**Practical 7**

**Aim: Using Pig Tool and creating Pig Latin Scripts**

**What is Pig in Hadoop?**

Pig is a scripting platform that runs on Hadoop clusters designed to process and analyze large datasets. Pig is extensible, self-optimizing, and easily programmed.

Programmers can use Pig to write data transformations without knowing Java. Pig uses both structured and unstructured data as input to perform analytics and uses HDFS to store the results.

**Components of Pig**

There are two major components of the Pig:

● Pig Latin script language

● A runtime engine

**Pig Latin script language:**

The Pig Latin script is a procedural data flow language. It contains syntax and commands that can be applied to implement business logic. Examples of Pig Latin are LOAD and STORE.

**A runtime engine:**

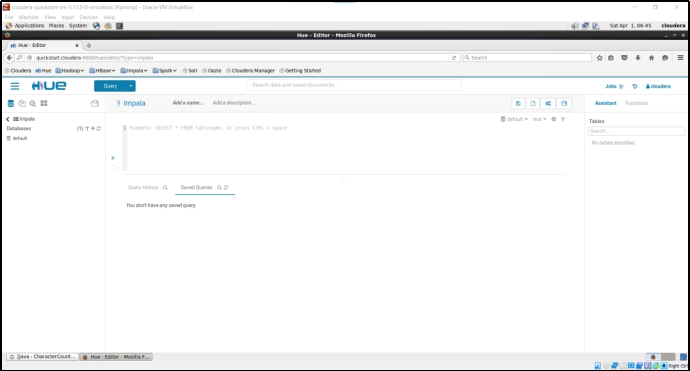
The runtime engine is a compiler that produces sequences of MapReduce programs. It uses HDFS to store and retrieve data. It is also used to interact with the Hadoop system (HDFS and MapReduce).

The runtime engine parses, validates, and compiles the script operations into a sequence of MapReduce jobs.

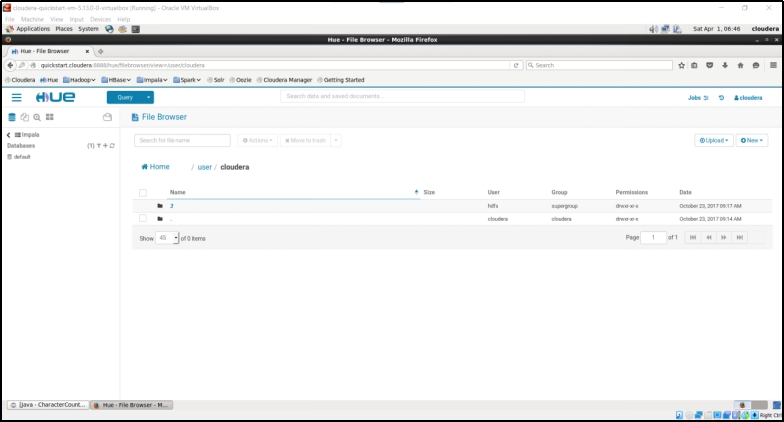
**Steps:**

1. Open cloudera browser
2. Locate Hue and get logged in with username cloudera and password cloudera.

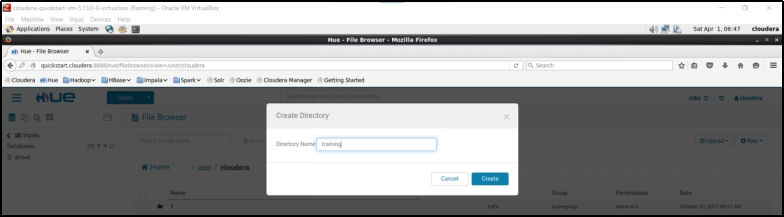
Open browser > Hue > Enter username & password



1. Locate file Browser and check for directory user/cloudera.



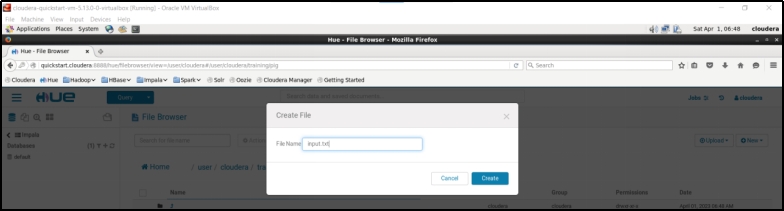
1. Create a new directory named ‘training’.



