In [5]:
```
from IPython.display import display, clear_output
```

In [1]:
```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *

spark = SparkSession\
    .builder\
    .appName("chapter-03-tour")\
    .getOrCreate()

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

Spark SQL

In [2]:
```
file_path = SPARK_BOOK_DATA_PATH + "/data/retail-data/by-day/*.csv"

retail_df = spark.read.csv(file_path, header=True, inferSchema=True)
```

```
retail_df = spark.read\
  .format("csv")\
  .option("header", "true")\
  .option("inferSchema", "true")\
  .load(file_path)
```

In [3]:
```
retail_df.count()
```

Out[3]: 541909

In [4]:
```
1 retail_df.show(5,False)
```

```
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
|InvoiceNo|StockCode|Description                   |Quantity|InvoiceD
ate        |UnitPrice|CustomerID|Country       |
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
|580538   |23084    |RABBIT NIGHT LIGHT            |48      |2011-12-
05 08:38:00|1.79     |14075.0   |United Kingdom|
|580538   |23077    |DOUGHNUT LIP GLOSS            |20      |2011-12-
05 08:38:00|1.25     |14075.0   |United Kingdom|
|580538   |22906    |12 MESSAGE CARDS WITH ENVELOPES|24      |2011-12-
05 08:38:00|1.65     |14075.0   |United Kingdom|
|580538   |21914    |BLUE HARMONICA IN BOX         |24      |2011-12-
05 08:38:00|1.25     |14075.0   |United Kingdom|
|580538   |22467    |GUMBALL COAT RACK             |6       |2011-12-
05 08:38:00|2.55     |14075.0   |United Kingdom|
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
only showing top 5 rows
```

In [6]:
```
1 display(retail_df.toPandas())
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cou
0	580538	23084	RABBIT NIGHT LIGHT	48	2011-12-05 08:38:00	1.79	14075.0	Ur King
1	580538	23077	DOUGHNUT LIP GLOSS	20	2011-12-05 08:38:00	1.25	14075.0	Ur King
2	580538	22906	12 MESSAGE CARDS WITH ENVELOPES	24	2011-12-05 08:38:00	1.65	14075.0	Ur King
3	580538	21914	BLUE HARMONICA IN BOX	24	2011-12-05 08:38:00	1.25	14075.0	Ur King
4	580538	22467	GUMBALL COAT RACK	6	2011-12-05 08:38:00	2.55	14075.0	Ur King
...	...	...	...	...	...	...	...	
541904	543282	22849	BREAD BIN DINER STYLE MINT	1	2011-02-06 16:08:00	16.95	12956.0	Ur King
541905	543282	84879	ASSORTED COLOUR BIRD ORNAMENT	8	2011-02-06 16:08:00	1.69	12956.0	Ur King
541906	543282	84659A	WHITE TRAVEL ALARM CLOCK	1	2011-02-06 16:08:00	2.55	12956.0	Ur King
541907	543282	82484	WOOD BLACK BOARD ANT WHITE FINISH	1	2011-02-06 16:08:00	7.95	12956.0	Ur King
541908	543282	22168	ORGANISER WOOD ANTIQUE WHITE	1	2011-02-06 16:08:00	8.50	12956.0	Ur King

541909 rows × 8 columns

In [7]:
```
1 retail_df.createOrReplaceTempView("retail_table")
```

In [8]:
```
1 staticSchema = retail_df.schema
```

In [9]:
```
1 print(staticSchema)
```

```
StructType(List(StructField(InvoiceNo,StringType,true),StructField(Sto
ckCode,StringType,true),StructField(Description,StringType,true),Struc
tField(Quantity,IntegerType,true),StructField(InvoiceDate,StringType,t
rue),StructField(UnitPrice,DoubleType,true),StructField(CustomerID,Dou
bleType,true),StructField(Country,StringType,true)))
```

In [10]:
```
1 retail_df.printSchema()
```

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

In [11]:
```
1 retail_df.describe().show()
```

```
+-------+-----------------+-----------------+--------------------+--
---------------+------------------+-----------------+---------------
--+-----------+
|summary|         InvoiceNo|        StockCode|         Description|
Quantity|       InvoiceDate|        UnitPrice|        CustomerID|
Country|
+-------+-----------------+-----------------+--------------------+--
---------------+------------------+-----------------+---------------
--+-----------+
|  count|           541909|           541909|              540455|
541909|            541909|           541909|            406829|     5
41909|
|   mean|  559965.752026781|27623.240210938104|             20713.0|
9.55224954743324|              null|4.611113626089641|15287.690570239
585|       null|
| stddev|13428.417280796697|16799.737628427683|                 NaN|21
8.0811578502335|              null|96.75985306117963| 1713.6003033215
97|       null|
|    min|           536365|            10002| 4 PURPLE FLOCK D...|
-80995|2010-12-01 08:26:00|        -11062.06|           12346.0|  Aust
ralia|
|    max|          C581569|                m|   wrongly sold sets|
80995|2011-12-09 12:50:00|          38970.0|           18287.0|Unspeci
fied|
+-------+-----------------+-----------------+--------------------+--
---------------+------------------+-----------------+---------------
--+-----------+
```

In [12]:
```
1 df = spark.sql("select * from retail_table limit 5")
```

In [13]:
```
df.show(truncate=False)   # disable truncate to show description in
```

```
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
|InvoiceNo|StockCode|Description                   |Quantity|InvoiceD
ate        |UnitPrice|CustomerID|Country       |
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
|580538   |23084    |RABBIT NIGHT LIGHT            |48      |2011-12-
05 08:38:00|1.79     |14075.0   |United Kingdom|
|580538   |23077    |DOUGHNUT LIP GLOSS            |20      |2011-12-
05 08:38:00|1.25     |14075.0   |United Kingdom|
|580538   |22906    |12 MESSAGE CARDS WITH ENVELOPES|24      |2011-12-
05 08:38:00|1.65     |14075.0   |United Kingdom|
|580538   |21914    |BLUE HARMONICA IN BOX         |24      |2011-12-
05 08:38:00|1.25     |14075.0   |United Kingdom|
|580538   |22467    |GUMBALL COAT RACK             |6       |2011-12-
05 08:38:00|2.55     |14075.0   |United Kingdom|
+---------+---------+------------------------------+--------+--------
-----------+---------+----------+--------------+
```

In [15]:

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

from pyspark.sql.functions import window, column, desc, col

(retail_df.selectExpr(
    "CustomerId",
    "(UnitPrice * Quantity) as total_cost",
    "InvoiceDate")
  .groupBy(col("CustomerId"), window(col("InvoiceDate"), "1 day"))
  .sum("total_cost")
  .sort(desc("sum(total_cost)"))
  .show(5, False)
)
```

```
+----------+------------------------------------------+-----------------
---+
|CustomerId|window                                    |sum(total_cost)
|
+----------+------------------------------------------+-----------------
---+
|17450.0   |[2011-09-19 20:00:00, 2011-09-20 20:00:00]|71601.44
|
|null      |[2011-11-13 19:00:00, 2011-11-14 19:00:00]|55316.08
|
|null      |[2011-11-06 19:00:00, 2011-11-07 19:00:00]|42939.17
|
|null      |[2011-03-28 20:00:00, 2011-03-29 20:00:00]|33521.399999999
98 |
|null      |[2011-12-07 19:00:00, 2011-12-08 19:00:00]|31975.590000000
007|
+----------+------------------------------------------+-----------------
---+
only showing top 5 rows
```

In [16]:
```
# col() can be omitted
retail_df.selectExpr(
    "CustomerId",
    "(UnitPrice * Quantity) as total_cost",
    "InvoiceDate")\
  .groupBy("CustomerId", window("InvoiceDate", "1 day"))\
  .sum("total_cost")\
  .sort(desc("sum(total_cost)"))\
  .withColumnRenamed("sum(total_cost)", "sum_total_cost")\
  .withColumnRenamed("window", "InvoiceDateWindow")\
  .show(5, truncate=False)
```

```
+----------+-----------------------------------------+--------------
---+
|CustomerId|InvoiceDateWindow                        |sum_total_cost
|
+----------+-----------------------------------------+--------------
---+
|17450.0   |[2011-09-19 20:00:00, 2011-09-20 20:00:00]|71601.44
|
|null      |[2011-11-13 19:00:00, 2011-11-14 19:00:00]|55316.08
|
|null      |[2011-11-06 19:00:00, 2011-11-07 19:00:00]|42939.17
|
|null      |[2011-03-28 20:00:00, 2011-03-29 20:00:00]|33521.399999999
98 |
|null      |[2011-12-07 19:00:00, 2011-12-08 19:00:00]|31975.590000000
007|
+----------+-----------------------------------------+--------------
---+
only showing top 5 rows
```

Spark Streaming

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

streamingDataFrame = (
    spark
    .readStream
    .format("csv")
    .schema(staticSchema)
    .option("maxFilesPerTrigger", 1)
    .option("header", "true")
    .load(SPARK_BOOK_DATA_PATH + "/data/retail-data/by-day/*.csv")
)
```

In [19]:
```
# Transform
purchaseByCustomerPerHour = (streamingDataFrame
  .selectExpr(
    "CustomerId",
    "(UnitPrice * Quantity) as total_cost",
    "InvoiceDate")
  .groupBy(col("CustomerId"), window(col("InvoiceDate"), "1 day"))
  .sum("total_cost")
)
```

