Program 1. Perform the following DB operations using Cassandra.

```
bmsce@bmsce-Precision-T1700:~/cassandra/apache-cassandra-3.11.0/bin$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
```

1. Create a key space by name Employee

```
cqlsh> create keyspace Employee with REPLICATION ={
   ... 'class':'SimpleStrategy','replication_factor':1
   ... };

cqlsh> use Employee;
cqlsh:employee> describe keyspaces;

students       system_auth  system_distributed  system_traces
system_schema  system       employee

cqlsh> describe keyspace employee;

CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_writes = true;
```

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 text,
            ... designation text,
            ... date_of_joining timestamp,
            ... salary double,
            ... dept_name text
            ... );
cqlsh:employee> describe tables

employee_info
```

```
cqlsh:employee> describe table employee_info

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    date_of_joining timestamp,
    dept_name text,
    designation text,
    emp_name text,
    salary double
) 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
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
```

3. Insert the values into the table in batch

```
cqlsh:employee> BEGIN BATCH
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(1,'Arun','Technical head','2020-03-01',50000,'Technical')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(2,'Ajay','HR manager','2020-06-11',60000,'HR')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(3,'Riya','Editor','2022-01-11',22000,'Markrting')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(4,'Kshma','Software Engineer','2021-05-11',35000,'Technical')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(5,'Ram','HR employee','2021-02-11',25000,'HR')
           ... APPLY BATCH;
```

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

```
 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 | Markrting |            Editor |     Riya |  22000
```

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

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 | Markrting |            Editor |     Riya |  22000

(5 rows)
```

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

cqlsh:employee> UPDATE employee_info SET emp_name = 'Raj' , dept_name = 'Sales' where emp_id = 3;

cqlsh:employee> select * from employee_info;

```
 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 |     Sales |            Editor |      Raj |  22000
```

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

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 |     Sales |            Editor |      Raj |  22000

(5 rows)
cqlsh:employee>
```

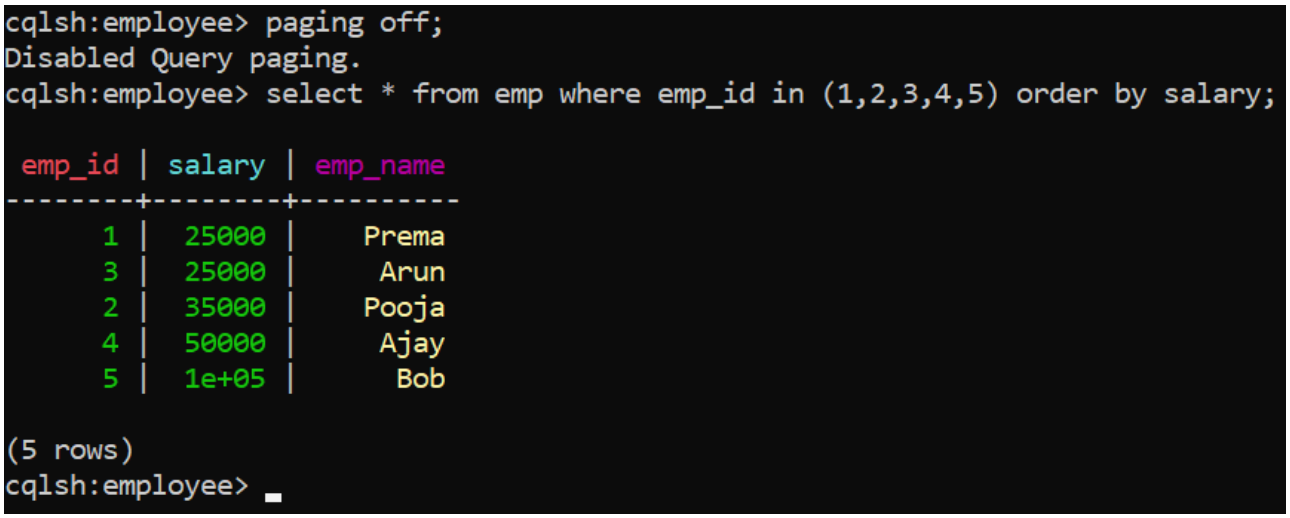
5. Sort the details of Employee records based on salary

```
CREATE TABLE emp(
        ...  emp_id int,
        ... salary double,
        ... emp_name text,
        ... PRIMARY KEY(emp_id,salary));
```

```
BEGIN BATCH
... insert into emp(emp_id,emp_name,salary) values(1,'Prema',25000)
... insert into emp(emp_id,emp_name,salary) values(2,'Pooja',35000)
... insert into emp(emp_id,emp_name,salary) values(3,'Arun',25000)
... insert into emp(emp_id,emp_name,salary) values(4,'Ajay',50000)
... insert into emp(emp_id,emp_name,salary) values(5,'Bob',100000)
... APPLY BATCH;
```

PAGING OFF;
select * from emp where emp_id in(1,2,3,4,5) order by salary;

```
 emp_id | salary | emp_name
--------+--------+----------
      1 |  25000 |    Prema
      3 |  25000 |     Arun
      2 |  35000 |    Pooja
      4 |  50000 |     Ajay
      5 |  1e+05 |      Bob
```



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.

```
cqlsh:employee> alter table employee_info
           ... add project text;
cqlsh:employee> select * from employee_info;
```

```
 emp_id | date_of_joining                 | dept_name | designation       | emp_name | project | salary
--------+---------------------------------+-----------+-------------------+----------+---------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |    null |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |    null |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |    null |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |    null |  35000
```

3 | 2022-01-10 18:30:00.000000+0000 | Sales | Editor | Raj | null | 22000

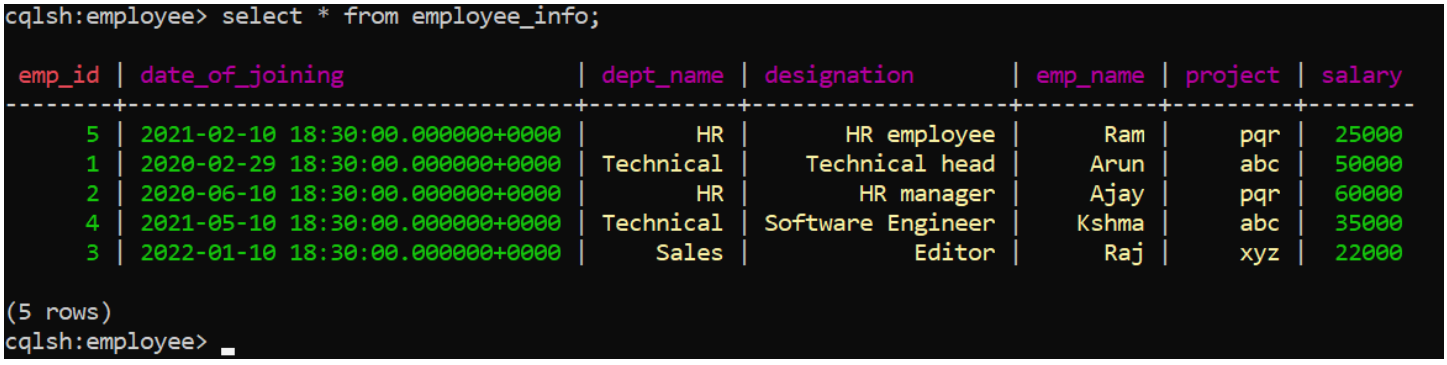
(5 rows)

7. Update the altered table to add project names.

```
cqlsh:employee> begin batch
           ... update employee_info set project = 'xyz' where emp_id = 3
           ... update employee_info set project = 'pqr' where emp_id = 5
           ... update employee_info set project = 'pqr' where emp_id = 2
           ... update employee_info set project = 'abc' where emp_id = 1
           ... update employee_info set project = 'abc' where emp_id = 4
           ... apply batch;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | project | salary
--------+---------------------------------+-----------+-------------------+----------+---------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |     pqr |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |     abc |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |     pqr |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |     abc |  35000
      3 | 2022-01-10 18:30:00.000000+0000 |     Sales |            Editor |      Raj |     xyz |  22000

(5 rows)
```



8 Create a TTL of 15 seconds to display the values of Employee

```
cqlsh:employee> insert into employee_info(emp_id,
date_of_joining,dept_name,designation,emp_name,project,salary)  values(6, '2021-02-28','HR','HR
employee','Anvi','xyz',20000) using TTL 15;
cqlsh:employee> select TTL(emp_name) from employee_info;

 ttl(emp_name)
---------------
```

null
null
null
null
5
null

(6 rows)

```
cqlsh:employee> select TTL(emp_name) from employee_info;

 ttl(emp_name)
---------------
          null
          null
          null
          null
             5
          null

(6 rows)
```