

1BM22CS158_Lab-5_BDA_cassandra

Part 1: Employee Database (Table name changed to **employee_details**)

#Create Keyspace

#Create Table **employee_details**

#Insert Values in Batch

```
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> CREATE KEYSPACE employee_db WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
AlreadyExists: Keyspace 'employee_db' already exists
cqlsh> USE employee_db;
cqlsh:employee_db> CREATE TABLE employee_details (
...     emp_id int PRIMARY KEY,
...     emp_name text,
...     designation text,
...     date_of_joining date,
...     salary decimal,
...     dept_name text
... );
cqlsh:employee_db> BEGIN BATCH
... INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name)
... VALUES (121, 'Alice', 'Manager', '2018-05-20', 75000.00, 'HR');
... INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name)
... VALUES (122, 'Bob', 'Developer', '2020-07-01', 60000.00, 'IT');
... INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name)
... VALUES (123, 'Charlie', 'Analyst', '2021-09-10', 50000.00, 'Finance');
... APPLY BATCH;
cqlsh:employee_db> select * from employee_details
... ;

emp_id | date_of_joining | dept_name | designation | emp_name | salary
-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | 60000.00
121 | 2018-05-20 | HR | Manager | Alice | 75000.00
(3 rows)
cqlsh:employee_db> 
```

#Update emp_name and dept_name of emp_id 121

#Sort Employees by Salary

#Alter Table to Add projects Column

#Update Projects for Employee

```
cqlsh:employee_db> UPDATE employee_details
... SET emp_name = 'Alicia', dept_name = 'Operations'
... WHERE emp_id = 121;
cqlsh:employee_db> select * from employee_details ;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
123	2021-09-10	Finance	Analyst	Charlie	50000.00
122	2020-07-01	IT	Developer	Bob	60000.00
121	2018-05-20	Operations	Manager	Alicia	75000.00

(3 rows)

```
cqlsh:employee_db> -- This needs ALLOW FILTERING as sorting isn't supported directly:
cqlsh:employee_db> SELECT * FROM employee_details WHERE salary > 0 ALLOW FILTERING;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
123	2021-09-10	Finance	Analyst	Charlie	50000.00
122	2020-07-01	IT	Developer	Bob	60000.00
121	2018-05-20	Operations	Manager	Alicia	75000.00

(3 rows)

```
cqlsh:employee_db> ALTER TABLE employee_details ADD projects set<text>;
```

#Update Projects for Employee

#Insert with TTL of 15 Seconds

```

cqlsh:employee_db> UPDATE employee_details
... SET projects = {'Project A', 'Project B'}
... WHERE emp_id = 121;
cqlsh:employee_db> INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name)
... VALUES (124, 'Eve', 'Intern', '2023-01-15', 30000.00, 'Marketing') USING TTL 15;
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00

(3 rows)
cqlsh:employee_db> INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (124, 'Eve', 'Intern', '2023-01-15', 30000.00, 'Marketing')
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00
124 | 2023-01-15 | Marketing | Intern | Eve | null | 30000.00

(4 rows)
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00
124 | 2023-01-15 | Marketing | Intern | Eve | null | 30000.00

(4 rows)
cqlsh:employee_db> INSERT INTO employee_details (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (124, 'Eve', 'Intern', '2023-01-15', 30000.00, 'Marketing') USING TTL 100;
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00
124 | 2023-01-15 | Marketing | Intern | Eve | null | 30000.00

(4 rows)
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00
124 | 2023-01-15 | Marketing | Intern | Eve | null | 30000.00

(4 rows)
cqlsh:employee_db> select * from employee_details ;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
123 | 2021-09-10 | Finance | Analyst | Charlie | null | 50000.00
122 | 2020-07-01 | IT | Developer | Bob | null | 60000.00
121 | 2018-05-20 | Operations | Manager | Alicia | ('Project A', 'Project B') | 75000.00

```

Part 2: Library Database (New Table Names)

Create Keyspace

Create Tables

A. Table: `library_student_info`

B. Table: `book_counter_info`

3 Insert Data in Batch

```

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> CREATE KEYSPACE library_db WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE library_db;
cqlsh:library_db> CREATE TABLE library_student_info (
...     stud_id int PRIMARY KEY,
...     stud_name text,
...     book_name text,
...     book_id int,
...     date_of_issue date
... );
cqlsh:library_db> CREATE TABLE book_counter_info (
...     stud_id int,
...     book_name text,
...     counter_value counter,
...     PRIMARY KEY (stud_id, book_name)
... );
cqlsh:library_db> BEGIN BATCH
... INSERT INTO library_student_info (stud_id, stud_name, book_name, book_id, date_of_issue)
... VALUES (112, 'David', 'BDA', 401, '2024-03-12');
... UPDATE book_counter_info SET counter_value = counter_value + 1
... WHERE stud_id = 112 AND book_name = 'BDA';
... APPLY BATCH;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Counter and non-counter mutations cannot exist in the same batch"

```

(No Mixing Counters in Batch):

You can repeat the **UPDATE** if you want to increment the counter multiple times.

To Simulate Borrowing Book “BDA” 2 Times by Student 112

Display Table & Increase Counter

Query: Student 112 took “BDA” 2 times

```

cqlsh:library_db> -- First: Insert normal data (non-counter)
cqlsh:library_db> INSERT INTO library_student_info (stud_id, stud_name, book_name, book_id, date_of_issue)
... VALUES (112, 'David', 'BDA', 401, '2024-03-12');
cqlsh:library_db> -- Then: Update the counter table separately
cqlsh:library_db> UPDATE book_counter_info
... SET counter_value = counter_value + 1
... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> -- Insert once (already done above)
cqlsh:library_db> -- Increment counter again
cqlsh:library_db> UPDATE book_counter_info
... SET counter_value = counter_value + 1
... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> SELECT * FROM library_student_info;

```

stud_id	book_id	book_name	date_of_issue	stud_name
112	401	BDA	2024-03-12	David

(1 rows)

```

cqlsh:library_db> SELECT * FROM book_counter_info;

```

stud_id	book_name	counter_value
112	BDA	2

(1 rows)

```

cqlsh:library_db>
cqlsh:library_db> -- Increment counter again:
cqlsh:library_db> UPDATE book_counter_info SET counter_value = counter_value + 1
... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> SELECT counter_value FROM book_counter_info
... WHERE stud_id = 112 AND book_name = 'BDA';

```

counter_value
3

Export Table to CSV

```

(1 rows)
cqlsh:library_db> cqlsh -e "COPY library_db.library_student_info TO 'library_export.csv' WITH HEADER = TRUE;"

```

```

cqlsh:library_db> cqlsh> COPY library_student_info (stud_id, stud_name, book_name, book_id, date_of_issue) FROM '/home/bmscece/documents/library_data.csv' WITH HEADER = TRUE;

```