DCBD Assignment 4

Himanshu, MDS202327

1. Assume the games.csv is in the present working directory.

A)

```
P= LOAD ’games.csv’ USING PigStorage(’,’) AS (id:chararray,rated:boolean,created_at:chararray,
last_move_at:chararray,turns:int,victory_status:chararray,winner:chararray,increment_code:char
array, white_id:chararray,white_rating:int,black_id:chararray,black_rating:int,moves:chararray,
opening_eco:chararray,opening_name:chararray,opening_ply:int);

# P stores the relation

Q = FILTER A BY winner == ’white’;

# Q selects those rows of A where the winner is white

STORE Q INTO ’partA.txt’ USING PigStorage(’,’);

# Q stored in a file partA.txt
```

B)

```
P = LOAD ’games.csv’ USING PigStorage(’,’) AS (id:chararray,rated:boolean,created_at:chararray,
last_move_at:chararray,turns:int,victory_status:chararray,winner:chararray,increment_code:char
array, white_id:chararray,white_rating:int,black_id:chararray,black_rating:int,moves:chararray,
opening_eco:chararray,opening_name:chararray,opening_ply:int);

# P relation made exactly as above

Q = FILTER P BY winner == ’white’;

# Q relation is again same as above, having only winner as white

R = GROUP Q BY white_id;

# Q is grouped by the white_id

S = FOREACH R GENERATE group AS white id, AVG(Q.white_rating) AS average rating;

# For every winning player (white_id), we store the white_id (written as group) and average of
that player’s ratings

DUMP S;

# S printed, as required
```

C)

```
P = LOAD ’games.csv’ USING PigStorage(’,’) AS (id:chararray,rated:boolean,created_at:chararray, last_move_at:chararray,turns:int,victory_status:chararray,winner:chararray,increment_code:chararray, white_id:chararray,white_rating:int,black_id:chararray,black_rating:int,moves:chararray, opening_eco:chararray,opening_name:chararray,opening_ply:int);

# Again P relation as same

Q = FILTER P BY turns ¿ 100;

# This time, Q has those rows of P where turns were more than 100

grouped_dummy = GROUP Q ALL;

# Here, grouped_dummy is a group with only one element as Q

row_count = FOREACH grouped_dummy GENERATE COUNT(Q) AS row count;

# Then for this element of group (the only element), we counted its number of rows

DUMP row_count;

# Number of rows printed, as required
```

2.

Let’s assume that we have a function f, that takes in a word and gives the number of letters in that word. Hence we have

f(’hello’) = 5 and f(’cat’) = 3

Next, in the map framework we tokenize each text into words and then for each word, we make a key-value pair, where the key will be the number of letters in that word and value will be 1.

Word x → ⟨f(x), 1⟩

Then in shuffling and sorting phase these key-value pairs will be sorted by their keys.

In the reduce framework, for each unique key separately, we add their values together. Hence for every number k as a key, it’s value will be the number of map-reduce pairs that were present with the key k.

Finally, for each key-value pair ⟨x, y⟩, we print

x, y

line by line.