Database Project (SWE3033) (Fall 2024)

Homework #4 (50pts, Due date: 10/22)

Student ID: ____2018310773___

Student Name: ____임승재____

Instruction: In this homework, we provide you with a jupyter notebook file (DBP_Homework4.ipynb). You should follow the instructions in these documents carefully.

Submit two files as follows:

- DBP_Homework4_StudentID.zip
  - DBP_Homework4_StudentID.ipynb
  - DBP_Homework4_StudentID.pdf

1. **[10pts]** Calculate the visit frequency for each user to the places ***Pat*** and ***Mat*** visited.
   a. Places that ***Pat*** visited:
      - [‘E-mart’, ‘Starbucks’, ‘GS25’, ‘Starbucks’, ‘HomePlus’, ‘CU’]
   b. Places that ***Mat*** visited:
      - [‘Starbucks’, ‘E-mart’, ‘Starbucks’, ‘LotteMart’, ‘LotteMart’]

[Answer]
Enter your code and result here. You must show your result (captured image).

```
#### EDIT HERE ####

# Convert python variable to RDD (HINE: parallelize())
pat = sc.parallelize(['E-mart', 'Starbucks', 'GS25', 'Starbucks', 'HomePlus', 'CU'])
mat = sc.parallelize(['Starbucks', 'E-mart', 'Starbucks', 'LotteMart', 'LotteMart'])

pat_visit_countByValue = pat.countByValue()
mat_visit_countByValue = mat.countByValue()

####################

print("Pat:", pat_visit_countByValue)
print("Mat:", mat_visit_countByValue)
```

```
Pat: defaultdict(<class 'int'>, {'E-mart': 1, 'Starbucks': 2, 'GS25': 1, 'HomePlus': 1, 'CU': 1})
Mat: defaultdict(<class 'int'>, {'Starbucks': 2, 'E-mart': 1, 'LotteMart': 2})
```

2. **[20pts]** Count the number of words in the given data using the following two operations and explain the difference between the two operations.

Data:

[('apple', 1), ('apple', 1), ('banana', 1), ('apple', 1), ('banana', 1), ('apple', 1), ('apple', 1), ('banana', 1), ('banana', 1)]

a. groupByKey()
b. reduceByKey()
c. Explain the difference between the two operations.

[Answer]

a)	[('banana', 4), ('apple', 5)]
b)	[('banana', 4), ('apple', 5)]
c)	groupByKey() groups the values by the key. However, this is very expensive operation and consumes much memory if the dataset is huge. This is because groupByKey() shuffles all values across the network. On the other hand, reduceByKey() minimizes data shuffling by combining values before shuffling.

Enter your code and result here. You must show your result (captured image).

```
data = sc.parallelize([('apple', 1), ('apple', 1), ('banana', 1), ('apple', 1), ('banana', 1), ('apple', 1), ('apple', 1), ('banana', 1), ('banana', 1)])

#### EDIT HERE ####

from operator import add

data_from_groupByKey = data.groupByKey().mapValues(sum).collect()
data_from_reduceByKey = data.reduceByKey(add).collect()

####################

print(data_from_groupByKey)
print(data_from_reduceByKey)
```

```
[('banana', 4), ('apple', 5)]
[('banana', 4), ('apple', 5)]
```

3. [20pts] The following data represents the songs ***Pat*** and ***Mat*** have listened to and the play counts. Answer the following three questions.

Data: key-value data in (music, # of plays) format
- ***Pat***: [('Thriller', 27), ('Everybody', 31), ('Everybody', 20), ('Billie_Jean', 1)]
- ***Mat***: [('Thriller', 20), ('Sorry', 17), ('Sorry', 3), ('Billie_Jean', 2)]

a. For each user, calculate the number of times each song has been listened to, and store it in a new RDD. (HINT: reduceByKey())
b. Create a new RDD containing songs that both users have listened to and their respective play counts. (HINT: join())

c. Calculate the total number of music plays that ***Pat*** and ***Mat*** have played in common.

[Answer]

Enter your code and result here. You must show your result (captured image).

a.

```
#### EDIT HERE ####
pat = sc.parallelize([('Thriller', 27), ('Everybody', 31), ('Everybody', 20), ('Billie_Jean', 1)])
mat = sc.parallelize([('Thriller', 20), ('Sorry', 17), ('Sorry', 3), ('Billie_Jean', 2)])

pat_reduceByKey = pat.reduceByKey(add)
mat_reduceByKey = mat.reduceByKey(add)

####################
print(pat_reduceByKey)
print(mat_reduceByKey)
```

```
PythonRDD[114] at RDD at PythonRDD.scala:53
PythonRDD[115] at RDD at PythonRDD.scala:53
```

b.

```
#### EDIT HERE ####
pat = sc.parallelize([('Thriller', 27), ('Everybody', 31), ('Everybody', 20), ('Billie_Jean', 1)])
mat = sc.parallelize([('Thriller', 20), ('Sorry', 17), ('Sorry', 3), ('Billie_Jean', 2)])

pat_reduceByKey = pat.reduceByKey(add)
mat_reduceByKey = mat.reduceByKey(add)
```

```
PythonRDD[99] at RDD at PythonRDD.scala:53
```

```
####################
print(pat_reduceByKey)
print(mat_reduceByKey)
```

c.

```
#### EDIT HERE ####

pat_mat_result = pat_mat_join.map(lambda x: (x[0], x[1][0] + x[1][1])).collect()

####################
print(pat_mat_result)
```

```
[('Thriller', 47), ('Billie_Jean', 3)]
```