

# Experiment No. 08

## Create and manage NoSQL Databases with Cassandra

**Name:** Aum Shendge

**Div.:** B

**Roll No.:** 17

**PRN:** 2122000270

---

### Problem Statement :

1. Create keyspace : employee
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> USE employee;  
cqlsh:employee>  
cqlsh:employee> CREATE TABLE 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:employee> INSERT INTO emp_table (emp_id, name, city, designation, experience)
... VALUES (1, 'Alice', 'New York', 'Engineer', 3.5);
cqlsh:employee> select * from employee.emp_table;
```

emp_id	city	designation	experience	name
1	New York	Engineer	3.5	Alice

(1 rows)

```
cqlsh:employee> INSERT INTO emp_table (emp_id, name, city, designation, experience)
... VALUES (2, 'Bob', 'San Francisco', 'Manager', 7.0);
cqlsh:employee> select * from employee.emp_table;
```

emp_id	city	designation	experience	name
1	New York	Engineer	3.5	Alice
2	San Francisco	Manager	7	Bob

(2 rows)

#### b. Update rows

```
cqlsh:employee> UPDATE emp_table
... SET city = 'Chicago', experience = 4.0
... WHERE emp_id = 1;
```

#### c. Update rows with upsert

```
cqlsh:employee> INSERT INTO emp_table (emp_id, name, city, designation, experience)
... VALUES (3, 'Charlie', 'Los Angeles', 'Analyst', 2.5);
cqlsh:employee> select * from employee.emp_table;
```

emp_id	city	designation	experience	name
1	Chicago	Engineer	4	Alice
2	San Francisco	Manager	7	Bob
3	Los Angeles	Analyst	2.5	Charlie

(3 rows)

#### d. Retrieve data from table

```
cqlsh:employee> SELECT * FROM emp_table;
```

emp_id	city	designation	experience	name
1	Chicago	Engineer	4	Alice
2	San Francisco	Manager	7	Bob
3	Los Angeles	Analyst	2.5	Charlie

(3 rows)

```
cqlsh:employee>
```

```
cqlsh:employee> SELECT name, city FROM emp_table WHERE emp_id = 1;
```

name	city
Alice	Chicago

- e. Alter table add columns ((email set<text>, expertise list<text>, prev\_jobs map<text, int>)

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

f.

Insert new rows

```
cqlsh:employee> INSERT INTO emp_table (emp_id, name, city, designation, experience, email, expertise, prev_jobs)
... VALUES (
... 4, 'Diana', 'Seattle', 'Architect', 10.0,
... {'diana@example.com'},
... ['Java', 'Python', 'C++'],
... {'CompanyA': 5, 'CompanyB': 3}
... );
cqlsh:employee> SELECT * FROM employee.emp_table;
```

emp_id	city	designation	email	experience	expertise	name	prev_jobs
1	Chicago	Engineer	null	4	null	Alice	null
2	San Francisco	Manager	null	7	null	Bob	null
4	Seattle	Architect	{'diana@example.com'}	10	['Java', 'Python', 'C++']	Diana	{'CompanyA': 5, 'CompanyB': 3}
3	Los Angeles	Analyst	null	2.5	null	Charlie	null

(4 rows)

- g. Delete rows and values

```
cqlsh:employee> DELETE FROM emp_table WHERE emp_id = 2;
```

#### 4. create table product(

- ... id uuid,
- ... name text,
- ... price float,
- ... quan int,
- ... primary key(id));

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

#### 5. Perform following operations on created table:

a. Insert rows

```
cqlsh:employee> INSERT INTO product (id, name, price, quan)
... VALUES (uuid(), 'Laptop', 1200.00, 10);
cqlsh:employee>
cqlsh:employee> INSERT INTO product (id, name, price, quan)
... VALUES (uuid(), 'Phone', 800.00, 25);
```

b. Alter table product add (inv\_date timestamp, available boolean);

```
cqlsh:employee> ALTER TABLE product
... ADD inv_date timestamp;
cqlsh:employee> ALTER TABLE product
... ADD available boolean;
```

c. Insert new rows

```
cqlsh:employee> INSERT INTO product (id, name, price, quan, inv_date, available)
... VALUES (uuid(), 'Tablet', 500.00, 15, toTimestamp(now()), true);
cqlsh:employee> select * from employee.product;
```

id	available	inv_date	name	price	quan
483e08ef-6746-4ab2-965f-ae30d78f9cf1	null	null	Phone	800	25
5a7b79c6-dee1-4377-8686-69a706aef3d7	null	null	Laptop	1200	10
cf2a5b97-0f10-4f05-8f72-91b4b67a6336	True	2024-11-25 05:31:41+0000	Tablet	500	15

(3 rows)