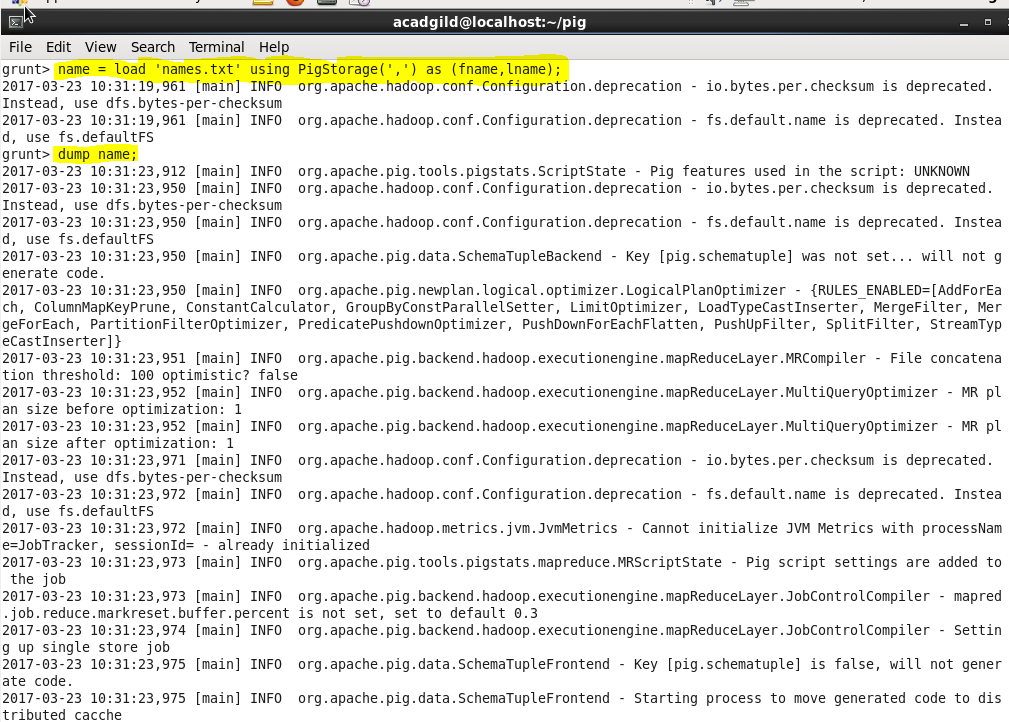
**Assignment 10.2**

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



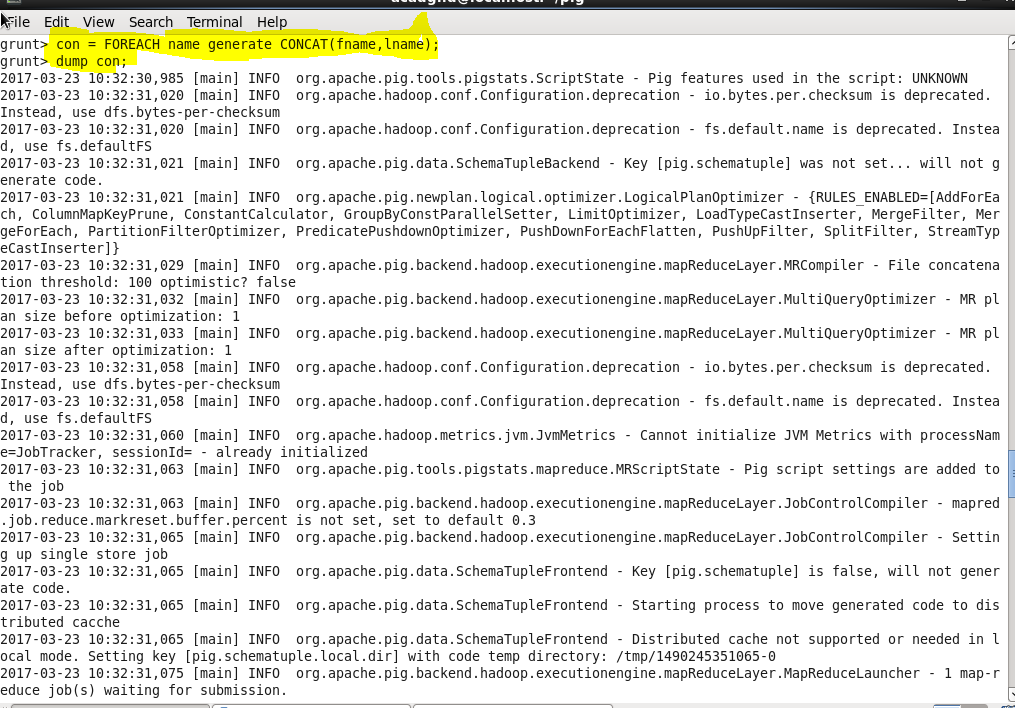


**Concat:**

The CONCAT() function of Pig Latin is used to concatenate two or more expressions of the same type.

**Syntax:**

CONCAT(expression,expression,[expression]);



**Output:**

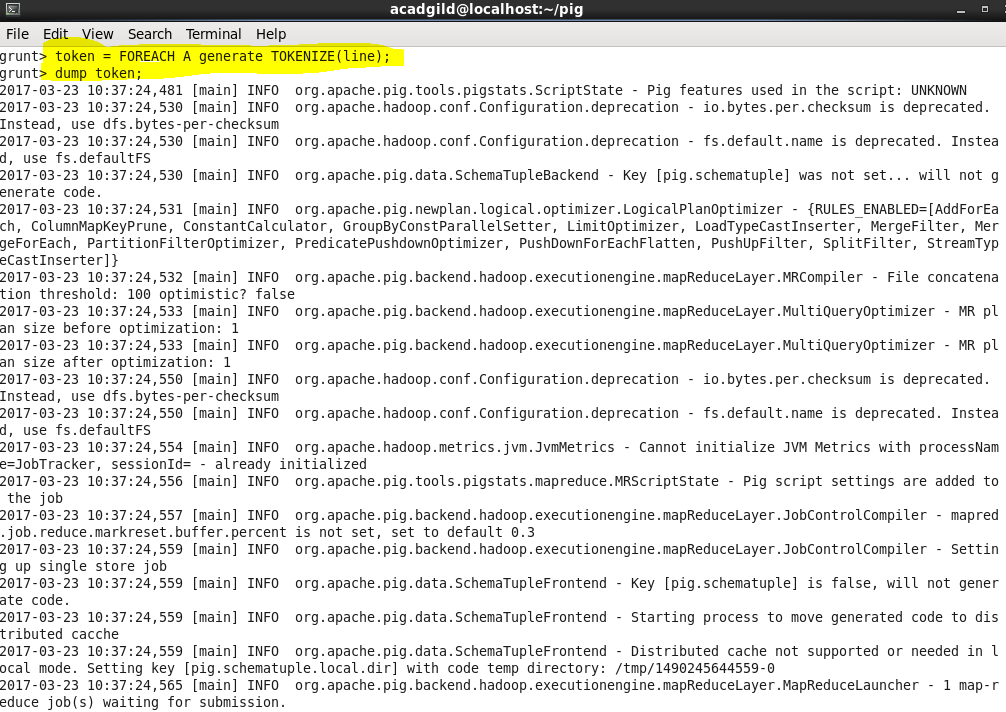


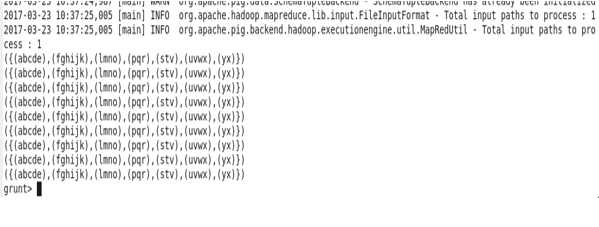
**Tokenize:**

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.

**Syntax:**

TOKENIZE(expression,[field\_delimiter’]);



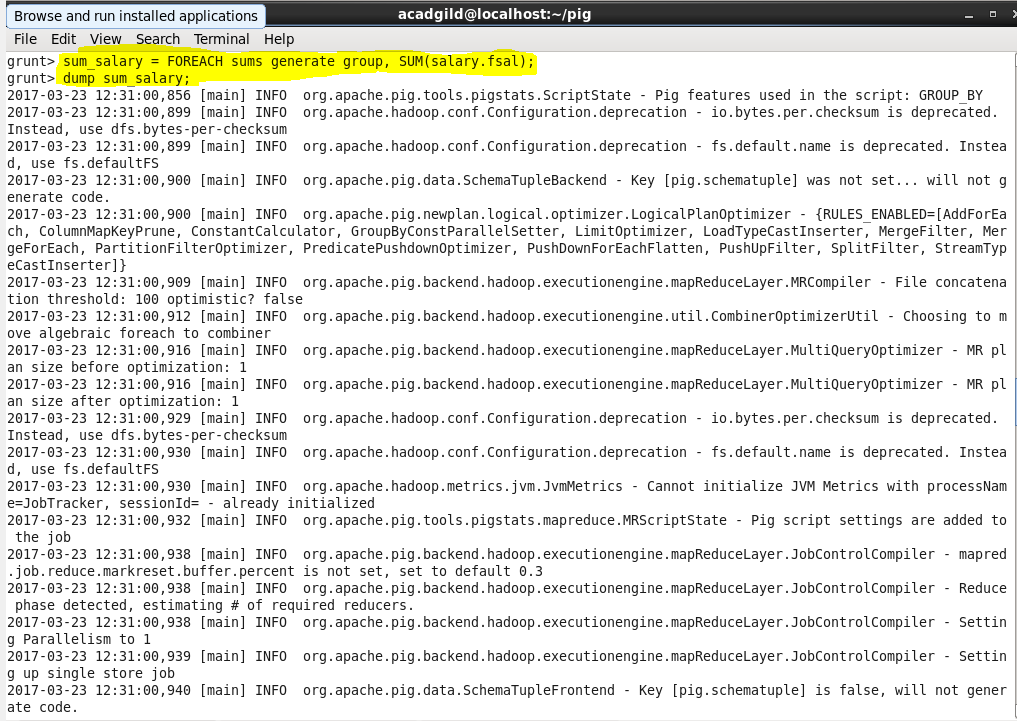


**Sum:**

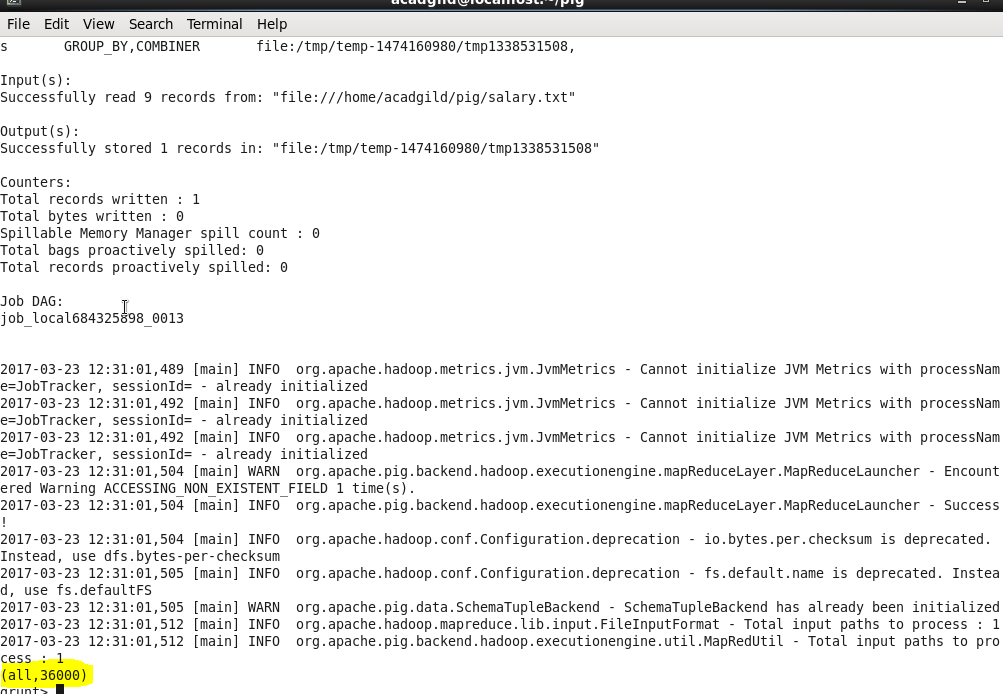
We can use the SUM() function of Pig Latin to get the total of the numeric values of a column in a single-column bag. While computing the total, the SUM() function ignores the NULL values.

**Syntax:**

SUM(expression);



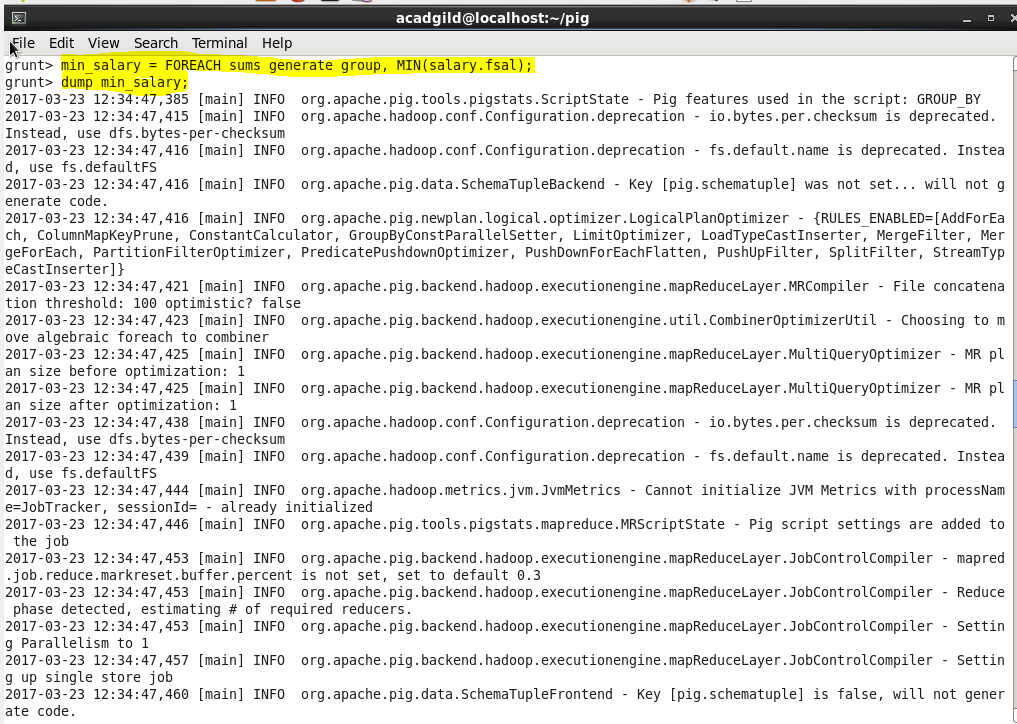
**Output:**



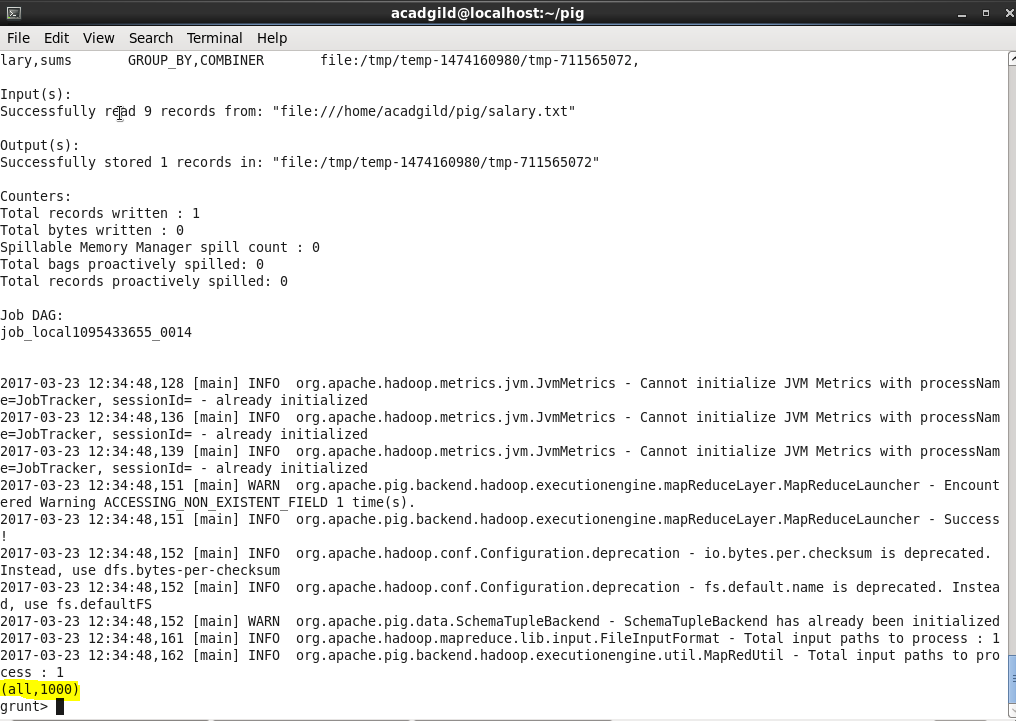
**Min:**

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. While calculating the minimum value, the MIN() function ignores the NULL values.

**Syntax:**

MIN(expression); 

**Output:**

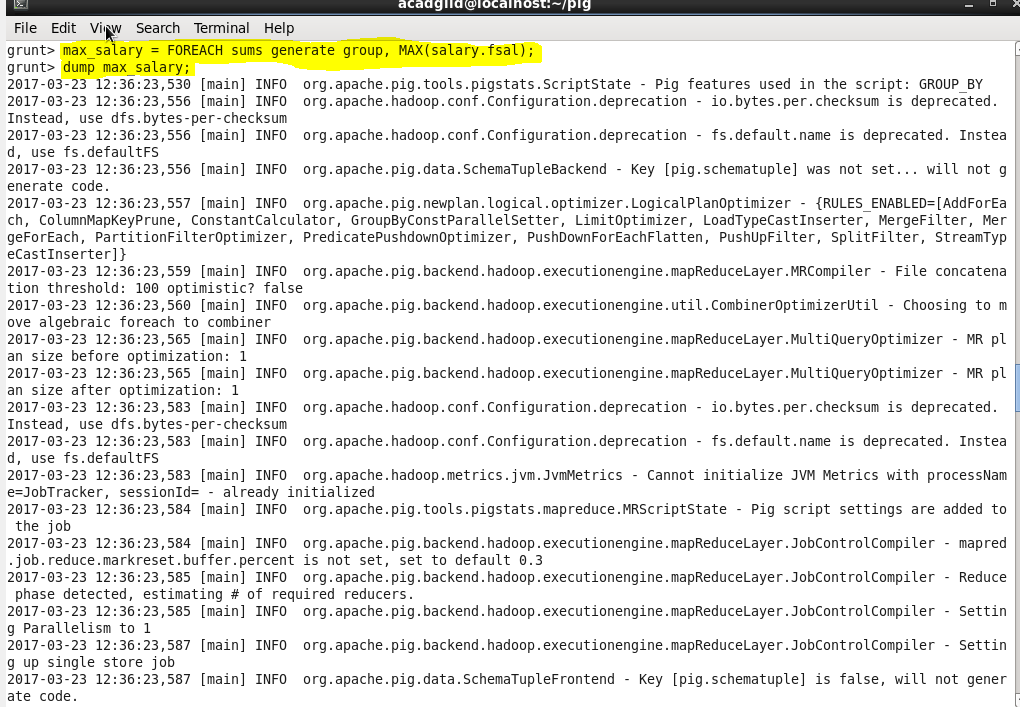


**Max:**

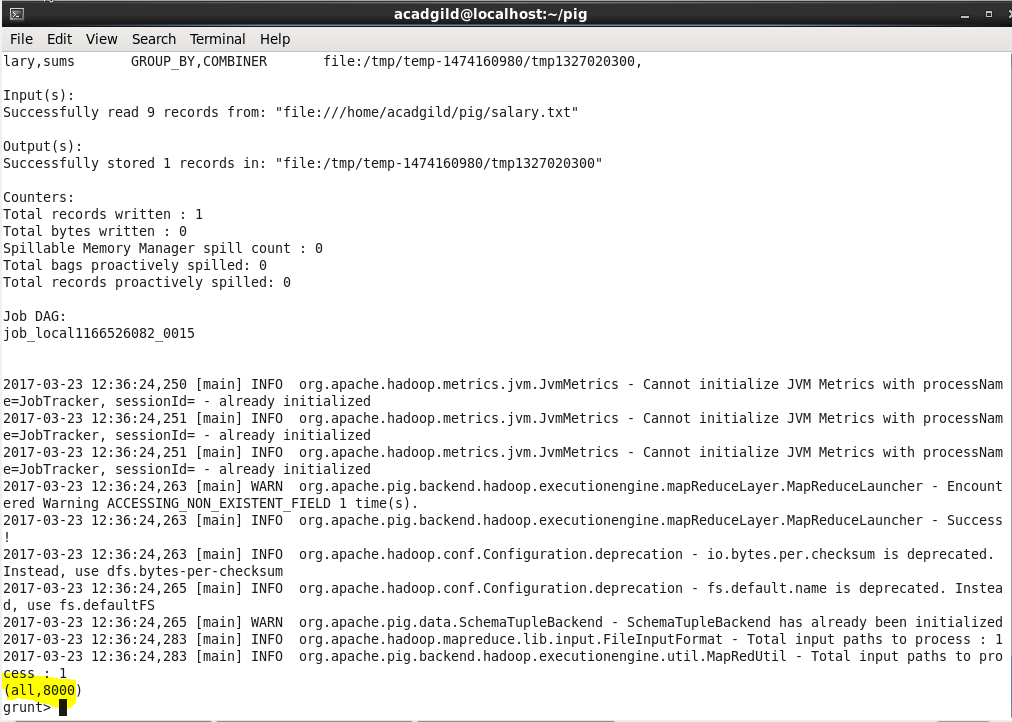
The Pig Latin MAX() function is used to calculate the highest value for a column (numeric values or chararrays) in a single-column bag. While calculating the maximum value, the Max() function ignores the NULL values.

Syntax:

MAX(expression);



Output:

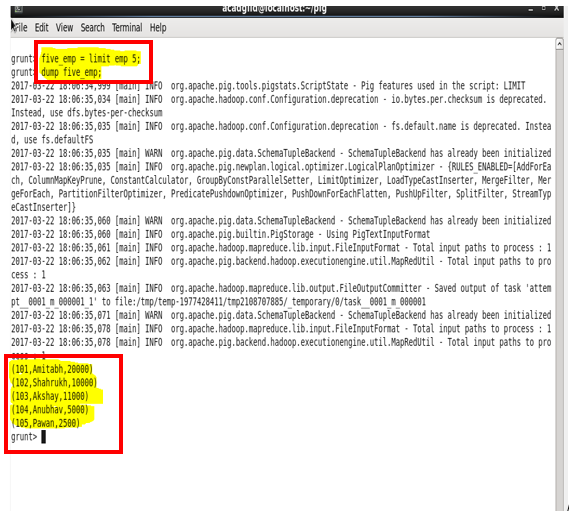


**Limit:**

Used to limit the number of outputs to the desired number.

**Syntax:**

Relation\_name2 = LIMIT Relation\_name1 n;

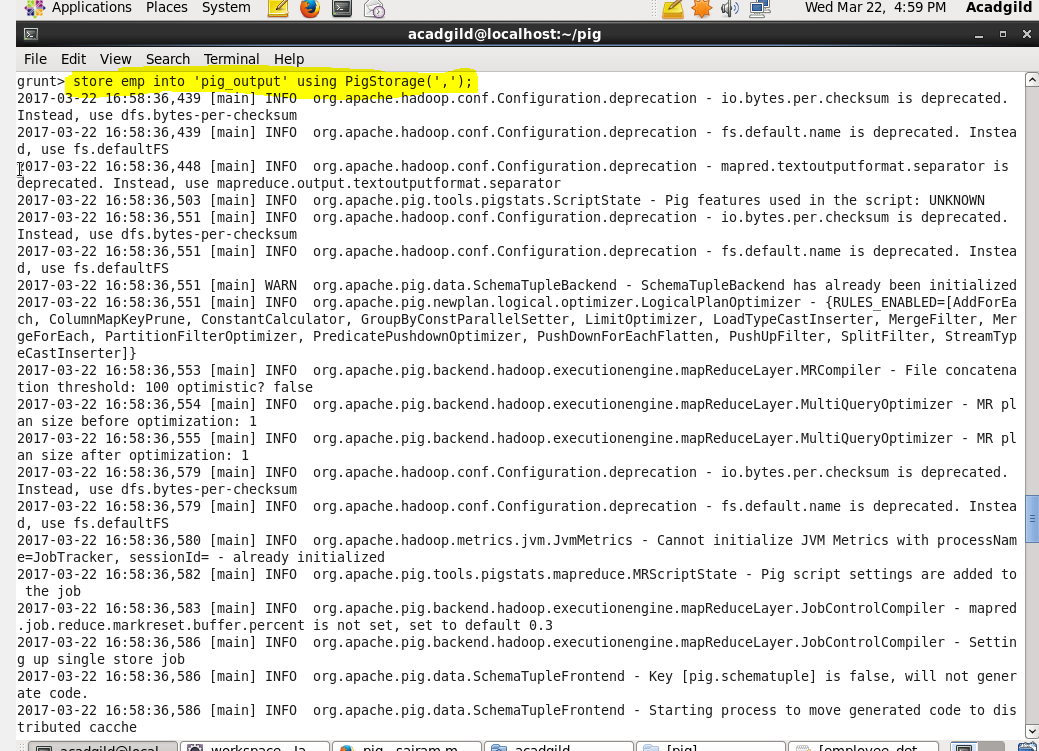


**STORE:**

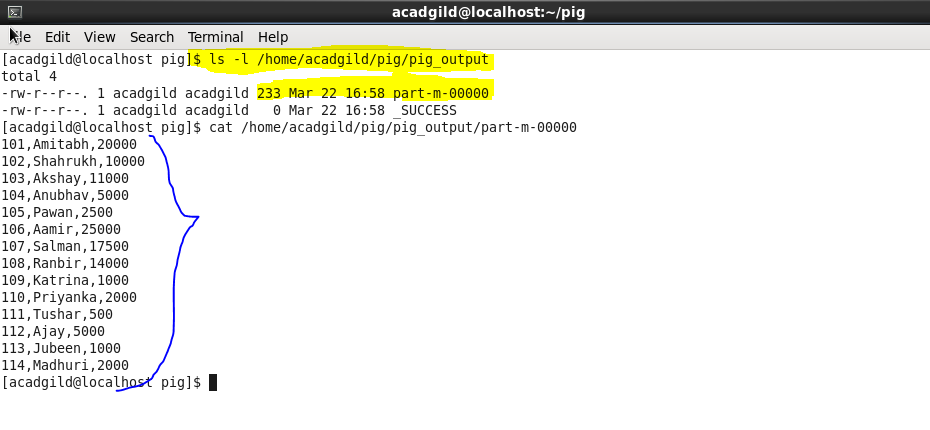
We can store the loaded data in the file system using the store operator.

**Syntax:**

STORE Relation\_name INTO ‘required\_directory\_path’ [USING Function];



**Output:**

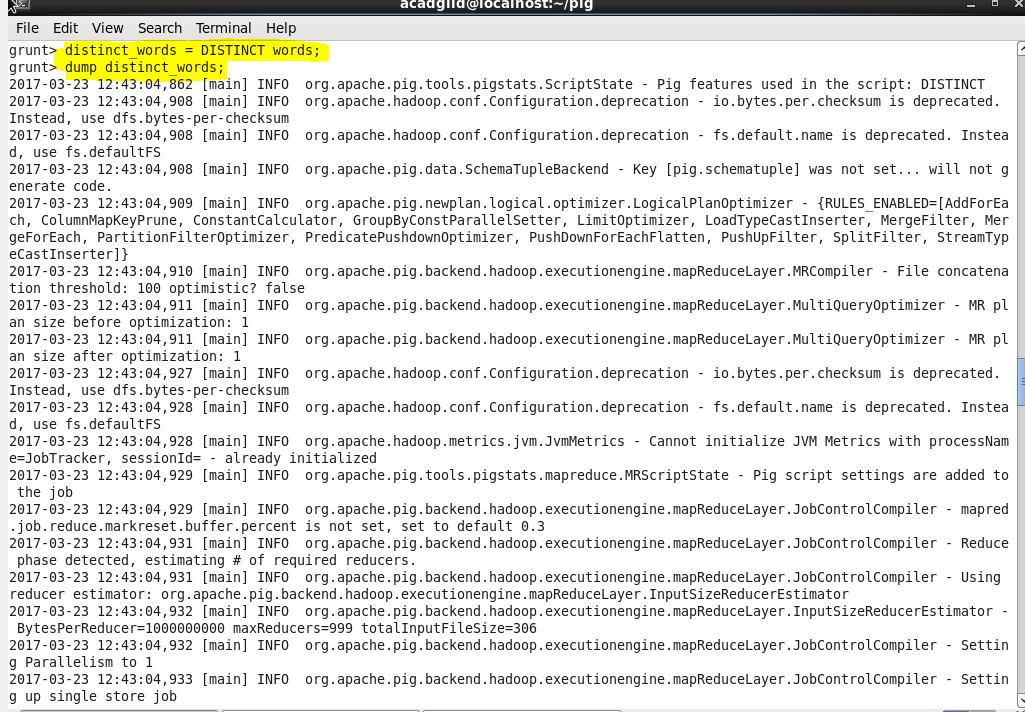


**Distinct:**

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

**Syntax:**

Relation\_name2 = DISTINCT Relation\_name1;

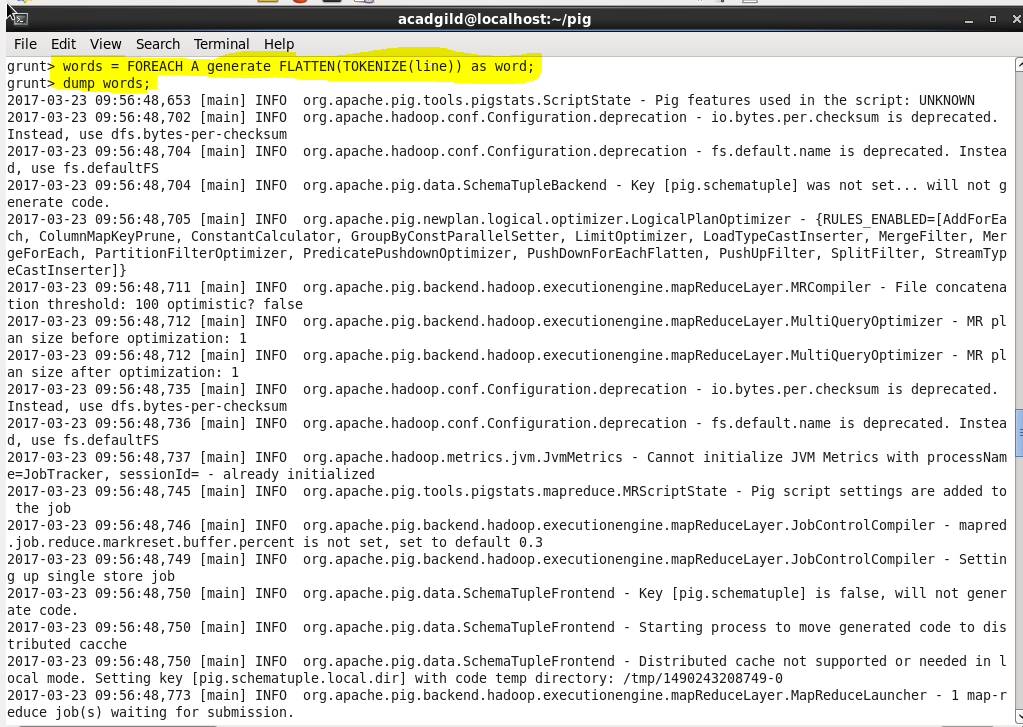


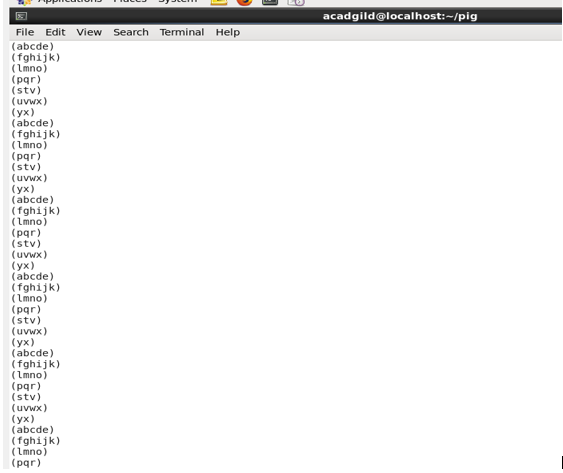
Output:

****

**Flatten:**

FLATTEN is an expression which will eliminate a level of nesting. Given a tuple which contains a bag, FLATTEN will emit several tuples each of which contains one record from the bag.



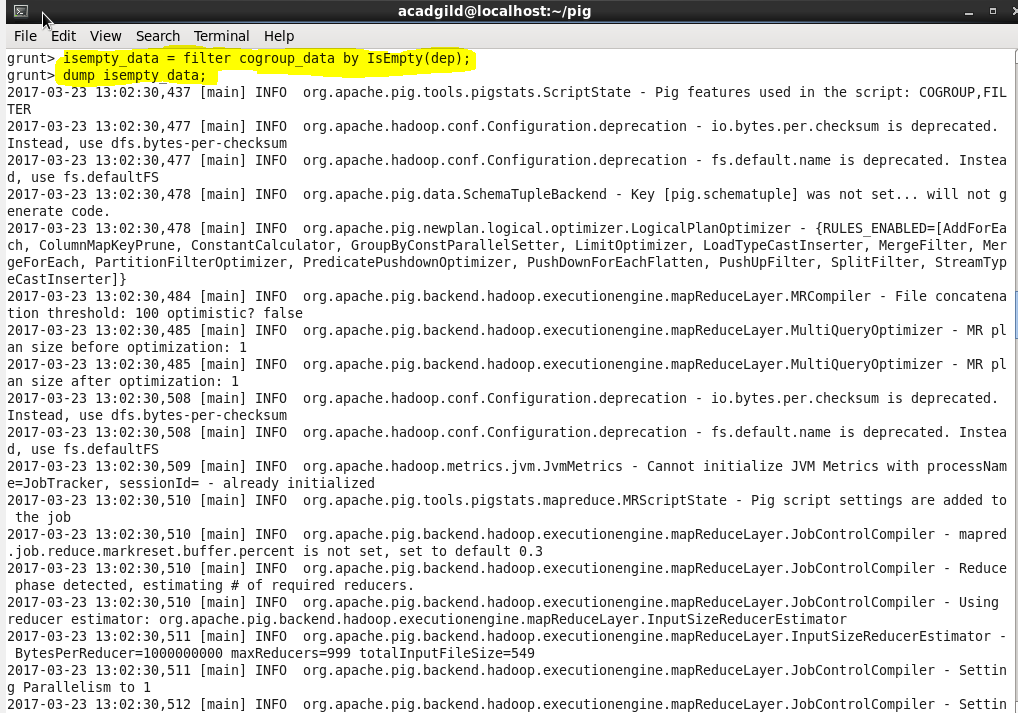


**IsEmpty:**

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

**Syntax:**

IsEmpty(expression);



**Output:**

