



Experiment 8

Student Name: Umang Kaushik

Branch: BE CSE

Semester: 5th

Subject Name: DBMS

UID: 23BCS10712

Section/Group: KRG-1-B

Date of Performance: 21th Oct

Subject Code: 23CSP-333

1. Problem Title: Triggers

2. Code and Output:

-----MEDIUM LEVEL PROBLEM-----

--REQUIREMENTS: DESIGN A TRIGGER WHICH:

--1. WHENEVER THERE IS A INSERTION ON STUDENT TABLE THEN,
THE CURRENTLY INSERTED OR DELETED

--ROW SHOULD BE PRINTED AS IT AS ON THE OUTPUT CONSOLE
WINDOW.

-- Re-create the table for this demo

CREATE TABLE student (

id SERIAL PRIMARY KEY,

name VARCHAR(100),

age INT,

class VARCHAR(20)

);



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

--SOLUTION

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;

\$\$; -- <--- FIXED: Added missing semicolon



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

-- Create the trigger to execute the function (Added for completeness)

```
CREATE TRIGGER trg_student_audit
```

```
AFTER INSERT OR DELETE
```

```
ON student
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION fn_student_audit();
```

-- Test the INSERT trigger

```
INSERT INTO student(name, age, class) VALUES ('Medium Test', 20, '12th');
```

-- Test the DELETE trigger

```
DELETE FROM student WHERE name = 'Medium Test';
```

-- Show that the table is empty again

```
SELECT * FROM STUDENT;
```

-----HARD LEVEL PROBLEM-----

/*

Requirements: DESIGN A PORSTGRESQL TRIGGERS THAT:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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.

*/

--THERE IS INSERT IN EMPLOYEE TABLE AT 'TIME'

```
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;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

\$\$; -- <--- FIXED: Added missing semicolon

```
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);
```

```
INSERT INTO tbl_employee(emp_name, emp_salary) VALUES ('Rohit', 60000);
```

-- Delete an employee

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

-- Check audit log

```
SELECT * FROM tbl_employee_audit;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

```
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
```

```
INSERT 0 1
```

```
DELETE 1
```

```
id | name | age | class
---+---+---+---
(0 rows)
```

```
CREATE TABLE
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
DELETE 1
```

```
sno | message
-----+-----
1 | Employee name Aman has been added at 2025-11-04 06:09:12.059901+00
2 | Employee name Rohit has been added at 2025-11-04 06:09:12.061924+00
3 | Employee name Aman has been deleted at 2025-11-04 06:09:12.063923+00
(3 rows)
```