CCA175 Q&As

CCA Spark and Hadoop Developer Exam

Pass Cloudera CCA175 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass2lead.com/cca175.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Cloudera Official Exam Center

QUESTION 1

Problem Scenario 76 : You have been given MySQL DB with following details. user=retail_dba password=cloudera database=retail_db table=retail_db.orders table=retail_db.order_items jdbc URL = jdbc:mysql://quickstart:3306/retail_db Columns of order table : (orderid , order_date , ordercustomerid, order_status} ..... Please accomplish following activities.

1.

Copy "retail_db.orders" table to hdfs in a directory p91_orders.

2.

Once data is copied to hdfs, using pyspark calculate the number of order for each status.

3.

Use all the following methods to calculate the number of order for each status. (You need to know all these functions and its behavior for real exam)

-countByKey() -groupByKey()

-reduceByKey() -aggregateByKey()

-combineByKey()

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Import Single table

sqoop import --connect jdbc:mysql://quickstart:3306/retail_db --username=retail dba password=cloudera -table=orders --target-dir=p91_orders

Note : Please check you dont have space between before or after \\'=\\' sign. Sqoop uses the

MapReduce framework to copy data from RDBMS to hdfs

Step 2 : Read the data from one of the partition, created using above command, hadoop fs

-cat p91_orders/part-m-00000

Step 3: countByKey #Number of orders by status allOrders = sc.textFile("p91_orders")

#Generate key and value pairs (key is order status and vale as an empty string keyValue =

aIIOrders.map(lambda line: (line.split(",")[3], ""))

#Using countByKey, aggregate data based on status as a key

output=keyValue.countByKey()Jtems()

for line in output: print(line)

Step 4 : groupByKey

#Generate key and value pairs (key is order status and vale as an one

keyValue = allOrders.map(lambda line: (line.split)",")[3], 1))

#Using countByKey, aggregate data based on status as a key output=

keyValue.groupByKey().map(lambda kv: (kv[0], sum(kv[1]}}}

tor line in output.collect(): print(line}

Step 5 : reduceByKey

#Generate key and value pairs (key is order status and vale as an one

keyValue = allOrders.map(lambda line: (line.split(","}[3], 1))

#Using countByKey, aggregate data based on status as a key output=

keyValue.reduceByKey(lambda a, b: a + b)

tor line in output.collect(): print(line}

Step 6: aggregateByKey

#Generate key and value pairs (key is order status and vale as an one keyValue =

allOrders.map(lambda line: (line.split(",")[3], line}}

output=keyValue.aggregateByKey(0, lambda a, b: a+1, lambda a, b: a+b}

for line in output.collect(): print(line}

Step 7 : combineByKey

#Generate key and value pairs (key is order status and vale as an one

keyValue = allOrders.map(lambda line: (line.split(",")[3], line))

output=keyValue.combineByKey(lambda value: 1, lambda ace, value: acc+1, lambda ace,

value: acc+value)

tor line in output.collect(): print(line)

#Watch Spark Professional Training provided by www.ABCTECH.com to understand more

on each above functions. (These are very important functions for real exam)

QUESTION 2

Problem Scenario 52 : You have been given below code snippet.

val b = sc.parallelize(List(1,2,3,4,5,6,7,8,2,4,2,1,1,1,1,1))

Operation_xyz

Write a correct code snippet for Operation_xyz which will produce below output.

scalaxollection.Map[lnt,Long] = Map(5 -> 1, 8 -> 1, 3 -> 1, 6 -> 1, 1 -> S, 2 -> 3, 4 -> 2, 7 ->

1)

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution : b.countByValue countByValue Returns a map that contains all unique values of the RDD and their respective occurrence counts. (Warning: This operation will finally aggregate the information in a single reducer.) Listing Variants def countByValue(): Map[T, Long]

QUESTION 3

Problem Scenario 26 : You need to implement near real time solutions for collecting information when submitted in file with below information. You have been given below directory location (if not available than create it) /tmp/nrtcontent. Assume your departments upstream service is continuously committing data in this directory as a new file (not stream of data, because it is near real time solution). As soon as file committed in this directory that needs to be available in hdfs in /tmp/flume location Data

echo "I am preparing for CCA175 from ABCTECH.com" > /tmp/nrtcontent/.he1.txt mv /tmp/nrtcontent/.he1.txt /tmp/nrtcontent/he1.txt After few mins echo "I am preparing for CCA175 from TopTech.com" > /tmp/nrtcontent/.qt1.txt mv /tmp/nrtcontent/.qt1.txt /tmp/nrtcontent/qt1.txt

Write a flume configuration file named flumes.conf and use it to load data in hdfs with following additional properties.

1.

Spool /tmp/nrtcontent

2.

File prefix in hdfs sholuld be events

3.

File suffix should be Jog

4.

If file is not commited and in use than it should have as prefix.

5.

Data should be written as text to hdfs

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution : Step 1 : Create directory mkdir /tmp/nrtcontent Step 2 : Create flume configuration file, with below configuration for source, sink and channel and save it in flume6.conf. agent1 .sources = source1 agent1 .sinks = sink1 agent1.channels = channel1 agent1 .sources.source1.channels = channel1 agent1 .sinks.sink1.channel = channel1 agent1 .sources.source1.type = spooldir agent1 .sources.source1.spoolDir = /tmp/nrtcontent agent1 .sinks.sink1 .type = hdfs agent1 .sinks.sink1.hdfs.path = /tmp/flume agent1.sinks.sink1.hdfs.filePrefix = events

agent1.sinks.sink1.hdfs.fileSuffix = .log agent1 .sinks.sink1.hdfs.inUsePrefix = _ agent1 .sinks.sink1.hdfs.fileType = Data Stream Step 4 : Run below command which will use this configuration file and append data in hdfs. Start flume service: flume-ng agent -conf /home/cloudera/flumeconf -conf-file /home/cloudera/flumeconf/flume6.conf --name agent1 Step 5 : Open another terminal and create a file in /tmp/nrtcontent echo "I am preparing for CCA175 from ABCTechm.com" > /tmp/nrtcontent/.he1.txt mv /tmp/nrtcontent/.he1.txt /tmp/nrtcontent/he1.txt After few mins echo "I am preparing for CCA175 from TopTech.com" > /tmp/nrtcontent/.qt1.txt mv /tmp/nrtcontent/.qt1.txt /tmp/nrtcontent/qt1.txt

QUESTION 4

Problem Scenario 56 : You have been given below code snippet.

val a = sc.parallelize(l to 100. 3)

operation1

Write a correct code snippet for operationl which will produce desired output, shown below.

Array [Array [I nt]] = Array(Array(1, 2, 3,4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19, 20,

21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33),

Array(34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55,

56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66),

Array(67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88,

89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100))

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution : a.glom.collect glom Assembles an array that contains all elements of the partition and embeds it in an RDD. Each returned array contains the contents of one panition

QUESTION 5

Problem Scenario 25 : You have been given below comma separated employee information. That needs to be added in /home/cloudera/flumetest/in.txt file (to do tail source) sex,name,city 1,alok,mumbai 1,jatin,chennai 1,yogesh,kolkata 2,ragini,delhi 2,jyotsana,pune 1,valmiki,banglore Create a flume conf file using fastest non-durable channel, which write data in hive warehouse directory, in two separate tables called flumemaleemployee1 and flumefemaleemployee1 (Create hive table as well for given data}. Please use tail source with /home/cloudera/flumetest/in.txt file. Flumemaleemployee1 : will contain only male employees data flumefemaleemployee1 : Will contain only woman employees data

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Create hive table for flumemaleemployeel and .\\'

CREATE TABLE flumemaleemployeel

(

sex_type int, name string, city string) ROW FORMAT DELIMITED FIELDS TERMINATED BY \\',\\'; CREATE TABLE flumefemaleemployeel (sex_type int, name string, city string) ROW FORMAT DELIMITED FIELDS TERMINATED BY \\',\\'; Step 2 : Create below directory and file mkdir /home/cloudera/flumetest/ cd /home/cloudera/flumetest/ Step 3 : Create flume configuration file, with below configuration for source, sink and channel and save it in flume5.conf. agent.sources = tailsrc agent.channels = mem1 mem2 agent.sinks = stdl std2 agent.sources.tailsrc.type = exec agent.sources.tailsrc.command = tail -F /home/cloudera/flumetest/in.txt agent.sources.tailsrc.batchSize = 1 agent.sources.tailsrc.interceptors = i1 agent.sources.tailsrc.interceptors.i1.type = regex_extractor agent.sources.tailsrc.interceptors.il.regex = A(\\d} agent.sources.tailsrc. interceptors. M.serializers = t1 agent.sources.tailsrc. interceptors, i1.serializers.t1. name = type agent.sources.tailsrc.selector.type = multiplexing agent.sources.tailsrc.selector.header = type agent.sources.tailsrc.selector.mapping.1 = memi agent.sources.tailsrc.selector.mapping.2 = mem2 agent.sinks.std1.type = hdfs agent.sinks.stdl.channel = mem1 agent.sinks.stdl.batchSize = 1 agent.sinks.std1.hdfs.path = /user/hive/warehouse/flumemaleemployeei agent.sinks.stdl.rolllnterval = 0 agent.sinks.stdl.hdfs.tileType = Data Stream agent.sinks.std2.type = hdfs agent.sinks.std2.channel = mem2 agent.sinks.std2.batchSize = 1 agent.sinks.std2.hdfs.path = /user/hi ve/warehouse/flumefemaleemployee1 agent.sinks.std2.rolllnterval = 0 agent.sinks.std2.hdfs.tileType = Data Stream agent.channels.mem1.type = memory agent.channels.meml.capacity = 100 agent.channels.mem2.type = memory agent.channels.mem2.capacity = 100 agent.sources.tailsrc.channels = mem1 mem2 Step 4 : Run below command which will use this configuration file and append data in hdfs. Start flume service: flume-ng agent -conf /home/cloudera/flumeconf -conf-file /home/cloudera/flumeconf/flume5.conf --name agent Step 5 : Open another terminal create a file at /home/cloudera/flumetest/in.txt. Step 6 : Enter below data in file and save it. l.alok.mumbai 1 jatin.chennai 1,yogesh,kolkata 2,ragini,delhi 2,jyotsana,pune 1,valmiki,banglore Step 7 : Open hue and check the data is available in hive table or not. Step 8 : Stop flume service by pressing ctrl+c

QUESTION 6

Problem Scenario 61 : You have been given below code snippet. val a = sc.parallelize(List("dog", "salmon", "salmon", "rat", "elephant"), 3)

val b = a.keyBy(_.length)

val c = sc.parallelize(List("dog","cat","gnu","salmon","rabbit","turkey","wolf","bear","bee"), 3)

val d = c.keyBy(_.length) operationl

Write a correct code snippet for operationl which will produce desired output, shown below.

Array[(lnt, (String, Option[String]}}] = Array((6,(salmon,Some(salmon))),

(6,(salmon,Some(rabbit))),

(6,(salmon,Some(turkey))), (6,(salmon,Some(salmon))), (6,(salmon,Some(rabbit))),

(6,(salmon,Some(turkey))), (3,(dog,Some(dog))), (3,(dog,Some(cat))),

(3,(dog,Some(dog))), (3,(dog,Some(bee))), (3,(rat,Some(dogg)), (3,(rat,Some(cat)j),

(3,(rat.Some(gnu))). (3,(rat,Some(bee))), (8,(elephant,None)))

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution : b.leftOuterJoin(d}.collect leftOuterJoin [Pair]: Performs an left outer join using two key-value RDDs. Please note that the keys must be generally comparable to make this work keyBy : Constructs twocomponent tuples (key-value pairs) by applying a function on each data item. Trie result of the function becomes the key and the original data item becomes the value of the newly created tuples.

QUESTION 7

Problem Scenario 11 : 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 called departments.

2.

Once import is done, please insert following 5 records in departments mysql table.

Insert into departments(10, physics);

Insert into departments(11, Chemistry);

Insert into departments(12, Maths);

Insert into departments(13, Science);

Insert into departments(14, Engineering);

3.

Now import only new inserted records and append to existring directory . which has been created in first step.

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Clean already imported data. (In real exam, please make sure you dont delete

data generated from previous exercise).

hadoop fs -rm -R departments

Step 2 : Import data in departments directory.

sqoop import \

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

--username=retail_dba \

-password=cloudera \

-table departments \

"target-dir/user/cloudera/departments

Step 3 : Insert the five records in departments table.

mysql -user=retail_dba --password=cloudera retail_db

Insert into departments values(10, "physics"); Insert into departments values(11,

"Chemistry"); Insert into departments values(12, "Maths"); Insert into departments

values(13, "Science"); Insert into departments values(14, "Engineering"); commit;

select\\' from departments;

Step 4 : Get the maximum value of departments from last import, hdfs dfs -cat

/user/cloudera/departments/part* that should be 7

Step 5 : Do the incremental import based on last import and append the results.

sqoop import \

--connect "jdbc:mysql://quickstart.cloudera:330G/retail_db" \

~username=retail_dba \

-password=cloudera \

-table departments \

--target-dir /user/cloudera/departments \

-append \

-check-column "department_id" \

-incremental append \

-last-value 7

Step 6 : Now check the result.

hdfs dfs -cat /user/cloudera/departments/part"

QUESTION 8

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 doen store the filtered data in a directory called "problem84" (On hdfs) Content.txt Hello this is ABCTECH.com This is ABYTECH.com Apache Spark TrainingThis is Spark Learning Session Spark is faster than MapReduce

Correct Answer: See the explanation for Step by Step Solution and configuration.

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

QUESTION 9

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

val a = sc.parallelize(List("dog", "tiger", "lion", "cat", "spider", "eagle"), 2)

val b = a.keyBy(_.length)

val c = sc.parallelize(List("ant", "falcon", "squid"), 2)

val d = c.keyBy(.length)

operation 1

Write a correct code snippet for operationl which will produce desired output, shown below.

Array[(lnt, String)] = Array((4,lion))

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution : b.subtractByKey(d).collect subtractByKey [Pair] : Very similar to subtract, but instead of supplying a function, the keycomponent of each pair will be automatically used as criterion for removing items from the first RDD.

QUESTION 10

Problem Scenario 58 : You have been given below code snippet.

val a = sc.parallelize(List("dog", "tiger", "lion", "cat", "spider", "eagle"), 2) val b =

a.keyBy(_.length)

operation1

Write a correct code snippet for operationl which will produce desired output, shown below.

Array[(lnt, Seq[String])] = Array((4,ArrayBuffer(lion)), (6,ArrayBuffer(spider)),

(3,ArrayBuffer(dog, cat)), (5,ArrayBuffer(tiger, eagle}}}

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

b.groupByKey.collect

groupByKey [Pair]

Very similar to groupBy, but instead of supplying a function, the key-component of each

pair will automatically be presented to the partitioner.

Listing Variants

def groupByKeyQ: RDD[(K, lterable[V]}]

def groupByKey(numPartittons: Int): RDD[(K, lterable[V])]

def groupByKey(partitioner: Partitioner): RDD[(K, lterable[V])]

QUESTION 11

Problem Scenario 45 : You have been given 2 files , with the content as given Below (spark12/technology.txt) (spark12/salary.txt) (spark12/technology.txt) first,last,technology Amit,Jain,java Lokesh,kumar,unix Mithun,kale,spark Rajni,vekat,hadoop Rahul,Yadav,scala (spark12/salary.txt) first,last,salary Amit,Jain,100000 Lokesh,kumar,95000 Mithun,kale,150000 Rajni,vekat,154000 Rahul,Yadav,120000 Write a Spark program, which will join the data based on first and last name and save the joined results in following format, first Last.technology.salary

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Create 2 files first using Hue in hdfs.

Step 2 : Load all file as an RDD

val technology = sc.textFile(Msparkl2/technology.txt").map(e => e.splitf\\',"))

val salary = sc.textFile("spark12/salary.txt").map(e => e.split("."))

Step 3 : Now create Key.value pair of data and join them.

val joined = technology.map(e=>((e(0),e(1)),e(2))).join(salary.map(e=>((e(0),e(1)),e(2))))

Step 4 : Save the results in a text file as below.

joined.repartition(1).saveAsTextFile("spark12/multiColumn Joined.txt")

QUESTION 12

Problem Scenario 38 : You have been given an RDD as below,

val rdd: RDD[Array[Byte]]

Now you have to save this RDD as a SequenceFile. And below is the code snippet.

import org.apache.hadoop.io.compress.GzipCodec

rdd.map(bytesArray => (A.get(), new B(bytesArray))).saveAsSequenceFile(\\'7output/path",classOt[GzipCodec]) What would be the correct replacement for A and B in above snippet.

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

A. NullWritable

B. BytesWritable

QUESTION 13

Problem Scenario 48 : You have been given below Python code snippet, with intermediate

output.

We want to take a list of records about people and then we want to sum up their ages and

count them.

So for this example the type in the RDD will be a Dictionary in the format of {name: NAME,

age:AGE, gender:GENDER}.

The result type will be a tuple that looks like so (Sum of Ages, Count)

people = []

people.append({\\'name\\':\\'Amit\\', \\'age\\':45,\\'gender\\':\\'M\\'})

people.append({\\'name\\':\\'Ganga\\', \\'age\\':43,\\'gender\\':\\'F\\'})

people.append({\\'name\\':\\'John\\', \\'age\\':28,\\'gender\\':\\'M\\'})

people.append({\\'name\\':\\'Lolita\\', \\'age\\':33,\\'gender\\':\\'F\\'})

people.append({\\'name\\':\\'Dont Know\\', \\'age\\':18,\\'gender\\':\\'T\\'})

peopleRdd=sc.parallelize(people) //Create an RDD

peopleRdd.aggregate((0,0), seqOp, combOp) //Output of above line : 167, 5)

Now define two operation seqOp and combOp , such that

seqOp : Sum the age of all people as well count them, in each partition. combOp :

Combine results from all partitions.

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

seqOp = (lambda x,y: (x[0] + y[\\'age\\'],x[1] + 1))

combOp = (lambda x,y: (x[0] + y[0], x[1] + y[1]))

QUESTION 14

Problem Scenario 18 : 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 Now accomplish following activities.

1.

Create mysql table as below.

mysql --user=retail_dba -password=cloudera

use retail_db

CREATE TABLE IF NOT EXISTS departments_hive02(id int, department_name

varchar(45), avg_salary int);

show tables;

2.

Now export data from hive table departments_hive01 in departments_hive02. While

exporting, please note following. wherever there is a empty string it should be loaded as a null value in

mysql.

wherever there is -999 value for int field, it should be created as null value.

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

Step 1 : Create table in mysql db as well.

mysql ~user=retail_dba -password=cloudera

use retail_db

CREATE TABLE IF NOT EXISTS departments_hive02(id int, department_name

varchar(45), avg_salary int);

show tables;

Step 2 : Now export data from hive table to mysql table as per the requirement.

sqoop export --connect jdbc:mysql://quickstart:3306/retail_db \

-username retaildba \

-password cloudera \

--table departments_hive02 \

-export-dir /user/hive/warehouse/departments_hive01 \

-input-fields-terminated-by \\'\001\\' \

--input-lines-terminated-by \\'\n\\' \

--num-mappers 1 \

-batch \

-Input-null-string "" \

-input-null-non-string -999

step 3 : Now validate the data,select * from departments_hive02;

QUESTION 15

Problem Scenario 51 : You have been given below code snippet.

val a = sc.parallelize(List(1, 2,1, 3), 1)

val b = a.map((_, "b"))

val c = a.map((_, "c"))

Operation_xyz

Write a correct code snippet for Operationxyz which will produce below output.

Output:

Array[(Int, (Iterable[String], Iterable[String]))] = Array(

(2,(ArrayBuffer(b),ArrayBuffer(c))),

(3,(ArrayBuffer(b),ArrayBuffer(c))),

(1,(ArrayBuffer(b, b),ArrayBuffer(c, c))))

Correct Answer: See the explanation for Step by Step Solution and configuration.

Solution :

b.cogroup(c).collect

cogroup [Pair], groupWith [Pair]

A very powerful set of functions that allow grouping up to 3 key-value RDDs together using

their keys.

Another example

val x = sc.parallelize(List((1, "apple"), (2, "banana"), (3, "orange"), (4, "kiwi")), 2)

val y = sc.parallelize(List((5, "computer"), (1, "laptop"), (1, "desktop"), (4, "iPad")), 2)

x.cogroup(y).collect

Array[(Int, (Iterable[String], Iterable[String]))] = Array(

(4,(ArrayBuffer(kiwi),ArrayBuffer(iPad))),

(2,(ArrayBuffer(banana),ArrayBuffer())),

(3,(ArrayBuffer(orange),ArrayBuffer())),

(1 ,(ArrayBuffer(apple),ArrayBuffer(laptop, desktop))),

(5,{ArrayBuffer(),ArrayBuffer(computer))))