

## **Experiment No.: 02**

### **Aim:**

To install and configure HBase to execute NoSQL commands.

### **Theory:**

HBase is a column-oriented non-relational database management system that runs on top of the Hadoop Distributed File System (HDFS). HBase provides a fault-tolerant way of storing sparse data sets, which are common in many big data use cases. It is well suited for real-time data processing or random read/write access to large volumes of data.

Unlike relational database systems, HBase does not support a structured query language like SQL; in fact, HBase isn't a relational data store at all. HBase applications are written in Java™ much like a typical Apache

MapReduce application. HBase does support writing applications in Apache Avro, REST, and Thrift.

An HBase system is designed to scale linearly. It comprises a set of standard tables with rows and columns, much like a traditional database. Each table must have an element defined as a primary key, and all access attempts to HBase tables must use this primary key.

### **HBase Shell**

HBase is the Hadoop database, which provides random, real-time read/write access to very large data. See the references on HBase for more information.

#### **HBase Shell**

HBase contains a shell using which you can communicate with HBase. HBase uses the Hadoop File System to store its data. It will have a master server and region servers. The data storage will be in the form of regions (tables). These regions will be split up and stored in region servers.

The master server manages these region servers, and all these tasks take place on HDFS. Given below are some of the commands supported by HBase Shell.

## General Commands

status - Provides the status of HBase, for example, the number of servers.  
version - Provides the version of HBase being used.

table\_help - Provides help for table-reference commands.

Who am i - Provides information about the user.

## Data Definition Language

These are the commands that operate on the tables in HBase. create - Creates a table.

list - Lists all the tables in HBase.

disable - Disables a table.

is\_disabled - Verifies whether a table is disabled.

enable - Enables a table.

is\_enabled - Verifies whether a table is enabled.

describe - Provides the description of a table.

alter - Alters a table.

exists - Verifies whether a table exists.

drop - Drops a table from HBase.

drop\_all - Drops the tables matching the 'regex' given in the command.

Java Admin API - Prior to all the above commands, Java provides an Admin API to achieve DDL functionalities through programming. Under org.apache.hadoop.hbase.client package, HBaseAdmin and HTableDescriptor are the two important classes in this package that provide DDL functionalities. **Data Manipulation Language**

put - Puts a cell value at a specified column in a specified row in a particular table.

get - Fetches the contents of row or a cell.

delete - Deletes a cell value in a table.

deleteall - Deletes all the cells in a given row.

scan - Scans and returns the table data.

count - Counts and returns the number of rows in a table.

truncate - Disables, drops, and recreates a specified table.

Java client API - Prior to all the above commands, Java provides a client API to achieve DML functionalities, CRUD (Create Retrieve Update Delete) operations and more through

programming, under org.apache.hadoop.hbase.client package. HTable Put and Get are the important classes in this package.

## Starting HBase Shell

To access the HBase shell, you have to navigate to the HBase home folder.

```
cd /usr/localhost/ cd
```

Hbase

You can start the HBase interactive shell using “hbase shell” command as shown below.

```
./bin/hbase shell
```

If you have successfully installed HBase in your system, then it gives you the HBase shell prompt as shown below.

HBase Shell; enter 'help<RETURN>' for list of supported commands.

Type "exit<RETURN>" to leave the HBase Shell

Version 0.94.23,

rf42302b28aceaab773b15f234aa8718fff7eea3c, Wed Aug 27

00:54:09 UTC 2014

```
hbase(main):001:0>
```

To exit the interactive shell command at any moment, type exit or use <ctrl+c>. Check the shell functioning before proceeding further. Use the list command for this purpose. List is a command used to get the list of all the tables in HBase. First of all, verify the installation and the configuration of HBase in your system using this command as shown below.

```
hbase(main):001:0> list
```

When you type this command, it gives you the following output.

```
hbase(main):001:0> list
```

TABLE

## Output:

```
Applications Places System cloudera@quickstart:~
Browse and run installed applications
File Edit View Search Terminal Help
0 row(s) in 1.3370 seconds

=> Hbase::Table - student
hbase(main):006:0> list
TABLE
student
1 row(s) in 0.0110 seconds

=> ["student"]
hbase(main):007:0> disable

ERROR: wrong number of arguments (0 for 1)

Start disable of named table:
  hbase> disable 't1'
  hbase> disable 'ns1:t1'

hbase(main):008:0> disable 'student'
0 row(s) in 2.4590 seconds

hbase(main):009:0> is_disabled 'student'
true
0 row(s) in 0.0270 seconds

hbase(main):010:0> enable 'student'
0 row(s) in 1.2920 seconds

hbase(main):011:0> is_enabled 'student'
true
0 row(s) in 0.0210 seconds

hbase(main):012:0> describe 'student'

[Cloudera Live : Welco... cloudera@quickstart:~ [Installing HBase throu...
Applications Places System
Access documents, folders and network places cloudera@quickstart:~
File Edit View Search Terminal Help
ce
t to table 't1', the corresponding command would be:

  hbase> t.delete 'r1', 'c1', ts1
  hbase> t.delete 'r1', 'c1', ts1, {VISIBILITY=>'PRIVATE|SECRET'}

hbase(main):030:0> delete 'student','3','rollno'
0 row(s) in 0.0340 seconds

hbase(main):031:0> deleteall 'student','3'
0 row(s) in 0.0080 seconds

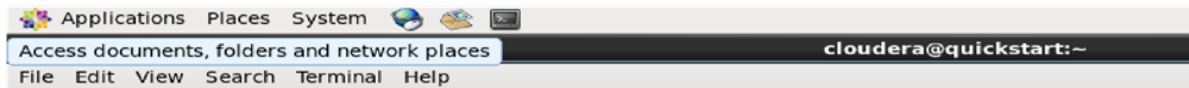
hbase(main):032:0> scan 'student'
ROW COLUMN+CELL
1 column=name:, timestamp=1660730755646, value=Shraddha
1 column=rollno:, timestamp=1660730831374, value=115
2 column=name:, timestamp=1660730859304, value=Kajal
2 column=rollno:, timestamp=1660730889867, value=106
4 column=rollno:, timestamp=1660731028866, value=108
3 row(s) in 0.0420 seconds

hbase(main):033:0> count 'student'
3 row(s) in 0.0240 seconds

=> 3
hbase(main):034:0> truncate 'student'
Truncating 'student' table (it may take a while):
- Disabling table...
- Truncating table...
0 row(s) in 3.4020 seconds

hbase(main):035:0> █

[Cloudera Live : Welco... cloudera@quickstart:~ [Installing HBase throu...
```



The same command can also be run on a table reference. Suppose you had a reference to table 't1', the corresponding command would be:

```
hbase> t.delete 'r1', 'c1', ts1
hbase> t.delete 'r1', 'c1', ts1, {VISIBILITY=>'PRIVATE|SECRET'}
```

