



Free Questions for CCA175 by actualtestdumps

Shared by Holloway on 18-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Problem Scenario 69 : Write down a Spark Application using Python,

In which it read a file 'Content.txt' (On hdfs) with following content.

And filter out the word which is less than 2 characters and ignore all empty lines.

Once done store the filtered data in a directory called 'problem84' (On hdfs)

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
```

```
from pyspark import SparkContext, SparkConf
```

```
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 84') sc = sparkContext(conf=conf)
#load data from hdfs
contentRDD = sc.textFile(MContent.txt')
#filter out non-empty lines
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
#Split line based on space
words = nonempty_lines.ffatMap(lambda x: x.split("{}")
#filter out all 2 letter words
finalRDD = words.filter(lambda x: len(x) > 2)
for word in finalRDD.collect():
print(word)
#Save final data finalRDD.saveAsTextFile('problem84M)
step 2 : Submit this application
spark-submit -master yarn problem84.py
```

B) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 84') sc = sparkContext(conf=conf)
#load data from hdfs
print(word)
#Save final data finalRDD.saveAsTextFile('problem84M)
step 2 : Submit this application
```

spark-submit -master yarn problem84.py

Answer:

A

Question 2

Question Type: MultipleChoice

Problem Scenario 70 : Write down a Spark Application using Python, In which it read a file 'Content.txt' (On hdfs) with following content.
Do the word count and save the results in a directory called 'problem85' (On hdfs)

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 85') sc = sparkContext(conf=conf)
#load data from hdfs
.reduceByKey(lambda x, y: x+y) \
.map(lambda x: (x[1], x[0])).sortByKey(False)
for word in wordcounts.collect(): print(word)
#Save final data ' wordcounts.saveAsTextFile('problem85')
```

step 2 : Submit this application

```
spark-submit -master yarn problem85.py
```

B) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 85') sc = sparkContext(conf=conf)
#load data from hdfs
contentRDD = sc.textFile(MContent.txt')
#filter out non-empty lines
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
#Split line based on space
words = nonempty_lines.ffatMap(lambda x: x.split("})
#Do the word count
```

```
wordcounts = words.map(lambda x: (x, 1)) \  
.reduceByKey(lambda x, y: x+y) \  
.map(lambda x: (x[1], x[0])).sortByKey(False) \  
for word in wordcounts.collect(): print(word) \  
#Save final data ' wordcounts.saveAsTextFile('problem85') \  
step 2 : Submit this application \  
spark-submit -master yarn problem85.py
```

Answer:

B

Question 3

Question Type: MultipleChoice

Problem Scenario 71 :

Write down a Spark script using Python,

In which it read a file 'Content.txt' (On hdfs) with following content.

After that split each row as (key, value), where key is first word in line and entire line as value.

Filter out the empty lines.

And save this key value in 'problem86' as Sequence file(On hdfs)

Part 2 : Save as sequence file , where key as null and entire line as value. Read back the stored sequence files.

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 :

```
# Import SparkContext and SparkConf
```

```
from pyspark import SparkContext, SparkConf
```

Step 2:

```
#load data from hdfs
```

```
contentRDD = sc.textFile(MContent.txt')
```

Step 3:

```
#filter out non-empty lines
```

```
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
```

Step 4:

```
#Split line based on space (Remember : It is mandatory to convert is in tuple} words = nonempty_lines.map(lambda x: tuple(x.split(" ", 1)))
```

```
words.saveAsSequenceFile('problem86')
```

Step 5: Check contents in directory problem86 hdfs dfs -cat problem86/part*

Step 6 : Create key, value pair (where key is null)

```
nonempty_lines.map(lambda line: (None, Mne)).saveAsSequenceFile('problem86_1')
```

Step 7 : Reading back the sequence file data using spark. seqRDD = sc.sequenceFile('problem86_1')

Step 8 : Print the content to validate the same.

```
for line in seqRDD.collect():
```

```
print(line)
```

B) Solution :

Step 1 :

```
# Import SparkContext and SparkConf
```

```
from pyspark import SparkContext, SparkConf
```

Step 2:

```
#load data from hdfs
```

```
contentRDD = sc.textFile(MContent.txt')
```

Step 3:

```
#filter out non-empty lines
```

```
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
```

Step 4:

```
#Split line based on space (Remember : It is mandatory to convert is in tuple} words = nonempty_lines.map(lambda x: tuple(x.split(" ", 1)))
```

```
words.saveAsSequenceFile('problem86')
```

Step 5 : Reading back the sequence file data using spark. seqRDD = sc.sequenceFile('problem86_1')

Step 6 : Print the content to validate the same.

```
for line in seqRDD.collect():
```

```
print(line)
```


Answer:

A

Question 4

Question Type: MultipleChoice

Problem Scenario 41 : You have been given below code snippet.

```
val au1 = sc.parallelize(List (('a' , Array(1,2)), ('b' , Array(1,2))))
```

```
val au2 = sc.parallelize(List (('a' , Array(3)), ('b' , Array(2))))
```

Apply the Spark method, which will generate below output.

```
Array[(String, Array[Int])] = Array((a,Array(1, 2)), (b,Array(1, 2)), (a(Array(3)), (b,Array(2))))
```

Options:

A) Solution:

```
au1.union(au2)
```

B) Solution:

```
au1.union(au3)
```

Answer:

A

Question 5

Question Type: FillInTheBlank

Problem Scenario 67 : You have been given below code snippet.

```
lines = sc.parallelize(['Its fun to have fun,','but you have to know how.'])
```

```
M = lines.map( lambda x: x.replace('7 ').replace('.', 'J.replaceC-V ').lower())
```

```
r2 = r1.flatMap(lambda x: x.split())
```

```
r3 = r2.map(lambda x: (x, 1))
```

```
operation1
```

```
r5 = r4.map(lambda x:(x[1],x[0]))
```

```
r6 = r5.sortByKey(ascending=False)
```

r6.take(20)

Write a correct code snippet for operationl which will produce desired output, shown below. [(2, 'fun'), (2, 'to'), (2, 'have'), (1, 'its'), (1, 'know1), (1, 'how1), (1, 'you'), (1, 'but')]

Answer:

Question 6

Question Type: MultipleChoice

Problem Scenario 70 : Write down a Spark Application using Python, In which it read a file 'Content.txt' (On hdfs) with following content.
Do the word count and save the results in a directory called 'problem85' (On hdfs)

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 85') sc = sparkContext(conf=conf)
#load data from hdfs
.reduceByKey(lambda x, y: x+y) \
.map(lambda x: (x[1], x[0])).sortByKey(False)
for word in wordcounts.collect(): print(word)
#Save final data ' wordcounts.saveAsTextFile('problem85')
```

step 2 : Submit this application

```
spark-submit -master yarn problem85.py
```

B) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 85') sc = sparkContext(conf=conf)
#load data from hdfs
contentRDD = sc.textFile(MContent.txt')
#filter out non-empty lines
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
```

```
#Split line based on space
words = nonempty_lines.ffatMap(lambda x: x.split(" "))
#Do the word count
wordcounts = words.map(lambda x: (x, 1)) \
.reduceByKey(lambda x, y: x+y) \
.map(lambda x: (x[1], x[0])).sortByKey(False)
for word in wordcounts.collect(): print(word)
#Save final data ' wordcounts.saveAsTextFile('problem85')
step 2 : Submit this application
spark-submit -master yarn problem85.py
```

Answer:

B

Question 7

Question Type: MultipleChoice

Problem Scenario 69 : Write down a Spark Application using Python,

In which it read a file 'Content.txt' (On hdfs) with following content.

And filter out the word which is less than 2 characters and ignore all empty lines.

Once done store the filtered data in a directory called 'problem84' (On hdfs)

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 84') sc = sparkContext(conf=conf)
#load data from hdfs
contentRDD = sc.textFile(MContent.txt')
#filter out non-empty lines
nonemptyjines = contentRDD.filter(lambda x: len(x) > 0)
#Split line based on space
words = nonempty_lines.ffatMap(lambda x: x.split("{}")
#filter out all 2 letter words
finalRDD = words.filter(lambda x: len(x) > 2)
```

```
for word in finalRDD.collect():
print(word)
#Save final data finalRDD.saveAsTextFile('problem84M)
step 2 : Submit this application
spark-submit -master yarn problem84.py
```

B) Solution :

Step 1 : Create an application with following code and store it in problem84.py

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
# Create configuration object and set App name
conf = SparkConf().setAppName('CCA 175 Problem 84') sc = sparkContext(conf=conf)
#load data from hdfs
print(word)
#Save final data finalRDD.saveAsTextFile('problem84M)
step 2 : Submit this application
spark-submit -master yarn problem84.py
```

Answer:

A

Question 8

Question Type: MultipleChoice

Problem Scenario 71 :

Write down a Spark script using Python,

In which it read a file 'Content.txt' (On hdfs) with following content.

After that split each row as (key, value), where key is first word in line and entire line as value.

Filter out the empty lines.

And save this key value in 'problem86' as Sequence file(On hdfs)

Part 2 : Save as sequence file , where key as null and entire line as value. Read back the stored sequence files.

Content.txt

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Options:

A) Solution :

Step 1 :

```
# Import SparkContext and SparkConf
```



```
from pyspark import SparkContext, SparkConf
```

Step 2:

```
#load data from hdfs
```

```
contentRDD = sc.textFile('MContent.txt')
```

Step 3:

```
#filter out non-empty lines
```

```
nonemptylines = contentRDD.filter(lambda x: len(x) > 0)
```

Step 4:

```
#Split line based on space (Remember : It is mandatory to convert is in tuple} words = nonempty_lines.map(lambda x: tuple(x.split(" ", 1)))
```

```
words.saveAsSequenceFile('problem86')
```

Step 5: Check contents in directory problem86 hdfs dfs -cat problem86/part*

Step 6 : Create key, value pair (where key is null)

```
nonempty_lines.map(lambda line: (None, Mne)).saveAsSequenceFile('problem86_1')
```

Step 7 : Reading back the sequence file data using spark. seqRDD = sc.sequenceFile('problem86_1')

Step 8 : Print the content to validate the same.

```
for line in seqRDD.collect():
```

```
print(line)
```

B) Solution :

Step 1 :

```
# Import SparkContext and SparkConf
```

```
from pyspark import SparkContext, SparkConf
```

Step 2:

```
#load data from hdfs
```

```
contentRDD = sc.textFile('MContent.txt')
```

Step 3:

```
#filter out non-empty lines
```

```
nonemptylines = contentRDD.filter(lambda x: len(x) > 0)
```

Step 4:

```
#Split line based on space (Remember : It is mandatory to convert is in tuple) words = nonempty_lines.map(lambda x: tuple(x.split(" ", 1)))
```

```
words.saveAsSequenceFile('problem86')
```

Step 5 : Reading back the sequence file data using spark. `seqRDD = sc.sequenceFile('problem86_1')`

Step 6 : Print the content to validate the same.

```
for line in seqRDD.collect():
```

```
print(line)
```

Answer:

A

To Get Premium Files for CCA175 Visit

<https://www.p2pexams.com/products/cca175>

For More Free Questions Visit

<https://www.p2pexams.com/cloudera/pdf/cca175>

