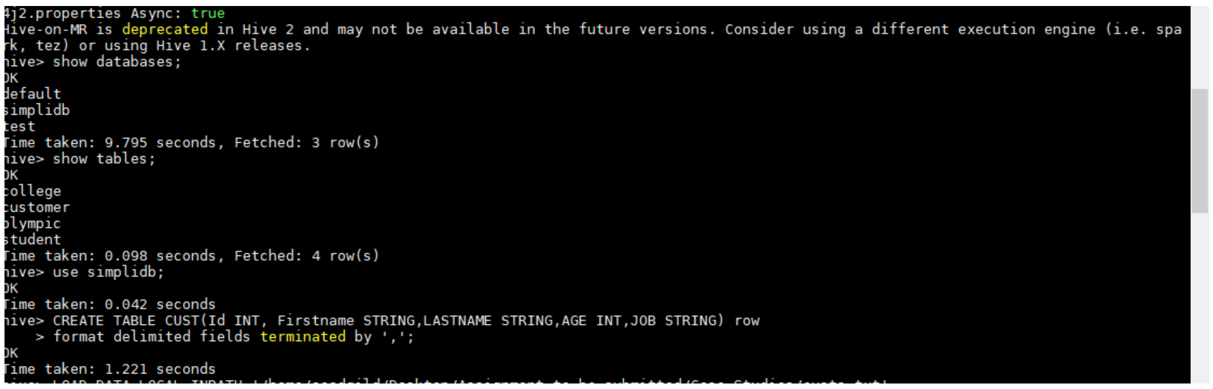
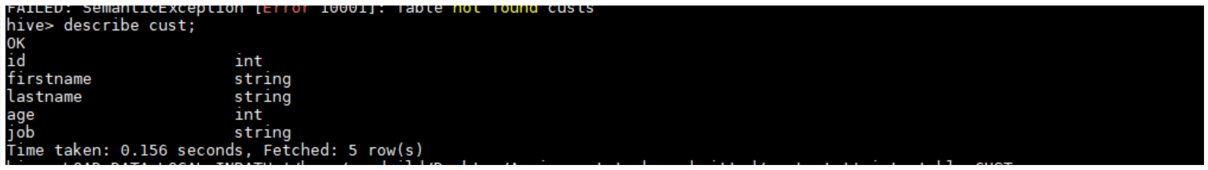
Case Study Description

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 :-

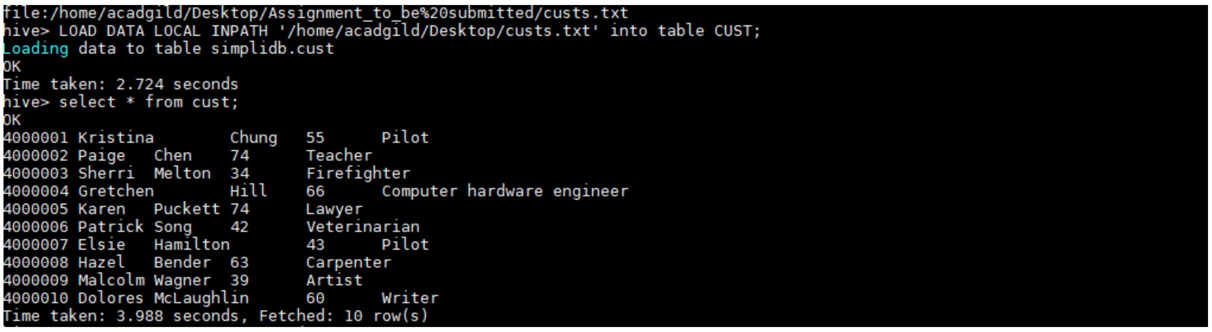
1. Find out the number of transaction done by each customer (These should be take up in module 8 itself)
2. Create a new table called TRANSACTIONS_COUNT. This table should have 3 fields - custid, fname and count. (Again to be done in module 8)
3. Now write a hive query in such a way that the query populates 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 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. (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.



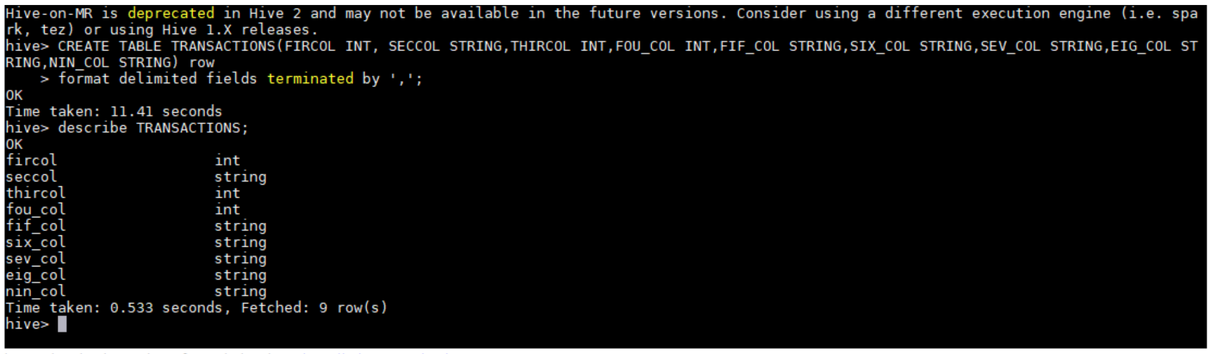
```
4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spa
rk, tez) or using Hive 1.X releases.
hive> show databases;
OK
default
simplidb
test
Time taken: 9.795 seconds, Fetched: 3 row(s)
hive> show tables;
OK
college
customer
olympic
student
Time taken: 0.098 seconds, Fetched: 4 row(s)
hive> use simplidb;
OK
Time taken: 0.042 seconds
hive> CREATE TABLE CUST(Id INT, Firstname STRING,LASTNAME STRING,AGE INT,JOB STRING) row
    > format delimited fields terminated by ',';
OK
Time taken: 1.221 seconds
```



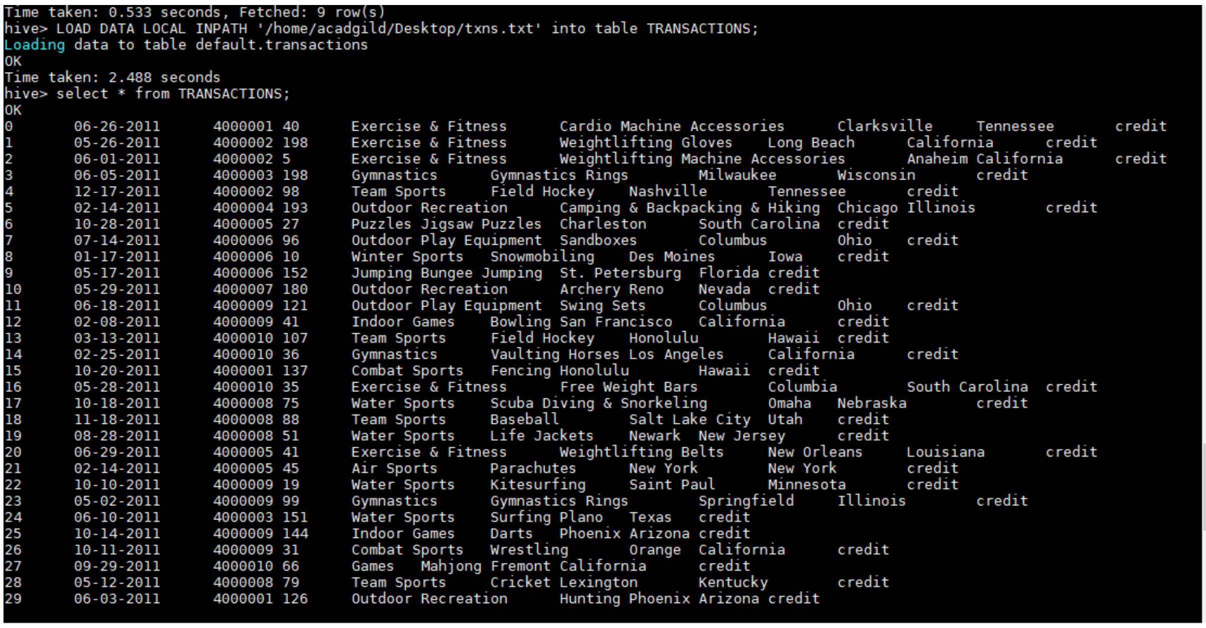
```
FAILED: SemanticException [Error 10001]: Table not found custs
hive> describe cust;
OK
id                          int
firstname                   string
lastname                    string
age                         int
job                         string
Time taken: 0.156 seconds, Fetched: 5 row(s)
```



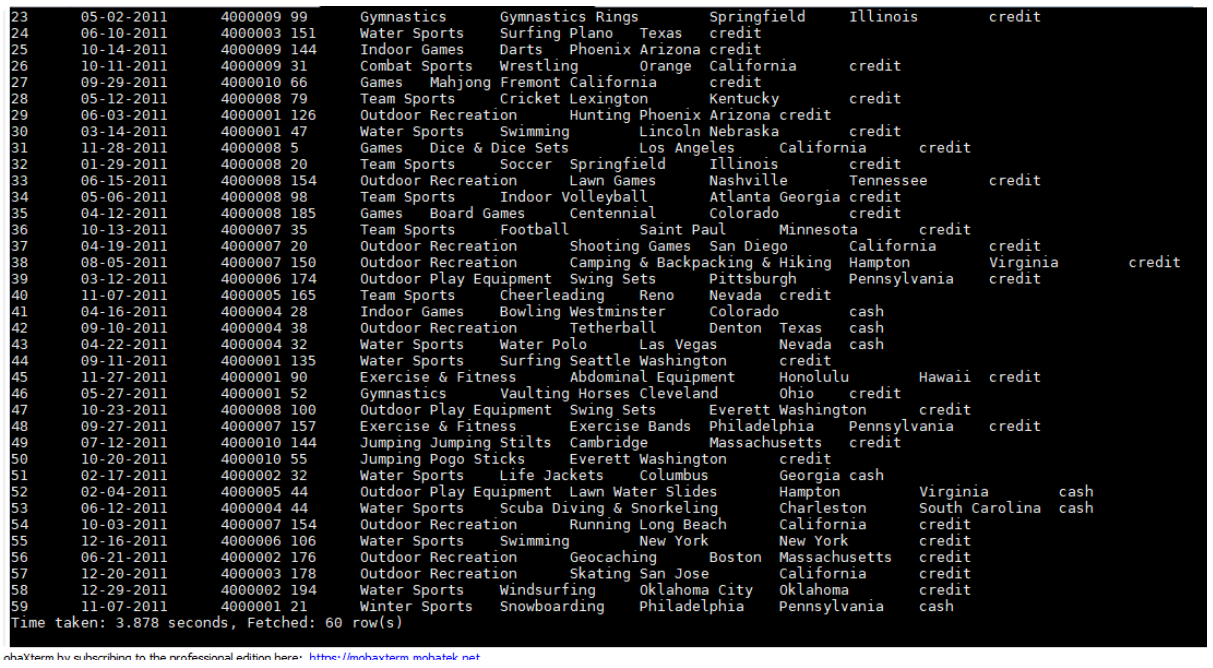
```
file:/home/acadgild/Desktop/Assignment_to_be%20submitted/custs.txt
hive> LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/custs.txt' into table CUST;
Loading data to table simplidb.cust
OK
Time taken: 2.724 seconds
hive> select * from cust;
OK
4000001 Kristina        Chung   55      Pilot
4000002 Paige   Chen    74      Teacher
4000003 Sherri  Melton  34      Firefighter
4000004 Gretchen        Hill    66      Computer hardware engineer
4000005 Karen   Puckett 74      Lawyer
4000006 Patrick Song    42      Veterinarian
4000007 Elsie   Hamilton        43      Pilot
4000008 Hazel   Bender  63      Carpenter
4000009 Malcolm Wagner  39      Artist
4000010 Dolores McLaughlin      60      Writer
Time taken: 3.988 seconds, Fetched: 10 row(s)
```



```
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. spa
rk, tez) or using Hive 1.X releases.
hive> CREATE TABLE TRANSACTIONS(FIRCOL INT, SECCOL STRING,THIRCOL INT,FOU_COL INT,FIF_COL STRING,SIX_COL STRING,SEV_COL STRING,EIG_COL ST
RING,NIN_COL STRING) row
    > format delimited fields terminated by ',';
OK
Time taken: 11.41 seconds
hive> describe TRANSACTIONS;
OK
fircol                  int
seccol                  string
thircol                 int
fou_col                 int
fif_col                 string
six_col                 string
sev_col                 string
eig_col                 string
nin_col                 string
Time taken: 0.533 seconds, Fetched: 9 row(s)
hive> █
```



```
Time taken: 0.533 seconds, Fetched: 9 row(s)
hive> LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/txns.txt' into table TRANSACTIONS;
Loading data to table default.transactions
OK
Time taken: 2.488 seconds
hive> select * from TRANSACTIONS;
OK
0       06-26-2011      4000001 40      Exercise & Fitness      Cardio Machine Accessories      Clarksville     Tennessee       credit
1       05-26-2011      4000002 198     Exercise & Fitness      Weightlifting Gloves    Long Beach      California      credit
2       06-01-2011      4000002 5       Exercise & Fitness      Weightlifting Machine Accessories       Anaheim California      credit
3       06-05-2011      4000003 198     Gymnastics      Gymnastics Rings        Milwaukee       Wisconsin       credit
4       12-17-2011      4000002 98      Team Sports     Field Hockey    Nashville       Tennessee       credit
5       02-14-2011      4000004 193     Outdoor Recreation      Camping & Backpacking & Hiking  Chicago Illinois        credit
6       10-28-2011      4000005 27      Puzzles Jigsaw Puzzles  Charleston      South Carolina  credit
7       07-14-2011      4000006 96      Outdoor Play Equipment  Sandboxes       Columbus        Ohio    credit
8       01-17-2011      4000006 10      Winter Sports   Snowmobiling    Des Moines      Iowa    credit
9       05-17-2011      4000006 152     Jumping Bungee Jumping  St. Petersburg  Florida credit
10      05-29-2011      4000007 180     Outdoor Recreation      Archery Reno    Nevada  credit
11      06-18-2011      4000009 121     Outdoor Play Equipment  Swing Sets      Columbus        Ohio    credit
12      02-08-2011      4000009 41      Indoor Games    Bowling San Francisco   California      credit
13      03-13-2011      4000010 107     Team Sports     Field Hockey    Honolulu        Hawaii  credit
14      02-25-2011      4000010 36      Gymnastics      Vaulting Horses Los Angeles     California      credit
15      10-20-2011      4000001 137     Combat Sports   Fencing Honolulu        Hawaii  credit
16      05-28-2011      4000010 35      Exercise & Fitness      Free Weight Bars        Columbia        South Carolina  credit
17      10-18-2011      4000008 75      Water Sports    Scuba Diving & Snorkeling       Omaha   Nebraska        credit
18      11-18-2011      4000008 88      Team Sports     Baseball        Salt Lake City  Utah    credit
19      08-28-2011      4000008 51      Water Sports    Life Jackets    Newark  New Jersey      credit
20      06-29-2011      4000005 41      Exercise & Fitness      Weightlifting Belts     New Orleans     Louisiana       credit
21      02-14-2011      4000005 45      Air Sports      Parachutes      New York        New York        credit
22      10-10-2011      4000009 19      Water Sports    Kitesurfing     Saint Paul      Minnesota       credit
23      05-02-2011      4000009 99      Gymnastics      Gymnastics Rings        Springfield     Illinois        credit
24      06-10-2011      4000003 151     Water Sports    Surfing Plano   Texas   credit
25      10-14-2011      4000009 144     Indoor Games    Darts   Phoenix Arizona credit
26      10-11-2011      4000009 31      Combat Sports   Wrestling       Orange  California      credit
27      09-29-2011      4000010 66      Games   Mahjong Fremont California      credit
28      05-12-2011      4000008 79      Team Sports     Cricket Lexington       Kentucky        credit
29      06-03-2011      4000001 126     Outdoor Recreation      Hunting Phoenix Arizona credit
```



