



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 5 : You have been given following mysql database details.

user=retail_dba password=cloudera database=retail_db jdbc URL = jdbc:mysql://quickstart:3306/retail_db Please accomplish following activities.

1.

List all the tables using sqoop command from retail_db

2.

Write simple sqoop eval command to check whether you have permission to read database tables or not.

3.

Import all the tables as avro files in /user/hive/warehouse/retail cca174.db

4.

Import departments table as a text file in /user/cloudera/departments.

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

Solution:

Step 1 : List tables using sqoop

```
sqoop list-tables --connect jdbc:mysql://quickstart:3306/retail_db --username retail_dba password cloudera
```

Step 2 : Eval command, just run a count query on one of the table.

```
sqoop eval \
```

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

```
-username retail_dba \
```

```
-password cloudera \
```

```
--query "select count(1) from ordeMtems"
```

Step 3 : Import all the tables as avro file.

```
sqoop import-all-tables \
```

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

```
-username=retail_dba \
```

```
-password=cloudera \
```

```
-as-avrodatafile \
```



```
-warehouse-dir=/user/hive/warehouse/retail stage.db \
```

```
-ml
```

Step 4 : Import departments table as a text file in /user/cloudera/departments

```
sqoop import \
```

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

```
-username=retail_dba \
```

```
-password=cloudera \
```

```
-table departments \
```

```
-as-textfile \
```

```
-target-dir=/user/cloudera/departments
```

Step 5 : Verify the imported data.

```
hdfs dfs -ls /user/cloudera/departments
```

```
hdfs dfs -ls /user/hive/warehouse/retailstage.db
```

```
hdfs dfs -ls /user/hive/warehouse/retail_stage.db/products
```

QUESTION 2

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" from departments_hive

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

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 (\001). Hence, while importing data we have to provide proper

delimiter.

```
sqoop import \
```

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

```
~username=retail_dba \
```

```
-password=cloudera \
```

```
--table departments \
```

```
--hive-home /user/hive/warehouse \
```

```
-hive-import \
```

```
-hive-overwrite \
```

```
--hive-table departments_hive \
```

```
--fields-terminated-by '\\001\\'
```

Step 3 : Check-the data in directory.

```
hdfs dfs -ls /user/hive/warehouse/departments_hive
```

```
hdfs dfs -cat/user/hive/warehouse/departmentshive/part\\'
```

Check data in hive table.

```
Select * from departments_hive;
```

QUESTION 3

Problem Scenario 95 : You have to run your Spark application on yarn with each executor Maximum heap size to be 512MB and Number of processor cores to allocate on each executor will be 1 and Your main application required three values as input arguments V1 V2 V3. Please replace XXX, YYY, ZZZ `./bin/spark-submit -class com.hadoopexam.MyTask --master yarn-cluster--num-executors 3 --driver-memory 512m XXX YYY lib/hadoopexam.jarZZZ`

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

Solution

XXX: `-executor-memory 512m` YYY: `-executor-cores 1` ZZZ : V1 V2 V3 Notes : spark-submit on yarn options Option Description archives Comma-separated list of archives to be extracted into the working directory of each executor. The path must be globally visible inside your cluster; see Advanced Dependency Management. executor-cores Number of processor cores to allocate on each executor. Alternatively, you can use the `spark.executor.cores` property, executor-memory Maximum heap size to allocate to each executor. Alternatively, you can use the `spark.executor.memory-`



property. num-executors Total number of YARN containers to allocate for this application. Alternatively, you can use the spark.executor.instances property. queue YARN queue to submit to. For more information, see Assigning Applications and Queries to Resource Pools. Default: default.

QUESTION 4

Problem Scenario 60 : You have been given below code snippet.

```
val a = sc.parallelize(List("dog", "salmon", "salmon", "rat", "elephant"), 3)
val b = a.keyBy(_.length)
val c = sc.parallelize(List("dog", "cat", "gnu", "salmon", "rabbit", "turkey", "woif", "bear", "bee"), 3)
val d = c.keyBy(_.length)
operation1
```

Write a correct code snippet for operation1 which will produce desired output, shown below.

```
Array[(Int, (String, String))] = Array((6,(salmon,salmon)), (6,(salmon,rabbit)),
(6,(salmon,turkey)), (6,(salmon,salmon)), (6,(salmon,rabbit)),
(6,(salmon,turkey)), (3,(dog,dog)), (3,(dog,cat)), (3,(dog,gnu)), (3,(dog,bee)), (3,(rat,dog)),
(3,(rat,cat)), (3,(rat,gnu)), (3,(rat,bee)))
```

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

solution: b.join(d).collect join [Pair]: Performs an inner 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. The result of the function becomes the data item becomes the key and the original value of the newly created tuples.

QUESTION 5

Problem Scenario 65 : You have been given below code snippet.

```
val a = sc.parallelize(List("dog", "cat", "owl", "gnu", "ant"), 2)
val b = sc.parallelize(1 to a.count.toInt, 2)
val c = a.zip(b)
operation1
```

Write a correct code snippet for operation1 which will produce desired output, shown below.

```
Array[(String, Int)] = Array((owl,3), (gnu,4), (dog,1), (cat,2), (ant,5))
```

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



Solution : `c.sortByKey(false).collect`

`sortByKey [Ordered]` : This function sorts the input RDD's data and stores it in a new RDD.

"The output RDD is a shuffled RDD because it stores data that is output by a reducer

which has been shuffled. The implementation of this function is actually very clever.

First, it uses a range partitioner to partition the data in ranges within the shuffled RDD.

Then it sorts these ranges individually with `mapPartitions` using standard sort mechanisms.

[Latest CCA175 Dumps](#)

[CCA175 PDF Dumps](#)

[CCA175 Practice Test](#)