Implement an application that stores big data in HBase using Hadoop.

Expt No: 8 May 27,2019
Author: Subalakshmi Shanthosi S (186001008)

Aim

Implementation of storage in big data analytics using columnar database - HBase using Hadoop.

Description

1. What is HBase?
   - Apache HBase is a column-oriented key/value data store built to run on top of the Hadoop Distributed File System (HDFS).
   - A non-relational (NoSQL) database that runs on top of HDFS.
   - Provides real-time read/write access to those large datasets.
   - Provides random, real time access to your data in Hadoop.
   - Great choice to store multi-structured or sparse data.
   - Low latency storage .
   - Versioned Database.
   - Installation of HBase in standalone mode.
2. Advantages and Disadvantages of clustering methodologies:
   - Advantages:
     - Great for analytics in association with Hadoop MapReduce.
     - It can handle very large volumes of data.
     - Supports scaling out in coordination with Hadoop file system even on commodity hardware.
     - Fault tolerance.
     - License free.
     - Very flexible on schema design/no fixed schema.
     - Auto Sharding.
     - Row-level atomicity, that is, the PUT operation will either write or fail.
   - Disadvantages:
     - Single point of failure (when only one HMaster is used).
     - No transaction support.
     - JOINs are handled in MapReduce layer rather than the database itself.
     - Indexed and sorted only on key, but RDBMS can be indexed on some arbitrary field.
     - No built-in authentication or permissions.

Column-oriented Database	Row oriented Database
When the situation comes to process and analytics we use this approach. Such as Online Analytical Processing and it's applications.	Online Transactional process such as banking and finance domains use this approach.
The amount of data that can able to store in this model is very huge like in terms of petabytes	It is designed for a small number of rows and columns.

Table 1: Column-oriented vs Row-oriented storages

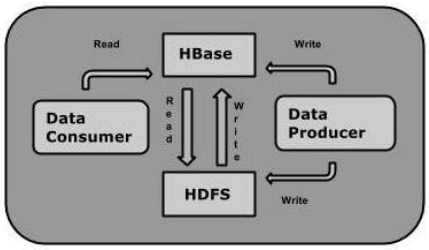


Figure 1: Hadoop Random Access Databases.

Procedure

1. Prepare a Virtual Machine Environment
2. Install Java
3. Install Hadoop
4. Install HBase

HBase Commands

Write about HBase Commands

1. **Create**: create a table in HBase with the specified name given according to the dictionary or specifications as per column family. In addition to this we can also pass some table-scope attributes

$$\textbf{create ¡tablename¿, ¡columnfamilyname¿} \tag{1}$$

2. **Put**: It will put a celli'value' at defined or specified table or row or column.It will optionally coordinate time stamp.

$$\boldsymbol{put <' tablename' >, <' rowname' >, <' columnvalue' >, <' value' >} \tag{2}$$

3. **Scan**: We can pass several optional specifications to this scan command to get more information about the tables present in the system.Scanner specifications may include one or more of the following attributes.These are TIMERANGE, FILTER, TIMESTAMP, LIMIT, MAXLENGTH, COLUMNS, CACHE, STARTROW and STOPROW.

$$\boldsymbol{scan <' tablename' >, Optionalparameters} \tag{3}$$

4. **Get**: You will get a row or cell contents present in the table. In addition to that you can also add additional parameters to it like TIMESTAMP, TIMERANGE,VERSIONS, FILTERS, etc. to get a particular row or cell content.

$$\boldsymbol{get <' tablename' >, <' rowname' >, < Additionalparameters >} \tag{4}$$

5. **Disable**: This command will start disabling the named table.f table needs to be deleted or dropped, it has to disable first.

$$\boldsymbol{disable < tablename >} \tag{5}$$

This command will disable all the tables matching the given regex.The implementation is same as delete command (Except adding regex for matching).Once the table gets disable the user can able to delete the table from HBase.Before delete or dropping table, it should be disabled first.

$$\boldsymbol{disable_all < "matchingregex" >} \tag{6}$$

6. **Drop**: To delete the table present in HBase, first we have to disable it. To drop the table present in HBase, first we have to disable it. So either table to drop or delete first the table should be disable using disable command.Here in above screenshot we are dropping table "education".Before execution of this command, it is necessary that you disable table "education".

$$\boldsymbol{drop < tablename >} \tag{7}$$

Output

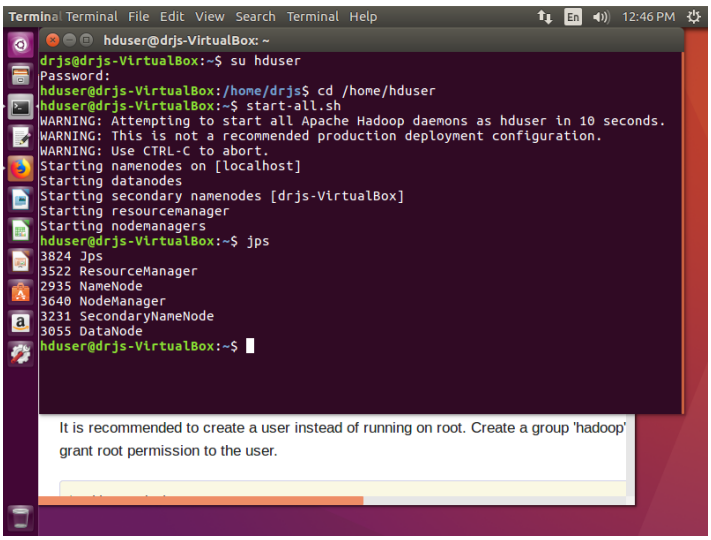


Figure 2: Starting of Hadoop.

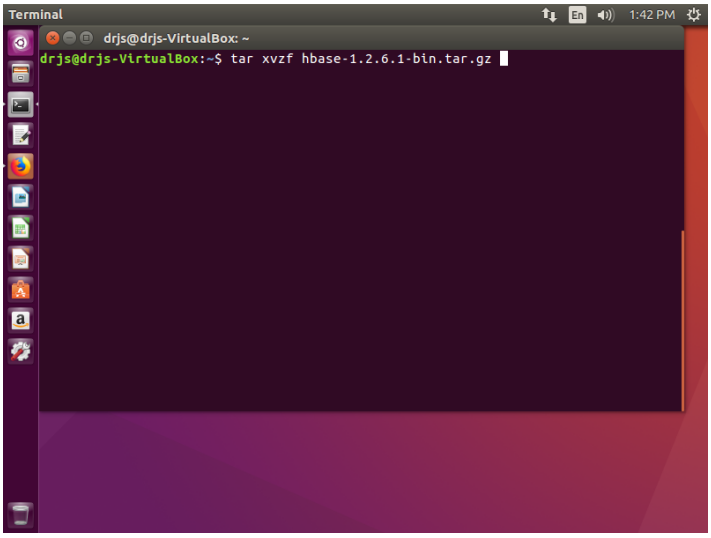


Figure 3: Unzipping the hadoop file using tar command.

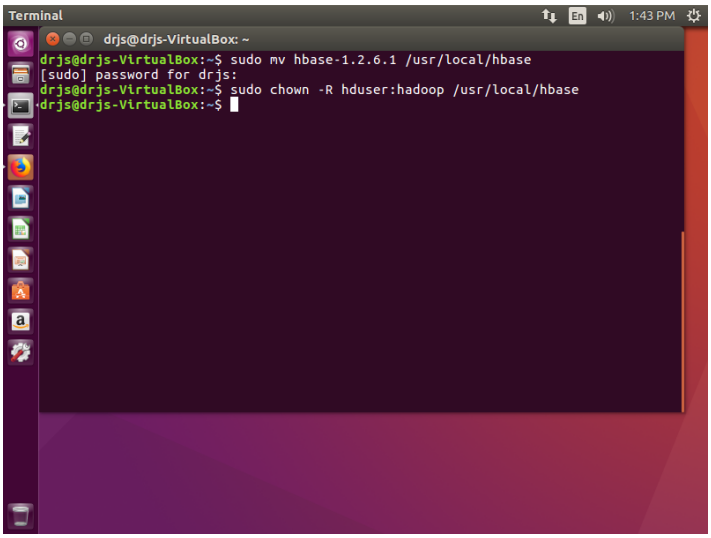


Figure 4: Moving and changing the owner to /usr/local/hadoop.

