# 

# Pig is a high-level data flow platform for executing Map Reduce programs of Hadoop. It was developed by Yahoo. The language for Pig is pig Latin.

The Pig scripts get internally converted to Map Reduce jobs and get executed on data stored in HDFS.

# 

# Apache Pig Run Modes

# 

pig -x local

pig -x mapreduce

# 

# Pig Data Types

**Apache Pig supports many data types. A list of Apache Pig Data Types with description and examples are given below.**

| **Type** | **Description** | **Example** |
| --- | --- | --- |
| **Int** | **Signed 32 bit integer** | **2** |
| **Long** | **Signed 64 bit integer** | **15L or 15l** |
| **Float** | **32 bit floating point** | **2.5f or 2.5F** |
| **Double** | **32 bit floating point** | **1.5 or 1.5e2 or 1.5E2** |
| **charArray** | **Character array** | **hello javatpoint** |
| **byteArray** | **BLOB(Byte array)** |  |
| **tuple** | **Ordered set of fields** | **(12,43)** |
| **bag** | **Collection f tuples** | **{(12,43),(54,28)}** |
| **map** | **collection of tuples** | **[open#apache]** |

**hdfs dfs -mkdir /pig\_data**

**Load Operator**

LOAD 'info' [USING FUNCTION] [AS SCHEMA];

Here,

* **LOAD** is a relational operator.
* **'info'** is a file that is required to load. It contains any type of data.
* **USING** is a keyword.
* **FUNCTION** is a load function.
* **AS** is a keyword.
* **SCHEMA** is a schema of passing file, enclosed in parentheses.

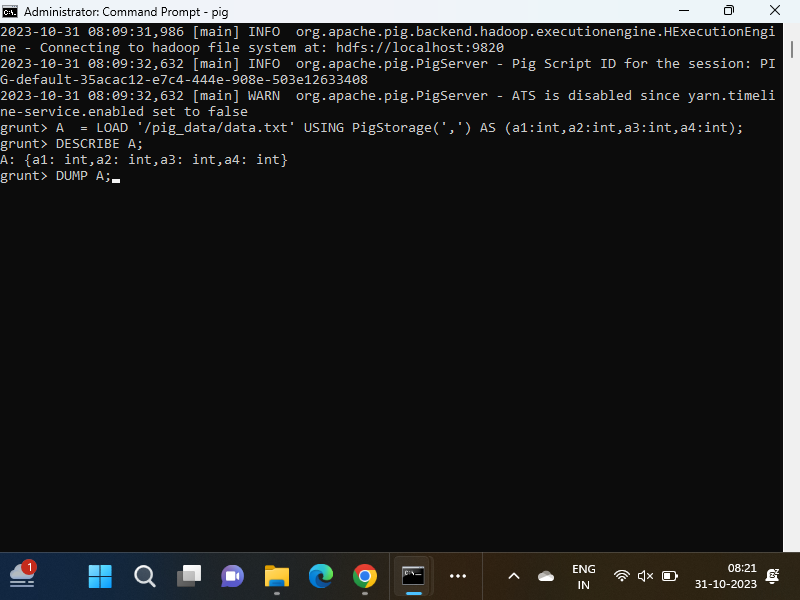
hdfs dfs -mkdir /pig\_data

hdfs dfs -put D:\data.txt /pig\_data

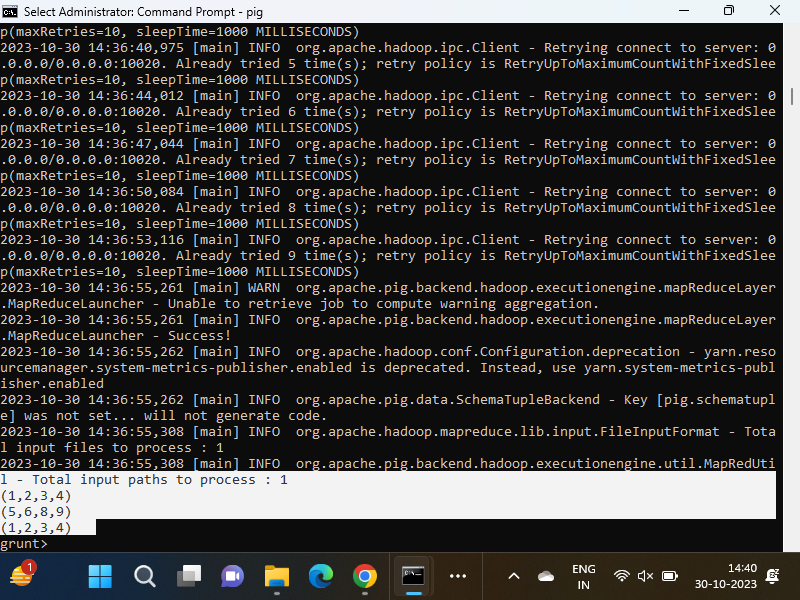
grunt**>** A = LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int);

grunt**>** DESCRIBE A;

grunt**>** DUMP A;



OUTPUT of DUMP command



The Apache Pig DISTINCT operator is used to remove duplicate tuples in a relation. Initially, Pig sorts the given data and then eliminates duplicates.

grunt**>** A =LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int) ;

Now, execute and verify the data.

grunt**>** DUMP A;

# Apache Pig DISTINCT Operator

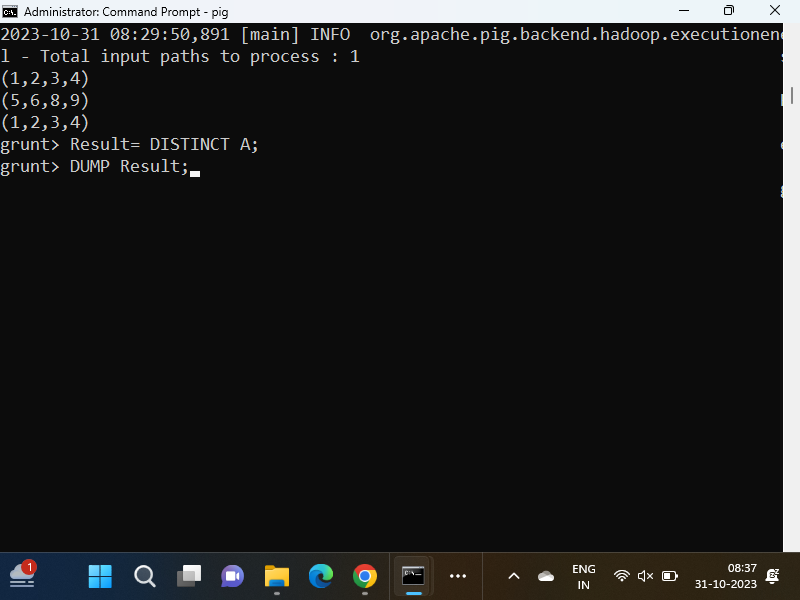
The Apache Pig DISTINCT operator is used to remove duplicate tuples in a relation. Initially, Pig sorts the given data and then eliminates duplicates.

