Contents

[Query1.pig 2](#_Toc7095225)

[Query\_1a output test 1 Input: 430 8](#_Toc7095226)

[Query\_1a output test 2 input: 405 8](#_Toc7095227)

[Query\_1a output test 3 input: 919 9](#_Toc7095228)

[Query\_1b output: 10](#_Toc7095229)

[Query\_1c test 1 input: 666 11](#_Toc7095230)

[Query\_1c test 2 input: 333 11](#_Toc7095231)

[Query\_1c test 3 input: 233 12](#_Toc7095232)

[Task 2 13](#_Toc7095233)

[query2.PIG: 13](#_Toc7095234)

[New Rank Twitter Account 15](#_Toc7095235)

[Copy output data into file: HW4old\_twitter\_account\_rank.csv 15](#_Toc7095236)

[Find k\_percentile\_accounts.pig 16](#_Toc7095237)

[Task3 output test 1 Input: 0.3 18](#_Toc7095238)

[Task3 output test 2 Input: 0.0042 18](#_Toc7095239)

[Task3 output test 2 Input: 0.005 19](#_Toc7095240)

# Query1.pig

-- Format:

--{follower\_twitter\_account\_id: int,subject\_twitter\_account\_id: int}

follows\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-follows\_account.txt' using PigStorage(',') AS (follower\_twitter\_account\_id: int,subject\_twitter\_account\_id: int);

-- Format:

--{twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

twitter\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-twitter\_account.txt' USING PigStorage(',') AS (twitter\_account\_id: int, email\_address: chararray , phone\_number: chararray , user\_location: chararray , num\_tweets: int);

-- Format:

--{twitter\_account\_id: int,twitter\_account\_rank: float}

old\_twitter\_account\_rank = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-old\_twitter\_account\_rank.txt' using PigStorage(',') AS (twitter\_account\_id:int, twitter\_account\_rank:float);

-- Format:

--{email\_address: chararray,reputation: int,num\_questions: int}

stack\_overflow\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-stack\_overflow\_account.txt' using PigStorage(',') AS (email\_address:chararray, reputation: int, num\_questions: int);

--query\_1a 1st test

----------------------------------------------------------------------------------

-- Format

-- {twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

filter\_phone = FILTER twitter\_account by STARTSWITH(phone\_number, '430');

DESCRIBE filter\_phone;

-- Format

-- {group: chararray,filter\_phone: {(twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int)}}

grouped\_phone = GROUP filter\_phone BY SUBSTRING(phone\_number, 0, 3);

DESCRIBE grouped;

-- Format

-- {group: chararray,{(email\_address: chararray)}}

result\_a1 = FOREACH grouped\_phone GENERATE group, $1.email\_address;

DESCRIBE result\_a1;

STORE result\_a1 INTO query\_1a1;

dump result\_a1;

--query\_1a 2rd test

----------------------------------------------------------------------------------

-- Format

-- {twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

filter\_phone = FILTER twitter\_account by STARTSWITH(phone\_number, '405');

DESCRIBE filter\_phone;

-- Format

-- {group: chararray,filter\_phone: {(twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int)}}

grouped\_phone = GROUP filter\_phone BY SUBSTRING(phone\_number, 0, 3);

DESCRIBE grouped;

-- Format

-- {group: chararray,{(email\_address: chararray)}}

result\_a2 = FOREACH grouped\_phone GENERATE group, $1.email\_address;

DESCRIBE result\_a2;

STORE result\_a2 INTO query\_1a2;

dump result\_a2;

--query\_1a 3rd test

----------------------------------------------------------------------------------

-- Format

-- {twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

filter\_phone = FILTER twitter\_account by STARTSWITH(phone\_number, '555');

DESCRIBE filter\_phone;

-- Format

-- {group: chararray,filter\_phone: {(twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int)}}

grouped\_phone = GROUP filter\_phone BY SUBSTRING(phone\_number, 0, 3);

DESCRIBE grouped;

-- Format

-- {group: chararray,{(email\_address: chararray)}}

result\_a3 = FOREACH grouped\_phone GENERATE group, $1.email\_address;

DESCRIBE result\_a3;

STORE result\_a3 INTO query\_1a3;

dump result\_a3;

--query\_1b

----------------------------------------------------------------------------------

--Format

--{group: chararray,twitter\_account: {(twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int)}}

group\_city = GROUP twitter\_account BY user\_location;

--Format

--result\_b = FOREACH group\_city GENERATE group AS user\_location, COUNT($1) AS num\_acc;

result\_b = FOREACH group\_city GENERATE group AS user\_location, COUNT($1) AS num\_acc;

DESCRIBE result\_b;

STORE result\_b INTO query\_1b;

dump result\_b;

--query\_1c 1st test

----------------------------------------------------------------------------------

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

joined = JOIN twitter\_account BY email\_address, stack\_overflow\_account by email\_address;

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

filtered\_num\_tweets = FILTER joined BY num\_tweets > 666;

--Format

--{twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int}

email\_reputations = FOREACH filtered\_num\_tweets GENERATE twitter\_account::num\_tweets, stack\_overflow\_account::reputation;

--Format

--{group: chararray,email\_reputations: {(twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int)}}

email\_reputations\_group = GROUP email\_reputations ALL;

--Format

--{num\_tweets: long,avg\_stack\_overflow\_reputation: double}

result\_c1 = FOREACH email\_reputations\_group GENERATE SUM($1.$0) AS num\_tweets, AVG($1.$1) AS avg\_stack\_overflow\_reputation;

DESCRIBE result\_c1;

STORE result\_c1 INTO query\_1c;

dump result\_c;

--query\_1c 2rd test

----------------------------------------------------------------------------------

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

joined = JOIN twitter\_account BY email\_address, stack\_overflow\_account by email\_address;

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

filtered\_num\_tweets = FILTER joined BY num\_tweets > 666;

--Format

--{twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int}

email\_reputations = FOREACH filtered\_num\_tweets GENERATE twitter\_account::num\_tweets, stack\_overflow\_account::reputation;

--Format

--{group: chararray,email\_reputations: {(twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int)}}

email\_reputations\_group = GROUP email\_reputations ALL;

--Format

--{num\_tweets: long,avg\_stack\_overflow\_reputation: double}

result\_c2 = FOREACH email\_reputations\_group GENERATE SUM($1.$0) AS num\_tweets, AVG($1.$1) AS avg\_stack\_overflow\_reputation;

DESCRIBE result\_c2;

STORE result\_c2 INTO query\_1c2;

dump result\_c2;

--query\_1c 2rd test

----------------------------------------------------------------------------------

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

joined = JOIN twitter\_account BY email\_address, stack\_overflow\_account by email\_address;

--Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,stack\_overflow\_account::email\_address: chararray,stack\_overflow\_account::reputation: int,stack\_overflow\_account::num\_questions: int}

filtered\_num\_tweets = FILTER joined BY num\_tweets > 666;

--Format

--{twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int}

email\_reputations = FOREACH filtered\_num\_tweets GENERATE twitter\_account::num\_tweets, stack\_overflow\_account::reputation;

--Format

--{group: chararray,email\_reputations: {(twitter\_account::num\_tweets: int,stack\_overflow\_account::reputation: int)}}

email\_reputations\_group = GROUP email\_reputations ALL;

--Format

--{num\_tweets: long,avg\_stack\_overflow\_reputation: double}

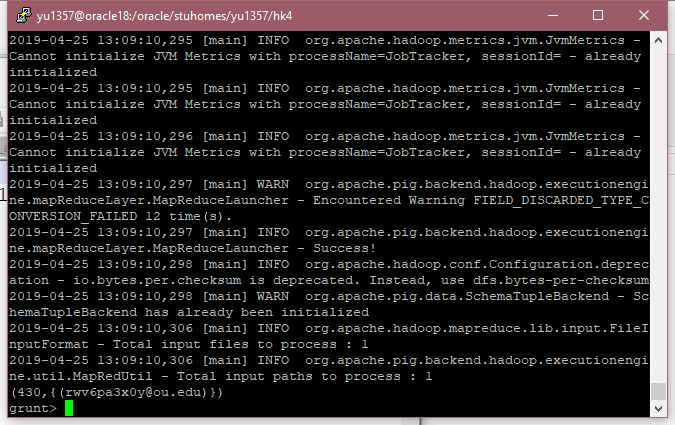
result\_c3 = FOREACH email\_reputations\_group GENERATE SUM($1.$0) AS num\_tweets, AVG($1.$1) AS avg\_stack\_overflow\_reputation;

