In [2]:
```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *

spark = SparkSession\
    .builder\
    .appName("chapter-07-aggregation")\
    .getOrCreate()

import os
SPARK_BOOK_DATA_PATH = os.environ['SPARK_BOOK_DATA_PATH']
```

In [3]:
```
file_path = SPARK_BOOK_DATA_PATH + "/data/retail-data/all/*.csv"

df = spark.read.format("csv")\
  .option("header", "true")\
  .option("inferSchema", "true")\
  .load(file_path)\
  .coalesce(3)

df.show(5,False)
```

```
+---------+---------+-----------------------------------+--------+----
----------+---------+----------+--------------+
|InvoiceNo|StockCode|Description                        |Quantity|Invo
iceDate   |UnitPrice|CustomerID|Country       |
+---------+---------+-----------------------------------+--------+----
----------+---------+----------+--------------+
|536365   |85123A   |WHITE HANGING HEART T-LIGHT HOLDER |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 HEARTS COAT HANGER     |8       |12/
1/2010 8:26|2.75     |17850     |United Kingdom|
|536365   |84029G   |KNITTED UNION FLAG HOT WATER BOTTLE|6       |12/
1/2010 8:26|3.39     |17850     |United Kingdom|
|536365   |84029E   |RED WOOLLY HOTTIE WHITE HEART.     |6       |12/
1/2010 8:26|3.39     |17850     |United Kingdom|
+---------+---------+-----------------------------------+--------+----
----------+---------+----------+--------------+
only showing top 5 rows
```

In [4]:
```
df.cache()
```

Out[4]:
```
DataFrame[InvoiceNo: string, StockCode: string, Description: string, Q
uantity: int, InvoiceDate: string, UnitPrice: double, CustomerID: int,
Country: string]
```

In [5]:
```
df.createOrReplaceTempView("dfTable")
```

In [6]:
```
# COMMAND ----------

from pyspark.sql.functions import count
df.select(count("StockCode")).show() # 541909
```

```
+----------------+
|count(StockCode)|
+----------------+
|          541909|
+----------------+
```

In [7]:
```
# COMMAND ----------

from pyspark.sql.functions import countDistinct,expr,desc
```

In [8]:
```
df.select(countDistinct("StockCode")).show() # 4070
```

```
+-------------------------+
|count(DISTINCT StockCode)|
+-------------------------+
|                     4070|
+-------------------------+
```

In [9]:
```
df.groupBy("StockCode").agg(expr("count(StockCode)")).show(10, False)
```

```
+---------+----------------+
|StockCode|count(StockCode)|
+---------+----------------+
|22728    |810             |
|21889    |607             |
|90210B   |7               |
|21259    |296             |
|21894    |135             |
|21452    |200             |
|22121    |141             |
|90022    |21              |
|21249    |119             |
|90143    |22              |
+---------+----------------+
only showing top 10 rows
```

In [10]:
```
df.where("StockCode in ('90026D', '90210B') ").orderBy("StockCode", des
```

```
+---------+---------+--------------------------------+--------+------
----------+---------+----------+--------------+
|InvoiceNo|StockCode|Description                     |Quantity|Invoic
eDate     |UnitPrice|CustomerID|Country       |
+---------+---------+--------------------------------+--------+------
----------+---------+----------+--------------+
|548545   |90026D   |GLASS BEAD HOOP NECKLACE AMETHYST|1       |3/31/2
011 19:12 |8.5      |13118     |United Kingdom|
|544463   |90026D   |GLASS BEAD HOOP NECKLACE AMETHYST|1       |2/20/2
011 14:31 |8.5      |12988     |United Kingdom|
|577315   |90026D   |GLASS BEAD HOOP NECKLACE AMETHYST|1       |11/18/
2011 13:25|8.5      |17811     |United Kingdom|
|568787   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |6       |9/29/2
011 9:19  |2.95     |13741     |United Kingdom|
|538071   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |1       |12/9/2
010 14:09 |2.96     |null      |United Kingdom|
|581434   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |10      |12/8/2
011 16:10 |1.0      |13599     |United Kingdom|
|537045   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |1       |12/5/2
010 10:54 |2.95     |15038     |United Kingdom|
|538661   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |12      |12/13/
2010 15:42|1.25     |15194     |United Kingdom|
|536409   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |1       |12/1/2
010 11:45 |2.95     |17908     |United Kingdom|
|573102   |90210B   |CLEAR ACRYLIC FACETED BANGLE     |10      |10/27/
2011 14:52|1.0      |13266     |United Kingdom|
+---------+---------+--------------------------------+--------+------
----------+---------+----------+--------------+
```

In [11]:
```
# COMMAND ----------

from pyspark.sql.functions import approx_count_distinct
df.select(approx_count_distinct("StockCode", 0.1)).show() # 3364
```

```
+--------------------------------+
|approx_count_distinct(StockCode)|
+--------------------------------+
|                            3364|
+--------------------------------+
```

In [12]:
```
# COMMAND ----------

from pyspark.sql.functions import first, last
df.select(first("StockCode"), last("StockCode")).show()
```

```
+-----------------------+----------------------+
|first(StockCode, false)|last(StockCode, false)|
+-----------------------+----------------------+
|                 85123A|                 22138|
+-----------------------+----------------------+
```

In [13]:
```
# COMMAND ----------

from pyspark.sql.functions import min, max
df.select(min("Quantity"), max("Quantity")).show()
```

```
+-------------+-------------+
|min(Quantity)|max(Quantity)|
+-------------+-------------+
|       -80995|        80995|
+-------------+-------------+
```

In [14]:
```
# COMMAND ----------

from pyspark.sql.functions import sum
df.select(sum("Quantity")).show() # 5176450
```

```
+-------------+
|sum(Quantity)|
+-------------+
|      5176450|
+-------------+
```

In [15]:
```
# COMMAND ----------

from pyspark.sql.functions import sumDistinct
df.select(sumDistinct("Quantity")).show() # 29310
```

```
+----------------------+
|sum(DISTINCT Quantity)|
+----------------------+
|                 29310|
+----------------------+
```

In [16]:
```
# COMMAND ----------

from pyspark.sql.functions import sum, count, avg, expr

df.select(
    count("Quantity").alias("total_transactions"),
    sum("Quantity").alias("total_purchases"),
    avg("Quantity").alias("avg_purchases"),
    expr("mean(Quantity)").alias("mean_purchases"))\
  .selectExpr(
    "total_purchases/total_transactions",
    "avg_purchases",
    "mean_purchases").show()
```

```
+-----------------------------------+---------------+-------------
---+
|(total_purchases / total_transactions)|   avg_purchases|  mean_purcha
ses|
+-----------------------------------+---------------+-------------
---+
|                     9.55224954743324|9.55224954743324|9.55224954743
324|
+-----------------------------------+---------------+-------------
---+
```

In [17]:
```
# COMMAND ----------

from pyspark.sql.functions import var_pop, stddev_pop
from pyspark.sql.functions import var_samp, stddev_samp
df.select(var_pop("Quantity"),
          var_samp("Quantity"),
          stddev_pop("Quantity"),
          stddev_samp("Quantity")).show()
```

```
+-----------------+-----------------+-------------------+-----------
----------+
|var_pop(Quantity)|var_samp(Quantity)|stddev_pop(Quantity)|stddev_samp
(Quantity)|
+-----------------+-----------------+-------------------+-----------
----------+
|47559.30364660928| 47559.39140929898|  218.08095663447847|    218.081
1578502347|
+-----------------+-----------------+-------------------+-----------
----------+
```

In [18]:
```
# COMMAND ----------

from pyspark.sql.functions import skewness, kurtosis
df.select(skewness("Quantity"), kurtosis("Quantity")).show()
```

```
+--------------------+------------------+
|  skewness(Quantity)|kurtosis(Quantity)|
+--------------------+------------------+
|-0.26407557610528154|  119768.054955306|
+--------------------+------------------+
```

In [19]:
```
# COMMAND ----------

from pyspark.sql.functions import corr, covar_pop, covar_samp
df.select(
        corr("InvoiceNo", "Quantity"),
        covar_samp("InvoiceNo", "Quantity"),
        covar_pop("InvoiceNo", "Quantity")).show()
```

```
+-------------------------+------------------------------+-----------
------------------+
|corr(InvoiceNo, Quantity)|covar_samp(InvoiceNo, Quantity)|covar_pop(I
nvoiceNo, Quantity)|
+-------------------------+------------------------------+-----------
------------------+
|     4.912186085639875E-4|             1052.7280543916152|
1052.726077875511|
+-------------------------+------------------------------+-----------
------------------+
```

In [22]:
```
# COMMAND ----------

from pyspark.sql.functions import collect_set, collect_list
df.agg(collect_set("Country"), collect_list("Country")).show(10, False)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [23]:
```
# COMMAND ----------

from pyspark.sql.functions import count

df.groupBy("InvoiceNo").agg(
    count("Quantity").alias("quan"),
    expr("count(Quantity)")).show()
```

```
+---------+----+---------------+
|InvoiceNo|quan|count(Quantity)|
+---------+----+---------------+
|   536596|   6|              6|
|   536938|  14|             14|
|   537252|   1|              1|
|   537691|  20|             20|
|   538041|   1|              1|
|   538184|  26|             26|
|   538517|  53|             53|
|   538879|  19|             19|
|   539275|   6|              6|
|   539630|  12|             12|
|   540499|  24|             24|
|   540540|  22|             22|
|  C540850|   1|              1|
|   540976|  48|             48|
|   541432|   4|              4|
|   541518| 101|            101|
|   541783|  35|             35|
|   542026|   9|              9|
|   542375|   6|              6|
|  C542604|   8|              8|
+---------+----+---------------+
only showing top 20 rows
```

In [24]:
```
# COMMAND ----------

df.groupBy("InvoiceNo").agg(
        expr("avg(Quantity)"),
        expr("stddev_pop(Quantity)")).show()
```

```
+---------+------------------+--------------------+
|InvoiceNo|     avg(Quantity)|stddev_pop(Quantity)|
+---------+------------------+--------------------+
|   536596|               1.5|  1.1180339887498947|
|   536938|33.142857142857146|  20.698023172885524|
|   537252|              31.0|                 0.0|
|   537691|              8.15|   5.597097462078001|
|   538041|              30.0|                 0.0|
|   538184|12.076923076923077|   8.142590198943392|
|   538517|3.0377358490566038|  2.3946659604837897|
|   538879|21.157894736842106|  11.811070444356483|
|   539275|              26.0|  12.806248474865697|
|   539630|20.333333333333332|  10.225241100118645|
|   540499|              3.75|  2.6653642652865788|
|   540540|2.1363636363636362|  1.0572457590557278|
|  C540850|              -1.0|                 0.0|
|   540976|10.520833333333334|   6.496760677872902|
|   541432|             12.25|  10.825317547305483|
|   541518| 23.10891089108911|  20.550782784878713|
|   541783|11.314285714285715|   8.467657556242811|
|   542026| 7.666666666666667|   4.853406592853679|
|   542375|               8.0|  3.4641016151377544|
|  C542604|              -8.0|  15.173990905493518|
+---------+------------------+--------------------+
only showing top 20 rows
```

In [25]:
```
# COMMAND ----------

from pyspark.sql.functions import col, to_date
dfWithDate = df.withColumn("date", to_date(col("InvoiceDate"), "MM/dd/y
dfWithDate.createOrReplaceTempView("dfWithDate")
```

In [26]:
```
dfWithDate.show(10)
```

```
+---------+---------+--------------------+--------+--------------+----
-----+----------+--------------+----------+
|InvoiceNo|StockCode|         Description|Quantity|   InvoiceDate|Unit
Price|CustomerID|       Country|      date|
+---------+---------+--------------------+--------+--------------+----
-----+----------+--------------+----------+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/2010 8:26|
2.55|     17850|United Kingdom|2010-12-01|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/2010 8:26|
3.39|     17850|United Kingdom|2010-12-01|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/2010 8:26|
2.75|     17850|United Kingdom|2010-12-01|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/2010 8:26|
3.39|     17850|United Kingdom|2010-12-01|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/2010 8:26|
3.39|     17850|United Kingdom|2010-12-01|
|   536365|    22752|SET 7 BABUSHKA NE...|       2|12/1/2010 8:26|
7.65|     17850|United Kingdom|2010-12-01|
|   536365|    21730|GLASS STAR FROSTE...|       6|12/1/2010 8:26|
4.25|     17850|United Kingdom|2010-12-01|
|   536366|    22633|HAND WARMER UNION...|       6|12/1/2010 8:28|
1.85|     17850|United Kingdom|2010-12-01|
|   536366|    22632|HAND WARMER RED P...|       6|12/1/2010 8:28|
1.85|     17850|United Kingdom|2010-12-01|
|   536367|    84879|ASSORTED COLOUR B...|      32|12/1/2010 8:34|
1.69|     13047|United Kingdom|2010-12-01|
+---------+---------+--------------------+--------+--------------+----
-----+----------+--------------+----------+
only showing top 10 rows
```

In [27]:
```
# COMMAND ----------

from pyspark.sql.window import Window
from pyspark.sql.functions import desc
windowSpec = Window\
  .partitionBy("CustomerId", "date")\
  .orderBy(desc("Quantity"))\
  .rowsBetween(Window.unboundedPreceding, Window.currentRow)
```

In [28]:
```
# COMMAND ----------

from pyspark.sql.functions import max
maxPurchaseQuantity = max(col("Quantity")).over(windowSpec)
```

In [29]:
```
# COMMAND ----------

from pyspark.sql.functions import dense_rank, rank
purchaseDenseRank = dense_rank().over(windowSpec)
purchaseRank = rank().over(windowSpec)
```

In [30]:
```
# COMMAND ----------

from pyspark.sql.functions import col

dfWithDate.where("CustomerId IS NOT NULL").orderBy("CustomerId")\
  .select(
    col("CustomerId"),
    col("date"),
    col("Quantity"),
    purchaseRank.alias("quantityRank"),
    purchaseDenseRank.alias("quantityDenseRank"),
    maxPurchaseQuantity.alias("maxPurchaseQuantity")).show()
```

