**Lab 2**

**Summary:**

To extract various data from the sensor tag and to store it in the HBase in column oriented approach effectively.

**Reading Data from Sensor:**

The following are the steps to be followed in order to create a file containing data recorded by sensor tag:

* Select the BLE Sensor Tag app and detect your sensor and select it.

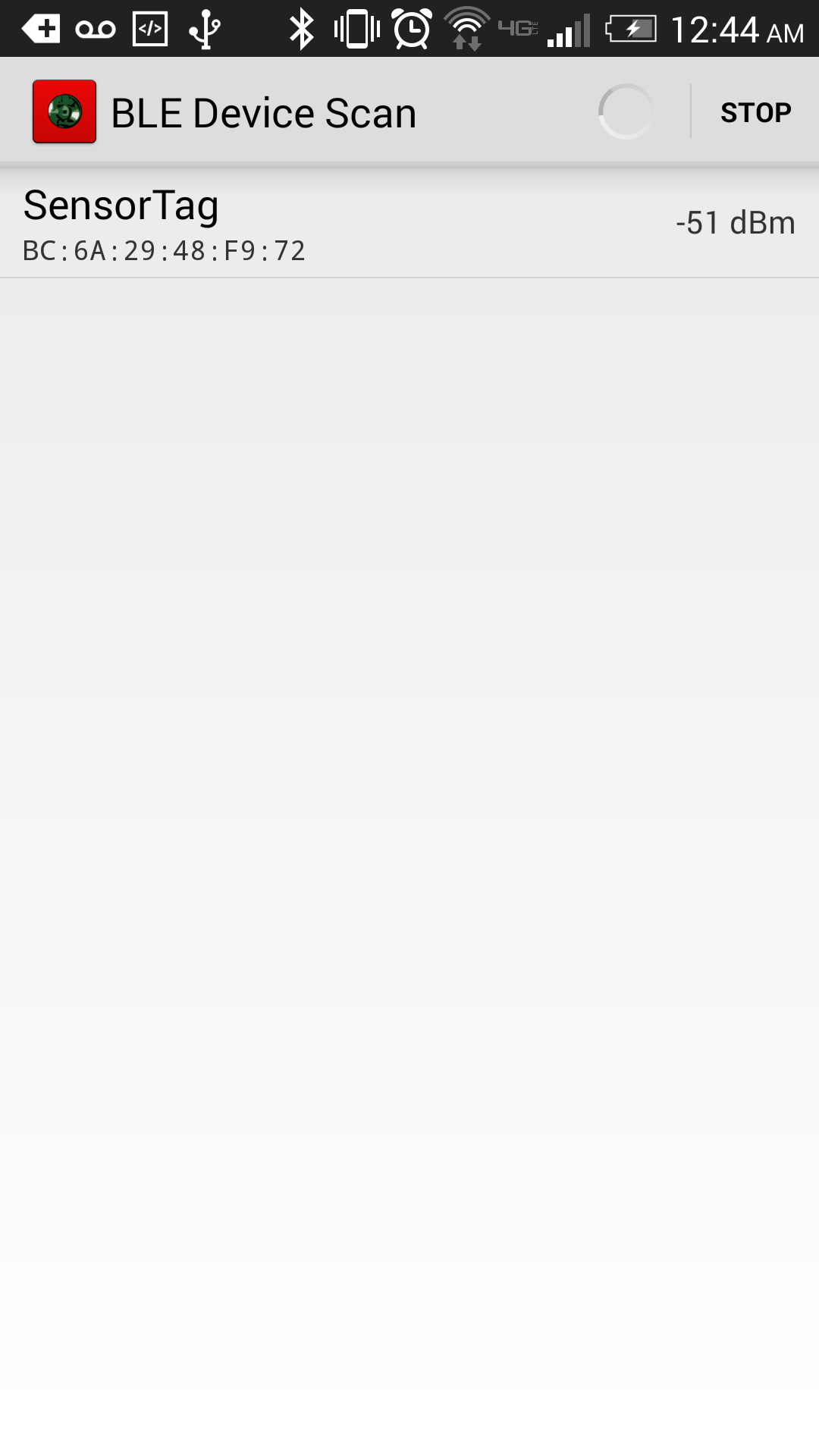


Figure Sensor detection

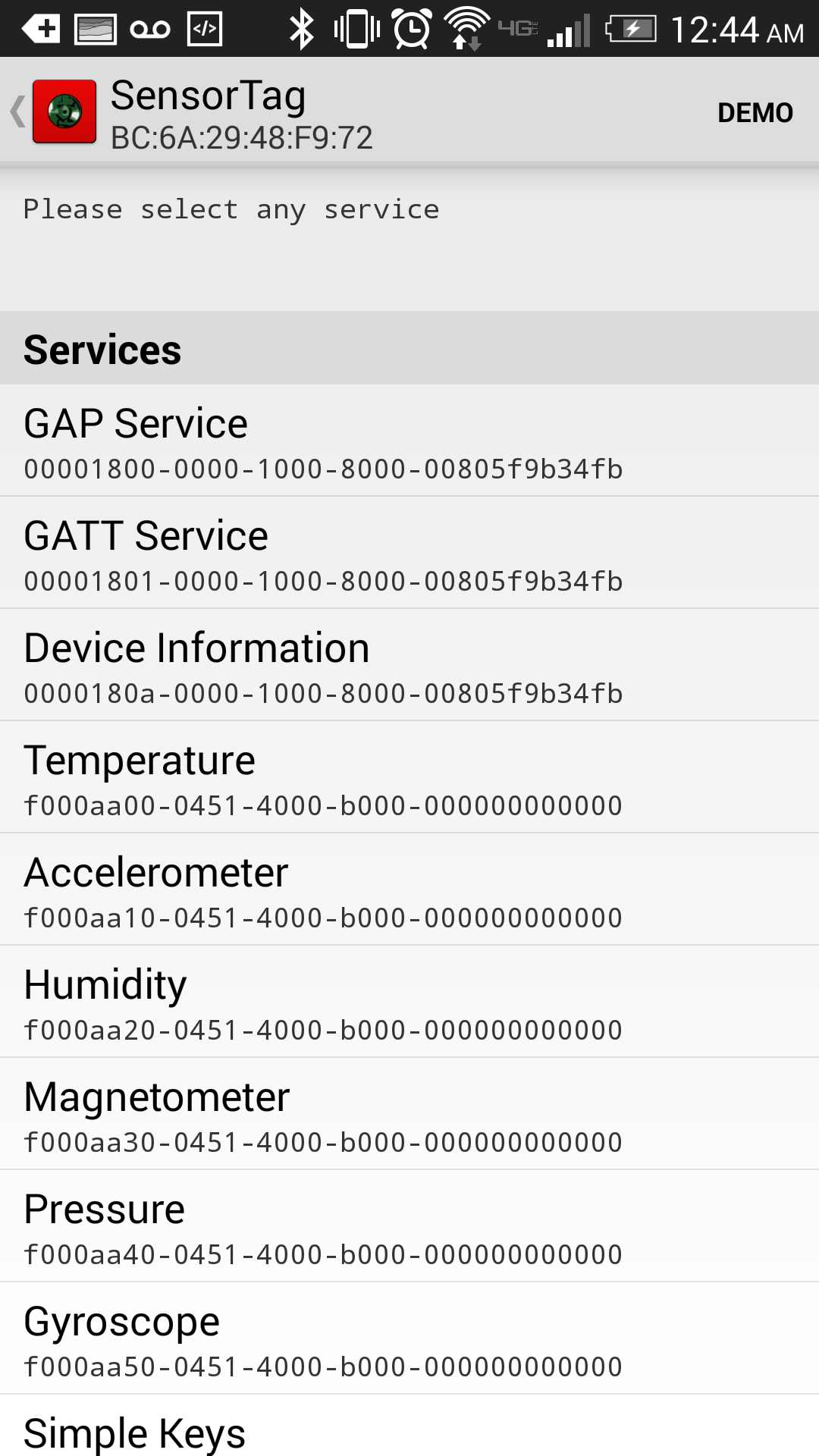


Figure Detecting the motions

* Now run the application in which we can collect the data from sensor tag on your android mobile device.
* Before running the application a couple of changes are required to record data
* You need to enable the humidity detection and pressure detection part of code in the application.
* The following changes are shown below:

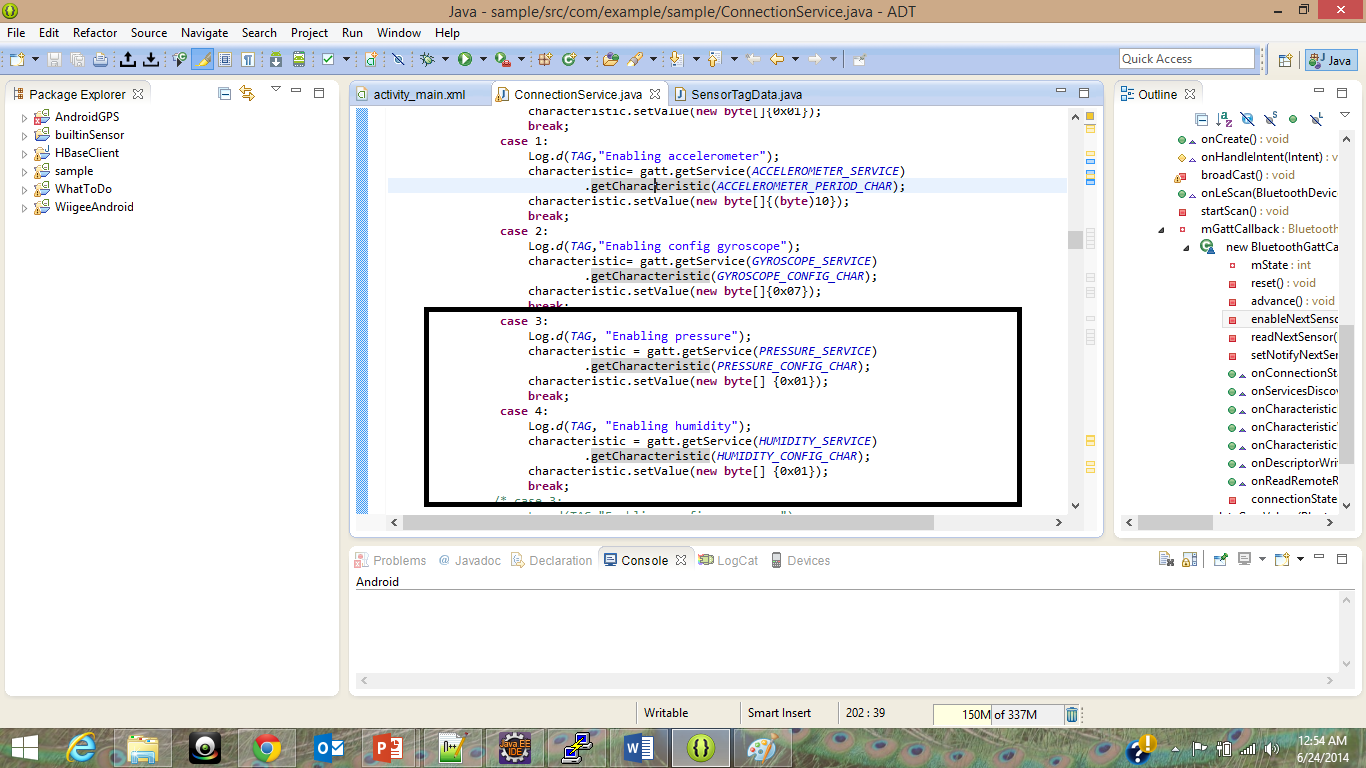


Figure enabling pressure and humidity detector

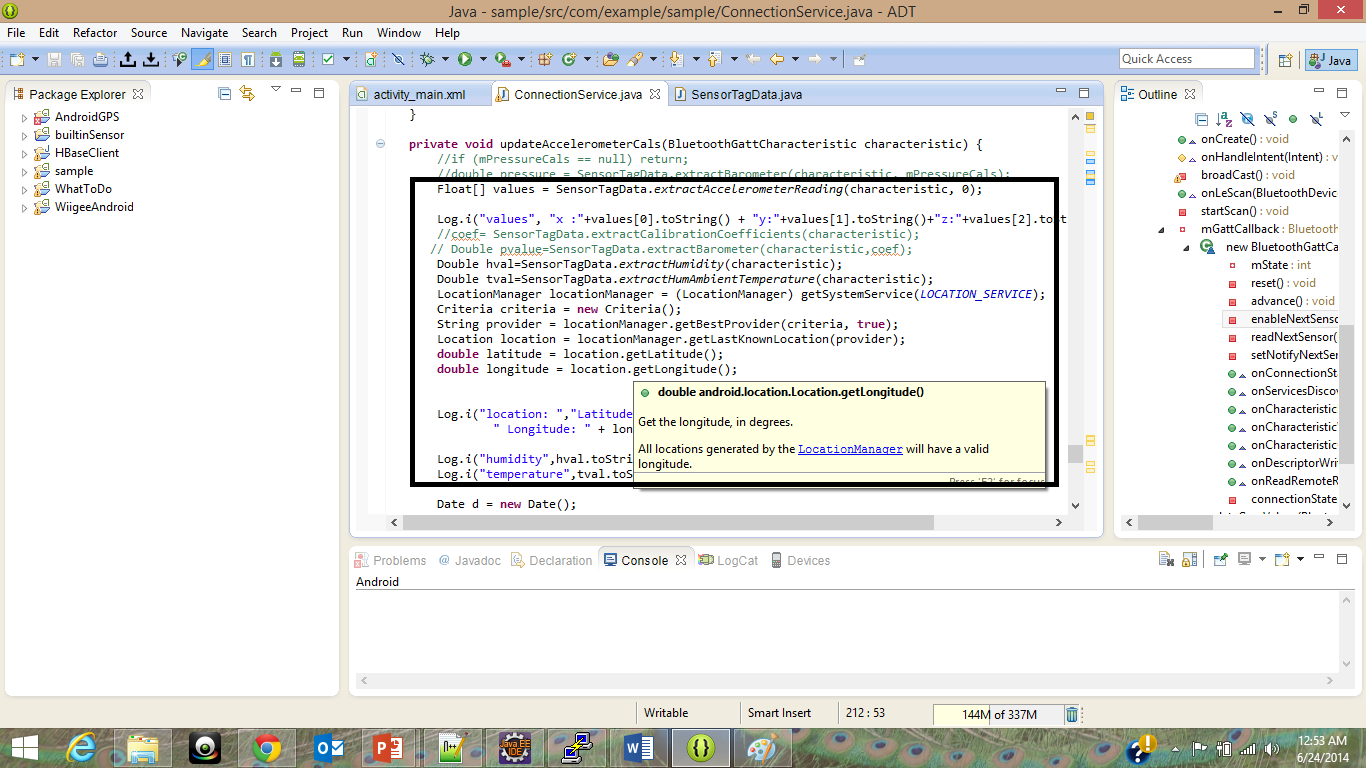


Figure evaluating the temperature and humidity

* Now you will have stored in the SD Card of your mobile device with longitude, latitude, date, accelerometer three