

ACADGILD

Project: MUSIC DATA ANALYSIS

BIG DATA HADOOP AND SPARK TRAINING

ANKITA JAISWAL

12/30/2017

A leading music-catering company is planning to analyse large amount of data received from varieties of sources, namely mobile app and website to track the behaviour of users, classify users, calculate royalties associated with the song and make appropriate business strategies. The file server receives data files periodically after every 3 hours.

[Type text]

TABLE OF CONTENTS

- Introduction
- Data-set
- Look-Up Tables Files
- Flow of Operations
- Steps to perform data analysis on the Music Data
- Step 1: Launch all necessary daemons
- Step 2: Start Job Scheduling
- Step 3: Populate Look-Up tables
- Step 4: Perform Data Formatting
- Step 5: Perform Data Enrichment and Cleaning
- Step 6: Perform Data Analysis
- Post Analysis

Introduction

This project work includes analysis of large amount of data received from varieties of sources namely from mobile app and website periodically after every 3 hours, to track the behaviour of users, to classify the users, to calculate royalties associated with the song and to make appropriate business strategies.

Dataset

Google Drive Link:

https://drive.google.com/drive/folders/0B_P3pWagdlrrMjJGVlNsSUEtbG8?usp=sharing

- Data coming from web applications resides in /data/web and has **xml** format.
- Data coming from mobile applications resides in /data/mob and has **csv** format.
- Data present in lookup directory is stored in **HBase** tables.

Fields present in the data files

Data files contain below fields.

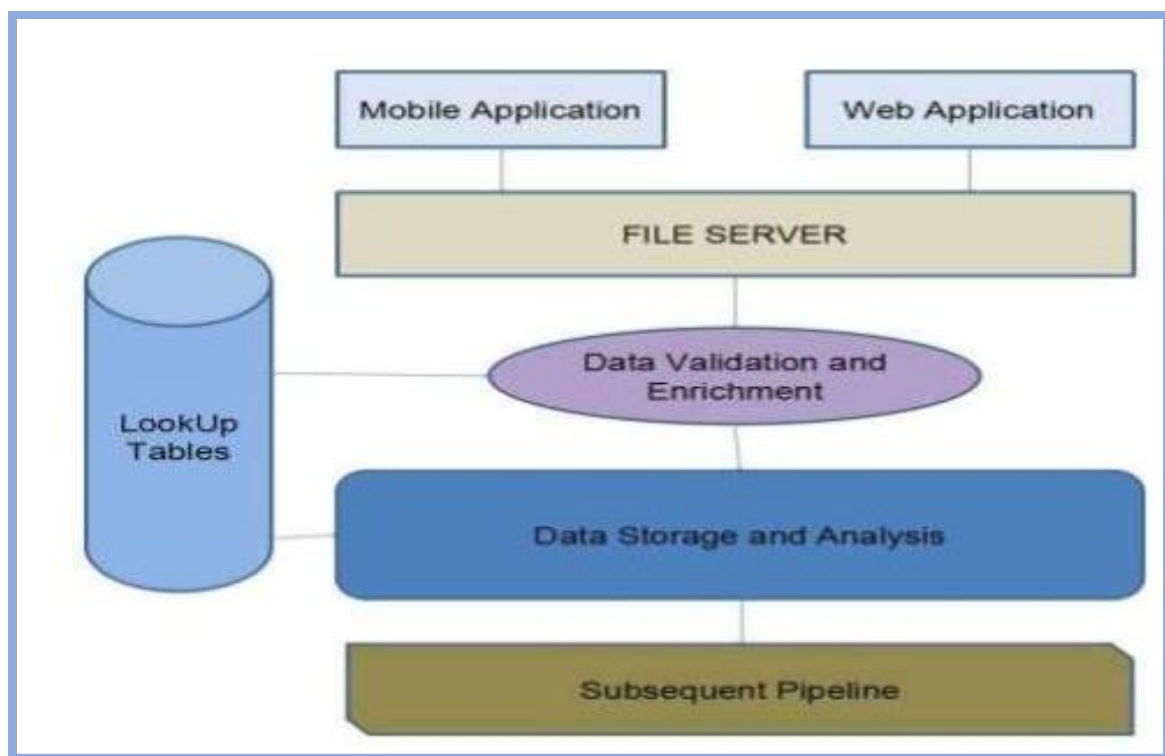
Column Name/Field Name	Column Description/Field Description
User_id	Unique identifier of every user
Song_id	Unique identifier of every song
Artist_id	Unique identifier of the lead artist of the song
Timestamp	Timestamp when the record was generated
Start_ts	Start timestamp when the song started to play
End_ts	End timestamp when the song was stopped
Geo_cd	Can be 'A' for USA region, 'AP' for asia pacific region, 'J' for Japan region, 'E' for europe and 'AU' for australia region
Station_id	Unique identifier of the station from where the song was played
Song_end_type	How the song was terminated. 0 means completed successfully 1 means song was skipped 2 means song was paused 3 means other type of failure like device issue, network error etc.
Like	0 means song was not liked 1 means song was liked
Dislike	0 means song was not disliked 1 means song was disliked

LookUp Tables

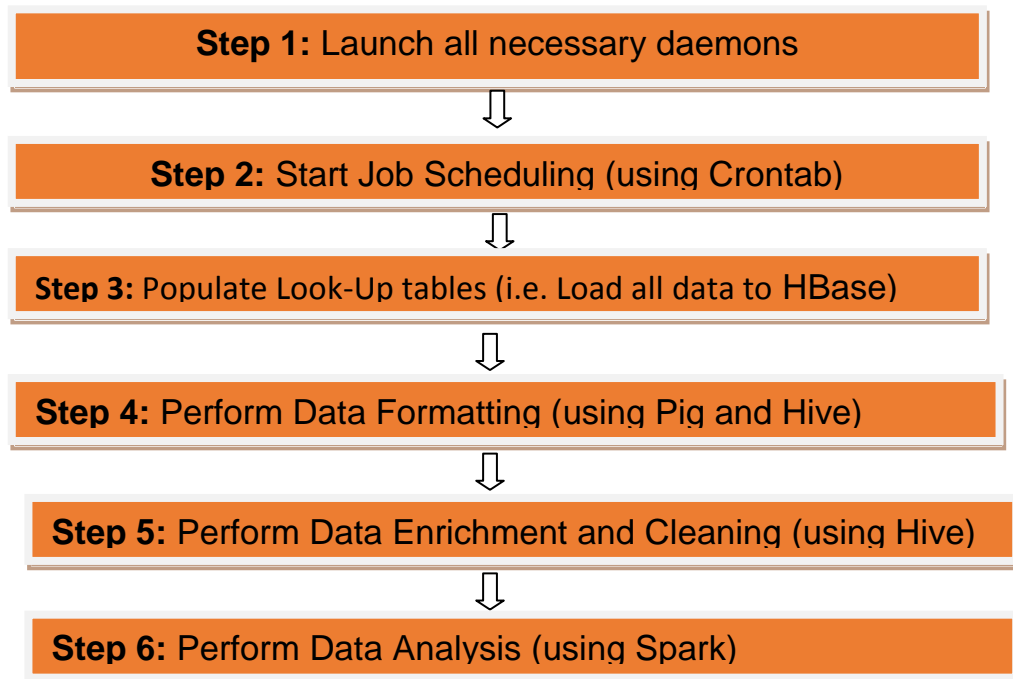
There are some existing look up tables present in NoSQL databases. They play an important role in data enrichment and analysis.

Table Name	Description
Station_Geo_Map	Contains mapping of a geo_cd with station_id
Subscribed_Users	Contains user_id, subscription_start_date and subscription_end_date. Contains details only for subscribed users
Song_Artist_Map	Contains mapping of song_id with artist_id alongwith royalty associated with each play of the song
User_Artist_Map	Contains an array of artist_id(s) followed by a user_id

Flow of Operations



Steps to perform data analysis on the Music Data:



Step 1: Launch all necessary daemons

- Give permissions to scripts folder in project, so we are able to run scripts from the bash shell.

```
[acadgild@localhost ~]$ chmod 774 /home/acadgild/project/scripts/*  
[acadgild@localhost ~]$
```

- Run the shell script [start-daemons.sh](#)

```
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/start-daemons.sh  
Batch file found!  
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh  
17/12/27 20:27:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl  
asses where applicable  
Starting namenodes on [localhost]  
localhost: namenode running as process 2781. Stop it first.  
localhost: datanode running as process 2882. Stop it first.  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: secondarynamenode running as process 3023. Stop it first.  
17/12/27 20:28:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl  
asses where applicable  
starting yarn daemons  
resourcemanager running as process 3243. Stop it first.  
localhost: nodemanager running as process 3343. Stop it first.  
master running as process 5776. Stop it first.  
historyserver running as process 5849. Stop it first.  
[sudo] password for acadgild:  
Starting mysqld: [ OK ]  
/usr/local/hive/bin/hive-config.sh: line 1: syntax error near unexpected token `('`  
/usr/local/hive/bin/hive-config.sh: line 1: `# Licensed to the Apache Software Foundation (ASF) under one or more'  
Starting Hive Metastore Server  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.clas  
s]  
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta  
ticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]  
█
```

- In the shell script `start-daemons.sh` used above, we perform the following operations:

```
start-daemons.sh X
#!/bin/bash

if [ -f "/home/acadgild/project/logs/current-batch.txt" ]
then
    echo "Batch file found!"
else
    echo -n "1" > "/home/acadgild/project/logs/current-batch.txt"
fi

chmod 777 /home/acadgild/project/logs/current-batch.txt

batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_${batchid}

chmod 777 /home/acadgild/project/logs/*

echo "Starting daemons..." >> $LOGFILE

#To start hadoop daemons, following command is used
$HADOOP_HOME/sbin/start-all.sh

#To start hbase, following command is used
$HBASE_HOME/bin/start-hbase.sh

#To start history server, following command is used
mr-jobhistory-daemon.sh start historyserver

#To start mysqld service, following command is used
sudo service mysqld start

#To start hive metastore, following command is used
hive --service metastore
```

- Check if a file **current-batch.txt** has been created or not,
- If already created, print **Batch File Found!** else create the file and add 1 to it to signify batch 1.

```
current-batch.txt X
1
```

- Give permissions to the file, so that we are able to modify it on the run.
- Get the batch id number from the batch file created above and create a **Log File** for the batch using the batch id. This will be **log_batch_1**.
- Through out the course of the analysis process this log file will document the tasks that are performed for the Music Data Analysis.
- Add a log to the Log File signifying that the all necessary daemons have been started.

```
log_batch_1 X
Starting daemons
```

Step 2:Start Job Scheduling

Open the crontab file and insert the statement:

`**/3 * * * /home/acadgild/project/scripts/wrapper.sh`

Crontab is used for Job Scheduling. In the -e mode, Crontab schedules execution of commands by a regular user. The statement above runs the wrapper.sh shell script every 3 hours.

```
acadmild@localhost:~  
File Edit View Search Terminal Help  
[acadmild@localhost ~]$ sudo crontab -e  
[sudo] password for admild:  
  
acadmild@localhost:~  
File Edit View Search Terminal Help  
* */3 * * * /home/acadmild/project/scripts/wrapper.sh  
~  
~  
~  
~  
~  
~  
-- INSERT --  
  
acadmild@localhost:~  
File Edit View Search Terminal Help  
[acadmild@localhost ~]$ sudo crontab -e  
[sudo] password for admild:  
crontab: installing new crontab  
[acadmild@localhost ~]$
```

In the shell script [wrapper.sh](#) used above, all the processes needed to perform analysis on the Music Data is called once every 3 hours thereby creating a new batch. This is the job scheduling.

```
*wrapper.sh x  
#!/bin/bash  
  
#This script calls other scripts in sequential fashion  
  
#Below scripts work on the data present inside lookup,web and mob folders which are already provided as part of this project  
  
#Below script starts all required services  
sh /home/acadmild/project/scripts/start-daemons.sh  
  
#Below script creates hbase and hive lookup tables  
sh /home/acadmild/project/scripts/populate-lookup.sh  
  
#Below script creates hive tables on top of hbase tables  
sh /home/acadmild/project/scripts/data_enrichment_filtering_schema.sh  
  
#Below script collects web and mob data, and creates a table accordingly  
sh /home/acadmild/project/scripts/dataformatting.sh  
  
#Below script enriches and filters the data  
sh /home/acadmild/project/scripts/data_enrichment.sh  
  
#Below script performs analysis on filtered and enriched data  
sh /home/acadmild/project/scripts/data_analysis.sh
```

Step 3:Populate Look-Up tables

Below is the shell script [populate-lookup.sh](#) that is used to load the data for the lookup tables into HBase tables.

The following operations are performed:

- Get the batch id number from the batch file and get the **Log File** for the batch using the batch id. This will be **log_batch_1**
- Add logs to the Log File signifying that the lookup tables are being created and populated
- Create the HBase tables for the lookup data files: **song-artist**, **stn-geocd** and **user-subscn** with their column families
- For every lookup data file, read each line, extract the columns (comma separated) and add the data as rows to the corresponding HBase tables created above
- Run the hive script [user-artist.hql](#). This will populate a hive table with the data in the lookup data file **user-artist**. This is because this file has an array column that is difficult to populate in HBase.

```
populate-lookup.sh X
#Populating station-geo-map lookup table
file="/home/acadgild/project/data/lookup/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"

#Populating subscribed-users lookup table
file="/home/acadgild/project/data/lookup/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"

#Populating song-artist-map lookup table
file="/home/acadgild/project/data/lookup/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"

#Populating user-artist lookup table using hive

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

populate-lookup.sh X user-artist.hql X
CREATE DATABASE IF NOT EXISTS project;
USE project;
CREATE TABLE user_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/data/lookup/user-artist.txt'
OVERWRITE INTO TABLE user_artists;

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/project/exporteddata/userartists'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
SELECT user_id,artists FROM user_artists LATERAL VIEW explode(artists_array) a AS artists;
```


Below is screenshot of execution of populate-lookup.sh

```
acadgild@localhost:~  
File Edit View Search Terminal Help  
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/populate-lookup.sh  
2017-12-26 19:23:15,011 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a  
vailable  
HBase Shell; enter 'help<RETURN>' for list of supported commands.  
Type "exit<RETURN>" to leave the HBase Shell  
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015  
  
create 'station-geo-map', 'geo'  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta  
ticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
2017-12-26 19:23:29,418 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b  
uiltin-java classes where applicable  
0 row(s) in 10.2900 seconds  
  
Hbase::Table - station-geo-map  
2017-12-26 19:23:56,471 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a  
vailable  
HBase Shell; enter 'help<RETURN>' for list of supported commands.  
Type "exit<RETURN>" to leave the HBase Shell  
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015  
  
create 'subscribed-users', 'subscn'  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta  
ticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
2017-12-26 19:24:11,386 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b  
uiltin-java classes where applicable  
0 row(s) in 6.2480 seconds  
  
Hbase::Table - subscribed-users  
2017-12-26 19:24:26,472 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a  
vailable  
HBase Shell; enter 'help<RETURN>' for list of supported commands.  
Type "exit<RETURN>" to leave the HBase Shell  
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015  
  
create 'song-artist-map', 'artist'  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta  
ticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
2017-12-26 19:24:35,138 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b  
uiltin-java classes where applicable  
0 row(s) in 2.8390 seconds  
  
Hbase::Table - song-artist-map  
2017-12-26 19:24:58,471 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a  
vailable  
HBase Shell; enter 'help<RETURN>' for list of supported commands.  
Type "exit<RETURN>" to leave the HBase Shell  
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015  
  
put 'station-geo-map', 'ST400', 'geo:geo_cd', 'A'  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta  
ticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
2017-12-26 19:25:11,575 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b  
uiltin-java classes where applicable  
0 row(s) in 3.3670 seconds  
  
2017-12-26 19:25:47,351 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a  
vailable
```

Click to switch to "Workspace 2"

```

[acadgild@localhost ~]$ hbase shell
2017-12-26 20:13:22,147 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a
vailable
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

hbase(main):001:0> list
TABLE
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/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.
2017-12-26 20:14:10,091 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uilt-in java classes where applicable
song-artist-map
station-geo-map
subscribed-users
3 row(s) in 9.9690 seconds

=> ["song-artist-map", "station-geo-map", "subscribed-users"]
hbase(main):002:0>

```

Output of Hbase table

```

hbase(main):002:0> scan 'song-artist-map'
ROW COLUMN+CELL
S200 column=artist:artistid, timestamp=1514297111449, value=A300
S201 column=artist:artistid, timestamp=1514297161298, value=A301
S202 column=artist:artistid, timestamp=1514297193860, value=A302
S203 column=artist:artistid, timestamp=1514297230738, value=A303
S204 column=artist:artistid, timestamp=1514297255970, value=A304
S205 column=artist:artistid, timestamp=1514297282168, value=A301
S206 column=artist:artistid, timestamp=1514297323331, value=A302
S207 column=artist:artistid, timestamp=1514297359435, value=A303
S208 column=artist:artistid, timestamp=1514297392423, value=A304
S209 column=artist:artistid, timestamp=1514297409694, value=A305
10 row(s) in 1.1980 seconds

```

```

hbase(main):003:0> scan 'station-geo-map'
ROW COLUMN+CELL
ST400 column=geo:geo_cd, timestamp=1514296519158, value=A
ST401 column=geo:geo_cd, timestamp=1514296574684, value=AU
ST402 column=geo:geo_cd, timestamp=1514296624505, value=AP
ST403 column=geo:geo_cd, timestamp=1514296667458, value=J
ST404 column=geo:geo_cd, timestamp=1514296697969, value=E
ST405 column=geo:geo_cd, timestamp=1514296741936, value=A
ST406 column=geo:geo_cd, timestamp=1514296783859, value=AU
ST407 column=geo:geo_cd, timestamp=1514296824234, value=AP
ST408 column=geo:geo_cd, timestamp=1514296853210, value=E
ST409 column=geo:geo_cd, timestamp=1514296888787, value=E
ST410 column=geo:geo_cd, timestamp=1514296935794, value=A
ST411 column=geo:geo_cd, timestamp=1514296973125, value=A
ST412 column=geo:geo_cd, timestamp=1514297006203, value=AP
ST413 column=geo:geo_cd, timestamp=1514297047771, value=J
ST414 column=geo:geo_cd, timestamp=1514297073247, value=E
15 row(s) in 0.6470 seconds

```

```

hbase(main):004:0> scan 'subscribed-users'
ROW COLUMN+CELL
U100 column=subscn:enndtt, timestamp=1514297457650, value=1465130523
U100 column=subscn:startdt, timestamp=1514297442888, value=1465230523
U101 column=subscn:enndtt, timestamp=1514297552754, value=1475130523
U101 column=subscn:startdt, timestamp=1514297504540, value=1465230523
U102 column=subscn:enndtt, timestamp=1514297634715, value=1475130523
U102 column=subscn:startdt, timestamp=1514297594463, value=1465230523
U103 column=subscn:enndtt, timestamp=1514297722328, value=1475130523
U103 column=subscn:startdt, timestamp=1514297676630, value=1465230523
U104 column=subscn:enndtt, timestamp=1514297807931, value=1475130523
U104 column=subscn:startdt, timestamp=1514297765768, value=1465230523
U105 column=subscn:enndtt, timestamp=1514297904401, value=1475130523
U105 column=subscn:startdt, timestamp=1514297864704, value=1465230523
U106 column=subscn:enndtt, timestamp=1514297957630, value=1485130523
U106 column=subscn:startdt, timestamp=1514297934161, value=1465230523
U107 column=subscn:enndtt, timestamp=1514298026850, value=1455130523
U107 column=subscn:startdt, timestamp=1514297992811, value=1465230523
U108 column=subscn:enndtt, timestamp=1514298133413, value=1465230623
U108 column=subscn:startdt, timestamp=1514298082739, value=1465230523
U109 column=subscn:enndtt, timestamp=1514298224656, value=1475130523
U109 column=subscn:startdt, timestamp=1514298183935, value=1465230523
U110 column=subscn:enndtt, timestamp=1514298314273, value=1475130523
U110 column=subscn:startdt, timestamp=1514298265261, value=1465230523
U111 column=subscn:enndtt, timestamp=1514298405145, value=1475130523
U111 column=subscn:startdt, timestamp=1514298364481, value=1465230523
U112 column=subscn:enndtt, timestamp=1514298482929, value=1475130523
U112 column=subscn:startdt, timestamp=1514298436866, value=1465230523
U113 column=subscn:enndtt, timestamp=1514298570812, value=1485130523
U113 column=subscn:startdt, timestamp=1514298534533, value=1465230523
U114 column=subscn:enndtt, timestamp=1514298630000, value=1468130523
U114 column=subscn:startdt, timestamp=1514298602095, value=1465230523
15 row(s) in 1.0160 seconds

```

```

hbase(main):005:0>

```

[acadgild@loc... [project] [scripts] [acadgild@loc... acadgild@local... acadgild@local...]

```

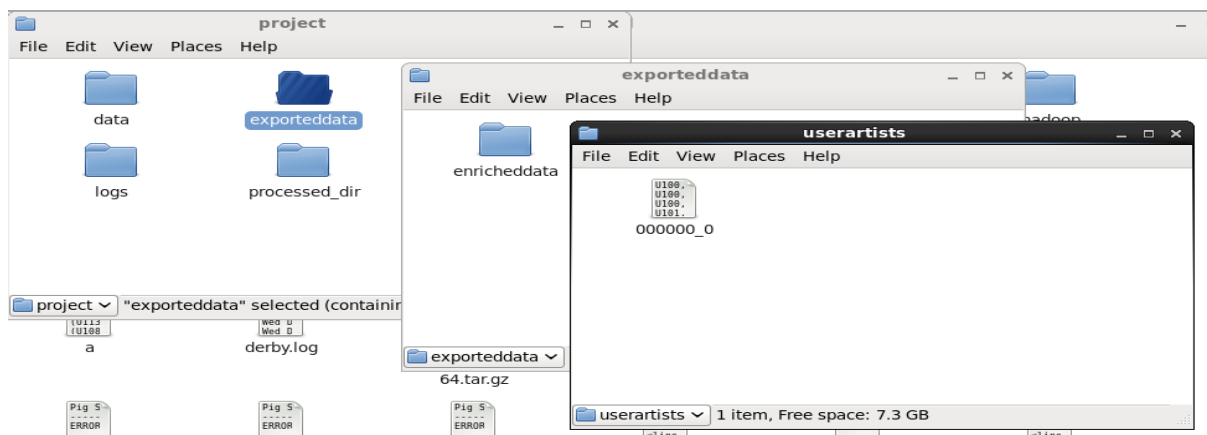
hive> CREATE DATABASE IF NOT EXISTS project;
OK
Time taken: 3.11 seconds
hive> USE project;
OK
Time taken: 0.152 seconds
hive> CREATE TABLE user_artists
> (
>   user_id STRING,
>   artists_array ARRAY<STRING>
> )
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> COLLECTION ITEMS TERMINATED BY '&';
OK
Time taken: 2.539 seconds

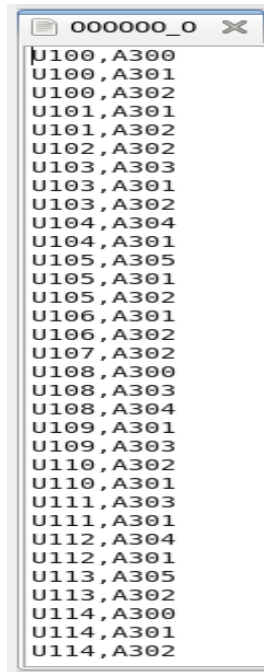
hive> LOAD DATA LOCAL INPATH '/home/acadgild/project/data/lookup/user-artist.txt'
> OVERWRITE INTO TABLE user_artists;
Loading data to table project.user_artists
Table project.user_artists stats: [numFiles=1, numRows=0, totalSize=240, rawDataSize=0]
OK
Time taken: 6.742 seconds

hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/project/exporteddata/userartists'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> STORED AS TEXTFILE
> SELECT user_id,artists FROM user_artists LATERAL VIEW explode(artists_array) a AS artists;
Query ID = acadgild_20171227133030_191d1388-0399-47ff-bf28-e34e4efde1e5
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_1514346487452_0001, Tracking URL = http://localhost:8088/proxy/application_1514346487452_0001/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job -kill job_1514346487452_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-12-27 13:32:26,139 Stage-1 map = 0%, reduce = 0%
2017-12-27 13:33:25,221 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.4 sec
MapReduce Total cumulative CPU time: 5 seconds 400 msec
Ended Job = job_1514346487452_0001
Copying data to local directory /home/acadgild/project/exporteddata/userartists
Copying data to local directory /home/acadgild/project/exporteddata/userartists
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 5.4 sec HDFS Read: 475 HDFS Write: 330 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 400 msec
OK
Time taken: 154.407 seconds
hive>
hive> SELECT * FROM user_artists;
OK
U100 ["A300","A301","A302"]
U101 ["A301","A302"]
U102 ["A302"]
U103 ["A303","A301","A302"]
U104 ["A304","A301"]
U105 ["A305","A301","A302"]
U106 ["A301","A302"]
U107 ["A302"]
U108 ["A300","A303","A304"]
U109 ["A301","A303"]
U110 ["A302","A301"]
U111 ["A303","A301"]
U112 ["A304","A301"]
U113 ["A305","A302"]
U114 ["A300","A301","A302"]
Time taken: 0.592 seconds, Fetched: 15 row(s)
hive>

```

Output of Hive Table stored in exported data as user artists





Step 4: Perform Data Formatting

Below is the shell script [dataformatting.sh](#) that is used to:

- Format the web xml data using **Pig** to a csv format *and*
 - Load the 2 data files, **mob** and **web** (formatted by Pig), to a Hive Table for data enrichment
- The following operations are performed:
- Get the batch id number from the batch file and get the **Log File** for the batch using the batch id. This will be **log_batch_1**
 - Add logs to the Log File signifying that the data is placed in the HDFS and the running of the Pig and Hive scripts for data formatting and loading respectively.
 - Delete, if they exist, folders for the mob, web and formattedweb. This is done in-case any old data remains because of execution failure.
 - Create the above folders web and mob that were deleted above and move the data from the Local FS to the HDFS. The formattedweb folder is created in the Pig Script.
 - Run the pig script [dataformatting.pig](#). This will format the web data (stored in the web folder in the HDFS) in xml format to csv format and store it in the HDFS in the folder formattedweb.
 - Run the hive script [formatted_hive_load.hql](#). This will load the data in the mob folder and formattedweb folder in the HDFS to a table formatted_input in Hive which will be used for data enrichment later.

```
dataformatting.sh X
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_${batchid}

echo "Placing data files from local to HDFS..." >> $LOGFILE

#Below three statements remove web, mob and formattedweb folders if they are already present
hadoop fs -rm -r /user/acadgild/project/batch${batchid}/web/
hadoop fs -rm -r /user/acadgild/project/batch${batchid}/mob/
hadoop fs -rm -r /user/acadgild/project/batch${batchid}/formattedweb/

#Below two statements create web and mob directories
hadoop fs -mkdir -p /user/acadgild/project/batch${batchid}/web/
hadoop fs -mkdir -p /user/acadgild/project/batch${batchid}/mob/

#Below two statements put the data from web and mob folders in [local file system] to web and mob folders in hdfs inside specific
#batchid folder
hadoop fs -put /home/acadgild/project/data/web/* /user/acadgild/project/batch${batchid}/web/
hadoop fs -put /home/acadgild/project/data/mob/* /user/acadgild/project/batch${batchid}/mob/

#Below pig script parses xml data present in web folder in hdfs
echo "Running pig script for data formatting..." >> $LOGFILE
pig -param batchid=${batchid} /home/acadgild/project/scripts/dataformatting.pig

#Below hive script creates table which contains data from web and mob folders
echo "Running hive script for formatted data load..." >> $LOGFILE
hive -hiveconf batchid=${batchid} -f /home/acadgild/project/scripts/formatted_hive_load.hql
```

dataformatting.pig

Stores the formatted data to a folder in the HDFS called formattedweb.

```
dataformatting.sh dataformatting.pig
REGISTER /home/acadgild/project/lib/piggybank.jar;

DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();

A = LOAD '/home/acadgild/project/data/web/file.xml.COMPLETED' using org.apache.pig.piggybank.storage.XMLLoader('record') as
(x:chararray);

B = Foreach A GENERATE TRIM(XPath(x, 'record/user_id')) AS user_id,
    TRIM(XPath(x, 'record/song_id')) as song_id,
    TRIM(XPath(x, 'record/artist_id')) as artist_id,
    ToUnixTime(ToDate(TRIM(XPath(x, 'record/timestamp')), 'yyyy-MM-dd HH:mm:ss')) as timestamp,
    ToUnixTime(ToDate(TRIM(XPath(x, 'record/start_ts')), 'yyyy-MM-dd HH:mm:ss')) as start_ts,
    ToUnixTime(ToDate(TRIM(XPath(x, 'record/end_ts')), 'yyyy-MM-dd HH:mm:ss')) as end_ts,
    TRIM(XPath(x, 'record/geo_cd')) as geo_cd,
    TRIM(XPath(x, 'record/station_id')) as station_id,
    TRIM(XPath(x, 'record/song_end_type')) as song_end_type,
    TRIM(XPath(x, 'record/like')) as like,
    TRIM(XPath(x, 'record/dislike')) as dislike;

STORE B INTO '/user/acadgild/project/batch1/formattedweb/' USING PigStorage(',');
```

formatted_hive_load.hql

Combines the data from mob and formattedweb to make one data-set and stores it partitioned by batchid.

```
dataformatting.sh dataformatting.pig formatted_hive_load.hql
USE project;

CREATE TABLE IF NOT EXISTS formatted_input
(
    User_id STRING,
    Song_id STRING,
    Artist_id STRING,
    Timestamp STRING,
    Start_ts STRING,
    End_ts STRING,
    Geo_cd STRING,
    Station_id STRING,
    Song_end_type INT,
    Like INT,
    Dislike INT
)
PARTITIONED BY
(batchid INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';

LOAD DATA INPATH '/user/acadgild/project/batch${hiveconf:batchid}/formattedweb/'
INTO TABLE formatted_input PARTITION (batchid=${hiveconf:batchid});

LOAD DATA INPATH '/user/acadgild/project/batch${hiveconf:batchid}/mob/'
INTO TABLE formatted_input PARTITION (batchid=${hiveconf:batchid});

LOAD DATA INPATH '/user/acadgild/project/batch1/formattedweb/part-m-00000'
INTO TABLE formatted_input PARTITION (batchid=1);

LOAD DATA INPATH '/user/acadgild/project/batch1/mob/'
INTO TABLE formatted_input PARTITION (batchid=1);
```


Below screenshots shows execution of dataformatting.sh

```
cadgild@localhost ~]$ sh /home/acadgild/project/scripts/dataformatting.sh
/12/27 13:52:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
: /user/acadgild/project/batch1/web/: No such file or directory
/12/27 13:52:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
: /user/acadgild/project/batch1/mob/: No such file or directory
/12/27 13:52:29 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
: /user/acadgild/project/batch1/formattedweb/: No such file or directory
/12/27 13:52:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
/12/27 13:52:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
/12/27 13:53:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
/12/27 13:53:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
ses where applicable
17-12-27 13:53:36,007 INFO [main] pig.ExecTypeProvider: Trying ExecType : LOCAL
17-12-27 13:53:36,016 INFO [main] pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
17-12-27 13:53:36,016 INFO [main] pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
17-12-27 13:53:36,349 [main] INFO org.apache.pig.Main - Apache Pig version 0.14.0 (r1640057) compiled Nov 16 2014, 18:02:0
17-12-27 13:53:36,349 [main] INFO org.apache.pig.Main - Logging error messages to: /home/acadgild/pig_1514363016343.log
F4J: Class path contains multiple SLF4J bindings.
F4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
F4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
cLoggerBinder.class]
F4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
17-12-27 13:53:37,850 [main] WARN org.apache.hadoop.util.NativeCodeLoader: Unable to load native-hadoop library for your
atorm... using builtin-java classes where applicable
17-12-27 13:53:39,344 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/acadgild/.pigbootstrap not found
17-12-27 13:53:40,172 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Ins
ad, use mapreduce.jobtracker.address
17-12-27 13:53:40,178 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instea
use fs.defaultFS
```

Checking HDFS for the files created

```
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1
17/12/27 19:42:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 3 items
drwxr-xr-x - acadgild supergroup 0 2017-12-27 15:35 /user/acadgild/project/batch1/formattedweb
drwxr-xr-x - acadgild supergroup 0 2017-12-27 15:36 /user/acadgild/project/batch1/mob
drwxr-xr-x - acadgild supergroup 0 2017-12-27 14:22 /user/acadgild/project/batch1/web
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/web
17/12/27 19:47:15 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 1 items
-rw-r--r-- 1 acadgild supergroup 6716 2017-12-27 14:22 /user/acadgild/project/batch1/web/file.xml
[acadgild@localhost ~]$ █

[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/mob
17/12/27 19:50:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 1 items
-rw-r--r-- 1 acadgild supergroup 1239 2017-12-27 19:49 /user/acadgild/project/batch1/mob/file.txt
[acadgild@localhost ~]$ █

[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/formattedweb
17/12/27 15:30:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 2 items
drwxr-xr-x - acadgild supergroup 0 2017-12-27 15:10 /user/acadgild/project/batch1/formattedweb/_SUCCESS
-rw-r--r-- 1 acadgild supergroup 1276 2017-12-27 15:29 /user/acadgild/project/batch1/formattedweb/part-m-000000
```

Below screenshots shows output of formatted input tables in hive

```
hive> use project;
OK
Time taken: 2.016 seconds
hive> show tables;
OK
formatted_input
user_artists
Time taken: 0.684 seconds, Fetched: 2 row(s)
hive> LOAD DATA INPATH '/user/acadgild/project/batch1/formattedweb/part-m-000000'
> INTO TABLE formatted_input PARTITION (batchid=1);
Loading data to table project.formatted_input partition (batchid=1)
Partition project.formatted_input{batchid=1} stats: [numFiles=1, numRows=0, totalSize=1276, rawDataSize=0]
OK
Time taken: 9.515 seconds
hive> LOAD DATA INPATH '/user/acadgild/project/batch1/mob/'
1/mob/'BLE formatted_input PARTITION (batchid=1);
INTO TABLE formatted_input PARTITION (batchid=1);
Loading data to table project.formatted_input partition (batchid=1)
Partition project.formatted_input{batchid=1} stats: [numFiles=2, numRows=0, totalSize=2515, rawDataSize=0]
OK
Time taken: 3.582 seconds
```

```
hive> SELECT * FROM formatted_input;
OK
U114 S207 A303 1465130523 1465230523 1475130523 A ST415 3 1 0 1
U107 S202 A303 1495130523 1465230523 1465230523 U ST415 0 1 1 1
U100 S204 A302 1495130523 1475130523 1465130523 AU ST408 2 1 1 1
U104 S202 A303 1465230523 1475130523 1465130523 A ST409 2 0 1 1
U102 S207 A301 1465230523 1485130523 1465230523 AU ST403 3 1 1 1
S203 A302 1495130523 1475130523 1465230523 E ST400 0 0 1 1
U106 S202 A302 1465230523 1465130523 1465130523 AU ST408 0 1 1 1
U105 S207 A300 1465230523 1485130523 1465130523 U ST400 2 0 1 1
U108 S205 A304 1465130523 1465130523 1475130523 ST410 2 1 0 1
U105 S203 1475130523 1465230523 1465130523 AU ST408 2 0 1 1
U110 S203 A300 1465230523 1465130523 1485130523 A ST415 0 1 1 1
U113 S200 A303 1465230523 1475130523 1465130523 E ST413 3 1 1 1
U119 S208 A302 1495130523 1465230523 1465230523 U ST415 3 0 0 1
U118 S208 A303 1475130523 1465130523 1465230523 E ST415 3 0 0 1
U107 S210 A302 1475130523 1485130523 1485130523 AP ST404 2 1 0 1
U118 S202 A300 1495130523 1465230523 1465230523 AP ST410 1 0 0 1
U111 S206 A305 1465130523 1465130523 1485130523 AU ST415 0 1 1 1
U116 S208 A303 1465230523 1485130523 1475130523 A ST413 1 0 1 1
U101 S202 A300 1465230523 1465130523 1475130523 U ST401 0 0 1 1
U120 S206 A303 1495130523 1485130523 1465130523 AU ST414 0 0 0 1
(U106 S205 A300 1462863262 1462863262 1494297562 AP ST407 2 1 NULL 1
(U114 S209 A303 1465490556 1462863262 1494297562 U ST411 2 1 NULL 1
(U113 S203 A304 1465490556 1465490556 1462863262 U ST405 0 0 NULL 1
(U108 S200 A302 1468094889 1462863262 1468094889 U ST414 0 0 NULL 1
(U102 S203 A305 1465490556 1465490556 1494297562 U ST404 2 0 NULL 1
( S208 A300 1465490556 1494297562 1465490556 U ST411 1 0 NULL 1
(U115 S200 A300 1465490556 1494297562 1465490556 AU ST404 3 0 NULL 1
(U111 S204 A300 1465490556 1465490556 1468094889 U ST410 3 1 NULL 1
(U120 S201 A300 1494297562 1465490556 1468094889 ST410 3 0 NULL 1
(U113 S203 1465490556 1465490556 1465490556 A ST402 1 1 NULL 1
(U109 S203 A304 1462863262 1494297562 1468094889 E ST405 1 1 NULL 1
(U110 S202 A303 1494297562 1494297562 1468094889 AU ST402 2 1 NULL 1
(U100 S200 A301 1494297562 1494297562 1494297562 AP ST410 3 1 NULL 1
(U101 S208 A300 1462863262 1468094889 1462863262 E ST408 0 1 NULL 1
(U106 S206 A300 1494297562 1465490556 1462863262 A ST405 3 1 NULL 1
(U107 S202 A304 1494297562 1468094889 1462863262 U ST409 0 0 NULL 1
(U103 S204 A300 1468094889 1494297562 1465490556 AU ST411 2 1 NULL 1
(U103 S202 A300 1465490556 1465490556 1465490556 A ST415 2 1 NULL 1
(U113 S203 A303 1462863262 1468094889 1494297562 U ST408 2 0 NULL 1
(U113 S204 A301 1494297562 1494297562 1465490556 E ST415 3 0 NULL 1
Time taken: 3.766 seconds, Fetched: 40 row(s)
hive>
```

