

## **Lab 4 Cassandra Part - I**

1.What is the command used to create a keyspace named **Employee** with SimpleStrategy and replication factor 1?

```
CREATE KEYSPACE Employee
```

```
WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
```

How do you create a table named **Employee\_Info** with fields for ID, name, designation, joining date, salary, and department?

```
CREATE TABLE Employee_Info (
```

```
    Emp_Id int PRIMARY KEY,
```

```
    Emp_Name text,
```

```
    Designation text,
```

```
    Date_of_Joining date,
```

```
    Salary float,
```

```
    Dept_Name text
```

```
);
```

2.How do you insert multiple records in a batch in Cassandra?

```
BEGIN BATCH
```

```
INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
```

```
VALUES (121, 'Anit', 'Manager', '2018-02-01', 70000.0, 'Sales');
```

```
INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
```

```
VALUES (122, 'Priya', 'Developer', '2020-06-15', 50000.0, 'IT');
```

```
INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
```

```
VALUES (123, 'Rahul', 'Analyst', '2019-11-20', 60000.0, 'Finance');
```

APPLY BATCH;

3.What query updates the name and department of the employee with Emp\_Id = 121?

```
UPDATE Employee_Info
```

```
SET Emp_Name = 'Anit Kumar', Dept_Name = 'Marketing'
```

```
WHERE Emp_Id = 121;
```

4.What is the correct query to fetch employees whose salary is greater than 0 using ALLOW FILTERING?

```
SELECT * FROM Employee_Info
```

```
WHERE Salary > 0
```

```
ALLOW FILTERING;
```

5.How do you add a new column Projects of type set<text> to the table?

```
ALTER TABLE Employee_Info ADD Projects set<text>;
```

6.How do you update the projects of employee with Emp\_Id = 121?

```
UPDATE Employee_Info
```

```
SET Projects = {'ProjectA', 'ProjectB'}
```

```
WHERE Emp_Id = 121;
```

7.How do you insert a new record into the updated table including the new Projects column with TTL?

```
INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
```

```
VALUES (124, 'Neha', 'HR', '2022-03-01', 45000.0, 'HR')
```

```
USING TTL 15;
```

```

cqlsh> CREATE KEYSPACE Employee
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (
...     Emp_Id int PRIMARY KEY,
...     Emp_Name text,
...     Designation text,
...     Date_of_Joining date,
...     Salary float,
...     Dept_Name text
... );
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (121, 'Amit', 'Manager', '2018-02-01', 70000.0, 'Sales');
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (122, 'Priya', 'Developer', '2020-06-15', 50000.0, 'IT');
... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (123, 'Rahul', 'Analyst', '2019-11-20', 60000.0, 'Finance');
... APPLY BATCH;
cqlsh:employee> UPDATE Employee_Info
... SET Emp_Name = 'Amit Kumar', Dept_Name = 'Marketing'
... WHERE Emp_Id = 121;
cqlsh:employee>
cqlsh:employee> SELECT * FROM Employee_Info
... WHERE Salary IS NOT NULL
... ALLOW FILTERING;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Unsupported restriction: salary IS NOT NULL"
cqlsh:employee> SELECT * FROM Employee_Info
... WHERE Salary > 0
... ALLOW FILTERING;

 emp_id | date_of_joining | dept_name | designation | emp_name | salary
-----+-----+-----+-----+-----+-----
  123 | 2019-11-20 | Finance | Analyst | Rahul | 60000
  122 | 2020-06-15 | IT | Developer | Priya | 50000
  121 | 2018-02-01 | Marketing | Manager | Amit Kumar | 70000

(3 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects set<text>;
cqlsh:employee> UPDATE Employee_Info
... SET Projects = {'ProjectA', 'ProjectB'}
... WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
... VALUES (124, 'Neha', 'HR', '2022-03-01', 45000.0, 'HR')
... USING TTL 15;
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----
  123 | 2019-11-20 | Finance | Analyst | Rahul | null | 60000
  122 | 2020-06-15 | IT | Developer | Priya | null | 50000
  121 | 2018-02-01 | Marketing | Manager | Amit Kumar | {'ProjectA', 'ProjectB'} | 70000

(3 rows)

```