Experiment-7

Student Name: Gagnesh Kakkar

UID: 23BCS11196

Branch: B.E-C.S.E

Section/Group: 23BCS_KRG-2B

Semester: 5th

Date of Performance: 08 October, 2025

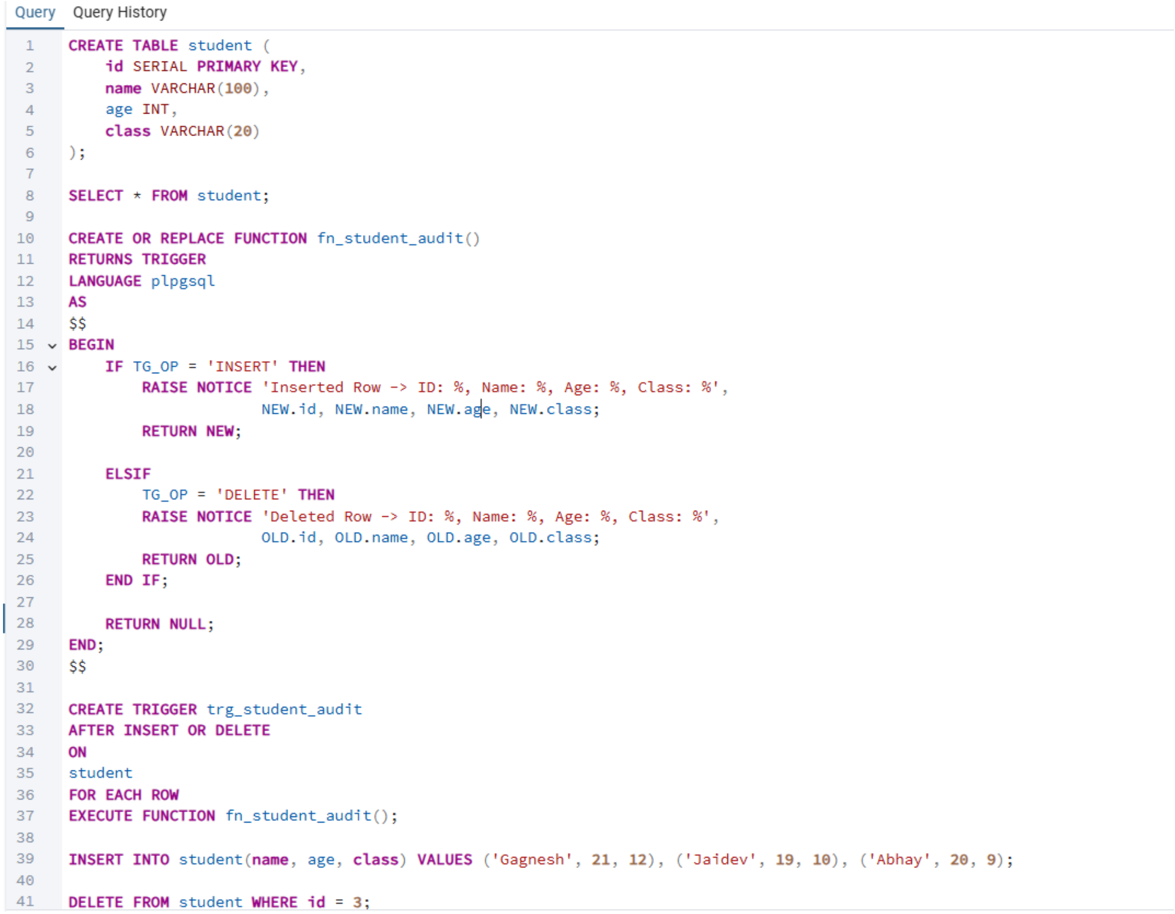
Subject Name: ADBMS

Subject Code: 23CSP-333

MEDIUM - LEVEL

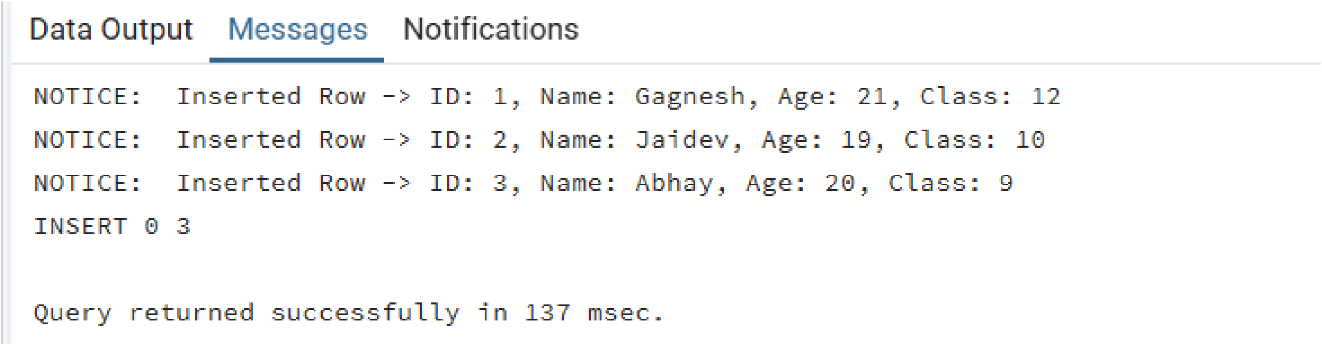
1. **Problem Title:** Design a Trigger Which:
   a). Whenever there is a insertion on student table then, the currently inserted or deleted, row should be printed as it ss on the output console window.

2. **SQL Commands:**

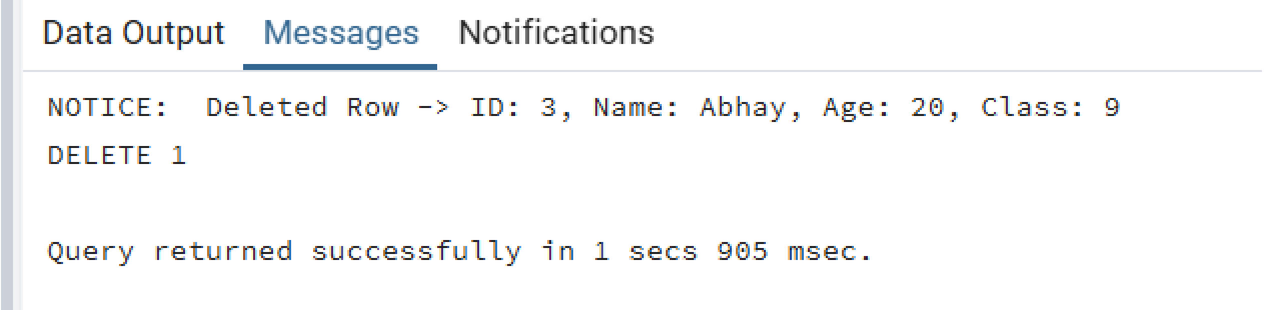


```sql
CREATE TABLE student (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    class VARCHAR(20)
);

SELECT * FROM student;

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 ('Gagnesh', 21, 12), ('Jaidev', 19, 10), ('Abhay', 20, 9);

DELETE FROM student WHERE id = 3;
```

3. **Output:**



```
NOTICE:  Inserted Row -> ID: 1, Name: Gagnesh, Age: 21, Class: 12
NOTICE:  Inserted Row -> ID: 2, Name: Jaidev, Age: 19, Class: 10
NOTICE:  Inserted Row -> ID: 3, Name: Abhay, Age: 20, Class: 9
INSERT 0 3

Query returned successfully in 137 msec.
```



```
NOTICE:  Deleted Row -> ID: 3, Name: Abhay, Age: 20, Class: 9
DELETE 1

Query returned successfully in 1 secs 905 msec.
```

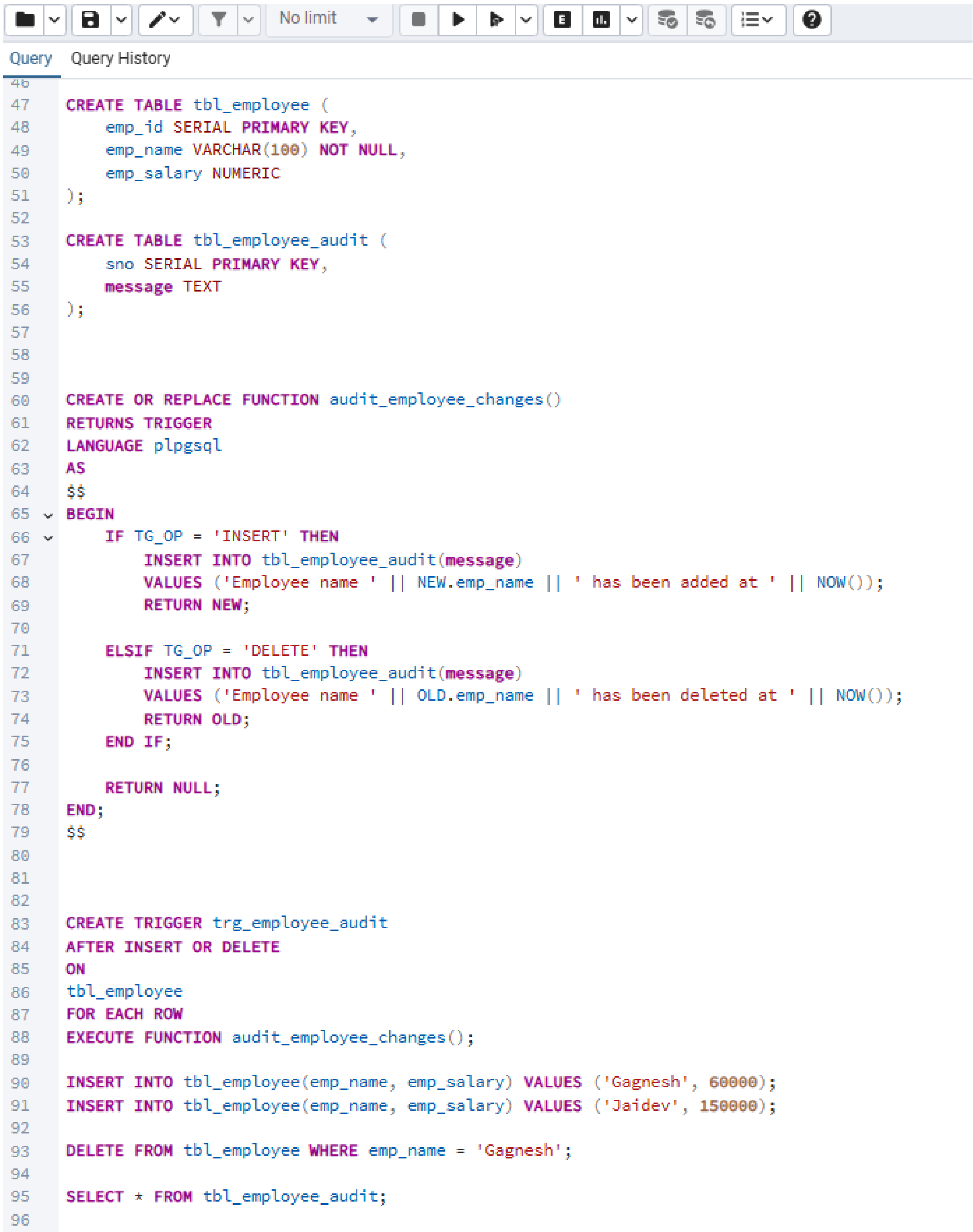
HARD - LEVEL

1. **Problem Title:** Design a PostgreSQL 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>"

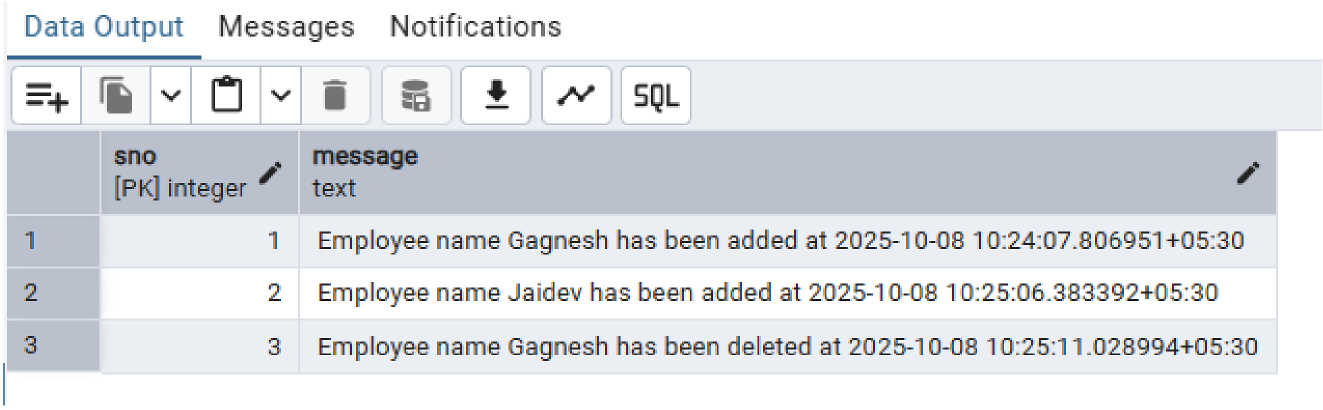
   Whenever an employee is deleted from tbl_employee, a record should be added to tbl_employee_audit like:
   "Employee name <emp_name> has been deleted at <current_time>"

2. **SQL Commands:**



```sql
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;
$$



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 ('Gagnesh', 60000);
INSERT INTO tbl_employee(emp_name, emp_salary) VALUES ('Jaidev', 150000);

DELETE FROM tbl_employee WHERE emp_name = 'Gagnesh';

SELECT * FROM tbl_employee_audit;

```

3. Output:



	sno [PK] integer	message text
1	1	Employee name Gagnesh has been added at 2025-10-08 10:24:07.806951+05:30
2	2	Employee name Jaidev has been added at 2025-10-08 10:25:06.383392+05:30
3	3	Employee name Gagnesh has been deleted at 2025-10-08 10:25:11.028994+05:30