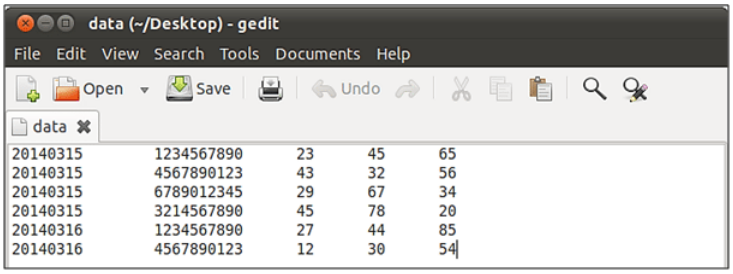
Sample HBase POC

Priyanka (https://www.edureka.co/blog/author/priyanka/) | Jul 16,2018

(https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2014/03/HBase.png)

In this blog we shall discuss about a sample Proof of Concept for HBase.

Here we have a Data set as in the below image.



(https://www.edureka.co/blog/sample-hbase-poc/)

This data set consists of the details about the duration of total incoming calls, outgoing calls and the messages sent from a particular mobile number on a specific date.

The first field represents date, the second field represents mobile number, the third field represents the total duration of incoming calls, fourth field represents total duration of outgoing calls, and fifth field represents the total number of messages sent.

Now our task is to retrieve the information of the duration of incoming and outgoing calls and messages sent, from a phone number on a particular date.

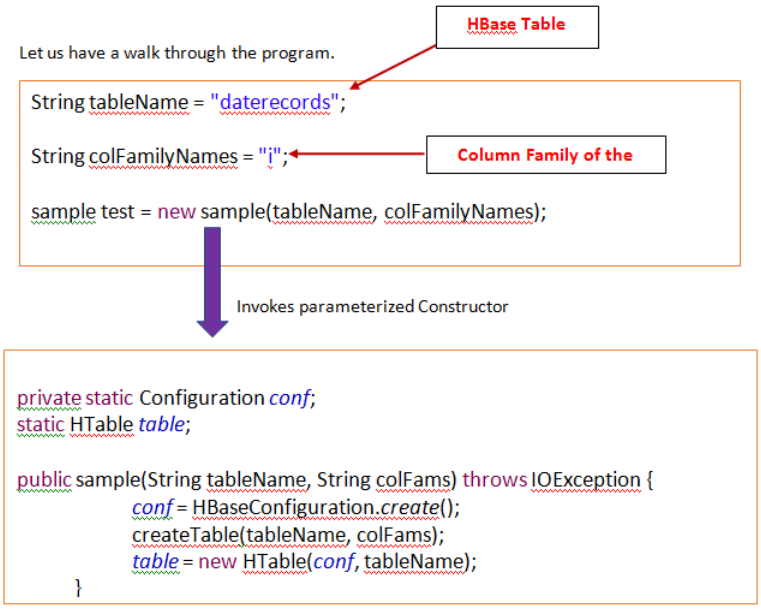
In this use case, I am trying to filter the records of 15th March 2014. Here is an HBase Program to achieve this.

Below is the complete code of it.

```
public class sample {
    private static Configuration conf;
    static HTable table;
  public sample(String tableName, String colFams) throws IOException {
        conf = HBaseConfiguration.create();
        createTable(tableName, colFams);
        table = new HTable(conf, tableName);
    }
    void createTable(String tableName, String colFams) throws IOException {
        HBaseAdmin hbase = new HBaseAdmin(conf);
        HTableDescriptor desc = new HTableDescriptor(tableName);
        HColumnDescriptor meta = new HColumnDescriptor(colFams.getBytes());
        desc.addFamily(meta);
        hbase.createTable(desc);
    }
```

```
                String colFamilyName, String colName, String values)
                throws IOException {
        byte[] rowKey = Bytes.toBytes(row);
        Put putdata = new Put(rowKey);
        putdata.add(Bytes.toBytes(colFamilyName), Bytes.toBytes(colName),
                Bytes.toBytes(values));
        table.put(putdata);
    }
    public static void getAllRecord(String tableName, String startPartialKey,
                String endPartialKey) throws IOException {
        try {
            Scan s;
            if (startPartialKey == null || endPartialKey == null)
                s = new Scan();
            else
                s = new Scan(Bytes.toBytes(startPartialKey),
                        Bytes.toBytes(endPartialKey));
            ResultScanner ss = table.getScanner(s);
           HashMap<String, HashMap<String, String>> outputRec = new HashMap<String, HashMap<String, String>>();
            String imsi = "";
            for (Result r : ss) {
                HashMap<String, String> keyVal = new HashMap<String, String>();
                for (KeyValue kv : r.raw()) {
                    imsi = new String(kv.getRow()).substring(10);
                    keyVal.put(new String(kv.getQualifier()),
                            new String(kv.getValue()));
                    outputRec.put(imsi, keyVal);
                    if (keyVal.size() == 3)
                        System.out.println(imsi + "\t" + "Incoming minutes:"
                                + keyVal.get("c1") + "\t Outcoming minutes:"
                                + keyVal.get("c2") + "\t Messages:"
                                + keyVal.get("c3"));
                }
            }
        } finally {
        }
    }
    public static void main(String[] args) throws IOException {
        String tableName = "daterecords";
        String colFamilyNames = "i";
       sample test = new sample(tableName, colFamilyNames);
        String fileName = "/home/cloudera/Desktop/data";
        // This will reference one line at a time
        String line = null;
```

```
                // FileReader reads text files in the default encoding.
                FileReader fileReader = new FileReader(fileName);
                // Always wrap FileReader in BufferedReader.
                BufferedReader bufferedReader = new BufferedReader(fileReader);
                while ((line = bufferedReader.readLine()) != null) {
                        String[] values = line.split("\t");
                        addColumnEntry(tableName, values[0] + "-" + values[1],
                                        colFamilyNames, "c1", values[2]);
                        addColumnEntry(tableName, values[0] + "-" + values[1],
                                        colFamilyNames, "c2", values[3]);
                        addColumnEntry(tableName, values[0] + "-" + values[1],
                                        colFamilyNames, "c3", values[4]);
                }
                bufferedReader.close();
        } catch (FileNotFoundException ex) {
                System.out.println("Unable to open file '" + fileName + "'");
        } catch (IOException ex) {
                System.out.println("Error reading file '" + fileName + "'");
                // Or we could just do this:
                // ex.printStackTrace();
        }
        getAllRecord(tableName, "20140315", "20140316");
    }
}
```



(https://www.edureka.co/blog/sample-hbase-poc/)

Here we have created an object of Configuration, HTable class and creating the Hbase Table with name: **daterecords** and the column family: **i**.

In this use case, we will be taking the combination of date and mobile number separated by '-' as row key for this Hbase table and the incoming , outgoing call durations', the number of messages sent as the columns 'c1', 'c2', 'c3' for the column family 'i'.

We have the input data stored in the local file system of Cloudera. So we need to write Java Logic that reads the data from the file.

Below is the Java logic.

```
String line = null;

try {

FileReader fileReader = new FileReader(fileName);
BufferedReader bufferedReader = new
BufferedReader(fileReader);

while ((line = bufferedReader.readLine()) != null)
{

        String[] values = line.split("\t");
        addColumnEntry(tableName, values[0] + "-" + values[1],
                                colFamilyNames, "c1", values[2]);
```

Calls the Method

```
public static void addColumnEntry(String tableName, String row,
                String colFamilyName, String colName, String values)
                        throws IOException {

                byte[] rowKey = Bytes.toBytes(row);
                Put putdata = new Put(rowKey);
                putdata.add(Bytes.toBytes(colFamilyName), Bytes.toBytes(colName),
                                Bytes.toBytes(values));
                table.put(putdata);

        }
```

(https://www.edureka.co/blog/sample-hbase-poc/)

In this method we are storing the data into the table for each column of the column family.

We can check the data stored in Hbase table 'daterecords' by using the scan command.

