Assignment 11_1 Advance HBase Case Study

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)

Answers:

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;
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: 5.489 seconds, Fetched: 8 row(s)
hive> SELECT * FROM TRANSACTIONS;
OK
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
Time taken: 0.4 seconds, Fetched: 8 row(s)
```

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**).**

```
hive> SELECT t.custno,c.fname,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 execution engine (i.e. spark, tez) or
sing Hive 1.X releases.
Query ID = acadgild_20180821190341_f0ad8ea7-70a6-4493-878b-d0b1e09526ca
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/slf4j/impl/StaticLoggerBind
.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]
2018-08-21 19:04:09     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-21 19:04:14     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21
9-03-41_306_292417341547970623 4-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2018-08-21 19:04:14     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-03-41_306_292417341547970623 4-1/-local
0005/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (469 bytes)
2018-08-21 19:04:14     End of local task; Time Taken: 4.861 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_1534850010376_0001, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0001
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:04:56,537 Stage-2 map = 0%,  reduce = 0%
2018-08-21 19:05:24,628 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 5.76 sec
2018-08-21 19:05:43,968 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 9.56 sec
2018-08-21 19:05:46,860 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 11.49 sec
MapReduce Total cumulative CPU time: 11 seconds 490 msec
Ended Job = job_1534850010376_0001
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.49 sec   HDFS Read: 13778 HDFS Write: 263 SUCCESS
```

```
Starting Job = job_1534850010376_0001, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0001/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0001
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:04:56,537 Stage-2 map = 0%,  reduce = 0%
2018-08-21 19:05:24,628 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 5.76 sec
2018-08-21 19:05:43,968 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 9.56 sec
2018-08-21 19:05:46,860 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 11.49 sec
MapReduce Total cumulative CPU time: 11 seconds 490 msec
Ended Job = job_1534850010376_0001
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.49 sec   HDFS Read: 13778 HDFS Write: 263 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 490 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: 126.935 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)

```
hive> CREATE TABLE TRANSACTIONS_COUNT(custid INT,fname STRING,count INT) row format delimited fields terminated by '\t';
OK
Time taken: 1.299 seconds
hive> create view trans_count_view_1 AS select t.custno,c.fname,count(txnno) from TRANSACTIONS t join CUSTOMER c on t.custno =c.custid group by t.custno,c.fnam
e;
OK
Time taken: 1.177 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 **_1 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;

```
hive> create view trans_count_view_1 AS select t.custno,c.fname,count(txnno) from TRANSACTIONS t join CUSTOMER c on t.custno =c.custid group by t.custno,c.fnam
e;
OK
Time taken: 1.177 seconds
```

(Creating a view to store the result of transaction count. With the help of this view data would be feeded into newly created table).

FROM trans_count_view_1

INSERT INTO TRANSACTIONS_COUNT **SELECT *;**

(Inserting into TRANSACTIONS_COUNT table for the view created.**)**

select * **from** TRANSACTIONS_COUNT;

(Displaying contents of TRANSACTIONS_COUNT table

```
hive> FROM trans_count_view_1 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. spark,
sing Hive 1.X releases.
Query ID = acadgild_20180821191040_e5b5bd94-ddb1-453e-914e-567a0738ece5
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.
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/StaticLog
.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]
2018-08-21 19:11:06     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-21 19:11:13     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_201
9-10-40_821_7143080575163628 42-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2018-08-21 19:11:13     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-10-40_821_7143080575163628 42-1/
003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (469 bytes)
2018-08-21 19:11:13     End of local task; Time Taken: 6.899 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_1534850010376_0002, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:11:38,143 Stage-2 map = 0%,  reduce = 0%
2018-08-21 19:12:11,131 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 5.33 sec
2018-08-21 19:13:09,310 Stage-2 map = 100%,  reduce = 33%, Cumulative CPU 6.56 sec
2018-08-21 19:13:13,728 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 7.17 sec
2018-08-21 19:13:33,608 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.72 sec
MapReduce Total cumulative CPU time: 11 seconds 500 msec
Ended Job = job_1534850010376_0002
```

```
2018-08-21 19:11:13     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-10-40_821_7143080575163628 42-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable
2018-08-21 19:11:13     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-10-40_821_7143080575163628 42-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile11--.hashtable (469 bytes)
2018-08-21 19:11:13     End of local task; Time Taken: 6.899 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_1534850010376_0002, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:11:38,143 Stage-2 map = 0%,  reduce = 0%
2018-08-21 19:12:11,131 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 5.33 sec
2018-08-21 19:13:09,310 Stage-2 map = 100%,  reduce = 33%, Cumulative CPU 6.56 sec
2018-08-21 19:13:13,728 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 7.17 sec
2018-08-21 19:13:33,608 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 7.72 sec
MapReduce Total cumulative CPU time: 11 seconds 500 msec
Ended Job = job_1534850010376_0002
Loading data to table acadgilddb.transactions_count
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.5 sec   HDFS Read: 14731 HDFS Write: 177 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 500 msec
OK
Time taken: 186.852 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: 5.293 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 TRANSACTIONS 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:

```
hive> use acadgilddb;
OK
Time taken: 0.035 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 ("hba
se.columns.mapping"=":key,details:name,details:txn_count")tblproperties("hbase.table.name"="TRANSACTIONS");
OK
Time taken: 3.83 seconds
hive> show tables;
OK
customer
trans_count_view_1
transaction_count_view
transactions
transactions_count
transactions_hbase
Time taken: 0.047 seconds, Fetched: 6 row(s)
```

```
hive> show tables;
OK
customer
trans_count_view_1
transaction_count_view
transactions
transactions_count
transactions_hbase
Time taken: 0.047 seconds, Fetched: 6 row(s)
```

```
hive> desc transactions_hbase;
OK
custid                  int
fname                   string
count                   int
Time taken: 0.33 seconds, Fetched: 3 row(s)
```

HBase:

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 *;

```
hive> FROM trans_count_view_1 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. spa
sing Hive 1.X releases.
Query ID = acadgild_20180821192315_92b9546f-761b-4e57-80f8-eee4100f57ba
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/StaticLoggerBin
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/Stati
.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]
2018-08-21 19:23:40     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-21 19:23:45     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive
9-23-15_602_5632595207495579143-1/-local-10002/HashTable-Stage-4/MapJoin-mapfile21--.hashtable
2018-08-21 19:23:45     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-23-15_602_56325952074955791
0002/HashTable-Stage-4/MapJoin-mapfile21--.hashtable (469 bytes)
2018-08-21 19:23:45     End of local task; Time Taken: 5.071 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_1534850010376_0003, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0003/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0003
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2018-08-21 19:24:17,163 Stage-4 map = 0%,  reduce = 0%
2018-08-21 19:24:41,554 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 5.91 sec
2018-08-21 19:25:08,125 Stage-4 map = 100%,  reduce = 67%, Cumulative CPU 10.69 sec
2018-08-21 19:25:17,607 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 17.57 sec
MapReduce Total cumulative CPU time: 17 seconds 570 msec
Ended Job = job_1534850010376_0003
MapReduce Jobs Launched:
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 17.57 sec   HDFS Read: 21314 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 570 msec
OK
Time taken: 124.467 seconds
```

```
hive> select * from Transactions_Hbase;
OK
101     Amitabh 2
102     Sharukh 1
104     Anubahv 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
Time taken: 1.031 seconds, Fetched: 7 row(s)
```

```
hive> select * from trans_count_view_1;
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)
sing Hive 1.X releases.
Query ID = acadgild_20180821192745_54bb65ec-51f2-4e02-a759-210778444d5e
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/slf4j/impl/StaticLoggerBi
.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]
2018-08-21 19:28:08     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-21 19:28:13     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-
9-27-45_242_451017599348496726-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile31--.hashtable
2018-08-21 19:28:14     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-27-45_242_451017599348496726-1/-loca
005/HashTable-Stage-2/MapJoin-mapfile31--.hashtable (469 bytes)
2018-08-21 19:28:14     End of local task; Time Taken: 5.842 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_1534850010376_0004, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0004
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:28:41,237 Stage-2 map = 0%,  reduce = 0%
```

