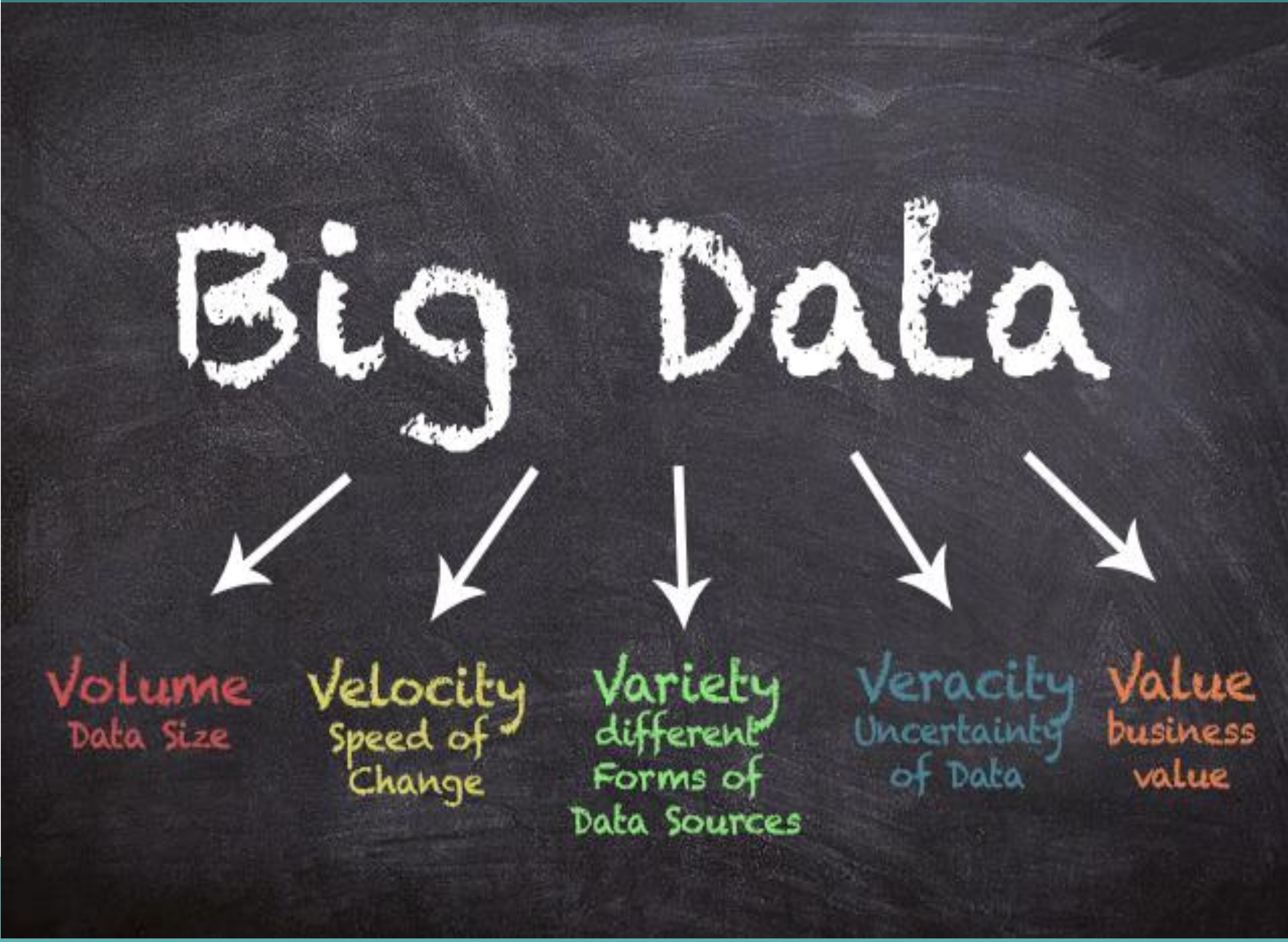
08/11/2018 | By: Ajit K Prasad



Big Data Engineering with Hadoop & Spark

Case Study II Customer & Transaction

Acadgild

Case Study II – Customer & Transaction data using Hive & HBase

This case study assignment is aimed at consolidating the concepts that was learnt during the various session of Hive & HBase of the course.

Problem Statement

Case Study Description

Let us take up the CUSTOMER and TRANSACTIONS table we have created in the "Let's Do Together" section. Let us solve the following use cases using these tables:

1. Find out the number of transactions done by each customer (These should be taken up in module 8 itself)
2. Create a new table called TRANSACTIONS_COUNT. This table should have fields - custid, fname and count. (Again, to be done in module 8)
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).
4. Now let's make the TRANSACTIONS_COUNT table Hbase complaint. In the sense, use SerDes And Storate handler features of hive to change the TRANSACTIONS_COUNT table to be able to create a TRANSACTIONS table in Hbase. (This has to be done in module 10)
5. Now insert the data in TRANSACTIONS_COUNT table using the query in step 3 again, this should populate the Hbase TRANSACTIONS table automatically (This has to be done in module 10)
6. Now from the Hbase level, write the Hbase java API code to access and scan the TRANSACTIONS table data from java level.

To begin with, a database was created, which was used and the 2 tables required were created:

Create a database ‘CaseStudyII’

hive> create database casestudy2;

Use the created database to further create tables and perform query functions

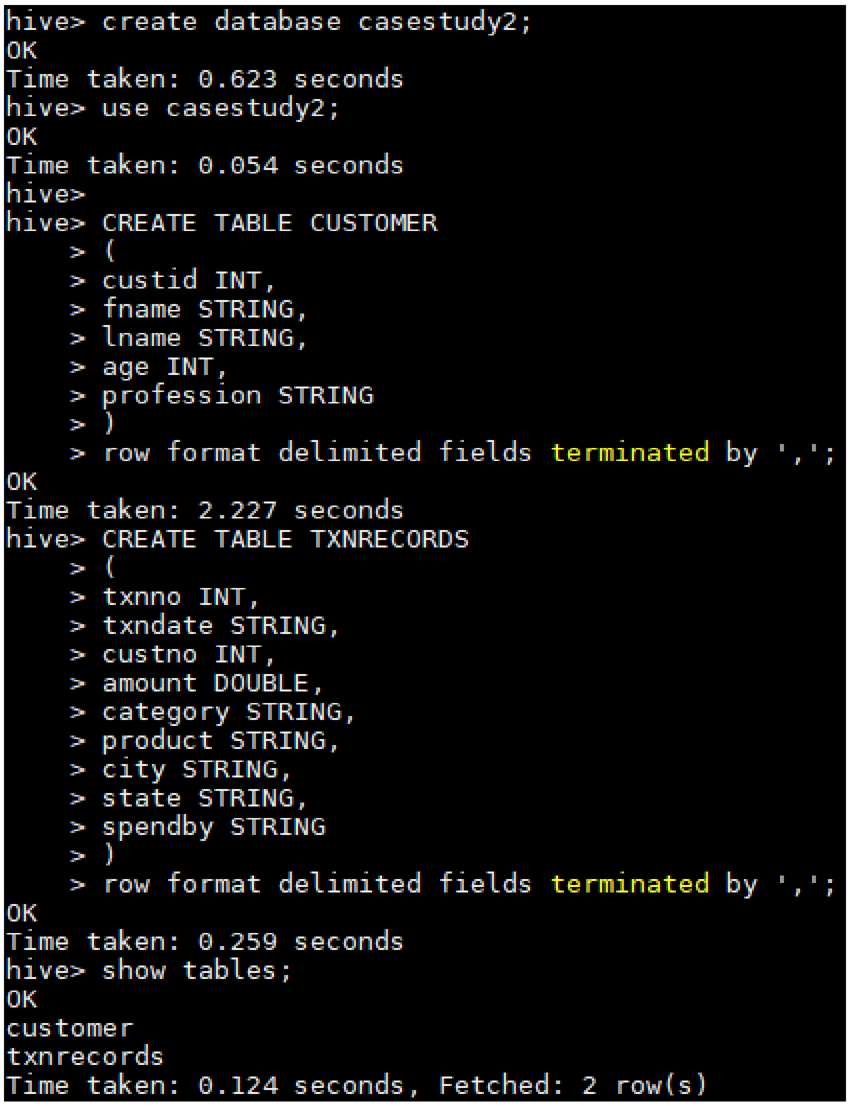
hive> use casestudy2;

Create table ‘CUSTOMER’

hive> CREATE TABLE CUSTOMER (custid INT, fname STRING, lname STRING, age INT, profession STRING) row format delimited fields terminated by ',';

Create table ‘TXNRECORDS’

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

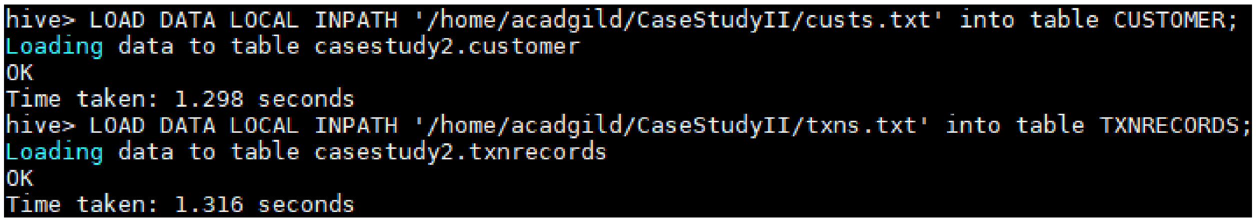


```
hive> create database casestudy2;
OK
Time taken: 0.623 seconds
hive> use casestudy2;
OK
Time taken: 0.054 seconds
hive>
hive> CREATE TABLE CUSTOMER
    > (
    > custid INT,
    > fname STRING,
    > lname STRING,
    > age INT,
    > profession STRING
    > )
    > row format delimited fields terminated by ',';
OK
Time taken: 2.227 seconds
hive> CREATE TABLE TXNRECORDS
    > (
    > txnno INT,
    > txndate STRING,
    > custno INT,
    > amount DOUBLE,
    > category STRING,
    > product STRING,
    > city STRING,
    > state STRING,
    > spendby STRING
    > )
    > row format delimited fields terminated by ',';
OK
Time taken: 0.259 seconds
hive> show tables;
OK
customer
txnrecords
Time taken: 0.124 seconds, Fetched: 2 row(s)
```

Load the data in the two tables created

hive> LOAD DATA LOCAL INPATH '/home/acadgild/CaseStudyII/custs.txt' into table CUSTOMER;

hive> LOAD DATA LOCAL INPATH '/home/acadgild/CaseStudyII/txns.txt' into table TXNRECORDS;



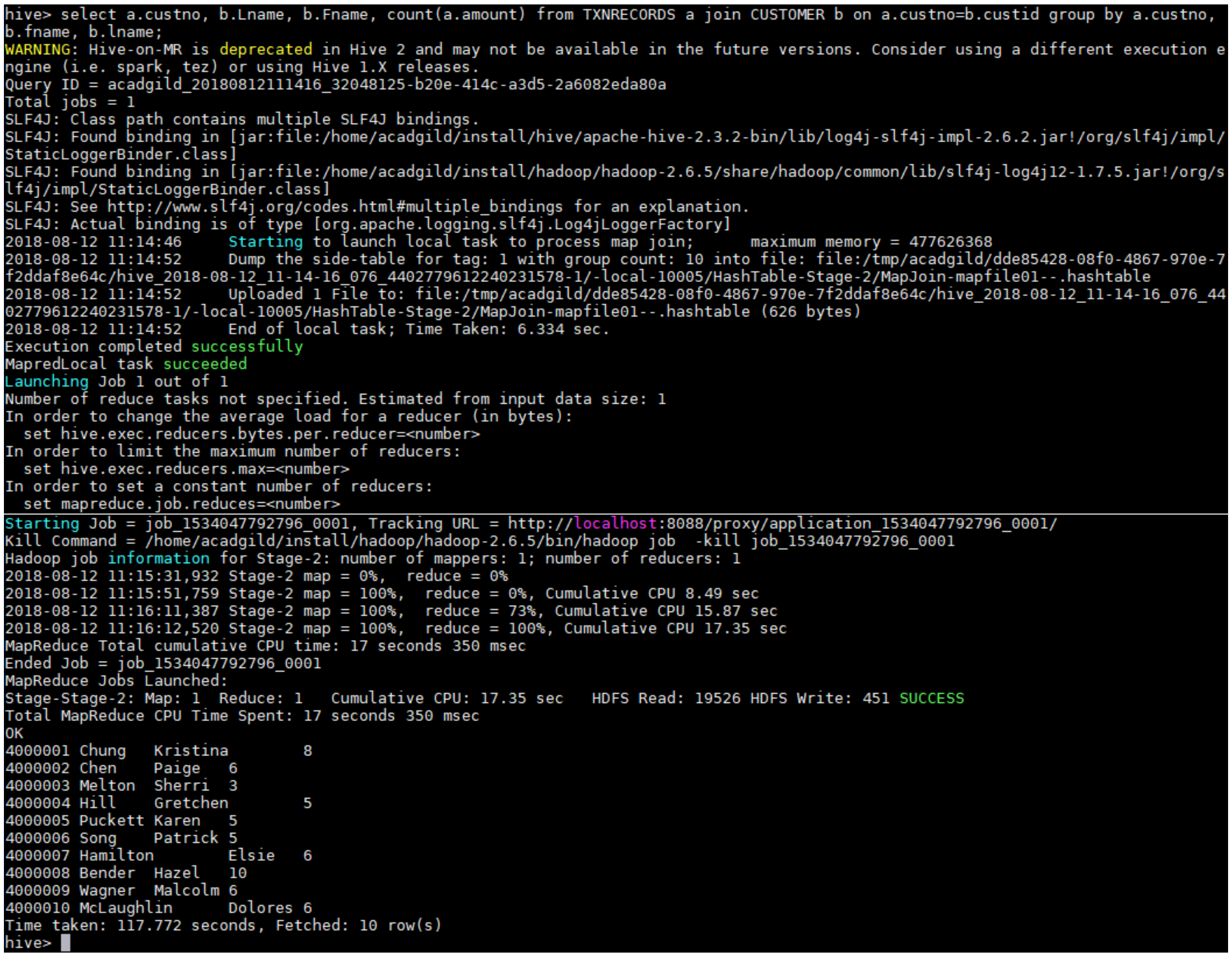
```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/CaseStudyII/custs.txt' into table CUSTOMER;
Loading data to table casestudy2.customer
OK
Time taken: 1.298 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/CaseStudyII/txns.txt' into table TXNRECORDS;
Loading data to table casestudy2.txnrecords
OK
Time taken: 1.316 seconds
```

Once the tables were ready, task were performed using them

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

Solution:

hive> select a.custno, b.lname, b.fname, count(a.amount) from TXNRECORDS a join CUSTOMER b on a.custno=b.custid group by a.custno, b.fname, b.lname;

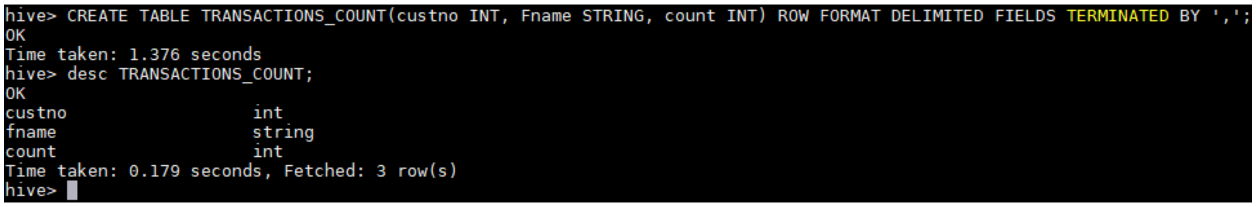


```
hive> select a.custno, b.Lname, b.Fname, count(a.amount) from TXNRECORDS a join CUSTOMER b on a.custno=b.custid group by a.custno,
b.fname, b.lname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180812111416_32048125-b20e-414c-a3d5-2a6082eda80a
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/s
lf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2018-08-12 11:14:46     Starting to launch local task to process map join;      maximum memory = 477626368
2018-08-12 11:14:52     Dump the side-table for tag: 1 with group count: 10 into file: file:/tmp/acadgild/dde85428-08f0-4867-970e-7
f2ddaf8e64c/hive_2018-08-12_11-14-16_076_4402779612240231578-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2018-08-12 11:14:52     Uploaded 1 File to: file:/tmp/acadgild/dde85428-08f0-4867-970e-7f2ddaf8e64c/hive_2018-08-12_11-14-16_076_44
02779612240231578-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (626 bytes)
2018-08-12 11:14:52     End of local task; Time Taken: 6.334 sec.
Execution completed successfully
MapredLocal task succeeded
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_1534047792796_0001, Tracking URL = http://localhost:8088/proxy/application_1534047792796_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534047792796_0001
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-12 11:15:31,932 Stage-2 map = 0%,  reduce = 0%
2018-08-12 11:15:51,759 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 8.49 sec
2018-08-12 11:16:11,387 Stage-2 map = 100%,  reduce = 73%, Cumulative CPU 15.87 sec
2018-08-12 11:16:12,520 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 17.35 sec
MapReduce Total cumulative CPU time: 17 seconds 350 msec
Ended Job = job_1534047792796_0001
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 17.35 sec   HDFS Read: 19526 HDFS Write: 451 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 350 msec
OK
4000001 Chung   Kristina        8
4000002 Chen    Paige   6
4000003 Melton  Sherri  3
4000004 Hill    Gretchen        5
4000005 Puckett Karen   5
4000006 Song    Patrick 5
4000007 Hamilton        Elsie   6
4000008 Bender  Hazel   10
4000009 Wagner  Malcolm 6
4000010 McLaughlin      Dolores 6
Time taken: 117.772 seconds, Fetched: 10 row(s)
hive> 
```

