8/4/25

Lab-05

Performing the following DB operations using Cassandra.

1. Create a Keyspace by name Employee

```
CREATE KEYSPACE Employee WITH
replication = {'class': 'SimpleStrategy',
'replication_factor': 1};
DESCRIBE KEYSPACE;
```

2. Create a column family by name Employee-Info with attributes Emp-ID Primary Key, Emp_Name, Designation, Date-of-joining, Salary, Dept-Name.

```
USE Employee
CREATE TABLE Employee-Info (
    Emp-Id int PRIMARY KEY,
    Emp-Name text,
    Designation text,
    Date-of-Joining date,
    Salary int,
    Dept-Name text
);
```

3. Insert the values into the table in batch

```
BEGIN BATCH
INSERT INTO Employee-Info (Emp-id, Emp-Name,
Designation, Date-of-joining, Salary, Dept-Name)
VALUES (101, 'John Doe', 'Developer', 2024-01-15, 60000,
'IT');
```

INSERT INTO Employee-Info (Emp-id, Emp-Name, Designation, Date-of-Joining, Salary, Dept-Name)
VALUES (121, 'Jane Smith', 'Manager', 2019-03-20', 80000, 'HR');

VALUES (131, 'Mike Johnson', 'Analyst', 2021-06-20', 55000, 'Finance');
APPLY BATCH

SELECT * from Employee-info;

4. Update the Employee name & department of Emp-ID 121

UPDATE Employee-Info SET Emp-Name = 'Jane Doe', Dept-Name = 'IT'
WHERE Emp ID = 121;
SELECT * from Employee-Info;

5. Alter the schema of the table Employee-Info to add a column Projects with stores a set of projects done by the corresponding Employee

ALTER TABLE Employee-Info ADD Projects set <text>

6. UPDATE Employee-Info SET Projects = {'Project A', 'Project B'} WHERE Emp-ID = 101;
UPDATE Employee-Info SET Projects = {'Project X', 'Project Y'} WHERE Emp id = 121;
UPDATE Employee-Info SET Projects = {'Project Z'}
WHERE Emp-id = 131;

1. Create a TTL of 15 seconds to display values of Employees

```
INSERT INTO Employee_Info (Emp_ID, Emp_Name,
Designation, Date_of_Joining, Salary,
Dept_No, Level, Projects) VALUES (101,
'John Doe', 'Developer', '2020-01-15',
60000, 'IT', 'Project A', 'Project B')
USING TTL 15;

SELECT * from Employee_Info;
```

Date: 15/4/25
Page

Lab-06

Hadoop Basic commands execution in Ubuntu

1. To start hadoop shell in terminal:
> start-all.sh
> jps

2. To check existing directories
> hadoop fs -ls /

3. If no directories exist, create new
> hadoop fs -mkdir /mjn

4. Create a file on Desktop (locally)
file1.txt

5. Copy the file from desktop to hadoop file system file name
> hadoop fs -copyFromLocal /home/hadoop/Desktop/file1.txt /mjn/test.txt

6. To view the hadoop file test.txt
> nano /home/hadoop/Desktop/file1 /mjn/test1.txt
Ctrl + O → To save
Ctrl + X → To exit

bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: ~

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: $ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE employee
   ... DROP KEYSPACE employee;
SyntaxException: line 2:0 mismatched input 'DROP' expecting EOF (DROP KEYSPACE employee[DROP]...)
cqlsh> DROP KEYSPACE employee;
cqlsh> DESCRIBE KEYSPACES;

employee1     students     system_distributed  system_views
student_data  system       system_schema       system_virtual_schema
student_new   system_auth  system_traces

cqlsh> CREATE KEYSPACE Employee
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> DESCRIBE KEYSPACES;

employee      student_new  system_auth         system_traces
employee1     students     system_distributed  system_views
student_data  system       system_schema       system_virtual_schema

cqlsh> USE employee;
cqlsh:employee> CREATE TABLE Employee_Info (
           ...     Emp_Id int PRIMARY KEY,
           ...     Emp_Name text,
           ...     Designation text,
           ...     Date_of_Joining date,
           ...     Salary int,
           ...     Dept_Name text
           ... );
cqlsh:employee> BEGIN BATCH
           ... INSERT INTO Employee_Info(Emp_Id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (101,'John Doe','D
eveloper','2020-01-15',60000,'IT');
           ... INSERT INTO Employee_Info(Emp_Id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (121,'Jane Smith',
'Manager','2019-03-10',80000,'HR');
           ... INSERT INTO Employee_Info(Emp_Id,Emp_Name,Designation,Date_of_Joining,Salary,Dept_Name)VALUES (131,'Mike Johnson
','Analyst','2021-06-20',55000,'Finance');
           ... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_Info;\
Invalid syntax at char 29
  SELECT * FROM Employee_Info;\
                              ^
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name     | salary
--------+-----------------+-----------+-------------+--------------+--------
    121 |      2019-03-10 |        HR |     Manager |   Jane Smith |  80000
    131 |      2021-06-20 |   Finance |     Analyst | Mike Johnson |  55000
    101 |      2020-01-15 |        IT |   Developer |     John Doe |  60000
```

bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: ~

```
icted by an EQ or an IN."
cqlsh:employee> SELECT * FROM Employee_Info WHERE Emp_Id IN (101, 121, 131) ORDER BY Salary ASC;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Order by is currently only supported on the clustered colu
mns of the PRIMARY KEY, got salary"
cqlsh:employee> SELECT * FROM Employee_Info WHERE Emp_Id = 101 ORDER BY Salary ASC;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Order by is currently only supported on the clustered colu
mns of the PRIMARY KEY, got salary"
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects set<text>;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project A','Project B'} WHERE Emp_Id = 101;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project X','Project Y'} WHERE Emp_Id = 121;
cqlsh:employee> UPDATE Employee_Info SET Projects = {'Project Z'} WHERE Emp_Id = 131;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name     | projects                    | salary
--------+-----------------+-----------+-------------+--------------+-----------------------------+--------
    121 |      2019-03-10 |        IT |     Manager |     Jane Doe | {'Project X', 'Project Y'} |  80000
    131 |      2021-06-20 |   Finance |     Analyst | Mike Johnson |              {'Project Z'} |  55000
    101 |      2020-01-15 |        IT |   Developer |     John Doe | {'Project A', 'Project B'} |  60000

(3 rows)
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects)
           ... VALUES (101, 'John Doe', 'Developer', '2020-01-15', 60000, 'IT', {'Project A', 'Project B'})
           ... USING TTL 15;
cqlsh:employee>
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects) VALUES (
101,'John Doe','Developer','2020-01-15',60000,'IT',{'Project A','Project B'}) USING TTL 15;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name     | projects                    | salary
--------+-----------------+-----------+-------------+--------------+-----------------------------+--------
    121 |      2019-03-10 |        IT |     Manager |     Jane Doe | {'Project X', 'Project Y'} |  80000
    131 |      2021-06-20 |   Finance |     Analyst | Mike Johnson |              {'Project Z'} |  55000
    101 |      2020-01-15 |        IT |   Developer |     John Doe | {'Project A', 'Project B'} |  60000

(3 rows)
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects) VALUES (
141,'John Doe','Developer','2020-01-15',60000,'IT',{'Project A','Project B'}) USING TTL 15;
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name     | projects                    | salary
--------+-----------------+-----------+-------------+--------------+-----------------------------+--------
    121 |      2019-03-10 |        IT |     Manager |     Jane Doe | {'Project X', 'Project Y'} |  80000
    141 |      2020-01-15 |        IT |   Developer |     John Doe | {'Project A', 'Project B'} |  60000
    131 |      2021-06-20 |   Finance |     Analyst | Mike Johnson |              {'Project Z'} |  55000

(3 rows)
cqlsh:employee> SELECT * from Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name     | projects                    | salary
--------+-----------------+-----------+-------------+--------------+-----------------------------+--------
    121 |      2019-03-10 |        IT |     Manager |     Jane Doe | {'Project X', 'Project Y'} |  80000
    131 |      2021-06-20 |   Finance |     Analyst | Mike Johnson |              {'Project Z'} |  55000

(2 rows)
cqlsh:employee>
```