Music Data Analysis

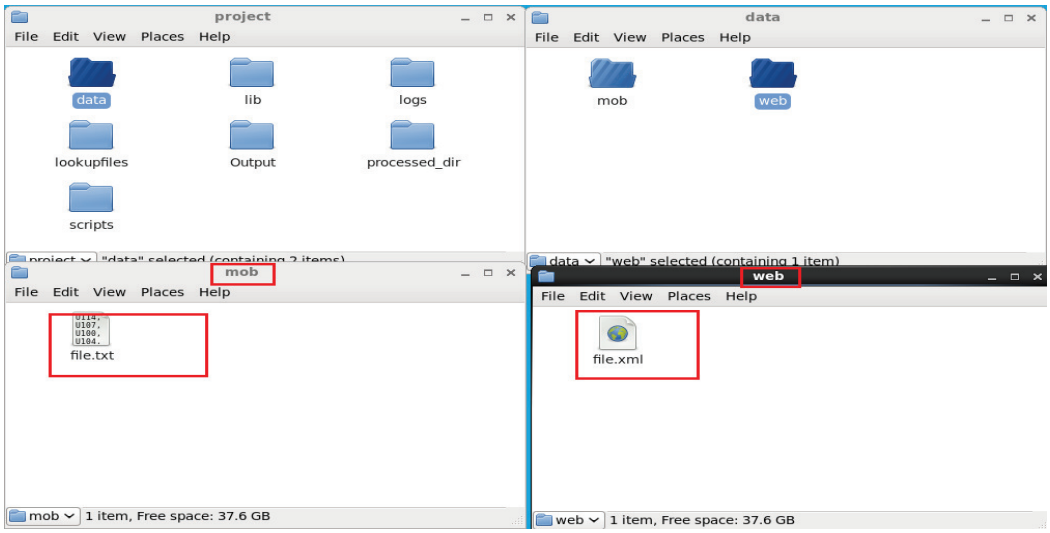
1. Introduction:

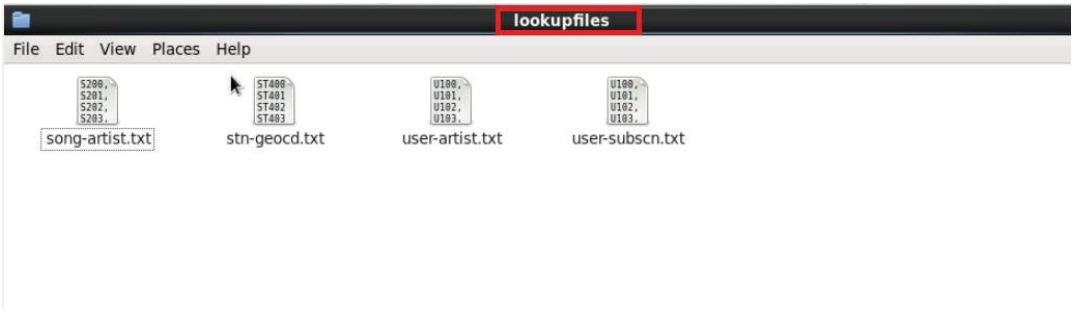
A leading music-catering company is planning to analyze large amount of data received from varieties of sources, namely mobile app and website to track the behaviour of users, classify users, calculate royalties associated with the song and make appropriate business strategies. The file server receives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

So as per periodic data for every 3 hours, we need to calculate the queries and return the results to business/customer.

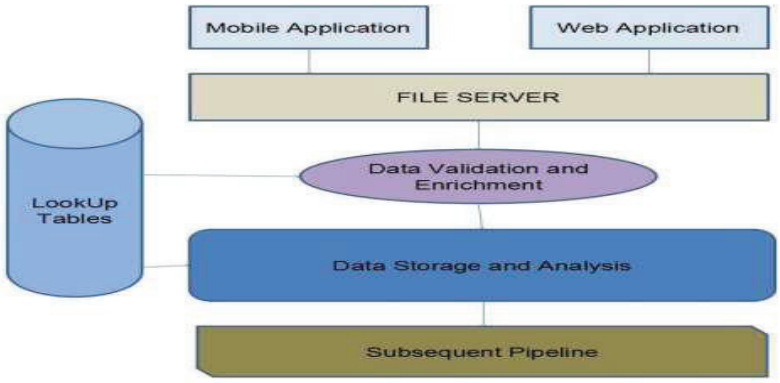
2. DATASET:

1. Data coming from web applications reside in /data/web and has xml format.
2. Data coming from mobile applications reside in /data/mob and has csv format.
3. Data present in lookup directory should be used in HBase.

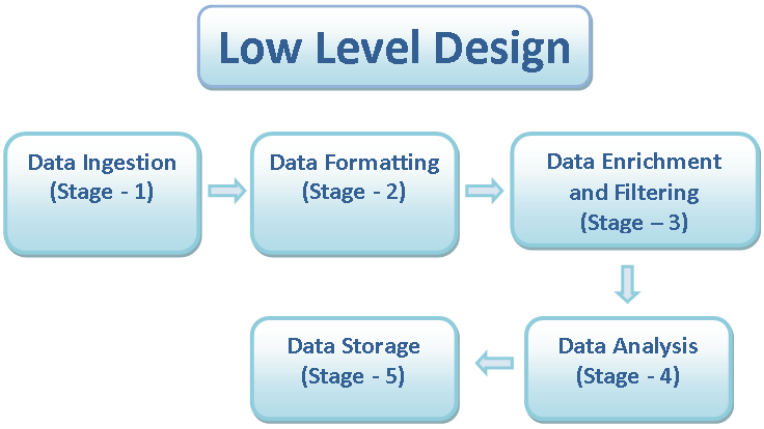




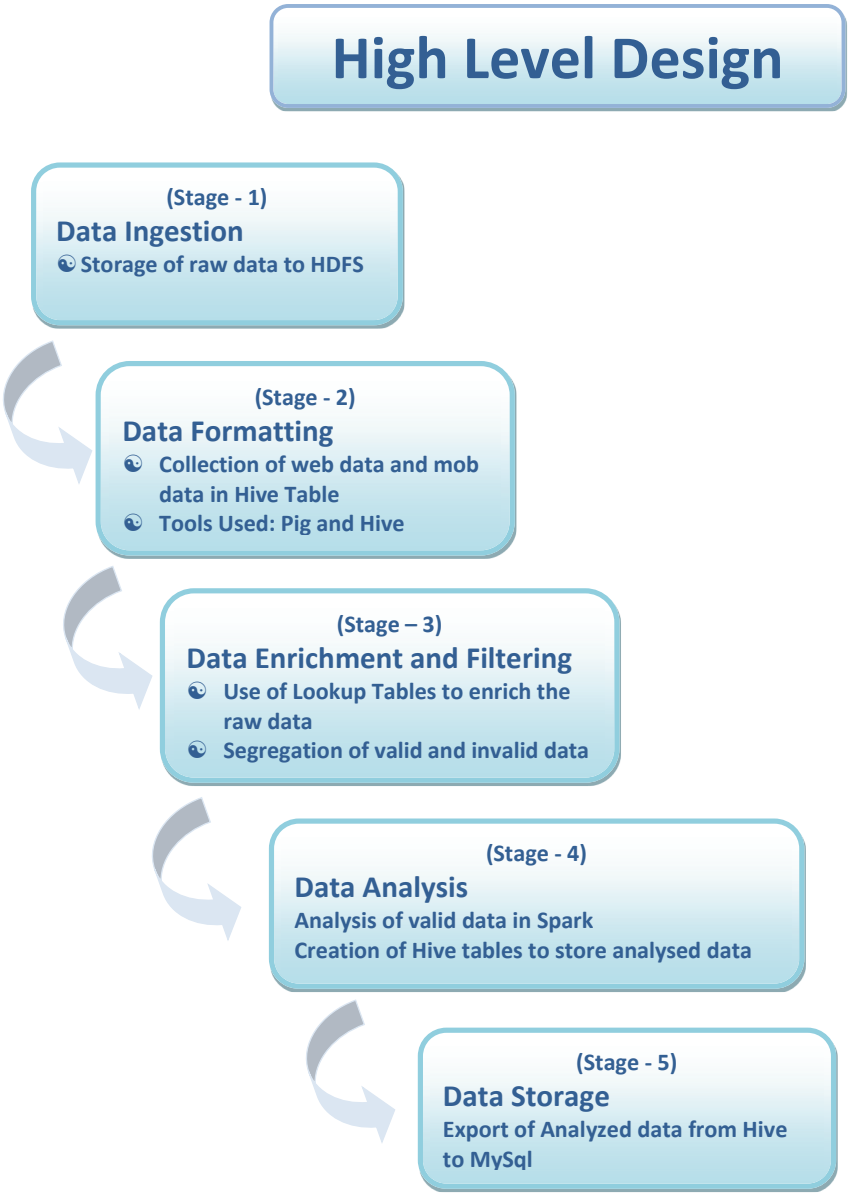
3. Flow of Operations:



4. **Low Level Design:**



5. **High Level Design**



6. Implementation:

a) Starting all daemons in Hadoop:

- In this script we will start all the daemons that are required to start the services like hive, hbase, MySQL etc.

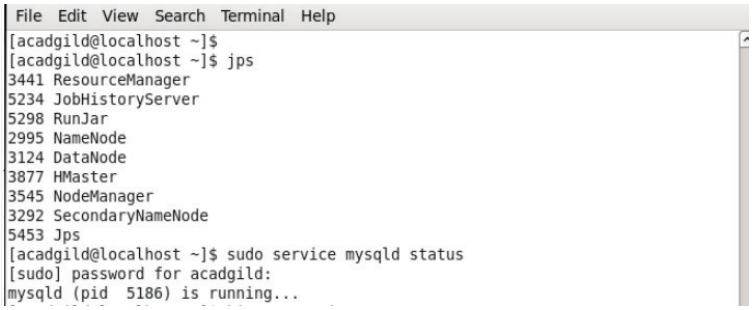
start-daemon.sh | data_export.sh | DataAnalysis.sh | wrapper.sh | log_batch_1 | log_batch_2

```
#!/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 775 /home/acadgild/project/logs/current-batch.txt
batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
echo "Starting daemons...." >> $LOGFILE

# To start the hadoop daemons:
/usr/local/hadoop-2.6.0/sbin/start-all.sh

# To start the Hmaster service:
/usr/local/hbase/bin/start-hbase.sh

# To start the Jobhistory server service:
mr-jobhistory-daemon.sh start historyserver

# To start mysql service:
sudo service mysqld start

# To start hive metastore:
hive --service metastore
```

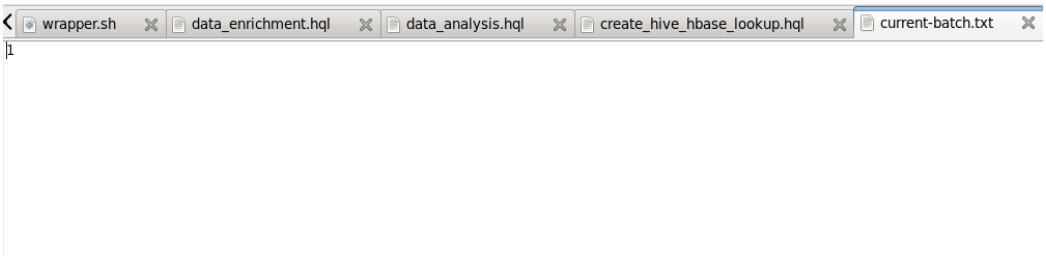
- Terminal with all the daemons started screenshot.

```
[acadgild@localhost scripts]$ sh start-daemons.sh
Batch file found!
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
17/09/11 12:42:30 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
acadgild@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-aca
dgild-namenode-localhost.localdomain.out
acadgild@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-aca
dgild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
acadgild@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.6.0/logs/had
oop-acadgild-secondarynamenode-localhost.localdomain.out
17/09/11 12:43:16 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-acadgild-
resourcemanager-localhost.localdomain.out
acadgild@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-ac
adgild-nodemanager-localhost.localdomain.out
starting master, logging to /usr/local/hbase/logs/hbase-acadgild-master-localhos
t.localdomain.out
starting historyserver, logging to /usr/local/hadoop-2.6.0/logs/mapred-acadgild-
historyserver-localhost.localdomain.out
Starting mysqld:                                           [  OK  ]
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-standalon
e.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/li
b/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]
```

- With the help of **jps** command we can list out the services that are currently in active state.



The start-daemon.sh script will check whether the current-batch.txt file is available in the logs folder or not. If not it will create the file and dump value '1' in that file and create LOGFILE with the current batchid.



Lookup Tables creation in HBASE:

By using the populate-lookup.sh script we will create lookup tables in Hbase. These tables have to be used in data formatting, data enrichment and analysis stage.

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

These 4 lookup files are used to create Hbase tables using populate-lookup.sh script and then with the help of data_enrichment_filtering_schema.sh file we will create hive tables on the top of Hbase tables using create_hive_hbase_lookup.hql

Script.

b) Creating Lookup Tables :

File Edit View Search Tools Documents Help

Open Save Undo

populate-lookup.sh | create_hive_hbase_lookup.hql

```
#!/bin/bash

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

LOGFILE=/home/cloudera/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 "create 'user-artist-map', 'artists'" | hbase shell

echo "Populating Lookup Tables..." >> $LOGFILE

file="/home/cloudera/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/cloudera/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/cloudera/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"

file="/home/cloudera/acadgild/project/lookupfiles/user-artist.txt"
touch /home/cloudera/acadgild/project/lookupfiles/user-artist1.txt
chmod 775 /home/cloudera/acadgild/project/lookupfiles/user-artist1.txt
file1="/home/cloudera/acadgild/project/lookupfiles/user-artist1.txt"
```

File Edit View Search Tools Documents Help

Open Save Undo

populate-lookup.sh | create_hive_hbase_lookup.hql

```
file="/home/cloudera/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/cloudera/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/cloudera/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"

file="/home/cloudera/acadgild/project/lookupfiles/user-artist.txt"
touch /home/cloudera/acadgild/project/lookupfiles/user-artist1.txt
chmod 775 /home/cloudera/acadgild/project/lookupfiles/user-artist1.txt
file1="/home/cloudera/acadgild/project/lookupfiles/user-artist1.txt"
```

```
awk '$1=$1' FS="&" OFS=" " $file > $file1
num=1
while IFS= read -r line
do
 userid=`echo $line | cut -d',' -f1`
 artists=`echo $line | cut -d',' -f2`
 for row in $artists
 do
 #artist= 'echo $row | cut -d',' -f$num
 echo "put 'user-artist-map', 'userid', 'artists:artist$num', '$row'" | hbase shell
 let "num=num+1"
done
num=1
done <"$file"
```

By using the shell scripting I created 4 lookup tables in Hbase NoSQL Database.

Music Data Analysis

```
File  Edit  View  Search  Terminal  Help
2017-08-31 20:28:01,616 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

'ut 'song-artist-map', 'S209', 'artist:artistid', 'A305
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.
2017-08-31 20:28:03,442 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
0 row(s) in 0.2850 seconds

2017-08-31 20:28:07,451 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 'subscribed-users', 'U100', 'subscn:startdt', '1465230523'
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.
2017-08-31 20:28:09,229 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
0 row(s) in 0.2700 seconds

2017-08-31 20:28:13,179 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

'ut 'subscribed-users', 'U100', 'subscn:enddt', '1465130523
```

```
File  Edit  View  Search  Terminal  Help

2017-09-06 04:21:59,787 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

'ut 'user-artist-map','U112', 'artists:artist2','A301
0 row(s) in 0.2180 seconds

2017-09-06 04:22:06,838 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

put 'user-artist-map','U113', 'artists:artist1','A305'
0 row(s) in 0.2250 seconds

2017-09-06 04:22:14,019 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

'ut 'user-artist-map','U113', 'artists:artist2','A302
0 row(s) in 0.2180 seconds

2017-09-06 04:22:20,818 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

put 'user-artist-map','U114', 'artists:artist1','A300'
0 row(s) in 0.2260 seconds

2017-09-06 04:22:27,997 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

put 'user-artist-map','U114', 'artists:artist2','A301'
0 row(s) in 0.2220 seconds

2017-09-06 04:22:35,159 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

put 'user-artist-map','U114', 'artists:artist3','A302'
0 row(s) in 0.2270 seconds
```

Now we can see the lookup tables in Hbase shell terminal as shown below:

```
hbase(main):027:0> list
TABLE
song-artist-map
station-geo-map
subscribed-users
user-artist-map
4 row(s) in 0.0220 seconds

=> ["song-artist-map", "station-geo-map", "subscribed-users", "user-artist-map"]
hbase(main):028:0>
```

Music Data Analysis

```
File  Edit  View  Search  Terminal  Help
hbase(main):023:0> scan 'user-artist-map'
ROW                                                   COLUMN+CELL
 U100                                                 column=artists:artist1, timestamp=1504715474511, value=A300
 U100                                                 column=artists:artist2, timestamp=1504715481424, value=A301
 U100                                                 column=artists:artist3, timestamp=1504715488438, value=A302
 U101                                                 column=artists:artist1, timestamp=1504715495344, value=A301
 U101                                                 column=artists:artist2, timestamp=1504715502153, value=A302
 U102                                                 column=artists:artist1, timestamp=1504715508818, value=A302
 U103                                                 column=artists:artist1, timestamp=1504715516638, value=A303
 U103                                                 column=artists:artist2, timestamp=1504715523493, value=A301
 U103                                                 column=artists:artist3, timestamp=1504715530594, value=A302
 U104                                                 column=artists:artist1, timestamp=1504715537538, value=A304
 U104                                                 column=artists:artist2, timestamp=1504715544515, value=A301
 U105                                                 column=artists:artist1, timestamp=1504715551495, value=A305
 U105                                                 column=artists:artist2, timestamp=1504715558408, value=A301
 U105                                                 column=artists:artist3, timestamp=1504715565421, value=A302
 U106                                                 column=artists:artist1, timestamp=1504715572397, value=A301
 U106                                                 column=artists:artist2, timestamp=1504715579414, value=A302
 U107                                                 column=artists:artist1, timestamp=1504715586302, value=A302
 U108                                                 column=artists:artist1, timestamp=1504715593424, value=A300
 U108                                                 column=artists:artist2, timestamp=1504715600445, value=A303
 U108                                                 column=artists:artist3, timestamp=1504715607403, value=A304
 U109                                                 column=artists:artist1, timestamp=1504715614139, value=A301
 U109                                                 column=artists:artist2, timestamp=1504715620862, value=A303
 U110                                                 column=artists:artist1, timestamp=1504715627634, value=A302
 U110                                                 column=artists:artist2, timestamp=1504715634562, value=A301
 U111                                                 column=artists:artist1, timestamp=1504715641378, value=A303
 U111                                                 column=artists:artist2, timestamp=1504715648252, value=A301
 U112                                                 column=artists:artist1, timestamp=1504715655386, value=A304
 U112                                                 column=artists:artist2, timestamp=1504715662398, value=A301
 U113                                                 column=artists:artist1, timestamp=1504715679145, value=A305
 U113                                                 column=artists:artist2, timestamp=1504715691872, value=A302
 U114                                                 column=artists:artist1, timestamp=1504715698861, value=A300
 U114                                                 column=artists:artist2, timestamp=1504715705820, value=A301
 U114                                                 column=artists:artist3, timestamp=1504715712697, value=A302
15 row(s) in 0.1300 seconds

hbase(main):024:0> scan 'song-artist-map'
ROW                                                   COLUMN+CELL
 S200                                                 column=artist:artistid, timestamp=1504715148289, value=A300
 S201                                                 column=artist:artistid, timestamp=1504715155432, value=A301
 S202                                                 column=artist:artistid, timestamp=1504715162559, value=A302
 S203                                                 column=artist:artistid, timestamp=1504715169733, value=A303
 S204                                                 column=artist:artistid, timestamp=1504715177021, value=A304
 S205                                                 column=artist:artistid, timestamp=1504715184088, value=A301
 S206                                                 column=artist:artistid, timestamp=1504715191114, value=A302
 S207                                                 column=artist:artistid, timestamp=1504715197964, value=A303
 S208                                                 column=artist:artistid, timestamp=1504715204901, value=A304
 S209                                                 column=artist:artistid, timestamp=1504715211710, value=A305
10 row(s) in 0.0530 seconds
```

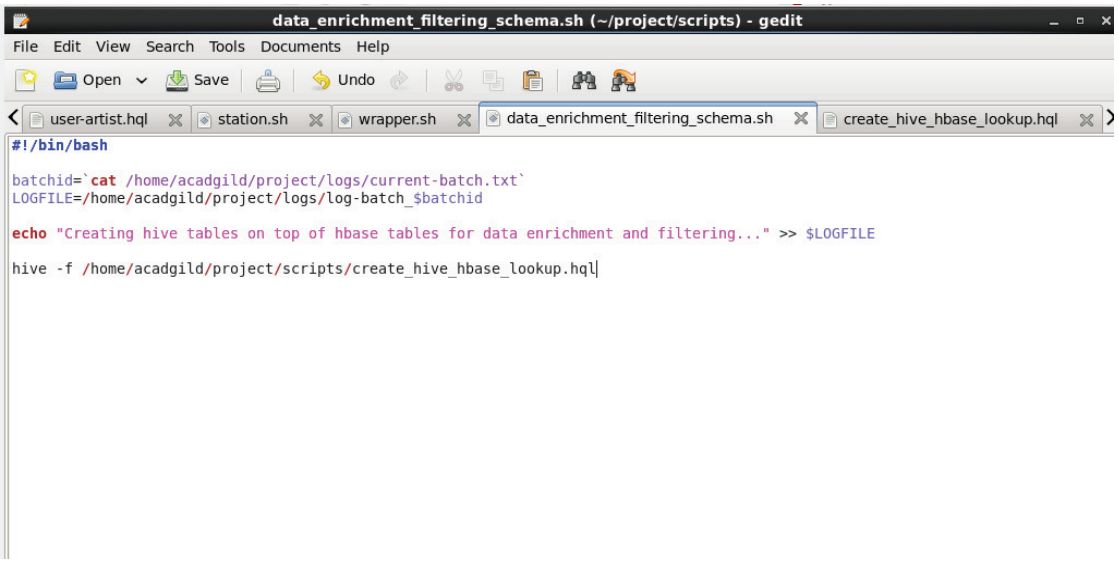
```
File  Edit  View  Search  Terminal  Help
hbase(main):025:0> scan 'station-geo-map'
ROW                                                   COLUMN+CELL
 ST400                                                column=geo:geo_cd, timestamp=1504715043586, value=A
 ST401                                                column=geo:geo_cd, timestamp=1504715050913, value=AU
 ST402                                                column=geo:geo_cd, timestamp=1504715057746, value=AP
 ST403                                                column=geo:geo_cd, timestamp=1504715064806, value=J
 ST404                                                column=geo:geo_cd, timestamp=1504715071617, value=E
 ST405                                                column=geo:geo_cd, timestamp=1504715078508, value=A
 ST406                                                column=geo:geo_cd, timestamp=1504715085747, value=AU
 ST407                                                column=geo:geo_cd, timestamp=1504715093025, value=AP
 ST408                                                column=geo:geo_cd, timestamp=1504715100009, value=E
 ST409                                                column=geo:geo_cd, timestamp=1504715107186, value=E
 ST410                                                column=geo:geo_cd, timestamp=1504715114039, value=A
 ST411                                                column=geo:geo_cd, timestamp=1504715121083, value=A
 ST412                                                column=geo:geo_cd, timestamp=1504715127794, value=AP
 ST413                                                column=geo:geo_cd, timestamp=1504715134632, value=J
 ST414                                                column=geo:geo_cd, timestamp=1504715141603, value=E
15 row(s) in 0.0480 seconds
```

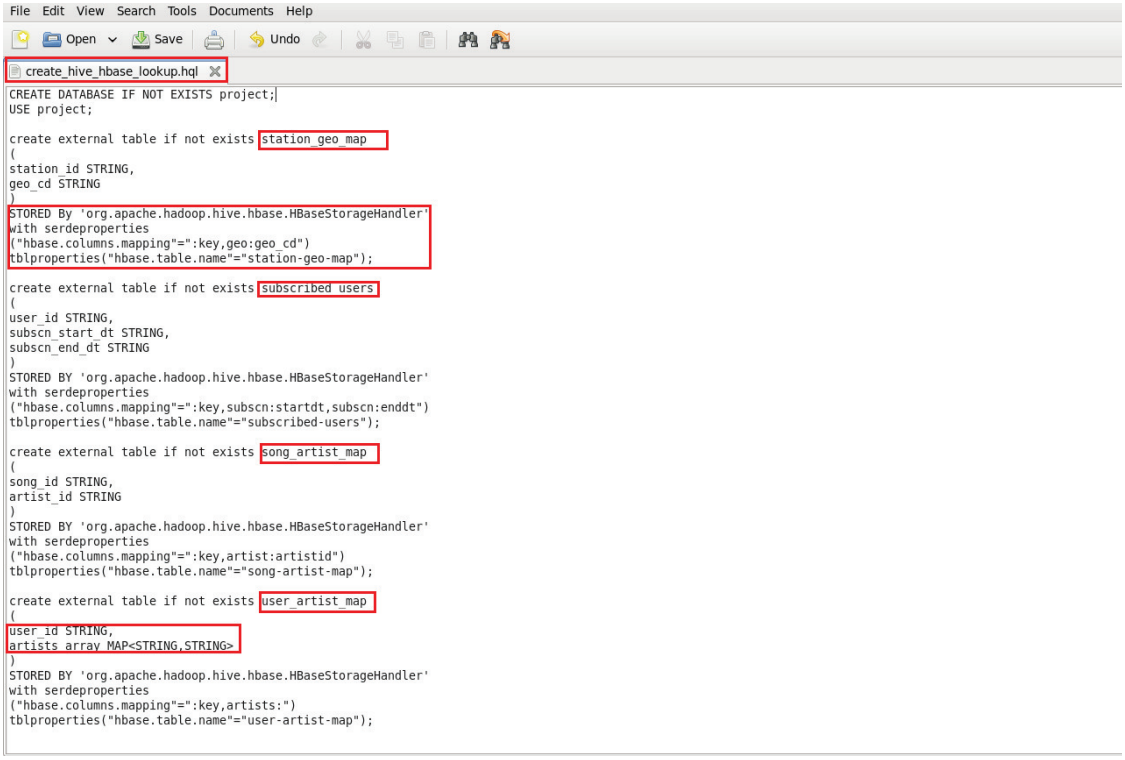
```
File  Edit  View  Search  Terminal  Help
 ST411                                               column=geo:geo_cd, timestamp=1504715121083, value=A
 ST412                                               column=geo:geo_cd, timestamp=1504715127794, value=AP
 ST413                                               column=geo:geo_cd, timestamp=1504715134632, value=J
 ST414                                               column=geo:geo_cd, timestamp=1504715141603, value=E
15 row(s) in 0.0480 seconds

hbase(main):026:0> scan 'subscribed-users'
ROW                                                  COLUMN+CELL
 U100                                                column=subscn:enddt, timestamp=1504715225823, value=1465130523
 U100                                                column=subscn:startdt, timestamp=1504715218773, value=1465230523
 U101                                                column=subscn:enddt, timestamp=1504715239964, value=1475130523
 U101                                                column=subscn:startdt, timestamp=1504715232971, value=1465230523
 U102                                                column=subscn:enddt, timestamp=1504715254171, value=1475130523
 U102                                                column=subscn:startdt, timestamp=1504715246875, value=1465230523
 U103                                                column=subscn:enddt, timestamp=1504715268395, value=1475130523
 U103                                                column=subscn:startdt, timestamp=1504715261047, value=1465230523
 U104                                                column=subscn:enddt, timestamp=1504715282780, value=1475130523
 U104                                                column=subscn:startdt, timestamp=1504715275591, value=1465230523
 U105                                                column=subscn:enddt, timestamp=1504715297219, value=1475130523
 U105                                                column=subscn:startdt, timestamp=1504715290108, value=1465230523
 U106                                                column=subscn:enddt, timestamp=1504715311103, value=1485130523
 U106                                                column=subscn:startdt, timestamp=1504715304199, value=1465230523
 U107                                                column=subscn:enddt, timestamp=1504715325247, value=1455130523
 U107                                                column=subscn:startdt, timestamp=1504715318471, value=1465230523
 U108                                                column=subscn:enddt, timestamp=1504715339075, value=1465230623
 U108                                                column=subscn:startdt, timestamp=1504715332078, value=1465230523
 U109                                                column=subscn:enddt, timestamp=1504715352911, value=1475130523
 U109                                                column=subscn:startdt, timestamp=1504715345909, value=1465230523
 U110                                                column=subscn:enddt, timestamp=1504715367212, value=1475130523
 U110                                                column=subscn:startdt, timestamp=1504715360147, value=1465230523
 U111                                                column=subscn:enddt, timestamp=1504715381001, value=1475130523
 U111                                                column=subscn:startdt, timestamp=1504715374138, value=1465230523
 U112                                                column=subscn:enddt, timestamp=1504715394817, value=1475130523
 U112                                                column=subscn:startdt, timestamp=1504715387953, value=1465230523
 U113                                                column=subscn:enddt, timestamp=1504715408681, value=1485130523
 U113                                                column=subscn:startdt, timestamp=1504715401790, value=1465230523
 U114                                                column=subscn:enddt, timestamp=1504715422632, value=1468130523
 U114                                                column=subscn:startdt, timestamp=1504715415809, value=1465230523
15 row(s) in 0.0590 seconds
```

Now Lookup table creation is completed. So now we need to link theses lookup tables in hive using the Hbase Storage Handler.

c) Creating Hive Tables on the top of Hbase:

In this stage with the help of Hbase storage handler & SerDe properties we are creating the hive external tables by matching the columns of Hbase tables to hive tables.





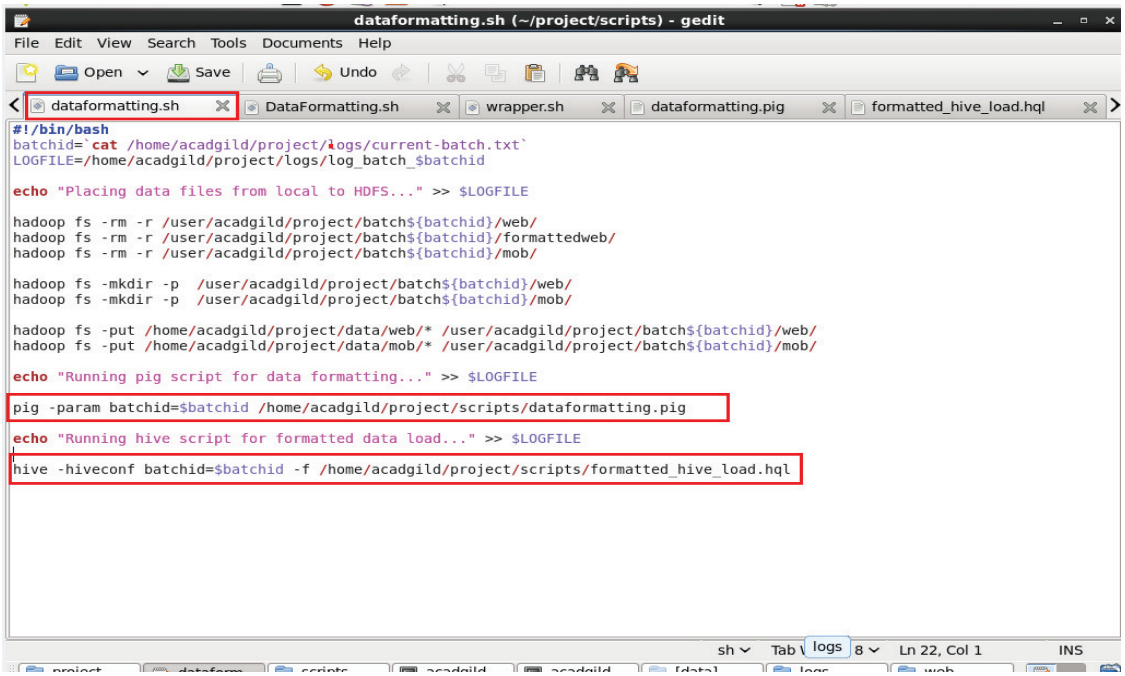
- In the below screenshot we can see tables getting created in hive by running the data_enrichement_filtering_schema.sh file.

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart scripts]$ ./data_enrichment_filtering_schema.sh

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
Time taken: 0.395 seconds
OK
Time taken: 0.102 seconds
OK
Time taken: 0.021 seconds
OK
Time taken: 0.018 seconds
OK
Time taken: 1.085 seconds
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
[cloudera@quickstart scripts]$
```

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> show databases;
OK
default
project
Time taken: 0.499 seconds, Fetched: 2 row(s)
hive> use project;
OK
Time taken: 0.047 seconds
hive> show tables;
OK
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.049 seconds, Fetched: 4 row(s)
hive>
```

d) Data Formatting:

In this stage we are merging the data coming from both web applications and mobile applications and create a common table for analyzing purpose and create partitioned data based on batchid, since we are running this scripts for every 3 hours.



I am running two scripts to format the data. They are: 1) Dataformatting.pig

2) Formatted_hive_load.hql

Pig script to parse the data from coming from web_data.xml to csv format and partition both web and mob data based on based on batch ID's

Music Data Analysis

dataformatting.pig (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

dataformatting.sh | DataFormatting.sh | wrapper.sh | dataformatting.pig | formatted_hive_load.hql

```
REGISTER /home/acadgild/project/lib/piggybank.jar;

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

 A = LOAD '/user/acadgild/project/batch${batchid}/web/' 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/batch${batchid}/formattedweb/' USING PigStorage(',');
```

Plain Text Tab logs 8 Ln 18, Col 44 INS

Hive script to load the parsed data from pig to hive and do analysis based on this data.

formatted_hive_load.hql (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

dataformatting.sh | DataFormatting.sh | wrapper.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});
```

Pl data t Tab Width: 8 Ln 24, Col 10 INS

In the below screenshot we can see the data both the scripts in action, first pig script will parse the data and then hive script will load the data into hive terminal successfully.

Music Data Analysis

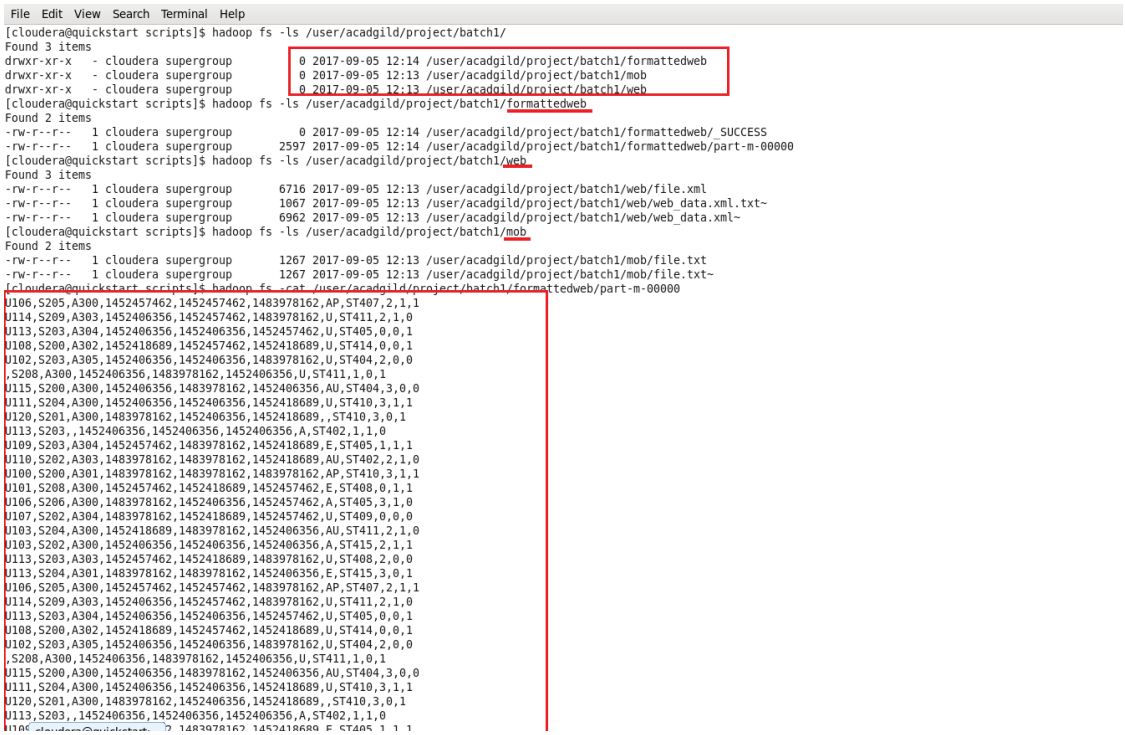
```
File  Edit  View  Search  Terminal  Help
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
[cloudera@quickstart scripts]$ ./dataformatting.sh
Deleted /user/acadgild/project/batch1/web
^Lrm: `/user/acadgild/project/batch1/formattedweb/': No such file or directory
Deleted /user/acadgild/project/batch1/mob
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2017-09-06 04:41:07,677 [main] INFO  org.apache.pig.Main - Apache Pig version 0.12.0-cdh5.10.0 (rexported) compiled Jan 20 2017, 12:05:43
2017-09-06 04:41:07,678 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/cloudera/acadgild/project/scripts/pig_1504698067660.log
2017-09-06 04:41:08,930 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/cloudera/.pigbootup not found
2017-09-06 04:41:09,049 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,050 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,050 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://quickstart.cloudera:8020
2017-09-06 04:41:09,501 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:8021
2017-09-06 04:41:09,564 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,565 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,599 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,601 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,652 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,652 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,688 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,690 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,737 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,738 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,790 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,790 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,818 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,818 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,850 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,850 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:09,895 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-09-06 04:41:09,896 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-09-06 04:41:10,498 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2017-09-06 04:41:10,547 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupE
timizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[Filter
}
2017-09-06 04:41:10,587 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutputformat.s
2017-09-06 04:41:10,694 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2017-09-06 04:41:10,725 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2017-09-06 04:41:10,725 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2017-09-06 04:41:10,826 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2017-09-06 04:41:11,074 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script settings are added to the job
2017-09-06 04:41:11,170 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce.reduce.markreset
2017-09-06 04:41:11,170 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not set, set to defa
2017-09-06 04:41:11,170 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.output.compress is deprecated. Instead, use mapreduce.output.fileoutputformat.compress
2017-09-06 04:41:11,838 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - creating jar file Job3874877649632062l5.jar
2017-09-06 04:41:15,855 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - jar file Job3874877649632062l5.jar created
2017-09-06 04:41:15,855 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.jar is deprecated. Instead, use mapreduce.job.jar
2017-09-06 04:41:15,876 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting up cloudera@quickstart:~
```

```
File  Edit  View  Search  Terminal  Help
2017-09-05 23:29:02,615 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete
2017-09-05 23:29:16,031 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 50% complete
2017-09-05 23:29:17,800 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2017-09-05 23:29:17,850 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2017-09-05 23:29:17,851 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.10.0 0.12.0-cdh5.10.0        cloudera        2017-09-05 23:28:55     2017-09-05 23:29:17     UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature C
job_1504678625866_0001  1       0       4       4       4       4       n/a     n/a     n/a     n/a     A,B     MAP_ONLY        /user/acadgild/project/batch1/formattedweb,

Input(s):
Successfully read 20 records (7353 bytes) from: "/user/acadgild/project/batch1/web"

Output(s):
Successfully stored 20 records (1236 bytes) in: "/user/acadgild/project/batch1/formattedweb"

Counters:
Total records written : 20
Total bytes written : 1236
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1504678625866_0001


2017-09-05 23:29:17,930 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
Time taken: 0.393 seconds
OK
Time taken: 0.446 seconds
Loading data to table project.formatted_input partition (batchid=1)
Partition project.formatted_input{batchid=1} stats: [numFiles=1, numRows=0, totalSize=1236, rawDataSize=0]
OK
Time taken: 0.908 seconds
Loading data to table project.formatted_input partition (batchid=1)
Partition project.formatted_input{batchid=1} stats: [numFiles=2, numRows=0, totalSize=2493, rawDataSize=0]
OK
Time taken: 0.742 seconds
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

In the above screenshot we can see the dataformatting.pig along with the formatted_hive_load.hql executed successfully.

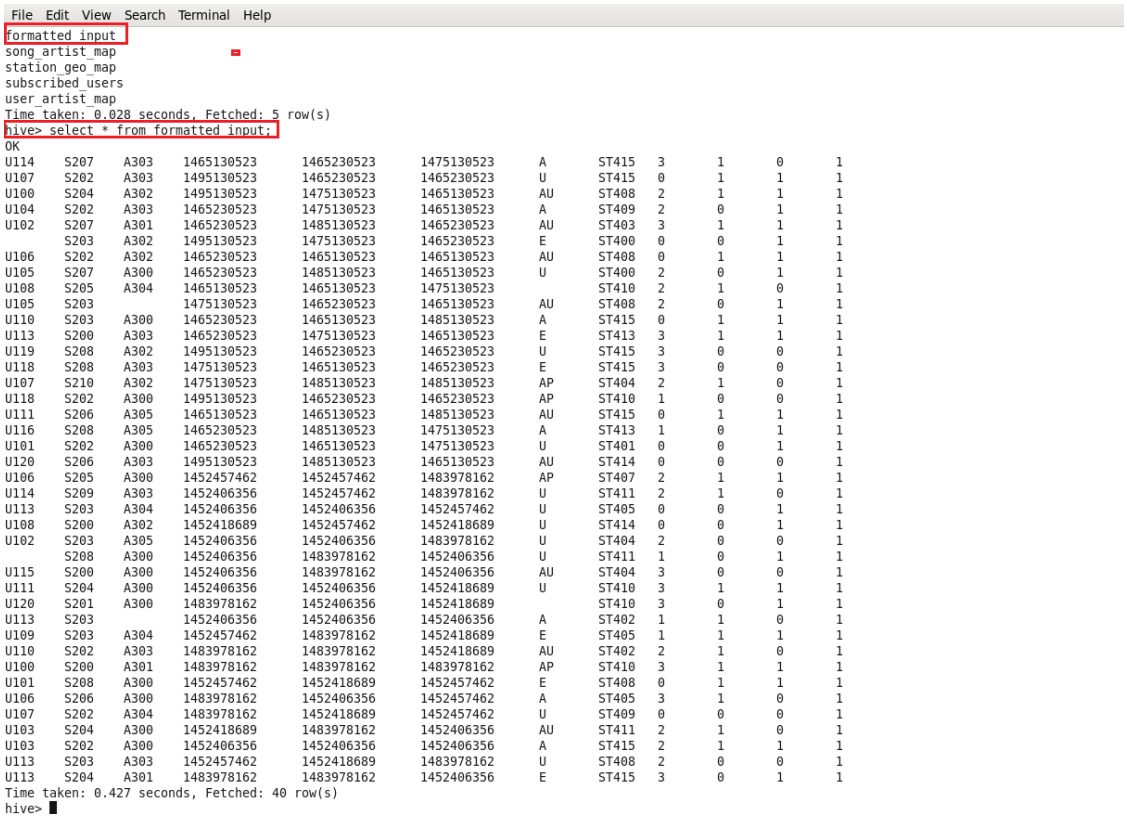
Music Data Analysis

The output of dataformatting.sh script in HDFS folders:



```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart scripts]$ hadoop fs -ls /user/acadgild/project/batch1/
Found 3 items
drwxr-xr-x   - cloudera supergroup          0 2017-09-05 12:14 /user/acadgild/project/batch1/formattedweb
drwxr-xr-x   - cloudera supergroup          0 2017-09-05 12:13 /user/acadgild/project/batch1/mob
drwxr-xr-x   - cloudera supergroup          0 2017-09-05 12:13 /user/acadgild/project/batch1/web
[cloudera@quickstart scripts]$ hadoop fs -ls /user/acadgild/project/batch1/formattedweb
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2017-09-05 12:14 /user/acadgild/project/batch1/formattedweb/_SUCCESS
-rw-r--r--   1 cloudera supergroup       2597 2017-09-05 12:14 /user/acadgild/project/batch1/formattedweb/part-m-00000
[cloudera@quickstart scripts]$ hadoop fs -ls /user/acadgild/project/batch1/web
Found 3 items
-rw-r--r--   1 cloudera supergroup       6716 2017-09-05 12:13 /user/acadgild/project/batch1/web/file.xml
-rw-r--r--   1 cloudera supergroup       1067 2017-09-05 12:13 /user/acadgild/project/batch1/web/web_data.xml.txt~
-rw-r--r--   1 cloudera supergroup       6962 2017-09-05 12:13 /user/acadgild/project/batch1/web/web_data.xml~
[cloudera@quickstart scripts]$ hadoop fs -ls /user/acadgild/project/batch1/mob
Found 2 items
-rw-r--r--   1 cloudera supergroup       1267 2017-09-05 12:13 /user/acadgild/project/batch1/mob/file.txt
-rw-r--r--   1 cloudera supergroup       1267 2017-09-05 12:13 /user/acadgild/project/batch1/mob/file.txt~
[cloudera@quickstart scripts]$ hadoop fs -cat /user/acadgild/project/batch1/formattedweb/part-m-00000
U106,S205,A300,1452457462,1452457462,1483978162,AP,ST407,2,1,1
U114,S209,A303,1452406356,1452457462,1483978162,U,ST411,2,1,0
U113,S203,A304,1452406356,1452406356,1452457462,U,ST405,0,0,1
U108,S200,A302,1452418689,1452457462,1452418689,U,ST414,0,0,1
U102,S203,A305,1452406356,1452406356,1483978162,U,ST404,2,0,0
,S208,A300,1452406356,1483978162,1452406356,U,ST411,1,0,1
U115,S200,A300,1452406356,1483978162,1452406356,AU,ST404,3,0,0
U111,S204,A300,1452406356,1452406356,1452418689,U,ST410,3,1,1
U120,S201,A300,1483978162,1452406356,1452418689,,ST410,3,0,1
U113,S203,,1452406356,1452406356,1452406356,A,ST402,1,1,0
U109,S203,A304,1452457462,1483978162,1452418689,E,ST405,1,1,1
U110,S202,A303,1483978162,1483978162,1452418689,AU,ST402,2,1,0
U100,S200,A301,1483978162,1483978162,1483978162,AP,ST410,3,1,1
U101,S208,A300,1452457462,1452418689,1452457462,E,ST408,0,1,1
U106,S206,A300,1483978162,1452406356,1452457462,A,ST405,3,1,0
U107,S202,A304,1483978162,1452418689,1452457462,U,ST409,0,0,0
U103,S204,A300,1452418689,1483978162,1452406356,AU,ST411,2,1,0
U103,S202,A300,1452406356,1452406356,1452406356,A,ST415,2,1,1
U113,S203,A303,1452457462,1452418689,1483978162,U,ST408,2,0,0
U113,S204,A301,1483978162,1483978162,1452406356,E,ST415,3,0,1
U106,S205,A300,1452457462,1452457462,1483978162,AP,ST407,2,1,1
U114,S209,A303,1452406356,1452457462,1483978162,U,ST411,2,1,0
U113,S203,A304,1452406356,1452406356,1452457462,U,ST405,0,0,1
U108,S200,A302,1452418689,1452457462,1452418689,U,ST414,0,0,1
U102,S203,A305,1452406356,1452406356,1483978162,U,ST404,2,0,0
,S208,A300,1452406356,1483978162,1452406356,U,ST411,1,0,1
U115,S200,A300,1452406356,1483978162,1452406356,AU,ST404,3,0,0
U111,S204,A300,1452406356,1452406356,1452418689,U,ST410,3,1,1
U120,S201,A300,1483978162,1452406356,1452418689,,ST410,3,0,1
U113,S203,,1452406356,1452406356,1452406356,A,ST402,1,1,0
U106 cloudera@quickstart ... 2 1483978162 1452418689 E ST405 1 1 1
```

DataFormatting.sh output in hive terminal:



```
File  Edit  View  Search  Terminal  Help
formatted_input
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.028 seconds, Fetched: 5 row(s)
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    A305    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    1452457462      1452457462      1483978162      AP      ST407   2       1       1       1
U114    S209    A303    1452406356      1452457462      1483978162      U       ST411   2       1       0       1
U113    S203    A304    1452406356      1452406356      1452457462      U       ST405   0       0       1       1
U108    S200    A302    1452418689      1452457462      1452418689      U       ST414   0       0       1       1
U102    S203    A305    1452406356      1452406356      1483978162      U       ST404   2       0       0       1
        S208    A300    1452406356      1483978162      1452406356      U       ST411   1       0       1       1
U115    S200    A300    1452406356      1483978162      1452406356      AU      ST404   3       0       0       1
U111    S204    A300    1452406356      1452406356      1452418689      U       ST410   3       1       1       1
U120    S201    A300    1483978162      1452406356      1452418689              ST410   3       0       1       1
U113    S203            1452406356      1452406356      1452406356      A       ST402   1       1       0       1
U109    S203    A304    1452457462      1483978162      1452418689      E       ST405   1       1       1       1
U110    S202    A303    1483978162      1483978162      1452418689      AU      ST402   2       1       0       1
U100    S200    A301    1483978162      1483978162      1483978162      AP      ST410   3       1       1       1
U101    S208    A300    1452457462      1452418689      1452457462      E       ST408   0       1       1       1
U106    S206    A300    1483978162      1452406356      1452457462      A       ST405   3       1       0       1
U107    S202    A304    1483978162      1452418689      1452457462      U       ST409   0       0       0       1
U103    S204    A300    1452418689      1483978162      1452406356      AU      ST411   2       1       0       1
U103    S202    A300    1452406356      1452406356      1452406356      A       ST415   2       1       1       1
U113    S203    A303    1452457462      1452418689      1483978162      U       ST408   2       0       0       1
U113    S204    A301    1483978162      1483978162      1452406356      E       ST415   3       0       1       1
Time taken: 0.427 seconds, Fetched: 40 row(s)
hive>
```

- In the above screenshot we can see the formatted input data with some null values in user_id, aritist_id and geo_cd columns which we will fill the enrichment script based on rules of enrichment for artist_id and geo_cd only. We will get neglect user_id because they didn't mentioned anything about user_id for enrichment purpose.
- Data Formatting phase is executed successfully by loading both mobile and web data and partitioned based on batchid.

e) Data Enrichment:

In this phase we will enrich the data coming from web and mobile applications using the lookup table stored in Hbase and divide the records based on the enrichment rules into 'pass' and 'fail' records.

Rules for data enrichment

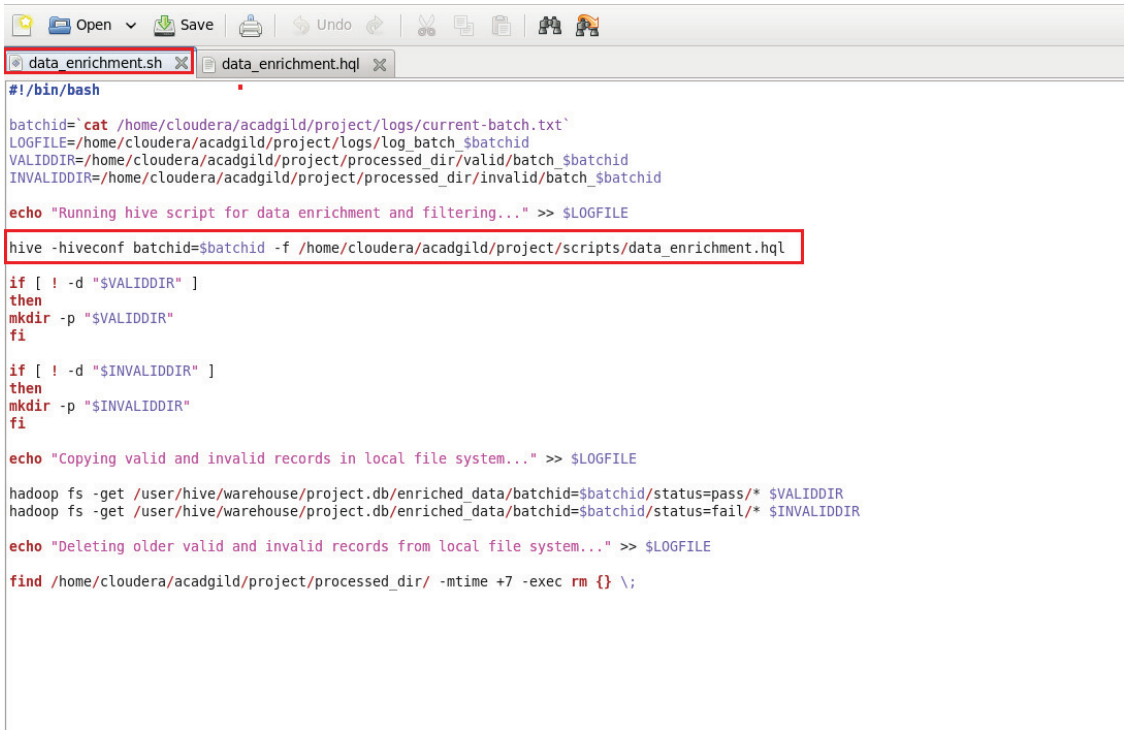
1. If any of like or dislike is NULL or absent, consider it as 0.

2. If fields like Geo_cd and Artist_id are NULL or absent, consult the lookup tables for fields

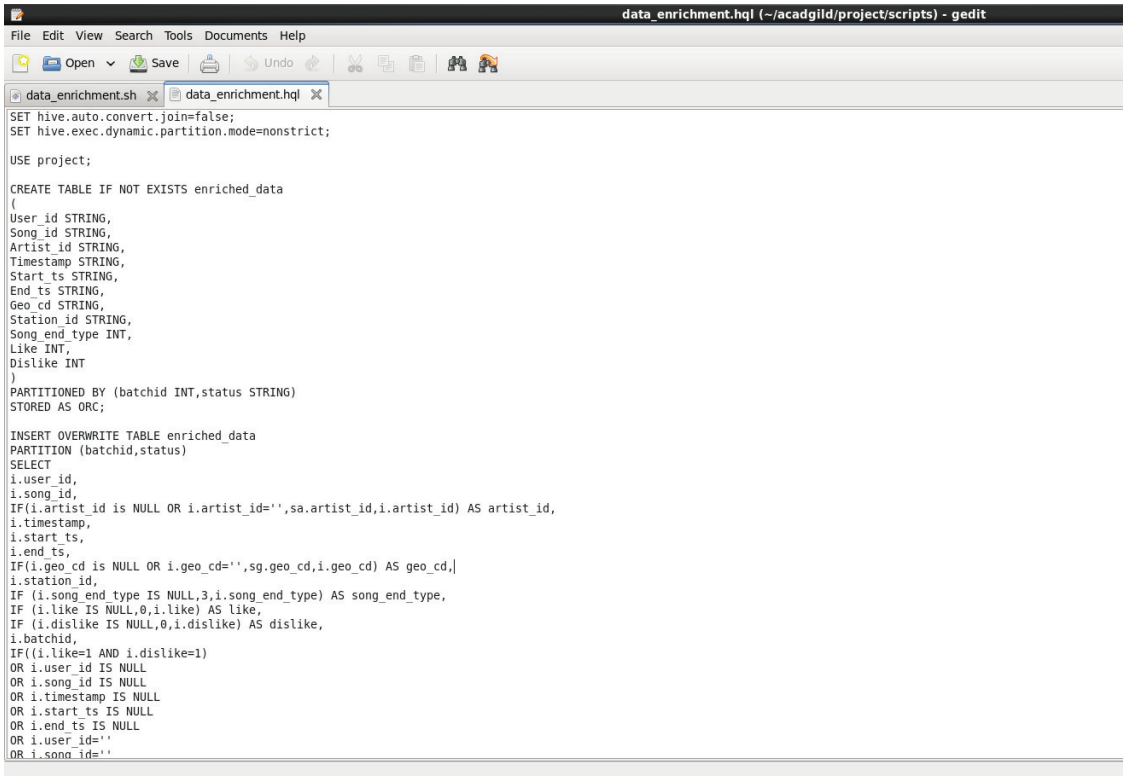
Station_id and Song_id respectively to get the values of Geo_cd and Artist_id.

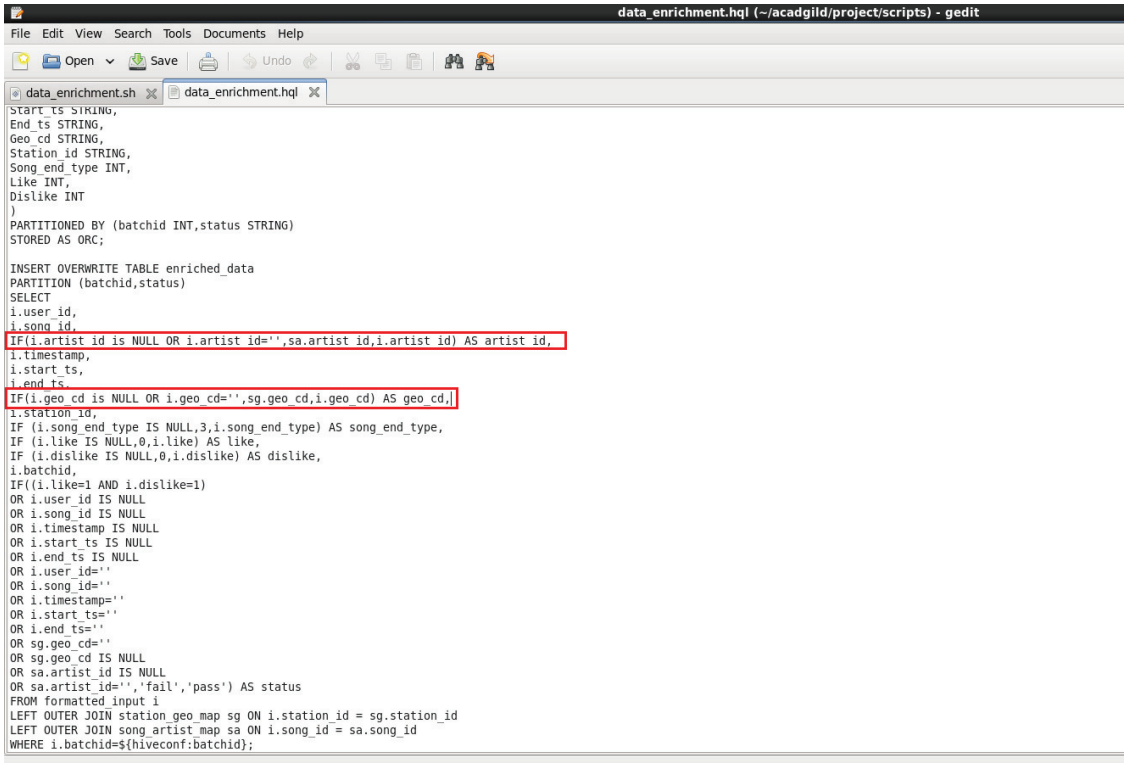
3. If corresponding lookup entry is not found, consider that record to be invalid

So based on the enrichment rules we will fill the null geo_cd and artist_id values with the help of corresponding lookup values in song-artist-map and station-geo-map tables in Hive-Hbase tables.



Music Data Analysis





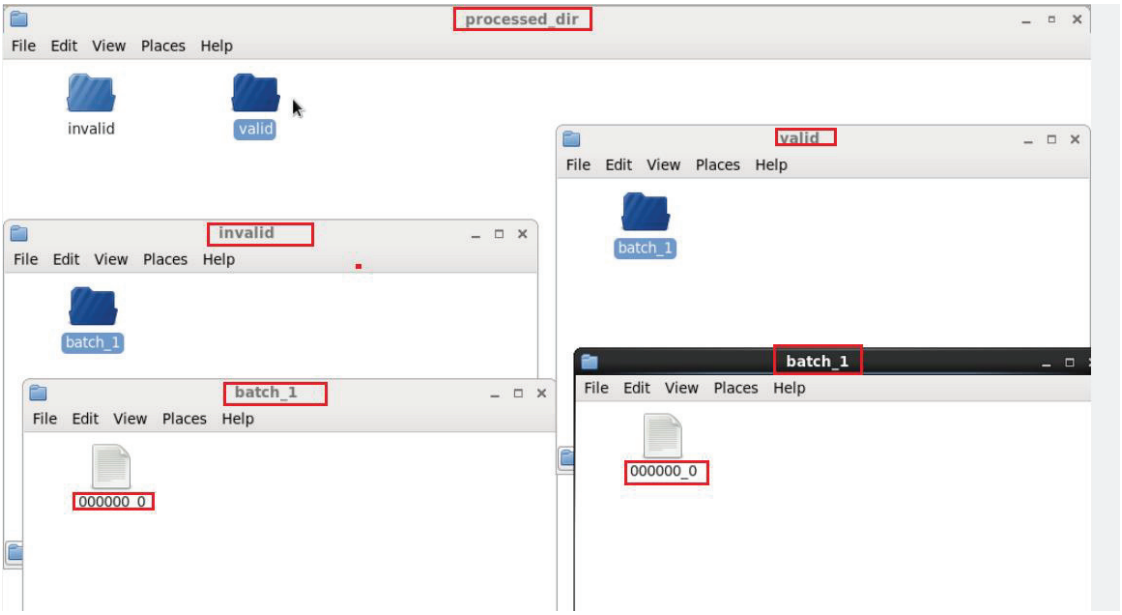
Music Data Analysis

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart project]$ cd scripts/
[cloudera@quickstart scripts]$ ./data_enrichment.sh

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
OK
Time taken: 0.391 seconds
OK
Time taken: 0.178 seconds
Query ID = cloudera_20170905233838_f151d27a-601b-405e-9313-6d9df9bd18f2
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):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1504678625866_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1504678625866_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1504678625866_0002
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2017-09-05 23:38:53,407 Stage-1 map = 0%,  reduce = 0%
2017-09-05 23:39:03,399 Stage-1 map = 33%,  reduce = 0%, Cumulative CPU 1.06 sec
2017-09-05 23:39:04,436 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 2.12 sec
2017-09-05 23:39:05,492 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.05 sec
2017-09-05 23:39:10,763 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.5 sec
MapReduce Total cumulative CPU time: 5 seconds 500 msec
Ended Job = job_1504678625866_0002
Launching Job 2 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):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1504678625866_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1504678625866_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1504678625866_0003
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 1
2017-09-05 23:39:25,758 Stage-2 map = 0%,  reduce = 0%
2017-09-05 23:39:33,225 Stage-2 map = 50%,  reduce = 0%, Cumulative CPU 0.95 sec
2017-09-05 23:39:34,284 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.85 sec
2017-09-05 23:39:40,558 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.58 sec
MapReduce Total cumulative CPU time: 4 seconds 580 msec
Ended Job = job_1504678625866_0003
Loading data to table project.enriched_data partition (batchid=null, status=null)
         Time taken for load dynamic partitions : 222
        Loading partition {batchid=1, status=fail}
        Loading partition {batchid=1, status=pass}
         Time taken for adding to write entity : 2
```

```
File  Edit  View  Search  Terminal  Help
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1504678625866_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1504678625866_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1504678625866_0002
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2017-09-05 23:38:53,407 Stage-1 map = 0%,  reduce = 0%
2017-09-05 23:39:03,399 Stage-1 map = 33%,  reduce = 0%, Cumulative CPU 1.06 sec
2017-09-05 23:39:04,436 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 2.12 sec
2017-09-05 23:39:05,492 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.05 sec
2017-09-05 23:39:10,763 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.5 sec
MapReduce Total cumulative CPU time: 5 seconds 500 msec
Ended Job = job_1504678625866_0002
Launching Job 2 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):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1504678625866_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1504678625866_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1504678625866_0003
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 1
2017-09-05 23:39:25,758 Stage-2 map = 0%,  reduce = 0%
2017-09-05 23:39:33,225 Stage-2 map = 50%,  reduce = 0%, Cumulative CPU 0.95 sec
2017-09-05 23:39:34,284 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.85 sec
2017-09-05 23:39:40,558 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.58 sec
MapReduce Total cumulative CPU time: 4 seconds 580 msec
Ended Job = job_1504678625866_0003
Loading data to table project.enriched_data partition (batchid=null, status=null)
         Time taken for load dynamic partitions : 222
        Loading partition {batchid=1, status=fail}
        Loading partition {batchid=1, status=pass}
         Time taken for adding to write entity : 2
Partition project.enriched_data{batchid=1, status=fail} stats: [numFiles=1, numRows=18, totalSize=1481, rawDataSize=12435]
Partition project.enriched_data{batchid=1, status=pass} stats: [numFiles=1, numRows=22, totalSize=1487, rawDataSize=16126]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3  Reduce: 1   Cumulative CPU: 5.5 sec   HDFS Read: 45891 HDFS Write: 3097 SUCCESS
Stage-Stage-2: Map: 2  Reduce: 1   Cumulative CPU: 4.58 sec   HDFS Read: 21135 HDFS Write: 3135 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 80 msec
OK
Time taken: 63.499 seconds
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

At the end script will automatically divide the records based on status pass & fail and dump the result into processed_dir folder with valid and invalid folders.

Music Data Analysis



- Now we can check whether the data properly loaded in the hive terminal or not .

```
hive> show tables;
OK
enriched_data
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.034 seconds, Fetched: 6 row(s)
```

- In the below screenshot we have data for data enrichment table where we filled the null values of artist_id and geo_cd of formatted input with the help of lookup tables

```
File  Edit  View  Search  Terminal  Help
formatted_input
song_artist_map
station_geo_map
subscribed_users
unsubscribed_users
user_artist_map
Time taken: 0.214 seconds, Fetched: 7 row(s)
hive> select * from enriched_data;
OK
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       1       fail
U100    S200    A301    1494342562      1494342562      1494342562      AP      ST410   3       1       1       1       fail
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       1       fail
U103    S202    A300    1465535556      1465535556      1465535556      A       ST415   2       1       1       1       fail
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       1       fail
U109    S203    A304    1462908262      1494342562      1468139889      E       ST405   1       1       1       1       fail
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       1       fail
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       1       fail
U111    S204    A300    1465535556      1465535556      1468139889      U       ST410   3       1       1       1       fail
U113    S204    A301    1494342562      1494342562      1465535556      E       ST415   3       0       1       1       fail
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       1       fail
U106    S205    A300    1462908262      1462908262      1494342562      AP      ST407   2       1       1       1       fail
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       1       fail
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       1       fail
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       1       fail
        S208    A300    1465535556      1494342562      1465535556      U       ST411   1       0       1       1       fail
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       1       fail
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       1       fail
U101    S208    A300    1462908262      1468139889      1462908262      E       ST408   0       1       1       1       fail
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       1       fail
U115    S200    A300    1465535556      1494342562      1465535556      AU      ST404   3       0       0       1       pass
U108    S200    A302    1468139889      1462908262      1468139889      U       ST414   0       0       1       1       pass
U120    S201    A300    1494342562      1465535556      1468139889      A       ST410   3       0       1       1       pass
U107    S202    A304    1494342562      1468139889      1462908262      U       ST409   0       0       0       1       pass
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       1       pass
U110    S202    A303    1494342562      1494342562      1468139889      AU      ST402   2       1       0       1       pass
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       1       pass
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       1       pass
U102    S203    A305    1465535556      1465535556      1494342562      U       ST404   2       0       0       1       pass
U113    S203    A304    1465535556      1465535556      1462908262      U       ST405   0       0       1       1       pass
U113    S203    A303    1462908262      1468139889      1494342562      U       ST408   2       0       0       1       pass
U105    S203    A303    1475130523      1465230523      1465130523      AU      ST408   2       0       1       1       pass
U113    S203    A303    1465535556      1465535556      1465535556      A       ST402   1       1       0       1       pass
U103    S204    A300    1468139889      1494342562      1465535556      AU      ST411   2       1       0       1       pass
U108    S205    A304    1465130523      1465130523      1475130523      A       ST410   2       1       0       1       pass
U106    S206    A300    1494342562      1465535556      1462908262      A       ST405   3       1       0       1       pass
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       1       pass
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       1       pass
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       1       pass
U114    S209    A303    1465535556      1462908262      1494342562      U       ST411   2       1       0       1       pass
Time taken: 0.419 seconds, Fetched: 40 row(s)
```

- Enrichment phase is executed successfully by applying all the rules of enrichment.

f) Data Analysis using Spark:

- In this stage we will do analysis on enriched data using Spark SQL and run the program using **Spark-Submit** command.
- Before running the spark-submit command we have to **zip –d command** to remove the bad manifests in created spark project jar file to avoid the invalid Signature exception.
- I used two spark-submits for analysis. 1) Spark_analysis for creating tables for each query/problem statement.
  2) Spark_analysis_2 for displaying results for each query in terminal.

start-daemon.sh | data_export.sh | log_batch_1 | DataAnalysis.sh | wrapper.sh | log_batch_2

```
#!/bin/bash

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

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

echo "Running script for data analysis using spark..." >> $LOGFILE
chmod 775 /home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar

zip -d /home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar META-INF/*.DSA META-
INF/*.RSA META-INF/*.SF

spark-submit    \
--class Spark_analysis \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/* \
/home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar $batchid

spark-submit    \
--class Spark_analysis_2 \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/* \
/home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar $batchid

echo "Exporting data to MYSQL using sqoop export..." >> $LOGFILE
sh /home/acadgild/project/scripts/data_export.sh

echo "Incrementing batchid..." >> $LOGFILE
batchid=`expr $batchid + 1`
echo -n $batchid > /home/acadgild/project/logs/current-batch.txt
```

Music Data Analysis

Spark Source Code:

I have created one Scala file for creating tables as per query wise.

Spark_analysis_2.scala | Spark_analysis.scala

```scala
import org.apache.hadoop.hive.serde2.`lazy`.LazySimpleSerDe
import org.apache.spark.sql.SparkSession

object Spark_analysis {

  def main(args: Array[String]): Unit = {
    val sparkSession = SparkSession.builder()
      .master("local[2]")
      .appName("Data Analysis Main_1")
      .config("spark.sql.warehouse.dir","/user/hive/warehouse")
      .config("hive.metastore.uris","thrift://127.0.0.1:9083")
      .enableHiveSupport()
      .getOrCreate()

    val batchId = args(0)

    //<<<<<<<<<<---------- PROBLEM 1 - Creation of table and Insertion of data ------------>>>>>>>>>>>>
    //Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.

    val set_properties = sparkSession.sqlContext.sql("set hive.auto.convert.join=false")

    val use_project_database = sparkSession.sqlContext.sql("USE project")

    val create_hive_table_top_10_stations = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_stations"+
      "("+
      " station_id STRING,"+
      " total_distinct_songs_played INT,"+
      " distinct_user_count INT"+
      ")"+
      " PARTITIONED BY (batchid INT)"+
      " ROW FORMAT DELIMITED"+
      " FIELDS TERMINATED BY ','"+
      " STORED AS TEXTFILE")


    val insert_into_top_10_stations = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_stations"+
      s" PARTITION (batchid=$batchId)"+
      " SELECT"+
      " station_id,"+
      " COUNT(DISTINCT song_id) AS total_distinct_songs_played,"+
      " COUNT(DISTINCT user_id) AS distinct_user_count"+
```

Spark_analysis_2.scala | Spark_analysis.scala

```scala
    //<<<<<<<<<<---------- PROBLEM 1 - Creation of table and Insertion of data ------------>>>>>>>>>>>>
    //Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.

    val set_properties = sparkSession.sqlContext.sql("set hive.auto.convert.join=false")

    val use_project_database = sparkSession.sqlContext.sql("USE project")

    val create_hive_table_top_10_stations = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_stations"+
      "("+
      " station_id STRING,"+
      " total_distinct_songs_played INT,"+
      " distinct_user_count INT"+
      ")"+
      " PARTITIONED BY (batchid INT)"+
      " ROW FORMAT DELIMITED"+
      " FIELDS TERMINATED BY ','"+
      " STORED AS TEXTFILE")


    val insert_into_top_10_stations = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_stations"+
      s" PARTITION (batchid=$batchId)"+
      " SELECT"+
      " station_id,"+
      " COUNT(DISTINCT song_id) AS total_distinct_songs_played,"+
      " COUNT(DISTINCT user_id) AS distinct_user_count"+
      " FROM project.enriched_data"+
      " WHERE status='pass'"+
      s" AND (batchid=$batchId)"+
      " AND like=1"+
      " GROUP BY station_id"+
      " ORDER BY total_distinct_songs_played DESC"+
      " LIMIT 10")
```

Music Data Analysis

Spark_analysis_2.scala | Spark_analysis.scala

```
//<<<<<<<<<<---------- PROBLEM 2 - Creation of table and Insertion of data ---------->>>>>>>>>>
/*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.*/

val create_hive_table_song_duration = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.song_duration"+
  "("+
  " user_id STRING,"+
  " user_type STRING,"+
  " song_id STRING,"+
  " artist_id STRING,"+
  " total_duration_in_minutes DOUBLE"+
  ")"+
  " PARTITIONED BY (batchid INT)"+
  " ROW FORMAT DELIMITED"+
  " FIELDS TERMINATED BY ','"+
  " STORED AS TEXTFILE")


val insert_into_song_duration = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.song_duration"+
  s" PARTITION (batchid=$batchId)"+
  " SELECT"+
  " e.user_id STRING,"+
  " IF(e.user_id!=s.user_id"+
  " OR (CAST(s.subscn_end_dt as BIGINT) < CAST(e.start_ts as BIGINT)),'unsubscribed','subscribed') AS user_type,"+
  " e.song_id STRING,"+
  " e.artist_id STRING,"+
  " (cast(e.end_ts as BIGINT)-cast(e.start_ts as BIGINT))/60 AS total_duration_in_minutes"+
  " FROM project.enriched_data e"+
  " LEFT OUTER JOIN project.subscribed_users s"+
  " ON e.user_id=s.user_id"+
  " WHERE e.status='pass'"+
  s" AND (batchid=$batchId)")
```

Spark_analysis_2.scala | Spark_analysis.scala

```
//<<<<<<<<<<---------- PROBLEM 3 - Creation of table and Insertion of data ---------->>>>>>>>>>
//Determine top 10 connected artists.
//Connected artists are those whose songs are most listened by the unique users who follow them.

val create_hive_table_top_10_connected_artists = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.connected_artists"+
  "("+
  " artist_id STRING,"+
  " total_distinct_songs INT,"+
  " unique_followers INT"+
  ")"+
  " PARTITIONED BY (batchid INT)"+
  " ROW FORMAT DELIMITED"+
  " FIELDS TERMINATED BY ','"+
  " STORED AS TEXTFILE")


val insert_into_top_10_connected_artists = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.connected_artists"+
  s" PARTITION (batchid=$batchId)"+
  " SELECT"+
  " artist_id,"+
  " COUNT(DISTINCT song_id) AS total_distinct_songs,"+
  " COUNT(DISTINCT user_id) AS unique_followers"+
  " FROM project.enriched_data"+
  " WHERE status='pass'"+
  s" AND (batchid=$batchId)"+
  " GROUP BY artist_id"+
  " ORDER BY unique_followers desc,total_distinct_songs desc"+
  " LIMIT 10")
```

Spark_analysis_2.scala | Spark_analysis.scala

```
//<<<<<<<<<<---------- PROBLEM 4 - Creation of table and Insertion of data ---------->>>>>>>>>>
//Determine top 10 songs who have generated the maximum revenue.
//NOTE: Royalty applies to a song only if it was liked or was completed successfully or both.

val create_hive_table_top_10_songs_maxrevenue = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_songs_maxrevenue"+
  "("+
  " song_id STRING,"+
  " artist_id STRING,"+
  " total_duration_in_minutes DOUBLE"+
  " )"+
  " PARTITIONED BY (batchid INT)"+
  " ROW FORMAT DELIMITED"+
  " FIELDS TERMINATED BY ','"+
  " STORED AS TEXTFILE")


val insert_into_top_10_songs_maxrevenue = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_songs_maxrevenue"+
  s" PARTITION (batchid=$batchId)"+
  " SELECT"+
  " song_id,"+
  " artist_id,"+
  " (cast(end_ts as BIGINT)-cast(start_ts as BIGINT))/60 AS total_duration_in_minutes"+
  " FROM project.enriched_data"+
  " WHERE status='pass'" +
  s" AND (batchid=$batchId)"+
  " AND (like=1 OR song_end_type=0 OR (like=1 and song_end_type=0))"+
  " ORDER BY total_duration_in_minutes desc"+
  " LIMIT 10")
```

Music Data Analysis

Spark_analysis_2.scala | Spark_analysis.scala

```
//<<<<<<<<<<---------- PROBLEM 5 - Creation of table and Insertion of data ------------>>>>>>>>>>>>
//Determine top 10 unsubscribed users who listened to the songs for the longest duration.

val create_hive_table_top_10_unsubscribed_users = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_unsubscribed_users"+
  "("+
  " user_id STRING,"+
  " song_id STRING,"+
  " artist_id STRING,"+
  " total_duration_in_minutes DOUBLE"+
  ")"+
  " PARTITIONED BY (batchid INT)"+
  " ROW FORMAT DELIMITED"+
  " FIELDS TERMINATED BY ','"+
  " STORED AS TEXTFILE")


val insert_into_unsubscribed_users = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_unsubscribed_users"+
  s" PARTITION (batchid=$batchId)"+
  " SELECT"+
  " user_id,"+
  " song_id,"+
  " artist_id,"+
  " total_duration_in_minutes"+
  " FROM project.song_duration"+
  " WHERE user_type='unsubscribed'"+
  " AND total_duration_in_minutes>=0"+
  s" AND (batchid=$batchId)"+
  " ORDER BY total_duration_in_minutes desc"+
  " LIMIT 10")


}
}
```

This second Scala file is for displaying results of each query in the terminal.

Spark_analysis_2.scala | Spark_analysis.scala

```
import ...
object Spark_analysis_2 {
  def main(args: Array[String]): Unit = {
    val sparkSession = SparkSession.builder.master("local").appName("Spark Session example")
      .config("spark.sql.warehouse.dir", "/user/hive/warehouse")
      .config("hive.metastore.uris", "thrift://localhost:9083")
      .enableHiveSupport().getOrCreate()
    val batchId = args(0)

    sparkSession.sqlContext.sql("USE project")
    sparkSession.sqlContext.sql("SELECT station_id from top_10_stations").show()
    sparkSession.sqlContext.sql("SELECT user_type,total_duration_in_minutes from song_duration").show()
    sparkSession.sqlContext.sql("SELECT artist_id from connected_artists").show()
    sparkSession.sqlContext.sql("SELECT song_id from top_10_songs_maxrevenue").show()
    sparkSession.sqlContext.sql("SELECT user_id from top_10_unsubscribed_users").show()


  }
}
```

- We will create jar for the above source code using File -> project structure -> Artifacts -> create jar using Build Artifacts command.
- Then we can observe **out** folder getting created with artifacts inside it.
- Now we will export this jar to VM where we will run using spark submit command and get the analysis result.

When we run the Data Analysis First it will submit Spark_analysis file and then it will run the Spark-analysis_2 file which displays the result for each query.

Music Data Analysis

Spark analysis output:

```
File  Edit  View  Search  Terminal  Help
[acadgild@localhost scripts]$ ./DataAnalysis.sh
        zip warning: name not matched: META-INF/*.RSA
deleting: META-INF/DUMMY.SF
deleting: META-INF/DUMMY.DSA
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:/home/acadgild/Downloads/spark-2.0.0-bin-hadoop2.6/jars/slf4j-log4j12-1.7.16.jar!/org/slf4
/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
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 default log4j profile: org/apache/spark/log4j-defaults.properties
17/09/10 16:35:51 INFO SparkContext: Running Spark version 2.0.0
17/09/10 16:35:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
 where applicable
17/09/10 16:35:52 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.
.4 instead (on interface eth5)
17/09/10 16:35:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
17/09/10 16:35:52 INFO SecurityManager: Changing view acls to: acadgild
17/09/10 16:35:52 INFO SecurityManager: Changing modify acls to: acadgild
17/09/10 16:35:52 INFO SecurityManager: Changing view acls groups to:
17/09/10 16:35:52 INFO SecurityManager: Changing modify acls groups to:
17/09/10 16:35:52 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permiss
```

```
ry: /tmp/23a6594d-4ee0-422c-8777-11487adea113_resources
2017-09-10 16:56:59,310 INFO  [main] session.SessionState: Created HDFS director
y: /tmp/hive/acadgild/23a6594d-4ee0-422c-8777-11487adea113
2017-09-10 16:56:59,332 INFO  [main] session.SessionState: Created local directo
ry: /tmp/acadgild/23a6594d-4ee0-422c-8777-11487adea113
2017-09-10 16:56:59,365 INFO  [main] session.SessionState: Created HDFS director
y: /tmp/hive/acadgild/23a6594d-4ee0-422c-8777-11487adea113/_tmp_space.db
2017-09-10 16:56:59,374 INFO  [main] client.HiveClientImpl: Warehouse location f
or Hive client (version 1.2.1) is /user/hive/warehouse
2017-09-10 16:57:07,173 INFO  [main] execution.SparkSqlParser: Parsing command:
USE project
2017-09-10 16:57:07,332 INFO  [main] execution.SparkSqlParser: Parsing command:
CREATE TABLE IF NOT EXISTS project.top_10_stations( station_id STRING, total_dis
tinct_songs_played INT, distinct_user_count INT) PARTITIONED BY (batchid INT) RO
W FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE
2017-09-10 16:57:08,926 INFO  [main] execution.SparkSqlParser: Parsing command:
INSERT OVERWRITE TABLE project.top_10_stations PARTITION (batchid= 1) SELECT sta
tion_id, COUNT(DISTINCT song_id) AS total_distinct_songs_played, COUNT(DISTINCT
user_id) AS distinct_user_count FROM project.enriched_data WHERE status='pass' A
ND (batchid= 1) AND like=1 GROUP BY station_id ORDER BY total_distinct_songs_pla
yed DESC LIMIT 10
2017-09-10 16:57:11,668 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 16:57:11,689 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
2017-09-10 16:57:11,711 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
2017-09-10 16:57:11,713 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
```

Music Data Analysis

```
2017-09-10 16:58:57,921 INFO  [dag-scheduler-event-loop] scheduler.DAGScheduler:
 ResultStage 11 (sql at DataAnalysisMain_1.scala:131) finished in 1.228 s
2017-09-10 16:58:57,922 INFO  [main] scheduler.DAGScheduler: Job 4 finished: sql
 at DataAnalysisMain_1.scala:131, took 2.648835 s
2017-09-10 16:58:59,255 INFO  [main] common.FileUtils: deleting  hdfs://localhos
t:9000/user/hive/warehouse/project.db/top_10_songs_maxrevenue/batchid=1/part-000
00
2017-09-10 16:58:59,261 INFO  [main] fs.TrashPolicyDefault: Namenode trash confi
guration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
2017-09-10 16:58:59,337 INFO  [main] metadata.Hive: Replacing src:hdfs://localho
st:9000/user/hive/warehouse/project.db/top_10_songs_maxrevenue/.hive-staging_hiv
e_2017-09-10_16-58-54_002_5741185161756501718-1/-ext-10000/part-00000, dest: hdf
s://localhost:9000/user/hive/warehouse/project.db/top_10_songs_maxrevenue/batchi
d=1/part-00000, Status:true
2017-09-10 16:58:59,833 INFO  [main] execution.SparkSqlParser: Parsing command:
CREATE TABLE IF NOT EXISTS project.top_10_unsubscribed_users( user_id STRING, so
ng_id STRING, artist_id STRING, total_duration_in_minutes DOUBLE) PARTITIONED BY
 (batchid INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE
2017-09-10 16:59:00,057 INFO  [main] execution.SparkSqlParser: Parsing command:
INSERT OVERWRITE TABLE project.top_10_unsubscribed_users PARTITION (batchid= 1)
SELECT user_id, song_id, artist_id, total_duration_in_minutes FROM project.song_
duration WHERE user_type='unsubscribed' AND total_duration_in_minutes>=0 AND (ba
tchid= 1) ORDER BY total_duration_in_minutes desc LIMIT 10
2017-09-10 16:59:00,496 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 16:59:00,496 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
2017-09-10 16:59:00,496 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
```

```
2017-09-10 16:59:03,391 INFO  [dag-scheduler-event-loop] scheduler.DAGScheduler:
 Submitting 1 missing tasks from ResultStage 13 (MapPartitionsRDD[79] at sql at
DataAnalysisMain_1.scala:161)
2017-09-10 16:59:03,392 INFO  [dag-scheduler-event-loop] scheduler.TaskScheduler
Impl: Adding task set 13.0 with 1 tasks
2017-09-10 16:59:03,394 INFO  [dispatcher-event-loop-0] scheduler.TaskSetManager
: Starting task 0.0 in stage 13.0 (TID 809, localhost, partition 0, PROCESS_LOCA
L, 5330 bytes)
2017-09-10 16:59:03,394 INFO  [Executor task launch worker-3] executor.Executor:
 Running task 0.0 in stage 13.0 (TID 809)
2017-09-10 16:59:03,483 INFO  [Executor task launch worker-3] storage.ShuffleBlo
ckFetcherIterator: Getting 0 non-empty blocks out of 1 blocks
2017-09-10 16:59:03,483 INFO  [Executor task launch worker-3] storage.ShuffleBlo
ckFetcherIterator: Started 0 remote fetches in 1 ms
2017-09-10 16:59:03,633 INFO  [Executor task launch worker-3] output.FileOutputC
ommitter: Saved output of task 'attempt_201709101659_0013_m_000000_0' to hdfs://
localhost:9000/user/hive/warehouse/project.db/top_10_unsubscribed_users/.hive-st
aging_hive_2017-09-10_16-59-01_939_5714149478602249311-1/-ext-10000/_temporary/0
/task_201709101659_0013_m_000000
2017-09-10 16:59:03,633 INFO  [Executor task launch worker-3] mapred.SparkHadoop
MapRedUtil: attempt_201709101659_0013_m_000000_0: Committed
2017-09-10 16:59:03,654 INFO  [Executor task launch worker-3] executor.Executor:
 Finished task 0.0 in stage 13.0 (TID 809). 1676 bytes result sent to driver
2017-09-10 16:59:03,655 INFO  [task-result-getter-1] scheduler.TaskSetManager: F
inished task 0.0 in stage 13.0 (TID 809) in 262 ms on localhost (1/1)
2017-09-10 16:59:03,655 INFO  [task-result-getter-1] scheduler.TaskSchedulerImpl
: Removed TaskSet 13.0, whose tasks have all completed, from pool
2017-09-10 16:59:03,655 INFO  [dag-scheduler-event-loop] scheduler.DAGScheduler:
 ResultStage 13 (sql at DataAnalysisMain_1.scala:161) finished in 0.254 s
2017-09-10 16:59:03,657 INFO  [main] scheduler.DAGScheduler: Job 5 finished: sql
 at DataAnalysisMain_1.scala:161, took 0.747655 s
```

```
2017-09-10 18:07:03,355 INFO  [main] session.SessionState: Created HDFS director
y: /tmp/hive/acadgild/d62b1207-5c3f-4105-b4cd-7b887ebda0f7
2017-09-10 18:07:03,395 INFO  [main] session.SessionState: Created local directo
ry: /tmp/acadgild/d62b1207-5c3f-4105-b4cd-7b887ebda0f7
2017-09-10 18:07:03,421 INFO  [main] session.SessionState: Created HDFS director
y: /tmp/hive/acadgild/d62b1207-5c3f-4105-b4cd-7b887ebda0f7/_tmp_space.db
2017-09-10 18:07:03,436 INFO  [main] client.HiveClientImpl: Warehouse location f
or Hive client (version 1.2.1) is /user/hive/warehouse
2017-09-10 18:07:12,732 INFO  [main] execution.SparkSqlParser: Parsing command:
USE project
2017-09-10 18:07:13,108 INFO  [main] execution.SparkSqlParser: Parsing command:
select station_id from top_10_stations
2017-09-10 18:07:19,846 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:19,910 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
2017-09-10 18:07:19,920 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:19,921 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:20,328 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:20,329 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
2017-09-10 18:07:20,330 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:20,330 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:07:24,383 INFO  [main] memory.MemoryStore: Block broadcast_0 store
d as values in memory (estimated size 148.4 KB, free 413.8 MB)
2017-09-10 18:07:24,987 INFO  [main] memory.MemoryStore: Block broadcast_0_piece
```

Spark analysis_2 output:

Query-1: Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.

```
2017-09-10 18:21:10,509 INFO  [main] scheduler.DAGScheduler: Job 0 finished: sho
w at DataAnalysisReadFromHive.scala:22, took 15.957828 s
2017-09-10 18:21:10,994 INFO  [main] codegen.CodeGenerator: Code generated in 19
9.417505 ms
+----------+
|station_id|
+----------+
|     ST402|
|     ST411|
|     ST405|
|     ST410|
+----------+

2017-09-10 18:21:11,227 INFO  [main] execution.SparkSqlParser: Parsing command:
select user_type,sum(total_duration_in_minutes) as total_song_duration from song
_duration where total_duration_in_minutes>=0 group by user_type
2017-09-10 18:21:11,715 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
```

Query-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.

```
inished task 186.0 in stage 4.0 (TID 201) in 470 ms on localhost (199/199)
2017-09-10 18:21:37,544 INFO  [task-result-getter-1] scheduler.TaskSchedulerImpl
: Removed TaskSet 4.0, whose tasks have all completed, from pool
2017-09-10 18:21:37,546 INFO  [main] scheduler.DAGScheduler: Job 2 finished: sho
w at DataAnalysisReadFromHive.scala:31, took 14.650864 s
2017-09-10 18:21:37,705 INFO  [main] codegen.CodeGenerator: Code generated in 86
.765409 ms
+----------+-------------------+
| user_type|total_song_duration|
+----------+-------------------+
|subscribed| 1904665.6500000001|
+----------+-------------------+

2017-09-10 18:21:37,721 INFO  [main] execution.SparkSqlParser: Parsing command:
select artist_id from connected_artists
2017-09-10 18:21:37,902 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:21:37,909 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
```

Music Data Analysis

Query-3: Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them

```
2017-09-10 18:21:39,401 INFO  [dag-scheduler-event-loop] scheduler.DAGScheduler:
 ResultStage 5 (show at DataAnalysisReadFromHive.scala:37) finished in 0.224 s
2017-09-10 18:21:39,404 INFO  [main] scheduler.DAGScheduler: Job 3 finished: sho
w at DataAnalysisReadFromHive.scala:37, took 0.297580 s
2017-09-10 18:21:39,407 INFO  [task-result-getter-2] scheduler.TaskSchedulerImpl
: Removed TaskSet 5.0, whose tasks have all completed, from pool
+---------+
|artist_id|
+---------+
|     A300|
|     A303|
|     A304|
|     A305|
|     A302|
+---------+

2017-09-10 18:21:39,421 INFO  [main] execution.SparkSqlParser: Parsing command:
select song_id from top_10_songs_maxrevenue
2017-09-10 18:21:40,225 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:21:40,226 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
```

Query-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

```
inished task 0.0 in stage 6.0 (TID 203) in 155 ms on localhost (1/1)
2017-09-10 18:21:41,756 INFO  [task-result-getter-3] scheduler.TaskSchedulerImpl
: Removed TaskSet 6.0, whose tasks have all completed, from pool
+-------+
|song_id|
+-------+
|   S209|
|   S202|
|   S205|
|   S200|
|   S203|
|   S203|
|   S206|
|   S202|
|   S206|
|   S202|
+-------+

2017-09-10 18:21:41,774 INFO  [main] execution.SparkSqlParser: Parsing command:
select user_id from top_10_unsubscribed_users
2017-09-10 18:21:41,880 INFO  [main] parser.CatalystSqlParser: Parsing command:
int
2017-09-10 18:21:41,882 INFO  [main] parser.CatalystSqlParser: Parsing command:
string
```

Query-5: Determine top 10 unsubscribed users who listened to the songs for the longest duration.

```
2017-09-10 18:21:43,240 INFO  [task-result-getter-0] scheduler.TaskSetManager: F
inished task 0.0 in stage 7.0 (TID 204) in 140 ms on localhost (1/1)
2017-09-10 18:21:43,241 INFO  [task-result-getter-0] scheduler.TaskSchedulerImpl
: Removed TaskSet 7.0, whose tasks have all completed, from pool
+-------+
|user_id|
+-------+
+-------+

2017-09-10 18:21:43,356 INFO  [Thread-2] spark.SparkContext: Invoking stop() fro
m shutdown hook
2017-09-10 18:21:43,576 INFO  [Thread-2] server.ServerConnector: Stopped ServerC
onnector@ff8277e{HTTP/1.1}{0.0.0.0:4040}
```

For this query we got no result because there is no unsubscriber whose has got positive total _duration_minutes_in_minutes. All the unsubscribed users got negative total_duration means their end timestamp is less than the start time stamp of the song which is not correct.

Music Data Analysis

Now we can see the output of Spark analysis in hive:

```
hive> show tables;
OK
enriched_data
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.04 seconds, Fetched: 6 row(s)
hive> show tables;
OK
connected_artists
enriched_data
formatted_input
song_artist_map
song_duration
station_geo_map
subscribed_users
top_10_songs_maxrevenue
top_10_stations
top_10_unsubscribed_users
users_artists
Time taken: 0.047 seconds, Fetched: 11 row(s)
hive>
```


```
hive> select * from song_duration;
OK
U115    subscribed      S200    A300    -480116.76666666666     1
U108    subscribed      S200    A302    87193.78333333334       1
U120    subscribed      S201    A300    43405.55        1
U107    unsubscribed    S202    A304    -87193.78333333334      1
U101    subscribed      S202    A300    166666.66666666666      1
U110    unsubscribed    S202    A303    -436711.2166666667      1
U118    subscribed      S202    A300    0.0     1
U104    subscribed      S202    A303    -166666.66666666666     1
U102    subscribed      S203    A305    480116.76666666666      1
U113    subscribed      S203    A304    -43788.23333333333      1
U113    subscribed      S203    A303    436711.2166666667       1
U105    subscribed      S203    A303    -1666.6666666666667     1
U113    subscribed      S203    A303    0.0     1
U103    unsubscribed    S204    A300    -480116.76666666666     1
U108    subscribed      S205    A304    166666.66666666666      1
U106    subscribed      S206    A300    -43788.23333333333      1
U120    subscribed      S206    A303    -333333.3333333333      1
U105    unsubscribed    S207    A300    -333333.3333333333      1
U116    subscribed      S208    A303    -166666.66666666666     1
U114    subscribed      S209    A303    523905.0        1
Time taken: 0.048 seconds, Fetched: 20 row(s)
hive> select * from top_10_stations;
OK
ST402   2       2       1
ST411   2       2       1
ST405   1       1       1
ST410   1       1       1
Time taken: 0.061 seconds, Fetched: 4 row(s)
hive>
```

```
hive> select * from top_10_songs_maxrevenue;
OK
S209    A303    523905.0        1
S202    A300    166666.66666666666      1
S205    A304    166666.66666666666      1
S200    A302    87193.78333333334       1
S203    A303    0.0     1
S203    A304    -43788.23333333333      1
S206    A300    -43788.23333333333      1
S202    A304    -87193.78333333334      1
S206    A303    -333333.3333333333      1
S202    A303    -436711.2166666667      1
Time taken: 0.063 seconds, Fetched: 10 row(s)
hive> select * from connected_artist;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'connected_artist'
hive> select * from connected_artists;
OK
A300    6       7       1
A303    5       7       1
A304    3       3       1
A305    1       1       1
A302    1       1       1
Time taken: 0.062 seconds, Fetched: 5 row(s)
hive> select * from top_10_unsubscribed_users;
OK
Time taken: 0.057 seconds
hive>
```


```
hive> select user_type,SUM(total_duration_in_minutes) from song_duration where total_duration_in_minutes >=0 group by user_ty
minutes) from song_duration where total_duration_in_minutes >=0 group by user_type;
Query ID = acadgild_20170911202020_14573a7d-939c-410f-aff0-0de67d02cbc2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1505113626668_0001, Tracking URL = http://localhost:8088/proxy/application_1505113626668_0001/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1505113626668_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-09-11 20:20:59,578 Stage-1 map = 0%,  reduce = 0%
2017-09-11 20:21:07,222 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.95 sec
2017-09-11 20:21:15,787 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.85 sec
MapReduce Total cumulative CPU time: 4 seconds 850 msec
Ended Job = job_1505113626668_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.85 sec   HDFS Read: 1102 HDFS Write: 30 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 850 msec
OK
subscribed      1904665.6500000001
Time taken: 30.85 seconds, Fetched: 1 row(s)
hive>
```

Now we have seen all the spark queries creating the tables for each query. So Data Analysis using Spark is executed successfully.

g) Data Export to MYSQL:

In this stage data will be exported from hive warehouse directory to MYSQL database using data_export.sh command mentioned in the DataAnalysis.sh script.

Music Data Analysis

data_export.sh | dataformatting_hadoop.sh | data_enrichment.sh | data_enrichment.hql | data_analysis_spark.sh

```
#!/bin/bash

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

echo "Creating mysql tables if not present..." >> $LOGFILE

mysql -u root </home/acadgild/project/scripts/create_schema.sql

echo "Running sqoop job for data export..." >> $LOGFILE

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--P \
--table 'top_10_stations' \
--export-dir '/user/hive/warehouse/project.db/top_10_stations/batchid=$batchid/part-00000' \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--P \
--table 'song_duration' \
--export-dir '/user/hive/warehouse/project.db/song_duration/batchid=$batchid/part-00000' \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--P \
--table 'connected_artists' \
--export-dir '/user/hive/warehouse/project.db/connected_artists/batchid=$batchid/part-00000' \
--input-fields-terminated-by ',' \
-m 1
```

```
sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--P \
--table 'top_10_songs_maxrevenue' \
--export-dir '/user/hive/warehouse/project.db/top_10_songs_maxrevenue/batchid=$batchid/part-00000' \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--P \
--table 'top_10_unsubscribed_users' \
--export-dir '/user/hive/warehouse/project.db/top_10_unsubscribed_users/batchid=$batchid/part-00000' \
--input-fields-terminated-by ',' \
-m 1
```

```
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2017-09-10 23:00:16,788 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2017-09-10 23:00:18,522 INFO  [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2017-09-10 23:00:18,523 INFO  [main] tool.CodeGenTool: Beginning code generation
2017-09-10 23:00:18,751 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `song_duration` AS t LIMIT
1
2017-09-10 23:00:18,770 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `song_duration` AS t LIMIT
1
2017-09-10 23:00:18,776 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
Note: /tmp/sqoop-acadgild/compile/d4aa31e4f5d01298a20c5991eb23795c/song_duration.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
2017-09-10 23:00:19,955 INFO  [main] orm.CompilationManager: Writing jar file: /tmp/sqoop-acadgild/compile/d4aa31e4f5d01298a2
0c5991eb23795c/song_duration.jar
2017-09-10 23:00:19,963 INFO  [main] mapreduce.ExportJobBase: Beginning export of song_duration
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.
2017-09-10 23:00:20,154 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
2017-09-10 23:00:20,159 INFO  [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2017-09-10 23:00:20,764 INFO  [main] Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Inst
ead, use mapreduce.reduce.speculative
2017-09-10 23:00:20,767 INFO  [main] Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead
, use mapreduce.map.speculative
2017-09-10 23:00:20,768 INFO  [main] Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.ma
```

[acadgild@localhos... | acadgild@localhost... | sqoop EXPORT - Th... | data_export.sh (~/p... | acadgild@localhost:~

```
File  Edit  View  Search  Terminal  Help
2017-09-10 23:00:21,836 INFO  [main] Configuration.deprecation: mapred.job.name is deprecated. Instead, use mapreduce.job.nam
e
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.cache.files.timestamps is deprecated. Instead, use map
reduce.job.cache.files.timestamps
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapreduce.map.class is deprecated. Instead, use mapreduce.job
.map.class
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.
fileinputformat.inputdir
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead
, use mapreduce.map.speculative
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapreduce.inputformat.class is deprecated. Instead, use mapre
duce.job.inputformat.class
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapr
educe.job.outputformat.class
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.cache.files is deprecated. Instead, use mapreduce.job.
cache.files
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.
working.dir
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.mapoutput.value.class is deprecated. Instead, use mapr
educe.map.output.value.class
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.mapoutput.key.class is deprecated. Instead, use mapred
uce.map.output.key.class
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: mapred.job.classpath.files is deprecated. Instead, use mapred
uce.job.classpath.files
2017-09-10 23:00:21,837 INFO  [main] Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
2017-09-10 23:00:21,838 INFO  [main] Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job
.reduces
2017-09-10 23:00:21,838 INFO  [main] Configuration.deprecation: mapred.cache.files.filesizes is deprecated. Instead, use mapr
educe.job.cache.files.filesizes
2017-09-10 23:00:21,942 INFO  [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1505059429035_0009
2017-09-10 23:00:22,143 INFO  [main] impl.YarnClientImpl: Submitted application application_1505059429035_0009 to ResourceMan
ager at /0.0.0.0:8032
2017-09-10 23:00:22,184 INFO  [main] mapreduce.Job: The url to track the job: http://http://localhost:8088/proxy/application_
1505059429035_0009/
2017-09-10 23:00:22,184 INFO  [main] mapreduce.Job: Running job: job_1505059429035_0009
2017-09-10 23:00:28,333 INFO  [main] mapreduce.Job: Job job_1505059429035_0009 running in uber mode : false
2017-09-10 23:00:28,336 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2017-09-10 23:00:34,425 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2017-09-10 23:00:34,435 INFO  [main] mapreduce.Job: Job job_1505059429035_0009 completed successfully
[acadgild@localhos...  acadgild@localhost...  sqoop EXPORT - Th...  data_export.sh (~/p...  acadgild@localhost:~
```

Now we can see the data exported successfully into the MYSQL Database for all the 5 queries.

MYSQL:

```
File  Edit  View  Search  Terminal  Help
mysql> use project;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------+
| Tables_in_project         |
+---------------------------+
| connected_artists         |
| song_duration             |
| top_10_royality_songs     |
| top_10_stations           |
| top_10_unsubscribed_users |
+---------------------------+
5 rows in set (0.00 sec)

mysql> select * from connected_artists;
+-----------+----------------------+------------+
| artist_id | total_distinct_songs | user_count |
+-----------+----------------------+------------+
| A300      |                    6 |          7 |
| A303      |                    5 |          7 |
| A304      |                    3 |          3 |
| A305      |                    1 |          1 |
| A302      |                    1 |          1 |
+-----------+----------------------+------------+
5 rows in set (0.05 sec)
```

Music Data Analysis

```
File  Edit  View  Search  Terminal  Help
mysql> select * from top_10_royality_songs;
+---------+-----------+-------------------+
| song_id | artist_id | duration          |
+---------+-----------+-------------------+
| S209    | A303      |            523905 |
| S202    | A300      |  166666.666666667 |
| S205    | A304      |  166666.666666667 |
| S200    | A302      |  87193.7833333333 |
| S203    | A303      |                 0 |
| S203    | A304      | -43788.2333333333 |
| S206    | A300      | -43788.2333333333 |
| S202    | A304      | -87193.7833333333 |
| S206    | A303      | -333333.333333333 |
| S202    | A303      | -436711.216666667 |
+---------+-----------+-------------------+
10 rows in set (0.01 sec)

mysql> select * from top_10_unsubscribed_users;
Empty set (0.00 sec)

mysql> select * from top_10_stations;
+------------+-----------------------------+---------------------+
| station_id | total_distinct_songs_played | distinct_user_count |
+------------+-----------------------------+---------------------+
| ST402      |                           2 |                   2 |
| ST411      |                           2 |                   2 |
| ST405      |                           1 |                   1 |
| ST410      |                           1 |                   1 |
+------------+-----------------------------+---------------------+
4 rows in set (0.02 sec)
```

```
File  Edit  View  Search  Terminal  Help
mysql> select * from top_10_stations;
+------------+-----------------------------+---------------------+
| station_id | total_distinct_songs_played | distinct_user_count |
+------------+-----------------------------+---------------------+
| ST402      |                           2 |                   2 |
| ST411      |                           2 |                   2 |
| ST405      |                           1 |                   1 |
| ST410      |                           1 |                   1 |
+------------+-----------------------------+---------------------+
4 rows in set (0.02 sec)

mysql> select * from song_duration;
+---------+--------------+---------+-----------+-------------------+
| user_id | user_type    | song_id | artist_id | total_duration    |
+---------+--------------+---------+-----------+-------------------+
| U115    | subscribed   | S200    | A300      | -480116.766666667 |
| U108    | subscribed   | S200    | A302      |  87193.7833333333 |
| U120    | subscribed   | S201    | A300      |          43405.55 |
| U107    | unsubscribed | S202    | A304      | -87193.7833333333 |
| U101    | subscribed   | S202    | A300      |  166666.666666667 |
| U110    | unsubscribed | S202    | A303      | -436711.216666667 |
| U118    | subscribed   | S202    | A300      |                 0 |
| U104    | subscribed   | S202    | A303      | -166666.666666667 |
| U102    | subscribed   | S203    | A305      |  480116.766666667 |
| U113    | subscribed   | S203    | A304      | -43788.2333333333 |
| U113    | subscribed   | S203    | A303      |  436711.216666667 |
| U105    | subscribed   | S203    | A303      | -1666.66666666667 |
| U113    | subscribed   | S203    | A303      |                 0 |
| U103    | unsubscribed | S204    | A300      | -480116.766666667 |
| U108    | subscribed   | S205    | A304      |  166666.666666667 |
| U106    | subscribed   | S206    | A300      | -43788.2333333333 |
| U120    | subscribed   | S206    | A303      | -333333.333333333 |
| U105    | unsubscribed | S207    | A300      | -333333.333333333 |
| U116    | subscribed   | S208    | A303      | -166666.666666667 |
| U114    | subscribed   | S209    | A303      |            523905 |
+---------+--------------+---------+-----------+-------------------+
20 rows in set (0.02 sec)

mysql>
acadgild@localhost:~   start-daemon.sh (~/D...   acadgild@localhost:~
```

```
mysql> select user_type,sum(total_duration) from song_duration where total_duration >=0 group by user_type;
+------------+----------------------+
| user_type  | sum(total_duration) |
+------------+----------------------+
| subscribed |           1904665.65 |
+------------+----------------------+
1 row in set (0.00 sec)

mysql>
acadgild@localhost:~   start-daemon.sh (~/D...   acadgild@localhost:~
```

Music Data Analysis

start-daemon.sh | data_export.sh | log_batch_1 | DataAnalysis.sh | wrapper.sh | log_batch_2

```
#!/bin/bash

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

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

echo "Running script for data analysis using spark..." >> $LOGFILE
chmod 775 /home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar

zip -d /home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar META-INF/*.DSA META-
INF/*.RSA META-INF/*.SF

spark-submit    \
--class Spark_analysis \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/* \
/home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar $batchid

spark-submit    \
--class Spark_analysis_2 \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/* \
/home/acadgild/project/Spark_Practice-1/out/artifacts/Spark_practice_1_jar/Spark_practice-1.jar $batchid

echo "Exporting data to MYSQL using sqoop export..." >> $LOGFILE
sh /home/acadgild/project/scripts/data_export.sh

echo "Incrementing batchid..." >> $LOGFILE
batchid=`expr $batchid + 1`
echo -n $batchid > /home/acadgild/project/logs/current-batch.txt
```

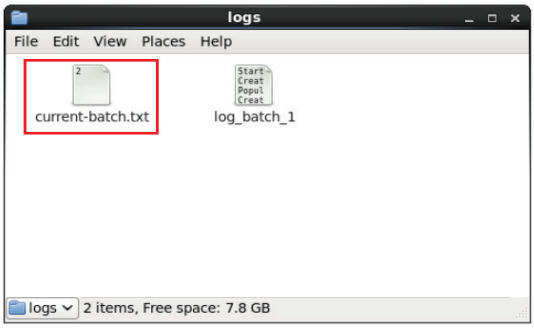
Now after exporting data into MySQL batchid will be incremented to 2 means one batch of data operations is successfully completed and new batch of data will be loaded for the analysis after every 3 hours.

```
[acadgild@localhost scripts]$ cat /home/acadgild/project/logs/current-batch.txt
2[acadgild@localhost scripts]$
```

We can check logs to track the behaviour of the operations we have done on the data and overcome failures in the pipeline and we can see the batchid incremented value in current-batch.txt.

start-daemon.sh | data_export.sh | log_batch_1 | DataAnalysis.sh | wrapper.sh

```
Starting daemons....
Creating Lookup Tables...
Populating Lookup Tables...
Creating hive tables on top hbase tables for data enrichment and filtering...
Placing data files from local to HDFS...
Running pig script for data formatting...
Running hive script for formatted data load...
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...
Running script for data analysis using spark...
Creating mysql tables if not present....
Exporting data to MYSQL using sqoop export...
Running sqoop job for data export...
Incrementing batchid...
```



- Wrapping all the scripts inside the single script file and scheduling this file to run at the periodic interval of every 3 hours.

start-daemon.sh | data_export.sh | DataAnalysis.sh | 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
```

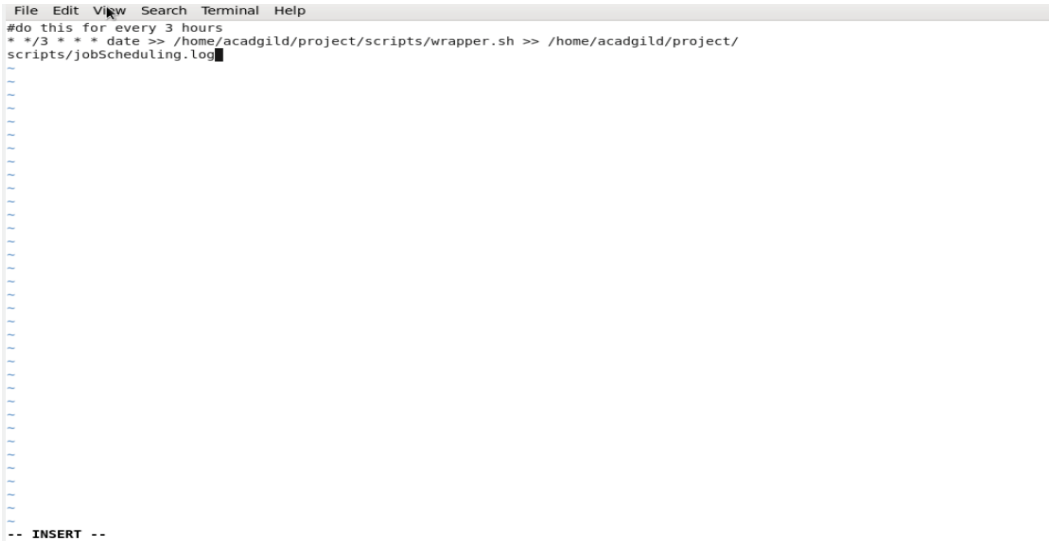
h) Job Scheduling:

- Creating Crontab to schedule the wrapper.sh script to run for every 3 hour interval.

```
[acadgild@localhost ~]$ cd project
[acadgild@localhost project]$ crontab -e
```

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Music Data Analysis




```
[acadgild@localhost project]$ crontab -e
no crontab for acadgild - using an empty one
crontab: installing new crontab
[acadgild@localhost project]$ crontab -l
* */3 * * * /home/acadgild/project/scripts/wrapper.sh >> /home/acadgild/project/
scripts/jobScheduling.log
[acadgild@localhost project]$
```

7. **Conclusion:**
   So we performed all the data operations as per the sequence mentioned in the wrapper.sh file and obtained results successfully for the one of the leading music company.