**Assignment 4.2**

**Problem Statement:**

**Create a sample dataset and implement the below Pig commands on the same dataset.**

1,The Nightmare Before Christmas,1993,3.9,4568

2,The Mummy,1932,3.5,4388

3,Orphans of the Storm,1921,3.2,9062

4,The Object of Beauty,1991,2.8,6150

5,Night Tide,1963,2.8,5126

6,One Magic Christmas,1985,3.8,5333

7,Muriel's Wedding,1994,3.5,6323

8,Mother's Boys,1994,3.4,5733

9,Nosferatu: Original Version,1929,3.5,5651

10,Nick of Time,1995,3.4,5333

**1) Concat**

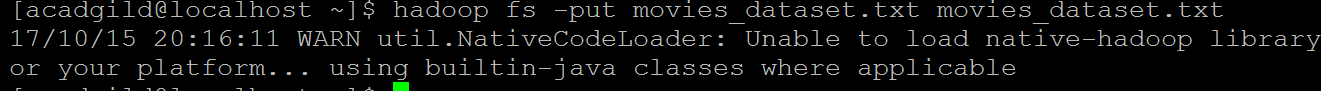
The CONCAT() function of Pig Latin is used to concatenate two or more expressions of the same type.

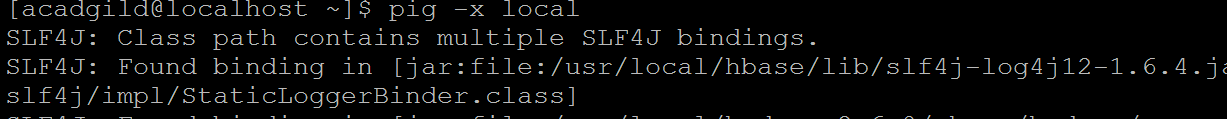
hadoop fs -put movies\_dataset.txt movies\_dataset.txt

movies = LOAD ' movies\_dataset.txt' USING PigStorage(',') as (id:int,name:chararray,year:chararray,rating:double,duration:int);

Dump movies;

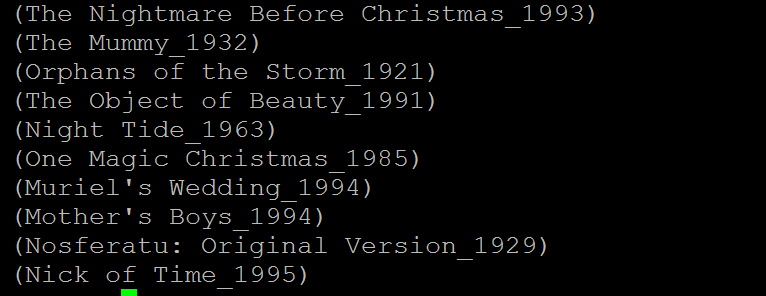
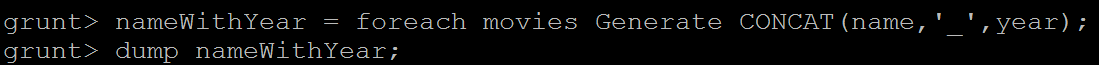
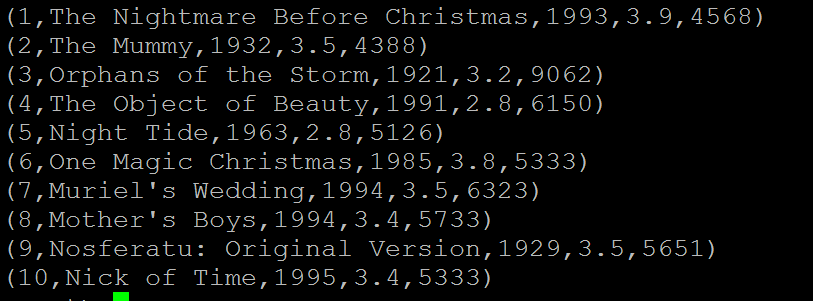
nameWithYear = foreach movies Generate CONCAT(name,'\_',year)







