



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT - 7

Student Name: Manit Saxena
Branch: BE-CSE
Semester: 5th
Subject Name: ADBMS

UID: 23BCS13532
Section/Group: KRG_1-B
Date of Performance: 28/10/2025
Subject Code: 23CSP-333

1. Aim: --- Medium Level Problem ---

a) **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.

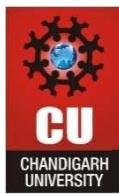
--- Hard Level Problem ---

b) **DESIGN A POSTGRESQL TRIGGERS THAT:**

1. 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>"
2. 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>"
3. The solution must use PostgreSQL triggers.

2. Platform Used:

Microsoft SQL Server Management Studio



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

3. SQL Code:

a) CREATE TABLE student (
 id SERIAL PRIMARY KEY,
 name VARCHAR(50),
 age INT,
 class VARCHAR(20)
);

```
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 TRIGGER trg_student_audit
AFTER INSERT OR DELETE
ON student
FOR EACH ROW
EXECUTE FUNCTION fn_student_audit();
```

```
INSERT INTO student (name, age, class)
VALUES ('Amit Sharma', 20, 'BCA');
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
INSERT INTO student (name, age, class)
VALUES ('Priya Singh', 21, 'BSc');
```

```
DELETE FROM student WHERE name = 'Amit Sharma';
```

OUTPUT-

```
NOTICE: Inserted Row -> ID: 1, Name: Amit Sharma, Age: 20, Class: BCA
NOTICE: Inserted Row -> ID: 2, Name: Priya Singh, Age: 21, Class: BSc
NOTICE: Deleted Row -> ID: 1, Name: Amit Sharma, Age: 20, Class: BCA
```

b) 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
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
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();

INSERT INTO tbl_employee(emp_name, emp_salary) VALUES ('Aman', 50000);
DELETE FROM tbl_employee WHERE emp_name = 'Aman';
SELECT * FROM tbl_employee_audit;
```

OUTPUT-

sno	message
1	Employee name Aman has been added at 2025-11-04 03:52:10.326481+00
2	Employee name Aman has been deleted at 2025-11-04 03:52:10.329438+00
(2 rows)	