Experiment No:7

Aim: Write Pig Latin scripts to perform data processing operations.

a)Grouping and joining data.

b)Sorting data

c)Combining and Splitting data.

Procedure:

Grouping data:

Syntax: variablename= group filename by Columnname;

1st-method : Create file in local file system

[cloudera@localhost ~]$ cat > a.txt

10
20
30
40
50
10
20
30
40
60
10
20
30
40
50

Transform the file from local system to hadoop environment

cloudera@localhost ~]$hadoop fs –put /home/cloudera/a.txt /user/cloudera/pigexample

Open the pig environment

[cloudera@localhost ~]$pig

Load the data into PigStorage

grunt>A= load 'user/cloudera/pigexample/a.txt' using PigStorage() as (age:int);

grunt>dump A;

```
(10)
(20)
(30)
(40)
(50)
(10)
(20)
(30)
(40)
(60)
(10)
(20)
(30)
(40)
(50)
grunt>
```

grunt>gr= group A by age;

grunt>dump gr;

```
cess : 1
(10,{(10),(10),(10)})
(20,{(20),(20),(20)})
(30,{(30),(30),(30)})
(40,{(40),(40),(40)})
(50,{(50),(50)})
(60,{(60)})
grunt>
```

2nd method:

[cloudera@localhost ~]$ cat > b.txt

```
[cloudera@localhost ~]$ vi h1.txt
[cloudera@localhost ~]$ cat h1.txt
5D0,mahi,CSE
5G4,Divya,CSE
5H1,Padma,CSE
5H7,Mounika,CSE
484,Ramya,ECE
4A5,Raju,ECE
1234,Siri,IT
12B6,Charan,IT
4201,Pavani,AIDS
[cloudera@localhost ~]$
```

cloudera@localhost ~]$hadoop fs –put /home/cloudera/b.txt /user/cloudera/pigexample

[cloudera@localhost ~]$pig

grunt>A= load ‘user/cloudera/pigexample/b.txt’ using PigStorage(‘,’) as (rno:chararray,sname:chararray,branch:chararray);

grunt>dump A;

```
2023-10-05 23:13:08,476 [
cess : 1
(5D0,mahi,CSE)
(5G4,Divya,CSE)
(5H1,Padma,CSE)
(5H7,Mounika,CSE)
(484,Ramya,ECE)
(4A5,Raju,ECE)
(1234,Siri,IT)
(12B6,Charan,IT)
(4201,Pavani,AIDS)
grunt>
```

g=group A by branch;

dump g;

```
cess : 1
(IT,{(1234,Siri,IT),(12B6,Charan,IT)})
(CSE,{(5D0,mahi,CSE),(5G4,Divya,CSE),(5H1,Padma,CSE),(5H7,Mounika,CSE)})
(ECE,{(484,Ramya,ECE),(4A5,Raju,ECE)})
(AIDS,{(4201,Pavani,AIDS)})
grunt>
```

Sort:

Syntax: Variable= order data by attributename ASC/DESC;

Sort by Ascending order

grunt> sort1 = order data by age ASC;

grunt> dump sort1;

(12)
(19)
(24)
(24)
(25)
(27)
(35)
(35)
(45)
(55)
(65)

Sort by Descending order

grunt> sort2 = order data by age DESC;

grunt> dump sort2;

(65)
(55)
(45)
(35)
(35)
(27)

(25)
(24)
(24)
(19)
(12)

JOIN:

Joins can be of the following types −

- Self-join
- Inner-join
- Outer-join − left join, right join, and full join

```
cloudera@localhost ~]$ cat>a.txt
1,2,3
4,2,1
8,3,4
4,3,3
7,2,5
8,4,3
[cloudera@localhost ~]$ cat>b.txt
2,4
8,9
1,3
2,7
2,9
4,6
4,9
[cloudera@localhost ~]$ hadoop fs -put a.txt
[cloudera@localhost ~]$ hadoop fs -put b.txt
```

