Cloudera

Exam CCA175

CCA Spark and Hadoop Developer Exam

Verson: Demo

[Total Questions: 10]

Question No : 1 CORRECT TEXT

Problem Scenario 75 : 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

Please accomplish following activities.

1. Copy "retail_db.order_items" table to hdfs in respective directory p90_order_items .

2. Do the summation of entire revenue in this table using pyspark.

3. Find the maximum and minimum revenue as well.

4. Calculate average revenue

Columns of ordeMtems table : (order_item_id , order_item_order_id , order_item_product_id, order_item_quantity,order_item_subtotal,order_item_subtotal,order_item_product_price)

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

Explanation:

Solution :

Step 1 : Import Single table .

sqoop import --connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba -password=cloudera -table=order_items --target -dir=p90 ordeMtems --m 1

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 p90_order_items/part-m-00000
Step 3 : In pyspark, get the total revenue across all days and orders. entire TableRDD = sc.textFile("p90_order_items")

#Cast string to float
extractedRevenueColumn = entireTableRDD.map(lambda line: float(line.split(",")[4]))

Step 4 : Verify extracted data
for revenue in extractedRevenueColumn.collect():
print revenue

#use reduce'function to sum a single column vale
totalRevenue = extractedRevenueColumn.reduce(lambda a, b: a + b)

Step 5 : Calculate the maximum revenue
maximumRevenue = extractedRevenueColumn.reduce(lambda a, b: (a if a>=b else b))

Step 6 : Calculate the minimum revenue
minimumRevenue = extractedRevenueColumn.reduce(lambda a, b: (a if a<=b else b))

Step 7 : Caclculate average revenue
count=extractedRevenueColumn.count()
averageRev=totalRevenue/count

Question No : 2 CORRECT TEXT

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}}}

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

Explanation:

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 No : 3 CORRECT TEXT

Problem Scenario 20 : You have been given MySQL DB with following details.

user=retail_dba

password=cloudera

database=retail_db

table=retail_db.categories

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

Please accomplish following activities.

1. Write a Sqoop Job which will import "retaildb.categories" table to hdfs, in a directory name "categories_targetJob".

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

Explanation:

Solution :

Step 1 : Connecting to existing MySQL Database mysql -user=retail_dba --password=cloudera retail_db

Step 2 : Show all the available tables show tables;

Step 3 : Below is the command to create Sqoop Job (Please note that - import space is mandatory)
sqoop job -create sqoopjob \ -- import \
-connect "jdbc:mysql://quickstart:3306/retail_db" \
-username=retail_dba \
-password=cloudera \
-table categories \
-target-dir categories_targetJob \
-fields-terminated-by '|' \
-lines-terminated-by '\n'

Step 4 : List all the Sqoop Jobs sqoop job --list

Step 5 : Show details of the Sqoop Job sqoop job --show sqoopjob

Step 6 : Execute the sqoopjob sqoopjob --exec sqoopjob

Step 7 : Check the output of import job
hdfs dfs -Is categories_target_job
hdfs dfs -cat categories_target_job/part*

Question No : 4 CORRECT TEXT

Problem Scenario 1:

You have been given MySQL DB with following details.

user=retail_dba

password=cloudera

database=retail_db

table=retail_db.categories

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

Please accomplish following activities.

1. Connect MySQL DB and check the content of the tables.

2. Copy "retaildb.categories" table to hdfs, without specifying directory name.

3. Copy "retaildb.categories" table to hdfs, in a directory name "categories_target".

4. Copy "retaildb.categories" table to hdfs, in a warehouse directory name "categories_warehouse".

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

Explanation:

Solution :

Step 1 : Connecting to existing MySQL Database mysql --user=retail_dba --password=cloudera retail_db

Step 2 : Show all the **available tables** show tables;

Step 3 : View/Count data from a table in MySQL select count(1} from categories;

Step 4 : Check the currently **available data in HDFS directory** hdfs dfs -ls

Step 5 : Import Single table (Without specifying directory).
sqoop import --connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba -password=cloudera -table=categories
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 6 : Read the data from one of the partition, created using above command, hdfs dfs -catxategories/part-m-00000

Step 7 : Specifying target directory in import command (We are using number of mappers =1, you can change accordingly) sqoop import -connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba -password=cloudera ~table=categories -target-dir=categortes_target --m 1

Step 8 : Check the content in one of the partition file.
hdfs dfs -cat categories_target/part-m-00000

Step 9 : Specifying parent directory so that you can copy more than one table in a specified target directory. Command to specify warehouse directory.
sqoop import -.-connect jdbc:mysql://quickstart:3306/retail_db --username=retail dba -password=cloudera -table=categories -warehouse-dir=categories_warehouse --m 1

Question No : 5 CORRECT TEXT

Problem Scenario 88 : You have been given below three files

product.csv (Create this file in hdfs)

productID,productCode,name,quantity,price,supplierid

1001,PEN,Pen Red,5000,1.23,501

1002,PEN,Pen Blue,8000,1.25,501

1003,PEN,Pen Black,2000,1.25,501

1004,PEC,Pencil 2B,10000,0.48,502

1005,PEC,Pencil 2H,8000,0.49,502

1006,PEC,Pencil HB,0,9999.99,502

2001,PEC,Pencil 3B,500,0.52,501

2002,PEC,Pencil 4B,200,0.62,501

2003,PEC,Pencil 5B,100,0.73,501

2004,PEC,Pencil 6B,500,0.47,502

supplier.csv

supplierid,name,phone

501,ABC Traders,88881111

502,XYZ Company,88882222

503,QQ Corp,88883333

products_suppliers.csv

productID,supplierID

2001,501

2002,501

2003,501

2004,502

2001,503

Now accomplish all the queries given in solution.

1. It is possible that, same product can be supplied by multiple supplier. Now find each product, its price according to each supplier.

2. Find all the supllier name, who are supplying 'Pencil 3B'

3. **Find all** the products , which are supplied by ABC Traders.

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

Explanation:

Solution :

Step 1 : It is possible that, same product can be supplied by multiple supplier. Now find each product, its price according to each supplier.

val results = sqlContext.sql(.....SELECT products.name AS Product Name', price, suppliers.name AS Supplier Name'

FROM products_suppliers

JOIN products ON products_suppliers.productID = products.productID JOIN suppliers ON products_suppliers.supplierID = suppliers.supplierID

null t

results.show()

Step 2 : Find all the supllier name, who are supplying 'Pencil 3B'

val results = sqlContext.sql(.....SELECT p.name AS 'Product Name", s.name AS "Supplier Name'

FROM products_suppliers AS ps
JOIN products AS p ON ps.productID = p.productID
JOIN suppliers AS s ON ps.supplierlD = s.supplierlD
WHERE p.name = 'Pencil 3B"",M)
results.show()

Step 3 : Find all the products , which are supplied by ABC Traders.
val results = sqlContext.sql(.....SELECT p.name AS 'Product Name", s.name AS "Supplier Name'
FROM products AS p, products_suppliers AS ps, suppliers AS s WHERE p.productID = ps.productID AND ps.supplierlD = s.supplierlD
AND s.name = 'ABC Traders".....)
results. show()

Question No : 6 CORRECT TEXT

Problem Scenario 23 : You have been given log generating service as below.

Start_logs (It will generate continuous logs)

Tail_logs (You can check , what logs are being generated)

Stop_logs (It will stop the log service)

Path where logs are generated using above service : /opt/gen_logs/logs/access.log

Now write a flume configuration file named flume3.conf , using that configuration file dumps logs in HDFS file system in a directory called flumeflume3/%Y/%m/%d/%H/%M

Means every minute new directory should be created). Please us the interceptors to provide timestamp information, if message header does not have header info.

And also note that you have to preserve existing timestamp, if message contains it. Flume channel should have following property as well. After every 100 message it should be committed, use non-durable/faster channel and it should be able to hold maximum 1000 events.

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

Explanation:
Solution :

Step 1 : Create flume configuration file, with below configuration for source, sink and channel.

#Define source , sink , channel and agent,
agent1 .sources = source1
agent1 .sinks = sink1
agent1.channels = channel1

Describe/configure source1
agent1 .sources.source1.type = exec
agentl.sources.source1.command = tail -F /opt/gen logs/logs/access.log

#Define interceptors
agent1 .sources.source1.interceptors=i1
agent1 .sources.source1.interceptors.i1.type=timestamp
agent1 .sources.source1.interceptors.i1.preserveExisting=true

Describe sink1
agent1 .sinks.sink1.channel = memory-channel
agent1 .sinks.sink1.type = hdfs
agent1 .sinks.sink1.hdfs.path = flume3/%Y/%m/%d/%H/%M
agent1 .sinks.sjnkl.hdfs.fileType = Data Stream

# Now we need to define channel1 property.
agent1.channels.channel1.type = memory
agent1.channels.channel1.capacity = 1000
agent1.channels.channel1.transactionCapacity = 100

# Bind the source and sink to the channel
Agent1.sources.source1.channels = channel1
agent1.sinks.sink1.channel = channel1

Step 2 : Run below command which **will use this** configuration file and append data in hdfs.
Start log service using : start_logs
Start flume service:
flume-ng agent -conf /home/cloudera/flumeconf -conf-file

/home/cloudera/flumeconf/flume3.conf -Dflume.root.logger=DEBUG,INFO,console –name agent1

Wait for few mins and than stop log service.

stop logs

Question No : 7 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 : 8 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

Question No : 9 CORRECT TEXT

Problem Scenario 44 : You have been given 4 files , with the content as given below:

spark11/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

spark11/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.

spark11/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

spark11/file4.txt

Apache Storm is focused on stream processing or what some call complex event processing. Storm implements a fault tolerant method for performing a computation or pipelining multiple computations on an event as it flows into a system. One might use Storm to transform unstructured data as it flows into a system into a desired format

(spark11Afile1.txt)

(spark11/file2.txt)

(spark11/file3.txt)

(sparkl 1/file4.txt)

Write a Spark program, which will give you the highest occurring words in each file. With their file name and highest occurring words.

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

Explanation:

Solution :

Step 1 : Create **all** 4 file first using Hue in hdfs.

Step 2 : **Load all file** as an RDD

val file1 = sc.textFile("sparkl1/filel.txt")

val file2 = sc.textFile("spark11/file2.txt")

val file3 = sc.textFile("spark11/file3.txt")

val file4 = sc.textFile("spark11/file4.txt")

Step 3 : Now do the word count for each file and sort in reverse order of count.

val contentl = filel.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_ + _).map(item => item.swap).sortByKey(false).map(e=>e.swap)

val content.2 = **file2.flatMap(line => line.splitf** ")).map(word => (word,1)).reduceByKey(_ + _).map(item => item.swap).sortByKey(false).map(e=>e.swap)

val content3 = **file3.flatMap(line > line.split)"** ")).map(word => (word,1)).reduceByKey(_ + _).map(item => item.swap).sortByKey(false).map(e=>e.swap)

val content4 = file4.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_ + _).map(item => item.swap).sortByKey(false).map(e=>e.swap)

Step 4 : Split the data and create RDD of all Employee objects.

val filelword = sc.makeRDD(Array(file1.name+"->"+content1(0)._1+"-"+content1(0)._2)) val file2word = sc.makeRDD(Array(file2.name+"->"+content2(0)._1+"-"+content2(0)._2)) val file3word = sc.makeRDD(Array(file3.name+"->"+content3(0)._1+"-"+content3(0)._2)) val file4word = sc.makeRDD(Array(file4.name+M->"+content4(0)._1+"-"+content4(0)._2))

Step 5: Union all the RDDS
val unionRDDs = filelword.union(file2word).union(file3word).union(file4word)

Step 6 : Save the results in a text file as below.
unionRDDs.repartition(1).saveAsTextFile("spark11/union.txt")

Question No : 10 CORRECT TEXT

Problem Scenario 29 : Please accomplish the following exercises using HDFS command line options.

1. Create a directory in hdfs named hdfs_commands.

2. Create a file in hdfs named data.txt in hdfs_commands.

3. Now copy this data.txt file on local filesystem, however while copying file please make sure file properties are not changed e.g. file permissions.

4. Now create a file in local directory named data_local.txt and move this file to hdfs in hdfs_commands directory.

5. Create a file data_hdfs.txt in hdfs_commands directory and copy it to local file system.

6. Create a file in **local** filesystem named file1.txt and put it to hdfs

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

Explanation:

Solution :

Step 1 : Create directory
hdfs dfs -mkdir hdfs_commands

Step 2 : Create a file in hdfs named data.txt in hdfs_commands. hdfs dfs -touchz hdfs_commands/data.txt

Step 3 : Now copy this data.txt file on local filesystem, however while copying file please make sure file properties are not changed e.g. file permissions.
hdfs dfs -copyToLocal -p hdfs_commands/data.txt/home/cloudera/Desktop/HadoopExam

Step 4 : Now create a file in local directory named data_local.txt and move this file to hdfs in hdfs_commands directory.
touch data_local.txt
hdfs dfs -moveFromLocal /home/cloudera/Desktop/HadoopExam/dataJocal.txt hdfs_commands/

Step 5 : Create a file data_hdfs.txt in hdfs_commands directory and copy it to local file system.
hdfs dfs -touchz hdfscommands/data hdfs.txt
hdfs dfs -getfrdfs_commands/data_hdfs.txt /home/cloudera/Desktop/HadoopExam/

Step 6 : Create a file in local filesystem named filel .txt and put it to hdfs
touch filel.txt
hdfs dfs -put/home/cloudera/Desktop/HadoopExam/file1.txt hdfs_commands/