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.pass4itsure.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 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 2

Problem Scenario 40 : You have been given sample data as below in a file called spark15/file1.txt 3070811,1963,1096,,"US","CA",,1, 3022811,1963,1096,,"US","CA",,1,56 3033811,1963,1096,,"US","CA",,1,23 Below is the code snippet to process this tile. val field= sc.textFile("spark15/f ilel.txt") val mapper = field.map(x=> A) mapper.map(x => x.map(x=> {B})).collect

Please fill in A and B so it can generate below final output

Array(Array(3070811,1963,109G, 0, "US", "CA", 0,1, 0)

,Array(3022811,1963,1096, 0, "US", "CA", 0,1, 56)

,Array(3033811,1963,1096, 0, "US", "CA", 0,1, 23)

)

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

Solution :

A. x.split(",”-1)

B. if (x. isEmpty) 0 else x

QUESTION 3

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 \\'Unit Price\\'

2.

Select code and name both separated by \\' -\\' and header name should be Product Description\\'.

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.

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

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 "Unit Price\\'

val results = sqlContext.sql(.....SELECT productID AS ID, code AS Code, name AS

Description, price AS Unit Price\\' FROM products ORDER BY ID"""

results.show()

Step 2 : Select code and name both separated by \\' -\\' and header name should be "Product

Description.

val results = sqlContext.sql(.....SELECT CONCAT(code,\\' -\\', name) AS Product Description,

price FROM products""")

results.showQ

Step 3 : Select all distinct prices.

val results = sqlContext.sql(.....SELECT DISTINCT price AS Distinct Price" FROM

products.....)

results.show()

Step 4 : Select distinct price and name combination.

val results = sqlContext.sql(.....SELECT DISTINCT price, name FROM products""")

results. showQ

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

val results = sqlContext.sql(\\'.....SELECT\\' FROM products ORDER BY code, productID\\'.....)

results.show()

Step 6 : count number of products.

val results = sqlContext.sql(.....SELECT COUNT(\\') AS \\'Count\\' FROM products.....)

results.show()

Step 7 : Count number of products for each code.

val results = sqlContext.sql(.....SELECT code, COUNT(\\'} FROM products GROUP BY

code.....)

results. showQ

val results = sqlContext.sql(.....SELECT code, COUNT(\\'} AS count FROM products

GROUP BY code ORDER BY count DESC.....)

results. showQ

QUESTION 4

Problem Scenario 84 : In Continuation of previous question, please accomplish following activities.

1.

Select all the products which has product code as null

2.

Select all the products, whose name starts with Pen and results should be order by Price descending order.

3.

Select all the products, whose name starts with Pen and results should be order by Price descending order and quantity ascending order.

4.

Select top 2 products by price

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

Solution : Step 1 : Select all the products which has product code as null val results = sqlContext.sql(.....SELECT\\' FROM products WHERE code IS NULL.....) results. showQ val results = sqlContext.sql(.....SELECT * FROM products WHERE code = NULL ",,M) results.showQ Step 2 : Select all the products , whose name starts with Pen and results should be order by Price descending order. val results = sqlContext.sql(.....SELECT * FROM products WHERE name LIKE \\'Pen %\\' ORDER BY price DESC.....) results. showQ Step 3 : Select all the products , whose name starts with Pen and results should be order by Price descending order and quantity ascending order. val results = sqlContext.sql(\\'.....SELECT * FROM products WHERE name LIKE \\'Pen %\\' ORDER BY price DESC, quantity.....) results. showQ Step 4 : Select top 2 products by price val results = sqlContext.sql(.....SELECT\\' FROM products ORDER BY price desc LIMIT2.....} results. show()

QUESTION 5

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, \\'"Data Science"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(\\') single quote and separate of the field should be (-} and line needs to be terminated by : (colon).

3.

If data itself contains the " (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.

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

Solution :

Step 1 : Connect to mysql database.

mysql --user=retail_dba -password=cloudera

show databases; use retail_db; show tables;

Insert record

Insert into departments values(9999, \\'"Data Science"\\');

select" from departments;

Step 2 : Import data as per requirement.

sqoop import \

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

~username=retail_dba \

--password=cloudera \

-table departments \

-target-dir /user/cloudera/departments_enclosedby \

-enclosed-by V -escaped-by \\ -fields-terminated-by--\\' -lines-terminated-by :

Step 3 : Check the result.

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