First Load the employee\_details.txt into a PIG tuples have schema

employee\_id:int

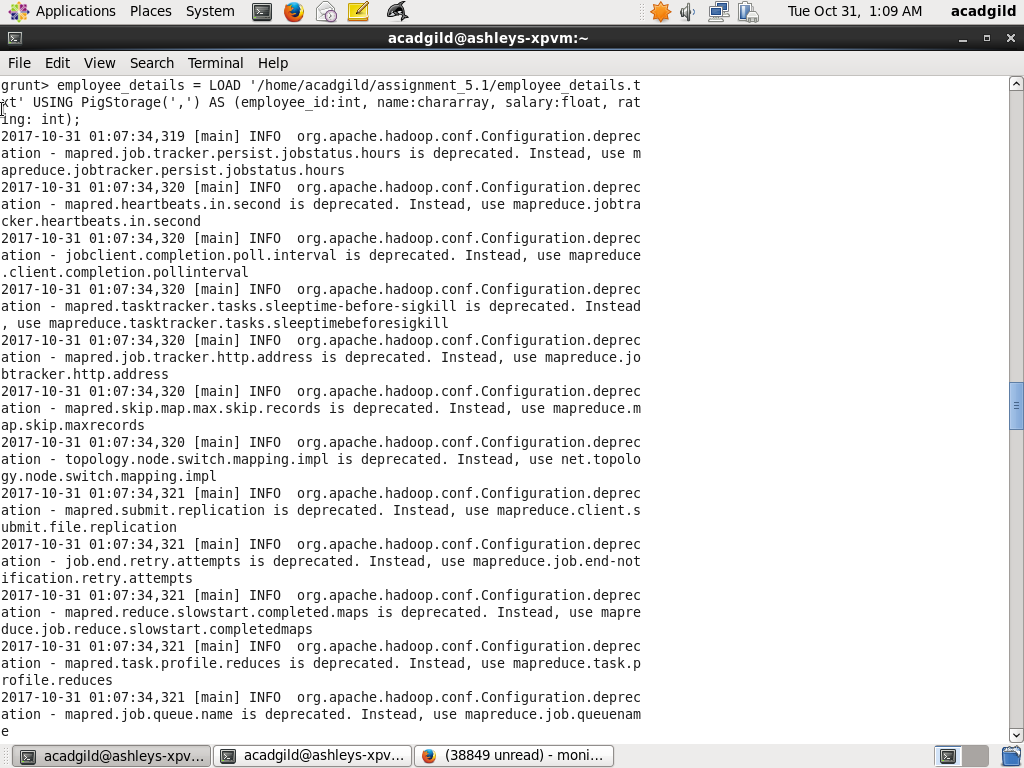
name: chararray

salary: float

rating: int

Using the command as below:

employee\_details = LOAD '/home/acadgild/assignment\_5.1/employee\_details.txt' USING PigStorage(',') AS (employee\_id:int, name:chararray, salary:float, rating: int);



Next load employee\_expenses into tuples having schema:

