**Spark RDD Actions**

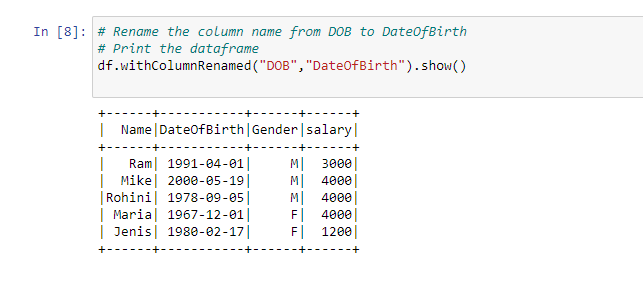
1. .collect(): This function returns all the elements of a distributed dataset (RDD) as a list in the driver program. Be cautious when using it with large datasets as it brings all data to the driver, which can cause memory issues.
2. .count(): Returns the number of elements in the RDD.
3. .first(): Returns the first element in the RDD. It's equivalent to .take(1)[0], but more efficient as it only retrieves the first element.
4. .take(n): Returns an array with the first n elements of the RDD. It's often used for previewing data or fetching a small subset for further analysis.
5. .reduce(): This function is an action that aggregates the elements of the RDD using a specified function. It takes a function that operates on two elements of the type in the RDD and returns a new element of the same type. It repeatedly applies this function to pairs of elements until there is only one element left. This function should be associative and commutative, as it's used in parallel computation across partitions.



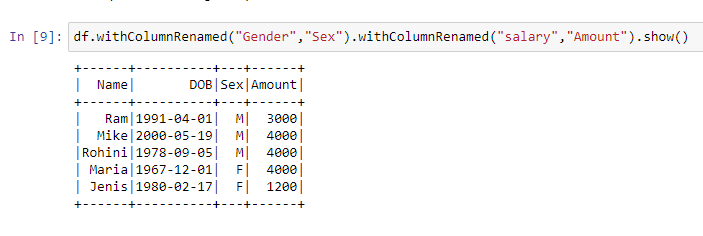
Selecting, Renaming, Filtering Data in a Pandas DataFrame

#### Method 1: Using withColumnRenamed()

We will use of withColumnRenamed() method to change the column names of pyspark data frame.



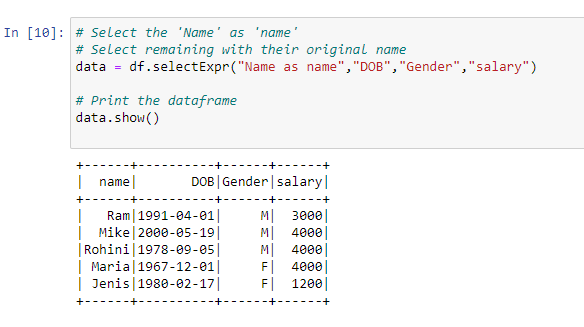
#### Example 2: Renaming multiple column names



#### Method 2: Using selectExpr()

Renaming the column names using selectExpr() method

Syntax : DataFrame.selectExpr(expr)



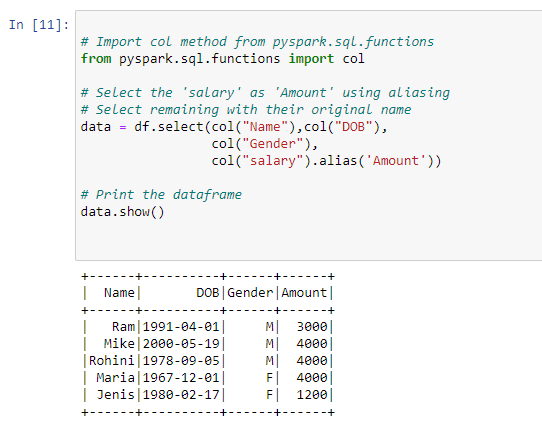
**Method 3: Using select() method**

***Syntax:****DataFrame.select(cols)*

***cols:****List of column names as strings.*

***Return type:****Selects the cols in the dataframe and returns a new DataFrame.*

Here we Rename the column name ‘salary’ to ‘Amount’



**Method 4: Using toDF()**

This function returns a new DataFrame that with new specified column names.

***Syntax:****toDF(\*col)*

*Where, col is a new column name*

In this example, we will create an order list of new column names and pass it into toDF function

