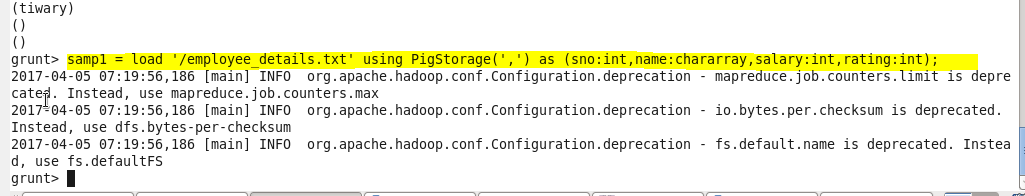
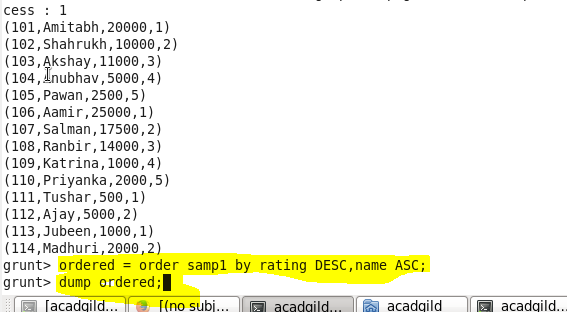
We have employee\_details and employee\_expenses files. Use local mode while running Pig and write Pig Latin script.

1. Top 5 employees (employee id and employee name) with highest rating. (In case two employees have same rating, employee with name coming first in dictionary should get preference)

STEP 1: LOAD “employee\_details” FILE



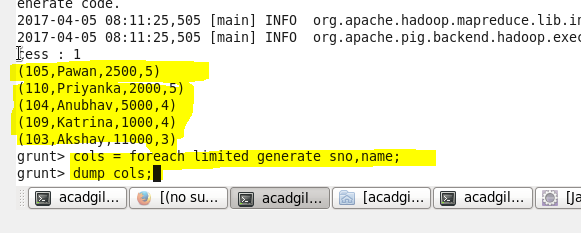
STEP 2: ORDERING OF DATA BASED ON “RATING “ IN DESCENDING ORDER TO GET THE HIGHEST RATING



STEP 3: LIMITING THE DATA TO TOP 5 RECORDS IN ALPHABETIACL ORDER N HIGHEST RATING

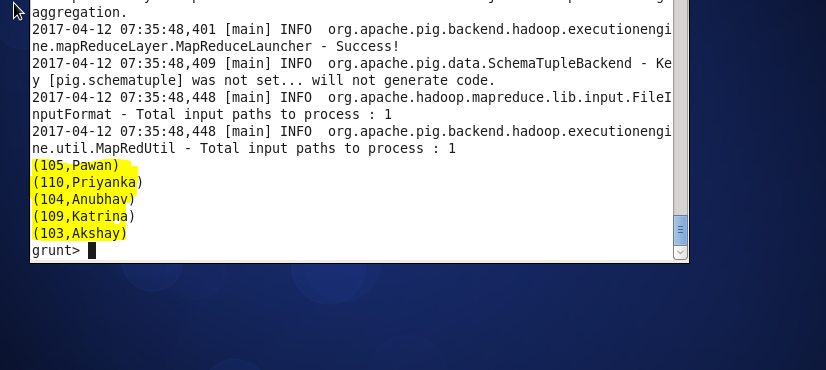


STEP 4: USING FOREACH COMMAND TO DISPLAY ONLY “SNO AND NAME “ COLUMNS



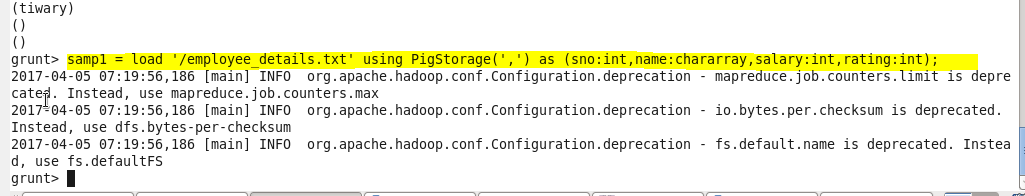
STEP 5: FINAL OUTPUT SHOWING TOP 5 RATING IN ALPHABETICAL ORDER

WITH SNO AND NAME FIELDS

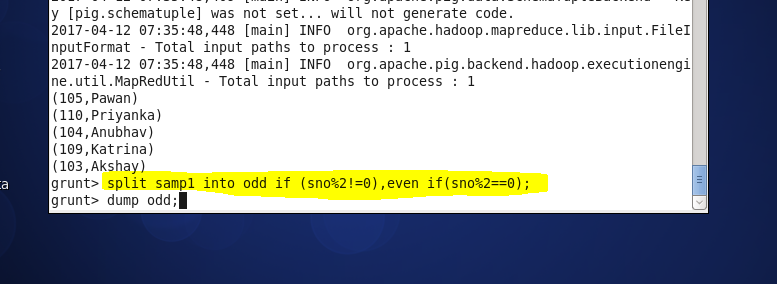


1. Top 3 employees (employee id and employee name) with highest salary, whose employee id is an odd number. (In case two employees have same salary, employee with name coming first in dictionary should get preference)

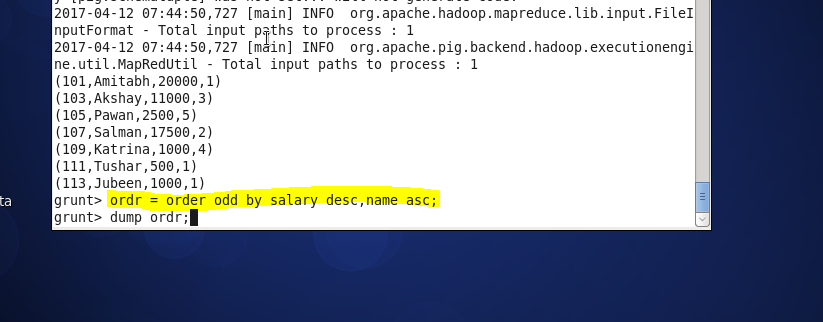
STEP1: LOAD "employee\_details" File In It



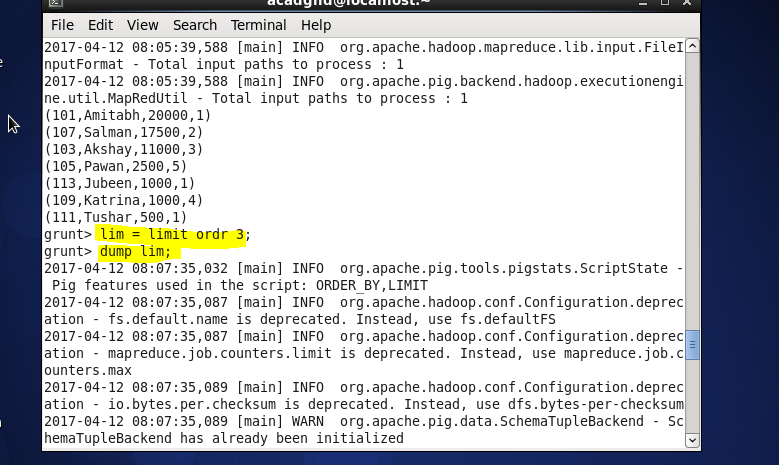
STEP2 : USE “SPLIT COMMAND” FOR GETTING ONLY ODD NUMBERED “SNO” IN EMPLOYEE



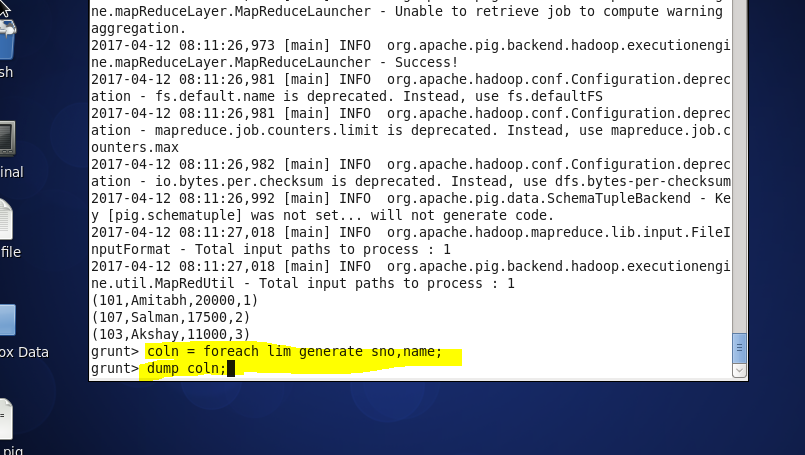
STEP 3: ORDER THE DATA IN ORDER OF HIGHEST SALARY ANDV ALPHABETICAL ORDER OF NAMES



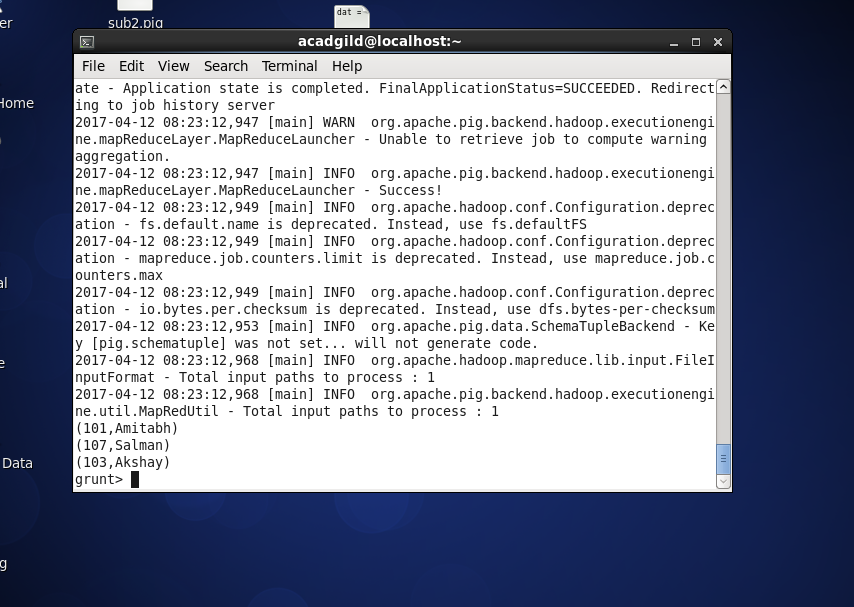
STEP 4: LIMIT THE DATA TO 3 RECORDS FROM EMPLOYEE



STEP 5: USE “FOREACH COMMAND” TO GET ONLY “SNO,NAME” FIELDS

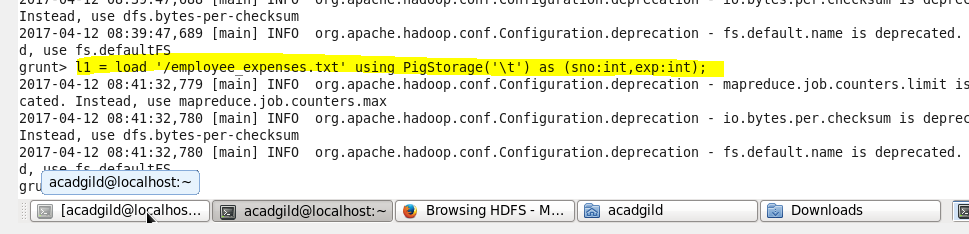


STEP 6: OUTPUT SHOWING “SNO,NAME” OF 3EMPLOYEE S WITH HIGHEST SALARY

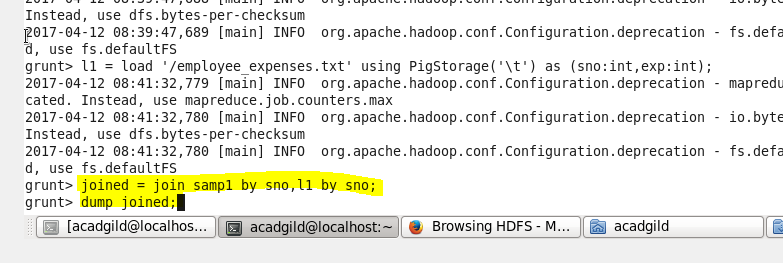


1. ) Employee (employee id and employee name) with maximum expense (In case two employees have same expense, employee with name coming first in dictionary should get preference

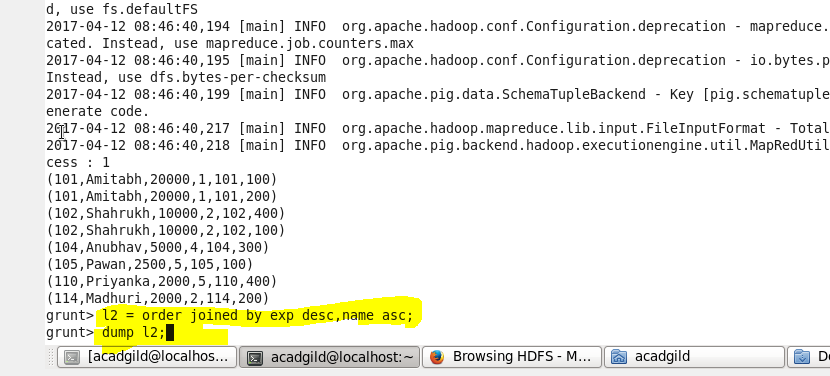
STEP 1: LOAD THE DATASET "employee\_expenses"



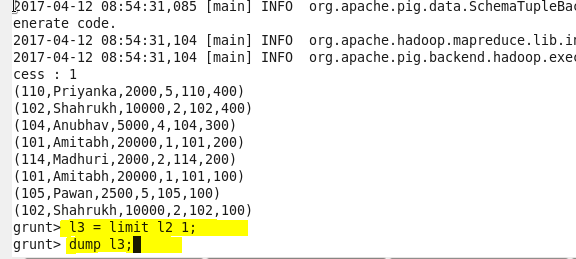
STEP2: JOIN THE 2 DATASETS AND PERFORMING ORDERING BASED ON HIGHEST EXPENSES



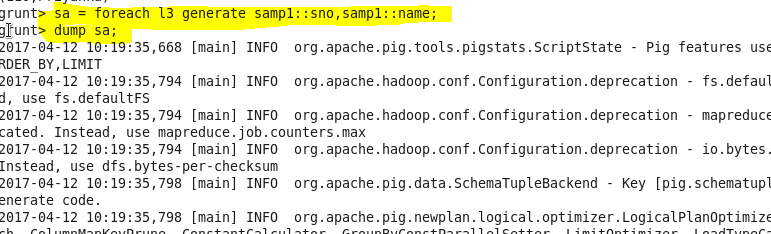
STEP 3: LIMIT THE DATA TO ONE RECORD USING LIMIT COMMAND

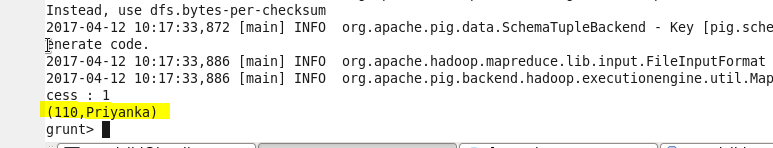


STEP 4: USE limit COMMAND TO PRINT ONLY SNO AN NAME OF EMPLOYEE WITH HIGHEST EXPENSE



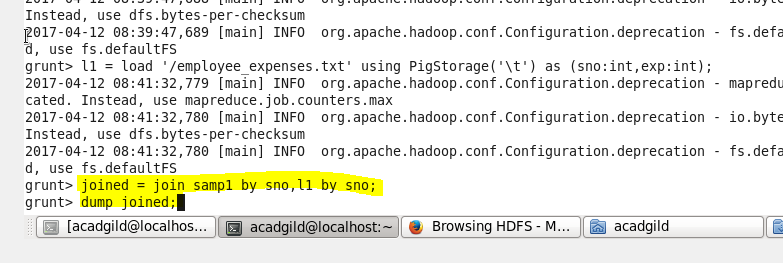
STEP 5 : OUTPUT



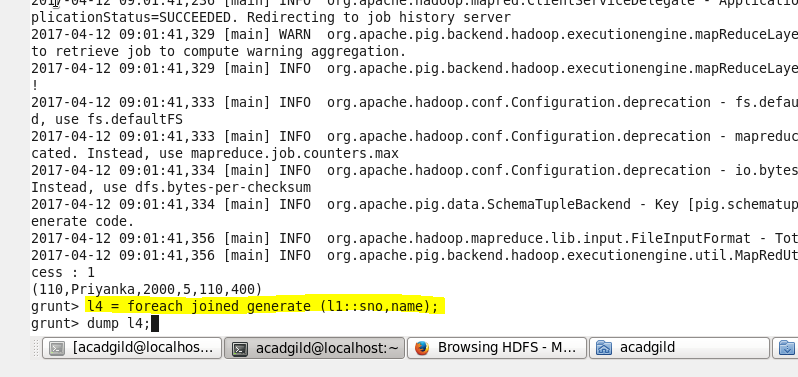


1. List of employees (employee id and employee name) having entries in employee\_expenses file.

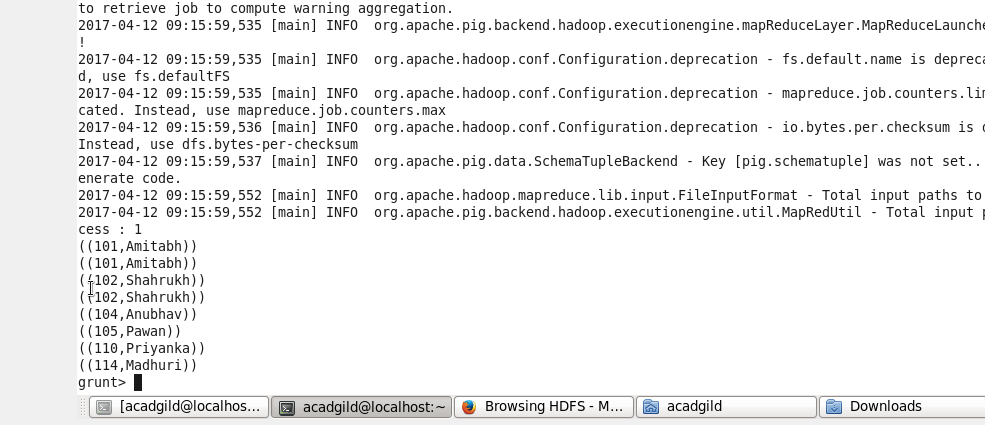
STEP 1: PERFORM SELF JOIN FOR 2 DATA SETS



STEP 2: USE FOREACH FOR DISPLAYING SNO AND NAME OF EMPLOYEES HAVING ENTRIES IN EXPENSE



STEP 3: OUTPUT

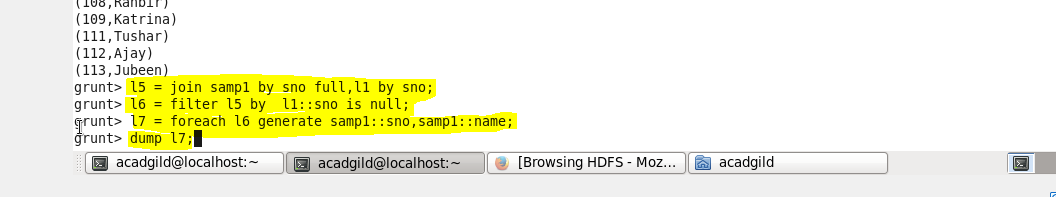


(e) List of employees (employee Id ) having no entry in employee\_expenses file.

STEP 1 : use full join on employee\_details and by sno on employee\_expenses

\* filter by leaving out null values in the above relation to eliminate entries from expenses

\* use foreach to generate sno,name

 STEP 2: output

