

Triggers

Un **trigger** (disparador) en SQL es un conjunto de instrucciones que se ejecuta automáticamente en respuesta a un evento específico en una tabla o vista. Este evento puede ser una operación de **INSERT**, **UPDATE** o **DELETE**.

Sintaxis de un trigger

```
CREATE TRIGGER nombre_trigger
{ BEFORE | AFTER } { INSERT | UPDATE | DELETE }
ON nombre_tabla
FOR EACH ROW
BEGIN
    -- Lógica del trigger
END;
```

Componentes de la sintaxis

1. **CREATE TRIGGER :**
 - Declara que estás creando un trigger.
2. **nombre_trigger :**
 - Es el nombre único del trigger en la base de datos.
3. **{ BEFORE | AFTER } :**
 - Especifica cuándo debe ejecutarse el trigger:
 - **BEFORE:** Antes de que ocurra la operación (se puede usar para validar datos o modificar valores antes de que se inserten o actualicen).
 - **AFTER:** Después de que la operación haya ocurrido (se usa típicamente para registrar cambios o realizar acciones posteriores).
4. **{ INSERT | UPDATE | DELETE } :**
 - Indica el tipo de operación que activará el trigger.
5. **ON nombre_tabla :**
 - Especifica la tabla en la que se debe activar el trigger.
6. **FOR EACH ROW :**

- Indica que el trigger se ejecutará una vez por cada fila afectada por la operación.

7. **BEGIN ... END :**

- Contiene la lógica del trigger, que puede incluir validaciones, cálculos, o inserciones en otras tablas.

Acceso a los datos en un trigger

Dentro del trigger, puedes usar:

- **NEW.columna** : Para acceder al valor de la columna después de la operación (en caso de **INSERT** o **UPDATE**).
- **OLD.columna** : Para acceder al valor de la columna antes de la operación (en caso de **UPDATE** o **DELETE**).

Ejemplos

Imagina que tienes una tabla `cliente` y deseas registrar los cambios realizados en una tabla `historial_cambios`.

```
DELIMITER //
```

```
CREATE TRIGGER TriggerActualizarCliente  
AFTER UPDATE ON cliente  
FOR EACH ROW  
BEGIN  
    -- Registro del cambio de nombre del cliente  
    IF OLD.nombre_cliente = NEW.nombre_cliente THEN  
        INSERT INTO historial_cambios (id_cliente, columna_modificada, valor_anterior,  
        valor_nuevo, fecha_cambio)  
        VALUES (NEW.id_cliente, 'nombre_cliente', OLD.nombre_cliente,  
        NEW.nombre_cliente, NOW());  
    END IF;  
END //
```

```
DELIMITER ;
```

Ejemplo 2.

Imagina que no quieres permitir registros en una tabla con un monto negativo.

```
DELIMITER //
```



```
CREATE TRIGGER TriggerValidarTransaccion  
BEFORE INSERT ON transacciones  
FOR EACH ROW  
BEGIN  
    IF NEW.monto_compra < 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'El monto de la transacción no puede ser negativo.';  
    END IF;  
END //
```



```
DELIMITER ;
```