Music Data Analysis

ABOUT THE PROJECT:

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

DATASET:

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

Fields present in the data files

Data files contain below fields.

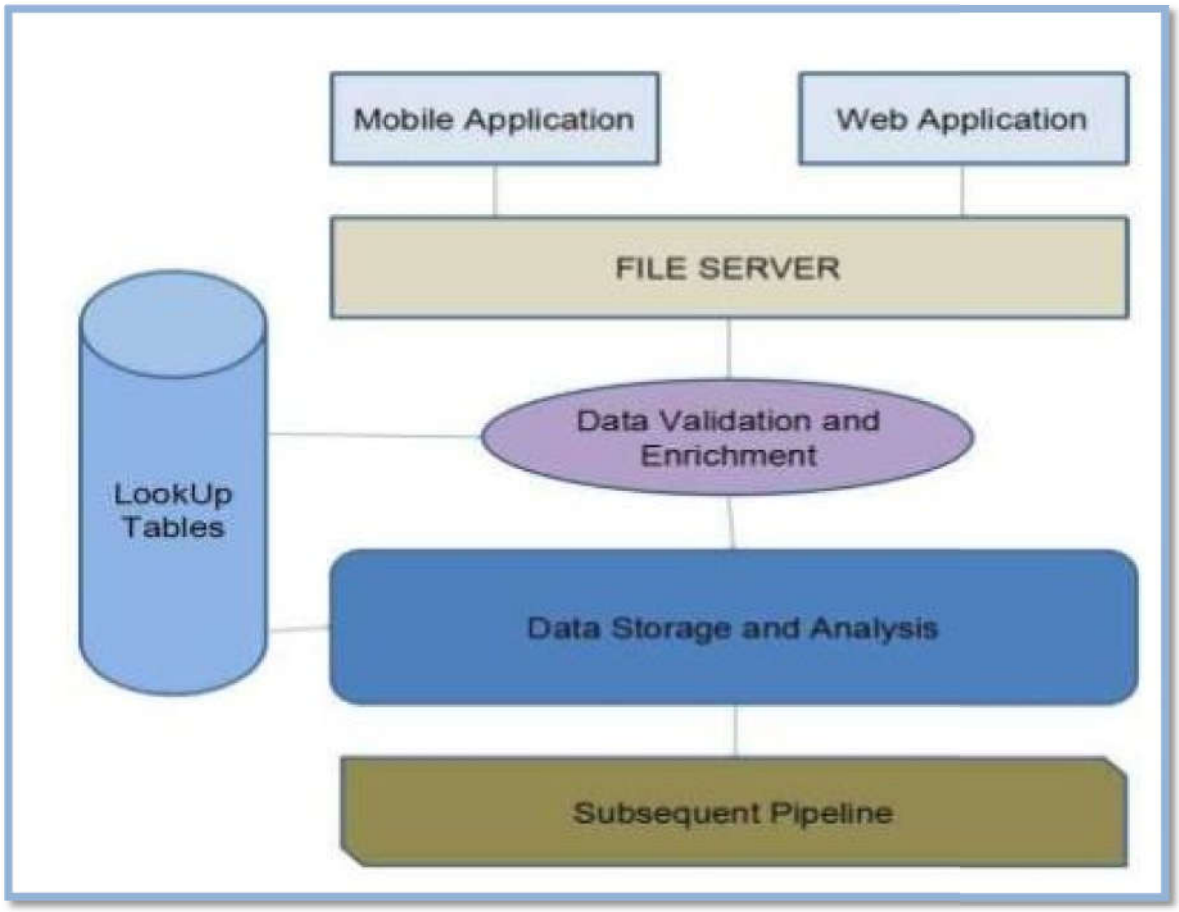
Column Name/Field Name	Column Description/Field Description
User_id	Unique identifier of every user
Song_id	Unique identifier of every song
Artist_id	Unique identifier of the lead artist of the song
Timestamp	Timestamp when the record was generated
Start_ts	Start timestamp when the song started to play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

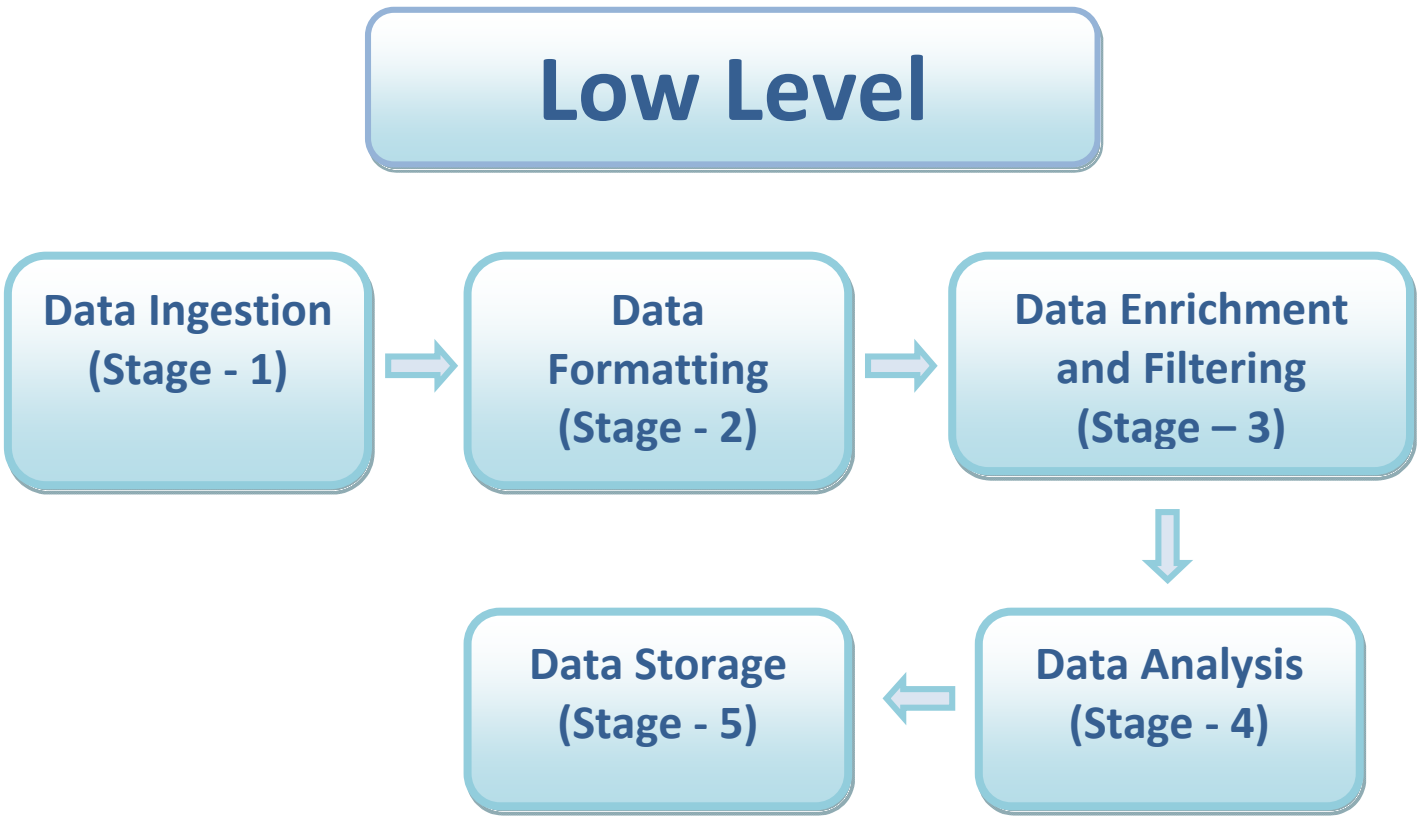
LookUp Tables

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

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

FLOW OF OPERATIONS:




Low Level
Data Ingestion (Stage - 1)
Data Formatting (Stage - 2)
Data Enrichment and Filtering (Stage – 3)
Data Analysis (Stage - 4)
Data Storage (Stage - 5)

High Level Design

(Stage - 1)

Data Ingestion

- Storage of raw data to HDFS

(Stage - 2)

Data Formatting

- Collection of web data and mob data in Hive Table
- Tools Used: Pig and Hive

(Stage – 3)

Data Enrichment and Filtering

- Use of Lookup Tables to enrich the raw data
- Segregation of valid and invalid data

(Stage - 4)

Data Analysis

- Analysis of valid data in Spark
- Creation of Hive tables to store analysed data

(Stage - 5)

Data Storage

- Export of Analyzed data from Hive to MySql

IMPLEMENTATION OF PROJECT:

1. **Below script wrapper.sh calls other scripts in sequential manner**

wrapper.sh

```
#!/bin/bash

#This script calls other scripts in sequential fashion

#Below scripts work on the data present inside web and mob folders which are already provided as part of this project

#Below script starts all required services
sh /home/acadgild/project/scripts/start-daemons.sh

#Below script creates hbase lookup tables
sh /home/acadgild/project/scripts/populate-lookup.sh

#Below script creates hive tables on top of hbase tables
sh /home/acadgild/project/scripts/data_enrichment_filtering_schema.sh

#Below script collects web and mob data, and creates a table accordingly
sh /home/acadgild/project/scripts/dataformatting_hadoop.sh

#Below script enriches and filters the data
sh /home/acadgild/project/scripts/data_enrichment.sh

#Below script performs analysis on filtered and enriched data
sh /home/acadgild/project/scripts/data_analysis.sh
```

2. **Below script start-daemons.sh starts all the required daemons**

start-daemons.sh

```
#!/bin/bash

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

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

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

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

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

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

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

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

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

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

All process started successfully

```
[acadgild@localhost ~]$ jps
8178 DataNode
8934 HMaster
8617 NodeManager
15306 Jps
8076 NameNode
8991 JobHistoryServer
8511 ResourceManager
8367 SecondaryNameNode
[acadgild@localhost ~]$
```

3. **Below script populate-lookup.sh creates lookup tables in hbase [NOSQL Database]**

```
#!/bin/bash

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

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

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

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

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

#Populating subscribed-users lookup table
file="/home/acadgild/project/lookupfiles/user-subscn.txt"
while IFS= read -r line
do
        userid=`echo $line | cut -d',' -f1`
        startdt=`echo $line | cut -d',' -f2`
        enddt=`echo $line | cut -d',' -f3`
        echo "put 'subscribed-users','$userid','subscn:startdt','$startdt'" | hbase shell
        echo "put 'subscribed-users','$userid','subscn:enddt','$enddt'" | hbase shell
done < "$file"
```

Screenshot 1

[Please refer next page for Screenshot 2]

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

#Populating user-artist-map lookup table
file="/home/acadgild/project/lookupfiles/user-artist.txt"
touch /home/cloudera/project/lookupfiles/user-artist1.txt
chmod 775 /home/cloudera/project/lookupfiles/user-artist1.txt
file1="/home/acadgild/project/lookupfiles/user-artist1.txt"
awk '$1=$1' FS="&" OFS=" " $file > $file1
num=1
while IFS= read -r line
do
        userid=`echo $line | cut -d',' -f1`
        artists=`echo $line | cut -d',' -f2`
        for row in $artists
        do
                #artist=`echo $row | cut -d',' -f$num`
                echo "put 'user-artist-map','$userid','artists:artist$num','$row'" | hbase shell
                let "num=num+1"
        done
        num=1
done < "$file1"
```

Screenshot 2

Below screenshot shows "execution of populate-lookup.sh"

```
File  Edit  View  Search  Terminal  Help
2017-08-31 20:28:01,616 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a
vailable
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

'ut 'song-artist-map', 'S209', 'artist:artistid', 'A305
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-08-31 20:28:03,442 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
0 row(s) in 0.2850 seconds

2017-08-31 20:28:07,451 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a
vailable
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

put 'subscribed-users', 'U100', 'subscn:startdt', '1465230523'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-08-31 20:28:09,229 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
0 row(s) in 0.2700 seconds

2017-08-31 20:28:13,179 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.a
vailable
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

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

