**Assignment 10.2**

**Problem Statement:**

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

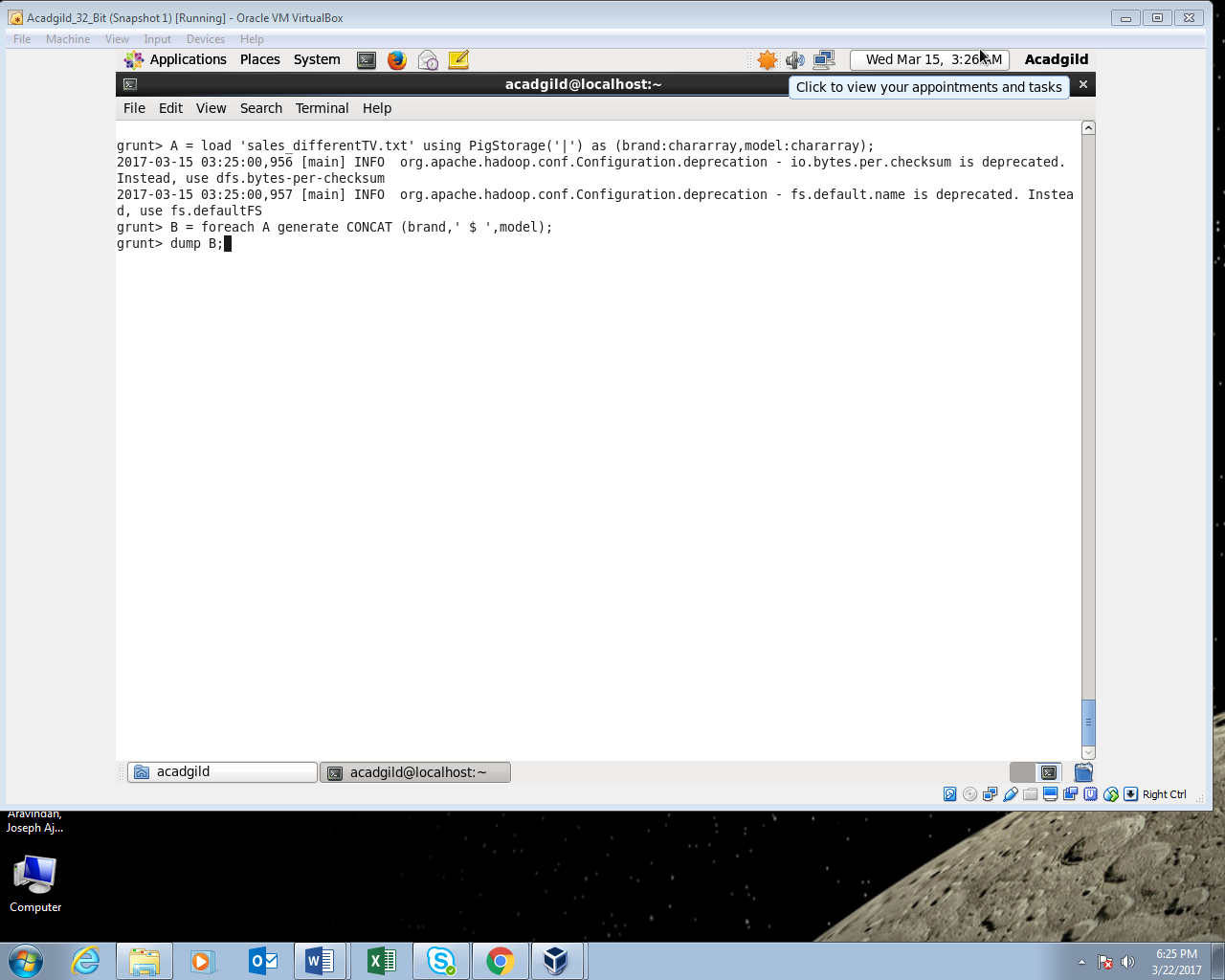
* **Concat:**

Concatenates two fields of type chararray or two fields of type bytearray.

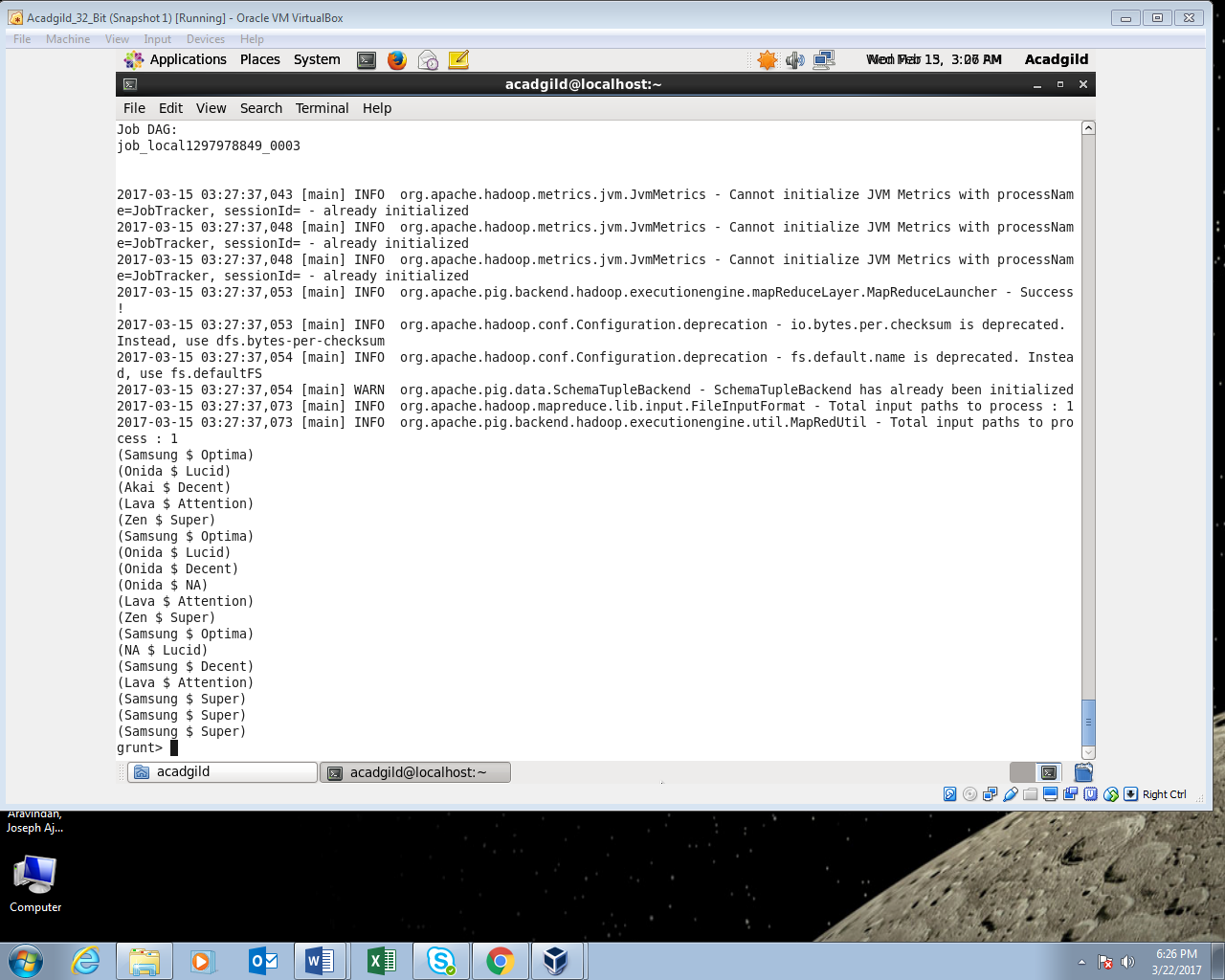
Syntax: CONCAT (expression, expression)

**For Example:**

**Input:**



**Output:**



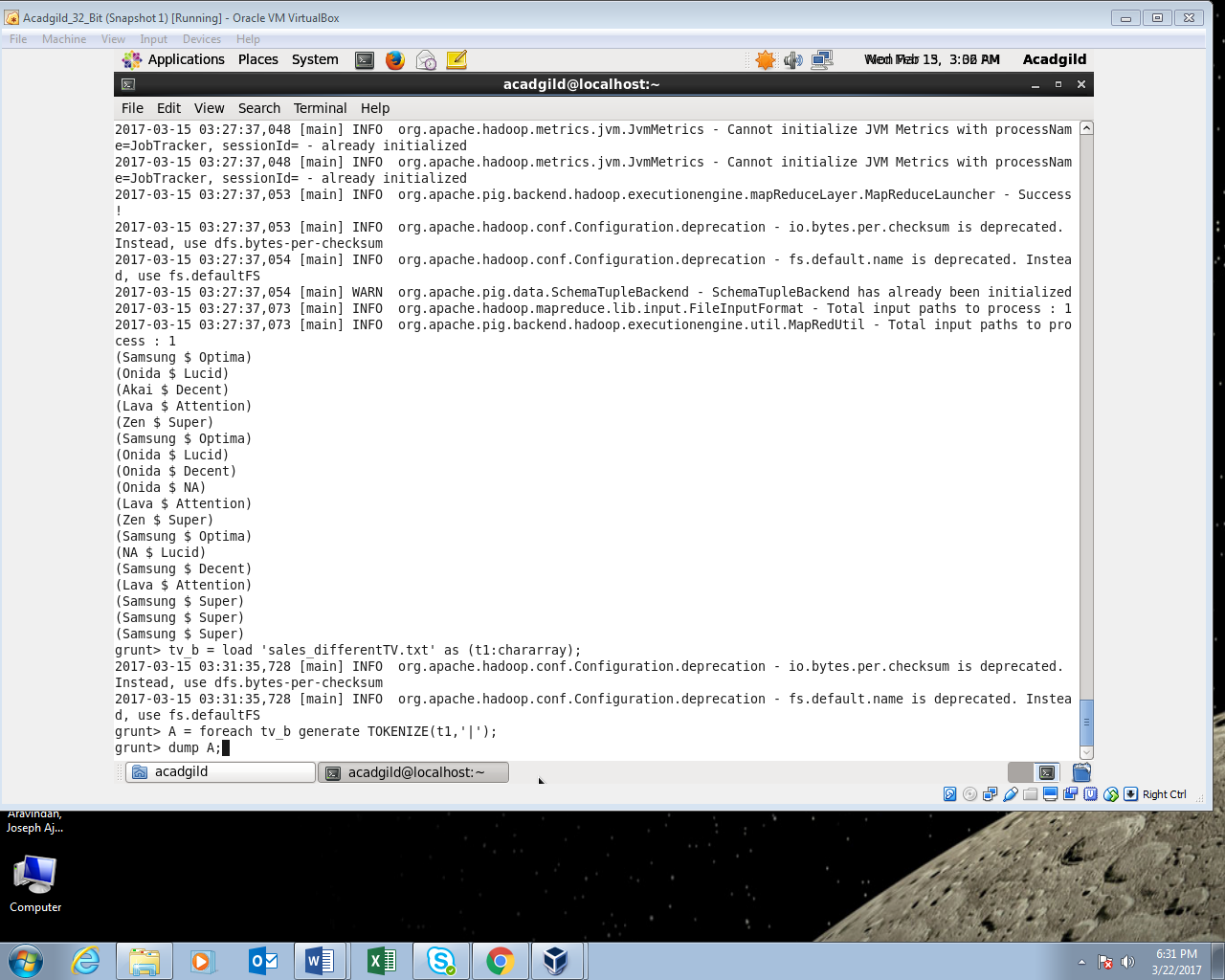
* **Tokenize:**

Splits a string and outputs a bag of words.

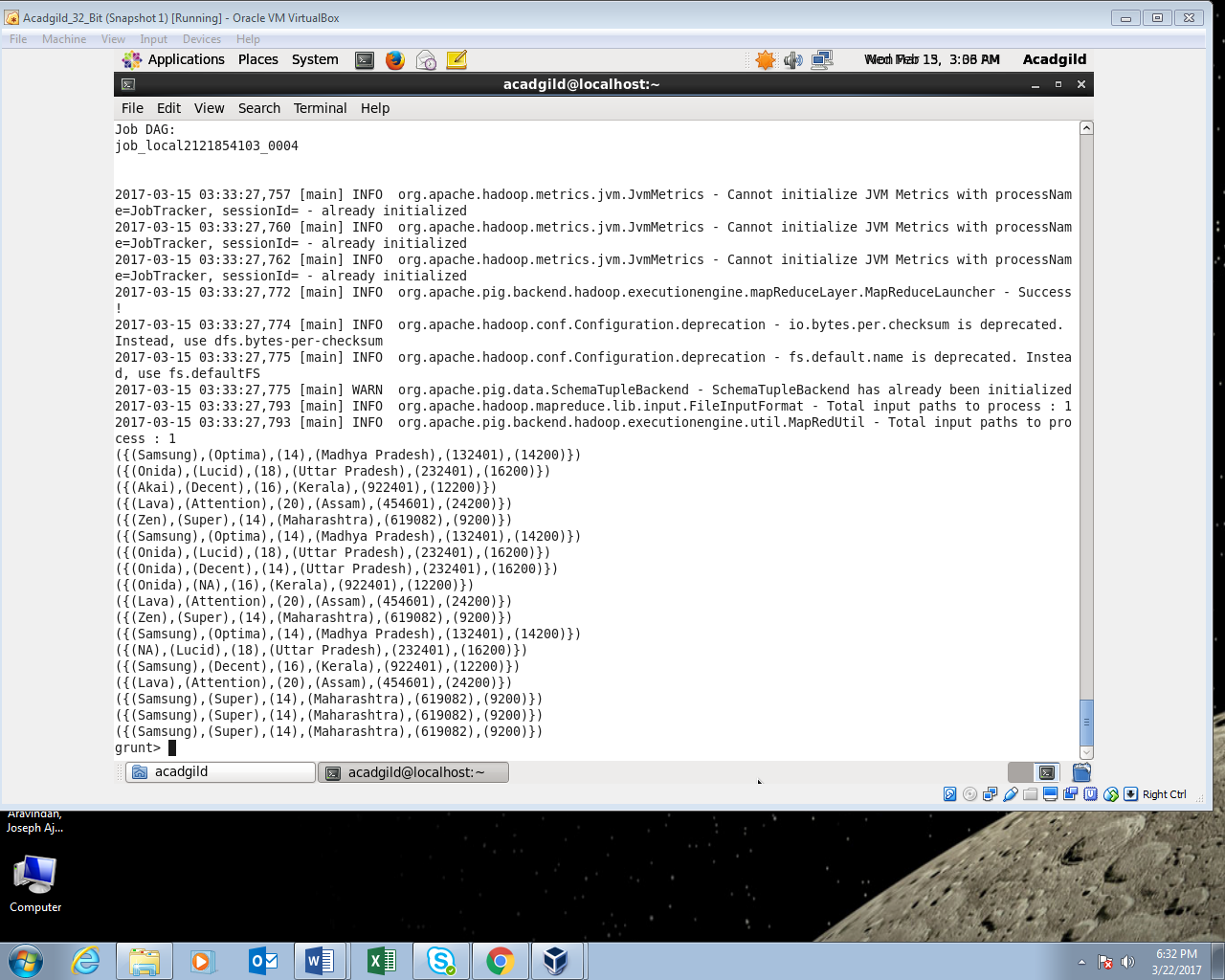
Syntax: TOKENIZE(expression)

**FOR EXAMPLE:**

**Input:**



**Output:**



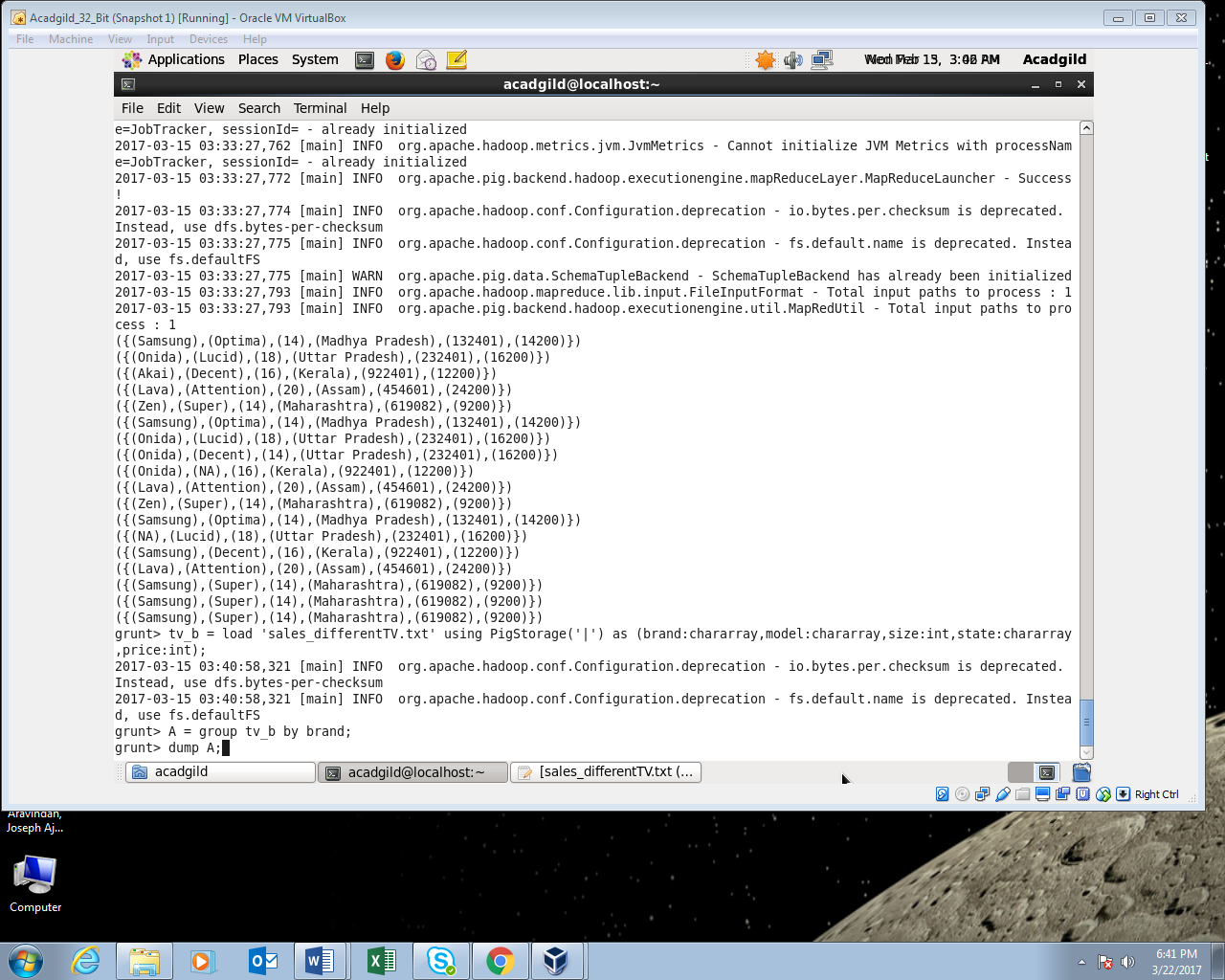
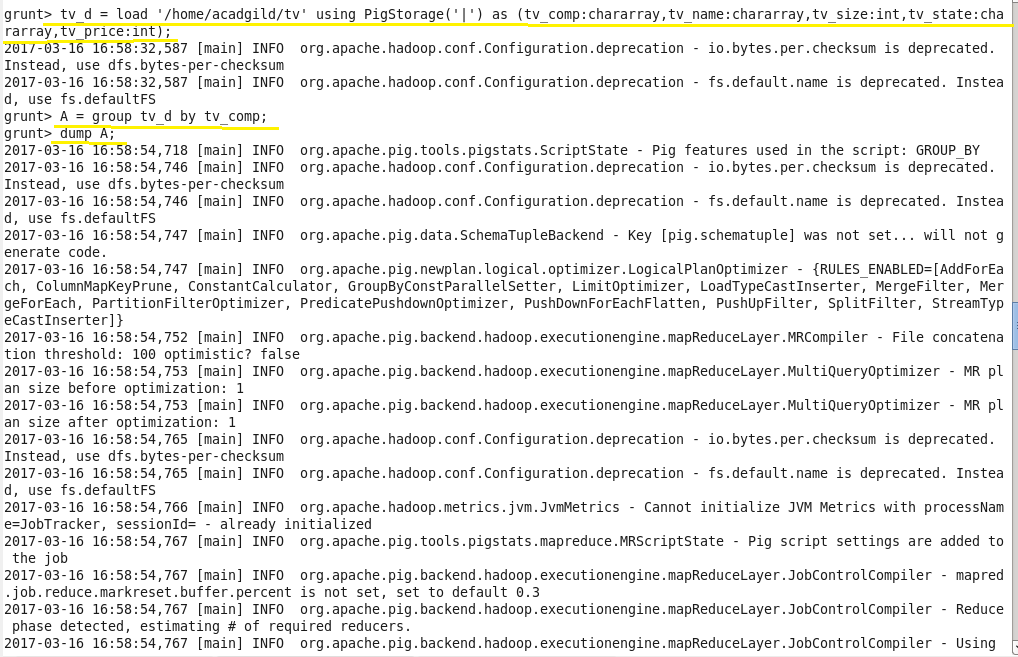
* **Sum:**

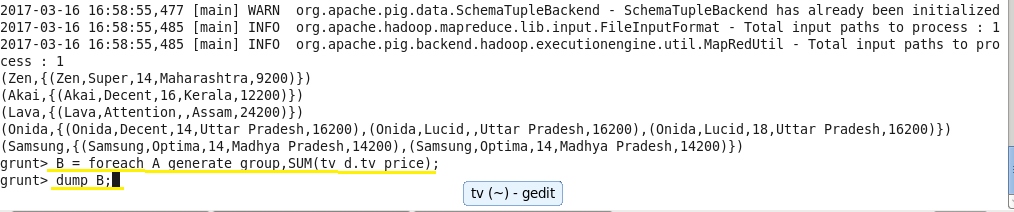
Computes the sum of the numeric values in a single-column bag. SUM requires a preceding GROUP ALL statement for global sums and a GROUP BY statement for group sums.

Syntax: SUM(expression)

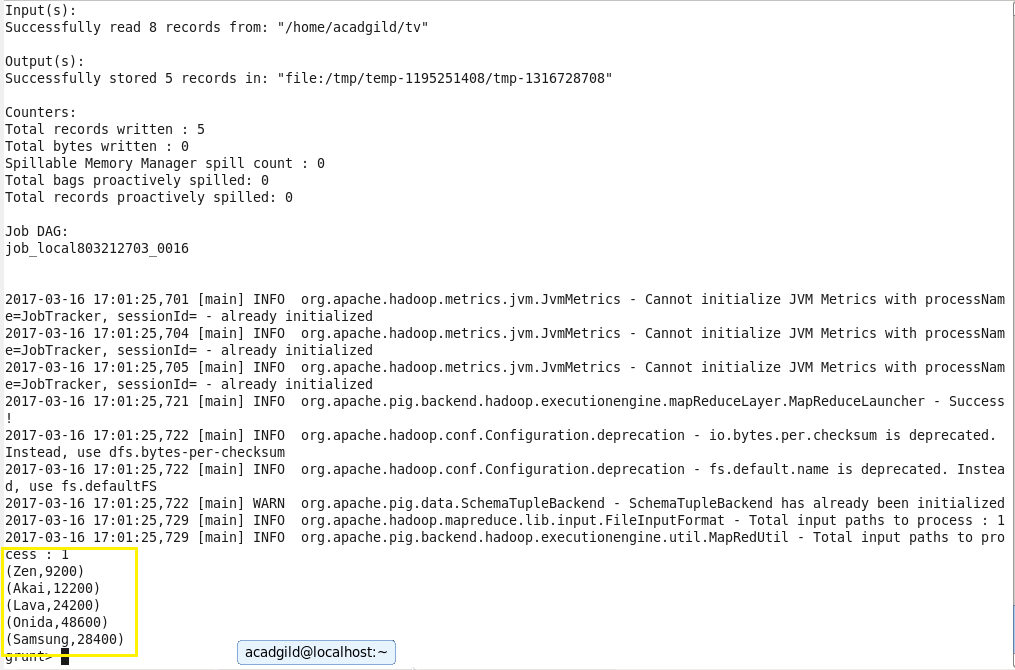
**For Example:**

**Input:**





**Output:**



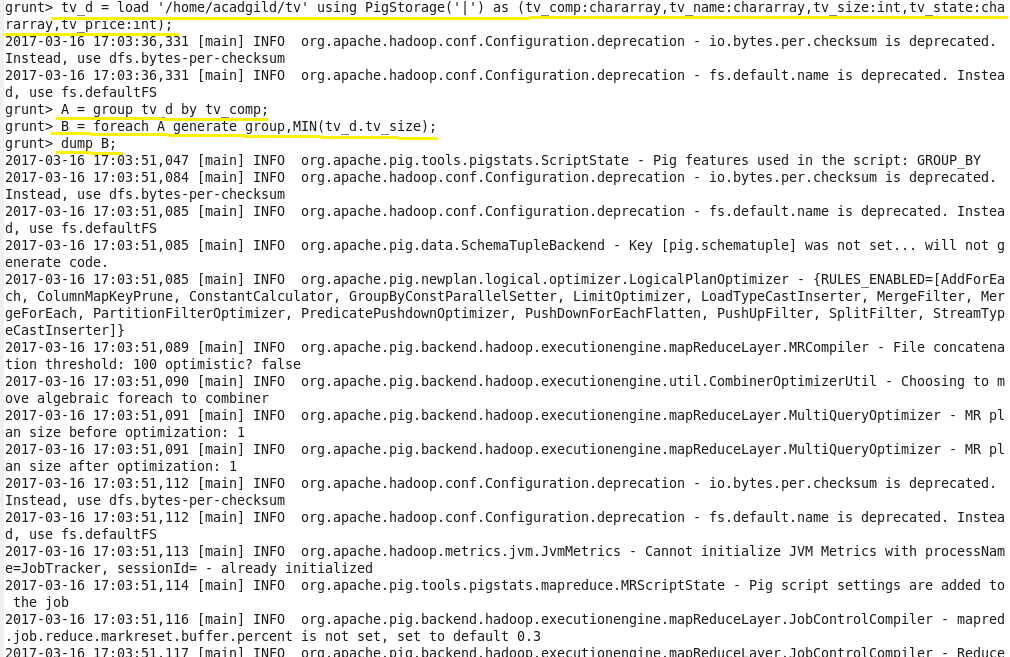
* **Min**:

Computes the minimum of the 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.

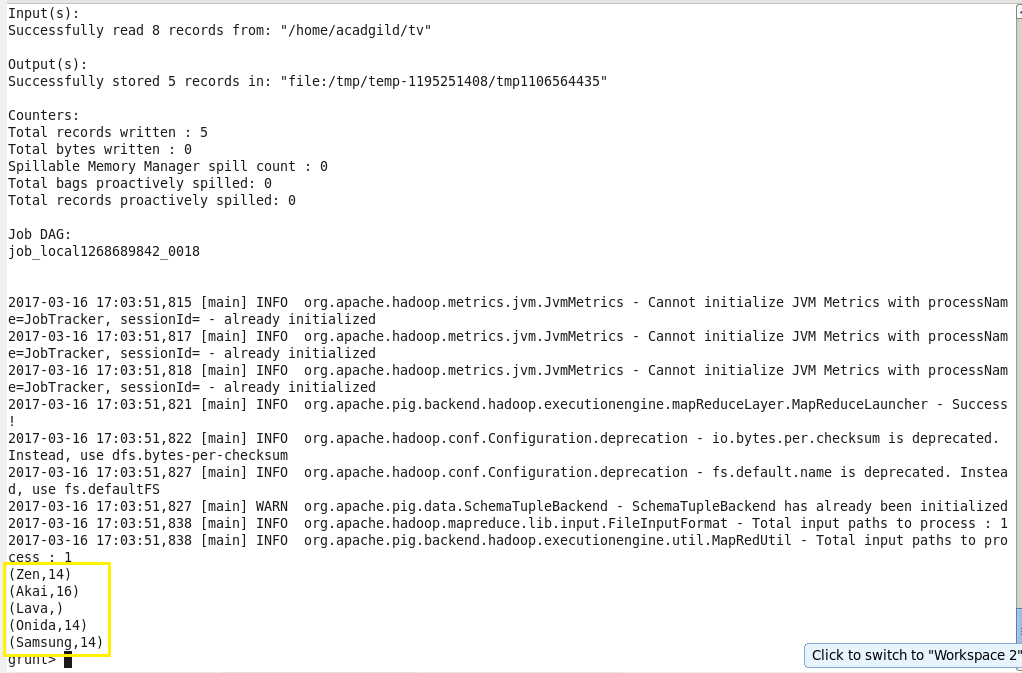
Syntax: MIN(expression)

**For Example:**

**Input:**



**Output:**



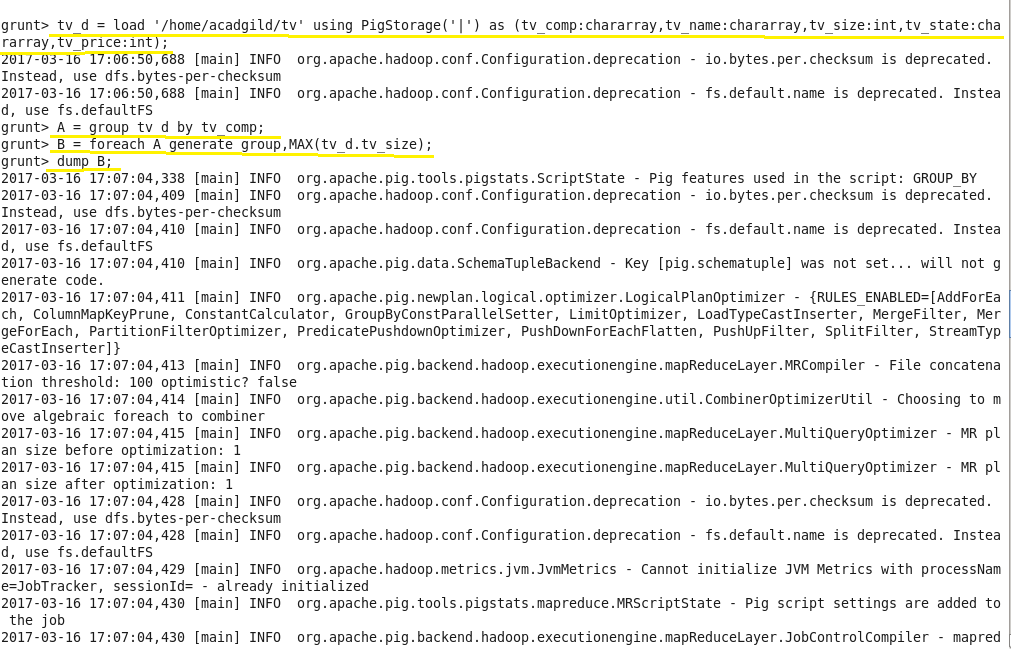
* **Max:**

Computes 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.

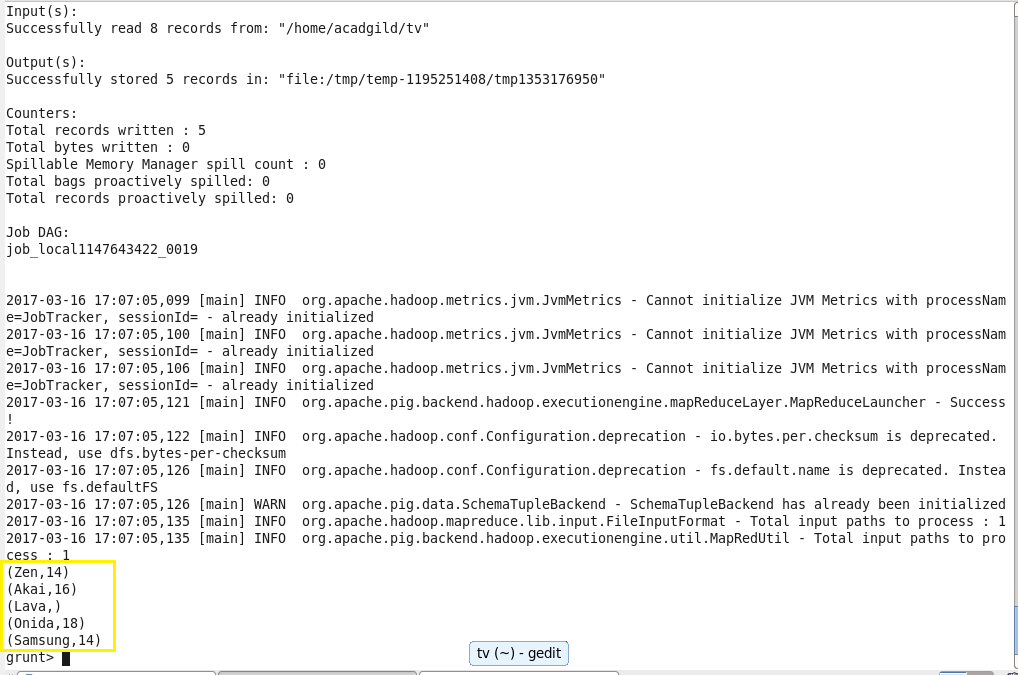
Syntax: MAX(expression)

**For example:**

**Input:**



**Output:**



* **Limit:**

Limits the number of output tuples.

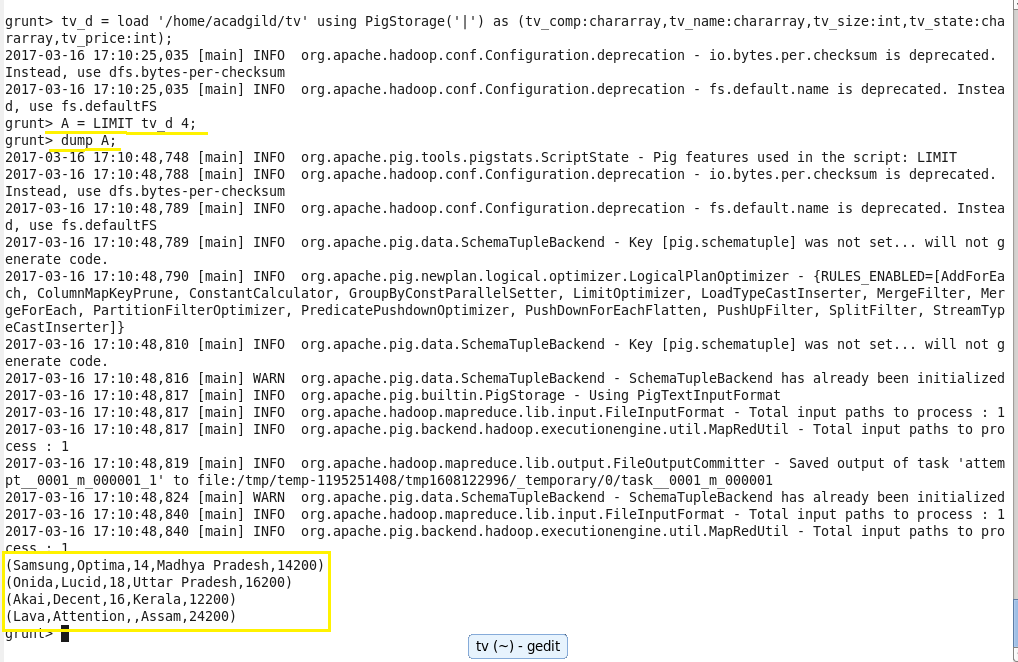
Syntax: alias = LIMIT alias n;

Where,

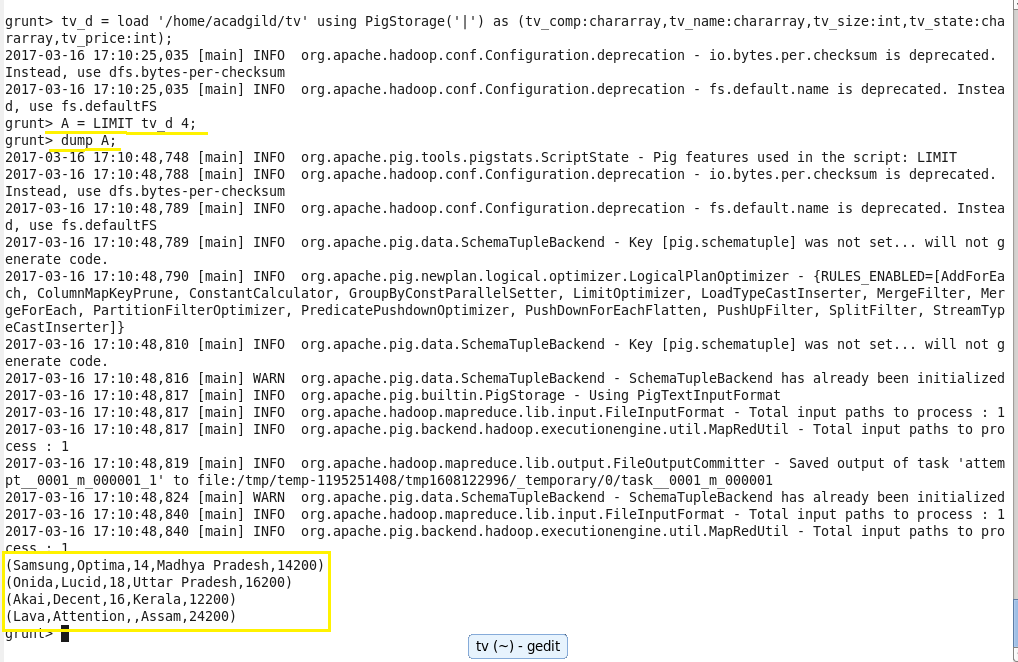
Alias-The name of a relation. n-The number of tuples.

**For Example:**

**Input:**



**Output:**



* **Store:**

Stores or saves results to the file system.

Syntax: STORE alias INTO 'directory' [USING function];

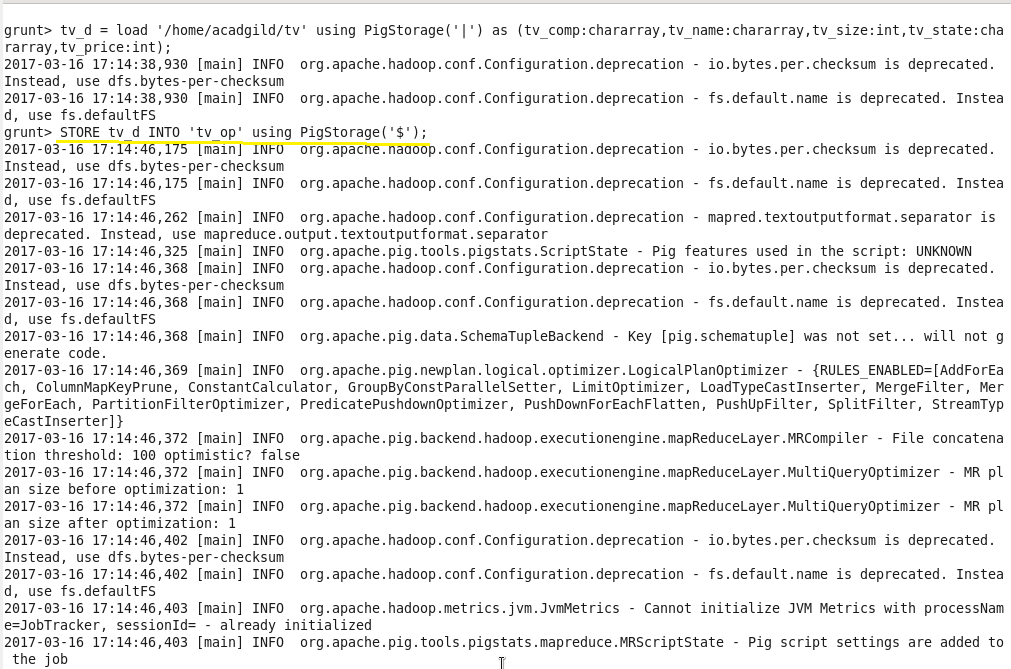
Where,

Alias-The name of a relation. INTO-Required keyword. 'directory'-The name of the storage directory, in quotes. If the directory already exists, the STORE operation will fail.

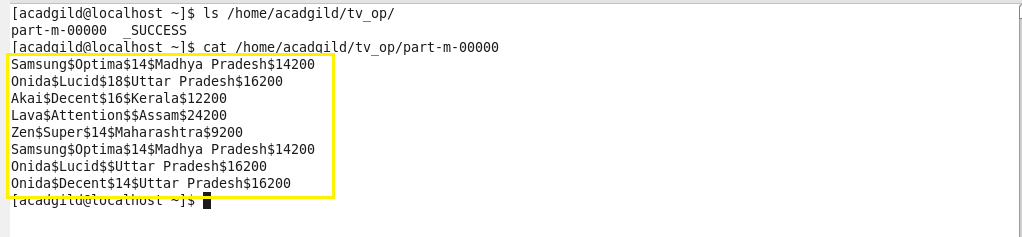
Function-The store function.

**For Example:**

**Input:**



**Output:**



* **Distinct:**

Removes duplicate tuples in a relation.

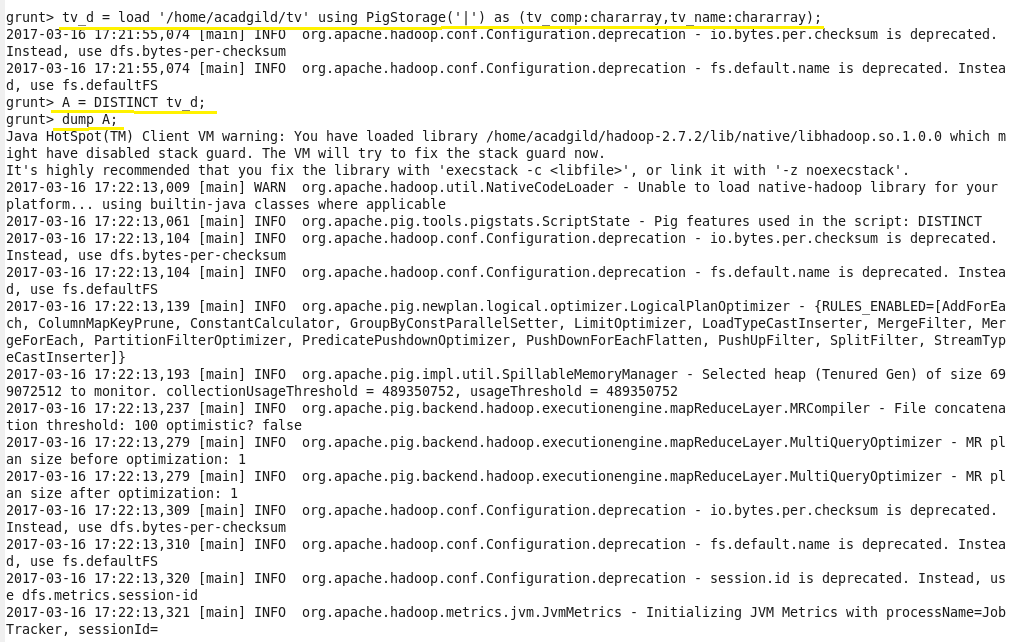
Syntax: alias = DISTINCT alias [PARALLEL n];

where,

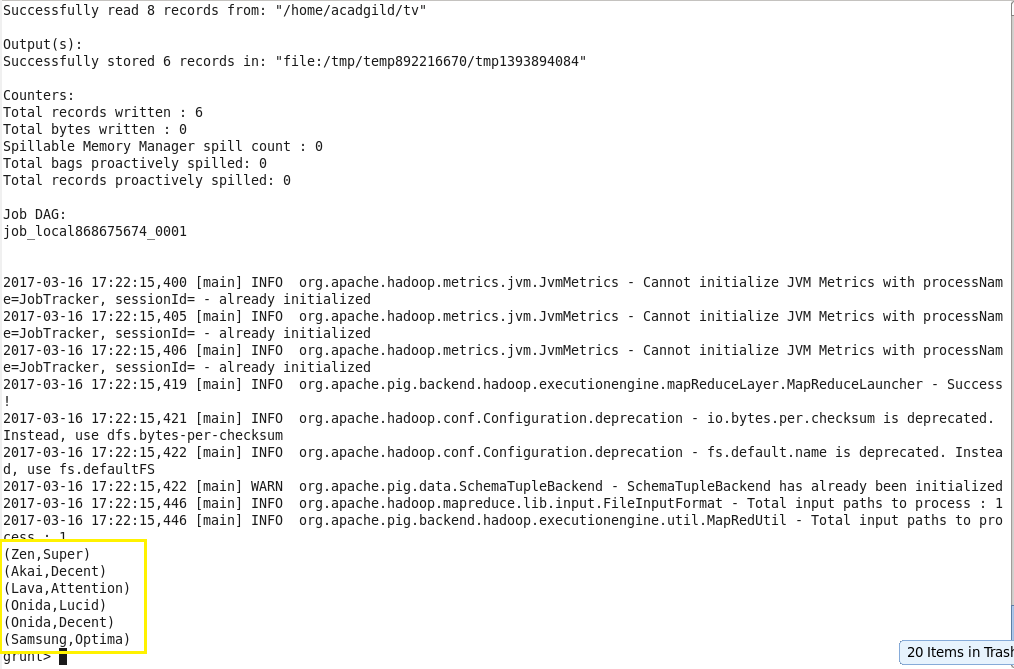
Alias-The name of the relation. PARALLEL n-Increase the parallelism of a job by specifying the number of reduce tasks, n. The default value for n is 1 (one reduce task)

**For Example:**

**Input:**



**Output:**

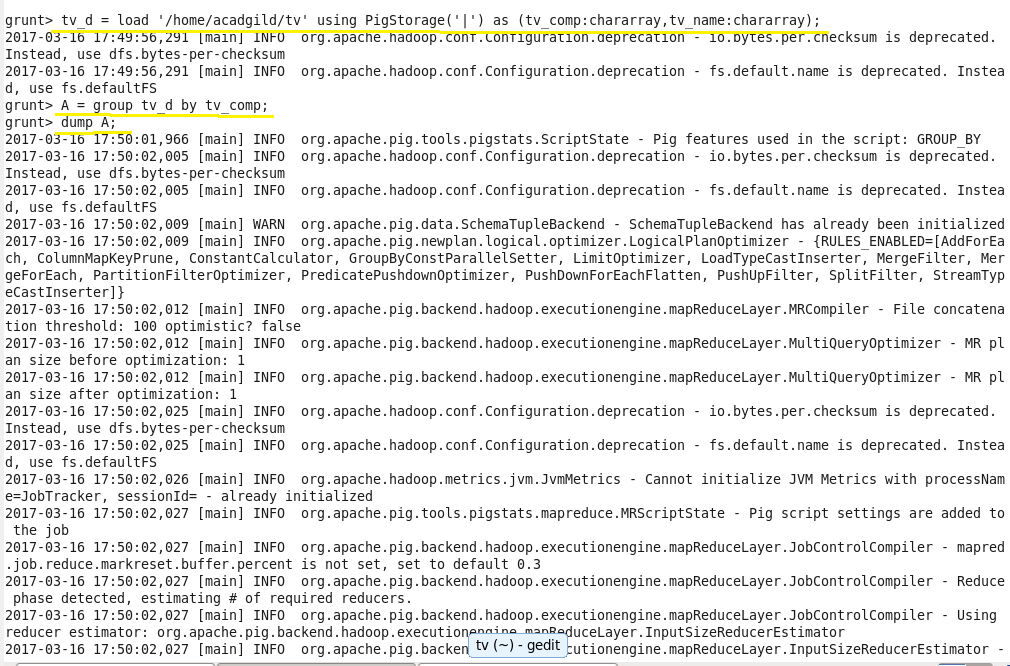


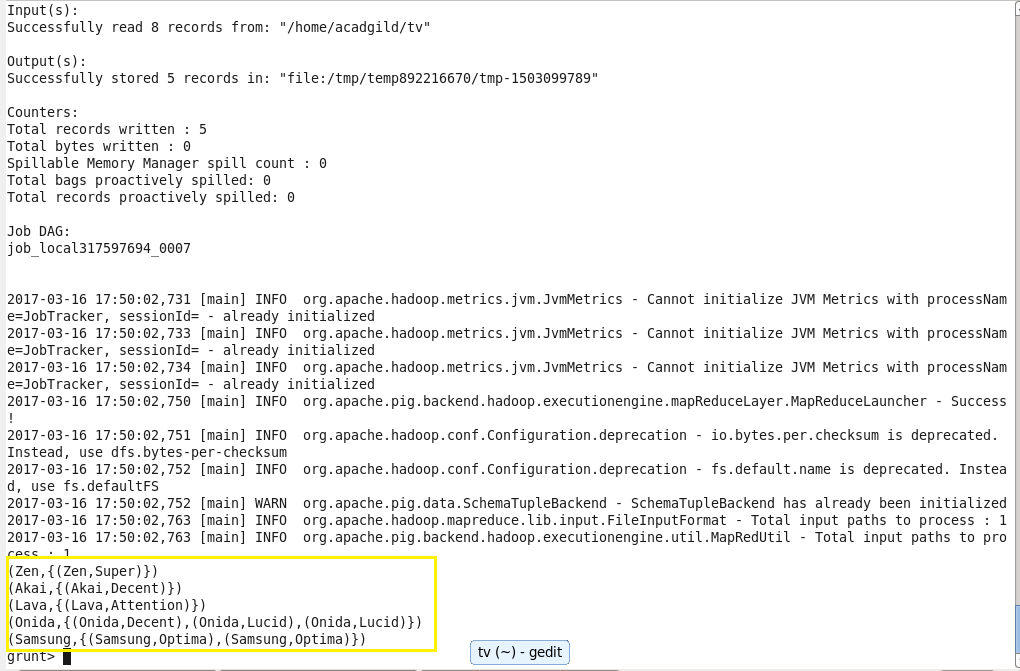
* **Flatten:**

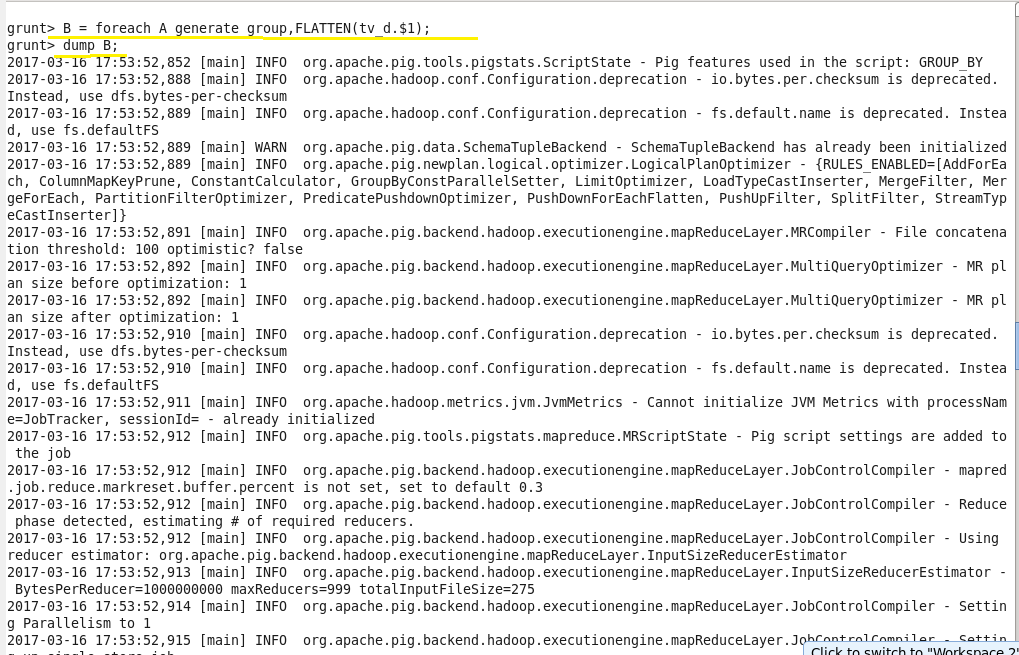
The FLATTEN operator looks like a UDF syntactically, but it is actually an operator that changes the structure of tuples and bags in a way that a UDF cannot. Flatten un-nests tuples as well as bags.

**For Example:**

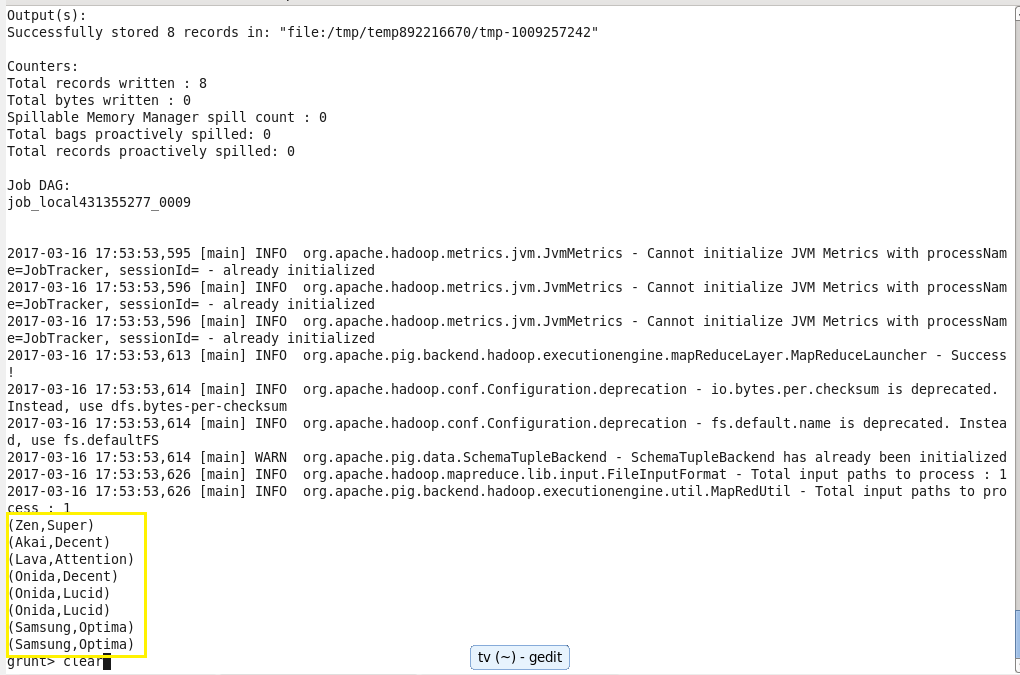
**Input:**







**Output:**



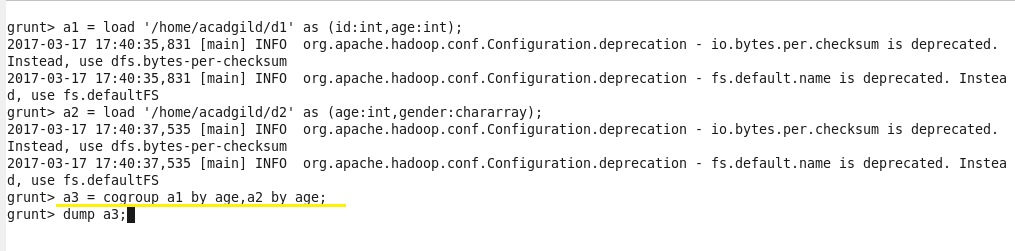
* **IsEmpty :**

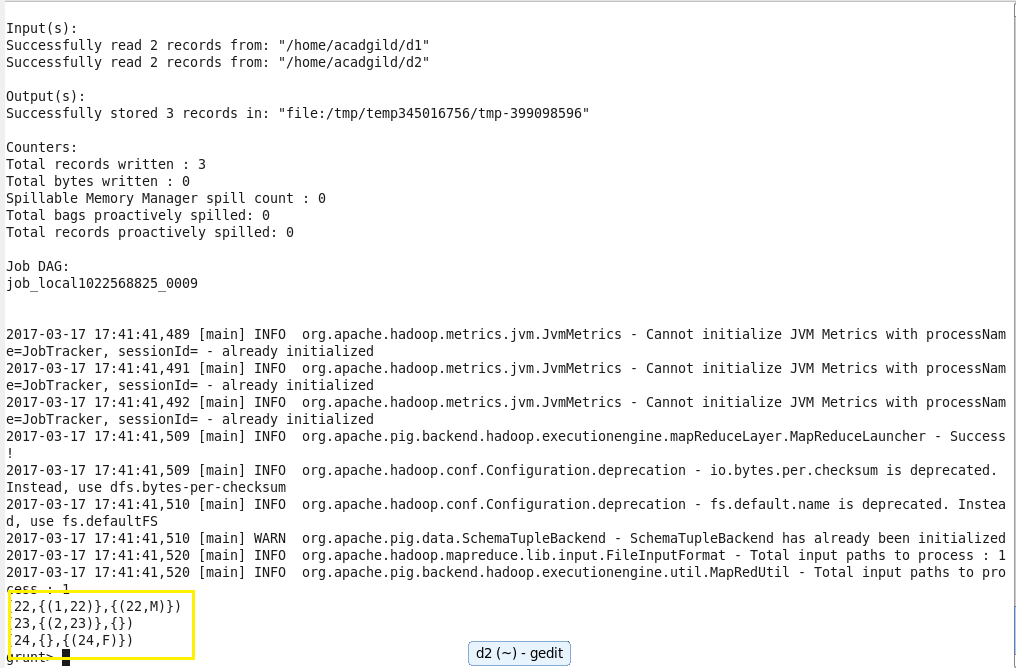
Checks if a bag or map is empty.

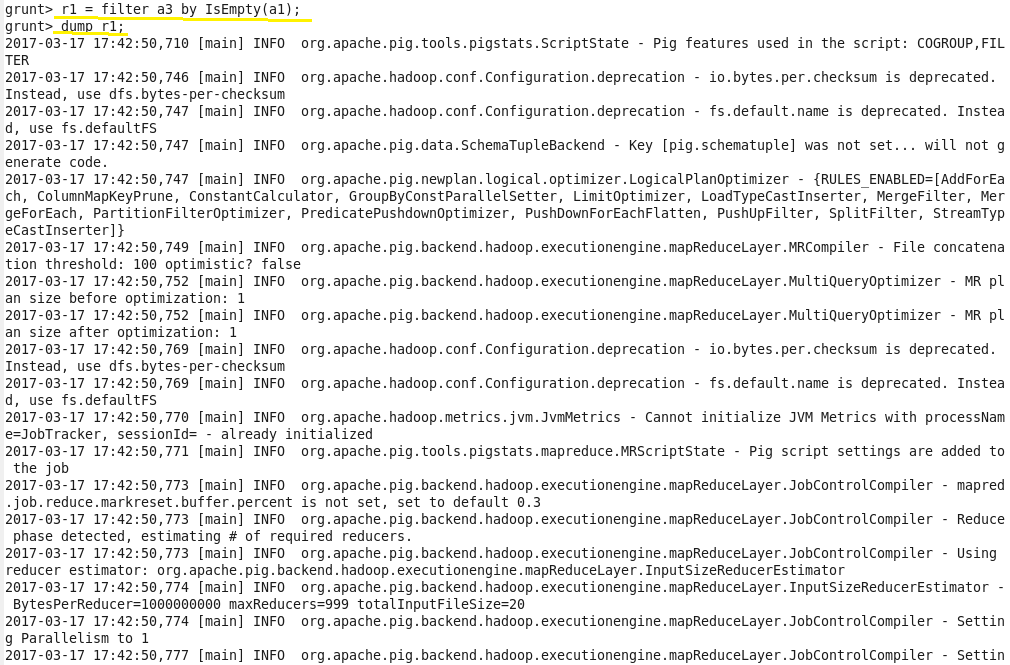
Syntax: IsEmpty(expression)

**For Example:**

**Input:**







**Output:**