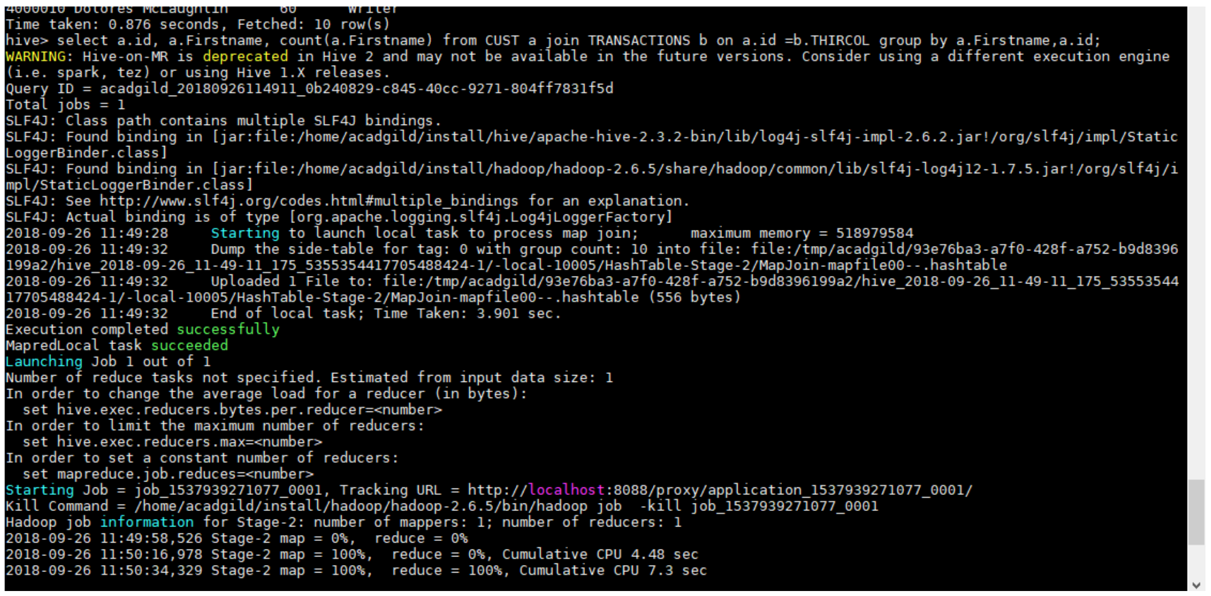
```
23      05-02-2011      4000009 99      Gymnastics      Gymnastics Rings        Springfield     Illinois        credit
24      06-10-2011      4000003 151     Water Sports    Surfing Plano   Texas   credit
25      10-14-2011      4000009 144     Indoor Games    Darts   Phoenix Arizona credit
26      10-11-2011      4000009 31      Combat Sports   Wrestling       Orange  California      credit
27      09-29-2011      4000010 66      Games   Mahjong Fremont California      credit
28      05-12-2011      4000008 79      Team Sports     Cricket Lexington       Kentucky        credit
29      06-03-2011      4000001 126     Outdoor Recreation      Hunting Phoenix Arizona credit
30      03-14-2011      4000001 47      Water Sports    Swimming        Lincoln Nebraska        credit
31      11-28-2011      4000008 5       Games   Dice & Dice Sets        Los Angeles     California      credit
32      01-29-2011      4000008 20      Team Sports     Soccer  Springfield     Illinois        credit
33      06-15-2011      4000008 154     Outdoor Recreation      Lawn Games      Nashville       Tennessee       credit
34      05-06-2011      4000008 98      Team Sports     Indoor Volleyball       Atlanta Georgia credit
35      04-12-2011      4000008 185     Games   Board Games     Centennial      Colorado        credit
36      10-13-2011      4000007 35      Team Sports     Football        Saint Paul      Minnesota       credit
37      04-19-2011      4000007 20      Outdoor Recreation      Shooting Games  San Diego       California      credit
38      08-05-2011      4000007 150     Outdoor Recreation      Camping & Backpacking & Hiking  Hampton         Virginia        credit
39      03-12-2011      4000006 174     Outdoor Play Equipment  Swing Sets      Pittsburgh      Pennsylvania    credit
40      11-07-2011      4000005 165     Team Sports     Cheerleading    Reno    Nevada  credit
41      04-16-2011      4000004 28      Indoor Games    Bowling Westminster     Colorado        cash
42      09-10-2011      4000004 38      Outdoor Recreation      Tetherball      Denton  Texas   cash
43      04-22-2011      4000004 32      Water Sports    Water Polo      Las Vegas       Nevada  cash
44      09-11-2011      4000001 135     Water Sports    Surfing Seattle Washington      credit
45      11-27-2011      4000001 90      Exercise & Fitness      Abdominal Equipment     Honolulu        Hawaii  credit
46      05-27-2011      4000001 52      Gymnastics      Vaulting Horses Cleveland       Ohio    credit
47      10-23-2011      4000008 100     Outdoor Play Equipment  Swing Sets      Everett Washington      credit
48      09-27-2011      4000007 157     Exercise & Fitness      Exercise Bands  Philadelphia    Pennsylvania    credit
49      07-12-2011      4000010 144     Jumping Jumping Stilts  Cambridge       Massachusetts   credit
50      10-20-2011      4000010 55      Jumping Pogo Sticks     Everett Washington      credit
51      02-17-2011      4000002 32      Water Sports    Life Jackets    Columbus        Georgia cash
52      02-04-2011      4000005 44      Outdoor Play Equipment  Lawn Water Slides       Hampton         Virginia        cash
53      06-12-2011      4000004 44      Water Sports    Scuba Diving & Snorkeling       Charleston      South Carolina  cash
54      10-03-2011      4000007 154     Outdoor Recreation      Running Long Beach      California      credit
55      12-16-2011      4000006 106     Water Sports    Swimming        New York        New York        credit
56      06-21-2011      4000002 176     Outdoor Recreation      Geocaching      Boston  Massachusetts   credit
57      12-20-2011      4000003 178     Outdoor Recreation      Skating San Jose        California      credit
58      12-29-2011      4000002 194     Water Sports    Windsurfing     Oklahoma City   Oklahoma        credit
59      11-07-2011      4000001 21      Winter Sports   Snowboarding    Philadelphia    Pennsylvania    cash
Time taken: 3.878 seconds, Fetched: 60 row(s)
```

obaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

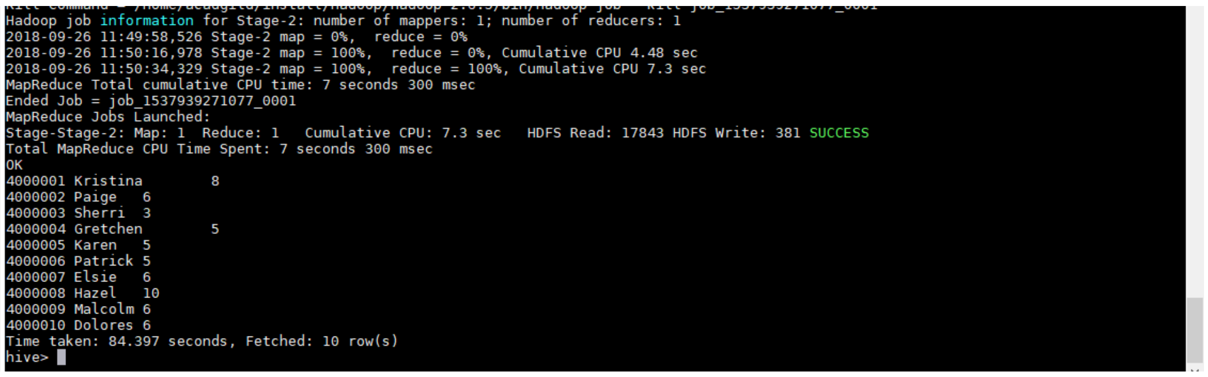
1. **Find out the number of transaction done by each customer (These should be take up in module 8 itself)**

Solution:

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



```
4000010 Dolores McLaughlin      60      Writer
Time taken: 0.876 seconds, Fetched: 10 row(s)
hive> select a.id, a.Firstname, count(a.Firstname) from CUST a join TRANSACTIONS b on a.id =b.THIRCOL group by a.Firstname,a.id;
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_20180926114911_0b240829-c845-40cc-9271-804ff7831f5d
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/Static
LoggerBinder.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/slf4j/i
mpl/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]
2018-09-26 11:49:28     Starting to launch local task to process map join;      maximum memory = 518979584
2018-09-26 11:49:32     Dump the side-table for tag: 0 with group count: 10 into file: file:/tmp/acadgild/93e76ba3-a7f0-428f-a752-b9d8396
199a2/hive_2018-09-26_11-49-11_175_5355354417705488424-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile00--.hashtable
2018-09-26 11:49:32     Uploaded 1 File to: file:/tmp/acadgild/93e76ba3-a7f0-428f-a752-b9d8396199a2/hive_2018-09-26_11-49-11_175_53553544
17705488424-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile00--.hashtable (556 bytes)
2018-09-26 11:49:32     End of local task; Time Taken: 3.901 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_1537939271077_0001, Tracking URL = http://localhost:8088/proxy/application_1537939271077_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1537939271077_0001
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-26 11:49:58,526 Stage-2 map = 0%,  reduce = 0%
2018-09-26 11:50:16,978 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.48 sec
2018-09-26 11:50:34,329 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.3 sec
```

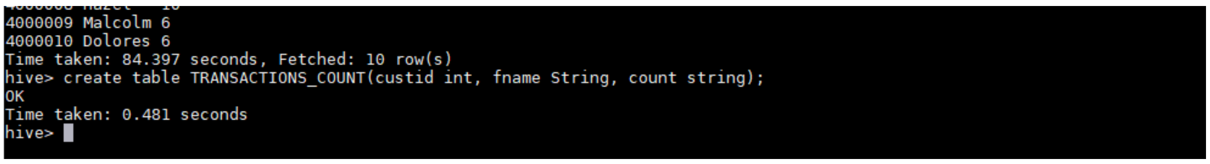


```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-26 11:49:58,526 Stage-2 map = 0%,  reduce = 0%
2018-09-26 11:50:16,978 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.48 sec
2018-09-26 11:50:34,329 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.3 sec
MapReduce Total cumulative CPU time: 7 seconds 300 msec
Ended Job = job_1537939271077_0001
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.3 sec   HDFS Read: 17843 HDFS Write: 381 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 300 msec
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: 84.397 seconds, Fetched: 10 row(s)
hive>
```

2. **Create a new table called TRANSACTIONS_COUNT. This table should have 3 fields - custid, fname and count. (Again to be done in module 8)**

Solution:

create table TRANSACTIONS_COUNT(custid int, fname String, count string);

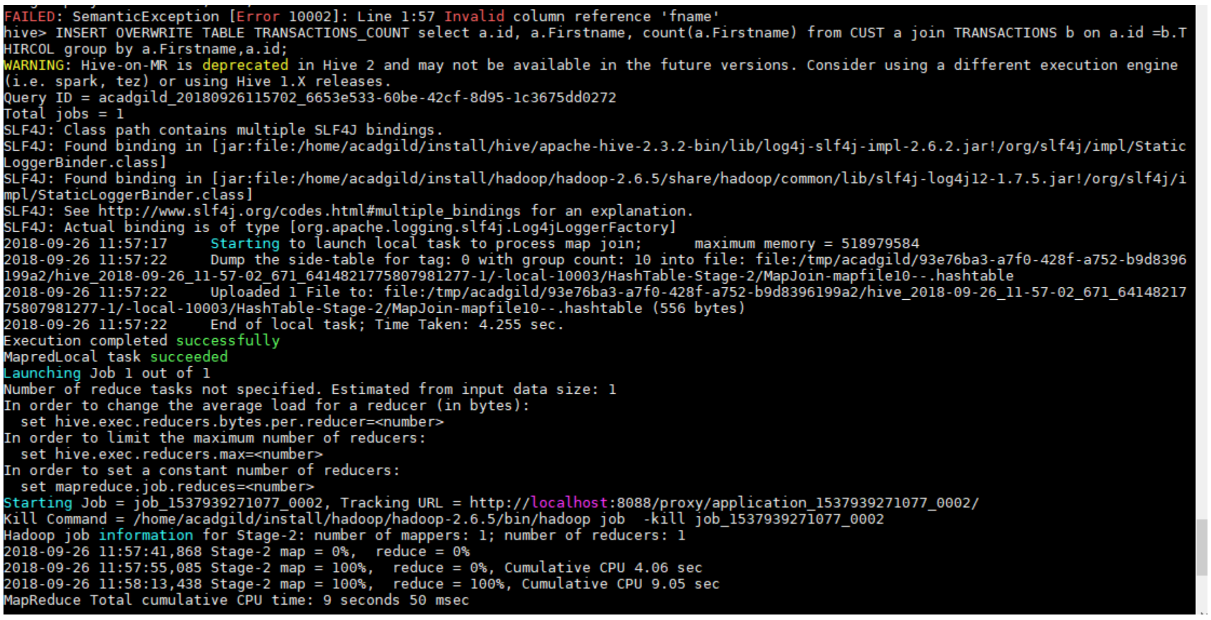


```
4000009 Malcolm 6
4000010 Dolores 6
Time taken: 84.397 seconds, Fetched: 10 row(s)
hive> create table TRANSACTIONS_COUNT(custid int, fname String, count string);
OK
Time taken: 0.481 seconds
hive>
```

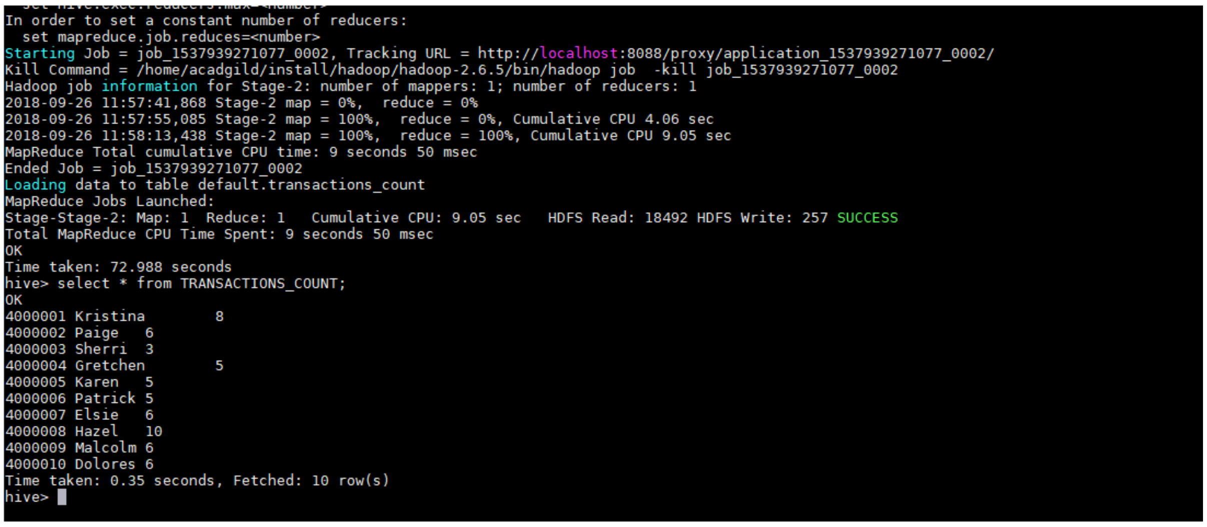
3. **Now write a hive query in such a way that the query populates the data obtained in Step 1 above and populate the table in step 2 above. (This has to be done in module 9).**

Solution:

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



```
FAILED: SemanticException [Error 10002]: Line 1:57 Invalid column reference 'fname'
hive> INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.id, a.Firstname, count(a.Firstname) from CUST a join TRANSACTIONS b on a.id =b.T
HIRCOL group by a.Firstname,a.id;
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_20180926115702_6653e533-60be-42cf-8d95-1c3675dd0272
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/Static
LoggerBinder.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/slf4j/i
mpl/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]
2018-09-26 11:57:17     Starting to launch local task to process map join;      maximum memory = 518979584
2018-09-26 11:57:22     Dump the side-table for tag: 0 with group count: 10 into file: file:/tmp/acadgild/93e76ba3-a7f0-428f-a752-b9d8396
199a2/hive_2018-09-26_11-57-02_671_6414821775807981277-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile10--.hashtable
2018-09-26 11:57:22     Uploaded 1 File to: file:/tmp/acadgild/93e76ba3-a7f0-428f-a752-b9d8396199a2/hive_2018-09-26_11-57-02_671_64148217
75807981277-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile10--.hashtable (556 bytes)
2018-09-26 11:57:22     End of local task; Time Taken: 4.255 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_1537939271077_0002, Tracking URL = http://localhost:8088/proxy/application_1537939271077_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1537939271077_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-26 11:57:41,868 Stage-2 map = 0%,  reduce = 0%
2018-09-26 11:57:55,085 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.06 sec
2018-09-26 11:58:13,438 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.05 sec
MapReduce Total cumulative CPU time: 9 seconds 50 msec
```

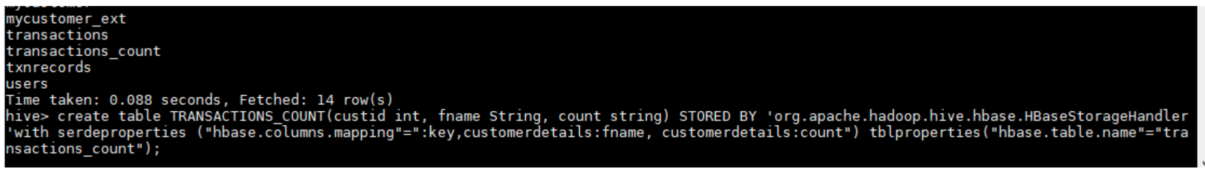


```
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1537939271077_0002, Tracking URL = http://localhost:8088/proxy/application_1537939271077_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1537939271077_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-26 11:57:41,868 Stage-2 map = 0%,  reduce = 0%
2018-09-26 11:57:55,085 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.06 sec
2018-09-26 11:58:13,438 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.05 sec
MapReduce Total cumulative CPU time: 9 seconds 50 msec
Ended Job = job_1537939271077_0002
Loading data to table default.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.05 sec   HDFS Read: 18492 HDFS Write: 257 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 50 msec
OK
Time taken: 72.988 seconds
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: 0.35 seconds, Fetched: 10 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. (This has to be done in module 10)

Solution:

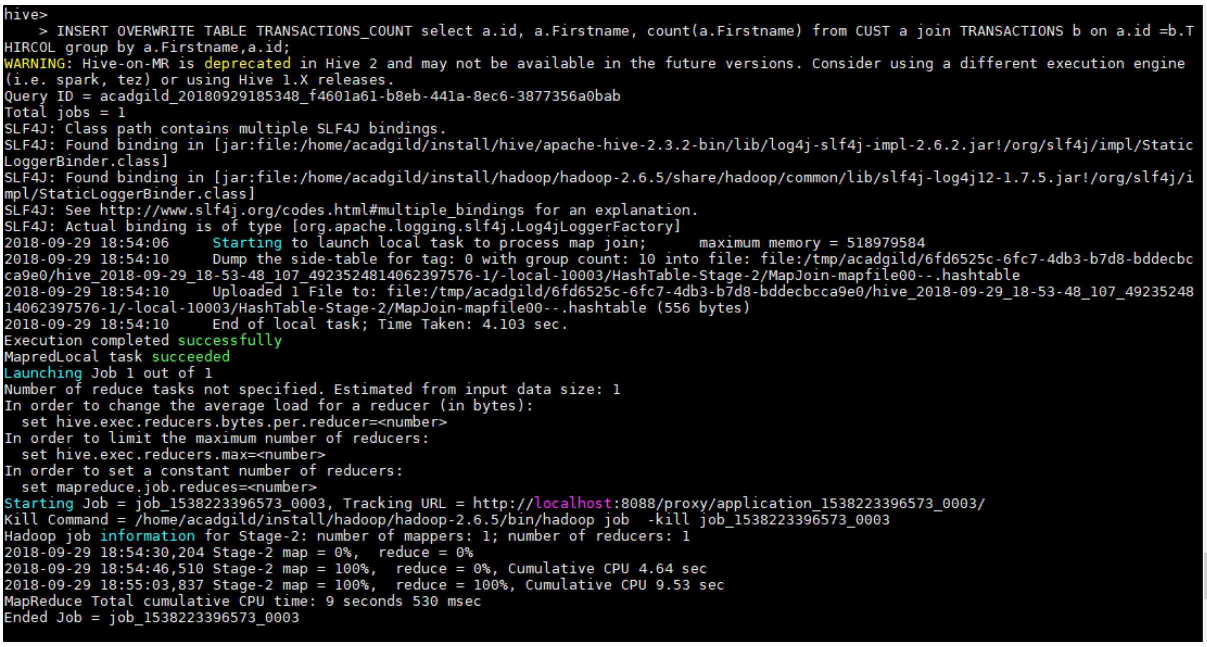
create table TRANSACTIONS_COUNT(custid int, fname String, count string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'with serdeproperties
("hbase.columns.mapping"=":key,customerdetails:fname, customerdetails:count")
tblproperties("hbase.table.name"="transactions_count");



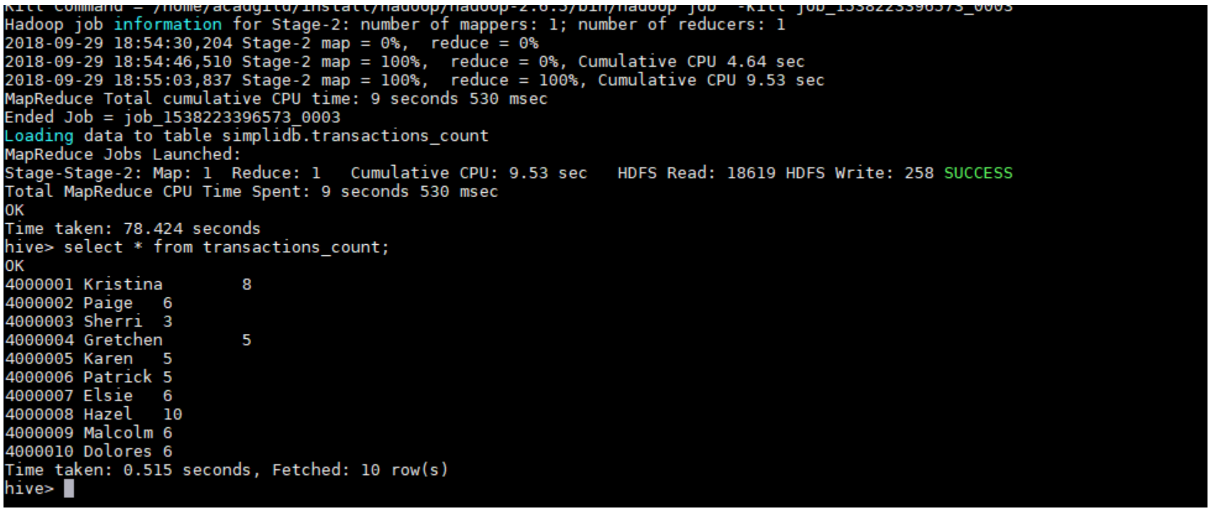
```
mycustomer_ext
transactions
transactions_count
txnrecords
users
Time taken: 0.088 seconds, Fetched: 14 row(s)
hive> create table TRANSACTIONS_COUNT(custid int, fname String, count string) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler
'with serdeproperties ("hbase.columns.mapping"=":key,customerdetails:fname, customerdetails:count") tblproperties("hbase.table.name"="tra
nsactions_count");
```

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 :



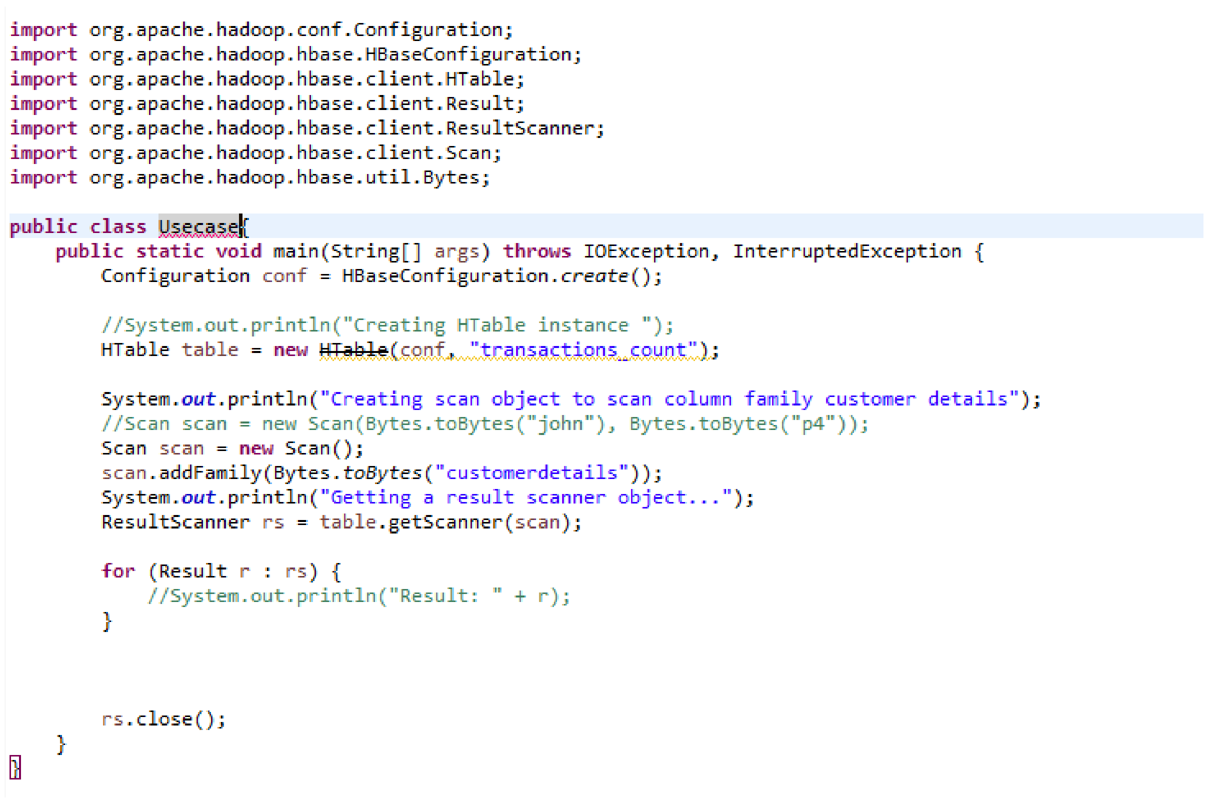
```
hive>
    > INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.id, a.Firstname, count(a.Firstname) from CUST a join TRANSACTIONS b on a.id =b.T
HIRCOL group by a.Firstname,a.id;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180929185348_f4601a61-b8eb-441a-8ec6-3877356a0bab
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/Static
LoggerBinder.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/slf4j/i
mpl/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]
2018-09-29 18:54:06     Starting to launch local task to process map join;      maximum memory = 518979584
2018-09-29 18:54:10     Dump the side-table for tag: 0 with group count: 10 into file: file:/tmp/acadgild/6fd6525c-6fc7-4db3-b7d8-bddecbc
ca9e0/hive_2018-09-29_18-53-48_107_4923524814062397576-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile00--.hashtable
2018-09-29 18:54:10     Uploaded 1 File to: file:/tmp/acadgild/6fd6525c-6fc7-4db3-b7d8-bddecbcca9e0/hive_2018-09-29_18-53-48_107_49235248
14062397576-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile00--.hashtable (556 bytes)
2018-09-29 18:54:10     End of local task; Time Taken: 4.103 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_1538223396573_0003, Tracking URL = http://localhost:8088/proxy/application_1538223396573_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1538223396573_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-29 18:54:30,204 Stage-2 map = 0%,  reduce = 0%
2018-09-29 18:54:46,510 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.64 sec
2018-09-29 18:55:03,837 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.53 sec
MapReduce Total cumulative CPU time: 9 seconds 530 msec
Ended Job = job_1538223396573_0003
```



