

Six Degrees of Kevin Bacon

Introduction - Six Degrees of Kevin Bacon is a game based on the "six degrees of separation" concept, which posits that any two people on Earth are six or fewer acquaintance links apart. Movie buffs challenge each other to find the shortest path between an arbitrary actor and prolific actor Kevin Bacon. It rests on the assumption that anyone involved in the film industry can be linked through their film roles to Bacon within six steps. The analysis of social networks can be a computationally intensive task, especially when dealing with large volumes of data. It is also a challenging problem to devise a correct methodology to infer an informative social network structure. Here, we will analyze a social network of actors and actresses that co-participated in movies. We will do some simple descriptive analysis, and in the end try to relate an actor/actress's position in the social network with the success of the movies in which they participate.

Rules & Notes - Please take your time to read the following points:

1. The submission deadline will be set for the 30th of May at 23:59h.
2. It is acceptable that you **discuss** with your colleagues different approaches to solve each step of the problem set, but the assignment is individual. That is, you are responsible for writing your own code, and analyzing the results. Clear cases of cheating will be penalized with 0 points in this assignment.
3. After review of your submission files, and before a mark is attributed, you might be called to orally defend your submission.
4. You will be scored first and foremost by the number of correct answers, secondly by the logic used in trying to approach each step of the problem set.
5. You can add as many cells as you like to answer the questions.
6. It is also important you clearly indicate what your final answer to each question is when you are using multiple cells (for example you can use `print("My final answer is:")` before your answer or use cell comments).
7. Consider skipping questions that you are stuck on, and get back to them later.
8. Expect computations to take a few minutes to finish in some of the steps.
9. It is recommended you read the whole assignment before starting.
10. You can make use of caching or persisting your RDDs or Dataframes, this may speed up performance. You do not need to cache every dataframe, but usually you want to do this at least once after the data has been imported.
11. If you have trouble with graphframes in databricks (specifically the import statement) you need to make sure the graphframes package is installed on the cluster you are running. If you click home on the left, then click on the graphframes library which you loaded in Lab 11 you can install the package on your cluster (check the graphframes checkbox and click install)
12. Be careful, you must not 'Publish' this notebook in databricks.
13. **IMPORTANT** It is expected you have developed skills beyond writing SQL queries. Any question where you directly write a SQL query (by for example creating a temporary view and then using `spark.sql` to pass the query) will receive a 25% penalty. Using the spark syntax (for example `dataframe.select(">").where("conditions")`) is acceptable and does not incur this penalty.
14. **Questions** – Any questions about this assignment should be posted in the Forum@Moodle. Questions by e-mail will not be answered. The lab will run at the normal time. During this period you can ask any questions you have about the exam (we can't provide you the actual answers of course, but there may be helpful tips if you are stuck on any of the steps). As such, it is probably useful to attempt the assignment before the scheduled lab.
15. **Delivery** - To fulfil this activity you will have to upload the following materials to Moodle:
 - An exported IPython notebook. From the menu at the top, select 'File', then 'Export', then 'IPython Notebook', to download the notebook. The notebook should be solved (have results displayed), but should contain all necessary code so that when the notebook is run in databricks it should also

replicate these results. This means that all data downloading and processing should be done in this notebook. It is also important you clearly indicate where your final answer to each question is when you are using multiple cells (for example you can use `print("My final answer is:")` before your answer or use cell comments).

- A PDF version of your code and answers. There are a couple of ways you can do this. You can convert the downloaded IPython Notebook to pdf (check out nbconvert if you have Jupyter notebook), or you can just copy your code and answers into a word file and save as pdf, or finally you can take screenshots of each page of the notebook and put them into a word file and save it as pdf. It is important that all code and answers are visible in this pdf.
- You will also need to provide a signed statement of authorship, which is available on Moodle.

Data Sources and Description

We will use data from IMDB. You can download raw datafiles from <https://datasets.imdbws.com> (<https://datasets.imdbws.com>). Note that the files are tab delimited (.tsv) You can find a description of the each datafile in <https://www.imdb.com/interfaces/> (<https://www.imdb.com/interfaces/>)

Questions

Data loading and preparation

Review the file descriptions and load the necessary data onto your databricks cluster and into spark dataframes or rdds. You will need to use shell commands to download the data, unzip the data, load the data into spark. Note that the data might require parsing and preprocessing to be ready for the questions below.

Hints You can use `gunzip` to unzip the .tz files. The data files will then be tab separated (.tsv), which you can load into a dataframe using the tab separated option instead of the comma separated option we have typically used in class: `.option("sep", "\t")`

In [3]:

```
%sh wget https://datasets.imdbws.com/name.basics.tsv.gz  
  
%sh  
gunzip name.basics.tsv.gz
```

```
--2020-05-30 13:04:40-- https://datasets.imdbws.com/name.basics.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.26, 13.224.13.3
2, 13.224.13.37, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.26|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 197587198 (188M) [binary/octet-stream]
Saving to: 'name.basics.tsv.gz.1'
```

0K	0%	3.95M	48s
50K	0%	7.85M	36s
100K	0%	16.1M	28s
150K	0%	12.8M	25s
200K	0%	18.1M	22s
250K	0%	9.01M	22s
300K	0%	130M	19s
350K	0%	142M	17s
400K	0%	82.8M	15s
450K	0%	31.6M	14s
500K	0%	8.81M	15s
550K	0%	71.3M	14s
600K	0%	77.8M	13s
650K	0%	61.7M	12s
700K	0%	95.3M	11s
750K	0%	78.0M	11s
800K	0%	64.2M	10s
850K	0%	103M	10s
900K	0%	74.3M	10s
950K	0%	92.7M	9s
1000K	0%	104M	9s
1050K	0%	16.5M	9s
1100K	0%	112M	9s
1150K	0%	74.2M	8s
1200K	0%	73.5M	8s
1250K	0%	88.3M	8s
1300K	0%	119M	8s
1350K	0%	77.2M	7s
1400K	0%	76.4M	7s
1450K	0%	125M	7s
1500K	0%	131M	7s
1550K	0%	93.4M	7s
1600K	0%	155M	7s
1650K	0%	128M	6s
1700K	0%	159M	6s
1750K	0%	126M	6s
1800K	0%	181M	6s
1850K	0%	110M	6s
1900K	1%	119M	6s
1950K	1%	130M	6s
2000K	1%	127M	6s
2050K	1%	142M	5s
2100K	1%	148M	5s
2150K	1%	162M	5s
2200K	1%	133M	5s

2250K	1%	140M	5s
2300K	1%	143M	5s
2350K	1%	144M	5s
2400K	1%	133M	5s
2450K	1%	162M	5s
2500K	1%	139M	5s
2550K	1%	147M	5s
2600K	1%	155M	5s
2650K	1%	146M	5s
2700K	1%	140M	4s
2750K	1%	104M	4s
2800K	1%	123M	4s
2850K	1%	144M	4s
2900K	1%	138M	4s
2950K	1%	115M	4s
3000K	1%	123M	4s
3050K	1%	146M	4s
3100K	1%	147M	4s
3150K	1%	140M	4s
3200K	1%	104M	4s
3250K	1%	151M	4s
3300K	1%	185M	4s
3350K	1%	84.9M	4s
3400K	1%	144M	4s
3450K	1%	155M	4s
3500K	1%	133M	4s
3550K	1%	125M	4s
3600K	1%	87.1M	4s
3650K	1%	140M	4s
3700K	1%	160M	4s
3750K	1%	164M	4s
3800K	1%	145M	4s
3850K	2%	125M	4s
3900K	2%	108M	4s
3950K	2%	54.6M	4s
4000K	2%	65.0M	4s
4050K	2%	97.7M	3s
4100K	2%	147M	3s
4150K	2%	124M	3s
4200K	2%	133M	3s
4250K	2%	116M	3s
4300K	2%	113M	3s
4350K	2%	145M	3s
4400K	2%	133M	3s
4450K	2%	202M	3s
4500K	2%	130M	3s
4550K	2%	107M	3s
4600K	2%	117M	3s
4650K	2%	112M	3s
4700K	2%	108M	3s
4750K	2%	146M	3s
4800K	2%	97.6M	3s
4850K	2%	158M	3s
4900K	2%	136M	3s
4950K	2%	199M	3s

5000K	2%	168M	3s
5050K	2%	186M	3s
5100K	2%	110M	3s
5150K	2%	123M	3s
5200K	2%	88.3M	3s
5250K	2%	151M	3s
5300K	2%	135M	3s
5350K	2%	161M	3s
5400K	2%	109M	3s
5450K	2%	93.4M	3s
5500K	2%	99.8M	3s
5550K	2%	80.6M	3s
5600K	2%	124M	3s
5650K	2%	112M	3s
5700K	2%	168M	3s
5750K	3%	141M	3s
5800K	3%	96.8M	3s
5850K	3%	137M	3s
5900K	3%	151M	3s
5950K	3%	137M	3s
6000K	3%	139M	3s
6050K	3%	142M	3s
6100K	3%	121M	3s
6150K	3%	141M	3s
6200K	3%	138M	3s
6250K	3%	138M	3s
6300K	3%	167M	3s
6350K	3%	110M	3s
6400K	3%	122M	3s
6450K	3%	136M	3s
6500K	3%	105M	3s
6550K	3%	135M	3s
6600K	3%	165M	3s
6650K	3%	121M	3s
6700K	3%	147M	3s
6750K	3%	134M	3s
6800K	3%	127M	3s
6850K	3%	104M	3s
6900K	3%	165M	3s
6950K	3%	156M	3s
7000K	3%	146M	3s
7050K	3%	147M	3s
7100K	3%	149M	3s
7150K	3%	96.2M	3s
7200K	3%	139M	3s
7250K	3%	182M	3s
7300K	3%	148M	3s
7350K	3%	158M	3s
7400K	3%	126M	3s
7450K	3%	146M	3s
7500K	3%	181M	2s
7550K	3%	115M	2s
7600K	3%	118M	2s
7650K	3%	149M	2s
7700K	4%	83.8M	2s

7750K	4%	143M	2s
7800K	4%	119M	2s
7850K	4%	144M	2s
7900K	4%	139M	2s
7950K	4%	134M	2s
8000K	4%	35.6M	2s
8050K	4%	151M	2s
8100K	4%	122M	2s
8150K	4%	139M	2s
8200K	4%	106M	2s
8250K	4%	144M	2s
8300K	4%	81.6M	2s
8350K	4%	124M	2s
8400K	4%	141M	2s
8450K	4%	164M	2s
8500K	4%	147M	2s
8550K	4%	137M	2s
8600K	4%	120M	2s
8650K	4%	98.0M	2s
8700K	4%	134M	2s
8750K	4%	142M	2s
8800K	4%	166M	2s
8850K	4%	129M	2s
8900K	4%	148M	2s
8950K	4%	133M	2s
9000K	4%	117M	2s
9050K	4%	144M	2s
9100K	4%	135M	2s
9150K	4%	123M	2s
9200K	4%	134M	2s
9250K	4%	146M	2s
9300K	4%	148M	2s
9350K	4%	107M	2s
9400K	4%	139M	2s
9450K	4%	185M	2s
9500K	4%	174M	2s
9550K	4%	167M	2s
9600K	5%	159M	2s
9650K	5%	182M	2s
9700K	5%	170M	2s
9750K	5%	161M	2s
9800K	5%	165M	2s
9850K	5%	180M	2s
9900K	5%	174M	2s
9950K	5%	158M	2s
10000K	5%	162M	2s
10050K	5%	187M	2s
10100K	5%	180M	2s
10150K	5%	197M	2s
10200K	5%	160M	2s
10250K	5%	177M	2s
10300K	5%	172M	2s
10350K	5%	158M	2s
10400K	5%	162M	2s
10450K	5%	175M	2s

10500K	5%	170M	2s
10550K	5%	167M	2s
10600K	5%	161M	2s
10650K	5%	188M	2s
10700K	5%	184M	2s
10750K	5%	162M	2s
10800K	5%	166M	2s
10850K	5%	188M	2s
10900K	5%	200M	2s
10950K	5%	196M	2s
11000K	5%	185M	2s
11050K	5%	180M	2s
11100K	5%	192M	2s
11150K	5%	178M	2s
11200K	5%	168M	2s
11250K	5%	209M	2s
11300K	5%	200M	2s
11350K	5%	199M	2s
11400K	5%	194M	2s
11450K	5%	205M	2s
11500K	5%	199M	2s
11550K	6%	175M	2s
11600K	6%	186M	2s
11650K	6%	211M	2s
11700K	6%	191M	2s
11750K	6%	199M	2s
11800K	6%	191M	2s
11850K	6%	204M	2s
11900K	6%	182M	2s
11950K	6%	170M	2s
12000K	6%	166M	2s
12050K	6%	206M	2s
12100K	6%	207M	2s
12150K	6%	209M	2s
12200K	6%	175M	2s
12250K	6%	189M	2s
12300K	6%	195M	2s
12350K	6%	166M	2s
12400K	6%	166M	2s
12450K	6%	187M	2s
12500K	6%	191M	2s
12550K	6%	198M	2s
12600K	6%	185M	2s
12650K	6%	188M	2s
12700K	6%	172M	2s
12750K	6%	166M	2s
12800K	6%	176M	2s
12850K	6%	187M	2s
12900K	6%	201M	2s
12950K	6%	201M	2s
13000K	6%	173M	2s
13050K	6%	189M	2s
13100K	6%	179M	2s
13150K	6%	179M	2s
13200K	6%	169M	2s

13250K	6%	205M	2s
13300K	6%	211M	2s
13350K	6%	175M	2s
13400K	6%	169M	2s
13450K	6%	211M	2s
13500K	7%	192M	2s
13550K	7%	181M	2s
13600K	7%	146M	2s
13650K	7%	207M	2s
13700K	7%	204M	2s
13750K	7%	198M	2s
13800K	7%	169M	2s
13850K	7%	199M	2s
13900K	7%	177M	2s
13950K	7%	171M	2s
14000K	7%	167M	2s
14050K	7%	192M	2s
14100K	7%	196M	2s
14150K	7%	203M	2s
14200K	7%	153M	2s
14250K	7%	188M	2s
14300K	7%	210M	2s
14350K	7%	144M	2s
14400K	7%	183M	2s
14450K	7%	208M	2s
14500K	7%	176M	2s
14550K	7%	205M	2s
14600K	7%	132M	2s
14650K	7%	219M	2s
14700K	7%	224M	2s
14750K	7%	180M	2s
14800K	7%	185M	2s
14850K	7%	189M	2s
14900K	7%	204M	2s
14950K	7%	208M	2s
15000K	7%	162M	2s
15050K	7%	185M	2s
15100K	7%	191M	2s
15150K	7%	177M	2s
15200K	7%	183M	2s
15250K	7%	209M	2s
15300K	7%	160M	2s
15350K	7%	197M	2s
15400K	8%	193M	2s
15450K	8%	211M	2s
15500K	8%	205M	2s
15550K	8%	167M	2s
15600K	8%	188M	2s
15650K	8%	208M	2s
15700K	8%	210M	2s
15750K	8%	195M	2s
15800K	8%	166M	2s
15850K	8%	211M	2s
15900K	8%	203M	2s
15950K	8%	182M	2s

16000K	8%	179M	2s
16050K	8%	205M	2s
16100K	8%	201M	2s
16150K	8%	199M	2s

*** WARNING: skipped 243960 bytes of output ***

176700K	91%	34.5M	0s
176750K	91%	5.51M	0s
176800K	91%	30.0M	0s
176850K	91%	107M	0s
176900K	91%	233M	0s
176950K	91%	37.6M	0s
177000K	91%	34.8M	0s
177050K	91%	26.9M	0s
177100K	91%	36.6M	0s
177150K	91%	53.7M	0s
177200K	91%	46.0M	0s
177250K	91%	201M	0s
177300K	91%	37.1M	0s
177350K	91%	21.4M	0s
177400K	91%	62.4M	0s
177450K	91%	87.4M	0s
177500K	92%	31.2M	0s
177550K	92%	24.7M	0s
177600K	92%	31.9M	0s
177650K	92%	29.4M	0s
177700K	92%	39.1M	0s
177750K	92%	73.0M	0s
177800K	92%	41.1M	0s
177850K	92%	10.3M	0s
177900K	92%	34.7M	0s
177950K	92%	29.7M	0s
178000K	92%	34.6M	0s
178050K	92%	66.8M	0s
178100K	92%	43.0M	0s
178150K	92%	19.8M	0s
178200K	92%	46.7M	0s
178250K	92%	51.5M	0s
178300K	92%	29.7M	0s
178350K	92%	123M	0s
178400K	92%	15.0M	0s
178450K	92%	34.4M	0s
178500K	92%	33.6M	0s
178550K	92%	29.0M	0s
178600K	92%	129M	0s
178650K	92%	10.7M	0s
178700K	92%	125M	0s
178750K	92%	162M	0s
178800K	92%	182M	0s
178850K	92%	203M	0s
178900K	92%	171M	0s
178950K	92%	167M	0s
179000K	92%	172M	0s
179050K	92%	210M	0s

179100K	92%	166M	0s
179150K	92%	90.0M	0s
179200K	92%	24.4M	0s
179250K	92%	93.3M	0s
179300K	92%	116M	0s
179350K	92%	79.3M	0s
179400K	93%	117M	0s
179450K	93%	146M	0s
179500K	93%	153M	0s
179550K	93%	123M	0s
179600K	93%	155M	0s
179650K	93%	152M	0s
179700K	93%	71.4M	0s
179750K	93%	187M	0s
179800K	93%	247M	0s
179850K	93%	215M	0s
179900K	93%	107M	0s
179950K	93%	146M	0s
180000K	93%	160M	0s
180050K	93%	133M	0s
180100K	93%	158M	0s
180150K	93%	128M	0s
180200K	93%	138M	0s
180250K	93%	166M	0s
180300K	93%	168M	0s
180350K	93%	116M	0s
180400K	93%	153M	0s
180450K	93%	157M	0s
180500K	93%	144M	0s
180550K	93%	119M	0s
180600K	93%	160M	0s
180650K	93%	163M	0s
180700K	93%	140M	0s
180750K	93%	137M	0s
180800K	93%	156M	0s
180850K	93%	138M	0s
180900K	93%	143M	0s
180950K	93%	133M	0s
181000K	93%	163M	0s
181050K	93%	144M	0s
181100K	93%	145M	0s
181150K	93%	143M	0s
181200K	93%	131M	0s
181250K	93%	154M	0s
181300K	93%	156M	0s
181350K	94%	120M	0s
181400K	94%	163M	0s
181450K	94%	136M	0s
181500K	94%	111M	0s
181550K	94%	141M	0s
181600K	94%	158M	0s
181650K	94%	150M	0s
181700K	94%	134M	0s
181750K	94%	135M	0s
181800K	94%	162M	0s

181850K	94%	150M	0s
181900K	94%	160M	0s
181950K	94%	145M	0s
182000K	94%	120M	0s
182050K	94%	109M	0s
182100K	94%	144M	0s
182150K	94%	123M	0s
182200K	94%	158M	0s
182250K	94%	155M	0s
182300K	94%	161M	0s
182350K	94%	120M	0s
182400K	94%	157M	0s
182450K	94%	156M	0s
182500K	94%	155M	0s
182550K	94%	124M	0s
182600K	94%	163M	0s
182650K	94%	152M	0s
182700K	94%	163M	0s
182750K	94%	144M	0s
182800K	94%	152M	0s
182850K	94%	153M	0s
182900K	94%	153M	0s
182950K	94%	136M	0s
183000K	94%	145M	0s
183050K	94%	147M	0s
183100K	94%	153M	0s
183150K	94%	140M	0s
183200K	94%	164M	0s
183250K	94%	147M	0s
183300K	95%	174M	0s
183350K	95%	108M	0s
183400K	95%	149M	0s
183450K	95%	147M	0s
183500K	95%	145M	0s
183550K	95%	129M	0s
183600K	95%	141M	0s
183650K	95%	143M	0s
183700K	95%	158M	0s
183750K	95%	127M	0s
183800K	95%	160M	0s
183850K	95%	144M	0s
183900K	95%	158M	0s
183950K	95%	139M	0s
184000K	95%	14.6M	0s
184050K	95%	75.5M	0s
184100K	95%	134M	0s
184150K	95%	65.6M	0s
184200K	95%	126M	0s
184250K	95%	83.2M	0s
184300K	95%	71.8M	0s
184350K	95%	19.0M	0s
184400K	95%	27.4M	0s
184450K	95%	33.4M	0s
184500K	95%	40.5M	0s
184550K	95%	35.7M	0s

184600K	95%	39.4M	0s
184650K	95%	36.4M	0s
184700K	95%	39.7M	0s
184750K	95%	30.9M	0s
184800K	95%	55.7M	0s
184850K	95%	83.3M	0s
184900K	95%	41.9M	0s
184950K	95%	51.7M	0s
185000K	95%	83.0M	0s
185050K	95%	85.3M	0s
185100K	95%	107M	0s
185150K	95%	70.4M	0s
185200K	96%	119M	0s
185250K	96%	34.0M	0s
185300K	96%	55.2M	0s
185350K	96%	52.4M	0s
185400K	96%	64.5M	0s
185450K	96%	84.3M	0s
185500K	96%	77.5M	0s
185550K	96%	79.1M	0s
185600K	96%	84.5M	0s
185650K	96%	38.9M	0s
185700K	96%	78.0M	0s
185750K	96%	81.1M	0s
185800K	96%	147M	0s
185850K	96%	128M	0s
185900K	96%	104M	0s
185950K	96%	140M	0s
186000K	96%	140M	0s
186050K	96%	143M	0s
186100K	96%	150M	0s
186150K	96%	130M	0s
186200K	96%	152M	0s
186250K	96%	137M	0s
186300K	96%	146M	0s
186350K	96%	97.2M	0s
186400K	96%	109M	0s
186450K	96%	106M	0s
186500K	96%	111M	0s
186550K	96%	89.8M	0s
186600K	96%	95.8M	0s
186650K	96%	109M	0s
186700K	96%	83.1M	0s
186750K	96%	141M	0s
186800K	96%	148M	0s
186850K	96%	147M	0s
186900K	96%	151M	0s
186950K	96%	134M	0s
187000K	96%	101M	0s
187050K	96%	107M	0s
187100K	96%	93.1M	0s
187150K	97%	88.1M	0s
187200K	97%	108M	0s
187250K	97%	118M	0s
187300K	97%	85.3M	0s

187350K	97%	104M	0s
187400K	97%	115M	0s
187450K	97%	84.2M	0s
187500K	97%	115M	0s
187550K	97%	149M	0s
187600K	97%	145M	0s
187650K	97%	156M	0s
187700K	97%	82.1M	0s
187750K	97%	98.7M	0s
187800K	97%	85.2M	0s
187850K	97%	112M	0s
187900K	97%	103M	0s
187950K	97%	90.1M	0s
188000K	97%	90.1M	0s
188050K	97%	121M	0s
188100K	97%	102M	0s
188150K	97%	127M	0s
188200K	97%	147M	0s
188250K	97%	150M	0s
188300K	97%	153M	0s
188350K	97%	143M	0s
188400K	97%	84.9M	0s
188450K	97%	101M	0s
188500K	97%	77.8M	0s
188550K	97%	129M	0s
188600K	97%	111M	0s
188650K	97%	101M	0s
188700K	97%	154M	0s
188750K	97%	145M	0s
188800K	97%	112M	0s
188850K	97%	108M	0s
188900K	97%	128M	0s
188950K	97%	120M	0s
189000K	97%	150M	0s
189050K	98%	159M	0s
189100K	98%	156M	0s
189150K	98%	76.8M	0s
189200K	98%	108M	0s
189250K	98%	80.6M	0s
189300K	98%	89.4M	0s
189350K	98%	78.2M	0s
189400K	98%	141M	0s
189450K	98%	145M	0s
189500K	98%	151M	0s
189550K	98%	86.1M	0s
189600K	98%	74.4M	0s
189650K	98%	215M	0s
189700K	98%	227M	0s
189750K	98%	133M	0s
189800K	98%	143M	0s
189850K	98%	156M	0s
189900K	98%	60.1M	0s
189950K	98%	78.5M	0s
190000K	98%	81.9M	0s
190050K	98%	152M	0s

190100K	98%	150M	0s
190150K	98%	129M	0s
190200K	98%	137M	0s
190250K	98%	103M	0s
190300K	98%	88.4M	0s
190350K	98%	106M	0s
190400K	98%	146M	0s
190450K	98%	157M	0s
190500K	98%	146M	0s
190550K	98%	117M	0s
190600K	98%	77.5M	0s
190650K	98%	101M	0s
190700K	98%	88.3M	0s
190750K	98%	138M	0s
190800K	98%	152M	0s
190850K	98%	160M	0s
190900K	98%	155M	0s
190950K	98%	130M	0s
191000K	99%	159M	0s
191050K	99%	91.7M	0s
191100K	99%	83.6M	0s
191150K	99%	82.6M	0s
191200K	99%	151M	0s
191250K	99%	162M	0s
191300K	99%	140M	0s
191350K	99%	64.9M	0s
191400K	99%	109M	0s
191450K	99%	156M	0s
191500K	99%	159M	0s
191550K	99%	147M	0s
191600K	99%	153M	0s
191650K	99%	156M	0s
191700K	99%	150M	0s
191750K	99%	121M	0s
191800K	99%	153M	0s
191850K	99%	148M	0s
191900K	99%	155M	0s
191950K	99%	142M	0s
192000K	99%	139M	0s
192050K	99%	156M	0s
192100K	99%	150M	0s
192150K	99%	134M	0s
192200K	99%	156M	0s
192250K	99%	152M	0s
192300K	99%	163M	0s
192350K	99%	141M	0s
192400K	99%	158M	0s
192450K	99%	149M	0s
192500K	99%	149M	0s
192550K	99%	132M	0s
192600K	99%	160M	0s
192650K	99%	153M	0s
192700K	99%	159M	0s
192750K	99%	147M	0s
192800K	99%	149M	0s


```
192850K ..... 99% 158M 0s
192900K ..... 99% 152M 0s
192950K ..... 100% 54.3M=1.5s
```

2020-05-30 13:04:41 (129 MB/s) - 'name.basics.tsv.gz.1' saved [197587198/197587198]

```
/bin/bash: line 2: fg: no job control
gzip: name.basics.tsv already exists;  not overwritten
```

In [4]:

```
names_basic = spark.read.option("sep", "\t").csv('file:/databricks/driver/name.basics.tsv', header=True, inferSchema = True)
names_basic.show(3)
```

```
+-----+-----+-----+-----+-----+-----+
+-----+
nm0000001|  Fred Astaire|    1899|    1987|soundtrack,actor,...|tt0043044,tt00531...|
nm0000002|  Lauren Bacall|    1924|    2014|  actress,soundtrack|tt0071877,tt01170...|
nm0000003|Brigitte Bardot|    1934|      \N|actress,soundtrac...|tt0054452,tt00491...|
+-----+-----+-----+-----+-----+-----+
+-----+
only showing top 3 rows
```

In [5]:

```
%sh wget https://datasets.imdbws.com/title.basics.tsv.gz
```

```
%sh  
gunzip title.basics.tsv.gz
```

```
--2020-05-30 13:05:03-- https://datasets.imdbws.com/title.basics.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.54, 13.224.13.2
6, 13.224.13.32, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.54|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 121710643 (116M) [binary/octet-stream]
Saving to: 'title.basics.tsv.gz'
```

0K	0%	4.81M	24s
50K	0%	8.09M	19s
100K	0%	15.9M	15s
150K	0%	19.9M	13s
200K	0%	15.7M	12s
250K	0%	22.4M	11s
300K	0%	38.6M	10s
350K	0%	28.2M	9s
400K	0%	45.7M	8s
450K	0%	34.2M	8s
500K	0%	39.1M	7s
550K	0%	105M	7s
600K	0%	52.0M	6s
650K	0%	52.9M	6s
700K	0%	54.0M	6s
750K	0%	83.0M	6s
800K	0%	79.6M	5s
850K	0%	65.8M	5s
900K	0%	85.5M	5s
950K	0%	141M	5s
1000K	0%	81.7M	5s
1050K	0%	57.3M	4s
1100K	0%	93.1M	4s
1150K	1%	94.5M	4s
1200K	1%	133M	4s
1250K	1%	274M	4s
1300K	1%	89.2M	4s
1350K	1%	97.6M	4s
1400K	1%	118M	4s
1450K	1%	104M	4s
1500K	1%	296M	3s
1550K	1%	187M	3s
1600K	1%	84.3M	3s
1650K	1%	103M	3s
1700K	1%	27.5M	3s
1750K	1%	101M	3s
1800K	1%	83.3M	3s
1850K	1%	73.8M	3s
1900K	1%	84.3M	3s
1950K	1%	88.9M	3s
2000K	1%	64.9M	3s
2050K	1%	69.1M	3s
2100K	1%	157M	3s
2150K	1%	182M	3s
2200K	1%	281M	3s

2250K	1%	148M	3s
2300K	1%	251M	3s
2350K	2%	147M	3s
2400K	2%	180M	3s
2450K	2%	158M	3s
2500K	2%	260M	2s
2550K	2%	151M	2s
2600K	2%	174M	2s
2650K	2%	152M	2s
2700K	2%	267M	2s
2750K	2%	161M	2s
2800K	2%	185M	2s
2850K	2%	134M	2s
2900K	2%	116M	2s
2950K	2%	55.8M	2s
3000K	2%	83.3M	2s
3050K	2%	58.8M	2s
3100K	2%	72.3M	2s
3150K	2%	80.4M	2s
3200K	2%	115M	2s
3250K	2%	68.2M	2s
3300K	2%	96.1M	2s
3350K	2%	62.7M	2s
3400K	2%	24.0M	2s
3450K	2%	89.8M	2s
3500K	2%	152M	2s
3550K	3%	106M	2s
3600K	3%	125M	2s
3650K	3%	119M	2s
3700K	3%	155M	2s
3750K	3%	138M	2s
3800K	3%	151M	2s
3850K	3%	133M	2s
3900K	3%	99.0M	2s
3950K	3%	150M	2s
4000K	3%	149M	2s
4050K	3%	133M	2s
4100K	3%	107M	2s
4150K	3%	151M	2s
4200K	3%	124M	2s
4250K	3%	100M	2s
4300K	3%	47.3M	2s
4350K	3%	154M	2s
4400K	3%	151M	2s
4450K	3%	138M	2s
4500K	3%	160M	2s
4550K	3%	151M	2s
4600K	3%	89.4M	2s
4650K	3%	126M	2s
4700K	3%	156M	2s
4750K	4%	148M	2s
4800K	4%	151M	2s
4850K	4%	59.6M	2s
4900K	4%	143M	2s
4950K	4%	84.8M	2s

5000K	4%	106M	2s
5050K	4%	80.0M	2s
5100K	4%	142M	2s
5150K	4%	160M	2s
5200K	4%	159M	2s
5250K	4%	177M	2s
5300K	4%	280M	2s
5350K	4%	278M	2s
5400K	4%	307M	2s
5450K	4%	258M	2s
5500K	4%	187M	2s
5550K	4%	279M	2s
5600K	4%	226M	2s
5650K	4%	278M	2s
5700K	4%	321M	2s
5750K	4%	321M	2s
5800K	4%	278M	2s
5850K	4%	53.0M	2s
5900K	5%	290M	2s
5950K	5%	265M	2s
6000K	5%	289M	2s
6050K	5%	263M	2s
6100K	5%	107M	2s
6150K	5%	272M	2s
6200K	5%	292M	2s
6250K	5%	232M	2s
6300K	5%	284M	2s
6350K	5%	314M	2s
6400K	5%	228M	1s
6450K	5%	84.1M	1s
6500K	5%	135M	1s
6550K	5%	106M	1s
6600K	5%	136M	1s
6650K	5%	137M	1s
6700K	5%	163M	1s
6750K	5%	156M	1s
6800K	5%	160M	1s
6850K	5%	43.5M	1s
6900K	5%	106M	1s
6950K	5%	92.1M	1s
7000K	5%	110M	1s
7050K	5%	84.1M	1s
7100K	6%	85.7M	1s
7150K	6%	117M	1s
7200K	6%	120M	1s
7250K	6%	80.7M	1s
7300K	6%	86.8M	1s
7350K	6%	120M	1s
7400K	6%	103M	1s
7450K	6%	135M	1s
7500K	6%	161M	1s
7550K	6%	171M	1s
7600K	6%	157M	1s
7650K	6%	90.5M	1s
7700K	6%	84.2M	1s

7750K	6%	169M	1s
7800K	6%	151M	1s
7850K	6%	137M	1s
7900K	6%	279M	1s
7950K	6%	185M	1s
8000K	6%	91.6M	1s
8050K	6%	183M	1s
8100K	6%	197M	1s
8150K	6%	202M	1s
8200K	6%	207M	1s
8250K	6%	80.3M	1s
8300K	7%	101M	1s
8350K	7%	148M	1s
8400K	7%	154M	1s
8450K	7%	143M	1s
8500K	7%	150M	1s
8550K	7%	143M	1s
8600K	7%	155M	1s
8650K	7%	137M	1s
8700K	7%	77.4M	1s
8750K	7%	96.2M	1s
8800K	7%	80.9M	1s
8850K	7%	106M	1s
8900K	7%	130M	1s
8950K	7%	150M	1s
9000K	7%	57.8M	1s
9050K	7%	139M	1s
9100K	7%	156M	1s
9150K	7%	83.5M	1s
9200K	7%	68.1M	1s
9250K	7%	117M	1s
9300K	7%	154M	1s
9350K	7%	139M	1s
9400K	7%	152M	1s
9450K	7%	140M	1s
9500K	8%	137M	1s
9550K	8%	69.9M	1s
9600K	8%	78.5M	1s
9650K	8%	53.8M	1s
9700K	8%	95.9M	1s
9750K	8%	70.5M	1s
9800K	8%	167M	1s
9850K	8%	123M	1s
9900K	8%	162M	1s
9950K	8%	97.5M	1s
10000K	8%	162M	1s
10050K	8%	133M	1s
10100K	8%	148M	1s
10150K	8%	157M	1s
10200K	8%	147M	1s
10250K	8%	137M	1s
10300K	8%	197M	1s
10350K	8%	165M	1s
10400K	8%	164M	1s
10450K	8%	71.9M	1s

10500K	8%	91.9M	1s
10550K	8%	108M	1s
10600K	8%	99.2M	1s
10650K	9%	134M	1s
10700K	9%	171M	1s
10750K	9%	156M	1s
10800K	9%	158M	1s
10850K	9%	137M	1s
10900K	9%	290M	1s
10950K	9%	273M	1s
11000K	9%	313M	1s
11050K	9%	125M	1s
11100K	9%	160M	1s
11150K	9%	270M	1s
11200K	9%	314M	1s
11250K	9%	292M	1s
11300K	9%	284M	1s
11350K	9%	309M	1s
11400K	9%	281M	1s
11450K	9%	173M	1s
11500K	9%	125M	1s
11550K	9%	156M	1s
11600K	9%	151M	1s
11650K	9%	85.5M	1s
11700K	9%	87.5M	1s
11750K	9%	85.1M	1s
11800K	9%	70.0M	1s
11850K	10%	72.8M	1s
11900K	10%	142M	1s
11950K	10%	190M	1s
12000K	10%	203M	1s
12050K	10%	181M	1s
12100K	10%	227M	1s
12150K	10%	185M	1s
12200K	10%	55.1M	1s
12250K	10%	66.8M	1s
12300K	10%	81.6M	1s
12350K	10%	85.7M	1s
12400K	10%	211M	1s
12450K	10%	192M	1s
12500K	10%	75.5M	1s
12550K	10%	97.6M	1s
12600K	10%	104M	1s
12650K	10%	75.9M	1s
12700K	10%	32.8M	1s
12750K	10%	26.5M	1s
12800K	10%	39.7M	1s
12850K	10%	35.3M	1s
12900K	10%	10.7M	1s
12950K	10%	144M	1s
13000K	10%	135M	1s
13050K	11%	134M	1s
13100K	11%	144M	1s
13150K	11%	118M	1s
13200K	11%	131M	1s

13250K	11%	136M	1s
13300K	11%	115M	1s
13350K	11%	153M	1s
13400K	11%	159M	1s
13450K	11%	126M	1s
13500K	11%	165M	1s
13550K	11%	164M	1s
13600K	11%	163M	1s
13650K	11%	136M	1s
13700K	11%	144M	1s
13750K	11%	153M	1s
13800K	11%	136M	1s
13850K	11%	136M	1s
13900K	11%	140M	1s
13950K	11%	138M	1s
14000K	11%	159M	1s
14050K	11%	41.3M	1s
14100K	11%	50.7M	1s
14150K	11%	130M	1s
14200K	11%	148M	1s
14250K	12%	79.8M	1s
14300K	12%	184M	1s
14350K	12%	297M	1s
14400K	12%	261M	1s
14450K	12%	272M	1s
14500K	12%	292M	1s
14550K	12%	264M	1s
14600K	12%	297M	1s
14650K	12%	244M	1s
14700K	12%	269M	1s
14750K	12%	291M	1s
14800K	12%	301M	1s
14850K	12%	102M	1s
14900K	12%	163M	1s
14950K	12%	84.7M	1s
15000K	12%	123M	1s
15050K	12%	138M	1s
15100K	12%	190M	1s
15150K	12%	277M	1s
15200K	12%	299M	1s
15250K	12%	119M	1s
15300K	12%	187M	1s
15350K	12%	110M	1s
15400K	12%	177M	1s
15450K	13%	146M	1s
15500K	13%	302M	1s
15550K	13%	276M	1s
15600K	13%	264M	1s
15650K	13%	100M	1s
15700K	13%	121M	1s
15750K	13%	150M	1s
15800K	13%	159M	1s
15850K	13%	157M	1s
15900K	13%	316M	1s
15950K	13%	281M	1s

16000K	13%	316M	1s
16050K	13%	287M	1s
16100K	13%	276M	1s
16150K	13%	134M	1s

*** WARNING: skipped 131252 bytes of output ***

102550K	86%	88.1M	0s
102600K	86%	128M	0s
102650K	86%	133M	0s
102700K	86%	119M	0s
102750K	86%	293M	0s
102800K	86%	253M	0s
102850K	86%	255M	0s
102900K	86%	234M	0s
102950K	86%	283M	0s
103000K	86%	293M	0s
103050K	86%	301M	0s
103100K	86%	137M	0s
103150K	86%	254M	0s
103200K	86%	178M	0s
103250K	86%	209M	0s
103300K	86%	102M	0s
103350K	86%	238M	0s
103400K	87%	172M	0s
103450K	87%	145M	0s
103500K	87%	165M	0s
103550K	87%	190M	0s
103600K	87%	178M	0s
103650K	87%	265M	0s
103700K	87%	178M	0s
103750K	87%	300M	0s
103800K	87%	297M	0s
103850K	87%	305M	0s
103900K	87%	185M	0s
103950K	87%	186M	0s
104000K	87%	297M	0s
104050K	87%	298M	0s
104100K	87%	204M	0s
104150K	87%	106M	0s
104200K	87%	278M	0s
104250K	87%	300M	0s
104300K	87%	180M	0s
104350K	87%	254M	0s
104400K	87%	266M	0s
104450K	87%	260M	0s
104500K	87%	269M	0s
104550K	88%	297M	0s
104600K	88%	306M	0s
104650K	88%	243M	0s
104700K	88%	211M	0s
104750K	88%	171M	0s
104800K	88%	55.2M	0s
104850K	88%	127M	0s
104900K	88%	134M	0s

104950K	88%	90.0M	0s
105000K	88%	146M	0s
105050K	88%	184M	0s
105100K	88%	139M	0s
105150K	88%	184M	0s
105200K	88%	200M	0s
105250K	88%	206M	0s
105300K	88%	153M	0s
105350K	88%	187M	0s
105400K	88%	191M	0s
105450K	88%	120M	0s
105500K	88%	107M	0s
105550K	88%	146M	0s
105600K	88%	143M	0s
105650K	88%	145M	0s
105700K	88%	135M	0s
105750K	89%	138M	0s
105800K	89%	103M	0s
105850K	89%	132M	0s
105900K	89%	128M	0s
105950K	89%	165M	0s
106000K	89%	157M	0s
106050K	89%	132M	0s
106100K	89%	155M	0s
106150K	89%	201M	0s
106200K	89%	272M	0s
106250K	89%	228M	0s
106300K	89%	151M	0s
106350K	89%	193M	0s
106400K	89%	153M	0s
106450K	89%	227M	0s
106500K	89%	187M	0s
106550K	89%	156M	0s
106600K	89%	170M	0s
106650K	89%	188M	0s
106700K	89%	224M	0s
106750K	89%	118M	0s
106800K	89%	278M	0s
106850K	89%	281M	0s
106900K	89%	276M	0s
106950K	90%	285M	0s
107000K	90%	225M	0s
107050K	90%	299M	0s
107100K	90%	248M	0s
107150K	90%	263M	0s
107200K	90%	310M	0s
107250K	90%	223M	0s
107300K	90%	178M	0s
107350K	90%	197M	0s
107400K	90%	206M	0s
107450K	90%	191M	0s
107500K	90%	135M	0s
107550K	90%	210M	0s
107600K	90%	268M	0s
107650K	90%	208M	0s

107700K	90%	96.6M	0s
107750K	90%	140M	0s
107800K	90%	170M	0s
107850K	90%	175M	0s
107900K	90%	212M	0s
107950K	90%	214M	0s
108000K	90%	278M	0s
108050K	90%	220M	0s
108100K	90%	198M	0s
108150K	91%	295M	0s
108200K	91%	294M	0s
108250K	91%	239M	0s
108300K	91%	185M	0s
108350K	91%	302M	0s
108400K	91%	179M	0s
108450K	91%	134M	0s
108500K	91%	242M	0s
108550K	91%	307M	0s
108600K	91%	293M	0s
108650K	91%	270M	0s
108700K	91%	156M	0s
108750K	91%	200M	0s
108800K	91%	202M	0s
108850K	91%	193M	0s
108900K	91%	173M	0s
108950K	91%	182M	0s
109000K	91%	191M	0s
109050K	91%	38.1M	0s
109100K	91%	35.3M	0s
109150K	91%	39.3M	0s
109200K	91%	41.0M	0s
109250K	91%	44.6M	0s
109300K	92%	36.5M	0s
109350K	92%	38.6M	0s
109400K	92%	19.4M	0s
109450K	92%	41.2M	0s
109500K	92%	30.5M	0s
109550K	92%	41.0M	0s
109600K	92%	37.2M	0s
109650K	92%	37.6M	0s
109700K	92%	34.0M	0s
109750K	92%	23.5M	0s
109800K	92%	26.6M	0s
109850K	92%	19.6M	0s
109900K	92%	34.6M	0s
109950K	92%	21.2M	0s
110000K	92%	200M	0s
110050K	92%	143M	0s
110100K	92%	168M	0s
110150K	92%	188M	0s
110200K	92%	187M	0s
110250K	92%	147M	0s
110300K	92%	152M	0s
110350K	92%	188M	0s
110400K	92%	188M	0s

110450K	92%	154M	0s
110500K	93%	100M	0s
110550K	93%	66.5M	0s
110600K	93%	129M	0s
110650K	93%	191M	0s
110700K	93%	55.3M	0s
110750K	93%	69.6M	0s
110800K	93%	64.5M	0s
110850K	93%	80.8M	0s
110900K	93%	128M	0s
110950K	93%	267M	0s
111000K	93%	264M	0s
111050K	93%	254M	0s
111100K	93%	250M	0s
111150K	93%	305M	0s
111200K	93%	305M	0s
111250K	93%	307M	0s
111300K	93%	250M	0s
111350K	93%	264M	0s
111400K	93%	302M	0s
111450K	93%	296M	0s
111500K	93%	252M	0s
111550K	93%	299M	0s
111600K	93%	270M	0s
111650K	93%	265M	0s
111700K	94%	94.9M	0s
111750K	94%	155M	0s
111800K	94%	77.5M	0s
111850K	94%	84.6M	0s
111900K	94%	153M	0s
111950K	94%	184M	0s
112000K	94%	184M	0s
112050K	94%	174M	0s
112100K	94%	162M	0s
112150K	94%	163M	0s
112200K	94%	183M	0s
112250K	94%	176M	0s
112300K	94%	152M	0s
112350K	94%	182M	0s
112400K	94%	171M	0s
112450K	94%	186M	0s
112500K	94%	232M	0s
112550K	94%	291M	0s
112600K	94%	295M	0s
112650K	94%	259M	0s
112700K	94%	241M	0s
112750K	94%	267M	0s
112800K	94%	257M	0s
112850K	94%	289M	0s
112900K	95%	270M	0s
112950K	95%	305M	0s
113000K	95%	304M	0s
113050K	95%	295M	0s
113100K	95%	109M	0s
113150K	95%	109M	0s

113200K	95%	164M	0s
113250K	95%	79.0M	0s
113300K	95%	127M	0s
113350K	95%	176M	0s
113400K	95%	181M	0s
113450K	95%	178M	0s
113500K	95%	153M	0s
113550K	95%	179M	0s
113600K	95%	55.3M	0s
113650K	95%	67.1M	0s
113700K	95%	171M	0s
113750K	95%	158M	0s
113800K	95%	202M	0s
113850K	95%	208M	0s
113900K	95%	171M	0s
113950K	95%	214M	0s
114000K	95%	200M	0s
114050K	95%	210M	0s
114100K	96%	193M	0s
114150K	96%	115M	0s
114200K	96%	23.7M	0s
114250K	96%	19.7M	0s
114300K	96%	32.9M	0s
114350K	96%	42.0M	0s
114400K	96%	48.2M	0s
114450K	96%	88.5M	0s
114500K	96%	88.0M	0s
114550K	96%	29.4M	0s
114600K	96%	148M	0s
114650K	96%	153M	0s
114700K	96%	103M	0s
114750K	96%	165M	0s
114800K	96%	150M	0s
114850K	96%	168M	0s
114900K	96%	138M	0s
114950K	96%	147M	0s
115000K	96%	147M	0s
115050K	96%	154M	0s
115100K	96%	138M	0s
115150K	96%	130M	0s
115200K	96%	158M	0s
115250K	97%	153M	0s
115300K	97%	136M	0s
115350K	97%	52.5M	0s
115400K	97%	39.0M	0s
115450K	97%	24.2M	0s
115500K	97%	34.4M	0s
115550K	97%	42.2M	0s
115600K	97%	39.8M	0s
115650K	97%	42.6M	0s
115700K	97%	23.4M	0s
115750K	97%	19.2M	0s
115800K	97%	39.8M	0s
115850K	97%	41.7M	0s
115900K	97%	32.0M	0s

115950K	97%	5.11M	0s
116000K	97%	43.7M	0s
116050K	97%	43.4M	0s
116100K	97%	38.7M	0s
116150K	97%	43.4M	0s
116200K	97%	44.7M	0s
116250K	97%	40.5M	0s
116300K	97%	38.9M	0s
116350K	97%	44.6M	0s
116400K	97%	38.8M	0s
116450K	98%	52.2M	0s
116500K	98%	172M	0s
116550K	98%	186M	0s
116600K	98%	181M	0s
116650K	98%	189M	0s
116700K	98%	154M	0s
116750K	98%	188M	0s
116800K	98%	194M	0s
116850K	98%	180M	0s
116900K	98%	170M	0s
116950K	98%	175M	0s
117000K	98%	185M	0s
117050K	98%	190M	0s
117100K	98%	154M	0s
117150K	98%	190M	0s
117200K	98%	159M	0s
117250K	98%	190M	0s
117300K	98%	191M	0s
117350K	98%	203M	0s
117400K	98%	186M	0s
117450K	98%	201M	0s
117500K	98%	219M	0s
117550K	98%	308M	0s
117600K	98%	305M	0s
117650K	99%	310M	0s
117700K	99%	242M	0s
117750K	99%	266M	0s
117800K	99%	258M	0s
117850K	99%	198M	0s
117900K	99%	147M	0s
117950K	99%	197M	0s
118000K	99%	201M	0s
118050K	99%	187M	0s
118100K	99%	169M	0s
118150K	99%	189M	0s
118200K	99%	180M	0s
118250K	99%	46.7M	0s
118300K	99%	242M	0s
118350K	99%	308M	0s
118400K	99%	309M	0s
118450K	99%	308M	0s
118500K	99%	194M	0s
118550K	99%	198M	0s
118600K	99%	191M	0s
118650K	99%	186M	0s

```
118700K ..... 99% 150M 0s
118750K ..... 99% 189M 0s
118800K ..... 99% 175M 0s
118850K ..... 100% 219M=1.0s
```

2020-05-30 13:05:04 (111 MB/s) - 'title.basics.tsv.gz' saved [121710643/121710643]

/bin/bash: line 2: fg: no job control

In [6]:

```
title_basics = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.basics.tsv', header=True, inferSchema = True)
title_basics.show(3)
```

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
tt0000001	short	Carmencita	Carmencita	0	1894		1	Documentary,Short
tt0000002	short	Le clown et ses c...	Le clown et ses c...	0	1892		5	Animation,Short
tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892		4	Animation,Comedy,...

only showing top 3 rows

In [7]:

```
%sh wget https://datasets.imdbws.com/title.akas.tsv.gz
```

```
%sh  
gunzip title.akas.tsv.gz
```



```
--2020-05-30 13:05:29-- https://datasets.imdbws.com/title.akas.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.37, 13.224.13.5
4, 13.224.13.26, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.37|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 192592731 (184M) [binary/octet-stream]
Saving to: 'title.akas.tsv.gz'
```

0K	0%	4.02M	46s
50K	0%	7.79M	35s
100K	0%	16.6M	27s
150K	0%	12.7M	24s
200K	0%	19.5M	21s
250K	0%	27.4M	18s
300K	0%	23.2M	17s
350K	0%	39.4M	15s
400K	0%	25.9M	14s
450K	0%	33.6M	14s
500K	0%	45.1M	13s
550K	0%	39.6M	12s
600K	0%	62.8M	11s
650K	0%	45.7M	11s
700K	0%	75.0M	10s
750K	0%	67.8M	10s
800K	0%	46.1M	9s
850K	0%	86.1M	9s
900K	0%	70.6M	9s
950K	0%	68.7M	8s
1000K	0%	89.5M	8s
1050K	0%	63.2M	8s
1100K	0%	106M	8s
1150K	0%	70.4M	7s
1200K	0%	106M	7s
1250K	0%	201M	7s
1300K	0%	87.1M	7s
1350K	0%	85.1M	7s
1400K	0%	110M	6s
1450K	0%	151M	6s
1500K	0%	115M	6s
1550K	0%	110M	6s
1600K	0%	108M	6s
1650K	0%	113M	6s
1700K	0%	121M	6s
1750K	0%	111M	5s
1800K	0%	128M	5s
1850K	1%	292M	5s
1900K	1%	167M	5s
1950K	1%	270M	5s
2000K	1%	144M	5s
2050K	1%	175M	5s
2100K	1%	105M	5s
2150K	1%	92.5M	5s
2200K	1%	100M	5s

2250K	1%	89.1M	5s
2300K	1%	112M	4s
2350K	1%	88.2M	4s
2400K	1%	87.1M	4s
2450K	1%	126M	4s
2500K	1%	161M	4s
2550K	1%	178M	4s
2600K	1%	138M	4s
2650K	1%	179M	4s
2700K	1%	266M	4s
2750K	1%	182M	4s
2800K	1%	220M	4s
2850K	1%	106M	4s
2900K	1%	108M	4s
2950K	1%	106M	4s
3000K	1%	118M	4s
3050K	1%	177M	4s
3100K	1%	164M	4s
3150K	1%	278M	4s
3200K	1%	143M	4s
3250K	1%	296M	4s
3300K	1%	191M	3s
3350K	1%	166M	3s
3400K	1%	150M	3s
3450K	1%	110M	3s
3500K	1%	59.4M	3s
3550K	1%	111M	3s
3600K	1%	60.4M	3s
3650K	1%	91.4M	3s
3700K	1%	90.3M	3s
3750K	2%	115M	3s
3800K	2%	103M	3s
3850K	2%	156M	3s
3900K	2%	118M	3s
3950K	2%	123M	3s
4000K	2%	113M	3s
4050K	2%	113M	3s
4100K	2%	168M	3s
4150K	2%	172M	3s
4200K	2%	262M	3s
4250K	2%	152M	3s
4300K	2%	283M	3s
4350K	2%	186M	3s
4400K	2%	119M	3s
4450K	2%	80.2M	3s
4500K	2%	116M	3s
4550K	2%	83.0M	3s
4600K	2%	65.1M	3s
4650K	2%	119M	3s
4700K	2%	91.3M	3s
4750K	2%	93.1M	3s
4800K	2%	96.7M	3s
4850K	2%	110M	3s
4900K	2%	159M	3s
4950K	2%	109M	3s

5000K	2%	145M	3s
5050K	2%	173M	3s
5100K	2%	264M	3s
5150K	2%	169M	3s
5200K	2%	156M	3s
5250K	2%	239M	3s
5300K	2%	165M	3s
5350K	2%	160M	3s
5400K	2%	17.0M	3s
5450K	2%	43.4M	3s
5500K	2%	86.4M	3s
5550K	2%	141M	3s
5600K	3%	128M	3s
5650K	3%	286M	3s
5700K	3%	181M	3s
5750K	3%	127M	3s
5800K	3%	105M	3s
5850K	3%	283M	3s
5900K	3%	106M	3s
5950K	3%	105M	3s
6000K	3%	264M	3s
6050K	3%	105M	3s
6100K	3%	105M	3s
6150K	3%	29.1M	3s
6200K	3%	64.4M	3s
6250K	3%	138M	3s
6300K	3%	140M	3s
6350K	3%	144M	3s
6400K	3%	127M	3s
6450K	3%	157M	3s
6500K	3%	152M	3s
6550K	3%	139M	3s
6600K	3%	115M	3s
6650K	3%	148M	3s
6700K	3%	63.8M	3s
6750K	3%	266M	3s
6800K	3%	149M	3s
6850K	3%	298M	3s
6900K	3%	112M	3s
6950K	3%	154M	2s
7000K	3%	88.4M	2s
7050K	3%	142M	2s
7100K	3%	164M	2s
7150K	3%	158M	2s
7200K	3%	105M	2s
7250K	3%	136M	2s
7300K	3%	156M	2s
7350K	3%	138M	2s
7400K	3%	140M	2s
7450K	3%	149M	2s
7500K	4%	168M	2s
7550K	4%	308M	2s
7600K	4%	290M	2s
7650K	4%	272M	2s
7700K	4%	285M	2s

7750K	4%	312M	2s
7800K	4%	257M	2s
7850K	4%	273M	2s
7900K	4%	311M	2s
7950K	4%	161M	2s
8000K	4%	156M	2s
8050K	4%	168M	2s
8100K	4%	173M	2s
8150K	4%	155M	2s
8200K	4%	159M	2s
8250K	4%	168M	2s
8300K	4%	42.9M	2s
8350K	4%	85.7M	2s
8400K	4%	80.8M	2s
8450K	4%	155M	2s
8500K	4%	87.9M	2s
8550K	4%	75.0M	2s
8600K	4%	130M	2s
8650K	4%	162M	2s
8700K	4%	161M	2s
8750K	4%	155M	2s
8800K	4%	144M	2s
8850K	4%	163M	2s
8900K	4%	158M	2s
8950K	4%	159M	2s
9000K	4%	79.8M	2s
9050K	4%	85.4M	2s
9100K	4%	89.5M	2s
9150K	4%	89.8M	2s
9200K	4%	141M	2s
9250K	4%	158M	2s
9300K	4%	151M	2s
9350K	4%	114M	2s
9400K	5%	142M	2s
9450K	5%	164M	2s
9500K	5%	161M	2s
9550K	5%	171M	2s
9600K	5%	160M	2s
9650K	5%	155M	2s
9700K	5%	53.5M	2s
9750K	5%	153M	2s
9800K	5%	131M	2s
9850K	5%	161M	2s
9900K	5%	160M	2s
9950K	5%	157M	2s
10000K	5%	244M	2s
10050K	5%	306M	2s
10100K	5%	303M	2s
10150K	5%	264M	2s
10200K	5%	114M	2s
10250K	5%	151M	2s
10300K	5%	68.9M	2s
10350K	5%	100M	2s
10400K	5%	84.4M	2s
10450K	5%	85.3M	2s

10500K	5%	118M	2s
10550K	5%	179M	2s
10600K	5%	201M	2s
10650K	5%	303M	2s
10700K	5%	164M	2s
10750K	5%	189M	2s
10800K	5%	150M	2s
10850K	5%	229M	2s
10900K	5%	304M	2s
10950K	5%	265M	2s
11000K	5%	241M	2s
11050K	5%	309M	2s
11100K	5%	269M	2s
11150K	5%	307M	2s
11200K	5%	270M	2s
11250K	6%	268M	2s
11300K	6%	301M	2s
11350K	6%	215M	2s
11400K	6%	80.8M	2s
11450K	6%	156M	2s
11500K	6%	139M	2s
11550K	6%	139M	2s
11600K	6%	54.5M	2s
11650K	6%	91.8M	2s
11700K	6%	87.1M	2s
11750K	6%	114M	2s
11800K	6%	69.5M	2s
11850K	6%	112M	2s
11900K	6%	105M	2s
11950K	6%	68.9M	2s
12000K	6%	106M	2s
12050K	6%	131M	2s
12100K	6%	140M	2s
12150K	6%	48.6M	2s
12200K	6%	133M	2s
12250K	6%	127M	2s
12300K	6%	159M	2s
12350K	6%	121M	2s
12400K	6%	81.1M	2s
12450K	6%	87.9M	2s
12500K	6%	86.9M	2s
12550K	6%	84.1M	2s
12600K	6%	78.1M	2s
12650K	6%	89.0M	2s
12700K	6%	87.1M	2s
12750K	6%	86.3M	2s
12800K	6%	85.8M	2s
12850K	6%	44.3M	2s
12900K	6%	26.0M	2s
12950K	6%	46.9M	2s
13000K	6%	29.3M	2s
13050K	6%	34.9M	2s
13100K	6%	53.2M	2s
13150K	7%	45.4M	2s
13200K	7%	78.8M	2s

13250K	7%	44.5M	2s
13300K	7%	36.9M	2s
13350K	7%	41.7M	2s
13400K	7%	117M	2s
13450K	7%	150M	2s
13500K	7%	143M	2s
13550K	7%	135M	2s
13600K	7%	127M	2s
13650K	7%	139M	2s
13700K	7%	152M	2s
13750K	7%	139M	2s
13800K	7%	121M	2s
13850K	7%	198M	2s
13900K	7%	299M	2s
13950K	7%	161M	2s
14000K	7%	164M	2s
14050K	7%	159M	2s
14100K	7%	185M	2s
14150K	7%	169M	2s
14200K	7%	132M	2s
14250K	7%	188M	2s
14300K	7%	166M	2s
14350K	7%	182M	2s
14400K	7%	140M	2s
14450K	7%	164M	2s
14500K	7%	160M	2s
14550K	7%	195M	2s
14600K	7%	144M	2s
14650K	7%	188M	2s
14700K	7%	159M	2s
14750K	7%	188M	2s
14800K	7%	170M	2s
14850K	7%	188M	2s
14900K	7%	145M	2s
14950K	7%	184M	2s
15000K	8%	152M	2s
15050K	8%	185M	2s
15100K	8%	167M	2s
15150K	8%	175M	2s
15200K	8%	186M	2s
15250K	8%	160M	2s
15300K	8%	169M	2s
15350K	8%	193M	2s
15400K	8%	161M	2s
15450K	8%	194M	2s
15500K	8%	173M	2s
15550K	8%	194M	2s
15600K	8%	163M	2s
15650K	8%	193M	2s
15700K	8%	169M	2s
15750K	8%	175M	2s
15800K	8%	146M	2s
15850K	8%	191M	2s
15900K	8%	170M	2s
15950K	8%	195M	2s

16000K	8%	147M	2s
16050K	8%	197M	2s
16100K	8%	163M	2s
16150K	8%	185M	2s

*** WARNING: skipped 236436 bytes of output ***

171750K	91%	136M	0s
171800K	91%	175M	0s
171850K	91%	155M	0s
171900K	91%	147M	0s
171950K	91%	89.2M	0s
172000K	91%	128M	0s
172050K	91%	113M	0s
172100K	91%	105M	0s
172150K	91%	131M	0s
172200K	91%	149M	0s
172250K	91%	157M	0s
172300K	91%	158M	0s
172350K	91%	125M	0s
172400K	91%	136M	0s
172450K	91%	131M	0s
172500K	91%	113M	0s
172550K	91%	158M	0s
172600K	91%	143M	0s
172650K	91%	72.2M	0s
172700K	91%	133M	0s
172750K	91%	144M	0s
172800K	91%	143M	0s
172850K	91%	158M	0s
172900K	91%	128M	0s
172950K	91%	171M	0s
173000K	92%	154M	0s
173050K	92%	145M	0s
173100K	92%	134M	0s
173150K	92%	144M	0s
173200K	92%	159M	0s
173250K	92%	95.6M	0s
173300K	92%	110M	0s
173350K	92%	143M	0s
173400K	92%	110M	0s
173450K	92%	150M	0s
173500K	92%	122M	0s
173550K	92%	145M	0s
173600K	92%	135M	0s
173650K	92%	147M	0s
173700K	92%	107M	0s
173750K	92%	92.6M	0s
173800K	92%	128M	0s
173850K	92%	162M	0s
173900K	92%	153M	0s
173950K	92%	156M	0s
174000K	92%	143M	0s
174050K	92%	166M	0s
174100K	92%	114M	0s

174150K	92%	136M	0s
174200K	92%	114M	0s
174250K	92%	130M	0s
174300K	92%	88.4M	0s
174350K	92%	131M	0s
174400K	92%	87.2M	0s
174450K	92%	105M	0s
174500K	92%	113M	0s
174550K	92%	154M	0s
174600K	92%	104M	0s
174650K	92%	165M	0s
174700K	92%	83.2M	0s
174750K	92%	136M	0s
174800K	92%	96.7M	0s
174850K	92%	144M	0s
174900K	93%	131M	0s
174950K	93%	153M	0s
175000K	93%	160M	0s
175050K	93%	161M	0s
175100K	93%	150M	0s
175150K	93%	145M	0s
175200K	93%	146M	0s
175250K	93%	129M	0s
175300K	93%	108M	0s
175350K	93%	97.9M	0s
175400K	93%	152M	0s
175450K	93%	130M	0s
175500K	93%	122M	0s
175550K	93%	87.8M	0s
175600K	93%	130M	0s
175650K	93%	143M	0s
175700K	93%	123M	0s
175750K	93%	167M	0s
175800K	93%	170M	0s
175850K	93%	127M	0s
175900K	93%	155M	0s
175950K	93%	154M	0s
176000K	93%	149M	0s
176050K	93%	160M	0s
176100K	93%	124M	0s
176150K	93%	46.5M	0s
176200K	93%	42.7M	0s
176250K	93%	40.8M	0s
176300K	93%	28.9M	0s
176350K	93%	38.5M	0s
176400K	93%	37.7M	0s
176450K	93%	38.4M	0s
176500K	93%	33.4M	0s
176550K	93%	37.4M	0s
176600K	93%	24.3M	0s
176650K	93%	38.1M	0s
176700K	93%	34.8M	0s
176750K	94%	39.4M	0s
176800K	94%	38.3M	0s
176850K	94%	38.8M	0s

176900K	94%	34.1M	0s
176950K	94%	38.5M	0s
177000K	94%	39.0M	0s
177050K	94%	39.3M	0s
177100K	94%	10.7M	0s
177150K	94%	39.2M	0s
177200K	94%	29.5M	0s
177250K	94%	38.3M	0s
177300K	94%	33.7M	0s
177350K	94%	39.4M	0s
177400K	94%	21.9M	0s
177450K	94%	12.9M	0s
177500K	94%	19.7M	0s
177550K	94%	38.0M	0s
177600K	94%	38.2M	0s
177650K	94%	39.5M	0s
177700K	94%	32.9M	0s
177750K	94%	38.9M	0s
177800K	94%	38.9M	0s
177850K	94%	36.8M	0s
177900K	94%	34.6M	0s
177950K	94%	37.7M	0s
178000K	94%	37.3M	0s
178050K	94%	59.8M	0s
178100K	94%	163M	0s
178150K	94%	193M	0s
178200K	94%	198M	0s
178250K	94%	200M	0s
178300K	94%	180M	0s
178350K	94%	202M	0s
178400K	94%	199M	0s
178450K	94%	199M	0s
178500K	94%	166M	0s
178550K	94%	197M	0s
178600K	94%	154M	0s
178650K	95%	196M	0s
178700K	95%	181M	0s
178750K	95%	198M	0s
178800K	95%	197M	0s
178850K	95%	199M	0s
178900K	95%	163M	0s
178950K	95%	201M	0s
179000K	95%	201M	0s
179050K	95%	194M	0s
179100K	95%	181M	0s
179150K	95%	196M	0s
179200K	95%	163M	0s
179250K	95%	164M	0s
179300K	95%	74.3M	0s
179350K	95%	39.7M	0s
179400K	95%	39.0M	0s
179450K	95%	38.2M	0s
179500K	95%	33.7M	0s
179550K	95%	39.1M	0s
179600K	95%	37.2M	0s

179650K	95%	56.7M	0s
179700K	95%	33.0M	0s
179750K	95%	39.3M	0s
179800K	95%	38.6M	0s
179850K	95%	38.6M	0s
179900K	95%	34.2M	0s
179950K	95%	37.7M	0s
180000K	95%	38.7M	0s
180050K	95%	39.1M	0s
180100K	95%	33.2M	0s
180150K	95%	39.5M	0s
180200K	95%	39.1M	0s
180250K	95%	39.5M	0s
180300K	95%	63.6M	0s
180350K	95%	152M	0s
180400K	95%	157M	0s
180450K	95%	124M	0s
180500K	95%	104M	0s
180550K	96%	171M	0s
180600K	96%	131M	0s
180650K	96%	161M	0s
180700K	96%	140M	0s
180750K	96%	152M	0s
180800K	96%	123M	0s
180850K	96%	143M	0s
180900K	96%	135M	0s
180950K	96%	143M	0s
181000K	96%	161M	0s
181050K	96%	164M	0s
181100K	96%	134M	0s
181150K	96%	160M	0s
181200K	96%	158M	0s
181250K	96%	143M	0s
181300K	96%	108M	0s
181350K	96%	149M	0s
181400K	96%	162M	0s
181450K	96%	177M	0s
181500K	96%	134M	0s
181550K	96%	157M	0s
181600K	96%	136M	0s
181650K	96%	156M	0s
181700K	96%	139M	0s
181750K	96%	168M	0s
181800K	96%	157M	0s
181850K	96%	149M	0s
181900K	96%	144M	0s
181950K	96%	143M	0s
182000K	96%	151M	0s
182050K	96%	130M	0s
182100K	96%	92.1M	0s
182150K	96%	142M	0s
182200K	96%	141M	0s
182250K	96%	93.1M	0s
182300K	96%	135M	0s
182350K	96%	148M	0s

182400K	97%	129M	0s
182450K	97%	156M	0s
182500K	97%	141M	0s
182550K	97%	181M	0s
182600K	97%	149M	0s
182650K	97%	101M	0s
182700K	97%	161M	0s
182750K	97%	147M	0s
182800K	97%	147M	0s
182850K	97%	136M	0s
182900K	97%	166M	0s
182950K	97%	158M	0s
183000K	97%	173M	0s
183050K	97%	90.5M	0s
183100K	97%	164M	0s
183150K	97%	169M	0s
183200K	97%	155M	0s
183250K	97%	128M	0s
183300K	97%	116M	0s
183350K	97%	135M	0s
183400K	97%	181M	0s
183450K	97%	167M	0s
183500K	97%	190M	0s
183550K	97%	189M	0s
183600K	97%	210M	0s
183650K	97%	147M	0s
183700K	97%	217M	0s
183750K	97%	211M	0s
183800K	97%	213M	0s
183850K	97%	192M	0s
183900K	97%	172M	0s
183950K	97%	193M	0s
184000K	97%	211M	0s
184050K	97%	168M	0s
184100K	97%	211M	0s
184150K	97%	203M	0s
184200K	97%	196M	0s
184250K	97%	165M	0s
184300K	98%	191M	0s
184350K	98%	192M	0s
184400K	98%	187M	0s
184450K	98%	173M	0s
184500K	98%	209M	0s
184550K	98%	198M	0s
184600K	98%	190M	0s
184650K	98%	177M	0s
184700K	98%	196M	0s
184750K	98%	192M	0s
184800K	98%	200M	0s
184850K	98%	167M	0s
184900K	98%	205M	0s
184950K	98%	204M	0s
185000K	98%	88.2M	0s
185050K	98%	36.6M	0s
185100K	98%	44.7M	0s

185150K	98%	42.6M	0s
185200K	98%	40.2M	0s
185250K	98%	35.8M	0s
185300K	98%	41.1M	0s
185350K	98%	41.5M	0s
185400K	98%	21.2M	0s
185450K	98%	40.4M	0s
185500K	98%	45.4M	0s
185550K	98%	46.8M	0s
185600K	98%	44.3M	0s
185650K	98%	47.3M	0s
185700K	98%	49.6M	0s
185750K	98%	165M	0s
185800K	98%	169M	0s
185850K	98%	152M	0s
185900K	98%	140M	0s
185950K	98%	157M	0s
186000K	98%	156M	0s
186050K	98%	193M	0s
186100K	98%	90.7M	0s
186150K	99%	111M	0s
186200K	99%	169M	0s
186250K	99%	165M	0s
186300K	99%	155M	0s
186350K	99%	161M	0s
186400K	99%	134M	0s
186450K	99%	152M	0s
186500K	99%	138M	0s
186550K	99%	151M	0s
186600K	99%	164M	0s
186650K	99%	168M	0s
186700K	99%	154M	0s
186750K	99%	136M	0s
186800K	99%	149M	0s
186850K	99%	141M	0s
186900K	99%	137M	0s
186950K	99%	171M	0s
187000K	99%	151M	0s
187050K	99%	148M	0s
187100K	99%	122M	0s
187150K	99%	156M	0s
187200K	99%	46.0M	0s
187250K	99%	151M	0s
187300K	99%	128M	0s
187350K	99%	128M	0s
187400K	99%	167M	0s
187450K	99%	135M	0s
187500K	99%	51.7M	0s
187550K	99%	31.2M	0s
187600K	99%	39.4M	0s
187650K	99%	68.3M	0s
187700K	99%	132M	0s
187750K	99%	164M	0s
187800K	99%	165M	0s
187850K	99%	120M	0s

```
187900K ..... 99% 137M 0s
187950K ..... 99% 161M 0s
188000K ..... 99% 168M 0s
188050K ..... 100% 100M=1.8s
```

2020-05-30 13:05:31 (101 MB/s) - 'title.akas.tsv.gz' saved [192592731/192592731]

/bin/bash: line 2: fg: no job control

In [8]:

```
title_akas = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.akas.tsv', header=True, inferSchema = True)
title_akas.show(3)
```

```
+-----+-----+-----+-----+-----+-----+
--+-----+
   titleId|ordering|           title|region|language|   types|  attribute
s|isOriginalTitle|
+-----+-----+-----+-----+-----+-----+
--+-----+
tt0000001|      1|      Карменцита|   UA|      \N|imdbDisplay|
\N|      0|
tt0000001|      2|      Carmencita|   DE|      \N|      \N|literal titl
e|      0|
tt0000001|      3|Carmencita - span...|  HU|      \N|imdbDisplay|
\N|      0|
+-----+-----+-----+-----+-----+-----+
--+-----+
only showing top 3 rows
```

In [9]:

```
%sh wget https://datasets.imdbws.com/title.crew.tsv.gz  
  
%sh  
gunzip title.crew.tsv.gz
```

```
--2020-05-30 13:06:31-- https://datasets.imdbws.com/title.crew.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.37, 13.224.13.5
4, 13.224.13.26, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.37|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 48399360 (46M) [binary/octet-stream]
Saving to: 'title.crew.tsv.gz'
```

0K	0%	3.60M	13s
50K	0%	7.16M	10s
100K	0%	13.4M	8s
150K	0%	11.5M	7s
200K	0%	16.7M	6s
250K	0%	24.9M	5s
300K	0%	19.4M	5s
350K	0%	23.4M	4s
400K	0%	38.7M	4s
450K	1%	26.5M	4s
500K	1%	49.6M	4s
550K	1%	32.2M	3s
600K	1%	58.2M	3s
650K	1%	52.8M	3s
700K	1%	45.7M	3s
750K	1%	48.1M	3s
800K	1%	49.3M	3s
850K	1%	80.3M	3s
900K	2%	62.8M	2s
950K	2%	48.2M	2s
1000K	2%	129M	2s
1050K	2%	65.6M	2s
1100K	2%	83.8M	2s
1150K	2%	58.9M	2s
1200K	2%	98.5M	2s
1250K	2%	123M	2s
1300K	2%	98.1M	2s
1350K	2%	69.2M	2s
1400K	3%	96.3M	2s
1450K	3%	150M	2s
1500K	3%	101M	2s
1550K	3%	110M	2s
1600K	3%	100M	2s
1650K	3%	104M	2s
1700K	3%	156M	2s
1750K	3%	97.8M	1s
1800K	3%	104M	1s
1850K	4%	120M	1s
1900K	4%	169M	1s
1950K	4%	225M	1s
2000K	4%	174M	1s
2050K	4%	113M	1s
2100K	4%	133M	1s
2150K	4%	126M	1s
2200K	4%	132M	1s

2250K	4%	129M	1s
2300K	4%	137M	1s
2350K	5%	116M	1s
2400K	5%	130M	1s
2450K	5%	148M	1s
2500K	5%	131M	1s
2550K	5%	129M	1s
2600K	5%	145M	1s
2650K	5%	141M	1s
2700K	5%	132M	1s
2750K	5%	124M	1s
2800K	6%	135M	1s
2850K	6%	142M	1s
2900K	6%	163M	1s
2950K	6%	200M	1s
3000K	6%	211M	1s
3050K	6%	208M	1s
3100K	6%	271M	1s
3150K	6%	171M	1s
3200K	6%	267M	1s
3250K	6%	202M	1s
3300K	7%	273M	1s
3350K	7%	251M	1s
3400K	7%	219M	1s
3450K	7%	99.5M	1s
3500K	7%	191M	1s
3550K	7%	97.3M	1s
3600K	7%	101M	1s
3650K	7%	133M	1s
3700K	7%	88.8M	1s
3750K	8%	82.0M	1s
3800K	8%	71.3M	1s
3850K	8%	63.6M	1s
3900K	8%	74.7M	1s
3950K	8%	29.9M	1s
4000K	8%	174M	1s
4050K	8%	118M	1s
4100K	8%	121M	1s
4150K	8%	106M	1s
4200K	8%	190M	1s
4250K	9%	80.9M	1s
4300K	9%	57.8M	1s
4350K	9%	101M	1s
4400K	9%	110M	1s
4450K	9%	87.8M	1s
4500K	9%	69.0M	1s
4550K	9%	93.7M	1s
4600K	9%	105M	1s
4650K	9%	78.8M	1s
4700K	10%	78.2M	1s
4750K	10%	74.4M	1s
4800K	10%	81.3M	1s
4850K	10%	94.1M	1s
4900K	10%	109M	1s
4950K	10%	102M	1s

5000K	10%	106M	1s
5050K	10%	112M	1s
5100K	10%	118M	1s
5150K	11%	127M	1s
5200K	11%	111M	1s
5250K	11%	115M	1s
5300K	11%	105M	1s
5350K	11%	73.8M	1s
5400K	11%	100M	1s
5450K	11%	59.3M	1s
5500K	11%	115M	1s
5550K	11%	65.2M	1s
5600K	11%	157M	1s
5650K	12%	94.3M	1s
5700K	12%	106M	1s
5750K	12%	139M	1s
5800K	12%	136M	1s
5850K	12%	154M	1s
5900K	12%	157M	1s
5950K	12%	126M	1s
6000K	12%	113M	1s
6050K	12%	116M	1s
6100K	13%	103M	1s
6150K	13%	119M	1s
6200K	13%	120M	1s
6250K	13%	84.3M	1s
6300K	13%	153M	1s
6350K	13%	87.2M	1s
6400K	13%	144M	1s
6450K	13%	107M	1s
6500K	13%	146M	1s
6550K	13%	114M	1s
6600K	14%	158M	1s
6650K	14%	142M	1s
6700K	14%	152M	1s
6750K	14%	136M	1s
6800K	14%	126M	1s
6850K	14%	104M	1s
6900K	14%	95.0M	1s
6950K	14%	147M	1s
7000K	14%	150M	1s
7050K	15%	160M	1s
7100K	15%	146M	1s
7150K	15%	150M	1s
7200K	15%	175M	1s
7250K	15%	34.6M	1s
7300K	15%	108M	1s
7350K	15%	103M	1s
7400K	15%	137M	1s
7450K	15%	102M	1s
7500K	15%	152M	1s
7550K	16%	128M	1s
7600K	16%	83.7M	1s
7650K	16%	80.5M	1s
7700K	16%	154M	1s

7750K	16%	138M	1s
7800K	16%	142M	1s
7850K	16%	164M	1s
7900K	16%	155M	1s
7950K	16%	77.9M	1s
8000K	17%	107M	1s
8050K	17%	101M	1s
8100K	17%	118M	1s
8150K	17%	79.8M	1s
8200K	17%	108M	1s
8250K	17%	106M	1s
8300K	17%	109M	1s
8350K	17%	69.7M	1s
8400K	17%	46.2M	1s
8450K	17%	96.1M	1s
8500K	18%	153M	1s
8550K	18%	111M	1s
8600K	18%	108M	1s
8650K	18%	76.9M	1s
8700K	18%	167M	1s
8750K	18%	138M	1s
8800K	18%	139M	1s
8850K	18%	149M	1s
8900K	18%	147M	1s
8950K	19%	77.5M	1s
9000K	19%	113M	1s
9050K	19%	108M	1s
9100K	19%	122M	1s
9150K	19%	75.6M	1s
9200K	19%	93.3M	1s
9250K	19%	166M	1s
9300K	19%	105M	1s
9350K	19%	121M	1s
9400K	19%	96.2M	1s
9450K	20%	144M	1s
9500K	20%	141M	1s
9550K	20%	130M	1s
9600K	20%	140M	0s
9650K	20%	139M	0s
9700K	20%	100M	0s
9750K	20%	91.9M	0s
9800K	20%	108M	0s
9850K	20%	156M	0s
9900K	21%	141M	0s
9950K	21%	138M	0s
10000K	21%	138M	0s
10050K	21%	148M	0s
10100K	21%	88.5M	0s
10150K	21%	96.8M	0s
10200K	21%	122M	0s
10250K	21%	119M	0s
10300K	21%	108M	0s
10350K	22%	84.3M	0s
10400K	22%	133M	0s
10450K	22%	83.2M	0s

10500K	22%	174M	0s
10550K	22%	93.7M	0s
10600K	22%	152M	0s
10650K	22%	158M	0s
10700K	22%	138M	0s
10750K	22%	129M	0s
10800K	22%	162M	0s
10850K	23%	164M	0s
10900K	23%	85.7M	0s
10950K	23%	93.1M	0s
11000K	23%	98.0M	0s
11050K	23%	84.4M	0s
11100K	23%	120M	0s
11150K	23%	139M	0s
11200K	23%	85.1M	0s
11250K	23%	111M	0s
11300K	24%	64.6M	0s
11350K	24%	103M	0s
11400K	24%	88.3M	0s
11450K	24%	96.9M	0s
11500K	24%	159M	0s
11550K	24%	82.1M	0s
11600K	24%	76.8M	0s
11650K	24%	141M	0s
11700K	24%	159M	0s
11750K	24%	135M	0s
11800K	25%	152M	0s
11850K	25%	150M	0s
11900K	25%	163M	0s
11950K	25%	65.5M	0s
12000K	25%	89.8M	0s
12050K	25%	109M	0s
12100K	25%	86.4M	0s
12150K	25%	127M	0s
12200K	25%	141M	0s
12250K	26%	92.2M	0s
12300K	26%	162M	0s
12350K	26%	111M	0s
12400K	26%	124M	0s
12450K	26%	142M	0s
12500K	26%	105M	0s
12550K	26%	105M	0s
12600K	26%	146M	0s
12650K	26%	106M	0s
12700K	26%	155M	0s
12750K	27%	72.3M	0s
12800K	27%	147M	0s
12850K	27%	154M	0s
12900K	27%	138M	0s
12950K	27%	136M	0s
13000K	27%	157M	0s
13050K	27%	88.4M	0s
13100K	27%	112M	0s
13150K	27%	101M	0s
13200K	28%	111M	0s

13250K	28%	73.0M	0s
13300K	28%	137M	0s
13350K	28%	142M	0s
13400K	28%	73.9M	0s
13450K	28%	156M	0s
13500K	28%	128M	0s
13550K	28%	71.8M	0s
13600K	28%	77.0M	0s
13650K	28%	116M	0s
13700K	29%	89.4M	0s
13750K	29%	123M	0s
13800K	29%	84.8M	0s
13850K	29%	123M	0s
13900K	29%	147M	0s
13950K	29%	140M	0s
14000K	29%	146M	0s
14050K	29%	134M	0s
14100K	29%	98.3M	0s
14150K	30%	82.8M	0s
14200K	30%	85.3M	0s
14250K	30%	95.6M	0s
14300K	30%	109M	0s
14350K	30%	174M	0s
14400K	30%	194M	0s
14450K	30%	92.8M	0s
14500K	30%	209M	0s
14550K	30%	189M	0s
14600K	30%	209M	0s
14650K	31%	87.2M	0s
14700K	31%	124M	0s
14750K	31%	137M	0s
14800K	31%	177M	0s
14850K	31%	120M	0s
14900K	31%	131M	0s
14950K	31%	116M	0s
15000K	31%	108M	0s
15050K	31%	113M	0s
15100K	32%	143M	0s
15150K	32%	139M	0s
15200K	32%	154M	0s
15250K	32%	148M	0s
15300K	32%	133M	0s
15350K	32%	79.9M	0s
15400K	32%	87.5M	0s
15450K	32%	89.3M	0s
15500K	32%	106M	0s
15550K	33%	137M	0s
15600K	33%	150M	0s
15650K	33%	155M	0s
15700K	33%	74.0M	0s
15750K	33%	135M	0s
15800K	33%	101M	0s
15850K	33%	104M	0s
15900K	33%	77.9M	0s
15950K	33%	140M	0s

16000K	33%	151M	0s
16050K	34%	96.0M	0s
16100K	34%	119M	0s
16150K	34%	72.7M	0s

*** WARNING: skipped 22420 bytes of output ***

30950K	65%	87.1M	0s
31000K	65%	81.0M	0s
31050K	65%	91.5M	0s
31100K	65%	159M	0s
31150K	66%	103M	0s
31200K	66%	131M	0s
31250K	66%	150M	0s
31300K	66%	158M	0s
31350K	66%	141M	0s
31400K	66%	155M	0s
31450K	66%	109M	0s
31500K	66%	89.1M	0s
31550K	66%	89.5M	0s
31600K	66%	166M	0s
31650K	67%	155M	0s
31700K	67%	153M	0s
31750K	67%	138M	0s
31800K	67%	63.0M	0s
31850K	67%	98.3M	0s
31900K	67%	152M	0s
31950K	67%	125M	0s
32000K	67%	165M	0s
32050K	67%	154M	0s
32100K	68%	72.7M	0s
32150K	68%	74.8M	0s
32200K	68%	79.2M	0s
32250K	68%	90.3M	0s
32300K	68%	117M	0s
32350K	68%	124M	0s
32400K	68%	167M	0s
32450K	68%	159M	0s
32500K	68%	164M	0s
32550K	68%	92.8M	0s
32600K	69%	112M	0s
32650K	69%	89.2M	0s
32700K	69%	107M	0s
32750K	69%	102M	0s
32800K	69%	157M	0s
32850K	69%	88.4M	0s
32900K	69%	109M	0s
32950K	69%	88.8M	0s
33000K	69%	154M	0s
33050K	70%	155M	0s
33100K	70%	150M	0s
33150K	70%	138M	0s
33200K	70%	69.2M	0s
33250K	70%	106M	0s
33300K	70%	79.8M	0s

