Advanced Hbase

Student Name:	Subbham Vishal
Course:	Big Data Hadoop & Spark Training

Assignment –
To-do

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 :-

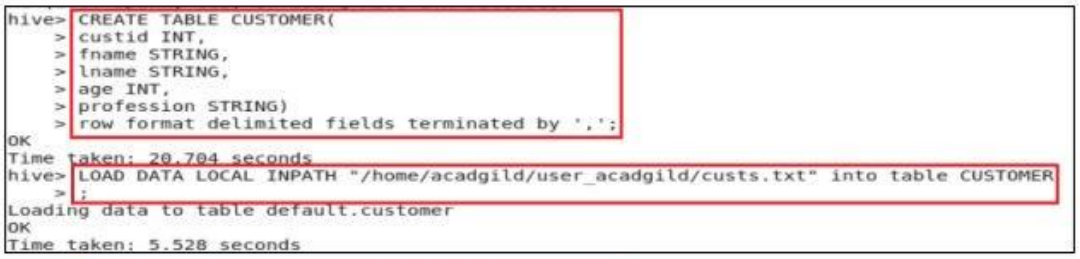
1. Find out the number of transaction done by each customer (These should be take up in module 8 itself)
2. Create a new table called TRANSACTIONS_COUNT. This table should have 3 fields - custid, fname and count. (Again to be done in module 8)
3. Now write a hive query in such a way that the query populates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

Problem Solution

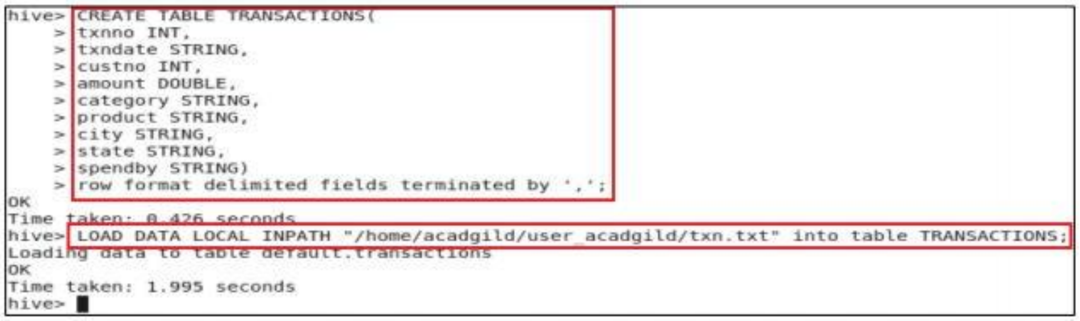
Case Study 2 – Customers and Transactions Data

The required tables CUSTOMERS and TRANSACTIONS are created using the To-do session of Hive basics session.

Creating the tables in hive.



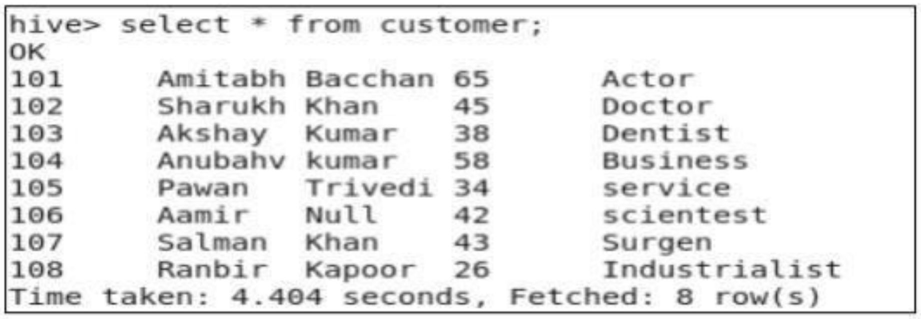
```
hive> CREATE TABLE CUSTOMER(
    > custid INT,
    > fname STRING,
    > lname STRING,
    > age INT,
    > profession STRING)
    > row format delimited fields terminated by ',';
OK
Time taken: 20.704 seconds
hive> LOAD DATA LOCAL INPATH "/home/acadgild/user_acadgild/custs.txt" into table CUSTOMER
    > ;
Loading data to table default.customer
OK
Time taken: 5.528 seconds
```



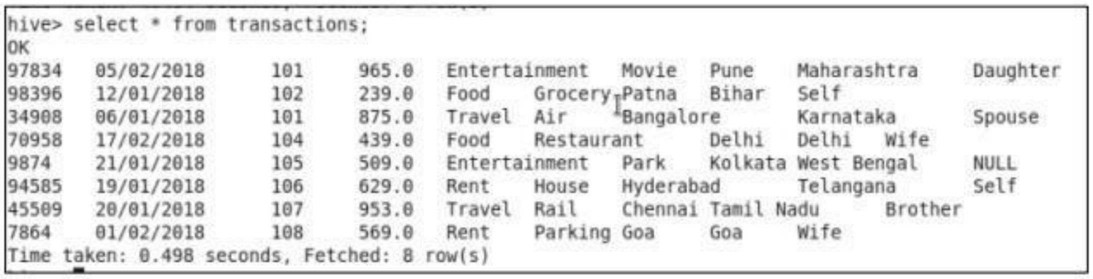
```
hive> CREATE TABLE TRANSACTIONS(
    > txnno INT,
    > txndate STRING,
    > custno INT,
    > amount DOUBLE,
    > category STRING,
    > product STRING,
    > city STRING,
    > state STRING,
    > spendby STRING)
    > row format delimited fields terminated by ',';
OK
Time taken: 0.426 seconds
hive> LOAD DATA LOCAL INPATH "/home/acadgild/user_acadgild/txn.txt" into table TRANSACTIONS;
Loading data to table default.transactions
OK
Time taken: 1.995 seconds
hive> █
```

The data from both these tables can be seen below.

The data from both these tables can be seen below.



```
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: 4.404 seconds, Fetched: 8 row(s)
```



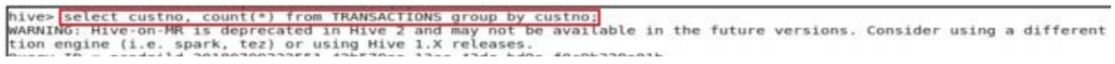
```
hive> select * from transactions;
OK
97834   05/02/2018      101     965.0   Entertainment   Movie   Pune    Maharashtra     Daughter
98396   12/01/2018      102     239.0   Food    Grocery Patna   Bihar   Self
34908   06/01/2018      101     875.0   Travel  Air     Bangalore       Karnataka       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: 0.498 seconds, Fetched: 8 row(s)
```

Objective 1:

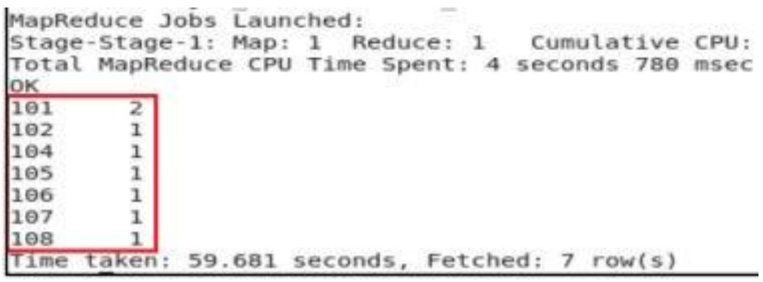
Find out the number of transaction done by each customer.

Only the custno and the number of transactions can be queries only the transactions table.

select custno, count(*) from TRANSACTIONS group by custno;



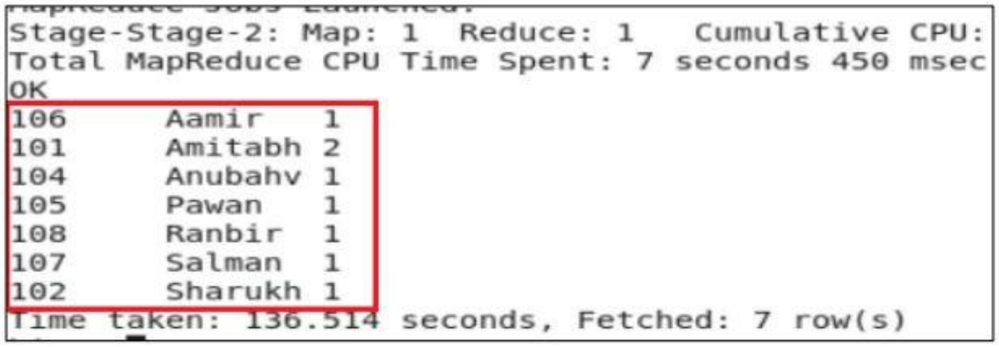
```
hive> select custno, count(*) from TRANSACTIONS group by custno;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
tion engine (i.e. spark, tez) or using Hive 1.X releases.
```



```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU:
Total MapReduce CPU Time Spent: 4 seconds 780 msec
OK
101     2
102     1
104     1
105     1
106     1
107     1
108     1
Time taken: 59.681 seconds, Fetched: 7 row(s)
```

The custid, along with the customer's first name can be queried as follows.

select t.custno,c.fname, count(*) from CUSTOMER c, TRANSACTIONS t where c.custid=t.custno group by c.fname,t.custno;



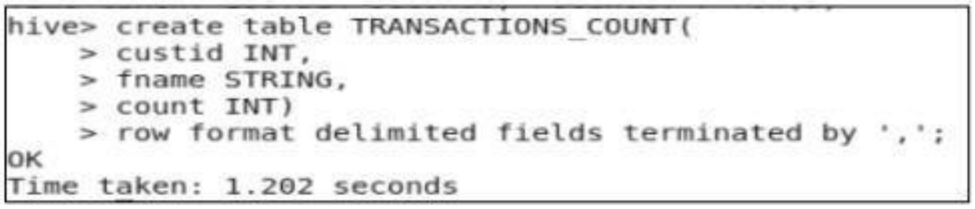
```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU:
Total MapReduce CPU Time Spent: 7 seconds 450 msec
OK
106     Aamir   1
101     Amitabh 2
104     Anubahv 1
105     Pawan   1
108     Ranbir  1
107     Salman  1
102     Sharukh 1
Time taken: 136.514 seconds, Fetched: 7 row(s)
```

The custid, customer name and the count of times the customer has occurred in the Transactions table is shown, thus showing the number of transactions per each customer.

Objective 2:

Create a new table called TRANSACTIONS_COUNT. This table should have 3 fields - custid, fname and count.

create table TRANSACTIONS_COUNT(
custid INT,
fname STRING,
count INT)
row format delimited fields terminated by ',';



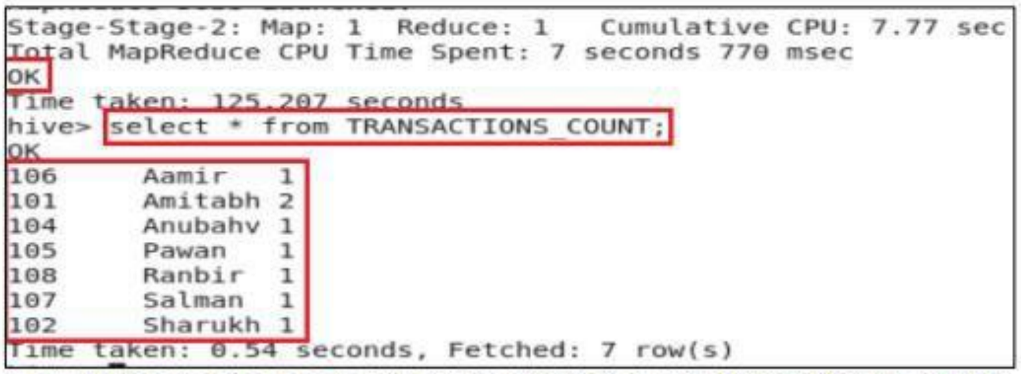
```
hive> create table TRANSACTIONS_COUNT(
    > custid INT,
    > fname STRING,
    > count INT)
    > row format delimited fields terminated by ',';
OK
Time taken: 1.202 seconds
```

The table with the given column names is created.

Objective 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.

insert overwrite table TRANSACTIONS_COUNT select t.custno,c.fname, count(*) from CUSTOMER c, TRANSACTIONS t where c.custid=t.custno group by c.fname,t.custno;



```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.77 sec
Total MapReduce CPU Time Spent: 7 seconds 770 msec
OK
Time taken: 125.207 seconds
hive> select * from TRANSACTIONS_COUNT;
OK
106     Aamir   1
101     Amitabh 2
104     Anubahv 1
105     Pawan   1
108     Ranbir  1
107     Salman  1
102     Sharukh 1
Time taken: 0.54 seconds, Fetched: 7 row(s)
```

The data from step 1 is inserted into the newly created table TRANSACTIONS_COUNT. The first line OK in the above screenshot is from when the command is run. And when queried the new table, the data is there.

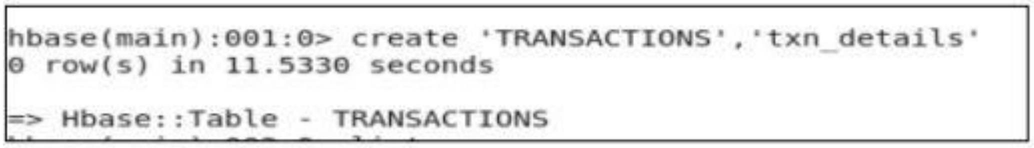
Objective 4:

Now let's make the TRANSACTIONS_COUNT table Hbase complaint. In the sense, use Ser Des and Storage handler features of hive to change the TRANSACTIONS_COUNT table to be able to create a TRANSACTIONS table in Hbase.

For a table to be Hbase compliant and to load data into a hbase table from hive, a table has to be created in HBASE. Then, that table name can be specified in the serde properties command in Hive.

Creation of table in Hbase.

create 'TRANSACTIONS','txn_details'



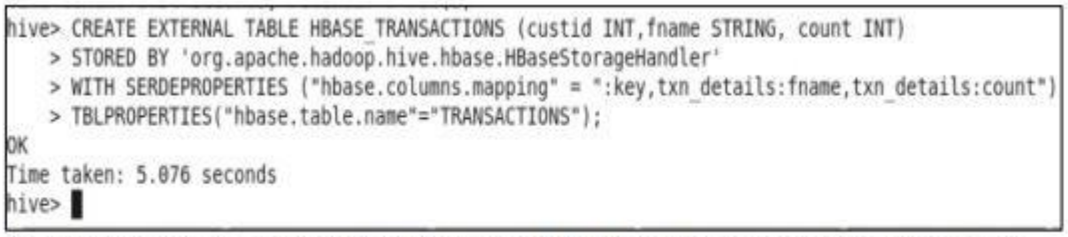
```
hbase(main):001:0> create 'TRANSACTIONS','txn_details'
0 row(s) in 11.5330 seconds

=> Hbase::Table - TRANSACTIONS
```

The table name is TRANSACTIONS and column family is txn_details. The columns from hive can be added into this column family.

Then, we have to create the Hive external table on top of HBase table that you want to populate.

CREATE EXTERNAL TABLE HBASE_TRANSACTIONS (custid INT,fname STRING, count INT)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" =
":key,txn_details:fname,txn_details:count")
TBLPROPERTIES("hbase.table.name"="transactions");



```
hive> CREATE EXTERNAL TABLE HBASE_TRANSACTIONS (custid INT,fname STRING, count INT)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,txn_details:fname,txn_details:count")
    > TBLPROPERTIES("hbase.table.name"="TRANSACTIONS");
OK
Time taken: 5.076 seconds
hive> █
```

An external table is created. The HBaseStorageHandler is used in creation of the table because the table has to be Hbase compliant.

In the serde properties, the column mappings are specified as to which column in hive table is mapped to which column in which column family in Hbase.

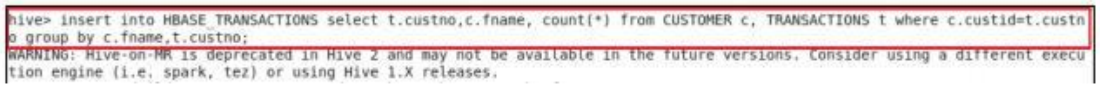
Then in the table properties, the name of the Hbase table is specified.

Objective 5:

Now insert the data in TRANSACTIONS_COUNT table using the query in step 3 again, this should populate the Hbase TRANSACTIONS table automatically.

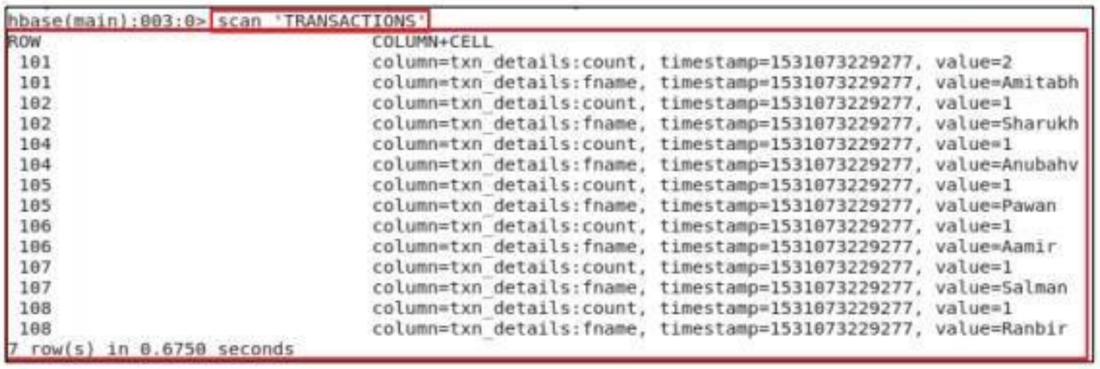
Populating the newly created external table.

> *insert into HBASE_TRANSACTIONS select t.custno,c.fname, count(*) from CUSTOMER c, TRANSACTIONS t where c.custid=t.custno group by c.fname,t.custno;*



```
hive> insert into HBASE_TRANSACTIONS select t.custno,c.fname, count(*) from CUSTOMER c, TRANSACTIONS t where c.custid=t.custn
o group by c.fname,t.custno;
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.
```

Once the command is successfully executed, the data can be seen in Hbase table too.



```
hbase(main):003:0> scan 'TRANSACTIONS'
ROW                                  COLUMN+CELL
 101                                 column=txn_details:count, timestamp=1531073229277, value=2
 101                                 column=txn_details:fname, timestamp=1531073229277, value=Amitabh
 102                                 column=txn_details:count, timestamp=1531073229277, value=1
 102                                 column=txn_details:fname, timestamp=1531073229277, value=Sharukh
 104                                 column=txn_details:count, timestamp=1531073229277, value=1
 104                                 column=txn_details:fname, timestamp=1531073229277, value=Anubahv
 105                                 column=txn_details:count, timestamp=1531073229277, value=1
 105                                 column=txn_details:fname, timestamp=1531073229277, value=Pawan
 106                                 column=txn_details:count, timestamp=1531073229277, value=1
 106                                 column=txn_details:fname, timestamp=1531073229277, value=Aamir
 107                                 column=txn_details:count, timestamp=1531073229277, value=1
 107                                 column=txn_details:fname, timestamp=1531073229277, value=Salman
 108                                 column=txn_details:count, timestamp=1531073229277, value=1
 108                                 column=txn_details:fname, timestamp=1531073229277, value=Ranbir
7 row(s) in 0.6750 seconds
```

The table contents in hbase are displayed and the 7 rows can be seen in hbase too.

So, the table in Hbase is automatically populated when the hive table is populated.

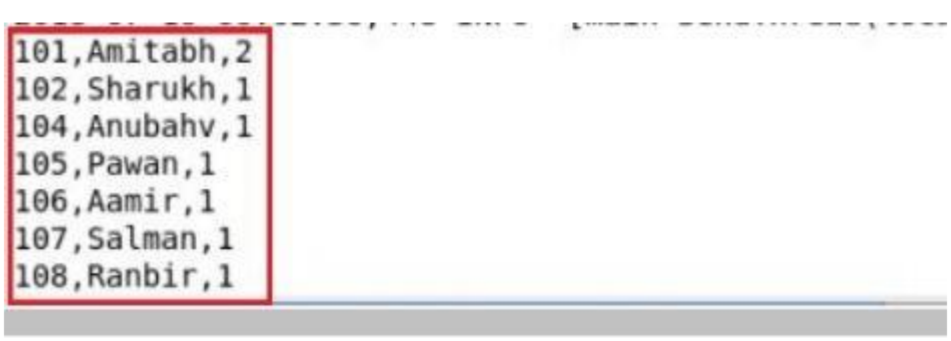
Objective 6:

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

```
package com.acadgild.cs2;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.util.Bytes;
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;

public class ScanTable {
  public static void main(String args[]) throws IOException {
    Configuration config = HBaseConfiguration.create();
    @SuppressWarnings({ "deprecation", "resource" })
    HTable table = new HTable(config, "TRANSACTIONS");
    Scan scan = new Scan();
    scan.addColumn(Bytes.toBytes("txn_details"), Bytes.toBytes("count"));
    scan.addColumn(Bytes.toBytes("txn_details"), Bytes.toBytes("fname"));
    ResultScanner scanner = table.getScanner(scan);
    for (Result result = scanner.next(); result != null; result = scanner.next()) {
      String Row = Bytes.toString(result.getRow());
      String name = Bytes.toString(result.getValue("txn_details".getBytes(),
"fname".getBytes()));
      String count = Bytes.toString(result.getValue("txn_details".getBytes(),
"count".getBytes()));
      System.out.println(Row + "," + name + "," + count);
      scanner.close();
    }
  }
}
```



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