

## Capítulo 6

# Lenguajes de Control de Datos y de Transacciones

---

Este capítulo está dedicado al control de los datos por parte de los usuarios como de la finalización con éxito o no de las transacciones contra la base de datos. Los usuarios son los propietarios de sus objetos pero pueden conceder o revocar privilegios sobre ellos a otros usuarios. Por otro lado, podemos hacer permanente o no los cambios realizados en una base de datos por parte de las transacciones que se ejecutan en ella.

### 6.1. El lenguaje de control de datos

El Lenguaje de Control de Datos<sup>1</sup> se usa para controlar la concesión o no de privilegios del sistema a los usuarios y a las funciones y, también, para la concesión de los privilegios de manipulación de datos sobre los objetos propiedad de los usuarios. Lo usa tanto el administrador de la base de datos, para conceder privilegios del sistema, como los usuarios, para dar o eliminar privilegios de uso sobre los objetos definidos en su esquema. Tiene dos órdenes que son:

- **grant** concede privilegios del sistema o sobre objetos a usuarios y/o a funciones.
- **revoke** elimina los privilegios concedidos del sistema o sobre objetos a usuarios y/o a funciones.

---

<sup>1</sup>DCL

## 6.2. La orden grant

Un usuario puede conceder privilegios de manipulación sobre cualquier tipo de objeto que posea a cualquier usuario. Para ello utilizará la orden del lenguaje de control `grant`.

### Formato:

```
GRANT {priv_objeto|all [privileges]}
      [(nom_columna [,nom_columna]...)]
      [, {priv_objeto|all [privileges]}
        [(nom_columna [,nom_columna]...)]]...
ON [esquema.]nom_objeto
TO {esquema|función|public}
   [, {usuario|función|public}]...
[with grant option];
```

*priv\_objeto* es el nombre de un privilegio que se concede a un objeto. Los privilegios sobre los objetos son:

```
alter   execute   index   references
select  insert    update  delete
```

`all privileges` otorga todos los privilegios al objeto indicado.

*nom\_columna* es el nombre de la columna de la tabla o vista sobre la que vamos a dar privilegios a otros usuarios.

`on` indica sobre qué objeto vamos a dar privilegios. Podemos dar privilegios sobre los siguientes objetos:

Tabla	Vista	Secuencia
Procedimiento	Instantánea	Sinónimo

`to` indica a qué usuario o función se le va a conceder los privilegios.

**función** es el nombre de una función o papel<sup>2</sup> a la que se le va a conceder privilegios sobre los objetos<sup>3</sup>.

`public` concede los privilegios a todos los usuarios.

`with grant option` permite que el receptor del privilegio se lo asigne a otros usuarios o a funciones. El receptor debe ser un usuario o `public` mejor que una función.

---

<sup>2</sup>Oracle los denomina `role`.

<sup>3</sup>Ver capítulo 8.

Privilegio	Acción
<code>alter</code>	Permite que el usuario o función pueda cambiar la definición de una tabla.
<code>delete</code>	Permite borrar los registros de una tabla.
<code>index</code>	Permite crear índices sobre una tabla.
<code>insert</code>	Permite insertar registros en una tabla.
<code>references</code>	Permite crear una restricción que referencia a una tabla.
<code>select</code>	Permite consultar una tabla.
<code>update</code>	Permite actualizar los datos de una tabla.

Tabla 6.1: Privilegios sobre una tabla.

Privilegio	Acción
<code>delete</code>	Permite borrar los registros desde una vista.
<code>insert</code>	Permite insertar registros desde una vista.
<code>select</code>	Permite consultar una tabla desde una vista.
<code>update</code>	Permite actualizar los datos de una tabla desde una vista.

Tabla 6.2: Privilegios sobre una vista.

Los privilegios que un usuario puede conceder con la orden `grant` van a depender del tipo de objeto. En las siguientes tablas 6.1, 6.2, 6.4, 6.5, 6.6 y 6.3 se especifican cuáles están permitidos y qué es lo que hacen.

El ejemplo 6.1 nos muestra cómo se concede el privilegio de actualizar los datos de la tabla *Suministros* al usuario *Exam3* y cómo hace uso de ese privilegio dicho usuario.

Privilegio	Acción
<code>all</code>	Los privilegios disponibles sobre los sinónimos son los mismos que los privilegios que tengan el objeto sobre el que se ha definido el sinónimo. Además, es indiferente usar el nombre del objeto o su sinónimo para cualquier orden de <i>SQL</i> .

Tabla 6.3: Privilegios sobre sinónimos.

Privilegio	Acción
alter	Permite cambiar la definición de una secuencia.
select	Permite examinar e incrementar valores de una secuencia utilizando las pseudocolumnas <code>currval</code> y <code>nextval</code> .

Tabla 6.4: Privilegios sobre una secuencia.

Privilegio	Acción
execute	Permite ejecutar el procedimiento o función o acceder a cualquier objeto declarado del programa en la especificación de un paquete.

Tabla 6.5: Privilegios sobre procedimientos, funciones y paquetes.

---

**Ejemplo 6.1** Concesión de modificación sobre una tabla


---

```
SQL> grant update on gadeschi.suministros
2 to exam3;
```

Concesion terminada correctamente.

```
SQL> conn exam3@juno
Enter password:
Connected.
```

```
SQL> update gadeschi.suministros
2 set smt_prv=0
3 where smt_art=15;
```

1 fila actualizada.

---

### 6.3. La orden revoke

La orden `revoke` retira o elimina los privilegios sobre un objeto determinado a usuarios o funciones. Si se retira los privilegios de un usuario, ya no podrá ejecutar las operaciones permitidas por el privilegio sobre el objeto. Si se le retira un privilegio a una función, ningún usuario al que se le asigne dicha función podrá ejecutar las operaciones, a menos

Privilegio	Acción
select	Permite consultar instantáneas.

Tabla 6.6: Privilegios sobre instantáneas.

que le sean permitidas por otra función. Si retiramos privilegios de `public`, ningún usuario que haya recibido el privilegio a través de `public` podrá ejecutar las operaciones permitidas por el privilegio de objeto.

**Formato:**

```
REVOKE {priv_objeto|all [privileges]}
      [, {priv_objeto|all [privileges]]}
ON [esquema.]objeto
FROM {esquema|función|public}[, {esquema|función|public}]
cascade constraints;
```

`on` indica a qué objeto le vamos a quitar los privilegios.

`from` identifica al usuario o función al que se le va a revocar los privilegios indicados.

`cascade constraints` borra cualquier restricción de integridad referencial que haya sido definida usando el privilegio `references` sobre el objeto que se está eliminando.

**Ejemplo 6.2**

Eliminación de privilegios

```
SQL> revoke update on suministros
2 from exam3 cascade constraints;
```

Revocacion terminada correctamente.

En el ejemplo 6.2 hemos revocado el privilegio dado al usuario *Exam3* de poder actualizar valores de la tabla *Suministros*, ver ejemplo 6.1.

## 6.4. Resumen de privilegios de usuarios

- Antes de permitir cualquier operación con la base de datos, Oracle comprueba los permisos del sistema concedidos al identificativo recién introducido<sup>4</sup>.
- Cuando queremos realizar una operación de manipulación de datos, primero comprobamos si tenemos concedidos los permisos adecuados sobre ese objeto.
- Retirar un privilegio de sistema o sobre un objeto a un usuario equivale a retirarlo a todas aquellas personas que lo recibieron de este último.
- Se pueden conceder todos los tipos de permisos sobre nuestros objetos a un usuario con una simple sentencia por medio de la cláusula `all`.

<sup>4</sup>Ver capítulo 1.

- Los privilegios pueden concederse de una sola vez a todos los usuarios por medio de la cláusula `public`.
- La forma correcta de concesión de permisos a otros usuarios debería ser realizada con la cláusula `with grant option`.

## 6.5. Órdenes de control de transacciones

Cuando utilizamos las órdenes de manipulación de datos `insert`, `update` o `delete` en una base de datos, se pueden anular o dar marcha atrás al trabajo hecho. Esto puede ser muy importante cuando se descubre un error.

Por el contrario, si utilizamos estas órdenes y queremos que las acciones se guarden en la base de datos (escritura en disco), debemos realizar un proceso de validación de esos datos obtenidos como consecuencia de usar las órdenes del lenguaje de manipulación de datos<sup>5</sup>.

Las órdenes de *SQL* `commit` y `rollback` son las encargadas de realizar estas acciones opuestas.

### 6.5.1. La orden `commit`

Esta orden se utiliza para validar inmediatamente una acción realizada por una orden de manipulación de datos, como puede ser introducir un registro, actualizarlo o borrarlo. Validará todos los cambios realizados en la base de datos desde el último `commit` ejecutado, bien de forma explícita o implícita.

#### Formato:

```
COMMIT [work];
```

`work` es opcional y `commit` hará lo mismo si lo ponemos o no, sólo es a efecto de cumplir con el *SQL* estándar.

El producto *SQL\*Plus* tiene la facilidad de validar automáticamente su trabajo sin tener que indicarlo explícitamente. Esto se controla con la orden `set auto[commit]`<sup>6</sup>. Si queremos saber el valor de esta variable del sistema utilizaremos la orden `show`<sup>7</sup>.