You will receive the data as in the below image.

```
hbase(main):002:0> scan 'daterecords'
ROW                  COLUMN+CELL
 20140315-1234567890  column=i:c1, timestamp=1395056542634, value=23
 20140315-1234567890  column=i:c2, timestamp=1395056542636, value=45
 20140315-1234567890  column=i:c3, timestamp=1395056542636, value=65
 20140315-3214567890  column=i:c1, timestamp=1395056542642, value=45
 20140315-3214567890  column=i:c2, timestamp=1395056542643, value=78
 20140315-3214567890  column=i:c3, timestamp=1395056542643, value=20
 20140315-4567890123  column=i:c1, timestamp=1395056542637, value=43
 20140315-4567890123  column=i:c2, timestamp=1395056542638, value=32
 20140315-4567890123  column=i:c3, timestamp=1395056542638, value=56
 20140315-6789012345  column=i:c1, timestamp=1395056542640, value=29
 20140315-6789012345  column=i:c2, timestamp=1395056542641, value=67
 20140315-6789012345  column=i:c3, timestamp=1395056542641, value=34
 20140316-1234567890  column=i:c1, timestamp=1395056542644, value=27
 20140316-1234567890  column=i:c2, timestamp=1395056542644, value=44
 20140316-1234567890  column=i:c3, timestamp=1395056542645, value=85
 20140316-4567890123  column=i:c1, timestamp=1395056542646, value=12
 20140316-4567890123  column=i:c2, timestamp=1395056542647, value=30
 20140316-4567890123  column=i:c3, timestamp=1395056542647, value=54
6 row(s) in 0.3390 seconds
```

(https://www.edureka.co/blog/sample-hbase-poc/)

Now we have inserted the data in to the HBase Table successfully.

Let us retrieve the records stored in the Table of a Particular date.

In this use case, we are trying to retrieve the records of the Date: 15th March 2014

To retrieve the records we have created a Method

getAllRecord(String tableName, String startPartialKey, String endPartialKey)

The First Parameter represents the table name, the second represents the start date from which we need to retrieve the data and the third one is the next date of start date.

E.g:

getAllRecord(tableName, "20140315", "20140316");

Now let us understand the **logic** of this method.

```
Scan s;
```

```
        s = new Scan();
else
        s = new Scan(Bytes.toBytes(startPartialKey),
```

(https://www.edureka.co/blog/sample-hbase-poc/)

We are trying to scan the Hbase Table by Using HBase API with the help of startPartialKey and endPartialKey.

As StartPartialKey -and endPartialkey are not null-, it will go to else block and scan the records having the value of startPartialKey.

```
Bytes.toBytes(endPartialKey));

ResultScanner ss = table.getScanner(s);

HashMap<String, HashMap<String, String>> outputRec = new
HashMap<String, HashMap<String, String>>();
```

(https://www.edureka.co/blog/sample-hbase-poc/)

We have created an object of Result scanner which stores the scanned records of the Hbase table and a HashMap to store the output that will be result.

```
String imsi = "";

for (Result r : ss) {

HashMap<String, String> keyVal = new HashMap<String, String>();

for (KeyValue kv : r.raw()) {

imsi = new String(kv.getRow()).substring(10);

keyVal.put(new String(kv.getQualifier()), new String(kv.getValue()));

outputRec.put(imsi, keyVal);

if (keyVal.size() == 3)

System.out.println(imsi + "\t" + "Incoming minutes:"
                                + keyVal.get("c1") + "\t Outcoming minutes:"
                                                + keyVal.get("c2") + "\t
Messages:"                                                          +
keyVal.get("c3"));
}

}
```

(https://www.edureka.co/blog/sample-hbase-poc/)

We are creating an object of Result to get the data store in the Result Scanner and executing a for loop.

imsi is the string that is defined to store the Mobile number and keyVal is a Hash Map that stores the output retrieved from the column of a particular phone.

We have given **20140315-1234567890** as the **rowkey** to the Hbase table. In this 20140315 represents the date and 1234567890 represents the Mobile number.

As we require only the mobile number we are using substring method to retrieve it.

We are retrieving the data from the r.raw() and storing it in the HashMap by using Put.

Finally we are trying to print them on the console.

The Output will be as in the below image.

```
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:java.library.path=/usr/lib/jvm/java-6-sun-1.6
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:java.io.tmpdir=/tmp
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:java.compiler=<NA>
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:os.name=Linux
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:os.arch=i386
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:os.version=2.6.35-28-generic
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:user.name=cloudera
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:user.home=/home/cloudera
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Client environment:user.dir=/home/cloudera/workspace/Hbase POC
14/03/17 06:41:46 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=localhost:2181 sessio
14/03/17 06:41:46 INFO zookeeper.ClientCnxn: Opening socket connection to server localhost/0:0:0:0:0:0:0:1:2
14/03/17 06:41:47 INFO zookeeper.ClientCnxn: Socket connection established to localhost/0:0:0:0:0:0:0:1:2181
14/03/17 06:41:47 INFO zookeeper.ClientCnxn: Session establishment complete on server localhost/0:0:0:0:0:0:0:
234567890       Incoming minutes:23     Outcoming minutes:45    Messages:65
214567890       Incoming minutes:45     Outcoming minutes:78    Messages:20
567890123       Incoming minutes:43     Outcoming minutes:32    Messages:56
789012345       Incoming minutes:29     Outcoming minutes:67    Messages:34
```

(https://www.edureka.co/blog/sample-hbase-poc/)

We have successfully retrieved the records of the Date: 15th March 2014.

About Priyanka (8 Posts (https://www.edureka.co/blog/author/priyanka/))

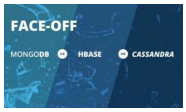
(https://plus.google.com/)

Share on (https://www.facebook.com/sharer.php?u=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fsample-hbase-poc/) (https://twitter.com/home?status=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fsample-hbase-poc/) (https://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fsample-hbase-poc/) (https://plus.google.com/share?url=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fsample-hbase-poc/)

PREVIOUS NEXT

Related Posts

Insights on HBase Architecture
13.1K
(https://www.edureka.co/blog/insights-on-hbase-architecture)



Face Off: MongoDB Vs HBase Vs Cassandra
4.8K
(https://www.edureka.co/blog/face-off-mongodb-vs-hbase-vs-cassandra/)

Top Hive Commands with Examples in HQL
174.1K
(https://www.edureka.co/blog/top-hive-commands-with-examples)

How to Set Up Hadoop Cluster with HDFS High Availability
46.9K
(https://www.edureka.co/blog/how-to-set-up-hadoop-cluster-with-hdfs-high-availability/)

Browse Categories

Big Data NoSQL (https://www.edureka.co/blog/category/big-data-nosql/) Blockchain (https://www.edureka.co/blog/category/blockchain/)
Business Intelligence (https://www.edureka.co/blog/category/business-intelligence/) Cloud Computing (https://www.edureka.co/blog/category/cloud-computing/)
Cyber Security (https://www.edureka.co/blog/category/cyber-security/) Deep Learning (https://www.edureka.co/blog/category/deep-learning/)
Finance (https://www.edureka.co/blog/category/finance/) Frameworks (https://www.edureka.co/blog/category/frameworks/)
Marketing (https://www.edureka.co/blog/category/marketing/) Mobile Development (https://www.edureka.co/blog/category/mobile-development/)
Operations (https://www.edureka.co/blog/category/operations/) Programming (https://www.edureka.co/blog/category/programming/)
Project Management (https://www.edureka.co/blog/category/project-management/) Robotic Process Automation (https://www.edureka.co/blog/category/robotic-process-automation/)
Success Story (https://www.edureka.co/blog/category/success-story/) Systems & Architecture (https://www.edureka.co/blog/category/systems-architecture/)
Systems Engineering (https://www.edureka.co/blog/category/systems-engineering/) Testing (https://www.edureka.co/blog/category/testing/)

Comments

5 Comments

Santhosh • 3 years ago
Very useful post. Thanks Edureka!
Reply • Share ›

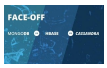
EdurekaSupport Mod ➚ Santhosh • 3 years ago
Glad you found it useful, Santhosh. Feel free to go through other posts as well.
Reply • Share ›

Sagar Morakhia • 4 years ago
nice sample example. :)
Reply • Share ›

EdurekaSupport Mod ➚ Sagar Morakhia • 4 years ago
Thanks Sagar!! Feel free to go through our other blog posts as well.
Reply • Share ›

Related Blogs

(https://www.edureka.co/blog/insights-on-hbase-architecture) Insights on HBase Architecture (https://www.edureka.co/blog/insights-on-hbase-architecture)



(https://www.edureka.co/blog/mongodb-vs-hbase-vs-cassandra/) Face Off: MongoDB Vs HBase Vs Cassandra (https://www.edureka.co/blog/mongodb-vs-hbase-vs-cassandra/)

(https://www.edureka.co/blog/hive-commands-with-examples) Top Hive Commands with Examples in HQL (https://www.edureka.co/blog/hive-commands-with-examples)


(https://www.edureka.co/blog/interview-questions/top-50-hadoop-interview-questions-2016/) Top 50 Hadoop Interview Questions You Must Prepare In 2018 (https://www.edureka.co/blog/interview-questions/top-50-hadoop-interview-questions-2016/)