databricksAssessment-3

```
df = spark.read.option("header", True).option("inferSchema",
True).csv("/FileStore/tables/data.csv")

df.show(2)
```

```
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|UnitPrice|Cu
stomerID|       Country|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|     2.55|
17850|United Kingdom|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
only showing top 2 rows
```

```
from pyspark.sql.functions import col
filteredDf = df.filter ( col("Quantity") > 0)

filteredDf.show(2)
```

```
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|UnitPrice|Cu
stomerID|       Country|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|     2.55|
17850|United Kingdom|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
only showing top 2 rows
```

```
projectionDf = filteredDf.select ("StockCode", "Quantity", "InvoiceDate")
projectionDf.show(2)
```

```
+---------+--------+--------------+
|StockCode|Quantity|   InvoiceDate|
+---------+--------+--------------+
|   85123A|       6|12/1/2010 8:26|
|    71053|       6|12/1/2010 8:26|
+---------+--------+--------------+
only showing top 2 rows
```

```
from pyspark.sql.functions import asc, desc, avg, col, count, avg, sum

inter = projectionDf.groupBy("StockCode")\
                    .agg(sum("Quantity"))

inter.show(2)
```

```
+---------+-------------+
|StockCode|sum(Quantity)|
+---------+-------------+
|    22728|         5364|
|    21889|         6403|
+---------+-------------+
only showing top 2 rows
```

```
from pyspark.sql.functions import split

# 1/17/2011 17:44 MONTH/DAY/YEAR
# projectionDf.withColumn("Date", split_col.getItem(0))\

split_col = split(projectionDf["InvoiceDate"]," ")



projectionDf2 = projectionDf.withColumn("Date", split_col.getItem(0))

split_date = split(projectionDf2["Date"],"/")

from pyspark.sql.functions import concat, lit
projectionDf2 = projectionDf2\
                            .withColumn("Month", split_date.getItem(0))\
                            .withColumn("Year", split_date.getItem(2))\
                            .drop("InvoiceDate")\
                            .select( concat( col("Year"), lit("-"),
col("Month") ).alias("Date"), "StockCode", "Quantity" )


#split_date = split(projectionDf["Date"],"/")

projectionDf2.show(2)
```

```
+-------+---------+--------+
|   Date|StockCode|Quantity|
+-------+---------+--------+
|2010-12|   85123A|       6|
|2010-12|    71053|       6|
+-------+---------+--------+
only showing top 2 rows
```

```
resultDf = projectionDf2.groupBy("StockCode", "Date")\
                    .agg(sum("Quantity"))\
                    .withColumnRenamed("sum(Quantity)", "Quantity")

resultDf.show(2)
```

```
+---------+-------+--------+
|StockCode|   Date|Quantity|
+---------+-------+--------+
|   85231G|2010-12|      45|
|    22445|2010-12|     119|
+---------+-------+--------+
only showing top 2 rows
```

```
outputFile = "/FileStore/tables/assessment-3/results.csv"
# repartition will suffle data, good to be used between your analytics work,
not end of the file writing
# coalesce - will not suffle data, reduce  the partition, good to be used
before producing file
resultDf = resultDf.coalesce(1)
resultDf.write.mode("overwrite").option("header",
True).format("csv").save(outputFile)
```

```
[Truncated to first 65536 bytes]
Out[35]: 'StockCode,Date,sum(Quantity)\n85231G,2010-12,45\n22445,2010-12,119\n
48184,2010-12,140\n21888,2010-12,143\n22831,2010-12,23\n47590B,2010-12,33\n791
64,2010-12,24\n21937,2010-12,15\n21792,2010-12,4\n20831,2010-12,6\n20704,2010-
12,81\n85062,2011-1,151\n90125B,2011-1,21\n15034,2011-1,177\n20700,2011-1,2\n2
1878,2011-1,33\n90202B,2011-1,3\n21929,2011-2,680\n22173,2011-2,261\n21815,201
1-2,1\n84688,2011-2,7\n21676,2011-2,4\n20718,2011-2,358\n85174,2011-2,40\n2227
1,2011-2,90\n84748,2011-2,8\n84306,2011-2,6\n90214R,2011-2,1\n22155,2011-2,1\n
90161A,2011-2,1\n20897,2011-2,8\n22130,2011-2,18\n22442,2011-2,8\n22198,2011-
2,21\n22996,2011-2,302\n20685,2011-3,242\n84536B,2011-3,128\n21544,2011-3,158
\n22262,2011-3,332\n84997D,2011-3,187\n22505,2011-3,175\n22385,2011-3,1444\n84
251B,2011-3,13\n22179,2011-3,111\n85054,2011-3,57\n84683,2011-3,21\n72140E,201
1-3,8\n21643,2011-3,67\n90130B,2011-3,1\n21376,2011-3,4\n35004G,2011-3,1\n8487
6D,2011-3,24\n21672,2011-4,325\n72801c,2011-4,1\n84993a,2011-4,6\n22973,2011-
4,51\n85159A,2011-4,37\n21830,2011-4,169\n22203,2011-4,30\n21781,2011-4,21\n85
230F,2011-4,2\n85035B,2011-4,2\n37446,2011-5,125\n85231B,2011-5,4\n20754,2011-
5,71\n22839,2011-5,54\n22955,2011-5,10\n84509a,2011-5,7\n84279P,2011-5,10\n901
98B,2011-5,3\n22203,2011-6,28\n22677,2011-6,75\n21231,2011-6,323\n21775,2011-
6,37\n22294,2011-6,223\n20819,2011-6,4\n22495,2011-6,22\n16216,2011-6,1726\n21
386,2011-6,118\n90138,2011-6,1\n22308,2011-6,59\n90199D,2011-6,1\n72369A,2011-
6,6\n21696,2011-6,24\n22424,2011-7,68\n23198,2011-7,650\n23290,2011-7,275\n228
```