-----------------------------------------------------------------------------------------------------------------------------

CS5590 – Big Data (Hadoop/Spark) – Lab Assignment 1 - Report

#10, Avni Mehta

-----------------------------------------------------------------------------------------------------------------------------

**Task 1**

**Objective**: Create a map reduce program to find mutual friends for the following input

A -> B C D

B -> A C D E

C -> A B D E

D -> A B C E

E -> B C D

**Algorithm**:

I am doing the following steps:

Mapper:

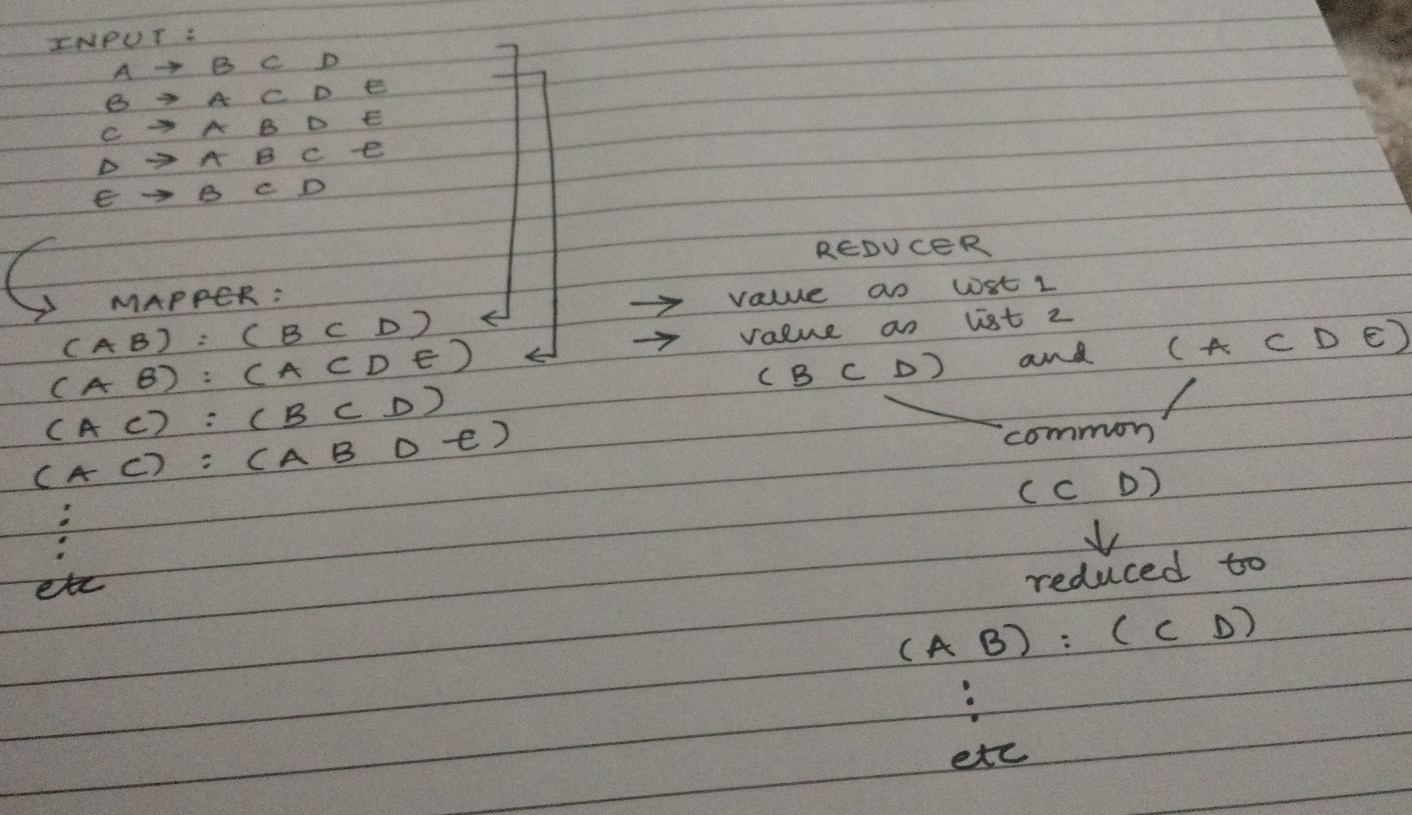
1. Split the input file line by line
2. Split the line by ‘->’ format. The left side of the split gives the user name and the right side gives all the friends.
3. The right split can be further split by space to get a list of friends.
4. For each friend f, store the key as sorted (user, f)
5. The mapper program will output (key, value) where value is the original list of friends from the right split.

Reducer:

1. Store the value into lists.
2. Compare the lists to find common elements
3. Output (key, value) where value is the common friends

Main Program:

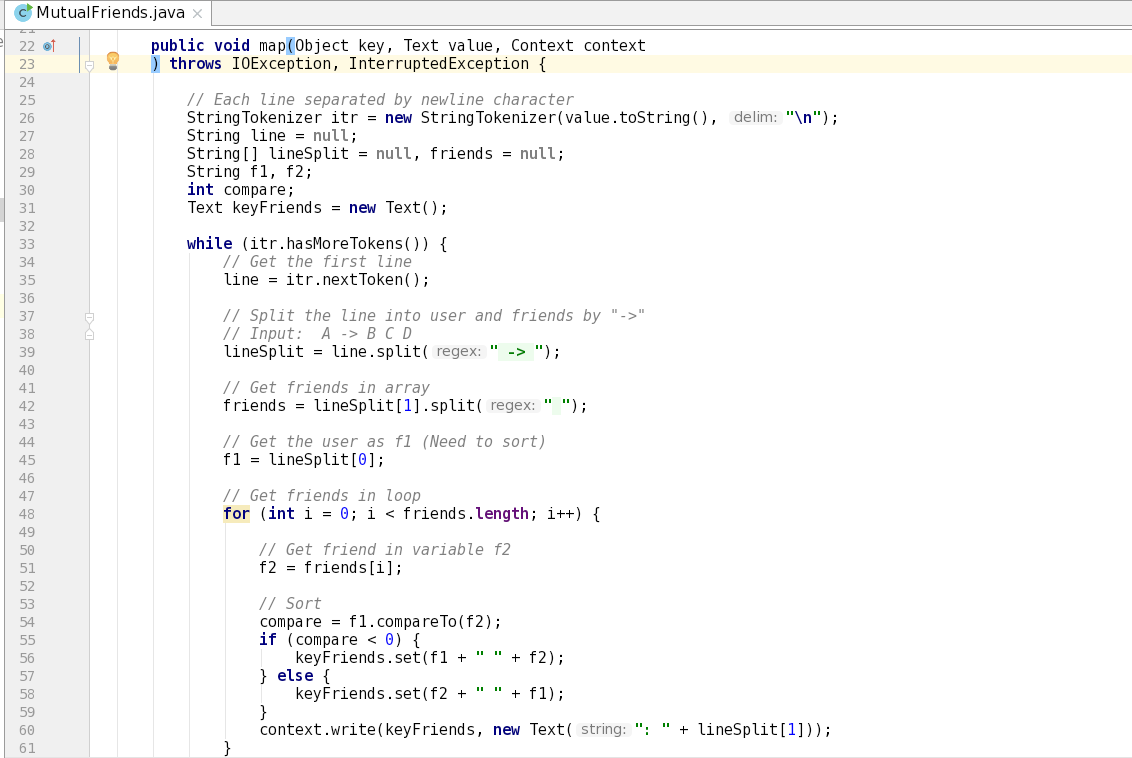
The main program is very much like the word count program.



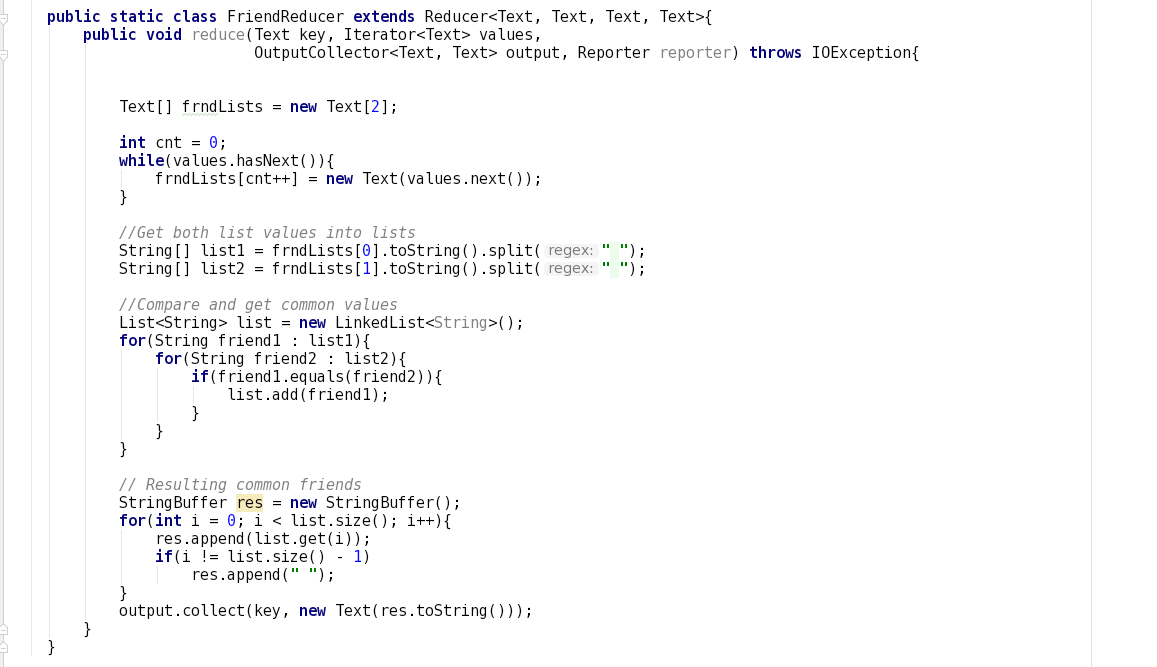
**Code:**

Comments are provided for each important step in the code.

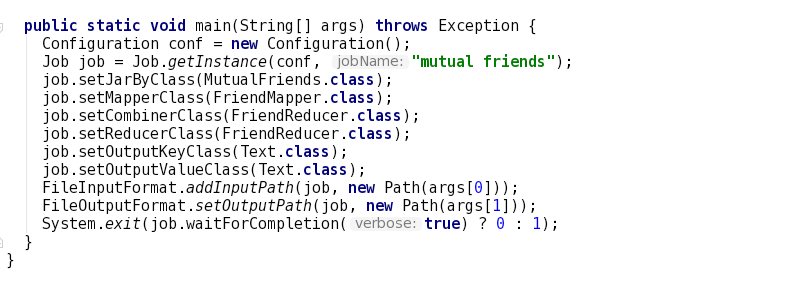
Mapper:

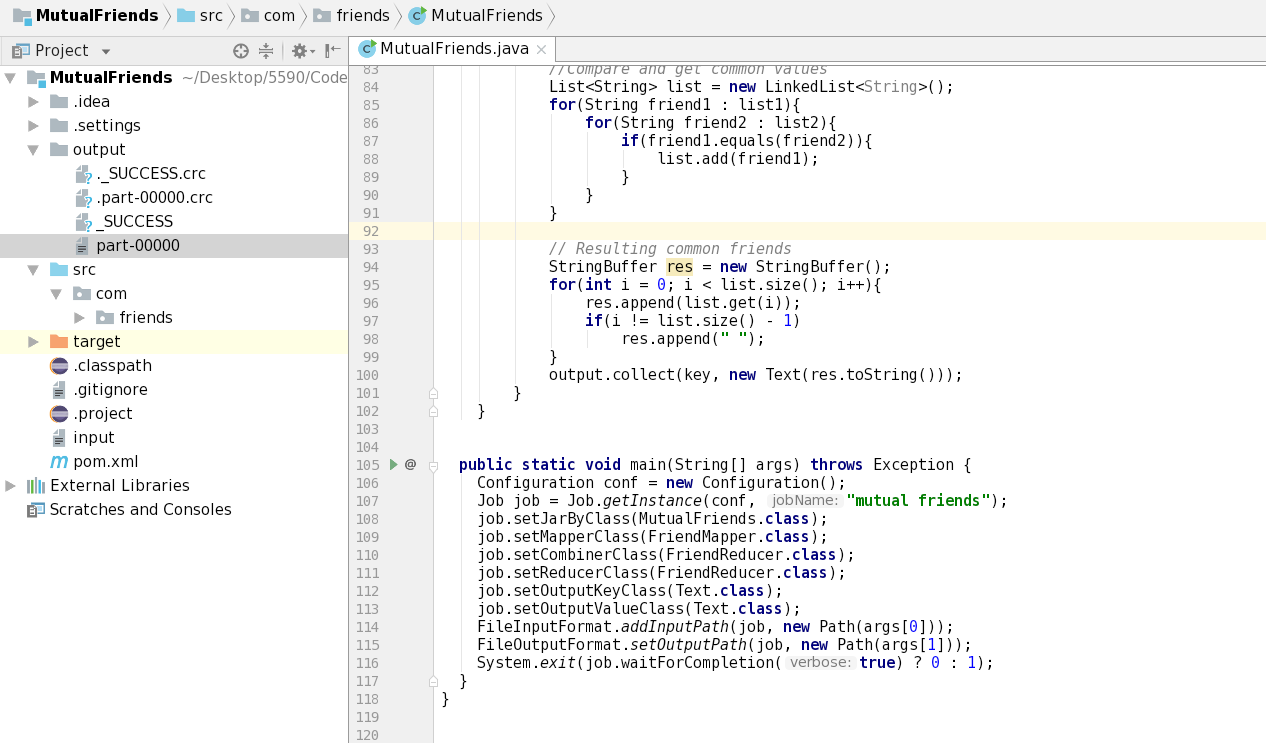


Reducer:

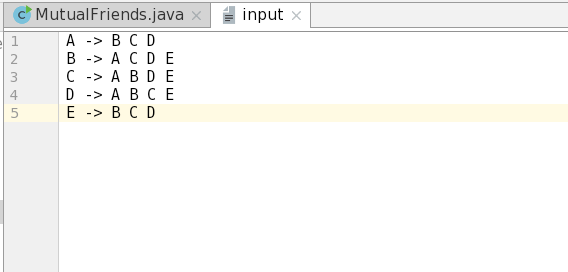


Main:

