DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment - 7.2

Name: Shivam Agarwal **UID : 23BCS13100**
Section: 23KRG-3B **Branch: BE-CSE**
Semester : 5th **Date of Performance : 28-Oct-2025**
Subject Name: ADBMS **Subject- Code : 23CSP - 333**

1. Aim:

HARD LEVEL PROBLEM:

Whenever a new employee is inserted in tbl_employee, a record should be added to tbl_employee_audit like:
"Employee name <emp_name> has been added at <current_time>"

Whenever an employee is deleted from tbl_employee, a record should be added to tbl_employee_audit like:
"Employee name <emp_name> has been deleted at <current_time>"

The solution must use PostgreSQL triggers.

2. Requirement :

- The solution must be implemented using PostgreSQL triggers.

3: Objective:

- To create a trigger function that automatically logs every insert and delete action performed on the tbl_employee table.
- To ensure that audit entries are recorded in tbl_employee_audit for tracking data changes.
- To include a timestamp (`current_time`) for each audit record to maintain a proper history of changes.
- To enhance data integrity, accountability, and traceability within the database system.

4: Code:

```
CREATE TABLE tbl_employee (
    emp_id SERIAL PRIMARY KEY,
    emp_name VARCHAR(100) NOT NULL,
    emp_salary NUMERIC
);

CREATE TABLE tbl_employee_audit (
    sno SERIAL PRIMARY KEY,
    message TEXT
);



CREATE OR REPLACE FUNCTION audit_employee_changes()
RETURNS TRIGGER
```

```
LANGUAGE plpgsql
AS
$$
BEGIN
   IF TG_OP = 'INSERT' THEN
      INSERT INTO tbl_employee_audit(message)
      VALUES ('Employee name ' || NEW.emp_name || ' has been added at ' || NOW());
      RETURN NEW;

   ELSIF TG_OP = 'DELETE' THEN
      INSERT INTO tbl_employee_audit(message)
      VALUES ('Employee name ' || OLD.emp_name || ' has been deleted at ' || NOW());
      RETURN OLD;
   END IF;

   RETURN NULL;
END;
$$



CREATE TRIGGER trg_employee_audit
AFTER INSERT OR DELETE
ON
tbl_employee
FOR EACH ROW
EXECUTE FUNCTION audit_employee_changes();



--TESTING THE TRIGGER
-- Insert an employee
```

```
INSERT INTO tbl_employee(emp_name, emp_salary) VALUES ('Aman', 50000);

-- Delete an employee
DELETE FROM tbl_employee WHERE emp_name = 'Aman';

-- Check audit log
SELECT * FROM tbl_employee_audit;
```

5: Output:

Output:

```
CREATE TABLE
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
INSERT 0 1
DELETE 1
 sno |                                message
-----+-----------------------------------------------------------------------
   1 | Employee name Aman has been added at 2025-10-30 15:31:04.709033+00
   2 | Employee name Aman has been deleted at 2025-10-30 15:31:04.712418+00
(2 rows)
```