JIGSAW ACADEMY
THE ONLINE SCHOOL OF ANALYTICS

# Non-Graded Assignment

## PIG-2 Solutions

Q1)
Write a Pig UDF function in Java which will reverse the sequence of given last name and first name. For example if the name passed is *Agarwal,Amit* then the UDF should return *Amit Agarwal* as the return value.

You are given a file Name-Data.txt. Use the Pig UDF (developed as above) in a Pig script to reverse all the names in the input file.

Note: This assignment is assumes more than basic level of Java programming skills, so it is meant for Java developers with intermediate level of skills.

**Solution:**

Pig UDF Java Source Code: Save this code in a file with name `revstring.java` as the class name used in the code below is `revstring`. You can compile the code and export it into a Jar file of any name of your choice and remember to use the same name in your Pig script for registering the Jar file. The Jar file name used in this example is: `pigudfrevstring.jar`

➔

```
package com.jigsawacademy;

import java.io.IOException;
import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;

public class revstring extends EvalFunc<String> {

public String exec(Tuple input) throws IOException {
        try {
            String str = (String)input.get(0);
            String[] newname = str.split(",");
            return newname[1] + " " + newname[0];
        }

        catch( Exception ex)
        {
            throw new IOException(ex.getMessage());
        }
```

© Jigsaw Academy Pvt Ltd

```
    }

}
```

Pig Script:

Note: You must write this code in Eclipse IDE and convert it into JAR file. The procedure is already discussed in MapReduce Module. Ensure that you mention the correct paths for the Jar file (`pigudfrevstring.jar`) and the input file (`Name-Data.txt`).

```
register <path on your system>/pigudfrevstring.jar
names= load 'Name-Data.txt'  as (val:chararray);
dump names;
result = foreach names generate
com.jigsawacademy.revstring(val);
dump result;
```

© Jigsaw Academy Pvt Ltd

Q2)

For all Open Georgia Salary/Travel data provided as CSV file with this assignment for the Fiscal Year 2010 and Organization Type of Local Boards of Education, produce a distinct list of all Job Titles along with the total number of employees aligned with each Job Title & the minimum/maximum/average salaries for each of the identified Job Titles.

Expected Steps:

1. Store the given input file salaryTravelReport.csv into the HDFS Location and save the given piggybank.jar file in your disk.
2. Load up the base UDF library – piggybank and get a handle on the REPLACE function.
    - This jar file `piggybank.jar` contains a class `org.apache.pig.piggybank.evaluation.string.REPLACE` which can replace any substring from the given string.
    - Open the grunt console and execute the following commands:

```
register "<Path>/piggybank.jar";
define REPLACE
org.apache.pig.piggybank.evaluation.string.REPLACE();
```

3. Create your Pig commands and execute in the console, hints are given below:
    - Load the salary file and declare its structure
    - Loop through the input data to clean up the number fields. Take out the commas from the salary and travel fields and cast to a float
    - Trim down to just Local Boards of Education
    - Further trim it down to just be for the year in question
    - Bucket them up by the job title
    - Loop through the titles and check how many are there under each title
    - Determine the minimum, maximum and average salaries for every title
    - Guarantee the order on the way out
    - Dump the results on the console
    - Save results back to HDFS

**Solution:**

Note:

Ensure that you give the correct paths for the `piggbank.jar` file and the input data file (`SalaryTravelReport_Sample.csv`) while loading them.

Also remember to copy the input data file to a sub-directory within your default directory on HDFS.

