

## LABORATORY PROGRAM – 2

### Perform the following DB operations using Cassandra

#### Questions:

- Create a keyspace by name Employee
- Create a column family by name
  - Employee-Info with attributes
  - Emp\_Id Primary Key, Emp\_Name,
  - Designation, Date\_of\_Joining,
  - Salary, Dept\_Name
- Insert the values into the table in batch
- Update Employee name and Department of Emp-Id 121
- Sort the details of Employee records based on salary
- Alter the schema of the table Employee\_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
- Update the altered table to add project names.
- Create a TTL of 15 seconds to display the values of Employees.

#### OBSERVATION

#### COMMAND WITH OUTPUT

```
cqlsh> CREATE KEYSPACE IF NOT EXISTS Employee
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info (
...     Emp_Id INT PRIMARY KEY,
...     Emp_Name TEXT,
...     Designation TEXT,
...     Date_of_Joining DATE,
...     Salary DOUBLE,
...     Dept_Name TEXT
... );
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (121, 'John Doe', 'Manager', '2018-01-01', 90000, 'HR');
...
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (122, 'Alice Smith', 'Developer', '2019-05-21', 75000, 'IT');
...
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (123, 'Rahul Roy', 'Analyst', '2020-07-15', 65000, 'IT');
... APPLY BATCH;
cqlsh:employee> UPDATE Employee_Info
... SET Emp_Name = 'John Smith', Dept_Name = 'Finance'
... WHERE Emp_Id = 121;
cqlsh:employee> select * from Employee_Info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
123	2020-07-15	IT	Analyst	Rahul Roy	65000
122	2019-05-21	IT	Developer	Alice Smith	75000
121	2018-01-01	Finance	Manager	John Smith	90000

(3 rows)

```

(3 rows)
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_By_Dept (
...   Dept_Name TEXT,
...   Salary DOUBLE,
...   Emp_Id INT,
...   Emp_Name TEXT,
...   Designation TEXT,
...   Date_of_Joining DATE,
...   PRIMARY KEY (Dept_Name, Salary, Emp_Id)
... ) WITH CLUSTERING ORDER BY (Salary DESC, Emp_Id ASC);
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
... VALUES ('HR', 90000, 121, 'John Smith', 'Manager', '2018-01-01');
...
... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
... VALUES ('IT', 75000, 122, 'Alice Smith', 'Developer', '2019-05-21');
...
... INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
... VALUES ('IT', 65000, 123, 'Rahul Roy', 'Analyst', '2020-07-15');
... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_By_Dept WHERE Dept_Name = 'IT';

dept_name | salary | emp_id | date_of_joining | designation | emp_name
-----+-----+-----+-----+-----+-----
IT | 75000 | 122 | 2019-05-21 | Developer | Alice Smith
IT | 65000 | 123 | 2020-07-15 | Analyst | Rahul Roy

(2 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects SET<TEXT>;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'ERP System', 'HR Portal'} WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (124, 'Sneha Kapoor', 'Tester', '2023-03-10', 55000, 'QA') USING TTL 15;
cqlsh:employee> select * from Employee_Info;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
123 | 2020-07-15 | IT | Analyst | Rahul Roy | null | 65000
122 | 2019-05-21 | IT | Developer | Alice Smith | null | 75000
121 | 2018-01-01 | Finance | Manager | John Smith | {'ERP System', 'HR Portal'} | 90000

(3 rows)

```