33350K	70%	135M	0s
33400K	70%	164M	0s
33450K	70%	155M	0s
33500K	70%	140M	0s
33550K	71%	139M	0s
33600K	71%	148M	0s
33650K	71%	153M	0s
33700K	71%	138M	0s
33750K	71%	74.3M	0s
33800K	71%	109M	0s
33850K	71%	74.2M	0s
33900K	71%	86.3M	0s
33950K	71%	75.2M	0s
34000K	72%	105M	0s
34050K	72%	168M	0s
34100K	72%	149M	0s
34150K	72%	144M	0s
34200K	72%	154M	0s
34250K	72%	154M	0s
34300K	72%	86.3M	0s
34350K	72%	68.9M	0s
34400K	72%	112M	0s
34450K	72%	140M	0s
34500K	73%	166M	0s
34550K	73%	150M	0s
34600K	73%	159M	0s
34650K	73%	148M	0s
34700K	73%	157M	0s
34750K	73%	139M	0s
34800K	73%	148M	0s
34850K	73%	81.9M	0s
34900K	73%	114M	0s
34950K	74%	73.8M	0s
35000K	74%	83.7M	0s
35050K	74%	70.3M	0s
35100K	74%	94.2M	0s
35150K	74%	123M	0s
35200K	74%	163M	0s
35250K	74%	163M	0s
35300K	74%	140M	0s
35350K	74%	125M	0s
35400K	75%	79.6M	0s
35450K	75%	98.7M	0s
35500K	75%	77.5M	0s
35550K	75%	81.1M	0s
35600K	75%	160M	0s
35650K	75%	131M	0s
35700K	75%	161M	0s
35750K	75%	143M	0s
35800K	75%	147M	0s
35850K	75%	148M	0s
35900K	76%	152M	0s
35950K	76%	45.7M	0s
36000K	76%	75.3M	0s
36050K	76%	86.0M	0s

36100K	76%	108M	0s
36150K	76%	83.6M	0s
36200K	76%	152M	0s
36250K	76%	138M	0s
36300K	76%	144M	0s
36350K	77%	138M	0s
36400K	77%	142M	0s
36450K	77%	85.1M	0s
36500K	77%	102M	0s
36550K	77%	60.6M	0s
36600K	77%	107M	0s
36650K	77%	149M	0s
36700K	77%	150M	0s
36750K	77%	122M	0s
36800K	77%	129M	0s
36850K	78%	146M	0s
36900K	78%	160M	0s
36950K	78%	81.0M	0s
37000K	78%	82.8M	0s
37050K	78%	109M	0s
37100K	78%	110M	0s
37150K	78%	79.8M	0s
37200K	78%	122M	0s
37250K	78%	149M	0s
37300K	79%	162M	0s
37350K	79%	137M	0s
37400K	79%	164M	0s
37450K	79%	162M	0s
37500K	79%	148M	0s
37550K	79%	138M	0s
37600K	79%	80.1M	0s
37650K	79%	88.3M	0s
37700K	79%	102M	0s
37750K	79%	84.5M	0s
37800K	80%	154M	0s
37850K	80%	144M	0s
37900K	80%	162M	0s
37950K	80%	139M	0s
38000K	80%	124M	0s
38050K	80%	169M	0s
38100K	80%	149M	0s
38150K	80%	65.2M	0s
38200K	80%	85.7M	0s
38250K	81%	73.0M	0s
38300K	81%	108M	0s
38350K	81%	93.7M	0s
38400K	81%	165M	0s
38450K	81%	166M	0s
38500K	81%	132M	0s
38550K	81%	138M	0s
38600K	81%	160M	0s
38650K	81%	143M	0s
38700K	81%	136M	0s
38750K	82%	66.2M	0s
38800K	82%	100M	0s

38850K	82%	101M	0s
38900K	82%	114M	0s
38950K	82%	83.9M	0s
39000K	82%	106M	0s
39050K	82%	148M	0s
39100K	82%	155M	0s
39150K	82%	118M	0s
39200K	83%	91.7M	0s
39250K	83%	76.4M	0s
39300K	83%	109M	0s
39350K	83%	80.8M	0s
39400K	83%	99.4M	0s
39450K	83%	96.6M	0s
39500K	83%	161M	0s
39550K	83%	140M	0s
39600K	83%	143M	0s
39650K	83%	152M	0s
39700K	84%	165M	0s
39750K	84%	141M	0s
39800K	84%	87.3M	0s
39850K	84%	119M	0s
39900K	84%	114M	0s
39950K	84%	101M	0s
40000K	84%	168M	0s
40050K	84%	132M	0s
40100K	84%	122M	0s
40150K	85%	158M	0s
40200K	85%	180M	0s
40250K	85%	129M	0s
40300K	85%	151M	0s
40350K	85%	132M	0s
40400K	85%	113M	0s
40450K	85%	130M	0s
40500K	85%	146M	0s
40550K	85%	110M	0s
40600K	86%	146M	0s
40650K	86%	147M	0s
40700K	86%	123M	0s
40750K	86%	129M	0s
40800K	86%	155M	0s
40850K	86%	131M	0s
40900K	86%	152M	0s
40950K	86%	143M	0s
41000K	86%	127M	0s
41050K	86%	155M	0s
41100K	87%	145M	0s
41150K	87%	112M	0s
41200K	87%	147M	0s
41250K	87%	151M	0s
41300K	87%	156M	0s
41350K	87%	115M	0s
41400K	87%	146M	0s
41450K	87%	150M	0s
41500K	87%	147M	0s
41550K	88%	121M	0s

41600K	88%	155M	0s
41650K	88%	138M	0s
41700K	88%	156M	0s
41750K	88%	164M	0s
41800K	88%	146M	0s
41850K	88%	141M	0s
41900K	88%	149M	0s
41950K	88%	131M	0s
42000K	88%	141M	0s
42050K	89%	150M	0s
42100K	89%	153M	0s
42150K	89%	122M	0s
42200K	89%	154M	0s
42250K	89%	152M	0s
42300K	89%	144M	0s
42350K	89%	133M	0s
42400K	89%	152M	0s
42450K	89%	158M	0s
42500K	90%	155M	0s
42550K	90%	137M	0s
42600K	90%	148M	0s
42650K	90%	152M	0s
42700K	90%	155M	0s
42750K	90%	119M	0s
42800K	90%	147M	0s
42850K	90%	122M	0s
42900K	90%	150M	0s
42950K	90%	135M	0s
43000K	91%	148M	0s
43050K	91%	149M	0s
43100K	91%	156M	0s
43150K	91%	134M	0s
43200K	91%	150M	0s
43250K	91%	156M	0s
43300K	91%	155M	0s
43350K	91%	131M	0s
43400K	91%	161M	0s
43450K	92%	149M	0s
43500K	92%	155M	0s
43550K	92%	129M	0s
43600K	92%	157M	0s
43650K	92%	153M	0s
43700K	92%	150M	0s
43750K	92%	145M	0s
43800K	92%	152M	0s
43850K	92%	150M	0s
43900K	92%	153M	0s
43950K	93%	121M	0s
44000K	93%	157M	0s
44050K	93%	152M	0s
44100K	93%	150M	0s
44150K	93%	137M	0s
44200K	93%	152M	0s
44250K	93%	164M	0s
44300K	93%	144M	0s

44350K	93%	98.7M	0s
44400K	94%	154M	0s
44450K	94%	38.2M	0s
44500K	94%	177M	0s
44550K	94%	165M	0s
44600K	94%	133M	0s
44650K	94%	130M	0s
44700K	94%	143M	0s
44750K	94%	142M	0s
44800K	94%	230M	0s
44850K	94%	213M	0s
44900K	95%	192M	0s
44950K	95%	206M	0s
45000K	95%	215M	0s
45050K	95%	121M	0s
45100K	95%	178M	0s
45150K	95%	231M	0s
45200K	95%	265M	0s
45250K	95%	286M	0s
45300K	95%	278M	0s
45350K	96%	278M	0s
45400K	96%	300M	0s
45450K	96%	316M	0s
45500K	96%	237M	0s
45550K	96%	146M	0s
45600K	96%	295M	0s
45650K	96%	190M	0s
45700K	96%	216M	0s
45750K	96%	207M	0s
45800K	97%	197M	0s
45850K	97%	185M	0s
45900K	97%	305M	0s
45950K	97%	101M	0s
46000K	97%	142M	0s
46050K	97%	174M	0s
46100K	97%	219M	0s
46150K	97%	214M	0s
46200K	97%	239M	0s
46250K	97%	193M	0s
46300K	98%	299M	0s
46350K	98%	252M	0s
46400K	98%	309M	0s
46450K	98%	290M	0s
46500K	98%	206M	0s
46550K	98%	234M	0s
46600K	98%	215M	0s
46650K	98%	224M	0s
46700K	98%	303M	0s
46750K	99%	175M	0s
46800K	99%	184M	0s
46850K	99%	184M	0s
46900K	99%	200M	0s
46950K	99%	180M	0s
47000K	99%	202M	0s
47050K	99%	204M	0s

```
47100K ..... 99% 199M 0s
47150K ..... 99% 157M 0s
47200K ..... 99% 187M 0s
47250K ..... 100% 164M=0.5s
```

2020-05-30 13:06:32 (101 MB/s) - 'title.crew.tsv.gz' saved [48399360/48399360]

/bin/bash: line 2: fg: no job control

In [10]:

```
title_crew = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.crew.ts
v', header=True, inferSchema = True)
title_crew.show(3)
```

```
+-----+-----+-----+
  tconst|directors|writers|
+-----+-----+-----+
tt0000001|nm0005690|    \N|
tt0000002|nm0721526|    \N|
tt0000003|nm0721526|    \N|
+-----+-----+-----+
only showing top 3 rows
```

In [11]:

```
%sh wget https://datasets.imdbws.com/title.episode.tsv.gz  
  
%sh  
gunzip title.episode.tsv.gz
```

```
--2020-05-30 13:06:45-- https://datasets.imdbws.com/title.episode.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.32, 13.224.13.3
7, 13.224.13.54, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.32|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 26570485 (25M) [binary/octet-stream]
Saving to: 'title.episode.tsv.gz'
```

0K	0%	4.76M	5s
50K	0%	8.62M	4s
100K	0%	14.0M	3s
150K	0%	19.6M	3s
200K	0%	16.1M	3s
250K	1%	21.7M	2s
300K	1%	39.0M	2s
350K	1%	27.9M	2s
400K	1%	47.1M	2s
450K	1%	31.6M	2s
500K	2%	59.9M	2s
550K	2%	57.2M	1s
600K	2%	52.3M	1s
650K	2%	52.7M	1s
700K	2%	53.3M	1s
750K	3%	86.6M	1s
800K	3%	79.0M	1s
850K	3%	66.5M	1s
900K	3%	93.0M	1s
950K	3%	184M	1s
1000K	4%	66.0M	1s
1050K	4%	59.8M	1s
1100K	4%	108M	1s
1150K	4%	80.2M	1s
1200K	4%	117M	1s
1250K	5%	139M	1s
1300K	5%	113M	1s
1350K	5%	116M	1s
1400K	5%	121M	1s
1450K	5%	81.0M	1s
1500K	5%	105M	1s
1550K	6%	115M	1s
1600K	6%	121M	1s
1650K	6%	146M	1s
1700K	6%	106M	1s
1750K	6%	152M	1s
1800K	7%	61.3M	1s
1850K	7%	57.1M	1s
1900K	7%	86.3M	1s
1950K	7%	124M	1s
2000K	7%	185M	1s
2050K	8%	175M	1s
2100K	8%	162M	1s
2150K	8%	166M	1s
2200K	8%	177M	1s

2250K	8%	115M	1s
2300K	9%	181M	1s
2350K	9%	123M	1s
2400K	9%	138M	1s
2450K	9%	104M	1s
2500K	9%	122M	0s
2550K	10%	124M	0s
2600K	10%	272M	0s
2650K	10%	143M	0s
2700K	10%	302M	0s
2750K	10%	314M	0s
2800K	10%	310M	0s
2850K	11%	257M	0s
2900K	11%	280M	0s
2950K	11%	163M	0s
3000K	11%	287M	0s
3050K	11%	253M	0s
3100K	12%	288M	0s
3150K	12%	46.6M	0s
3200K	12%	67.5M	0s
3250K	12%	144M	0s
3300K	12%	50.9M	0s
3350K	13%	92.6M	0s
3400K	13%	68.2M	0s
3450K	13%	49.7M	0s
3500K	13%	98.0M	0s
3550K	13%	116M	0s
3600K	14%	160M	0s
3650K	14%	142M	0s
3700K	14%	103M	0s
3750K	14%	157M	0s
3800K	14%	86.6M	0s
3850K	15%	101M	0s
3900K	15%	293M	0s
3950K	15%	119M	0s
4000K	15%	263M	0s
4050K	15%	249M	0s
4100K	15%	306M	0s
4150K	16%	87.9M	0s
4200K	16%	128M	0s
4250K	16%	115M	0s
4300K	16%	277M	0s
4350K	16%	118M	0s
4400K	17%	116M	0s
4450K	17%	83.7M	0s
4500K	17%	111M	0s
4550K	17%	151M	0s
4600K	17%	159M	0s
4650K	18%	138M	0s
4700K	18%	159M	0s
4750K	18%	68.5M	0s
4800K	18%	102M	0s
4850K	18%	146M	0s
4900K	19%	150M	0s
4950K	19%	96.3M	0s

5000K	19%	159M	0s
5050K	19%	101M	0s
5100K	19%	104M	0s
5150K	20%	150M	0s
5200K	20%	158M	0s
5250K	20%	145M	0s
5300K	20%	110M	0s
5350K	20%	269M	0s
5400K	21%	299M	0s
5450K	21%	255M	0s
5500K	21%	241M	0s
5550K	21%	108M	0s
5600K	21%	156M	0s
5650K	21%	153M	0s
5700K	22%	278M	0s
5750K	22%	303M	0s
5800K	22%	274M	0s
5850K	22%	222M	0s
5900K	22%	304M	0s
5950K	23%	105M	0s
6000K	23%	77.6M	0s
6050K	23%	292M	0s
6100K	23%	286M	0s
6150K	23%	290M	0s
6200K	24%	287M	0s
6250K	24%	70.5M	0s
6300K	24%	127M	0s
6350K	24%	134M	0s
6400K	24%	230M	0s
6450K	25%	252M	0s
6500K	25%	284M	0s
6550K	25%	302M	0s
6600K	25%	268M	0s
6650K	25%	235M	0s
6700K	26%	279M	0s
6750K	26%	115M	0s
6800K	26%	302M	0s
6850K	26%	237M	0s
6900K	26%	293M	0s
6950K	26%	306M	0s
7000K	27%	311M	0s
7050K	27%	204M	0s
7100K	27%	259M	0s
7150K	27%	281M	0s
7200K	27%	273M	0s
7250K	28%	274M	0s
7300K	28%	310M	0s
7350K	28%	285M	0s
7400K	28%	276M	0s
7450K	28%	234M	0s
7500K	29%	267M	0s
7550K	29%	316M	0s
7600K	29%	34.2M	0s
7650K	29%	240M	0s
7700K	29%	306M	0s

7750K	30%	110M	0s
7800K	30%	34.9M	0s
7850K	30%	133M	0s
7900K	30%	162M	0s
7950K	30%	96.1M	0s
8000K	31%	119M	0s
8050K	31%	164M	0s
8100K	31%	304M	0s
8150K	31%	312M	0s
8200K	31%	267M	0s
8250K	31%	197M	0s
8300K	32%	115M	0s
8350K	32%	206M	0s
8400K	32%	208M	0s
8450K	32%	177M	0s
8500K	32%	56.5M	0s
8550K	33%	83.0M	0s
8600K	33%	94.8M	0s
8650K	33%	65.0M	0s
8700K	33%	91.3M	0s
8750K	33%	181M	0s
8800K	34%	173M	0s
8850K	34%	124M	0s
8900K	34%	153M	0s
8950K	34%	284M	0s
9000K	34%	291M	0s
9050K	35%	55.2M	0s
9100K	35%	36.4M	0s
9150K	35%	25.0M	0s
9200K	35%	16.2M	0s
9250K	35%	15.1M	0s
9300K	36%	30.9M	0s
9350K	36%	73.3M	0s
9400K	36%	154M	0s
9450K	36%	164M	0s
9500K	36%	206M	0s
9550K	36%	154M	0s
9600K	37%	179M	0s
9650K	37%	168M	0s
9700K	37%	201M	0s
9750K	37%	201M	0s
9800K	37%	203M	0s
9850K	38%	182M	0s
9900K	38%	197M	0s
9950K	38%	201M	0s
10000K	38%	204M	0s
10050K	38%	171M	0s
10100K	39%	203M	0s
10150K	39%	207M	0s
10200K	39%	77.5M	0s
10250K	39%	183M	0s
10300K	39%	21.1M	0s
10350K	40%	40.1M	0s
10400K	40%	35.2M	0s
10450K	40%	35.5M	0s

10500K	40%	22.0M	0s
10550K	40%	19.6M	0s
10600K	41%	19.1M	0s
10650K	41%	26.2M	0s
10700K	41%	33.7M	0s
10750K	41%	44.6M	0s
10800K	41%	153M	0s
10850K	42%	115M	0s
10900K	42%	209M	0s
10950K	42%	210M	0s
11000K	42%	206M	0s
11050K	42%	175M	0s
11100K	42%	109M	0s
11150K	43%	155M	0s
11200K	43%	49.4M	0s
11250K	43%	120M	0s
11300K	43%	126M	0s
11350K	43%	80.9M	0s
11400K	44%	76.9M	0s
11450K	44%	85.2M	0s
11500K	44%	68.1M	0s
11550K	44%	89.5M	0s
11600K	44%	108M	0s
11650K	45%	46.4M	0s
11700K	45%	220M	0s
11750K	45%	21.6M	0s
11800K	45%	36.2M	0s
11850K	45%	67.9M	0s
11900K	46%	33.2M	0s
11950K	46%	119M	0s
12000K	46%	158M	0s
12050K	46%	133M	0s
12100K	46%	151M	0s
12150K	47%	144M	0s
12200K	47%	158M	0s
12250K	47%	137M	0s
12300K	47%	162M	0s
12350K	47%	188M	0s
12400K	47%	305M	0s
12450K	48%	169M	0s
12500K	48%	200M	0s
12550K	48%	290M	0s
12600K	48%	279M	0s
12650K	48%	258M	0s
12700K	49%	318M	0s
12750K	49%	315M	0s
12800K	49%	322M	0s
12850K	49%	245M	0s
12900K	49%	281M	0s
12950K	50%	285M	0s
13000K	50%	311M	0s
13050K	50%	290M	0s
13100K	50%	322M	0s
13150K	50%	313M	0s
13200K	51%	292M	0s

13250K	51%	237M	0s
13300K	51%	289M	0s
13350K	51%	319M	0s
13400K	51%	309M	0s
13450K	52%	291M	0s
13500K	52%	315M	0s
13550K	52%	293M	0s
13600K	52%	244M	0s
13650K	52%	114M	0s
13700K	52%	150M	0s
13750K	53%	158M	0s
13800K	53%	220M	0s
13850K	53%	246M	0s
13900K	53%	307M	0s
13950K	53%	321M	0s
14000K	54%	286M	0s
14050K	54%	231M	0s
14100K	54%	281M	0s
14150K	54%	303M	0s
14200K	54%	323M	0s
14250K	55%	283M	0s
14300K	55%	320M	0s
14350K	55%	292M	0s
14400K	55%	275M	0s
14450K	55%	238M	0s
14500K	56%	307M	0s
14550K	56%	320M	0s
14600K	56%	74.9M	0s
14650K	56%	139M	0s
14700K	56%	159M	0s
14750K	57%	166M	0s
14800K	57%	142M	0s
14850K	57%	137M	0s
14900K	57%	153M	0s
14950K	57%	166M	0s
15000K	58%	158M	0s
15050K	58%	139M	0s
15100K	58%	165M	0s
15150K	58%	81.5M	0s
15200K	58%	146M	0s
15250K	58%	98.6M	0s
15300K	59%	151M	0s
15350K	59%	167M	0s
15400K	59%	127M	0s
15450K	59%	121M	0s
15500K	59%	152M	0s
15550K	60%	132M	0s
15600K	60%	150M	0s
15650K	60%	133M	0s
15700K	60%	140M	0s
15750K	60%	143M	0s
15800K	61%	165M	0s
15850K	61%	149M	0s
15900K	61%	143M	0s
15950K	61%	160M	0s

