



**DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING**

Discover. Learn. Empower.

WORKSHEET 7

Student Name: Naveen

UID: 23BAI70658

Branch: B.E CSE-H (AIML)

Section: 23-AIT_KRG_G1

Semester: 5th

Date of Performance: 28-10-2025

Subject Name: ADBMS

Subject Code: 23CSP-333

1. AIM:

i) Triggers: Student Data Change Monitoring (Medium)

EduSmart Institute wants to monitor all insertions and deletions in the student database. Whenever a new student record is inserted or deleted from the student table, the details of that record should be displayed on the PostgreSQL console window.

Objective:

Design a PostgreSQL trigger that:

1. Prints the complete details of the inserted or deleted student record using RAISE NOTICE.
2. Activates automatically after every INSERT or DELETE operation on the student table.

ii) Triggers: Employee Activity Logging (Hard)

TechSphere Solutions wants to maintain an automatic audit trail for all employee additions and deletions in the company database.

Whenever a new employee is added or removed from the tbl_employee table, an entry should be recorded in the tbl_employee_audit table for tracking purposes.

Objective:

Design a PostgreSQL trigger that:

1. Inserts a message in tbl_employee_audit whenever a new employee is added or deleted.
2. The message should include the employee's name and the current timestamp.
3. Activates automatically after every INSERT or DELETE operation on tbl_employee.

2. Tools Used: Postgres

Solutions:

MEDIUM

```
create table student( Name  
varchar(20),      Uid
```

```
smallint Primary Key,  
    course char(4)  
);
```

```
SET client_min_messages TO 'NOTICE';  
SHOW client_min_messages;
```

```
delete from student;
```

```
insert into student values('Souradeep', 132, 'AIML'),  
                      ('Sukhi', 231, 'AIML'),  
                      ('Palash', 332, 'WEBD'),  
                      ('Aadhitya', 425, 'DATA'),  
                      ('Aayush', 556, 'DATA');
```

```
select * from student;
```

```
create or replace function insert_del_statement()  
returns trigger language plpgsql  
as $$  
begin  
    if TG_OP = 'INSERT' then      raise notice 'Inserted rows:- %, %, %',  
        new.Name, new.Uid, new.course;  
        return new;  
    elseif TG_OP = 'DELETE' then          raise notice 'Deleted  
rows:- %, %, %', old.Name, old.Uid, old.course;  
        return old;  
    end if;  
  
    return null;  
end;  
$$
```

```
create or replace trigger trg_student  
after insert or delete on  
student for each row execute function  
insert_del_statement();
```

```
insert into student values('Arka', 442, 'AIML'), ('Rohit', 390, 'WEBD');
delete from student where Uid = 442 or Uid = 390; delete from
student where Uid = 556;
```

HARD

```
create table tbl_employee(
    emp_id int primary key,
    empName varchar(20),
    salary numeric
);
```

```
create table tbl_employee_audit(
    S_no serial primary key,
    logging text
);
```

```
SET client_min_messages TO 'NOTICE';
```

```
create or replace function
func_employee() returns trigger language
plpgsql
as
$$
begin
    if TG_OP = 'DELETE' then
        insert into
tbl_employee_audit(logging) values('Employee name ' || old.empName || 'has been
deleted at ' || now());
        return old;
    elseif TG_OP = 'INSERT' then
        insert into
tbl_employee_audit(logging) values('Employee name ' || new.empName || 'has
been deleted at ' || now());
        return new;
    end if;
```

```
        return null;  
    end;  
$$;
```

```
create or replace trigger trg_func_employee  
after insert or delete on  
tbl_employee for each row  
execute function  
func_employee();
```

```
insert into tbl_employee values(123, 'Souradeep', 120000.23),  
                                (321, 'Sukhi', 2311123.00),  
                                (424, 'Palash', 9800010);
```

```
delete from tbl_employee where emp_id = 123;
```

```
select * from tbl_employee_audit;
```

3. Output:

MEDIUM

```
NOTICE: Deleted rows:- Souradeep, 132, AIML  
NOTICE: Deleted rows:- Sukhi, 231, AIML  
NOTICE: Deleted rows:- Palash, 332, WEBD  
NOTICE: Deleted rows:- Aadhitya, 425, DATA  
DELETE 4
```

```
Query returned successfully in 41 msec.
```

```
NOTICE: Inserted rows:- Souradeep, 132, AIML  
NOTICE: Inserted rows:- Sukhi, 231, AIML  
NOTICE: Inserted rows:- Palash, 332, WEBD  
NOTICE: Inserted rows:- Aadhitya, 425, DATA  
NOTICE: Inserted rows:- Aayush, 556, DATA  
INSERT 0 5
```

```
Query returned successfully in 34 msec.
```

HARD

	s_no [PK] integer	logging text
1	7	Employee name Souradeephas been deleted at 2025-11-04 11:18:36.217566+0...
2	8	Employee name Sukhihas been deleted at 2025-11-04 11:18:36.217566+05:30
3	9	Employee name Palashhas been deleted at 2025-11-04 11:18:36.217566+05:30
4	10	Employee name Souradeephas been deleted at 2025-11-04 11:18:40.082253+0...

4. Learning Outcomes:

1. Understand the concept and purpose of database triggers in PostgreSQL.
2. Learn how to automate data tracking using AFTER INSERT and AFTER DELETE triggers.
3. Gain hands-on experience with trigger functions written in PL/pgSQL.
4. Develop the ability to implement audit logging for real-time database monitoring.
5. Enhance skills in maintaining data integrity and traceability in relational databases.