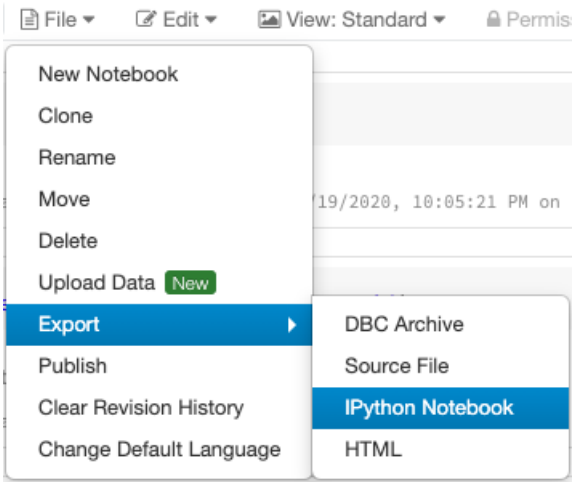
DS 610 Week 4 Assignment
Big Data Analytics

Due Date:

Please Note: As you will be working on Databricks console for this assignment, please submit the IPython Notebook [File-> Export-> IPython Notebook]. Use Markdown. Submissions in the form of screenshots / word documents or in any other format will **NOT** be evaluated.



1.
Which Spark component handles input/output operations?
A) Spark Core
B) Spark SQL
C) Spark I/O
D) Spark Streaming

2.
Categorize the below operations into Actions & Transformation
show, select, distinct, sum, count, collect, groupBy, orderBy, save, filter, limit

Use '/databricks-datasets/online_retail/data-001/data.csv' for #3, #4, #5, #6

3.
Print all the distinct countries in ascending order.
Output:

```
+--------------+
|       Country|
+--------------+
|     Australia|
|       Austria|
|       Bahrain|
|       Belgium|
|Channel Islands|
|        Cyprus|
|       Denmark|
|          EIRE|
|       Finland|
|        France|
|       Germany|
|       Iceland|
|        Israel|
|         Italy|
|         Japan|
|     Lithuania|
|   Netherlands|
|        Norway|
|        Poland|
|      Portugal|
|         Spain|
|        Sweden|
|   Switzerland|
| United Kingdom|
+--------------+
```

4.
Show the order total (unit price times quantity) for each invoice number.
Output:

```
+---------+--------------------------+
|InvoiceNo|sum((UnitPrice * Quantity))|
+---------+--------------------------+
|   536365|                    139.12|
|   536366|      22.200000000000003|
|   536367|                    278.73|
|   536368|       70.05000000000001|
|   536369|                     17.85|
|   536370|                    855.86|
|   536371|                     204.0|
|   536372|      22.200000000000003|
|   536373|                    259.86|
|   536374|                     350.4|
+---------+--------------------------+
only showing top 10 rows
```

5.
Show the StockCode, Description, UnitPrice for InvoiceNo 536596
Output:

```
+---------+--------------------+---------+
|StockCode|         Description|UnitPrice|
+---------+--------------------+---------+
|    21624|VINTAGE UNION JAC...|     5.95|
|    22900| SET 2 TEA TOWELS...|     2.95|
|    22114|HOT WATER BOTTLE ...|     3.95|
|    21967|PACK OF 12 SKULL ...|     0.29|
|   84926A|WAKE UP COCKEREL ...|     1.25|
|    22802|FAUX FUR CHOCOLAT...|    19.95|
+---------+--------------------+---------+
```

6.
Show United Kingdom's top 10 highest selling (unit price times quantity) product description.
Output:

```
+--------------+--------------------+------------------+
|       Country|         Description|        sum(Total)|
+--------------+--------------------+------------------+
|United Kingdom|      DOTCOM POSTAGE| 34177.85999999999|
|United Kingdom|REGENCY CAKESTAND...| 30512.56000000003|
|United Kingdom|WHITE HANGING HEA...|22248.690000000024|
|United Kingdom|       CHILLI LIGHTS|12475.610000000004|
|United Kingdom|RED WOOLLY HOTTIE...| 9355.869999999997|
|United Kingdom|PAPER CHAIN KIT 5...| 9313.069999999996|
|United Kingdom|WHITE SKULL HOT W...| 8867.309999999998|
|United Kingdom|HEART OF WICKER L...| 8175.289999999995|
|United Kingdom|HOT WATER BOTTLE ...| 7946.580000000001|
|United Kingdom|CHOCOLATE HOT WAT...| 7825.719999999996|
+--------------+--------------------+------------------+
```

7.
Using *customer-orders.csv*, show the total amount spent by each customer. The first column represents customer id, second column represents item id, and the third column shows the amount spent.
Output:

```
+-----------+------------------+
|customer_id| sum(amount_spent)|
+-----------+------------------+
|         45|3309.3800055980682|
|         79| 3790.569982469082|
|         96|3924.2299877405167|
|         23| 4042.650001913309|
|         99| 4172.290024012327|
|         75| 4178.499995291233|
|         36| 4278.049998521805|
|         98| 4297.259994864464|
|         47| 4316.299998342991|
|         77| 4327.730022907257|
+-----------+------------------+
only showing top 10 rows
```

8.
Using *fakefriends-header.csv*, show Beverly's data where the number of friends is greater than 200 but less than 300. Sort the results by age in descending order.
Output:

```
+------+-------+---+-------+
|userID|   name|age|friends|
+------+-------+---+-------+
|   290|Beverly| 62|    290|
|   269|Beverly| 55|    289|
|   302|Beverly| 37|    263|
|    52|Beverly| 19|    269|
+------+-------+---+-------+
```

9.
Using *fakefriends-header.csv,* show the data for names starting with 'W' and ending with either 'l' or 'f'.
Output:

```
+------+----+---+-------+
|userID|name|age|friends|
+------+----+---+-------+
|     0|Will| 33|    385|
|     7|Will| 54|    307|
|    76|Will| 62|    201|
|    89|Worf| 24|    492|
|    98|Will| 44|    178|
|   107|Will| 64|    419|
|   114|Worf| 33|    275|
|   119|Worf| 29|    344|
|   136|Will| 19|    335|
|   164|Will| 31|    172|
+------+----+---+-------+
only showing top 10 rows
```

10.
Using *fakefriends-header.csv,* show the data for user ids greater than the number of friends.
Output:

```
+------+--------+---+-------+
|userID|    name|age|friends|
+------+--------+---+-------+
|    17|     Odo| 35|     13|
|    24|  Julian| 25|      1|
|    37|  Geordi| 58|     21|
|    48|     Nog| 20|      1|
|    53|  Geordi| 62|     31|
|    54|   Brunt| 19|      5|
|    62|   Keiko| 69|      9|
|    63|Jean-Luc| 58|     54|
|    64|    Elim| 31|     15|
|    91|     Rom| 46|     88|
+------+--------+---+-------+
only showing top 10 rows
```

Thank you.