Experiment 8

Student Name: Sunny Prabhakar	**UID: 23BIS70024**
Branch: BE CSE IS	**Section/Group:23AIT_KRG-2/A**
Semester: 5	**Date of Performance:16/08/25**
Subject Name: Advanced Database and Management System	**Subject Code: 23CSP-333**

1. Aim:

[HARD] 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. Tools Used: pgAdmin4

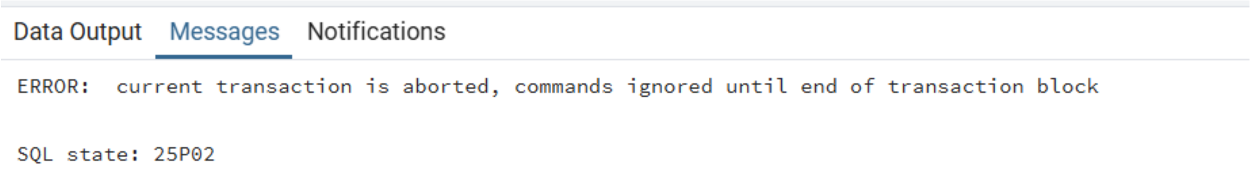
3. Code:

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


DO $$
BEGIN
    -- Start a transaction
    BEGIN
        -- Insert multiple students
        INSERT INTO students(name, age, class) VALUES
('Anisha',16,8);
        INSERT INTO students(name, age, class) VALUES
('Neha',17,8);
        INSERT INTO students(name, age, class) VALUES
('Mayank',19,9);
```

```
        -- If all succeed
        RAISE NOTICE ' Transaction Successfully Done';

    EXCEPTION
        WHEN OTHERS THEN
            -- If any insert fails
            RAISE NOTICE 'Transaction Failed..! Rolling back changes.';
            RAISE;  -- this will rollback the entire transaction
    END;
END;
$$;


SELECT * FROM students;

-------------------------WRONG DATA TYPE SCENARIO-------------------------
BEGIN;  -- start transaction

SAVEPOINT sp1;
INSERT INTO students(name, age, class) VALUES ('Aarav',16,8);

SAVEPOINT sp2;
BEGIN
    INSERT INTO students(name, age, class) VALUES ('Rahul','wrong',9);  -- fails
EXCEPTION WHEN OTHERS THEN
    RAISE NOTICE 'Failed to insert Rahul, rolling back to savepoint sp2';
    ROLLBACK TO SAVEPOINT sp2;
END;

-- Next insert
INSERT INTO students(name, age, class) VALUES ('Sita',17,10);

COMMIT;  -- commit all successful inserts
```

4. Output:

[HARD]



```
ERROR:  current transaction is aborted, commands ignored until end of transaction block

SQL state: 25P02
```

5. Learning Outcomes:

- Understand transaction control in PostgreSQL
- Implement save points for partial rollbacks.
- Handle run time errors using exception blocks.