Trigger

-An SQL trigger allows you to specify SQL actions that should be executed automatically when a specific event occurs in the database.

- SQL triggers are generally associated with a particular table.

This means that when the table is deleted, all its associated triggers are deleted accordingly. Given a table

-an SQL trigger can be invoked before or after the following events:

* INSERT: a new row is inserted in the table.
* UPDATE: an existing row of the table gets updated.
* DELETE: a row in the table gets deleted.

- There are two types of triggers: **row-level triggers** and **statement-level triggers**.

### **Row-Level Triggers**

A row-level trigger is executed once for each row affected by the triggering event, which is typically an INSERT, UPDATE, or DELETE statement.

- Statement-level triggers are useful to perform an action based on the overall effect of an INSERT, UPDATE, or DELETE statement, rather than on individual rows.

- SQL triggers can be used to automatically enforce business rules at the database level. For example, a trigger can be used to ensure that the price of a product is never set to less than its cost + 10%.

- A trigger is automatically called whenever a data modification event against a table takes place, which is the main distinction between a trigger and a procedure. On the other hand, a stored procedure must be called directly

- Triggers cannot be manually invoked or executed.

There is no chance that triggers will receive parameters.

A transaction cannot be committed or rolled back inside a trigger

*create*

- *syntax*

*create trigger [trigger\_name]*

*[before | after]*

*{insert | update | delete}*

*on [table\_name]*

*[for each row]*

*[trigger\_body]*

*-example*

A screenshot of a computer

Description automatically generated

After any update on table set the date with today date