**Customer – Transactions Case Study**

Let us take up the CUSTOMER and TRANSACTIONS table we have created in the HIVE sessions and 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,

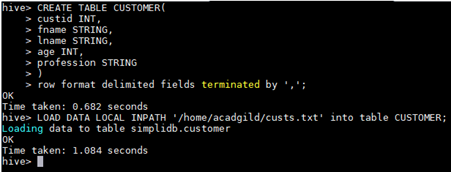
profession STRING

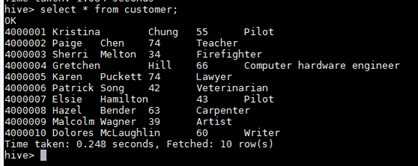
)

row format delimited fields terminated by ',';

**Loading Data:**

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

Checking whether data is loaded.

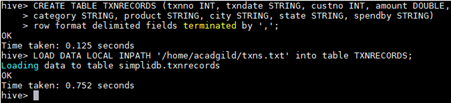
Create Transaction record table as mentioned below and load it with data.

CREATE TABLE TXNRECORDS (txnno INT, txndate STRING, custno INT, amount DOUBLE,

category STRING, product STRING, city STRING, state STRING, spendby STRING)

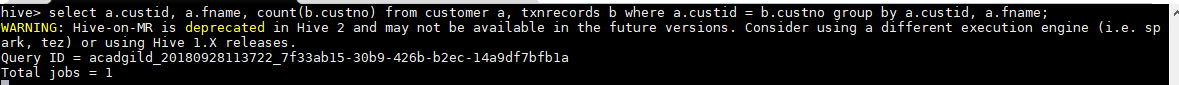
row format delimited fields terminated by ',';

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

***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;



Output

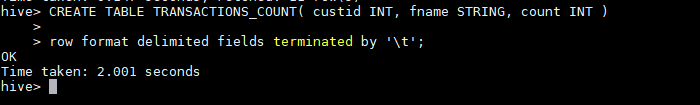


**Task 2: Create a new table called TRANSACTIONS\_COUNT. This table should have 3 fields - custid, fname and count. (Again to be done in module 8).**

***Answer:***

CREATE TABLE TRANSACTIONS\_COUNT( custid INT, fname STRING, count INT )

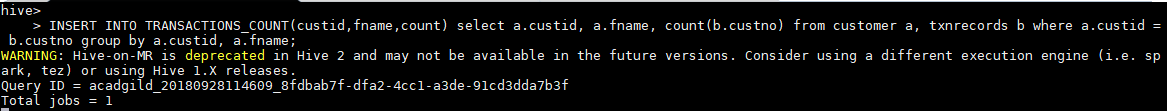
row format delimited fields terminated by '\t';



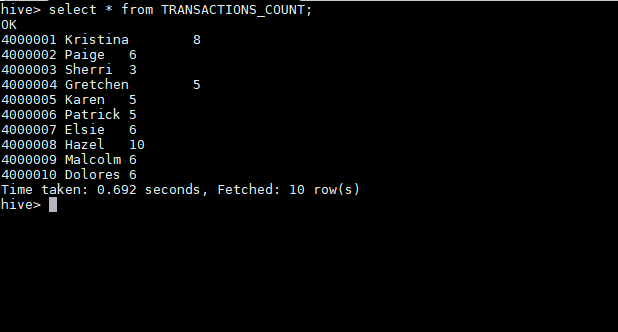
**Task 3: Now write a hive query in such a way that the query populates the data obtained in Step 1 above and populate the table in step 2 above. (This has to be done in module 9).**

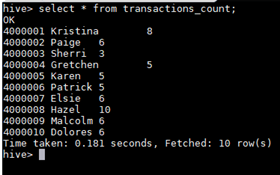
***Answer:***

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;



Check the records are inserted or not using SELECT command.

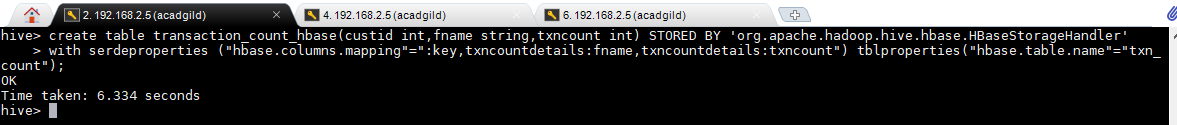


***Task 4:* Now lets make the TRANSACTIONS\_COUNT table Hbase complaint. In the sence, use Ser Des And Storate handler features of hive to change the TRANSACTIONS\_COUNT table to be able to create a TRANSACTIONS table in Hbase. (This has to be done in module 10).**

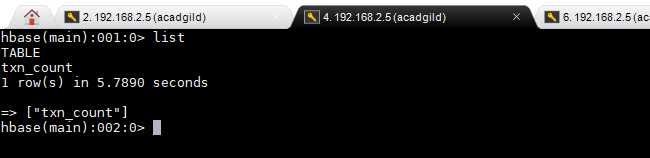
***Answer:***

create table transaction\_count\_hbase(custid int,fname string,txncount int) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

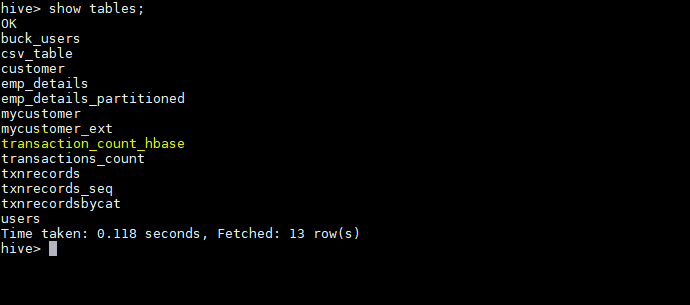
with serdeproperties ("hbase.columns.mapping"=":key,txncountdetails:fname,txncountdetails:txncount") tblproperties("hbase.table.name"="txn\_count");

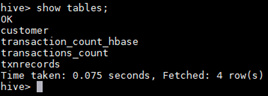


To check this we are going to list the tables in HBase shell.



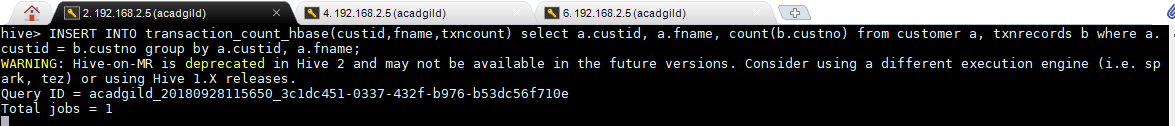
Even in HIVE we can observe the new table created –“ transaction\_count\_hbase”.

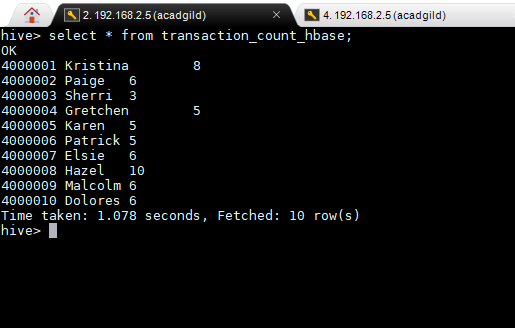


***Task 5:* Now insert the data in TRANSACTIONS\_COUNT table using the query in step 3 again, this should populate the Hbase TRANSACTIONS table automatically (This has to be done in module 10).**

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;

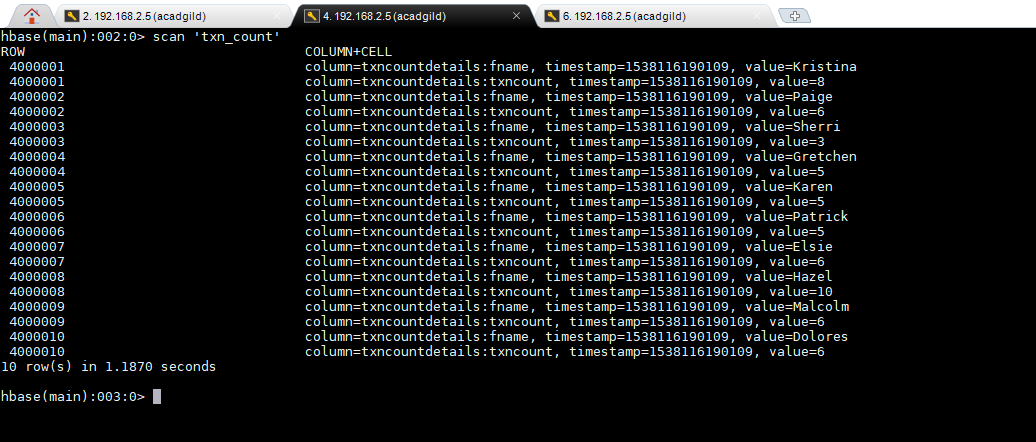




***Task 6:* Now from the Hbase level, write the Hbase java API code to access and scan the TRANSACTIONS table data from java level.**

***Answer:***

scan 'txn\_count'



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