NETFLIX

DATABASE

Who's presenting?

Nehmat Touma
201600300

Mahdi Hallal
201906289

Pia Chouaifaty
201504706

Presented to
Dr. Haidar Harmanani
CSC375
Database Management Systems

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

Table of Contents

Abstract

We present a “Netflix Database” that streamlines the process of choosing a movie/TV Show to watch for the average user by providing a user-friendly interface whereby the user can select movies/TV Shows based on genre, views, decade, ratings from several rating websites as well as type of awards won. We combined the data from several publicly available datasets using R and built the backend using HTML and PHP, and the frontend user interface using HTML, JavaScript and CSS, yielding a web interface that can be easily browsed and accessed. SQL was the database management system used.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

I. Introduction

Streaming services have replaced traditional media as the primary forms of entertainment consumption and drivers of market trends. Netflix, a major player with 87% market share in the USA (Netflix Revenue and Usage Statistics, 2020) is the main streaming service people sign up for.

Netflix users start watching specific shows for one or more of the following reasons:

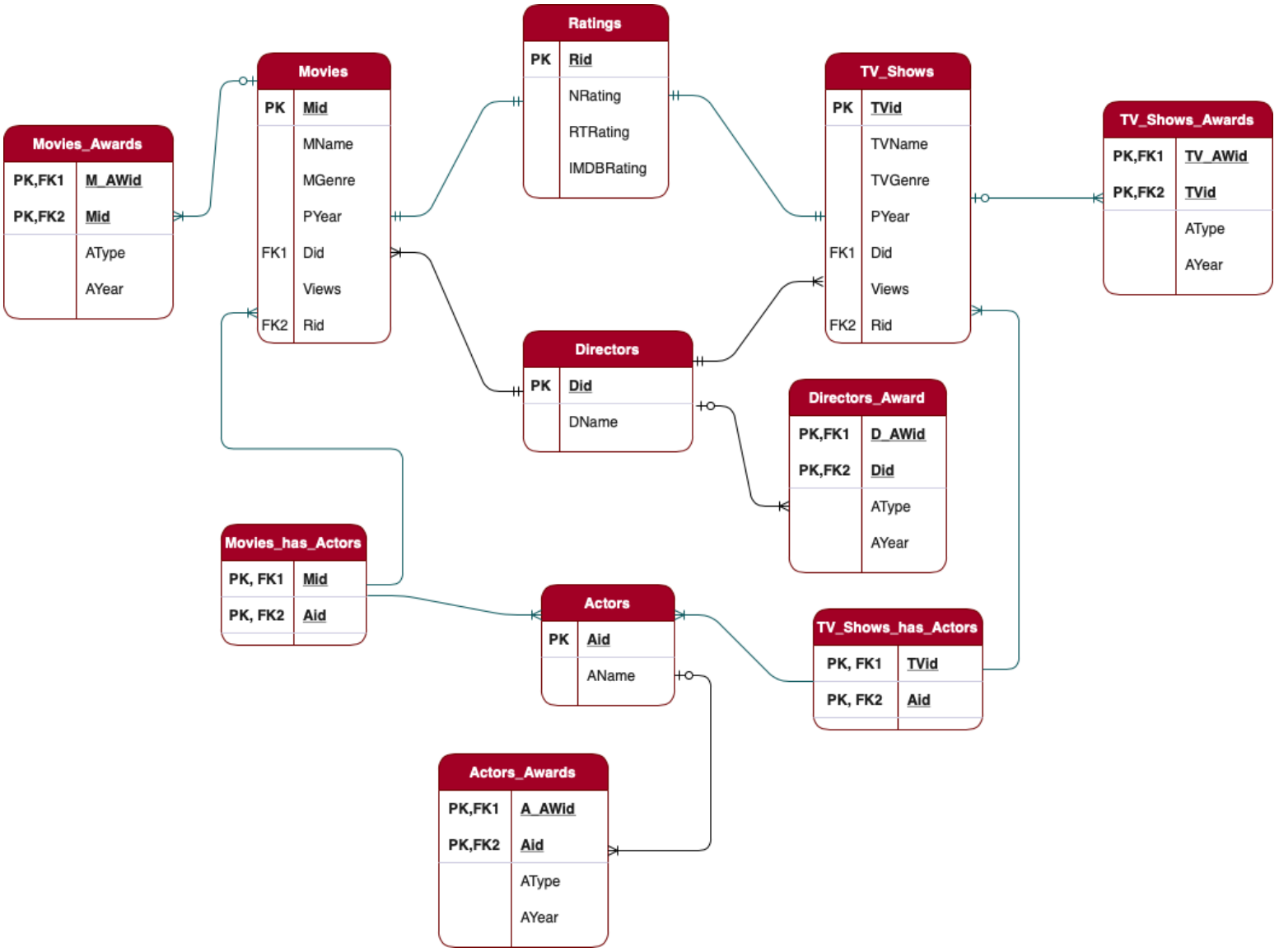
- While looking for a specific genre
- Looking for a specific movie/TV Show/Documentary
- After seeing a suggestion in the homepage
- A recommendation from someone

