Pig - Session 4 Assignment 2

1. **CONCAT:**

Command used to create Database from the dataset in the local system –

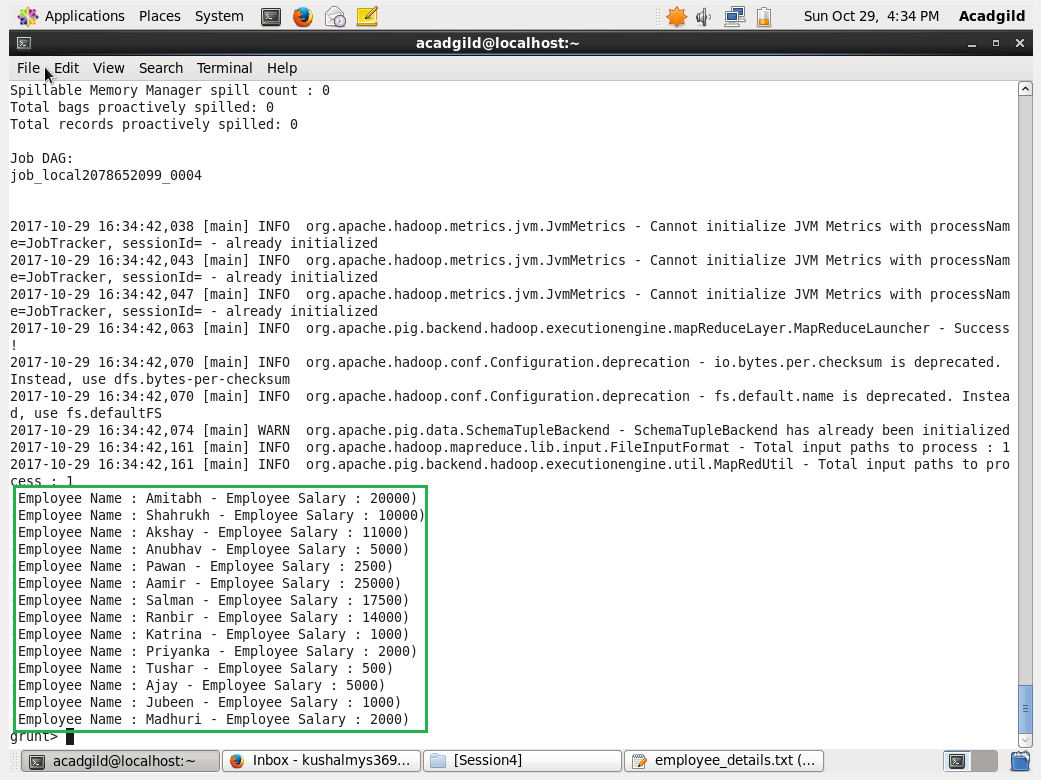
* empSchm = LOAD '/home/acadgild/Kushal/Session4/employee\_details.txt' USING PigStorage(',') AS (emp\_id:int, emp\_Name:chararray, emp\_salary:int);

To display the created schema -

* dump empSchm;

To Demonstrate CONCAT –

* empConcat = FOREACH empSchm GENERATE CONCAT('Employee Name : ', emp\_Name, ' - ', 'Employee Salary : ', (chararray)emp\_salary);



Boxed is the Output of CONCAT.

1. **TOKENIZE:**

Command used to create Database from the dataset in the local system –

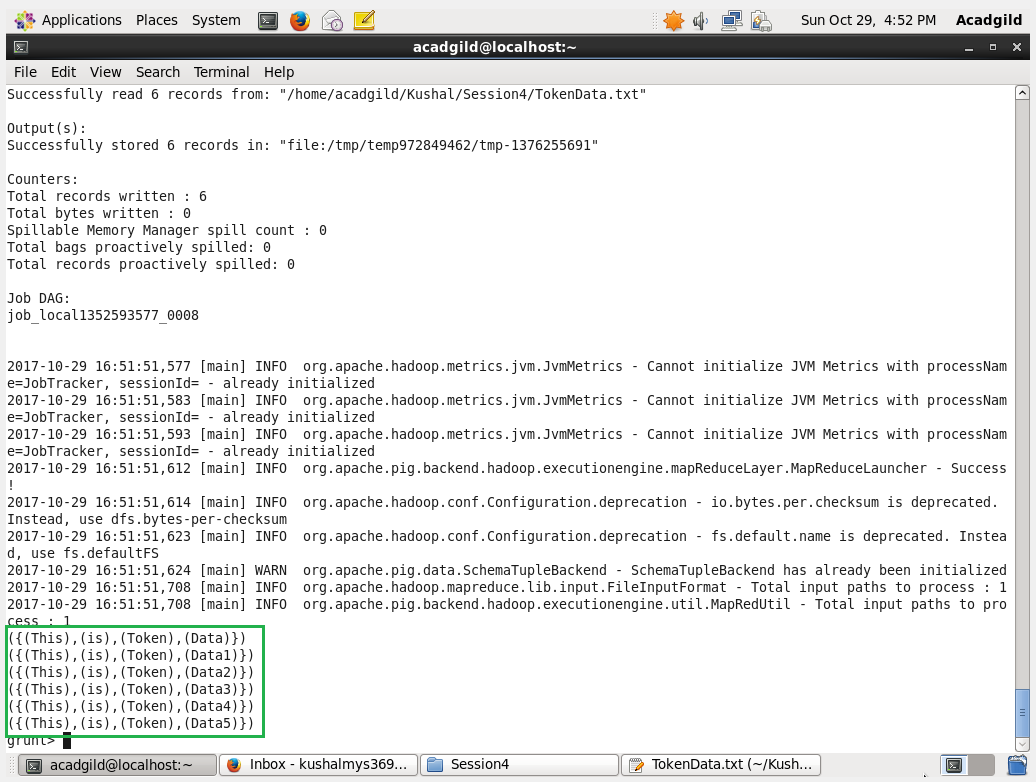
* tokenSchm = LOAD '/home/acadgild/Kushal/Session4/TokenData.txt' AS (field:chararray);

To display the created schema -

* dump tokenSchm;

To Demonstrate CONCAT –

* tokenizedData = FOREACH tokenSchm GENERATE TOKENIZE(field);



Boxed is the Output of TOKONIZE.

1. **SUM:**

To do SUM we need to first Group the similar data together Command –

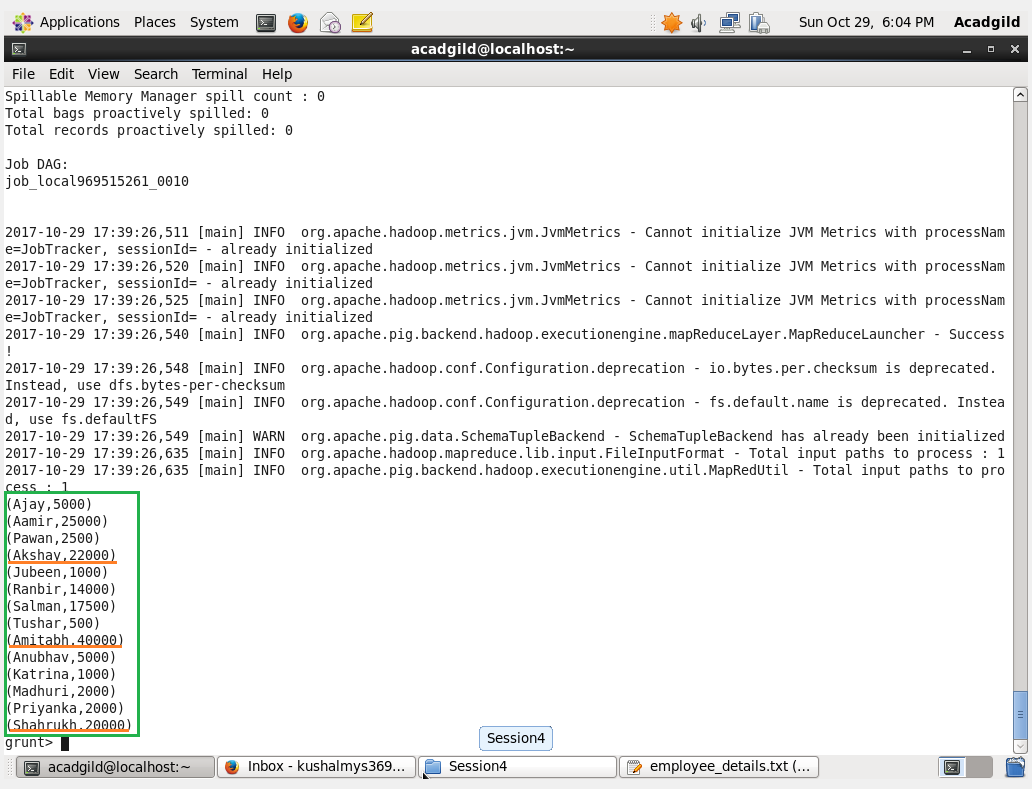
* groupEmpSchm = GROUP empSchm BY emp\_Name;

To display the created schema -

* dump groupEmpSchm;

To Demonstrate SUM –

* empSUM = FOREACH groupEmpSchm GENERATE group SUM (empSchm.emp\_ salary);



Boxed is the Output and the underlined Data are the result of the SUM.

1. **MIN:**

To do MIN we need to first Group the similar data together Command and works on bag of data–

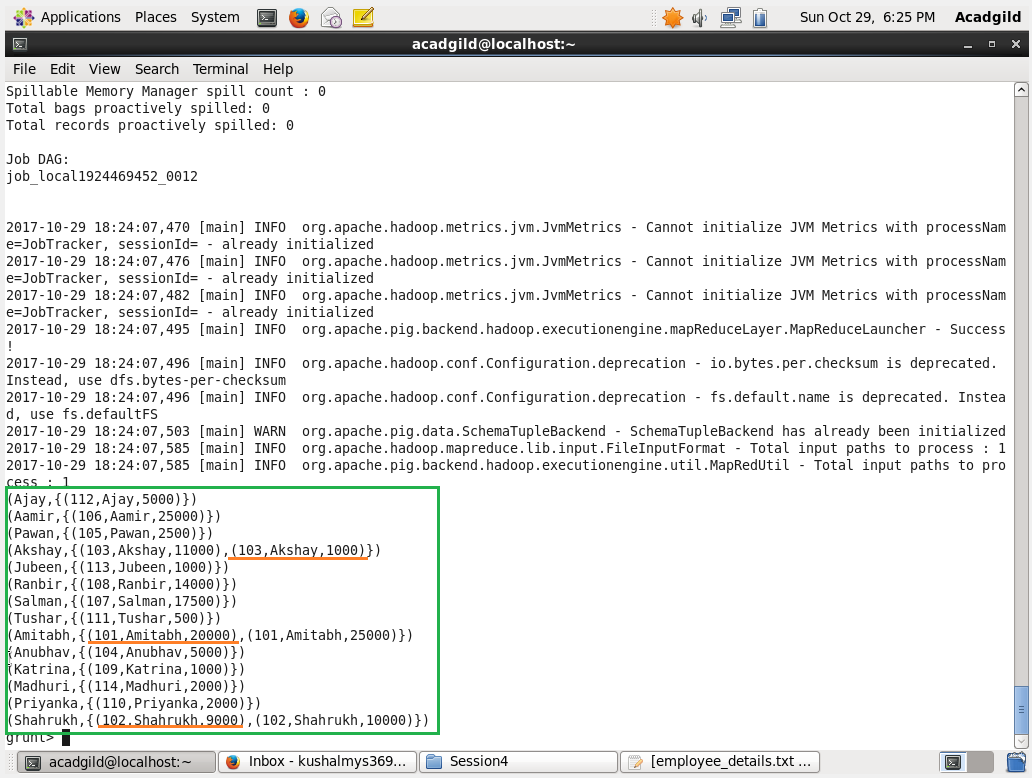
* groupEmpSchm = GROUP empSchm BY emp\_Name;

To display the created schema -

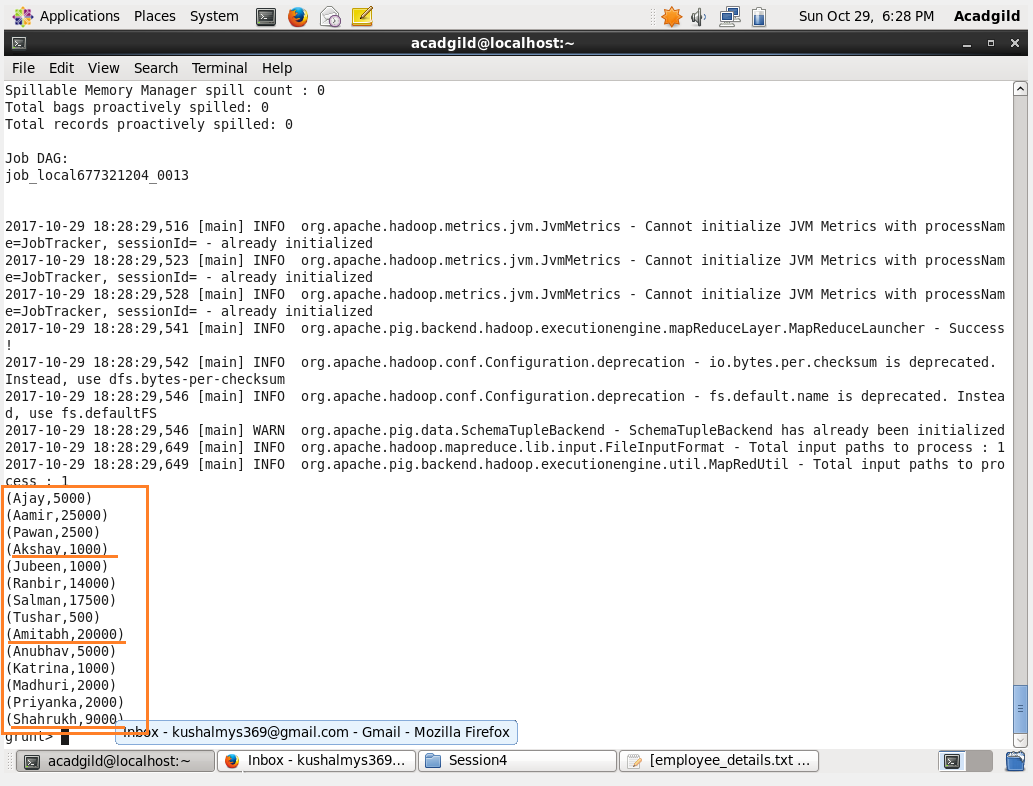
* dump groupEmpSchm;

