Vijay's Assignment – Hbase3

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)

```
hive> select custid, fname, count(*) from customer a join transactions b on a.custid=b.custno group by custid, fname;

101     Amitabh 2
102     Sharukh 1
104     Anubhav 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
```

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);
OK
Time taken: 0.356 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 has to

be done in module 9).

```
hive> INSERT INTO TRANSACTIONS_COUNT SELECT CUSTID, FNAME, COUNT(*) FROM CUSTOMER A JOIN TRANSACTIONS B ON A.CUSTID=B.CUSTNO
GROUP BY CUSTID, FNAME;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20190109220639_0ed51498-c242-47a8-a645-56f2caca66c4
Total jobs = 1
```

```
hive> SELECT * FROM TRANSACTIONS_COUNT;
OK
101     Amitabh 2
102     Sharukh 1
104     Anubhav 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
Time taken: 0.297 seconds, Fetched: 7 row(s)
```

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)

```
[acadgild@localhost lib]$ hive --auxpath /home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-hbase-handler-2.3.2.jar
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/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]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.j
ar!/hive-log4j2.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 engi
ne (i.e. spark, tez) or using Hive 1.X releases.
```

```
hive> CREATE TABLE TRANSACTIONS_COUNT(CUSTID INT, FNAME STRING, COUNT INT) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStor
geHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key, cf1:FNAME, cf1:count") TBLPROPERTIES ("hbase.table.name"="
RANSACTIONS", "hbase.mapred.output.outputtable"="TRANSACTIONS");
OK
Time taken: 3.189 seconds
hive>
```

```
hbase(main):004:0> describe 'TRANSACTIONS'
Table TRANSACTIONS is ENABLED
TRANSACTIONS
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEE
P_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COM
PRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '655
36', REPLICATION_SCOPE => '0'}
1 row(s) in 0.0800 seconds
```

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)

```
Time taken: 3.189 seconds
hive> INSERT INTO TRANSACTIONS_COUNT SELECT CUSTID, FNAME, COUNT(*) FROM CUSTOMER A JOIN TRANSACTIONS B ON A.CUSTID=B.CUSTNO
GROUP BY CUSTID, FNAME;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20190109234506_79561a0f-5b47-42d8-816b-9de8c4e37f44
```

```
hive> SELECT * FROM TRANSACTIONS_COUNT;
OK
101     Amitabh 2
102     Sharukh 1
104     Anubhav 1
105     Pawan   1
106     Aamir   1
107     Salman  1
108     Ranbir  1
Time taken: 1.023 seconds, Fetched: 7 row(s)
```

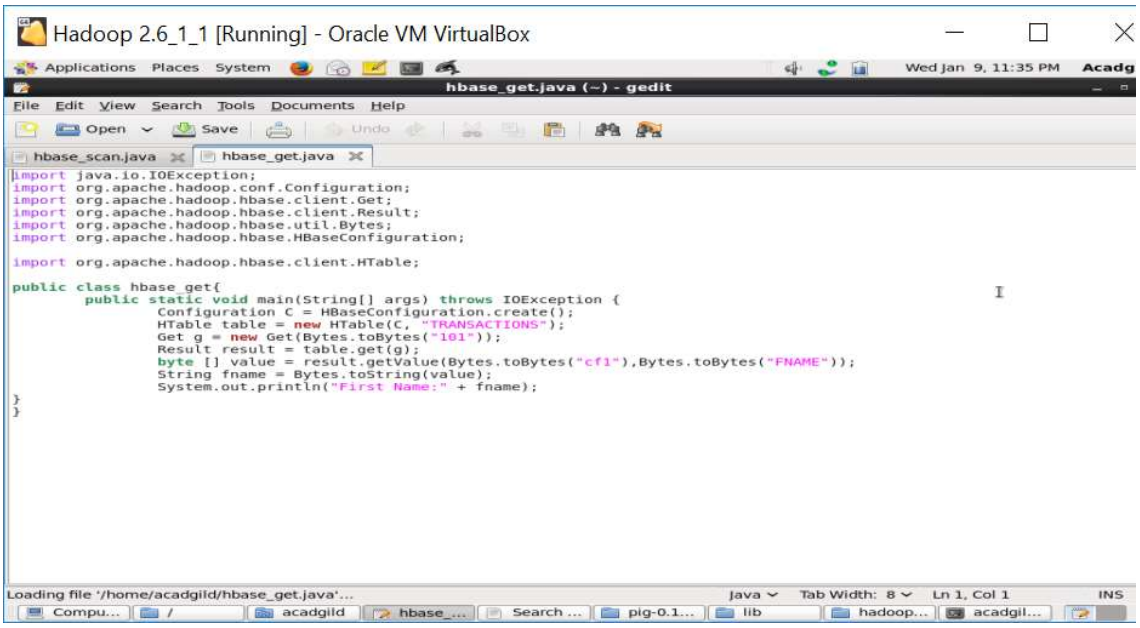
```
hbase(main):005:0> scan 'TRANSACTIONS'
ROW                   COLUMN+CELL
 101                  column=cf1:FNAME, timestamp=1547057806680, value=Amitabh
 101                  column=cf1:count, timestamp=1547057806680, value=2
 102                  column=cf1:FNAME, timestamp=1547057806680, value=Sharukh
 102                  column=cf1:count, timestamp=1547057806680, value=1
 104                  column=cf1:FNAME, timestamp=1547057806680, value=Anubhav
 104                  column=cf1:count, timestamp=1547057806680, value=1
 105                  column=cf1:FNAME, timestamp=1547057806680, value=Pawan
 105                  column=cf1:count, timestamp=1547057806680, value=1
 106                  column=cf1:FNAME, timestamp=1547057806680, value=Aamir
 106                  column=cf1:count, timestamp=1547057806680, value=1
 107                  column=cf1:FNAME, timestamp=1547057806680, value=Salman
 107                  column=cf1:count, timestamp=1547057806680, value=1
 108                  column=cf1:FNAME, timestamp=1547057806680, value=Ranbir
 108                  column=cf1:count, timestamp=1547057806680, value=1
7 row(s) in 0.6680 seconds
```

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

the TRANSACTIONS table data from java level.

```
[acadgild@localhost install]$ export CLASSPATH=$CLASSPATH:/home/acadgild/install/hbase/hbase-1.2.6/lib/*
[acadgild@localhost install]$ export CLASSPATH=$CLASSPATH:/home/acadgild/install/pig/*
```

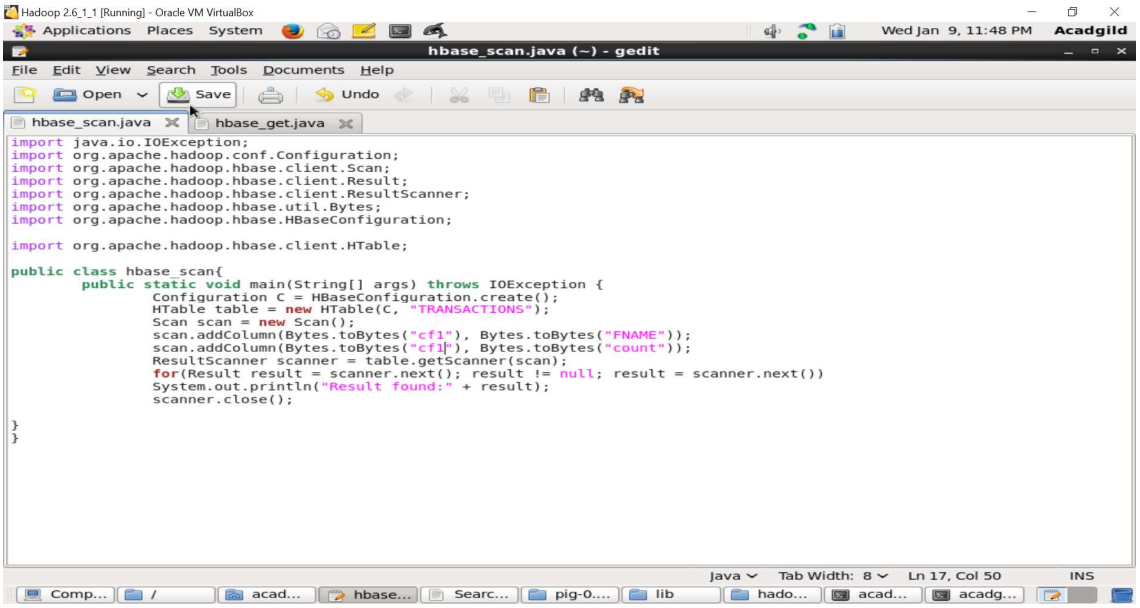
Java Get example:



Result:

```
[acadgild@localhost ~]$ java hbase_get
log4j:WARN No appenders could be found for logger (org.apache.hadoop.security.Groups).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
First Name:Amitabh
[acadgild@localhost ~]$
```

Java Scan example



Result:

```
[acadgild@localhost ~]$ java hbase_scan
log4j:WARN No appenders could be found for logger (org.apache.hadoop.security.Groups).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Result found:keyvalues={101/cf1:FNAME/1547057806680/Put/vlen=7/seqid=0, 101/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={102/cf1:FNAME/1547057806680/Put/vlen=7/seqid=0, 102/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={104/cf1:FNAME/1547057806680/Put/vlen=7/seqid=0, 104/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={105/cf1:FNAME/1547057806680/Put/vlen=5/seqid=0, 105/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={106/cf1:FNAME/1547057806680/Put/vlen=5/seqid=0, 106/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={107/cf1:FNAME/1547057806680/Put/vlen=6/seqid=0, 107/cf1:count/1547057806680/Put/vlen=1/seqid=0}
Result found:keyvalues={108/cf1:FNAME/1547057806680/Put/vlen=6/seqid=0, 108/cf1:count/1547057806680/Put/vlen=1/seqid=0}
```