**5.1 Assignment**

**(a)**

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)

**Solution**

// Load data from text file to pig –x local

**emp\_details = LOAD 'employee\_details.txt' USING PigStorage(',') as (emp\_id:int,emp\_name:chararray,emp\_salary:int,emp\_rating:int);**



//grouping emp\_details

**emp\_analysis = foreach (group emp\_details all)**

**{**

**//** ordering fields based on the ratings as descending and ordering name as alphabet wise

**sort = ORDER emp\_details by emp\_rating DESC,emp\_name ASC;**

**//** displaying 1st 5 rows

**result = LIMIT sort 5;**

**generate flatten(result);**

**}**

**emp\_result = foreach emp\_analysis generate result::emp\_id,result::emp\_name;**

//displaying processed data

**dump emp\_analysis**

**Final output for (a)**

****

**(b)**

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)

**Solution**

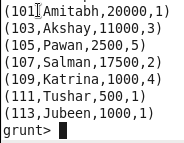
// Load data from text file to pig –x local

**emp\_details = LOAD 'employee\_details.txt' USING PigStorage(',') as (emp\_id:int,emp\_name:chararray,emp\_salary:int,emp\_rating:int);**



// Filtering odd ids

**filter\_id = FILTER emp\_details BY emp\_id %2 == 1;**

****

// grouping above output

**emp\_analysis = foreach (group filter\_id all)**

**{**

**//** ordering fields based on the salary as descending and ordering name as alphabet wise

**sort = ORDER filter\_id by emp\_salary DESC,emp\_name ASC;**

**//** displaying 1st 5 rows

**result = LIMIT sort 3;**

**//** generating group to fields using flatten

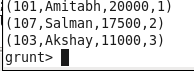
**generate flatten(result);**

**}**

//displaying processed data

**dump emp\_analysis**

**Final output for (b)**

****

**(c)**

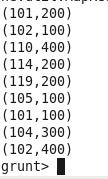
Employee (employee id and employee name) with maximum expense (In case two employees have same expense, employee with name coming first in dictionary should get preference)

**Solution**

**emp\_details = LOAD 'employee\_details.txt' USING PigStorage(',') as (emp\_id:int,emp\_name:chararray,emp\_salary:int,emp\_rating:int);**

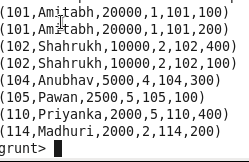
****

**b\_emp\_details = LOAD 'employee\_expenses.txt' as (emp\_id:int,emp\_expense:int);**

****

**//**joining two files using id as common

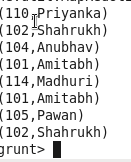
**c = join emp\_details by emp\_id,b\_emp\_details by emp\_id;**

****

**sort = ORDER c by b\_emp\_details::emp\_expense DESC,emp\_details::emp\_name ASC;**

**result = foreach sort generate emp\_details::emp\_id,emp\_details::emp\_name;**

**dump result;**

****

**(d)**

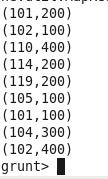
List of employees (employee id and employee name) having entries in employee\_expenses file.

**Solution:**

**emp\_details = LOAD 'employee\_details.txt' USING PigStorage(',') as (emp\_id:int,emp\_name:chararray,emp\_salary:int,emp\_rating:int);**

****

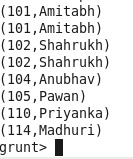
**b\_emp\_details = LOAD 'employee\_expenses.txt' as (emp\_id:int,emp\_expense:int);**

****

**c = join emp\_details by emp\_id,b\_emp\_details by emp\_id;**

**result = foreach c generate emp\_details::emp\_id,emp\_details::emp\_name;**

**dump result;**

****

**(e)**

List of employees (employee id and employee name) having no entry in employee\_expenses

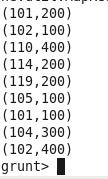
file.

**Solution:**

emp\_details = LOAD 'employee\_details.txt' USING PigStorage(',') as (emp\_id:int,emp\_name:chararray,emp\_salary:int,emp\_rating:int);

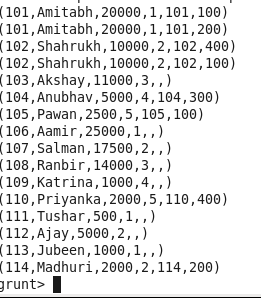


**b\_emp\_details = LOAD 'employee\_expenses.txt' as (emp\_id:int,emp\_expense:int);**



//left outer join

**c = join emp\_details by emp\_id LEFT OUTER,b\_emp\_details by emp\_id;**



**filter\_id = FILTER c BY b\_emp\_details::emp\_id is null;**



**result = foreach filter\_id generate emp\_details::emp\_id,emp\_details::emp\_name;**

**dump result;**

