CREATING HBASE TABLES

****

- SONG_ARTIST

```
hbase(main):065:0> create 'Song_Artist_Map','details'
0 row(s) in 1.2730 seconds

=> Hbase::Table - Song_Artist_Map
hbase(main):066:0>
hbase(main):067:0*
hbase(main):068:0*
hbase(main):069:0* put 'Song_Artist_Map','S200','details:artist_id','A300'
0 row(s) in 0.0220 seconds

hbase(main):070:0>
hbase(main):071:0* put 'Song_Artist_Map','S201','details:artist_id','A301'
0 row(s) in 0.0070 seconds

hbase(main):072:0>
hbase(main):073:0* put 'Song_Artist_Map','S202','details:artist_id','A302'
0 row(s) in 0.0120 seconds

hbase(main):074:0>
hbase(main):075:0* put 'Song_Artist_Map','S203','details:artist_id','A303'
0 row(s) in 0.0120 seconds

hbase(main):076:0>
hbase(main):077:0* put 'Song_Artist_Map','S204','details:artist_id','A304'
0 row(s) in 0.0140 seconds

hbase(main):078:0>
hbase(main):079:0* put 'Song_Artist_Map','S205','details:artist_id','A301'
0 row(s) in 0.0090 seconds

hbase(main):080:0>
hbase(main):081:0* put 'Song_Artist_Map','S206','details:artist_id','A302'

hbase(main):082:0>
hbase(main):083:0* put 'Song_Artist_Map','S207','details:artist_id','A303'
0 row(s) in 0.0070 seconds

hbase(main):084:0>
hbase(main):085:0* put 'Song_Artist_Map','S208','details:artist_id','A304'
0 row(s) in 0.0080 seconds

hbase(main):086:0>
hbase(main):087:0* put 'Song_Artist_Map','S209','details:artist_id','A305'
```

OUTPUT

```
hbase(main):088:0> scan 'Song_Artist_Map'
ROW                                   COLUMN+CELL
 S200                                 column=details:artist_id, timestamp=1522128036223, value=A300
 S201                                 column=details:artist_id, timestamp=1522128036275, value=A301
 S202                                 column=details:artist_id, timestamp=1522128036331, value=A302
 S203                                 column=details:artist_id, timestamp=1522128036402, value=A303
 S204                                 column=details:artist_id, timestamp=1522128036478, value=A304
 S205                                 column=details:artist_id, timestamp=1522128036534, value=A301
 S206                                 column=details:artist_id, timestamp=1522128036595, value=A302
 S207                                 column=details:artist_id, timestamp=1522128036654, value=A303
 S208                                 column=details:artist_id, timestamp=1522128036709, value=A304
 S209                                 column=details:artist_id, timestamp=1522128201838, value=A305
10 row(s) in 0.0620 seconds
```

****

USER_ARTIST

```
hbase(main):089:0> create 'User_Artist_Map','details'
0 row(s) in 1.2810 seconds

=> Hbase::Table - User_Artist_Map
hbase(main):090:0>
hbase(main):091:0*
hbase(main):092:0*
hbase(main):093:0* put 'User_Artist_Map','U100','details:artist_id','A300&A301&A302'
0 row(s) in 0.0220 seconds

hbase(main):094:0>
hbase(main):095:0* put 'User_Artist_Map','U101','details:artist_id','A301&A302'
0 row(s) in 0.0120 seconds

hbase(main):096:0>
hbase(main):097:0* put 'User_Artist_Map','U102','details:artist_id','A302'
0 row(s) in 0.0090 seconds

hbase(main):098:0>
hbase(main):099:0* put 'User_Artist_Map','U103','details:artist_id','A303&A301&A302'
0 row(s) in 0.0160 seconds

hbase(main):100:0>
hbase(main):101:0* put 'User_Artist_Map','U104','details:artist_id','A304&A301'
0 row(s) in 0.0170 seconds

hbase(main):102:0>
hbase(main):103:0* put 'User_Artist_Map','U105','details:artist_id','A305&A301&A302'
0 row(s) in 0.0280 seconds

hbase(main):104:0>
hbase(main):105:0* put 'User_Artist_Map','U106','details:artist_id','A301&A302'
```

```
hbase(main):106:0>
hbase(main):107:0* put 'User_Artist_Map','U107','details:artist_id','A302'
0 row(s) in 0.0060 seconds

hbase(main):108:0>
hbase(main):109:0* put 'User_Artist_Map','U108','details:artist_id','A300&A303&A304'
0 row(s) in 0.0120 seconds

hbase(main):110:0>
hbase(main):111:0* put 'User_Artist_Map','U109','details:artist_id','A301&A303'
0 row(s) in 0.0460 seconds

hbase(main):112:0>
hbase(main):113:0* put 'User_Artist_Map','U110','details:artist_id','A302&A301'
0 row(s) in 0.0100 seconds

hbase(main):114:0>
hbase(main):115:0* put 'User_Artist_Map','U111','details:artist_id','A303&A301'
0 row(s) in 0.0130 seconds

hbase(main):116:0>
hbase(main):117:0* put 'User_Artist_Map','U112','details:artist_id','A304&A301'
0 row(s) in 0.0140 seconds

hbase(main):118:0>
hbase(main):119:0* put 'User_Artist_Map','U113','details:artist_id','A305&A302'
0 row(s) in 0.0070 seconds
```

```
hbase(main):120:0>
hbase(main):121:0* put 'User_Artist_Map','U114','details:artist_id','A300&A301&A302'
0 row(s) in 0.0130 seconds
```

OUTPUT

```
hbase(main):122:0> scan 'User_Artist_Map'
ROW                                   COLUMN+CELL
 U100                                 column=details:artist_id, timestamp=1522128268269, value=A300&A301&A302
 U101                                 column=details:artist_id, timestamp=1522128268334, value=A301&A302
 U102                                 column=details:artist_id, timestamp=1522128268462, value=A302
 U103                                 column=details:artist_id, timestamp=1522128268672, value=A303&A301&A302
 U104                                 column=details:artist_id, timestamp=1522128268900, value=A304&A301
 U105                                 column=details:artist_id, timestamp=1522128269086, value=A305&A301&A302
 U106                                 column=details:artist_id, timestamp=1522128269211, value=A301&A302
 U107                                 column=details:artist_id, timestamp=1522128269264, value=A302
 U108                                 column=details:artist_id, timestamp=1522128269317, value=A300&A303&A304
 U109                                 column=details:artist_id, timestamp=1522128269510, value=A301&A303
 U110                                 column=details:artist_id, timestamp=1522128269563, value=A302&A301
 U111                                 column=details:artist_id, timestamp=1522128269609, value=A303&A301
 U112                                 column=details:artist_id, timestamp=1522128269680, value=A304&A301
 U113                                 column=details:artist_id, timestamp=1522128269729, value=A305&A302
 U114                                 column=details:artist_id, timestamp=1522128654165, value=A300&A301&A302
15 row(s) in 0.1280 seconds
```

