Experiment No 7

Student Name: Bhavya Gupta
Branch: BE CSE (IS)
Semester: 5th
Subject:ADBMS

UID: 23BIS70147
Section/Group: 23AIT-KRG(2A)
Date of Performance:10-Oct-2025
Subject Code: 23-CSP-333

[Medium]

Requirements: Design a trigger
Whenever there is an insertion on Student table then, the currently inserted or deleted row should be printed as it is on the output console window.

[Hard]

Requirements: Design a PostgreSQL trigger that:
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>"

SQL Quries

```
----------------------MEDIUM LEVEL PROBLEM----------------------------
/* REQUIREMENTS: DESIGN A TRIGGER
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
*/
-- Creating student table
CREATE TABLE student (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    grade VARCHAR(50)
);

-- Trigger function
CREATE OR REPLACE FUNCTION fn_student()
RETURNS TRIGGER
LANGUAGE plpgsql
AS
$$
BEGIN
    IF TG_OP = 'INSERT' THEN
        RAISE NOTICE 'Inserted Row -> ID: %, Name: %, Age: %, Grade: %',
                     NEW.id, NEW.name, NEW.age, NEW.grade;
        RETURN NEW;

    ELSIF TG_OP = 'DELETE' THEN
        RAISE NOTICE 'Deleted Row -> ID: %, Name: %, Age: %, Grade: %',
                     OLD.id, OLD.name, OLD.age, OLD.grade;
        RETURN OLD;
    END IF;

    RETURN NULL;
END;
$$;

-- Creating trigger
CREATE TRIGGER trg_student_audit
AFTER INSERT OR DELETE
ON student
FOR EACH ROW
EXECUTE FUNCTION fn_student();


-- Sample test data
INSERT INTO student (name, age, grade)
VALUES ('Aarav Sharma', 19, '12th Grade');

DELETE FROM student WHERE name = 'Aarav Sharma';

CREATE TABLE student (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    class VARCHAR(50)
);
```

```
----------------------HARD LEVEL PROBLEM----------------------------
/*
Requirements: DESIGN A PORSTGRESQL TRIGGERS THAT:

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>"
*/
-- Creating employee table
CREATE TABLE tbl_employee (
    emp_id SERIAL PRIMARY KEY,
    emp_name VARCHAR(100),
    designation VARCHAR(50),
    salary NUMERIC(10,2)
);

-- Creating employee audit table
CREATE TABLE tbl_employee_audit (
    audit_id SERIAL PRIMARY KEY,
    message TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Trigger function
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;
$$;

-- Creating trigger
CREATE TRIGGER trg_employee_audit
AFTER INSERT OR DELETE
ON tbl_employee
FOR EACH ROW
EXECUTE FUNCTION audit_employee_changes();

-- Sample test data
INSERT INTO tbl_employee (emp_name, designation, salary)
VALUES ('Rohan Mehta', 'Data Analyst', 72000);

SELECT * FROM tbl_employee_audit;

DELETE FROM tbl_employee WHERE emp_name = 'Rohan Mehta';

SELECT * FROM tbl_employee_audit;
```

Output:

Output:

```
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
INSERT 0 1
DELETE 1

psql:commands.sql:46: NOTICE:  Inserted Row -> ID: 1, Name: Aarav Sharma, Age: 19, Grade: 12th Grade
psql:commands.sql:48: NOTICE:  Deleted Row -> ID: 1, Name: Aarav Sharma, Age: 19, Grade: 12th Grade
psql:commands.sql:55: ERROR:  relation "student" already exists
```

Learning Outcome:

- Learned how to create and implement triggers in PostgreSQL.
- Understood how to use RAISE NOTICE to display inserted or deleted rows.
- Gained practical experience in linking trigger functions with database tables.
- Learned to automate audit logging for data insertion and deletion events.
- Understood the importance of triggers in maintaining data integrity and traceability.