

Project – Music Data Analysis

Section 1: Project Overview

A leading music-catering company is planning to analyze large amount of data received from varieties of sources, namely mobile app and website to track the behavior 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.

1.1 Fields present in the data files

Data files contain below fields.

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

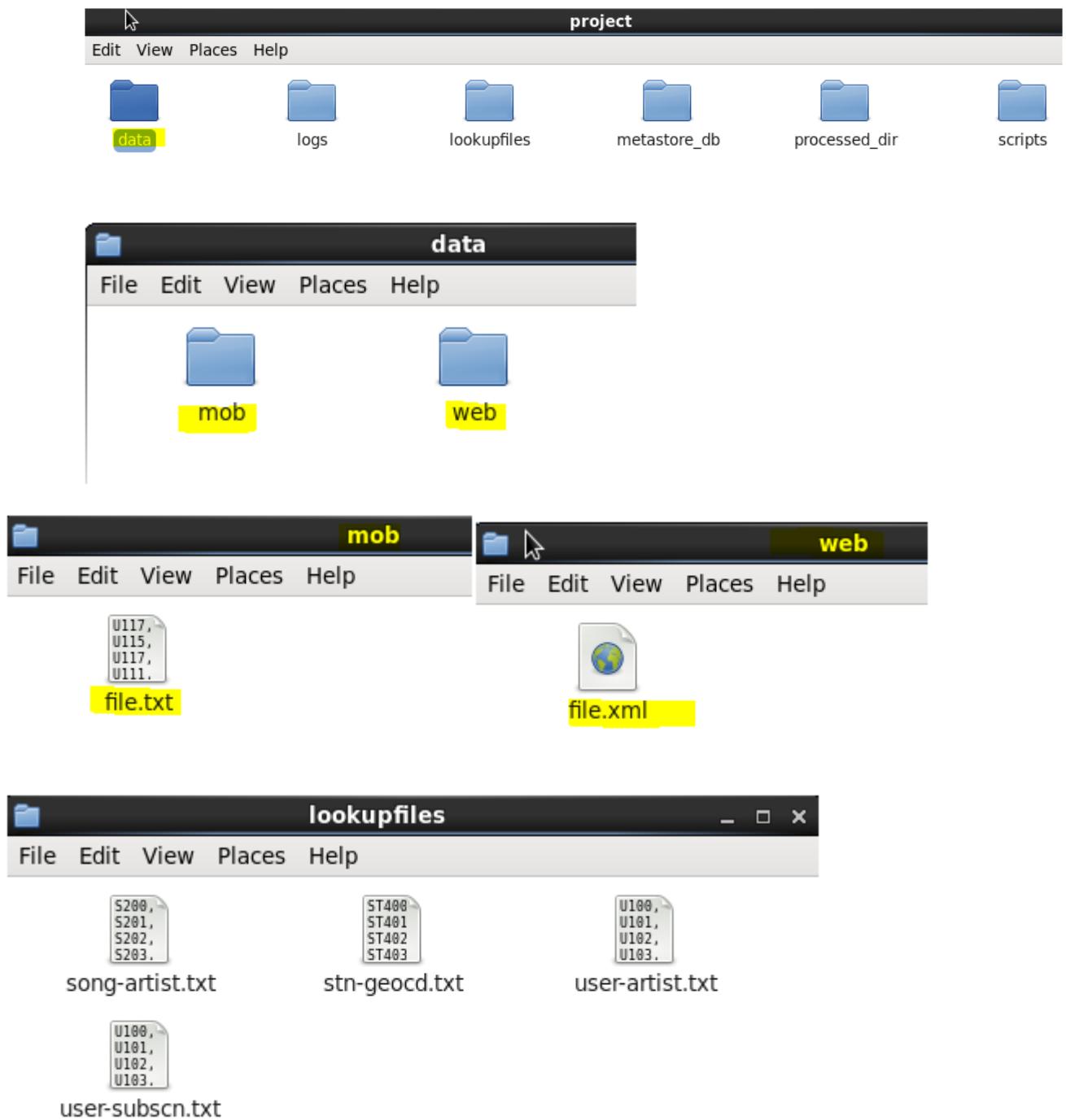
1.2 LookUp Tables

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

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

1.3 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.



1.4 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

1.5 Data Analysis

It is not only the data which is important, rather it is the insight it can be used to generate important. Once we have made the data ready for analysis, we have to perform below analysis on a daily basis.

1. Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.
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.
3. Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.
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.
5. Determine top 10 unsubscribed users who listened to the songs for the longest duration.

Section 2: Data Creation:

We have used the python script to generate the data. There are two script files one is for web data and another is for Mob data.

generate_web_data.py -- Generates some random data coming from web application

python /home/acadgild/project/scripts/generate_web_data.py

```
[acadgild@localhost scripts]$ cat generate_web_data.py
>>> from random import randint
>>> from random import choice
>>>
file = open("/home/acadgild/project/data/web/file.xml", "w")
count = 20
file.write("<records>\n")
while (count > 0):
    geo_cd_list=[ "A", "E", "AU", "AP", "U"]
    song_end_type_list=[ "0", "1", "2", "3" ]
    timestamp_list=[ "2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22" ]
    start_ts_list=[ "2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22" ]
    end_ts_list=[ "2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22" ]
    if (count%15 == 0):
        user_id = ""
    else:
        user_id = "U" + str(randint(100,120))
    song_id = "S" + str(randint(200,210))
    if (count%11 == 0):
        artist_id = ""
    else:
        artist_id = "A" + str(randint(300,305))
    timestamp = choice(timestamp_list)
    start_ts = choice(start_ts_list)
    end_ts = choice(end_ts_list)
    if (count%12 == 0):
        geo_cd = ""
    else:
        geo_cd = choice(geo_cd_list)
    station_id = "ST" + str(randint(400,415))
    file.write("    <record>\n        <station_id>" + station_id + "</station_id>\n        <song_id>" + song_id + "</song_id>\n        <artist_id>" + artist_id + "</artist_id>\n        <timestamp>" + timestamp + "</timestamp>\n        <start_ts>" + start_ts + "</start_ts>\n        <end_ts>" + end_ts + "</end_ts>\n        <geo_cd>" + geo_cd + "</geo_cd>\n    </record>\n")
    count = count - 1
file.close()
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2.127.0.0.1(acadgild) Xserver Exit
»
if (count%ll == 0):
    artist_id = ""
else:
    artist_id = "A" + str(randint(300,305))

timestamp = choice(timestamp_list)
start_ts = choice(start_ts_list)
end_ts = choice(end_ts_list)

if (count%l2 == 0):
    geo_cd = ""
else:
    geo_cd = choice(geo_cd_list)

station_id = "S" + str(randint(400,415))
song_end_type = choice(song_end_type_list)
like = str(randint(0,1))
dislike = str(randint(0,1))
file.write("<record>\n")
file.write("<user_id>%s</user_id>\n" % (user_id))
file.write("<song_id>%s</song_id>\n" % (song_id))
file.write("<artist_id>%s</artist_id>\n" % (artist_id))
file.write("<timestamp>%s</timestamp>\n" % (timestamp))
file.write("<start_ts>%s</start_ts>\n" % (start_ts))
file.write("<end_ts>%s</end_ts>\n" % (end_ts))
file.write("<geo_cd>%s</geo_cd>\n" % (geo_cd))
file.write("<station_id>%s</station_id>\n" % (station_id))
file.write("<song_end_type>%s</song_end_type>\n" % (song_end_type))
file.write("<like>%s</like>\n" % (like))
file.write("<dislike>%s</dislike>\n" % (dislike))
file.write("</record>\n")

count = count-1
file.write("</records>")
file.close()
[acadgild@localhost scripts]$ 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

generate_mob_data.py -- Generates some random data coming from mobile application
python /home/acadgild/project/scripts/generate_mob_data.py

```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2.127.0.0.1(acadgild) Xserver Exit
»
[acadgild@localhost scripts]$ cat generate_mob_data.py
from random import randint
from random import choice

file = open("/home/acadgild/project/data/mob/file.txt", "w")
count = 20

while (count > 0):
    geo_cd_list=["A", "E", "AU", "AP", "U"]
    song_end_type_list=[0,1,2,3]
    timestamp_list=[1465230523, "1465130523", "1475130523", "1495130523"]
    start_ts_list=[1465230523, "1465130523", "1475130523", "1485130523"]
    end_ts_list=[1465230523, "1465130523", "1475130523", "1485130523"]

    if (count%15 == 0):
        user_id = ""
    else:
        user_id = "U" + str(randint(100,120))

    song_id = "S" + str(randint(200,210))

    if (count%ll == 0):
        artist_id = ""
    else:
        artist_id = "A" + str(randint(300,305))

    timestamp = choice(timestamp_list)
    start_ts = choice(start_ts_list)
    end_ts = choice(end_ts_list)

    if (count%l2 == 0):
        geo_cd = ""
    else:
        geo_cd = choice(geo_cd_list)

    station_id = "S" + str(randint(400,415))
    song_end_type = choice(song_end_type_list)
    like = str(randint(0,1))
    dislike = str(randint(0,1))
    file.write("<record>\n")
    file.write("<user_id>%s</user_id>\n" % (user_id))
    file.write("<song_id>%s</song_id>\n" % (song_id))
    file.write("<artist_id>%s</artist_id>\n" % (artist_id))
    file.write("<timestamp>%s</timestamp>\n" % (timestamp))
    file.write("<start_ts>%s</start_ts>\n" % (start_ts))
    file.write("<end_ts>%s</end_ts>\n" % (end_ts))
    file.write("<geo_cd>%s</geo_cd>\n" % (geo_cd))
    file.write("<station_id>%s</station_id>\n" % (station_id))
    file.write("<song_end_type>%s</song_end_type>\n" % (song_end_type))
    file.write("<like>%s</like>\n" % (like))
    file.write("<dislike>%s</dislike>\n" % (dislike))
    file.write("</record>\n")

    count = count-1
file.write("</records>")
file.close()
[acadgild@localhost scripts]$ 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```
127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2.127.0.0.1(acadgild) X Xserver Exit
» start_ts_list=[“1465230523”, “1465130523”, “1475130523”, “1485130523”]
end_ts_list=[“1465230523”, “1465130523”, “1475130523”, “1485130523”]
if (count%15 == 0):
    user_id = “”
else:
    user_id = “U” + str(randint(100,120))
song_id = “S” + str(randint(200,210))
if (count%11 == 0):
    artist_id = “”
else:
    artist_id = “A” + str(randint(300,305))
timestamp = choice(timestamp_list)
start_ts = choice(start_ts_list)
end_ts = choice(end_ts_list)
if (count%12 == 0):
    geo_cd = “”
else:
    geo_cd = choice(geo_cd_list)
station_id = “ST” + str(randint(400,415))
song_end_type = choice(song_end_type_list)
like = str(randint(0,1))
dislike = str(randint(0,1))
file.write(“%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\n” % (user_id, song_id, artist_id, timestamp, start_ts, end_ts, geo_cd, station_id, song_end_type, like, dislike))
)
count = count-1
file.close()
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```

UNREGISTERED VERSION - Please support Mobaxterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Windows Start Type here to search Taskbar 18:37 ENG 24-01-2019

Section 3: Lookup table and Intial validation

Start the Daemons

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

```

#!/bin/bash

if [ -f "/home/acadgild/Project_2_Music_Data_Analysis/logs/current-batch.txt" ]
then
echo "Batch File Found!"

else
echo -n "1" > "/home/acadgild/Project_2_Music_Data_Analysis/logs/current-batch.txt"
fi

chmod 775 /home/acadgild/Project_2_Music_Data_Analysis/logs/current-batch.txt
batchid=`cat /home/acadgild/Project_2_Music_Data_Analysis/logs/current-batch.txt`
LOGFILE=/home/acadgild/Project_2_Music_Data_Analysis/logs/log_batch_$batchid

echo "Starting daemons" >> $LOGFILE

# To Start Hadoop Daemons:
start-all.sh

# To start the HMASTER service:
start-hbase.sh

# To Start the JobHistory server Services:
mr-jobhistory-daemon.sh start historyserver

# To Start the mysql service
sudo service mysqld start

# To Start HIVE metastore:
hive --service metastore

```

The screenshot shows a terminal window titled '127.0.0.1 (acadgild)' in MobaXterm. The session is active and displays the following terminal session:

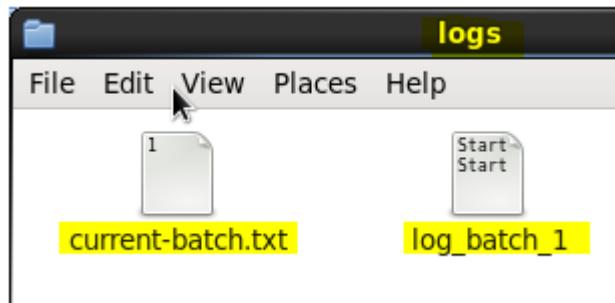
```

[acadgild@localhost ~]$ python /home/acadgild/project/scripts/generate_mob_data.py
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
[acadgild@localhost ~]$
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/start-daemons.sh
Batch script found!
★ This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
19/01/20 04:23:58 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-namenode-localhost.localdomain.out
localhost: starting datanode, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-secondarynamenode-localhost.localdomain.out
19/01/20 04:25:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-nodemanager-localhost.localdomain.out
localhost: starting zookeeper, logging to /home/acadgild/install/hbase/hbase-1.2.6/logs/hbase-acadgild-zookeeper-localhost.localdomain.out
starting master, logging to /home/acadgild/install/hbase/hbase-1.2.6/logs/hbase-acadgild-master-localhost.localdomain.out
starting regionserver, logging to /home/acadgild/install/hbase/hbase-1.2.6/logs/hbase-acadgild-1-regionserver-localhost.localdomain.out
starting historyserver, logging to /home/acadgild/install/hadoop/hadoop-2.6.5/logs/mapred-acadgild-historyserver-localhost.localdomain.out
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ jps
[acadgild@localhost ~]$

```

At the bottom of the terminal window, there is a status bar with the text 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

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



Use the "**populate-lookup.sh**" script to create lookup tables in **Hbase**. These tables have to be used in,

- Data formatting,
- Data enrichment and
- Analysis stage

Lookup Tables

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

"**populate-lookup.sh**" script

The "**populate-lookup.sh**" shell script creates the above 4 lookup tables in the Hbase and populate the data into the lookup tables from the dataset files.

In the below screen shots, we can see the create-lookup.sh scripts and the following screen shots shows the tables creation and population of the data in the Hbase. Also, the values loaded into the Hbase Tables are also shown, please see the below screen shots.

```

1  #!/bin/bash
2
3  batchid=`cat /home/acadgild/project/logs/current-batch.txt`
4
5  LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
6
7  echo "Creating LookUp Tables" >> $LOGFILE
8
9  echo "create 'station-geo-map', 'geo'" | hbase shell
10 echo "create 'subscribed-users', 'subscn'" | hbase shell
11 echo "create 'song-artist-map', 'artist'" | hbase shell
12
13
14 echo "Populating LookUp Tables" >> $LOGFILE
15
16 file="/home/acadgild/project/lookupfiles/stn-geocd.txt"
17 while IFS= read -r line
18 do
19   stnid=`echo $line | cut -d',' -f1`
20   geocd=`echo $line | cut -d',' -f2`
21   echo "put 'station-geo-map', '$stnid', 'geo:geo_cd', '$geocd'" | hbase shell
22 done <"$file"
23
24
25 file="/home/acadgild/project/lookupfiles/song-artist.txt"
26 while IFS= read -r line
27 do
28   songid=`echo $line | cut -d',' -f1`
29   artistid=`echo $line | cut -d',' -f2`
30   echo "put 'song-artist-map', '$songid', 'artist:artistid', '$artistid'" | hbase shell
31 done <"$file"
32
33
34 file="/home/acadgild/project/lookupfiles/user-subscn.txt"
35 while IFS= read -r line
36 do
37   userid=`echo $line | cut -d',' -f1`
38   startdt=`echo $line | cut -d',' -f2`
39   enddt=`echo $line | cut -d',' -f3`
40   echo "put 'subscribed-users', '$userid', 'subscn:startdt', '$startdt'" | hbase shell
41   echo "put 'subscribed-users', '$userid', 'subscn:enddt', '$enddt'" | hbase shell
42 done <"$file"
43
44 hive -f /home/acadgild/project/scripts/user-artist.hql
45

```

```

[acadgild@localhost ~]$ ls sh /home/acadgild/project/scripts/populate-lookup.sh
2019-01-20 04:35:12.203 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help->' for list of supported commands.
Type "exit->" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

create 'station-geo-map', 'geo'
0 row(s) in 5.4280 seconds

Hbase::Table - station-geo-map
2019-01-20 04:35:49.776 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help->' for list of supported commands.
Type "exit->" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

create 'subscribed-users', 'subscrn'
0 row(s) in 2.9770 seconds

Hbase::Table - subscribed-users
2019-01-20 04:36:05.057 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

```

[acadgild@localhost ~]$ ls sh /home/acadgild/project/scripts/populate-lookup.sh
2019-01-20 04:36:05.057 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help->' for list of supported commands.
Type "exit->" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

create 'song-artist-map', 'artist'
0 row(s) in 2.8600 seconds

Hbase::Table - song-artist-map
2019-01-20 04:36:56.065 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help->' for list of supported commands.
Type "exit->" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

put 'station-geo-map', 'ST400', 'geo:geo_cd', 'A'
0 row(s) in 2.5520 seconds

2019-01-20 04:37:29.707 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple SLF4J bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help->' for list of supported commands.
Type "exit->" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

put 'station-geo-map', 'ST401', 'geo:geo_cd', 'AU'
0 row(s) in 1.7780 seconds

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

We can see the lookup tables created using the “*populate-lookup.sh*” in the below screen shot, Lookup Tables in the hbase shell,

```

[acadgild@localhost ~]$ hbase shell
2019-01-20 05:13:26,636 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple [jar:] bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
★ HBase Shell; enter 'help->' for list of supported commands.
Type "exit-RETURN" to leave the HBase Shell
Version 1.2.6, runknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> list
TABLE
song-artist-map
station-geo-map
subscribed-users
3 row(s) in 1.3840 seconds
=> ["song-artist-map", "station-geo-map", "subscribed-users"]
hbase(main):002:0> 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

The values loaded in the Lookup tables are shown below,

```

[acadgild@localhost ~]$ hbase shell
2019-01-20 05:13:26,636 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
SLF4J: Class path contains multiple [jar:] bindings
SLF4J: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
★ HBase Shell; enter 'help->' 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

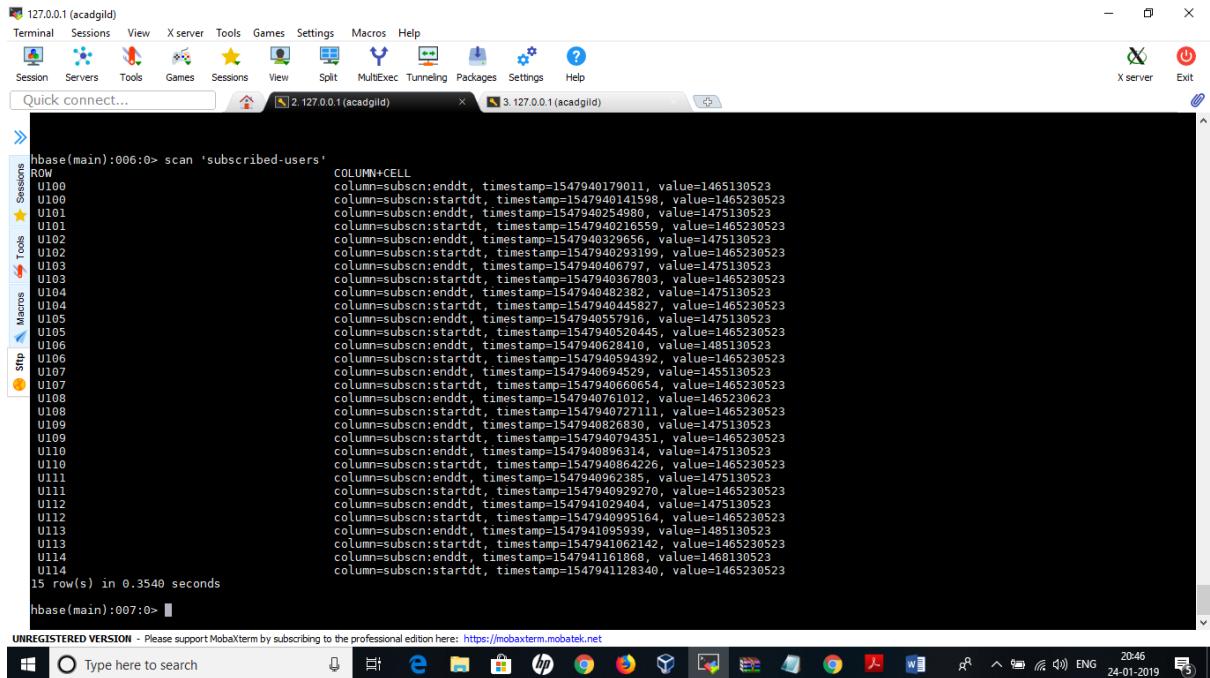
hbase(main):001:0> scan 'subscribed-users'
3 row(s) in 1.3840 seconds
=> ["song-artist-map", "station-geo-map", "subscribed-users"]

hbase(main):002:0> scan 'song-artist-map'
ROW
S200          COLUMN+CELL
S201          column=artist:artistid, timestamp=1547939761164, value=A300
S202          column=artist:artistid, timestamp=1547939706480, value=A301
S203          column=artist:artistid, timestamp=1547939834674, value=A302
S204          column=artist:artistid, timestamp=1547939872008, value=A303
S205          column=artist:artistid, timestamp=1547939909414, value=A304
S206          column=artist:artistid, timestamp=1547939948009, value=A301
S207          column=artist:artistid, timestamp=1547939985638, value=A302
S208          column=artist:artistid, timestamp=1547940024549, value=A303
S209          column=artist:artistid, timestamp=1547940063597, value=A304
S209          column=artist:artistid, timestamp=1547940101537, value=A305
10 row(s) in 0.7900 seconds

hbase(main):003:0> scan 'station-geo-map'
ROW
ST400          COLUMN+CELL
ST401          column=geo:geo_cd, timestamp=1547939226544, value=A
ST402          column=geo:geo_cd, timestamp=1547939258381, value=AU
ST403          column=geo:geo_cd, timestamp=1547939325921, value=JP
ST404          column=geo:geo_cd, timestamp=154793932411, value=US
ST405          column=geo:geo_cd, timestamp=1547939362028, value=E
ST406          column=geo:geo_cd, timestamp=15479393626263, value=A
ST407          column=geo:geo_cd, timestamp=1547939429819, value=AU
ST408          column=geo:geo_cd, timestamp=1547939465291, value=AP
ST409          column=geo:geo_cd, timestamp=1547939501327, value=F
ST410          column=geo:geo_cd, timestamp=1547939536453, value=E
ST411          column=geo:geo_cd, timestamp=1547939573723, value=A
ST412          column=geo:geo_cd, timestamp=1547939610994, value=A
ST413          column=geo:geo_cd, timestamp=1547939650869, value=AP
ST414          column=geo:geo_cd, timestamp=1547939687475, value=J
ST414          column=geo:geo_cd, timestamp=1547939724352, value=E
15 row(s) in 0.2450 seconds


```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 127.0.0.1(acadgild) x 3. 127.0.0.1(acadgild) ↻

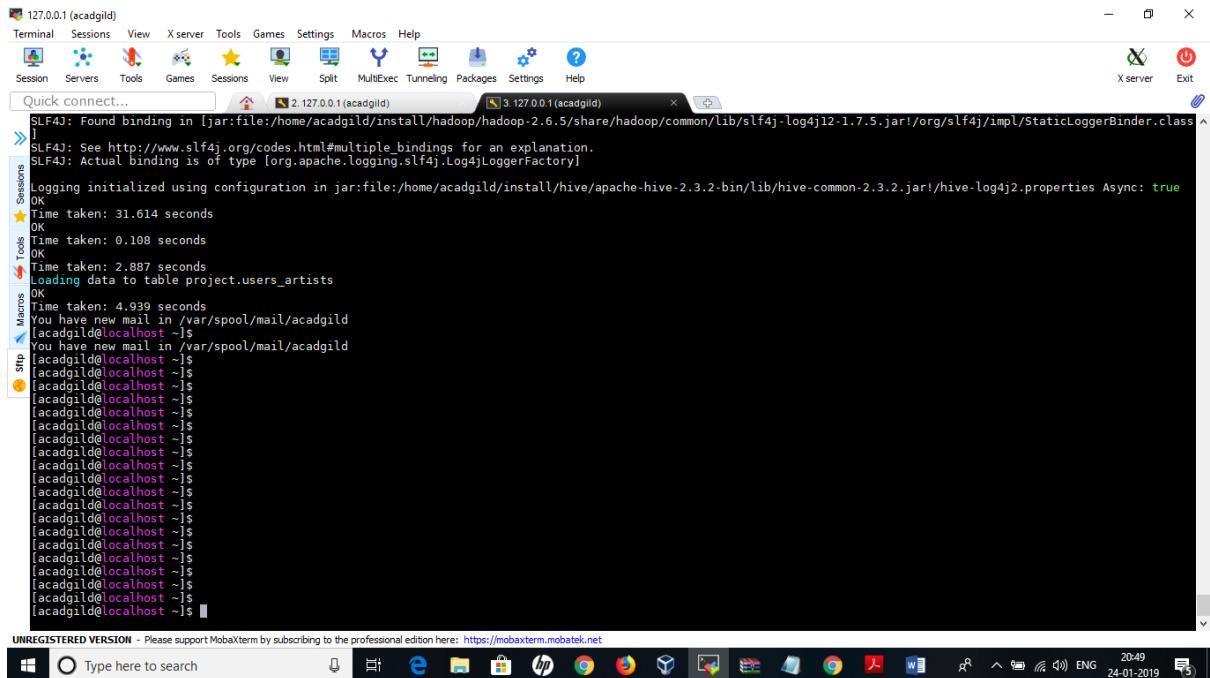
> hbase(main):006:0> scan 'subscribed-users'
ROW                                     COLUMN+CELL
U100                                    column=subscn:enddt, timestamp=1547940179011, value=1465130523
U100                                    column=subscn:startdt, timestamp=1547940141598, value=1465230523
U101                                    column=subscn:enddt, timestamp=1547940254980, value=1475130523
U101                                    column=subscn:startdt, timestamp=1547940216559, value=1465230523
U102                                    column=subscn:enddt, timestamp=1547940329656, value=1475130523
U102                                    column=subscn:startdt, timestamp=1547940293199, value=1465230523
U103                                    column=subscn:enddt, timestamp=1547940406797, value=1475130523
U103                                    column=subscn:startdt, timestamp=1547940367803, value=1465230523
U104                                    column=subscn:enddt, timestamp=1547940482382, value=1475130523
U104                                    column=subscn:startdt, timestamp=1547940482387, value=1465230523
U105                                    column=subscn:enddt, timestamp=1547940557916, value=1465230523
U105                                    column=subscn:startdt, timestamp=1547940520445, value=1465230523
U106                                    column=subscn:enddt, timestamp=1547940628410, value=1485130523
U106                                    column=subscn:startdt, timestamp=1547940594392, value=1465230523
U107                                    column=subscn:enddt, timestamp=1547940694523, value=1455130523
U107                                    column=subscn:startdt, timestamp=1547940660654, value=1465230523
U108                                    column=subscn:enddt, timestamp=1547940761012, value=1465230623
U108                                    column=subscn:startdt, timestamp=1547940727111, value=1465230523
U109                                    column=subscn:enddt, timestamp=1547940826833, value=1475130523
U109                                    column=subscn:startdt, timestamp=1547940794351, value=1465230523
U110                                    column=subscn:enddt, timestamp=1547940896314, value=1475130523
U110                                    column=subscn:startdt, timestamp=1547940864226, value=1465230523
U111                                    column=subscn:enddt, timestamp=1547940962385, value=1475130523
U111                                    column=subscn:startdt, timestamp=1547940929270, value=1465230523
U112                                    column=subscn:enddt, timestamp=1547941029404, value=1475130523
U112                                    column=subscn:startdt, timestamp=15479410995164, value=1465230523
U113                                    column=subscn:enddt, timestamp=1547941095939, value=1485130523
U113                                    column=subscn:startdt, timestamp=15479411062142, value=1465220523
U114                                    column=subscn:enddt, timestamp=15479411161868, value=1468130523
U114                                    column=subscn:startdt, timestamp=1547941128340, value=1465230523
15 row(s) in 0.3540 seconds
hbase(main):007:0>

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

We have successfully created the lookup tables in the Hbase.

The populate-lookup.sh also creates a lookup table “**users_artists**” in the HIVE, loading the data from the **user-artist.txt**, the below screen shot shows that the table has been created in the HIVE.



```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 127.0.0.1(acadgild) x 3. 127.0.0.1(acadgild) ↻

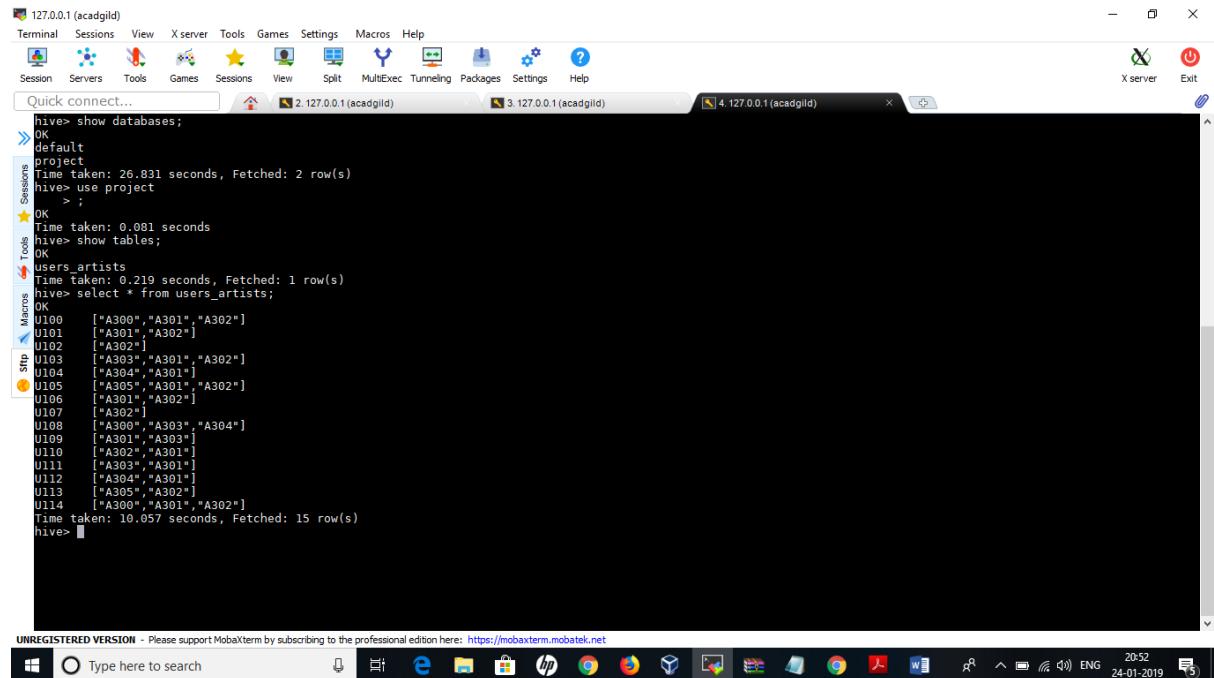
> ! SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!org/slf4j/impl/StaticLoggerBinder.class]
> ! SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
> ! SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

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: 31.614 seconds
OK
Time taken: 0.108 seconds
OK
Time taken: 2.887 seconds
Loading data to table project.users_artists
OK
Time taken: 4.939 seconds
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

```
hive> Select * From users_artists;
```



A screenshot of the MobaXterm terminal window titled "127.0.0.1 (acadgild)". The terminal shows the following Hive session:

```
hive> show databases;
OK
default
Time taken: 26.831 seconds, Fetched: 2 row(s)
hive> use project
OK
hive> select * from users_artists;
OK
Time taken: 0.219 seconds, Fetched: 1 row(s)
hive> select * from users_artists;
OK
U100  ["A300","A301","A302"]
U101  ["A301","A302"]
U102  ["A302"]
U103  ["A303","A301","A302"]
U104  ["A304","A301"]
U105  ["A305","A301","A302"]
U106  ["A301","A302"]
U107  ["A302"]
U108  ["A309","A303","A304"]
U109  ["A301","A303"]
U110  ["A302","A301"]
U111  ["A303","A301"]
U112  ["A304","A301"]
U113  ["A305","A302"]
U114  ["A300","A301","A302"]
Time taken: 10.057 seconds, Fetched: 15 row(s)
hive>
```

The terminal also displays the system status bar at the bottom.

Now we need to link these lookup tables in hive using the Hbase Storage Handler.

With the help of "**data_enrichment_filtering_schema.sh**" file we will create hive tables on the top of Hbase tables using "**create_hive_hbase_lookup.hql**".

Creating Hive Tables on the top of Hbase:

In this section 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.

