PIG Assignment 1

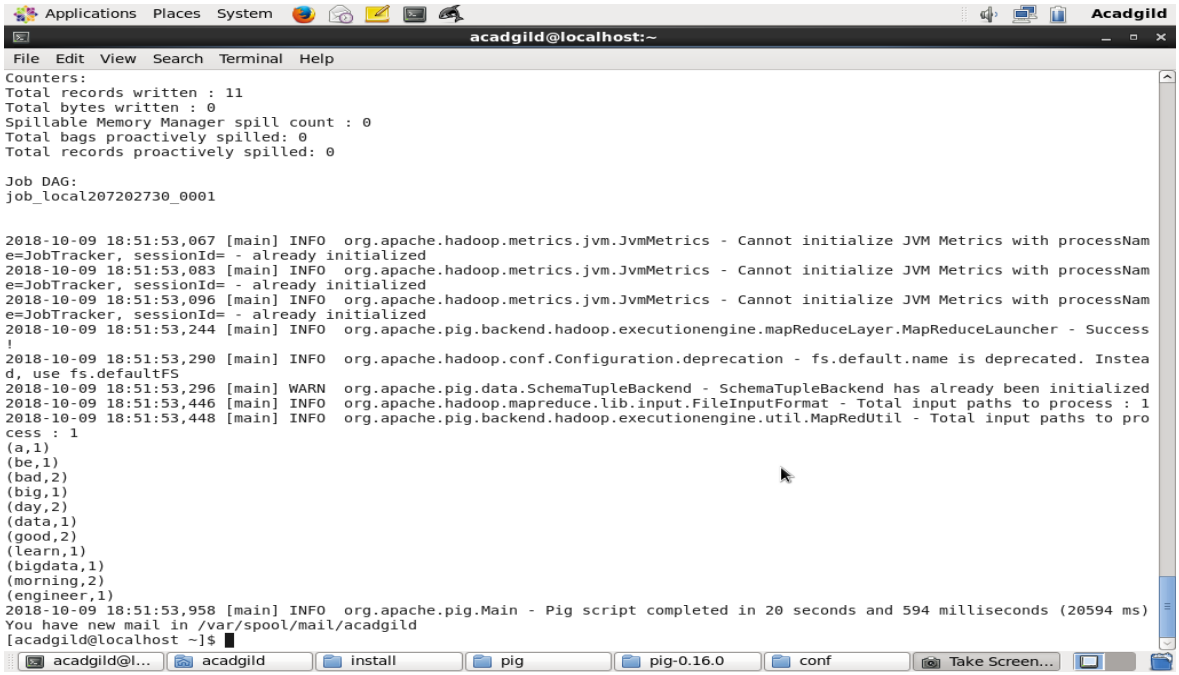
Task 1:

Write a program to implement word count using PIG.

Code:

```
lines = LOAD '/home/acadgild/Desktop/wordcount.txt' AS (line:chararray);
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;
grouped = GROUP words BY word;
wordcount = FOREACH grouped GENERATE group, COUNT(words);
DUMP wordcount;
```

Output:



Explanation:

- Text file was read into variable lines using default delimiter as comma (,).
- Tokenized lines into words based on delimiter.
- Grouped the whole set of words based on each word.
- Taken count of each word in set.

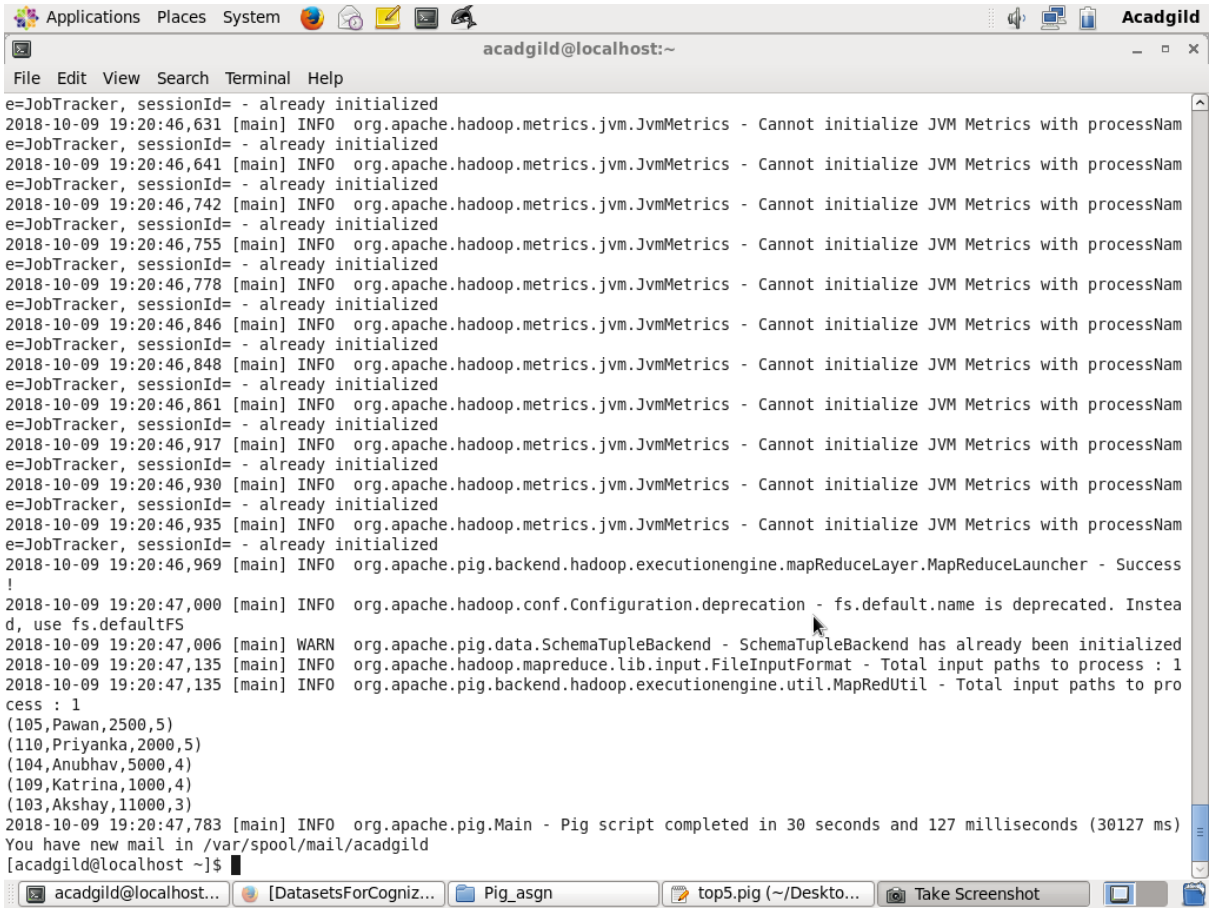
- Dumped the result set.

Task 2:

2(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)

```
emp_details = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_details.txt' using PigStorage(',')
as(emp_id:int,ename:chararray,sal:int,ranking:int);

sort_by_rank_ename = ORDER emp_details by ranking DESC,ename ASC;

top5 = LIMIT sort_by_rank_ename 5;

DUMP top5;
```

Output:



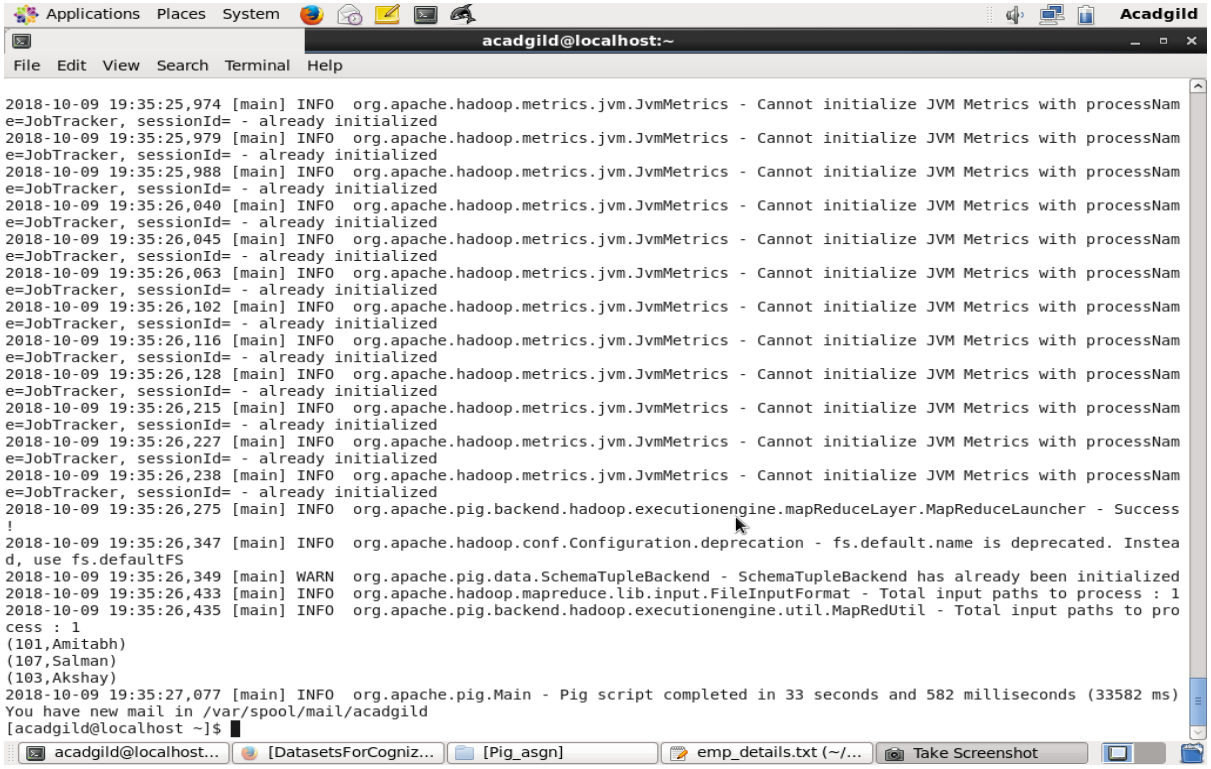
Explanation:

- Read emp details into PIG variable.
- Done double sorting – First on rankings DESC and then by ename ASC.
- Fetched top 5 by using limit operator.
- Dumped the result.

2(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)

```
emp_details = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_details.txt' using PigStorage(',')
as(emp_id:int,ename:chararray,sal:int,ranking:int);

odd_emp_id = FILTER emp_details by emp_id%2!=0;

sort_by_sal_ename = ORDER odd_emp_id by sal DESC,ename ASC;

top3 = LIMIT sort_by_sal_ename 3;

res = FOREACH top3 GENERATE $0,$1;

DUMP res;
```

Output:



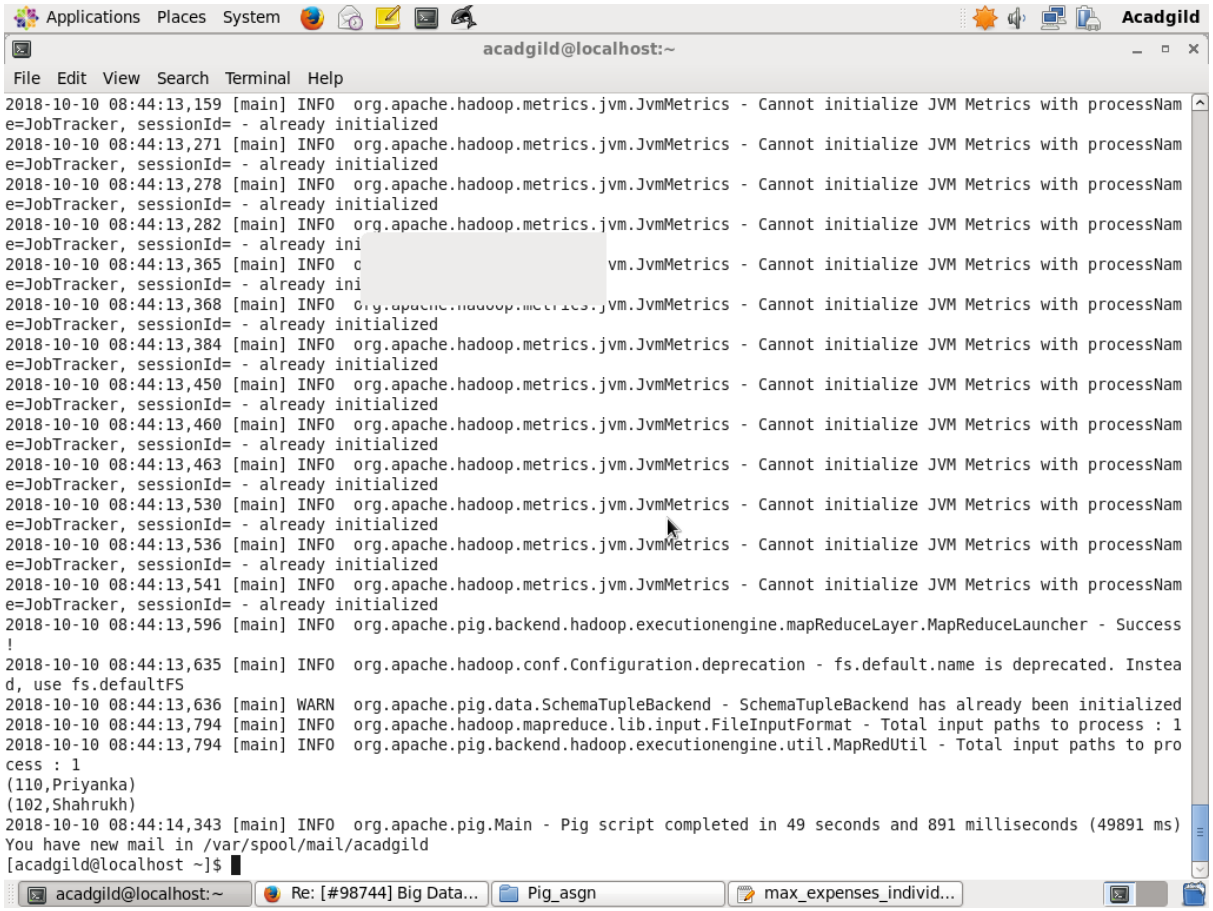
Explanation:

- Load emp details into a PIG variable.
- Filter data set with only odd emp id's using FILTER clause emp_id%2!=0.
- Done double sorting – First on sal DESC and then by ename ASC.
- Fetched top 3 by using limit operator.
- Fetched empid and ename out of the result set using FOREACH.
- Dumped the result.

2(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)

```
emp_expenses = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_expenses.txt' using
PigStorage('\t') as(emp_id:int,expenses:int);

emp_details = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_details.txt' using
PigStorage(',') as(emp_id:int,ename:chararray,sal:int,ranking:int);

grp_all = GROUP emp_expenses all;

max_exp = FOREACH grp_all GENERATE MAX(emp_expenses.expenses);

join_res = JOIN emp_expenses by expenses,max_exp by $0;
```

Output:



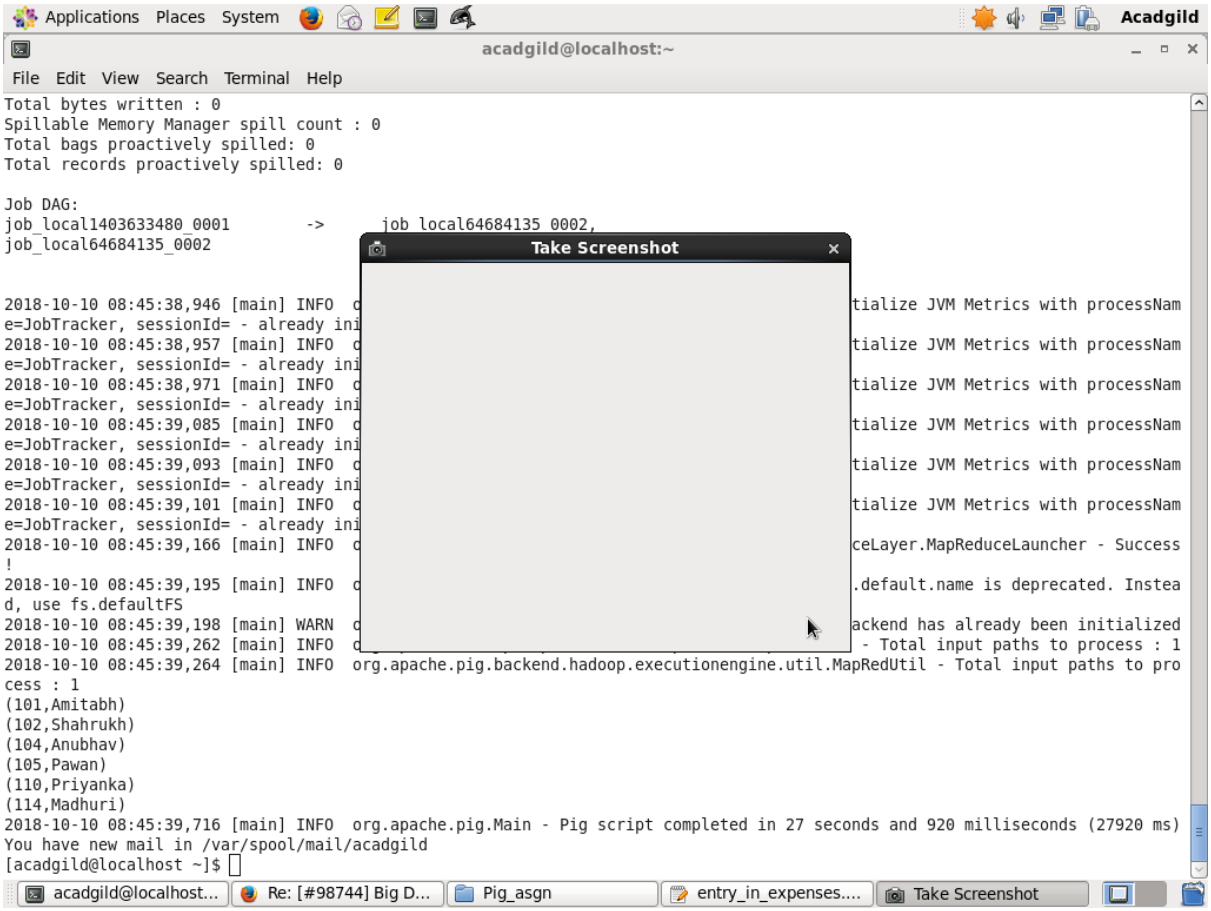
Explantion:

- Read emp details and expenses into two different PIG variables.
- Group emp expenses all.
- Find MAX of expenses out of above result set.
- Join MAX(expense) with emp expenses data set to get emp id.
- Join emp id yielded in above step with emp details.
- Sort data using ename.
- Dumped the result.

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

```
emp_expenses = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_expenses.txt' using PigStorage('\t') as(emp_id:int,expenses:int);

emp_details = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_details.txt' using PigStorage(',') as(emp_id:int,ename:chararray,sal:int,ranking:int);

emp_in_exp = FOREACH emp_expenses GENERATE emp_id;

dist_emp = DISTINCT emp_in_exp;

join_left_outer = JOIN  emp_details by emp_id LEFT OUTER,dist_emp by $0;

res = FILTER join_left_outer by $4 IS NOT NULL;

res1 = FOREACH res GENERATE $0,$1;

DUMP res1;
```

Output:



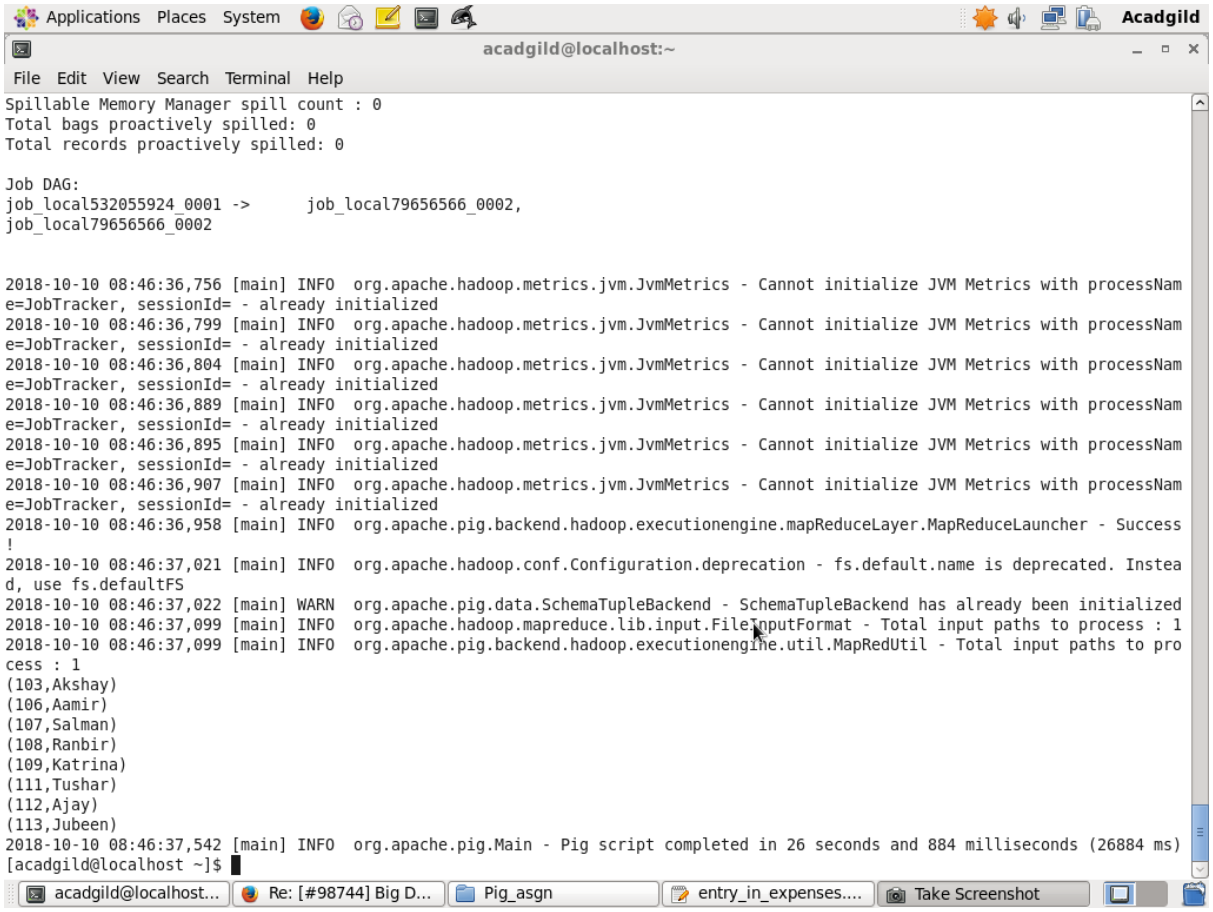
Explanation:

- Load expenses and details into two different PIG variables.
- As there are duplicates in expenses find out distinct entries in expenses using DISTINCT.
- Do a left outer join between emp details and distinct employees on emp id.
- Filter data set whose emp id from dist emp(dataset) is NOT NULL.
- Fetch empid and ename.
- Dumped the result.

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

```
emp_expenses = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_expenses.txt' using PigStorage('\t') as(emp_id:int,expenses:int);

emp_details = LOAD '/home/acadgild/Desktop/Pig_asgn/emp_details.txt' using PigStorage(',') as(emp_id:int,ename:chararray,sal:int,ranking:int);

emp_in_exp = FOREACH emp_expenses GENERATE emp_id;

dist_emp = DISTINCT emp_in_exp;

join_left_outer = JOIN  emp_details by emp_id LEFT OUTER,dist_emp by $0;

res = FILTER join_left_outer by $4 IS NULL;

res1 = FOREACH res GENERATE $0,$1;

DUMP res1;
```

Output:



Explanation:

- Load expenses and details into two different PIG variables.
- As there are duplicates in expenses find out distinct entries in expenses using DISTINCT.
- Do a left outer join between emp details and distinct employees on emp id.
- Filter data set whose emp id from dist emp(dataset) is NULL.
- Fetch empid and ename.
- Dumped the result.

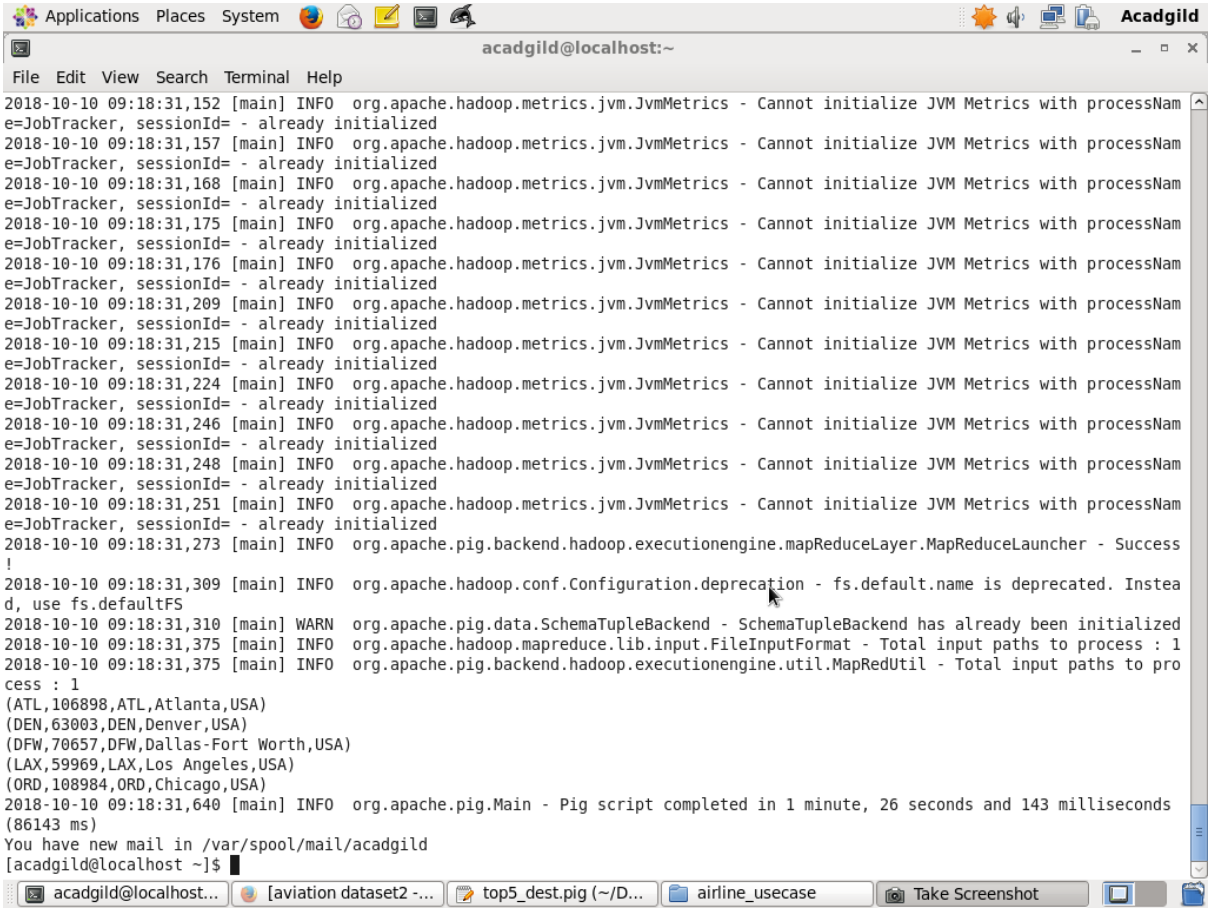
Task 3:

Implement the use case present in below blog link and share the complete steps along with screenshot(s) from your end.

a)Find out the top 5 most visited destinations.

```
REGISTER '/home/acadgild/Desktop/airline_usecase/piggybank.jar';

delayed_flights = load '/home/acadgild/Desktop/airline_usecase/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

src_dest = foreach delayed_flights generate (int)$1 as year, (int)$10 as flight_num, (chararray)$17 as
origin,(chararray) $18 as dest;

dest_not_null = filter src_dest by dest is not null;

group_dest = group dest_not_null by dest;

count_dest = foreach group_dest generate group, COUNT(dest_not_null.dest);

sort_dest = order count_dest by $1 DESC;

top5_dest = LIMIT sort_dest 5;

airports = load '/home/acadgild/Desktop/airline_usecase/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

airports_dest = foreach airports generate (chararray)$0 as dest, (chararray)$2 as city, (chararray)$4 as
country;

joined_table = join top5_dest by $0, airports_dest by dest;

dump joined_table;
```

Output:



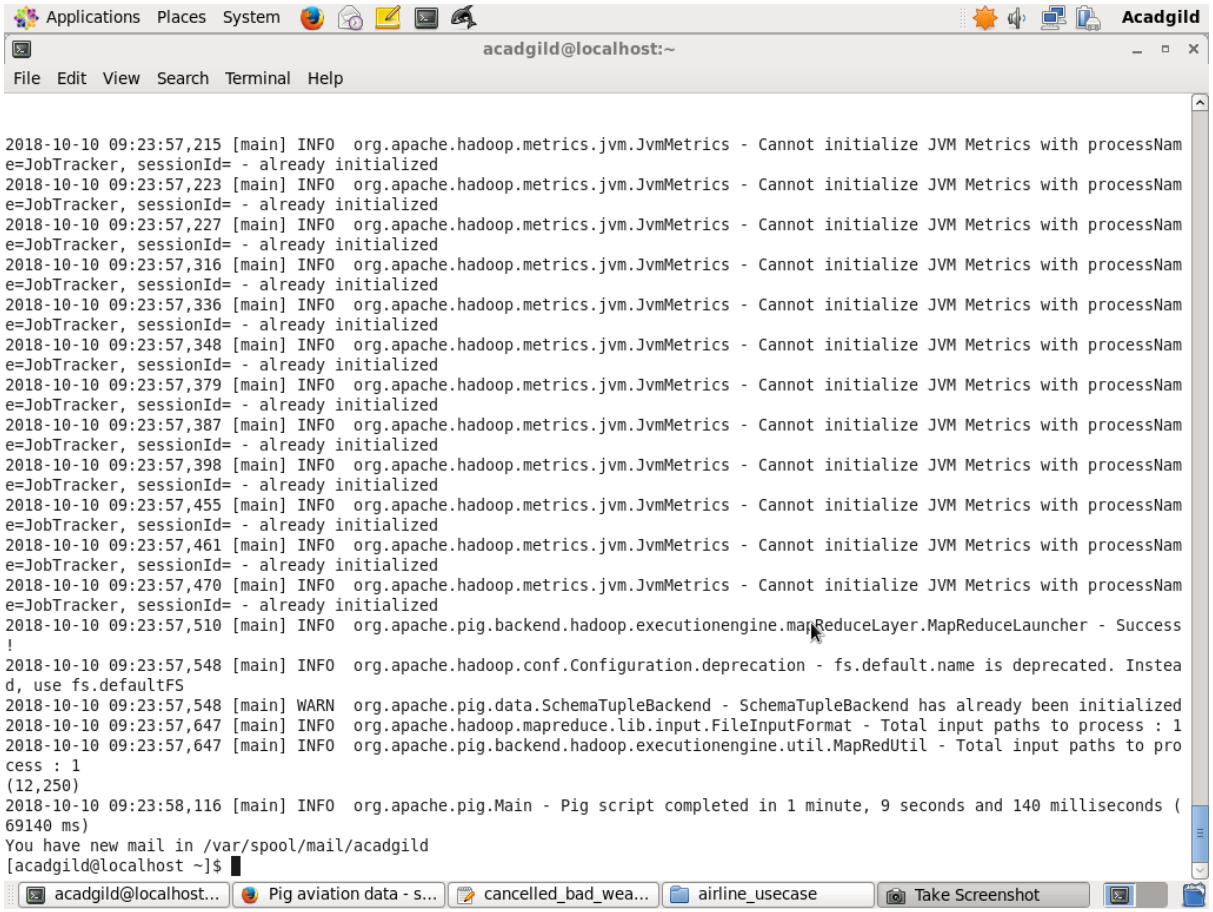
Explanation:

- Register piggyback.jar to load CSV data.
- Load delay flights and airports data into PIG variables.
- Fetch destination details which are not null.
- Group by destination.
- Do count on each destination.
- Sort them in descending order.
- Pull top 5 results.
- Perform a join between destination and airports and display the results.

b) Which month has seen the most number of cancellations due to bad weather?

```
REGISTER '/home/acadgild/Desktop/airline_usecase/piggybank.jar';

A = load '/home/acadgild/Desktop/airline_usecase/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B = foreach A generate (int)$2 as month,(int)$10 as flight_num,(int)$22 as cancelled,(chararray)$23 as
cancel_code;

C = filter B by cancelled == 1 AND cancel_code =='B';

D = group C by month;

E = foreach D generate group, COUNT(C.cancelled);

F= order E by $1 DESC;

Result = limit F 1;

dump Result;
```

Output:



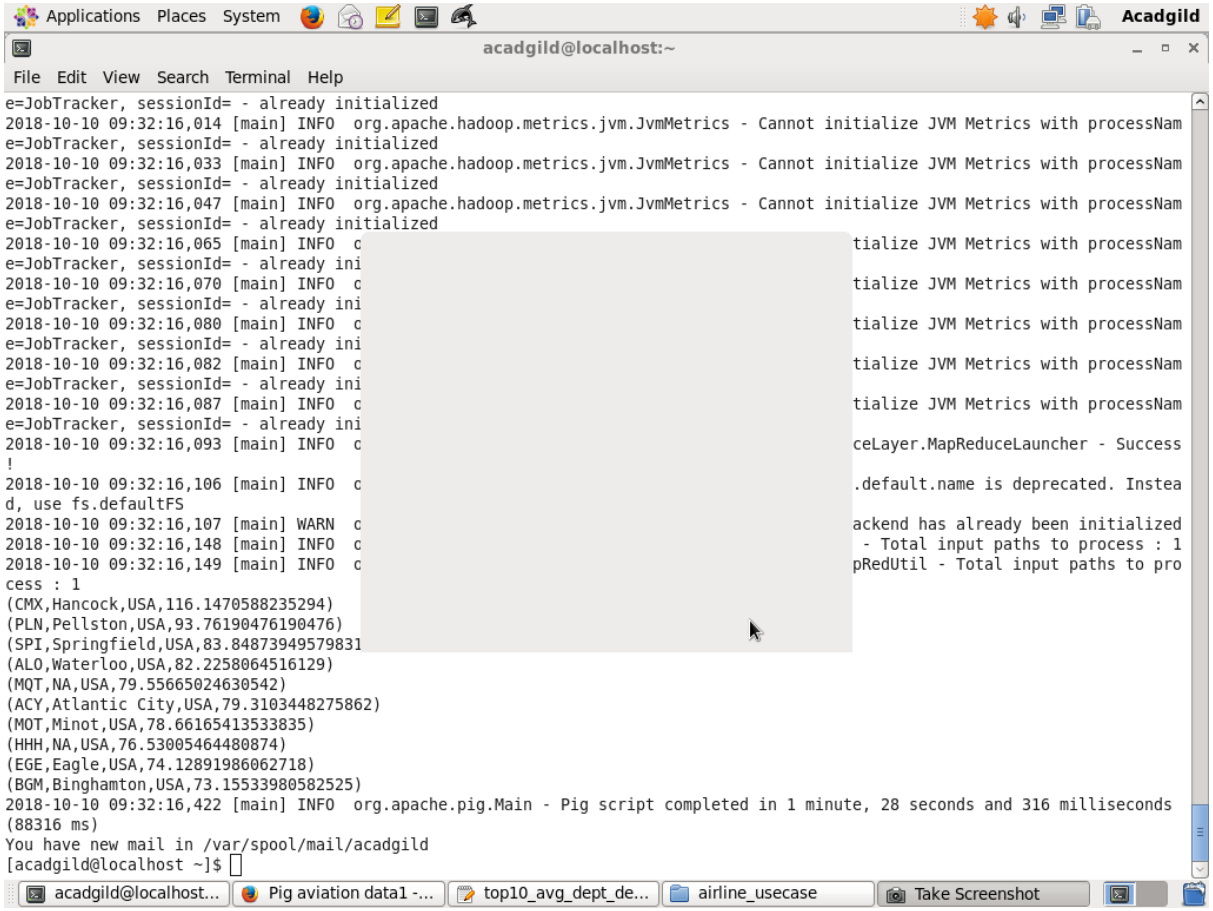
Explanation:

- Load delayed flights data.
- Fetch required attributes.
- Filter data on cancelled ==1 and cancel code =’B’
- Group data by month
- Calculate count and do a sort on values
- Limit the top row.
- Dump the result.

c) **Top ten origins with the highest AVG departure delay**

```
REGISTER '/home/acadgild/Desktop/airline_usecase/piggybank.jar';

A = load '/home/acadgild/Desktop/airline_usecase/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B1 = foreach A generate (int)$16 as dep_delay, (chararray)$17 as origin;

C1 = filter B1 by (dep_delay is not null) AND (origin is not null);

D1 = group C1 by origin;

E1 = foreach D1 generate group, AVG(C1.dep_delay);

Result = order E1 by $1 DESC;

Top_ten = limit Result 10;

Lookup = load '/home/acadgild/Desktop/airline_usecase/airports.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

Lookup1 = foreach Lookup generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as
country;

Joined = join Lookup1 by origin, Top_ten by $0;

Final = foreach Joined generate $0,$1,$2,$4;

Final_Result = ORDER Final by $3 DESC;

dump Final_Result;
```

Output:



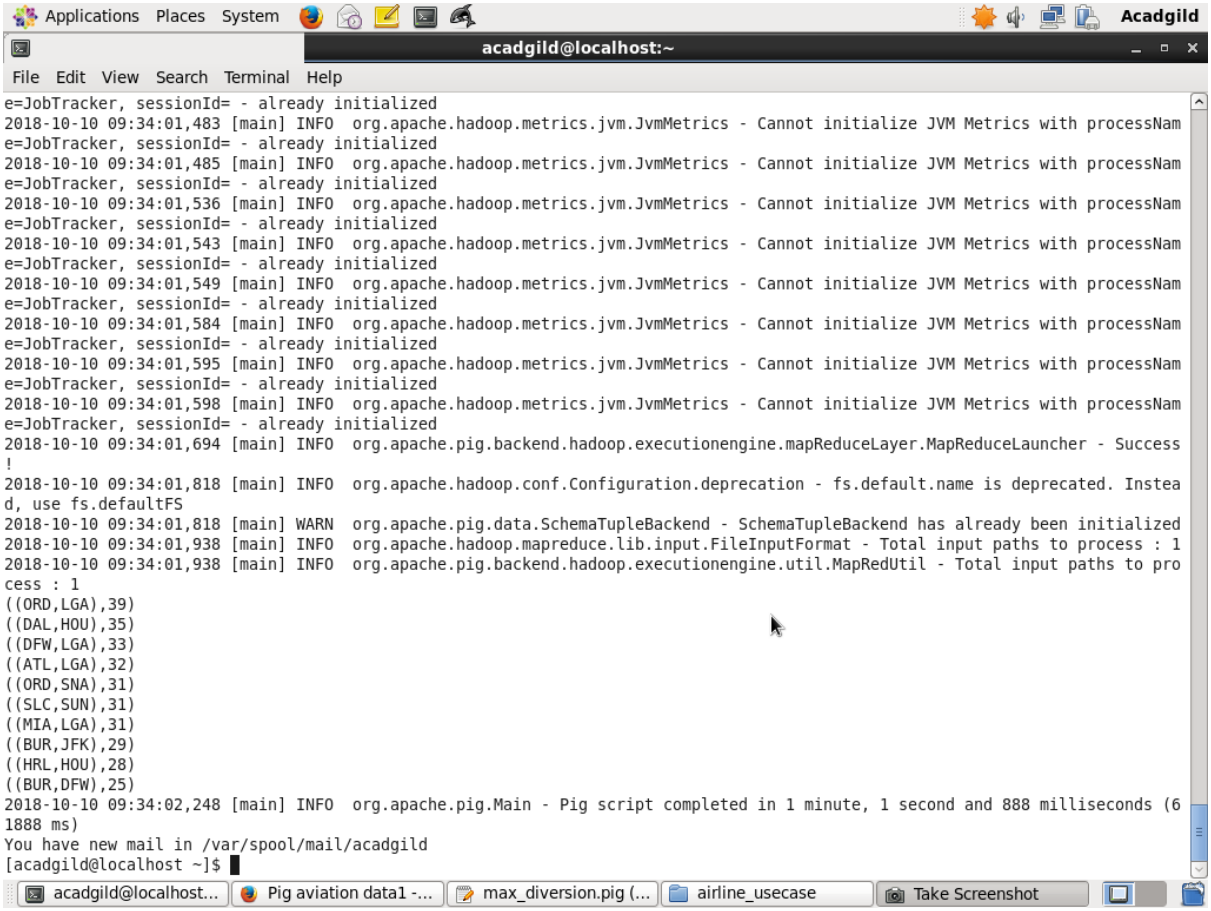
Explanation:

- Load Delayed flights data.
- Fetch departure delay and origin.
- Filter data set which is not null.
- Group by origin
- Calculated AVG of dep delay against each origin.
- Sort by avg dep delay DESC
- Fetch top 10
- Join with airports based on origin to fetch airport details.
- Dump the results.

d) Which route (origin & destination) has seen the maximum diversion?

```
REGISTER '/home/acadgild/Desktop/airline_usecase/piggybank.jar';

A = load '/home/acadgild/Desktop/airline_usecase/DelayedFlights.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','UNIX','SKIP_INPUT_HEADER');

B = FOREACH A GENERATE (chararray)$17 as origin, (chararray)$18 as dest, (int)$24 as diversion;

C = FILTER B BY (origin is not null) AND (dest is not null) AND (diversion == 1);

D = GROUP C by (origin,dest);

E = FOREACH D generate group, COUNT(C.diversion);

F = ORDER E BY $1 DESC;

Result = limit F 10;

dump Result;
```

Output:



Explanation:

- Load Filghts delay data.
- Filter by diversion ==1
- Group by data
- Calculate count
- Order data
- Limit top 10
- Dump the result.
-