distinct( ) Transformation

This transformation is used to ensure there are no duplicates in the dataset

In [6]:
```
from pyspark.sql import SparkSession

spark = SparkSession \
        .builder \
        .master("local[*]") \
        .appName("Sample Transformation") \
        .getOrCreate()

#Two lists
list1 = [1, 2, 1, 4, 5, 3, 2, 5, 1, 10]

rdd = spark.sparkContext.parallelize(list1, 4)

#with distinct(), we find the output with not any duplicates.
distinct_rdd = rdd.distinct()

print(distinct_rdd.collect())
```

```
[4, 1, 5, 2, 10, 3]
```

In [2]:
```
#Applying the distinct() transformation on a dataset
#The output prints out only distinct elements.
str_rdd = spark.sparkContext.parallelize(['hi','John','how','are','you','doing','David how','how', 'coping'])


str_rdd = str_rdd.distinct()
print(str_rdd.collect())
```

```
['how', 'you', 'are', 'coping', 'David how', 'hi', 'John', 'doing']
```

In [3]:
```
#Below, we are reading data from a file on the local machine.

input_folder_path = "/Users/vaishaliyasala/Desktop/Github/Spark/Exercise_Dependencies/distinct_file.txt"


file_overview_rdd = spark.sparkContext.textFile(input_folder_path, 4)

print(file_overview_rdd.collect())
```

```
["Recently I had the pleasure of seeing one of William Shakespeare's most beloved comedies, A Midsummer Night's
Dream, performed beautifully at the Los Angeles Repertory Theatre in downtown Los Angeles. At first glance, thi
s performance space looks more like an industrial warehouse than an art house, but walking in you are transform
ed to the magical land of Midsummer."]
```

In [4]:
```
#Use mapPartitions() transformation is applied on each partition of the RDDs.
#A custom function is used to split it into inividual elements


def tokenize(iterator):
    mylist = []
    for words in iterator:
        mylist = words.split(" ")
    return mylist


rdd_new = file_overview_rdd.mapPartitions(tokenize)

#Look at the element count in this RDD.
print("rdd_new = ", rdd_new.collect())
print("number of elements =", len(rdd_new.collect()))
```

```
rdd_new =  ['Recently', 'I', 'had', 'the', 'pleasure', 'of', 'seeing', 'one', 'of', 'William', "Shakespeare's",
'most', 'beloved', 'comedies,', 'A', 'Midsummer', "Night's", 'Dream,', 'performed', 'beautifully', 'at', 'the',
'Los', 'Angeles', 'Repertory', 'Theatre', 'in', 'downtown', 'Los', 'Angeles.', 'At', 'first', 'glance,', 'thi
s', 'performance', 'space', 'looks', 'more', 'like', 'an', 'industrial', 'warehouse', 'than', 'an', 'art', 'hou
se,', 'but', 'walking', 'in', 'you', 'are', 'transformed', 'to', 'the', 'magical', 'land', 'of', 'Midsummer.']
number of elements = 58
```

In [5]:
```
#distinct() transformation gives all the words that are unique to the input RDD.

str_rdd = rdd_new.distinct()
print("distinct_rdd = ", str_rdd.collect())
print("number of elements = ", len(str_rdd.collect()))
```

```
distinct_rdd =  ['of', "Shakespeare's", 'beloved', 'in', 'At', 'glance,', 'performance', 'space', 'like', 'an',
'warehouse', 'but', 'are', 'transformed', 'magical', 'the', 'William', 'most', 'A', 'downtown', 'Angeles.', 'fi
rst', 'industrial', 'art', 'house,', 'you', 'to', 'seeing', 'comedies,', 'Midsummer', "Night's", 'performed',
'beautifully', 'at', 'Angeles', 'this', 'looks', 'more', 'than', 'Recently', 'I', 'had', 'pleasure', 'one', 'Dr
eam,', 'Los', 'Repertory', 'Theatre', 'walking', 'land', 'Midsummer.']
number of elements =  51
```