

Cassandra-2

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.

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

```
CREATE TABLE IF NOT EXISTS Employee.Employee_Info (  
    Emp_Id INT PRIMARY KEY,  
    Emp_name TEXT,  
    Designation TEXT,  
    date_of_joining DATE,  
    Salary FLOAT,  
    Dep_name TEXT,  
    Projects SET<TEXT>  
);  
  
USE Employee;
```

```
CREATE TABLE IF NOT EXISTS Employee_Info (Emp_Id INT PRIMARY KEY, Emp_name TEXT, designation  
TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET<TEXT>);
```

```
UPDATE employee_info USING TTL 15 SET salary = 0 WHERE emp_id = 121;
```

UPDATE employee_info SET emp_name = 'Priyanka GH' WHERE emp_id = 120;

UPDATE employee_info SET bonus = 12000 WHERE emp_id = 120;

UPDATE employee_info USING TTL 15 SET salary = 0 WHERE emp_id = 121;

```
binance@binance-HP-Elite-Tower-800-G5-Desktop-PC: $ cqlsh
Connected to binance 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 with replication = ('class': 'SimpleStrategy', 'replication_factor': '1');
cqlsh> create keyspace Employee WITH replication = ('class': 'SimpleStrategy', 'replication_factor': '1');
cqlsh> create keyspace Employee WITH replication = ('class': 'SimpleStrategy', 'replication_factor': '1');
cqlsh> DESCRIBE KEYSPACES
keyspace system_auth system_schema system_views
system system_distributed system_traces system_virtual_schema
cqlsh> CREATE TABLE IF NOT EXISTS Employee_Info(
... Emp_id INT PRIMARY KEY,
... Emp_name TEXT,
... designation TEXT,
... date_of_joining DATE,
... Salary FLOAT,
... Dep_name TEXT,
... Projects SET(TEXT));
[UnavailableError: From server: code=2200 [Invalid query] message="No keyspace has been specified. Use a keyspace, or explicitly specify keyspace.tablename"]
cqlsh> USE Employee
cqlsh> USE Employee;
cqlsh> employee> CREATE TABLE IF NOT EXISTS Employee_Info( Emp_id INT PRIMARY KEY, Emp_name TEXT, designation TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET(TEXT));
cqlsh> employee> describe keyspace Employee
CREATE KEYSPACE employee WITH replication = ('class': 'SimpleStrategy', 'replication_factor': '1') AND durable_writes = true;
CREATE TABLE employee.employee_info (
  emp_id int PRIMARY KEY,
  date_of_joining date,
  dep_name text,
  designation text,
  emp_name text,
  salary float,
  projects set(text)
) WITH additional_write_policy = '99p'
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
```

```
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;
```

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	12000	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	11000	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	0

(4 rows)

```
cqlsh:employee> select * from employee_info;
```

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	12000	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	11000	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	null

(4 rows)

```
cqlsh:employee>
```

```

AND speculative_retry = '99p';
cqsh:employee> select * from employee_info;

emp_id | date_of_joining | dep_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
120 | 2024-05-06 | Engineering | Developer | Priyanka | ('Project B', 'ProjectA') | 1e+06
123 | 2024-05-07 | Engineering | Engineer | Sadhana | ('Project M', 'Project P') | 1.2e+06
122 | 2024-05-06 | Management | HR | Rachana | ('Project C', 'Project N') | 9e+05
121 | 2024-05-06 | Management | Developer | Shreya | ('Project C', 'ProjectA') | 9e+05
(4 rows)

cqsh:employee> update employee_info set emp_name = 'Priyanka GH' where emp_id = '120';
ERROR: Error from server: code=2200 [Invalid query] message=Invalid string constant (120) for "emp_id" of type int
cqsh:employee> update employee_info set emp_name = 'Priyanka GH' where emp_id=120;
cqsh:employee> select * from employee_info;

emp_id | date_of_joining | dep_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----
120 | 2024-05-06 | Engineering | Developer | Priyanka GH | ('Project B', 'ProjectA') | 1e+06
123 | 2024-05-07 | Engineering | Engineer | Sadhana | ('Project M', 'Project P') | 1.2e+06
122 | 2024-05-06 | Management | HR | Rachana | ('Project C', 'Project N') | 9e+05
121 | 2024-05-06 | Management | Developer | Shreya | ('Project C', 'ProjectA') | 9e+05
(4 rows)

cqsh:employee> select * from employee_info order by salary;
ERROR: 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."
cqsh:employee> alter table employee_info add bonus INT;
cqsh:employee> select * from employee_info;

emp_id | bonus | date_of_joining | dep_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----|-----
120 | null | 2024-05-06 | Engineering | Developer | Priyanka GH | ('Project B', 'ProjectA') | 1e+06
123 | null | 2024-05-07 | Engineering | Engineer | Sadhana | ('Project M', 'Project P') | 1.2e+06
122 | null | 2024-05-06 | Management | HR | Rachana | ('Project C', 'Project N') | 9e+05
121 | null | 2024-05-06 | Management | Developer | Shreya | ('Project C', 'ProjectA') | 9e+05
(4 rows)

cqsh:employee> update employee_info set bonus = 12000 where emp_id = 120;
cqsh:employee> select * from employee_info;

emp_id | bonus | date_of_joining | dep_name | designation | emp_name | projects | salary
-----|-----|-----|-----|-----|-----|-----|-----
120 | 12000 | 2024-05-06 | Engineering | Developer | Priyanka GH | ('Project B', 'ProjectA') | 1e+06
123 | null | 2024-05-07 | Engineering | Engineer | Sadhana | ('Project M', 'Project P') | 1.2e+06
122 | null | 2024-05-06 | Management | HR | Rachana | ('Project C', 'Project N') | 9e+05
121 | null | 2024-05-06 | Management | Developer | Shreya | ('Project C', 'ProjectA') | 9e+05
(4 rows)

cqsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
cqsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
ERROR: Error from server: code=2200 [Invalid query] message="cannot select from employee_info [using] ttl..."
cqsh:employee> select * from employee_info where emp_id = 121 using ttl 15;
ERROR: Error from server: code=2200 [Invalid query] message="cannot select from employee_info where emp_id = 121 [using]..."
cqsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqsh:employee> select * from employee_info;

```