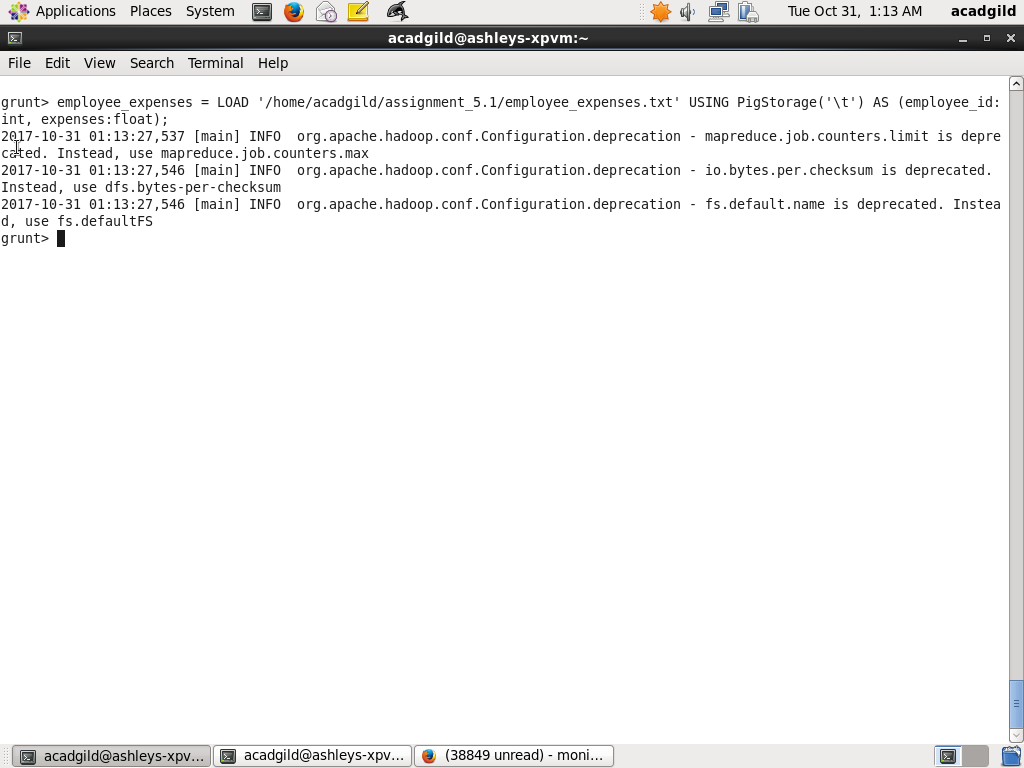
employee\_id: int

expense:float

Using the command:

employee\_expenses = LOAD '/home/acadgild/assignment\_5.1/employee\_expenses.txt' USING PigStorage('\t') AS (employee\_id:int, expenses:float);

The screenshot is shown below:



Task a. Top 5 employees with highest rating.

The following PIG script will first sort employees by rating, next select columns employee\_id,name.. Next first 5 are taken using LIMIT 5 statement

Step1:

will first sort employee\_details by rating and form the relation employee\_sort\_by\_rating

employee\_sort\_by\_rating = ORDER employee\_details BY rating, name;

Step2:

Select only fields employee\_id, name and form the relation rated\_employess

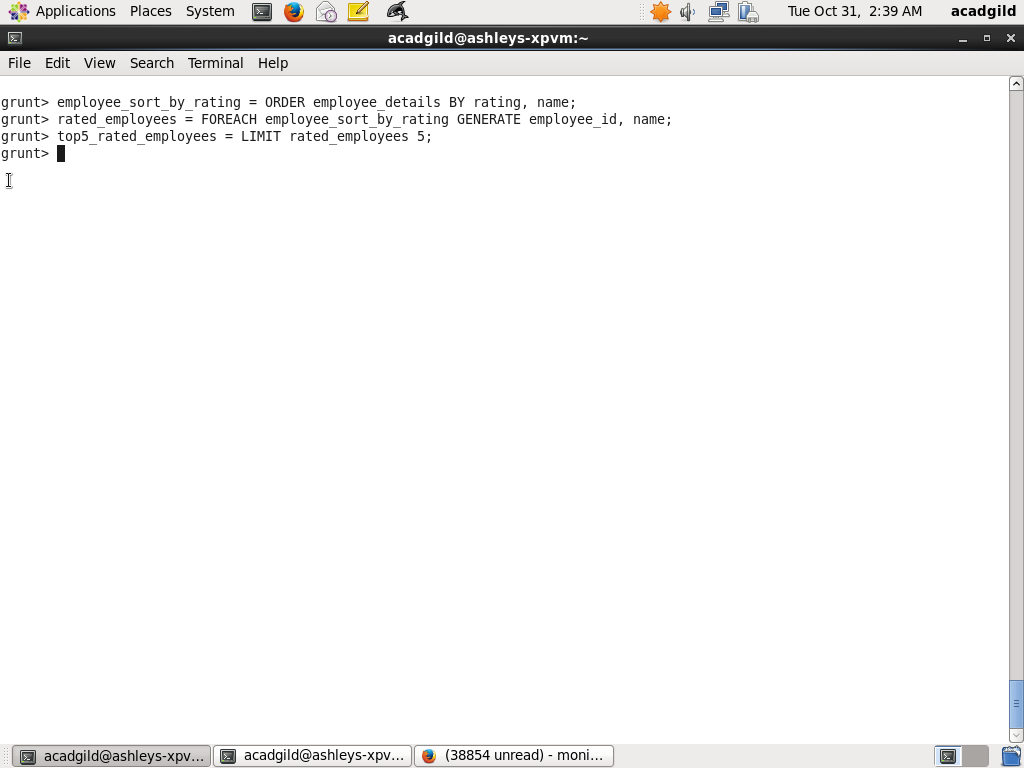
rated\_employees = FOREACH employee\_sort\_by\_rating GENERATE employee\_id, name;

Step3 :

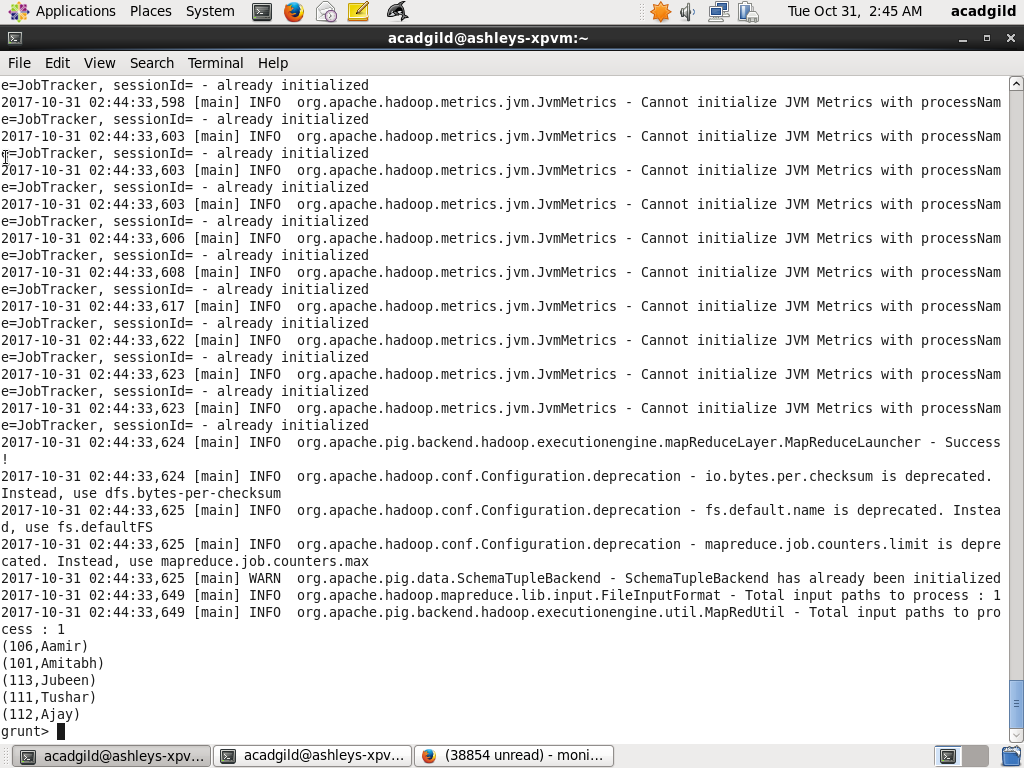
Get only first 5 tuples using LIMIT 5

top5\_rated\_employees = LIMIT rated\_employees 5;

Screenshot is as below:



Output of Pig script is as below:



Task b. Top 3 employees with highest salary whose employee id is odd

The following script will first filter odd numbered employees by using modulus of employee\_id and checking value is 1. Next sort them using saraly descending and name acending. Next select first 3 tuples by using LIMIT 3 statement

Step1:

Filter odd numbered employee\_id by using modulus operator (%) on

Employee\_id and check for value aas 1 and forming the relation odd\_numbered\_employees

odd\_numbered\_employees = FILTER employee\_details BY employee\_id % 2 == 1;

Step2:

Sort the odd numbered employees by salary descending and name ascending and derive relation odd\_numbered\_employees\_sorted\_by\_salary

odd\_numbered\_employees\_sorted\_by\_salary = ORDER odd\_numbered\_employees BY salary DESC, name;

Step3:

Slect only fields employeed\_id, name from odd\_numbered\_employees\_sorted\_by\_salary and form the new relation odd\_numbered\_employees\_id\_name\_sorted\_by\_salary

