Problem Statement:

We have employee_details and employee_expenses files. Use local mode while running Pig and write Pig Latin script to get below results:

https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee_details.txt

```
102     400grunt> cat employee_details;
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,2000,2grunt>
```

https://github.com/prateekATacadgild/DatasetsForCognizant/blob/master/employee_expenses.txt

```
grunt> cat employee_expenses;
101     200
102     100
110     400
114     200
119     200
105     100
101     100
104     300
102     400grunt>
```

(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:

```
grunt>emp = LOAD 'employee_details' USING PigStorage(',') AS
(emp_id:int, emp_name:chararray,emp_salary:int,emp_rating:int);

grunt>emp_order = ORDER emp by emp_rating DESC, emp_name ASC;
grunt>five_limit = LIMIT emp_order 5;
```

```
grunt>emp = FOREACH five_limit GENERATE emp_id, emp_name,
emp_rating;

grunt>dump emp;
```

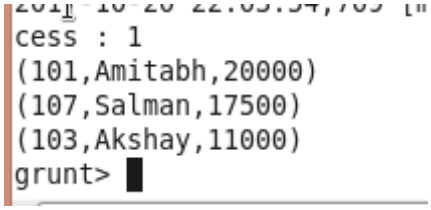
Output:

(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:

```
grunt>emp = LOAD 'employee_details' USING PigStorage(',') AS
(emp_id:int, emp_name:chararray,emp_salary:int,emp_rating:int);

grunt>emp_filter = FILTER emp by (emp_id %2) != 0;

grunt>emp_order = ORDER emp_filter by emp_salary DESC, emp_name ASC;

grunt>three_limit = LIMIT emp_order 3;

grunt>emp = FOREACH three_limit GENERATE emp_id, emp_name,
emp_salary;

grunt>dump emp;
```

Output:

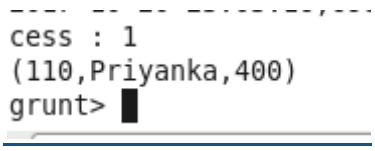


(c) Employee (employee id and employee name) with maximum expense (In case two employees have same expense, employee with name coming first in dictionary should get Preference)

Solution:

```
grunt>emp = LOAD 'employee_details' USING PigStorage(',') AS
(emp_id: int, emp_name: chararray, emp_salary: int, emp_rating:int);

grunt>emp_expense = LOAD 'employee_expenses' USING PigStorage('\t')
AS(emp_id:int, emp_expenses:int);

grunt>join_emp = JOIN emp by emp_id, emp_expense by emp_id;


grunt>emp_order = ORDER join_emp by emp_expenses DESC, emp_name ASC;

grunt>emp_limit = LIMIT emp_order 1;

grunt>emp = FOREACH emp_limit GENERATE $0,$1,$5;

grunt>dump emp;
```

Output:



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

Solution:

```
grunt>emp = LOAD 'employee_details' USING PigStorage(',') AS
(emp_id: int, emp_name: chararray, emp_salary: int, emp_rating:int);

grunt>emp_expense = LOAD 'employee_expenses' USING PigStorage('\t')
AS(emp_id:int, emp_expenses:int);

grunt>join_emp = JOIN emp by emp_id, emp_expense by emp_id;

grunt>emp = FOREACH join_emp GENERATE $0,$1;

grunt>distinct_emp = DISTINCT emp;


grunt>dump distinct_emp;
```

Output:

```
cess : 1
(101,Amitabh)
(102,Shahrukh)
(104,Anubhav)
(105,Pawan)
(110,Priyanka)
(114,Madhuri)
grunt>
```

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

Solution:

```
grunt>emp = LOAD 'employee_details' USING PigStorage(',') AS
(emp_id: int, emp_name: chararray, emp_salary: int, emp_rating:int);

grunt>emp_expense = LOAD 'employee_expenses' USING PigStorage('\t')
AS(emp_id:int, emp_expenses:int);

grunt>join_emp = JOIN emp by emp_id LEFT OUTER, emp_expense by
emp_id;

grunt>emp_filter = FILTER join_emp by $4 is null and $5 is null;

grunt>emp = FOREACH emp_filter GENERATE $0,$1;

grunt>distinct_emp = DISTINCT emp;

grunt>dump distinct_emp;
```

Output:

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