# Triggers

- A trigger is a stored procedure that is associated with a particular table
- It  is defined to activate for INSERT, DELETE, or UPDATE statements for that table.
- A trigger can be set to activate either before or after each row processed by the statement.
- The trigger definition includes a statement that executes when the trigger activates.
- A trigger can examine or change new data values to be inserted or used to update a row.
- This enables you to enforce data integrity constraints, such as verifying that a
- percentage is a value from 0 to 100.
- It also makes it possible to perform input data filtering.
- A trigger can supply default values for a column based on an expression.
- A trigger can examine the current contents of a row before it is deleted or updated.
  - This capability can be exploited to perform logging of changes to existing rows, for example.

- **Row Level Triggers**
  - FOR EACH ROW
  - Can access Original and New column values processed by the SQL statement.
  - Fired every time the table is affected by the SQL statement.
  - Access column values as follows
    - :**new**.column-name [only for INSERT/UPDATE]
    - :**old**.column-name [for UPDATE/DELETE]

# Triggers-Before or After ?

- When defining a trigger, timing of trigger can be specified
  - BEFORE INSERT
  - AFTER INSERT
  - BEFORE UPDATE
  - AFTER UPDATE
  - BEFORE DELETE
  - AFTER DELETE

**BEFORE**

indicates that trigger before executing the triggering statement.

**AFTER**

indicates that the trigger after executing the triggering statement.

**DELETE**

indicates that the trigger whenever a DELETE statement removes a row from the table.

**INSERT**

indicates that the trigger whenever an INSERT statement adds a row to table.