# 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.geekcert.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 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;

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

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 -ls /user/hive/warehouse/departments_hive01

hdfs dfs -cat/user/hive/warehouse/departments_hive01/part"

Check data in hive table.

Select * from departments_hive01;

**QUESTION 2**

Problem Scenario 51 : You have been given below code snippet.

val a = sc.parallelize(List(1, 2,1, 3), 1)

val b = a.map((_, "b"))

val c = a.map((_, "c"))

Operation_xyz

Write a correct code snippet for Operationxyz which will produce below output.

Output:

Array[(lnt, (lterable[String], lterable[String]))] = Array(

(2,(ArrayBuffer(b),ArrayBuffer(c))),

(3,(ArrayBuffer(b),ArrayBuffer(c))),

(1,(ArrayBuffer(b, b),ArrayBuffer(c, c))) )

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

Solution :

b.cogroup(c).collect

cogroup [Pair], groupWith [Pair]

A very powerful set of functions that allow grouping up to 3 key-value RDDs together using

their keys.

Another example

val x = sc.parallelize(List((1, "apple"), (2, "banana"), (3, "orange"), (4, "kiwi")), 2)

val y = sc.parallelize(List((5, "computer"), (1, "laptop"), (1, "desktop"), (4, "iPad")), 2)

x.cogroup(y).collect

Array[(lnt, (lterable[String], lterable[String]))] = Array(

(4,(ArrayBuffer(kiwi),ArrayBuffer(iPad))),

(2,(ArrayBuffer(banana),ArrayBuffer())),

(3,(ArrayBuffer(orange),ArrayBuffer())),

(1 ,(ArrayBuffer(apple),ArrayBuffer(laptop, desktop))),

(5,{ArrayBuffer(),ArrayBuffer(computer))))

**QUESTION 3**

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: Iterator[(Int)]): Iterator[String] = {

iter.toList.map(x => "[partID:" + index + ", val: " + x + "]").iterator

}

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

z.mapPartitionsWithIndex(myfunc).collect

res28: Array[String] = Array([partID:0, val: 1], [partID:0, val: 2], [partID:0, val: 3], [partID:1,

val: 4], [partID:1, val: S], [partID: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.

Correct Answer: z.aggregate(5)(math.max(_, J, _ + _)

**QUESTION 4**

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.

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

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 5**

Problem Scenario 28 : You need to implement near real time solutions for collecting information when submitted in file with below

Data

echo "IBM,100,20160104" >> /tmp/spooldir2/.bb.txt echo "IBM,103,20160105" >> /tmp/spooldir2/.bb.txt mv /tmp/spooldir2/.bb.txt /tmp/spooldir2/bb.txt After few mins echo "IBM,100.2,20160104" >> /tmp/spooldir2/.dr.txt echo "IBM,103.1,20160105" >> /tmp/spooldir2/.dr.txt mv /tmp/spooldir2/.dr.txt /tmp/spooldir2/dr.txt You have been given below directory location (if not available than create it) /tmp/spooldir2 . As soon as file committed in this directory that needs to be available in hdfs in /tmp/flume/primary as well as /tmp/flume/secondary location. However, note that/tmp/flume/secondary is optional, if transaction failed which writes in this directory need not to be rollback. Write a flume configuration file named flumeS.conf and use it to load data in hdfs with following additional properties .

1.

Spool /tmp/spooldir2 directory

2.

File prefix in hdfs sholuld be events

3.

File suffix should be .log

4.

If file is not committed and in use than it should have _ as prefix.

5.

Data should be written as text to hdfs

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

Solution : Step 1 : Create directory mkdir /tmp/spooldir2 Step 2 : Create flume configuration file, with below configuration for source, sink and channel and save it in flume8.conf. agent1 .sources = source1 agent1.sinks = sink1a

sink1bagent1.channels = channel1a channel1b agent1.sources.source1.channels = channel1a channel1b agent1.sources.source1.selector.type = replicating agent1.sources.source1.selector.optional = channel1b agent1.sinks.sink1a.channel = channel1a agent1 .sinks.sink1b.channel = channel1b agent1.sources.source1.type = spooldir agent1 .sources.sourcel.spoolDir = /tmp/spooldir2 agent1.sinks.sink1a.type = hdfs agent1 .sinks, sink1a.hdfs. path = /tmp/flume/primary agent1 .sinks.sink1a.hdfs.tilePrefix = events agent1 .sinks.sink1a.hdfs.fileSuffix = .logagent1 .sinks.sink1a.hdfs.fileType = Data Stream agent1 .sinks.sink1b.type = hdfs agent1 .sinks.sink1b.hdfs.path = /tmp/flume/secondary agent1 .sinks.sink1b.hdfs.filePrefix = events agent1.sinks.sink1b.hdfs.fileSuffix = .log agent1 .sinks.sink1b.hdfs.fileType = Data Stream agent1.channels.channel1a.type = file agent1.channels.channel1b.type = memory step 4 : Run below command which will use this configuration file and append data in hdfs. Start flume service: flume-ng agent -conf /home/cloudera/flumeconf -conf-file /home/cloudera/flumeconf/flume8.conf --name age Step 5 : Open another terminal and create a file in /tmp/spooldir2/ echo "IBM,100,20160104" » /tmp/spooldir2/.bb.txt echo "IBM,103,20160105" » /tmp/spooldir2/.bb.txt mv /tmp/spooldir2/.bb.txt /tmp/spooldir2/bb.txt After few mins echo "IBM.100.2,20160104" »/tmp/spooldir2/.dr.txt echo "IBM,103.1,20160105" » /tmp/spooldir2/.dr.txt mv /tmp/spooldir2/.dr.txt /tmp/spooldir2/dr.txt