groupByKey( ) Transformation

By definition, When called on a dataset of (K, V) pairs, returns a dataset of (K, Iterable) pairs.

To group the values for each key in the RDD into a single sequence.

In [5]:
```
from pyspark.sql import SparkSession

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

#Create a list of tuples
data = [('david','Alameda', '29'), \
        ('alex','Berkeley','31'), \
        ('john','Sunnyvale','42'), \
        ('michael','Sunnyvale','23'), \
        ('anna','Gilroy','34'),
        ('jennifer','Cupertino','27')]


rdd = spark.sparkContext.parallelize(data)
print(rdd.collect())
```

```
[('david', 'Alameda', '29'), ('alex', 'Berkeley', '31'), ('john', 'Sunnyvale', '42'), ('michael', 'Sunnyvale',
'23'), ('anna', 'Gilroy', '34'), ('jennifer', 'Cupertino', '27')]
```

In [6]:
```
# apply a map() transformation to rdd to create (K, V) pairs
#In this key-value pair, key is the name and the tuple (location, age) is the value

rdd2 = rdd.map(lambda x: (x[0],(x[1],x[2])))
print(rdd2.collect())
```

```
[('david', ('Alameda', '29')), ('alex', ('Berkeley', '31')), ('john', ('Sunnyvale', '42')), ('michael', ('Sunny
vale', '23')), ('anna', ('Gilroy', '34')), ('jennifer', ('Cupertino', '27'))]
```

In [7]:
```
# apply groupByKey() transformation to rdd2 to return a dataset of (K, Iterable<V>) pairs.

rdd3 =rdd2.groupByKey()
print(rdd3.collect())
```

```
[('alex', <pyspark.resultiterable.ResultIterable object at 0x104c783a0>), ('anna', <pyspark.resultiterable.Resu
ltIterable object at 0x106eaba00>), ('jennifer', <pyspark.resultiterable.ResultIterable object at 0x106eab9a0
>), ('john', <pyspark.resultiterable.ResultIterable object at 0x106eab820>), ('michael', <pyspark.resultiterabl
e.ResultIterable object at 0x106eab910>), ('david', <pyspark.resultiterable.ResultIterable object at 0x106eab9d
0>)]
```

In [8]:
```
# RDD.mapValues - Pass each value in the key-value pair RDD through a map function without changing the keys
# mapValues operates on the values only
# this also changes the original RDD's partitioning.

print("rdd3.mapValues().collect() = ", rdd3.mapValues(lambda values: list(values)).collect())
```

```
rdd3.mapValues().collect() =  [('alex', [('Berkeley', '31')]), ('anna', [('Gilroy', '34')]), ('jennifer', [('Cu
pertino', '27')]), ('john', [('Sunnyvale', '42')]), ('michael', [('Sunnyvale', '23')]), ('david', [('Alameda',
'29')])]
```