Assignment 19.2

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

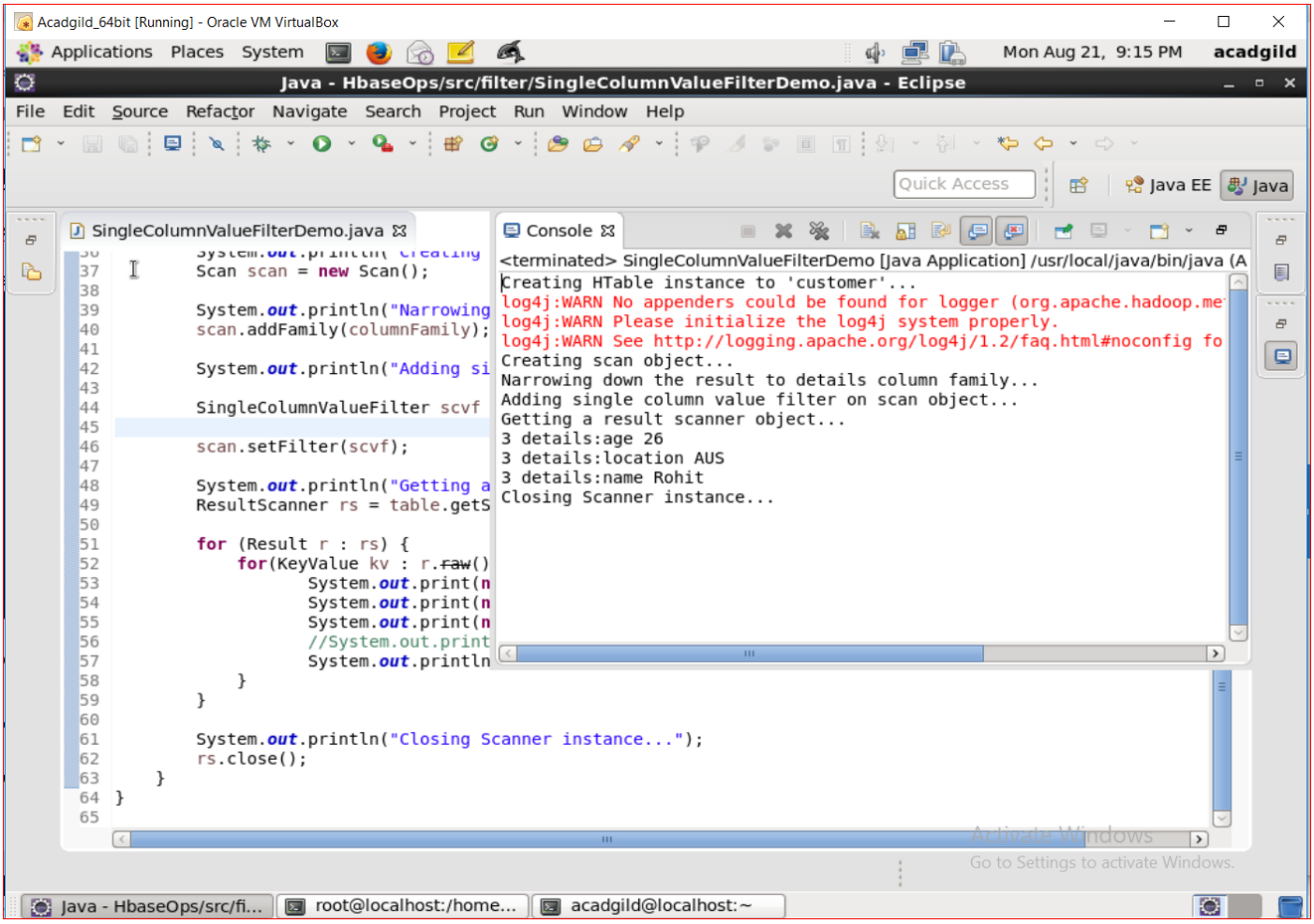
Scan the table throgh hbase shell to see available data in the table.

```
hbase(main):003:0> scan 'customer'
ROW                                   COLUMN+CELL
 1                                    column=details:age, timestamp=1503131028718, value=18
 1                                    column=details:location, timestamp=1503131028718, value=IND
 1                                    column=details:name, timestamp=1503131028718, value=Amit
 2                                    column=details:age, timestamp=1503131028718, value=20
 2                                    column=details:location, timestamp=1503131028718, value=PAK
 2                                    column=details:name, timestamp=1503131028718, value=Sumit
 3                                    column=details:age, timestamp=1503131028718, value=26
 3                                    column=details:location, timestamp=1503131028718, value=AUS
 3                                    column=details:name, timestamp=1503131028718, value=Rohit
 4                                    column=details:age, timestamp=1503131028718, value=24
 4                                    column=details:location, timestamp=1503131028718, value=UK
 4                                    column=details:name, timestamp=1503131028718, value=Namit
4 row(s) in 0.1850 seconds
```

Write a code in SingleColumnValueFilterDemo.java for retrieving the customer information where location is AUS.