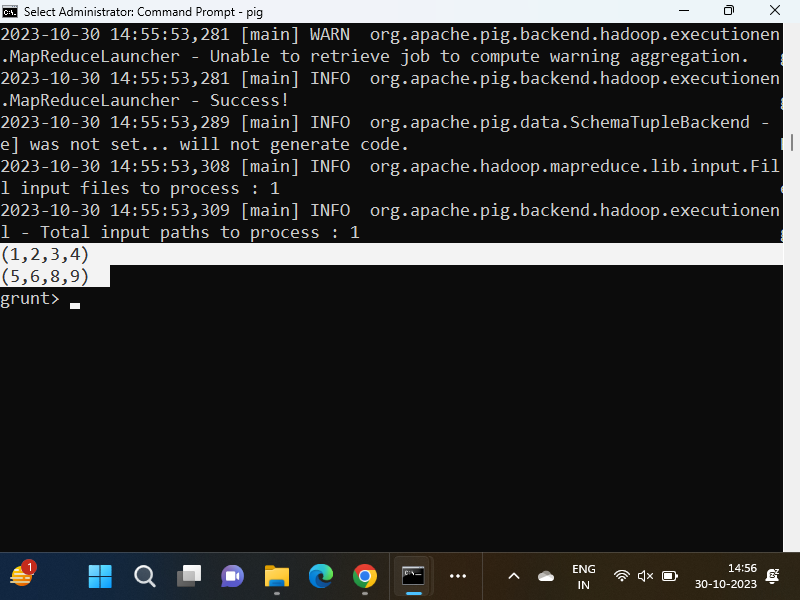
## Example of DISTINCT Operator

grunt**>** Result = DISTINCT A;

Now, execute and verify the data.

grunt**>** DUMP Result;





The Apache Pig FILTER operator is used to remove duplicate tuples in a relation. Initially, Pig sorts the given data and then eliminates duplicates.

## Example of FILTER Operator

In this example, we eliminate duplicate tuples.

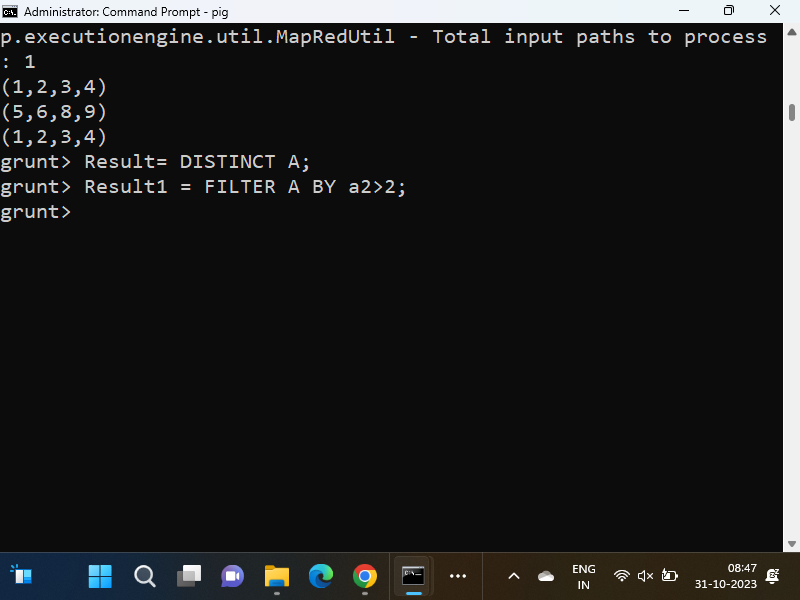
Load the file that contains the data.

grunt**>** A =LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int) ;

Now, execute and verify the data

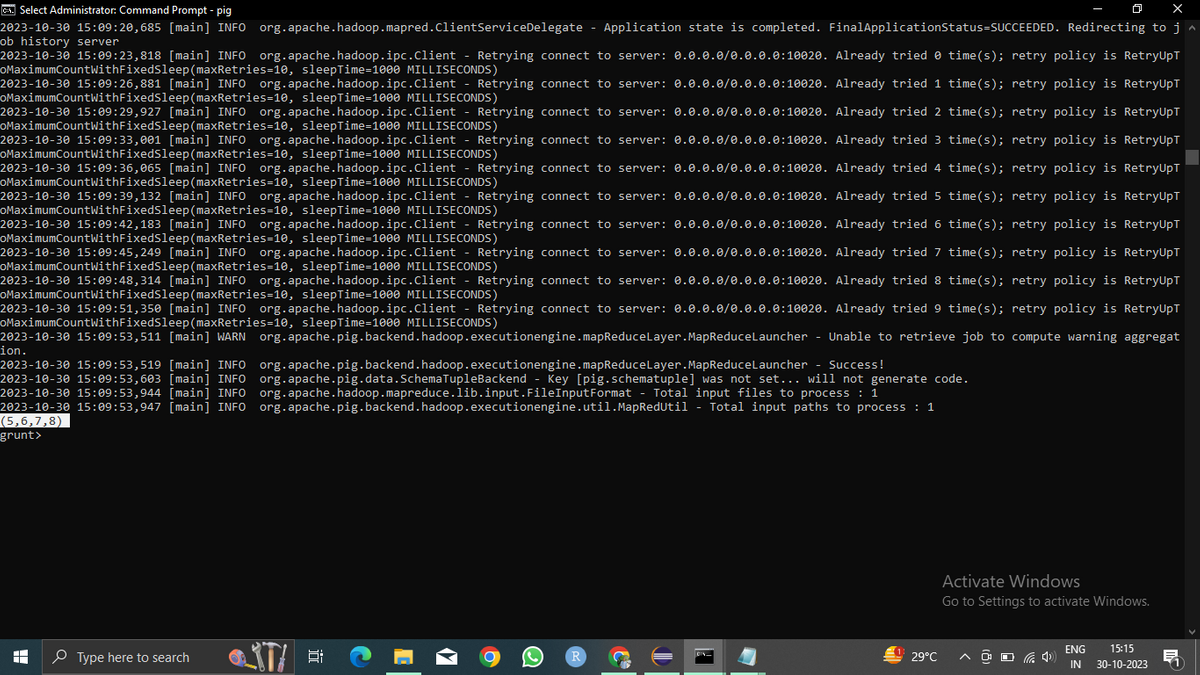
grunt**>** DUMP A;

grunt**>** Result1 = FILTER A BY a2>2;



Now, execute and verify the data.

grunt**>** DUMP Result1;



The Apache Pig ORDER BY operator sorts a relation based on one or more fields. It maintains the order of tuples.

## Example of ORDER BY Operator

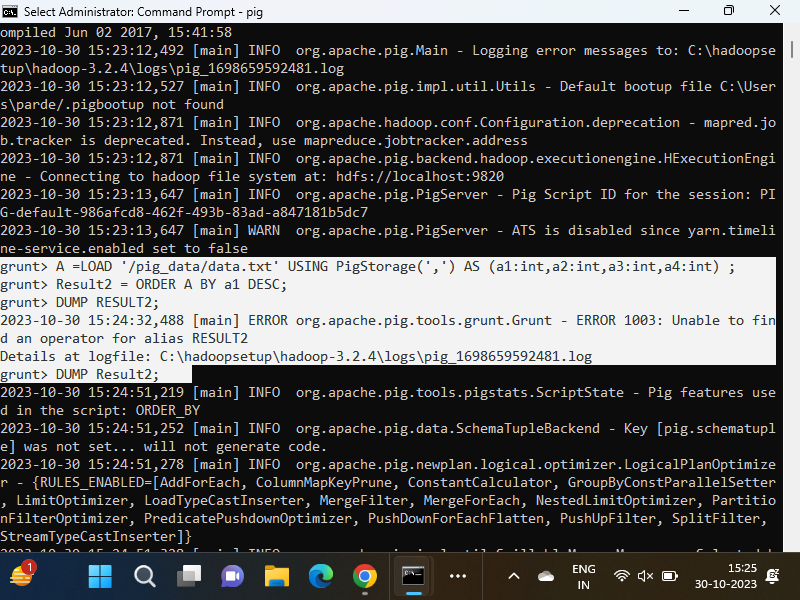
In this example, we return only two tuples from all the tuples in the relation.

Load the file that contains the data.

grunt**>** A =LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int) ;

Now, execute and verify the data.

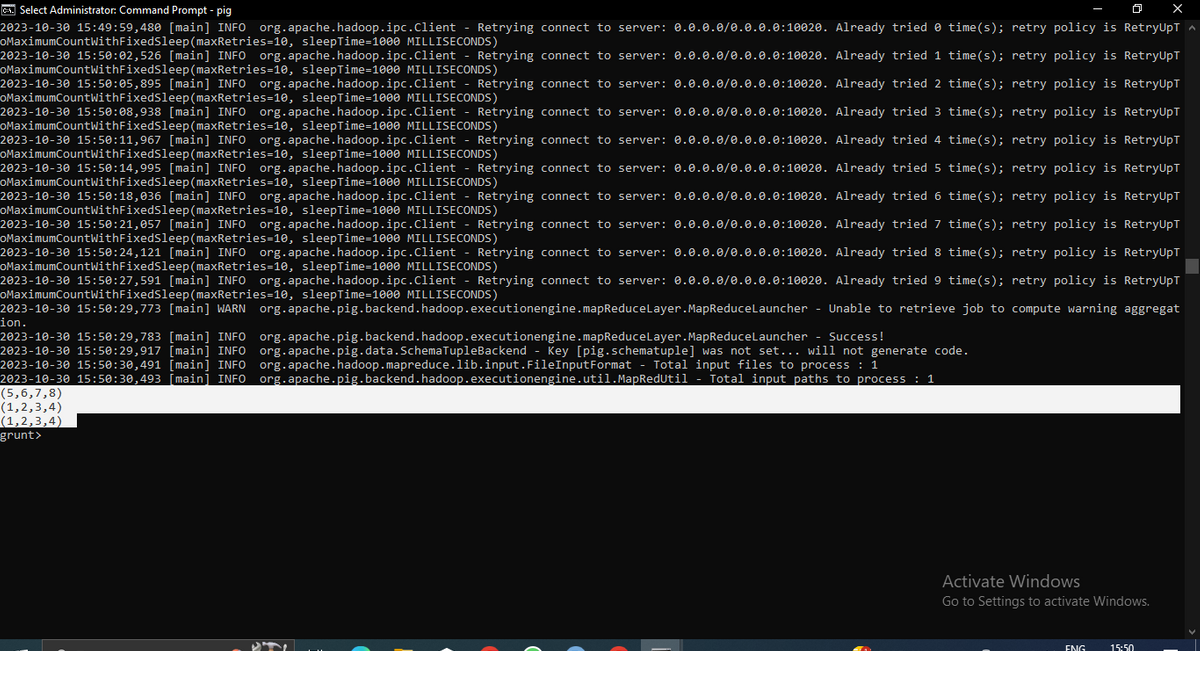
grunt**>** DUMP A;



Let's return the first two tuples.

grunt**>** Result2 = ORDER A BY a1 DESC;

grunt**>** DUMP Result2;



# Apache Pig UNION Operator

The Apache Pig UNION operator is used to compute the union of two or more relations. It doesn't maintain the order of tuples. It also doesn't eliminate the duplicate tuples.

## Example of UNION Operator

hdfs dfs -put D:\data2.txt /pig\_data

grunt**>** A =LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int) ;

Now, execute and verify the data.

grunt**>** DUMP A;

Load the another file that contains the data.

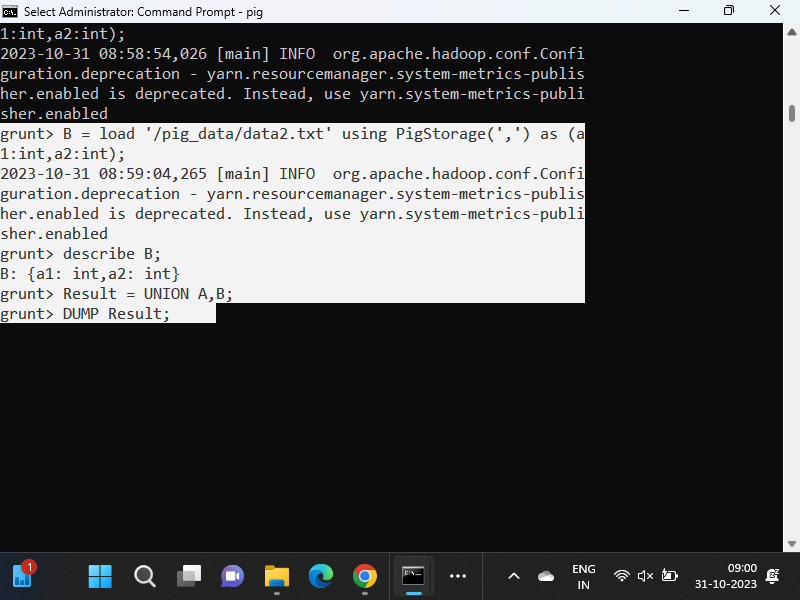
grunt**>** B = LOAD '/pig\_data/data2.txt' USING PigStorage(',') AS (a1:int,a2:int);

Now, execute and verify the data.

grunt**>** DUMP B;

Let's perform the union operation between both files.

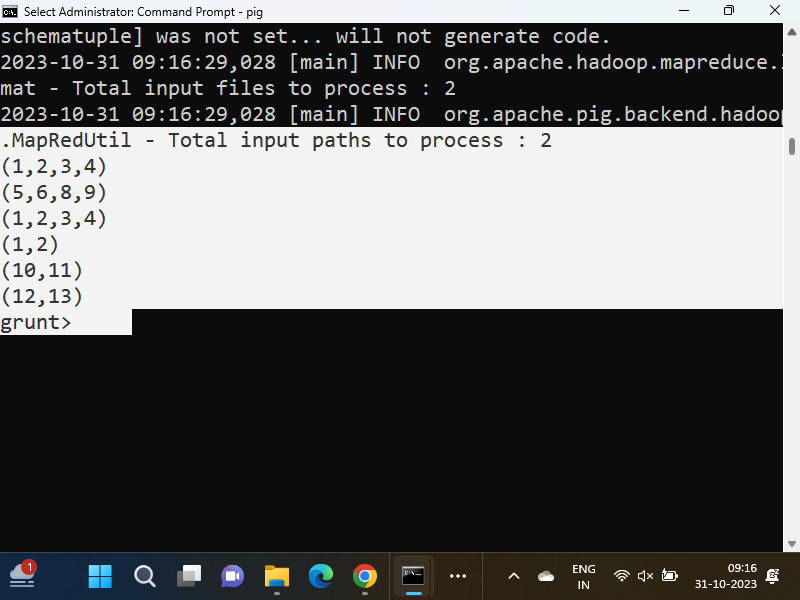
grunt**>** Result = UNION A,B;



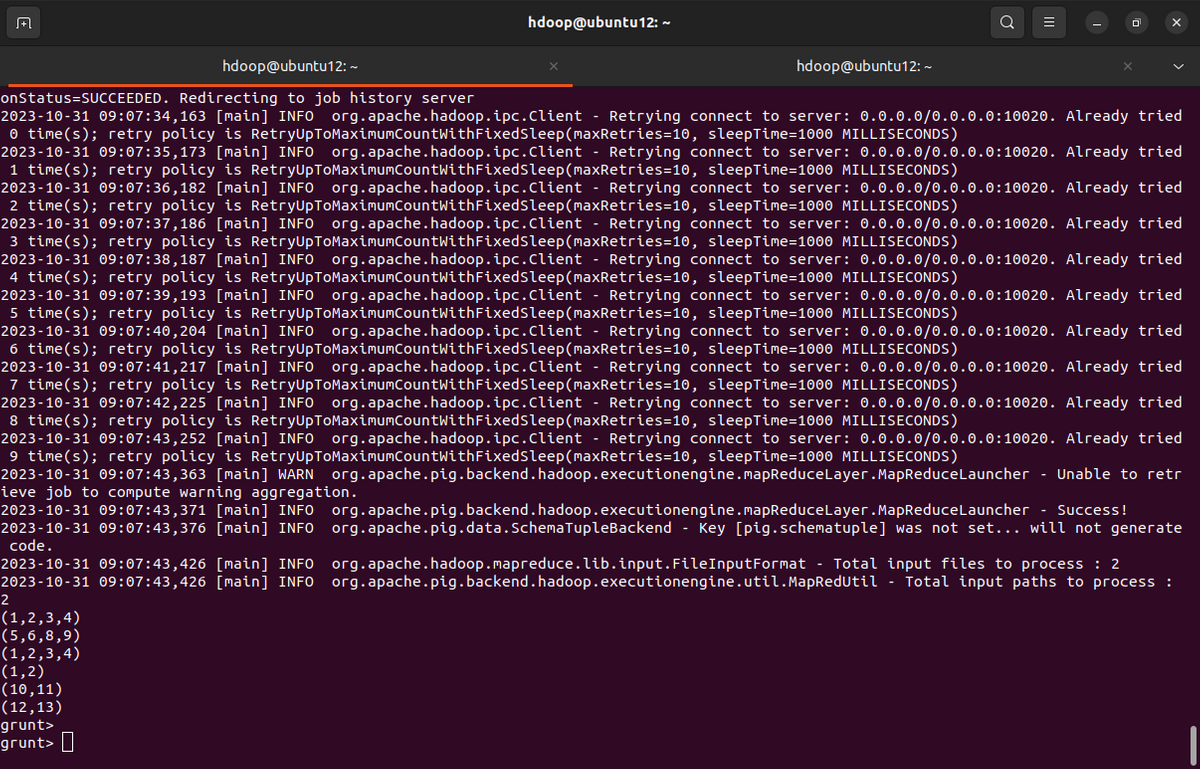
Now, execute and verify the data.

grunt**>** DUMP Result;

OUTPUT of the DUMP Result



LINUX-



Student table

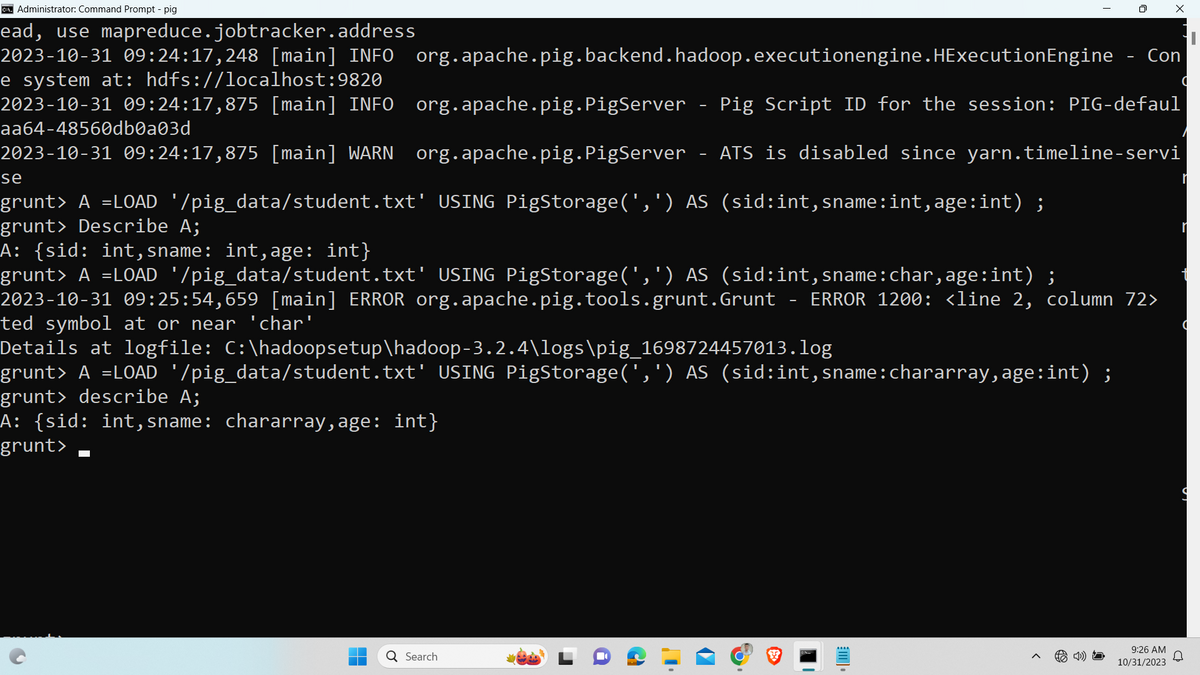
(HDFS Command)

hdfs dfs -mkdir /pig\_data

hdfs dfs -put D:\student.txt /pig\_data

(Pig Command)

C = LOAD '/pig\_data/student.txt' USING PigStorage(',') AS (sid:int,sname:chararray,age:int);



# Apache Pig FOREACH Operator

The Apache Pig FOREACH operator generates data transformations based on columns of data. It is recommended to use FILTER operation to work with tuples of data.

grunt**>** A =LOAD '/pig\_data/data.txt' USING PigStorage(',') AS (a1:int,a2:int,a3:int,a4:int) ;

Now, execute and verify the data.

grunt**>** DUMP A;

1. grunt**>** fe = FOREACH A generate a1,a2;
2. grunt**>** DUMP fe;

## Eval Functions

Given below is the list of **eval** functions provided by Apache Pig.

| **S.N.** | **Function & Description** |
| --- | --- |
| 1 | [**AVG()**](https://www.tutorialspoint.com/apache_pig/apache_pig_avg.htm)  To compute the average of the numerical values within a bag. |
| 2 | [**BagToString()**](https://www.tutorialspoint.com/apache_pig/apache_pig_bagtostring.htm)  To concatenate the elements of a bag into a string. While concatenating, we can place a delimiter between these values (optional). |
| 3 | [**CONCAT()**](https://www.tutorialspoint.com/apache_pig/apache_pig_concat.htm)  To concatenate two or more expressions of same type. |
| 4 | [**COUNT()**](https://www.tutorialspoint.com/apache_pig/apache_pig_count.htm)  To get the number of elements in a bag, while counting the number of tuples in a bag. |
| 5 | [**COUNT\_STAR()**](https://www.tutorialspoint.com/apache_pig/apache_pig_count_star.htm)  It is similar to the **COUNT()** function. It is used to get the number of elements in a bag. |
| 6 | [**DIFF()**](https://www.tutorialspoint.com/apache_pig/apache_pig_diff.htm)  To compare two bags (fields) in a tuple. |
| 7 | [**IsEmpty()**](https://www.tutorialspoint.com/apache_pig/apache_pig_isempty.htm)  To check if a bag or map is empty. |
| 8 | [**MAX()**](https://www.tutorialspoint.com/apache_pig/apache_pig_max.htm)  To calculate the highest value for a column (numeric values or chararrays) in a single-column bag. |
| 9 | [**MIN()**](https://www.tutorialspoint.com/apache_pig/apache_pig_min.htm)  To get the minimum (lowest) value (numeric or chararray) for a certain column in a single-column bag. |
| 10 | [**PluckTuple()**](https://www.tutorialspoint.com/apache_pig/apache_pig_plucktuple.htm)  Using the Pig Latin **PluckTuple()** function, we can define a string Prefix and filter the columns in a relation that begin with the given prefix. |
| 11 | [**SIZE()**](https://www.tutorialspoint.com/apache_pig/apache_pig_size.htm)  To compute the number of elements based on any Pig data type. |
| 12 | [**SUBTRACT()**](https://www.tutorialspoint.com/apache_pig/apache_pig_subtract.htm)  To subtract two bags. It takes two bags as inputs and returns a bag which contains the tuples of the first bag that are not in the second bag. |
| 13 | [**SUM()**](https://www.tutorialspoint.com/apache_pig/apache_pig_sum.htm)  To get the total of the numeric values of a column in a single-column bag. |
| 14 | [**TOKENIZE()**](https://www.tutorialspoint.com/apache_pig/apache_pig_tokenize.htm)  To split a string (which contains a group of words) in a single tuple and return a bag which contains the output of the split operation. |

Result = FOREACH A GENERATE CONCAT (a1,'\_',a2);

DUMP Result;

MAX(exp)