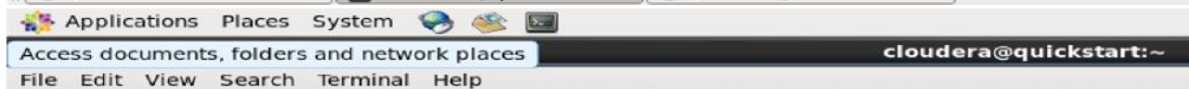
```
hbase(main):030:0> delete 'student','3','rollno'
0 row(s) in 0.0340 seconds
```

```
hbase(main):031:0> deleteall 'student','3'
0 row(s) in 0.0080 seconds
```

```
hbase(main):032:0> scan 'student'
ROW COLUMN+CELL
1 column=name:, timestamp=1660730755646, value=Shraddha
1 column=rollno:, timestamp=1660730831374, value=115
2 column=name:, timestamp=1660730859304, value=Kajal
2 column=rollno:, timestamp=1660730889867, value=106
4 column=rollno:, timestamp=1660731028866, value=108
3 row(s) in 0.0420 seconds
```

```
hbase(main):033:0> count 'student'
3 row(s) in 0.0240 seconds
```

```
=> 3
hbase(main):034:0> truncate 'student'
Truncating 'student' table (it may take a while):
- Disabling table...
- Truncating table...
0 row(s) in 3.4020 seconds
```



```
rollno: timestamp=1660730831374, value=115
2 row(s) in 0.0220 seconds
```

```
hbase(main):026:0> get 'student','2'
COLUMN CELL
name: timestamp=1660730859304, value=Kajal
rollno: timestamp=1660730889867, value=106
2 row(s) in 0.0130 seconds
```

```
hbase(main):027:0> put 'student','3','name','Neha'
0 row(s) in 0.0250 seconds
```

```
hbase(main):028:0> put 'student','4','rollno','108'
0 row(s) in 0.0120 seconds
```

```
hbase(main):029:0> delete 'student','3'
```

ERROR: wrong number of arguments (2 for 3)

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 'ns1:t1', 'r1', 'c1', ts1
hbase> delete 't1', 'r1', 'c1', ts1
hbase> delete 't1', 'r1', 'c1', ts1, {VISIBILITY=>'PRIVATE|SECRET'}
```

The same command can also be run on a table reference. Suppose you had a reference to table 't1', the corresponding command would be:



```
Applications Places System cloudera@quickstart:~
Change desktop appearance and behavior, get help, or log out
File Edit View Search Terminal Help
0 row(s) in 0.0370 seconds

hbase(main):017:0> drop 'student'

ERROR: Table student is enabled. Disable it first.

Drop the named table. Table must first be disabled:
hbase> drop 't1'
hbase> drop 'ns1:t1'

hbase(main):018:0> disable 'student'
0 row(s) in 2.2680 seconds

hbase(main):019:0> drop 'student'
0 row(s) in 1.2950 seconds

hbase(main):020:0> create 'student','name','rollno'
0 row(s) in 1.2510 seconds

=> Hbase::Table - student
hbase(main):021:0> put 'student','1','name','Shraddha'
0 row(s) in 0.1110 seconds

hbase(main):022:0> put 'student','1','rollno','115'
0 row(s) in 0.0130 seconds

hbase(main):023:0> put 'student','2','name','Kajal'
0 row(s) in 0.0190 seconds

hbase(main):024:0> put 'student','2','rollno','106'
0 row(s) in 0.0080 seconds
```

```
[Cloudera Live : Welco... cloudera@quickstart:~ [Installing HBase throu...
Applications Places System cloudera@quickstart:~
Access documents, folders and network places
File Edit View Search Terminal Help
hbase(main):012:0> describe 'student'
Table student is ENABLED
student
COLUMN FAMILIES DESCRIPTION
{NAME => 'name', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATIO
N SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL
=> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY =>
'false', BLOCKCACHE => 'true'}
{NAME => 'roll', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATIO
N SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL
=> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY =>
'false', BLOCKCACHE => 'true'}
2 row(s) in 0.0480 seconds

hbase(main):013:0> alter 'student', roll=>115,name=>'Shraddha'
NameError: undefined local variable or method `roll' for #<Object:0x1bd4269b>

