Create Spark session

In [1]:

```
from pyspark.sql import SparkSession

spark = SparkSession \
    .builder \
    .master("local[*]")\
    .appName("Python Spark SQL basic example") \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()
# local[*] means pseudo mode with all available CPU cores
# You can use spark://IP-address , the URL you find from Spark web ui
# to enable cluster mode, such as spark://JIAYU1AB6.localdomain:7077
# Make sure you shutdown and restart this notebook when switch modes
```

```
21/09/09 12:34:14 WARN Utils: Your hostname, pyspark resolves to a loopback address: 127.
0.1.1; using 10.0.2.15 instead (on interface enp0s3)
21/09/09 12:34:14 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/pypsar
k/spark-3.0.3-bin-hadoop3.2/jars/spark-unsafe_2.12-3.0.3.jar) to constructor java.nio.Dire
ctByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Plat
form
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access
operations
WARNING: All illegal access operations will be denied in a future release
21/09/09 12:34:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
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 setLogLevel(newLeve
l).
```

Create DataFrames

In [2]:

```
# spark is an existing SparkSession
df = spark.read.json("spark-3.0.3-bin-hadoop3.2/examples/src/main/resources/people.json")
# Displays the content of the DataFrame to stdout
df.show()
# +----+-------+
# | age|   name|
# +----+-------+
# |null|Michael|
# |  30|   Andy|
# |  19| Justin|
# +----+-------+
```

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

Untyped Dataset Operations (aka DataFrame Operations)

In [3]:

```
# spark, df are from the previous example
# Print the schema in a tree format
df.printSchema()
# root
# |-- age: long (nullable = true)
# |-- name: string (nullable = true)

# Select only the "name" column
df.select("name").show()
# +-------+
# |   name|
# +-------+
# |Michael|
# |   Andy|
# | Justin|
# +-------+

# Select everybody, but increment the age by 1
df.select(df['name'], df['age'] + 1).show()
# +-------+---------+
# |   name|(age + 1)|
# +-------+---------+
# |Michael|     null|
# |   Andy|       31|
# | Justin|       20|
# +-------+---------+

# Select people older than 21
df.filter(df['age'] > 21).show()
# +---+----+
# |age|name|
# +---+----+
# | 30|Andy|
# +---+----+

# Count people by age
df.groupBy("age").count().show()
# +----+-----+
# | age|count|
# +----+-----+
# |  19|    1|
# |null|    1|
# |  30|    1|
# +----+-----+
```

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

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

+-------+---------+
|   name|(age + 1)|
+-------+---------+
```

```
|Michael|     null|
|   Andy|       31|
| Justin|       20|
+-------+---------+

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


+----+-----+
| age|count|
+----+-----+
|  19|    1|
|null|    1|
|  30|    1|
+----+-----+
```

Running SQL Queries Programmatically

In [4]:

```
# Register the DataFrame as a SQL temporary view
df.createOrReplaceTempView("people")

sqlDF = spark.sql("SELECT * FROM people")
sqlDF.show()
# +----+-------+
# | age|   name|
# +----+-------+
# |null|Michael|
# |  30|   Andy|
# |  19| Justin|
# +----+-------+
```

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

Convert Spark DataFrame to Pandas DataFrame

In [5]:

```
pandasDf = sqlDF.toPandas()
print(pandasDf)
```

```
    age     name
0   NaN  Michael
1  30.0     Andy
2  19.0   Justin
```

In [ ]:

In [ ]:

In [ ]: