

Project:-Music Data Analysis

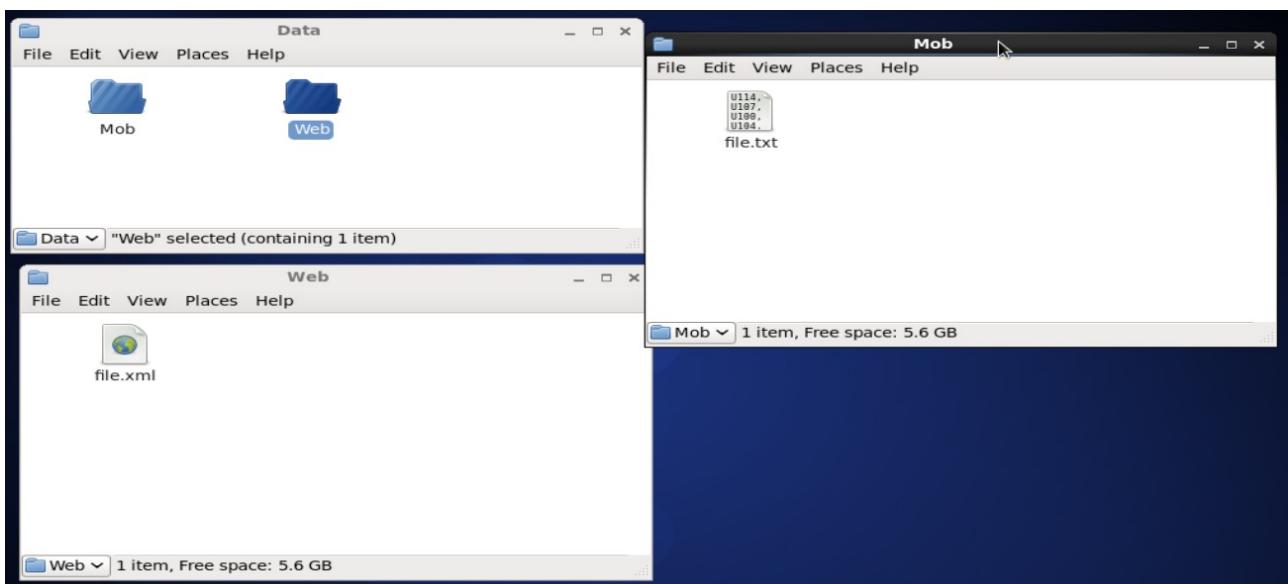
Problem Statement :-

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

Solution:-

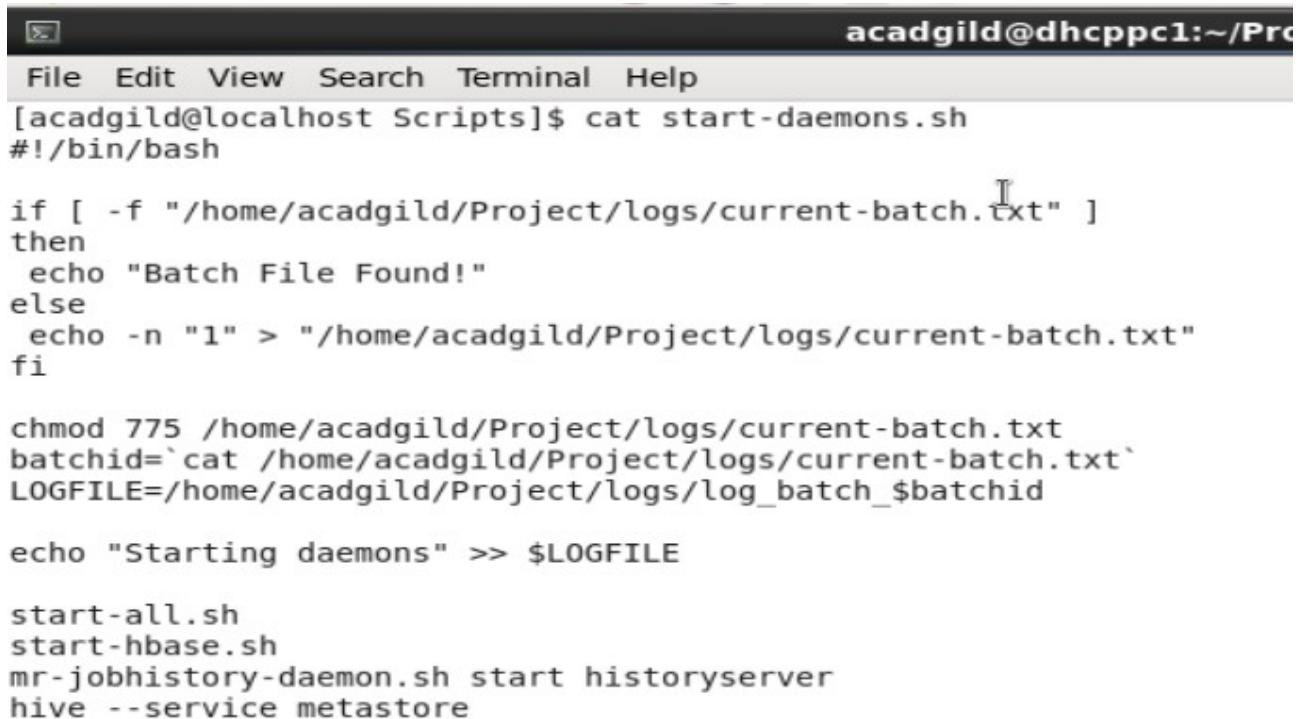
DATASET:

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.



Step 1: To launch required daemons.

- Launch all necessary daemons
- Launch the Mysql Service (needed for Hive)
- Run the shell script start-daemons.sh



```
acadgild@dhcppc1:~/Pro
File Edit View Search Terminal Help
[acadgild@localhost Scripts]$ cat start-daemons.sh
#!/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

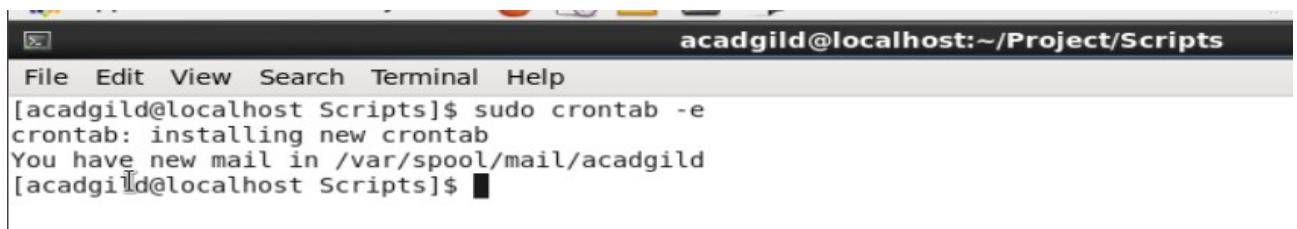
start-all.sh
start-hbase.sh
mr-jobhistory-daemon.sh start historyserver
hive --service metastore
```

The script performs the operations:-

1. Check if a file current-batch.txt has been created or not,
2. If already created, print Batch File Found! else create the file and add 1 to it to signify batch 1.
3. Give permissions to the file, so that we are able to modify it on the run.
4. Get the batch id number from the batch file created above and create a Log File for the batch using the batch id. This will be log_batch_1.

Step 2 : Perform Job Scheduling

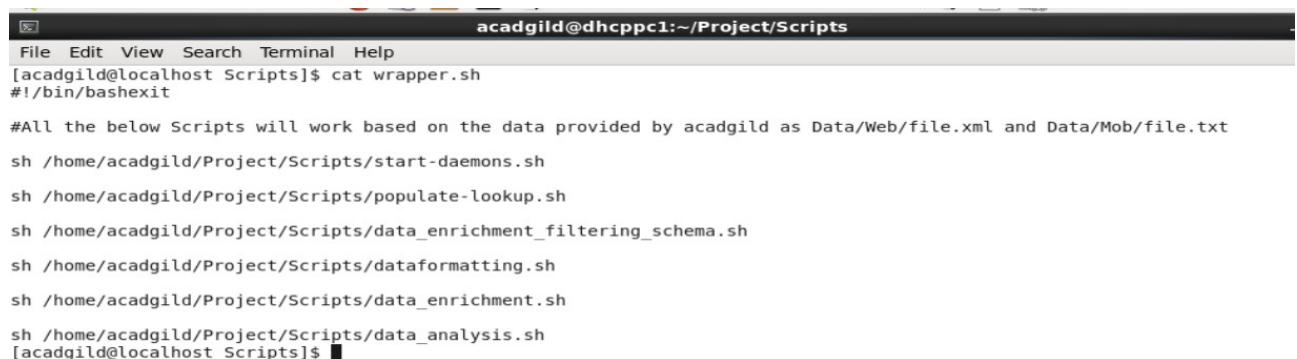
- The shell script **wrapper.sh** has all the scripts for scheduling all the jobs
- Using crontab to schedule a Job in the -e mode. We have scheduled this job for every 3 Hours.



```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@localhost Scripts]$ sudo crontab -e
crontab: installing new crontab
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$
```




```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
*/3 * * * * /home/acadgild/Project/Scripts/wrapper.sh
```



```
acadgild@dhcppc1:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@localhost Scripts]$ cat wrapper.sh
#!/bin/bashexit

#All the below Scripts will work based on the data provided by acadgild as Data/Web/file.xml and Data/Mob/file.txt
sh /home/acadgild/Project/Scripts/start-daemons.sh
sh /home/acadgild/Project/Scripts/populate-lookup.sh
sh /home/acadgild/Project/Scripts/data_enrichment_filtering_schema.sh
sh /home/acadgild/Project/Scripts/dataformatting.sh
sh /home/acadgild/Project/Scripts/data_enrichment.sh
sh /home/acadgild/Project/Scripts/data_analysis.sh
[acadgild@localhost Scripts]$
```

