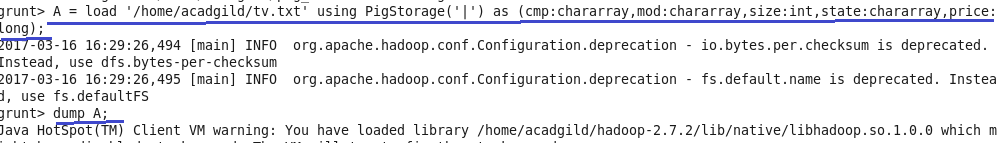
Assignment 10.2

1.Create a sample dataset and implement the below Pig commands on the same dataset. 1) Concat 2) Tokenize 3) Sum 4) Min 5) Max 6) Limit 7) Store 8) Distinct 9) Flatten 10) IsEmpty





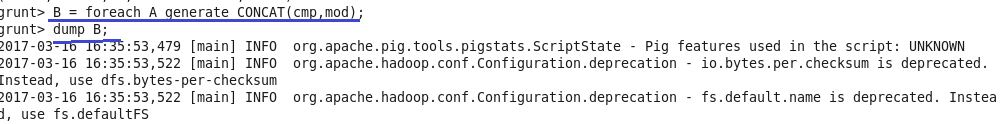
1.Concat

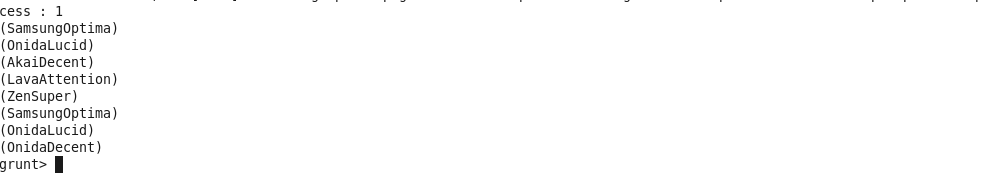
Concatenates two fields of type chararray or two fields of type bytearray.The data type of the two elements must be the same, either chararray or bytearray.

Syntax

CONCAT(expression,expression)

An expression with data types chararray or bytearray.





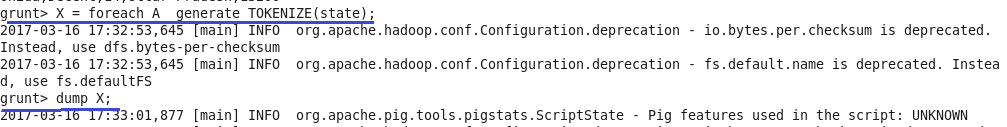
2.Tokenize

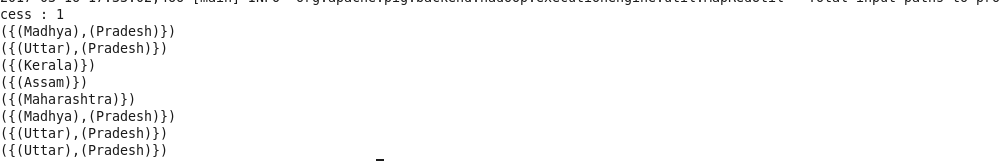
It is used to split a string of words (all words in a single tuple) into a bag of words (each word in a single tuple). The following characters are considered to be word separators: space, double quote("), coma(,) parenthesis(()), star(\*).

Syntax

TOKENIZE(expression)

An expression with data type chararray.





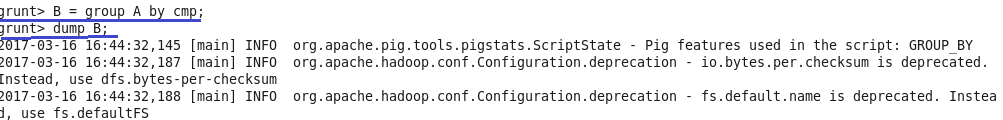
3.Sum

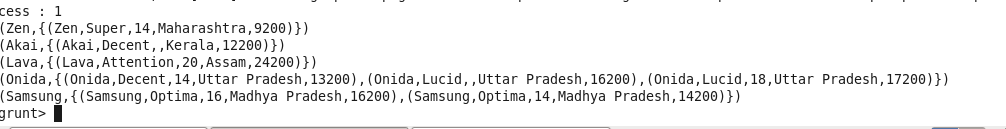
It is used to compute the sum of a set of numeric values in a single-column bag. SUM requires a preceding GROUP ALL statement for global sums and a GROUP BY statement for group sums.

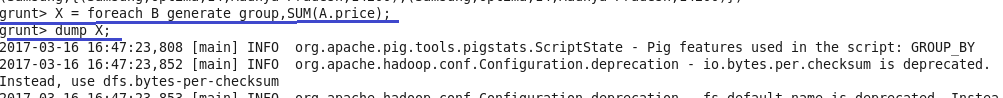
Syntax

SUM(expression)

An expression with data types int, long, float, double, or bytearray cast as double.









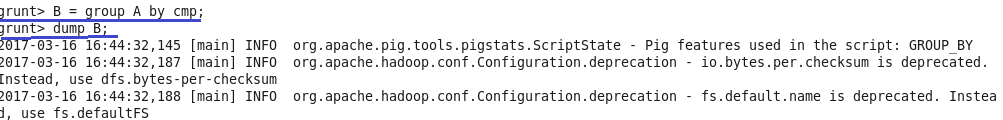
4. Min

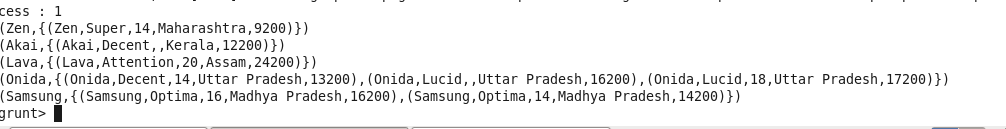
It is used to compute the minimum of a set of numeric values or chararrays in a single-column bag.  MIN requires a preceding GROUP… ALL statement for global minimums and a GROUP … BY statement for group minimums.

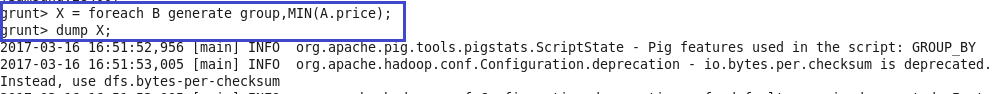
Syntax

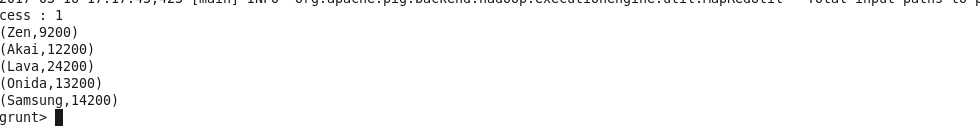
MIN(expression)

An expression with data types int, long, float, double, or chararray.









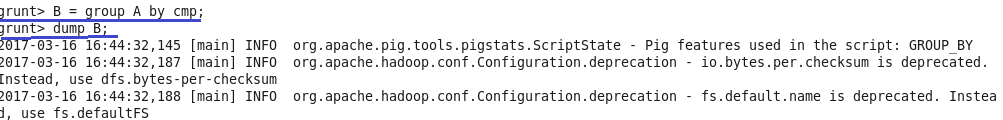
5.Max

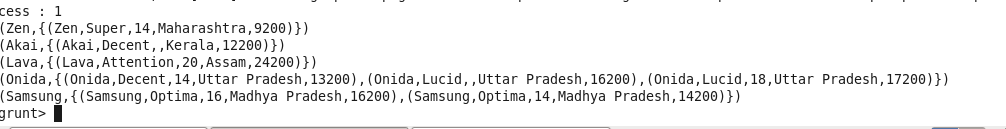
It is used to compute the maximum of the numeric values or chararrays in a single-column bag. MAX requires a preceding GROUP ALL statement for global maximums and a GROUP BY statement for group maximums.

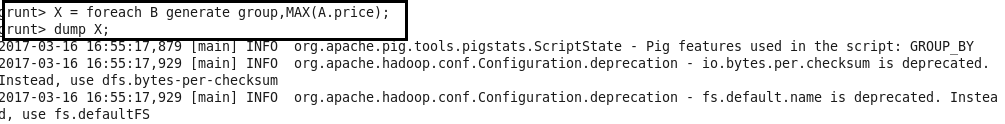
Syntax

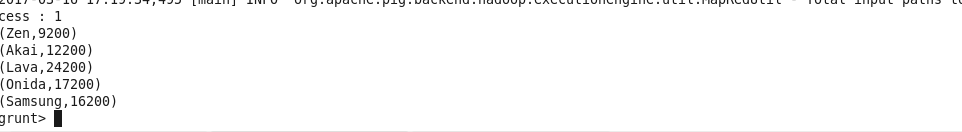
MAX(expression)

An expression with data types int, long, float, double, or chararray.









6.Limit

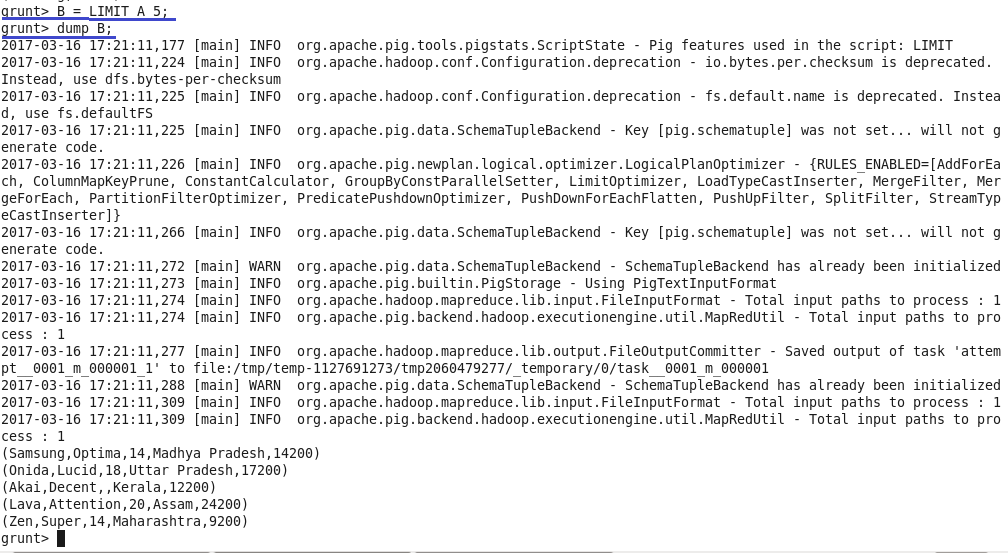
It is used to limit the number of output tuples. If the specified number of output tuples is equal to or exceeds the number of tuples in the relation, the output will include all tuples in the relation.

Syntax

alias = LIMIT alias n

where alias=name of relation

n=number of tuples



7.Store

It is used to run (execute) Pig Latin statements and save (persist) results to the file system.

Syntax

STORE alias INTO 'directory' [USING function];

Alias=name of relation

INTO=Required keyword

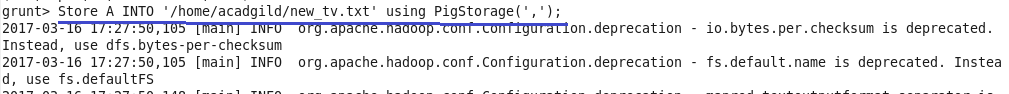
‘directory’= The name of the storage directory, in quotes.

USING Keyword. It is used to name the store function.

If the USING clause is omitted, the default store function PigStorage is used.

Function: the store function

 PigStorage is the default store function and does not need to be specified.





8.Distinct

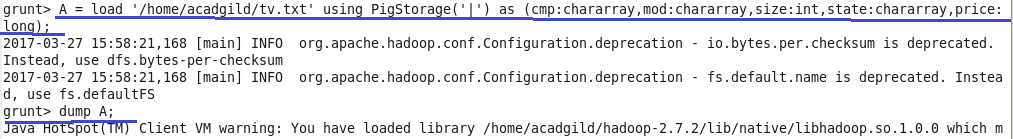
It is used to remove duplicate tuples in a relation.

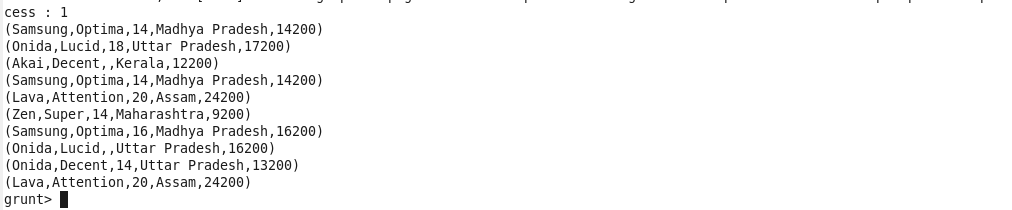
Syntax

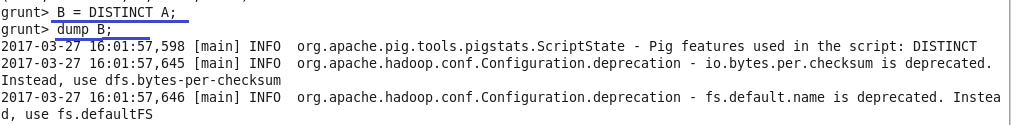
alias = DISTINCT alias [PARALLEL n];

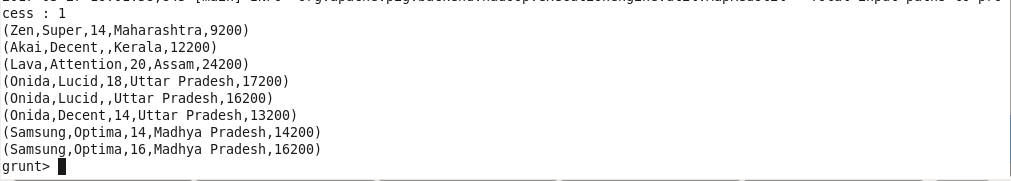
alias=name of relation

PARALLEL n= Increase the parallelism of a job by specifying the number of reduce tasks, n. The default value for n is 1 (one reduce task)









9.Flatten

The FLATTEN operator changes the structure of tuples and bags in a way that a UDF cannot. Flatten un-nests tuples as well as 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)). The expression GENERATE $0, flatten($1), will cause that tuple to become (a, b, c).

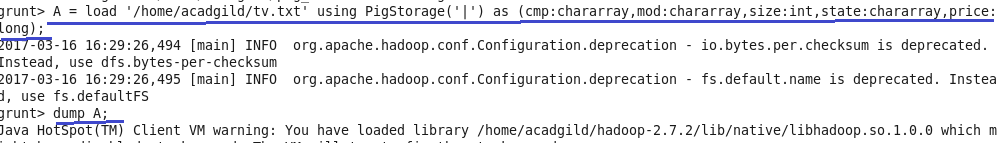
For bags, the situation becomes more complicated. When we un-nest a bag, we create new tuples. If we have a relation that is made up of tuples of the form ({(b,c),(d,e)}) and we apply GENERATE flatten($0), we end up with two tuples (b,c) and (d,e). When we remove a level of nesting in a bag, sometimes we cause a cross product to happen. For example, consider a relation that has a tuple of the form (a, {(b,c), (d,e)}), commonly produced by the GROUP operator. If we apply the expression GENERATE $0, flatten($1) to this tuple, we will create new tuples: (a, b, c) and (a, d, e).

Syntax

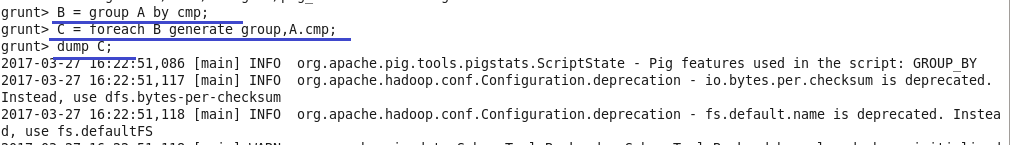
{(data\_type) |  (tuple(data\_type))  | (bag{tuple(data\_type)}) | (map[]) } field

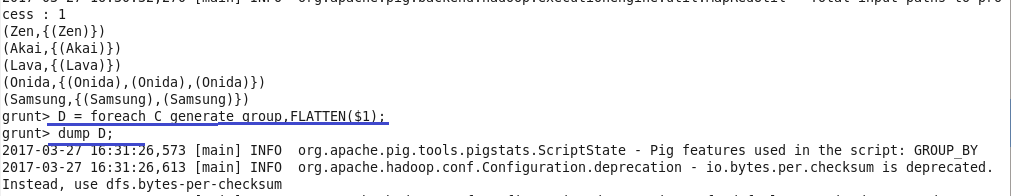
data\_type-The data type you want to cast to, enclosed in parentheses.

Field-The field whose type we want to change.











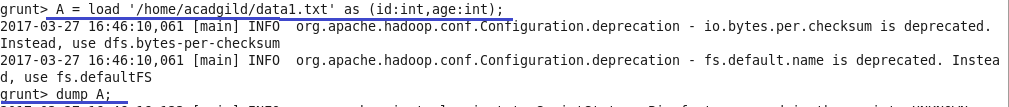
10.IsEmpty

It checks if a bag or map is empty (has no data). The function can be used to filter data.

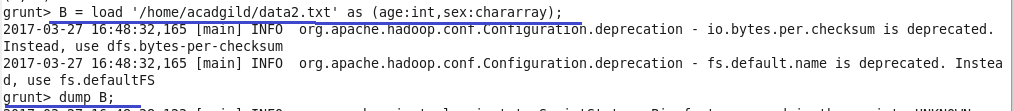
Syntax

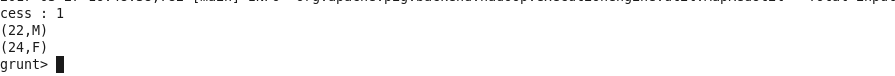
IsEmpty(expression)

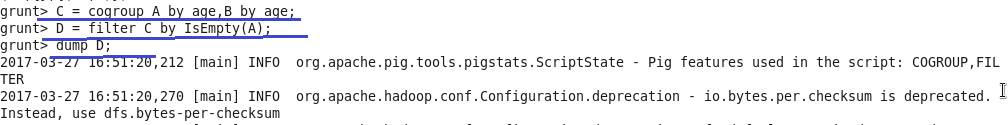
An expression with any datatype.











C:\Users\612883\Desktop\is6.png