

# EJERCICIO RESUELTO

## Módulo: BASES DE DATOS

### Disparadores

#### Descripción:

Sobre el caso práctico de esta base de datos donde una librería almacena los datos de sus libros en una tabla denominada "LIBROS" y controla las actualizaciones del precio de los libros almacenando en la tabla "CONTROL" el nombre del usuario, la fecha, el precio anterior y el nuevo, crear los diferentes disparadores que se solicitan.

LIBROS		Control	
Codigo	Number (6)	Usuario	Varchar2 (30)
Titulo	Varchar2 (40)	Fecha	date
Autor	Varchar2 (30)	Codigo	Number (6)
Editorial	Varchar2 (30)	PrecioAnterior	Number (6,2)
Precio	Number (6,2)	PrecioNuevo	Number (6,2)

- Creación de un disparador llamado "**Actualizar\_Precio\_Libros**" 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 insertará 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.
- Modificar el disparador anteriormente creado para que controle el precio que se está actualizando, ya que solo se quiere controlar el cambio que supere los 100 euros debido a una oferta en la librería.
- Modificar el disparador anteriormente creado para que controle que si insertamos o updatamos en la tabla libros, inserte en la tabla control, pero si borramos en la tabla libros no haga nada en la tabla control.

#### Objetivos:

- Identificar la sintaxis de creación de disparadores.
- Comprobar los diferentes métodos para implementar disparadores: predicados condicionales, cualificadores y basados en condición.

**Recursos:**

- Acceso a Internet.
- Software ofimático.
- Oracle.



## Resolución:

Creación de un disparador llamado "**Actualizar\_Precio\_Libros**" 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 insertará 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.

```
--Declaramos el disparador Actualizar_Precio_Libros
CREATE OR REPLACE TRIGGER 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 Actualizar_Precio_Libros;
```

```
--Para emitir el disparador, lanzamos un update
UPDATE libros SET precio=40 WHERE codigo=100;
```

```
--Para verificar que se ha insertado el registro en la tabla control
SELECT * FROM control;
```

Modificar el disparador anteriormente creado para que controle el precio que se está actualizando, ya que solo se quiere controlar el cambio que supere los 100 euros debido a una oferta en la librería.

```
--Declaramos el disparador Actualizar_Precio_Libros
CREATE OR REPLACE TRIGGER Actualizar_Precio_Libros

BEFORE UPDATE OF precio ON libros
FOR EACH ROW

--Verificamos que el nuevo precio supera los 100 euros
WHEN (NEW.precio>100)
BEGIN
    INSERT INTO control
    VALUES(USER,SYSDATE,NEW.codigo,OLD.precio,NEW.precio);
END Actualizar_Precio_Libros;
```

```
--Para emitir el disparador, lanzamos un update
UPDATE libros SET precio=137 WHERE codigo=120;
```

```
--Para verificar que se ha insertado el registro en la tabla control
SELECT * FROM control;
```

Modificar el disparador anteriormente creado para que controle que si insertamos o updatamos en la tabla libros, inserte en la tabla control, pero si borramos en la tabla libros no haga nada en la tabla control.

--Declaramos el disparador Actualizar\_Precio\_Libros

CREATE OR REPLACE TRIGGER Actualizar\_Precio\_Libros

BEFORE UPDATE OR INSERT OR DELETE OF precio ON libros  
FOR EACH ROW

BEGIN

IF (UPDATING ('precio') OR INSERTING) THEN

INSERT INTO control

VALUES(USER,SYSDATE,

:NEW.codigo,:OLD.precio,:NEW.precio);

ELSE

DBMS\_OUTPUT.PUT\_LINE ('No ha borrado nada');

END IF;

END Actualizar\_Precio\_Libros;

--Para emitir el disparador, lanzamos una sentencia

INSERT INTO libros VALUES(101,'La vuelta al mundo en 80 días',  
'Julio Verne','Planeta',80);

UPDATE libros SET precio=88 WHERE codigo=101;

--Para emitir el disparador, lanzamos una sentencia y esta no borrará nada

DELETE libros WHERE codigo=101;

--Para verificar la tabla control

SELECT \* FROM control;