Checking the hive warehouse in hdfs for formatted_input

```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/warehouse/project.db/formatted_input/batchid=1
17/12/27 15:43:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup 1239 2017-12-27 14:22 /user/hive/warehouse/project.db/formatted_input/batchid=1/file
.txt
-rw-r--r-- 1 acadgild supergroup 1276 2017-12-27 15:29 /user/hive/warehouse/project.db/formatted_input/batchid=1/part
-m-000000
[acadgild@localhost ~]$
```

Step 5: Perform Data Enrichment and Cleaning

Below data_enrichment_filtering_schema.sh script calls create_hive_hbase_lookup.hql which creates hive tables on top of hbase tables

```
data_enrichment_filtering_schema.sh
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current_batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_${batchid}

#echo "Creating hive tables on top of hbase tables for data enrichment and filtering..." >> $LOGFILE

hive -f /home/acadgild/project/scripts/create_hive_hbase_lookup.hql
```

create_hive_hbase_lookup.hql

Create Hive lookup tables and save lookup table **subscribed_users** to Local FS.

```
data_enrichment_filtering_schema.sh create_hive_hbase_lookup.hql
USE project;

CREATE EXTERNAL TABLE IF NOT EXISTS station_geo_map
(
  station_id STRING,
  geo_cd STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key,geo:geo_cd")
tblproperties("hbase.table.name"="station-geo-map");

CREATE EXTERNAL TABLE IF NOT EXISTS subscribed_users
(
  user_id STRING,
  subscn_start_dt STRING,
  subscn_end_dt STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key,subscn:startdt,subscn:enddt")
tblproperties("hbase.table.name"="subscribed-users");

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/project/exporteddata/subscribeduser'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
SELECT * FROM subscribed_users;

CREATE EXTERNAL TABLE IF NOT EXISTS song_artist_map
(
  song_id STRING,
```

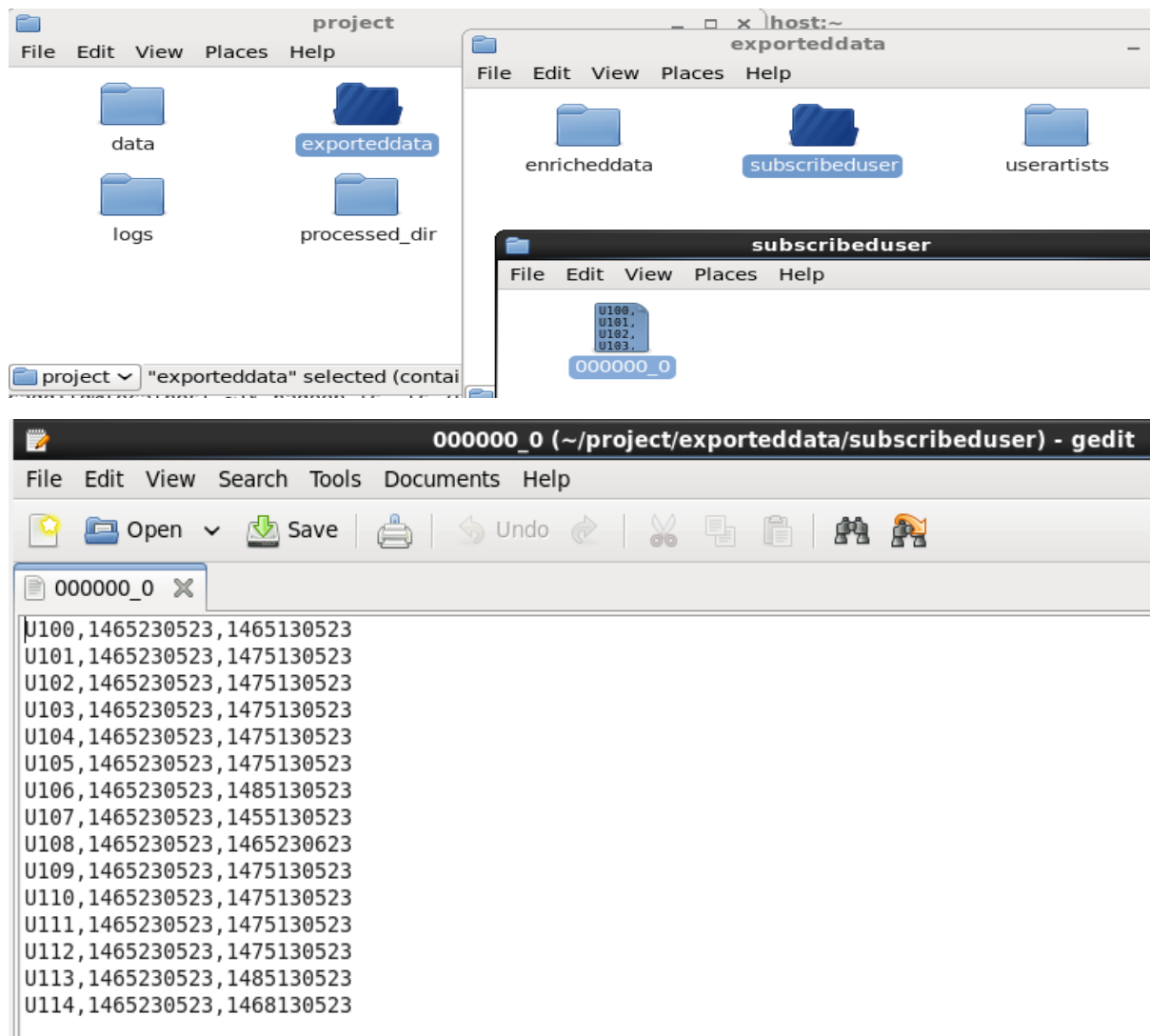
Below is screenshot of execution of [data_enrichment_filtering_schema.sh](#)

```
acadgild@localhost:~
File Edit View Search Terminal Help
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/data_enrichment_filtering_schema.sh

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-0.14.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/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]
OK
Time taken: 1.466 seconds
OK
Time taken: 2.198 seconds
OK
Time taken: 0.297 seconds
Query ID = acadgild_20171006013838_b459e5fd-ce6e-44a4-b846-3a962417819a

hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/project/exporteddata/subscribeduser'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> STORED AS TEXTFILE
> SELECT * FROM subscribed_users;
Query ID = acadgild_20171227155656_1a907ce6-ae6-4d4c-a450-aa851c561435
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_1514346487452_0005, Tracking URL = http://localhost:8088/proxy/application_1514346487452_0005/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job -kill job_1514346487452_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-12-27 15:57:23,760 Stage-1 map = 0%, reduce = 0%
2017-12-27 15:58:13,620 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.14 sec
MapReduce Total cumulative CPU time: 9 seconds 140 msec
Ended Job = job_1514346487452_0005
Copying data to local directory /home/acadgild/project/exporteddata/subscribeduser
Copying data to local directory /home/acadgild/project/exporteddata/subscribeduser
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 9.14 sec HDFS Read: 276 HDFS Write: 405 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 140 msec
OK
Time taken: 129.233 seconds
hive>
```


Output of the saved table subscribed users in the Local FS:



The screenshot displays a file explorer window for the 'project' directory. It shows subdirectories: 'data', 'logs', 'exporteddata', and 'processed_dir'. The 'exporteddata' directory is expanded, showing 'enricheddata', 'subscribeduser', and 'userartists'. The 'subscribeduser' directory is further expanded, showing a file named '000000_0'. Below the file explorer, a gedit window titled '000000_0 (~/.project/exporteddata/subscribeduser) - gedit' shows the contents of the file. The file contains 15 rows of data, each representing a user with a unique ID and two associated numerical values.

User ID	Value 1	Value 2
U100	1465230523	1465130523
U101	1465230523	1475130523
U102	1465230523	1475130523
U103	1465230523	1475130523
U104	1465230523	1475130523
U105	1465230523	1475130523
U106	1465230523	1485130523
U107	1465230523	1455130523
U108	1465230523	1465230623
U109	1465230523	1475130523
U110	1465230523	1475130523
U111	1465230523	1475130523
U112	1465230523	1475130523
U113	1465230523	1485130523
U114	1465230523	1468130523

Output in hive

```
hive> SELECT * FROM subscribed_users;
OK
U100      1465230523      1465130523
U101      1465230523      1475130523
U102      1465230523      1475130523
U103      1465230523      1475130523
U104      1465230523      1475130523
U105      1465230523      1475130523
U106      1465230523      1485130523
U107      1465230523      1455130523
U108      1465230523      1465230623
U109      1465230523      1475130523
U110      1465230523      1475130523
U111      1465230523      1475130523
U112      1465230523      1475130523
U113      1465230523      1485130523
U114      1465230523      1468130523
Time taken: 5.782 seconds, Fetched: 15 row(s)
hive>
```

```

hive> SELECT * FROM station_geo_map;
OK
ST400    A
ST401    AU
ST402    AP
ST403    J
ST404    E
ST405    A
ST406    AU
ST407    AP
ST408    E
ST409    E
ST410    A
ST411    A
ST412    AP
ST413    J
ST414    E
Time taken: 0.823 seconds, Fetched: 15 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: 0.712 seconds, Fetched: 10 row(s)

```

Below script **data_enrichment.sh** filters and enriches the data of **formatted_input** table present in hive

```

data_enrichment.sh
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_${batchid}

VALIDDIR=/home/acadgild/project/processed_dir/valid/batch_${batchid}
INVALIDDIR=/home/acadgild/project/processed_dir/invalid/batch_${batchid}

echo "Running hive script for data enrichment and filtering..." >> $LOGFILE

hive -hiveconf batchid=${batchid} -f /home/acadgild/project/scripts/data_enrichment.hql

if [ ! -d "$VALIDDIR" ]
then
    mkdir -p "$VALIDDIR"
fi
|
if [ ! -d "$INVALIDDIR" ]
then
    mkdir -p "$INVALIDDIR"
fi

echo "Copying valid and invalid records in local file system..." >> $LOGFILE

hadoop fs -get /user/hive/warehouse/project.db/enriched_data/batchid=${batchid}/status=pass/* $VALIDDIR
hadoop fs -get /user/hive/warehouse/project.db/enriched_data/batchid=${batchid}/status=fail/* $INVALIDDIR

echo "Deleting older valid and invalid records from local file system..." >> $LOGFILE

find /home/acadgild/project/processed_dir/ -mtime +7 -exec rm {} \;

```

Above script calls **data_enrichment.hql** scripts

```
data_enrichment.sh x data_enrichment.hql x
i.station_id,
IF (i.song_end_type IS NULL,3,i.song_end_type) AS song_end_type,
IF (i.like IS NULL,0,i.like) AS like,
IF (i.dislike IS NULL,0,i.dislike) AS dislike,
i.batchid,
IF((i.like=1 AND i.dislike=1)
OR i.user_id IS NULL
OR i.song_id IS NULL
OR i.timestamp IS NULL
OR i.start_ts IS NULL
OR i.end_ts IS NULL
OR i.geo_cd IS NULL
OR i.user_id=''
OR i.song_id=''
OR i.timestamp=''
OR i.start_ts=''
OR i.end_ts=''
OR i.geo_cd=''
OR sg.geo_cd IS NULL
OR sg.geo_cd=''
OR sa.artist_id IS NULL
OR sa.artist_id='', 'fail', 'pass') AS status
FROM formatted_input i
LEFT OUTER JOIN station_geo_map sg ON i.station_id = sg.station_id
LEFT OUTER JOIN song_artist_map sa ON i.song_id = sa.song_id
WHERE i.batchid=${hiveconf:batchid};

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/project/exporteddata/enricheddata'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
SELECT * FROM enriched_data;
```

Below screenshots shows executions of data_enrichment.sh

```
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/data_enrichment.sh
/usr/local/hive/bin/hive-config.sh: line 1: syntax error near unexpected token `('
/usr/local/hive/bin/hive-config.sh: line 1: `# Licensed to the Apache Software Foundation (ASF) under one or more'
```

```
Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-0.14.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/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]
OK
Time taken: 3.14 seconds
OK
Time taken: 3.935 seconds
Query ID = acadgild_20171227161818_fe943e02-c57a-4d60-ba8c-5435777a5d96
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
```

```
-----
Loading data to table project.enriched_data partition (batchid=null, status=null)
Time taken for load dynamic partitions : 1573
Loading partition {batchid=1, status=pass}
Loading partition {batchid=1, status=pass}
Time taken for adding to write entity : 38
Partition project.enriched_data{batchid=1, status=pass} stats: [numFiles=1, numRows=16, totalSize=1384, rawDataSize=10928]
MapReduce Jobs Launched:
Partition project.enriched_data{batchid=1, status=pass} stats: [numFiles=1, numRows=24, totalSize=1460, rawDataSize=17568]
Stage-Stage-1: Map: 3 Reduce: 1 Cumulative CPU: 28.84 sec HDFS Read: 3277 HDFS Write: 3067 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 1 Cumulative CPU: 25.58 sec HDFS Read: 3746 HDFS Write: 3011 SUCCESS
Total MapReduce CPU Time Spent: 54 seconds 420 msec
OK
Time taken: 443.407 seconds
Query ID = acadgild_20171227162626_15a5a6f4-6f3a-447f-8685-37f4461bcd1b
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_1514346487452_0008, Tracking URL = http://localhost:8088/proxy/application_1514346487452_0008/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job -kill job_1514346487452_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-12-27 16:26:50,534 Stage-1 map = 0%, reduce = 0%
2017-12-27 16:27:21,666 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.54 sec
MapReduce Total cumulative CPU time: 5 seconds 540 msec
Ended Job = job_1514346487452_0008
Copying data to local directory /home/acadgild/project/exporteddata/enricheddata
Copying data to local directory /home/acadgild/project/exporteddata/enricheddata
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 5.54 sec HDFS Read: 4554 HDFS Write: 2782 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 540 msec
OK
Time taken: 75.007 seconds
17/12/27 16:27:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/27 16:27:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ █
```

Output show the enriched_data tables created in hive

```
hive> use project;
OK
Time taken: 0.32 seconds
hive> show tables;
OK
enriched_data
formatted_input
song_artist_map
station_geo_map
subscribed_users
user_artists
Time taken: 0.786 seconds, Fetched: 6 row(s)
hive> █
```

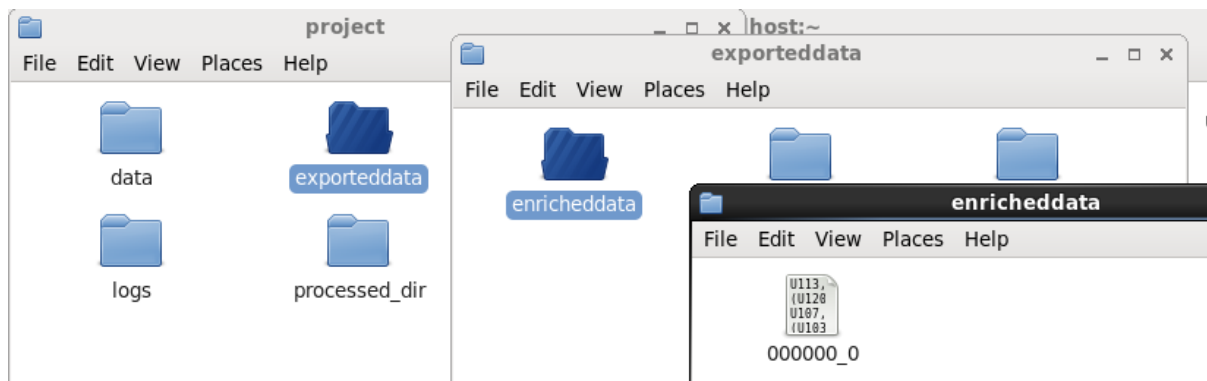
Below screenshot shows that failed (invalid) and passed (valid) data are segregated successfully

```
hive> SELECT * FROM enriched_data;
OK
U113 S200 A300 1465230523 1475130523 1465130523 J ST413 3 1 1 1 fail
(U120 S201 A301 1494297562 1465490556 1468094889 A ST410 3 0 0 1 fail
U107 S202 A302 1495130523 1465230523 1465230523 NULL ST415 0 1 1 1 fail
(U103 S202 A302 1465490556 1465490556 1465490556 NULL ST415 2 1 0 1 fail
U106 S202 A302 1465230523 1465130523 1465130523 E ST408 0 1 1 1 fail
S203 A303 1495130523 1475130523 1465230523 A ST400 0 0 1 1 fail
U110 S203 A303 1465230523 1465130523 1485130523 NULL ST415 0 1 1 1 fail
(U113 S204 A304 1494297562 1494297562 1465490556 NULL ST415 3 0 0 1 fail
U100 S204 A304 1495130523 1475130523 1465130523 E ST408 2 1 1 1 fail
U108 S205 A301 1465130523 1465130523 1475130523 A ST410 2 1 0 1 fail
U111 S206 A302 1465130523 1465130523 1485130523 NULL ST415 0 1 1 1 fail
U114 S207 A303 1465130523 1465230523 1475130523 NULL ST415 3 1 0 1 fail
U102 S207 A303 1465230523 1485130523 1465230523 J ST403 3 1 1 1 fail
U118 S208 A304 1475130523 1465130523 1465230523 NULL ST415 3 0 0 1 fail
U119 S208 A304 1495130523 1465230523 1465230523 NULL ST415 3 0 0 1 fail
U107 S210 NULL 1475130523 1485130523 1485130523 E ST404 2 1 0 1 fail
(U115 S200 A300 1465490556 1494297562 1465490556 E ST404 3 0 0 1 pass
(U100 S200 A300 1494297562 1494297562 1494297562 A ST410 3 1 0 1 pass
(U108 S200 A300 1468094889 1462863262 1468094889 E ST414 0 0 0 1 pass
(U107 S202 A302 1494297562 1468094889 1462863262 E ST409 0 0 0 1 pass
U101 S202 A302 1465230523 1465130523 1475130523 AU ST401 0 0 1 1 pass
(U110 S202 A302 1494297562 1494297562 1468094889 AP ST402 2 1 0 1 pass
U118 S202 A302 1495130523 1465230523 1465230523 A ST410 1 0 0 1 pass
U104 S202 A302 1465230523 1475130523 1465130523 E ST409 2 0 1 1 pass
(U102 S203 A303 1465490556 1465490556 1494297562 E ST404 2 0 0 1 pass
(U109 S203 A303 1462863262 1494297562 1468094889 A ST405 1 1 0 1 pass
(U113 S203 A303 1465490556 1465490556 1462863262 A ST405 0 0 0 1 pass
(U113 S203 A303 1462863262 1468094889 1494297562 E ST408 2 0 0 1 pass
U105 S203 A303 1475130523 1465230523 1465130523 E ST408 2 0 1 1 pass
(U113 S203 A303 1465490556 1465490556 1465490556 AP ST402 1 1 0 1 pass
(U111 S204 A304 1465490556 1465490556 1468094889 A ST410 3 1 0 1 pass
(U103 S204 A304 1468094889 1494297562 1465490556 A ST411 2 1 0 1 pass
(U106 S205 A301 1462863262 1462863262 1494297562 AP ST407 2 1 0 1 pass
(U106 S206 A302 1494297562 1465490556 1462863262 A ST405 3 1 0 1 pass
(U106 S206 A302 1494297562 1465490556 1462863262 A ST405 3 1 0 1 pass
U106 S206 A302 1494297562 1465490556 1462863262 A ST405 3 1 0 1 pass
U120 S206 A302 1495130523 1485130523 1465130523 E ST414 0 0 0 1 pass
U105 S207 A303 1465230523 1485130523 1465130523 A ST400 2 0 1 1 pass
( S208 A304 1465490556 1494297562 1465490556 A ST411 1 0 0 1 pass
U116 S208 A304 1465230523 1485130523 1475130523 J ST413 1 0 1 1 pass
(U101 S208 A304 1462863262 1468094889 1462863262 E ST408 0 1 0 1 pass
(U114 S209 A305 1465490556 1462863262 1494297562 A ST411 2 1 0 1 pass
Time taken: 5.711 seconds, Fetched: 40 row(s)
hive> █
```

This shows the enriched_data created in hive warehouse

```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/warehouse/project.db/enriched_data/batchid=1
17/12/27 16:42:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 2 items
drwxr-xr-x - acadgild supergroup 0 2017-12-27 16:25 /user/hive/warehouse/project.db/enriched_data/batchid=1/status
=fail
drwxr-xr-x - acadgild supergroup 0 2017-12-27 16:25 /user/hive/warehouse/project.db/enriched_data/batchid=1/status
=pass
[acadgild@localhost ~]$ █
```

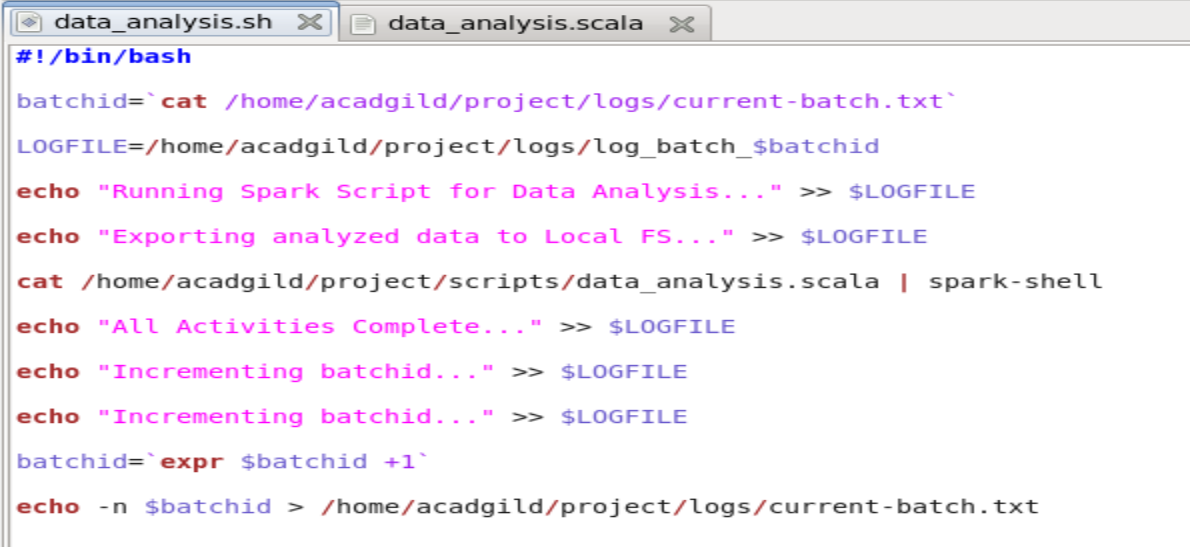
Output showing enricheddata which was stored in local file system



```
000000_0 (~/.project/exporteddata/enricheddata) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
000000_0 x
U113,S200,A300,1465230523,1475130523,1465130523,J,ST413,3,1,1,1,fail
(U120,S201,A301,1494297562,1465490556,1468094889,A,ST410,3,0,0,1,fail
U107,S202,A302,1495130523,1465230523,1465230523,\N,ST415,0,1,1,1,fail
(U103,S202,A302,1465490556,1465490556,1465490556,\N,ST415,2,1,0,1,fail
U106,S202,A302,1465230523,1465130523,1465130523,E,ST408,0,1,1,1,fail
,S203,A303,1495130523,1475130523,1465230523,A,ST400,0,0,1,1,fail
U110,S203,A303,1465230523,1465130523,1485130523,\N,ST415,0,1,1,1,fail
(U113,S204,A304,1494297562,1494297562,1465490556,\N,ST415,3,0,0,1,fail
U100,S204,A304,1495130523,1475130523,1465130523,E,ST408,2,1,1,1,fail
U108,S205,A301,1465130523,1465130523,1475130523,A,ST410,2,1,0,1,fail
U111,S206,A302,1465130523,1465130523,1485130523,\N,ST415,0,1,1,1,fail
U114,S207,A303,1465130523,1465230523,1475130523,\N,ST415,3,1,0,1,fail
U102,S207,A303,1465230523,1485130523,1465230523,J,ST403,3,1,1,1,fail
U118,S208,A304,1475130523,1465130523,1465230523,\N,ST415,3,0,0,1,fail
U119,S208,A304,1495130523,1465230523,1465230523,\N,ST415,3,0,0,1,fail
U107,S210,\N,1475130523,1485130523,1485130523,E,ST404,2,1,0,1,fail
(U115,S200,A300,1465490556,1494297562,1465490556,E,ST404,3,0,0,1,pass
(U100,S200,A300,1494297562,1494297562,1494297562,A,ST410,3,1,0,1,pass
(U108,S200,A300,1468094889,1462863262,1468094889,E,ST414,0,0,0,1,pass
(U107,S202,A302,1494297562,1468094889,1462863262,E,ST409,0,0,0,1,pass
U101,S202,A302,1465230523,1465130523,1475130523,AU,ST401,0,0,1,1,pass
(U110,S202,A302,1494297562,1494297562,1468094889,AP,ST402,2,1,0,1,pass
U118,S202,A302,1495130523,1465230523,1465230523,A,ST410,1,0,0,1,pass
U104,S202,A302,1465230523,1475130523,1465130523,E,ST409,2,0,1,1,pass
(U102,S203,A303,1465490556,1465490556,1494297562,E,ST404,2,0,0,1,pass
(U109,S203,A303,1462863262,1494297562,1468094889,A,ST405,1,1,0,1,pass
(U113,S203,A303,1465490556,1465490556,1462863262,A,ST405,0,0,0,1,pass
(U113,S203,A303,1462863262,1468094889,1494297562,E,ST408,2,0,0,1,pass
U105,S203,A303,1475130523,1465230523,1465130523,E,ST408,2,0,1,1,pass
(U113,S203,A303,1465490556,1465490556,1465490556,AP,ST402,1,1,0,1,pass
(U113,S204,A304,1465490556,1465490556,1468094889,A,ST410,3,1,0,1,pass
```

Step 6: Perform Data Analysis

Below data_analysis.sh scripts executes the data_analysis.scala scripts. Once the executions successfully completed the batch id is incremented in the log file.

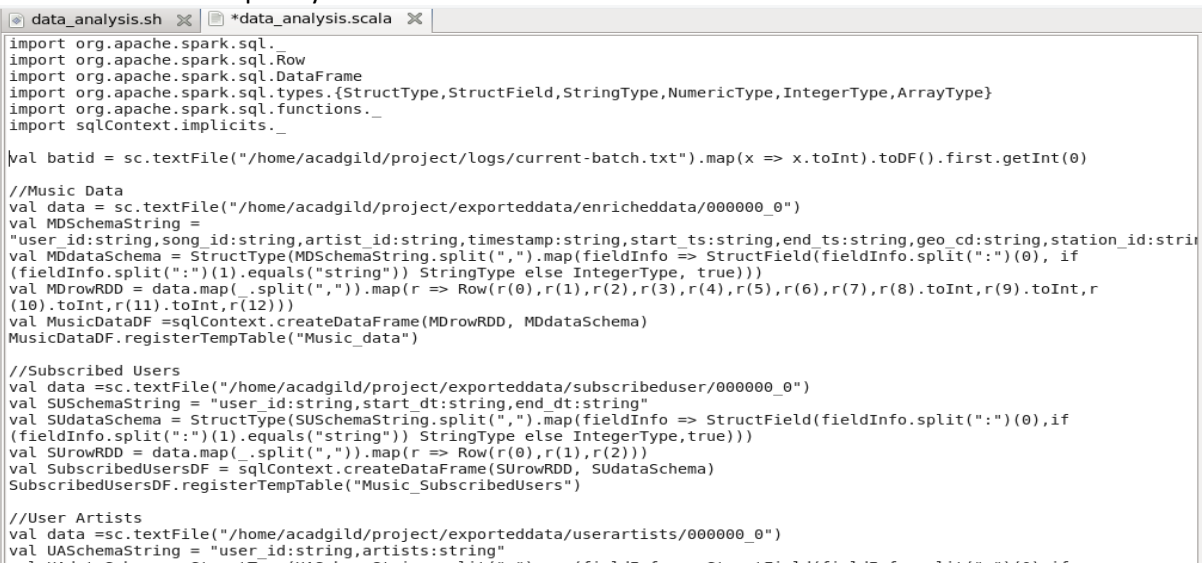


```
data_analysis.sh X data_analysis.scala X
#!/bin/bash
batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
echo "Running Spark Script for Data Analysis..." >> $LOGFILE
echo "Exporting analyzed data to Local FS..." >> $LOGFILE
cat /home/acadgild/project/scripts/data_analysis.scala | spark-shell
echo "All Activities Complete..." >> $LOGFILE
echo "Incrementing batchid..." >> $LOGFILE
echo "Incrementing batchid..." >> $LOGFILE
batchid=`expr $batchid +1`
echo -n $batchid > /home/acadgild/project/logs/current-batch.txt
```

Below is the file **data_analysis.scala** that will perform the data analysis for the given problem statements on the data that was saved to the Local FS.

Initialization:

- Import Row, DataFrame, Structure type and function dependencies needed to perform analysis.
- Get the batchid from the batch file and store it in the variable **batid**
- Get the data that was exported and saved in the Local FS from the steps above i.e. enriched_data, subscribed_user and user_artists and perform the foll. on each of them:
- Create the schema for the data
- Create a DataFrame from the schema and data
- Create a temporary table from the DataFrame created



```
data_analysis.sh X *data_analysis.scala X
import org.apache.spark.sql._
import org.apache.spark.sql.Row
import org.apache.spark.sql.DataFrame
import org.apache.spark.sql.types.{StructType, StructField, StringType, NumericType, IntegerType, ArrayType}
import org.apache.spark.sql.functions._
import sqlContext.implicits._

val batid = sc.textFile("/home/acadgild/project/logs/current-batch.txt").map(x => x.toInt).toDF().first.getInt(0)

//Music Data
val data = sc.textFile("/home/acadgild/project/exporteddata/enricheddata/000000_0")
val MDSchemaString =
"user_id:string,song_id:string,artist_id:string,timestamp:string,start_ts:string,end_ts:string,geo_cd:string,station_id:string"
val MDdataSchema = StructType(MDSchemaString.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0), if
(fieldInfo.split(":")(1).equals("string")) StringType else IntegerType, true)))
val MDrowRDD = data.map(_.split(",")).map(r => Row(r(0),r(1),r(2),r(3),r(4),r(5),r(6),r(7),r(8).toInt,r(9).toInt,r
(10).toInt,r(11).toInt,r(12)))
val MusicDataDF = sqlContext.createDataFrame(MDrowRDD, MDdataSchema)
MusicDataDF.registerTempTable("Music_data")

//Subscribed Users
val data = sc.textFile("/home/acadgild/project/exporteddata/subscribeduser/000000_0")
val SUSchemaString = "user_id:string,start_dt:string,end_dt:string"
val SUDataSchema = StructType(SUSchemaString.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0), if
(fieldInfo.split(":")(1).equals("string")) StringType else IntegerType, true)))
val SUrowRDD = data.map(_.split(",")).map(r => Row(r(0),r(1),r(2)))
val SubscribedUsersDF = sqlContext.createDataFrame(SUrowRDD, SUDataSchema)
SubscribedUsersDF.registerTempTable("Music_SubscribedUsers")

//User Artists
val data = sc.textFile("/home/acadgild/project/exporteddata/userartists/000000_0")
val UASchemaString = "user_id:string,artists:string"
```


Below screenshots shows execution of data analysis.sh

```
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/data_analysis.sh
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Using Spark's repl log4j profile: org/apache/spark/log4j-defaults-repl.properties
To adjust logging level use sc.setLogLevel("INFO")
Welcome to
```



```
Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_65)
Type in expressions to have them evaluated.
Type :help for more information.
17/12/27 15:26:04 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 10.0.2.15
instead (on interface eth6)
17/12/27 15:26:04 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Spark context available as sc.
17/12/27 15:26:23 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/27 15:26:25 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/27 15:26:52 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 1.2.0
17/12/27 15:26:53 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
17/12/27 15:27:05 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/27 15:27:07 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
SQL context available as sqlContext.
```

Problem Statement 1:

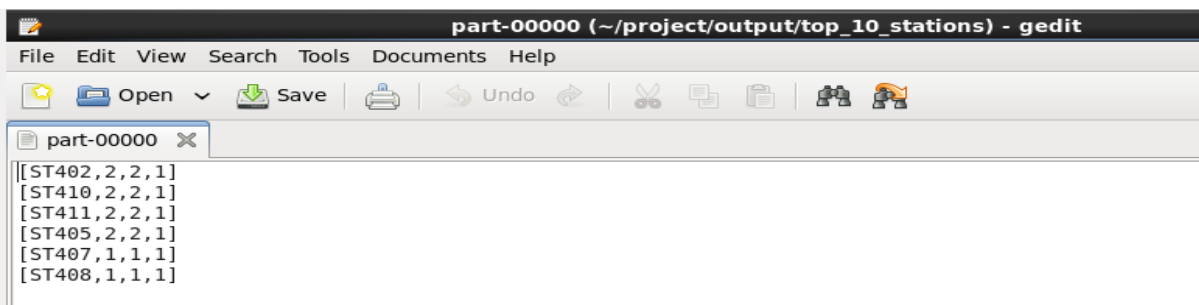
Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.

Code:

```
val Top10Stations = sqlContext.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:



Problem Statement 2:

Determine 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.

Code:

```
val users_behavior = sqlContext.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:

```
part-00000 (~/.project/output/users_behavior) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
part-00000 X
[SUBSCRIBED,157978279,1]
[UNSUBSCRIBED,98100227,1]
```

Problem Statement 3:

Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.

Code:

```
val connected_artists = sqlContext.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:

```
part-00000 (~/.project/output/connected_artists) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
part-00000 X
[A302,1,1]
[A300,1,1]
```

Problem Statement 4:

Determine 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.

Code:

```
val top_10_royalty_songs = sqlContext.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:

```
part-00000 (~/.project/output/top_10_royalty_songs) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
part-00000 X
[S202,41434300,1]
[S209,31434300,1]
[S205,31434300,1]
[S204,31411339,1]
[S203,28829967,1]
[S206,22627294,1]
[S208,5231627,1]
[S200,5231627,1]
```

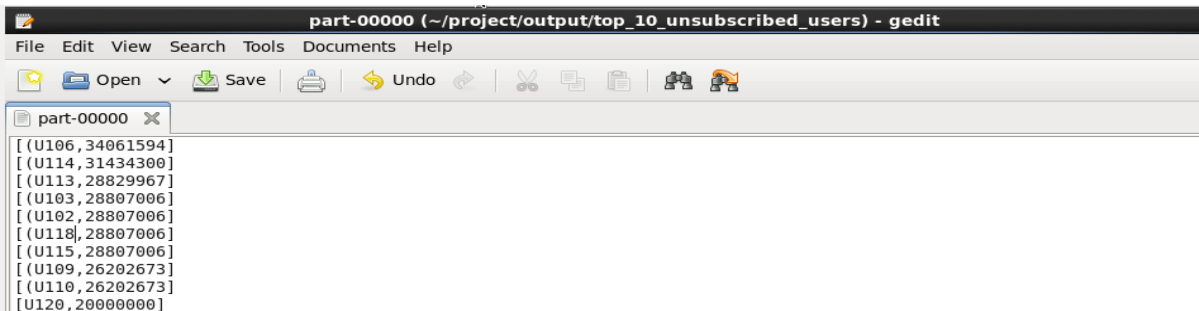

Problem Statement 5:

Determine top 10 unsubscribed users who listened to the songs for the longest duration.

Code:

```
val top_10_unsubscribed_users = sqlContext.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=$batchid 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")
```

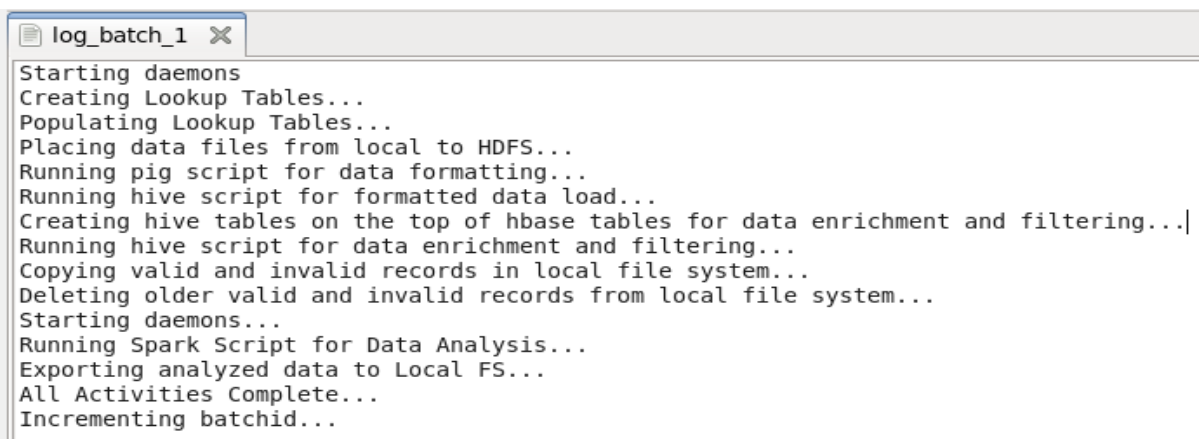
Output:



```
part-00000  
[(U106,34061594)  
[(U114,31434300)  
[(U113,28829967)  
[(U103,28807006)  
[(U102,28807006)  
[(U118,28807006)  
[(U115,28807006)  
[(U109,26202673)  
[(U110,26202673)  
[(U120,20000000)
```

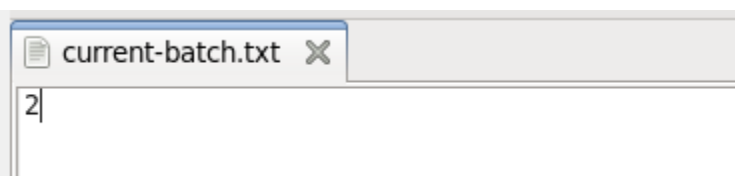
Post Analysis:

A view of the log file post analysis.



```
log_batch_1  
Starting daemons  
Creating Lookup Tables...  
Populating Lookup Tables...  
Placing data files from local to HDFS...  
Running pig script for data formatting...  
Running hive script for formatted data load...  
Creating hive tables on the top of hbase tables for data enrichment and filtering...  
Running hive script for data enrichment and filtering...  
Copying valid and invalid records in local file system...  
Deleting older valid and invalid records from local file system...  
Starting daemons...  
Running Spark Script for Data Analysis...  
Exporting analyzed data to Local FS...  
All Activities Complete...  
Incrementing batchid...
```

The batchid is incremented from 1 to 2:



```
current-batch.txt  
2
```

Note:

The project zip folder consist of following files:

Data, scripts, logs, lib, output, exporteddata,processed_dir

Lib folder contains piggybank.jar files to be used in execution of pig query.