW.

```
SQL> create role USUARIO;
```

```
Rol creado.
```

```
SQL> create role USUARIO;
```

```
Rol creado.
```

```
SQL> grant insert, select on depart to USUARIO;
```

```
Concesión terminada correctamente.
```

```
SQL>
```

```
Rol creado.
```

```
SQL> grant insert, select on depart to USUARIO;  
SP2-0734: inicio "grant inse..." de comando desconocido - resto de la línea ignorado.
```

```
SQL> grant insert, select on depart to USUARIO;
```

```
Concesión terminada correctamente.
```

```
SQL> grant insert, select on emple to USUARIO;
```

```
Concesión terminada correctamente.
```

```
SQL> _
```

```
Rol creado.  
SQL> grant insert, select on depart to USUARIO;  
SP2-0734: inicio "grant inse..." de comando desconocido - resto de la línea igno  
rado.  
SQL> grant insert, select on depart to USUARIO;  
Concesión terminada correctamente.  
SQL> grant insert, select on emple to USUARIO;  
Concesión terminada correctamente.  
SQL> grant create session, create database link, create table, create view to US  
UARIO;  
Concesión terminada correctamente.  
SQL> _
```

- ④ Crear un usuario llamado COMPRADOR. El *tablespace*, por defecto, es COMPRAS. Se le asigna 1 Megabyte en el *tablespace* COMPRAS y 0 de cuota en el *tablespace* SYSTEM. El *tablespace* temporal será TEMPORARY_DATA. Se le asigna el rol anterior.

```
SQL> create user COMPRADOR identified by COMPRADOR default tablespace COMPRAS te  
mporary tablespace TEMP quota 1M on COMPRAS quota 0 on system;  
Usuario creado.  
SQL> _
```

```
SQL> create user COMPRADOR identified by COMPRADOR default tablespace COMPRAS te  
mporary tablespace TEMP quota 1M on COMPRAS quota 0 on system;  
Usuario creado.  
SQL> grant USUARIO to COMPRADOR;  
Concesión terminada correctamente.  
SQL>
```

- 5** Realizar la siguiente secuencia de instrucciones en el orden indicado:

1. Crear un usuario de base de datos que tenga funciones de administrador.

```
SQL> create user ADMINISTRADOR identified by ADMINISTRADOR;
Usuario creado.
SQL> grant DBA to ADMINISTRADOR;
Concesión terminada correctamente.
SQL>
```

- 5** Realizar la siguiente secuencia de instrucciones en el orden indicado:

1. Crear un usuario de base de datos que tenga funciones de administrador.
2. Conectarse con el nombre de usuario creado.

```
SQL> connect ADMINISTRADOR;
Introduzca la contraseña:
Conectado.
SQL> _
```

- 5** Realizar la siguiente secuencia de instrucciones en el orden indicado:

1. Crear un usuario de base de datos que tenga funciones de administrador.
2. Conectarse con el nombre de usuario creado.
3. Crear varias tablas en el propio esquema.

```
SQL> connect ADMINISTRADOR;
Introduzca la contraseña:
Conectado.
SQL> CREATE TABLE CLIENTE (
  2  DNI VARCHAR2<10>,
  3  NOMBRE VARCHAR2<20>,
  4  FECHA_NAC DATE NOT NULL,
  5  DIRECCION VARCHAR2<200>,
  6  TFNO VARCHAR2<10>,
  7  APELLIDOS VARCHAR2<100>,
  8  CONSTRAINT CLIENTE_PK PRIMARY KEY <DNI>);

Tabla creada.

SQL> CREATE TABLE PROVEEDOR(
  2  NIF VARCHAR2<10>,
  3  NOMBRE VARCHAR2<20>,
  4  DIRECCION VARCHAR2<200>,
  5  TFNO VARCHAR2<10>,
  6  CONSTRAINT PROVEEDOR_PK PRIMARY KEY <NIF>);

Tabla creada.

SQL> CREATE TABLE PRODUCTO<
  2  NOMBRE VARCHAR2<50>,
  3  CODIGO VARCHAR2<50>,
  4  PRECIO NUMBER<10> NOT NULL,
  5  PRU_NIF VARCHAR2<10>,
  6  CONSTRAINT PRODUCTO_PK PRIMARY KEY <CODIGO>,
  7  CONSTRAINT PROVEEDOR_FK FOREIGN KEY<PRU_NIF> REFERENCES PROVEEDOR <NIF>;

Tabla creada.

SQL> CREATE TABLE CLIENTE_PRODUCTO<
  2  PRD_CODIGO VARCHAR<50>,
  3  CLN_DNI VARCHAR<10>,
  4  CONSTRAINT CLIENTE_PRODUCTO_PK PRIMARY KEY <PRD_CODIGO, CLN_DNI>,
  5  CONSTRAINT PRODUCTOCP_FK FOREIGN KEY <PRD_CODIGO> REFERENCES PRODUCTO<CODIGO>,
  6  CONSTRAINT CLIENTECP_FK FOREIGN KEY <CLN_DNI> REFERENCES CLIENTE<DNI>;

Tabla creada.

SQL>
```

- 5) Realizar la siguiente secuencia de instrucciones en el orden indicado:

1. Crear un usuario de base de datos que tenga funciones de administrador.
2. Conectarse con el nombre de usuario creado.
3. Crear varias tablas en el propio esquema.
4. Crear 5 usuarios nuevos asignándoles un *tablespace* por defecto y cuota (USU1, USU2, USU3, USU4 y USU5).

```
SQL> CREATE USER USU1 IDENTIFIED BY USU1 DEFAULT TABLESPACE
  2  COMPRAS
  3  QUOTA 2M ON COMPRAS;

Usuario creado.

SQL>
SQL> CREATE USER USU2 IDENTIFIED BY USU2 DEFAULT TABLESPACE
  2  COMPRAS
  3  QUOTA 2M ON COMPRAS;

Usuario creado.

SQL>
SQL> CREATE USER USU3 IDENTIFIED BY USU3 DEFAULT TABLESPACE
  2  COMPRAS
  3  QUOTA 2M ON COMPRAS;

Usuario creado.

SQL>
SQL> CREATE USER USU4 IDENTIFIED BY USU4 DEFAULT TABLESPACE
  2  COMPRAS
  3  QUOTA 2M ON COMPRAS;

Usuario creado.

SQL>
SQL> CREATE USER USU5 IDENTIFIED BY USU5 DEFAULT TABLESPACE
  2  COMPRAS
  3  QUOTA 2M ON COMPRAS;

Usuario creado.

SQL> _
```

6. Crear un rol que permita conectarse a la base de datos y hacer SELECT sobre algunas tablas.

```
SQL> GRANT CREATE SESSION TO ROL_USU;
Concesión terminada correctamente.

SQL> GRANT SELECT ON CLIENTE TO ROL_USU;
Concesión terminada correctamente.

SQL> GRANT SELECT ON PROVEEDOR TO ROL_USU;
Concesión terminada correctamente.

SQL> GRANT SELECT ON PRODUCTO TO ROL_USU;
Concesión terminada correctamente.

SQL> GRANT SELECT ON CLIENTE_PRODUCTO TO ROL_USU;
```

7. Conceder el rol creado a dos de los usuarios creados anteriormente (USU2 y USU3).

```
SQL> GRANT ROL_USU TO USU2, USU3;
Concesión terminada correctamente.

SQL>
```

- Conceder al usuario USU4 privilegios sobre algunas tablas con la opción de poder concedérselos a otros usuarios.

```
SQL> GRANT INSERT, DELETE ON CLIENTE TO USU4 WITH GRANT OPTION;
Concesión terminada correctamente.

SQL> GRANT INSERT, DELETE ON PROVEEDOR TO USU4 WITH GRANT OPTION;
Concesión terminada correctamente.

SQL>
```

- Conceder al usuario USU5 cuatro privilegios de sistema, dos de ellos, con la opción de poder concedérselos a otros usuarios.

```
SQL> GRANT CREATE ANY TABLE,CREATE USER TO USU5 WITH ADMIN
  2 OPTION;
Concesión terminada correctamente.

SQL> GRANT ALTER TABLESPACE, CREATE TABLESPACE TO USU5;
Concesión terminada correctamente.

SQL>
```

10. Conceder a todos los usuarios de la base de datos privilegios para que puedan modificar ciertas columnas de algunas tablas.

```
SQL> GRANT UPDATE(DIRECCION, TFNO) ON CLIENTE TO PUBLIC;
Concesión terminada correctamente.

SQL> GRANT UPDATE(DIRECCION, TFNO) ON PROVEEDOR TO PUBLIC;
Concesión terminada correctamente.

SQL>
```

11. Quitar a los usuarios USU3 y USU4 todos los privilegios que tenían asignados.

```
SQL> REVOKE ROL_USU FROM USU3;
Revocación terminada correctamente.

SQL> REVOKE ALL ON CLIENTE FROM USU4;
Revocación terminada correctamente.

SQL> REVOKE ALL ON PROVEEDOR FROM USU4;
Revocación terminada correctamente.
```