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

Step 3: Populating LookUp tables

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

```
acadgild@dhcppc0:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@dhcppc0 Scripts]$ cat populate-lookup.sh
#!/bin/bash

batchid=`cat /home/acadgild/Project/logs/current-batch.txt`
LOGFILE=/home/acadgild/Project/logs/log_batch_$batchid
echo "Creating LookUp Tables" >> $LOGFILE

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

echo "Populating LookUp Tables" >> $LOGFILE

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

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

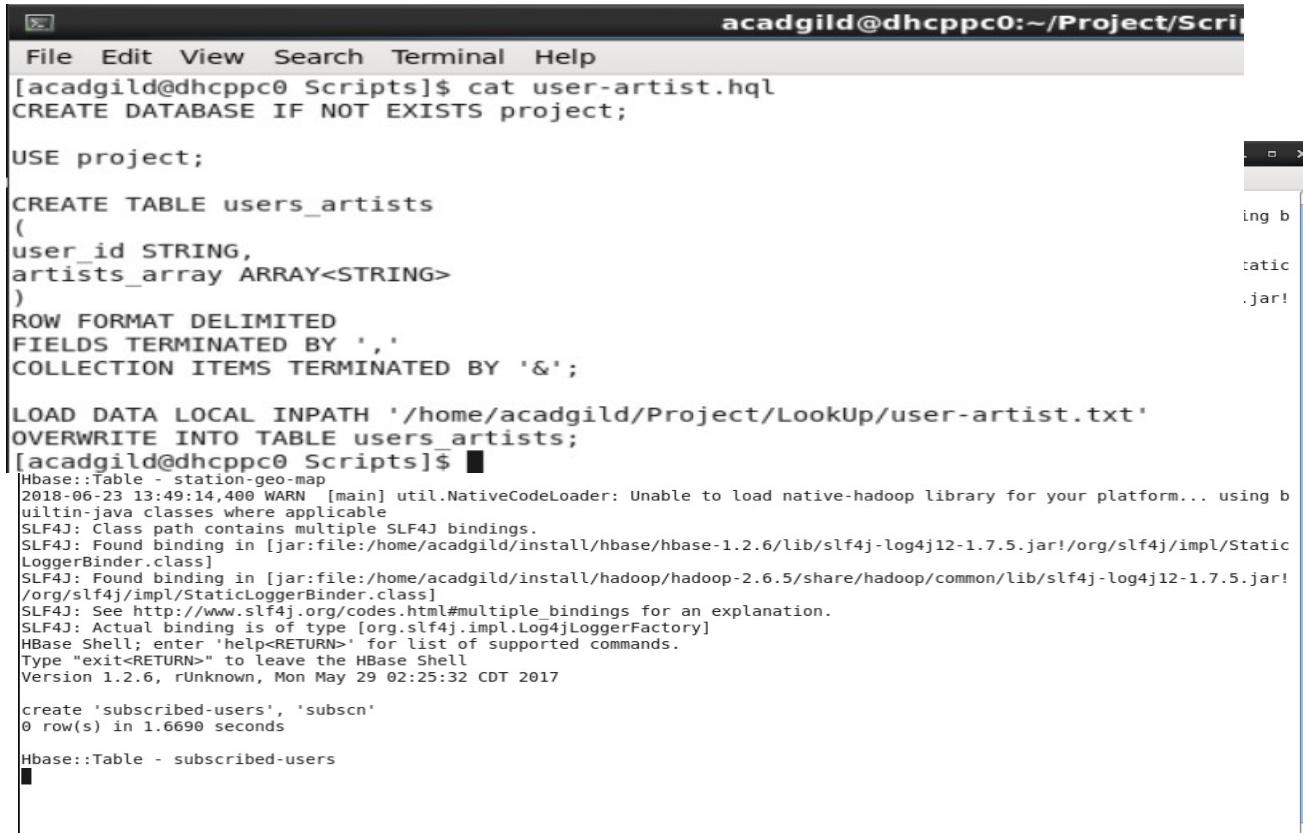
file="/home/acadgild/Project/LookUp/user-subscn.txt"
while IFS= read -r line
do
  userid=`echo $line | cut -d',' -f1`
  startdt=`echo $line | cut -d',' -f2`
```



```
enddt= echo $line | cut -d',' -f3
echo "put 'subscribed-users', '$userid', 'subscn:startdt', '$startdt'" | hbase shell
echo "put 'subscribed-users', '$userid', 'subscn:enddt', '$enddt'" | hbase shell
done <"$file"

hive -f /home/acadgild/Project/Scripts/user-artist.hql
```

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



```

acadgild@dhcppc0:~/Project/Scripts]$ cat user-artist.hql
CREATE DATABASE IF NOT EXISTS project;

USE project;

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

LOAD DATA LOCAL INPATH '/home/acadgild/Project/LookUp/user-artist.txt'
OVERWRITE INTO TABLE users_artists;
[acadgild@dhcppc0 Scripts]$

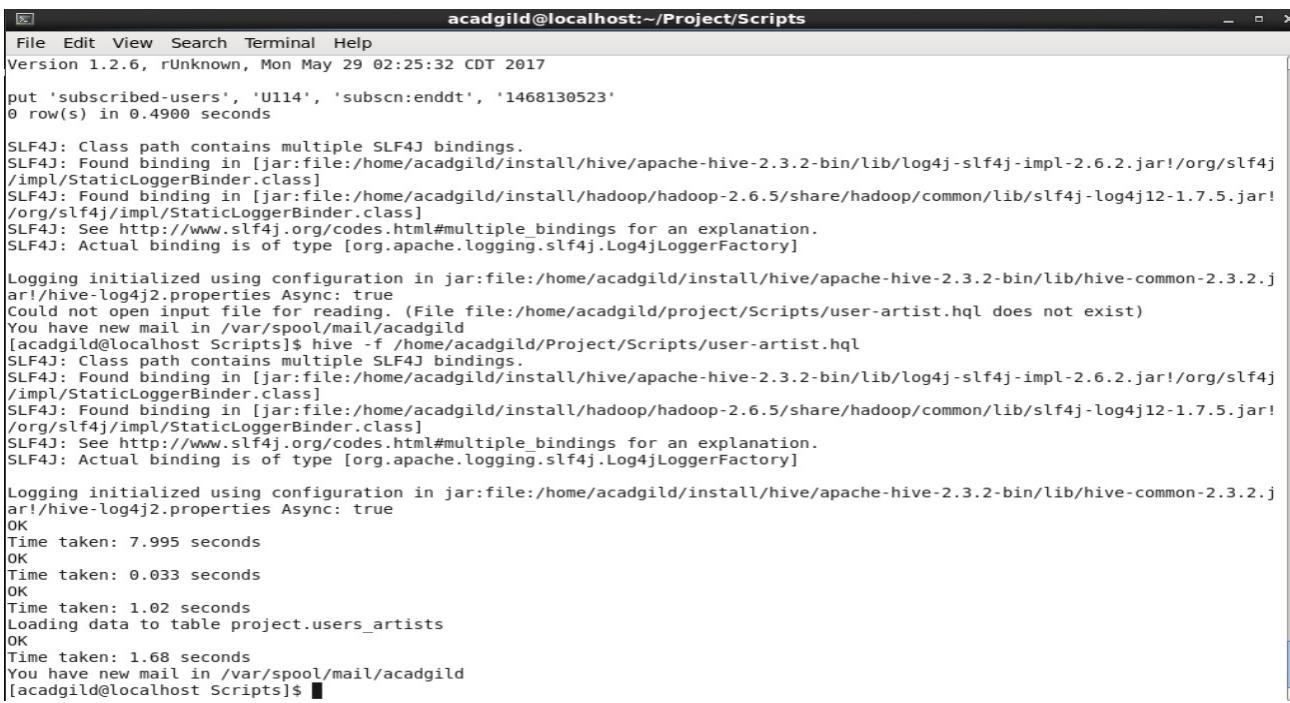
Hbase::Table - station-geo-map
2018-06-23 13:49:14,400 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

create 'subscribed-users', 'subscn'
0 row(s) in 1.6690 seconds

Hbase::Table - subscribed-users

```

- Running **populate-lookup.sh** The following operations are performed:
 1. Get the batch id number from the batch file and get the Log File for the batch using the batch id. This will be batch1
 2. Add logs to the Log File signifying that the lookup tables are being created and populated
 3. Create the HBase tables for the lookup data files: song-artist, stn-geocd and user-subscn with their column families
 4. For every lookup data file, read each line, extract the columns (comma separated) and add the data as rows to the corresponding HBase tables created above
 5. Run the hive script **user-artist.hql**. This will populate a hive table with the data in the lookup data file user-artist. This is because this file has an array column that is difficult to populate in HBase.



```

acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017
put 'subscribed-users', 'U114', 'subscn:enndt', '1468130523'
0 row(s) in 0.4900 seconds

SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.j
ar!/hive-log4j2.properties Async: true
Could not open input file for reading. (File file:/home/acadgild/project/Scripts/user-artist.hql does not exist)
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$ hive -f /home/acadgild/Project/Scripts/user-artist.hql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

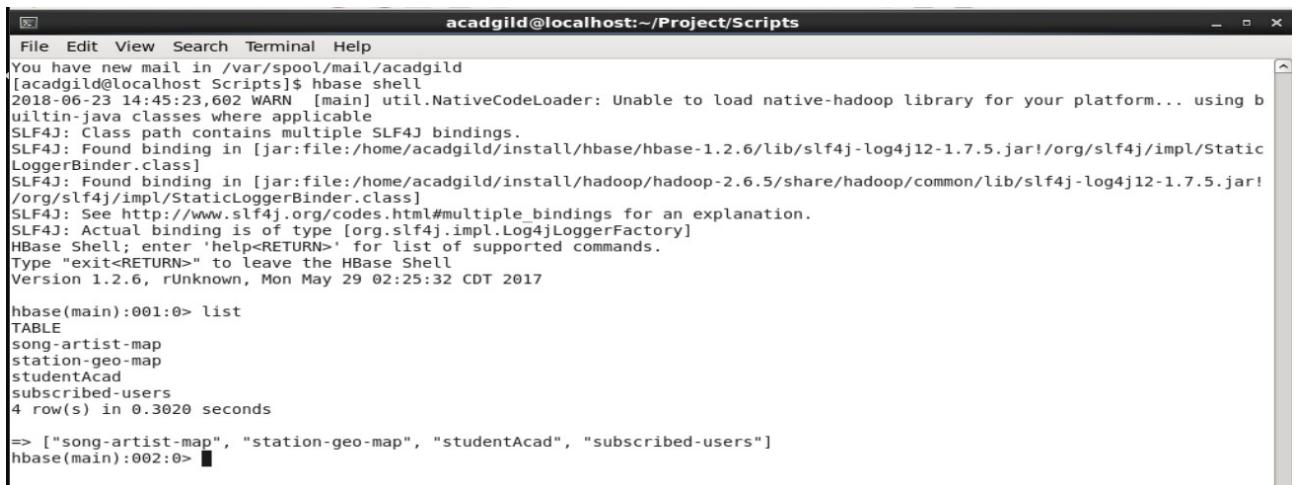
Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.j
ar!/hive-log4j2.properties Async: true
OK
Time taken: 7.995 seconds
OK
Time taken: 0.033 seconds
OK
Time taken: 1.02 seconds
Loading data to table project.users_artists
OK
Time taken: 1.68 seconds
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$ 

```

By using the shell scripting , in the above screenshot we can see 4 lookup tables is created in Hbase NoSQL Database.

- Verifying the tables created in HBase

Command used - list



```

acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$ hbase shell
2018-06-23 14:45:23,602 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> list
TABLE
song-artist-map
station-geo-map
studentAcad
subscribed-users
4 row(s) in 0.3020 seconds

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

```

In the above screenshot it is evident that in the hbase lookup tables are created. Output of the above (Hbase).

- Verifying the tables created in Hive

```

acadgild@localhost:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/:org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/:org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show databases;
OK
default
project
Time taken: 6.996 seconds, Fetched: 2 row(s)
hive> use project;
OK
Time taken: 0.033 seconds
hive> show tables;
OK
users_artists
Time taken: 0.085 seconds, Fetched: 1 row(s)
hive> █

```

- Checking the content of look up tables in Hbase.

```

acadgild@localhost:~/Project/Scripts$ File Edit View Search Terminal Help
hbase(main):002:0> scan 'song-artist-map'
ROW
S200          COLUMN+CELL
S201          column=artist:artistid, timestamp=1529742135301, value=A300
S202          column=artist:artistid, timestamp=1529742145470, value=A301
S203          column=artist:artistid, timestamp=1529742155590, value=A302
S204          column=artist:artistid, timestamp=1529742165466, value=A303
S205          column=artist:artistid, timestamp=1529742175641, value=A304
S206          column=artist:artistid, timestamp=1529742185822, value=A301
S207          column=artist:artistid, timestamp=1529742195754, value=A302
S208          column=artist:artistid, timestamp=1529742206159, value=A303
S209          column=artist:artistid, timestamp=1529742216266, value=A304
                                         column=artist:artistid, timestamp=1529742226456, value=A305
10 row(s) in 0.2470 seconds

hbase(main):002:0> scan 'station-geo-map'
ROW
ST400         COLUMN+CELL
ST401         column=geo:geo_cd, timestamp=1531034877291, value=A
ST402         column=geo:geo_cd, timestamp=1531034887128, value=AU
ST403         column=geo:geo_cd, timestamp=1531034897260, value=AP
ST404         column=geo:geo_cd, timestamp=1531034907451, value=J
ST405         column=geo:geo_cd, timestamp=1531034917503, value=E
ST406         column=geo:geo_cd, timestamp=1531034927520, value=A
ST407         column=geo:geo_cd, timestamp=1531034937375, value=AU
ST408         column=geo:geo_cd, timestamp=1531034947228, value=AP
ST409         column=geo:geo_cd, timestamp=1531034957341, value=E
ST410         column=geo:geo_cd, timestamp=1531034957246, value=E
ST411         column=geo:geo_cd, timestamp=1531034977051, value=A
ST412         column=geo:geo_cd, timestamp=1531034987193, value=A
ST413         column=geo:geo_cd, timestamp=1531034996903, value=AP
ST414         column=geo:geo_cd, timestamp=1531035007002, value=J
                                         column=geo:geo_cd, timestamp=1531035016786, value=E
15 row(s) in 0.2820 seconds

```

```

hbase(main):003:0> scan 'subscribed-users'
ROW                                     COLUMN+CELL
U100                                    column=subscn:enddt, timestamp=1531035136421, value=1465130523
U100                                    column=subscn:startdt, timestamp=1531035126361, value=1465230523
U101                                    column=subscn:enddt, timestamp=1531035157128, value=1475130523
U101                                    column=subscn:startdt, timestamp=1531035147452, value=1465230523
U102                                    column=subscn:enddt, timestamp=1531035176769, value=1475130523
U102                                    column=subscn:startdt, timestamp=1531035167132, value=1465230523
U103                                    column=subscn:enddt, timestamp=1531035196456, value=1475130523
U103                                    column=subscn:startdt, timestamp=1531035186699, value=1465230523
U104                                    column=subscn:enddt, timestamp=1531035216070, value=1475130523
U104                                    column=subscn:startdt, timestamp=1531035206199, value=1465230523
U105                                    column=subscn:enddt, timestamp=1531035235561, value=1475130523
U105                                    column=subscn:startdt, timestamp=1531035225860, value=1465230523
U106                                    column=subscn:enddt, timestamp=1531035255309, value=1485130523
U106                                    column=subscn:startdt, timestamp=1531035245143, value=1465230523
U107                                    column=subscn:enddt, timestamp=1531035275057, value=1455130523
U107                                    column=subscn:startdt, timestamp=1531035265229, value=1465230523
U108                                    column=subscn:enddt, timestamp=1531035294922, value=1465230623
U108                                    column=subscn:startdt, timestamp=1531035285150, value=1465230523
U109                                    column=subscn:enddt, timestamp=1531035314672, value=1475130523
U109                                    column=subscn:startdt, timestamp=1531035305023, value=1465230523
U110                                    column=subscn:enddt, timestamp=1531035334181, value=1475130523
U110                                    column=subscn:startdt, timestamp=1531035324573, value=1465230523
U111                                    column=subscn:enddt, timestamp=1531035353901, value=1475130523
U111                                    column=subscn:startdt, timestamp=1531035343911, value=1465230523
U112                                    column=subscn:enddt, timestamp=1531035373772, value=1475130523
U112                                    column=subscn:startdt, timestamp=1531035363985, value=1465230523
U113                                    column=subscn:enddt, timestamp=1531035393295, value=1485130523
U113                                    column=subscn:startdt, timestamp=1531035383595, value=1465230523
U114                                    column=subscn:enddt, timestamp=1531035413135, value=1468130523
U114                                    column=subscn:startdt, timestamp=1531035403171, value=1465230523
15 row(s) in 0.1890 seconds

```

Step 4: Performing Data Formatting

- Loading and Formatting data using pig and hive using the dataformatting script(i.e **dataformatting.pig**)

```

acadgild@dhcppc1:~/Project/Scripts$ cat dataformatting.pig
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.storageXMLLoader('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${batchid}/formattedweb/' USING PigStorage(',');
[acadgild@dhcppc1 Scripts]$ 

```



```

acadgild@dhcppc1:~/Project/Scripts$ cat dataformatting.sh
#!/bin/bash

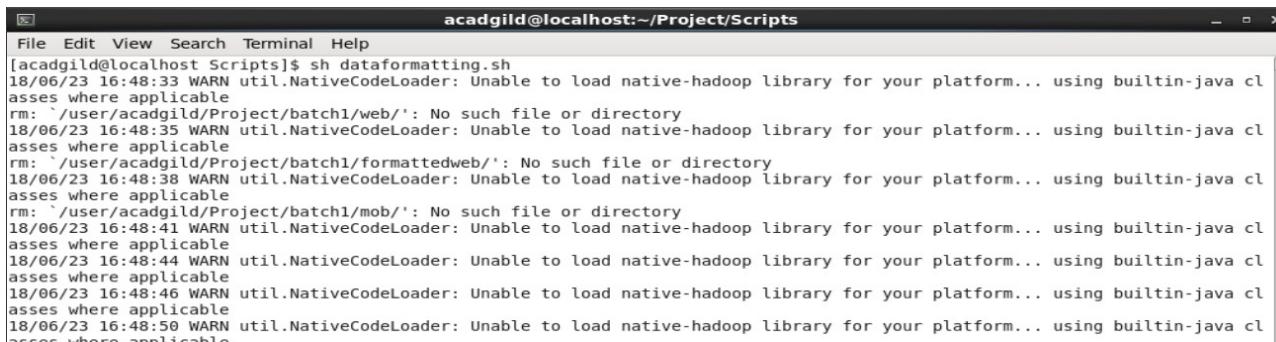
batchid=`cat /home/acadgild/Project/logs/current-batch.txt`
LOGFILE=/home/acadgild/Project/logs/log_batch_$batchid
echo "Placing data files from local to HDFS..." >> $LOGFILE
hadoop fs -rm -r /user/acadgild/Project/batch${batchid}/web/
hadoop fs -rm -r /user/acadgild/Project/batch${batchid}/formattedweb/
hadoop fs -rm -r /user/acadgild/Project/batch${batchid}/mob/
hadoop fs -mkdir -p /user/acadgild/Project/batch${batchid}/web/
hadoop fs -mkdir -p /user/acadgild/Project/batch${batchid}/mob/
hadoop fs -put /home/acadgild/Project/Data/Web/* /user/acadgild/Project/batch${batchid}/web/
hadoop fs -put /home/acadgild/Project/Data/Mob/* /user/acadgild/Project/batch${batchid}/mob/
echo "Running pig script for data formatting..." >> $LOGFILE
pig -param batchid=$batchid /home/acadgild/Project/Scripts/dataformatting.pig
echo "Running hive script for formatted data load..." >> $LOGFILE
hive -hiveconf batchid=$batchid -f /home/acadgild/Project/Scripts/formatted_hive_load.hql

[acadgild@dhcppc1 Scripts]$ 

```

- By running the **dataformatting.sh** file The following operations are performed:

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



```

acadgild@localhost:~/Project/Scripts$ sh dataformatting.sh
18/06/23 16:48:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rm: '/user/acadgild/Project/batch1/web/': No such file or directory
18/06/23 16:48:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rm: '/user/acadgild/Project/batch1/formattedweb/': No such file or directory
18/06/23 16:48:38 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rm: '/user/acadgild/Project/batch1/mob/': No such file or directory
18/06/23 16:48:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/06/23 16:48:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/06/23 16:48:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/06/23 16:48:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

```

```

acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
y tried 4 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:07,832 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 5 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:08,833 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 6 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:09,835 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 7 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:10,836 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 8 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:11,837 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 9 time(s); retry policy is RetryUpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
2018-06-23 16:52:11,938 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Unable to retrieve job to compute warning aggregation.
2018-06-23 16:52:11,938 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success !
2018-06-23 16:52:11,962 [main] INFO org.apache.pig.Main - Pig script completed in 3 minutes, 18 seconds and 440 milliseconds (198440 ms)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
OK
Time taken: 7.936 seconds
OK
Time taken: 0.767 seconds
Loading data to table project.formatted_input partition (batchid=1)
OK
Time taken: 2.319 seconds
Loading data to table project.formatted_input partition (batchid=1)
OK
Time taken: 1.647 seconds
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$ 

```

- Checking the formatted data stored in hive

```

acadgild@localhost:~
File Edit View Search Terminal Help
hive> show tables;
OK
formatted_input
users_artists
Time taken: 0.067 seconds, Fetched: 2 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 1465130523 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 1475130523 1465130523 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 1485130523 1485130523 AU ST415 0 1 1 1
U116 S208 A303 1465230523 1485130523 1475130523 A ST413 1 0 1 1
U101 S202 A300 1465230523 1465130523 1475130523 U ST401 0 0 1 1
U120 S206 A303 1495130523 1485130523 1465130523 AU ST414 0 0 0 1
U106 S205 A300 1462863262 1462863262 1494297562 AP ST407 2 1 1 1
U114 S209 A303 1465490556 1462863262 1494297562 U ST411 2 1 0 1
U113 S203 A304 1465490556 1465490556 1462863262 U ST405 0 0 1 1
U108 S200 A302 1468094889 1462863262 1468094889 U ST414 0 0 1 1
U102 S203 A305 1465490556 1465490556 1494297562 U ST404 2 0 0 1
S208 A300 1465490556 1494297562 1465490556 U ST411 1 0 1 1
U115 S200 A300 1465490556 1494297562 1465490556 AU ST404 3 0 0 1
U111 S204 A300 1465490556 1465490556 1468094889 U ST410 3 1 1 1
U120 S201 A300 1494297562 1465490556 1468094889 ST410 3 0 1 1
U113 S203 1465490556 1465490556 1465490556 A ST402 1 1 0 1
U109 S203 A304 1462863262 1494297562 1468094889 E ST405 1 1 1 1
U110 S202 A303 1494297562 1494297562 1468094889 AU ST402 2 1 0 1

```

- Checking the data files stored in HDFS

```
acadgild@localhost:~$ hadoop fs -ls /user/acadgild/Project/batch1
18/06/23 17:38:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x  - acadgild supergroup          0 2018-06-23 16:52 /user/acadgild/Project/batch1/mob
drwxr-xr-x  - acadgild supergroup          0 2018-06-23 16:48 /user/acadgild/Project/batch1/web
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/Project/batch1
```

Step 5: Performing Data Enrichment and Cleaning

- Creating lookup tables in Hive by importing data from 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.

```
acadgild@dhcppc1:~/Project/Scripts$ cat data_enrichment_filtering_schema.sh
#!/bin/bash

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

echo "Creating hive tables on top of hbase tables for data enrichment and filtering..." >> $LOGFILE
hive -f /home/acadgild/Project/Scripts/create_hive_hbase_lookup.hql

[acadgild@dhcppc1 Scripts]$
```

```
acadgild@dhcppc1:~/Project/Scripts$ cat create_hive_hbase_lookup.hql
USE project;
create external table if not exists station_geo_map
(
station_id String,
geo_cd string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=:key,geo:geo_cd")
tblproperties("hbase.table.name"="station-geo-map");

create external table if not exists subscribed_users
(
user_id STRING,
subscn_start_dt STRING,
subscn_end_dt STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=:key,subscn:startdt,subscn:enddt")
tblproperties("hbase.table.name"="subscribed-users");

create external table if not exists song_artist_map
(
song_id STRING,
artist_id STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=:key,artist:artistid")
tblproperties("hbase.table.name"="song-artist-map");
```

```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@localhost Scripts]$ sh data_enrichment_filtering_schema.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.
ar!/hive-log4j2.properties Async: true
OK
Time taken: 8.145 seconds
OK
Time taken: 3.973 seconds
OK
Time taken: 0.481 seconds
OK
Time taken: 0.397 seconds
[acadgild@localhost Scripts]$
```

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

- Verifying the tables created and data inserted

```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
hive> show tables;
OK
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.048 seconds, Fetched: 5 row(s)
hive> select * from song_artist_map;
OK
S200      A300
S201      A301
S202      A302
S203      A303
S204      A304
S205      A301
S206      A302
S207      A303
S208      A304
S209      A305
Time taken: 0.573 seconds, Fetched: 10 row(s)
hive> select * from station_geo_map;
OK
ST400      A
ST401      AU
ST402      AP
ST403      J
ST404      E
ST405      A
ST406      AU
ST407      AP
ST408      E
ST409      E
ST410      A
ST411      A
ST412      AP
ST413      J
ST414      E
Time taken: 0.621 seconds, Fetched: 15 row(s)
```

```

hive> select * from subscribed_users;
OK
U100    1465230523    1465130523
U101    1465230523    1475130523
U102    1465230523    1475130523
U103    1465230523    1475130523
U104    1465230523    1475130523
U105    1465230523    1475130523
U106    1465230523    1485130523
U107    1465230523    1455130523
U108    1465230523    1465230623
U109    1465230523    1475130523
U110    1465230523    1475130523
U111    1465230523    1475130523
U112    1465230523    1475130523
U113    1465230523    1485130523
U114    1465230523    1468130523
Time taken: 0.568 seconds, Fetched: 15 row(s)
hive> ■

```

```

acadgild@dhcppc2:~ 
File Edit View Search Terminal Help
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    1475130523    1465130523    ST410    2    1    0    1
U105    S203    1475130523    1465230523    1465130523    AU   ST408    2    0    1    1
U110    S203    A300    1465230523    1465130523    1485130523    A    ST415    0    1    1    1
U113    S200    A303    1465230523    1475130523    1465130523    E    ST413    3    1    1    1
U119    S208    A302    1495130523    1465230523    1465230523    U    ST415    3    0    0    1
U118    S208    A303    1475130523    1465130523    1465230523    E    ST415    3    0    0    1
U107    S210    A302    1475130523    1485130523    1485130523    AP   ST404    2    1    0    1
U118    S202    A300    1495130523    1465230523    1465230523    AP   ST410    1    0    0    1
U111    S206    A305    1465130523    1465130523    1485130523    AU   ST415    0    1    1    1
U116    S208    A303    1465230523    1485130523    1475130523    A    ST413    1    0    1    1
U101    S202    A300    1465230523    1465130523    1475130523    U    ST401    0    0    1    1
U120    S206    A303    1495130523    1485130523    1465130523    AU   ST414    0    0    0    1
U106    S205    A300    1462863262    1462863262    1494297562    AP   ST407    2    1    1    1
U114    S209    A303    1465490556    1462863262    1494297562    U    ST411    2    1    0    1
U113    S203    A304    1465490556    1465490556    1462863262    U    ST405    0    0    1    1
U108    S200    A302    1468094889    1462863262    1468094889    U    ST414    0    0    1    1
U102    S203    A305    1465490556    1465490556    1494297562    U    ST404    2    0    0    1
S208    A300    1465490556    1494297562    1465490556    U    ST411    1    0    1    1
U115    S200    A300    1465490556    1494297562    1465490556    AU   ST404    3    0    0    1
U111    S204    A300    1465490556    1465490556    1468094889    U    ST410    3    1    1    1
U120    S201    A300    1494297562    1465490556    1468094889    ST410    3    0    1    1
U113    S203    A304    1465490556    1465490556    1465490556    A    ST402    1    1    0    1
U109    S203    A304    1462863262    1494297562    1468094889    E    ST405    1    1    1    1
U110    S202    A303    1494297562    1494297562    1468094889    AU   ST402    2    1    0    1
U100    S200    A301    1494297562    1494297562    1494297562    AP   ST410    3    1    1    1
U101    S208    A300    1462863262    1468094889    1462863262    E    ST408    0    1    1    1
U106    S206    A300    1494297562    1465490556    1462863262    A    ST405    3    1    0    1
U107    S202    A304    1494297562    1468094889    1462863262    U    ST409    0    0    0    1
U103    S204    A300    1468094889    1494297562    1465490556    AU   ST411    2    1    0    1
Time taken: 3.482 seconds, Fetched: 40 row(s)

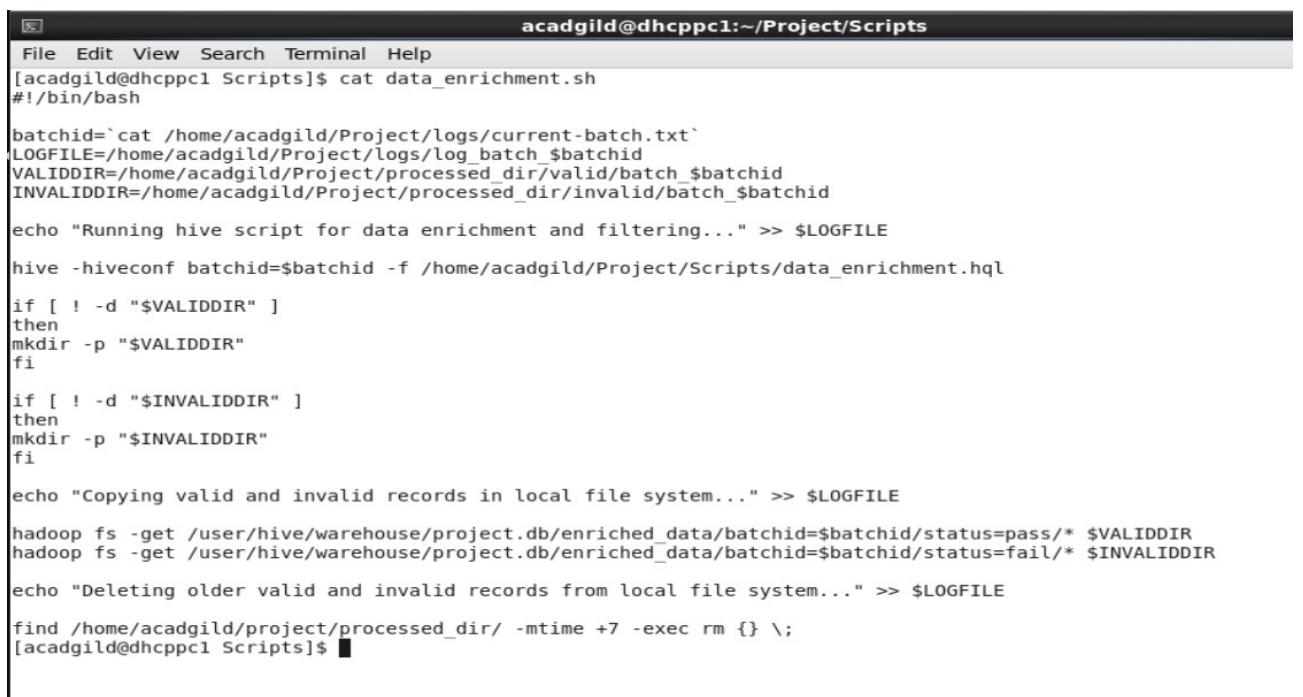
```

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

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



```
acadgild@dhcppc1:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@dhcppc1 Scripts]$ cat data_enrichment.sh
#!/bin/bash

batchid=`cat /home/acadgild/Project/logs/current-batch.txt`
LOGFILE=/home/acadgild/Project/logs/log_batch_$batchid
VALIDDIR=/home/acadgild/Project/processed_dir/valid/batch_$batchid
INVALIDDIR=/home/acadgild/Project/processed_dir/invalid/batch_$batchid

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

hive -hiveconf batchid=$batchid -f /home/acadgild/Project/Scripts/data_enrichment.hql

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

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

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

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

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

find /home/acadgild/project/processed_dir/ -mtime +7 -exec rm {} \;
[acadgild@dhcppc1 Scripts]$
```

```
acadgild@dhcppc1:~/Project/Scr
File Edit View Search Terminal Help
[acadgild@dhcppc1 Scripts]$ cat data_enrichment.hql
SET hive.auto.convert.join=false;
SET hive.exec.dynamic.partition.mode=nonstrict;

USE project;

CREATE TABLE IF NOT EXISTS enriched_data
(
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, status STRING)
STORED AS ORC;

INSERT OVERWRITE TABLE enriched_data
PARTITION (batchid, status)
SELECT
i.user_id,
i.song_id,
sa.artist_id,
i.`timestamp`,
i.start_ts,
i.end_ts,
sg.geo_cd,
i.station_id,
IF (i.song_end_type IS NULL, 3, i.song_end_type) AS song_end_type,
IF (i.`like` IS NULL, 0, i.`like`) AS `like`,
IF (i.dislike IS NULL, 0, i.dislike) AS dislike,
i.batchid,
IF((i.`like`=1 AND i.dislike=1)
OR i.user_id IS NULL
OR i.song_id IS NULL
OR i.`timestamp` IS NULL
OR i.start_ts IS NULL
OR i.end_ts IS NULL
OR i.geo_cd IS NULL
OR i.user_id=''
OR i.song_id=''
OR i.`timestamp`=''
OR i.start_ts=''
OR i.end_ts=''
OR i.geo_cd=''
OR sg.geo_cd IS NULL
OR sg.geo_cd=''
OR sa.artist_id IS NULL
OR sa.artist_id='', 'fail', 'pass') AS status
FROM formatted_input i
LEFT OUTER JOIN station_geo_map sg ON i.station_id = sg.station_id
LEFT OUTER JOIN song_artist_map sa ON i.song_id = sa.song_id
WHERE i.batchid=${hiveconf:batchid};
[acadgild@dhcppc1 Scripts]$ █
```

- **data_enrichment.sh** is executed

```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
[acadgild@localhost Scripts]$ sh data_enrichment.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.j
ar!/hive-log4j2.properties Async: true
OK
Time taken: 8.734 seconds
OK
Time taken: 0.81 seconds
No Stats for project@formatted_input, Columns: start_ts, song_id, user_id, end_ts, like, dislike, station_id, geo_cd, song_en
d_type, timestamp
No Stats for project@station_geo_map, Columns: station_id, geo_cd
No Stats for project@song_artist_map, Columns: song_id, artist_id
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execu
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180623175904_6de48958-cc17-48c9-8372-b781840486d2
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_1529741826906_0007, Tracking URL = http://localhost:8088/proxy/application_1529741826906_0007/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1529741826906_0007
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2018-06-23 17:59:36,674 Stage-1 map = 0%, reduce = 0%
2018-06-23 18:00:00,721 Stage-1 map = 67%, reduce = 0%, Cumulative CPU 3.05 sec
2018-06-23 18:00:04,713 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.16 sec
2018-06-23 18:00:19,997 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.74 sec
MapReduce Total cumulative CPU time: 7 seconds 740 msec
```

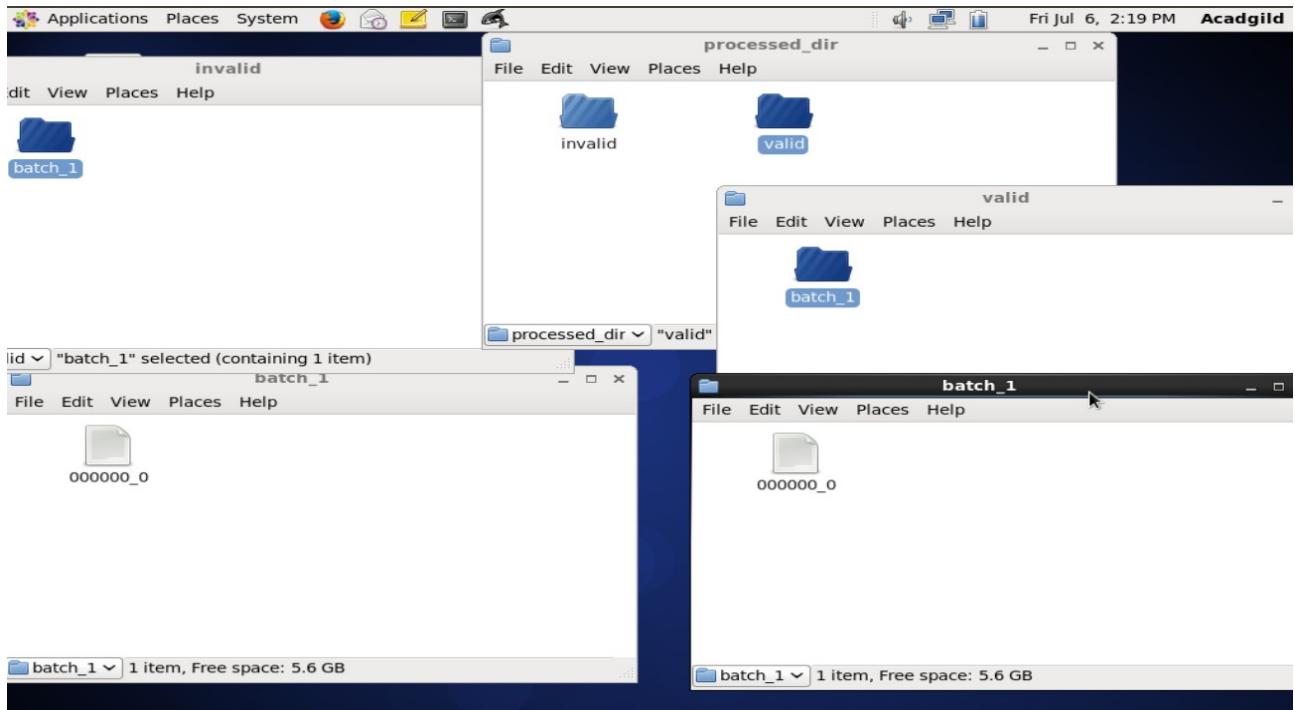
In this phase , a table enriched_data is created and the table is overwritten with the result of the below operations:

1. The data in the formatted_input table is joined with the lookup tables station_geo_map and song_artist_map to fill in the data gaps that can be obtained by said tables.
2. The same data is then filtered by the rules given above and

```
acadgild@localhost:~/Project/Scripts
File Edit View Search Terminal Help
2018-06-23 18:00:04,713 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.16 sec
2018-06-23 18:00:19,997 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.74 sec
MapReduce Total cumulative CPU time: 7 seconds 740 msec
Ended Job = job_1529741826906_0007
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_1529741826906_0008, Tracking URL = http://localhost:8088/proxy/application_1529741826906_0008/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1529741826906_0008
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 1
2018-06-23 18:00:42,514 Stage-2 map = 0%, reduce = 0%
2018-06-23 18:00:56,514 Stage-2 map = 50%, reduce = 0%, Cumulative CPU 1.26 sec
2018-06-23 18:00:58,801 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.39 sec
2018-06-23 18:01:16,218 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.98 sec
MapReduce Total cumulative CPU time: 5 seconds 980 msec
Ended Job = job_1529741826906_0008
Loading data to table project.enriched_data partition (batchid=null, status=null)

Loaded : 2/2 partitions.
  Time taken to load dynamic partitions: 1.69 seconds
  Time taken for adding to write entity : 0.002 seconds
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3 Reduce: 1 Cumulative CPU: 7.74 sec HDFS Read: 49563 HDFS Write: 3067 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 1 Cumulative CPU: 5.98 sec HDFS Read: 24131 HDFS Write: 3195 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 720 msec
OK
Time taken: 136.834 seconds
18/06/23 18:01:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/06/23 18:01:26 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
find: '/home/acadgild/project/processed_dir/': No such file or directory
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Scripts]$
```

partitioned by status (pass or fail) & batchid.



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

A screenshot of a terminal window titled 'hive>'. The window displays the output of a 'show tables;' command. The output lists several tables: 'enriched_data', 'formatted_input', 'song_artist_map', 'station_geo_map', 'subscribed_users', and 'users_artists'. Below the table names, the message 'Time taken: 0.051 seconds, Fetched: 6 row(s)' is shown. The terminal window has a standard menu bar with File, Edit, View, Search, Terminal, and Help.

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

```

hive> select * from enriched_data;
OK
U113 S200 A300 1465230523 1475130523 1465130523 J ST413 3 1 1 1 fail
U100 S200 A300 1494297562 1494297562 1494297562 A ST410 3 1 1 1 fail
U120 S201 A301 1494297562 1465490556 1468094889 A ST410 3 0 1 1 fail
U107 S202 A302 1495130523 1465230523 1465230523 NULL ST415 0 1 1 1 fail
U103 S202 A302 1465490556 1465490556 1465490556 NULL ST415 2 1 1 1 fail
U106 S202 A302 1465230523 1465130523 1465130523 E ST408 0 1 1 1 fail
U109 S203 A303 1462863262 1494297562 1468094889 A ST405 1 1 1 1 fail
S203 A303 1495130523 1475130523 1465230523 A ST400 0 0 1 1 fail
U110 S203 A303 1465230523 1465130523 1485130523 NULL ST415 0 1 1 1 fail
U111 S204 A304 1465490556 1465490556 1468094889 A ST410 3 1 1 1 fail
U113 S204 A304 1494297562 1494297562 1465490556 NULL ST415 3 0 1 1 fail
U100 S204 A304 1495130523 1475130523 1465130523 E ST408 2 1 1 1 fail
U106 S205 A301 1462863262 1462863262 1494297562 AP ST407 2 1 1 1 fail
U108 S205 A301 1465130523 1475130523 1465130523 A ST410 2 1 0 1 fail
U111 S206 A302 1465130523 1465130523 1485130523 NULL ST415 0 1 1 1 fail
U114 S207 A303 1465130523 1465230523 1475130523 NULL ST415 3 1 0 1 fail
U102 S207 A303 1465230523 1485130523 1465230523 J ST403 3 1 1 1 fail
S208 A304 1465490556 1494297562 1465490556 A ST411 1 0 1 1 fail
U118 S208 A304 1475130523 1465130523 1465230523 NULL ST415 3 0 0 1 fail
U119 S208 A304 1495130523 1465230523 1465230523 NULL ST415 3 0 0 1 fail
U101 S208 A304 1462863262 1468094889 1462863262 E ST408 0 1 1 1 fail
U107 S210 NULL 1475130523 1485130523 1485130523 E ST404 2 1 0 1 fail
U115 S200 A300 1465490556 1494297562 1465490556 E ST404 3 0 0 1 pass
U108 S200 A300 1468094889 1462863262 1468094889 E ST414 0 0 1 1 pass
U107 S202 A302 1494297562 1468094889 1462863262 E ST409 0 0 0 1 pass
U101 S202 A302 1465230523 1465130523 1475130523 AU ST401 0 0 1 1 pass
U110 S202 A302 1494297562 1494297562 1468094889 AP ST402 2 1 0 1 pass
U118 S202 A302 1495130523 1465230523 1465230523 A ST410 1 0 0 1 pass
U104 S202 A302 1465230523 1475130523 1465130523 E ST409 2 0 1 1 pass
U102 S203 A303 1465490556 1465490556 1494297562 E ST404 2 0 0 1 pass
U113 S203 A303 1465490556 1465490556 1462863262 A ST405 0 0 1 1 pass
U113 S203 A303 1462863262 1468094889 1494297562 E ST408 2 0 0 1 pass
U105 S203 A303 1475130523 1465230523 1465130523 E ST408 2 0 1 1 pass
U113 S203 A303 1465490556 1465490556 1465490556 AP ST402 1 1 0 1 pass
U103 S204 A304 1468094889 1494297562 1465490556 A ST411 2 1 0 1 pass
U106 S206 A302 1494297562 1465490556 1462863262 A ST405 3 1 0 1 pass
U120 S206 A302 1495130523 1485130523 1465130523 E ST414 0 0 0 1 pass

U105 S207 A303 1465230523 1485130523 1465130523 A ST400 2 0 1 1 pass
U116 S208 A304 1465230523 1485130523 1475130523 J ST413 1 0 1 1 pass
U114 S209 A305 1465490556 1462863262 1494297562 A ST411 2 1 0 1 pass
Time taken: 0.382 seconds. Fetched: 40 row(s)

```

Step 6: Performing Data Analysis

Below is the shell script will **data_analysis.sh** which will perform :

- Get the batch id number from the batch file and get the Log File for the batch using the batch id.
- Add logs to the Log File signifying that the data analysis is being performed using Spark
- and that the result is being exported to the Local FS.
- Run the spark script SparkAnalysis.scala. This will perform the data analysis required in the problem statement given and save the result to the HDFS in csv format.
- Add logs to the Log File signifying that the data analysis has completed and that the batch is being incremented. Here from 1 to 2
- Get batchid number from batch file and increment the batchid by 1.

```
File Edit View Search Terminal Help
acadgild@dhcppc1:~/Project/Scripts
ix-tree-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-procedure-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-protocol-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-resource-bundle-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-rest-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-server-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-server-1.2.6-tests.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-shell-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/hbase-thrift-1.2.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/htrace-core-3.1.0-incubating.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/httpclient-4.2.5.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/httpcore-4.4.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jackson-core-asl-1.9.13.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jackson-jaxrs-1.9.13.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jackson-xmlbuilder-0.4.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jasper-runtime-5.5.23.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jasper-compiler-5.5.23.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jasper-api-2.2.2.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jaxb-api-1.0.8.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jcodings-1.0.8.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-client-1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-core-1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-guice-1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jersey-json-1.9.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jettison-1.3.3.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jetty-6.1.26.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jetty-sslengine-6.1.26.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jetty-util-6.1.26.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/joni-2.1.2.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jruby-complete-1.6.8.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jsch-0.1.42.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jsp-2.1-6.1.14.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/jsp-api-2.1-6.1.14.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/junit-4.12.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/leveldbjni-all-1.8.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/libthrift-0.9.3.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/log4j-1.2.17.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/metrics-core-2.2.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/netty-all-4.0.23.Final.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/paranamer-2.3.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/protobuf-java-2.5.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/servlet-api-2.5-6.1.14.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/snappy-java-1.0.4.1.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/spymemcached-2.11.6.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/xmlenc-0.52.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/xz-1.0.jar:/home/acadgild/install/hbase/hbase-1.2.6/lib/zookeeper-3.4.6.jar:/home/acadgild/install/hadoop/hadoop-2.6.5/etc/hadoop:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/hdfs/lib/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/hdfs/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/lib/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/yarn/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/lib/*:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/*:/home/acadgild/install/hadoop/hadoop-2.6.5/contrib/capacity-scheduler/*.jar does not exist, skipping.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
```

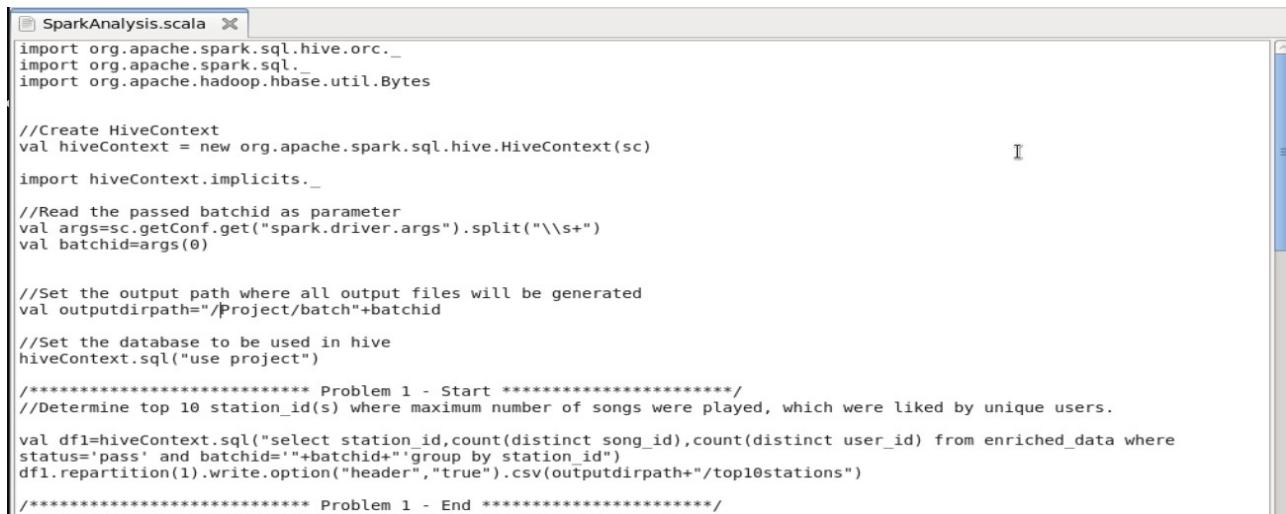
```

Spark context Web UI available at http://192.168.1.3:4040
Spark context available as 'sc' (master = local[*], app id = local-1531057582300).
Spark session available as 'spark'.
Loading /home/acadgild/Project/Scripts/SparkAnalysis.scala...
import org.apache.spark.sql.hive.orc...
import org.apache.spark.sql...
import org.apache.hadoop.hbase.util.Bytes
warning: there was one deprecation warning; re-run with -deprecation for details
hiveContext: org.apache.spark.sql.hive.HiveContext = org.apache.spark.sql.hive.HiveContext@a120b9
import hiveContext.implicits...
args: Array[String] = Array(3)
batchid: String = 3
outputdirpath: String = /Test/batch3
res0: org.apache.spark.sql.DataFrame = []
df1: org.apache.spark.sql.DataFrame = [station_id: string, count(DISTINCT song_id): bigint ... 1 more field]

```

In the **SparkAnalysis.scala** operation performed:

1. Created an hive context.
2. Get the batchid from the batch file and store it in the variable batchid
3. Get the data that was exported and saved in the Local FS from the steps above i.e. enriched_data, subscribed_user and user_artists and perform the foll. on each of them:
4. Create the schema for the data
5. Create a DataFrame from the schema and data
6. Stored the the putput in csv format in HDFS.



```

SparkAnalysis.scala
import org.apache.spark.sql.hive.orc...
import org.apache.spark.sql...
import org.apache.hadoop.hbase.util.Bytes

//Create HiveContext
val hiveContext = new org.apache.spark.sql.hive.HiveContext(sc)

import hiveContext.implicits...

//Read the passed batchid as parameter
val args=sc.getConf.get("spark.driver.args").split("\\s+")
val batchid=args(0)

//Set the output path where all output files will be generated
val outputdirpath="/Project/batch"+batchid

//Set the database to be used in hive
hiveContext.sql("use project")

***** Problem 1 - Start *****
//Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.
val df1=hiveContext.sql("select station_id,count(distinct song_id),count(distinct user_id) from enriched_data where status='pass' and batchid='"+batchid+"' group by station_id")
df1.repartition(1).write.option("header","true").csv(outputdirpath+"/top10stations")

***** Problem 1 - End *****

```

```

SparkAnalysis.scala
import org.apache.spark.sql.hive.orc._
import org.apache.spark.sql._
import org.apache.hadoop.hbase.util.Bytes

//Create HiveContext
val hiveContext = new org.apache.spark.sql.hive.HiveContext(sc)

import hiveContext.implicits._

//Read the passed batchid as parameter
val args=sc.getConf.get("spark.driver.args").split("\\s+")
val batchid=args(0)

//Set the output path where all output files will be generated
val outputdirpath="/Project/batch"+batchid

//Set the database to be used in hive
hiveContext.sql("use project")

***** Problem 1 - Start *****
//Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.

val df1=hiveContext.sql("select station_id,count(distinct song_id),count(distinct user_id) from enriched_data where status='pass' and batchid='"+batchid+"' group by station_id")
df1.repartition(1).write.option("header","true").csv(outputdirpath+"/top10stations")

***** Problem 1 - End *****

```

```

SparkAnalysis.scala
***** Problem 2 - Start *****
//Determine total duration of songs played by each type of user, where type of user can be 'subscribed' or 'unsubscribed'

val df2=hiveContext.sql("SELECT CASE WHEN (su.user_id IS NULL OR CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (su.user_id IS NOT NULL AND CAST(ed.timestamp AS DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'SUBSCRIBED' END AS user_type,SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration FROM enriched_data ed LEFT OUTER JOIN subscribed_users su ON ed.user_id=su.user_id WHERE ed.status='pass' AND ed.batchid='"+batchid+"' GROUP BY CASE WHEN (su.user_id IS NULL OR CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (su.user_id IS NOT NULL AND CAST(ed.timestamp AS DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'SUBSCRIBED' END")

//Write output of the above query to csv file
df2.repartition(1).write.option("header","true").csv(outputdirpath+"/total_songs_played_byeach_usertype")

***** Problem 2 - End *****

***** Problem 3 - Start *****
//Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.

val df3=hiveContext.sql("SELECT ua.artist_id,COUNT(DISTINCT ua.user_id) AS user_count FROM(SELECT user_id, artist_id FROM user_artist_map LATERAL VIEW explode(artists_array) artists AS artist_id ) ua INNER JOIN(SELECT artist_id, song_id, user_id FROM enriched_data WHERE status='pass' AND batchid='"+batchid+"'') ed ON ua.artist_id=ed.artist_id AND ua.user_id=ed.user_id GROUP BY ua.artist_id ORDER BY user_count DESC LIMIT 10")

//Write output of the above query to csv file
df3.repartition(1).write.option("header","true").csv(outputdirpath+"/top_10_connected_artists")

***** Problem 3 - End *****

```

```

***** Problem 4 - Start *****
//Determine top 10 songs who have generated the maximum revenue

val df4 = hiveContext.sql("SELECT song_id,SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL(20,0)))) AS duration FROM enriched_data WHERE status='pass' AND batchid='"+batchid+"' AND (like=1 OR song_end_type=0)GROUP BY song_id ORDER BY duration DESC LIMIT 10")

//Write output of the above query to csv file
df4.repartition(1).write.option("header","true").csv(outputdirpath+"/top_10_songs")

***** Problem 4 - End *****

***** Problem 5 - Start *****
//Determine top 10 unsubscribed users who listened to the songs for the longest duration

val df5=hiveContext.sql("SELECT ed.user_id,SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration FROM enriched_data ed LEFT OUTER JOIN subscribed_users su ON ed.user_id=su.user_id WHERE ed.status='pass' AND ed.batchid='"+batchid+"' AND (su.user_id IS NULL OR (CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))))GROUP BY ed.user_id ORDER BY duration DESC LIMIT 10")

//Write output of the above query to csv file
df5.repartition(1).write.option("header","true").csv(outputdirpath+"/top_10_unsubscribed_songs")

***** Problem 5 - End *****

System.exit(0)

```

- After the execution , if we check the hdfs . In the below screenshot, we can see the analysis is done and the output are stored in folders.

```
acadgild@dhcppc1:~$ hadoop fs -ls /Project/batch1
18/07/08 18:19:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 5 items
drwxr-xr-x - acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top10stations
drwxr-xr-x - acadgild supergroup          0 2018-07-08 18:19 /Project/batch1/top_10_connected_artists
drwxr-xr-x - acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top_10_songs
drwxr-xr-x - acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top_10_unsubscribed_songs
drwxr-xr-x - acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/total_songs_played_byeach_usertype
[acadgild@localhost ~]$
```

Problem Statement 1:

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

Output :

```
acadgild@dhcppc1:~$ hadoop fs -ls /Project/batch1/top10stations
18/07/08 18:20:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top10stations/_SUCCESS
-rw-r--r-- 1 acadgild supergroup        169 2018-07-08 18:14 /Project/batch1/top10stations/part-00000-38870cba-fb1b-42af-baed-267da08293c4-c000.csv
[acadgild@localhost ~]$ hadoop fs -cat /Project/batch1/top10stations/part-00000-38870cba-fb1b-42af-baed-267da08293c4-c000.csv
18/07/08 18:20:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
station_id,count(DISTINCT song_id),count(DISTINCT user_id)
ST402,2,2
ST400,1,1
ST404,2,2
ST414,2,2
ST405,2,2
ST409,1,2
ST410,1,1
ST411,2,2
ST401,1,1
ST408,1,2
ST413,1,1
[acadgild@localhost ~]$
```

Problem Statement 2:

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

Output -

```

acadgild@dhcppc1:~$ hadoop fs -ls /Project/batch1/total_songs_played_byeach_usertype
18/07/08 18:26:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/total_songs_played_byeach_usertype/_SUCCESS
-rw-r--r-- 1 acadgild supergroup        62 2018-07-08 18:14 /Project/batch1/total_songs_played_byeach_usertype/part-00000-f47a6bf1-9c8c-44ce-92a6-b86db8c22cb9-c000.csv
[acadgild@localhost ~]$ hadoop fs -cat /Project/batch1/total_songs_played_byeach_usertype/part-00000-f47a6bf1-9c8c-44ce-92a6-b86db8c22cb9-c000.csv
18/07/08 18:26:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
user_type,duration
UNSUBSCRIBED,98100227
SUBSCRIBED,157978279

```

Problem Statement 3:

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

Output -

```

acadgild@dhcppc1:~$ hadoop fs -ls /Project/batch1/top_10_connected_artists
18/07/08 18:47:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup          0 2018-07-08 18:30 /Project/batch1/top_10_connected_artists/_SUCCESS
-rw-r--r-- 1 acadgild supergroup        35 2018-07-08 18:47 /Project/batch1/top_10_connected_artists/part-00000-dttf0801e-4f56-4c32-65874-93961c3a86c1-c000.csv
[acadgild@localhost ~]$ hadoop fs -cat /Project/batch1/top_10_connected_artists/part-00000-dttf0801e-4f56-4c32-65874-93961c3a86c1-c000.csv
18/07/08 18:48:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
artist_id,user_count
A302,4
A300,1
[acadgild@localhost ~]$ 

```

Problem Statement 4:

Determine top 10 songs who have generated the maximum revenue. Royalty applies to a song only if it was liked or was completed successfully or both.

Output -

```

acadgild@dhcppc1:~$ hadoop fs -ls /Project/batch1/top_10_songs
18/07/08 18:22:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top_10_songs/_SUCCESS
-rw-r--r-- 1 acadgild supergroup        99 2018-07-08 18:14 /Project/batch1/top_10_songs/part-00000-f35a4551-f7e2-4ad8-85dd-7b3052ab56e8-c000.csv
[acadgild@localhost ~]$ hadoop fs -cat /Project/batch1/top_10_songs/part-00000-f35a4551-f7e2-4ad8-85dd-7b3052ab56e8-c000.csv
18/07/08 18:22:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
song_id,duration
S202,41434300
S209,31434300
S204,28807006
S206,22627294
S200,5231627
S203,2627294
[acadgild@localhost ~]$ 

```

Problem Statement 5:

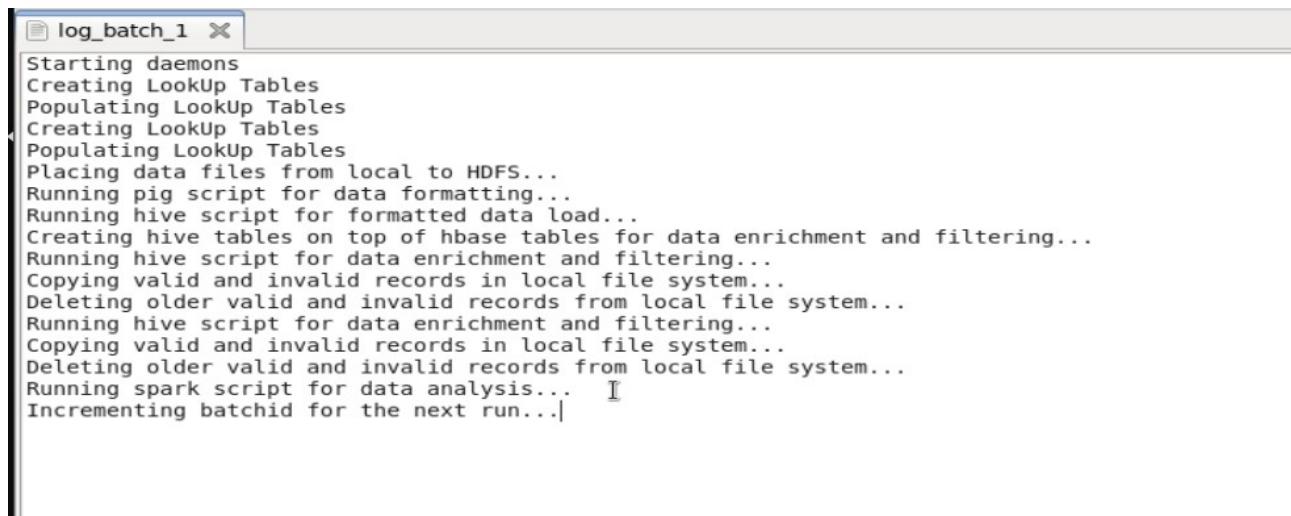
Determine top 10 unsubscribed users who listened to the songs for the longest duration.

Output -

```
acadgild@dhcppc1:~$ File Edit View Search Terminal Help
[acadgild@localhost ~]$ hadoop fs -ls /Project/batch1/top_10_unsubscribed_songs
18/07/08 18:24:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 acadgild supergroup          0 2018-07-08 18:14 /Project/batch1/top_10_unsubscribed_songs/_SUCCESS
-rw-r--r-- 1 acadgild supergroup      119 2018-07-08 18:14 /Project/batch1/top_10_unsubscribed_songs/part-00000-eff0801e-6d58-4c32-9168-93961c3a86b1-c000.csv
[acadgild@localhost ~]$ hadoop fs -cat /Project/batch1/top_10_unsubscribed_songs/part-00000-eff0801e-6d58-4c32-9168-93961c3a86b1-c000.csv
18/07/08 18:25:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
user_id,duration
user_id,duration
U115,28807006
U110,26202673
U120,20000000
U116,10000000
U107,5231627
U108,5231627
U106,2627294
U118,0
[acadgild@localhost ~]$
```

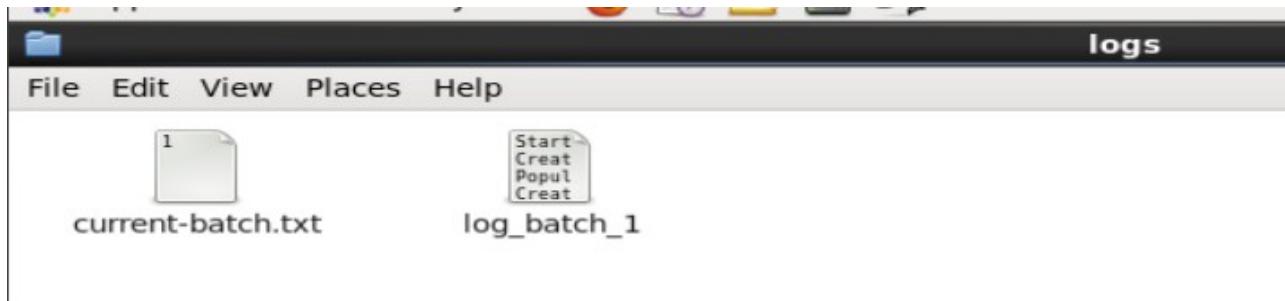
Post Analysis:

- A view of the log file post analysis.



```
log_batch_1
Starting daemons
Creating LookUp Tables
Populating LookUp Tables
Creating LookUp Tables
Populating LookUp Tables
Placing data files from local to HDFS...
Running pig script for data formatting...
Running hive script for formatted data load...
Creating hive tables on top of hbase tables for data enrichment and filtering...
Running hive script for data enrichment and filtering...
Copying valid and invalid records in local file system...
Deleting older valid and invalid records from local file system...
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 spark script for data analysis... I
Incrementing batchid for the next run...]
```

- A view of the log folder:



- The batchid is incremented from 1 to 2 after execution of 1 batch



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

