Lab: Phoenix – Populating Data using Pig

About This Lab

Objective:	The *StoreFunc* allows users to write data in Phoenix-encoded format to HBase tables using Pig scripts. This is a nice way to bulk upload data from a *MapReduce* job in parallel to a Phoenix table in HBase.
File locations:	N/A
Successful outcome:	You will: Connect to your lab environment; log in and use pig to populate data to phoenix using MapReduce
Before you begin:	Start and connect to your classroom lab environment, executed lab 1
Related lesson:	Phoenix Integrations

Data population Using StoreFunc:

1. Retrieve and build datasets, then switch user to `maria_dev` to load both into HDFS:

```
[root@sandbox datasets]# wget
https://raw.githubusercontent.com/HortonworksUniversity/RealTime_Labs/master/da
tasets/phoenix/pig/testdata
[root@sandbox datasets]# echo -e
"1\tJohn\n2\tGeorge\n3\tPeter\n4\tJose\n5\tJoe\n6\tKrish" >> dummy
[root@sandbox datasets]# cat dummy
1       John
2       George
3       Peter
4       Jose
5       Joe
6       Krish
[root@sandbox datasets]# su - maria_dev
[maria_dev@sandbox ~]$ cd /root/rtlabs/datasets/
[maria_dev@sandbox datasets]$ hdfs dfs -put dummy
[maria_dev@sandbox datasets]$ hdfs dfs -put testdata
[maria_dev@sandbox datasets]$
```

2. Login to Phoenix command line and create tables to store the data:

# /usr/hdp/current/phoenix-client/bin/sqlline.py localhost:2181:/hbase-unsecure

CREATE TABLE TESTPHX(A VARCHAR NOT NULL, B VARCHAR, C INTEGER, D VARCHAR, E VARCHAR CONSTRAINT pk PRIMARY KEY(A));

CREATE TABLE DUMMY(ID VARCHAR NOT NULL, Name VARCHAR CONSTRAINT pk PRIMARY KEY(ID));

```
[maria_dev@sandbox datasets]$ /usr/hdp/current/phoenix-client/bin/sqlline.py
localhost:2181:/hbase-unsecure
Setting property: [incremental, false]
Setting property: [isolation, TRANSACTION_READ_COMMITTED]
issuing: !connect jdbc:phoenix:localhost:2181:/hbase-unsecure none none
org.apache.phoenix.jdbc.PhoenixDriver
Connecting to jdbc:phoenix:localhost:2181:/hbase-unsecure
Connected to: Phoenix (version 4.7)
Driver: PhoenixEmbeddedDriver (version 4.7)
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED
Building list of tables and columns for tab-completion (set fastconnect to true
to skip)...
```

```
91/91 (100%) Done
Done
sqlline version 1.1.8
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> CREATE TABLE TESTPHX( A VARCHAR
NOT NULL, B VARCHAR, C INTEGER, D VARCHAR, E VARCHAR CONSTRAINT pk PRIMARY
KEY(A));
No rows affected (1.286 seconds)
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> CREATE TABLE DUMMY( ID VARCHAR
NOT NULL, Name VARCHAR CONSTRAINT pk PRIMARY KEY(ID));
No rows affected (1.237 seconds)
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> !quit
Closing: org.apache.phoenix.jdbc.PhoenixConnection
[maria_dev@sandbox datasets]$
```

3. Change filesystem location to home directory and create pig scripts which contains commands to generate mapreduce jobs to populate data:

# cd ~
# vi testphx.pig

add below:

register /usr/hdp/current/phoenix-client/phoenix-client.jar
register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
A = load 'testdata' as (a:chararray, b:chararray, c:int, d:chararray, e: chararray);
STORE A into 'hbase://TESTPHX' using
org.apache.phoenix.pig.PhoenixHBaseStorage('localhost:/hbase-unsecure','-batchSize 1000');