12. Hacer que USU5 sólo pueda conectarse en dos sesiones concurrentes a la vez.

```
SQL> GRANT CREATE ANY TABLE,CREATE USER TO USU5 WITH ADMIN  
2 OPTION;  
Concesión terminada correctamente.  
SQL> GRANT ALTER TABLESPACE, CREATE TABLESPACE TO USU5  
2  
SQL> GRANT ALTER TABLESPACE, CREATE TABLESPACE TO USU5;  
Concesión terminada correctamente.  
SQL>
```

13. Limitar el tiempo de conexión a la base de datos a 5 minutos a los usuarios USU2 y USU3.

```
SQL> CREATE PROFILE TIEMPOLIMITE LIMIT CONNECT_TIME 5;  
Perfil creado.  
SQL> ALTER USER USU2 IDENTIFIED BY USU2 PROFILE TIEMPOLIMITE;  
Usuario modificado.  
SQL> ALTER USER USU3 IDENTIFIED BY USU3 PROFILE TIEMPOLIMITE;  
Usuario modificado.  
SQL>
```

- ⑥ Crear una tabla y, seguidamente, crear una secuencia que genere números empezando en 10 y con incremento de 10. Insertar filas en la tabla creada utilizando la secuencia que se ha creado en alguna de las columnas.

```
create sequence secuencia
start with 10
increment by 10
;
```

```
SQL> create sequence secuencia
  2  start with 10
  3  increment by 10
  4 ;
Secuencia creada.
```

```
select secuencia.nextval from dual;
```

```
SQL> select secuencia.nextval from dual;
  NEXTVAL
  -----
    10
```

```
select secuencia.currrval from dual;
```

```
SQL> select secuencia.currrval from dual;
  CURRVAL
  -----
    10
```

```
create table sec (
categoriaA varchar(20),
categoriaB varchar(20),
categoriaC varchar(20) PRIMARY KEY);
```

```
SQL> create table sec (
  2  categoriaA varchar(20),
  3  categoriaB varchar(20),
  4  categoriaC varchar(20) primary key);
Tabla creada.
```

**INSERT INTO sec values
(secuencia.nextval,'Futbol3','Voley3');**

```
SQL> INSERT INTO sec values (secuencia.nextval,'Futbol3','Voley3');  
1 fila creada.
```

**INSERT INTO sec values
(secuencia.netxval,'Futbol5','Voley5');**

```
SQL> INSERT INTO sec values (secuencia.nextval,'Futbol5','Voley5');  
1 fila creada.  
SQL> select * from sec;  
CATEGORIAA          CATEGORIAB          CATEGORIAC  
-----          -----          -----  
10                  Futbol              Voley  
10                  Futbol1             Voley1  
10                  Futbol2             Voley2  
20                  Futbol3             Voley3  
20                  Futbol4             Voley4  
40                  Futbol5             Voley5
```

**INSERT INTO sec values
(secuencia.nextval,'Futbol6','Voley6');**

```
SQL> INSERT INTO sec values (secuencia.nextval,'Futbol6','Voley6');  
1 fila creada.  
SQL> select * from sec;  
CATEGORIAA          CATEGORIAB          CATEGORIAC  
-----          -----          -----  
10                  Futbol              Voley  
10                  Futbol1             Voley1  
10                  Futbol2             Voley2  
20                  Futbol3             Voley3  
20                  Futbol4             Voley4  
40                  Futbol5             Voley5  
50                  Futbol6             Voley6
```

**INSERT INTO sec values
(secuencia.nextval,'Futbol7','Voley7');.....**

```

SQL> INSERT INTO sec values (secuencia.nextval,'Futbol7','Voley7');
1 fila creada.

SQL> select * from sec;

CATEGORIAA          CATEGORIAB          CATEGORIAC
-----            -----            -----
10                  Futbol1           Voley
10                  Futbol11          Voley1
10                  Futbol2           Voley2
20                  Futbol13          Voley3
20                  Futbol4           Voley4
40                  Futbol15          Voley5
50                  Futbol16          Voley6
60                  Futbol17          Voley7

3 filas seleccionadas.

```

select * from sec;

```

C:\WINDOWS\system32\cmd.exe - sqlplus

SQL> select * from sec;

CATEGORIAA          CATEGORIAB          CATEGORIAC
-----            -----            -----
10                  Futbol1           Voley
10                  Futbol11          Voley1
10                  Futbol2           Voley2
20                  Futbol13          Voley3
20                  Futbol4           Voley4
40                  Futbol15          Voley5
50                  Futbol16          Voley6
60                  Futbol17          Voley7
80                  Futbol18          Voley8
90                  Futbol19          Voley9
100                 Futbol10          Voley10
CATEGORIAA          CATEGORIAB          CATEGORIAC
-----            -----            -----
110                 Futbol11          Voley11
120                 Futbol12          Voley12

13 filas seleccionadas.

```