However, the Netflix platform itself is not geared towards allowing the user complete control over his/her viewing choices. Cross-platform ratings (IMDB, Rotten Tomatoes, Netflix user ratings) are as of now not available on Netflix, meaning the user needs to access the Rotten Tomatoes and IMDB websites to be able to view these ratings. Moreover, even rating websites such as RT and IMDb fail to provide a way for users to browse only ratings for titles that are available on Netflix. In addition, despite the Netflix platform including an “award-winning” filter, it does not let users select titles based on the specific type of award (Golden Globe, Oscar, Emmy…). Searching for movies on Award Academy websites is a hassle, and even then, no information on Netflix availability is provided. Also, Netflix does not have an appropriate “decade” date filter for titles nor a way to sort titles by popularity (views).

Thus, our database successfully solves these shortcomings by condensing all this scattered data into a user-friendly web interface that lets users browse award-winning movies and TV Shows by type of award and ratings by platform. Additionally, it offers genre filters.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

II. ER Diagram



Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

III. Schema

Movies (Mid, MName, MGenre, PYear, Did, Views, Rid)

TV_Shows (TVid, TVName, TVGenre, PYear, Did, Views, Rid)

Actors (Aid, AName)

Directors (Did, AName)

Ratings (Rid, NRating, RTRating, IMDBRating)

TV_Shows_Awards (TV_AWid, TVid, AType, AYear)

Movie_Awards (M_AWid, Mid, AType, AYear)

Actors_Awards (A_AWid, Aid, AType, AYear)

Directors_Awards (D_AWid, Did, AType, AYear)

Movies_has_Actors (Mid, Aid)

TV_Shows_has_Actors (TVid, Aid)

IV. Business Rules

The Netflix Database contains data about movies, TV shows, actors, directors, ratings, and awards. A movie has a unique id, name, genre, premier year, director, views, and ratings. Also, a TV show has a unique id, name, genre, premier year, director, views, and ratings. An actor has a unique id and a name. Similarly, a director has a unique id and a name. Ratings consist of ratings for TV shows and movies on Netflix, Rotten Tomatoes, and IMDB. Each of the movies, TV shows, actors, and directors have their own awards that are characterized by a unique id the movie, TV show, actor, or director that received such award, the name of the award, and the year of receiving such an award. A movie and a TV show have many actors acting in it, such that an actor can act in many movies and TV shows. Moreover, a movie and

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

a TV show have one director directing them, however, a director can direct many movies and TV shows. For the ratings, each movie and TV show have a certain rating on Netflix, Rotten Tomatoes, and IMDB. Any movie, TV show, actor, or director can win multiple awards, but the award must be won by a movie, TV show, actor, or director.

V. Entities

Movies: a list of the movies available for streaming on Netflix.

- **Mid** (primary key): A unique, 5-digit identifier
- MName: Movie name
- MGenre: The genre/category of the movie
- PYear: Premiere year or alternatively, release year for non-theatre titles
- Did (foreign key from *Directors*): Director ID
- Views: The number of times the movie has been viewed from the time it was added to Netflix until then end of November 2020.
- Rid (foreign key from *Ratings*): The Rating ID of the movie's corresponding entry in the Ratings table.

TV_Shows: a list of the TV Shows available for streaming on Netflix.

- **TVid** (primary key): A unique, 5-digit identifier
- TVName: TV show name
- MGenre: The genre/category of the TV Show
- PYear: release year (of the first season)
- Did (foreign key from *Directors*): Director ID
- Views: The number of times the TV Show has been viewed from the time it was added to Netflix until then end of November 2020.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

- Rid (foreign key from *Ratings*): The Rating ID of the TV Show's corresponding entry in the Ratings table.

Actors: a list of the actors with titles on Netflix.

- **Aid** (primary key): A unique, 5-digit identifier
- AName: Actor name

Directors: a list of the directors with titles on Netflix.

- **Did** (primary key): A unique, 5-digit identifier
- DName: Director name

Ratings: a list of rating entries for each title

- Rid (primary key): A unique, 5-digit identifier
- NRating: Netflix user ratings
- RTRating: Tomatometer rating as per Rotten Tomatoes
- IMDBRating: Rating as per IMDb

TV_Shows_Awards: A list of awards won by TV Shows

- TV_AWid (primary key): Award ID
- TVid: TV Show ID
- AType: One of two types of awards – Emmy or Golden Globe
- AYear: Year received

Movie_Awards: A list of awards won by movies

- M_AWid (primary key): Award ID
- Mid: Movie ID
- AType: One of two types of awards – Oscar or Golden Globe
- AYear: Year received

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

Actors_Awards: A list of awards won by actors

- A_AWid (primary key): Award ID
- Aid: Actor ID
- AType: One of three types of awards – Oscar, Golden Globe or Emmy
- AYear: Year received

Directors_Awards: A list of awards won by directors

- D_AWid (primary key): Award ID
- Did: Director ID
- AType: One of three types of awards – Oscar, Golden Globe or Emmy
- AYear: Year received
-

VI. Relationships

- **Movies_has_Actors:**
  It is a many-to-many relationship between movies and actors. A single movie has many actors acting in it, and a single actor can be part of many movies. In addition, there is a total participation constraint on the participation of both entities in this relationship as each movie must have at least one actor in it, and each actor must be part of at least one movie.
- **TV_Shows_has_Actors:**
  It is a many-to-many relationship between TV shows and Actors. A single TV show has many actors acting in it, and a single actor can be part of many TV shows. In addition, there is a total participation constraint on the participation of both entities in this relationship as each TV show must have at least one actor in it, and each actor must be part of at least one TV show.
- **Movies_has_Awards:**

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

It is a one-to-many relationship between Movies and Movies_Awards. The one referring to Movies and the many referring to Movies_Awards. A single movie can have many awards; however, an award is won by a single movie. In addition, there is partial participation constraint on Movies and a total participation constraint on Movies_Awards in this relationship. A movie may or may not win an award, and the movie award must have a movie that won it.

- **TV_Shows_has_Awards:**

  It is a one-to-many relationship between TV_Shows and TV_Shows_Awards. The one referring to TV_Shows and the many referring to TV_Shows_Awards. A single TV show can have many awards; however, an award is won by a single TV show. In addition, there is partial participation constraint on TV_Shows and a total participation constraint on TV_Shows_Awards in this relationship. A TV show may or may not win an award, and the award must have a TV show that won it.

- **Directors_has_Awards:**

  It is a one-to-many relationship between Directors and Directors_Award. The one referring to Directors and the many referring to Directors_Award. A single director can have many awards; however, an award is won by a single director. In addition, there is partial participation constraint on Directors and a total participation constraint on Directors_Award in this relationship. A director may or may not win an award, and the award must have a director that won it.

- **Actors_has_Awards:**

  It is a one-to-many relationship between Actors and Actors_Awards. The one referring to Actors and the many referring to Actors_Awards. A single actor can have many awards; however, an award is won by a single actor. In addition, there is partial participation constraint on Actors and a total participation constraint on Actors_Awards in this relationship. An actor may or may not win an award, and the award must have an actor that won it.

- **Movies_has_Directors:**

  It is a one-to-many relationship between Directors and Movies. The one referring to Directors and the many referring to Movies. A single director can direct many

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

movies; however, a movie must be directed by a single director. In addition, there are total participation constraints on both entities in this relationship. A director must direct at least one movie, and a movie must have at least one director.

- **TV_Shows_has_Directors:**

  It is a one-to-many relationship between Directors and TV_Shows. The one referring to Directors and the many referring to TV_Shows. A single director can direct many TV shows; however, a TV show must be directed by a single director. In addition, there are total participation constraints on both entities in this relationship. A director must direct at least one TV show, and a TV show must have at least one director.

