CATEGORY 2: HADOOP ECOSYSTEM

APACHE HBASE TOOL/FRAMEWORK

# PART 1

The Apache HBase (database management system) is a column-oriented system that operates at top of the Hadoop Distributed File System (Hadoop Distributed File System). HBase is a replica of Google's Bigtable, another distributed storage solution for structured data. Apache HBase delivers Bigtable-like functionality on top of Hadoop and HDFS, just as Bigtable does by utilizing the distributed data storage provided by the Google File System. HBase's purpose is to provide a platform on top of clusters of commodity hardware for the storage and management of massive tables with billions of rows and millions of columns. The HBase database is a key-value store, making it a kind of the non-relational, or NoSQL, database. When creating a table with HBase, column families, rather than individual columns, are specified.

HBASE'S ARCHITECTURAL PARTS/COMPONENTS

The three parts of HBASE are as follows:

HBase's Master Server is implemented using HMaster. As part of this procedure, regions are sent to the region server, and DDL (new, delete table) actions are carried out. It keeps an eye on each and every instance of Region Server within the cluster. Master, when used in a distributed setting, will have multiple threads running in the background. HMaster's various functions include load balancing and failover control. etc.

HBase tables are split into horizontal sections called Regions based on the range of row keys. Tables in an HBase cluster are organized into "regions," and each region is made up of a collection of Column families. HDFS DataNode in a Hadoop cluster is where the Region Server is hosted and managed. HBase reads and writes, as well as other administrative tasks, are all under the purview of the Region Server's regions. Standard region size is 256 MB.

Zookeeper is HBase's equivalent of a coordinator. It offers functionality including configuration management, name resolution, distributed synchronization, server failure notification, and more. For clients to talk to their respective region servers, zookeeper is used.

### PART 2

NoSQL distributed massive data store Apache HBase is available for free. Access to petabytes of data can be made available at any time in a manner that is both random and consistent. HBase excels at managing massive, sparse datasets.

Alternatives to Apache HBASE

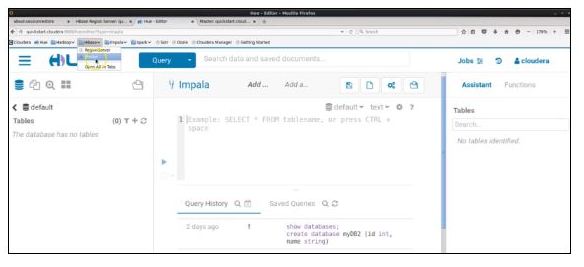
Apache HBase isn't the only database option available. We'll go over a few of them below:

1. Snowflakes - Snowflake's cloud data platform breaks through the walls that have kept businesses of all sizes from maximizing their data's potential. Snowflake has been adopted by thousands of customers who have seen unprecedented growth thanks to the insights gleaned from their data by their entire staff of business users. Snowflake provides businesses with everything they need in one unified platform: the only data warehouse designed for the cloud; instant, secure, and governed access to their entire data network; and a core architecture to enable many different types of data workloads, including a single platform for developing cutting-edge data applications.
2. Google Cloud BigQuery - To analyze Big Data on the cloud, try Google Cloud BigQuery. Submit SQL-like queries to datasets that are many terabytes in size and get results back in a matter of seconds. BigQuery provides real-time insights into your data, and it's scalable and simple to use.
3. Amazon Redshift - Using Amazon Redshift, a fast and fully-managed data warehouse, you can easily and affordably analyze all of your data using just ordinary SQL and your current Business Intelligence (BI) tools.
4. Google Cloud BigTable - Distributed storage solution for managing structured data, Google Cloud Bigtable can handle petabytes of information over thousands of commodity computers.

### PART 3

Practical example(Demo) - use of commands

We'll be using Cloudera QuickStart, which was pre-installed on Oracle VirtualBox, to get some hands-on experience with the platform. As a first step, open the Hue interface and choose the HBase Master node as displayed below.



By selecting the Master, you can examine data on the software's configuration in general, as well as its regional servers, tables, jobs, and ZooKeeper version. Then, to get started with our working example, launch a terminal window. The terminal window can be magnified to make typing easier. You can begin by typing to enter the HBase shell.

Hbase shell // command to open the HBase shell



You'll be able to enter the HBase shell and begin typing HBase commands after a brief delay. The following commands will get you started:

List // lists down all tables available in HBase

create ‘newtbl’, ‘knowledge’ // creates a new table in HBase

describe ‘newtbl’ // confirms if table was created

status ‘summary’ // checks the status of HBase

With the new created table, lets add some data;

put ‘newtbl’, ‘r1’, ‘knowledge:sports’, ‘cricket’

put ‘newtbl’, ‘r1’, ‘knowledge: science’, ‘chemistry’

put ‘newtbl’, ‘r1’, ‘knowledge:science’, ‘physics’

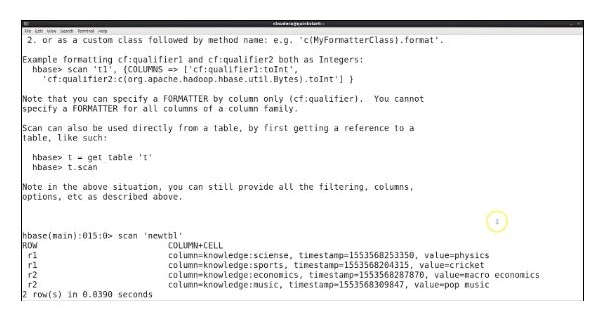
put ‘newtbl’, ‘r2’, ‘knowledge:economics’, ‘macro economics’

put ‘newtbl’, ‘r2’, ‘knowledge:music’, ‘pop music’

Let's now type out a list of the table's contents:

Scan ‘newtbl’

The results will look like this:



Since the above graphic only shows the most recent update (in this case, "physics"), it is impossible to view "chemistry." The following commands can now be entered:

Is\_enabled ‘newtbl’ //confirms if table is enabled

Disable ‘newtbl’ //disables the table in HBase

Scan ‘newtbl’ //lists the contents of the table. Since our table is disabled, this will throw an error.

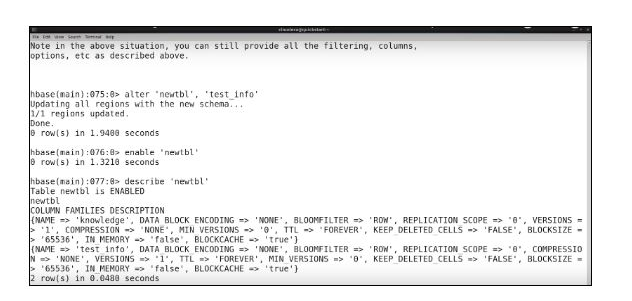
// With that said, lets do alterations to enable our table

alter ‘newtbl’, ‘test\_info’ // Updates column family in the table

enable ‘newtbl’ //enables the table

describe ‘newtbl’ //checks the column families after updation

Here is the output:



Then, using the following commands, we can extract values from a specific row and also learn how to update existing data in a row.

get ‘newtbl’, ‘r1’ // extracts the values from r1 in the tables

put ‘newtbl’, ‘r1’, ‘knowledge:economics’, ‘market economics’ //Adds new info to r1. This updates the table but does not override the info.

get ‘newtbl’, ‘r1’ //displays the results for r1

The output looks as below;

