

UNIVERSITY OF MALAYA

EXAMINATION FOR THE DEGREE OF MASTER OF DATA SCIENCE

ACADEMIC SESSION 2019/2020 : SEMESTER II

WQD7005 : Data Mining

June 2020

StudentID: WQD180124

INSTRUCTIONS TO CANDIDATES :

Answer **ALL** questions (50 marks).

(This question paper consists of 5 questions on 3 printed pages)

Mini-assignment (50 marks)

Instructions: Work individually, submission via Spectrum.

1. You are required to make a user-agent that will crawl the WWW (your familiar domain) to produce dataset of a particular website.
 - the web site can be as simple as a list of webpages and what other pages they link to
 - the output does not need to be in XHTML (or HTML) form
a multi-stage approach (e.g. produce the xhtml or html in csv format)(10 marks)
2. Draw snowflake schema diagram for the above dataset. Justify your attributes to be selected in the respective dimensions.
(10 marks)
3. It will be appeared in week 13.
(10 marks)
4. It will be appeared in week 13.
(10 marks)
5. It will be appeared in week 14.
(10 marks)

Submissions:

The student is expected to submit answers to each question individually, and submit the document in PDF format. The student can include online materials, screenshots, videos and/or codes (ipynb format) to support your answer

Question 1

Targeted website: <https://www.rottentomatoes.com/>

First import all the relevant libraries

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
```

I've decided to scrap the top 1000 reviewed movies.

First, I proceed to the critics page @ <https://www.rottentomatoes.com/critics/authors>

Current Critics List

The Current Critics List includes many of the critics who have actively contributed to the Tomatometer® within the last calendar year.

- Critics with individual Tomatometer® approval are marked as "Tomatometer-approved critic." Their reviews are eligible for inclusion regardless of publication.
- Critics whose reviews are included at specific Tomatometer-approved publications will have those outlets listed next to their name.

For a list of many of the critics who are no longer active, but whose reviews are still part of the Tomatometer®, [click here](#).

LISTING BY

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

[Josefine A.](#) when published by [One Room With A View](#)

[Lysalex Hernández A.](#) when published by [La Nación \(Costa Rica\)](#)

[Alex Abad-Santos](#) when published by [Vox](#)

[Nicole Abadee](#) when published by [The Age \(Australia\)](#)

[Davide Abbatescianni](#) when published by [The Arts Desk](#)

[Kate Abbott](#) when published by [Guardian](#)

[Alana Joli Abbott](#) when published by [Den of Geek](#)

Observed that there is 26 pages and we want to scrap all the critics with their name and url.

Based on this, I've created a function to go through all 26 pages and obtain the name and url for the critics.

```

critic_name= []
critic_url= []
for letter in 'abcdefghijklmnopqrstuvwxyz':
    critics_url = "https://www.rottentomatoes.com/critics/authors?letter="+letter
    get_critics = requests.get(critics_url)
    critics_soup = BeautifulSoup(get_critics.text,"html.parser")
    for i in critics_soup.find_all('li', 'critics__list-item'):
        critic_name.append(i.find("a").string)
        critic_url.append("https://www.rottentomatoes.com"+str(i.find("a")['href']))

```





Then, I put in into a dataframe.

	Critic Name	URL
0	Josefine A.	https://www.rottentomatoes.com/critic/josefine-a
1	Lysalex Hernández A.	https://www.rottentomatoes.com/critic/lysalex-hernandez-a
2	Alex Abad-Santos	https://www.rottentomatoes.com/critic/alex-abad-santos
3	Nicole Abadee	https://www.rottentomatoes.com/critic/nicole-abadee
4	Davide Abbatescianni	https://www.rottentomatoes.com/critic/davide-abbatescianni
...
3848	Esther Zuckerman	https://www.rottentomatoes.com/critic/esther-zuckerman
3849	Lauren Zupkus	https://www.rottentomatoes.com/critic/lauren-zupkus
3850	David Zurawik	https://www.rottentomatoes.com/critic/david-zurawik
3851	Janire Zurbano	https://www.rottentomatoes.com/critic/janire-zurbano
3852	Natalie Zutter	https://www.rottentomatoes.com/critic/natalie-zutter

Next, we request each of the link for each critic.

Movies

« ‹ Showing 1 - 36 of 36 › »

Rating	T-Meter	Title Year	Review
 3/5	 100%	Charlatan (2020)	Despite the turbulent events of Mikoláek's life, Holland and screenwriter Marek Epstein have produced a remarkably calm film. - One Room With A View Read More Posted Mar 9, 2020
 5/5	 100%	Welcome to Chechnya (2020)	[David] France has created a documentary which is both incendiary and vital: we must bear witness to these atrocities and cannot allow them to be forgotten. - One Room With A View Read More Posted Mar 6, 2020
 4/5	No Score Yet	Dry Wind (2020)	Dry Wind's powerful and unabashed representation of homosexuality is momentous, all the more so when viewed against the backdrop of Brazil's current political climate. - One Room With A View Read More Posted Mar 5, 2020

For each critic we scrap all the ratings, T-meter, title and review for each movie that they have reviewed.

I've created a function to scrap all of the reviews.

```
190): def Critic_scrapper(URL,Name):
    #init df_table
    critics_url_2 = URL+"/movies?page=1"
    print(critics_url_2)
    get_critics_2 = requests.get(critics_url_2)
    critics_soup_2 = BeautifulSoup(get_critics_2.text,"html.parser")
    No_Review_tag = critics_soup_2.find_all("section",{"class":"panel panel-rt panel-box"})
    if "No reviews available." in str(No_Review_tag):
        pass
    else:
        Movie_tag = critics_soup_2.find_all("h2",{"class":"panel-heading js-review-type"})[0].string
        if "Movie" in Movie_tag:
            page = critics_soup_2.find('table', {'class': 'table table-striped critic-review-table responsive-table'})
            movie_url_critics = []
            for i in page.find_all("td",{"class":"col-xs-12 col-sm-6 critic-review-table__title-column"}):
                try:
                    movie_url_critics.append(i.find("a")["href"])
                except:
                    movie_url_critics.append("No data")
            ###check
            df_5 = pd.read_html(str(page))[0]
            df_5['Movie URL'] = movie_url_critics
            df_5['Critic Name'] = Name
            df_5
            print("Page 1 done"+Name)
            #
            #INIT DF for current critics
            tmp_tag = 0
            count=2
            while tmp_tag != -1:
                critics_url_2 = URL+"/movies?page="+str(count)
                #print(critics_url_2)
                get_critics_2 = requests.get(critics_url_2)
                critics_soup_2 = BeautifulSoup(get_critics_2.text,"html.parser")
                page = critics_soup_2.find('table', {'class': 'table table-striped critic-review-table responsive-table'})
                tmp_checker = len(page.find_all("td",{"class":"col-xs-12 col-sm-6 critic-review-table__title-column"}))
                #print(tmp_checker)
                if tmp_checker==0:
                    break
                tmp_tag = -1
                movie_url_critics = []
                #print(page.find_all("td",{"class":"col-xs-12 col-sm-6 critic-review-table__title-column"}))
                for i in page.find_all("td",{"class":"col-xs-12 col-sm-6 critic-review-table__title-column"}):
                    #print(i)
                    try:
                        movie_url_critics.append(i.find("a")["href"])
                    except:
                        movie_url_critics.append("No data")
                df_5_tmp = pd.read_html(str(page))[0]
                df_5_tmp['Movie URL'] = movie_url_critics
                df_5_tmp['Critic Name'] = Name
                df_5_tmp.append(df_5_tmp)
                #print("Page "+str(count)+" done"+Name)
                #
                count+=1
            return df_5
```

Then, we'll have to execute this function for all of the critics in the first dataframe.

Since there might be connection issue, I've created a while loop to force python to scrap my desired content.

```
b922
6xc6bf:      b922
        6xc6bf:      b922
        i+=j
        qt_2`iut`=qt_2`iut`+bbbeuq(L6znJf)
        L6znJf = CLtftc zcl9bbbeL(qt_3[,N8f,])[i]'qt_3[,CLtftc i9w6,][i])
        fLλ:
        M7Jf6 L6znJf i2 i0u6:
        L6znJf = i0u6
        fLλ:
        M7Jf6 i:= 3e28:
        i = j
        #10k6z a roug fime to zclab off qata (i0z qoue iu auoifuel uo2e00k)
        #10 zclab off fye CLtftc2 fLwou uoffeu iowato
```

The output will be in another dataframe like this:

df_5_init.head()							
	Unnamed: 0	Rating	T-Meter	Title Year	Review	Movie URL	Critic Name
0	0	3/5	100%	Charlatan (2020)	Despite the turbulent events of Mikolášek's life, Holland and screenwriter Marek Epstein have produced a remarkably calm film. - One Room With A View EDIT Read More Posted Mar 5, 2020	https://www.rottentomatoes.com/m/charlatan_2020	Josefine A.
1	1	5/5	100%	Welcome to Chechnya (2020)	[David] France has created a documentary which is both incendiary and vital: we must bear witness to these atrocities and cannot allow them to be forgotten. - One Room With A View EDIT Read More Posted Mar 6, 2020	https://www.rottentomatoes.com/m/welcome_to_chechnya	Josefine A.
2	2	4/5	No Score Yet	Dry Wind (2020)	Dry Wind's powerful and unabashed representation of homosexuality is momentous, all the more so when viewed against the backdrop of Brazil's current political climate. - One Room With A View EDIT Read More Posted Mar 5, 2020	https://www.rottentomatoes.com/m/dry_wind	Josefine A.
3	3	2/5	No Score Yet	Delete History (Effacer l'histoire) (2020)	Drawn into complete ridicule, there is very little that remains relatable, and the superficial narrative which leads all three to ultimately renounce their digital dependency (at least for a while) fails to move. - One Room With A View EDIT Read More Posted Mar 2, 2020	https://www.rottentomatoes.com/m/delete_history	Josefine A.
4	4	3/5	45%	Surge (2020)	It's undoubtedly Whishaw who keeps the film compelling with his startling performance, even when the script stumbles somewhat in the final half hour. - One Room With A View EDIT Read More Posted Mar 2, 2020	https://www.rottentomatoes.com/m/surge	Josefine A.

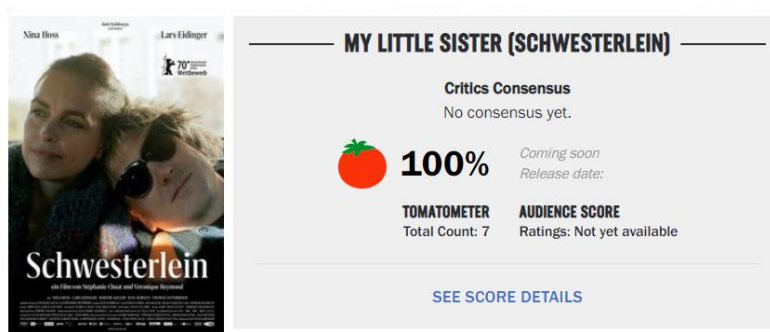
Next, we group the scrapped data frame by the movie title and sum it to find the frequency.

Then we select the top 1000 movies with the most reviews.

```
top_500_reviewed_movies.head()
```

	Title Year	Movie URL	Freq
0	Joker (2019)	https://www.rottentomatoes.com/m/joker_2019	530
1	Once Upon a Time In Hollywood (2019)	https://www.rottentomatoes.com/m/once_upon_a_time_in_hollywood	512
2	Us (2019)	https://www.rottentomatoes.com/m/us_2019	483
3	Avengers: Endgame (2019)	https://www.rottentomatoes.com/m/avengers_endgame	475
4	Captain Marvel (2019)	https://www.rottentomatoes.com/m/captain_marvel	466

From here, we can start scrapping our movie information. We want to scrap the following info (As shown in the image)



The score

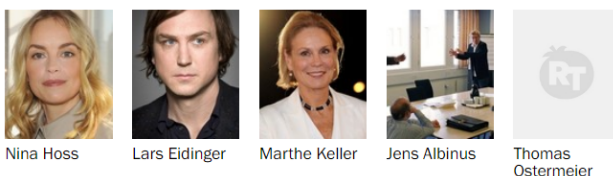
MOVIE INFO

Brilliant playwright, Lisa, no longer writes. She lives in Switzerland with her family but her heart has stayed in Berlin, beating to the rhythm of her brother's heart. The ties between the twins have grown stronger since Sven was diagnosed with an aggressive type of leukemia. He's a famous theater actor and Lisa refuses to accept his fate, moving heaven and earth to get him back on stage. She gives her all for her soul mate, neglecting everything else, even risking her marriage. Her relationship with her husband starts to fall apart, but Lisa only has eyes for her brother, her mirror, who connects her back with her deepest...

[SHOW MORE](#)

Rating: NR
Genre: Drama
Directed By: [Stéphanie Chuat, Véronique Reymond](#)
Written By: [Stéphanie Chuat, Véronique Reymond](#)
Runtime: 99 minutes
Studio: Vega Film

CAST



The movie info and casts.

Based on that, I've created another function.

```
def movie_info_fn_cast(URL, Movie_name):
    dict_formula = {'Rating':0, 'Genre':1, 'Directed By':2, 'Written By':3, 'In Theaters':4, 'On Disc/Streaming':5, 'Box Office':6}
    movie_url = URL
    get_movie = requests.get(movie_url)
    movie_soup = BeautifulSoup(get_movie.text, "html.parser")
    movie_info = movie_soup.find("div", {"class": "panel-body content_body"}).find_all("div")
    synopsis = movie_soup.find_all("div", {"id": "movieSynopsis"})[0].string
    synopsis = synopsis.strip().replace("\n", "")
    movie_detail_head = movie_soup.find("div", {"class": "panel-body content_body"}).find_all("div", {"class": "meta-label subtle"})
    movie_detail_head_names = [i.string.replace(":", "") for i in movie_detail_head]
    movie_detail_head_names.append("Synopsis")
    movie_detail_head_names.append("Movie Title")
    movie_detail_head_names.append("Rating Score")
    movie_detail_head_names.append("Rating Count")
    movie_detail_values = movie_soup.find("div", {"class": "panel-body content_body"}).find_all("div", {"class": "meta-value"})
    movie_detail_values_val = []
    ratings = movie_soup.find_all("div", {"class": "mop-ratings-wrap_half"})[0]
    for num, i in enumerate(movie_detail_head_names):
        if dict_formula[i] == 0:
            movie_detail_values_val.append(movie_detail_values[num].string.strip())
        if dict_formula[i] == 1:
            movie_detail_values_val.append(movie_detail_values[num].string.replace("\n", "\n"))
        if dict_formula[i] == 2:
            movie_detail_values_val.append([(((i.string)+":https://www.rottentomatoes.com"+i["href"])] for i in movie_detail_values_val))
        if dict_formula[i] == 3:
            movie_detail_values_val.append([(((i.string)+":https://www.rottentomatoes.com"+i["href"])] for i in movie_detail_values_val))
        if dict_formula[i] == 4:
            movie_detail_values_val.append(movie_detail_values[num].find("time").string.strip())
        if dict_formula[i] == 5:
            movie_detail_values_val.append(movie_detail_values[num].find("time").string.strip())
        if dict_formula[i] == 6:
            movie_detail_values_val.append(movie_detail_values[num].string)
        if dict_formula[i] == 7:
            movie_detail_values_val.append(movie_detail_values[num].find("time").string.strip())
        if dict_formula[i] == 8:
            movie_detail_values_val.append(movie_detail_values[num].string.strip())
        if dict_formula[i] == 9:
            movie_detail_values_val.append(synopsis)
        if dict_formula[i] == 10:
            movie_detail_values_val.append(Movie_name)
        if dict_formula[i] == 11:
            if ratings.find("span", {"class": "mop-ratings-wrap_percentage"}) == None:
                movie_detail_values_val.append("No Data")
            else:
                movie_detail_values_val.append(ratings.find("span", {"class": "mop-ratings-wrap_percentage"}).string.strip())
        if dict_formula[i] == 12:
            if ratings.find("small", {"class": "mop-ratings-wrap_text--small"}) == None:
                movie_detail_values_val.append("No Data")
            else:
                movie_detail_values_val.append(ratings.find("small", {"class": "mop-ratings-wrap_text--small"}).string.strip())
    tmp_df_2 = pd.DataFrame([movie_detail_values_val], columns=movie_detail_head_names)
    critics_soup = movie_soup
    cast_list = []
    cast_url = []
    init_cast = critics_soup.find_all("div", {"class": "cast-item media inlineBlock"})
    for i in range(len(init_cast)):
        cast_list.append(init_cast[i].find("span").string.strip())
        cast_url.append("https://www.rottentomatoes.com"+init_cast[i].find("a")["href"])
    if "cast-item media inlineBlock castRemaining" in str(critics_soup):
        remaining_cast = critics_soup.find_all("div", {"class": "cast-item media inlineBlock castRemaining"})
        for i in range(len(remaining_cast)):
            cast_list.append(remaining_cast[i].find("span").string.strip())
            cast_url.append("https://www.rottentomatoes.com"+remaining_cast[i].find("a")["href"])
    _ = {"Cast": cast_list, "URL": cast_url}
    df_cast = pd.DataFrame(_)
    df_cast['Movie Name'] = Movie_name
    return tmp_df_2, df_cast
```

As highlighted, dictionary is used since each of the movies has different infos. I've created a dictionary that have specific formula for the info. On the 3rd highlighted part, it is used to scrap the cast information.

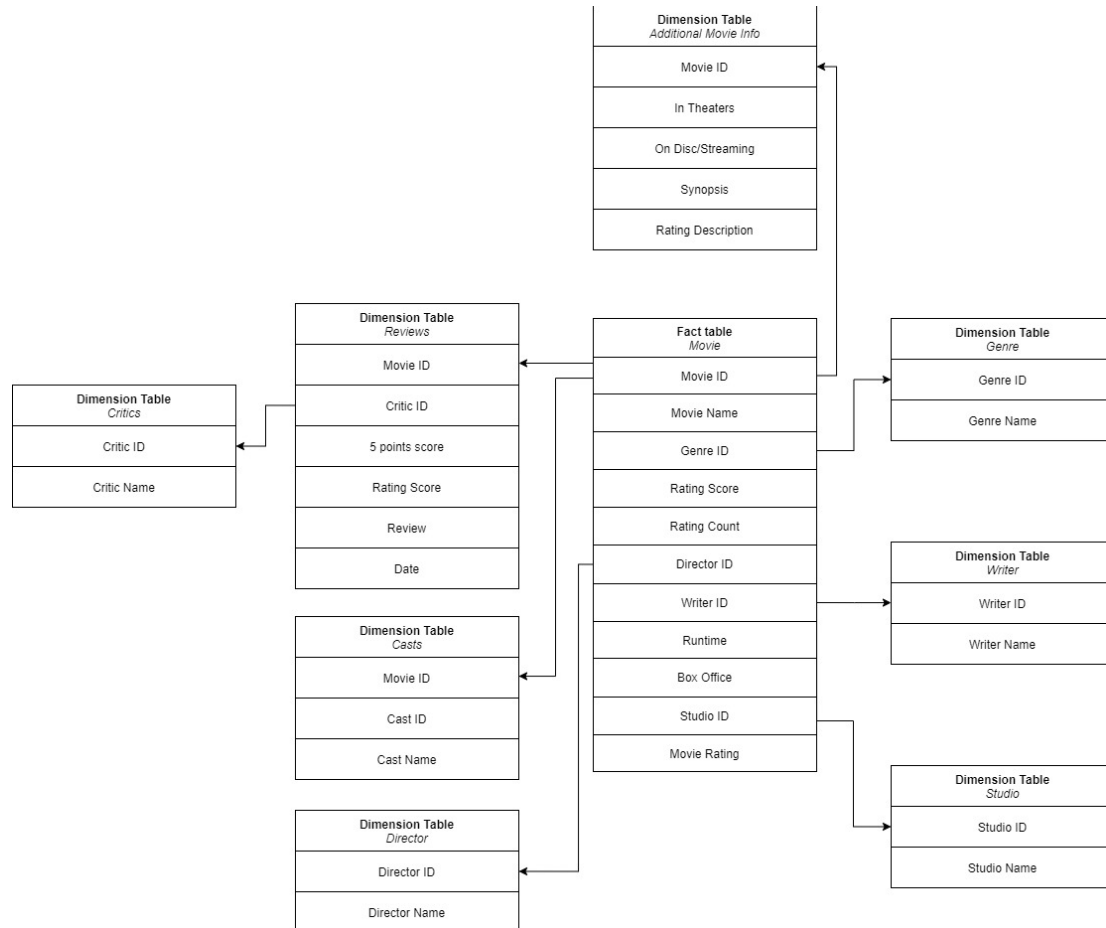
Then using the same while loop method, this is executed for the top 1000 movies.

```
i=1
while i != len(top_500_reviewed_movies):
    result = []
    while len(result) != 2:
        try:
            info, cast = movie_info_fn_cast(top_500_reviewed_movies["Movie URL"][i], top_500_reviewed_movies["Title | Year"][i])
            result = [info, cast]
            init_info = init_info.append(result[0])
            init_cast = init_cast.append(result[1])
            print(i, "Done")
            i += 1
        except:
            pass
```


Next, we will have two useful table: Review list table and movie info table.

Question 2

To convert the table into snowflake schema data format.



In this table, it shows the snowflake schema of my scrapped data.

I've selected these attributes so that it can have a better data quality (Data is more structured and data integrity problems are reduced). Besides that, less disk space is used then in a denormalized model.

If we used the initial scrapped dataframe, it contains a lot of text data which loads up the memory of the system. Doing this will reduce the chance of it happening.

Overall, I've generated multiple tables.

For the movie info table, I've removed all the director, writer, studio, genre and replaced it with IDs.

I've also removed the unnecessary columns such as Rating Description, Synopsis, In theaters, On disk streaming and put in under another table. Since these data mainly consist of text and may affect the running performance. It is placed in a separate table so that user can still access it when needed.

Example for the processing into dimensional table. (Repeated for each ID table)

- Find unique name
- Generate unique id for each name
- Put it in a dataframe
- Merge to required tables.

```
: unique_critics = top_1000_critics_list['Critic Name'].unique()

: unique_critics

: array(['Josefine A.', 'Lysalex Hernández A.', 'Alex Abad-Santos', ...,
        'Guy Webster', 'Princess Weekes', 'James Wegg'], dtype=object)

: critics_id = ["C%s"%i for i in range(len(unique_critics))]

: data = {
:     "Critic Name":unique_critics,
:     "Critic ID":critics_id
: }
: critics_table = pd.DataFrame(data)
: critics_table.head(5)

:
:      Critic Name  Critic ID
: 0    Josefine A.      C0
: 1  Lysalex Hernández A.  C1
: 2    Alex Abad-Santos  C2
: 3    Alana Joli Abbott  C3
: 4    Harrison Abbott   C4

: top_1000_critics_list= top_1000_critics_list.merge(critics_table, on='Critic Name', how='left')
```

Output as shown below:

	Rating Score	T-Meter	Movie Name	Review	Movie URL	Critic Name	Critic ID
0	4/5	88%	Shirley (2020)	Shirley unquestionably does its subject justice, both on a personal level and in the wider context of female authorship. - One Room With A View EDIT Read More Posted Mar 2, 2020	https://www.rottentomatoes.com/m/shirley_2020	Josefine A.	C0
1	5/5	99%	Never Rarely Sometimes Always (2020)	With a narrative that is both universal and deeply personal, <i>Never Rarely Sometimes Always</i> is a film of the utmost urgency, a gut punch of the very best kind. - One Room With A View EDIT Read More Posted Mar 2, 2020	https://www.rottentomatoes.com/m/never_rarely_sometimes_always	Josefine A.	C0
2	3/5	91%	Bacurau (Nighthawk) (2020)	With a set up as large as this, <i>Bacurau</i> would have benefitted from the more generous space of a (mini-)series in order to explore the nuances of this highly ambitious concept more fully. - One Room With A View EDIT Read More Posted Jul 8, 2019	https://www.rottentomatoes.com/m/bacurau	Josefine A.	C0
3	4/5	99%	Amazing Grace (2019)	The film is a precious time capsule, preserving the raw and thrilling presence of the Queen of Soul both for those who saw her on stage before, and those who never had the chance. - One Room With A View EDIT Read More Posted Feb 19, 2019	https://www.rottentomatoes.com/m/amazing_grace_2018	Josefine A.	C0
4	NaN	99%	Apollo 11 (2019)	an excellent opportunity to look at the past and dream about what man or woman can do in the not too distant future. [Full review in Spanish] - La Nación (Costa Rica) EDIT Read More Posted Jul 29, 2019	https://www.rottentomatoes.com/m/apollo_11_2019	Lysalex Hernández A.	C1

Several preprocessing has been done on the tables using the similar method above.

Finally, 9 tables are generated as shown in the snowflake schema above.

Table Reference below

```
# First table is the movie list & info table
output_main_table.head()
```

	Movie ID	Movie Name	Genre_ID	Rating Score	Rating Count	Director ID	Writer ID	Runtime	Box Office	Studio ID	Movie Rating
0	M0	Joker (2019)	[G0, G1, G2]	68%	555	[D0]	[W0, W1]	122 minutes	No Data	S0	R
1	M1	Once Upon a Time in Hollywood (2019)	[G3, G1]	85%	532	[D1]	[W2]	159 minutes	No Data	S1	R
2	M2	Us (2019)	[G4, G2]	90%	516	[D2]	[W3]	120 minutes	No Data	S2	R
3	M3	Avengers: Endgame (2019)	[G0, G1, G5]	94%	510	[D3]	[W4, W5]	182 minutes	No Data	S3	PG-13
4	M4	Captain Marvel (2019)	[G0, G5]	78%	505	[D4, D5]	[W6, W7, W8]	128 minutes	No Data	S3	PG-13

```
# Second table is additional movie info table
movie_add_info.head()
```

	Movie ID	In Theaters	On Disc/Streaming	Synopsis	Rating Description
0	M0	Oct 4, 2019	Dec 17, 2019	"Joker" centers around the iconic arch nemesis and is an original, standalone fictional story not seen before on the big screen. Phillips exploration of Arthur Fleck, who is indelibly portrayed by Joaquin Phoenix, is of a man struggling to find his way in Gothams fractured society. A clown-for-hire by day, he aspires to be a stand-up comic at night...but finds the joke always seems to be on him. Caught in a cyclical existence between apathy and cruelty, Arthur makes one bad decision that brings about a chain reaction of escalating events in this gritty character study.	R (for strong bloody violence, disturbing behavior, language and brief sexual images)
1	M1	Jul 26, 2019	Nov 22, 2019	Quentin Tarantino's ninth feature film is a story that takes place in Los Angeles in 1969, at the height of hippy Hollywood. The two lead characters are Rick Dalton (Leonardo DiCaprio), former star of a western TV series, and his longtime stunt double Cliff Booth (Brad Pitt). Both are struggling to make it in a Hollywood they don't recognize anymore. But Rick has a very famous next-door neighbor...Sharon Tate.	R (for language throughout, some strong graphic violence, drug use, and sexual references)
2	M2	Mar 22, 2019	Jun 18, 2019	Set in present day along the iconic Northern California coastline, Us, from Monkeypaw Productions, stars Oscar (R) winner Lupita Nyong'o as Adelaide Wilson, a woman returning to her beachside childhood home with her husband, Gabe (Black Panthers Winston Duke), and their two children (Shahad Wright Joseph, Evan Alex) for an idyllic summer getaway. Haunted by an unexplainable and unresolved trauma from her past and compounded by a string of eerie coincidences, Adelaide feels her paranoia elevate to high-alert as she grows increasingly certain that something bad is going to befall her family. After spending a tense beach day with their friends, the Tylers (Emmy winner Elisabeth Moss, Tim Heidecker, Cali Sheldon, Noelle Sheldon), Adelaide and her family return to their vacation home. When darkness falls, the Wilsons discover the silhouette of four figures holding hands as they stand in the driveway. Us pits an endearing American family against a terrifying and uncanny opponent: doppelgängers of themselves.	R (for violence/terror, and language)
3	M3	Apr 26, 2019	Jul 30, 2019	The grave course of events set in motion by Thanos that wiped out half the universe and fractured the Avengers ranks compels the remaining Avengers to take one final stand in Marvel Studios grand conclusion to twenty-two films, "Avengers: Endgame."	PG-13 (for sequences of sci-fi violence and action, and some language)
4	M4	Mar 8, 2019	Jun 11, 2019	The story follows Carol Danvers as she becomes one of the universe's most powerful heroes when Earth is caught in the middle of a galactic war between two alien races. Set in the 1990s, Captain Marvel is an all-new adventure from a previously unseen period in the history of the Marvel Cinematic Universe.	PG-13 (for sequences of sci-fi violence and action, and brief suggestive language)

```
# Third table is additional studio table
studio_table.head()
```

	Studio Name	Studio ID
0	Warner Bros. Pictures	S0
1	Columbia Pictures	S1
2	Universal Pictures	S2
3	Marvel Studios	S3
4	Walt Disney Pictures	S4

```
# 4th table is the writer table
writer_table.head()
```

	Writer Name	Writer ID
0	Todd Phillips	W0
1	Scott Silver	W1
2	Quentin Tarantino	W2
3	Jordan Peele	W3
4	Christopher Markus	W4

```
# 5th table is the director table
director_table.head()
```

	Director Name	Director ID
0	Todd Phillips	D0
1	Quentin Tarantino	D1
2	Jordan Peele	D2
3	Anthony Russo	D3
4	Anna Boden	D4

```
# 6th table is the genre table
genre_table.head()
```

	Genre	Genre ID
0	Action & Adventure	G0
1	Drama	G1
2	Mystery & Suspense	G2
3	Comedy	G3
4	Horror	G4

```
# 7th table is the cast table
cast_table_final.head()
```

	Movie ID	Cast ID	Cast Name
0	M0	A0	Joaquin Phoenix
1	M0	A1	Robert De Niro
2	M0	A2	Zazie Beetz
3	M0	A3	Bill Camp
4	M0	A4	Frances Conroy

```
# 8th table is the critics table
critics_table.head()
```

	Critic Name	Critic ID
0	Josefine A.	C0
1	Lysalex Hernández A.	C1
2	Alex Abad-Santos	C2
3	Alana Joli Abbott	C3
4	Harrison Abbott	C4

```
# 9th table is the critics review table
critics_review_table.head()
```

	Movie ID	Critic ID	5 points score	Rating Score		Review	Date
0	M596	C0	4/5	88%	Shirley unquestionably does its subject justice, both on a personal level and in the wider context of female authorship.		Mar 2, 2020
1	M552	C0	5/5	99%	With a narrative that is both universal and deeply personal, <i>Never Rarely Sometimes Always</i> is a film of the utmost urgency, a gut punch of the very best kind.		Mar 2, 2020
2	M880	C0	3/5	91%	With a set up as large as this, <i>Bacurau</i> would have benefitted from the more generous space of a (mini		Jul 8, 2019
3	M874	C0	4/5	99%	The film is a precious time capsule, preserving the raw and thrilling presence of the Queen of Soul both for those who saw her on stage before, and those who never had the chance.		Feb 19, 2019
4	M536	C1	NaN	99%	an excellent opportunity to look at the past and dream about what man or woman can do in the not too distant future. [Full review in Spanish]		Jul 29, 2019

