25/25

In [1]:
```
#first imort pyspark and buld an object to access pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Sample Data Processing").getOrCreate()
```

In [2]:
```
#create a dataframe df to read the movies.csv file and store the data
MoviesDF=spark.read.csv("F:\\movies.csv",inferSchema=True,header=True)
```

In [3]:
```
#verify if df is showing the data it loaded correct
MoviesDF.show(5)
```

```
+-----------------+-------------+----+
|            actor|        title|year|
+-----------------+-------------+----+
|McClure, Marc (I)|Freaky Friday|2003|
|McClure, Marc (I)| Coach Carter|2005|
|McClure, Marc (I)|  Superman II|1980|
|McClure, Marc (I)|    Apollo 13|1995|
|McClure, Marc (I)|     Superman|1978|
+-----------------+-------------+----+
only showing top 5 rows
```

In [4]:
```
#1.How many movies are included in movies.csv?
print("The number of movies included in movies.csv are:",MoviesDF.count())
```

```
The number of movies included in movies.csv are: 31394
```

In [5]:
```
#2.     What features (or attributes) are recorded for each movie?
print("The features recorded for each movie are:",MoviesDF.columns)
```

```
The features recorded for each movie are: ['actor', 'title', 'year']
```

In [6]:
```
#3.     What is the shape of your data?
print("Spark does not support shape function like Pandas,shape can be printed
 like below")
print(MoviesDF.count(),len(MoviesDF.columns))
```

```
Spark does not support shape function like Pandas,shape can be printed like b
elow
31394 3
```

In [7]:
```
#4.     Provide a schema of the movies data set.
print("The schema of the movies data set is as below:")
print("--------------------------------")
MoviesDF.printSchema()
```

```
The schema of the movies data set is as below:
--------------------------------
root
 |-- actor: string (nullable = true)
 |-- title: string (nullable = true)
 |-- year: integer (nullable = true)
```

In [8]:
```
#5.     Provide a listing of the first 5 movies. For each movie, display the m
ovie name and year it was produced, in that order.
MoviesDF.select('title','year').show(5)
```

```
+-------------+----+
|        title|year|
+-------------+----+
|Freaky Friday|2003|
| Coach Carter|2005|
|  Superman II|1980|
|    Apollo 13|1995|
|     Superman|1978|
+-------------+----+
only showing top 5 rows
```

In [9]:
```
#6.     Provide a count of all movies produced before the year 2000.
print('The count of all movies produced before the year 2000 is:')
MoviesDF.filter(MoviesDF['year']<2000).count()
```

```
The count of all movies produced before the year 2000 is:
```

Out[9]:
```
8400
```

In [10]:
```
#7.     List the names of all actors who have acted in the movie Contagion. Li
st only the names of actor.
print("The names of all actors who have acted in the movie Contagion are:")
ListDF=MoviesDF.dropDuplicates()
ActorsDF=ListDF.filter(ListDF['title']=='Contagion')
ActorsDF.select('actor').show(50)
```

```
The names of all actors who have acted in the movie Contagion are:
+--------------------+
|               actor|
+--------------------+
|        Lavell, Mark|
| Spence, Rebecca (I)|
|      Wiggins, Roger|
|Edwards, Shannon (I)|
|        Ortlieb, Jim|
|    Paltrow, Gwyneth|
|Chamberlain, Cabr...|
|Panzarella, Russ (V)|
|      Ehle, Jennifer|
| Weston II, James D.|
|       Lathan, Sanaa|
|         Damon, Matt|
|     Meadows, Samuel|
|      Stern, January|
|  Mitchell, E. Roger|
|     Armour, Annabel|
|    Thomas, Chris D.|
|Curnen, Monique G...|
|       Thurner, John|
|Kanellakos, Alexa...|
|   Colantoni, Enrico|
|       Winslet, Kate|
|       Clarke, Larry|
| Price, Steven James|
|Cohen, David (XXV...|
|Smith, Gregory Ma...|
| Bartlett, Tommy (I)|
|          Kress, Don|
|           Law, Jude|
|      Nazimek, Larry|
|        Marino, Carl|
|  Stewart, Thomas W.|
|   Cotillard, Marion|
|    Young, Robert A.|
| Fishburne, Laurence|
|     Cranston, Bryan|
|      Gould, Elliott|
|    Hawkes, John (I)|
+--------------------+
```

truncate = False

to prevent actor name from being truncated

In [11]:
```
#8.     List all movies in which John Travolta has acted. List the movie name
 and year produced.
print("The list of movies in which John Travolta has acted are:")
MovieYearDF=MoviesDF.filter(MoviesDF['actor']=='Travolta, John')
MovieYearDF.select('title','year').show()
```

```
The list of movies in which John Travolta has acted are:
+--------------------+----+
|               title|year|
+--------------------+----+
|          Phenomenon|1996|
|           Wild Hogs|2007|
|             Michael|1996|
|Saturday Night Fever|1977|
|           Hairspray|2007|
|              Grease|1978|
|      Primary Colors|1998|
|        Pulp Fiction|1994|
|   The Thin Red Line|1998|
|           Swordfish|2001|
|      A Civil Action|1998|
|Magnificent Desol...|2005|
|  Look Who's Talking|1989|
|Austin Powers in ...|2002|
|Domestic Disturbance|2001|
|The General's Dau...|1999|
|            Face/Off|1997|
|                Bolt|2008|
|           Ladder 49|2004|
|The Taking of Pel...|2009|
+--------------------+----+
```

truncate : False

In [12]:
```
#9.     Provide a count of movies each actor has acted in (group by actor's na
me). The list should display actor's
#name and count of movies. The list should be alphabetized by actor's name.
MoviesDF.groupBy('actor').count().orderBy('actor',ascending=True).show()
```

```
+------------------+-----+
|             actor|count|
+------------------+-----+
|   Aaron, Caroline|    6|
|    Aarons, Bonnie|    5|
|   Abadie, William|    3|
|   Abbott, Deborah|    4|
|       Abdoo, Rose|    5|
|    Abdullah, Haji|    5|
|   Abell, Alistair|    3|
|  Abercrombie, Ian|    5|
|  Abergel, Rakefet|    3|
|    Abernathy, Don|   18|
|   Aboutboul, Alon|    3|
|Abraham, F. Murray|    4|
|    Abrahams, Doug|    3|
| Abrahams, Jon (I)|    3|
|      Abrell, Brad|    4|
|    Abustan, Jason|    3|
|     Acheson, Mark|    4|
|     Ackland, Joss|    4|
|      Acovone, Jay|    8|
|   Acres, Isabella|    4|
+------------------+-----+
only showing top 20 rows
```

In [13]:
```
#10.    Create a standard pyspark UDF to designate any movie produced before 2
000 as Old while  movies produced in 2000
#or later are designated as New. Then, apply the UDF to the movies data so a n
ew column named ‘age_category’ is added permanently.
#Display the first 10 movie records (Movie name, year produced, age_category).
from pyspark.sql.functions import udf #user defined function
from pyspark.sql.types import StringType,DoubleType
year_udf=udf(lambda year:'New' if year>=2000 else 'Old', StringType())
YearCategoryDF=MoviesDF.withColumn('age_category',year_udf(MoviesDF.year))
YearCategoryDF.select('title','year','age_category').show(10)
```

```
+--------------------+----+------------+
|               title|year|age_category|
+--------------------+----+------------+
|       Freaky Friday|2003|         New|
|        Coach Carter|2005|         New|
|         Superman II|1980|         Old|
|           Apollo 13|1995|         Old|
|            Superman|1978|         Old|
|  Back to the Future|1985|         Old|
|Back to the Futur...|1990|         Old|
|  Me, Myself & Irene|2000|         New|
|         October Sky|1999|         Old|
|              Capote|2005|         New|
+--------------------+----+------------+
only showing top 10 rows
```

In [ ]: