databricksExercise 6 - Assessment1

```
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()
kaggledata.show(5)
```

```
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)

+---------+---------+--------------------+--------+-------------+---------+--
--------+--------------+
|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.functions import col

kaggledf_amount = kaggledata.withColumn("Amount_Spent", col("Quantity") *
col("UnitPrice"))
kaggledf_amount.show(5)
filtered_kaggle_df = kaggledf_amount.filter("Quantity > 0 and InvoiceNo not
like '%C' and CustomerID not like '%C'")
filtered_kaggle_df.show(5)
kaggledf_amount = kaggledata.withColumn("Amount_Spent", col("Quantity") *
col("UnitPrice"))
kaggledf_amount.show(5)
kaggle_df_group =
filtered_kaggle_df.groupBy("Country","CustomerID").sum("Amount_Spent").withColu
mnRenamed("sum(Amount_spent)","Amount Spent")
display(kaggle_df_group)
```

	Country	CustomerID	Amount Spent
1	United Kingdom	15100	876
2	United Kingdom	16048	256.44000000000005
3	United Kingdom	17025	357.77
4	United Kingdom	17732	303.97
5	United Kingdom	18041	4183.389999999999
6	United Kingdom	17967	123.07000000000002
7	United Kingdom	16353	6675.710000000001
8	United Kingdom	18037	70.02

Showing the first 1000 rows.

	Country	CustomerID	Amount Spent	rank
1	Sweden	17404	31906.820000000003	1
2	Sweden	12483	2484.9799999999996	2

3	Sweden	12676	1331.39	3
4	Singapore	12744	21279.29	1
5	Germany	12471	19824.05	1
6	Germany	12621	13689.669999999995	2
7	Germany	12477	13219.739999999998	3
8	RSA	12446	1002.3099999999998	1

Showing all 90 rows.