dimensional coordinates, humidity and temperature.

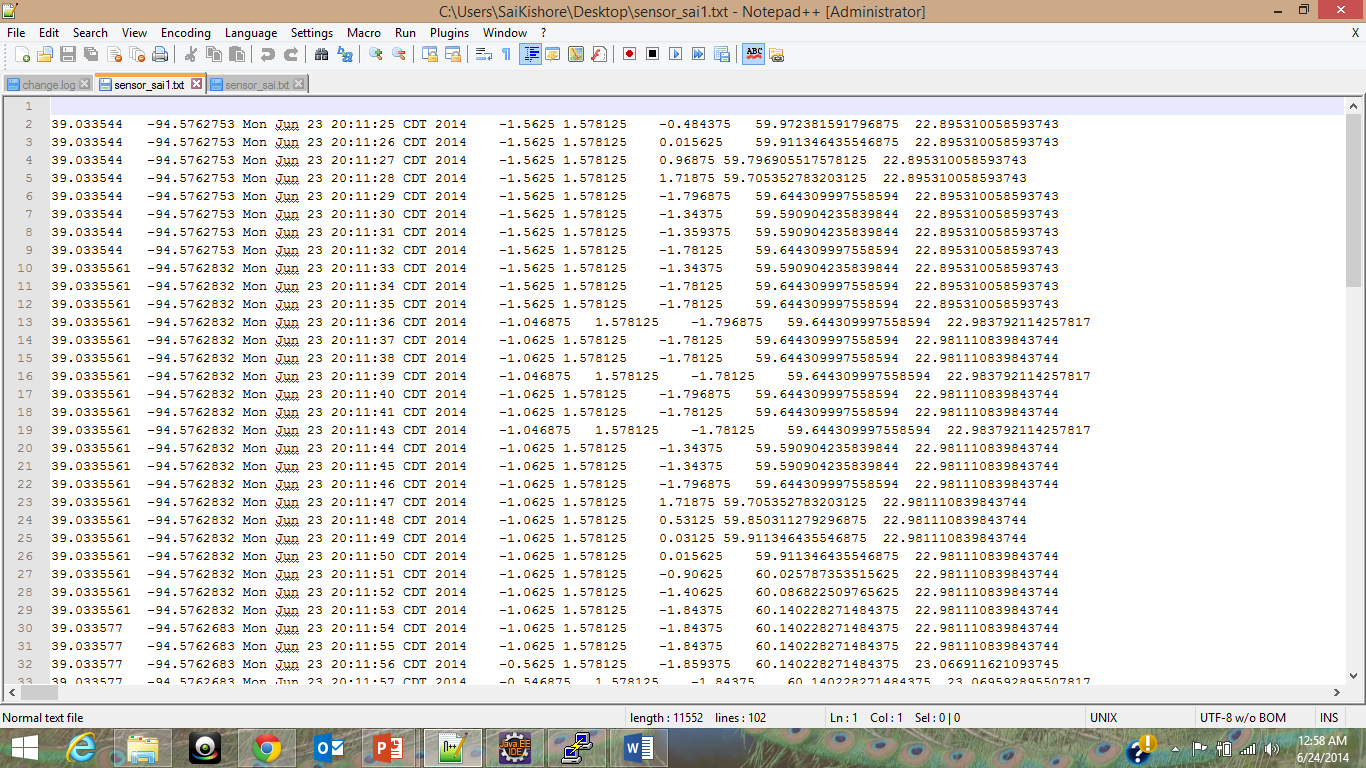


Figure extracted data

**Storing data in HBase:**

The above collected now will be stored in the HBase. The following are the steps to be followed for the operations to be done on HBase using these data.

**Creation of Table:**

* First add the domain of your HBase in the hosts file in your local machine in the following manner:
  + “ipaddress” localhost.localdomain localhost
* We have designed the table in the following manner:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Row key | Timestamp | Geographical coordinates | | date | Accelerometer readings | | | humidity | temperature |
|  |  | longitudes | latitudes |  | X | y | z |  |  |
|  |  |  |  |  |  |  |  |  |  |

* Now run the application with createtable() function uncommented and rest of functions being commented.
* The following is the output you can see after creating the table:

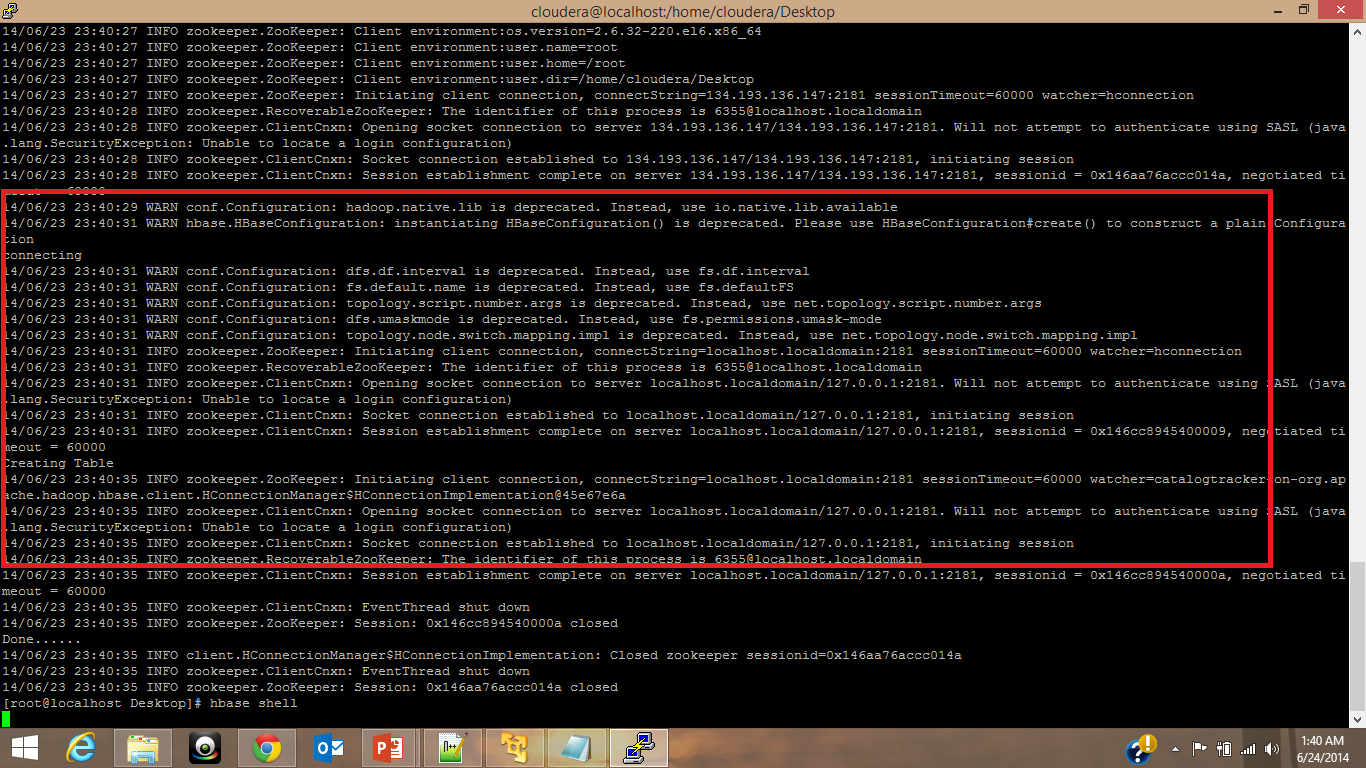


Figure 6Table Creation

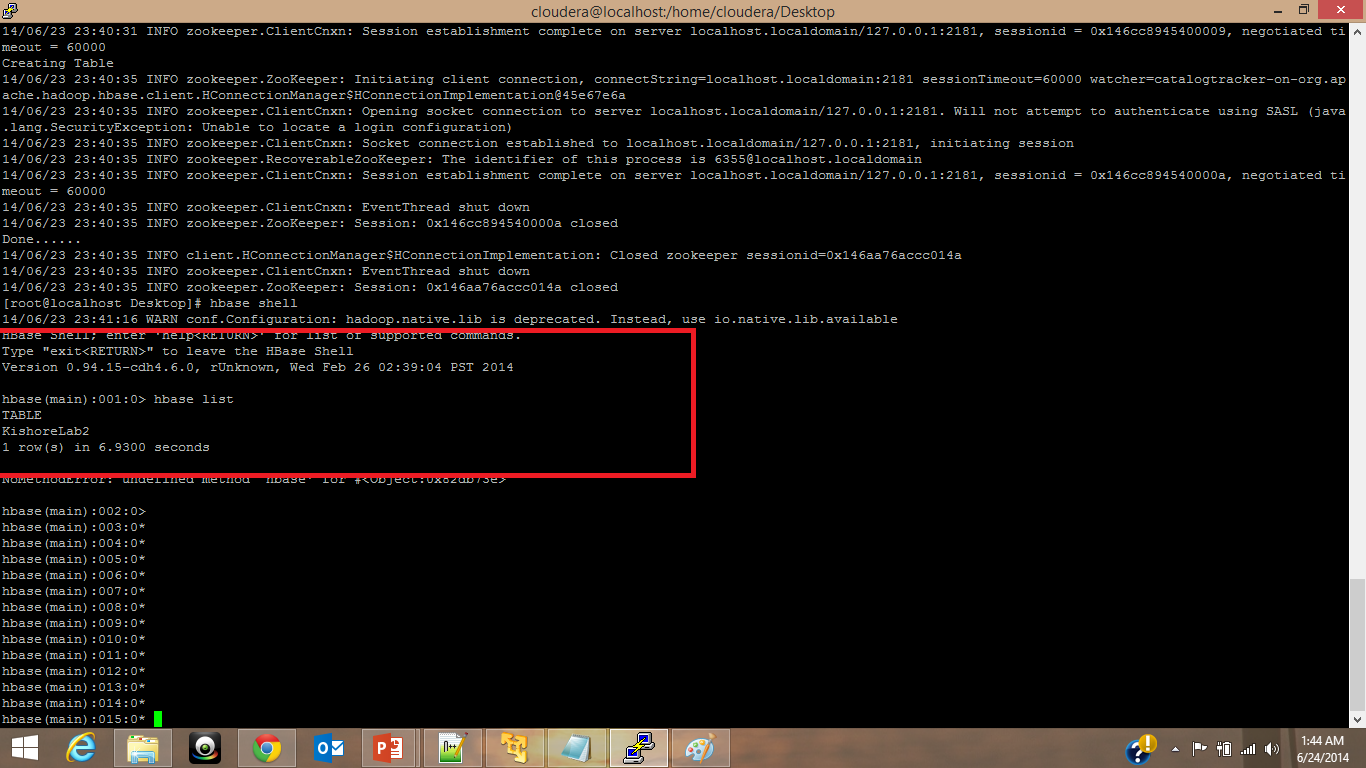


Figure 7List of tables created

**Insert operation:**

* Now enable the insertatbel() function and run the application, since we have the results in the tab separated form, the values will be inserted into the table as per the design which was mentioned above
* The following is the output of the insert operation :

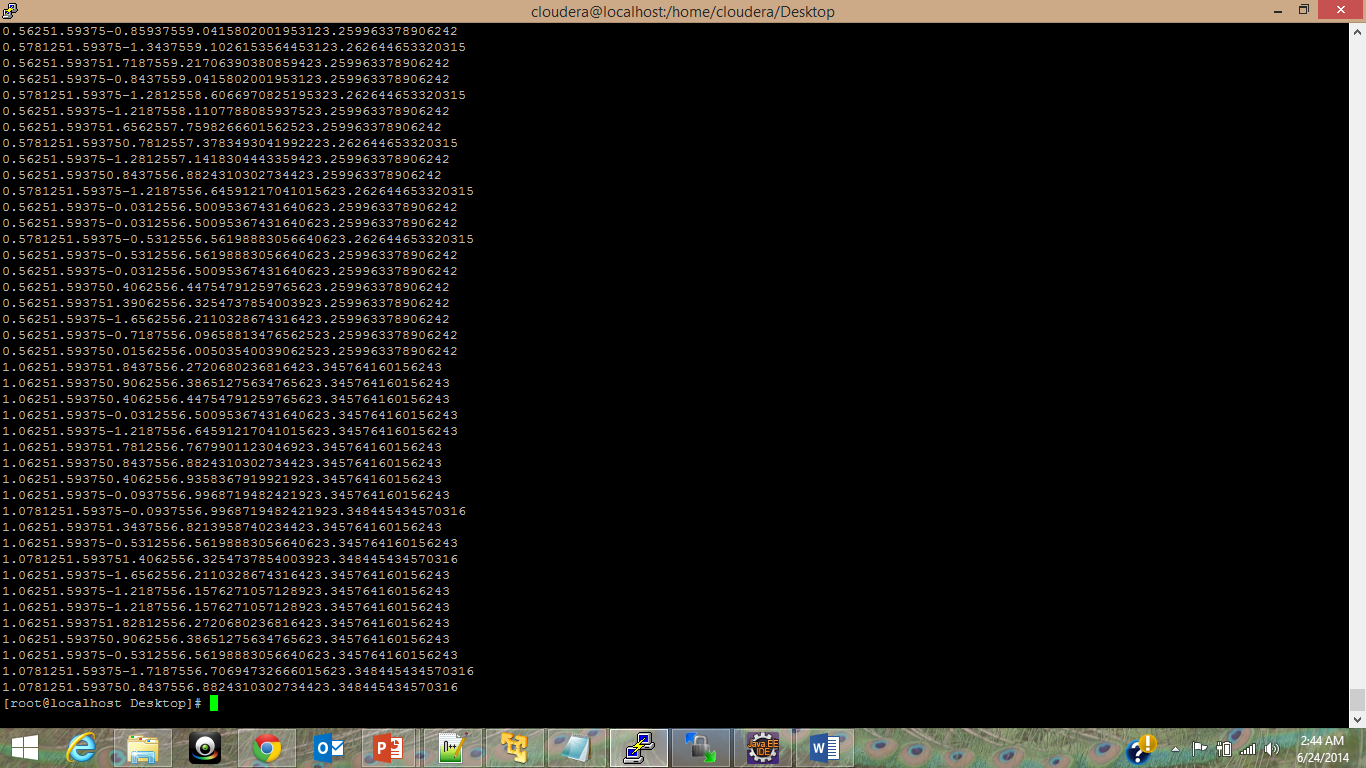


Figure 8insert operation

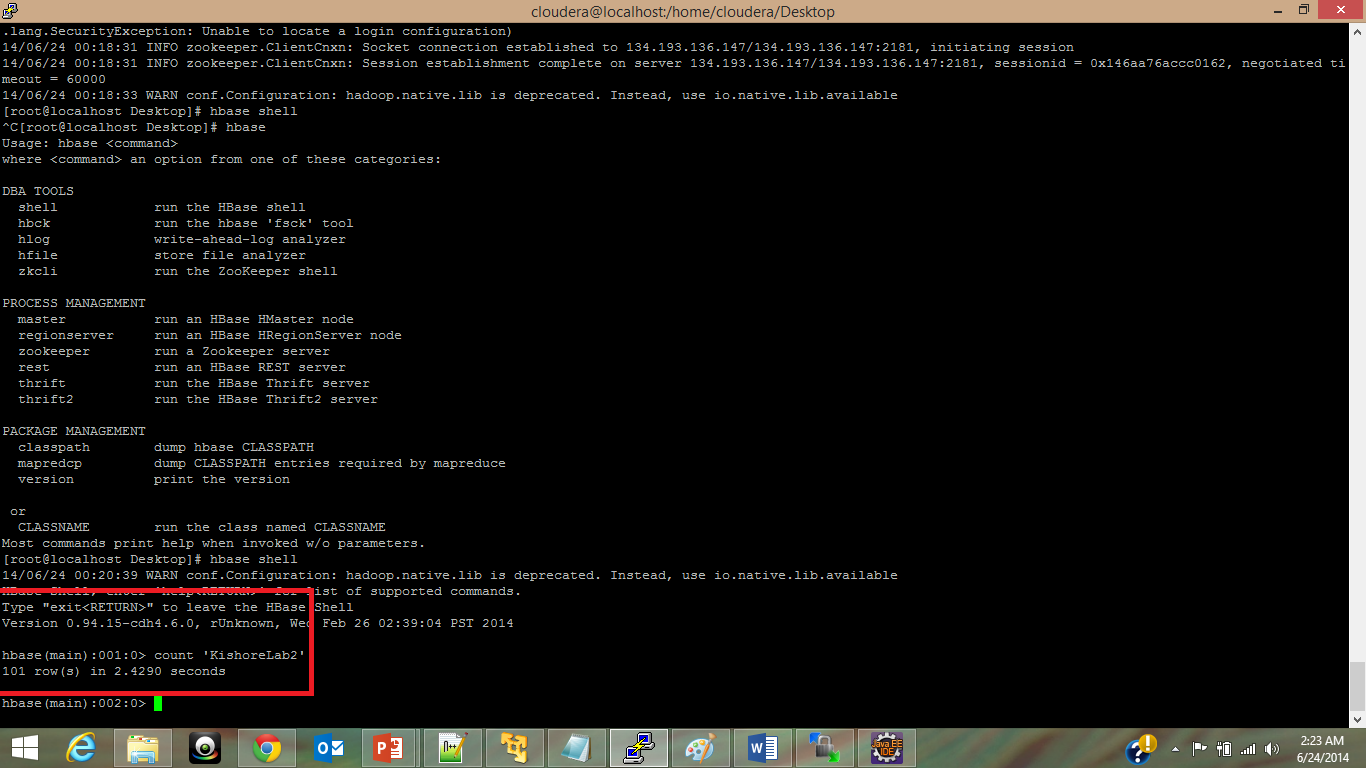


Figure 9Number of rows inserted

**Retrieve Data:**

* Now enable the retreiveTable() function in the program and run the application, here we are doing a get operation so the results will be based on the row key. Instead if we do a scan the results will be retrieved based on the column family.
* The following are the results of the retrieve operation :

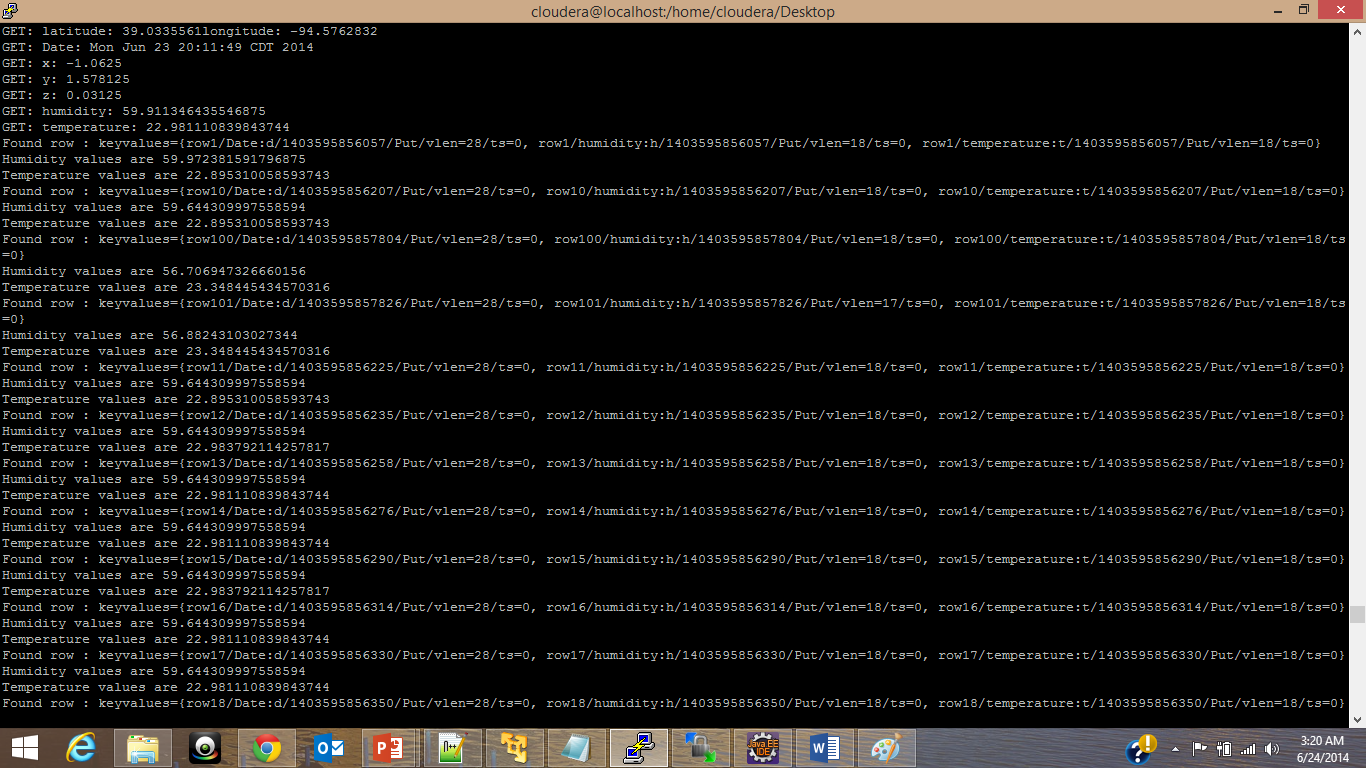


Figure 10Retreived results

In the above figures you can see by using get you will get one row of results of the respective key but by using scan you can get all the results of a column family.

**Troubleshooting:**

If you have any problem with the eclipse while running the HBase program you can use the below commands for running the code:

a) javac -classpath `hbase classpath` HbaseStart.java

b) java -cp `hbase classpath` HbaseStart

These doesn’t require any prior settings, it only needs a cloudera image and hbase in good health.