```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-09-29 18:54:30,204 Stage-2 map = 0%,  reduce = 0%
2018-09-29 18:54:46,510 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.64 sec
2018-09-29 18:55:03,837 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.53 sec
MapReduce Total cumulative CPU time: 9 seconds 530 msec
Ended Job = job_1538223396573_0003
Loading data to table simplidb.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.53 sec   HDFS Read: 18619 HDFS Write: 258 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 530 msec
OK
Time taken: 78.424 seconds
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: 0.515 seconds, Fetched: 10 row(s)
hive> 
```

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

Solution:



```java
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;

public class Usecase{
    public static void main(String[] args) throws IOException, InterruptedException {
        Configuration conf = HBaseConfiguration.create();

        //System.out.println("Creating HTable instance ");
        HTable table = new ~~HTable~~(conf, "transactions_count");

        System.out.println("Creating scan object to scan column family customer details");
        //Scan scan = new Scan(Bytes.toBytes("john"), Bytes.toBytes("p4"));
        Scan scan = new Scan();
        scan.addFamily(Bytes.toBytes("customerdetails"));
        System.out.println("Getting a result scanner object...");
        ResultScanner rs = table.getScanner(scan);

        for (Result r : rs) {
            //System.out.println("Result: " + r);
        }



        rs.close();
    }
}
```