In [1]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('basics').getOrCreate()
```

In [6]:
```
df.show()
```

```
+----+-------+
| age|   name|
+----+-------+
|null|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

In [7]:
```
df.printSchema()
```

```
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)
```

In [60]:
```
print(df.columns)
```

```
['age', 'name']
```

In [9]:
```
# get description of data frame
df.describe().show()
```

```
+-------+------------------+-------+
|summary|               age|   name|
+-------+------------------+-------+
|  count|                 2|      3|
|   mean|              24.5|   null|
| stddev|7.7781745930520225|   null|
|    min|                19|   Andy|
|    max|                30|Michael|
+-------+------------------+-------+
```

In [62]:
```
df.printSchema()
```

```
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)
```

In [63]:
```
# Chnage the data types
from pyspark.sql.types import (StructField, StringType, IntegerType,StructType)
```

In [64]:
```
data_schema = [StructField('age',IntegerType(),True),
               StructField('name',StringType(),True)]
```

In [65]:
```
final_struct = StructType(fields=data_schema)
```

In [66]:
```
df = spark.read.json(r'D:\Python\PySpark\Python-and-Spark-for-Big-Data-master\Spark_DataFrames\people.json',schema= fi
nal_struct)
```

In [67]:
```
df.show()
```

```
+----+-------+
| age|   name|
+----+-------+
|null|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

In [68]:
```
df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- name: string (nullable = true)
```

In [69]:
```
# print single column
df.select('age').show()
```

```
+----+
| age|
+----+
|null|
|  30|
|  19|
+----+
```

In [70]:
```
# print multiple columns
df.select(['age','name']).show()
```

```
+----+-------+
| age|   name|
+----+-------+
|null|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

In [71]:
```
# create a new column and copy existing values of 'age'
df.withColumn('newage',df['age']).show()
```

```
+----+-------+------+
| age|   name|newage|
+----+-------+------+
|null|Michael|  null|
|  30|   Andy|    30|
|  19| Justin|    19|
+----+-------+------+
```

In [79]:
```
# create new DF and add new column
d_age_df = df.withColumn('double_age',df['age']*2)
```

In [81]:
```
d_age_df.show()
```

```
+----+-------+----------+
| age|   name|double_age|
+----+-------+----------+
|null|Michael|      null|
|  30|   Andy|        60|
|  19| Justin|        38|
+----+-------+----------+
```

In [86]:
```
# creating a view and querying the view using SQL
df.createOrReplaceTempView('people') # here 'people is the name of the view
```

In [88]:
```
result = spark.sql('select * from people')
result.show()
```

```
+----+-------+
| age|   name|
+----+-------+
|null|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

Basic Operations on DataFrame

In [7]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('df_operations').getOrCreate()
```

In [8]:
```
spark
```

Out[8]:

SparkSession - in-memory

SparkContext

Spark UI (http://WISHYD3007.WISSIND.COM:4040)

Version

v3.1.1

Master

local[*]

AppName

df_operations

In [11]:
```
df_csv = spark.read.csv(r'D:\Python\PySpark\Python-and-Spark-for-Big-Data-master\Spark_DataFrames\appl_stock.csv',infe
rSchema= True, header= True)
```

In [23]:
```
# Show column datatypes
df_csv.printSchema()
```

```
root
 |-- Date: string (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Volume: integer (nullable = true)
 |-- Adj Close: double (nullable = true)
```

In [27]:
```
# get count of the rows
df_csv.count()
```

Out[27]: 1762

In [28]:
```
## print data, by default it prints top 20 rows
df_csv.show(10)
```

```
+----------+------------------+------------------+------------------+------------------+---------+------------------+
|      Date|              Open|              High|               Low|             Close|   Volume|         Adj Close|
+----------+------------------+------------------+------------------+------------------+---------+------------------+
|2010-01-04|        213.429998|        214.499996|212.38000099999996|        214.009998|123432400|         27.727039|
|2010-01-05|        214.599998|        215.589994|        213.249994|        214.379993|150476200|27.774976000000002|
|2010-01-06|        214.379993|            215.23|        210.750004|        210.969995|138040000|27.333178000000004|
|2010-01-07|            211.75|        212.000006|        209.050005|            210.58|119282800|          27.28265|
|2010-01-08|        210.299994|        212.000006|209.06000500000002|211.98000499999998|111902700|         27.464034|
|2010-01-11|212.79999700000002|        213.000002|        208.450005|210.11000299999998|115557400|         27.221758|
|2010-01-12|209.18999499999998|209.76999500000002|        206.419998|        207.720001|148614900|          26.91211|
|2010-01-13|        207.870005|210.92999500000002|        204.099998|        210.650002|151473000|          27.29172|
|2010-01-14|210.11000299999998|210.45999700000002|        209.020004|            209.43|108223500|         27.133657|
|2010-01-15|210.92999500000002|211.59999700000003|        205.869999|            205.93|148516900|26.680197999999997|
+----------+------------------+------------------+------------------+------------------+---------+------------------+
only showing top 10 rows
```

In [37]:
```
# Shape of Spark DF
print('Shape is:({},{})'.format(df_csv.count(),len(df_csv.columns)))
```

```
Shape is:(1762,7)
```

Filtering the Data :

In [26]:
```
# filter rows based on the condition
df_csv.filter(df_csv['Date'] > '2010-01-11').show()
```

```
+----------+------------------+------------------+------------------+------------------+---------+------------------+
|      Date|              Open|              High|               Low|             Close|   Volume|         Adj Close|
+----------+------------------+------------------+------------------+------------------+---------+------------------+
|2010-01-12|209.18999499999998|209.76999500000002|        206.419998|        207.720001|148614900|          26.91211|
|2010-01-13|        207.870005|210.92999500000002|        204.099998|        210.650002|151473000|          27.29172|
|2010-01-14|210.11000299999998|210.45999700000002|        209.020004|            209.43|108223500|         27.133657|
|2010-01-15|210.92999500000002|211.59999700000003|        205.869999|            205.93|148516900|26.680197999999997|
|2010-01-19|        208.330002|215.18999900000003|        207.240004|        215.039995|182501900|27.860484999999997|
|2010-01-20|        214.910006|        215.549994|        209.500002|            211.73|153038200|         27.431644|
|2010-01-21|        212.079994|213.30999599999998|        207.210003|        208.069996|152038600|         26.957455|
|2010-01-22|206.78000600000001|        207.499996|            197.16|            197.75|220441900|         25.620401|
|2010-01-25|202.51000200000001|        204.699999|        200.190002|        203.070002|266424900|26.309658000000002|
|2010-01-26|205.95000100000001|        213.710005|        202.580004|        205.940001|466777500|         26.681494|
|2010-01-27|        206.849995|            210.58|        199.530001|        207.880005|430642100|26.932840000000002|
|2010-01-28|        204.930004|        205.500004|        198.699995|        199.289995|293375600|25.819922000000002|
|2010-01-29|        201.079996|        202.199995|        190.250002|        192.060003|311488100|          24.883208|
|2010-02-01|192.36999699999998|             196.0|191.29999899999999|        194.729998|187469100|          25.229131|
|2010-02-02|        195.909998|        196.319994|193.37999299999998|        195.859997|174585600|25.375532999999997|
|2010-02-03|        195.169994|        200.200003|        194.420004|        199.229994|153832000|25.812148999999998|
|2010-02-04|        196.730003|        198.370001|        191.570005|        192.050003|189413000|          24.881912|
|2010-02-05|192.63000300000002|             196.0|        190.850002|        195.460001|212576700|25.323710000000002|
|2010-02-08|        195.690006|197.88000300000002|        193.999994|194.11999699999998|119567700|            25.1501|
|2010-02-09|        196.419996|        197.499994|        194.749998|196.19000400000002|158221700|          25.418289|
+----------+------------------+------------------+------------------+------------------+---------+------------------+
only showing top 20 rows
```

In [21]:
```
df_csv.filter(df_csv['Date']>'2010-01-11').select(['Date','Open','High','Low']).show()
```

```
+----------+------------------+------------------+------------------+
|      Date|              Open|              High|               Low|
+----------+------------------+------------------+------------------+
|2010-01-12|209.18999499999998|209.76999500000002|        206.419998|
|2010-01-13|        207.870005|210.92999500000002|        204.099998|
|2010-01-14|210.11000299999998|210.45999700000002|        209.020004|
|2010-01-15|210.92999500000002|211.59999700000003|        205.869999|
|2010-01-19|        208.330002|215.18999900000003|        207.240004|
|2010-01-20|        214.910006|        215.549994|        209.500002|
|2010-01-21|        212.079994|213.30999599999998|        207.210003|
|2010-01-22|206.78000600000001|        207.499996|            197.16|
|2010-01-25|202.51000200000001|        204.699999|        200.190002|
|2010-01-26|205.95000100000001|        213.710005|        202.580004|
|2010-01-27|        206.849995|            210.58|        199.530001|
|2010-01-28|        204.930004|        205.500004|        198.699995|
|2010-01-29|        201.079996|        202.199995|        190.250002|
|2010-02-01|192.36999699999998|             196.0|191.29999899999999|
|2010-02-02|        195.909998|        196.319994|193.37999299999998|
|2010-02-03|        195.169994|        200.200003|        194.420004|
|2010-02-04|        196.730003|        198.370001|        191.570005|
|2010-02-05|192.63000300000002|             196.0|        190.850002|
|2010-02-08|        195.690006|197.88000300000002|        193.999994|
|2010-02-09|        196.419996|        197.499994|        194.749998|
+----------+------------------+------------------+------------------+
only showing top 20 rows
```

In [96]:
```
# filter rows based on the condition
df_csv.filter("High > 210 ").show()
```

```
+----------+-----------------+-----------------+-----------------+-----------------+---------+-----------------+
|      Date|             Open|             High|              Low|            Close|   Volume|        Adj Close|
+----------+-----------------+-----------------+-----------------+-----------------+---------+-----------------+
|2010-01-04|       213.429998|       214.499996|212.38000099999996|       214.009998|123432400|         27.727039|
|2010-01-05|       214.599998|       215.589994|       213.249994|       214.379993|150476200|27.774976000000002|
|2010-01-06|       214.379993|           215.23|       210.750004|       210.969995|138040000|27.333178000000004|
|2010-01-07|           211.75|       212.000006|       209.050005|           210.58|119282800|          27.28265|
|2010-01-08|       210.299994|       212.000006|209.06000500000002|211.98000499999998|111902700|         27.464034|
|2010-01-11|212.79999700000002|       213.000002|       208.450005|210.11000299999998|115557400|         27.221758|
|2010-01-13|       207.870005|210.92999500000002|       204.099998|       210.650002|151473000|          27.29172|
|2010-01-14|210.11000299999998|210.45999700000002|       209.020004|           209.43|108223500|         27.133657|
|2010-01-15|210.92999500000002|211.59999700000003|       205.869999|           205.93|148516900|26.680197999999997|
|2010-01-19|       208.330002|215.18999900000003|       207.240004|       215.039995|182501900|27.860484999999997|
|2010-01-20|       214.910006|       215.549994|       209.500002|           211.73|153038200|         27.431644|
|2010-01-21|       212.079994|213.30999599999998|       207.210003|       208.069996|152038600|         26.957455|
|2010-01-26|205.95000100000001|       213.710005|       202.580004|       205.940001|466777500|         26.681494|
|2010-01-27|       206.849995|           210.58|       199.530001|       207.880005|430642100|26.932840000000002|
|2010-03-02|       209.929998|       210.830006|       207.740002|           208.85|141636600|         27.058512|
|2010-03-04|       209.279997|       210.919994|       208.629995|210.71000299999997| 91510300|         27.299493|
|2010-03-05|       214.940006|219.69999500000003|214.62999900000003|218.95000499999998|224905100|28.367064000000003|
|2010-03-08|220.01000200000001|       220.090004|       218.250002|       219.079994|107472400|         28.383906|
|2010-03-09|218.31000299999997|       224.999996|       217.889994|       223.020004|230064800|28.894371999999997|
|2010-03-10|       223.829996|225.48000699999997|223.19999500000003|224.83999300000002|149054500|29.130167999999998|
+----------+-----------------+-----------------+-----------------+-----------------+---------+-----------------+
only showing top 20 rows
```

In [111]:
```
# get only selected columns
df_csv.filter(" High > 200 ").select(['Volume', 'Adj Close']).show()
```

```
+---------+-----------------+
|   Volume|        Adj Close|
+---------+-----------------+
|123432400|         27.727039|
|150476200|27.774976000000002|
|138040000|27.333178000000004|
|119282800|          27.28265|
|111902700|         27.464034|
|115557400|         27.221758|
|148614900|          26.91211|
|151473000|          27.29172|
|108223500|         27.133657|
|148516900|26.680197999999997|
|182501900|27.860484999999997|
|153038200|         27.431644|
|152038600|         26.957455|
|220441900|         25.620401|
|266424900|26.309658000000002|
|466777500|         26.681494|
|430642100|26.932840000000002|
|293375600|25.819922000000002|
|311488100|         24.883208|
|153832000|25.812148999999998|
+---------+-----------------+
only showing top 20 rows
```

In [112]:
```
# another way to filter
df_csv.filter(df_csv['High'] > 200).select(['Volume', 'Adj Close']).show()
```

```
+---------+------------------+
|   Volume|         Adj Close|
+---------+------------------+
|123432400|         27.727039|
|150476200|27.774976000000002|
|138040000|27.333178000000004|
|119282800|          27.28265|
|111902700|         27.464034|
|115557400|         27.221758|
|148614900|          26.91211|
|151473000|          27.29172|
|108223500|         27.133657|
|148516900|26.680197999999997|
|182501900|27.860484999999997|
|153038200|         27.431644|
|152038600|         26.957455|
|220441900|         25.620401|
|266424900|26.309658000000002|
|466777500|         26.681494|
|430642100|26.932840000000002|
|293375600|25.819922000000002|
|311488100|         24.883208|
|153832000|25.812148999999998|
+---------+------------------+
only showing top 20 rows
```

In [38]:
```
## if you have multiple conditions then pass the each condition in different paranthesis
df_csv.filter((df_csv['High']>200) & (df_csv['High'] < 203)).select(['Volume', 'Adj Close']).show()
```

```
+---------+------------------+
|   Volume|         Adj Close|
+---------+------------------+
|311488100|         24.883208|
|153832000|25.812148999999998|
|163867200|25.961142000000002|
| 97640900|         25.966324|
|143773700|         25.531005|
|115141600|         25.997419|
|166281500|          26.17103|
+---------+------------------+
```

In [ ]:
```

```