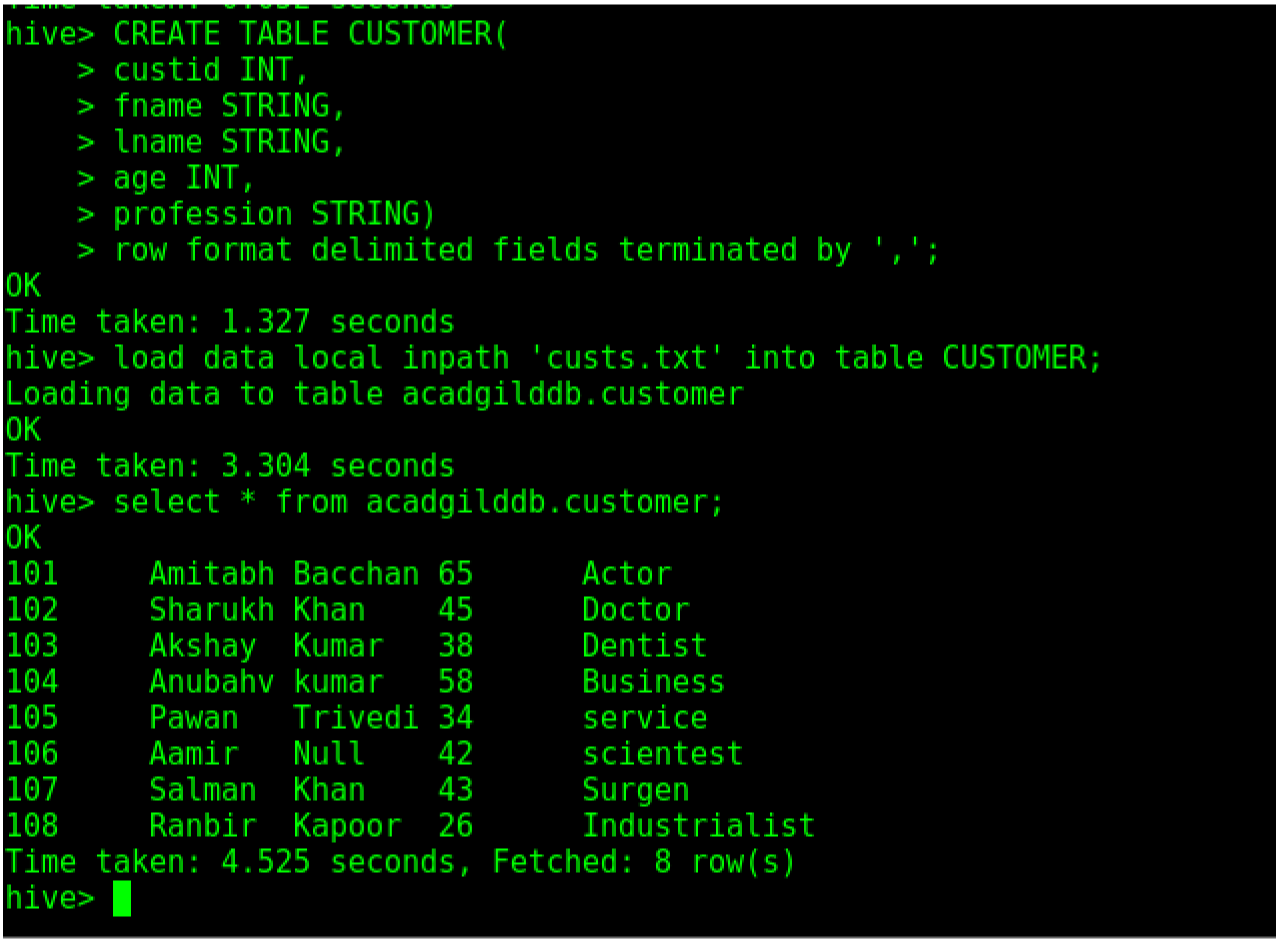
Create an internal table by the name customer as follows :-
CREATE TABLE CUSTOMER(
custid INT,
fname STRING,
lname STRING,
age INT,
profession STRING)
row format delimited fields terminated by ',';

