In [1]:
```
sc.stop()
```

In [2]:
```
spark.stop()
```

a) Create a new Spark Session with new SparkConfig

In [3]:
```
from pyspark import SparkConf, SparkContext
config = SparkConf().setMaster("local[4]").setAppName("PySpark Assignment")
sc = SparkContext(conf=config)
```

b) Create new instance of Spark SQL session and define new DataFrame using sales_data_sample.csv dataset.

In [4]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("AssignmentSession").getOrCreate()
```

In [5]:
```
spark
```

Out[5]:

SparkSession - hive

SparkContext

Spark UI

Version	v2.4.8
Master	local[4]
AppName	PySpark Assignment

In [6]:
```
sales_data = spark.read.csv('file:///home/hadoop/Downloads/sales_data_sample.
                            header=True,
                            inferSchema=True)
```

c) Find the shape of DataFrame.

In [7]:
```
print("Number of rows:", sales_data.count())
print("Number of cols:", len(sales_data.columns))
```

```
Number of rows: 2823
Number of cols: 25
```

d) Find the Summary of DataFrame for all numerical data columns.

In [8]:
```
sales_data.schema.fields
```

Out[8]:
```
[StructField(ORDERNUMBER,IntegerType,true),
 StructField(QUANTITYORDERED,IntegerType,true),
 StructField(PRICEEACH,DoubleType,true),
 StructField(ORDERLINENUMBER,IntegerType,true),
 StructField(SALES,DoubleType,true),
 StructField(ORDERDATE,StringType,true),
```

```
StructField(STATUS,StringType,true),
StructField(QTR_ID,IntegerType,true),
StructField(MONTH_ID,IntegerType,true),
StructField(YEAR_ID,IntegerType,true),
StructField(PRODUCTLINE,StringType,true),
StructField(MSRP,IntegerType,true),
StructField(PRODUCTCODE,StringType,true),
StructField(CUSTOMERNAME,StringType,true),
StructField(PHONE,StringType,true),
StructField(ADDRESSLINE1,StringType,true),
StructField(ADDRESSLINE2,StringType,true),
StructField(CITY,StringType,true),
StructField(STATE,StringType,true),
StructField(POSTALCODE,StringType,true),
StructField(COUNTRY,StringType,true),
StructField(TERRITORY,StringType,true),
StructField(CONTACTLASTNAME,StringType,true),
StructField(CONTACTFIRSTNAME,StringType,true),
StructField(DEALSIZE,StringType,true)]
```

In [9]:

```
from pyspark.sql.types import IntegerType, LongType, ShortType, ByteType, Flo

numerical_types = (IntegerType, LongType, ShortType, ByteType, FloatType, Dou

numerical_columns = [field.name for field in sales_data.schema.fields if isin

numerical_columns
```

Out[9]:

```
['ORDERNUMBER',
 'QUANTITYORDERED',
 'PRICEEACH',
 'ORDERLINENUMBER',
 'SALES',
 'QTR_ID',
 'MONTH_ID',
 'YEAR_ID',
 'MSRP']
```

In [10]:

```
sales_data.select(numerical_columns).describe().show()
```

```
+-------+-----------------+-----------------+------------------+------------
-----+-----------------+-----------------+------------------+-------------
----+------------------+
|summary|      ORDERNUMBER|  QUANTITYORDERED|         PRICEEACH|  ORDERLINEN
UMBER|            SALES|           QTR_ID|          MONTH_ID|          YEA
R_ID|             MSRP|
+-------+-----------------+-----------------+------------------+------------
-----+-----------------+-----------------+------------------+-------------
----+------------------+
|  count|             2823|             2823|              2823|
2823|             2823|             2823|              2823|             2
823|              2823|
|   mean|10258.725115125753|35.09280906836698| 83.65854410201929|6.4661707403
47148|  3553.88907190932|2.7176762309599716|7.0924548352816155|2003.815090329
4368|100.71555083244775|
| stddev|  92.0854775957196| 9.74144273706958|20.174276527840536| 4.225840964
69094|1841.8651057401842| 1.203878088001756| 3.656633307661765|0.699670154130
0869| 40.18791167720266|
|    min|            10100|                6|             26.88|
1|           482.13|                1|                 1|              2003
|               33|
|    max|            10425|               97|             100.0|
18|          14082.8|                4|                12|              200
5|              214|
+-------+-----------------+-----------------+------------------+------------
-----+-----------------+-----------------+------------------+-------------
```

```
----+-----------------+
```

e) Identify and handle missing or null values in the columns.

In [11]:
```
from pyspark.sql.functions import col, sum

null_counts = sales_data.select([sum(col(c).isNull().cast("int")).alias(c) fo
null_counts.show()
```

```
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|SALES|ORDERDATE|STATUS
|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|CUSTOMERNAME|PHONE|ADDR
ESSLINE1|ADDRESSLINE2|CITY|STATE|POSTALCODE|COUNTRY|TERRITORY|CONTACTLASTNAME
|CONTACTFIRSTNAME|DEALSIZE|
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
|          0|             0|        0|             0|    0|        0|     0
|     0|       0|      0|          0|   0|          0|           0|    0|
0|        2521|   0| 1486|        76|      0|        0|             0|
0|       0|
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
```

In [12]:
```
#ADDRESSLINE2
#STATE
#POSTALCODE
sales_data.select(['ADDRESSLINE2']).show()
```

```
+------------+
|ADDRESSLINE2|
+------------+
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|     Level 3|
|   Suite 101|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
|        null|
+------------+
only showing top 20 rows
```

In [13]:
```
cleaned_sales_data = sales_data.fillna({'ADDRESSLINE2': 'NA', 'STATE': 'NA'})
cleaned_sales_data = cleaned_sales_data.dropna()
```

In [14]:
```
cleaned_sales_data.select([sum(col(c).isNull().cast("int")).alias(c) for c in
```

```
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|SALES|ORDERDATE|STATUS
|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|CUSTOMERNAME|PHONE|ADDR
ESSLINE1|ADDRESSLINE2|CITY|STATE|POSTALCODE|COUNTRY|TERRITORY|CONTACTLASTNAME
|CONTACTFIRSTNAME|DEALSIZE|
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
|          0|             0|        0|             0|    0|        0|     0
|     0|       0|      0|          0|   0|          0|           0|    0|
0|           0|   0|    0|         0|      0|        0|             0|
0|       0|
+-----------+--------------+---------+--------------+-----+---------+------
+------+--------+-------+-----------+----+-----------+------------+-----+----
--------+------------+----+-----+----------+-------+---------+--------------
+---------------+--------+
```

f) Calculate the total revenue generated per country by combining the columns QUANTITYORDERED and PRICEEACH using Spark DataFrame operations?

In [15]:
```
cleaned_sales_data = cleaned_sales_data.withColumn("TOTAL_REVENUE", col("QUAN
cleaned_sales_data.show()
```

```
+-----------+--------------+---------+--------------+-------+-------------
-+-------+------+--------+-------+-----------+----+-----------+-------------
------+---------------+--------------------+------------+-----------+-------
-+----------+---------+---------+--------------+---------------+--------+--
---------------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|      ORDERDAT
E| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|        CUSTOM
ERNAME|           PHONE|        ADDRESSLINE1|ADDRESSLINE2|       CITY|   STAT
E|POSTALCODE|  COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
TOTAL_REVENUE|
+-----------+--------------+---------+--------------+-------+-------------
-+-------+------+--------+-------+-----------+----+-----------+-------------
------+---------------+--------------------+------------+-----------+-------
-+----------+---------+---------+--------------+---------------+--------+--
---------------+
|      10107|            30|     95.7|             2| 2871.0| 2/24/2003 0:0
0|Shipped|     1|       2|   2003|Motorcycles|  95|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|            Yu|            Kwai|   Small|
2871.0|
|      10121|            34|    81.35|             5| 2765.9|  5/7/2003 0:0
0|Shipped|     2|       5|   2003|Motorcycles|  95|   S10_1678|  Reims Collec
tables|      26.47.1555|  59 rue de l'Abbaye|          NA|      Reims|      N
A|     51100|   France|     EMEA|        Henriot|            Paul|   Small|27
65.8999999999996|
|      10134|            41|    94.74|             2|3884.34|  7/1/2003 0:0
0|Shipped|     3|       7|   2003|Motorcycles|  95|   S10_1678|     Lyon Souv
eniers|+33 1 46 62 7555|27 rue du Colonel...|          NA|      Paris|      N
A|     75508|   France|     EMEA|       Da Cunha|          Daniel|  Medium|38
84.3399999999997|
|      10145|            45|    83.26|             6| 3746.7| 8/25/2003 0:0
0|Shipped|     3|       8|   2003|Motorcycles|  95|   S10_1678|   Toys4GrownU
ps.com|      6265557265|  78934 Hillside Dr.|          NA|   Pasadena|      C
```

```
A|     90003|      USA|       NA|          Young|           Julie|  Medium|37
46.7000000000003|
|      10168|             36|    96.66|              1|3479.76|10/28/2003 0:0
0|Shipped|     4|      10|   2003|Motorcycles|  95|   S10_1678|Technics Store
s Inc.|      6505556809|   9408 Furth Circle|          NA| Burlingame|      C
A|     94217|      USA|       NA|         Hirano|            Juri|  Medium|34
79.7599999999998|
|      10180|             29|    86.13|              9|2497.77|11/11/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|Daedalus Desig
ns ...|      20.16.1555|184, chausse de T...|          NA|      Lille|      N
A|     59000|   France|     EMEA|          Rance|         Martine|   Small|
2497.77|
|      10188|             48|    100.0|             1|5512.32|11/18/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|        Herkku
Gifts|   +47 2267 3215|Drammen 121, PR 7...|          NA|     Bergen|      NA
|    N 5804|   Norway|     EMEA|         Oeztan|          Veysel|  Medium|
4800.0|
|      10211|             41|    100.0|            14|4708.44| 1/15/2004 0:0
0|Shipped|     1|       1|   2004|Motorcycles|  95|   S10_1678|    Auto Canal
Petit|  (1) 47.55.6555|   25, rue Lauriston|          NA|      Paris|      NA
|     75016|   France|     EMEA|        Perrier|       Dominique|  Medium|
4100.0|
|      10223|             37|    100.0|             1|3965.66| 2/20/2004 0:0
0|Shipped|     1|       2|   2004|Motorcycles|  95|   S10_1678|Australian Col
lec...|    03 9520 4555|   636 St Kilda Road|     Level 3|  Melbourne|Victori
a|      3004|Australia|     APAC|       Ferguson|           Peter|  Medium|
3700.0|
|      10237|             23|    100.0|             7|2333.12|  4/5/2004 0:0
0|Shipped|     2|       4|   2004|Motorcycles|  95|   S10_1678|     Vitachrom
e Inc.|      2125551500|   2678 Kingston Rd.|   Suite 101|        NYC|      N
Y|     10022|      USA|       NA|          Frick|         Michael|   Small|
2300.0|
|      10251|             28|    100.0|             2|3188.64| 5/18/2004 0:0
0|Shipped|     2|       5|   2004|Motorcycles|  95|   S10_1678|Tekni Collecta
ble...|      2015559350|       7476 Moss Rd.|          NA|     Newark|      N
J|     94019|      USA|       NA|          Brown|         William|  Medium|
2800.0|
|      10263|             34|    100.0|             2|3676.76| 6/28/2004 0:0
0|Shipped|     2|       6|   2004|Motorcycles|  95|   S10_1678|     Gift Depo
t Inc.|      2035552570| 25593 South Bay Ln.|          NA|Bridgewater|      C
T|     97562|      USA|       NA|           King|           Julie|  Medium|
3400.0|
|      10275|             45|    92.83|             1|4177.35| 7/23/2004 0:0
0|Shipped|     3|       7|   2004|Motorcycles|  95|   S10_1678|   La Rochelle
Gifts|      40.67.8555|67, rue des Cinqu...|          NA|     Nantes|      NA
|     44000|   France|     EMEA|        Labrune|          Janine|  Medium|
4177.35|
|      10285|             36|    100.0|             6|4099.68| 8/27/2004 0:0
0|Shipped|     3|       8|   2004|Motorcycles|  95|   S10_1678|Marta's Replic
as Co.|      6175558555| 39323 Spinnaker Dr.|          NA|  Cambridge|      M
A|     51247|      USA|       NA|      Hernandez|           Marta|  Medium|
3600.0|
|      10299|             23|    100.0|             9|2597.39| 9/30/2004 0:0
0|Shipped|     3|       9|   2004|Motorcycles|  95|   S10_1678|Toys of Finlan
d, Co.|     90-224 8555|       Keskuskatu 45|          NA|   Helsinki|      N
A|     21240|  Finland|     EMEA|      Karttunen|           Matti|   Small|
2300.0|
|      10309|             41|    100.0|             5|4394.38|10/15/2004 0:0
0|Shipped|     4|      10|   2004|Motorcycles|  95|   S10_1678|  Baane Mini I
mports|      07-98 9555|Erling Skakkes ga...|          NA|    Stavern|      N
A|      4110|   Norway|     EMEA|     Bergulfsen|           Jonas|  Medium|
4100.0|
|      10318|             46|    94.74|             1|4358.04| 11/2/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Diecast Classi
cs ...|      2155551555|    7586 Pompton St.|          NA|  Allentown|      P
A|     70267|      USA|       NA|             Yu|           Kyung|  Medium|
4358.04|
|      10329|             42|    100.0|             1|4396.14|11/15/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|   Land of Toy
```

```
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|  Medium|
4200.0|
|      10341|             41|    100.0|              9|7737.93|11/24/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Salzburg Colle
cta...|       6562-9555|         Geislweg 14|          NA|   Salzburg|      N
A|      5020|  Austria|     EMEA|          Pipps|           Georg|   Large|
4100.0|
|      10361|             20|    72.55|             13| 1451.0|12/17/2004 0:0
0|Shipped|     4|      12|   2004|Motorcycles|  95|   S10_1678|Souveniers And
Th...| +61 2 9495 8555|Monitor Money Bui...|     Level 6|  Chatswood|     NSW
|      2067|Australia|     APAC|         Huxley|          Adrian|   Small|
1451.0|
+----------+--------------+---------+--------------+-------+-------------
-+-------+------+--------+-------+-----------+----+-----------+-------------
------+----------------+--------------------+------------+-----------+-------
-+----------+---------+---------+--------------+----------------+--------+--
----------------+
only showing top 20 rows
```

In [16]:

```
cleaned_sales_data.groupBy('COUNTRY').agg(sum('TOTAL_REVENUE').alias('TOTAL_R
.orderBy('TOTAL_REVENUE', ascending=False).show()
```

```
+-----------+------------------+
|    COUNTRY|     TOTAL_REVENUE|
+-----------+------------------+
|        USA|2765243.1999999993|
|      Spain|1021705.9700000002|
|     France| 919257.8499999997|
|  Australia|521598.45999999985|
|         UK|413203.33999999997|
|      Italy| 309402.8699999999|
|    Finland|268714.70000000007|
|     Norway| 246115.8000000001|
|  Singapore| 227985.5000000001|
|     Canada|193504.34000000003|
|    Denmark|         192747.63|
|    Germany|         178689.08|
|     Sweden|174264.10000000006|
|    Austria|172793.05000000002|
|      Japan|153076.68999999994|
|    Belgium|          94528.88|
|Switzerland| 93344.90999999999|
|Philippines| 80291.16999999998|
|    Ireland|          43237.24|
+-----------+------------------+
```

g) Determine the top 5 products with the highest total sales revenue using Spark DataFrame?

In [47]:

```
cleaned_sales_data.groupBy('PRODUCTCODE').agg(sum('TOTAL_REVENUE').alias('TOT
```

```
+-----------+-------------+
|PRODUCTCODE|TOTAL_REVENUE|
+-----------+-------------+
|   S18_3232|    174026.63|
|   S18_4600|     101835.0|
|   S24_3856|     99989.89|
|   S24_2300|      99600.0|
|   S18_2238|      96300.0|
+-----------+-------------+
only showing top 5 rows
```

h) Find the average order quantity for each product using groupBy and agg operations?

In [18]:
```
from pyspark.sql.functions import avg
cleaned_sales_data.groupBy('PRODUCTCODE').agg(avg('QUANTITYORDERED').alias('A
```

```
+-----------+------------------+
|PRODUCTCODE|           AVG_QTY|
+-----------+------------------+
|   S18_4600| 38.18518518518518|
|   S18_1749| 36.80952380952381|
|   S12_3891| 35.42307692307692|
|   S18_2248|34.095238095238095|
|  S700_1138| 34.69230769230769|
|   S32_1268|32.333333333333336|
|   S12_1099|33.208333333333336|
|   S18_2795|             29.64|
|   S24_1937|            33.625|
|   S32_3522| 35.53846153846154|
|   S18_1097| 35.46153846153846|
|   S18_1662| 36.15384615384615|
|   S12_1666| 35.34615384615385|
|   S24_3969|33.714285714285715|
|   S24_1578|35.333333333333336|
|   S24_4048| 32.46153846153846|
|   S18_3320|34.333333333333336|
|   S24_3816|             33.92|
|   S18_3136|              31.8|
|   S32_2509|34.107142857142854|
+-----------+------------------+
only showing top 20 rows
```

i) Using Spark DataFrame, filter orders where the SALES value exceeds 10,000 and sort the results by the ORDERDATE column?

In [19]:
```
from pyspark.sql.functions import col, to_timestamp

cleaned_sales_data.withColumn('ORDERDATE', to_timestamp(col('ORDERDATE'), 'M/
.filter((col('SALES') > 10000))\
.orderBy('ORDERDATE')\
.select(['PRODUCTCODE', 'SALES', 'ORDERDATE']).show()
```

```
+-----------+-------+-------------------+
|PRODUCTCODE|  SALES|          ORDERDATE|
+-----------+-------+-------------------+
|   S12_1108|11279.2|2003-06-03 00:00:00|
|   S10_1949|10993.5|2003-09-19 00:00:00|
|   S12_1108|10606.2|2004-05-05 00:00:00|
|   S10_1949|10172.7|2004-10-11 00:00:00|
|   S10_1949|11623.7|2004-10-21 00:00:00|
|   S18_2325|12536.5|2004-11-04 00:00:00|
|   S24_3151|10758.0|2004-11-23 00:00:00|
|   S24_4278|10039.6|2005-02-03 00:00:00|
|  S700_1691|10066.6|2005-03-03 00:00:00|
|   S10_4698|11886.6|2005-04-08 00:00:00|
|   S24_3856|11739.7|2005-04-14 00:00:00|
|   S18_3685|10468.9|2005-04-15 00:00:00|
|   S18_1749|14082.8|2005-04-22 00:00:00|
|   S18_3232|11887.8|2005-05-03 00:00:00|
|   S10_1949|12001.0|2005-05-31 00:00:00|
+-----------+-------+-------------------+
```

j) Filter out rows where the STATUS is 'Cancelled' and calculate the total sales from the remaining orders?

In [20]:
```
cleaned_sales_data\
.filter(col('STATUS') != 'Cancelled')\
.agg(sum('SALES').alias('TOTAL_SALES'))\
.show()
```

```
+-----------------+
|      TOTAL_SALES|
+-----------------+
|9565734.230000012|
+-----------------+
```

k) Use Spark Data Frame transformations to derive the yearly sales for each customer (CUSTOMERNAME) based on the ORDERDATE column?

In [21]:
```
cleaned_sales_data.select(['CUSTOMERNAME']).distinct().orderBy('CUSTOMERNAME'
```

```
+--------------------+
|        CUSTOMERNAME|
+--------------------+
|      AV Stores, Co.|
|        Alpha Cognac|
|  Amica Models & Co.|
|Anna's Decoration...|
|   Atelier graphique|
|Australian Collec...|
|Australian Collec...|
|Australian Gift N...|
|  Auto Assoc. & Cie.|
|    Auto Canal Petit|
|Auto-Moto Classic...|
|  Baane Mini Imports|
|Bavarian Collecta...|
|Blauer See Auto, Co.|
|   Boards & Toys Co.|
|         CAF Imports|
|Cambridge Collect...|
|Canadian Gift Exc...|
|Classic Gift Idea...|
|Classic Legends Inc.|
+--------------------+
only showing top 20 rows
```

In [22]:
```
from pyspark.sql.functions import to_date, year, to_timestamp

yearly_data = cleaned_sales_data.withColumn("YEAR", year(to_timestamp(col("OR

yearly_data.groupBy(['CUSTOMERNAME', 'YEAR'])\
.agg(sum('SALES').alias('TOTAL_SALES'))\
.orderBy(['CUSTOMERNAME','YEAR']).show()
```

```
+--------------------+----+-----------------+
|        CUSTOMERNAME|YEAR|      TOTAL_SALES|
+--------------------+----+-----------------+
|      AV Stores, Co.|2003| 51017.91999999999|
|      AV Stores, Co.|2004|         106789.89|
|        Alpha Cognac|2003| 55349.31999999999|
|        Alpha Cognac|2005|15139.119999999999|
```

```
|  Amica Models & Co.|2004| 94117.26000000002|
|Anna's Decoration...|2003| 88983.70999999999|
|Anna's Decoration...|2005|          65012.42|
|   Atelier graphique|2003|           16560.3|
|   Atelier graphique|2004|           7619.66|
|Australian Collec...|2003|          37878.55|
|Australian Collec...|2004|          12334.82|
|Australian Collec...|2005|          14378.09|
|Australian Collec...|2003|60135.840000000004|
|Australian Collec...|2004|140859.56999999998|
|Australian Gift N...|2003|37739.090000000004|
|Australian Gift N...|2005|          21730.03|
|  Auto Assoc. & Cie.|2004| 64834.32000000001|
|    Auto Canal Petit|2004| 79103.85999999999|
|    Auto Canal Petit|2005|           14066.8|
|Auto-Moto Classic...|2003|           7277.35|
+--------------------+----+------------------+
only showing top 20 rows
```

l) Add a new column to the DataFrame that categorizes orders as High, Medium, or Low sales based on the SALES value?

In [23]:

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

percentile_33, percentile_67 = cleaned_sales_data.approxQuantile("SALES", [0.

sales_data_with_category = cleaned_sales_data.withColumn(
    "CATEGORY",
    when(col("SALES") > percentile_67, "High")\
    .when(col("SALES") > percentile_33, "Medium")
    .otherwise("Low")
)


sales_data_with_category.select(['SALES', 'CATEGORY']).show(100)
```

```
+-------+--------+
|  SALES|CATEGORY|
+-------+--------+
| 2871.0|  Medium|
| 2765.9|  Medium|
|3884.34|  Medium|
| 3746.7|  Medium|
|3479.76|  Medium|
|2497.77|  Medium|
|5512.32|    High|
|4708.44|    High|
|3965.66|    High|
|2333.12|     Low|
|3188.64|  Medium|
|3676.76|  Medium|
|4177.35|    High|
|4099.68|    High|
|2597.39|  Medium|
|4394.38|    High|
|4358.04|    High|
|4396.14|    High|
|7737.93|    High|
| 1451.0|     Low|
| 733.11|     Low|
|3207.12|  Medium|
|2434.56|     Low|
|7516.08|    High|
|5404.62|    High|
|7209.11|    High|
```

```
|7329.06|    High|
| 7374.1|    High|
|10993.5|    High|
|4860.24|    High|
|8014.82|    High|
|5372.57|    High|
|7290.36|    High|
|9064.89|    High|
| 6075.3|    High|
|6463.23|    High|
|6120.34|    High|
|7680.64|    High|
|4905.39|    High|
|8014.82|    High|
|7136.19|    High|
|10172.7|    High|
|11623.7|    High|
| 6000.4|    High|
| 3944.7|    High|
|5691.84|    High|
|4514.92|    High|
|2416.56|     Low|
|4140.23|    High|
|12001.0|    High|
|3896.49|  Medium|
|2793.86|  Medium|
|3307.77|  Medium|
|5192.95|    High|
|3660.93|  Medium|
| 4695.6|    High|
|3660.92|  Medium|
|3009.09|  Medium|
|5422.39|    High|
|2852.08|  Medium|
|5756.52|    High|
| 4472.0|    High|
|2904.44|  Medium|
|6484.59|    High|
|3757.26|  Medium|
|4043.96|    High|
| 5566.5|    High|
| 3176.0|  Medium|
| 2756.8|  Medium|
| 1329.9|     Low|
|5288.01|    High|
| 2225.5|     Low|
| 5833.8|    High|
|5887.35|    High|
|6065.55|    High|
|9264.86|    High|
|7023.98|    High|
|5176.38|    High|
| 4183.0|    High|
| 8892.9|    High|
| 8714.7|    High|
| 6123.4|    High|
|9774.03|    High|
| 7023.9|    High|
|7078.23|    High|
|8336.94|    High|
|6901.92|    High|
|5438.07|    High|
|6683.34|    High|
| 4570.4|    High|
|7667.14|    High|
| 5868.2|    High|
|2990.13|  Medium|
| 3664.1|  Medium|
|3834.38|  Medium|
```

```
|1822.17|     Low|
|11886.6|    High|
|9218.16|    High|
| 7208.0|    High|
| 5004.8|    High|
+-------+--------+
only showing top 100 rows
```

m) Assume , If you have another DataFrame with customer demographic data, how would you perform a join to compute the total sales per demographic group?

In [24]:
```
demographic_data = [
    ("Toms Spezialitten", "Germany", 480000, "Europe"),
    ("Oulu Toy Supplies", "Finland", 55000, "Europe"),
    ("Petit Auto", "Belgium", 160000, "Europe"),
    ("Corporate Gift Ideas Co.", "USA", 331000, "North America"),
    ("Cambridge Collectables Co.", "USA", 310000, "North America"),
    ]


demographic_df = spark.createDataFrame(demographic_data, ["CUSTOMERNAME", "CO

joined_df = cleaned_sales_data.join(demographic_df, on=["CUSTOMERNAME",'COUNT

sales_by_country = joined_df.groupBy("COUNTRY","CUSTOMERNAME",'INCOME', 'REGI


sales_by_country.show()
```

```
+-------+--------------------+------+-------------+-----------------+
|COUNTRY|        CUSTOMERNAME|INCOME|       REGION|      TOTAL_SALES|
+-------+--------------------+------+-------------+-----------------+
|    USA|Cambridge Collect...|310000|North America|         36163.62|
|Belgium|          Petit Auto|160000|       Europe|74972.51999999999|
+-------+--------------------+------+-------------+-----------------+
```

n) Can you implement a cumulative distribution function (CDF) over the SALES value for each CUSTOMERNAME? What insights can you gather from analyzing the CDF distribution for each customer?

In [25]:
```
cleaned_sales_data.printSchema
```

Out[25]:
```
<bound method DataFrame.printSchema of DataFrame[ORDERNUMBER: int, QUANTITYOR
DERED: int, PRICEEACH: double, ORDERLINENUMBER: int, SALES: double, ORDERDAT
E: string, STATUS: string, QTR_ID: int, MONTH_ID: int, YEAR_ID: int, PRODUCTL
INE: string, MSRP: int, PRODUCTCODE: string, CUSTOMERNAME: string, PHONE: str
ing, ADDRESSLINE1: string, ADDRESSLINE2: string, CITY: string, STATE: string,
POSTALCODE: string, COUNTRY: string, TERRITORY: string, CONTACTLASTNAME: stri
ng, CONTACTFIRSTNAME: string, DEALSIZE: string, TOTAL_REVENUE: double]>
```

In [26]:
```
from pyspark.sql.functions import col, count, rank
from pyspark.sql import Window
from pyspark.sql.functions import cume_dist
import numpy as np

window_w = Window.partitionBy("CUSTOMERNAME").orderBy("SALES")
```

```
cdf_df = cleaned_sales_data.withColumn(
    "CDF",
    cume_dist().over(window_w)
)

cdf_df.select(["CUSTOMERNAME", "SALES", "CDF"]).show()
```

```
+------------------+-------+------------------+
|       CUSTOMERNAME|  SALES|               CDF|
+------------------+-------+------------------+
|Suominen Souveniers| 891.03|0.03333333333333333|
|Suominen Souveniers| 1086.6|0.06666666666666667|
|Suominen Souveniers|1103.76|               0.1|
|Suominen Souveniers|1629.04|0.13333333333333333|
|Suominen Souveniers| 1988.4|0.16666666666666666|
|Suominen Souveniers|2140.11|               0.2|
|Suominen Souveniers|2447.76|0.23333333333333334|
|Suominen Souveniers|2632.89|0.26666666666666666|
|Suominen Souveniers| 2773.8|               0.3|
|Suominen Souveniers|2775.08| 0.3333333333333333|
|Suominen Souveniers|2817.87|0.36666666666666664|
|Suominen Souveniers|2851.84|               0.4|
|Suominen Souveniers|2931.98|0.43333333333333335|
|Suominen Souveniers|3128.65| 0.4666666666666667|
|Suominen Souveniers|3288.82|               0.5|
|Suominen Souveniers|3595.62| 0.5333333333333333|
|Suominen Souveniers|3686.54| 0.5666666666666667|
|Suominen Souveniers| 3784.8|               0.6|
|Suominen Souveniers| 4068.7| 0.6333333333333333|
|Suominen Souveniers|4142.64| 0.6666666666666666|
+------------------+-------+------------------+
only showing top 20 rows
```

50 percentage of Suominen Souveniers sales is below 3288.82

20% of customers have total upto 2000, indicating low spending customers\ median value is around 4000, half of the customers have total sales below 4000\ after 8000 there are fewer high spending customers

o) Write spark dataframe code to rank products by total revenue within each country (COUNTRY)?

