CSP554—Big Data Technologies

Assignment #11

Worth: 5 points (1 point for each problem)

Due by the start of the next class period

Assignments should be uploaded via the Blackboard portal.

Readings

NoSQL Distilled: Chapters 8 and 10

Starting HBase

To start HBase first establish ssh tunneling as usual. Then access the Ambari Hadoop admin console by pointing your browser to localhost:8080. Log on Ambari with username and password maria_dev. In the upper right you should see a selection 'Services.' Click on it and choose 'HBase.' Then, from the HBase page choose 'Service Actions' and select start. Then log on to maria_dev and enter 'hbase shell.'

There are two documents about the HBase shell on the blackboard in the "Free Books and Chapters" section that will help with the assignment: HBase Shell, HBase Shell Commands.

Exercises

Exercise 1)

Create an HBase table with the following characteristics

Table Name: csp554Tbl

First column family: cf1

Second column family: cf2

Then execute the DESCRIBE command on the table and return command you wrote and the output as the results of this exercise.

Answer:

create ‘csp554Tbl’, ‘cf1’, ‘cf2’
describe ‘csp554Tbl’

```
hbase(main):003:0> create 'csp554Tbl', 'cf1', 'cf2'
0 row(s) in 2.4910 seconds

=> Hbase::Table - csp554Tbl
hbase(main):004:0> DESCRIBE csp554Tbl
NameError: undefined local variable or method `csp554Tbl' for #<Object:0x7459a21e>

hbase(main):005:0> DESCRIBE 'csp554Tbl'
NoMethodError: undefined method `DESCRIBE' for #<Object:0x7459a21e>

hbase(main):006:0> describe 'csp554Tbl'
Table csp554Tbl is ENABLED
csp554Tbl
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION
 => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'cf2', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION
 => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
2 row(s) in 0.1020 seconds
```

Exercise 2)

Put the following data into the table created in exercise 1:

Row Key	Column Family	Column (Qualifier)	Value
Row1	cf1	name	Sam
Row2	cf1	name	Ahmed
Row1	cf2	job	Pilot
Row2	cf2	job	Doctor
Row1	cf2	level	LZ3
Row2	cf2	level	AR7

Execute the SCAN command on this table returning all rows, column families and columns as the result of the exercise.

Answer:

put ‘csp554Tbl’,’Row1’,’cf1:name’,’Sam’
put ‘csp554Tbl’,’Row2’,’cf1:name’,’Ahmed’

put ‘csp554Tbl’,’Row1’,’cf2:job’,’Pilot’
put ‘csp554Tbl’,’Row2’,’cf2:job’,’Doctor’
put ‘csp554Tbl’,’Row1’,’cf2:level’,’LZ3’
put ‘csp554Tbl’,’Row2’,’cf2:level’,’AR7’
scan ‘csp554Tbl’

```
hbase(main):003:0> put 'csp554Tbl', 'Row1', 'cf1:name', 'Sam'
0 row(s) in 0.0910 seconds

hbase(main):004:0> put 'csp554Tbl', 'Row2', 'cf1:name', 'Ahmed'
0 row(s) in 0.0030 seconds

hbase(main):005:0> put 'csp554Tbl', 'Row1', 'cf2:job', 'Pilot'
0 row(s) in 0.0050 seconds

hbase(main):006:0> put 'csp554Tbl', 'Row2', 'cf2:job', 'Doctor'
0 row(s) in 0.0070 seconds

hbase(main):007:0> put 'csp554Tbl', 'Row1', 'cf2:level', 'LZ3'
0 row(s) in 0.0080 seconds

hbase(main):008:0> put 'csp554Tbl', 'Row2', 'cf2:level', 'AR7'
0 row(s) in 0.0080 seconds
```

```
hbase(main):009:0> scan 'csp554Tbl'
ROW                   COLUMN+CELL
 Row1                 column=cf1:name, timestamp=1542155065977, value=Sam
 Row1                 column=cf2:job, timestamp=1542155108720, value=Pilot
 Row1                 column=cf2:level, timestamp=1542155150339, value=LZ3
 Row2                 column=cf1:name, timestamp=1542155085137, value=Ahmed
 Row2                 column=cf2:job, timestamp=1542155130557, value=Doctor
 Row2                 column=cf2:level, timestamp=1542155165915, value=AR7
2 row(s) in 0.0260 seconds
```

Exercise 3)

Using the above table write command that will get the value associated with row (Row1), column family (cf1) and column/qualifier (level). Provide the command and its result as the output of this exercise.

Answer:

get ‘csp554Tbl’,’Row1’,{COLUMN => ‘cf1:level’}

```
hbase(main):010:0> get 'csp554Tbl', 'Row1', {COLUMN => 'cf1:level'}
COLUMN                CELL
 cf1:level            timestamp=1542155150339, value=LZ3
1 row(s) in 0.0270 seconds
```

Exercise 4)

Using the above table write command that will get the value associated with row (Row2), column family (cf1) and column/qualifier (name). Provide the command and its result as the output of this exercise.

Answer:

get 'csp554Tbl','Row2',{COLUMN=>'cf1:name'}

```
hbase(main):011:0> get 'csp554Tbl', 'Row2', {COLUMN => 'cf1:name'}
COLUMN                CELL
 cf1:name             timestamp=1542155085137, value=Ahmed
1 row(s) in 0.0070 seconds
```

Exercise 5)

Using the above table write a SCAN command that will return information about only one row using the LIMIT modifier. Provide the command and its result as the output of this exercise.

Answer:

scan 'csp554Tbl',{LIMIT => 1}

```
hbase(main):017:0> scan 'csp554Tbl', {COLUMN => ['cf1','cf2'], LIMIT =>1}
ROW                   COLUMN+CELL
 Row1                 column=cf1:name, timestamp=1542155065977, value=Sam
 Row1                 column=cf2:job, timestamp=1542155108720, value=Pilot
 Row1                 column=cf2:level, timestamp=1542155150339, value=LZ3
1 row(s) in 0.0150 seconds
```