



Experiment 7

Student Name: Ankush

Branch: BE-CSE

Semester: 5th

Subject name: ADBMS

UID: 23BCS12742

Section/Group: KRG-3B

Date of performance: 28-10-2025

Subject code: 23CSP-333

AIM: design a trigger which:

1. Whenever there is a insertion or deletion on student table then, the currently inserted or deleted row should be printed on the console.

CODE:

-- create student table for testing

CREATE TABLE IF NOT EXISTS student

(id SERIAL PRIMARY KEY, name

TEXT NOT NULL, age INTEGER, class

TEXT

);

-- function for student audit (prints inserted/deleted row using RAISE NOTICE)

CREATE OR REPLACE FUNCTION fn_student_audit()

RETURNS TRIGGER

LANGUAGE plpgsql

AS

\$\$ BE

GIN

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;

$$;

-- trigger for student table

DROP TRIGGER IF EXISTS trg_student_audit ON student;

CREATE TRIGGER trg_student_audit

AFTER INSERT OR DELETE ON student

FOR EACH ROW

EXECUTE FUNCTION fn_student_audit();
```

```
INSERT 0 1
INSERT 0 1
DELETE 1
 sno | message
-----+-----
 1 | Employee name Jyoti has been added at 2025-10-30 07:17:39
 2 | Employee name Neha has been added at 2025-10-30 07:17:39
 3 | Employee name Jyoti has been deleted at 2025-10-30 07:17:39
(3 rows)
```

HARD LEVEL PROBLEM

Requirements: Design postgresql triggers that:

1. Whenever a new employee is inserted in tbl_employee, a record should be added to tbl_employee_audit:

"Employee name <emp_name> has been added at <current_time>"

2. Whenever an employee is deleted from tbl_employee, a record should be added to tbl_employee_audit:

"Employee name <emp_name> has been deleted at <current_time>"

CODE:

-- Create employee tables

CREATE TABLE IF NOT EXISTS tbl_employee

(emp_id SERIAL PRIMARY KEY, emp_name

VARCHAR(100) NOT NULL, emp_salary

NUMERIC

);

CREATE TABLE IF NOT EXISTS tbl_employee_audit

(sno SERIAL PRIMARY KEY, message

TEXT

);

-- function to insert audit messages into tbl_employee_audit

CREATE OR REPLACE FUNCTION audit_employee_changes()

RETURNS TRIGGER

LANGUAGE plpgsql

AS

\$\$ BE

GIN

```
IF TG_OP = 'INSERT' THEN
    INSERT INTO tbl_employee_audit(message)
    VALUES (
        'Employee name ' || NEW.emp_name || ' has been added at ' ||
        to_char(NOW(), 'YYYY-MM-DD HH24:MI:SS')
    );
    RETURN NEW;
ELSIF TG_OP = 'DELETE' THEN
    INSERT INTO tbl_employee_audit(message)
    VALUES (
        'Employee name ' || OLD.emp_name || ' has been deleted at ' ||
        to_char(NOW(), 'YYYY-MM-DD HH24:MI:SS')
    );
    RETURN OLD;
END IF;

RETURN NULL;

END;

$$;

-- trigger for tbl_employee
DROP TRIGGER IF EXISTS trg_employee_audit ON tbl_employee;
CREATE TRIGGER trg_employee_audit
AFTER INSERT OR DELETE ON tbl_employee
FOR EACH ROW
EXECUTE FUNCTION audit_employee_changes();
```

----- TESTING THE TRIGGERS -----

-- clear tables for a clean test (optional)

```
TRUNCATE TABLE tbl_employee_audit RESTART IDENTITY;
```

```
TRUNCATE TABLE tbl_employee RESTART IDENTITY;
```

```
TRUNCATE TABLE student RESTART IDENTITY;
```

-- Test student trigger (you should see RAISE NOTICE messages in the console)

```
INSERT INTO student(name, age, class) VALUES ('Aniket', 15, '10A');
```

```
INSERT INTO student(name, age, class) VALUES ('Tanmay', 16, '11B');
```

```
DELETE FROM student WHERE name = 'Aniket';
```

-- Test employee trigger

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

```
INSERT INTO tbl_employee(emp_name, emp_salary) VALUES ('Neha', 65000);
```

```
DELETE FROM tbl_employee WHERE emp_name = 'Jyoti';
```

-- Check audit logs

```
SELECT * FROM tbl_employee_audit ORDER BY sno;
```

```
psql:commands.sql:42: NOTICE: trigger "trg_student_audit" for relation "student" does not exist, skipping
psql:commands.sql:100: NOTICE: trigger "trg_employee_audit" for relation "tbl_employee" does not exist, skipping
psql:commands.sql:115: NOTICE: Inserted Row -> ID: 1, Name: Aniket, Age: 15, Class: 10A
psql:commands.sql:116: NOTICE: Inserted Row -> ID: 2, Name: Tanmay, Age: 16, Class: 11B
psql:commands.sql:117: NOTICE: Deleted Row -> ID: 1, Name: Aniket, Age: 15, Class: 10A
```