**Practical 8**

**Aim: Using HBase Tool**

**What is HBase?**

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 JavaTM much like a typical Apache MapReduce application. HBase does support writing applications in Apache Avro, REST and Thrift.

A 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.

Avro, as a component, supports a rich set of primitive data types including: numeric, binary data and strings; and a number of complex types including arrays, maps, enumerations and records. A sort order can also be defined for the data.

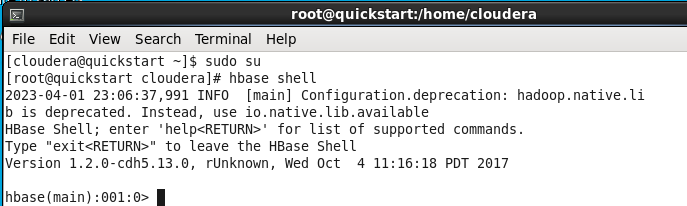
HBase relies on ZooKeeper for high-performance coordination. ZooKeeper is built into HBase, but if you’re running a production cluster, it’s suggested that you have a dedicated ZooKeeper cluster that’s integrated with your HBase cluster.

HBase works well with Hive, a query engine for batch processing of big data, to enable fault tolerant big data applications.

**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.

1. Start with HBase Shell.



1. Check for HBase Prompt. If not, restart the services.

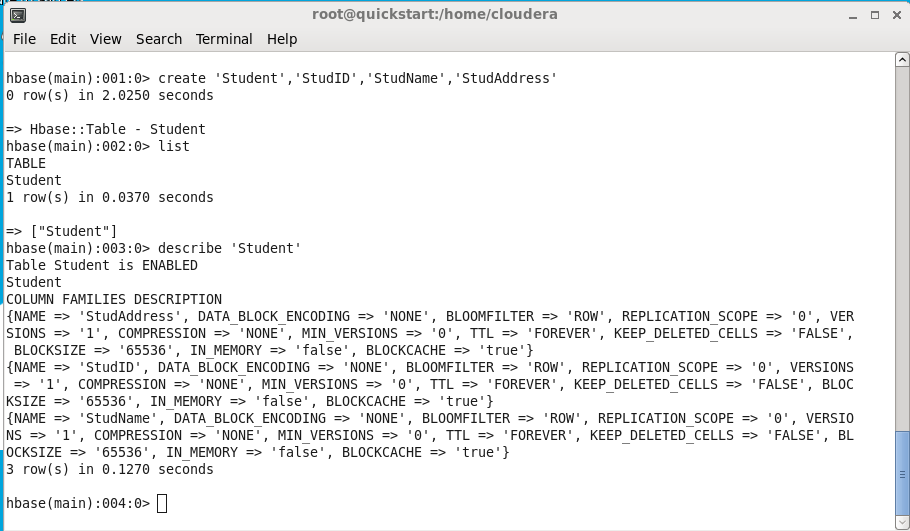
$ sudo su – This commands is to become super user

$ service hbase-master restart – This commands is to restart hbase-master services

$ service hbase-regionserver restart – This commands is to restart hbase-regionserver services



1. Create a HBase table named ‘Student’ with column families StudID, StudName and StudAddress. Also cross check the creation and schema details of the table.



1. Insert following data into the student table.

(Syntax: put 'Student','S01','StudName:FName','Amit' )

S01,Amit Prakash Vaze, Mulund

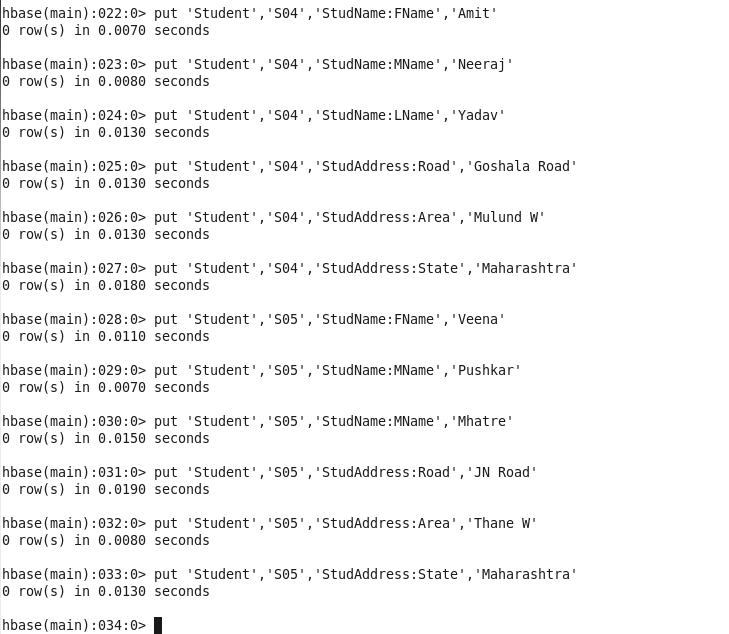
S02,Sumit Neeraj Kulkarni, Devi Dayal Road, Mulund W

S03,Alka Deepak Yadav, PK Road, Thane, Maharashtra

S04,Amit Neeraj Yadav, Goshala Road, Mulund W, Maharashtra

S05,Veena Pushkar Mhatre, JN Road, Thane W, Maharashtra

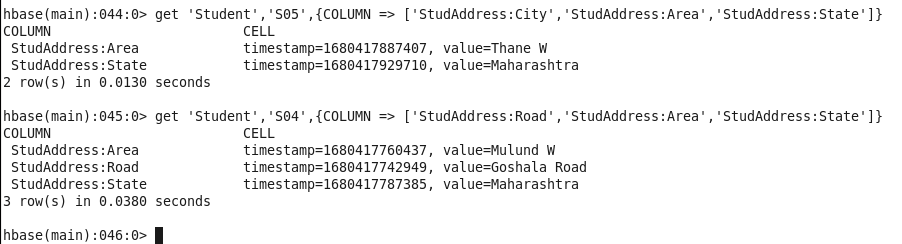
StudName -> FName MName LName StudAddress -> Road, Area, State



1. Display all rows from the Student table. (scan command)



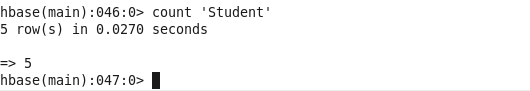
1. Display address list of all students. (get command)



1. Display list of students.

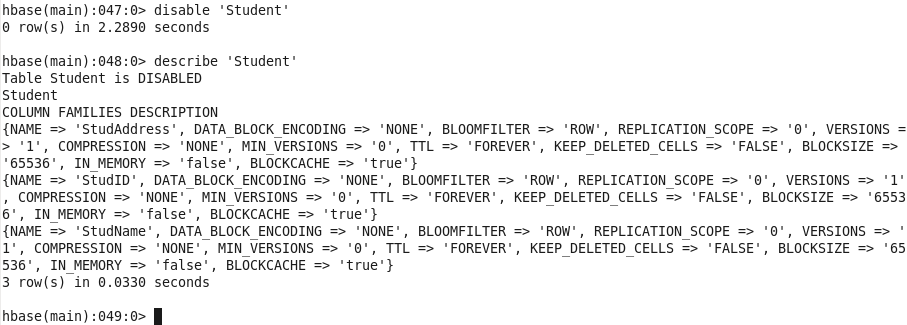


1. Display total number of students.



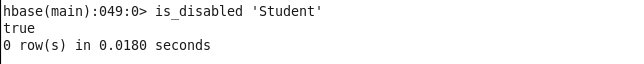
1. Disable Student table and display the table schema using describe command.

>disable ‘Student’



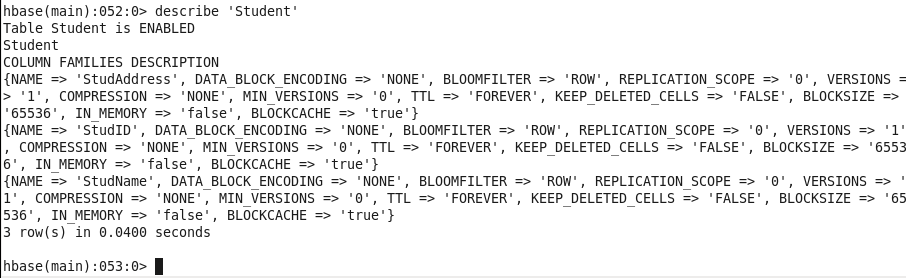
1. Check disabled status of the table using is-disabled command.

>is\_disabled ‘Student’

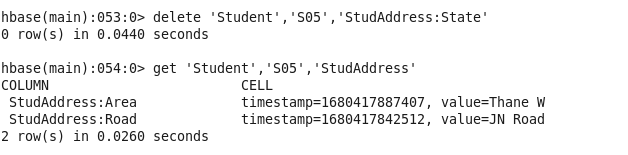


1. Enable Student table and display the table schema.



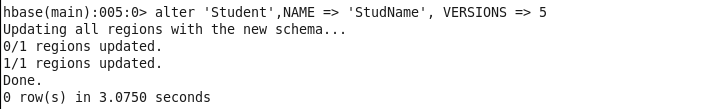


1. Demonstrate the deletion of column values and a row.



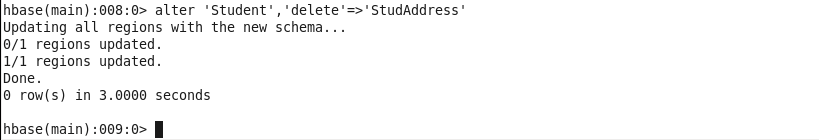
1. Set the maximum number of cell changes to 5.

>alter ‘Student’, Name=’NewColFamily’, VERSIONS=>5



1. Delete column family ‘Address’ from Student table.

>alter ‘Student’,‘delete’ => ‘Address’



1. Check the existence of the Student table.

>exists ‘Student’



1. Drop Table ‘Student’

