wget <http://www.utdallas.edu/~axn112530/cs6350/Pig/data.zip>

unzip data.zip

cd data

hdfs dfs -copyFromLocal data.zip /Pig

hdfs dfs -ls /pig

pig -x local

divs = load 'NYSE\_dividends' as (exchange, symbol, date, dividends);

prices = load 'NYSE\_daily' as (exchange, symbol, date, open, high, low, close, volume, adj\_close);

bball = load 'baseball' as (name:chararray, team:chararray, position:bag{t:(p:chararray)}, bat:map[]);

crawl = load 'webcrawl' as (url, pageid, outpages:bag{t:(p:chararray)} );

describe prices;

**Question 1**

grpd = group prices by symbol;

describe grpd;

avgClose = foreach grpd GENERATE group, AVG(prices.close);

describe avgClose;

store avgClose into 'average\_Price\_close';

cat average\_Price\_close;

**Question 2**

grpd\_v = group divs by symbol;

maxDiv = foreach grpd\_v GENERATE group, MAX(divs.dividends);

store maxDiv into 'Max\_div';

cat Max\_div;

**Question 3**

**jnd = join prices by symbol, divs by symbol;**

**grpd = group jnd by prices::symbol;**

**x = foreach grpd generate group as stock, AVG(jnd.prices::open) as avgOpen;**

store x into 'Avg\_P\_Open';

cat Avg\_P\_Close;

**y = foreach grpd generate group as stock, AVG(jnd.** divs::dividends**) as avgDiv;**

store y into 'Avg\_D\_Div';

cat Avg\_D\_Div;

**Question 4**

players = load 'baseball' as (name:chararray, team:chararray, position:bag{t:(p:chararray)}, bat:map[]);

pos = foreach players generate name, flatten(position) as position;

bypos = group pos by position;

// bypos = group pos by position parallel 10;

countPlayers = foreach bypos GENERATE group, COUNT(pos.name);

dump countPlayers;

store countPlayers into 'Player\_count';

cat Player\_count;

**Question 5**

divs = load 'NYSE\_dividends' as (exchange, symbol, date, dividends);

prices = load 'NYSE\_daily' as (exchange, symbol, date, open, high, low, close, volume, adj\_close);

grpd = COGROUP divs by symbol, prices by symbol;

nodivs = FILTER grpd by COUNT(divs) == 0;

describe nodivs;

out = foreach nodivs GENERATE flatten(prices);

OR

// out = foreach nodivs GENERATE flatten(prices.symbol);

OR

// out = foreach nodivs GENERATE group;

describe out;

store out into 'Stocks';

cat Stocks;

**Question 6**

crawl = load 'webcrawl' as (url, pageid, outpages:bag{t:(p:chararray)} );  
fltnd = foreach crawl generate url, flatten(outpages) as outpage;  
grpd = group fltnd by url;  
outcount = foreach grpd generate group, COUNT(fltnd.outpage) as outlinks;  
store outcount into 'outcount';

cat outcount;

grpdin = group fltnd by outpage;  
incount = foreach grpdin generate group, COUNT(fltnd.url) as inlinks;  
store incount into 'in\_count';

cat in\_count;

**Question 7**

crawl = load 'webcrawl' as (url, pageid, outpages:bag{t:(p:chararray)} );

previous\_pagerank = foreach crawl generate url, 1 as pagerank, outpages as links;

outbound\_pagerank = FOREACH previous\_pagerank GENERATE pagerank / COUNT (links) AS pagerank, FLATTEN (links) AS to\_url;

new\_pagerank = FOREACH (COGROUP outbound\_pagerank BY to\_url, previous\_pagerank BY url INNER ) GENERATE group AS url, ( 1 - .5 ) + .5 \* SUM (outbound\_pagerank.pagerank) AS pagerank, FLATTEN (previous\_pagerank.links ) AS links;

STORE new\_pagerank INTO 'pagerank';

cat pagerank;

**Question 8**

grpd = group bball by team;

cntplayers = foreach grpd GENERATE group, COUNT(bball.name) as cnt;

store cntplayers into 'cntplayers';

cat cntplayers;

x = order cntplayers by cnt desc;

y = limit x 5;

dump y;

store y into 'plyrs';

cat plyrs;