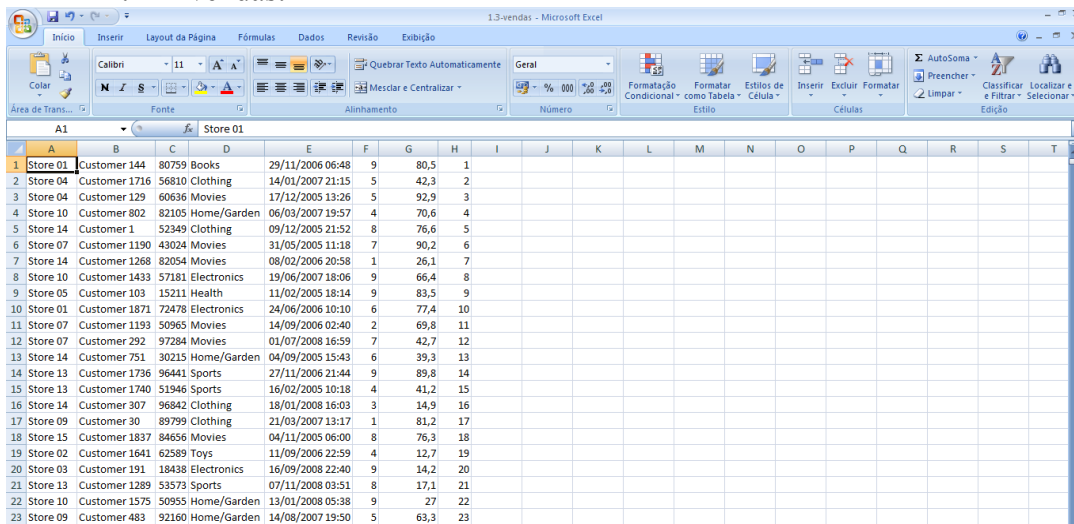


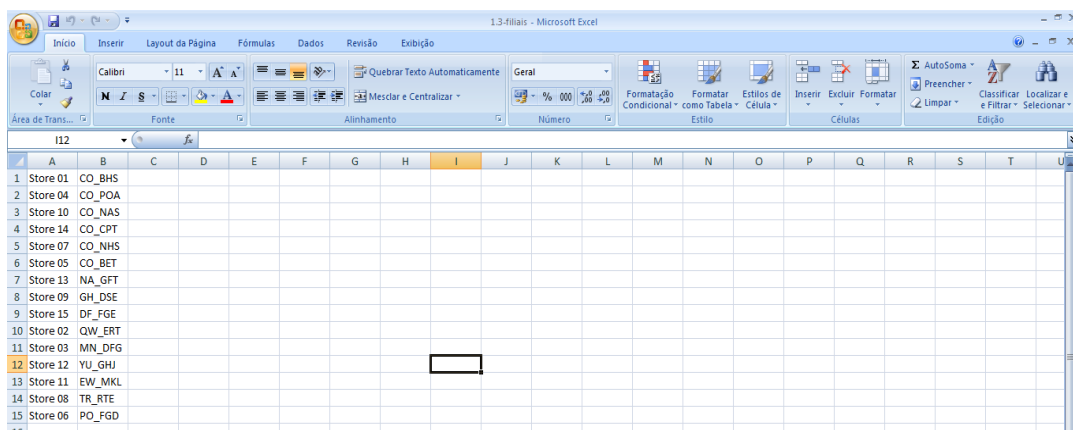
Neste caso vamos implementar um junção e agrupamento de duas tabelas que estão representadas por dois arquivos:

## I. Vendas:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Store 01	Customer 144	80759	Books	29/11/2006 06:48	9	80,5	1												
2	Store 04	Customer 1716	56810	Clothing	14/01/2007 21:15	5	42,3	2												
3	Store 04	Customer 129	60636	Movies	17/12/2005 13:26	5	92,9	3												
4	Store 10	Customer 802	82105	Home/Garden	06/03/2007 19:57	4	70,6	4												
5	Store 14	Customer 1	52349	Clothing	09/12/2005 21:52	8	76,6	5												
6	Store 07	Customer 1190	43024	Movies	31/05/2005 11:18	7	90,2	6												
7	Store 14	Customer 1268	82054	Movies	08/02/2006 20:58	1	26,1	7												
8	Store 10	Customer 1433	57181	Electronics	19/06/2007 18:06	9	66,4	8												
9	Store 05	Customer 103	15211	Health	11/02/2005 18:14	9	83,5	9												
10	Store 01	Customer 1871	72478	Electronics	24/06/2006 10:10	6	77,4	10												
11	Store 07	Customer 1193	50965	Movies	14/09/2006 02:40	2	69,8	11												
12	Store 07	Customer 292	97284	Movies	01/07/2008 16:59	7	42,7	12												
13	Store 14	Customer 751	30215	Home/Garden	04/09/2005 15:43	6	39,3	13												
14	Store 13	Customer 1736	96441	Sports	27/11/2006 21:44	9	89,8	14												
15	Store 13	Customer 1740	51946	Sports	16/02/2005 10:18	4	41,2	15												
16	Store 14	Customer 307	96842	Clothing	18/01/2008 16:03	3	14,9	16												
17	Store 09	Customer 30	89799	Clothing	21/03/2007 13:17	1	81,2	17												
18	Store 15	Customer 1837	84656	Movies	04/11/2005 06:00	8	76,3	18												
19	Store 02	Customer 1641	62589	Toys	11/09/2006 22:59	4	12,7	19												
20	Store 03	Customer 191	18438	Electronics	16/09/2008 22:40	9	14,2	20												
21	Store 13	Customer 1289	53573	Sports	07/11/2008 03:51	8	17,1	21												
22	Store 10	Customer 1575	50955	Home/Garden	13/01/2008 05:38	9	27	22												
23	Store 09	Customer 483	92160	Home/Garden	14/08/2007 19:50	5	63,3	23												

## II. Filiais:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Store 01	CO_BHS																			
2	Store 04	CO_POA																			
3	Store 10	CO_NAS																			
4	Store 14	CO_CPT																			
5	Store 07	CO_NHS																			
6	Store 05	CO_BET																			
7	Store 13	NA_GFT																			
8	Store 09	GH_DSE																			
9	Store 15	DF_FGE																			
10	Store 02	QW_ERT																			
11	Store 03	MN_DFG																			
12	Store 12	YU_GHJ																			
13	Store 11	EW_MKL																			
14	Store 08	TR_RTE																			
15	Store 06	PO_FGD																			

Neste caso o que se deseja é fazer um junção dos dois arquivos para que seja apresentado um resultado que seria: Código da filial ( arquivo Filial - campo1), descrição da filial ( arquivo Filial - campo2) e total de itens pedidos ( arquivo Filial – campo6);

Na verdade seria um SQL com join e um group by:

```
Select cod_filial, des_filial, sum(qtd_item) as total
from vendas inner join filial on(filial.cod_filial = vendas.cod_filial)
group by cod_filial, des_filial
```

Implementação:

1. Faça a instalação do python 2.7;
2. Crie o diretório exerc\;
3. Crie o diretório exerc\join;
4. Copie o arquivo zipado(2.2 Join.zip) para o diretório criado no passo anterior;

5. Copie o arquivo python do mincemeat para o diretório exerc\
 <https://github.com/michaelfairley/mincemeatpy>
6. Crie um arquivo texto e digite a primeira parte do código – Importação dos arquivos:

```
import mincemeat
import glob
import csv

text_files = glob.glob('G:\\NOSQL\\Exerc\\Join\\*')

def file_contents(file_name):
    f = open(file_name)
    try:
        return f.read()
    finally:
        f.close()

source = dict((file_name, file_contents(file_name)) for file_name in text_files)
```

Esta parte faz a importação dos dados e gera um objeto 'source' que é do tipo dicionário: nome do arquivo e conteúdo do arquivo;

7. Faça a implementação do método *map*:

```
def mapfn(k, v):
    print 'map ' + k
    for line in v.splitlines():
        if k == 'G:\\NOSQL\\Exercicios\\1.3 Join\\1.3-vendas.csv':
            yield line.split(';')[0], 'vendas' + ';' + line.split(';')[5]
        if k == 'G:\\NOSQL\\Exercicios\\1.3 Join\\1.3-filiais.csv':
            yield line.split(';')[0], 'Filial' + ';' + line.split(';')[1]
```

Altera os 'if' acima para o caminho e nomes dos arquivos da sua estação de trabalho;

8. Faça a implementação do método *reduce*. Implementação simples para entender o que acontece:

```
def reducefn(k, v):
    print 'reduce ' + k
    return v
```

9. Utilize agora o *mincemeat* ele vai simular o papel da DFS e do 'name mode':

```
s = mincemeat.Server()

# A fonte de dados pode ser qualquer objeto do tipo dicionário
s.datasource = source
s.mapfn = mapfn
s.reducefn = reducefn

results = s.run_server(password="changeme")

w = csv.writer(open("Exerc\\RESULT.csv", "w"))
for k, v in results.items():
    w.writerow([k, v])
```

10. Salve o nome do arquivo como exerc22.py;
11. Faça a execução paralela deste código:
  - a- server: python exerc22.py
  - b- Crie dois ou três clientes com o comando:
 

```
python mincemeat.py -p changeme localhost
```

 Para executar remoto, faça a instalação do python e mincemeat conforme passos 1 e 4, respectivamente;
12. Veja o arquivo gerado
13. Faça a implementação final do método *reduce*:

```
def reducefn(k, v):
    print 'reduce' + k
    total = 0
    for index, item in enumerate(v):
        if item.split(":")[0] == 'vendas':
            total = int(item.split(":")[1]) + total
        if item.split(":")[0] == 'Filial':
            NomeFilial = item.split(":")[1]
    L = list()
    L.append(NomeFilial + " , " + str(total))
    return L
```

14. Utilize agora o *mincemeat* novamente:

```
s = mincemeat.Server()
s.datasource = source
s.mapfn = mapfn
s.reducefn = reducefn

results = s.run_server(password="changeme")

w = csv.writer(open("G:\\NOSQL\\Exercicios\\RESULT_1.3.csv", "w"))
for k, v in results.items():
    w.writerow([k, str(v).replace("[", "").replace("]", "").replace("'", "").replace(' ', '')])
```

15. Faça a execução paralela deste código:

a- server: python exerc22.py

b- Crie dois ou três clientes com o comando:

python mincemeat.py -p changeme localhost

*Para executar remoto, faça a instalação do python e mincemeat conforme passos 1 e 4, respectivamente;*

16. Veja o arquivo gerado