UMKC

UNIVERSITY OF MISSOURI-KANSAS CITY

Bigdata Hadoop Programming

# Lab 1 Assignment

Team Members:

Pragathi Thammaneni

Sridevi Mallipudi

LAB -1 Assignment

Introduction:

The main core concept for executing the Lab 1 Assignment is to implement the Map Reduce Algorithm for finding the solution for the Facebook common friends problem and to run the MapReduce job on Apache Hadoop. And implementing the No Sql comparison over HBase and Cassandra.

Objectives:

To code for the 2 questions the below concepts are implemented.

Map Reduce Algorithm for Facebook common friend's problem

Cassandra

HBase

No SQL Comparison over HBase and Cassandra

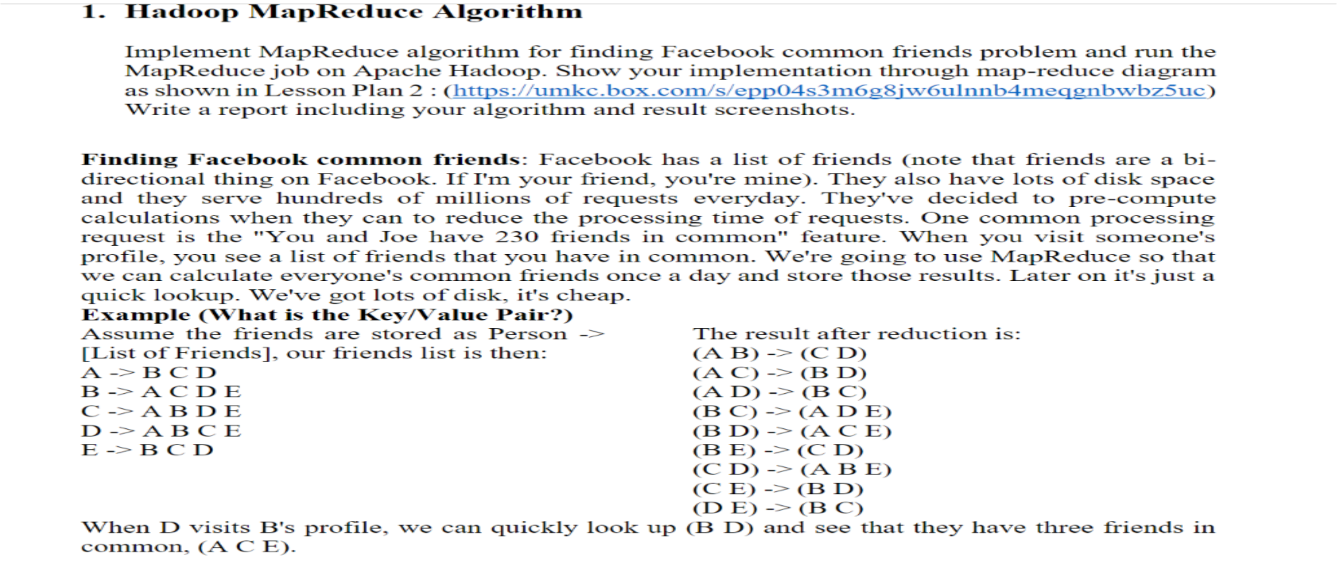
Approaches /Methods:

Using Hadoop 1.2.1, HBase 0.94.8, Intellij (Community edition)

Workflow &Datasets/Parameters and Evaluation:

The below each question will follow different approaches to solve. Coding is done to perform the evaluation of each individual snippet to execute the datasets which are provided as the input parameters.

Question 1:



1. Hadoop MapReduce Algorithm

Implement MapReduce algorithm for finding Facebook common friends problem and run the MapReduce job on Apache Hadoop. Show your implementation through map-reduce diagram as shown in Lesson Plan 2 : (https://umkc.box.com/s/epp04s3m6g8jw6ulnnb4meqgnbwbz5uc) Write a report including your algorithm and result screenshots.

Finding Facebook common friends: Facebook has a list of friends (note that friends are a bi-directional thing on Facebook. If I'm your friend, you're mine). They also have lots of disk space and they serve hundreds of millions of requests everyday. They've decided to pre-compute calculations when they can to reduce the processing time of requests. One common processing request is the "You and Joe have 230 friends in common" feature. When you visit someone's profile, you see a list of friends that you have in common. We're going to use MapReduce so that we can calculate everyone's common friends once a day and store those results. Later on it's just a quick lookup. We've got lots of disk, it's cheap.

Example (What is the Key/Value Pair?)

Assume the friends are stored as Person -> [List of Friends], our friends list is then:

A -> B C D
B -> A C D E
C -> A B D E
D -> A B C E
E -> B C D

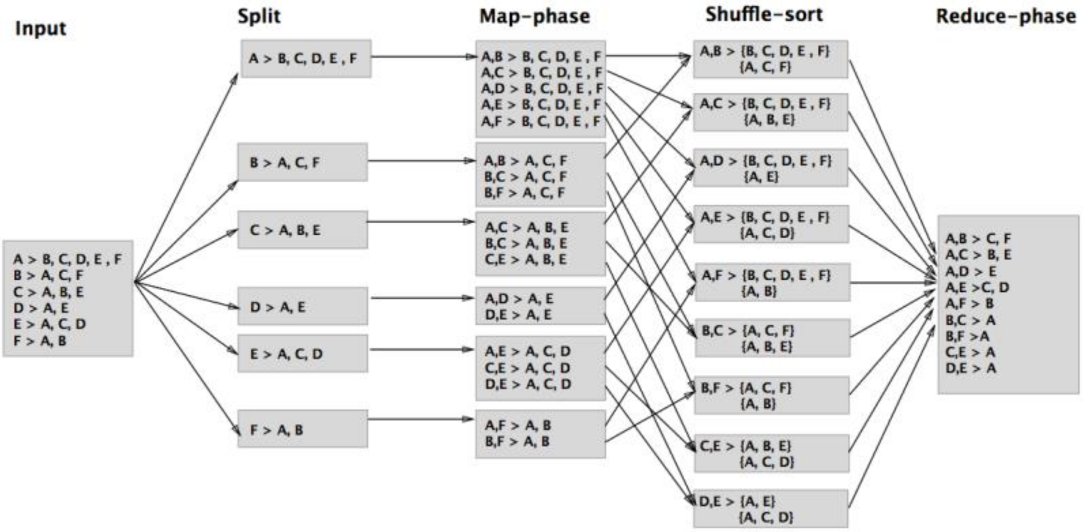
The result after reduction is:

(A B) -> (C D)
(A C) -> (B D)
(A D) -> (B C)
(B C) -> (A D E)
(B D) -> (A C E)
(B E) -> (C D)
(C D) -> (A B E)
(C E) -> (B D)
(D E) -> (B C)

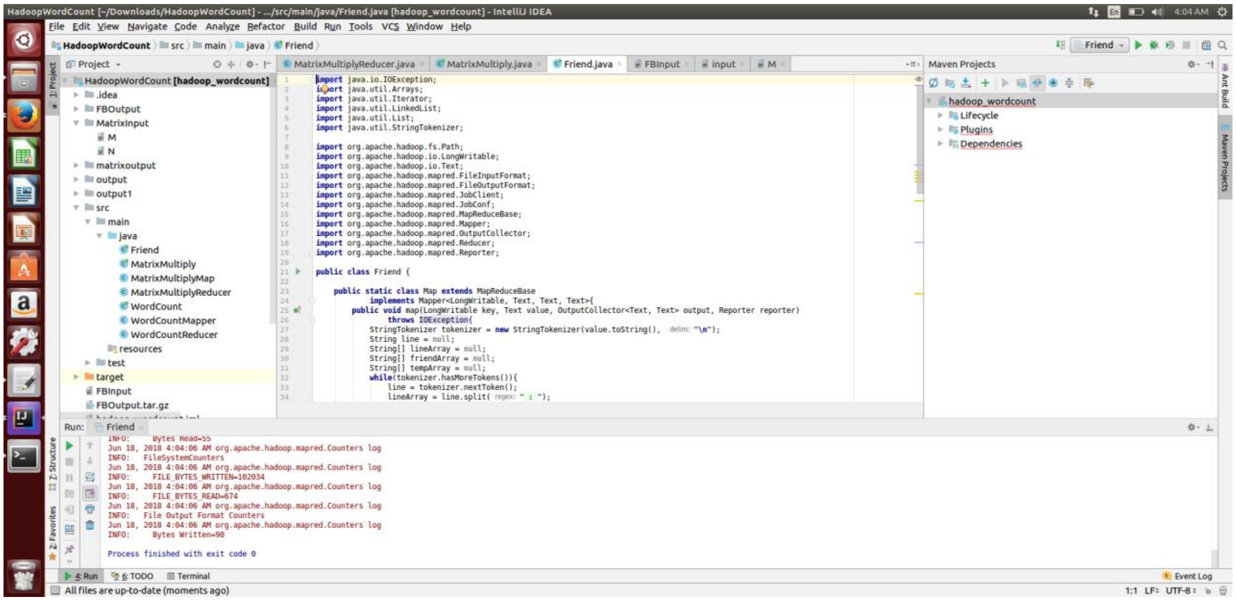
When D visits B's profile, we can quickly look up (B D) and see that they have three friends in common, (A C E).

For solving the below question a map reduce algorithm is used to find out the face book common friend problem. Below is the sample use case for implementing the Map reduce .

Use Case Diagram:



Below is the screen shot for the Map Reduce Algorithm

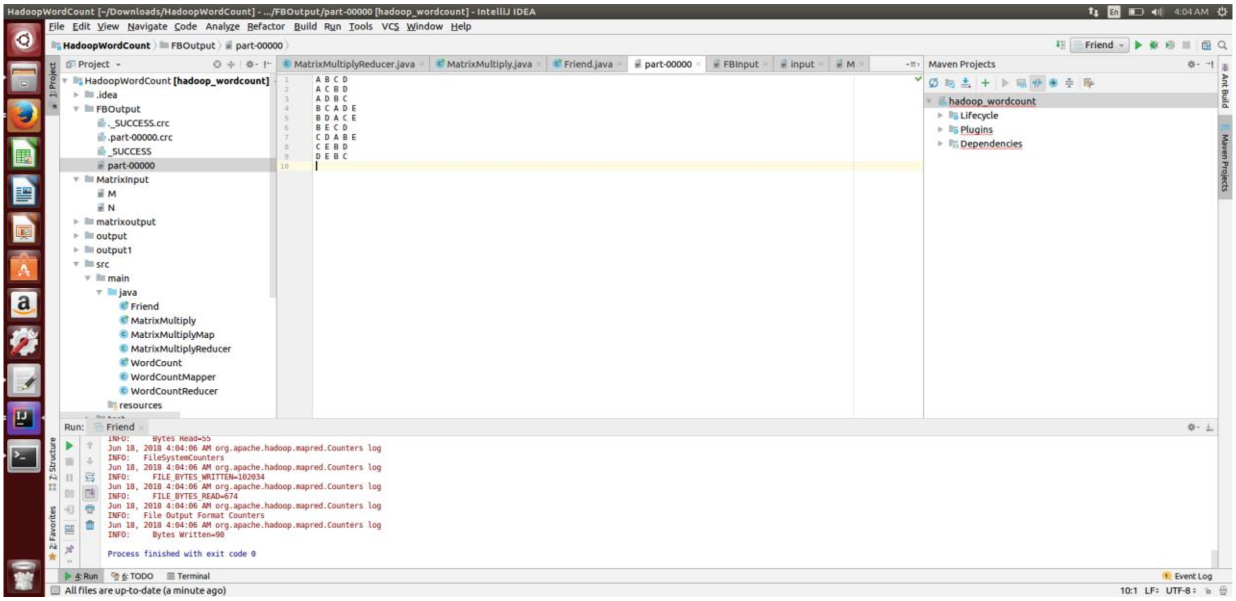


```
import java.io.IOException;
import java.util.Arrays;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.StringTokenizer;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class Friend {

    public static class Map extends MapReduceBase
            implements Mapper<LongWritable, Text, Text, Text>{
        public void map(LongWritable key, Text value, OutputCollector<Text, Text> output, Reporter reporter)
                throws IOException{
            StringTokenizer tokenizer = new StringTokenizer(value.toString(),  delim: "\n");
            String line = null;
            String[] lineArray = null;
            String[] friendArray = null;
            String[] tempArray = null;
            while(tokenizer.hasMoreTokens()){
                line = tokenizer.nextToken();
                lineArray = line.split( regex: " : ");
```

```
INFO:     Bytes Read=55
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:   FileSystemCounters
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     FILE_BYTES_WRITTEN=102034
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     FILE_BYTES_READ=674
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:   File Output Format Counters
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     Bytes Written=98

Process finished with exit code 0
```

All files are up-to-date (moments ago)

Output Screenshot:

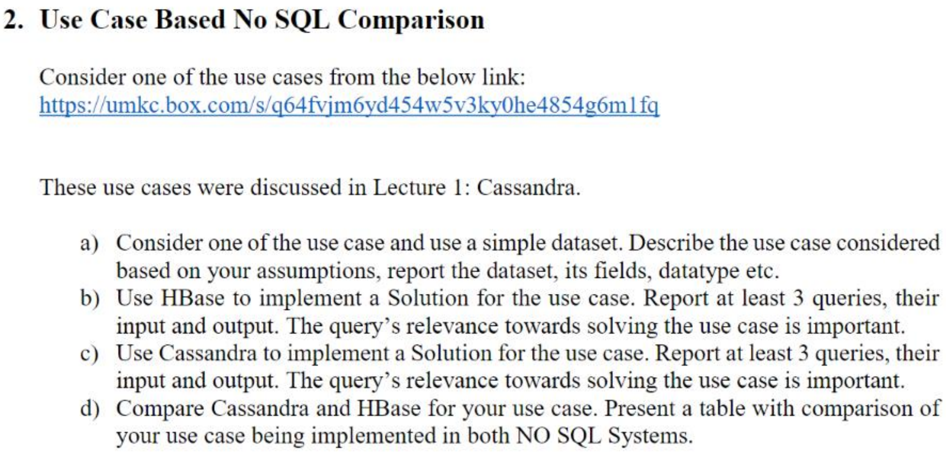


```
A B C D
A C B D
A D B C
B C A D E
B D A C E
B E C D
C D A B E
C E B D
D E B C
```

```
INFO:     Bytes Read=55
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:   FileSystemCounters
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     FILE_BYTES_WRITTEN=102034
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     FILE_BYTES_READ=674
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:   File Output Format Counters
Jun 18, 2018 4:04:06 AM org.apache.hadoop.mapred.Counters log
INFO:     Bytes Written=90

Process finished with exit code 0
```

All files are up-to-date (a minute ago)

Question 2:



2. Use Case Based No SQL Comparison

Consider one of the use cases from the below link:
https://umkc.box.com/s/q64fvjm6yd454w5v3ky0he4854g6m1fq

These use cases were discussed in Lecture 1: Cassandra.

a) Consider one of the use case and use a simple dataset. Describe the use case considered based on your assumptions, report the dataset, its fields, datatype etc.
b) Use HBase to implement a Solution for the use case. Report at least 3 queries, their input and output. The query's relevance towards solving the use case is important.
c) Use Cassandra to implement a Solution for the use case. Report at least 3 queries, their input and output. The query's relevance towards solving the use case is important.
d) Compare Cassandra and HBase for your use case. Present a table with comparison of your use case being implemented in both NO SQL Systems.

For solving the above 4 cases below work flow is been followed, created tables in Hbase and Cassandra and then performed three queries on each. Also Performed the comparison over Cassandra and Hbase for the usecase being implemented in both No SQL Systems**.**

No Sql Comparison over Hbase and Cassandra

CASSANDRA implementation:

For solving this use case we opt to choose Couresera data set, the following are the fields and data types used for implementing the Cassandra.

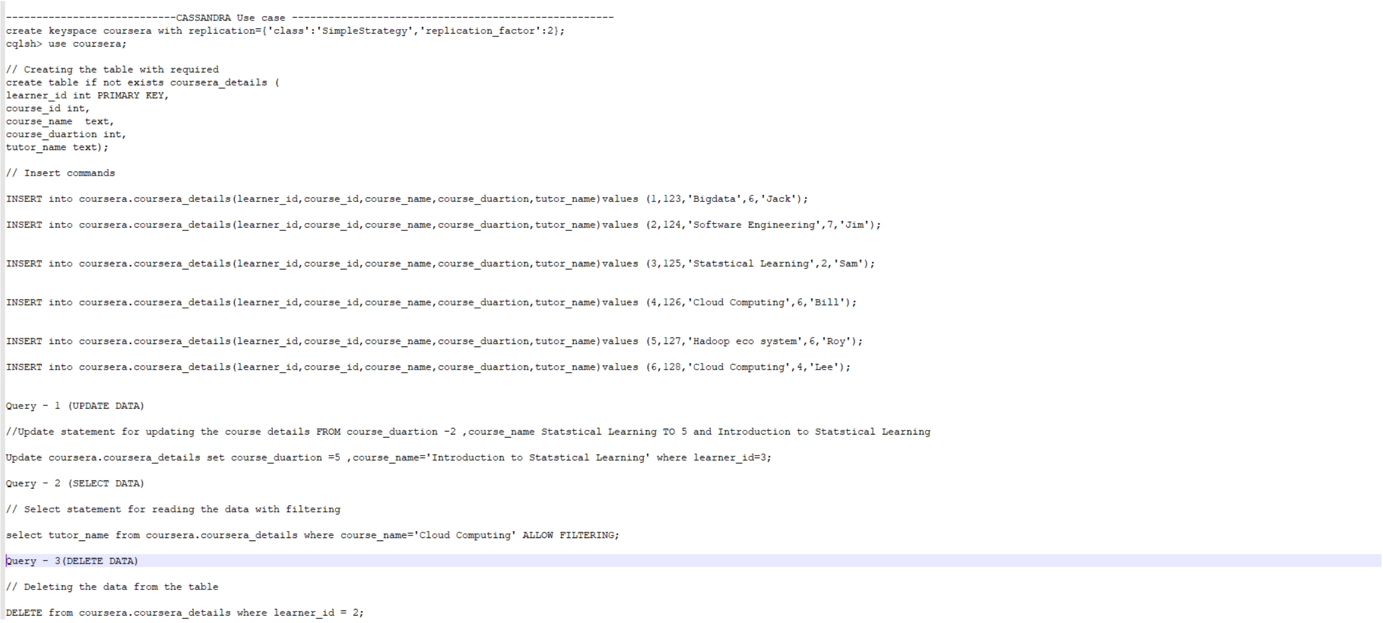
The data set will include below fields and data types

```
create keyspace coursera with replication={'class':'SimpleStrategy','replication_factor':2};
cqlsh> use coursera;

// Creating the table with required
create table if not exists coursera_details (
learner_id int PRIMARY KEY,
course_id int,
course_name  text,
course_duartion int,
tutor_name text);
```

Created a table with the above details so that the below CRUD operations can be implemented to retrieve the data

Update ,Select and delete operations are performed on the data set .Below are the commands and respective screen shorts for the execution

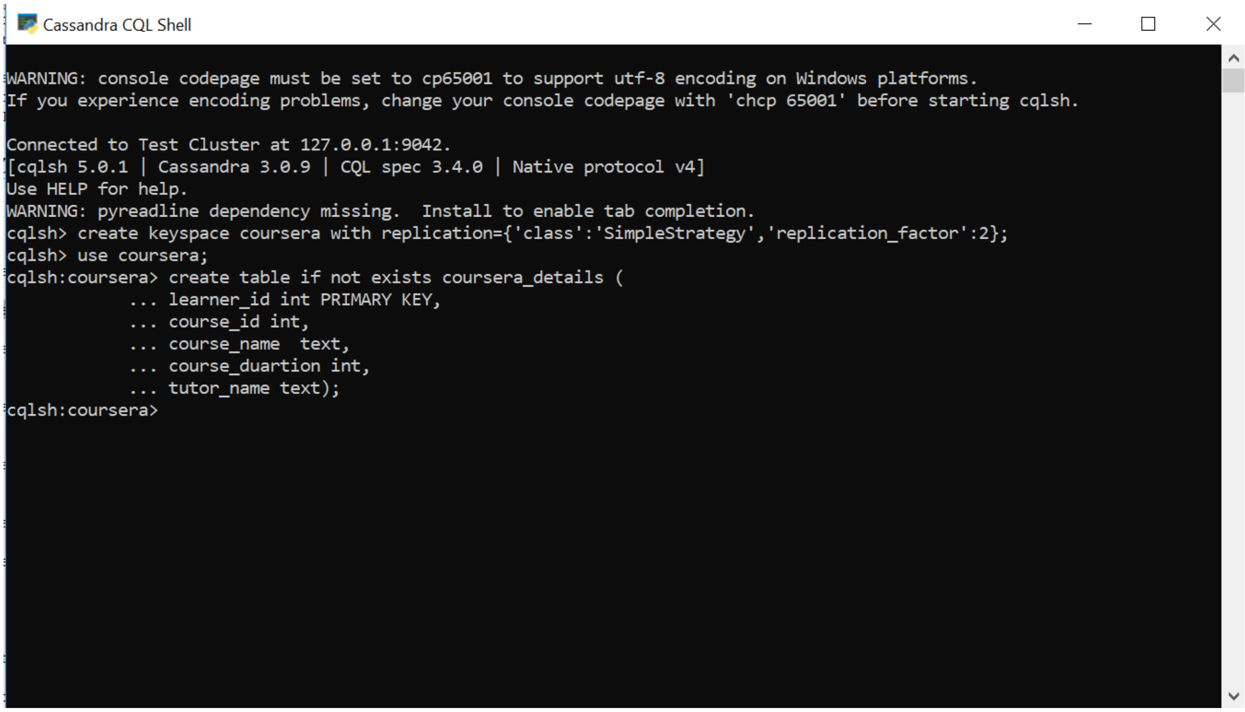
Queries for implementing the Cassandra:



```
----------------------------CASSANDRA Use case -----------------------------------------------------
create keyspace coursera with replication={'class':'SimpleStrategy','replication_factor':2};
cqlsh> use coursera;

// Creating the table with required
create table if not exists coursera_details (
learner_id int PRIMARY KEY,
course_id int,
course_name  text,
course_duartion int,
tutor_name text);

// Insert commands

INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (1,123,'Bigdata',6,'Jack');

INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (2,124,'Software Engineering',7,'Jim');


INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (3,125,'Statstical Learning',2,'Sam');


INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (4,126,'Cloud Computing',6,'Bill');


INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (5,127,'Hadoop eco system',6,'Roy');

INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values (6,128,'Cloud Computing',4,'Lee');


Query - 1 (UPDATE DATA)

//Update statement for updating the course details FROM course_duartion -2 ,course_name Statstical Learning TO 5 and Introduction to Statstical Learning

Update coursera.coursera_details set course_duartion =5 ,course_name='Introduction to Statstical Learning' where learner_id=3;

Query - 2 (SELECT DATA)

// Select statement for reading the data with filtering

select tutor_name from coursera.coursera_details where course_name='Cloud Computing' ALLOW FILTERING;

Query - 3(DELETE DATA)

// Deleting the data from the table

DELETE from coursera.coursera_details where learner_id = 2;
```

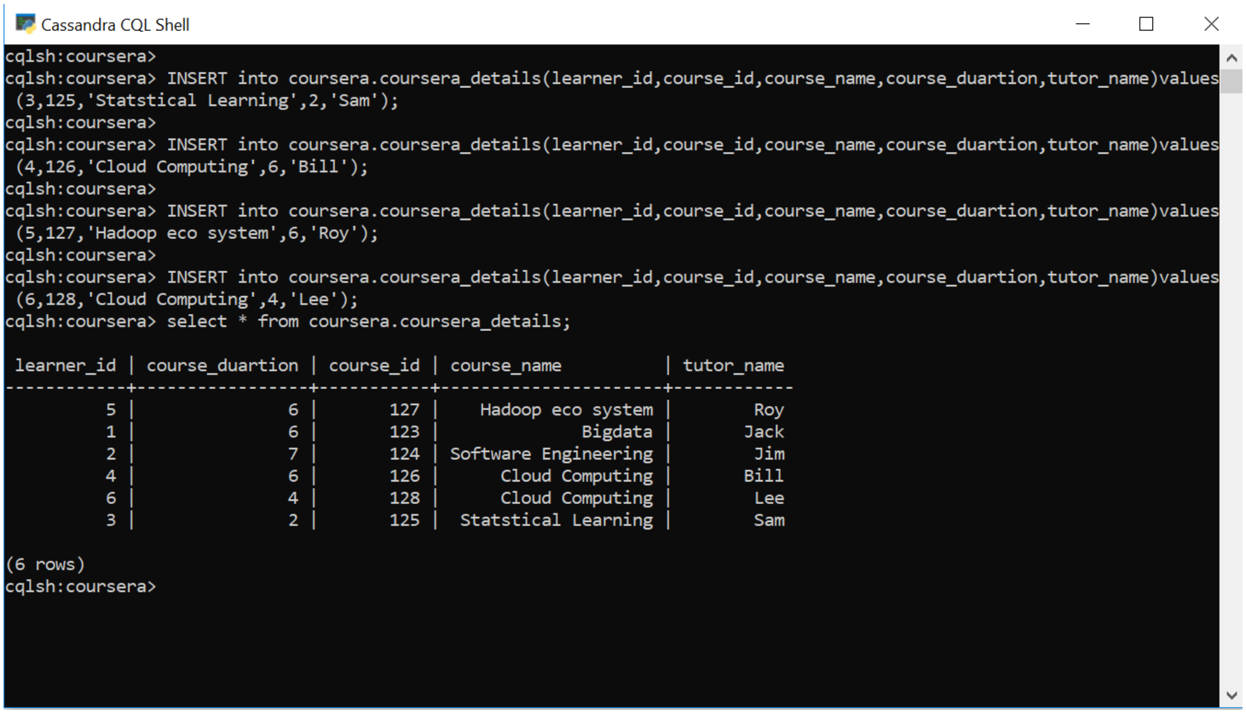
Execution for Cassandra use case:

Table Creation



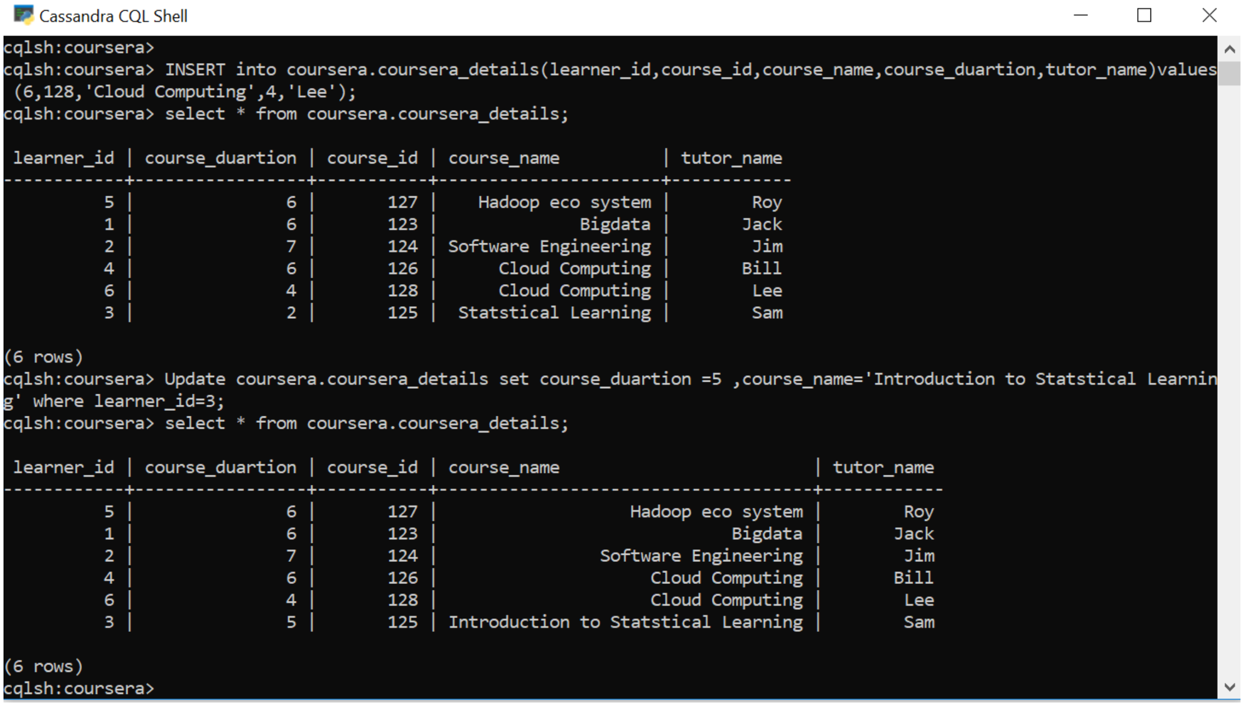
```
WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> create keyspace coursera with replication={'class':'SimpleStrategy','replication_factor':2};
cqlsh> use coursera;
cqlsh:coursera> create table if not exists coursera_details (
            ... learner_id int PRIMARY KEY,
            ... course_id int,
            ... course_name  text,
            ... course_duartion int,
            ... tutor_name text);
cqlsh:coursera>
```

Insert Statements



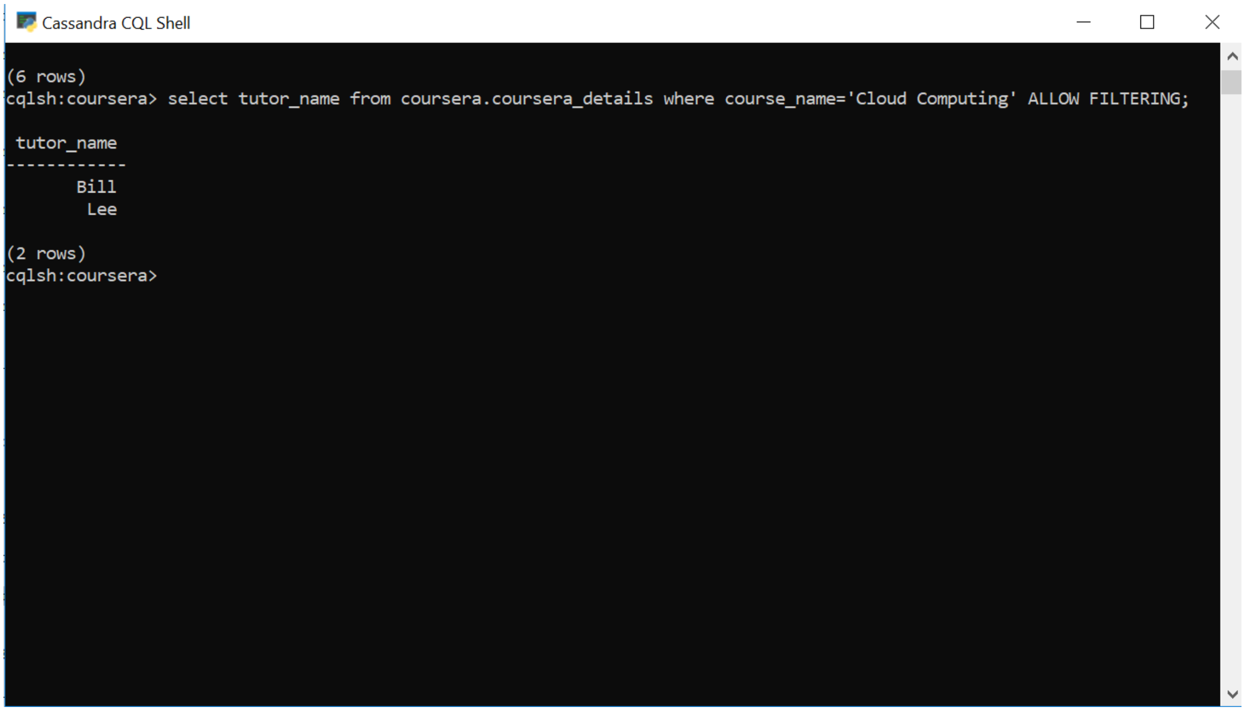
```
cqlsh:coursera>
cqlsh:coursera> INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values
 (3,125,'Statstical Learning',2,'Sam');
cqlsh:coursera>
cqlsh:coursera> INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values
 (4,126,'Cloud Computing',6,'Bill');
cqlsh:coursera>
cqlsh:coursera> INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values
 (5,127,'Hadoop eco system',6,'Roy');
cqlsh:coursera>
cqlsh:coursera> INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values
 (6,128,'Cloud Computing',4,'Lee');
cqlsh:coursera> select * from coursera.coursera_details;

 learner_id | course_duartion | course_id | course_name          | tutor_name
------------+-----------------+-----------+----------------------+------------
          5 |               6 |       127 |    Hadoop eco system |        Roy
          1 |               6 |       123 |              Bigdata |       Jack
          2 |               7 |       124 | Software Engineering |        Jim
          4 |               6 |       126 |      Cloud Computing |       Bill
          6 |               4 |       128 |      Cloud Computing |        Lee
          3 |               2 |       125 |  Statstical Learning |        Sam

(6 rows)
cqlsh:coursera>
```

Query 1- UPDATE DATA



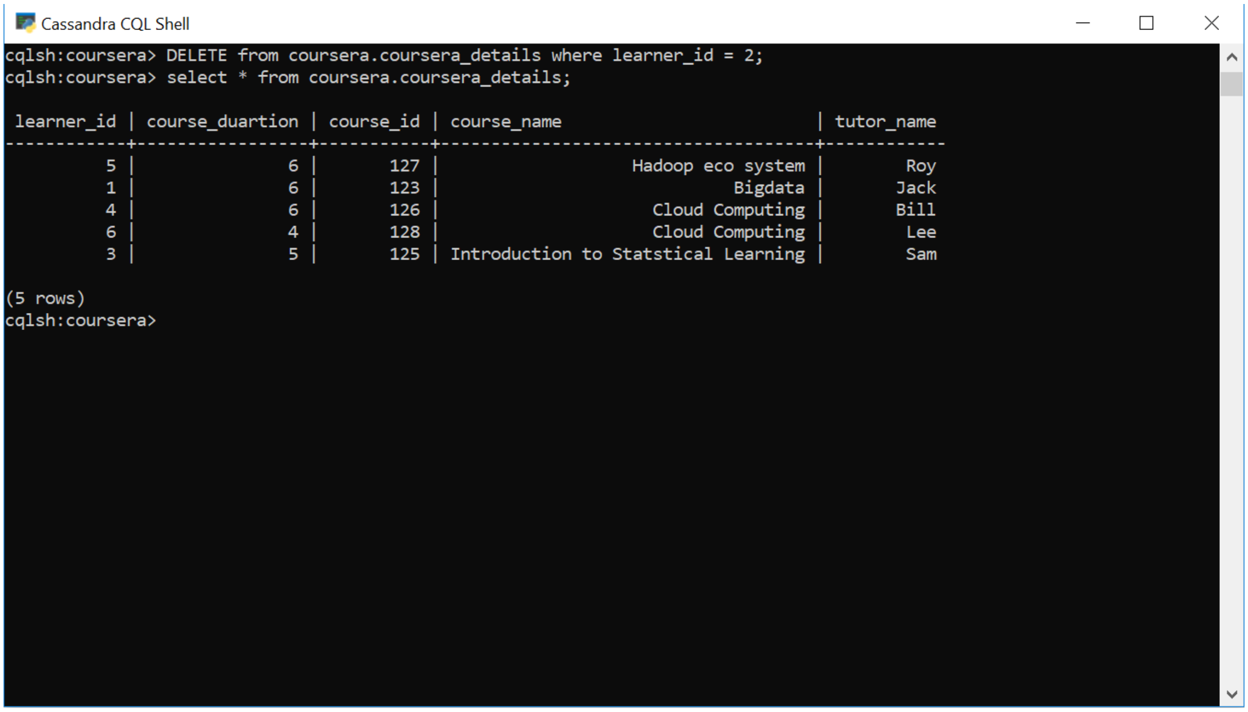
```
cqlsh:coursera>
cqlsh:coursera> INSERT into coursera.coursera_details(learner_id,course_id,course_name,course_duartion,tutor_name)values
 (6,128,'Cloud Computing',4,'Lee');
cqlsh:coursera> select * from coursera.coursera_details;

 learner_id | course_duartion | course_id | course_name          | tutor_name
------------+-----------------+-----------+----------------------+------------
          5 |               6 |       127 |    Hadoop eco system |        Roy
          1 |               6 |       123 |              Bigdata |       Jack
          2 |               7 |       124 | Software Engineering |        Jim
          4 |               6 |       126 |      Cloud Computing |       Bill
          6 |               4 |       128 |      Cloud Computing |        Lee
          3 |               2 |       125 |  Statstical Learning |        Sam

(6 rows)
cqlsh:coursera> Update coursera.coursera_details set course_duartion =5 ,course_name='Introduction to Statstical Learnin
g' where learner_id=3;
cqlsh:coursera> select * from coursera.coursera_details;

 learner_id | course_duartion | course_id | course_name                         | tutor_name
------------+-----------------+-----------+-------------------------------------+------------
          5 |               6 |       127 |                   Hadoop eco system |        Roy
          1 |               6 |       123 |                             Bigdata |       Jack
          2 |               7 |       124 |                Software Engineering |        Jim
          4 |               6 |       126 |                     Cloud Computing |       Bill
          6 |               4 |       128 |                     Cloud Computing |        Lee
          3 |               5 |       125 | Introduction to Statstical Learning |        Sam

(6 rows)
cqlsh:coursera>
```

Query 2- SELECT DATA



```
(6 rows)
cqlsh:coursera> select tutor_name from coursera.coursera_details where course_name='Cloud Computing' ALLOW FILTERING;

 tutor_name
------------
       Bill
        Lee

(2 rows)
cqlsh:coursera>
```

Query 3- DELETE DATA



```
cqlsh:coursera> DELETE from coursera.coursera_details where learner_id = 2;
cqlsh:coursera> select * from coursera.coursera_details;

 learner_id | course_duartion | course_id | course_name                        | tutor_name
------------+-----------------+-----------+------------------------------------+------------
          5 |               6 |       127 |                  Hadoop eco system |        Roy
          1 |               6 |       123 |                            Bigdata |       Jack
          4 |               6 |       126 |                    Cloud Computing |       Bill
          6 |               4 |       128 |                    Cloud Computing |        Lee
          3 |               5 |       125 | Introduction to Statstical Learning |        Sam

(5 rows)
cqlsh:coursera>
```

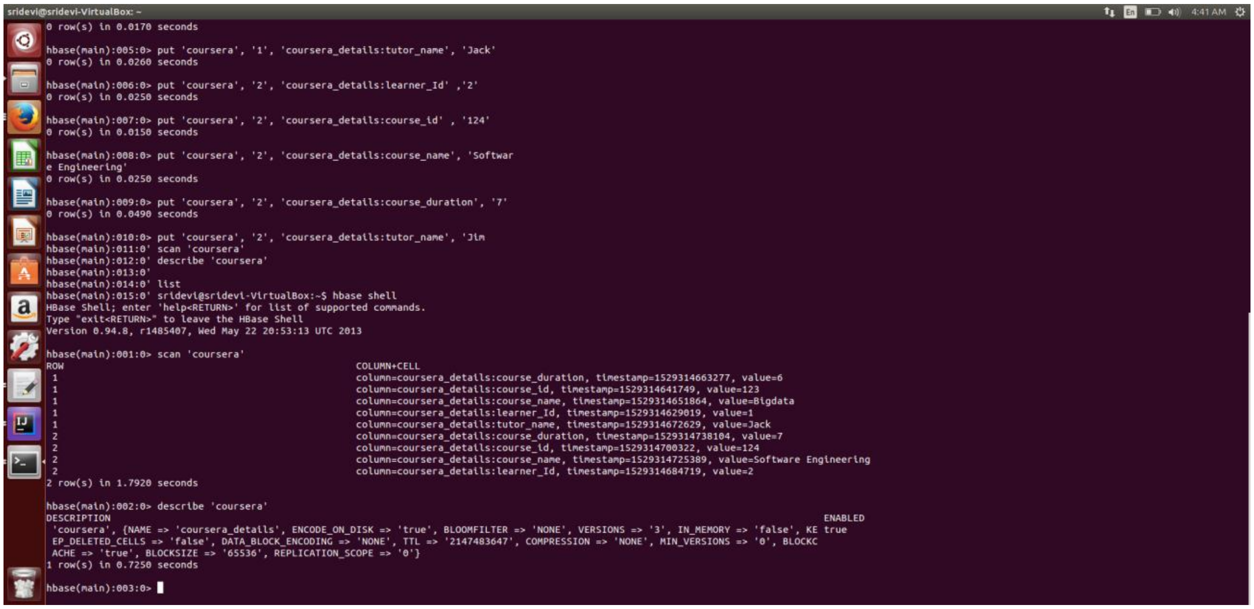
HBASE Implementation:

For solving this use case we opt to choose Couresera data set, the following are the fields and data types used for implementing the Hbase.

```
create 'coursera', 'coursera_details'


> put 'coursera', '1', 'coursera_details:learner_Id' ,'1'
> put 'coursera', '1', 'coursera_details:course_id' , '123'
> put 'coursera', '1', 'coursera_details:course_name', 'Bigdata'
> put 'coursera', '1', 'coursera_details:course_duration', '6'
> put 'coursera', '1', 'coursera_details:tutor_name', 'Jack'
> put 'coursera', '2', 'coursera_details:learner_Id' ,'2'
> put 'coursera', '2', 'coursera_details:course_id' , '124'
> put 'coursera', '2', 'coursera_details:course_name', 'Software Engineering'
> put 'coursera', '2', 'coursera_details:course_duration', '7'
> put 'coursera', '2', 'coursera_details:tutor_name', 'Jim'




updating

put 'coursera', '2', 'coursera_details:course_name', 'Sattistical Learning'



Selecting:

get 'coursera', '1'
get 'coursera', '2'


Deleting:

deleteall 'coursera','1'
```

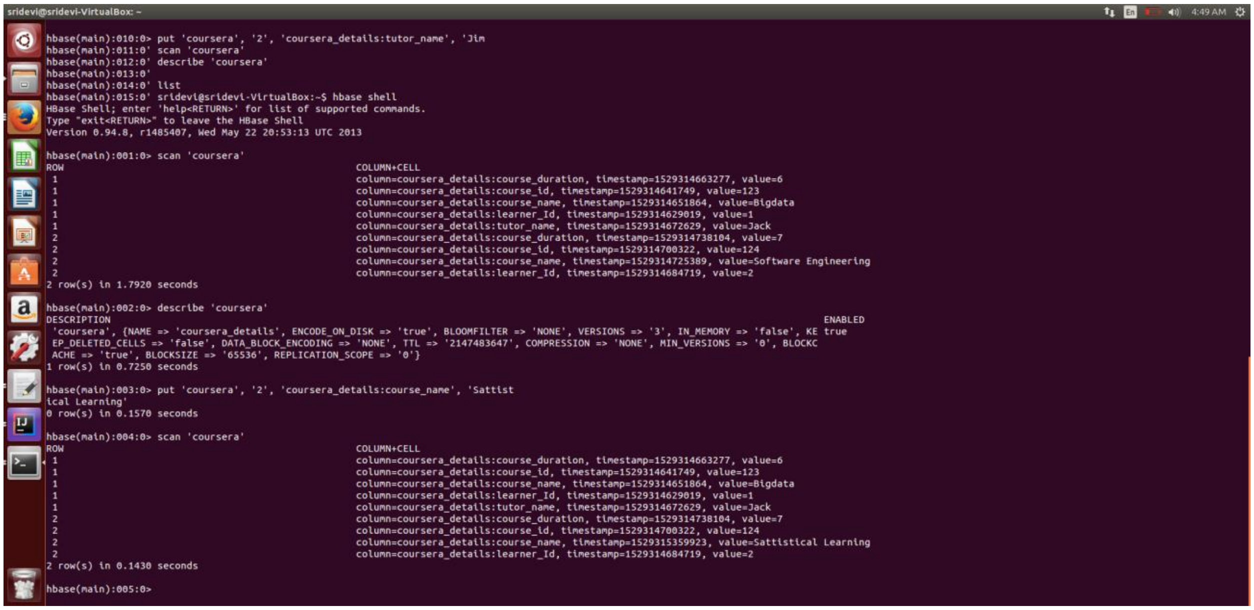
Output Screen shots for HBase:

Table creation:



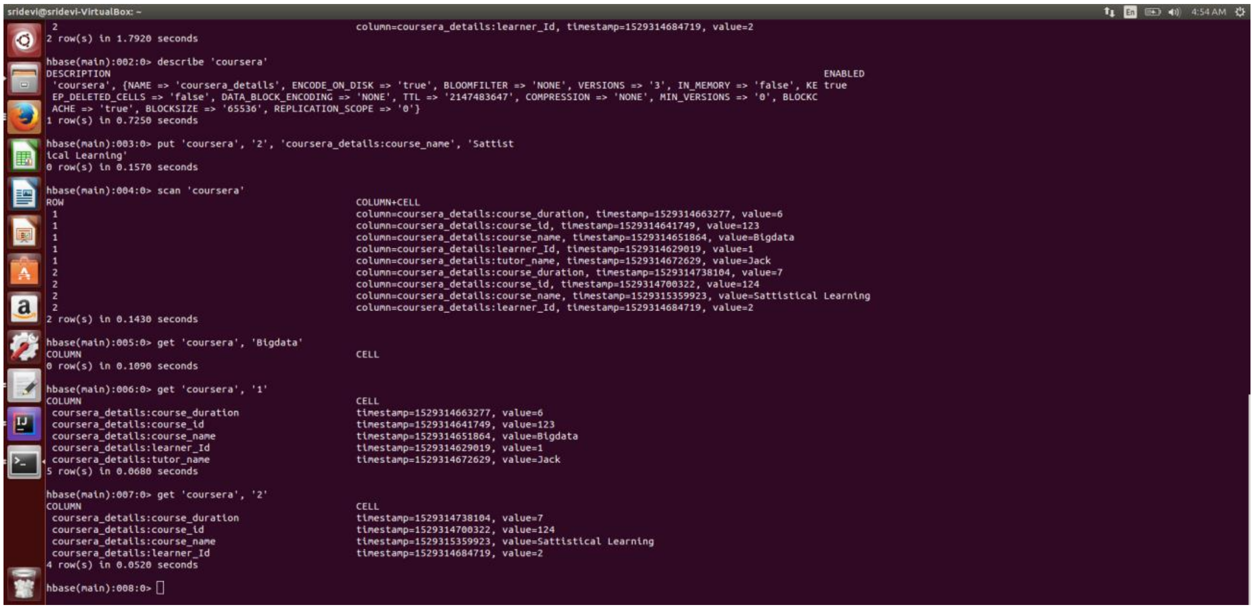
```
0 row(s) in 0.0170 seconds

hbase(main):005:0> put 'coursera', '1', 'coursera_details:tutor_name', 'Jack'
0 row(s) in 0.0260 seconds

hbase(main):006:0> put 'coursera', '2', 'coursera_details:learner_Id' ,'2'
0 row(s) in 0.0250 seconds

hbase(main):007:0> put 'coursera', '2', 'coursera_details:course_id' , '124'
0 row(s) in 0.0150 seconds

hbase(main):008:0> put 'coursera', '2', 'coursera_details:course_name', 'Softwar
e Engineering'
0 row(s) in 0.0250 seconds

hbase(main):009:0> put 'coursera', '2', 'coursera_details:course_duration', '7'
0 row(s) in 0.0490 seconds

hbase(main):010:0> put 'coursera', '2', 'coursera_details:tutor_name', 'Jim
hbase(main):011:0' scan 'coursera'
hbase(main):012:0' describe 'coursera'
hbase(main):013:0'
hbase(main):014:0' list
hbase(main):015:0' sridevi@sridevi-VirtualBox:~$ hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.94.8, r1485407, Wed May 22 20:53:13 UTC 2013

hbase(main):001:0> scan 'coursera'
ROW                                                    COLUMN+CELL
 1                                                     column=coursera_details:course_duration, timestamp=1529314663277, value=6
 1                                                     column=coursera_details:course_id, timestamp=1529314641749, value=123
 1                                                     column=coursera_details:course_name, timestamp=1529314651864, value=Bigdata
 1                                                     column=coursera_details:learner_Id, timestamp=1529314629019, value=1
 1                                                     column=coursera_details:tutor_name, timestamp=1529314672629, value=Jack
 2                                                     column=coursera_details:course_duration, timestamp=1529314738104, value=7
 2                                                     column=coursera_details:course_id, timestamp=1529314700322, value=124
 2                                                     column=coursera_details:course_name, timestamp=1529314725389, value=Software Engineering
 2                                                     column=coursera_details:learner_Id, timestamp=1529314684719, value=2
2 row(s) in 1.7920 seconds

hbase(main):002:0> describe 'coursera'
DESCRIPTION                                                                                                                  ENABLED
 'coursera', {NAME => 'coursera_details', ENCODE_ON_DISK => 'true', BLOOMFILTER => 'NONE', VERSIONS => '3', IN_MEMORY => 'false', KE true
 EP_DELETED_CELLS => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => '2147483647', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKC
 ACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.7250 seconds

hbase(main):003:0> 
```

Query 1- UPDATE DATA



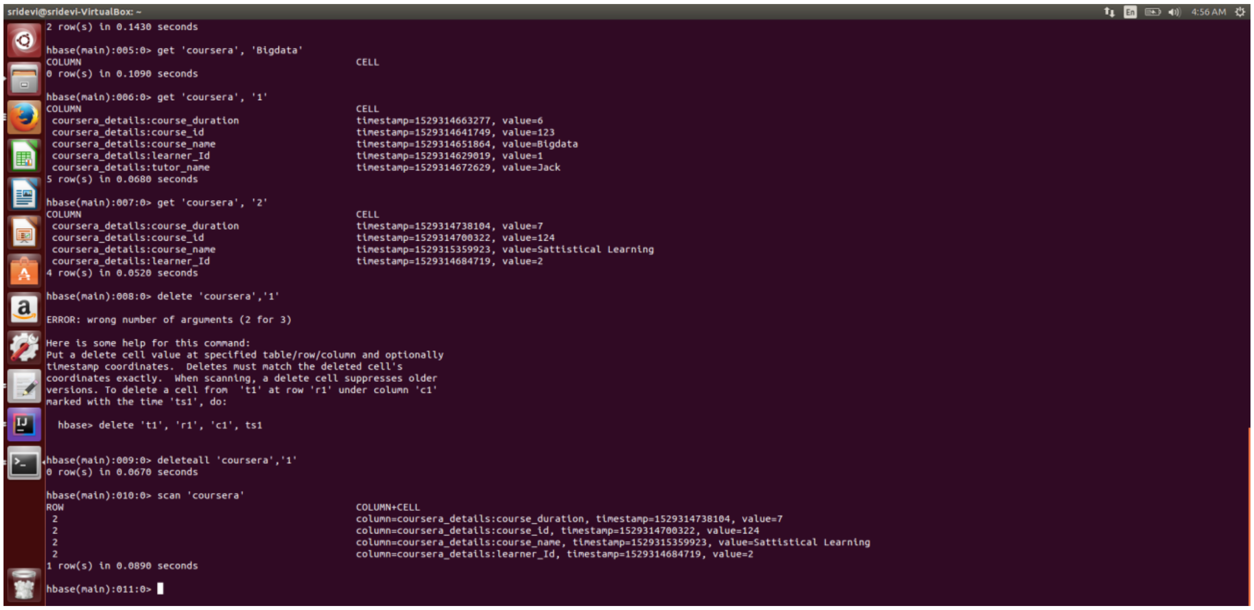
```
hbase(main):010:0> put 'coursera', '2', 'coursera_details:tutor_name', 'Jim
hbase(main):011:0' scan 'coursera'
hbase(main):012:0' describe 'coursera'
hbase(main):013:0'
hbase(main):014:0' list
hbase(main):015:0' sridevi@sridevi-VirtualBox:~$ hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.94.8, r1485407, Wed May 22 20:53:13 UTC 2013

hbase(main):001:0> scan 'coursera'
ROW                                                  COLUMN+CELL
 1                                                   column=coursera_details:course_duration, timestamp=1529314663277, value=6
 1                                                   column=coursera_details:course_id, timestamp=1529314641749, value=123
 1                                                   column=coursera_details:course_name, timestamp=1529314651864, value=Bigdata
 1                                                   column=coursera_details:learner_Id, timestamp=1529314629019, value=1
 1                                                   column=coursera_details:tutor_name, timestamp=1529314672629, value=Jack
 2                                                   column=coursera_details:course_duration, timestamp=1529314738104, value=7
 2                                                   column=coursera_details:course_id, timestamp=1529314700322, value=124
 2                                                   column=coursera_details:course_name, timestamp=1529314725389, value=Software Engineering
 2                                                   column=coursera_details:learner_Id, timestamp=1529314684719, value=2
2 row(s) in 1.7920 seconds

hbase(main):002:0> describe 'coursera'
DESCRIPTION                                                                                                                          ENABLED
 'coursera', {NAME => 'coursera_details', ENCODE_ON_DISK => 'true', BLOOMFILTER => 'NONE', VERSIONS => '3', IN_MEMORY => 'false', KE true
 EP_DELETED_CELLS => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => '2147483647', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKC
 ACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.7250 seconds

hbase(main):003:0> put 'coursera', '2', 'coursera_details:course_name', 'Sattist
ical Learning'
0 row(s) in 0.1570 seconds

hbase(main):004:0> scan 'coursera'
ROW                                                  COLUMN+CELL
 1                                                   column=coursera_details:course_duration, timestamp=1529314663277, value=6
 1                                                   column=coursera_details:course_id, timestamp=1529314641749, value=123
 1                                                   column=coursera_details:course_name, timestamp=1529314651864, value=Bigdata
 1                                                   column=coursera_details:learner_Id, timestamp=1529314629019, value=1
 1                                                   column=coursera_details:tutor_name, timestamp=1529314672629, value=Jack
 2                                                   column=coursera_details:course_duration, timestamp=1529314738104, value=7
 2                                                   column=coursera_details:course_id, timestamp=1529314700322, value=124
 2                                                   column=coursera_details:course_name, timestamp=1529315359923, value=Sattistical Learning
 2                                                   column=coursera_details:learner_Id, timestamp=1529314684719, value=2
2 row(s) in 0.1430 seconds

hbase(main):005:0>
```

Query 2- SELECT DATA



```
 2                                                   column=coursera_details:learner_Id, timestamp=1529314684719, value=2
2 row(s) in 1.7920 seconds

hbase(main):002:0> describe 'coursera'
DESCRIPTION                                                                                                                          ENABLED
 'coursera', {NAME => 'coursera_details', ENCODE_ON_DISK => 'true', BLOOMFILTER => 'NONE', VERSIONS => '3', IN_MEMORY => 'false', KE true
 EP_DELETED_CELLS => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => '2147483647', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKC
 ACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.7250 seconds

hbase(main):003:0> put 'coursera', '2', 'coursera_details:course_name', 'Sattist
ical Learning'
0 row(s) in 0.1570 seconds

hbase(main):004:0> scan 'coursera'
ROW                                                  COLUMN+CELL
 1                                                   column=coursera_details:course_duration, timestamp=1529314663277, value=6
 1                                                   column=coursera_details:course_id, timestamp=1529314641749, value=123
 1                                                   column=coursera_details:course_name, timestamp=1529314651864, value=Bigdata
 1                                                   column=coursera_details:learner_Id, timestamp=1529314629019, value=1
 1                                                   column=coursera_details:tutor_name, timestamp=1529314672629, value=Jack
 2                                                   column=coursera_details:course_duration, timestamp=1529314738104, value=7
 2                                                   column=coursera_details:course_id, timestamp=1529314700322, value=124
 2                                                   column=coursera_details:course_name, timestamp=1529315359923, value=Sattistical Learning
 2                                                   column=coursera_details:learner_Id, timestamp=1529314684719, value=2
2 row(s) in 0.1430 seconds

hbase(main):005:0> get 'coursera', 'Bigdata'
COLUMN                                               CELL
0 row(s) in 0.1090 seconds

hbase(main):006:0> get 'coursera', '1'
COLUMN                                               CELL
 coursera_details:course_duration                    timestamp=1529314663277, value=6
 coursera_details:course_id                          timestamp=1529314641749, value=123
 coursera_details:course_name                        timestamp=1529314651864, value=Bigdata
 coursera_details:learner_Id                         timestamp=1529314629019, value=1
 coursera_details:tutor_name                         timestamp=1529314672629, value=Jack
5 row(s) in 0.0680 seconds

hbase(main):007:0> get 'coursera', '2'
COLUMN                                               CELL
 coursera_details:course_duration                    timestamp=1529314738104, value=7
 coursera_details:course_id                          timestamp=1529314700322, value=124
 coursera_details:course_name                        timestamp=1529315359923, value=Sattistical Learning
 coursera_details:learner_Id                         timestamp=1529314684719, value=2
4 row(s) in 0.0520 seconds

hbase(main):008:0> 
```

Query 3- DELETE DATA



```
2 row(s) in 0.1430 seconds

hbase(main):005:0> get 'coursera', 'Bigdata'
COLUMN                                                    CELL
0 row(s) in 0.1090 seconds

hbase(main):006:0> get 'coursera', '1'
COLUMN                                                    CELL
 coursera_details:course_duration                         timestamp=1529314663277, value=6
 coursera_details:course_id                               timestamp=1529314641749, value=123
 coursera_details:course_name                             timestamp=1529314651864, value=Bigdata
 coursera_details:learner_Id                              timestamp=1529314629019, value=1
 coursera_details:tutor_name                              timestamp=1529314672629, value=Jack
5 row(s) in 0.0680 seconds

hbase(main):007:0> get 'coursera', '2'
COLUMN                                                    CELL
 coursera_details:course_duration                         timestamp=1529314738104, value=7
 coursera_details:course_id                               timestamp=1529314700322, value=124
 coursera_details:course_name                             timestamp=1529315359923, value=Sattistical Learning
 coursera_details:learner_Id                              timestamp=1529314684719, value=2
4 row(s) in 0.0520 seconds

hbase(main):008:0> delete 'coursera','1'

ERROR: wrong number of arguments (2 for 3)

Here is some help for this command:
Put a delete cell value at specified table/row/column and optionally
timestamp coordinates.  Deletes must match the deleted cell's
coordinates exactly.  When scanning, a delete cell suppresses older
versions. To delete a cell from  't1' at row 'r1' under column 'c1'
marked with the time 'ts1', do:

  hbase> delete 't1', 'r1', 'c1', ts1


hbase(main):009:0> deleteall 'coursera','1'
0 row(s) in 0.0670 seconds

hbase(main):010:0> scan 'coursera'
ROW                                                       COLUMN+CELL
 2                                                        column=coursera_details:course_duration, timestamp=1529314738104, value=7
 2                                                        column=coursera_details:course_id, timestamp=1529314700322, value=124
 2                                                        column=coursera_details:course_name, timestamp=1529315359923, value=Sattistical Learning
 2                                                        column=coursera_details:learner_Id, timestamp=1529314684719, value=2
1 row(s) in 0.0890 seconds

hbase(main):011:0> █
```

No Sql Comparison over HBase and Cassandra

The objective behind implementing the No Sql systems is to obtain data volume, and to overcome the limitations of the Relational model.

Cassandra:

No single point failure

Scalable, High performance

HBase:

Open source

Non-relational

NoSQL Performance can be varied when taken the load and throughput in to consideration.

Operations	CASSANDRA	HBase
Update	Update coursera.coursera_details set course_duartion =5 ,course_name='Introduction to Statstical Learning' where learner_id=3;	put 'coursera', '2', 'coursera_details:course_name', 'Sattistical Learning'
Select	select tutor_name from coursera.coursera_details where course_name='Cloud Computing' ALLOW FILTERING;	get 'coursera', '1' get 'coursera', '2'
Delete	DELETE from	deleteall 'coursera','2'

	coursera.coursera_details where learner_id = 2;	

For the above sample test case data set for coursera when taken in to consideration- **Cassandra's fast write and read performance** has been observed when compared to HBase

Conclusion:

As stated, the above workflow with certain set of parameters is followed in solving the execution by implementing the core and basic concepts of Map reduce algorithm, Cassandra, HBase and Hadoop eco file system.

Source code link:

https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/tree/master/Lab%201/Source%20Code

Video Link: https://youtu.be/cJYrfReYFW0

Wiki Link:

https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/wiki/Lab-1-Assignment