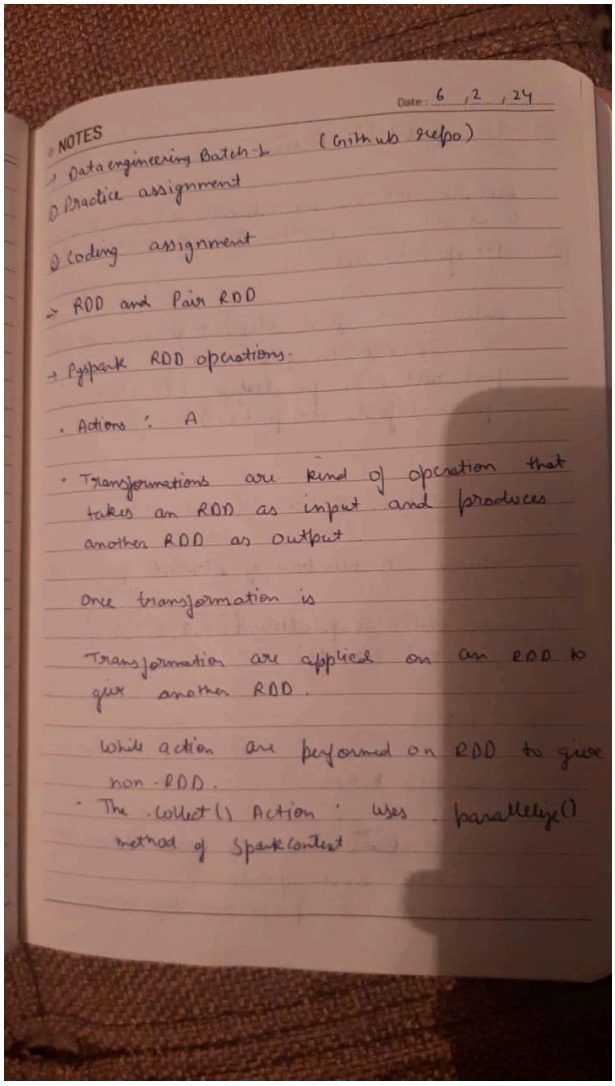
Asmita Porwal
Batch-1
Day-13
6/2/2024
Data engineering

Assignment-13

Hand written notes during the session :



NOTES

Date : ...../...../.....

② The .count() Action

return the no of elements of our RDD
count_rdd uses parallize function and then count

③ The .first() action.

return the first element from our RDD

```
sc = SparkContext.getOrCreate()
first_rdd = sc.parallelize([1,2,3,4,5])
print (first_rdd.first())
```

④ The .take() Action.

returns n numbers of elements from RDD

```
take_rdd = sc.parallelize([1,2,3,4,5])
take_rdd.take(3)
```

o/p = 1, 2, 3

⑤ The .reduce() Action

It takes 2 elements from the given RDD and operates.
operation is performed by using

NOTES Date : ../../..

Anonymous function or lambda

(6) The .saveAsTextFile() Action.

Used to serve the resultant RDD as a text file.
We can even specify the path to which file needed to be saved.

```
save_rdd = sc.parallelize([1,2,3])
save_rdd.saveAsTextFile('file.txt')
```

→ CreateDataFrame

```
df = spark.createDataFrame(data = data, schema = columns)
```
(data → create data; columns → Just column name)

→ toDF → It helps in changing the column name

```
new_df = df.toDF(*newcolumns)
```

→

Pyspark RDD operations

Spark context

In [1]:
```
#!pip install findspark
```

```
Requirement already satisfied: findspark in c:\users\asmita porwal\anaconda3\lib\site-packages (2.0.1)
```

In [2]:
```
import pyspark
import findspark
findspark.init()
```

In [3]:
```
#Creating a SparkContext
from pyspark import SparkContext
sc = SparkContext("local", "RDD Transformation")
sc
```

Out[3]:

SparkContext

Spark UI

Version
`v3.5.0`

Master
`local`

AppName
`RDD Transformation`

Actions:

In [ ]:
```
```

In [34]:
```
collect_rdd = sc.parallelize([1,2,3,4,5])
print(collect_rdd.collect())
```

```
[1, 2, 3, 4, 5]
```

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

```
10
```

In [5]:
```
reduce_rdd = sc.parallelize([1,3,4,6])
print(reduce_rdd.reduce(lambda x, y : x + y))
```

```
14
```

In [32]:
```
take_rdd = sc.parallelize([1,2,3,4,5])
take_rdd.take(3)
```

Out[32]:
```
[1, 2, 3]
```

In [33]:
```
first_rdd = sc.parallelize([1,2,3,4,5])
first_rdd.first()
```

Out[33]:
```
1
```

In [36]:
```
my_rdd = sc.parallelize([1,2,3,4])
```

Transformation:

