

INVESTIGACIÓN ACTIVIDAD 29

¿Qué son los disparadores (triggers) en SQL?

Un disparador o trigger en SQL es un bloque de código que se ejecuta automáticamente en respuesta a ciertos eventos que ocurren en una tabla o vista de la base de datos.

Estos eventos pueden ser:

INSERT (cuando se inserta un nuevo registro)

UPDATE (cuando se actualiza un registro)

DELETE (cuando se elimina un registro)

El trigger se ejecuta sin necesidad de que lo llame un usuario o aplicación, lo hace el propio sistema de la base de datos.

¿Para qué sirven los disparadores?

Los disparadores son útiles para:

Validar datos antes de que se inserten o modifiquen.

Mantener la integridad de los datos entre varias tablas.

Registrar acciones (crear logs o auditorías) cuando se realizan cambios.

Evitar errores o acciones no deseadas, como eliminar información crítica.

Actualizar automáticamente otras tablas relacionadas.

Ventajas y desventajas

Ventajas:

Automatización: Ejecutan acciones sin intervención del usuario.

Consistencia: Aseguran que ciertas reglas o procesos siempre se cumplan.

Auditoría: Se pueden usar para llevar un registro de cambios.

Reutilización de código: Se programa una vez y se ejecuta siempre que se cumpla la condición.

Desventajas:

Complejidad: Pueden dificultar la lectura y el mantenimiento de la base de datos.

Difíciles de depurar: Si algo falla, puede no estar claro por qué ocurrió.

Impacto en el rendimiento: Si el trigger hace muchas operaciones, puede ralentizar los procesos.

Dependencia oculta: A veces los triggers hacen cambios que no son evidentes para los desarrolladores.

¿Cómo se usan los disparadores?

Se crean con el comando CREATE TRIGGER. Se define cuándo y dónde se ejecuta el trigger (antes o después de una acción, y en qué tabla).

Sintaxis básica / template

sql

Copy

Edit

```
CREATE TRIGGER nombre_del_trigger
{BEFORE | AFTER} {INSERT | UPDATE | DELETE}
ON nombre_de_la_tabla
FOR EACH ROW
BEGIN
    -- Código SQL que se ejecutará
END;
```

Explicación:

CREATE TRIGGER: crea el disparador.

BEFORE o AFTER: indica si el trigger se ejecuta antes o después del evento.

INSERT, UPDATE, DELETE: el tipo de evento que lo activa.

ON nombre_de_la_tabla: en qué tabla se aplica.

FOR EACH ROW: se ejecuta por cada fila afectada.

BEGIN ... END: bloque de código que contiene la acción que se va a realizar.

Ejemplo explicado

Supongamos que queremos registrar en una tabla de auditoría cada vez que se agregue un nuevo cliente:

1. Crear tabla de auditoría:

sql

Copy

Edit

```
CREATE TABLE auditoria_clientes (
    id_auditoria INT AUTO_INCREMENT PRIMARY KEY,
    nombre_cliente VARCHAR(100),
    fecha_insertado DATETIME
);
```

2. Crear trigger:

sql

Copy

Edit

```
CREATE TRIGGER registrar_cliente
```

```
AFTER INSERT ON clientes
FOR EACH ROW
BEGIN
    INSERT INTO auditoria_clientes (nombre_cliente, fecha_insertado)
    VALUES (NEW.nombre, NOW());
END;
```

¿Qué hace este trigger?

Cada vez que se inserta un nuevo cliente en la tabla clientes, el trigger se activa automáticamente y guarda el nombre del cliente y la fecha en la tabla auditoria_clientes.