DESCRIBE result\_c3;

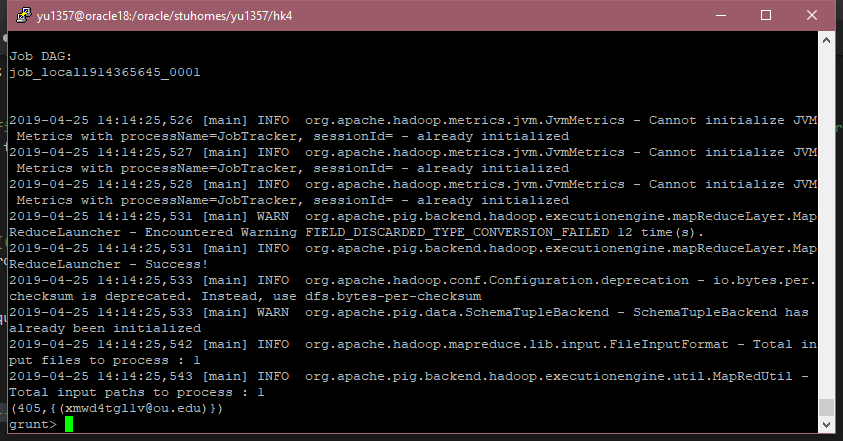
STORE result\_c3 INTO query\_1c3;

dump result\_c3;

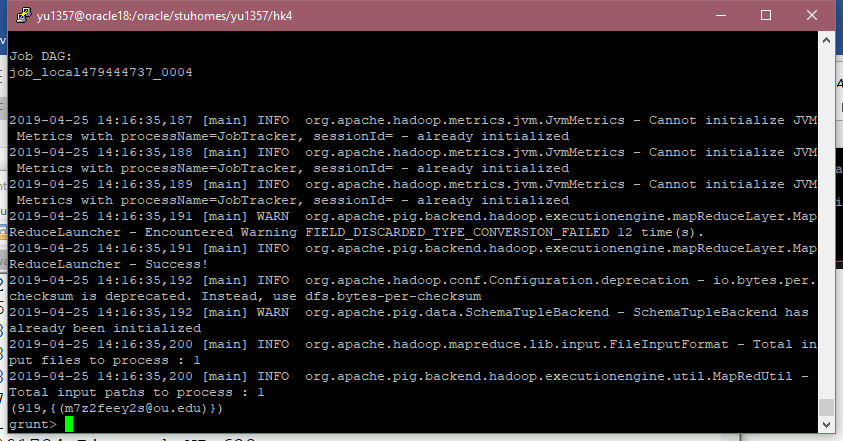
## Query\_1a output test 1 Input: 430



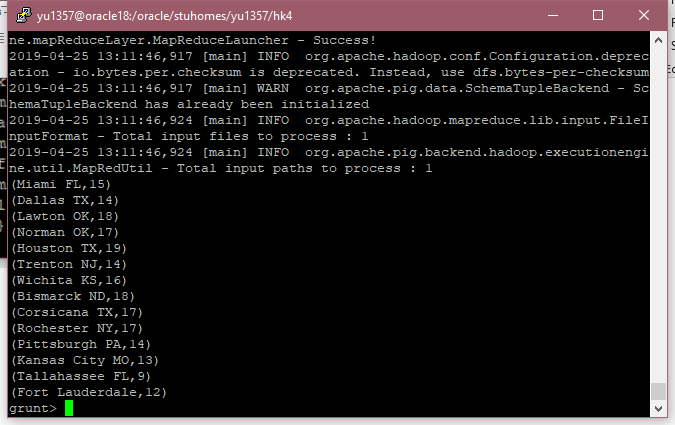
## Query\_1a output test 2 input: 405



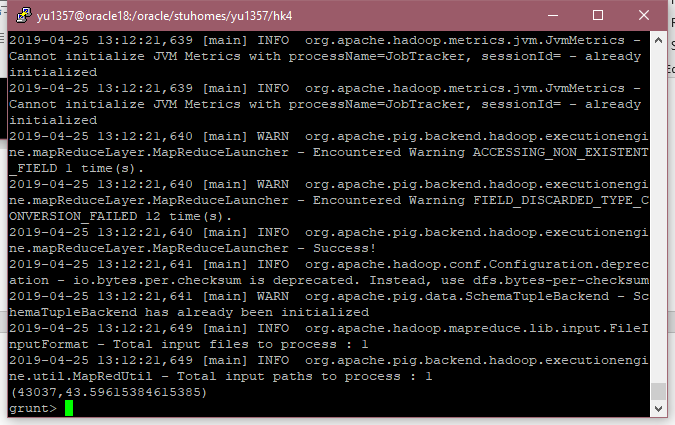
## Query\_1a output test 3 input: 919



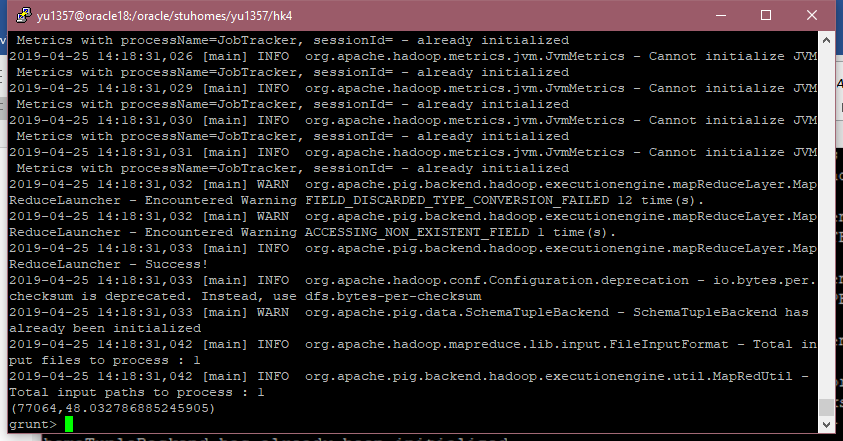
## Query\_1b output:



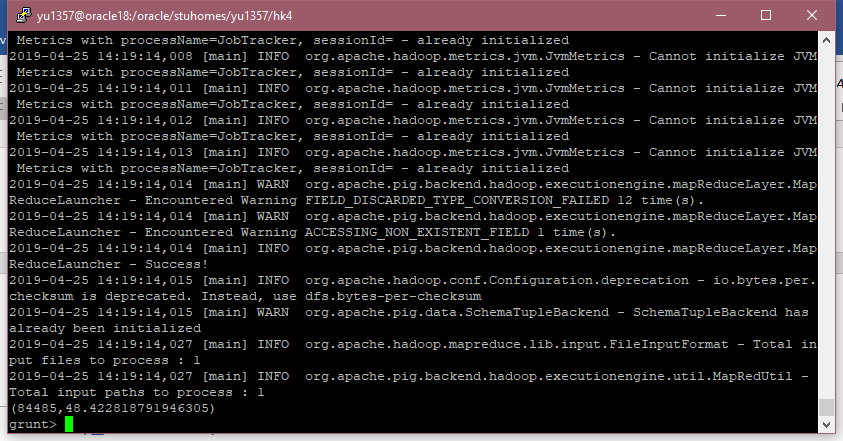
## Query\_1c test 1 input: 666



## Query\_1c test 2 input: 333



## Query\_1c test 3 input: 233



# Task 2

## query2.PIG:

-- Format:

--{follower\_twitter\_account\_id: int,subject\_twitter\_account\_id: int}

follows\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-follows\_account.txt' using PigStorage(',') AS (follower\_twitter\_account\_id: int,subject\_twitter\_account\_id: int);

-- Format:

--{twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

twitter\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-twitter\_account.txt' USING PigStorage(',') AS (twitter\_account\_id: int, email\_address: chararray , phone\_number: chararray , user\_location: chararray , num\_tweets: int);

-- Format:

--{twitter\_account\_id: int,twitter\_account\_rank: float}

old\_twitter\_account\_rank = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-old\_twitter\_account\_rank.txt' using PigStorage(',') AS (twitter\_account\_id:int, twitter\_account\_rank:float);

--Format

--{group: int,follows\_account: {(follower\_twitter\_account\_id: int,subject\_twitter\_account\_id: int)}}

B = GROUP follows\_account BY follower\_twitter\_account\_id;

--Format

--{group: int,long}

follower = FOREACH B GENERATE group, COUNT($1);

--Format

--{old\_twitter\_account\_rank::twitter\_account\_id: int,old\_twitter\_account\_rank::twitter\_account\_rank: float,follower::group: int,long}

follower = JOIN old\_twitter\_account\_rank BY twitter\_account\_id, follower BY $0;

--Format

--{follower\_twitter\_account\_id: int,follower\_rank: float,num\_following: long}

follower = FOREACH follower GENERATE $0 AS follower\_twitter\_account\_id, $1 AS follower\_rank, $3 AS num\_following;

--Format

--{follows\_account::follower\_twitter\_account\_id: int,follows\_account::subject\_twitter\_account\_id: int,follower::follower\_twitter\_account\_id: int,follower::follower\_rank: float,follower::num\_following: long}

sub\_folnum = JOIN follows\_account BY follower\_twitter\_account\_id, follower BY $0;

--Format

--{subject\_twitter\_account\_id: int,follower\_rank: float,num\_following: long}

sub\_count = FOREACH sub\_folnum GENERATE $1 AS subject\_twitter\_account\_id, $3 AS follower\_rank, $4 AS num\_following;

--Format

--{follows\_account::follower\_twitter\_account\_id: int,float}

sub\_count1 = FOREACH sub\_folnum GENERATE $0, (follower\_rank/num\_following);

--Format

--{group: int,sub\_count1: {(follows\_account::follower\_twitter\_account\_id: int,float)}}

C = GROUP sub\_count1 BY $0;

--Format

--{group: int,{(float)}}

temp = FOREACH C GENERATE group, $1.$1;

--Format

--{group: int,double}

temp1 = FOREACH temp GENERATE $0 , SUM($1);

--Format

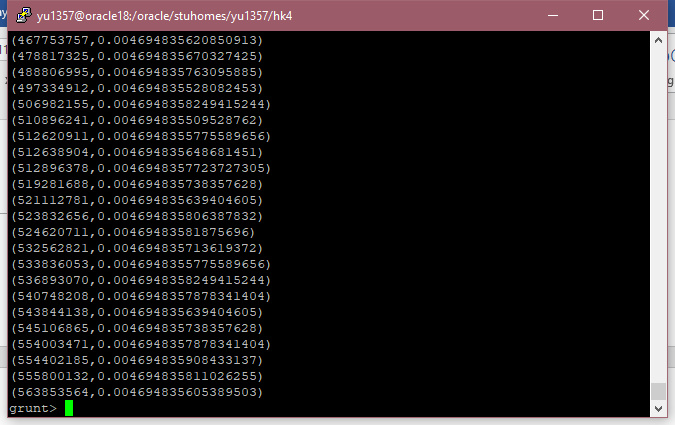
--{group: int,double}

result = FOREACH temp1 GENERATE $0, ((1-0.85)/213 + 0.85\*$1);

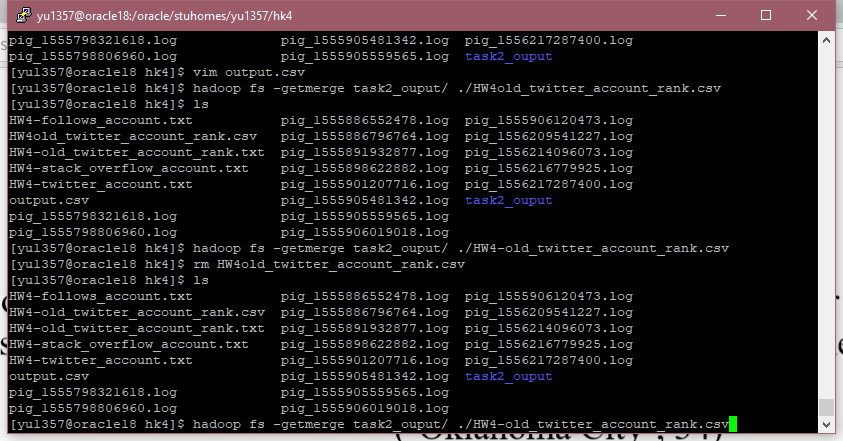
dump result;

STORE result INTO 'myoutput.txt' using PigStorage(',');

## New Rank Twitter Account



## Copy output data into file: HW4old\_twitter\_account\_rank.csv



# Find k\_percentile\_accounts.pig

-- Format:

--{twitter\_account\_id: int,email\_address: chararray,phone\_number: chararray,user\_location: chararray,num\_tweets: int}

twitter\_account = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-twitter\_account.txt' USING PigStorage(',') AS (twitter\_account\_id: int, email\_address: chararray , phone\_number: chararray , user\_location: chararray , num\_tweets: int);

-- Format:

--{twitter\_account\_id: int,twitter\_account\_rank: float}

old\_twitter\_account\_rank = LOAD '/oracle/stuhomes/yu1357/hk4/HW4-old\_twitter\_account\_rank.csv' using PigStorage(',') AS (twitter\_account\_id:int, twitter\_account\_rank:float);

--------------------------------first time testing----------------------------

-- Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,old\_twitter\_account\_rank::twitter\_account\_id: int,old\_twitter\_account\_rank::twitter\_account\_rank: float}

join\_twitter = JOIN twitter\_account BY twitter\_account\_id, old\_twitter\_account\_rank BY twitter\_account\_id;

--Format

--{twitter\_account\_id: int,email\_address: chararray,twitter\_rank: float}

percentile\_accounts = FOREACH join\_twitter GENERATE $0 AS twitter\_account\_id, $1 AS email\_address, $6 AS twitter\_rank;

result1 = FILTER percentile\_accounts BY twitter\_rank > 0.3;

DESCRIBE result1;

STORE result1 INTO task3\_1;

dump result1;

--------------------------------Second time testing----------------------------

-- Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,old\_twitter\_account\_rank::twitter\_account\_id: int,old\_twitter\_account\_rank::twitter\_account\_rank: float}

join\_twitter = JOIN twitter\_account BY twitter\_account\_id, old\_twitter\_account\_rank BY twitter\_account\_id;

--Format

--{twitter\_account\_id: int,email\_address: chararray,twitter\_rank: float}

percentile\_accounts = FOREACH join\_twitter GENERATE $0 AS twitter\_account\_id, $1 AS email\_address, $6 AS twitter\_rank;

result2 = FILTER percentile\_accounts BY twitter\_rank > 0.0042;

DESCRIBE result2;

STORE result2 INTO task3\_2;

dump result2;

--------------------------------thrid time testing----------------------------

-- Format

--{twitter\_account::twitter\_account\_id: int,twitter\_account::email\_address: chararray,twitter\_account::phone\_number: chararray,twitter\_account::user\_location: chararray,twitter\_account::num\_tweets: int,old\_twitter\_account\_rank::twitter\_account\_id: int,old\_twitter\_account\_rank::twitter\_account\_rank: float}

join\_twitter = JOIN twitter\_account BY twitter\_account\_id, old\_twitter\_account\_rank BY twitter\_account\_id;

--Format

--{twitter\_account\_id: int,email\_address: chararray,twitter\_rank: float}

percentile\_accounts = FOREACH join\_twitter GENERATE $0 AS twitter\_account\_id, $1 AS email\_address, $6 AS twitter\_rank;

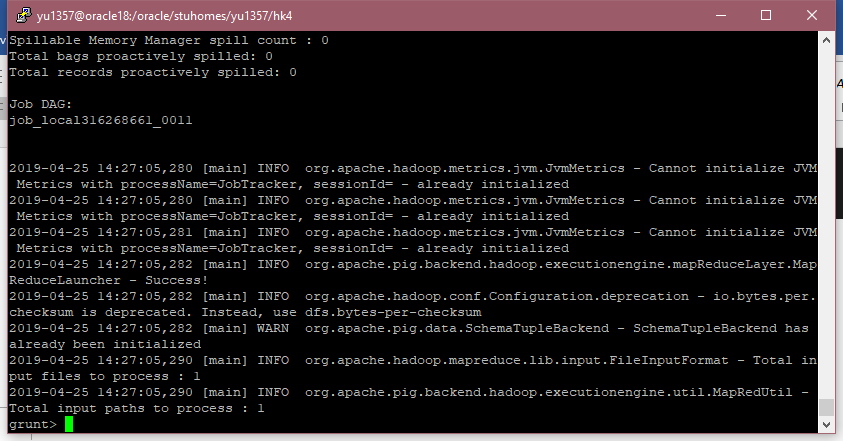
result3 = FILTER percentile\_accounts BY twitter\_rank > 0.005;

DESCRIBE result3;

STORE result3 INTO task3\_3;

dump result3;

## Task3 output test 1 Input: 0.3



## Task3 output test 2 Input: 0.0042



## Task3 output test 2 Input: 0.005