2. Create a new table called TRANSACTIONS_COUNT. This table should have fields - custid, fname and count. (Again, to be done in module 8)

Solution:

hive> CREATE TABLE TRANSACTIONS_COUNT(custno INT, fname STRING, count INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

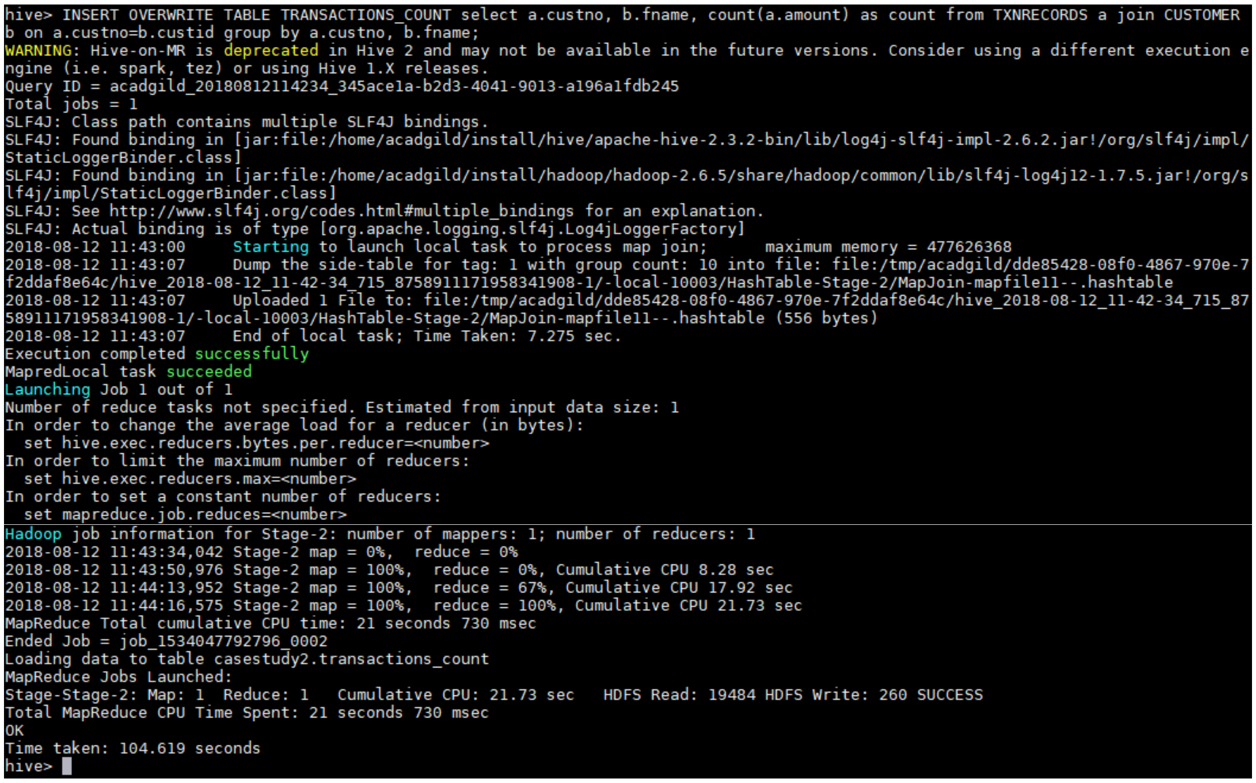


```
hive> CREATE TABLE TRANSACTIONS_COUNT(custno INT, Fname STRING, count INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 1.376 seconds
hive> desc TRANSACTIONS_COUNT;
OK
custno                  int
fname                   string
count                   int
Time taken: 0.179 seconds, Fetched: 3 row(s)
hive>
```

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).

