Lab-5
05 November
1BM17CS065

Cassandra Programs-To Do

I. Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee

```
cqlsh> create keyspace Employee with replication={'class':'SimpleStrategy', 'r
eplication_factor': 1};
```

2. Create a column family by name Employee-Info with attributes
Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary,
Dept_Name

```
cqlsh:employee> create table employee_info(
            ... emp_id int  PRIMARY KEY ,
            ... emp_name varchar,
            ... dept_name varchar,
            ... designation varchar,
            ... doj varchar,
            ... salary int);
```

3. Insert the values into the table in batch

```
cqlsh:employee> BEGIN BATCH
            ... insert INTO employee_info
            ... (emp_id,emp_name,dept_name, designation , doj , salary )
            ... VALUES (
            ... 2,'Saifur Rahman','TE','head','01-02-20',5000);
            ... insert INTO employee_info
            ... (emp_id,emp_name,dept_name, designation , doj , salary )
            ... VALUES (
            ... 3,'Mayank Wali','ME','Teacher','02-03-1299',10000);
            ... APPLY BATCH ;
cqlsh:employee> SELECT * FROM employee_info ;

 emp_id | dept_name | designation | doj        | emp_name      | salary
--------+-----------+-------------+------------+---------------+--------
      1 |       CSE |     Student |   01-09-29 | Prateek Aryan |    500
      2 |        TE |        head |   01-02-20 | Saifur Rahman |   5000
      3 |        ME |     Teacher | 02-03-1299 |   Mayank Wali |  10000
```

Lab-5
05 November
1BM17CS065

Next page

3. Update Employee name and Department of Emp-Id 121

```
cqlsh:employee> UPDATE
            ... employee_info
            ... SET
            ... emp_name='Prateek Aryan',dept_name='IT'
            ... WHERE
            ... emp_id =1;
cqlsh:employee> select* FROM employee_info;

 emp_id | dept_name | designation | doj        | emp_name      | salary
--------+-----------+-------------+------------+---------------+--------
      1 |        IT |     Student |   01-09-29 | Prateek Aryan |    500
      2 |        TE |        head |   01-02-20 | Saifur Rahman |   5000
      3 |        ME |     Teacher | 02-03-1299 |   Mayank Wali |  10000
```

4. Sort the details of Employee records based on salary

```
cqlsh:employee> select* FROM employee_info ORDER BY salary;
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY
 is only supported when the partition key is restricted by an EQ or an IN."
```

5. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

```
cqlsh:employee> ALTER TABLE employee_info
            ... ADD project set <varchar >;
cqlsh:employee> select * FROM employee_info ;

 emp_id | dept_name | designation | doj        | emp_name      | project | salary
--------+-----------+-------------+------------+---------------+---------+--------
      1 |        IT |     Student |   01-09-29 | Prateek Aryan |    null |    500
      2 |        TE |        head |   01-02-20 | Saifur Rahman |    null |   5000
      3 |        ME |     Teacher | 02-03-1299 |   Mayank Wali |    null |  10000
```

Lab-5
05 November
1BM17CS065

---

6. Update the altered table to add project names.

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

 emp_id | dept_name | designation | doj        | emp_name      | project                                    | salary
--------+-----------+-------------+------------+---------------+--------------------------------------------+--------
    120 |      null |        null |       null |          null | {'Investor Platform', 'Research Tool'} |   null
      1 |        IT |     Student |   01-09-29 | Prateek Aryan |                                       null |    500
      2 |        TE |        head |   01-02-20 | Saifur Rahman |                                       null |   5000
      3 |        ME |     Teacher | 02-03-1299 |   Mayank Wali |                                       null |  10000
```

7.Create a TTL of 15 seconds to display the values of Employees.

```
cqlsh:employee> BEGIN BATCH
           ... INSERT INTO
           ... employee_info (emp_id, emp_name, designation, doj , salary , dept_name )
           ... VALUES ( 123,'Harsh','Master','05-09-23',60000,'ME') USING TTL 15;
           ... APPLY BATCH ;
cqlsh:employee> SELECT TTL(designation) FROM employee_info WHERE ememp_id=123;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Undefined column name ememp
_id"
cqlsh:employee> SELECT TTL(designation) FROM employee_info WHERE emp_id=123;

 ttl(designation)
------------------

(0 rows)
```

II. Perform the following DB operations using Cassandra.

1.Create a keyspace by name Library

2. Create a column family by name Library-Info with attributes
Stud_Id Primary Key,
Counter_value of type Counter,
Stud_Name, Book-Name, Book-Id, Date_of_issue

```
cqlsh:employee> CREATE KEYSPACE library WITH replication = {'class': 'SimpleStrategy
', 'replication_factor': 3};
cqlsh:employee> CREATE COLUMNFAMILY library_info(
           ... stud_id uuid,
           ... counter_value counter,
           ... stud_name varchar ,
           ... book_name varchar ,
           ... book_id int,
           ... doi varchar ,
           ... primary KEY (stud_id, stud_name, book_name, book_id,doi));
```

3. Insert the values into the table in batch

---

3. Display the details of the table created and increase the value of the counter
4. Write a query to show that a student with id 112 has taken a book “BDA” 2 times.
5. Export the created column to a csv file
6. Import a given csv dataset from local file system into Cassandra column family