Name : Bhavani Rajpurohit

Class : AIA-3 Subject: DBMS LAB

Roll No : 2213688 Batch : B

ASSIGNMENT NO: 11

Aim:

Use HBase to perform following operations

1.Create a table

2.Add, Retrieve, Modify, and delete the record(s)

3.Drop the table

Software required:

1. Ubuntu 18 / 18

2. Hadoop 3.0.0 framework

3. HBase

Theory:

HBase – Introduction

- Apache HBase is a distributed, column-oriented database built on top of HDFS.
- Apache HBase can scale horizontally to thousands of commodity servers and petabytes of data by indexing the storage.
- Apache HBase is an open-source, distributed, and versioned non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data.
- The goal of HBase is to host very large tables with billions of rows and millions of columns, atop clusters of commodity hardware.

Characteristics of HBase

HBase is a type of NoSQL database and is classified as a key-value store. In HBase,

- Value is identified with a key.
- Both key and value are byte-array, which means binary formats can be stored easily.

- Values are stored in key-orders and can be accessed quickly by their keys.
- HBase is a database in which tables have no schema. Column families and not columns are defined at the time of table creation.

HBase Architecture

HBase has two types of Nodes which are Master and RegionServer.

Master

- There is only one Master node running at a time whereas there can be one or more RegionServers. The high availability of the Master node is maintained with ZooKeeper.
- The Master node manages cluster operations like an assignment, load balancing, and splitting. It is not a part of read or write path.

RegionServer

- The RegionServer hosts tables, performs reads, and buffers writes. Clients communicate with RegionServer to read and write.
- A region in HBase is the subset of a table's rows. The Master node detects the status of RegionServers and assigns regions to RegionServers.

Commands for HBase

Starting HBase

```
./bin/hbase shell
```

Create Table Statement

```
hbase> create 'emp', 'personal data', 'professional data'
```

Verification

```
hbase> list
```

Disabling a Table using HBase

```
hbase> disable 'emp'
```

Enabling a Table using HBase

```
hbase> enable 'emp'
```

Describe

```
hbase> describe 'table name'
```

Alter

```
hbase> alter 't1', NAME ⇒ 'f1', VERSIONS ⇒ 5
```

Dropping a Table

```
hbase> disable 'emp'
```

Add data in the table

```
put ’<table name>’,’row1’,’<colfamily:colname>’,’<value>’

hbase> put 'emp','1','personal data:name','raju'

hbase> put 'emp','1','professional data:designation','manager

hbase> scan 'emp'
```

Updating Data

```
Hbase> put 'emp','row1','personal:city','Delhi'
```

Retrieve

```
        get ’<table name>’,’row1’

hbase> get 'emp', '1'
```

Reading a Specific Column

```
hbase> get 'table name', ‘rowid’, {COLUMN ⇒ ‘column family:column name ’}

hbase> get 'emp', 'row1', {COLUMN ⇒ 'personal:name'}
```

Deleting a Specific Cell in a Table

```
        delete ‘<table name>’, ‘<row>’, ‘<column name >’, ‘<time stamp>’

hbase> delete 'emp', '1', 'personal data:city',1417521848375
```

Deleting All Cells in a Table

```
deleteall ‘<table name>’, ‘<row>’,

hbase(main):007:0> deleteall 'emp','1'
```

Stopping HBase

```
./bin/stop-hbase.sh
```

Conclusion:

Using HBase we are able to perform the operations like Creating an employee table ,Adding Retrieve, Modify, and delete the records in the employee table. Finaly we deleted the table using disable command