intersection Transformation

In [6]:

```
from pyspark.sql import SparkSession

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

#Two lists
list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
list2 = [11, 12, 13, 4, 15, 16, 7, 18, 9, 20]

rdd = spark.sparkContext.parallelize(list1, 5)
rdd1 = spark.sparkContext.parallelize(list2, 2)

#with intersection(), we find the intersection of two datasets.
#The output will not contain any duplicates.
intersection_rdd = rdd.intersection(rdd1)

print(intersection_rdd.collect())
```

```
[7, 9, 4]
```

In [7]:

```
#Finding the intersection between two strings
str_rdd_1 = spark.sparkContext.parallelize(['hi','John','how','are','you','doing'])
str_rdd_2 = spark.sparkContext.parallelize(['hi','David','how','are','you','coping'])

str_rdd = str_rdd_1.intersection(str_rdd_2)
print(str_rdd.collect())
```

```
['how', 'you', 'are', 'hi']
```

In [3]:

```
#Below, we are comparing two files read from a folder on the local machine.

input_folder_path = "/Users/vaishaliyasala/Desktop/Github/Spark/Exercise_Dependencies/"

file1_path = input_folder_path + "file.txt"
file2_path = input_folder_path + "tech_overview.txt"

file1_overview_rdd = spark.sparkContext.textFile(file1_path, 4)
file2_overview_rdd = spark.sparkContext.textFile(file2_path, 4)

print(file1_overview_rdd.collect())
print(file2_overview_rdd.collect())
```

```
['Machine learning is an application of artificial intelligence (AI) that', 'provides systems the ability to au
tomatically', 'To learn and improve from experience', 'Without being explicitly programmed.']
['mapPartitionsWithIndex(func) ', 'Similar to mapPartitions, but also provides func ', 'with an integer value r
epresenting the index of ', 'the partition, so func must be of ', 'type (Int, Iterator<T>) => Iterator<U> ', 'w
hen running on an RDD of type T.']
```

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



rdd_new1 = file1_overview_rdd.mapPartitions(tokenize)
rdd_new2 = file2_overview_rdd.mapPartitions(tokenize)

print("rdd_new1 = ", rdd_new1.collect())
print(" ")
print("rdd.new2() = ", rdd_new2.collect())
```

```
rdd_new1 =  ['Machine', 'learning', 'is', 'an', 'application', 'of', 'artificial', 'intelligence', '(AI)', 'tha
t', 'provides', 'systems', 'the', 'ability', 'to', 'automatically', 'To', 'learn', 'and', 'improve', 'from', 'e
xperience', 'Without', 'being', 'explicitly', 'programmed.']

rdd.new2() =  ['Similar', 'to', 'mapPartitions,', 'but', 'also', 'provides', 'func', '', 'with', 'an', 'intege
r', 'value', 'representing', 'the', 'index', 'of', '', 'type', '(Int,', 'Iterator<T>)', '=>', 'Iterator<U>',
'', 'when', 'running', 'on', 'an', 'RDD', 'of', 'type', 'T.']
```

In [5]:

```
#instersection() transformation gives all the words that are a subset of each other in the RDD.

str_rdd = rdd_new1.intersection(rdd_new2)
print(str_rdd.collect())
```

```
['an', 'of', 'provides', 'the', 'to']
```