****

SUBSCRIBED_USERS

```
hbase(main):037:0> create 'Subscribed_Users','details'
0 row(s) in 1.2850 seconds

=> Hbase::Table - Subscribed_Users
hbase(main):038:0>
hbase(main):039:0*
hbase(main):040:0*
hbase(main):041:0* put 'Subscribed_Users','U100','details:subscription_start_date','1465230523'
0 row(s) in 0.0330 seconds

hbase(main):042:0>
hbase(main):043:0* put 'Subscribed_Users','U100','details:subscription_end_date','1465130523'
0 row(s) in 0.0100 seconds

hbase(main):044:0>
hbase(main):045:0* put 'Subscribed_Users','U101','details:subscription_start_date','1465230523'
0 row(s) in 0.0110 seconds

hbase(main):046:0>
hbase(main):047:0* put 'Subscribed_Users','U101','details:subscription_end_date','1475130523'
0 row(s) in 0.0100 seconds

hbase(main):048:0>
hbase(main):049:0* put 'Subscribed_Users','U102','details:subscription_start_date','1465230523'
0 row(s) in 0.0090 seconds

hbase(main):050:0>
hbase(main):051:0* put 'Subscribed_Users','U102','details:subscription_end_date','1475130523'
0 row(s) in 0.0100 seconds

hbase(main):052:0>
hbase(main):053:0* put 'Subscribed_Users','U103','details:subscription_start_date','1465230523'


hbase(main):054:0>
hbase(main):055:0* put 'Subscribed_Users','U103','details:subscription_end_date','1475130523'
0 row(s) in 0.0230 seconds

hbase(main):056:0>
hbase(main):057:0* put 'Subscribed_Users','U104','details:subscription_start_date','1465230523'
0 row(s) in 0.0200 seconds

hbase(main):058:0>
hbase(main):059:0* put 'Subscribed_Users','U104','details:subscription_end_date','1475130523'
0 row(s) in 0.0110 seconds

hbase(main):060:0>
hbase(main):061:0* put 'Subscribed_Users','U105','details:subscription_start_date','1465230523'
0 row(s) in 0.0130 seconds

hbase(main):062:0>
hbase(main):063:0* put 'Subscribed_Users','U105','details:subscription_end_date','1475130523'
0 row(s) in 0.0260 seconds

hbase(main):064:0>
hbase(main):065:0* put 'Subscribed_Users','U106','details:subscription_start_date','1465230523'
0 row(s) in 0.0090 seconds

hbase(main):066:0>
hbase(main):067:0* put 'Subscribed_Users','U106','details:subscription_end_date','1485130523'
0 row(s) in 0.0110 seconds

hbase(main):068:0>
hbase(main):069:0* put 'Subscribed_Users','U107','details:subscription_start_date','1465230523'
0 row(s) in 0.0090 seconds
```

```
hbase(main):070:0>
hbase(main):071:0* put 'Subscribed_Users','U107','details:subscription_end_date','1455130523'
0 row(s) in 0.0080 seconds

hbase(main):072:0>
hbase(main):073:0* put 'Subscribed_Users','U108','details:subscription_start_date','1465230523'
0 row(s) in 0.0080 seconds

hbase(main):074:0>
hbase(main):075:0* put 'Subscribed_Users','U108','details:subscription_end_date','1465230623'
0 row(s) in 0.0100 seconds

hbase(main):076:0>
hbase(main):077:0* put 'Subscribed_Users','U109','details:subscription_start_date','1465230523'
0 row(s) in 0.0160 seconds

hbase(main):078:0>
hbase(main):079:0* put 'Subscribed_Users','U109','details:subscription_end_date','1475130523'
0 row(s) in 0.0110 seconds

hbase(main):080:0>
hbase(main):081:0* put 'Subscribed_Users','U110','details:subscription_start_date','1465230523'
0 row(s) in 0.0090 seconds

hbase(main):082:0>
hbase(main):083:0* put 'Subscribed_Users','U110','details:subscription_end_date','1475130523'
0 row(s) in 0.0100 seconds

hbase(main):084:0>
hbase(main):085:0* put 'Subscribed_Users','U111','details:subscription_start_date','1465230523'
0 row(s) in 0.0120 seconds

hbase(main):086:0>
hbase(main):087:0* put 'Subscribed_Users','U111','details:subscription_end_date','1475130523'
0 row(s) in 0.0130 seconds

hbase(main):088:0>
hbase(main):089:0* put 'Subscribed_Users','U112','details:subscription_start_date','1465230523'
0 row(s) in 0.0180 seconds

hbase(main):090:0>
hbase(main):091:0* put 'Subscribed_Users','U112','details:subscription_end_date','1475130523'
0 row(s) in 0.0230 seconds

hbase(main):092:0>
hbase(main):093:0* put 'Subscribed_Users','U113','details:subscription_start_date','1465230523'
0 row(s) in 0.0170 seconds

hbase(main):094:0>
hbase(main):095:0* put 'Subscribed_Users','U113','details:subscription_end_date','1485130523'
0 row(s) in 0.0100 seconds

hbase(main):096:0>
hbase(main):097:0* put 'Subscribed_Users','U114','details:subscription_start_date','1465230523'
0 row(s) in 0.0190 seconds

hbase(main):098:0>
hbase(main):099:0* put 'Subscribed_Users','U114','details:subscription_end_date','1468130523'
```

OUTPUT

```
hbase(main):064:0> scan 'Subscribed_Users'
ROW                                   COLUMN+CELL
 U100                                 column=details:subscription_end_date, timestamp=1522127864179, value=1465130523
 U100                                 column=details:subscription_start_date, timestamp=1522127863963, value=1465230523
 U101                                 column=details:subscription_end_date, timestamp=1522127864613, value=1475130523
 U101                                 column=details:subscription_start_date, timestamp=1522127864399, value=1465230523
 U102                                 column=details:subscription_end_date, timestamp=1522127864864, value=1475130523
 U102                                 column=details:subscription_start_date, timestamp=1522127864735, value=1465230523
 U103                                 column=details:subscription_end_date, timestamp=1522127865073, value=1475130523
 U103                                 column=details:subscription_start_date, timestamp=1522127865005, value=1465230523
 U104                                 column=details:subscription_end_date, timestamp=1522127865259, value=1475130523
 U104                                 column=details:subscription_start_date, timestamp=1522127865176, value=1465230523
 U105                                 column=details:subscription_end_date, timestamp=1522127865534, value=1475130523
 U105                                 column=details:subscription_start_date, timestamp=1522127865346, value=1465230523
 U106                                 column=details:subscription_end_date, timestamp=1522127865713, value=1485130523
 U106                                 column=details:subscription_start_date, timestamp=1522127865622, value=1465230523
 U107                                 column=details:subscription_end_date, timestamp=1522127866113, value=1455130523
 U107                                 column=details:subscription_start_date, timestamp=1522127865879, value=1465230523
 U108                                 column=details:subscription_end_date, timestamp=1522127866341, value=1465230623
 U108                                 column=details:subscription_start_date, timestamp=1522127866286, value=1465230523
 U109                                 column=details:subscription_end_date, timestamp=1522127866490, value=1475130523
 U109                                 column=details:subscription_start_date, timestamp=1522127866419, value=1465230523
 U110                                 column=details:subscription_end_date, timestamp=1522127866636, value=1475130523
 U110                                 column=details:subscription_start_date, timestamp=1522127866555, value=1465230523
 U111                                 column=details:subscription_end_date, timestamp=1522127866832, value=1475130523
 U111                                 column=details:subscription_start_date, timestamp=1522127866741, value=1465230523
 U112                                 column=details:subscription_end_date, timestamp=1522127867103, value=1475130523
 U112                                 column=details:subscription_start_date, timestamp=1522127866929, value=1465230523
 U113                                 column=details:subscription_end_date, timestamp=1522127867373, value=1485130523
 U113                                 column=details:subscription_start_date, timestamp=1522127867300, value=1465230523
 U114                                 column=details:subscription_end_date, timestamp=1522127875334, value=1468130523
 U114                                 column=details:subscription_start_date, timestamp=1522127867479, value=1465230523
```

****

STATION_CODE

```
hbase(main):001:0> create 'Station_Geo_Map','details'
0 row(s) in 3.4910 seconds

=> Hbase::Table - Station_Geo_Map
hbase(main):002:0>
hbase(main):003:0*
hbase(main):004:0*
hbase(main):005:0* put 'Station_Geo_Map','ST400','details:geo_cd','A'
0 row(s) in 0.5680 seconds

hbase(main):006:0>
hbase(main):007:0* put 'Station_Geo_Map','ST401','details:geo_cd','AU'
0 row(s) in 0.0130 seconds

hbase(main):008:0>
hbase(main):009:0* put 'Station_Geo_Map','ST402','details:geo_cd','AP'
0 row(s) in 0.0250 seconds

hbase(main):010:0>
hbase(main):011:0* put 'Station_Geo_Map','ST403','details:geo_cd','J'
0 row(s) in 0.0070 seconds

hbase(main):012:0>
hbase(main):013:0* put 'Station_Geo_Map','ST404','details:geo_cd','E'
0 row(s) in 0.0130 seconds

hbase(main):014:0>
hbase(main):015:0* put 'Station_Geo_Map','ST405','details:geo_cd','A'
0 row(s) in 0.0070 seconds
```

```
hbase(main):016:0>
hbase(main):017:0* put 'Station_Geo_Map','ST406','details:geo_cd','AU'
```

```
hbase(main):013:0* put 'Station_Geo_Map','ST404','details:geo_cd','E'
0 row(s) in 0.0130 seconds

hbase(main):014:0>
hbase(main):015:0* put 'Station_Geo_Map','ST405','details:geo_cd','A'
0 row(s) in 0.0070 seconds

hbase(main):016:0>
hbase(main):017:0* put 'Station_Geo_Map','ST406','details:geo_cd','AU'
0 row(s) in 0.0080 seconds

hbase(main):018:0>
hbase(main):019:0* put 'Station_Geo_Map','ST407','details:geo_cd','AP'
0 row(s) in 0.0090 seconds

hbase(main):020:0>
hbase(main):021:0* put 'Station_Geo_Map','ST408','details:geo_cd','E'
0 row(s) in 0.0330 seconds

hbase(main):022:0>
hbase(main):023:0* put 'Station_Geo_Map','ST409','details:geo_cd','E'
0 row(s) in 0.0210 seconds

hbase(main):024:0>
hbase(main):025:0* put 'Station_Geo_Map','ST410','details:geo_cd','A'
0 row(s) in 0.0560 seconds

hbase(main):026:0>
hbase(main):027:0* put 'Station_Geo_Map','ST411','details:geo_cd','A'
0 row(s) in 0.0120 seconds
```

