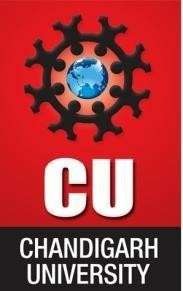
University Institute of Engineering

 Department of Computer Science & Engineering

**EXPERIMENT: 07**

**NAME: Abhay Mall UID: 23BCS13972**

**BRANCH: BE-CSE KRG\_2 SECTION / GROUP: A**

**SEMESTER: 5TH SUBJECT CODE: 23CSP-339**

# **SUBJECT NAME: ADBMS**

**1. Aim:**

1. Design a PostgreSQL trigger that performs the following task:

a. Whenever a new record is inserted into the student table, the inserted row

should be displayed on the output console.

b. Similarly, when a record is deleted from the student table, the deleted row should

also be displayed on the console.

2. Create PostgreSQL triggers to maintain an audit log for employee actions.

a. Whenever a new employee is inserted into tbl\_employee, a record should be

inserted into tbl\_employee\_audit with the message:"Employee name

<emp\_name> has been added at <current\_time>"

b. Whenever an employee is deleted from tbi\_employee, a record should be

inserted into tbl\_employee\_audit with the message:"Employee name

<emp\_name> has been deleted at <current\_time>" .

**2. Objective:**

• Maintain a complete and reliable record of all employee insertions and deletions for accountability and auditing purposes.

• Automatically insert descriptive audit messages into tbl\_employee\_audit whenever changes occur in tbl\_employee, without requiring manual input.

• Guarantee that every change in the employee table is consistently tracked in real-time, reducing the risk of unrecorded modifications.

• Store timestamps and employee names in the audit log to create a chronological history of employee activity for future reference and compliance checks.

• Increase visibility into employee-related database actions, supporting internal monitoring, troubleshooting, and security reviews.

3. Code:

1.

-- Create the student table

CREATE TABLE student (

id SERIAL PRIMARY KEY,

name VARCHAR(100),

age INT,

class VARCHAR(50)

);

-- Create the trigger function

CREATE OR REPLACE FUNCTION fn\_student\_audit()

RETURNS TRIGGER

LANGUAGE plpgsql

AS

$$

BEGIN

IF TG\_OP = 'INSERT' THEN

RAISE NOTICE 'Inserted Row -> ID: %, Name: %, Age: %, Class: %',

NEW.id, NEW.name, NEW.age, NEW.class;

RETURN NEW;

ELSIF TG\_OP = 'DELETE' THEN

RAISE NOTICE 'Deleted Row -> ID: %, Name: %, Age: %, Class: %',

OLD.id, OLD.name, OLD.age, OLD.class;

RETURN OLD;

END IF;

RETURN NULL;

END;

$$;

-- Create the trigger

CREATE TRIGGER trg\_student\_audit

AFTER INSERT OR DELETE

ON student

FOR EACH ROW

EXECUTE FUNCTION fn\_student\_audit();

-- Test the trigger

-- Insert records

INSERT INTO student(name, age, class) VALUES ('Shivanshu', 20, 'B.Tech');

INSERT INTO student(name, age, class) VALUES ('Tanya', 21, 'B.Tech');

INSERT INTO student(name, age, class) VALUES ('Devanshu', 19, 'Non-CSE');

-- Delete a record

DELETE FROM student WHERE name = 'Devanshu';

SELECT \* FROM student;

2.

-- Create employee and audit tables

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 the trigger function

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 the trigger

CREATE TRIGGER trg\_employee\_audit

AFTER INSERT OR DELETE

ON tbl\_employee

FOR EACH ROW

EXECUTE FUNCTION audit\_employee\_changes();

-- Test the trigger

-- Insert employees

INSERT INTO tbl\_employee(emp\_name, emp\_salary) VALUES ('Shivanshu', 90000);

INSERT INTO tbl\_employee(emp\_name, emp\_salary) VALUES ('Tanya', 95000);

INSERT INTO tbl\_employee(emp\_name, emp\_salary) VALUES ('Karan', 100000);

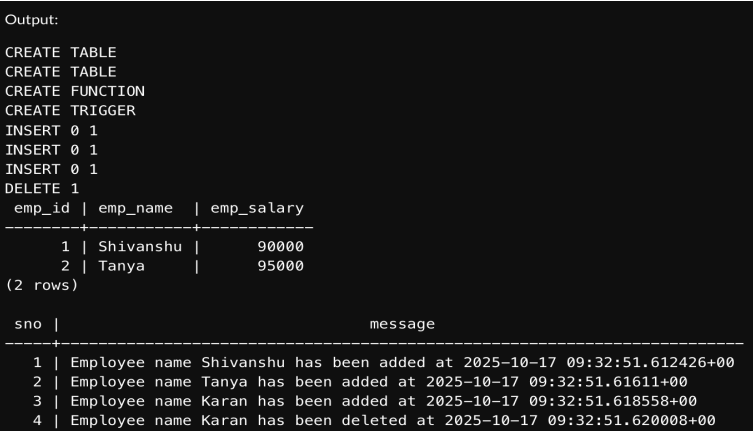
-- Delete one employee

DELETE FROM tbl\_employee WHERE emp\_name = 'Karan';

SELECT \* FROM tbl\_employee;

SELECT \* FROM tbl\_employee\_audit;

**4. Output:**

****