**Apache PIG Transformations**

**Input data used:**

* empdata.log

10101 | UnitedHealthGroup | 300000 | Hyderabad

10201 | InfosysTechLTD | 260000 | Bangalore

10301 | TechMahindraPvtLTD | 390000 | Chennai

10401 | CognizantTechServices | 420000 | Mumbai

10501 | DeloitteLLP | 310000 | Guwahati

* data1.log

100 A 209000

101 B 320000

101 C 330000

102 D 210003

100 E 170000

101 F 265000

104 G 370500

102 H 249030

103 I 350000

100 J 249000

* data2.log

101 X 240000

104 Y 239600

101 Z 259000

100 W 390200

103 V 259330

* tokenData.log

Hello this is Gayathri!

I am learning Hadoop at Kelly Technologies.

Apache PIG seems very interesting!

I am looking forward to get a certification in Hadoop.

This is a sample file to test the "TOKENIZE" word in PIG.

* marksData.log

219 Gayathri A

225 Keshari B

229 Karthik C

240 Manogna C

243 Mounika B

220 Ghousia C

211 Charishma C

218 Eshwar B

256 Pavani C

251 Navya A

253 Niteesh A

1. GRUNT SHELL IN LOCAL MODE:
2. JOIN:

LEFT Join:

pig -x local

A = LOAD 'data1.log' using PigStorage(‘\t') as (id:int, name:chararray, sal:int);

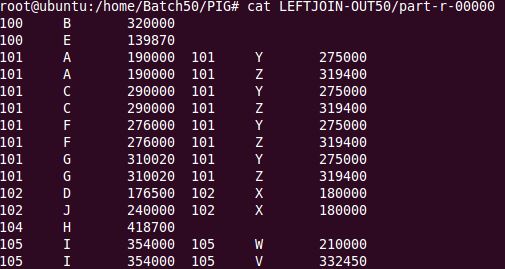
B = LOAD 'data2.log' using PigStorage('\t') as (Id: int, Name:chararray, Sal:int);

LJoin = JOIN A BY id LEFT, B BY Id;

STORE LJoin INTO 'LEFTJOIN-OUT50';

quit;

cat LEFTJOIN-OUT50/part-r-00000



INNER Join:

pig -x local

A = LOAD 'data1.log' using PigStorage('\t') as (id:int, name:chararray, sal:int);

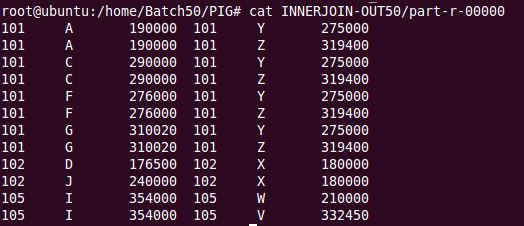
B = LOAD 'data2.log' using PigStorage('\t') as (Id: int, Name:chararray, Sal:int);

IJoin = JOIN A BY id, B BY Id;

STORE IJoin INTO 'INNERJOIN-OUT50';

quit;

cat INNERJOIN-OUT50/part-r-00000



RIGHT Join:

pig -x local

A = LOAD 'data1.log' using PigStorage('\t') as (id:int, name:chararray, sal:int);

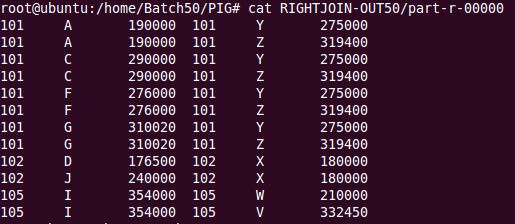
B = LOAD 'data2.log' using PigStorage('\t') as (Id: int, Name:chararray, Sal:int);

RJoin = JOIN A BY id RIGHT, B BY Id;

STORE RJoin INTO 'RIGHTJOIN-OUT50';

quit;

cat RIGHTJOIN-OUT50/part-r-00000



FULL Join:

Pig –x local

A = LOAD 'data1.log' using PigStorage('\t') as (id:int, name:chararray, sal:int);

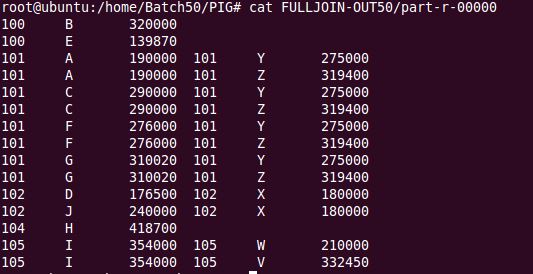
B = LOAD 'data2.log' using PigStorage('\t') as (Id: int, Name:chararray, Sal:int);

FJoin = JOIN A BY id FULL, B BY Id;

STORE FJoin INTO 'FULLJOIN-OUT50';

quit;

cat FULLJOIN-OUT50/part-r-00000

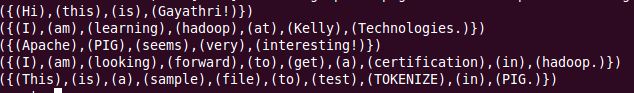


1. TOKENIZE:

A = LOAD 'tokenData.log' using PigStorage('\n') as (line:chararray);

B = FOREACH A GENERATE TOKENIZE(line);

DUMP B;



1. FLATTEN:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

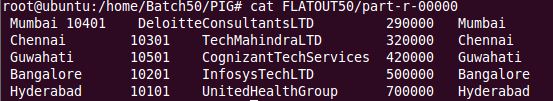
B = GROUP A BY cloc;

C = FOREACH B GENERATE group, FLATTEN(A);

STORE C INTO 'FLATOUT50';

quit;

cat FLATOUT50/part-r-00000



1. AGG FUNCTIONS

MAX:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

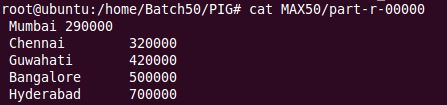
B = GROUP BY cloc;

C = FOREACH B GENERATE group, MAX(A.cheadcnt);

STORE C INTO 'MAX50';

quit;

cat MAX50/part-r-00000



MIN:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

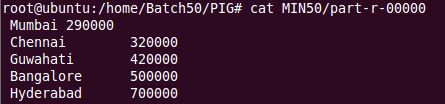
B = GROUP BY cloc;

C = FOREACH B GENERATE group, MIN(A.cheadcnt);

STORE C INTO MIN50;

quit;

cat MIN50/part-r-00000



AVG:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

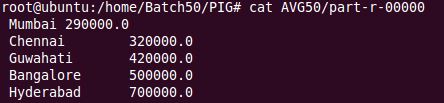
B = GROUP BY cloc;

C = FOREACH B GENERATE group, AVG(A.cheadcnt);

STORE C INTO AVG50;

quit;

cat AVG50/part-r-00000



SUM:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

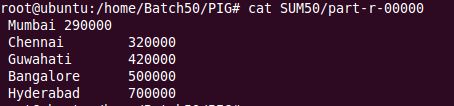
B = GROUP BY cloc;

C = FOREACH B GENERATE group, SUM(A.cheadcnt);

STORE C INTO 'SUM50';

quit;

cat SUM50/part-r-00000



COUNT:

pig –x local

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

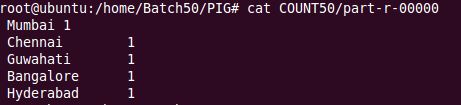
B = GROUP BY cloc;

C = FOREACH B GENERATE group, COUNT(A.cheadcnt);

STORE C INTO COUNT50;

quit;

cat COUNT50/part-r-00000



1. COGROUP:

pig –x local

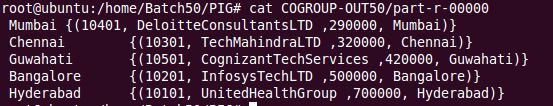
A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = COGROUP A BY cloc;