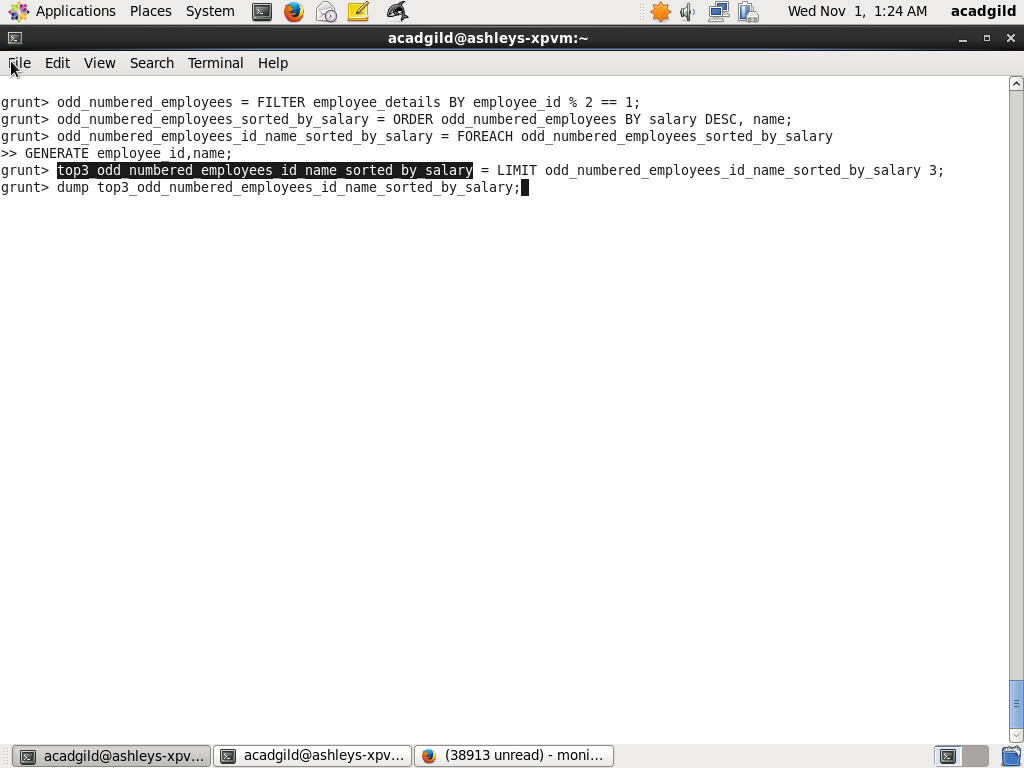
odd\_numbered\_employees\_id\_name\_sorted\_by\_salary = FOREACH odd\_numbered\_employees\_sorted\_by\_salary GENERATE employee\_id, name;

Step4:

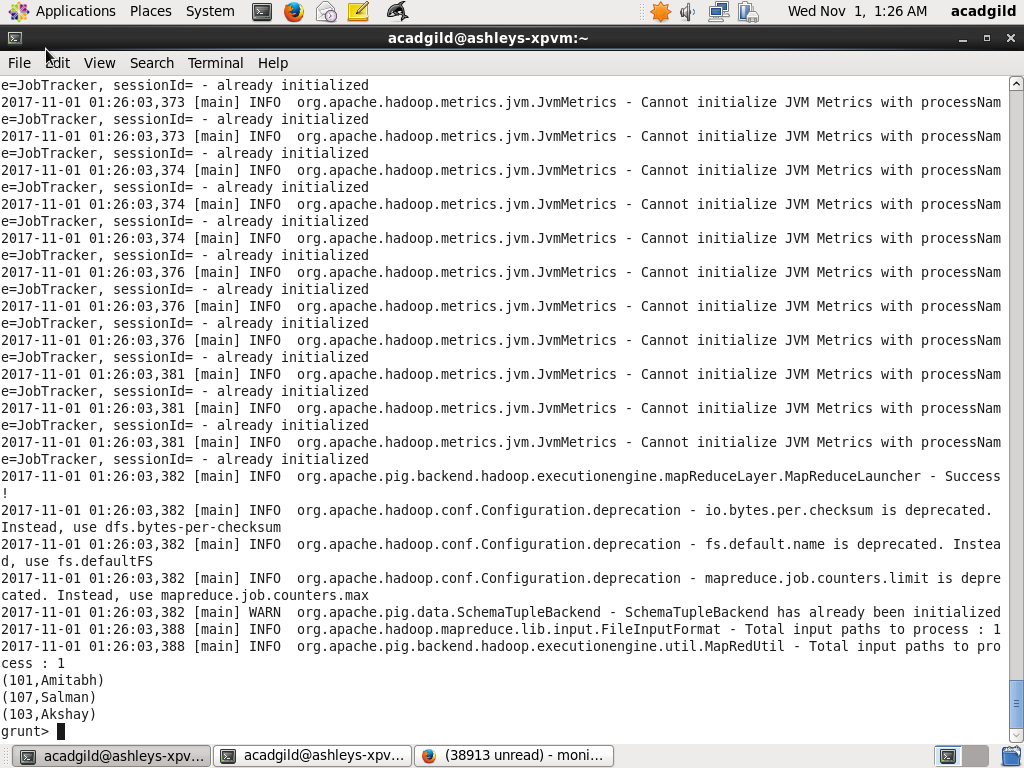
Get only first 3 tuples using LIMIT

top3\_odd\_numbered\_employees\_id\_name\_sorted\_by\_salary = LIMIT odd\_numbered\_employees\_id\_name\_sorted\_by\_salary 3;

Screenshot is as below



When I dump top3\_odd\_numbered\_employees\_id\_name\_sorted\_by\_salary, the output is as below (Amitabh, Salman, Akshay)



Task c

Employee with maximum expense

The following script below first join employee\_details and employee\_expenses using COGROUP on employee\_id. Next calculate total expenses for each employee using SUM operator. Next, group all employees by ALL and get the maximum total expenses. This relation is again joined with employee\_details\_expenses relation by total\_expenses. From this relation select only employee\_id and name. Next flatten the list. Last step, sort by name

Step1:

Join employee\_details and employee\_expenses by employee\_id using COGROUP and form relation employee\_details\_expenses

employee\_details\_expenses = COGROUP employee\_details BY employee\_id, employee\_expenses BY employee\_id;

Step2:

Sum expenses on each employee and select name as emp\_name, employee\_id as emp\_id and sum of expenses as total\_expenses and form the relation employee\_details\_total\_expenses

employee\_details\_total\_expenses = FOREACH employee\_details\_expenses GENERATE employee\_details.name AS emp\_name, employee\_details.employee\_id AS emp\_id, SUM(employee\_expenses.expenses) AS total\_expenses;

Step3:

Group employee\_details\_total\_expenses by all

employee\_details\_total\_expenses\_group\_all = GROUP employee\_details\_total\_expenses ALL;

Step4:

Calculate maximum total expense on employee\_details\_total\_expenses\_group\_all and form the relation max\_total\_expenses

max\_total\_expenses = FOREACH employee\_details\_total\_expenses\_group\_all GENERATE group AS grp, MAX(employee\_details\_total\_expenses.total\_expenses) AS total\_expenses;

Step5:

Join the relations employee\_details\_total\_expenses, max\_total\_expenses by total\_expenses and form the new relation employee\_details\_total\_expenses\_join\_max\_total\_expenses

employee\_details\_total\_expenses\_join\_max\_total\_expenses = JOIN employee\_details\_total\_expenses BY total\_expenses, max\_total\_expenses BY total\_expenses;

Step6:

From relation employee\_details\_total\_expenses\_join\_max\_total\_expenses, select only em\_id and emp\_name and form the relation employee\_id\_name\_with\_max\_total\_expenses

employee\_id\_name\_with\_max\_total\_expenses = FOREACH employee\_details\_total\_expenses\_join\_max\_total\_expenses GENERATE $0 AS emp\_id, $1 AS emp\_name;

Step7:

Flatten fields n relation employee\_id\_name\_with\_max\_total\_expenses and form relation employee\_id\_name\_with\_max\_total\_expenses\_flatten

employee\_id\_name\_with\_max\_total\_expenses\_flatten = FOREACH employee\_id\_name\_with\_max\_total\_expenses GENERATE FLATTEN($1), FLATTEN($0) ;

Step8:

Sort employee\_id\_name\_with\_max\_total\_expenses\_flatten by employee name

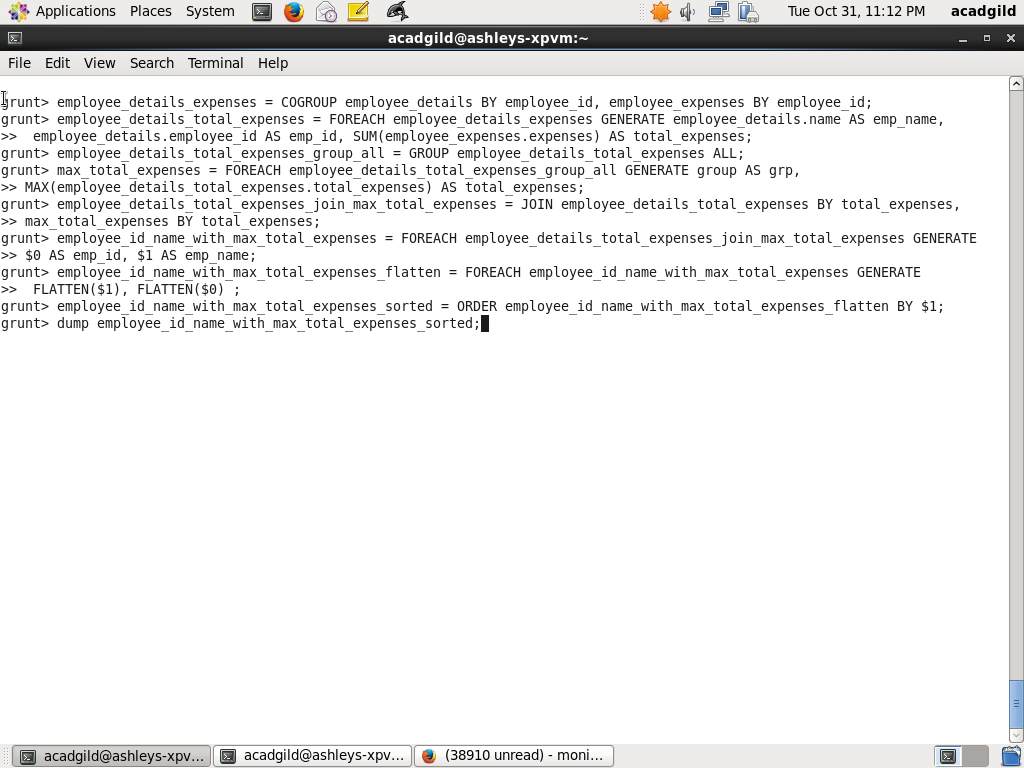
employee\_id\_name\_with\_max\_total\_expenses\_sorted = ORDER employee\_id\_name\_with\_max\_total\_expenses\_flatten BY $1;

Step9:

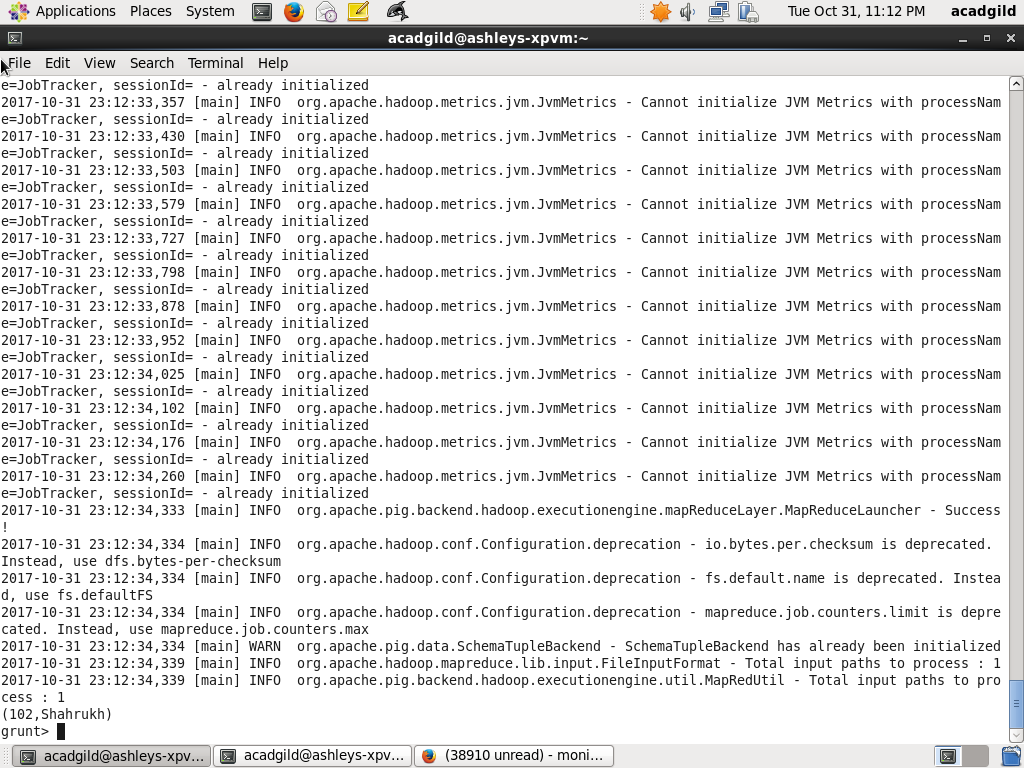
Dump the relation employee\_id\_name\_with\_max\_total\_expenses\_sorted

dump employee\_id\_name\_with\_max\_total\_expenses\_sorted;

Screenshot is as below:



Screenshot of last page of output of dump employee\_id\_name\_with\_max\_total\_expenses\_sorted is as where employee\_id: 102, name: Shahrukh has maximum expenses



Task d. List of employees who have entries in employee\_expenses file is as below:

Step1: Join is done between employee\_details and employee\_expenses and relation employee\_details\_expenses relation is formed.

employee\_details\_expenses = COGROUP employee\_details BY employee\_id, employee\_expenses BY employee\_id;

Step2: Next, FILTER those records in employee\_details\_expenses which has employee\_expenses is not empty by checking IsEmpty function returning false

employee\_details\_expenses\_non\_empty = FILTER employee\_details\_expenses BY IsEmpty(employee\_expenses) == false;

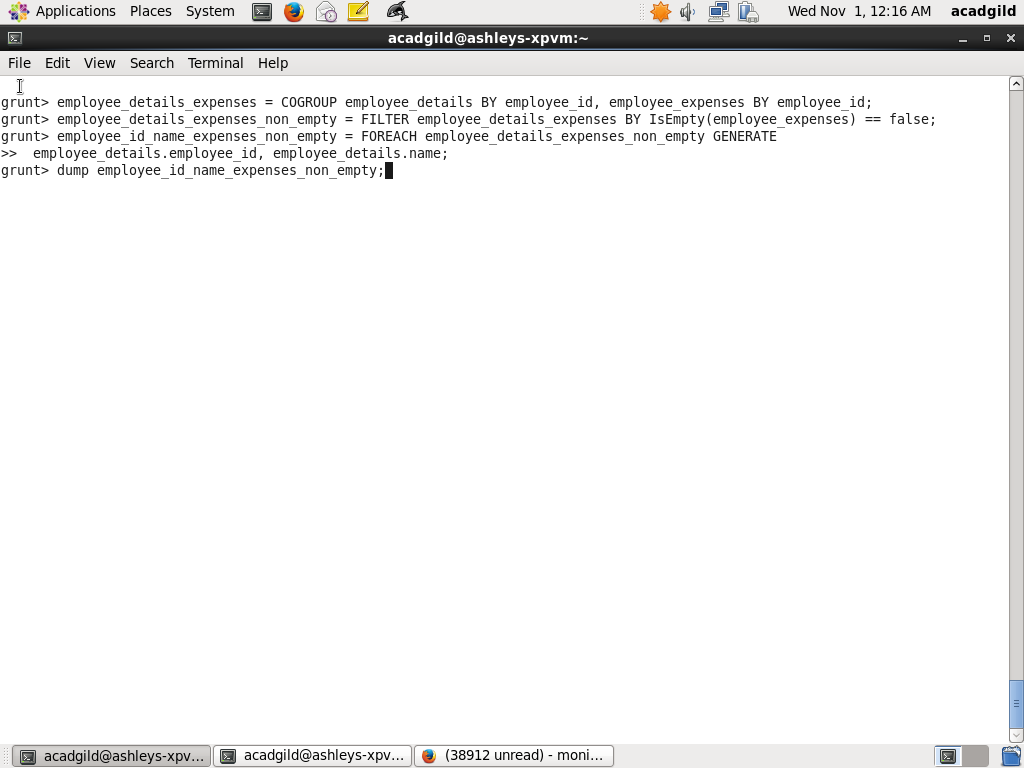
Step3. select only fields employee\_id and name and form the relation employee\_id\_name\_expenses\_non\_empty

employee\_id\_name\_expenses\_non\_empty = FOREACH employee\_details\_expenses\_non\_empty GENERATE employee\_details.employee\_id, employee\_details.name;

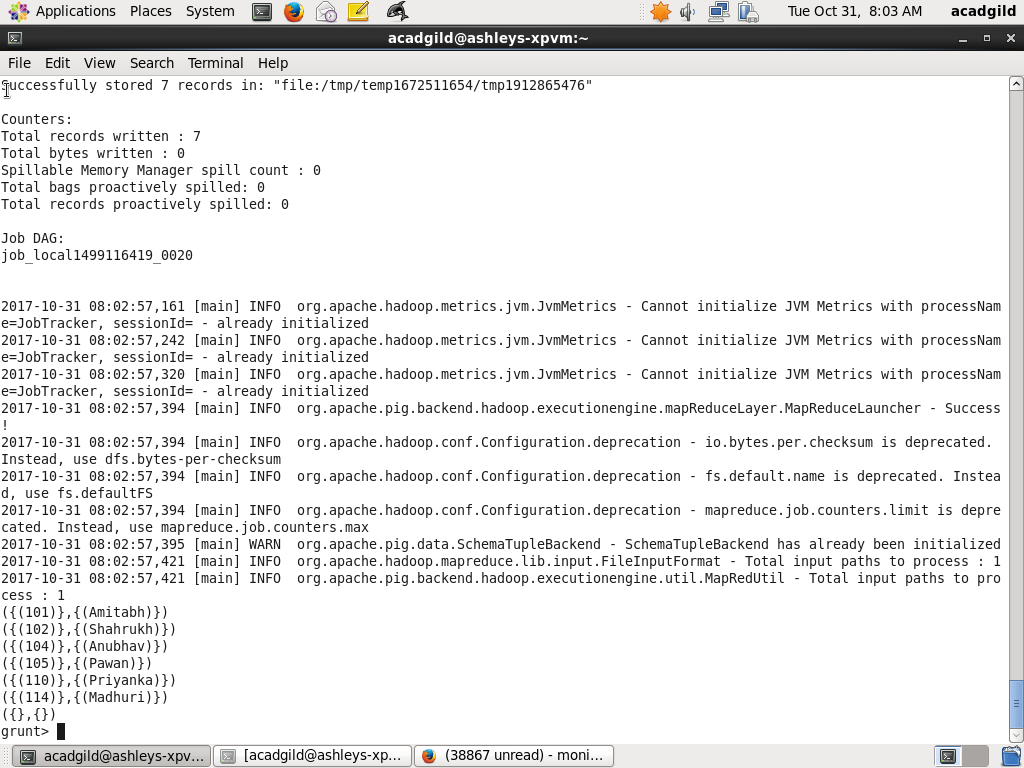
Step4

dump employee\_id\_name\_expenses\_non\_empty

Screenhosts are as below



When I dump employee\_id\_name\_expenses\_non\_empty I get the following screenshot which shows Amitabh, Shahrukh, Anubhav, Pawan, Priyanka, Madhuri



Task e. List of employees who have no entries in employee\_expenses file is as below:

Step1: Join is done between employee\_details and employee\_expenses using COGROUP and relation employee\_details\_expenses relation is formed.

employee\_details\_expenses = COGROUP employee\_details BY employee\_id, employee\_expenses BY employee\_id;

Step2: Next, FILTER those records in employee\_details\_expenses which has employee\_expenses has IsEmpty function returning true

employee\_details\_expenses\_empty = FILTER employee\_details\_expenses BY IsEmpty(employee\_expenses);

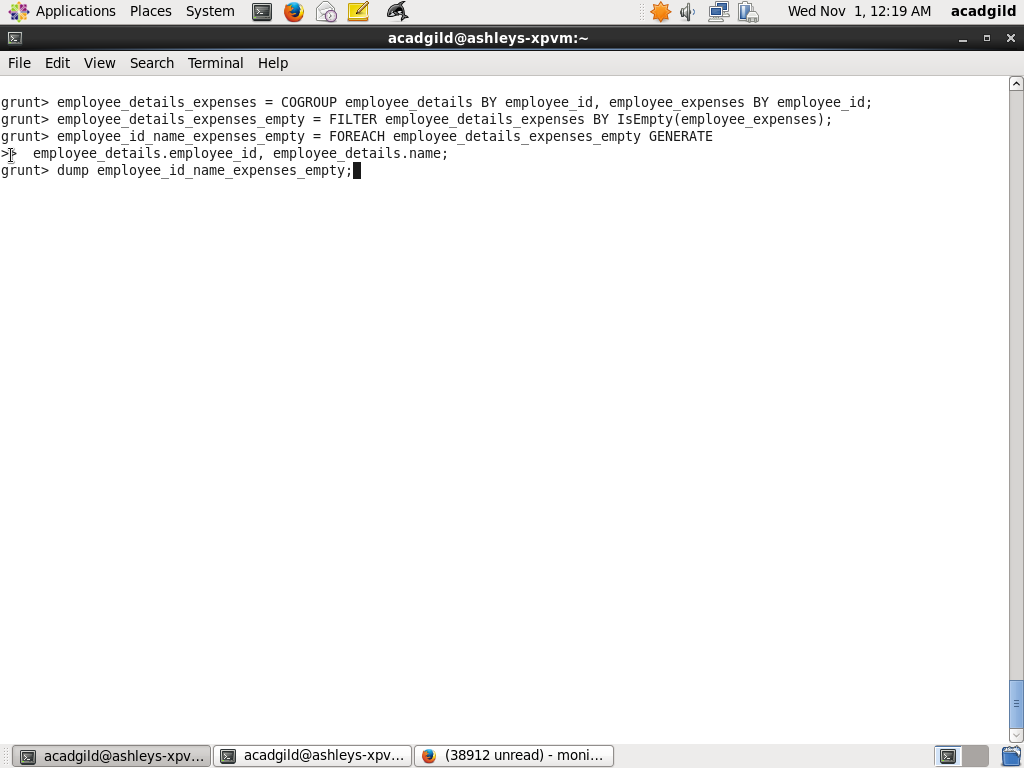
Step3. select only fields employee\_id and name and form the relation employee\_id\_name\_expenses\_non\_empty

employee\_id\_name\_expenses\_empty = FOREACH employee\_details\_expenses\_empty GENERATE employee\_details.employee\_id, employee\_details.name;

Step4

dump employee\_id\_name\_expenses\_empty

Screenhosts are as below



When I issue dump on employee\_id\_name\_expenses\_empty, the screenshot is as below:

