databricksExercise1-Assessment1

```
data = sc.textFile("/FileStore/tables/data.csv")
head = data.first()
data.take(5)
```

```
Out[1]: ['InvoiceNo,StockCode,Description,Quantity,InvoiceDate,UnitPrice,Custo
merID,Country',
 '536365,85123A,WHITE HANGING HEART T-LIGHT HOLDER,6,12/1/2010 8:26,2.55,1785
0,United Kingdom',
 '536365,71053,WHITE METAL LANTERN,6,12/1/2010 8:26,3.39,17850,United Kingdo
m',
 '536365,84406B,CREAM CUPID HEARTS COAT HANGER,8,12/1/2010 8:26,2.75,17850,Uni
ted Kingdom',
 '536365,84029G,KNITTED UNION FLAG HOT WATER BOTTLE,6,12/1/2010 8:26,3.39,1785
0,United Kingdom']
```

```
#Removing header
import re


data_produced = data.filter(lambda line: line != head)\
               .map(lambda line: re.sub(r'(?!(([^"]*"){2})*[^"]*$),', '',
line))\
               .map(lambda line: line.split(','))\
               .map (lambda arr: list(map(str.strip, arr)))\
               .map(lambda arr: arr[:3] + [ int(arr[3]), arr[4].split(' ')
[0],float(arr[5]) ] + arr[6:] )\
               .filter(lambda row: ( (row[3] > 0) and row[0].isnumeric() ) )



data_produced.collect()
```

```
Out[3]: [['536365',
  '85123A',
  'WHITE HANGING HEART T-LIGHT HOLDER',
  6,
  '12/1/2010',
  2.55,
  '17850',
  'United Kingdom'],
 ['536365',
  '71053',
```

```
 'WHITE METAL LANTERN',
 6,
 '12/1/2010',
 3.39,
 '17850',
 'United Kingdom'],
['536365',
 '84406B',
 'CREAM CUPID HEARTS COAT HANGER',
 8,
```

```
import datetime
#date_time_str = 'Jun 28 2018 7:40AM'
#date_time_obj = datetime.datetime.strptime(date_time_str, '%b %d %Y %I:%M%p')
dt_str = '12/1/2010 8:26'
d = dt_str.split(" ")[0]
date_time_obj = datetime.datetime.strptime(d, '%m/%d/%Y')

date_time_obj = datetime.datetime.strptime(dt_str, '%m/%d/%Y %H:%M')
print('Date:', str(date_time_obj.date()))
print('Time:', date_time_obj.time())
print('Date-time:', date_time_obj)

Date: 2010-12-01
Time: 08:26:00
Date-time: 2010-12-01 08:26:00

Countrydata = sc.textFile("/FileStore/tables/country_code_list.csv")
#Countrydata.take(10)
country_rdd1 = Countrydata.filter(lambda line: True if (len(line) >0) else
False)
country_rdd1.take(10)



Out[42]: ['Name,Code',
 'Afghanistan,AF',
 'Åland Islands,AX',
 'Albania,AL',
 'Algeria,DZ',
 'American Samoa,AS',
 'Andorra,AD',
 'Angola,AO',
 'Anguilla,AI',
 'Antarctica,AQ']
```

```
country_rdd2 = country_rdd1.filter(lambda line: False if ("Name" in line)else
True)
country_rdd3 = country_rdd2.filter(lambda line: line.split(","))
country_rdd4 = country_rdd3.filter(lambda arr: tuple(arr))
country_rdd4.take(10)
```

```
Out[41]: ['Afghanistan,AF',
 'Åland Islands,AX',
 'Albania,AL',
 'Algeria,DZ',
 'American Samoa,AS',
 'Andorra,AD',
 'Angola,AO',
 'Anguilla,AI',
 'Antarctica,AQ',
 'Antigua and Barbuda,AG']
```

```
result_data = data_produced.map(lambda row: ((row[4],row[7]), row[3]))\
                           .reduceByKey(lambda x,y:x+y)\
                           .map(lambda x: (x[0][0],x[0][1],x[1])) \
                           .repartition(1)


result_data.take(10)
```

```
Out[45]: [('12/1/2010', 'France', 449),
 ('12/1/2010', 'Australia', 107),
 ('12/1/2010', 'Netherlands', 97),
 ('12/1/2010', 'Norway', 1852),
 ('12/1/2010', 'EIRE', 243),
 ('12/2/2010', 'EIRE', 6),
 ('12/3/2010', 'France', 239),
 ('12/3/2010', 'Switzerland', 110),
 ('12/3/2010', 'EIRE', 2575),
 ('12/3/2010', 'Poland', 140)]
```