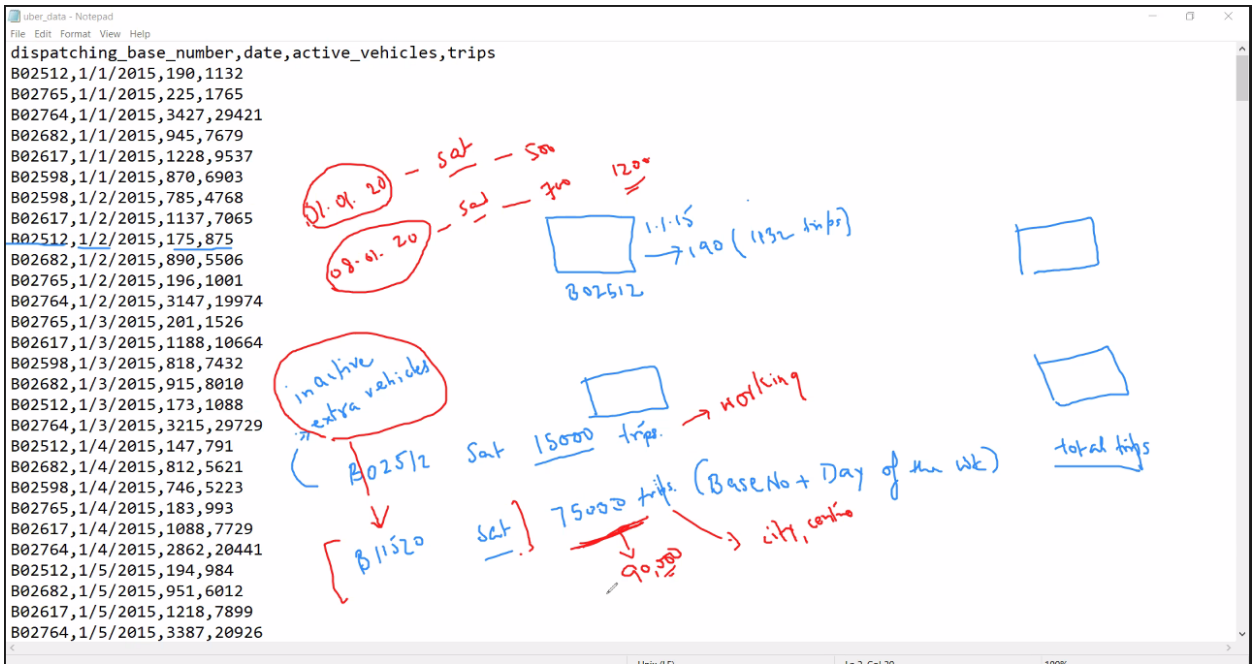
→ Uber Project

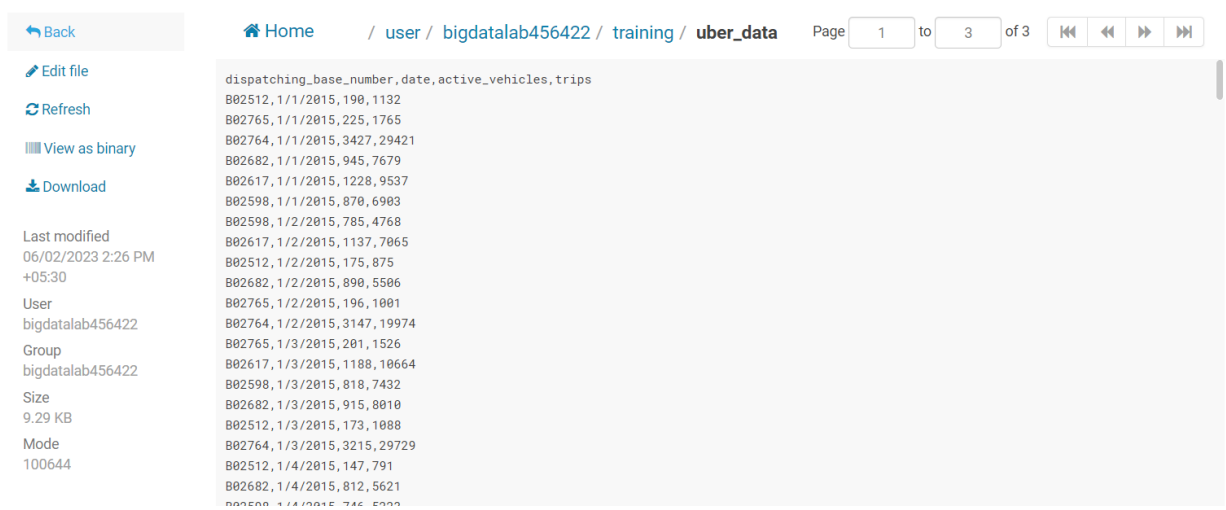


[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l uber_data

```
[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l uber_data
-rw-rw-r-- 1 bigdatalab456422 bigdatalab456422 9511 Jun  2 08:55 uber_data
```

[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop fs -put uber_data training

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop fs -put uber_data training
[bigdatalab456422@ip-10-1-1-204 ~]$
```



```
[bigdatalab456422@ip-10-1-1-204 ~]$ pyspark
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ pyspark
Python 3.7.6 (default, Jan  8 2020, 19:59:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/06/02 08:52:42 WARN cluster.YarnSchedulerBackend$YarnSchedulerEndpoint: Attempted to request executors before the AM has registered!
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.2.1
      /_/

Using Python version 3.7.6 (default, Jan  8 2020 19:59:22)
SparkSession available as 'spark'.
>>>
```

```
>>> dataset = sc.textFile("hdfs://nameservice1/user/bigdatalab456422/training/uber_data")
```

```
>>> dataset = sc.textFile("hdfs://nameservice1/user/bigdatalab456422/training/uber_data")
```

```
>>> dataset.count()
```

```
>>> dataset.count()
355
>>>
```

```
>>> dataset.first()
```

```
>>> dataset.first()
'dispatching_base_number,date,active_vehicles,trips'
>>>
```

```
>>> header = dataset.first()
```

```
>>> header = dataset.first()
>>>
```

```
>>> print(header)
```

```
>>> print(header)
dispatching_base_number,date,active_vehicles,trips
>>>
```

```
>>> eliminate = dataset.filter(lambda line : line != header)
```

```
>>> for a in eliminate.take(5):
...     print(a)
...
```

```
>>> for a in eliminate.take(5):
...     print(a)
...
B02512,1/1/2015,190,1132
B02765,1/1/2015,225,1765
B02764,1/1/2015,3427,29421
B02682,1/1/2015,945,7679
B02617,1/1/2015,1228,9537
>>>
```

```
>>>eliminate.count()
```

```
>>>eliminate.count()
354
>>>
```

```
>>> format_data = "%m/%d/%Y"
```

```
>>> format_data = "%m/%d/%Y"
>>>
```

```
>>> import datetime
```

```
>>> import datetime
>>>
```

```
>>> print(datetime.datetime.strptime('06/02/2023',format_data))
```

```
>>> print(datetime.datetime.strptime('06/02/2023',format_data))
2023-06-02 00:00:00
>>>
```

```
>>>
print(datetime.datetime.strptime('06/02/2023',format_data).strftime("
%A")
```

```
>>> print(datetime.datetime.strptime('06/02/2023',format_data).strftime("%A"))
Friday
>>>
```

```
>>>
print(datetime.datetime.strptime('06/02/2023',format_data).strftime("
%B"))
```

```
>>> print(datetime.datetime.strptime('06/02/2023',format_data).strftime("%B"))
June
>>>
```

```
>>> split = eliminate.map(lambda a : (a.split(",")[0],
datetime.datetime.strptime(a.split(",")[1],
format_data).strftime("%A"), a.split(",")[3]) )
```

```
>>> split = eliminate.map(lambda a : (a.split(",")[0], datetime.datetime.strptime(a.split(",")[1], format_data).strftime("%A"), a.split(",")[3]) )
>>>
```

```
>>> for a in split.take(5):
...     print(a)
...
```

```
>>> for a in split.take(5):
...     print(a)
...
('B02512', 'Thursday', '1132')
('B02765', 'Thursday', '1765')
('B02764', 'Thursday', '29421')
('B02682', 'Thursday', '7679')
('B02617', 'Thursday', '9537')
>>>
```

```
>>> combine = split.map(lambda x : ( x[0] +" "+x[1], int(x[2]) ))
```

```
>>> combine = split.map(lambda x : ( x[0] +" "+x[1], int(x[2]) ))
>>>
```

```
>>> for a in combine.take(5):
...     print(a)
...
```

```
>>> for a in combine.take(5):
...     print(a)
...
('B02512 Thursday', 1132)
('B02765 Thursday', 1765)
('B02764 Thursday', 29421)
('B02682 Thursday', 7679)
('B02617 Thursday', 9537)
>>>
```

```
>>> arrange = combine.reduceByKey(lambda a,b : a+b)
```

```
>>> arrange = combine.reduceByKey(lambda a,b : a+b)
>>>
```

```
>>> for a in arrange.take(5):
...     print(a)
...
```

```
>>> for a in arrange.take(5):
...     print(a)
...
('B02764 Thursday', 304200)
('B02682 Thursday', 106643)
('B02598 Thursday', 90333)
('B02682 Friday', 114662)
('B02764 Friday', 326968)
>>>
```

```
>>> sortbyval = arrange.sortBy(lambda a : -a[1])
```

```
>>> sortbyval = arrange.sortBy(lambda a : -a[1])
>>>
```

```
>>> for a in sortbyval.collect():
...     print(a)
...
```

```
>>> for a in sortbyval.collect():
...     print(a)
...
('B02764 Saturday', 356789)
('B02764 Friday', 326968)
('B02764 Thursday', 304200)
('B02764 Sunday', 249896)
('B02764 Wednesday', 241137)
('B02764 Tuesday', 221343)
('B02764 Monday', 214116)
('B02617 Saturday', 127902)
('B02617 Friday', 125067)
('B02682 Saturday', 120283)
('B02617 Thursday', 118254)
('B02682 Friday', 114662)
('B02512 Saturday', 15026)
('B02512 Wednesday', 12691)
('B02512 Tuesday', 12041)
('B02512 Monday', 11297)
('B02512 Sunday', 10487)
>>>
```

→ more partitions makes faster processing but will use more processor cores
→ Recommended to use At max 5 cores
→ spark creates partitions for processing, but hadoop creates partitions for storage

```
127 login: bigdatalab45644
bigdatalab45644@127.0.0.1's password:
Last login: Thu Jun  1 11:07:58 2023 from localhost
[bigdatalab45644@ip-10-1-1-204 ~]$ pyspark
Python 3.7.6 (default, Jan  8 2020, 19:59:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.2.1
      /_/

Using Python version 3.7.6 (default, Jan  8 2020 19:59:22)
SparkSession available as 'spark'.
>>> dataset = sc.textFile("hdfs://nameservice1/user/bigdatalab45644/training/uber_data")
>>> dataset.count()
[Stage 0:>                                                          (0 + 0) / 2]23/06/02 08:58:59 WARN cluster.YarnScheduler: Initial job has not accepted any resources
; check your cluster UI to ensure that workers are registered and have sufficient resources
355
>>> dataset.first()
'dispatching_base_number,date,active_vehicles,trips'
>>> header = dataset.first()
>>> print(header)
dispatching_base_number,date,active_vehicles,trips
>>> dataset.getNumPartitions()
2
>>>
```

20 cores

5

? 5

p0 p1

No of partition = no. of cores

2 – cores
4 – partitions – queue

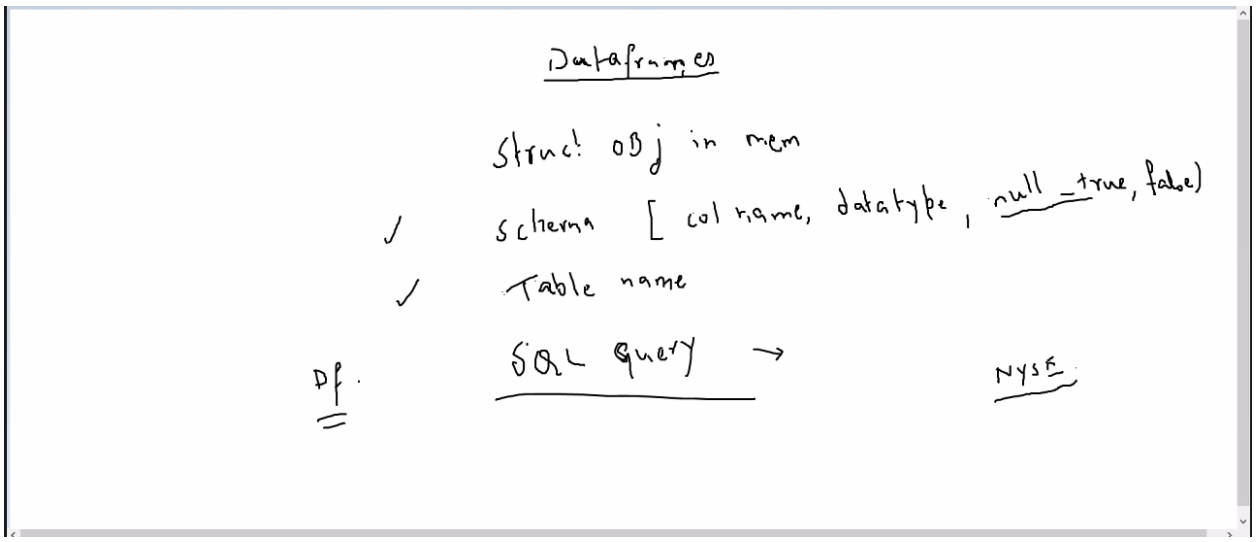
```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.2.1
      /_/

Using Python version 3.7.6 (default, Jan  8 2020 19:59:22)
SparkSession available as 'spark'.
>>> dataset = sc.textFile("hdfs://nameservice1/user/bigdatalab45644/training/uber_data")
>>> dataset.count()
[Stage 0:>                                                          (0 + 0) / 2]23/06/02 08:58:59 WARN cluster.YarnScheduler: Initial job has not accepted any resources
; check your cluster UI to ensure that workers are registered and have sufficient resources
355
>>> dataset.first()
'dispatching_base_number,date,active_vehicles,trips'
>>> header = dataset.first()
>>> print(header)
dispatching_base_number,date,active_vehicles,trips
>>> dataset.getNumPartitions()
2
>>> eliminate = dataset.filter(lambda line : line != header)
>>> for a in eliminate.take(5):
...     print(a)
...
B02512,1/1/2015,190,1132
B02765,1/1/2015,225,1765
B02764,1/1/2015,3427,29421
B02682,1/1/2015,945,7679
B02617,1/1/2015,1228,9537
>>> eliminate.count()
354
>>> format_data = "%m/%d/%Y"
>>> import datetime
>>> print(datetime.datetime.strptime('06/02/2023',format_data)
... )
2023-06-02 00:00:00
>>> print(datetime.datetime.strptime('06/02/2023',format_data).strftime("%A"))
Friday
>>> print(datetime.datetime.strptime('06/02/2023',format_data).strftime("%B"))
June
>>>
```

strptime = converts string to dt
strftime = dt to str

→ DataFrames

a. Structured object in memory
b. Has a schema [col name, data type, null_true, false]
c. We have to give a table name
d. Once table is created, we can use SQL query to do operations



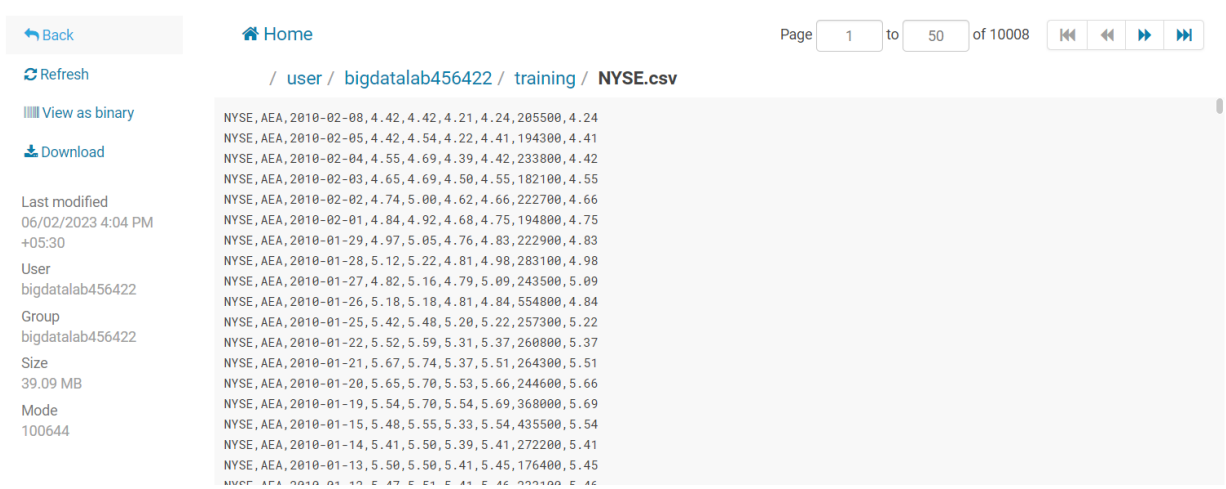
→ DataFrames Project with NYSE data

```
[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l NYSE.csv
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l NYSE.csv
-rw-rw-r-- 1 bigdatalab456422 bigdatalab456422 40990862 May 17 09:21 NYSE.csv
[bigdatalab456422@ip-10-1-1-204 ~]$
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop fs -put NYSE.csv training
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop fs -put NYSE.csv training
[bigdatalab456422@ip-10-1-1-204 ~]$
```



```
[bigdatalab456422@ip-10-1-1-204 ~]$ pyspark
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ pyspark
Python 3.7.6 (default, Jan  8 2020, 19:59:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.2.1
      /_/

Using Python version 3.7.6 (default, Jan  8 2020 19:59:22)
SparkSession available as 'spark'.
>>> 
```

```
>>> from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType, LongType
```

```
>>> from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType, LongType
>>>
```

creates an empty schema for mapping the dataFrame with dataFile

```
>>> schema9 = StructType().add("exchange_name",StringType(),True).add("stock_id",StringType(),True).add("stock_dt",StringType(),True).add("open",DoubleType(),True).add("high",DoubleType(),True).add("low",DoubleType(),True).add("close",DoubleType(),True).add("volume",LongType(),True).add("adj_close",DoubleType(),True)
```

```
>>> schema9 = StructType().add("exchange_name",StringType(),True).add("stock_id",StringType(),True).add("stock_dt",StringType(),True).add("open",DoubleType(),True).add(
"high",DoubleType(),True).add("low",DoubleType(),True).add("close",DoubleType(),True).add("volume",LongType(),True).add("adj_close",DoubleType(),True)
>>> 
```

```
>>>print(schema9)
```

```
>>>print(schema9)
StructType(List(StructField(exchange_name,StringType,true),StructField(stock_id,StringType,true),StructField(stock_dt,StringType,true),StructField(open,DoubleType,true)
,StructField(high,DoubleType,true),StructField(low,DoubleType,true),StructField(close,DoubleType,true),StructField(volume,LongType,true),StructField(adj_close,DoubleTyp
e,true)))
>>>
```

creating dataFrame with schema and loading the textFile into DataFrame

```
>>> df_with_schema = spark.read.format("csv").option("header","False").schema(schema9).load("hdfs://nameservice1/user/bigdatalab456422/training/NYSE.csv")
```

```
>>> df_with_schema = spark.read.format("csv").option("header","False").schema(schema9).load("hdfs://nameservice1/user/bigdatalab456422/training/NYSE.csv")
>>>
```

```
>>> df_with_schema.printSchema()
```

```
>>> df_with_schema.printSchema()
root
 |-- exchange_name: string (nullable = true)
 |-- stock_id: string (nullable = true)
 |-- stock_dt: string (nullable = true)
 |-- open: double (nullable = true)
 |-- high: double (nullable = true)
 |-- low: double (nullable = true)
 |-- close: double (nullable = true)
 |-- volume: long (nullable = true)
 |-- adj_close: double (nullable = true)