16000K	61%	160M	0s
16050K	62%	129M	0s
16100K	62%	160M	0s
16150K	62%	44.3M	0s
16200K	62%	95.9M	0s
16250K	62%	66.5M	0s
16300K	63%	99.9M	0s
16350K	63%	86.3M	0s
16400K	63%	101M	0s
16450K	63%	72.9M	0s
16500K	63%	191M	0s
16550K	63%	159M	0s
16600K	64%	89.3M	0s
16650K	64%	91.1M	0s
16700K	64%	108M	0s
16750K	64%	168M	0s
16800K	64%	158M	0s
16850K	65%	117M	0s
16900K	65%	152M	0s
16950K	65%	167M	0s
17000K	65%	80.3M	0s
17050K	65%	95.2M	0s
17100K	66%	98.4M	0s
17150K	66%	93.0M	0s
17200K	66%	68.2M	0s
17250K	66%	80.1M	0s
17300K	66%	94.9M	0s
17350K	67%	70.8M	0s
17400K	67%	89.9M	0s
17450K	67%	93.2M	0s
17500K	67%	136M	0s
17550K	67%	140M	0s
17600K	68%	159M	0s
17650K	68%	145M	0s
17700K	68%	21.2M	0s
17750K	68%	40.1M	0s
17800K	68%	137M	0s
17850K	68%	43.8M	0s
17900K	69%	156M	0s
17950K	69%	156M	0s
18000K	69%	117M	0s
18050K	69%	125M	0s
18100K	69%	161M	0s
18150K	70%	169M	0s
18200K	70%	50.4M	0s
18250K	70%	76.4M	0s
18300K	70%	87.9M	0s
18350K	70%	138M	0s
18400K	71%	152M	0s
18450K	71%	138M	0s
18500K	71%	152M	0s
18550K	71%	151M	0s
18600K	71%	167M	0s
18650K	72%	66.9M	0s
18700K	72%	73.1M	0s

18750K	72%	139M	0s
18800K	72%	151M	0s
18850K	72%	35.4M	0s
18900K	73%	22.2M	0s
18950K	73%	76.0M	0s
19000K	73%	156M	0s
19050K	73%	149M	0s
19100K	73%	145M	0s
19150K	73%	162M	0s
19200K	74%	155M	0s
19250K	74%	132M	0s
19300K	74%	159M	0s
19350K	74%	148M	0s
19400K	74%	111M	0s
19450K	75%	89.7M	0s
19500K	75%	63.2M	0s
19550K	75%	52.9M	0s
19600K	75%	59.8M	0s
19650K	75%	127M	0s
19700K	76%	159M	0s
19750K	76%	143M	0s
19800K	76%	152M	0s
19850K	76%	88.6M	0s
19900K	76%	145M	0s
19950K	77%	149M	0s
20000K	77%	134M	0s
20050K	77%	132M	0s
20100K	77%	143M	0s
20150K	77%	147M	0s
20200K	78%	147M	0s
20250K	78%	89.1M	0s
20300K	78%	69.0M	0s
20350K	78%	74.4M	0s
20400K	78%	127M	0s
20450K	79%	120M	0s
20500K	79%	127M	0s
20550K	79%	165M	0s
20600K	79%	150M	0s
20650K	79%	126M	0s
20700K	79%	162M	0s
20750K	80%	144M	0s
20800K	80%	74.4M	0s
20850K	80%	91.6M	0s
20900K	80%	161M	0s
20950K	80%	166M	0s
21000K	81%	164M	0s
21050K	81%	152M	0s
21100K	81%	136M	0s
21150K	81%	124M	0s
21200K	81%	93.3M	0s
21250K	82%	68.5M	0s
21300K	82%	109M	0s
21350K	82%	83.8M	0s
21400K	82%	165M	0s
21450K	82%	154M	0s

21500K	83%	171M	0s
21550K	83%	166M	0s
21600K	83%	169M	0s
21650K	83%	121M	0s
21700K	83%	150M	0s
21750K	84%	91.3M	0s
21800K	84%	98.1M	0s
21850K	84%	85.2M	0s
21900K	84%	37.1M	0s
21950K	84%	20.1M	0s
22000K	84%	24.4M	0s
22050K	85%	32.9M	0s
22100K	85%	23.5M	0s
22150K	85%	136M	0s
22200K	85%	144M	0s
22250K	85%	131M	0s
22300K	86%	158M	0s
22350K	86%	140M	0s
22400K	86%	154M	0s
22450K	86%	133M	0s
22500K	86%	135M	0s
22550K	87%	164M	0s
22600K	87%	148M	0s
22650K	87%	132M	0s
22700K	87%	160M	0s
22750K	87%	158M	0s
22800K	88%	138M	0s
22850K	88%	122M	0s
22900K	88%	115M	0s
22950K	88%	140M	0s
23000K	88%	144M	0s
23050K	89%	108M	0s
23100K	89%	142M	0s
23150K	89%	160M	0s
23200K	89%	156M	0s
23250K	89%	260M	0s
23300K	89%	283M	0s
23350K	90%	283M	0s
23400K	90%	317M	0s
23450K	90%	259M	0s
23500K	90%	287M	0s
23550K	90%	313M	0s
23600K	91%	294M	0s
23650K	91%	251M	0s
23700K	91%	290M	0s
23750K	91%	291M	0s
23800K	91%	315M	0s
23850K	92%	244M	0s
23900K	92%	317M	0s
23950K	92%	280M	0s
24000K	92%	290M	0s
24050K	92%	222M	0s
24100K	93%	281M	0s
24150K	93%	277M	0s
24200K	93%	304M	0s

24250K	93%	255M	0s
24300K	93%	300M	0s
24350K	94%	28.3M	0s
24400K	94%	44.0M	0s
24450K	94%	40.9M	0s
24500K	94%	125M	0s
24550K	94%	278M	0s
24600K	94%	10.2M	0s
24650K	95%	135M	0s
24700K	95%	53.4M	0s
24750K	95%	110M	0s
24800K	95%	91.6M	0s
24850K	95%	75.4M	0s
24900K	96%	109M	0s
24950K	96%	83.3M	0s
25000K	96%	185M	0s
25050K	96%	259M	0s
25100K	96%	299M	0s
25150K	97%	289M	0s
25200K	97%	311M	0s
25250K	97%	229M	0s
25300K	97%	319M	0s
25350K	97%	20.9M	0s
25400K	98%	108M	0s
25450K	98%	164M	0s
25500K	98%	259M	0s
25550K	98%	107M	0s
25600K	98%	160M	0s
25650K	99%	134M	0s
25700K	99%	73.1M	0s
25750K	99%	161M	0s
25800K	99%	263M	0s
25850K	99%	279M	0s
25900K	100%	264M=0.3s	

2020-05-30 13:06:45 (93.7 MB/s) - 'title.episode.tsv.gz' saved [26570485/26570485]

/bin/bash: line 2: fg: no job control

In [12]:

```
title_episode = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.episode.tsv', header=True, inferSchema = True)
title_episode.show(3)
```

```
+-----+-----+-----+-----+
  tconst|parentTconst|seasonNumber|episodeNumber|
+-----+-----+-----+-----+
tt0041951|  tt0041038|          1|          9|
tt0042816|  tt0989125|          1|         17|
tt0042889|  tt0989125|         \N|         \N|
+-----+-----+-----+-----+
only showing top 3 rows
```

In [13]:

```
%sh wget https://datasets.imdbws.com/title.principals.tsv.gz  
  
%sh  
gunzip title.principals.tsv.gz
```

```
--2020-05-30 13:06:54-- https://datasets.imdbws.com/title.principals.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.26, 13.224.13.3
2, 13.224.13.37, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.26|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 323355058 (308M) [binary/octet-stream]
Saving to: 'title.principals.tsv.gz'
```

0K	0%	3.85M	80s
50K	0%	6.93M	62s
100K	0%	12.7M	50s
150K	0%	11.2M	44s
200K	0%	16.6M	39s
250K	0%	25.2M	35s
300K	0%	19.1M	32s
350K	0%	22.0M	30s
400K	0%	33.5M	27s
450K	0%	26.9M	26s
500K	0%	46.3M	24s
550K	0%	33.6M	23s
600K	0%	55.3M	21s
650K	0%	38.5M	20s
700K	0%	64.8M	19s
750K	0%	45.8M	19s
800K	0%	47.8M	18s
850K	0%	74.0M	17s
900K	0%	37.0M	17s
950K	0%	88.8M	16s
1000K	0%	179M	15s
1050K	0%	45.8M	15s
1100K	0%	61.1M	15s
1150K	0%	43.4M	14s
1200K	0%	178M	14s
1250K	0%	180M	13s
1300K	0%	161M	13s
1350K	0%	37.7M	13s
1400K	0%	191M	12s
1450K	0%	188M	12s
1500K	0%	173M	12s
1550K	0%	105M	11s
1600K	0%	94.9M	11s
1650K	0%	101M	11s
1700K	0%	157M	11s
1750K	0%	134M	10s
1800K	0%	96.7M	10s
1850K	0%	81.8M	10s
1900K	0%	169M	10s
1950K	0%	147M	10s
2000K	0%	152M	9s
2050K	0%	183M	9s
2100K	0%	109M	9s
2150K	0%	134M	9s
2200K	0%	163M	9s

2250K	0%	162M	9s
2300K	0%	181M	8s
2350K	0%	132M	8s
2400K	0%	162M	8s
2450K	0%	185M	8s
2500K	0%	161M	8s
2550K	0%	146M	8s
2600K	0%	165M	8s
2650K	0%	173M	8s
2700K	0%	157M	7s
2750K	0%	76.7M	7s
2800K	0%	253M	7s
2850K	0%	175M	7s
2900K	0%	197M	7s
2950K	0%	175M	7s
3000K	0%	219M	7s
3050K	0%	281M	7s
3100K	0%	295M	7s
3150K	1%	86.5M	7s
3200K	1%	155M	7s
3250K	1%	141M	7s
3300K	1%	93.6M	6s
3350K	1%	125M	6s
3400K	1%	96.9M	6s
3450K	1%	121M	6s
3500K	1%	159M	6s
3550K	1%	135M	6s
3600K	1%	146M	6s
3650K	1%	153M	6s
3700K	1%	152M	6s
3750K	1%	141M	6s
3800K	1%	153M	6s
3850K	1%	153M	6s
3900K	1%	158M	6s
3950K	1%	117M	6s
4000K	1%	143M	6s
4050K	1%	142M	6s
4100K	1%	136M	6s
4150K	1%	144M	6s
4200K	1%	155M	6s
4250K	1%	146M	6s
4300K	1%	240M	5s
4350K	1%	178M	5s
4400K	1%	297M	5s
4450K	1%	189M	5s
4500K	1%	209M	5s
4550K	1%	184M	5s
4600K	1%	266M	5s
4650K	1%	200M	5s
4700K	1%	84.3M	5s
4750K	1%	121M	5s
4800K	1%	158M	5s
4850K	1%	114M	5s
4900K	1%	77.1M	5s
4950K	1%	129M	5s

5000K	1%	154M	5s
5050K	1%	157M	5s
5100K	1%	155M	5s
5150K	1%	111M	5s
5200K	1%	209M	5s
5250K	1%	207M	5s
5300K	1%	116M	5s
5350K	1%	151M	5s
5400K	1%	195M	5s
5450K	1%	233M	5s
5500K	1%	140M	5s
5550K	1%	119M	5s
5600K	1%	157M	5s
5650K	1%	137M	5s
5700K	1%	157M	5s
5750K	1%	143M	5s
5800K	1%	75.3M	5s
5850K	1%	256M	5s
5900K	1%	286M	5s
5950K	1%	149M	5s
6000K	1%	172M	4s
6050K	1%	254M	4s
6100K	1%	264M	4s
6150K	1%	187M	4s
6200K	1%	304M	4s
6250K	1%	86.6M	4s
6300K	2%	109M	4s
6350K	2%	113M	4s
6400K	2%	73.7M	4s
6450K	2%	127M	4s
6500K	2%	122M	4s
6550K	2%	108M	4s
6600K	2%	127M	4s
6650K	2%	119M	4s
6700K	2%	120M	4s
6750K	2%	108M	4s
6800K	2%	131M	4s
6850K	2%	80.9M	4s
6900K	2%	120M	4s
6950K	2%	89.2M	4s
7000K	2%	178M	4s
7050K	2%	186M	4s
7100K	2%	182M	4s
7150K	2%	76.1M	4s
7200K	2%	132M	4s
7250K	2%	97.4M	4s
7300K	2%	108M	4s
7350K	2%	165M	4s
7400K	2%	105M	4s
7450K	2%	131M	4s
7500K	2%	186M	4s
7550K	2%	113M	4s
7600K	2%	170M	4s
7650K	2%	99.2M	4s
7700K	2%	183M	4s

7750K	2%	82.8M	4s
7800K	2%	181M	4s
7850K	2%	122M	4s
7900K	2%	84.5M	4s
7950K	2%	113M	4s
8000K	2%	188M	4s
8050K	2%	101M	4s
8100K	2%	177M	4s
8150K	2%	87.7M	4s
8200K	2%	171M	4s
8250K	2%	181M	4s
8300K	2%	284M	4s
8350K	2%	216M	4s
8400K	2%	275M	4s
8450K	2%	247M	4s
8500K	2%	303M	4s
8550K	2%	70.4M	4s
8600K	2%	76.1M	4s
8650K	2%	174M	4s
8700K	2%	125M	4s
8750K	2%	151M	4s
8800K	2%	89.6M	4s
8850K	2%	178M	4s
8900K	2%	128M	4s
8950K	2%	160M	4s
9000K	2%	93.7M	4s
9050K	2%	182M	4s
9100K	2%	123M	4s
9150K	2%	101M	4s
9200K	2%	115M	4s
9250K	2%	130M	4s
9300K	2%	114M	4s
9350K	2%	115M	4s
9400K	2%	110M	4s
9450K	3%	132M	4s
9500K	3%	120M	4s
9550K	3%	81.8M	4s
9600K	3%	123M	4s
9650K	3%	182M	4s
9700K	3%	181M	4s
9750K	3%	92.1M	4s
9800K	3%	130M	4s
9850K	3%	176M	4s
9900K	3%	135M	4s
9950K	3%	113M	4s
10000K	3%	85.1M	4s
10050K	3%	173M	4s
10100K	3%	118M	4s
10150K	3%	123M	4s
10200K	3%	120M	4s
10250K	3%	152M	4s
10300K	3%	178M	4s
10350K	3%	249M	4s
10400K	3%	176M	4s
10450K	3%	228M	4s

10500K	3%	170M	4s
10550K	3%	179M	4s
10600K	3%	302M	4s
10650K	3%	78.5M	4s
10700K	3%	120M	4s
10750K	3%	129M	3s
10800K	3%	110M	3s
10850K	3%	175M	3s
10900K	3%	135M	3s
10950K	3%	113M	3s
11000K	3%	113M	3s
11050K	3%	122M	3s
11100K	3%	178M	3s
11150K	3%	84.6M	3s
11200K	3%	111M	3s
11250K	3%	130M	3s
11300K	3%	104M	3s
11350K	3%	117M	3s
11400K	3%	131M	3s
11450K	3%	114M	3s
11500K	3%	122M	3s
11550K	3%	107M	3s
11600K	3%	116M	3s
11650K	3%	125M	3s
11700K	3%	135M	3s
11750K	3%	103M	3s
11800K	3%	124M	3s
11850K	3%	129M	3s
11900K	3%	114M	3s
11950K	3%	99.1M	3s
12000K	3%	93.9M	3s
12050K	3%	122M	3s
12100K	3%	105M	3s
12150K	3%	115M	3s
12200K	3%	118M	3s
12250K	3%	157M	3s
12300K	3%	132M	3s
12350K	3%	100M	3s
12400K	3%	120M	3s
12450K	3%	122M	3s
12500K	3%	110M	3s
12550K	3%	87.4M	3s
12600K	4%	177M	3s
12650K	4%	90.0M	3s
12700K	4%	123M	3s
12750K	4%	135M	3s
12800K	4%	83.8M	3s
12850K	4%	134M	3s
12900K	4%	114M	3s
12950K	4%	116M	3s
13000K	4%	135M	3s
13050K	4%	167M	3s
13100K	4%	266M	3s
13150K	4%	137M	3s
13200K	4%	290M	3s

13250K	4%	177M	3s
13300K	4%	260M	3s
13350K	4%	236M	3s
13400K	4%	217M	3s
13450K	4%	224M	3s
13500K	4%	301M	3s
13550K	4%	218M	3s
13600K	4%	277M	3s
13650K	4%	124M	3s
13700K	4%	288M	3s
13750K	4%	274M	3s
13800K	4%	277M	3s
13850K	4%	74.1M	3s
13900K	4%	182M	3s
13950K	4%	156M	3s
14000K	4%	130M	3s
14050K	4%	125M	3s
14100K	4%	164M	3s
14150K	4%	117M	3s
14200K	4%	185M	3s
14250K	4%	185M	3s
14300K	4%	124M	3s
14350K	4%	111M	3s
14400K	4%	118M	3s
14450K	4%	199M	3s
14500K	4%	266M	3s
14550K	4%	173M	3s
14600K	4%	299M	3s
14650K	4%	176M	3s
14700K	4%	263M	3s
14750K	4%	223M	3s
14800K	4%	293M	3s
14850K	4%	295M	3s
14900K	4%	308M	3s
14950K	4%	253M	3s
15000K	4%	20.9M	3s
15050K	4%	96.0M	3s
15100K	4%	111M	3s
15150K	4%	88.7M	3s
15200K	4%	120M	3s
15250K	4%	124M	3s
15300K	4%	121M	3s
15350K	4%	89.3M	3s
15400K	4%	25.4M	3s
15450K	4%	79.2M	3s
15500K	4%	109M	3s
15550K	4%	82.6M	3s
15600K	4%	162M	3s
15650K	4%	134M	3s
15700K	4%	146M	3s
15750K	5%	88.8M	3s
15800K	5%	126M	3s
15850K	5%	86.7M	3s
15900K	5%	102M	3s
15950K	5%	82.7M	3s

16000K	5%	119M	3s
16050K	5%	68.4M	3s
16100K	5%	98.7M	3s
16150K	5%	90.6M	3s

*** WARNING: skipped 430540 bytes of output ***

299450K	94%	211M	0s
299500K	94%	164M	0s
299550K	94%	82.5M	0s
299600K	94%	122M	0s
299650K	94%	84.4M	0s
299700K	94%	205M	0s
299750K	94%	106M	0s
299800K	94%	97.9M	0s
299850K	94%	207M	0s
299900K	94%	211M	0s
299950K	95%	61.4M	0s
300000K	95%	50.1M	0s
300050K	95%	84.7M	0s
300100K	95%	55.0M	0s
300150K	95%	66.1M	0s
300200K	95%	73.7M	0s
300250K	95%	86.7M	0s
300300K	95%	86.4M	0s
300350K	95%	114M	0s
300400K	95%	144M	0s
300450K	95%	146M	0s
300500K	95%	122M	0s
300550K	95%	132M	0s
300600K	95%	46.9M	0s
300650K	95%	74.3M	0s
300700K	95%	135M	0s
300750K	95%	128M	0s
300800K	95%	139M	0s
300850K	95%	60.8M	0s
300900K	95%	125M	0s
300950K	95%	148M	0s
301000K	95%	161M	0s
301050K	95%	261M	0s
301100K	95%	204M	0s
301150K	95%	199M	0s
301200K	95%	279M	0s
301250K	95%	296M	0s
301300K	95%	210M	0s
301350K	95%	241M	0s
301400K	95%	286M	0s
301450K	95%	272M	0s
301500K	95%	307M	0s
301550K	95%	242M	0s
301600K	95%	223M	0s
301650K	95%	267M	0s
301700K	95%	233M	0s
301750K	95%	262M	0s
301800K	95%	287M	0s

301850K	95%	307M	0s
301900K	95%	288M	0s
301950K	95%	212M	0s
302000K	95%	298M	0s
302050K	95%	268M	0s
302100K	95%	298M	0s
302150K	95%	271M	0s
302200K	95%	295M	0s
302250K	95%	263M	0s
302300K	95%	269M	0s
302350K	95%	227M	0s
302400K	95%	97.8M	0s
302450K	95%	261M	0s
302500K	95%	278M	0s
302550K	95%	260M	0s
302600K	95%	289M	0s
302650K	95%	298M	0s
302700K	95%	309M	0s
302750K	95%	217M	0s
302800K	95%	249M	0s
302850K	95%	294M	0s
302900K	95%	279M	0s
302950K	95%	273M	0s
303000K	95%	295M	0s
303050K	95%	256M	0s
303100K	96%	82.6M	0s
303150K	96%	35.5M	0s
303200K	96%	60.4M	0s
303250K	96%	209M	0s
303300K	96%	140M	0s
303350K	96%	127M	0s
303400K	96%	173M	0s
303450K	96%	303M	0s
303500K	96%	167M	0s
303550K	96%	140M	0s
303600K	96%	171M	0s
303650K	96%	71.4M	0s
303700K	96%	169M	0s
303750K	96%	174M	0s
303800K	96%	309M	0s
303850K	96%	299M	0s
303900K	96%	311M	0s
303950K	96%	13.2M	0s
304000K	96%	96.7M	0s
304050K	96%	98.9M	0s
304100K	96%	23.7M	0s
304150K	96%	42.6M	0s
304200K	96%	98.0M	0s
304250K	96%	153M	0s
304300K	96%	134M	0s
304350K	96%	179M	0s
304400K	96%	222M	0s
304450K	96%	190M	0s
304500K	96%	233M	0s
304550K	96%	103M	0s

304600K	96%	185M	0s
304650K	96%	144M	0s
304700K	96%	150M	0s
304750K	96%	149M	0s
304800K	96%	183M	0s
304850K	96%	185M	0s
304900K	96%	20.4M	0s
304950K	96%	176M	0s
305000K	96%	197M	0s
305050K	96%	204M	0s
305100K	96%	174M	0s
305150K	96%	140M	0s
305200K	96%	58.0M	0s
305250K	96%	50.7M	0s
305300K	96%	35.3M	0s
305350K	96%	56.3M	0s
305400K	96%	68.4M	0s
305450K	96%	161M	0s
305500K	96%	150M	0s
305550K	96%	79.1M	0s
305600K	96%	13.8M	0s
305650K	96%	36.9M	0s
305700K	96%	38.6M	0s
305750K	96%	53.7M	0s
305800K	96%	151M	0s
305850K	96%	134M	0s
305900K	96%	154M	0s
305950K	96%	114M	0s
306000K	96%	116M	0s
306050K	96%	171M	0s
306100K	96%	147M	0s
306150K	96%	136M	0s
306200K	96%	145M	0s
306250K	96%	160M	0s
306300K	97%	141M	0s
306350K	97%	126M	0s
306400K	97%	160M	0s
306450K	97%	82.2M	0s
306500K	97%	118M	0s
306550K	97%	153M	0s
306600K	97%	137M	0s
306650K	97%	136M	0s
306700K	97%	157M	0s
306750K	97%	131M	0s
306800K	97%	148M	0s
306850K	97%	164M	0s
306900K	97%	146M	0s
306950K	97%	131M	0s
307000K	97%	155M	0s
307050K	97%	156M	0s
307100K	97%	144M	0s
307150K	97%	127M	0s
307200K	97%	158M	0s
307250K	97%	163M	0s
307300K	97%	169M	0s

307350K	97%	147M	0s
307400K	97%	145M	0s
307450K	97%	107M	0s
307500K	97%	149M	0s
307550K	97%	132M	0s
307600K	97%	151M	0s
307650K	97%	130M	0s
307700K	97%	157M	0s
307750K	97%	145M	0s
307800K	97%	96.0M	0s
307850K	97%	152M	0s
307900K	97%	127M	0s
307950K	97%	105M	0s
308000K	97%	126M	0s
308050K	97%	145M	0s
308100K	97%	159M	0s
308150K	97%	142M	0s
308200K	97%	153M	0s
308250K	97%	147M	0s
308300K	97%	161M	0s
308350K	97%	134M	0s
308400K	97%	159M	0s
308450K	97%	158M	0s
308500K	97%	152M	0s
308550K	97%	161M	0s
308600K	97%	147M	0s
308650K	97%	155M	0s
308700K	97%	159M	0s
308750K	97%	129M	0s
308800K	97%	141M	0s
308850K	97%	95.7M	0s
308900K	97%	141M	0s
308950K	97%	114M	0s
309000K	97%	142M	0s
309050K	97%	131M	0s
309100K	97%	145M	0s
309150K	97%	131M	0s
309200K	97%	169M	0s
309250K	97%	161M	0s
309300K	97%	133M	0s
309350K	97%	119M	0s
309400K	97%	89.5M	0s
309450K	98%	147M	0s
309500K	98%	146M	0s
309550K	98%	138M	0s
309600K	98%	137M	0s
309650K	98%	158M	0s
309700K	98%	129M	0s
309750K	98%	132M	0s
309800K	98%	165M	0s
309850K	98%	114M	0s
309900K	98%	149M	0s
309950K	98%	132M	0s
310000K	98%	168M	0s
310050K	98%	155M	0s

310100K	98%	173M	0s
310150K	98%	134M	0s
310200K	98%	140M	0s
310250K	98%	162M	0s
310300K	98%	118M	0s
310350K	98%	92.5M	0s
310400K	98%	150M	0s
310450K	98%	119M	0s
310500K	98%	134M	0s
310550K	98%	141M	0s
310600K	98%	137M	0s
310650K	98%	161M	0s
310700K	98%	135M	0s
310750K	98%	130M	0s
310800K	98%	113M	0s
310850K	98%	136M	0s
310900K	98%	139M	0s
310950K	98%	123M	0s
311000K	98%	122M	0s
311050K	98%	141M	0s
311100K	98%	112M	0s
311150K	98%	116M	0s
311200K	98%	152M	0s
311250K	98%	146M	0s
311300K	98%	148M	0s
311350K	98%	152M	0s
311400K	98%	160M	0s
311450K	98%	158M	0s
311500K	98%	128M	0s
311550K	98%	129M	0s
311600K	98%	159M	0s
311650K	98%	82.3M	0s
311700K	98%	118M	0s
311750K	98%	84.3M	0s
311800K	98%	150M	0s
311850K	98%	127M	0s
311900K	98%	141M	0s
311950K	98%	121M	0s
312000K	98%	139M	0s
312050K	98%	115M	0s
312100K	98%	149M	0s
312150K	98%	128M	0s
312200K	98%	114M	0s
312250K	98%	106M	0s
312300K	98%	123M	0s
312350K	98%	76.0M	0s
312400K	98%	142M	0s
312450K	98%	138M	0s
312500K	98%	109M	0s
312550K	98%	132M	0s
312600K	99%	142M	0s
312650K	99%	154M	0s
312700K	99%	155M	0s
312750K	99%	117M	0s
312800K	99%	165M	0s

312850K	99%	165M	0s
312900K	99%	136M	0s
312950K	99%	138M	0s
313000K	99%	140M	0s
313050K	99%	123M	0s
313100K	99%	150M	0s
313150K	99%	108M	0s
313200K	99%	129M	0s
313250K	99%	112M	0s
313300K	99%	103M	0s
313350K	99%	141M	0s
313400K	99%	149M	0s
313450K	99%	115M	0s
313500K	99%	141M	0s
313550K	99%	151M	0s
313600K	99%	93.3M	0s
313650K	99%	138M	0s
313700K	99%	145M	0s
313750K	99%	126M	0s
313800K	99%	145M	0s
313850K	99%	99.9M	0s
313900K	99%	129M	0s
313950K	99%	135M	0s
314000K	99%	147M	0s
314050K	99%	160M	0s
314100K	99%	163M	0s
314150K	99%	146M	0s
314200K	99%	153M	0s
314250K	99%	139M	0s
314300K	99%	186M	0s
314350K	99%	89.9M	0s
314400K	99%	187M	0s
314450K	99%	75.6M	0s
314500K	99%	162M	0s
314550K	99%	116M	0s
314600K	99%	162M	0s
314650K	99%	161M	0s
314700K	99%	101M	0s
314750K	99%	95.5M	0s
314800K	99%	143M	0s
314850K	99%	162M	0s
314900K	99%	148M	0s
314950K	99%	148M	0s
315000K	99%	161M	0s
315050K	99%	158M	0s
315100K	99%	159M	0s
315150K	99%	119M	0s
315200K	99%	164M	0s
315250K	99%	155M	0s
315300K	99%	155M	0s
315350K	99%	135M	0s
315400K	99%	152M	0s
315450K	99%	146M	0s
315500K	99%	160M	0s
315550K	99%	134M	0s

```
315600K ..... 99% 159M 0s
315650K ..... 99% 161M 0s
315700K ..... 99% 146M 0s
315750K ..... 100% 147M=3.4s
```

2020-05-30 13:06:58 (92.0 MB/s) - 'title.principals.tsv.gz' saved [323355058/323355058]

/bin/bash: line 2: fg: no job control

In [14]:

```
title_principals = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.principals.tsv', header=True, inferSchema = True)
title_principals.show(3)
```

tconst	ordering	nconst	category	job	characters
tt0000001	1	nm1588970	self	\N	["Self"]
tt0000001	2	nm0005690	director	\N	\N
tt0000001	3	nm0374658	cinematographer	director of photo...	\N

only showing top 3 rows

In [15]:

```
%sh wget https://datasets.imdbws.com/title.ratings.tsv.gz  
  
%sh  
gunzip title.ratings.tsv.gz
```

```
--2020-05-30 13:08:31-- https://datasets.imdbws.com/title.ratings.tsv.gz
Resolving datasets.imdbws.com (datasets.imdbws.com)... 13.224.13.32, 13.224.13.3
7, 13.224.13.54, ...
Connecting to datasets.imdbws.com (datasets.imdbws.com)|13.224.13.32|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 5165953 (4.9M) [binary/octet-stream]
Saving to: 'title.ratings.tsv.gz'
```

0K	0%	4.23M	1s
50K	1%	7.61M	1s
100K	2%	12.5M	1s
150K	3%	15.8M	1s
200K	4%	13.2M	1s
250K	5%	28.9M	0s
300K	6%	31.0M	0s
350K	7%	17.9M	0s
400K	8%	42.1M	0s
450K	9%	27.1M	0s
500K	10%	49.4M	0s
550K	11%	44.9M	0s
600K	12%	42.7M	0s
650K	13%	61.2M	0s
700K	14%	45.3M	0s
750K	15%	51.6M	0s
800K	16%	64.7M	0s
850K	17%	59.7M	0s
900K	18%	83.0M	0s
950K	19%	142M	0s
1000K	20%	52.1M	0s
1050K	21%	75.1M	0s
1100K	22%	76.0M	0s
1150K	23%	68.6M	0s
1200K	24%	83.0M	0s
1250K	25%	232M	0s
1300K	26%	71.8M	0s
1350K	27%	78.9M	0s
1400K	28%	88.1M	0s
1450K	29%	148M	0s
1500K	30%	109M	0s
1550K	31%	96.6M	0s
1600K	32%	41.0M	0s
1650K	33%	122M	0s
1700K	34%	177M	0s
1750K	35%	224M	0s
1800K	36%	119M	0s
1850K	37%	152M	0s
1900K	38%	291M	0s
1950K	39%	81.2M	0s
2000K	40%	270M	0s
2050K	41%	135M	0s
2100K	42%	164M	0s
2150K	43%	158M	0s
2200K	44%	241M	0s

2250K	45%	188M	0s
2300K	46%	151M	0s
2350K	47%	157M	0s
2400K	48%	162M	0s
2450K	49%	153M	0s
2500K	50%	158M	0s
2550K	51%	155M	0s
2600K	52%	237M	0s
2650K	53%	124M	0s
2700K	54%	251M	0s
2750K	55%	106M	0s
2800K	56%	187M	0s
2850K	57%	170M	0s
2900K	58%	143M	0s
2950K	59%	247M	0s
3000K	60%	271M	0s
3050K	61%	306M	0s
3100K	62%	286M	0s
3150K	63%	48.3M	0s
3200K	64%	151M	0s
3250K	65%	87.1M	0s
3300K	66%	156M	0s
3350K	67%	81.3M	0s
3400K	68%	107M	0s
3450K	69%	75.1M	0s
3500K	70%	110M	0s
3550K	71%	82.3M	0s
3600K	72%	106M	0s
3650K	73%	90.5M	0s
3700K	74%	109M	0s
3750K	75%	80.4M	0s
3800K	76%	67.7M	0s
3850K	77%	76.3M	0s
3900K	78%	148M	0s
3950K	79%	139M	0s
4000K	80%	103M	0s
4050K	81%	141M	0s
4100K	82%	104M	0s
4150K	83%	112M	0s
4200K	84%	118M	0s
4250K	85%	114M	0s
4300K	86%	153M	0s
4350K	87%	125M	0s
4400K	88%	149M	0s
4450K	89%	152M	0s
4500K	90%	143M	0s
4550K	91%	145M	0s
4600K	92%	155M	0s
4650K	93%	145M	0s
4700K	94%	158M	0s
4750K	95%	142M	0s
4800K	96%	156M	0s
4850K	97%	147M	0s
4900K	98%	170M	0s
4950K	99%	139M	0s

5000K 100% 159M=0.08s

2020-05-30 13:08:31 (62.0 MB/s) - 'title.ratings.tsv.gz' saved [5165953/5165953]

/bin/bash: line 2: fg: no job control

In [16]:

```
title_ratings = spark.read.option("sep", "\t").csv('file:/databricks/driver/title.ratings.tsv', header=True, inferSchema = True)
title_ratings.show(3)
```

+-----+-----+-----+		
tconst	averageRating	numVotes
+-----+-----+-----+		
tt0000001	5.6	1617
tt0000002	6.0	198
tt0000003	6.5	1299
+-----+-----+-----+		
only showing top 3 rows		

In [17]:

```
names_basic.show(3)
title_basics.show(3)
title_principals.show(3)
title_akas.show(3)
title_crew.show(3)
title_episode.show(3)
title_ratings.show(3)
```

```

+-----+-----+-----+-----+-----+-----+
-----+
nconst|    primaryName|birthYear|deathYear|    primaryProfession|    knownFo
rTitles|
+-----+-----+-----+-----+-----+-----+
-----+
nm0000001|    Fred Astaire|    1899|    1987|soundtrack,actor,...|tt0043044,tt0
0531...|
nm0000002|    Lauren Bacall|    1924|    2014|    actress,soundtrack|tt0071877,tt0
1170...|
nm0000003|Brigitte Bardot|    1934|    \N|actress,soundtrac...|tt0054452,tt0
0491...|

```

```

+-----+-----+-----+-----+-----+-----+
-----+
only showing top 3 rows

```

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
tconst|titleType|    primaryTitle|    originalTitle|isAdult|startYear|
endYear|runtimeMinutes|    genres|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
tt0000001|    short|    Carmencita|    Carmencita|    0|    1894|
\N|    1|    Documentary,Short|
tt0000002|    short|Le clown et ses c...|Le clown et ses c...|    0|    1892|
\N|    5|    Animation,Short|
tt0000003|    short|    Pauvre Pierrot|    Pauvre Pierrot|    0|    1892|
\N|    4|Animation,Comedy,...|

```

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
only showing top 3 rows

```

```

+-----+-----+-----+-----+-----+-----+
tconst|ordering|    nconst|    category|    job|characters|
+-----+-----+-----+-----+-----+-----+
tt0000001|    1|nm1588970|    self|    \N|    ["Self"]|
tt0000001|    2|nm0005690|    director|    \N|    \N|
tt0000001|    3|nm0374658|cinematographer|director of photo...|    \N|

```

```

+-----+-----+-----+-----+-----+-----+
only showing top 3 rows

```

```

+-----+-----+-----+-----+-----+-----+
--+-----+
titleId|ordering|    title|region|language|    types|    attribute
s|isOriginalTitle|
+-----+-----+-----+-----+-----+-----+
--+-----+
tt0000001|    1|    Карменцита|    UA|    \N|imdbDisplay|
\N|    0|
tt0000001|    2|    Carmencita|    DE|    \N|    \N|literal titl
e|    0|
tt0000001|    3|Carmencita - span...|    HU|    \N|imdbDisplay|
\N|    0|
+-----+-----+-----+-----+-----+-----+

```

```
--+-----+
only showing top 3 rows
```

```
+-----+-----+-----+
  tconst|directors|writers|
+-----+-----+-----+
tt0000001|nm0005690|      \N|
tt0000002|nm0721526|      \N|
tt0000003|nm0721526|      \N|
+-----+-----+-----+
only showing top 3 rows
```

```
+-----+-----+-----+-----+
  tconst|parentTconst|seasonNumber|episodeNumber|
+-----+-----+-----+-----+
tt0041951|  tt0041038|           1|           9|
tt0042816|  tt0989125|           1|          17|
tt0042889|  tt0989125|          \N|          \N|
+-----+-----+-----+-----+
only showing top 3 rows
```

```
+-----+-----+-----+
  tconst|averageRating|numVotes|
+-----+-----+-----+
tt0000001|           5.6|    1617|
tt0000002|           6.0|     198|
tt0000003|           6.5|    1299|
+-----+-----+-----+
only showing top 3 rows
```

Network Inference, Let's build a network

In the following questions you will look to summarise the data and build a network. We want to examine a network that abstracts how actors and actress are related through their co-participation in movies. To that end perform the following steps:

Q1 Create a DataFrame that combines the information on each of the titles (i.e., movies, tv-shows, etc ...) and the information on the participants in those movies (i.e., actors, directors, etc ...), make sure the actual names of the movies and participants are included. It may be worth reviewing the following questions to see how this dataframe will be used.

How many rows does your dataframe have?

In [19]:

```
joined1 = title_principals.join(title_basics, title_principals.tconst == title_basics.tconst, how='left')
df_network = joined1.join(names_basic, joined1.nconst == names_basic.nconst, how='left')
```

In [20]:

```
print("My final answer is:", df_network.count())
```

My final answer is: 39471189

In [21]:

```
# #Second chance
# df_network = title_principals.join(title_basics,['tconst'],how='left_outer').join(names_basic,['nconst'],how='left_outer')
# df_network.take(1)
```

md **Q2** Create a new DataFrame based on the previous step, with the following removed:

1. Any participant that is not an actor or actress (as measured by the category column);
2. All adult movies;
3. All dead actors or actresses;
4. All actors or actresses born before 1920 or with no date of birth listed;
5. All titles that are not of the type movie.

How many rows does your dataframe have?

In [23]:

```
df_filter = df_network.filter(df_network.category.isin(['actor', 'actress'])) \
    .filter(df_network.isAdult != 1) \
    .filter(df_network.deathYear == "\\N") \
    .filter(df_network.birthYear.isNotNull()) \
    .filter(df_network.birthYear >= 1920) \
    .filter(df_network.titleType == 'movie')
```

In [24]:

```
print("My final answer is:", df_filter.count())
```

My final answer is: 451091

Q3 Convert the above Dataframe to an RDD (you can use `.rdd` to convert a dataframe to and RDD of row objects). Use map and reduce to create a paired RDD which counts how many movies each actor / actress appears in.

Display names of the top 10 actors/actresses according to the number of movies in which they appeared. Be careful to deal with different actors / actresses with the same name, these could be different people.

In [26]:

```
df_1 = df_filter.rdd.map(lambda x :((x[2], x[16]), 1))
df_2 = df_1.reduceByKey(lambda x,y : x+y)
df_f = df_2.sortBy(lambda x: x[1], False)
```

In [27]:

```

from graphframes import *
from pyspark.sql.types import *
import graphframes.graphframe as gfm

df_ff = df_f.map(lambda x: (x[0][0],x[0][1],x[1]))
df_ff

Id_actor = StructField("Id_actor",StringType(),True)
Movie = StructField("Movie",StringType(),True)
nvotes = StructField("nvotes",StringType(),True)

df_ff_3 = sqlContext.createDataFrame(df_ff, StructType([Id_actor, Movie,nvotes])).persist()

print("My final answer is:")
print(df_ff_3.show(10))

```

My final answer is:

```

+-----+-----+-----+
| Id_actor | Movie | nvotes |
+-----+-----+-----+
| nm0103977 | Brahmanandam | 809 |
| nm0007123 | Mammootty | 379 |
| nm0482320 | Mohanlal | 343 |
| nm0149822 | Mithun Chakraborty | 332 |
| nm0007106 | Shakti Kapoor | 309 |
| nm0415549 | Jagathi Sreekumar | 303 |
| nm0035067 | Cüneyt Arkin | 294 |
| nm0374974 | Helen | 281 |
| nm0534867 | Madhu | 277 |
| nm0004429 | Dharmendra | 270 |
+-----+-----+-----+

```

only showing top 10 rows

None

Q4 Start with the dataframe from **Q2**. Generate a DataFrame that lists all links of your network. Here we shall consider that a link connects a pair of actors/actresses if they participated in at least one movie together (actors / actresses should be represented by their unique ID's). For every link we then need anytime a pair of actors were together in a movie as a link in each direction (A -> B and B -> A). However links should be distinct we do not need duplicates when two actors worked together in several movies.

In [29]:

```

actors_rdd_1 = df_filter.rdd.map(list).map(lambda x : (x[0], x[2]))
actors_rdd_2 = df_filter.rdd.map(list).map(lambda x : (x[0], x[2]))
actors_rdd_2.take(4)

```

```

Out[249]: [('tt0110116', 'nm0000198'),
 ('tt1345836', 'nm0000198'),
 ('tt0097125', 'nm0000198'),
 ('tt3239932', 'nm0000198')]

```

In [30]:

```
from pyspark.sql import Row
actor_pairs_row = actors_rdd_1.join(actors_rdd_2).map(lambda x :Row(x[1][0], x[1][1]))
.filter(lambda x: x[0] != x[1]).distinct()
```

In [31]:

```
ActorA = StructField("ActorA",StringType(),True)
ActorB = StructField("ActorB",StringType(),True)

df_actors_link = sqlContext.createDataFrame(actor_pairs_row, StructType([ActorA, ActorB
])).persist()
df_ff_3.show(10)
df_actors_link.show(4)
```

```
+-----+-----+
  ActorA|  ActorB|
+-----+-----+
nm3216408|nm0453304|
nm0544425|nm0000778|
nm2507102|nm0102403|
nm0668271|nm0001151|
+-----+-----+
only showing top 4 rows
```

In [32]:

```
print("My final answer is:", actor_pairs_row.count(),df_actors_link.show(10))
```

```
+-----+-----+
  ActorA|  ActorB|
+-----+-----+
nm3216408|nm0453304|
nm0544425|nm0000778|
nm2507102|nm0102403|
nm0668271|nm0001151|
nm0429385|nm0005541|
nm0059847|nm0036924|
nm0000665|nm0000546|
nm0879203|nm0863831|
nm1231899|nm0695177|
nm0666140|nm0744037|
+-----+-----+
only showing top 10 rows
```

My final answer is: 712112 None

Q5 Compute the page rank of each actor. This can be done using GraphFrames or by using RDDs and the iterative implementation of the PageRank algorithm. Do not take more than 5 iterations and use reset probability = 0.1.

