

# Triggers

# Índice

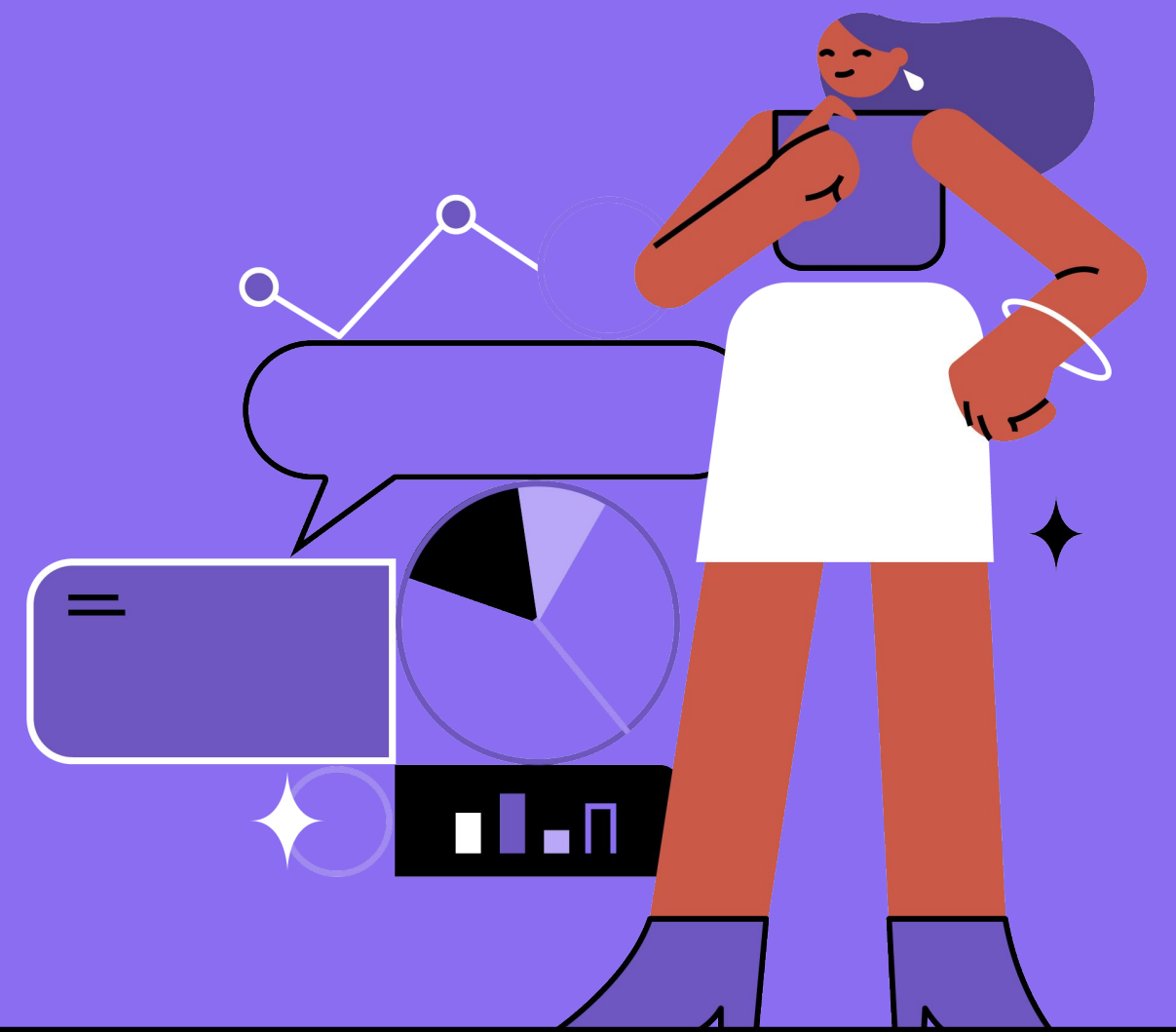
- 01 [Definición](#)
- 02 [Creación](#)
- 03 [Eliminación](#)
- 04 [Cuándo usarlos](#)



01

# Definición

Los triggers **son objetos de base de datos** que nos sirven para ejecutar código SQL luego de que se haya ejecutado una sentencia de Insert, Update o Delete sobre una tabla específica.

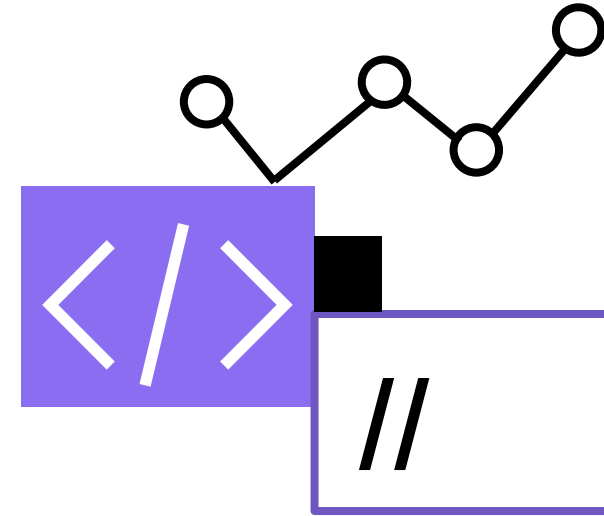


- Un trigger siempre está asociado a una tabla.
- Los triggers no se pueden definir para tablas temporales como tampoco para vistas.



02

# Creación



# Creación

SQL

```
CREATE TRIGGER nombreTrigger  
tiempoTrigger eventoTrigger  
ON NombreTabla FOR EACH ROW  
ordenTrigger (opcional)  
cuerpoTrigger
```

# Creación

- **tiempoTrigger:** pueden ser los valores **BEFORE** o **AFTER**. Esto es para definir cuándo se ejecuta el trigger. MySQL permite ejecutar el código ANTES de una operación sobre la tabla o DESPUÉS de la operación sobre la tabla.
- **eventoTrigger:** pueden ser los valores **INSERT**, **UPDATE** o **DELETE** . Este es el evento que se ejecuta sobre nuestra tabla que va a disparar el trigger.
- **ordenTrigger:** pueden ser los valores **FOLLOWS** o **PRECEDES** nombreTrigger. En caso de que tengamos varios triggers en nuestra tabla, tenemos la opción de agregar el orden de ejecución de los mismos.
- **cuerpoTrigger:** cuerpo del trigger. Si queremos ejecutar varias sentencias, podemos agregar el bloque **BEGIN...END**



# Creación



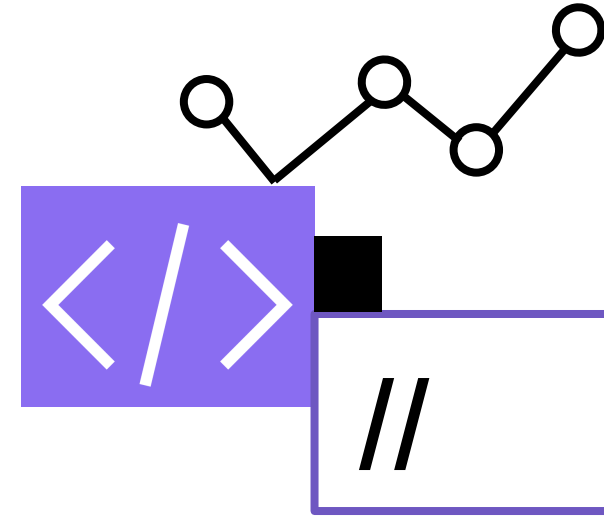
Dentro del cuerpo del trigger, podemos hacer referencia a el objeto **NEW** y **OLD**.

