In [ ]:

```
# PYSPARK - Python
# By : AMIT KUMAR SINGH
# Email : singhamit8467@gmail.com
```

In [1]:

```
pip install pyspark
```

```
Defaulting to user installation because normal site-packages is not writeable
Looking in links: /usr/share/pip-wheels
Requirement already satisfied: pyspark in ./.local/lib/python3.11/site-packages
(3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in ./.local/lib/python3.11/site-pac
kages (from pyspark) (0.10.9.7)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import pyspark
from pyspark import SparkContext
```

In [3]:

```
sc=SparkContext()
```

```
24/06/26 13:38:30 WARN Utils: Your hostname, blue-nbjupyterhub6 resolves to a loo
pback address: 127.0.0.1; using 10.0.0.48 instead (on interface ens5)
24/06/26 13:38:30 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another a
ddress
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel
(newLevel).
24/06/26 13:38:31 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
```

In [4]:

```
Input=sc.textFile('people.txt')
```

In [5]:

```
Input
```

Out[5]:

```
people.txt MapPartitionsRDD[1] at textFile at NativeMethodAccessorImpl.java:0
```

In [6]:

```
Input.top(2)
```

Out[6]:

```
['Michael, 29', 'Justin, 19']
```

In [7]:

```
from pyspark.sql import SparkSession
from pyspark.sql.types import *
```

In [8]:

```
spark=SparkSession.builder.config("spark.com.config.option","some-value").getOrC
```

In [9]:

```
df=spark.read.json('people.json')
```

In [10]:

```
df.show()
```

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

In [11]:
```
df.printSchema()
```

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

In [12]:
```
df.select("name").show()
```

```
+-------+
|   name|
+-------+
|Michael|
|   Andy|
| Justin|
+-------+
```

In [13]:
```
df.select(df['name']=='Michael',df['age']).show()
```

```
+----------------+----+
|(name = Michael)| age|
+----------------+----+
|            true|NULL|
|           false|  30|
|           false|  19|
+----------------+----+
```

In [14]:
```
df.select(df['name']=='Justin',df['age']).show()
```

```
+---------------+----+
|(name = Justin)| age|
+---------------+----+
|          false|NULL|
|          false|  30|
|           true|  19|
+---------------+----+
```

In [25]:
```
df.createOrReplaceTempView("people")
```

In [27]:
```
sqlDF=spark.sql('select * from people')
```

In [29]:
```
sqlDF.show()
```

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

In [34]:
```
sqlDF
```

Out[34]:
```
DataFrame[age: bigint, name: string]
```

In [36]:
```
x=spark.sql('select * from people where age>20')
```

In [38]:
```
x
```

Out[38]:
```
DataFrame[age: bigint, name: string]
```

In [40]:
```
x.show()
```

```
+---+----+
|age|name|
+---+----+
| 30|Andy|
+---+----+
```

In [42]:
```
x=spark.sql('select Distinct(name) from people')
x
```

Out[42]:
```
DataFrame[name: string]
```

In [44]:
```
    x.show()
```

```
+-------+
|   name|
+-------+
|Michael|
|   Andy|
| Justin|
+-------+
```

In [48]:
```
    y=x.columns
```

In [52]:
```
    print(y)
```

```
['name']
```

In [66]:
```
    sqlDF.show()
```

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

In [76]:

```
sqlDF.limit(2).show()
```

```
+----+-------+
| age|   name|
+----+-------+
|NULL|Michael|
|  30|   Andy|
+----+-------+
```

In [ ]:

```
```