CU CHANDIGARH UNIVERSITY

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Advanced Database Management System

Experiment 3.2

23CSP-333

Submitted To:
Faculty Name: Er. Alok Kumar

Submitted By:
Name: Garvi Dabas
UID: 23BCS11346
Section: KRG - 2B
Semester: 5th

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:

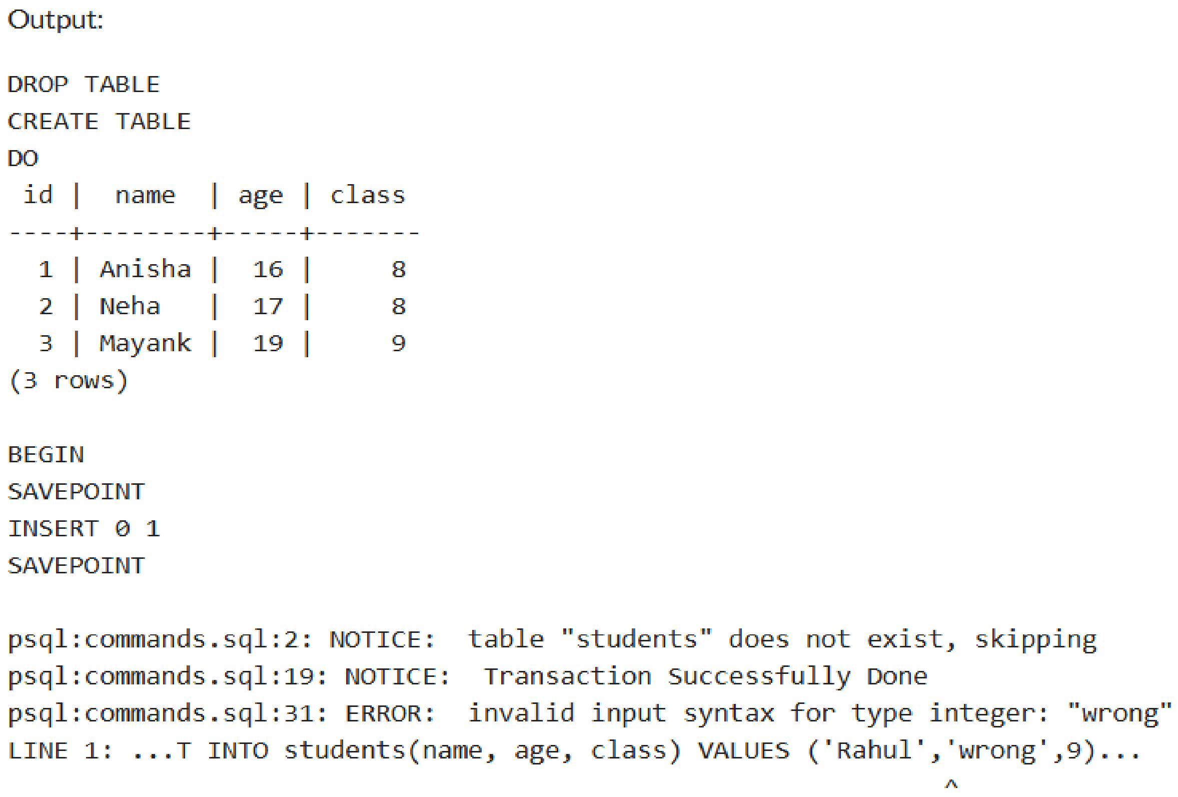
```
-- Drop and recreate table
DROP TABLE IF EXISTS students;
CREATE TABLE students (
id SERIAL PRIMARY KEY,
name VARCHAR(50),
age INT,
class INT
);

-- First transaction: all valid data
DO $$
BEGIN
INSERT INTO students(name, age, class) VALUES ('Anisha',16,8);
INSERT INTO students(name, age, class) VALUES ('Neha',17,8);
INSERT INTO students(name, age, class) VALUES ('Mayank',19,9);
RAISE NOTICE 'Transaction Successfully Done';
END;
$$;
SELECT * FROM students;

-- Second scenario: partial rollback using savepoints
BEGIN;  -- start transaction
SAVEPOINT sp1;
INSERT INTO students(name, age, class) VALUES ('Aarav',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 ('Sita',17,10);
```

COMMIT;
-- Show final data
SELECT * FROM students;commit all successful inserts

Output:



Output:

```
DROP TABLE
CREATE TABLE
DO
 id |  name  | age | class
----+--------+-----+-------
  1 | Anisha |  16 |     8
  2 | Neha   |  17 |     8
  3 | Mayank |  19 |     9
(3 rows)

BEGIN
SAVEPOINT
INSERT 0 1
SAVEPOINT

psql:commands.sql:2: NOTICE:  table "students" does not exist, skipping
psql:commands.sql:19: NOTICE:  Transaction Successfully Done
psql:commands.sql:31: ERROR:  invalid input syntax for type integer: "wrong"
LINE 1: ...T INTO students(name, age, class) VALUES ('Rahul','wrong',9)...
                                                             ^
```

Conclusion:

This experiment successfully demonstrates the concept of transactions and savepoints in PostgreSQL.
By using explicit transactions, savepoints, and rollback operations, we ensured that only invalid operations were undone while valid data remained intact.
It highlights the ACID properties of transactions:

- Atomicity: Ensures that operations are treated as a single logical unit.
- Consistency: Maintains data integrity after every transaction.
- Isolation: Prevents one transaction from interfering with another.
- Durability: Guarantees that committed changes persist in the database.

Through the use of savepoints, the experiment achieved partial rollbacks, allowing fine-grained control over transaction flow and error handling.
Thus, the designed PostgreSQL transaction system efficiently preserved data integrity and consistency, even in the presence of insertion errors.