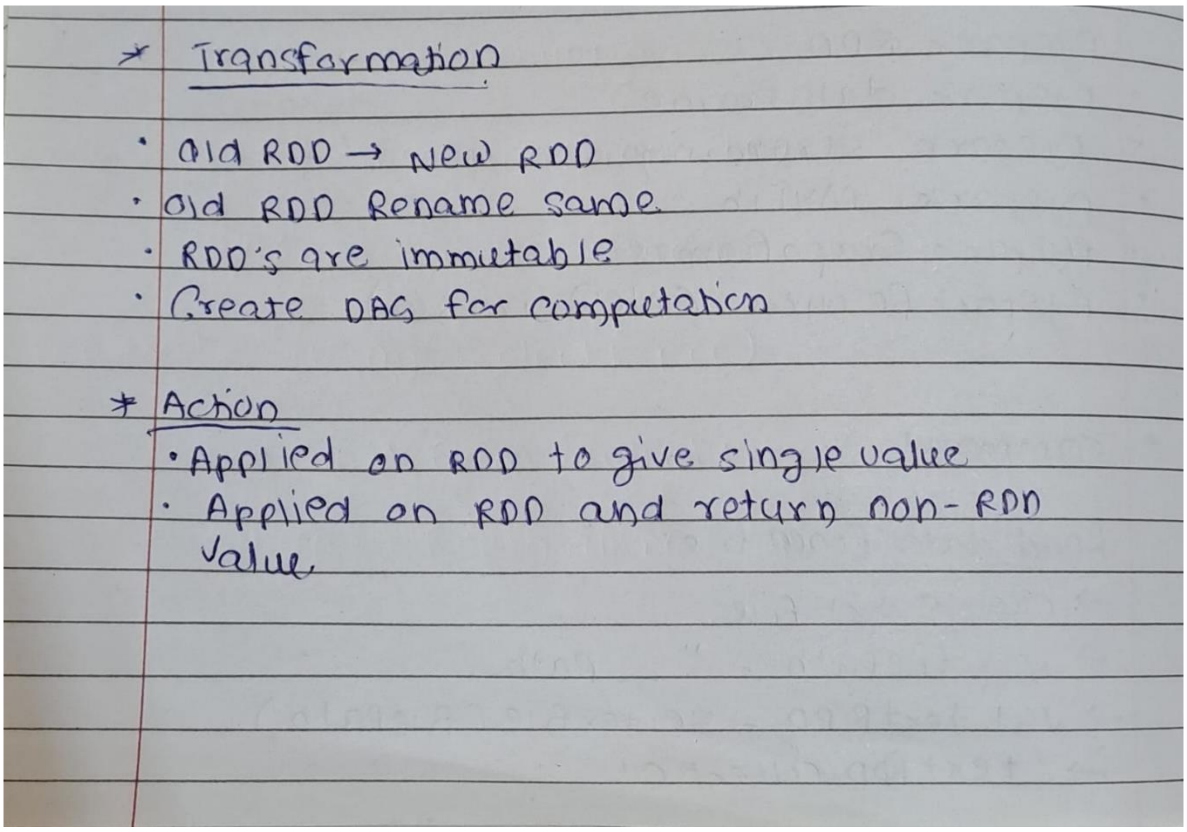
PySpark Assignment 3

Pratik Wani

- Notes:



* Transformation

- old RDD → New RDD
- old RDD Rename same
- RDD's are immutable
- Create DAG for computation

* Action

- Applied on RDD to give single value
- Applied on RDD and return non-RDD value

PAGE NO.:

* Sparkcontex

```
from pyspark import spartcontext
SC = SparkContext.getorcreatec)
```

* RDD operation: (Action)

① .Count() → Return the number of elements in RDD

② .first() → Returns first element of RDD

③ .take() → parameter (n) → number of elements we want fetch from RDD.

④ .reduce() → Applies any Aggregate function on RDD and gives single output.

⑤ .saveASTextfile() → save the RDD in files such as success. Part-0001.

* RDD operation: (Transfortion)

① .map() → Apply perticular function on each element of RDD and Returns the result.

2) .filter() :- Applies with lambda function
it will filter the elements from
RDD based on Lambad function

3) .union() → will combine two RDD and
return the result.

