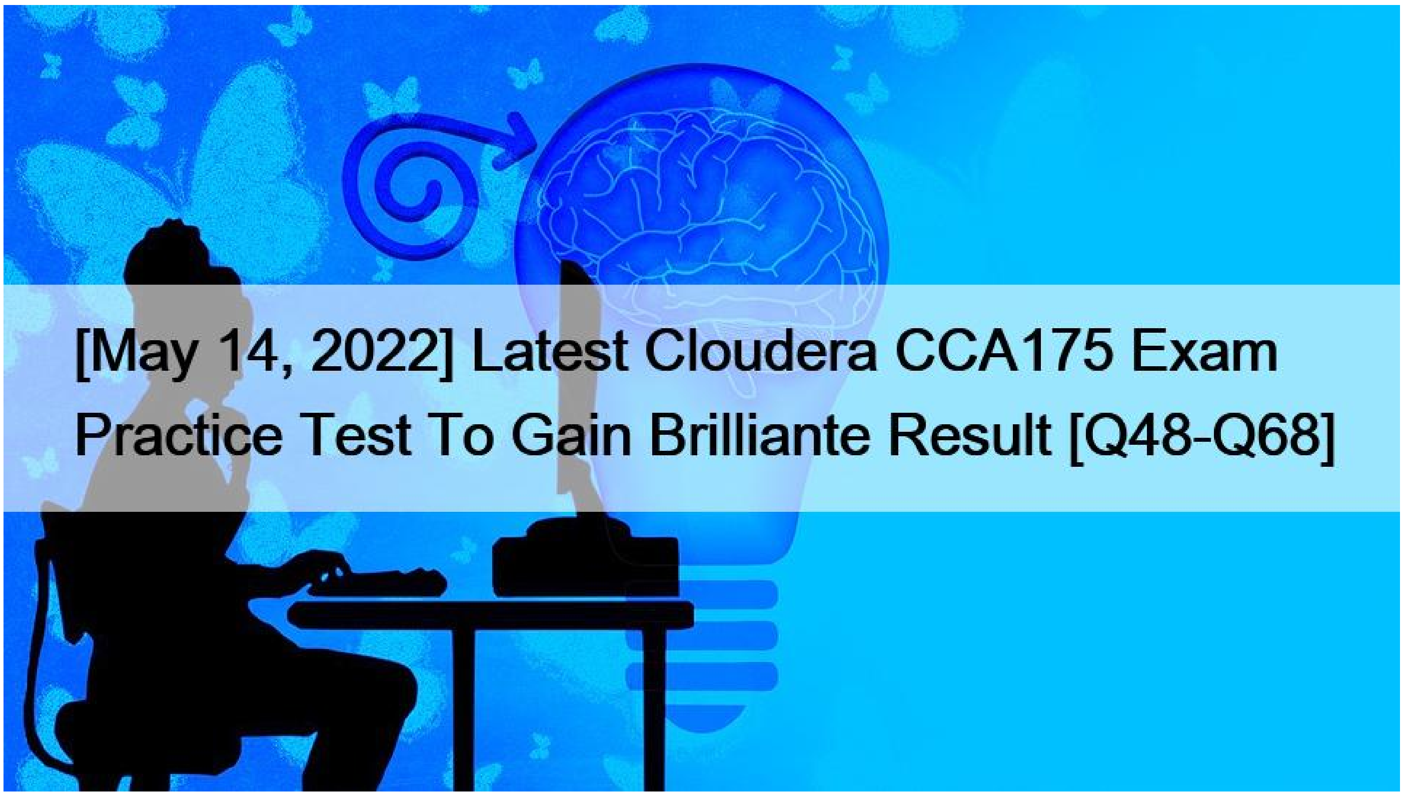
[May 14, 2022 Latest Cloudera CCA175 Exam Practice Test To Gain Brilliante Result [Q48-Q68



Latest [May 14, 2022] Cloudera CCA175 Exam Practice Test To Gain Brilliante Result
Take a Leap Forward in Your Career by Earning Cloudera CCA175

Factors for Success in CCA175 Exam

Incorrect answers are being ignored to get success in the exam. Correct answers are being used to get success in the exam. CCA175 Exam is based on Cloudera technologies. Questions of CCA175 exam are analyzed in a perfect manner. Scenarios of CCA175 exam are analyzed in the most appropriate manner. **Cloudera CCA175 exam dumps** are of great use to get success in the exam. Solved CCA175 exam questions are sufficient to get success in the exam. Streaming data is being used to get success in the exam. Free CCA175 exam questions are being provided for the candidates. Syllabus of CCA175 exam is sufficient to get success in the exam. Code of CCA175 exam questions is enough to get success in the exam. Querying abilities are being used to get success in the exam. Associate the exam questions with the real exam. Sqoop is being used to solve the CCA175 exam questions. Configuration of CCA175 exam questions is sufficient to get success in the exam. Interview of Cloudera Certified Advanced Architect- Data Engineer exam is sufficient to get success in the exam.

How much the Exam Cost of CCA Spark and Hadoop Developer (CCA175) Exam

CCA Spark and Hadoop Developer (CCA175) certification exam cost is US $295.

Grabbing Cloudera Certified Advanced Architect- Data Engineer Exam

Industry experts are analyzing the exam questions to solve them in the most appropriate manner. Loads of CCA175 exam questions are adequate to get success in the exam. Registered companies are using the best IT experts to get success in the exam. It is read that the Cloudera Certified Advanced Architect- Data Engineer exam is being updated constantly. Metastore is being used to solve the

exam questions. Hive metastore is being used to get success in the exam. Jar jobs are being used to get success in the exam. Failure to get success in the exam will not be entertained by any means. Flume is being used to get success in the exam. Days of the exam are being a source to get success in the exam. Solving all the vendors codes of CCA175 exam questions is enough to get success in the exam. Core engine of Cloudera Certified Advanced Architect- Data Engineer exam is sufficient to get success in the exam. Exampreparation of CCA175 exam questions is enough to get success in the exam.
Subscribe the Cloudera Certified Advanced Architect- Data Engineer exam to get success in the exam. Videos of CCA175 exam questions are sufficient to get success in the exam. Operationswriting is one of the ways to solve the exam questions. Ingest is being used to get success in the exam.

NEW QUESTION 48

CORRECT TEXT

Problem Scenario 57 : You have been given below code snippet.

val a = sc.parallelize(1 to 9, 3) operationl

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

Array[(String, Seq[lnt])] = Array((even,ArrayBuffer(2, 4, G, 8)), (odd,ArrayBuffer(1, 3, 5, 7,

9)))
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

a.groupBy(x => {if (x % 2 == 0) &#8220;even&#8221; else &#8220;odd&#8221; }).collect

NEW QUESTION 49

CORRECT TEXT

Problem Scenario 39 : You have been given two files

spark16/file1.txt

1,9,5

2,7,4

3,8,3

spark16/file2.txt

1 ,g,h

2 ,i,j

3 ,k,l

Load these two tiles as Spark RDD and join them to produce the below results

(l,((9,5),(g,h)))

(2, ((7,4), (i,j))) (3, ((8,3), (k,l)))

And write code snippet which will sum the second columns of above joined results (5+4+3).
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Create tiles in hdfs using Hue.

Step 2 : Create pairRDD for both the files.

val one = sc.textFile(&#8220;spark16/file1.txt&#8221;).map{

_.split(&#8220;,&#8221;,-1) match {

case Array(a, b, c) => (a, (b, c))

} }

val two = sc.textFHe(Mspark16/file2.txt&#8221;).map{

_ .split(&#8216;7-1) match {

case Array(a, b, c) => (a, (b, c))

} }

Step 3 : Join both the RDD. val joined = one.join(two)

Step 4 : Sum second column values.

val sum = joined.map {

case (_, ((_, num2), (_, _))) => num2.tolnt

}.reduce(_ + _)

NEW QUESTION 50

CORRECT TEXT

Problem Scenario 85 : In Continuation of previous question, please accomplish following activities.

1. Select all the columns from product table with output header as below. productID AS ID code AS Code name AS Description price AS &#8216;Unit Price&#8217;

2. Select code and name both separated by &#8216; -&#8216; and header name should be Product

Description&#8217;.

3. Select all distinct prices.

4 . Select distinct price and name combination.

5 . Select all price data sorted by both code and productID combination.

6 . count number of products.

7 . Count number of products for each code.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Select all the columns from product table with output header as below. productID

AS ID code AS Code name AS Description price AS &#8220;Unit Price&#8217;

val results = sqlContext.sql(&#8230;&#8230;SELECT productID AS ID, code AS Code, name AS

Description, price AS Unit Price&#8217; FROM products ORDER BY ID&#8221;&#8221;&#8221;

results.show()

Step 2 : Select code and name both separated by &#8216; -&#8216; and header name should be &#8220;Product

Description.

val results = sqlContext.sql(&#8230;&#8230;SELECT CONCAT(code,&#8217; -&#8216;, name) AS Product Description, price FROM products&#8221;&#8221;&#8221;) results.showQ

Step 3 : Select all distinct prices.

val results = sqlContext.sql(&#8230;&#8230;SELECT DISTINCT price AS Distinct Price&#8221; FROM products&#8230;&#8230;) results.show()

Step 4 : Select distinct price and name combination.

val results = sqlContext.sql(&#8230;&#8230;SELECT DISTINCT price, name FROM products&#8221;&#8221;&#8221;) results. showQ

Step 5 : Select all price data sorted by both code and productID combination.

val results = sqlContext.sql(&#8216;&#8230;..SELECT&#8217; FROM products ORDER BY code, productID&#8217;&#8230;..) results.show()

Step 6 : count number of products.

val results = sqlContext.sql(&#8230;&#8230;SELECT COUNT(&#8216;) AS &#8216;Count&#8217; FROM products&#8230;&#8230;) results.show()

Step 7 : Count number of products for each code.

val results = sqlContext.sql(&#8230;&#8230;SELECT code, COUNT(&#8216;} FROM products GROUP BY code&#8230;&#8230;) results. showQ val results = sqlContext.sql(&#8230;&#8230;SELECT code, COUNT(&#8216;} AS count FROM products

GROUP BY code ORDER BY count DESC&#8230;&#8230;)

results. showQ

NEW QUESTION 51

CORRECT TEXT

Problem Scenario 61 : You have been given below code snippet.

val a = sc.parallelize(List(&#8220;dog&#8221;, &#8220;salmon&#8221;, &#8220;salmon&#8221;, &#8220;rat&#8221;, &#8220;elephant&#8221;), 3) val b = a.keyBy(_.length) val c = sc.parallelize(List(&#8220;dog&#8221;,&#8221;cat&#8221;,&#8221;gnu&#8221;,&#8221;salmon&#8221;,&#8221;rabbit&#8221;,&#8221;turkey&#8221;,&#8221;wolf&#8221;,&#8221;bear&#8221;,&#8221;bee&#8221;), 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)))
See the explanation for Step by Step Solution and configuration.

Explanation:

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 two- component 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.

NEW QUESTION 52

CORRECT TEXT

Problem Scenario 86 : In Continuation of previous question, please accomplish following activities.

1 . Select Maximum, minimum, average , Standard Deviation, and total quantity.

2 . Select minimum and maximum price for each product code.

3. Select Maximum, minimum, average , Standard Deviation, and total quantity for each product code, hwoever make sure Average and Standard deviation will have maximum two decimal values.

4. Select all the product code and average price only where product count is more than or equal to 3.

5. Select maximum, minimum , average and total of all the products for each code. Also produce the same across all the products.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Select Maximum, minimum, average , Standard Deviation, and total quantity.

val results = sqlContext.sql(&#8216;&#8230;..SELECT MAX(price) AS MAX , MIN(price) AS MIN ,

AVG(price) AS Average, STD(price) AS STD, SUM(quantity) AS total_products FROM products&#8230;&#8230;) results. showQ

Step 2 : Select minimum and maximum price for each product code.

val results = sqlContext.sql(&#8230;&#8230;SELECT code, MAX(price) AS Highest Price&#8217;, MIN(price)

AS Lowest Price&#8217;

FROM products GROUP BY code&#8230;&#8230;)

results. showQ

Step 3 : Select Maximum, minimum, average , Standard Deviation, and total quantity for each product code, hwoever make sure Average and Standard deviation will have maximum two decimal values.

val results = sqlContext.sql(&#8230;&#8230;SELECT code, MAX(price), MIN(price), CAST(AVG(price} AS DECIMAL(7,2)) AS Average&#8217;, CAST(STD(price) AS DECIMAL(7,2))

AS &#8216;Std Dev SUM(quantity) FROM products

GROUP BY code&#8230;&#8230;)

results. showQ

Step 4 : Select all the product code and average price only where product count is more than or equal to 3.

val results = sqlContext.sql(&#8230;&#8230;SELECT code AS Product Code&#8217;,

COUNTf) AS Count&#8217;,

CAST(AVG(price) AS DECIMAL(7,2)) AS Average&#8217; FROM products GROUP BY code

HAVING Count >=3&#8243;M&#8221;) results. showQ

Step 5 : Select maximum, minimum , average and total of all the products for each code.

Also produce the same across all the products.

val results = sqlContext.sql(&#8220;&#8221;&#8221;SELECT

code,

MAX(price),

MIN(pnce),

CAST(AVG(price) AS DECIMAL(7,2)) AS Average&#8217;,

SUM(quantity)-

FROM products

GROUP BY code

WITH ROLLUP&#8221;&#8221;&#8221;)

results. show()

NEW QUESTION 53

CORRECT TEXT

Problem Scenario 12 : 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. Create a table in retailedb with following definition.

CREATE table departments_new (department_id int(11), department_name varchar(45), created_date T1MESTAMP DEFAULT NOW());

2 . Now isert records from departments table to departments_new

3 . Now import data from departments_new table to hdfs.

4 . Insert following 5 records in departmentsnew table. Insert into departments_new values(110, &#8220;Civil&#8221; , null); Insert into departments_new values(111, &#8220;Mechanical&#8221; , null);

Insert into departments_new values(112, &#8220;Automobile&#8221; , null); Insert into departments_new values(113, &#8220;Pharma&#8221; , null);

Insert into departments_new values(114, &#8220;Social Engineering&#8221; , null);

5. Now do the incremental import based on created_date column.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Login to musql db

mysql &#8211;user=retail_dba -password=cloudera

show databases;

use retail db; show tables;

Step 2 : Create a table as given in problem statement.

CREATE table departments_new (department_id int(11), department_name varchar(45), createddate T1MESTAMP DEFAULT NOW()); show tables;

Step 3 : isert records from departments table to departments_new insert into departments_new select a.&#8221;, null from departments a;

Step 4 : Import data from departments new table to hdfs.

sqoop import

-connect jdbc:mysql://quickstart:330G/retail_db

~ username=retail_dba

-password=cloudera

-table departments_new

&#8211;target-dir /user/cloudera/departments_new

&#8211;split-by departments

Stpe 5 : Check the imported data.

hdfs dfs -cat /user/cloudera/departmentsnew/part&#8221;

Step 6 : Insert following 5 records in departmentsnew table.

Insert into departments_new values(110, &#8220;Civil&#8221; , null);

Insert into departments_new values(111, &#8220;Mechanical&#8221; , null);

Insert into departments_new values(112, &#8220;Automobile&#8221; , null);

Insert into departments_new values(113, &#8220;Pharma&#8221; , null);

Insert into departments_new values(114, &#8220;Social Engineering&#8221; , null);

commit;

Stpe 7 : Import incremetal data based on created_date column.

sqoop import

-connect jdbc:mysql://quickstart:330G/retaiI_db

-username=retail_dba

-password=cloudera

&#8211;table departments_new

-target-dir /user/cloudera/departments_new

-append

-check-column created_date

-incremental lastmodified

-split-by departments

-last-value &#8220;2016-01-30 12:07:37.0&#8221;

Step 8 : Check the imported value.

hdfs dfs -cat /user/cloudera/departmentsnew/part&#8221;

NEW QUESTION 54

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.
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 -DfIume.root.logger=DEBUG,INFO,console -name agent1
```

Wait for few mins and than stop log service.

stop logs

NEW QUESTION 55

CORRECT TEXT

Problem Scenario 31 : You have given following two files

1 . Content.txt: Contain a huge text file containing space separated words.

2 . Remove.txt: Ignore/filter all the words given in this file (Comma Separated).

Write a Spark program which reads the Content.txt file and load as an RDD, remove all the words from a broadcast variables (which is loaded as an RDD of words from Remove.txt).

And count the occurrence of the each word and save it as a text file in HDFS.

Content.txt

Hello this is ABCTech.com

This is TechABY.com

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Remove.txt

Hello, is, this, the
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Create all three files in hdfs in directory called spark2 (We will do using Hue).

However, you can first create in local filesystem and then upload it to hdfs

Step 2 : Load the Content.txt file

val content = sc.textFile(&#8220;spark2/Content.txt&#8221;) //Load the text file

Step 3 : Load the Remove.txt file

val remove = sc.textFile(&#8220;spark2/Remove.txt&#8221;) //Load the text file

Step 4 : Create an RDD from remove, However, there is a possibility each word could have trailing spaces, remove those whitespaces as well. We have used two functions here flatMap, map and trim.

val removeRDD= remove.flatMap(x=> x.splitf&#8217;,&#8221;)).map(word=>word.trim)//Create an array of words

Step 5 : Broadcast the variable, which you want to ignore

val bRemove = sc.broadcast(removeRDD.collect().toList) // It should be array of Strings

Step 6 : Split the content RDD, so we can have Array of String. val words = content.flatMap(line => line.split(&#8221; &#8220;))

Step 7 : Filter the RDD, so it can have only content which are not present in &#8220;Broadcast

Variable&#8221;. val filtered = words.filter{case (word) => !bRemove.value.contains(word)}

Step 8 : Create a PairRDD, so we can have (word,1) tuple or PairRDD. val pairRDD = filtered.map(word => (word,1))

Step 9 : Nowdo the word count on PairRDD. val wordCount = pairRDD.reduceByKey(_ + _)

Step 10 : Save the output as a Text file.

wordCount.saveAsTextFile(&#8220;spark2/result.txt&#8221;)

NEW QUESTION 56

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, &#8220;Not known&#8221;,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 &#8220;&#8221; (empty string) and for id column with -999 select * from departments_hive;
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&#8217; from departments_hive01;

Now iserts null records as given in problem. insert into departments_hive01 values(777,

&#8220;Not known&#8221;,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

&#8211;password=cloudera

-table departments_hive01

&#8211;hive-home /user/hive/warehouse

&#8211;hive-import

-hive-overwrite

-hive-table departments_hive0l

&#8211;fields-terminated-by &#8216;01&#8217;

&#8211;null-string M&#8221;

&#8211;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&#8221;

Check data in hive table.

Select * from departments_hive01;

NEW QUESTION 57

CORRECT TEXT

Problem Scenario 4: 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.

Import Single table categories (Subset data} to hive managed table , where category_id between 1 and 22
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Import Single table (Subset data)

sqoop import &#8211;connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba &#8211; password=cloudera -table=categories -where &#8220;&#8217;category_id&#8217; between 1 and 22&#8221; &#8211;hive- import &#8211;m 1

Note: Here the &#8216; is the same you find on ~ key

This command will create a managed table and content will be created in the following directory.

/user/hive/warehouse/categories

Step 2 : Check whether table is created or not (In Hive)

show tables;

select * from categories;

NEW QUESTION 58

CORRECT TEXT

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

val a = sc.parallelize(List(&#8220;dogM, &#8220;tiger&#8221;;, &#8220;lion&#8221;;, &#8220;cat&#8221;;, &#8220;panther&#8221;;, &#8220;eagle&#8221;;), 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))
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

b.mapValuesf&#8217;x&#8221; + _ + &#8220;x&#8221;).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.

NEW QUESTION 59

CORRECT TEXT

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

val a = sc.parallelize(List(&#8220;dog&#8221;, &#8220;tiger&#8221;, &#8220;lion&#8221;, &#8220;cat&#8221;, &#8220;spider&#8221;, &#8220;eagle&#8221;), 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}}}
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])]

NEW QUESTION 60

CORRECT TEXT

Problem Scenario 68 : You have given a file as below.

spark75/f ile1.txt

File contain some text. As given Below

spark75/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

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.

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

For a slightly more complicated task, lets look into splitting up sentences from our documents into word bigrams. A bigram is pair of successive tokens in some sequence.

We will look at building bigrams from the sequences of words in each sentence, and then try to find the most frequently occuring ones.

The first problem is that values in each partition of our initial RDD describe lines from the file rather than sentences. Sentences may be split over multiple lines. The glom() RDD method is used to create a single entry for each document containing the list of all lines, we can then join the lines up, then resplit them into sentences using &#8220;.&#8221; as the separator, using flatMap so that every object in our RDD is now a sentence.

A bigram is pair of successive tokens in some sequence. Please build bigrams from the sequences of words in each sentence, and then try to find the most frequently occuring ones.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

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

Step 2 : The first problem is that values in each partition of our initial RDD describe lines from the file rather than sentences. Sentences may be split over multiple lines.

The glom() RDD method is used to create a single entry for each document containing the list of all lines, we can then join the lines up, then resplit them into sentences using &#8220;.&#8221; as the separator, using flatMap so that every object in our RDD is now a sentence.

sentences = sc.textFile(&#8220;spark75/file1.txt&#8221;) .glom()

map(lambda x: &#8221; &#8220;.join(x)) .flatMap(lambda x: x.spllt(&#8220;.&#8221;))

Step 3 : Now we have isolated each sentence we can split it into a list of words and extract the word bigrams from it. Our new RDD contains tuples containing the word bigram (itself a tuple containing the first and second word) as the first value and the number 1 as the second value. bigrams = sentences.map(lambda x:x.split())

.flatMap(lambda x: [((x[i],x[i+1]),1)for i in range(0,len(x)-1)])

Step 4 : Finally we can apply the same reduceByKey and sort steps that we used in the wordcount example, to count up the bigrams and sort them in order of descending frequency. In reduceByKey the key is not an individual word but a bigram.

freq_bigrams = bigrams.reduceByKey(lambda x,y:x+y)

map(lambda x:(x[1],x[0]))

sortByKey(False)

freq_bigrams.take(10)

NEW QUESTION 61

CORRECT TEXT

Problem Scenario 47 : You have been given below code snippet, with intermediate output.

val z = sc.parallelize(List(1,2,3,4,5,6), 2)

// lets first print out the contents of the RDD with partition labels

def myfunc(index: Int, iter: lterator[(lnt)]): lterator[String] = {

iter.toList.map(x => &#8220;[partID:&#8221; + index + &#8220;, val: &#8221; + x + &#8220;]&#8221;).iterator

}

//In each run , output could be different, while solving problem assume belowm output only.

z.mapPartitionsWithlndex(myfunc).collect

res28: Array[String] = Array([partlD:0, val: 1], [partlD:0, val: 2], [partlD:0, val: 3], [partlD:1, val: 4], [partlD:1, val: S], [partlD:1, val: 6])

Now apply aggregate method on RDD z , with two reduce function , first will select max value in each partition and second will add all the maximum values from all partitions.

Initialize the aggregate with value 5. hence expected output will be 16.
z.aggregate(5)(math.max(_, J, _ + _)

NEW QUESTION 62

CORRECT TEXT

Problem Scenario 15 : 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 activities.

1. In mysql departments table please insert following record. Insert into departments values(9999, &#8216;&#8221;Data Science&#8221;1);

2. Now there is a downstream system which will process dumps of this file. However, system is designed the way that it can process only files if fields are enlcosed in(&#8216;) single quote and separate of the field should be (-} and line needs to be terminated by : (colon).

3. If data itself contains the &#8221; (double quote } than it should be escaped by .

4. Please import the departments table in a directory called departments_enclosedby and file should be able to process by downstream system.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Connect to mysql database.

mysql &#8211;user=retail_dba -password=cloudera

show databases; use retail_db; show tables;

Insert record

Insert into departments values(9999, &#8216;&#8221;Data Science&#8221;&#8216;);

select&#8221; from departments;

Step 2 : Import data as per requirement.

sqoop import

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

~ username=retail_dba

&#8211;password=cloudera

-table departments

-target-dir /user/cloudera/departments_enclosedby

-enclosed-by V -escaped-by \ -fields-terminated-by&#8211;&#8216; -lines-terminated-by :

Step 3 : Check the result.

hdfs dfs -cat/user/cloudera/departments_enclosedby/part&#8221;

NEW QUESTION 63

CORRECT TEXT

Problem Scenario 82 : You have been given table in Hive with following structure (Which you have created in previous exercise).

productid int code string name string quantity int price float

Using SparkSQL accomplish following activities.

1 . Select all the products name and quantity having quantity <= 2000

2 . Select name and price of the product having code as &#8216;PEN&#8217;

3 . Select all the products, which name starts with PENCIL

4 . Select all products which &#8220;name&#8221; begins with &#8216;P followed by any two characters, followed by space, followed by zero or more characters
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Copy following tile (Mandatory Step in Cloudera QuickVM) if you have not done it.

sudo su root

cp /usr/lib/hive/conf/hive-site.xml /usr/lib/sparkVconf/

Step 2 : Now start spark-shell

Step 3 ; Select all the products name and quantity having quantity <= 2000 val results = sqlContext.sql(&#8230;&#8230;SELECT name, quantity FROM products WHERE quantity

< = 2000&#8230;&#8230;)

results.showQ

Step 4 : Select name and price of the product having code as &#8216;PEN&#8217;

val results = sqlContext.sql(&#8230;&#8230;SELECT name, price FROM products WHERE code =

&#8216;PEN&#8230;&#8230;.)

results. showQ

Step 5 : Select all the products , which name starts with PENCIL

val results = sqlContext.sql(&#8230;&#8230;SELECT name, price FROM products WHERE upper(name) LIKE &#8216;PENCIL%&#8230;&#8230;.} results. showQ

Step 6 : select all products which &#8220;name&#8221; begins with &#8216;P&#8217;, followed by any two characters, followed by space, followed byzero or more characters

&#8212; &#8220;name&#8221; begins with &#8216;P&#8217;, followed by any two characters,

&#8211; followed by space, followed by zero or more characters

val results = sqlContext.sql(&#8230;&#8230;SELECT name, price FROM products WHERE name LIKE

&#8216;P_ %&#8230;&#8230;.)

results. show()

NEW QUESTION 64

CORRECT TEXT

Problem Scenario 74 : 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 : (orderjd , order_date , ordercustomerid, order status}

Columns of orderjtems table : (order_item_td , order_item_order_id ,

order_item_product_id,

order_item_quantity,order_item_subtotal,order_item_product_price)

Please accomplish following activities.

1. Copy &#8220;retaildb.orders&#8221; and &#8220;retaildb.orderjtems&#8221; table to hdfs in respective directory p89_orders and p89_order_items .

2. Join these data using orderjd in Spark and Python

3. Now fetch selected columns from joined data Orderld, Order date and amount collected on this order.

4. Calculate total order placed for each date, and produced the output sorted by date.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution:

Step 1 : Import Single table .

sqoop import &#8211;connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba &#8211; password=cloudera -table=orders &#8211;target-dir=p89_orders &#8211; -m1 sqoop import &#8211;connect jdbc:mysql://quickstart:3306/retail_db -username=retail_dba &#8211; password=cloudera -table=order_items ~target-dir=p89_ order items -m 1

Note : Please check you dont have space between before or after &#8216;=&#8217; 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, hadoopfs

-cat p89_orders/part-m-00000 hadoop fs -cat p89_order_items/part-m-00000

Step 3 : Load these above two directory as RDD using Spark and Python (Open pyspark terminal and do following). orders = sc.textFile(&#8220;p89_orders&#8221;;) orderitems = sc.textFile(&#8220;p89_order_items&#8221;;)

Step 4 : Convert RDD into key value as (orderjd as a key and rest of the values as a value)

#First value is orderjd

ordersKeyValue = orders.map(lambda line: (int(line.split(&#8220;,&#8221;)[0]), line))

#Second value as an Orderjd

orderltemsKeyValue = orderltems.map(lambda line: (int(line.split(&#8220;,&#8221;)[1]), line))

Step 5 : Join both the RDD using orderjd

joinedData = orderltemsKeyValue.join(ordersKeyValue)

#print the joined data

tor line in joinedData.collect():

print(line)

Format of joinedData as below.

[Orderld, &#8216;All columns from orderltemsKeyValue&#8217;, &#8216;All columns from orders Key Value&#8217;]

Step 6 : Now fetch selected values Orderld, Order date and amount collected on this order.

revenuePerOrderPerDay = joinedData.map(lambda row: (row[0](row[1][1].split(&#8220;,&#8221;)[1](f!oat(row[1][0].split(&#8216;M}[4]}}}

#printthe result

for line in revenuePerOrderPerDay.collect():

print(line)

Step 7 : Select distinct order ids for each date.

#distinct(date,order_id)

distinctOrdersDate = joinedData.map(lambda row: row[1][1].split(&#8216;&#8221;)[1] + &#8220;,&#8221; + str(row[0])).distinct() for line in distinctOrdersDate.collect(): print(line)

Step 8 : Similar to word count, generate (date, 1) record for each row. newLineTuple = distinctOrdersDate.map(lambda line: (line.split(&#8220;,&#8221;)[0], 1))

Step 9 : Do the count for each key(date), to get total order per date. totalOrdersPerDate = newLineTuple.reduceByKey(lambda a, b: a + b}

#print results

for line in totalOrdersPerDate.collect():

print(line)

step 10 : Sort the results by date sortedData=totalOrdersPerDate.sortByKey().collect()

#print results

for line in sortedData:

print(line)

NEW QUESTION 65

CORRECT TEXT

Problem Scenario 2 :

There is a parent organization called &#8220;ABC Group Inc&#8221;, which has two child companies named Tech Inc and MPTech.

Both companies employee information is given in two separate text file as below. Please do the following activity for employee details.

Tech Inc.txt

1,Alok,Hyderabad

2,Krish,Hongkong

3,Jyoti,Mumbai

4 ,Atul,Banglore

5 ,Ishan,Gurgaon

MPTech.txt

6 ,John,Newyork

7 ,alp2004,California

8 ,tellme,Mumbai

9 ,Gagan21,Pune

1 0,Mukesh,Chennai

1 . Which command will you use to check all the available command line options on HDFS and How will you get the Help for individual command.

2. Create a new Empty Directory named Employee using Command line. And also create an empty file named in it Techinc.txt

3. Load both companies Employee data in Employee directory (How to override existing file in HDFS).

4. Merge both the Employees data in a Single tile called MergedEmployee.txt, merged tiles should have new line character at the end of each file content.

5. Upload merged file on HDFS and change the file permission on HDFS merged file, so that owner and group member can read and write, other user can read the file.

6. Write a command to export the individual file as well as entire directory from HDFS to local file System.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Check All Available command hdfs dfs

Step 2 : Get help on Individual command hdfs dfs -help get

Step 3 : Create a directory in HDFS using named Employee and create a Dummy file in it called e.g. Techinc.txt hdfs dfs -mkdir Employee

Now create an emplty file in Employee directory using Hue.

Step 4 : Create a directory on Local file System and then Create two files, with the given data in problems.

Step 5 : Now we have an existing directory with content in it, now using HDFS command line , overrid this existing Employee directory. While copying these files from local file

System to HDFS. cd /home/cloudera/Desktop/ hdfs dfs -put -f Employee

Step 6 : Check All files in directory copied successfully hdfs dfs -Is Employee

Step 7 : Now merge all the files in Employee directory, hdfs dfs -getmerge -nl Employee

MergedEmployee.txt

Step 8 : Check the content of the file. cat MergedEmployee.txt

Step 9 : Copy merged file in Employeed directory from local file ssytem to HDFS. hdfs dfs &#8211; put MergedEmployee.txt Employee/

Step 10 : Check file copied or not. hdfs dfs -Is Employee

Step 11 : Change the permission of the merged file on HDFS hdfs dfs -chmpd 664

Employee/MergedEmployee.txt

Step 12 : Get the file from HDFS to local file system, hdfs dfs -get Employee

Employee_hdfs

NEW QUESTION 66

CORRECT TEXT

Problem Scenario 35 : You have been given a file named spark7/EmployeeName.csv

(id,name).

EmployeeName.csv

E01,Lokesh

E02,Bhupesh

E03,Amit

E04,Ratan

E05,Dinesh

E06,Pavan

E07,Tejas

E08,Sheela

E09,Kumar

E10,Venkat

1. Load this file from hdfs and sort it by name and save it back as (id,name) in results directory. However, make sure while saving it should be able to write In a single file.
See the explanation for Step by Step Solution and configuration.

Explanation:

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 EmployeeName.csv file from hdfs and create PairRDDs

```
val name = sc.textFile(&#8220;spark7/EmployeeName.csv&#8221;)
```

```
val namePairRDD = name.map(x=> (x.split(&#8220;,&#8221;)(0),x.split(&#8220;,&#8221;)(1)))
```

Step 3 : Now swap namePairRDD RDD.

```
val swapped = namePairRDD.map(item => item.swap)
```

step 4: Now sort the rdd by key.

```
val sortedOutput = swapped.sortByKey()
```

Step 5 : Now swap the result back

```
val swappedBack = sortedOutput.map(item => item.swap}
```

Step 6 : Save the output as a Text file and output must be written in a single file.

```
swappedBack. repartition(1).saveAsTextFile(&#8220;spark7/result.txt&#8221;)
```

NEW QUESTION 67

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(&#8216;7output/path&#8221;,classOt[GzipCodec])

What would be the correct replacement for A and B in above snippet.
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

A. NullWritable

B. BytesWritable

NEW QUESTION 68

CORRECT TEXT

Problem Scenario 16 : 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_hive(department_id int, department_name string);

2. Now import data from mysql table departments to this hive table. Please make sure that data should be visible using below hive command, select&#8221; from departments_hive
See the explanation for Step by Step Solution and configuration.

Explanation:

Solution :

Step 1 : Create hive table as said.

hive

show tables;

create table departments_hive(department_id int, department_name string);

Step 2 : The important here is, when we create a table without delimiter fields. Then default delimiter for hive is A (01). Hence, while importing data we have to provide proper delimiter.

sqoop import

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

~ username=retail_dba

-password=cloudera

&#8211;table departments

&#8211;hive-home /user/hive/warehouse

-hive-import

-hive-overwrite

&#8211;hive-table departments_hive

&#8211;fields-terminated-by &#8216;01&#8217;

Step 3 : Check-the data in directory.

hdfs dfs -Is /user/hive/warehouse/departments_hive

hdfs dfs -cat/user/hive/warehouse/departmentshive/part&#8217;

Check data in hive table.

Select * from departments_hive;

Authentic Best resources for CCA175 Online Practice Exam: https://www.examslabs.com/Cloudera/Cloudera-Certified/best-CCA175-exam-dumps.html]