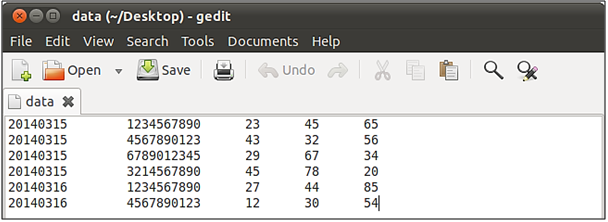
In this blog we shall discuss about a sample Proof of Concept for HBase.

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

[](http://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);

            }

**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);

            }

**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**;

**try** {

                                    // 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:

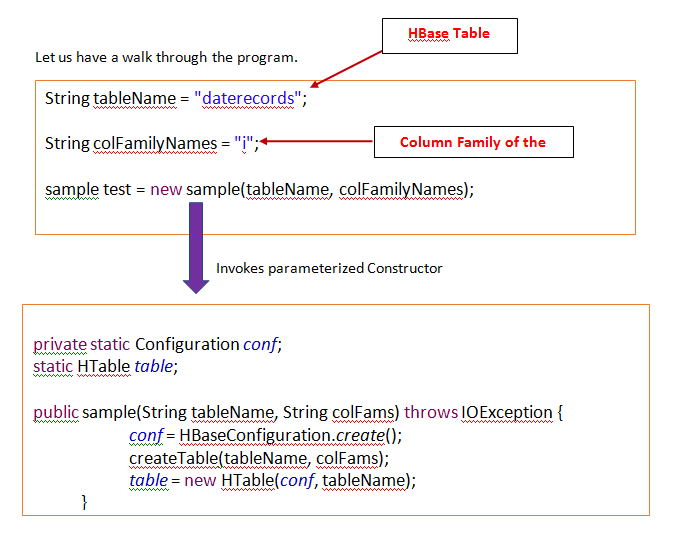
                                    // ex.printStackTrace();

                        }

*getAllRecord*(tableName, “20140315”, “20140316”);

            }

}

[](http://www.edureka.co/blog/sample-hbase-poc/)

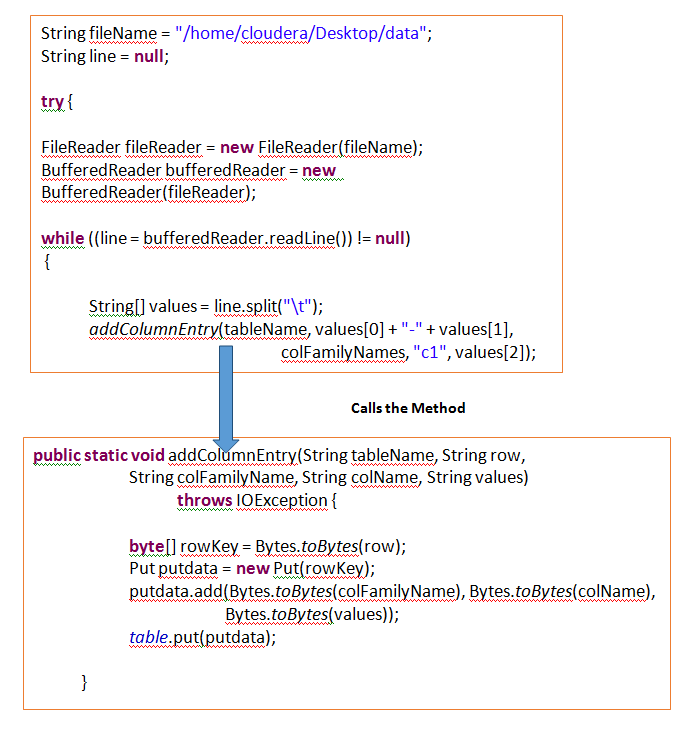
|  |  |
| --- | --- |
|  |  |

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.

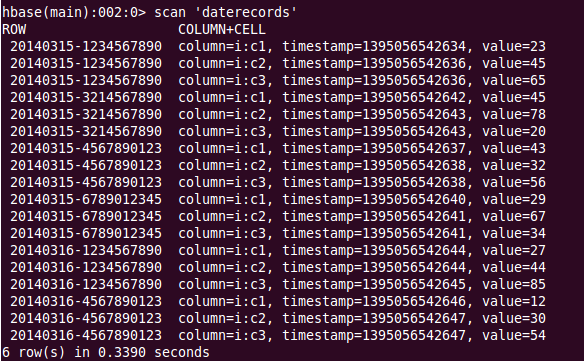
Below is the Java logic.

[](http://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.

[](http://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

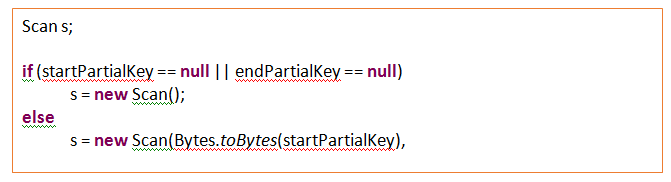
getAllRecord(String tableName, String startPartialKey, String endPartialKey)

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:

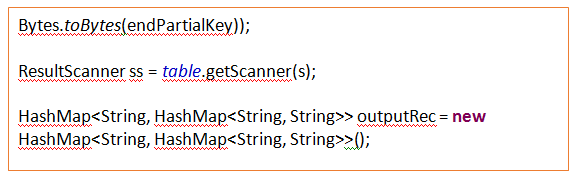
getAllRecord(tableName, “20140315”, “20140316”);

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

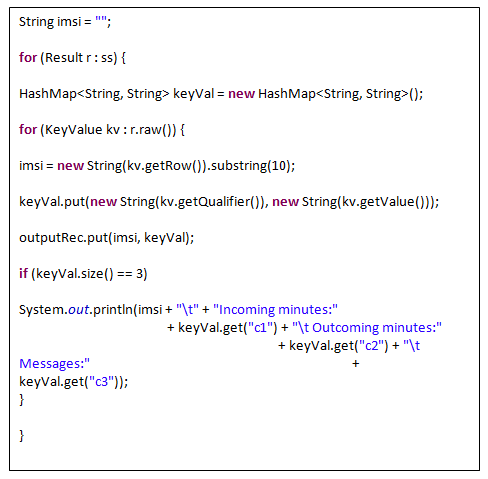
[](http://www.edureka.co/blog/sample-hbase-poc/)

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.

**[](http://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.

[](http://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.

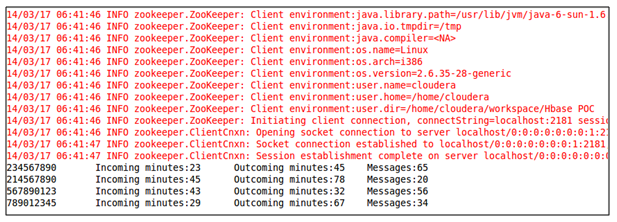
We have given **20140315-1234567890** as the **rowkey** to the Hbase table. In this 20140315 represents the date and 1234567890 represents the Mobile number.

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.

[](http://www.edureka.co/blog/sample-hbase-poc/)

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