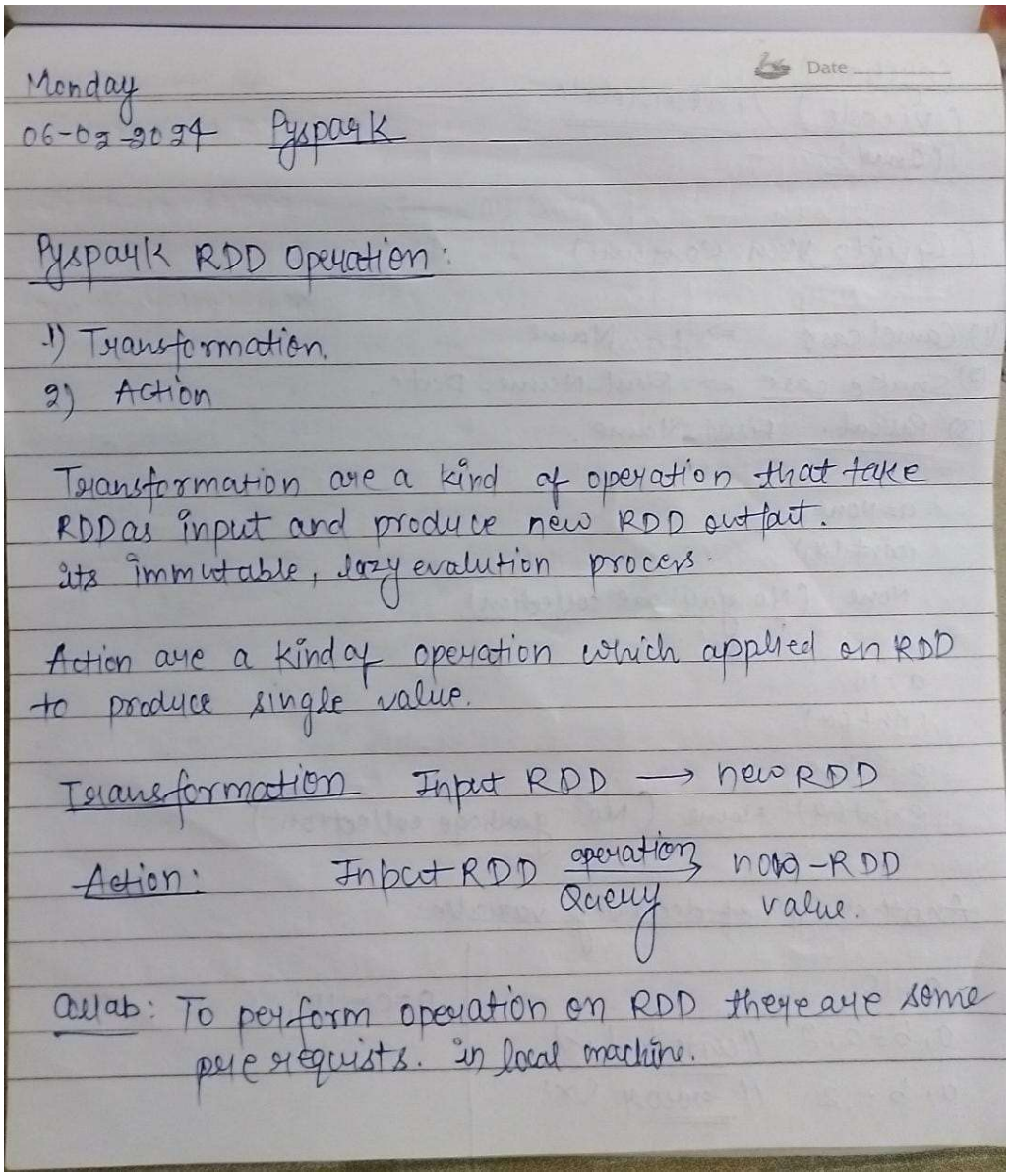
Name: Abhishek Kanoujia

DATA ENGINEERING BATCH 1

DAY 14 ASSIGNMENT

Class hand written notes:-



Monday
06-02-2024 Pyspark

Pyspark RDD Operation:

1) Transformation.
2) Action

Transformation are a kind of operation that take RDDas input and produce new RDD output.
its immutable, lazy evalution process.

Action are a kind of operation which applied on RDD to produce single value.

Transformation Input RDD → new RDD

Action: Input RDD —operation/Query→ non-RDD value.

Collab: To perform operation on RDD there are some pre requists. in local machine.

Date

Operation:

(1). Count = Return count.
(2). first = Return first
(3). take = Return n no from RDD.
(4). Reduce = landa.
(5). saveAsText = save data in text file.
(6). filter.
(7). Union

Method:

(1) ~~Create~~ :- Using with Column Renaming
(2) ~~Select~~ :- selectExpr()
(3) Using select()
(4) Using toDF()
(5)

} example for practise

Transforming data with PySpark RDDs & its hands-on:-

Screenshots are attach below

Selecting, Renaming, and Filtering Data in a Pandas DataFrame:-

Screenshots are attach below

databricks

abhishek7621cse@gmail.com

2nd program

Python

File Edit View Run Help Last edit was 4 minutes ago New cell UI: OFF

Run all | My Cluster | Share | Publish

Cmd 1

```
import 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())
```

(1) Spark Jobs

```
10
```

Command took 5.81 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 3:10:59 PM on My Cluster

Cmd 2

Python

```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
reduce_rdd = sc.parallelize([1,3,4,6])
print(reduce_rdd.reduce(lambda x, y : x + y))
```

(1) Spark Jobs

```
14
```

Command took 0.86 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 3:15:40 PM on My Cluster

Cmd 3

```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
save_rdd = sc.parallelize([1,2,3,4,5,6])
save_rdd.saveAsTextFile('file4.txt')


```

(1) Spark Jobs

Command took 2.15 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 3:16:13 PM on My Cluster

Cmd 4

```
from pyspark.sql import SparkSession

# Create a spark session
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()
```

(3) Spark Jobs

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

```
+------+----------+------+------+
|  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 19.79 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:36:16 PM on My Cluster

Cmd 5

```
df.withColumnRenamed("DOB","DateOfBirth").show()
```

(3) Spark Jobs

```
+------+-----------+------+------+
|  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 1.23 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:38:06 PM on My Cluster

Cmd 6

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

```
2
```

▸ (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 0.91 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:38:36 PM on My Cluster

Cmd 7

```
1
2  # Select the 'Name' as 'name'
3  # Select remaining with their original name
4  data = df.selectExpr("Name as name","DOB","Gender","salary")
5
6  # Print the dataframe
7  data.show()
8
9
```

▸ (3) Spark Jobs

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

```
+------+----------+------+------+
|  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 1.20 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:39:13 PM on My Cluster

Cmd 8

```
1  # Import col method from pyspark.sql.functions
2  from pyspark.sql.functions import col
3
4  # Select the 'salary' as 'Amount' using aliasing
5  # Select remaining with their original name
6  data = df.select(col("Name"),col("DOB"),
7                   col("Gender"),
8                   col("salary").alias('Amount'))
9
10 # Print the dataframe
11 data.show()
12
```

▸ (3) Spark Jobs

▾ data: pyspark.sql.dataframe.DataFrame

```
Name: string
DOB: string
Gender: string
Amount: long
```

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

Command took 1.20 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:40:28 PM on My Cluster

Cmd 9

```
1  Data_list = ["Emp Name","Date of Birth",
2               " Gender-m/f","Paid salary"]
3
4  new_df = df.toDF(*Data_list)
5  new_df.show()
6
```

▸ (3) Spark Jobs

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

```
+--------+-------------+-----------+-----------+
|Emp Name|Date of Birth| Gender-m/f|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 1.16 seconds -- by abhishek7621cse@gmail.com at 2/6/2024, 4:41:05 PM on My Cluster

Cmd 10

Python

```
1
```

[Shift+Enter] to run
[Shift+Ctrl+Enter] to run selected text

jupyter 2nd Last Checkpoint: 5 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
import pyspark
import findspark
findspark.init()
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())
```

```
10
```

In [2]:
```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
reduce_rdd = sc.parallelize([1,3,4,6])
print(reduce_rdd.reduce(lambda x, y : x + y))
```

```
14
```

In [20]:
```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
save_rdd = sc.parallelize([1,2,3,4,5,6])
save_rdd.saveAsTextFile('file4.txt')
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
Cell In[20], line 4
      2 sc = SparkContext.getOrCreate()
      3 save_rdd = sc.parallelize([1,2,3,4,5,6])
----> 4 save_rdd.saveAsTextFile('file4.txt')

File C:\pythan310\lib\site-packages\pyspark\rdd.py:3425, in RDD.saveAsTextFile(self, path, compressionCodecClass)
   3423     keyed._jrdd.map(self.ctx._jvm.BytesToString()).saveAsTextFile(path, compressionCodec)
   3424 else:
-> 3425     keyed._jrdd.map(self.ctx._jvm.BytesToString()).saveAsTextFile(path)

File C:\pythan310\lib\site-packages\py4j\java_gateway.py:1322, in JavaMember.__call__(self, *args)
   1316 command = proto.CALL_COMMAND_NAME +\
   1317     self.command_header +\
   1318     args_command +\
   1319     proto.END_COMMAND_PART
   1321 answer = self.gateway_client.send_command(command)
-> 1322 return_value = get_return_value(
```

In [ ]:

In [ ]:

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

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

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

In [ ]:
```
pip install findspark
```

In [ ]:
```
python.exe -m pip install --upgrade pip
```

In [ ]:
```
pip install --upgrade
```

In [8]:
```
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())
first_rdd = sc.parallelize([1,2,3,4,5,6,7,8,9,10])
print(first_rdd.first())
```

```
10
1
```

In [9]:
```
take_rdd = sc.parallelize([1,2,3,4,5])
print(take_rdd.take(3))
```

```
[1, 2, 3]
```

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

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

In [11]:
```
filter_rdd = sc.parallelize([2, 3, 4, 5, 6, 7])
print(filter_rdd.filter(lambda x: x%2 == 0).collect())
```

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

In [12]:
```
filter_rdd_2 = sc.parallelize(['Rahul', 'Swati', 'Rohan', 'Shreya', 'Priya'])
print(filter_rdd_2.filter(lambda x: x.startswith('R')).collect())
```

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

In [13]:
```
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 [14]:
```
flatmap_rdd = sc.parallelize(["Hey there", "This is PySpark RDD Transformations"])
(flatmap_rdd.flatMap(lambda x: x.split(" ")).collect())
```

Out[14]:
```
['Hey', 'there', 'This', 'is', 'PySpark', 'RDD', 'Transformations']
```

In [15]:
```
marks = [('Rahul', 88), ('Swati', 92), ('Shreya', 83), ('Abhay', 93), ('Rohan', 78)]
sc.parallelize(marks).collect()
```

Out[15]:
```
[('Rahul', 88), ('Swati', 92), ('Shreya', 83), ('Abhay', 93), ('Rohan', 78)]
```

In [16]:
```
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())
```

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

In [17]:
```
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 [18]:
```
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))
```

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

In [19]:
```
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 [ ]:
```
```