Lab 06 – Introduction to HBASE Commands

Full name: Chloe Tee Rouyi
Student ID: 0354731

Important Note: You are required to perform the following tasks and provide a report based on your observations and understanding of the commands, the output, and any possible errors or additional information provided by the system.

Tasks:

1. Open terminal

2. Check running services: **sudo jps**

```
[cloudera@quickstart ~]$ sudo jps
6099 RESTServer
7920 Bootstrap
5587 Bootstrap
5708 NodeManager
6334 RunJar
5637 JobHistoryServer
5220 DataNode
6903 HistoryServer
5457 SecondaryNameNode
6870 Bootstrap
5299 JournalNode
5873 ResourceManager
6452 RunJar
5148 QuorumPeerMain
9096 org.eclipse.equinox.launcher_1.3.0.v20140415-2008.jar
5388 NameNode
7995
15226 Jps
7717 Bootstrap
7956
6223 ThriftServer
```

sudo = admin command, **jps** = list of services running

3. Execute the following HBASE commands to understand the database setting.
    a. Start "HBase Master Server" service: **sudo service hbase-master start**

```
[cloudera@quickstart ~]$ sudo service hbase-master start
starting master, logging to /var/log/hbase/hbase-hbase-master-quickstart.cloudera.out
Started HBase master daemon (hbase-master):                [  OK  ]
[cloudera@quickstart ~]$ sudo jps
6099 RESTServer
7920 Bootstrap
5587 Bootstrap
15390 HMaster
5708 NodeManager
6334 RunJar
5637 JobHistoryServer
5220 DataNode
6903 HistoryServer
5457 SecondaryNameNode
6870 Bootstrap
5299 JournalNode
15590 Jps
5873 ResourceManager
6452 RunJar
5148 QuorumPeerMain
5388 NameNode
7995
7717 Bootstrap
7956
6223 ThriftServer
```

    b. Start "HBase Region Server" service: **sudo hbase-regionserver start** **Note:** If the above command doesn't work, then you may try below command or

```
[cloudera@quickstart ~]$ sudo hbase-regionserver start
sudo: hbase-regionserver: command not found
```

Error occurred as **sudo hbase-regionserver start** is not the correct syntax

```
[cloudera@quickstart ~]$ sudo service hadoop-hbase-regionserver start
hadoop-hbase-regionserver: unrecognized service
```

Error occurred as **sudo service hadoop-hbase-regionserver start** is not the correct syntax

```
[cloudera@quickstart ~]$ sudo service hbase-regionserver start
Starting Hadoop HBase regionserver daemon: starting regionserver, logging to /var/log/hbase/hbase-hbase-regionserver-quick
start.cloudera.out
hbase-regionserver.
```

'sudo service hbase-regionserver start' is the correct syntax

c. follow the above link: **sudo service hadoop-hbase-regionserver start**

```
[cloudera@quickstart ~]$ sudo service hadoop-hbase-regionserver start
hadoop-hbase-regionserver: unrecognized service
```

Error occurred as **sudo service hadoop-hbase-regionserver start** is not the correct

d. hbase shell to start the hbase: **hbase shell**

```
[cloudera@quickstart ~]$ hbase shell
2024-11-07 01:14:07,542 INFO  [main] Configuration.deprecation: hadoop.native.li
b is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017
```

4. Use command **status** to get some information about the status of the system and the details of the servers running on it.

```
hbase(main):001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load
```

5. Use command **versions** to see the version of HBase available in the system.

```
hbase(main):002:0> versions
NameError: undefined local variable or method `versions' for #<Object:0xcfa5d7b>
```

'versions' is the incorrect syntax, **'version'** is the correct syntax

```
hbase(main):003:0> version
1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017
```

6. Use command **table_help** to get some information on what and how to use tablereferenced commands.

```
hbase(main):004:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get', etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:

  hbase> t.put 'r', 'cf:q', 'v'

which puts a row 'r' with column family 'cf', qualifier 'q' and value 'v' into table t.

To read the data out, you can scan the table:

  hbase> t.scan

which will read all the rows in table 't'.

Essentially, any command that takes a table name can also be done via table reference.
Other commands include things like: get, delete, deleteall,
get_all_columns, get_counter, count, incr. These functions, along with
the standard JRuby object methods are also available via tab completion.

For more information on how to use each of these commands, you can also just type:

   hbase> t.help 'scan'

which will output more information on how to use that command.

You can also do general admin actions directly on a table; things like enable, disable,
flush and drop just by typing:

   hbase> t.enable
   hbase> t.flush
   hbase> t.disable
   hbase> t.drop

Note that after dropping a table, your reference to it becomes useless and further usage
is undefined (and not recommended).
```

7. Use command **whoami** to get some information about the current user of HBase.

```
hbase(main):005:0> whoami
cloudera (auth:SIMPLE)
    groups: cloudera, default
```

8. Creating/Updating Hbase tables:
    a. **create 't1','cf'**

```
hbase(main):006:0> create 't1','c1'
0 row(s) in 1.3430 seconds

=> Hbase::Table - t1
```

9. see the list of available/created tables: **list**

```
hbase(main):007:0> list
TABLE
t1
1 row(s) in 0.0200 seconds

=> ["t1"]
```

10. create another table: **create 'htest','cf'**

```
hbase(main):008:0> create 'htest','cf'
0 row(s) in 1.2330 seconds

=> Hbase::Table - htest
```

