**MSCD664**

**Lab 1 – HBase NoSQL Database**

**Name: \_\_\_\_\_\_\_\_Rohan Purohit\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_Sept 4 2016\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

The purpose of this lab is to become familiar with the Hortonworks Sandbox HBase environment. Please respond to the instructions/questions highlighted in yellow.

***It is critical that you use Hortonworks 2.1. Later versions, such as version Hortonworks Sandbox 2.2, will not work on your PC and you will be frustrated.***

You will perform the following tasks in the lab:

* Starting your HBase database within the Hortonworks environment.
* Create a few Hbase column families (table) and then add data.
* Query data from the HBase database.
* Disable and Drop the HBase families (tables).
* Shutdown Hortonworks gracefully and exit the virtual box.

**Read Chapter 4 in the Seven Weeks text. Also, ensure you have reviewed sections from the online manual on the open source Apache HBase website:**

**http://hbase.apache.org/book.html#datamodel**

These items should be reviewed and are important for an understanding what you are doing in this lab.

[Data Model](http://hbase.apache.org/book.html#datamodel) - [19. Conceptual View](http://hbase.apache.org/book.html#conceptual.view)

[20. Physical View](http://hbase.apache.org/book.html#physical.view)

[21. Namespace](http://hbase.apache.org/book.html#_namespace)

[22. Table](http://hbase.apache.org/book.html#_table)

[23. Row](http://hbase.apache.org/book.html#_row)

[24. Column Family](http://hbase.apache.org/book.html#columnfamily)

[25. Cells](http://hbase.apache.org/book.html#_cells)

[26. Data Model Operations](http://hbase.apache.org/book.html#_data_model_operations)

[27. Versions](http://hbase.apache.org/book.html#versions)

[28. Sort Order](http://hbase.apache.org/book.html#dm.sort)

[29. Column Metadata](http://hbase.apache.org/book.html#dm.column.metadata)

[30. Joins](http://hbase.apache.org/book.html#_joins)

[31. ACID](http://hbase.apache.org/book.html#_acid)

To get an “A” grade on this lab:

* Answer all questions.
* Provide output of steps that require you to perform a task. The output can be a cut and paste from the screen to a word document or screen shots; the windows snipping tool works well. I need to see something to verify that you ran the labs. I’m expecting you to turn in a word document that contains each question and also the corresponding answers to questions asked and output from HBase statements. It is okay to paste your results into this word document and then submit the completed document.
* Tasks that require an output from you will be indicated clearly with yellow.

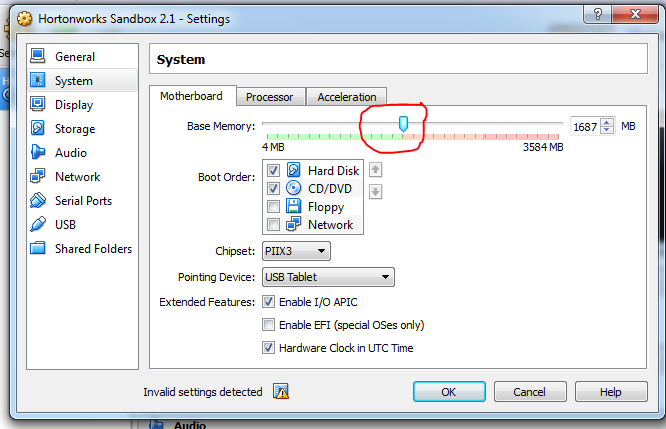
**-----------------------------------------------------------------------------------**

**Part 1: Understanding Your HBase Environment**

**-----------------------------------------------------------------------------------**

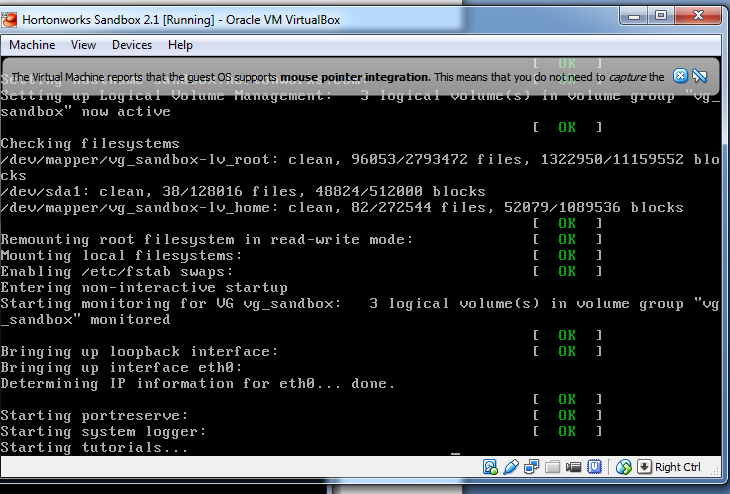
Hortonworks

1. Start the virtual machine and then start Hortonworks. If you have a small amount of memory on your PC (such as 4 Gig of RAM), consider adjusting the Hortonworks System settings as shown below. If you use up all of the memory on your PC, it will lock up and you’ll have to reboot your PC. I have been using roughly 2 Gig of memory and Hortonworks is working fine. If you get an error about your BIOS, contact your instructor for more information.

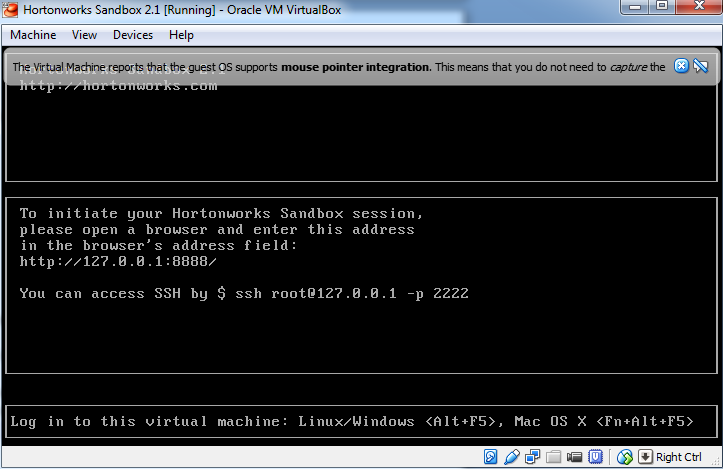


When you start Hortonworks, it will start very slowly.

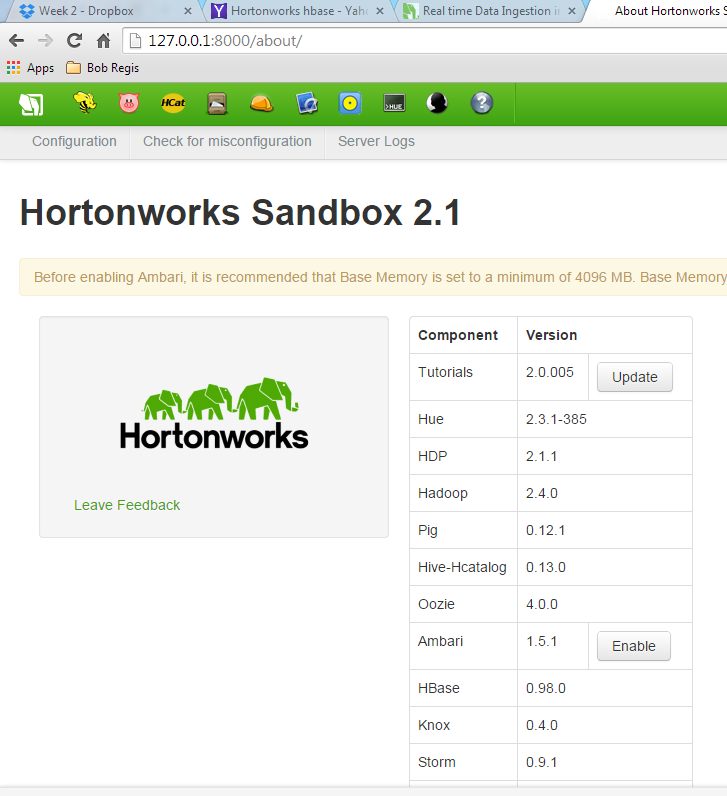
As the startup progresses, your window will contain status items.



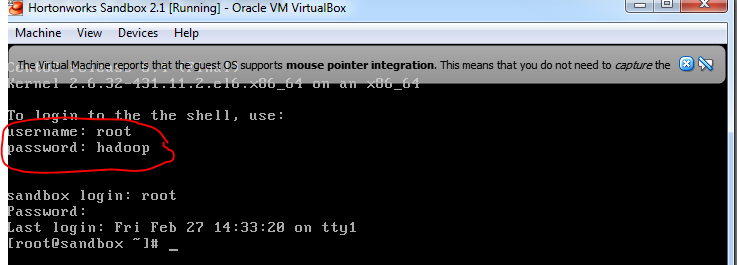
1. When Hortonworks Sandbox is ready to use, your window will appear as shown below.



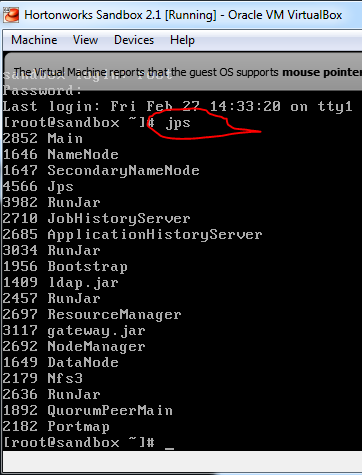
1. You will connect with your web browser (Chrome is my preferred browser) using: 127.0.0.1:8000. Also, open another webpage using 127.0.0.1:8888. We won’t be using these webpages in this particular lab, however it is important to know about the web interface and were to find the HBase tutorials.



1. Now login into the Hadoop machine using the windows option by switching to the VM black window for Hortonworks Sandbox 2.1 and then press the **Alt+F5** keys if you are using windows. You will then login by using the **root ID and hadoop password** that is provided (shown below).



1. Run the **jps** command to see what processes are running in your Hadoop environment.



1. If **HMaster** and **HRegionServer** are missing in the ‘jps’ command output list, as seen above, we need to restart the HBase services. To start the HBase master login as hbase user

**su hbase**

-- start the hbase master

**/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf start master**

***Note: you can use the up arrow key to display the prior command.***

-- start the hbase region serve//

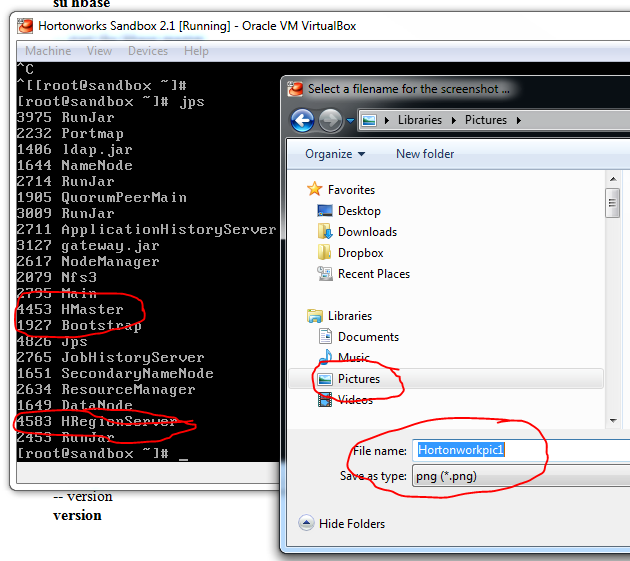
**/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf start regionserver**

***Note: if you take an extended break, the master and regionserver will shutdown. Thus, you will have to restart them to continue with the lab. You will notice that they have shutdown when the get and put commands do not work anymore and you get format errors. Simply exit the HBase shell using exit, then restart the two servers. Then, start the hbase shell again and continue with your labs.***