© Jigsaw Academy Pvt Ltd

Code:

```
 -- load up the base UDF (piggybank) and get a handle on the REPLACE
function

register /home/data/piggybank.jar;
define REPLACE org.apache.pig.piggybank.evaluation.string.REPLACE();

 -- load the salary file and declare its structure

inputFile = LOAD 'piginput/salaryTravelReport.csv' using
org.apache.pig.piggybank.storage.CSVExcelStorage()  as
(name:chararray, title:chararray, salary:chararray,
travel:chararray, orgType:chararray, org:chararray, year:int);

 -- loop through the input data to clean up the number fields a bit

cleanedUpNumbers = foreach inputFile GENERATE name as name, title as
title, (float)REPLACE(salary, ',','') as salary,
(float)REPLACE(travel, ',','') as travel, orgType as orgType, org as
org, year as year;

 -- trim down to just Local Boards of Education

onlySchoolBoards = filter cleanedUpNumbers by orgType == 'LBOE';

 -- further trim it down to just be for the year in question

onlySchoolBoardsFor2010 = filter onlySchoolBoards by year == 2010;

 -- bucket them up by the job title

byTitle = GROUP onlySchoolBoardsFor2010 BY title;

 -- loop through the titles and for each one.

salaryBreakdown = FOREACH byTitle GENERATE group as title,
COUNT(onlySchoolBoardsFor2010), MIN(onlySchoolBoardsFor2010.salary),
MAX(onlySchoolBoardsFor2010.salary),
AVG(onlySchoolBoardsFor2010.salary);

 -- guarantee the order on the way out

sortedSalaryBreakdown = ORDER salaryBreakdown by title;

DUMP sortedSalaryBreakdown;

 -- save results back to HDFS STORE

STORE sortedSalaryBreakdown into
'pigassignout/opengeorgia/pigoutput';
```

© Jigsaw Academy Pvt Ltd

Q3)

Airline Data Analysis with Pig: Travel industry generates huge amounts of data every day. Refer Airline_data_schema for the Table description.

Write Pig scripts for the following Queries:

Question 3A: Find out top 20 airports by total volume of flights. What are the busiest airports by total flight traffic? JFK will feature, but what are the others? For each airport code compute the number of inbound, outbound and all flights. Compute the top 20 airports per month per year based on total traffic (inbound + outbound).

**Solution:**

Q3A:

Note:

Make sure you copy the data file `AirlinesData.csv` on to HDFS preferably to a sub-directory in your default directory i.e. `/user/<your login>/`

Code:

```
Step 1: Loading the data

RAW_DATA = LOAD
'/home/hduser/Desktop/pig_case_study/case_study2/AirlinesData.csv'
USING PigStorage(',') AS (year: int, month: int, day: int, dow: int,
dtime: int, sdtime: int, arrtime: int, satime: int, carrier:
chararray,fn: int, tn: chararray, etime: int, setime: int, airtime:
int, adelay: int, ddelay: int, scode: chararray,dcode:chararray,dist:
int, tintime: int, touttime: int, cancel: chararray, cancelcode:
chararray, diverted: int, cdelay: int, wdelay: int, ndelay: int,
sdelay: int, latedelay: int);

Step 2: Grouping the data based on the source airport code (scode),
dcode stands for DESTINATION code/arrival code (Group by the IATA code
of the departure airport/source airport)

SOURCE_IATA_GROUP = GROUP RAW_DATA BY (scode,month);

Step 3:

Note: counting the total number of tuples in the above grouped (BAG)
would give us the count of outbound traffic for that airport
code/source code/departure airport code

OUTBOUND_IATA_COUNT = FOREACH SOURCE_IATA_GROUP GENERATE group as IATA,
COUNT(RAW_DATA) AS num_out_flights;

Step 4:
```

© Jigsaw Academy Pvt Ltd

```
total_out = foreach OUTBOUND_IATA_COUNT generate
flatten(IATA),num_out_flights ;

dump total_out
```

(first column is airport departure code, month, total count of flights taking off from that airport for that month) ----------(IAD,1,25) (LAX,1,2) (PDX,1,3)
Do the same thing (step 2, 3 and 4) for inbound flights for and store it into a different table.

```
DEST_IATA_GROUP = GROUP RAW_DATA BY (dcode,month);
INBOUND_IATA_COUNT = FOREACH DEST_IATA_GROUP GENERATE group as IATA,
COUNT(RAW_DATA) AS num_in_flights;

total_in = foreach INBOUND_IATA_COUNT generate
flatten(IATA),num_in_flights ;
dump total_in
```

(sample output) ----------ORD,1,27 SFO,1,3 IAD,1,34 LAX,1,4 PDX,2,34
HINT: We need to do a union operation on these 2 tables!

```
unioned = union total_in, total_out;
grouped = group unioned by ($0, $1);

dump grouped ------------
```

(sample output would look like this) ((IAD,1),{(IAD,1,25),(IAD,1,34)}) ((LAX,1),{(LAX,1,2),(LAX,1,4)}) ((ORD,1),{(ORD,1,27)}) ((PDX,1),{(PDX,1,3)}) ((PDX,2),{(PDX,2,34)}) ((SFO,1),{(SFO,1,3)})

We have now got monthwise grouping of each airport's inbound and outbound flight data!

STEP 5:

(Finally counting the 3rd field of each tuple in every bag)

```
describe grouped

final_count = foreach grouped generate group, SUM(unioned.$2) ;
dump final_count;
```

(THE SAMPLE O/P WOULD LOOK LIKE THIS)
((IAD,1),59.0) ((LAX,1),6.0) ((ORD,1),27.0) ((PDX,1),3.0) ((PDX,2),34.0) ((SFO,1),3.0)
NOW flatten the above op

```
describe final_count;
flat_data = foreach final_count generate flatten(group), $1 ;
dump flat_data
```

© Jigsaw Academy Pvt Ltd

```
(IAD,1,59.0) (LAX,1,6.0) (ORD,1,27.0) (PDX,1,3.0) (PDX,2,34.0)
(SFO,1,3.0) (SFO,4,340) (IAD,2,200) (JFK,1,300) (JFK,1,100)
```

Now we have a month wise total of inbound and outbound count for every airport code.

```
describe flat_data;
grouped = group flat_data by $1;
dump grouped;
```

SAMPLE OUTPUT could look like this ----------

```
(1,{(IAD,1,59.0),(LAX,1,6.0),(ORD,1,27.0),(PDX,1,3.0),(SFO,1,3.0)})
(2,{(PDX,2,34.0)})
```

```
describe grouped;
```

We now have a table of monthwise collection of list of all airports with total traffic for that month. According to the problem statement we need to just pick the top 20 tuples from each bag, we use a built-in function called TOP

```
top3 = foreach grouped generate group, TOP(3,2,flat_data) ;
```

NOTE: TOP (3,2, flat_data) means ..
From the ordered bag, pick only top 3 tuples (selection based on field number 2 in the flat_data table)

```
dump top3;
```

© Jigsaw Academy Pvt Ltd

Question 3B: Carrier Popularity – Some carriers come and go, others demonstrate regular growth. Compute the (log base 10) volume -- total flights -- over each year, by each carrier. The carriers are ranked by their median volume (over the 10-year span).

**Solution:**

Q3B:

Note:

Make sure you copy the data file `AirlinesData.csv` on to HDFS preferably to a sub-directory in your default directory i.e. `/user/<your login>/`

Code:

```
RAW_DATA = LOAD '/user/<your login>/piginput/AirlinesData.csv' USING
PigStorage(',') AS
(year:int,month:int,day:int,dow:int,
dtime:int,sdtime:int,arrtime:int,satime:int,
carrier:chararray,fn:int,tn:chararray,
etime:int,setime:int,airtime:int,
adelay:int,ddelay:int,
scode:chararray,dcode:chararray,dist:int,
tintime:int,touttime:int,
cancel:chararray,cancelcode:chararray,diverted:int,
cdelay:int,wdelay:int,ndelay:int,sdelay:int,latedelay:int);

CARRIER_DATA = FOREACH RAW_DATA GENERATE month AS m, carrier AS
cname;

GROUP_CARRIERS = GROUP CARRIER_DATA BY (m,cname);

COUNT_CARRIERS = FOREACH GROUP_CARRIERS GENERATE FLATTEN(group),
LOG10(COUNT(CARRIER_DATA)) AS popularity;

dump COUNT_CARRIERS

OUTPUTDATA = order COUNT_CARRIERS by popularity DESC;

dump OUTPUTDATA
```

© Jigsaw Academy Pvt Ltd