Assignment – Day 13

-Sarthak Niranjan Kulkarni (Maverick)

- [sarthakkul2311@gmail.com](mailto:sarthakkul2311@gmail.com) - (+91) 93256 02791

**20/11/2024 (Wednesday)**

**Transaction & Action Practice:-**

1. **Creating RDDs and DataFrame in PySpark with a Schema: -**

#to create rdds and  dataframe

from pyspark import SparkContext

from pyspark.sql import  SparkSession

sc =SparkContext.getOrCreate()

spark = SparkSession.builder.appName('pyspark first program').getOrCreate()

#create the 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)

mydata = ['Division','English','Mathematics','Physics','Chemistry']

marks\_df = spark.createDataFrame(rdd, schema=mydata)

print(rdd.collect())

print(rdd) #---Transformation which gives rdd value

rdd.collect() #----Action gives non rdd value

A screenshot of a computer

Description automatically generated

1. **Creating RDDs, Converting to DataFrame, and Performing Actions in PySpark: -**

#to create rdds and  dataframe

from pyspark import SparkContext

from pyspark.sql import  SparkSession

sc =SparkContext.getOrCreate()

spark = SparkSession.builder.appName('pyspark first program').getOrCreate()

#create the 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)

mydata = ['Division','English','Mathematics','Physics','Chemistry']

marks\_df = spark.createDataFrame(rdd, schema=mydata)

print(rdd.count())

rdd.take(2) ##Action gives non rdd value

A screenshot of a computer

Description automatically generated

1. **Counting the Number of Elements in an RDD Using PySpark: -**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

count\_rdd = sc.parallelize([1,2,3,4,5,5,6,7,8,9])

print(count\_rdd.count())

count\_rdd.count()

A screenshot of a computer

Description automatically generated

1. **Using count() and first() Actions to Analyze an RDD in PySpark: -**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

first\_rdd = sc.parallelize([1,2,3,4,5,5,6,7,8,9])

print(first\_rdd.count())

first\_rdd.count()

first\_rdd.first() #First method is action

A screenshot of a computer

Description automatically generated

1. **Filtering Elements of an RDD Based on a Condition in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

filter\_rdd\_2 = sc.parallelize(['Rahul', 'Sarthak', 'Rohan', 'Shreya', 'Priya'])

print(filter\_rdd\_2.filter(lambda x: x.startswith('S')).collect())

A screenshot of a computer

Description automatically generated

1. **Applying Filters and Performing Union Operation on RDDs in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

uninon\_inp = sc.parallelize([2,4,5,6,7,8,9,10])

uninon\_rdd\_1 = uninon\_inp.filter(lambda x:x % 2 == 0)

uninon\_rdd\_2 = uninon\_inp.filter(lambda x:x % 3 == 0)

print(uninon\_rdd\_1.union(uninon\_rdd\_2).collect())

**A screenshot of a phone

Description automatically generated**

1. **Using flatMap() Transformation to Split and Flatten Data in an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

flatmap\_rdd = sc.parallelize(["Hey there", "This is PySpark RDD Transformations"])

print(flatmap\_rdd.flatMap(lambda x :x.split(" ").collect()))

flatmap\_rdd.flatMap(lambda x :x.split(" ").collect())

A screenshot of a computer

Description automatically generated

1. **Using flatMap() to Split and Flatten Strings into Words in an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

flatmap\_rdd = sc.parallelize(["Hey there", "This is PySpark RDD Transformations"])

(flatmap\_rdd.flatMap(lambda x: x.split(" ")).collect())

A screen shot of a computer

Description automatically generated

1. **Using reduce() Action to Aggregate Elements in an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

reduce\_rdd = sc.parallelize([1,2,3])

print(reduce\_rdd.reduce(lambda x,y:x+y))

A screenshot of a computer

Description automatically generated

1. **Saving an RDD to a Text File Using saveAsTextFile() in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

save\_rdd = sc.parallelize([1,2,3,4,5,5])

save\_rdd.saveAsTextFile('file3.txt')



1. **Using map() Transformation to Modify Elements in an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

map\_rdd = sc.parallelize([1,2,3])

print(map\_rdd.map(lambda x:x+10))

print(map\_rdd.map(lambda x:x+10).collect())

**A screenshot of a computer

Description automatically generated**

1. **Using filter() Transformation to Select Even Numbers from an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

filter\_rdd = sc.parallelize([1,2,3])

print(filter\_rdd.filter(lambda x:x%2==0))

print(filter\_rdd.filter(lambda x:x%2==0).collect())

A screenshot of a computer

Description automatically generated

1. **Creating an RDD, Converting it to a DataFrame, and Registering it as a Temp View in PySpark**

from pyspark import SparkContext

from pyspark.sql import  SparkSession

sc =SparkContext.getOrCreate()

spark = SparkSession.builder.appName('pyspark first program').getOrCreate()

#create the 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)

mydata = ['Division','English','Mathematics','Physics','Chemistry']

marks\_df = spark.createDataFrame(rdd, schema=mydata)

print(marks\_df.createOrReplaceTempView("dataofmarks"))

A screen shot of a computer

Description automatically generated

1. **Using take() Action to Retrieve the First N Elements from an RDD in PySpark**

from pyspark import SparkContext

sc = SparkContext.getOrCreate()

count\_rdd = sc.parallelize([1,2,3,4,5,5,6,7,8,9])

print(count\_rdd.take(2))

count\_rdd.take(6)

A screenshot of a computer

Description automatically generated

**Summary of Transaction & Action:-**

1. **Creating RDDs:**

* RDDs are created using sc.parallelize() with sample data.

1. **Creating DataFrames:**

* RDDs are converted to DataFrames using spark.createDataFrame() with a predefined schema.

1. **RDD Transformations:**

* map(): Applies a function to each element of the RDD.
* filter(): Filters elements based on a condition.
* flatMap(): Flattens the results of applied functions (e.g., splitting strings into words).
* union(): Combines two RDDs into one.

1. **RDD Actions:**

* count(): Counts the number of elements in an RDD.
* first(): Returns the first element of the RDD.
* reduce(): Reduces the RDD to a single value (e.g., summing elements).
* collect(): Collects all elements from the RDD to the driver.
* saveAsTextFile(): Saves the RDD to a text file.

1. **DataFrame Operations:**

* A DataFrame can be registered as a temporary SQL view using createOrReplaceTempView(), enabling SQL queries on the data.

1. **File Operations:**

* The code includes file I/O operations like saving an RDD to a text file using saveAsTextFile().

1. **Lazy and Eager Execution:**

* Transformations are lazy (not executed immediately), while actions are eager (trigger computation and return results).