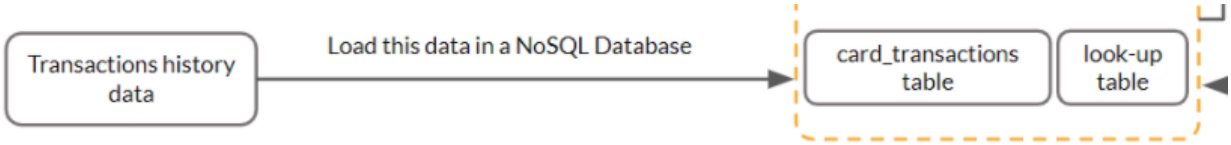
Loading Historical Transactions Data into NoSQL Database

Commands to load the past transactions data into NoSQL database

NoSQL database used : Hbase
Format of transaction history data : csv



- Created a table card_transactions and used pyspark code to load the data in the card_transactions table.

This below code is going to load all CSV into NoSQL in batch mode.

#first import happybase and create connection to happybase

```
import happybase
connection = happybase.Connection('localhost', port=9090 ,autoconnect=False)
```

#open connection to perform operations

```
def open_connection():
connection.open()
```

#close the opened connection

```
def close_connection():
connection.close()
```

#list all tables in Hbase

```
def list_tables():
print "fetching all table"
open_connection()
tables = connection.tables()
close_connection()
print "all tables fetched"
return tables
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
#create the required table
def create_table(name,cf):
print "creating table " + name
tables = list_tables()
if name not in tables:
 open_connection()
 connection.create_table(name, cf)
 close_connection()
 print "table created"
else:
 print "table already present"

#get the pointer to a table

def get_table(name):
open_connection()
table = connection.table(name)
close_connection()
return table

#batch insert data in events table

def batch_insert_data(filename,tableName):
print "starting batch insert of events"
file = open(filename, "r")
table = get_table(tableName)
open_connection()
i=0
for line in file:
 temp = line.strip().split(",")

 #Skip the first row
 if temp[0]!='card_id':
 table.put(bytes(i) , { 'info:card_id':bytes(temp[0]),
'info:member_id':bytes(temp[1]),
'info:amount':bytes(temp[2]),
'info:postcode':bytes(temp[3]),
'info:pos_id':bytes(temp[4]),
'info:transaction_dt':bytes(temp[5]),
'info:status':bytes(temp[6])})
 i=i+1

file.close()
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

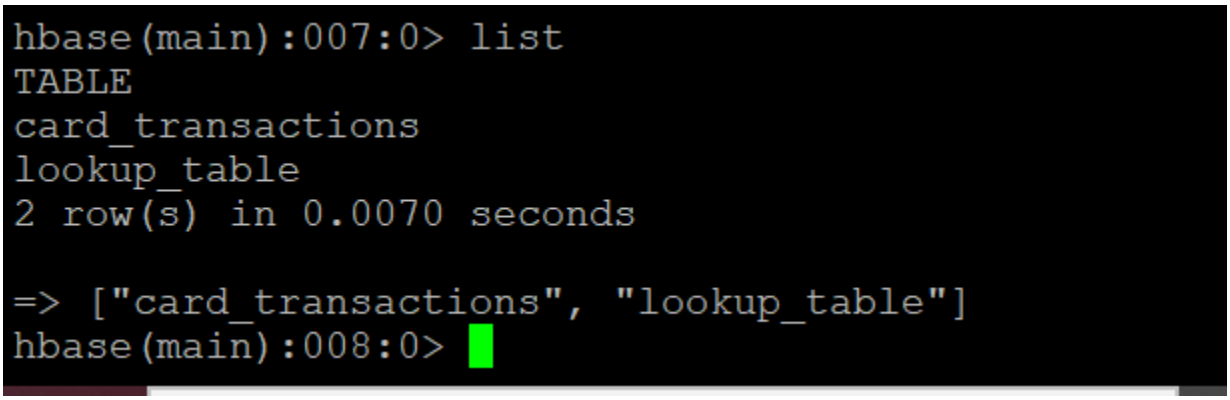
print "batch insert done"
close_connection()

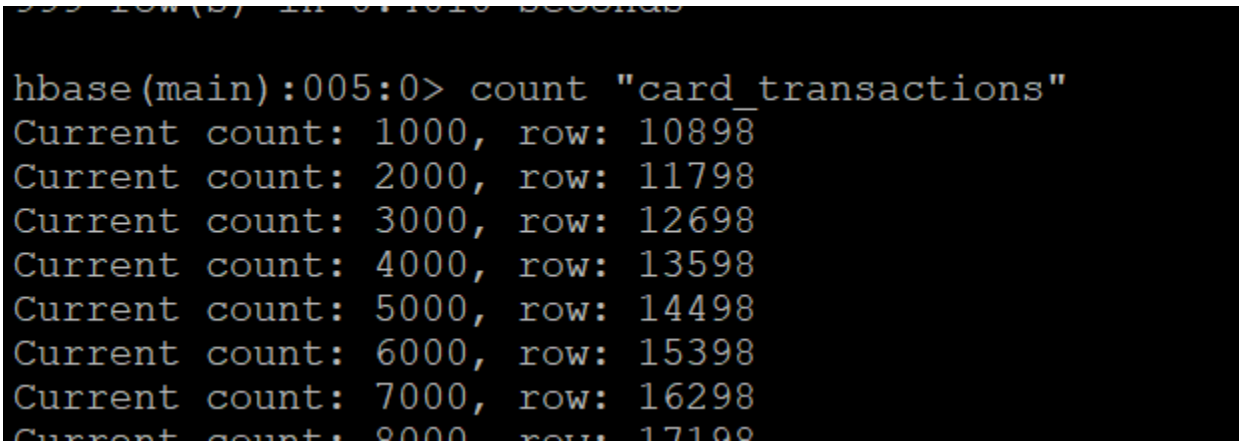
 # Batch insert data of card_transactions.csv file.
create_table('card_transactions', {'info' : dict(max_versions=5) })
batch_insert_data('card_transactions.csv','card_transactions')

Command to list the table in which the data is loaded and the command to get the count of the rows of the table

1) Login to Putty and enter as root user. (sudo -i)
2) Run thrift server using below from root user
/opt/cloudera/parcels/CDH/lib/hbase/bin/hbase-daemon.sh start thrift -p 9090
3) Give hbase shell and press enter.
4) Give "list" to see all tables in hbase

Screenshot of the table created and count





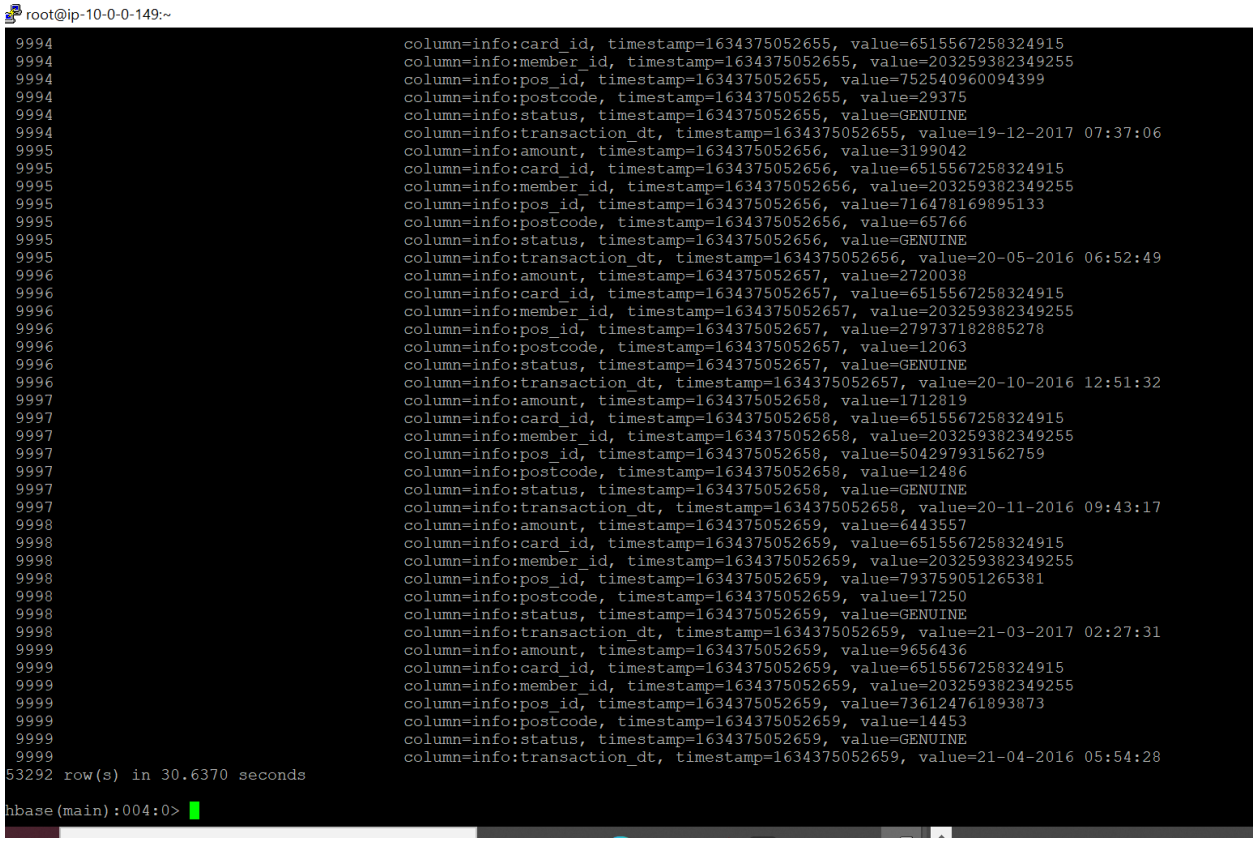
© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
Current count: 48000, row: 53198
Current count: 49000, row: 6134
Current count: 50000, row: 7034
Current count: 51000, row: 7935
Current count: 52000, row: 8835
Current count: 53000, row: 9735
53292 row(s) in 3.1270 seconds
```

```
hbase(main):006:0> count "lookup_table"
999 row(s) in 0.0400 seconds

=> 999
hbase(main):007:0>
```

scan ‘card_transactions’



© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

scan ‘lookup_table’

```
 9999                                                    column=info:t
53292 row(s) in 30.6370 seconds

hbase(main):004:0> scan "lookup_table"
```

root@ip-10-0-0-149:~

```
 6591175617713393                    column=info:transaction_date, timestamp=1634375007372, value=2018-01-31 13:10:37
 6592184145413632                    column=info:UCL, timestamp=1634375006787, value=13734342.65
 6592184145413632                    column=info:card_id, timestamp=1634375006787, value=6592184145413632
 6592184145413632                    column=info:postcode, timestamp=1634375006787, value=53186
 6592184145413632                    column=info:score, timestamp=1634375006787, value=456
 6592184145413632                    column=info:transaction_date, timestamp=1634375006787, value=2018-01-28 00:54:30
 6594248319343442                    column=info:UCL, timestamp=1634375006872, value=15065362.77
 6594248319343442                    column=info:card_id, timestamp=1634375006872, value=6594248319343442
 6594248319343442                    column=info:postcode, timestamp=1634375006872, value=24927
 6594248319343442                    column=info:score, timestamp=1634375006872, value=350
 6594248319343442                    column=info:transaction_date, timestamp=1634375006872, value=2018-01-31 23:42:38
 6595638658736751                    column=info:UCL, timestamp=1634375007621, value=14005069.97
 6595638658736751                    column=info:card_id, timestamp=1634375007621, value=6595638658736751
 6595638658736751                    column=info:postcode, timestamp=1634375007621, value=68328
 6595638658736751                    column=info:score, timestamp=1634375007621, value=310
 6595638658736751                    column=info:transaction_date, timestamp=1634375007621, value=2018-01-30 10:50:34
 6595814135833988                    column=info:UCL, timestamp=1634375007288, value=14332708.84
 6595814135833988                    column=info:card_id, timestamp=1634375007288, value=6595814135833988
 6595814135833988                    column=info:postcode, timestamp=1634375007288, value=22508
 6595814135833988                    column=info:score, timestamp=1634375007288, value=210
 6595814135833988                    column=info:transaction_date, timestamp=1634375007288, value=2018-01-30 02:03:54
 6595928469079750                    column=info:UCL, timestamp=1634375008323, value=11824730.01
 6595928469079750                    column=info:card_id, timestamp=1634375008323, value=6595928469079750
 6595928469079750                    column=info:postcode, timestamp=1634375008323, value=98349
 6595928469079750                    column=info:score, timestamp=1634375008323, value=412
 6595928469079750                    column=info:transaction_date, timestamp=1634375008323, value=2018-01-24 12:38:22
 6597703848279563                    column=info:UCL, timestamp=1634375007681, value=15250624.49
 6597703848279563                    column=info:card_id, timestamp=1634375007681, value=6597703848279563
 6597703848279563                    column=info:postcode, timestamp=1634375007681, value=95699
 6597703848279563                    column=info:score, timestamp=1634375007681, value=218
 6597703848279563                    column=info:transaction_date, timestamp=1634375007681, value=2018-01-27 10:51:49
 6598830758632447                    column=info:UCL, timestamp=1634375007878, value=12685782.48
 6598830758632447                    column=info:card_id, timestamp=1634375007878, value=6598830758632447
 6598830758632447                    column=info:postcode, timestamp=1634375007878, value=19421
 6598830758632447                    column=info:score, timestamp=1634375007878, value=293
 6598830758632447                    column=info:transaction_date, timestamp=1634375007878, value=2018-01-30 00:18:34
 6599900931314251                    column=info:UCL, timestamp=1634375008288, value=12487392.07
 6599900931314251                    column=info:card_id, timestamp=1634375008288, value=6599900931314251
 6599900931314251                    column=info:postcode, timestamp=1634375008288, value=97423
 6599900931314251                    column=info:score, timestamp=1634375008288, value=297
 6599900931314251                    column=info:transaction_date, timestamp=1634375008288, value=2018-01-31 11:25:16
999 row(s) in 0.4010 seconds

hbase(main):005:0>
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved