Case-Study Day 11.2

Task 1:
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 let's 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.

Answers:

hive -S
(Starting HIVE in suppress mode to avoid excess info)

create database acadgilddb;
(Creating a database by the name acadgilddb)

show databases;
(Listing the databases present)

acadgilddb
use acadgilddb;

(Use acadgilddb to create both the tables **customer** & **transactions**.)

Screenshot:

```
hive> use acadgilddb;
hive> show tables
    > ;
hive>
hive> CREATE TABLE CUSTOMER(
    > custid INT,
    > fname STRING,
    > lname STRING,
    > age INT,
    > profession STRING)
    > row format delimited fields terminated by ',';
hive> LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/TestHadoop/hive/custs.txt'
    > into table CUSTOMER;
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 '/home/acadgild/Desktop/TestHadoop/hive/txn.txt'
    > into table TRANSACTIONS;
```

```
hive> select * from CUSTOMER;
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
hive> select * from TRANSACTIONS;
97834   05/02/2018      101     965.0   Entertainment   Movie   Pune    Maharashtra     Daughter
98396   12/01/2018      102     239.0   Food    Grocery Patna   Bihar   Self
34908   06/01/2018      101     875.0   Travel  Air     Bangalore       Karnataka       Spouse
70958   17/02/2018      104     439.0   Food    Restaurant      Delhi   Delhi   Wife
9874    21/01/2018      105     509.0   Entertainment   Park    Kolkata West Bengal     NULL
94585   19/01/2018      106     629.0   Rent    House   Hyderabad       Telangana       Self
45509   20/01/2018      107     953.0   Travel  Rail    Chennai Tamil Nadu      Brother
7864    01/02/2018      108     569.0   Rent    Parking Goa     Goa     Wife
```

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

Ans:

SELECT t.custno,c.fname,count(txnno) FROM transactions t JOIN customer c on t.custno = c.custid GROUP BY t.custno,c.fname;

(listing out names of all such customers who have done a transaction by joining both the tables on cust id**)**

Screenshot:

```
hive> select c.fname, t.custno, count(txnno) from TRANSACTIONS t join CUSTOMER c on t.custno=c.custid group by t.custno,c.fname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different executio
ark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180806220013_a0f977ee-f25f-4586-82a1-d49a9fea6a8c
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/im
nder.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!/or
icLoggerBinder.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]
2018-08-06 22:00:29     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-06 22:00:33     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/4f191ae7-e14d-4cc2-b22b
e_2018-08-06_22-00-13_413_1094874905996884690-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2018-08-06 22:00:33     Uploaded 1 File to: file:/tmp/acadgild/4f191ae7-e14d-4cc2-b22b-433fe808fe5b/hive_2018-08-06_22-00-13_413
690-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (469 bytes)
2018-08-06 22:00:33     End of local task; Time Taken: 3.492 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_1533488475608_0003, Tracking URL = http://localhost:8088/proxy/application_1533488475608_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-06 22:00:52,746 Stage-2 map = 0%,  reduce = 0%
2018-08-06 22:01:05,935 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.83 sec
2018-08-06 22:01:19,987 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.81 sec
MapReduce Total cumulative CPU time: 6 seconds 810 msec
Ended Job = job_1533488475608_0003
MapReduce Jobs Launched:
```

```
2018-08-06 22:00:29     Starting to launch local task to process map join;      maximum memory = 518979
2018-08-06 22:00:33     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgil
e_2018-08-06_22-00-13_413_1094874905996884690-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hash
2018-08-06 22:00:33     Uploaded 1 File to: file:/tmp/acadgild/4f191ae7-e14d-4cc2-b22b-433fe808fe5b/hive
690-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (469 bytes)
2018-08-06 22:00:33     End of local task; Time Taken: 3.492 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_1533488475608_0003, Tracking URL = http://localhost:8088/proxy/application_153348847
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-06 22:00:52,746 Stage-2 map = 0%,  reduce = 0%
2018-08-06 22:01:05,935 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.83 sec
2018-08-06 22:01:19,987 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.81 sec
MapReduce Total cumulative CPU time: 6 seconds 810 msec
Ended Job = job_1533488475608_0003
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.81 sec   HDFS Read: 13992 HDFS Write: 263 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 810 msec
OK
Amitabh 101     2
Sharukh 102     1
Anubahv 104     1
Pawan   105     1
Aamir   106     1
Salman  107     1
Ranbir  108     1
Time taken: 67.893 seconds, Fetched: 7 row(s)
```

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)

Ans:

Screenshot:

```
hive> CREATE TABLE TRANSACTIONS_COUNT(
    > custid INT,
    > fname STRING,
    > count INT
    > )
    > row format delimited fields terminated by '\t';
OK
Time taken: 1.399 seconds
```

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 must be done in module 9).

Ans:

CREATE VIEW trans_count_view AS
SELECT t.custno,c.fname,count(txnno) from transactions t join customer c on t.custno=c.custid **GROUP BY** t.custno,c.fname;
(Creating a view to store the result of transaction count. With the help of this view data would be feeded into newly created table).

Note: Data could have been directly inserted from the query itself but i have created a view to reduce the code complexity.

ScreenShot:

```
hive> CREATE VIEW trans_count_view AS
    > select t.custno,c.fname,count(txnno) from TRANSACTIONS t join CUSTOMER c on t.custno=c.custid group by t.custno,c.fname;
OK
Time taken: 6.199 seconds
```

FROM trans_count_view

INSERT INTO TRANSACTIONS_COUNT **SELECT *;**
(Inserting into TRANSACTIONS_COUNT table for the view created.**)**

select * **from** TRANSACTIONS_COUNT;
(Displaying contents of TRANSACTIONS_COUNT table)

Screenshot:

```
hive> FROM trans_count_view
    > INSERT INTO TRANSACTIONS_COUNT SELECT *;
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. sp
ark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180806223434_d6e80ce5-9711-4b8b-9b10-d501b4204d76
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/StaticLoggerBi
nder.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/impl/Stat
icLoggerBinder.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]
2018-08-06 22:34:51     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-06 22:34:56     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/b11b2db4-b649-4dd9-bbd2-240a646ed94e/hiv
e_2018-08-06_22-34-34_004_2604854635531492418-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2018-08-06 22:34:56     Uploaded 1 File to: file:/tmp/acadgild/b11b2db4-b649-4dd9-bbd2-240a646ed94e/hive_2018-08-06_22-34-34_004_2604854635531492
418-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (469 bytes)
2018-08-06 22:34:56     End of local task; Time Taken: 5.075 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_1533488475608_0004, Tracking URL = http://localhost:8088/proxy/application_1533488475608_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0004
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-06 22:35:21,522 Stage-2 map = 0%,  reduce = 0%
2018-08-06 22:35:36,429 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.0 sec
2018-08-06 22:35:53,979 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 8.75 sec
MapReduce Total cumulative CPU time: 8 seconds 750 msec
Ended Job = job_1533488475608_0004
```

```
2018-08-06 22:34:56     End of local task; Time Taken: 5.075 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_1533488475608_0004, Tracking URL = http://localhost:8088/proxy/application_1533488475608_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0004
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-06 22:35:21,522 Stage-2 map = 0%,  reduce = 0%
2018-08-06 22:35:36,429 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.0 sec
2018-08-06 22:35:53,979 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 8.75 sec
MapReduce Total cumulative CPU time: 8 seconds 750 msec
Ended Job = job_1533488475608_0004
Loading data to table acadgilddb.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 8.75 sec   HDFS Read: 14715 HDFS Write: 177 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 750 msec
OK
Time taken: 83.313 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
Time taken: 0.526 seconds, Fetched: 7 row(s)
```

4. Now let's 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 must be done in module 10)

Ans:

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");

(Creating a table transaction in HBase with ***details*** as column family along with a transactions_hbase table in HIVE. The **rowkey, name & txn_count** of TRANSACTIONS table in **HBase** are mapping to **custid, fname & count** columns of TRANSACTIONS_HBase table in **HIVE)**

ScreenShot:

NOTE: Before create table command in HIVE.