In [36]:
```
my_rdd = sc.parallelize([1,2,3,4])
print(my_rdd.map(lambda x: x+ 10).collect())
```

```
[11, 12, 13, 14]
```

In [37]:
```
filter_rdd = sc.parallelize([2, 3, 4, 5, 6, 7])
print(filter_rdd.filter(lambda x: x%2 == 0).collect())
```

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

In [38]:
```
filter_rdd_2 = sc.parallelize(['Rahul', 'Swati', 'Rohan', 'Shreya', 'Priya'])
print(filter_rdd_2.filter(lambda x: x.startswith('R')).collect())
```

```
['Rahul', 'Rohan']
```

In [39]:
```
#The .union() Transformation
union_inp = sc.parallelize([2,4,5,6,7,8,9])
union_rdd_1 = union_inp.filter(lambda x: x % 2 == 0)
union_rdd_2 = union_inp.filter(lambda x: x % 3 == 0)
print(union_rdd_1.union(union_rdd_2).collect())
```

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

In [40]:
```
# The .reduceByKey() Transformation
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29), ('Rohan', 22), ('Rahul', 23), ('Swati',
print(marks_rdd.reduceByKey(lambda x, y: x + y).collect())
```

In [40]:
```
# The .reduceByKey() Transformation
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29), ('Rohan', 22), ('Rahul', 23), ('Swati',
print(marks_rdd.reduceByKey(lambda x, y: x + y).collect())
```

```
[('Rahul', 48), ('Swati', 45), ('Shreya', 50), ('Abhay', 55), ('Rohan', 44)]
```

In [41]:
```
# The .sortByKey() Transformation
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29), ('Rohan', 22), ('Rahul', 23), ('Swati',
print(marks_rdd.sortByKey('ascending').collect())
```

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

In [42]:
```
#The .groupByKey() Transformation
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Shreya', 22), ('Abhay', 29), ('Rohan', 22), ('Rahul', 23), ('Swati',
dict_rdd = marks_rdd.groupByKey().collect()
for key, value in dict_rdd:
    print(key, list(value))
```

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

In [43]:
```
#. The countByKey() Action
marks_rdd = sc.parallelize([('Rahul', 25), ('Swati', 26), ('Rohan', 22), ('Rahul', 23), ('Swati', 19), ('Shreya', 28), ('Abhay',
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
```

In [14]:
```
#Creating a Resilient Data Structure (RDD)
rdd = sc.parallelize([('C',85,76,87,91), ('B',85,76,87,91), ("A", 85,78,96,92), ("A", 92,76,89,96)], 4)
print(type(rdd))
```

```
<class 'pyspark.rdd.RDD'>
```

In [ ]:
```

```

In [16]:
```
#Converting the RDD into PySpark DataFrame
sub = ['Division','English','Mathematics','Physics','Chemistry']
marks_df = spark.createDataFrame(rdd, schema=sub)
```

Out[16]:
```
DataFrame[Division: string, English: bigint, Mathematics: bigint, Physics: bigint, Chemistry: bigint]
```

In [17]:
```
#Contents of PySpark DataFrame
marks_df.show()
```

In [17]:
```
#Contents of PySpark DataFrame
marks_df.show()
```

```
+--------+-------+-----------+-------+---------+
|Division|English|Mathematics|Physics|Chemistry|
+--------+-------+-----------+-------+---------+
|       C|     85|         76|     87|       91|
|       B|     85|         76|     87|       91|
|       A|     85|         78|     96|       92|
|       A|     92|         76|     89|       96|
+--------+-------+-----------+-------+---------+
```

In [18]:
```
#The dataType of PySpark DataFrame
print(type(marks_df))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
```

In [19]:
```
#Schema of PySpark DataFrame
marks_df.printSchema()
```

```
root
 |-- Division: string (nullable = true)
 |-- English: long (nullable = true)
 |-- Mathematics: long (nullable = true)
 |-- Physics: long (nullable = true)
 |-- Chemistry: long (nullable = true)
```

In [11]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('pyspark - example join').getOrCreate()
# Create data in dataframe
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)]
# Column names in dataframe
columns = ["Name", "DOB", "Gender", "salary"]
# Create the spark dataframe
df = spark.createDataFrame(data=data,
                           schema=columns)
# Print the dataframe
df.show()
```

```
+------+----------+------+------+
|  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|
+------+----------+------+------+
```

In [44]:
```
df.withColumnRenamed("DOB","DateOfBirth").show()
```

```
+------+-----------+------+------+
|  Name|DateOfBirth|Gender|salary|
```

In [44]:
```
df.withColumnRenamed("DOB","DateOfBirth").show()
```

```
+------+-----------+------+------+
|  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|
+------+-----------+------+------+
```

In [47]:
```
df.withColumnRenamed("Gender","Sex").withColumnRenamed("salary","Amount").show()
```

```
+------+----------+---+------+
|  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|
+------+----------+---+------+
```

In [48]:
```
#Renaming the column names using selectExpr() method
data = df.selectExpr("Name as name","DOB","Gender","salary")
data.show()
```

```
+------+----------+------+------+
|  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|
+------+----------+------+------+
```

In [49]:
```
#Selects the cols in the dataframe and returns a new DataFrame

from pyspark.sql.functions import col
data = df.select(col("Name"),col("DOB"),
                 col("Gender"),
                 col("salary").alias('Amount'))
data.show()
```

```
+------+----------+------+------+
|  Name|       DOB|Gender|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|
+------+----------+------+------+
```

In [13]:
```
columns = ["NewName", "NewDOB", "NewGender", "Newsalary"]
```

In [13]:
```
columns = ["NewName", "NewDOB", "NewGender", "Newsalary"]
new_df= df.toDF(*columns)
new_df.show()
```

```
+-------+----------+---------+---------+
|NewName|    NewDOB|NewGender|Newsalary|
+-------+----------+---------+---------+
|    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|
+-------+----------+---------+---------+
```

In [20]:
```
#Create PySpark DataFrame From an External File

spark = SparkSession.builder.appName('PySpark DataFrame From External Files').getOrCreate()
```

In [21]:
```
#Reading External Files into PySpark DataFrame
# Reading a CSV File
csv_file = spark.read.csv('Output.csv', sep = ',', inferSchema = True, header = True)
```

In [23]:
```
#Reading a TXT File
txt_file = spark.read.text("D:\DataengineeringBatch-1\Practice_Assignment\Assignment-12 Day-12\pyspark.txt")
```

In [24]:
```
# Reading a JSON File
json_file = spark.read.json("D:\DataEngineeringhexa\Python\sample.json", multiLine=True)
```

In [24]:
```
# Reading a JSON File
json_file = spark.read.json("D:\DataEngineeringhexa\Python\sample.json", multiLine=True)
```

In [25]:
```
#Checking DataTypes of PySpark DataFrames
print(type(csv_file))

print(type(txt_file))

print(type(json_file))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
<class 'pyspark.sql.dataframe.DataFrame'>
<class 'pyspark.sql.dataframe.DataFrame'>
```

In [27]:
```
#Checking Schema of PySpark DataFrames
csv_file.printSchema()
txt_file.printSchema()
json_file.printSchema()
```

```
root
 |-- _c0: integer (nullable = true)
 |-- Payer: string (nullable = true)
 |-- Payee: string (nullable = true)
 |-- Amount: integer (nullable = true)
 |-- lineage: string (nullable = true)

root
 |-- value: string (nullable = true)
```

```
 |-- cgpa: double (nullable = true)
 |-- name: string (nullable = true)
 |-- phonenumber: string (nullable = true)
 |-- rollno: long (nullable = true)
```

In [29]:
```
csv_file.show()
```

```
+---+-----+-----+--------+-------+
|_c0|Payer|Payee|  Amount|lineage|
+---+-----+-----+--------+-------+
|  0|   E3|   E5| 5900883|   NULL|
|  1|   E4|   E5| 7393544|   NULL|
|  2|   E7|  E10|  151314|   NULL|
|  3|   E8|  E10|12350990|   NULL|
|  4|   E1|   E4| 3283881|   NULL|
|  5|   E8|   E4|21998843|   NULL|
|  6|   E9|   E3| 4245816|   NULL|
|  7|   E2|   E8| 8550660|   NULL|
|  8|   E7|   E8|  306645|   NULL|
|  9|   E6|   E2|14744711|   NULL|
| 10|   E1|   E6|11359928|   NULL|
| 11|   E7|   E6|  913563|   NULL|
| 12|   E9|   E6|21564905|   NULL|
| 13|   E9|   E1| 2812166|   NULL|
+---+-----+-----+--------+-------+
```

In [30]:
```
txt_file.show()
```

```
+--------------------+
|               value|
+--------------------+
|PySpark has been ...|
+--------------------+
```

In [31]:
```
json_file.show()
```

```
+----+-----------+-----------+------+
|cgpa|       name|phonenumber|rollno|
+----+-----------+-----------+------+
| 8.6|sathiyajith| 9976770500|    56|
+----+-----------+-----------+------+
```

In [ ]: