

Material Adicional para la Implementación de Triggers

Disparador (old y new)

Cuando trabajamos con trigger a nivel de fila, Oracle provee de dos tablas temporales a las cuales se puede acceder, que contienen los antiguos y nuevos valores de los campos del registro afectado por la sentencia que disparó el trigger. El nuevo valor es ":new" y el viejo valor es ":old". Para referirnos a ellos debemos especificar su campo separado por un punto ":new.CAMPO" y ":old.CAMPO".

El acceso a estos campos depende del evento del disparador.

En un trigger disparado por un "insert", se puede acceder al campo ":new" unicamente, el campo ":old" contiene "null".

En una inserción se puede emplear ":new" para escribir nuevos valores en las columnas de la tabla.

En un trigger que se dispara con "update", se puede acceder a ambos campos. En una actualización, se pueden comparar los valores de ":new" y ":old".

En un trigger de borrado, unicamente se puede acceder al campo "old", ya que el campo ":new" no existe luego que el registro es eliminado, el campo ":new" contiene "null" y no puede ser modificado.

Los valores de "old" y "new" están disponibles en triggers after y before.

El valor de ":new" puede modificarse en un trigger before, es decir, se puede acceder a los nuevos valores antes que se ingresen en la tabla y cambiar los valores asignando a ":new.CAMPO" otro valor.

El valor de ":new" NO puede modificarse en un trigger after, esto es porque el trigger se activa luego que los valores de "new" se almacenaron en la tabla.

El campo ":old" nunca se modifica, sólo puede leerse.

Pueden usarse en una cláusula "when" (que veremos posteriormente).

En el cuerpo el trigger, los campos "old" y "new" deben estar precedidos por ":" (dos puntos), pero si está en "when" no.

Veamos un ejemplo.

Creamos un trigger a nivel de fila que se dispara "antes" que se ejecute un "update" sobre el campo "precio" de la tabla "libros". En el cuerpo del disparador se debe ingresar en la

tabla "control", el nombre del usuario que realizó la actualización, la fecha, el código del libro que ha sido modificado, el precio anterior y el nuevo:

```
create or replace trigger tr_actualizar_precio_libros  
before update of precio  
on libros  
for each row  
begin  
  insert into control values(user,sysdate,:new.codigo,:old.precio,:new.precio);  
end tr_actualizar_precio_libros;  
/
```

Cuando el trigger se dispare, antes de ingresar los valores a la tabla, almacenará en "control", además del nombre del usuario y la fecha, el precio anterior del libro y el nuevo valor.

El siguiente trigger debe controlar el precio que se está actualizando, si supera los 50 pesos, se debe redondear tal valor a entero, hacia abajo (empleando "floor"), es decir, se modifica el valor ingresado accediendo a ":new.precio" asignándole otro valor:

```
create or replace trigger tr_actualizar_precio_libros  
before update of precio  
on libros  
for each row  
begin  
  if (:new.precio>50) then  
    :new.precio:=floor(:new.precio);  
  end if;  
  insert into control values(user,sysdate,:new.codigo,:old.precio,:new.precio);  
end tr_actualizar_precio_libros;  
/
```

Si al actualizar el precio de un libro colocamos un valor superior a 50, con decimales, tal valor se redondea al entero más cercano hacia abajo. Por ejemplo, si el nuevo valor es "54.99", se almacenará "54".

Podemos crear un disparador para múltiples eventos, que se dispare al ejecutar "insert", "update" y "delete" sobre "libros". En el cuerpo del trigger se realiza la siguiente acción: se almacena el nombre del usuario, la fecha y los antiguos y viejos valores del campo "precio":

```
create or replace trigger tr_libros  
before insert or update or delete  
on libros  
for each row  
begin  
  insert into control values(user,sysdate,:old.codigo,:old.precio,:new.precio);  
end tr_libros;
```

end tr_libros;
/

Si ingresamos un registro, el campo ":old.codigo" y el campo ":old.precio" contendrán "null". Si realizamos una actualización del campo de un campo que no sea "precio", los campos ":old.precio" y ":new.precio" guardarán el mismo valor.

Si eliminamos un registro, el campo ":new.precio" contendrá "null".

Entonces, se puede acceder a los valores de ":new" y ":old" en disparadores a nivel de fila (no en disparadores a nivel de sentencia). Están disponibles en triggers after y before. Los valores de ":old" solamente pueden leerse, nunca modificarse. Los valores de ":new" pueden modificarse únicamente en triggers before (nunca en triggers after).

Para controlar el proceso transaccional de acuerdo a cada situación en solo trigger

```
CREATE OR REPLACE TRIGGER carga_control
AFTER UPDATE OR INSERT OR DELETE ON cuentaxcobrar
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO aud_cxc (lista de valores) VALUES (lista de valores, user, sysdate);
  ELSIF UPDATING THEN
    INSERT INTO aud_cxc (lista de valores) VALUES (lista de valores, user, sysdate);
  ELSIF DELETING THEN
    INSERT INTO aud_cxc (lista de valores) VALUES (lista de valores, user, sysdate);
  END IF;
END IF;
END carga_control; /
```