Name : Avdhut Pailwan **Roll No. :** BTB22
PRN : 2122000380 **Subject :** Advanced Database Systems Lab

Experiment No. : 8

Create and manage NoSQL Databases with Cassandra

Problem Statement:

1. Create keyspace : employee

```
cqlsh> CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> CREATE TABLE employee.emp_table (
   ...     emp_id int PRIMARY KEY,
   ...     name text,
   ...     city text,
   ...     designation text,
   ...     experience float
   ... );
```

2. Create : emp_table (
   ... emp_id int,
   ... name text,
   ... city text,
   ... designation text,
   ... experience float,
   ... primary key(emp_id));

```
cqlsh> CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> CREATE TABLE employee.emp_table (
   ...     emp_id int PRIMARY KEY,
   ...     name text,
   ...     city text,
   ...     designation text,
   ...     experience float
   ... );
```

3. Perform following operations on created table:
   a. Insert rows

```
cqlsh> INSERT INTO employee.emp_table1 (emp_id, name, city, designation, experience) VALUES (1, 'Alice', 'New York', 'Software Engineer', 5.0);
cqlsh> SELECT * FROM employee.emp_table1;

 emp_id | city     | designation       | experience | name
--------+----------+-------------------+------------+-------
      1 | New York | Software Engineer |          5 | Alice

(1 rows)
cqlsh> INSERT INTO employee.emp_table1 (emp_id, name, city, designation, experience)
   ... VALUES (2, 'Bob', 'San Francisco', 'Data Scientist', 3.5);
cqlsh> SELECT * FROM employee.emp_table1;

 emp_id | city          | designation       | experience | name
--------+---------------+-------------------+------------+-------
      1 |      New York | Software Engineer |          5 | Alice
      2 | San Francisco |    Data Scientist |        3.5 |   Bob
```

   b. Update rows

```
cqlsh> UPDATE employee.emp_table1 SET city = 'Boston' WHERE emp_id = 2;
cqlsh> INSERT INTO employee.emp_table1 (emp_id, name, city, designation, experience)
   ... VALUES (4, 'David', 'Chicago', 'Project Manager', 6.0);
cqlsh> SELECT * FROM employee.emp_table1;

 emp_id | city        | designation       | experience | name
--------+-------------+-------------------+------------+---------
      1 |    New York | Software Engineer |          5 |   Alice
      2 |      Boston |    Data Scientist |        3.5 |     Bob
      4 |     Chicago |   Project Manager |          6 |   David
      3 | Los Angeles |   DevOps Engineer |          4 | Charlie
```

c. Update rows with upsert


```
cqlsh> UPDATE employee.emp_table1 SET city = 'Boston' WHERE emp_id = 2;
cqlsh> INSERT INTO employee.emp_table1 (emp_id, name, city, designation, experience)
   ... VALUES (4, 'David', 'Chicago', 'Project Manager', 6.0);
cqlsh> SELECT * FROM employee.emp_table1;

 emp_id | city        | designation       | experience | name
--------+-------------+-------------------+------------+---------
      1 |    New York | Software Engineer |          5 |   Alice
      2 |      Boston |    Data Scientist |        3.5 |     Bob
      4 |     Chicago |   Project Manager |          6 |   David
      3 | Los Angeles |   DevOps Engineer |          4 | Charlie
```


d. Retrieve data from table


```
cqlsh> SELECT * FROM employee.emp_table1;

 emp_id | city        | designation       | experience | name
--------+-------------+-------------------+------------+---------
      1 |    New York | Software Engineer |          5 |   Alice
      2 |      Boston |    Data Scientist |        3.5 |     Bob
      4 |     Chicago |   Project Manager |          6 |   David
      3 | Los Angeles |   DevOps Engineer |          4 | Charlie
```


e. Alter table add columns ((email set<text>, expertise list<text>, prev_jobs map<text, int>

```
cqlsh> ALTER TABLE employee.emp_table1 ADD email set<text>;
cqlsh> ALTER TABLE employee.emp_table1 ADD expertise list<text>;
cqlsh> ALTER TABLE employee.emp_table1 ADD prev_jobs map<text, int>;
```

f. Insert new rows

```
cqlsh> INSERT INTO employee.emp_table1 (emp_id, name, city, designation, experience, email, expertise, prev_jobs)
   ... VALUES (
   ...     5,
   ...     'Emma',
   ...     'Seattle',
   ...     'UI/UX Designer',
   ...     4.5,
   ...     {'emma@gmail.com', 'emma.doe@company.com'},
   ...     ['UI Design', 'Frontend Development'],
   ...     {'Company A': 2, 'Company B': 3}
   ... );
```

g. Delete rows and values

```
cqlsh> DELETE FROM employee.emp_table1 WHERE emp_id = 3;
cqlsh> DELETE email['emma@gmail.com'] FROM employee.emp_table1 WHERE emp_id = 5;
```

4. create table product(
... id uuid,
... name text,
... price float,
... quan int,
... primary key(id));

```
cqlsh> CREATE TABLE employee.product (
   ...     id uuid PRIMARY KEY,
   ...     name text,
   ...     price float,
   ...     quan int
   ... );
```

5. Perform following operations on created table:

a. Insert rows

```
cqlsh> INSERT INTO employee.product (id, name, price, quan)
   ... VALUES (uuid(), 'Laptop', 1500.00, 10);
cqlsh> INSERT INTO employee.product (id, name, price, quan)
   ... VALUES (uuid(), 'Smartphone', 800.00, 50);
cqlsh> INSERT INTO employee.product (id, name, price, quan)
   ... VALUES (uuid(), 'Headphones', 200.00, 100);
```

b. Alter table product add (inv_date timestamp, available boolean);

```
cqlsh> ALTER TABLE employee.product ADD inv_date timestamp;
cqlsh> ALTER TABLE employee.product ADD available boolean;
cqlsh> INSERT INTO employee.product (id, name, price, quan, inv_date, available)
   ... VALUES (uuid(), 'Tablet', 300.00, 25, toTimestamp(now()), true);
cqlsh> INSERT INTO employee.product (id, name, price, quan, inv_date, available)
   ... VALUES (uuid(), 'Smartwatch', 250.00, 15, toTimestamp(now()), false);
```

c. Insert new rows

```
cqlsh> SELECT * FROM employee.product;

 id                                   | available | inv_date
 | name       | price | quan
--------------------------------------+-----------+--------------------------+------------+-------+------
 1426c416-cb64-4caa-a68e-a06897dd0898 |     False | 2024-11-23 04:33:45+0000 | Smartwatch |   250 |   15
 3ee31249-4b46-4228-be9b-f8e3fbc5cf0b |      null |                     null | Smartphone |   800 |   50
 92dbb20c-d0df-455d-a79f-d3b73349330b |      null |                     null |     Laptop |  1500 |   10
 f93fe752-592c-4f43-a16a-0a1e034e6b46 |      True | 2024-11-23 04:33:33+0000 |     Tablet |   300 |   25
 b47aed49-efc2-4b84-92ee-767b63b874f1 |      null |                     null | Headphones |   200 |  100
```