**Old** hace referencia al registro que existe en la tabla y va a ser modificado o eliminado.

**New** hace referencia al nuevo registro que va a ser insertado o el nuevo registro que fue modificado.

03

# Eliminación



# Eliminación de triggers

Para eliminar un trigger, ya que es un objeto de la base de datos, utilizamos la sentencia **DROP**.

SQL

```
DROP TRIGGER nombreTrigger;
```

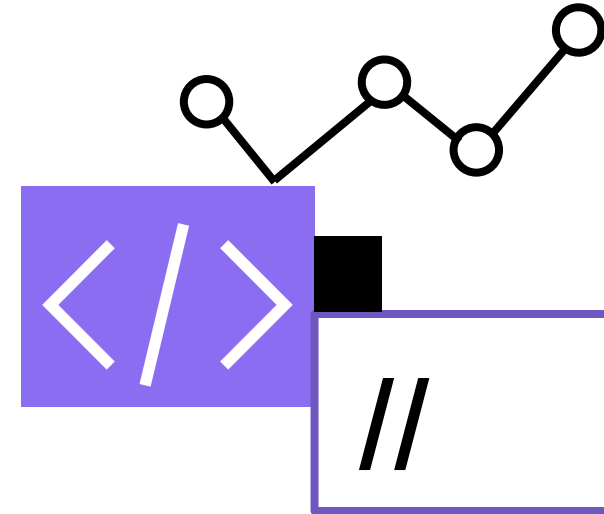
04

# Cuándo usarlos

# Cuándo usarlos

Los triggers los podemos usar para:

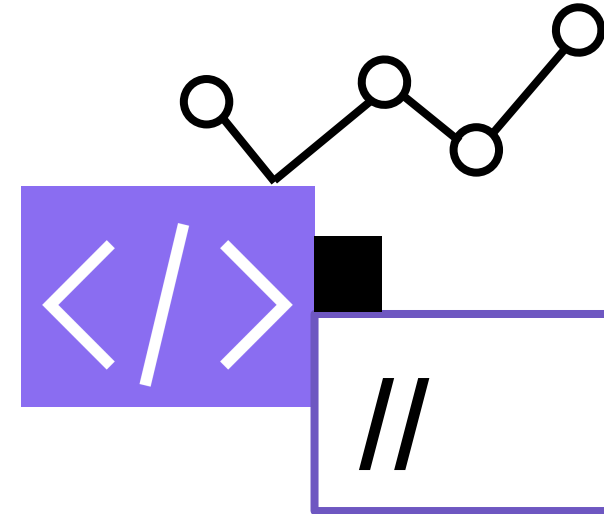
- Modificar datos
- Reglas complejas
- Auditoría



# Ejemplo 1

```
USE ADVENTUREWORKS;  
  
DELIMITER $$  
CREATE TRIGGER trigger1  
BEFORE INSERT ON address FOR EACH ROW  
BEGIN  
    SET NEW.stateProvinceId = '80';  
END;$$
```

Por cada **Insert** en la tabla address, **se ejecutará el código del trigger ANTES de realizar la inserción**, y modificará el valor de stateProvinceId a 80. Por lo tanto, si consultamos sobre la tabla address, nuestro último registro tendrá el valor 80 en stateProvinceId



## Ejemplo 2

```
USE ADVENTUREWORKS;

DELIMITER $$
CREATE TRIGGER trigger1
BEFORE UPDATE ON address FOR EACH ROW
BEGIN
    SET NEW.stateProvinceId = '70';
    SET NEW.modifiedDate = now();
END;$$
```

Por cada **Update** en la tabla address, se ejecutará el código del trigger **ANTES** de realizar la actualización, y modificará el valor de stateProvinceId a 70. Además, modificará el valor de la columna ModifiedDate por el valor que genere la función now() al momento de ejecutarse.

Por lo tanto, si consultamos sobre la tabla address, nuestro último registro luego de una actualización tendrá el valor 70 en stateProvinceId y la fecha que lo ejecutamos en el campo ModifiedDate.

¡Muchas gracias!