>>> 
```

```
>>> df_with_schema.show()
```

```
>>> df_with_schema.show()
+-------------+--------+----------+----+----+----+-----+------+---------+
|exchange_name|stock_id|  stock_dt|open|high| low|close|volume|adj_close|
+-------------+--------+----------+----+----+----+-----+------+---------+
|         NYSE|     AEA|2010-02-08|4.42|4.42|4.21| 4.24|205500|     4.24|
|         NYSE|     AEA|2010-02-05|4.42|4.54|4.22| 4.41|194300|     4.41|
|         NYSE|     AEA|2010-02-04|4.55|4.69|4.39| 4.42|233800|     4.42|
|         NYSE|     AEA|2010-02-03|4.65|4.69| 4.5| 4.55|182100|     4.55|
|         NYSE|     AEA|2010-02-02|4.74| 5.0|4.62| 4.66|222700|     4.66|
|         NYSE|     AEA|2010-02-01|4.84|4.92|4.68| 4.75|194800|     4.75|
|         NYSE|     AEA|2010-01-29|4.97|5.05|4.76| 4.83|222900|     4.83|
|         NYSE|     AEA|2010-01-28|5.12|5.22|4.81| 4.98|283100|     4.98|
|         NYSE|     AEA|2010-01-27|4.82|5.16|4.79| 5.09|243500|     5.09|
|         NYSE|     AEA|2010-01-26|5.18|5.18|4.81| 4.84|554800|     4.84|
|         NYSE|     AEA|2010-01-25|5.42|5.48| 5.2| 5.22|257300|     5.22|
|         NYSE|     AEA|2010-01-22|5.52|5.59|5.31| 5.37|260800|     5.37|
|         NYSE|     AEA|2010-01-21|5.67|5.74|5.37| 5.51|264300|     5.51|
|         NYSE|     AEA|2010-01-20|5.65| 5.7|5.53| 5.66|244600|     5.66|
|         NYSE|     AEA|2010-01-19|5.54| 5.7|5.54| 5.69|368000|     5.69|
|         NYSE|     AEA|2010-01-15|5.48|5.55|5.33| 5.54|435500|     5.54|
|         NYSE|     AEA|2010-01-14|5.41| 5.5|5.39| 5.41|272200|     5.41|
|         NYSE|     AEA|2010-01-13| 5.5| 5.5|5.41| 5.45|176400|     5.45|
|         NYSE|     AEA|2010-01-12|5.47|5.51|5.41| 5.46|233100|     5.46|
|         NYSE|     AEA|2010-01-11|5.64|5.64|5.49| 5.55|178900|     5.55|
+-------------+--------+----------+----+----+----+-----+------+---------+
only showing top 20 rows

>>>
```

converting dataframe to SQL table, as we can't run SQL queries on DataFrame because DataFrame is a python object

```
>>> df_with_schema.registerTempTable("nyse")
```

```
>>> df_with_schema.registerTempTable("nyse")
>>>
```

→ creates 200 partitions by default for dataframes, but for sql tables it creates 1 partition

```
>>> df_StockVol = spark.sql("SELECT stock_id, sum(volume) FROM nyse GROUP BY stock_id")
```

```
>>> df_StockVol = spark.sql("SELECT stock_id, sum(volume) FROM nyse GROUP BY stock_id")
>>>
```

```
>>> df_StockVol.count()
```

```
>>> df_StockVol.count()
203
>>>
```

```
>>> df_StockVol.show()
```

```
>>> df_StockVol.show()
+--------+-----------+
|stock_id|sum(volume)|
+--------+-----------+
|     APX|  140637400|
|     AIV| 2922335500|
|     AVY| 2597060500|
|     AVX| 1347876500|
|     ARL|   46445200|
|     AXP|40263020300|
|     AAV|  834246600|
|     AOB| 1597133000|
|     ARM| 2087366800|
|     ASH| 2717488700|
|     AEB|   53273300|
|     ALE|  465167800|
|     ACH| 1448279800|
|     ABA|   11686500|
|     ASF|  848352700|
|     ABK|11899868300|
|     ATU| 1226088700|
|     ALG|   64657600|
|      AM| 2963437400|
|      AA|42061448400|
+--------+-----------+
only showing top 20 rows

>>>
```

```
>>> df_StockVol = spark.sql("SELECT stock_id, sum(volume) AS total
FROM nyse GROUP BY stock_id ORDER BY total DESC")
```

```
>>> df_StockVol = spark.sql("SELECT stock_id, sum(volume) AS total FROM nyse GROUP BY stock_id ORDER BY total DESC")
>>>
```

```
>>> df_StockVol.count()
```

```
>>> df_StockVol.count()
203
>>>
```

```
>>> df_StockVol.show()
```

```
>>> df_StockVol.show()
+--------+-----------+
|stock_id|      total|
+--------+-----------+
|     AMD|47252808500|
|      AA|42061448400|
|     AXP|40263020300|
|     AET|30218027200|
|     ABT|25664130200|
|     AMR|22505621700|
|     AVP|20750196700|
|     ABX|16691172100|
|     APC|15555731900|
|     ADM|15354593500|
|     AEO|14731442100|
|     ADI|14597316000|
|     ALU|12804053900|
|     ABK|11899868300|
|     AES|11884945300|
|     ALL|11492379500|
|     APA|11482389600|
|     ABC|11439581700|
|     ADP|11358284900|
|     AUY|11034706100|
+--------+-----------+
only showing top 20 rows

>>>
```

```
>>> df_StockVol.rdd.getNumPartitions()
```

```
>>> df_StockVol.rdd.getNumPartitions()
200
>>>
```

→ use coalesce(1) to reduce no of partitions
→ use repartition(1) to increase no of partitions
##alt `df_new = df_StockVol.repartition(1)`

```
>>> df_new = df_StockVol.coalesce(1)
```

```
>>> df_new = df_StockVol.coalesce(1)
>>>
```

```
>>> df_new.rdd.getNumPartitions()
```

```
>>> df_new.rdd.getNumPartitions()
1
>>>
```

now we are saving the file

```
>>>
df_new.write.csv("hdfs://nameservice1/user/bigdatalab456422/training/
spark4")
```

```
>>> df_new.write.csv("hdfs://nameservice1/user/bigdatalab456422/training/spark4")
>>>
```

Home / user / bigdatalab456422 / training / **spark4**

Trash

Name	Size	User	Group	Permissions	Date
↑		bigdatalab456422	bigdatalab456422	drwxr-xr-x	June 02, 2023 04:56 AM
.		bigdatalab456422	bigdatalab456422	drwxr-xr-x	June 02, 2023 04:57 AM
_SUCCESS	0 bytes	bigdatalab456422	bigdatalab456422	-rw-r--r--	June 02, 2023 04:57 AM
part-00000-ff257a66-bb02-41cd-92d8-fd733fe6e32b-c000.csv	2.8 KB	bigdatalab456422	bigdatalab456422	-rw-r--r--	June 02, 2023 04:57 AM

Show 45 of 2 items

Page 1 of 1

Back

Edit file

Refresh

View as binary

Download

Last modified
06/02/2023 5:27 PM
+05:30

User
bigdatalab456422

Group
bigdatalab456422

Size
2.85 KB

Mode
100644

Home

Page 1 to 1 of 1

/ user / bigdatalab456422 / training / spark4 /
part-00000-ff257a66-bb02-41cd-92d8-fd733fe6e32b-c000.csv

```
AMD,47252808500
AA,42061448400
AXP,40263020300
AET,30218027200
ABT,25664130200
AMR,22505621700
AVP,20750196700
ABX,16691172100
APC,15555731900
ADM,15354593500
AEO,14731442100
ADI,14597316000
ALU,12804053900
ABK,11899868300
AES,11884945300
ALL,11492379500
APA,11482389600
```

→ Scala & Spark installation
Refer spark installation.docx file

```
sdevsinx@bdt0:~$ java -version
```

```
sdevsinx@bdt0:~$ java -version
openjdk version "1.8.0_362"
OpenJDK Runtime Environment (build 1.8.0_362-8u372-ga~us1-0ubuntu1~22.04-b09)
OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)
sdevsinx@bdt0:~$
```

```
sdevsinx@bdt0:~$ su hduser
```

```
sdevsinx@bdt0:~$ su hduser
Password:
hduser@bdt0:/home/sdevsinx$
```

```
hduser@bdt0:/home/sdevsinx$ cd ~
```

```
hduser@bdt0:/home/sdevsinx$ cd ~
hduser@bdt0:~$
```

```
hduser@bdt0:~$ sudo apt update
```

```
hduser@bdt0:~$ sudo apt install scala
```

```
hduser@bdt0:~$ scala -version
```

```
hduser@bdt0:~$ scala -version
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
hduser@bdt0:~$
```

```
hduser@bdt0:~$ wget
http://archive.apache.org/dist/spark/spark-2.1.0/spark-2.1.0-bin-hadoop2.6.tgz
```

```
hduser@bdt0:~$ wget http://archive.apache.org/dist/spark/spark-2.1.0/spark-2.1.0
-bin-hadoop2.6.tgz
--2023-06-03 12:28:04--  http://archive.apache.org/dist/spark/spark-2.1.0/spark-
2.1.0-bin-hadoop2.6.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a
:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:80... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: 193281941 (184M) [application/x-gzip]
Saving to: ‘spark-2.1.0-bin-hadoop2.6.tgz’

spark-2.1.0-bin-had 100%[===================>] 184.33M  2.41MB/s    in 72s

2023-06-03 12:29:16 (2.57 MB/s) - ‘spark-2.1.0-bin-hadoop2.6.tgz’ saved [1932819
41/193281941]

hduser@bdt0:~$
```

```
hduser@bdt0:~$ tar -xvzf spark-2.1.0-bin-hadoop2.6.tgz
```

```
hduser@bdt0:~$ mv spark-2.1.0-bin-hadoop2.6 spark
```

```
hduser@bdt0:~$ sudo mv spark /usr/local/
```

```
hduser@bdt0:~$ nano ~/.bashrc
```

```
# HIVE VARIABLES END


# SPARK VARIABLES START
export PATH=$PATH:/usr/local/spark/bin
# SPARK VARIABLES END


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
hduser@bdt0:~$ source ~/.bashrc
```

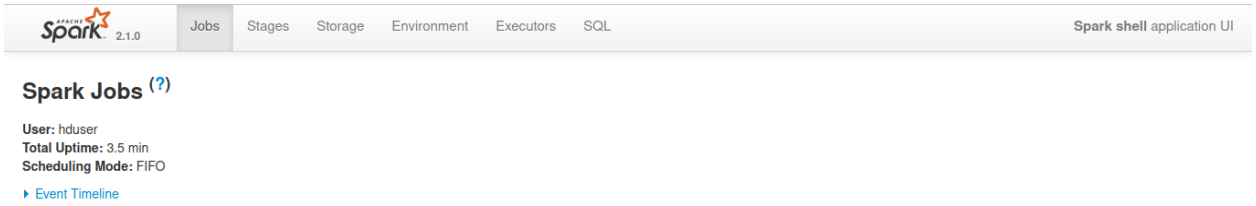
```
hduser@bdt0:~$ spark-shell
```

```
hduser@bdt0:~$ spark-shell
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
23/06/03 12:42:45 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
23/06/03 12:42:45 WARN Utils: Your hostname, bdt0 resolves to a loopback address
: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
23/06/03 12:42:45 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
23/06/03 12:43:05 WARN ObjectStore: Version information not found in metastore.
hive.metastore.schema.verification is not enabled so recording the schema versio
n 1.2.0
23/06/03 12:43:06 WARN ObjectStore: Failed to get database default, returning No
SuchObjectException
23/06/03 12:43:09 WARN ObjectStore: Failed to get database global_temp, returnin
g NoSuchObjectException
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1685776368437
).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.1.0
      /_/

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_362)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

For Web UI logon to
http://localhost:4040/



Spark Jobs (?)

User: hduser
Total Uptime: 3.5 min
Scheduling Mode: FIFO

Event Timeline

wget http://archive.apache.org/dist/spark/spark-2.1.0/spark-2.1.0-bin-hadoop2.6.tgz