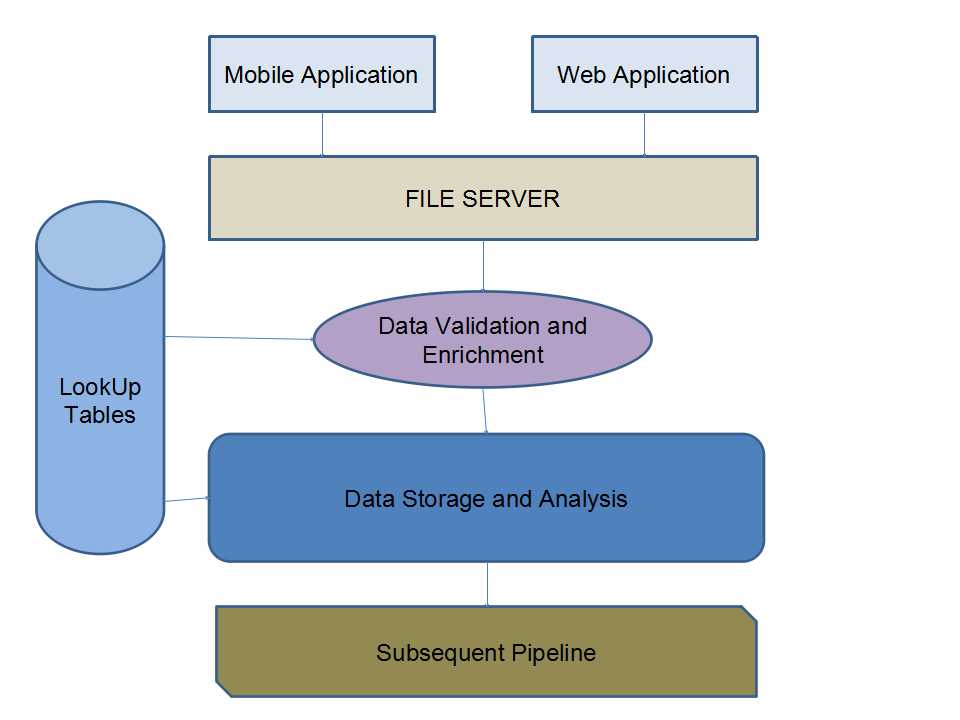
**MUSIC DATA ANALYSIS**

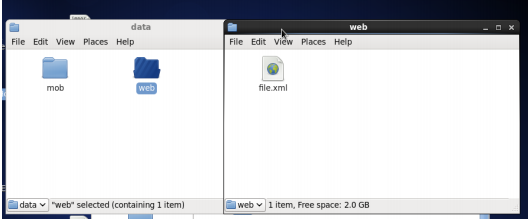
**PROJECT-2**

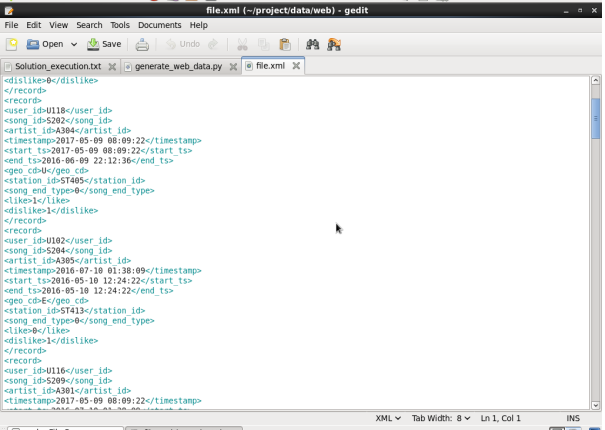
*Flow of Operations:*



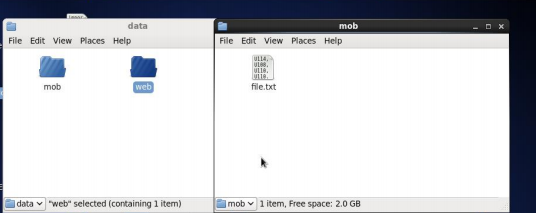
DATASET:

1. Data coming from web applications reside in /data/web and has xml format.

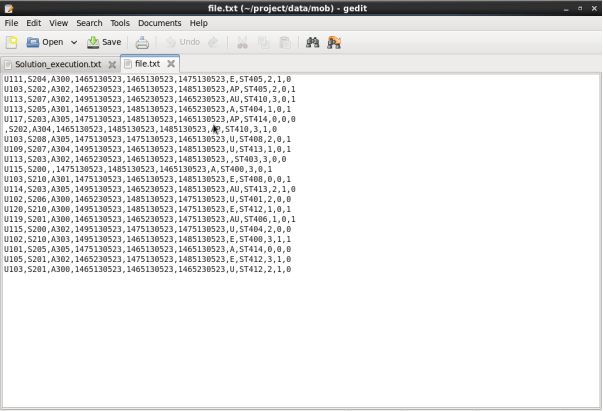




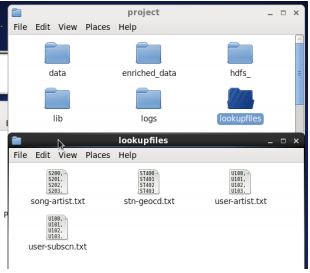
1. Data coming from mobile applications reside in /data/mob and has csv format.



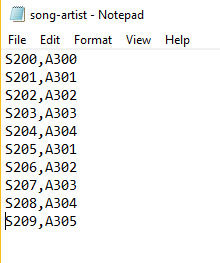
Below is the screenshot of data in the file:



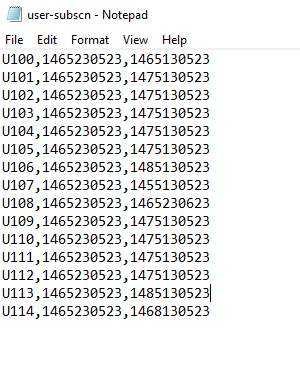
1. Data present in lookup directory should be used in HBase.



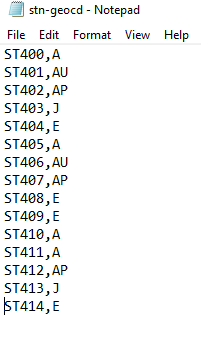
* *Columns*: song\_id,artist\_id



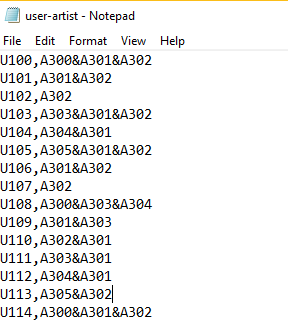
* *Columns:* user\_id,subscn\_start\_dt, subscn\_end\_dt



* *Columns:* station\_id, geo\_cd



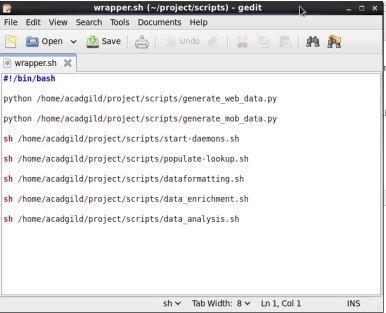
* *Columns*: user\_id, artists\_array



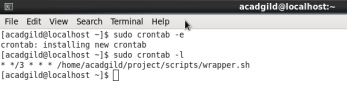
Once the data to be loaded is available, below steps will be performed:

* ***Scheduling Cron Tab:***

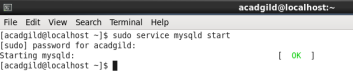
Below are the contents of wrapper.sh file used for scheduling data for analysis.



Scheduling the job to run every 3 hous via the command in the below screenshot.

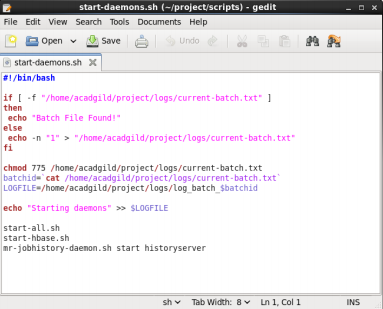


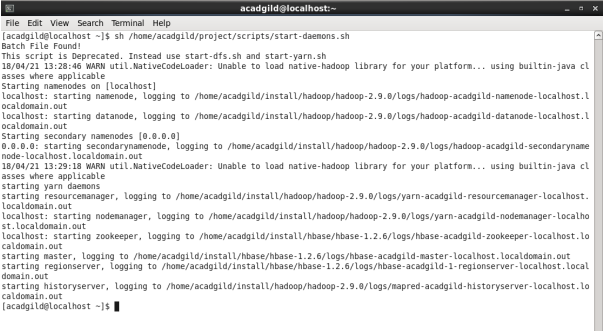
* ***Launching all the services and daemons required for execution***



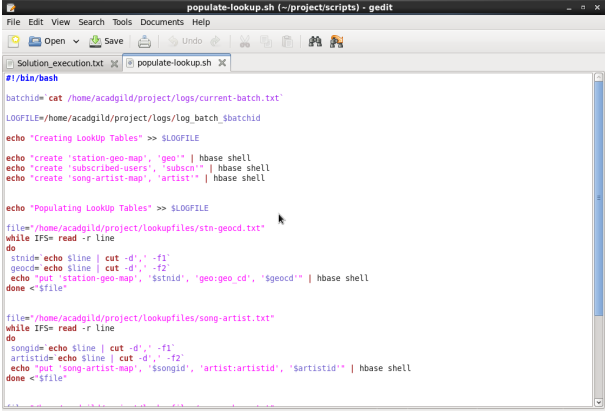
Below is the *start-daemons.sh* file screenshot. Executing this file would in turn starts up the historyserver, dfs, yarn and other required services.

* *Current-batch.sh* is the file which contains batch id.
* Chmod 775 provides all the necessary permissions to current-batch.sh file.
* Echo command writes the string to the Logfile.



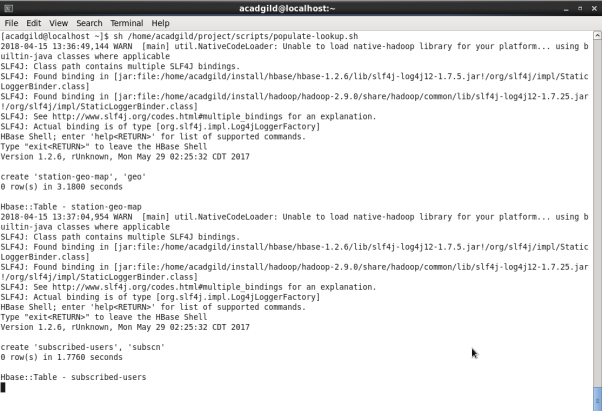


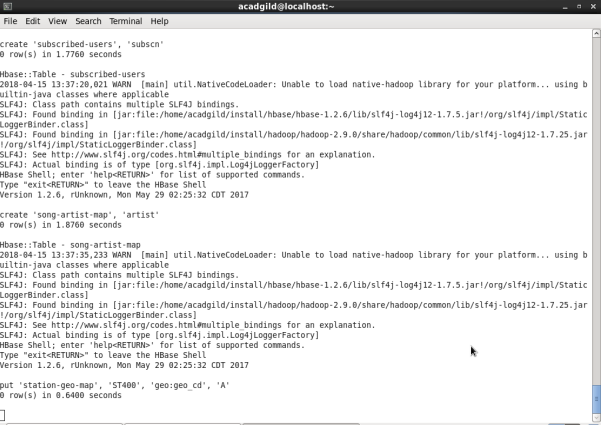
* ***Populate tables in hbase via populate-lookup.sh:***

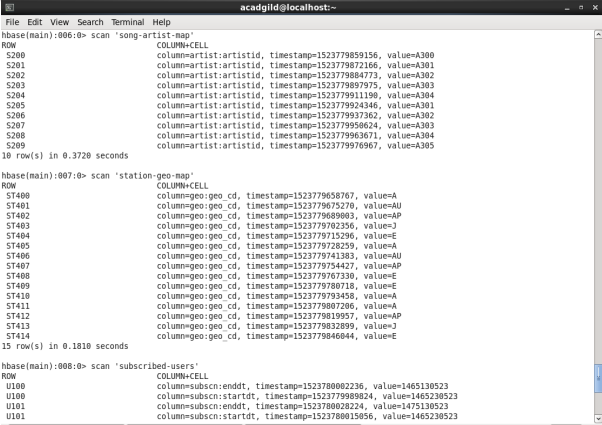


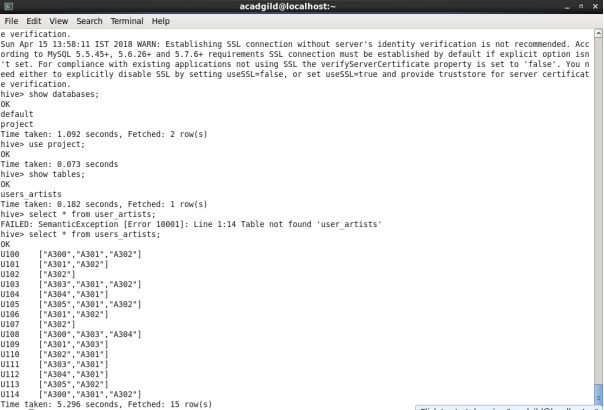
The following operations are performed:

* Get the batch id number from the batch file and get the Log File for the batch using the batch id.
* Create the HBase tables for the lookup data files: *song-artist, stn-geocd and user-subscn* with their column families.
* For every lookup data file, read each line, extract the columns (comma separated) and add the data as rows to the corresponding HBase tables.
* Run the hive script user-artist.hql. This will populate a hive table with the data in the lookup data file *user-artist*. This is because this file has an array column that is difficult to populate in HBase.

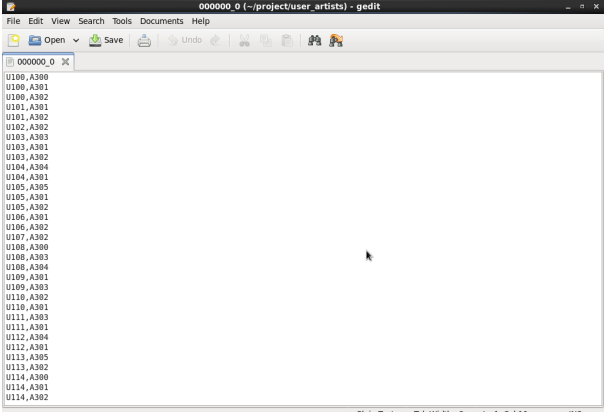








Output of hive command:

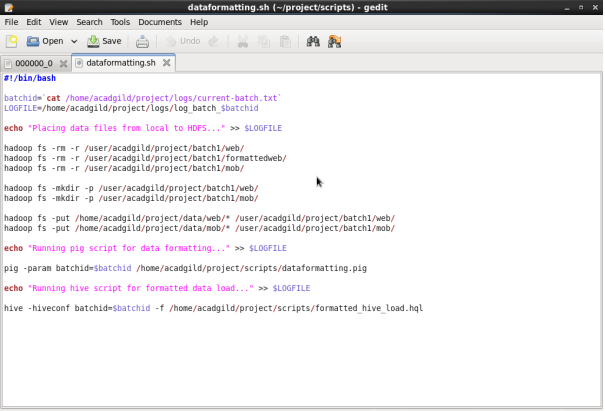


* ***Perform Data Formatting:***

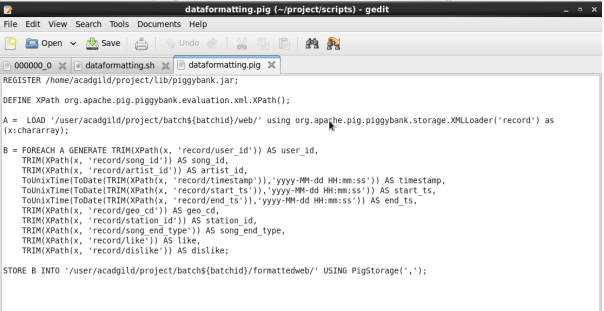
Following operations are performed in dataformatting.sh :

* Fetching batch id from the current\_batch file.
* Writing to the log file about process start.
* Checking if the directory is present in HDFS if yes, then deleting it and creating new blank directory.
* Copying mob and formatted web data to HDFS from local.
* Run pig script dataformatting.pig, to format web data stored in HDFS as xml to csv format and store formatted file to formattedweb directory.
* Run hive script formatted\_hive\_load.hql to load data from mob and formattedweb directory to hive tables.

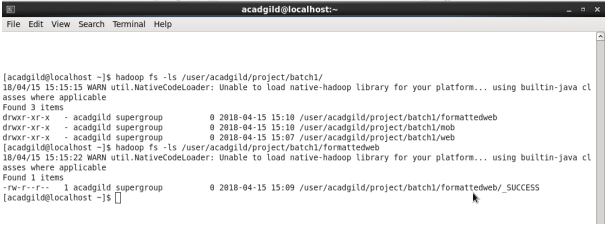
***DataFormatting.sh***

******

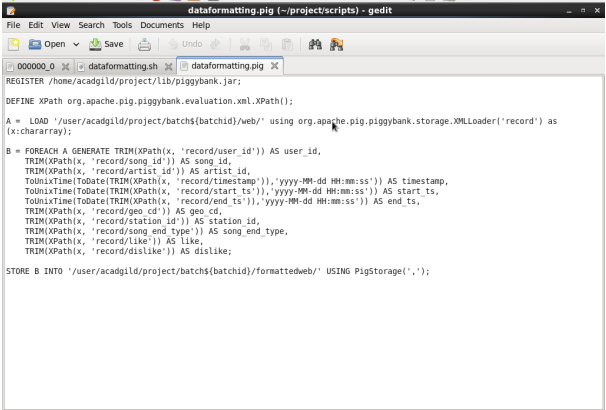
***DataFormatting.pig***

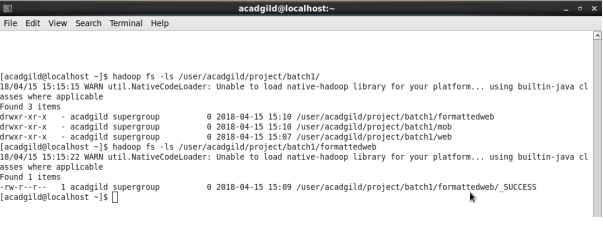
******

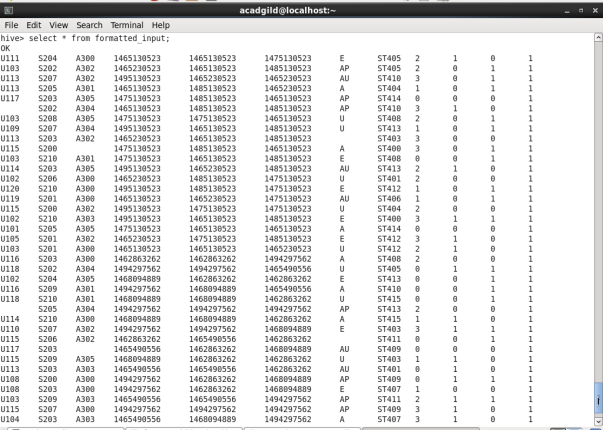
***OUTPUT:***

******

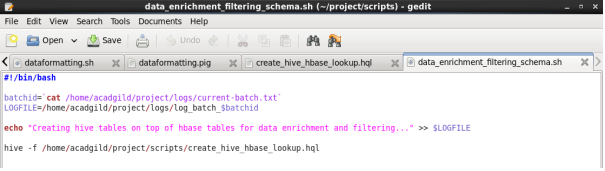
***Formatted\_hive\_load.hql***

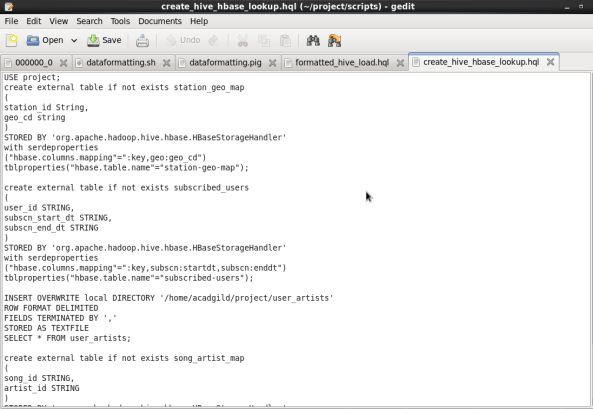
******

******

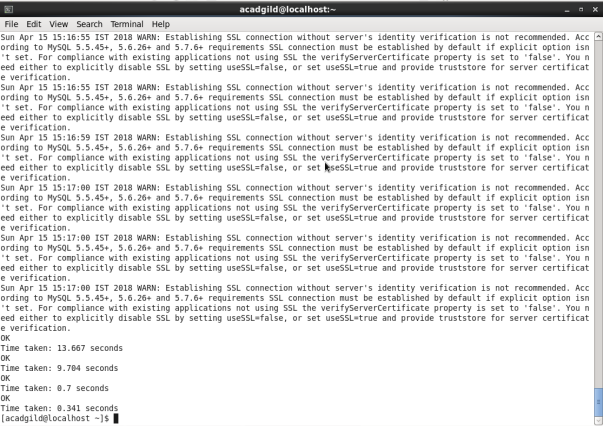
******

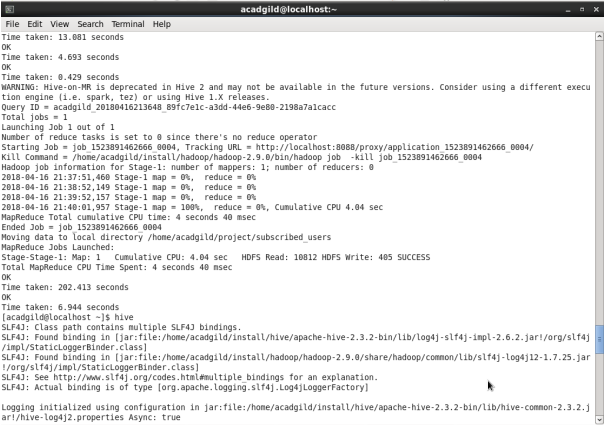
Run the hive script create\_hive\_hbase\_lookup.hql. This will create the lookup tables in Hive and import the data from the HBase lookup tables to the Hive lookup tables.

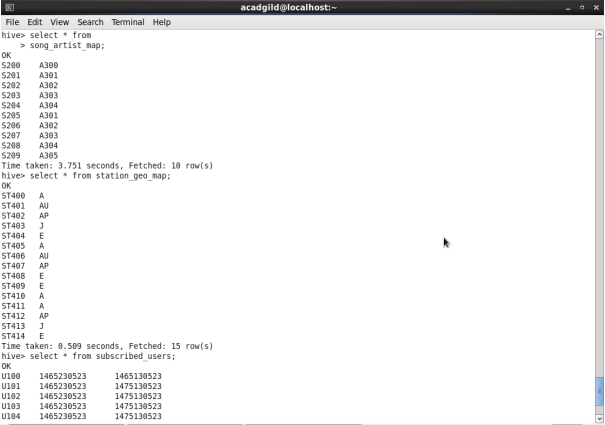
******

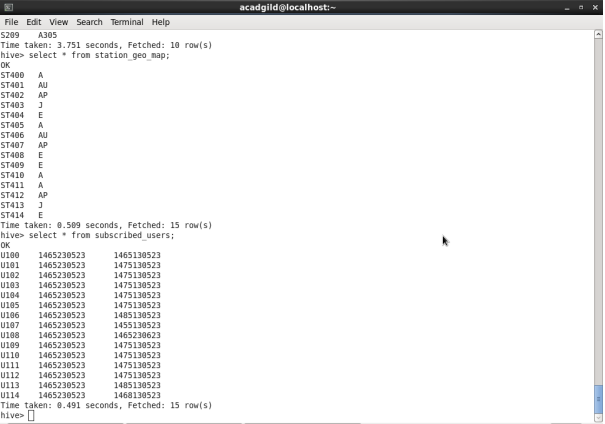
******

***OUTPUT:***

******

******

******

******

***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,

u\_Timestamp STRING,

Start\_ts STRING,

End\_ts STRING,

Geo\_cd STRING,

Station\_id STRING,

Song\_end\_type INT,

u\_Like INT,

Dislike INT

)

PARTITIONED BY

(batchid INT,

status STRING)

STORED AS ORC;

INSERT OVERWRITE TABLE enriched\_data

PARTITION (batchid, status)

SELECT

i.user\_id,

i.song\_id,

sa.artist\_id,

i.u\_timestamp,

i.start\_ts,

i.end\_ts,

sg.geo\_cd,

i.station\_id,

IF (i.song\_end\_type IS NULL, 3, i.song\_end\_type) AS song\_end\_type,

IF (i.u\_like IS NULL, 0, i.u\_like) AS u\_like,

IF (i.dislike IS NULL, 0, i.dislike) AS dislike,

i.batchid,

IF((i.u\_like=1 AND i.dislike=1)

OR i.user\_id IS NULL

OR i.song\_id IS NULL

OR i.u\_timestamp IS NULL

OR i.start\_ts IS NULL

OR i.end\_ts IS NULL

OR i.geo\_cd IS NULL

OR i.user\_id=''

OR i.song\_id=''

OR i.u\_timestamp=''

OR i.start\_ts=''

OR i.end\_ts=''

OR i.geo\_cd=''

OR sg.geo\_cd IS NULL

OR sg.geo\_cd=''

OR sa.artist\_id IS NULL

OR sa.artist\_id='', 'fail', 'pass') AS status

FROM formatted\_input i

LEFT OUTER JOIN station\_geo\_map sg ON i.station\_id = sg.station\_id

LEFT OUTER JOIN song\_artist\_map sa ON i.song\_id = sa.song\_id

WHERE i.batchid=${hiveconf:batchid};

INSERT OVERWRITE local DIRECTORY '/home/acadgild/project/enriched\_data'

ROW FORMAT DELIMITED

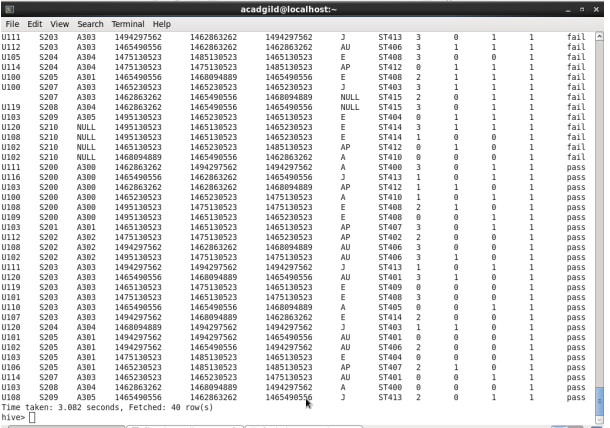
FIELDS TERMINATED BY ','

STORED AS TEXTFILE

SELECT \* FROM enriched\_data;

***OUTPUT:***

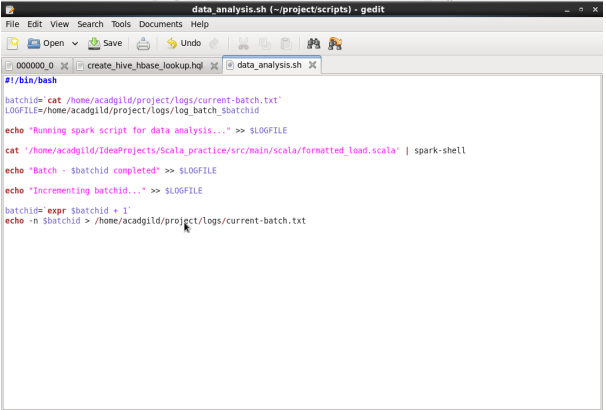
******

******

* ***Perform Data Analysis:***

Below is the shell script *data\_analysis.sh* where the following operations are performed:

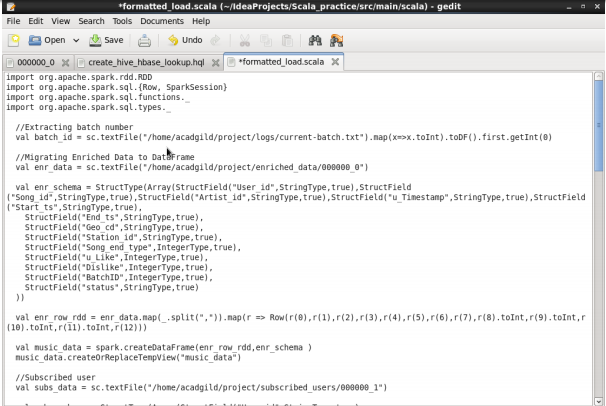
* Get the batch id number from the batch file and get the Log File for the batch using the batch id.
* Add logs to the Log File signifying that the data analysis is being performed using Spark and that the result is being exported to the Local FS.
* Run the spark script formatted\_load.scala. This will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



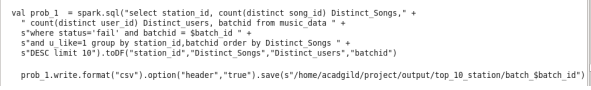
In the *file formatted\_load.scala* that will perform data analysis and store each result into local drive by creating a new directory with every new batch id.

*Steps :*

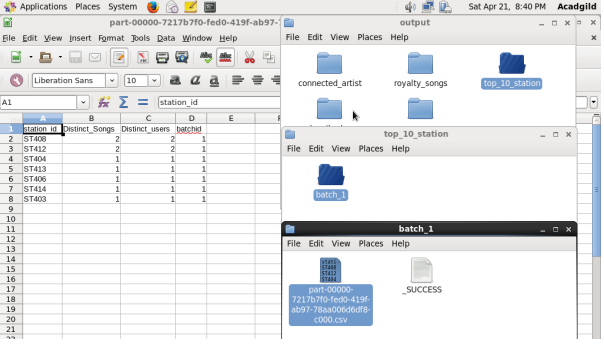
* Importing row, Sparksession, RDD, functions and types which are need for analysis.
* Then, we will retrieve the batch id from current batch file.
* We will now import data stored in local FS i.e. enriched\_data, subscribed\_user and user\_artists and perform the following operations for each : o Create the schema for the data o Create data frame
* Create temporary table now we are ready for analysis.



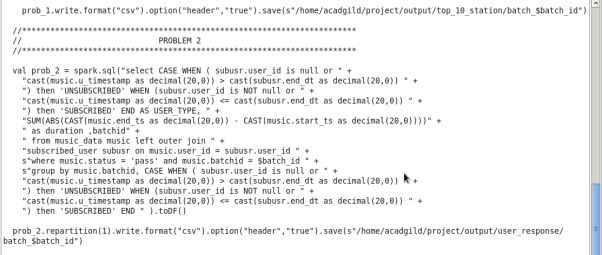
***PROBLEM 1:*** Determine top 10 station\_id(s) where maximum number of songs were played, which were liked by unique users.



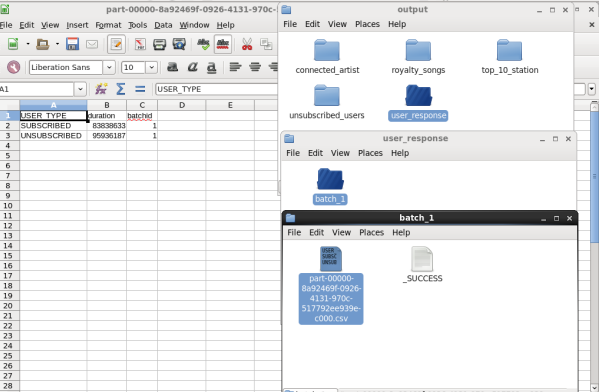
***OUTPUT:***

******

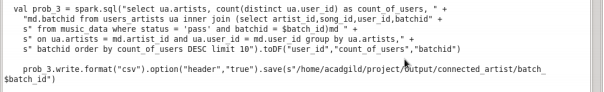
***PROBLEM 2:*** Determine total duration of songs played by each type of user, where type of user can be 'subscribed' or 'unsubscribed'. An unsubscribed user is the one whose record is either not present in Subscribed\_users lookup table or has subscription\_end\_date earlier than the timestamp of the song played by him.



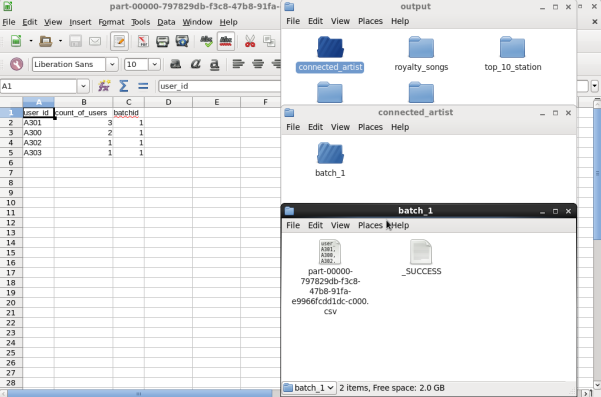
***OUTPUT:***

******

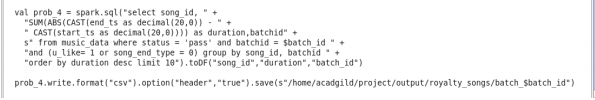
***PROBLEM 3:*** Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.

******

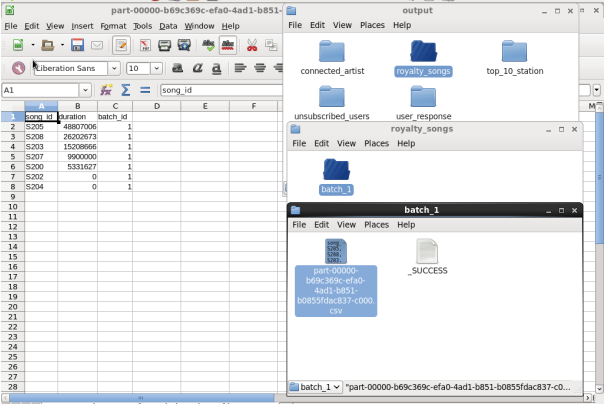
***OUTPUT:***

******

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

******

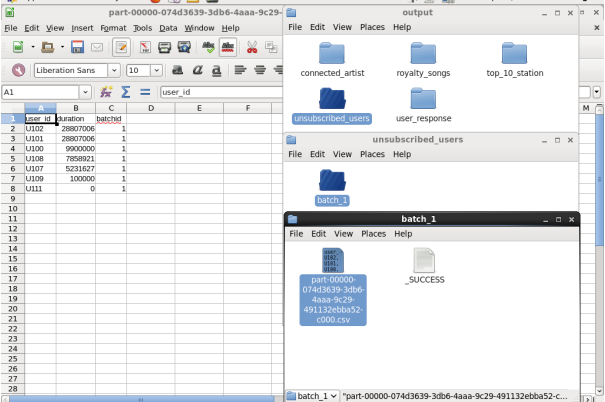
***OUTPUT:***

******

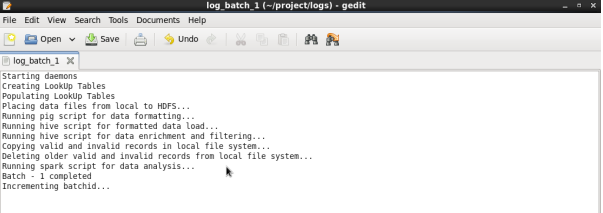
***PROBLEM 5:*** Determine top 10 unsubscribed users who listened to the songs for the longest duration.

******

***OUTPUT:***

******

***Log file:***

******