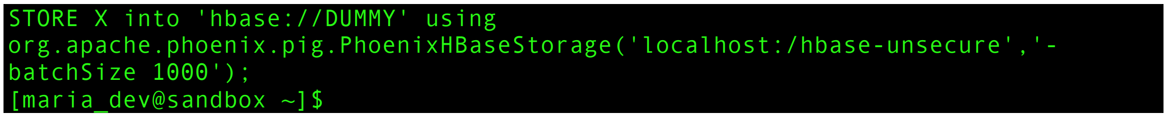
```
[maria_dev@sandbox datasets]$ cd ~
[maria_dev@sandbox ~]$ pwd
/home/maria_dev
[maria_dev@sandbox ~]$ vi testphx.pig
[maria_dev@sandbox ~]$ cat testphx.pig
register /usr/hdp/current/phoenix-client/phoenix-client.jar
register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
A = load 'testdata' as (a:chararray, b:chararray, c:int, d:chararray,
e:chararray);
STORE A into 'hbase://TESTPHX' using
org.apache.phoenix.pig.PhoenixHBaseStorage('localhost:/hbase-unsecure','-
batchSize 1000');
[maria_dev@sandbox ~]$
```

# vi dummyphx.pig

add below:

register /usr/hdp/current/phoenix-client/phoenix-client.jar
register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
X = load 'dummy' as (id:chararray, name:chararray);
STORE X into 'hbase://DUMMY' using
org.apache.phoenix.pig.PhoenixHBaseStorage('localhost:/hbase-unsecure','-batchSize 1000');

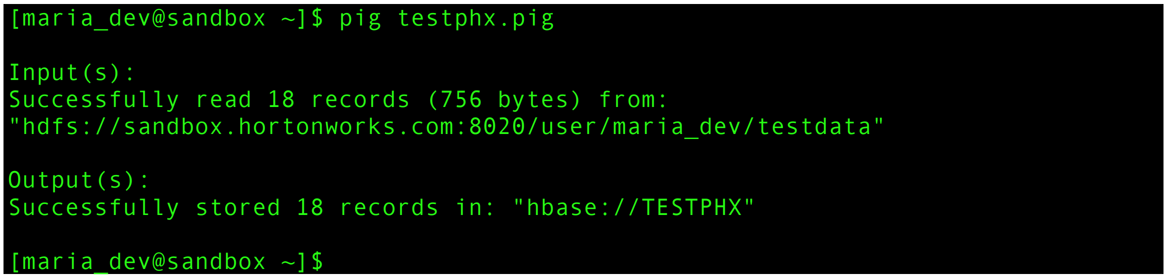
```
[maria_dev@sandbox ~]$ vi dummyphx.pig
[maria_dev@sandbox ~]$ cat dummyphx.pig
register /usr/hdp/current/phoenix-client/phoenix-client.jar
register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
X = load 'dummy' as (id:chararray, name:chararray);
```



```
STORE X into 'hbase://DUMMY' using
org.apache.phoenix.pig.PhoenixHBaseStorage('localhost:/hbase-unsecure','-
batchSize 1000');
[maria_dev@sandbox ~]$
```

4. Below command to load data to table TESTPHX will generate a Mapreduce and populate data accordingly. Once done we can Verify the data from phoenix command line. You can verify the same from hbase as well. Execute below command:

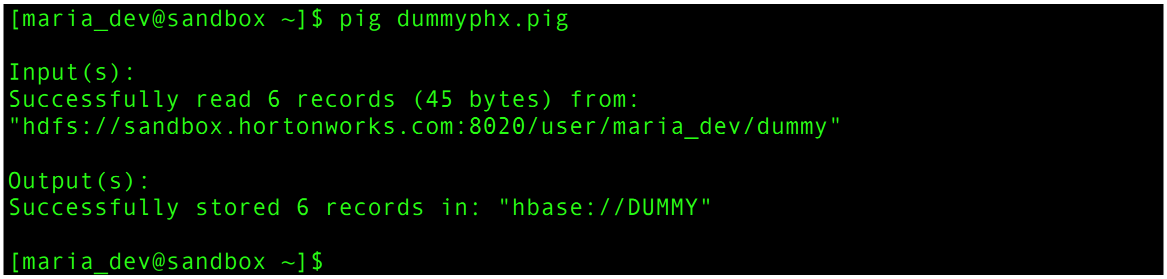
   # **pig testphx.pig**



```
[maria_dev@sandbox ~]$ pig testphx.pig

Input(s):
Successfully read 18 records (756 bytes) from:
"hdfs://sandbox.hortonworks.com:8020/user/maria_dev/testdata"

Output(s):
Successfully stored 18 records in: "hbase://TESTPHX"

[maria_dev@sandbox ~]$
```

5. Execute below command to load data into second table DUMMY

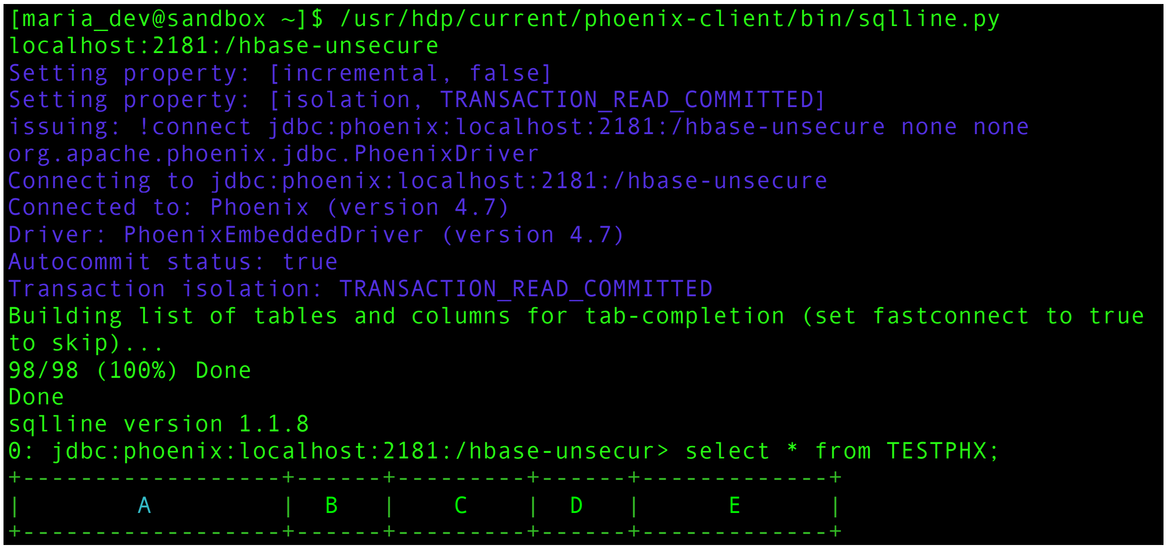
   # **pig dummyphx.pig**



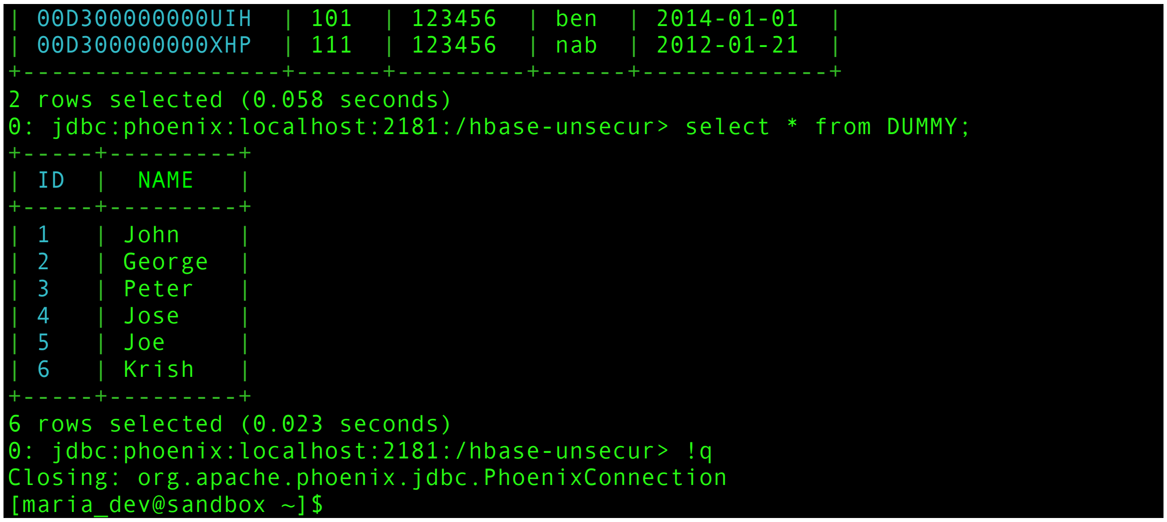
```
[maria_dev@sandbox ~]$ pig dummyphx.pig

Input(s):
Successfully read 6 records (45 bytes) from:
"hdfs://sandbox.hortonworks.com:8020/user/maria_dev/dummy"

Output(s):
Successfully stored 6 records in: "hbase://DUMMY"

[maria_dev@sandbox ~]$
```

6. Lets go ahead and verify the tables created and its data availability in phoenix:

# */usr/hdp/current/phoenix-client/bin/sqlline.py localhost:2181:/hbase-unsecure*



```
[maria_dev@sandbox ~]$ /usr/hdp/current/phoenix-client/bin/sqlline.py
localhost:2181:/hbase-unsecure
Setting property: [incremental, false]
Setting property: [isolation, TRANSACTION_READ_COMMITTED]
issuing: !connect jdbc:phoenix:localhost:2181:/hbase-unsecure none none
org.apache.phoenix.jdbc.PhoenixDriver
Connecting to jdbc:phoenix:localhost:2181:/hbase-unsecure
Connected to: Phoenix (version 4.7)
Driver: PhoenixEmbeddedDriver (version 4.7)
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED
Building list of tables and columns for tab-completion (set fastconnect to true
to skip)...
98/98 (100%) Done
Done
sqlline version 1.1.8
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> select * from TESTPHX;
+------------------+------+---------+------+-------------+
|        A         |  B   |    C    |  D   |      E      |
+------------------+------+---------+------+-------------+
```



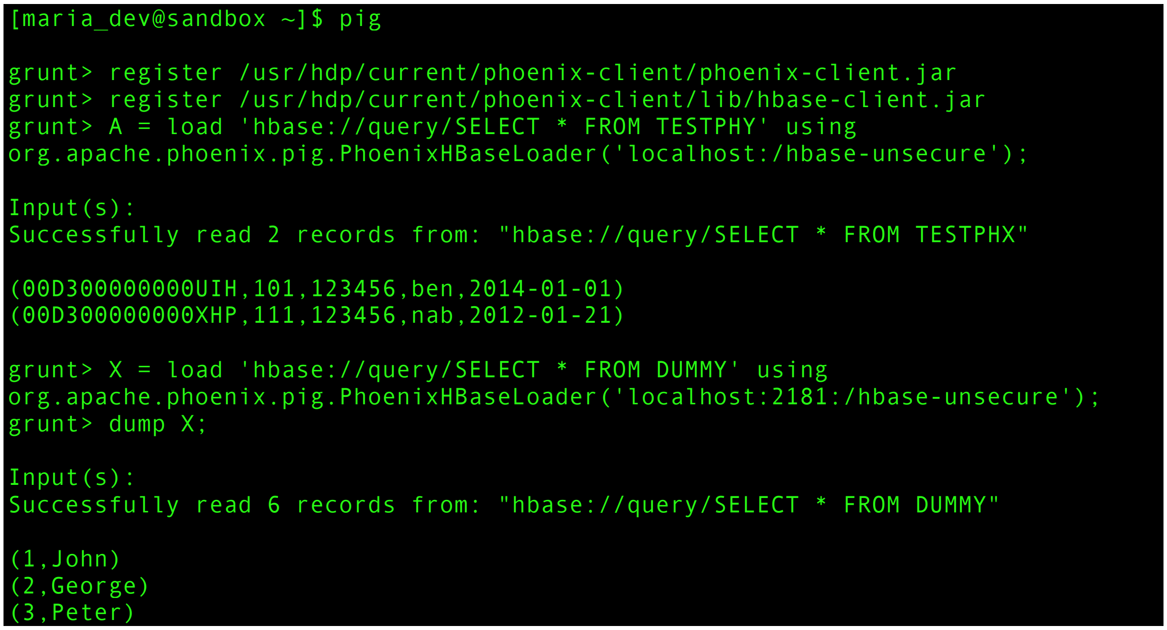
```
| 00D300000000UIH  | 101  | 123456  | ben  | 2014-01-01  |
| 00D300000000XHP  | 111  | 123456  | nab  | 2012-01-21  |
+------------------+------+---------+------+-------------+
2 rows selected (0.058 seconds)
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> select * from DUMMY;
+-----+---------+
| ID  |  NAME   |
+-----+---------+
| 1   | John    |
| 2   | George  |
| 3   | Peter   |
| 4   | Jose    |
| 5   | Joe     |
| 6   | Krish   |
+-----+---------+
6 rows selected (0.023 seconds)
0: jdbc:phoenix:localhost:2181:/hbase-unsecur> !q
Closing: org.apache.phoenix.jdbc.PhoenixConnection
[maria_dev@sandbox ~]$
```

Data retrieval Using PigLoader

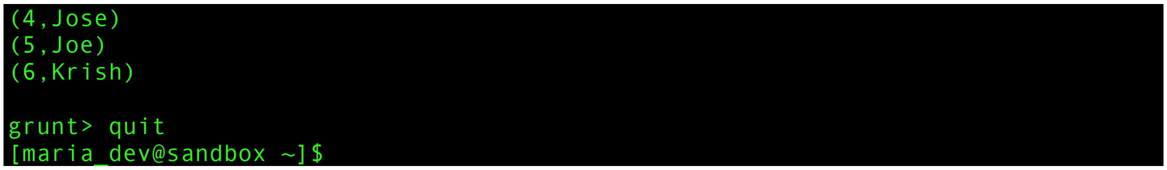
1. A Pig data loader allows users to read data from Phoenix backed HBase tables within a Pig script. Launch pig grunt shell from the command line and execute the below commands:

pig →

register /usr/hdp/current/phoenix-client/phoenix-client.jar
register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
*A = load 'hbase://query/SELECT * FROM TESTPHX' using*
org.apache.phoenix.pig.PhoenixHBaseLoader('localhost:/hbase-unsecure');
dump A;
*X = load 'hbase://query/SELECT * FROM DUMMY' using*
org.apache.phoenix.pig.PhoenixHBaseLoader('localhost:/hbase-unsecure');
dump X;



```
[maria_dev@sandbox ~]$ pig

grunt> register /usr/hdp/current/phoenix-client/phoenix-client.jar
grunt> register /usr/hdp/current/phoenix-client/lib/hbase-client.jar
grunt> A = load 'hbase://query/SELECT * FROM TESTPHY' using
org.apache.phoenix.pig.PhoenixHBaseLoader('localhost:/hbase-unsecure');

Input(s):
Successfully read 2 records from: "hbase://query/SELECT * FROM TESTPHX"

(00D300000000UIH,101,123456,ben,2014-01-01)
(00D300000000XHP,111,123456,nab,2012-01-21)

grunt> X = load 'hbase://query/SELECT * FROM DUMMY' using
org.apache.phoenix.pig.PhoenixHBaseLoader('localhost:2181:/hbase-unsecure');
grunt> dump X;

Input(s):
Successfully read 6 records from: "hbase://query/SELECT * FROM DUMMY"

(1,John)
(2,George)
(3,Peter)
```



```
(4,Jose)
(5,Joe)
(6,Krish)

grunt> quit
[maria_dev@sandbox ~]$
```

Result

Phoenix tables are populated with help of pig StorFunc and Pigloader is used to read the data from tables.