CCA175$^{\text{Q\&As}}$

CCA Spark and Hadoop Developer Exam

Pass Cloudera CCA175 Exam with 100% Guarantee

Following Questions and Answers are all new published by Cloudera Official Exam Center

QUESTION 1

Problem Scenario 64 : 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(Ust("dog","cat","gnu","salmon","rabbit","turkey","wolf","bear","bee"), 3)

val d = c.keyBy(_.length)

operation1

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

Array[(lnt, (Option[String], String))] = Array((6,(Some(salmon),salmon)),

(6,(Some(salmon),rabbit}}, (6,(Some(salmon),turkey)), (6,(Some(salmon),salmon)),

(6,(Some(salmon),rabbit)), (6,(Some(salmon),turkey)), (3,(Some(dog),dog)),

(3,(Some(dog),cat)), (3,(Some(dog),gnu)), (3,(Some(dog),bee)), (3,(Some(rat),

(3,(Some(rat),cat)), (3,(Some(rat),gnu)), (3,(Some(rat),bee)), (4,(None,wo!f)),

(4,(None,bear)))

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

solution : b.rightOuterJqin(d).collect rightOuterJoin [Pair] : Performs an right outer join using two key-value RDDs. Please note that the keys must be generally comparable to make this work correctly.

QUESTION 2

Problem Scenario 11 : 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.

Import departments table in a directory called departments.

2.

Once import is done, please insert following 5 records in departments mysql table.

Insert into departments(10, physics);

Insert into departments(11, Chemistry);

Insert into departments(12, Maths);

Insert into departments(13, Science);

Insert into departments(14, Engineering);

3.

Now import only new inserted records and append to existring directory . which has been created in first step.

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

Solution :

Step 1 : Clean already imported data. (In real exam, please make sure you dont delete

data generated from previous exercise).

```
hadoop fs -rm -R departments
```

Step 2 : Import data in departments directory.

```
sqoop import \

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

--username=retail_dba \

-password=cloudera \

-table departments \

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

Step 3 : Insert the five records in departments table.

```
mysql -user=retail_dba --password=cloudera retail_db

Insert into departments values(10, "physics"); Insert into departments values(11,

"Chemistry"); Insert into departments values(12, "Maths"); Insert into departments

values(13, "Science"); Insert into departments values(14, "Engineering"); commit;

select\\' from departments;
```

Step 4 : Get the maximum value of departments from last import, hdfs dfs -cat

/user/cloudera/departments/part* that should be 7

Step 5 : Do the incremental import based on last import and append the results.

```
sqoop import \

--connect "jdbc:mysql://quickstart.cloudera:330G/retail_db" \

~username=retail_dba \

-password=cloudera \
-table departments \

--target-dir /user/cloudera/departments \

-append \

-check-column "department_id" \

-incremental append \

-last-value 7
```

Step 6 : Now check the result.

```
hdfs dfs -cat /user/cloudera/departments/part"
```

QUESTION 3

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

Check data in hive table.

```
Select * from departments_hive;
```

QUESTION 4

Problem Scenario 54 : You have been given below code snippet.

```
val a = sc.parallelize(List("dog", "tiger", "lion", "cat", "panther", "eagle"))
val b = a.map(x => (x.length, x))
operation1
```

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

```
Array[(Int, String)] = Array((4,lion), (7,panther), (3,dogcat), (5,tigereagle))
```

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

Solution :

```
b.foidByKey("")(_ + J.collect
```

foldByKey [Pair]

Very similar to fold, but performs the folding separately for each key of the RDD. This function is only available if the RDD consists of two-component tuples

Listing Variants

```
def foldByKey(zeroValue: V)(func: (V, V) => V): RDD[(K, V}]
```

def foldByKey(zeroValue: V, numPartitions: Int)(func: (V, V) => V): RDD[(K, V)]

def foldByKey(zeroValue: V, partitioner: Partitioner)(func: (V, V) => V): RDD[(K, V}]

QUESTION 5

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[(Int, Seq[String])] = Array((4,ArrayBuffer(lion)), (6,ArrayBuffer(spider)),

(3,ArrayBuffer(dog, cat)), (5,ArrayBuffer(tiger, eagle}}}

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

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, Iterable[V]}]

def groupByKey(numPartittons: Int): RDD[(K, Iterable[V])]

def groupByKey(partitioner: Partitioner): RDD[(K, Iterable[V])]

CCA175 PDF Dumps CCA175 Practice Test CCA175 Braindumps