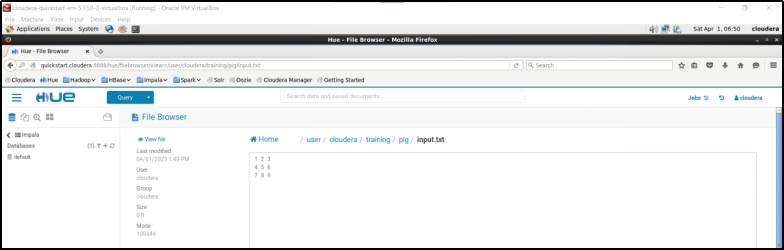
1. Create the pig directory inside the training directory.



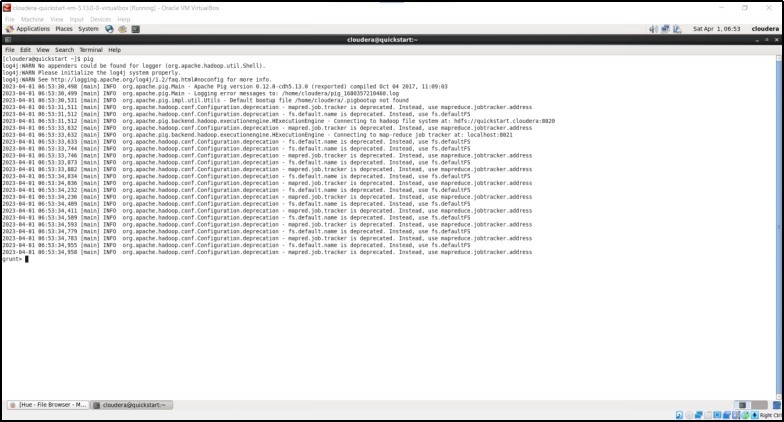
1. Create an input.txt file in the pig directory.



1. Add some text into the input.txt and save the file.

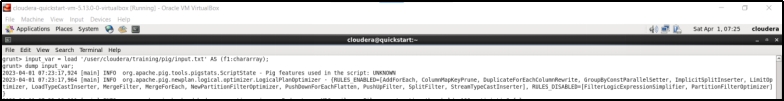


1. Open terminal and start pig tool by executing pig command.



1. Load input.txt file in input variable.

input = LOAD ‘/user/cloudera/training/pig/input.txt’ AS (f1:chararray);



1. Display the contents of the input variable.

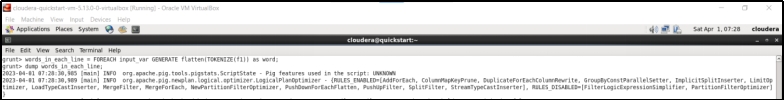
DUMP input;



1. Tokenize the text that is stored in variable input.

words\_in\_each\_line = FOREACH input\_var GENERATE flatten(TOKENIZE(f1)) as word;

dump words\_in\_each\_line;

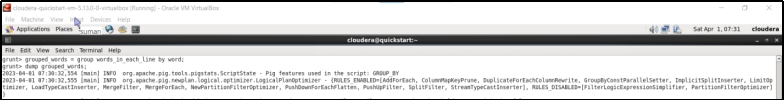




1. Group all similar words.

grouped\_words = GROUP words\_in\_each\_line by word;

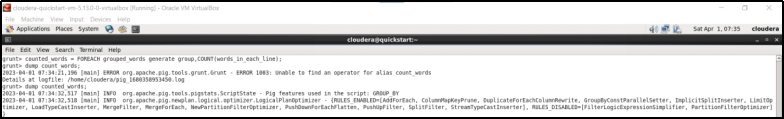
dump grouped\_words;





1. Count the number of occurrences of each word.

counted\_words = FOREACH grouped\_words generate group, COUNT(words\_in\_each\_line);



dump counted\_words;



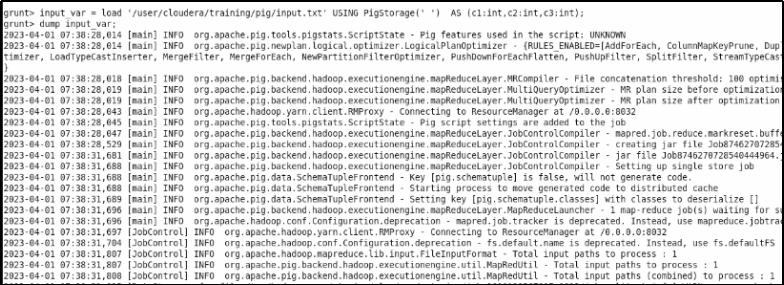
**Loading matrix in a pig variable.**

1. **Create input.txt from Hue**



1. **input\_var = LOAD ‘/user/cloudera/Training/pig/mydata.txt’ USING PigStorage(‘ ’) AS (c1:int,c2:int,c3:int);**

**dump input\_var;**





**Pig script for word count**

1. Create pig file and write the commands

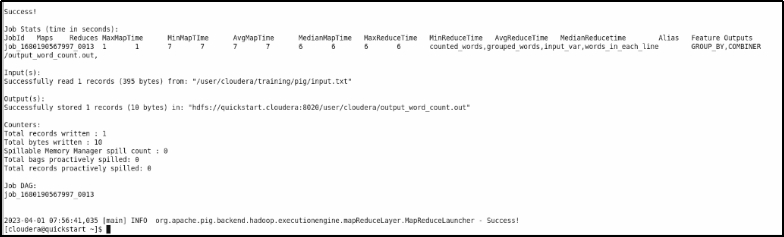
gedit wordcount.pig



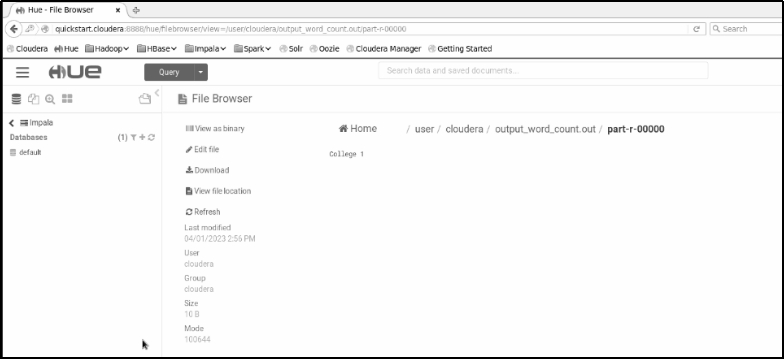
1. Execute the pig file

pig -x mapreduce word\_count.pig;





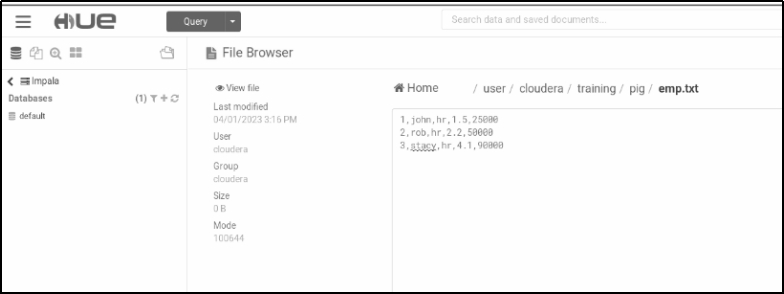
1. Check the output in the Hue output folder



**Pig script for employee data**

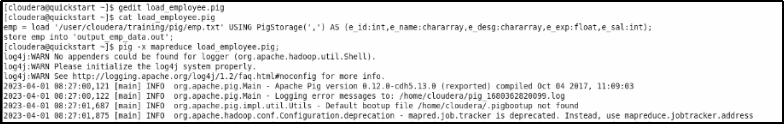
1. Create emp.txt with Hue

empID, empName,designation,experience,salary



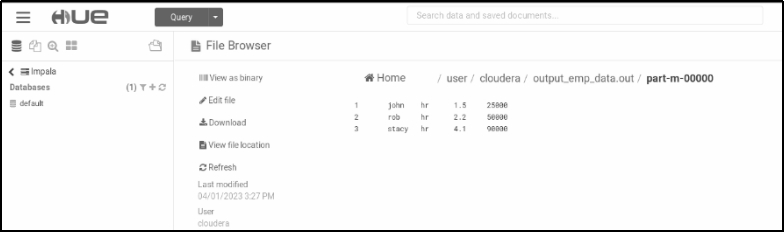
1. Load employees data into emp variable. Also specify the proper schema for the same.

Send or store emp data in empout.out file.





1. Check the contents of the empout.out file.



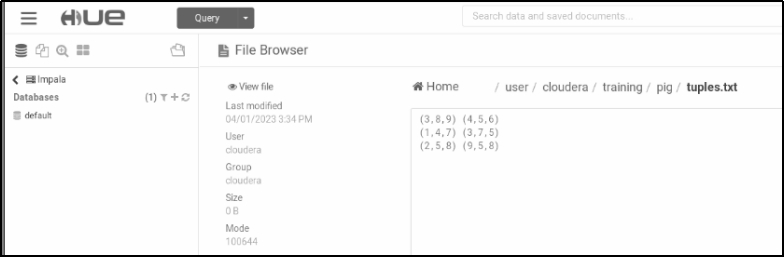
**Pig script for tuples**

1. Create a relation named ‘tupledata’ and enter the following data.

(3,8,9) (4,5,6)

(1,4,7) (3,7,5)

(2,5,8) (9,5,8)



1. Load tupledata into a pig variable named ‘inputtuple’.

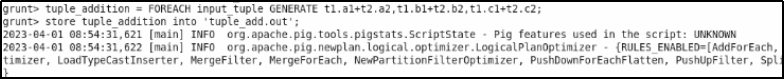
input\_tuple = load '/user/cloudera/training/pig/tuples.txt' USING PigStorage(' ') AS (t1:tuple(a1:int,b1:int,c1:int),t2:tuple(a2:int,b2:int,c2:int));

Dump input\_tuple;





1. Add the columns with a similar index of each tuple and store the result in addout.out file.



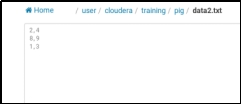
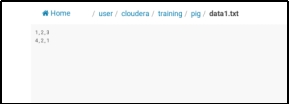


1. View the results

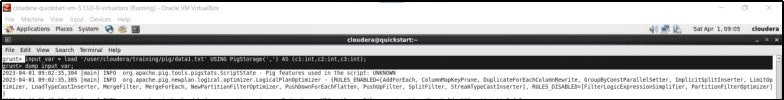


**Filtering using Pig script**

1. Create data1.txt and data2.txt in Hue

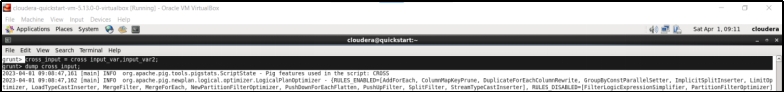
 

1. Load and dump the data1.txt and data2.txt in pig

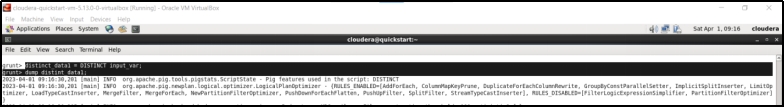


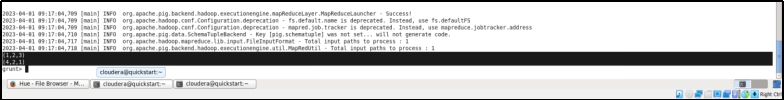


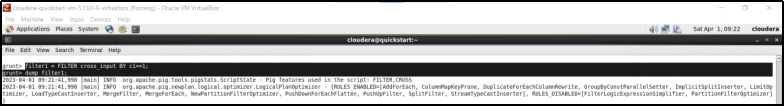
1. Cross the data1 and data2 and dump the output



1. Using DISTINCT and FILTER

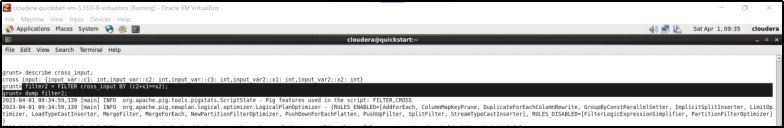


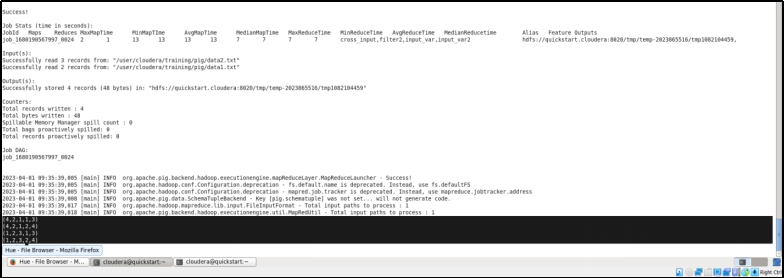






1. Filter all rows for 2nd column val+4th column val=5th column val



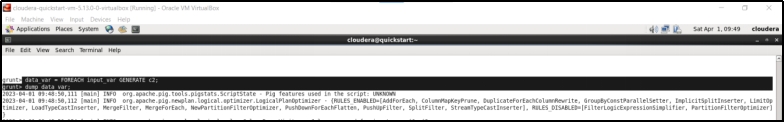


1. Filter all rows for 2nd column val+4th column val=5th column val and also check for 1st column val as



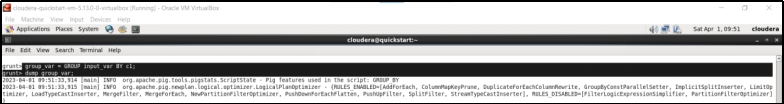


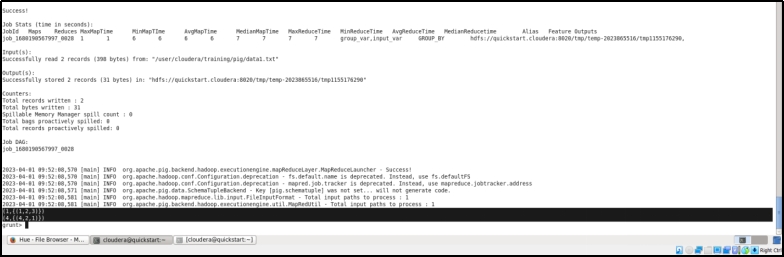
1. Dump value of data1



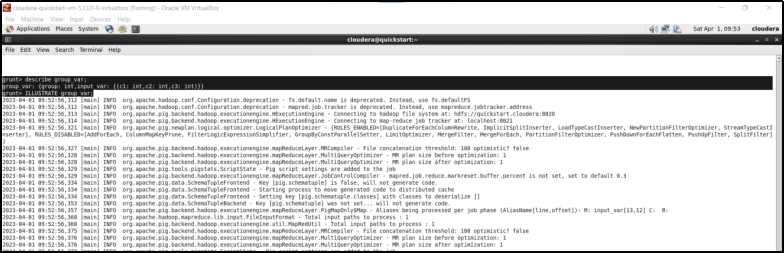


1. group\_var = group by 1st value

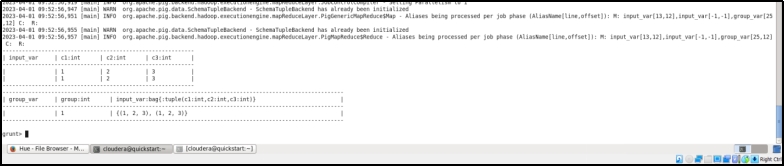




1. Describe group\_var



1. ILLustrate group\_var



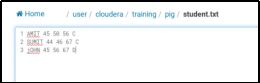
**Pig Latin Student**

1. CREATE FILE TO STORE THE FOLLOWING DATA in stud.txt

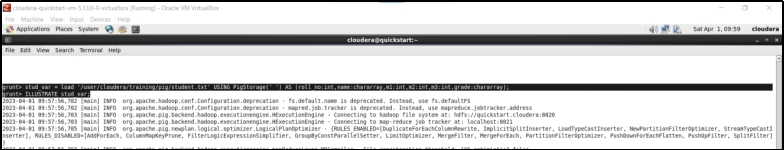
1 AMIT 45 50 56 C

2 SUMIT 44 46 67 C

3 jOHN 45 56 67 D



1. Load Students data into a pig relation named ‘studInfo’.



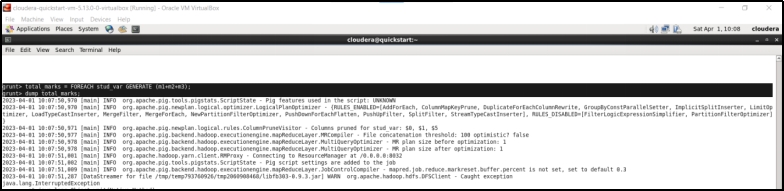


1. Display the list of students rollno and name.

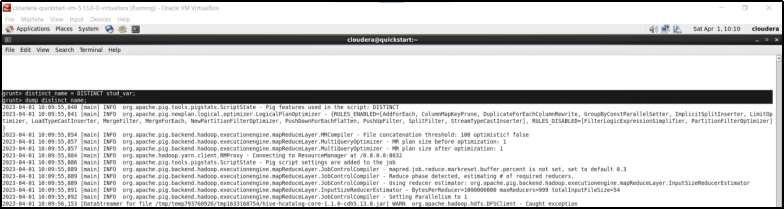




1. Display the total marks of each student.Display distinct names of students

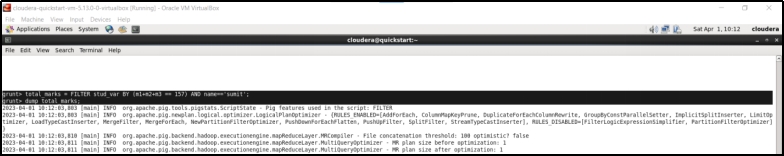








1. Check if the total marks of the sumit is 157 or not.



1. Display distinct marks of subject1





**Pig Latin Nested Projection**

1. Create the file named ‘data’ for the following data.

1,2,3

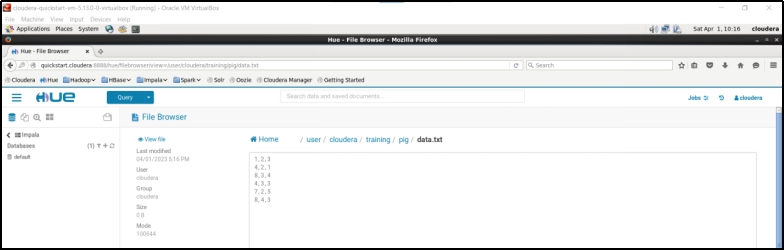
4,2,1

8,3,4

4,3,3

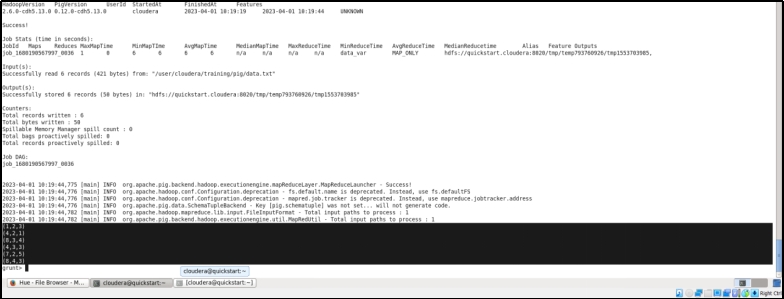
7,2,5

8,4,3

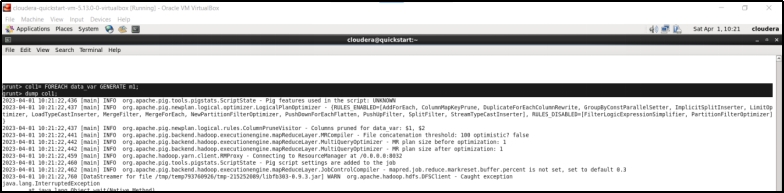


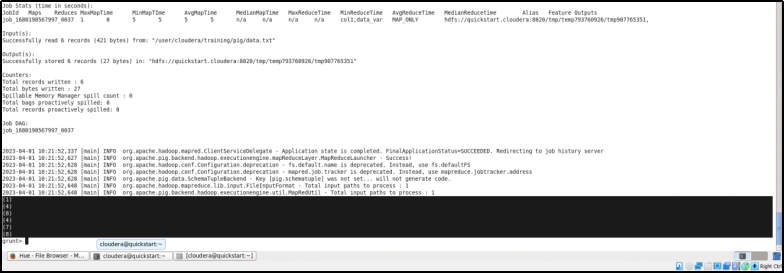
1. Load the contents of ‘data’ into relation data\_var.



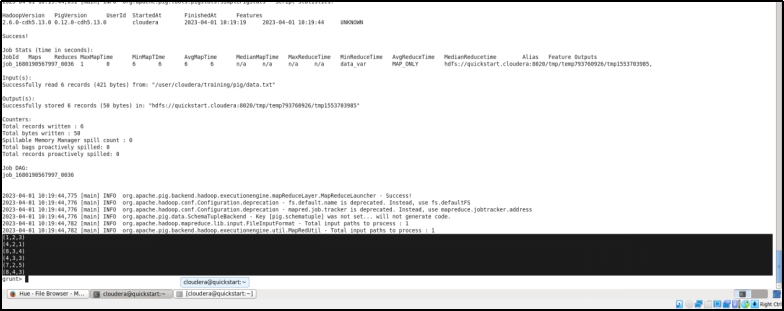


1. Display the elements of the first column.





1. Display all tuples of relation data\_var.



1. Display the first two columns of relation data\_var.





1. Create a file named ‘data1’ to store the following matrix.

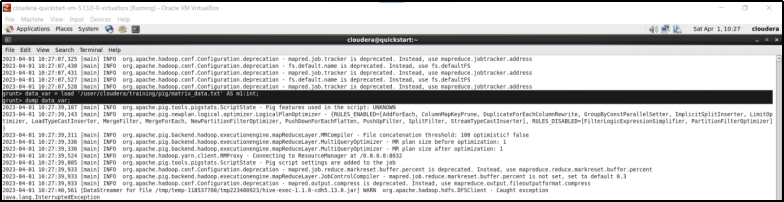
1

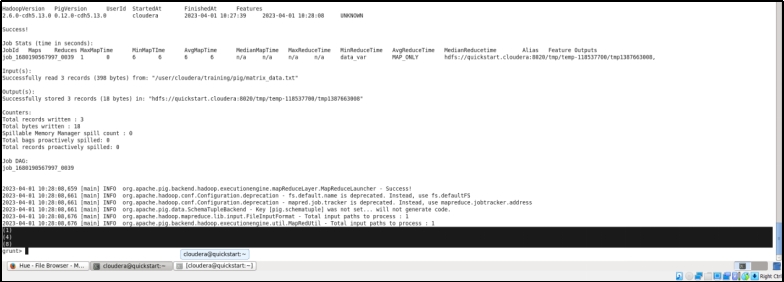
4

8

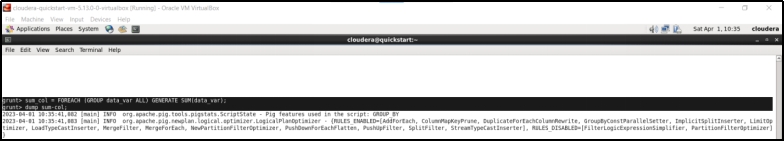


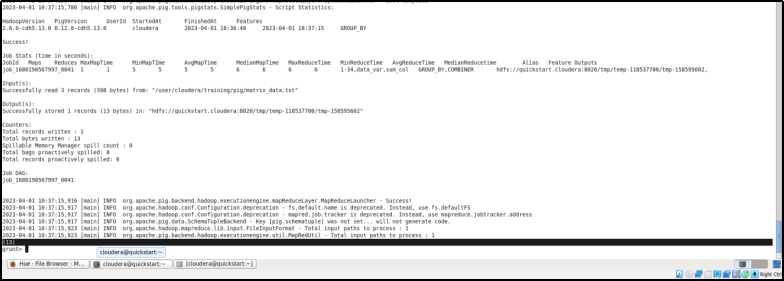
1. Load data of ‘data1’ into relation data\_var.





1. Sum column a1





1. Nested projection

