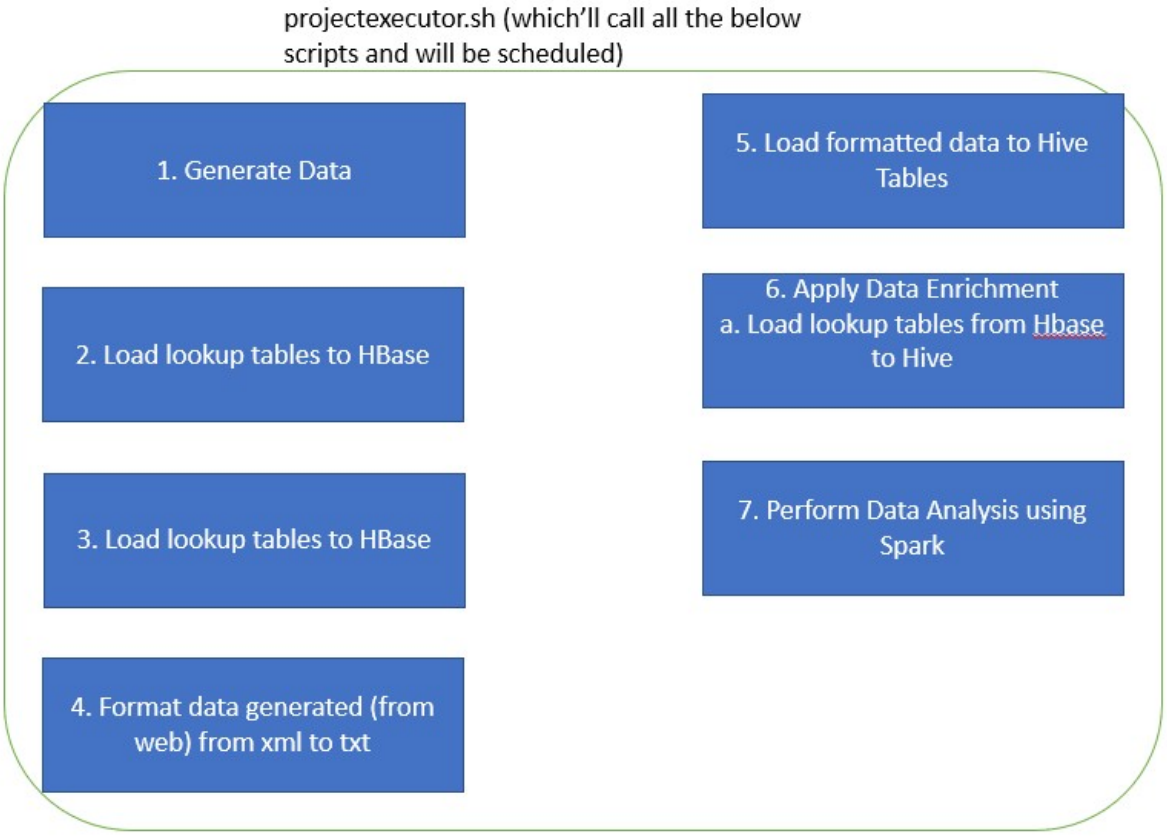
Project – Music Data Analysis

Overall Design:



Step 1: Download the datafiles and lookup files and place them into a directory (generate_mob_data.py & generate_web_data.py)

Project directory structure.

Generating input files will be through two python scripts generate_mob_data.py and generate_web_data.py. These two files will generate files with random data. Generate_mob_data.py will generated data in csv format while generate_web_data.py will generate data in xml format. Once generated they will be placed in /home/maria_dev/project/Web & /home/maria_dev/project/Mob

maria_dev@sandbox-hdp:~/project

```
[maria_dev@sandbox-hdp project]$ pwd
/home/maria_dev/project
[maria_dev@sandbox-hdp project]$ ls
lib  logs  Lookup  Mob  scripts  Web
[maria_dev@sandbox-hdp project]$
```

Step 2: Setting the file for creating logs with new id's (currentbatchno.txt)

We have created a file currentbatchno.txt which holds the start id for the log file.

maria_dev@sandbox-hdp:~/project/logs

```
[maria_dev@sandbox-hdp project]$ cd logs
[maria_dev@sandbox-hdp logs]$ ls
currentbatchno.txt
[maria_dev@sandbox-hdp logs]$ cat currentbatchno.txt
100
[maria_dev@sandbox-hdp logs]$
```

Step 3: Load the data from the files under Lookup folder into the hbase tables. (populatelookup.sh)

populatelookup.sh will load data from lookup files to hbase tables.

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

jobid=`cat /home/maria_dev/project/logs/currentbatchno.txt`

LOGFILE=/home/maria_dev/project/logs/log_batch_$jobid

#---------------------------------------------------
# Create lookup tables in HBase
#---------------------------------------------------
echo "Creating LookUp Tables - Start" >> $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 "Creating LookUp Tables - End" >> $LOGFILE

#---------------------------------------------------
# Read from files are populate the lookup tables
#---------------------------------------------------

#-------------------------------------
#Populate table Station_Geo_Map
#-------------------------------------
file="/home/maria_dev/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"
#-------------------------------------
#Populate table Song_Artist_Map
#-------------------------------------
file="/home/maria_dev/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"
```

```
#---------------------------------
#Populate table Subscribe_Users
#---------------------------------
file="/home/maria_dev/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"

#Call hive script to populate User_Artist_Map table
hive -f /home/maria_dev/project/scripts/user-artist.hql
```

user-artist.hql will load data in User_Artist_Map table.

maria_dev@sandbox-hdp:~/project/scripts

```
CREATE DATABASE IF NOT EXISTS project;

USE project;

CREATE TABLE User_Artist_Map

(
user_id STRING,
artists_array ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '&';
LOAD DATA LOCAL INPATH '/home/maria_dev/project/Lookup/user-artist.txt'

OVERWRITE INTO TABLE User_Artist_Map;
~
~
```

Execute script populatelookupsh

maria_dev@sandbox-hdp:~/project/scripts

```
[maria_dev@sandbox-hdp scripts]$ ./populatelookup.sh
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

create 'Station_Geo_Map', 'geo'
0 row(s) in 5.2530 seconds

Hbase::Table - Station_Geo_Map
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

create 'Subscribed_Users', 'subscn'
0 row(s) in 5.2390 seconds

Hbase::Table - Subscribed_Users
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

create 'Song_Artist_Map', 'artist'
0 row(s) in 4.9950 seconds

Hbase::Table - Song_Artist_Map
```

```
put 'Station_Geo_Map', 'ST400', 'geo:geo_cd', 'A'
0 row(s) in 1.1310 seconds

HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

put 'Station_Geo_Map', 'ST401', 'geo:geo_cd', 'AU'
0 row(s) in 0.7440 seconds
```

```
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

put 'Subscribed_Users', 'U114', 'subscn:startdt', '1465230523'
0 row(s) in 0.7590 seconds

HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.4.0-91, r2a88e694af7238290a5747f963a4fa0079c55bf9, Thu Jan  4 10:32:40 UTC 2018

put 'Subscribed_Users', 'U114', 'subscn:enddt', '1468130523'
0 row(s) in 0.6260 seconds

log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.4.0-91/0/hive-log4j.properties
OK
Time taken: 24.263 seconds
OK
Time taken: 1.204 seconds
OK
Time taken: 9.864 seconds
Loading data to table project.user_artist_map
Table project.user_artist_map stats: [numFiles=1, numRows=0, totalSize=240, rawDataSize=0]
OK
Time taken: 10.89 seconds
[maria_dev@sandbox-hdp scripts]$
```

Once the script has completed successfully, verify the tables and data.

```
hbase(main):001:0> list
TABLE
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/bigdata/deepak/hbase_dir/hbase-0.98.4-hadoop2/lib/slf4j-log4j12-1.6.4.
SLF4J: Found binding in [jar:file:/home/bigdata/hadoop-3.1.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Acadgild_spark_test
Song_Artist_Map
Station_Geo_Map
Subscribed_Users
User_Artist_Map
5 row(s) in 3.0650 seconds

=> ["Acadgild_spark_test", "Song_Artist_Map", "Station_Geo_Map", "Subscribed_Users", "User_Artist_Map"]
hbase(main):002:0>
```

We can see that all the three tables have been created.

```
hbase(main):001:0> list
TABLE
ATLAS_ENTITY_AUDIT_EVENTS
Song_Artist_Map
Station_Geo_Map
Subscribed_Users
atlas_titan
iemployee
test
7 row(s) in 0.8380 seconds

=> ["ATLAS_ENTITY_AUDIT_EVENTS", "Song_Artist_Map", "Station_Geo_Map", "Subscribed_Users", "atlas_titan", "iemployee", "test"]
hbase(main):002:0>
```

Verify the data.

Table – Song_Artist_Map

```
hbase(main):002:0> scan 'Song_Artist_Map'
ROW                                   COLUMN+CELL
 S200                                 column=artist:artistid, timestamp=1529260274579, value=A300
 S201                                 column=artist:artistid, timestamp=1529260298329, value=A301
 S202                                 column=artist:artistid, timestamp=1529260319785, value=A302
 S203                                 column=artist:artistid, timestamp=1529260343184, value=A303
 S204                                 column=artist:artistid, timestamp=1529260363462, value=A304
 S205                                 column=artist:artistid, timestamp=1529260385821, value=A301
 S206                                 column=artist:artistid, timestamp=1529260408672, value=A302
 S207                                 column=artist:artistid, timestamp=1529260430057, value=A303
 S208                                 column=artist:artistid, timestamp=1529260451597, value=A304
 S209                                 column=artist:artistid, timestamp=1529260476335, value=A305
10 row(s) in 0.7980 seconds

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

Table – Station_Geo_Map

```
hbase(main):003:0> scan 'Station_Geo_Map'
ROW                                   COLUMN+CELL
 ST400                                column=geo:geo_cd, timestamp=1529259910496, value=A
 ST401                                column=geo:geo_cd, timestamp=1529259934472, value=AU
 ST402                                column=geo:geo_cd, timestamp=1529259958943, value=AP
 ST403                                column=geo:geo_cd, timestamp=1529259981946, value=J
 ST404                                column=geo:geo_cd, timestamp=1529260003608, value=E
 ST405                                column=geo:geo_cd, timestamp=1529260026701, value=A
 ST406                                column=geo:geo_cd, timestamp=1529260059386, value=AU
 ST407                                column=geo:geo_cd, timestamp=1529260084448, value=AP
 ST408                                column=geo:geo_cd, timestamp=1529260105785, value=E
 ST409                                column=geo:geo_cd, timestamp=1529260128156, value=E
 ST410                                column=geo:geo_cd, timestamp=1529260152831, value=A
 ST411                                column=geo:geo_cd, timestamp=1529260175489, value=A
 ST412                                column=geo:geo_cd, timestamp=1529260199420, value=AP
 ST413                                column=geo:geo_cd, timestamp=1529260226099, value=J
 ST414                                column=geo:geo_cd, timestamp=1529260250186, value=E
15 row(s) in 0.2150 seconds
```

Table – Subscribed_Users

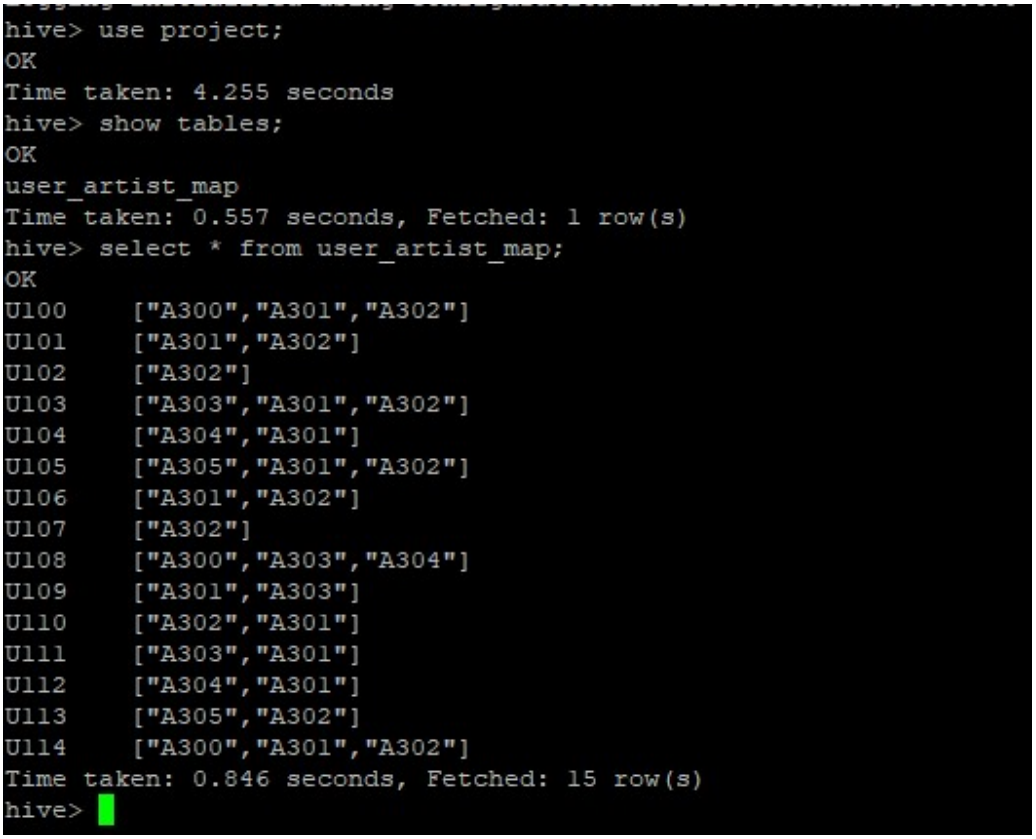
Table – User_Artist_Map

```
hbase(main):006:0> scan 'User_Artist_Map'
ROW                                   COLUMN+CELL
 U100                                 column=user:artistid, timestamp=1529199164521, value=A300&A301&A302
 U101                                 column=user:artistid, timestamp=1529199181710, value=A301&A302
 U102                                 column=user:artistid, timestamp=1529199200594, value=A302
 U103                                 column=user:artistid, timestamp=1529199219833, value=A303&A301&A302
 U104                                 column=user:artistid, timestamp=1529199237534, value=A304&A301
 U105                                 column=user:artistid, timestamp=1529199254863, value=A305&A301&A302
 U106                                 column=user:artistid, timestamp=1529199274590, value=A301&A302
 U107                                 column=user:artistid, timestamp=1529199292779, value=A302
 U108                                 column=user:artistid, timestamp=1529199309951, value=A300&A303&A304
 U109                                 column=user:artistid, timestamp=1529199329563, value=A301&A303
 U110                                 column=user:artistid, timestamp=1529199348187, value=A302&A301
 U111                                 column=user:artistid, timestamp=1529199365150, value=A303&A301
 U112                                 column=user:artistid, timestamp=1529199383349, value=A304&A301
 U113                                 column=user:artistid, timestamp=1529199402861, value=A305&A302
 U114                                 column=user:artistid, timestamp=1529199419443, value=A300&A301&A302
15 row(s) in 0.0790 seconds

hbase(main):007:0>
```

Table – Subscribed_Users

```
hbase(main):004:0> scan 'Subscribed_Users'
ROW                                    COLUMN+CELL
 U100                                  column=subscn:enddt, timestamp=1529260523439, value=1465130523
 U100                                  column=subscn:startdt, timestamp=1529260501008, value=1465230523
 U101                                  column=subscn:enddt, timestamp=1529260567169, value=1475130523
 U101                                  column=subscn:startdt, timestamp=1529260546373, value=1465230523
 U102                                  column=subscn:enddt, timestamp=1529260611388, value=1475130523
 U102                                  column=subscn:startdt, timestamp=1529260589101, value=1465230523
 U103                                  column=subscn:enddt, timestamp=1529260657819, value=1475130523
 U103                                  column=subscn:startdt, timestamp=1529260633276, value=1465230523
 U104                                  column=subscn:enddt, timestamp=1529260705659, value=1475130523
 U104                                  column=subscn:startdt, timestamp=1529260681224, value=1465230523
 U105                                  column=subscn:enddt, timestamp=1529260754493, value=1475130523
 U105                                  column=subscn:startdt, timestamp=1529260729630, value=1465230523
 U106                                  column=subscn:enddt, timestamp=1529260802885, value=1485130523
 U106                                  column=subscn:startdt, timestamp=1529260778120, value=1465230523
 U107                                  column=subscn:enddt, timestamp=1529260847048, value=1455130523
 U107                                  column=subscn:startdt, timestamp=1529260824287, value=1465230523
 U108                                  column=subscn:enddt, timestamp=1529260895991, value=1465230623
 U108                                  column=subscn:startdt, timestamp=1529260870847, value=1465230523
 U109                                  column=subscn:enddt, timestamp=1529260941441, value=1475130523
 U109                                  column=subscn:startdt, timestamp=1529260919574, value=1465230523
 U110                                  column=subscn:enddt, timestamp=1529260986287, value=1475130523
 U110                                  column=subscn:startdt, timestamp=1529260963685, value=1465230523
 U111                                  column=subscn:enddt, timestamp=1529261029697, value=1475130523
 U111                                  column=subscn:startdt, timestamp=1529261008341, value=1465230523
 U112                                  column=subscn:enddt, timestamp=1529261072059, value=1475130523
 U112                                  column=subscn:startdt, timestamp=1529261052411, value=1465230523
 U113                                  column=subscn:enddt, timestamp=1529261114343, value=1485130523
 U113                                  column=subscn:startdt, timestamp=1529261093608, value=1465230523
 U114                                  column=subscn:enddt, timestamp=1529261154542, value=1468130523
 U114                                  column=subscn:startdt, timestamp=1529261135381, value=1465230523
15 row(s) in 0.4390 seconds
```

user_artist_map table



So, we can see that all the tables have been created and all the required data also have been loaded.

Step 4 – Data Formatting (formatdata.sh)

In this step we will format the file file.xml under /home/maria_dev/project/Web to text format, which will later loaded along with /home/maria_dev/project/Mob/file.txt for data enrichment.

We will use pig to convert the xml file to txt file.

Batch script formatdata.sh to invoke the pig script formatdata.pig

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

#--------------------------------------------
#Fetch the batch id
#--------------------------------------------
batchid=`cat /home/maria_dev/project/logs/currentbatchno.txt`
LOGFILE=/home/maria_dev/project/logs/log_batch_$batchid


echo "Placing data files from local to HDFS..." >> $LOGFILE
#--------------------------------------------
#Delete the existing folders if exists
#--------------------------------------------
hadoop fs -rm -r /maria_dev/project/batch${batchid}/Web/
hadoop fs -rm -r /maria_dev/project/batch${batchid}/formattedweb/
hadoop fs -rm -r /maria_dev/project/batch${batchid}/Mob/
#--------------------------------------------
#Again Create the folder structures
#--------------------------------------------
hadoop fs -mkdir -p /maria_dev/project/batch${batchid}/Web/
hadoop fs -mkdir -p /maria_dev/project/batch${batchid}/Mob/

#-----------------------------------------------------------------------
#Put the xml file and the txt file from Web and Mobile directory to HDFS
#-----------------------------------------------------------------------
hadoop fs -put /home/maria_dev/project/Web/* /maria_dev/project/batch${batchid}/Web/
hadoop fs -put /home/maria_dev/project/Mob/* /maria_dev/project/batch${batchid}/Mob/

echo "Running pig script for data formatting..." >> $LOGFILE

#--------------------------------------------
#Invoke the pig script to format xml to txt
#--------------------------------------------
pig -param batchid=$batchid /home/maria_dev/project/scripts/formatdata.pig

echo "Running hive script for formatted data load..." >> $LOGFILE
```

Pig Script formatdata.pig

maria_dev@sandbox-hdp:~/project/scripts

```
REGISTER /usr/hdp/current/pig-client/lib/piggybank.jar;

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

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

Execute the batch script formatdata.sh which invoked formatdata.pig to format the xml file.

```
[maria_dev@sandbox-hdp scripts]$ ./formatdata.sh
rm: `/maria_dev/project/batch100/Web/': No such file or directory
rm: `/maria_dev/project/batch100/formattedweb/': No such file or directory
rm: `/maria_dev/project/batch100/Mob/': No such file or directory
18/06/17 20:02:41 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
18/06/17 20:02:41 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
18/06/17 20:02:41 INFO pig.ExecTypeProvider: Trying ExecType : TEZ_LOCAL
18/06/17 20:02:41 INFO pig.ExecTypeProvider: Trying ExecType : TEZ
18/06/17 20:02:41 INFO pig.ExecTypeProvider: Picked TEZ as the ExecType
2018-06-17 20:02:41,823 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0.2.6.4.0-91 (rexported) compiled Jan 04 2018, 10:39:57
2018-06-17 20:02:41,824 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/maria_dev/project/scripts/pig_1529265761820.log
2018-06-17 20:02:43,990 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/maria_dev/.pigbootup not found
2018-06-17 20:02:44,418 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://sandbox-hdp.h
ortonworks.com:8020
2018-06-17 20:02:45,816 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-formatdata.pig-5434fd46-73cd-4c1d-be43-8d0a3fea4ce2
2018-06-17 20:02:47,224 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox-hdp.hortonworks.com:8
188/ws/v1/timeline/
2018-06-17 20:02:47,918 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
2018-06-17 20:02:51,422 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2018-06-17 20:02:51,540 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2018-06-17 20:02:51,672 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCal
culator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, PartitionFilterOptimizer, PredicatePushdownOptimizer, Pus
hDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2018-06-17 20:02:51,902 [main] INFO  org.apache.pig.impl.util.SpillableMemoryManager - Selected heap (PS Old Gen) of size 699400192 to monitor. collectionUsage
Threshold = 489580128, usageThreshold = 489580128
2018-06-17 20:02:52,297 [main] INFO  org.apache.pig.backend.hadoop.executionengine.tez.TezLauncher - Tez staging directory is /tmp/maria_dev/staging and resour
ces directory is /tmp/temp-1064519882
2018-06-17 20:02:52,533 [main] INFO  org.apache.pig.backend.hadoop.executionengine.tez.plan.TezCompiler - File concatenation threshold: 100 optimistic? false
2018-06-17 20:02:53,260 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2018-06-17 20:02:53,266 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2018-06-17 20:02:53,333 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2018-06-17 20:02:57,114 [main] INFO  org.apache.pig.backend.hadoop.executionengine.tez.TezJobCompiler - Local resource: joda-time-2.9.4.jar
```

```
Success!


DAG 0:
                                  Name: PigLatin:formatdata.pig-0_scope-0
                         ApplicationId: job_1529249751056_0005
                    TotalLaunchedTasks: 1
                         FileBytesRead: 0
                      FileBytesWritten: 0
                         HdfsBytesRead: 6716
                      HdfsBytesWritten: 1236
      SpillableMemoryManager spill count: 0
                Bags proactively spilled: 0
            Records proactively spilled: 0

DAG Plan:
Tez vertex scope-71

Vertex Stats:
VertexId Parallelism TotalTasks   InputRecords   ReduceInputRecords  OutputRecords  FileBytesRead FileBytesWritten  HdfsBytesRead HdfsBytesWritten Alias
eature  Outputs
scope-71          1          1             20                   0             20              0               0           6716            1236 A,B
maria_dev/project/batch100/formattedweb,

Input(s):
Successfully read 20 records (6716 bytes) from: "/maria_dev/project/batch100/Web"

Output(s):
Successfully stored 20 records (1236 bytes) in: "/maria_dev/project/batch100/formattedweb"

2018-06-17 20:03:46,225 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 5 seconds and 194 milliseconds (65194 ms)
2018-06-17 20:03:46,225 [main] INFO  org.apache.pig.backend.hadoop.executionengine.tez.TezLauncher - Shutting down thread pool
2018-06-17 20:03:46,452 [pool-1-thread-1] INFO  org.apache.pig.backend.hadoop.executionengine.tez.TezSessionManager - Shutting down Tez session org.apache.te
client.TezClient@119de226
2018-06-17 20:03:46 Shutting down Tez session , sessionName=PigLatin:formatdata.pig, applicationId=application_1529249751056_0005
2018-06-17 20:03:46,459 [pool-1-thread-1] INFO  org.apache.tez.client.TezClient - Shutting down Tez Session, sessionName=PigLatin:formatdata.pig, application
=application_1529249751056_0005
[maria_dev@sandbox-hdp scripts]$
```

Now, we can see that the file.xml has been converted to txt file

```
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project/batch100/formattedweb
Found 2 items
-rw-r--r--   1 maria_dev hdfs          0 2018-06-17 20:03 /maria_dev/project/batch100/formattedweb/_SUCCESS
-rw-r--r--   1 maria_dev hdfs       1236 2018-06-17 20:03 /maria_dev/project/batch100/formattedweb/part-v000-o000-r-00000
[maria_dev@sandbox-hdp scripts]$ hadoop fs -cat /maria_dev/project/batch100/formattedweb/part-v000-o000-00000
cat: `/maria_dev/project/batch100/formattedweb/part-v000-o000-00000': No such file or directory
[maria_dev@sandbox-hdp scripts]$ hadoop fs -cat /maria_dev/project/batch100/formattedweb/part-v000-o000-r-00000
U106,S205,A300,1462883062,1462883062,1494317362,AP,ST407,2,1,1
U114,S209,A303,1465510356,1462883062,1494317362,U,ST411,2,1,0
U113,S203,A304,1465510356,1465510356,1462883062,U,ST405,0,0,1
U108,S200,A302,1468114689,1462883062,1468114689,U,ST414,0,0,1
U102,S203,A305,1465510356,1465510356,1494317362,U,ST404,2,0,0
,S208,A300,1465510356,1494317362,1465510356,U,ST411,1,0,1
U115,S200,A300,1465510356,1494317362,1465510356,AU,ST404,3,0,0
U111,S204,A300,1465510356,1465510356,1468114689,U,ST410,3,1,1
U120,S201,A300,1494317362,1465510356,1468114689,,ST410,3,0,1
U113,S203,,1465510356,1465510356,1465510356,A,ST402,1,1,0
U109,S203,A304,1462883062,1494317362,1468114689,E,ST405,1,1,1
U110,S202,A303,1494317362,1494317362,1468114689,AU,ST402,2,1,0
U100,S200,A301,1494317362,1494317362,1494317362,AP,ST410,3,1,1
U101,S208,A300,1462883062,1468114689,1462883062,E,ST408,0,1,1
U106,S206,A300,1494317362,1465510356,1462883062,A,ST405,3,1,0
U107,S202,A304,1494317362,1468114689,1462883062,U,ST409,0,0,0
U103,S204,A300,1468114689,1494317362,1465510356,AU,ST411,2,1,0
U103,S202,A300,1465510356,1465510356,1465510356,A,ST415,2,1,1
U113,S203,A303,1462883062,1468114689,1494317362,U,ST408,2,0,0
U113,S204,A301,1494317362,1494317362,1465510356,E,ST415,3,0,1
[maria_dev@sandbox-hdp scripts]$
```

Also, we have just pasted the file.txt from local Mob directory to HDFS Mob directory

```
[maria_dev@sandbox-hdp scripts]$ hadoop fs -cat /maria_dev/project/batch100/Mob/file.txt
U114,S207,A303,1465130523,1465230523,1475130523,A,ST415,3,1,0
U107,S202,A303,1495130523,1465230523,1465230523,U,ST415,0,1,1
U100,S204,A302,1495130523,1475130523,1465130523,AU,ST408,2,1,1
U104,S202,A303,1465230523,1475130523,1465130523,A,ST409,2,0,1
U102,S207,A301,1465230523,1485130523,1465230523,AU,ST403,3,1,1
,S203,A302,1495130523,1475130523,1465230523,E,ST400,0,0,1
U106,S202,A302,1465230523,1465130523,1465130523,AU,ST408,0,1,1
U105,S207,A300,1465230523,1485130523,1465130523,U,ST400,2,0,1
U108,S205,A304,1465130523,1465130523,1475130523,,ST410,2,1,0
U105,S203,,1475130523,1465230523,1465130523,AU,ST408,2,0,1
U110,S203,A300,1465230523,1465130523,1485130523,A,ST415,0,1,1
U113,S200,A303,1465230523,1475130523,1465130523,E,ST413,3,1,1
U119,S208,A302,1495130523,1465230523,1465230523,U,ST415,3,0,0
U118,S208,A303,1475130523,1465130523,1465230523,E,ST415,3,0,0
U107,S210,A302,1475130523,1485130523,1485130523,AP,ST404,2,1,0
U118,S202,A300,1495130523,1465230523,1465230523,AP,ST410,1,0,0
U111,S206,A305,1465130523,1465130523,1485130523,AU,ST415,0,1,1
U116,S208,A303,1465230523,1485130523,1475130523,A,ST413,1,0,1
U101,S202,A300,1465230523,1465130523,1475130523,U,ST401,0,0,1
U120,S206,A303,1495130523,1485130523,1465130523,AU,ST414,0,0,0
[maria_dev@sandbox-hdp scripts]$
```

Now, we will invoke hive sql script to load these data into hive table so that data enrichment rules can be applied to this data.

Batch script (loadformatteddata.sh) to invoke hive script.

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

#-----------------------------------------
#Fetch the batch id
#-----------------------------------------
batchid=`cat /home/maria_dev/project/logs/currentbatchno.txt`
LOGFILE=/home/maria_dev/project/logs/log_batch_$batchid

#Invoke the hive script
hive -hiveconf batchid=$batchid -f /home/maria_dev/project/scripts/load_formatted_data.hql
~
~
```

Hive Script – load_formatted_data.hql

maria_dev@sandbox-hdp:~/project/scripts

```
------------------------------------------------------------
-- Created By: Deepak Ray
-- Date: 18/06/2018
-- Project: Acadgild Music Data Analysis
------------------------------------------------------------
-- Load formatted data from Mod and Web directories  to hive.
------------------------------------------------------------
USE project;

--Use this statement to avoid reserve word exception
SET hive.support.sqlll.reserved.keywords=false;

--Create the table
CREATE TABLE IF NOT EXISTS formatted_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)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';

--Load web data from formattedweb folder
LOAD DATA INPATH '/maria_dev/project/batch${hiveconf:batchid}/formattedweb/'
INTO TABLE formatted_data PARTITION (batchid=${hiveconf:batchid});

--Load mobile data from Mob folder
LOAD DATA INPATH '/maria_dev/project/batch${hiveconf:batchid}/Mob/'
INTO TABLE formatted_data PARTITION (batchid=${hiveconf:batchid});

~
~
```

Execute the script ./foadformatteddata.sh

maria_dev@sandbox-hdp:~/project/scripts

```
[maria_dev@sandbox-hdp scripts]$ ./loadformatteddata.sh
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.4.0-91/0/hive-log4j.properties
OK
Time taken: 8.664 seconds
OK
Time taken: 0.779 seconds
Loading data to table project.formatted_data partition (batchid=100)
chgrp: changing ownership of 'hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/project.db/formatted_data/batchid=1
hadoop
Partition project.formatted_data{batchid=100} stats: [numFiles=1, numRows=0, totalSize=1236, rawDataSize=0]
OK
Time taken: 6.756 seconds
Loading data to table project.formatted_data partition (batchid=100)
Partition project.formatted_data{batchid=100} stats: [numFiles=2, numRows=0, totalSize=2475, rawDataSize=0]
OK
Time taken: 8.264 seconds
[maria_dev@sandbox-hdp scripts]$
```

Verify the data in hive

```
OK
Time taken: 6.245 seconds
hive> select * from formatted_data;
OK
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       100
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       100
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       100
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       100
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       100
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       100
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       100
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       100
U108    S205    A304    1465130523      1465130523      1475130523              ST410   2       1       0       100
U105    S203            1475130523      1465230523      1465130523      AU      ST408   2       0       1       100
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       100
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       100
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       100
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       100
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       100
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       100
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       100
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       100
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       100
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       100
U106    S205    A300    1462883062      1462883062      1494317362      AP      ST407   2       1       1       100
U114    S209    A303    1465510356      1462883062      1494317362      U       ST411   2       1       0       100
U113    S203    A304    1465510356      1465510356      1462883062      U       ST405   0       0       1       100
U108    S200    A302    1468114689      1462883062      1468114689      U       ST414   0       0       1       100
U102    S203    A305    1465510356      1465510356      1494317362      U       ST404   2       0       0       100
        S208    A300    1465510356      1494317362      1465510356      U       ST411   1       0       1       100
U115    S200    A300    1465510356      1494317362      1465510356      AU      ST404   3       0       0       100
U111    S204    A300    1465510356      1465510356      1468114689      U       ST410   3       1       1       100
U120    S201    A300    1494317362      1465510356      1468114689              ST410   3       0       1       100
U113    S203            1465510356      1465510356      1465510356      A       ST402   1       1       0       100
U109    S203    A304    1462883062      1494317362      1468114689      E       ST405   1       1       1       100
U110    S202    A303    1494317362      1494317362      1468114689      AU      ST402   2       1       0       100
U100    S200    A301    1494317362      1494317362      1494317362      AP      ST410   3       1       1       100
U101    S208    A300    1462883062      1468114689      1462883062      E       ST408   0       1       1       100
U106    S206    A300    1494317362      1465510356      1462883062      A       ST405   3       1       0       100
U107    S202    A304    1494317362      1468114689      1462883062      U       ST409   0       0       0       100
U103    S204    A300    1468114689      1494317362      1465510356      AU      ST411   2       1       0       100
U103    S202    A300    1465510356      1465510356      1465510356      A       ST415   2       1       1       100
U113    S203    A303    1462883062      1468114689      1494317362      U       ST408   2       0       0       100
U113    S204    A301    1494317362      1494317362      1465510356      E       ST415   3       0       1       100
Time taken: 2.007 seconds, Fetched: 40 row(s)
hive>
```

In the next step we will apply data enrichment rules.

Step 5 – Data Enrichment (apply_data_enrichment.sh)

Now, once data is uploaded into the tables, we will perform the following formatting.

Data Enrichment

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.

NULL or absent field	Look up field	Look up table (Table from which record can be updated)
Geo_cd	Station_id	Station_Geo_Map
Artist_id	Song_id	Song_Artist_Map

We can see in the previous step that many of the values are blank. So, in order to correct those we have to apply the above rules.

But, before that, we need to load the lookup tables from hbase to hive. So, that these tables can be used in hive to carry out data enrichment.

Script – load_lookupdata.hql

maria_dev@sandbox-hdp:~/project/scripts

```
USE project;
--create and load lookup table Station_Geo_Map in hive
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 and load lookup table subscribed_users in hive
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 and lookup table song_artist_map in hive
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");
```

```
[maria_dev@sandbox-hdp scripts]$ hive -hiveconf batchid=$batchid -f /home/maria_dev/project/scripts/load_lookupdata_hive.hql
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.4.0-91/0/hive-log4j.properties
OK
Time taken: 9.639 seconds
OK
Time taken: 7.884 seconds
OK
Time taken: 1.848 seconds
OK
Time taken: 2.831 seconds
[maria_dev@sandbox-hdp scripts]$
```

Verify data of the lookup tables in hive

```
hive> use project;
OK
Time taken: 6.519 seconds
hive> show tables;
OK
formatted_data
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.965 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: 2.545 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: 0.855 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: 1.041 seconds, Fetched: 15 row(s)
hive>
```

Now, we will perform the data enrichment rules.

Batch script apply_data_enrichmemnt.sh to trigger the data_enrichment.hql file

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

batchid=`cat /home/maria_dev/project/logs/currentbatchno.txt`
LOGFILE=/home/maria_dev/project/logs/log_batch_$batchid


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

hive -hiveconf batchid=$batchid -f /home/maria_dev/project/scripts/data_enrichment.hql
```

Hive Script

Here, in the data enrichment script we are first applying the data enrichment rules and storing into a table. Then, we are again inserting the data into final enrichment table and we are maintaing a column which says pass or fail for the rows where data is still not correct even after applying data enrichment.

maria_dev@sandbox-hdp:~/project/scripts

```
--------------------------------------------------------------------------------
-- Created By: Deepak Ray
-- Date: 18/06/2018
-- Project: Acadgild Music Data Analysis
--------------------------------------------------------------------------------
-- Perform data enrichment on formatted_data based on the following conditions:
-- 1. If like is null put 0
-- 2. If dislike is 0 put 0
-- 3. If artist_id is null fetch it from song_artist_map table based on song_id
-- 4. If geo_cd is null fetch it from station_geo_map table based on station_id
--------------------------------------------------------------------------------

SET hive.auto.convert.join=false;
SET hive.exec.dynamic.partition.mode=nonstrict;
SET hive.support.sqlll.reserved.keywords=false;

USE project;
------------------------------------------------------------
-- Create a temporary table to apply the enrichment rules
------------------------------------------------------------
CREATE TABLE IF NOT EXISTS enriched_data_temp
(
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)
STORED AS ORC;
```

maria_dev@sandbox-hdp:~/project/scripts

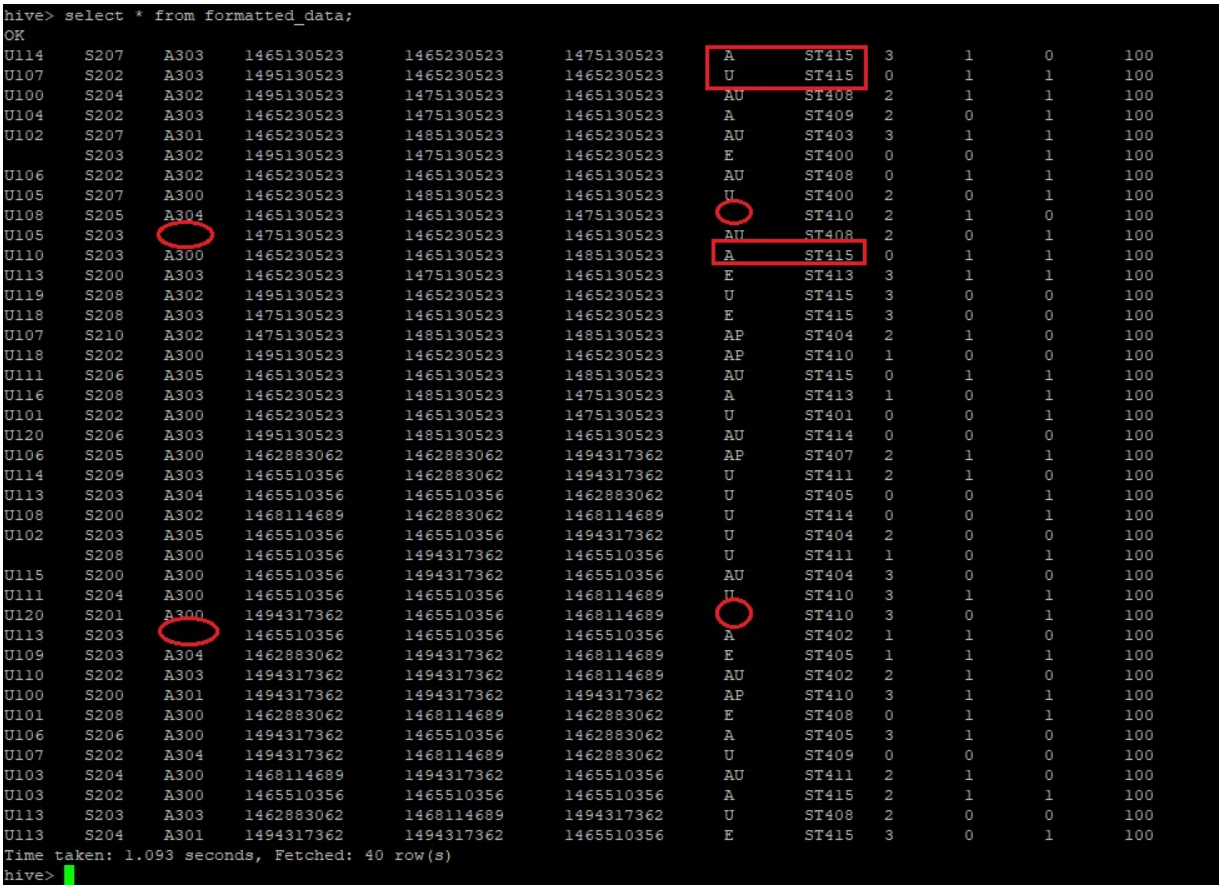
```
INSERT OVERWRITE TABLE enriched_data_temp
PARTITION (batchid)
SELECT
i.user_id,
i.song_id,
sa.artist_id,
i.timestamp,
i.start_ts,
i.end_ts,
sg.geo_cd,
i.station_id,
I.song_end_type,
(CASE WHEN i.like IS NULL THEN 0 ELSE i.like END) AS like,
(CASE WHEN i.dislike IS NULL THEN 0 ELSE i.dislike END) AS dislike,
i.batchid AS status
FROM formatted_data 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};

--------------------------------------------------------------------
-- Create the final enrichment table which will fetch data from
-- the above table and will have a column to mark if a row of
-- data is passed or failed.
--------------------------------------------------------------------
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,
i.artist_id,
i.timestamp,
i.start_ts,
i.end_ts,
i.geo_cd,
i.station_id,
i.song_end_type,
i.like,
i.dislike,
i.batchid,
(CASE WHEN(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 i.artist_id IS NULL
OR i.artist_id='' THEN 'fail' ELSE 'pass' END) AS status
FROM enriched_data_temp i
WHERE i.batchid=${hiveconf:batchid};
```

Execute the script to perform data enrichment.

Verify the data.

Before data enrichment.



```
hive> select * from formatted_data;
OK
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       100
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       100
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       100
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       100
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       100
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       100
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       100
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       100
U108    S205    A304    1465130523      1465130523      1475130523              ST410   2       1       0       100
U105    S203            1475130523      1465230523      1465130523      AU      ST408   2       0       1       100
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       100
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       100
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       100
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       100
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       100
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       100
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       100
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       100
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       100
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       100
U106    S205    A300    1462883062      1462883062      1494317362      AP      ST407   2       1       1       100
U114    S209    A303    1465510356      1462883062      1494317362      U       ST411   2       1       0       100
U113    S203    A304    1465510356      1465510356      1462883062      U       ST405   0       0       1       100
U108    S200    A302    1468114689      1462883062      1468114689      U       ST414   0       0       1       100
U102    S203    A305    1465510356      1465510356      1494317362      U       ST404   2       0       0       100
        S208    A300    1465510356      1494317362      1465510356      U       ST411   1       0       1       100
U115    S200    A300    1465510356      1494317362      1465510356      AU      ST404   3       0       0       100
U111    S204    A300    1465510356      1465510356      1468114689      U       ST410   3       1       1       100
U120    S201    A300    1494317362      1465510356      1468114689              ST410   3       0       1       100
U113    S203            1465510356      1465510356      1465510356      A       ST402   1       1       0       100
U109    S203    A304    1462883062      1494317362      1468114689      E       ST405   1       1       1       100
U110    S202    A303    1494317362      1494317362      1468114689      AU      ST402   2       1       0       100
U100    S200    A301    1494317362      1494317362      1494317362      AP      ST410   3       1       1       100
U101    S208    A300    1462883062      1468114689      1462883062      E       ST408   0       1       1       100
U106    S206    A300    1494317362      1465510356      1462883062      A       ST405   3       1       0       100
U107    S202    A304    1494317362      1468114689      1462883062      U       ST409   0       0       0       100
U103    S204    A300    1468114689      1494317362      1465510356      AU      ST411   2       1       0       100
U103    S202    A300    1465510356      1465510356      1465510356      A       ST415   2       1       1       100
U113    S203    A303    1462883062      1468114689      1494317362      U       ST408   2       0       0       100
U113    S204    A301    1494317362      1494317362      1465510356      E       ST415   3       0       1       100
Time taken: 1.093 seconds, Fetched: 40 row(s)
hive>
```

We can see that some artist_id and geo_cd are null. Also, the data enrichment script will correct the incorrect mappings of artist id and geo cd based on song id and station id respectively.

Also, it has corrected

After data enrichment.

Final, data enrichment table

```
hive> select * from enriched_data;
OK
U100    S200    A300    1494317362      1494317362      1494317362      A       ST410   3       1       1       100     fail
U113    S200    A300    1465230523      1475130523      1465130523      J       ST413   3       1       1       100     fail
U107    S202    A302    1495130523      1465230523      1465230523      NULL    ST415   0       1       1       100     fail
U103    S202    A302    1465510356      1465510356      1465510356      NULL    ST415   2       1       1       100     fail
U106    S202    A302    1465230523      1465130523      1465130523      E       ST408   0       1       1       100     fail
U109    S203    A303    1462883062      1494317362      1468114689      A       ST405   1       1       1       100     fail
        S203    A303    1495130523      1475130523      1465230523      A       ST400   0       0       1       100     fail
U110    S203    A303    1465230523      1465130523      1485130523      NULL    ST415   0       1       1       100     fail
U111    S204    A304    1465510356      1465510356      1468114689      A       ST410   3       1       1       100     fail
U100    S204    A304    1495130523      1475130523      1465130523      E       ST408   2       1       1       100     fail
U113    S204    A304    1494317362      1494317362      1465510356      NULL    ST415   3       0       1       100     fail
U106    S205    A301    1462883062      1462883062      1494317362      AP      ST407   2       1       1       100     fail
U111    S206    A302    1465130523      1465130523      1485130523      NULL    ST415   0       1       1       100     fail
U114    S207    A303    1465130523      1465230523      1475130523      NULL    ST415   3       1       0       100     fail
U102    S207    A303    1465230523      1485130523      1465230523      J       ST403   3       1       1       100     fail
U118    S208    A304    1475130523      1465130523      1465230523      NULL    ST415   3       0       0       100     fail
U119    S208    A304    1495130523      1465230523      1465230523      NULL    ST415   3       0       0       100     fail
U101    S208    A304    1462883062      1468114689      1462883062      E       ST408   0       1       1       100     fail
        S208    A304    1465510356      1494317362      1465510356      A       ST411   1       0       1       100     fail
U107    S210    NULL    1475130523      1485130523      1485130523      E       ST404   2       1       0       100     fail
U108    S200    A300    1468114689      1462883062      1468114689      E       ST414   0       0       1       100     pass
U115    S200    A300    1465510356      1494317362      1465510356      E       ST404   3       0       0       100     pass
U120    S201    A301    1494317362      1465510356      1468114689      A       ST410   3       0       1       100     pass
U107    S202    A302    1494317362      1468114689      1462883062      E       ST409   0       0       0       100     pass
U104    S202    A302    1465230523      1475130523      1465130523      E       ST409   2       0       1       100     pass
U110    S202    A302    1494317362      1494317362      1468114689      AP      ST402   2       1       0       100     pass
U118    S202    A302    1495130523      1465230523      1465230523      A       ST410   1       0       0       100     pass
U101    S202    A302    1465230523      1465130523      1475130523      AU      ST401   0       0       1       100     pass
U113    S203    A303    1465510356      1465510356      1462883062      A       ST405   0       0       1       100     pass
U105    S203    A303    1475130523      1465230523      1465130523      E       ST408   2       0       1       100     pass
U113    S203    A303    1462883062      1468114689      1494317362      E       ST408   2       0       0       100     pass
U102    S203    A303    1465510356      1465510356      1494317362      E       ST404   2       0       0       100     pass
U113    S203    A303    1465510356      1465510356      1465510356      AP      ST402   1       1       0       100     pass
U103    S204    A304    1468114689      1494317362      1465510356      A       ST411   2       1       0       100     pass
U108    S205    A301    1465130523      1465130523      1475130523      A       ST410   2       1       0       100     pass
U106    S206    A302    1494317362      1465510356      1462883062      A       ST405   3       1       0       100     pass
U120    S206    A302    1495130523      1485130523      1465130523      E       ST414   0       0       0       100     pass
U105    S207    A303    1465230523      1485130523      1465130523      A       ST400   2       0       1       100     pass
U116    S208    A304    1465230523      1485130523      1475130523      J       ST413   1       0       1       100     pass
U114    S209    A305    1465510356      1462883062      1494317362      A       ST411   2       1       0       100     pass
Time taken: 0.304 seconds, Fetched: 40 row(s)
```

Step 6 – Data Analysis (data_analysis.sh)

So, now we are all set to perform data analysis.

We will perform data analysis on spark.

Shell script to invoke the spark script.

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

#Get the batch id for logging purpose
batchid=`cat /home/maria_dev/project/logs/currentbatchno.txt`
LOGFILE=/home/maria_dev/project/logs/log_batch_$batchid

echo "Running spark script for data analysis..." >> $LOGFILE

#Add Hbase Classpath
hbase_path=`hbase classpath`

#Remove, if any directory already exists for output
hadoop fs -rm -r /maria_dev/project/batch$batchid

#invoke the spark shell script
spark-shell -i /home/maria_dev/project/scripts/data_analysis.scala --conf spark.driver.args=$batchid --jars /usr/hdp/2.6.4.0-91/hbase/lib/hive-hbase-handler-1.
2.1000.2.6.4.0-91.jar,$hbase_path

echo "Incrementing batchid for the next run..." >> $LOGFILE

#Finally, increment the batch id, once process is over
batchid=`expr $batchid + 1`
echo -n $batchid > /home/maria_dev/project/logs/currentbatchno.txt
```

Spark Script for Data Analysis:

maria_dev@sandbox-hdp:~/project/scripts

```
/*********************************************/
/* Author: Deepak Ray                        */
/* Date: 18/06/2018                          */
/* Project: Acadgild Music Data Analysis     */
/*********************************************/
/* Data Analysis                             */
/*********************************************/

//import the required packages
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="/maria_dev/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+"'grou
p by station_id")
df1.repartition(1).write.option("header","true").csv(outputdirpath+"/top10stations")

/*************************** Problem 1 - End ***********************/
```

```
/*************************** 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 'UNSUBSCRI
BED' 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(AB
S(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 D
ECIMAL(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)
```

Output:

Once the script is executed, all the output will be written to /maria_dev/project/batch100 (this will vary based on batch id)

```
maria_dev@sandbox-hdp:~/project/scripts
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project
Found 1 items
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100
[maria_dev@sandbox-hdp scripts]$
```

Inside this we will have five folder, one for each analysis question.

```
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project/batch100
Found 5 items
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100/top10stations
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100/top_10_connected_artists
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100/top_10_songs
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:08 /maria_dev/project/batch100/top_10_unsubscribed_songs
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100/total_songs_played_byeach_usertype
[maria_dev@sandbox-hdp scripts]$
```

Within each of this folder, there will be csv file holding the results.

```
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project/batch100/top10stations
Found 2 items
-rw-r--r--   1 maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100/top10stations/_SUCCESS
-rw-r--r--   1 maria_dev hdfs        169 2018-06-20 20:07 /maria_dev/project/batch100/top10stations/part-00000-5a118d5e-6d39-4016-9222-6f3ff043ce42-c000.csv
[maria_dev@sandbox-hdp scripts]$ hadoop fs -cat /maria_dev/project/batch100/top10stations/part-00000-5a118d5e-6d39-4016-9222-6f3ff043ce42-c000.csv
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
ST410,3,3
ST411,2,2
ST401,1,1
ST408,1,2
ST413,1,1
[maria_dev@sandbox-hdp scripts]$
```

Step 7 – Wrapping All Scripts to Once (projectexecutor.sh)

Now, we will wrap all the above scripts to one projectwrapper.sh, so that it can be scheduled.

maria_dev@sandbox-hdp:~/project/scripts

```
#!/bin/bash

scriptpath=/home/maria_dev/project/scripts

#Generate web input files
python $scriptpath/generate_web_data.py

#Generate mobile input files
python $scriptpath/generate_mob_data.py

#Call script to format the data
sh $scriptpath/formatdata.sh

#Load formatted data to hive tables
sh $scriptpath/loadformatteddata.sh

#Load HBase tables to Hive Tables
sh $scriptpath/data_enrichment_lookup.sh

#Apply Validation and Data Enrichment Rules
sh $scriptpath/apply_data_enrichment.sh

#Perform Data Analysis
sh $scriptpath/data_analysis.sh
~
~
```

With every run a new log file will get generated and new output folder will get generated.

```
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project/
Found 5 items
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 20:07 /maria_dev/project/batch100
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 21:54 /maria_dev/project/batch101
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 22:41 /maria_dev/project/batch102
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:06 /maria_dev/project/batch103
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104
[maria_dev@sandbox-hdp scripts]$ hadoop fs -ls /maria_dev/project/batch104
Found 5 items
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104/top10stations
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104/top_10_connected_artists
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104/top_10_songs
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104/top_10_unsubscribed_songs
drwxr-xr-x   - maria_dev hdfs          0 2018-06-20 23:13 /maria_dev/project/batch104/total_songs_played_byeach_usertype
[maria_dev@sandbox-hdp scripts]$
```

Step 8 – Scheduling the Script

```
[maria_dev@sandbox-hdp scripts]$ sudo crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
```

maria_dev@sandbox-hdp:~/project/scripts

```
*/5 * * * * /home/maria_dev/project/scripts/projectexecutor.sh
~
~
~
~
~
```

maria_dev@sandbox-hdp:~/project/scripts

```
[maria_dev@sandbox-hdp logs]$ cd ..
[maria_dev@sandbox-hdp project]$ cd scripts/
[maria_dev@sandbox-hdp scripts]$ ./projectwrapper.sh
```