```
File  Edit  View  Search  Terminal  Help

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

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

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

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

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

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

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

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

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

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

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

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

Below screenshots show that lookup tables created successfully inside hbase:

```
[acadgild@localhost ~]$ hbase shell
2017-09-11 16:30:12,586 INFO  [main] Configuration.deprecation: hadoop.native.li
b is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 2
2:35:44 PDT 2015

hbase(main):001:0> list
TABLE
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-09-11 16:32:52,785 WARN  [main] util.NativeCodeLoader: Unable to load nativ
e-hadoop library for your platform... using builtin-java classes where applicabl
e
clicks
song-artist-map
station-geo-map
subscribed-users
user-artist-map
5 row(s) in 7.7630 seconds

=> ["clicks", "song-artist-map", "station-geo-map", "studentAcad", "subscribed-u
sers", "user-artist-map"]
hbase(main):002:0>
```

```
hbase(main):002:0> describe 'user-artist-map'
Table user-artist-map is ENABLED
user-artist-map
COLUMN FAMILIES DESCRIPTION
{NAME => 'artists', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
 KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER',
 COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE =>
'65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.4970 seconds

hbase(main):003:0> describe 'song-artist-map'
Table song-artist-map is ENABLED
song-artist-map
COLUMN FAMILIES DESCRIPTION
{NAME => 'artist', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER',
COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '
65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.2130 seconds

hbase(main):004:0>
```

```
hbase(main):004:0> describe 'subscribed-users'
Table subscribed-users is ENABLED
subscribed-users
COLUMN FAMILIES DESCRIPTION
{NAME => 'subscn', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER',
COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '
65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1610 seconds

hbase(main):005:0> describe 'station-geo-map'
Table station-geo-map is ENABLED
station-geo-map
COLUMN FAMILIES DESCRIPTION
{NAME => 'geo', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEE
P_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COM
PRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '655
36', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1790 seconds

hbase(main):006:0>
```

```
hbase(main):006:0> scan 'station-geo-map'
ROW                   COLUMN+CELL
 ST400                column=geo:geo_cd, timestamp=1504898698724, value=A
 ST401                column=geo:geo_cd, timestamp=1504898736294, value=AU
 ST402                column=geo:geo_cd, timestamp=1504898771286, value=AP
 ST403                column=geo:geo_cd, timestamp=1504898813394, value=J
 ST404                column=geo:geo_cd, timestamp=1504898851287, value=E
 ST405                column=geo:geo_cd, timestamp=1504898891110, value=A
 ST406                column=geo:geo_cd, timestamp=1504898930854, value=AU
 ST407                column=geo:geo_cd, timestamp=1504898968175, value=AP
 ST408                column=geo:geo_cd, timestamp=1504899002137, value=E
 ST409                column=geo:geo_cd, timestamp=1504899037199, value=E
 ST410                column=geo:geo_cd, timestamp=1504899072294, value=A
 ST411                column=geo:geo_cd, timestamp=1504899106366, value=A
 ST412                column=geo:geo_cd, timestamp=1504899140262, value=AP
 ST413                column=geo:geo_cd, timestamp=1504899175273, value=J
 ST414                column=geo:geo_cd, timestamp=1504899211477, value=E
15 row(s) in 0.9250 seconds

hbase(main):007:0> scan 'song-artist-map'
ROW                   COLUMN+CELL
 S200                 column=artist:artistid, timestamp=1504900311045, value=A30
                      0
 S201                 column=artist:artistid, timestamp=1504900345180, value=A30
                      1
 S202                 column=artist:artistid, timestamp=1504900379094, value=A30
                      2
 S203                 column=artist:artistid, timestamp=1504900414748, value=A30
                      3
 S204                 column=artist:artistid, timestamp=1504900454884, value=A30
                      4
 S205                 column=artist:artistid, timestamp=1504900494111, value=A30
                      1
 S206                 column=artist:artistid, timestamp=1504900531288, value=A30
                      2
 S207                 column=artist:artistid, timestamp=1504900566063, value=A30
                      3
 S208                 column=artist:artistid, timestamp=1504900598693, value=A30
                      4
 S209                 column=artist:artistid, timestamp=1504900635661, value=A30
                      5
10 row(s) in 0.4300 seconds

hbase(main):008:0>
```

```
hbase(main):011:0> scan 'subscribed-users'
ROW                   COLUMN+CELL
 U100                 column=subscn:enddt, timestamp=1504899292063, value=1465130523
 U100                 column=subscn:startdt, timestamp=1504899253710, value=1465230523
 U101                 column=subscn:enddt, timestamp=1504899372837, value=1475130523
 U101                 column=subscn:startdt, timestamp=1504899334221, value=1465230523
 U102                 column=subscn:enddt, timestamp=1504899447735, value=1475130523
 U102                 column=subscn:startdt, timestamp=1504899407150, value=1465230523
 U103                 column=subscn:enddt, timestamp=1504899519614, value=1475130523
 U103                 column=subscn:startdt, timestamp=1504899481615, value=1465230523
 U104                 column=subscn:enddt, timestamp=1504899588535, value=1475130523
 U104                 column=subscn:startdt, timestamp=1504899554199, value=1465230523
 U105                 column=subscn:enddt, timestamp=1504899659222, value=1475130523
 U105                 column=subscn:startdt, timestamp=1504899622785, value=1465230523
 U106                 column=subscn:enddt, timestamp=1504899729379, value=1485130523
 U106                 column=subscn:startdt, timestamp=1504899694638, value=1465230523
 U107                 column=subscn:enddt, timestamp=1504899798027, value=1455130523
 U107                 column=subscn:startdt, timestamp=1504899763785, value=1465230523
 U108                 column=subscn:enddt, timestamp=1504899869065, value=1465230623
 U108                 column=subscn:startdt, timestamp=1504899833529, value=1465230523
 U109                 column=subscn:enddt, timestamp=1504899937623, value=1475130523
 U109                 column=subscn:startdt, timestamp=1504899903555, value=1465230523
 U110                 column=subscn:enddt, timestamp=1504900000904, value=1475130523
 U110                 column=subscn:startdt, timestamp=1504899971861, value=1465230523
 U111                 column=subscn:enddt, timestamp=1504900068726, value=1475130523
 U111                 column=subscn:startdt, timestamp=1504900034255, value=1465230523
 U112                 column=subscn:enddt, timestamp=1504900138122, value=1475130523
 U112                 column=subscn:startdt, timestamp=1504900103578, value=1465230523
 U113                 column=subscn:enddt, timestamp=1504900207293, value=1485130523
 U113                 column=subscn:startdt, timestamp=1504900171950, value=1465230523
 U114                 column=subscn:enddt, timestamp=1504900276067, value=1468130523
 U114                 column=subscn:startdt, timestamp=1504900242124, value=1465230523
15 row(s) in 0.3260 seconds
```

```
hbase(main):012:0> scan 'user-artist-map'
ROW                   COLUMN+CELL
 U100                 column=artists:artist1, timestamp=1504900671440, value=A300
 U100                 column=artists:artist2, timestamp=1504900708066, value=A301
 U100                 column=artists:artist3, timestamp=1504900742990, value=A302
 U101                 column=artists:artist1, timestamp=1504900777029, value=A301
 U101                 column=artists:artist2, timestamp=1504900812440, value=A302
 U102                 column=artists:artist1, timestamp=1504900845407, value=A302
 U103                 column=artists:artist1, timestamp=1504900878380, value=A303
 U103                 column=artists:artist2, timestamp=1504900912984, value=A301
 U103                 column=artists:artist3, timestamp=1504900947135, value=A302
 U104                 column=artists:artist1, timestamp=1504900978986, value=A304
 U104                 column=artists:artist2, timestamp=1504901015667, value=A301
 U105                 column=artists:artist1, timestamp=1504901045571, value=A305
 U105                 column=artists:artist2, timestamp=1504901079996, value=A301
 U105                 column=artists:artist3, timestamp=1504901114531, value=A302
 U106                 column=artists:artist1, timestamp=1504901152148, value=A301
 U106                 column=artists:artist2, timestamp=1504901188820, value=A302
 U107                 column=artists:artist1, timestamp=1504901234888, value=A302
 U108                 column=artists:artist1, timestamp=1504901280121, value=A300
 U108                 column=artists:artist2, timestamp=1504901322723, value=A303
 U108                 column=artists:artist3, timestamp=1504901360593, value=A304
 U109                 column=artists:artist1, timestamp=1504901395181, value=A301
 U109                 column=artists:artist2, timestamp=1504901429758, value=A303
 U110                 column=artists:artist1, timestamp=1504901464191, value=A302
 U110                 column=artists:artist2, timestamp=1504901497797, value=A301
 U111                 column=artists:artist1, timestamp=1504901531082, value=A303
 U111                 column=artists:artist2, timestamp=1504901566347, value=A301
 U112                 column=artists:artist1, timestamp=1504901600945, value=A304
 U112                 column=artists:artist2, timestamp=1504901635459, value=A301
 U113                 column=artists:artist1, timestamp=1504901670997, value=A305
 U113                 column=artists:artist2, timestamp=1504901705092, value=A302
 U114                 column=artists:artist1, timestamp=1504901739174, value=A300
 U114                 column=artists:artist2, timestamp=1504901773482, value=A301
 U114                 column=artists:artist3, timestamp=1504901807190, value=A302
15 row(s) in 0.3720 seconds
```

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

data_enrichment_filtering_schema.sh

```
#!/bin/bash

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

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

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

create_hive_hbase_lookup.hql

```
CREATE DATABASE IF NOT EXISTS project;

USE project;

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


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

CREATE EXTERNAL TABLE IF NOT EXISTS song_artist_map
(
song_id STRING,
artist_id STRING
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key,artist:artistid")
tblproperties("hbase.table.name"="song-artist-map");
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS user_artist_map
(
user_id STRING,
artists_array MAP<STRING,STRING>
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
```

Below screenshots show that all tables in hive created successfully

```
[acadgild@localhost ~]$ hive

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-com
mon-0.14.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalon
e.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hive> show databases;
OK
default
project
Time taken: 1.493 seconds, Fetched: 2 row(s)
hive> use project;
OK
Time taken: 0.201 seconds
hive>
```

```
hive> show tables;
OK
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.115 seconds, Fetched: 4 row(s)
hive> describe station_geo_map;
OK
station_id              string                  from deserializer
geo_cd                  string                  from deserializer
Time taken: 1.048 seconds, Fetched: 2 row(s)
hive> describe subscribed_users;
OK
user_id                 string                  from deserializer
subscn_start_dt         string                  from deserializer
subscn_end_dt           string                  from deserializer
Time taken: 0.539 seconds, Fetched: 3 row(s)
hive> describe song_artist_map;
OK
song_id                 string                  from deserializer
artist_id               string                  from deserializer
Time taken: 0.513 seconds, Fetched: 2 row(s)
hive> describe user_artist_map;
OK
user_id                 string                  from deserializer
artists_array           map<string,string>      from deserializer
Time taken: 0.561 seconds, Fetched: 2 row(s)
hive>
```

```
hive> select * from station_geo_map;
OK
ST400    A
ST401    AU
ST402    AP
ST403    J
ST404    E
ST405    A
ST406    AU
ST407    AP
ST408    E
ST409    E
ST410    A
ST411    A
ST412    AP
ST413    J
ST414    E
Time taken: 2.681 seconds, Fetched: 15 row(s)
hive> select * from subscribed_users;
OK
U100    1465230523      1465130523
U101    1465230523      1475130523
U102    1465230523      1475130523
U103    1465230523      1475130523
U104    1465230523      1475130523
U105    1465230523      1475130523
U106    1465230523      1485130523
U107    1465230523      1455130523
U108    1465230523      1465230623
U109    1465230523      1475130523
U110    1465230523      1475130523
U111    1465230523      1475130523
U112    1465230523      1475130523
U113    1465230523      1485130523
U114    1465230523      1468130523
Time taken: 0.926 seconds, Fetched: 15 row(s)
hive>
```

```
hive> select * from song_artist_map;
OK
S200    A300
S201    A301
S202    A302
S203    A303
S204    A304
S205    A301
S206    A302
S207    A303
S208    A304
S209    A305
Time taken: 0.569 seconds, Fetched: 10 row(s)
hive> select * from user_artist_map;
OK
U100    {"artist1":"A300","artist2":"A301","artist3":"A302"}
U101    {"artist1":"A301","artist2":"A302"}
U102    {"artist1":"A302"}
U103    {"artist1":"A303","artist2":"A301","artist3":"A302"}
U104    {"artist1":"A304","artist2":"A301"}
U105    {"artist1":"A305","artist2":"A301","artist3":"A302"}
U106    {"artist1":"A301","artist2":"A302"}
U107    {"artist1":"A302"}
U108    {"artist1":"A300","artist2":"A303","artist3":"A304"}
U109    {"artist1":"A301","artist2":"A303"}
U110    {"artist1":"A302","artist2":"A301"}
U111    {"artist1":"A303","artist2":"A301"}
U112    {"artist1":"A304","artist2":"A301"}
U113    {"artist1":"A305","artist2":"A302"}
U114    {"artist1":"A300","artist2":"A301","artist3":"A302"}
Time taken: 0.71 seconds, Fetched: 15 row(s)
hive>
```

5. **Below script dataformatting_hadoop.sh places data from local file system to hdfs**

dataformatting_hadoop.sh

```
#!/bin/bash

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

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

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

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

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

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

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

Above script calls dataformatting.pig and formatted_hive_load.hql scripts

dataformatting.pig

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

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

A = LOAD '/user/acadgild/project/batch${batchid}/web/' using org.apache.pig.piggybank.storage.XMLLoader('record') as (x:chararray);


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


STORE B INTO '/user/acadgild/project/batch${batchid}/formattedweb/' USING PigStorage(',');
```

formatted_hive_load.hql

```
USE project;

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

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

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

Below screenshots show that data has been placed successfully inside hdfs

```
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild
17/09/11 14:30:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 8 items
drwxr-xr-x   - acadgild supergroup          0 2015-11-20 11:46 /user/acadgild/Pictures
drwxr-xr-x   - acadgild supergroup          0 2017-08-22 02:13 /user/acadgild/employee
drwxr-xr-x   - acadgild supergroup          0 2017-07-05 02:14 /user/acadgild/hadoop
drwxr-xr-x   - acadgild supergroup          0 2017-08-20 14:50 /user/acadgild/my_oozie
drwxr-xr-x   - acadgild supergroup          0 2017-07-08 20:22 /user/acadgild/mydb
drwxr-xr-x   - acadgild supergroup          0 2017-08-20 23:51 /user/acadgild/oozie-acad
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:22 /user/acadgild/project
drwxr-xr-x   - acadgild supergroup          0 2015-11-17 02:00 /user/acadgild/share
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project;
17/09/11 14:30:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:24 /user/acadgild/project/bat
ch1
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1
17/09/11 14:31:13 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:26 /user/acadgild/project/bat
ch1/formattedweb
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:26 /user/acadgild/project/bat
ch1/mob
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:22 /user/acadgild/project/bat
ch1/web
[acadgild@localhost ~]$
```

```
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/mob
17/09/11 14:32:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 acadgild supergroup       6716 2017-09-09 13:17 /user/acadgild/project/bat
ch1/mob/file.txt
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/web
17/09/11 14:32:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 acadgild supergroup       6716 2017-09-09 13:22 /user/acadgild/project/bat
ch1/web/file.xml
[acadgild@localhost ~]$ hadoop fs -ls /user/acadgild/project/batch1/formattedweb
17/09/11 14:32:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 acadgild supergroup          0 2017-09-09 13:25 /user/acadgild/project/bat
ch1/formattedweb/_SUCCESS
[acadgild@localhost ~]$
```

Below screenshots show that formatted_input table is created successfully in project database inside hive

```
hive> show tables;
OK
formatted_input
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.115 seconds, Fetched: 5 row(s)
hive>
```

```
hive> describe formatted_input;
OK
user_id                 string
song_id                 string
artist_id               string
timestamp               string
start_ts                string
end_ts                  string
geo_cd                  string
station_id              string
song_end_type           int
like                    int
dislike                 int
batchid                 int

# Partition Information
# col_name              data_type               comment

batchid                 int
Time taken: 0.606 seconds, Fetched: 17 row(s)
```

```
hive> dfs -ls /user/hive/warehouse/project.db/formatted_input;
Found 1 items
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 13:26 /user/hive/warehouse/project.db/formatted_input/batchid=1
hive> dfs -ls /user/hive/warehouse/project.db/formatted_input/batchid=1;
Found 2 items
-rw-r--r--   1 acadgild supergroup       1239 2017-09-09 13:23 /user/hive/warehouse/project.db/formatted_input/batchid=1/file.txt
-rw-r--r--   1 acadgild supergroup       1236 2017-09-09 13:25 /user/hive/warehouse/project.db/formatted_input/batchid=1/part-m-00000
```

```
hive> select * from formatted_input;
OK
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       1
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       1
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       1
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       1
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       1
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       1
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       1
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       1
U108    S205    A304    1465130523      1465130523      1475130523              ST410   2       1       0       1
U105    S203            1475130523      1465230523      1465130523      AU      ST408   2       0       1       1
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       1
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       1
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       1
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       1
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       1
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       1
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       1
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       1
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       1
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       1
U106    S205    A300    1462863262      1462863262      1494297562      AP      ST407   2       1       1       1
U114    S209    A303    1465490556      1462863262      1494297562      U       ST411   2       1       0       1
U113    S203    A304    1465490556      1465490556      1462863262      U       ST405   0       0       1       1
U108    S200    A302    1468094889      1462863262      1468094889      U       ST414   0       0       1       1
U102    S203    A305    1465490556      1465490556      1494297562      U       ST404   2       0       0       1
        S208    A300    1465490556      1494297562      1465490556      U       ST411   1       0       1       1
U115    S200    A300    1465490556      1494297562      1465490556      AU      ST404   3       0       0       1
U111    S204    A300    1465490556      1465490556      1468094889      U       ST410   3       1       1       1
U120    S201    A300    1494297562      1465490556      1468094889              ST410   3       0       1       1
U113    S203            1465490556      1465490556      1465490556      A       ST402   1       1       0       1
U109    S203    A304    1462863262      1494297562      1468094889      E       ST405   1       1       1       1
U110    S202    A303    1494297562      1494297562      1468094889      AU      ST402   2       1       0       1
U100    S200    A301    1494297562      1494297562      1494297562      AP      ST410   3       1       1       1
U101    S208    A300    1462863262      1468094889      1462863262      E       ST408   0       1       1       1
U106    S206    A300    1494297562      1465490556      1462863262      A       ST405   3       1       0       1
U107    S202    A304    1494297562      1468094889      1462863262      U       ST409   0       0       0       1
U103    S204    A300    1468094889      1494297562      1465490556      AU      ST411   2       1       0       1
U103    S202    A300    1465490556      1465490556      1465490556      A       ST415   2       1       1       1
U113    S203    A303    1462863262      1468094889      1494297562      U       ST408   2       0       0       1
U113    S204    A301    1494297562      1494297562      1465490556      E       ST415   3       0       1       1
Time taken: 0.405 seconds, Fetched: 40 row(s)
hive>
```

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

data_enrichment.sh

```
#!/bin/bash

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

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

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

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

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

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

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

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

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

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

data_enrichment.hql

```
SET hive.auto.convert.join=false;
SET hive.exec.dynamic.partition.mode=nonstrict;

USE project;

CREATE TABLE IF NOT EXISTS enriched_data
(
User_id STRING,
Song_id STRING,
Artist_id STRING,
Timestamp STRING,
Start_ts STRING,
End_ts STRING,
Geo_cd STRING,
Station_id STRING,
Song_end_type INT,
Like INT,
Dislike INT
)
PARTITIONED BY (batchid INT,status STRING)
STORED AS ORC;
```

Creation of table

```
INSERT OVERWRITE TABLE enriched_data
PARTITION (batchid,status)
SELECT
i.user_id,
i.song_id,
IF(i.artist_id IS NULL OR i.artist_id='',sa.artist_id,i.artist_id) AS artist_id,
i.timestamp,
i.start_ts,
i.end_ts,
IF(i.geo_cd IS NULL OR i.geo_cd='',sg.geo_cd,i.geo_cd) AS geo_cd,
i.station_id,
IF (i.song_end_type IS NULL,3,i.song_end_type) AS song_end_type,
IF (i.like IS NULL,0,i.like) AS like,
IF (i.dislike IS NULL,0,i.dislike) AS dislike,
i.batchid,
IF((i.like=1 AND i.dislike=1)
OR i.user_id IS NULL
OR i.song_id IS NULL
OR i.timestamp IS NULL
OR i.start_ts IS NULL
OR i.end_ts IS NULL
OR i.user_id=''
OR i.song_id=''
OR i.timestamp=''
OR i.start_ts=''
OR i.end_ts=''
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};
```

Populating data inside enriched_data table

Below screenshots show that enriched_data table created successfully in project database in hive

```
hive> show tables;
OK
enriched_data
formatted_input
song_artist_map
station_geo_map
subscribed_users
user_artist_map
Time taken: 0.115 seconds, Fetched: 6 row(s)
hive>
```

```
hive> describe enriched_data;
OK
user_id                 string
song_id                 string
artist_id               string
timestamp               string
start_ts                string
end_ts                  string
geo_cd                  string
station_id              string
song_end_type           int
like                    int
dislike                 int
batchid                 int
status                  string

# Partition Information
# col_name              data_type               comment

batchid                 int
status                  string
Time taken: 0.668 seconds, Fetched: 19 row(s)
hive>
```

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

```
hive> select * from enriched_data;
OK
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       1       fail
U100    S200    A301    1494297562      1494297562      1494297562      AP      ST410   3       1       1       1       fail
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       1       fail
U103    S202    A300    1465490556      1465490556      1465490556      A       ST415   2       1       1       1       fail
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       1       fail
U109    S203    A304    1462863262      1494297562      1468094889      E       ST405   1       1       1       1       fail
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       1       fail
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       1       fail
U111    S204    A300    1465490556      1465490556      1468094889      U       ST410   3       1       1       1       fail
U113    S204    A301    1494297562      1494297562      1465490556      E       ST415   3       0       1       1       fail
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       1       fail
U106    S205    A300    1462863262      1462863262      1494297562      AP      ST407   2       1       1       1       fail
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       1       fail
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       1       fail
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       1       fail
        S208    A300    1465490556      1494297562      1465490556      U       ST411   1       0       1       1       fail
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       1       fail
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       1       fail
U101    S208    A300    1462863262      1468094889      1462863262      E       ST408   0       1       1       1       fail
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       1       fail
U115    S200    A300    1465490556      1494297562      1465490556      AU      ST404   3       0       0       1       pass
U108    S200    A302    1468094889      1462863262      1468094889      U       ST414   0       0       1       1       pass
U120    S201    A300    1494297562      1465490556      1468094889      A       ST410   3       0       1       1       pass
U107    S202    A304    1494297562      1468094889      1462863262      U       ST409   0       0       0       1       pass
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       1       pass
U110    S202    A303    1494297562      1494297562      1468094889      AU      ST402   2       1       0       1       pass
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       1       pass
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       1       pass
U102    S203    A305    1465490556      1465490556      1494297562      U       ST404   2       0       0       1       pass
U113    S203    A304    1465490556      1465490556      1462863262      U       ST405   0       0       1       1       pass
U113    S203    A303    1462863262      1468094889      1494297562      U       ST408   2       0       0       1       pass
U105    S203    A303    1475130523      1465230523      1465130523      AU      ST408   2       0       1       1       pass
U113    S203    A303    1465490556      1465490556      1465490556      A       ST402   1       1       0       1       pass
U103    S204    A300    1468094889      1494297562      1465490556      AU      ST411   2       1       0       1       pass
U108    S205    A304    1465130523      1465130523      1475130523      A       ST410   2       1       0       1       pass
U106    S206    A300    1494297562      1465490556      1462863262      A       ST405   3       1       0       1       pass
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       1       pass
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       1       pass
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       1       pass
U114    S209    A303    1465490556      1462863262      1494297562      U       ST411   2       1       0       1       pass
Time taken: 0.459 seconds, Fetched: 40 row(s)
hive>
```

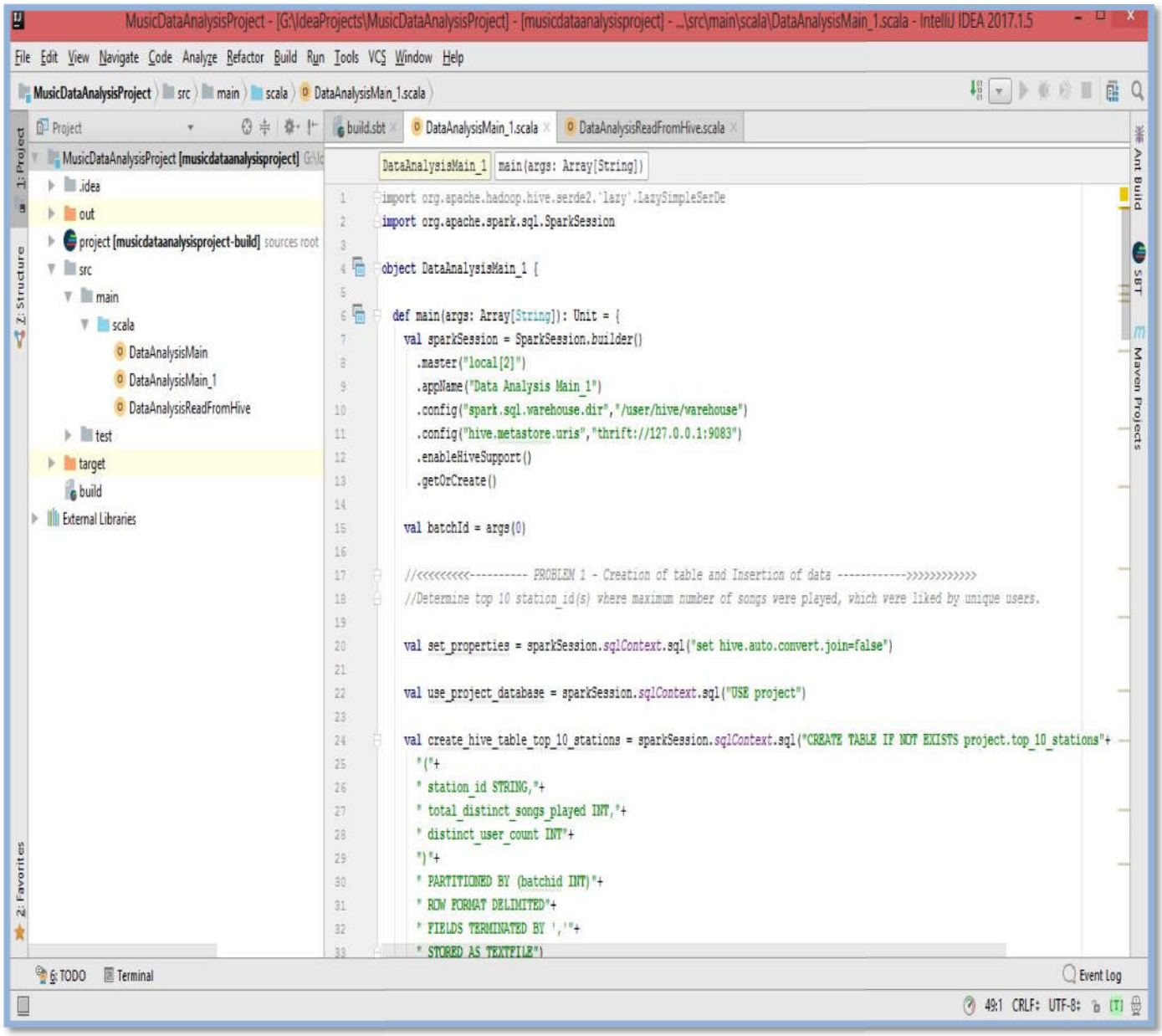
```
hive> dfs -ls /user/hive/warehouse/project.db/enriched_data;
Found 1 items
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 20:12 /user/hive/warehouse/project.db/enriched_data/batchid=1
hive> dfs -ls /user/hive/warehouse/project.db/enriched_data/batchid=1;
Found 2 items
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 20:12 /user/hive/warehouse/project.db/enriched_data/batchid=1/status=fail
drwxr-xr-x   - acadgild supergroup          0 2017-09-09 20:12 /user/hive/warehouse/project.db/enriched_data/batchid=1/status=pass
hive> dfs -ls /user/hive/warehouse/project.db/enriched_data/batchid=1/status=fail;
Found 1 items
-rw-r--r--   1 acadgild supergroup       1407 2017-09-09 20:12 /user/hive/warehouse/project.db/enriched_data/batchid=1/status=fail/000000_0
hive> dfs -ls /user/hive/warehouse/project.db/enriched_data/batchid=1/status=pass;
Found 1 items
-rw-r--r--   1 acadgild supergroup       1410 2017-09-09 20:12 /user/hive/warehouse/project.db/enriched_data/batchid=1/status=pass/000000_0
hive>
```

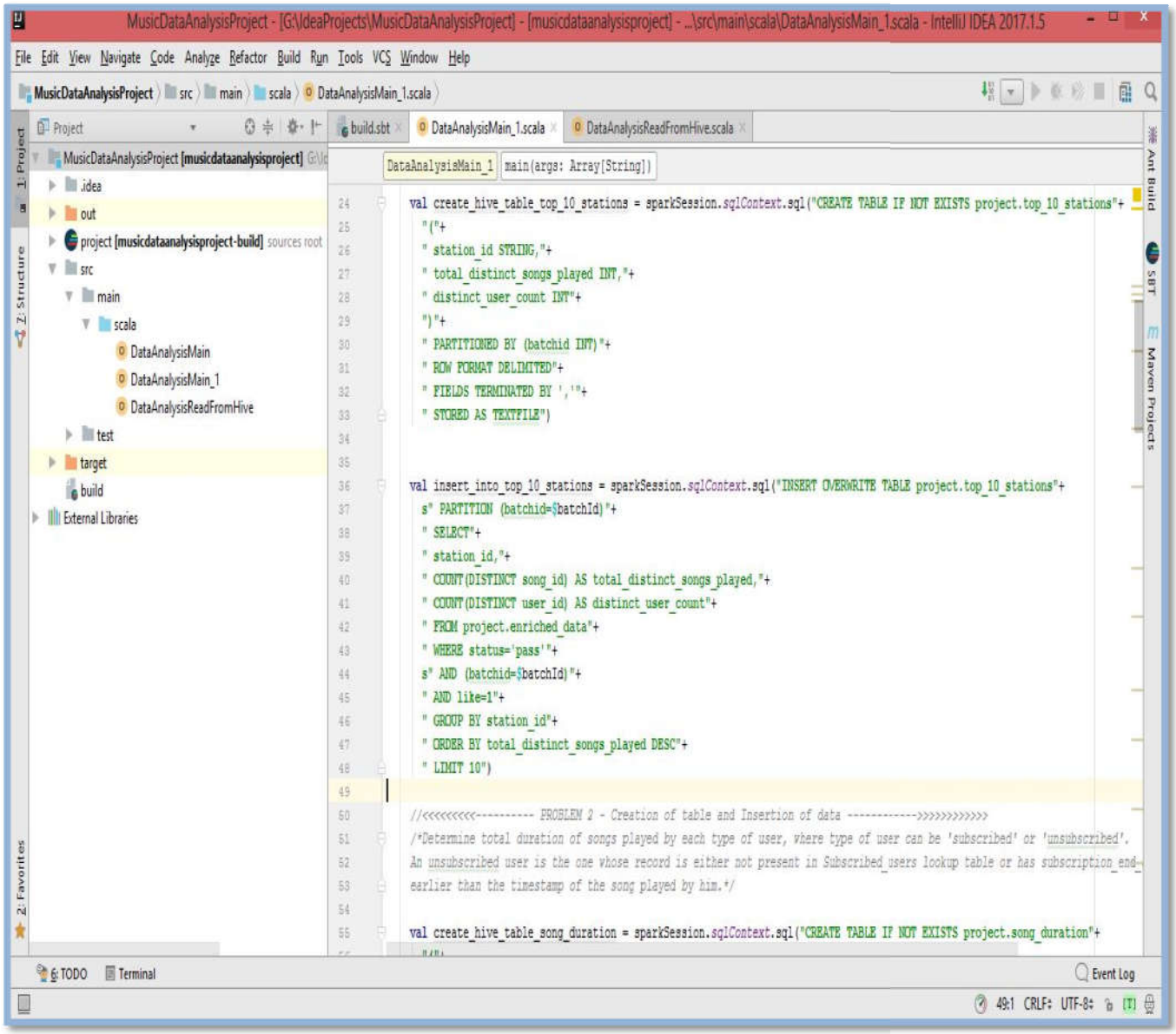
7. **Below script data_analysis_spark.sh performs following steps:**
   - ☯ **creates hive table using spark**
   - ☯ **populates the data in hive tables using spark**
   - ☯ **selects the specific data from spark**

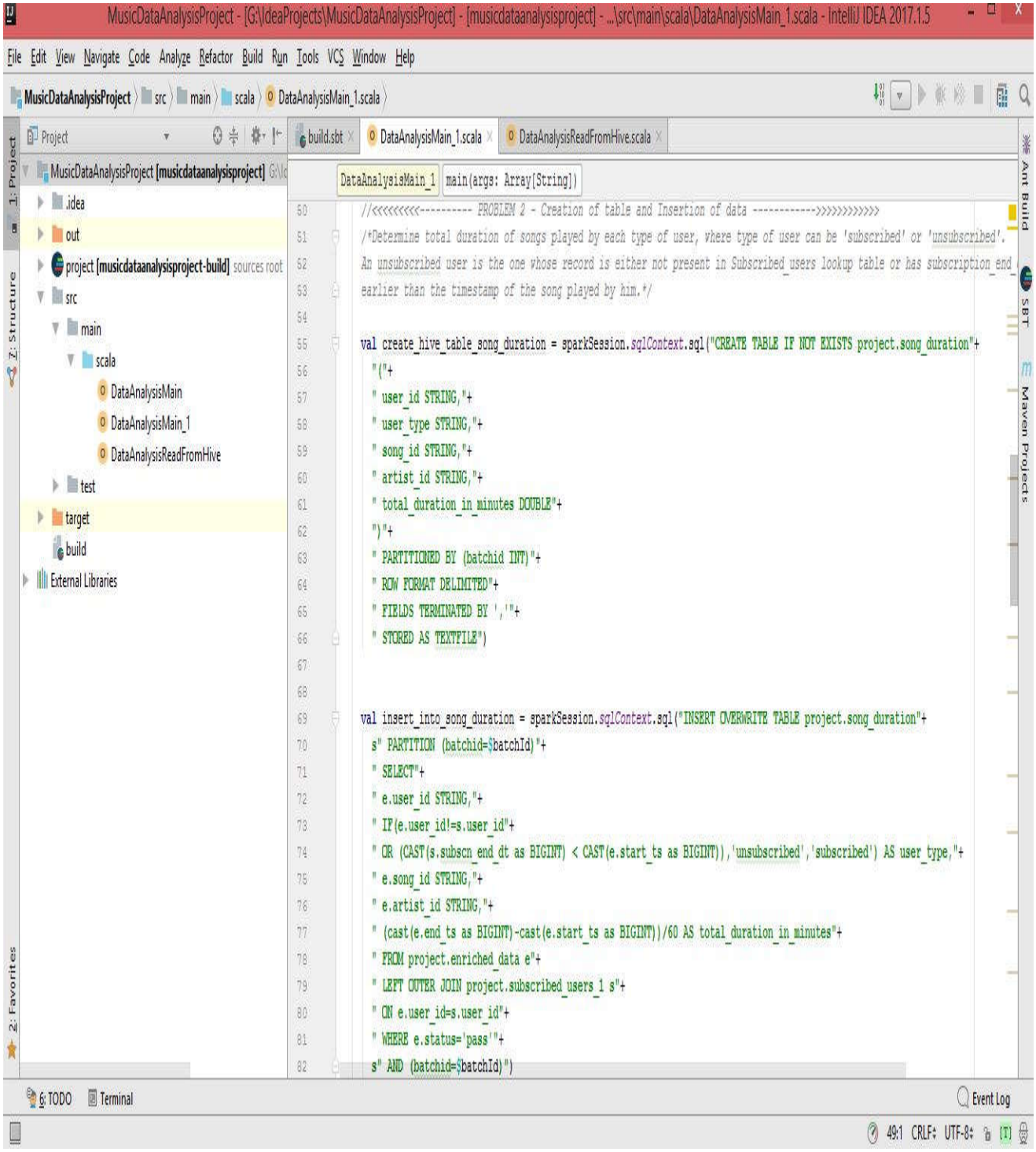
data_analysis_spark.sh

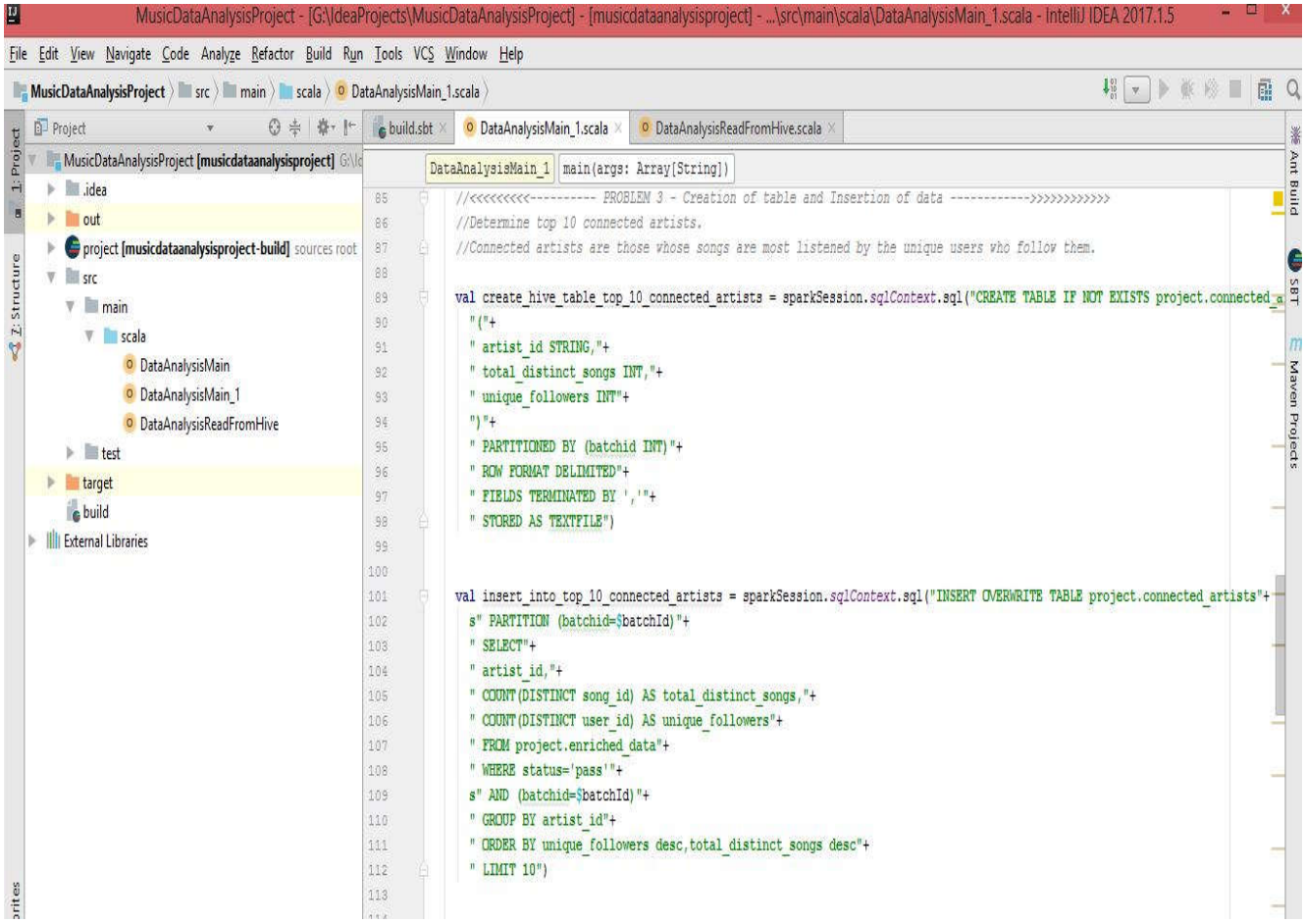
```
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
HBASE_PATH=`/usr/local/hbase/bin/hbase classpath`

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

chmod 775 /home/acadgild/project/MusicDataAnalysisProject/out/artifacts/MusicDataAnalysisProject_jar3/MusicDataAnalysisProject.jar

zip -d /home/acadgild/project/MusicDataAnalysisProject/out/artifacts/MusicDataAnalysisProject_jar3/MusicDataAnalysisProject.jar META-INF/*.RSA
META-INF/*.DSA META-INF/*.SF

spark-submit \
--class DataAnalysisMain_1 \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/*:$HBASE_PATH \
/home/acadgild/project/MusicDataAnalysisProject/out/artifacts/MusicDataAnalysisProject_jar3/MusicDataAnalysisProject.jar \$batchid

spark-submit \
--class DataAnalysisReadFromHive \
--master local[2] \
--driver-class-path /usr/local/hive/lib/hive-hbase-handler-0.14.0.jar:/usr/local/hbase/lib/*:$HBASE_PATH \
/home/acadgild/project/MusicDataAnalysisProject/out/artifacts/MusicDataAnalysisProject_jar3/MusicDataAnalysisProject.jar

echo "Exporting data to mysql..." >> $LOGFILE

#Below script exports the data from hive tables to mysql tables
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
```

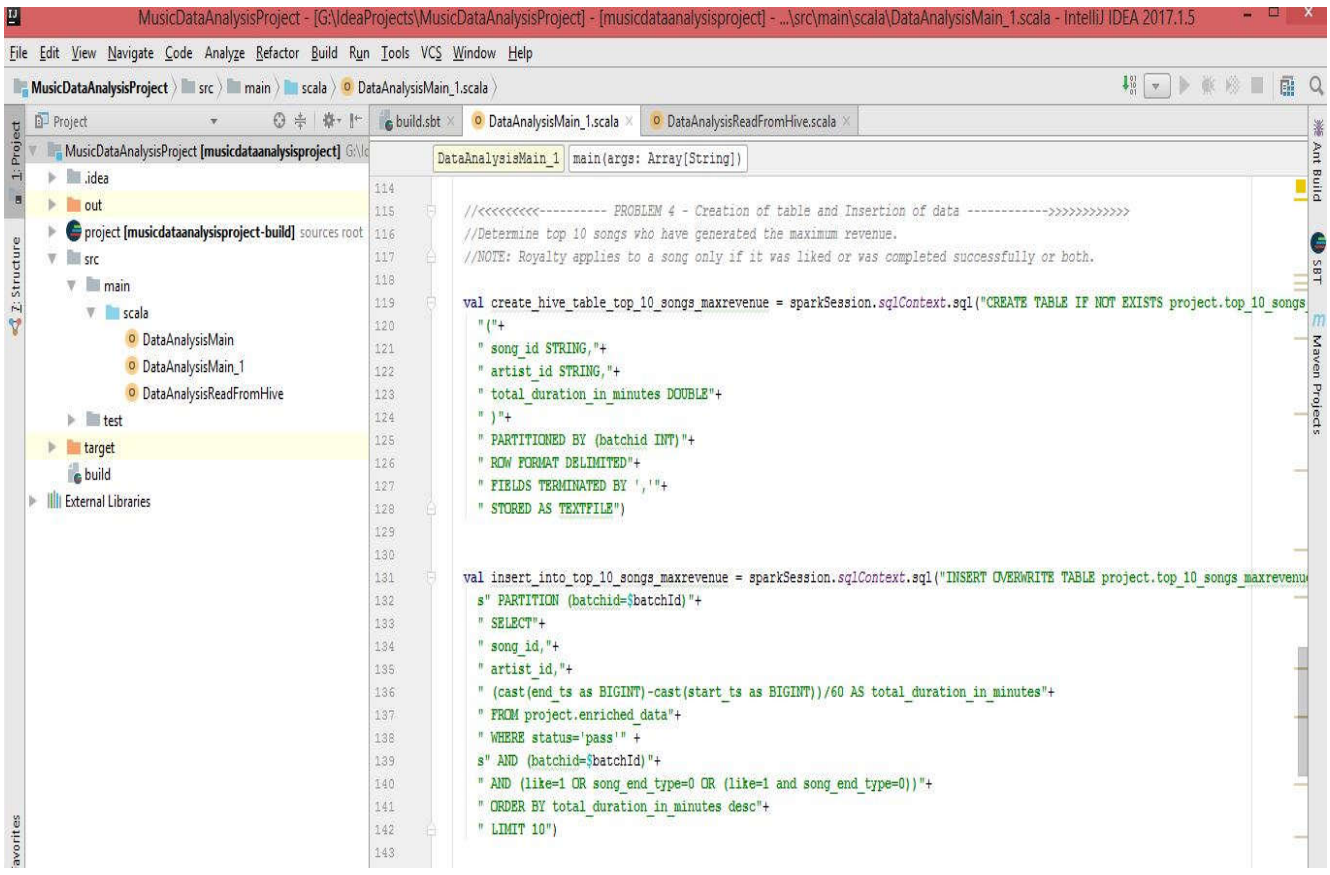
Spark Code to create hive tables and to insert data to hive tables:
[DataAnalysisMain_1.scala]

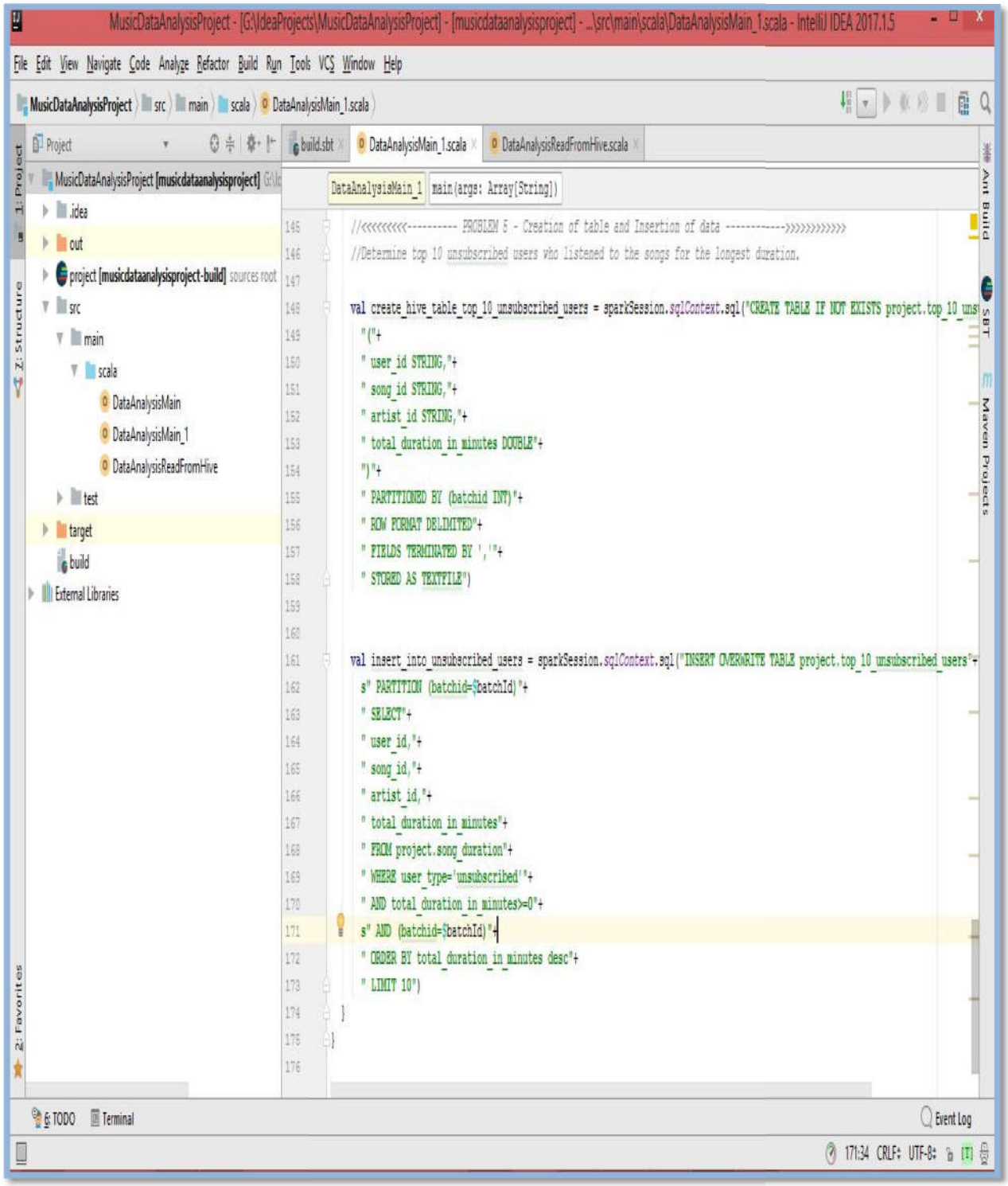



MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisMain_1.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisMain_1.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisMain_1
main(args: Array[String])
val create_hive_table_top_10_stations = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_stations"+
"("+
" station_id STRING,"+
" total_distinct_songs_played INT,"+
" distinct_user_count INT"+
")"+
" PARTITIONED BY (batchid INT)"+
" ROW FORMAT DELIMITED"+
" FIELDS TERMINATED BY ','"+
" STORED AS TEXTFILE")
val insert_into_top_10_stations = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_stations"+
s" PARTITION (batchid=$batchId)"+
" SELECT"+
" station_id,"+
" COUNT(DISTINCT song_id) AS total_distinct_songs_played,"+
" COUNT(DISTINCT user_id) AS distinct_user_count"+
" FROM project.enriched_data"+
" WHERE status='pass'"+
s" AND (batchid=$batchId)"+
" AND like=1"+
" GROUP BY station_id"+
" ORDER BY total_distinct_songs_played DESC"+
" LIMIT 10")
//<<<<<<<<<<---------- PROBLEM 2 - Creation of table and Insertion of data ---------->>>>>>>>>>
/*Determine total duration of songs played by each type of user, where type of user can be 'subscribed' or 'unsubscribed'.
An unsubscribed user is the one whose record is either not present in Subscribed_users lookup table or has subscription_end
earlier than the timestamp of the song played by him.*/
val create_hive_table_song_duration = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.song_duration"+
TODO
Terminal
Event Log
49:1 CRLF: UTF-8
Ant Build
SBT
Maven Projects
1: Project
Z: Structure
2: Favorites

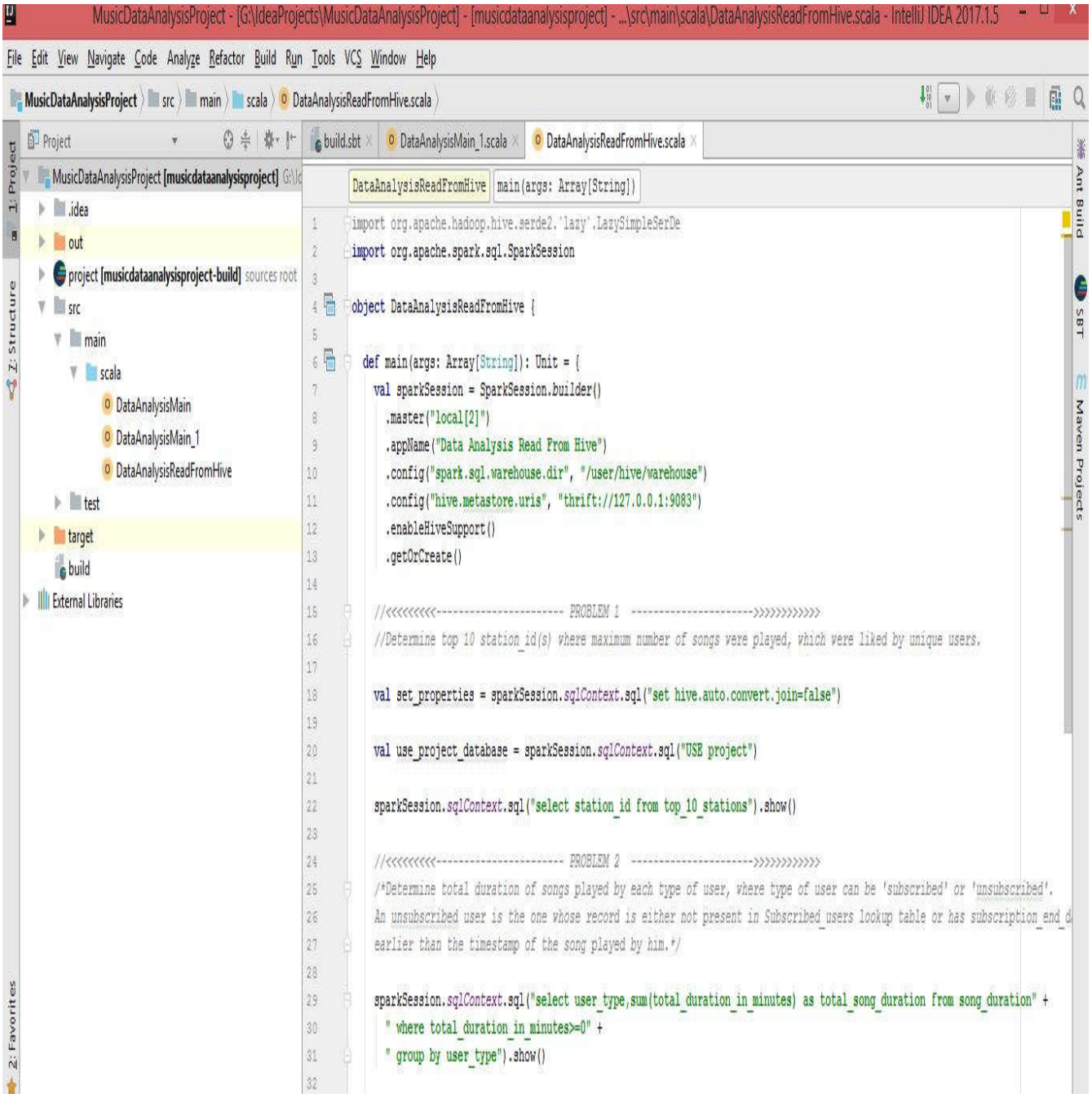

MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisMain_1.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisMain_1.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisMain_1
main(args: Array[String])
//<<<<<<<<<<---------- PROBLEM 2 - Creation of table and Insertion of data ---------->>>>>>>>>>
/*Determine total duration of songs played by each type of user, where type of user can be 'subscribed' or 'unsubscribed'.
An unsubscribed user is the one whose record is either not present in Subscribed_users lookup table or has subscription end_
earlier than the timestamp of the song played by him.*/
val create_hive_table_song_duration = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.song_duration"+
"("+
" user_id STRING,"+
" user_type STRING,"+
" song_id STRING,"+
" artist_id STRING,"+
" total_duration_in_minutes DOUBLE"+
")"+
" PARTITIONED BY (batchid INT)"+
" ROW FORMAT DELIMITED"+
" FIELDS TERMINATED BY ','"+
" STORED AS TEXTFILE")
val insert_into_song_duration = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.song_duration"+
s" PARTITION (batchid=$batchId)"+
" SELECT"+
" e.user_id STRING,"+
" IF(e.user_id!=s.user_id"+
" OR (CAST(s.subscn_end_dt as BIGINT) < CAST(e.start_ts as BIGINT)),'unsubscribed','subscribed') AS user_type,"+
" e.song_id STRING,"+
" e.artist_id STRING,"+
" (cast(e.end_ts as BIGINT)-cast(e.start_ts as BIGINT))/60 AS total_duration_in_minutes"+
" FROM project.enriched_data e"+
" LEFT OUTER JOIN project.subscribed_users_1 s"+
" ON e.user_id=s.user_id"+
" WHERE e.status='pass'"+
s" AND (batchid=$batchId)")
6: TODO
Terminal
Event Log
49:1 CRLF: UTF-8:
Ant Build
SBT
Maven Projects
1: Project
2: Favorites
Z: Structure

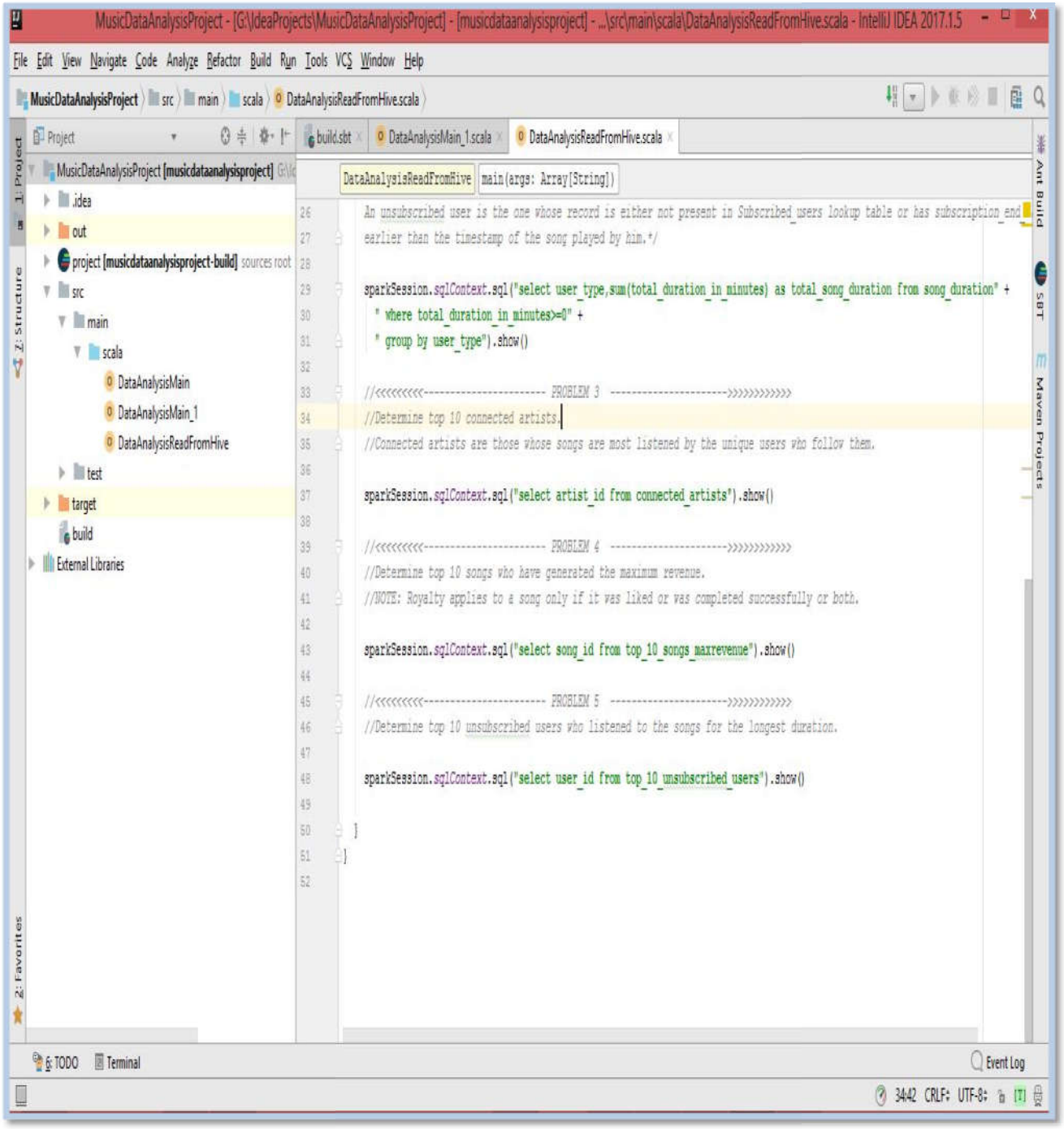

MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisMain_1.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisMain_1.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisMain_1
main(args: Array[String])
//<<<<<<<<<<---------- PROBLEM 3 - Creation of table and Insertion of data ---------->>>>>>>>>>
//Determine top 10 connected artists.
//Connected artists are those whose songs are most listened by the unique users who follow them.
val create_hive_table_top_10_connected_artists = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.connected_a
("+
" artist_id STRING,"+
" total_distinct_songs INT,"+
" unique_followers INT"+
")"+
" PARTITIONED BY (batchid INT)"+
" ROW FORMAT DELIMITED"+
" FIELDS TERMINATED BY ','"+
" STORED AS TEXTFILE")
val insert_into_top_10_connected_artists = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.connected_artists"+
s" PARTITION (batchid=$batchId)"+
" SELECT"+
" artist_id,"+
" COUNT(DISTINCT song_id) AS total_distinct_songs,"+
" COUNT(DISTINCT user_id) AS unique_followers"+
" FROM project.enriched_data"+
" WHERE status='pass'"+
s" AND (batchid=$batchId)"+
" GROUP BY artist_id"+
" ORDER BY unique_followers desc,total_distinct_songs desc"+
" LIMIT 10")
Ant Build
SBT
Maven Projects
1: Project
Z: Structure


MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisMain_1.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisMain_1.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisMain_1
main(args: Array[String])
//<<<<<<<<<<---------- PROBLEM 4 - Creation of table and Insertion of data ---------->>>>>>>>>>
//Determine top 10 songs who have generated the maximum revenue.
//NOTE: Royalty applies to a song only if it was liked or was completed successfully or both.
val create_hive_table_top_10_songs_maxrevenue = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_songs
("+
" song_id STRING,"+
" artist_id STRING,"+
" total_duration_in_minutes DOUBLE"+
")"+
" PARTITIONED BY (batchid INT)"+
" ROW FORMAT DELIMITED"+
" FIELDS TERMINATED BY ','"+
" STORED AS TEXTFILE")
val insert_into_top_10_songs_maxrevenue = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_songs_maxrevenu
s" PARTITION (batchid=$batchId)"+
" SELECT"+
" song_id,"+
" artist_id,"+
" (cast(end_ts as BIGINT)-cast(start_ts as BIGINT))/60 AS total_duration_in_minutes"+
" FROM project.enriched_data"+
" WHERE status='pass'" +
s" AND (batchid=$batchId)"+
" AND (like=1 OR song_end_type=0 OR (like=1 and song_end_type=0))"+
" ORDER BY total_duration_in_minutes desc"+
" LIMIT 10")
Ant Build
SBT
Maven Projects
1: Project
Z: Structure


MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisMain_1.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisMain_1.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisMain_1
main(args: Array[String])
//<<<<<<<<<---------- PROBLEM 5 - Creation of table and Insertion of data ---------->>>>>>>>>>
//Determine top 10 unsubscribed users who listened to the songs for the longest duration.
val create_hive_table_top_10_unsubscribed_users = sparkSession.sqlContext.sql("CREATE TABLE IF NOT EXISTS project.top_10_uns
"("+
" user_id STRING,"+
" song_id STRING,"+
" artist_id STRING,"+
" total_duration_in_minutes DOUBLE"+
")"+
" PARTITIONED BY (batchid INT)"+
" ROW FORMAT DELIMITED"+
" FIELDS TERMINATED BY ','"+
" STORED AS TEXTFILE")
val insert_into_unsubscribed_users = sparkSession.sqlContext.sql("INSERT OVERWRITE TABLE project.top_10_unsubscribed_users"+
s" PARTITION (batchid=$batchId)"+
" SELECT"+
" user_id,"+
" song_id,"+
" artist_id,"+
" total_duration_in_minutes"+
" FROM project.song_duration"+
" WHERE user_type='unsubscribed'"+
" AND total_duration_in_minutes>=0"+
s" AND (batchid=$batchId)"+
" ORDER BY total_duration_in_minutes desc"+
" LIMIT 10")
}
}
6: TODO
Terminal
Event Log
171:34 CRLF: UTF-8:
Ant Build
SBT
Maven Projects
1: Project
2: Favorites
Z: Structure

Spark Code to fetch data from hive tables [DataAnalysisReadFromHive.scala]

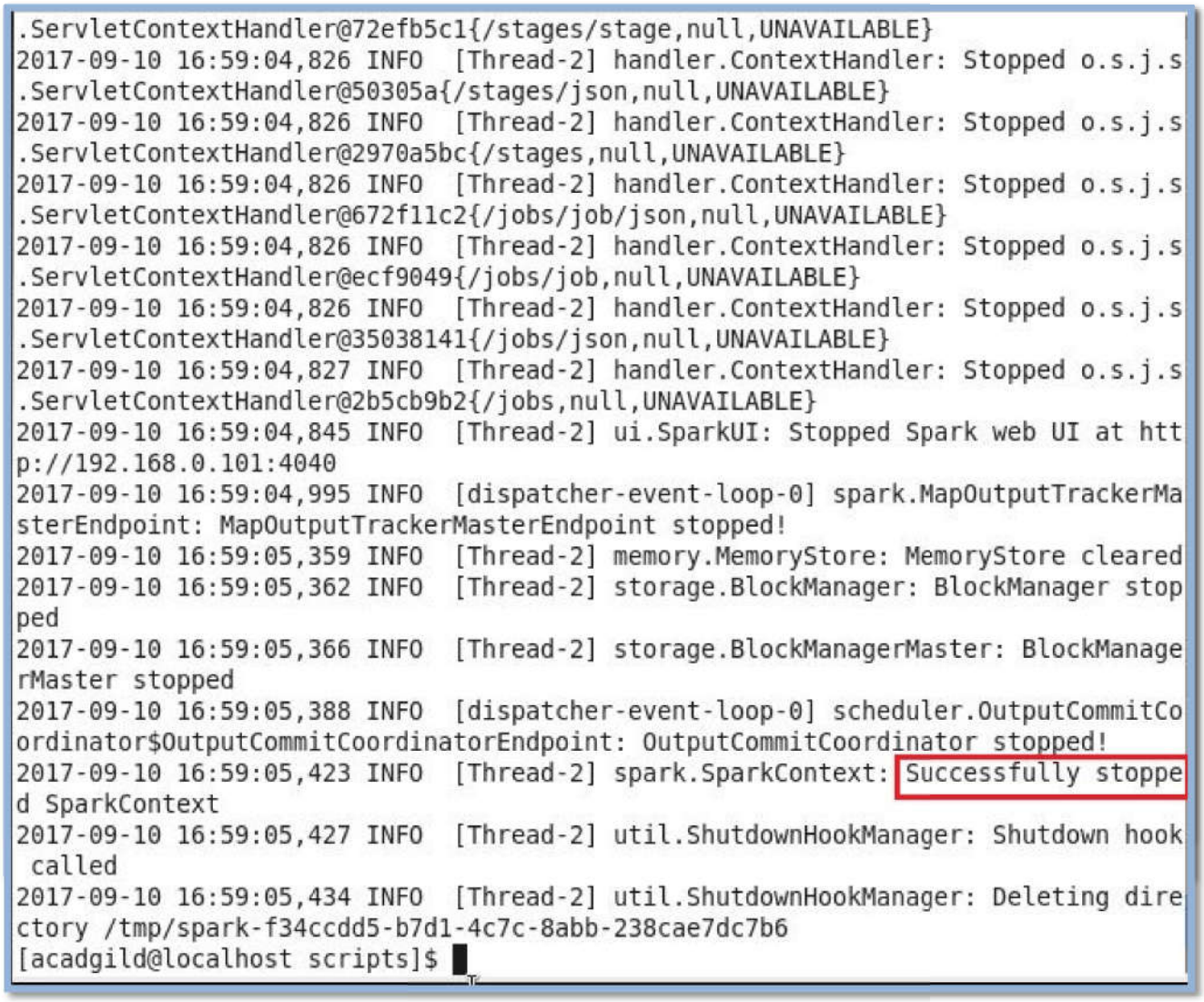



MusicDataAnalysisProject - [G:\IdeaProjects\MusicDataAnalysisProject] - [musicdataanalysisproject] - ...\src\main\scala\DataAnalysisReadFromHive.scala - IntelliJ IDEA 2017.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MusicDataAnalysisProject src main scala DataAnalysisReadFromHive.scala
Project
MusicDataAnalysisProject [musicdataanalysisproject]
.idea
out
project [musicdataanalysisproject-build] sources root
src
main
scala
DataAnalysisMain
DataAnalysisMain_1
DataAnalysisReadFromHive
test
target
build
External Libraries
build.sbt
DataAnalysisMain_1.scala
DataAnalysisReadFromHive.scala
DataAnalysisReadFromHive
main(args: Array[String])
An unsubscribed user is the one whose record is either not present in Subscribed_users lookup table or has subscription end
earlier than the timestamp of the song played by him.*/
sparkSession.sqlContext.sql("select user_type,sum(total_duration_in_minutes) as total_song_duration from song_duration" +
" where total_duration_in_minutes>=0" +
" group by user_type").show()
//<<<<<<<<<<---------- PROBLEM 3 ---------->>>>>>>>>>
//Determine top 10 connected artists.
//Connected artists are those whose songs are most listened by the unique users who follow them.
sparkSession.sqlContext.sql("select artist_id from connected_artists").show()
//<<<<<<<<<<---------- PROBLEM 4 ---------->>>>>>>>>>
//Determine top 10 songs who have generated the maximum revenue.
//NOTE: Royalty applies to a song only if it was liked or was completed successfully or both.
sparkSession.sqlContext.sql("select song_id from top_10_songs_maxrevenue").show()
//<<<<<<<<<<---------- PROBLEM 5 ---------->>>>>>>>>>
//Determine top 10 unsubscribed users who listened to the songs for the longest duration.
sparkSession.sqlContext.sql("select user_id from top_10_unsubscribed_users").show()
}
}
Ant Build
SBT
Maven Projects
1: Project
Z: Structure
2: Favorites
6: TODO
Terminal
Event Log
34:42 CRLF: UTF-8:

Below screenshot shows execution of data_analysis_spark.sh

```
        zip warning: name not matched: META-INF/*.RSA
        zip warning: name not matched: META-INF/*.DSA
        zip warning: name not matched: META-INF/*.SF

zip error: Nothing to do! (/home/acadgild/project/MusicDataAnalysisProject/out/a
rtifacts/MusicDataAnalysisProject_jar3/MusicDataAnalysisProject.jar)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/spark-2.0.0-bin-hadoop2.6/jars/slf4
j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-09-10 16:56:23,625 INFO  [main] spark.SparkContext: Running Spark version 2
.0.0
2017-09-10 16:56:25,629 WARN  [main] util.NativeCodeLoader: Unable to load nativ
e-hadoop library for your platform... using builtin-java classes where applicabl
e
2017-09-10 16:56:26,377 WARN  [main] util.Utils: Your hostname, localhost.locald
omain resolves to a loopback address: 127.0.0.1; using 192.168.0.101 instead (on
 interface eth5)
2017-09-10 16:56:26,380 WARN  [main] util.Utils: Set SPARK_LOCAL_IP if you need
to bind to another address
```

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


.ServletContextHandler@72efb5c1{/stages/stage,null,UNAVAILABLE}
2017-09-10 16:59:04,826 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@50305a{/stages/json,null,UNAVAILABLE}
2017-09-10 16:59:04,826 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@2970a5bc{/stages,null,UNAVAILABLE}
2017-09-10 16:59:04,826 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@672f11c2{/jobs/job/json,null,UNAVAILABLE}
2017-09-10 16:59:04,826 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@ecf9049{/jobs/job,null,UNAVAILABLE}
2017-09-10 16:59:04,826 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@35038141{/jobs/json,null,UNAVAILABLE}
2017-09-10 16:59:04,827 INFO [Thread-2] handler.ContextHandler: Stopped o.s.j.s
.ServletContextHandler@2b5cb9b2{/jobs,null,UNAVAILABLE}
2017-09-10 16:59:04,845 INFO [Thread-2] ui.SparkUI: Stopped Spark web UI at htt
p://192.168.0.101:4040
2017-09-10 16:59:04,995 INFO [dispatcher-event-loop-0] spark.MapOutputTrackerMa
sterEndpoint: MapOutputTrackerMasterEndpoint stopped!
2017-09-10 16:59:05,359 INFO [Thread-2] memory.MemoryStore: MemoryStore cleared
2017-09-10 16:59:05,362 INFO [Thread-2] storage.BlockManager: BlockManager stop
ped
2017-09-10 16:59:05,366 INFO [Thread-2] storage.BlockManagerMaster: BlockManage
rMaster stopped
2017-09-10 16:59:05,388 INFO [dispatcher-event-loop-0] scheduler.OutputCommitCo
ordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
2017-09-10 16:59:05,423 INFO [Thread-2] spark.SparkContext: Successfully stoppe
d SparkContext
2017-09-10 16:59:05,427 INFO [Thread-2] util.ShutdownHookManager: Shutdown hook
called
2017-09-10 16:59:05,434 INFO [Thread-2] util.ShutdownHookManager: Deleting dire
ctory /tmp/spark-f34ccdd5-b7d1-4c7c-8abb-238cae7dc7b6
[acadgild@localhost scripts]$

Below screenshots show the output of hive tables using spark

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

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

Output of Problem 1

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

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

Output of Problem 2

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

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

Output of Problem 3

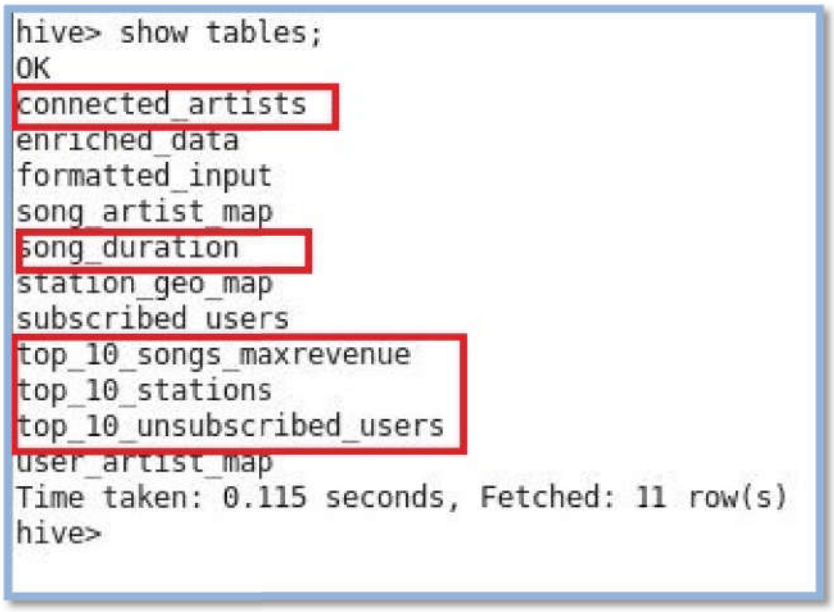
```
inished task 0.0 in stage 6.0 (TID 203) in 155 ms on localhost (1/1)
2017-09-10 18:21:41,756 INFO  [task-result-getter-3] scheduler.TaskSchedulerImpl
: Removed TaskSet 6.0, whose tasks have all completed, from pool
+-------+
|song_id|
+-------+
|   S209|
|   S202|
|   S205|
|   S200|
|   S203|
|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

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

Output of Problem 4

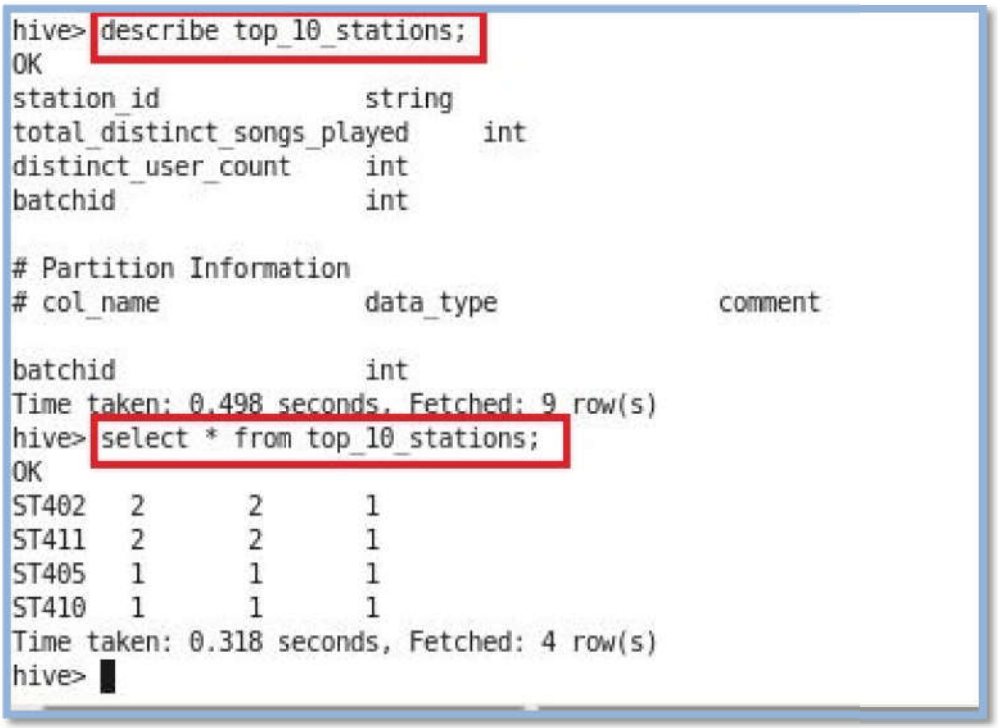
```
2017-09-10 18:21:43,240 INFO  [task-result-getter-0] scheduler.TaskSetManager: F
inished task 0.0 in stage 7.0 (TID 204) in 140 ms on localhost (1/1)
2017-09-10 18:21:43,241 INFO  [task-result-getter-0] scheduler.TaskSchedulerImpl
: Removed TaskSet 7.0, whose tasks have all completed, from pool
+-------+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

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

Output of Problem 5

Below screenshot shows that for all 5 problems of data analysis, five tables created successfully inside hive



Output of Problem 1 shown in hive



Output of Problem 2 shown in hive

```
hive> describe song_duration;
OK
user_id                 string
user_type               string
song_id                 string
artist_id               string
total_duration_in_minutes       double
batchid                 int

# Partition Information
# col_name              data_type               comment

batchid                 int
Time taken: 0.528 seconds, Fetched: 11 row(s)
hive> select * from song_duration;
OK
U115    subscribed      S200    A300    -480116.76666666666     1
U108    subscribed      S200    A302    87193.78333333334       1
U120    subscribed      S201    A300    43405.55        1
U107    unsubscribed    S202    A304    -87193.78333333334      1
U101    subscribed      S202    A300    166666.66666666666      1
U110    unsubscribed    S202    A303    -436711.2166666667      1
U118    subscribed      S202    A300    0.0     1
U104    subscribed      S202    A303    -166666.66666666666     1
U102    subscribed      S203    A305    480116.76666666666      1
U113    subscribed      S203    A304    -43788.23333333333      1
U113    subscribed      S203    A303    436711.2166666667       1
U105    subscribed      S203    A303    -1666.6666666666667     1
U113    subscribed      S203    A303    0.0     1
U103    unsubscribed    S204    A300    -480116.76666666666     1
U108    subscribed      S205    A304    166666.66666666666      1
U106    subscribed      S206    A300    -43788.23333333333      1
U120    subscribed      S206    A303    -333333.3333333333      1
U105    unsubscribed    S207    A300    -333333.3333333333      1
U116    subscribed      S208    A303    -166666.66666666666     1
U114    subscribed      S209    A303    523905.0        1
Time taken: 0.234 seconds, Fetched: 20 row(s)
hive>
```

```
hive> select user_type,sum(total_duration_in_minutes) as total_duration from song_duratio
n where total_duration_in_minutes>=0 group by user_type;
Query ID = acadgild_20170911192222_ea6ab2c6-3e67-45f2-b6a5-ac665374a198
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1505127197818_0002, Tracking URL = http://localhost:8088/proxy/applica
tion_1505127197818_0002/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1505127197818_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-09-11 19:23:19,590 Stage-1 map = 0%,  reduce = 0%
2017-09-11 19:23:57,839 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 7.36 sec
2017-09-11 19:24:36,617 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.67 sec
MapReduce Total cumulative CPU time: 11 seconds 670 msec
Ended Job = job_1505127197818_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.01 sec   HDFS Read: 1102 HDFS Write
: 30 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 10 msec
OK
subscribed      1904665.6500000001
Time taken: 123.055 seconds, Fetched: 1 row(s)
hive>
```

Output of Problem 2

Output of Problem 3 in Hive

```
hive> describe top_10_songs_maxrevenue
    > ;
OK
song_id                 string
artist_id               string
total_duration_in_minutes       double
batchid                 int

# Partition Information
# col_name              data_type               comment

batchid                 int
Time taken: 0.512 seconds, Fetched: 9 row(s)
hive> select * from top_10_songs_maxrevenue;
OK
S209    A303    523905.0        1
S202    A300    166666.66666666666      1
S205    A304    166666.66666666666      1
S200    A302    87193.78333333334       1
S203    A303    0.0     1
S203    A304    -43788.23333333333      1
S206    A300    -43788.23333333333      1
S202    A304    -87193.78333333334      1
S206    A303    -333333.3333333333      1
S202    A303    -436711.2166666667      1
Time taken: 0.257 seconds, Fetched: 10 row(s)
hive>
```

Output of Problem 4 in Hive

```
hive> describe song_duration;
OK
user_id                 string
user_type               string
song_id                 string
artist_id               string
total_duration_in_minutes       double
batchid                 int

# Partition Information
# col_name              data_type               comment

batchid                 int
Time taken: 0.528 seconds, Fetched: 11 row(s)
hive> select * from song_duration;
OK
U115    subscribed      S200    A300    -480116.76666666666     1
U108    subscribed      S200    A302    87193.78333333334       1
U120    subscribed      S201    A300    43405.55        1
U107    unsubscribed    S202    A304    -87193.78333333334      1
U101    subscribed      S202    A300    166666.66666666666      1
U110    unsubscribed    S202    A303    -436711.2166666667      1
U118    subscribed      S202    A300    0.0     1
U104    subscribed      S202    A303    -166666.66666666666     1
U102    subscribed      S203    A305    480116.76666666666      1
U113    subscribed      S203    A304    -43788.23333333333      1
U113    subscribed      S203    A303    436711.2166666667       1
U105    subscribed      S203    A303    -1666.6666666666667     1
U113    subscribed      S203    A303    0.0     1
U103    unsubscribed    S204    A300    -480116.76666666666     1
U108    subscribed      S205    A304    166666.66666666666      1
U106    subscribed      S206    A300    -43788.23333333333      1
U120    subscribed      S206    A303    -333333.3333333333      1
U105    unsubscribed    S207    A300    -333333.3333333333      1
U116    subscribed      S208    A303    -166666.66666666666     1
U114    subscribed      S209    A303    523905.0        1
Time taken: 0.234 seconds, Fetched: 20 row(s)
hive>
```

Output of Problem 5 in Hive

```
hive> describe top_10_songs_maxrevenue
    > ;
OK
song_id                 string
artist_id               string
total_duration_in_minutes       double
batchid                 int

