Spark Lab Assessment

Q.1. Perform the following transformation operations on RDD

- Distinct
- Filter

Answer 1

Note: I have completed this Lab Assessment using Jupyter Notebook

Spark Lab Assessment by Mr. Harshit Dawar!

In [1]:
```
# Importing SparkSession to create a session of Spark
from pyspark.sql import SparkSession

# Required to create SparkContext & Processing
from pyspark import SparkConf, SparkContext, SQLContext
```

In [2]:
```
# Creatings SparkContext & Session
configuration = SparkConf().setAppName("Spark Lab!").setMaster("local")
sc = SparkContext(conf = configuration)
spark = SparkSession.builder.appName("Spark Lab!").config(
    "spark.sql.broadcastTimeout", "3600").config(     # Setting the timeout to be 1 Hour!
    "spark.driver.memory", "6G").config(         # Setting the Spark Drive Memory to 6GB
    "spark.sql.autoBroadcastJoinThreshold", "-1").getOrCreate()       # Disabling Broadcasting
```

Creating RDD from Parallelizing Collection!

In [13]:
```
rdd_type_1 = sc.parallelize([("Machine Learning", 100),
                             ("Deep Learning", 100),
                             ("Data Science", 100),
                             ("Big Data", 100),
                             ("DevOps", 100),
                           ("Machine Learning", 100),
                             ("Deep Learning", 100),
                             ("Data Science", 100),
                             ("Big Data", 100),
                             ("DevOps", 100)]
                          )
```

```
In [14]: 1 rdd_type_1.collect()

Out[14]: [('Machine Learning', 100),
          ('Deep Learning', 100),
          ('Data Science', 100),
          ('Big Data', 100),
          ('DevOps', 100),
          ('Machine Learning', 100),
          ('Deep Learning', 100),
          ('Data Science', 100),
          ('Big Data', 100),
          ('DevOps', 100)]
```

Applying Distinct function on rdd created from parallelizing the collection!

```
In [15]: 1 rdd_type_1.distinct().collect()

Out[15]: [('Machine Learning', 100),
          ('Deep Learning', 100),
          ('Data Science', 100),
          ('Big Data', 100),
          ('DevOps', 100)]
```

Applying Filter function on rdd created from parallelizing the collection!

In [16]:
```
1 rdd_type_1.filter(lambda x: x[0] == "Data Science").collect()
```

Out[16]:
```
[('Data Science', 100), ('Data Science', 100)]
```

Creating RDD from the external dataset (loaded from hdfs)!

In [20]:
```
1 rdd_type_2 = spark.read.csv("2010-12-01.csv").rdd
```

Applying Distinct function on rdd created from external dataset!

In [37]:
```
1 rdd_type_2.distinct().count()
```

Out[37]:
```
3065
```

In [39]:
```
1 rdd_type_2.take(5)
```

Out[39]:
```
[Row(_c0='InvoiceNo', _c1='StockCode', _c2='Description', _c3='Quantity', _c4='InvoiceDate', _c5='UnitPrice', _c6='Cu
stomerID', _c7='Country'),
 Row(_c0='536365', _c1='85123A', _c2='WHITE HANGING HEART T-LIGHT HOLDER', _c3='6', _c4='2010-12-01 08:26:00', _c5
='2.55', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536365', _c1='71053', _c2='WHITE METAL LANTERN', _c3='6', _c4='2010-12-01 08:26:00', _c5='3.39', _c6='1785
0.0', _c7='United Kingdom'),
 Row(_c0='536365', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 08:26:00', _c5='2.7
5', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536365', _c1='84029G', _c2='KNITTED UNION FLAG HOT WATER BOTTLE', _c3='6', _c4='2010-12-01 08:26:00', _c5
='3.39', _c6='17850.0', _c7='United Kingdom')]
```

Applying Filter function on rdd created from external dataset!

In [46]:
```
1 rdd_type_2.filter(lambda x: x[2] == "CREAM CUPID HEARTS COAT HANGER").count()
```

Out[46]: 5

In [47]:
```
1 rdd_type_2.filter(lambda x: x[2] == "CREAM CUPID HEARTS COAT HANGER").collect()
```

Out[47]:
```
[Row(_c0='536365', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 08:26:00',
5', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536373', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 09:02:00',
5', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536375', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 09:32:00',
5', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536396', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 10:51:00',
5', _c6='17850.0', _c7='United Kingdom'),
 Row(_c0='536406', _c1='84406B', _c2='CREAM CUPID HEARTS COAT HANGER', _c3='8', _c4='2010-12-01 11:33:00',
5', _c6='17850.0', _c7='United Kingdom')]
```

Creating RDD from the existing RDD!

In [60]:
```
1 rdd_type_3 = rdd_type_1.map(lambda x: (x[0][0], x))
```

In [61]:
```
1 rdd_type_3.collect()
```

Out[61]:
```
[('M', ('Machine Learning', 100)),
 ('D', ('Deep Learning', 100)),
 ('D', ('Data Science', 100)),
 ('B', ('Big Data', 100)),
 ('D', ('DevOps', 100)),
 ('M', ('Machine Learning', 100)),
 ('D', ('Deep Learning', 100)),
 ('D', ('Data Science', 100)),
 ('B', ('Big Data', 100)),
 ('D', ('DevOps', 100))]
```

Applying Distinct function on rdd created from existing RDD!

In [63]:
```
1 rdd_type_3.distinct().collect()
```

Out[63]:
```
[('M', ('Machine Learning', 100)),
 ('D', ('Deep Learning', 100)),
 ('D', ('Data Science', 100)),
 ('B', ('Big Data', 100)),
 ('D', ('DevOps', 100))]
```

Applying Filter function on rdd created from existing RDD!

In [65]:
```
1 rdd_type_3.filter(lambda x: x[0] == "M").collect()
```

Out[65]:
```
[('M', ('Machine Learning', 100)), ('M', ('Machine Learning', 100))]
```

Lab Work Completed!

Made By: Mr. Harshit Dawar
Roll Number: R172217022
Branch: Big Data
Batch: B1
Semester: 6th
Subject: In Memory Processing Lab
Submitted To: Ms. Shweta Mongia