Create a sample dataset and implement the below Pig commands on the same dataset.

Concat 2) Tokenize 3) Sum 4) Min 5) Max 6) Limit 7) Store 8) Distinct 9) Flatten 10) IsEmpty

Dataset file student\_details.txt

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

1.CONCAT

**CONCAT()** is used to concatenate two or more expressions of **same type**.

A = LOAD '/home/acadgild/leena/ass/ass-72/student\_details.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

concatName = FOREACH A GENERATE CONCAT(firstname, ' ', lastname) AS fullName;

Output :

(Rajiv Reddy)

(siddarth Battacharya)

(Rajesh Khanna)

(Preethi Agarwal)

(Trupthi Mohanthy)

(Archana Mishra)

(Komal Nayak)

(Bharathi Nambiayar)

**2)TOKENIZE**

The TOKENIZE() function of Pig Latin is used to **split a string** (which contains a group of words) in a single tuple and returns a bag which contains the output of the split operation.

TOKENIZE(expression [, 'field\_delimiter'])

Using the above relation concatName,

tokenName = FOREACH concatName GENERATE TOKENIZE(fullName);

Output

({(Rajiv),(Reddy)})

({(siddarth),(Battacharya)})

({(Rajesh),(Khanna)})

({(Preethi),(Agarwal)})

({(Trupthi),(Mohanthy)})

({(Archana),(Mishra)})

({(Komal),(Nayak)})

({(Bharathi),(Nambiayar)})

**3)Sum**

Sum is a function of PigLatin which gets the total of numeric values of a column in a **single-column bag**. SUM ignores all null values.

To get the sum of a column, we need to group it first using **GROUP** operator.

Syntax SUM(expression)

In above example, suppose we plan to add the integer field say the age;

We have the relation A

A = LOAD '/home/acadgild/leena/ass/ass-72/student\_details.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

Now, using GROUP

B = GROUP A all;

sumAge = FOREACH B GENERATE SUM(A.age);

Output

(180)

4) Min

The **MIN()** function of Pig Latin is used to get the minimum (lowest) value (numeric or chararray) for a certain column in a single-column bag. Here again it ignores all the null values.

Also, the GROUP is required here.

MIN(expression)

In the above example, using the above example, finding the min age

We have

A = LOAD '/home/acadgild/leena/ass/ass-72/student\_details.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

Now, using GROUP

B = GROUP A all;

minAge = FOREACH B GENERATE (MIN(A.age));

Output : 21

5) Max

The **MAX()** function of Pig Latin is used to get the maximum (highest) value (numeric or chararray) for a certain column in a single-column bag. Here again it ignores all the null values.

Also, the GROUP is required here.

MAX(expression)

In the above example, using the above example, finding the min age

We have

A = LOAD '/home/acadgild/leena/ass/ass-72/student\_details.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

Now, using GROUP

B = GROUP A all;

maxAge = FOREACH B GENERATE (MAX(A.age));

Output : (24)

**6)LIMIT**

The **LIMIT** operator is used to get a limited number of tuples from a relation.

SYNTAX

Relation-name = LIMIT relation-name n; where n is a positive integer.

This is required to view a few tuples rather than listing all of them especially when dealing with large datasets.

We have the relation A

A = LOAD '/home/acadgild/leena/ass/ass-72/student\_details.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

Now, if you are interested in just the first 3, we can use LIMIT 3

firstThree = LIMIT A 3;

Output

(1,Rajiv,Reddy,21,,Hyderabad,89)

(2,siddarth,Battacharya,22,,Kolkata,78)

(3,Rajesh,Khanna,22,,Delhi,90)

7)Store

To store the loaded data in the file system can be accomplished using the **store** operator.

Syntax

STORE relation-name INTO ‘file-path’ [USING function];

For example, we are storing the first 3 students details into a file

We already have the relation

firstThree = LIMIT A 3;

STORE firstThree INTO '/home/acadgild/leena/ass/ass-72/three\_student\_detail' USING PigStorage('\*');

Output

[acadgild@localhost ~]$ ls -l /home/acadgild/leena/ass/ass-72/three\_student\_detail

total 4

-rw-r--r--. 1 acadgild acadgild 98 Jul 1 22:49 part-r-00000

-rw-r--r--. 1 acadgild acadgild 0 Jul 1 22:49 \_SUCCESS

[acadgild@localhost ~]$ cat /home/acadgild/leena/ass/ass-72/three\_student\_detail/part-r-00000

1\*Rajiv\*Reddy\*21\*\*Hyderabad\*89

2\*siddarth\*Battacharya\*22\*\*Kolkata\*78

3\*Rajesh\*Khanna\*22\*\*Delhi\*90

[acadgild@localhost ~]$

8)DISTINCT

The **DISTINCT** operator is used to remove redundant (duplicate) tuples from a relation.

Syntax

Rel\_name = DISTINCT rel;

Suppose we have the student\_details-dup.txt with some duplicate records (I have appended some records)

001,Rajiv,Reddy,9848022337,Hyderabad

002,siddarth,Battacharya,9848022338,Kolkata

002,siddarth,Battacharya,9848022338,Kolkata

003,Rajesh,Khanna,9848022339,Delhi

003,Rajesh,Khanna,9848022339,Delhi

004,Preethi,Agarwal,9848022330,Pune

005,Trupthi,Mohanthy,9848022336,Bhuwaneshwar

006,Archana,Mishra,9848022335,Chennai

006,Archana,Mishra,9848022335,Chennai

E = LOAD '/home/acadgild/leena/ass/ass-72/student\_details-dup.txt' USING PigStorage(',') AS (rollno:int,firstname:chararray,lastname:chararray,age:int,phone:int,place:chararray,pincode:int);

To remove the redundant data

F = DISTINCT E;

(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)

9)FLATTEN

Flatten un-nests tuples and bags.

For tuples, flatten substitutes the fields of a tuple in place of the tuple. For example, consider a relation that has a tuple of the form (a, (b, c)).

GENERATE $0, FLATTEN($1) would result in : (a,b,c)  
Consider an input file sample.txt with some text content

sampleFile = LOAD '/home/acadgild/leena/ass/ass-72/sample.txt' AS (line:chararray);

sampleFileContent = FOREACH sampleFile GENERATE TOKENIZE(line) as sampleText;

Output

({(Swami),(Vivekananda),(was),(born),(on),(12),(January),(1863),(in),(Calcutta.)})

({(Narendranath),(Datta),(was),(his),(original),(name)})

({(His),(father),(Vishwanath),(Datta),(was),(an),(attorney),(at),(the),(Calcutta),(High),(Court.)})

({(Durgacharan),(Datta),(Narendra's),(grandfather),(was),(a),(Sanskrit),(and),(Persian),(scholar),(who),(left),(his),(family),(and),(became),(a),(monk),(at),(age),(twenty-five.),(Narendra's),(mother),(Bhuvaneswari),(Devi),(was),(a),(devout),(housewife.)})

({(The),(progressive),(rational),(attitude),(of),(Narendra's),(father),(and),(the),(religious),(temperament),(of),(his),(mother),(helped),(shape),(his),(thinking),(and),(personality.)})

({(Narendra),(was),(interested),(in),(spirituality),(from),(a),(young),(age),(and),(used),(to),(play),(at),(meditating),(before),(the),(images),(of),(deities),(such),(as),(Shiva),(Rama),(and),(Sita.),(He),(was),(fascinated),(by),(wandering),(ascetics),(and),(monks.Narendra),(was),(naughty),(and),(restless),(as),(a),(child),(and),(his),(parents),(often),(had),(difficulty),(controlling),(him.),(His),(mother),(said),(I),(prayed),(to),(Shiva),(for),(a),(son),(and),(he),(has),(sent),(me),(one),(of),(his),(demons)})

Now applying FLATTEN

eachword = FOREACH sampleFileContent GENERATE FLATTEN(sampleText);

(Swami)

(Vivekananda)

(was)

(born)

(on)

(12)

(January)

(1863)

(in)

(Calcutta.)

(Narendranath)

(Datta)

(was)

(his)

(original)

(name)

(His)

(father)

(Vishwanath) ..... <truncated>

10) IsEmpty()

The **IsEmpty()** function of Pig Latin is used to check if a bag or map is empty.

IsEmpty(expression)

emp\_sales = LOAD '/home/acadgild/leena/ass/ass-72/emp\_sales.txt' USING PigStorage(',') AS (sno:int, name:chararray, age:int, salary:int, dept:chararray);

emp\_bonus = LOAD '/home/acadgild/leena/ass/ass-72/emp\_bonus.txt' USING PigStorage(',') AS (sno:int, name:chararray, age:int, salary:int, dept:chararray);

grunt> cpgroup\_data = COGROUP emp\_sales BY age, emp\_bonus BY age;

grunt> DESCRIBE cpgroup\_data

cpgroup\_data: {group: int,emp\_sales: {(sno: int,name: chararray,age: int,salary: int,dept: chararray)},emp\_bonus: {(sno: int,name: chararray,age: int,salary: int,dept: chararray)}}

Output

(22,{(6,Maggy,22,35000,sales),(1,Robin,22,25000,sales )},{(1,Robin,22,25000,sales )})

(23,{(5,David,23,45000,sales ),(3,Maya,23,25000,sales ),(2,BOB,23,30000,sales )},{(5,David,23,45000,sales ),(3,Maya,23,25000,sales ),(2,Jaya,23,20000,admin )})

(25,{(4,Sara,25,40000,sales )},{(4,Alia,

isempty\_fetch = FILTER cpgroup\_data BY IsEmpty(emp\_sales);

Output

(30,{},{(6,Omar,30,30000,admin)})