¿Qué es un TRIGGER?

Un **Trigger**, también llamado **Disparador**, en una base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida.

Es una especie de script de programación SQL para bases de datos. **Los TRIGGER son procedimientos** que se ejecutarán según nuestras indicaciones cuando se realicen operaciones sobre la información de la base de datos. Estas operaciones pueden ser de actualización (UPDATE), inserción (INSERT) y borrado (DELETE).

¿Para qué sirve un TRIGGER?

Los TRIGGERS son una de las funcionalidades más útiles que disponemos cuando se implementan y se mantienen bases de datos. Gracias a ellos se puede implementar ciertas características de nuestra base de datos sin necesidad de desarrollar programación en otros lenguajes externos.

Los TRIGGERS pueden modificar la información de la base de datos e incluso detener la ejecución de consultas erróneas.

Imaginemos por ejemplo que se quiere duplicar en una segunda tabla toda la información que se inserte en otra. No existe ninguna forma de indicarle esto a la base de datos que lo realice de forma automática. Sin embargo, gracias a un TRIGGER que se ejecute tras una sentencia INSERT, podemos insertar esa información en la segunda tabla, todo esto sin que el usuario/programador que lanzó el INSERT tenga que hacer nada.

La utilidad que se le acostumbra mayoritariamente a dar, es para prevenir errores de datos, actualizar tablas, modificar valores, entre muchas utilidades que el administrador le quiera dar.

Cómo crear TRIGGER en SQL

Crear un TRIGGER en SQL es sencillo, casi todos los sistemas de bases de datos (SGBD) tienen el soporte para su manejo, e incluso algunos pueden incorporar un asistente que nos guie en el proceso.

Sintaxis general:

```
CREATE TRIGGER trigger_name
[AFTER | BEFORE] [INSERT | UPDATE | DELETE] ON table
FOR EACH ROW
BEGIN
    - Líneas de código SQL que ejecutará el trigger
END;
```

Donde *trigger name*, es el nombre que servirá después para identificar el trigger, así como para buscarlo y borrarlo de la base de datos.

A continuación debemos indicar **CUANDO DEBE EJECUTARSE el TRIGGER**. Las indicaciones posibles en este punto son AFTER o BEFORE para indicar si el disparador se ejecutará ANTES o DESPUÉS de la orden lanzada por el usuario. Estas órdenes serán: INSERT, DELETE o UPDATE. Por último indicaremos sobre que tabla actuará.

Por último mediante el comando BEGIN y END indicamos las líneas de código SQL que ejecutará el trigger.

Ejemplo:

```
CREATE TRIGGER trigger_historico
AFTER INSERT ON usuario
FOR EACH ROW
BEGIN
    - Líneas de código SQL que ejecutará el trigger
END;
```

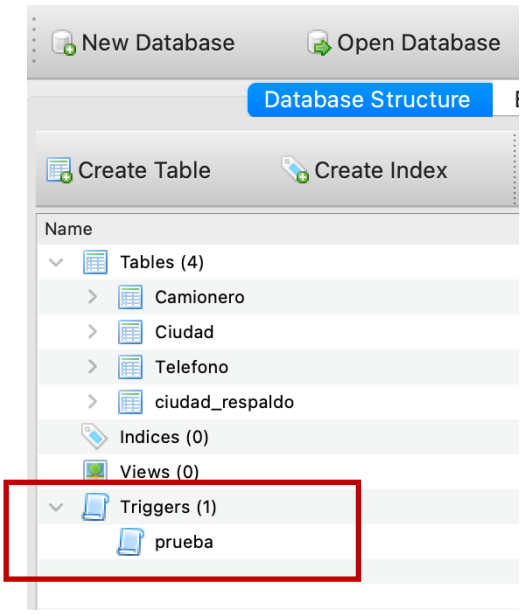
En este ejemplo el trigger se llama *trigger_histórico*, se especifica que la ejecución será tras una inserción de la tabla *usuario* y que se debe aplicar a cada fila insertada. Y entre el *begin* y el *end* se especificará el cóodigo a realizar.

