Cloudera

Exam CCA175

CCA Spark and Hadoop Developer Exam

Verson: Demo

[Total Questions: 10]

Question No : 1 CORRECT TEXT

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.

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

Explanation:

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 No : 2 CORRECT TEXT

Problem Scenario 32 : You have given three files as below.

spark3/sparkdir1/file1.txt

spark3/sparkd ir2ffile2.txt

spark3/sparkd ir3Zfile3.txt

Each file contain some text.

spark3/sparkdir1/file1.txt

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework

spark3/sparkdir2/file2.txt

The core of Apache Hadoop consists of a storage part known as Hadoop Distributed File System (HDFS) and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers packaged code for nodes to process in parallel based on the data that needs to be processed.

spark3/sparkdir3/file3.txt

his approach takes advantage of data locality nodes manipulating the data they have access to to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking

Now write a Spark code in scala which will load all these three files from hdfs and do the word count by filtering following words. And result should be sorted by word count in reverse order.

Filter words ("a","the","an", "as", "a","with","this","these","is","are","in", "for", "to","and","The","of")

Also please make sure you load all three files as a Single RDD (All three files must be loaded using single API call).

You have also been given following codec

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

Please use above codec to compress file, while saving in hdfs.

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

Explanation:

Solution **:**

Step 1 : Create all **three files in hdfs (We will do using Hue). However, you** can **first create in local filesystem and then upload it to hdfs.**

Step 2 : Load content from all files.

val **content** = **sc.textFile("spark3/sparkdir1/file1.txt,spark3/sparkdir2/file2.txt,spark3/sparkdir3/file3.txt")** //Load **the text file**

Step 3 : **Now create** split each line and **create** RDD of words.

val flatContent = content.flatMap(word=>word.split(" "))

step 4 : Remove space after each word (trim it)

val trimmedContent = f1atContent.map(word=>word.trim)

Step 5 : Create an RDD from remove, all the words that needs to be removed.

val removeRDD = sc.parallelize(List("a","theM,ManM, "as",

"a","with","this","these","is","are\"in\" "for", "to","and","The","of"))

Step 6 : Filter the RDD, so it can have only content which are not present in removeRDD.

val filtered = trimmedContent.subtract(removeRDD}

Step 7 : Create a PairRDD, so we can have (word,1) tuple or PairRDD. val pairRDD = filtered.map(word => (word,1))

Step 8 : Now do the word count on PairRDD. val wordCount = pairRDD.reduceByKey(_ + _)

Step 9 : Now swap PairRDD.

val swapped = wordCount.map(item => item.swap)

Step 10 : Now revers order the content. val sortedOutput = swapped.sortByKey(false)

Step 11 : Save the output as a Text file. sortedOutput.saveAsTextFile("spark3/result")

Step 12 : Save compressed output.
import org.apache.hadoop.io.compress.GzipCodec
sortedOutput.saveAsTextFile("spark3/compressedresult", classOf[GzipCodec])

Question No : 3 CORRECT TEXT

Problem Scenario 19 : 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. Import departments **table** from mysql to hdfs as textfile in departments_text directory.

2. Import departments table from mysql to hdfs as sequncefile in departments_sequence directory.

3. Import departments table from mysql to hdfs as avro file in departments avro directory.

4. Import departments table from mysql to hdfs as parquet file in departments_parquet directory.

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

Explanation:

Solution :

Step 1 : Import departments table from mysql to hdfs as textfile
sqoop import \
-connect jdbc:mysql://quickstart:3306/retail_db \
~username=retail_dba \
-password=cloudera \
-table departments \
-as-textfile \
-target-dir=departments_text
verify imported data
hdfs dfs -cat departments_text/part"

Step 2 : Import departments table from mysql to hdfs as sequncetlle
sqoop import \
-connect jdbc:mysql://quickstart:330G/retaiI_db \
~username=retail_dba \
-password=cloudera \
--table departments \
-as-sequencetlle \
-~target-dir=departments sequence
verify imported data
hdfs dfs -cat departments_sequence/part*

Step 3 : Import departments table from mysql to hdfs as sequncetlle
sqoop import \
-connect jdbc:mysql://quickstart:330G/retaiI_db \
~username=retail_dba \
--password=cloudera \
--table departments \
--as-avrodatafile \
--target-dir=departments_avro
verify imported data
hdfs dfs -cat departments avro/part*

Step 4 : Import departments table from mysql to hdfs as sequncetlle
sqoop import \
-connect jdbc:mysql://quickstart:330G/retaiI_db \
~username=retail_dba \
--password=cloudera \
-table departments \
-as-parquetfile \

-target-dir=departments_parquet
verify imported data
hdfs dfs -cat departmentsparquet/part*

Question No : 4 CORRECT TEXT

Problem Scenario 62 : You have been given below code snippet.

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

val b = a.map(x => (x.length, x))

operation1

Write a correct code snippet for operationl which will produce desired output, shown below.
Array[(lnt, String)] = Array((3,xdogx), (5,xtigerx), (4,xlionx), (3,xcatx), (7,xpantherx), (5,xeaglex))

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

b.mapValuesf'x" + _ + "x").collect

mapValues [Pair] : Takes the values of a RDD that consists of two-component tuples, and applies the provided function to transform each value. Tlien,.it.forms newtwo-componend tuples using the key and the transformed value and stores them in a new RDD.

Question No : 5 CORRECT TEXT

Problem Scenario 46 : You have been given belwo list in scala (name,sex,cost) for each work done.

List(("Deeapak" , "male", 4000), ("Deepak" , "male", 2000), ("Deepika" , "female", 2000),("Deepak" , "female", 2000), ("Deepak" , "male", 1000) , ("Neeta" , "female", 2000))

Now write a Spark program to load this list as an RDD and do the sum of cost for combination of name and sex (as key)

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

Explanation:

Solution :

Step 1 : Create an RDD out of this list

val rdd = sc.parallelize(List(("Deeapak" , "male", 4000}, ("Deepak" , "male", 2000), ("Deepika" , "female", 2000),("Deepak" , "female", 2000), ("Deepak" , "male", 1000} , ("Neeta" , "female", 2000}}}

Step 2 : Convert this RDD in pair RDD

val byKey = rdd.map({case (name,sex,cost) => (name,sex)->cost})

Step 3 : Now group by Key

val byKeyGrouped = byKey.groupByKey

Step 4 : Nowsum the cost for each group

val result = byKeyGrouped.map{case ((id1,id2),values) => (id1,id2,values.sum)}

Step 5 : Save the results result.repartition(1).saveAsTextFile("spark12/result.txt")

Question No : 6 CORRECT TEXT

Problem Scenario 17 : 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 below assignment.

1. Create a table in hive as below, create table departments_hiveOl(department_id int, department_name string, avg_salary int);

2. Create another table in mysql using below statement CREATE TABLE IF NOT EXISTS departments_hive01(id int, department_name varchar(45), avg_salary int);

3. Copy all the data from departments table to departments_hive01 using insert into departments_hive01 select a.*, null from departments a;

Also insert following records as below

insert into departments_hive01 values(777, "Not known",1000);

insert into departments_hive01 values(8888, null,1000);

insert into departments_hive01 values(666, null,1100);

4. Now import data from mysql table departments_hive01 to this hive table. Please make sure that data should be visible using below hive command. Also, while importing if null value found for department_name column replace it with "" (empty string) and for id column with -999 select * from departments_hive;

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

Explanation:

Solution :

Step 1 : Create hive table as below.

hive

show tables;

create table departments_hive01(department_id int, department_name string, avgsalary int);

Step 2 : Create table in mysql db as well.

mysql -user=retail_dba -password=cloudera

use retail_db

CREATE TABLE IF NOT EXISTS departments_hive01(id int, department_name varchar(45), avg_salary int);

show tables;

step 3 : Insert data in mysql table.
insert into departments_hive01 select a.*, null from departments a;
check data inserts
select' from departments_hive01;

Now iserts null records as given in problem. insert into departments_hive01 **values(777,** "Not known",1000); insert into departments_hive01 values(8888, null,1000); insert into departments_hive01 values(666, null,1100);

Step 4 : Now import data in hive as per requirement.
sqoop import \
-connect jdbc:mysql://quickstart:3306/retail_db \
~username=retail_dba \
--password=cloudera \
-table departments_hive01 \
--hive-home /user/hive/warehouse \
--hive-import \
-hive-overwrite \
-hive-table departments_hive0l \
--fields-terminated-by '\001' \
--null-string M"\
--null-non-strlng -999 \
-split-by id \
-m 1

Step 5 : Checkthe data in directory.
hdfs dfs -Is /user/hive/warehouse/departments_hive01
hdfs dfs -cat/user/hive/warehouse/departments_hive01/part"
Check **data in** hive **table.**
Select * from departments_hive01;

Question No : 7 CORRECT TEXT

Problem Scenario 6 : 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

Compression Codec : org.apache.hadoop.io.compress.SnappyCodec

Please accomplish following.

1. Import entire database such that it can be used as a hive tables, it must be created in default schema.

2. Also make sure each tables file is partitioned in 3 files e.g. **part-00000, part-00002, part-00003**

3. **Store** all the Java files in a directory called java_output to evalute the further

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

Explanation:

Solution :

Step 1 : Drop all the tables, which we have created in previous problems. Before implementing the solution.

Login to hive and execute following command.

show tables;

drop table categories;
drop table customers;
drop table departments;
drop table employee;
drop table ordeMtems;
drop table orders;
drop table products;

show tables;

Check warehouse directory. hdfs dfs -Is /user/hive/warehouse

Step 2 : Now we have cleaned database. Import entire retail db with all the required parameters as problem statement is asking.

```
sqoop import-all-tables \
-m3\
-connect jdbc:mysql://quickstart:3306/retail_db \
--username=retail_dba \
-password=cloudera \
-hive-import \
--hive-overwrite \
-create-hive-table \
--compress \
--compression-codec org.apache.hadoop.io.compress.SnappyCodec \
--outdir java_output
```

Step 3 : Verify the work is accomplished or not.

a. Go to hive and check all the tables hive
show tables;
select count(1) from customers;

b. Check the-warehouse directory and number of partitions,
hdfs dfs -Is /user/hive/warehouse
hdfs dfs -Is /user/hive/warehouse/categories

c. Check the output Java directory.
Is -Itr java_output/

Question No : 8 CORRECT TEXT

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.

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

Explanation:

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 No : 9 CORRECT TEXT

Problem Scenario 7 : 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 department tables using your custom boundary query, which import departments between 1 to 25.

2. Also make sure each tables file is partitioned in 2 files e.g. part-00000, part-00002

3. Also make sure you have imported only two columns from table, which are department_id,department_name

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

Explanation:

Solutions :

Step 1 : Clean the hdfs tile 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_itmes
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 /user/cloudera/departments \
-m2\
-boundary-query "select 1, 25 from departments" \
-columns department_id,department_name
```

Step 3 : Check imported data.

```
hdfs dfs -Is departments
hdfs dfs -cat departments/part-m-00000
hdfs dfs -cat departments/part-m-00001
```

Question No : 10 CORRECT TEXT

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.

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

Explanation:

Solution :

A. NullWritable

B. BytesWritable