Solution:

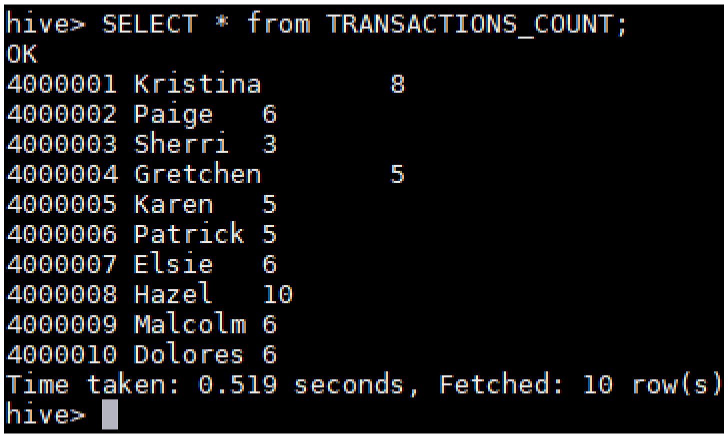
hive> INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.custno, b.fname, count(a.amount) as count from TXNRECORDS a join CUSTOMER b on a.custno=b.custid group by a.custno, b.fname;



```
hive> INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.custno, b.fname, count(a.amount) as count from TXNRECORDS a join CUSTOMER
b on a.custno=b.custid group by a.custno, b.fname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180812114234_345ace1a-b2d3-4041-9013-a196a1fdb245
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/s
lf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2018-08-12 11:43:00     Starting to launch local task to process map join;      maximum memory = 477626368
2018-08-12 11:43:07     Dump the side-table for tag: 1 with group count: 10 into file: file:/tmp/acadgild/dde85428-08f0-4867-970e-7
f2ddaf8e64c/hive_2018-08-12_11-42-34_715_8758911171958341908-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2018-08-12 11:43:07     Uploaded 1 File to: file:/tmp/acadgild/dde85428-08f0-4867-970e-7f2ddaf8e64c/hive_2018-08-12_11-42-34_715_87
58911171958341908-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (556 bytes)
2018-08-12 11:43:07     End of local task; Time Taken: 7.275 sec.
Execution completed successfully
MapredLocal task succeeded
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>
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-12 11:43:34,042 Stage-2 map = 0%,  reduce = 0%
2018-08-12 11:43:50,976 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 8.28 sec
2018-08-12 11:44:13,952 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 17.92 sec
2018-08-12 11:44:16,575 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 21.73 sec
MapReduce Total cumulative CPU time: 21 seconds 730 msec
Ended Job = job_1534047792796_0002
Loading data to table casestudy2.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 21.73 sec   HDFS Read: 19484 HDFS Write: 260 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 730 msec
OK
Time taken: 104.619 seconds
hive>
```

Check if the data was populated as required

*hive> SELECT * from TRANSACTIONS_COUNT;*



```
hive> SELECT * from TRANSACTIONS_COUNT;
OK
4000001 Kristina        8
4000002 Paige   6
4000003 Sherri  3
4000004 Gretchen        5
4000005 Karen   5
4000006 Patrick 5
4000007 Elsie   6
4000008 Hazel   10
4000009 Malcolm 6
4000010 Dolores 6
Time taken: 0.519 seconds, Fetched: 10 row(s)
hive>
```

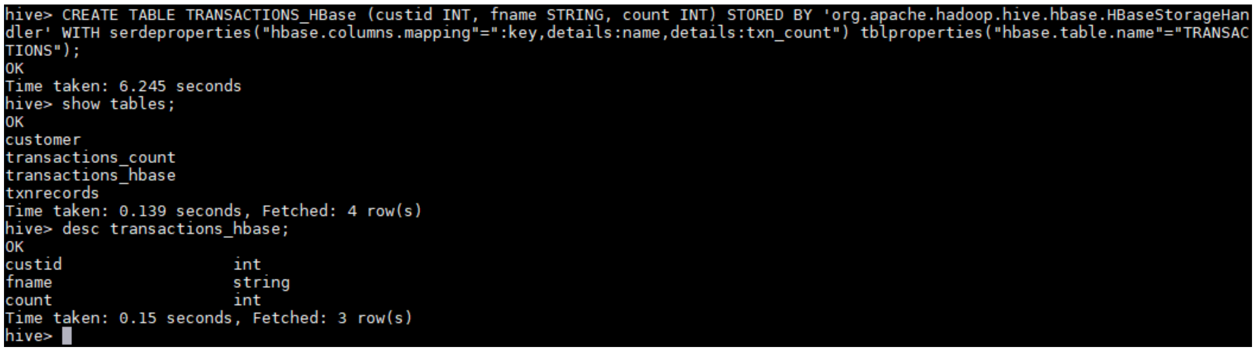
4. Now let's make the TRANSACTIONS_COUNT table HBase complaint. In the sense, use SerDes And Storate handler features of hive to change the TRANSACTIONS_COUNT table to be able to create a TRANSACTIONS table in Hbase. (This has to be done in module 10)

Solution:

hive> CREATE TABLE TRANSACTIONS_HBase (custid INT, fname STRING, count INT) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH serdeproperties("hbase.columns.mapping"=":key,details:name,details:txn_count") tblproperties("hbase.table.name"="TRANSACTIONS");

hive> show tables;

hive> desc TRANSACTIONS_HBase;



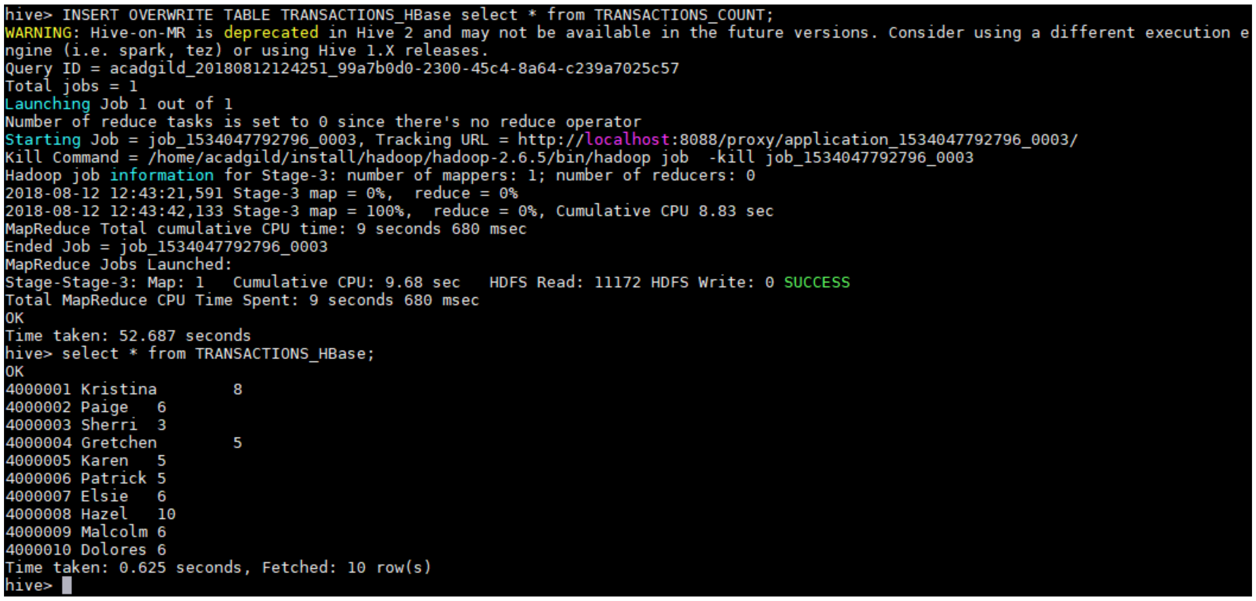
```
hive> CREATE TABLE TRANSACTIONS_HBase (custid INT, fname STRING, count INT) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHan
dler' WITH serdeproperties("hbase.columns.mapping"=":key,details:name,details:txn_count") tblproperties("hbase.table.name"="TRANSAC
TIONS");
OK
Time taken: 6.245 seconds
hive> show tables;
OK
customer
transactions_count
transactions_hbase
txnrecords
Time taken: 0.139 seconds, Fetched: 4 row(s)
hive> desc transactions_hbase;
OK
custid                  int
fname                   string
count                   int
Time taken: 0.15 seconds, Fetched: 3 row(s)
hive>
```

5. Now insert the data in TRANSACTIONS_COUNT table using the query in step 3 again, this should populate the Hbase TRANSACTIONS table automatically (This has to be done in module 10)

Solution:

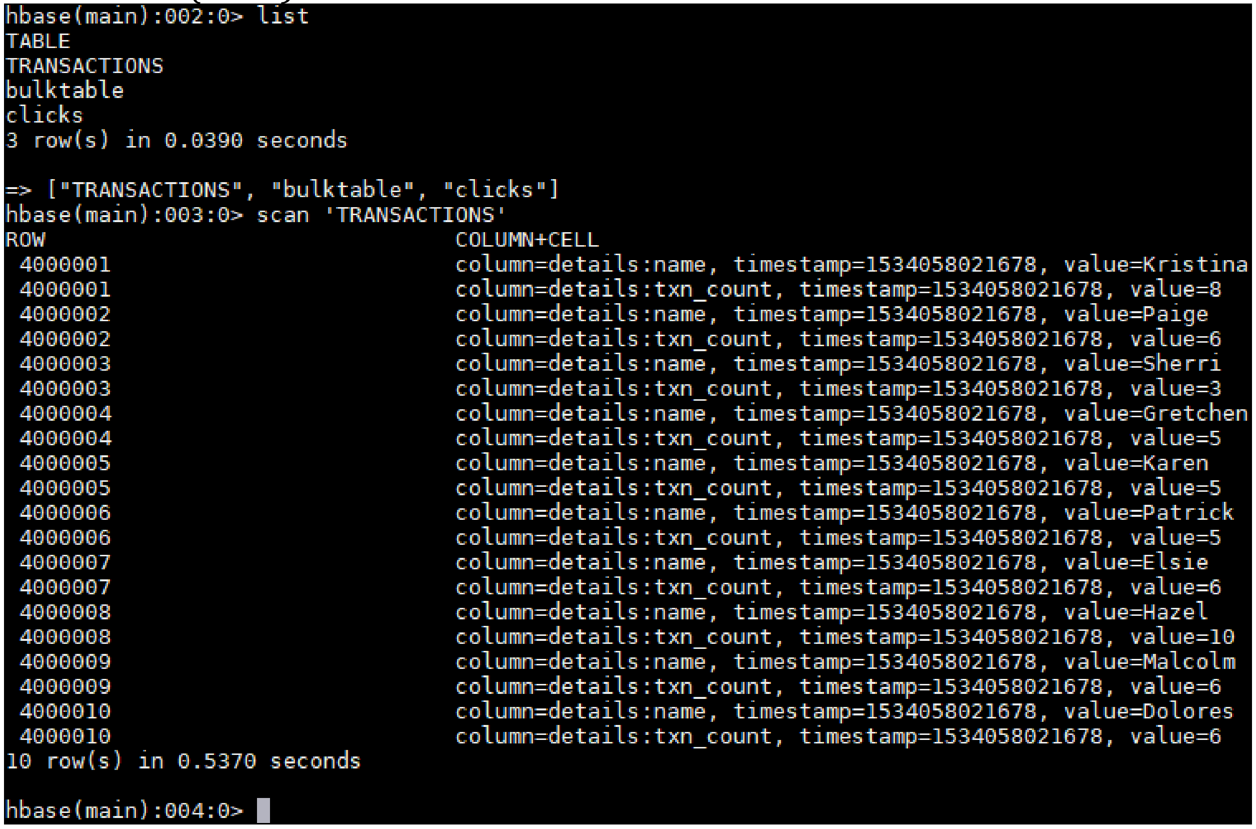
*hive> INSERT OVERWRITE TABLE TRANSACTIONS_HBase select * from TRANSACTIONS_COUNT;*

*hive> select * from TRANSACTIONS_HBase;*



```
hive> INSERT OVERWRITE TABLE TRANSACTIONS_HBase select * from TRANSACTIONS_COUNT;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution e
ngine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180812124251_99a7b0d0-2300-45c4-8a64-c239a7025c57
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_1534047792796_0003, Tracking URL = http://localhost:8088/proxy/application_1534047792796_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534047792796_0003
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2018-08-12 12:43:21,591 Stage-3 map = 0%,  reduce = 0%
2018-08-12 12:43:42,133 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 8.83 sec
MapReduce Total cumulative CPU time: 9 seconds 680 msec
Ended Job = job_1534047792796_0003
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 9.68 sec   HDFS Read: 11172 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 680 msec
OK
Time taken: 52.687 seconds
hive> select * from TRANSACTIONS_HBase;
OK
4000001 Kristina        8
4000002 Paige   6
4000003 Sherri  3
4000004 Gretchen        5
4000005 Karen   5
4000006 Patrick 5
4000007 Elsie   6
4000008 Hazel   10
4000009 Malcolm 6
4000010 Dolores 6
Time taken: 0.625 seconds, Fetched: 10 row(s)
hive>
```

hbase(main):003:0> scan 'TRANSACTIONS'

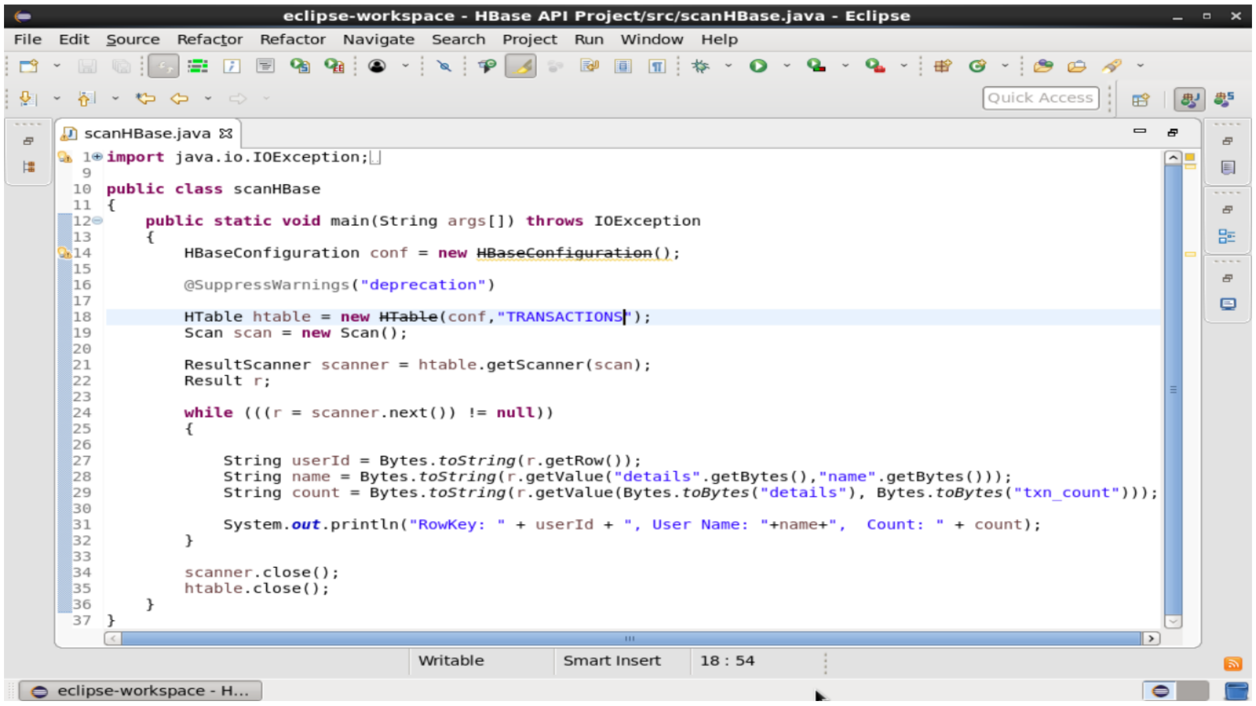


```
hbase(main):002:0> list
TABLE
TRANSACTIONS
bulktable
clicks
3 row(s) in 0.0390 seconds

=> ["TRANSACTIONS", "bulktable", "clicks"]
hbase(main):003:0> scan 'TRANSACTIONS'
ROW                              COLUMN+CELL
 4000001                         column=details:name, timestamp=1534058021678, value=Kristina
 4000001                         column=details:txn_count, timestamp=1534058021678, value=8
 4000002                         column=details:name, timestamp=1534058021678, value=Paige
 4000002                         column=details:txn_count, timestamp=1534058021678, value=6
 4000003                         column=details:name, timestamp=1534058021678, value=Sherri
 4000003                         column=details:txn_count, timestamp=1534058021678, value=3
 4000004                         column=details:name, timestamp=1534058021678, value=Gretchen
 4000004                         column=details:txn_count, timestamp=1534058021678, value=5
 4000005                         column=details:name, timestamp=1534058021678, value=Karen
 4000005                         column=details:txn_count, timestamp=1534058021678, value=5
 4000006                         column=details:name, timestamp=1534058021678, value=Patrick
 4000006                         column=details:txn_count, timestamp=1534058021678, value=5
 4000007                         column=details:name, timestamp=1534058021678, value=Elsie
 4000007                         column=details:txn_count, timestamp=1534058021678, value=6
 4000008                         column=details:name, timestamp=1534058021678, value=Hazel
 4000008                         column=details:txn_count, timestamp=1534058021678, value=10
 4000009                         column=details:name, timestamp=1534058021678, value=Malcolm
 4000009                         column=details:txn_count, timestamp=1534058021678, value=6
 4000010                         column=details:name, timestamp=1534058021678, value=Dolores
 4000010                         column=details:txn_count, timestamp=1534058021678, value=6
10 row(s) in 0.5370 seconds

hbase(main):004:0>
```

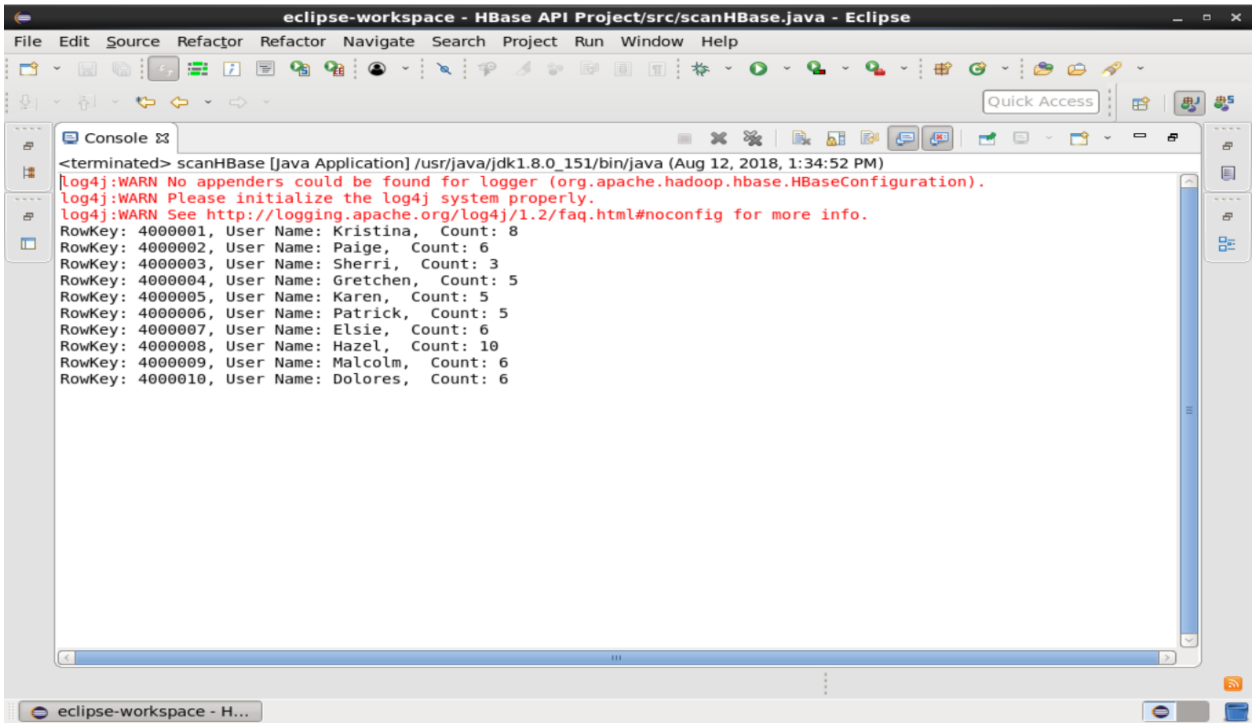
6. Now from the Hbase level, write the Hbase java API code to access and scan the TRANSACTIONS table data from java level.

Solution:

Access and Scan of “TRANSACTIONS” table data using Java API code



```java
import java.io.IOException;

public class scanHBase
{
	public static void main(String args[]) throws IOException
	{
		HBaseConfiguration conf = new ~~HBaseConfiguration~~();

		@SuppressWarnings("deprecation")

		HTable htable = new ~~HTable~~(conf,"TRANSACTIONS");
		Scan scan = new Scan();

		ResultScanner scanner = htable.getScanner(scan);
		Result r;

		while (((r = scanner.next()) != null))
		{

			String userId = Bytes.toString(r.getRow());
			String name = Bytes.toString(r.getValue("details".getBytes(),"name".getBytes()));
			String count = Bytes.toString(r.getValue(Bytes.toBytes("details"), Bytes.toBytes("txn_count")));

			System.out.println("RowKey: " + userId + ", User Name: "+name+",  Count: " + count);
		}

		scanner.close();
		htable.close();
	}
}
```



```
<terminated> scanHBase [Java Application] /usr/java/jdk1.8.0_151/bin/java (Aug 12, 2018, 1:34:52 PM)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.hbase.HBaseConfiguration).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
RowKey: 4000001, User Name: Kristina,  Count: 8
RowKey: 4000002, User Name: Paige,  Count: 6
RowKey: 4000003, User Name: Sherri,  Count: 3
RowKey: 4000004, User Name: Gretchen,  Count: 5
RowKey: 4000005, User Name: Karen,  Count: 5
RowKey: 4000006, User Name: Patrick,  Count: 5
RowKey: 4000007, User Name: Elsie,  Count: 6
RowKey: 4000008, User Name: Hazel,  Count: 10
RowKey: 4000009, User Name: Malcolm,  Count: 6
RowKey: 4000010, User Name: Dolores,  Count: 6
```