Report for HW2

Yihan Bao
Huan Ye

1.

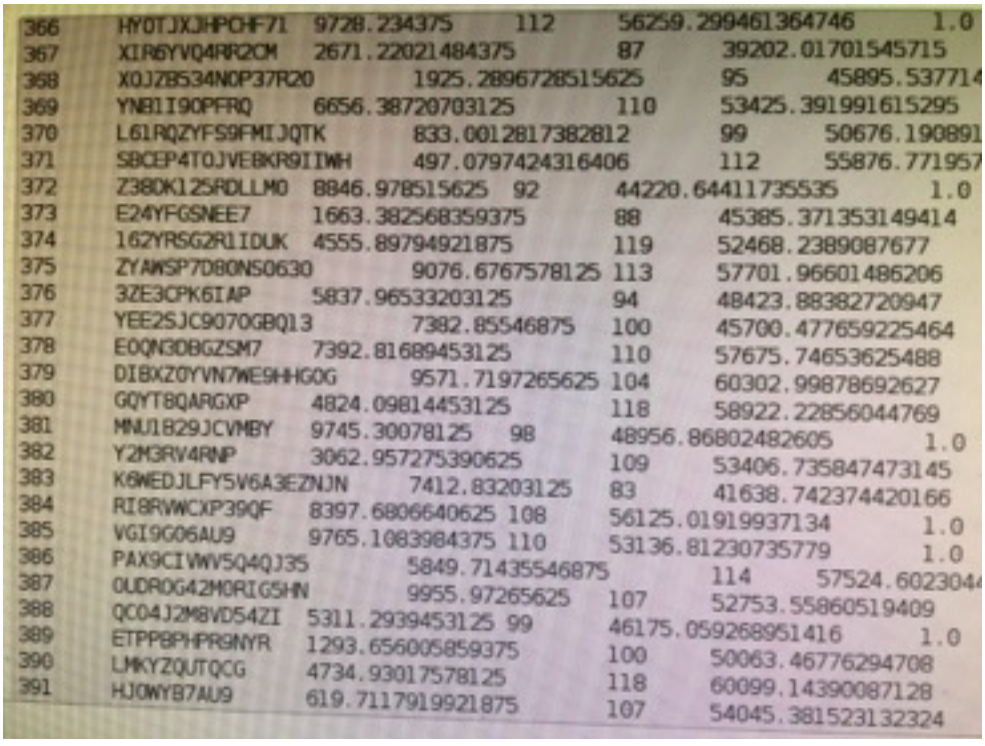
Load *Transactions* relation and grouped it by *CustID. C*ounted number of transactions and got sum of transactions.

Result shows as following:

```
1,85,40701.15019989014
2,88,44565.23631286621
3,106,54254.725447654724
4,113,57380.133120536804
5,88,48799.698254585266
6,96,45264.22575378418
7,84,43528.53710651398
8,111,57900.93301200867
9,98,45418.06533718109
10,113,58015.9346408844
11,97,47776.19218444824
12,71,34708.485929489136
13,106,57131.29683589935
14,90,47265.41662979126
15,99,52208.582010269165
16,93,45283.01296520233
17,100,48259.49715805054
```

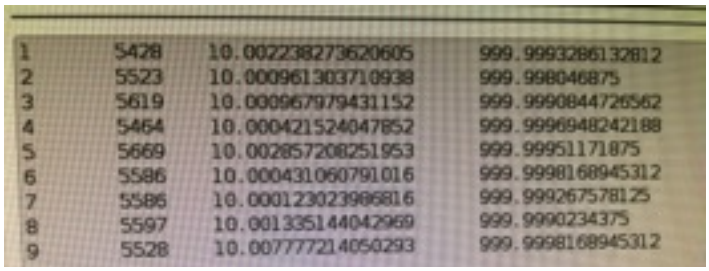
2.

Load customer file and transaction file. Group transaction by CustId and for each group, generate the customer Id, number of transactions ,sum of transactions and the minimum number of items. For each customer in customers relation, generate the Id, customer name and the salary. Then join the result from grouping transaction and selected features in customer relation by the common customer ID.



```
366  HYOTJXJHPCHF71  9728.234375     112  56259.299461364746     1.0
367  XIR6YVQ4RR2CM   2671.22021484375     87  39202.01701545715
368  XOJZB534NOP37R20     1925.2896728515625  95  45895.537714
369  YN8II9OPFRQ     6656.38720703125    110  53425.391991615295
370  L61RQZYFS9FMIJQTK    833.0012817382812   99  50676.190891
371  SBCEP4TOJVEBKR9IIWH  497.0797424316406   112  55876.771957
372  Z38DK125RDLLM0  8846.978515625  92  44220.64411735535      1.0
373  E24YFGSNEE7     1663.382568359375   88  45385.371353149414
374  162YRSG2R1IDUK  4555.89794921875    119  52468.2389087677
375  ZYAWSP7D80NSO630     9076.6767578125  113  57701.96601486206
376  3ZE3CPK6IAP     5837.96533203125    94  48423.88382720947
377  YEE2SJC9070GBQ13     7382.85546875   100  45700.477659225464
378  EOQN3OBGZSM7    7392.81689453125    110  57675.74653625488
379  DIBXZOYVN7WE9HHGOG   9571.7197265625  104  60302.99878692627
380  GQYT8QARGXP     4824.09814453125    118  58922.22856044769
381  MNU1B29JCVMBY   9745.30078125  98  48956.86802482605      1.0
382  Y2M3RV4RNP      3062.957275390625   109  53406.73584747345
383  K8WEDJLFY5V6A3EZNJN  7412.83203125  83  41638.742374420166
384  RI8RWWCXP39QF   8397.6806640625 108  56125.01919937134      1.0
385  VGI9GO6AU9      9765.1083984375 110  53136.81230735779      1.0
386  PAX9CIWWV5Q4QJ35     5849.71435546875    114  57524.6023044
387  OUDROG42MORIG5HN     9955.97265625   107  52753.55860519409
388  QCO4J2M8VD54ZI  5311.2939453125 99  46175.059268951416     1.0
389  ETPPBPHPR9NYR   1293.656005859375   100  50063.46776294708
390  LMKYZQUTQCG     4734.93017578125    118  60099.14390087128
391  HJOWYB7AU9      619.7117919921875   107  54045.381523132324
```

3.
Load customer file and transaction file. For each customer, project ID and CountryCode. For each transaction, project the customer ID and transTotal. Join results form the previous two steps by customer ID. Group the joined results based on countryCode.
Step 6: For each group, project the countryCode, minimum of transaction amount and maximum of transaction amount.
Step 7: Then group result from step two by CountryCode, project the countryCode and the number of customers.
Step8: Join results from step 6 and step7.



1	5428	10.002238273620605	999.999328613281
2	5523	10.000961303710938	999.99804687
3	5619	10.000967979431152	999.9990844726562
4	5464	10.000421524047852	999.9996948242188
5	5669	10.002857208251953	999.99951171875
6	5586	10.000431060791016	999.9998168945312
7	5586	10.000123023986816	999.999267578125
8	5597	10.001335144042969	999.9990234375
9	5528	10.007777214050293	999.9998168945312

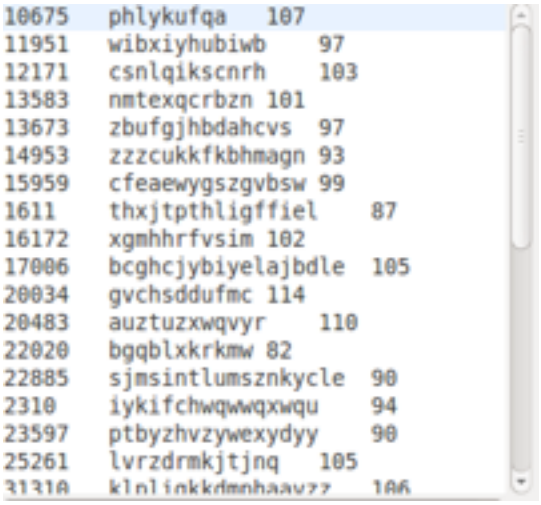
4.
Loaded and did projection of *Customers* and *Transactions* relation. Joined them use distributed cache(*replicated*). Grouped join table by *ID* and *name,* got count of every ID and name. Then group result all, find the minimum. After getting minimum, we wanted to get all minimum customers, so we did select in our count result.

Result shows as following:

```
wezyzyvuyx,60
```

5.
Used *mapper.py* to do filter and projection of each relation and *reducer.py* to count every customer's transactions, because the key value passed from mapper to reducer is sorted by their keys.

Result shows as following:



10675	phlykufqa	107
11951	wibxiyhubiwb	97
12171	csnlqikscnrh	103
13583	nmtexqcrbzn	101
13673	zbufgjhbdahcvs	97
14953	zzzcukkfkbhmagn	93
15959	cfeaewygszgvbsw	99
1611	thxjtpthligffiel	87
16172	xgmhhrfvsim	102
17006	bcghcjybiyelajbdle	105
20034	gvchsddufmc	114
20483	auztuzxwqvyr	110
22020	bgqblxkrkmw	82
22885	sjmsintlumsznkycle	90
2310	iykifchwqwwqxwqu	94
23597	ptbyzhvzywexydyy	90
25261	lvrzdrmkjtjnq	105
31310	klnlingkkdmnhaavzz	106

6.
Read the customer file and in mapper, for each customer, project the countryCode as key and “1” as value. Then read the output from mapper, count the number of customers for each countryCode.