



## Experiment 8

**Student Name: ROHIT KUMAR**

**Branch: CSE**

**Semester: 5th**

**Subject Name: ADBMS**

**UID: 23BCS12640**

**Section/Group: KRG 3-A**

**Date of Performance: 09/10/2025**

**Subject Code: 23CSP-333**

### **1. Aim:**

#### **Problem :**

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.

### **2. Objective:**

- Understand how to implement robust transactions in PostgreSQL.
- Learn to use savepoints to rollback only failing statements without affecting successful inserts.
- Develop skills in PL/pgSQL error handling using BEGIN ... EXCEPTION ... END.
- Ensure data integrity by preventing invalid data from being committed.
- Learn to provide clear runtime messages for successful and failed inserts.

### **3. DBMS script and output:**

-- Drop the table if it exists

DROP TABLE IF EXISTS students;

-- Create the students table

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

DO \$\$



BEGIN

-- Insert Anisha

BEGIN

INSERT INTO students(name, age, class) VALUES ('Anisha', 16, 8);

RAISE NOTICE 'Inserted Anisha successfully';

EXCEPTION WHEN OTHERS THEN

RAISE NOTICE 'Failed to insert Anisha';

END;

-- Insert Neha

BEGIN

INSERT INTO students(name, age, class) VALUES ('Neha', 17, 8);

RAISE NOTICE 'Inserted Neha successfully';

EXCEPTION WHEN OTHERS THEN

RAISE NOTICE 'Failed to insert Neha';

END;

-- Insert Mayank

BEGIN

INSERT INTO students(name, age, class) VALUES ('Mayank', 19, 9);

RAISE NOTICE 'Inserted Mayank successfully';

EXCEPTION WHEN OTHERS THEN

RAISE NOTICE 'Failed to insert Mayank';

END;

-- Insert Rahul (wrong data type)

BEGIN

INSERT INTO students(name, age, class) VALUES ('Rahul', 'wrong', 9); -- this will fail

RAISE NOTICE 'Inserted Rahul successfully';

EXCEPTION WHEN OTHERS THEN

RAISE NOTICE 'Failed to insert Rahul (invalid data type)';

END;

-- Insert Sita

BEGIN

```
INSERT INTO students(name, age, class) VALUES ('Sita', 17, 10);
```

```
RAISE NOTICE 'Inserted Sita successfully';
```

EXCEPTION WHEN OTHERS THEN

```
RAISE NOTICE 'Failed to insert Sita';
```

END;

END

\$\$;

-- Verify final table contents

```
SELECT * FROM students;
```

Data Output Messages Notifications				
	id [PK] integer	name character varying (50)	age integer	class integer
1	1	Anisha	16	8
2	2	Neha	17	8
3	3	Mayank	19	9
4	4	Sita	17	10

## 4. Learning Outcomes (What I have Learnt):

- Ability to insert multiple student records safely in a single transaction.
- Ability to rollback only the failing insert while preserving previous successful inserts.
- Competence in handling runtime errors without aborting the whole transaction.
- Demonstrated ability to maintain the integrity and consistency of the database.
- Ability to provide clear, descriptive messages for each insert operation indicating success or failure.