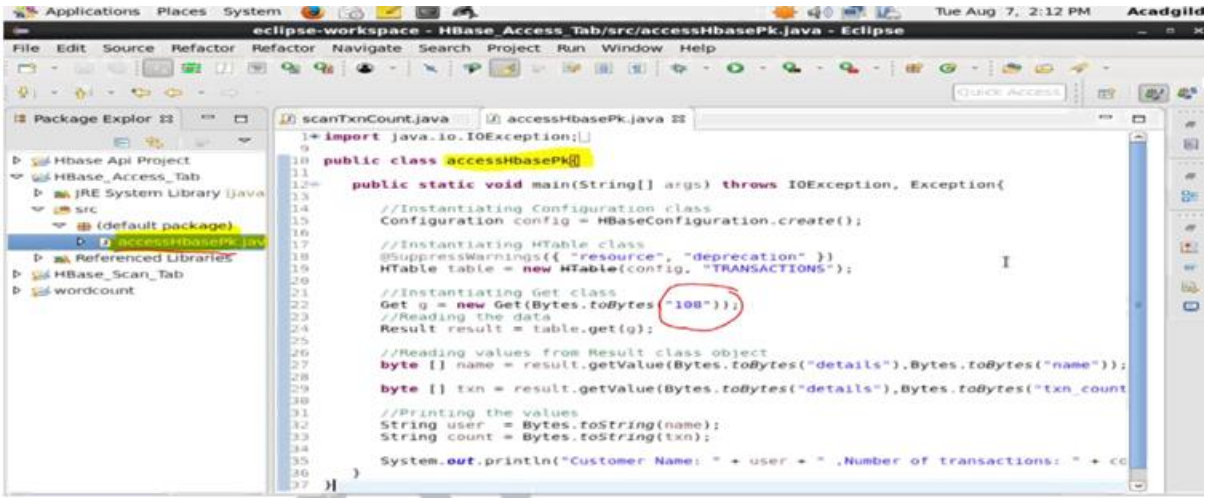
```
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBin
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Stati
.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2018-08-21 19:28:08     Starting to launch local task to process map join;      maximum memory = 518979584
2018-08-21 19:28:13     Dump the side-table for tag: 1 with group count: 8 into file: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive
9-27-45_242_451017599348496726-1/-local-10005/HashTable-Stage-2/MapJoin-mapfile31--.hashtable
2018-08-21 19:28:14     Uploaded 1 File to: file:/tmp/acadgild/111b6a97-8e8e-4730-a48e-1ce340ba7ecf/hive_2018-08-21_19-27-45_242_45101759934849672
005/HashTable-Stage-2/MapJoin-mapfile31--.hashtable (469 bytes)
2018-08-21 19:28:14     End of local task; Time Taken: 5.842 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_1534850010376_0004, Tracking URL = http://localhost:8088/proxy/application_1534850010376_0004/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1534850010376_0004
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-08-21 19:28:41,237 Stage-2 map = 0%,  reduce = 0%
2018-08-21 19:29:01,056 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.69 sec
2018-08-21 19:29:22,200 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.02 sec
MapReduce Total cumulative CPU time: 9 seconds 20 msec
Ended Job = job_1534850010376_0004
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.02 sec   HDFS Read: 13984 HDFS Write: 263 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 20 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: 99.399 seconds, Fetched: 7 row(s)
```

```
hbase(main):007:0> scan "TRANSACTIONS"
ROW                                   COLUMN+CELL
 101                                  column=details:name, timestamp=1534859715976, value=Amitabh
 101                                  column=details:txn_count, timestamp=1534859715976, value=2
 102                                  column=details:name, timestamp=1534859715976, value=Sharukh
 102                                  column=details:txn_count, timestamp=1534859715976, value=1
 104                                  column=details:name, timestamp=1534859715976, value=Anubahv
 104                                  column=details:txn_count, timestamp=1534859715976, value=1
 105                                  column=details:name, timestamp=1534859715976, value=Pawan
 105                                  column=details:txn_count, timestamp=1534859715976, value=1
 106                                  column=details:name, timestamp=1534859715976, value=Aamir
 106                                  column=details:txn_count, timestamp=1534859715976, value=1
 107                                  column=details:name, timestamp=1534859715976, value=Salman
 107                                  column=details:txn_count, timestamp=1534859715976, value=1
 108                                  column=details:name, timestamp=1534859715976, value=Ranbir
 108                                  column=details:txn_count, timestamp=1534859715976, value=1
7 row(s) in 0.9300 seconds
```

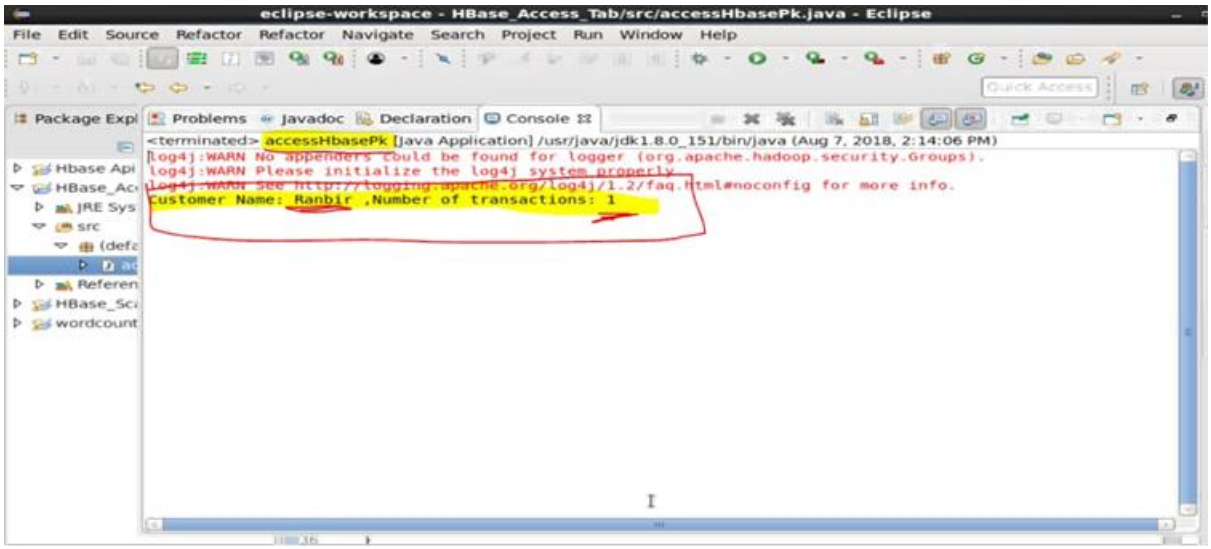
6. Now from the Hbase level, write the Hbase java API code to access and scan the TRANSACTIONS table data from java level.


```
hbase(main):007:0> scan "TRANSACTIONS"
ROW                                   COLUMN+CELL
 101                                  column=details:name, timestamp=1534859715976, value=Amitabh
 101                                  column=details:txn_count, timestamp=1534859715976, value=2
 102                                  column=details:name, timestamp=1534859715976, value=Sharukh
 102                                  column=details:txn_count, timestamp=1534859715976, value=1
 104                                  column=details:name, timestamp=1534859715976, value=Anubahv
 104                                  column=details:txn_count, timestamp=1534859715976, value=1
 105                                  column=details:name, timestamp=1534859715976, value=Pawan
 105                                  column=details:txn_count, timestamp=1534859715976, value=1
 106                                  column=details:name, timestamp=1534859715976, value=Aamir
 106                                  column=details:txn_count, timestamp=1534859715976, value=1
 107                                  column=details:name, timestamp=1534859715976, value=Salman
 107                                  column=details:txn_count, timestamp=1534859715976, value=1
 108                                  column=details:name, timestamp=1534859715976, value=Ranbir
 108                                  column=details:txn_count, timestamp=1534859715976, value=1
7 row(s) in 0.9300 seconds
```

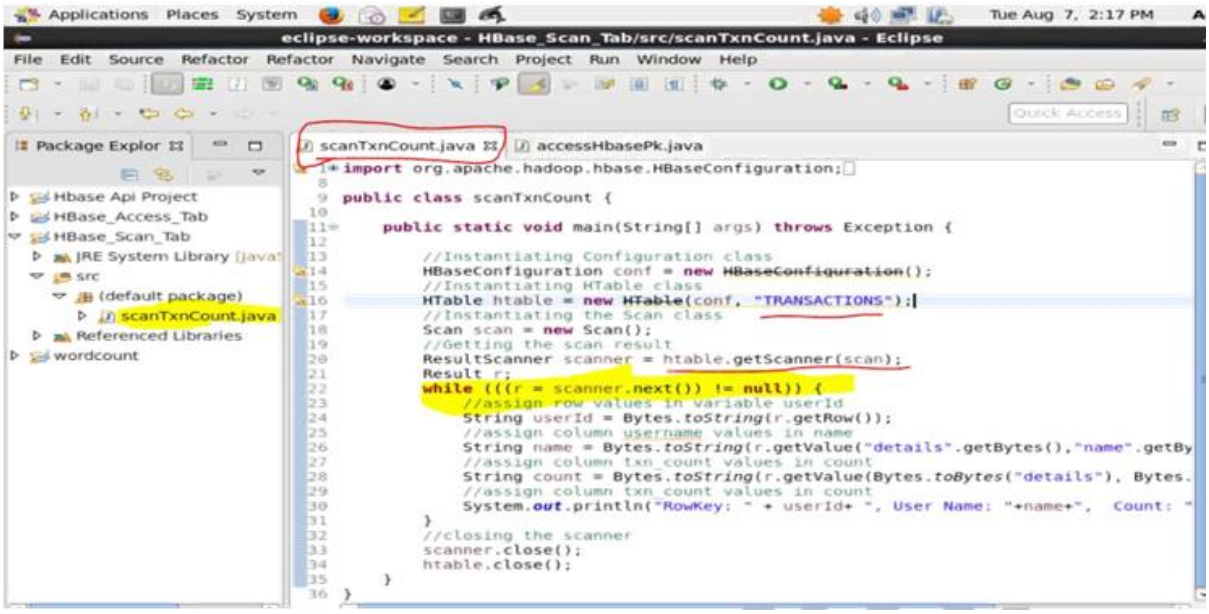
For Acces HBase:



OutPut:



For **Scan** HBase:



Output: