Experiment-8

Student Name: Gagnesh Kakkar | **UID:** 23BCS11196

Branch: B.E-C.S.E | **Section/Group:** 23BCS_KRG-2B

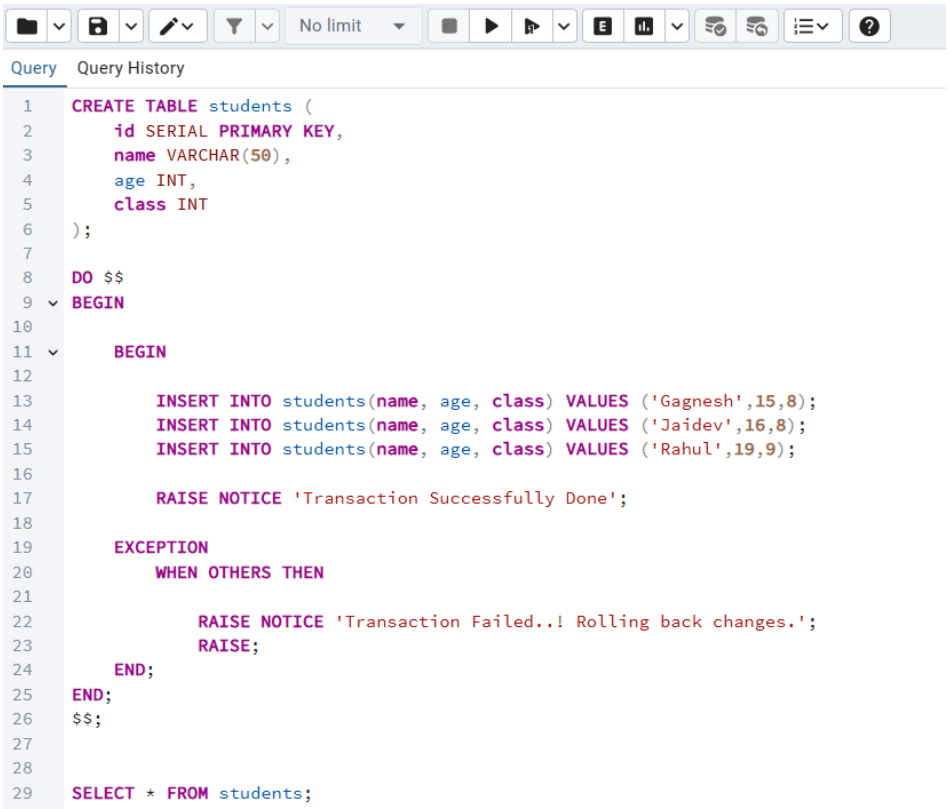
Semester: 5th | **Date of Performance:** 08 October, 2025

Subject Name: ADBMS | **Subject Code:** 23CSP-333

HARD - LEVEL

1. **Problem Title:** 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. **SQL Commands:**




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

DO $$
BEGIN

    BEGIN

        INSERT INTO students(name, age, class) VALUES ('Gagnesh',15,8);
        INSERT INTO students(name, age, class) VALUES ('Jaidev',16,8);
        INSERT INTO students(name, age, class) VALUES ('Rahul',19,9);

        RAISE NOTICE 'Transaction Successfully Done';

    EXCEPTION
        WHEN OTHERS THEN

            RAISE NOTICE 'Transaction Failed..! Rolling back changes.';
            RAISE;
    END;
END;
$$;


SELECT * FROM students;
```

```
Query  Query History

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

DO $$
BEGIN

    BEGIN

        INSERT INTO students(name, age, class) VALUES ('Gagnesh',15,8);
        INSERT INTO students(name, age, class) VALUES ('Jaidev',16,8);
        INSERT INTO students(name, age, class) VALUES ('Rahul',19,9);
        INSERT INTO students(name, age, class) VALUES ('Jatin',19,9);
        INSERT INTO students(name, age, class) VALUES (19,9);

        RAISE NOTICE 'Transaction Successfully Done';

    EXCEPTION
        WHEN OTHERS THEN

            RAISE NOTICE 'Transaction Failed..! Rolling back changes.';
            RAISE;
    END;
END;
$$;


SELECT * FROM students;
```

3. Output:

Data Output Messages Notifications

```
NOTICE:  Transaction Successfully Done

Successfully run. Total query runtime: 113 msec.
3 rows affected.
```

Data Output Messages Notifications

	id [PK] integer	name character varying (50)	age integer	class integer
1	1	Gagnesh	15	8
2	2	Jaidev	16	8
3	3	Rahul	19	9

Data Output Messages Notifications

```
NOTICE:  Transaction Failed..! Rolling back changes.

ERROR:  INSERT has more target columns than expressions
LINE 1: INSERT INTO students(name, age, class) VALUES (19,9)
                                        ^
QUERY:  INSERT INTO students(name, age, class) VALUES (19,9)
CONTEXT:  PL/pgSQL function inline_code_block line 10 at SQL statement

SQL state: 42601
```

Data Output Messages Notifications

	id [PK] integer	name character varying (50)	age integer	class integer
1	1	Gagnesh	15	8
2	2	Jaidev	16	8
3	3	Rahul	19	9
4	4	Jatin	19	9