**Create a HBase Filter to filter out the customer information where location is AUS**

package advanced;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.HBaseConfiguration;

import org.apache.hadoop.hbase.client.HTable;

import org.apache.hadoop.hbase.client.Result;

import org.apache.hadoop.hbase.client.ResultScanner;

import org.apache.hadoop.hbase.client.Scan;

import org.apache.hadoop.hbase.filter.BinaryComparator;

import org.apache.hadoop.hbase.filter.CompareFilter.CompareOp;

import org.apache.hadoop.hbase.filter.ValueFilter;

import org.apache.hadoop.hbase.util.Bytes;

public class ValueFilterDemo {

public static void main(String[] args) throws IOException, InterruptedException {

Configuration conf = HBaseConfiguration.create();

System.out.println("Creating HTable instance to 'customer'...");

HTable table = new HTable(conf, "customer");

System.out.println("Creating scan object...");

Scan scan = new Scan();

System.out.println("Narrowing down the result to details column family...");

scan.addFamily(Bytes.toBytes("details"));

System.out.println("Adding value filter on scan object...");

scan.setFilter(new ValueFilter(CompareOp.EQUAL, new BinaryComparator(Bytes.toBytes("AUS"))));

System.out.println("Getting a result scanner object...");

ResultScanner rs = table.getScanner(scan);

for (Result r : rs) {

System.out.println("Result: " + r);

}

System.out.println("Closing Scanner instance...");

rs.close();

}

}

**Create a Python program that connects to HBase using Thrift protocol and gets names of all HBase tables**

**--Steps**

1.thrift --gen py Hbase.thrift

2. export PYTHONPATH=$PYTHONPATH:/<path to gen-py>

3. $hbase thrift start

4. from thrift.transport.TSocket import TSocket

from thrift.transport.TTransport import TBufferedTransport

from thrift.protocol import TBinaryProtocol

from hbase import Hbase

5. transport = TBufferedTransport(TSocket('localhost', 9090))

transport.open()

protocol = TBinaryProtocol.TBinaryProtocol(transport)

client = Hbase.Client(protocol)

print(client.getTableNames())

6. transport = TBufferedTransport(TSocket('localhost', 9090))

7. transport.open()

8. protocol = TBinaryProtocol.TBinaryProtocol(transport)

9. client = Hbase.Client(protocol)

10. print(client.getTableNames())