12-09-2025 TRIGGERS

You have a table Employees with columns EmployeeID, Name, and Salary.

```
CREATE TABLE Employees (
   EmployeeID INT PRIMARY KEY,
   Name VARCHAR(100),
   Salary DECIMAL(10, 2)
);

CREATE TABLE AuditLog (
   LogID INT IDENTITY(1,1) PRIMARY KEY,
   Action VARCHAR(50),
   ActionDate DATETIME DEFAULT GETDATE(),
   Details VARCHAR(255)
);
```

Exercises:

1. Create a trigger that automatically inserts a record into an AuditLog table whenever a new employee is added.
2. Write an AFTER INSERT trigger on Employees that inserts a row into AuditLog with the message "New employee added: [Employee Name]".
3. Modify the Employees table so that the salary cannot be updated to a value lower than the current salary. If an attempt is made to decrease the salary, roll back the update and raise an error. Create an INSTEAD OF UPDATE or BEFORE UPDATE trigger that prevents salary decreases.
4. When an employee is deleted from the Employees table, the deleted record should be saved into an EmployeesArchive table before removal.

```
CREATE TABLE EmployeesArchive (
   EmployeeID INT,
   Name VARCHAR(100),
   Salary DECIMAL(10, 2),
   DeletedDate DATETIME DEFAULT GETDATE()
);
```

1. Create an AFTER DELETE trigger on Employees that inserts the deleted employee’s data into EmployeesArchive.

2. Create a table called EmployeeCount with a single integer column Count. It stores the total number of employees. Write triggers to automatically update this count after inserts or deletes on Employees. Create the EmployeeCount table initialized to zero.Create triggers on Employees to increment or decrement Count appropriately.
3. Sometimes, multiple rows get inserted or updated at once. Create a trigger that logs all newly inserted employees into AuditLog with their names and salaries. Write an AFTER INSERT trigger on Employees that inserts one log entry per employee inserted, handling multi-row inserts correctly.