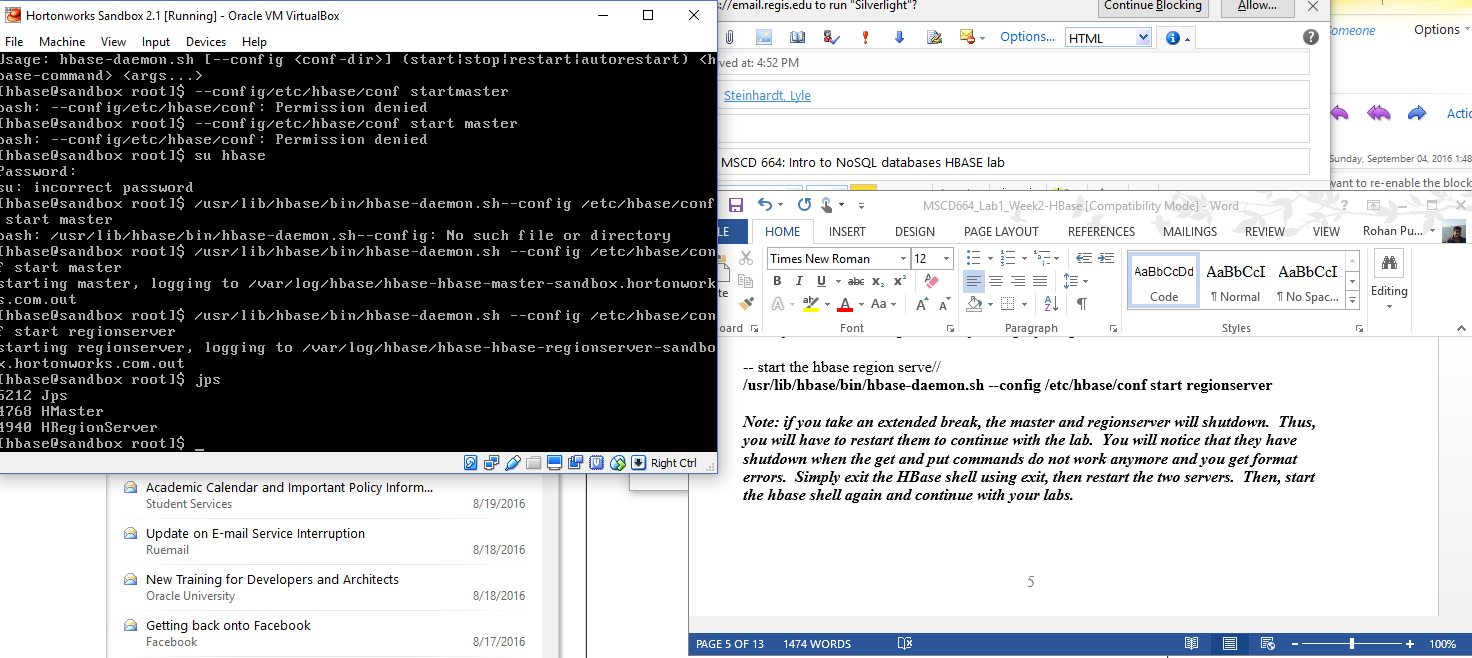
**--** Run the jps command again to see if the master and region servers are running.

***To provide a screen shot, in the Horton Window, click on Machine to see a drop down list. Then click on Take a Screen Shot within the drop down list. You will be prompted for a file name for the screen shot. In this example, I called my screen shot hortonworkpic1. Notice that I can see the HMaster and HRegionServer are running.***

***Or the EASY way is to use the snipping tool in windows grab the screen shot of the HBase interface and then paste it into this document as shown below.***

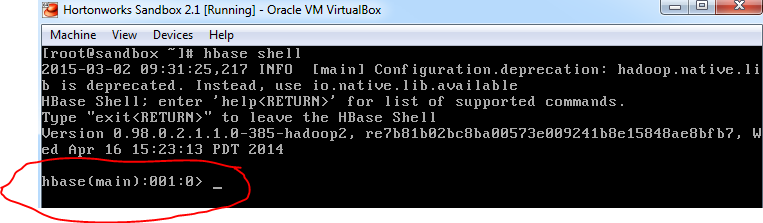


Provide a screenshot of your session to show that the **HMaster** and **HRegionServer***are running*:



Start the hbase shell – note it may take a few minutes for the shell to start depending upon your PC

**hbase shell**



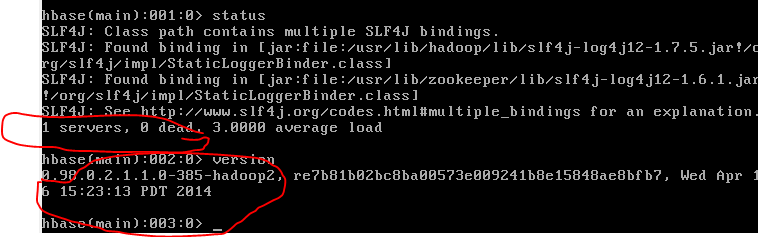
***Note: for future reference, to exit the hbase shell, type exit <return>***

-- check the status

**status**

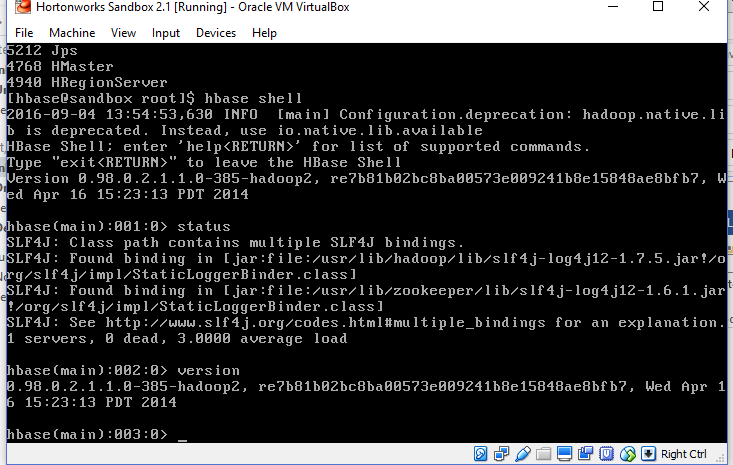
-- version

**version**



What version of HBase do you have and how many servers do you have?

I have 1 server and it is running 0.98.0.2.1.1.0-385-hadoop2 from Wednesday April 15 2014.

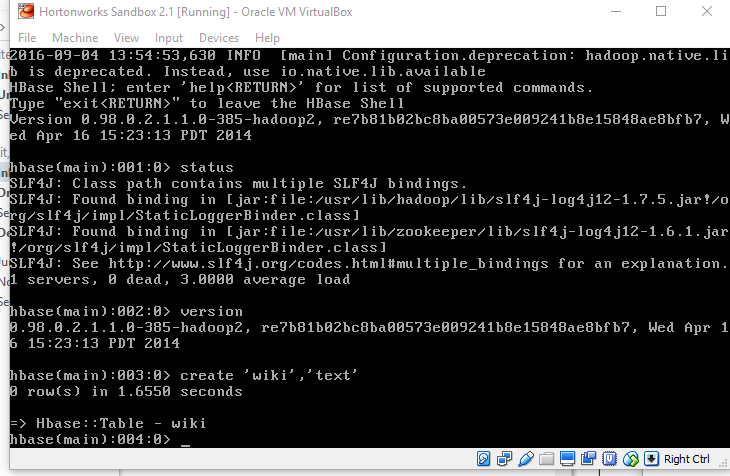


1. **Create Tables, Insert Data and Retrieve Data**

***-- note as you issue different commands, the prompt will increment. I didn’t show this in the examples below and I kept it as 000:0 since your values may be different.***

-- create a table from chapter 4 - page 98

**hbase(main):000:0>create 'wiki','text'**



-- insert data a row of data. Home is the key and Welcome is the value.

**hbase(main):000:0> put 'wiki', 'Home', 'text:', 'Welcome to the wiki!'**

-- retrieve data

**hbase(main):000:0> get 'wiki', 'Home', 'text:'**

-- insert more data into a new row of data, Home is the key and Welcome is the value

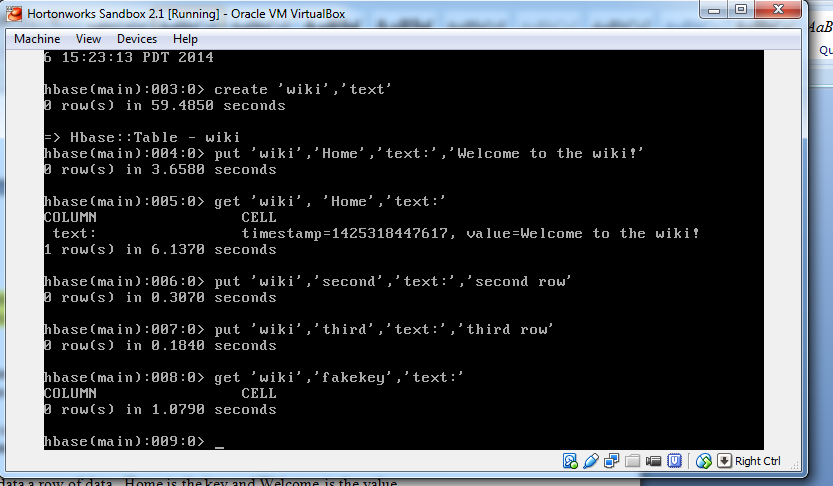
**hbase(main):000:0> put 'wiki', 'second', 'text:', 'second row'**

-- insert more data into a new row of data, Home is the key and Welcome is the value

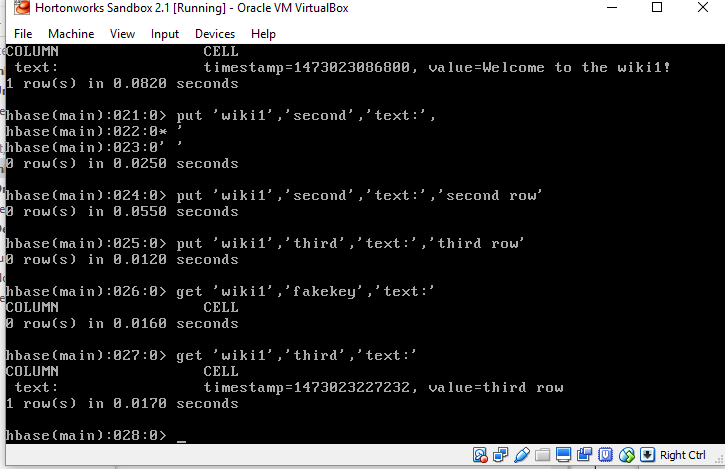
**hbase(main):000:0> put 'wiki', 'third', 'text:', 'third row'**

-- try retrieving data with a fake key

**hbase(main):000:0> get 'wiki', 'fakekey', 'text:'**



Provide a screenshot of your session to show that you created the table, inserted rows and retrieved data and answer the questions below:



**Questions:**

**How many rows were retrieved for the fakekey? Zero/None.**

**Besides the value for the row, what else is retrieved from HBase?**

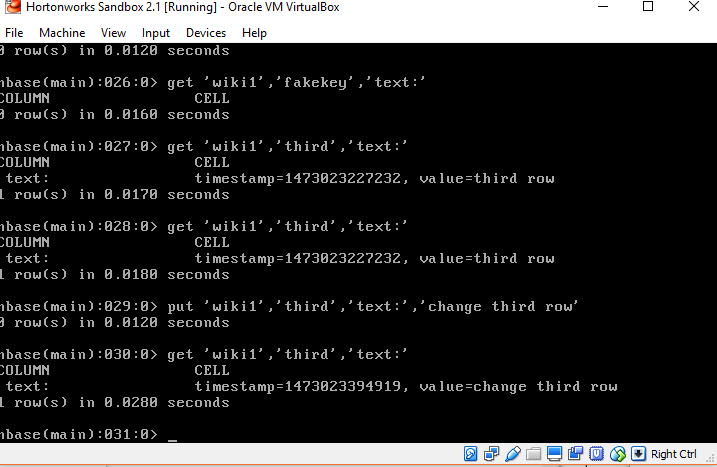
**We get a timestamp and a value for the row.**

Now retrieve the third row, put a new value in the row and retrieve it again as shown below.

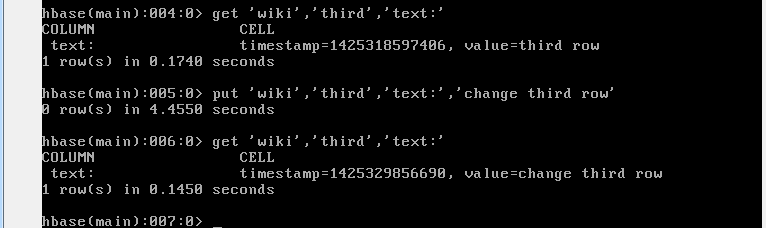
**get ‘wiki’,’third’,’text:’**

**put ‘wiki’,’third’,’text:’,’change third row’**

**get ‘wiki’,’third’,’text:’**



Why did the value change? Is the timestamp different? Yes the value changed because we put the new value via the put command. The timestamp also changed.



-- Issue the following command to create a ‘students’ table with two column families

**create 'students','account','address'**

-- add data to the table using multiple values for street, zipcode, state

**put 'students','student1','account:name','Alice'**

**put 'students','student1','address:street','123 Ballmer Av'**

**put 'students','student1','address:zipcode','12345'**

**put 'students','student1','address:state','CA'**

**put 'students','student2','account:name','Bob'**

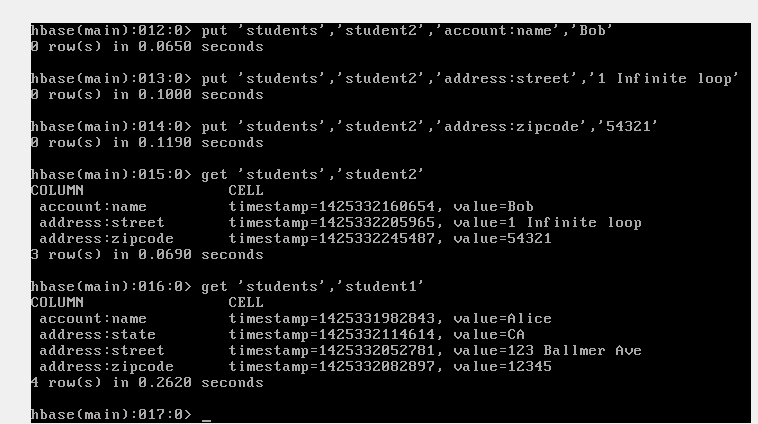
**put 'students','student2','address:street','1 Infinite Loop'**

**put 'students','student2','address:zipcode','12345'**

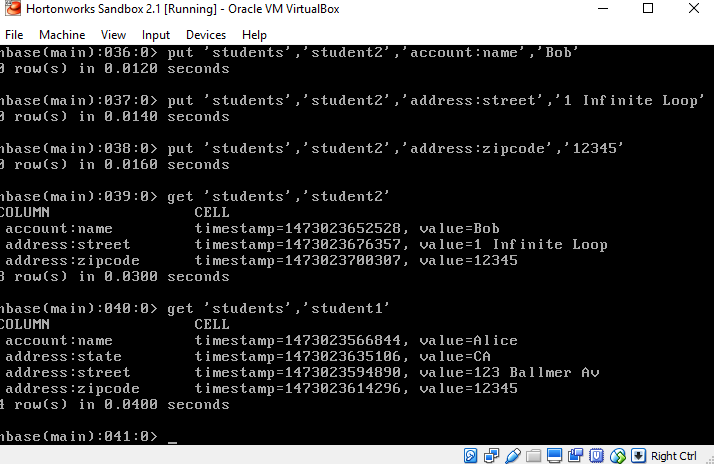
Retrieve the data the values for using the get command as shown below in the screen shot.

**get ‘students’,’student2’**

**get ‘students’,’student1’**



Provide a screenshot of your session to show that you created the table, inserted rows and retrieved data and answer the questions below:



When pundits describe NoSQL Databases as ‘sparsely populated’, what does that mean?

Sparsely populated means that we can have many attributes but not have much data values for those attributes.

How is sparse data shown in the examples above?

In this example we have the state attribute for student1 named Alice however we don’t have data for the state attribute for student2 called Bob.

Why did the one get statement return all of the values for one key?

Because the one region is identified by a collection of related rows in the database.

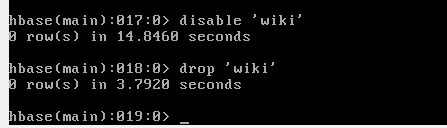
1. **Remove a table and shutdown HBase.**

Prior to dropping the tables, we need to disable them. Use the following commands to disable the wiki table that you created. We will keep the students table.

**disable ‘wiki’**

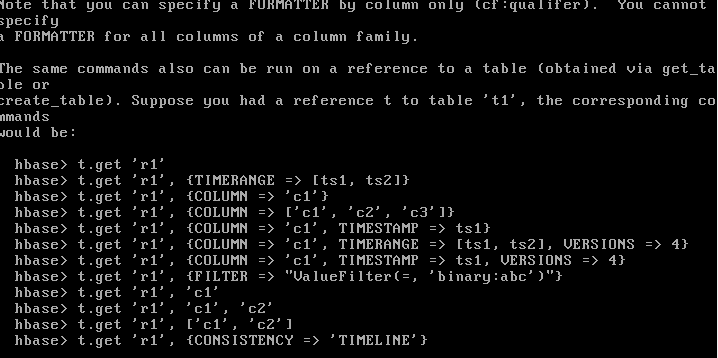
Then drop the ‘wiki’ table

**drop ‘wiki’**



Provide a screenshot of your session to show that you dropped the wiki table:

If you try to get data from the wiki table, you will now get an error.



Exit from the HBase Shell using exit.

**exit**



Notice that you are still logged in with HBase user ID. If not, then su hbase again.

**su hbase**

-- stop the hbase master

**/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf stop master**

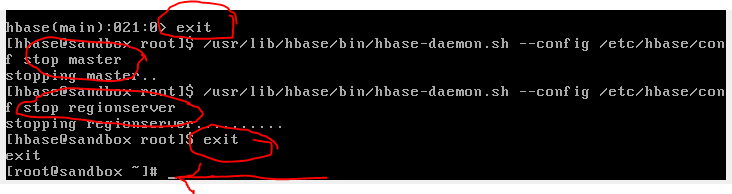
***Note: you can use the up arrow key to display the prior command.***

-- stop the hbase region server

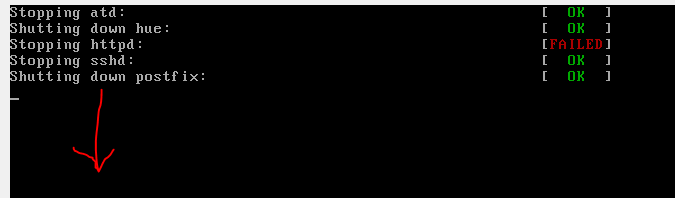
**/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf stop regionserver**

Exit from the hbase ID by type exit again.

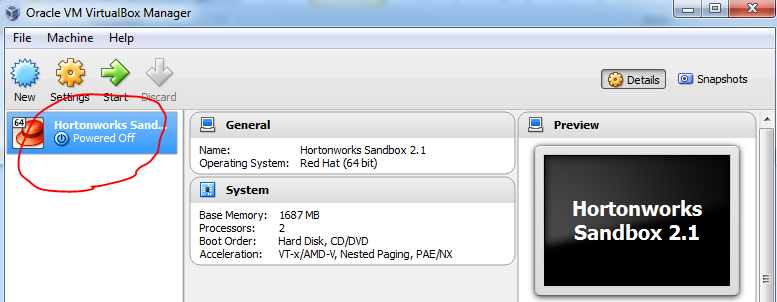
**exit**



1. Shutdown the Hortonworks Sandbox gracefully by clicking on **Machine** to get the dropdown list of options, then click on the **ACPI Shutdown** option. It will take several minutes to go through the shutdown process as shown below.



At this point, your Hortworks Sandbox 2.1 will be shutdown and you can close the Oracle Virtual Box.



We hope that you enjoyed this brief lab using HBase. If you run the same lab on a HBase cluster with many server machines, the results will be the same.

Upload your completed lab assignment to the Dropbox titled for this assignment by the date specified by your facilitator.