```
+----------+----------+--------+------------+----------------+-------
------------+
|CustomerId|      date|Quantity|quantityRank|quantityDenseRank|maxPurc
haseQuantity|
+----------+----------+--------+------------+----------------+-------
------------+
|     12346|2011-01-18|   74215|           1|               1|
74215|
|     12346|2011-01-18|  -74215|           2|               2|
74215|
|     12347|2010-12-07|      36|           1|               1|
36|
|     12347|2010-12-07|      30|           2|               2|
36|
|     12347|2010-12-07|      24|           3|               3|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|      12|           4|               4|
36|
|     12347|2010-12-07|       6|          17|               5|
```

```
36|
|     12347|2010-12-07|       6|          17|                 5|
36|
+----------+----------+--------+------------+------------------+-------
------------+
only showing top 20 rows
```

In [31]:
```
# COMMAND ----------

dfNoNull = dfWithDate.drop()
dfNoNull.createOrReplaceTempView("dfNoNull")
```

In [32]:
```
# COMMAND ----------

rolledUpDF = dfNoNull.rollup("Date", "Country").agg(sum("Quantity"))\
  .selectExpr("Date", "Country", "`sum(Quantity)` as total_quantity")\
  .orderBy("Date")
rolledUpDF.show()
```

```
+----------+--------------+--------------+
|      Date|       Country|total_quantity|
+----------+--------------+--------------+
|      null|          null|       5176450|
|2010-12-01|   Netherlands|            97|
|2010-12-01|          EIRE|           243|
|2010-12-01|     Australia|           107|
|2010-12-01|        France|           449|
|2010-12-01|       Germany|           117|
|2010-12-01|          null|         26814|
|2010-12-01|United Kingdom|         23949|
|2010-12-01|        Norway|          1852|
|2010-12-02|          null|         21023|
|2010-12-02|       Germany|           146|
|2010-12-02|          EIRE|             4|
|2010-12-02|United Kingdom|         20873|
|2010-12-03|        France|           239|
|2010-12-03|         Italy|           164|
|2010-12-03|       Germany|           170|
|2010-12-03|   Switzerland|           110|
|2010-12-03|         Spain|           400|
|2010-12-03|        Poland|           140|
|2010-12-03|          null|         14830|
+----------+--------------+--------------+
only showing top 20 rows
```

In [33]:
```
# COMMAND ----------

from pyspark.sql.functions import sum

dfNoNull.cube("Date", "Country")\
    .agg(sum(col("Quantity")), count(col("Quantity")))\
    .select("Date", "Country", "sum(Quantity)", "count(Quantity)")\
    .orderBy("Date").show()
```

```
+----+--------------------+-------------+---------------+
|Date|             Country|sum(Quantity)|count(Quantity)|
+----+--------------------+-------------+---------------+
|null|               Japan|        25218|            358|
|null|           Australia|        83653|           1259|
|null|            Portugal|        16180|           1519|
|null|                null|      5176450|         541909|
|null|                 RSA|          352|             58|
|null|             Finland|        10666|            695|
|null|United Arab Emirates|          982|             68|
|null|           Singapore|         5234|            229|
|null|         Unspecified|         3300|            446|
|null|             Germany|       117448|           9495|
|null|     Channel Islands|         9479|            758|
|null|                 USA|         1034|            291|
|null|           Hong Kong|         4769|            288|
|null|             Denmark|         8188|            389|
|null|      Czech Republic|          592|             30|
|null|  European Community|          497|             61|
|null|             Lebanon|          386|             45|
|null|               Spain|        26824|           2533|
|null|              Cyprus|         6317|            622|
|null|              Norway|        19247|           1086|
+----+--------------------+-------------+---------------+
only showing top 20 rows
```

In [34]:
```
# COMMAND ----------

pivoted = dfWithDate.groupBy("date").pivot("Country").sum()


# COMMAND ----------
```

In [41]:
```
# pivoted.show()
```

In [ ]:
```

```