**Homework #7**

**CINS 370 Introduction to Databases**

**2. Question.**

**Employee table creation and transfer in hadoop,**

Vim empt.txt

Cat emp.txt

Hadoop fs -put emp.txt emp.txt

A picture containing clock, meter

Description automatically generated

**Department table creation and transfer in hadoop**

Vim dept.txt

Cat dep.txt

Hadoop fs -put dept.txt dept.txt

A picture containing clock

Description automatically generated

Grunt> cat emp.txt

Grunt> cat dept.txt

A picture containing clock

Description automatically generated

**Loading employee into E;**

E = Load ‘emp.txt’ using pigstorage(‘,’) as (EmpID:int,Ename:Chararray,Salary:int, DeptID:int)

Describe E;

Dump E;



A picture containing clock, object, green, street

Description automatically generated

**Loading Department into D;**

D = Load ‘dept.txt’ using pigstorage(‘,’) as (DepID:int,Dname:Chararray,Location:Chararray)

Describe D;

Dump D;



A clock on each of it s sides

Description automatically generated

**Left Outer Join on E and D via Depid**

P = Join E by DepId Left outer, D by DeptId

Describe P;

Dump P;



A large clock mounted to the side

Description automatically generated

**Vertical cut on P via Ename, Salary, Dname**

O = Foreach P generate Ename, Salary, Dname;

Describe O;

Dump O;

A large clock mounted to the side

Description automatically generated

A picture containing object, clock, sitting, black

Description automatically generated

**Order by on O via Salary in Ascending**

Ot = Order O by Salary ASC;

Dump Ot;



A clock on each of it s sides

Description automatically generated

**3.Question**

**Creating Count file**

Vim Count.txt

Hadoop fs -put Count.txt Count.txt

Pig

Grunt>ls

**A picture containing clock

Description automatically generated**

Grunt> cat Count.txt

A picture containing object, clock, black, sitting

Description automatically generated

**Loading Count.txt into A**

A = Load ‘Count.txt’ using pigstorage(‘\t’) as (count:Chararray);

Dump A;





**Applying Foreach and Tokenizing count by word**

C = Foreach A generate flatten(Tokenize(count)) as word;

Dump C;



A picture containing object, clock

Description automatically generated

**Grouping C by Word and Generating count**

S = Group C by word;

Final = Foeach S generate group, Count(C);

Dump Final;



A picture containing object, clock

Description automatically generated

**4. Question**

**Uploading Poverty.csv and State.csv**

hadoop fs -put Poverty.csv P.txt

hadoop fs -put State.csv S.txt

Pig

Grunt>ls

A picture containing sitting, black

Description automatically generated

A close up of a sign

Description automatically generated

**Overview of Poverty table**

A screenshot of a cell phone

Description automatically generated

**Overview of State table**

A picture containing sitting, green, phone

Description automatically generated

**Loading Poverty table in to A;**

A = Load ‘P.txt’ using pigstorage(‘,’) as (State:Chararray,LowerPeople:int, Upperpeople:int, LowerChildren:int, UpperChildren:int)

Dump A;



A picture containing green

Description automatically generated

**Loading State table in to B;**

A = Load ‘S.txt’ using pigstorage(‘,’) as (Country:Chararray,State:Chararray,Population:int)

Dump B;

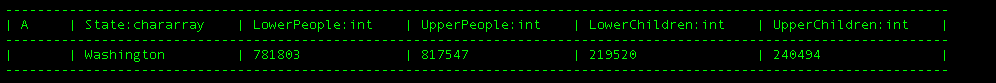


A screenshot of a cell phone

Description automatically generated

**Preview of Poverty Table**

Illustrate A;



**Preview of State Table**

Illustrate B;

A close up of a computer

Description automatically generated

**Applying Left Outer Join on Both the table using State**

J = JOIN A by State left outer, B by state;



A picture containing green

Description automatically generated

**Applying Foreach on Join using State, Population, LowerChildren.**

G = Foreach J generate State,Population,LowerChildren;

Describe G;

Dump G;

A picture containing clock, ball, object, green

Description automatically generated

A circuit board

Description automatically generated

**Applying Limit on G of 50**

L = Limit G 50;

DUMP L;



A picture containing green, circuit

Description automatically generated

**Applying Order by on Obtained Limited L**

O = Order L by State ASC;

Describe O;

Dump O;

A picture containing clock, ball, object, green

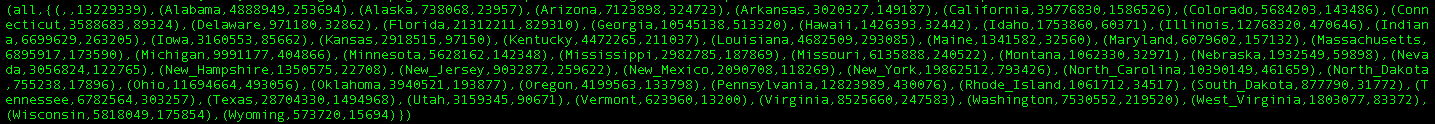
Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Applying Group ALL**





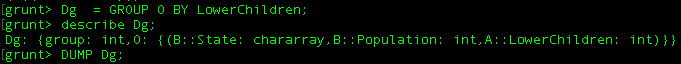
**Applying Group By State**



A picture containing circuit

Description automatically generated

**Applying Group By LowerChildren**



A green and black text

Description automatically generated