



Experiment 7

Student Name: Ankit Kushwaha

UID: 23BCS14105

Branch: CSE

Section/Group: 23BCS_KRG-1/A

Semester: 5

Date of Performance: 16/08/25

Subject Name: Advanced Database
and Management System

Subject Code: 23CSP-333

1. Aim:

[MEDIUM] Design a Trigger such that whenever there is an insertion on student table then currently inserted or deleted row should be printed as it is on the output console window.

[HARD] Design a Postgres Trigger that (i) 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>". Do the same for deletion operation.

2. Tools Used: pgAdmin4

3. Code:

```
-- MEDIUM
CREATE TABLE TBL_STUDENT
(
  UID SERIAL PRIMARY KEY,
  NAME VARCHAR(20),
  AGE INT
);

INSERT INTO TBL_STUDENT(NAME, AGE)
VALUES
  ('ANKIT KUSHWAHA', 20),
  ('ANAND', 26),
  ('SAHIL', 22),
  ('PRISHA', 23);
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
CREATE OR REPLACE FUNCTION FN_TRG_STUDENT()  
RETURNS TRIGGER  
LANGUAGE plpgsql  
$$  
BEGIN  
    IF TG_OP = 'INSERT' THEN  
        RAISE NOTICE 'ID: % NAME: % AGE: %', NEW.UID,  
NEW.NAME, NEW.AGE;  
        RETURN NEW;  
  
    ELSIF TG_OP = 'DELETE' THEN  
        RAISE NOTICE 'ID: % NAME: % AGE: %', OLD.UID,  
OLD.NAME, OLD.AGE;  
        RETURN OLD;  
  
    END IF;  
  
    RETURN NULL;  
END;  
$;
```

```
CREATE OR REPLACE TRIGGER TRG_STUDENT  
AFTER INSERT OR DELETE  
ON TBL_STUDENT  
EXECUTE FUNCTION FN_TG_STUDENT();
```

----- HARD -----

```
CREATE OR REPLACE FUNCTION audit_employee_changes()  
RETURNS TRIGGER  
LANGUAGE plpgsql  
AS  
$$  
BEGIN  
    IF TG_OP = 'INSERT' THEN  
        INSERT INTO tbl_employee_audit(message)
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
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
('Punit', 50000);

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

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

4. Output: [MEDIUM]

Data Output	Messages	Notifications
<pre>NOTICE: ID: 1 NAME: PUNIT KUMAR AGE: 20 NOTICE: ID: 2 NAME: ANAND AGE: 26 NOTICE: ID: 3 NAME: SAHIL AGE: 22 NOTICE: ID: 4 NAME: PRISHA AGE: 23 INSERT 0 4 Query returned successfully in 44 msec.</pre>		

[HARD]

Data Output	Messages	Notifications
<div> <div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div> </div>		
sno [PK] integer	message text	
1	1	Employee name Aman has been added at 2025-10-30 00:38:02.449016+05:30
2	2	Employee name Aman has been deleted at 2025-10-30 00:38:02.449016+05:30

5. Learning Outcomes:

- Understand the concept of Database triggers – Learn how triggers automatically execute a function in response to database events like INSERT, DELETE etc.
- Implement Trigger Function using PLPGSQL.
- Differentiate between BEFORE and AFTER Triggers.
- Gained hands on experience for real life Trigger Applications.