Exercise 1: Text File Import in Databricks

filePath = "dbfs:/databricks-datasets/SPARK\_README.md"

Question 1: How to check file system?

%fs ls

Question 2: How to load a text file?

filePath = "dbfs:/databricks-datasets/SPARK\_README.md" # path in Databricks File System

lines = sc.textFile(filePath) # read the file into the cluster

Question 3: Show me first 10 lines in the file

lines.take(10)

Question 4: Show me number of lines

lines.count()

Question 5: Show me first line

lines.first()

Question 6: Show me how many times “spark” word contains

linesWithSpark = lines.filter(lambda line: "Spark" in line)

linesWithSpark.count()

Question 7: Split each line into a list of words separated by a space from the dataset

words = lines.flatMap(**lambda** x: x.split(' ')) # split each line into a list of words

words.take(10) # display the first 10 words

Question 8: **Filter the list of words to exclude common stop words**

stopWords = ['','a','\*','and','is','of','the','a'] # define the list of stop words filteredWords = words.filter(**lambda** x: x.lower() **not** **in** stopWords) # filter the words

filteredWords.take(10) # display the first 10 filtered words

Question 9: **Cache the filtered dataset in memory to speed up future actions.**

filteredWords.cache() # cache filtered dataset into memory across the cluster worker nodes

Question 10: **Transform filtered words into list of (word,1) tuples for WordCount**

word1Tuples = filteredWords.map(**lambda** x: (x, 1)) # map the words into (word,1) tuples

word1Tuples.take(10) # display the (word,1) tuples

Question 11: **Aggregate the (word,1) tuples into (word,count) tuples**

wordCountTuples = word1Tuples.reduceByKey(**lambda** x, y: x + y) # aggregate counts for each word

wordCountTuples.take(10) # display the first 10 (word,count) tuples

Question 12: **Display the top 10 (word,count) tuples by count**

sortedWordCountTuples = wordCountTuples.top(10,key=**lambda** (x, y): y) # top 10 (word,count) tuples

**for** tuple **in** sortedWordCountTuples: # display the top 10 (word,count) tuples by count

print str(tuple)

Question 13: **Create a table from the (word,count) tuples**

**from** pyspark.sql **import** Row

wordCountRows = wordCountTuples.map(**lambda** p: Row(word=p[0], count=int(p[1])))

wordCountRows.toDF().createOrReplaceTempView("word\_count")

Question 14: **Use SQL to visualize the words with count >= 2**

%sql

**SELECT** word, count **FROM** word\_count **HAVING** count >= 2 **ORDER** **BY** count **DESC**