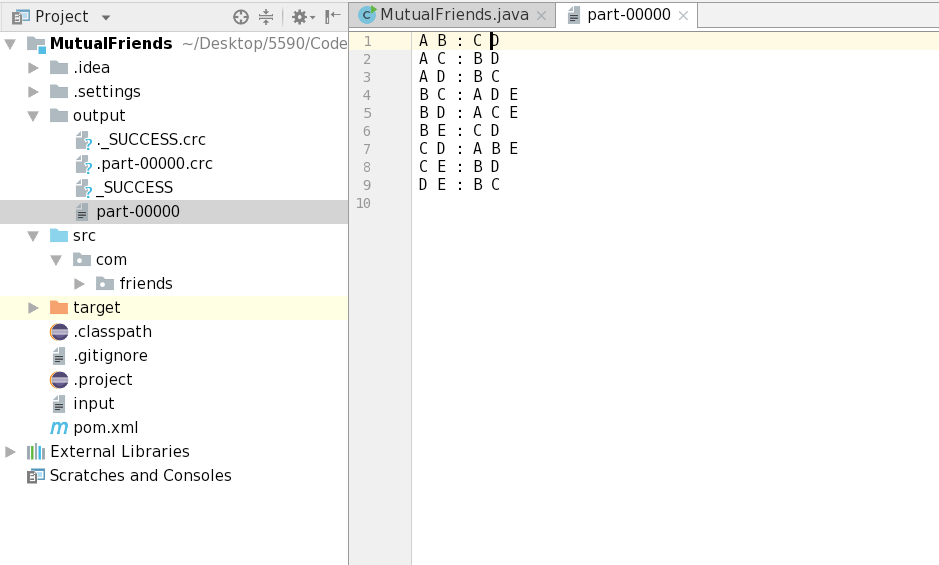




**Input:**

****

**Output:**

****

**Task 2**

**Objective:**

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.

Use Case Considered = Coursera

Problem: Need a high performing and highly scalable database where new features can be easily added.

Solution: They chose Cassandra. But in this task, we will compare both HBase and Cassandra

Dataset:

The table should contain columns like user\_id, user\_name, course\_id, course\_name, course\_completion\_status, course\_start\_date etc.

The user\_id and course\_id columns would be numeric and the user\_name, course\_name and course\_completion\_status columns would be text. Column course\_start\_date is date type.

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.

HBase table creation:

create 'user\_courses', 'user\_info', 'course\_info'

HBase insert rows:

put 'user\_courses', '1', 'user\_info:user\_id', '2341'

put 'user\_courses', '1', 'user\_info:user\_name', 'Avni Mehta'

put 'user\_courses', '1', 'course\_info:course\_id', '21'

put 'user\_courses', '1', 'course\_info:course\_name', 'Python'

put 'user\_courses', '1', 'course\_info:course\_completion\_status', 'In Progress'

put 'user\_courses', '1', 'course\_info:course\_start\_date', '04/23/2018'

put 'user\_courses', '2', 'user\_info:user\_id', '2341'

put 'user\_courses', '2', 'user\_info:user\_name', 'Avni Mehta'

put 'user\_courses', '2', 'course\_info:course\_id', '341'

put 'user\_courses', '2', 'course\_info:course\_name', 'AWS'

put 'user\_courses', '2', 'course\_info:course\_completion\_status', 'Complete'

put 'user\_courses', '2', 'course\_info:course\_start\_date', '06/18/2018'

put 'user\_courses', '3', 'user\_info:user\_id', '10'