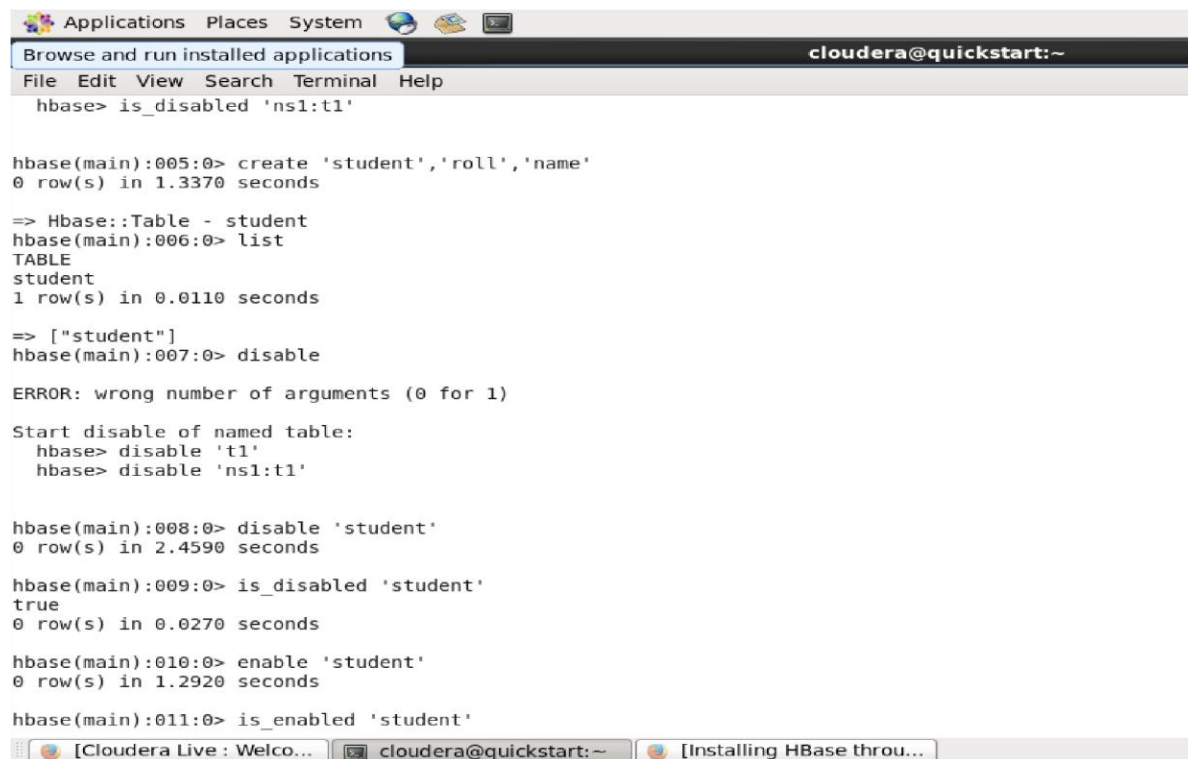
hbase(main):014:0> alter 'student', ROLL=>115,NAME=>'Shraddha'
NameError: uninitialized constant ROLL

hbase(main):015:0> alter 'student', NAME=>'roll',VERSIONS=>5
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 2.0120 seconds

hbase(main):016:0> exists 'student'
Table student does exist
0 row(s) in 0.0370 seconds

hbase(main):017:0> drop 'student'

ERROR: Table student is enabled. Disable it first.
```



```
Applications Places System
Browse and run installed applications cloudera@quickstart:~
File Edit View Search Terminal Help
hbase> is_disabled 'ns1:t1'

hbase(main):005:0> create 'student','roll','name'
0 row(s) in 1.3370 seconds

=> Hbase::Table - student
hbase(main):006:0> list
TABLE
student
1 row(s) in 0.0110 seconds

=> ["student"]
hbase(main):007:0> disable

ERROR: wrong number of arguments (0 for 1)

Start disable of named table:
  hbase> disable 't1'
  hbase> disable 'ns1:t1'

hbase(main):008:0> disable 'student'
0 row(s) in 2.4590 seconds

hbase(main):009:0> is_disabled 'student'
true
0 row(s) in 0.0270 seconds

hbase(main):010:0> enable 'student'
0 row(s) in 1.2920 seconds

hbase(main):011:0> is_enabled 'student'
```

## Conclusion:

Hence, we have successfully installed and configured HBase to execute NoSQL commands.