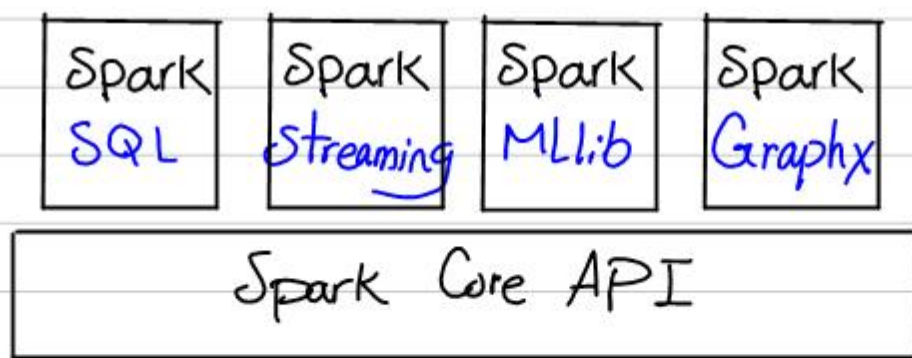
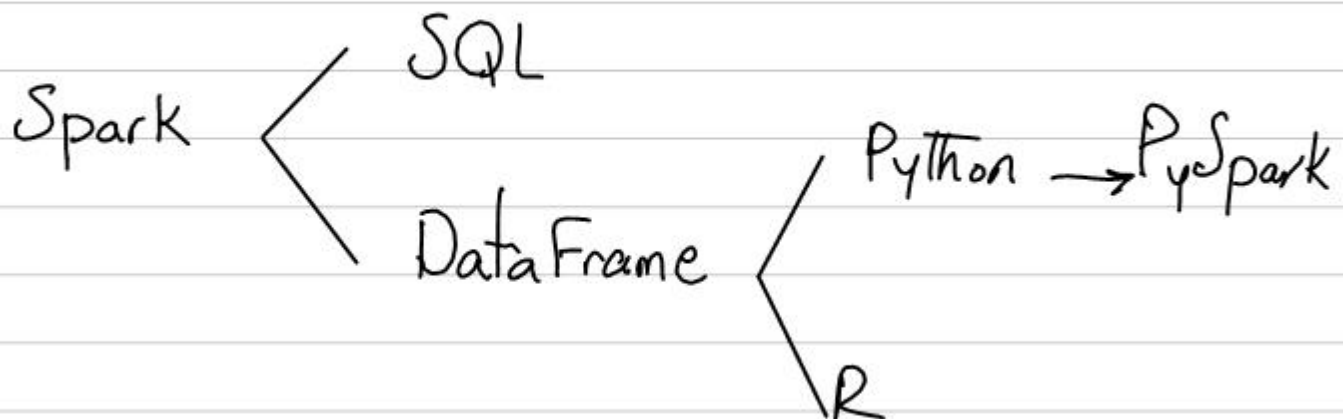


■ Apache Spark | PySpark

- is developed on Scala



- Task Scheduling
- Memory Management
- Interacting with Storage System
- Lazy evaluation



Spark

vs.

Pandas

Tabular data

Can handle millions rows

limited to a single machine

↓ Spark
distributed

Hadoop

distributed

Compute System
(MapReduce)

Storage System
(HDFS)

Hadoop Distributed File System

↓
read from & write to disk

↓ Spark
in memory

Spark Data APIs

- DataFrames → high-level API
- RDDs → low-level API

```
From pyspark.sql.functions import min, max, arg, asc, desc
```

```
From pyspark.sql.types import FloatType, IntegerType
```

```
From pyspark.ml.feature import Imputer
```

```
From pyspark.ml.feature import VectorAssembler
```

```
From pyspark.ml.regression import LinearRegression
```

```
From pyspark.ml.classification import LogisticRegression,  
RandomForestClassifier
```

Pandas

```
import pandas as pd
```

```
df = pd.read_csv(filepath, header=)
```

```
pd.to_csv(' ')
```

```
df.head()
```

```
df.sample(5)
```

```
df.tail()
```

```
df.describe()
```

```
df.columns
```

```
df.shape
```

```
df.dtypes
```

```
df['col name']
```

PySpark

```
import pyspark
```

```
from pyspark.sql import SparkSession  
spark = SparkSession.builder.getOrCreate()
```

```
df = spark.read.csv(filepath, header=)
```

```
df.write.format('csv').save(' ')
```

```
df.show()
```

```
df.head()
```

```
df.take()
```

```
df.limit()
```

```
df.collect() ← Be careful!
```

```
df.describe().show()
```

```
df.columns
```

```
len(df.columns) → #columns
```

```
df.count() → #rows
```

```
df.dtypes
```

```
df.printSchema()
```

```
df.select('col name')
```


Pandas

`df['new col'] = value`

`df.drop('col name')`

`df.rename({'old': 'new'})`

`df [df ['Age'] > 35]`

`df.toPandas (sparkdf)`

PySpark

`df.withColumn ('newcol', value)`

`df.drop('col name')`

`df.withColumnRename ('old name',
'new name')`

`df [df ['Age'] > 35]`

`df.filter ('Age > 35')`

`*df.where ('Age < 30')`

`spark.createDataFrame (pd_df)`

-groupBy

df	Age	Sex	weight
1	18	M	72
2	22	F	61
3	22	F	57
4	24	F	63
5	18	M	78
6	18	F	60
	:	:	:

df.groupBy('Age').count() | mean('weight')

Age	count	mean(w)	min(w)
18	28	60	42
21	13	61	43
22	11	63	48
24	18	67	49
	:		

↑ ascending (کوچک به بزرگ) / افزایش
 ↓ descending (بزرگ به کوچک) / کاهش

- JOINS

محل	محل	کفیت	قیمت خریدی	بزرگ کد	خواب	۱- حنظل	سوسن	کاربر
۱۴۳۲	سدیا							
۱۴۳۱	سدیا							
	storage							

storage

↳ inefficient

↳ inefficient

user_id	id	name	...	p_id	id	name	B_id	user id	# P	id P
1	1	Abdul		1	1	iphone	1	742	3	.
2				2			2	721	8	
⋮				⋮			⋮	3	1	

Abdul Kabir
Abdul Majeed
Abdul Fawaz

Diagram illustrating two separate data structures:

- Structure 1 (Left):

id1 df1	
a	x
b	y
c	z
- Structure 2 (Right):

id2 df2	
b	A
c	B
d	C

Inner

-Otter

Left Outlier

✓ Right Otter

id

b	y	B
c	z	C

id

a	x	M
b	y	A
c	z	B
d	M	C

a	x	Null
b	y	A
c	z	B

A	y	A
C	z	B
d	-	C

Joins in PySpark