**Assignment 13**

**Name:** Jeevan Sai Badana

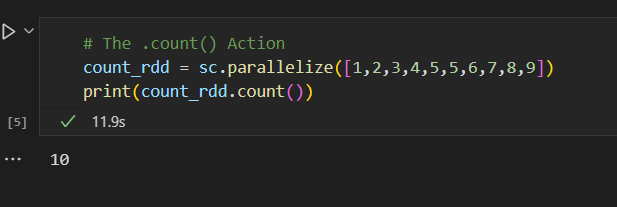
**Mail:** [jeevansai100@gmail.com](mailto:jeevansai100@gmail.com)

**Topic:** RDD Actions and renaming columns and running notes

**Actions in PySpark RDDs**

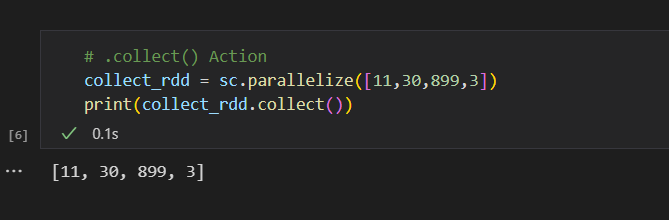
1.The .count() Action

* The **.count()** action on an RDD is an operation that returns the number of elements of our RDD



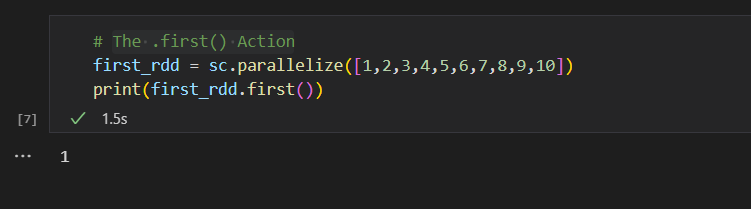
2. .collect() Action

* The .collect() action on an RDD returns a list of all the elements of the RDD



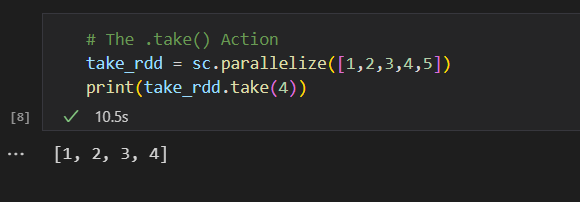
3. The .first() Action

* The .first() action on an RDD returns the first element from our RDD.



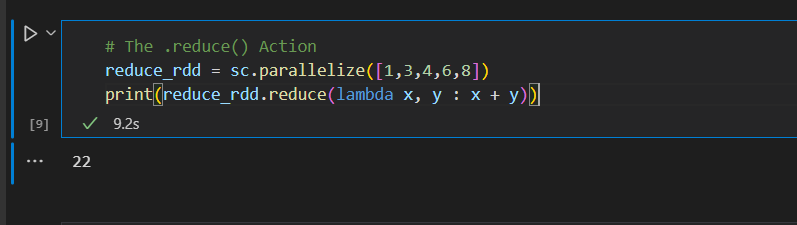
4. The .take() Action

* The .take(n) action on an RDD returns n number of elements from the RDD.



5. The .reduce() Action

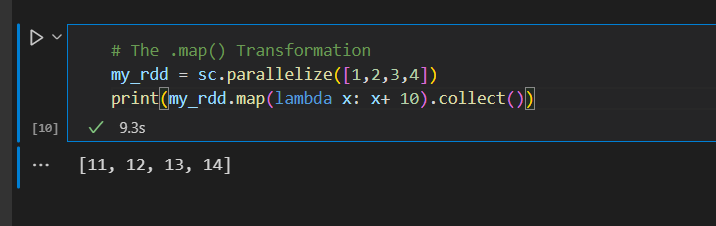
* The .reduce() Actiontakes two elements from the given RDD and operates.



**Transformations in PySpark RDDs**

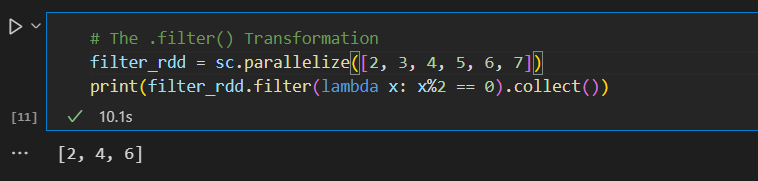
**1.The .map() Transformation**

As the name suggests, the .map() transformation maps a value to the elements of an RDD. The .map() transformation takes in an anonymous function and applies this function to each of the elements in the RDD.



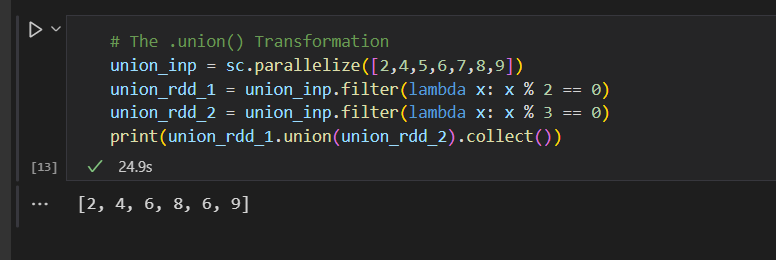
2. The .filter() Transformation

* A .filter() transformation is an operation in PySpark for filtering elements from a PySpark RDD. The .filter() transformation takes in an anonymous function with a condition.



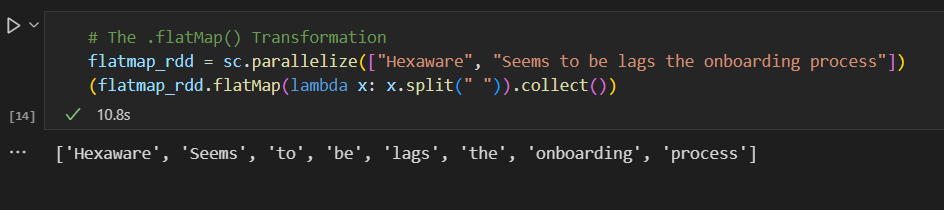
3. The .union() Transformation

* The .union() transformation combines two RDDs and returns the union of the input two RDDs.

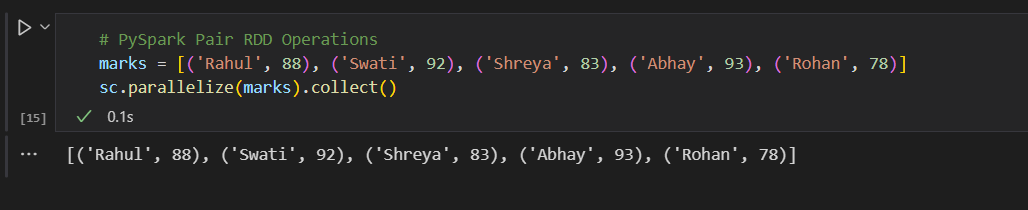


4. The .flatMap() Transformation

The .flatMap() transformation peforms same as the .map() transformation except the fact that .flatMap() transformation return seperate values for each element from original RDD.



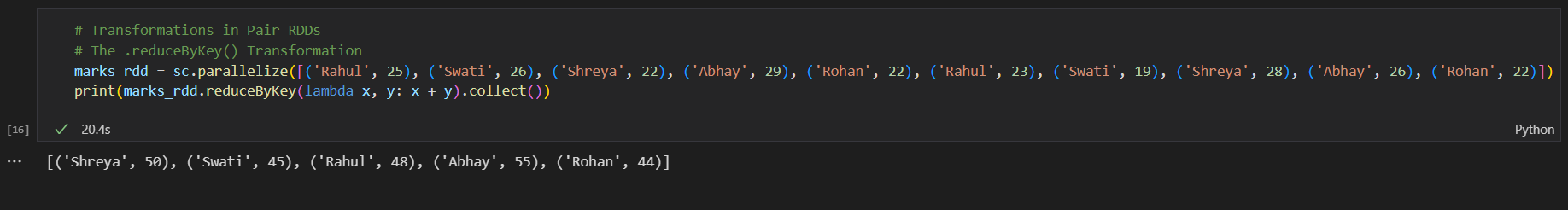
**PySpark Pair RDD Operations**



**Transformations in Pair RDDs**

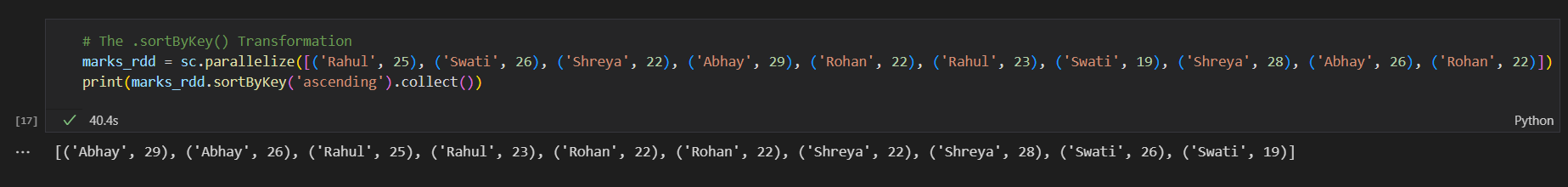
1.The .reduceByKey() Transformation

The .reduceByKey() transformation performs multiple parallel processes for each key in the data and combines the values for the same keys.



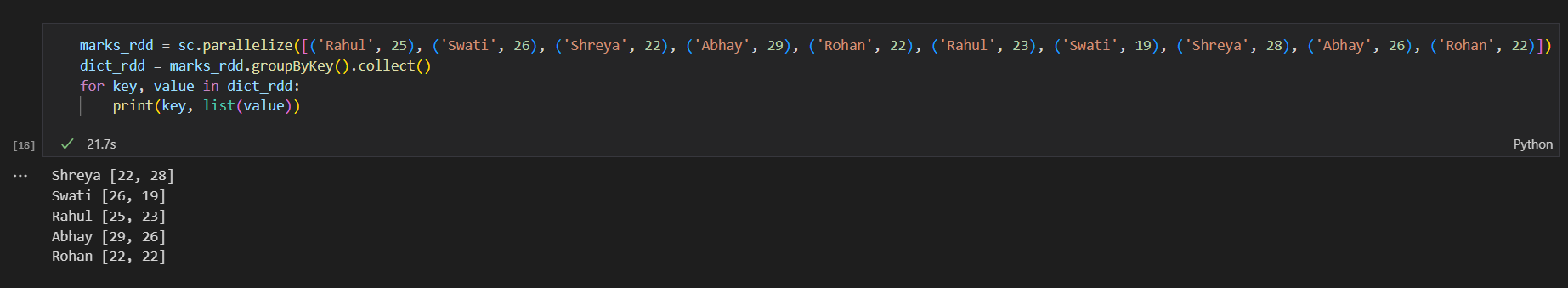
2. The .sortByKey() Transformation

The .sortByKey() transformation sorts the input data by keys from key-value pairs either in ascending or descending order



3. The .groupByKey() Transformation

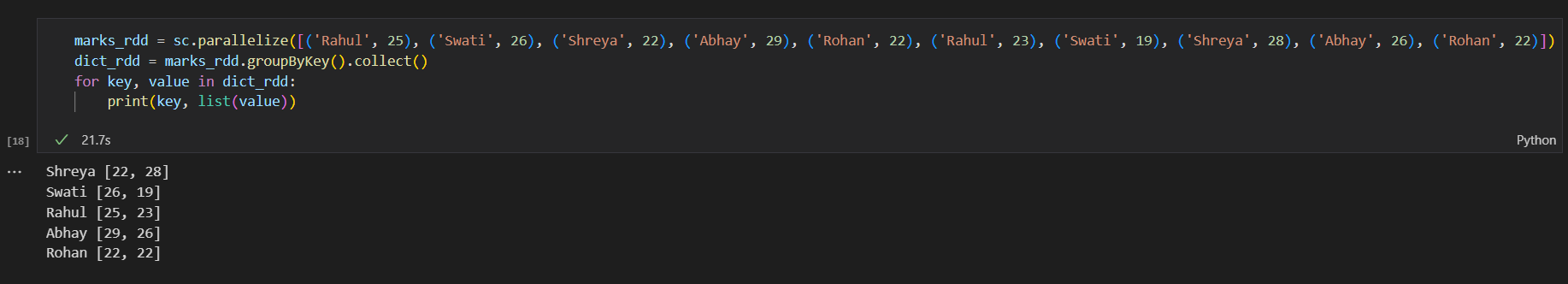
The .groupByKey() transformation groups all the values in the given data with the same key together. It returns a new RDD as a result.



**Actions in Pair RDDs**

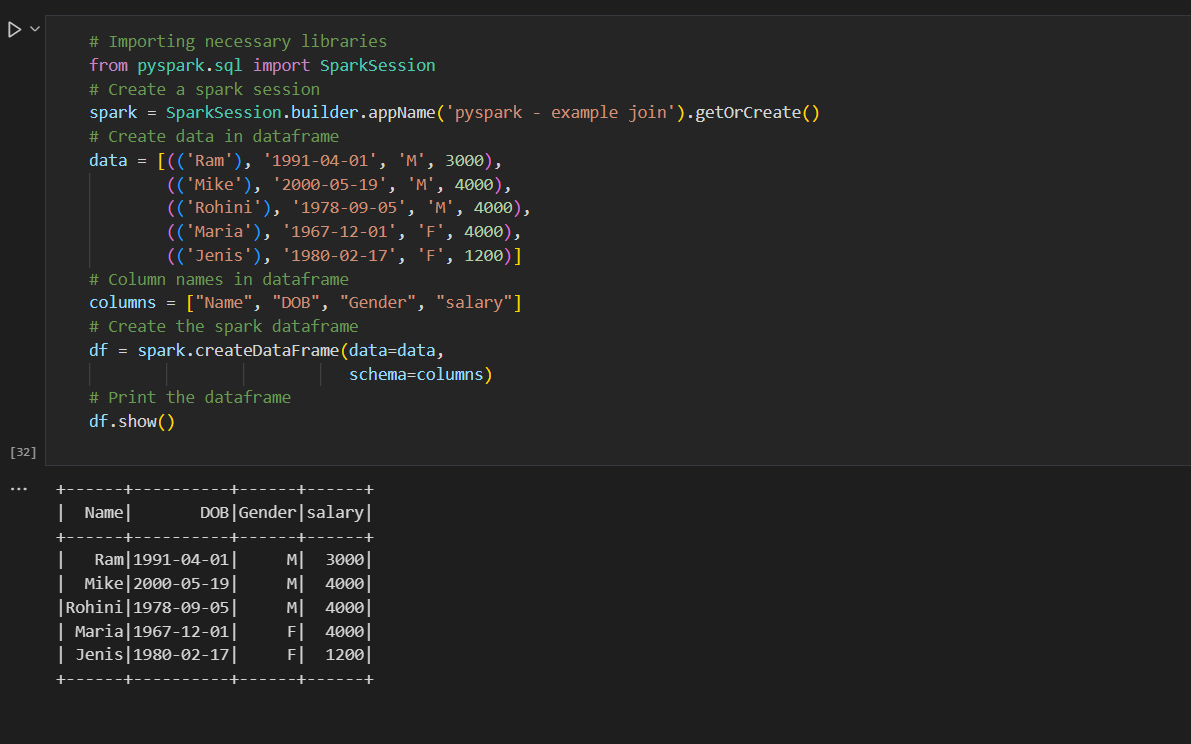
**1.**The countByKey() Action

The .countByKey() option is used to count the number of values for each key in the given data. This action returns a dictionary and one can extract the keys and values by iterating over the extracted dictionary using loops

****

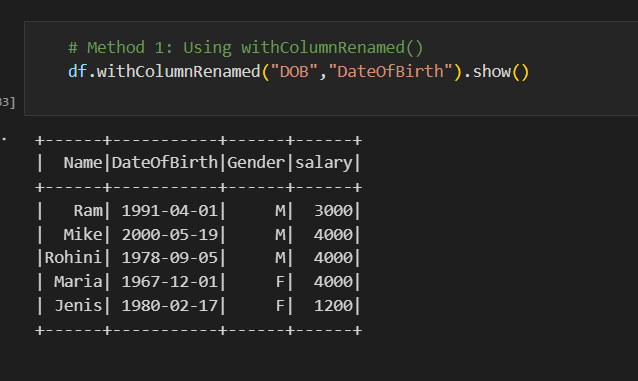
**Renaming column names**

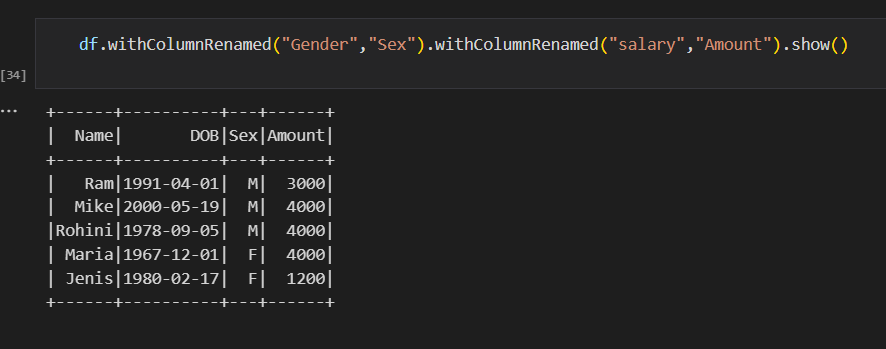
* First we create the following the data frame as below

****

**Method 1: Using withColumnRenamed()**

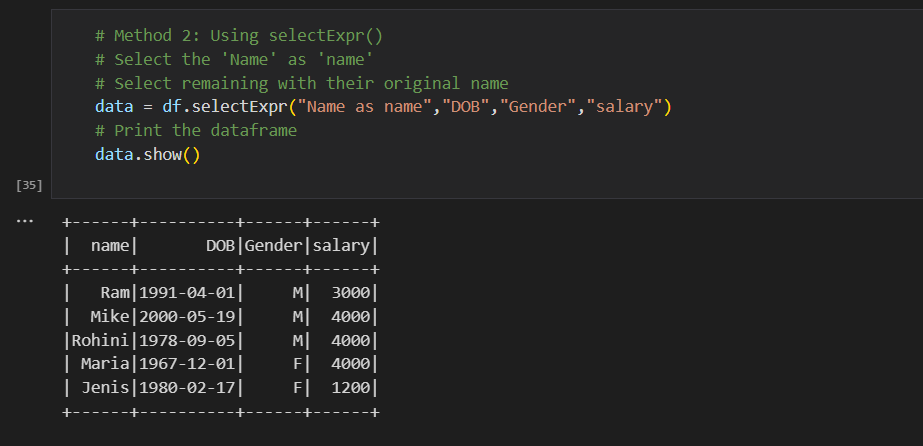
Syntax: DataFrame.withColumnRenamed(existing, new)





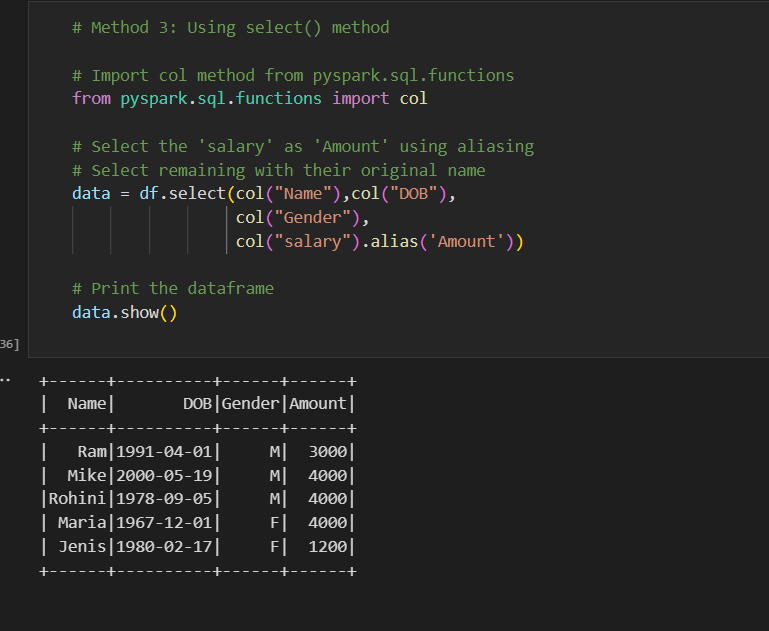
**Method 2: Using selectExpr()**

Syntax : DataFrame.selectExpr(expr)



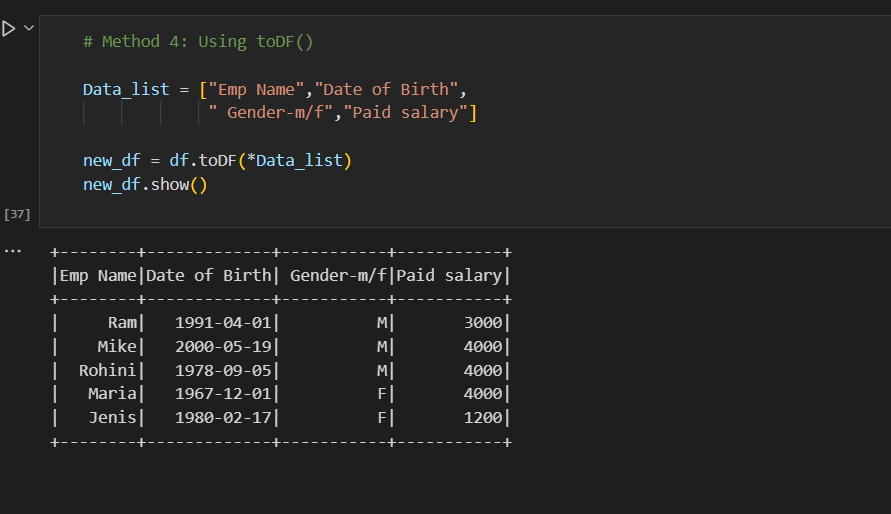
**Method 3: Using select() method**

Syntax: DataFrame.select(cols)

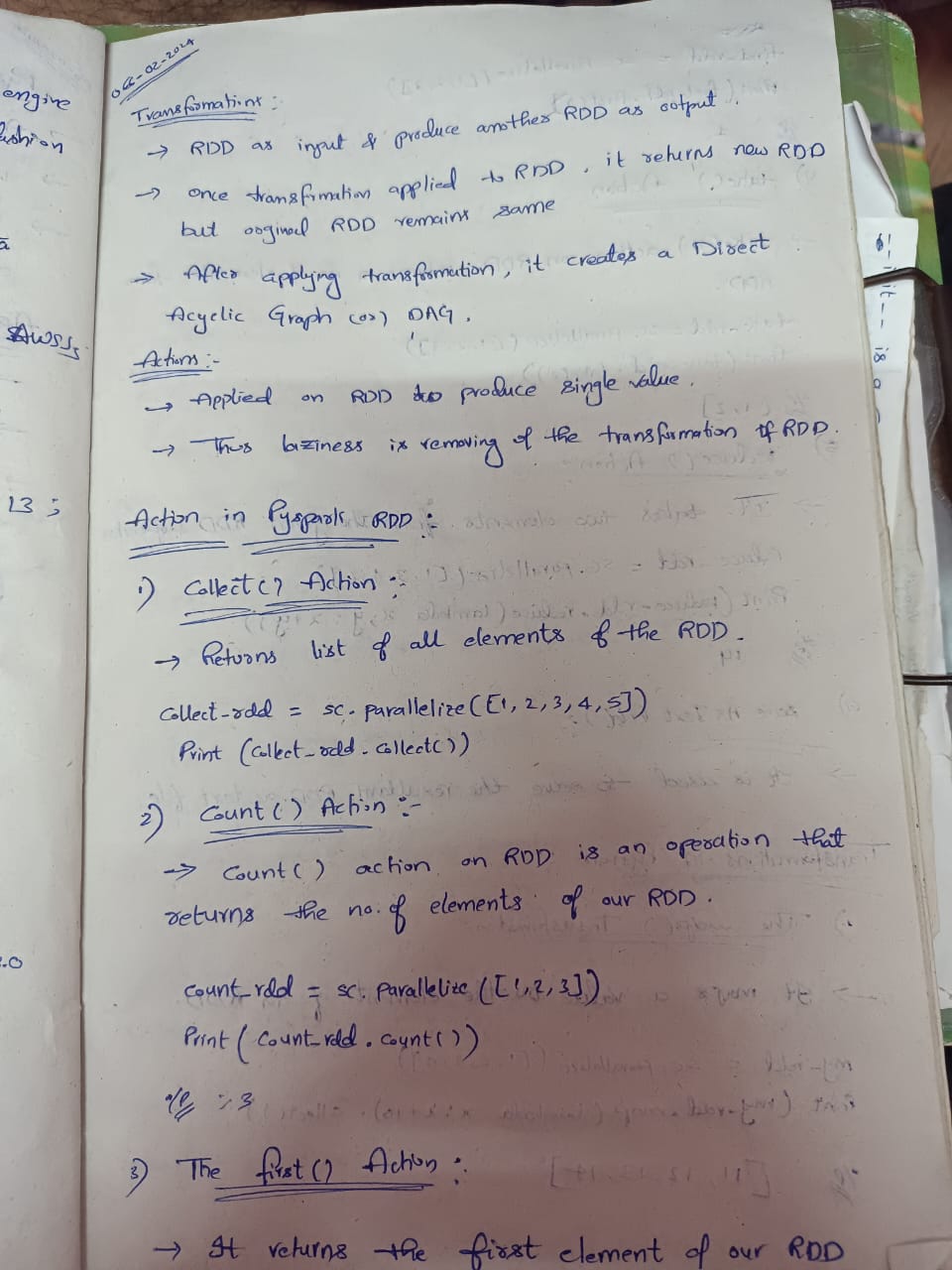


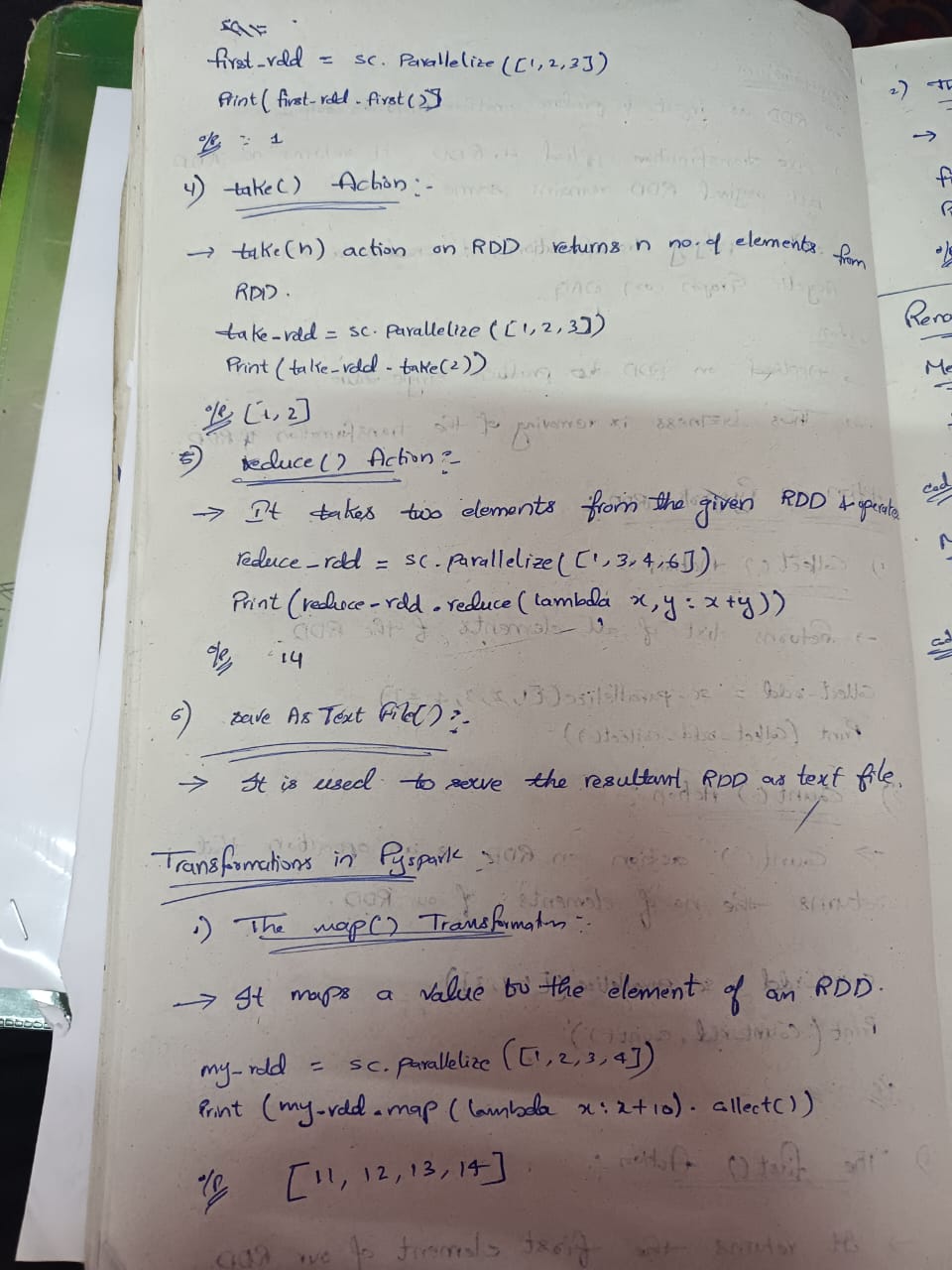
**Method 4: Using toDF()**

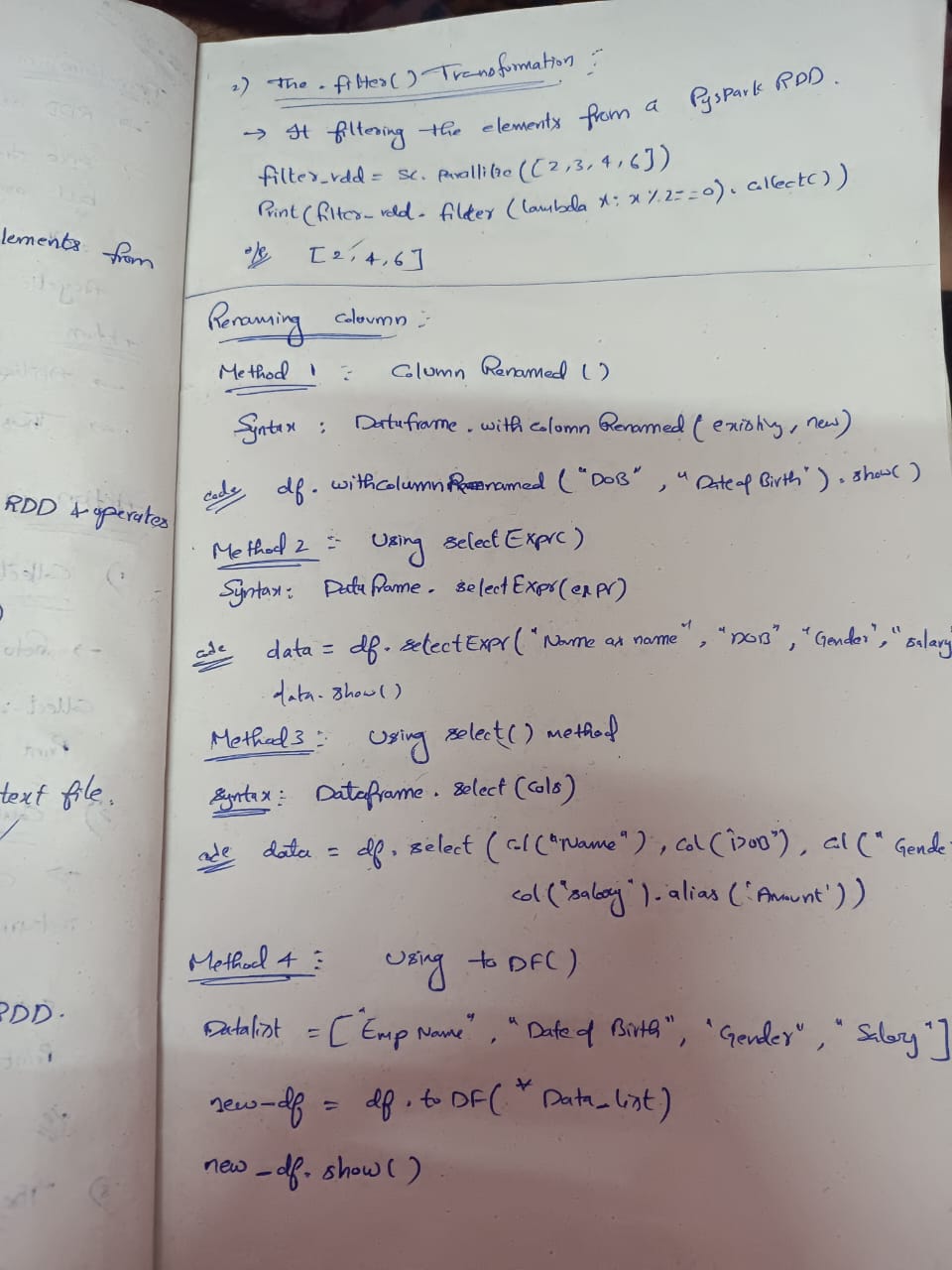
Syntax: toDF(\*col)



**Running notes**

****

****

****