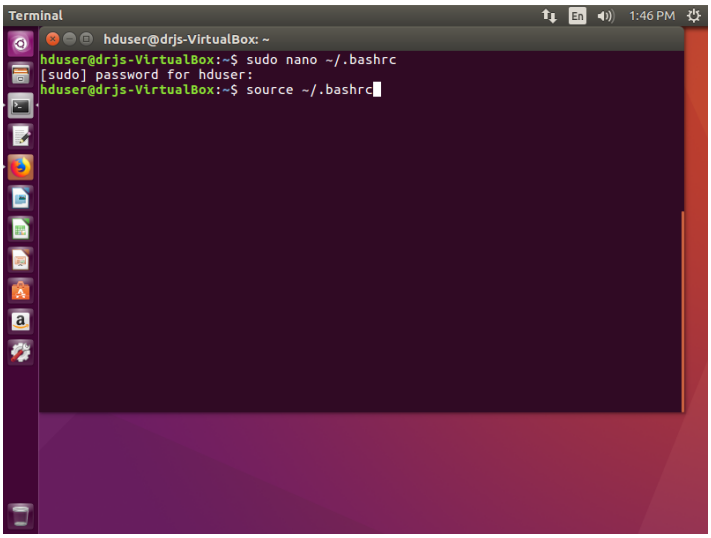


Figure 5: Configuring bashrc.

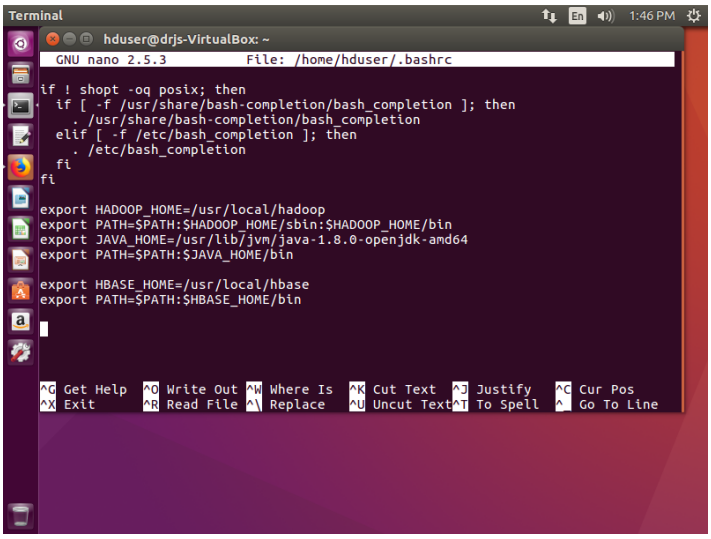


Figure 6: Setting the hbase home and path.

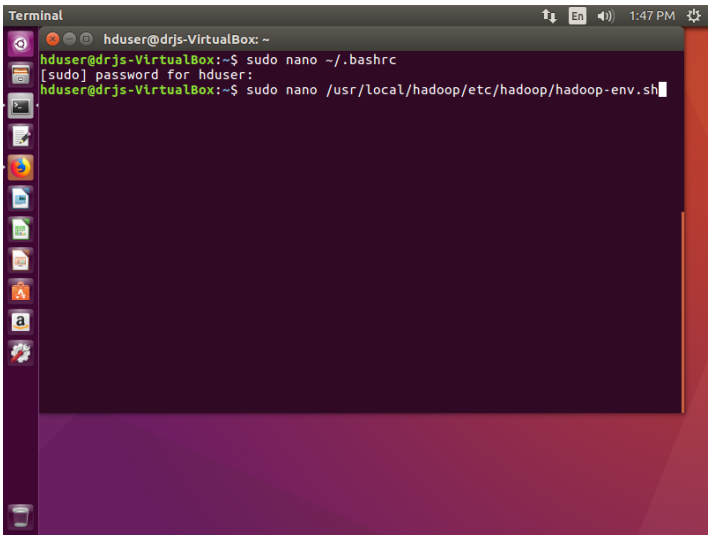


Figure 7: Configuring hadoop-env.sh file.

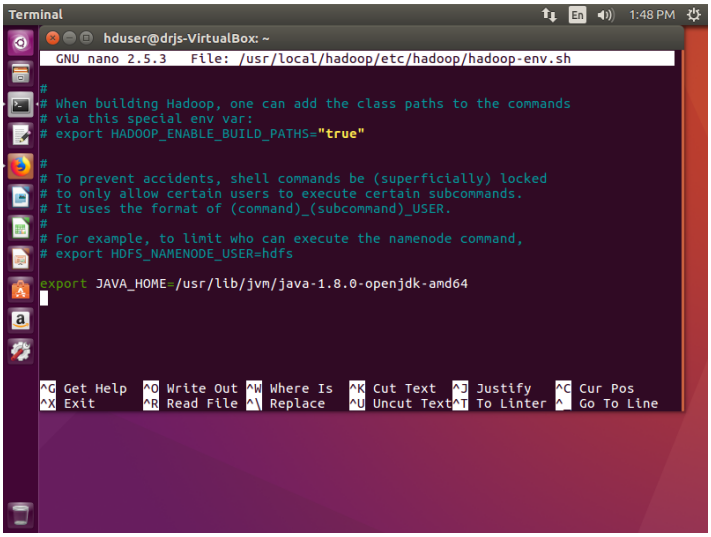


Figure 8: Setting java path.

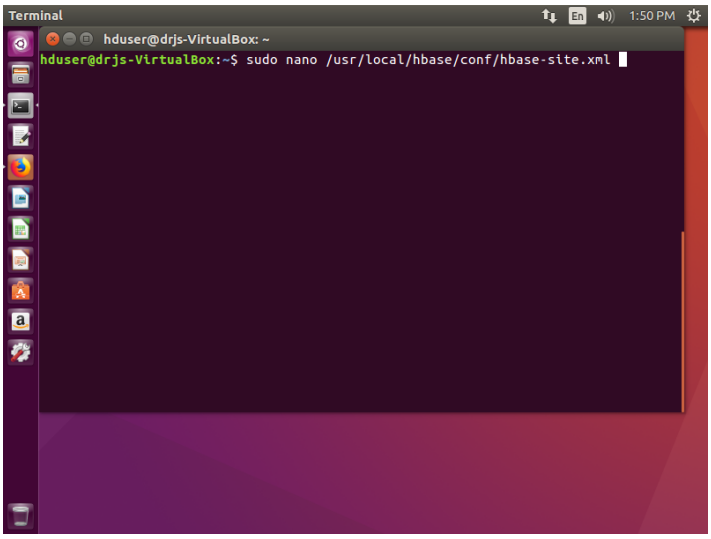


Figure 9: Configuring hbase-site.xml file.

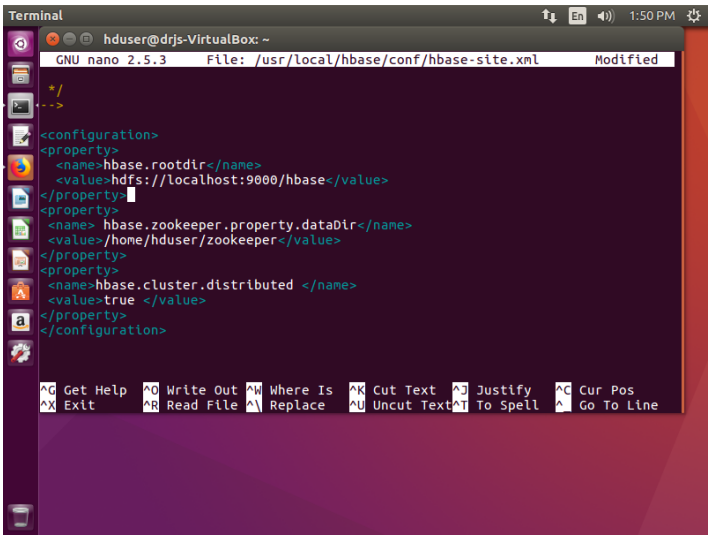


Figure 10: hbase-site.xml.

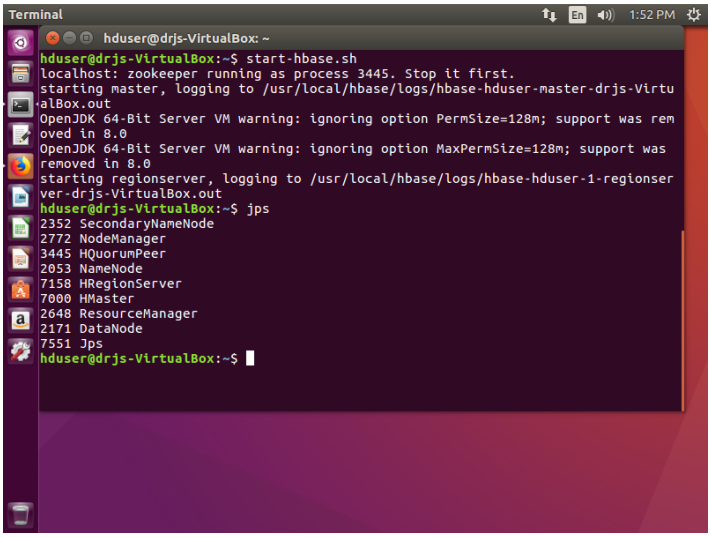


Figure 11: Starting HBase.

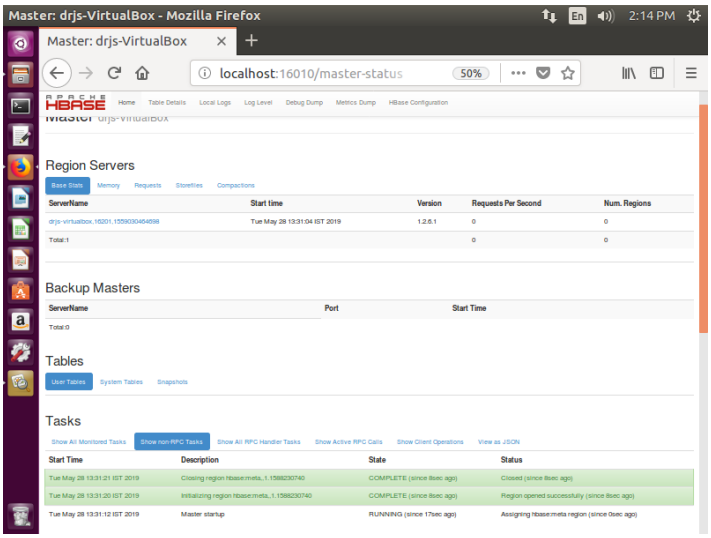


Figure 12: Master status.

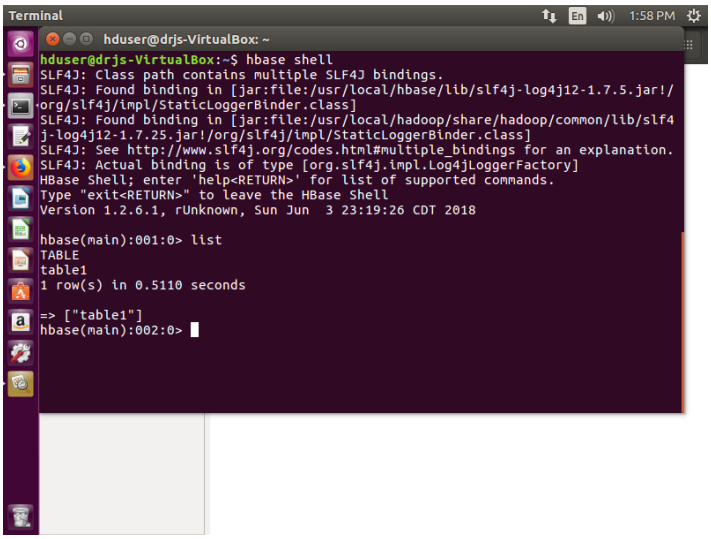


Figure 13: hbase shell.

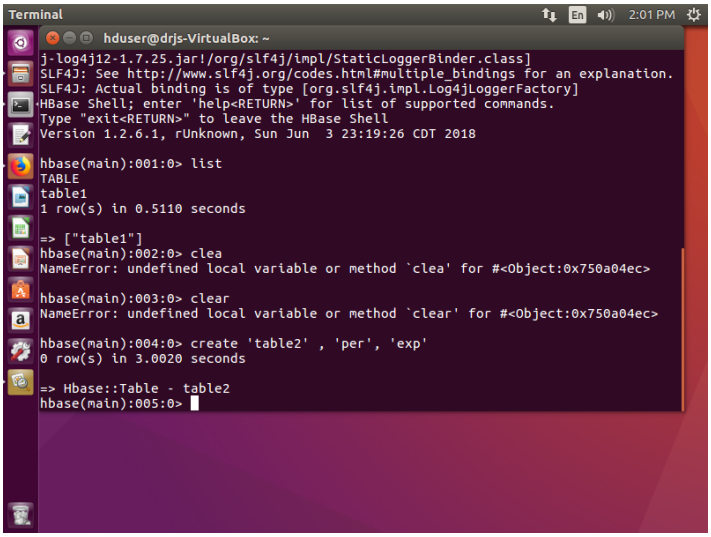


Figure 14: Creation of table.

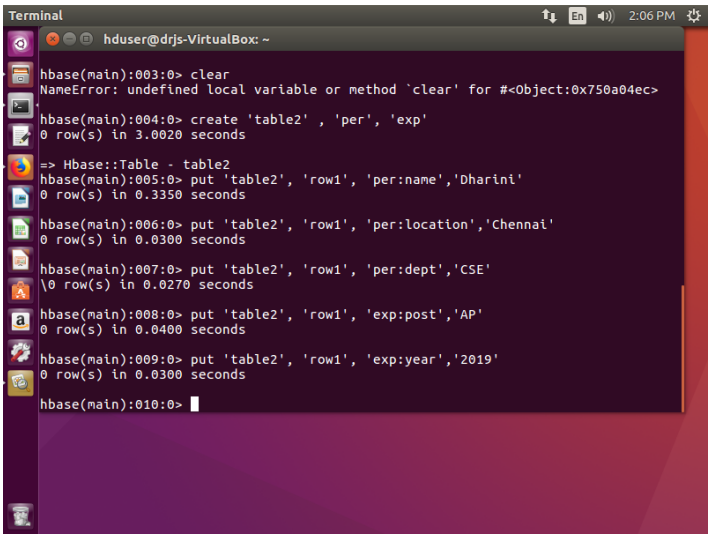
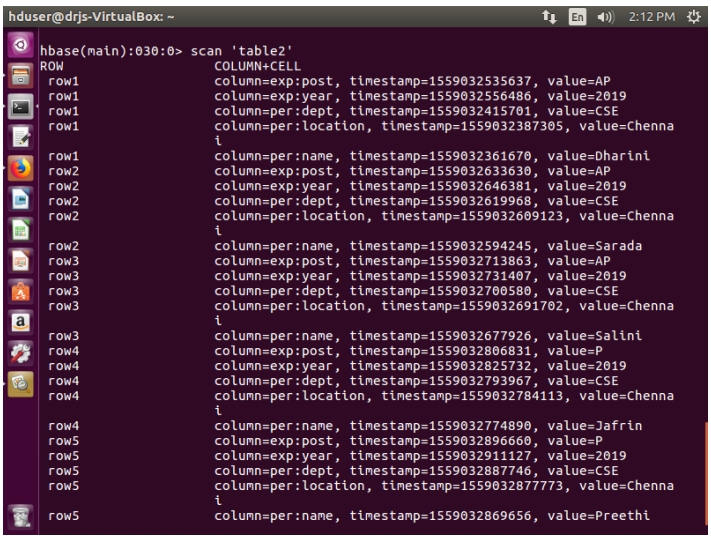
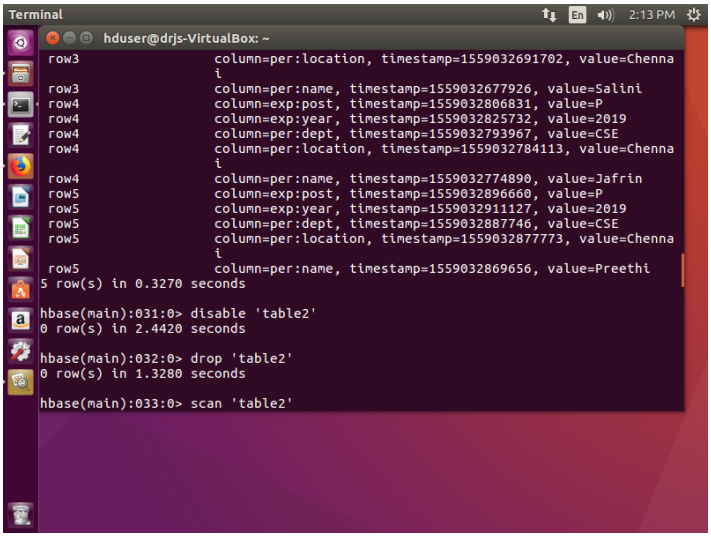


Figure 15: Inserting rows using "put".



```
hduser@drjs-VirtualBox: ~
hbase(main):030:0> scan 'table2'
ROW                    COLUMN+CELL
 row1                  column=exp:post, timestamp=1559032535637, value=AP
 row1                  column=exp:year, timestamp=1559032556486, value=2019
 row1                  column=per:dept, timestamp=1559032415701, value=CSE
 row1                  column=per:location, timestamp=1559032387305, value=Chenna
                       i
 row1                  column=per:name, timestamp=1559032361670, value=Dharini
 row2                  column=exp:post, timestamp=1559032633630, value=AP
 row2                  column=exp:year, timestamp=1559032646381, value=2019
 row2                  column=per:dept, timestamp=1559032619968, value=CSE
 row2                  column=per:location, timestamp=1559032609123, value=Chenna
                       i
 row2                  column=per:name, timestamp=1559032594245, value=Sarada
 row3                  column=exp:post, timestamp=1559032713863, value=AP
 row3                  column=exp:year, timestamp=1559032731407, value=2019
 row3                  column=per:dept, timestamp=1559032700580, value=CSE
 row3                  column=per:location, timestamp=1559032691702, value=Chenna
                       i
 row3                  column=per:name, timestamp=1559032677926, value=Salini
 row4                  column=exp:post, timestamp=1559032806831, value=P
 row4                  column=exp:year, timestamp=1559032825732, value=2019
 row4                  column=per:dept, timestamp=1559032793967, value=CSE
 row4                  column=per:location, timestamp=1559032784113, value=Chenna
                       i
 row4                  column=per:name, timestamp=1559032774890, value=Jafrin
 row5                  column=exp:post, timestamp=1559032896660, value=P
 row5                  column=exp:year, timestamp=1559032911127, value=2019
 row5                  column=per:dept, timestamp=1559032887746, value=CSE
 row5                  column=per:location, timestamp=1559032877773, value=Chenna
                       i
 row5                  column=per:name, timestamp=1559032869656, value=Preethi
```

Figure 16: Displaying the rows in table using "scan".



```
Terminal
hduser@drjs-VirtualBox: ~
 row3                  column=per:location, timestamp=1559032691702, value=Chenna
                       i
 row3                  column=per:name, timestamp=1559032677926, value=Salini
 row4                  column=exp:post, timestamp=1559032806831, value=P
 row4                  column=exp:year, timestamp=1559032825732, value=2019
 row4                  column=per:dept, timestamp=1559032793967, value=CSE
 row4                  column=per:location, timestamp=1559032784113, value=Chenna
                       i
 row4                  column=per:name, timestamp=1559032774890, value=Jafrin
 row5                  column=exp:post, timestamp=1559032896660, value=P
 row5                  column=exp:year, timestamp=1559032911127, value=2019
 row5                  column=per:dept, timestamp=1559032887746, value=CSE
 row5                  column=per:location, timestamp=1559032877773, value=Chenna
                       i
 row5                  column=per:name, timestamp=1559032869656, value=Preethi
5 row(s) in 0.3270 seconds

hbase(main):031:0> disable 'table2'
0 row(s) in 2.4420 seconds

hbase(main):032:0> drop 'table2'
0 row(s) in 1.3280 seconds

hbase(main):033:0> scan 'table2'
```

Figure 17: Disabling and Dropping the table 2.

1 Result

Thus the implementation of Hbase using hadoop is executed successfully.