Self join

```
grunt> ONE= load 'a.txt' using PigStorage(',') as (a1:int,a2:int,a3:int);
grunt> TWO = load 'a.txt' using PigStorage(',') as (a1:int,a2:int,a3:int);
SELFJ = JOIN ONE by a1 , TWO BY a1;
grunt> describe SELFJ;
SELFJ: {ONE::a1: int,ONE::a2: int,ONE::a3: int,TWO::a1: int,TWO::a2: int,TWO::a3: int}
```

Equi-join.

```
grunt> A = load 'a.txt' using PigStorage(',') as (a1:int,a2:int,a3:int);
grunt> B = load 'b.txt' using PigStorage(',') as (b1:int,b2:int,b3:int);
grunt> X = Join A by a1, B by b1;
grunt> Dump X;
(1,2,3,1,3,)
(4,2,1,4,6,)
(4,2,1,4,9,)
(4,3,3,4,6,)
(4,3,3,4,9,)
(8,3,4,8,9,)
(8,4,3,8,9,)
```

Left outer join

```
A = LOAD  ‘A.txt' using PigStorage(',') AS (a1:int,a2:int,a3:int);
B = LOAD, ‘B.txt' using PigStorage(',') AS (b1:int,b2:int);
LEFTJ = JOIN A by a1 LEFT OUTER, B BY b1;
DUMP LEFTJ;
(1,2,3,1,3)
(4,3,3,4,9)
(4,3,3,4,6)
(4,2,1,4,9)
(4,2,1,4,6)
(7,2,5,,)
(8,4,3,8,9)
(8,3,4,8,9)
```

Right outer join

```
A = LOAD  ‘A.txt' using PigStorage(',') AS (a1:int,a2:int,a3:int);
B = LOAD, ‘B.txt' using PigStorage(',') AS (b1:int,b2:int);
RIGHTJ = JOIN A by a1 RIGHT OUTER, B BY b1;
DUMP RIGHTJ;
(1,2,3,1,3)
(,,,2,4)
(,,,2,7)
(,,,2,9)
(4,2,1,4,6)
(4,2,1,4,9)
(4,3,3,4,6)
(4,3,3,4,9)
(8,3,4,8,9)
(8,4,3,8,9)
```

Full join

```
A = LOAD  ‘A.txt' using PigStorage(',') AS (a1:int,a2:int,a3:int);
B = LOAD, ‘B.txt' using PigStorage(',') AS (b1:int,b2:int);
FULLJ = JOIN A by a1 FULL, B BY b1;
DUMP FULLJ;
(1,2,3,1,3)
(,,,2,4)
(,,,2,7)
(,,,2,9)
(4,2,1,4,6)
(4,2,1,4,9)
(4,3,3,4,6)
(4,3,3,4,9)
(7,2,5,,)
(8,3,4,8,9)
(8,4,3,8,9)
```

UNION & SPLIT

UNION combines multiple relations together whereas SPLIT partitions a relation in to multiple ones.

```
grunt> cat a.txt
1,2,3
4,2,1
8,3,4
grunt> cat b.txt
4,3,3
7,2,5
8,4,3
grunt> a = load 'a.txt' using PigStorage(',') as (a1:int, a2:int, a3:int);
grunt> b = load 'b.txt' using PigStorage(',') as (b1:int, b2:int, b3:int);
grunt> dump a;
(1,2,3)
(4,2,1)
(8,3,4)
grunt> dump b;
(4,3,3)
(7,2,5)
(8,4,3)
grunt> c = UNION a, b;
(1,2,3)
(4,2,1)
(8,3,4)
(4,3,3)
(7,2,5)
(8,4,3)
```

SPLIT:

```
grunt> SPLIT c into sp1 if $0 == 4, sp2 if $0 == 8;
```

Split operation on ‘c’ sends a tuple to sp1 if its first field ($0) is 0 , and to sp2 if it’s 1

```
grunt> dump sp1;
(4,3,3)
(4,2,1)
grunt > dump sp2;
(8,4,3)
(8,3,4)
```