LOAD DATA LOCAL INPATH ‘custs.txt into table CUSTOMER;



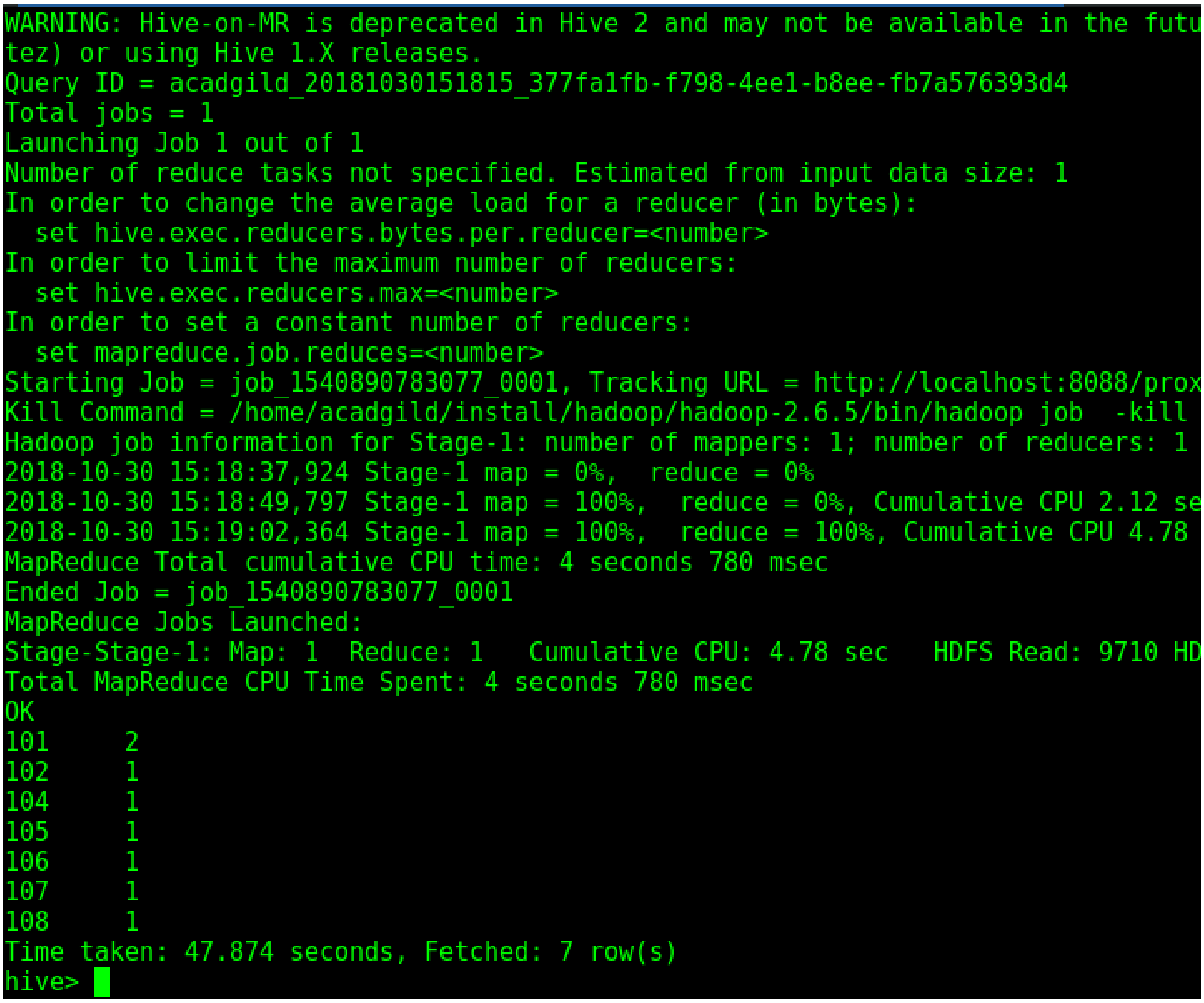
```
hive> CREATE TABLE CUSTOMER(
    > custid INT,
    > fname STRING,
    > lname STRING,
    > age INT,
    > profession STRING)
    > row format delimited fields terminated by ',';
OK
Time taken: 1.327 seconds
hive> load data local inpath 'custs.txt' into table CUSTOMER;
Loading data to table acadgilddb.customer
OK
Time taken: 3.304 seconds
hive> select * from acadgilddb.customer;
OK
101     Amitabh Bacchan 65      Actor
102     Sharukh Khan    45      Doctor
103     Akshay  Kumar   38      Dentist
104     Anubahv kumar   58      Business
105     Pawan   Trivedi 34      service
106     Aamir   Null    42      scientest
107     Salman  Khan    43      Surgen
108     Ranbir  Kapoor  26      Industrialist
Time taken: 4.525 seconds, Fetched: 8 row(s)
hive> 
```

CREATE TABLE TRANSACTIONS (
txnno INT,
txndate STRING,
custno INT,
amount DOUBLE,
category STRING,
product STRING,
city STRING,
state STRING,
spendby STRING)
row format delimited fields terminated by ',';

hive> load data local inpath 'txn.txt' into table TRANSACTIONS;

1. **Find out the number of transaction done by each customer (These should be take up in module 8 itself)**

hive> select custno , count(*) from TRANSACTIONS group by custno;



```
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
tez) or using Hive 1.X releases.
Query ID = acadgild_20181030151815_377fa1fb-f798-4ee1-b8ee-fb7a576393d4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1540890783077_0001, Tracking URL = http://localhost:8088/prox
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-10-30 15:18:37,924 Stage-1 map = 0%,  reduce = 0%
2018-10-30 15:18:49,797 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.12 se
2018-10-30 15:19:02,364 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.78
MapReduce Total cumulative CPU time: 4 seconds 780 msec
Ended Job = job_1540890783077_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.78 sec   HDFS Read: 9710 HD
Total MapReduce CPU Time Spent: 4 seconds 780 msec
OK
101     2
102     1
104     1
105     1
106     1
107     1
108     1
Time taken: 47.874 seconds, Fetched: 7 row(s)
hive>
```

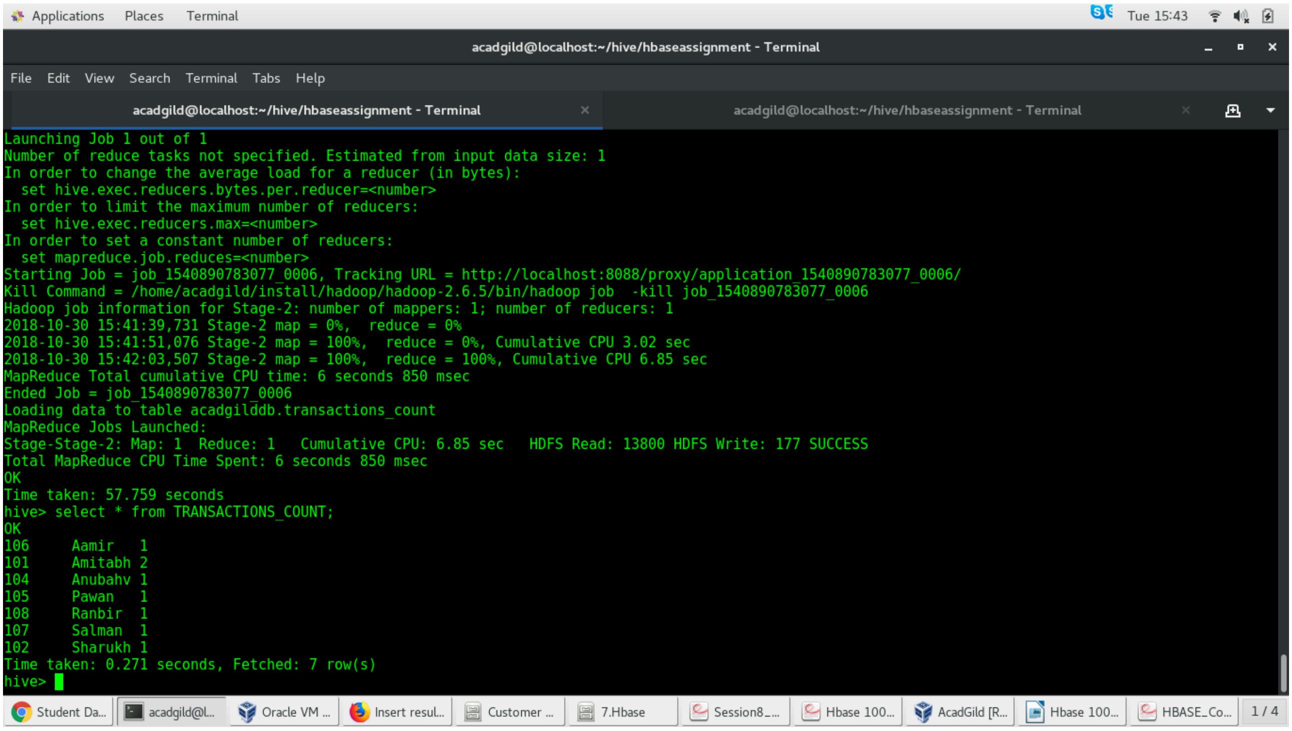
2.
Create a new table called TRANSACTIONS_COUNT. This table should have 3 fields - custid, fname and count.

2 and 4 task done by below command

```
create table TRANSACTIONS_COUNT (custid int,fname STRING, count int)
   > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
   >  with serdeproperties
("hbase.columns.mapping"=":key,personaldetails:fname,personaldetails:count")
   >  tblproperties("hbase.table.name"="TRANSACTIONS");
```

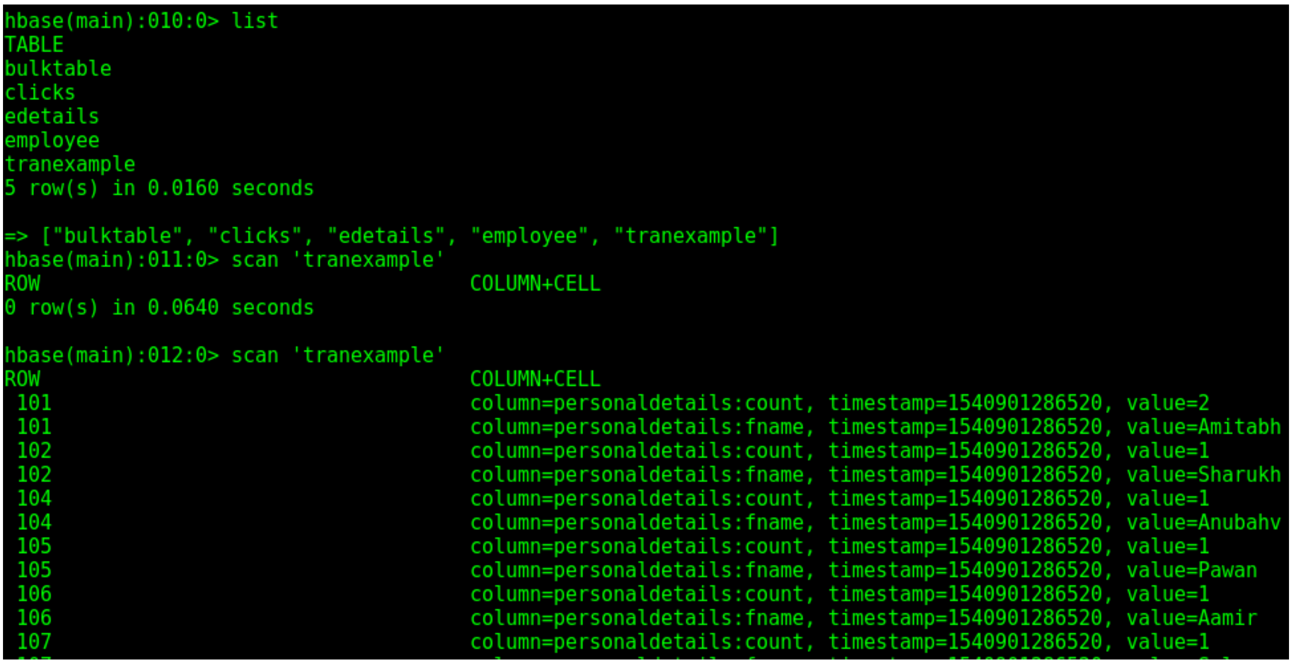
3.
Now write a hive query in such a way that the query populates the data obtained in Step 1 above and populate the table in step 2 above. (This has to be done in module 9).

insert into TRANSACTIONS_COUNT select b.custno,a.fname, count(*) as number_of_transc from customer a join TRANSACTIONS b on a.custid=b.custno group by b.custno,a.fname;



```
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1540890783077_0006, Tracking URL = http://localhost:8088/proxy/application_1540890783077_0006/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1540890783077_0006
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-10-30 15:41:39,731 Stage-2 map = 0%,  reduce = 0%
2018-10-30 15:41:51,076 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.02 sec
2018-10-30 15:42:03,507 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.85 sec
MapReduce Total cumulative CPU time: 6 seconds 850 msec
Ended Job = job_1540890783077_0006
Loading data to table acadgilddb.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.85 sec   HDFS Read: 13800 HDFS Write: 177 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 850 msec
OK
Time taken: 57.759 seconds
hive> select * from TRANSACTIONS_COUNT;
OK
106     Aamir   1
101     Amitabh 2
104     Anubahv 1
105     Pawan   1
108     Ranbir  1
107     Salman  1
102     Sharukh 1
Time taken: 0.271 seconds, Fetched: 7 row(s)
hive>
```

4.
Now lets make the TRANSACTIONS_COUNT table Hbase complaint. In the sence, use Ser Des And Storate handler features of hive to change the TRANSACTIONS_COUNT table to be able to create a TRANSACTIONS table in Hbase.