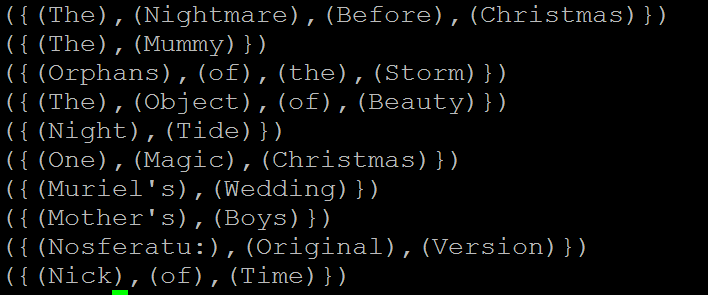




**2) Tokenize**

The Tokenize function of Pig Latin is used 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.

command used : moviename = foreach movies Generate TOKENIZE(name);



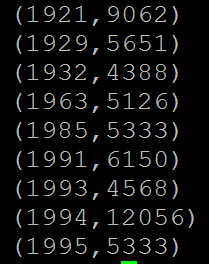
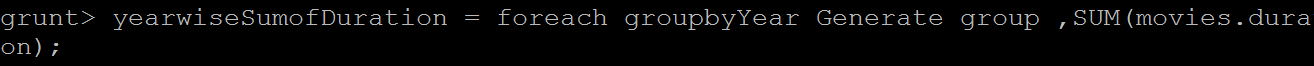
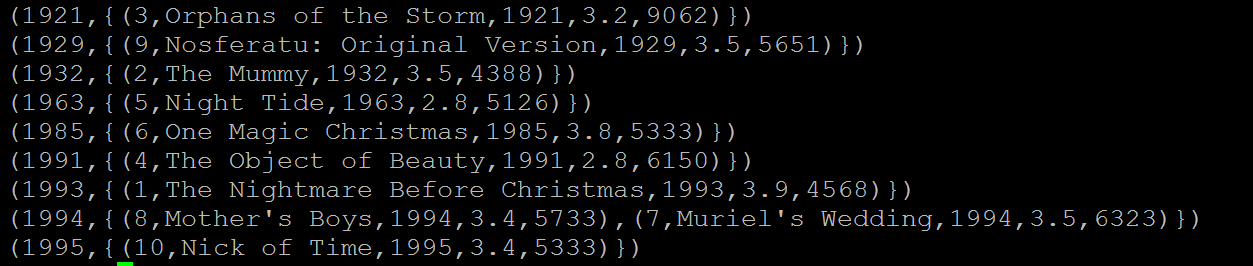
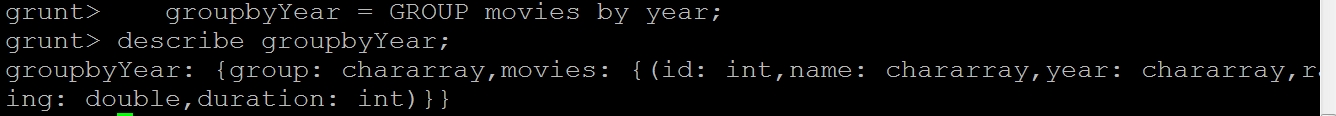
**3) Sum**

This is aggregate function and used with GROUP function. It computes the sum of the numeric values in a single-column bag. While computing the total, the SUM() function ignores the NULL values.

To Find Year wise total duration of all movies we use sum() operator on Group

groupbyYear = GROUP movies by year;

yearwiseSumofDuration = foreach groupbyYear Generate group ,SUM(movies.duration);



**4) 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.

Data Set:

001,Rajiv,Reddy,21,9848022337,Hyderabad,89

002,siddarth,Battacharya,22,9848022338,Kolkata,78

003,Rajesh,Mathur,22,9848022319,Delhi,90

004,Preethi,Agarwal,21,9848022330,Pune,93

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75

006,Archana,Mishra,23,9848022335,Chennai,87

007,Komal,Nayak,24,9848022334,trivendram,83

008,Bharathi,Nambiayar,24,9848022333,Chennai,72

Calculating the Minimum GPA

student\_details = LOAD 'studentdetails.txt' USING PigStorage(',')

as (id:int, firstname:chararray, lastname:chararray, age:int,

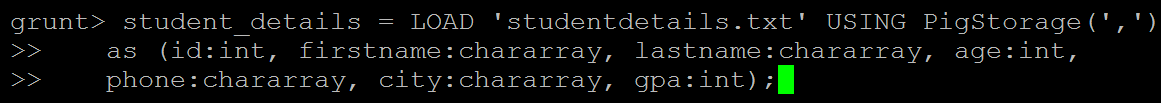
phone:chararray, city:chararray, gpa:int);

student\_group\_all = Group student\_details All;

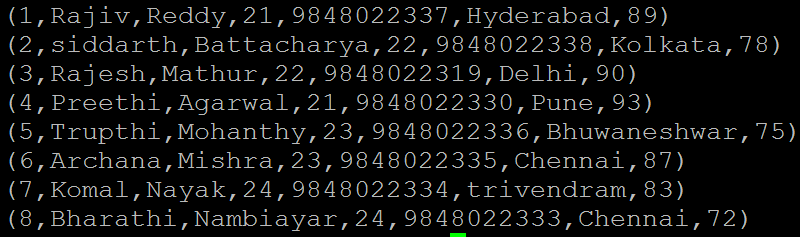
Dump student\_group\_all;

student\_gpa\_min = foreach student\_group\_all Generate (student\_details.firstname, student\_details.gpa), MIN(student\_details.gpa);

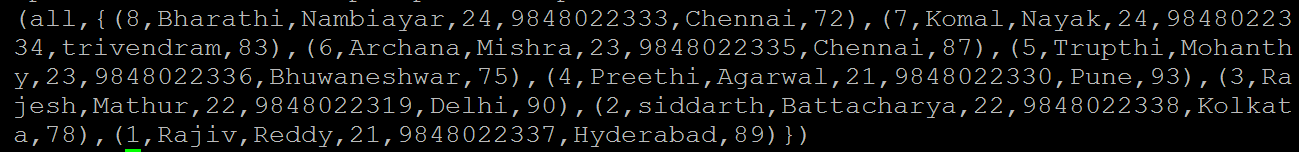
Dump student\_gpa\_ min;

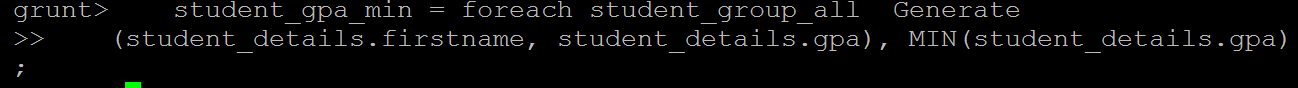
















**5) Max**

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.

Data Set: same as pervious section

001,Rajiv,Reddy,21,9848022337,Hyderabad,89

002,siddarth,Battacharya,22,9848022338,Kolkata,78

003,Rajesh,Mathur,22,9848022319,Delhi,90

004,Preethi,Agarwal,21,9848022330,Pune,93

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75

006,Archana,Mishra,23,9848022335,Chennai,87

007,Komal,Nayak,24,9848022334,trivendram,83

008,Bharathi,Nambiayar,24,9848022333,Chennai,72

we want to calculating the Maximum GPA

student\_details = LOAD 'studentdetails.txt' USING PigStorage(',')

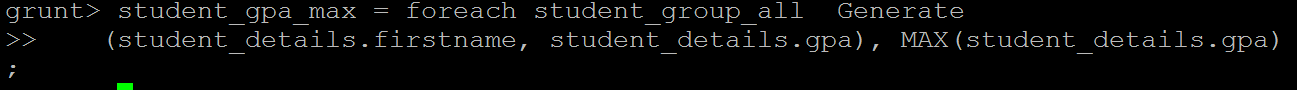
as (id:int, firstname:chararray, lastname:chararray, age:int,

phone:chararray, city:chararray, gpa:int);

student\_group\_all = Group student\_details All;

student\_gpa\_max = foreach student\_group\_all Generate (student\_details.firstname, student\_details.gpa), MAX(student\_details.gpa);

Dump student\_gpa\_max;







**6) Limit**

The LIMIT operator is used to get a limited number of tuples from a relation.

Often you are not interested in the entire output but rather a sample or top results. In such cases, using LIMIT can yield a much better performance as we push the limit as high as possible to minimize the amount of data travelling through the pipeline.

Data Set: same as pervious section

001,Rajiv,Reddy,21,9848022337,Hyderabad,89

002,siddarth,Battacharya,22,9848022338,Kolkata,78

003,Rajesh,Mathur,22,9848022319,Delhi,90

004,Preethi,Agarwal,21,9848022330,Pune,93

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75

006,Archana,Mishra,23,9848022335,Chennai,87

007,Komal,Nayak,24,9848022334,trivendram,83

008,Bharathi,Nambiayar,24,9848022333,Chennai,72

we want to limit the number of records to 4 for relation students.