HBase does not consists of transactions table.

```
hbase(main):018:0> list
TABLE
bulktable
clicks
customer
dept_tbl
employee
htest
people
t1
8 row(s) in 0.0190 seconds

=> ["bulktable", "clicks", "customer", "dept_tbl", "employee", "htest", "people", "t1"]
```

After the above create table command in HIVE:

ScreenShot:

```
hive> use acadgilddb;
OK
Time taken: 0.041 seconds
hive> show tables;
OK
customer
trans_count_view
transactions
transactions_count
Time taken: 0.133 seconds, Fetched: 4 row(s)
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");
OK
Time taken: 1.725 seconds
hive> show tables;
OK
customer
trans_count_view
transactions
transactions_count
transactions_hbase
Time taken: 0.134 seconds, Fetched: 5 row(s)
hive> desc transactions_hbase;
OK
custid                  int
fname                   string
count                   int
Time taken: 0.225 seconds, Fetched: 3 row(s)
hive>
```

HBase :

```
hbase(main):019:0> list
TABLE
TRANSACTIONS
bulktable
clicks
customer
dept_tbl
employee
htest
people
t1
9 row(s) in 0.0170 seconds

=> ["TRANSACTIONS", "bulktable", "clicks", "customer", "dept_tbl", "employee", "htest", "people", "t1"]
hbase(main):020:0> desc "TRANSACTIONS"
Table TRANSACTIONS is ENABLED
TRANSACTIONS
COLUMN FAMILIES DESCRIPTION
{NAME => 'details', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TT
L => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1970 seconds
```

NOTE: If HBase **transactions** table is disabled & dropped at this point the transactions_hbase table is HIVE would also automatically get dropped.

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 must be done in module 10)

Ans:
Using the same view as in Step 3 above to insert the data in newly created TRANSACTION_HBASE table.
FROM trans_count_view
INSERT INTO transactions_hbase SELECT *;

Screenshot:

```
hive> FROM trans_count_view
    > INSERT INTO TRANSACTIONS_HBase SELECT *;
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. sp
ark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180807120237_c7849a01-d799-42fb-9af0-ebc36e36286d
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/StaticLoggerBi
nder.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/impl/Stat
icLoggerBinder.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]
2018-08-07 12:02:57     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-07 12:03:00     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/4a6313be-7ac2-427b-83d6-008e41adf559/hiv
e_2018-08-07_12-02-37_126_3774401790868693631-1/-local-10002/HashTable-Stage-4/MapJoin-mapfile01--.hashtable
2018-08-07 12:03:01     Uploaded 1 File to: file:/tmp/acadgild/4a6313be-7ac2-427b-83d6-008e41adf559/hive_2018-08-07_12-02-37_126_3774401790868693
631-1/-local-10002/HashTable-Stage-4/MapJoin-mapfile01--.hashtable (469 bytes)
2018-08-07 12:03:01     End of local task; Time Taken: 3.11 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
```

```
  set mapreduce.job.reduces=<number>
Starting Job = job_1533488475608_0013, Tracking URL = http://localhost:8088/proxy/application_1533488475608_0013/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0013
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2018-08-07 12:03:24,384 Stage-4 map = 0%,  reduce = 0%
2018-08-07 12:03:40,238 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 4.25 sec
2018-08-07 12:03:55,445 Stage-4 map = 100%,  reduce = 76%, Cumulative CPU 9.31 sec
2018-08-07 12:03:57,879 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 11.13 sec
MapReduce Total cumulative CPU time: 11 seconds 130 msec
Ended Job = job_1533488475608_0013
MapReduce Jobs Launched:
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 11.13 sec   HDFS Read: 21293 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 130 msec
OK
Time taken: 82.033 seconds
hive> select * from trans_count_view;
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. sp
ark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180807120502_1c100c7d-aba7-4f3c-b838-ef2d927c84fb
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/StaticLoggerBi
nder.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/impl/Stat
icLoggerBinder.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]
2018-08-07 12:05:20     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-07 12:05:24     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/4a6313be-7ac2-427b-83d6-008e41adf559/hiv
e_2018-08-07_12-05-02_716_5586750326835459835-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2018-08-07 12:05:24     Uploaded 1 File to: file:/tmp/acadgild/4a6313be-7ac2-427b-83d6-008e41adf559/hive_2018-08-07_12-05-02_716_5586750326835459
835-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (469 bytes)
2018-08-07 12:05:24     End of local task; Time Taken: 4.005 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
```

```
2018-08-07 12:05:20     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-07 12:05:24     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/4a6313be-
e_2018-08-07_12-05-02_716_5586750326835459835-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2018-08-07 12:05:24     Uploaded 1 File to: file:/tmp/acadgild/4a6313be-7ac2-427b-83d6-008e41adf559/hive_2018-08-0
835-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (469 bytes)
2018-08-07 12:05:24     End of local task; Time Taken: 4.005 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_1533488475608_0014, Tracking URL = http://localhost:8088/proxy/application_1533488475608_0014/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533488475608_0014
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-07 12:05:40,388 Stage-2 map = 0%,  reduce = 0%
2018-08-07 12:05:54,587 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.21 sec
2018-08-07 12:06:08,649 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.29 sec
MapReduce Total cumulative CPU time: 7 seconds 290 msec
Ended Job = job_1533488475608_0014
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.29 sec   HDFS Read: 13985 HDFS Write: 263 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 290 msec
OK
101     Amitabh 2
102     Sharukh 1
104     Anubahv 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
Time taken: 67.123 seconds, Fetched: 7 row(s)
hive>
```

Within HBase **7** rows inserted simultaneously.

```
hbase(main):021:0> scan "TRANSACTIONS"
ROW                                    COLUMN+CELL
0 row(s) in 0.1070 seconds

hbase(main):022:0> scan "TRANSACTIONS"
ROW                                    COLUMN+CELL
 101                                   column=details:name, timestamp=1533623637153, value=Amitabh
 101                                   column=details:txn_count, timestamp=1533623637153, value=2
 102                                   column=details:name, timestamp=1533623637153, value=Sharukh
 102                                   column=details:txn_count, timestamp=1533623637153, value=1
 104                                   column=details:name, timestamp=1533623637153, value=Anubahv
 104                                   column=details:txn_count, timestamp=1533623637153, value=1
 105                                   column=details:name, timestamp=1533623637153, value=Pawan
 105                                   column=details:txn_count, timestamp=1533623637153, value=1
 106                                   column=details:name, timestamp=1533623637153, value=Aamir
 106                                   column=details:txn_count, timestamp=1533623637153, value=1
 107                                   column=details:name, timestamp=1533623637153, value=Salman
 107                                   column=details:txn_count, timestamp=1533623637153, value=1
 108                                   column=details:name, timestamp=1533623637153, value=Ranbir
 108                                   column=details:txn_count, timestamp=1533623637153, value=1
 7 row(s) in 0.2230 seconds
```

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

Ans: JAVA API's attached as separate files.

Note: Program files are properly documented for a detailed description of each instruction used within the program.