To Demonstrate MIN –

* empMIN = FOREACH groupEmpSchm GENERATE group, MIN(empSchm.emp\_ salary);



Boxed has the Data grouped by Name with different Salary.



Boxed with Underlined has the min salary value in the Group using MIN.

1. **MAX:**

To do MAX we need to first Group the similar data together Command and works on bag of data–

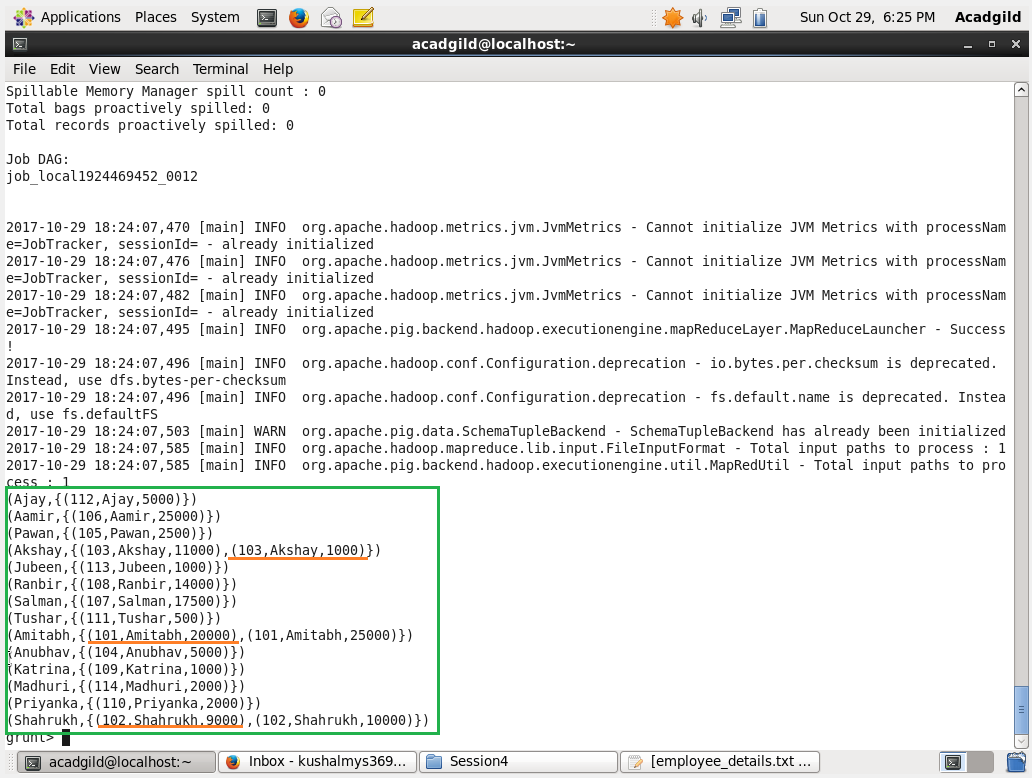
* groupEmpSchm = GROUP empSchm BY emp\_Name;

To display the created schema -

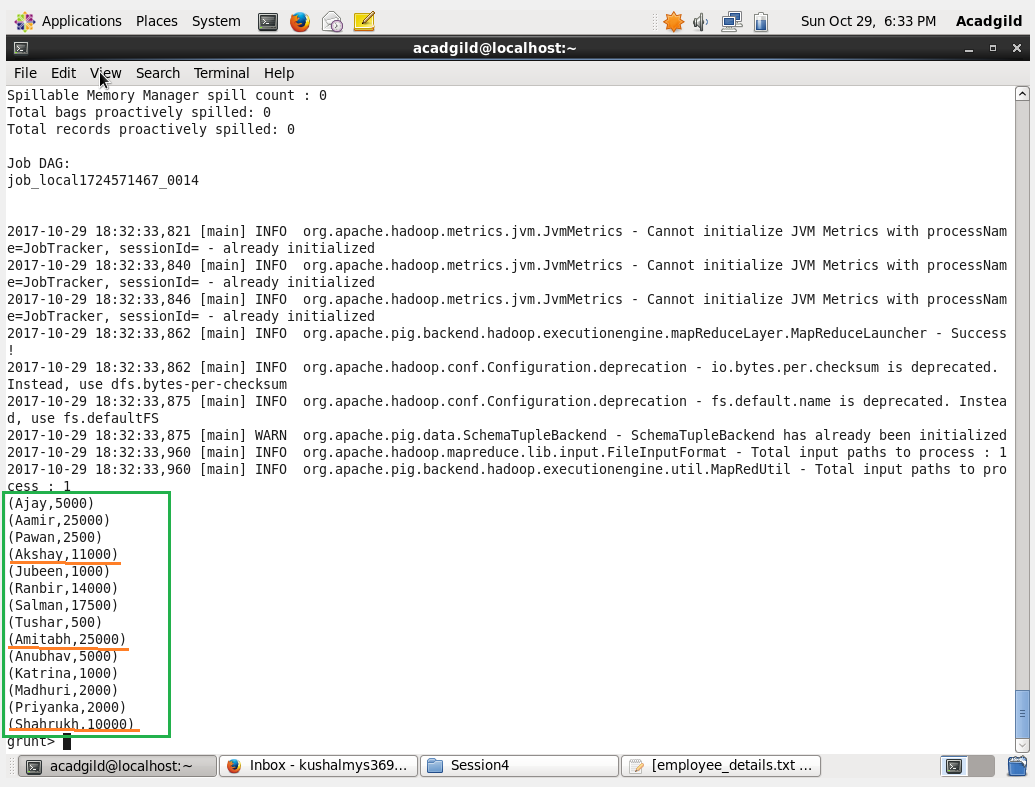
* dump groupEmpSchm;

To Demonstrate MAX –

* empMAX = FOREACH groupEmpSchm GENERATE group, MAX(empSchm.emp\_ salary);



Boxed has the Data grouped by Name with different Salary.

****

Boxed with Underlined has the max salary value in the Group using MAX.

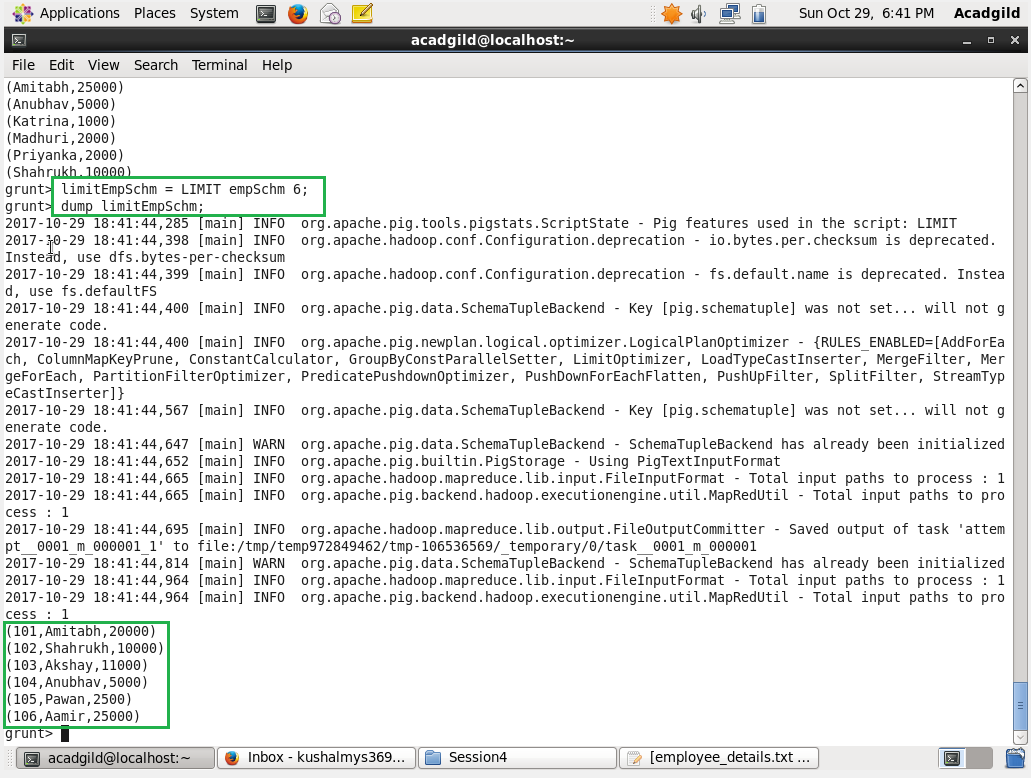
1. **LIMIT:**

Command to LIMIT –

* limitEmpSchm = LIMIT empSchm 6;

To display the result schema of LIMIT-

* dump limitEmpSchm;



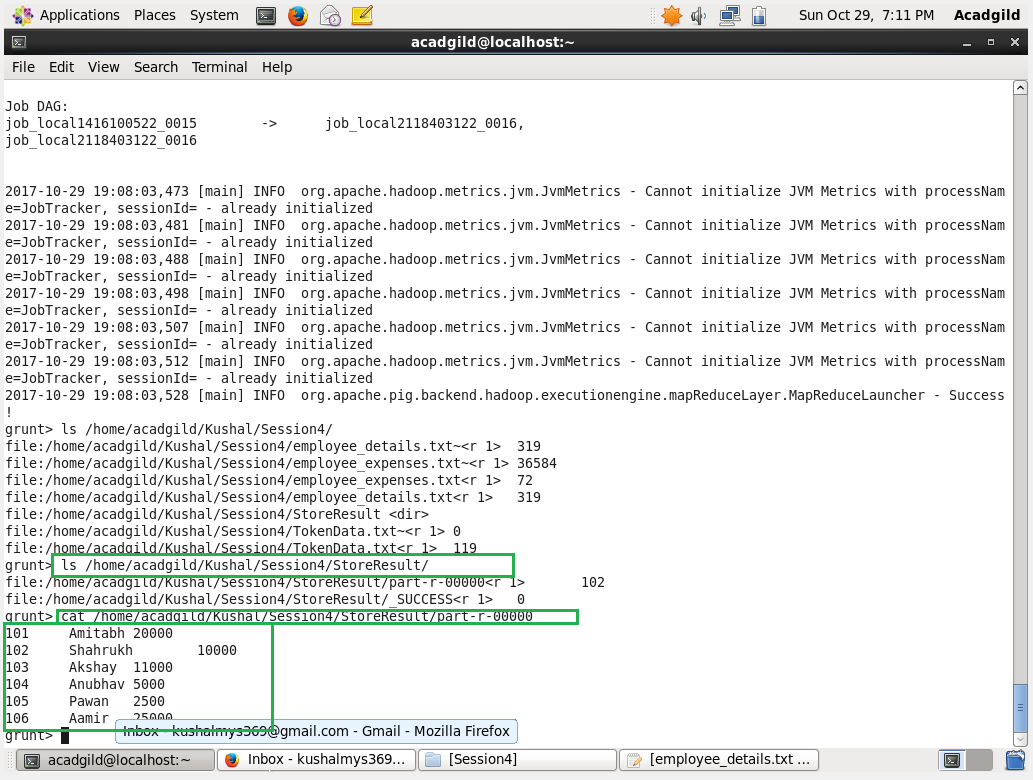
First boxed contains the LIMIT command.

Second boxed contains the LIMIT data.

1. **STORE:**

Command to STORE –

* store limitEmpSchm into '/home/acadgild/Kushal/Session4/StoreResult/';

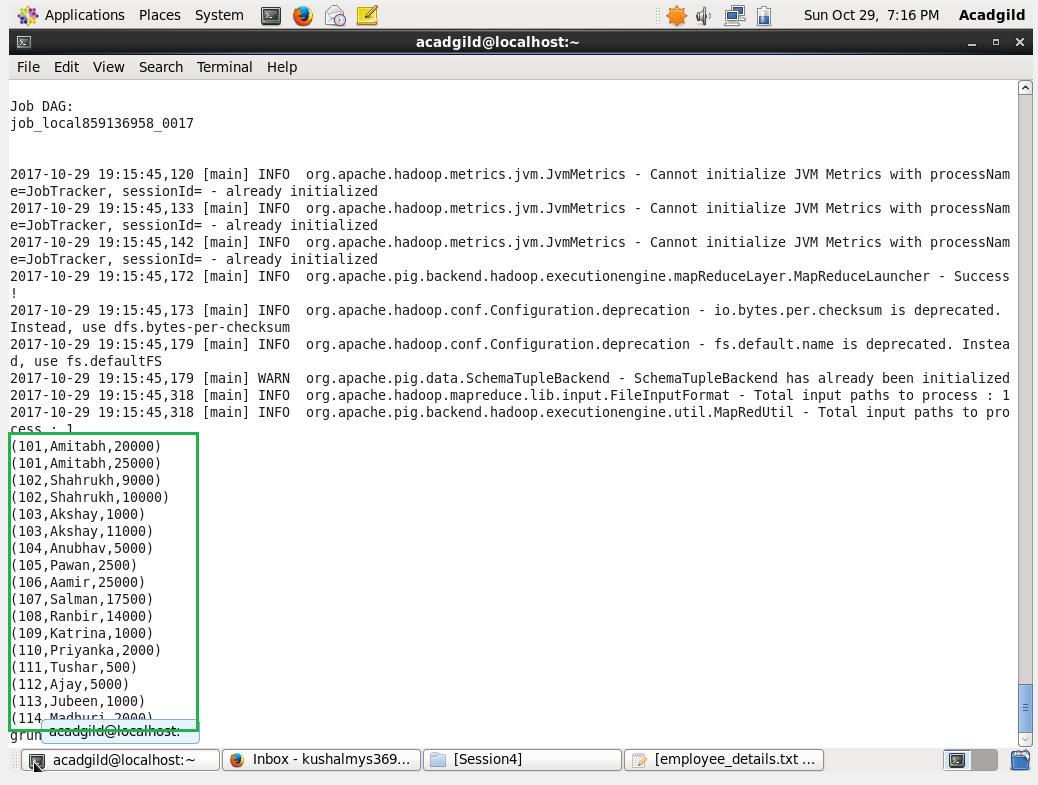


Boxed Contains the directory created with data as result from the Command STORE.

1. **DISTINCT:**

Command to DISTINCT –

* distinctEmpSchm = DISTINCT empSchm;

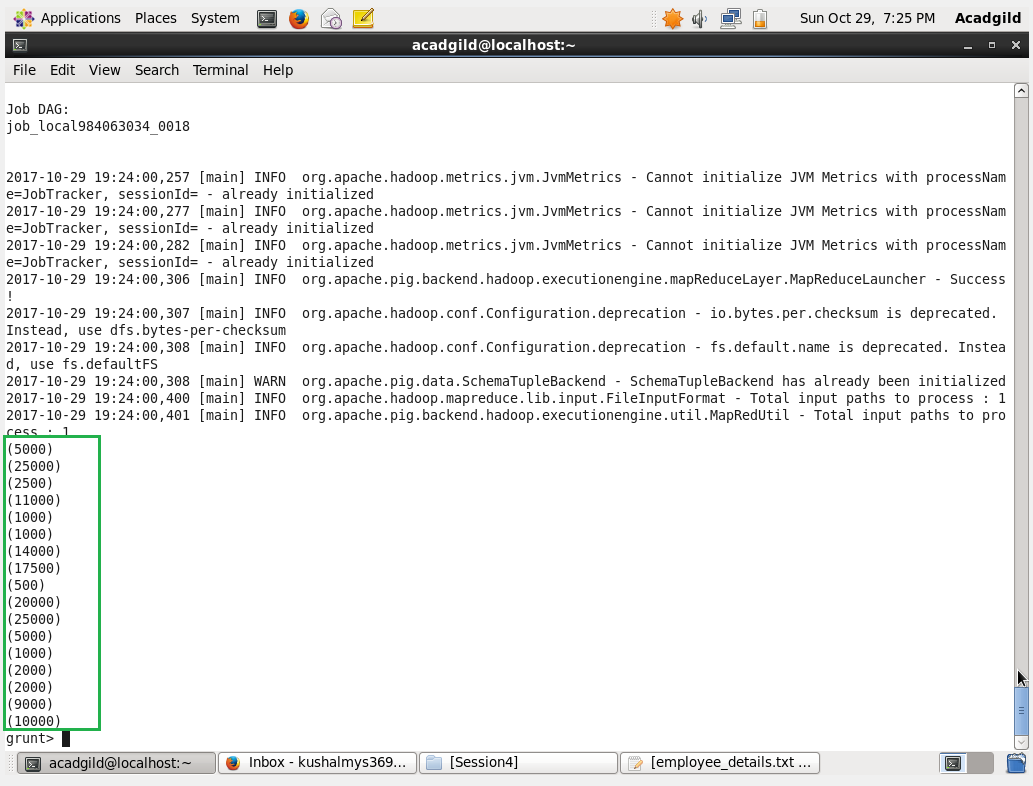


Boxed Data contains the distinct Result of data using DISTINCT.

1. **FLATTEN:**

Command to FLATTEN –

* flatEmpSchm = FOREACH groupEmpSchm GENERATE FLATTEN(empSchm.emp\_salary);



Boxed contains the result of the FLATTEN.

1. **ISEMPTY:**

Command to Load Second data set –

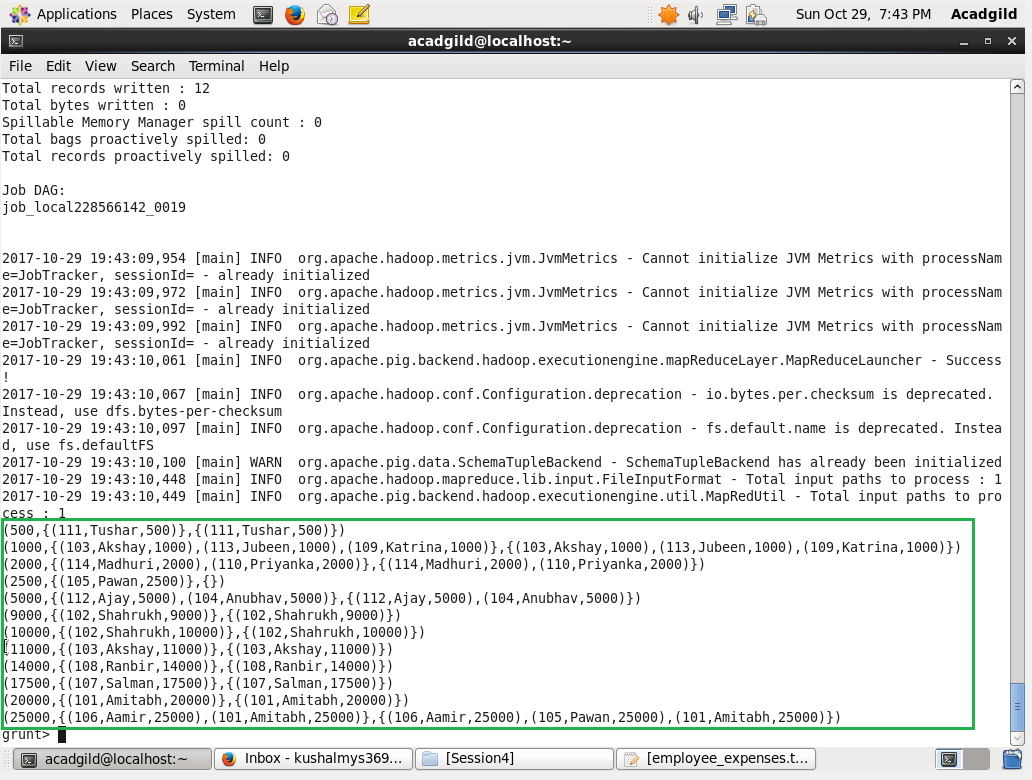
* empSchmExt = LOAD '/home/acadgild/Kushal/Session4/employee\_details\_ext.txt' USING PigStorage(',') AS (emp\_id:int, emp\_Name:chararray, emp\_salary:int);

Command to Combine two data set schema –

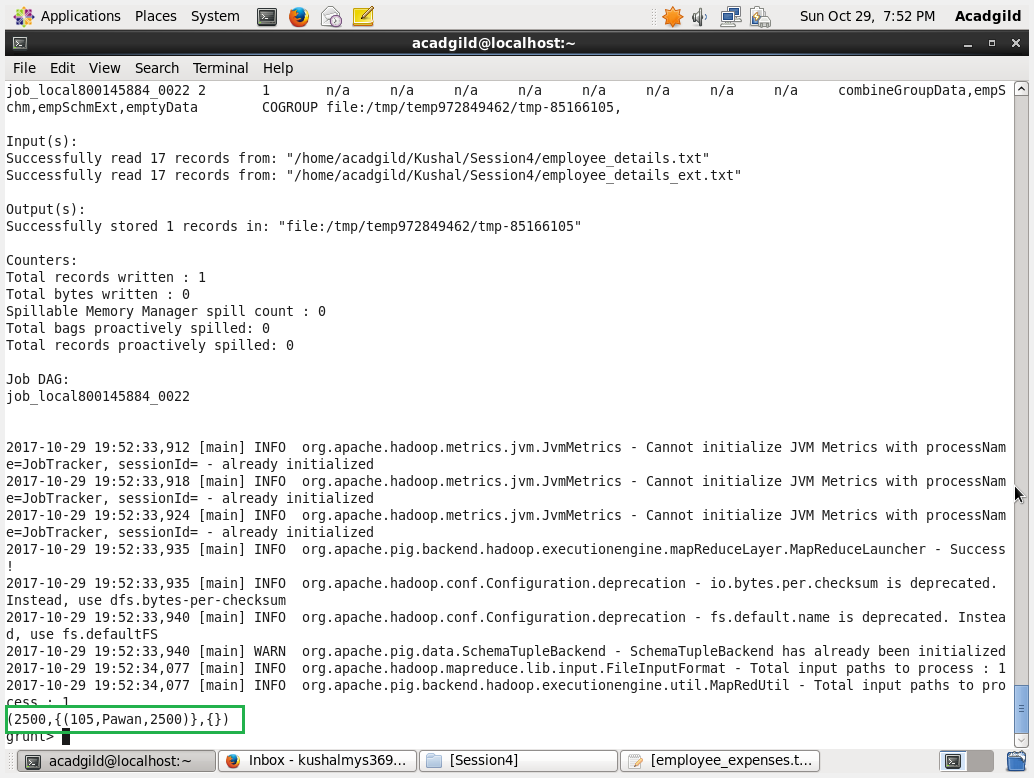
* combineGroupData = COGROUP empSchm by emp\_salary, empSchmExt by emp\_salary;

To Demonstrate IsEmpty –

* emptyData = FILTER combineGroupData by IsEmpty(empSchm);



Boxed Contains the CoGrouped Data of Two data set.



Boxed Contains the Data which satisfies the IsEmpty.