In [20]:
```
(
purchaseByCustomerPerHour
    .writeStream.queryName("customer_purchases")
    .format("memory")
    .outputMode("complete")
    .start()
)
```

Out[20]:
```
<pyspark.sql.streaming.StreamingQuery at 0x7fd992895dc0>
```

a SQL table `customer_purchases` is created by the `queryName`

use `Ctrl-Enter` to execute below cell repeatly to see streaming result as more data are read

In [21]:
```
spark.sql("""
  SELECT *
  FROM customer_purchases
  ORDER BY `sum(total_cost)` DESC
  """).show(5, False)
```

```
+----------+------+--------------+
|CustomerId|window|sum(total_cost)|
+----------+------+--------------+
+----------+------+--------------+
```

In [22]:

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

spark.sql("""
  SELECT *
  FROM customer_purchases
  ORDER BY `sum(total_cost)` DESC
  """)\
  .show(5,truncate=False)
```

```
+----------+-----------------------------------------+-----------------
---+
|CustomerId|window                                   |sum(total_cost)
|
+----------+-----------------------------------------+-----------------
---+
|12678.0   |[2011-10-27 20:00:00, 2011-10-28 20:00:00]|8947.960000000000
05 |
|13694.0   |[2011-10-27 20:00:00, 2011-10-28 20:00:00]|3304.030000000000
01 |
|null      |[2011-10-27 20:00:00, 2011-10-28 20:00:00]|3270.980000000000
03 |
|13199.0   |[2011-10-27 20:00:00, 2011-10-28 20:00:00]|1912.799999999999
997|
|15290.0   |[2011-10-27 20:00:00, 2011-10-28 20:00:00]|1510.360000000000
001|
+----------+-----------------------------------------+-----------------
---+
only showing top 5 rows
```

In [24]:

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

spark.sql("""
  SELECT *
  FROM customer_purchases
  ORDER BY `sum(total_cost)` DESC
  """)\
  .show(5,truncate=False)
```

```
+----------+-----------------------------------------+-----------------
---+
|CustomerId|window                                   |sum(total_cost)
|
+----------+-----------------------------------------+-----------------
---+
|null      |[2011-11-06 19:00:00, 2011-11-07 19:00:00]|42939.17
|
|null      |[2011-07-03 20:00:00, 2011-07-04 20:00:00]|13667.659999999
93 |
|18102.0   |[2011-07-03 20:00:00, 2011-07-04 20:00:00]|13282.0
|
|null      |[2010-11-30 19:00:00, 2010-12-01 19:00:00]|12584.299999999
988|
|null      |[2011-09-06 20:00:00, 2011-09-07 20:00:00]|12446.109999999
957|
+----------+-----------------------------------------+-----------------
---+
only showing top 5 rows
```

Spark ML Pipeline

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

from pyspark.sql.functions import date_format, col

preppedDataFrame = retail_df\
  .na.fill(0)\
  .withColumn("day_of_week", date_format(col("InvoiceDate"), "EEEE")
  .coalesce(5)

preppedDataFrame.show(3, truncate=False)
```

```
+---------+---------+-----------------------------+--------+--------
----------+---------+----------+--------------+-----------+
|InvoiceNo|StockCode|Description                  |Quantity|InvoiceD
ate        |UnitPrice|CustomerID|Country       |day_of_week|
+---------+---------+-----------------------------+--------+--------
----------+---------+----------+--------------+-----------+
|580538   |23084    |RABBIT NIGHT LIGHT           |48      |2011-12-
05 08:38:00|1.79     |14075.0   |United Kingdom|Monday     |
|580538   |23077    |DOUGHNUT LIP GLOSS           |20      |2011-12-
05 08:38:00|1.25     |14075.0   |United Kingdom|Monday     |
|580538   |22906    |12 MESSAGE CARDS WITH ENVELOPES|24      |2011-12-
05 08:38:00|1.65     |14075.0   |United Kingdom|Monday     |
+---------+---------+-----------------------------+--------+--------
----------+---------+----------+--------------+-----------+
only showing top 3 rows
```

In [26]:
```
# COMMAND ----------

trainDataFrame = preppedDataFrame\
  .where("InvoiceDate < '2011-07-01'")

testDataFrame = preppedDataFrame\
  .where("InvoiceDate >= '2011-07-01'")
```

In [27]:
```
trainDataFrame.show(3)
```

```
+---------+---------+--------------------+--------+------------------
+---------+----------+--------------+-----------+
|InvoiceNo|StockCode|         Description|Quantity|        InvoiceDate
|UnitPrice|CustomerID|       Country|day_of_week|
+---------+---------+--------------------+--------+------------------
+---------+----------+--------------+-----------+
|   537226|    22811|SET OF 6 T-LIGHTS...|       6|2010-12-06 08:34:00
|     2.95|   15987.0|United Kingdom|     Monday|
|   537226|    21713|CITRONELLA CANDLE...|       8|2010-12-06 08:34:00
|      2.1|   15987.0|United Kingdom|     Monday|
|   537226|    22927|GREEN GIANT GARDE...|       2|2010-12-06 08:34:00
|     5.95|   15987.0|United Kingdom|     Monday|
+---------+---------+--------------------+--------+------------------
+---------+----------+--------------+-----------+
only showing top 3 rows
```

In [28]:
```
testDataFrame.show(3)
```

```
+---------+---------+--------------------+--------+--------------------
+---------+----------+--------------+-----------+
|InvoiceNo|StockCode|         Description|Quantity|        InvoiceDate
|UnitPrice|CustomerID|       Country|day_of_week|
+---------+---------+--------------------+--------+--------------------
+---------+----------+--------------+-----------+
|   580538|    23084|  RABBIT NIGHT LIGHT|      48|2011-12-05 08:38:00
|     1.79|   14075.0|United Kingdom|     Monday|
|   580538|    23077| DOUGHNUT LIP GLOSS |      20|2011-12-05 08:38:00
|     1.25|   14075.0|United Kingdom|     Monday|
|   580538|    22906|12 MESSAGE CARDS ...|      24|2011-12-05 08:38:00
|     1.65|   14075.0|United Kingdom|     Monday|
+---------+---------+--------------------+--------+--------------------
+---------+----------+--------------+-----------+
only showing top 3 rows
```

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

from pyspark.ml.feature import StringIndexer

indexer = StringIndexer()\
  .setInputCol("day_of_week")\
  .setOutputCol("day_of_week_index")
```

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

from pyspark.ml.feature import OneHotEncoder

encoder = OneHotEncoder()\
  .setInputCol("day_of_week_index")\
  .setOutputCol("day_of_week_encoded")
```

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

from pyspark.ml.feature import VectorAssembler

vectorAssembler = VectorAssembler()\
  .setInputCols(["UnitPrice", "Quantity", "day_of_week_encoded"])\
  .setOutputCol("features")
```

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

from pyspark.ml import Pipeline

transformationPipeline = Pipeline()\
  .setStages([indexer, encoder, vectorAssembler])
```

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

fittedPipeline = transformationPipeline.fit(trainDataFrame)
```

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

transformedTraining = fittedPipeline.transform(trainDataFrame)
```

In [35]:
```
transformedTraining.show(5, truncate=False)
```

```
--------+-------------------+---------------------------+
|537226   |22811    |SET OF 6 T-LIGHTS CACTI       |6       |2010-12-
06 08:34:00|2.95     |15987.0   |United Kingdom|Monday     |2.0
|(5,[2],[1.0])      |(7,[0,1,4],[2.95,6.0,1.0]) |
|537226   |21713    |CITRONELLA CANDLE FLOWERPOT   |8       |2010-12-
06 08:34:00|2.1      |15987.0   |United Kingdom|Monday     |2.0
|(5,[2],[1.0])      |(7,[0,1,4],[2.1,8.0,1.0])  |
|537226   |22927    |GREEN GIANT GARDEN THERMOMETER|2       |2010-12-
06 08:34:00|5.95     |15987.0   |United Kingdom|Monday     |2.0
|(5,[2],[1.0])      |(7,[0,1,4],[5.95,2.0,1.0]) |
|537226   |20802    |SMALL GLASS SUNDAE DISH CLEAR |6       |2010-12-
06 08:34:00|1.65     |15987.0   |United Kingdom|Monday     |2.0
|(5,[2],[1.0])      |(7,[0,1,4],[1.65,6.0,1.0]) |
|537226   |22052    |VINTAGE CARAVAN GIFT WRAP     |25      |2010-12-
06 08:34:00|0.42     |15987.0   |United Kingdom|Monday     |2.0
|(5,[2],[1.0])      |(7,[0,1,4],[0.42,25.0,1.0])|
+---------+---------+------------------------------+--------+--------
----------+---------+----------+--------------+----------+---------
--------+-------------------+---------------------------+
only showing top 5 rows
```

Spark ML Clustering

In [46]:
```
# COMMAND ----------

from pyspark.ml.clustering import KMeans

kmeans = KMeans()\
  .setK(20)\
  .setSeed(10)
```

In [47]:
```
# COMMAND ----------

kmModel = kmeans.fit(transformedTraining)
```

In [48]:
```
type(kmModel)
```

Out[48]:
```
pyspark.ml.clustering.KMeansModel
```

In [51]:
```
kmModel.summary
```

Out[51]:
```
<pyspark.ml.clustering.KMeansSummary at 0x7f551744e1f0>
```

In [ ]:

```
1
```