Ques 1:

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

Solution:

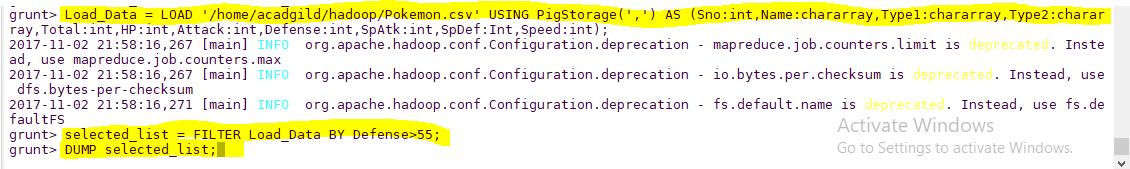
Code:

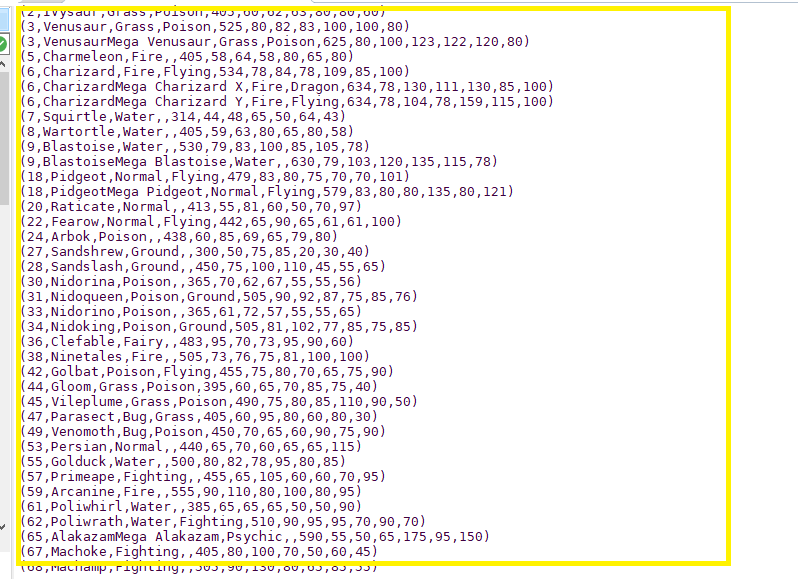
Load\_Data = LOAD '/home/acadgild/hadoop/Pokemon.csv' USING PigStorage(',') AS (Sno:int,Name:chararray,Type1:chararray,Type2:chararray,Total:int,HP:int,Attack:int,Defense:int,SpAtk:int,SpDef:Int,Speed:int);

selected\_list = FILTER Load\_Data BY Defense>55;

DUMP selected\_list;

Screen-Shot





Ques 2:

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

Solution:

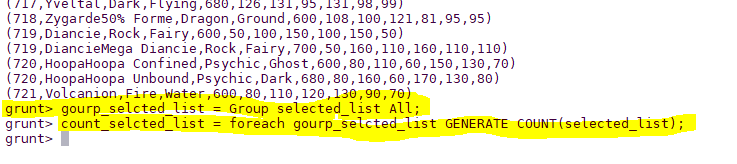
Code:

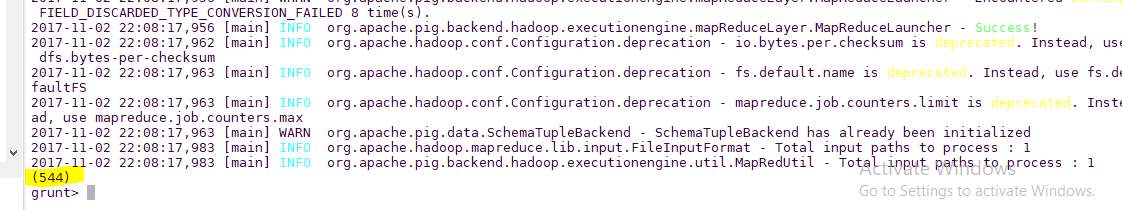
gourp\_selcted\_list = Group selected\_list All;

count\_selcted\_list = foreach gourp\_selcted\_list GENERATE COUNT(selected\_list);

DUMP count\_selcted\_list;

Code:





So, All the 544 players taking part will be alphabetically arranged and two teams of 5 Pokémons need to be extracted out randomly from the earlier list.

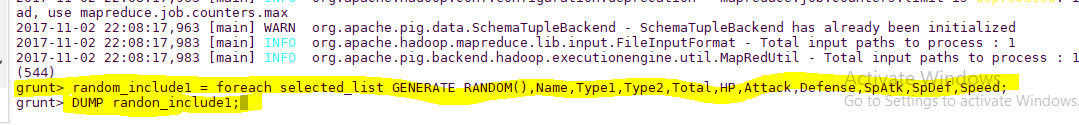
Seems like,this way we will have 2 lists containing 5 Pokémon each so to fight each other.

Ques 3:

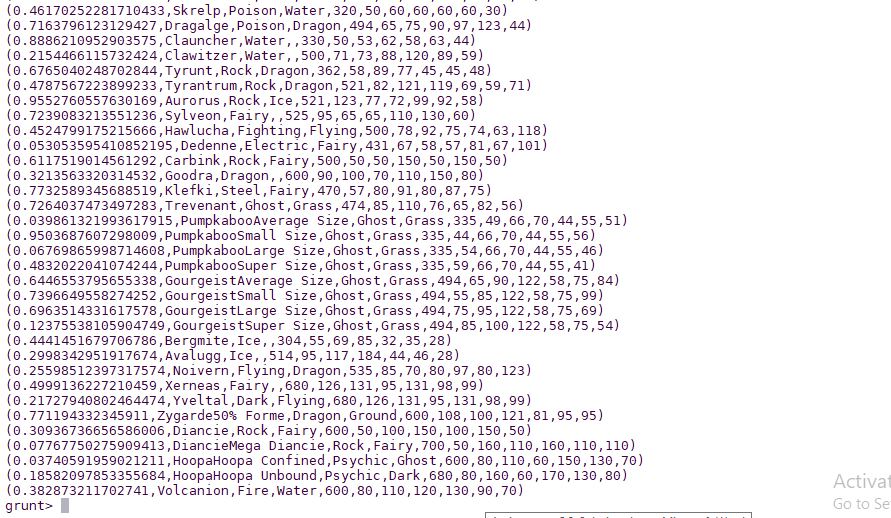
Using random() generate random numbers for each Pokémon on the selected list.

random\_include1 = foreach selected\_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAtk,SpDef,Speed;

DUMP random\_include1;



Hence sample for the list after adding random numbers:



Ques 4:

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

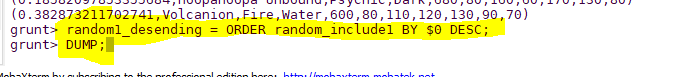
**Explanation**: This will give us consequently a layer arranged to pick the random list which 1st player will

choose.

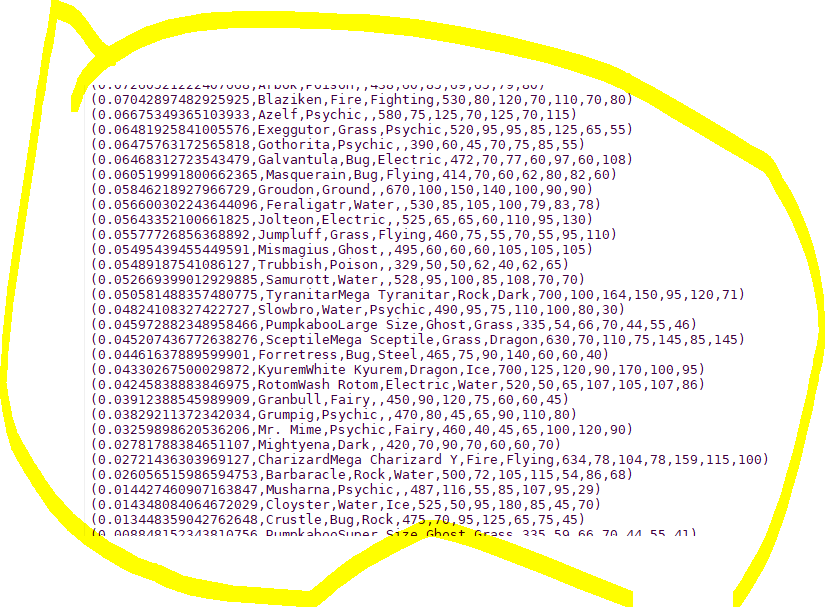
Code:

random1\_desending = ORDER random\_include1 BY $0 DESC;

dump;



Hence the sample for the list after the query.



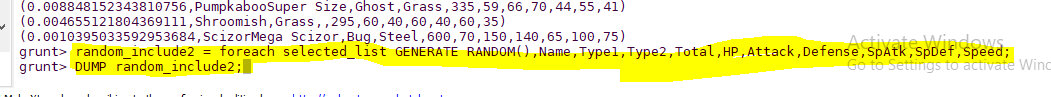
Yet we want 1 more list with random arrangements of Pokémons which will be therefore chosen by the 2nd player later on.

Ques 5: Now on a new relation again associate random numbers for each Pokémon and arrange in descending order according to column random.

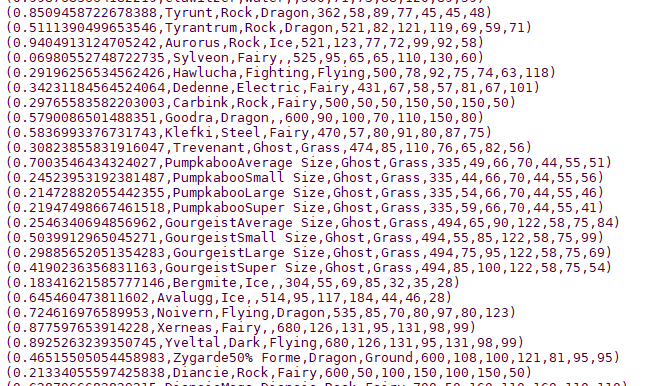
**Explanation**: We will be repeating above two steps again to form the 2nd list.

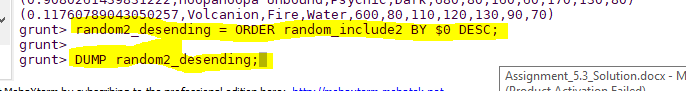
random\_include2 = foreach selected\_list GENERATE RANDOM(),Name,Type1,Type2,Total,HP,Attack,Defense,SpAtk,SpDef,Speed;

random2\_desending = ORDER random\_include2 BY $0 DESC;

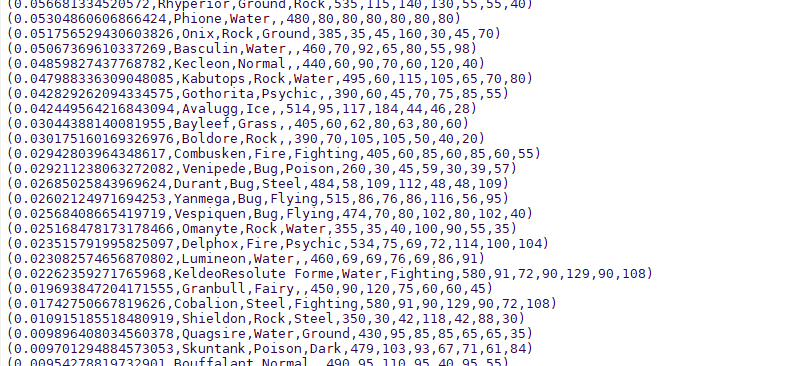


Hence sample for the list.





Hence sample for the list.



Now, especially relevant selecting the top 5.

Ques:

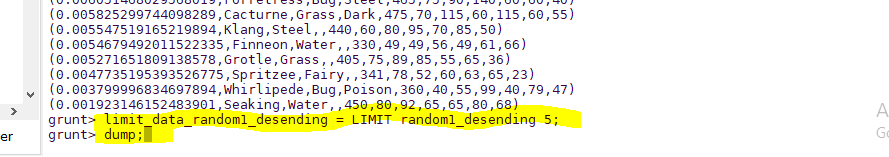
 From the two different descending lists of random Pokémons, select the top 5 Pokémons for 2 different players.

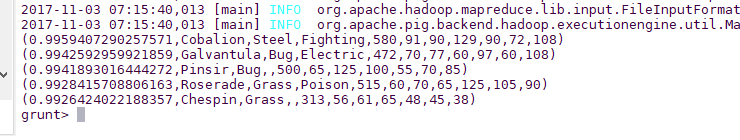
limit\_data\_random1\_desending = LIMIT random1\_desending 5;

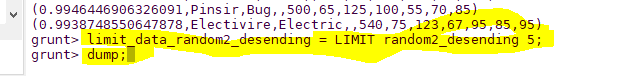
dump;

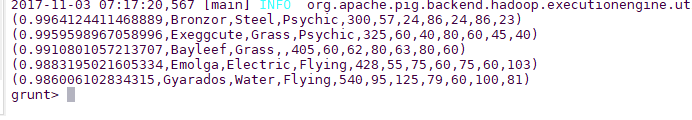
limit\_data\_random2\_desending = LIMIT random2\_desending 5;

dump;





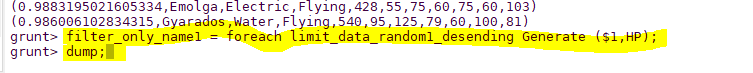




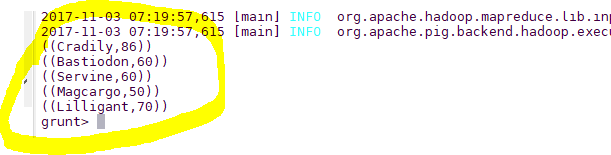
Ques:

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).

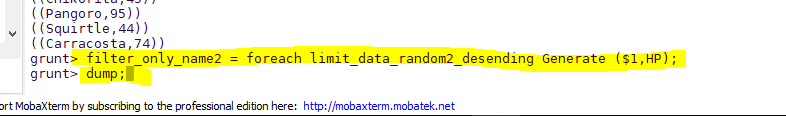
filter\_only\_name1 = foreach limit\_data\_random1\_desending Generate ($1,HP);



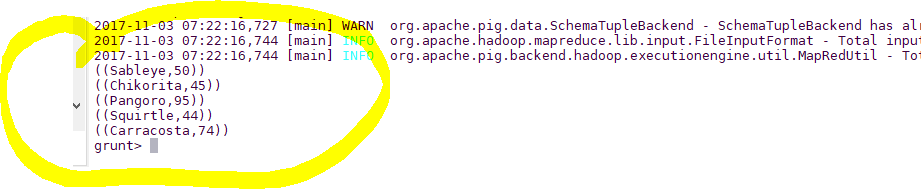
Since for Player1 we have:



filter\_only\_name2 = foreach limit\_data\_random2\_desending Generate ($1,HP);



Since for Player2 we have:



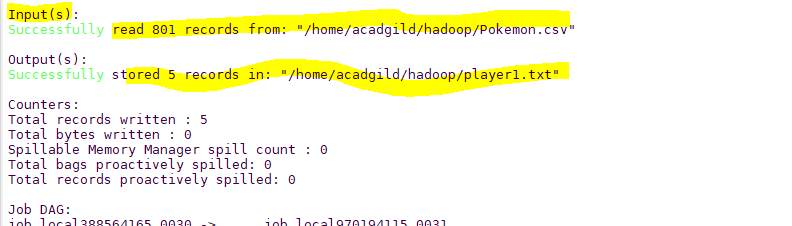
Therefore querying is over using some simple pre-defined functions to get 2 sets of 5 Pokémons, which get select randomly.

In conclusion, let’s store this result in our local system  so we can use it as input to our next blog. Especially relevant where we will see UDF using PIG and calculations will be done through user-defined formulas.

STORE limit\_data\_random1\_desending INTO ‘/home/acadgild/hadoop/player1.txt’;

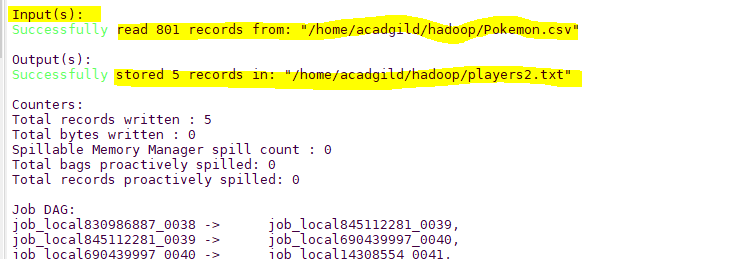


As a Result:





As a Result:



As a result, the Pokémons for both players got selected . This player will be fighting consequently in the Finals with their respective Pokémons assigned.