



**CHANDIGARH  
UNIVERSITY**  
Discover. Learn. Empower.

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Advanced Database Management System**

**Experiment 7**

**23CSP-333**

**Submitted To:  
Faculty Name:  
Er. Alok Kumar**

**Submitted By:  
Name:  
Himanshu Gupta  
UID: 23BCS10889  
Section: KRG - 2B  
Semester: 5<sup>th</sup>**



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Aim:

Design a robust PostgreSQL transaction system for the students table where multiple student records are inserted in a single transaction. If any insert fails due to invalid data, only that insert should be rolled back while preserving the previous successful inserts using savepoints. The system should provide clear messages for both successful and failed insertions, ensuring data integrity and controlled error handling.

## Procedure:

- Step 1: Drop the existing table (if any) and create a new students table.
- Step 2: Perform a transaction block with multiple inserts and exception handling.
- Step 3: Demonstrate a wrong data type scenario with SAVEPOINTS.

## Code:

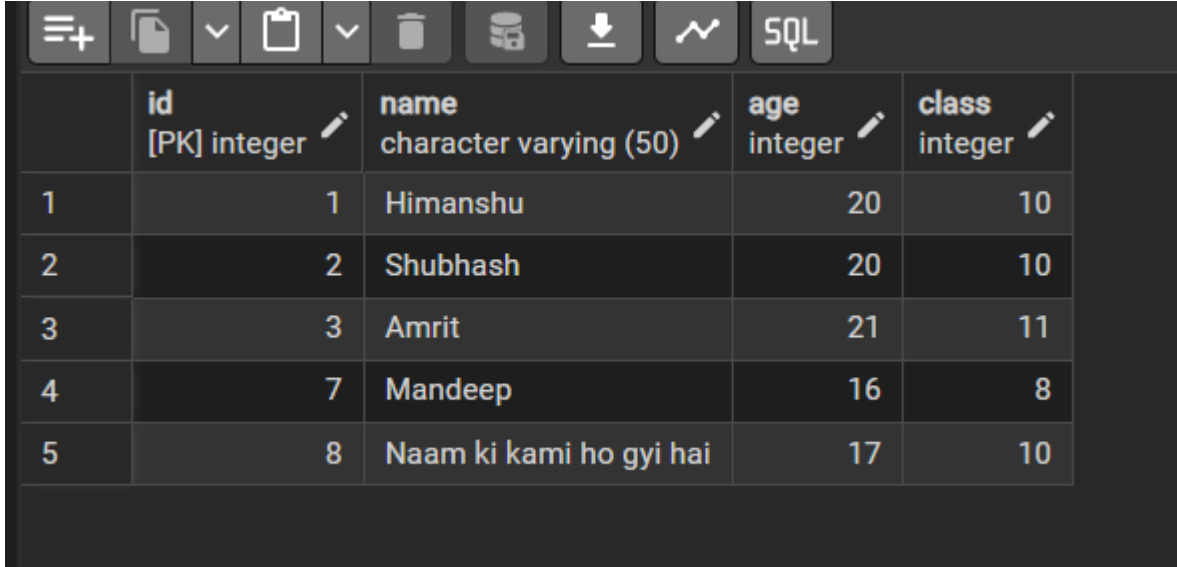
```
DROP TABLE IF EXISTS students;
CREATE TABLE students (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    class INT
);

DO $$
BEGIN
    INSERT INTO students(name, age, class) VALUES ('Himanshu', 20, 10);
    INSERT INTO students(name, age, class) VALUES ('Shubhash', 20, 10);
    INSERT INTO students(name, age, class) VALUES ('Amrit', 21, 11);
    RAISE NOTICE 'Transaction Successfully Done';
END;
$$;

SELECT * FROM students;

BEGIN; -- start transaction
SAVEPOINT sp1;
INSERT INTO students(name, age, class) VALUES (Mandeep,16,8);
SAVEPOINT sp2;
-- Try an invalid insert (will fail)
INSERT INTO students(name, age, class) VALUES ('Rahul','wrong',9);
-- Rollback only the failed one
ROLLBACK TO SAVEPOINT sp2;
-- Continue with valid data
INSERT INTO students(name, age, class) VALUES (Naam ki kami ho gyi hai,17,10);
ROLLBACK;

SELECT * FROM students;
```

**Output:**

	id [PK] integer	name character varying (50)	age integer	class integer
1	1	Himanshu	20	10
2	2	Shubhash	20	10
3	3	Amrit	21	11
4	7	Mandeep	16	8
5	8	Naam ki kami ho gyi hai	17	10

**Conclusion:**

This experiment demonstrated the use of **PostgreSQL triggers** to automate database actions and maintain audit logs.

- In the **medium-level problem**, the trigger displayed inserted or deleted student rows immediately in the console, providing real-time feedback.
- In the **hard-level problem**, the trigger recorded every insertion and deletion of employees into an audit table with timestamps, ensuring traceability and data integrity.

Overall, triggers help in **automating tasks, auditing changes, and maintaining consistent and reliable database operations** without manual intervention.