EXPERIMENT- 08

Student Name:Himanshi Mehta **UID: 23BCS10694**
Branch: BE-CSE **Section/Group: KRG 1(B)**
Semester: 05 **Date of Performance: 23/10/25**
Subject Name: ADBMS **Subject Code: 23CSP-333**

Medium-Level Problem

1. Aim: To understand and implement transactions in PostgreSQL, including the use of BEGIN, COMMIT, ROLLBACK, and SAVEPOINT commands to ensure data integrity and control over changes.

2. Objective:

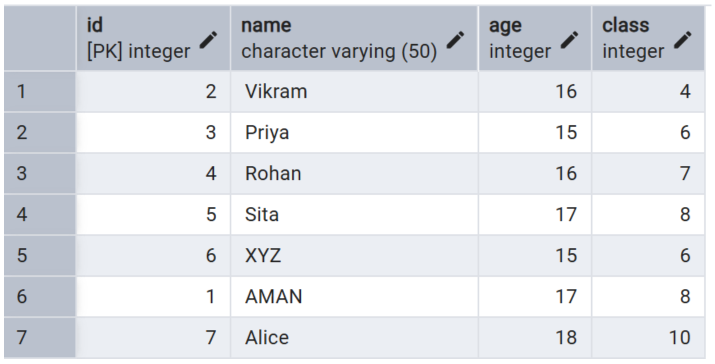
- Learn about implicit and explicit transactions.
- Understand **ACID** properties (Atomicity, Consistency, Isolation, Durability).
- Implement transaction control using **COMMIT**, **ROLLBACK**, and **SAVEPOINT**.
- Manage partial rollbacks using savepoints.

3. DBMS script and output:

```
CREATE TABLE Students
   ( Id INT PRIMARY KEY,
   Name VARCHAR(50) UNIQUE,
   Age INT,
   Class INT
);

INSERT INTO Students (ID, Name, Age, Class) VALUES
(1,'Aarav',17,8),
(2,'Vikram',16,4),
(3,'Priya',15,6),
(4,'Rohan',16,7),
(5,'Sita',17,8),
(6,'Kiran',15,6);

-- IMPLICIT TRANSACTION
UPDATE Students SET Name = 'XYZ' WHERE Id = 6;
```

```
-- EXPLICIT TRANSACTION
BEGIN TRANSACTION;
UPDATE Students SET Name = 'AMAN' WHERE Id = 1;
COMMIT;

-- ROLLBACK
BEGIN TRANSACTION;
UPDATE Students SET Name = 'TEMP' WHERE Id = 3;
ROLLBACK;

-- SAVEPOINTS
BEGIN TRANSACTION;
INSERT INTO Students(Id, Name, Age, Class) VALUES (7, 'Alice', 18, 10);
SAVEPOINT sp1;

INSERT INTO Students(Id, Name, Age, Class) VALUES (8, 'Bob', 17, 11);
SAVEPOINT sp2;

INSERT INTO Students(Id, Name, Age, Class) VALUES (9, 'Charlie', 16, 9);

-- Undo last insertion only
ROLLBACK TO SAVEPOINT sp2;

-- Continue
INSERT INTO Students(Id, Name, Age, Class) VALUES (10, 'Dina', 15, 8);

-- Undo all after sp1
ROLLBACK TO SAVEPOINT sp1;

COMMIT;

SELECT * FROM Students;
```

4. Output:



	id [PK] integer	name character varying (50)	age integer	class integer
1	2	Vikram	16	4
2	3	Priya	15	6
3	4	Rohan	16	7
4	5	Sita	17	8
5	6	XYZ	15	6
6	1	AMAN	17	8
7	7	Alice	18	10

Hard-Level Problem

1.Aim: Design a robust PostgreSQL transaction system for the **students** table where multiple student records are inserted within a single transaction.
If any insertion fails (due to duplicate or invalid data), only that particular insertion should be rolled back using **savepoints**, ensuring previously successful inserts remain intact.

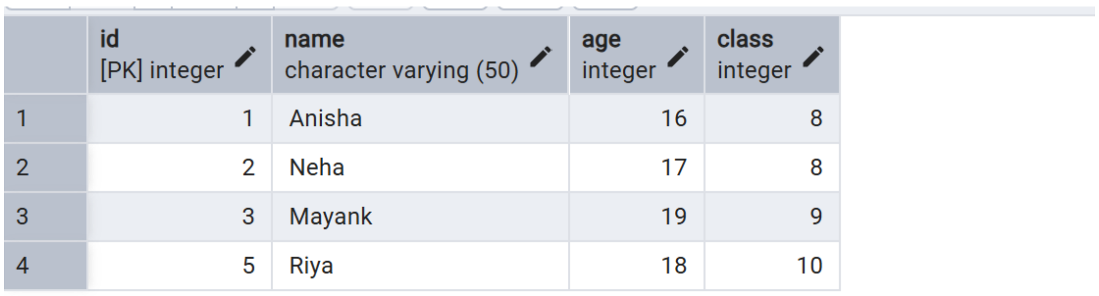
2. Objective:

- Implement transaction management with error handling.
- Use **SAVEPOINTS** to rollback partial transactions.
- Maintain data integrity during multi-step insert operations.
- Handle exceptions gracefully in PostgreSQL using `DO $$ BEGIN ... EXCEPTION ... END $$;`

3. DBMS script and output:

```
DROP TABLE IF EXISTS students;

CREATE TABLE students (

    id SERIAL PRIMARY KEY,
    name VARCHAR(50) UNIQUE,
    age INT,
    class INT
);

DO $$
DECLARE
BEGIN
    BEGIN
        INSERT INTO students(name, age, class) VALUES ('Anisha',16,8);
        RAISE NOTICE 'Inserted record: Anisha';
    EXCEPTION WHEN unique_violation THEN
        RAISE NOTICE 'Duplicate entry: Anisha skipped';
    END;

    BEGIN
        INSERT INTO students(name, age, class) VALUES ('Neha',17,8);
        RAISE NOTICE 'Inserted record: Neha';
    EXCEPTION WHEN unique_violation THEN
        RAISE NOTICE 'Duplicate entry: Neha skipped';
    END;

    BEGIN
        INSERT INTO students(name, age, class) VALUES ('Mayank',19,9);
        RAISE NOTICE 'Inserted record: Mayank';
    EXCEPTION WHEN unique_violation THEN
        RAISE NOTICE 'Duplicate entry: Mayank skipped';
    END;

    BEGIN
        INSERT INTO students(name, age, class) VALUES ('Anisha',17,9);
        RAISE NOTICE 'Inserted record: Anisha (second)';
    EXCEPTION WHEN unique_violation THEN
        RAISE NOTICE 'Duplicate entry: Anisha (second) skipped';
    END;

    BEGIN
        INSERT INTO students(name, age, class) VALUES ('Riya',18,10);
        RAISE NOTICE 'Inserted record: Riya';
    EXCEPTION WHEN unique_violation THEN
        RAISE NOTICE 'Duplicate entry: Riya skipped';
    END;
END;
$$;

SELECT * FROM students;
```

5. Output:



	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	5	Riya	18	10