

Assignment7.3

```
grunt> emp_details = LOAD '/home/acadgild/Desktop/data/employee_details.txt' USING PigStorage(',') AS (id:int, name:chararray, salary:int, rating:int);
2017-03-25 07:43:30,578 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.persist.jobstatus.hours is deprecated. Instead, use mapreduce.jobtracker.persist.jobstatus.hours
2017-03-25 07:43:30,578 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.heartbeats.in.second is deprecated. Instead, use mapreduce.jobtracker.heartbeats.in.second
2017-03-25 07:43:30,578 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - jobclient.completion.poll.interval is deprecated. Instead, use mapreduce.client.completion.pollinterval
2017-03-25 07:43:30,578 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.tasktracker.tasks.sleep-time-before-sigkill is deprecated. Instead, use mapreduce.tasktracker.tasks.sleep-time-before-sigkill
2017-03-25 07:43:30,578 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.http.address is deprecated. Instead, use mapreduce.job.tracker.http.address
```

```
(101,Amitabh,20000,1)
(102,Shahrukh,10000,2)
(103,Akshay,11000,3)
(104,Anubhav,5000,4)
(105,Pawan,2500,5)
(106,Aamir,25000,1)
(107,Salman,17500,2)
(108,Ranbir,14000,3)
(109,Katrina,1000,4)
(110,Priyanka,2000,5)
(111,Tushar,500,1)
(112,Ajay,5000,2)
(113,Jubeen,1000,1)
(114,Madhuri,2000,2)
```

```
grunt> emp_expenses = LOAD '/home/acadgild/Desktop/data/employee_expenses.txt' USING PigStorage('\t') AS (id:int, expenses:int);
2017-03-25 07:43:42,805 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.counters.max
2017-03-25 07:43:42,805 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-03-25 07:43:42,805 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
```

```
(101,200)
(102,100)
(110,400)
(114,200)
(119,200)
(105,100)
(101,100)
(104,300)
(102,400)
```

- (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)

```
grunt> emp_order = order emp_details by rating desc, name asc;
grunt> lim = limit emp_order 5;
grunt> emp = foreach lim generate id, name;
grunt> dump emp;
```

```
(105, Pawan)
(110, Priyanka)
(104, Anubhav)
(109, Katrina)
(103, Akshay)
```

- (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)

```
grunt> emp_fil = filter emp_details by (id % 2) != 0;
grunt> empl_order = order emp_fil by salary desc;
grunt> lim1 = limit empl_order 3;
grunt> empl = foreach lim1 generate id, name;
grunt> dump empl;
```

```
(101, Amitabh)
(107, Salman)
(103, Akshay)
```

- (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)

```
grunt> emp_join = join emp_details by id, emp_expenses by id;
grunt> emp2_order = order emp_join by expenses desc, name asc;
grunt> lim2 = limit emp2_order 1;
grunt> emp2 = foreach lim2 generate $0,$1;
grunt> dump emp2;
```

```
Process : 1
```

```
(110,Priyanka)
```

- (d) List of employees (employee id and employee name) having entries in employee_expenses file.

```
grunt> emp_join1 = join emp_details by id, emp_expenses by id;
grunt> emp3 = foreach emp_join1 generate $0,$1;
grunt> dist = distinct emp3;
grunt> dump dist;
```

```
(101,Amitabh)
(102,Shahrukh)
(104,Anubhav)
(105,Pawan)
(110,Priyanka)
(114,Madhuri)
```

(e) List of employees (employee id and employee name) having no entry in employee_expenses file.

```
grunt> emp_join2 = join emp_details by id left outer, emp_expenses by id;  
grunt> emp4_fil = filter emp_join2 by $4 is null and $5 is null;  
grunt> emp4 = foreach emp4_fil generate $0,$1;  
grunt> dist1 = distinct emp4;  
grunt> dump dist1;
```

```
{103, Akshay}  
{106, Aamir}  
{107, Salman}  
{108, Ranbir}  
{109, Katrina}  
{111, Tushar}  
{112, Ajay}  
{113, Jubeen}
```