**Pig scripts used in the session:**

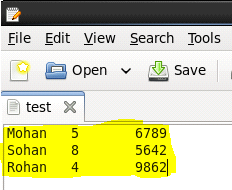
**A: Load/Store functions**

**PigStorage():** PigStorage is the default function for the LOAD and STORE operators and works with both simple and complex data types.

PigStorage expects data to be formatted using field delimiters, either the tab character ('\t') or other specified character.

**Note: Tab is the default delimiter in Pig and in the case data is delimited by tab then we do not need to specify the PigStorage() FUNCTION explicitly but in the case dataset is delimited by some other delimiter like comma or semicolon then we need to specify the delimiter in the PigStorage().**

Refer the sample Tab separated data:



A = load '/test' as (name:chararray,roll\_no:int,salary:int);

dump A;

STORE A INTO '/output' USING PigStorage('\*');

**B: Relational operators**

**Format of test\_file.txt:(tab separated values)**

1 2 3

4 5 6

7 8 9

**First copy the file input file to HDFS.**

hadoop dfs -put test\_file.txt /

**1 . LOAD COMMAND:** Loads data from the file system.

In this example the default load function, PigStorage, loads data from myfile.txt to form relation A. The two LOAD statements are equivalent. Note that, because no schema is specified, the fields are not named and all fields default to type bytearray

A = load '/test\_file.txt' using PigStorage('\t');

dump A;

**In this example a schema is specified using the AS keyword.**

A = LOAD '/test\_file.txt' USING PigStorage('\t') AS (f1:int, f2:int, f3:int);

dump A;

**2. FILTER COMMAND:**Selects tuples from a relation based on some condition.

In this example the condition states that if the third field equals 3, then include the tuple with relation X.

X = FILTER A BY f3 == 3;

DUMP X;

**In this example the condition states that if the first field equals 8 or if the sum of fields f2 and f3 is not greater than first field, then include the tuple relation X.**

X = FILTER A BY (f1 == 8) OR (NOT (f2+f3 > f1));

dump X;

**3. FOREACH COMMAND:** Generates data transformations based on columns of data.

X = FOREACH A GENERATE f1;

**4. GROUP COMMAND:**

**stud\_details dataset:**

Aashish,23,Delhi

Srinivas,34,Chennai

Aashish,38,Mumbai

John,24,Patna

Srinivas,29,Bangalore

**Scripts:**

A = load '/stud\_details' using PigStorage(',') AS (name:chararray,age:int,city:chararray);

B = GROUP A BY name;

dump B;

**5. JOIN COMMAND(INNER):** Performs inner, equijoin of two or more relations based on common field values. The JOIN operator always performs an inner join. Inner joins ignore null keys, so it makes sense to filter them out before the join.

**Dataset1:** Test1 **Dataset2:** Test2

1,4,8 1,9

4,6,7 4,9

3,5,8 6,2

A = LOAD '/Test1' using PigStorage(',') AS (a1:int,a2:int,a3:int);

B = LOAD '/Test2' ' using PigStorage(',') AS (b1:int,b2:int);

X = JOIN A BY a1, B BY b1;

dump X;

**6. STORE COMMAND:**

Stores or saves results to the file system.

Scripts:

A = LOAD '/Test1' ' using PigStorage(',') AS (a1:int,a2:int,a3:int);

dump A;

STORE A INTO '/result' USING PigStorage ('\*');

**From command line:** hadoop dfs -ls /result

hadoop dfs -cat /result/part-m-00000

**7. UNION COMMAND:**

Merge the contents of two or more relations.It does not preserve the order of tuples. Both the input and output relations are interpreted as unordered bags of tuples.

A = LOAD '/Test1' ' using PigStorage(',') AS (a1:int,a2:int,a3:int);

B = LOAD '/Test2' ' using PigStorage(',') AS (b1:int,b2:int);

X = UNION A, B;

**8. SAMPLE:**

Partitions a relation into two or more relations.

A = LOAD '/Test1' ' using PigStorage(',') AS (f1:int,f2:int,f3:int);

X = SAMPLE A 0.01;

**9. SPLIT OPERATOR:**

SPLIT operator partitions the contents of a relation into two or more relations based on some expression.

A = LOAD '/Test1' ' using PigStorage(',') AS (f1:int,f2:int,f3:int);

dump A;

SPLIT A INTO X IF f1<7, Y IF f2==5, Z IF (f3<6 OR f3>6);

dump X;

dump Y;

dump Z;

**C. EVAL FUNCTIONS:**

**10. TOKENIZE:**

TOKENIZE function split a string of words (all words in a single tuple) into a bag of words (each word in a single tuple).

The following characters are considered to be word separators: space, double quote("), coma(,) parenthesis(()), star(\*).

**Format of the test\_data.**

(Here is the first string.)

(Here is the second string.)

(Here is the third string.)

A = LOAD '/test\_data' using PigStorage(',') AS (f1:chararray);

X = FOREACH A GENERATE TOKENIZE(f1);

DUMP X;

**11. AVG:**

AVG function computes the average of the numeric values in a single-column bag. AVG requires a preceding GROUP ALL statement for global averages and a GROUP BY statement for group averages.

**Employee dataset:**

Ram,IT,23000

Ram,Food,25000

Shyam,Security,12000

Rahul,mechanics,67000

Shyam,HR,29000

A = LOAD '/employee' using PigStorage(',') AS (name:chararray, dept:chararray, gpa:float);

B = GROUP A BY name;

C = FOREACH B GENERATE A.name, AVG(A.gpa);

**12.Max and Min:**

A = LOAD '/employee' using PigStorage(',') AS (name:chararray, session:chararray, gpa:float);

B = GROUP A BY name;

X = FOREACH B GENERATE group, MAX(A.gpa);

X = FOREACH B GENERATE group, MIN(A.gpa);

**13. COUNT:**

Computes the number of elements in the bag.

COUNT function compute the number of elements in a bag.

**Count\_input dataset**

**1,5,8**

**4,8,9**

**2,1,7**

**6,7,8**

A = LOAD '/Count\_input' using PigStorage(',') AS (f1:int,f2:int,f3:int);

B = GROUP A BY f1;

X = FOREACH B GENERATE COUNT(A);

**14. SUM**

SUM function computes the sum of a set of numeric values in a single-column bag.

**Pet\_data dataset**

David,dog,2

John,cat,3

Maria,turtle,5

Shiv,goldfish,6

A = LOAD '/pet\_data' using PigStorage(',') AS (owner:chararray, pet\_type:chararray, pet\_num:int);

B = GROUP A BY owner;

X = FOREACH B GENERATE group, SUM(A.pet\_num);

D: DIAGNOSTIC OPERAATORS

**D: DIAGNOSTIC OPERATORS:**

**15. DESCRIBE**

Returns the schema of an alias.

A = LOAD '/pet\_data' using PigStorage(',') AS (owner:chararray, pet\_type:chararray, pet\_num:int);

B = GROUP A BY owner;

X = FOREACH B GENERATE group, SUM(A.pet\_num);

DESCRIBE A;

DESCRIBE B;

DESCRIBE C;