**Assignment 4.2**

Considering a sample data set of student\_details as below:

001,Rajiv,Reddy,21,9848022337,Hyderabad,89

002,siddarth,Battacharya,22,9848022338,Kolkata,78

003,Rajesh,Khanna,22,9848022339,Delhi,90

004,Preethi,Agarwal,21,9848022330,Pune,93

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75

006,Archana,Mishra,23,9848022335,Chennai,87

007,Komal,Nayak,24,9848022334,trivendram,83

008,Bharathi,Nambiayar,24,9848022333,Chennai,72

Loading the data set into Pig with below command;

grunt> student\_details = LOAD '/home/acadgild/pig/student\_details.txt' USING PigStorage(',')

as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray, gpa:int);

>> DUMP student\_details

* Pig Commands Used on above data set are explained below :

1. **Concat**:

grunt> student\_name\_concat = foreach student\_details Generate CONCAT (firstname,' ', lastname);

grunt> DUMP student\_name\_concat;

(Rajiv Reddy)

(siddarth Battacharya)

(Rajesh Khanna)

(Preethi Agarwal)

(Trupthi Mohanthy)

(Archana Mishra)

(Komal Nayak)

(Bharathi Nambiayar)

1. **Tokenize:**

grunt> student\_name\_tokenize = foreach student\_details Generate TOKENIZE(firstname);

grunt> DUMP student\_name\_ tokenize;

({(Rajiv)})

({(siddarth)})

({(Rajesh)})

({(Preethi)})

({(Trupthi)})

({(Archana)})

({(Komal)})

({(Bharathi)})

1. **Sum**:

grunt> student\_group = Group student\_details all;

grunt> student\_gpa\_sum = foreach student\_group Generate

>> (student\_details.firstname,student\_details.gpa),SUM(student\_details.gpa);

(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv)},{(72),(83),(87),(75),(93),(90),(78),(89)}),667)

1. **Min:**

grunt> student\_group\_all = Group student\_details All;

grunt> student\_gpa\_min = foreach student\_group\_all Generate

(student\_details.firstname, student\_details.gpa), MIN(student\_details.gpa);

grunt> Dump student\_gpa\_min;

(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv) } ,

{ (72) , (83) , (87) , (75) , (93) , (90) , (78) , (89) }) ,72)

1. **Max:**

grunt> student\_group\_all = Group student\_details All;

grunt> student\_gpa\_max = foreach student\_group\_all Generate

(student\_details.firstname, student\_details.gpa), MAX(student\_details.gpa);

Dump student\_gpa\_max;

(({(Bharathi),(Komal),(Archana),(Trupthi),(Preethi),(Rajesh),(siddarth),(Rajiv) } ,

{ (72) , (83) , (87) , (75) , (93) , (90) , (78) , (89) }) ,93)

1. **Limit:**

grunt> limit\_data = LIMIT student\_details 4;

grunt> Dump limit\_data;

(1,Rajiv,Reddy,21,9848022337,Hyderabad)

(2,siddarth,Battacharya,22,9848022338,Kolkata)

(3,Rajesh,Khanna,22,9848022339,Delhi)

(4,Preethi,Agarwal,21,9848022330,Pune)

1. **Store:**

grunt> STORE student\_details INTO ' /home/acadgild/hadoop/student\_output' USING PigStorage (',');

1. **Distinct:**

grunt> distinct\_data = DISTINCT student\_details;

grunt> Dump distinct\_data;

(1,Rajiv,Reddy,9848022337,Hyderabad)

(2,siddarth,Battacharya,9848022338,Kolkata)

(3,Rajesh,Khanna,9848022339,Delhi)

(4,Preethi,Agarwal,9848022330,Pune)

(5,Trupthi,Mohanthy,9848022336,Bhuwaneshwar)

(6,Archana,Mishra,9848022335,Chennai)

1. **Flatten:**

X = FOREACH C GENERATE group, FLATTEN(A);

DUMP X;

(1,1,2,3)

(4,4,2,1)

(4,4,3,3)

(8,8,3,4)

(8,8,4,3)

1. **IsEmpty:**

SSN = load 'ssn.txt' using PigStorage() as (ssn:long);

SSN\_NAME = load 'students.txt' using PigStorage() as (ssn:long, name:chararray);

-- do a left out join of SSN with SSN\_Name

X = cogroup SSN by ssn inner, SSN\_NAME by ssn;

-- only keep those ssn's for which there is no name

Y = filter X by IsEmpty(SSN\_NAME);