

# Triggers



# Introducción

---

Un trigger o disparador es un objeto que se almacena en la base de datos y que se ejecuta cuando sucede un evento determinado en la tabla a la que está asociado.



# ¿Cuándo se ejecutan los Trigger?

---

Los eventos que hacen que se ejecute un trigger son las operaciones de inserción (**INSERT**), borrado (**DELETE**) o actualización (**UPDATE**), ya que modifican los datos de una tabla.

Se pueden ejecutar antes (**BEFORE**) o después (**AFTER**) de que se produzca la operación.

# Características

---

Algunas de las características de los Triggers son las siguientes:

- **Automatización:** No requieren que un usuario los ejecute.
- **Prevención:** Permite la prevención de errores de datos realizando las comprobaciones necesarias en la ejecución de los triggers.
- **Sincronización:** Se podrían tener tablas sincronizadas programando triggers que realicen esta tarea.
- **Control:** Permite un mayor control de la base de datos.

# Sintaxis

---

La sintaxis para crear un Trigger es la siguiente:

```
CREATE TRIGGER trigger_name trigger_tiempo trigger_evento  
ON tabla FOR EACH ROW trigger_body;
```

trigger\_tiempo: { BEFORE | AFTER }

trigger\_evento: { INSERT | UPDATE | DELETE }

\* Para poder crearlos en la base de datos hay que hacer uso de delimitadores como en las Funciones y Procedimientos.

# OLD y NEW

---

Cuando se ejecuta un Trigger se puede obtener el valor del registro a insertar o eliminar y, en el caso de las actualizaciones, el valor del registro a actualizar.

Para ello se utilizan las palabras OLD y NEW:

- **NEW:** Permite conocer el valor de los datos a insertar o actualizar.
- **OLD:** Permite conocer el valor actual del registro antes de actualizar o eliminar.

# Ejemplo after + insert

Creamos un Trigger que actualizará el stock después de una venta:

```
CREATE TRIGGER despues_de_insertar AFTER INSERT
```

```
ON ventas FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE productos SET pro_unidades=pro_unidades-1
```

```
    WHERE NEW.ven_codpro=pro_codpro;
```

```
END
```

---

Se ejecutará al realizar: `INSERT INTO ventas VALUES ('CODIGO','DESC_PRODUCTO');`

# Ejemplo before + update

---

Creamos un Trigger que guardará la nota anterior del alumno antes de modificarlo:

```
CREATE TRIGGER antes_de_actualizar BEFORE UPDATE
```

```
ON alumnos FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO notas_anteriores(noa_dni, noa_nota)  
    VALUES (OLD.alu_dni,OLD.alu_nota);
```

---

```
END
```