Project - 2
Music Data Analysis

- **Below steps will be performed in sequence,**

  - **Step 1: Need to Launch required daemons**
  - **Step 2: Performing Job Scheduling**
  - **Step 3: Populating LookUp tables**
  - **Step 4: Performing Data Formatting**
  - **Step 5: Performing Data Enrichment and Cleaning**
  - **Step 6: Performing Data Analysis and solutions**

Step 1: Need to Launch required daemons

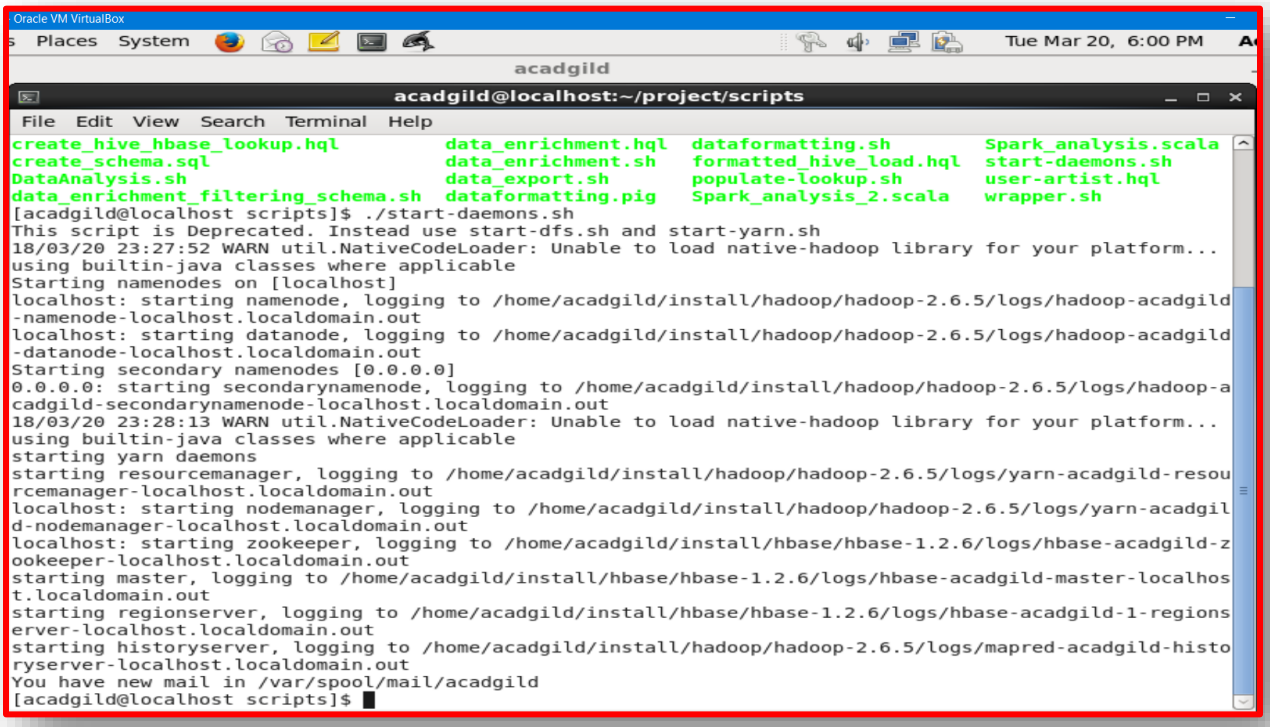
- **Navigating to the Scripts folder and listing all scripts**

```
[acadgild@localhost scripts]$ cd /home/acadgild/project/scripts
[acadgild@localhost scripts]$ ls
create_hive_hbase_lookup.hql         data_enrichment.hql
dataformatting.sh        Spark_analysis.scala
create_schema.sql                    data_enrichment.sh
formatted_hive_load.hql  start-daemons.sh
DataAnalysis.sh                      data_export.sh       populate-
lookup.sh       user-artist.hql
data_enrichment_filtering_schema.sh  dataformatting.pig
Spark_analysis_2.scala   wrapper.sh
```

- **Running start-daemons.sh script to launch required daemons**

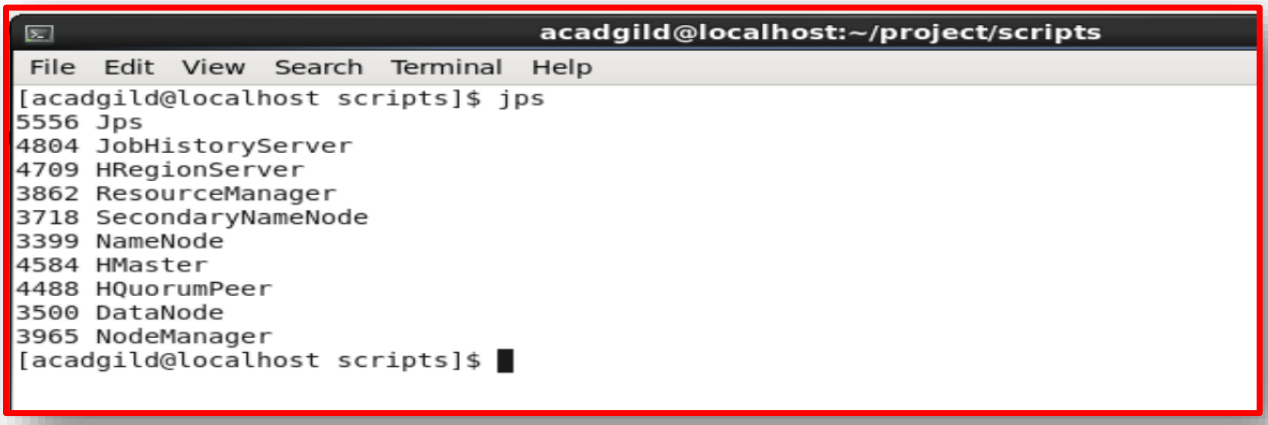
```
[acadgild@localhost scripts]$ ./start-daemons.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
18/03/20 23:27:52 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-
namenode-localhost.localdomain.out
localhost: starting datanode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-
datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-
secondarynamenode-localhost.localdomain.out
18/03/20 23:28:13 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
starting yarn daemons
starting resourcemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-
resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-
nodemanager-localhost.localdomain.out
localhost: starting zookeeper, logging to
/home/acadgild/install/hbase/hbase-1.2.6/logs/hbase-acadgild-
zookeeper-localhost.localdomain.out
starting master, logging to /home/acadgild/install/hbase/hbase-
1.2.6/logs/hbase-acadgild-master-localhost.localdomain.out
starting regionserver, logging to /home/acadgild/install/hbase/hbase-
1.2.6/logs/hbase-acadgild-1-regionserver-localhost.localdomain.out
```

```
starting historyserver, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/mapred-acadgild-
historyserver-localhost.localdomain.out
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```



■ Checking the running daemons with jps command

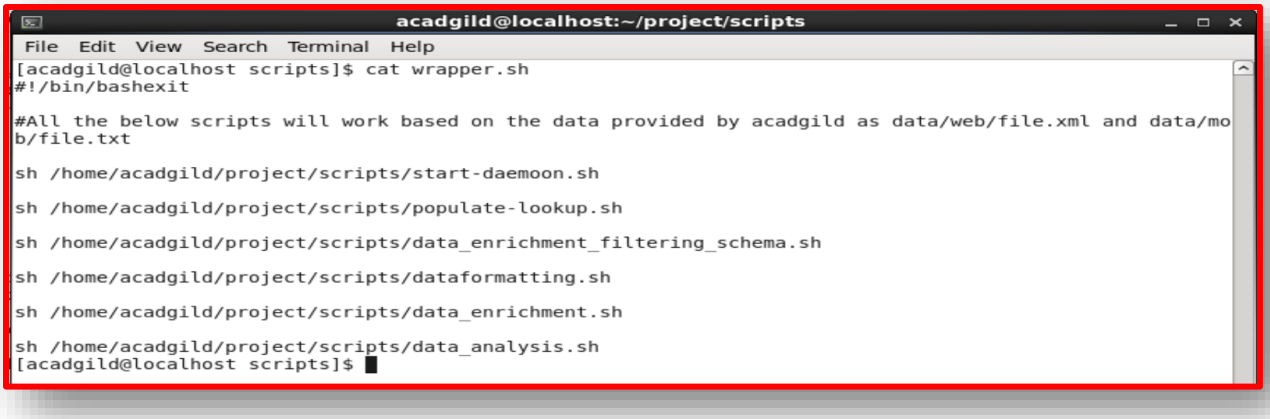
```
[acadgild@localhost scripts]$ jps
4804 JobHistoryServer
4709 HRegionServer
3862 ResourceManager
3718 SecondaryNameNode
3399 NameNode
5495 Jps
4584 HMaster
4488 HQuorumPeer
3500 DataNode
3965 NodeManager
[acadgild@localhost scripts]$
```


acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
[acadgild@localhost scripts]$ jps
5556 Jps
4804 JobHistoryServer
4709 HRegionServer
3862 ResourceManager
3718 SecondaryNameNode
3399 NameNode
4584 HMaster
4488 HQuorumPeer
3500 DataNode
3965 NodeManager
[acadgild@localhost scripts]$

Step 2: Performing Job Scheduling

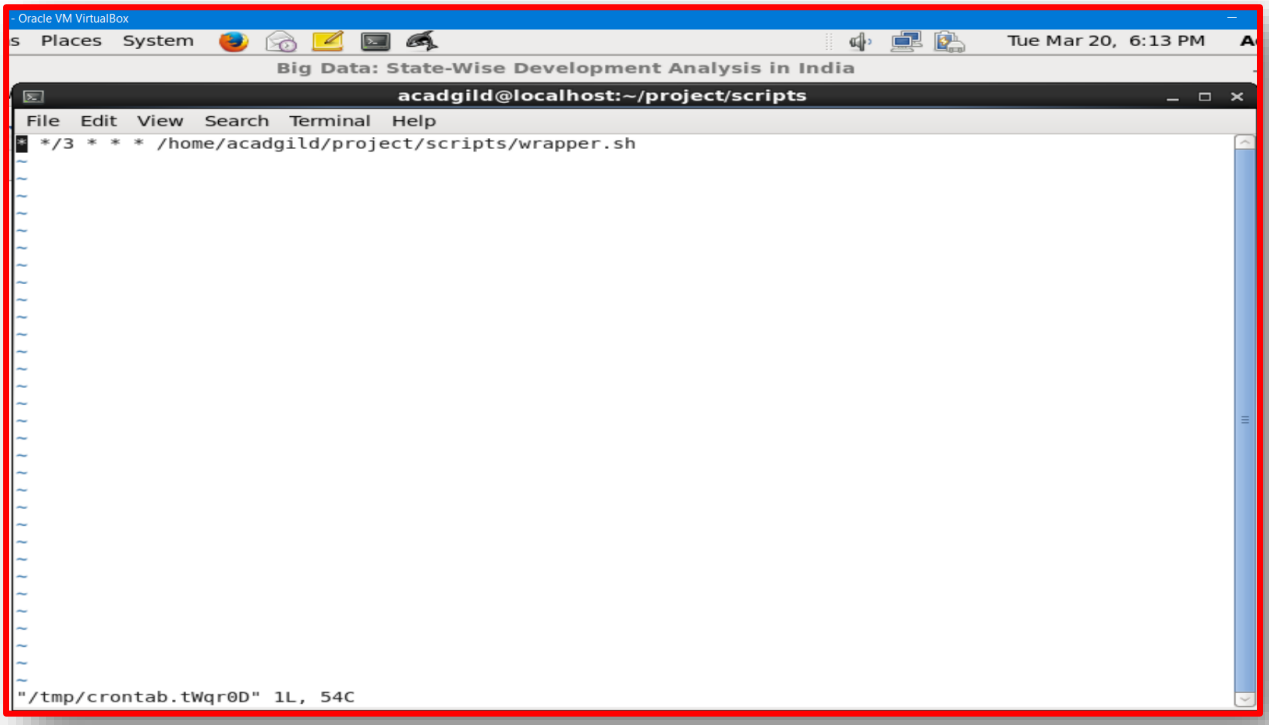
- **The shell script wrapper.sh has all the scripts for scheduling a job**

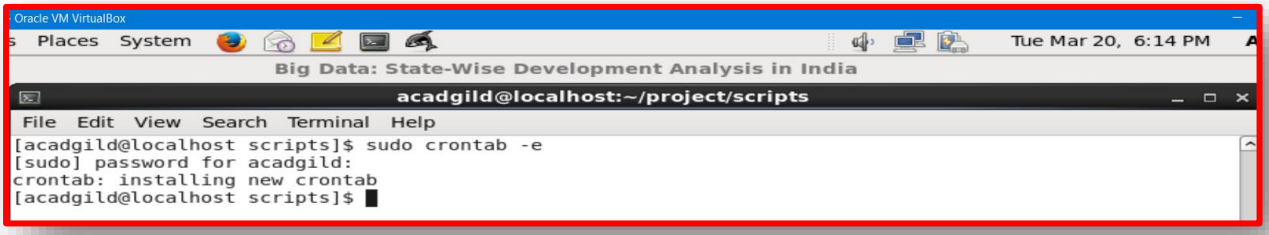
```
[acadgild@localhost scripts]$ cat wrapper.sh
#!/bin/bashexit

#All the below scripts will work based on the data provided by
acadgild as data/web/file.xml and data/mob/file.txt

sh /home/acadgild/project/scripts/start-daemoon.sh

sh /home/acadgild/project/scripts/populate-lookup.sh

sh /home/acadgild/project/scripts/data_enrichment_filtering_schema.sh

sh /home/acadgild/project/scripts/dataformatting.sh

sh /home/acadgild/project/scripts/data_enrichment.sh

sh /home/acadgild/project/scripts/data_analysis.sh
[acadgild@localhost scripts]$
```



- **Using crontab to schedule a Job in the -e mode. We have scheduled this job for every 3 Hours.**

```
[acadgild@localhost scripts]$ sudo crontab -e
[sudo] password for acadgild:
crontab: installing new crontab
[acadgild@localhost scripts]$
```

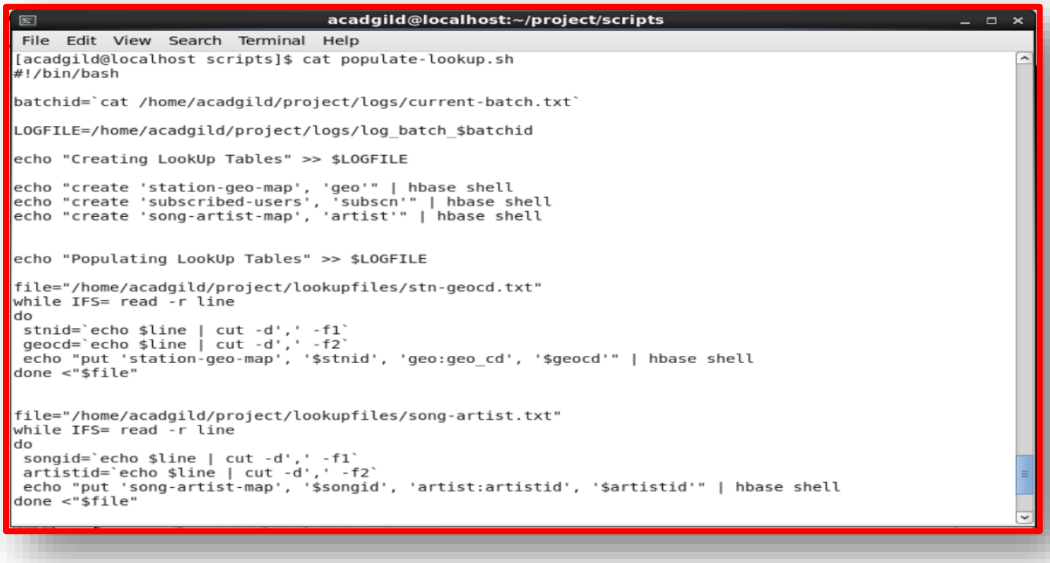




Step 3: Populating LookUp tables

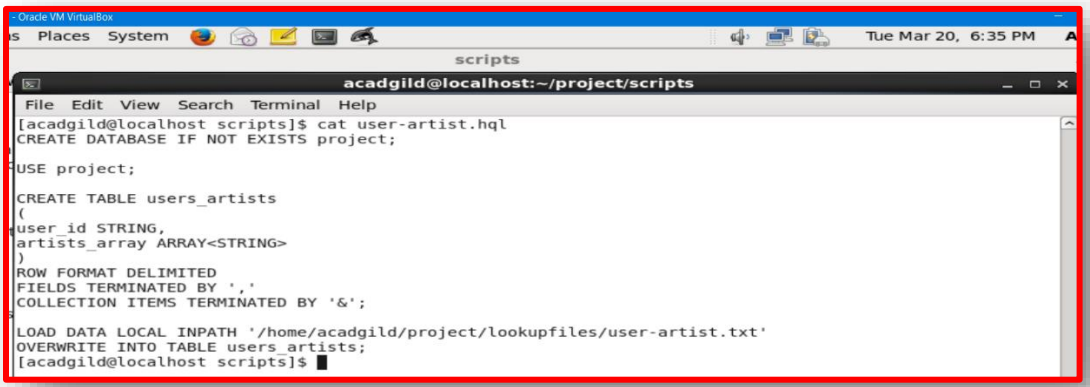
- **Displaying the script populate-lookup.sh which creates tables in hbase and hive**

```
[acadgild@localhost scripts]$ cat populate-lookup.sh
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`

LOGFILE=/home/acadgild/project/logs/log_batch_$batchid

echo "Creating LookUp Tables" >> $LOGFILE

echo "create 'station-geo-map', 'geo'" | hbase shell
echo "create 'subscribed-users', 'subscn'" | hbase shell
echo "create 'song-artist-map', 'artist'" | hbase shell


echo "Populating LookUp Tables" >> $LOGFILE

file="/home/acadgild/project/lookupfiles/stn-geocd.txt"
while IFS= read -r line
do
 stnid=`echo $line | cut -d',' -f1`
 geocd=`echo $line | cut -d',' -f2`
 echo "put 'station-geo-map', '$stnid', 'geo:geo_cd', '$geocd'" |
hbase shell
done <"$file"


file="/home/acadgild/project/lookupfiles/song-artist.txt"
while IFS= read -r line
do
 songid=`echo $line | cut -d',' -f1`
 artistid=`echo $line | cut -d',' -f2`
 echo "put 'song-artist-map', '$songid', 'artist:artistid',
'$artistid'" | hbase shell
done <"$file"


file="/home/acadgild/project/lookupfiles/user-subscn.txt"
while IFS= read -r line
do
 userid=`echo $line | cut -d',' -f1`
 startdt=`echo $line | cut -d',' -f2`
 enddt=`echo $line | cut -d',' -f3`
 echo "put 'subscribed-users', '$userid', 'subscn:startdt',
'$startdt'" | hbase shell
 echo "put 'subscribed-users', '$userid', 'subscn:enddt', '$enddt'" |
hbase shell
done <"$file"

hive -f /home/acadgild/project/scripts/user-artist.hql

[acadgild@localhost scripts]$
```



- **Displaying the script user_artist.hql which creates tables in hive**

```
[acadgild@localhost scripts]$ cat user-artist.hql
CREATE DATABASE IF NOT EXISTS project;

USE project;

CREATE TABLE users_artists
(
user_id STRING,
artists_array ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '&';

LOAD DATA LOCAL INPATH '/home/acadgild/project/lookupfiles/user-
artist.txt'
OVERWRITE INTO TABLE users_artists;
[acadgild@localhost scripts]$
```



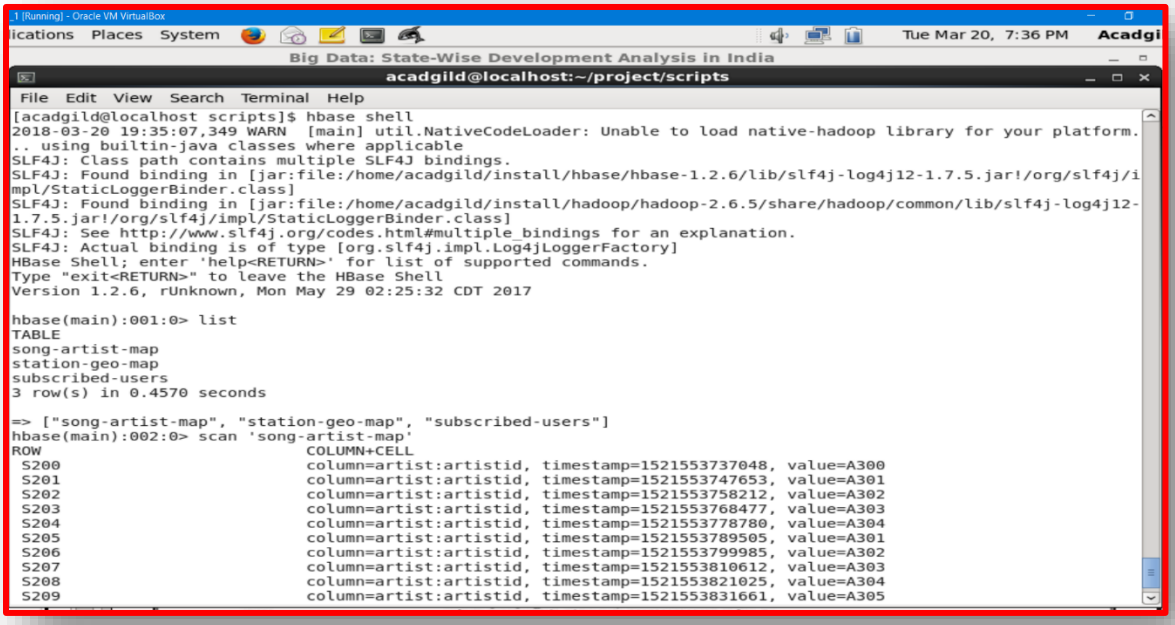
■ Running populate-lookup.sh which performs the table creation activity in hbase and hive

```
[acadgild@localhost scripts]$ ./populate-lookup.sh
2018-03-20 19:25:39,612 WARN  [main] util.NativeCodeLoader: Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-
1.2.6/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

put 'subscribed-users', 'U114', 'subscn:enddt', '1468130523'
0 row(s) in 0.6450 seconds

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-
2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/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]

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-
common-2.3.2.jar!/hive-log4j2.properties Async: true
OK
Time taken: 9.291 seconds
OK
Time taken: 0.028 seconds
OK
Time taken: 0.899 seconds
Loading data to table project.users_artists
OK
Time taken: 2.085 seconds
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```

■ Verifying the tables created in HBase

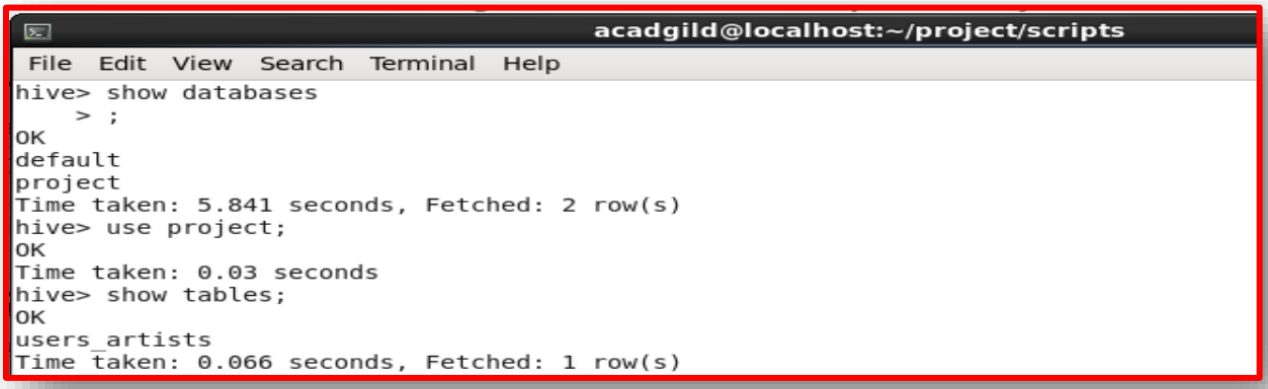
```
[acadgild@localhost scripts]$ hbase shell
2018-03-20 19:35:07,349 WARN  [main] util.NativeCodeLoader: Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-
1.2.6/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
[jar:file:/home/acadgild/install/hadoop/hadoop-
2.6.5/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> list
TABLE
song-artist-map
station-geo-map
subscribed-users
3 row(s) in 0.4570 seconds

=> ["song-artist-map", "station-geo-map", "subscribed-users"]
hbase(main):002:0> scan 'song-artist-map'
ROW                           COLUMN+CELL
 S200                         column=artist:artistid,
timestamp=1521553737048, value=A300
 S201                         column=artist:artistid,
timestamp=1521553747653, value=A301
 S202                         column=artist:artistid,
timestamp=1521553758212, value=A302
```

```
 S203                          column=artist:artistid,
timestamp=1521553768477, value=A303
 S204                          column=artist:artistid,
timestamp=1521553778780, value=A304
 S205                          column=artist:artistid,
timestamp=1521553789505, value=A301
 S206                          column=artist:artistid,
timestamp=1521553799985, value=A302
 S207                          column=artist:artistid,
timestamp=1521553810612, value=A303
 S208                          column=artist:artistid,
timestamp=1521553821025, value=A304
 S209                          column=artist:artistid,
timestamp=1521553831661, value=A305
10 row(s) in 0.2530 seconds

hbase(main):003:0>
```



■ Verifying the tables created in Hive

```
hive> show databases
    > ;
OK
default
project
Time taken: 5.841 seconds, Fetched: 2 row(s)
hive> use project;
OK
Time taken: 0.03 seconds
hive> show tables;
OK
users_artists
Time taken: 0.066 seconds, Fetched: 1 row(s)
```


acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
hive> show databases
> ;
OK
default
project
Time taken: 5.841 seconds, Fetched: 2 row(s)
hive> use project;
OK
Time taken: 0.03 seconds
hive> show tables;
OK
users_artists
Time taken: 0.066 seconds, Fetched: 1 row(s)

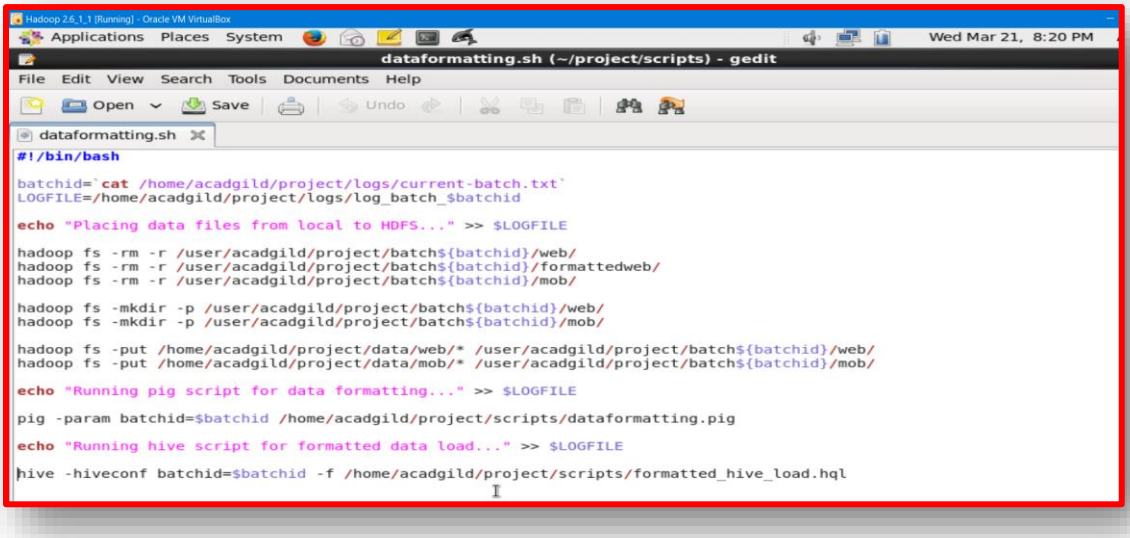
Step 4: Performing Data Formatting

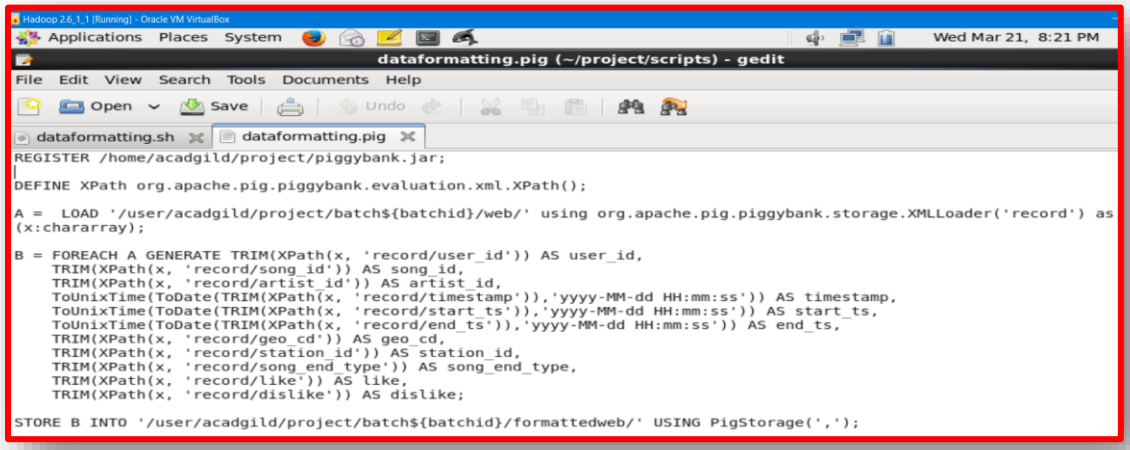
- **Registering piggybank jar to use built in utilities and Defining a name for XPath utility.**

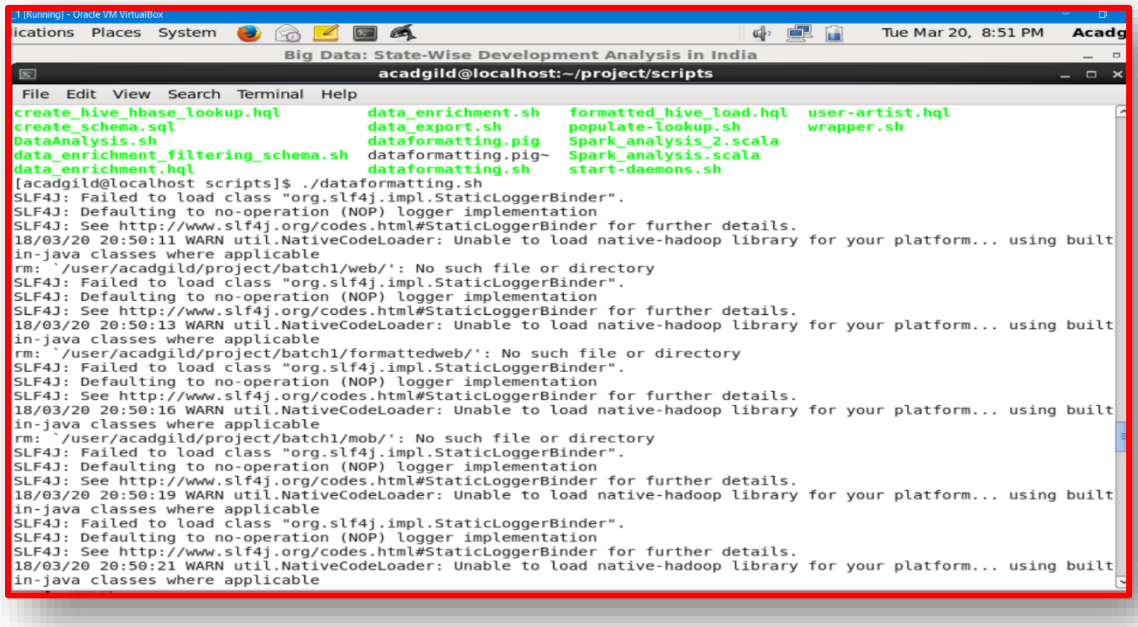
```
grunt> REGISTER /home/acadgild/project/piggybank.jar;
grunt> DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();
```

- **Loading and Formatting data using pig and hive**

```
[acadgild@localhost scripts]$ ./dataformatting.sh
2018-03-20 19:25:39,612 WARN  [main] util.NativeCodeLoader: Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
2018-03-20 19:25:39,612 WARN  [main] util.NativeCodeLoader: Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
2018-03-20 19:25:39,612 WARN  [main] util.NativeCodeLoader: Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-
1.2.6/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in
```





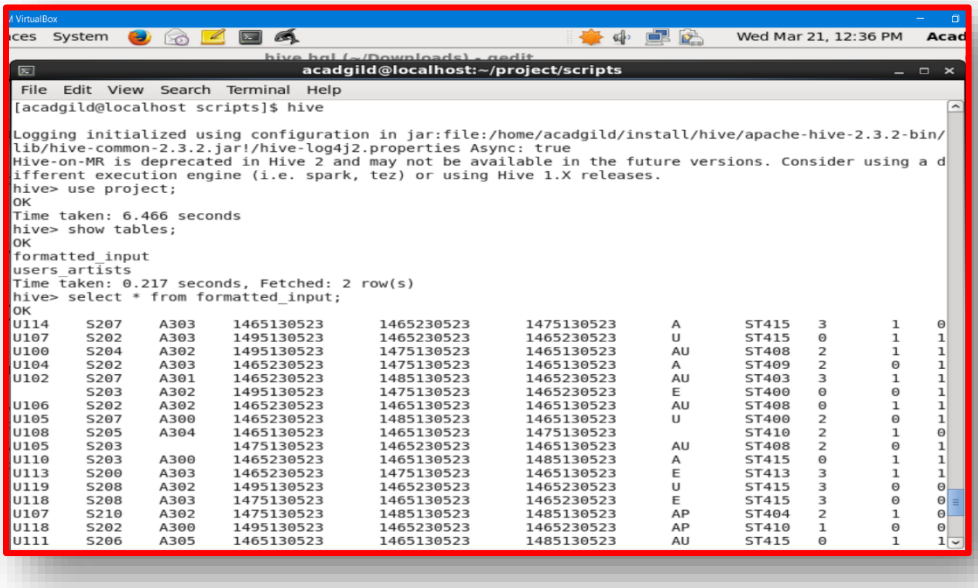


■ Checking the formatted data stored in hive

```
[acadgild@localhost scripts]$ hive

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-
common-2.3.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the
future versions. Consider using a different execution engine (i.e.
spark, tez) or using Hive 1.X releases.
hive> use project;
OK
Time taken: 6.466 seconds
hive> show tables;
OK
formatted_input
users_artists
```

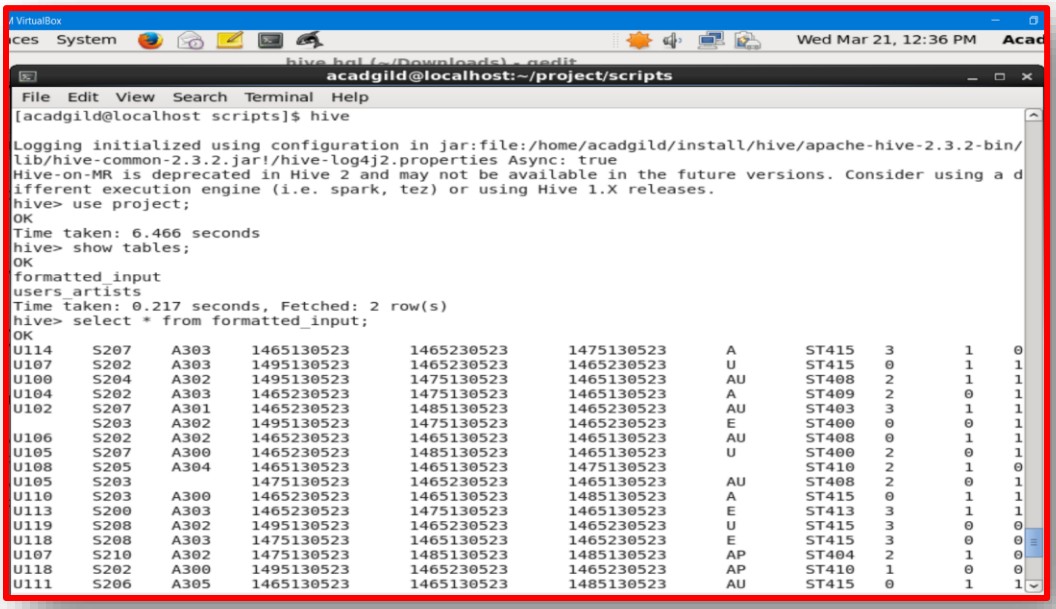
```
Time taken: 0.217 seconds, Fetched: 2 row(s)
hive> select * from formatted_input;
OK
U114  S207  A303  1465130523 1465230523 1475130523 A     ST415 3     1
      0
U107  S202  A303  1495130523 1465230523 1465230523 U     ST415 0     1
      1
U100  S204  A302  1495130523 1475130523 1465130523 AU    ST408 2     1
      1
U104  S202  A303  1465230523 1475130523 1465130523 A     ST409 2     0
      1
U102  S207  A301  1465230523 1485130523 1465230523 AU    ST403 3     1
      1
      S203  A302  1495130523 1475130523 1465230523 E     ST400 0     0
      1
U106  S202  A302  1465230523 1465130523 1465130523 AU    ST408 0     1
      1
U105  S207  A300  1465230523 1485130523 1465130523 U     ST400 2     0
      1
U108  S205  A304  1465130523 1465130523 1475130523       ST410 2     1
      0
U105  S203        1475130523 1465230523 1465130523 AU    ST408 2     0
      1
U110  S203  A300  1465230523 1465130523 1485130523 A     ST415 0     1
      1
U113  S200  A303  1465230523 1475130523 1465130523 E     ST413 3     1
      1
U119  S208  A302  1495130523 1465230523 1465230523 U     ST415 3     0
      0
U118  S208  A303  1475130523 1465130523 1465230523 E     ST415 3     0
      0
U107  S210  A302  1475130523 1485130523 1485130523 AP    ST404 2     1
      0
U118  S202  A300  1495130523 1465230523 1465230523 AP    ST410 1     0
      0
U111  S206  A305  1465130523 1465130523 1485130523 AU    ST415 0     1
      1
U116  S208  A303  1465230523 1485130523 1475130523 A     ST413 1     0
      1
U101  S202  A300  1465230523 1465130523 1475130523 U     ST401 0     0
      1
U120  S206  A303  1495130523 1485130523 1465130523 AU    ST414 0     0
      0
U106  S205  A300  1462863262 1462863262 1494297562 AP    ST407 2     1
      1
U114  S209  A303  1465490556 1462863262 1494297562 U     ST411 2     1
      0
U113  S203  A304  1465490556 1465490556 1462863262 U     ST405 0     0
      1
U108  S200  A302  1468094889 1462863262 1468094889 U     ST414 0     0
      1
U102  S203  A305  1465490556 1465490556 1494297562 U     ST404 2     0
      0
      S208  A300  1465490556 1494297562 1465490556 U     ST411 1     0
      1
U115  S200  A300  1465490556 1494297562 1465490556 AU    ST404 3     0
      0
U111  S204  A300  1465490556 1465490556 1468094889 U     ST410 3     1
      1
```

```
U120  S201  A300  1494297562 1465490556 1468094889       ST410 3     0
      1
U113  S203        1465490556 1465490556 1465490556 A     ST402 1     1
      0
U109  S203  A304  1462863262 1494297562 1468094889 E     ST405 1     1
      1
U110  S202  A303  1494297562 1494297562 1468094889 AU    ST402 2     1
      0
U100  S200  A301  1494297562 1494297562 1494297562 AP    ST410 3     1
      1
U101  S208  A300  1462863262 1468094889 1462863262 E     ST408 0     1
      1
U106  S206  A300  1494297562 1465490556 1462863262 A     ST405 3     1
      0
U107  S202  A304  1494297562 1468094889 1462863262 U     ST409 0     0
      0
U103  S204  A300  1468094889 1494297562 1465490556 AU    ST411 2     1
      0
U103  S202  A300  1465490556 1465490556 1465490556 A     ST415 2     1
      1
U113  S203  A303  1462863262 1468094889 1494297562 U     ST408 2     0
      0
U113  S204  A301  1494297562 1494297562 1465490556 E     ST415 3     0
      1
Time taken: 3.323 seconds, Fetched: 40 row(s)
hive>
hive> You have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```



■ Checking the data files stored in HDFS

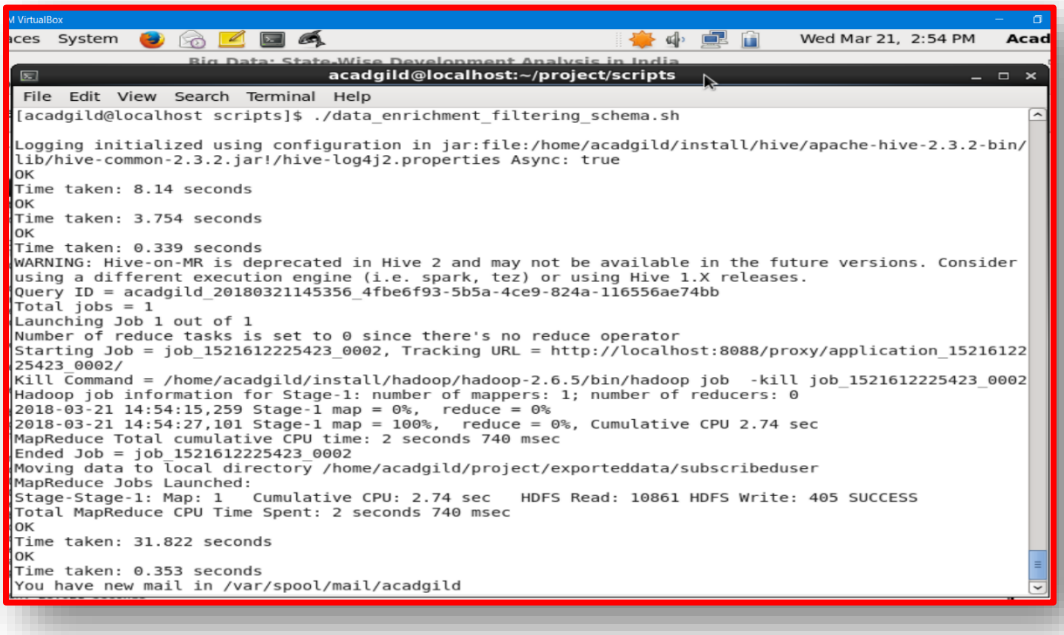
```
[acadgild@localhost scripts]$ hadoop fs -ls
/user/hive/warehouse/project.db/formatted_input
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for
further details.
18/03/21 12:51:27 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
Found 2 items
-rwxr-xr-x   1 acadgild supergroup       1239 2018-03-20 20:50
/user/hive/warehouse/project.db/formatted_input/file.txt
-rwxr-xr-x   1 acadgild supergroup       1236 2018-03-20 20:51
/user/hive/warehouse/project.db/formatted_input/part-m-00000
[acadgild@localhost scripts]$
```



Step 5: Performing Data Enrichment and Cleaning

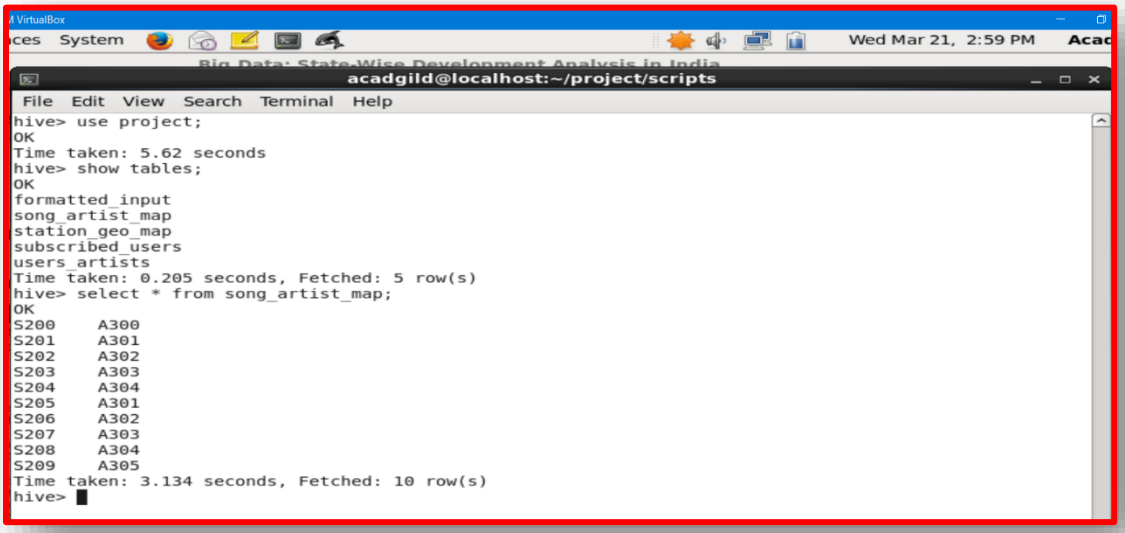
- **Creating lookup tables in Hive by importing data from HBase**

```
[acadgild@localhost scripts]$ ./data_enrichment_filtering_schema.sh

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-
common-2.3.2.jar!/hive-log4j2.properties Async: true
OK
Time taken: 8.14 seconds
OK
Time taken: 3.754 seconds
OK
Time taken: 0.339 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180321145356_4fbe6f93-5b5a-4ce9-824a-
116556ae74bb
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1521612225423_0002, Tracking URL =
http://localhost:8088/proxy/application_1521612225423_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop
job  -kill job_1521612225423_0002
Hadoop job information for Stage-1: number of mappers: 1; number of
reducers: 0
2018-03-21 14:54:15,259 Stage-1 map = 0%,  reduce = 0%
2018-03-21 14:54:27,101 Stage-1 map = 100%,  reduce = 0%, Cumulative
CPU 2.74 sec
MapReduce Total cumulative CPU time: 2 seconds 740 msec
Ended Job = job_1521612225423_0002
Moving data to local directory
/home/acadgild/project/exporteddata/subscribeduser
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.74 sec   HDFS Read: 10861
HDFS Write: 405 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 740 msec
OK
Time taken: 31.822 seconds
OK
Time taken: 0.353 seconds
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```



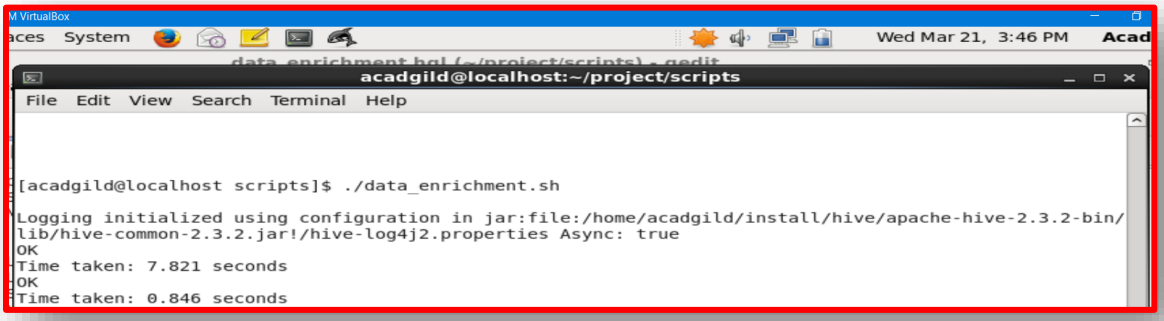
- **Verifying the tables created and data inserted**

```
hive> use project;
OK
Time taken: 5.62 seconds
hive> show tables;
OK
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.205 seconds, Fetched: 5 row(s)
hive> select * from song_artist_map;
OK
S200  A300
S201  A301
S202  A302
S203  A303
S204  A304
S205  A301
S206  A302
S207  A303
S208  A304
S209  A305
Time taken: 3.134 seconds, Fetched: 10 row(s)
hive>
```



- **Running data_enrichment.sh to create a table in Hive which has enriched data and based on the partition rules**

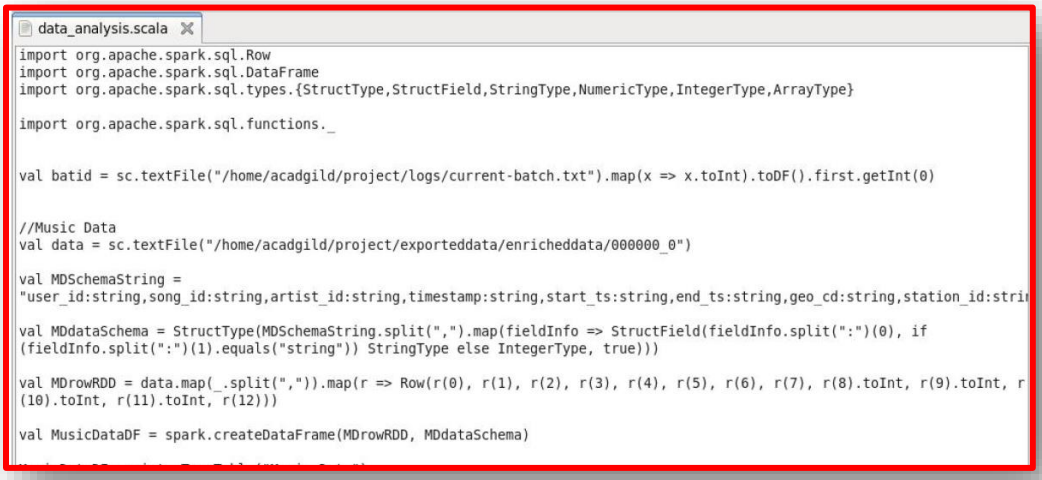
```
[acadgild@localhost scripts]$ ./data_enrichment.sh

Logging initialized using configuration in
jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-
common-2.3.2.jar!/hive-log4j2.properties Async: true
OK
Time taken: 8.14 seconds
OK
Time taken: 3.754 seconds
OK
Time taken: 0.339 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180321145356_4fbe6f93-5b5a-4ce9-824a-
116556ae74bb
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1521612225423_0002, Tracking URL =
http://localhost:8088/proxy/application_1521612225423_0002/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop
job  -kill job_1521612225423_0002
Hadoop job information for Stage-1: number of mappers: 1; number of
reducers: 0
2018-03-21 14:54:15,259 Stage-1 map = 0%,  reduce = 0%
2018-03-21 14:54:27,101 Stage-1 map = 100%,  reduce = 0%, Cumulative
CPU 2.74 sec
MapReduce Total cumulative CPU time: 2 seconds 740 msec
Ended Job = job_1521612225423_0002
```


M VirtualBox
aces System
Wed Mar 21, 3:46 PM
Acad
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
[acadgild@localhost scripts]$ ./data_enrichment.sh
Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
OK
Time taken: 7.821 seconds
OK
Time taken: 0.846 seconds

Step 6: Performing Data Analysis

- **Running data_analysis.sh which performs the below actions,**

  - Batch id number from the batch file will be picked and respective Log File will be considered.
  - This will in turn run the data_analysis.scala. This will perform the data analysis required in the problem statement given and save the result to the Local FS.
  - The data_analysis.scala will Import Row, DataFrame, Structure type and function dependencies needed to perform analysis
  - Creating the schema for the data, DataFrame from the schema and data and temporary table from the DataFrame created.

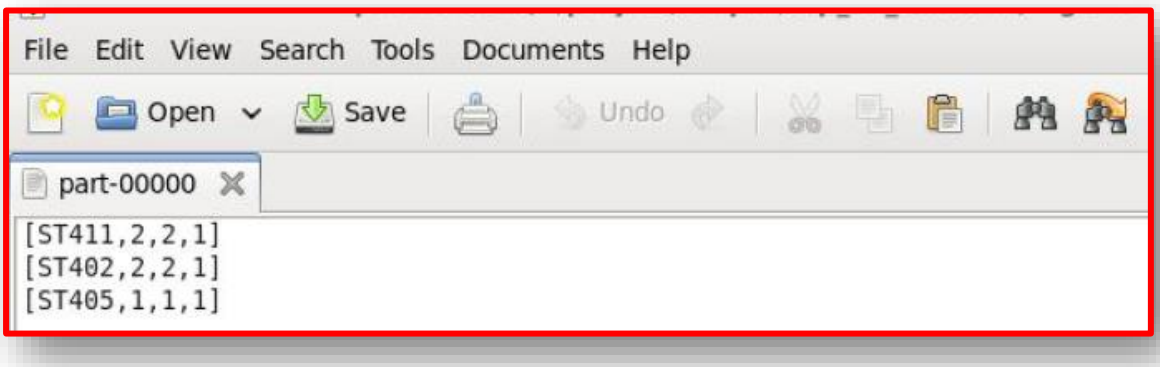


- **TASK 1: Top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.**

Command Line Script:

```
val Top10Stations = spark.sql(s"SELECT station_id, COUNT(DISTINCT song_id) AS total_distinct_songs_played, COUNT(DISTINCT
user_id) AS distinct_user_count, batchid FROM Music_Data WHERE status='pass' AND batchid=$batid AND like=1 GROUP BY
station_id,batchid ORDER BY total_distinct_songs_played DESC LIMIT 10");

Top10Stations.rdd.saveAsTextFile("/home/acadgild/project/output/top 10 stations")
```

Output:

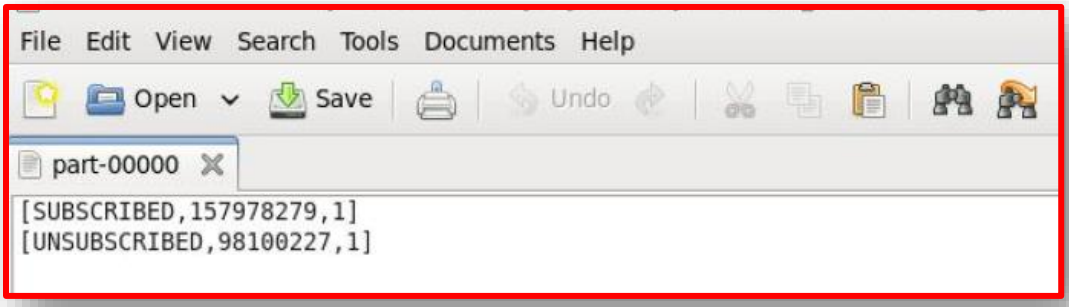


- **Task 2 : Total duration of songs played by each type of user, where type of user can be 'subscribed' or 'unsubscribed'. An unsubscribed user is the one whose record is either not present in Subscribed_users lookup table or has subscription_end_date earlier than the timestamp of the song played by him.**

Command Line Script:

```
val users_behavior = spark.sql(s"SELECT CASE WHEN (subusers.user_id IS NULL OR CAST(music.timestamp AS DECIMAL(20,0)) > CAST
(subusers.end_dt AS DECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (subusers.user_id IS NOT NULL AND CAST(music.timestamp AS
DECIMAL(20,0)) <= CAST(subusers.end_dt AS DECIMAL(20,0))) THEN 'SUBSCRIBED' END AS user_type, SUM(ABS(CAST(music.end_ts AS
DECIMAL(20,0))-CAST(music.start_ts AS DECIMAL(20,0)))) AS duration, batchid FROM Music_Data music LEFT OUTER JOIN
Music_SubscribedUsers subusers ON music.user_id=subusers.user_id WHERE music.status='pass' AND music.batchid=$batid GROUP
BY CASE WHEN (subusers.user_id IS NULL OR CAST(music.timestamp AS DECIMAL(20,0)) > CAST(subusers.end_dt AS DECIMAL(20,0)))
THEN 'UNSUBSCRIBED' WHEN (subusers.user_id IS NOT NULL AND CAST(music.timestamp AS DECIMAL(20,0)) <= CAST(subusers.end_dt
AS DECIMAL(20,0))) THEN 'SUBSCRIBED' END,batchid")

users_behavior.rdd.saveAsTextFile("/home/acadgild/project/output/users_behavior")
```

Output:

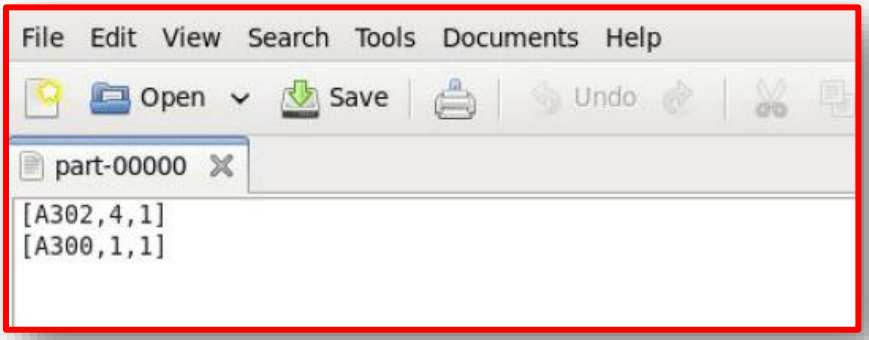


- **Task 3 : Top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.**

Command Line Script:

```
val connected_artists = spark.sql(s"SELECT ua.artists, COUNT(DISTINCT ua.user_id) AS user_count, md.batchid FROM
Music_UserArtists ua INNER JOIN ( SELECT artist_id, song_id, user_id, batchid FROM Music_Data WHERE status='pass' AND
batchid=$batid ) md ON ua.artists=md.artist_id AND ua.user_id=md.user_id GROUP BY ua.artists,batchid ORDER BY user_count
DESC LIMIT 10")

connected_artists.rdd.saveAsTextFile("/home/acadgild/project/output/connected_artists")
```

Output:

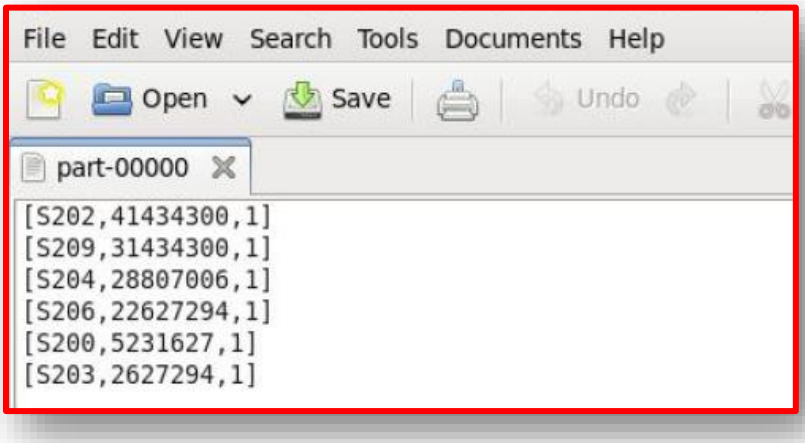


- **Task 4: Top 10 songs who have generated the maximum revenue. Royalty applies to a song only if it was liked or was completed successfully or both.**

Command Line Script:

```
val top_10_royalty_songs = spark.sql(s"SELECT song_id, SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL
(20,0)))) AS duration, batchid FROM Music_Data WHERE status='pass' AND batchid=$batid AND (like=1 OR song_end_type=0) GROUP
BY song_id,batchid ORDER BY duration DESC LIMIT 10")

top_10_royalty_songs.rdd.saveAsTextFile("/home/acadgild/project/output/top_10_royalty_songs")
```

Output:



- **Task 5: Top 10 unsubscribed users who listened to the songs for the longest duration.**

Command Line Script:

```
val top_10_unsubscribed_users = spark.sql(s"SELECT md.user_id, SUM(ABS(CAST(md.end_ts AS DECIMAL(20,0))-CAST(md.start_ts AS DECIMAL(20,0)))) AS duration FROM Music_Data md LEFT OUTER JOIN Music_SubscribedUsers su ON md.user_id=su.user_id WHERE md.status='pass' AND md.batchid=$batid AND (su.user_id IS NULL OR (CAST(md.timestamp AS DECIMAL(20,0)) > CAST(su.end_dt AS DECIMAL(20,0)))) GROUP BY md.user_id ORDER BY duration DESC LIMIT 10")

top_10_unsubscribed_users.rdd.saveAsTextFile("/home/acadgild/project/output/top_10_unsubscribed_users")
```

Output: