In [139]:
```
1 df_4.show()
```

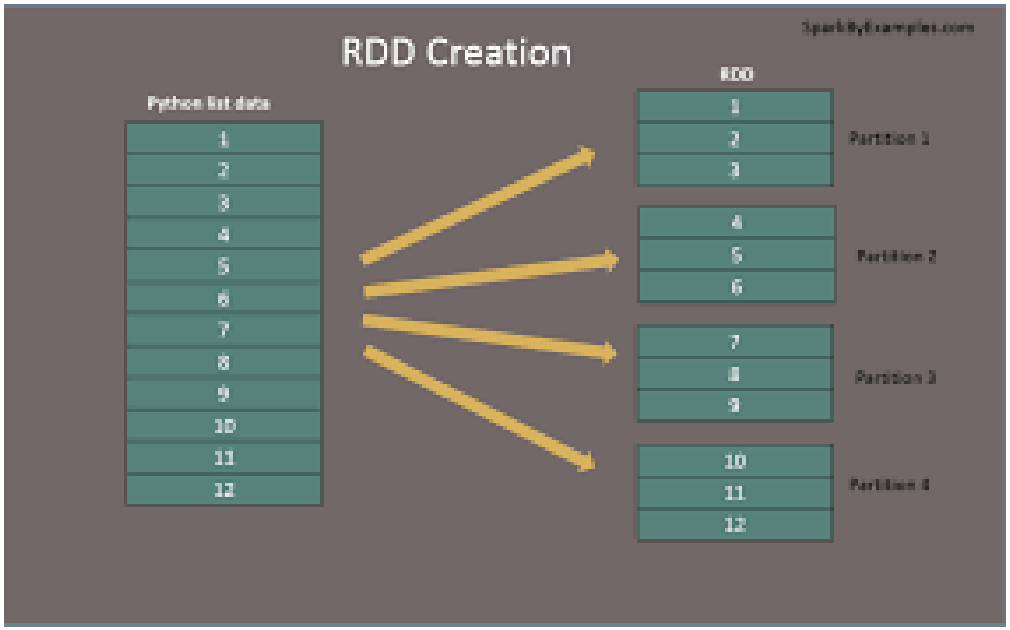
```
+---------+------------+------+
|     Name| Departments|salary|
+---------+------------+------+
|    Krish|Data Science| 10000|
|    Krish|         IOT|  5000|
|   Mahesh|    Big Data|  4000|
|    Krish|    Big Data|  4000|
|   Mahesh|Data Science|  3000|
|Sudhanshu|Data Science| 20000|
|Sudhanshu|         IOT| 10000|
|Sudhanshu|    Big Data|  5000|
|    Sunny|Data Science| 10000|
|    Sunny|    Big Data|  2000|
+---------+------------+------+
```

In [ ]:
```
1
```

RDD

RDD (Resilient Distributed Dataset) is a fundamental building block of Pyspark which is fault tolerant, immutable distributed collection of objects.

Once you create a record in an RDD you cannot change it. Each record in RDD is divided into logical partition which can be computed on different node of cluster.



In [ ]:
```
1
```

In [ ]:
```
1
```

creatting rdd

In [36]:
```
1 data = [23,45,67,86,78,3,4,5,6,10,11,12,23,45,67,10]
2 rdd1 = spark.sparkContext.parallelize(data)
3 print(rdd1)
```

```
ParallelCollectionRDD[28] at readRDDFromFile at PythonRDD.scala:274
```

In [13]:
```
1 type(rdd1)
```

Out[13]:
```
pyspark.rdd.RDD
```

In [38]:
```
1 rdd1.collect()
```

Out[38]: [23, 45, 67, 86, 78, 3, 4, 5, 6, 10, 11, 12, 23, 45, 67, 10]

In [39]:
```
1 rdd1.count()
```

Out[39]: 16

In [18]:
```
1 type(result)
```

Out[18]: list

In [21]:
```
1 for val in rdd1.collect():
2     print(val)
```

```
23
45
67
86
78
3
4
5
6
10
11
12
```

In [23]:
```
1 rdd1.count()
```

Out[23]: 12

In [35]:
```
1 rdd1.distinct().collect()
```

Out[35]: [10, 67, 3, 11, 4, 12, 45, 5, 86, 78, 6, 23]

In [40]:
```
1 rdd1.distinct().count()
```

Out[40]: 12

In [ ]:
```
1
```

In [41]:
```
1 rdd1.collect()
```

Out[41]:
```
[23, 45, 67, 86, 78, 3, 4, 5, 6, 10, 11, 12, 23, 45, 67, 10]
```

In [42]:
```
1 rdd1.filter(lambda x : x<=20).collect()
```

Out[42]:
```
[3, 4, 5, 6, 10, 11, 12, 10]
```

In [43]:
```
1 rdd1.collect()
```

Out[43]:
```
[23, 45, 67, 86, 78, 3, 4, 5, 6, 10, 11, 12, 23, 45, 67, 10]
```

In [ ]:
```
1
```

In [44]:
```
1 rdd1.first()
```

Out[44]:
```
23
```

In [45]:
```
1 rdd1.take(5)
```

Out[45]:
```
[23, 45, 67, 86, 78]
```

In [ ]:
```
1
```

In [46]:
```
1 rdd1.collect()
```

Out[46]:
```
[23, 45, 67, 86, 78, 3, 4, 5, 6, 10, 11, 12, 23, 45, 67, 10]
```

In [47]:
```
1 rdd1.reduce(lambda x,y : x+y)
```

Out[47]:
```
495
```

In [48]:
```
1 rdd1.saveAsTextFile('file.txt')
```

In [ ]:
```
1
```

In [49]:
```
1 rdd2 = spark.sparkContext.parallelize([1,2,3,4,5])
```

In [50]:
```
1 rdd2
```

Out[50]:
```
ParallelCollectionRDD[43] at readRDDFromFile at PythonRDD.scala:274
```

In [51]:
```
1 rdd2.map(lambda x: x**3).collect()
```

Out[51]:
```
[1, 8, 27, 64, 125]
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [52]:
```
1 rdd3 = spark.sparkContext.parallelize([2,4,5,6,7,8,9])
```

In [54]:
```
1 uni1 = rdd3.filter(lambda x : x%2 == 0)
2 uni2 = rdd3.filter(lambda x : x%3 == 0)
```

In [55]:
```
1 uni1.collect()
```

Out[55]:
```
[2, 4, 6, 8]
```

In [56]:
```
1 uni2.collect()
```

Out[56]:
```
[6, 9]
```

In [58]:
```
1 final = uni1.union(uni2)
```

In [59]:
```
1 final.collect()
```

Out[59]:
```
[2, 4, 6, 8, 6, 9]
```

In [60]:
```
from pyspark.mllib.linalg import Matrix,Matrices
```

In [62]:
```
data = [10,20,30,40,50,60]

res = Matrices.dense(3,2,data)
```

In [63]:
```
res
```

Out[63]:
```
DenseMatrix(3, 2, [10.0, 20.0, 30.0, 40.0, 50.0, 60.0], False)
```

In [64]:
```
print(res)
```

```
DenseMatrix([[10., 40.],
             [20., 50.],
             [30., 60.]])
```

In [65]:
```
type(res)
```

Out[65]:
```
pyspark.mllib.linalg.DenseMatrix
```

it returns seperate value for each element in RDD -- flatmap

In [66]:
```
data = spark.sparkContext.parallelize(["Hey There","This is RDD Session in Pyspark"])
```

In [67]:
```
data.collect()
```

Out[67]:
```
['Hey There', 'This is RDD Session in Pyspark']
```

In [68]:
```
data.flatMap(lambda x:x.split(" ")).collect()
```

Out[68]:
```
['Hey', 'There', 'This', 'is', 'RDD', 'Session', 'in', 'Pyspark']
```

In [ ]:
```

```

In [73]:
```
marks = [('Punit',55),('Salam',70),('Dharmesh',80),('Rohan',80),('Amit',55),('Sumit',90),('Punit',60),('
```

In [70]:
```
1 marks
```

Out[70]:
```
[('Punit', 55),
 ('Salam', 70),
 ('Dharmesh', 80),
 ('Rohan', 80),
 ('Amit', 55),
 ('Sumit', 90)]
```

In [74]:
```
1 rdd5 = spark.sparkContext.parallelize(marks)
```

In [75]:
```
1 rdd5.collect()
```

Out[75]:
```
[('Punit', 55),
 ('Salam', 70),
 ('Dharmesh', 80),
 ('Rohan', 80),
 ('Amit', 55),
 ('Sumit', 90),
 ('Punit', 60),
 ('Dharmesh', 70)]
```

reduce by key :

It performs multiple parallel process for each key in the data and combines the value for same key

It uses lambda to perfrom task

In [76]:
```
1 rdd5.reduceByKey(lambda x,y : x+y).collect()
```

Out[76]:
```
[('Punit', 115),
 ('Dharmesh', 150),
 ('Sumit', 90),
 ('Rohan', 80),
 ('Amit', 55),
 ('Salam', 70)]
```

In [ ]:
```
1
```

In [77]:
```
1 rdd5.sortByKey('ascending').collect()
```

Out[77]:
```
[('Amit', 55),
 ('Dharmesh', 80),
 ('Dharmesh', 70),
 ('Punit', 55),
 ('Punit', 60),
 ('Rohan', 80),
 ('Salam', 70),
 ('Sumit', 90)]
```

In [79]:
```
1 rdd5.sortByKey(ascending=False).collect()
```

Out[79]:
```
[('Sumit', 90),
 ('Salam', 70),
 ('Rohan', 80),
 ('Punit', 55),
 ('Punit', 60),
 ('Dharmesh', 80),
 ('Dharmesh', 70),
 ('Amit', 55)]
```

In [81]:
```
1 rdd5.collect()
```

Out[81]:
```
[('Punit', 55),
 ('Salam', 70),
 ('Dharmesh', 80),
 ('Rohan', 80),
 ('Amit', 55),
 ('Sumit', 90),
 ('Punit', 60),
 ('Dharmesh', 70)]
```

In [83]:
```
1 result = rdd5.groupByKey().collect()
```

In [85]:
```
for key,val in result:
    print(key,list(val))
```

```
Punit [55, 60]
Dharmesh [80, 70]
Sumit [90]
Rohan [80]
Amit [55]
Salam [70]
```

In [86]:
```
count = rdd5.countByKey().items()
```

In [87]:
```
for k,v in count:
    print(k,v)
```

```
Punit 2
Salam 1
Dharmesh 2
Rohan 1
Amit 1
Sumit 1
```

In [ ]:
```

```

In [88]:
```
empty = spark.sparkContext.emptyRDD
```

In [91]:
```
empty
```

Out[91]:
```
<bound method SparkContext.emptyRDD of <SparkContext master=local[*] appName=session1>>
```

In [98]:
```
lst = [6,7,8,9,98,87,66,54,33,54]
```

In [99]:
```
rdd6 = spark.sparkContext.parallelize(lst,5)
```

In [101]:
```
rdd6.saveAsTextFile('demo.txt')
```

In [102]:
```
1 rdd6.getNumPartitions()
```

Out[102]: 5

In [103]:
```
1 rdd6.max()
```

Out[103]: 98

In [104]:
```
1 rdd6.min()
```

Out[104]: 6

In [106]:
```
1 rdd6.collect()
```

Out[106]: [6, 7, 8, 9, 98, 87, 66, 54, 33, 54]

In [ ]:
```
1
```

In [ ]:
```
1
```

In [108]:
```
1 rdd5.collect()
```

Out[108]:
```
[('Punit', 55),
 ('Salam', 70),
 ('Dharmesh', 80),
 ('Rohan', 80),
 ('Amit', 55),
 ('Sumit', 90),
 ('Punit', 60),
 ('Dharmesh', 70)]
```

In [109]:
```
1 cols = ["Name","Marks"]
```

In [110]:
```
1 marks_df = rdd5.toDF(cols)
```

In [111]:
```
1 marks_df.show()
```

```
+--------+-----+
|    Name|Marks|
+--------+-----+
|   Punit|   55|
|   Salam|   70|
|Dharmesh|   80|
|   Rohan|   80|
|    Amit|   55|
|   Sumit|   90|
|   Punit|   60|
|Dharmesh|   70|
+--------+-----+
```

In [112]:
```
1 col = ["Data"]
```

In [114]:
```
1 from pyspark.sql import Row
```

In [115]:
```
1 row = Row("Data")
```

In [117]:
```
1 res = rdd6.map(row).toDF()
```

In [118]:
```
1 res.show()
```

```
+----+
|Data|
+----+
|   6|
|   7|
|   8|
|   9|
|  98|
|  87|
|  66|
|  54|
|  33|
|  54|
+----+
```

In [ ]:
```
1
```