# Partition Information
# col_name              data_type               comment

batchid                 int
Time taken: 0.512 seconds, Fetched: 9 row(s)
hive> select * from top_10_songs_maxrevenue;
OK
S209    A303    523905.0        1
S202    A300    166666.66666666666      1
S205    A304    166666.66666666666      1
S200    A302    87193.78333333334       1
S203    A303    0.0     1
S203    A304    -43788.23333333333      1
S206    A300    -43788.23333333333      1
S202    A304    -87193.78333333334      1
S206    A303    -333333.3333333333      1
S202    A303    -436711.2166666667      1
Time taken: 0.257 seconds, Fetched: 10 row(s)
hive>
```

8. **data_anlaysis_spark.sh calls data_export.sh screenshot of which is as follows:**

data_export.sh

```
#!/bin/bash

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

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

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

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

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

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

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

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

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

Above script calls create_schema.sql script, which creates Mysql tables if they don't exist

create_schema.sql

```
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 song_duration
(
user_id VARCHAR(50),
user_type VARCHAR(50),
song_id VARCHAR(50),
artist_id VARCHAR(50),
total_duration DOUBLE
);

CREATE TABLE IF NOT EXISTS connected_artists
(
artist_id VARCHAR(50),
total_distinct_songs INT,
unique_followers INT
);
```

```
CREATE TABLE IF NOT EXISTS top_10_songs_maxrevenue
(
song_id VARCHAR(50),
artist_id VARCHAR(50),
total_duration DOUBLE
);

CREATE TABLE IF NOT EXISTS top_10_unsubscribed_users
(
user_id VARCHAR(50),
song_id VARCHAR(50),
artist_id VARCHAR(50),
total_duration DOUBLE
);
```

Below screenshots show that project database and 5 tables got created successfully in mysql

```
[acadgild@localhost ~]$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 63
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| acadDB1            |
| b1                 |
| metastore          |
| mysql              |
| project            |
| test               |
+--------------------+
7 rows in set (0.19 sec)

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

Database changed
mysql>
```

```
mysql> use project;
Database changed
mysql> show tables;
+---------------------------+
| Tables_in_project         |
+---------------------------+
| connected_artists         |
| song_duration             |
| top_10_songs_maxrevenue   |
| top_10_stations           |
| top_10_unsubscribed_users |
+---------------------------+
5 rows in set (0.00 sec)

mysql>
```

Below screenshots show the execution of data_export.sh

```
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail
.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fai
l.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2017-09-10 22:36:59,339 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2017-09-10 22:37:08,667 INFO  [main] manager.MySQLManager: Preparing to use a My
SQL streaming resultset.
2017-09-10 22:37:08,667 INFO  [main] tool.CodeGenTool: Beginning code generation
2017-09-10 22:37:10,036 INFO  [main] manager.SqlManager: Executing SQL statement
: SELECT t.* FROM `top_10_stations` AS t LIMIT 1
2017-09-10 22:37:10,243 INFO  [main] manager.SqlManager: Executing SQL statement
: SELECT t.* FROM `top_10_stations` AS t LIMIT 1
2017-09-10 22:37:10,289 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME
is /usr/local/hadoop-2.6.0
Note: /tmp/sqoop-acadgild/compile/c69fe2436b4a7912cbfef65beade6926/top_10_statio
ns.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
2017-09-10 22:37:21,942 INFO  [main] orm.CompilationManager: Writing jar file: /
tmp/sqoop-acadgild/compile/c69fe2436b4a7912cbfef65beade6926/top_10_stations.jar
2017-09-10 22:37:21,998 INFO  [main] mapreduce.ExportJobBase: Beginning export o
f top_10_stations
```

```
.value.class is deprecated. Instead, use mapreduce.map.output.value.class
2017-09-10 22:37:37,761 INFO  [main] Configuration.deprecation: mapred.mapoutput
.key.class is deprecated. Instead, use mapreduce.map.output.key.class
2017-09-10 22:37:37,761 INFO  [main] Configuration.deprecation: mapred.job.class
path.files is deprecated. Instead, use mapreduce.job.classpath.files
2017-09-10 22:37:37,761 INFO  [main] Configuration.deprecation: user.name is dep
recated. Instead, use mapreduce.job.user.name
2017-09-10 22:37:37,761 INFO  [main] Configuration.deprecation: mapred.reduce.ta
sks is deprecated. Instead, use mapreduce.job.reduces
2017-09-10 22:37:37,762 INFO  [main] Configuration.deprecation: mapred.cache.fil
es.filesizes is deprecated. Instead, use mapreduce.job.cache.files.filesizes
2017-09-10 22:37:39,871 INFO  [main] mapreduce.JobSubmitter: Submitting tokens f
or job: job_1505031035214_0015
2017-09-10 22:37:44,634 INFO  [main] impl.YarnClientImpl: Submitted application
application_1505031035214_0015 to ResourceManager at /0.0.0.0:8032
2017-09-10 22:37:46,603 INFO  [main] mapreduce.Job: The url to track the job: ht
tp://http://localhost:8088/proxy/application_1505031035214_0015/
2017-09-10 22:37:46,609 INFO  [main] mapreduce.Job: Running job: job_15050310352
14_0015
2017-09-10 22:38:33,585 INFO  [main] mapreduce.Job: Job job_1505031035214_0015 r
unning in uber mode : false
2017-09-10 22:38:33,946 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2017-09-10 22:39:05,104 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2017-09-10 22:39:11,458 INFO  [main] mapreduce.Job: Job job_1505031035214_0015 c
ompleted successfully
```

Below screenshots show that data is inserted successfully from hive tables to mysql tables successfully

```
mysql> select * from connected_artists;
+-----------+----------------------+------------------+
| artist_id | total_distinct_songs | unique_followers |
+-----------+----------------------+------------------+
| A300      |                    6 |                7 |
| A303      |                    5 |                7 |
| A304      |                    3 |                3 |
| A305      |                    1 |                1 |
| A302      |                    1 |                1 |
+-----------+----------------------+------------------+
5 rows in set (0.00 sec)

mysql> select * from top_10_songs_maxrevenue;
+---------+-----------+-------------------+
| song_id | artist_id | total_duration    |
+---------+-----------+-------------------+
| S209    | A303      |            523905 |
| S202    | A300      |  166666.666666667 |
| S205    | A304      |  166666.666666667 |
| S200    | A302      |  87193.7833333333 |
| S203    | A303      |                 0 |
| S203    | A304      | -43788.2333333333 |
| S206    | A300      | -43788.2333333333 |
| S202    | A304      | -87193.7833333333 |
| S206    | A303      | -333333.333333333 |
| S202    | A303      | -436711.216666667 |
+---------+-----------+-------------------+
10 rows in set (0.00 sec)

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

mysql>
```

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

mysql> select * from song_duration;
+---------+--------------+---------+-----------+-------------------+
| user_id | user_type    | song_id | artist_id | total_duration    |
+---------+--------------+---------+-----------+-------------------+
| U115    | subscribed   | S200    | A300      | -480116.766666667 |
| U108    | subscribed   | S200    | A302      |  87193.7833333333 |
| U120    | subscribed   | S201    | A300      |          43405.55 |
| U107    | unsubscribed | S202    | A304      | -87193.7833333333 |
| U101    | subscribed   | S202    | A300      |  166666.666666667 |
| U110    | unsubscribed | S202    | A303      | -436711.216666667 |
| U118    | subscribed   | S202    | A300      |                 0 |
| U104    | subscribed   | S202    | A303      | -166666.666666667 |
| U102    | subscribed   | S203    | A305      |  480116.766666667 |
| U113    | subscribed   | S203    | A304      | -43788.2333333333 |
| U113    | subscribed   | S203    | A303      |  436711.216666667 |
| U105    | subscribed   | S203    | A303      | -1666.66666666667 |
| U113    | subscribed   | S203    | A303      |                 0 |
| U103    | unsubscribed | S204    | A300      | -480116.766666667 |
| U108    | subscribed   | S205    | A304      |  166666.666666667 |
| U106    | subscribed   | S206    | A300      | -43788.2333333333 |
| U120    | subscribed   | S206    | A303      | -333333.333333333 |
| U105    | unsubscribed | S207    | A300      | -333333.333333333 |
| U116    | subscribed   | S208    | A303      | -166666.666666667 |
| U114    | subscribed   | S209    | A303      |            523905 |
+---------+--------------+---------+-----------+-------------------+
20 rows in set (0.00 sec)

mysql> select * from top_10_stations;
+------------+-----------------------------+---------------------+
| station_id | total_distinct_songs_played | distinct_user_count |
+------------+-----------------------------+---------------------+
| ST402      |                           2 |                   2 |
| ST411      |                           2 |                   2 |
| ST405      |                           1 |                   1 |
| ST410      |                           1 |                   1 |
+------------+-----------------------------+---------------------+
4 rows in set (0.07 sec)

mysql>
```

After successful export of data to mysql, below commands present in data_analysis_spark.sh get executed:

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

batchid=`expr $batchid + 1`

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

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

Batchid incremented from 1 to 2

```
[acadgild@localhost project]$ cd logs
[acadgild@localhost logs]$ ls -lrt
total 60
-rwxrwxrwx. 1 acadgild acadgild  1803 Sep 10 23:33 log_batch_1
-rwxrwxrwx. 1 acadgild acadgild     1 Sep 10 23:33 current-batch.txt
-rwxrwxrwx. 1 acadgild acadgild 45608 Sep 11 20:57 log_batch_2
[acadgild@localhost logs]$
```

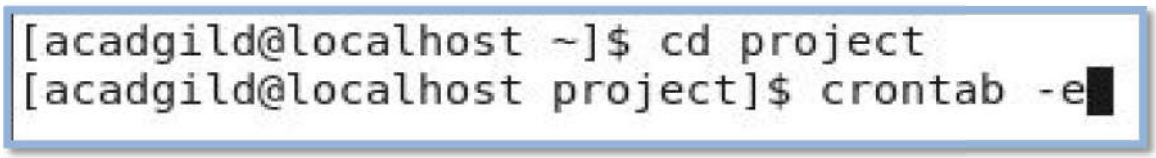
New log file is created after incrementing batchid

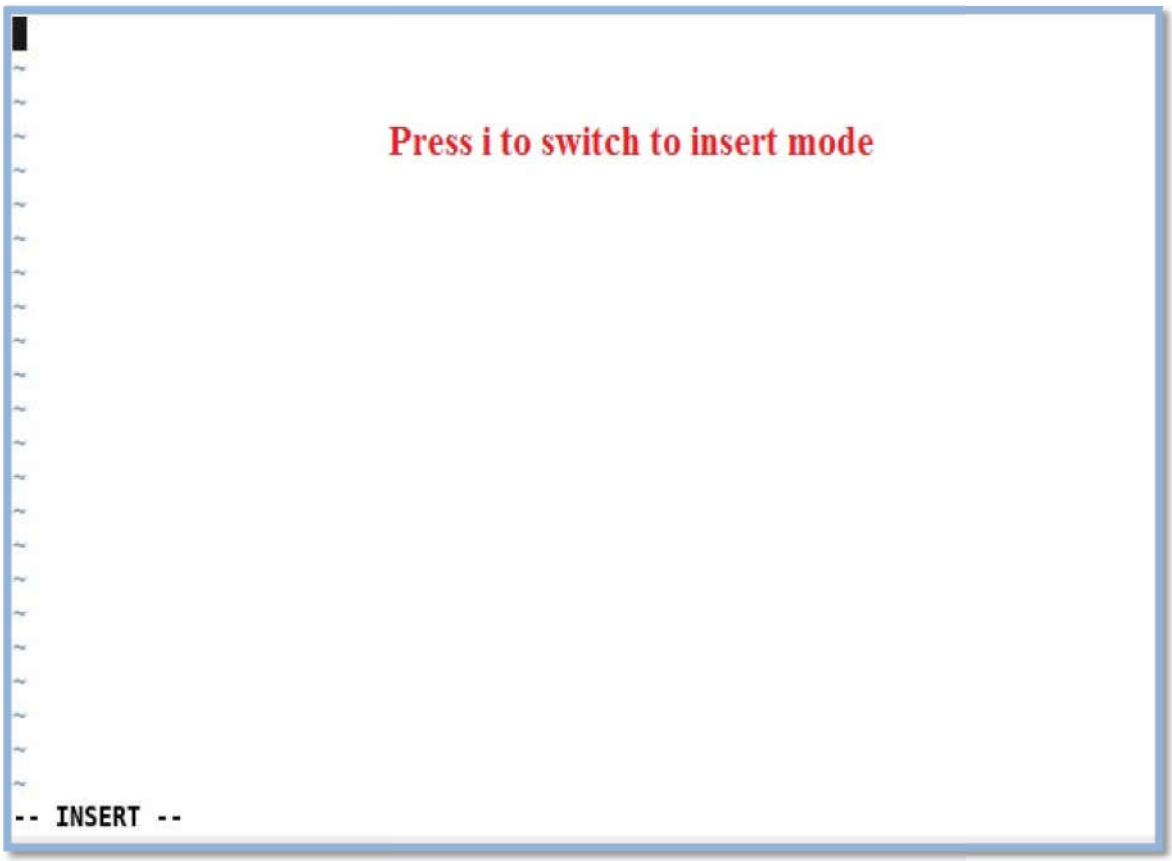
Contents of log_batch_1 file

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 spark script for data analysis...
Exporting data to mysql...
Creating mysql tables if not present...
Running sqoop job for data export...
Incrementing batchid...
```

9. **To automate the whole work, i.e. to run all scripts automatically after 3 hours, crontab job is created. Below screenshots show how crontab job is created:**
   Crontab –e is used to create crontab job





```
* */3 * * * /home/acadgild/project/scripts/wrapper.sh >> /home/acadgild/project/scripts/j
obScheduling.log
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

job which is to be run automatically after 3 hours

log information is sent to jobScheduling.log file as and when crontab job is run

type this to save and exit

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

this command lists crontab job

Future work of this Project

In this project, formatting, cleaning, validation steps are done using hive and pig and analysis part is done in spark.

So for future work of this project, formatting, cleaning and validation also can be done in spark.