OUTPUT

```
hbase(main):036:0> scan 'Station_Geo_Map'
ROW                   COLUMN+CELL
 ST400                column=details:geo_cd, timestamp=1522126817093, value=A
 ST401                column=details:geo_cd, timestamp=1522126817248, value=AU
 ST402                column=details:geo_cd, timestamp=1522126817361, value=AP
 ST403                column=details:geo_cd, timestamp=1522126817531, value=J
 ST404                column=details:geo_cd, timestamp=1522126817715, value=E
 ST405                column=details:geo_cd, timestamp=1522126817813, value=A
 ST406                column=details:geo_cd, timestamp=1522126817908, value=AU
 ST407                column=details:geo_cd, timestamp=1522126817959, value=AP
 ST408                column=details:geo_cd, timestamp=1522126818038, value=E
 ST409                column=details:geo_cd, timestamp=1522126818171, value=E
 ST410                column=details:geo_cd, timestamp=1522126818349, value=A
 ST411                column=details:geo_cd, timestamp=1522126818592, value=A
 ST412                column=details:geo_cd, timestamp=1522126818675, value=AP
 ST413                column=details:geo_cd, timestamp=1522126818726, value=J
 ST414                column=details:geo_cd, timestamp=1522126852226, value=E
15 row(s) in 0.3220 seconds
```

****

CREATING HIVE TABLE FROM HBASE LOOK-UP TABLES

- **Station_Geo_Map**

```
hive> create external table Station_Geo_Map(stationid String,geo_cd string)
    >
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    > with serdeproperties ("hbase.columns.mapping"=":key,details:geo_cd")
    >
    > tblproperties("hbase.table.name"="Station_Geo_Map");
OK
Time taken: 6.929 seconds
hive>
```

Output

```
hive> select * from Station_Geo_Map;
OK
ST400   A
ST401   AU
ST402   AP
ST403   J
ST404   E
ST405   A
ST406   AU
ST407   AP
ST408   E
ST409   E
ST410   A
ST411   A
ST412   AP
ST413   J
ST414   E
Time taken: 2.457 seconds, Fetched: 15 row(s)
hive>
```

****

Subscribed Users (Start)

```
hive> create external table Subscribed_Users_start(user_id String,subscription_s
tart_date string)
    >
    >
    >
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    >
    >
    > with serdeproperties ("hbase.columns.mapping"=":key,details:subscription_s
tart_date")
    >
    >
    >
    > tblproperties("hbase.table.name"="Subscribed_Users");
OK
Time taken: 0.847 seconds
hive>
```

Output

```
hive> Select * from Subscribed_Users_start
    > ;
OK
U100    1465230523
U101    1465230523
U102    1465230523
U103    1465230523
U104    1465230523
U105    1465230523
U106    1465230523
U107    1465230523
U108    1465230523
U109    1465230523
U110    1465230523
U111    1465230523
U112    1465230523
U113    1465230523
U114    1465230523
Time taken: 0.324 seconds, Fetched: 15 row(s)
```

Converting the Start_ts To ToDate Format

Command

```
grunt> A = LOAD '/home/acadgild/Desktop/ProjectData/user-subscn.txt' using PigStorage(',') AS (User_id:chararray,start_ts:cha
rarray,end_ts:chararray);
2018-05-09 11:28:28,615 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
grunt> B = FOREACH A GENERATE ToString( ToDate((long) start_ts * 1000), 'yyyy-MM-dd') as Start;
2018-05-09 11:28:31,982 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 1 ti
me(s).
grunt> dump;
```

Output

```
Total records written : 15
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local903004308_0001


2018-05-09 11:28:38,397 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:28:38,399 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:28:38,408 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:28:38,436 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success
!
2018-05-09 11:28:38,444 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-05-09 11:28:38,445 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-05-09 11:28:38,481 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2018-05-09 11:28:38,482 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
(2016-06-06)
```

acadgild@localhost:~

****

Subscribed Users(End)

```
hive> create external table Subscribed_Users_end(user_id String,subscription_end_date string)
    >
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    > with serdeproperties ("hbase.columns.mapping"=":key,details:subscription_end_date")
    >
    > tblproperties("hbase.table.name"="Subscribed_Users");
OK
Time taken: 0.447 seconds
hive>
```

Output

```
hive> Select * from Subscribed_Users_end;
OK
U100    1465130523
U101    1475130523
U102    1475130523
U103    1475130523
U104    1475130523
U105    1475130523
U106    1485130523
U107    1455130523
U108    1465230623
U109    1475130523
U110    1475130523
U111    1475130523
U112    1475130523
U113    1485130523
U114    1468130523
Time taken: 0.285 seconds, Fetched: 15 row(s)
```

Converting the End_ts to ToDate Format,

Command,

```
grunt> A = LOAD '/home/acadgild/Desktop/ProjectData/user-subsch.txt' using PigStorage(',') AS (User_id:chararray,start_ts:cha
rarray,end_ts:chararray);
2018-05-09 11:23:18,697 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-05-09 11:23:18,715 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 1 ti
me(s).
grunt> B = FOREACH A GENERATE ToString( ToDate((long) end_ts * 1000), 'yyyy-MM-dd') as End;
2018-05-09 11:23:22.894 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 2 ti
me(s).
grunt> dump
```

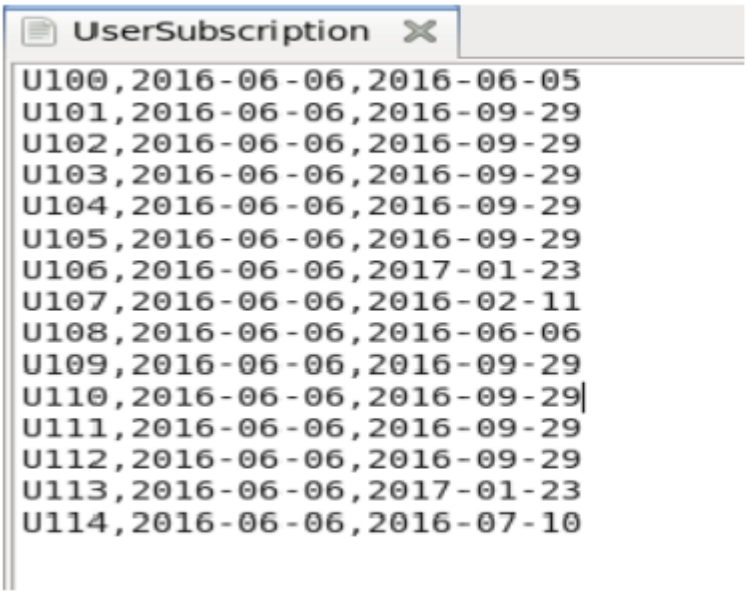
OutPut

```
Total records written : 15
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1821258382_0002


2018-05-09 11:23:26,300 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:23:26,302 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:23:26,306 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processNam
e=JobTracker, sessionId= - already initialized
2018-05-09 11:23:26,329 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success
!
2018-05-09 11:23:26,330 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-05-09 11:23:26,331 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2018-05-09 11:23:26,367 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2018-05-09 11:23:26,367 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(2016-06-05)
(2016-09-29)
(2016-09-29)
(2016-09-29)
(2016-09-29)
(2016-09-29)
(2017-01-23)
(2016-02-11)
(2016-06-05)
(2016-09-29)
(2016-09-29)
(2016-09-29)
(2016-09-29)
(2017-01-23)
(2016-07-18)
```

********** Subscribed_Users Data converted to ToDate Format**********



****

- **Song_Artist_Map**

```
hive> create external table Song_Artist_Map(song_id String,artist_id string)
    >
    >
    >
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    >
    >
    > with serdeproperties ("hbase.columns.mapping"=":key,details:artist_id")
    >
    >
    >
    > tblproperties("hbase.table.name"="Song_Artist_Map");
OK
Time taken: 0.641 seconds
```

Output

```
hive> Select * from Song_Artist_Map;
OK
S200    A300
S201    A301
S202    A302
S203    A303
S204    A304
S205    A301
S206    A302
S207    A303
S208    A304
S209    A305
Time taken: 0.365 seconds, Fetched: 10 row(s)
```

****

- **User_Artist_Map**

```
hive> create external table User_Artist_Map(user_id String,artist_id array<String>)
    >
    >
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' COLLECTION ITEMS TERMINATED BY '&'
    >
    >
    >
    >
    >
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    >
    >
    > with serdeproperties ("hbase.columns.mapping"=":key,details:artist_id")
    >
    >
    >
    > tblproperties("hbase.table.name"="User_Artist_Map");
OK
Time taken: 0.49 seconds
```

Output

```
hive> Select * from User_Artist_Map;
OK
U100    ["A300","A301","A302"]
U101    ["A301","A302"]
U102    ["A302"]
U103    ["A303","A301","A302"]
U104    ["A304","A301"]
U105    ["A305","A301","A302"]
U106    ["A301","A302"]
U107    ["A302"]
U108    ["A300","A303","A304"]
U109    ["A301","A303"]
U110    ["A302","A301"]
U111    ["A303","A301"]
U112    ["A304","A301"]
U113    ["A305","A302"]
U114    ["A300","A301","A302"]
Time taken: 0.423 seconds, Fetched: 15 row(s)
```

****

LOADING HIVE TABLES IN SPARK

- Song_Artist_Map

```
scala>  sqlContext.sql("CREATE TABLE IF NOT EXISTS Song_Artist_Map1(song_id string,artist_id string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '
\n'");
res4: org.apache.spark.sql.DataFrame = [result: string]

scala> sqlContext.sql("LOAD DATA LOCAL INPATH 'file:///home/cloudera/Desktop/ProjectData/song-artist.txt' INTO table Song_Artist_Map1").show;
+------+
|result|
+------+
+------+
```

OutPut

```
scala> sqlContext.sql("Select * from Song_Artist_Map1").show;
+-------+---------+
|song_id|artist_id|
+-------+---------+
|   S200|     A300|
|   S201|     A301|
|   S202|     A302|
|   S203|     A303|
|   S204|     A304|
|   S205|     A301|
|   S206|     A302|
|   S207|     A303|
|   S208|     A304|
|   S209|     A305|
+-------+---------+
```

- Station_Geo_Map

```
scala>  sqlContext.sql("CREATE TABLE IF NOT EXISTS Station_Geo_Map(station_id string,geo_cd string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\
n'");
res8: org.apache.spark.sql.DataFrame = [result: string]

scala> sqlContext.sql("LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/ProjectData/stn-geocd.txt'  INTO table Station_Geo_Map").show;
+------+
|result|
+------+
+------+
```

OutPut

```
scala> sqlContext.sql("Select * from Station_Geo_Map").show;
+----------+------+
|station_id|geo_cd|
+----------+------+
|     ST400|     A|
|     ST401|    AU|
|     ST402|    AP|
|     ST403|     J|
|     ST404|     E|
|     ST405|     A|
|     ST406|    AU|
|     ST407|    AP|
|     ST408|     E|
|     ST409|     E|
|     ST410|     A|
|     ST411|     A|
|     ST412|    AP|
|     ST413|     J|
|     ST414|     E|
+----------+------+
```

- Subscribed_Users_Start

Command,

```
scala> sqlContext.sql("CREATE TABLE  Subscribed(User_id String,start_ts String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
")
18/05/09 11:52:05 WARN HiveMetaStore: Location: hdfs://localhost:8020/user/hive/warehouse/subscribed specified for non-extern
al table:subscribed
res5: org.apache.spark.sql.DataFrame = []

scala> sqlContext.sql("LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/ProjectData/UserSubscription' INTO TABLE Subscribed")
res6: org.apache.spark.sql.DataFrame = []

scala> sqlContext.sql("select * from Subscribed").show
18/05/09 11:53:15 WARN LazyStruct: Extra bytes detected at the end of the row! Ignoring similar problems.
```

OutPut

```
scala> sqlContext.sql("select * from Subscribed").show
18/05/09 11:53:15 WARN LazyStruct: Extra bytes detected at the end of the row! Ignoring similar problems.
+-------+----------+
|User_id|  start_ts|
+-------+----------+
|   U100|2016-05-06|
|   U101|2016-05-06|
|   U102|2016-05-06|
|   U103|2016-05-06|
|   U104|2016-05-06|
|   U105|2016-05-06|
|   U106|2016-05-06|
|   U107|2016-05-06|
|   U108|2016-05-06|
|   U109|2016-05-06|
|   U110|2016-05-06|
|   U111|2016-05-06|
|   U112|2016-05-06|
|   U113|2016-05-06|
|   U114|2016-05-06|
|       |      null|
+-------+----------+
```

- Subscribed_Users_End

Command,

```
scala> sqlContext.sql("CREATE TABLE  Nonsubscribed(User_id String,end_ts String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',
'")
18/05/09 12:00:32 WARN HiveMetaStore: Location: hdfs://localhost:8020/user/hive/warehouse/nonsubscribed specified for non-ext
ernal table:nonsubscribed
res17: org.apache.spark.sql.DataFrame = []

scala> sqlContext.sql("LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/ProjectData/UserSubscription' INTO TABLE Nonsubscribed"
)
res18: org.apache.spark.sql.DataFrame = []
```

OutPut

```
scala> sqlContext.sql("select * from Nonsubscribed").show
18/05/09 12:01:03 WARN LazyStruct: Extra bytes detected at the end of the row! Ignoring similar problems.
+-------+----------+
|User_id|    end_ts|
+-------+----------+
|   U100|2016-06-06|
|   U101|2016-06-06|
|   U102|2016-06-06|
|   U103|2016-06-06|
|   U104|2016-06-06|
|   U105|2016-06-06|
|   U106|2016-06-06|
|   U107|2016-06-06|
|   U108|2016-06-06|
|   U109|2016-06-06|
|   U110|2016-06-06|
|   U111|2016-06-06|
|   U112|2016-06-06|
|   U113|2016-06-06|
|   U114|2016-06-06|
|       |      null|
+-------+----------+
```

- User_Artist_Map

```
scala> sqlContext.sql("CREATE TABLE IF NOT EXISTS User_Artist_Map(user_id string, artist_id array<string>) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' COLLECTION ITEM
S TERMINATED BY '&' ")
res45: org.apache.spark.sql.DataFrame = [result: string]

scala> sqlContext.sql("LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/ProjectData/user-artist.txt' INTO table User_Artist_Map").show
+------+
|result|
+------+
+------+
```

OutPut

```
scala> sqlContext.sql("select * from User_Artist_Map").show
+-------+------------------+
|user_id|         artist_id|
+-------+------------------+
|   null|[A300, A301, A302]|
|   null|      [A301, A302]|
|   null|            [A302]|
|   null|[A303, A301, A302]|
|   null|      [A304, A301]|
|   null|[A305, A301, A302]|
|   null|      [A301, A302]|
|   null|            [A302]|
|   null|[A300, A303, A304]|
|   null|      [A301, A303]|
|   null|      [A302, A301]|
|   null|      [A303, A301]|
|   null|      [A304, A301]|
```

MOBILE DATA

COVERSION FRO CHARARRAY TODATE

- File.txt Input

```
U114,S207,A303,1465130523,1465230523,1475130523,A,ST415,3,1,0
U107,S202,A303,1495130523,1465230523,1465230523,U,ST415,0,1,1
U100,S204,A302,1495130523,1475130523,1465130523,AU,ST408,2,1,1
U104,S202,A303,1465230523,1475130523,1465130523,A,ST409,2,0,1
U102,S207,A301,1465230523,1485130523,1465230523,AU,ST403,3,1,1
,S203,A302,1495130523,1475130523,1465230523,E,ST400,0,0,1
U106,S202,A302,1465230523,1465130523,1465130523,AU,ST408,0,1,1
U105,S207,A300,1465230523,1485130523,1465130523,U,ST400,2,0,1
U108,S205,A304,1465130523,1465130523,1475130523,,ST410,2,1,0
U105,S203,,1475130523,1465230523,1465130523,AU,ST408,2,0,1
U110,S203,A300,1465230523,1465130523,1485130523,A,ST415,0,1,1
U113,S200,A303,1465230523,1475130523,1465130523,E,ST413,3,1,1
U119,S208,A302,1495130523,1465230523,1465230523,U,ST415,3,0,0
U118,S208,A303,1475130523,1465130523,1465230523,E,ST415,3,0,0
U107,S210,A302,1475130523,1485130523,1485130523,AP,ST404,2,1,0
U118,S202,A300,1495130523,1465230523,1465230523,AP,ST410,1,0,0
U111,S206,A305,1465130523,1465130523,1485130523,AU,ST415,0,1,1
U116,S208,A303,1465230523,1485130523,1475130523,A,ST413,1,0,1
U101,S202,A300,1465230523,1465130523,1475130523,U,ST401,0,0,1
U120,S206,A303,1495130523,1485130523,1465130523,AU,ST414,0,0,0
```

Commands Used to Converting chararray to TimeStamp

```
grunt> A = LOAD '/home/acadgild/Desktop/ProjectData/file.txt'  Using PigStorage(',') as (user_id:chararray,song_id:chararray,
artist_id:chararray,timestamp:chararray,start_ts:chararray,end_ts:chararray,song_end_type:int,like:int,dislike:int);
2018-04-10 11:58:43,446 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-04-10 11:58:43,490 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 3 ti
me(s).
grunt> B = FOREACH A GENERATE ToString(ToDate((long) timestamp*1000),'yyyy-MM-dd') as timestamp;
2018-04-10 11:58:49,080 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 4 ti
me(s).
grunt> dump;
```

```
grunt> A = LOAD '/home/acadgild/Desktop/ProjectData/file.txt'  Using PigStorage(',') as (user_id:chararray,song_id:chararray,
artist_id:chararray,timestamp:chararray,start_ts:chararray,end_ts:chararray,song_end_type:int,like:int,dislike:int);
2018-04-10 11:57:53,821 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-04-10 11:57:53,841 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 2 ti
me(s).
grunt> B = FOREACH A GENERATE ToString(ToDate((long) start_ts*1000),'yyyy-MM-dd') as start;
2018-04-10 11:57:58,633 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_LONG 3 ti
me(s).
grunt> dump;
```

```
grunt> A = LOAD '/home/acadgild/Desktop/ProjectData/file.txt'  Using PigStorage(
',') as (user_id:chararray,song_id:chararray,artist_id:chararray,timestamp:chara
rray,start_ts:chararray,end_ts:chararray,song_end_type:int,like:int,dislike:int)
;
2018-04-10 11:46:42,715 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2018-04-10 11:46:42,724 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - E
ncountered Warning IMPLICIT_CAST_TO_LONG 1 time(s).
grunt> B = FOREACH A GENERATE ToString(ToDate((long) end_ts*1000),'yyyy-MM-dd')
as end;
2018-04-10 11:47:03,549 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - E
ncountered Warning IMPLICIT_CAST_TO_LONG 2 time(s).
grunt> dump;
```

OutPut

```
U114,S207,A303,2016-06-05,2016-06-06,2016-09-29,A,ST415,3,1,0
U107,S202,A303,2017-05-18,2016-06-06,2016-06-06,U,ST415,0,1,1
U100,S204,A302,2017-05-18,2016-09-29,2016-06-05,AU,ST408,2,1,1
U104,S202,A303,2016-06-06,2016-09-29,2016-06-05,A,ST409,2,0,1
U102,S207,A301,2016-06-06,2017-01-23,2016-06-06,AU,ST403,3,1,1
,S203,A302,2017-05-18,2016-09-29,2016-06-06,E,ST400,0,0,1
U106,S202,A302,2016-06-06,2016-06-05,2016-06-05,AU,ST408,0,1,1
U105,S207,A300,2016-06-06,2017-01-23,2016-06-05,U,ST400,2,0,1
U108,S205,A304,2016-06-05,2016-06-05,2016-09-29,,ST410,2,1,0
U105,S203,,2016-09-29,2016-06-06,2016-06-05,AU,ST408,2,0,1
U110,S203,A300,2016-06-06,2016-06-05,2017-01-23,A,ST415,0,1,1
U113,S200,A303,2016-06-06,2016-09-29,2016-06-05,E,ST413,3,1,1
U119,S208,A302,2017-05-18,2016-06-06,2016-06-06,U,ST415,3,0,0
U118,S208,A303,2016-09-29,2016-06-05,2016-06-06,E,ST415,3,0,0
U107,S210,A302,2016-09-29,2017-01-23,2017-01-23,AP,ST404,2,1,0
U118,S202,A300,2017-05-18,2016-06-06,2016-06-06,AP,ST410,1,0,0
U111,S206,A305,2016-06-05,2016-06-05,2017-01-23,AU,ST415,0,1,1
U116,S208,A303,2016-06-06,2017-01-23,2016-09-29,A,ST413,1,0,1
U101,S202,A300,2016-06-06,2016-06-05,2016-09-29,U,ST401,0,0,1
U120,S206,A303,2017-05-18,2017-01-23,2016-06-05,AU,ST414,0,0,0
```

WEBDATA

1. Data input


```
<records>
  <record>
    <user_id>U106</user_id>
    <song_id>S205</song_id>
    <artist_id>A300</artist_id>
    <timestamp>2016-05-10 12:24:22</timestamp>
    <start_ts>2016-05-10 12:24:22</start_ts>
    <end_ts>2017-05-09 08:09:22</end_ts>
    <geo_cd>AP</geo_cd>
    <station_id>ST407</station_id>
    <song_end_type>2</song_end_type>
    <like>1</like>
    <dislike>1</dislike>
  </record>
  <record>
    <user_id>U114</user_id>
    <song_id>S209</song_id>
    <artist_id>A303</artist_id>
    <timestamp>2016-06-09 22:12:36</timestamp>
    <start_ts>2016-05-10 12:24:22</start_ts>
    <end_ts>2017-05-09 08:09:22</end_ts>
    <geo_cd>U</geo_cd>
    <station_id>ST411</station_id>
    <song_end_type>2</song_end_type>
    <like>1</like>
    <dislike>0</dislike>
  </record>
```

Commands,

```
grunt> register '/home/acadgild/pig_xml.jar'
grunt> A= load '/home/acadgild/Desktop/ProjectData/file.xml' using org.apache.pig.piggybank.storage.XMLLoader('record') as (x
:chararray);
2018-05-05 16:14:09,139 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated.
Instead, use dfs.bytes-per-checksum
2018-05-05 16:14:09,144 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning NO_LOAD_FUNCTION_FOR_CASTI
NG_BYTEARRAY 29 time(s).
grunt> B = foreach A GENERATE FLATTEN(REGEX_EXTRACT_ALL(x,'<record>\\s*<user_id>(.*)</user_id>\\s*<song_id>(.*)</song_id>\\s*
<artist_id>(.*)</artist_id>\\s*<timestamp>(.*)</timestamp>\\s*<start_ts>(.*)</start_ts>\\s*<end_ts>(.*)</end_ts>\\s*<geo_cd>(
.*)</geo_cd>\\s*<station_id>(.*)</station_id>\\s*<song_end_type>(.*)</song_end_type>\\s*<like>(.*)</like>\\s*<dislike>(.*)</d
islike>\\s*</record>')) as (user_id:chararray,song_id:chararray,artist_id:chararray,timestamp:chararray,start_ts:chararray,en
d_ts:chararray,station_id:chararray,geo_cd:chararray,song_end_type:int,like:int,dislike:int);
2018-05-05 16:14:11,852 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning NO_LOAD_FUNCTION_FOR_CASTI
NG_BYTEARRAY 40 time(s).
grunt>
```

OUTPUT

```
(U106,S205,A300,2016-05-10 12:24:22,2016-05-10 12:24:22,2017-05-09 08:09:22,AP,ST407,2,1,1)
(U114,S209,A303,2016-06-09 22:12:36,2016-05-10 12:24:22,2017-05-09 08:09:22,U,ST411,2,1,0)
(U113,S203,A304,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-05-10 12:24:22,U,ST405,0,0,1)
(U108,S200,A302,2016-07-10 01:38:09,2016-05-10 12:24:22,2016-07-10 01:38:09,U,ST414,0,0,1)
(U102,S203,A305,2016-06-09 22:12:36,2016-06-09 22:12:36,2017-05-09 08:09:22,U,ST404,2,0,0)
(,S208,A300,2016-06-09 22:12:36,2017-05-09 08:09:22,2016-06-09 22:12:36,U,ST411,1,0,1)
(U115,S200,A300,2016-06-09 22:12:36,2017-05-09 08:09:22,2016-06-09 22:12:36,AU,ST404,3,0,0)
(U111,S204,A300,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-07-10 01:38:09,U,ST410,3,1,1)
(U120,S201,A300,2017-05-09 08:09:22,2016-06-09 22:12:36,2016-07-10 01:38:09,,ST410,3,0,1)
(U113,S203,,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-06-09 22:12:36,A,ST402,1,1,0)
(U109,S203,A304,2016-05-10 12:24:22,2017-05-09 08:09:22,2016-07-10 01:38:09,E,ST405,1,1,1)
(U110,S202,A303,2017-05-09 08:09:22,2017-05-09 08:09:22,2016-07-10 01:38:09,AU,ST402,2,1,0)
(U100,S200,A301,2017-05-09 08:09:22,2017-05-09 08:09:22,2017-05-09 08:09:22,AP,ST410,3,1,1)
(U101,S208,A300,2016-05-10 12:24:22,2016-07-10 01:38:09,2016-05-10 12:24:22,E,ST408,0,1,1)
(U106,S206,A300,2017-05-09 08:09:22,2016-06-09 22:12:36,2016-05-10 12:24:22,A,ST405,3,1,0)
(U107,S202,A304,2017-05-09 08:09:22,2016-07-10 01:38:09,2016-05-10 12:24:22,U,ST409,0,0,0)
(U103,S204,A300,2016-07-10 01:38:09,2017-05-09 08:09:22,2016-06-09 22:12:36,AU,ST411,2,1,0)
(U103,S202,A300,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-06-09 22:12:36,A,ST415,2,1,1)
(U113,S203,A303,2016-05-10 12:24:22,2016-07-10 01:38:09,2017-05-09 08:09:22,U,ST408,2,0,0)
(U113,S204,A301,2017-05-09 08:09:22,2017-05-09 08:09:22,2016-06-09 22:12:36,E,ST415,3,0,1)
```

****

FINALOUTPUT

```
U106,S205,A300,2016-05-10 12:24:22,2016-05-10 12:24:22,2017-05-09 08:09:22,AP,ST407,2,1,1
U114,S209,A303,2016-06-09 22:12:36,2016-05-10 12:24:22,2017-05-09 08:09:22,U,ST411,2,1,0
U113,S203,A304,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-05-10 12:24:22,U,ST405,0,0,1
U108,S200,A302,2016-07-10 01:38:09,2016-05-10 12:24:22,2016-07-10 01:38:09,U,ST414,0,0,1
U102,S203,A305,2016-06-09 22:12:36,2016-06-09 22:12:36,2017-05-09 08:09:22,U,ST404,2,0,0
,S208,A300,2016-06-09 22:12:36,2017-05-09 08:09:22,2016-06-09 22:12:36,U,ST411,1,0,1
U115,S200,A300,2016-06-09 22:12:36,2017-05-09 08:09:22,2016-06-09 22:12:36,AU,ST404,3,0,0
U111,S204,A300,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-07-10 01:38:09,U,ST410,3,1,1
U120,S201,A300,2017-05-09 08:09:22,2016-06-09 22:12:36,2016-07-10 01:38:09,,ST410,3,0,1
U113,S203,,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-06-09 22:12:36,A,ST402,1,1,0
U109,S203,A304,2016-05-10 12:24:22,2017-05-09 08:09:22,2016-07-10 01:38:09,E,ST405,1,1,1
U110,S202,A303,2017-05-09 08:09:22,2017-05-09 08:09:22,2016-07-10 01:38:09,AU,ST402,2,1,0
U100,S200,A301,2017-05-09 08:09:22,2017-05-09 08:09:22,2017-05-09 08:09:22,AP,ST410,3,1,1
U101,S208,A300,2016-05-10 12:24:22,2016-07-10 01:38:09,2016-05-10 12:24:22,E,ST408,0,1,1
U106,S206,A300,2017-05-09 08:09:22,2016-06-09 22:12:36,2016-05-10 12:24:22,A,ST405,3,1,0
U107,S202,A304,2017-05-09 08:09:22,2016-07-10 01:38:09,2016-05-10 12:24:22,U,ST409,0,0,0
U103,S204,A300,2016-07-10 01:38:09,2017-05-09 08:09:22,2016-06-09 22:12:36,AU,ST411,2,1,0
U103,S202,A300,2016-06-09 22:12:36,2016-06-09 22:12:36,2016-06-09 22:12:36,A,ST415,2,1,1
U113,S203,A303,2016-05-10 12:24:22,2016-07-10 01:38:09,2017-05-09 08:09:22,U,ST408,2,0,0
U113,S204,A301,2017-05-09 08:09:22,2017-05-09 08:09:22,2016-06-09 22:12:36,E,ST415,3,0,1
U114,S207,A303,2016-06-05,2016-06-06,2016-09-29,A,ST415,3,1,0

U107,S202,A303,2017-05-18,2016-06-06,2016-06-06,U,ST415,0,1,1
```

TOTAL DATA READ

Commands

```
scala> val dataFrame = sqlContext.read.format("file:///home/acadgild/Desktop/Pro
jectData/FinalOutput")
dataFrame: org.apache.spark.sql.DataFrameReader = org.apache.spark.sql.DataFrame
Reader@4b808427

scala> val DF = dataFrame.csv("file:///home/acadgild/Desktop/ProjectData/FinalOu
tput")
DF: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 9 more fields
]

scala> val DataFrame1 = DF.toDF()
DataFrame1: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 9 mor
e fields]

scala> DataFrame1.createOrReplaceTempView("FinalData")
```

Output

```
scala> sqlContext.sql("select * from FinalData").show
+----+----+----+-------------------+-------------------+-------------------+----
+-----+---+---+----+
| _c0| _c1| _c2|                _c3|                _c4|                _c5| _c6
|  _c7|_c8|_c9|_c10|
+----+----+----+-------------------+-------------------+-------------------+----
+-----+---+---+----+
|U106|S205|A300|2016-05-10 12:24:22|2016-05-10 12:24:22|2017-05-09 08:09:22|  AP
|ST407|  2|  1|   1|
|U114|S209|A303|2016-06-09 22:12:36|2016-05-10 12:24:22|2017-05-09 08:09:22|   U
|ST411|  2|  1|   0|
|U113|S203|A304|2016-06-09 22:12:36|2016-06-09 22:12:36|2016-05-10 12:24:22|   U
|ST405|  0|  0|   1|
|U108|S200|A302|2016-07-10 01:38:09|2016-05-10 12:24:22|2016-07-10 01:38:09|   U
|ST414|  0|  0|   1|
|U102|S203|A305|2016-06-09 22:12:36|2016-06-09 22:12:36|2017-05-09 08:09:22|   U
|ST404|  2|  0|   0|
|null|S208|A300|2016-06-09 22:12:36|2017-05-09 08:09:22|2016-06-09 22:12:36|   U
|ST411|  1|  0|   1|
|U115|S200|A300|2016-06-09 22:12:36|2017-05-09 08:09:22|2016-06-09 22:12:36|  AU
|ST404|  3|  0|   0|
|U111|S204|A300|2016-06-09 22:12:36|2016-06-09 22:12:36|2016-07-10 01:38:09|   U
|ST410|  3|  1|   1|
|U120|S201|A300|2017-05-09 08:09:22|2016-06-09 22:12:36|2016-07-10 01:38:09|null
|ST410|  3|  0|   1|
|U113|S203|null|2016-06-09 22:12:36|2016-06-09 22:12:36|2016-06-09 22:12:36|   A
|ST402|  1|  1|   0|
|U109|S203|A304|2016-05-10 12:24:22|2017-05-09 08:09:22|2016-07-10 01:38:09|   E
|ST405|  1|  1|   1|
|U110|S202|A303|2017-05-09 08:09:22|2017-05-09 08:09:22|2016-07-10 01:38:09|  AU
|ST402|  2|  1|   0|
|U100|S200|A301|2017-05-09 08:09:22|2017-05-09 08:09:22|2017-05-09 08:09:22|  AP
|ST410|  3|  1|   1|
|U101|S208|A300|2016-05-10 12:24:22|2016-07-10 01:38:09|2016-05-10 12:24:22|   E
|ST408|  0|  1|   1|
|U106|S206|A300|2017-05-09 08:09:22|2016-06-09 22:12:36|2016-05-10 12:24:22|   A
|ST405|  3|  1|   0|
|U107|S202|A304|2017-05-09 08:09:22|2016-07-10 01:38:09|2016-05-10 12:24:22|   U
|ST409|  0|  0|   0|
```

REMOVING NULL VALUES

```
scala> sqlContext.sql("select _c0,_c1,_c2,_c3,_c4,_c5,_c6,_c7,_c8,_c9,_c10 from FinalData where _c8 != 'null' and _c2 != 'nul
l' and _c6 != 'null' GROUP BY _c0,_c1,_c2,_c3,_c4,_c5,_c6,_c7,_c8,_c9,_c10 ").show
+----+----+----+-------------------+-------------------+-------------------+-------------------+---+-----+---+---+----+
| _c0| _c1| _c2|                _c3|                _c4|                _c5|_c6|  _c7|_c8|_c9|_c10|
+----+----+----+-------------------+-------------------+-------------------+---+-----+---+---+----+
|U118|S202|A390|         2017-05-18|         2016-06-06|         2016-06-06| AP|ST410|  1|  0|   0|
|U106|S206|A306|2017-05-09 08:09:22|2016-06-09 22:12:36|2016-05-10 12:24:22|  A|ST405|  3|  1|   0|
|U105|S207|A300|         2016-06-06|         2017-01-23|         2016-06-05|  U|ST400|  2|  0|   1|
|U100|S204|A302|         2017-05-18|         2016-09-29|         2016-06-05| AU|ST408|  2|  1|   1|
|U106|S202|A302|         2016-06-06|         2016-06-05|         2016-06-05| AU|ST408|  8|  1|   1|
|U111|S204|A300|2016-06-09 22:12:36|2016-06-09 22:12:36|2016-07-10 01:38:09|  U|ST410|  3|  1|   1|
|U102|S203|A305|2016-06-09 22:12:36|2016-06-09 22:12:36|2017-05-09 08:09:22|  U|ST404|  2|  0|   0|
|U113|S200|A303|         2016-06-06|         2016-09-29|         2016-06-05|  E|ST413|  3|  1|   1|
|U119|S208|A302|         2017-05-18|         2016-06-06|         2016-06-06|  U|ST415|  3|  0|   0|
|U110|S203|A300|         2016-06-06|         2016-06-05|         2017-01-23|  A|ST415|  0|  1|   1|
|U101|S202|A300|         2016-06-06|         2016-06-05|         2016-09-29|  U|ST401|  6|  0|   1|
|U107|S202|A304|2017-05-09 08:09:22|2016-07-10 01:38:09|2016-05-10 12:24:22|  U|ST409|  9|  0|   0|
|U107|S210|A302|         2016-09-29|         2017-01-23|         2017-01-23| AP|ST404|  2|  1|   0|
|U110|S202|A303|2017-05-09 08:09:22|2017-05-09 08:09:22|2016-07-10 01:38:09| AU|ST402|  2|  1|   0|
|U109|S203|A304|2016-05-10 12:24:22|2017-05-09 08:09:22|2016-07-10 01:38:09|  E|ST405|  1|  1|   1|
|U107|S202|A303|         2017-05-18|         2016-06-06|         2016-06-06|  U|ST415|  0|  1|   1|
|U103|S204|A306|2016-07-10 01:38:09|2017-05-09 08:09:22|2016-06-09 22:12:36| AU|ST411|  2|  1|   0|
|U104|S202|A303|         2016-06-06|         2016-09-29|         2016-06-05|  A|ST409|  2|  0|   1|
|U100|S208|A301|2017-05-09 08:09:22|2017-05-09 08:09:22|2017-05-09 08:09:22| AP|ST410|  3|  1|   1|
|U115|S208|A306|2016-06-09 22:12:36|2017-05-09 08:09:22|2016-06-09 22:12:36| AU|ST404|  3|  0|   0|
+----+----+----+-------------------+-------------------+-------------------+---+-----+---+---+----+
only showing top 20 rows
```

---

1. DETERMINE TOP 10 Station_Id where max songs are played

```
scala> sqlContext.sql("select COUNT(distinct _c1),COUNT(distinct _c0), _c0, _c1, _c7 from FinalData where _c9 > '0' GROUP BY _c0
,_c1,_c7 ORDER BY _c7 Limit 10").show
+------------------+------------------+----+----+-----+
|count(DISTINCT _c1)|count(DISTINCT _c0)| _c0| _c1|  _c7|
+------------------+------------------+----+----+-----+
|                 1|                 1|U113|S203|ST402|
|                 1|                 1|U110|S202|ST402|
|                 1|                 1|U102|S207|ST403|
|                 1|                 1|U107|S210|ST404|
|                 1|                 1|U106|S206|ST405|
|                 1|                 1|U109|S203|ST405|
|                 1|                 1|U106|S205|ST407|
|                 1|                 1|U101|S208|ST408|
|                 1|                 1|U106|S202|ST408|
|                 1|                 1|U100|S204|ST408|
+------------------+------------------+----+----+-----+
```

---

2. Total Duration of Songs played by each user Type

1. Reading the Final output MusicData File,

```
scala> val MusicString = "user_id:String,Artist_id:String,Song_Id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong,st
ation_id:String,Geo_cd:String,song_end_type:toInt,like:toInt,dislike:toInt"
MusicString: String = user_id:String,Artist_id:String,Song_Id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong,statio
n_id:String,Geo_cd:String,song_end_type:toInt,like:toInt,dislike:toInt

scala> val MusicSchema = StructType(MusicString.split(",").map(fieldInfo=>StructField(fieldInfo.split(":")(0),if(fieldInfo.sp
lit(":")(1).equals("string")) StringType else IntegerType,true))
```

---

2. Reading the User-subscn.txt File

```
scala> val SubscriberSchema = StructType(MusicString.split(",").map(fieldInfo=>StructField(fieldInfo.split(":")(0),if(fieldIn
fo.split(":")(1).equals("string")) StringType else IntegerType,true))
SubscriberSchema: org.apache.spark.sql.types.StructType = StructType(StructField(user_id,IntegerType,true), StructField(TimeS
tamp_ts,IntegerType,true), StructField(start_ts,IntegerType,true), StructField(end_ts,IntegerType,true))

scala> val SubscriberString = "user_id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong"
SubscriberString: String = user_id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong

scala> val SubscriberData = sc.textFile("/home/acadgild/Desktop/ProjectData/UserSubscription").map(x=>(x(0).toString,x(1).toL
ong,x(2).toLong)).toDF
SubscriberData: org.apache.spark.sql.DataFrame = [_1: string, _2: bigint ... 1 more field]

scala> MusicData.registerTempTable("SubscriberOutput")
warning: there was one deprecation warning; re-run with -deprecation for details
```

Command

```
scala> val User_Behaviour = sqlContext.sql(s"SELECT CASE WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) > CAST(Subscri
berOutput.subscription_end AS DECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) <= CA
ST(SubscriberOutput.subscription_end AS DECIMAL(20,0))) THEN 'SUBSCRIBED' END AS UserType,LEFT OUTER JOIN SubscribedOutput su
busers ON MusicOutput.user_id = SubscribedOutput.user_id GROUP BY CASE WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0))
> CAST(SubscriberOutput.subscription_end AS DECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (CAST(MusicOutput.timestamp_ts AS DECIMA
L(20,0)) <= CAST(SubscriberOutput.subscription_end AS DECIMAL(20,0))) THEN 'SUBSCRIBED' ")
org.apache.spark.sql.catalyst.parser.ParseException:
extraneous input 'JOIN' expecting {<EOF>, ',', 'FROM', 'WHERE', 'GROUP', 'ORDER', 'HAVING', 'LIMIT', 'LATERAL', 'WINDOW', 'UN
ION', 'EXCEPT', 'MINUS', 'INTERSECT', 'SORT', 'CLUSTER', 'DISTRIBUTE'}(line 1, pos 300)

== SQL ==
SELECT CASE WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) > CAST(SubscriberOutput.subscription_end AS DECIMAL(20,0)))
 THEN 'UNSUBSCRIBED' WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) <= CAST(SubscriberOutput.subscription_end AS DECIM
AL(20,0))) THEN 'SUBSCRIBED' END AS UserType,LEFT OUTER JOIN SubscribedOutput subusers ON MusicOutput.user_id = SubscribedOut
put.user_id GROUP BY CASE WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) > CAST(SubscriberOutput.subscription_end AS D
ECIMAL(20,0))) THEN 'UNSUBSCRIBED' WHEN (CAST(MusicOutput.timestamp_ts AS DECIMAL(20,0)) <= CAST(SubscriberOutput.subscriptio
n_end AS DECIMAL(20,0))) THEN 'SUBSCRIBED'
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------^^^

  at org.apache.spark.sql.catalyst.parser.ParseException.withCommand(ParseDriver.scala:217)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parse(ParseDriver.scala:114)
  at org.apache.spark.sql.execution.SparkSqlParser.parse(SparkSqlParser.scala:48)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parsePlan(ParseDriver.scala:68)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:632)
  at org.apache.spark.sql.SQLContext.sql(SQLContext.scala:691)
  ... 50 elided

scala>
```

1

****

3. Top 10 connected Artist

Reading the Music DataFile

```
scala> val MusicString = "user_id:String,Artist_id:String,Song_Id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong,st
ation_id:String,Geo_cd:String,song_end_type:toInt,like:toInt,disLike:toInt"
MusicString: String = user_id:String,Artist_id:String,Song_Id:String,TimeStamp_ts:toLong,start_ts:toLong,end_ts:toLong,statio
n_id:String,Geo_cd:String,song_end_type:toInt,like:toInt,disLike:toInt

scala> val MusicSchema = StructType(MusicString.split(",").map(fieldInfo=>StructField(fieldInfo.split(":")(0),if(fieldInfo.sp
lit(":")(1).equals("string")) StringType else IntegerType,true)))
MusicSchema: org.apache.spark.sql.types.StructType = StructType(StructField(user_id,IntegerType,true), StructField(Artist_id,
IntegerType,true), StructField(Song_Id,IntegerType,true), StructField(TimeStamp_ts,IntegerType,true), StructField(start_ts,In
tegerType,true), StructField(end_ts,IntegerType,true), StructField(station_id,IntegerType,true), StructField(Geo_cd,IntegerTy
pe,true), StructField(song_end_type,IntegerType,true), StructField(like,IntegerType,true), StructField(disLike,IntegerType,tr
ue))

scala> val MusicData = sc.textFile("/home/acadgild/Desktop/ProjectData/Music").map(x=>(x(0).toString,x(1).toString,x(2).toStr
ing,x(3).toLong,x(4).toLong,x(5).toLong,x(6).toString,x(7).toString,x(8).toInt,x(9).toInt,x(10).toInt)).toDF()
MusicData: org.apache.spark.sql.DataFrame = [_1: string, _2: string ... 9 more fields]

scala> MusicData.registerTempTable("MusicOutput")
warning: there was one deprecation warning; re-run with -deprecation for details
```

Readng the User-artist File

```
scala> val ArtistString = "user_id:String,Artist_id:array<String>"
ArtistString: String = user_id:String,Artist_id:array<String>

scala> val ArtistSchema = StructType(MusicString.split(",").map(fieldInfo=>StructField(fieldInfo.split(":")(0),if(fieldInfo.s
plit(":")(1).equals("string")) StringType else IntegerType,true)))
ArtistSchema: org.apache.spark.sql.types.StructType = StructType(StructField(user_id,IntegerType,true), StructField(Artist_id
,IntegerType,true))

scala> val ArtistData = sc.textFile("/home/acadgild/Desktop/ProjectData/user-artist.txt").map(x=>(x(0).toString,x(1).toString
)).toDF()
ArtistData: org.apache.spark.sql.DataFrame = [_1: string, _2: string]


       ArtistData.registerTempTable("ArtistOutput")
warning: there was one deprecation warning; re-run with -deprecation for details
```

Command

```
scala> val data = sqlContext.sql(s"select ua.artist,COUNT(DISTINCT ua.user_id) as User_Count ua INNER JOIN (SELECT artist_id,
song_id,User_id from MusicOutput)ed ON ua.artist_id = ed.artist_id GROUP BY ua_artist_id")
org.apache.spark.sql.catalyst.parser.ParseException:
extraneous input 'ua' expecting {<EOF>, ',', 'FROM', 'WHERE', 'GROUP', 'ORDER', 'HAVING', 'LIMIT', 'LATERAL', 'WINDOW', 'UNIO
N', 'EXCEPT', 'MINUS', 'INTERSECT', 'SORT', 'CLUSTER', 'DISTRIBUTE'}(line 1, pos 58)

== SQL ==
select ua.artist,COUNT(DISTINCT ua.user_id) as User_Count ua INNER JOIN (SELECT artist_id,song_id,User_id from MusicOutput)ed
 ON ua.artist_id = ed.artist_id GROUP BY ua_artist_id
----------------------------------------------------------^^^

  at org.apache.spark.sql.catalyst.parser.ParseException.withCommand(ParseDriver.scala:217)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parse(ParseDriver.scala:114)
  at org.apache.spark.sql.execution.SparkSqlParser.parse(SparkSqlParser.scala:48)
  at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parsePlan(ParseDriver.scala:68)
  at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:632)
  at org.apache.spark.sql.SQLContext.sql(SQLContext.scala:691)
  ... 50 elided

scala>
```

****

4. Determine top 10 Songs who have generated maximum revenue

```
scala> sqlContext.sql("select song_id,SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL(20,0))))AS Duration From
 FinalMusic where (like = 1 and song_end_type = 0)GROUP BY song_id")
res10: org.apache.spark.sql.DataFrame = [song_id: string, Duration: decimal(31,0)]
```

OUTPUT

```
scala> sqlContext.sql("select song_id,SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL(20,0))))AS Duration From
 FinalMusic where (like = 1 and song_end_type = 0)GROUP BY song_id ORDER BY Duration").show
+-------+--------+
|song_id|Duration|
+-------+--------+
|   5202|    null|
|   5205|    null|
|   5200|    null|
|   5203|    null|
+-------+--------+
```

****

5. Top 10 unsubscribed users who listened to songs for maximum duration

```
scala> sqlContext.sql("select ed.user_id,SUM(ABS(CAST(end_ts AS DECIMAL(20,0))-CAST(start_ts AS DECIMAL(20,0))))AS Duration F
rom FinalMusic ed LEFT OUTER JOIN Subscript_end su ON ed.user_id = su.user_id WHERE su.user_id is NULL OR (CAST(ed.timestamp_
ts AS DECIMAL(20,0)) > CAST(su.subscription_end AS DECIMAL(20,0))) ORDER BY Duration")
org.apache.spark.sql.AnalysisException: grouping expressions sequence is empty, and 'ed.`user_id`' is not an aggregate functi
on. Wrap '(sum(abs((CAST(CAST(ed.`end_ts` AS DECIMAL(20,0)) AS DECIMAL(21,0)) - CAST(CAST(ed.`start_ts` AS DECIMAL(20,0)) AS
DECIMAL(21,0))))) AS `Duration`)' in windowing function(s) or wrap 'ed.`user_id`' in first() (or first_value) if you don't ca
re which value you get.;;
Sort [Duration#137 ASC NULLS FIRST], true
+- Aggregate [user_id#140, sum(abs(CheckOverflow((promote_precision(cast(cast(end_ts#143 as decimal(20,0)) as decimal(21,0)))
 - promote_precision(cast(cast(start_ts#142 as decimal(20,0)) as decimal(21,0)))), DecimalType(21,0),true)) AS Duration#137]
   +- Filter (isnull(user_id#149) || (cast(timestamp_ts#141 as decimal(20,0)) > cast(subscription_end#150 as decimal(20,0))))
      +- Join LeftOuter, (user_id#140 = user_id#149)
         :- SubqueryAlias ed
         :  +- SubqueryAlias finalmusic
         :     +- HiveTableRelation `default`.`finalmusic`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [Artist_id#13
8, song_id#139, User_id#140, timestamp_ts#141, start_ts#142, end_ts#143, Station_id#144, Geo_cd#145, song_end_type#146, like#
147, dislike#148]
         +- SubqueryAlias su
            +- SubqueryAlias subscript_end
               +- HiveTableRelation `default`.`subscript_end`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, [User_id#1
49, subscription_end#150]

  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$class.failAnalysis(CheckAnalysis.scala:39)
  at org.apache.spark.sql.catalyst.analysis.Analyzer.failAnalysis(Analyzer.scala:91)
  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1.org$apache$spark$sql$catalyst$analysis$Che
ckAnalysis$class$$anonfun$$checkValidAggregateExpression$1(CheckAnalysis.scala:239)
  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$9.apply(CheckAnalysis.scala
:288)
  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$9.apply(CheckAnalysis.scala
:288)
  at scala.collection.immutable.List.foreach(List.scala:381)
  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1.apply(CheckAnalysis.scala:288)
  at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1.apply(CheckAnalysis.scala:78)
  at org.apache.spark.sql.catalyst.trees.TreeNode.foreachUp(TreeNode.scala:127)
  at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$foreachUp$1.apply(TreeNode.scala:126)
  at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$foreachUp$1.apply(TreeNode.scala:126)
  at scala.collection.immutable.List.foreach(List.scala:381)
```

****