PROJECT-2_INFO-5709

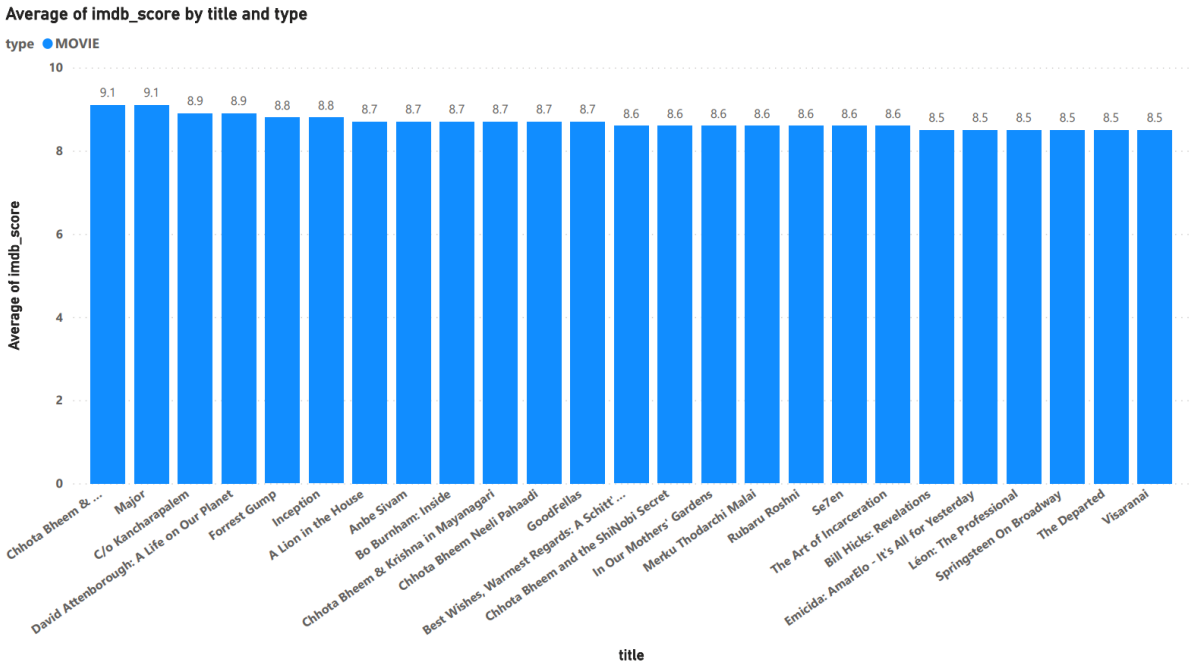
PART-1:
For this project, we took a movie dataset from kaggle (https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies?resource=download). The datasets are credits and titles.

Nominal(N), Quantitative(Q), Ordinal(O).

The Credit dataset contains person_id(Q), id(N), name(N), character(N), role(N).
Titles dataset contains id(N), title(N), type(N), description(N), release_yr(Q), age_certifications(O), runtime(Q), genre(N), production_countries(N), seasons(Q), imdb_score(Q).

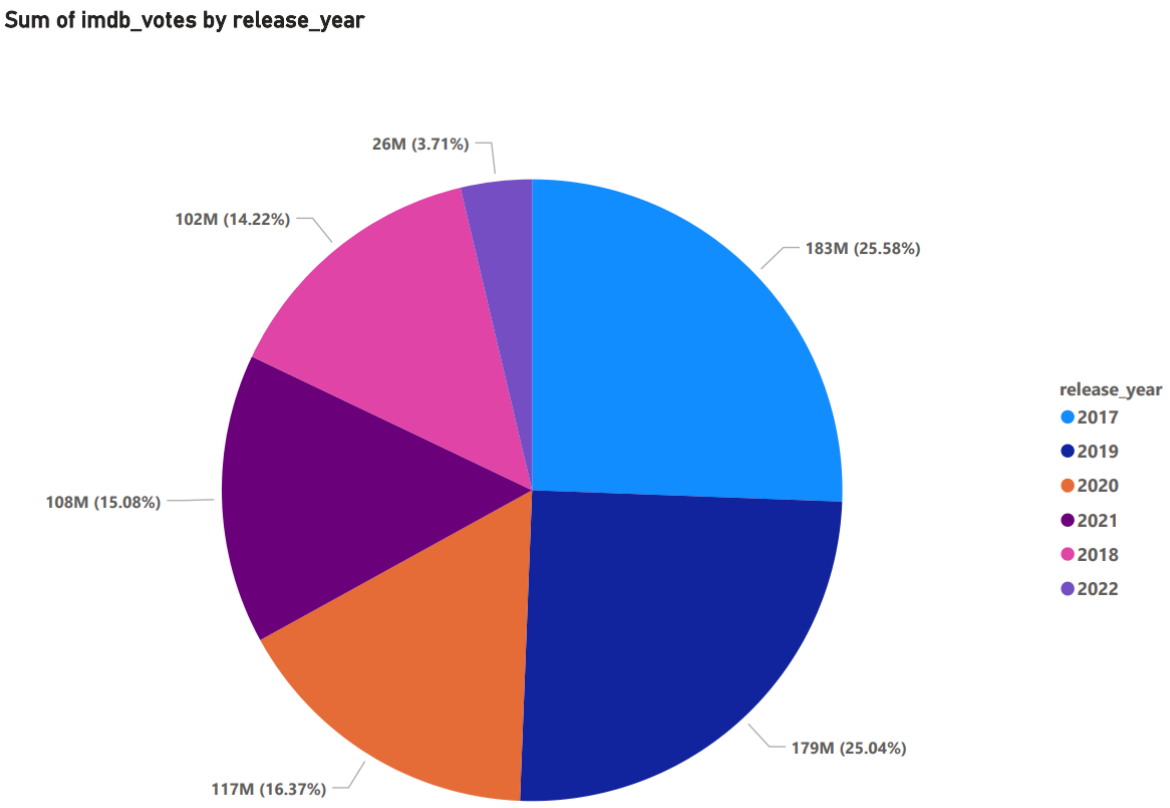
The three questions for the data, that I have analyzed are

1. Listing the movie titles of rating having greater than 8 and finding the best movie over them.

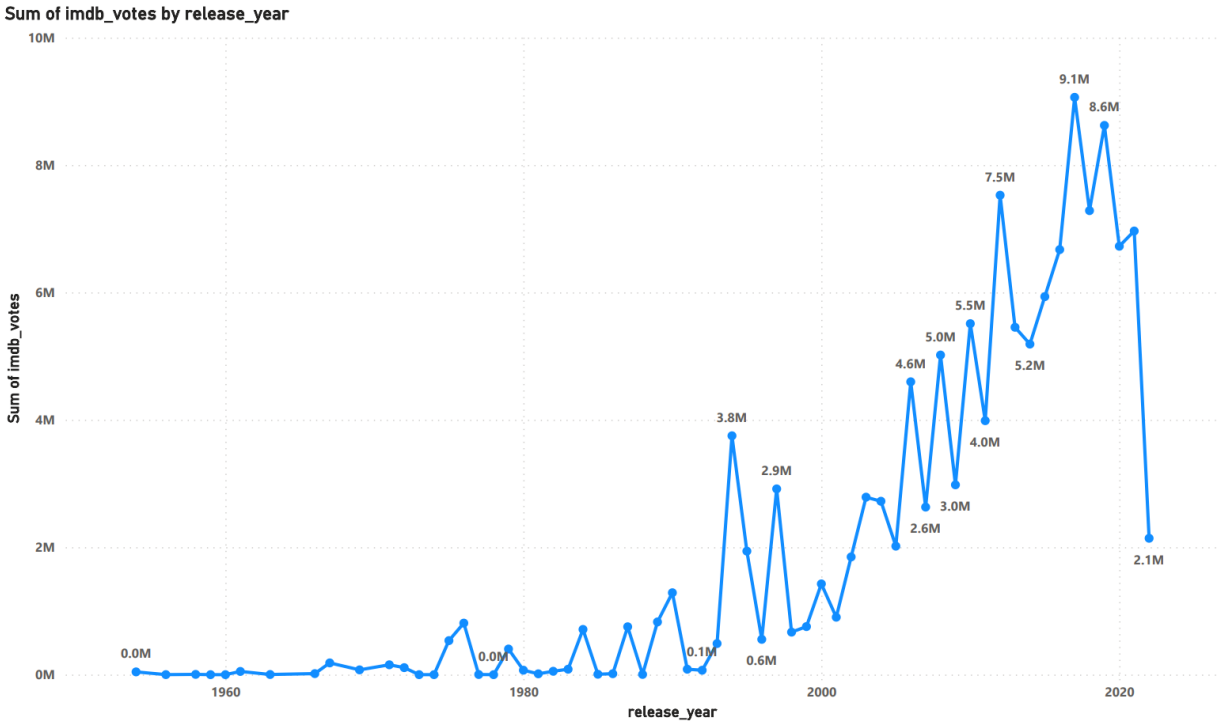


Here to know the answer, I have taken the average of my imdb score and rating greater than 8.I used stacked column chart on the scale of x-axis with movie titles and y-axis with average of imdb_score on scale of 0 to 10 and filtering out only with the ratings greater than 8. The size of the image is 649x1164. The color of each stacked column used is blue. There are a total of 25 movies with ratings greater than 8. In which, the highest is Chhota Bheem with 9.1 rating.

2. To know, what % of votes have been submitted in the last 5 years.

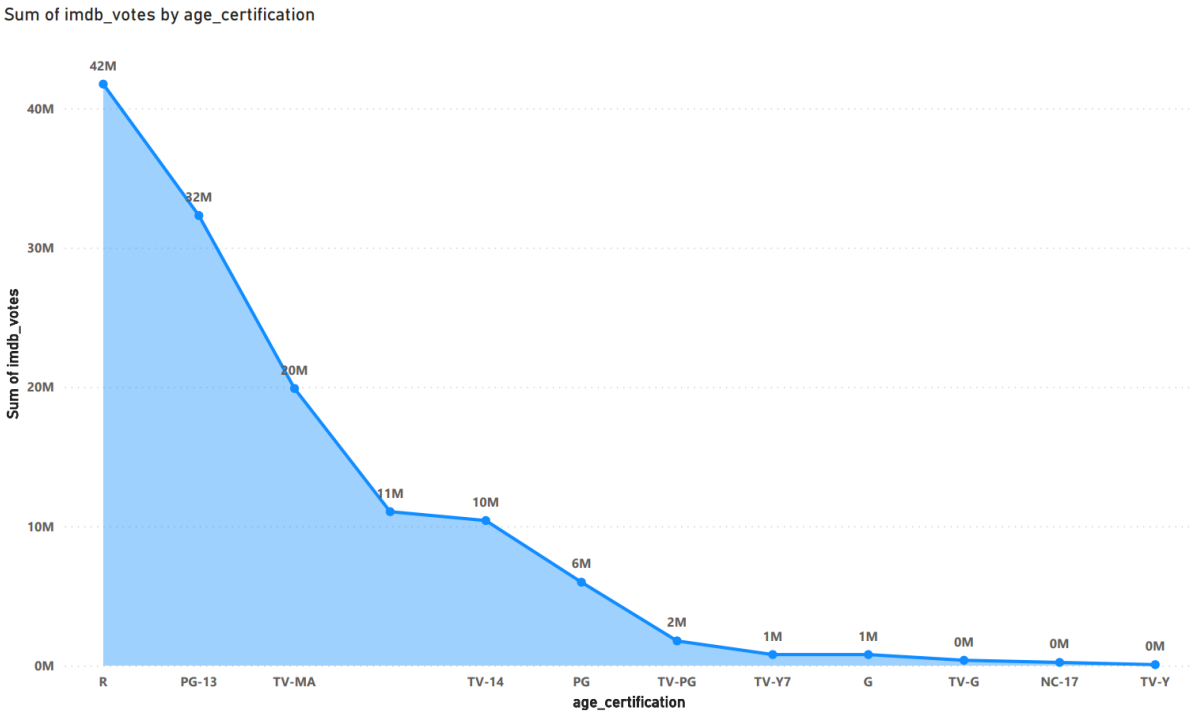


To know the answer for this, I have used pie chart visualization with release year as legend variable, Here the size of image is 695x887. Here, sky blue represents 2017(25.58%), dark blue is 2019(25.04%), orange is 2020 (16.37%), purple is 2021 (15.08%), pink is 2018(14.22%) and light purple is 2022(3.71%). The highest number of votes that were upcasted in 2017 over the last 5 years.



Here, for the same Question, I used another visual approach. Which is a Line chart. Where on x-axis it is year wise on a scale of 1960 to 2020, and on y-axis it is the sum of imdb votes on a scale of 0 to 10 million. It has a size of 889x912. As we can see, in 2016-2019, there are massive amounts of votes that have been upvoted.

3. Based on the age_certification category, At what rate, the imdb voting have been upvoted?



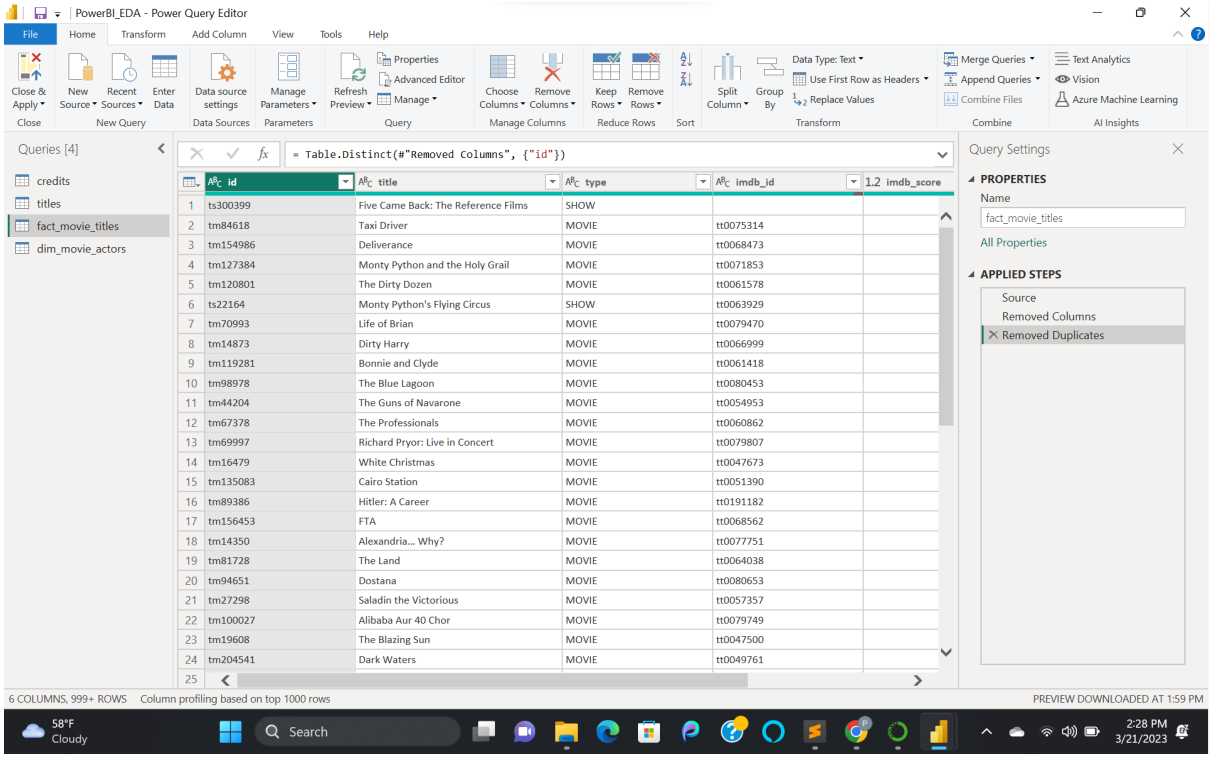
Here, I used the area chart, with age certification of different labels on x-axis and sum of imdb votings for each age_cert on y-axis on a scale of 0 to 40M. The size of the image is 694x791. As we can see, R age certificated movies have got the most imdb votes.

PART-2:

So, to display the visualizations for data understanding purposes, I have used PowerBI. It is interactive software used for data visualizations, which is developed by Microsoft by focusing on business intelligence. This tool is used to display the various kinds of visualizations, which will be helpful for decision making. Also helps to reveal patterns and relations between data points and to highlight interesting details. Also there are some issues related to this particular software. As it does not provide data cleaning solutions and might have some performance issues, limited sharing of the data and complex in nature.

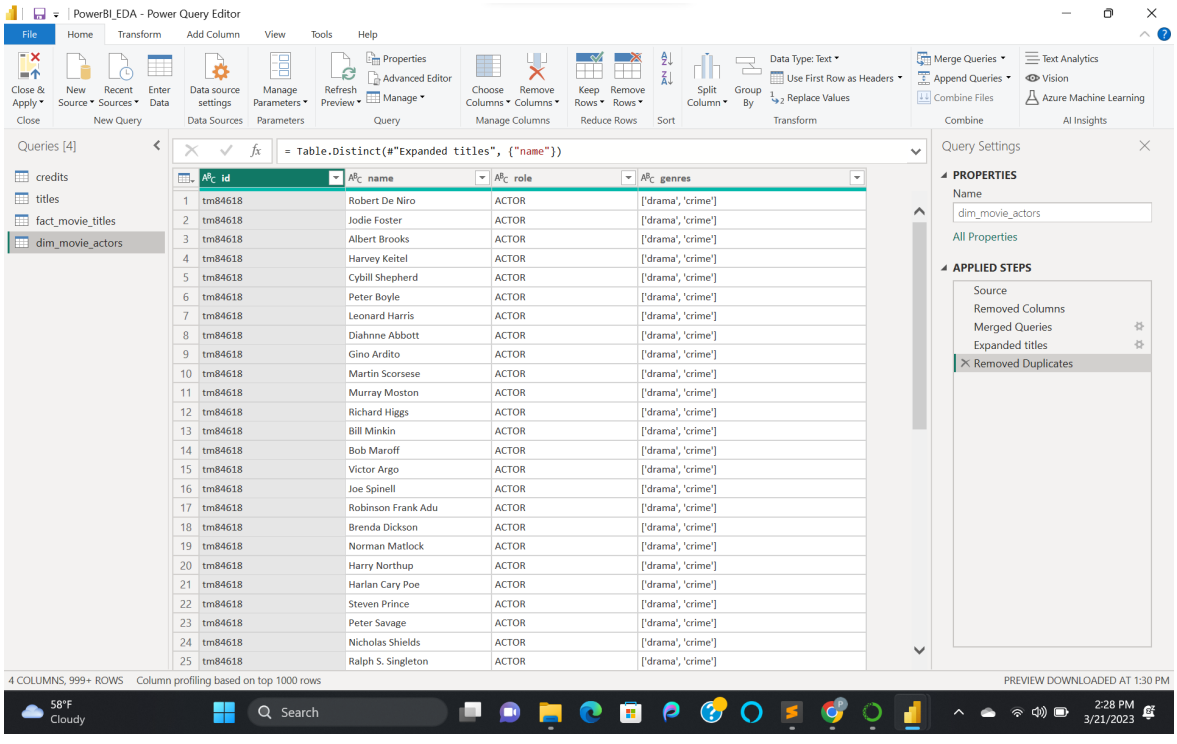
EXTRA CREDIT:

Here we have built the ETL process to transform the source data into star schema.



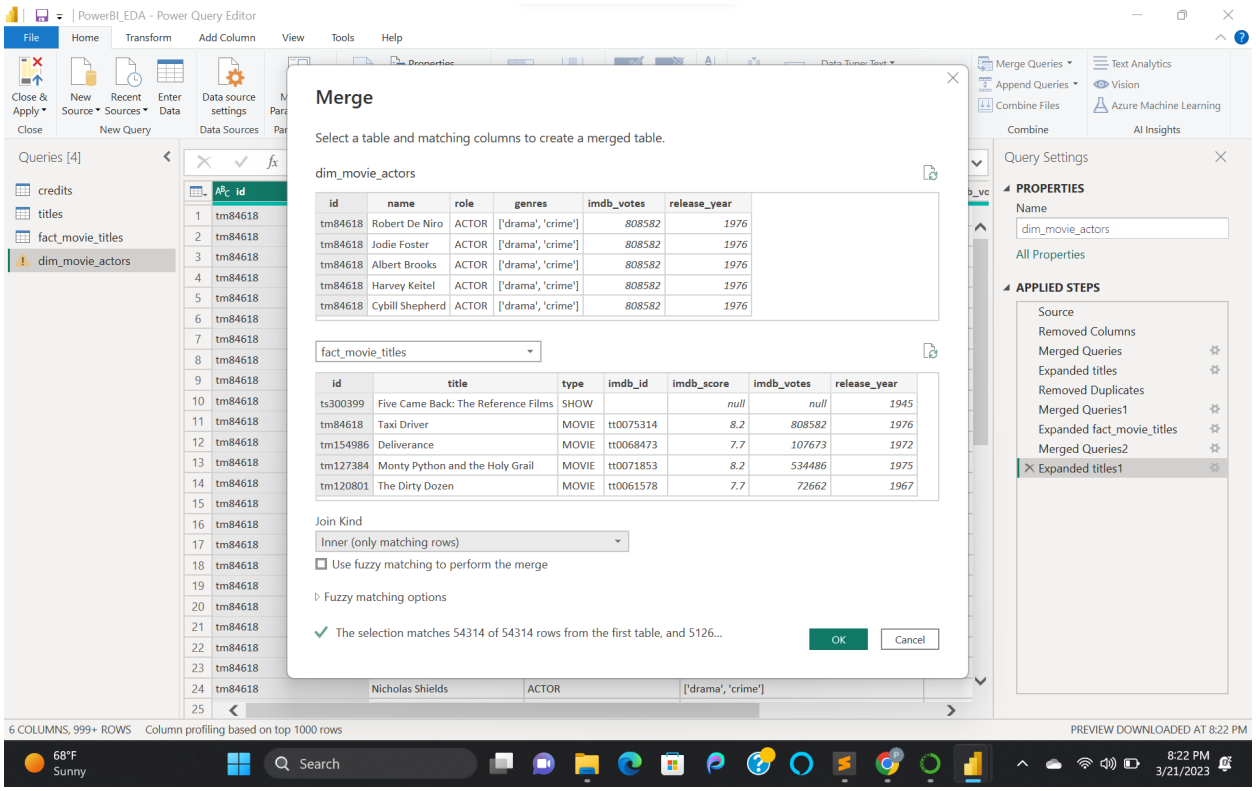
	id	title	type	imdb_id	imdb_score
1	ts300399	Five Came Back: The Reference Films	SHOW		
2	tm84618	Taxi Driver	MOVIE	tt0075314	
3	tm154986	Deliverance	MOVIE	tt0068473	
4	tm127384	Monty Python and the Holy Grail	MOVIE	tt0071853	
5	tm120801	The Dirty Dozen	MOVIE	tt0061578	
6	ts22164	Monty Python's Flying Circus	SHOW	tt0063929	
7	tm70993	Life of Brian	MOVIE	tt0079470	
8	tm14873	Dirty Harry	MOVIE	tt0066999	
9	tm119281	Bonnie and Clyde	MOVIE	tt0061418	
10	tm98978	The Blue Lagoon	MOVIE	tt0080453	
11	tm44204	The Guns of Navarone	MOVIE	tt0054953	
12	tm67378	The Professionals	MOVIE	tt0060862	
13	tm69997	Richard Pryor: Live in Concert	MOVIE	tt0079807	
14	tm16479	White Christmas	MOVIE	tt0047673	
15	tm135083	Cairo Station	MOVIE	tt0051390	
16	tm89386	Hitler: A Career	MOVIE	tt0191182	
17	tm156453	FTA	MOVIE	tt0068562	
18	tm14350	Alexandria... Why?	MOVIE	tt0077751	
19	tm81728	The Land	MOVIE	tt0064038	
20	tm94651	Dostana	MOVIE	tt0080653	
21	tm27298	Saladin the Victorious	MOVIE	tt0057357	
22	tm100027	Alibaba Aur 40 Chor	MOVIE	tt0079749	
23	tm19608	The Blazing Sun	MOVIE	tt0047500	
24	tm204541	Dark Waters	MOVIE	tt0049761	
25					

Here first, I went to transform data and then took reference of two tables and created new tables and then renamed them.



	id	name	role	genres
1	tm84618	Robert De Niro	ACTOR	['drama', 'crime']
2	tm84618	Jodie Foster	ACTOR	['drama', 'crime']
3	tm84618	Albert Brooks	ACTOR	['drama', 'crime']
4	tm84618	Harvey Keitel	ACTOR	['drama', 'crime']
5	tm84618	Cybill Shepherd	ACTOR	['drama', 'crime']
6	tm84618	Peter Boyle	ACTOR	['drama', 'crime']
7	tm84618	Leonard Harris	ACTOR	['drama', 'crime']
8	tm84618	Diahnne Abbott	ACTOR	['drama', 'crime']
9	tm84618	Gino Ardito	ACTOR	['drama', 'crime']
10	tm84618	Martin Scorsese	ACTOR	['drama', 'crime']
11	tm84618	Murray Moston	ACTOR	['drama', 'crime']
12	tm84618	Richard Higgs	ACTOR	['drama', 'crime']
13	tm84618	Bill Minkin	ACTOR	['drama', 'crime']
14	tm84618	Bob Maroff	ACTOR	['drama', 'crime']
15	tm84618	Victor Argo	ACTOR	['drama', 'crime']
16	tm84618	Joe Spinell	ACTOR	['drama', 'crime']
17	tm84618	Robinson Frank Adu	ACTOR	['drama', 'crime']
18	tm84618	Brenda Dickson	ACTOR	['drama', 'crime']
19	tm84618	Norman Matlock	ACTOR	['drama', 'crime']
20	tm84618	Harry Northup	ACTOR	['drama', 'crime']
21	tm84618	Harlan Cary Poe	ACTOR	['drama', 'crime']
22	tm84618	Steven Prince	ACTOR	['drama', 'crime']
23	tm84618	Peter Savage	ACTOR	['drama', 'crime']
24	tm84618	Nicholas Shields	ACTOR	['drama', 'crime']
25	tm84618	Ralph S. Singleton	ACTOR	['drama', 'crime']

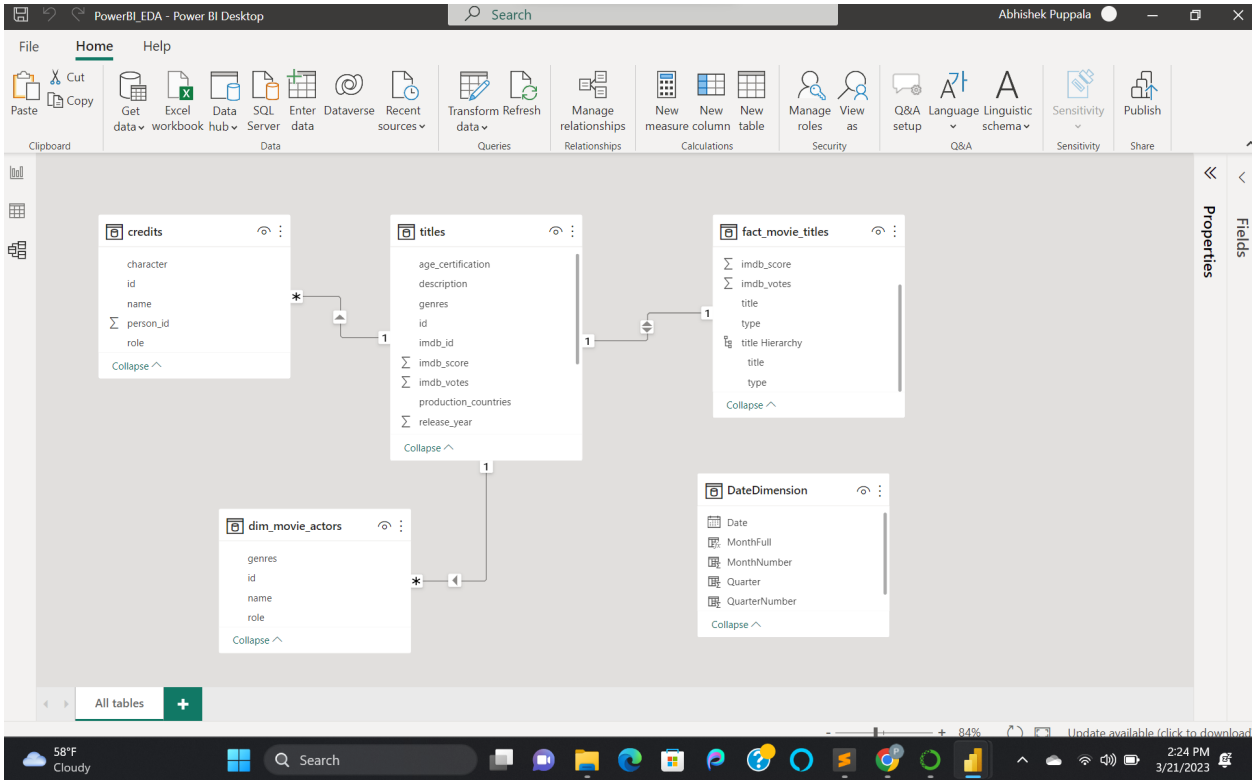
After renaming them, I have performed some of the data processing operations such as merging, filtering, and removing duplicates.



Here I have merged the titles data to dim_credits using a merge query based on id and selecting titles data with join kind as inner_join and then applied.

After applying the above operation, then I have filtered or selected the only needed data columns from titles table data. Then I have removed all duplicate rows from the table data. And also removed unnecessary columns.

We can see all the various operations performed under applied steps on the right side. After completing all the operations, then I have clicked on close and apply.



After all the preprocessing, the above star schema has been generated, which has dim_movie_actors and fact_movie_titles.

ETL source codes :

Fact_movie_titles:

removed cols = Table.RemoveColumns(Source,{"release_year", "age_certification", "runtime", "genres", "production_countries", "seasons", "tmdb_popularity", "tmdb_score", "description"})

Removed_dups = Table.Distinct(#"Removed Columns", {"id"})

Dim_movie_actors:

removed cols = Table.RemoveColumns(Source,{"person_id", "character"})

Join inner = Table.NestedJoin(#"Removed Columns", {"id"}, titles, {"id"}, "titles", JoinKind.Inner)

```
expanded title = Table.ExpandTableColumn(#"Merged Queries", "titles", {"genres"}, {"genres"})

removed dups = Table.Distinct(#"Expanded titles", {"name"})
```