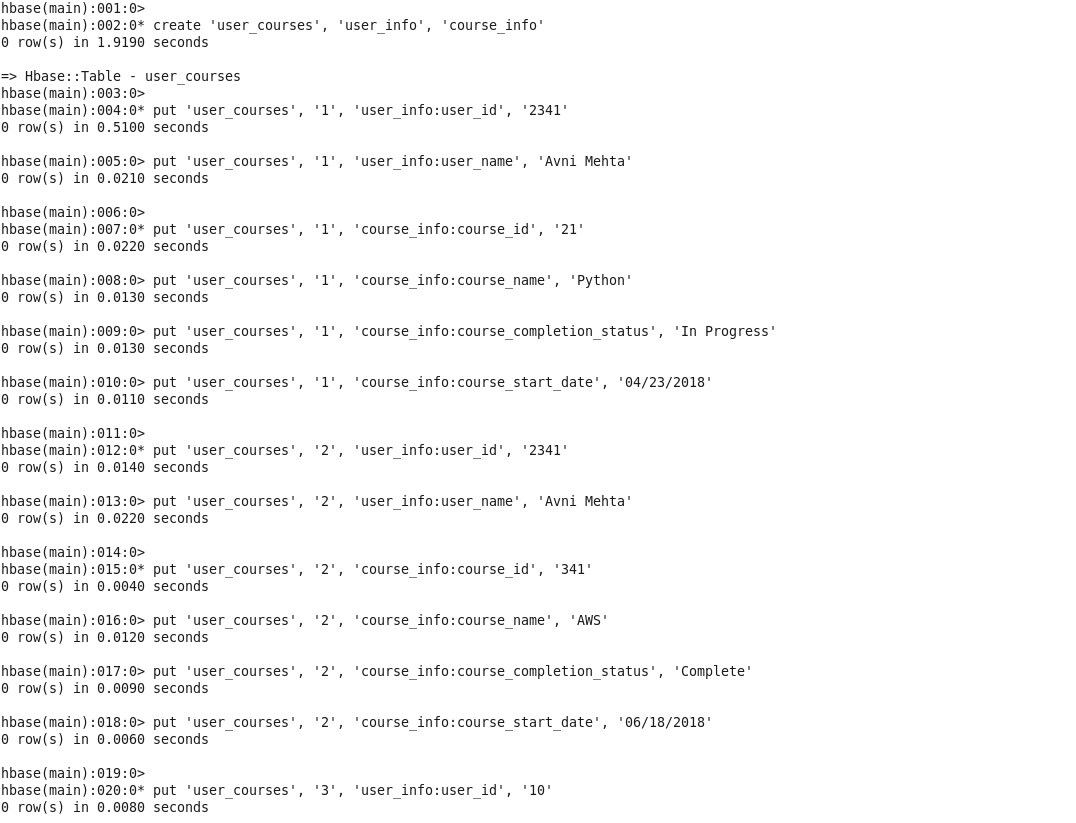
put 'user\_courses', '3', 'user\_info:user\_name', 'Hardik Mehta'

put 'user\_courses', '3', 'course\_info:course\_id', '21'

put 'user\_courses', '3', 'course\_info:course\_name', 'Python'

put 'user\_courses', '3', 'course\_info:course\_completion\_status', 'Complete'

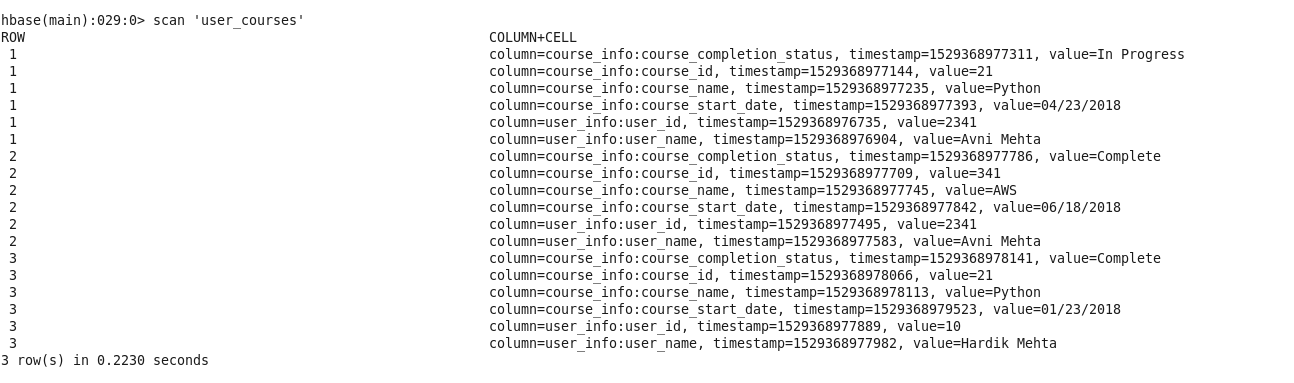
put 'user\_courses', '3', 'course\_info:course\_start\_date', '01/23/2018'



HBase queries:

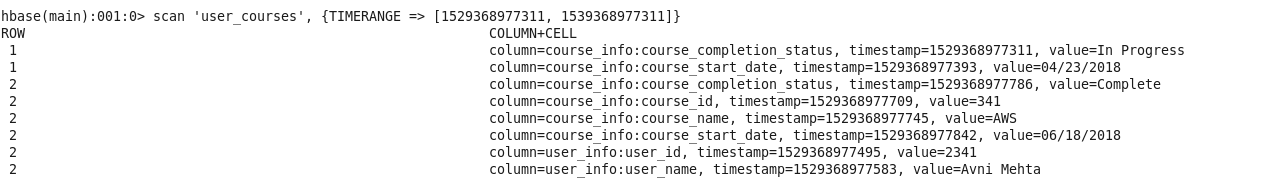
# Scan all rows of table

scan 'user\_courses'



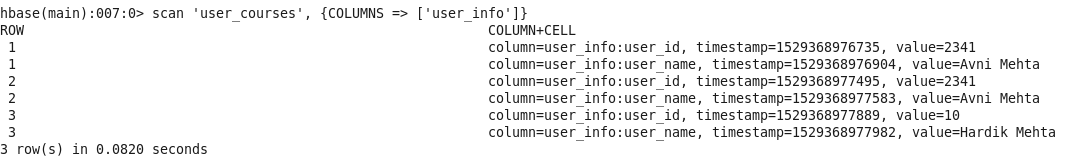
# Get all rows in a specific a timerange (eg. courses enrolled today)

scan 'user\_courses', {TIMERANGE => [1303668804, 1303668904]}



# Get user name and course name

scan 'user\_courses', {COLUMNS => ['user\_name', 'course\_name']}



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.

Table Creation

create keyspace coursera with replication={'class':'SimpleStrategy', 'replication\_factor':1};

USE coursera;

CREATE TABLE user\_courses (

user\_id int,

user\_name text,

course\_id int,

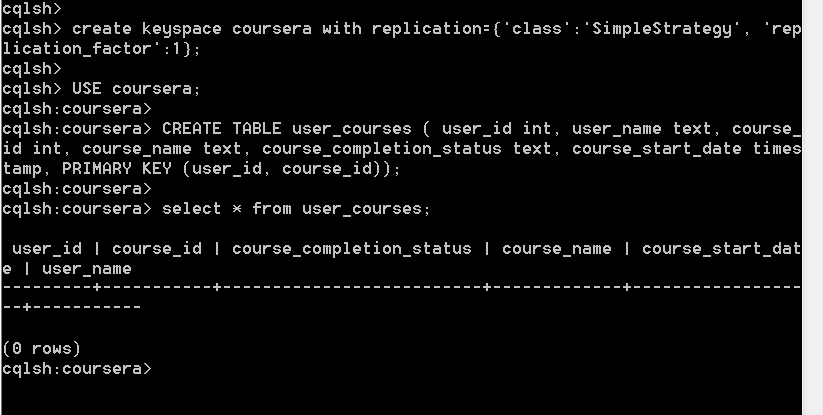
course\_name text,

course\_completion\_status text,

course\_start\_date timestamp,

PRIMARY KEY(user\_id, course\_id)

);



Insert Records

INSERT INTO user\_courses (user\_id, user\_name, course\_id, course\_name, course\_completion\_status, course\_start\_date)

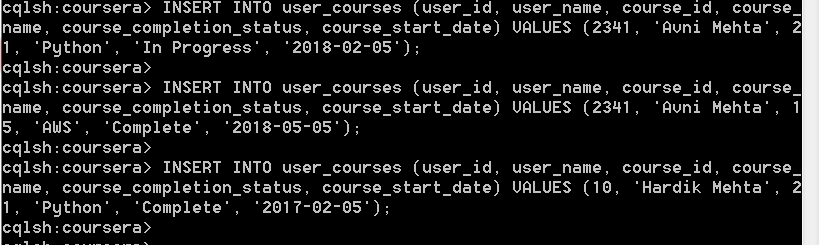
VALUES (2341, 'Avni Mehta', 21, 'Python', 'In Progress', '2018-02-05');

INSERT INTO user\_courses (user\_id, user\_name, course\_id, course\_name, course\_completion\_status, course\_start\_date)

VALUES (2341, 'Avni Mehta', 15, 'AWS', 'Complete', '2018-05-05');

INSERT INTO user\_courses (user\_id, user\_name, course\_id, course\_name, course\_completion\_status, course\_start\_date)

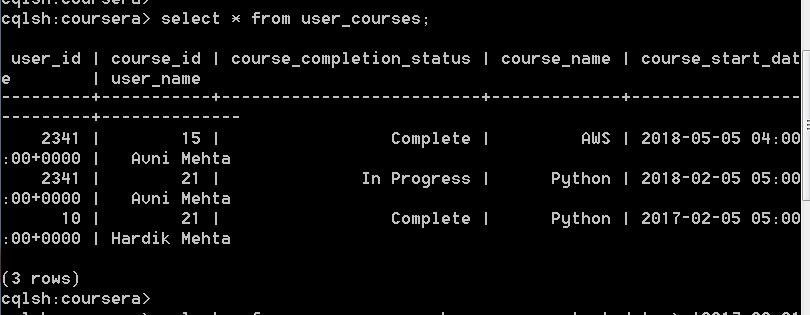
VALUES (10, 'Hardik Mehta', 21, 'Python', 'Complete', '2017-05-01');



Cassandra Queries

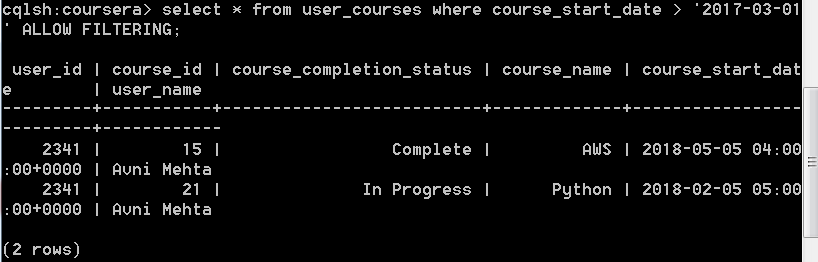
# Scan all rows of table

select \* from user\_courses



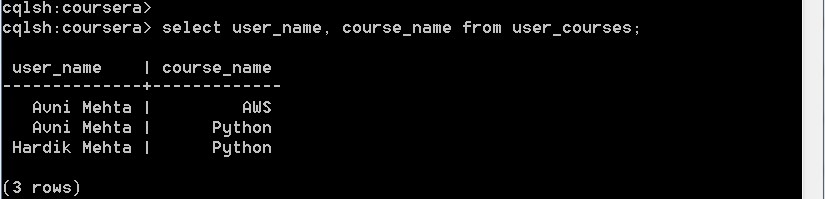
# Get all rows in a specific a timerange

select \* from user\_courses where course\_start\_date > '2017-05-01' ALLOW FILTERING;



# Get user name and course name

select user\_name, course\_name from user\_courses;



d) Compare Cassandra and HBase for your use case

|  |  |
| --- | --- |
| **Cassandra** | **HBase** |
| High availability (No single point of failure) | Master-slave model |
| Flexible parsing/wide column requirement | Semi-structured content management (high scalability) |
| Eventually consistent | Highly consistent |
| Minimal administration |  |
| Fast random read/write | High-speed logging, caching |
| AP on CAP | CP on CAP |

Reference: <https://www.linkedin.com/pulse/real-comparison-nosql-databases-hbase-cassandra-mongodb-sahu/>

**Demo Video:**

Video for the demo can be found at <https://youtu.be/jPJVIpPsUOM>