11. Store Data
    a. **put 'htest' , 'r1' , 'cf:c1' , 'v1'**
    b. **put 'htest' , 'r1' , 'cf:c2' , 'v2'**
    c. **put 'htest' , 'r1' , 'cf:c3' , 'v3'**

```
hbase(main):009:0> put 'htest','r1','cf:c1','v1'
0 row(s) in 0.1070 seconds

hbase(main):010:0> put 'htest','r1','cf:c2','v2'
0 row(s) in 0.0130 seconds

hbase(main):011:0> put 'htest','r1','cf:c3','v3'
0 row(s) in 0.0050 seconds
```

12. get the value from hbase

a. **scan 'htest'**

```
hbase(main):012:0> scan 'htest'
ROW                              COLUMN+CELL
 r1                              column=cf:c1, timestamp=1730971979591, value=v1
 r1                              column=cf:c2, timestamp=1730972000227, value=v2
 r1                              column=cf:c3, timestamp=1730972013769, value=v3
1 row(s) in 0.0490 seconds
```

b. **get 'htest' , 'r1'**

```
hbase(main):013:0> get 'htest','r1'
COLUMN                           CELL
 cf:c1                           timestamp=1730971979591, value=v1
 cf:c2                           timestamp=1730972000227, value=v2
 cf:c3                           timestamp=1730972013769, value=v3
3 row(s) in 0.0160 seconds
```

13. update/Modify

put 'htest' , 'r1', 'cf:c3' , 'v3_upadte'

put 'htest' , 'r1', 'cf:c3' , 'v3_update_David'

```
hbase(main):014:0> put 'htest', 'r1', 'cf:c3', 'v3_upadte'
0 row(s) in 0.0180 seconds

hbase(main):015:0> put 'htest', 'r1', 'cf:c3', 'v3_update_chloe'
0 row(s) in 0.0070 seconds
```

14. Use the following commands and investigate what will happen if you execute them:

a. **get 'htest' , 'r1' , {COLUMN=>'cf:c3',VERSIONS=>3}**

```
hbase(main):016:0> get 'htest', 'r1', {COLUMN=>'cf:c3',VERSIONS=>3}
COLUMN                           CELL
 cf:c3                           timestamp=1730972420553, value=v3_update_chloe
1 row(s) in 0.0130 seconds
```

b. **get 'htest' , 'r1' , {COLUMN=>'cf:c3',TIMESTAMP=> XXXXXXXXXXXX}**

```
hbase(main):019:0> get 'htest', 'r1', {COLUMN=>'cf:c3',TIMESTAMP=>1730972420553}
COLUMN                           CELL
 cf:c3                           timestamp=1730972420553, value=v3_update_chloe
1 row(s) in 0.0060 seconds
```

15. To alter a table:

step 1: Disable the table

disable 'htest'

```
hbase(main):020:0> disable 'htest'
0 row(s) in 2.2980 seconds
```

Step 2: Alter the table

alter 'htest' , {NAME=>'cf_altered'}

```
hbase(main):021:0> alter 'htest', {NAME=>'cf_altered'}
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9430 seconds
```

Step 3: enable the table

enable 'htest'

```
hbase(main):022:0> enable 'htest'
0 row(s) in 1.2510 seconds
```

16. To describe a table and get some information about it:

describe 'htest'

```
hbase(main):023:0> describe 'htest'
Table htest is ENABLED
htest
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION
 => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY => 'fals
e', BLOCKCACHE => 'true'}
{NAME => 'cf_altered', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', COMPRESSION => 'NONE
', VERSIONS => '1', TTL => 'FOREVER', MIN_VERSIONS => '0', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY
=> 'false', BLOCKCACHE => 'true'}
2 row(s) in 0.0270 seconds
```

17. Try this command:

put 'htest' , 'r1' , 'cf_altered:c3','david'

```
hbase(main):003:0> put 'htest', 'r1', 'cf_altered:c3', 'chloe'
0 row(s) in 0.0260 seconds
```

18. Retrieve all the records of the table:

scan 'htest'

```
hbase(main):004:0> scan 'htest'
ROW                             COLUMN+CELL
 r1                             column=cf:c1, timestamp=1730971979591, value=v1
 r1                             column=cf:c2, timestamp=1730972000227, value=v2
 r1                             column=cf:c3, timestamp=1730972420553, value=v3_update_chloe
 r1                             column=cf_altered:c3, timestamp=1730973388311, value=chloe
1 row(s) in 0.0290 seconds
```

19. Try to delete data

delete 'htest' , 'r1' , 'cf:c3'

```
hbase(main):005:0> delete 'htest','r1','cf:c3'
0 row(s) in 0.0250 seconds
```

20. To drop a table:

Step 1: **disable 'htest'**

```
hbase(main):006:0> disable 'htest'
0 row(s) in 2.2970 seconds
```

Step 2: **drop 'htest'**

```
hbase(main):007:0> drop 'htest'
0 row(s) in 1.2800 seconds
```

21. Check the list of tables:

Command: **list**

```
hbase(main):008:0> list
TABLE
t1
1 row(s) in 0.0140 seconds

=> ["t1"]
```

22. Save and submit your Lab documents in PDF with the following filename format.
Submit your report via the submission link for this available on MyTIMeS.
Filename format: Name_ID_Lab05