Music Data Analysis Project

Project Description:

A leading music-catering company is planning to analyze large amount of data received from varieties of sources, namely mobile app and website to track the behavior of users, classify users, calculate royalties associated with the song and make appropriate business strategies. The file server receives data files periodically after every 3 hours.

Problem Statement:

- Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users. It is not only the data which is important, rather it is the insight it can be used to generate important. Once we have made the data ready for analysis, we have to perform below analysis on a daily basis.
- 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.
- Determine top 10 connected artists. Connected artists are those whose songs are most listened by the unique users who follow them.
- 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.
- Determine top 10 unsubscribed users who listened to the songs for the longest duration.

Solution:

There are five phases in this project:

- Data generation for web and mobile sources
- Data population in the lookup tables
- Data formatting
- Data enrichment
- Data analysis

In the course of converting an existing Hadoop project into a Spark project, I have written code for data formatting, data enrichment and data analysis phases.

Here are the rules to follow for data ingestion and formatting:

- Data coming from mobile applications reside in /data/mob and has csv format.
- Data coming from web applications reside in /data/web and has xml format.
- All the timestamp fields in data coming from web application is of the format YYYY-MM-DD HH:MM:SS.
- All the timestamp fields in data coming from mobile application is a long integer interpreted as UNIX timestamps.
- Finally, all timestamps must have the format of a long integer to be interpreted as UNIX timestamps.

Let’s have a glance at the source code written in Scala language:

DataFormatting.scala

```
// import required Spark packages
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.sql

object DataFormatting {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("Data Formatting")
    val sc = new SparkContext(conf)                    // create spark context using spark ‘conf’ object
    val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc) //create a Hive context object
    val batchId = args(0)                              // get batch id from command line arguments
// SQL query to create table ‘formatted_input’ with the given schema description
    val create_hive_table = s"""CREATE TABLE IF NOT EXISTS project.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 ','
                        """
```

// Hive command to load **mobile data** from local file system to Hive table ‘formatted_input’

```
  val load_mob_data = s"""LOAD DATA LOCAL INPATH '/home/acadgild/project/data/mob/
file.txt' INTO TABLE project.formatted_input PARTITION (batchid='$batchId')"""
```

// Hive query to insert **web data** to Hive table ‘formatted_input’ converting time stamp format

```
  val load_web_data = s"""INSERT INTO project.formatted_input
                PARTITION(batchid='$batchId')
                SELECT user_id,
                song_id,
                artist_id,
                unix_timestamp(timestamp,'yyyy-MM-dd HH:mm:ss') AS timestamp,
                unix_timestamp(start_ts,'yyyy-MM-dd HH:mm:ss') AS start_ts,
                unix_timestamp(end_ts,'yyyy-MM-dd HH:mm:ss') AS end_ts,
                geo_cd,
                station_id,
                              song_end_type,
                              like,
                              dislike
                              FROM web_data
                              """
    try {
// read web data stored in XML format from local file system
         val xmlData = sqlContext.read.format("com.databricks.spark.xml").option("rowTag",
"record").load("/home/acadgild/project/data/web/file.xml")
xmlData.createOrReplaceTempView("web_data")               //create a view from XML data loaded
         sqlContext.sql(create_hive_table)                  // execute the HiveQL queries prepared above
         sqlContext.sql(load_mob_data)
         sqlContext.sql(load_web_data)
        }
      catch{
       case e: Exception=>e.printStackTrace()                         // print stack trace in case of errors
      }
   }
}
```

Rules for data enrichment:

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

DataEnrichment.scala

```
// import required Spark packages
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.sql
```

```
object DataEnrichment {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("Data Formatting")
    val sc = new SparkContext(conf)                          //create spark context using spark ‘conf’ object
    val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc) //create a Hive context object
    val batchId = args(0)                                    // get batch id from command line arguments
// Hive query to insert web data to Hive table ‘enriched_data’
    val create_hive_table = s"""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
                         """
// Hive query to insert data from ‘formatted_output’ into the table ‘enriched_data’
    val load_data = s"""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,
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.geo_cd IS NULL
OR i.user_id=''
OR i.song_id=''
OR i.timestamp=''
OR i.start_ts=''
OR i.end_ts=''
OR i.geo_cd=''
OR 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=$batchId
                    """
    try {
        sqlContext.sql("SET hive.auto.convert.join=false")          // configure Hive properties
        sqlContext.sql("SET hive.exec.dynamic.partition.mode=nonstrict")
        sqlContext.sql("USE project")                    // switch to an existing database 'project'

        sqlContext.sql(create_hive_table)               // execute the HiveQL queries prepared above
        sqlContext.sql(load_data)
      }
      catch{
       case e: Exception=>e.printStackTrace()          // print stack trace in case of errors
      }
    }
}
```

Data Analysis:

The following code contains HiveQL queries to obtain the data specified in the problem statement:

DataAnalysis.scala

```
// import required Spark packages
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.sql

object DataAnalysis {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("Data Analysis")
    val sc = new SparkContext(conf)                    //create spark context using spark 'conf' object
    val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc) //create a Hive context object
    val batchId = args(0)                                  // get batch id from command line arguments

// Hive query to create a table to store top 10 station id’s where maximum number of songs played

val create_top_10_stations = """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"""

// Hive query to determine top 10 station_id(s) and store results into ‘top_10_stations' table
val load_top_10_stations = s"""INSERT OVERWRITE TABLE top_10_stations
                                PARTITION(batchid='$batchId')
                                SELECT station_id,
                                COUNT(DISTINCT song_id) AS total_distinct_songs_played,
                                COUNT(DISTINCT user_id) AS distinct_user_count
                                FROM enriched_data
                                WHERE status='pass'
                                AND batchid='$batchId'
                                AND like=1
                                GROUP BY station_id
                                ORDER BY total_distinct_songs_played DESC
                                LIMIT 10"""
```

```
// Hive query to create a table to store the total duration of songs played by each type of user

val create_users_behaviour = """CREATE TABLE IF NOT EXISTS users_behaviour

                                (
                                user_type STRING,
                                duration INT
                                )
                                PARTITIONED BY (batchid INT)
                                ROW FORMAT DELIMITED
                                FIELDS TERMINATED BY ','
                                STORED AS TEXTFILE"""
// Hive query to determine total duration of songs played by each type of user and store results into
// ‘users_behaviour’ table
val load_users_behaviour = s"""INSERT OVERWRITE TABLE users_behaviour
                                PARTITION(batchid='$batchId')
                                SELECT
                                CASE WHEN (su.user_id IS NULL OR CAST(ed.timestamp AS
                                DECIMAL(20,0)) > CAST(su.subscn_end_dt AS
                                DECIMAL(20,0))) THEN 'UNSUBSCRIBED'
                                WHEN (su.user_id IS NOT NULL AND CAST(ed.timestamp AS
                                DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS
                                DECIMAL(20,0))) THEN 'SUBSCRIBED'
                                END AS user_type,
                                SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-
                                CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration
                                FROM enriched_data ed
                                LEFT OUTER JOIN subscribed_users su
                                ON ed.user_id=su.user_id
                                WHERE ed.status='pass'
                                AND ed.batchid='$batchId'
                                GROUP BY CASE WHEN (su.user_id IS NULL OR
                                CAST(ed.timestamp AS DECIMAL(20,0)) >
                                CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN
                                'UNSUBSCRIBED'
                                WHEN (su.user_id IS NOT NULL AND CAST(ed.timestamp AS
                                DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS
                                DECIMAL(20,0))) THEN 'SUBSCRIBED' END"""
// Hive query to create a table to store data of top 10 connected artists.
val create_connected_artists = """CREATE TABLE IF NOT EXISTS connected_artists
                                (
                                artist_id STRING,
                                user_count INT
                                )
                                PARTITIONED BY (batchid INT)
                                ROW FORMAT DELIMITED
                                FIELDS TERMINATED BY ','
                                STORED AS TEXTFILE"""
// Hive query to determine top 10 connected artists and store results into ‘connected_artists’ table
val load_connected_artists = s"""INSERT OVERWRITE TABLE connected_artists
                                PARTITION(batchid='$batchId')
                                SELECT ua.artist_id,
                                COUNT(DISTINCT ua.user_id) AS user_count
                                FROM
                                (
                                SELECT user_id, artist_id FROM users_artists
                                LATERAL VIEW explode(artists_array) artists AS artist_id
                                ) ua
                                INNER JOIN
                                (
                                SELECT artist_id, song_id, user_id
                                FROM enriched_data
                                WHERE status='pass'
                                AND batchid='$batchId'
```

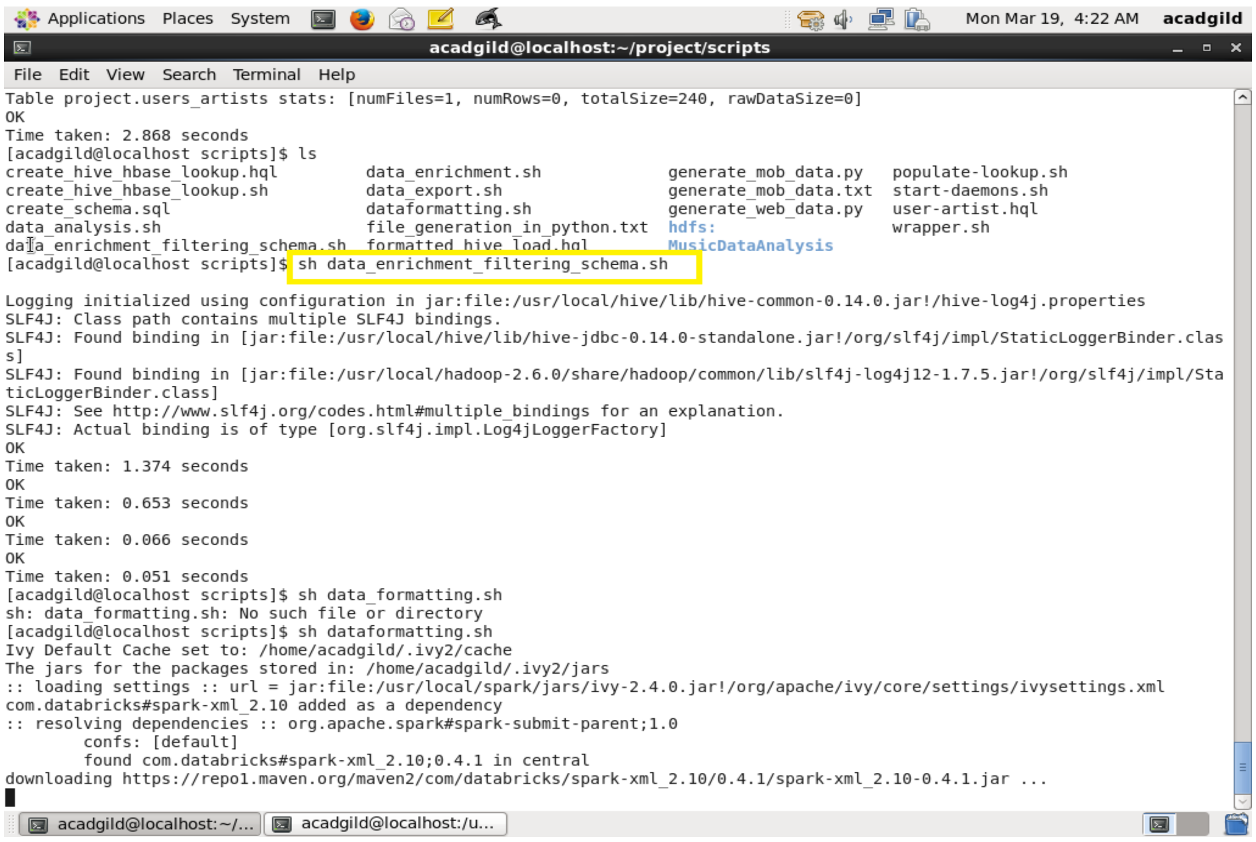
```
                                        ) ed
                                        ON ua.artist_id=ed.artist_id
                                        AND ua.user_id=ed.user_id
                                        GROUP BY ua.artist_id
                                        ORDER BY user_count DESC
                                        LIMIT 10"""
// Hive query to create a table to store data of top 10 royalty songs
val create_top_10_royalty_songs = """CREATE TABLE IF NOT EXISTS top_10_royalty_songs
                                        (
                                        song_id STRING,
                                        duration INT
                                        )
                                        PARTITIONED BY (batchid INT)
                                        ROW FORMAT DELIMITED
                                        FIELDS TERMINATED BY ','
                                        STORED AS TEXTFILE"""
// Hive query to determine top 10 royalty songs and store results into top_10_royalty_songs table
val load_top_10_royalty_songs = s"""INSERT OVERWRITE TABLE top_10_royalty_songs
                                        PARTITION(batchid='$batchId')
                                        SELECT song_id,
                                        SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS
                                        DECIMAL(20,0)))) AS duration
                                        FROM enriched_data
                                        WHERE status='pass'
                                        AND batchid='$batchId'
                                        AND (like=1 OR song_end_type=0)
                                        GROUP BY song_id
                                        ORDER BY duration DESC
                                        LIMIT 10"""
```

```
// Hive query to create a table to store data of top 10 unsubscribed users
val create_top_10_unsubscribed_users = """CREATE TABLE IF NOT EXISTS top_10_unsubscribed_users
                                          (
                                          user_id STRING,
                                          duration INT
                                          )
                                          PARTITIONED BY (batchid INT)
                                          ROW FORMAT DELIMITED
                                          FIELDS TERMINATED BY ','
                                          STORED AS TEXTFILE"""
// Hive query to determine top 10 unsubscribed users who listened to songs for the longest duration
val load_top_10_unsubscribed_users = s"""INSERT OVERWRITE TABLE top_10_unsubscribed_users
                                          PARTITION(batchid='$batchId')
                                          SELECT ed.user_id,
                                          SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration
                                          FROM enriched_data ed
                                          LEFT OUTER JOIN subscribed_users su
                                          ON ed.user_id=su.user_id
                                          WHERE ed.status='pass'
                                          AND ed.batchid='$batchId'
                                          AND (su.user_id IS NULL OR (CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))))
                                          GROUP BY ed.user_id
                                          ORDER BY duration DESC
                                          LIMIT 10"""
  try {
      sqlContext.sql("SET hive.auto.convert.join=false")    // configure Hive properties
```

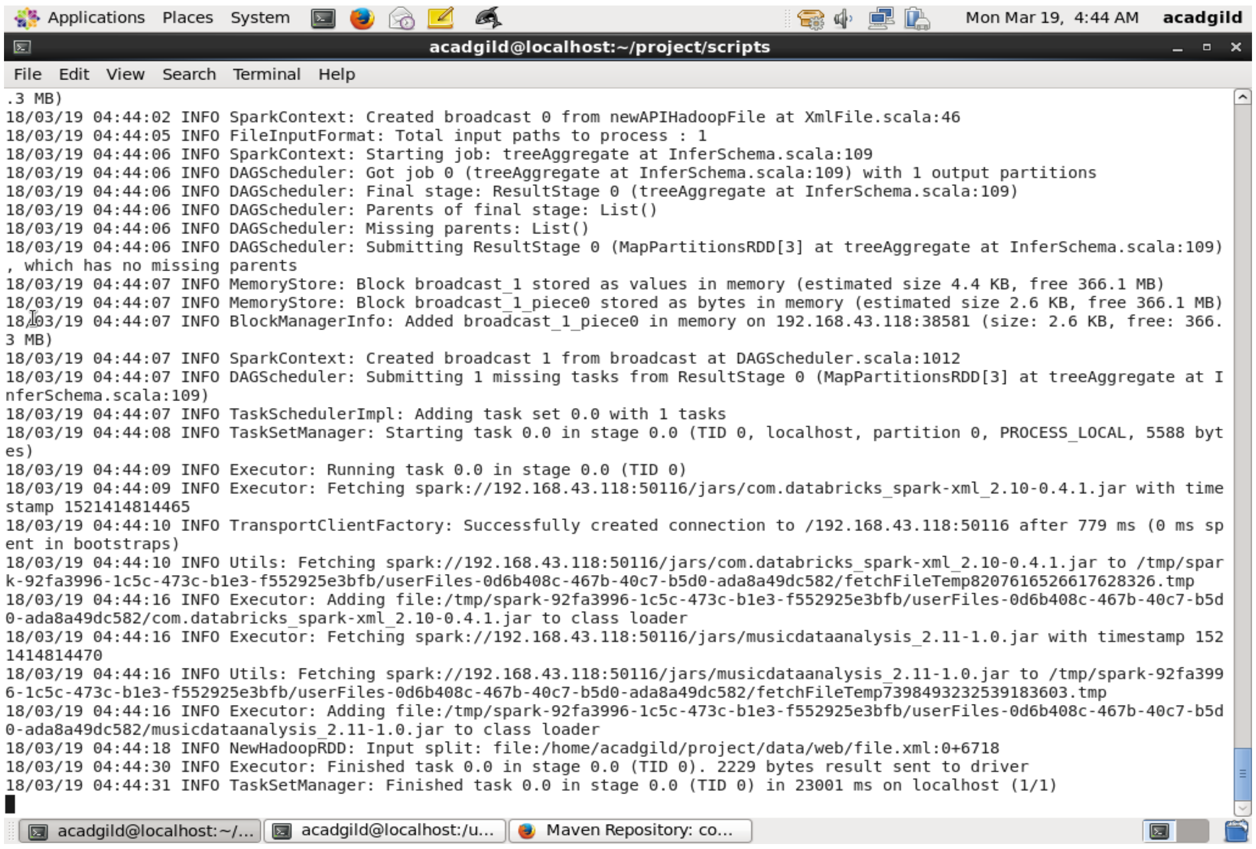
```
        sqlContext.sql("USE project")                          // switch to the working database 'project'
        sqlContext.sql(create_top_10_stations)              // execute all the Hive queries created above
        sqlContext.sql(load_top_10_stations)
        sqlContext.sql(create_users_behaviour)
        sqlContext.sql(load_users_behaviour)
        sqlContext.sql(create_connected_artists)
        sqlContext.sql(load_connected_artists)
        sqlContext.sql(create_top_10_royalty_songs)
        sqlContext.sql(load_top_10_royalty_songs)
        sqlContext.sql(create_top_10_unsubscribed_users)
        sqlContext.sql(load_top_10_unsubscribed_users)
      }
     catch{
      case e: Exception=>e.printStackTrace()                       // print stack trace in case of errors
     }
   }
}
```

Output:

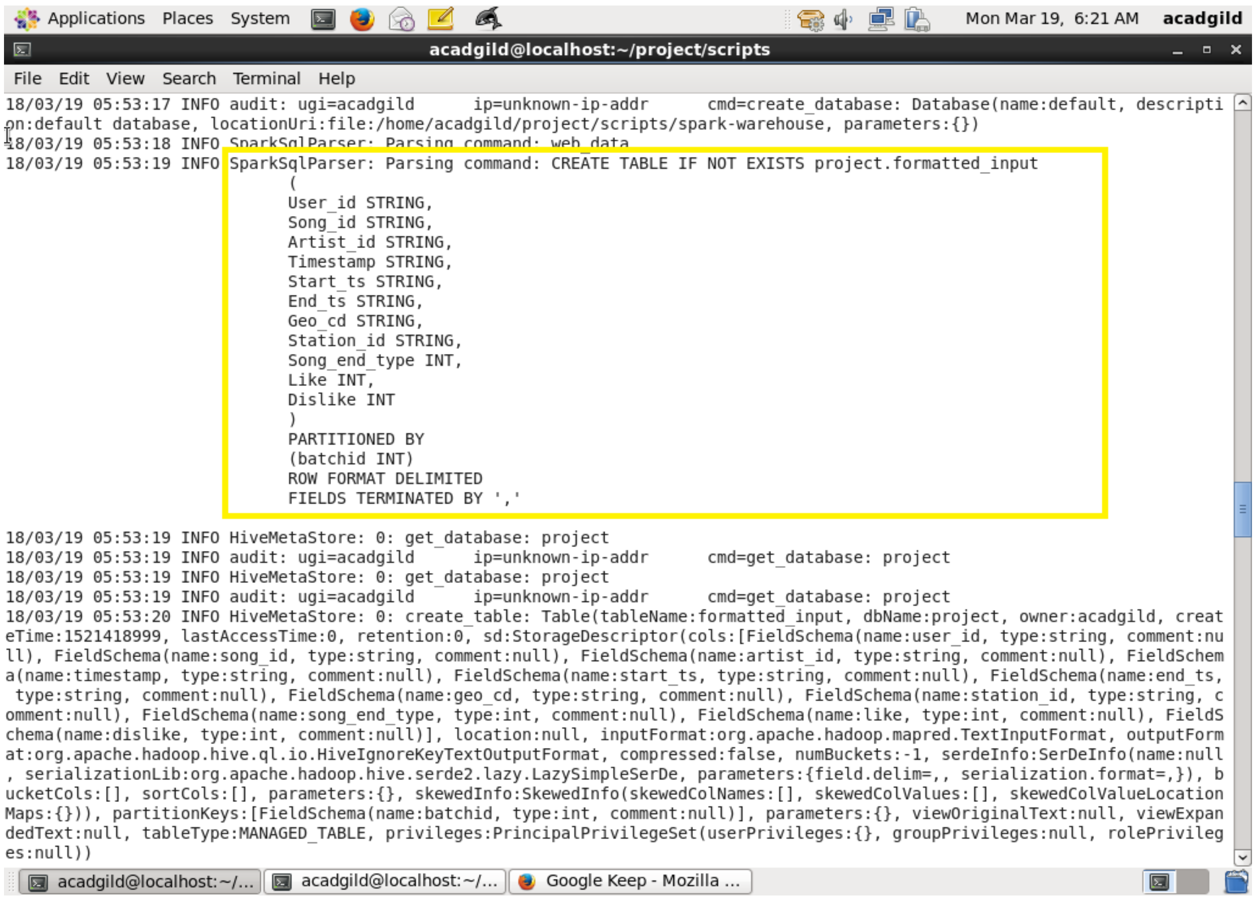
Data Formatting – Screenshots of execution:



```
Table project.users_artists stats: [numFiles=1, numRows=0, totalSize=240, rawDataSize=0]
OK
Time taken: 2.868 seconds
[acadgild@localhost scripts]$ ls
create_hive_hbase_lookup.hql        data_enrichment.sh             generate_mob_data.py   populate-lookup.sh
create_hive_hbase_lookup.sh         data_export.sh                 generate_mob_data.txt  start-daemons.sh
create_schema.sql                   dataformatting.sh              generate_web_data.py   user-artist.hql
data_analysis.sh                    file_generation_in_python.txt  hdfs:                  wrapper.sh
data_enrichment_filtering_schema.sh formatted_hive_load.hql        MusicDataAnalysis
[acadgild@localhost scripts]$ sh data_enrichment_filtering_schema.sh

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-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-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.clas
s]
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.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
OK
Time taken: 1.374 seconds
OK
Time taken: 0.653 seconds
OK
Time taken: 0.066 seconds
OK
Time taken: 0.051 seconds
[acadgild@localhost scripts]$ sh data_formatting.sh
sh: data_formatting.sh: No such file or directory
[acadgild@localhost scripts]$ sh dataformatting.sh
Ivy Default Cache set to: /home/acadgild/.ivy2/cache
The jars for the packages stored in: /home/acadgild/.ivy2/jars
:: loading settings :: url = jar:file:/usr/local/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
com.databricks#spark-xml_2.10 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent;1.0
        confs: [default]
        found com.databricks#spark-xml_2.10;0.4.1 in central
downloading https://repo1.maven.org/maven2/com/databricks/spark-xml_2.10/0.4.1/spark-xml_2.10-0.4.1.jar ...
```

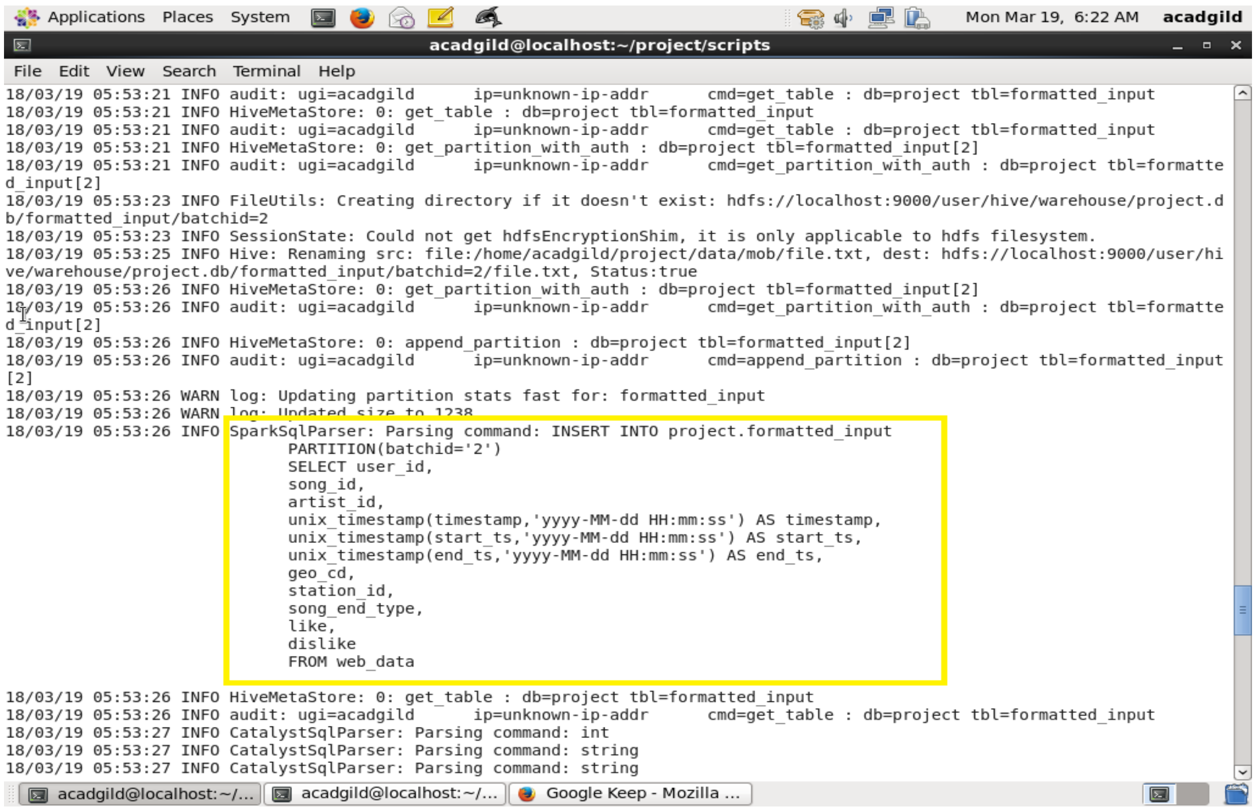


```
.3 MB)
18/03/19 04:44:02 INFO SparkContext: Created broadcast 0 from newAPIHadoopFile at XmlFile.scala:46
18/03/19 04:44:05 INFO FileInputFormat: Total input paths to process : 1
18/03/19 04:44:06 INFO SparkContext: Starting job: treeAggregate at InferSchema.scala:109
18/03/19 04:44:06 INFO DAGScheduler: Got job 0 (treeAggregate at InferSchema.scala:109) with 1 output partitions
18/03/19 04:44:06 INFO DAGScheduler: Final stage: ResultStage 0 (treeAggregate at InferSchema.scala:109)
18/03/19 04:44:06 INFO DAGScheduler: Parents of final stage: List()
18/03/19 04:44:06 INFO DAGScheduler: Missing parents: List()
18/03/19 04:44:06 INFO DAGScheduler: Submitting ResultStage 0 (MapPartitionsRDD[3] at treeAggregate at InferSchema.scala:109)
, which has no missing parents
18/03/19 04:44:07 INFO MemoryStore: Block broadcast_1 stored as values in memory (estimated size 4.4 KB, free 366.1 MB)
18/03/19 04:44:07 INFO MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 2.6 KB, free 366.1 MB)
18/03/19 04:44:07 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on 192.168.43.118:38581 (size: 2.6 KB, free: 366.
3 MB)
18/03/19 04:44:07 INFO SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1012
18/03/19 04:44:07 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 0 (MapPartitionsRDD[3] at treeAggregate at I
nferSchema.scala:109)
18/03/19 04:44:07 INFO TaskSchedulerImpl: Adding task set 0.0 with 1 tasks
18/03/19 04:44:08 INFO TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, localhost, partition 0, PROCESS_LOCAL, 5588 byt
es)
18/03/19 04:44:09 INFO Executor: Running task 0.0 in stage 0.0 (TID 0)
18/03/19 04:44:09 INFO Executor: Fetching spark://192.168.43.118:50116/jars/com.databricks_spark-xml_2.10-0.4.1.jar with time
stamp 1521414814465
18/03/19 04:44:10 INFO TransportClientFactory: Successfully created connection to /192.168.43.118:50116 after 779 ms (0 ms sp
ent in bootstraps)
18/03/19 04:44:10 INFO Utils: Fetching spark://192.168.43.118:50116/jars/com.databricks_spark-xml_2.10-0.4.1.jar to /tmp/spar
k-92fa3996-1c5c-473c-b1e3-f552925e3bfb/userFiles-0d6b408c-467b-40c7-b5d0-ada8a49dc582/fetchFileTemp8207616526617628326.tmp
18/03/19 04:44:16 INFO Executor: Adding file:/tmp/spark-92fa3996-1c5c-473c-b1e3-f552925e3bfb/userFiles-0d6b408c-467b-40c7-b5d
0-ada8a49dc582/com.databricks_spark-xml_2.10-0.4.1.jar to class loader
18/03/19 04:44:16 INFO Executor: Fetching spark://192.168.43.118:50116/jars/musicdataanalysis_2.11-1.0.jar with timestamp 152
1414814470
18/03/19 04:44:16 INFO Utils: Fetching spark://192.168.43.118:50116/jars/musicdataanalysis_2.11-1.0.jar to /tmp/spark-92fa399
6-1c5c-473c-b1e3-f552925e3bfb/userFiles-0d6b408c-467b-40c7-b5d0-ada8a49dc582/fetchFileTemp7398493232539183603.tmp
18/03/19 04:44:16 INFO Executor: Adding file:/tmp/spark-92fa3996-1c5c-473c-b1e3-f552925e3bfb/userFiles-0d6b408c-467b-40c7-b5d
0-ada8a49dc582/musicdataanalysis_2.11-1.0.jar to class loader
18/03/19 04:44:18 INFO NewHadoopRDD: Input split: file:/home/acadgild/project/data/web/file.xml:0+6718
18/03/19 04:44:30 INFO Executor: Finished task 0.0 in stage 0.0 (TID 0). 2229 bytes result sent to driver
18/03/19 04:44:31 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 23001 ms on localhost (1/1)
```

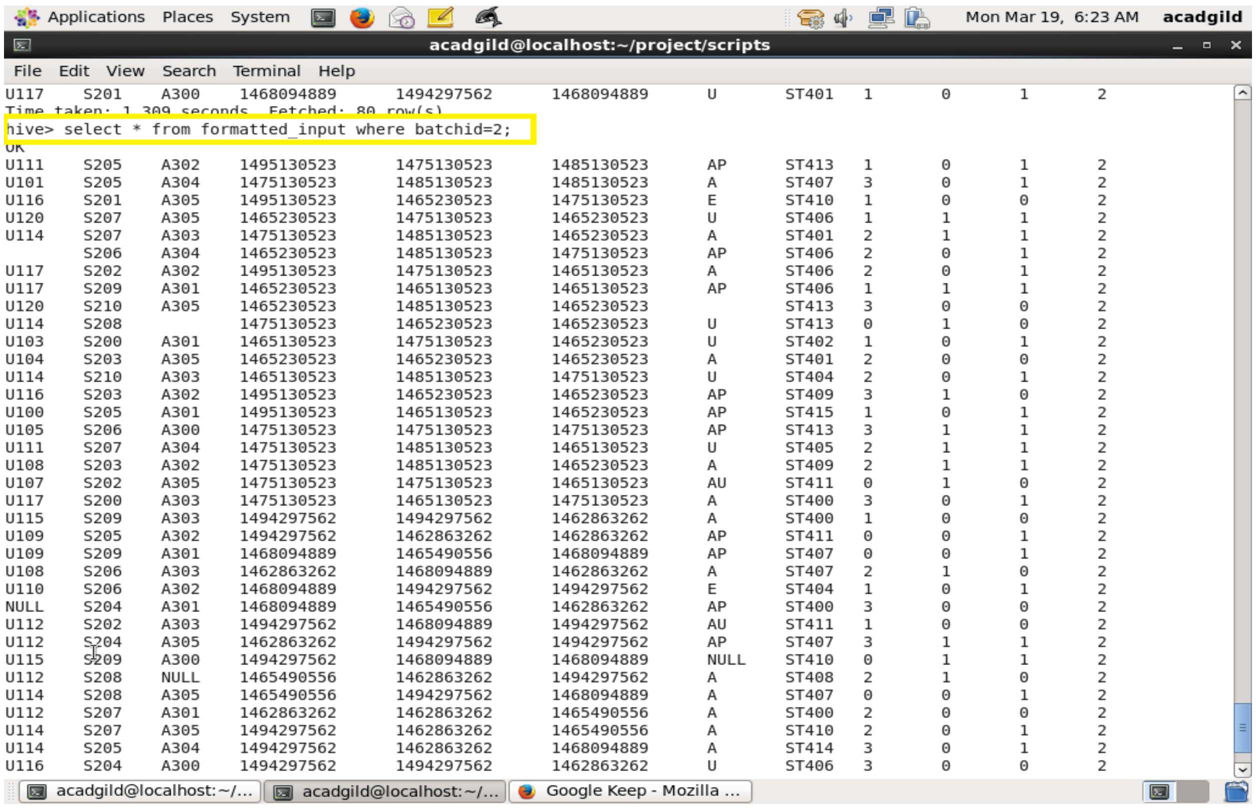

Applications Places System
Mon Mar 19, 6:21 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help

```
18/03/19 05:53:17 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_database: Database(name:default, descripti
on:default database, locationUri:file:/home/acadgild/project/scripts/spark-warehouse, parameters:{})
18/03/19 05:53:18 INFO SparkSqlParser: Parsing command: web_data
18/03/19 05:53:19 INFO SparkSqlParser: Parsing command: CREATE TABLE IF NOT EXISTS project.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 ','

18/03/19 05:53:19 INFO HiveMetaStore: 0: get_database: project
18/03/19 05:53:19 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 05:53:19 INFO HiveMetaStore: 0: get_database: project
18/03/19 05:53:19 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 05:53:20 INFO HiveMetaStore: 0: create_table: Table(tableName:formatted_input, dbName:project, owner:acadgild, creat
eTime:1521418999, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(name:user_id, type:string, comment:nu
ll), FieldSchema(name:song_id, type:string, comment:null), FieldSchema(name:artist_id, type:string, comment:null), FieldSchem
a(name:timestamp, type:string, comment:null), FieldSchema(name:start_ts, type:string, comment:null), FieldSchema(name:end_ts,
 type:string, comment:null), FieldSchema(name:geo_cd, type:string, comment:null), FieldSchema(name:station_id, type:string, c
omment:null), FieldSchema(name:song_end_type, type:int, comment:null), FieldSchema(name:like, type:int, comment:null), FieldS
chema(name:dislike, type:int, comment:null)], location:null, inputFormat:org.apache.hadoop.mapred.TextInputFormat, outputForm
at:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null
, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{field.delim=,, serialization.format=,}), b
ucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedColValueLocation
Maps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:null)], parameters:{}, viewOriginalText:null, viewExpan
dedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(userPrivileges:{}, groupPrivileges:null, rolePrivileg
es:null))
```

acadgild@localhost:~/...
acadgild@localhost:~/...
Google Keep - Mozilla ...

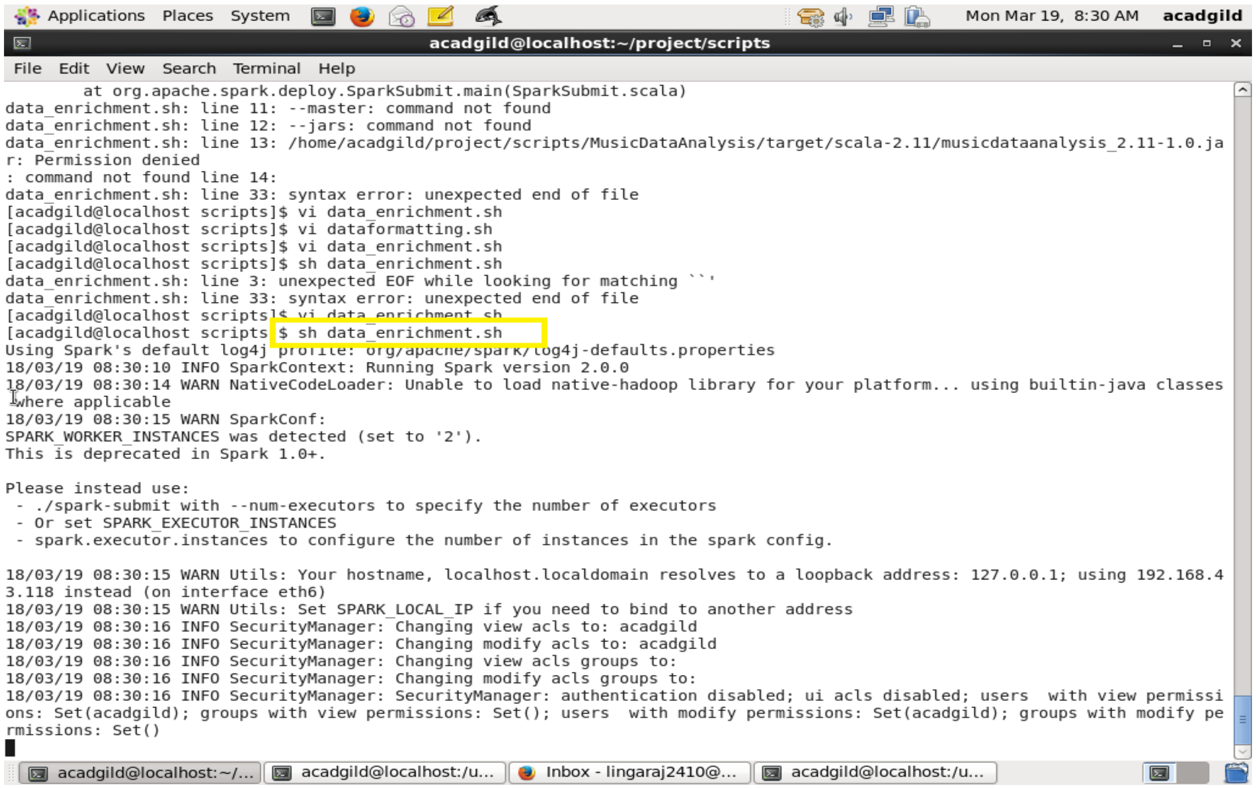

Applications Places System
Mon Mar 19, 6:22 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help

```
18/03/19 05:53:21 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=formatted_input
18/03/19 05:53:21 INFO HiveMetaStore: 0: get_table : db=project tbl=formatted_input
18/03/19 05:53:21 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=formatted_input
18/03/19 05:53:21 INFO HiveMetaStore: 0: get_partition_with_auth : db=project tbl=formatted_input[2]
18/03/19 05:53:21 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_partition_with_auth : db=project tbl=formatte
d_input[2]
18/03/19 05:53:23 INFO FileUtils: Creating directory if it doesn't exist: hdfs://localhost:9000/user/hive/warehouse/project.d
b/formatted_input/batchid=2
18/03/19 05:53:23 INFO SessionState: Could not get hdfsEncryptionShim, it is only applicable to hdfs filesystem.
18/03/19 05:53:25 INFO Hive: Renaming src: file:/home/acadgild/project/data/mob/file.txt, dest: hdfs://localhost:9000/user/hi
ve/warehouse/project.db/formatted_input/batchid=2/file.txt, Status:true
18/03/19 05:53:26 INFO HiveMetaStore: 0: get_partition_with_auth : db=project tbl=formatted_input[2]
18/03/19 05:53:26 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_partition_with_auth : db=project tbl=formatte
d_input[2]
18/03/19 05:53:26 INFO HiveMetaStore: 0: append_partition : db=project tbl=formatted_input[2]
18/03/19 05:53:26 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=append_partition : db=project tbl=formatted_input
[2]
18/03/19 05:53:26 WARN log: Updating partition stats fast for: formatted_input
18/03/19 05:53:26 WARN log: Updated size to 1238
18/03/19 05:53:26 INFO SparkSqlParser: Parsing command: INSERT INTO project.formatted_input
         PARTITION(batchid='2')
         SELECT user_id,
         song_id,
         artist_id,
         unix_timestamp(timestamp,'yyyy-MM-dd HH:mm:ss') AS timestamp,
         unix_timestamp(start_ts,'yyyy-MM-dd HH:mm:ss') AS start_ts,
         unix_timestamp(end_ts,'yyyy-MM-dd HH:mm:ss') AS end_ts,
         geo_cd,
         station_id,
         song_end_type,
         like,
         dislike
         FROM web_data

18/03/19 05:53:26 INFO HiveMetaStore: 0: get_table : db=project tbl=formatted_input
18/03/19 05:53:26 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=formatted_input
18/03/19 05:53:27 INFO CatalystSqlParser: Parsing command: int
18/03/19 05:53:27 INFO CatalystSqlParser: Parsing command: string
18/03/19 05:53:27 INFO CatalystSqlParser: Parsing command: string
```

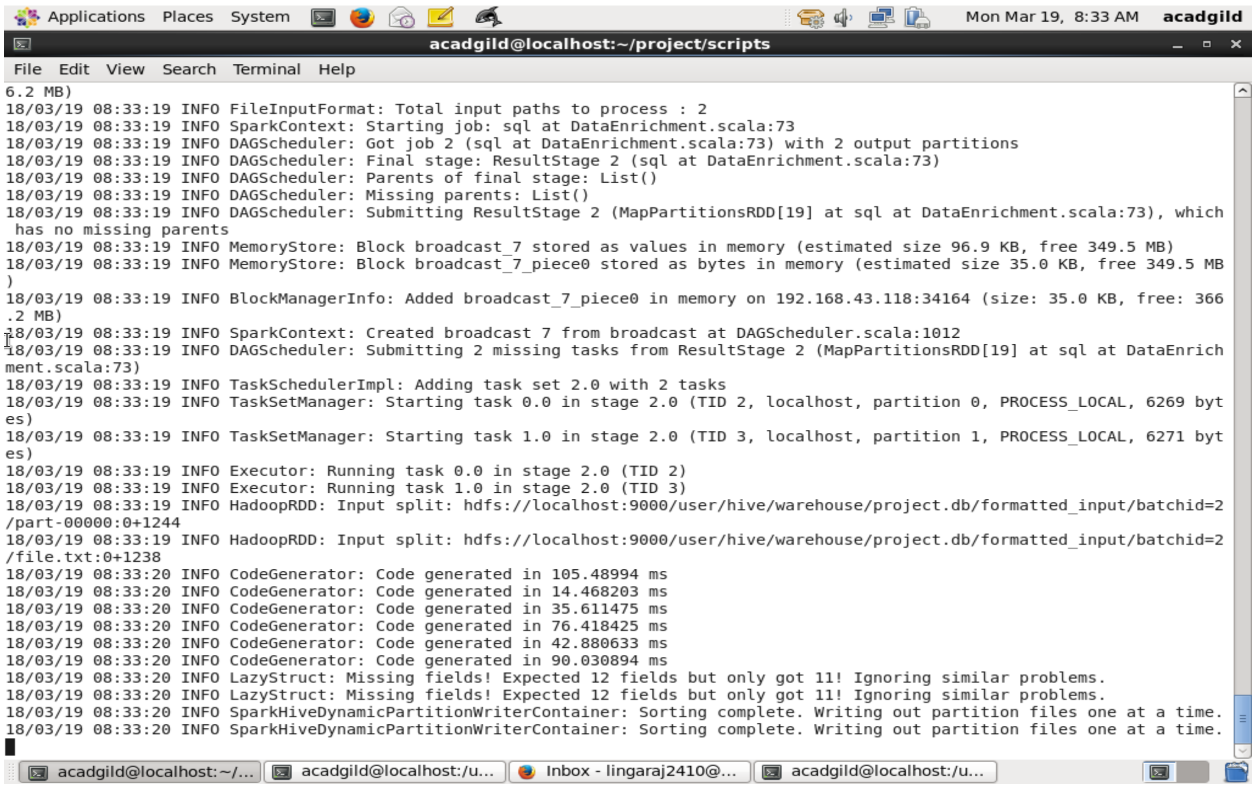


```
U117    S201    A300    1468094889      1494297562      1468094889      U       ST401   1       0       1       2
Time taken: 1.309 seconds, Fetched: 80 row(s)
hive> select * from formatted_input where batchid=2;
OK
U111    S205    A302    1495130523      1475130523      1485130523      AP      ST413   1       0       1       2
U101    S205    A304    1475130523      1485130523      1485130523      A       ST407   3       0       1       2
U116    S201    A305    1495130523      1465230523      1475130523      E       ST410   1       0       0       2
U120    S207    A305    1465230523      1475130523      1465230523      U       ST406   1       1       1       2
U114    S207    A303    1475130523      1485130523      1465230523      A       ST401   2       1       1       2
        S206    A304    1465230523      1485130523      1475130523      AP      ST406   2       0       1       2
U117    S202    A302    1495130523      1475130523      1465130523      A       ST406   2       0       1       2
U117    S209    A301    1465230523      1465130523      1465130523      AP      ST406   1       1       1       2
U120    S210    A305    1465230523      1485130523      1465230523              ST413   3       0       0       2
U114    S208            1475130523      1465230523      1465230523      U       ST413   0       1       0       2
U103    S200    A301    1465130523      1475130523      1465230523      U       ST402   1       0       1       2
U104    S203    A305    1465230523      1465230523      1465230523      A       ST401   2       0       0       2
U114    S210    A303    1465130523      1485130523      1475130523      U       ST404   2       0       1       2
U116    S203    A302    1495130523      1465230523      1465230523      AP      ST409   3       1       0       2
U100    S205    A301    1495130523      1465130523      1465230523      AP      ST415   1       0       1       2
U105    S206    A300    1475130523      1475130523      1475130523      AP      ST413   3       1       1       2
U111    S207    A304    1475130523      1485130523      1465130523      U       ST405   2       1       1       2
U108    S203    A302    1475130523      1485130523      1465230523      A       ST409   2       1       1       2
U107    S202    A305    1475130523      1475130523      1465130523      AU      ST411   0       1       0       2
U117    S200    A303    1475130523      1465130523      1475130523      A       ST400   3       0       1       2
U115    S209    A303    1494297562      1494297562      1462863262      A       ST400   1       0       0       2
U109    S205    A302    1494297562      1462863262      1462863262      AP      ST411   0       0       1       2
U109    S209    A301    1468094889      1465490556      1468094889      AP      ST407   0       0       1       2
U108    S206    A303    1462863262      1468094889      1462863262      A       ST407   2       1       0       2
U110    S206    A302    1468094889      1494297562      1494297562      E       ST404   1       0       1       2
NULL    S204    A301    1468094889      1465490556      1462863262      AP      ST400   3       0       0       2
U112    S202    A303    1494297562      1468094889      1494297562      AU      ST411   1       0       0       2
U112    S204    A305    1462863262      1494297562      1494297562      AP      ST407   3       1       1       2
U115    S209    A300    1494297562      1468094889      1468094889      NULL    ST410   0       1       1       2
U112    S208    NULL    1465490556      1462863262      1494297562      A       ST408   2       1       0       2
U114    S208    A305    1465490556      1494297562      1468094889      A       ST407   0       0       1       2
U112    S207    A301    1462863262      1462863262      1465490556      A       ST400   2       0       0       2
U114    S207    A305    1494297562      1462863262      1465490556      A       ST410   2       0       1       2
U114    S205    A304    1494297562      1462863262      1468094889      A       ST414   3       0       1       2
U116    S204    A300    1494297562      1494297562      1462863262      U       ST406   3       0       0       2
```

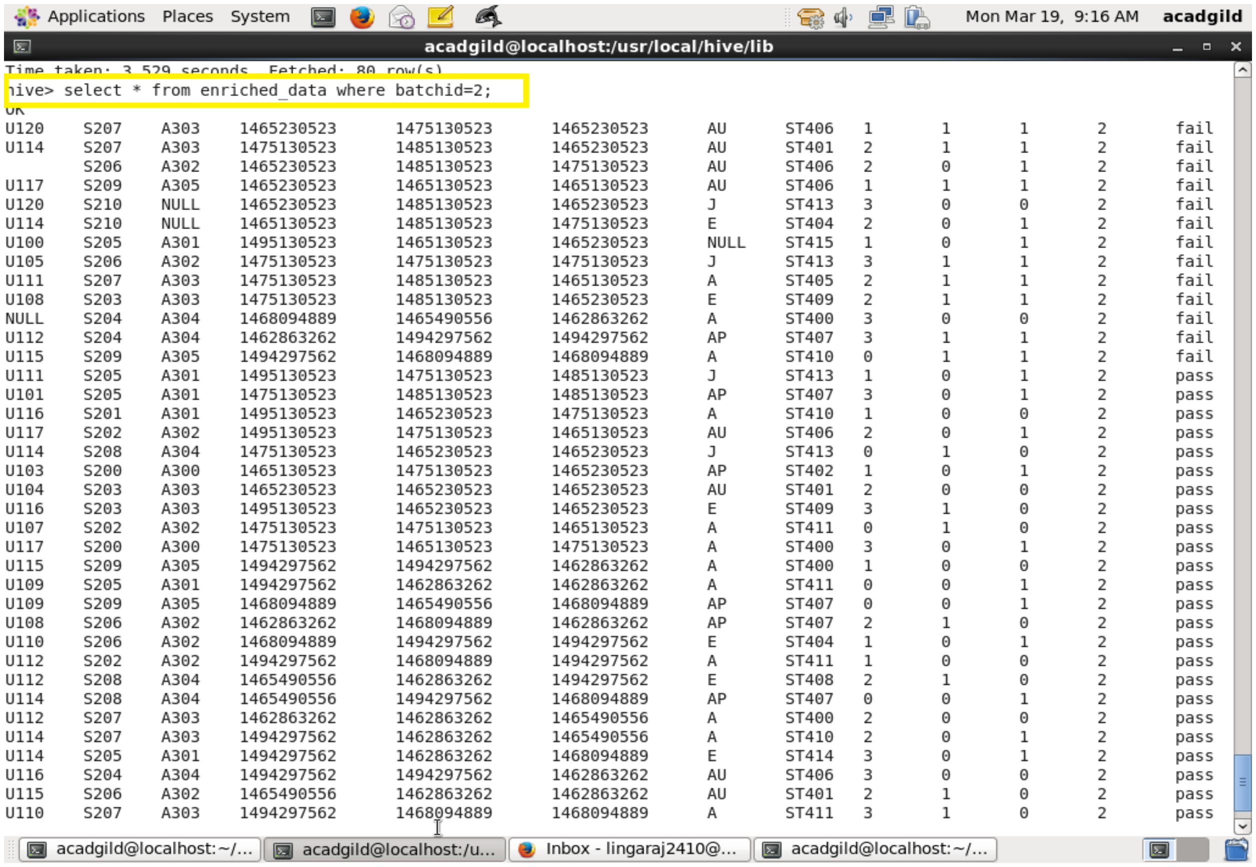
Data Enrichment – Screenshots of execution:



```
        at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
data_enrichment.sh: line 11: --master: command not found
data_enrichment.sh: line 12: --jars: command not found
data_enrichment.sh: line 13: /home/acadgild/project/scripts/MusicDataAnalysis/target/scala-2.11/musicdataanalysis_2.11-1.0.ja
r: Permission denied
: command not found line 14:
data_enrichment.sh: line 33: syntax error: unexpected end of file
[acadgild@localhost scripts]$ vi data_enrichment.sh
[acadgild@localhost scripts]$ vi dataformatting.sh
[acadgild@localhost scripts]$ vi data_enrichment.sh
[acadgild@localhost scripts]$ sh data_enrichment.sh
data_enrichment.sh: line 3: unexpected EOF while looking for matching ``'
data_enrichment.sh: line 33: syntax error: unexpected end of file
[acadgild@localhost scripts]$ vi data_enrichment.sh
[acadgild@localhost scripts]$ sh data_enrichment.sh
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/03/19 08:30:10 INFO SparkContext: Running Spark version 2.0.0
18/03/19 08:30:14 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
 where applicable
18/03/19 08:30:15 WARN SparkConf:
SPARK_WORKER_INSTANCES was detected (set to '2').
This is deprecated in Spark 1.0+.

Please instead use:
 - ./spark-submit with --num-executors to specify the number of executors
 - Or set SPARK_EXECUTOR_INSTANCES
 - spark.executor.instances to configure the number of instances in the spark config.

18/03/19 08:30:15 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.4
3.118 instead (on interface eth6)
18/03/19 08:30:15 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
18/03/19 08:30:16 INFO SecurityManager: Changing view acls to: acadgild
18/03/19 08:30:16 INFO SecurityManager: Changing modify acls to: acadgild
18/03/19 08:30:16 INFO SecurityManager: Changing view acls groups to:
18/03/19 08:30:16 INFO SecurityManager: Changing modify acls groups to:
18/03/19 08:30:16 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissi
ons: Set(acadgild); groups with view permissions: Set(); users  with modify permissions: Set(acadgild); groups with modify pe
rmissions: Set()
```

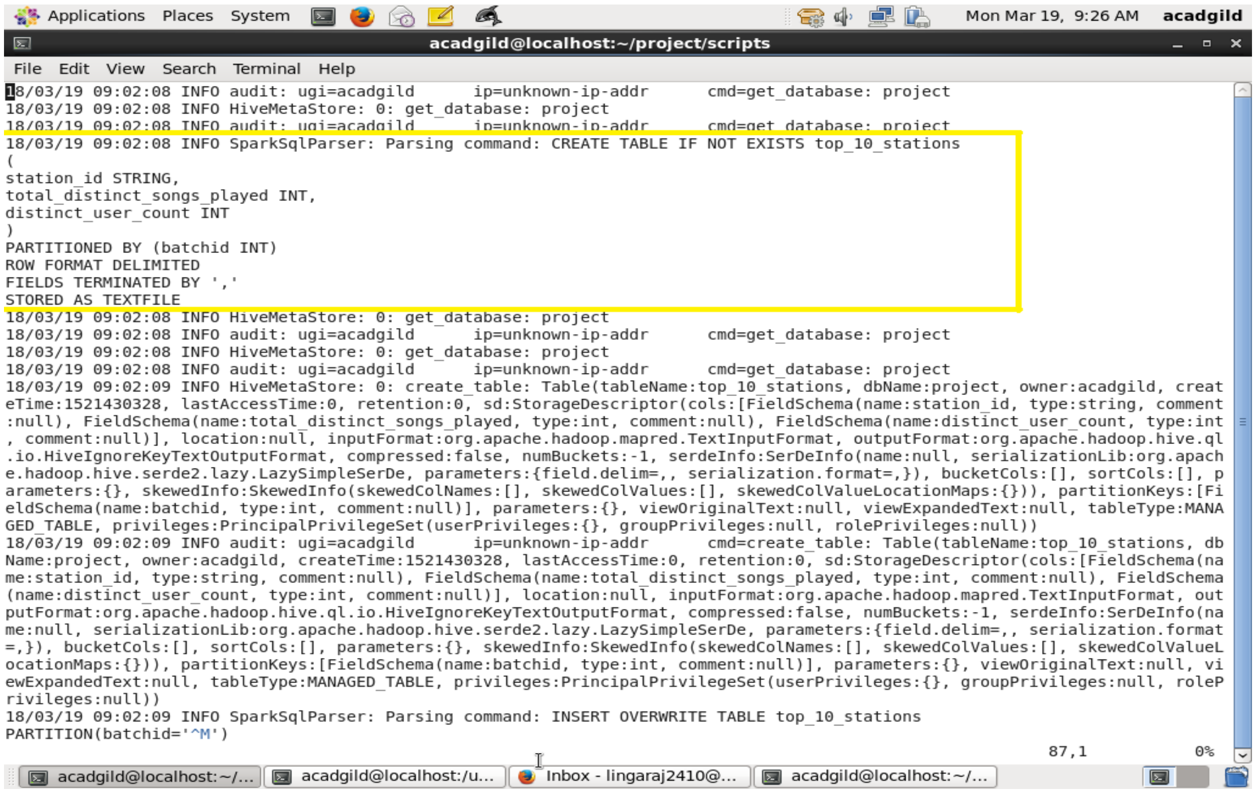

Applications Places System
Mon Mar 19, 8:33 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
acadgild@localhost:~/...
acadgild@localhost:/u...
Inbox - lingaraj2410@...
acadgild@localhost:/u...

```
6.2 MB)
18/03/19 08:33:19 INFO FileInputFormat: Total input paths to process : 2
18/03/19 08:33:19 INFO SparkContext: Starting job: sql at DataEnrichment.scala:73
18/03/19 08:33:19 INFO DAGScheduler: Got job 2 (sql at DataEnrichment.scala:73) with 2 output partitions
18/03/19 08:33:19 INFO DAGScheduler: Final stage: ResultStage 2 (sql at DataEnrichment.scala:73)
18/03/19 08:33:19 INFO DAGScheduler: Parents of final stage: List()
18/03/19 08:33:19 INFO DAGScheduler: Missing parents: List()
18/03/19 08:33:19 INFO DAGScheduler: Submitting ResultStage 2 (MapPartitionsRDD[19] at sql at DataEnrichment.scala:73), which
 has no missing parents
18/03/19 08:33:19 INFO MemoryStore: Block broadcast_7 stored as values in memory (estimated size 96.9 KB, free 349.5 MB)
18/03/19 08:33:19 INFO MemoryStore: Block broadcast_7_piece0 stored as bytes in memory (estimated size 35.0 KB, free 349.5 MB
)
18/03/19 08:33:19 INFO BlockManagerInfo: Added broadcast_7_piece0 in memory on 192.168.43.118:34164 (size: 35.0 KB, free: 366
.2 MB)
18/03/19 08:33:19 INFO SparkContext: Created broadcast 7 from broadcast at DAGScheduler.scala:1012
18/03/19 08:33:19 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 2 (MapPartitionsRDD[19] at sql at DataEnrich
ment.scala:73)
18/03/19 08:33:19 INFO TaskSchedulerImpl: Adding task set 2.0 with 2 tasks
18/03/19 08:33:19 INFO TaskSetManager: Starting task 0.0 in stage 2.0 (TID 2, localhost, partition 0, PROCESS_LOCAL, 6269 byt
es)
18/03/19 08:33:19 INFO TaskSetManager: Starting task 1.0 in stage 2.0 (TID 3, localhost, partition 1, PROCESS_LOCAL, 6271 byt
es)
18/03/19 08:33:19 INFO Executor: Running task 0.0 in stage 2.0 (TID 2)
18/03/19 08:33:19 INFO Executor: Running task 1.0 in stage 2.0 (TID 3)
18/03/19 08:33:19 INFO HadoopRDD: Input split: hdfs://localhost:9000/user/hive/warehouse/project.db/formatted_input/batchid=2
/part-00000:0+1244
18/03/19 08:33:19 INFO HadoopRDD: Input split: hdfs://localhost:9000/user/hive/warehouse/project.db/formatted_input/batchid=2
/file.txt:0+1238
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 105.48994 ms
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 14.468203 ms
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 35.611475 ms
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 76.418425 ms
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 42.880633 ms
18/03/19 08:33:20 INFO CodeGenerator: Code generated in 90.030894 ms
18/03/19 08:33:20 INFO LazyStruct: Missing fields! Expected 12 fields but only got 11! Ignoring similar problems.
18/03/19 08:33:20 INFO LazyStruct: Missing fields! Expected 12 fields but only got 11! Ignoring similar problems.
18/03/19 08:33:20 INFO SparkHiveDynamicPartitionWriterContainer: Sorting complete. Writing out partition files one at a time.
18/03/19 08:33:20 INFO SparkHiveDynamicPartitionWriterContainer: Sorting complete. Writing out partition files one at a time.
```

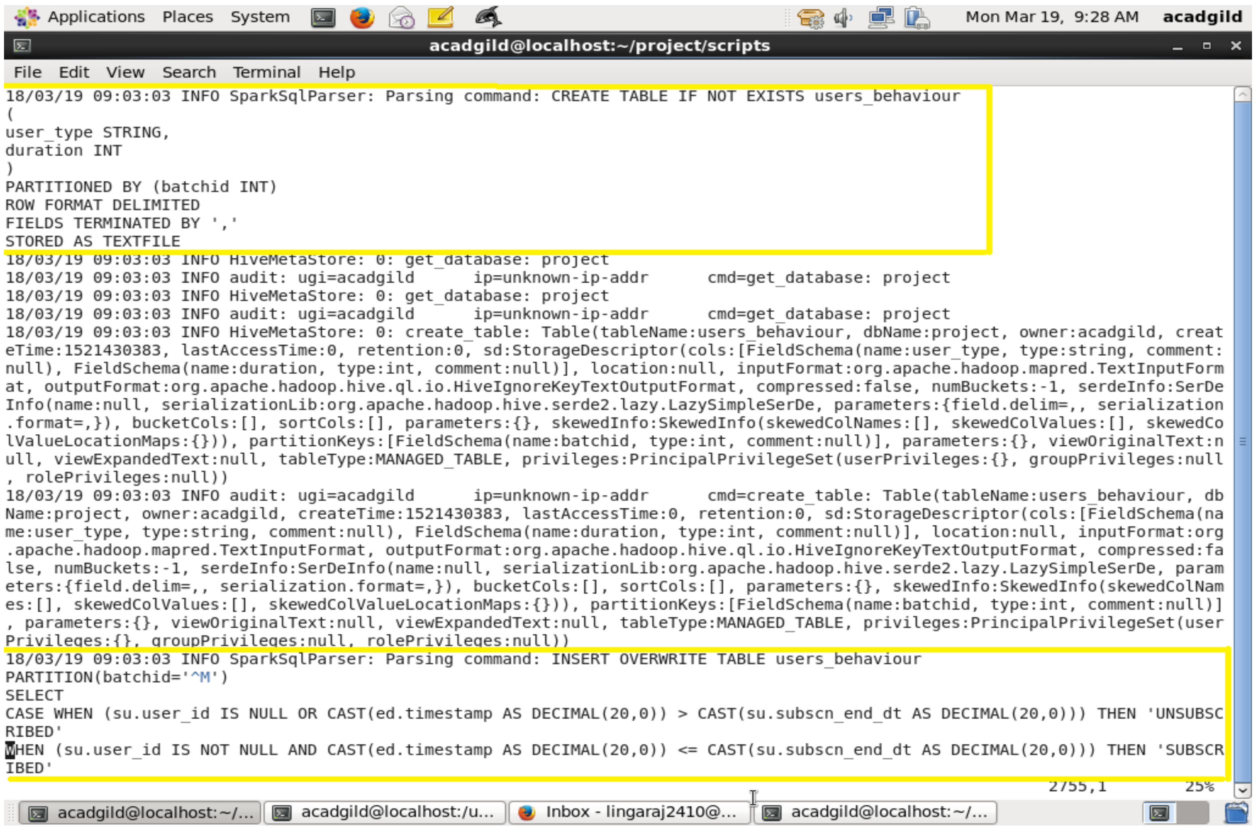


```
Time taken: 3.529 seconds, Fetched: 80 row(s)
hive> select * from enriched_data where batchid=2;
OK
U120    S207    A303    1465230523      1475130523      1465230523      AU      ST406   1       1       1       2       fail
U114    S207    A303    1475130523      1485130523      1465230523      AU      ST401   2       1       1       2       fail
        S206    A302    1465230523      1485130523      1475130523      AU      ST406   2       0       1       2       fail
U117    S209    A305    1465230523      1465130523      1465130523      AU      ST406   1       1       1       2       fail
U120    S210    NULL    1465230523      1485130523      1465230523      J       ST413   3       0       0       2       fail
U114    S210    NULL    1465130523      1485130523      1475130523      E       ST404   2       0       1       2       fail
U100    S205    A301    1495130523      1465130523      1465230523      NULL    ST415   1       0       1       2       fail
U105    S206    A302    1475130523      1475130523      1475130523      J       ST413   3       1       1       2       fail
U111    S207    A303    1475130523      1485130523      1465130523      A       ST405   2       1       1       2       fail
U108    S203    A303    1475130523      1485130523      1465230523      E       ST409   2       1       1       2       fail
NULL    S204    A304    1468094889      1465490556      1462863262      A       ST400   3       0       0       2       fail
U112    S204    A304    1462863262      1494297562      1494297562      AP      ST407   3       1       1       2       fail
U115    S209    A305    1494297562      1468094889      1468094889      A       ST410   0       1       1       2       fail
U111    S205    A301    1495130523      1475130523      1485130523      J       ST413   1       0       1       2       pass
U101    S205    A301    1475130523      1485130523      1485130523      AP      ST407   3       0       1       2       pass
U116    S201    A301    1495130523      1465230523      1475130523      A       ST410   1       0       0       2       pass
U117    S202    A302    1495130523      1475130523      1465130523      AU      ST406   2       0       1       2       pass
U114    S208    A304    1475130523      1465230523      1465230523      J       ST413   0       1       0       2       pass
U103    S200    A300    1465130523      1475130523      1465230523      AP      ST402   1       0       1       2       pass
U104    S203    A303    1465230523      1465230523      1465230523      AU      ST401   2       0       0       2       pass
U116    S203    A303    1495130523      1465230523      1465230523      E       ST409   3       1       0       2       pass
U107    S202    A302    1475130523      1475130523      1465130523      A       ST411   0       1       0       2       pass
U117    S200    A300    1475130523      1465130523      1475130523      A       ST400   3       0       1       2       pass
U115    S209    A305    1494297562      1494297562      1462863262      A       ST400   1       0       0       2       pass
U109    S205    A301    1494297562      1462863262      1462863262      A       ST411   0       0       1       2       pass
U109    S209    A305    1468094889      1465490556      1468094889      AP      ST407   0       0       1       2       pass
U108    S206    A302    1462863262      1468094889      1462863262      AP      ST407   2       1       0       2       pass
U110    S206    A302    1468094889      1494297562      1494297562      E       ST404   1       0       1       2       pass
U112    S202    A302    1494297562      1468094889      1494297562      A       ST411   1       0       0       2       pass
U112    S208    A304    1465490556      1462863262      1494297562      E       ST408   2       1       0       2       pass
U114    S208    A304    1465490556      1494297562      1468094889      AP      ST407   0       0       1       2       pass
U112    S207    A303    1462863262      1462863262      1465490556      A       ST400   2       0       0       2       pass
U114    S207    A303    1494297562      1462863262      1465490556      A       ST410   2       0       1       2       pass
U114    S205    A301    1494297562      1462863262      1468094889      E       ST414   3       0       1       2       pass
U116    S204    A304    1494297562      1494297562      1462863262      AU      ST406   3       0       0       2       pass
U115    S206    A302    1465490556      1462863262      1462863262      AU      ST401   2       1       0       2       pass
U110    S207    A303    1494297562      1468094889      1468094889      A       ST411   3       1       0       2       pass
```

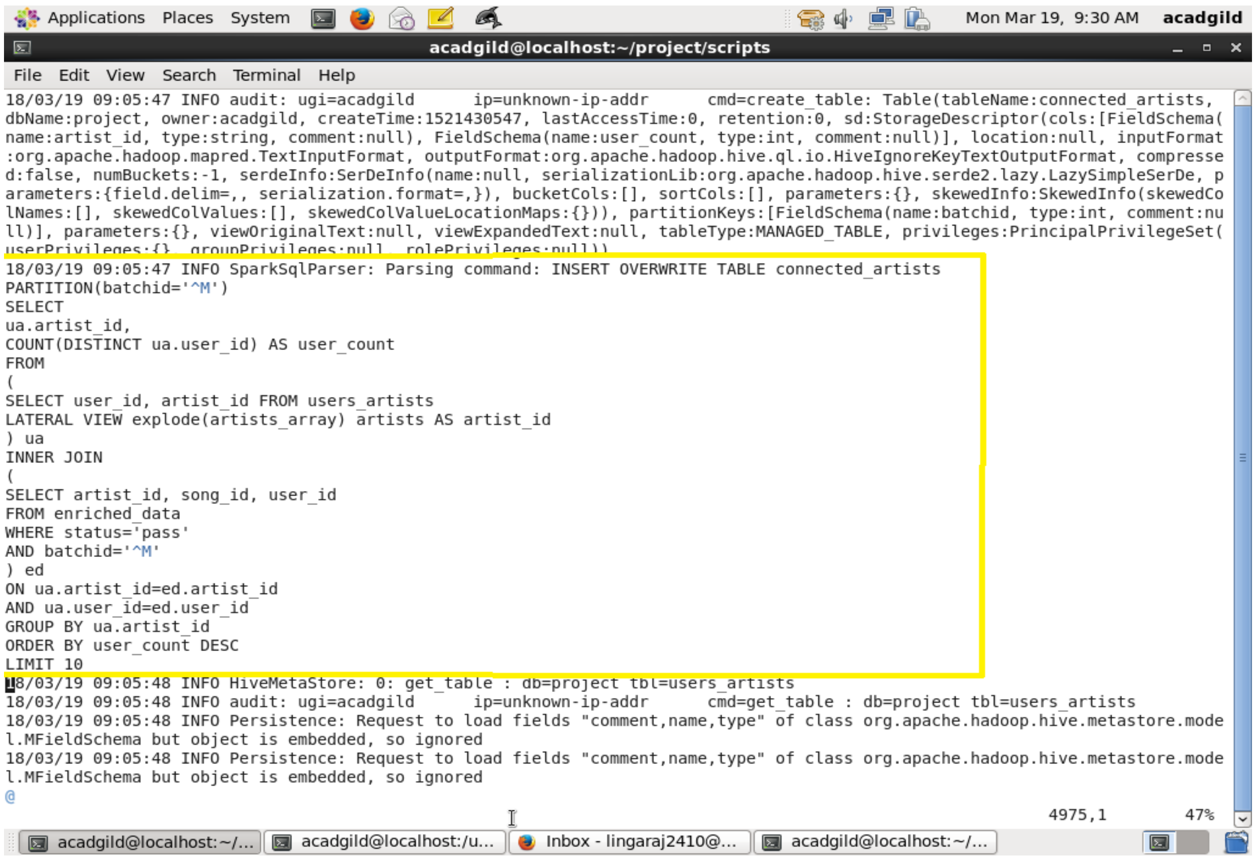
Data analysis – screenshots of execution:


Applications Places System
Mon Mar 19, 9:26 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
acadgild@localhost:~/...
acadgild@localhost:/u...
Inbox - lingaraj2410@...
acadgild@localhost:~/...

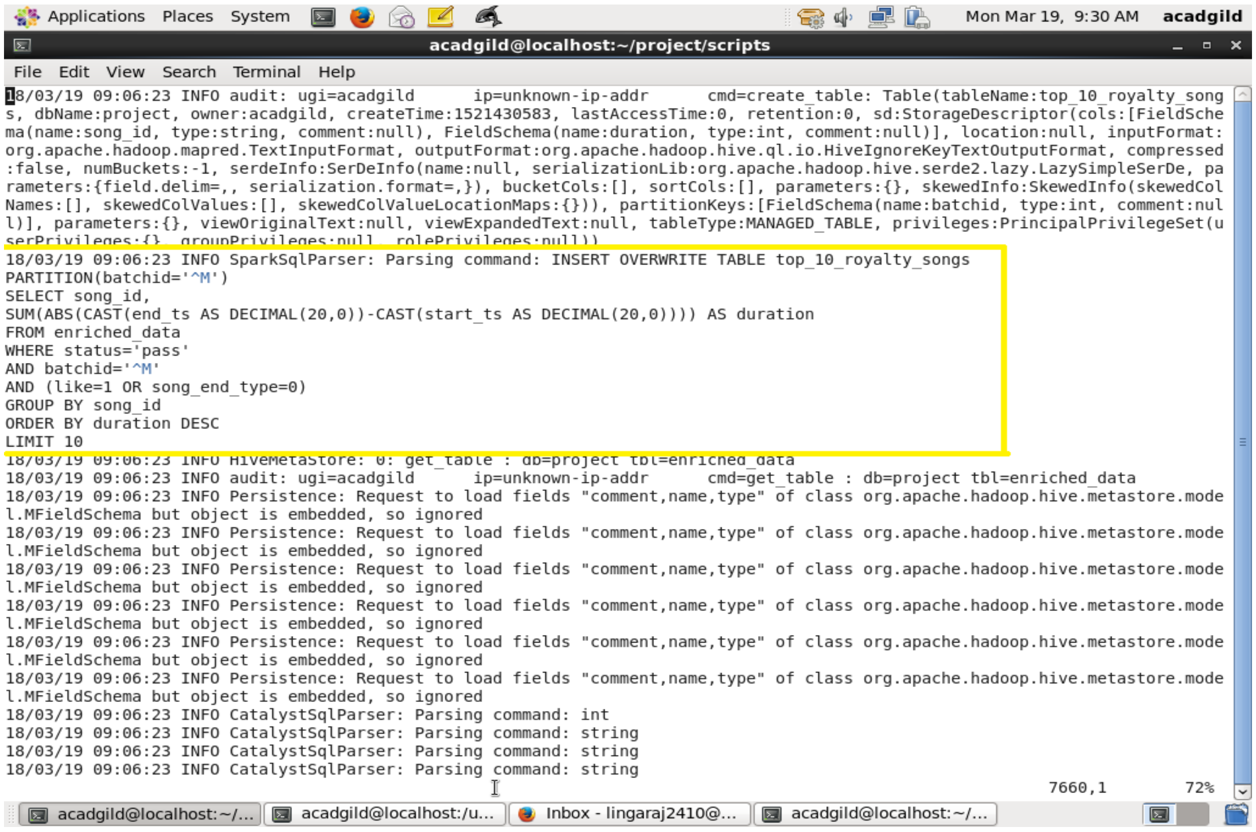
```
18/03/19 09:02:08 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:02:08 INFO HiveMetaStore: 0: get_database: project
18/03/19 09:02:08 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:02:08 INFO SparkSqlParser: Parsing command: 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
18/03/19 09:02:08 INFO HiveMetaStore: 0: get_database: project
18/03/19 09:02:08 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:02:08 INFO HiveMetaStore: 0: get_database: project
18/03/19 09:02:08 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:02:09 INFO HiveMetaStore: 0: create_table: Table(tableName:top_10_stations, dbName:project, owner:acadgild, creat
eTime:1521430328, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(name:station_id, type:string, comment
:null), FieldSchema(name:total_distinct_songs_played, type:int, comment:null), FieldSchema(name:distinct_user_count, type:int
, comment:null)], location:null, inputFormat:org.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.hadoop.hive.ql
.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apach
e.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{field.delim=,, serialization.format=,}), bucketCols:[], sortCols:[], p
arameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[Fi
eldSchema(name:batchid, type:int, comment:null)], parameters:{}, viewOriginalText:null, viewExpandedText:null, tableType:MANA
GED_TABLE, privileges:PrincipalPrivilegeSet(userPrivileges:{}, groupPrivileges:null, rolePrivileges:null))
18/03/19 09:02:09 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_table: Table(tableName:top_10_stations, db
Name:project, owner:acadgild, createTime:1521430328, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(na
me:station_id, type:string, comment:null), FieldSchema(name:total_distinct_songs_played, type:int, comment:null), FieldSchema
(name:distinct_user_count, type:int, comment:null)], location:null, inputFormat:org.apache.hadoop.mapred.TextInputFormat, out
putFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1, serdeInfo:SerDeInfo(na
me:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{field.delim=,, serialization.format
=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedColValueL
ocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:null)], parameters:{}, viewOriginalText:null, vi
ewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(userPrivileges:{}, groupPrivileges:null, roleP
rivileges:null))
18/03/19 09:02:09 INFO SparkSqlParser: Parsing command: INSERT OVERWRITE TABLE top_10_stations
PARTITION(batchid='^M')
                                                                                        87,1          0%
```


Applications Places System
Mon Mar 19, 9:28 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
acadgild@localhost:~/...
acadgild@localhost:/u...
Inbox - lingaraj2410@...
acadgild@localhost:~/...

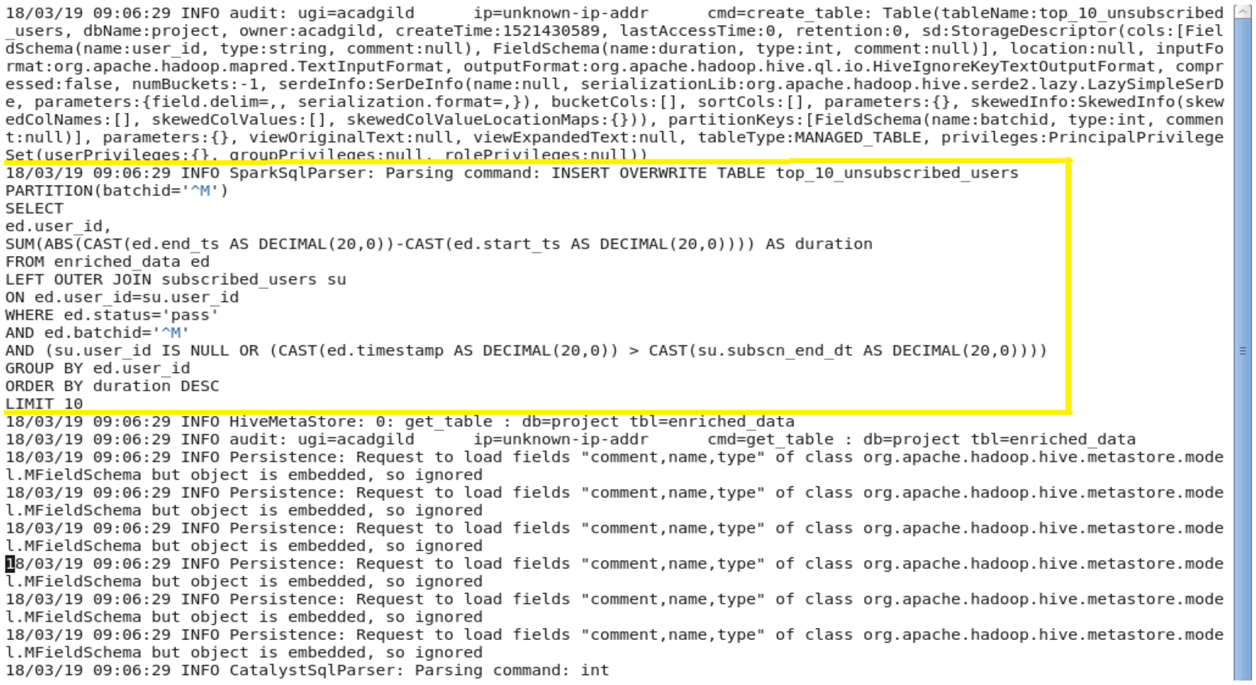
```
18/03/19 09:03:03 INFO SparkSqlParser: Parsing command: CREATE TABLE IF NOT EXISTS users_behaviour
(
user_type STRING,
duration INT
)
PARTITIONED BY (batchid INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
18/03/19 09:03:03 INFO HiveMetaStore: 0: get_database: project
18/03/19 09:03:03 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:03:03 INFO HiveMetaStore: 0: get_database: project
18/03/19 09:03:03 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_database: project
18/03/19 09:03:03 INFO HiveMetaStore: 0: create_table: Table(tableName:users_behaviour, dbName:project, owner:acadgild, creat
eTime:1521430383, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(name:user_type, type:string, comment:
null), FieldSchema(name:duration, type:int, comment:null)], location:null, inputFormat:org.apache.hadoop.mapred.TextInputForm
at, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1, serdeInfo:SerDe
Info(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{field.delim=,, serialization
.format=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedCo
lValueLocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:null)], parameters:{}, viewOriginalText:n
ull, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(userPrivileges:{}, groupPrivileges:null
, rolePrivileges:null))
18/03/19 09:03:03 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_table: Table(tableName:users_behaviour, db
Name:project, owner:acadgild, createTime:1521430383, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(na
me:user_type, type:string, comment:null), FieldSchema(name:duration, type:int, comment:null)], location:null, inputFormat:org
.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:fa
lse, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, param
eters:{field.delim=,, serialization.format=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedColNam
es:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:null)]
, parameters:{}, viewOriginalText:null, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(user
Privileges:{}, groupPrivileges:null, rolePrivileges:null))
18/03/19 09:03:03 INFO SparkSqlParser: Parsing command: INSERT OVERWRITE TABLE users_behaviour
PARTITION(batchid='^M')
SELECT
CASE WHEN (su.user_id IS NULL OR CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'UNSUBSC
RIBED'
WHEN (su.user_id IS NOT NULL AND CAST(ed.timestamp AS DECIMAL(20,0)) <= CAST(su.subscn_end_dt AS DECIMAL(20,0))) THEN 'SUBSCR
IBED'
                                                                                        2755,1        25%
```


Applications Places System
Mon Mar 19, 9:30 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
acadgild@localhost:~/... acadgild@localhost:/u... Inbox - lingaraj2410@... acadgild@localhost:~/...

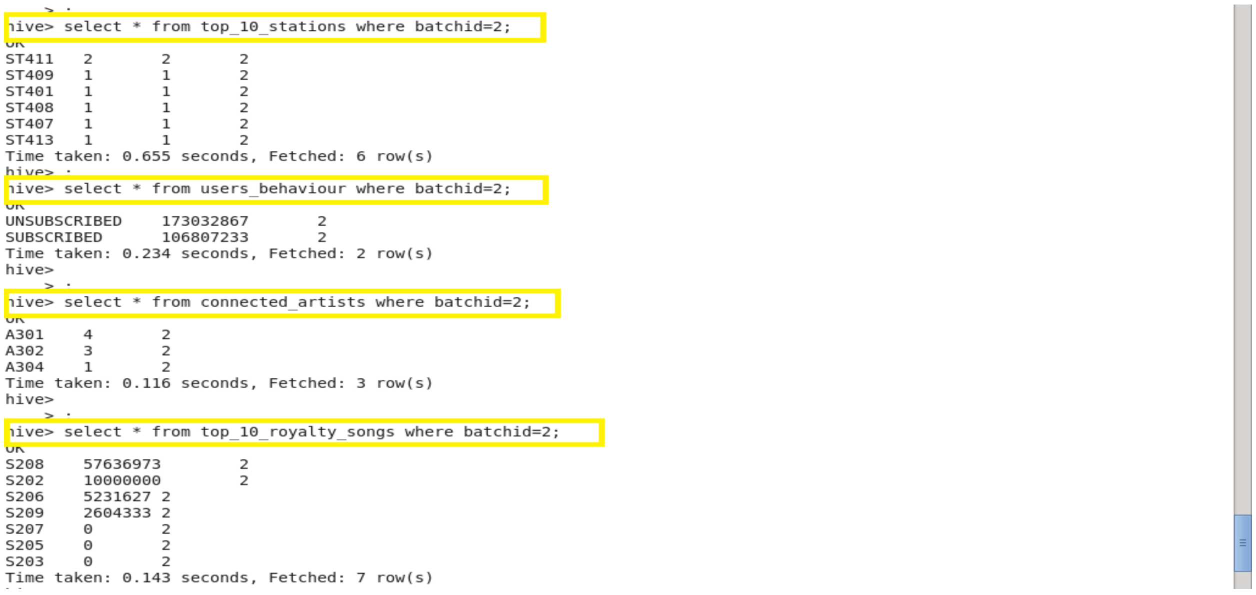
```
18/03/19 09:05:47 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_table: Table(tableName:connected_artists,
dbName:project, owner:acadgild, createTime:1521430547, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSchema(
name:artist_id, type:string, comment:null), FieldSchema(name:user_count, type:int, comment:null)], location:null, inputFormat
:org.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compresse
d:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, p
arameters:{field.delim=,, serialization.format=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedCo
lNames:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:nu
ll)], parameters:{}, viewOriginalText:null, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(
userPrivileges:{}, groupPrivileges:null, rolePrivileges:null))
18/03/19 09:05:47 INFO SparkSqlParser: Parsing command: INSERT OVERWRITE TABLE connected_artists
PARTITION(batchid='^M')
SELECT
ua.artist_id,
COUNT(DISTINCT ua.user_id) AS user_count
FROM
(
SELECT user_id, artist_id FROM users_artists
LATERAL VIEW explode(artists_array) artists AS artist_id
) ua
INNER JOIN
(
SELECT artist_id, song_id, user_id
FROM enriched_data
WHERE status='pass'
AND batchid='^M'
) ed
ON ua.artist_id=ed.artist_id
AND ua.user_id=ed.user_id
GROUP BY ua.artist_id
ORDER BY user_count DESC
LIMIT 10
18/03/19 09:05:48 INFO HiveMetaStore: 0: get_table : db=project tbl=users_artists
18/03/19 09:05:48 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=users_artists
18/03/19 09:05:48 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:05:48 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
@
                                                                                                 4975,1         47%
```


Applications Places System
Mon Mar 19, 9:30 AM acadgild
acadgild@localhost:~/project/scripts
File Edit View Search Terminal Help
acadgild@localhost:~/... acadgild@localhost:/u... Inbox - lingaraj2410@... acadgild@localhost:~/...

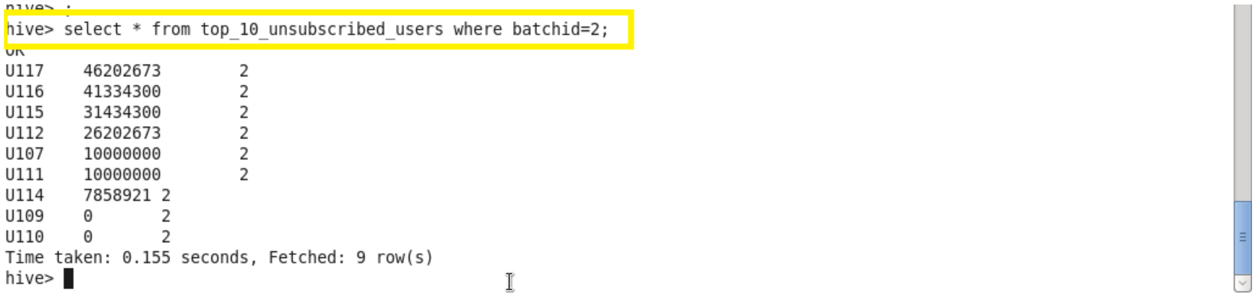
```
18/03/19 09:06:23 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_table: Table(tableName:top_10_royalty_song
s, dbName:project, owner:acadgild, createTime:1521430583, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldSche
ma(name:song_id, type:string, comment:null), FieldSchema(name:duration, type:int, comment:null)], location:null, inputFormat:
org.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed
:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, pa
rameters:{field.delim=,, serialization.format=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skewedCol
Names:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, comment:nul
l)], parameters:{}, viewOriginalText:null, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(u
serPrivileges:{}, groupPrivileges:null, rolePrivileges:null))
18/03/19 09:06:23 INFO SparkSqlParser: Parsing command: INSERT OVERWRITE TABLE top_10_royalty_songs
PARTITION(batchid='^M')
SELECT song_id,
SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL(20,0)))) AS duration
FROM enriched_data
WHERE status='pass'
AND batchid='^M'
AND (like=1 OR song_end_type=0)
GROUP BY song_id
ORDER BY duration DESC
LIMIT 10
18/03/19 09:06:23 INFO HiveMetaStore: 0: get_table : db=project tbl=enriched_data
18/03/19 09:06:23 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=enriched_data
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:23 INFO CatalystSqlParser: Parsing command: int
18/03/19 09:06:23 INFO CatalystSqlParser: Parsing command: string
18/03/19 09:06:23 INFO CatalystSqlParser: Parsing command: string
18/03/19 09:06:23 INFO CatalystSqlParser: Parsing command: string
                                                                                                 7660,1         72%
```



```
18/03/19 09:06:29 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=create_table: Table(tableName:top_10_unsubscribed
_users, dbName:project, owner:acadgild, createTime:1521430589, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[Fiel
dSchema(name:user_id, type:string, comment:null), FieldSchema(name:duration, type:int, comment:null)], location:null, inputFo
rmat:org.apache.hadoop.mapred.TextInputFormat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compr
essed:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerD
e, parameters:{field.delim=,, serialization.format=,}), bucketCols:[], sortCols:[], parameters:{}, skewedInfo:SkewedInfo(skew
edColNames:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[FieldSchema(name:batchid, type:int, commen
t:null)], parameters:{}, viewOriginalText:null, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilege
Set(userPrivileges:{}, groupPrivileges:null, rolePrivileges:null))
18/03/19 09:06:29 INFO SparkSqlParser: Parsing command: INSERT OVERWRITE TABLE top_10_unsubscribed_users
PARTITION(batchid='^M')
SELECT
ed.user_id,
SUM(ABS(CAST(ed.end_ts AS DECIMAL(20,0))-CAST(ed.start_ts AS DECIMAL(20,0)))) AS duration
FROM enriched_data ed
LEFT OUTER JOIN subscribed_users su
ON ed.user_id=su.user_id
WHERE ed.status='pass'
AND ed.batchid='^M'
AND (su.user_id IS NULL OR (CAST(ed.timestamp AS DECIMAL(20,0)) > CAST(su.subscn_end_dt AS DECIMAL(20,0))))
GROUP BY ed.user_id
ORDER BY duration DESC
LIMIT 10
18/03/19 09:06:29 INFO HiveMetaStore: 0: get_table : db=project tbl=enriched_data
18/03/19 09:06:29 INFO audit: ugi=acadgild      ip=unknown-ip-addr      cmd=get_table : db=project tbl=enriched_data
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO Persistence: Request to load fields "comment,name,type" of class org.apache.hadoop.hive.metastore.mode
l.MFieldSchema but object is embedded, so ignored
18/03/19 09:06:29 INFO CatalystSqlParser: Parsing command: int
```



```
    > ;
hive> select * from top_10_stations where batchid=2;
OK
ST411   2       2       2
ST409   1       1       2
ST401   1       1       2
ST408   1       1       2
ST407   1       1       2
ST413   1       1       2
Time taken: 0.655 seconds, Fetched: 6 row(s)
hive> ;
hive> select * from users_behaviour where batchid=2;
OK
UNSUBSCRIBED    173032867       2
SUBSCRIBED      106807233       2
Time taken: 0.234 seconds, Fetched: 2 row(s)
hive>
    > ;
hive> select * from connected_artists where batchid=2;
OK
A301    4       2
A302    3       2
A304    1       2
Time taken: 0.116 seconds, Fetched: 3 row(s)
hive>
    > ;
hive> select * from top_10_royalty_songs where batchid=2;
OK
S208    57636973        2
S202    10000000        2
S206    5231627 2
S209    2604333 2
S207    0       2
S205    0       2
S203    0       2
Time taken: 0.143 seconds, Fetched: 7 row(s)
```



```
hive> ;
hive> select * from top_10_unsubscribed_users where batchid=2;
OK
U117    46202673        2
U116    41334300        2
U115    31434300        2
U112    26202673        2
U107    10000000        2
U111    10000000        2
U114    7858921 2
U109    0       2
U110    0       2
Time taken: 0.155 seconds, Fetched: 9 row(s)
hive>
```