Cuando nosotros estamos trabajando con nuestros objetos y realizamos cambios, estos cambios son guardados en memoria principal y otro usuario con permiso sobre nuestros

---

<sup>5</sup>DML

<sup>6</sup>Ver la tabla 1.9, del capítulo 1.

<sup>7</sup>Ver capítulo 1.

objetos no verá los cambios realizados hasta que nosotros hagamos una validación explícita o implícita.

### 6.5.2. La orden `rollback`

Es posible realizar un gran número de inserciones, actualizaciones y borrados de nuestros datos en una sesión de trabajo y después no validar y, por tanto, deshacer el trabajo. Esto se consigue con la orden `rollback`.

#### Formato:

```
ROLLBACK [work];
```

Si utilizamos esta orden, nos aparece el mensaje «`rollback` terminado». Esto puede ser algo confuso y significa exclusivamente que se ha dado marcha atrás a una transacción que no ha sido validada. Si validamos una serie de transacciones bien explícitamente con la orden `commit` o bien implícitamente con cualquier otra acción, el mensaje de «`rollback` terminado» no significa nada realmente.

### 6.5.3. La orden `commit` implícito

Las acciones que forzarán que se ejecute un `commit`, sin ni siquiera indicarlo son:

<code>alter</code>	<code>audit</code>	<code>connect</code>	<code>create table</code>
<code>create view</code>	<code>disconnect</code>	<code>drop table</code>	<code>drop view</code>
<code>exit</code>	<code>grant</code>	<code>noaudit</code>	<code>revoke</code>

Cuando nosotros ejecutamos cualquiera de estas órdenes, a su término se realiza un `commit` implícito.

### 6.5.4. La orden `rollback` implícito

Si se ha realizado una serie de operaciones con los datos de una base de datos pero aún no se han validado explícita o implícitamente, y se presentan serias dificultades, tales como un fallo del ordenador, Oracle realizará automáticamente un `rollback` sobre cualquier trabajo no validado. Si la máquina o la base de datos se «viene abajo», esto conllevará la eliminación del trabajo efectuado para la próxima vez que la base de datos se ponga en funcionamiento, o sea cuando se vuelva a levantar la *instance*.

En los ejemplos 6.3 y 6.4 se muestra el uso de las órdenes de control de transacciones. Primero actualizamos el nombre del proveedor nº 9. Comprobamos que se ha modificado el nombre y damos la orden `rollback` para que no se haga permanente el cambio (ejemplo

6.3). En el ejemplo 6.4 volvemos a actualizar el nombre del proveedor pero esta vez sí que permitimos que la actualización se vea reflejada en la base de datos.

**Ejemplo 6.3**

Utilización de las órdenes de control de transacciones

```
SQL> select *
  2  from proveedores;
```

```
PRV_NUM PRV_NOM
-----
      1 catio electronic
      2 estilograficas reunidas
      3 mecanica de precision
      4 sanjita
      5 electrolamp
      6 copisteria
      7 el corte ingles
      8 rusian
      9 audi
```

9 filas seleccionadas.

```
SQL> update proveedores
  2  set prv_nom='mercedes'
  3  where prv_num=9;
```

1 fila actualizada.

```
SQL> select *
  2  from proveedores;
```

```
PRV_NUM PRV_NOM
-----
      1 catio electronic
      2 estilograficas reunidas
      3 mecanica de precision
      4 sanjita
      5 electrolamp
      6 copisteria
      7 el corte ingles
      8 rusian
      9 mercedes
```

9 filas seleccionadas.

```
SQL> rollback;
```

Rollback terminado.

---

**Ejemplo 6.4**

Utilización de las órdenes de control de transacciones (cont.)

```
SQL> select *
      2  from proveedores;
```

```
PRV_NUM PRV_NOM
-----
      1 catio electronic
      2 estilograficas reunidas
      3 mecanica de precision
      4 sanjita
      5 electrolamp
      6 copisteria
      7 el corte ingles
      8 rusian
      9 audi
```

9 filas seleccionadas.

```
SQL> update proveedores
      2  set prv_nom='mercedes'
      3  where prv_num=9;
```

1 fila actualizada.

```
SQL> commit;
```

Validacion terminada.

```
SQL> select *
      2  from proveedores;
pulsa una tecla
```

```
PRV_NUM PRV_NOM
-----
      1 catio electronic
      2 estilograficas reunidas
      3 mecanica de precision
      4 sanjita
      5 electrolamp
      6 copisteria
      7 el corte ingles
      8 rusian
      9 mercedes
```

9 filas seleccionadas.

---