

BDA Lab 6 - 1BM17CS086

Cassandra Programs-To Do

I. Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee
2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name
3. Insert the values into the table in batch
4. Update Employee name and Department of Emp-Id 121
5. Sort the details of Employee records based on salary
6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
7. Update the altered table to add project names.
8. Create a TTL of 15 seconds to display the values of Employees.

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

Task I.1:

Create a keyspace by name Employee

```
CREATE KEYSPACE Employees WITH replication = { 'class': 'SimpleStrategy', 'replication_factor': 3};
```

```
saif@badger:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.8 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE Employees WITH replication = { 'class': 'SimpleStrategy', 'replication_factor': 3};
```

Task I.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 employees;
CREATE COLUMNFAMILY employee_info(Emp_Id INT PRIMARY KEY, Emp_Name VARCHAR, Designation VARCHAR, Date_of_Joining VARCHAR, Salary FLOAT, Dept_Name VARCHAR);
```

```
cqlsh> use employees;
cqlsh:employees> CREATE COLUMNFAMILY employee_info(Emp_Id INT PRIMARY KEY, Emp_Name VARCHAR, Designation VARCHAR, Date_of_Joining VARCHAR, Salary FLOAT, Dept_Name VARCHAR);
cqlsh:employees> █
```

Task I.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 (120, 'Carol', 'SDE', '01/06/2010', 85000, 'SE');

INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (121, 'Cecelia', 'SDE', '01/08/2010', 70000, 'SE');

INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (122, 'Sharon', 'Accountant', '01/04/2012', 50000, 'Finance');

INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (123, 'Barbara', 'Sales Manager', '01/04/2012', 45000, 'Sales');

APPLY BATCH;

SELECT * FROM employee_info;
```

```
cqlsh:employees> BEGIN BATCH INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (120, 'Carol', 'SDE', '01/06/2010', 85000, 'SE');
...
... INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (121, 'Cecelia', 'SDE', '01/08/2010', 70000, 'SE');
...
... INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (122, 'Sharon', 'Accountant', '01/04/2012', 50000, 'Finance');
...
... INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (123, 'Barbara', 'Sales Manager', '01/04/2012', 45000, 'Sales');
...
... APPLY BATCH;
cqlsh:employees> SELECT * FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
120	01/06/2010	SE	SDE	Carol	85000
123	01/04/2012	Sales	Sales Manager	Barbara	45000
122	01/04/2012	Finance	Accountant	Sharon	50000
121	01/08/2010	SE	SDE	Cecelia	70000

```
(4 rows)
cqlsh:employees> █
```

Task I.4:

Update Employee name and Department of Emp-Id 121

```
UPDATE employee_info SET emp_name = 'Nancy', dept_name = 'DA' WHERE emp_id = 121;
```

```
cqlsh:employees> UPDATE employee_info SET emp_name = 'Nancy', dept_name = 'DA' WHERE emp_id = 121
cqlsh:employees> SELECT * FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
120	01/06/2010	SE	SDE	Carol	85000
123	01/04/2012	Sales	Sales Manager	Barbara	45000
122	01/04/2012	Finance	Accountant	Sharon	50000
121	01/08/2010	DA	SDE	Nancy	70000

(4 rows)

Task I.5:

Sort the details of Employee records based on salary

```
SELECT * FROM employee_info ORDER BY salary;
```

Task I.6:

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

```
ALTER TABLE employee_info ADD project SET<VARCHAR>;

SELECT * FROM EMPLOYEE_INFO;
```

```
cqlsh:employees> ALTER TABLE employee_info ADD project SET<VARCHAR>;
cqlsh:employees>
cqlsh:employees> SELECT * FROM EMPLOYEE_INFO;
```

emp_id	date_of_joining	dept_name	designation	emp_name	project	salary
120	01/06/2010	SE	SDE	Carol	null	85000
123	01/04/2012	Sales	Sales Manager	Barbara	null	45000
122	01/04/2012	Finance	Accountant	Sharon	null	50000
121	01/08/2010	DA	SDE	Nancy	null	70000

(4 rows)

```
cqlsh:employees>
```

Task I.7:

Update the altered table to add project names.

```
UPDATE employee_info SET project = project + {'Research Tool', 'Investor Platform'} WHERE emp_id = 120;

SELECT * FROM employee_info;
```

```
cqlsh:employees> UPDATE employee_info SET project = project + {'Research Tool', 'Investor Platform'} WHERE emp_id = 120;
cqlsh:employees>
cqlsh:employees> SELECT * FROM employee_info;
```

emp_id	date_of_joining	dept_name	designation	emp_name	project	salary
120	01/06/2010	SE	SDE	Carol	{'Investor Platform', 'Research Tool'}	85000
123	01/04/2012	Sales	Sales Manager	Barbara		45000
122	01/04/2012	Finance	Accountant	Sharon		50000
121	01/08/2010	DA	SDE	Nancy		70000

```
(4 rows)
cqlsh:employees>
```

Task I.8:

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

```
BEGIN BATCH INSERT INTO employee_info(emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (124, 'Maria', 'Team Lead', '05/07/2008', 60000, 'SE') USING TTL 15;

APPLY BATCH;

SELECT TTL(designation) FROM employee_info WHERE emp_id = 124;
```

```
cqlsh:employees> BEGIN BATCH INSERT INTO employee_info(emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (124, 'Maria', 'Team Lead', '05/07/2008', 60000, 'SE') USING TTL 15;
...
... APPLY BATCH;
cqlsh:employees>
cqlsh:employees> SELECT TTL(designation) FROM employee_info WHERE emp_id = 124;
```

ttdesignation)
14

```
(1 rows)
cqlsh:employees>
```

Task II.1:

Create a keyspace by name Library

```
CREATE KEYSPACE Library WITH replication = { 'class': 'SimpleStrategy', 'replication_factor': 3};
```

Task II.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

```
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));
```