- **Movies_has_Ratings:**

  It is a one-to-one relationship between Movies and Ratings. A movie has exactly one rating, and a rating corresponds to exactly one movie (Please refer to the Ratings part in the Entities section). In addition, there are total participation constraints on both entities in this relationship. A movie must have a rating, and a rating must correspond to a movie.

- **TV_Shows_has_Ratings:**

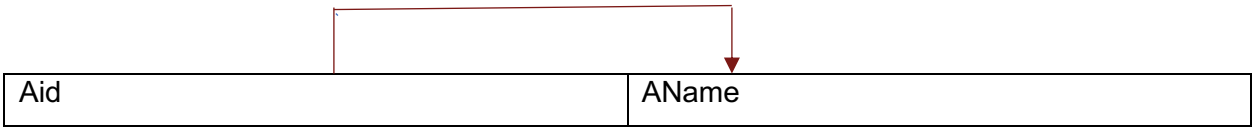
  It is a one-to-one relationship between TV_Shows and Ratings. A TV show has exactly one rating, and a rating corresponds to exactly one TV show (Please refer to the Ratings part in the Entities section). In addition, there are total participation constraints on both entities in this relationship. A TV show must have a rating, and a rating must correspond to a TV show.

VII. Normalization up to BCNF

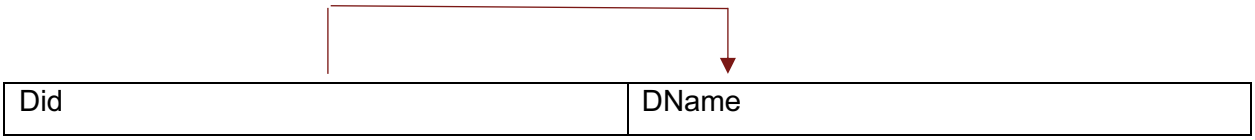
The normal form that was targeted was Boyce-Codd Normal Form, BCNF. After the mapping of ER into relations, it turned out that the schema obtained in every relation was already in BCNF. Every relation's attributes were determined by the key attribute and by themselves (trivial dependency) only. The below is an illustration of how the resulting schema in each relation is in BCNF form:

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

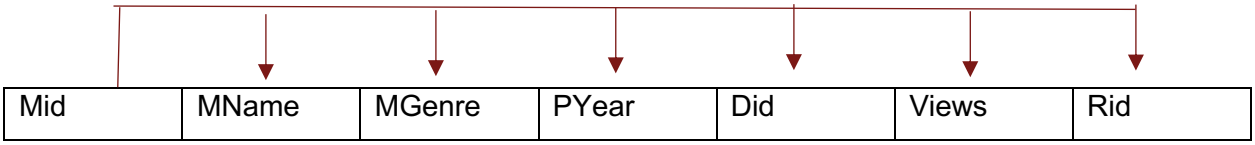
Actors:



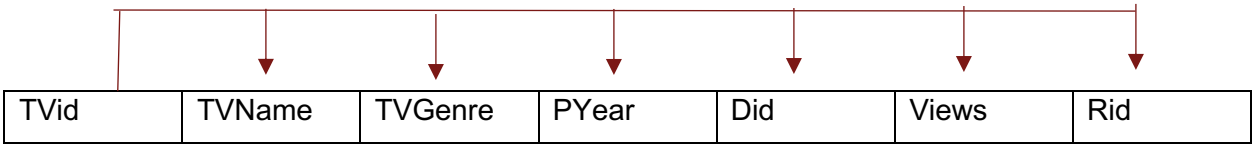
Directors:



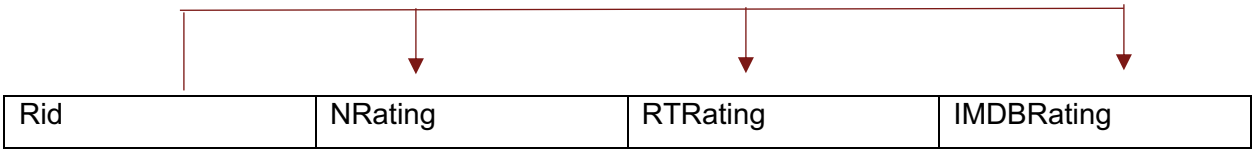
Movies:



TV Shows:



Ratings:



Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

TV_Shows_Awards:

TV_AWid	TVid	AType	AYear

Movie_Awards:

M_AWid	Mid	AType	AYear

Actors_Awards:

A_AWid	Aid	AType	AYear

Directors_Awards:

D_AWid	Did	AType	AYear

Movies_has_Actors:

Mid	Aid

TV_Shows_has_Actors:

TVid	Aid

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

All of the above relations are clearly in BCNF as they follow the principle of "The key, the whole key, and nothing but the key, so help me Codd". It is worth mentioning that in the relations Movies_has_Actors and TV_Shows_has_Actors have no arrows since the combination of each relation's attribute is a primary key.

VIII. Queries

The queries were tackled from two fronts, the first including movies and TV shows and the second actors and directors, i.e. the queries applied to movies were applied to TV shows, and the same for actors and directors. Same attribute output were applied to different entities and are detailed below; as such, duplicated queries - in logic - will be included in the appendix.

- Displaying **one of three** ratings of **Movies/TV Shows** ordered by **genre**:
  (Done for ratings of: IMDB, Netflix and Rotten Tomatoes, Appendix B)

```
SELECT r.IMDBrating, tv.TVName, tv.TVGenre
FROM tv_shows tv, ratings r
WHERE tv.Rid=r.Rid
GROUP BY tv.TVGenre
```

- Displaying the **Movies/TV Shows** that premiered in a **specific decade:**
  (From 1940s-2000s for movies and 1990s and 2000s for TV Shows, in line with the data we are working with, Appendix C)

```
SELECT *
FROM movies
WHERE PYear>1939 AND PYear<1950;
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

- Displaying **Movies/TV Shows** per **genre**:
  (16 genres for Movies and for 11 TV Shows, Appendix D)

```
SELECT *
FROM movies
WHERE MGenre='Comedies';
```

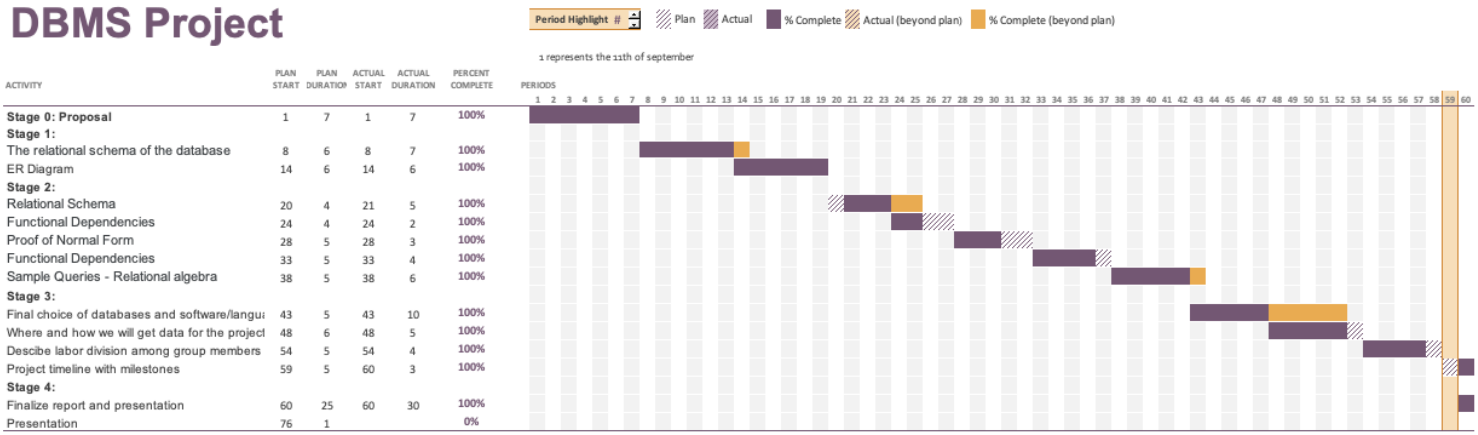
- Displaying **Movies/TV Shows/Actors/Directors** that got a specific award with its corresponding **year,** ordered **from most recent to least recent**:
  (Oscar and Golden Globes for movies and Emmy and Golden Globe for TV Shows Oscar, Golden Globe and Emmy for both Actors and Directors, Appendix E)

```
SELECT a.AName, aa.AYear
FROM actors a , actors_awards aa
WHERE a.Aid=aa.Aid AND aa.AType='Oscar'
ORDER BY aa.AYear DESC;
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

IX. Methodology of Implementation

Timeline



Data Collection & Clean-up

The raw data was gathered from Kaggle and other sources; the links are included in Appendix A.

Initially we had:

- A dataset with all available titles on Netflix (movies & TV shows)
- A dataset of Netflix user ratings for some TV Shows
- A dataset with Golden Globe nominees and winners (Movies, TV, Directors and Actors)
- A dataset with Oscar nominees and winners (Movies, Directors and Actors)
- A dataset with Emmy Globe nominees and winners (Movies, Directors and Actors)

The raw datasets were imported into RStudio, where the table with the list of titles available on Netflix was used as a primary reference.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

Rows with missing cast/directors were eliminated, and out of all the columns, only the relevant ones (genre, title, director, cast, release year) were retained.

In line with keeping all value atomic, we kept only the first genre in rows where titles had more than 1 genre listed, using regular expressions.

Next, the Netflix titles dataset was split into two separate tables: Movies and TV_Shows. Some entries were duplicated, so we also handled that by keeping only unique entries.

We assigned randomly generated unique IDs to each of Movies and TV Shows.

We extract the director names into the Directors table.

Initially, actors are provided as a list of names in the column cast for each title. We separate them using regular expressions, and add them to the Actors table, keeping only unique entries.

We then randomly assign unique IDs to each entry in Actors and Directors.

Views are simulated using random numbers within a certain range.

We joined the available Netflix user ratings from the user ratings dataset to each of the Movies and TV Shows tables. For the Rotten Tomatoes rating, which is usually on a scale of 1-100, we simulated it using the following formula:

$$RT\ Rating = user\ rating - (random\ number\ between\ 10\ and\ 20)$$

Because Rotten Tomatoes ratings are usually critic ratings, they tend to be lower than user ratings, hence the subtraction.

For the IMDB ratings, we simulated them using the following formula:

$$IMDb\ Rating = \frac{RT\ Rating + (random\ number\ between\ 5\ and\ 8)}{10}$$

IMDb ratings are on a scale of 1-10 and usually more generous than Rotten Tomatoes.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

We assign randomly generated Rids to the Ratings and match them to their corresponding Movie/TV Show by name.

The Oscars, Emmys and Golden Globes datasets included winners as well as nominees, so we kept only the winners, also removing all columns that are irrelevant for our purposes.

Moreover, the awards datasets did not include specifically whether the recipient is a movie, TV Show, actor or director; they simply listed the "recipient name". For this reason, we compiled a list of "terms" in award categories that we know to be specific to movies: i.e ("score," "screenplay," "cinematography"…), actors ("actor," "actress," "supporting role," "leading role"…) and so on, customized to each award. We used the presence of terms in these lists to correctly categorize award recipients as either movies, actors, directors or TV Shows.

Afterwards, we also assigned randomly generated unique IDs to the Awards tables.

We manipulated the data to be left with tables exactly matching the structure of our schema.

The full R code, as well as the raw data and finalized data are available in the supplementary material.

Graphical User Interface (GUI)

In order to visualize the data collected, we constructed a website that displays the information the user wishes to access.
The files are available in the supplementary material.

a. Back-End

In the back-end we decide which information we want to extract from our database using SQL, which was coded using Php, which in turn enables the front end to get access to said output. A total of 52 SQL queries were applied on the data, discussed in the Queries section.

b. Front-End

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

To bring the Netflix Database to life, the front end is essential. We used HTML and CSS and JavaScript to design and refine the web interface. CSS was responsible for the design aesthetics with HTML bringing it to the surface and finally, JavaScript applied a function on the output of the queries to specify the information needed to be displayed for the user.

X. Conclusion and Limitations

The specific collection of data outlined above is as of now unavailable in a user-friendly format that the average person would use. A possible exception is the format of many search engine results (e.g. Google) for a movie/series query which outline the title, creators, ratings on IMDb, Rotten Tomatoes as well as user ratings (Google users submit ratings through their Google accounts). Search engine results also provide suggested titles based on creators/genres. It might be more intuitive to search for a title using a search engine, but if the user is interested in making a selection depending on specific criteria (cast/creators, ratings, popularity), they will find it convenient to use our interface. Other rating platforms (IMDb, Rotten Tomatoes…) do not have information about each other, nor about Netflix user ratings and also lack the functionality to sort by best rated title for genre/cast member/crew etc. They also do not provide time-course data, which may be of less importance to the average person but provides added value. Our model fills the gaps in the aforementioned areas. However, one shortcoming of our model is the fact that the data is not dynamic and will need to be manually updated. This is not the case for other platforms.

Our main limitation was the unavailability of some data, which we simulated to the best of our ability, but will still not be completely representative of reality. Potential improvements include the use of web scraping and/or access to more comprehensive datasets.

Potential future applications include the possibility of designing an add-on or extension for the browser version of Netflix that would provide additional information to the user on titles they might be looking at. Additionally, there is potential for expansion to include other streaming services' such as Hulu, Amazon Prime, HBO+, etc. repertoires. Moreover,

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

given additional time-course data on viewership trends, a predictive approach may be able to yield important insights into streaming services and their viewership trends.

Overall, we delivered a promising and well-designed database, with clean data and user-friendly GUI.

Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

Appendix A: Links to Datasets

- https://www.kaggle.com/unanimad/the-oscar-award
- https://github.com/ygterl/EDA-Netflix-2020-in-R
- https://www.kaggle.com/vikassingh1996/netflix-movies-and-shows-plotly-recommender-sys
- https://www.kaggle.com/ruchi798/tv-shows-on-netflix-prime-video-hulu-and-disney
- https://www.kaggle.com/chasewillden/netflix-shows
- https://www.kaggle.com/unanimad/emmy-awards?select=the_emmy_awards.csv
- https://www.kaggle.com/unanimad/golden-globe-awards

Appendix B: RATINGS (IMDb – Rotten Tomatoes – Netflix)

TV Shows:

```
SELECT r.IMDBrating, tv.TVName, tv.TVGenre
FROM tv_shows tv, ratings r WHERE tv.Rid=r.Rid
GROUP BY tv.TVGenre
```

```
SELECT r.RTrating, tv.TVName, tv.TVGenre
FROM tv_shows tv, ratings r
WHERE tv.Rid=r.Rid
GROUP BY tv.TVGenre;
```

```
SELECT r.Nrating, tv.TVName, tv.TVGenre
FROM tv_shows tv, ratings r
WHERE tv.Rid=r.Rid
GROUP BY tv.TVGenre;
```

Movies:

```
SELECT r.IMDBrating, m.MName, m.MGenre
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

```
FROM movies m, ratings r
WHERE m.Rid=r.Rid
GROUP BY m.MGenre;

SELECT r.RTrating, m.MName, m.MGenre
FROM movies m, ratings r
WHERE m.Rid=r.Rid
GROUP BY m.MGenre


SELECT r.Nrating, m.MName, m.MGenre
FROM movies m, ratings r
WHERE m.Rid=r.Rid
GROUP BY m.MGenre;
```

Appendix C: DISPLAY PER DECADE

Movies:

```
SELECT *
FROM movies
WHERE PYear>1939 AND PYear<1950;

SELECT *
FROM movies
WHERE PYear>1999 AND PYear<2020;

SELECT *
FROM movies
WHERE PYear>1989 AND PYear<2000;
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

```
SELECT *
FROM movies
WHERE PYear>1979 AND PYear<1990;

SELECT *
FROM movies
WHERE PYear>1969 AND PYear<1980;

SELECT *
FROM movies
WHERE PYear>1959 AND PYear<1970;

SELECT *
FROM movies
WHERE PYear>1949 AND PYear<1960;
```

TV Shows:

```
SELECT *
FROM tv_shows
WHERE PYear>1989 AND PYear<2000;

SELECT *
FROM tv_shows
WHERE PYear>1979 AND PYear<1990;
```

Appendix D: BY GENRE

Movies:

```
SELECT *
FROM movies
WHERE MGenre='Thrillers';

SELECT *
FROM movies
WHERE MGenre='Stand-Up Comedy';

SELECT *
FROM movies
WHERE MGenre='Sci-Fi & Fantasy';

SELECT *
FROM movies
WHERE MGenre='Romantic Movies';

SELECT *
FROM movies
WHERE MGenre='Music & Musicals';

SELECT *
FROM movies
WHERE MGenre='International Movies';
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

```
SELECT *
FROM movies
WHERE MGenre='Independent Movies';

SELECT *
FROM movies
WHERE MGenre='Horror Movies';

SELECT *
FROM movies
WHERE MGenre='Children & Family Movies';

SELECT *
FROM movies
WHERE MGenre='Dramas';

SELECT *
FROM movies
WHERE MGenre='Documentaries';

SELECT *
FROM movies
WHERE MGenre='Cult Movies';

SELECT *
FROM movies
WHERE MGenre='Comedies';

SELECT *
FROM movies
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

```
WHERE MGenre='Classic Movies';

SELECT *
FROM movies
WHERE MGenre='Anime Features';

SELECT *
FROM movies
WHERE MGenre='Action & Adventure';
```

TV Shows:

```
SELECT *
FROM tv_shows
WHERE TVGenre= 'Kids\' TV';

SELECT *
FROM tv_shows
WHERE TVGenre='International TV Shows';

SELECT *
FROM tv_shows
WHERE TVGenre='Anime Series';

SELECT *
FROM tv_shows
WHERE TVGenre='British TV Shows';

SELECT *
FROM tv_shows
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

```
WHERE TVGenre='Classic & Cult TV';

SELECT *
FROM tv_shows
WHERE TVGenre='Crime TV Shows';

SELECT *
FROM tv_shows
WHERE TVGenre='Docuseries';

SELECT *
FROM tv_shows
WHERE TVGenre='Stand-Up Comedy & Talk Shows';

SELECT *
FROM tv_shows
WHERE TVGenre='TV Action & Adventure';

SELECT *
FROM tv_shows
WHERE TVGenre='TV Comedies';

SELECT *
FROM tv_shows
WHERE TVGenre='TV Dramas';
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

Appendix E: AWARDS

TV Shows:

```
SELECT tv.TVName, tva.AYear
FROM tv_shows tv, tv_show_awards tva
WHERE tv.TVid=tva.TVid AND tva.AType='Golden Globe'
ORDER BY tva.AYear DESC;
```

```
SELECT tv.TVName, tva.AYear
FROM tv_shows tv, tv_show_awards tva
WHERE tv.TVid=tva.TVid AND tva.AType='Emmy'
ORDER BY tva.AYear DESC;
```

Movies:

```
SELECT m.MName, ma.AYear
FROM movies m, movies_awards ma
WHERE m.Mid=ma.Mid AND ma.AType='Oscar'
ORDER BY ma.AYear DESC;
```

```
SELECT m.MName, ma.AYear
FROM movies m, movies_awards ma
WHERE m.Mid=ma.Mid AND ma.AType='Golden Globe'
ORDER BY ma.AYear DESC;
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty

Actors:

```
SELECT a.AName, aa.AYear
FROM actors a , actors_awards aa
WHERE a.Aid=aa.Aid AND aa.AType='Oscar'
ORDER BY aa.AYear DESC;

SELECT a.AName, aa.AYear
FROM actors a , actors_awards aa
WHERE a.Aid=aa.Aid AND aa.AType='Golden Globe'
ORDER BY aa.AYear DESC;

SELECT a.AName, aa.AYear
FROM actors a , actors_awards aa
WHERE a.Aid=aa.Aid AND aa.AType='Emmy'
ORDER BY aa.AYear DESC;
```

Directors:

```
SELECT d.DName, da.AYear
FROM directors d , directors_awards da
WHERE d.Did=da.Did AND da.AType='Oscar'
ORDER BY da.AYear DESC;

SELECT d.DName, da.AYear
FROM directors d , directors_awards da
WHERE d.Did=da.Did AND da.AType='Golden Globe'
ORDER BY da.AYear DESC;

SELECT d.DName, da.AYear
```

```
FROM directors d , directors_awards da
WHERE d.Did=da.Did AND da.AType='Emmy'
ORDER BY da.AYear DESC;
```


Nehmat Touma
Mahdi Hallal
Pia Chouaifaty