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.leads4pass.com/cca175.html

100% Passing Guarantee
100% Money Back Assurance

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

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

Problem Scenario 65 : You have been given below code snippet.

val a = sc.parallelize(List("dog", "cat", "owl", "gnu", "ant"), 2)

val b = sc.parallelize(1 to a.count.tolnt, 2)

val c = a.zip(b)

operation1

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

Array[(String, Int)] = Array((owl,3), (gnu,4), (dog,1), (cat,2>, (ant,5))

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

Solution : c.sortByKey(false).collect

sortByKey [Ordered] : This function sorts the input RDD\\'s data and stores it in a new RDD.

"The output RDD is a shuffled RDD because it stores data that is output by a reducer

which has been shuffled. The implementation of this function is actually very clever.

First, it uses a range partitioner to partition the data in ranges within the shuffled RDD.

Then it sorts these ranges individually with mapPartitions using standard sort mechanisms.

QUESTION 2

Problem Scenario 90 : You have been given below two files course.txt id,course 1,Hadoop 2,Spark 3,HBase fee.txt id,fee 2,3900 3,4200 4,2900 Accomplish the following activities.

1.

Select all the courses and their fees , whether fee is listed or not.

2.

Select all the available fees and respective course. If course does not exists still list the fee

3.

Select all the courses and their fees , whether fee is listed or not. However, ignore records having fee as null.

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

Solution :

Step 1:

hdfs dfs -mkdir sparksql4

hdfs dfs -put course.txt sparksql4/

hdfs dfs -put fee.txt sparksql4/

Step 2 : Now in spark shell

// load the data into a new RDD

val course = sc.textFile("sparksql4/course.txt")

val fee = sc.textFile("sparksql4/fee.txt")

// Return the first element in this RDD

course.fi rst()

fee.fi rst()

//define the schema using a case class case class Course(id: Integer, name: String) case

class Fee(id: Integer, fee: Integer)

// create an RDD of Product objects

val courseRDD = course.map(_.split(",")).map(c => Course(c(0).tolnt,c(1)))

val feeRDD =fee.map(_.split(",")).map(c => Fee(c(0}.tolnt,c(1}.tolnt))

courseRDD.first()

courseRDD.count{}

feeRDD.first()

feeRDD.countQ

// change RDD of Product objects to a DataFrame val courseDF = courseRDD.toDF{} val

feeDF = feeRDD.toDF{}

// register the DataFrame as a temp table courseDF. registerTempTable("course") feeDF.

registerTempTablef\\'fee")

// Select data from table

val results = sqlContext.sql(.....SELECT\\' FROM course """)

results. showQ

val results = sqlContext.sql(.....SELECT\\' FROM fee.....)

results. showQ

val results = sqlContext.sql(.....SELECT * FROM course LEFT JOIN fee ON course.id =

fee.id.....)

results-showQ

val results ="sqlContext.sql(.....SELECT * FROM course RIGHT JOIN fee ON course.id =

fee.id "MM)

results. showQ

val results = sqlContext.sql(.....SELECT\\' FROM course LEFT JOIN fee ON course.id =

fee.id where fee.id IS NULL"

results. show()

QUESTION 3

Problem Scenario 34 : You have given a file named spark6/user.csv. Data is given below: user.csv id,topic,hits Rahul,scala,120 Nikita,spark,80 Mithun,spark,1 myself,cca175,180 Now write a Spark code in scala which will remove the header part and create RDD of values as below, for all rows. And also if id is myself" than filter out row. Map(id -> om, topic -> scala, hits -> 120)

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

Solution :

Step 1 : Create file in hdfs (We will do using Hue). However, you can first create in local

filesystem and then upload it to hdfs.

Step 2 : Load user.csv file from hdfs and create PairRDDs val csv =

sc.textFile("spark6/user.csv")

Step 3 : split and clean data

val headerAndRows = csv.map(line => line.split(",").map(_.trim))

Step 4 : Get header row

val header = headerAndRows.first

Step 5 : Filter out header (We need to check if the first val matches the first header name)

val data = headerAndRows.filter(_(0) != header(O))

Step 6 : Splits to map (header/value pairs)

val maps = data.map(splits => header.zip(splits).toMap)

step 7: Filter out the user "myself

val result = maps.filter(map => mapf\\'id") != "myself")

Step 8 : Save the output as a Text file. result.saveAsTextFile("spark6/result.txt")

QUESTION 4

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\

222 \

/path/to/hadoopexam.jar \

1000

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

Solution

XXX: -master yarn YYY : -executor-memory 20G ZZZ: -num-executors 50

QUESTION 5

Problem Scenario 72 : You have been given a table named "employee2" with following detail. first_name string last_name string Write a spark script in python which read this table and print all the rows and individual column values.

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

Solution :

Step 1 : Import statements for HiveContext from pyspark.sql import HiveContext

Step 2 : Create sqlContext sqlContext = HiveContext(sc)

Step 3 : Query hive

employee2 = sqlContext.sql("select\\' from employee2")

Step 4 : Now prints the data for row in employee2.collect(): print(row)

Step 5 : Print specific column for row in employee2.collect(): print(row.fi rst_name)

Latest CCA175 Dumps CCA175 Study Guide CCA175 Exam Questions