Queries solving using Pig

File :cust file

we have create schema of a relation by using AS and defining its fields

----------->Each pig script consists of a relation and an operation

------------>left side --->relation right side------->operation and ;(for termination)

cust1 = Load '/home/hduser/custs.txt' using PigStorage(',') as (id:int,name,comm,age,profession);

dump cust1;

---------->viewing the schema of a relation with its fields

describe cust1;

cust1:{id:int,name:bytearray,comm:bytearray,age:bytearray,profession:bytearray}

------>for first top 10 records

limit1 = limit cust1 10;

dump limit1

(4000001,Kristina,Chung,55,Pilot)

(4000002,Paige,Chen,74,Teacher)

(4000003,Sherri,Melton,34,Firefighter)

(4000004,Gretchen,Hill,66,Computer hardware engineer)

(4000005,Karen,Puckett,74,Lawyer)

(4000006,Patrick,Song,42,Veterinarian)

(4000007,Elsie,Hamilton,43,Pilot)

(4000008,Hazel,Bender,63,Carpenter)

(4000009,Malcolm,Wagner,39,Artist)

(4000010,Dolores,McLaughlin,60,Writer)

describe limit1

limit1: {id: int,name: bytearray,comm: bytearray,age: bytearray,profession: bytearray}

-----------🡪find age of each profession

group1 = group limit1 by profession;

dump group1;

(Pilot,{(4000007,Elsie,Hamilton,43,Pilot),(4000001,Kristina,Chung,55,Pilot)})

(Artist,{(4000009,Malcolm,Wagner,39,Artist)})

(Lawyer,{(4000005,Karen,Puckett,74,Lawyer)})

(Writer,{(4000010,Dolores,McLaughlin,60,Writer)})

(Teacher,{(4000002,Paige,Chen,74,Teacher)})

(Carpenter,{(4000008,Hazel,Bender,63,Carpenter)})

(Firefighter,{(4000003,Sherri,Melton,34,Firefighter)})

(Veterinarian,{(4000006,Patrick,Song,42,Veterinarian)})

(Computer hardware engineer,{(4000004,Gretchen,Hill,66,Computer hardware engineer)})

describe group1;

group1 :{group:bytearray,limit1:{(id: int,name: bytearray,comm: bytearray,age: bytearray,profession: bytearray)}}

for1 = foreach group1 generate group , limit1.age;

dump for1;

(Pilot,{(43),(55)})

(Artist,{(39)})

(Lawyer,{(74)})

(Writer,{(60)})

(Teacher,{(74)})

(Carpenter,{(63)})

(Firefighter,{(34)})

(Veterinarian,{(42)})

(Computer hardware engineer,{(66)})

describe for1;

for1: { group:bytearray,{(age:bytearray)}}

|  |
| --- |
| JOIN CUSTOMER AND TRANSACTION |
|  | ............................. |
|  | a7 = join a6 by $0,a1 by $0; |
|  |  |
|  | a8 = foreach a7 generate $0,$1,$3,$4,$5,$6;---here $--represents the previous data--i.e --a7 |
|  |  |
|  |  |
|  |  |
|  | Total Sales |
|  | ................................................................... |
|  |  |
|  | a2 = load 'transa.txt' using PigStorage(',') as (id,dt:chararray,custid,amt:double,m1:chararray,m2:chararray,m3:chararray,m4:chararray,mode:chararray); |
|  |  |
|  | a3 = group a2 all; |
|  | a4 = foreach a3 generate SUM(a2.amt); |

ques 2

book1 = load '/home/hduser/book1' using PigStorage() as (name:chararray);

book2 = load '/home/hduser/book2' using PigStorage() as (name:chararray);

bookcombined = union book1,book2;

grunt> dump bookcombined;

(this is a sentence one)

(this is a sentence four)

(this is a sentence two)

(this is a sentence five)

(this is a sentence three)

(this is a sentence six)

(this is a sentence eight)

(this is a sentence seven)

split bookcombined into book3 if SUBSTRING(name,5,7) == 'is',book4 if SUBSTRING(name,19,24) == 'three';

ques 3-------------🡪 log files

log1 = load '/home/hduser/hadoop-hduser-secondarynamenode-ubuntu.log' using PigStorage() as (lines:chararray);

split log1 into log2 if SUBSTRING(lines,24,28) =='INFO',log3 if SUBSTRING(lines,24,29) =='ERROR',log4 if SUBSTRING(lines,24,28) =='WARN',

dump log2;

dump log3;

dump log4;

grunt> groupalllog2 = group log2 ALL;

grunt> DUMP groupalllog2;

groupalllog3 = group log3 ALL;

DUMP groupalllog3

groupalllog4 = group log4 ALL;

USER DEFINED FUNCTIONS UDF

-----------------------------------------------------------------------------------

nano pigdata

lower case char

lower case sentence

----------------------------------------------------

1)lowecase to uppercase

---------------------------

start-all.sh

export pig jar files (3) pig-0.13-h1.jar pig-0.13-withouthadoop-h1.jar pig-0.13-withouthadoop-h2.jar

adding jar/registering a jar

REGISTER /home/hduser/pigUDF.jar;

DEFINE ConverLowerToUpper myudfs.UPPER();

bag1 =load /home/hduser/pigdata' using PigStorage() as (name:chararray);

bag2 = foreach bag1 generate CovertLowerToUpper(name);

------------------------------------------------------------------------------------------