NIRANJAN V S

1BM17CS054

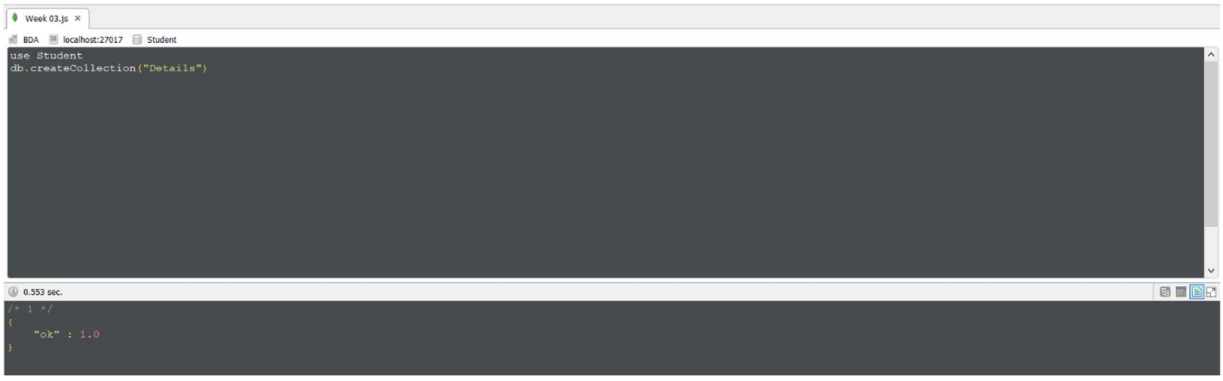
BIG DATA ANALYTICS LAB RECORD

21 December 2020

Perform the following DB operations using MongoDB.

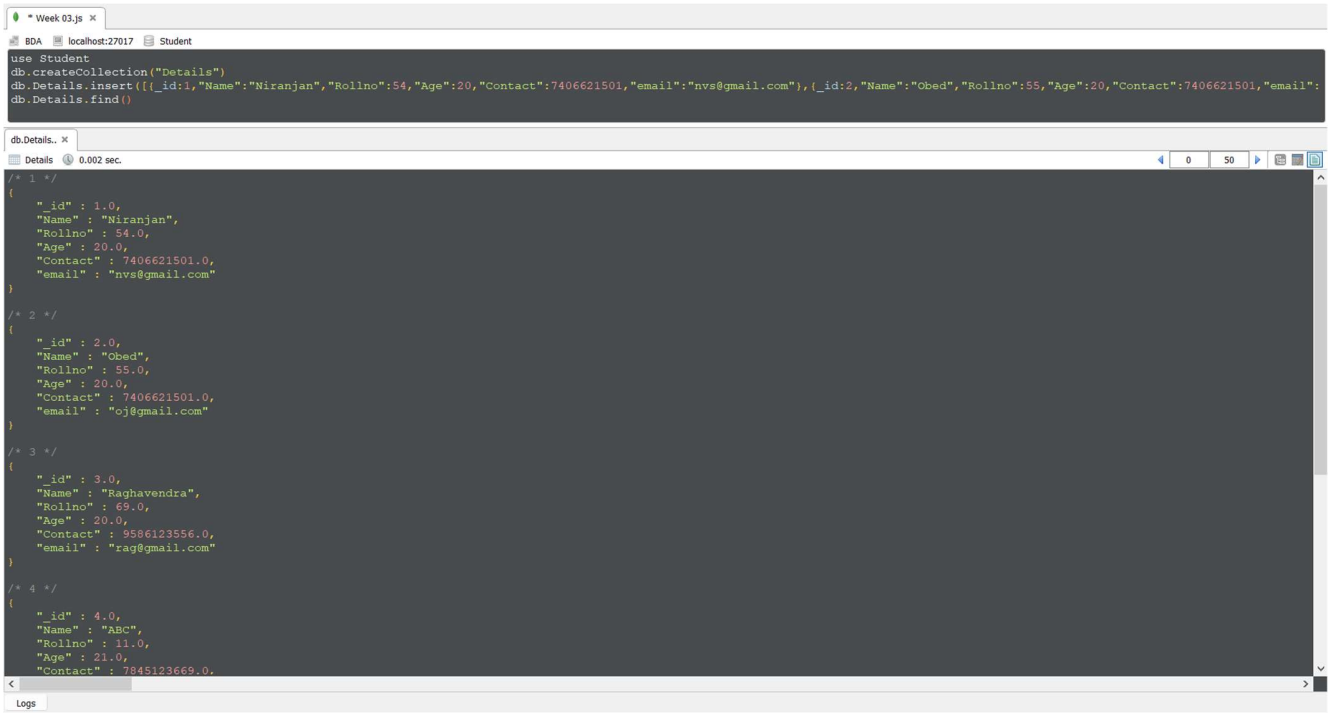
1. **Create a database “Student” with the following attributes Roll no, Age, Contact No, Email-Id.**

```
use Student
db.createCollection("Details")
```



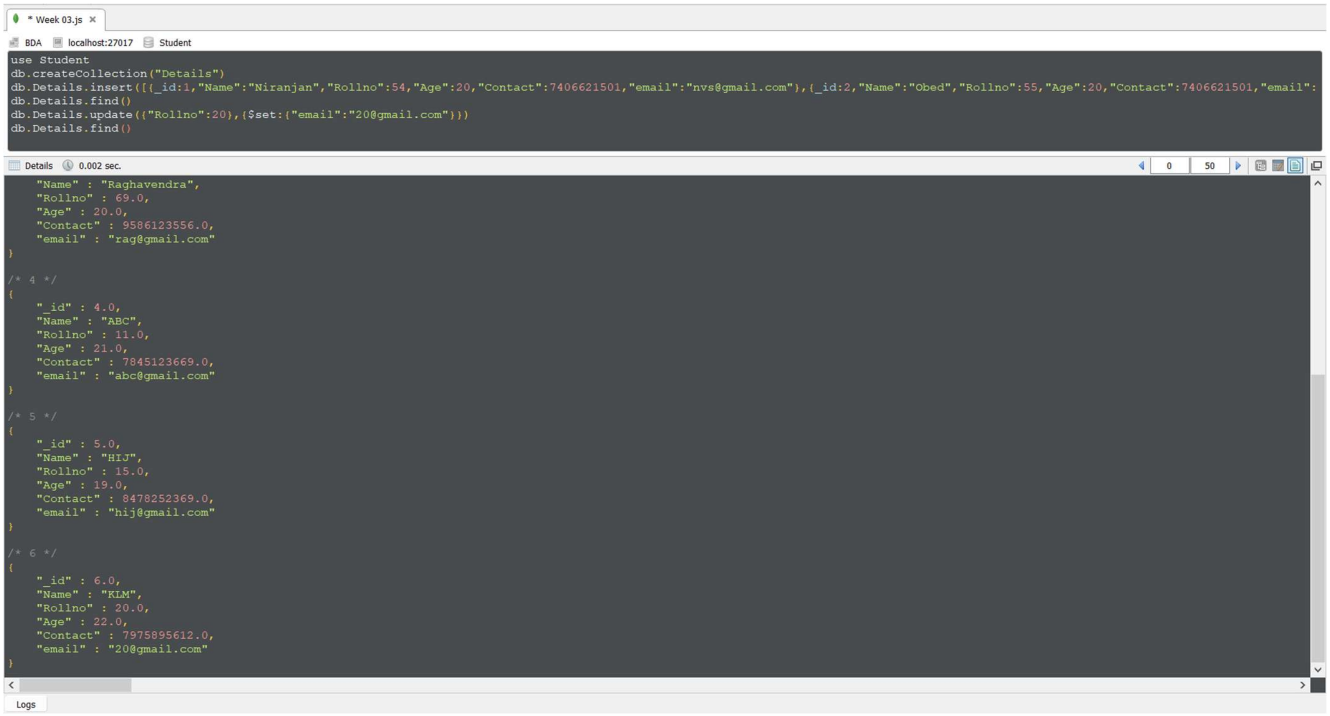
2. **Insert appropriate values**

```
db.Details.insert([{_id:1,"Name":"Niranjan","Rollno":54,"Age":20,"Contact":7406621501,"email":"nvs@gmail.com"},{_id:2,"Name":"Obed","Rollno":55,"Age":20,"Contact":7406621501,"email":"oj@gmail.com"},{_id:3,"Name":"Raghavendra","Rollno":69,"Age":20,"Contact":9586123556,"email":"rag@gmail.com"},{_id:4,"Name":"ABC","Rollno":11,"Age":21,"Contact":7845123669,"email":"abc@gmail.com"},{_id:5,"Name":"HIJ","Rollno":15,"Age":19,"Contact":8478252369,"email":"hij@gmail.com"},{_id:6,"Name":"KLM","Rollno":20,"Age":22,"Contact":7975895612,"email":"klm@gmail.com"}])
```



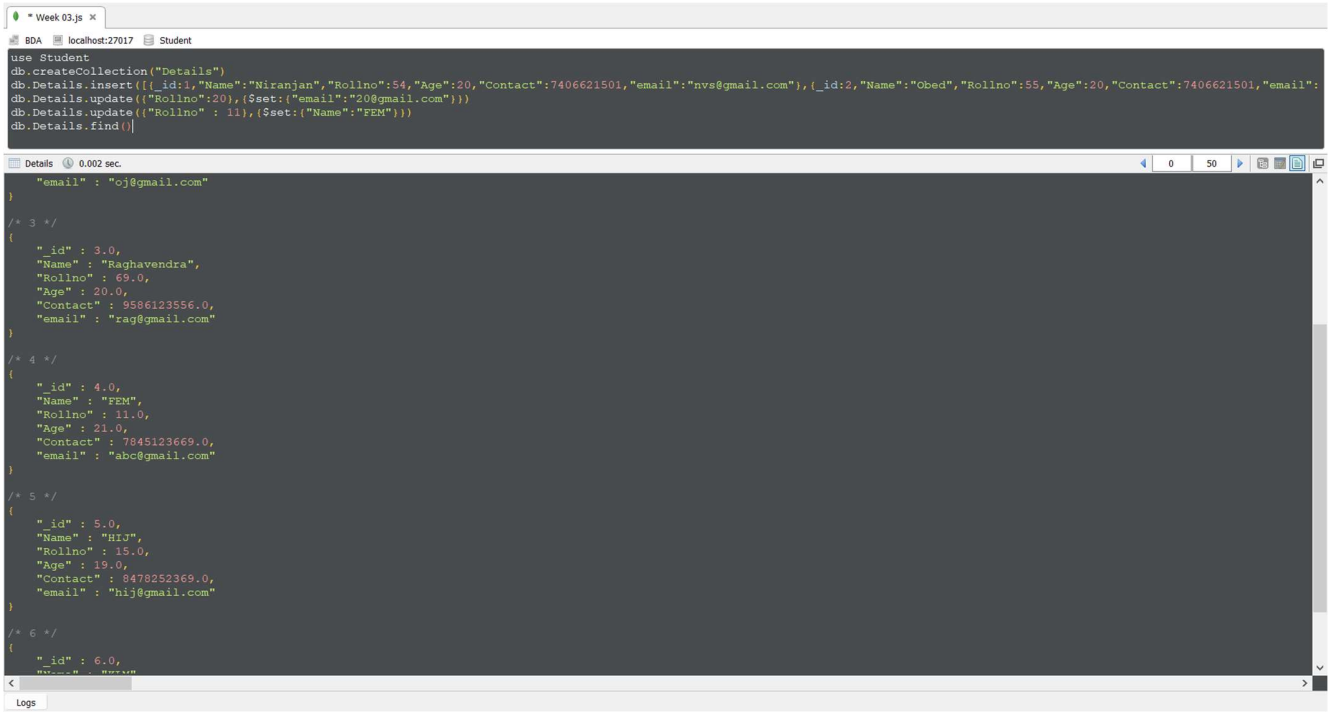
3. **Write query to update Email-Id of a student with roll no 20.**

db.Details.update({"Rollno":20},{$set:{"email":"20@gmail.com"}})



4. **Replace the student name from “ABC” to “FEM” of roll no 11.**

db.Details.update({"Rollno" : 11},{$set:{"Name":"FEM"}})



5. **Export the created table into local file system**

mongoexport -c Details -d Student –o Student.csv

```
D:\GIT\1BM17CS054-BDA>mongoexport -c Details -d Student -o Student.json
2020-10-08T15:38:34.362+0530    connected to: mongodb://localhost/
2020-10-08T15:38:34.430+0530    exported 6 records

D:\GIT\1BM17CS054-BDA>
```

Student.json | Student.csv

D: > GIT > 1BM17CS054-BDA > Student.csv

```
{"_id":1.0,"Name":"Niranjan","Rollno":54.0,"Age":20.0,"Contact":7.406621501E+09,"email":"nvs@gmail.com"}
{"_id":2.0,"Name":"Obed","Rollno":55.0,"Age":20.0,"Contact":7.406621501E+09,"email":"oj@gmail.com"}
{"_id":3.0,"Name":"Raghavendra","Rollno":69.0,"Age":20.0,"Contact":9.586123556E+09,"email":"rag@gmail.com"}
{"_id":4.0,"Name":"FEM","Rollno":11.0,"Age":21.0,"Contact":7.845123669E+09,"email":"abc@gmail.com"}
{"_id":5.0,"Name":"HIJ","Rollno":15.0,"Age":19.0,"Contact":8.478252369E+09,"email":"hij@gmail.com"}
{"_id":6.0,"Name":"KLM","Rollno":20.0,"Age":22.0,"Contact":7.975895612E+09,"email":"20@gmail.com"}
```

6. **Drop the table**

db.Details.drop()

7. **Import a given csv dataset from local file system into mongo dB collection.**

mongoimport -c Details -d Student –-file Student.csv

```
D:\GIT\1BM17CS054-BDA>mongoimport -c Details -d Student --file Student.csv
2020-10-08T15:42:16.402+0530    connected to: mongodb://localhost/
2020-10-08T15:42:16.674+0530    6 document(s) imported successfully. 0 document(s) failed to import.

D:\GIT\1BM17CS054-BDA>
```

Perform the following DB operations using MongoDB.

1. **Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type**

   use Customer

   db.createCollection("Details")

2. **Insert at least 5 values into the table**

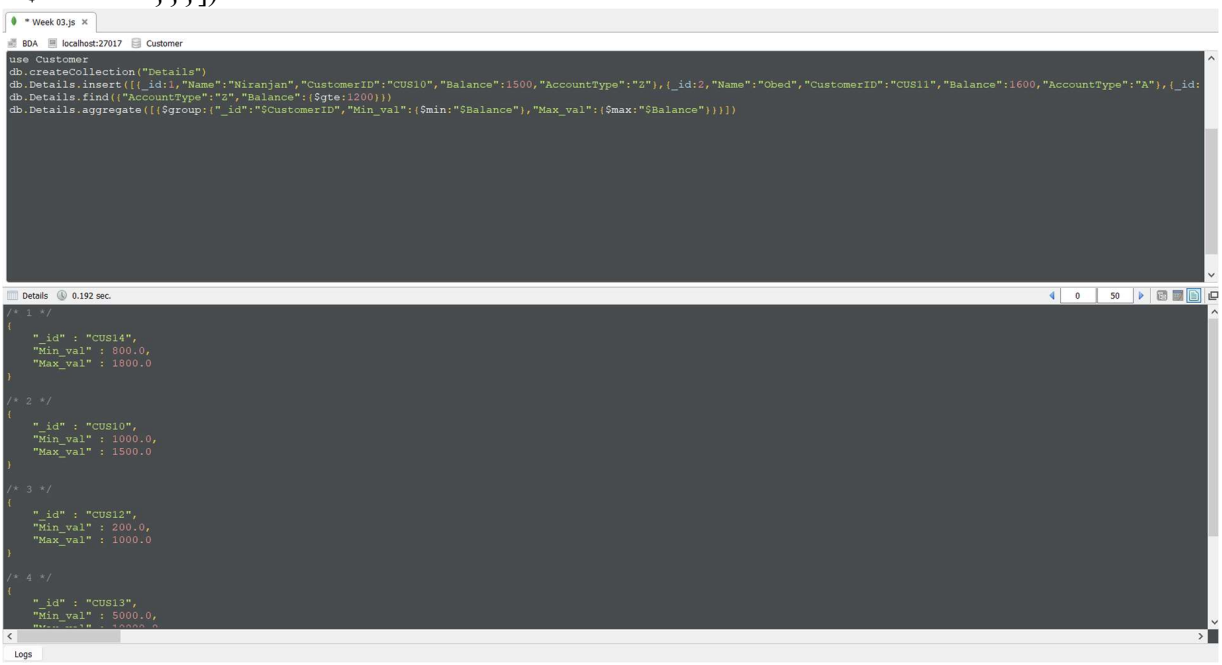
   db.Details.insert([{_id:1,"Name":"Niranjan","CustomerID":"CUS10","Balance":1500,"AccountType":"Z"},{_id:2,"Name":"Obed","CustomerID":"CUS11","Balance":1600,"AccountType":"A"},{_id:3,"Name":"Raghavendra","CustomerID":"CUS12","Balance":1000,"AccountType":"Z"},{_id:4,"Name":"Shreyas","CustomerID":"CUS13","Balance":10000,"AccountType":"A"},{_id:5,"Name":"Tarun","CustomerID":"CUS14","Balance":800,"AccountType":"Z"}])

3. **Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.**

   db.Details.find({"AccountType":"Z","Balance":{$gte:1200}})

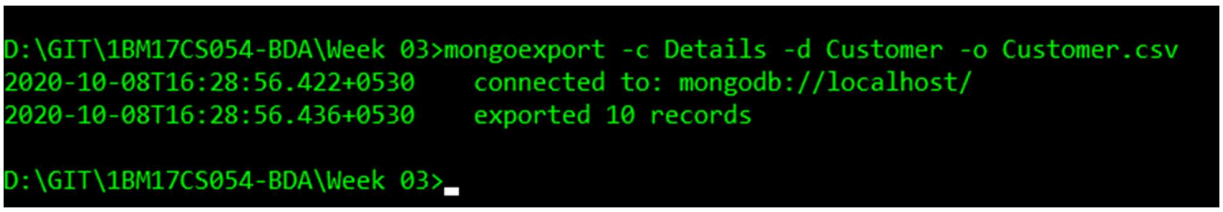
4. **Determine Minimum and Maximum account balance for each customer_id.**

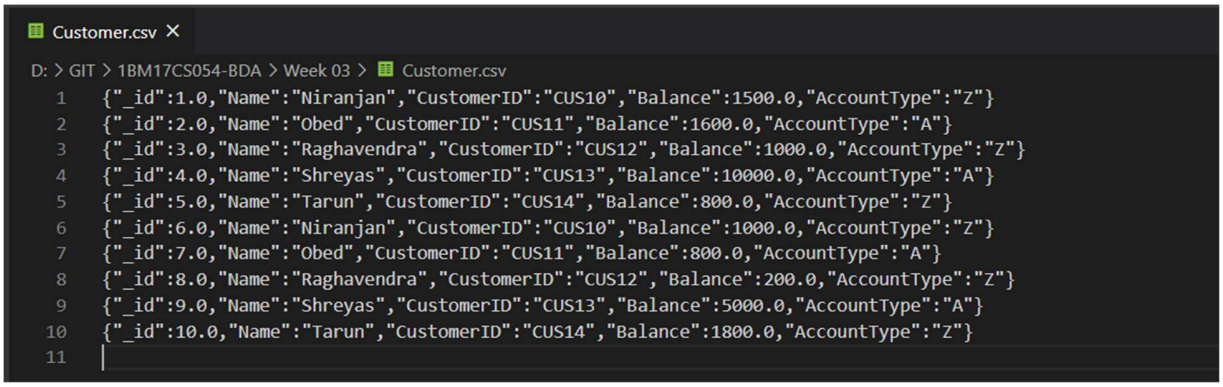
db.Details.aggregate([{$group:{"_id":"$CustomerID","Min_val":{$min:"$Balance"},"Max_val":{$max:"$Balance"}}}])



5. **Export the created collection into local file system**

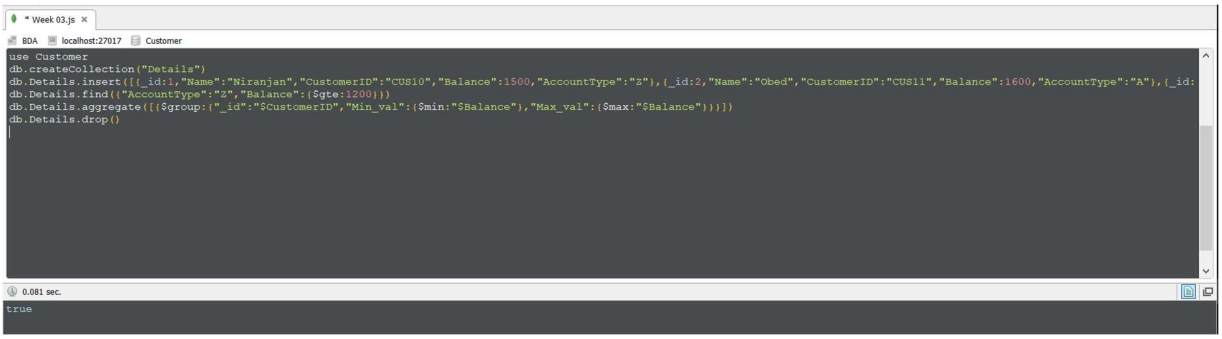
mongoexport -c Details -d Customer –o Customer.csv





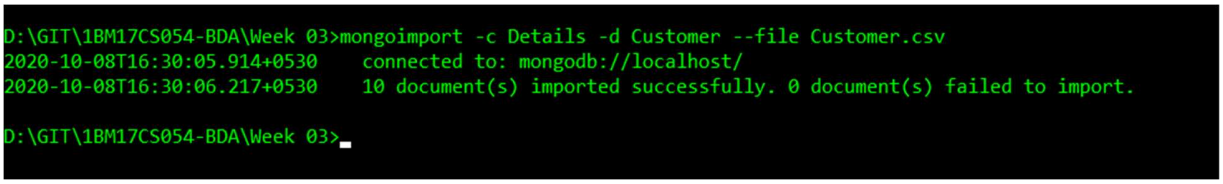
6. Drop the table

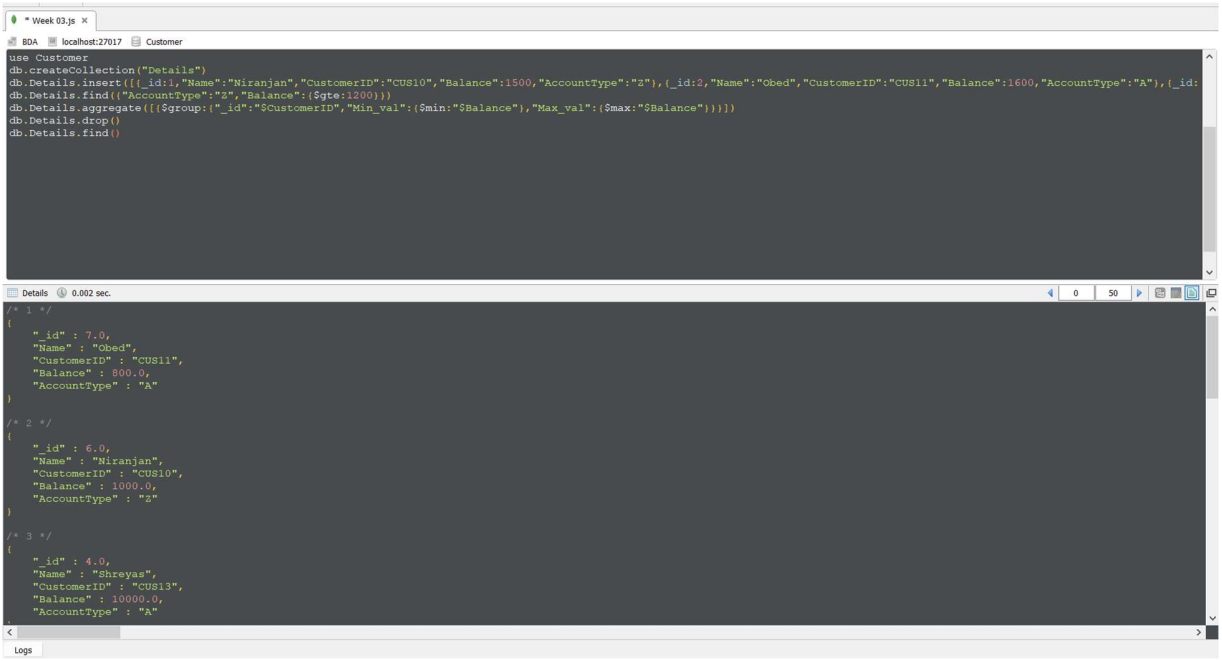
db.Details.drop()



7. Import a given csv dataset from local file system into mongodb collection.

mongoimport -c Details -d Customer –-file Customer.csv





Perform the following DB operations using Cassandra.

1. **Create a keyspace by name Employee**

   CREATE KEYSPACE employee WITH REPLICATION = {'class':'SimpleStrategy','replication_factor':1};

   USE 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**

   CREATE TABLE employee_info (employee_id int, employee_name text,designation text, date_of_joining timestamp, salary double,department_name text, PRIMARY KEY(employee_id, salary));

3. **Insert the values into the table in batch**

   BEGIN BATCH

   INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(117,'Niranjan V S','Software Developer','2020-10-19',60000,'Development')

   INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(118,'Obed Junias','Software Developer in Test','2020-01-15',55000,'Testing')

   INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(119,'Raghavendra','Operations Lead','2018-04-17',90000,'Operations')

   INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(120,'Tarun M Krishna','New Talent Recruiter','2019-02-03',65000,'Recruitment')

   INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(121,'Ranveer Singh','HR','2019-12-23',62000,'HR')

   APPLY BATCH;

   SELECT * FROM employee_info;

4. **Update Employee name and Department of Emp-Id 121**

   UPDATE employee_info SET employee_name = 'Shreyas K' , department_name='Sales' WHERE employee_id=121;

   SELECT * FROM employee_info;

5. **Sort the details of Employee records based on salary**

   SELECT * FROM employee_info WHERE employee_id IN (117,118,119,120,121) ORDER BY 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.**

   ALTER TABLE employee.employee_info ADD projects set<text>;

```
cqlsh> CREATE KEYSPACE employee WITH REPLICATION = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> USE employee;
cqlsh:employee> CREATE TABLE employee_info (employee_id int PRIMARY KEY, employee_name text,designation text, date_of_joining timestamp, salary double,department_name text);
cqlsh:employee> BEGIN BATCH
           ... INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(117,'Niranjan V S','Software Developer','2020-10-19',60000,'Development')
           ... INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(118,'Obed Junias','Software Developer in Test','2020-01-15',55000,'Testing')
           ... INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(119,'Raghavendra','Operations Lead','2018-04-17',90000,'Operations')
           ... INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(120,'Tarun M Krishna','New Talent Recruiter','2019-02-03',65000,'Recruitment'
)
           ... INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(121,'Ranveer Singh','HR','2019-12-23',62000,'HR')
           ... APPLY BATCH;
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation                | employee_name   | salary
-------------+---------------------------------+-----------------+----------------------------+-----------------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |          Software Developer |    Niranjan V S |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias |  55000
         121 | 2019-12-22 18:30:00.000000+0000 |              HR |                         HR |   Ranveer Singh |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |             Operations Lead |     Raghavendra |  90000

(5 rows)
cqlsh:employee> UPDATE employee_info SET employee_name = 'Shreyas K' , department_name='Sales' WHERE employee_id=121;
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation                | employee_name   | salary
-------------+---------------------------------+-----------------+----------------------------+-----------------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |          Software Developer |    Niranjan V S |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias |  55000
         121 | 2019-12-22 18:30:00.000000+0000 |           Sales |                         HR |       Shreyas K |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |             Operations Lead |     Raghavendra |  90000

(5 rows)
cqlsh:employee> ALTER TABLE employee.employee_info ADD projects set<text>;
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation                | employee_name   | projects | salary
-------------+---------------------------------+-----------------+----------------------------+-----------------+----------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |          Software Developer |    Niranjan V S |     null |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |     null |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias |     null |  55000
         121 | 2019-12-22 18:30:00.000000+0000 |           Sales |                         HR |       Shreyas K |     null |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |             Operations Lead |     Raghavendra |     null |  90000

(5 rows)
```

7. **Update the altered table to add project names.**

   UPDATE employee_info SET projects = projects + {'Libaray Management System'} WHERE employee_id = 117;

   UPDATE employee_info SET projects = projects + {'Student Information System'} WHERE employee_id = 118;

   UPDATE employee_info SET projects = projects + {'Student Information Management System'} WHERE employee_id = 119;

   UPDATE employee_info SET projects = projects + {'Stock Management System'} WHERE employee_id = 120;

   UPDATE employee_info SET projects = projects + {'Project Management System'} WHERE employee_id = 121;

   SELECT * FROM employee_info;

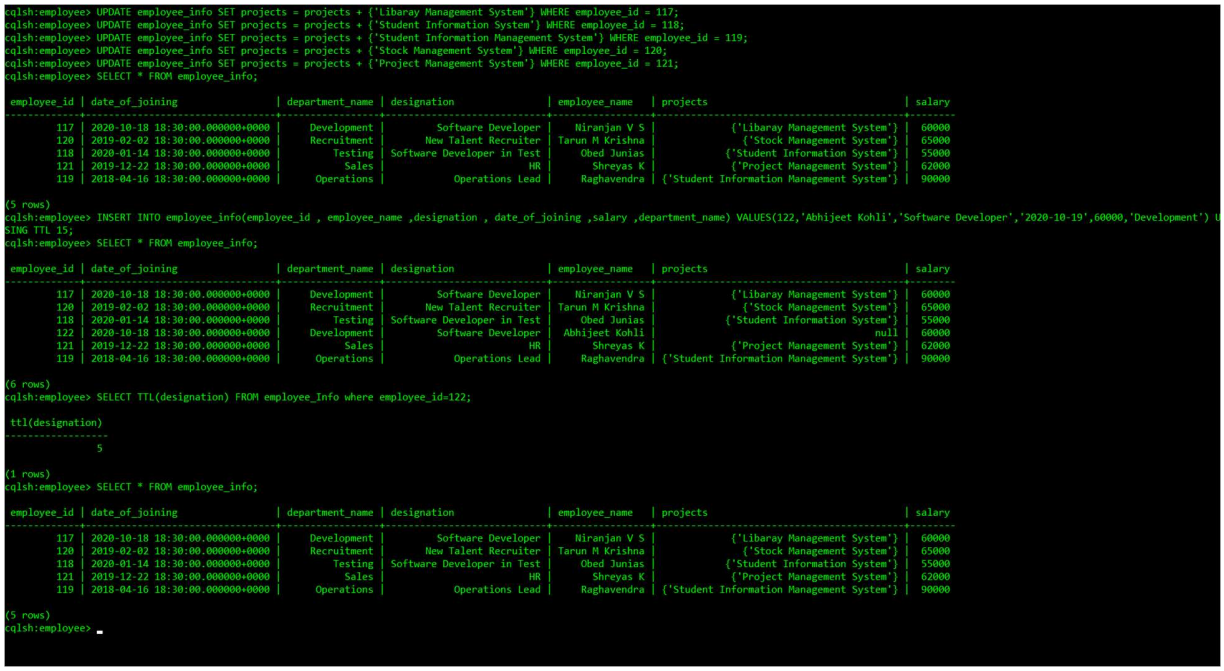
7.Create a TTL of 15 seconds to display the values of Employees.

INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(122,'Abhijeet Kohli','Software Developer','2020-10-19',60000,'Development') USING TTL 15;

SELECT * FROM employee_info;

SELECT TTL(designation) FROM employee_Info where employee_id=122;

SELECT * FROM employee_info;



```
cqlsh:employee> UPDATE employee_info SET projects = projects + {'Libaray Management System'} WHERE employee_id = 117;
cqlsh:employee> UPDATE employee_info SET projects = projects + {'Student Information System'} WHERE employee_id = 118;
cqlsh:employee> UPDATE employee_info SET projects = projects + {'Student Information Management System'} WHERE employee_id = 119;
cqlsh:employee> UPDATE employee_info SET projects = projects + {'Stock Management System'} WHERE employee_id = 120;
cqlsh:employee> UPDATE employee_info SET projects = projects + {'Project Management System'} WHERE employee_id = 121;
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation               | employee_name   | projects                                  | salary
-------------+---------------------------------+-----------------+---------------------------+-----------------+-------------------------------------------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |         Software Developer |    Niranjan V S | {'Libaray Management System'} |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |   {'Stock Management System'} |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias | {'Student Information System'} |  55000
         121 | 2019-12-22 18:30:00.000000+0000 |           Sales |                        HR |       Shreyas K |  {'Project Management System'} |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |           Operations Lead |     Raghavendra | {'Student Information Management System'} |  90000

(5 rows)
cqlsh:employee> INSERT INTO employee_info(employee_id , employee_name ,designation , date_of_joining ,salary ,department_name) VALUES(122,'Abhijeet Kohli','Software Developer','2020-10-19',60000,'Development') U
SING TTL 15;
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation               | employee_name   | projects                                  | salary
-------------+---------------------------------+-----------------+---------------------------+-----------------+-------------------------------------------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |         Software Developer |    Niranjan V S | {'Libaray Management System'} |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |   {'Stock Management System'} |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias | {'Student Information System'} |  55000
         122 | 2020-10-18 18:30:00.000000+0000 |     Development |         Software Developer |  Abhijeet Kohli |                                      null |  60000
         121 | 2019-12-22 18:30:00.000000+0000 |           Sales |                        HR |       Shreyas K |  {'Project Management System'} |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |           Operations Lead |     Raghavendra | {'Student Information Management System'} |  90000

(6 rows)
cqlsh:employee> SELECT TTL(designation) FROM employee_Info where employee_id=122;

 ttl(designation)
------------------
                5

(1 rows)
cqlsh:employee> SELECT * FROM employee_info;

 employee_id | date_of_joining                 | department_name | designation               | employee_name   | projects                                  | salary
-------------+---------------------------------+-----------------+---------------------------+-----------------+-------------------------------------------+--------
         117 | 2020-10-18 18:30:00.000000+0000 |     Development |         Software Developer |    Niranjan V S | {'Libaray Management System'} |  60000
         120 | 2019-02-02 18:30:00.000000+0000 |     Recruitment |       New Talent Recruiter | Tarun M Krishna |   {'Stock Management System'} |  65000
         118 | 2020-01-14 18:30:00.000000+0000 |         Testing | Software Developer in Test |     Obed Junias | {'Student Information System'} |  55000
         121 | 2019-12-22 18:30:00.000000+0000 |           Sales |                        HR |       Shreyas K |  {'Project Management System'} |  62000
         119 | 2018-04-16 18:30:00.000000+0000 |      Operations |           Operations Lead |     Raghavendra | {'Student Information Management System'} |  90000

(5 rows)
cqlsh:employee>
```

Perform the following DB operations using Cassandra.

1.Create a keyspace by name Library

CREATE KEYSPACE library WITH REPLICATION = {'class':'SimpleStrategy','replication_factor':1};

DESCRIBE KEYSPACES;

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

USE library;

CREATE TABLE library_info (student_id int,counter_value counter, student_name text,book_name text,book_id int,date_of_issue timestamp,PRIMARY KEY(student_id,student_name,book_name,book_id,date_of_issue));

DESCRIBE TABLES;

```
cqlsh> CREATE KEYSPACE library WITH REPLICATION = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

students       system_auth  library            employee
system_schema  system       system_distributed  system_traces

cqlsh> USE library;
cqlsh:library> CREATE TABLE library_info (student_id int,counter_value counter, student_name text,book_name text,book_id int,date_of_issue timestamp,PRIMARY KEY(student_id,student_name,book_name,book_id,date_of
issue));
cqlsh:library> DESCRIBE TABLES;

library_info

cqlsh:library> DESCRIBE TABLE library_info;

CREATE TABLE library.library_info (
    student_id int,
    student_name text,
    book_name text,
    book_id int,
    date_of_issue timestamp,
    counter_value counter,
    PRIMARY KEY (student_id, student_name, book_name, book_id, date_of_issue)
) WITH CLUSTERING ORDER BY (student_name ASC, book_name ASC, book_id ASC, date_of_issue ASC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    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_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

3. Insert the values into the table in batch

BEGIN BATCH

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=114 and student_name='Niranjan V S' AND book_name='SQM' and book_id=141 and date_of_issue='2020-11-03' ;

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=111 and student_name='Obed Junias' AND book_name='DSR' and book_id=131 and date_of_issue='2020-11-05' ;

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=113 and student_name='Raghavendra' AND book_name='BDA' and book_id=121 and date_of_issue='2020-11-05' ;

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=112 and student_name='Tarun M Krishna' AND book_name='BDA' and book_id=122 and date_of_issue='2020-10-05' ;

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=115 and student_name='Shreyas K' AND book_name='DSR' and book_id=132 and date_of_issue='2020-11-04' ;

APPLY BATCH

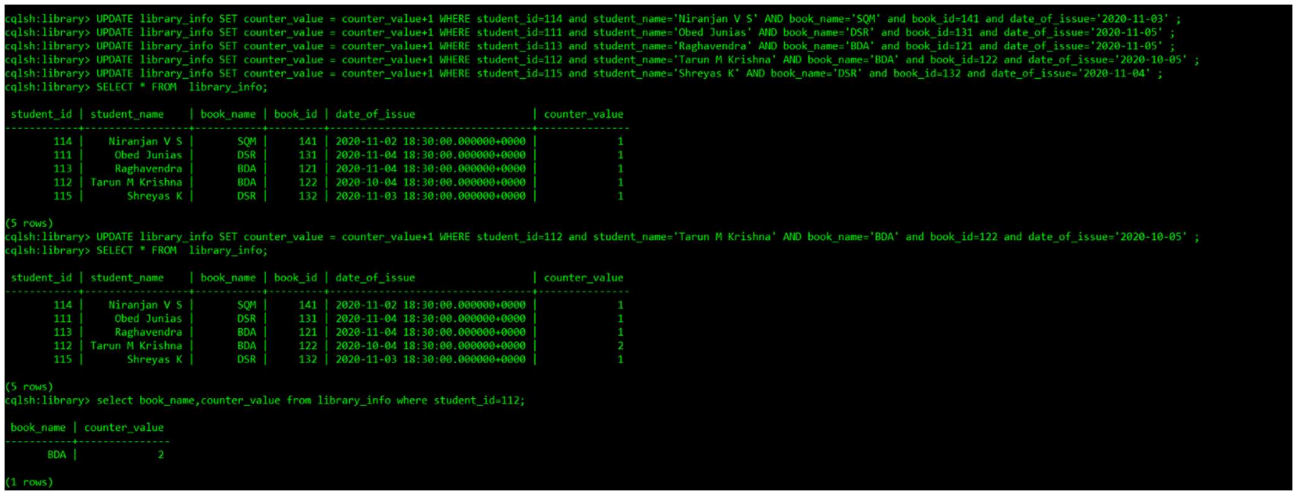
4. Display the details of the table created and increase the value of the counter

SELECT * FROM library_info;

UPDATE library_info SET counter_value = counter_value+1 WHERE student_id=112 and student_name='Tarun M Krishna' AND book_name='BDA' and book_id=122 and date_of_issue='2020-10-05' ;

5. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.

SELECT book_name,counter_value FROM library_info WHERE student_id=112;



6. Export the created column to a csv file

COPY library_info(student_id,counter_value,student_name,book_name,book_id,date_of_issue) TO '.\library_information.csv';

7. Import a given csv dataset from local file system into Cassandra column family

COPY library_info(student_id,counter_value,student_name,book_name,book_id,date_of_issue) FROM '.\library_information.csv';

SELECT * FROM library_info;

```
cqlsh:library> COPY library_info(student_id,counter_value,student_name,book_name,book_id,date_of_issue) TO '.\library_information.csv';
Using 7 child processes

Starting copy of library.library_info with columns [student_id, counter_value, student_name, book_name, book_id, date_of_issue].
Processed: 5 rows; Rate:       5 rows/s; Avg. rate:       4 rows/s
5 rows exported to 1 files in 1.258 seconds.
cqlsh:library> COPY library_info(student_id,counter_value,student_name,book_name,book_id,date_of_issue) FROM '.\library_information.csv';
Using 7 child processes

Starting copy of library.library_info with columns [student_id, counter_value, student_name, book_name, book_id, date_of_issue].
PPProcess ImportProcess-8:     5 rows/s; Avg. rate:       5 rows/s
rocess ImportProcess-9:
rocess ImportProcess-10:
TPPPTPTraceback (most recent call last):
rocess ImportProcess-11:
rocess ImportProcess-12:
rocess ImportProcess-13:
raceback (most recent call last):
rocess ImportProcess-14:
raceback (most recent call last):
 TTT T  File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
raceback (most recent call last):
raceback (most recent call last):
raceback (most recent call last):
 File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
raceback (most recent call last):
 File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
        self.run()
 File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
 File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
  self.run()
 File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
  self.run()
      File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
  self.run()
  self.run()
  self.run()
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
  self.run()
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
     File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
  self.close()
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
  self.close()
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
      self.close()
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2332, in close
  self.close()
```

```
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
  self._connection.close()
  File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
  File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
    AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
   File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
  self._connection.close()
  self._connection.close()
  AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
  AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
   AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
      AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
  File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
   cls._loop.add_timer(timer)
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
 File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
   AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
  cls._loop.add_timer(timer)
    AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
  A   cls._loop.add_timer(timer)
ttributeError: 'NoneType' object has no attribute 'add_timer'
A   cls._loop.add_timer(timer)
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
  cls._loop.add_timer(timer)
AAA  ttributeError: 'NoneType' object has no attribute 'add_timer'
ttributeError: 'NoneType' object has no attribute 'add_timer'
ttributeError: 'NoneType' object has no attribute 'add_timer'
ttributeError: 'NoneType' object has no attribute 'add_timer'
  cls._loop.add_timer(timer)
  cls._loop.add_timer(timer)
AAttributeError: 'NoneType' object has no attribute 'add_timer'
ttributeError: 'NoneType' object has no attribute 'add_timer'
Processed: 5 rows; Rate:       2 rows/s; Avg. rate:       3 rows/s
5 rows imported from 1 files in 1.809 seconds (0 skipped).
cqlsh:library> SELECT * FROM  library_info;

 student_id | student_name    | book_name | book_id | date_of_issue                   | counter_value
------------+-----------------+-----------+---------+---------------------------------+---------------
        114 |    Niranjan V S |       SQM |     141 | 2020-11-02 18:30:00.000000+0000 |             2
        111 |     Obed Junias |       DSR |     131 | 2020-11-04 18:30:00.000000+0000 |             2
        113 |     Raghavendra |       BDA |     121 | 2020-11-04 18:30:00.000000+0000 |             2
        112 | Tarun M Krishna |       BDA |     122 | 2020-10-04 18:30:00.000000+0000 |             4
        115 |       Shreyas K |       DSR |     132 | 2020-11-03 18:30:00.000000+0000 |             2

(5 rows)
cqlsh:library>
```

Develop a MapReduce program to count the number of occurrences of words in a given file.

To start all the Hadoop deamons

$ ssh localhost

$ cd Hadoop/hadoop-3.2.1

$ sbin/start-dfs.sh

$ sbin/start-yarn.sh

To create a directory in hdfs

$ hadoop fs -mkdir /rgs1

To view all the directories in hdfs

$ hadoop fs -ls /

To copy a file from local system to hdfs directory

$ Hadoop fs -copyFromLocal /home/niranjanvs/Desktop/file1.txt /rgs1/test.txt

To view all files in /rgs1 hdfs directory

$ hadoop fs -ls /rgs1

To run a MapReduce program

$ hadoop jar /home/niranjanvs/Desktop/wordcount.jar WordCount /rgs1/test.txt /rgs1/output

To view the output text

$ hadoop fs -cat /rgs1/output/part-r-00000

To stop all the Hadoop deamons

$ sbin/stop-yarn.sh

$ sbin/stop-dfs.sh

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class WordCount{
    public static class Map extends
Mapper&lt;LongWritable,Text,Text,IntWritable&gt; {
        public void map(LongWritable key, Text value,Context context) throws
IOException,InterruptedException{
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                 value.set(tokenizer.nextToken());
                 context.write(value, new IntWritable(1));
            }
        }
    }
    public static class Reduce extends
Reducer&lt;Text,IntWritable,Text,IntWritable&gt; {
        public void reduce(Text key, Iterable&lt;IntWritable&gt;
values,Context context) throws IOException,InterruptedException {
           int sum=0;
           for(IntWritable x: values)
           {
               sum+=x.get();
           }
           context.write(key, new IntWritable(sum));
        }
    }
    public static void main(String[] args) throws Exception {
        Configuration conf= new Configuration();
        Job job = new Job(conf,"My Word Count Program");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        Path outputPath = new Path(args[1]);
        //Configuring the input/output path from the filesystem into the job
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //deleting the output path automatically from hdfs so that we don't
have to delete it explicitly
        outputPath.getFileSystem(conf).delete(outputPath);
        //exiting the job only if the flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

```
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop jar /home/niranjanvs/Desktop/wordcount.jar WordCount /rgs1/test.txt /rgs1/output
2020-11-26 01:28:01,185 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-11-26 01:28:02,501 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/niranjanvs/.staging/job_1606381906540_0001
2020-11-26 01:28:02,679 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-11-26 01:28:02,860 INFO input.FileInputFormat: Total input files to process : 1
2020-11-26 01:28:03,000 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-11-26 01:28:03,056 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-11-26 01:28:03,091 INFO mapreduce.JobSubmitter: number of splits:1
2020-11-26 01:28:03,348 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-11-26 01:28:03,427 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1606381906540_0001
2020-11-26 01:28:03,427 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-11-26 01:28:03,806 INFO conf.Configuration: resource-types.xml not found
2020-11-26 01:28:03,807 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-11-26 01:28:04,299 INFO impl.YarnClientImpl: Submitted application application_1606381906540_0001
2020-11-26 01:28:04,511 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1606381906540_0001/
2020-11-26 01:28:04,512 INFO mapreduce.Job: Running job: job_1606381906540_0001
2020-11-26 01:28:22,023 INFO mapreduce.Job: Job job_1606381906540_0001 running in uber mode : false
2020-11-26 01:28:22,026 INFO mapreduce.Job:  map 0% reduce 0%
2020-11-26 01:28:30,290 INFO mapreduce.Job:  map 100% reduce 0%
2020-11-26 01:28:36,332 INFO mapreduce.Job:  map 100% reduce 100%
2020-11-26 01:28:37,348 INFO mapreduce.Job: Job job_1606381906540_0001 completed successfully
2020-11-26 01:28:37,471 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=115
                FILE: Number of bytes written=451503
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=190
                HDFS: Number of bytes written=69
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=5970
                Total time spent by all reduces in occupied slots (ms)=3671
                Total time spent by all map tasks (ms)=5970
                Total time spent by all reduce tasks (ms)=3671
                Total vcore-milliseconds taken by all map tasks=5970
                Total vcore-milliseconds taken by all reduce tasks=3671
                Total megabyte-milliseconds taken by all map tasks=6113280
                Total megabyte-milliseconds taken by all reduce tasks=3759104
        Map-Reduce Framework
                Map input records=6
                Map output records=20
                Map output bytes=169
                Map output materialized bytes=115
                Input split bytes=100
                Combine input records=20
                Combine output records=10
                Reduce input groups=10
                Reduce shuffle bytes=115
```

```
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ jps
17472 NodeManager
17111 SecondaryNameNode
17944 Jps
17311 ResourceManager
16895 DataNode
16751 NameNode
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -copyFromLocal /home/niranjanvs/Desktop/file1.txt /rgs1/test.txt
copyFromLocal: `/rgs1/test.txt': No such file or directory: `hdfs://localhost:9000/rgs1/test.txt'
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -mkdir /rgs1
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -copyFromLocal /home/niranjanvs/Desktop/file1.txt /rgs1/test.txt
2020-11-26 01:16:44,134 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /rgs1
Found 1 items
-rw-r--r--   1 niranjanvs supergroup         90 2020-11-26 01:16 /rgs1/test.txt
```

```
                Total vcore-milliseconds taken by all reduce tasks=3671
                Total megabyte-milliseconds taken by all map tasks=6113280
                Total megabyte-milliseconds taken by all reduce tasks=3759104
        Map-Reduce Framework
                Map input records=6
                Map output records=20
                Map output bytes=169
                Map output materialized bytes=115
                Input split bytes=100
                Combine input records=20
                Combine output records=10
                Reduce input groups=10
                Reduce shuffle bytes=115
                Reduce input records=10
                Reduce output records=10
                Spilled Records=20
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=161
                CPU time spent (ms)=1390
                Physical memory (bytes) snapshot=444325888
                Virtual memory (bytes) snapshot=5059944448
                Total committed heap usage (bytes)=355467264
                Peak Map Physical memory (bytes)=267751424
                Peak Map Virtual memory (bytes)=2526457856
                Peak Reduce Physical memory (bytes)=176574464
                Peak Reduce Virtual memory (bytes)=2533486592
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=90
        File Output Format Counters
                Bytes Written=69
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /rgs1
Found 2 items
drwxr-xr-x   - niranjanvs supergroup          0 2020-11-26 01:28 /rgs1/output
-rw-r--r--   1 niranjanvs supergroup         90 2020-11-26 01:16 /rgs1/test.txt
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat /rgs1/output/part-r-00000
2020-11-26 01:30:20,398 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$
```

For the given file, Create a Map Reduce program to Find the average temperature for each year from NCDC data set.

To create jar files using .java files

$ javac AverageReducer.java AverageDriver.java AverageMapper.java -cp $(hadoop classpath)

$ jar -cf Average.jar AverageReducer.class AverageDriver.class AverageMapper.class

To start all the Hadoop deamons

$ ssh localhost

$ cd Hadoop/hadoop-3.2.1

$ sbin/start-dfs.sh

$ sbin/start-yarn.sh

To create a directory in hdfs

$ hadoop fs -mkdir /rgs1

To view all the directories in hdfs

$ hadoop fs -ls /

To copy a file from local system to hdfs directory

$ Hadoop fs -copyFromLocal /home/niranjanvs/Desktop/1901 /rgs1/AverageTest.txt

To view all files in /rgs1 hdfs directory

$ hadoop fs -ls /rgs1

To run a MapReduce program

$ hadoop jar /home/niranjanvs/Desktop/Average.jar AverageDriver /rgs1/AverageTest.txt /rgs1/AverageOutput

To view the output text

$ hadoop fs -cat /rgs1/output/part-r-00000

To stop all the Hadoop deamons

$ sbin/stop-yarn.sh

$ sbin/stop-dfs.sh

AverageDriver.java

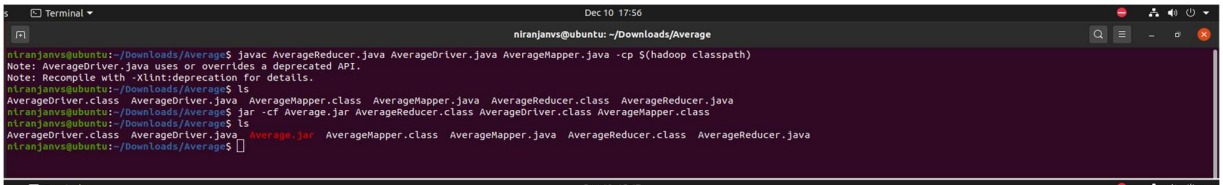
```
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class AverageDriver
{

        public static void main (String[] args) throws Exception
        {
                if (args.length != 2)
                {
                    System.err.println("Please Enter the input and output parameters");
                    System.exit(-1);
                }

               Job job = new Job();
               job.setJarByClass(AverageDriver.class);
               job.setJobName("Max temperature");

               FileInputFormat.addInputPath(job,new Path(args[0]));
               FileOutputFormat.setOutputPath(job,new Path (args[1]));

               job.setMapperClass(AverageMapper.class);
               job.setReducerClass(AverageReducer.class);

               job.setOutputKeyClass(Text.class);
               job.setOutputValueClass(IntWritable.class);

               System.exit(job.waitForCompletion(true)?0:1);
        }
}
```

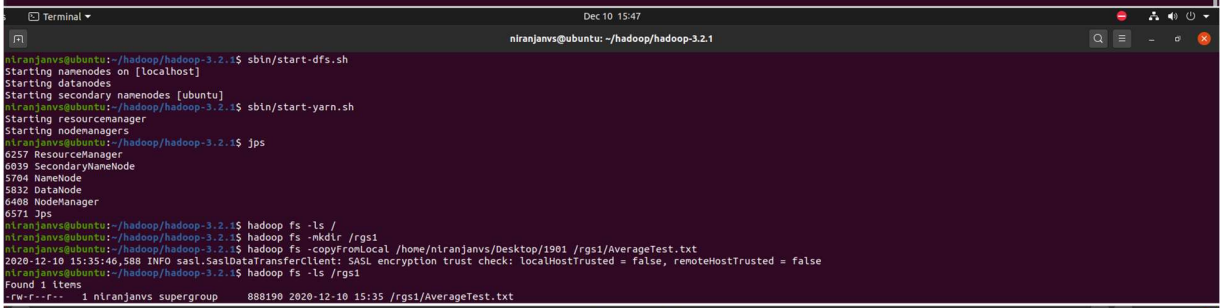
AverageMapper.java

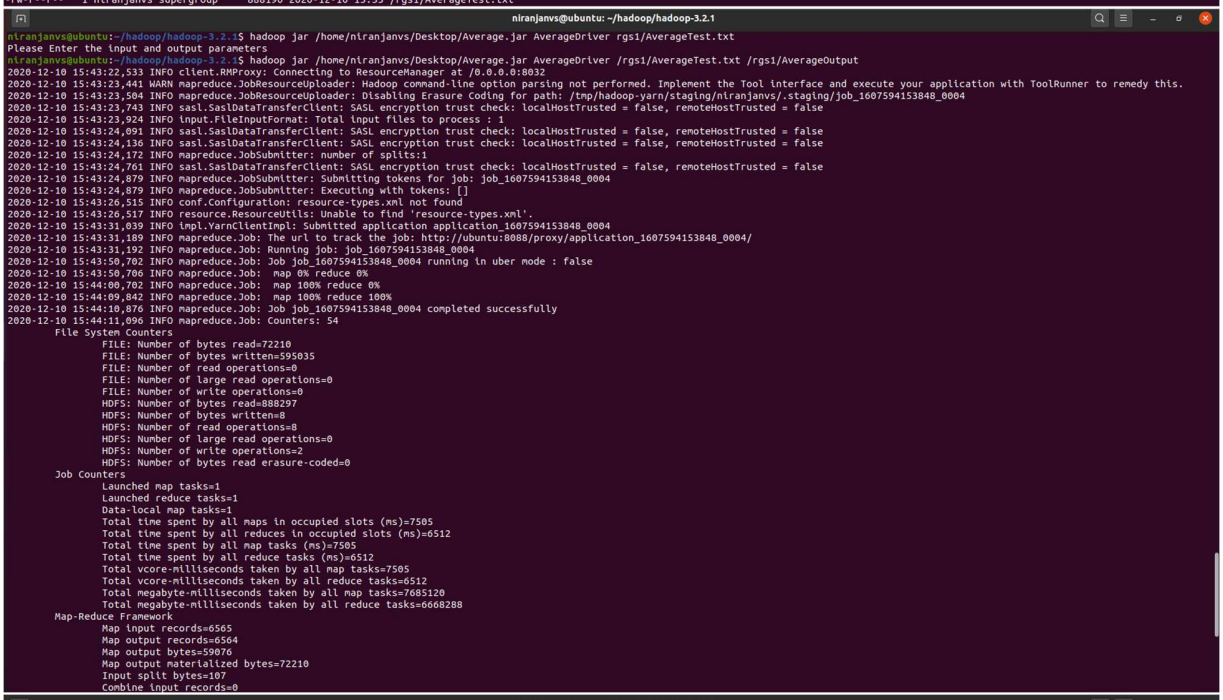
```
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{

public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws
IOException,  InterruptedException
        {
                String line = value.toString();
                String year = line.substring(15,19);
                int temperature;
                if (line.charAt(87)=='+')
                        temperature = Integer.parseInt(line.substring(88, 92));
                else
                        temperature = Integer.parseInt(line.substring(87, 92));

                String quality = line.substring(92, 93);
                if(temperature != MISSING && quality.matches("[01459]"))
                context.write(new Text(year),new IntWritable(temperature));
        }
}
```

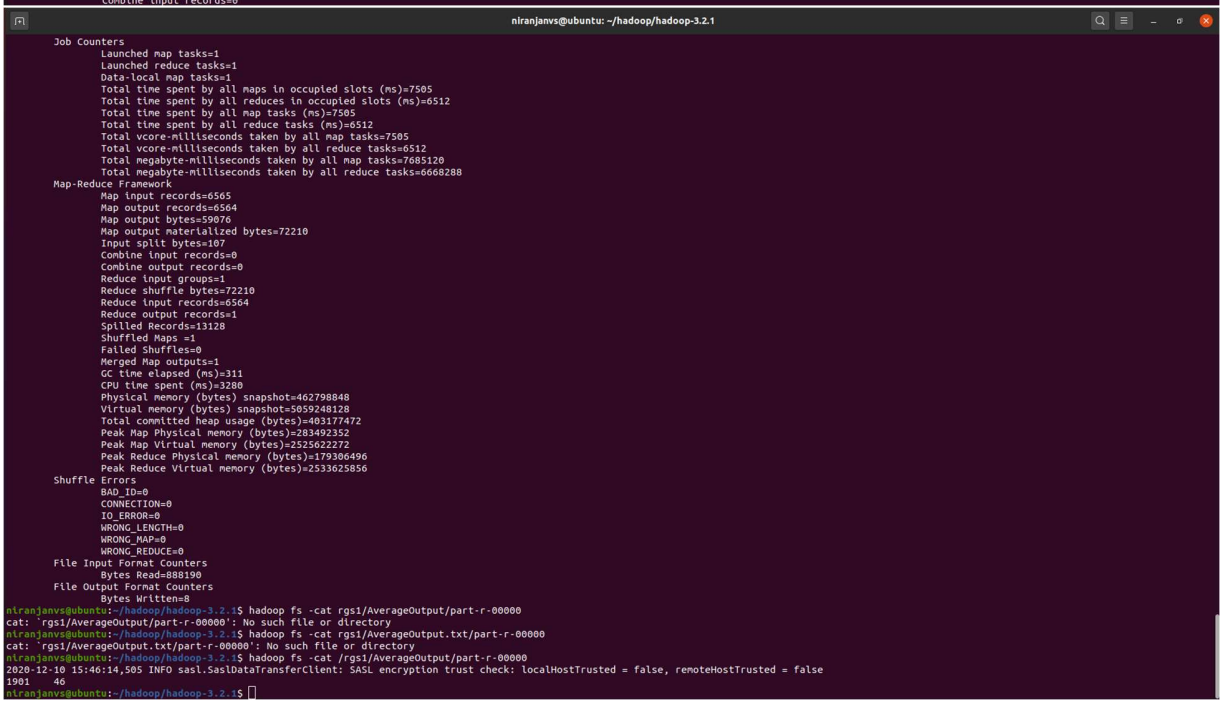
AverageReducer.java

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable >
 {
public void reduce(Text key,  Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException
        {
        int max_temp = 0;
        int count = 0;
        for (IntWritable value : values)
                {
                        max_temp += value.get();
                        count+=1;
                }
        context.write(key, new IntWritable(max_temp/count));
        }
}
```


Terminal
Dec 10 17:56
niranjanvs@ubuntu: ~/Downloads/Average
niranjanvs@ubuntu:~/Downloads/Average$ javac AverageReducer.java AverageDriver.java AverageMapper.java -cp $(hadoop classpath)
Note: AverageDriver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
niranjanvs@ubuntu:~/Downloads/Average$ ls
AverageDriver.class AverageDriver.java AverageMapper.class AverageMapper.java AverageReducer.class AverageReducer.java
niranjanvs@ubuntu:~/Downloads/Average$ jar -cf Average.jar AverageReducer.class AverageDriver.class AverageMapper.class
niranjanvs@ubuntu:~/Downloads/Average$ ls
AverageDriver.class AverageDriver.java Average.jar AverageMapper.class AverageMapper.java AverageReducer.class AverageReducer.java
niranjanvs@ubuntu:~/Downloads/Average$


Terminal
Dec 10 15:47
niranjanvs@ubuntu: ~/hadoop/hadoop-3.2.1
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ jps
6257 ResourceManager
6039 SecondaryNameNode
5704 NameNode
5832 DataNode
6408 NodeManager
6571 Jps
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -mkdir /rgs1
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -copyFromLocal /home/niranjanvs/Desktop/1901 /rgs1/AveragеTest.txt
2020-12-10 15:35:46,588 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /rgs1
Found 1 items
-rw-r--r-- 1 niranjanvs supergroup 888190 2020-12-10 15:35 /rgs1/AveragеTest.txt


niranjanvs@ubuntu: ~/hadoop/hadoop-3.2.1
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop jar /home/niranjanvs/Desktop/Average.jar AverageDriver rgs1/AveragеTest.txt
Please Enter the input and output parameters
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop jar /home/niranjanvs/Desktop/Average.jar AverageDriver /rgs1/AveragеTest.txt /rgs1/AverageOutput
2020-12-10 15:43:22,533 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-10 15:43:23,441 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2020-12-10 15:43:23,504 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/niranjanvs/.staging/job_1607594153848_0004
2020-12-10 15:43:23,743 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:23,924 INFO input.FileInputFormat: Total input files to process : 1
2020-12-10 15:43:24,091 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,136 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,172 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-10 15:43:24,761 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,879 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607594153848_0004
2020-12-10 15:43:24,879 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-10 15:43:26,515 INFO conf.Configuration: resource-types.xml not found
2020-12-10 15:43:26,517 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-10 15:43:31,039 INFO impl.YarnClientImpl: Submitted application application_1607594153848_0004
2020-12-10 15:43:31,189 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1607594153848_0004/
2020-12-10 15:43:31,192 INFO mapreduce.Job: Running job: job_1607594153848_0004
2020-12-10 15:43:50,702 INFO mapreduce.Job: Job job_1607594153848_0004 running in uber mode : false
2020-12-10 15:43:50,706 INFO mapreduce.Job: map 0% reduce 0%
2020-12-10 15:44:00,702 INFO mapreduce.Job: map 100% reduce 0%
2020-12-10 15:44:09,842 INFO mapreduce.Job: map 100% reduce 100%
2020-12-10 15:44:10,876 INFO mapreduce.Job: Job job_1607594153848_0004 completed successfully
2020-12-10 15:44:11,096 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=72210
FILE: Number of bytes written=595035
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=888297
HDFS: Number of bytes written=8
HDFS: Number of read operations=8
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=7505
Total time spent by all reduces in occupied slots (ms)=6512
Total time spent by all map tasks (ms)=7505
Total time spent by all reduce tasks (ms)=6512
Total vcore-milliseconds taken by all map tasks=7505
Total vcore-milliseconds taken by all reduce tasks=6512
Total megabyte-milliseconds taken by all map tasks=7685120
Total megabyte-milliseconds taken by all reduce tasks=6668288
Map-Reduce Framework
Map input records=6565
Map output records=6564
Map output bytes=59076
Map output materialized bytes=72210
Input split bytes=107
Combine input records=0


niranjanvs@ubuntu: ~/hadoop/hadoop-3.2.1
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=7505
Total time spent by all reduces in occupied slots (ms)=6512
Total time spent by all map tasks (ms)=7505
Total time spent by all reduce tasks (ms)=6512
Total vcore-milliseconds taken by all map tasks=7505
Total vcore-milliseconds taken by all reduce tasks=6512
Total megabyte-milliseconds taken by all map tasks=7685120
Total megabyte-milliseconds taken by all reduce tasks=6668288
Map-Reduce Framework
Map input records=6565
Map output records=6564
Map output bytes=59076
Map output materialized bytes=72210
Input split bytes=107
Combine input records=0
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=72210
Reduce input records=6564
Reduce output records=1
Spilled Records=13128
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=311
CPU time spent (ms)=3280
Physical memory (bytes) snapshot=462798848
Virtual memory (bytes) snapshot=5059248128
Total committed heap usage (bytes)=403177472
Peak Map Physical memory (bytes)=283492352
Peak Map Virtual memory (bytes)=2525622272
Peak Reduce Physical memory (bytes)=179306496
Peak Reduce Virtual memory (bytes)=2533625856
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=888190
File Output Format Counters
Bytes Written=8
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat rgs1/AverageOutput/part-r-00000
cat: `rgs1/AverageOutput/part-r-00000': No such file or directory
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat rgs1/AverageOutput.txt/part-r-00000
cat: `rgs1/AverageOutput.txt/part-r-00000': No such file or directory
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat /rgs1/AverageOutput/part-r-00000
2020-12-10 15:46:14,505 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
1901 46
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$

Write Queries in Hive to do the following

1. Create an external table named with the following attributes -> Empl_ID ->Emp_Name -> Designation -> Salary

create database if not exists Employee comment 'BDA LAB WEEK 09';

use Employee;

create external table if not exists Employee (Empl_ID int, Emp_Name String, Designation String, Salary int) row format delimited fields terminated by ',' lines terminated by '\n';

2. Load data into table from a given file

load data local inpath '/home/niranjanvs/Desktop/employee' overwrite into table Employee;
select * from Employee;

3. Create a view to Generate a query to retrieve the employee details who earn a salary of more than Rs 30000.

create view Employee_view as select * from Employee where Salary>30000;

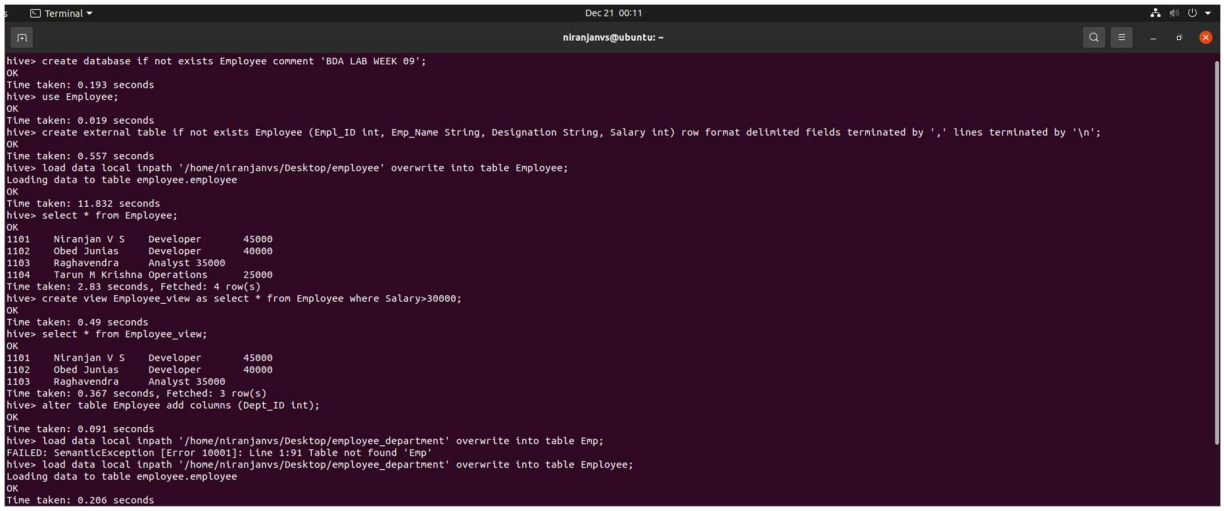
select * from Employee_view;

4. Alter the table to add a column Dept_Id and Generate a query to retrieve the employee details in order by using Dept_Id

alter table Employee add columns (Dept_ID int);

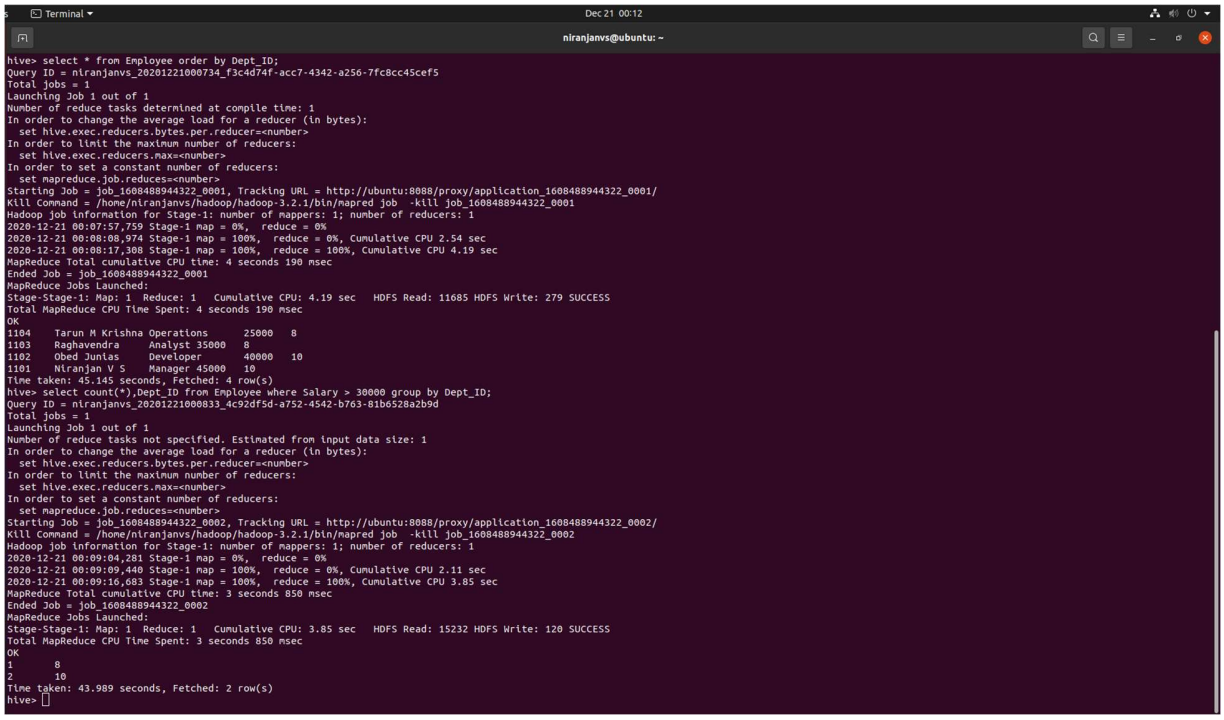
load data local inpath '/home/niranjanvs/Desktop/employee_department' overwrite into table Employee;

select * from Employee order by Dept_ID;



5. Generate a query to retrieve the number of employees in each department whose salary is greater than 30000

select count(*),Dept_ID from Employee where Salary > 30000 group by Dept_ID;

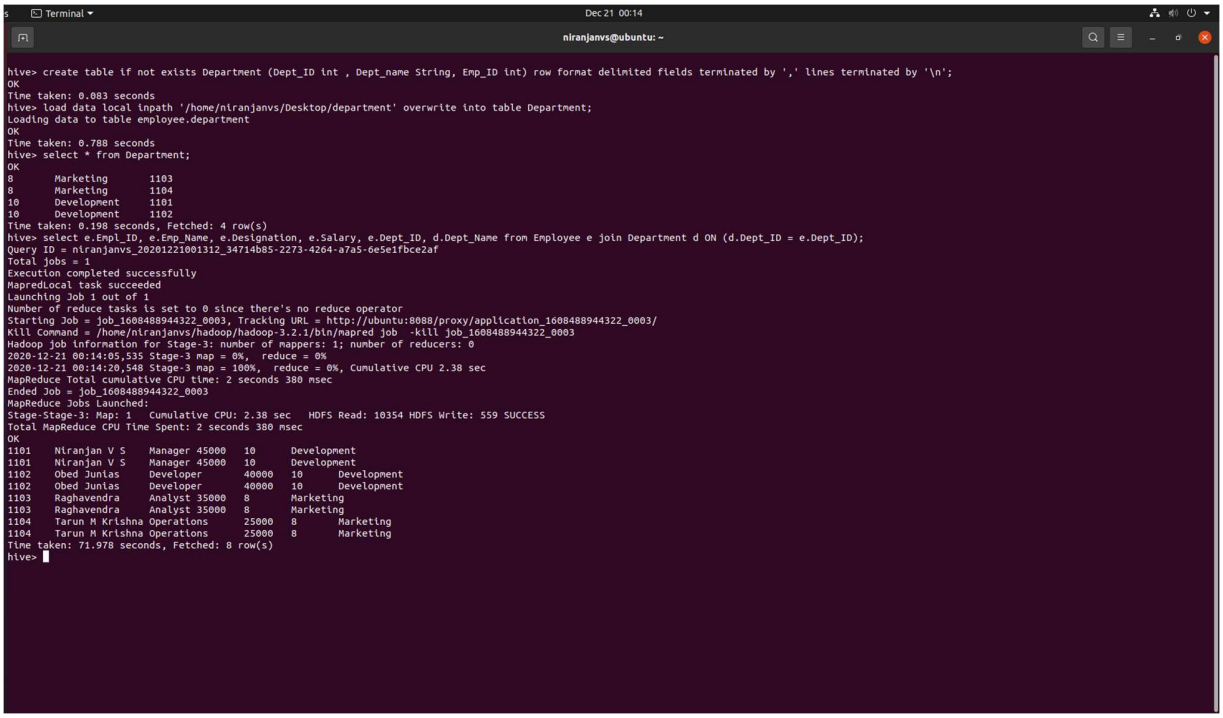


6. Create another table Department with attributes -> Dept_Id ->Dept_name ->Emp_Id

create table if not exists Department (Dept_ID int , Dept_name String, Emp_ID int) row format delimited fields terminated by ',' lines terminated by '\n';
load data local inpath '/home/niranjanvs/Desktop/department' overwrite into table Department;
select * from Department;

7.Display the cumulative details of each employee along with department details

select e.Empl_ID, e.Emp_Name, e.Designation, e.Salary, e.Dept_ID, d.Dept_Name from Employee e join Department d ON (d.Dept_ID = e.Dept_ID);


Terminal
Dec 21 00:14
niranjanvs@ubuntu: ~
hive> create table if not exists Department (Dept_ID int , Dept_name String, Emp_ID int) row format delimited fields terminated by ',' lines terminated by '\n';
OK
Time taken: 0.083 seconds
hive> load data local inpath '/home/niranjanvs/Desktop/department' overwrite into table Department;
Loading data to table employee.department
OK
Time taken: 0.788 seconds
hive> select * from Department;
OK
8 Marketing 1103
8 Marketing 1104
10 Development 1101
10 Development 1102
Time taken: 0.198 seconds, Fetched: 4 row(s)
hive> select e.Empl_ID, e.Emp_Name, e.Designation, e.Salary, e.Dept_ID, d.Dept_Name from Employee e join Department d ON (d.Dept_ID = e.Dept_ID);
Query ID = niranjanvs_20201221001312_34714b85-2273-4264-a7a5-6e5e1fbce2af
Total jobs = 1
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1608488944322_0003, Tracking URL = http://ubuntu:8088/proxy/application_1608488944322_0003/
Kill Command = /home/niranjanvs/hadoop/hadoop-3.2.1/bin/mapred job -kill job_1608488944322_0003
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2020-12-21 00:14:05,535 Stage-3 map = 0%, reduce = 0%
2020-12-21 00:14:20,548 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.38 sec
MapReduce Total cumulative CPU time: 2 seconds 380 msec
Ended Job = job_1608488944322_0003
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1 Cumulative CPU: 2.38 sec HDFS Read: 10354 HDFS Write: 559 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 380 msec
OK
1101 Niranjan V S Manager 45000 10 Development
1101 Niranjan V S Manager 45000 10 Development
1102 Obed Junias Developer 40000 10 Development
1102 Obed Junias Developer 40000 10 Development
1103 Raghavendra Analyst 35000 8 Marketing
1103 Raghavendra Analyst 35000 8 Marketing
1104 Tarun M Krishna Operations 25000 8 Marketing
1104 Tarun M Krishna Operations 25000 8 Marketing
Time taken: 71.978 seconds, Fetched: 8 row(s)
hive>