Hadoop commands

```
1-Go to CMD
2->Start-all
3- Get the files from data folder
  >hadoop fs -mkdir /finance/
  >hadoop fs -mkdir "/finance/stock=AAPL"
  >hadoop fs -mkdir "/finance/stock=AIG"
  >hadoop fs -mkdir "/finance/stock=AMZN"
  >hadoop fs -mkdir "/finance/stock=BA"
  >hadoop fs -mkdir "/finance/stock=AXP"

4-Load the file from GitHub to Hadoop
```

In [ ]:
```
import findspark
import pyspark
import random
findspark.init()
```

In [5]:
```
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
spark = SparkSession.builder.appName('stocks').getOrCreate()
```

In [7]:
```
df = spark.read.csv('hdfs://localhost:9000/finance',inferSchema=True,header=True
df.select('stock').distinct().show()
#groupBy=df.select('country','city').groupBy('country').count()
```

```
+-----+
|stock|
+-----+
|  AXP|
| AAPL|
|  AIG|
|   BA|
| AMZN|
+-----+
```

In [8]:
```
df.printSchema()
```

```
root
 |-- date: timestamp (nullable = true)
 |-- open: double (nullable = true)
 |-- close: double (nullable = true)
 |-- stock: string (nullable = true)
```

In [9]:
```
df.select("date","stock").groupBy("stock").count().show()
```

```
+-----+-----+
|stock|count|
+-----+-----+
|  AXP| 1258|
| AAPL| 1258|
|  AIG| 1258|
|   BA| 1258|
| AMZN| 1258|
+-----+-----+
```

In [10]:
```
#df.withColumn("datetime", col("datetime").cast("timestamp")
#df.withColumn("date", toTimeStamp(df("date"))).groupBy("stock").max("date").shou
#df.select("date").show()
df2=df.withColumnRenamed('date',('dateR'))
df2.show()
```

```
+-------------------+-----+-----+-----+
|              dateR| open|close|stock|
+-------------------+-----+-----+-----+
|2003-01-02 00:00:00|14.36| 14.8| AAPL|
|2003-01-03 00:00:00| 14.8| 14.9| AAPL|
|2003-01-06 00:00:00|15.03| 14.9| AAPL|
|2003-01-07 00:00:00|14.79|14.85| AAPL|
|2003-01-08 00:00:00|14.58|14.55| AAPL|
|2003-01-09 00:00:00|14.62|14.68| AAPL|
|2003-01-10 00:00:00|14.58|14.72| AAPL|
|2003-01-13 00:00:00| 14.9|14.63| AAPL|
|2003-01-14 00:00:00|14.69|14.61| AAPL|
|2003-01-15 00:00:00|14.59|14.43| AAPL|
|2003-01-16 00:00:00|14.21|14.62| AAPL|
|2003-01-17 00:00:00|14.56| 14.1| AAPL|
|2003-01-21 00:00:00|14.21|14.02| AAPL|
|2003-01-22 00:00:00|13.98|13.88| AAPL|
|2003-01-23 00:00:00|14.05|14.17| AAPL|
|2003-01-24 00:00:00|14.24| 13.8| AAPL|
|2003-01-27 00:00:00|13.68|14.13| AAPL|
|2003-01-28 00:00:00|14.24|14.58| AAPL|
|2003-01-29 00:00:00|14.24|14.58| AAPL|
|2003-01-30 00:00:00|14.98|14.32| AAPL|
+-------------------+-----+-----+-----+
only showing top 20 rows
```

Casting

In [14]:
```
from pyspark.sql.types import StringType
dfCast=df.withColumn("Ndate", df["date"].cast(StringType()))
dfCast.show()
```

```
+-------------------+-----+-----+-----+-------------------+
|               date| open|close|stock|              Ndate|
+-------------------+-----+-----+-----+-------------------+
|2003-01-02 00:00:00|14.36| 14.8| AAPL|2003-01-02 00:00:00|
|2003-01-03 00:00:00| 14.8| 14.9| AAPL|2003-01-03 00:00:00|
|2003-01-06 00:00:00|15.03| 14.9| AAPL|2003-01-06 00:00:00|
|2003-01-07 00:00:00|14.79|14.85| AAPL|2003-01-07 00:00:00|
|2003-01-08 00:00:00|14.58|14.55| AAPL|2003-01-08 00:00:00|
|2003-01-09 00:00:00|14.62|14.68| AAPL|2003-01-09 00:00:00|
|2003-01-10 00:00:00|14.58|14.72| AAPL|2003-01-10 00:00:00|
|2003-01-13 00:00:00| 14.9|14.63| AAPL|2003-01-13 00:00:00|
|2003-01-14 00:00:00|14.69|14.61| AAPL|2003-01-14 00:00:00|
|2003-01-15 00:00:00|14.59|14.43| AAPL|2003-01-15 00:00:00|
|2003-01-16 00:00:00|14.21|14.62| AAPL|2003-01-16 00:00:00|
|2003-01-17 00:00:00|14.56| 14.1| AAPL|2003-01-17 00:00:00|
|2003-01-21 00:00:00|14.21|14.02| AAPL|2003-01-21 00:00:00|
|2003-01-22 00:00:00|13.98|13.88| AAPL|2003-01-22 00:00:00|
|2003-01-23 00:00:00|14.05|14.17| AAPL|2003-01-23 00:00:00|
|2003-01-24 00:00:00|14.24| 13.8| AAPL|2003-01-24 00:00:00|
|2003-01-27 00:00:00|13.68|14.13| AAPL|2003-01-27 00:00:00|
|2003-01-28 00:00:00|14.24|14.58| AAPL|2003-01-28 00:00:00|
|2003-01-29 00:00:00|14.24|14.58| AAPL|2003-01-29 00:00:00|
|2003-01-30 00:00:00|14.98|14.32| AAPL|2003-01-30 00:00:00|
+-------------------+-----+-----+-----+-------------------+
only showing top 20 rows
```

In [21]:
```
from pyspark.sql.types import TimestampType
import datetime
newdf=df.select("date","stock",year("date").alias('year'),month("date").alias('m
newdf.show()
```

```
+-------------------+-----+----+-----+---+
|               date|stock|year|month|day|
+-------------------+-----+----+-----+---+
|2003-01-02 00:00:00| AAPL|2003|    1|  2|
|2003-01-03 00:00:00| AAPL|2003|    1|  3|
|2003-01-06 00:00:00| AAPL|2003|    1|  6|
|2003-01-07 00:00:00| AAPL|2003|    1|  7|
|2003-01-08 00:00:00| AAPL|2003|    1|  8|
|2003-01-09 00:00:00| AAPL|2003|    1|  9|
|2003-01-10 00:00:00| AAPL|2003|    1| 10|
|2003-01-13 00:00:00| AAPL|2003|    1| 13|
|2003-01-14 00:00:00| AAPL|2003|    1| 14|
|2003-01-15 00:00:00| AAPL|2003|    1| 15|
|2003-01-16 00:00:00| AAPL|2003|    1| 16|
|2003-01-17 00:00:00| AAPL|2003|    1| 17|
|2003-01-21 00:00:00| AAPL|2003|    1| 21|
|2003-01-22 00:00:00| AAPL|2003|    1| 22|
|2003-01-23 00:00:00| AAPL|2003|    1| 23|
|2003-01-24 00:00:00| AAPL|2003|    1| 24|
|2003-01-27 00:00:00| AAPL|2003|    1| 27|
|2003-01-28 00:00:00| AAPL|2003|    1| 28|
|2003-01-29 00:00:00| AAPL|2003|    1| 29|
|2003-01-30 00:00:00| AAPL|2003|    1| 30|
+-------------------+-----+----+-----+---+
only showing top 20 rows
```

In [154]:
```
newdf.groupBy('stock').max().show()
```

```
+-----+---------+----------+--------+
|stock|max(year)|max(month)|max(day)|
+-----+---------+----------+--------+
|  AXP|     2007|        12|      31|
| AAPL|     2007|        12|      31|
|  AIG|     2007|        12|      31|
|   BA|     2007|        12|      31|
| AMZN|     2007|        12|      31|
+-----+---------+----------+--------+
```

In [16]:
```
import pyspark.sql.functions as F
df1 = df.withColumn("unix_timestamp",F.unix_timestamp(df.date,'dd-MMM-yyyy HH:mm
df1.show(5)
```

```
+-------------------+-----+-----+-----+--------------+
|               date| open|close|stock|unix_timestamp|
+-------------------+-----+-----+-----+--------------+
|2003-01-02 00:00:00|14.36| 14.8| AAPL|    1041483600|
|2003-01-03 00:00:00| 14.8| 14.9| AAPL|    1041570000|
|2003-01-06 00:00:00|15.03| 14.9| AAPL|    1041829200|
|2003-01-07 00:00:00|14.79|14.85| AAPL|    1041915600|
|2003-01-08 00:00:00|14.58|14.55| AAPL|    1042002000|
+-------------------+-----+-----+-----+--------------+
only showing top 5 rows
```

In [17]:
```
df2 = df1.withColumn("TimestampType",F.to_timestamp(df1["unix_timestamp"]))
print(df2.printSchema)
df2.show(n=2,truncate=False)
```

```
<bound method DataFrame.printSchema of DataFrame[date: timestamp, open: double,
close: double, stock: string, unix_timestamp: bigint, TimestampType: timestamp]
>
+-------------------+-----+-----+-----+--------------+-------------------+
|date               |open |close|stock|unix_timestamp|TimestampType      |
+-------------------+-----+-----+-----+--------------+-------------------+
|2003-01-02 00:00:00|14.36|14.8 |AAPL |1041483600    |2003-01-02 00:00:00|
|2003-01-03 00:00:00|14.8 |14.9 |AAPL |1041570000    |2003-01-03 00:00:00|
+-------------------+-----+-----+-----+--------------+-------------------+
only showing top 2 rows
```

In [18]:
```
df2=df1.groupBy("stock").max("unix_timestamp").withColumnRenamed('max(unix_times
#df2.show()

df2.withColumn("TimestampType",F.to_timestamp(df2["unix_timestamp"])).show()
```

```
+-----+--------------+-------------------+
|stock|unix_timestamp|      TimestampType|
+-----+--------------+-------------------+
|  AXP|    1199077200|2007-12-31 00:00:00|
| AAPL|    1199077200|2007-12-31 00:00:00|
|  AIG|    1199077200|2007-12-31 00:00:00|
|   BA|    1199077200|2007-12-31 00:00:00|
| AMZN|    1199077200|2007-12-31 00:00:00|
+-----+--------------+-------------------+
```

In [19]:
```
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf

spark = SparkSession.builder.appName('abc').getOrCreate()

sc = spark.sparkContext
```

In [20]:
```
import datetime

from pyspark.sql.functions import year, month, dayofmonth

elevDF = sc.parallelize([(datetime.datetime(1984, 1, 1, 0, 0), 1, 638.55),(datet

elevDF.select(year("date").alias('year'),month("date").alias('month'),dayofmonth
```

```
+----+-----+---+
|year|month|day|
+----+-----+---+
|1984|    1|  1|
|1984|    1|  1|
|1984|    1|  1|
|1984|    1|  1|
|1984|    1|  1|
+----+-----+---+
```

In [ ]: