Crear el SparkSession y el SparkContext

:
```
from pyspark.sql import SparkSession

spark = SparkSession.builder\
        .master("local[*]")\
        .appName('PySpark_training')\
        .getOrCreate()
```

:
```
spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext
```

```python
from pyspark.sql.types import StructField,StringType,IntegerType,StructType
```

```python
data_schema = [StructField("age", IntegerType(), True),StructField("name", StringType(), True)]
```

```python
final_struc = StructType(fields=data_schema)
```

```python
df = spark.read.json('people.json', schema=final_struc)
```

```python
df.printSchema()
```

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

C: > LuisAlex > CURSO_BigData > Lec2 > {} people.json > ...

```
1  {"name":"Michael"}
2  {"name":"Andy", "age":30}
3  {"name":"Justin", "age":19}
4
```

```
df.show()
```

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

```
df.printSchema()
```

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

```
df.columns
```

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

```
df.describe()
```

```
DataFrame[summary: string, age: string, name: string]
```

```
df.select('age').show()
```

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

```
df.select(['age','name']).show()
```

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

Creating new columns

```
# Adding a new column with a simple copy
df.withColumn('newage',df['age']).show()
```

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

```
df.show()
```

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

```
# Simple Rename
df.withColumnRenamed('age','supernewage').show()
```

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

```
df.withColumn('doubleage',df['age']*2).show()
```

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

```
# Using SQL with .select()
df.filter("Close<500").select('Open').show()
```

```
+------------------+
|              Open|
+------------------+
|         213.429998|
|         214.599998|
|         214.379993|
|             211.75|
|         210.299994|
|212.79999700000002|
|208.48999900000002|
```

```
# Mean
df.groupBy("Company").mean().show()
```

```
+-------+------------------+
|Company|       avg(Sales)|
+-------+------------------+
|   APPL|             370.0|
|   GOOG|             220.0|
|     FB|             610.0|
|   MSFT|322.3333333333333|
+-------+------------------+
```

```
# Sum
df.groupBy("Company").sum().show()
```

```
+-------+----------+
|Company|sum(Sales)|
+-------+----------+
|   APPL|    1480.0|
|   GOOG|     660.0|
|     FB|    1220.0|
|   MSFT|     967.0|
+-------+----------+
```

```
from pyspark.sql.functions import countDistinct, avg,stddev
```

```
df.select(countDistinct("Sales")).show()
```

```
+----------------------+
|count(DISTINCT Sales)|
+----------------------+
|                    11|
+----------------------+
```

Often you will want to change the name, use the .alias() method for this:

```
df.select(countDistinct("Sales").alias("Distinct Sales")).show()
```

```
+--------------+
|Distinct Sales|
+--------------+
|            11|
+--------------+
```

```
df.select(avg('Sales')).show()
```

```
+------------------+
|        avg(Sales)|
+------------------+
|360.58333333333333|
+------------------+
```

```
: # OrderBy
  # Ascending
  df.orderBy("Sales").show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|   GOOG|Charlie|120.0|
|   MSFT|    Amy|124.0|
|   APPL|  Linda|130.0|
|   GOOG|    Sam|200.0|
|   MSFT|Vanessa|243.0|
|   APPL|   John|250.0|
|   GOOG|  Frank|340.0|
|     FB|  Sarah|350.0|
|   APPL|  Chris|350.0|
|   MSFT|   Tina|600.0|
|   APPL|   Mike|750.0|
|     FB|   Carl|870.0|
+-------+-------+-----+
```

```
# Descending call off the column itself.
df.orderBy(df["Sales"].desc()).show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|     FB|   Carl|870.0|
|   APPL|   Mike|750.0|
|   MSFT|   Tina|600.0|
|     FB|  Sarah|350.0|
|   APPL|  Chris|350.0|
|   GOOG|  Frank|340.0|
|   APPL|   John|250.0|
|   MSFT|Vanessa|243.0|
|   GOOG|    Sam|200.0|
|   APPL|  Linda|130.0|
|   MSFT|    Amy|124.0|
|   GOOG|Charlie|120.0|
+-------+-------+-----+
```

```
emp = [(1, "AAA", "dept1", 1000),
    (2, "BBB", "dept1", 1100),
    (3, "CCC", "dept1", 3000),
    (4, "DDD", "dept1", 1500),
    (5, "EEE", "dept2", 8000),
    (6, "FFF", "dept2", 7200),
    (7, "GGG", "dept3", 7100),
    (8, "HHH", "dept3", 3700),
    (9, "III", "dept3", 4500),
    (10, "JJJ", "dept5", 3400)]

dept = [("dept1", "Department - 1"),
        ("dept2", "Department - 2"),
        ("dept3", "Department - 3"),
        ("dept4", "Department - 4")

       ]

df = spark.createDataFrame(emp, ["id", "name", "dept", "salary"])

deptdf = spark.createDataFrame(dept, ["id", "name"])
```

```
df.show()
```

```
+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
|  2| BBB|dept1|  1100|
|  3| CCC|dept1|  3000|
|  4| DDD|dept1|  1500|
|  5| EEE|dept2|  8000|
|  6| FFF|dept2|  7200|
|  7| GGG|dept3|  7100|
|  8| HHH|dept3|  3700|
|  9| III|dept3|  4500|
| 10| JJJ|dept5|  3400|
+---+----+-----+------+
```

```
df.count()
```

```
10
```

```
df.columns
```

```
['id', 'name', 'dept', 'salary']
```

:

```
df.select("id", "name").show()
```

```
+---+----+
| id|name|
+---+----+
|  1| AAA|
|  2| BBB|
|  3| CCC|
|  4| DDD|
|  5| EEE|
|  6| FFF|
|  7| GGG|
|  8| HHH|
|  9| III|
| 10| JJJ|
+---+----+
```

```
df.filter(df["id"] == 1).show()
df.filter(df.id == 1).show()
```

```
+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
+---+----+-----+------+

+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
+---+----+-----+------+
```

```
df.filter(col("id") == 1).show()
df.filter("id = 1").show()
```

```
+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
+---+----+-----+------+

+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
+---+----+-----+------+
```

```
newdf = df.drop("id")
newdf.show(2)
```

```
+----+-----+------+
|name| dept|salary|
+----+-----+------+
| AAA|dept1|  1000|
| BBB|dept1|  1100|
+----+-----+------+
only showing top 2 rows
```

```
df.groupBy("dept").agg(
        count("salary").alias("count"),
        sum("salary").alias("sum"),
        max("salary").alias("max"),
        min("salary").alias("min"),
        avg("salary").alias("avg")
        ).show()
```

```
+-----+-----+-----+---+---+------+
| dept|count|  sum|max|min|   avg|
+-----+-----+-----+---+---+------+
|dept1|    4| 6600|3000|1000|1650.0|
|dept2|    2|15200|8000|7200|7600.0|
|dept5|    1| 3400|3400|3400|3400.0|
|dept3|    3|15300|7100|3700|5100.0|
+-----+-----+-----+---+---+------+
```

```
df.sort("salary").show(5)
```

```
+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  1| AAA|dept1|  1000|
|  2| BBB|dept1|  1100|
|  4| DDD|dept1|  1500|
|  3| CCC|dept1|  3000|
| 10| JJJ|dept5|  3400|
+---+----+-----+------+
only showing top 5 rows
```

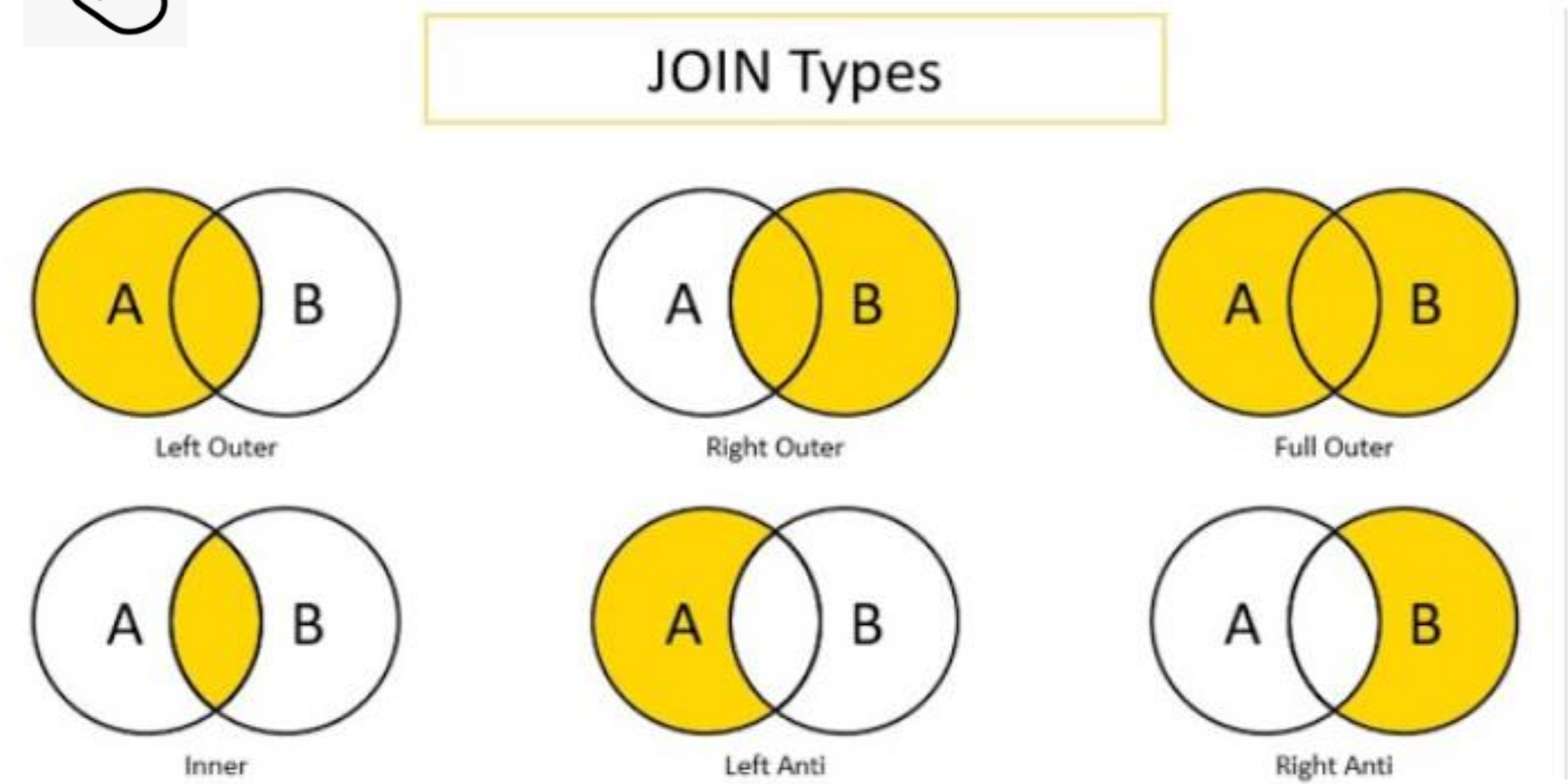
```
# Sort the data in descending order.
df.sort(desc("salary")).show(5)
```

```
+---+----+-----+------+
| id|name| dept|salary|
+---+----+-----+------+
|  5| EEE|dept2|  8000|
|  6| FFF|dept2|  7200|
|  7| GGG|dept3|  7100|
|  9| III|dept3|  4500|
|  8| HHH|dept3|  3700|
+---+----+-----+------+
only showing top 5 rows
```

```
: df.withColumn("bonus", col("salary") * .1).show()
```

```
+---+----+-----+------+-----+
| id|name| dept|salary|bonus|
+---+----+-----+------+-----+
|  1| AAA|dept1|  1000|100.0|
|  2| BBB|dept1|  1100|110.0|
|  3| CCC|dept1|  3000|300.0|
|  4| DDD|dept1|  1500|150.0|
|  5| EEE|dept2|  8000|800.0|
|  6| FFF|dept2|  7200|720.0|
|  7| GGG|dept3|  7100|710.0|
|  8| HHH|dept3|  3700|370.0|
|  9| III|dept3|  4500|450.0|
| 10| JJJ|dept5|  3400|340.0|
+---+----+-----+------+-----+
```

PySpark Join Types | Join Two DataFrames



```
# Inner JOIN.
df.join(deptdf, df["dept"] == deptdf["id"]).show()
```

```
emp = [(1, "AAA", "dept1", 1000),
    (2, "BBB", "dept1", 1100),
    (3, "CCC", "dept1", 3000),
    (4, "DDD", "dept1", 1500),
    (5, "EEE", "dept2", 8000),
    (6, "FFF", "dept2", 7200),
    (7, "GGG", "dept3", 7100),
    (8, "HHH", "dept3", 3700),
    (9, "III", "dept3", 4500),
    (10, "JJJ", "dept5", 3400)]

dept = [("dept1", "Department - 1"),
        ("dept2", "Department - 2"),
        ("dept3", "Department - 3"),
        ("dept4", "Department - 4")

      ]
```

```
+---+----+-----+------+-----+--------------+
| id|name| dept|salary|   id|          name|
+---+----+-----+------+-----+--------------+
|  1| AAA|dept1|  1000|dept1|Department - 1|
|  2| BBB|dept1|  1100|dept1|Department - 1|
|  3| CCC|dept1|  3000|dept1|Department - 1|
|  4| DDD|dept1|  1500|dept1|Department - 1|
|  5| EEE|dept2|  8000|dept2|Department - 2|
|  6| FFF|dept2|  7200|dept2|Department - 2|
|  7| GGG|dept3|  7100|dept3|Department - 3|
|  8| HHH|dept3|  3700|dept3|Department - 3|
|  9| III|dept3|  4500|dept3|Department - 3|
+---+----+-----+------+-----+--------------+
```

```
df = spark.createDataFrame(emp, ["id", "name", "dept", "salary"])

deptdf = spark.createDataFrame(dept, ["id", "name"])
```

Left Outer Join

```
emp = [(1, "AAA", "dept1", 1000),
    (2, "BBB", "dept1", 1100),
    (3, "CCC", "dept1", 3000),
    (4, "DDD", "dept1", 1500),
    (5, "EEE", "dept2", 8000),
    (6, "FFF", "dept2", 7200),
    (7, "GGG", "dept3", 7100),
    (8, "HHH", "dept3", 3700),
    (9, "III", "dept3", 4500),
    (10, "JJJ", "dept5", 3400)]

dept = [("dept1", "Department - 1"),
        ("dept2", "Department - 2"),
        ("dept3", "Department - 3"),
        ("dept4", "Department - 4")

       ]
```

```
: df.join(deptdf, df["dept"] == deptdf["id"], "left_outer").show()
```

```
+---+----+-----+------+-----+--------------+
| id|name| dept|salary|   id|          name|
+---+----+-----+------+-----+--------------+
|  1| AAA|dept1|  1000|dept1|Department - 1|
|  2| BBB|dept1|  1100|dept1|Department - 1|
|  3| CCC|dept1|  3000|dept1|Department - 1|
|  4| DDD|dept1|  1500|dept1|Department - 1|
|  5| EEE|dept2|  8000|dept2|Department - 2|
|  6| FFF|dept2|  7200|dept2|Department - 2|
|  7| GGG|dept3|  7100|dept3|Department - 3|
|  8| HHH|dept3|  3700|dept3|Department - 3|
|  9| III|dept3|  4500|dept3|Department - 3|
| 10| JJJ|dept5|  3400| null|          null|
+---+----+-----+------+-----+--------------+
```

Right Outer Join ¶

```
emp = [(1, "AAA", "dept1", 1000),
    (2, "BBB", "dept1", 1100),
    (3, "CCC", "dept1", 3000),
    (4, "DDD", "dept1", 1500),
    (5, "EEE", "dept2", 8000),
    (6, "FFF", "dept2", 7200),
    (7, "GGG", "dept3", 7100),
    (8, "HHH", "dept3", 3700),
    (9, "III", "dept3", 4500),
    (10, "JJJ", "dept5", 3400)]

dept = [("dept1", "Department - 1"),
        ("dept2", "Department - 2"),
        ("dept3", "Department - 3"),
        ("dept4", "Department - 4")

      ]
```

```
df.join(deptdf, df["dept"] == deptdf["id"], "right_outer").show()
```

```
+----+----+-----+------+-----+-------------+
|  id|name| dept|salary|   id|         name|
+----+----+-----+------+-----+-------------+
|   1| AAA|dept1|  1000|dept1|Department - 1|
|   2| BBB|dept1|  1100|dept1|Department - 1|
|   3| CCC|dept1|  3000|dept1|Department - 1|
|   4| DDD|dept1|  1500|dept1|Department - 1|
|   5| EEE|dept2|  8000|dept2|Department - 2|
|   6| FFF|dept2|  7200|dept2|Department - 2|
|   7| GGG|dept3|  7100|dept3|Department - 3|
|   8| HHH|dept3|  3700|dept3|Department - 3|
|   9| III|dept3|  4500|dept3|Department - 3|
|null|null| null|  null|dept4|Department - 4|
+----+----+-----+------+-----+-------------+
```

Full Outer Join

```
emp = [(1, "AAA", "dept1", 1000),
    (2, "BBB", "dept1", 1100),
    (3, "CCC", "dept1", 3000),
    (4, "DDD", "dept1", 1500),
    (5, "EEE", "dept2", 8000),
    (6, "FFF", "dept2", 7200),
    (7, "GGG", "dept3", 7100),
    (8, "HHH", "dept3", 3700),
    (9, "III", "dept3", 4500),
    (10, "JJJ", "dept5", 3400)]

dept = [("dept1", "Department - 1"),
        ("dept2", "Department - 2"),
        ("dept3", "Department - 3"),
        ("dept4", "Department - 4")

       ]
```

```
df.join(deptdf, df["dept"] == deptdf["id"], "outer").show()
```

```
+----+----+-----+------+-----+-------------+
|  id|name| dept|salary|   id|         name|
+----+----+-----+------+-----+-------------+
|   1| AAA|dept1|  1000|dept1|Department - 1|
|   2| BBB|dept1|  1100|dept1|Department - 1|
|   3| CCC|dept1|  3000|dept1|Department - 1|
|   4| DDD|dept1|  1500|dept1|Department - 1|
|   5| EEE|dept2|  8000|dept2|Department - 2|
|   6| FFF|dept2|  7200|dept2|Department - 2|
|   7| GGG|dept3|  7100|dept3|Department - 3|
|   8| HHH|dept3|  3700|dept3|Department - 3|
|   9| III|dept3|  4500|dept3|Department - 3|
|null|null| null|  null|dept4|Department - 4|
|  10| JJJ|dept5|  3400| null|         null|
+----+----+-----+------+-----+-------------+
```