```
sales = load 'custsales.txt' using PigStorage(' ') as
(salesid:chararray, custid:int, saleamount:long);

cust = load 'custid.txt' using PigStorage(' ') as
(custid:int, name:chararray);


joined = join sales by custid, cust by custid ;



grouped = group joined by cust::custid ;

filtered = filter grouped by COUNT(joined) > 2;



cog = cogroup cust by custid, sales by custid ;



u = union cust, sales ;
describe u;


u = union onschema cust, sales;
describe u;
```

Rank is used to rank a tuple by a field

```
ranked=rank sales by amount;

ranked = rank sales by amount desc;

ranked = rank sales by amount desc dense;
```

How to pick the top 5 in the ranked data ?

```
top5 = filter ranked by rank_sales < 5
```

Note: rank_sales is the name of the variable it since "sales" is the
name of the relation into which we loaded our original file custsales.txt (scroll to the top of the page to see this)

```
==============================
parameter substitution
==============================
```

```
input_daily= load 'NYSE_daily' as (exchange:chararray, symbol:chararray, date:chararray, open:float,
high:float, low:float,close:float, volume:int, adj_close:float);

stock_prices= filter input_daily by date == '$date';

dump stock_prices;

pig -p date=7/1/2003 <name of the pig script>
```

```
========parameters in a file ===============
input_daily= load 'NYSE_daily' as (exchange:chararray, symbol:chararray, date:chararray, open:float,
high:float, low:float, close:float, volume:int, adj_close:float);
stock_prices= filter input_daily by date == '$date' AND close > $threshold;
dump stock_prices;
```

contents of the params file <myparams>

```
7/1/2003
5
```

```
pig -param_file <name of the param file> <name of the pig script>
```

```
=========AVG function===============

input_divs = LOAD 'NYSE_dividends' as (exchange:chararray, symbol:chararray, date:chararray,
dividend:float);

grouped = GROUP input_divs BY symbol;

average = FOREACH grouped GENERATE group, AVG(input_divs.dividend);
```

```
========COUNT========
input_divs= LOAD 'NYSE_dividends' AS (exchange:chararray, symbol:chararray, date:chararray,
dividend:float);

grouped = GROUP input_divs BY symbol;

counted = FOREACH grouped GENERATE group, COUNT(input_divs);
```

```
========CONCAT====
input_divs = LOAD 'NYSE_dividends' as (exchange:chararray, symbol:chararray, date:chararray,
dividend:float);
```

concatenated = FOREACH input_divs GENERATE CONCAT(exchange, symbol);

OR

concatenated = FOREACH input_divs GENERATE CONCAT(CONCAT(exchange, '-'), symbol);

=======MIN MAX=====
input_divs= LOAD 'NYSE_dividends' AS (exchange:chararray, symbol:chararray, date:chararray, dividend:float);

grouped = GROUP input_divs BY symbol;

maximum = FOREACH grouped GENERATE input_divs.symbol, MAX(input_divs.dividend);

OR

maximum = FOREACH grouped GENERATE group, MAX(input_divs.dividend);

========SIZE=====
size = FOREACH input_divs GENERATE SIZE(exchange), SIZE(dividend);

========UDF (run in mapreduce mode)

http://svn.apache.org/viewvc/pig/trunk/contrib/piggybank/java/src/main/java/org/apache/pig/piggybank//

register '/home/hduser/pig/contrib/piggybank/java/piggybank.jar';

define reverse org.apache.pig.piggybank.evaluation.string.Reverse();

myline = LOAD 'original.txt' as (name:chararray);

reversed = FOREACH myline GENERATE reverse($0);