# DataLake_project

March 23, 2021

In [28]:
```
#%run -i 'etl.py'
```

In [13]:
```
import configparser
from datetime import datetime
import os
from pyspark.sql import SparkSession
from pyspark.sql.functions import udf, col
from pyspark.sql.functions import year, month, dayofmonth, hour, weekofyear, date_forma


config = configparser.ConfigParser()
config.read('dl.cfg')

os.environ['AWS_ACCESS_KEY_ID']=config['AWS']['AWS_ACCESS_KEY_ID']
os.environ['AWS_SECRET_ACCESS_KEY']=config['AWS']['AWS_SECRET_ACCESS_KEY']


spark = SparkSession \
        .builder \
        .config("spark.jars.packages", "org.apache.hadoop:hadoop-aws:2.7.0") \
        .getOrCreate()
```

In [14]:
```
input_data = "s3a://udacity-dend/"
song_data = input_data + "/*/*/*/*.json"
```

In [15]:
```
df = spark.read.json(song_data)
```

```
---------------------------------------------------------------------------

Py4JJavaError                             Traceback (most recent call last)

<ipython-input-15-58ad0d0bf08d> in <module>()
----> 1 df = spark.read.json(song_data)


/opt/spark-2.4.3-bin-hadoop2.7/python/pyspark/sql/readwriter.py in json(self, path, sche
272             path = [path]
```

```
    273         if type(path) == list:
--> 274             return self._df(self._jreader.json(self._spark._sc._jvm.PythonUtils.toSe
    275         elif isinstance(path, RDD):
    276             def func(iterator):


    /opt/spark-2.4.3-bin-hadoop2.7/python/lib/py4j-0.10.7-src.zip/py4j/java_gateway.py in __
   1255         answer = self.gateway_client.send_command(command)
   1256         return_value = get_return_value(
-> 1257             answer, self.gateway_client, self.target_id, self.name)
   1258
   1259         for temp_arg in temp_args:


    /opt/spark-2.4.3-bin-hadoop2.7/python/pyspark/sql/utils.py in deco(*a, **kw)
     61     def deco(*a, **kw):
     62         try:
---> 63             return f(*a, **kw)
     64         except py4j.protocol.Py4JJavaError as e:
     65             s = e.java_exception.toString()


    /opt/spark-2.4.3-bin-hadoop2.7/python/lib/py4j-0.10.7-src.zip/py4j/protocol.py in get_re
    326                 raise Py4JJavaError(
    327                     "An error occurred while calling {0}{1}{2}.\n".
--> 328                     format(target_id, ".", name), value)
    329             else:
    330                 raise Py4JError(


    Py4JJavaError: An error occurred while calling o208.json.
: com.amazonaws.services.s3.model.AmazonS3Exception: Status Code: 403, AWS Service: Amazon S
        at com.amazonaws.http.AmazonHttpClient.handleErrorResponse(AmazonHttpClient.java:798
        at com.amazonaws.http.AmazonHttpClient.executeHelper(AmazonHttpClient.java:421)
        at com.amazonaws.http.AmazonHttpClient.execute(AmazonHttpClient.java:232)
        at com.amazonaws.services.s3.AmazonS3Client.invoke(AmazonS3Client.java:3528)
        at com.amazonaws.services.s3.AmazonS3Client.invoke(AmazonS3Client.java:3480)
        at com.amazonaws.services.s3.AmazonS3Client.listObjects(AmazonS3Client.java:604)
        at org.apache.hadoop.fs.s3a.S3AFileSystem.getFileStatus(S3AFileSystem.java:962)
        at org.apache.hadoop.fs.s3a.S3AFileSystem.listStatus(S3AFileSystem.java:734)
        at org.apache.hadoop.fs.Globber.listStatus(Globber.java:69)
        at org.apache.hadoop.fs.Globber.glob(Globber.java:217)
        at org.apache.hadoop.fs.FileSystem.globStatus(FileSystem.java:1657)
        at org.apache.spark.deploy.SparkHadoopUtil.globPath(SparkHadoopUtil.scala:245)
        at org.apache.spark.deploy.SparkHadoopUtil.globPathIfNecessary(SparkHadoopUtil.scala
        at org.apache.spark.sql.execution.datasources.DataSource$$anonfun$org$apache$spark$s
        at org.apache.spark.sql.execution.datasources.DataSource$$anonfun$org$apache$spark$s
        at scala.collection.TraversableLike$$anonfun$flatMap$1.apply(TraversableLike.scala:2
```

```
        at scala.collection.TraversableLike$$anonfun$flatMap$1.apply(TraversableLike.scala:2
        at scala.collection.immutable.List.foreach(List.scala:392)
        at scala.collection.TraversableLike$class.flatMap(TraversableLike.scala:241)
        at scala.collection.immutable.List.flatMap(List.scala:355)
        at org.apache.spark.sql.execution.datasources.DataSource.org$apache$spark$sql$execut
        at org.apache.spark.sql.execution.datasources.DataSource.resolveRelation(DataSource.
        at org.apache.spark.sql.DataFrameReader.loadV1Source(DataFrameReader.scala:223)
        at org.apache.spark.sql.DataFrameReader.load(DataFrameReader.scala:211)
        at org.apache.spark.sql.DataFrameReader.json(DataFrameReader.scala:391)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:748)
```

In [ ]:

In [ ]:
```
df.printSchema()
df.show(5)
```

In [ ]: