Northeastern University

Final Project Document

Topic : Netflix Movies & TV shows Dataset Analysis

Course: INFO 7250 - Engineering of Big Data

Academic Year: Fall 2020

Professor: Mr. Yusuf Ozbek

Submitted By:

Snehal Vasudeo Pathak

Submission Date:

December 15, 2020

Summary

For the Project, I have selected the Netflix Movies & TV shows Dataset for analysis. The dataset is downloaded from Kaggle.com from below link:

https://www.kaggle.com/shivamb/netflix-shows

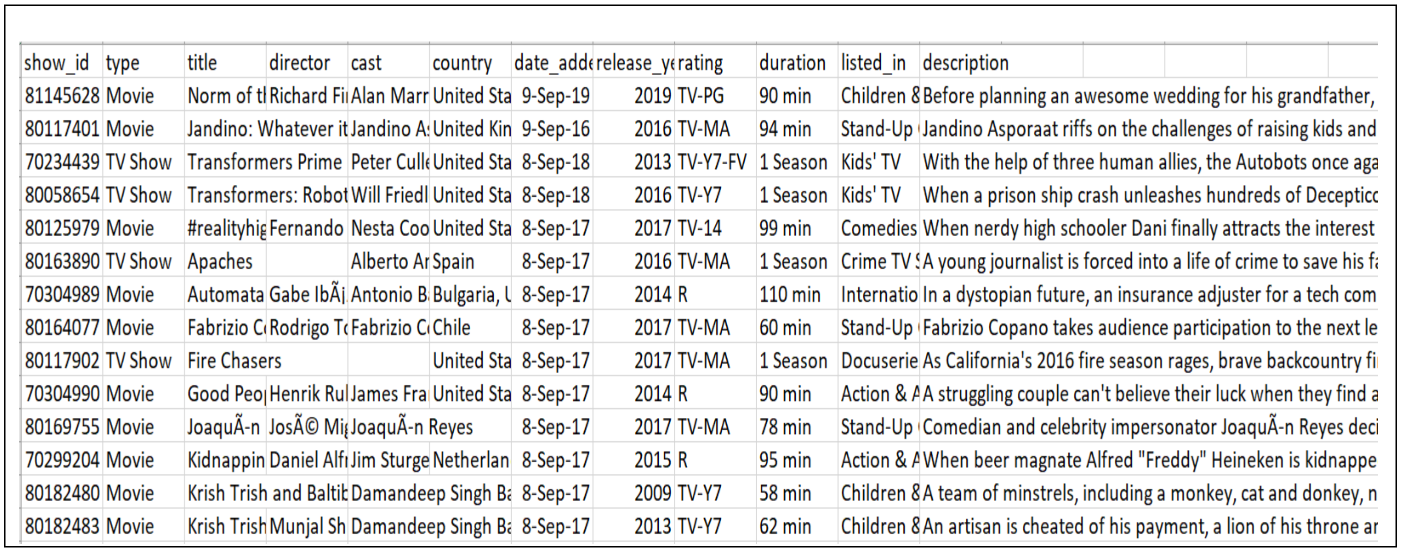
The dataset has one csv file named as '**netflix_titles.csv**'

Netflix_titles.csv

This file has around 6235 records.

Columns in the dataset:

- show_id
- type
- title
- director
- cast
- country
- date_added
- release_year
- rating
- duration
- listed_in
- description



show_id	type	title	director	cast	country	date_adde	release_ye	rating	duration	listed_in	description
81145628	Movie	Norm of tl	Richard Fi	Alan Marr	United Sta	9-Sep-19	2019	TV-PG	90 min	Children 8	Before planning an awesome wedding for his grandfather,
80117401	Movie	Jandino: Whatever it		Jandino A:	United Kin	9-Sep-16	2016	TV-MA	94 min	Stand-Up	Jandino Asporaat riffs on the challenges of raising kids and
70234439	TV Show	Transformers Prime		Peter Culle	United Sta	8-Sep-18	2013	TV-Y7-FV	1 Season	Kids' TV	With the help of three human allies, the Autobots once aga
80058654	TV Show	Transformers: Robot		Will Friedl	United Sta	8-Sep-18	2016	TV-Y7	1 Season	Kids' TV	When a prison ship crash unleashes hundreds of Decepticc
80125979	Movie	#realityhig	Fernando	Nesta Coo	United Sta	8-Sep-17	2017	TV-14	99 min	Comedies	When nerdy high schooler Dani finally attracts the interest
80163890	TV Show	Apaches		Alberto Ar	Spain	8-Sep-17	2016	TV-MA	1 Season	Crime TV S	A young journalist is forced into a life of crime to save his fa
70304989	Movie	Automata	Gabe IbÃ¡	Antonio B	Bulgaria, U	8-Sep-17	2014	R	110 min	Internatio	In a dystopian future, an insurance adjuster for a tech com
80164077	Movie	Fabrizio C	Rodrigo To	Fabrizio C	Chile	8-Sep-17	2017	TV-MA	60 min	Stand-Up	Fabrizio Copano takes audience participation to the next le
80117902	TV Show	Fire Chasers			United Sta	8-Sep-17	2017	TV-MA	1 Season	Docuserie	As California's 2016 fire season rages, brave backcountry fi
70304990	Movie	Good Peo	Henrik Rul	James Fra	United Sta	8-Sep-17	2014	R	90 min	Action & A	A struggling couple can't believe their luck when they find a
80169755	Movie	JoaquÃ-n	JosÃ© Mi	JoaquÃ-n Reyes		8-Sep-17	2017	TV-MA	78 min	Stand-Up	Comedian and celebrity impersonator JoaquÃ-n Reyes deci
70299204	Movie	Kidnappin	Daniel Alfr	Jim Sturge	Netherlan	8-Sep-17	2015	R	95 min	Action & A	When beer magnate Alfred "Freddy" Heineken is kidnappe
80182480	Movie	Krish Trish and Baltib		Damandeep Singh Ba		8-Sep-17	2009	TV-Y7	58 min	Children 8	A team of minstrels, including a monkey, cat and donkey, n
80182483	Movie	Krish Trish	Munjal Sh	Damandeep Singh Ba		8-Sep-17	2013	TV-Y7	62 min	Children 8	An artisan is cheated of his payment, a lion of his throne ar

Analysis Done for the Project:

Sr No	Analysis Description	Implementation Details
1	To Count Number of Movies and TV shows by the year they were released	MongoDB - **MapReduce**
2	Find movies and TV shows by Country and listed_in category using mongoDB indexes	MongoDB - **Custom Index**
3	Count number of Total Movies and TV shows in dataset	Hadoop - **MapReduce**
4	Movies and TV show analysis based on ratings using custom counter	MapReduce – **Custom Counter**
5	Implement partitioning on the basis on year the Movies and TV shows added in Netflix dataset	MapReduce – **Data Organization Techniques** Partitioning
6	Find Distinct Genres in the dataset	MapReduce – **Filtering Techniques** Distinct Pattern
7	Get movies and TV shows which are released before year 1970	Apache **Hive**
8	Find Movie or TV shows from Netflix data which are listed as “Stand-Up Comedy” and cast is “Russell Peters”	Apache Hive
9	Find Directors from India with most contents	Apache Hive
10	Find Movies details based on duration of the movie	Apache **Pig**
11	Percent Increase/Decrease in Netflix Data wrt release year 2000	Apache Pig
12	Number of Movies/TV shows by Countries	**Tableau**

MONGODB ANALYSIS:

Steps to start Mongo shell :

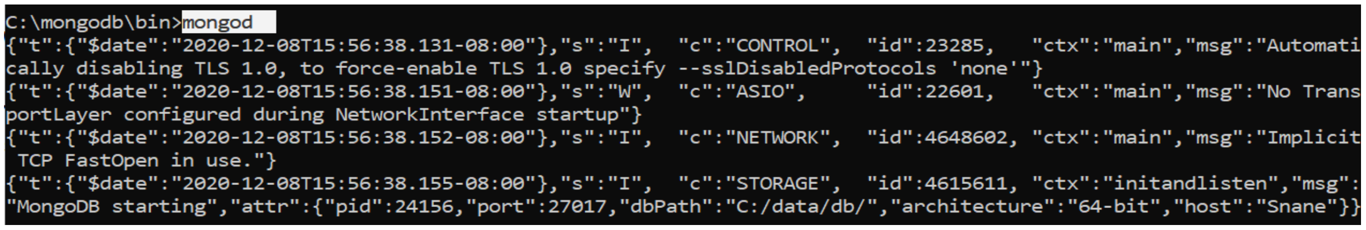
1. Navigate to MongoDB directory
   cd C:/mongoDB/bin

2. Start MongoDB Daemon
   mongod

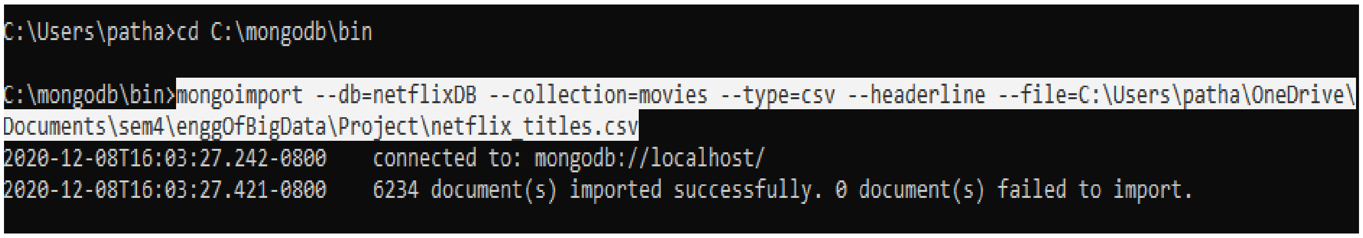
3. Import Dataset into netflixDB using mongoimport
   mongodb --db=netflixDB --collection=movies --type=csv --headerline --file=C:\Users\patha\OneDrive\Documents\sem4\enggOfBigData\Project\netflix_titles.csv

4. Start mongo shell
   mongo

5. Switch to netflixDB
   use netflixDB;



```
C:\mongodb\bin>mongod
{"t":{"$date":"2020-12-08T15:56:38.131-08:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automati
cally disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2020-12-08T15:56:38.151-08:00"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No Trans
portLayer configured during NetworkInterface startup"}
{"t":{"$date":"2020-12-08T15:56:38.152-08:00"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit
 TCP FastOpen in use."}
{"t":{"$date":"2020-12-08T15:56:38.155-08:00"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":
"MongoDB starting","attr":{"pid":24156,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"Snane"}}
```

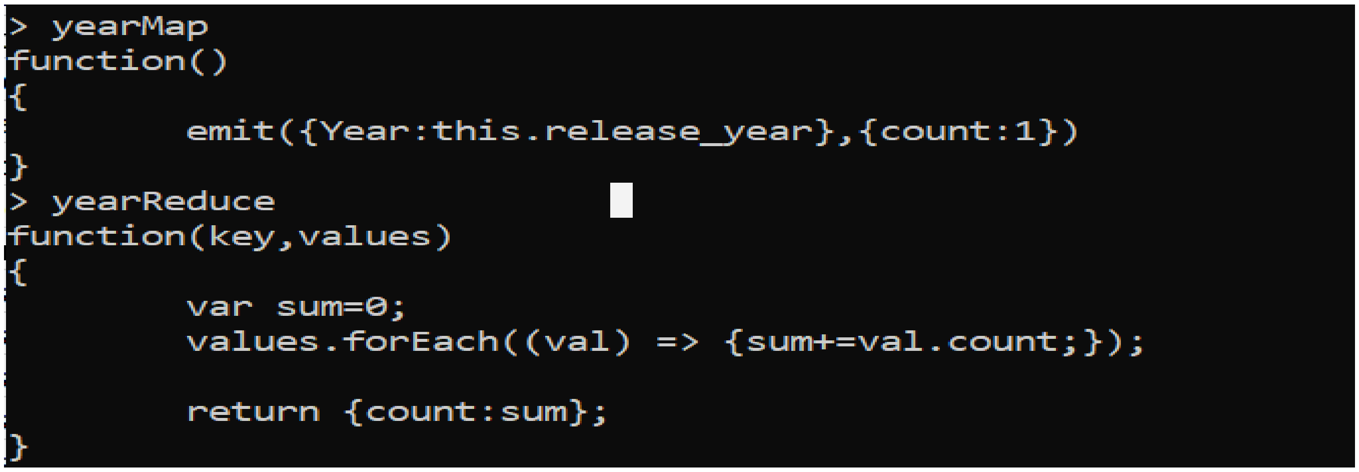


```
C:\Users\patha>cd C:\mongodb\bin

C:\mongodb\bin>mongoimport --db=netflixDB --collection=movies --type=csv --headerline --file=C:\Users\patha\OneDrive\
Documents\sem4\enggOfBigData\Project\netflix_titles.csv
2020-12-08T16:03:27.242-0800    connected to: mongodb://localhost/
2020-12-08T16:03:27.421-0800    6234 document(s) imported successfully. 0 document(s) failed to import.
```

Analysis 01 - Count No of Movies and TV shows by the year they were released

In this analysis, Total number of Movies and TV shows are counted for each year using MongoDB MapReduce

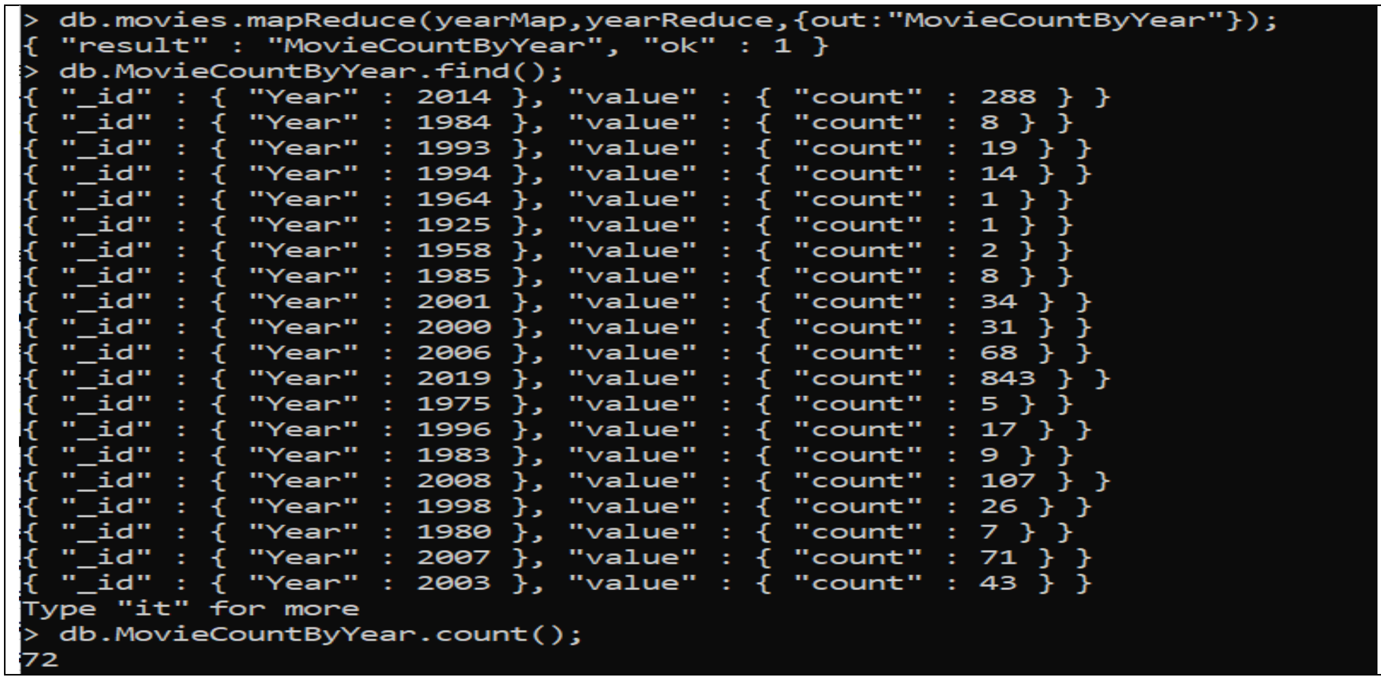
- Write Map and Reduce function to perform analysis:



```
> yearMap
function()
{
        emit({Year:this.release_year},{count:1})
}
> yearReduce
function(key,values)
{
        var sum=0;
        values.forEach((val) => {sum+=val.count;});

        return {count:sum};
}
```

- Execute MapReduce Job
  Db.movies.mapReduce(yearMap, yearReduce, {out: “MoviesCountByYear”});

- Print results
  Db.MoviesCountByYear.find();



```
> db.movies.mapReduce(yearMap,yearReduce,{out:"MovieCountByYear"});
{ "result" : "MovieCountByYear", "ok" : 1 }
> db.MovieCountByYear.find();
{ "_id" : { "Year" : 2014 }, "value" : { "count" : 288 } }
{ "_id" : { "Year" : 1984 }, "value" : { "count" : 8 } }
{ "_id" : { "Year" : 1993 }, "value" : { "count" : 19 } }
{ "_id" : { "Year" : 1994 }, "value" : { "count" : 14 } }
{ "_id" : { "Year" : 1964 }, "value" : { "count" : 1 } }
{ "_id" : { "Year" : 1925 }, "value" : { "count" : 1 } }
{ "_id" : { "Year" : 1958 }, "value" : { "count" : 2 } }
{ "_id" : { "Year" : 1985 }, "value" : { "count" : 8 } }
{ "_id" : { "Year" : 2001 }, "value" : { "count" : 34 } }
{ "_id" : { "Year" : 2000 }, "value" : { "count" : 31 } }
{ "_id" : { "Year" : 2006 }, "value" : { "count" : 68 } }
{ "_id" : { "Year" : 2019 }, "value" : { "count" : 843 } }
{ "_id" : { "Year" : 1975 }, "value" : { "count" : 5 } }
{ "_id" : { "Year" : 1996 }, "value" : { "count" : 17 } }
{ "_id" : { "Year" : 1983 }, "value" : { "count" : 9 } }
{ "_id" : { "Year" : 2008 }, "value" : { "count" : 107 } }
{ "_id" : { "Year" : 1998 }, "value" : { "count" : 26 } }
{ "_id" : { "Year" : 1980 }, "value" : { "count" : 7 } }
{ "_id" : { "Year" : 2007 }, "value" : { "count" : 71 } }
{ "_id" : { "Year" : 2003 }, "value" : { "count" : 43 } }
Type "it" for more
> db.MovieCountByYear.count();
72
```

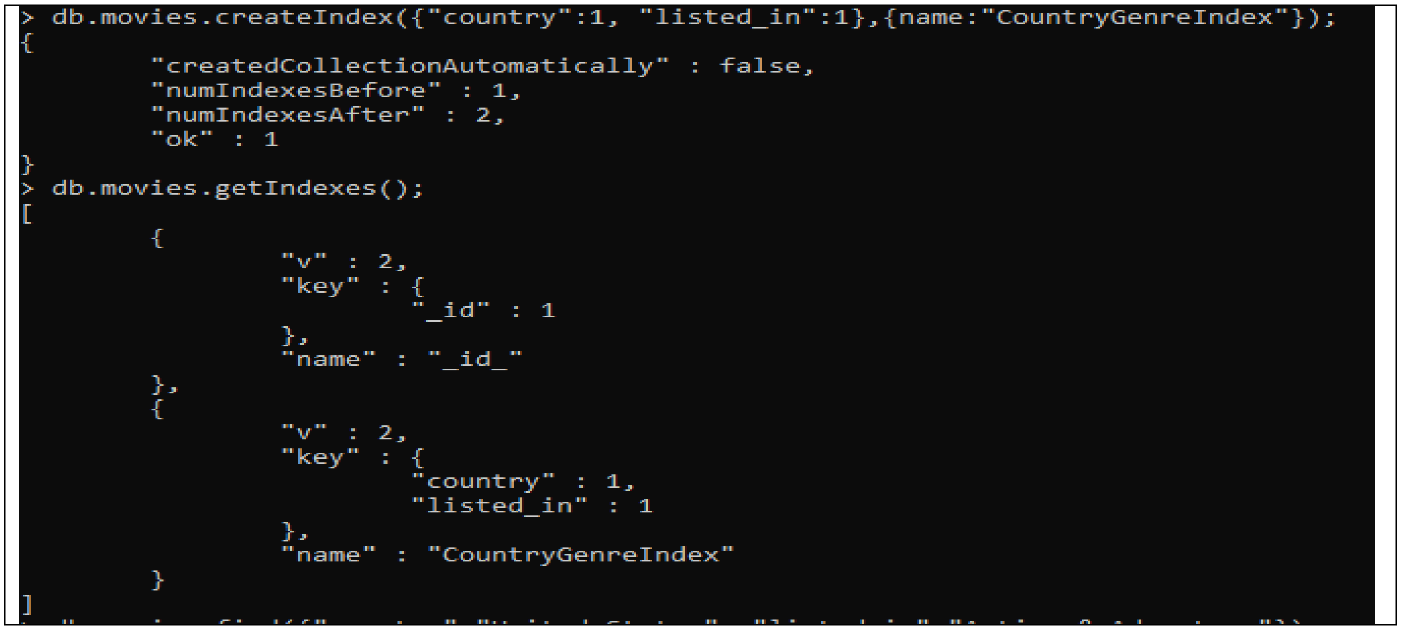
Analysis 02 - Find Movies and TV shows by country and listed_in category using mongoDB indexes

In this Analysis, Custom index is created to search Movies and TV shows based on Country and listed_in (genre).

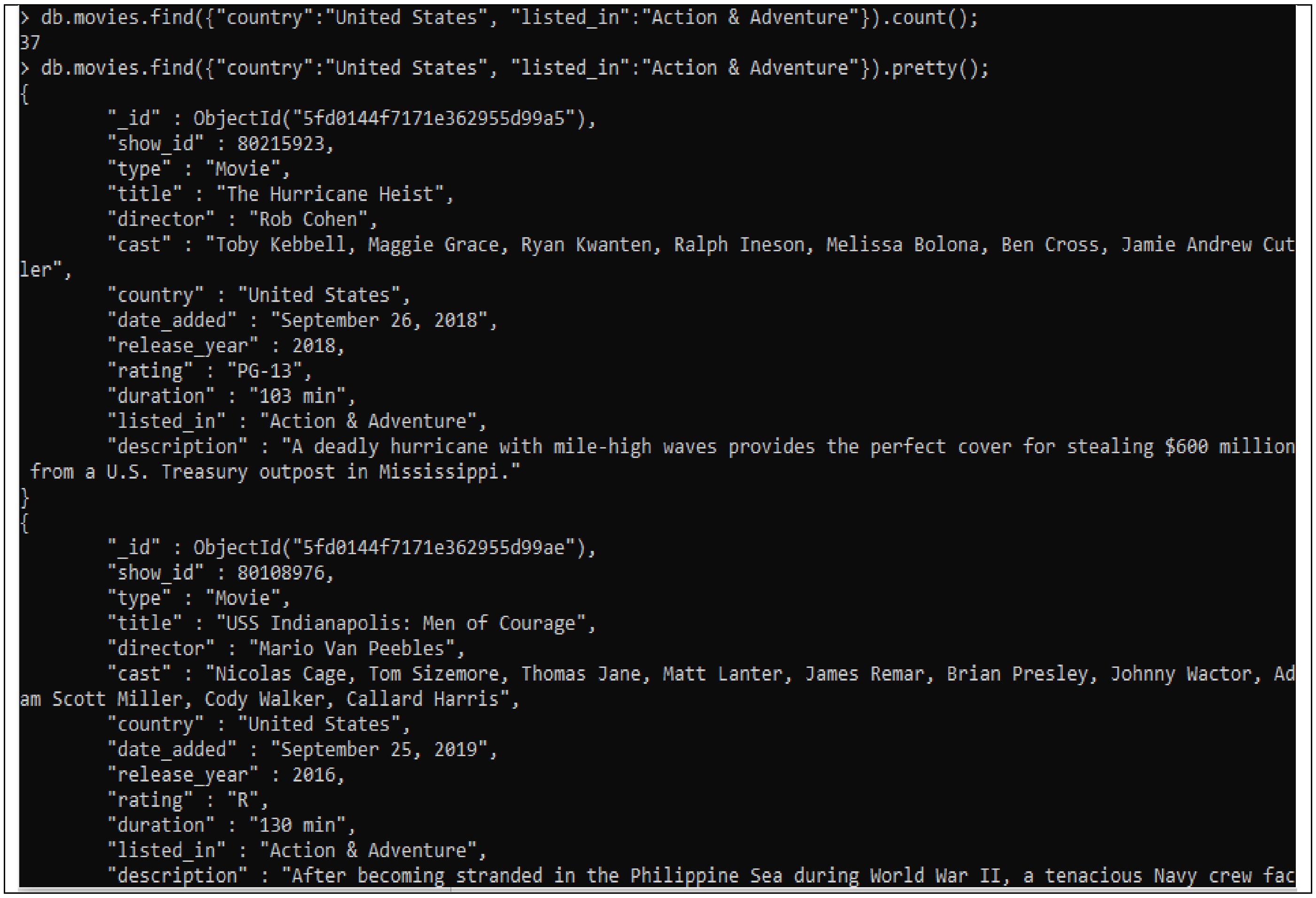
- Create Index “CountryGenreIndex” for country and listed_in fields
  db.movies.createIndex({“country”:1,”listed_in”:1},{name:”CountryGenreIndex”});

- Check Index created
  db.movies.getIndexes();

- Find Movies and TV shows using created Index
  Db.movies.find({“country”: “United States”, ”listed_in”:”Action & Adventure”}).pretty();



```
> db.movies.createIndex({"country":1, "listed_in":1},{name:"CountryGenreIndex"});
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.movies.getIndexes();
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "country" : 1,
                        "listed_in" : 1
                },
                "name" : "CountryGenreIndex"
        }
]
```



```
> db.movies.find({"country":"United States", "listed_in":"Action & Adventure"}).count();
37
> db.movies.find({"country":"United States", "listed_in":"Action & Adventure"}).pretty();
{
        "_id" : ObjectId("5fd0144f7171e362955d99a5"),
        "show_id" : 80215923,
        "type" : "Movie",
        "title" : "The Hurricane Heist",
        "director" : "Rob Cohen",
        "cast" : "Toby Kebbell, Maggie Grace, Ryan Kwanten, Ralph Ineson, Melissa Bolona, Ben Cross, Jamie Andrew Cut
ler",
        "country" : "United States",
        "date_added" : "September 26, 2018",
        "release_year" : 2018,
        "rating" : "PG-13",
        "duration" : "103 min",
        "listed_in" : "Action & Adventure",
        "description" : "A deadly hurricane with mile-high waves provides the perfect cover for stealing $600 million
 from a U.S. Treasury outpost in Mississippi."
}
{
        "_id" : ObjectId("5fd0144f7171e362955d99ae"),
        "show_id" : 80108976,
        "type" : "Movie",
        "title" : "USS Indianapolis: Men of Courage",
        "director" : "Mario Van Peebles",
        "cast" : "Nicolas Cage, Tom Sizemore, Thomas Jane, Matt Lanter, James Remar, Brian Presley, Johnny Wactor, Ad
am Scott Miller, Cody Walker, Callard Harris",
        "country" : "United States",
        "date_added" : "September 25, 2019",
        "release_year" : 2016,
        "rating" : "R",
        "duration" : "130 min",
        "listed_in" : "Action & Adventure",
        "description" : "After becoming stranded in the Philippine Sea during World War II, a tenacious Navy crew fac
```

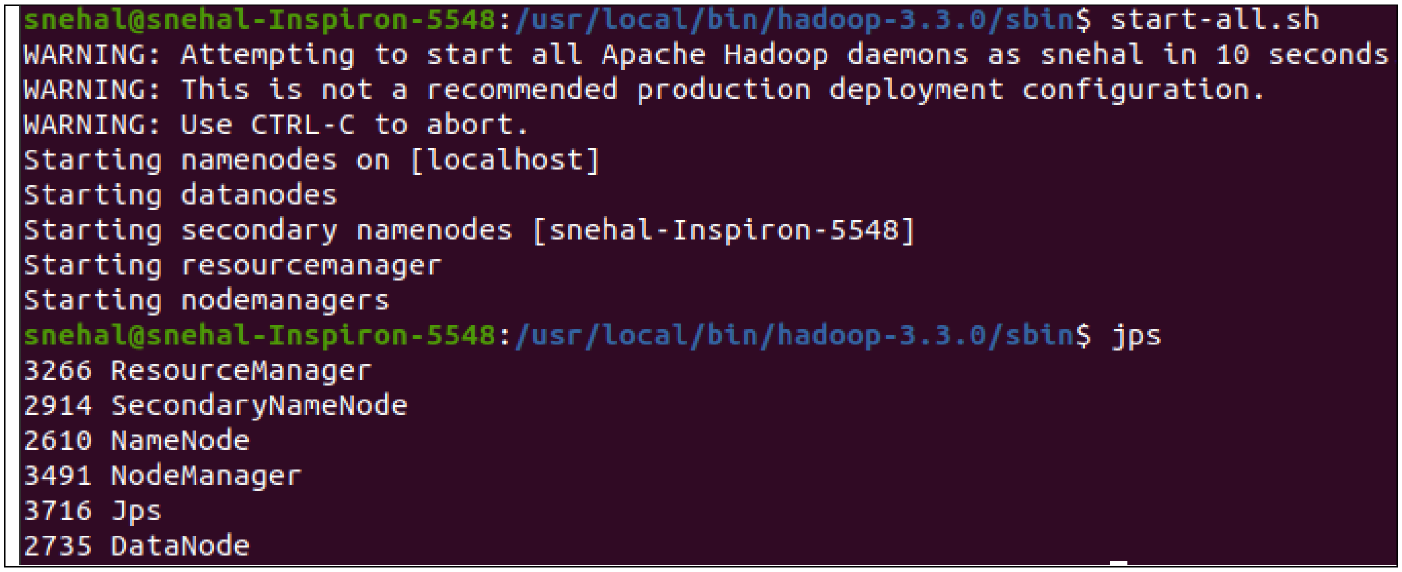
HADOOP ANALYSIS

Steps to start HADOOP

1. Navigate to Hadoop Directory:
   cd /usr/local/bin/Hadoop-3.3.0/sbin

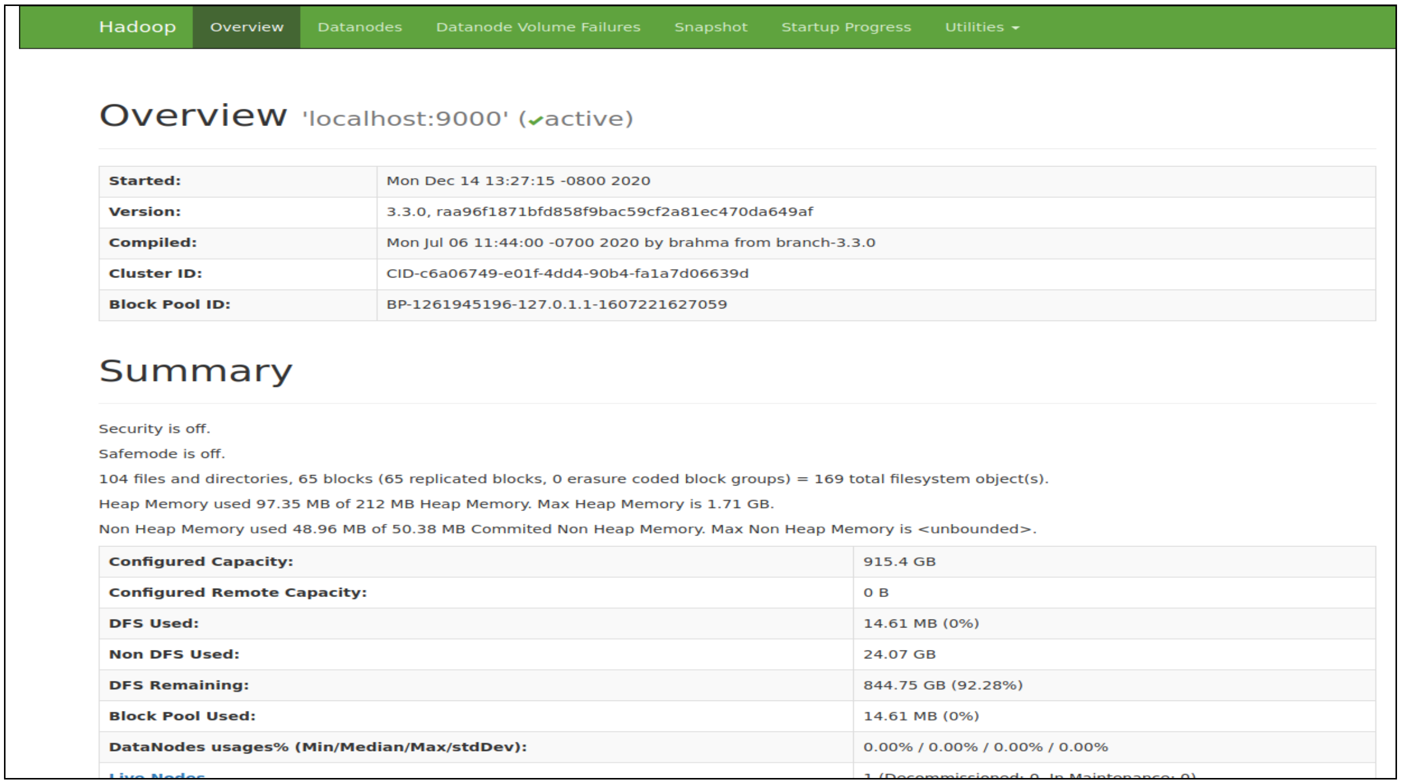
2. Start Hadoop daemons
   ./start-all.sh

3. Check Hadoop Daemons
   jps



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/sbin$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as snehal in 10 seconds
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [snehal-Inspiron-5548]
Starting resourcemanager
Starting nodemanagers
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/sbin$ jps
3266 ResourceManager
2914 SecondaryNameNode
2610 NameNode
3491 NodeManager
3716 Jps
2735 DataNode
```

4. Navigate to Hadoop UI
   http://localhost:9870/



Overview 'localhost:9000' (✓active)

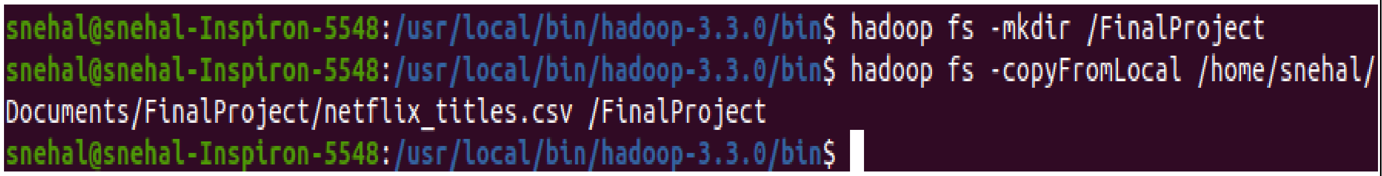
Started:	Mon Dec 14 13:27:15 -0800 2020
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Mon Jul 06 11:44:00 -0700 2020 by brahma from branch-3.3.0
Cluster ID:	CID-c6a06749-e01f-4dd4-90b4-fa1a7d06639d
Block Pool ID:	BP-1261945196-127.0.1.1-1607221627059

Summary

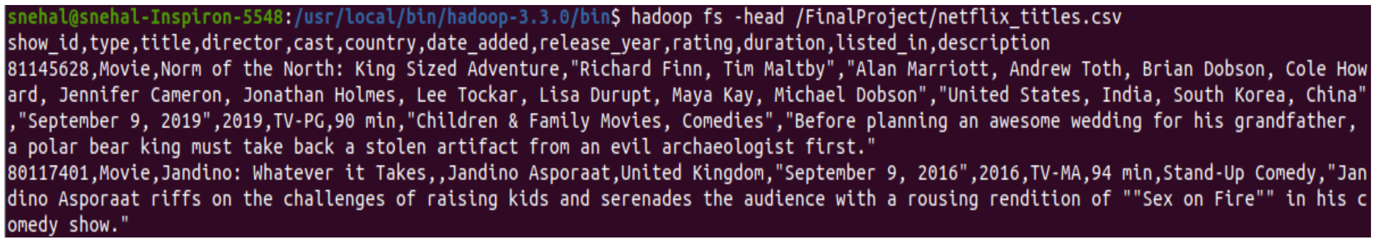
Security is off.
Safemode is off.
104 files and directories, 65 blocks (65 replicated blocks, 0 erasure coded block groups) = 169 total filesystem object(s).
Heap Memory used 97.35 MB of 212 MB Heap Memory. Max Heap Memory is 1.71 GB.
Non Heap Memory used 48.96 MB of 50.38 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	915.4 GB
Configured Remote Capacity:	0 B
DFS Used:	14.61 MB (0%)
Non DFS Used:	24.07 GB
DFS Remaining:	844.75 GB (92.28%)
Block Pool Used:	14.61 MB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

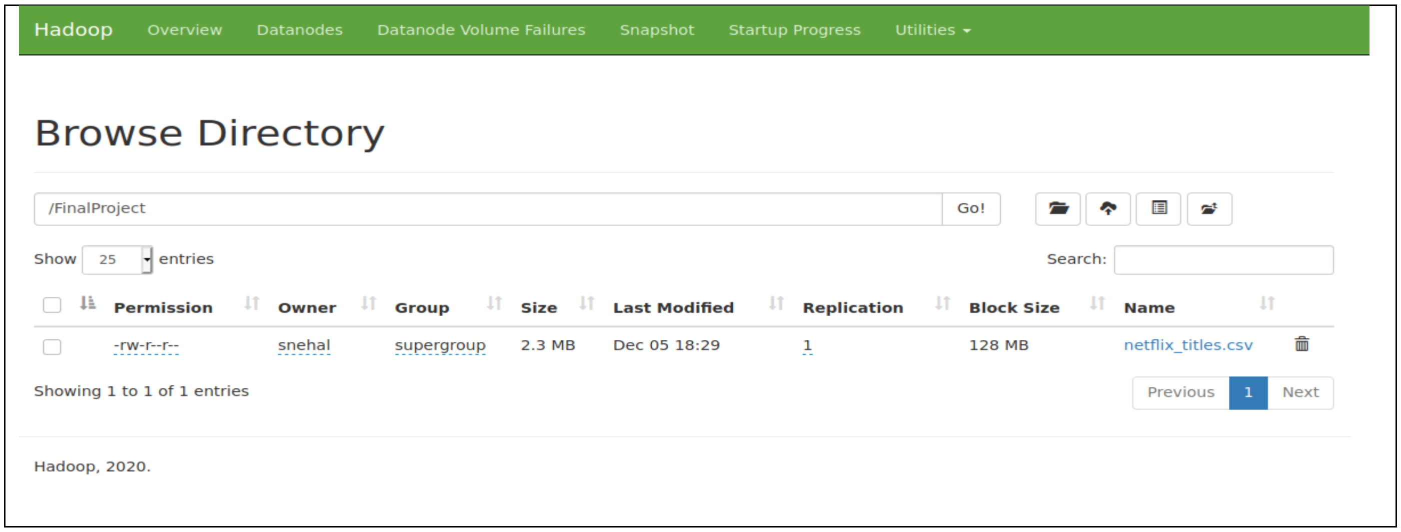
Copy Netflix_titles.csv file from Local File System to HDFS



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -mkdir /FinalProject
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -copyFromLocal /home/snehal/
Documents/FinalProject/netflix_titles.csv /FinalProject
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ 
```



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -head /FinalProject/netflix_titles.csv
show_id,type,title,director,cast,country,date_added,release_year,rating,duration,listed_in,description
81145628,Movie,Norm of the North: King Sized Adventure,"Richard Finn, Tim Maltby","Alan Marriott, Andrew Toth, Brian Dobson, Cole How
ard, Jennifer Cameron, Jonathan Holmes, Lee Tockar, Lisa Durupt, Maya Kay, Michael Dobson","United States, India, South Korea, China"
,"September 9, 2019",2019,TV-PG,90 min,"Children & Family Movies, Comedies","Before planning an awesome wedding for his grandfather,
a polar bear king must take back a stolen artifact from an evil archaeologist first."
80117401,Movie,Jandino: Whatever it Takes,,Jandino Asporaat,United Kingdom,"September 9, 2016",2016,TV-MA,94 min,Stand-Up Comedy,"Jan
dino Asporaat riffs on the challenges of raising kids and serenades the audience with a rousing rendition of ""Sex on Fire"" in his c
omedy show."
```



Browse Directory

/FinalProject

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	snehal	supergroup	2.3 MB	Dec 05 18:29	1	128 MB	netflix_titles.csv

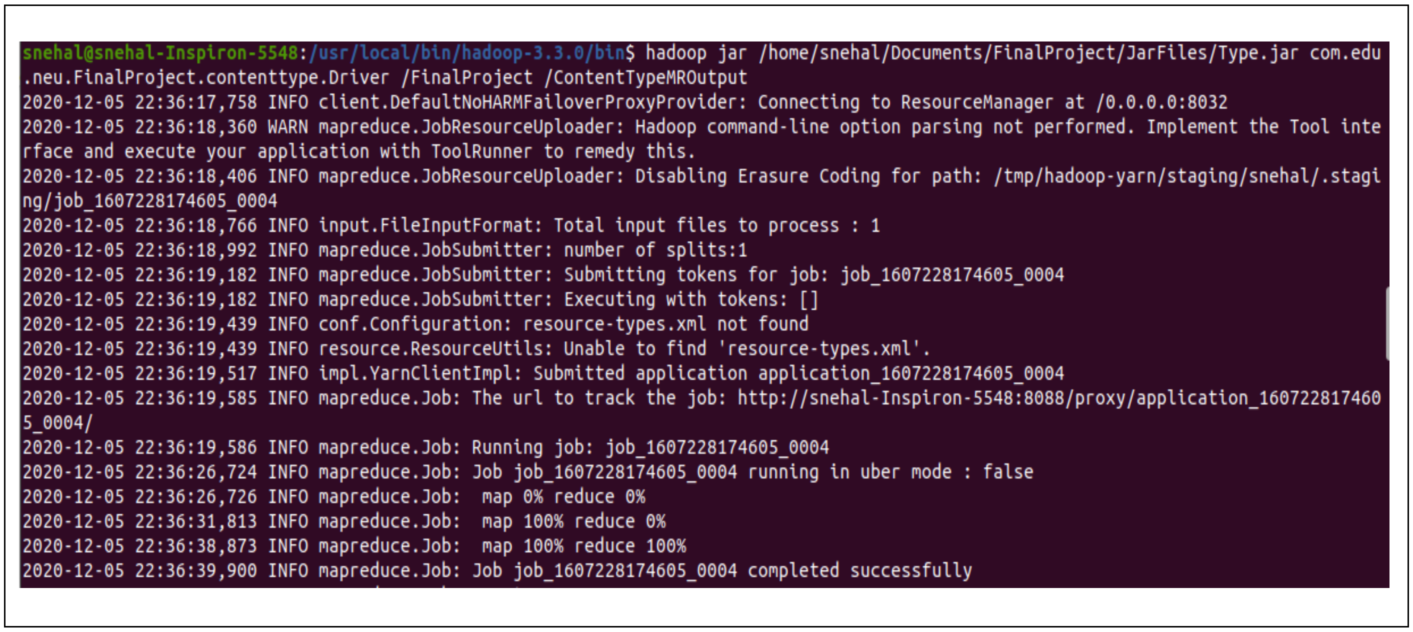
Showing 1 to 1 of 1 entries

Hadoop, 2020.

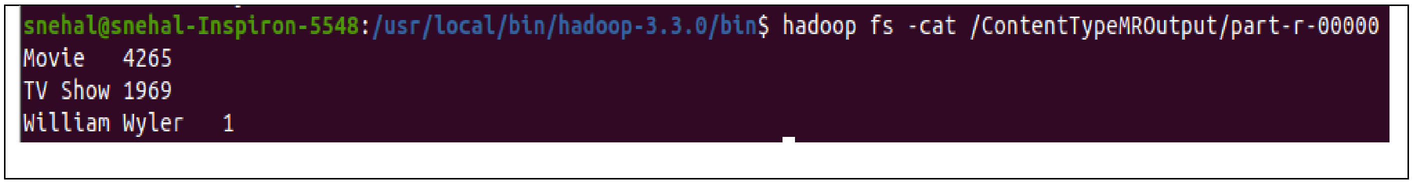
Analysis 03 – Count number of Total Movies and TV shows in dataset

In this Analysis, Total number of Movies and TV shows in the dataset are counted using Hadoop MapReduce.

- Write Map, Reduce and Driver functions
- Create an executable jar file
- Run the jar file using Hadoop jar command
- Display the output using Hadoop cat command



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop jar /home/snehal/Documents/FinalProject/JarFiles/Type.jar com.edu
.neu.FinalProject.contenttype.Driver /FinalProject /ContentTypeMROutput
2020-12-05 22:36:17,758 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-05 22:36:18,360 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
rface and execute your application with ToolRunner to remedy this.
2020-12-05 22:36:18,406 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/snehal/.stagi
ng/job_1607228174605_0004
2020-12-05 22:36:18,766 INFO input.FileInputFormat: Total input files to process : 1
2020-12-05 22:36:18,992 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-05 22:36:19,182 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607228174605_0004
2020-12-05 22:36:19,182 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-05 22:36:19,439 INFO conf.Configuration: resource-types.xml not found
2020-12-05 22:36:19,439 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-05 22:36:19,517 INFO impl.YarnClientImpl: Submitted application application_1607228174605_0004
2020-12-05 22:36:19,585 INFO mapreduce.Job: The url to track the job: http://snehal-Inspiron-5548:8088/proxy/application_160722817460
5_0004/
2020-12-05 22:36:19,586 INFO mapreduce.Job: Running job: job_1607228174605_0004
2020-12-05 22:36:26,724 INFO mapreduce.Job: Job job_1607228174605_0004 running in uber mode : false
2020-12-05 22:36:26,726 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-05 22:36:31,813 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-05 22:36:38,873 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-05 22:36:39,900 INFO mapreduce.Job: Job job_1607228174605_0004 completed successfully
```

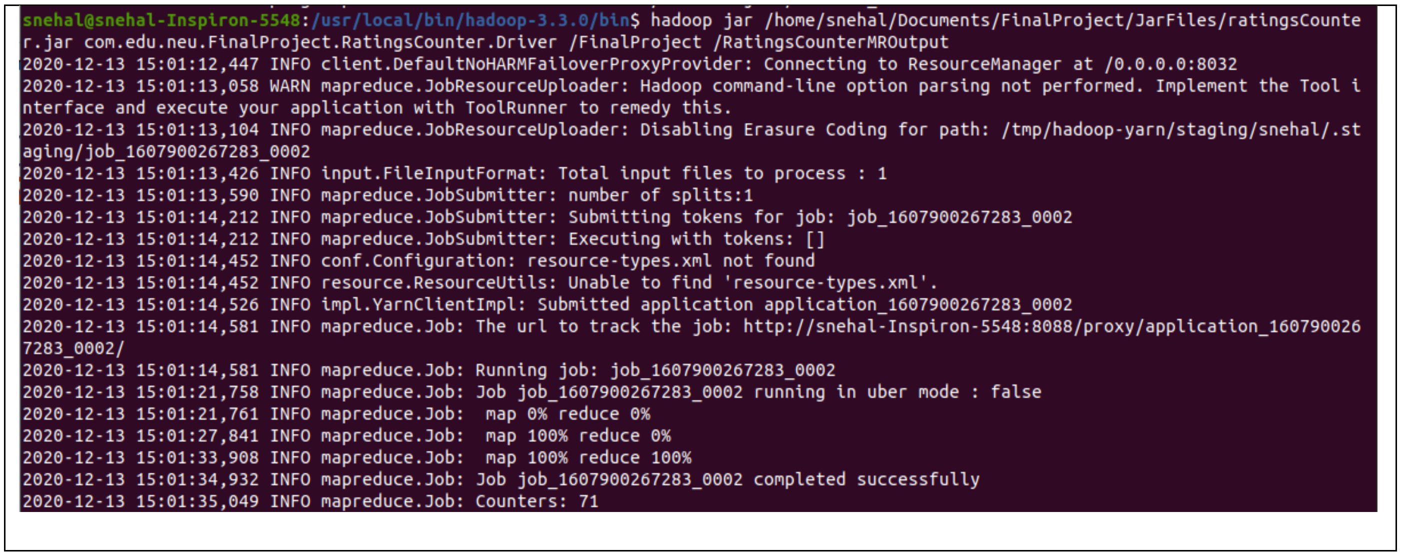


```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -cat /ContentTypeMROutput/part-r-00000
Movie	4265
TV Show 1969
William Wyler	1
```

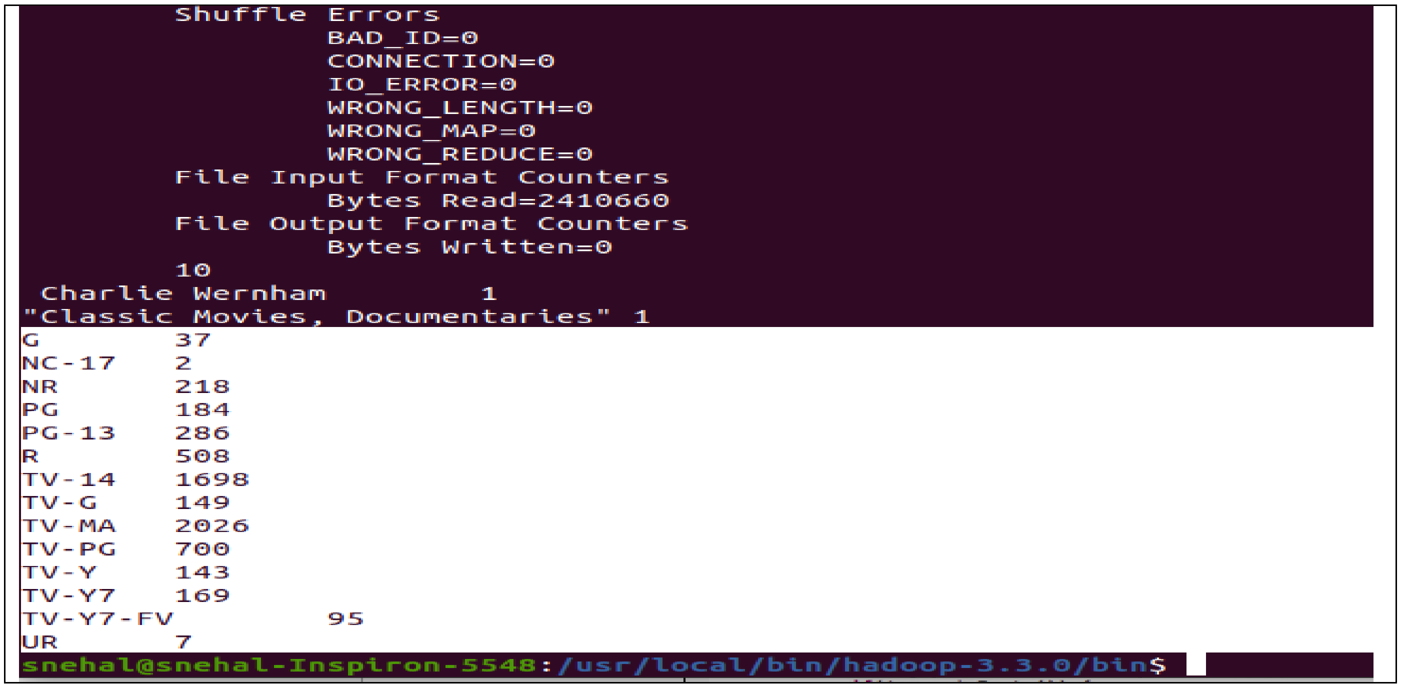
Analysis 04 - Movies and TV show analysis based on ratings using custom counter

Custom Counters : Hadoop allows to create Custom Counters to count or summation metrics from the dataset. The counting is done in Map phase only so no need of Reduce function.

- Write Map and Driver functions (Appendix section)
- Create an executable jar file
- Run the jar file using Hadoop jar command
- Result is displayed in the CLI after the jar is executed successfully.

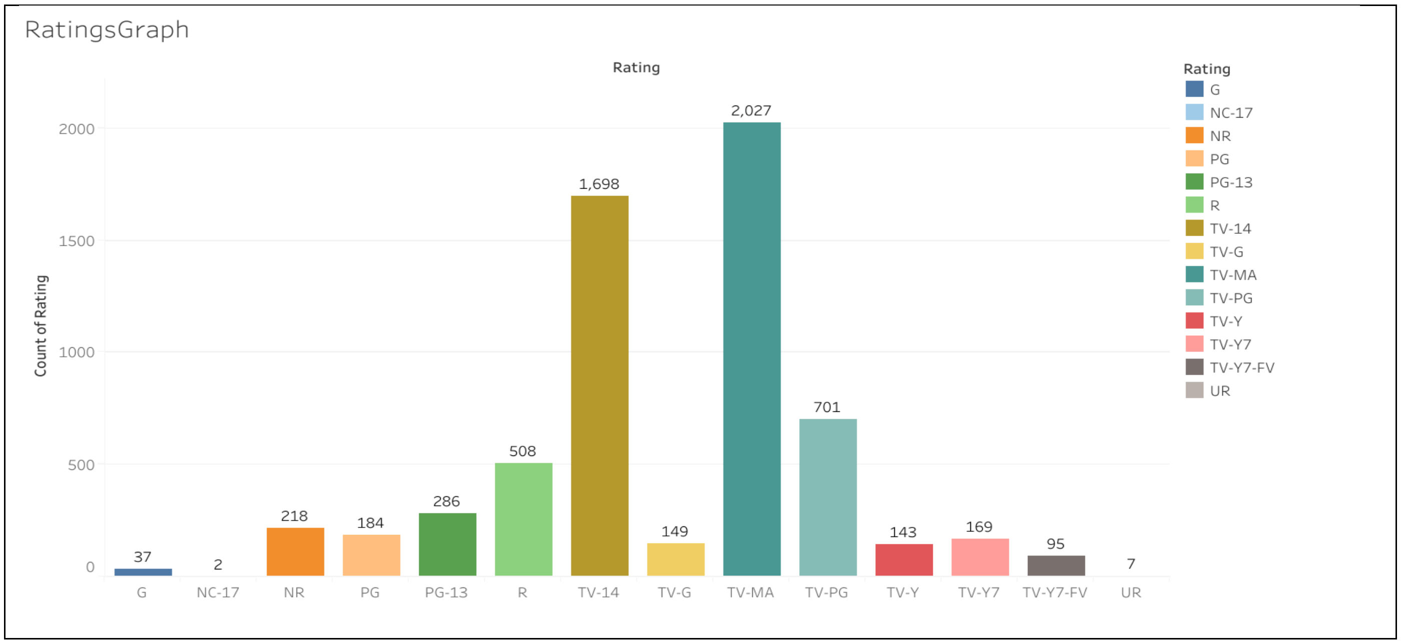


```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop jar /home/snehal/Documents/FinalProject/JarFiles/ratingsCounte
r.jar com.edu.neu.FinalProject.RatingsCounter.Driver /FinalProject /RatingsCounterMROutput
2020-12-13 15:01:12,447 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-13 15:01:13,058 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool i
nterface and execute your application with ToolRunner to remedy this.
2020-12-13 15:01:13,104 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/snehal/.st
aging/job_1607900267283_0002
2020-12-13 15:01:13,426 INFO input.FileInputFormat: Total input files to process : 1
2020-12-13 15:01:13,590 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-13 15:01:14,212 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607900267283_0002
2020-12-13 15:01:14,212 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-13 15:01:14,452 INFO conf.Configuration: resource-types.xml not found
2020-12-13 15:01:14,452 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-13 15:01:14,526 INFO impl.YarnClientImpl: Submitted application application_1607900267283_0002
2020-12-13 15:01:14,581 INFO mapreduce.Job: The url to track the job: http://snehal-Inspiron-5548:8088/proxy/application_160790026
7283_0002/
2020-12-13 15:01:14,581 INFO mapreduce.Job: Running job: job_1607900267283_0002
2020-12-13 15:01:21,758 INFO mapreduce.Job: Job job_1607900267283_0002 running in uber mode : false
2020-12-13 15:01:21,761 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-13 15:01:27,841 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-13 15:01:33,908 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-13 15:01:34,932 INFO mapreduce.Job: Job job_1607900267283_0002 completed successfully
2020-12-13 15:01:35,049 INFO mapreduce.Job: Counters: 71
```



```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2410660
        File Output Format Counters
                Bytes Written=0
        10
 Charlie Wernham         1
"Classic Movies, Documentaries"  1
G       37
NC-17   2
NR      218
PG      184
PG-13   286
R       508
TV-14   1698
TV-G    149
TV-MA   2026
TV-PG   700
TV-Y    143
TV-Y7   169
TV-Y7-FV        95
UR      7
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$
```

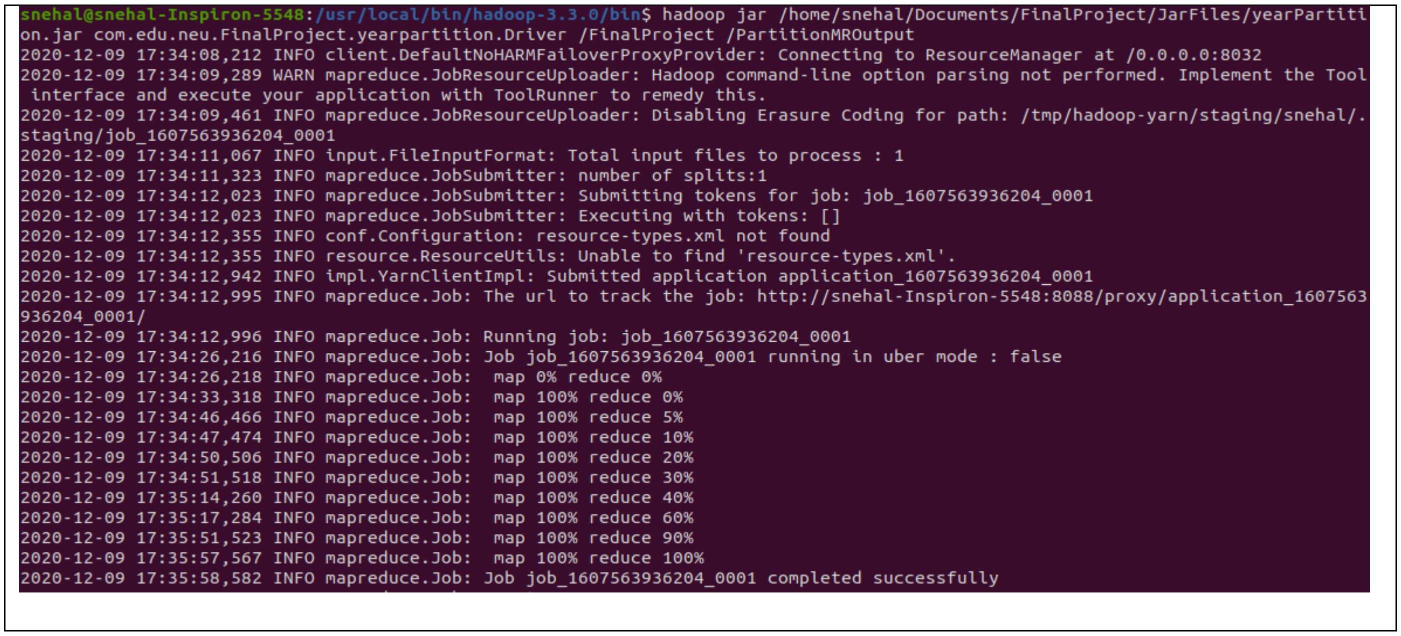
Data visualization using Tableau :



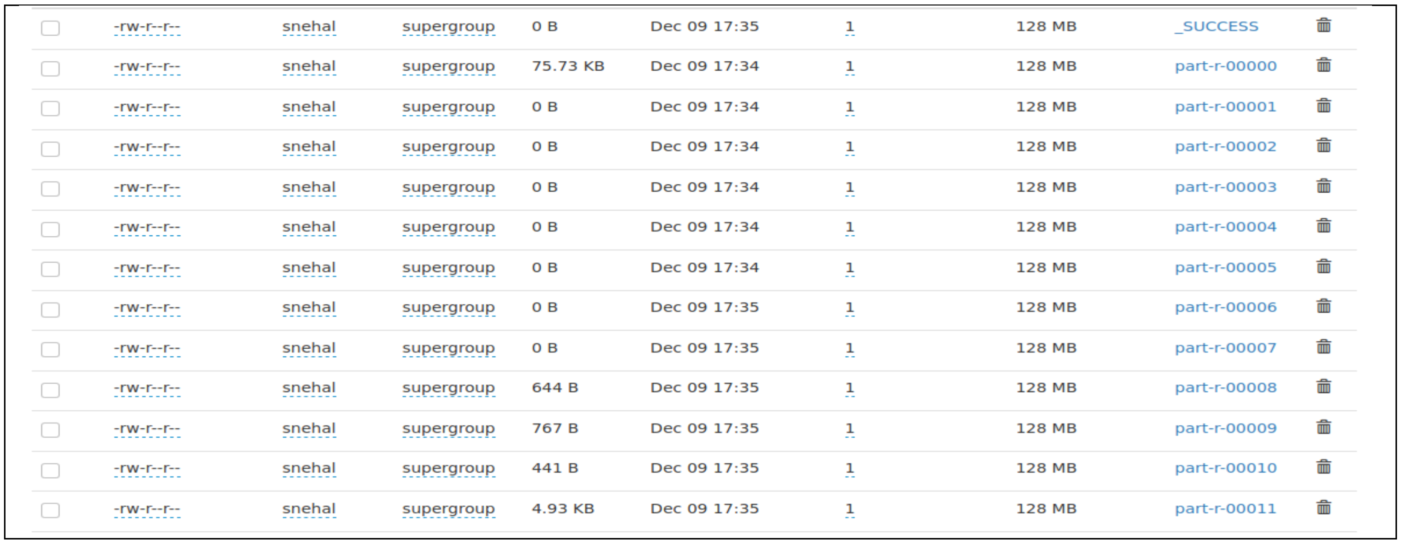
Analysis 05 – Implement partitioning on the basis on year the Movies and TV shows added in Netflix dataset

In this Analysis, one of the Data Organization Technique, Partitioning Pattern is used.
HashPartitioner is used to partition data based on year the Movies and TV shows were added in the dataset.
Year is extracted from date_added column in the dataset.

- Write Map, Reduce, Partitioner and Driver functions
- Create an executable jar file
- Run the jar file using Hadoop jar command
- Check the HDFS UI for Partitions



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop jar /home/snehal/Documents/FinalProject/JarFiles/yearPartiti
on.jar com.edu.neu.FinalProject.yearpartition.Driver /FinalProject /PartitionMROutput
2020-12-09 17:34:08,212 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-09 17:34:09,289 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool
 interface and execute your application with ToolRunner to remedy this.
2020-12-09 17:34:09,461 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/snehal/.
staging/job_1607563936204_0001
2020-12-09 17:34:11,067 INFO input.FileInputFormat: Total input files to process : 1
2020-12-09 17:34:11,323 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-09 17:34:12,023 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607563936204_0001
2020-12-09 17:34:12,023 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-09 17:34:12,355 INFO conf.Configuration: resource-types.xml not found
2020-12-09 17:34:12,355 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-09 17:34:12,942 INFO impl.YarnClientImpl: Submitted application application_1607563936204_0001
2020-12-09 17:34:12,995 INFO mapreduce.Job: The url to track the job: http://snehal-Inspiron-5548:8088/proxy/application_1607563
936204_0001/
2020-12-09 17:34:12,996 INFO mapreduce.Job: Running job: job_1607563936204_0001
2020-12-09 17:34:26,216 INFO mapreduce.Job: Job job_1607563936204_0001 running in uber mode : false
2020-12-09 17:34:26,218 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-09 17:34:33,318 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-09 17:34:46,466 INFO mapreduce.Job:  map 100% reduce 5%
2020-12-09 17:34:47,474 INFO mapreduce.Job:  map 100% reduce 10%
2020-12-09 17:34:50,506 INFO mapreduce.Job:  map 100% reduce 20%
2020-12-09 17:34:51,518 INFO mapreduce.Job:  map 100% reduce 30%
2020-12-09 17:35:14,260 INFO mapreduce.Job:  map 100% reduce 40%
2020-12-09 17:35:17,284 INFO mapreduce.Job:  map 100% reduce 60%
2020-12-09 17:35:51,523 INFO mapreduce.Job:  map 100% reduce 90%
2020-12-09 17:35:57,567 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-09 17:35:58,582 INFO mapreduce.Job: Job job_1607563936204_0001 completed successfully
```



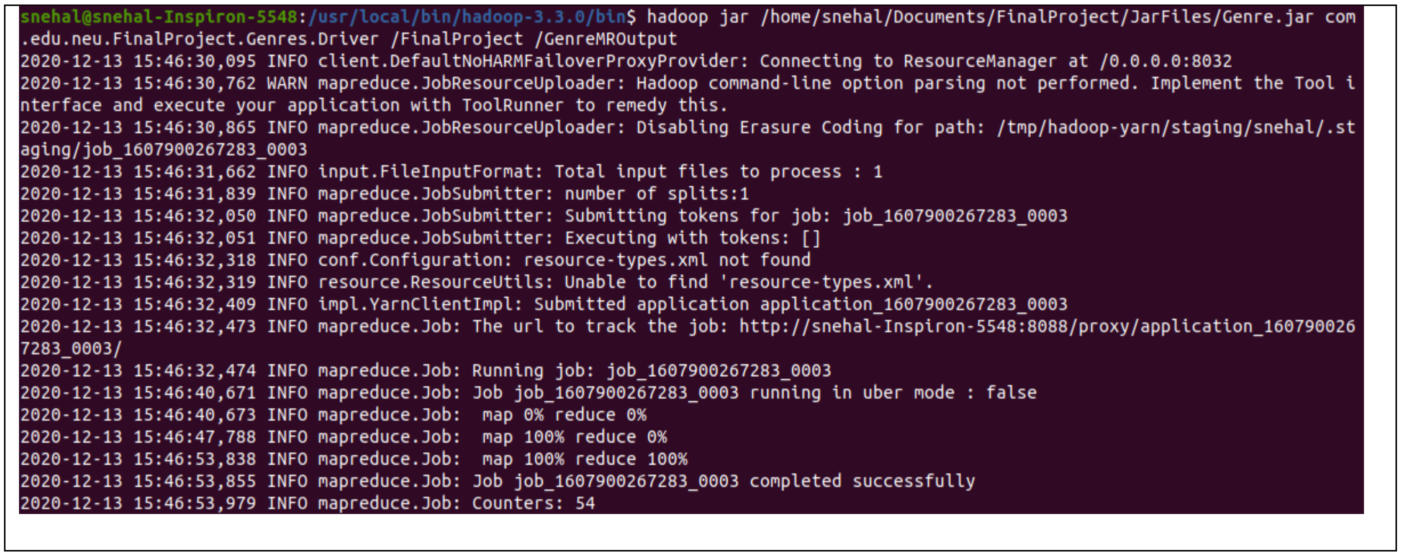
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:35	1	128 MB	_SUCCESS	🗑
☐	-rw-r--r--	snehal	supergroup	75.73 KB	Dec 09 17:34	1	128 MB	part-r-00000	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:34	1	128 MB	part-r-00001	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:34	1	128 MB	part-r-00002	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:34	1	128 MB	part-r-00003	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:34	1	128 MB	part-r-00004	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:34	1	128 MB	part-r-00005	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:35	1	128 MB	part-r-00006	🗑
☐	-rw-r--r--	snehal	supergroup	0 B	Dec 09 17:35	1	128 MB	part-r-00007	🗑
☐	-rw-r--r--	snehal	supergroup	644 B	Dec 09 17:35	1	128 MB	part-r-00008	🗑
☐	-rw-r--r--	snehal	supergroup	767 B	Dec 09 17:35	1	128 MB	part-r-00009	🗑
☐	-rw-r--r--	snehal	supergroup	441 B	Dec 09 17:35	1	128 MB	part-r-00010	🗑
☐	-rw-r--r--	snehal	supergroup	4.93 KB	Dec 09 17:35	1	128 MB	part-r-00011	🗑

Analysis 06 – Find Distinct Genres in the dataset

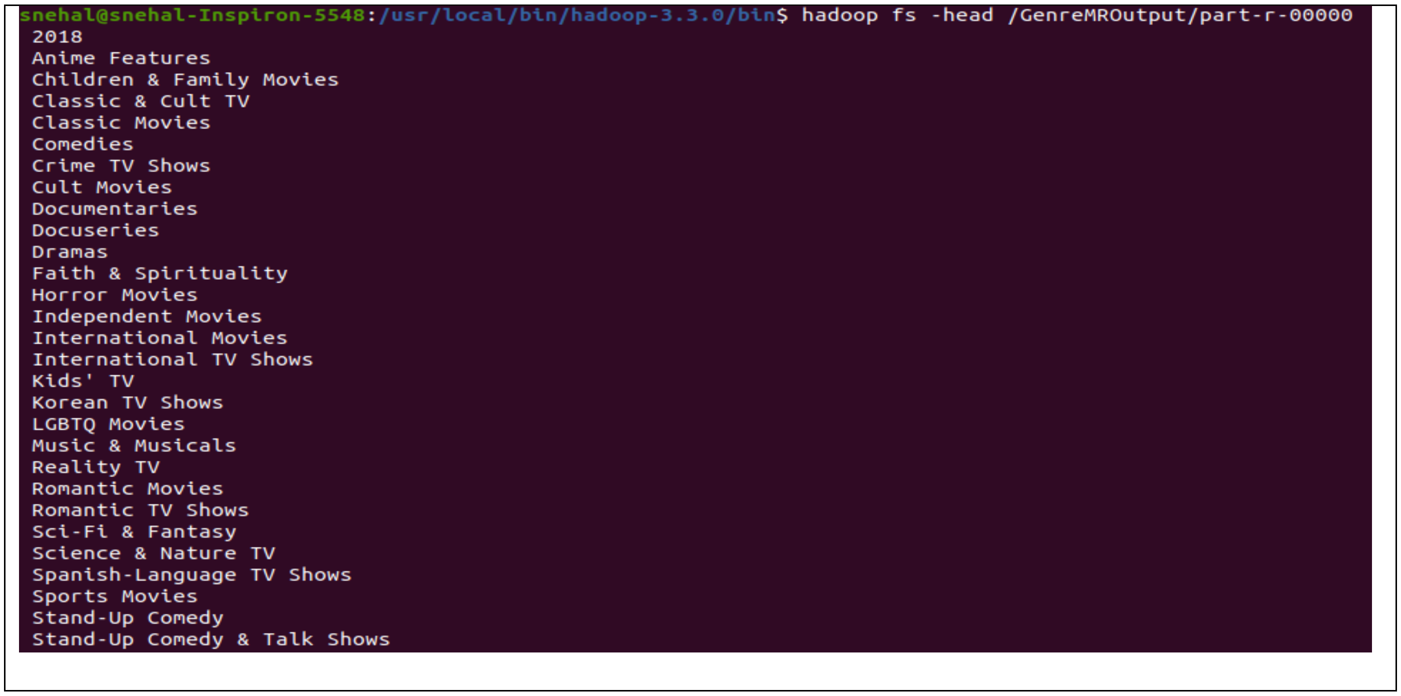
In this Analysis, one of the Filtering Pattern called Distinct Pattern is used.

This Pattern filters the whole set to generate a set of unique records.

- Write Map, Reduce and Driver functions (Appendix Section 2)
- Create an executable jar file
- Run the jar file using Hadoop jar command
- Display distinct genres using Hadoop head command



```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop jar /home/snehal/Documents/FinalProject/JarFiles/Genre.jar com
.edu.neu.FinalProject.Genres.Driver /FinalProject /GenreMROutput
2020-12-13 15:46:30,095 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-13 15:46:30,762 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool i
nterface and execute your application with ToolRunner to remedy this.
2020-12-13 15:46:30,865 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/snehal/.st
aging/job_1607900267283_0003
2020-12-13 15:46:31,662 INFO input.FileInputFormat: Total input files to process : 1
2020-12-13 15:46:31,839 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-13 15:46:32,050 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607900267283_0003
2020-12-13 15:46:32,051 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-13 15:46:32,318 INFO conf.Configuration: resource-types.xml not found
2020-12-13 15:46:32,319 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-13 15:46:32,409 INFO impl.YarnClientImpl: Submitted application application_1607900267283_0003
2020-12-13 15:46:32,473 INFO mapreduce.Job: The url to track the job: http://snehal-Inspiron-5548:8088/proxy/application_160790026
7283_0003/
2020-12-13 15:46:32,474 INFO mapreduce.Job: Running job: job_1607900267283_0003
2020-12-13 15:46:40,671 INFO mapreduce.Job: Job job_1607900267283_0003 running in uber mode : false
2020-12-13 15:46:40,673 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-13 15:46:47,788 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-13 15:46:53,838 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-13 15:46:53,855 INFO mapreduce.Job: Job job_1607900267283_0003 completed successfully
2020-12-13 15:46:53,979 INFO mapreduce.Job: Counters: 54
```

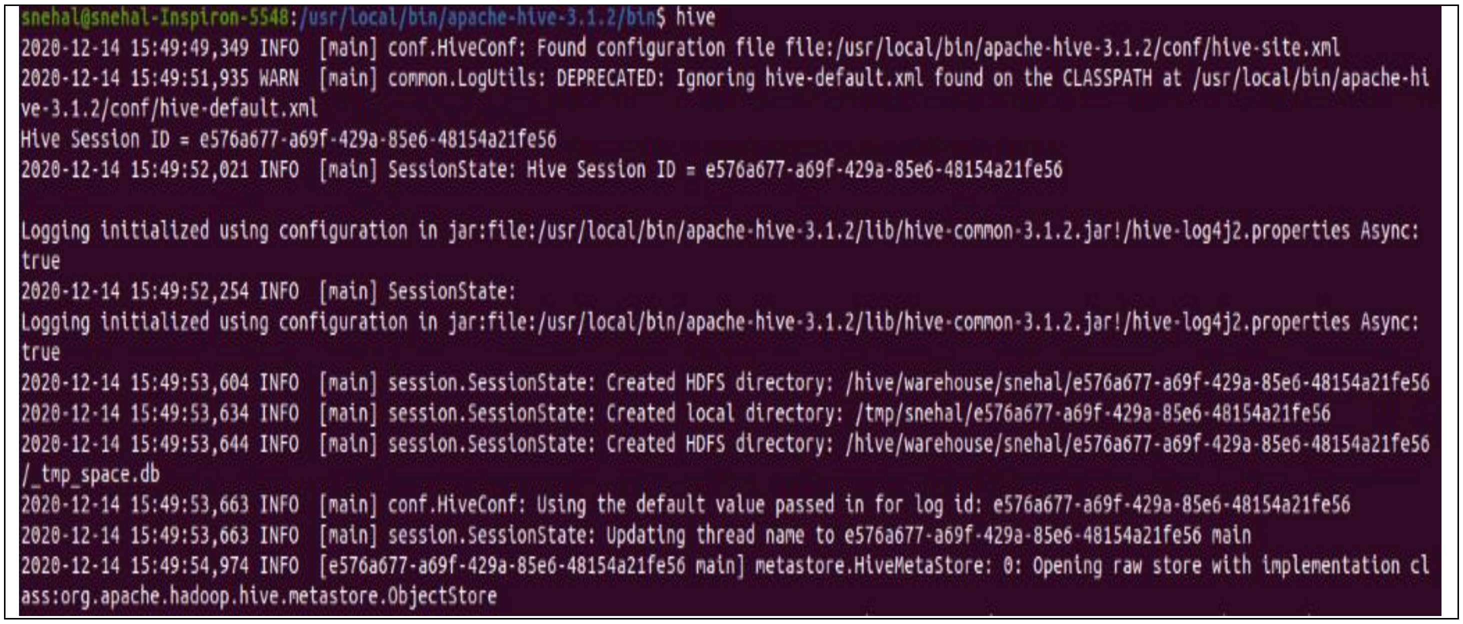


```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -head /GenreMROutput/part-r-00000
 2018
 Anime Features
 Children & Family Movies
 Classic & Cult TV
 Classic Movies
 Comedies
 Crime TV Shows
 Cult Movies
 Documentaries
 Docuseries
 Dramas
 Faith & Spirituality
 Horror Movies
 Independent Movies
 International Movies
 International TV Shows
 Kids' TV
 Korean TV Shows
 LGBTQ Movies
 Music & Musicals
 Reality TV
 Romantic Movies
 Romantic TV Shows
 Sci-Fi & Fantasy
 Science & Nature TV
 Spanish-Language TV Shows
 Sports Movies
 Stand-Up Comedy
 Stand-Up Comedy & Talk Shows
```

HIVE ANALYSIS

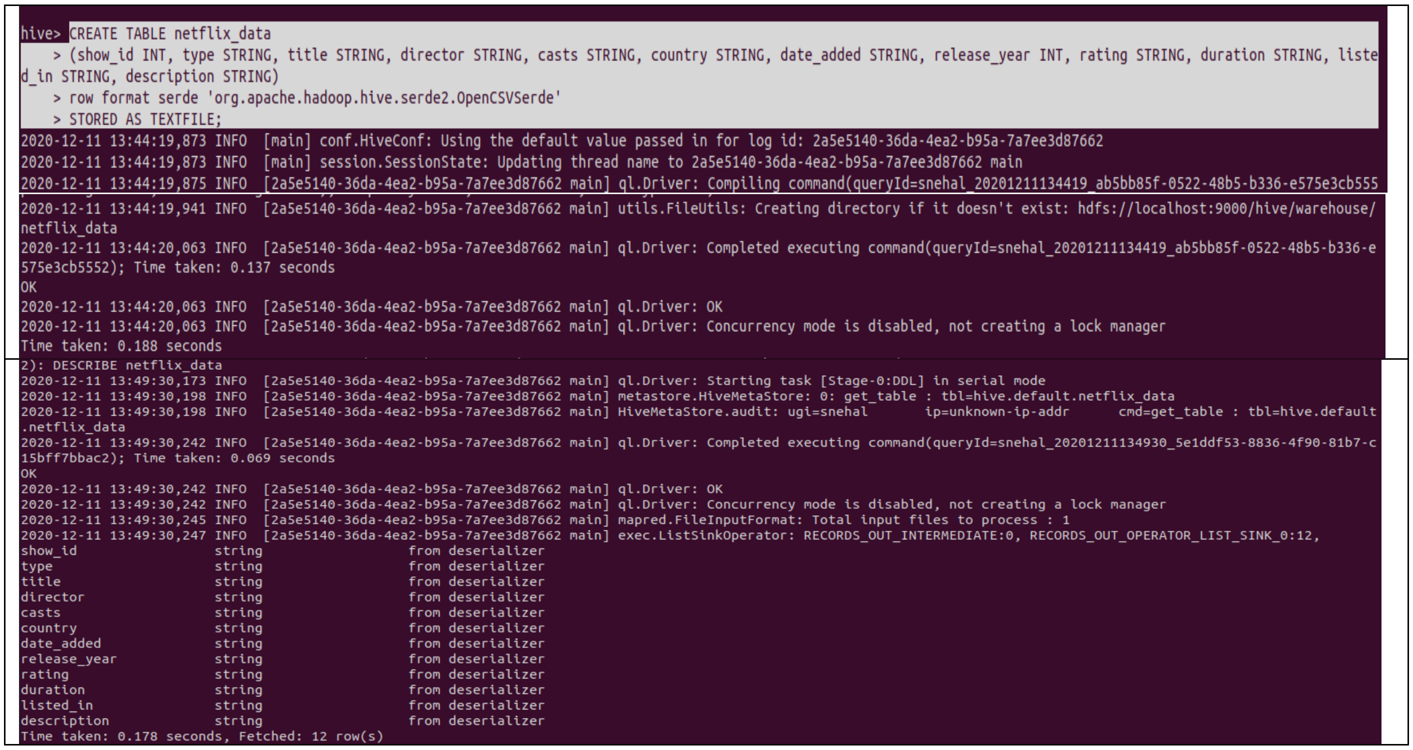
Steps to start HIVE shell

1. Navigate to Hive Directory
   cd /usr/local/bin/apache-hive-3.1.2/bin
2. Start hive shell
   hive



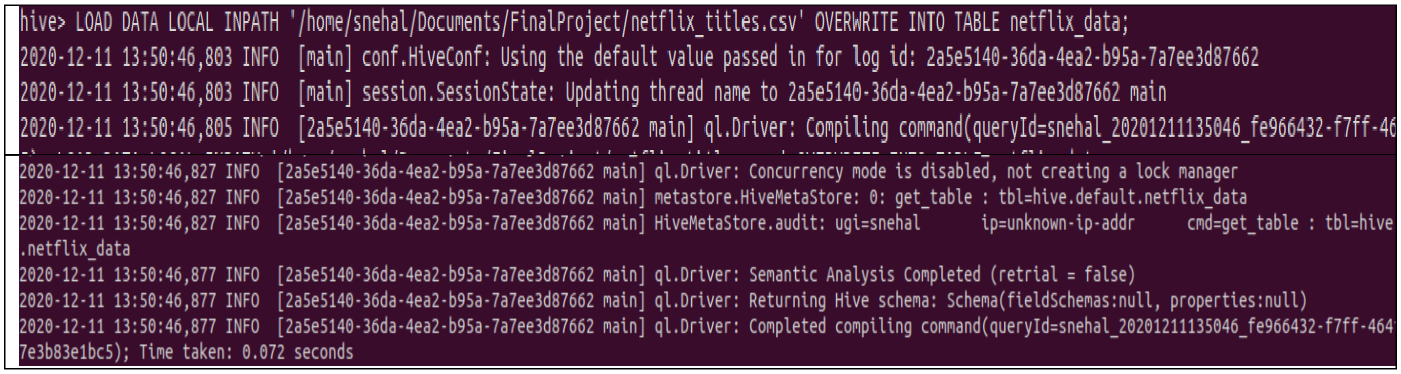
```
snehal@snehal-Inspiron-5548:/usr/local/bin/apache-hive-3.1.2/bin$ hive
2020-12-14 15:49:49,349 INFO  [main] conf.HiveConf: Found configuration file file:/usr/local/bin/apache-hive-3.1.2/conf/hive-site.xml
2020-12-14 15:49:51,935 WARN  [main] common.LogUtils: DEPRECATED: Ignoring hive-default.xml found on the CLASSPATH at /usr/local/bin/apache-hi
ve-3.1.2/conf/hive-default.xml
Hive Session ID = e576a677-a69f-429a-85e6-48154a21fe56
2020-12-14 15:49:52,021 INFO  [main] SessionState: Hive Session ID = e576a677-a69f-429a-85e6-48154a21fe56

Logging initialized using configuration in jar:file:/usr/local/bin/apache-hive-3.1.2/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async:
true
2020-12-14 15:49:52,254 INFO  [main] SessionState:
Logging initialized using configuration in jar:file:/usr/local/bin/apache-hive-3.1.2/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async:
true
2020-12-14 15:49:53,604 INFO  [main] session.SessionState: Created HDFS directory: /hive/warehouse/snehal/e576a677-a69f-429a-85e6-48154a21fe56
2020-12-14 15:49:53,634 INFO  [main] session.SessionState: Created local directory: /tmp/snehal/e576a677-a69f-429a-85e6-48154a21fe56
2020-12-14 15:49:53,644 INFO  [main] session.SessionState: Created HDFS directory: /hive/warehouse/snehal/e576a677-a69f-429a-85e6-48154a21fe56
/_tmp_space.db
2020-12-14 15:49:53,663 INFO  [main] conf.HiveConf: Using the default value passed in for log id: e576a677-a69f-429a-85e6-48154a21fe56
2020-12-14 15:49:53,663 INFO  [main] session.SessionState: Updating thread name to e576a677-a69f-429a-85e6-48154a21fe56 main
2020-12-14 15:49:54,974 INFO  [e576a677-a69f-429a-85e6-48154a21fe56 main] metastore.HiveMetaStore: 0: Opening raw store with implementation cl
ass:org.apache.hadoop.hive.metastore.ObjectStore
```

3. Create netflix_data table



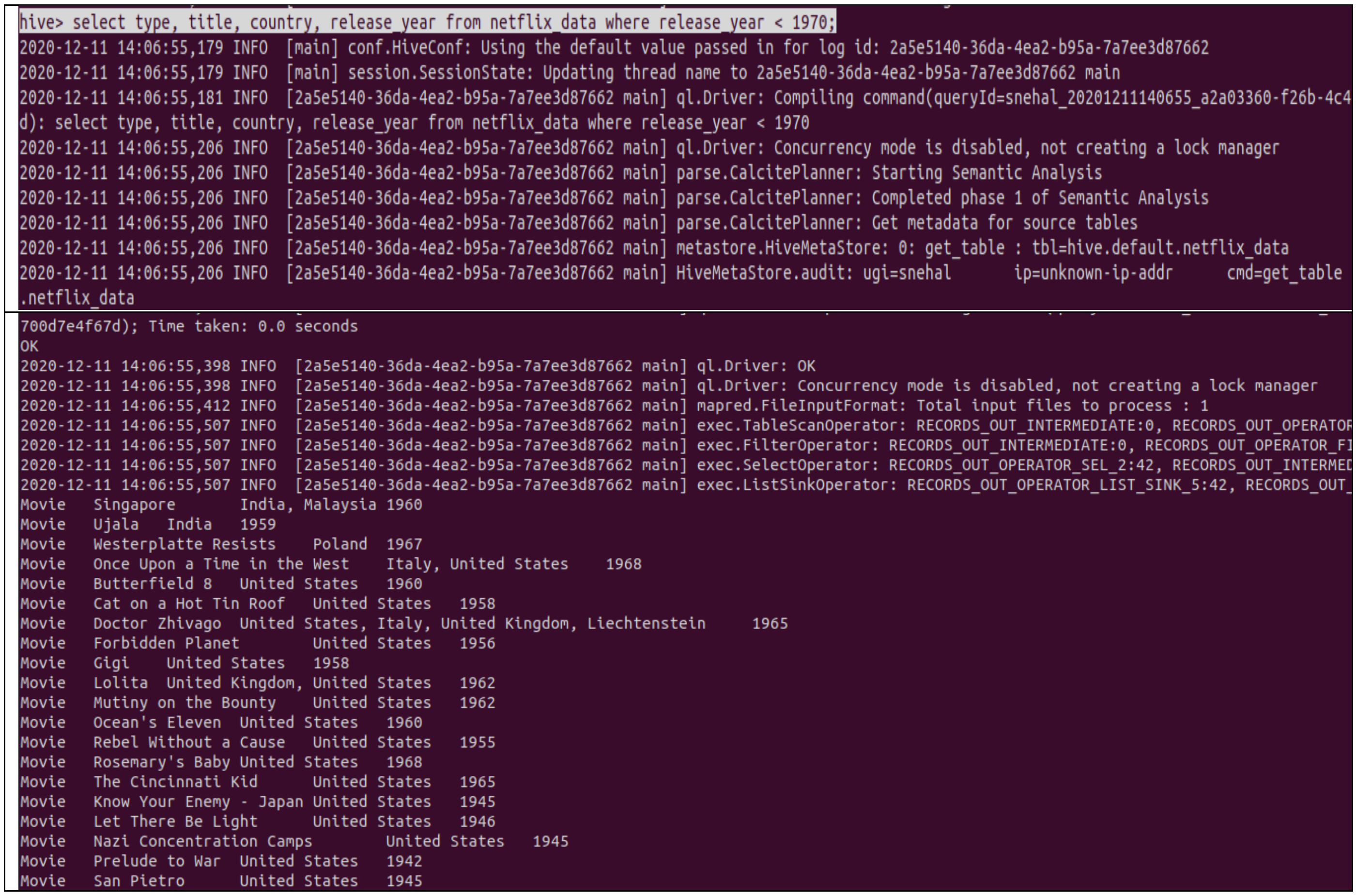
```
hive> CREATE TABLE netflix_data
    > (show_id INT, type STRING, title STRING, director STRING, casts STRING, country STRING, date_added STRING, release_year INT, rating STRING, duration STRING, liste
d_in STRING, description STRING)
    > row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > STORED AS TEXTFILE;
2020-12-11 13:44:19,873 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 13:44:19,873 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 13:44:19,875 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211134419_ab5bb85f-0522-48b5-b336-e575e3cb555
2020-12-11 13:44:19,941 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] utils.FileUtils: Creating directory if it doesn't exist: hdfs://localhost:9000/hive/warehouse/
netflix_data
2020-12-11 13:44:20,063 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Completed executing command(queryId=snehal_20201211134419_ab5bb85f-0522-48b5-b336-e
575e3cb5552); Time taken: 0.137 seconds
OK
2020-12-11 13:44:20,063 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: OK
2020-12-11 13:44:20,063 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
Time taken: 0.188 seconds
2): DESCRIBE netflix_data
2020-12-11 13:49:30,173 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Starting task [Stage-0:DDL] in serial mode
2020-12-11 13:49:30,198 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: get_table : tbl=hive.default.netflix_data
2020-12-11 13:49:30,198 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] HiveMetaStore.audit: ugi=snehal       ip=unknown-ip-addr      cmd=get_table : tbl=hive.default
.netflix_data
2020-12-11 13:49:30,242 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Completed executing command(queryId=snehal_20201211134930_5e1ddf53-8836-4f90-81b7-c
15bff7bbac2); Time taken: 0.069 seconds
OK
2020-12-11 13:49:30,242 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: OK
2020-12-11 13:49:30,242 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 13:49:30,245 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] mapred.FileInputFormat: Total input files to process : 1
2020-12-11 13:49:30,247 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.ListSinkOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_LIST_SINK_0:12,
show_id                 string                  from deserializer
type                    string                  from deserializer
title                   string                  from deserializer
director                string                  from deserializer
casts                   string                  from deserializer
country                 string                  from deserializer
date_added              string                  from deserializer
release_year            string                  from deserializer
rating                  string                  from deserializer
duration                string                  from deserializer
listed_in               string                  from deserializer
description             string                  from deserializer
Time taken: 0.178 seconds, Fetched: 12 row(s)
```

4. Load csv Data into Netflix_data table
   LOAD DATA LOCAL INPATH '/home/snehal/Documents/FinalProject/Netflix_titles.csv' OVERWRITE INTO TABLE Netflix_data;



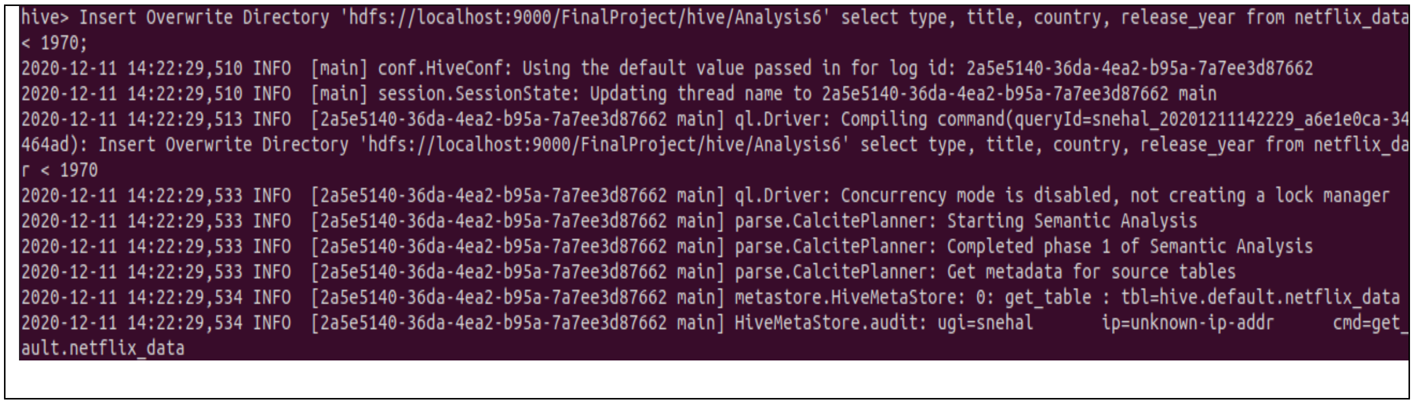
```
hive> LOAD DATA LOCAL INPATH '/home/snehal/Documents/FinalProject/netflix_titles.csv' OVERWRITE INTO TABLE netflix_data;
2020-12-11 13:50:46,803 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 13:50:46,803 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 13:50:46,805 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211135046_fe966432-f7ff-46
2020-12-11 13:50:46,827 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 13:50:46,827 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: get_table : tbl=hive.default.netflix_data
2020-12-11 13:50:46,827 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] HiveMetaStore.audit: ugi=snehal       ip=unknown-ip-addr      cmd=get_table : tbl=hive
.netflix_data
2020-12-11 13:50:46,877 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Semantic Analysis Completed (retrial = false)
2020-12-11 13:50:46,877 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Returning Hive schema: Schema(fieldSchemas:null, properties:null)
2020-12-11 13:50:46,877 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Completed compiling command(queryId=snehal_20201211135046_fe966432-f7ff-464
7e3b83e1bc5); Time taken: 0.072 seconds
```

Analysis 07 - Get movies and TV shows which are released before year 1970 using where clause

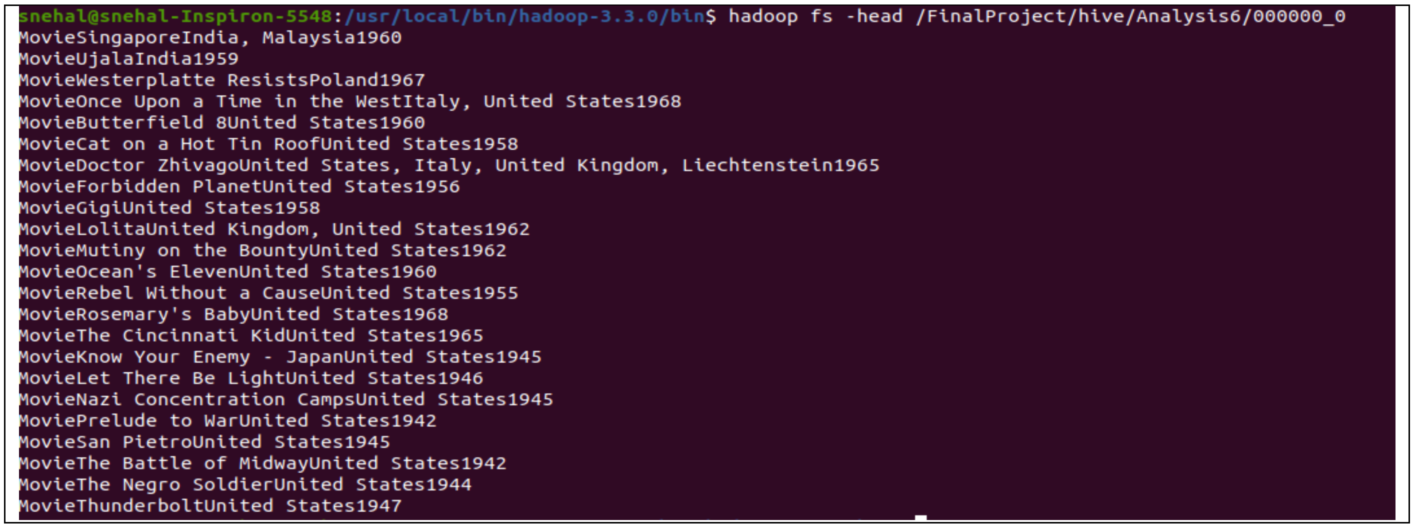


```
hive> select type, title, country, release_year from netflix_data where release_year < 1970;
2020-12-11 14:06:55,179 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 14:06:55,179 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 14:06:55,181 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211140655_a2a03360-f26b-4c4
d): select type, title, country, release_year from netflix_data where release_year < 1970
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Starting Semantic Analysis
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Completed phase 1 of Semantic Analysis
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Get metadata for source tables
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: get_table : tbl=hive.default.netflix_data
2020-12-11 14:06:55,206 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] HiveMetaStore.audit: ugi=snehal       ip=unknown-ip-addr      cmd=get_table
.netflix_data
700d7e4f67d); Time taken: 0.0 seconds
OK
2020-12-11 14:06:55,398 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: OK
2020-12-11 14:06:55,398 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 14:06:55,412 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] mapred.FileInputFormat: Total input files to process : 1
2020-12-11 14:06:55,507 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.TableScanOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR
2020-12-11 14:06:55,507 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.FilterOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_FI
2020-12-11 14:06:55,507 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.SelectOperator: RECORDS_OUT_OPERATOR_SEL_2:42, RECORDS_OUT_INTERMED
2020-12-11 14:06:55,507 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.ListSinkOperator: RECORDS_OUT_OPERATOR_LIST_SINK_5:42, RECORDS_OUT_
Movie   Singapore       India, Malaysia 1960
Movie   Ujala   India   1959
Movie   Westerplatte Resists    Poland  1967
Movie   Once Upon a Time in the West    Italy, United States    1968
Movie   Butterfield 8   United States   1960
Movie   Cat on a Hot Tin Roof   United States   1958
Movie   Doctor Zhivago  United States, Italy, United Kingdom, Liechtenstein     1965
Movie   Forbidden Planet        United States   1956
Movie   Gigi    United States   1958
Movie   Lolita  United Kingdom, United States   1962
Movie   Mutiny on the Bounty    United States   1962
Movie   Ocean's Eleven  United States   1960
Movie   Rebel Without a Cause   United States   1955
Movie   Rosemary's Baby United States   1968
Movie   The Cincinnati Kid      United States   1965
Movie   Know Your Enemy - Japan United States   1945
Movie   Let There Be Light      United States   1946
Movie   Nazi Concentration Camps        United States   1945
Movie   Prelude to War  United States   1942
Movie   San Pietro      United States   1945
```

Insert Hive output to HDFS using INSERT Overwrite Directory command

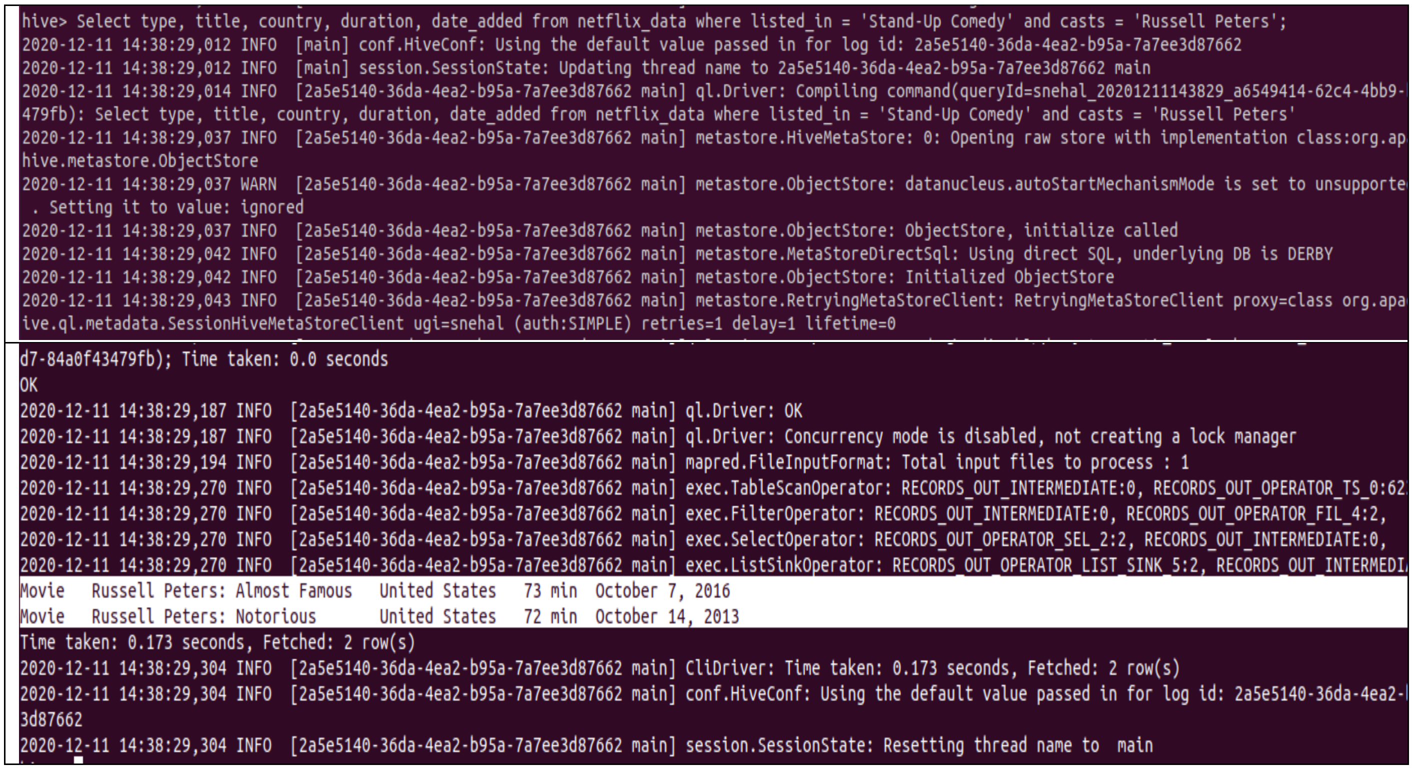


```
hive> Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/hive/Analysis6' select type, title, country, release_year from netflix_data
< 1970;
2020-12-11 14:22:29,510 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 14:22:29,510 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 14:22:29,513 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211142229_a6e1e0ca-34
464ad): Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/hive/Analysis6' select type, title, country, release_year from netflix_da
r < 1970
2020-12-11 14:22:29,533 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 14:22:29,533 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Starting Semantic Analysis
2020-12-11 14:22:29,533 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Completed phase 1 of Semantic Analysis
2020-12-11 14:22:29,533 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Get metadata for source tables
2020-12-11 14:22:29,534 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: get_table : tbl=hive.default.netflix_data
2020-12-11 14:22:29,534 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] HiveMetaStore.audit: ugi=snehal       ip=unknown-ip-addr      cmd=get_
ault.netflix_data
```



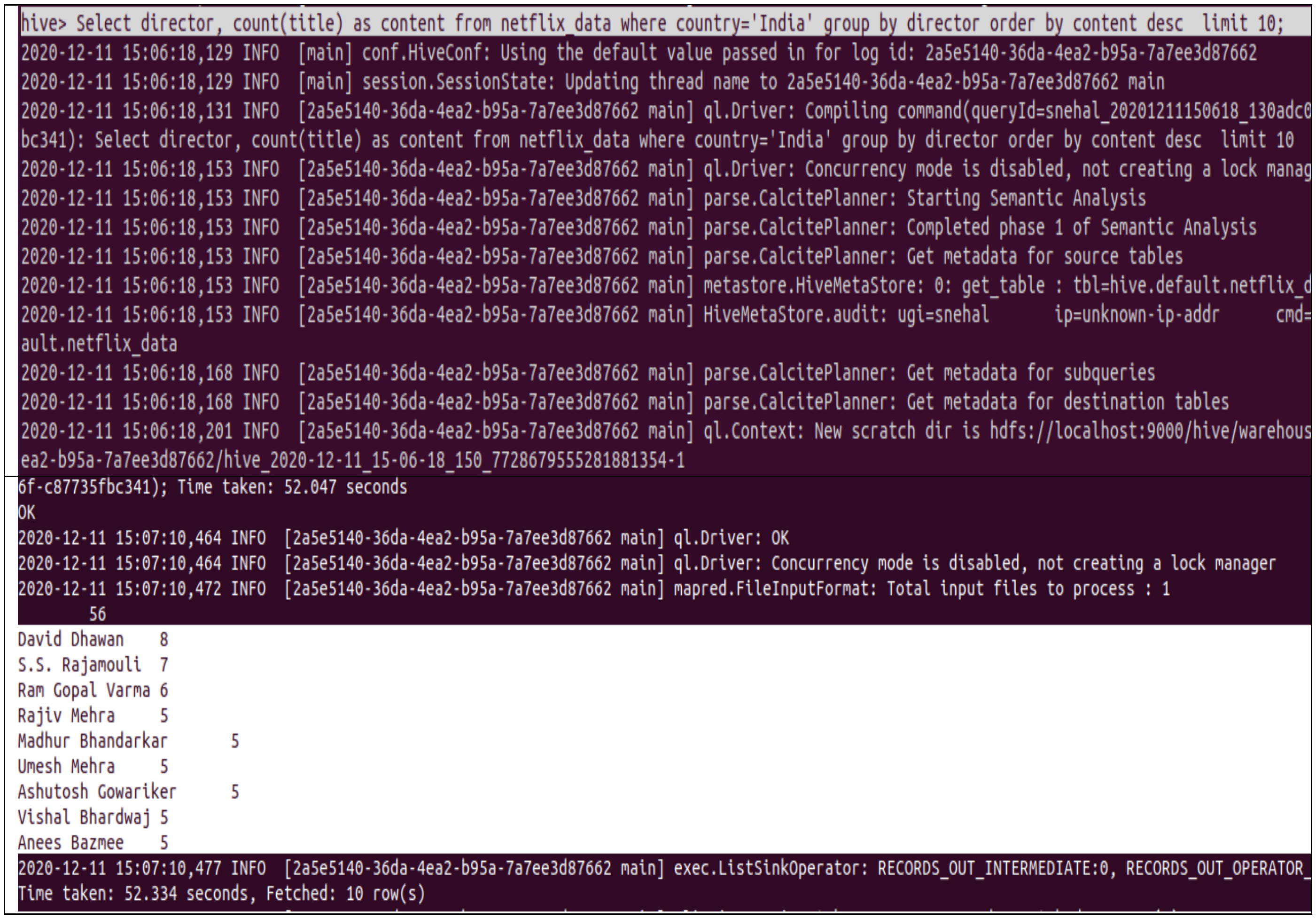
```
snehal@snehal-Inspiron-5548:/usr/local/bin/hadoop-3.3.0/bin$ hadoop fs -head /FinalProject/hive/Analysis6/000000_0
MovieSingaporeIndia, Malaysia1960
MovieUjalaIndia1959
MovieWesterplatte ResistsPoland1967
MovieOnce Upon a Time in the WestItaly, United States1968
MovieButterfield 8United States1960
MovieCat on a Hot Tin RoofUnited States1958
MovieDoctor ZhivagoUnited States, Italy, United Kingdom, Liechtenstein1965
MovieForbidden PlanetUnited States1956
MovieGigiUnited States1958
MovieLolitaUnited Kingdom, United States1962
MovieMutiny on the BountyUnited States1962
MovieOcean's ElevenUnited States1960
MovieRebel Without a CauseUnited States1955
MovieRosemary's BabyUnited States1968
MovieThe Cincinnati KidUnited States1965
MovieKnow Your Enemy - JapanUnited States1945
MovieLet There Be LightUnited States1946
MovieNazi Concentration CampsUnited States1945
MoviePrelude to WarUnited States1942
MovieSan PietroUnited States1945
MovieThe Battle of MidwayUnited States1942
MovieThe Negro SoldierUnited States1944
MovieThunderboltUnited States1947
```

Analysis 08 – Find Movie or TV shows from Netflix data which are listed as "Stand-Up Comedy" and cast is "Russell Peters"



```
hive> Select type, title, country, duration, date_added from netflix_data where listed_in = 'Stand-Up Comedy' and casts = 'Russell Peters';
2020-12-11 14:38:29,012 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 14:38:29,012 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 14:38:29,014 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211143829_a6549414-62c4-4bb9-
479fb): Select type, title, country, duration, date_added from netflix_data where listed_in = 'Stand-Up Comedy' and casts = 'Russell Peters'
2020-12-11 14:38:29,037 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: Opening raw store with implementation class:org.ap
hive.metastore.ObjectStore
2020-12-11 14:38:29,037 WARN  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupporte
 . Setting it to value: ignored
2020-12-11 14:38:29,037 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.ObjectStore: ObjectStore, initialize called
2020-12-11 14:38:29,042 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is DERBY
2020-12-11 14:38:29,042 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.ObjectStore: Initialized ObjectStore
2020-12-11 14:38:29,043 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.RetryingMetaStoreClient: RetryingMetaStoreClient proxy=class org.apa
ive.ql.metadata.SessionHiveMetaStoreClient ugi=snehal (auth:SIMPLE) retries=1 delay=1 lifetime=0
d7-84a0f43479fb); Time taken: 0.0 seconds
OK
2020-12-11 14:38:29,187 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: OK
2020-12-11 14:38:29,187 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 14:38:29,194 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] mapred.FileInputFormat: Total input files to process : 1
2020-12-11 14:38:29,270 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.TableScanOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_TS_0:62
2020-12-11 14:38:29,270 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.FilterOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_FIL_4:2,
2020-12-11 14:38:29,270 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.SelectOperator: RECORDS_OUT_OPERATOR_SEL_2:2, RECORDS_OUT_INTERMEDIATE:0,
2020-12-11 14:38:29,270 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.ListSinkOperator: RECORDS_OUT_OPERATOR_LIST_SINK_5:2, RECORDS_OUT_INTERMEDI
Movie   Russell Peters: Almost Famous   United States   73 min  October 7, 2016
Movie   Russell Peters: Notorious       United States   72 min  October 14, 2013
Time taken: 0.173 seconds, Fetched: 2 row(s)
2020-12-11 14:38:29,304 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] CliDriver: Time taken: 0.173 seconds, Fetched: 2 row(s)
2020-12-11 14:38:29,304 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-
3d87662
2020-12-11 14:38:29,304 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] session.SessionState: Resetting thread name to  main
```

Analysis 09 - Find Directors from India with most contents



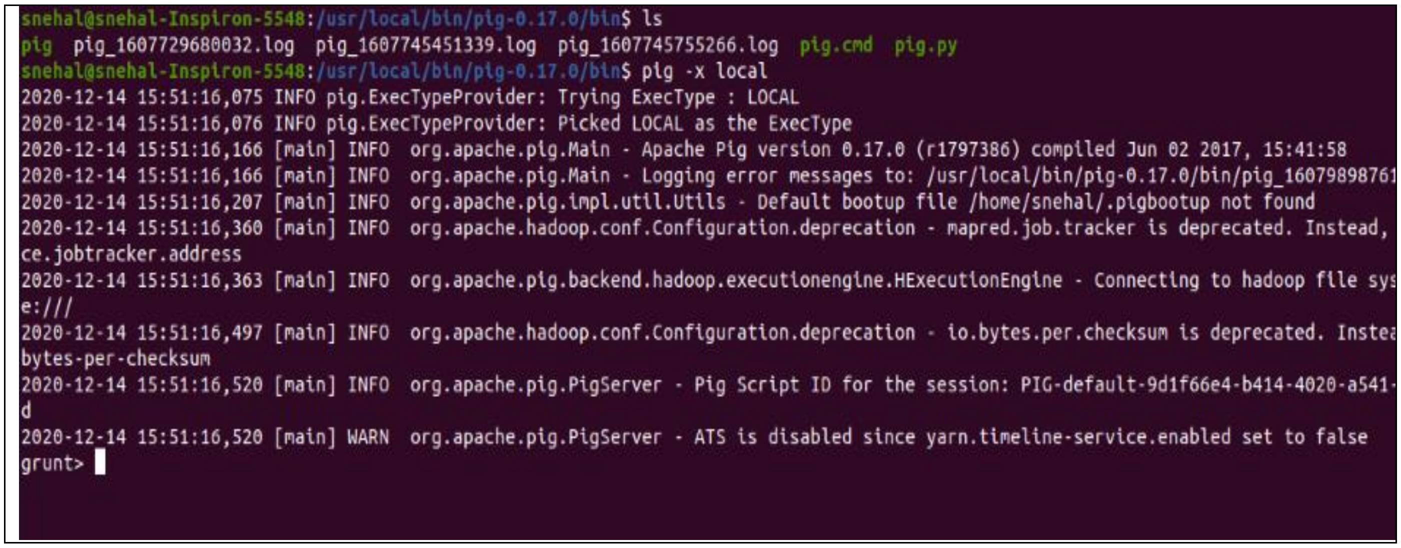
```
hive> Select director, count(title) as content from netflix_data where country='India' group by director order by content desc  limit 10;
2020-12-11 15:06:18,129 INFO  [main] conf.HiveConf: Using the default value passed in for log id: 2a5e5140-36da-4ea2-b95a-7a7ee3d87662
2020-12-11 15:06:18,129 INFO  [main] session.SessionState: Updating thread name to 2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main
2020-12-11 15:06:18,131 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Compiling command(queryId=snehal_20201211150618_130adc0
bc341): Select director, count(title) as content from netflix_data where country='India' group by director order by content desc  limit 10
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manag
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Starting Semantic Analysis
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Completed phase 1 of Semantic Analysis
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Get metadata for source tables
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] metastore.HiveMetaStore: 0: get_table : tbl=hive.default.netflix_d
2020-12-11 15:06:18,153 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] HiveMetaStore.audit: ugi=snehal       ip=unknown-ip-addr      cmd=
ault.netflix_data
2020-12-11 15:06:18,168 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Get metadata for subqueries
2020-12-11 15:06:18,168 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] parse.CalcitePlanner: Get metadata for destination tables
2020-12-11 15:06:18,201 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Context: New scratch dir is hdfs://localhost:9000/hive/warehous
ea2-b95a-7a7ee3d87662/hive_2020-12-11_15-06-18_150_7728679555281881354-1
6f-c87735fbc341); Time taken: 52.047 seconds
OK
2020-12-11 15:07:10,464 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: OK
2020-12-11 15:07:10,464 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] ql.Driver: Concurrency mode is disabled, not creating a lock manager
2020-12-11 15:07:10,472 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] mapred.FileInputFormat: Total input files to process : 1
        56
David Dhawan    8
S.S. Rajamouli  7
Ram Gopal Varma 6
Rajiv Mehra     5
Madhur Bhandarkar       5
Umesh Mehra     5
Ashutosh Gowariker      5
Vishal Bhardwaj 5
Anees Bazmee    5
2020-12-11 15:07:10,477 INFO  [2a5e5140-36da-4ea2-b95a-7a7ee3d87662 main] exec.ListSinkOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_
Time taken: 52.334 seconds, Fetched: 10 row(s)
```

PIG ANALYSIS:

STEPS TO START PIG:

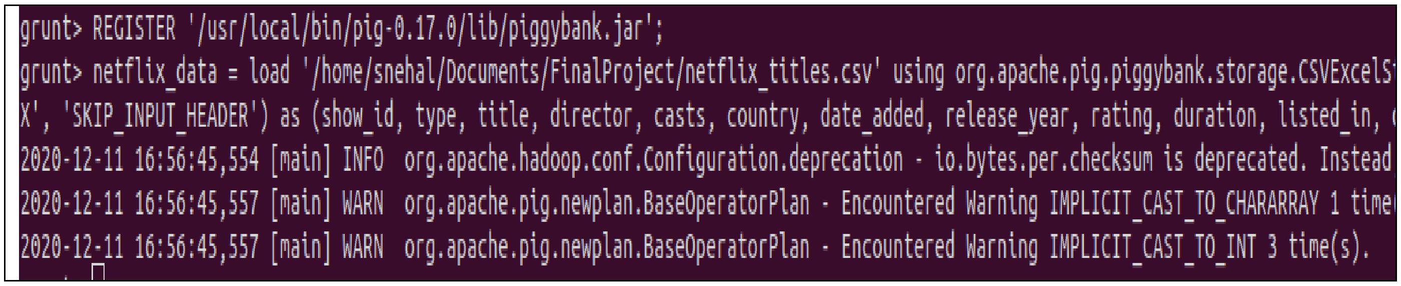
1. Navigate to Pig Directory
   cd /usr/local/bin/pig-0.17.0/bin

2. Start Pig in local mode
   pig -x local



```
snehal@snehal-Inspiron-5548:/usr/local/bin/pig-0.17.0/bin$ ls
pig  pig_1607729680032.log  pig_1607745451339.log  pig_1607745755266.log  pig.cmd  pig.py
snehal@snehal-Inspiron-5548:/usr/local/bin/pig-0.17.0/bin$ pig -x local
2020-12-14 15:51:16,075 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2020-12-14 15:51:16,076 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2020-12-14 15:51:16,166 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2020-12-14 15:51:16,166 [main] INFO  org.apache.pig.Main - Logging error messages to: /usr/local/bin/pig-0.17.0/bin/pig_16079898761
2020-12-14 15:51:16,207 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/snehal/.pigbootup not found
2020-12-14 15:51:16,360 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead,
ce.jobtracker.address
2020-12-14 15:51:16,363 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file sys
e:///
2020-12-14 15:51:16,497 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instea
bytes-per-checksum
2020-12-14 15:51:16,520 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-9d1f66e4-b414-4020-a541-
d
2020-12-14 15:51:16,520 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt> 
```

3. Load csv data into Netflix_data alias variable using CSVExcelStorage and skip the Header
   - Register the jar ‘piggybank.jar’ to use CSVExcelStorage function
   - netflix_data = load ‘local_path’ using org.apache.pig.piggybank.storage.CSVExcelStorage (',', 'NO_MULTILINE','UNIX', 'SKIP_INPUT_HEADER');



```
grunt> REGISTER '/usr/local/bin/pig-0.17.0/lib/piggybank.jar';
grunt> netflix_data = load '/home/snehal/Documents/FinalProject/netflix_titles.csv' using org.apache.pig.piggybank.storage.CSVExcelS
X', 'SKIP_INPUT_HEADER') as (show_id, type, title, director, casts, country, date_added, release_year, rating, duration, listed_in, 
2020-12-11 16:56:45,554 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead
2020-12-11 16:56:45,557 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time
2020-12-11 16:56:45,557 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_INT 3 time(s).
```

Analysis 10 - Find Movies details based on duration of the movie

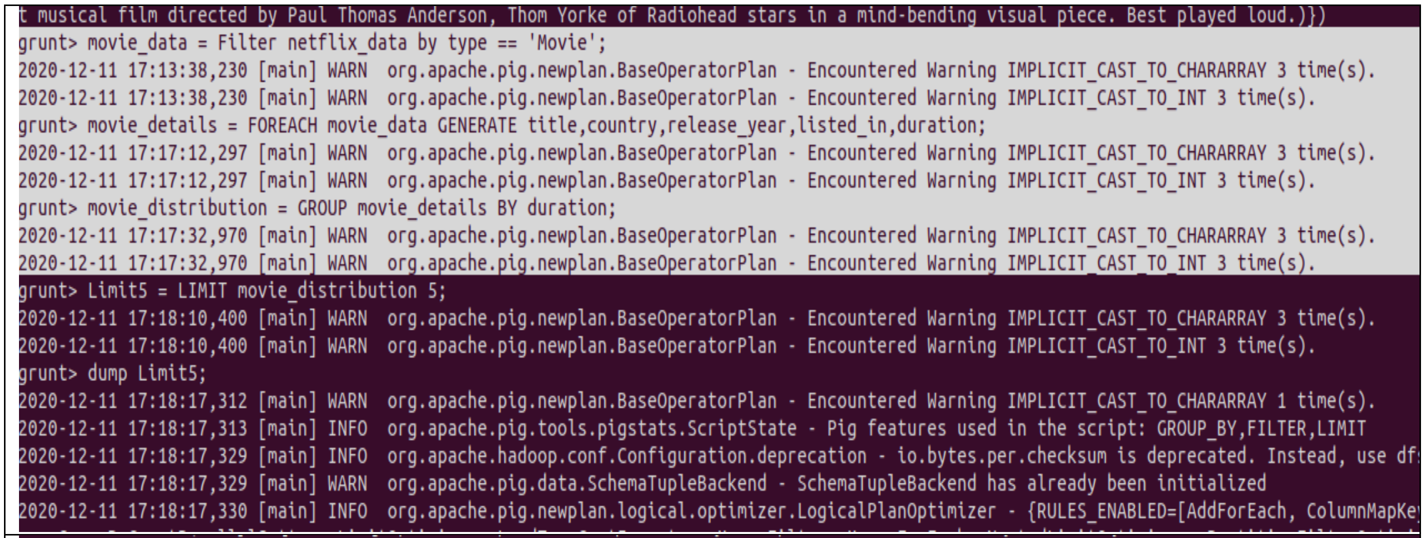
1. Filter netflix_data by type
   movie_data = Filter Netflix_data by type == “Movie”

2. Generate title, country, release year, genre, duration for each movie in movie_data
   movie_details = FOREACH movie_data GENERATE title, country, release_year, listed_in, duration;

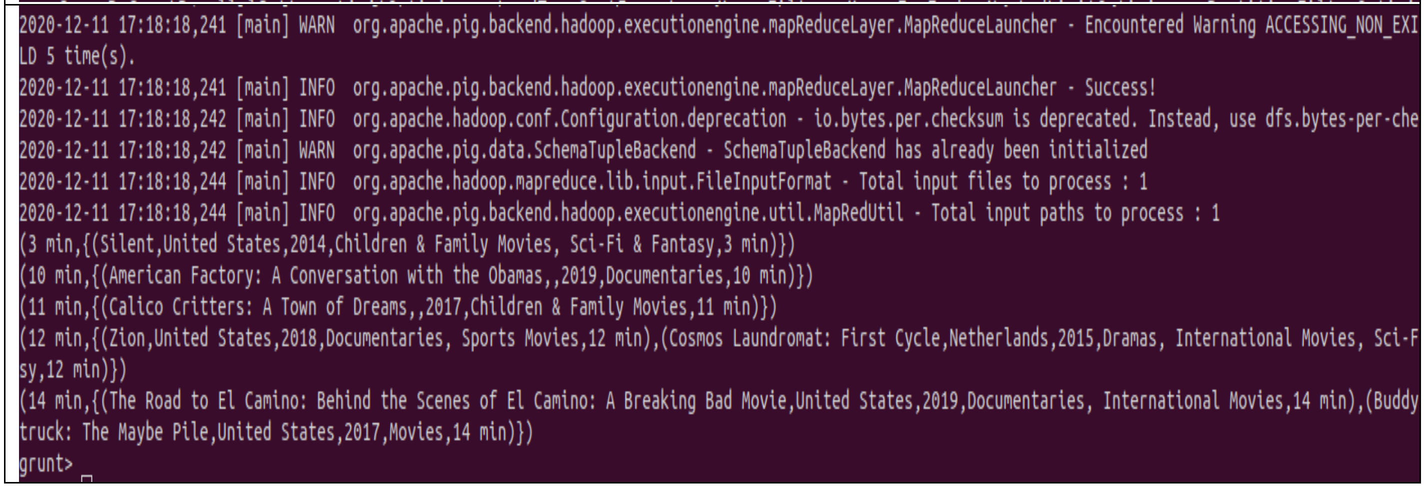
3. Group movies by duration
   movie_distribution = GROUP movie_details BY duration;

4. Show 5 movies from the distribution

Limit5 = Limit movie_distribution 5;
Dump Limit5;



```
t musical film directed by Paul Thomas Anderson, Thom Yorke of Radiohead stars in a mind-bending visual piece. Best played loud.)})
grunt> movie_data = Filter netflix_data by type == 'Movie';
2020-12-11 17:13:38,230 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 3 time(s).
2020-12-11 17:13:38,230 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_INT 3 time(s).
grunt> movie_details = FOREACH movie_data GENERATE title,country,release_year,listed_in,duration;
2020-12-11 17:17:12,297 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 3 time(s).
2020-12-11 17:17:12,297 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_INT 3 time(s).
grunt> movie_distribution = GROUP movie_details BY duration;
2020-12-11 17:17:32,970 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 3 time(s).
2020-12-11 17:17:32,970 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_INT 3 time(s).
grunt> Limit5 = LIMIT movie_distribution 5;
2020-12-11 17:18:10,400 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 3 time(s).
2020-12-11 17:18:10,400 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_INT 3 time(s).
grunt> dump Limit5;
2020-12-11 17:18:17,312 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 17:18:17,313 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY,FILTER,LIMIT
2020-12-11 17:18:17,329 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use df
2020-12-11 17:18:17,329 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2020-12-11 17:18:17,330 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKe
```



```
2020-12-11 17:18:18,241 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EXI
LD 5 time(s).
2020-12-11 17:18:18,241 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2020-12-11 17:18:18,242 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-che
2020-12-11 17:18:18,242 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2020-12-11 17:18:18,244 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-12-11 17:18:18,244 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(3 min,{(Silent,United States,2014,Children & Family Movies, Sci-Fi & Fantasy,3 min)})
(10 min,{(American Factory: A Conversation with the Obamas,,2019,Documentaries,10 min)})
(11 min,{(Calico Critters: A Town of Dreams,,2017,Children & Family Movies,11 min)})
(12 min,{(Zion,United States,2018,Documentaries, Sports Movies,12 min),(Cosmos Laundromat: First Cycle,Netherlands,2015,Dramas, International Movies, Sci-F
sy,12 min)})
(14 min,{(The Road to El Camino: Behind the Scenes of El Camino: A Breaking Bad Movie,United States,2019,Documentaries, International Movies,14 min),(Buddy
truck: The Maybe Pile,United States,2017,Movies,14 min)})
grunt> _
```

Analysis 11 - Percent Increase/Decrease in Netflix Data wrt release year 2000

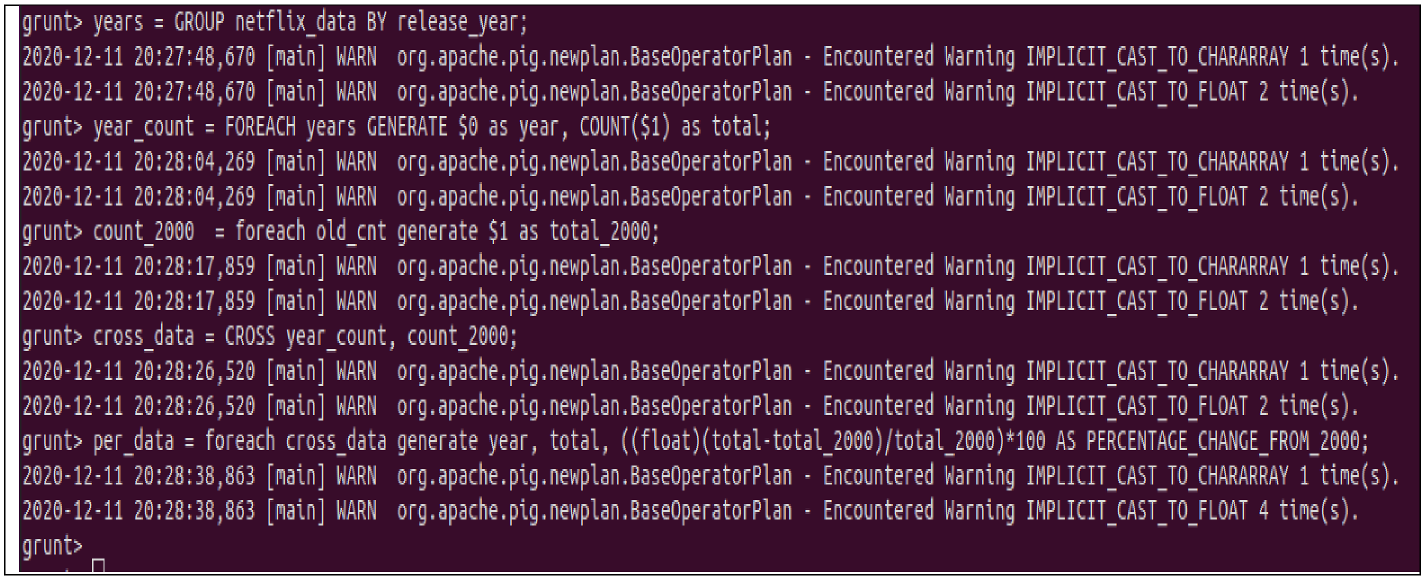
1. Group Netflix_data by release year
   years = GROUP netflix_data BY release_year;

2. Generate Year and Total number of Movies/TV shows for each year
   year_count = FOREACH years GENERATE $0 as year, COUNT($1) as total;

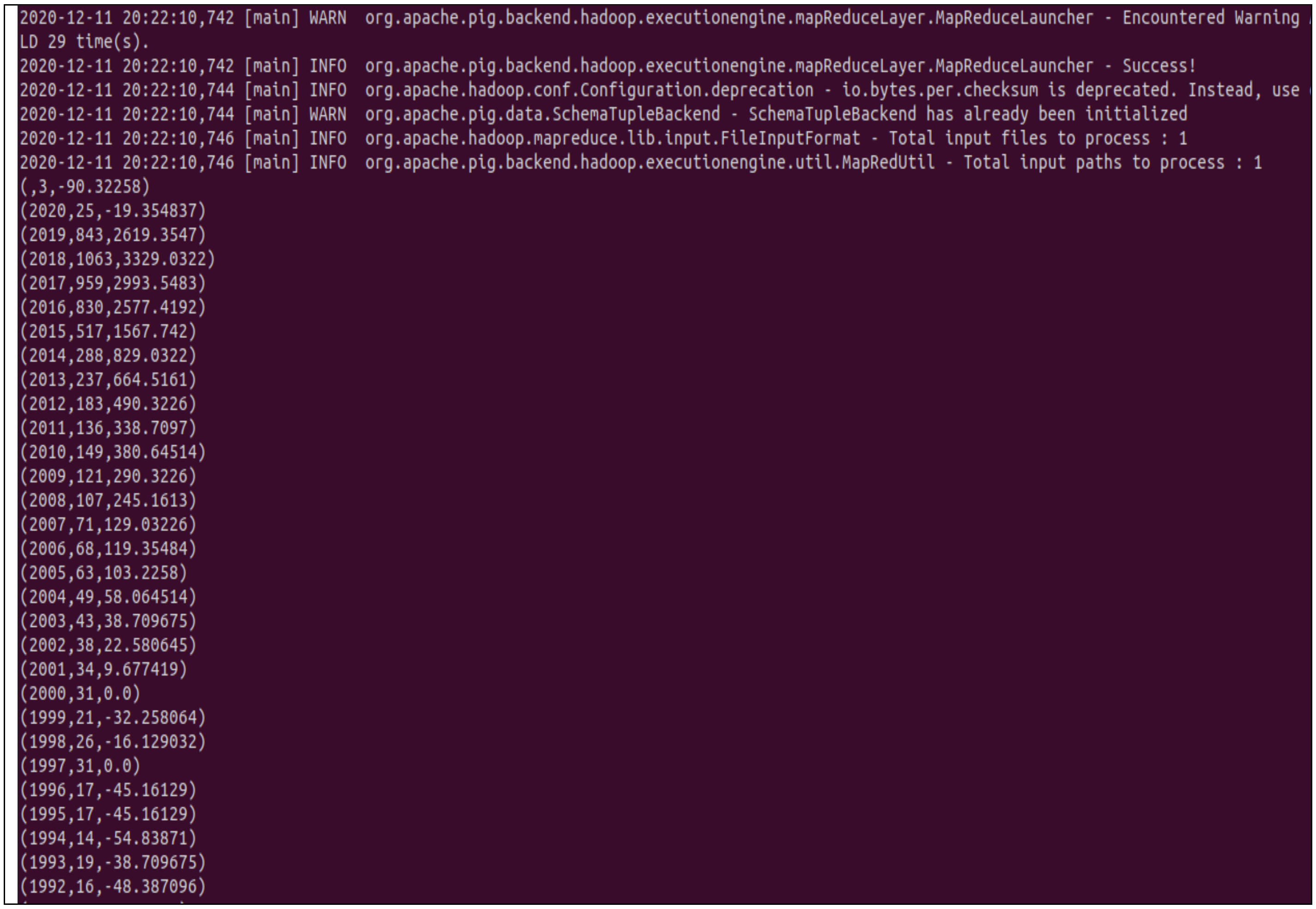
3. Get the Total number of Movies/TV shows for year 2000
   Old_cnt = Filter year_count by year == 2000;
   Count_2000 = foreach old_cnt generate $1 as total_2000;

4. Cross year_count and count_2000 so count_2000 is included in the generated variable
   cross_data = CROSS year_count, count_2000;

5. Calculate the percent increase/decrease for each year
   per_data = foreach cross_data generate year, total, ((float)(total-total_2000)/total_2000)*100 AS PERCENTAGE_CHANGE_FROM_2000;



```
grunt> years = GROUP netflix_data BY release_year;
2020-12-11 20:27:48,670 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 20:27:48,670 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_FLOAT 2 time(s).
grunt> year_count = FOREACH years GENERATE $0 as year, COUNT($1) as total;
2020-12-11 20:28:04,269 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 20:28:04,269 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_FLOAT 2 time(s).
grunt> count_2000  = foreach old_cnt generate $1 as total_2000;
2020-12-11 20:28:17,859 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 20:28:17,859 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_FLOAT 2 time(s).
grunt> cross_data = CROSS year_count, count_2000;
2020-12-11 20:28:26,520 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 20:28:26,520 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_FLOAT 2 time(s).
grunt> per_data = foreach cross_data generate year, total, ((float)(total-total_2000)/total_2000)*100 AS PERCENTAGE_CHANGE_FROM_2000;
2020-12-11 20:28:38,863 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_CHARARRAY 1 time(s).
2020-12-11 20:28:38,863 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_FLOAT 4 time(s).
grunt>
```

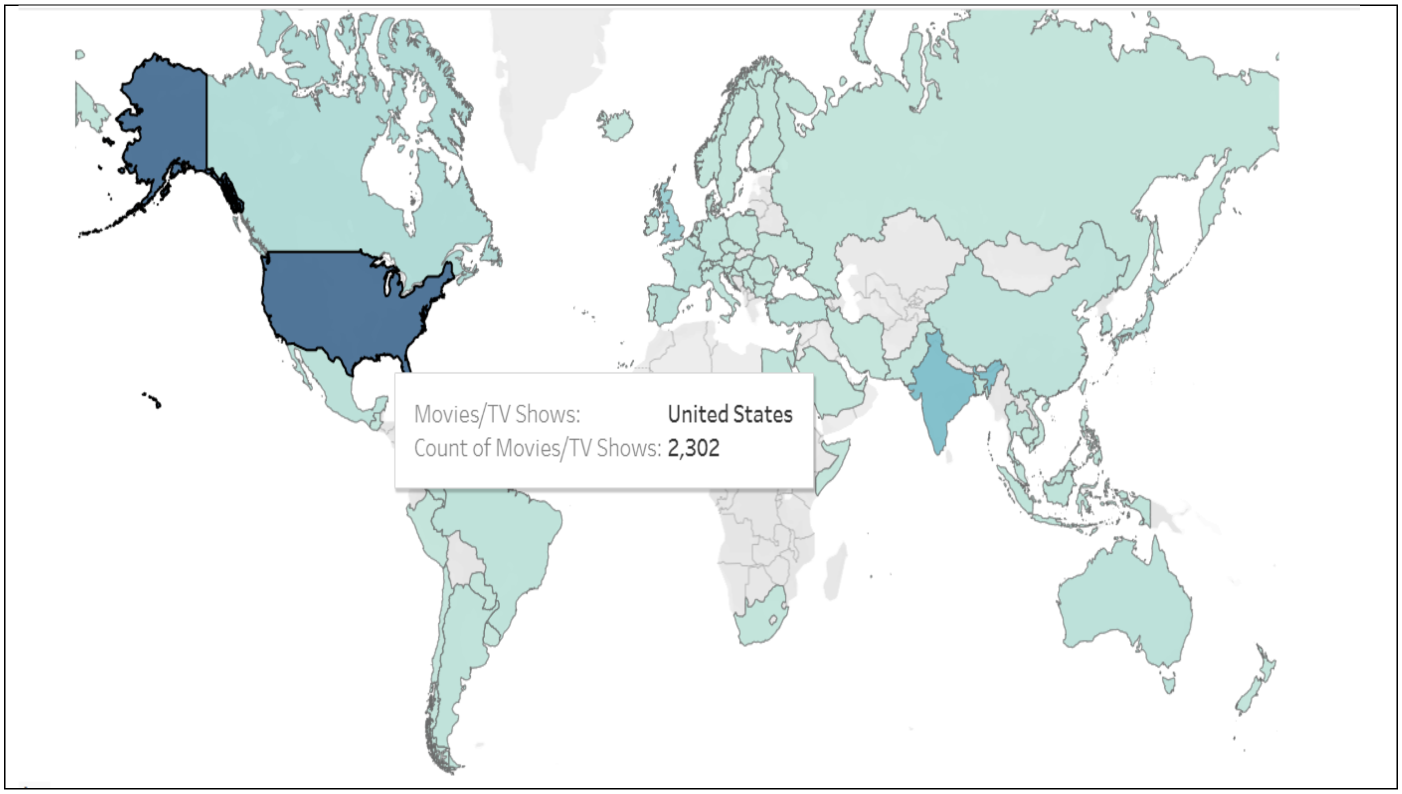


```
2020-12-11 20:22:10,742 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning
LD 29 time(s).
2020-12-11 20:22:10,742 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2020-12-11 20:22:10,744 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
2020-12-11 20:22:10,744 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2020-12-11 20:22:10,746 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2020-12-11 20:22:10,746 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(,3,-90.32258)
(2020,25,-19.354837)
(2019,843,2619.3547)
(2018,1063,3329.0322)
(2017,959,2993.5483)
(2016,830,2577.4192)
(2015,517,1567.742)
(2014,288,829.0322)
(2013,237,664.5161)
(2012,183,490.3226)
(2011,136,338.7097)
(2010,149,380.64514)
(2009,121,290.3226)
(2008,107,245.1613)
(2007,71,129.03226)
(2006,68,119.35484)
(2005,63,103.2258)
(2004,49,58.064514)
(2003,43,38.709675)
(2002,38,22.580645)
(2001,34,9.677419)
(2000,31,0.0)
(1999,21,-32.258064)
(1998,26,-16.129032)
(1997,31,0.0)
(1996,17,-45.16129)
(1995,17,-45.16129)
(1994,14,-54.83871)
(1993,19,-38.709675)
(1992,16,-48.387096)
```

Analysis 12 – Total number of Movies/TV shows for each Country

In this Analysis, Total number of Movies and TV shows are calculated based on the Countries.

This analysis is done using Tableau Public 2020.3 Application


Movies/TV Shows: United States
Count of Movies/TV Shows: 2,302

CONCLUSION:

- Performed multiple analysis on this dataset by running Hadoop on single machine.

- Hadoop MapReduce is implemented in Java language and eclipse IDE. Hadoop dependencies were installed by including dependencies in pom.xml. An executable jar is created to run MapReduce jobs on Hadoop CLI.

- Apache Hive provides a platform to write queries in SQL language to perform analysis on the dataset. Hive compile and run SQL queries as MapReduce jobs and the result can be stored in to HDFS.

- Apache Pig provides a pig CLI called as grunt which helps to write pig data flow scripts in pig's language, Pig Latin to perform MapReduce operations on the dataset.

APPENDIX

Analysis 01 – Count No of Movies and TV shows by the year they were released

- Map Function

```
> yearMap
function() {
      emit({Year:this.release_year},{count:1});
}
```

- Reduce Function

```
> yearReduce
function(key, values) {
       var sum = 0;
       values.forEach((val) => {sum += val.count;});
      return {count: sum};
}
```

- Execute MapReduce Job
  Db.movies.mapReduce(yearMap, yearReduce, {out: “MoviesCountByYear”});

- Print results
  Db.MoviesCountByYear.find();

Analysis 02 - Find movies and TV shows by Country and listed_in category using mongoDB indexes

- Create Index “CountryGenreIndex” for country and listed_in fields
  db.movies.createIndex({“country”:1,”listed_in”:1},{name:”CountryGenreIndex”});

- Check Index created
  db.movies.getIndexes();

- Find Movies and TV shows using created Index

Db.movies.find({“country”: “United States”, ”listed_in”:”Action & Adventure”}).pretty();

Analysis 03 – Count number of Total Movies and TV shows in dataset

```
public class TypeMapper extends Mapper<LongWritable, Text, Text,
IntWritable>{

    public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
        String input = value.toString();

        String[] movieData = input.split(",");
       try {
        String contentType = movieData[1];

        if(contentType.equals("type"))
            return;

        IntWritable one = new IntWritable(1);
        context.write(new Text(movieData[1]), one);
       }
       catch(Exception ex) {

       }
    }
}
```

```
public class TypeReducer extends Reducer<Text, IntWritable, Text,
IntWritable> {

    protected void reduce(Text key, Iterable<IntWritable> values,
Context context)
            throws IOException, InterruptedException {
        int count = 0;
        for (IntWritable val : values) {
            count += val.get();
        }
        context.write(key, new IntWritable(count));
    }
}
```

```
public class Driver {
    public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {
        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(conf);

        if (fs.exists(new Path(args[1]))) {
            fs.delete(new Path(args[1]), true);
        }

        Job job = Job.getInstance(conf);

        job.setMapperClass(TypeMapper.class);
        job.setReducerClass(TypeReducer.class);

        job.setJarByClass(Driver.class);

        TextInputFormat.addInputPath(job, new Path(args[0]));
        TextOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Analysis 04 - Movies and TV show analysis based on ratings using custom counter

```
public class RatingsMapper extends Mapper<LongWritable, Text, Text, 
IntWritable>{

     public static final String RATING = "Rating";

    public void map(LongWritable key, Text value, Context context) 
throws IOException, InterruptedException {
      String input = value.toString();
      //Split data on comma and ignore comma within double quotes
        String[] movieData = 
input.split(",(?=(?:[^\"]*\"[^\"]*\")*[^\"]*$)", -1);
       try {
        String ratings = movieData[8];

        if(ratings.equals("rating"))
```

```
             return;

         context.getCounter(RATING, ratings).increment(1);
        }
        catch(Exception ex) {

        }
    }
}
```

```
public class Driver {

     public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {

           Configuration conf = new Configuration();
           Job job = Job.getInstance(conf);

           job.setJarByClass(Driver.class);

           job.setMapperClass(RatingsMapper.class);

           FileInputFormat.addInputPath(job, new Path(args[0]));
           FileOutputFormat.setOutputPath(job, new Path(args[1]));

           FileSystem fs = FileSystem.get(conf);
           fs.delete(new Path(args[1]), true);

           int code = job.waitForCompletion(true) ? 0 : 1;

           if(code == 0) {
                for(Counter counter :
job.getCounters().getGroup(RatingsMapper.RATING)) {
                      System.out.println(counter.getDisplayName() +
"\t"+ counter.getValue());
                }
           }

           FileSystem.get(conf).delete(new Path(args[1]), true);

           System.exit(code);
     }
}
```

Analysis 05 – Implement partitioning on the basis on year the Movies and TV shows added in Netflix dataset

```
public class YearMapper extends Mapper<LongWritable, Text, IntWritable, Text>
{

      private IntWritable outKey = new IntWritable();

      public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
            String input = value.toString();
            // Split data on comma and ignore comma within double quotes
            String[] movieData =
input.split(",(?=(?:[^\"]*\"[^\"]*\")*[^\"]*$)", -1);
            try {
                  String dateAdded = movieData[6].replace("\"", "");

                  if (dateAdded.equals("date_added"))
                        return;

                  String year = dateAdded.split(",")[1];
                  if(!year.isEmpty()) {
                        int yearAdded = Integer.parseInt(year.replace(" ",
""));

                        outKey.set(yearAdded);
                  }

            } catch (Exception ex) {
                  System.out.println(ex);
            }
            context.write(outKey, value);
      }
}
```

```
public class YearReducer extends Reducer<IntWritable, Text, Text,
NullWritable> {

    protected void reduce(IntWritable key, Iterable<Text> values, Context
context)
            throws IOException, InterruptedException {

        for (Text val : values) {
            context.write(val, NullWritable.get());
        }

    }
}
```

```
public class YearPartition extends Partitioner<IntWritable, Text> {

     @Override
     public int getPartition(IntWritable key, Text value, int numPartition) {

          return (key.hashCode()) % numPartition;
     }
}
```

```
public class Driver {
     public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {
          Configuration conf = new Configuration();
          Job job = Job.getInstance(conf);
          job.setJarByClass(Driver.class);

          FileSystem fs = FileSystem.get(conf);

          if (fs.exists(new Path(args[1]))) {
               fs.delete(new Path(args[1]), true);
          }

          job.setNumReduceTasks(20);

          job.setMapperClass(YearMapper.class);
          job.setReducerClass(YearReducer.class);

          job.setPartitionerClass(YearPartition.class);

          TextInputFormat.addInputPath(job, new Path(args[0]));
          TextOutputFormat.setOutputPath(job, new Path(args[1]));

          job.setMapOutputKeyClass(IntWritable.class);
          job.setMapOutputValueClass(Text.class);

          job.setOutputKeyClass(Text.class);
          job.setOutputValueClass(NullWritable.class);

          System.exit(job.waitForCompletion(true) ? 0 : 1);
     }
}
```

Analysis 06 – Find Distinct Genres in the dataset

```
public class GenreMapper extends Mapper<LongWritable, Text, Text,
NullWritable>{

    public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
         String input = value.toString();
         //Split data on comma and ignore comma within double quotes
          String[] movieData =
input.split(",(?=(?:[^\"]*\"[^\"]*\")*[^\"]*$)", -1);
         try {
          String genreList = movieData[10];

          if(genreList.equals("listed_in"))
              return;

          String[] genres = genreList.replace("\"", "").split(",");
          for(String genre : genres) {
             context.write(new Text(genre), NullWritable.get());
          }
         }
         catch(Exception ex) {

         }
    }
}
```

```
public class GenreReducer extends Reducer<Text, NullWritable, Text,
NullWritable> {

    protected void reduce(Text key, Iterable<NullWritable> values,
Context context)
            throws IOException, InterruptedException {

          context.write(key, NullWritable.get());
    }
}
```

```
public class Driver {

     public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {
           Configuration conf = new Configuration();

           FileSystem fs = FileSystem.get(conf);
```

```
		if (fs.exists(new Path(args[1]))) {
			fs.delete(new Path(args[1]), true);
		}

		Job job = Job.getInstance(conf);

		job.setMapperClass(GenreMapper.class);
		job.setReducerClass(GenreReducer.class);
		job.setCombinerClass(GenreReducer.class);

		job.setJarByClass(Driver.class);

		TextInputFormat.addInputPath(job, new Path(args[0]));
		TextOutputFormat.setOutputPath(job, new Path(args[1]));

		job.setMapOutputKeyClass(Text.class);
		job.setMapOutputValueClass(NullWritable.class);

		job.setOutputKeyClass(Text.class);
		job.setOutputValueClass(NullWritable.class);

		System.exit(job.waitForCompletion(true) ? 0 : 1);
	}
}
```

Analysis 07 - Get movies and TV shows which are released before year 1970 using where clause

- Select Movie and TV shows details released before year 1970
  Select type, title, country, release_year from Netflix_data where release_year < 1970;

Analysis 08 – Find Movie or TV shows from Netflix data which are listed as "Stand-Up Comedy" and cast is "Russell Peters"

- Select Movie/TV shows which are listed as "Stand-up comedy" and casts = "Russell Peters"
  Select type, title, country, duration, date_added from Netflix_data where listed_in = 'Stand-Up Comedy' and casts = 'Russell Peters';

Analysis 09 - Find Directors from India with most contents

Select director, count(title) as content from Netflix_data where country='India' group by director order content desc limit 10;

Analysis 10 - Find Movies details based on duration of the movie

1. Filter netflix_data by type
   movie_data = Filter Netflix_data by type == "Movie"

2. Generate title, country, release year, genre, duration for each movie in movie_data
   movie_details = FOREACH movie_data GENERATE title, country, release_year, listed_in, duration;

3. Group movies by duration
   movie_distribution = GROUP movie_details BY duration;

4. Show 5 movies from the distribution
   Limit5 = Limit movie_distribution 5;
   Dump Limit5;

Analysis 11 - Percent Increase/Decrease in Netflix Data wrt release year 2000

1. Group Netflix_data by release year
   years = GROUP netflix_data BY release_year;

2. Generate Year and Total number of Movies/TV shows for each year
   year_count = FOREACH years GENERATE $0 as year, COUNT($1) as total;

3. Get the Total number of Movies/TV shows for year 2000
   Old_cnt = Filter year_count By year == 2000;
   Count_2000 = foreach old_cnt generate $1 as total_2000;

4. Cross year_count and count_2000 so count_2000 is included in the generated variable
   cross_data = CROSS year_count, count_2000;

5. Calculate the percent increase/decrease for each year
   per_data = foreach cross_data generate year, total, ((float)(total-total_2000)/total_2000)*100 AS PERCENTAGE_CHANGE_FROM_2000;