

## Simple Join in Spark

### Load datasets

Open the pyspark shell and load the datasets:

```
fileA = sc.textFile("input/join1_FileA.txt")
```

Ensure the file content is correct:

```
fileA.collect()
```

should return:

```
Out[:]: [u'able,991', u'about,11', u'burger,15', u'actor,22']
```

Load the second dataset:

```
fileB = sc.textFile("input/join1_FileB.txt")
```

verify:

```
fileB.collect()
```

```
Out[29]:
```

```
[u'Jan-01 able,5',  
 u'Feb-02 about,3',  
 u'Mar-03 about,8 ',  
 u'Apr-04 able,13',  
 u'Feb-22 actor,3',  
 u'Feb-23 burger,5',  
 u'Mar-08 burger,2',  
 u'Dec-15 able,100']
```

# Mapper for fileA

Create a map function for fileA that takes a line, splits it on the comma and turns the count to an integer.

Copy and paste the following function into the pyspark console, and edit the 2 <ENTER\_CODE\_HERE> lines to perform the necessary operations:

```
def split_fileA(line):  
    # split the input line in word and count on the comma  
    <ENTER_CODE_HERE>  
    # turn the count to an integer  
    <ENTER_CODE_HERE>  
    return (word, count)
```

Test the function by defining a test variable:

```
test_line = "able,991"
```

and ensure that:

```
split_fileA(test_line)
```

returns:

```
Out[: ('able', 991)
```

Run the map transformation to the fileA RDD:

```
fileA_data = fileA.map(split_fileA)
```

If the mapper is correct, then the result is:

```
fileA_data.collect()  
Out[: [(u'able', 991), (u'about', 11), (u'burger', 15), (u'actor', 22)]
```

Ensure that the key of each pair is a string ( i.e. delimited by ' ') and the value is an integer.

# Mapper for fileB

The mapper for fileB is more complex to extract:

```
def split_fileB(line):  
    # split the input line into word, date and count_string  
    <ENTER_CODE_HERE>  
    <ENTER_CODE_HERE>  
    return (word, date + " " + count_string)
```

running:

```
fileB_data = fileB.map(split_fileB)
```

and then gathering the output back to the pyspark Driver console:

```
fileB_data.collect()
```

should give the result:

```
Out[ ]:  
[(u'able', u'Jan-01 5'),  
 (u'about', u'Feb-02 3'),  
 (u'about', u'Mar-03 8 '),  
 (u'able', u'Apr-04 13'),  
 (u'actor', u'Feb-22 3'),  
 (u'burger', u'Feb-23 5'),  
 (u'burger', u'Mar-08 2'),  
 (u'able', u'Dec-15 100')]
```

## Run join

The goal is to join the two datasets using the words as keys and print for each word the wordcount for a specific date and then the total output from A.

For each word in fileB, print the date and count from fileB but also the total count from fileA.

Spark implements the join transformation that given a RDD of (K, V) pairs to be joined with another RDD of (K, W) pairs, returns a dataset that contains (K, (V, W)) pairs.

```
fileB_joined_fileA = fileB_data.join(fileA_data)
```

## Verify the result

Inspect the full result with:

```
fileB_joined_fileA.collect()
```

Assert that the result is as expected.