Run the script: `sh /home/acadgild/project/scripts /data_enrichment_filtering_schema.sh`

The script will run the "**create_hive_hbase_lookup.hql**" which will create the HIVE external tables with the help of **Hbase storage handler & SerDe properties**. The hive external tables will match the columns of **Hbase tables to HIVE tables**.

```
1 #!/bin/bash
2
3 batchid=`cat /home/acadgild/project/logs/current-batch.txt`
4 LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
5
6 echo "Creating hive tables on top of hbase tables for data enrichment and filtering..." >> $LOGFILE
7
8 hive -f /home/acadgild/project/scripts/create_hive_hbase_lookup.hql
9
10
```

create_hive_hbase_lookup.hql

```
1 USE project;
2 create external table if not exists station_geo_map
3 (
4   station_id String,
5   geo_cd string
6 )
7 STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
8 with serdeproperties
9   ("hbase.columns.mapping"":key,geo:geo_cd")
10  tblproperties("hbase.table.name""station-geo-map");
11
12 create external table if not exists subscribed_users
13 (
14   user_id STRING,
15   subscn_start_dt STRING,
16   subscn_end_dt STRING
17 )
18 STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
19 with serdeproperties
20   ("hbase.columns.mapping"":key,subscn:startdt,subscn:enddt")
21  tblproperties("hbase.table.name""subscribed-users");
22
23 create external table if not exists song_artist_map
24 (
25   song_id STRING,
26   artist_id STRING
27 )
28 STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
29 with serdeproperties
30   ("hbase.columns.mapping"":key,artist:artistid")
31  tblproperties("hbase.table.name""song-artist-map");
32
```

The below screenshot we can see tables getting created in hive by running the “**“data_enrichement_filtering_schema.sh file”**

```
[acadgild@localhost scripts]$ ./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/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: 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: 29.198 seconds
OK
Time taken: 10.334 seconds
OK
Time taken: 1.056 seconds
OK
Time taken: 0.957 seconds
you have new mail in /var/spool/mail/acadgild
[acadgild@localhost scripts]$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

hive>Show Tables;

hive>Select * From song_artist_map

hive>Select * From station_geo_map

hive>Select * From Subscribed_users

```
hive> show tables;
» OK
song_artist_map
station_geo_map
subscribed_users
users_artists
Time Taken: 0.142 seconds, Fetched: 4 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: 2.431 seconds, Fetched: 10 row(s)
hive> select * from station_geo_map;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'station_geo_map'
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
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2.127.0.0.1(acadgild) 3.127.0.0.1(acadgild) 4.127.0.0.1(acadgild)
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'station_geo_map'
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 | MP  |
| ST413 | J   |
| ST414 | E   |
+-----+
Time taken: 1.999 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 | 1495130523 |
| U108 | 1465230523 | 1465230623 |
| U109 | 1465230523 | 1475130523 |
| U110 | 1465230523 | 1475130523 |
| U111 | 1465230523 | 1475130523 |
| U112 | 1465230523 | 1475130523 |
| U113 | 1465230523 | 1485130523 |
| U114 | 1465230523 | 1468130523 |
+-----+
Time taken: 1.68 seconds, Fetched: 15 row(s)
hive>

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

Section-4 Data Formatting:

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

Run the script: *sh /home/acadgild/project/scripts/dataformatting.sh*

```

1  #!/bin/bash
2
3  batchid=`cat /home/acadgild/project/logs/current-batch.txt`
4  LOGFILE=/home/acadgild/project/logs/log_batch_${batchid}
5
6  echo "Placing data files from local to HDFS..." >> $LOGFILE
7
8  hadoop fs -rm -r /user/acadgild/project/batch${batchid}/web/
9  hadoop fs -rm -r /user/acadgild/project/batch${batchid}/formattedweb/
10 hadoop fs -rm -r /user/acadgild/project/batch${batchid}/mob/
11
12 hadoop fs -mkdir -p /user/acadgild/project/batch${batchid}/web/
13 hadoop fs -mkdir -p /user/acadgild/project/batch${batchid}/mob/
14
15 hadoop fs -put /home/acadgild/project/data/web/* /user/acadgild/project/batch${batchid}/web/
16 hadoop fs -put /home/acadgild/project/data/mob/* /user/acadgild/project/batch${batchid}/mob/
17
18 echo "Running pig script for data formatting..." >> $LOGFILE
19
20 pig -param batchid=${batchid} /home/acadgild/project/scripts/dataformatting.pig
21
22 echo "Running hive script for formatted data load..." >> $LOGFILE
23
24 hive -hiveconf batchid=${batchid} -f /home/acadgild/project/scripts/formatted_hive_load.hql
25

```

```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
4.127.0.0.1(acadgild) 3.127.0.0.1(acadgild) 4.127.0.0.1(acadgild) 5.127.0.0.1(acadgild)
You have new mail in /var/spool/mail/acadgild
> [acadgild@localhost ~]$ ls
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/dataformatting.sh
19/01/30 21:36:04 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
rm: '/user/acadgild/project/batch1/formattedweb/': No such file or directory
19/01/30 21:36:23 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
19/01/30 21:36:34 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/01/30 21:36:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/01/30 21:37:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/01/30 21:37:20 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
19/01/30 21:37:20 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2019-01-30 21:37:20,544 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2019-01-30 21:37:20,545 [main] INFO org.apache.pig.Main - Logging error messages to: /home/acadgild/pig_1548864440534.log
SLF4J: Class path contains multiple SLF4J bindings
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: Found binding in [jar:file:/home/acadgild/install/hbase/hbase-1.2.6/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2019-01-30 21:37:23,376 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-01-30 21:37:24,595 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/acadgild/.pigbootup not found
2019-01-30 21:37:25,500 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2019-01-30 21:37:25,501 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFs
2019-01-30 21:37:25,501 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:8020
2019-01-30 21:37:29,613 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: Pig-dataformatting.pig-57635712-0ba5-424c-95a9-5576b2b63e3f
2019-01-30 21:37:29,614 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
2019-01-30 21:37:29,563 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFs
2019-01-30 21:37:29,600 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFs
2019-01-30 21:37:37,015 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFs
2019-01-30 21:37:37,062 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutputformat.separator
2019-01-30 21:37:37,826 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

We are running two scripts to format the data. They are:

- Dataformatting.pig
- Formatted_hive_load.hql

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

Dataformatting.pig

```

1 REGISTER /home/acadgild/project/lib/piggybank.jar;
2
3 DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();
4
5 A = LOAD '/user/acadgild/project/batch$batchid/web/' using org.apache.pig.piggybank.storageXMLLoader('record') as (x:chararray);
6
7 B = FOREACH A GENERATE TRIM(XPath(x, 'record/user_id')) AS user_id,
8     TRIM(XPath(x, 'record/song_id')) AS song_id,
9     TRIM(XPath(x, 'record/artist_id')) AS artist_id,
10    ToUnixTimeToDate(TRIM(XPath(x, 'record.timestamp')),'yyyy-MM-dd HH:mm:ss')) AS timestamp,
11    ToUnixTimeToDate(TRIM(XPath(x, 'record/start_ts')),'yyyy-MM-dd HH:mm:ss')) AS start_ts,
12    ToUnixTimeToDate(TRIM(XPath(x, 'record/end_ts')),'yyyy-MM-dd HH:mm:ss')) AS end_ts,
13    TRIM(XPath(x, 'record/geo_cd')) AS geo_cd,
14    TRIM(XPath(x, 'record/station_id')) AS station_id,|
15    TRIM(XPath(x, 'record/song_end_type')) AS song_end_type,
16    TRIM(XPath(x, 'record/like')) AS like,
17    TRIM(XPath(x, 'record/dislike')) AS dislike;
18
19 STORE B INTO '/user/acadgild/project/batch$batchid/formattedweb/' USING PigStorage(',');
20

```

formatted_hive_load.hql

The screenshot shows a Gedit window titled "formatted_hive_load.hql (~/project/scripts) - gedit". The window contains a Hive load script. The script starts with `USE project;` followed by a CREATE TABLE statement for "formatted_input" with columns for User_id, Song_id, Artist_id, u_Timestamp, Start_ts, End_ts, Geo_cd, Station_id, Song_end_type, u_Like, and Dislike. It is PARTITIONED BY batchid and has ROW FORMAT DELIMITED FIELDS TERMINATED BY ','. It then loads data from two paths into the table: '/user/acadgild/project/batch\${hiveconf:batchid}/formattedweb/' and '/user/acadgild/project/batch\${hiveconf:batchid}/mob/'. The script ends with a semi-colon. The status bar at the bottom indicates "Plain Text" and "Ln 1, Col 1". Below the window, there are tabs for "acadgild", "project", "scripts", and "formatted_hive_load.h...".

```

USE project;

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

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

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

```

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

Pig script successful completion,

The screenshot shows a MobaXterm terminal window titled "4. 127.0.0.1 (acadgild)". The terminal output shows a successful execution of a Pig script. It includes logs for org.apache.hadoop.ipc.Client\$Connection.setupConnection, org.apache.hadoop.ipc.Client\$Connection.setupIOstreams, org.apache.hadoop.ipc.Client.access\$2800, org.apache.hadoop.ipc.Client.getConnection, org.apache.hadoop.ipc.Client.call, and org.apache.hadoop.ipc.Client\$Connection\$ClientProtocol\$Procedure. The log also shows the start of a MapReduce job with ID job_1548860494249_0001, running on Hadoop version 2.6.5 and Pig version 0.16.0, with a success message. The output section shows that 0 records were read from the input path and stored in the output path. The counters section shows total records written, total bytes written, and spillable memory manager statistics. The job DAG section shows the job ID. The status bar at the bottom indicates "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net".

```

at org.apache.hadoop.ipc.Client$Connection.setupConnection(Client.java:609)
at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:707)
at org.apache.hadoop.ipc.Client.access$2800(Client.java:370)
at org.apache.hadoop.ipc.Client.getConnection(Client.java:1523)
at org.apache.hadoop.ipc.Client.call(Client.java:1440)
...
2019-01-30 21:41:31,632 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2019-01-30 21:41:31,643 [main] INFO org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.6.5 0.16.0 acadgild 2019-01-30 21:37:40 2019-01-30 21:41:31 UNKNOWN
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReduceTime Alias
job_1548860494249_0001 1 0 n/a n/a n/a 0 0 0 A,B MAP_ONLY /user/acadgild/project/batch1/formattedweb,
Input(s):
Successfully read 0 records from: "/user/acadgild/project/batch1/web"
Output(s):
Successfully stored 0 records in: "/user/acadgild/project/batch1/formattedweb"
Counters:
Total records written : 0
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_1548860494249_0001
2019-01-30 21:41:31,658 [main] INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at localhost/127.0.0.1:8032

```

Hive script successfully load the data into hive terminal,

```

2019-01-30 21:43:06,428 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 3 time(s); retry policy is Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:08,429 [main] INFO org.apache.hadoop.ipc.Client - Retrying connect to server: 0.0.0.0/0.0.0.0:10020. Already tried 4 time(s); retry policy is Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:09,430 [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 Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:09,432 [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 Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:10,434 [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 Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:11,435 [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 Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:12,436 [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 Retry
UpToMaximumCountWithFixedSleep(maxRetries=10, sleepTime=1000 MILLISSECONDS)
2019-01-30 21:43:12,538 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Unable to retrieve job to compute warning aggregation.
2019-01-30 21:43:12,538 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2019-01-30 21:43:12,737 [main] INFO org.apache.pig.Main - Pig script completed in 5 minutes, 54 seconds and 313 milliseconds (35431 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-2.0.2.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: 25.904 seconds
OK
Time taken: 2.043 seconds
Loading data to table project.formatted_input partition (batchid=1)
OK
Time taken: 5.822 seconds
Loading data to table project.formatted_input partition (batchid=1)
OK
Time taken: 3.606 seconds
you have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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

The output of **dataformatting.sh** script in HDFS folders:

```

drwxr-xr-x - acadgild supergroup 0 2018-01-20 16:29 project
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
18/01/20 19:05:34 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
able
Found 1 items
drwxr-xr-x - acadgild supergroup 0 2018-01-20 18:12 /user/acadgild/project/batch1
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
18/01/20 19:05:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
able
Found 3 items
drwxr-xr-x - acadgild supergroup 0 2018-01-20 18:12 /user/acadgild/project/batch1/formattedweb
drwxr-xr-x - acadgild supergroup 0 2018-01-20 18:12 /user/acadgild/project/batch1/mob
drwxr-xr-x - acadgild supergroup 0 2018-01-20 18:11 /user/acadgild/project/batch1/web
[acadgild@localhost ~]$
[acadgild@localhost ~]$
[acadgild@localhost ~]$
[acadgild@localhost ~]$
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/formattedweb
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
18/01/20 19:07:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
able
Found 2 items
-rw-r--r-- 1 acadgild supergroup 0 2018-01-20 18:12 /user/acadgild/project/batch1/formattedweb/_SUCCESS
-rw-r--r-- 1 acadgild supergroup 1241 2018-01-20 18:12 /user/acadgild/project/batch1/formattedweb/part-m-00000
[acadgild@localhost ~]$ 
```

The output of the **formattedweb** data obtained from the **Dataformatting.pig** is shown in the below screen shot,

Command,

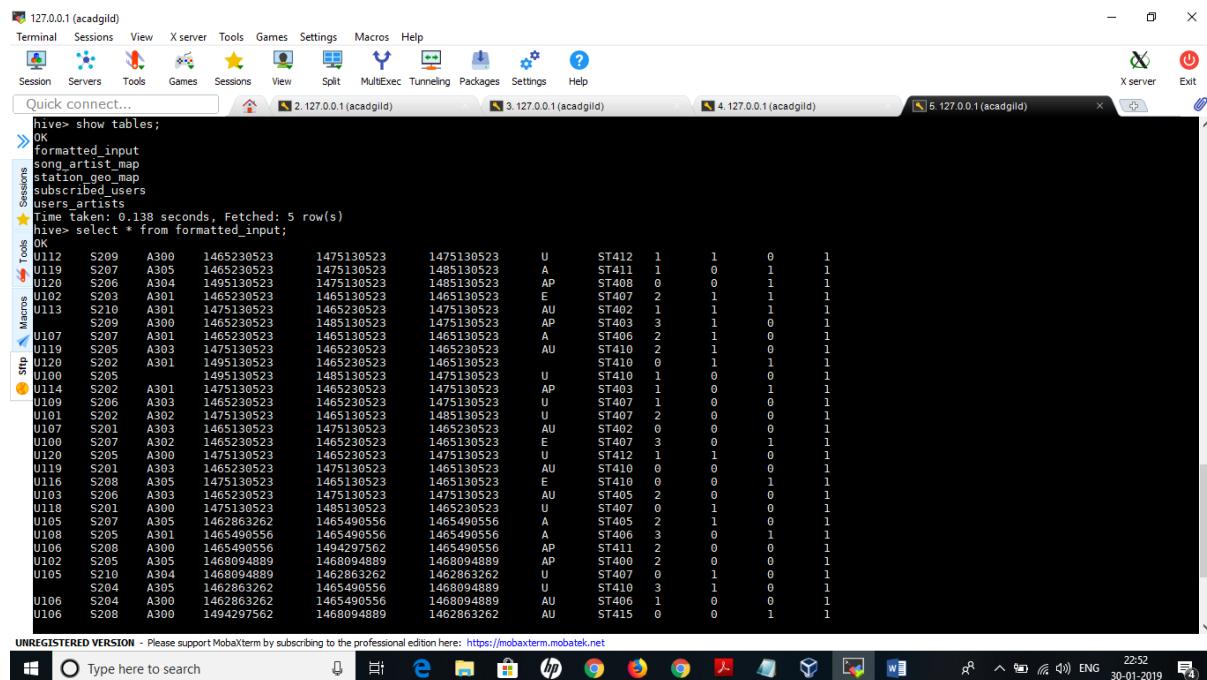
```
hadoop fs -cat /user/acadgild/project/batch1/formattedweb/*
```

```
[acadgild@localhost ~]$ 
[acadgild@localhost ~]$ hadoop fs -cat /user/acadgild/project/batch1/formattedweb/*
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7
stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or li
18/01/20 19:09:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library fo
able
U113,S205,A305,1462863262,1465490556,1462863262,AP,ST407,3,0,1
U102,S200,A301,1494297562,1465490556,1465490556,A,ST400,1,0,1
U115,S207,A301,1494297562,1468094889,1465490556,AU,ST406,2,1,1
U110,S201,A300,1468094889,1462863262,1468094889,AU,ST413,2,0,1
U102,S203,A305,1465490556,1494297562,1465490556,A,ST414,2,0,0
,S209,A304,1465490556,1462863262,1465490556,E,ST412,0,0,1
U105,S203,A300,1462863262,1468094889,1468094889,U,ST407,2,1,1
U113,S205,A303,1462863262,1468094889,1468094889,E,ST415,2,0,1
U120,S205,A302,1494297562,1494297562,,ST400,0,1,0
U105,S210,,1468094889,1462863262,1494297562,E,ST410,1,0,1
U117,S206,A300,1468094889,1468094889,1465490556,A,ST414,2,0,0
U114,S200,A301,1462863262,1468094889,1462863262,AP,ST408,1,1,1
U110,S208,A303,1494297562,1468094889,1468094889,E,ST405,1,0,1
U115,S201,A303,1465490556,1465490556,1494297562,AU,ST407,2,1,1
U103,S209,A305,1465490556,1468094889,1468094889,AU,ST408,3,0,1
U112,S210,A303,1494297562,1494297562,1462863262,AU,ST408,2,1,0
U118,S202,A301,1468094889,1465490556,1468094889,AP,ST414,0,0,1
U100,S200,A301,1462863262,1494297562,1494297562,AU,ST408,2,0,0
U113,S210,A304,1468094889,1465490556,1494297562,E,ST403,2,0,1
U104,S203,A300,1468094889,1468094889,1494297562,AU,ST406,1,0,1
[acadgild@localhost ~]$
[acadgild@localhost ~]$
```

The new Tables has been created and show below,

DataFormatting.sh output in hive terminal,

```
hive> select * from formatted_input;
```



A screenshot of the MobaXterm terminal window titled "127.0.0.1 (acadgild)". The window shows the output of a Hive query:

```
hive> show tables;
OK
Formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.138 seconds, Fetched: 5 row(s)
hive> select * from formatted_input;
OK
U112 S209 A300 1465230523 1475130523 1475130523 U ST412 1 1 0 1
U119 S207 A305 1465230523 1475130523 1485130523 A ST411 1 0 1 1
U120 S206 A304 1495130523 1475130523 1485130523 AP ST408 0 0 1 1
U102 S203 A301 1465230523 1465130523 1465130523 E ST407 2 1 1 1
U113 S210 A301 1475130523 1465230523 AU ST402 1 1 1 1
S209 A300 1465230523 1485130523 AP ST403 3 1 0 1
U107 S207 A301 1465230523 1465130523 A ST406 2 1 0 1
U119 S205 A303 1495130523 1465230523 1465230523 AU ST410 2 1 0 1
U120 S202 A301 1495130523 1465230523 1465230523 ST410 0 1 1 1
U100 S205 1495130523 1475130523 U ST410 1 0 0 1
U114 S202 A301 1475130523 1465230523 AP ST403 1 0 1 1
U109 S206 A303 1465230523 1475130523 U ST407 1 0 0 1
U101 S202 A302 1475130523 1465130523 1485130523 U ST407 2 0 0 1
U107 S201 A303 1475130523 1465230523 AU ST402 0 0 0 1
U106 S207 A302 1465230523 1465230523 E ST407 3 0 1 1
U120 S205 A300 1475130523 1465230523 U ST412 1 1 0 1
U119 S201 A303 1465230523 1475130523 AU ST410 0 0 0 1
U116 S208 A305 1475130523 1465130523 E ST410 0 0 1 1
U103 S206 A303 1465230523 1475130523 AU ST405 2 0 0 1
U118 S201 A300 1475130523 1465230523 U ST407 0 1 0 1
U105 S207 A305 1462863262 1465490556 1465490556 A ST405 2 1 0 1
U108 S205 A301 1465490556 1465490556 A ST406 3 0 1 1
U106 S208 A300 1465490556 1494297562 1465490556 AP ST411 2 0 0 1
U102 S205 A305 1468094889 1468094889 AP ST400 2 0 0 1
U105 S204 A301 1462863262 1462863262 U ST401 0 1 0 1
S204 A305 1462863262 1465490556 1468094889 U ST410 3 1 0 1
U106 S204 A300 1462863262 1465490556 1468094889 AU ST406 1 0 0 1
U106 S208 A300 1494297562 1468094889 1462863262 AU ST415 0 0 1 1
```

The terminal also displays a message at the bottom: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

- In the above screenshot we can see the formatted input data with some null values in **user_id**, **aritist_id** and **geo_cd** columns which we will fill the enrichment script based on

rules of enrichment for **artist_id** and **geo_cd** only. We will get neglect **user_id** because they didn't mentioned anything about **user_id** for enrichment purpose.

- Data formatting phase is executed successfully by loading both **mobile** and **web** data and partitioned based on **batchid**.

Section 5 - Data Enrichment & Filtering:

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

Rules for data enrichment,

1. If any of like or dislike is **NULL** or **absent**, consider it as **0**.
2. If fields like **Geo_cd** and **Artist_id** are **NULL** or absent, consult the lookup tables for fields **Station_id** and **Song_id** respectively to get the values of **Geo_cd** and **Artist_id**.
3. If corresponding lookup entry is not found, consider that **record** to be **invalid**

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

data_enrichment.sh

```
1  #!/bin/bash
2
3  batchid=`cat /home/acadgild/project/logs/current-batch.txt`
4  LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
5  VALIDDIR=/home/acadgild/project/processed_dir/valid/batch_$batchid
6  INVALIDDIR=/home/acadgild/project/processed_dir/invalid/batch_$batchid
7
8  echo "Running hive script for data enrichment and filtering..." >> $LOGFILE
9
10 hive -hiveconf batchid=$batchid -f /home/acadgild/project/scripts/data_enrichment.hql
11
12 if [ ! -d "$VALIDDIR" ]
13 then
14 mkdir -p "$VALIDDIR"
15 fi
16
17 if [ ! -d "$INVALIDDIR" ]
18 then
19 mkdir -p "$INVALIDDIR"
20 fi
21
22 echo "Copying valid and invalid records in local file system..." >> $LOGFILE
23
24 hadoop fs -get /user/hive/warehouse/project.db/enriched_data/batchid=$batchid/status=pass/* $VALIDDIR
25 hadoop fs -get /user/hive/warehouse/project.db/enriched_data/batchid=$batchid/status=fail/* $INVALIDDIR
26
27 echo "Deleting older valid and invalid records from local file system..." >> $LOGFILE
28
29 find /home/acadgild/project/processed_dir/ -mtime +7 -exec rm {} \;
```

data_enrichment.hql

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

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace Find Next Find Previous Find in Files Find in Lines Find in Selection Find in Buffer Find in All Find in All Lines Find in All Selection Find in All Buffer Find in All Lines Find in All Selection Find in All Buffer

< data_analysis.sh X data_export.sh X wrapper.sh X data_enrichment.hql X formatted_hive_load.hql X >

```
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,
u_Timestamp STRING,
Start_ts STRING,
End_ts STRING,
Geo_cd STRING,
Station_id STRING,
Song_end_type INT,
u_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.u_timestamp,
i.start_ts,
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

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

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace Find in Files Find in Lines Find in Selection Find in Buffer Find in All Find in All Lines Find in All Selection Find in All Buffer Find in All Lines Find in All Selection Find in All Buffer

< data_analysis.sh X data_export.sh X wrapper.sh X data_enrichment.hql X formatted_hive_load.hql X >

```
i.user_id,
i.song_id,
sa.artist_id,
i.u_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.u_like IS NULL, 0, i.u_like) AS u_like,
IF (i.dislike IS NULL, 0, i.dislike) AS dislike,
i.batchid,
IF((i.u_like=1 AND i.dislike=1)
OR i.user_id IS NULL
OR i.song_id IS NULL
OR i.u_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.u_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};
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

[acadgild@localhost ~]$ ls
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/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.jar!/hive-log4j2.properties Async: true
OK
Time taken: 25.353 seconds
OK
Time taken: 1.711 seconds
No Stats for project@formatted_input, Columns: start_ts, song_id, u_timestamp, user_id, end_ts, u_like, dislike, station_id, geo_cd, song_end_type
No Stats for project@station_geo_map, Columns: station_id, geo_cd
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20190130225658_75d009af-285d-43c4-839e-cedb6f0c9547
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_1548860494249_0003, Tracking URL = http://localhost:8088/proxy/application_1548860494249_0003
Kill Command = /home/acadgild/.installs/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1548860494249_0003
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2019-01-30 22:59:57 Stage-1 map = 0%, reduce = 0%
2019-01-30 23:00:40,167 Stage-1 map = 100%, reduce = 0%
2019-01-30 23:00:40,197 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 10.76 sec
2019-01-30 23:00:40,773 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 20.74 sec
2019-01-30 23:00:40,816 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 24.37 sec
2019-01-30 23:00:43,151 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 26.89 sec

```

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
[acadgild@localhost ~]$ Type here to search └─ 127.0.0.1 (acadgild) └─ 3.127.0.0.1 (acadgild) └─ 4.127.0.0.1 (acadgild) └─ 5.127.0.0.1 (acadgild)
[acadgild@localhost ~]$ 
MapReduce Total cumulative CPU time: 26 seconds 890 msec
>>> Ended Job = job_1548860494249_0003
>>> 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_1548860494249_0004, Tracking URL = http://localhost:8088/proxy/application_1548860494249_0004
Kill Command = /home/acadgild/.installs/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1548860494249_0004
Hadoop job information for Stage-2: number of mappers: 2; number of reducers: 1
2019-01-30 23:01:47,019 Stage-2 map = 0%, reduce = 0%
2019-01-30 23:02:32,520 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 4.23 sec
2019-01-30 23:02:32,520 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 11.67 sec
2019-01-30 23:02:51,583 Stage-2 map = 100%, reduce = 67%, Cumulative CPU 16.65 sec
2019-01-30 23:02:57,847 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 21.47 sec
MapReduce Total cumulative CPU time: 21 seconds 470 msec
Ended Job = job_1548860494249_0004
>Loading data to table project.enriched_data partition (batchid=null, status=null)

Loaded : 2/2 partitions.
  Time taken to load dynamic partitions: 2.517 seconds
  Time taken for adding to write entity : 0.021 seconds
MapReduce Jobs Launched:
Stage-Stage-1: Map: 3 Reduce: 1 Cumulative CPU: 26.89 sec HDFS Read: 49613 HDFS Write: 3135 SUCCESS
Stage-Stage-2: Map: 2 Reduce: 1 Cumulative CPU: 21.47 sec HDFS Read: 24393 HDFS Write: 3271 SUCCESS
Total MapReduce CPU Time Spent: 48 seconds 360 msec
OK
Time taken: 364.874 seconds
19/01/30 23:03:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/01/30 23:03:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rm: cannot remove '/home/acadgild/project/processed_dir/': Is a directory
rm: cannot remove '/home/acadgild/project/processed_dir/invalid': Is a directory
rm: cannot remove '/home/acadgild/project/processed_dir/valid': Is a directory
you have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ 

```

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

```

total 160
drwxrwxr-x . 4 acadgild acadgild 4096 Sep 25 2017 data
-rw-rw-r-- . 1 acadgild acadgild 29957 Mar 14 2017 derby_log
-rw-rw-r-- . 1 acadgild acadgild 37428 Mar 14 2017 Flow_of_operations.jpg
-rw-rw-r-- . 1 acadgild acadgild 62209 Mar 14 2017 Flow_of_operations.pptx
drwxrwxr-x . 2 acadgild acadgild 4096 Sep 25 2017 hdfs_
drwxrwxr-x . 2 acadgild acadgild 4096 Sep 25 2017 lib_
drwxrwxr-x . 2 acadgild acadgild 4096 Sep 25 2017 logs_
drwxrwxr-x . 5 acadgild acadgild 4096 Sep 25 2017 metastore_db
drwxrwxr-x . 4 acadgild acadgild 4096 Sep 25 2017 processed_dir
drwxrwxr-x . 3 acadgild acadgild 4096 Jan 30 22:34 scripts
[acadgild@localhost project]$ cd processed_dir/
[acadgild@localhost processed_dir]$ ls -l
total 8
drwxrwxr-x . 3 acadgild acadgild 4096 Sep 25 2017 invalid
drwxrwxr-x . 3 acadgild acadgild 4096 Sep 25 2017 valid
[acadgild@localhost processed_dir]$ cd invalid/
you have new mail in /var/spool/mail/acadgild
[acadgild@localhost invalid]$ ls -l
total 4
drwxrwxr-x . 2 acadgild acadgild 4096 Jan 30 23:03 batch_1
[acadgild@localhost invalid]$ cd batch_1/
[acadgild@localhost batch_1]$ ls
.bash: LS: command not found
[acadgild@localhost batch_1]$ ls
000000_0
[acadgild@localhost batch_1]$ cd ..
[acadgild@localhost invalid]$ cd ..
[acadgild@localhost processed_dir]$ cd valid/
[acadgild@localhost valid]$ ls -l
total 4
drwxrwxr-x . 2 acadgild acadgild 4096 Jan 30 23:03 batch_1
[acadgild@localhost valid]$ cd batch_1/
[acadgild@localhost batch_1]$ ls -l
total 4
drwxrwxr-x . 1 acadgild acadgild 1601 Jan 30 23:03 000000_0
[acadgild@localhost batch_1]$ 

```

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

```

hive> show tables;
OK
+-----+
| enriched_data |
+-----+
Time taken: 0.202 seconds, Fetched: 6 row(s)
hive> select * from enriched_data;
OK
+-----+
| S200   A300   1462863262  1494297562  1468094889  AP  ST412  3   1   1   1   fail |
| S201   A301   1465490556  1462863262  1468094889  A   ST410  2   1   1   1   fail |
| S202   A302   1495130523  1465230523  1465130523  A   ST410  0   1   1   1   fail |
| S203   A303   1465230523  1465130523  1465130523  AP  ST407  2   1   1   1   fail |
| S204   A304   1465490556  1462863262  1468094889  A   ST411  2   0   1   1   fail |
| S205   A305   1465490556  1462863262  1468094889  A   ST410  3   1   0   1   fail |
+-----+
Time taken: 0.202 seconds, Fetched: 6 row(s)
hive>

```

By applying the provided rules, we have successfully accomplished Data enrichment and Filtering stage.

Section 6: Data Analysis:

In this stage we will do analysis on enriched_data by the use of hive. We will perform the analysis with the help of script file DataAnalysis.sh

DataAnalysis.sh

data_analysis.sh (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.sh data_export.sh wrapper.sh data_enrichment.hql formatted_hive_load.hql

```
#!/bin/bash

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

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

hive -hiveconf batchid=$batchid -f /home/acadgild/project/scripts/data_analysis.hql

sh /home/acadgild/project/scripts/data_export.sh

echo "Incrementing batchid..." >> $LOGFILE

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

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

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.hql create_schema.sql data_export.sh wrapper.sh

```
SET hive.auto.convert.join=false;
USE project;

CREATE TABLE IF NOT EXISTS top_10_stations
(
station_id STRING,
total_distinct_songs_played INT,
distinct_user_count INT
)
PARTITIONED BY (batchid INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

INSERT OVERWRITE TABLE top_10_stations
PARTITION(batchid=${hiveconf:batchid})
SELECT
station_id,
COUNT(DISTINCT song_id) AS total_distinct_songs_played,
COUNT(DISTINCT user_id) AS distinct_user_count
FROM enriched_data
WHERE status='pass'
AND batchid=${hiveconf:batchid}
AND u_like=1
GROUP BY station_id
ORDER BY total_distinct_songs_played DESC
LIMIT 10;

CREATE TABLE IF NOT EXISTS users_behaviour
(
user_type STRING,
duration INT
```

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

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.hql create_schema.sql data_analysis.sh data_export.sh wrapper.sh

```

CREATE TABLE IF NOT EXISTS usersBehaviour
(
    user_type STRING,
    duration INT
)
PARTITIONED BY (batchid INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

INSERT OVERWRITE TABLE usersBehaviour
PARTITION(batchid=${hiveconf:batchid})
SELECT
CASE WHEN (su.user_id IS NULL OR CAST(ed.u_timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN
'UNSUBSCRIBED'
WHEN (su.user_id IS NOT NULL AND CAST(ed.u_timestamp AS DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN
'SUBSCRIBED'
END AS user_type,
SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration
FROM enriched_data ed
LEFT OUTER JOIN subscribed_users su
ON ed.user_id=su.user_id
WHERE ed.status='pass'
AND ed.batchid=${hiveconf:batchid}
GROUP BY CASE WHEN (su.user_id IS NULL OR CAST(ed.u_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.u_timestamp AS DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN
'SUBSCRIBED' END;

CREATE TABLE IF NOT EXISTS connected_artists
(

```

Plain Text Tab Width: 8 Ln 131, Col 22 INS

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

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.hql create_schema.sql data_analysis.sh data_export.sh wrapper.sh

```

CREATE TABLE IF NOT EXISTS connected_artists
(
    artist_id STRING,
    user_count INT
)
PARTITIONED BY (batchid INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

INSERT OVERWRITE TABLE connected_artists
PARTITION(batchid=${hiveconf:batchid})
SELECT
ua.artist_id,
COUNT(DISTINCT ua.user_id) AS user_count
FROM
(
    SELECT user_id, artist_id FROM users_artists
    LATERAL VIEW explode(artists_array) artists AS artist_id
) ua
INNER JOIN
(
    SELECT artist_id, song_id, user_id
    FROM enriched_data
    WHERE status='pass'
    AND batchid=${hiveconf: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;

```

Plain Text Tab Width: 8 Ln 131, Col 22 INS

Command: sh /home/acadgild/project/scripts/data_analysis.sh

127.0.0.1 (acadgild)

Terminal Sessions View Xserver Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Xserver Exit

Quick connect...

2.127.0.0.1(acadgild) 3.127.0.0.1(acadgild) 4.127.0.0.1(acadgild) 5.127.0.0.1(acadgild)

```
m: cannot remove '/home/acadgild/project/processed_dir/valid/batch_1': Is a directory
> you have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ ls
[acadgild@localhost ~]$ sh /home/acadgild/project/scripts/data_analysis.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/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: 26.399 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20190203020725_abla448d-0314-41df-92f7-48b493ba7217
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_1549136617217_0004, Tracking URL = http://localhost:8088/proxy/application_1549136617217_0004
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1549136617217_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-02-03 02:08:16,088 Stage-1 map = 0%, reduce = 0%
2019-02-03 02:08:45,090 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.83 sec
2019-02-03 02:10:33,099 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 11.98 sec
MapReduce Total cumulative CPU time: 11 seconds 980 msec
Ended Job = job_1549136617217_0004
Launching Job 2 out of 2.
Number of reduce tasks determined at compile time: 1
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

127.0.0.1 (acadgild)

Terminal Sessions View Xserver Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Xserver Exit

Quick connect...

2.127.0.0.1(acadgild) 3.127.0.0.1(acadgild) 4.127.0.0.1(acadgild) 5.127.0.0.1(acadgild)

```
MapReduce Total cumulative CPU time: 11 seconds 980 msec
> Ended Job = job_1549136617217_0004
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 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_1549136617217_0005, Tracking URL = http://localhost:8088/proxy/application_1549136617217_0005
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1549136617217_0005
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2019-02-03 02:09:47,331 Stage-2 map = 0%, reduce = 0%
2019-02-03 02:10:33,099 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 3.22 sec
2019-02-03 02:10:33,099 Stage-2 map = 100%, reduce = 67%, Cumulative CPU 8.24 sec
2019-02-03 02:10:38,109 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 11.48 sec
MapReduce Total cumulative CPU time: 11 seconds 480 msec
Ended Job = job_1549136617217_0005
loading data to table project.top_10_stations partition (batchid=1)
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 11.98 sec HDFS Read: 12940 HDFS Write: 221 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 11.48 sec HDFS Read: 7402 HDFS Write: 139 SUCCESS
Total MapReduce CPU Time Spent: 23 seconds 460 msec
OK
Time taken: 198.271 seconds
OK
Time taken: 0.263 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20190203021043_2c88d67c-8442-4a7e-bc91-63279230bec5
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>
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



```

127.0.0.1 (acadgild)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 127.0.0.1(acadgild) 3. 127.0.0.1(acadgild) 4. 127.0.0.1(acadgild) 5. 127.0.0.1(acadgild)
Xserver Exit
Sessions Tools Macros Stop
Launching Job 1 out of 3
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_1549136617217_0008, Tracking URL = http://localhost:8088/proxy/application_1549136617217_0008/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1549136617217_0008
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2019-02-03 02:14:34,367 Stage-1 map = 0%, reduce = 0%
2019-02-03 02:15:24,241 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 7.47 sec
2019-02-03 02:15:25,333 Stage-1 map = 100%, reduce = 0%, cumulative CPU 15.48 sec
2019-02-03 02:15:46,747 Stage-1 map = 100%, reduce = 67%, cumulative CPU 20.16 sec
2019-02-03 02:15:48,183 Stage-1 map = 100%, reduce = 100%, cumulative CPU 21.23 sec
MapReduce Total cumulative CPU time: 21 seconds 230 msec
Ended Job = job_1549136617217_0008
Launching Job 2 out of 3
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_1549136617217_0009, Tracking URL = http://localhost:8088/proxy/application_1549136617217_0009/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job -kill job_1549136617217_0009
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2019-02-03 02:16:20,023 Stage-2 map = 0%, reduce = 0%
2019-02-03 02:16:40,784 Stage-2 map = 100%, reduce = 0%, cumulative CPU 3.34 sec
2019-02-03 02:17:03,264 Stage-2 map = 100%, reduce = 100%, cumulative CPU 8.47 sec
MapReduce Total cumulative CPU time: 8 seconds 470 msec
Ended Job = job_1549136617217_0009
Launching Job 3 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Windows Start Type here to search Taskbar 02:44 03-02-2019

```

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

station_id
ST407
ST414
ST411
ST402
ST406
ST405

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

user_type	duration
SUBSCRIBED	93861594
UNSUBSCRIBED	105594881

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

artist_id
A303
A302
A300

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

song_id
S208
S207
S206
S209
S200
S204
S202
S205

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

user_id
U117
U118
U110
U120
U115
U107
U108
U109
U106
U100

The data analysis result is shown in the Hive tables below in the screen shot,

```

Time taken: 0.097 seconds, Fetched: 11 row(s)
hive> Select * From connected_artists;
OK
connected_artists.artist_id      connected_artists.user_count      connected_artists.batchid
A303      2          1
A302      2          1
A300      1          1
Time taken: 0.225 seconds, Fetched: 3 row(s)
hive> Select * From top_10_royalty_songs;
OK
top_10_royalty_songs.song_id    top_10_royalty_songs.duration    top_10_royalty_songs.batchid
S208      22627294      1
S207      20000000      1
S206      19900000      1
S209      15254588      1
S200      9900000      1
S204      2604333      1
S202      100000      1
S205      0          1
Time taken: 0.237 seconds, Fetched: 8 row(s)
hive> Select * From top_10_stations;
OK
top_10_stations.station_id      top_10_stations.total_distinct_songs_played      top_10_stations.distinct_user_count      top_10_stations.batchid
ST407      2          3          1
ST414      1          1          1
ST411      1          1          1
ST402      1          2          1
ST406      1          1          1
ST405      1          1          1
Time taken: 0.336 seconds, Fetched: 6 row(s)
hive> Select * From top_10_unsubscribed_users;
OK
top_10_unsubscribed_users.user_id      top_10_unsubscribed_users.duration      top_10_unsubscribed_users.batchid
U117      20000000      1
U118      20000000      1
U110      20000000      1
U120      12627294      1
U115      12527294      1
U115      12527294      1

```

```

Time taken: 0.237 seconds, Fetched: 8 row(s)
hive> Select * From top_10_stations;
OK
top_10_stations.station_id      top_10_stations.total_distinct_songs_played      top_10_stations.distinct_user_count      top_10_stations.batchid
ST407      2          3          1
ST414      1          1          1
ST411      1          1          1
ST402      1          2          1
ST406      1          1          1
ST405      1          1          1
Time taken: 0.336 seconds, Fetched: 6 row(s)
hive> Select * From top_10_unsubscribed_users;
OK
top_10_unsubscribed_users.user_id      top_10_unsubscribed_users.duration      top_10_unsubscribed_users.batchid
U117      20000000      1
U118      20000000      1
U110      20000000      1
U120      12627294      1
U115      12527294      1
U107      10000000      1
U108      5231627      1
U109      2604333      1
U106      2604333      1
U100      0          1
Time taken: 0.275 seconds, Fetched: 10 row(s)
hive> Select * From usersBehaviour;
OK
usersBehaviour.user_type      usersBehaviour.duration      usersBehaviour.batchid
SUBSCRIBED      93861594      1
UNSUBSCRIBED      105594881      1
Time taken: 0.274 seconds, Fetched: 2 row(s)
hive>
>
>
```

Section 7 – Data Storage in MYSQL

Using the bash file shown below, **data_export.sh** we are going to export the data from the hive tables into mysql using **Sqoop** export.

create_schema.sql – Make sure that you logged in to MySql. The below schema will create the database and tables in the MySQL.

create_schema.sql (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.hql create_schema.sql data_analysis.sh data_export.sh wrapper.sh

```

CREATE DATABASE IF NOT EXISTS project;
USE project;

CREATE TABLE IF NOT EXISTS top_10_stations
(
station_id VARCHAR(50),
total_distinct_songs_played INT,
distinct_user_count INT
);

CREATE TABLE IF NOT EXISTS users_behaviour
(
user_type VARCHAR(50),
duration BIGINT
);

CREATE TABLE IF NOT EXISTS connected_artists
(
artist_id VARCHAR(50),
user_count INT
);

CREATE TABLE IF NOT EXISTS top_10_royalty_songs
(
song_id VARCHAR(50),
duration BIGINT
);

CREATE TABLE IF NOT EXISTS top_10_unsubscribed_users
(
user_id VARCHAR(50),
duration BIGINT
);

```

SQL Tab Width: 8 Ln 1, Col 1 INS

data_export.sh

data_export.sh (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

data_analysis.hql create_schema.sql data_analysis.sh data_export.sh wrapper.sh

```

#!/bin/bash

#This script is not working.
#Either change table to text or use STRING as type of partitioned column

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

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

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

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

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--password 'Root@123' \
--table top_10_stations \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/top_10_stations/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--password 'Root@123' \
--table users_behaviour \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/users_behaviour/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1

sqoop export \

```

sh Tab Width: 8 Ln 1, Col 1 INS

data_export.sh (~/project/scripts) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace Find Next Find Previous Find in Files Find in Lines Find in Selection Find in Buffer Find in Tab

data_analysis.hql create_schema.sql data_analysis.sh data_export.sh wrapper.sh

```
--password 'Root@123' \
--table usersBehaviour \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/usersBehaviour/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--password 'Root@123' \
--table connectedArtists \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/connected_artists/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--password 'Root@123' \
--table top10RoyaltySongs \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/top10RoyaltySongs/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1

sqoop export \
--connect jdbc:mysql://localhost/project \
--username 'root' \
--password 'Root@123' \
--table top10UnsubscribedUsers \
--export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/top10UnsubscribedUsers/batchid=$batchid \
--input-fields-terminated-by ',' \
-m 1
```

sh Tab Width: 8 Ln 1, Col 1 1 Item in Trash

```
stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
18/01/24 09:57:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
drwxrwxr-x  - acadgild supergroup          0 2018-01-24 09:34 hdfs://localhost:9000/user/hive/warehouse/project.db/top10stations/batchid=1
[acadgild@localhost ~]$ sqoop export --connect jdbc:mysql://localhost/project --username root --password acadgild --table top10stations --export-dir hdfs://localhost:9000/user/hive/warehouse/project.db/top10stations/batchid=1 --input-fields-terminated-by ',' -m 1
Warning: /home/acadgild/sqoop-1.4.6-bin_hadoop-2.0.4-alpha../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation
Warning: /home/acadgild/sqoop-1.4.6-bin_hadoop-2.0.4-alpha../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /home/acadgild/sqoop-1.4.6-bin_hadoop-2.0.4-alpha../zookeeper does not exist! Zookeeper imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-24 09:58:23,657 INFO [main] sqoop.Sqoop: Running Sqoop version: 1.4.6
2018-01-24 09:58:23,694 WARN [main] tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2018-01-24 09:58:24,084 INFO [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-24 09:58:24,085 INFO [main] tool.CodeGenTool: Beginning code generation
2018-01-24 09:58:24,696 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `top10stations` AS t LIMIT 1
2018-01-24 09:58:24,767 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `top10stations` AS t LIMIT 1
2018-01-24 09:58:24,810 INFO [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /home/acadgild/hadoop-2.7.2
Note: /tmp/sqoop-acadgild/compile/flae2653abc7121116a39d1b97e8735b/top10stations.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
2018-01-24 09:58:29,735 INFO [main] orm.CompilationManager: Writing jar file: /tmp/sqoop-acadgild/compile/flae2653abc7121116a39d1b97e8735b/top10stations.jar
2018-01-24 09:58:29,763 INFO [main] mapreduce.ExportJobBase: Beginning export of top10stations
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/hbase-1.0.3/lib/slf4j-log4j12-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
2018-01-24 09:58:30,180 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2018-01-24 09:58:30,199 INFO [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2018-01-24 09:58:32,079 INFO [main] Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, use mapreduce
```

```

2018-01-24 10:06:14,238 INFO [main] input.FileInputFormat: Total input paths to process : 1
2018-01-24 10:06:14,435 INFO [main] mapreduce.JobSubmitter: number of splits:1
2018-01-24 10:06:14,464 INFO [main] Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.ma
p.speculative
2018-01-24 10:06:14,791 INFO [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1516764714140_0018
2018-01-24 10:06:16,293 INFO [main] impl.YarnClientImpl: Submitted application application_1516764714140_0018
2018-01-24 10:06:16,420 INFO [main] mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1516764714140_0018/
2018-01-24 10:06:16,424 INFO [main] mapreduce.Job: Running job: job_1516764714140_0018
2018-01-24 10:06:45,186 INFO [main] mapreduce.Job: Job job_1516764714140_0018 running in uber mode : false
2018-01-24 10:06:45,191 INFO [main] mapreduce.Job: map 0% reduce 0%
2018-01-24 10:07:01,634 INFO [main] mapreduce.Job: map 100% reduce 0%
2018-01-24 10:07:02,707 INFO [main] mapreduce.Job: Job job_1516764714140_0018 completed successfully
2018-01-24 10:07:03,246 INFO [main] mapreduce.Job: Counters: 30
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=136426
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=311
    HDFS: Number of bytes written=0
    HDFS: Number of read operations=4
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=0
  Job Counters
    Launched map tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=12452
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=12452
    Total vcore-seconds taken by all map tasks=12452
    Total megabyte-seconds taken by all map tasks=12750848
  Map-Reduce Framework
    Map input records=10
    Map output records=10
    Input split bytes=178
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0

```

Output:

```

mysql>
mysql> use project;
Database changed
mysql> show tables;
+-----+
| Tables_in_project |
+-----+
| connected_artists
| top_10_royalty_songs
| top_10_stations
| top_10_unsubscribed_users
| users_behaviour |
+-----+
5 rows in set (0.00 sec)

mysql> Select * From top_10_stations;
+-----+-----+-----+
| station_id | total_distinct_songs_played | distinct_user_count |
+-----+-----+-----+
| ST407      | 2                | 3                 |
| ST414      | 1                | 1                 |
| ST411      | 1                | 1                 |
| ST402      | 1                | 2                 |
| ST406      | 1                | 1                 |
| ST405      | 1                | 1                 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> Select * From connected_artists;
+-----+-----+
| artist_id | user_count |
+-----+-----+
| A303     | 2          |
| A302     | 2          |
| A300     | 1          |
+-----+-----+
3 rows in set (0.00 sec)

```

```

mysql> Select * From top_10_royalty_songs;
+-----+-----+
| song_id | duration |
+-----+-----+
| S208    | 22627294 |
| S207    | 20000000 |
| S206    | 19900000 |
| S209    | 15254588 |
| S200    | 9900000  |
| S204    | 2604333  |
| S202    | 100000   |
| S205    | 0        |
+-----+-----+
8 rows in set (0.00 sec)

```

```

mysql> Select * From top_10_unsubscribed_users;
+-----+-----+
| user_id | duration |
+-----+-----+
| U117    | 20000000 |
| U118    | 20000000 |
| U110    | 20000000 |
| U120    | 12627294 |
| U115    | 12527294 |
| U107    | 10000000 |
| U108    | 5231627  |
| U109    | 2604333  |
| U106    | 2604333  |
| U100    | 0        |
+-----+-----+
10 rows in set (0.01 sec)

mysql> Select * From users_behaviour;
+-----+-----+
| user_type | duration |
+-----+-----+
| SUBSCRIBED | 93861594 |
| UNSUBSCRIBED | 105594881 |
+-----+-----+
2 rows in set (0.00 sec)

```

Job Scheduling:

Now after exporting data into MySQL **batchid** will be incremented to additional 1 means one batch of data operations is successfully completed and new batch of data will be loaded for the analysis after every 3 hours.

```

21   --driver-class-path /home/acadgild/apache-hive-2.1.0-bin/lib/hive-hbase-handler-
22   /home/acadgild/project/lib/sparkanalysis.jar $batchid
23
24 echo "Exporting data to MYSQL using sqoop export..." >> $LOGFILE
25 sh /home/acadgild/project/scripts/data_export.sh
26
27 echo "Incrementing batchid..." >> $LOGFILE
28 batchid=`expr $batchid + 1`
29 echo -n $batchid > /home/acadgild/project/logs/current-batch.txt
30
31

```

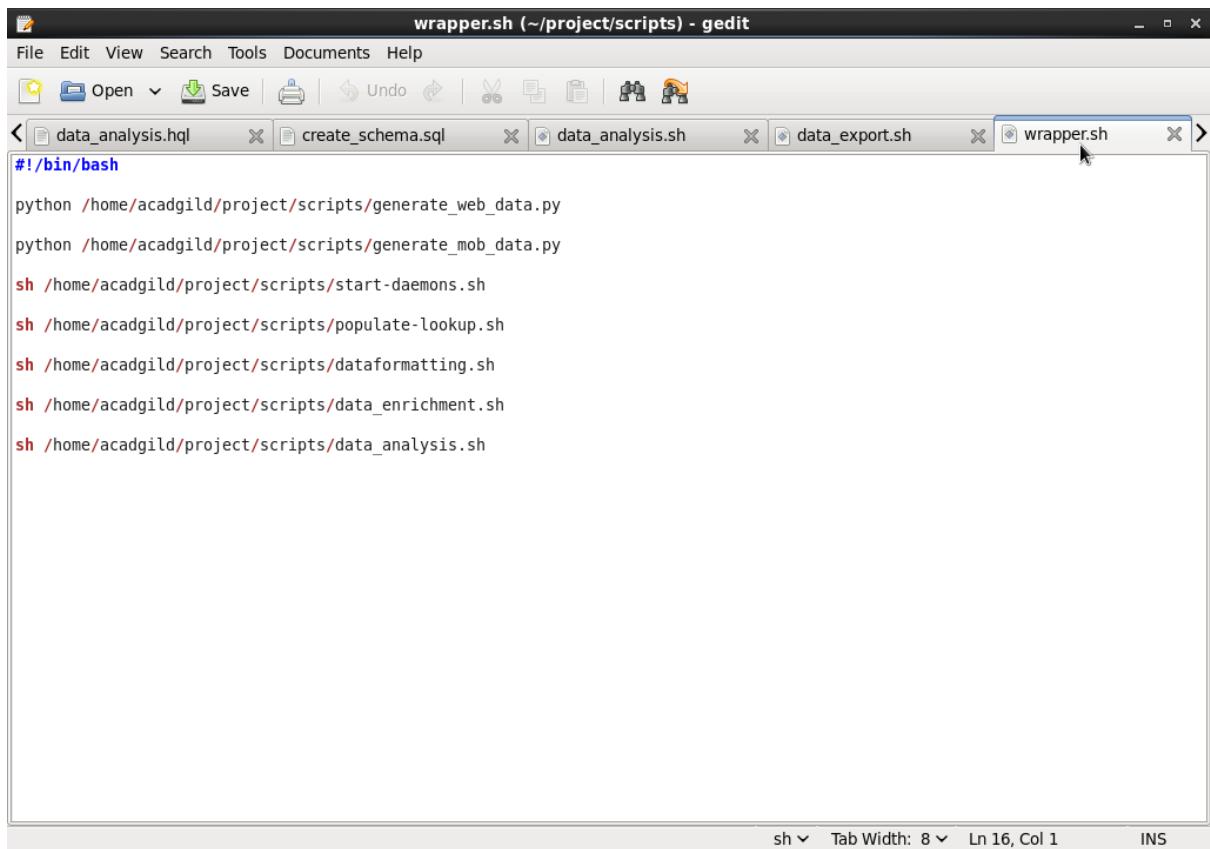
We can check logs to track the behavior of the operations we have done on the data and overcome failures in the pipeline and we can see the **batchid** incremented value in **current-batch.txt**

```
[acadgild@localhost project]$ cd logs
[acadgild@localhost logs]$ ls -l
total 24
-rwxrwxr-x. 1 acadgild acadgild 1 Jan 24 09:44 current-batch.txt
-rw-rw-r--. 1 acadgild acadgild 679 Jan 24 09:03 derby.log
drwxrwxr-x. 3 acadgild acadgild 4096 Jan 24 09:02 hdfs:
-rw-rw-r--. 1 acadgild acadgild 523 Jan 24 09:44 log_batch_1
-rw-rw-r--. 1 acadgild acadgild 77 Jan 24 09:44 log_batch_1???
drwxrwxr-x. 5 acadgild acadgild 4096 Jan 24 09:03 metastore_db
[acadgild@localhost logs]$ cat current-batch.txt
2[acadgild@localhost logs]$
[acadgild@localhost logs]$
[acadgild@localhost logs]$ █
```

```
[acadgild@localhost logs]$ cat log_batch_1
Starting daemons
Creating LookUp Tables
Populating LookUp Tables
Creating hive tables on top of hbase tables for data enrichment and filtering...
Placing data files from local to HDFS...
Running pig script for data formatting...
Running hive script for formatted data load...
Running hive script for data enrichment and filtering...
Copying valid and invalid records in local file system...
Deleting older valid and invalid records from local file system...
Running hive script for data analysis...
Incrementing batchid...
[acadgild@localhost logs]$ █
```

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

wrapper.sh



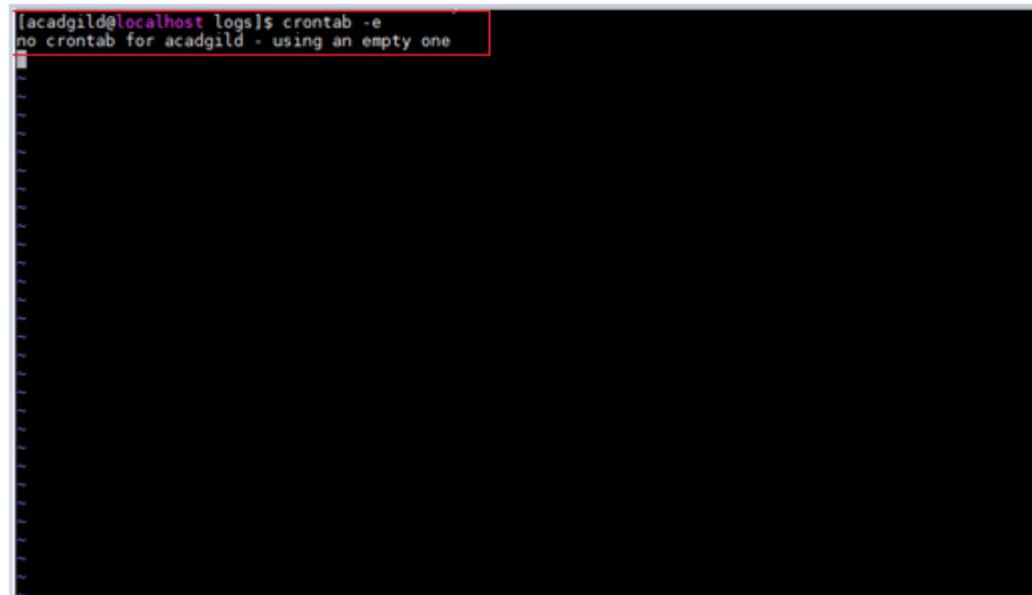
The screenshot shows a GIMP image editor window titled "wrapper.sh (~/project/scripts) - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for Open, Save, Undo, Redo, Cut, Copy, Paste, Find, and Replace. There are five tabs visible: "data_analysis.hql", "create_schema.sql", "data_analysis.sh", "data_export.sh", and "wrapper.sh". The "wrapper.sh" tab is active, displaying the following script content:

```
#!/bin/bash
python /home/acadgild/project/scripts/generate_web_data.py
python /home/acadgild/project/scripts/generate_mob_data.py
sh /home/acadgild/project/scripts/start-daemons.sh
sh /home/acadgild/project/scripts/populate-lookup.sh
sh /home/acadgild/project/scripts/dataformatting.sh
sh /home/acadgild/project/scripts/data_enrichment.sh
sh /home/acadgild/project/scripts/data_analysis.sh
```

The status bar at the bottom shows "sh" as the current shell, "Tab Width: 8", "Ln 16, Col 1", and "INS" indicating insert mode.

The **wrapper.sh** will be running for every 3 hours as per the job scheduling done below, as per the above order the wrapper.sh will run the scripts.

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



The screenshot shows a terminal window with the following command and output:

```
[acadgild@localhost logs]$ crontab -e
no crontab for acadgild - using an empty one
```

The terminal window has a black background and white text. The command "crontab -e" was entered, and the response "no crontab for acadgild - using an empty one" is shown. The cursor is positioned at the end of the command line.

```
#do this for every 3 hours
* */3 * * * date>>/home/acadgild/project/scripts/wrapper.sh >> /home/acadgild/project/scripts/jobsheduling.log
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
 
:wq!
```

```
-bash: cd: crontab: No such file or directory
[acadgild@localhost logs]$ crontab -e
no crontab for acadgild - using an empty one
crontab: installing new crontab
[acadgild@localhost logs]$
```

*Installing the **crontab** in the vm,*

The **crontab** job scheduler will run the **wrapper.sh** every 3 hours and for every 3 hours we will get incremental batch ID's. **Hence, as per the request this job scheduling has been done.**

```
Deleting older valid and invalid records from local file system...
Running hive script for data analysis...
Incrementing batchid...
[acadgild@localhost logs]$ cd
[acadgild@localhost ~]$ crontab -l
#do this for every 3 hours
* */3 * * * date>>/home/acadgild/project/scripts/wrapper.sh >> /home/acadgild/project/scripts/jobsheduling.log
[acadgild@localhost ~]$
[acadgild@localhost ~]$
[acadgild@localhost ~]$
```