Final Project

The unix script which will run the project-

music_project_master.sh

```
# Create data
echo "Preparing to execute python scripts to generate data..."

rm -r /home/acadgild/examples/music/data/web

rm -r /home/acadgild/examples/music/data/mob

mkdir -p /home/acadgild/examples/music/data/web

mkdir -p /home/acadgild/examples/music/data/mob

python /home/acadgild/examples/music/generate_web_data.py
python /home/acadgild/examples/music/generate_mob_data.py

echo "Data Generated Successfully !"

# Call Stop start daemon scripts to start hadoop daemons

echo "Starting the daemons...."
#sh start-daemons.sh

# run jps commands to check the daemons

jps

echo "All hadoop daemons started !"

echo "Upload the look up tables now in Hbase..."

#sh populate-lookup.sh

echo "Done with data population in look up tables !"
```

Creating a data every 3 hours –

This will be moved to scheduler to run every 3 hours

```
from random import randint
from random import choice

file = open("/home/acadgild/examples/music/data/web/file.xml", "w")
count = 20

file.write("<records>\n")

while (count > 0):
        geo_cd_list=["A", "E", "AU", "AP", "U"]
        song_end_type_list=["0","1","2","3"]
        timestamp_list=["2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22"]
        start_ts_list=["2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22"]
        end_ts_list=["2016-05-10 12:24:22", "2016-06-09 22:12:36", "2016-07-10 01:38:09", "2017-05-09 08:09:22"]

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

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

        if (count%11 == 0):
                artist_id = ""
        else:
                artist_id = "A" + str(randint(300,305))

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

        if (count%12 == 0):
                geo_cd = ""
```

Creating a directory called music with all the required files

```
File  Edit  View  Search  Terminal  Help
[acadgild@localhost music]$ ls -l
total 100
-rwxr-xr-x. 1 acadgild acadgild   912 Oct 13 23:49 create_hive_hbase_lookup.hql
-rw-r--r--. 1 acadgild acadgild   912 Oct 13 23:48 create_hive_hbase_lookup.hql~
-rwxr-xr-x. 1 acadgild acadgild    83 Jun 10  2018 create_hive_hbase_lookup.sh
-rw-r--r--. 1 acadgild acadgild    84 Jun 10  2018 create_hive_hbase_lookup.sh~
-rwxr-xr-x. 1 acadgild acadgild   312 Oct 14 03:35 data_enrichment_filtering_schema.sh
-rw-r--r--. 1 acadgild acadgild   313 Jun 10  2018 data_enrichment_filtering_schema.sh~
-rwxr-xr-x. 1 acadgild acadgild   415 Jun 10  2018 dataformatting.sh
-rw-r--r--. 1 acadgild acadgild   416 Jun 10  2018 dataformatting.sh~
-rw-rw-r--. 1 acadgild acadgild 10824 Jun 12  2018 employee.java
-rwxrwxr-x. 1 acadgild acadgild  1200 Jun 10  2018 generate_mob_data.py
-rwxrwxr-x. 1 acadgild acadgild  1868 Jun 10  2018 generate_web_data.py
drwxrwxr-x. 2 acadgild acadgild  4096 Jun 10  2018 lookupfiles
drwxrwxr-x. 6 acadgild acadgild  4096 Oct 14 04:30 MusicDataAnalysis
-rwxrwxr-x. 1 acadgild acadgild  1256 Dec 15 13:34 music_project_master.sh
-rw-rw-r--. 1 acadgild acadgild  1255 Oct 14 04:58 music_project_master.sh~
-rw-rw-r--. 1 acadgild acadgild 16507 Jun 12  2018 Person.java
-rwxrwxr-x. 1 acadgild acadgild  1645 Jun 10  2018 populate-lookup.sh
-rwxrwxr-x. 1 acadgild acadgild   688 Jun 11  2018 start-daemons.sh
-rw-rw-r--. 1 acadgild acadgild   691 Jun 10  2018 start-daemons.sh~
[acadgild@localhost music]$
```

Once the file music_project_master.sh is run, data folder is created within which the xml will be present-

```
-rwxr-xr-x. 1 acadgild acadgild    83 Jun 10  2018 create_hive_hbase_lookup.sh
-rw-r--r--. 1 acadgild acadgild    84 Jun 10  2018 create_hive_hbase_lookup.sh~
drwxrwxr-x. 4 acadgild acadgild  4096 Dec 15 13:36 data
-rwxr-xr-x. 1 acadgild acadgild   312 Oct 14 03:35 data_enrichment_filtering_schema.sh
-rw-r--r--. 1 acadgild acadgild   313 Jun 10  2018 data_enrichment_filtering_schema.sh~
```

Starting the Hadoop

```
[acadgild@localhost music]$ jps
6049 HRegionServer
5090 SecondaryNameNode
5347 NodeManager
5923 HMaster
4804 NameNode
6471 Jps
5835 HQuorumPeer
4907 DataNode
5244 ResourceManager
6142 JobHistoryServer
[acadgild@localhost music]$
```

Creating lookup tables with some lookup data-

It takes the batch id from the batch.txt file

```bash
#!/bin/bash

batchid=`cat /home/acadgild/examples/music/logs/current-batch.txt`

LOGFILE=/home/acadgild/examples/music/logs/log_batch_$batchid

echo "Creating LookUp Tables" >> $LOGFILE

echo "disable 'station-geo-map'" | hbase shell
echo "drop 'station-geo-map'" | hbase shell
echo "disable 'subscribed-users'" | hbase shell
echo "drop 'subscribed-users'" | hbase shell
echo "disable 'song-artist-map'" | hbase shell
echo "drop 'song-artist-map'" | hbase shell

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


echo "Populating LookUp Tables" >> $LOGFILE

file="/home/acadgild/examples/music/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"
```

Created tables with copied data in it.

```
hbase(main):001:0> list
TABLE
bulktable
click
clicks
employee
htest
song-artist-map
station-geo-map
subscribed-users
8 row(s) in 0.7640 seconds

=> ["bulktable", "click", "clicks", "employee", "htest", "song-artist-map", "station-geo-map", "
hbase(main):002:0> scan 'station-geo-map'
ROW                                    COLUMN+CELL
 ST400                                 column=geo:geo_cd, timestamp=1542831929253, value=A
 ST401                                 column=geo:geo_cd, timestamp=1542831943102, value=AU
 ST402                                 column=geo:geo_cd, timestamp=1542831956654, value=AP
 ST403                                 column=geo:geo_cd, timestamp=1542831970737, value=J
 ST404                                 column=geo:geo_cd, timestamp=1542831984552, value=E
 ST405                                 column=geo:geo_cd, timestamp=1542831998485, value=A
 ST406                                 column=geo:geo_cd, timestamp=1542832012536, value=AU
 ST407                                 column=geo:geo_cd, timestamp=1542832028982, value=AP
 ST408                                 column=geo:geo_cd, timestamp=1542832054352, value=E
 ST409                                 column=geo:geo_cd, timestamp=1542832069317, value=E
 ST410                                 column=geo:geo_cd, timestamp=1542832083315, value=A
 ST411                                 column=geo:geo_cd, timestamp=1542832097508, value=A
 ST412                                 column=geo:geo_cd, timestamp=1542832112084, value=AP
```

Completion of data enrichment

```
Done with data population in look up tables !
Lets do some data formatting now....
data formatting complete !
Creating hive tables on top of hbase tables for data enrichment and filtering...
Hive table with Hbase Mapping Complete !
Let us do data enrichment as per the requirement...
Data Enrichment Complete
Lets run some use cases now...
USE CASES COMPLETE !!
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost music]$
```

Completion of data deformatting

```
:: USE VERBOSE OR DEBUG MESSAGE LEVEL FOR MORE DETAILS
Exception in thread "main" java.lang.RuntimeException: [unresolved dependency: com.databricks#spark-xml_2.10;0.4.1: not found
]
        at org.apache.spark.deploy.SparkSubmitUtils$.resolveMavenCoordinates(SparkSubmit.scala:1197)
        at org.apache.spark.deploy.SparkSubmit$.prepareSubmitEnvironment(SparkSubmit.scala:304)
        at org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:153)
        at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:119)
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
data formatting complete !
Creating hive tables on top of hbase tables for data enrichment and filtering...
Hive table with Hbase Mapping Complete !
Let us do data enrichment as per the requirement...
Data Enrichment Complete
Lets run some use cases now...
USE CASES COMPLETE !!
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost music]$
```

Creating a hive table as below-

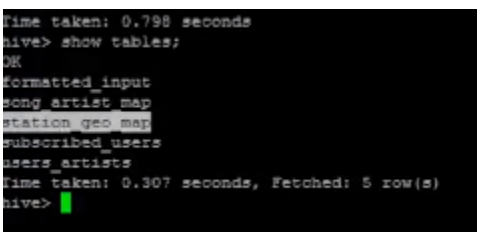
```
acadgild@localhost:~/examples/music   acadgild@localhost:~/examples/music   acadgild@localhost:~/examples/music
acadgild@localhost music]$ hive -f /home/acadgild/project/scripts/create_hive_hbase_lookup.hql
```

```
USE project;

create external table if not exists station_geo_map
(
station_id String,
geo_cd string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
with serdeproperties
("hbase.columns.mapping"=":key,geo:geo_cd")
tblproperties("hbase.table.name"="station-geo-map");

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

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



Creating the enriched data

```
USE project;

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


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

populated enriched data -

```
subscribed_users
users_artists
Time taken: 0.324 seconds, Fetched: 6 row(s)
hive> select * from enriched_data;
OK
U106    S200    A300    1465130523      1475130523      1465290523      AP      ST412   2       1       1       1       fail
        S202    A302    1462863262      1468094889      1494297562      J       ST413   0       1       0       1       fail
        S203    A303    1475130523      1465130523      1475130523      AP      ST407   3       1       0       1       fail
U104    S204    A304    1494297562      1462863262      1494297562      A       ST411   2       1       1       1       fail
U117    S208    A304    1494297562      1462863262      1465490556      E       ST409   3       1       1       1       fail
U111    S208    A304    1462863262      1462863262      1462863262      AU      ST401   2       1       1       1       fail
U114    S208    A304    1465130523      1485130523      1485130523      E       ST414   0       1       1       1       fail
U118    S210    NULL    1475130523      1465230523      1465130523      AU      ST406   1       1       1       1       fail
U109    S210    NULL    1475130523      1465130523      1485130523      E       ST409   3       0       0       1       fail
U102    S210    NULL    1465130523      1465230523      1485130523      A       ST411   1       1       0       1       fail
U113    S210    NULL    1462863262      1494297562      1465490556      AU      ST406   1       1       1       1       fail
U104    S210    NULL    1465230523      1485130523      1475130523      E       ST409   0       1       0       1       fail
U106    S210    NULL    1468094889      1468094889      1465490556      A       ST411   1       0       0       1       fail
U101    S210    NULL    1468094889      1465490556      1462863262      A       ST410   1       0       1       1       fail
U115    S200    A300    1465490556      1468094889      1465490556      E       ST409   0       0       1       1       pass
U120    S200    A300    1468094889      1465490556      1462863262      J       ST403   0       1       0       1       pass
U114    S201    A301    1495130523      1475130523      1475130523      A       ST405   2       1       0       1       pass
U104    S201    A301    1495130523      1475130523      1475130523      AP      ST412   1       1       0       1       pass
U112    S202    A302    1465490556      1465490556      1462863262      E       ST404   1       0       1       1       pass
U119    S202    A302    1468094889      1465490556      1462863262      AP      ST407   1       0       0       1       pass
U108    S202    A302    1495130523      1475130523      1485130523      A       ST411   3       1       0       1       pass
U111    S202    A302    1475130523      1485130523      1485130523      A       ST400   1       0       0       1       pass
U112    S202    A302    1465490556      1468094889      1468094889      E       ST409   0       0       1       1       pass
U101    S202    A302    1495130523      1485130523      1475130523      AU      ST406   1       0       0       1       pass
U120    S202    A302    1465490556      1468094889      1468094889      E       ST408   1       0       1       1       pass
U107    S203    A303    1475130523      1475130523      1485130523      E       ST408   1       0       0       1       pass
U108    S203    A303    1495130523      1475130523      1485130523      A       ST405   0       1       0       1       pass
U111    S204    A304    1495130523      1465230523      1465130523      AP      ST412   0       1       0       1       pass
U103    S204    A304    1494297562      1494297562      1468094889      AP      ST407   1       0       0       1       pass
U104    S204    A304    1465490556      1468094889      1462863262      A       ST410   2       0       0       1       pass
U117    S205    A301    1465290523      1485130523      1465290523      E       ST409   3       0       0       1       pass
U106    S207    A303    1468094889      1465490556      1465490556      A       ST400   2       0       0       1       pass
U110    S207    A303    1462863262      1462863262      1468094889      A       ST411   1       0       0       1       pass
U109    S207    A303    1468094889      1462863262      1494297562      AP      ST407   0       1       0       1       pass
U119    S207    A303    1468094889      1468094889      1465490556      AU      ST406   1       0       0       1       pass
U113    S207    A303    1494297562      1465490556      1465490556      E       ST404   0       1       0       1       pass
U116    S208    A304    1465130523      1485130523      1485130523      AU      ST406   3       0       0       1       pass
U100    S208    A304    1465130523      1465230523      1465290523      J       ST403   2       0       1       1       pass
```

Data Analysis

To find the top 10 station ids

with unique users

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

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

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

Output

```
Time taken: 122.108 seconds
hive> select * from  top_10_stations;
OK
ST412   2       2       1
ST405   2       2       1
ST411   1       1       1
ST407   1       1       1
ST406   1       1       1
ST404   1       1       1
ST403   1       1       1
Time taken: 0.357 seconds, Fetched: 7 row(s)
hive>
```