



# BASE DE DATOS PARA APLICACIONES



# I. Manipulación Avanzada

Objetivo:

El alumno optimizará base de datos para mejorar el rendimiento de aplicaciones de software.

# CONTENIDO

## Disparadores

- Concepto.
- Sintaxis para la • Inserción • Borrado • Actualización
- Ejemplos

# ¿Qué es un Disparador?

Un trigger o un disparador en una BD es un evento que se ejecuta automáticamente cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

(DESENCADENADORES,  
DISPARADORES)

---

Las palabras reservadas DELETE, INSERT y UPDATE corresponden a cada una de las acciones para las cuales se puede definir un desencadenador dentro de la tabla especificada. El bloque de sentencias permite prácticamente cualquier tipo de ellas dentro del lenguaje T-SQL, pero con ciertas limitaciones. Por ejemplo, no se podrá utilizar la sentencia SELECT, ya que un TRIGGER no puede devolver datos al usuario, sino que simplemente se ejecuta para cambiar o comprobar los datos que se van a insertar, actualizar o borrar.

los TRIGGERS en SQL Server tienen una serie de limitaciones:

1. No se puede disparar un TRIGGER dentro de otro TRIGGER, ya que daría lugar a un bucle infinito
2. Por esta razón, un TRIGGER no puede ejecutar instrucciones DDL (lenguaje de definición de datos)
3. No se pueden ejecutar sentencias como SELECT INTO o de creación de dispositivos dentro de un TRIGGER

## Sintaxis para borrar un disparador

- Del mismo modo, para borrar un TRIGGER, deberemos ejecutar la sentencia `DROP TRIGGER TRIGGER`.

Por ejemplo, si queremos borrar el TRIGGER anteriormente creado, ejecutaremos `DROP TRIGGER modificacion_Clientes`

Las tablas `DELETED` e `INSERTED` Dentro de la definición de un TRIGGER, podemos hacer referencia a un par de tablas lógicas, cuya estructura es similar a la tabla donde se está ejecutando el TRIGGER; es decir, es una copia de la tabla en la cual se van a insertar o borrar los datos, y que contiene, precisamente, los datos que van a ser añadidos o borrados. La utilidad de estas dos tablas es la de realizar comprobaciones entre los datos antiguos y los nuevos. Así, por ejemplo, si queremos recuperar los datos de la tabla que estamos borrando, dentro del TRIGGER, se deberá ejecutar el siguiente código: `SELECT * FROM deleted`

## Tipos de desencadenadores

- SQL-Server permite la definición de varios tipos de TRIGGERS, entre los cuales cabe destacar los siguientes:
  - Desencadenadores múltiples: para una misma tabla, se pueden definir distintos TRIGGERS para la misma acción, es decir, si definimos un TRIGGER para insert, y resulta que dicha tabla ya tenía definido un TRIGGER para esa misma acción, se ejecutarán ambos TRIGGERS cuando ocurra dicho evento sobre la tabla.
  - Desencadenadores recursivos: se permite la recursividad entre las llamadas a los TRIGGERS, es decir, un TRIGGER puede llamar a su vez a otro, bien de forma directa, bien de forma indirecta



## Tipos de desencadenadores

- Desencadenadores anidados: si un TRIGGER cambia una tabla en la que se encuentra definido otro TRIGGER, se provoca la llamada de este último que, si a su vez vuelve a modificar otra tabla, puede provocar la ejecución de otro TRIGGER, y así sucesivamente. Si se supera el nivel de anidamiento permitido, se cancelará la ejecución de los TRIGGERS.

# Consideraciones acerca del uso de desencadenadores

Los desencadenadores son reactivos, mientras que las restricciones son proactivas

Las restricciones se comprueban antes

Las tablas pueden tener varios desencadenadores para cualquier acción

Comparación del estado de los datos antes y después de su modificación

# Referencias Bibliográficas

<http://www.sqlserverya.com.ar/temarios/descripcion.php?cod=147&punto=141>

<http://www.devjoker.com/contenidos/catss/278/Triggers-en-Transact-SQL.aspx>

# PREGUNTAS Y COMENTARIOS

