



Free Questions for CCA175 by go4braindumps

Shared by Todd on 07-06-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Problem Scenario 94 : You have to run your Spark application on yarn with each executor 20GB and number of executors should be 50. Please replace XXX, YYY, ZZZ

```
export HADOOP_CONF_DIR=XXX
```

```
./bin/spark-submit \
```

```
-class com.hadoopexam.MyTask \
```

```
xxx \
```

```
-deploy-mode cluster \ # can be client for client mode
```

```
YYY \
```

```
ZZZ \
```

```
/path/to/hadoopexam.jar \
```

```
1000
```

Options:

A) Solution

XXX: -master yarn

YYY : -executor-memory 20G

ZZZ: -num-executors 50

B) Solution

XXX: -master yarn

YYY : -executor-memory 40G

ZZZ: -num-executors 80

Answer:

A

Question 2

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 operation1 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 3

Question Type: MultipleChoice

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

```
val aul = 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 4

Question Type: MultipleChoice

Problem Scenario 9 : You have been given following mysql database details as well as other info.

user=retail_dba

password=cloudera

database=retail_db

jdbc URL = jdbc:mysql://quickstart:3306/retail_db

Please accomplish following.

1. Import departments table in a directory.
2. Again import departments table same directory (However, directory already exist hence it should not override and append the results)
3. Also make sure your results fields are terminated by '|' and lines terminated by '\n'

Options:

A) Solutions :

Step 1 : Clean the hdfs file system, if they exists clean out.

```
hadoop fs -rm -R departments
```

```
hadoop fs -rm -R categories
```

```
hadoop fs -rm -R products
```

```
hadoop fs -rm -R orders
```

```
hadoop fs -rm -R order_items
```

```
hadoop fs -rm -R customers
```

Step 2 : Now import the department table as per requirement.

```
sqoop import \  
-connect jdbc:mysql://quickstart:3306/retail_db \  
--username=retail_dba \  
-password=cloudera \  
-table departments \  
-target-dir=departments \  
-fields-terminated-by '|' \  
-lines-terminated-by '\n' \  
-m1
```

Step 3 : Check imported data.

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00000
```

Step 4 : Now again import data and needs to appended.

```
sqoop import \  
-connect jdbc:mysql://quickstart:3306/retail_db \  
--username=retail_dba \  
-password=cloudera \  
-table departments \  
-target-dir departments \  
-append \  
-fields-terminated-by '|' \  
-lines-terminated-by '\n' \  
-m1
```

Step 5 : Again Check the results

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00001
```

B) Solutions :

Step 1 : Clean the hdfs file system, if they exists clean out.

```
hadoop fs -rm -R departments
```

```
hadoop fs -rm -R categories
```

```
hadoop fs -rm -R products
```

```
hadoop fs -rm -R orders
```

```
hadoop fs -rm -R order_items
```

```
hadoop fs -rm -R customers
```

Step 2 : Now import the department table as per requirement.

```
sqoop import \
```

```
-connect jdbc:mysql://quickstart:3306/retail_db \
```

```
--username=retail_dba \
```

```
-password=cloudera \
```

```
-table departments \
```

```
-target-dir=departments \
```

```
-fields-terminated-by '|' \
```

```
-lines-terminated-by '\n' \
```

```
-ml
```

Step 3 : Check imported data.

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-000
```

Step 4 : Now again import data and needs to appended.

```
sqoop import \
```

```
-connect jdbc:mysql://quickstart:3306/retail_db \
```

```
--username=retail_dba \
```



```
-password=cloudera \  
-table departments \  
-target-dir departments \  
-append \  
-fields-terminated-by '|' \  
-lines-terminated-by '\n' \  
-ml
```

Step 5 : Again Check the results

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00006
```

Answer:

A

Question 5

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 6

Question Type: MultipleChoice

Problem Scenario 9 : You have been given following mysql database details as well as other info.

```
user=retail_dba
```

password=cloudera

database=retail_db

jdbc URL = jdbc:mysql://quickstart:3306/retail_db

Please accomplish following.

1. Import departments table in a directory.
2. Again import departments table same directory (However, directory already exist hence it should not override and append the results)
3. Also make sure your results fields are terminated by '|' and lines terminated by '\n'

Options:

A) Solutions :

Step 1 : Clean the hdfs file system, if they exists clean out.

```
hadoop fs -rm -R departments
```

```
hadoop fs -rm -R categories
```

```
hadoop fs -rm -R products
```

```
hadoop fs -rm -R orders
```

```
hadoop fs -rm -R order_items
```

```
hadoop fs -rm -R customers
```

Step 2 : Now import the department table as per requirement.

```
sqoop import \
```

```
-connect jdbc:mysql://quickstart:3306/retail_db \
```

```
--username=retail_dba \  
-password=cloudera \  
-table departments \  
-target-dir=departments \  
-fields-terminated-by '|' \  
-lines-terminated-by '\n' \  
-ml
```

Step 3 : Check imported data.

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00000
```

Step 4 : Now again import data and needs to appended.

```
sqoop import \  

```

```
-connect jdbc:mysql://quickstart:3306/retail_db \  

```

```
--username=retail_dba \  

```

```
-password=cloudera \  

```

```
-table departments \  

```

```
-target-dir departments \  

```

```
-append \  

```

```
-fields-terminated-by '|' \  

```

```
-lines-terminated-by '\n' \  

```

```
-ml
```

Step 5 : Again Check the results

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00001
```

B) Solutions :

Step 1 : Clean the hdfs file system, if they exists clean out.

```
hadoop fs -rm -R departments
```

```
hadoop fs -rm -R categories
```

```
hadoop fs -rm -R products
```

```
hadoop fs -rm -R orders
```

```
hadoop fs -rm -R order_items
```

```
hadoop fs -rm -R customers
```

Step 2 : Now import the department table as per requirement.

```
sqoop import \
```

```
-connect jdbc:mysql://quickstart:3306/retail_db \
```

```
--username=retail_dba \
```

```
-password=cloudera \
```

```
-table departments \
```

```
-target-dir=departments \
```

```
-fields-terminated-by '|' \
```

```
-lines-terminated-by '\n' \
```

```
-ml
```

Step 3 : Check imported data.

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-000
```

Step 4 : Now again import data and needs to appended.

```
sqoop import \
```

```
-connect jdbc:mysql://quickstart:3306/retail_db \
```

```
--username=retail_dba \
```

```
-password=cloudera \
```

```
-table departments \
```

```
-target-dir departments \  
-append \  
-fields-terminated-by '|' \  
-lines-terminated-by '\n' \  
-ml
```

Step 5 : Again Check the results

```
hdfs dfs -ls departments
```

```
hdfs dfs -cat departments/part-m-00006
```

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>