STORE B INTO 'COGROUP-OUT50';

quit;

cat COGROUP-OUT50/part-r-00000



1. DESCRIBE:

pig –x local

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

DESCRIBE A;

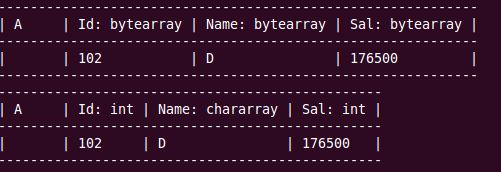
C:\Users\ASUS\Desktop\Assignment-Pig\HW\DescribeL.JPG

1. ILLUSTRATE:

pig –x local

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

ILLUSTRATE A;

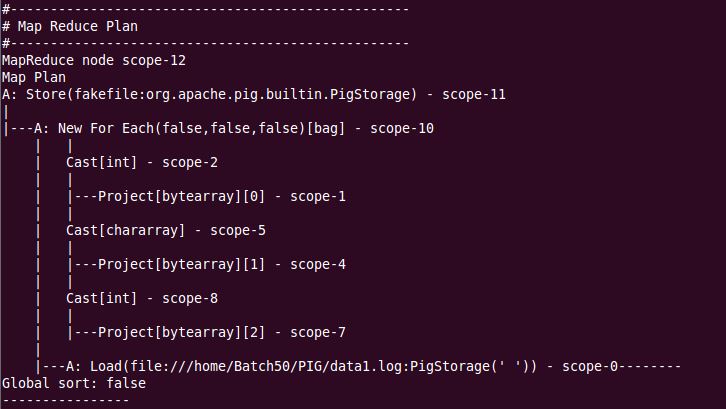


1. EXPLAIN:

pig –x local

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

EXPLAIN A;



1. DISTINCT:

pig –x local

A = LOAD 'marksData.log' using PigStorage('\t') as (ID:int, Names:chararray, Grades:chararray);

B = DISCTINCT A;

SPLIT B INTO C if Grades == 'A', D if Grades == 'B', E if Grades == 'C';

STORE C INTO 'AGrade';

STORE D INTO 'BGrade';

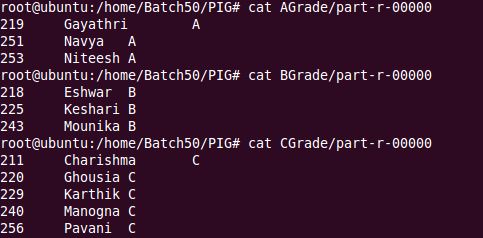
STORE E INTO 'CGrade';

quit;

cat AGrade/part-r-00000

cat BGrade/part-r-00000

cat CGrade/part-r-00000



1. CROSS:

data1 = LOAD 'data1.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

data2 = LOAD 'data2.log' using PigStorage('|') as (Cid: int, Cname:chararray, Cheadcnt:int, Cloc:chararray);

crsData = CROSS data1, data2;

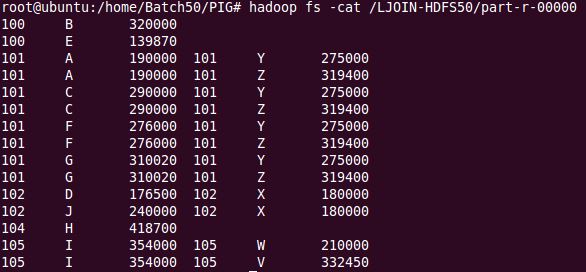
STORE crsData INTO 'CROSS-OUT50';

quit;

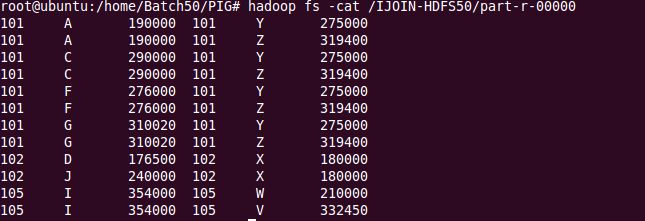
cat CROSS-OUT50/part-r-00000

1. GRUNT SHELL IN HDFS MODE:
2. JOIN:

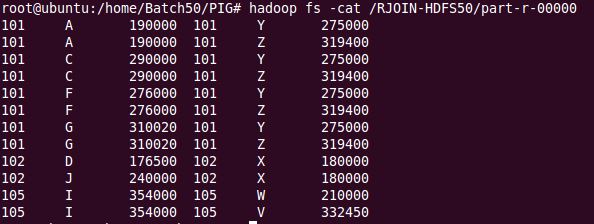
LEFT Join:



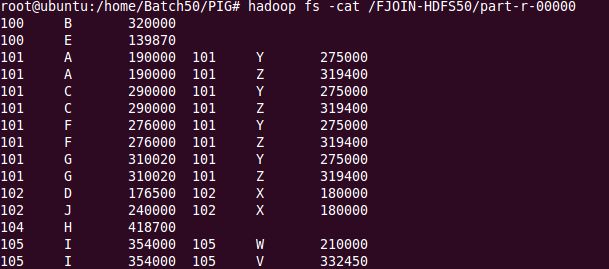
INNER Join:



RIGHT Join:



FULL Join:

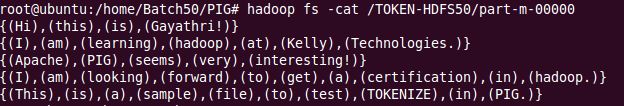


1. TOKENIZE:

A = LOAD '/pig50/tokenData.log' using PigStorage('\n') as (line:chararray);

B = FOREACH A GENERATE TOKENIZE(line);

DUMP B;



1. FLATTEN:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

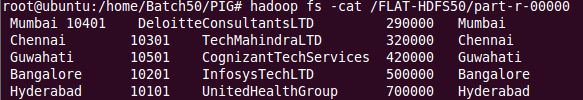
B = GROUP A BY cloc;

C = FOREACH B GENERATE group, FLATTEN(A);

STORE C INTO '/FLAT-HDFS50';

quit;

hadoop fs -cat FLAT-HDFS50/part-r-00000



1. AGG FUNCTIONS

MAX:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

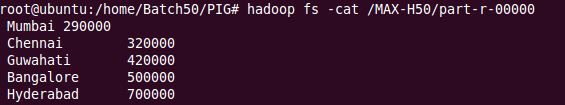
C = FOREACH B GENERATE group, MAX(A.cheadcnt);

STORE C INTO '/MAX-H50';

quit;

hadoop fs -ls /MAX-H50

hadoop fs -cat /MAX-H50/part-r-00000



MIN:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

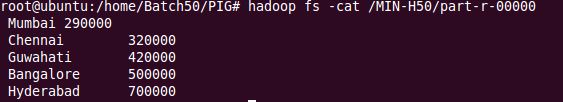
C = FOREACH B GENERATE group, MIN(A.cheadcnt);

STORE C INTO '/MIN-H50';

quit;

hadoop fs -ls /MIN-H50

hadoop fs -cat /MIN-H50/part-r-00000



AVG:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

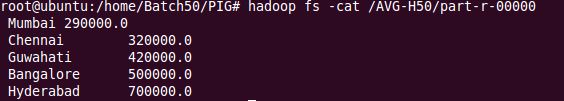
C = FOREACH B GENERATE group, AVG(A.cheadcnt);

STORE C INTO '/AVG-H50';

quit;

hadoop fs -ls /AVG-H50

hadoop fs -cat /AVG-H50/part-r-00000



SUM:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

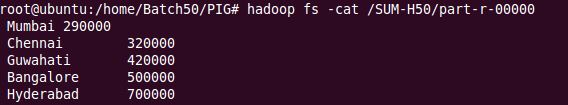
C = FOREACH B GENERATE group, SUM(A.cheadcnt);

STORE C INTO '/SUM-H50';

quit;

hadoop fs -ls /SUM-H50

hadoop fs -cat /SUM-H50/part-r-00000



COUNT:

pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

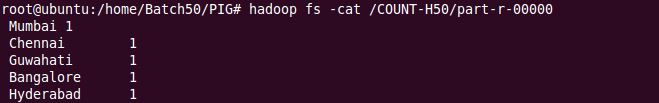
C = FOREACH B GENERATE group, COUNT(A.cheadcnt);

STORE C INTO '/COUNT-H50';

quit;

hadoop fs -ls /COUNT-H50

hadoop fs -cat /COUNT-H50/part-r-00000



1. COGROUP:

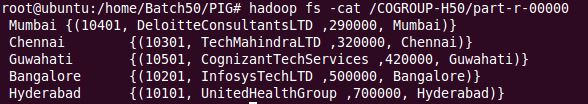
pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = COGROUP A BY cloc;

STORE B INTO '/COGROUP-H50;

Hadoop fs –cat /COGROUP-H50/part-r-00000



1. DESCRIBE:

pig

A = LOAD '/pig50/data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

DESCRIBE A;

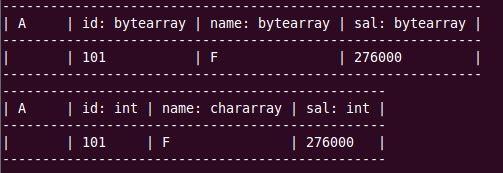
C:\Users\ASUS\Desktop\Assignment-Pig\HW\DescribeH.JPG

1. ILLUSTRATE:

pig

A = LOAD '/pig50/data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

ILLUSTRATE A;

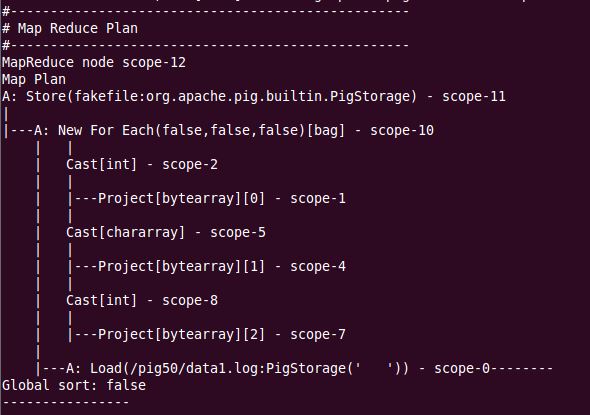


1. EXPLAIN:

pig

A = LOAD '/pig50/data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

EXPLAIN A;



1. DISTINCT:

pig

A = LOAD '/pig50/marksData.log' using PigStorage('\t') as (ID:int, Names:chararray, Grades:chararray);

B = DISCTINCT A;

SPLIT B INTO C if Grades == 'A', D if Grades == 'B', E if Grades == 'C';

STORE C INTO '/AGRADE-H50;

STORE D INTO '/BGRADE-H50';

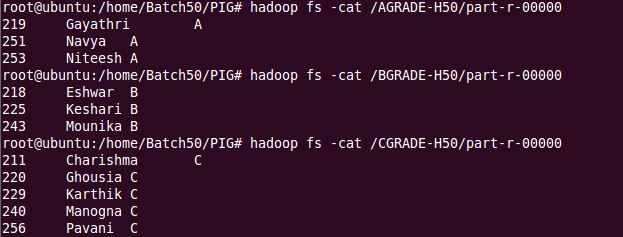
STORE E INTO '/CGRADE-H50';

quit;

hadoop fs –cat /AGRADE-H50/part-r-00000

hadoop fs –cat /BGRADE-H50/part-r-00000

hadoop fs –cat /CGRADE-H50/part-r-00000



1. CROSS:

data1 = LOAD '/pig50/data1.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

data2 = LOAD '/pig50/data2.log' using PigStorage('|') as (Cid: int, Cname:chararray, Cheadcnt:int, Cloc:chararray);

crsData = CROSS data1, data2;

STORE crsData INTO '/CROSS-OUT50';

hadoop fs -cat /CROSSH-OUT50/part-r-00000

hadoop fs -cat /CROSSH-OUT50/part-r-00000 wc -l

1. SCRIPT MODE IN LOCAL MODE:
2. TOKENIZE:

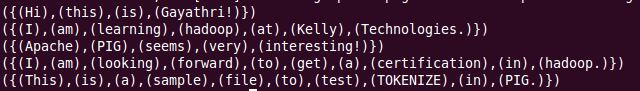
nano tokenData.pig

A = LOAD 'tokenData.log' using PigStorage('\n') as (line:chararray);

B = FOREACH A GENERATE TOKENIZE(line);

DUMP B;

pig –x local tokenData.pig



1. FLATTEN:

nano flattenScript.pig

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

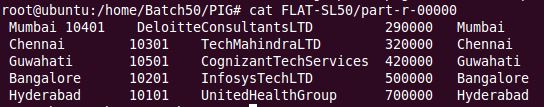
B = GROUP A BY cloc;

C = FOREACH B GENERATE group, FLATTEN(A);

STORE C INTO 'FLAT-SL50';

pig –x local flattenScript.pig

cat FLAT-SL50/part-r-00000



1. AGG FUNCTIONS:

nano aggScript.pig

A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

C = FOREACH B GENERATE group, MAX(A.cheadcnt);

STORE C INTO 'MAX-SL50';

D = FOREACH B GENERATE group, MIN(A.cheadcnt);

STORE D INTO 'MIN-SL50';

E = FOREACH B GENERATE group, AVG(A.cheadcnt);

STORE E INTO 'AVG-SL50';

F = FOREACH B GENERATE group, SUM(A.cheadcnt);

STORE F INTO 'SUM-SL50';

G = FOREACH B GENERATE group, COUNT(A.cheadcnt);

STORE G INTO 'COUNT-SL50';

pig –x local aggScript.pig

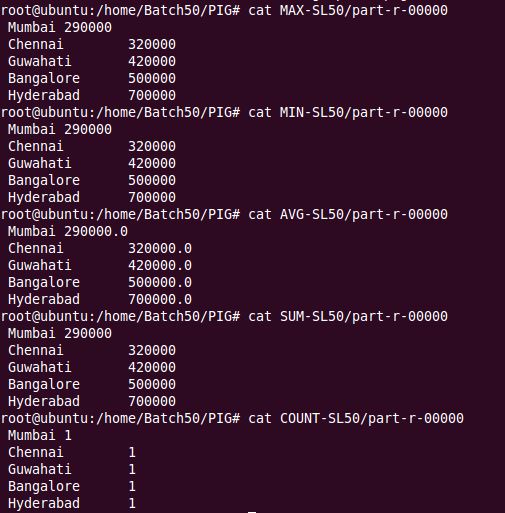
cat MAX-SL50/part-r-00000

cat MIN-SL50/part-r-00000

cat AVG-SL50/part-r-00000

cat SUM-SL50/part-r-00000

cat COUNT-SL50/part-r-00000



1. COGROUP:

nano flattenScript.pig

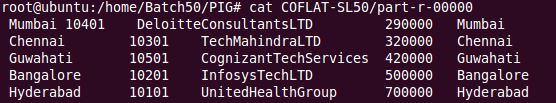
A = LOAD 'empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = COGROUP A BY cloc;

STORE B INTO 'COFLAT-SL50';

pig –x local flattenScript.pig

cat COFLAT-SL50/part-r-00000



1. DESCRIBE:

nano extraScript.pig

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

DESCRIBE A;

pig -x local extraScript.pig

C:\Users\ASUS\Desktop\Assignment-Pig\HW\DescribeL-SM.JPG

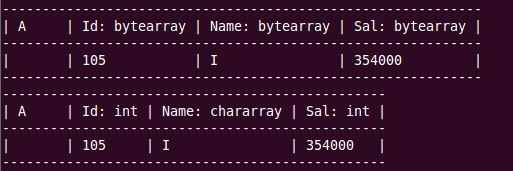
1. ILLUSTRATE:

nano extraScript.pig

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

ILLUSTRATE A;

pig -x local extraScript.pig



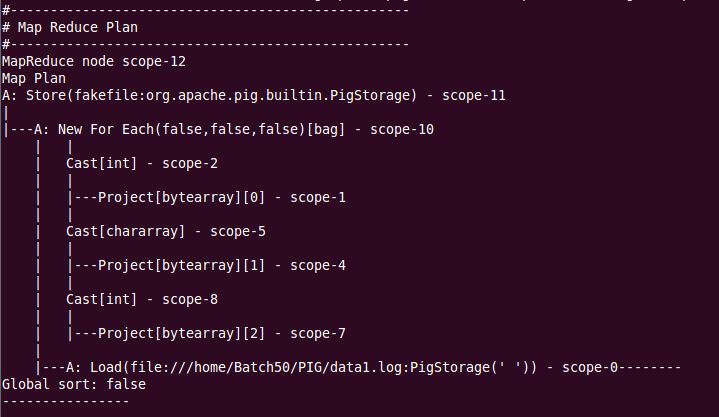
1. EXPLAIN:

nano extraScript.pig

A = LOAD 'data1.log' using PigStorage('\t') as (Id:int, Name:chararray, Sal:int);

EXPLAIN A;

pig -x local extraScript.pig



1. DISTINCT:

nano marksScript.pig

A = LOAD 'marksData.log' using PigStorage('\t') as (ID:int, Names:chararray, Grades:chararray);

B = DISCTINCT A;

SPLIT B INTO C if Grades == 'A', D if Grades == 'B', E if Grades == 'C';

STORE C INTO 'AGRADE-SL';

STORE D INTO 'BGRADE-SL';

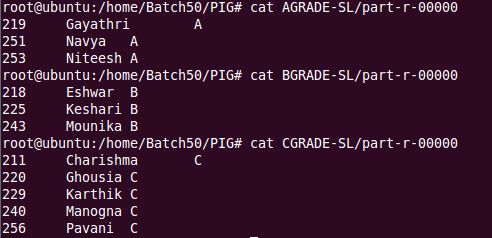
STORE E INTO 'CGRADE-SL';

pig –x local marksScript.pig

cat AGRADE-SL/part-r-00000

cat BGRADE-SL/part-r-00000

cat CGRADE-SL/part-r-00000



1. SCRIPT MODE IN HDFS MODE:
2. TOKENIZE:

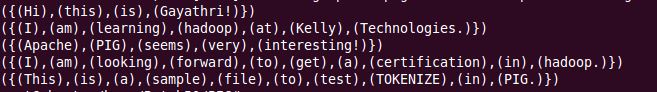
nano tokenData.pig

A = LOAD '/pig50/tokenData.log' using PigStorage('\n') as (line:chararray);

B = FOREACH A GENERATE TOKENIZE(line);

DUMP B;

pig tokenData.pig



1. FLATTEN:

nano flattenScript.pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

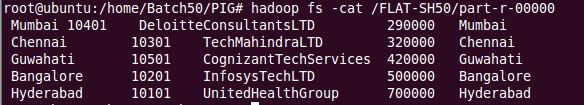
B = GROUP A BY cloc;

C = FOREACH B GENERATE group, FLATTEN(A);

STORE C INTO '/FLAT-SH50';

pig flattenScript.pig

hadoop fs -cat FLAT-SH50/part-r-00000



1. AGG FUNCTIONS:

nano aggScript.pig

A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = GROUP BY cloc;

C = FOREACH B GENERATE group, MAX(A.cheadcnt);

STORE C INTO '/MAX-SH50';

D = FOREACH B GENERATE group, MIN(A.cheadcnt);

STORE D INTO '/MIN-SH50';

E = FOREACH B GENERATE group, AVG(A.cheadcnt);

STORE E INTO '/AVG-SH50';

F = FOREACH B GENERATE group, SUM(A.cheadcnt);

STORE F INTO '/SUM-SH50';

G = FOREACH B GENERATE group, COUNT(A.cheadcnt);

STORE G INTO '/COUNT-SH50';

pig aggScript.pig

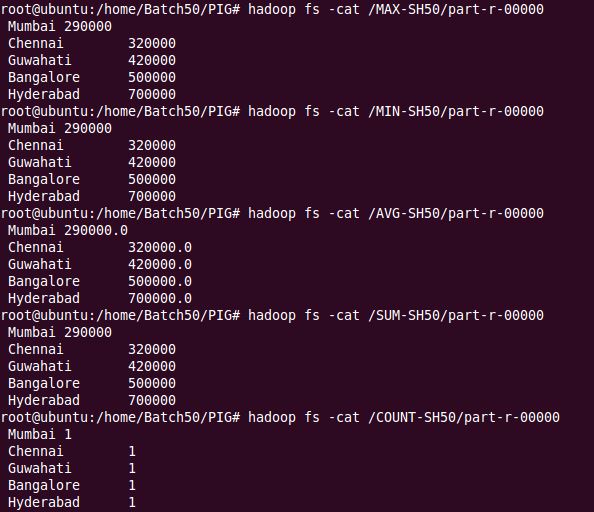
hadoop fs -cat MAX-SL50/part-r-00000

hadoop fs -cat MIN-SL50/part-r-00000

hadoop fs -cat AVG-SL50/part-r-00000

hadoop fs -cat SUM-SL50/part-r-00000

hadoop fs -cat COUNT-SL50/part-r-00000



1. COGROUP:

nano flattenScript.pig

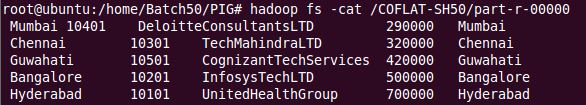
A = LOAD '/pig50/empdata.log' using PigStorage('|') as (cid: int, cname:chararray, cheadcnt:int, cloc:chararray);

B = COGROUP A BY cloc;

STORE B INTO '/COFLAT-SH50';

pig flattenScript.pig

hadoop fs -cat COFLAT-SH50/part-r-00000



1. DISTINCT:

nano marksScript.pig

A = LOAD '/pig50/marksData.log' using PigStorage('\t') as (ID:int, Names:chararray, Grades:chararray);

B = DISCTINCT A;

SPLIT B INTO C if Grades == 'A', D if Grades == 'B', E if Grades == 'C';

STORE C INTO '/AGRADE-SH';

STORE D INTO '/BGRADE-SH';

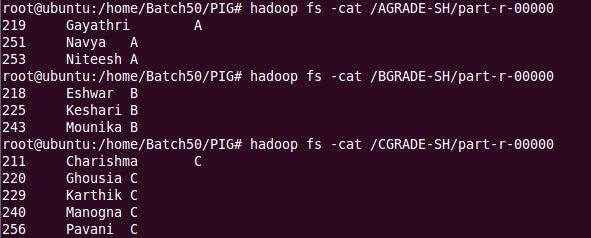
STORE E INTO '/CGRADE-SH';

pig marksScript.pig

hadoop fs -cat AGRADE-SH/part-r-00000

hadoop fs -cat BGRADE-SH/part-r-00000

hadoop fs -cat CGRADE-SH/part-r-00000



1. LIMIT, DISTINCT, FILTER, ORDER in SCRIPT MODE:

pig –x local

iniData = LOAD ‘empdata.log’ using PigStorage(‘|’) as (<cid:int>, cname:chararray, cheadcnt:chararray, cloc:int);

filData = FILTER iniData BY cloc == ‘Hyderabad’;

ordData = ORDER filData BY cid;

disData = DISTINCT iniData;

limData = LIMIT iniData 3;

STORE filData INTO ‘FILTER-OUT50’;

STORE ordData INTO ‘ORD-OUT50’;

STORE disData INTO ‘DISDATA50’;

STORE limData INTO ‘LIMDATA50’;

cat DISDATA50/part-r-00000

cat LIMITDATA50/part-r-00000