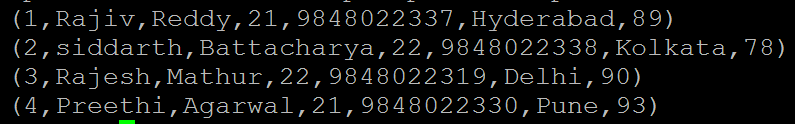
student\_details = LOAD 'studentdetails.txt' USING PigStorage(',')

as (id:int, firstname:chararray, lastname:chararray, age:int,

phone:chararray, city:chararray, gpa:int);

limit\_data = LIMIT student\_details 4;

dump limit\_data;



**7) Store**

Stores or saves results to the file system.

the syntax of the Store statement.

STORE Relation\_name INTO ' required\_directory\_path ' [USING function];

Data Set: same as pervious section

001,Rajiv,Reddy,21,9848022337,Hyderabad,89

002,siddarth,Battacharya,22,9848022338,Kolkata,78

003,Rajesh,Mathur,22,9848022319,Delhi,90

004,Preethi,Agarwal,21,9848022330,Pune,93

005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar,75

006,Archana,Mishra,23,9848022335,Chennai,87

007,Komal,Nayak,24,9848022334,trivendram,83

008,Bharathi,Nambiayar,24,9848022333,Chennai,72

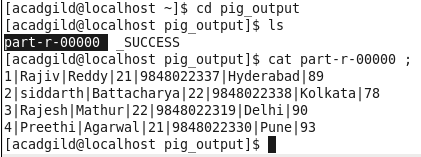
student\_details = LOAD 'studentdetails.txt' USING PigStorage(',')

as (id:int, firstname:chararray, lastname:chararray, age:int,

phone:chararray, city:chararray, gpa:int);

limit\_data = LIMIT student\_details 4;

store limit\_data into 'pig\_output/' USING PigStorage('|');



**8) Distinct**

The DISTINCT operator is used to remove redundant (duplicate) tuples from a relation. DISTINCT does not preserve the original order of the contents

movies\_with\_dups = LOAD 'moviesdataset\_duplicates.txt'

USING PigStorage(',')

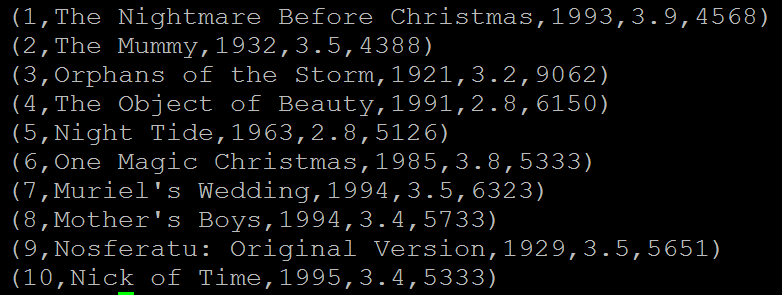
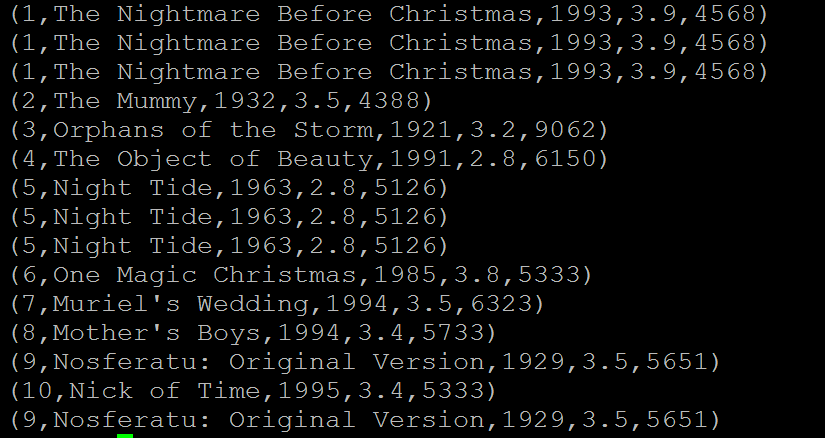
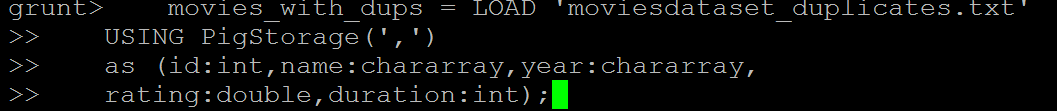
as (id:int,name:chararray,year:chararray,

rating:double,duration:int);

dump movies\_with\_dups

no\_dups = DISTINCT movies\_with\_dups;

DUMP no\_dups;

****

**9) 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. The idea is the same, but the operation and result is different for each type of structure.

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).

DataSets :

bus,cycle,train

train,areoplane,motorbike

bus,boat,ship,cycle

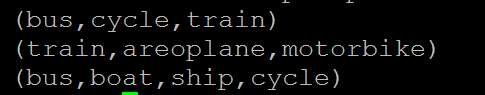
wordfile = load 'wordsInTransport.txt' using PigStorage(',');

flattenWord = foreach wordfile generate flatten(TOKENIZE((chararray)$0)) as word;

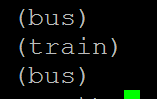
dump flattenWord;











**10) IsEmpty**

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

DataSets

**emp\_sales.txt**

1,Robin,22,25000,sales

2,BOB,23,30000,sales

3,Maya,23,25000,sales

4,Sara,25,40000,sales

5,David,23,45000,sales

6,Maggy,22,35000,sales

**emp\_bonus.txt**

1,Robin,22,25000,sales

2,Jaya,23,20000,admin

3,Maya,23,25000,sales

4,Alia,25,50000,admin

5,David,23,45000,sales

6,Omar,30,30000,admin

Find the tuples in **emp\_sales** relation that are not there in the relation **emp\_bonus**.

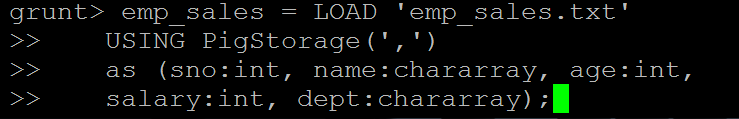
1. emp\_sales = LOAD 'emp\_sales.txt' USING PigStorage(',')

as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

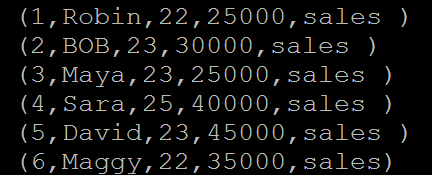
1. emp\_bonus = LOAD 'emp\_bonus.txt' USING PigStorage(',')

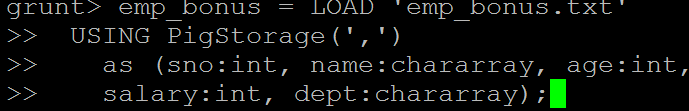
as (sno:int, name:chararray, age:int, salary:int, dept:chararray);

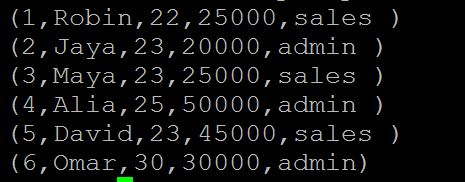
1. cogroup\_data = COGROUP emp\_sales by age, emp\_bonus by age;
2. isempty\_data = filter cogroup\_data by IsEmpty(emp\_sales);



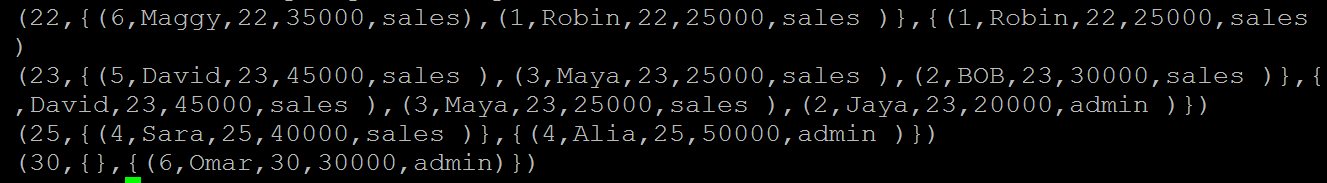












**isempty\_data tells** the **emp\_sales** relation tuples that are not there in the relation **emp\_bonus**.

