cs523 - BDT

Lab 8 – Apache HBase

- Submit your *own work* on time. No credit will be given if the lab is submitted after the due date.
- Note that the completed lab should be submitted in .zip format only.

This document is divided into two parts.

1. Practice Lab

a. With HBase Shell commands

Just try to run through all the steps and see if they work properly for you.
No need to submit this part.

If you are getting error in the shell and even the `list` command is not working, then probably your HBase Master and RegionServers are down.

Use the following command to check whether they are up and running.

```
[cloudera@quickstart ~]$ service --status-all;
```


Use the following commands to start them again if they were not running.

```
[cloudera@quickstart ~]$ sudo service hbase-master start;
```

```
[cloudera@quickstart ~]$ sudo service hbase-regionserver start;
```

2. Homework Lab

a. Using HBase Java API

You need to submit a .zip file wherein I should be able to find all the instructions and .java files.

Paste screenshots wherever applicable.

HBase Shell Commands Practice Lab

1. Connect to HBase.

   Connect to your running instance of HBase using the `hbase shell` command. The HBase Shell prompt ends with a `>` character.

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

2. Display HBase Shell Help Text.

   Type `help` and press Enter, to display some basic usage information for HBase Shell, as well as several example commands. Notice that table names, rows, columns all must be enclosed in quote characters.

3. Create a table.

   Use the `create` command to create a new table. You must specify the table name and the ColumnFamily name.

   ```
   hbase(main):001:0> create 'test', 'cf'
   0 row(s) in 0.4170 seconds

   => Hbase::Table - test
   ```

4. List Information About your Table

   Use the `list` command to see the tables.

   ```
   hbase(main):002:0> list 'test'
   TABLE
   test
   1 row(s) in 0.0180 seconds

   => ["test"]
   ```

5. Put data into your table.

   To put data into your table, use the `put` command.

   ```
   hbase(main):003:0> put 'test', 'row1', 'cf:a', 'value1'
   0 row(s) in 0.0850 seconds

   hbase(main):004:0> put 'test', 'row2', 'cf:b', 'value2'
   0 row(s) in 0.0110 seconds

   hbase(main):005:0> put 'test', 'row3', 'cf:c', 'value3'
   0 row(s) in 0.0100 seconds

   hbase(main):006:0> put 'test', 'row3', 'cf:a', 'value4'
   0 row(s) in 0.0100 seconds
   ```

   Here, we insert four values, one at a time. The first insert is at `row1`, column `cf:a`, with a value of `value1`. Columns in HBase are comprised of a column family prefix, `cf` in this example, followed by a colon and then a column qualifier suffix, `a` in this case.

6. Scan the table for all data at once.

   One of the ways to get data from HBase is to scan. Use the `scan` command to scan the table for data. You can limit your scan, but for now, all data is fetched.

```
hbase(main):006:0> scan 'test'
ROW                COLUMN+CELL
 row1              column=cf:a, timestamp=1421762485768, value=value1
 row2              column=cf:b, timestamp=1421762491785, value=value2
 row3              column=cf:a, timestamp=1431465798166, value=value4
 row3              column=cf:c, timestamp=1421762496210, value=value3
3 row(s) in 0.0230 seconds
```

7. Get a single row of data.

   To get a single row of data at a time, use the `get` command.

```
hbase(main):007:0> get 'test', 'row1'
COLUMN                          CELL
 cf:a                           timestamp=1421762485768, value=value1
1 row(s) in 0.0350 seconds
```

8. Disable a table.

   If you want to delete a table or change its settings, as well as in some other situations, you need to disable the table first, using the `disable` command. You can re-enable it using the `enable` command.

```
hbase(main):008:0> disable 'test'
0 row(s) in 1.1820 seconds

hbase(main):009:0> enable 'test'
0 row(s) in 0.1770 seconds
```

   Disable the table again if you tested the `enable` command above:

```
hbase(main):010:0> disable 'test'
0 row(s) in 1.1820 seconds
```

9. Drop the table.

   To drop (delete) a table, use the `drop` command.

```
hbase(main):011:0> drop 'test'
0 row(s) in 0.1370 seconds
```

10. Exit the HBase Shell.

    To exit the HBase Shell and disconnect from your cluster, use the `quit` command. HBase will still continue to run in the background.

HBase Java API HW Lab

- Download "`MyFirstHbaseTable.java`" program and run it successfully in eclipse.

  Note: Along with the Hadoop jars, now you'll also need HBase jars from the following locations. Add them to the build path of the project.

  - **`File System/usr/lib/hbase`**
    and
    `File System/usr/lib/hbase/lib`

- Check the created table using HBase browser.
- If you get some error, then probably your Master and RegionServers are down. Check the beginning of this document as to how to start them again.

Now solve the following 3 questions. For the first 2 questions submit the .java files with a screen shot of the table when viewed from HBase data browser and when "scanned" through HBase shell.

1. [5] **Modify "`MyFirstHbaseTable.java`" program to create the following table structure and data in HBase.**
   Note that Row Key is the given EmpId.

Row Key	Personal Data		Professional Data	
	Name	**City**	**Designation**	**salary**
1	Peter	Fairfield	Manager	110,000
2	Nancy	Burlington	Sr. Engineer	120,000
3	Bob	Weymouth	Jr. Engineer	90,000

2. [3] **Now *programmatically* promote Bob to Sr. Engineer position and increase his salary by 3%.**
   You can assume that you already know Bob's EmpId (that's rowkey).

3. [2] **Figure out at least two different ways to find the number of rows in an HBase table.** Write commands and paste screenshots to show that you tried to do this.