In [29]:

```
from pyspark.sql.functions import dense_rank, desc
from pyspark.sql.window import Window

cleaned_sales_data\
.groupBy(['PRODUCTCODE', 'COUNTRY']).agg(sum("TOTAL_REVENUE").alias("TOTAL_RE
.withColumn('RANK', dense_rank().over(Window.partitionBy('COUNTRY').orderBy(d
.select(['COUNTRY', 'PRODUCTCODE','RANK', 'TOTAL_REVENUE']).show(100)
```

```
+-----------+-----------+----+------------------+
|    COUNTRY|PRODUCTCODE|RANK|     TOTAL_REVENUE|
+-----------+-----------+----+------------------+
|     Sweden|   S18_4600|   1|            9700.0|
|     Sweden|   S24_2300|   2|            9000.0|
|     Sweden|   S24_2011|   3|            7400.0|
|     Sweden|   S18_2949|   4|            7000.0|
|     Sweden|   S10_1949|   5|            6600.0|
|     Sweden|   S12_1099|   6|           5675.04|
|     Sweden|   S10_4962|   7|            5600.0|
|     Sweden|  S700_1138|   8|5579.620000000001|
|     Sweden|   S12_3990|   9|           5319.32|
|     Sweden|   S12_3380|  10|            5309.5|
|     Sweden|   S24_3151|  11|5113.049999999999|
|     Sweden|   S12_4675|  12|            4700.0|
```

```
|      Sweden|   S18_2319|  13|            4600.0|
|      Sweden|   S24_1578|  14|            4500.0|
|      Sweden|   S10_4757|  15|            4400.0|
|      Sweden|   S18_4522|  16|            4300.5|
|      Sweden|   S18_1662|  17|            4300.0|
|      Sweden|   S24_3816|  18|4276.9400000000005|
|      Sweden|   S18_1097|  19|            4100.0|
|      Sweden|   S12_1666|  19|            4100.0|
|      Sweden|   S24_2000|  20|3988.6000000000004|
|      Sweden|   S18_2625|  21|            3900.0|
|      Sweden|   S18_1889|  22|3881.7799999999997|
|      Sweden|   S18_2432|  23|            3587.9|
|      Sweden|   S18_3856|  24|            3400.0|
|      Sweden|   S18_3029|  25|3363.5200000000004|
|      Sweden|  S700_2610|  26|           3225.06|
|      Sweden|   S18_3259|  27|            3200.0|
|      Sweden|  S700_2047|  28|            2900.0|
|      Sweden|  S700_2824|  29|            2800.0|
|      Sweden|  S700_3962|  29|            2800.0|
|      Sweden|   S18_3482|  30|            2500.0|
|      Sweden|  S700_3505|  31|            2400.0|
|      Sweden|   S18_3136|  32|            2300.0|
|      Sweden|   S32_3522|  33|            2296.0|
|      Sweden|   S32_1268|  34|           2178.54|
|      Sweden|   S24_3420|  35|           2019.84|
|      Sweden|   S18_3232|  36|            2000.0|
|      Sweden|   S18_2957|  37|1871.8300000000002|
|      Sweden|   S72_3212|  38|           1846.42|
|      Sweden|   S72_1253|  39|            1668.4|
|      Sweden|   S24_3371|  40|           1591.98|
|      Sweden|   S18_4668|  41|           1502.78|
|      Sweden|   S24_2841|  42|           1467.48|
| Philippines|   S18_4721|   1|            9400.0|
| Philippines|   S18_3482|   2|            6600.0|
| Philippines|   S24_2360|   3|           5463.71|
| Philippines|   S18_3782|   4| 4933.379999999999|
| Philippines|   S12_3380|   5|            4500.0|
| Philippines|  S700_2466|   6|            4200.0|
| Philippines|   S18_1662|   6|            4200.0|
| Philippines|   S24_3856|   7|            4000.0|
| Philippines|   S12_4675|   8|            3600.0|
| Philippines|   S12_3990|   9|3485.8199999999997|
| Philippines|   S12_1099|  10|            3300.0|
| Philippines|   S24_2841|  11|3255.3599999999997|
| Philippines|   S18_1889|  12|           3130.82|
| Philippines|   S32_4485|  13|            3100.0|
| Philippines|   S24_3949|  14|           2846.17|
| Philippines|   S50_4713|  15|           2821.66|
| Philippines|  S700_4002|  16|            2546.8|
| Philippines|   S24_4620|  17|2130.0099999999998|
| Philippines|   S24_3420|  18|1935.0900000000001|
| Philippines|   S24_3371|  19|            1892.1|
| Philippines|   S18_3278|  20|            1777.1|
| Philippines|   S32_2206|  21|           1173.15|
|   Singapore|   S18_4600|   1|           12800.0|
|   Singapore|   S24_2300|   2|           10700.0|
|   Singapore|   S12_4473|   3|            9400.0|
|   Singapore|   S10_4962|   3|            9400.0|
|   Singapore|   S18_2319|   4|7979.9400000000005|
|   Singapore|   S12_3148|   5|            7700.0|
|   Singapore|   S12_1108|   5|            7700.0|
|   Singapore|   S50_1392|   6|           7456.16|
|   Singapore|   S18_3232|   7|            7400.0|
|   Singapore|   S18_3259|   8|            7011.8|
|   Singapore|   S12_3891|   9|            6600.0|
|   Singapore|   S32_1268|  10|            6000.6|
|   Singapore|   S18_2238|  11|            5900.0|
|   Singapore|   S18_2432|  12|            5829.0|
|   Singapore|   S32_3522|  13|           5270.77|
```

```
|  Singapore|   S18_4027|  14|             5000.0|
|  Singapore|   S24_2887|  15|             4800.0|
|  Singapore|  S700_2824|  16|             4728.0|
|  Singapore|   S10_1949|  17|             4500.0|
|  Singapore|   S18_2949|  18|            4285.46|
|  Singapore|   S32_1374|  19|             4175.6|
|  Singapore|   S24_3432|  20|             4100.0|
|  Singapore|   S24_2011|  20|             4100.0|
|  Singapore|   S32_2509|  21|            3960.72|
|  Singapore|  S700_3962|  22|             3753.9|
|  Singapore|   S18_2325|  23|             3700.0|
|  Singapore|   S50_1514|  24|            3571.05|
|  Singapore|   S18_1097|  25|             3400.0|
|  Singapore|   S18_1749|  26|             3200.0|
|  Singapore|   S24_1444|  27|            3044.96|
|  Singapore|  S700_1938|  28|            3027.84|
|  Singapore|   S12_1666|  29|             3000.0|
|  Singapore|   S24_1937|  30|            2908.87|
|  Singapore|   S18_4409|  31|2834.6499999999996|
+-----------+-----------+----+------------------+
only showing top 100 rows
```

p) Calculate a running total of SALES for each customer and show the top 5 customers by this cumulative total?

In [30]:

```
w = Window.partitionBy('CUSTOMERNAME')\
  .orderBy('SALES')\
  .rowsBetween(Window.unboundedPreceding, Window.currentRow)

cleaned_sales_data\
.withColumn('RUNNINGTOTAL', sum('SALES').over(w))\
.select(['CUSTOMERNAME', 'SALES', 'RUNNINGTOTAL']).show()
```

```
+------------------+-------+------------------+
|       CUSTOMERNAME|  SALES|      RUNNINGTOTAL|
+------------------+-------+------------------+
|Suominen Souveniers| 891.03|            891.03|
|Suominen Souveniers| 1086.6|1977.6299999999999|
|Suominen Souveniers|1103.76|           3081.39|
|Suominen Souveniers|1629.04|           4710.43|
|Suominen Souveniers| 1988.4|           6698.83|
|Suominen Souveniers|2140.11|           8838.94|
|Suominen Souveniers|2447.76|           11286.7|
|Suominen Souveniers|2632.89|          13919.59|
|Suominen Souveniers| 2773.8|          16693.39|
|Suominen Souveniers|2775.08|          19468.47|
|Suominen Souveniers|2817.87|          22286.34|
|Suominen Souveniers|2851.84|          25138.18|
|Suominen Souveniers|2931.98|          28070.16|
|Suominen Souveniers|3128.65|          31198.81|
|Suominen Souveniers|3288.82|34487.630000000005|
|Suominen Souveniers|3595.62| 38083.25000000001|
|Suominen Souveniers|3686.54| 41769.79000000001|
|Suominen Souveniers| 3784.8| 45554.59000000001|
|Suominen Souveniers| 4068.7| 49623.29000000001|
|Suominen Souveniers|4142.64| 53765.93000000001|
+------------------+-------+------------------+
only showing top 20 rows
```

q) Find and handle Invalid and Outliers values in entire DataFrame. [Check for only continuous dataset].

In [31]:

```
cleaned_sales_data.printSchema
```

Out[31]:
```
<bound method DataFrame.printSchema of DataFrame[ORDERNUMBER: int, QUANTITYOR
DERED: int, PRICEEACH: double, ORDERLINENUMBER: int, SALES: double, ORDERDAT
E: string, STATUS: string, QTR_ID: int, MONTH_ID: int, YEAR_ID: int, PRODUCTL
INE: string, MSRP: int, PRODUCTCODE: string, CUSTOMERNAME: string, PHONE: str
ing, ADDRESSLINE1: string, ADDRESSLINE2: string, CITY: string, STATE: string,
POSTALCODE: string, COUNTRY: string, TERRITORY: string, CONTACTLASTNAME: stri
ng, CONTACTFIRSTNAME: string, DEALSIZE: string, TOTAL_REVENUE: double]>
```

In [ ]:
```
```

In [32]:
```
quantiles = cleaned_sales_data.approxQuantile("SALES", [0.25, 0.75], 0.05)
Q1, Q3 = quantiles[0], quantiles[1]

IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

cleaned_sales_data.filter((col("SALES") < lower_bound) | (col("SALES") > uppe
```

Out[32]:
```
122
```

In [33]:
```
capped_sales_data = cleaned_sales_data.withColumn(
    "SALES",
    when(col("SALES") < lower_bound, lower_bound)
    .when(col("SALES") > upper_bound, upper_bound)
    .otherwise(col("SALES"))
)

capped_sales_data.show()
```

```
+-----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+---------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|      ORDERDAT
E| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|        CUSTOM
ERNAME|           PHONE|        ADDRESSLINE1|ADDRESSLINE2|       CITY|   STAT
E|POSTALCODE|  COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
TOTAL_REVENUE|
+-----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+---------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
|      10107|             30|     95.7|              2| 2871.0| 2/24/2003 0:0
0|Shipped|     1|       2|   2003|Motorcycles|  95|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|   Small|
2871.0|
|      10121|             34|    81.35|              5| 2765.9|  5/7/2003 0:0
0|Shipped|     2|       5|   2003|Motorcycles|  95|   S10_1678|  Reims Collec
tables|      26.47.1555|  59 rue de l'Abbaye|          NA|      Reims|      N
A|     51100|   France|     EMEA|        Henriot|            Paul|   Small|27
65.8999999999996|
|      10134|             41|    94.74|              2|3884.34|  7/1/2003 0:0
0|Shipped|     3|       7|   2003|Motorcycles|  95|   S10_1678|     Lyon Souv
eniers|+33 1 46 62 7555|27 rue du Colonel...|          NA|      Paris|      N
A|     75508|   France|     EMEA|       Da Cunha|          Daniel|  Medium|38
84.3399999999997|
|      10145|             45|    83.26|              6| 3746.7| 8/25/2003 0:0
0|Shipped|     3|       8|   2003|Motorcycles|  95|   S10_1678|   Toys4GrownU
```

```
ps.com|      6265557265|  78934 Hillside Dr.|          NA|   Pasadena|      C
A|     90003|      USA|       NA|          Young|           Julie|  Medium|37
46.7000000000003|
|      10168|             36|    96.66|              1|3479.76|10/28/2003 0:0
0|Shipped|     4|      10|   2003|Motorcycles|  95|   S10_1678|Technics Store
s Inc.|      6505556809|   9408 Furth Circle|          NA| Burlingame|      C
A|     94217|      USA|       NA|         Hirano|            Juri|  Medium|34
79.7599999999998|
|      10180|             29|    86.13|              9|2497.77|11/11/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|Daedalus Desig
ns ...|      20.16.1555|184, chausse de T...|          NA|      Lille|      N
A|     59000|   France|     EMEA|          Rance|         Martine|   Small|
2497.77|
|      10188|             48|    100.0|             1|5512.32|11/18/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|        Herkku
Gifts|   +47 2267 3215|Drammen 121, PR 7...|          NA|     Bergen|      NA
|    N 5804|   Norway|     EMEA|         Oeztan|          Veysel|  Medium|
4800.0|
|      10211|             41|    100.0|            14|4708.44| 1/15/2004 0:0
0|Shipped|     1|       1|   2004|Motorcycles|  95|   S10_1678|    Auto Canal
Petit|  (1) 47.55.6555|   25, rue Lauriston|          NA|      Paris|      NA
|     75016|   France|     EMEA|        Perrier|       Dominique|  Medium|
4100.0|
|      10223|             37|    100.0|             1|3965.66| 2/20/2004 0:0
0|Shipped|     1|       2|   2004|Motorcycles|  95|   S10_1678|Australian Col
lec...|    03 9520 4555|   636 St Kilda Road|     Level 3|  Melbourne|Victori
a|      3004|Australia|     APAC|       Ferguson|           Peter|  Medium|
3700.0|
|      10237|             23|    100.0|             7|2333.12|  4/5/2004 0:0
0|Shipped|     2|       4|   2004|Motorcycles|  95|   S10_1678|     Vitachrom
e Inc.|      2125551500|   2678 Kingston Rd.|   Suite 101|        NYC|      N
Y|     10022|      USA|       NA|          Frick|         Michael|   Small|
2300.0|
|      10251|             28|    100.0|             2|3188.64| 5/18/2004 0:0
0|Shipped|     2|       5|   2004|Motorcycles|  95|   S10_1678|Tekni Collecta
ble...|      2015559350|       7476 Moss Rd.|          NA|     Newark|      N
J|     94019|      USA|       NA|          Brown|         William|  Medium|
2800.0|
|      10263|             34|    100.0|             2|3676.76| 6/28/2004 0:0
0|Shipped|     2|       6|   2004|Motorcycles|  95|   S10_1678|     Gift Depo
t Inc.|      2035552570| 25593 South Bay Ln.|          NA|Bridgewater|      C
T|     97562|      USA|       NA|           King|           Julie|  Medium|
3400.0|
|      10275|             45|    92.83|             1|4177.35| 7/23/2004 0:0
0|Shipped|     3|       7|   2004|Motorcycles|  95|   S10_1678|   La Rochelle
Gifts|      40.67.8555|67, rue des Cinqu...|          NA|     Nantes|      NA
|     44000|   France|     EMEA|        Labrune|          Janine|  Medium|
4177.35|
|      10285|             36|    100.0|             6|4099.68| 8/27/2004 0:0
0|Shipped|     3|       8|   2004|Motorcycles|  95|   S10_1678|Marta's Replic
as Co.|      6175558555| 39323 Spinnaker Dr.|          NA|  Cambridge|      M
A|     51247|      USA|       NA|      Hernandez|           Marta|  Medium|
3600.0|
|      10299|             23|    100.0|             9|2597.39| 9/30/2004 0:0
0|Shipped|     3|       9|   2004|Motorcycles|  95|   S10_1678|Toys of Finlan
d, Co.|     90-224 8555|       Keskuskatu 45|          NA|   Helsinki|      N
A|     21240|  Finland|     EMEA|      Karttunen|           Matti|   Small|
2300.0|
|      10309|             41|    100.0|             5|4394.38|10/15/2004 0:0
0|Shipped|     4|      10|   2004|Motorcycles|  95|   S10_1678|  Baane Mini I
mports|      07-98 9555|Erling Skakkes ga...|          NA|    Stavern|      N
A|      4110|   Norway|     EMEA|     Bergulfsen|           Jonas|  Medium|
4100.0|
|      10318|             46|    94.74|             1|4358.04| 11/2/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Diecast Classi
cs ...|      2155551555|    7586 Pompton St.|          NA|  Allentown|      P
A|     70267|      USA|       NA|             Yu|           Kyung|  Medium|
4358.04|
|      10329|             42|    100.0|             1|4396.14|11/15/2004 0:0
```

```
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|  Medium|
4200.0|
|      10341|             41|    100.0|              9|7211.16|11/24/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Salzburg Colle
cta...|       6562-9555|         Geislweg 14|          NA|   Salzburg|      N
A|      5020|  Austria|     EMEA|          Pipps|           Georg|   Large|
4100.0|
|      10361|             20|    72.55|             13| 1451.0|12/17/2004 0:0
0|Shipped|     4|      12|   2004|Motorcycles|  95|   S10_1678|Souveniers And
Th...| +61 2 9495 8555|Monitor Money Bui...|     Level 6|  Chatswood|     NSW
|      2067|Australia|     APAC|         Huxley|          Adrian|   Small|
1451.0|
+----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+----------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
only showing top 20 rows
```

In [34]:

```
def capp_outliers(df, column):
    quantiles = df.approxQuantile(column, [0.25, 0.75], 0.05)
    Q1, Q3 = quantiles[0], quantiles[1]

    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    df.filter((col(column) < lower_bound) | (col(column) > upper_bound)).coun
    capped_sales_data = df.withColumn(
        column,
        when(col(column) < lower_bound, lower_bound)\
        .when(col(column) > upper_bound, upper_bound)\
        .otherwise(col(column))
    )

    return capped_sales_data

capped_data = capp_outliers(cleaned_sales_data, "SALES")
capped_data = capp_outliers(capped_data, "MSRP")
capped_data = capp_outliers(capped_data, "QUANTITYORDERED")
capped_data = capp_outliers(capped_data, "PRICEEACH")

capped_data.show()
```

```
+----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+----------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|      ORDERDAT
E| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|        CUSTOM
ERNAME|           PHONE|        ADDRESSLINE1|ADDRESSLINE2|       CITY|   STAT
E|POSTALCODE|  COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
TOTAL_REVENUE|
+----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+----------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
|      10107|           30.0|     95.7|              2| 2871.0| 2/24/2003 0:0
0|Shipped|     1|       2|   2003|Motorcycles|95.0|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
```

```
Y|     10022|      USA|       NA|             Yu|            Kwai|   Small|
2871.0|
|      10121|           34.0|    81.35|              5| 2765.9|  5/7/2003 0:0
0|Shipped|     2|       5|   2003|Motorcycles|95.0|   S10_1678|  Reims Collec
tables|      26.47.1555|  59 rue de l'Abbaye|          NA|      Reims|      N
A|     51100|   France|     EMEA|        Henriot|            Paul|   Small|27
65.8999999999996|
|      10134|           41.0|    94.74|              2|3884.34|  7/1/2003 0:0
0|Shipped|     3|       7|   2003|Motorcycles|95.0|   S10_1678|     Lyon Souv
eniers|+33 1 46 62 7555|27 rue du Colonel...|          NA|      Paris|      N
A|     75508|   France|     EMEA|       Da Cunha|          Daniel|  Medium|38
84.3399999999997|
|      10145|           45.0|    83.26|              6| 3746.7| 8/25/2003 0:0
0|Shipped|     3|       8|   2003|Motorcycles|95.0|   S10_1678|   Toys4GrownU
ps.com|      6265557265|  78934 Hillside Dr.|          NA|   Pasadena|      C
A|     90003|      USA|       NA|          Young|           Julie|  Medium|37
46.7000000000003|
|      10168|           36.0|    96.66|              1|3479.76|10/28/2003 0:0
0|Shipped|     4|      10|   2003|Motorcycles|95.0|   S10_1678|Technics Store
s Inc.|      6505556809|   9408 Furth Circle|          NA| Burlingame|      C
A|     94217|      USA|       NA|         Hirano|            Juri|  Medium|34
79.7599999999998|
|      10180|           29.0|    86.13|              9|2497.77|11/11/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|95.0|   S10_1678|Daedalus Desig
ns ...|      20.16.1555|184, chausse de T...|          NA|      Lille|      N
A|     59000|   France|     EMEA|          Rance|         Martine|   Small|
2497.77|
|      10188|           48.0|    100.0|              1|5512.32|11/18/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|95.0|   S10_1678|        Herkku
Gifts|   +47 2267 3215|Drammen 121, PR 7...|          NA|     Bergen|      NA
|    N 5804|   Norway|     EMEA|         Oeztan|          Veysel|  Medium|
4800.0|
|      10211|           41.0|    100.0|             14|4708.44| 1/15/2004 0:0
0|Shipped|     1|       1|   2004|Motorcycles|95.0|   S10_1678|    Auto Canal
Petit|  (1) 47.55.6555|   25, rue Lauriston|          NA|      Paris|      NA
|     75016|   France|     EMEA|        Perrier|       Dominique|  Medium|
4100.0|
|      10223|           37.0|    100.0|              1|3965.66| 2/20/2004 0:0
0|Shipped|     1|       2|   2004|Motorcycles|95.0|   S10_1678|Australian Col
lec...|    03 9520 4555|   636 St Kilda Road|     Level 3|  Melbourne|Victori
a|      3004|Australia|     APAC|       Ferguson|           Peter|  Medium|
3700.0|
|      10237|           23.0|    100.0|              7|2333.12|  4/5/2004 0:0
0|Shipped|     2|       4|   2004|Motorcycles|95.0|   S10_1678|     Vitachrom
e Inc.|      2125551500|   2678 Kingston Rd.|   Suite 101|        NYC|      N
Y|     10022|      USA|       NA|          Frick|         Michael|   Small|
2300.0|
|      10251|           28.0|    100.0|              2|3188.64| 5/18/2004 0:0
0|Shipped|     2|       5|   2004|Motorcycles|95.0|   S10_1678|Tekni Collecta
ble...|      2015559350|       7476 Moss Rd.|          NA|     Newark|      N
J|     94019|      USA|       NA|          Brown|         William|  Medium|
2800.0|
|      10263|           34.0|    100.0|              2|3676.76| 6/28/2004 0:0
0|Shipped|     2|       6|   2004|Motorcycles|95.0|   S10_1678|     Gift Depo
t Inc.|      2035552570| 25593 South Bay Ln.|          NA|Bridgewater|      C
T|     97562|      USA|       NA|           King|           Julie|  Medium|
3400.0|
|      10275|           45.0|    92.83|              1|4177.35| 7/23/2004 0:0
0|Shipped|     3|       7|   2004|Motorcycles|95.0|   S10_1678|   La Rochelle
Gifts|      40.67.8555|67, rue des Cinqu...|          NA|     Nantes|      NA
|     44000|   France|     EMEA|        Labrune|          Janine|  Medium|
4177.35|
|      10285|           36.0|    100.0|              6|4099.68| 8/27/2004 0:0
0|Shipped|     3|       8|   2004|Motorcycles|95.0|   S10_1678|Marta's Replic
as Co.|      6175558555| 39323 Spinnaker Dr.|          NA|  Cambridge|      M
A|     51247|      USA|       NA|      Hernandez|           Marta|  Medium|
3600.0|
|      10299|           23.0|    100.0|              9|2597.39| 9/30/2004 0:0
0|Shipped|     3|       9|   2004|Motorcycles|95.0|   S10_1678|Toys of Finlan
```

```
d, Co.|     90-224 8555|       Keskuskatu 45|          NA|   Helsinki|      N
A|     21240|  Finland|     EMEA|      Karttunen|           Matti|   Small|
2300.0|
|      10309|           41.0|    100.0|              5|4394.38|10/15/2004 0:0
0|Shipped|     4|      10|   2004|Motorcycles|95.0|   S10_1678|  Baane Mini I
mports|      07-98 9555|Erling Skakkes ga...|          NA|    Stavern|      N
A|      4110|   Norway|     EMEA|     Bergulfsen|           Jonas|  Medium|
4100.0|
|      10318|           46.0|    94.74|              1|4358.04| 11/2/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|95.0|   S10_1678|Diecast Classi
cs ...|      2155551555|    7586 Pompton St.|          NA|  Allentown|      P
A|     70267|      USA|       NA|             Yu|           Kyung|  Medium|
4358.04|
|      10329|           42.0|    100.0|              1|4396.14|11/15/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|95.0|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|  Medium|
4200.0|
|      10341|           41.0|    100.0|              9|7211.16|11/24/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|95.0|   S10_1678|Salzburg Colle
cta...|       6562-9555|         Geislweg 14|          NA|   Salzburg|      N
A|      5020|  Austria|     EMEA|          Pipps|           Georg|   Large|
4100.0|
|      10361|           20.0|    72.55|             13| 1451.0|12/17/2004 0:0
0|Shipped|     4|      12|   2004|Motorcycles|95.0|   S10_1678|Souveniers And
Th...| +61 2 9495 8555|Monitor Money Bui...|     Level 6|  Chatswood|     NSW
|      2067|Australia|     APAC|         Huxley|          Adrian|   Small|
1451.0|
+----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+----------------+--------------------+-----------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+
only showing top 20 rows
```

r) How would you cache a DataFrame containing sales data from the top 10 countries by sales to avoid recomputation in subsequent transformations? What persistence level (e.g. MEMORY_ONLY, MEMORY_AND_DISK) would you choose and why?

In [35]:

```
from pyspark.storagelevel import StorageLevel

top_10_countries_df = cleaned_sales_data \
    .groupBy("COUNTRY").agg(sum("SALES").alias("TOTAL_SALES")) \
    .orderBy(col("TOTAL_SALES").desc()) \
    .limit(10) \
    .select(["COUNTRY", "TOTAL_SALES"])

cached_df = cleaned_sales_data.join(top_10_countries_df, on="COUNTRY", how="i
cached_df.persist(StorageLevel.MEMORY_ONLY)
cached_df.count()
```

Out[35]: 2345

- MEMORY ONLY Caches the data in memory(RAM) only
- MEMORY_AND_DISK caches data in memory and disk
- Since top 10 data is small memory only is sufficient

s) How would you pivot the data to show PRODUCTLINE as columns and the total SALES for each ORDERDATE as the

values? What are the implications of pivoting large datasets in Spark?

In [ ]:

In [36]:
```
from pyspark.sql.functions import avg, round

pivoted_df = cleaned_sales_data.groupBy("ORDERDATE").pivot("PRODUCTLINE").agg
pivoted_df = pivoted_df.fillna(0)
pivoted_df.show()
```

```
+--------------+------------+-----------+-------+-------+-------+------------
----+------------+
|     ORDERDATE|Classic Cars|Motorcycles| Planes|  Ships| Trains|Trucks and B
uses|Vintage Cars|
+--------------+------------+-----------+-------+-------+-------+------------
----+------------+
|3/29/2004 0:00|         0.0|        0.0|    0.0|2466.86|    0.0|
0.0|      3788.4|
|5/30/2005 0:00|         0.0|        0.0|    0.0|    0.0|    0.0|
0.0|     2082.68|
|3/19/2004 0:00|     7665.35|        0.0|    0.0|    0.0|    0.0|
0.0|         0.0|
| 9/7/2004 0:00|         0.0|        0.0|    0.0|    0.0|    0.0|
0.0|     3836.69|
| 5/4/2004 0:00|     5113.62|        0.0|    0.0|    0.0|    0.0|         378
7.66|     2694.15|
|11/9/2004 0:00|         0.0|        0.0|    0.0|3336.65|3807.68|
0.0|     3221.78|
|11/4/2003 0:00|     3441.18|    3146.19|    0.0|    0.0|    0.0|
0.0|         0.0|
| 7/1/2003 0:00|         0.0|     3660.7|    0.0|    0.0|    0.0|
0.0|         0.0|
|12/1/2003 0:00|         0.0|        0.0|3560.48|    0.0|    0.0|
0.0|      1113.6|
| 7/2/2004 0:00|     6167.41|        0.0|    0.0|    0.0|    0.0|
0.0|         0.0|
|1/29/2003 0:00|      5087.9|        0.0|    0.0|    0.0|    0.0|         329
1.59|     2732.87|
| 9/3/2003 0:00|     4444.54|    3155.58|    0.0|    0.0|    0.0|
0.0|         0.0|
|3/23/2005 0:00|     6109.29|        0.0|    0.0|4160.44| 2154.0|
0.0|     3902.65|
|10/4/2003 0:00|     3407.22|        0.0|    0.0|    0.0|    0.0|
0.0|         0.0|
|7/24/2003 0:00|     4892.34|        0.0|    0.0|    0.0|    0.0|         372
8.57|     1254.83|
| 3/3/2003 0:00|     3873.26|    2527.83|    0.0|    0.0|    0.0|
0.0|         0.0|
|1/23/2005 0:00|     3308.46|        0.0|    0.0|    0.0| 2986.5|         265
7.41|         0.0|
|4/26/2004 0:00|         0.0|        0.0|    0.0|    0.0|    0.0|
0.0|      3564.5|
|2/16/2005 0:00|      3059.4|        0.0|    0.0|    0.0|    0.0|
0.0|     2857.71|
|4/21/2003 0:00|         0.0|        0.0|    0.0| 4219.2|    0.0|
0.0|         0.0|
+--------------+------------+-----------+-------+-------+-------+------------
----+------------+
only showing top 20 rows
```

if the data is very large pivot data may take huge amount of memory and time due to shuffling of data. If the dataset is large and the data is not evenly distributed, it may cause incosistancies in data

t) How would you calculate the percentage growth of total sales month over month for each PRODUCTLINE using Spark DataFrame?

In [37]:
```
from pyspark.sql.functions import sum, lag, col, coalesce

monthly_sales = cleaned_sales_data.groupBy("PRODUCTLINE", "YEAR_ID", "MONTH_I
                                  .agg(sum("SALES").alias("TOTAL_SALES"))

window= Window.partitionBy("PRODUCTLINE").orderBy("YEAR_ID", "MONTH_ID")

monthly_sales_with_prev = monthly_sales.withColumn("PREV_TOTAL_SALES", lag("T

monthly_sales_with_growth = monthly_sales_with_prev.withColumn(
    "PERCENTAGE_GROWTH",
    (col("TOTAL_SALES") - coalesce(col("PREV_TOTAL_SALES"), col("TOTAL_SALES"
    coalesce(col("PREV_TOTAL_SALES"), col("TOTAL_SALES")) * 100
)

monthly_sales_with_growth.show()
```

```
+-----------+-------+--------+------------------+------------------+---------
----------+
|PRODUCTLINE|YEAR_ID|MONTH_ID|       TOTAL_SALES|  PREV_TOTAL_SALES|  PERCENT
AGE_GROWTH|
+-----------+-------+--------+------------------+------------------+---------
----------+
|Motorcycles|   2003|       2|25783.760000000002|              null|
0.0|
|Motorcycles|   2003|       3|          12639.15|25783.760000000002| -50.9801
9063162239|
|Motorcycles|   2003|       4|23475.590000000004|          12639.15|   85.737
0946622202|
|Motorcycles|   2003|       5|          22097.32|23475.590000000004|-5.871077
1486467594|
|Motorcycles|   2003|       6|           2642.01|          22097.32| -88.0437
5372217082|
|Motorcycles|   2003|       7| 37924.23000000001|           2642.01|  1335.43
0978686682|
|Motorcycles|   2003|       8|44164.909999999996| 37924.23000000001| 16.45565
3812878953|
|Motorcycles|   2003|       9|           3155.58|44164.909999999996| -92.8550
0638402749|
|Motorcycles|   2003|      10|          26791.38|           3155.58|  749.016
0287490733|
|Motorcycles|   2003|      11|          109345.5|          26791.38| 308.1368
7088906954|
|Motorcycles|   2004|       1|          41200.52|          109345.5| -62.3207
9052178646|
|Motorcycles|   2004|       2|           49066.5|          41200.52| 19.09194
3499742246|
|Motorcycles|   2004|       4| 36269.07000000001|           49066.5| -26.0818
0734309558|
|Motorcycles|   2004|       5|46848.950000000004| 36269.07000000001|  29.1705
3015144859|
|Motorcycles|   2004|       6|          47237.41|46848.950000000004| 0.829175
4671129217|
|Motorcycles|   2004|       7|           22774.0|          47237.41|-51.78821
1927791984|
|Motorcycles|   2004|       8|          62704.93|           22774.0|  175.335
6020022833|
|Motorcycles|   2004|       9| 42471.04999999999|          62704.93| -32.2684
0377622623|
|Motorcycles|   2004|      10|          39413.96| 42471.04999999999|-7.198056
0876173065|
```

```
|Motorcycles|   2004|      11|151711.85999999996|          39413.96|  284.919
0997301463|
+----------+-------+--------+------------------+------------------+--------
----------+
only showing top 20 rows
```

u) How can you rebalance the data by portioning based on the COUNTRY column to ensure that large data partitions are avoided?

In [38]:
```
cleaned_sales_data.rdd.getNumPartitions()
```

Out[38]: 1

In [39]:
```
cleaned_sales_data.select("COUNTRY").distinct().count()
```

Out[39]: 19

In [40]:
```
# Partition by each country
cleaned_sales_data.repartition(19,"COUNTRY").rdd.getNumPartitions()
```

Out[40]: 19

v) Suppose you have a smaller lookup table with customer details. How would you perform a broadcast join with the large sales_data_sample dataset to improve join performance? What are the key considerations when using broadcast joins?

In [41]:
```
from pyspark.sql.functions import broadcast

customer_details = spark.createDataFrame([
    (1, "Land of Toys Inc.", "landoftoys@example.com"),
    (2, "Reims Collectables", "reims.co@example.com")
], ["customer_id", "customer_name", "customer_email"])

broadcast_df = broadcast(customer_details)

cleaned_sales_data = cleaned_sales_data.cache()

joined_df = cleaned_sales_data.join(
    broadcast_df,
    cleaned_sales_data["CUSTOMERNAME"] == customer_details["customer_name"],
    "inner"
)

joined_df.select(["ORDERNUMBER", "CUSTOMERNAME"]).show()
```

```
+-----------+------------------+
|ORDERNUMBER|      CUSTOMERNAME|
+-----------+------------------+
|      10107| Land of Toys Inc.|
|      10121|Reims Collectables|
|      10329| Land of Toys Inc.|
|      10107| Land of Toys Inc.|
|      10329| Land of Toys Inc.|
|      10107| Land of Toys Inc.|
|      10329| Land of Toys Inc.|
```

```
|      10248| Land of Toys Inc.|
|      10359|Reims Collectables|
|      10329| Land of Toys Inc.|
|      10359|Reims Collectables|
|      10107| Land of Toys Inc.|
|      10121|Reims Collectables|
|      10329| Land of Toys Inc.|
|      10329| Land of Toys Inc.|
|      10359|Reims Collectables|
|      10329| Land of Toys Inc.|
|      10292| Land of Toys Inc.|
|      10329| Land of Toys Inc.|
|      10137|Reims Collectables|
+-----------+------------------+
only showing top 20 rows
```

w) Create a UDF that categorizes the sales values (SALES) into custom buckets like “Low”, “Medium”, “High”. Apply this UDF to the DataFrame and calculate the count of orders in each category per COUNTRY.

In [42]:
```
from pyspark.sql.functions import udf, col
from pyspark.sql.types import StringType

def categorize_sales(sales_amount):
    if sales_amount > percentile_67:
        return "High"
    elif sales_amount > percentile_33:
        return "Medium"
    else:
        return "Low"

categorize_sales_udf = udf(categorize_sales, StringType())
```

In [43]:
```
sales_data_with_category = cleaned_sales_data.withColumn(
    "SALES_CATEGORY", categorize_sales_udf(col("SALES"))
)

sales_data_with_category.select(['SALES','SALES_CATEGORY']).show()
```

```
+-------+--------------+
|  SALES|SALES_CATEGORY|
+-------+--------------+
| 2871.0|        Medium|
| 2765.9|        Medium|
|3884.34|        Medium|
| 3746.7|        Medium|
|3479.76|        Medium|
|2497.77|        Medium|
|5512.32|          High|
|4708.44|          High|
|3965.66|          High|
|2333.12|           Low|
|3188.64|        Medium|
|3676.76|        Medium|
|4177.35|          High|
|4099.68|          High|
|2597.39|        Medium|
|4394.38|          High|
|4358.04|          High|
|4396.14|          High|
|7737.93|          High|
| 1451.0|           Low|
```

```
+-------+-------------+
only showing top 20 rows
```

In [44]:

```
sales_counts_per_country = sales_data_with_category.groupBy("COUNTRY", "SALES
    .count() \
    .orderBy("COUNTRY", "SALES_CATEGORY")

sales_counts_per_country.show()
```

```
+---------+-------------+-----+
|  COUNTRY|SALES_CATEGORY|count|
+---------+-------------+-----+
|Australia|          High|   60|
|Australia|           Low|   66|
|Australia|        Medium|   59|
|  Austria|          High|   20|
|  Austria|           Low|   15|
|  Austria|        Medium|   20|
|  Belgium|          High|   11|
|  Belgium|           Low|   14|
|  Belgium|        Medium|    8|
|   Canada|          High|   18|
|   Canada|           Low|   28|
|   Canada|        Medium|   24|
|  Denmark|          High|   23|
|  Denmark|           Low|   18|
|  Denmark|        Medium|   22|
|  Finland|          High|   32|
|  Finland|           Low|   24|
|  Finland|        Medium|   36|
|   France|          High|  101|
|   France|           Low|  111|
+---------+-------------+-----+
only showing top 20 rows
```

x) Create a Python UDF to calculate discounts for specific product lines. For example, give a 10% discount for Classic Cars and 5% for Motorcycles. Apply this UDF to derive new discounted sales values.

In [45]:

```
def apply_discount(product_line, sales_amount):
    if product_line == "Classic Cars":
        return sales_amount * 0.90
    elif product_line == "Motorcycles":
        return sales_amount * 0.95
    else:
        return sales_amount

apply_discount_udf = udf(apply_discount, DoubleType())

sales_data_with_discount = cleaned_sales_data.withColumn(
    "DISCOUNTED_SALES", apply_discount_udf(col("PRODUCTLINE"), col("SALES"))
)


sales_data_with_discount.show()
```

```
+-----------+---------------+---------+---------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+---------------+--------------------+-----------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
----------------+------------------+
```

```
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|      ORDERDAT
E| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|PRODUCTCODE|        CUSTOM
ERNAME|           PHONE|        ADDRESSLINE1|ADDRESSLINE2|       CITY|   STAT
E|POSTALCODE|  COUNTRY|TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
TOTAL_REVENUE|  DISCOUNTED_SALES|
+-----------+---------------+---------+---------------+-------+-------------
-+-------+------+--------+-------+-----------+----+-----------+-------------
------+----------------+--------------------+------------+-----------+-------
-+----------+---------+---------+---------------+----------------+--------+--
---------------+-----------------+
|      10107|             30|     95.7|              2| 2871.0| 2/24/2003 0:0
0|Shipped|     1|       2|   2003|Motorcycles|  95|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|   Small|
2871.0|           2727.45|
|      10121|             34|    81.35|              5| 2765.9|  5/7/2003 0:0
0|Shipped|     2|       5|   2003|Motorcycles|  95|   S10_1678|  Reims Collec
tables|      26.47.1555|  59 rue de l'Abbaye|          NA|      Reims|      N
A|     51100|   France|     EMEA|        Henriot|            Paul|   Small|27
65.8999999999996|          2627.605|
|      10134|             41|    94.74|              2|3884.34|  7/1/2003 0:0
0|Shipped|     3|       7|   2003|Motorcycles|  95|   S10_1678|     Lyon Souv
eniers|+33 1 46 62 7555|27 rue du Colonel...|          NA|      Paris|      N
A|     75508|   France|     EMEA|       Da Cunha|          Daniel|  Medium|38
84.3399999999997|          3690.123|
|      10145|             45|    83.26|              6| 3746.7| 8/25/2003 0:0
0|Shipped|     3|       8|   2003|Motorcycles|  95|   S10_1678|   Toys4GrownU
ps.com|      6265557265|  78934 Hillside Dr.|          NA|   Pasadena|      C
A|     90003|      USA|       NA|          Young|           Julie|  Medium|37
46.7000000000003|          3559.365|
|      10168|             36|    96.66|              1|3479.76|10/28/2003 0:0
0|Shipped|     4|      10|   2003|Motorcycles|  95|   S10_1678|Technics Store
s Inc.|      6505556809|   9408 Furth Circle|          NA| Burlingame|      C
A|     94217|      USA|       NA|         Hirano|            Juri|  Medium|34
79.7599999999998|          3305.772|
|      10180|             29|    86.13|              9|2497.77|11/11/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|Daedalus Desig
ns ...|      20.16.1555|184, chausse de T...|          NA|      Lille|      N
A|     59000|   France|     EMEA|          Rance|         Martine|   Small|
2497.77|         2372.8815|
|      10188|             48|    100.0|              1|5512.32|11/18/2003 0:0
0|Shipped|     4|      11|   2003|Motorcycles|  95|   S10_1678|        Herkku
Gifts|   +47 2267 3215|Drammen 121, PR 7...|          NA|     Bergen|      NA
|    N 5804|   Norway|     EMEA|         Oeztan|          Veysel|  Medium|
4800.0|          5236.704|
|      10211|             41|    100.0|             14|4708.44| 1/15/2004 0:0
0|Shipped|     1|       1|   2004|Motorcycles|  95|   S10_1678|    Auto Canal
Petit|  (1) 47.55.6555|   25, rue Lauriston|          NA|      Paris|      NA
|     75016|   France|     EMEA|        Perrier|       Dominique|  Medium|
4100.0| 4473.017999999999|
|      10223|             37|    100.0|              1|3965.66| 2/20/2004 0:0
0|Shipped|     1|       2|   2004|Motorcycles|  95|   S10_1678|Australian Col
lec...|    03 9520 4555|   636 St Kilda Road|     Level 3|  Melbourne|Victori
a|      3004|Australia|     APAC|       Ferguson|           Peter|  Medium|
3700.0|3767.3769999999995|
|      10237|             23|    100.0|              7|2333.12|  4/5/2004 0:0
0|Shipped|     2|       4|   2004|Motorcycles|  95|   S10_1678|     Vitachrom
e Inc.|      2125551500|   2678 Kingston Rd.|   Suite 101|        NYC|      N
Y|     10022|      USA|       NA|          Frick|         Michael|   Small|
2300.0|          2216.464|
|      10251|             28|    100.0|              2|3188.64| 5/18/2004 0:0
0|Shipped|     2|       5|   2004|Motorcycles|  95|   S10_1678|Tekni Collecta
ble...|      2015559350|       7476 Moss Rd.|          NA|     Newark|      N
J|     94019|      USA|       NA|          Brown|         William|  Medium|
2800.0|3029.2079999999996|
|      10263|             34|    100.0|              2|3676.76| 6/28/2004 0:0
0|Shipped|     2|       6|   2004|Motorcycles|  95|   S10_1678|     Gift Depo
t Inc.|      2035552570| 25593 South Bay Ln.|          NA|Bridgewater|      C
T|     97562|      USA|       NA|           King|           Julie|  Medium|
```

```
3400.0|          3492.922|
|      10275|             45|    92.83|              1|4177.35| 7/23/2004 0:0
0|Shipped|     3|       7|   2004|Motorcycles|  95|   S10_1678|   La Rochelle
Gifts|      40.67.8555|67, rue des Cinqu...|          NA|     Nantes|      NA
|     44000|   France|     EMEA|        Labrune|          Janine|  Medium|
4177.35|         3968.4825|
|      10285|             36|    100.0|              6|4099.68| 8/27/2004 0:0
0|Shipped|     3|       8|   2004|Motorcycles|  95|   S10_1678|Marta's Replic
as Co.|      6175558555| 39323 Spinnaker Dr.|          NA|  Cambridge|      M
A|     51247|      USA|       NA|      Hernandez|           Marta|  Medium|
3600.0|          3894.696|
|      10299|             23|    100.0|              9|2597.39| 9/30/2004 0:0
0|Shipped|     3|       9|   2004|Motorcycles|  95|   S10_1678|Toys of Finlan
d, Co.|     90-224 8555|       Keskuskatu 45|          NA|   Helsinki|      N
A|     21240|  Finland|     EMEA|      Karttunen|           Matti|   Small|
2300.0|2467.5204999999996|
|      10309|             41|    100.0|              5|4394.38|10/15/2004 0:0
0|Shipped|     4|      10|   2004|Motorcycles|  95|   S10_1678|  Baane Mini I
mports|      07-98 9555|Erling Skakkes ga...|          NA|    Stavern|      N
A|      4110|   Norway|     EMEA|     Bergulfsen|           Jonas|  Medium|
4100.0|          4174.661|
|      10318|             46|    94.74|              1|4358.04| 11/2/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Diecast Classi
cs ...|      2155551555|    7586 Pompton St.|          NA|  Allentown|      P
A|     70267|      USA|       NA|             Yu|           Kyung|  Medium|
4358.04|          4140.138|
|      10329|             42|    100.0|              1|4396.14|11/15/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|   Land of Toy
s Inc.|      2125557818|897 Long Airport ...|          NA|        NYC|      N
Y|     10022|      USA|       NA|             Yu|            Kwai|  Medium|
4200.0|4176.3330000000005|
|      10341|             41|    100.0|              9|7737.93|11/24/2004 0:0
0|Shipped|     4|      11|   2004|Motorcycles|  95|   S10_1678|Salzburg Colle
cta...|       6562-9555|         Geislweg 14|          NA|   Salzburg|      N
A|      5020|  Austria|     EMEA|          Pipps|           Georg|   Large|
4100.0|          7351.0335|
|      10361|             20|    72.55|             13| 1451.0|12/17/2004 0:0
0|Shipped|     4|      12|   2004|Motorcycles|  95|   S10_1678|Souveniers And
Th...| +61 2 9495 8555|Monitor Money Bui...|     Level 6|  Chatswood|     NSW
|      2067|Australia|     APAC|         Huxley|          Adrian|   Small|
1451.0|           1378.45|
+----------+--------------+---------+--------------+-------+--------------
-+-------+------+--------+-------+-----------+----+-----------+--------------
------+----------------+--------------------+------------+----------+-------
-+----------+---------+---------+--------------+----------------+--------+--
----------------+-----------------+
only showing top 20 rows
```

y) How would you set up an incremental loading mechanism for orders placed daily based on the ORDERDATE column? How can Spark checkpointing can be used with incremental load to ensure no data loss occurs during failures?

In [46]:

```
spark.sparkContext.setCheckpointDir('file:///home/hadoop/checkpoint')

try:
    print(last_processed_date)
except:
    last_processed_date = "2003-01-04 00:00:00"

new_sales = cleaned_sales_data.filter(col("ORDERDATE") > last_processed_date)

new_sales_checkpointed = new_sales.checkpoint()

new_sales_checkpointed.write.format('csv').mode('append').save('file:///home/
```

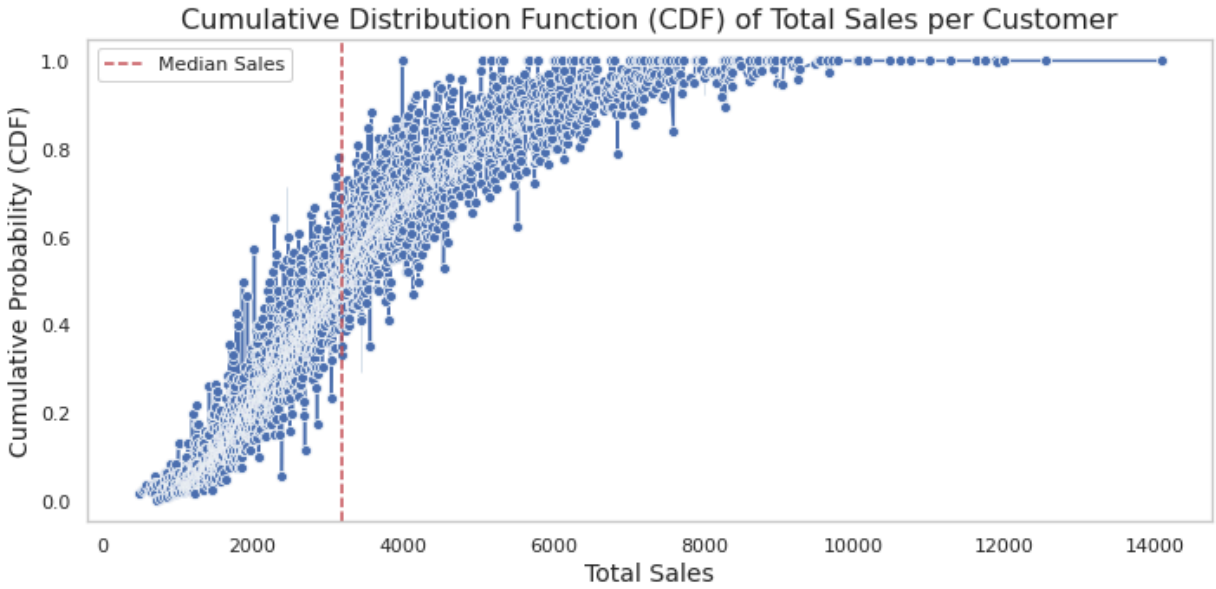
```
last_processed_date = new_sales.agg(max('ORDERDATE'))
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-46-f12049803f15> in <module>
     12 new_sales_checkpointed.write.format('csv').mode('append').save('fil
e:///home/hadoop/increment')
     13
---> 14 last_processed_date = new_sales.agg(max('ORDERDATE'))

/usr/local/spark/python/pyspark/sql/dataframe.py in agg(self, *exprs)
   1448         [Row(min(age)=2)]
   1449         """
-> 1450         return self.groupBy().agg(*exprs)
   1451
   1452     @since(2.0)

/usr/local/spark/python/pyspark/sql/group.py in agg(self, *exprs)
    111         else:
    112             # Columns
--> 113             assert all(isinstance(c, Column) for c in exprs), "all ex
prs should be Column"
    114             jdf = self._jgd.agg(exprs[0]._jc,
    115                                 _to_seq(self.sql_ctx._sc, [c._jc for
c in exprs[1:]]))

AssertionError: all exprs should be Column
```

In [ ]:
```
spark.sparkContext.setCheckpointDir('file:///home/hadoop/checkpoint')
```

z) How do you implement a cumulative distribution function (CDF) over the SALES value for each CUSTOMERNAME? What insights can you gather from analyzing the CDF distribution for each customer?

In [48]:
```
cdf_pandas = cdf_df.select("SALES", "CDF").toPandas()

sns.set(style="whitegrid")


plt.figure(figsize=(10, 5))

sns.lineplot(data=cdf_pandas, y="CDF", x="SALES", marker='o')

median_value = np.median(cdf_pandas['SALES'])
plt.axvline(median_value, color='r', linestyle='--', label='Median Sales')

plt.title("Cumulative Distribution Function (CDF) of Total Sales per Customer
plt.xlabel("Total Sales", fontsize=14)
plt.ylabel("Cumulative Probability (CDF)", fontsize=14)
plt.legend()
plt.grid()
plt.tight_layout()

plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: