BIG DATA HADOOP AND SPARK DEVLOPMENT

CASE STUDY I

Table of Contents:

BIG DATA HADOOPAND SPARK DEVELOPMENT

1. Introduction

In this case study, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

2. Objective

This case study consolidates the deeper understanding of the Sessions

3. Problem Statement

- Task 1

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.

4. Expected Output

- ### Task 1

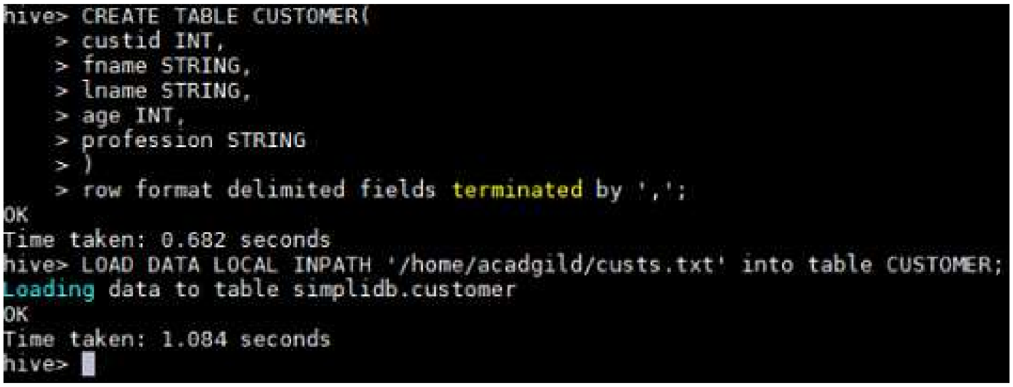
Let us take up the CUSTOMER and TRANSACTIONS table we have created in the Let’s Do Together section. Let us solve the following use cases using these tables :-

Now as mentioned we have already created tables, still I am repeating the same steps again to begin with scratch. First – Creating CUSTOMER table in HIVE shell and Load the data in the table.

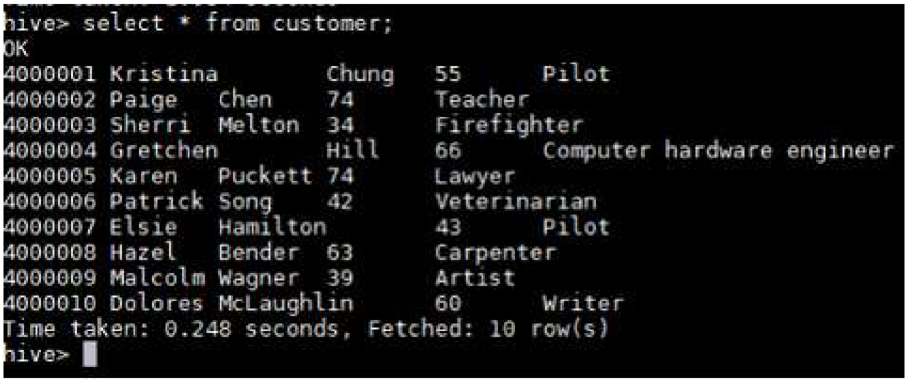
CREATE TABLE CUSTOMER(custid INT, fname STRING, lname STRING, age INT, profession STRING) row format delimited fields terminated by ',';

Loading Data:

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



```
hive> CREATE TABLE CUSTOMER(
    > custid INT,
    > fname STRING,
    > lname STRING,
    > age INT,
    > profession STRING
    > )
    > row format delimited fields terminated by ',';
OK
Time taken: 0.682 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/custs.txt' into table CUSTOMER;
Loading data to table simplidb.customer
OK
Time taken: 1.084 seconds
hive> █
```

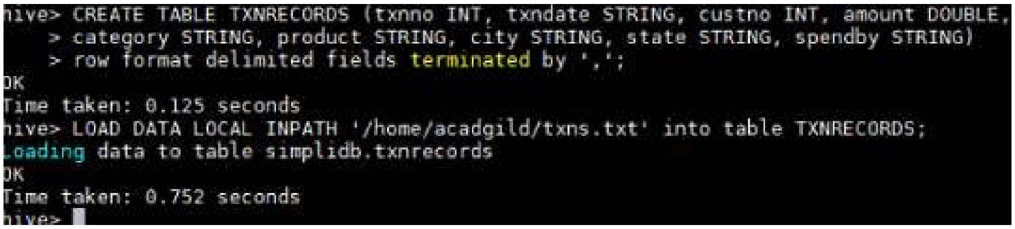


```
hive> select * from customer;
OK
4000001 Kristina        Chung   55      Pilot
4000002 Paige   Chen    74      Teacher
4000003 Sherri  Melton  34      Firefighter
4000004 Gretchen        Hill    66      Computer hardware engineer
4000005 Karen   Puckett 74      Lawyer
4000006 Patrick Song    42      Veterinarian
4000007 Elsie   Hamilton        43      Pilot
4000008 Hazel   Bender  63      Carpenter
4000009 Malcolm Wagner  39      Artist
4000010 Dolores McLaughlin      60      Writer
Time taken: 0.248 seconds, Fetched: 10 row(s)
hive> █
```

Create Transaction record table as mentioned below and load it with data.

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

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

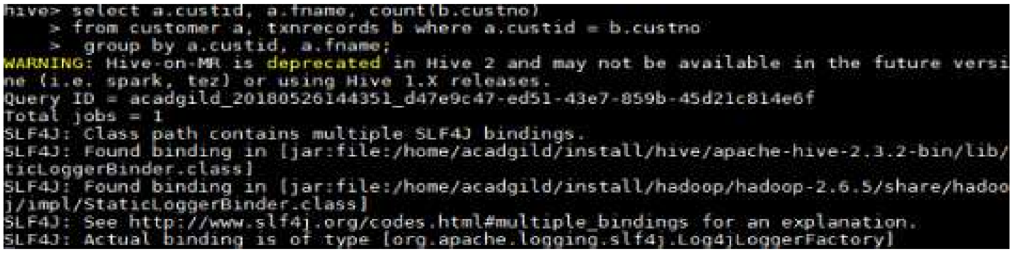


```
hive> CREATE TABLE TXNRECORDS (txnno INT, txndate STRING, custno INT, amount DOUBLE,
    > category STRING, product STRING, city STRING, state STRING, spendby STRING)
    > row format delimited fields terminated by ',';
OK
Time taken: 0.125 seconds
hive> LOAD DATA LOCAL INPATH '/home/acadgild/txns.txt' into table TXNRECORDS;
Loading data to table simplidb.txnrecords
OK
Time taken: 0.752 seconds
hive> 
```

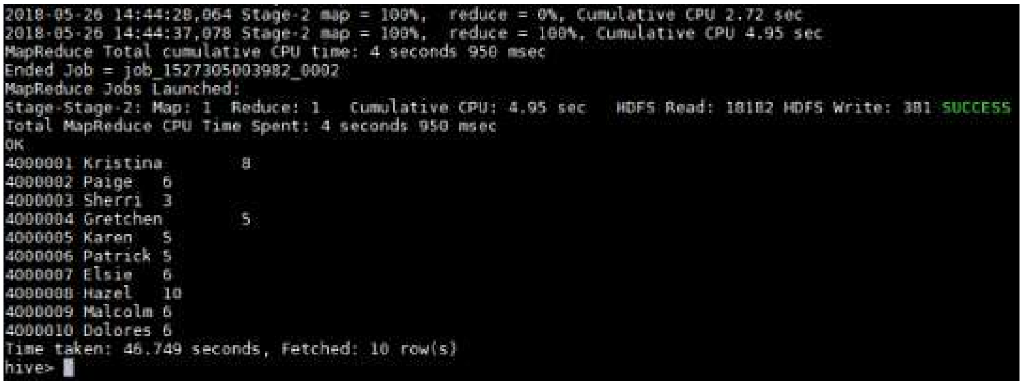
Task 1: Find out the number of transaction done by each customer (These should be take up in module 8 itself).

Answer:

select a.custid, a.fname, count(b.custno) from customer a, txnrecords b where a.custid = b.custno group by a.custid, a.fname;



```
hive> select a.custid, a.fname, count(b.custno)
    > from customer a, txnrecords b where a.custid = b.custno
    >  group by a.custid, a.fname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versi
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180526144351_d47e9c47-ed51-43e7-859b-45d21c814e6f
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/
ticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoo
j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
```

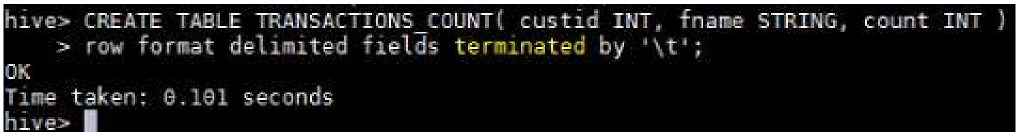


```
2018-05-26 14:44:28,064 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.72 sec
2018-05-26 14:44:37,078 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.95 sec
MapReduce Total cumulative CPU time: 4 seconds 950 msec
Ended Job = job_1527305003982_0002
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.95 sec   HDFS Read: 18182 HDFS Write: 381 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 950 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: 46.749 seconds, Fetched: 10 row(s)
hive> 
```

Task 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).

Answer:

CREATE TABLE TRANSACTIONS_COUNT(custid INT, fname STRING, count INT) row format delimited fields terminated by '\t';

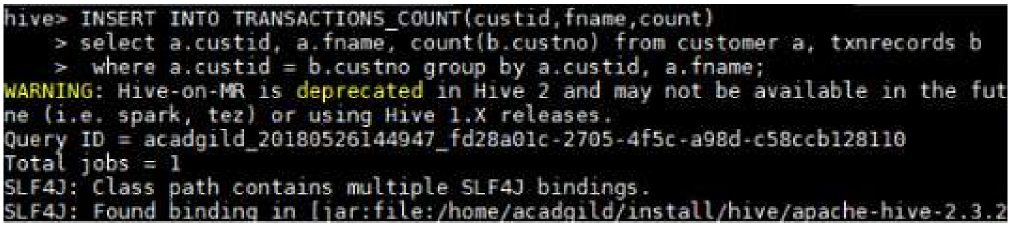


```
hive> CREATE TABLE TRANSACTIONS_COUNT( custid INT, fname STRING, count INT )
    > row format delimited fields terminated by '\t';
OK
Time taken: 0.101 seconds
hive> 
```

Task 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).

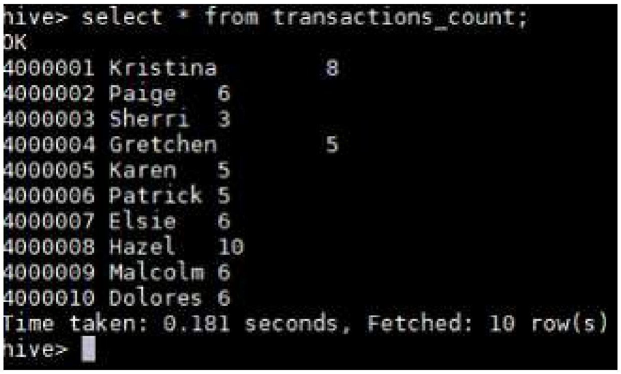
Answer:

INSERT INTO TRANSACTIONS_COUNT(custid,fname,count) select a.custid, a.fname, count(b.custno) from customer a, txnrecords b where a.custid = b.custno group by a.custid, a.fname;



```
hive> INSERT INTO TRANSACTIONS_COUNT(custid,fname,count)
    > select a.custid, a.fname, count(b.custno) from customer a, txnrecords b
    > where a.custid = b.custno group by a.custid, a.fname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the fut
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180526144947_fd28a01c-2705-4f5c-a98d-c58ccb128110
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
```

Check the records are inserted or not using SELECT command.

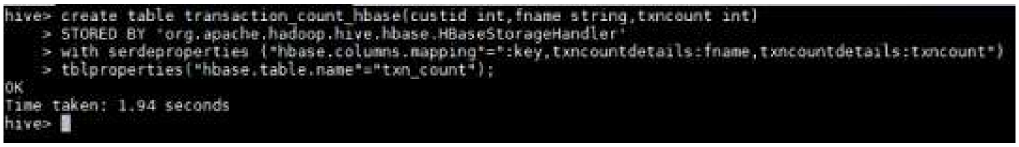


```
hive> select * from transactions_count;
OK
4000001 Kristina        8
4000002 Paige   6
4000003 Sherri  3
4000004 Gretchen        5
4000005 Karen   5
4000006 Patrick 5
4000007 Elsie   6
4000008 Hazel   10
4000009 Malcolm 6
4000010 Dolores 6
Time taken: 0.181 seconds, Fetched: 10 row(s)
hive> 
```

Task 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).

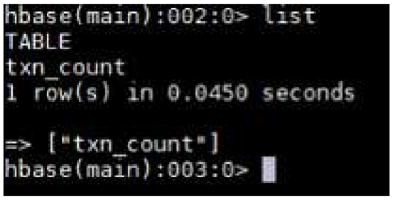
Answer:

create table transaction_count_hbase(custid int,fname string,txncount int) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' with serdeproperties ("hbase.columns.mapping"=":key,txncountdetails:fname,txncountdetails:txncount") tblproperties("hbase.table.name"="txn_count");



```
hive> create table transaction_count_hbase(custid int,fname string,txncount int)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > with serdeproperties ("hbase.columns.mapping"=":key,txncountdetails:fname,txncountdetails:txncount")
    > tblproperties("hbase.table.name"="txn_count");
OK
Time taken: 1.94 seconds
hive> 
```

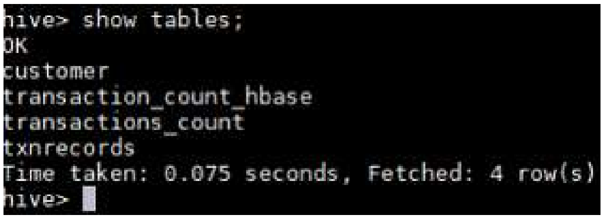
To check this we are going to list the tables in HBase shell.



```
hbase(main):002:0> list
TABLE
txn_count
1 row(s) in 0.0450 seconds

=> ["txn_count"]
hbase(main):003:0> 
```

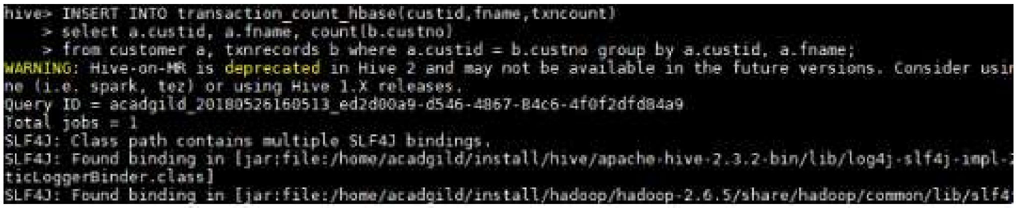
Even in HIVE we can observe the new table created –“ transaction_count_hbase”.



```
hive> show tables;
OK
customer
transaction_count_hbase
transactions_count
txnrecords
Time taken: 0.075 seconds, Fetched: 4 row(s)
hive> █
```

Task 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).
Answer:

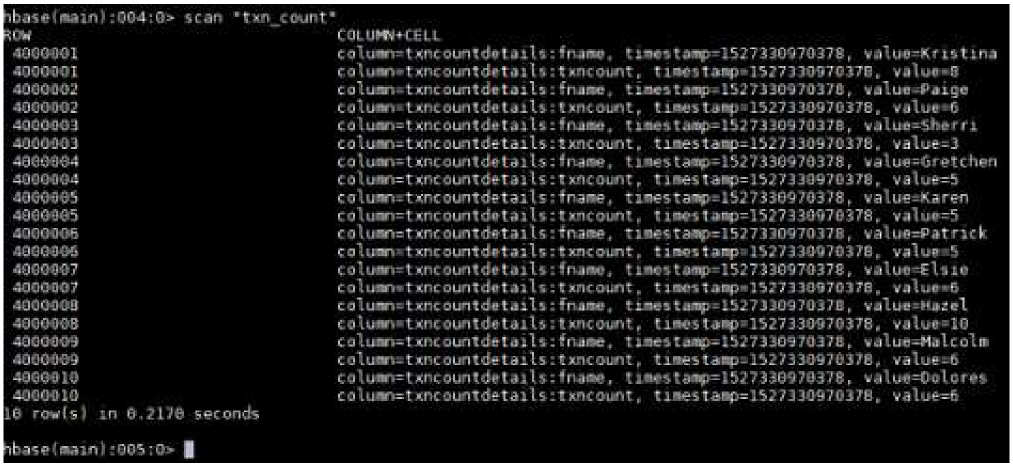
INSERT INTO transaction_count_hbase(custid,fname,txncount) select a.custid, a.fname, count(b.custno) from customer a, txnrecords b where a.custid = b.custno group by a.custid, a.fname;



```
hive> INSERT INTO transaction_count_hbase(custid,fname,txncount)
    > select a.custid, a.fname, count(b.custno)
    > from customer a, txnrecords b where a.custid = b.custno group by a.custid, a.fname;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider usin
ne (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180526160513_ed2d00a9-d546-4867-84c6-4f0f2dfd84a9
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
ticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4
```

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

scan 'txn_count'



```
hbase(main):004:0> scan 'txn_count'
ROW                                     COLUMN+CELL
 4000001                                column=txncountdetails:fname, timestamp=1527330970378, value=Kristina
 4000001                                column=txncountdetails:txncount, timestamp=1527330970378, value=8
 4000002                                column=txncountdetails:fname, timestamp=1527330970378, value=Paige
 4000002                                column=txncountdetails:txncount, timestamp=1527330970378, value=6
 4000003                                column=txncountdetails:fname, timestamp=1527330970378, value=Sherri
 4000003                                column=txncountdetails:txncount, timestamp=1527330970378, value=3
 4000004                                column=txncountdetails:fname, timestamp=1527330970378, value=Gretchen
 4000004                                column=txncountdetails:txncount, timestamp=1527330970378, value=5
 4000005                                column=txncountdetails:fname, timestamp=1527330970378, value=Karen
 4000005                                column=txncountdetails:txncount, timestamp=1527330970378, value=5
 4000006                                column=txncountdetails:fname, timestamp=1527330970378, value=Patrick
 4000006                                column=txncountdetails:txncount, timestamp=1527330970378, value=5
 4000007                                column=txncountdetails:fname, timestamp=1527330970378, value=Elsie
 4000007                                column=txncountdetails:txncount, timestamp=1527330970378, value=6
 4000008                                column=txncountdetails:fname, timestamp=1527330970378, value=Hazel
 4000008                                column=txncountdetails:txncount, timestamp=1527330970378, value=10
 4000009                                column=txncountdetails:fname, timestamp=1527330970378, value=Malcolm
 4000009                                column=txncountdetails:txncount, timestamp=1527330970378, value=6
 4000010                                column=txncountdetails:fname, timestamp=1527330970378, value=Dolores
 4000010                                column=txncountdetails:txncount, timestamp=1527330970378, value=6
10 row(s) in 0.2170 seconds

hbase(main):005:0> █
```

Here we can see the table is loaded into HBASE via HIVE shell through the use case.