List the top 10 actors / actresses by pagerank.

In [34]:

```
df_vertices = df_actors_link.select(df_actors_link['ActorA']).selectExpr("ActorA as id")
                        ).distinct()
df_link = df_actors_link.selectExpr("ActorA as src", "ActorB as dst")
df_link.take(3)
```

```
Out[253]: [Row(src='nm3216408', dst='nm0453304'),
Row(src='nm0544425', dst='nm0000778'),
Row(src='nm2507102', dst='nm0102403')]
```

In [35]:

```
ourGraph = gfm.GraphFrame(df_vertices, df_link)
# ourGraph.vertices.show()
# ourGraph.edges.show()
```

In [36]:

```
pageRanks = ourGraph.pageRank(resetProbability=0.1, maxIter = 5)
#pageRanks.vertices.sort("pagerank", ascending = False).show(10)
```

In [37]:

```
print("My final answer is:")
print(pageRanks.vertices.sort("pagerank", ascending = False).show(10))
```

My final answer is:

```
+-----+-----+
      id|      pagerank|
+-----+-----+
nm0000616| 40.6478874245532|
nm0000514|24.226138058070163|
nm0001744|23.419848197831612|
nm0001803|20.555441795470156|
nm0000448|17.663207647023903|
nm0001698|17.583580336085777|
nm0004193|16.881922296919786|
nm0000367|16.832482667099892|
nm0000800|16.111563992409874|
nm0626259|15.962778479477862|
+-----+-----+
only showing top 10 rows
```

None

Q6: Create an RDD with the number of outDegrees for each actor. Display the top 10 by outDegrees.

In [39]:

```
#ID, PrimaryName and Outdegree
q6 = ourGraph.outDegrees.sort('outDegree', ascending = False)
print("My final answer is:")
print(q6.show(10))
```

```
My final answer is:
+-----+-----+
      id|outDegree|
+-----+-----+
nm0000616|      438|
nm0000514|      289|
nm0000367|      263|
nm0001744|      261|
nm0945189|      253|
nm0451600|      239|
nm0149822|      232|
nm0001803|      231|
nm0874676|      227|
nm0938893|      225|
+-----+-----+
only showing top 10 rows
```

None

Let's play Kevin's own game

Q7 Start with the graphframe / dataframe you developed in the previous section. Using Spark GraphFrame and/or Spark Core library perform the following steps:

1. Identify the id of Kevin Bacon, there are two actors named 'Kevin Bacon', we will use the one with the highest degree, that is, the one that participated in most titles;
2. Estimate the shortest path between every actor/actress in the database and Kevin Bacon, keep a dataframe with a column that includes the number of steps to Kevin Bacon as you will need it later (this will require a little processing to get from the graphframes output);
3. Summarise the data, that is, count the number of actors at each number of degress from kevin bacon (you will need to deal with actors unconnected to kevin bacon, if not connected to Kevin Bacon given these actors / actresses a score of 20). You could use the display() barchart functionality of databricks to easily display the distribution of the data.

Note: The solution time on this step can be ~15 minutes

In [41]:

```
df_filter1 = df_filter.join(df_filter,['nconst','primaryName'],how='left')
```

In [42]:

```
import pyspark.sql.functions
from pyspark.sql.functions import split, substring, length, col, expr
from pyspark.sql import functions as F
from pyspark.sql.functions import *
```


In [43]:

```
# Q7(2)
q7_2 = ourGraph.shortestPaths(landmarks=["nm0000102"])
```

In [44]:

```
from pyspark.sql.functions import explode_outer
distances_value = q7_2.select("id", explode_outer("distances"))
```

In [45]:

Summarise the data, that is, count the number of actors at each number of degrees from Kevin Bacon (you will need to deal with actors unconnected to Kevin Bacon, if not connected to Kevin Bacon given these actors / actresses a score of 20). You could use the display() barchart functionality of databricks to easily display the distribution of the data.

```
#groupby e count
q7_f = distances_value.groupBy('value').count().sort("count", ascending = False)
print("My final answer is:")
print(q7_f.show(10))
```

My final answer is:

```
+-----+-----+
value|count|
+-----+-----+
  4|29068|
  3|18311|
  5|13095|
null| 6529|
  2| 3302|
  6| 2251|
  7|  311|
  1|  126|
  8|   38|
 10|   12|
+-----+-----+
only showing top 10 rows
```

None

In [46]:

```
# Alternative
# Id = StructField("Id",StringType(),True)
# Distance = StructField("Distance",StringType(),True)

# q7_3 = sqlContext.createDataFrame(q7_2_1, StructType([Id, Distance])).persist()
# q7_3.show(4)
# q7_4 = q7_3.withColumn('Distance1', split(q7_3['Distance'], '=')[1])
# q7_5 = q7_4.withColumn('Distance2', split(q7_4['Distance1'], '')[0])
# q7_6 = q7_5.withColumn('code', split(q7_5['Distance'], '=')[0])
# q7_8 = q7_6.drop('Distance', 'Distance1')
# q7_7 = q7_8.select('code', substring('code', 2, 10000).alias('nconst_1'))
# new_df = q7_7.join(q7_8, ['code'])
# new_df = new_df.drop('code')
# new_df.show(3)
```

Exploring the data with RDD's

Using RDDs and (not dataframes) answer the following questions (if you loaded your data into spark in a dataframe you can convert to an RDD of rows easily using `.rdd`) :

Hint: paired RDD's will be useful.

Q8 Movies can have multiple genres. Considering only titles of the type 'movie' what is the combination of genres that is the most popular (as measured by number of reviews)?

In [48]:

```
movies_genres = df_filter.join(title_ratings, ['tconst'])
```

In [49]:

```
title = movies_genres.rdd.map(list).map(lambda x : (x[14], x[22]))
# title.collect()
```

In [50]:

```
# title.distinct().countByKey().sortBy()
title_1 = title.reduceByKey(lambda x,y: x+y)
title_2 = title_1.sortBy(lambda x: x[1], False)

print("My final answer is:", title_2.take(1))
```

```
My final answer is: [('Action,Adventure,Sci-Fi', 164846210)]
```

Q9 Movies can have multiple genres. Considering only titles of the type 'movie', and movies with more than 500 ratings, what is the combination of genres that has the highest **average movie rating** (you can average the movie rating for each movie in that genre combination).

In [52]:

```
Rdd_q9 = title_basics.rdd.map(lambda x:(x[0],(x[1],x[-1]))).join(title_ratings.rdd.map(
lambda x: (x[0],(x[1],x[2]))))
# Rdd_q9.take(5)
```

In [53]:

```
movie = Rdd_q9.map(lambda x: (x[1][0][0],x[1][0][1],x[1][1][0],x[1][1][1]))
```

In [54]:

```
q9 = movie.filter(lambda x: (x[0] == 'movie') & (x[3] >= 500)).map(lambda x: (x[1],x[2]
))
```

In [55]:

```
Avg_q9 = q9.mapValues(lambda x: (x,1))
Avg_q9 = Avg_q9.reduceByKey(lambda x,y: (x[0]+y[0], x[1]+y[1])).mapValues(lambda x: x[0]
]/x[1]).sortBy(lambda x: x[1], False)
print("My final answer is:", Avg_q9.take(1))
```

```
My final answer is: [('Music,Musical', 8.5)]
```

In [56]:

```
#Alternative
# average_moving_Rate2 = average_moving_Rate1.reduceByKey(lambda x,y: x+y)
# average_moving_Rate3 = average_moving_Rate2.sortBy(lambda x: x[1], False)
# average_moving_Rate3.take(5)
# average_moving_Rate1.distinct().countByKey()
# avg_by_key = average_moving_Rate1 \
#     .mapValues(lambda v: (v, 1)) \
#     .reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1])) \
#     .mapValues(lambda v: v[0]/v[1])
# Sorted_avg_by_key = avg_by_key.sortBy(lambda x: x[1], False)
# Sorted_avg_by_key.take(5)
```

Q10 Movies can have multiple genres. What is **the individual genre** which is the most popular as measured by number of votes. Votes for multiple genres count towards each genre listed.

Hint: Think about the wordcount exercise we have done with RDDs.

In [58]:

```
Rdd_q10 = title_basics.rdd.map(lambda x:(x[0],(x[1],x[-1]))).join(title_ratings.rdd.map(
(lambda x: (x[0],(x[1],x[2]))))
```

In [59]:

```
movie_10 = Rdd_q10.map(lambda x: (x[1][0][0],x[1][0][1],x[1][1][0],x[1][1][1]))
movie_10.take(5)
```

```
Out[55]: [('short', 'Documentary,Short', 5.6, 1617),
          ('short', 'Documentary,Short', 5.7, 1539),
          ('short', 'Documentary,Short', 5.5, 812),
          ('short', 'Documentary,Short,Sport', 4.1, 147),
          ('short', 'Documentary,Short', 4.3, 15)]
```

In [60]:

```
q10 = movie_10.filter(lambda x: (x[0] == 'movie')).map(lambda x: (x[1],x[3]))
```

```
Out[56]: [('Drama', 17),
          ('Drama', 12),
          ('\\N', 13),
          ('\\N', 17),
          ('Crime,Thriller', 12)]
```

In [61]:

```
Split_RDD = q10.map(lambda x: (x[0].split(","), x[1]))
Split_RDD0 = Split_RDD.flatMap(lambda x: [(y, int(x[1])) for y in x[0]])
Split_RDD1 = Split_RDD0.reduceByKey(lambda x,y: x+y)
Split_RDD_Final = Split_RDD1.sortBy(lambda x: x[1], False)
print("My final answer is:", Split_RDD_Final.take(1))
```

```
My final answer is: [('Drama', 405750616)]
```

Engineering the perfect cast

We have created a number of potential features for predicting the rating of a movie based on its cast. Use sparkML to build a simple linear model to predict the rating of a movie based on the following features:

1. The total number of movies in which the actors / actresses in the current movie have acted (based on Q3)
2. The average pagerank of the cast in each movie (based on Q5)
3. The average outDegree of the cast in each movie (based on Q6)
4. The average value for for the cast of degrees of Kevin Bacon (based on Q7).

If you were unable to generate any of these features as you could not answer the previous questions, just skip that particular feature.

You will need to create a dataframe with the required features and label. Use a pipeline to create the vectors required by sparkML and apply the model. Remember to split your dataset, leave 30% of the data for testing, when splitting your data use the option `seed=0`.

Q11 Provide the coefficients of the regression and the accuracy of your model on the test dataset according to RSME.

In [63]:

```
# Movie id, Actor ID, -> join -> table with movie ID, actor ID and the information for
each actor
# Get the table from question 2
# And then join the info about the score rate
#Table: Movie id, actors, all variables about the questions, avg rating too,
# One movie appears many times, is one line per actor
# Ex Titanic - Leonardo has 15 , and Olivia has 10 -> there will be 25
# Score Rating, use the max to take it
# title_principals.show()
```

```

+-----+-----+-----+-----+-----+-----+
--+
  tconst|ordering|  nconst|      category|      job|      characters
|
+-----+-----+-----+-----+-----+-----+
--+
tt0000001|      1|nm1588970|      self|      \N|      ["Self"]
|
tt0000001|      2|nm0005690|    director|      \N|      \N
|
tt0000001|      3|nm0374658|cinematographer|director of photo...|      \N
|
tt0000002|      1|nm0721526|    director|      \N|      \N
|
tt0000002|      2|nm1335271|    composer|      \N|      \N
|
tt0000003|      1|nm0721526|    director|      \N|      \N
|
tt0000003|      2|nm5442194|    producer|    producer|      \N
|
tt0000003|      3|nm1335271|    composer|      \N|      \N
|
tt0000003|      4|nm5442200|      editor|      \N|      \N
|
tt0000004|      1|nm0721526|    director|      \N|      \N
|
tt0000004|      2|nm1335271|    composer|      \N|      \N
|
tt0000005|      1|nm0443482|      actor|      \N|["Blacksmith"]
|
tt0000005|      2|nm0653042|      actor|      \N|["Assistant"]
|
tt0000005|      3|nm0005690|    director|      \N|      \N
|
tt0000005|      4|nm0249379|    producer|    producer|      \N
|
tt0000006|      1|nm0005690|    director|      \N|      \N
|
tt0000007|      1|nm0179163|      actor|      \N|      \N
|
tt0000007|      2|nm0183947|      actor|      \N|      \N
|
tt0000007|      3|nm0005690|    director|      \N|      \N
|
tt0000007|      4|nm0374658|    director|      \N|      \N
|
+-----+-----+-----+-----+-----+-----+
--+
only showing top 20 rows

```

In [64]:

```
# Question 3 (SUM)
df_f

df_ff = df_f.map(lambda x: (x[0][0],x[0][1],x[1]))
df_ff.collect()

Id_actor = StructField("nconst",StringType(),True)
Movie = StructField("Movie",StringType(),True)
nvotes = StructField("TotalActs",StringType(),True)

df_ff_3 = sqlContext.createDataFrame(df_ff, StructType([Id_actor, Movie,nvotes])).persist()

q11_MR = df_ff_3.join(title_principals,['nconst']).select(['nconst','TotalActs','Movie','tconst']).distinct().groupBy('tconst').agg({'TotalActs':'sum'})
```

In [65]:

```
#Question 5 (AVG) #PageRank
rank_renamed = pageRanks.vertices.withColumnRenamed('id','nconst')

q11_PR = rank_renamed.join(title_principals,'nconst').select(['tconst','nconst','pagerank']).distinct().groupBy('tconst').agg({'pagerank':'avg'})
q11_PR.show()
```

In [66]:

```
#Question 6 (AVG)

# df_out = spark.createDataFrame(ourGraph.outDegrees, schema=['nconst','OutDegree'])
df_out = ourGraph.outDegrees.withColumnRenamed("id","nconst")

q11_OD = df_out.join(title_principals,'nconst').select(['tconst','nconst','outDegree']).distinct().groupBy('tconst').agg({'outDegree':'avg'})
```

In [67]:

```
#Question 7 (AVG)

distancia = distances_value.withColumnRenamed("id","nconst")
distancia = distancia.withColumnRenamed("value","distance")

q11_DD = distancia.join(title_principals,'nconst').select(['nconst','distance','tconst']).distinct().groupBy('tconst').agg({'distance':'avg'})
```

In [68]:

```
all_together = q11_MR.join(q11_PR, ['tconst'],'inner').join(q11_OD, ['tconst'],'inner').join(q11_DD, ['tconst'],'inner').join(title_ratings, ["tconst"], how="left_outer")
```

In [69]:

```
all_together_without_nan = all_together.na.drop()
```

In [70]:

```
#Don't need the tconst, numVotes and averageRating columns to create the features set  
all_together_without_nan = all_together_without_nan.drop("tconst", "numVotes")
```

In [71]:

```
train_test = all_together_without_nan.randomSplit([0.7,0.3], seed=0)  
train = train_test[0]  
test = train_test[1]
```

In [72]:

```
features_Columns = all_together_without_nan.columns  
#removing the target from features  
features_Columns.remove("averageRating")  
#Concatenate all the features columns into a single feature vector in a new column called rawfeatures  
vectorAssembler = VectorAssembler(inputCols = features_Columns, outputCol = "features")
```

In [73]:

```
# Define LinearRegression algorithm  
model = LinearRegression(labelCol = "averageRating")  
  
#Combine all part assembled into a single pipeline  
pipeline = Pipeline(stages = [vectorAssembler, model])  
  
# # Print the fitted model parameters  
# print(">>> ModelA intercept: %r, coefficient: %r" % (modelA.intercept, modelA.coefficients[0]))
```

In [74]:

```
#Fitting the model  
pipelineModel = pipeline.fit(train)
```


In [75]:

```
pred = pipelineModel.transform(test)
print('My pre final answer is:')
print(pred.show())
```

sum(TotalActs) avg(pagerank) avg(outDegree) avg(distance) averageRating				
features	prediction			
1.0 0.15388023022273636	1.0	5.0	5.2	
[1.0,0.1538802302... 6.738432970469312	1.0	3.0	6.3	
1.0 0.162737367142549	1.0	3.0	6.9	
[1.0,0.1627373671... 7.185733923450423	1.0	3.0	7.2	
1.0 0.162737367142549	1.0	4.0	3.7	
[1.0,0.1627373671... 7.185733923450423	1.0	4.0	7.0	
1.0 0.162737367142549	1.0	5.0	7.2	
[1.0,0.1627373671... 7.185733923450423	1.0	5.0	5.9	
1.0 0.16791307168844505	1.0	4.0	6.6	
[1.0,0.1679130716... 6.9608558268126215	1.0	4.0	8.1	
1.0 0.19473857188077948	2.0	3.0	7.8	
[1.0,0.1947385718... 6.957426985871079	2.0	3.0	8.1	
1.0 0.20332216586340757	3.0	3.0	7.4	
[1.0,0.2033221658... 6.7321132921058755	3.0	3.0	8.2	
1.0 0.2105218908680771	2.0	4.0	7.5	
[1.0,0.2105218908... 6.731193021777761	3.0	3.0	7.4	
1.0 0.22704666506160767	2.0	4.0	6.5	
[1.0,0.2270466650... 6.953297358768236	1.0	4.0	7.4	
1.0 0.22965060706387092	3.0	3.0	8.2	
[1.0,0.2296506070... 6.952964522366948	2.0	4.0	7.5	
1.0 0.24709725743081595	3.0	3.0	7.4	
[1.0,0.2470972574... 7.177427593951279	2.0	4.0	6.5	
1.0 0.26841102627865837	1.0	4.0	7.4	
[1.0,0.2684110262... 7.174703263641683	2.0	3.0	8.6	
1.0 0.2715918183356083	2.0	3.0	6.7	
[1.0,0.2715918183... 7.176773263143577				
1.0 0.27810086881857393				
[1.0,0.2781008688... 7.175941274974614				
1.0 0.2912773061673283				
[1.0,0.2912773061... 6.947563954031903				
1.0 0.2919258536181646				
[1.0,0.2919258536... 7.1741741625794235				
1.0 0.2927553532000262				
[1.0,0.2927553532... 6.947375029756758				
1.0 0.30756869810114995				
[1.0,0.3075686981... 6.943005015974899				
1.0 0.31456468460774367				
[1.0,0.3145646846... 6.944587356539025				
1.0 0.3181350057007517				
[1.0,0.3181350057... 7.168347534384205				
only showing top 20 rows				

localhost:8888/nbconvert/html/Documents/Portugal/NOVA/Disciplinas/Second Semester/Big Data Analytics/Final Project/M20190925 Exam ... 106/108

In [76]:

```
from pyspark.ml.evaluation import RegressionEvaluator

RMSE = RegressionEvaluator(labelCol = "averageRating", predictionCol = "prediction", metricName = "rmse")
print('My final answer is:')
rmse = RMSE.evaluate(pred)
print(rmse)
```

My final answer is:
1.3404186671846205

Q12 What score would your model predict for the 1997 movie Titanic and how does this compare to it's actual score.

In [78]:

```
titanic = df_filter.filter((df_filter.primaryTitle == "Titanic") & (df_filter.startYear == 1997)).distinct()
```

In [79]:

```
toPredict = all_together.filter(all_together.tconst == "tt0120338")
```

In [80]:

```
#Preprocess to do the prediction
toPredict = toPredict.drop("tconst", "numVotes")
```

In [81]:

```
prediction = pipelineModel.transform(toPredict)
# print("And My final Answer is: ")
# print(prediction.select("averageRating", 'Prediction').show())
prediction.select("averageRating", 'Prediction').show()
```

```
+-----+-----+
averageRating|      Prediction|
+-----+-----+
          7.8|6.3255040028494625|
+-----+-----+
```

In [82]:

```
# from pyspark.ml.feature import VectorAssembler, VectorIndexer
# featuresCols = toPredict.columns

# This concatenates all feature columns into a single feature vector in a new column "rawFeatures".
# vectorAssembler = VectorAssembler(inputCols=featuresCols, outputCol="rawFeatures")
# This identifies categorical features and indexes them.
# vectorIndexer = VectorIndexer(inputCol="rawFeatures", outputCol="features", maxCategories=4)
```

Q13 Create dummy variables for each of the top 10 movie genres from **Q10**. These variable should have a value of 1 if the movie was rated with that genre and 0 otherwise. For example the 1997 movie Titanic should have a 1 in the dummy variable column for Romance, and a 1 in the dummy variable column for Drama, and 0's in all the other dummy variable columns.

If you were unable to answer Q10 you can just select 10 different genres and construct the same data.

Note: Question 10 uses the number of votes per genre and not the average votes per genre.

Does adding these variables to the regression improve your results? What is the new RMSE and predicted rating for the 1997 movie Titanic.

In [84]:

```
### Pipeline  
# I tried to do but i got negaive scores
```

Q14 Improve your model by testing different machine learning algorithms, using hyperparameter tuning on these algorithms, changing the included features. Be careful not to cheat and use test data in the training of your model.

Note: We are not testing your knowledge of different algorithms, we are just testing that you can apply the different tools in the spark toolkit and can compare between them.

What is the RMSE of you final model and what rating does it predict for the 1997 movie Titanic.

In [86]:

```
#Pipeline  
# It was taking so much to run and i couldn't finish it
```