SingleColumnValueFilterDemo.java

Run the java code to get the results in Acadgild Sandbox's eclipse.



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

First, we need to install all language specific dependencies on operating system where the Thrift server is started.

Install Python dependencies for centos.

*sudo yum install boost-devel php-devel pcre-devel automake libtool flex bison pkgconfig gcc-c++ boost-devel libevent-devel zlib-devel python-devel ruby-devel libtool**

Download the Thrift server using the below command:

wget https://archive.apache.org/dist/thrift/0.6.0/thrift-0.6.0.tar.gz

Untar the thrift file using the below command and then change the directory to thrift-0.6.1

$ tar xfz thrift-0.6.1.tar.gz

$ cd thrift-0.6.1/

We need to perform configure operation from the thrift-0.6.1 directory.

$./configure

After performing the 'Configure' command we will be able to see what other languages are supported by Thrift.

The configure script is responsible for getting ready to build the software on your specific system. It makes sure all of the dependencies for the rest of the build and install process are available, and finds out whatever it needs to know to use those dependencies

Install Thrift by below commands:

$ sudo make

$ sudo make install

Download Hbase.thrift from link https://drive.google.com/file/d/0B5dejdhAYHztOGt6OGs5ZTZ3WEk/view and execute below command for binding:

thrift --gen py Hbase.thrift

thrift -gen py /path/to/hbase/source/hbase-VERSION/src/main/resources/org/apache/hadoop/hbase/thrift/Hbase.thrift

Create path PYTHONPATH:

export PYTHONPATH=$PYTHONPATH:/home/acadgild/thrift-0.6.0/gen-py/

Start the hbase thrift server:

$hbase thrift start

Create table.py python code file:

```
from thrift.transport.TSocket import TSocket
from thrift.transport.TTransport import TBufferedTransport
from thrift.protocol import TBinaryProtocol
from hbase import Hbase
    #Above commands will import all the required HBase Thrift modules.

    #The below command will create the socket transport and line protocol and allows the Thrift client to connect and talk to
    #the Thrift server.
```

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

transport.open() *#Next we need to open the socket to the Thrift server.*

protocol = TBinaryProtocol.TBinaryProtocol(transport) *#Tbinary is binary implementation of thrift (converting transport to binary implementation)*

client = Hbase.Client(protocol) *#The below lines create the Client object which will be used to interact with HBase. From this client object, you will issue all your Gets and Puts.*

print(client.getTableNames()) *#gives the list of the tables in hbase.*

Run the table.py file by below command:

$python table.py

```
[acadgild@localhost ~]$ python table.py
['clicks', 'customer']
[acadgild@localhost ~]$
```

Hbase Thrift server:

```
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:java.library.path=/usr/local/hadoop-2.6.0/lib/native
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:java.io.tmpdir=/tmp
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:java.compiler=<NA>
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:os.name=Linux
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:os.arch=amd64
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:os.version=2.6.32-573.el6.x86_64
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:user.name=acadgild
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:user.home=/home/acadgild
2017-08-21 22:41:01,714 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Client environment:user.dir=/home/acadgild/dataset
2017-08-21 22:41:01,718 INFO  [thrift-worker-0] zookeeper.ZooKeeper: Initiating client connection, connectString=localhost:2181 sessionTimeout=9000
0 watcher=hconnection-0x151f7a420x0, quorum=localhost:2181, baseZNode=/hbase
2017-08-21 22:41:01,750 INFO  [thrift-worker-0-SendThread(localhost:2181)] zookeeper.ClientCnxn: Opening socket connection to server localhost/127.
0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2017-08-21 22:41:01,758 INFO  [thrift-worker-0-SendThread(localhost:2181)] zookeeper.ClientCnxn: Socket connection established to localhost/127.0.0
.1:2181, initiating session
2017-08-21 22:41:01,765 INFO  [thrift-worker-0-SendThread(localhost:2181)] zookeeper.ClientCnxn: Session establishment complete on server localhost
/127.0.0.1:2181, sessionid = 0x15dfeb0c07e0045, negotiated timeout = 40000
2017-08-21 22:51:07,543 INFO  [ConnectionCleaner] client.HConnectionManager$HConnectionImplementation: Closing master protocol: MasterService
2017-08-21 22:51:07,543 INFO  [ConnectionCleaner] client.HConnectionManager$HConnectionImplementation: Closing zookeeper sessionid=0x15dfeb0c07e004
5
2017-08-21 22:51:07,545 INFO  [ConnectionCleaner] zookeeper.ZooKeeper: Session: 0x15dfeb0c07e0045 closed
2017-08-21 22:51:07,545 INFO  [thrift-worker-0-EventThread] zookeeper.ClientCnxn: EventThread shut down
```