**Problem Statement**

**Implement the use case present in below blog link**

**https://acadgild.com/blog/pig-use-case-pokemon-data-analysis/**

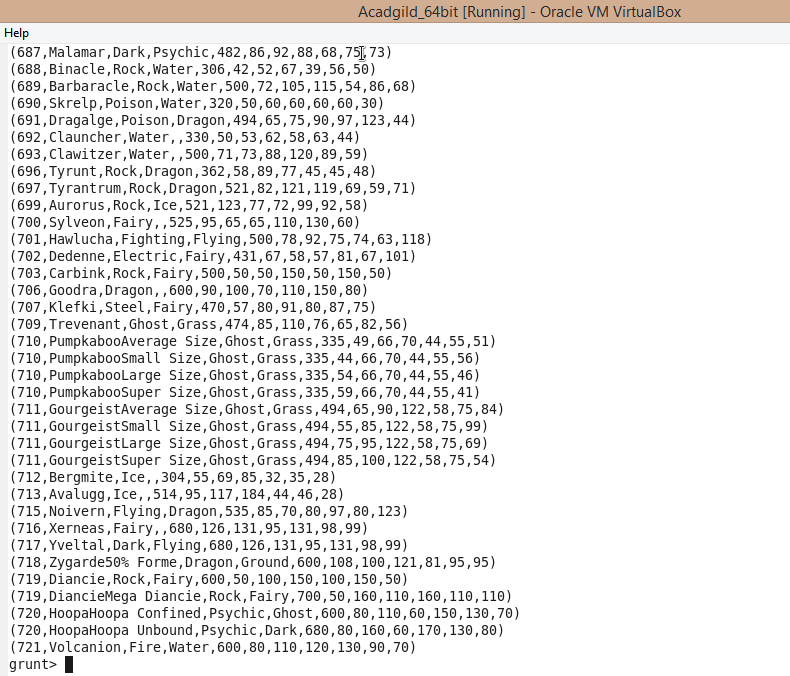
**Problem1:**

Find the list of players that have been selected in the qualifying round (DEFENCE>55).

Solution:

|  |
| --- |
| --Load Pokemon.csv Data  pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefenct:int,speed:int); |
| -- filter qualifying data |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ;  -- display results |
|  |
| dump qualifying\_data; |

Output:



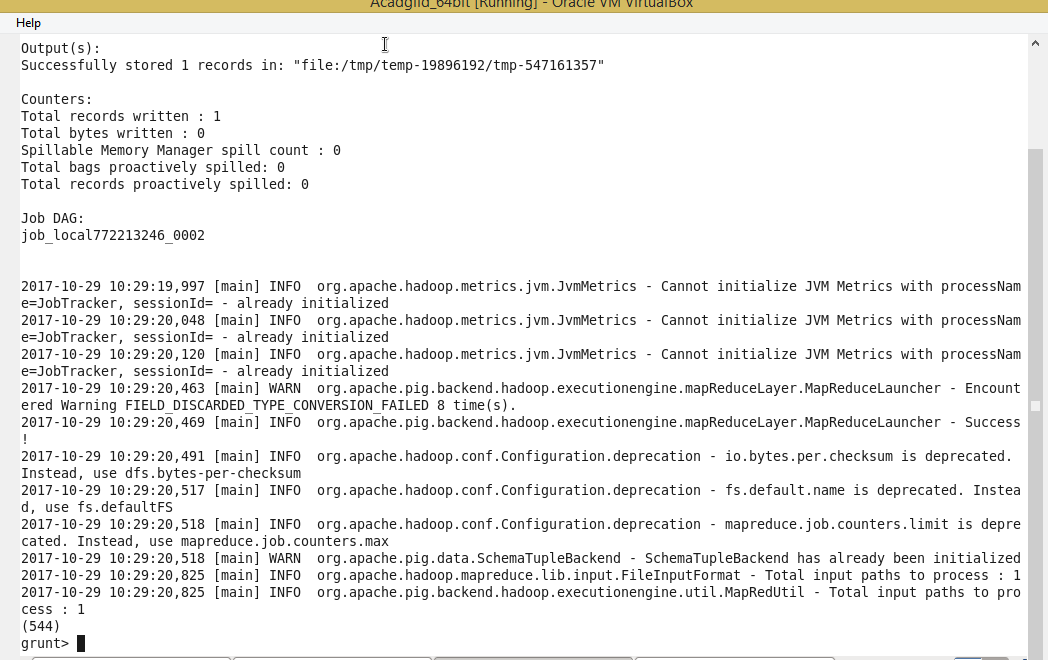
**Problem2:**

State the number of players taking part in the competition after getting selected in the qualifying round.

Solution:

|  |
| --- |
| -- Load Data |
| pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefenct:int,speed:int); |
|  |
| -- filter qualifying data |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ; |
|  |
| -- group the qualifying data |
| group\_qualifying\_data = GROUP qualifying\_data ALL ; |
|  |
| -- get count |
| count\_qualifying\_data = foreach group\_qualifying\_data GENERATE COUNT(qualifying\_data) ; |
|  |
| -- display results |
| dump count\_qualifying\_data ; |

Output:



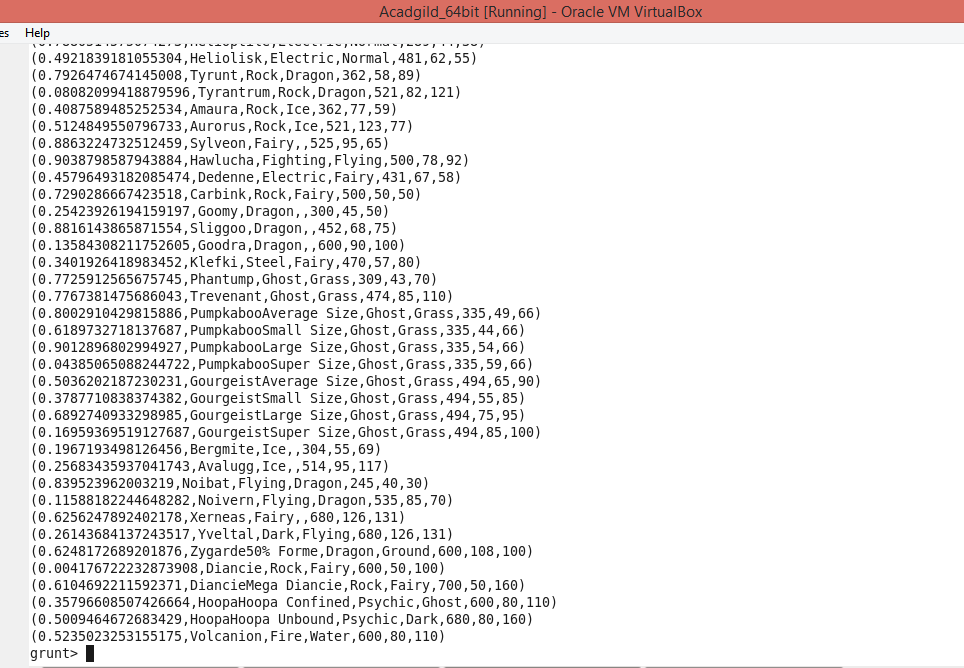
**Problem3:**

Using random() generate random numbers for each Pokémon on the selected list.

Solution:

|  |
| --- |
| -- Load Data |
| pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefence:int,speed:int);  -- filter qualifying data |
|  |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ; |
|  |
| -- generate random for relation 1 |
| random\_include1 = foreach qualifying\_data GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,splatck,spldefence,speed; |
|  |
| -- display result for random relation 1 |
| dump random\_include1; |

Output:



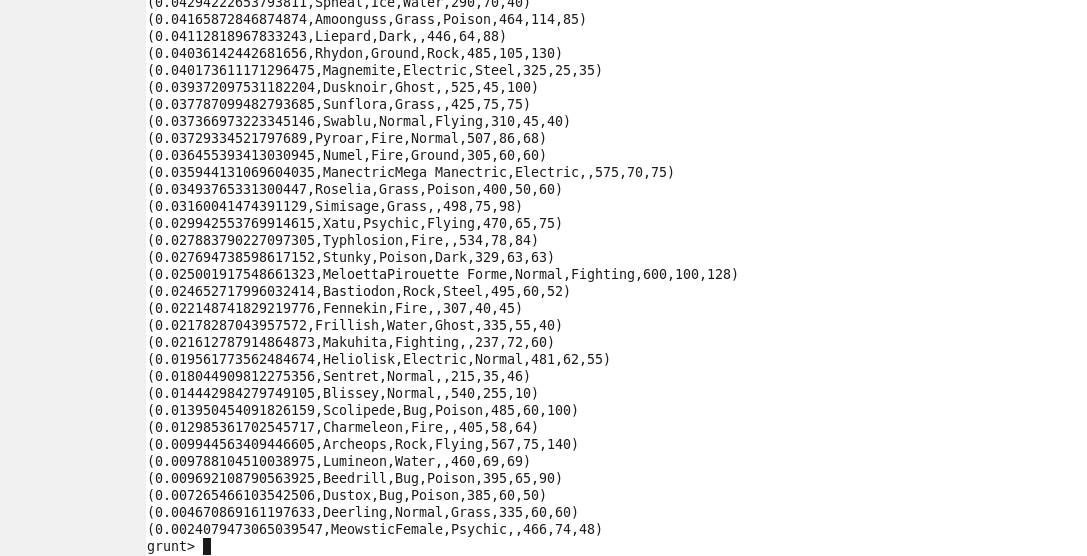
**Problem4:**

Arrange the new list in a descending order according to a column randomly.

Solution:

|  |
| --- |
| -- Load Data |
| pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefence:int,speed:int);  --Filter qualifying data |
|  |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ; |
|  |
| -- generate random for relation 1 |
| random\_include1 = foreach qualifying\_data GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,splatck,spldefence,speed; |
|  |
|  |
| -- sort by random function relation 1 |
| random1\_desending = ORDER random\_include1 BY $0 DESC;  -- displaying result of random function relation 1  dump random1\_desending; |

Output:



**Problem5:**

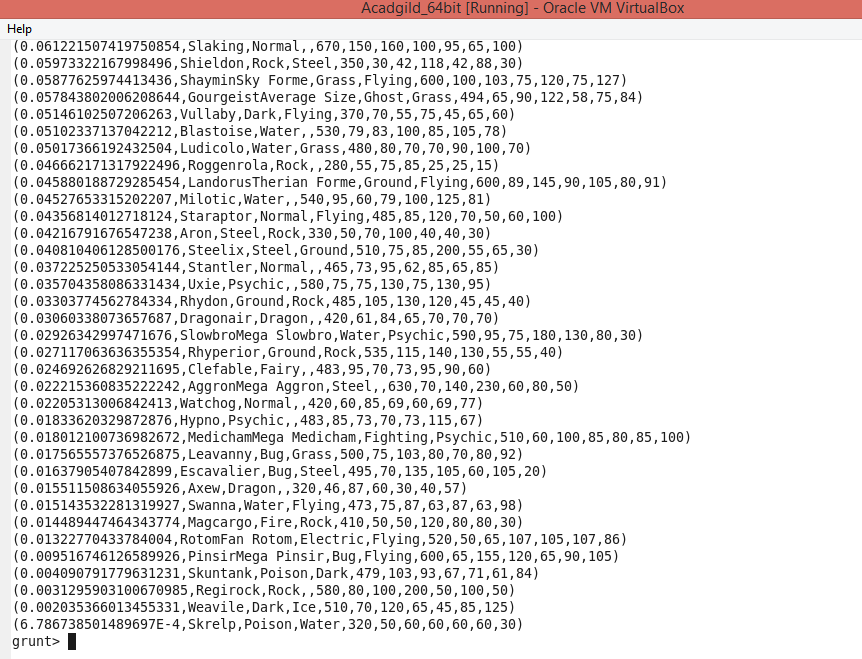
Now on a new relation again associate random numbers for each Pokémon and arrange in descending order according to

column random.

Solution:

|  |
| --- |
| -- Load Data |
| pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefence:int,speed:int); |
|  |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ; |
|  |
| -- generate random for relation 2 |
| random\_include2 = foreach qualifying\_data GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,splatck,spldefence,speed; |
|  |
| -- sort by random function relation 2 |
| random2\_desending = ORDER random\_include2 BY $0 DESC; |
|  |
| -- display results of random function for relation 2  dump random2\_desending; |
|  |

Output:



**Problem6:**

From the two different descending lists of random Pokémons, select the top 5 Pokémons for 2 different players.

Solution:

--Limit random function relation1

limit\_random1 = LIMIT random1\_desending 5;

--display result for random function for relation1

dump limit\_random1;

--Limit random function relation2

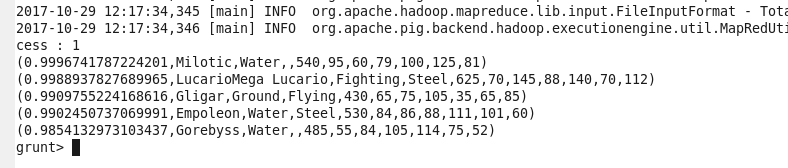
limit\_random2 = LIMIT random1\_desending 5;

--display result for random function for relation1

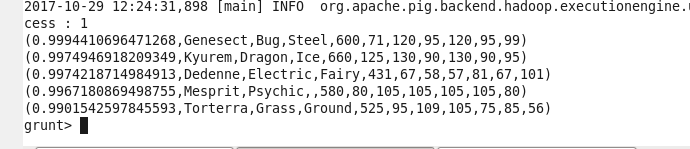
dump limit\_random2;

Output:

Result for random relation1---

****

Result for random relation2---



**Problem7:**

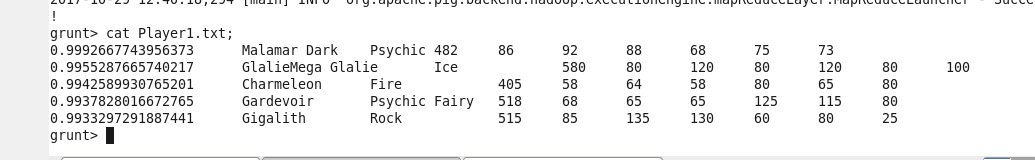
Store the data on a local drive to announce for the final match. By the name player1 and player2 (only show the NAME and HP).

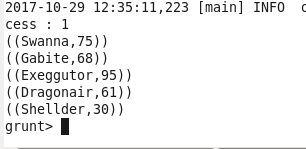
Solution:

|  |
| --- |
| -- Load Data |
| pokemon\_data = LOAD '/home/acadgild/ankita/Assignment5\_3/Pokemon' USING PigStorage(',') AS(Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,splatck:int,spldefence:int,speed:int); |
|  |
| qualifying\_data = FILTER pokemon\_data by Defense>55 ; |
|  |
| -- generate random for relation 1 |
| random\_include1 = foreach qualifying\_data GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,splatck,spldefence,speed; |
|  |
| -- sort by random function relation 1 |
| random1\_desending = ORDER random\_include1 BY $0 DESC; |
|  |
| -- generate random for relation 2 |
| random\_include2 = foreach qualifying\_data GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,splatck,spldefence,speed; |
|  |
| -- sort by random function relation 2 |
| random2\_desending = ORDER random\_include2 BY $0 DESC; |
|  |
| -- top 5 from relation 1 |
| Limit\_random1 = LIMIT random1\_desending 5 ; |
|  |
| -- top 5 from relation 2 |
| limit\_random2 = LIMIT random2\_desending 5 ; |
|  |
| -- filter name from relation 1 |
| filter\_only\_name1 = foreach limit\_random1 Generate ($1,HP); |
| -- filter name from relation 2 |
| filter\_only\_name2 = foreach limit\_random2 Generate ($1,HP); |
|  |
| -- store relation 1 data to local file |
| STORE limit\_random1 INTO '/home/acadgild/ankita/Assignment5\_3/player1.txt'; |
|  |
| -- store relation 2 data to local file |
| STORE limit\_random2 INTO '/home/acadgild/ankita/Assignment5\_3/player2.txt'; |

Output:

Result of Player1.txt—





Result of Player2.txt—