*) pyspark pair RDD operations

1) .reduceByKey() → Applies Aggregate function
Based on the key and returns
Result by applying aggregate
function for same key.

2) .

* Renaming column:-

method 1: withColumnRenamed()

Syn: Dataframe.withColumnRenamed (old, new)

method 2: using selectExpr()

Syn: Dataframe .selectExpr (→ all col names)

return : df with all new col names.

PAGE NO.:

method 3 using select method

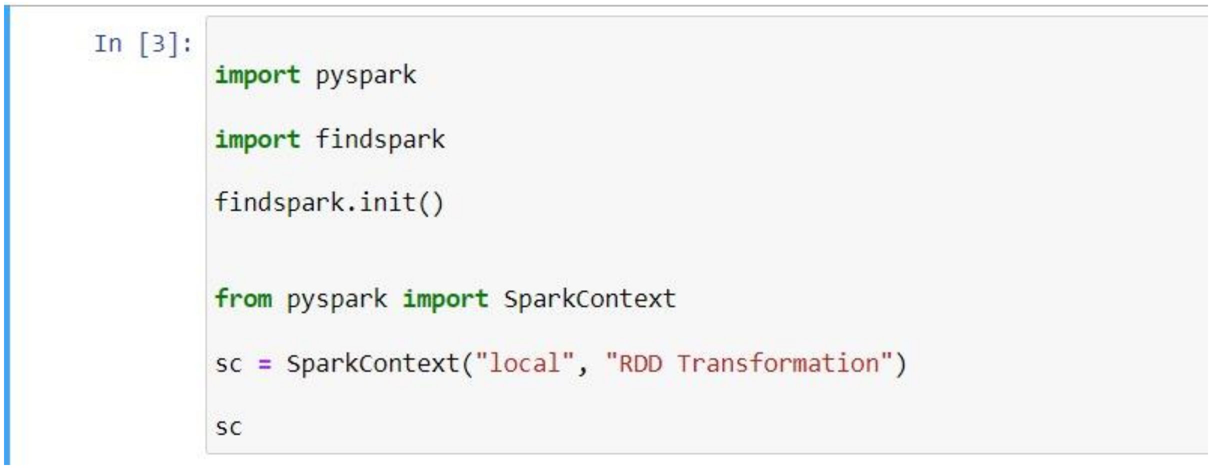
```
import pyspark.sql.functions import col

data = df.select ( col(—), col(—),
                   col(—) AS Newname )
```

method 4 using toDF()

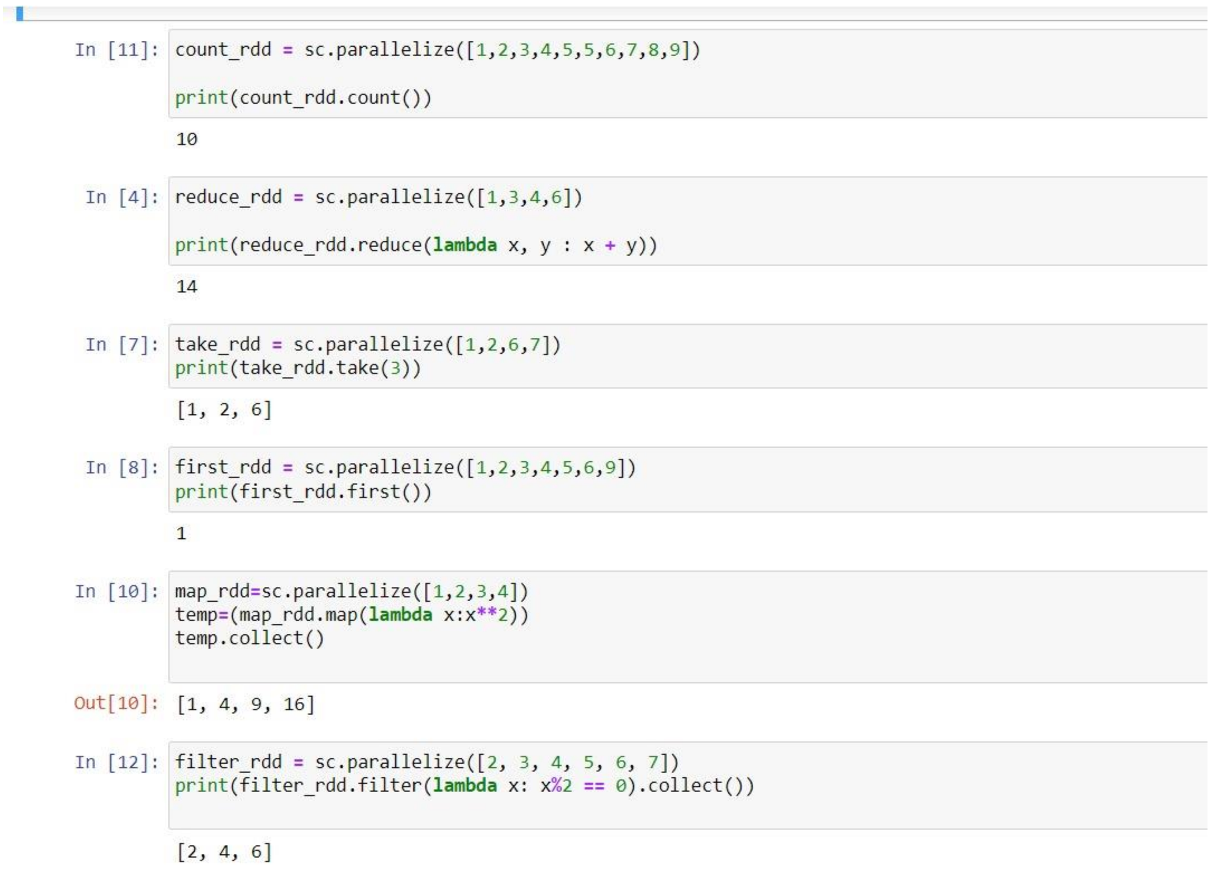
Syn :-

```
List = ['newcolname1', 'Newcolname2'
        ——]

newdf = df.toDF(* List)
```

- Operations On RDD



```
In [3]:
import pyspark

import findspark

findspark.init()


from pyspark import SparkContext

sc = SparkContext("local", "RDD Transformation")

sc
```



```
In [11]: count_rdd = sc.parallelize([1,2,3,4,5,5,6,7,8,9])

         print(count_rdd.count())

         10

In [4]:  reduce_rdd = sc.parallelize([1,3,4,6])

         print(reduce_rdd.reduce(lambda x, y : x + y))

         14

In [7]:  take_rdd = sc.parallelize([1,2,6,7])
         print(take_rdd.take(3))

         [1, 2, 6]

In [8]:  first_rdd = sc.parallelize([1,2,3,4,5,6,9])
         print(first_rdd.first())

         1

In [10]: map_rdd=sc.parallelize([1,2,3,4])
         temp=(map_rdd.map(lambda x:x**2))
         temp.collect()

Out[10]: [1, 4, 9, 16]

In [12]: filter_rdd = sc.parallelize([2, 3, 4, 5, 6, 7])
         print(filter_rdd.filter(lambda x: x%2 == 0).collect())

         [2, 4, 6]
```

```
[2, 4, 6]
```

In [13]:
```python
all= sc.parallelize([2,4,5,6,7,8,9])
even= all.filter(lambda x: x % 2 == 0)
odd = all.filter(lambda x: x % 2 != 0)
print(even.union(odd).collect())
```

```
[2, 4, 6, 8, 5, 7, 9]
```

In [16]:
```python
flatmap_rdd = sc.parallelize(["Hey My name is Pratik Arun Wani", "This is my 1st PySpark RDD Transformations program"])
(flatmap_rdd.flatMap(lambda x: x.split(" ")).collect())
```

Out[16]:
```
['Hey',
 'My',
 'name',
 'is',
 'Pratik',
 'Arun',
 'Wani',
 'This',
 'is',
 'my',
 '1st',
 'PySpark',
 'RDD',
 'Transformations',
 'program']
```

In [17]:
```python
marks = [('Pratik', 88), ('Jagan', 91), ('Shirin', 90), ('Abhijeet', 90), ('John', 71)]
sc.parallelize(marks).collect()
```

Out[17]:
```
[('Pratik', 88), ('Jagan', 91), ('Shirin', 90), ('Abhijeet', 90), ('John', 71)]
```

In [20]:
```python
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29), ('Rohan', 22), ('Rahul', 23),
                            ('Swati', 19), ('Shreya', 28), ('Abhay', 26), ('Rohan', 22)])
print(marks_rdd.reduceByKey(lambda x,y : x*y ).collect())
```

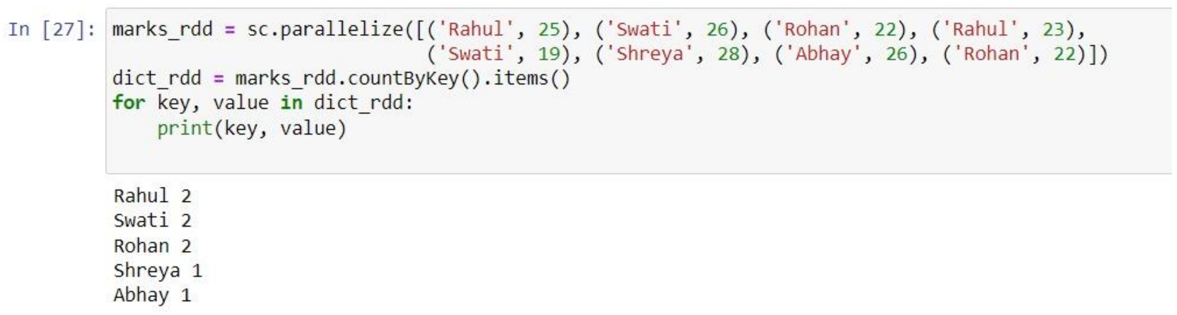
```
[('Rahul', 575), ('Swati', 494), ('Shreya', 616), ('Abhay', 754), ('Rohan', 484)]
```

In [21]:
```python
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29),
                            ('Rohan', 22), ('Rahul', 23), ('Swati', 19), ('Shreya', 28), ('Abhay', 26), ('Rohan', 22)])
print(marks_rdd.sortByKey().collect())
```

```
[('Abhay', 29), ('Abhay', 26), ('Rahul', 25), ('Rahul', 23), ('Rohan', 22), ('Rohan', 22), ('Shreya', 22), ('Shreya', 28), ('Sw
ati', 26), ('Swati', 19)]
```

In [26]:
```python
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29),
                            ('Rohan', 22), ('Rahul', 23), ('Swati', 19), ('Shreya', 28), ('Abhay', 26), ('Rohan', 22)])
dict_rdd = marks_rdd.groupByKey().collect()
for item in dict(dict_rdd).items():
    print(item[0]," ",list(item[1]))
```

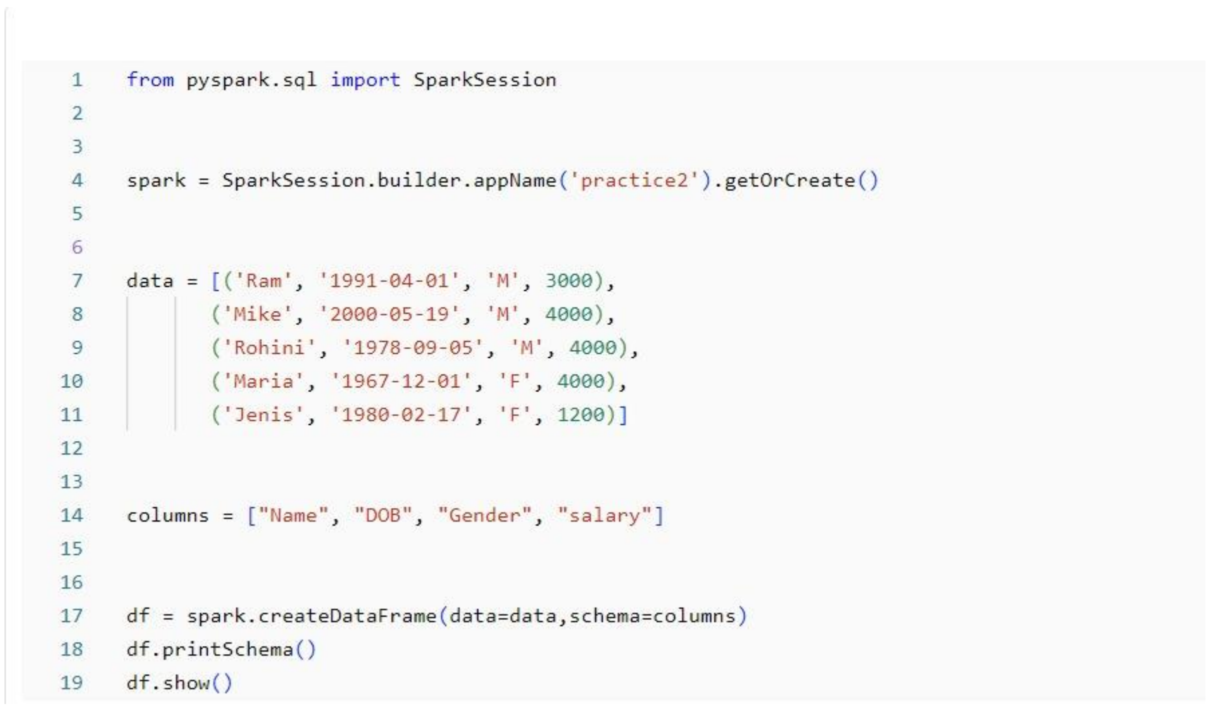
```
Rahul   [25, 23]
Swati   [26, 19]
Shreya   [22, 28]
Abhay   [29, 26]
Rohan   [22, 22]
```



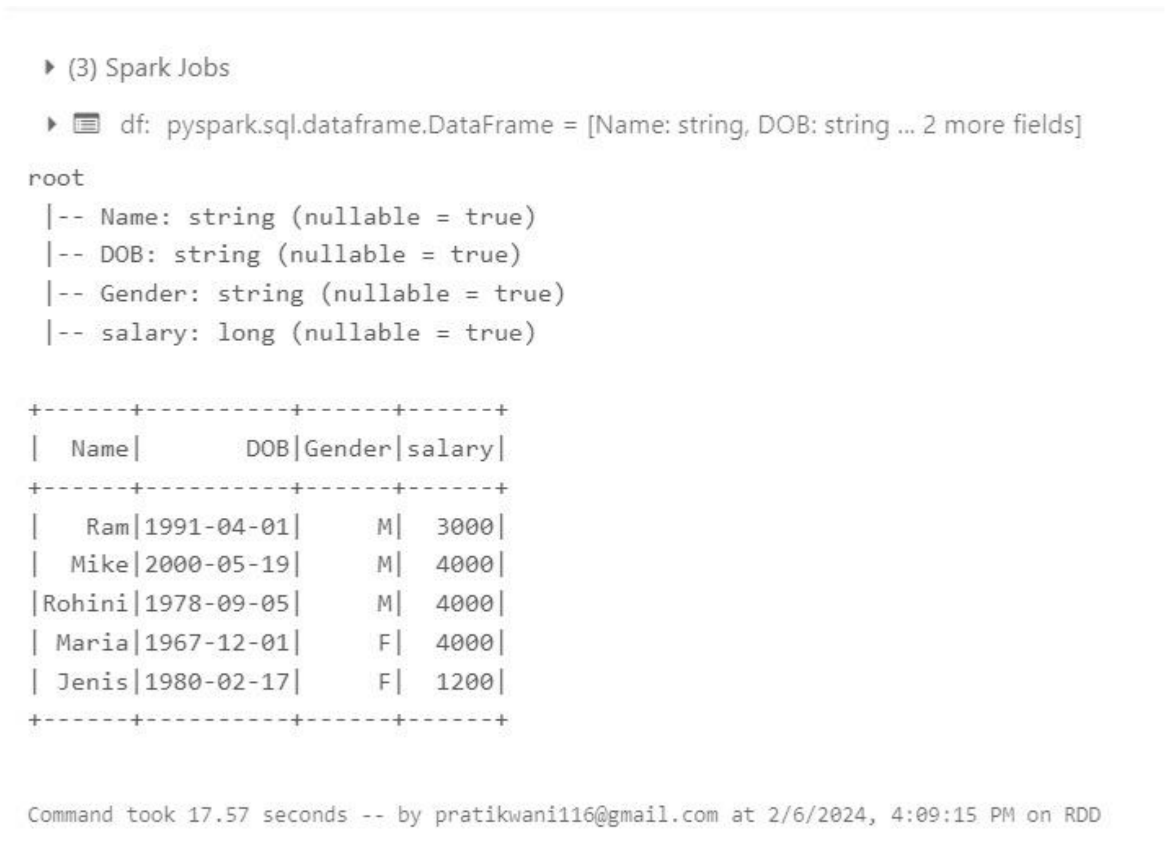
```
In [27]: marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Rohan', 22), ('Rahul', 23),
                                     ('Swati', 19), ('Shreya', 28), ('Abhay', 26), ('Rohan', 22)])
         dict_rdd = marks_rdd.countByKey().items()
         for key, value in dict_rdd:
             print(key, value)
```

```
Rahul 2
Swati 2
Rohan 2
Shreya 1
Abhay 1
```

- DataFrames and Renaming Columns



```
from pyspark.sql import SparkSession


spark = SparkSession.builder.appName('practice2').getOrCreate()


data = [('Ram', '1991-04-01', 'M', 3000),
        ('Mike', '2000-05-19', 'M', 4000),
        ('Rohini', '1978-09-05', 'M', 4000),
        ('Maria', '1967-12-01', 'F', 4000),
        ('Jenis', '1980-02-17', 'F', 1200)]


columns = ["Name", "DOB", "Gender", "salary"]


df = spark.createDataFrame(data=data,schema=columns)
df.printSchema()
df.show()
```

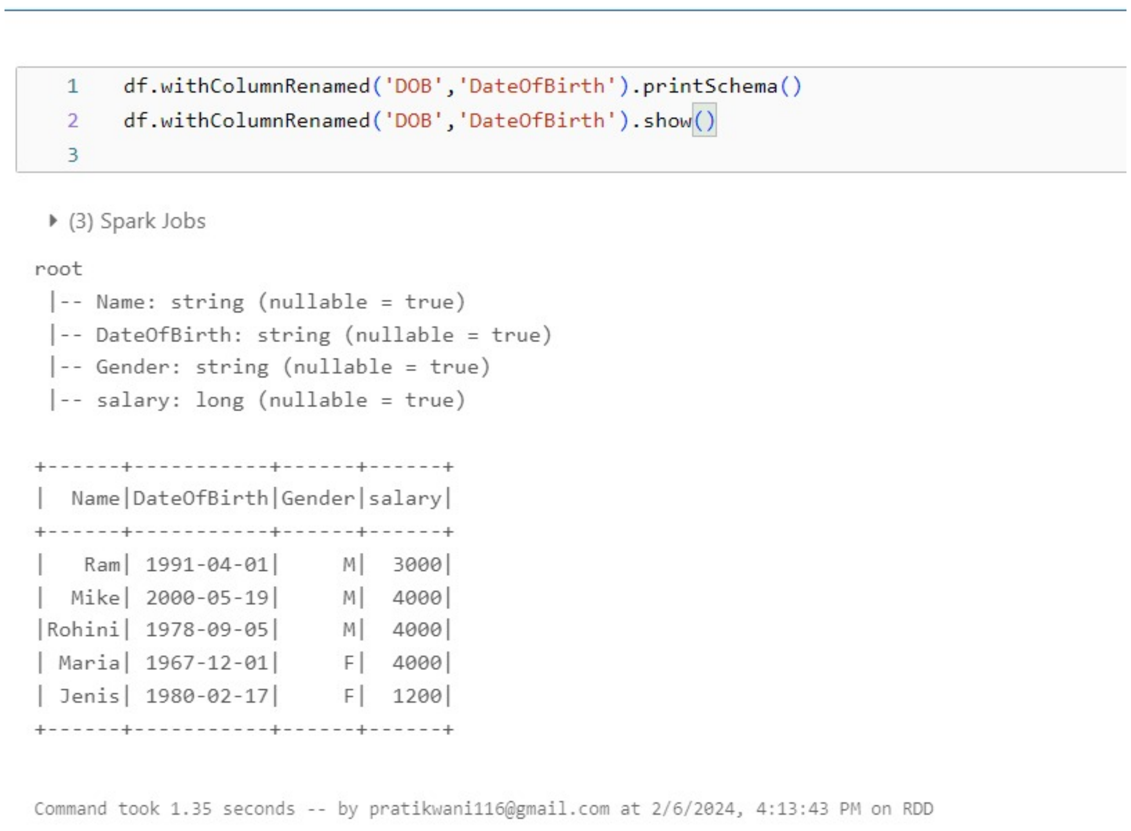


▸ (3) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [Name: string, DOB: string ... 2 more fields]

```
root
 |-- Name: string (nullable = true)
 |-- DOB: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- salary: long (nullable = true)

+------+----------+------+------+
|  Name|       DOB|Gender|salary|
+------+----------+------+------+
|   Ram|1991-04-01|     M|  3000|
|  Mike|2000-05-19|     M|  4000|
|Rohini|1978-09-05|     M|  4000|
| Maria|1967-12-01|     F|  4000|
| Jenis|1980-02-17|     F|  1200|
+------+----------+------+------+
```

Command took 17.57 seconds -- by pratikwani116@gmail.com at 2/6/2024, 4:09:15 PM on RDD

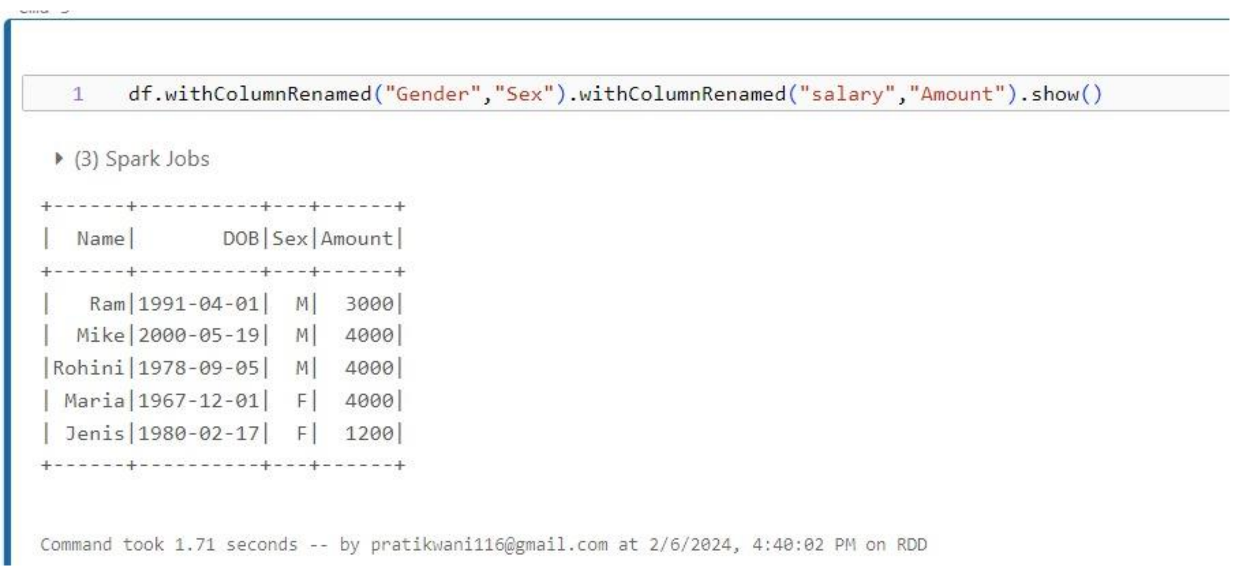


```
df.withColumnRenamed('DOB','DateOfBirth').printSchema()
df.withColumnRenamed('DOB','DateOfBirth').show()
```

▸ (3) Spark Jobs

```
root
 |-- Name: string (nullable = true)
 |-- DateOfBirth: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- salary: long (nullable = true)

+------+-----------+------+------+
|  Name|DateOfBirth|Gender|salary|
+------+-----------+------+------+
|   Ram| 1991-04-01|     M|  3000|
|  Mike| 2000-05-19|     M|  4000|
|Rohini| 1978-09-05|     M|  4000|
| Maria| 1967-12-01|     F|  4000|
| Jenis| 1980-02-17|     F|  1200|
+------+-----------+------+------+
```

Command took 1.35 seconds -- by pratikwani116@gmail.com at 2/6/2024, 4:13:43 PM on RDD

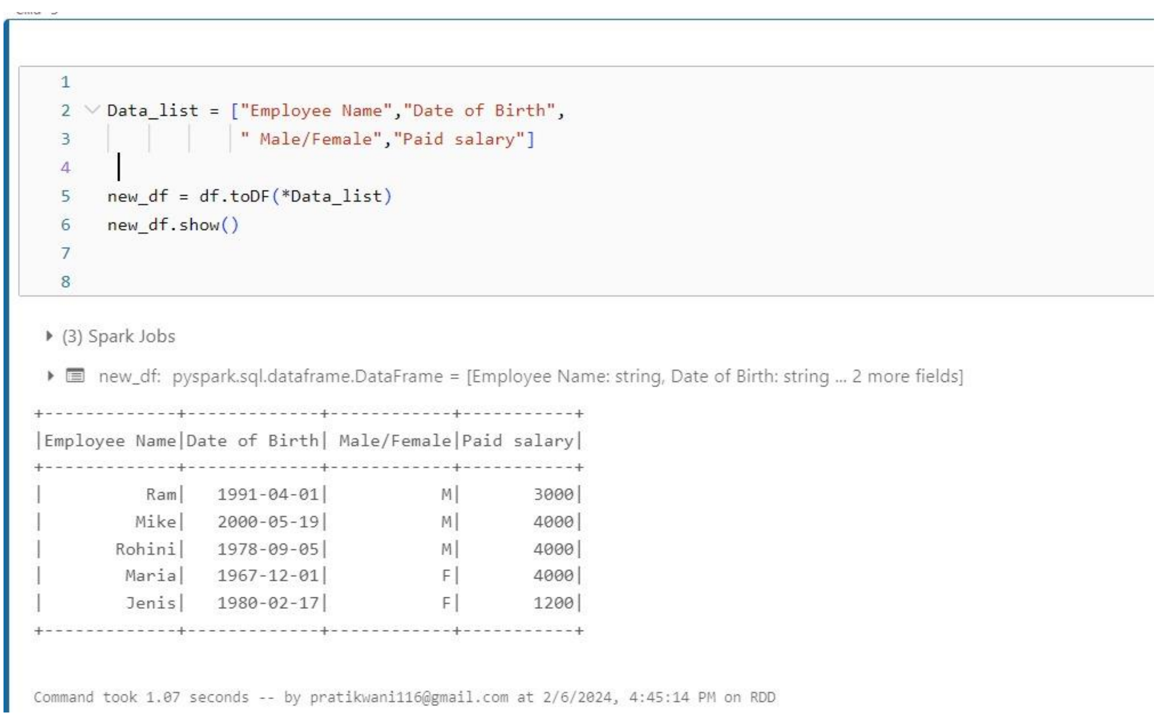


```
df.withColumnRenamed("Gender","Sex").withColumnRenamed("salary","Amount").show()
```

▸ (3) Spark Jobs

```
+------+----------+---+------+
|  Name|       DOB|Sex|Amount|
+------+----------+---+------+
|   Ram|1991-04-01|  M|  3000|
|  Mike|2000-05-19|  M|  4000|
|Rohini|1978-09-05|  M|  4000|
| Maria|1967-12-01|  F|  4000|
| Jenis|1980-02-17|  F|  1200|
+------+----------+---+------+
```

Command took 1.71 seconds -- by pratikwani116@gmail.com at 2/6/2024, 4:40:02 PM on RDD



```

Data_list = ["Employee Name","Date of Birth",
             " Male/Female","Paid salary"]

new_df = df.toDF(*Data_list)
new_df.show()


```

▸ (3) Spark Jobs

▸ new_df: pyspark.sql.dataframe.DataFrame = [Employee Name: string, Date of Birth: string ... 2 more fields]

```
+-------------+-------------+------------+-----------+
|Employee Name|Date of Birth| Male/Female|Paid salary|
+-------------+-------------+------------+-----------+
|          Ram|   1991-04-01|           M|       3000|
|         Mike|   2000-05-19|           M|       4000|
|       Rohini|   1978-09-05|           M|       4000|
|        Maria|   1967-12-01|           F|       4000|
|        Jenis|   1980-02-17|           F|       1200|
+-------------+-------------+------------+-----------+
```

Command took 1.07 seconds -- by pratikwani116@gmail.com at 2/6/2024, 4:45:14 PM on RDD