databricksExcercise5-Assessment1

```
kaggledata = spark.read.format("csv") \
          .option("header",True) \
          .load("FileStore/tables/data.csv")

kaggledata.show(5)
```

```
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
|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|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/2010 8:26|     2.75|
17850|United Kingdom|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+
only showing top 5 rows
```

```
import pyspark
from pyspark.sql.types import
StructType,StringType,IntegerType,DoubleType,FloatType,DateType

ex5_schema = StructType() \
        .add("InvoiceNo",StringType(),False) \
        .add("StockCode",StringType(),False) \
        .add("Description",StringType(),True) \
        .add("Quantity",IntegerType(),True) \
        .add("InvoiceDate",StringType(),True) \
        .add("UnitPrice",DoubleType(),True) \
        .add("CustomerID",IntegerType(),True) \
        .add("Country",StringType(),True)

kaggledata = spark.read.format("csv") \
          .option("header",True) \
          .schema(ex5_schema) \
          .load("FileStore/tables/data.csv")

kaggledata.printSchema()

root
 |-- InvoiceNo: string (nullable = true)
 |-- StockCode: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- InvoiceDate: string (nullable = true)
 |-- UnitPrice: double (nullable = true)
 |-- CustomerID: integer (nullable = true)
 |-- Country: string (nullable = true)


coun_schema = StructType() \
              .add("Country",StringType(), True) \
              .add("CountryCode",StringType(), True)

Countrydata = spark.read.format("csv") \
          .option("header",True) \
          .schema(coun_schema) \
          .load("FileStore/tables/country_code_list.csv")

Countrydata.show(5)

+--------------+-----------+
|       Country|CountryCode|
+--------------+-----------+
```

```
|   Afghanistan|         AF|
| Åland Islands|         AX|
|       Albania|         AL|
|       Algeria|         DZ|
|American Samoa|         AS|
+--------------+-----------+
only showing top 5 rows
```

```
kaggledata.createOrReplaceTempView("InvoiceData")
spark.sql("select InvoiceNo,Quantity,UnitPrice,Country from
InvoiceData").show(5)
```

```
+---------+--------+---------+--------------+
|InvoiceNo|Quantity|UnitPrice|       Country|
+---------+--------+---------+--------------+
|   536365|       6|     2.55|United Kingdom|
|   536365|       6|     3.39|United Kingdom|
|   536365|       8|     2.75|United Kingdom|
|   536365|       6|     3.39|United Kingdom|
|   536365|       6|     3.39|United Kingdom|
+---------+--------+---------+--------------+
only showing top 5 rows
```

```
import pyspark
from pyspark.sql.functions import col

kaggledf_amount = kaggledata.withColumn("Amount_Spent", col("Quantity") *
col("UnitPrice"))
kaggledf_amount.show(5)
```

```
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|UnitPrice|Cu
stomerID|       Country|      Amount_Spent|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|     2.55|
17850|United Kingdom|15.299999999999999|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/2010 8:26|     2.75|
17850|United Kingdom|              22.0|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
```

```
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
only showing top 5 rows


#Cleaning
from pyspark.sql import *

filtered_kaggle_df = kaggledf_amount.filter("Quantity > 0 and InvoiceNo not
like '%C'")
filtered_kaggle_df.show(5)

+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|UnitPrice|Cu
stomerID|       Country|      Amount_Spent|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|     2.55|
17850|United Kingdom|15.299999999999999|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/2010 8:26|     2.75|
17850|United Kingdom|              22.0|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/2010 8:26|     3.39|
17850|United Kingdom|             20.34|
+---------+---------+--------------------+--------+--------------+---------+--
--------+--------------+------------------+
only showing top 5 rows


kaggle_df_group =
filtered_kaggle_df.groupBy("Country").sum("Amount_Spent").withColumnRenamed("su
m(Amount_spent)","Amount Spent")
display(kaggle_df_group)
```

	Country	Amount Spent
1	Sweden	38378.33
2	Singapore	21279.29
3	Germany	228867.1400000001
4	RSA	1002.3099999999998

5	France	209715.11000000002
6	Greece	4760.52
7	European Community	1300.2499999999998
8	Belgium	41196.34000000001

Showing all 38 rows.

```
kaggledatajoin = kaggle_df_group.join(Countrydata, on=['Country'], how='inner')
result_kaggle_df = kaggledatajoin.select("CountryCode","Amount Spent")
```

	CountryCode	Amount Spent
1	SE	38378.33
2	SG	21279.29
3	DE	228867.1400000001
4	FR	209715.11000000002
5	GR	4760.52
6	BE	41196.34000000001
7	FI	22546.079999999998
8	MT	2725.5900000000006

Showing all 32 rows.