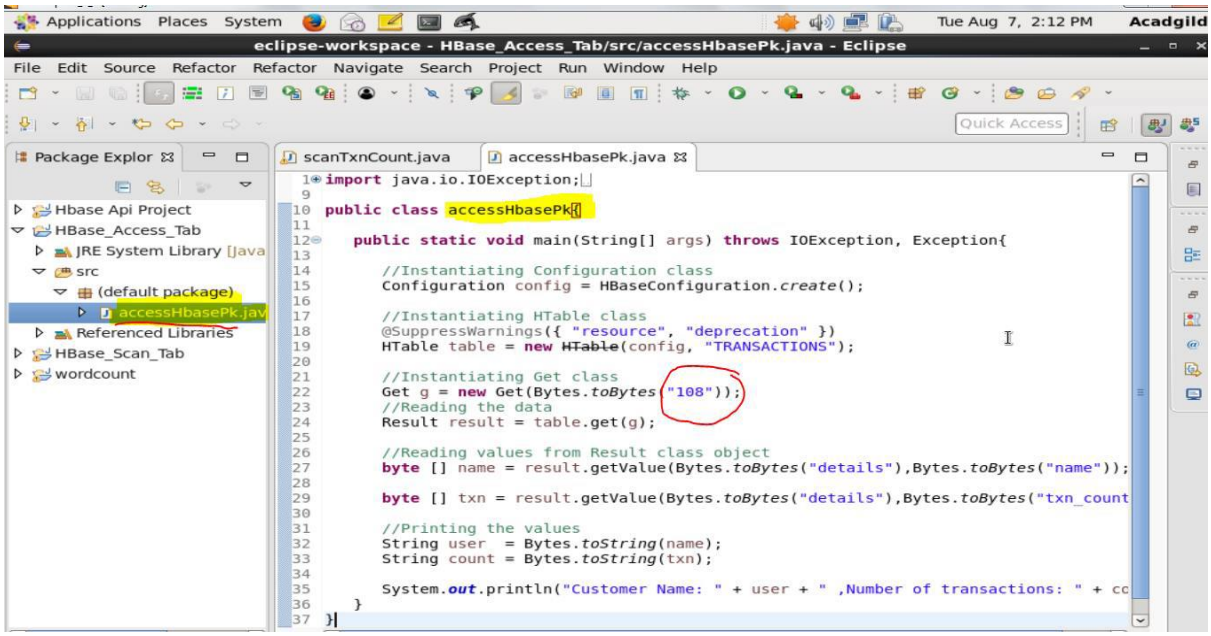
ScreenShot:

The HBase table:

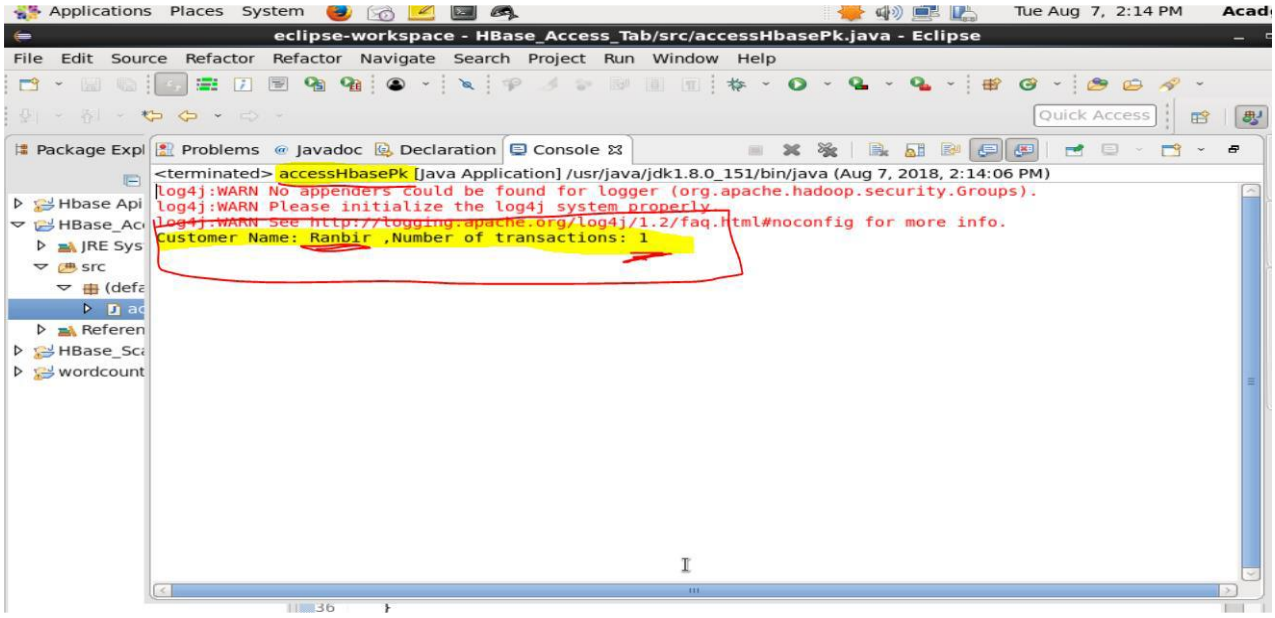
```
hbase(main):021:0> scan "TRANSACTIONS"
ROW                                    COLUMN+CELL
0 row(s) in 0.1070 seconds

hbase(main):022:0> scan "TRANSACTIONS"
ROW                                    COLUMN+CELL
 101                                   column=details:name, timestamp=1533623637153, value=Amitabh
 101                                   column=details:txn_count, timestamp=1533623637153, value=2
 102                                   column=details:name, timestamp=1533623637153, value=Sharukh
 102                                   column=details:txn_count, timestamp=1533623637153, value=1
 104                                   column=details:name, timestamp=1533623637153, value=Anubahv
 104                                   column=details:txn_count, timestamp=1533623637153, value=1
 105                                   column=details:name, timestamp=1533623637153, value=Pawan
 105                                   column=details:txn_count, timestamp=1533623637153, value=1
 106                                   column=details:name, timestamp=1533623637153, value=Aamir
 106                                   column=details:txn_count, timestamp=1533623637153, value=1
 107                                   column=details:name, timestamp=1533623637153, value=Salman
 107                                   column=details:txn_count, timestamp=1533623637153, value=1
 108                                   column=details:name, timestamp=1533623637153, value=Ranbir
 108                                   column=details:txn_count, timestamp=1533623637153, value=1
 7 row(s) in 0.2230 seconds
```

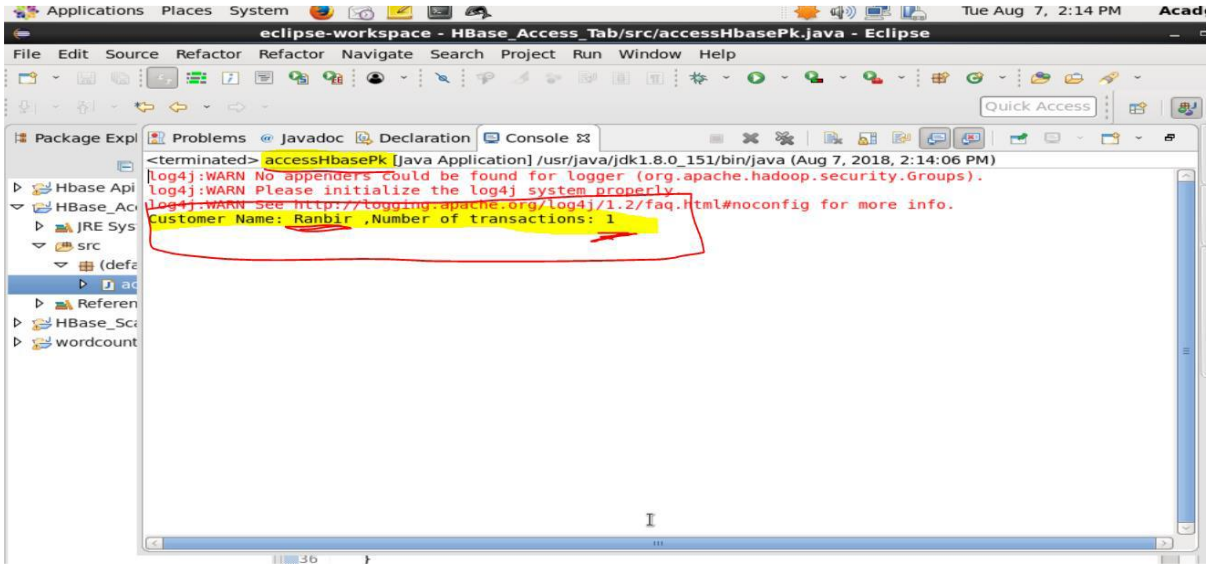
For **Access** HBase:



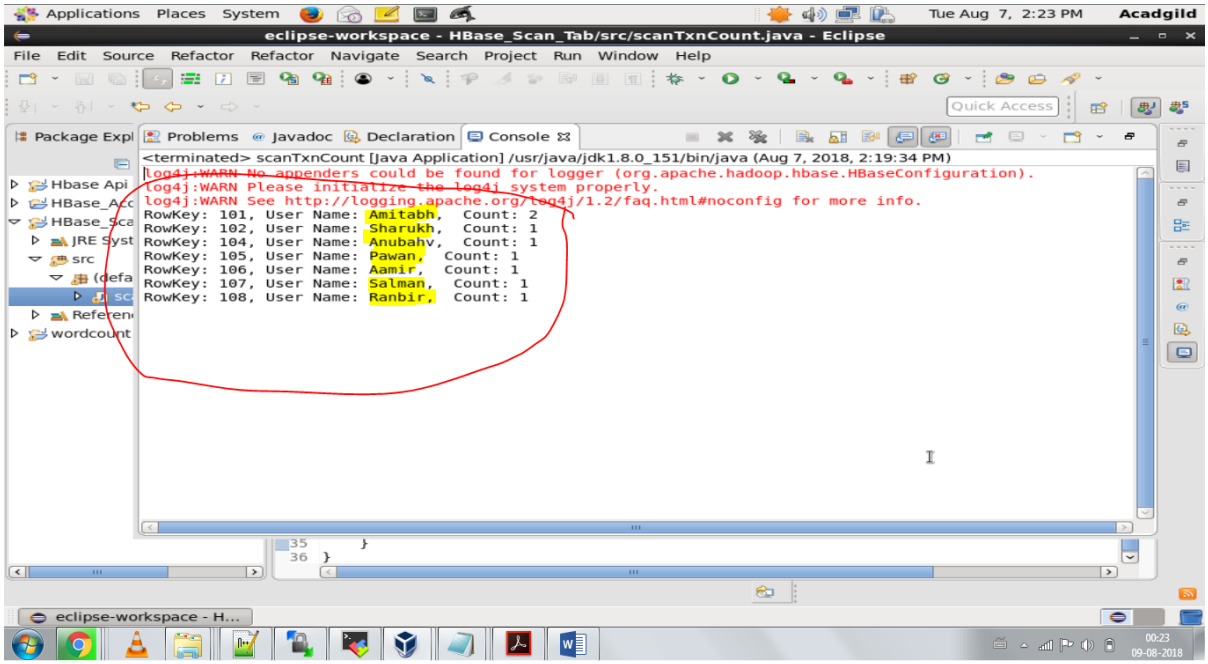
OutPut:



For **Scan** HBase:



OutPut:


eclipse-workspace - HBase_Scan_Tab/src/scanTxnCount.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console
<terminated> scanTxnCount [Java Application] /usr/java/jdk1.8.0_151/bin/java (Aug 7, 2018, 2:19:34 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: 101, User Name: Amitabh, Count: 2
RowKey: 102, User Name: Sharukh, Count: 1
RowKey: 104, User Name: Anubahv, Count: 1
RowKey: 105, User Name: Pawan, Count: 1
RowKey: 106, User Name: Aamir, Count: 1
RowKey: 107, User Name: Salman, Count: 1
RowKey: 108, User Name: Ranbir, Count: 1

********** End **********