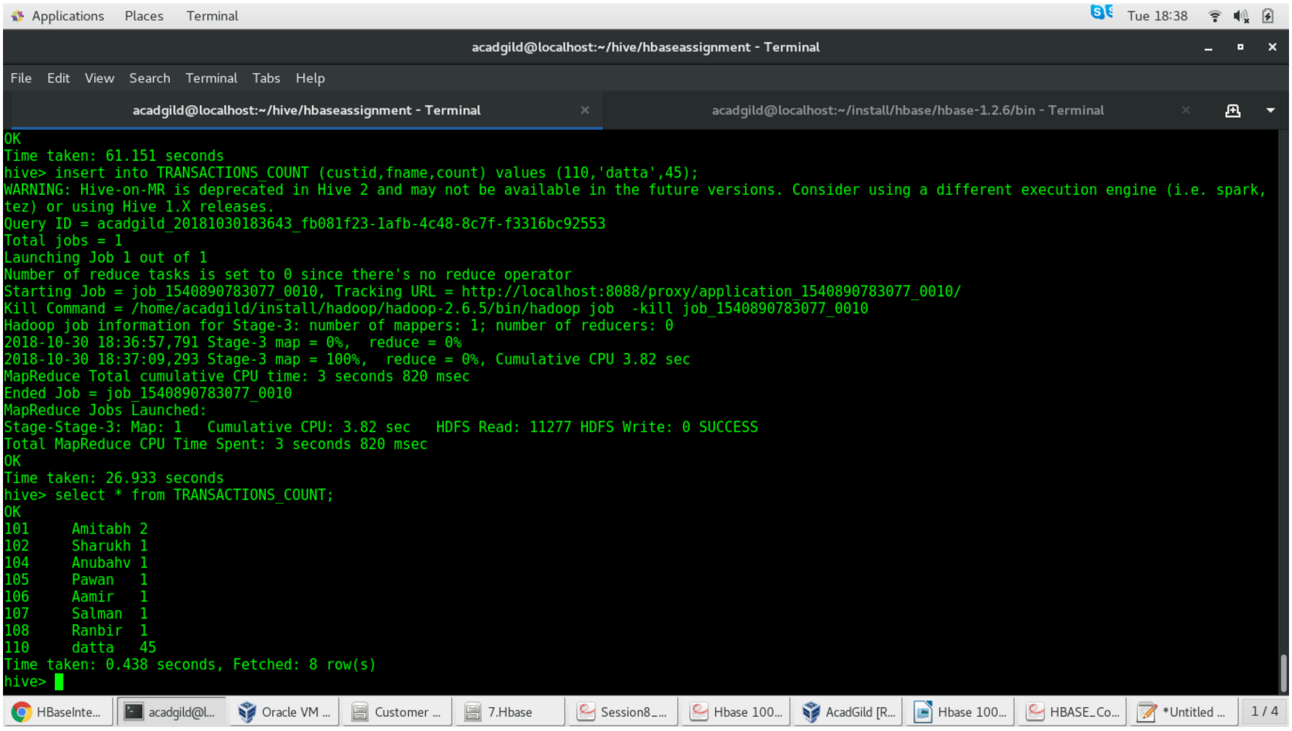
```
hbase(main):010:0> list
TABLE
bulktable
clicks
edetails
employee
tranexample
5 row(s) in 0.0160 seconds

=> ["bulktable", "clicks", "edetails", "employee", "tranexample"]
hbase(main):011:0> scan 'tranexample'
ROW                                    COLUMN+CELL
0 row(s) in 0.0640 seconds

hbase(main):012:0> scan 'tranexample'
ROW                                    COLUMN+CELL
 101                                   column=personaldetails:count, timestamp=1540901286520, value=2
 101                                   column=personaldetails:fname, timestamp=1540901286520, value=Amitabh
 102                                   column=personaldetails:count, timestamp=1540901286520, value=1
 102                                   column=personaldetails:fname, timestamp=1540901286520, value=Sharukh
 104                                   column=personaldetails:count, timestamp=1540901286520, value=1
 104                                   column=personaldetails:fname, timestamp=1540901286520, value=Anubahv
 105                                   column=personaldetails:count, timestamp=1540901286520, value=1
 105                                   column=personaldetails:fname, timestamp=1540901286520, value=Pawan
 106                                   column=personaldetails:count, timestamp=1540901286520, value=1
 106                                   column=personaldetails:fname, timestamp=1540901286520, value=Aamir
 107                                   column=personaldetails:count, timestamp=1540901286520, value=1
```

After loading data in hive , it will reflected into hbase table.

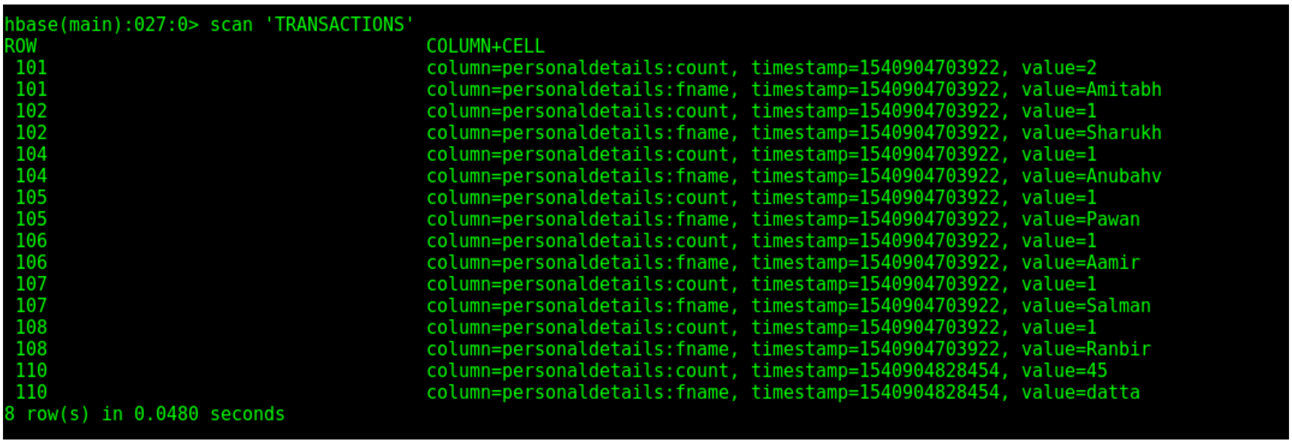
5.
Now insert the data in TRANSACTIONS_COUNT table using the query in step 3 again, this should populate the Hbase TRANSACTIONS table automatically

Insert value in hive table TRANSACTIONS_COUNT

hive> insert into TRANSACTIONS_COUNT (custid,fname,count) values (110,'datta',45);

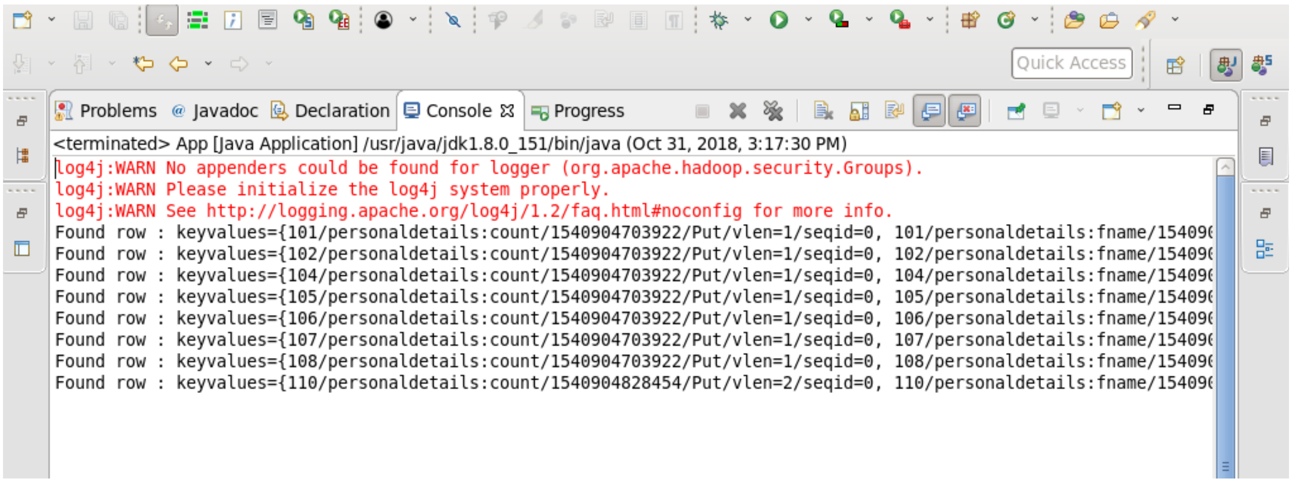

Applications
Places
Terminal
Tue 18:38
acadgild@localhost:~/hive/hbaseassignment - Terminal
File
Edit
View
Search
Terminal
Tabs
Help
acadgild@localhost:~/hive/hbaseassignment - Terminal
acadgild@localhost:~/install/hbase/hbase-1.2.6/bin - Terminal

```
OK
Time taken: 61.151 seconds
hive> insert into TRANSACTIONS_COUNT (custid,fname,count) values (110,'datta',45);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark,
tez) or using Hive 1.X releases.
Query ID = acadgild_20181030183643_fb081f23-1afb-4c48-8c7f-f3316bc92553
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1540890783077_0010, Tracking URL = http://localhost:8088/proxy/application_1540890783077_0010/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1540890783077_0010
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2018-10-30 18:36:57,791 Stage-3 map = 0%,  reduce = 0%
2018-10-30 18:37:09,293 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 3.82 sec
MapReduce Total cumulative CPU time: 3 seconds 820 msec
Ended Job = job_1540890783077_0010
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 3.82 sec   HDFS Read: 11277 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 820 msec
OK
Time taken: 26.933 seconds
hive> select * from TRANSACTIONS_COUNT;
OK
101     Amitabh 2
102     Sharukh 1
104     Anubahv 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
110     datta   45
Time taken: 0.438 seconds, Fetched: 8 row(s)
hive> 
```



```
hbase(main):027:0> scan 'TRANSACTIONS'
ROW                                   COLUMN+CELL
 101                                  column=personaldetails:count, timestamp=1540904703922, value=2
 101                                  column=personaldetails:fname, timestamp=1540904703922, value=Amitabh
 102                                  column=personaldetails:count, timestamp=1540904703922, value=1
 102                                  column=personaldetails:fname, timestamp=1540904703922, value=Sharukh
 104                                  column=personaldetails:count, timestamp=1540904703922, value=1
 104                                  column=personaldetails:fname, timestamp=1540904703922, value=Anubahv
 105                                  column=personaldetails:count, timestamp=1540904703922, value=1
 105                                  column=personaldetails:fname, timestamp=1540904703922, value=Pawan
 106                                  column=personaldetails:count, timestamp=1540904703922, value=1
 106                                  column=personaldetails:fname, timestamp=1540904703922, value=Aamir
 107                                  column=personaldetails:count, timestamp=1540904703922, value=1
 107                                  column=personaldetails:fname, timestamp=1540904703922, value=Salman
 108                                  column=personaldetails:count, timestamp=1540904703922, value=1
 108                                  column=personaldetails:fname, timestamp=1540904703922, value=Ranbir
 110                                  column=personaldetails:count, timestamp=1540904828454, value=45
 110                                  column=personaldetails:fname, timestamp=1540904828454, value=datta
8 row(s) in 0.0480 seconds
```

6.
Now from the Hbase level, write the Hbase java API code to access and scan the TRANSACTIONS table data from java level.



```
<terminated> App [Java Application] /usr/java/jdk1.8.0_151/bin/java (Oct 31, 2018, 3:17:30 PM)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.security.Groups).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Found row : keyvalues={101/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 101/personaldetails:fname/15409(
Found row : keyvalues={102/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 102/personaldetails:fname/15409(
Found row : keyvalues={104/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 104/personaldetails:fname/15409(
Found row : keyvalues={105/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 105/personaldetails:fname/15409(
Found row : keyvalues={106/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 106/personaldetails:fname/15409(
Found row : keyvalues={107/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 107/personaldetails:fname/15409(
Found row : keyvalues={108/personaldetails:count/1540904703922/Put/vlen=1/seqid=0, 108/personaldetails:fname/15409(
Found row : keyvalues={110/personaldetails:count/1540904828454/Put/vlen=2/seqid=0, 110/personaldetails:fname/15409(
```

```
package Acadgild.HbaseDemo;

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.util.Bytes;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args ) throws IOException
    {
        Configuration config = HBaseConfiguration.create();
        HTable hTable = new HTable(config, "TRANSACTIONS");
        Scan scan = new Scan();
        // Scanning the required columns
        scan.addColumn(Bytes.toBytes("personaldetails"), Bytes.toBytes("fname"));
        scan.addColumn(Bytes.toBytes("personaldetails"), Bytes.toBytes("count"));

        // Getting the scan result
        ResultScanner scanner = hTable.getScanner(scan);

        // Reading values from scan result
        for (Result result = scanner.next(); result != null; result = scanner.next())

        System.out.println("Found row : " + result);
        //closing the scanner
        scanner.close();

    }
}
```