

Problem Statement 21 (Triggers)

Employee(emp_id, emp_name, salary, designation)

Salary_Backup(emp_id, old_salary, new_salary, salary_difference)

Create a Trigger to record salary change of the employee. Whenever salary is updated insert the details in

Salary_Backup table.

Create a Trigger that will prevent deleting the employee record having designation as CEO.

.....

-- Step 1: Create the Database

```
CREATE DATABASE EmployeeDB;
```

-- Step 2: Use the newly created database

```
USE EmployeeDB;
```

-- Step 3: Create the Employee table

```
CREATE TABLE Employee (  
    emp_id INT PRIMARY KEY,  
    emp_name VARCHAR(100),  
    salary DECIMAL(10, 2),  
    designation VARCHAR(50)  
);
```

-- Step 4: Create the Salary_Backup table

```
CREATE TABLE Salary_Backup (  
    emp_id INT,  
    old_salary DECIMAL(10, 2),  
    new_salary DECIMAL(10, 2),  
    salary_difference DECIMAL(10, 2),  
    FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)  
);
```

-- Step 5: Create the Trigger to record salary changes in the Salary_Backup table

```
CREATE TRIGGER after_salary_update  
AFTER UPDATE ON Employee  
FOR EACH ROW  
BEGIN  
    IF OLD.salary != NEW.salary THEN  
        INSERT INTO Salary_Backup (emp_id, old_salary, new_salary,  
salary_difference)  
        VALUES (OLD.emp_id, OLD.salary, NEW.salary, NEW.salary - OLD.salary);  
    END IF;  
END;
```

-- Step 6: Create the Trigger to prevent deleting an employee with designation 'CEO'

```
CREATE TRIGGER before_employee_delete
```

```
BEFORE DELETE ON Employee
FOR EACH ROW
BEGIN
    IF OLD.designation = 'CEO' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete an employee with
designation as CEO';
    END IF;
END;
```