

TRIGGERS o DISPARADORES de Bases de Datos

Los triggers o disparadores de bases de datos son bloques PL/SQL almacenados asociados a una tabla que se ejecutan o disparan automáticamente cuando se producen ciertos eventos o sucesos que afectan a la tabla (inserción, borrado o modificación de filas)

Se suelen utilizar para:

- Implementar restricciones de seguridad o integridad
- Prevenir transacciones erróneas
- Implementar reglas administrativas complejas
- Generar automáticamente valores derivados
- Auditar y controlar actualizaciones

Elementos de un trigger

```
CREATE [OR REPLACE]
TRIGGER nombrettrigger
{ BEFORE | AFTER }
{ DELETE | INSERT | UPDATE [OF <lista_columnas>] }
ON nombrettabla
[FOR EACH {STATEMENT | ROW}]
< CUERPO DEL TRIGGER (BLOQUE PL/SQL) >
```

En los disparadores podemos distinguir los siguientes elementos:

- Nombre del trigger

CREATE OR REPLACE TRIGGER Nombre_Del_Trigger

- Tipo del trigger

- Por un lado hace referencia al momento en que se ejecute:

- antes (BEFORE) o
- después (AFTER)

de que se ejecute la orden DML

- Por otro, hace referencia a si el trigger se activa una sola vez para cada orden o si se activa una vez por cada fila afectada al ejecutarse la orden

- A nivel de orden: por defecto, FOR EACH STATEMENT, no es necesario, se asume por omisión. Se ejecuta una única vez por orden, independientemente de a cuantas filas afecta

TEMA 2.2: Acceso a la Información: Disparadores de BBDD

- A nivel de fila: **FOR EACH ROW**, el trigger se activará una vez por cada fila afectada por la orden.
- Evento de disparo
 - Es el suceso que producirá la ejecución del trigger
 - Será siempre una orden DML: **INSERT, DELETE, UPDATE** o UPDATE OF <lista de columnas>
 - Se pueden especificar varios eventos de disparo para un mismo trigger utilizando la cláusula OR
- Nombre de la tabla sobre la que actúa
ON Nombre_De_Tabla
- Cuerpo del trigger
Es el código que se ejecutará cuando se cumplan las condiciones específicas de la cabecera. Se trata de un bloque PL/SQL

EJEMPLOS

```
CREATE OR REPLACE TRIGGER ACCION_DEPART
BEFORE INSERT OR UPDATE OR DELETE
ON DEPART
BEGIN
IF TO_CHAR(SYSDATE,'HH24')<=7 THEN
  RAISE_APPLICATION_ERROR(-20001, 'NO SON HORAS DE MODIFICACIONES');
ELSE
  MOSTRAR('SE VA A PRODUCIR UN CAMBIO EN LA TABLA DEPART');
END IF;
END;
```

Valores :NEW y :OLD

Se puede hacer referencia a los valores anterior y posterior de una columna cuando ésta se está actualizando.

Esto solo puede hacerse en un trigger a nivel de fila (FOR EACH ROW)

Lo haremos como :old.nombre_columna y :new.nombre_columna

TEMA 2.2: Acceso a la Información: Disparadores de BBDD

- Cuando el evento que dispara el trigger es DELETE, debemos hacer referencia a :old.nombre_columna, ya que el valor new no existe
- Cuando el evento que dispara el trigger es INSERT, debemos hacer referencia a :new.nombre_columna, ya que el valor old no existe
- Para los triggers cuyo evento de disparo es UPDATE, tienen sentido los dos valores

EJEMPLOS

```
CREATE OR REPLACE TRIGGER NO_BAJO_SALARIO
BEFORE UPDATE
ON EMPLE
FOR EACH ROW
BEGIN
IF :NEW.SALARIO < :OLD.SALARIO THEN
:NEW.SALARIO := :OLD.SALARIO;
VER ('EL SALARIO DE ' || :OLD.APELLIDO || ' NO SE VA A MODIFICAR');
END IF;
END;
```

Ejemplo:

Vamos a crear una tabla para llevar una auditoría de las modificaciones que se realizan al salario de los empleados de EMPLE

```
CREATE TABLE AUDITA_EMPLE
(EMP_NO NUMBER,
SALARIO NUMBER,
FECHA DATE);
```

```
CREATE OR REPLACE TRIGGER audit_subida_salario
AFTER UPDATE OF salario
ON emple
FOR EACH ROW
BEGIN
INSERT INTO audita_emple
VALUES (:OLD.emp_no, :NEW.SALARIO, SYSDATE );
END;
```

TEMA 2.2: Acceso a la Información: Disparadores de BBDD

Cada vez que se modifique el salario de un empleado quedará grabada una fila en la tabla `audita_emple` con el número del empleado, el nuevo salario y la fecha de la modificación.

Tipos de disparadores y orden de ejecución

Una misma tabla puede tener varios disparadores. Al ejecutarse una actualización (INSERT, UPDATE, DELETE), lo que ocurre es lo siguiente:

1º Se ejecutan los triggers `BEFORE... FOR EACH STATEMENT`

Después por cada fila afectada

2º Se ejecutan los disparadores `BEFORE... FOR EACH ROW`

3º Se ejecuta la actualización de la fila (INSERT, UPDATE, DELETE). En este momento se bloquea la fila hasta que la transacción se confirme

4º Se ejecutan los disparadores `AFTER... FOR EACH ROW`

5º Se ejecutan los triggers `AFTER... FOR EACH STATEMENT`

Observacion:

- Si un trigger falla, Oracle da por fallida la actualización completa produciéndose `ROLLBACK` de toda la actualización

Múltiples eventos de disparo

Un mismo trigger puede ser disparado por distintas operaciones o eventos de disparo.

Para indicarlo, se utilizará el operador `OR`. Por ejemplo

`BEFORE DELETE OR UPDATE`

`ON EMPLE`

En estos casos es posible que las acciones del bloque PL/SQL dependan del tipo de evento que disparó el trigger. Podemos preguntar por el evento que desencadenó el trigger de la siguiente manera:

TEMA 2.2: Acceso a la Información: Disparadores de BBDD

INSERTING	Devuelve TRUE si el evento que disparó el trigger fue un comando INSERT
DELETING	Devuelve TRUE si el evento que disparó el trigger fue un comando DELETE
UPDATING	Devuelve TRUE si el evento que disparó el trigger fue un comando UPDATE
UPDATING ('columna')	Devuelve TRUE si el evento que disparó el trigger fue un comando UPDATE y la columna especificada ha sido actualizada

Podremos después preguntar en el código del trigger

IF INSERTING THEN

IF DELETING THEN

IF UPDATING THEN

IF UPDATING ('SALARIO') THEN

Restricciones para la creación de triggers

El código PL/SQL del cuerpo del trigger puede contener instrucciones de consulta y de manipulación de datos, así como llamadas a funciones y procedimientos.

Existen, sin embargo, algunas restricciones:

- No pueden contener las sentencias COMMIT ni ROLLBACK
- Tampoco se puede llamar a subprogramas que transgredan la restricción anterior
- En triggers FOR EACH ROW, no se pueden consultar ni modificar tablas mutantes, que son aquellas que están siendo modificadas por una sentencia UPDATE, DELETE o INSERT en la misma sesión, principalmente la tabla afectada por el trigger.