

## DESENCADENADORES (TRIGGERS)

Un desencadenador es un tipo especial de procedimientos almacenados que entra en vigor cuando se modifican datos en una tabla especificada utilizando una o más operaciones de modificación de datos: UPDATE (actualización), INSERT (inserción) o DELETE (eliminación). Los desencadenadores pueden consultar otras tablas e incluir instrucciones SQL complejas. Son especialmente útiles para exigir reglas o requisitos complejos. Por ejemplo, se puede controlar que se permita insertar un pedido basándose en el estado de cuenta actual de un cliente.

Los desencadenadores también son útiles para exigir la integridad referencial, que conserva las relaciones definidas entre tablas cuando se agregan, actualizan o eliminan filas en esas tablas. No obstante, la mejor manera de exigir la integridad referencial es definiendo restricciones de clave principal y de clave externa en las tablas relacionadas. Si se utilizan diagramas de bases de datos, se puede crear relaciones entre tablas para crear automáticamente una restricción de clave externa.

### Ventajas de utilizar desencadenadores

Los desencadenadores tienen varias utilidades:

- Los desencadenadores son automáticos: se activan inmediatamente después de que se efectúen modificaciones en los datos de la tabla, como una entrada manual o una acción de la aplicación.
- Pueden realizar cambios en cascada a través de tablas relacionadas de la base de datos.
- Los desencadenadores pueden hacer referencia a columnas de otras tablas. Por ejemplo, un desencadenador puede deshacer actualizaciones que intenten aplicar descuentos (almacenados en la tabla discounts) a libros (almacenados en la tabla titles) con un precio inferior a \$10.

*Las tablas **INSERTED** (se genera con las instrucciones INSERT y UPDATE) y **DELETED** (se genera con las instrucciones UPDATE, DELETE) hacen referencia, al interior de un Trigger, a la tabla en mención y/o modificada.*

### Creación:

Sintaxis (Simplificada):

```
CREATE TRIGGER <nombre_trigger> ON <nombre_tabla>  
INSTEAD OF [FOR [INSERT,DELETE,UPDATE]  
AS  
<conjunto_Instrucciones>
```

Donde:

**FOR:** Después de la actualización de la (s) tablas

**INSTEAD OF:** Antes de la actualización de la (s) tablas.

**Modificación:**

Sintaxis:

```
ALTER TRIGGER <nombre_trigger> ON <nombre_tabla>  
INSTEAD OF |FOR [INSERT,DELETE,UPDATE]  
AS  
<conjunto_instrucciones>
```

**Eliminación:**

Sintaxis:

```
DROP TRIGGER <nombre_desencadenador>
```

**Activar/Desactivar Desencadenador en una tabla:**

Sintaxis:

```
ALTER TABLE <nombre_tabla>  
ENABLE|DISABLE <nombre_desencadenador>
```

**Nota:**

Para mostrar los desencadenadores de una tabla, se da la siguiente instrucción, desde el analizador de consultas, por ejemplo:

**sp\_helptrigger conceptos**

Para mostrar el texto de un trigger, se da la siguiente instrucción, desde el analizador de consultas, por ejemplo:

**sp\_helptext novoconcepto**

**Ejemplos:**

```
CREATE TRIGGER nuevoconcepto ON conceptos  
FOR INSERT  
AS  
Print "Se adicionó un Concepto"
```

```
CREATE TRIGGER actconcepto ON conceptos  
FOR update  
AS  
Print "Se modificó un concepto"
```

```
CREATE TRIGGER borrarconcepto ON conceptos  
FOR delete  
AS  
Print "Se eliminó un concepto"
```

```
CREATE TRIGGER adicconceptos2 ON Conceptos  
FOR INSERT  
AS  
print "se adicionó tambien en conceptos1"  
insert into conceptos1 select codigoc,descripcionc from inserted
```

```
CREATE TRIGGER BorrarConcepto ON Conceptos
FOR DELETE
AS
delete from conceptos1
where conceptos1.codigoc in (select codigoc from deleted)
```

```
CREATE TRIGGER actualizaconcepto ON Conceptos
FOR UPDATE
AS
update conceptos1 set descripcionc = (select descripcionc from inserted)
where conceptos1.codigoc in (select codigoc from inserted)
```

## PRÁCTICA DE DESENCADENADORES

Para este taller tenga en cuenta la base de datos INVENTARIO.

1. Qué es un desencadenador?
2. Dónde se almacenan los desencadenadores?
- 3.Cuál es el objetivo de los desencadenadores?
4. Cuáles son las acciones para que un desencadenador se ejecute?
5. Qué es INSERTED y DELETED en un desencadenador?
6. En qué momento se crea INSERTED y DELETED?
7. Difiera entre procedimientos almacenados y desencadenadores
8. Qué hace SP\_HELPTRIGGER?

Crear desencadenadores para:

9. Al agregar un PROVEEDOR, también se agregue en la tabla PROVEEDORC
10. Al actualizar un PROVEEDOR, también se actualice en la tabla PROVEEDORC
11. Al borrar los datos de un PROVEEDOR, también se borre en la tabla PROVEEDORC.
12. Al realizar una compra, en detalle de compra, se incremente la existencia de ese producto, con base en la cantidad.
13. Al realizar una compra, en detalle de compra, el precio unitario del producto se incremente en un 10% más del precio unitario de la compra.
14. Al realizar una venta, en detalle venta, se disminuya la existencia de ese producto, respecto a la cantidad.
15. Al borrar una compra se eliminen todos sus detalles de compra.
16. Al borrar una venta, se eliminen todos sus detalles de venta.
17. Al borrar un cliente, se borren todas sus transacciones (ventas y detalles de ventas).
18. Al borrar un proveedor, se borren todas transacciones (compras y detalles de compras).
19. Desactive el desencadenador generado en el numeral anterior.
20. Elimine los desencadenadores generados en esta práctica.