The below Questions use 2 file employee_details.txt and employee_expenses.txt as input

Contents of employee_details.txt and employee_expenses.txt is as below:

employee_details.txt (emp) (id, name,salary,rating)	employee_expenses.txt (emp_exp) (id, expenses)
101,Amitab,20000,1	101 200
102,Shahrukh,10000,2	102 100
103,Akshay,11000,3	110 400
104,Anubhav,5000,4	114 200
105,Pawan,2500,5	119 200
106,Aamir,25000,1	105 100
107,Salman,17500,2	101 100
108,Ranbir,14000,3	104 300
109,Katrina,1000,4	102 400
110,Priyanka,2000,5	
111,Tushar,500,1	
112,Ajay,5000,2	
113,Jubeen,1000,1	
114,Madhuri,2000,2	

Q. 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)
File used is employee_details.txt

Loading the employee_details.txt

```
emp = Load '/home/acadgild/pig/employee_details.txt' USING PigStorage(',') AS (id:int, name:chararray, salary:int, rating:int);
```

Steps to find the Top 5 employees (with the above said condition)

1. Ordering emp by rating in descending order(highest rating will come first) and name in ascending;
2. Limiting the result size to 3.
3. Fetching the id & name for the 3 employees with highest rating

```
emp_group_rating = ORDER emp BY rating DESC,name ASC;
emp_limit_rating = LIMIT emp_group_rating 5;
emp_by_rating_res = FOREACH emp_limit_rating GENERATE id,name;
DUMP emp_by_rating_res;
```

Result:

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

Q. 2) 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)

File used is employee_details.txt

Loading the employee_details.txt

```
emp = Load '/home/acadgild/pig/employee_details.txt' USING PigStorage(',') AS (id:int, name:chararray, salary:int, rating:int);
```

Steps to find the Top 3 employees (with the above said condition)

1. Filtering the employee records that contains id in odd numbers;
2. Ordering emp by salary in descending order(highest salary will come first) and name in ascending;
3. Limiting the result size to 3.
4. Fetching the id & name for the 3 employees with highest salary

```
emp_filter = FILTER emp BY (id%2!=0);
emp_group_sal = ORDER emp_filter BY salary DESC,name ASC;
emp_limit_sal = LIMIT emp_group_sal 3;
emp_by_sal_res = FOREACH emp_limit_sal GENERATE id,name;
DUMP emp_by_sal_res ;
```

Result:

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

For Q.3, Q.4 & Q.5 we need to combine emp & emp_exp table based on id. It is done as below. This result will be used in Q.3, Q.4 & Q.5

Loading the employee_details.txt

```
emp = Load '/home/acadgild/pig/employee_details.txt' USING PigStorage(',') AS (id:int, name:chararray, salary:int, rating:int);
```

Loading the employee_expenses.txt

```
emp_exp = Load '/home/acadgild/pig/employee_expenses.txt' AS (id:int, expenses:int);
```

1. One employee has more than one expenses entry in employee_expenses.txt, first we are grouping the expenses details in terms of id
2. Then foreach group(id's) we are generating the total expenditure
3. Then we are performing left outer join of emp & emp_total_exp, since we need the details of all the employees in the employee_details.txt (though they have no entry in the employee_expenses file)
4. Since both the tables are joined, Id column may be duplicated from both employee_details.txt and employee_expenses.txt. So we are generating (full_emp) with id, name,salary, rating and expenses and discarding the id taken from expenses.

```
exp_grp = GROUP emp_exp BY id;
emp_total_exp = FOREACH exp_grp GENERATE group AS id, SUM(emp_exp.$1) AS expenses;

joined_emp = JOIN emp by id left outer, emp_total_exp by id;
full_emp = FOREACH joined_emp generate emp::id AS id,name AS name,salary AS salary,rating as rating,expenses as expenses;

DUMP full_emp;
```

Result: (id, name, salary, rating, expenses)

```
(101,Amitab,20000,1,300)
(102,Shahrukh,10000,2,500)
(103,Akshay,11000,3,)
(104,Anubhav,5000,4,300)
(105,Pawan,2500,5,100)
(106,Aamir,25000,1,)
(107,Salman,17500,2,)
(108,Ranbir,14000,3,)
(109,Katrina,1000,4,)
(110,Priyanka,2000,5,400)
(111,Tushar,500,1,)
(112,Ajay,5000,2,)
(113,Jubeen,1000,1,)
(114,Madhuri,2000,2,200)
```

Q. 3) 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)
File used is employee_details.txt and employee_expenses.txt

Steps to find the employees with maximum expenses

1. Ordering emp by expenses in descending order(highest expenses will come first) and name in ascending;
2. Limiting the result size to 3.
3. Fetching the id & name for the 3 employees with highest expenses.

```
emp_group_exp = ORDER full_emp BY expenses DESC,name ASC;
emp_limit_exp = LIMIT emp_group_exp 3;
emp_exp_result_final = FOREACH emp_limit_exp GENERATE id,name;
DUMP emp_exp_result_final;
```

Result:

```
(102,Shahrukh)
(110,Priyanka)
(101,Amitab)
```

Q. 4) List of employees (employee id and employee name) having entries in employee_expenses file.
File used is employee_details.txt and employee_expenses.txt

Steps to find the employees having entry in employee_expenses file

1. Filtering the employee records that contains expenses details;
2. Fetching the id & name for the employees with entry in employee_expenses.txt table

```
exp_not_null = FILTER full_emp BY expenses IS NOT  NULL;;
exp_not_null_res = FOREACH exp_not_null GENERATE id,name;
DUMP exp_not_null_res;
```

Result:

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

Q. 5) List of employees (employee id and employee name) having no entry in employee_expenses file.
File used is employee_details.txt and employee_expenses.txt

Steps to find the employees not having entry in employee_expenses file

1. Filtering the employee records that does not contains expenses details;
2. Fetching the id & name for the employees without entry in employee_expenses.txt table

```
exp_null = FILTER full_emp BY expenses IS NULL;
exp_null_res = FOREACH exp_null GENERATE id,name;
DUMP exp_null_res;
```

Result:

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