Ejemplo 1 en SQLite, utilizando la BD *CamionerosTrigger.db* publicada en didacTIC

En la ventana de ejecutar sentencias de SQL poner el siguiente código y ejecutar la siguiente sentencia

```
CREATE TRIGGER prueba
AFTER INSERT on Camionero
for EACH ROW
BEGIN
    Insert into Telefono values("PruebaTrigger", "345");
END
```

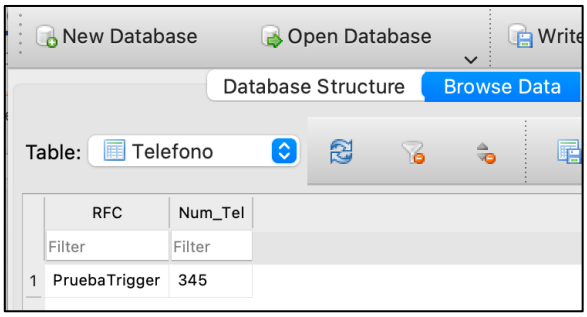
Observe que se ha creado el trigger *prueba* en la estructura de la BD y cada que se inserte un registro en la tabla *Camionero* se ejecutará este trigger de forma automática insertando un registro en la tabla *Telefono* con los valores especificados.



Ahora para probar el trigger debemos insertar un registro en la tabla *Camionero*, por ejemplo

```
INSERT INTO Camionero VALUES ("ABC", "maria", "calle", 456.4, 45, "correo");
```

Verifique que al insertar el registro en la tabla *Camionero* también se hizo la inserción definida en el trigger en la tabla *Telefono*.



Para que las líneas de código SQL puedan acceder a la información que interviene en el TRIGGER, es decir, la información a la que afecta un UPDATE, INSERT o DELETE, se dispone de dos variables especiales NEW y OLD.

La variable NEW de un TRIGGER

NEW es la variable que almacena la nueva información que aporta la consulta a la base de datos, es decir, cada una de las filas que intervienen en un INSERT O UPDATE. Si por ejemplo se ha realizado un INSERT, gracias a NEW podremos acceder a los datos introducidos para cada columna de la tabla. *NEW.nombre* por ejemplo almacena la información de la columna nombre que tendrá el nuevo registro insertado en la tabla.

Hay que tener en cuenta que NEW no estará disponible en todos los tipos de TRIGGER. En concreto los TRIGGER relacionados con un DELETE no dispondrán de información en esta variable ya que tan solo tendremos información antigua que es eliminada.

La variable OLD de un TRIGGER

OLD es la variable que almacena la información antigua relacionada con la consulta que ejecuta el TRIGGER, esta información la componen las filas que van a ser borradas o modificadas. En un DELETE por ejemplo, OLD tendrá la información de todas las columnas de los registros borrados.

Al igual que NEW no está disponible para todos los TRIGGER que creemos, la variable OLD no podrá ser utilizada para un INSERT, ya que en este tan solo existe nueva información que va a ser insertada en la base de datos.

Ejemplo de un TRIGGER sencillo

Vamos a resolver un TRIGGER que nos permita mantener una copia de todos los clientes que se inserten en una base de datos de una tienda online. Para esto tendremos dos tablas: cliente y cliente_historico. Así, el siguiente TRIGGER insertará toda la información del cliente (nombre, dni, direccion) más una columna extra: fecha_registro.

```
CREATE TRIGGER trigger_cliente_historico
AFTER INSERT ON cliente
FOR EACH ROW
BEGIN
        INSERT INTO cliente_historico(nombre, dni, direccion, fecha)
        VALUES(NEW.nombre, NEW.dni, NEW.direccion, CURDATE());
END;
```

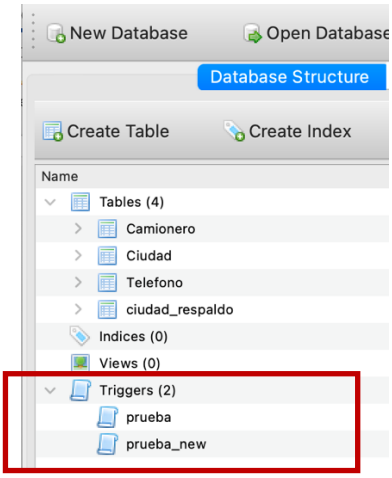
La función CURDATE() regresa la fecha actual.

Ejemplo 2 en SQLite, utilizando la BD *CamionerosTrigger.db* publicada en didacTIC

En la ventana de ejecutar sentencias de SQL poner el siguiente código y ejecutar la siguiente sentencia, observe el uso de la variable NEW

```
CREATE TRIGGER prueba_new
AFTER INSERT on Ciudad
for EACH ROW
BEGIN
        INSERT INTO ciudad_respaldo VALUES(NEW.CP, NEW.Nombre);
END
```

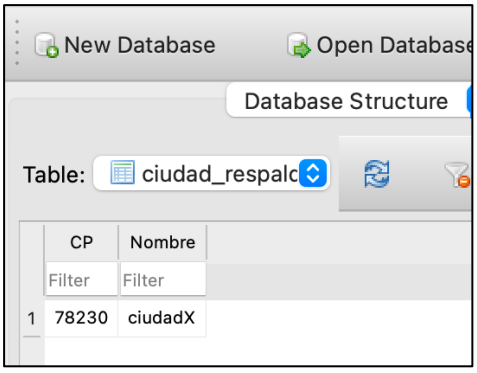
Verifique la creación del trigger visualizando la estructura de la BD



Ahora para probar el trigger debemos insertar un registro en la tabla *Ciudad*, por ejemplo

```
INSERT INTO Ciudad VALUES(78230, "ciudadX");
```

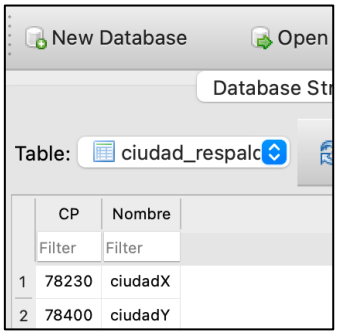
Verifique que al insertar el registro en la tabla *Ciudad* también se hizo la inserción definida en el trigger en la tabla *ciudad_respaldo* con los datos obtenidos de la variable NEW.



Por último, inserte otro registro

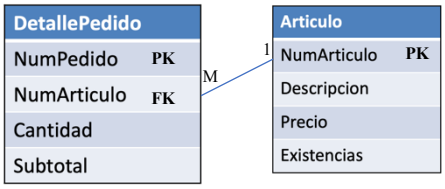
INSERT INTO Ciudad VALUES(78400, "ciudadY");

Y verifique el contenido de la tabla *ciudad_respaldo*



Algunos ejemplos clásicos de triggers son los siguientes

Ejemplo 1



Cuando se inserta un registro en la tabla DetallePedido
DetallePedido(NumPedido, NumArticulo, Cantidad, Subtotal)

Acción del disparador:
Se actualiza la columna *Existencias* de la tabla *Articulo*
(Existencias = Existencias – Cantidad)

Ejemplo 2

Cuando se inserta una tupla del detalle de pedido
DetallePedido(NumPedido, NumArticulo, Cantidad, Subtotal)

Acción del disparador:
Se actualiza la columna *Subtotal* de la misma tabla, haciendo una consulta del *Precio* según el número del artículo de la tabla *Artículo*
(Subtotal = Cantidad * Precio)

Ejemplo 3

Cuando se inserta un registro en la tabla *Pedido*
Pedido(NumPedido, Fecha, NumDireccion, Total)

Acción del disparador:
Se actualiza el campo *Fecha* de la misma tabla con la fecha del sistema.

Ejemplo 4

Cuando se inserta un registro en la tabla *Alumno*
Alumno(ClaveUnica, Nombres, ApellidoP, ApellidoM, FechaNac, Edad)

Acción del disparador:
Se actualiza el campo *Edad* de la misma tabla.
(Edad = FechaActual - FechaNac)

Referencias bibliográficas

https://www.srcodigofuente.es/aprender-sql/triggers-sql

Material de apoyo, presentación Disparadores(Triggers). M.E.M Claudia Alicia Méndez Hernández

Material de apoyo, presentación Disparadores_Ejemplos (Triggers). M.E.M Claudia Alicia Méndez Hernández

http://www.w3big.com/es/sqlite/sqlite-trigger.html

https://sqlite.org/lang_createtrigger.html