TASK 2 : We have employee_details and employee_expenses files. Use local mode while running Pig and write Pig Latin script to get below results:

employee_details (EmpID,Name,Salary,DepartmentID)

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

employee_expenses(EmpID,Expence)

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

```
1)grunt> employee_details = load '/home/acadgild/employee_details.txt' using PigStorage(',') as (empId:int,empName:chararray,salary:int,rating:int);
```

```
2)dump 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,2000,2)
```

```
3)employee_expenses = LOAD '/home/acadgild/employee_expenses.txt' using PigStorage('\t') as (empId:int,empExpense:Int);
```

```
4)dump employee_expenses
```

```
(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)

SOLUTION:

```
1)ratings_order = ORDER employee_details by rating DESC, empName;

2)dump ratings_order
```

```
(105,Pawan,2500,5)

(110,Priyanka,2000,5)

(104,Anubhav,5000,4)

(109,Katrina,1000,4)

(103,Akshay,11000,3)

(108,Ranbir,14000,3)

(112,Ajay,5000,2)

(114,Madhuri,2000,2)

(107,Salman,17500,2)
```

```
(102,Shahrukh,10000,2)

(106,Aamir,25000,1)

(101,Amitabh,20000,1)

(113,Jubeen,1000,1)

(111,Tushar,500,1)
```

```
3)output_a = LIMIT ratings_order 5;

4)dump output_a;------------------------------------->RESULT
```

```
(105,Pawan,2500,5)

(110,Priyanka,2000,5)

(104,Anubhav,5000,4)

(109,Katrina,1000,4)

(103,Akshay,11000,3)
```

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)

```
1)emp_oddID = FILTER employee_details by empId%2==1;

2)dump emp_oddID;
```

```
(101,Amitabh,20000,1)

(103,Akshay,11000,3)

(105,Pawan,2500,5)

(107,Salman,17500,2)

(109,Katrina,1000,4)

(111,Tushar,500,1)

(113,Jubeen,1000,1)
```

```
3)highest_salary= ORDER emp_oddID by salary DESC,empName;

4)dump highest_salary;

(101,Amitabh,20000,1)

(107,Salman,17500,2)

(103,Akshay,11000,3)

(105,Pawan,2500,5)

(113,Jubeen,1000,1)

(109,Katrina,1000,4)

(111,Tushar,500,1)

5)ouput_b= LIMIT highest_salary 3;

6)dump ouput_b------------------------------------------------>RESULT

(101,Amitabh,20000,1)

(107,Salman,17500,2)

(103,Akshay,11000,3)
```

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)

```
1)IN_join = JOIN employee_expenses by empId , employee_details by empId;

2)dump IN_join;

(101,100,101,Amitabh,20000,1)

(101,200,101,Amitabh,20000,1)

(102,400,102,Shahrukh,10000,2)

(102,100,102,Shahrukh,10000,2)

(104,300,104,Anubhav,5000,4)

(105,100,105,Pawan,2500,5)
```

```
(110,400,110,Priyanka,2000,5)

(114,200,114,Madhuri,2000,2)
```

```
3)max_expense = ORDER IN_join by employee_expenses::empExpense
DESC,employee_details::empName;

4)dump max_expense;
```

```
(110,400,110,Priyanka,2000,5)

(102,400,102,Shahrukh,10000,2)

(104,300,104,Anubhav,5000,4)

(101,200,101,Amitabh,20000,1)

(114,200,114,Madhuri,2000,2)

(101,100,101,Amitabh,20000,1)

(105,100,105,Pawan,2500,5)

(102,100,102,Shahrukh,10000,2)
```

```
5)ouput_c= LIMIT max_expense 1;----------------------------->RESULT
```

```
(110,400,110,Priyanka,2000,5)
```

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

```
1)IN_join = JOIN employee_expenses by empId , employee_details by empId;

2)IN_foreach = FOREACH combo generate employee_details::empId, employee_details::empName;

3)dump IN_foreach;
```

```
(101,Amitabh)

(101,Amitabh)

(102,Shahrukh)

(102,Shahrukh)

(104,Anubhav)

(105,Pawan)
```

```
(110,Priyanka)

(114,Madhuri)
```

```
4)output_d = DISTINCT IN_foreach;

5)dump output_d;-------------------------------------------------->RESULT
```

```
(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.

```
1)join_right= JOIN employee_expenses by empId RIGHT OUTER, employee_details by empId;

2)dump join_right;
```

```
(101,100,101,Amitabh,20000,1)

(101,200,101,Amitabh,20000,1)

(102,400,102,Shahrukh,10000,2)

(102,100,102,Shahrukh,10000,2)

(,,103,Akshay,11000,3)

(104,300,104,Anubhav,5000,4)

(105,100,105,Pawan,2500,5)

(,,106,Aamir,25000,1)

(,,107,Salman,17500,2)

(,,108,Ranbir,14000,3)

(,,109,Katrina,1000,4)
```

```
(110,400,110,Priyanka,2000,5)

(,,111,Tushar,500,1)

(,,112,Ajay,5000,2)

(,,113,Jubeen,1000,1)

(114,200,114,Madhuri,2000,2)
```

3)filter_expense= FILTER join_right by employee_expenses::empId is null;

4)dump filter_expense;

```
(,,103,Akshay,11000,3)

(,,106,Aamir,25000,1)

(,,107,Salman,17500,2)

(,,108,Ranbir,14000,3)

(,,109,Katrina,1000,4)

(,,111,Tushar,500,1)

(,,112,Ajay,5000,2)

(,,113,Jubeen,1000,1)
```

5)output_e= FOREACH filter_expense GENERATE employee_details::empId,employee_details::empName;

6)dump output_e;-->RESULT

```
(103,Akshay)

(106,Aamir)

(107,Salman)

(108,Ranbir)

(109,Katrina)

(111,Tushar)

(112,Ajay)

(113,Jubeen)
```